

UNIVERSITY OF THE PHILIPPINES MANILA
COLLEGE OF ARTS AND SCIENCES
DEPARTMENT OF PHYSICAL SCIENCES AND MATHEMATICS

GEOTECHNICAL INVESTIGATION REPORTS
GENERATOR FOR LA GLOBAL EDCC

A special problem in partial fulfillment
of the requirements for the degree of
Bachelor of Science in Computer Science

Submitted by:

Earl Andrea G. Abad

May 2016

Permission is given for the following people to have access to this SP:

Available to the general public	Yes
Available only after consultation with author/SP adviser	No
Available only to those bound by confidentiality agreement	No

ACCEPTANCE SHEET

The Special Problem entitled “Geotechnical Investigation Reports Generator for LA Global EDCC” prepared and submitted by Earl Andrea G. Abad in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

Ma. Sheila A. Magboo, M.Sc.
Adviser

EXAMINERS:

	Approved	Disapproved
1. Gregorio B. Baes, Ph.D. (<i>candidate</i>)	_____	_____
2. Avegail D. Carpio, M.Sc.	_____	_____
3. Richard Bryann L. Chua, M.Sc.	_____	_____
4. Perlita E. Gasmien, M.Sc. (<i>candidate</i>)	_____	_____
5. Marvin John C. Ignacio, M.Sc. (<i>cand.</i>)	_____	_____
6. Vincent Peter C. Magboo, M.D., M.Sc.	_____	_____

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

<hr/> Ma. Sheila A. Magboo, M.Sc. Unit Head Mathematical and Computing Sciences Unit Department of Physical Sciences and Mathematics	<hr/> Marcelina B. Lirazan, Ph.D. Chair Department of Physical Sciences and Mathematics
---	---

Leonardo R. Estacio Jr., Ph.D.
Dean
College of Arts and Sciences

Abstract

Geotechnical Investigation Reports Generator is a web-based computational and information system that computes necessary calculations and generates geotechnical investigation reports for LA Global EDCC's site investigation projects.

Keywords: geotechnical investigation report, soil testing, borehole

Contents

Acceptance Sheet	i
Abstract	ii
List of Figures	v
List of Tables	vii
I. Introduction	1
A. Background of the Study	1
B. Statement of the Problem	2
C. Objectives of the Study	2
D. Significance of the Project	5
E. Scope and Limitations	5
F. Assumptions	5
II. Review of Related Literature	7
A. Boreholes and Borehole Logging	7
B. Site Investigations using Boreholes	8
C. Related Systems	10
1. gINT Logs	10
2. Strater [®] 4	10
3. LogPlot	10
4. LOGDRAFT	11
III. Theoretical Framework	12
A. Geotechnical Investigation	12
1. Borehole Log	13
2. Soil Profile of Borehole	14
3. Liquid and Plastic Limits Test Report	14
4. Grain Size Analysis	15

B.	Soil Bearing Capacity for Foundations	16
1.	Shallow Foundation	16
2.	Deep Foundation	17
3.	Drilled Shaft Foundation	18
C.	Laravel	19
IV.	Design and Implementation	20
A.	Use Case Diagram	20
B.	Context Diagram	21
C.	Entity Relationship Diagram	21
D.	Data Dictionary	23
E.	System Architecture	26
F.	Technical Architecture	26
V.	Results	27
A.	General View	27
B.	Admin View	36
VI.	Discussions	42
VII.	Conclusions	43
VIII.	Recommendations	44
IX.	Bibliography	45
X.	Appendix	50
XI.	Acknowledgement	240

List of Figures

1	Example of borehole log [21]	14
2	Particle size ranges [29]	15
3	Example of particle size distribution curves [29]	16
4	Use Case Diagram	20
5	Context Diagram	21
6	Entity Relationship Diagram	22
7	Login Page	27
8	Registration Page	28
9	Edit User Profile	28
10	View Project Information	29
11	View Project Information	29
12	Edit Project Information	30
13	Add Borehole Information	30
14	Add Sample Information	31
15	Manage Borehole and Its Samples	31
16	View Detailed Sample Information	32
17	View Grain Size Analysis	32
18	View Liquid and Plastic Limits Test Report	33
19	View Borehole Log	33
20	View Soil Profile	34
21	View Existing Sections and Add New Section	35
22	View Section	35
23	Edit Section	36
24	View Compiled Report	36
25	Manage Users	37
26	Manage Users	37
27	Manage Projects	38
28	Add or Remove Engineers from Project	38

29	Receive Notifications	39
30	Archive Tagged Project	39
31	Create Email (with attached compiled report)	39
32	Edit Shallow Foundation	40
33	Edit Deep Foundation	40
34	Edit Drilled Shaft Foundation	41

List of Tables

1	users - contains both the system administrator (lead engineer) and the engineers	23
2	projects - contains the company's site investigation projects	23
3	boreholes - contains a borehole's information	24
4	samples - contains the soil samples taken from a borehole	25
5	project-engineers - contains engineers with their projects	25
6	uscs - contains soil classifications	25

I. Introduction

A. Background of the Study

Geotechnical investigation is a process performed by engineers to collect information and analyze subsurface properties around a site in order to design and construct a structure's foundation [1]. Information such as the distribution, type and physical properties of subsurface materials and geological formations are obtained through carrying out testing methods. These can be accomplished in different ways including the use of test pits, trenching, boring, and in-situ tests [2].

Geotechnical site investigation includes testing of boreholes, which is a drilled hole into the subsurface in order to extract or investigate a material at a certain depth [3]. The material, such as soil (for engineering purposes), is then observed or tested. The results are recorded in different types of logs and reports such as borehole logs, liquid and plastic limits test report and several others. The examination of subsurface materials from boreholes and corresponding reports play an important role in the design and analyses of projects [4].

After subsoil conditions and properties in a project site are analyzed exhaustively through these visualization logs and reports, proper foundation type is then selected by the engineer. It is necessary for foundations to be capable of carrying imposed loading without causing structural damage or failure. The pressure that a foundation can withstand without vertical or horizontal soil displacement is called the soil's bearing capacity [5, 6].

L.A. Global Estate Development & Construction Corp. (LA Global EDCC) is a construction and development firm in the design and build environment [7]. They specialize in micropiling works, foundation engineering, structural engineering design and construction. They are also beginning to undertake geotechnical investigations as well.

Presently, for their geotechnical investigation reports, the company uses AutoCAD to manually create graphs and report layouts. They also use Microsoft Excel

for inputting data matrices in order to compute for bearing capacities to be used in their foundation recommendations. Microsoft Word is also used for compiling all the reports. The compiled reports which includes the logs, analyses, and recommendations, are then submitted to the client, which in turn will be submitted to the building officials.

This study is under computational science, a field in computer science which uses advanced computing and data analysis for solving scientific and engineering problems.

B. Statement of the Problem

The plotting of graphs individually and the use of separate programs tend to be time-consuming and prone to graphing inaccuracies and formatting errors. For instance, some graphs are not plotted in a semi-logarithmic scale although they should be. Certain reports require the users to input the same data repeatedly because the company's current applications are not specifically designed for creating geotechnical investigation reports. This sometimes causes inconsistent inputs, and consequently, incorrect outputs. Moreover, users tend to forget certain labels and inputs that are necessary for the analyzation of the reports. As might be expected, the programs do not notify them of their empty or inconsistent fields.

The company does not make use of existing logging applications because they are costly. Moreover, they only generate logs and lacks the capability to collaboratively create a document which the company needs. They also do not include all the graphs and logs required by the company, as well as computing soil bearing capacities.

C. Objectives of the Study

The purpose of the study is to create a document generator/information system/-calculating tool for LA Global EDCC. It will include the ability to create borehole logs, liquid and plastic limits test report, particle size analysis report, and borehole

soil profiles. It will also include the option to choose recommendations provided by the company, as well as other templates and features necessary for the creation of the whole report.

The specific objectives of this study are as follows:

1. Allows the users (engineers) to
 - (a) Register an account
 - (b) Sign in to/sign out of their account
 - (c) Edit account information
 - (d) Add/view/edit/delete borehole information to a project
 - (e) Add/view/edit/delete sample information to a borehole
 - (f) Add a report to a project
 - (g) Create the following subreports:
 - i. Borehole log
 - ii. Soil profile of boreholes
 - iii. Liquid and plastic limits test report
 - iv. Particle size analysis
 - (h) Add other report sections (as deemed necessary by the engineer)
 - (i) Tag project with complete report as 'for admin validation'
 - (j) View reports (compiled subreports and other sections)
 - (k) Download reports in pdf format
 - (l) View audit trail
2. Allows the administrator to
 - (a) Accept/reject pending user registration
 - (b) Disable user account

- (c) Sign in to/sign out of his/her account
- (d) Edit account information
- (e) Add/remove project
- (f) Edit project information
- (g) Add/remove engineers to/from projects
- (h) Add a report to a project
- (i) Add/view/edit/delete borehole information to a project
- (j) Add/view/edit/delete sample information to a borehole
- (k) Create the following subreports:
 - i. Borehole log
 - ii. Soil profile of boreholes
 - iii. Liquid and plastic limits test report
 - iv. Particle size analysis
- (l) Choose from and add the following foundation types to recommend if report is tagged:
 - i. Shallow foundation
 - ii. Deep foundation
 - iii. Drilled shaft foundation
- (m) Add other report sections (as deemed necessary by the engineer)
- (n) Tag project with finished/complete reports as 'for admin validation'
- (o) Receive notifications regarding:
 - i. Pending users
 - ii. Tagged projects with complete reports
- (p) Validate tagged projects (archive)
- (q) View reports (compiled subreports and other sections)

- (r) Download reports in pdf format
- (s) Send validated (archived) reports via email
- (t) View audit trail

D. Significance of the Project

With a documentation system designed for geotechnical investigation reports, it will no longer be necessary for the company to use multiple applications to individually plot graphs, interpret them, and compile them into one. Thus, the company will be able to save generous amount of time and make room for productivity in other areas of work. The users will also be able to collaborate and track their activities. Most importantly, investigation reports of the company, which are an important aid to designing and constructing structures, will be reduced of incorrect and inaccurate computations and outputs.

E. Scope and Limitations

1. The program will not include the use of maps and coordinates for visualizing borehole locations.
2. The program will not create other logs and reports aside from the ones specified by the company (i.e. aside from borehole logs, liquid and plastic limits test report, particle size analysis report, and soil profile of boreholes).

F. Assumptions

1. Appropriate permits have been secured before adding data to the system.
2. Laboratory results and analysis are just entered in the system; they are not created by the system.
3. The lead engineer is also the system administrator.

4. The users of the program have knowledge in geotechnical investigation or are qualified engineers to avoid misuse.

II. Review of Related Literature

A. Boreholes and Borehole Logging

A borehole is a drilled or dug hole into the subsurface for extracting or investigating a material at a certain depth. Boreholes are commonly cylindrical, deep and narrow. The earliest deep boreholes were bored by the Chinese and had a maximum depth of around 200 m. Some of the earliest borehole records date back to the 18th century [3].

One of the main purposes for drilling boreholes is site investigation and construction [3]. It plays an important role in design and analyses of projects [4]. According to the New York City Building Code under 1802.4 [8], soil characteristics shall be based on tests and observations disclosed by borings, test pits, or other subsurface exploration methods.

Boreholes are also used as water supply. For instance, many people depend on boreholes as their water supply in Nigeria [9]. Water distribution facilities fail to deliver or meet the water requirements of the community due to certain factors. The scarcity of water made it necessary to find alternative sources of water [10, 11]. In a desperate attempt to obtain water, landlords have drilled boreholes in order to tap the groundwater [10]. Although groundwater is generally a "safe source" of drinking water because of its low microbial load, groundwater resources are susceptible to pollution [12].

In the oil and gas industry, boreholes are specifically used for the analysis of the properties of geological formations such as rocks "to find hydrocarbon zones in the geological formations intersected by a borehole" [13].

Borehole logging, also known as well logging [13], is the process of making a detailed record of geological formation [14]. The detailed record is called a borehole log, which is a basic visualization tool in analyzing subsurface characteristics [4] and formation samples made by instruments that are lowered into a borehole [14].

Electric well logging is considered invented by Conrad and Marcel Schlum-

berger. The first well log was created on September 15, 1927, when one of Schlumberger's men working for him lowered an electric tool in a well in Pechelbronn, Alsace, France [14].

B. Site Investigations using Boreholes

Collapse of a structure is a major concern to the public as it endangers lives and properties. For instance, such incidences have been constant in Nigeria the past few years [15]. According to Ebehikhalu & Dawam [16], the incidence of building collapses and failures in Nigeria is beginning to reach an epidemic proportion.

The following are some related studies (mostly in Nigeria) and investigations on structures and sites which made use of borehole testing and logging in analyzing subsurface and other geotechnical formation.

An investigation of a proposed four-storey building site in Lagos, Nigeria was conducted by Festus et al. [15]. Geotechnical testing (such as drilling, sampling, test pitting, and in-situ laboratory testing) was supplemented with geophysical methods in order to establish lateral and vertical variations. Eleven boreholes were drilled to a maximum depth of 30 m and samples were collected for examination and analysis. Several other tests were conducted to investigate the specimens. The results indicated soft to stiff clays as well as presence of sands and silts in the clayey deposit. Along with other results including internal friction, water content, bulk density, and allowable bearing capacity, it was interpreted that the site lies underneath an extensive zone of ductile and low strength founding materials having compressibility of medium to high. Settlement value also exceeded the suitable tolerable limit for founding of a four-storey building. Therefore, according to the researchers, the proposed site is not suitable for the construction of a four-storey building and should be avoided.

Another study [17] aimed to establish subsoil profile and characteristics and recommend appropriate foundation design and construction in a site in Nigeria. Three boreholes with maximum depth of 30 m were drilled to obtain data on soil

and water level monitoring properties. Particle size distributions of the samples were determined using sieve analysis. The results indicated low to medium plasticity silty clay. Under the clay was a layer of loose to medium dense and dense sand. The researchers suggested that proposed structures in the area be supported by raft foundation within the upper clay layer.

Yusuf et al. [18] studied productivity and quality analysis of borehole water in selected locations in Nigeria. Pumping test was carried out to determine the discharge from the boreholes sampled. Seven boreholes were functional out of ten sampled in Offa. Four out of five boreholes sampled in Erin-Ile were working, while only 13 out of 21 boreholes sampled in Ilesa were working. Three boreholes were sampled in Osogbo with only one functional. The results revealed that boreholes can relieve water problem in the areas during dry season in the country as it gives considerable quantity of water. Boreholes are also usually free from E-Coli bacteria and other pathogens. Moreover, chemical contents in the samples were determined to cause no disease to humans. Therefore, groundwater of boreholes in the mentioned locations is suitable for consumption.

A study for road construction purposes, conducted by Youdeowei and Nwankwoala [19], aimed to investigate subsurface by drilling boreholes within and along the Nun River Channel covering three communities. Eight boreholes with a maximum depth of 14 m from the river bed were drilled in the three locations. Results showed very fine grained and high plasticity clay soil with soft to stiff consistency in the borehole samples. The unit varies in depth and beneath were medium grained sands.

Another study [20] aimed to determine subsoil profiles and geotechnical characteristics in parts of Yenagoa, Bayelsa State, Nigeria. Four boreholes were drilled in the area and samples were representative samples were collected. The results revealed that the area is predominantly underlain by medium to firm silty clay, with some silty sand in different depths and boreholes. Water level was also recorded as well as moisture content, liquid limit, plastic limit, plasticity index, liquidity

index, and hydraulic conductivity. This study had analyzed the area to be used for future foundation designs in order to minimize post construction problems.

The studies differ in their purposes for applying borehole logging methods. The first two applied borehole logging methods for building constructions purposes. Other purposes are for groundwater analysis and road construction. It is also used in the oil and gas industry [13]. This shows that borehole testing and logging is thus a very significant part of the analysis of geotechnical subsurfaces and formations for engineering, hydrogeology, and such.

C. Related Systems

1. gINT Logs

gINT Logs is a software for creating and managing borehole logs and geophysical logs. gINT provides two more applications: gINT Professional and gINT Professional Plus. The former also provides the capabilities of gINT Logs along with additional features. With gINT Professional, subsurface data can be managed using custom reports which includes graphics, photos, charts, maps, and more. It allows the user to create fence diagrams, graphs, tables, and such. gINT Professional Plus, on the other hand, provides SQL Server support for a more powerful reporting and centralized data management [21].

2. Strater[®] 4

Strater, like gINT Logs, is also basically a visualization software for borehole logs, maps, and cross sections. Strater offers a modern, intuitive and easy-to-use user interface as well as design and layout flexibility. It also offers 14 customizable log types for subsurface data representations [22].

3. LogPlot

LogPlot is another borehole log software. It features LogPlot Log Designer which offers user-friendly and flexible borehole log designs. It also offers LogPlot Data

Designer which has a tabbed spreadsheet window for easier inputting of data. Users can also print and export logs with LogPlot [23]. A standard single license costs \$699.00, while a 3-pack and 5-pack single license costs \$1,887.30 and \$2,621.25, respectively [24].

4. LOGDRAFT

LOGDRAFT's current release is a stand-alone borehole log program. It offers user-designed boring log reports, data entry screens, and stratigraphy symbols. Along with the package comes sample borehole logs, monitoring well installation log, rock core log, test pit log and cross section report formats [25].

III. Theoretical Framework

A. Geotechnical Investigation

Geotechnical investigation is the activity in which geological and environmental characteristics are analyzed and evaluated [26]. It is a requirement in order to aid the design, construction, and maintenance of most civil engineering structures [2]. Hence, it is the most important basic work for construction of structures [26]. Soil exploration or geotechnical testing are performed in different ways including the use of test pits, trenching, boring, and in-situ tests [2]. Investigation subjects include buildings such as houses, industrial workshops, schools, hospitals, and more [26].

Before the design of a project begins, investigations are accomplished in line with existing guides and codes regarding the quality of the proposed site. Investigations are carried out in order to prevent structural failures from occurring [15]. Accurate information and competent understanding of the geotechnical properties and behaviors of soils are very important [19] as they take part in solving engineering and environmental issues [27]. Understanding of geotechnical characteristics through subsurface investigation is also important for the design and construction of foundations for proposed structures [20]. Information gathered in the investigation is helpful in guaranteeing that the effects of the project on the environment are properly analyzed [19].

Incorrect geotechnical information or analysis can bring structure failures. Building failures cause loss of life as well as properties which cost several millions of dollars. Failures can cause disasters where public safety is seriously compromised. Nigeria and Lagos, for instance, are locations where rate and frequency of building failures are alarming [15]. Environmental problems will also arise if secure and environmental-friendly measures are not accomplished. Such environmental issues include soil and groundwater contamination [19].

According to the New York City Building Code [8], a written report of the

investigation shall be signed, sealed, and submitted by the engineer accountable, where required by the commissioner. The written report shall include information specified under section 1802.6. Among many are: a plot of the borehole locations, a record of the soil profile and soil sample descriptions, results of laboratory testing, recommendations for foundation type and design criteria, and pile foundation recommendations.

The mentioned information will be summarized by the program by generating the following subreports:

1. Borehole Log

A borehole log is a detailed record for analyzing subsoil characteristics made by instruments lowered into a borehole [4].

From the given borehole data, a borehole log can be divided into three parts: histogram head, histogram body, and histogram bottom. The histogram head contains basic information such as project name, borehole number, and other user-defined information. The histogram body, in a form of a column, includes stratum and sampling information. The column body provides the visual and graphical representation of the data. The histogram bottom mainly includes the page number of the borehole log [4].

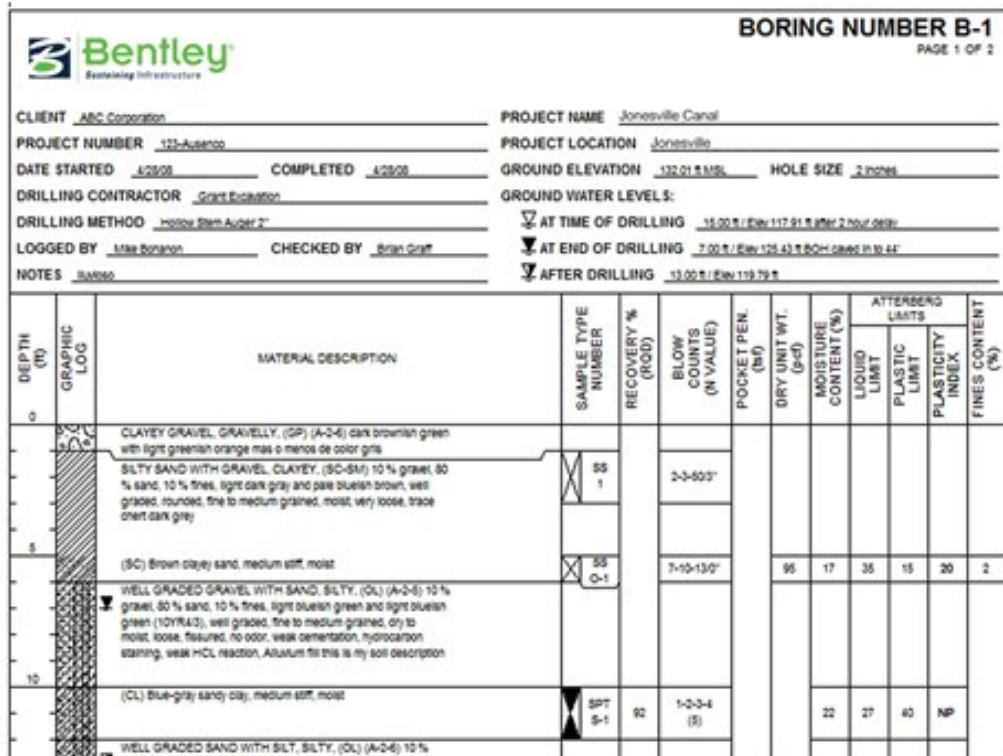


Figure 1: Example of borehole log [21]

2. Soil Profile of Borehole

Soil profile is one of the basis used by soil scientists and other professionals for understanding soil properties. The profile description should accurately correspond to what is observed in the site. The level of detail depends on the objective of the client, proposed land use, and observation method. However, most (if not all) soil profile descriptions include horizon, depth, soil properties of structure, color, structure, consistence, and thickness [28].

3. Liquid and Plastic Limits Test Report

Plasticity is the ability of a soil to undergo permanent deformation without crumbling. A soil may be in the liquid, plastic, semi-solid, and solid states, depending on its water content [29]. A soil behaves as a liquid when it has a sufficient water content. When it is gradually reduced, the state changes from liquid to plastic and semi-solid, and finally to solid state [29, 30].

Liquid limit and plastic limit, known as the Atterberg limits, provide necessary

data for identifying and classifying clays [30]. The upper limit of the range of water content over which the soil is in plastic state is defined as the liquid limit (w_L), while the lower limit is defined as the plastic limit (w_P). Their difference, the water content range, is the plasticity index (I_P) [29].

$$I_P = w_P - w_L \quad (1)$$

Test results are plotted on a semilogarithmic plot with water content as the ordinates and number of blows as the abscissa [31].

4. Grain Size Analysis

Particle size analysis, also called grain size analysis, determines the size range and distribution of a soil. After passing through a series of standard test sieves to determine the particle size of the soil sample, the size distribution is plotted. It is represented as a curve on a semilogarithmic plot, with the percentage by mass of particles smaller than the size ranges as the ordinates and the size ranges (see Figure 2) as the abscissa. Steeper curve means smaller particle size range and flatter curve means larger size range [29].

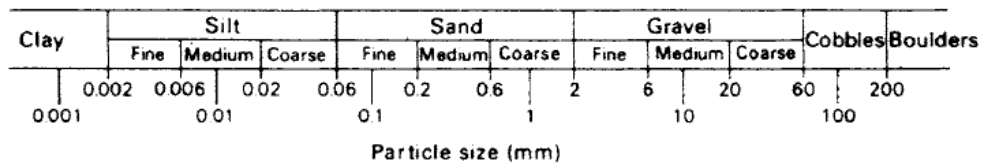


Figure 2: Particle size ranges [29]

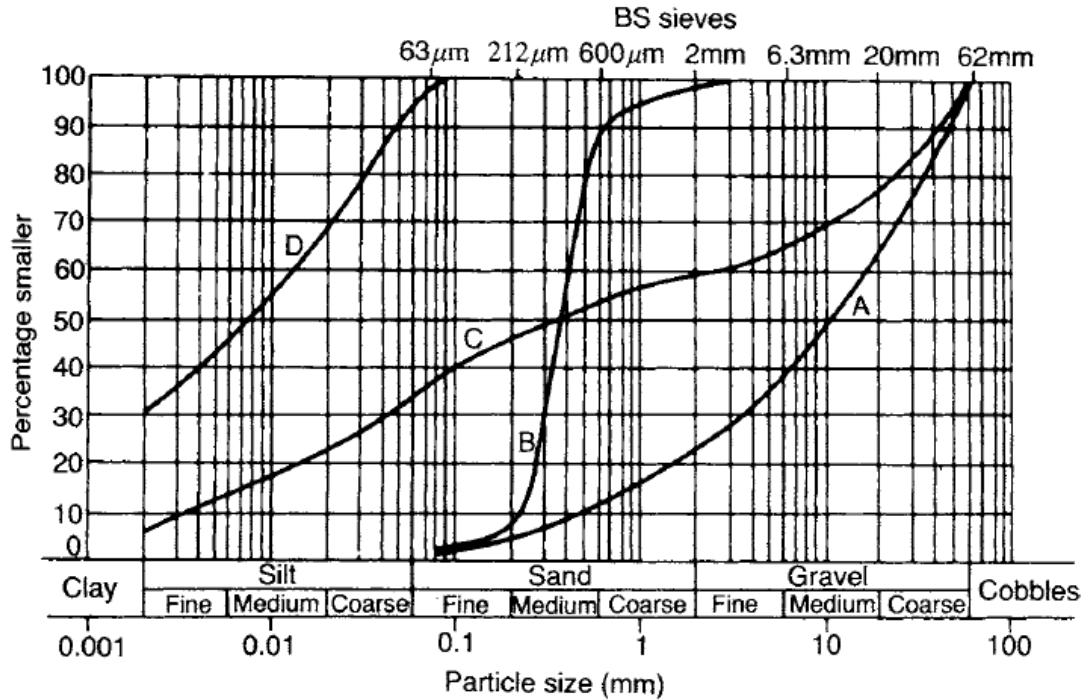


Figure 3: Example of particle size distribution curves [29]

B. Soil Bearing Capacity for Foundations

After subsurface conditions and properties in an investigation site are analyzed exhaustively, selection of proper foundation type can be made [5]. The foundation is the lowest part of a structure that connects its weight to the ground (underlying soil or rock) [32, 6]. It is necessary for foundations to be capable of carrying imposed loading without causing structural damage or failure [5].

The pressure that a foundation can withstand without vertical or horizontal soil displacement is called the soil's bearing capacity [5, 6]. Inadequate bearing capacity of the soil beneath the foundation causes abrupt and harmful failures [6]. Therefore, allowable bearing capacities play an important role to avoid the collapse of structures and to ensure their stability [6, 33].

1. Shallow Foundation

Bearing capacity and settlement are the two key properties necessary in the design of shallow foundations. House subdivisions, antenna towers, bridges, and power

plants are some of the structures which can be constructed on shallow foundations [34].

The classical bearing capacity equation for long footings [5], developed by Terzaghi & Peck in the 1940's, is as follows:

$$q_{ult} = cN_c + 0.5B\gamma_1N_\gamma + \gamma_2D_fN_q \quad (2)$$

where N_c, N_γ, N_q = soil bearing capacity factors whose values depend on the angle of internal friction

c = cohesion of the soil below footing level

γ_1 = effective unit weight of soil below footing level

γ_2 = effective unit weight of soil above footing level

B = footing width

D_f = depth of footing below lowest adjacent soil surface

q_{ult} = ultimate gross bearing capacity or soil-bearing pressure

2. Deep Foundation

Bearing capacity and settlement are the two key properties necessary in the design of shallow foundations. House subdivisions, antenna towers, bridges, and power plants are some of the structures which can be constructed on shallow foundations [34].

The classical bearing capacity equation for long footings [5], developed by Terzaghi & Peck in the 1940's, is as follows:

$$q_{ult} = cN_c + 0.5B\gamma_1N_\gamma + \gamma_2D_fN_q \quad (3)$$

where N_c, N_γ, N_q = soil bearing capacity factors whose values depend
on the angle of internal friction

c = cohesion of the soil below footing level

γ_1 = effective unit weight of soil below footing level

γ_2 = effective unit weight of soil above footing level

B = footing width

D_f = depth of footing below lowest adjacent soil surface

q_{ult} = ultimate gross bearing capacity or soil-bearing pressure

3. Drilled Shaft Foundation

Bearing capacity and settlement are the two key properties necessary in the design of shallow foundations. House subdivisions, antenna towers, bridges, and power plants are some of the structures which can be constructed on shallow foundations [34].

The classical bearing capacity equation for long footings [5], developed by Terzaghi & Peck in the 1940's, is as follows:

$$q_{ult} = cN_c + 0.5B\gamma_1N_\gamma + \gamma_2D_fN_q \quad (4)$$

where N_c, N_γ, N_q = soil bearing capacity factors whose values depend
on the angle of internal friction

c = cohesion of the soil below footing level

γ_1 = effective unit weight of soil below footing level

γ_2 = effective unit weight of soil above footing level

B = footing width

D_f = depth of footing below lowest adjacent soil surface

q_{ult} = ultimate gross bearing capacity or soil-bearing pressure

C. Laravel

Laravel [35] is an open-source PHP web application framework created by Taylor Otwell in 2011. Laravel, like CodeIgniter, uses the model-view-controller (MVC) pattern. It offers a powerful template engine, a simple ORM which works with major database servers, and several others [36]. Aside from having a simple and elegant code, Laravel's documentation is nearly perfect. [37].

According to Google Trends [38] and sitepoint [37], Laravel is currently the best and most picked among other PHP frameworks such as CodeIgniter, Zend, Yii, and Symfony.

IV. Design and Implementation

A. Use Case Diagram

The system will have two users: engineers and system administrator.

The engineers will be able to register an account, sign in and out of their accounts, manage borehole information, manage sample information, create sub-reports and report sections, tag projects, and view and download reports.

The system administrator will be able to do everything an engineer can. He/she will also be responsible for managing user accounts, adding and removing engineers from a project, adding and deleting projects, archiving projects, calculate bearing capacities, receive notifications, and send reports via email.

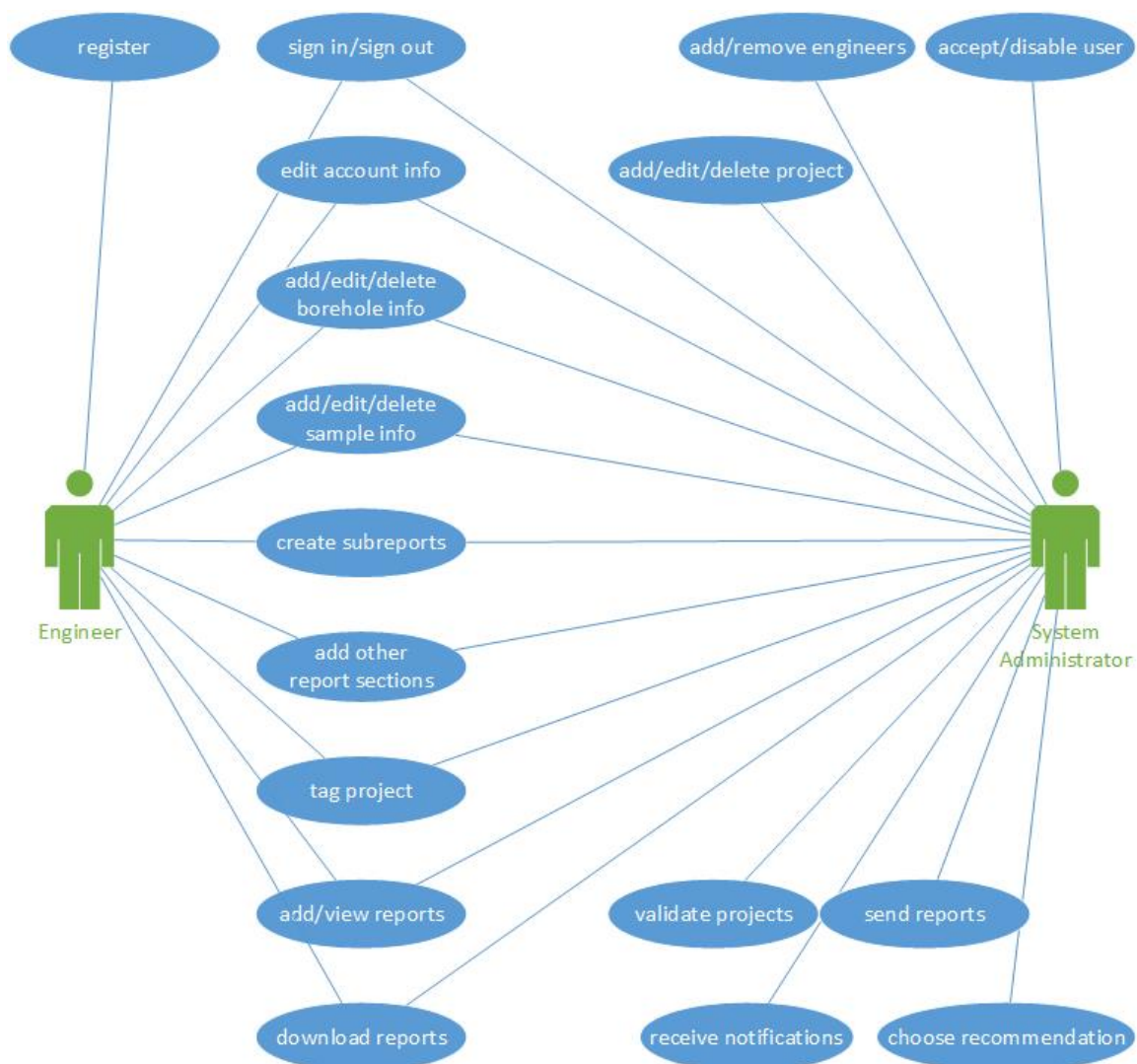


Figure 4: Use Case Diagram

B. Context Diagram

The system has two types of users, namely the system administrator (lead engineer) and regular users (engineers). Since the system administrator is also the lead engineer, he/she is able to access all resources. Only the regular users have limited access to functionalities and resources.

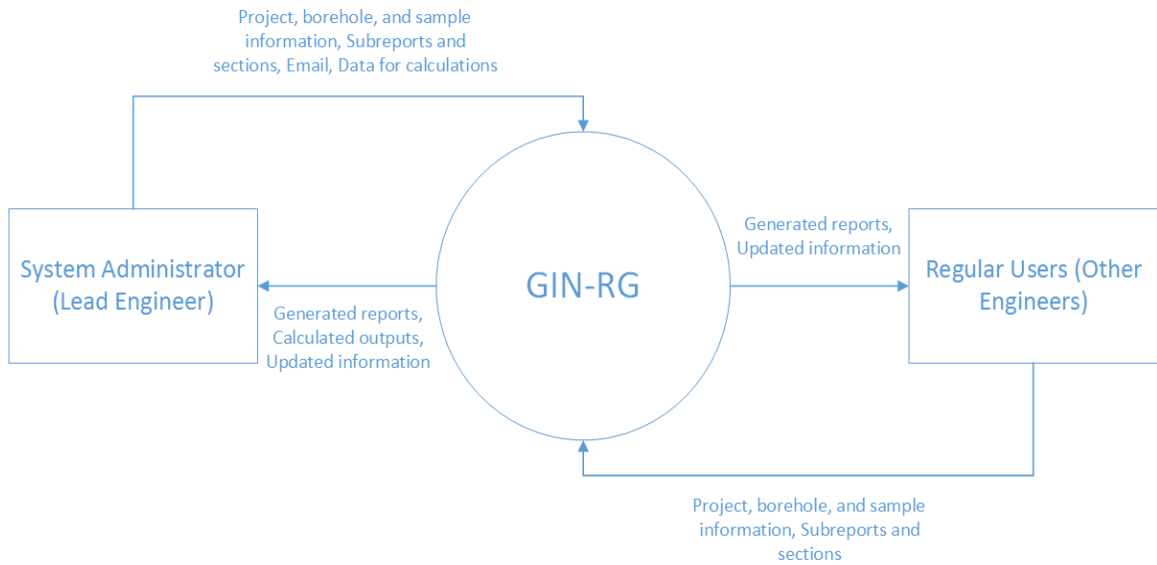


Figure 5: Context Diagram

C. Entity Relationship Diagram

A database is necessary to store resources and allow access to them. Figure 6 displays the Entity Relationship Diagram for the system.

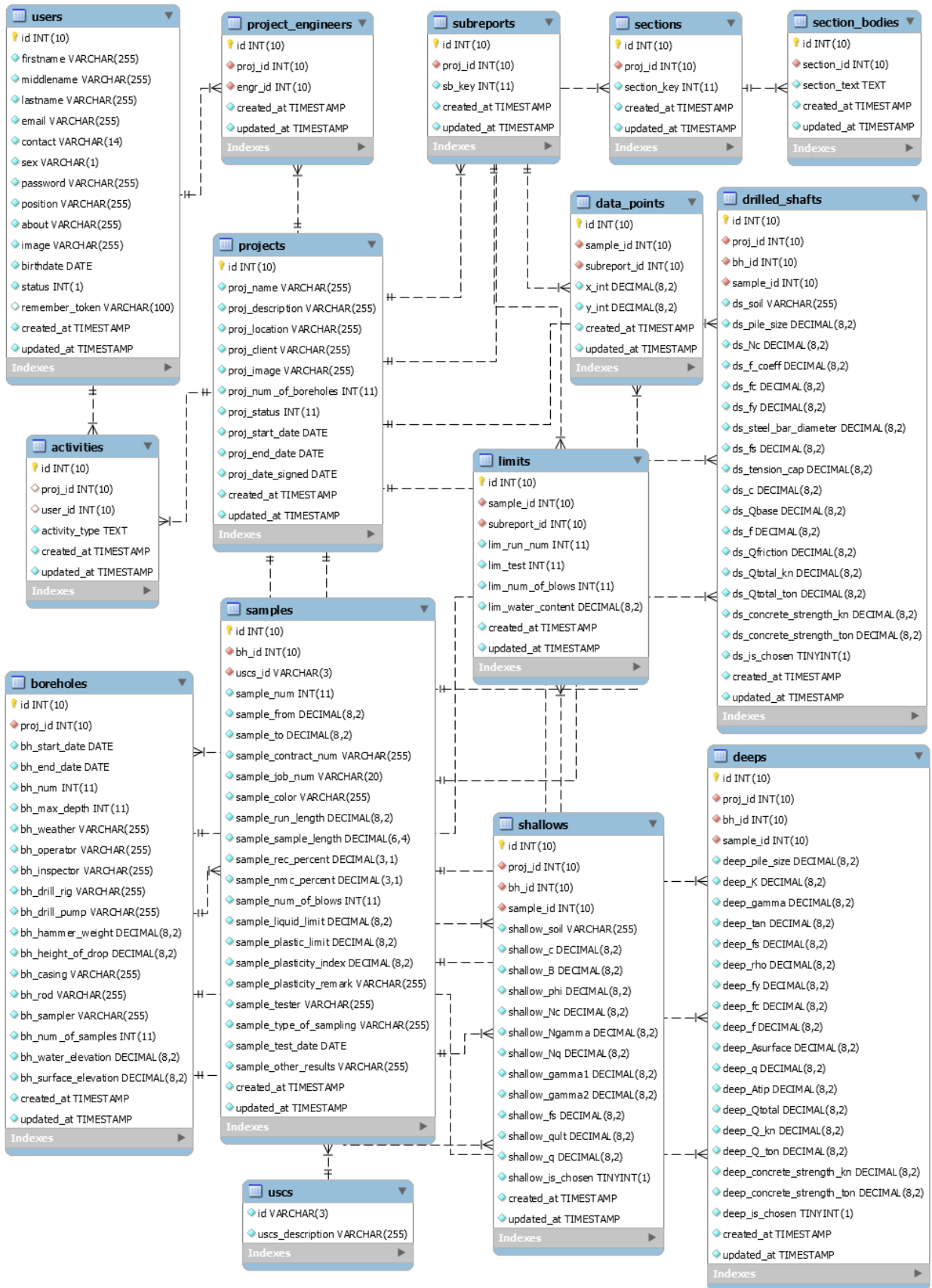


Figure 6: Entity Relationship Diagram

D. Data Dictionary

Table 1: **users** - contains both the system administrator (lead engineer) and the engineers

Data field	Type	Description
id	VARCHAR	User's unique identifier; Used by the user to log in
user_firstname	VARCHAR	User's first name
user_middlename	VARCHAR	User's middle name
user_lastname	VARCHAR	User's last name
user_email	VARCHAR	User's email address
user_password	VARCHAR	User's password
user_contact	VARCHAR	User's contact number
user_sex	VARCHAR	User's sex
user_status	VARCHAR	User's current state - 0 if pending, 1 if active, 2 if admin, and 3 if disabled
user_position	VARCHAR	User's position in the company
user_image	VARCHAR	User's profile image
user_about	VARCHAR	User's description

Table 2: **projects** - contains the company's site investigation projects

Data field	Type	Description
id	INT	Primary Key; Project's unique identifier
proj_name	VARCHAR	Project name
proj_location	VARCHAR	Project location
proj_client	VARCHAR	Project's client
proj_description	VARCHAR	Project's description
proj_date_signed	DATE	Date of reports approval
proj_num_of_boreholes	INT	Number of boreholes drilled for the project
proj_start_date	DATE	Project's start date
proj_end_date	DATE	Project's end date
proj_soil_date_signed	DATE	Project's sign date

Table 3: **boreholes** - contains a borehole's information

Data field	Type	Description
id	INT	Primary Key; Borehole's unique identifier
proj_id (FK)	INT	Foreign Key; Project in which borehole is drilled
bh_start_date	DATE	Boring start date
bh_end_date	DATE	Boring end date
bh_max_depth	INT	Borehole's maximum depth
bh_weather	VARCHAR	Weather
bh_operator	VARCHAR	Borehole perator
bh_drill_rig	VARCHAR	Drill rig
bh_drill_pump	VARCHAR	Drill pump
bh_hammer_weight	DECIMAL	Hammer weight
bh_height_of_drop	DECIMAL	Height of drop
bh_casing	VARCHAR	Casing
bh_rod	VARCHAR	Rod
bh_sampler	VARCHAR	Sampler
bh_num_of_samples	INT	Number of samples taken
bh_type_of_sampling	VARCHAR	Type of sampling method
bh_water_elevation	DECIMAL	Water elevation
bh_surface_elevation	DECIMAL	Surface elevation

Table 4: **samples** - contains the soil samples taken from a borehole

Data field	Type	Description
id	INT	Sample's unique identifier
bh_id (FK)	INT	Borehole from where the sample was taken
uscs_id (FK)	VARCHAR	Sample's soil classification
sample_from	DECIMAL	Depth of sample (min.)
sample_to	DECIMAL	Depth of sample (max.)
sample_color	VARCHAR	Color's sample
sample_run_length	DECIMAL	Run length
sample_sample_length	DECIMAL	Sample length
sample_rec_percent	DECIMAL	Percentage recovery
sample_num_of_blows	INT	Number of blows performed in the SPT
sample_liquid_limit	DECIMAL	Sample's liquid limit
sample_plastic_limit	DECIMAL	Sample's plastic limit
sample_plasticity_index	DECIMAL	Sample's plasticity index
sample_plasticity_remark	VARCHAR	Remarks regarding plasticity
sample_tester	VARCHAR	Liquid and plastic limits tester
sample_start_date	DATE	Test start date
sample_end_date	DATE	End start date
sample_other_results	VARCHAR	Remarks or other results

Table 5: **project-engineers** - contains engineers with their projects

Data field	Type	Description
user_id	VARCHAR	Primary Key; Used by the user to log in.
proj_id	INT	Primary Key; Project's primary key

Table 6: **uscs** - contains soil classifications

Data field	Type	Description
id	VARCHAR	Soil classification
uscs_description	VARCHAR	Soil classification description

E. System Architecture

The program is coded in PHP using the Laravel PHP framework. It uses a MySQL database and is deployed on an Apache server. The program is be compatible for most browsers however, Mozilla Firefox or Google Chrome is recommended.

F. Technical Architecture

The following are the minimum requirements for the server machine:

- PHP v5.6
- MySQL v5
- Apache v4
- 500 MB RAM

The following are the minimum requirements for the client machine:

- Java-script enabled browser
- 1 GB RAM

Recommended browsers:

- Google Chrome v51
- Mozilla Firefox v45

V. Results

A. General View

Figure 7 shows the login page. This also serves as the home page of unauthorized users.

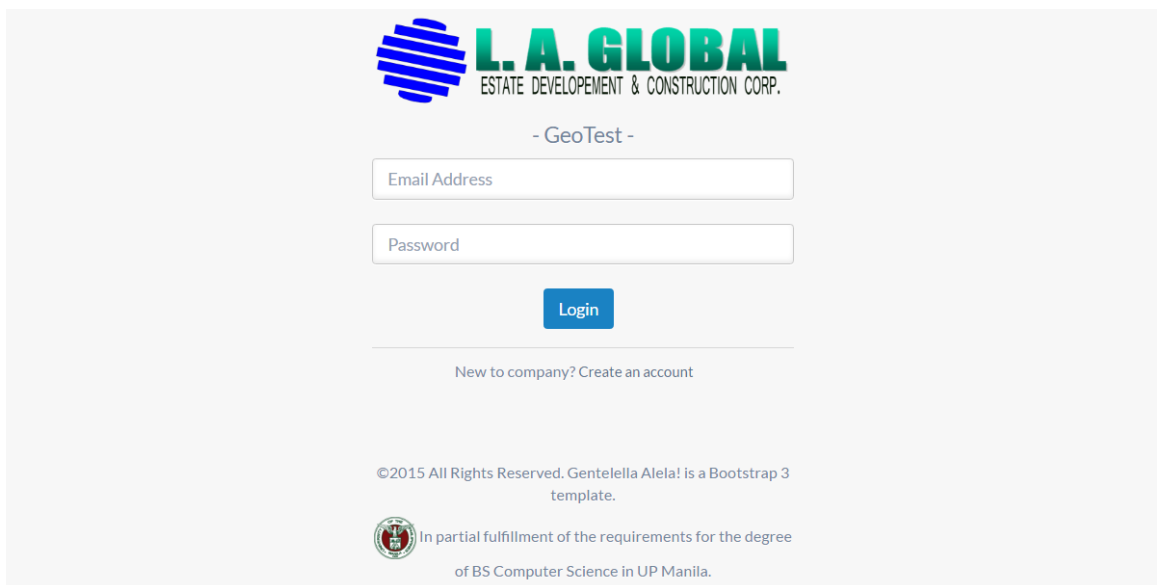
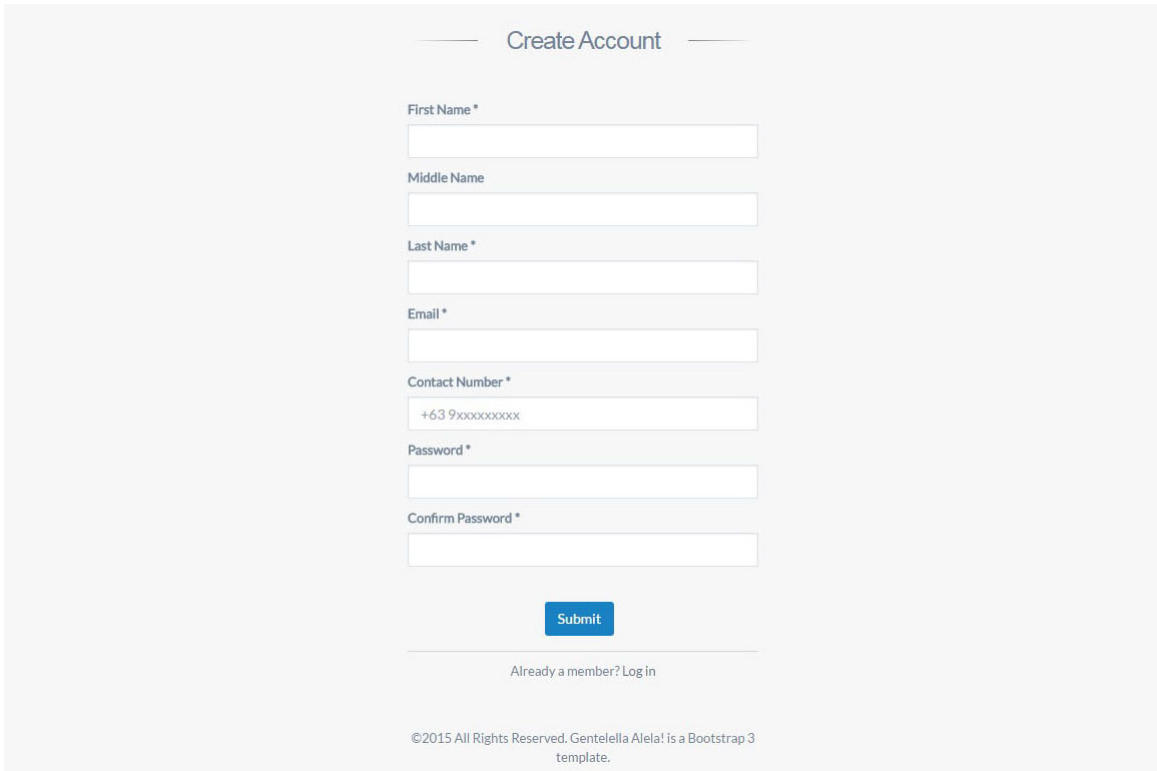


Figure 7: Login Page

Users with no existing account in the system can register an account by clicking 'Create an account' on the login page. The registration page is then shown after clicking the link (see Figure 8).



Create Account

First Name *

Middle Name

Last Name *

Email *

Contact Number *

+63 9xxxxxxxx

Password *

Confirm Password *

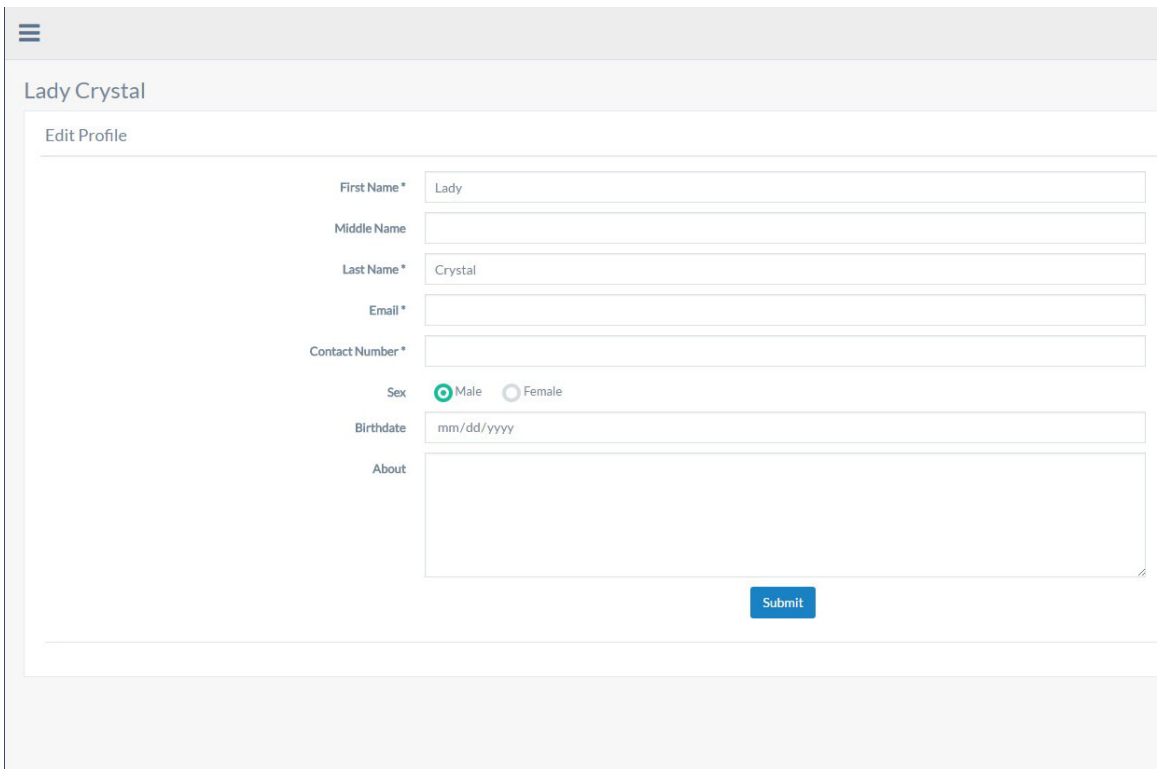
[Submit](#)

Already a member? [Log in](#)

©2015 All Rights Reserved. Gentelella Alela! is a Bootstrap 3 template.

Figure 8: Registration Page

The following figures show the functionalities common to both administrator and regular users.



☰

Lady Crystal

Edit Profile

First Name * Lady

Middle Name

Last Name * Crystal

Email *

Contact Number *

Sex Male Female

Birthdate mm/dd/yyyy

About

[Submit](#)

Figure 9: Edit User Profile

Most of the functionalities that users need are in the project page (see Figures 10 and 11). Here, users can edit the project, tag it (if it is active), add borehole information, manage subreports and sections, and view the generated (compiled) report.

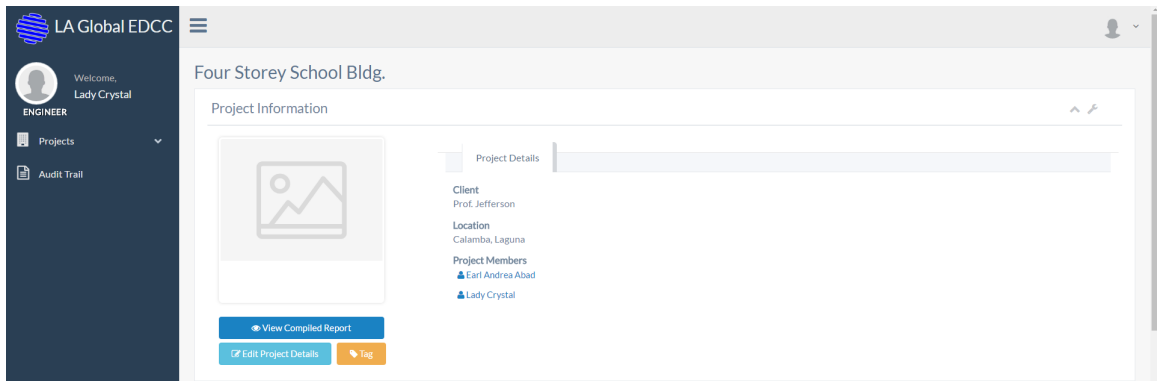


Figure 10: View Project Information

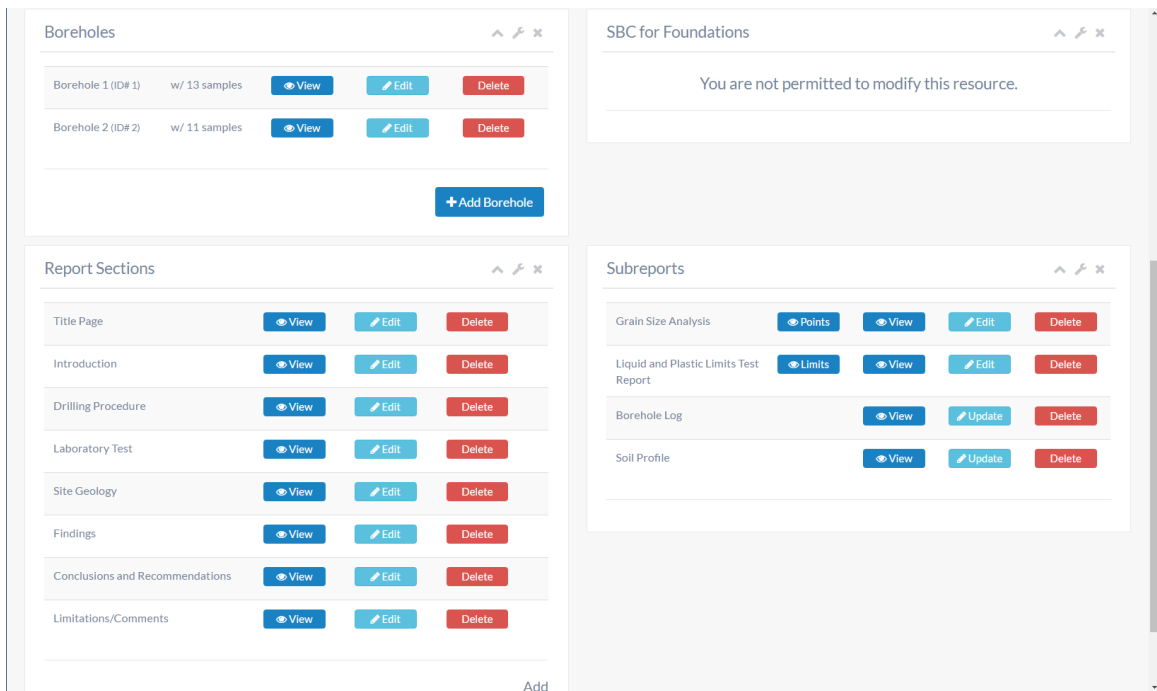


Figure 11: View Project Information

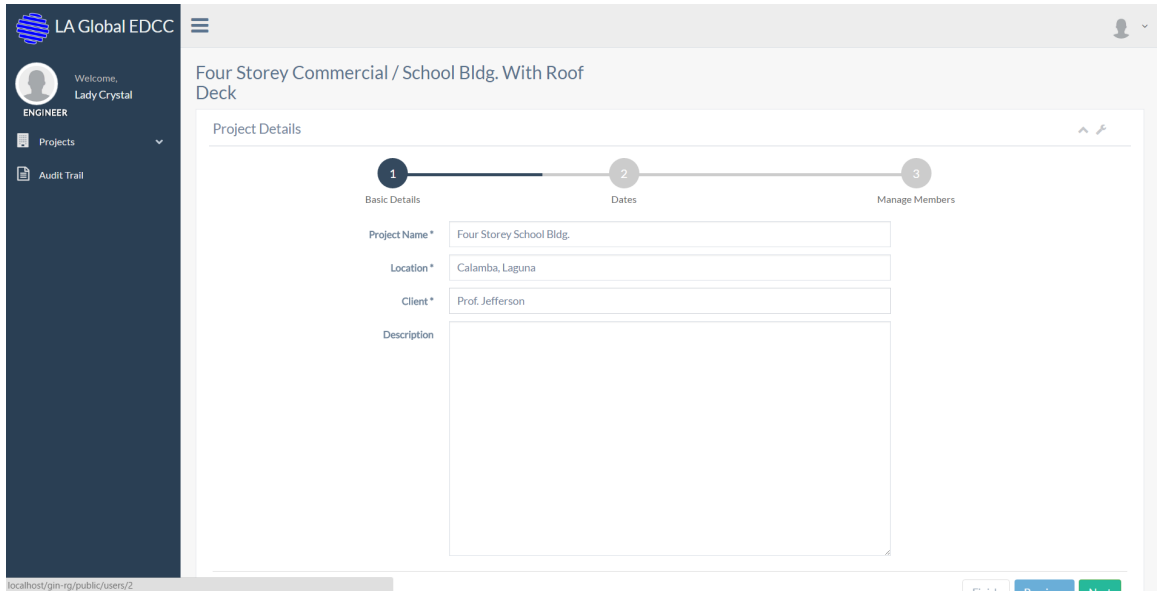


Figure 12: Edit Project Information

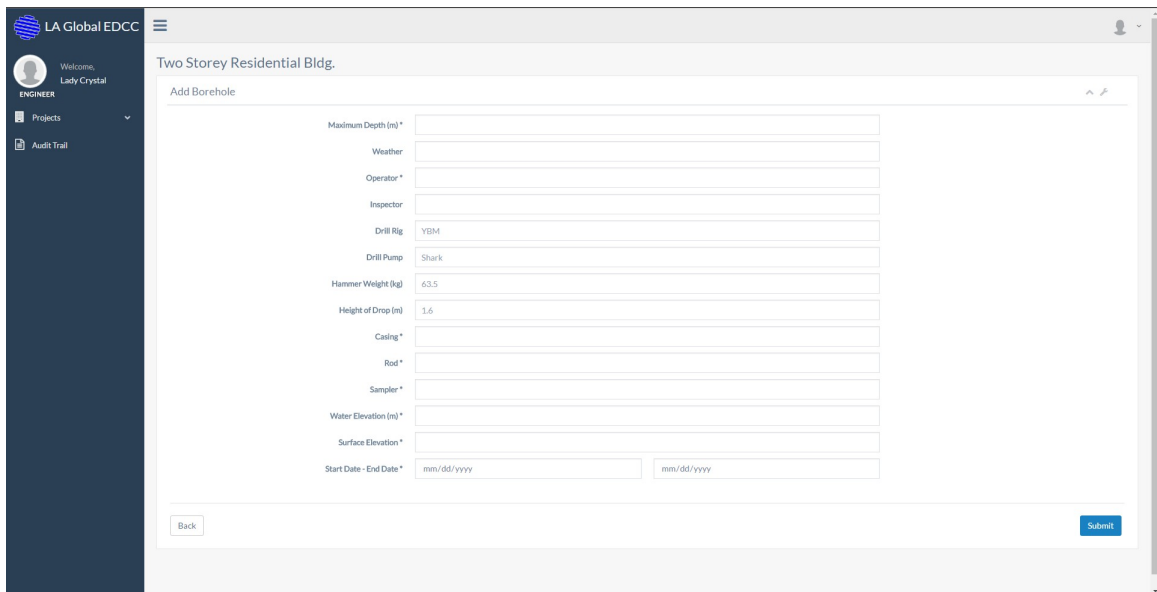


Figure 13: Add Borehole Information

LA Global EDCC

Welcome, Lady Crystal
ENGINEER

Projects
Audit Trail

Two Storey Residential Bldg.

Add Sample

Soil Classification * Choose soil classification...

Depth (m) From: 5.6 From: 5.7

Contract No.

Job No.

Sample Color *

Length of Run (m) 0.30

Length of Sample (m) *

Rec. (N) *

No. of Blows (SPT N-Value) *

Type of Sampling

NMC (%)

Liquid Limit *

Plastic Limit *

Plasticity Index *

Plasticity Remark

Tester

Figure 14: Add Sample Information

LA Global EDCC

Welcome, Lady Crystal
ENGINEER

Projects
Audit Trail

Two Storey Residential Bldg.

Borehole 1 (ID#3)

Start Date - End Date	March 18, 2016 - March 23, 2016	Operator	RCDL
Weather	Fair	Inspector	
Maximum Borehole Depth	15 m.	Drill Rig	Shark
Hammer Weight	63.50 kg	Drill Pump	Shark
Height of Drop	0.76 m.	Casing	NW
Water Elevation	0.00 m.	Rod	AW
Surface Elevation	0.00 m.	Sampler	SPT

Delete

+ Add Sample (1 of 1 Borehole 1 (ID#3))

Samples

No.	Depth	Soil Classification	Color	Length of Sample	No. of Blows SPT N-Value	Liquid Limit	Plastic Limit	Plasticity Index	Options
1	0.00 m - 0.30 m.	C	Brown	0.0375 m.	2	83.76 %	56.74 %	27.02 %	View Edit Delete
2	0.30 m - 0.60 m.	CML	Brown	0.3000 m.	4	42.12 %	38.24 %	4.00 %	View Edit Delete
3	0.60 m - 0.90 m.	C	Brown	0.3000 m.	12	47.24 %	38.81 %	8.71 %	View Edit Delete
4	0.90 m - 1.20 m.	C	Brown	0.1500 m.	15	58.74 %	43.44 %	15.30 %	View Edit Delete
5	1.20 m - 1.50 m.	C	Brown	0.0750 m.	13	50.70 %	35.66 %	15.04 %	View Edit Delete

Figure 15: Manage Borehole and Its Samples

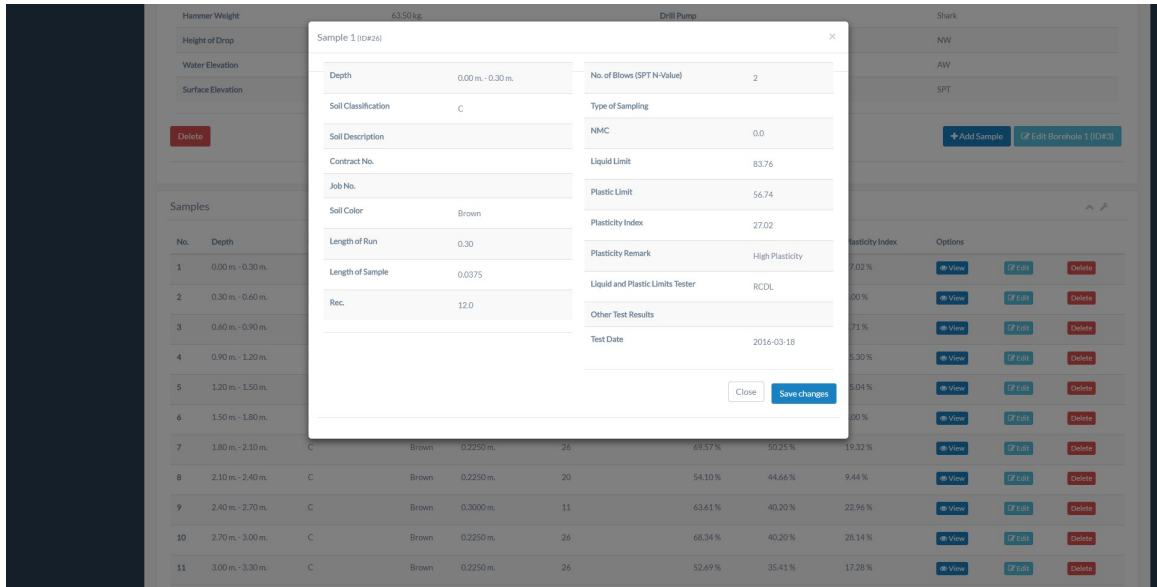


Figure 16: View Detailed Sample Information

After adding complete boreholes and samples information, users can manage (create, edit, view, and delete) subreports. The following figures show the views of the subreports.

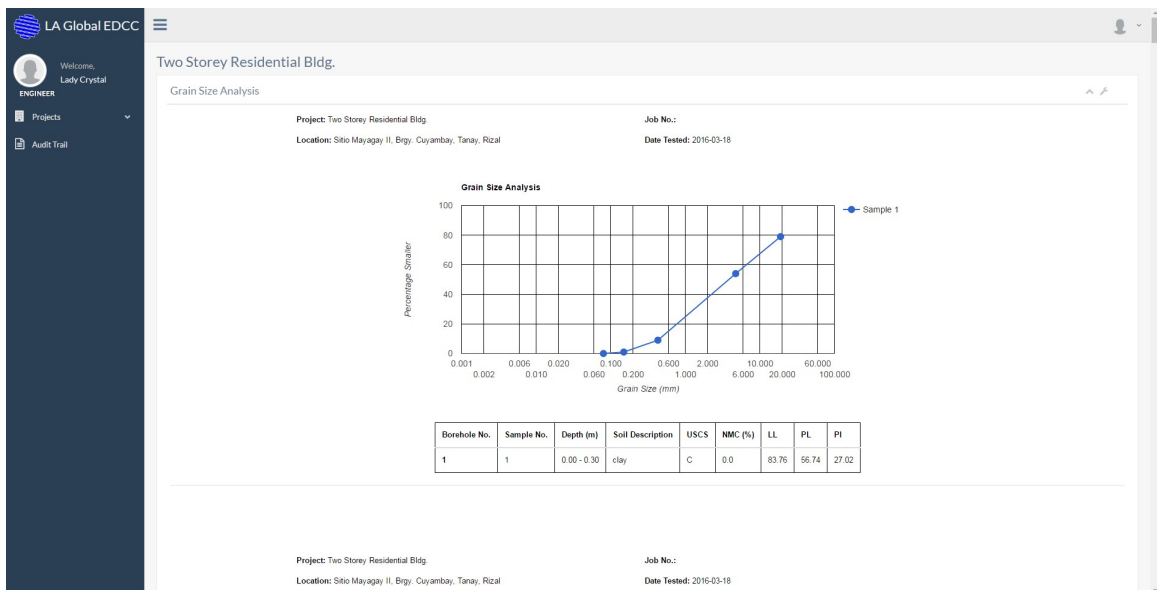


Figure 17: View Grain Size Analysis

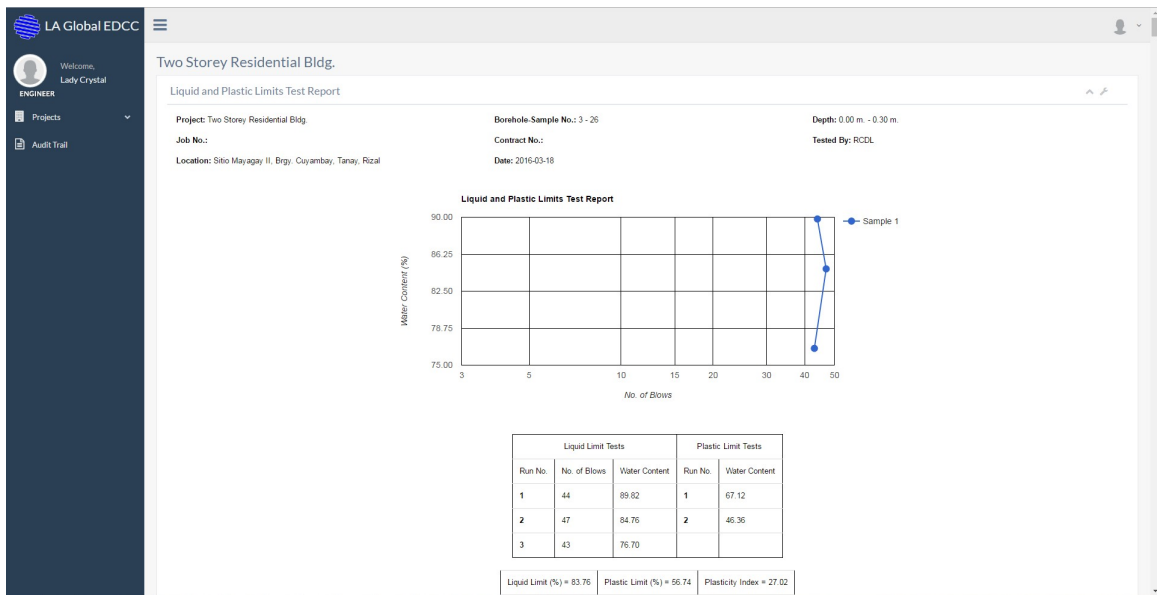


Figure 18: View Liquid and Plastic Limits Test Report

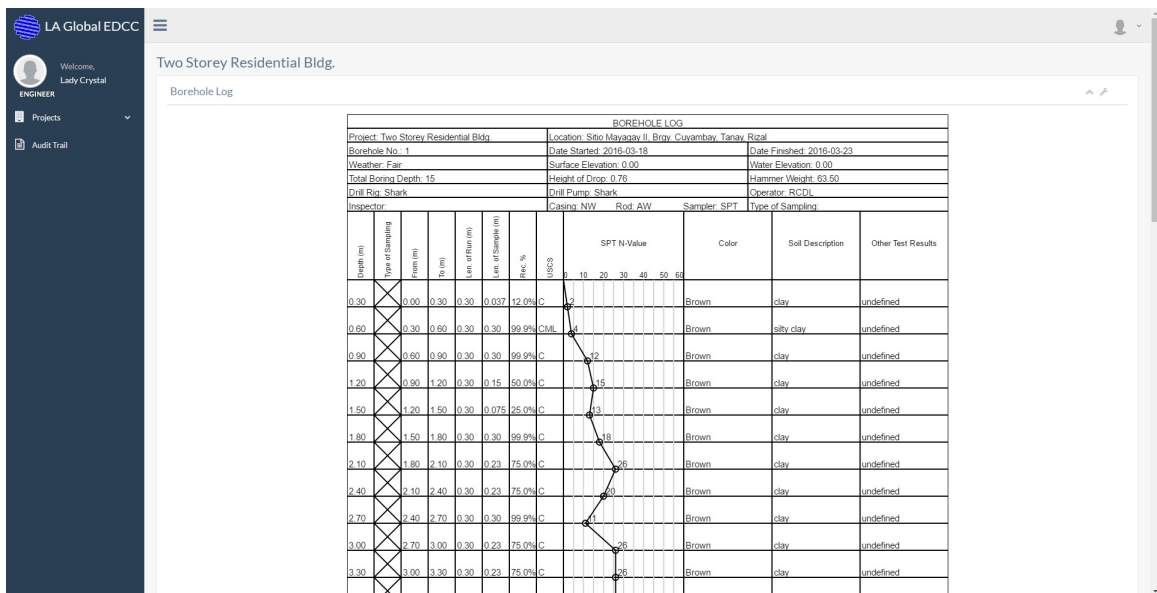


Figure 19: View Borehole Log

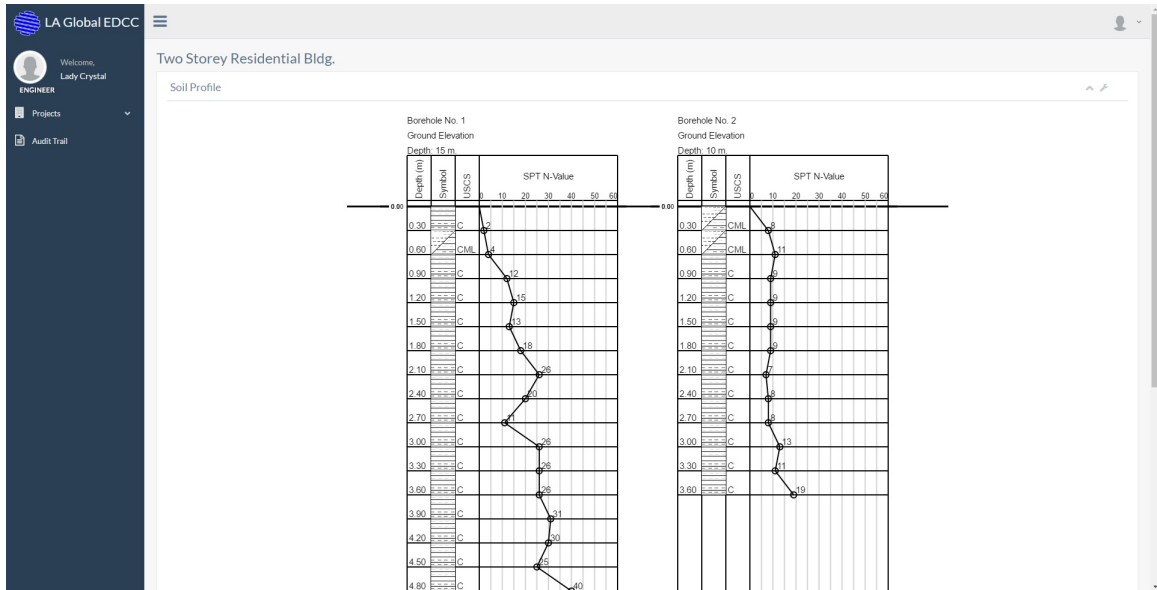


Figure 20: View Soil Profile

Users can also manage (create, edit, view, and delete) report sections as shown in the following figures.

Report Sections ^ ↗ ✕

Title Page	View	Edit	Delete
Introduction	View	Edit	Delete
Drilling Procedure	View	Edit	Delete
Laboratory Test	View	Edit	Delete
Site Geology	View	Edit	Delete
Findings	View	Edit	Delete
Conclusions and Recommendations	View	Edit	Delete
Limitations/Comments	View	Edit	Delete

Add

▼
Submit

Figure 21: View Existing Sections and Add New Section

Drilling Procedure ^ ↗

DRILLING PROCEDURE

Borehole #1 and #2 were drilled for a depth of 15m. Wash boring method was used in the drilling. Soil samples were taken at an interval of 0.3m from ground surface.

Standard Penetration Tests (SPT) for the two boreholes were conducted in order to obtain information on the soil bearing capacity in relation to the number of blows, as required to drive the 2-inch outside diameter split spoon sampler which is being hammered continuously by a 63.5kgs weight. The number of blows required to drive the sampler through the 12-inch penetration are recorded as the N-value of the soil layer being tested.

Coring is employed in hard soil formation to advance the drill rod and to obtain rock or hard soil samples on which laboratory tests are performed to establish their physical characteristics.

[Delete](#)
[Edit](#)

Figure 22: View Section

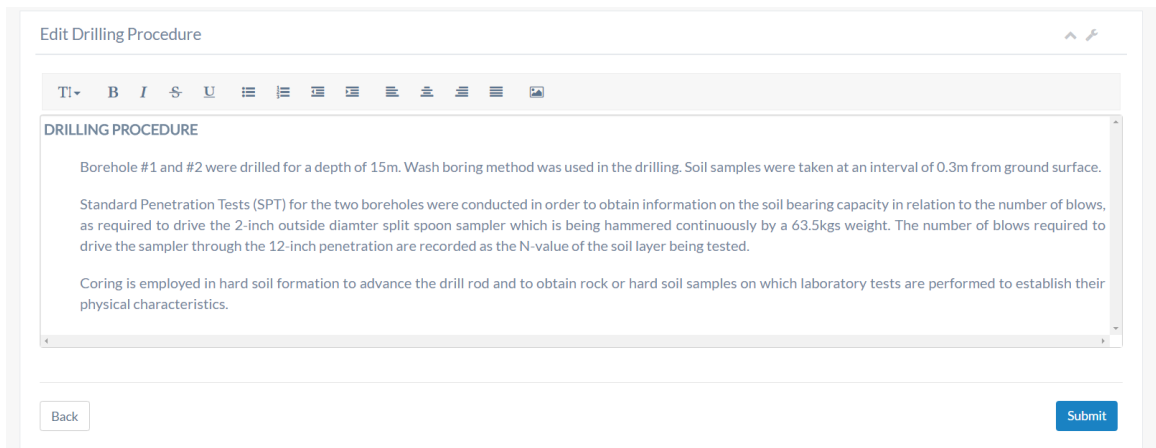


Figure 23: Edit Section

The report will be generated properly after subreports and sections have been successfully added. Users can view and download the report in the project's main page (see Figure 10).

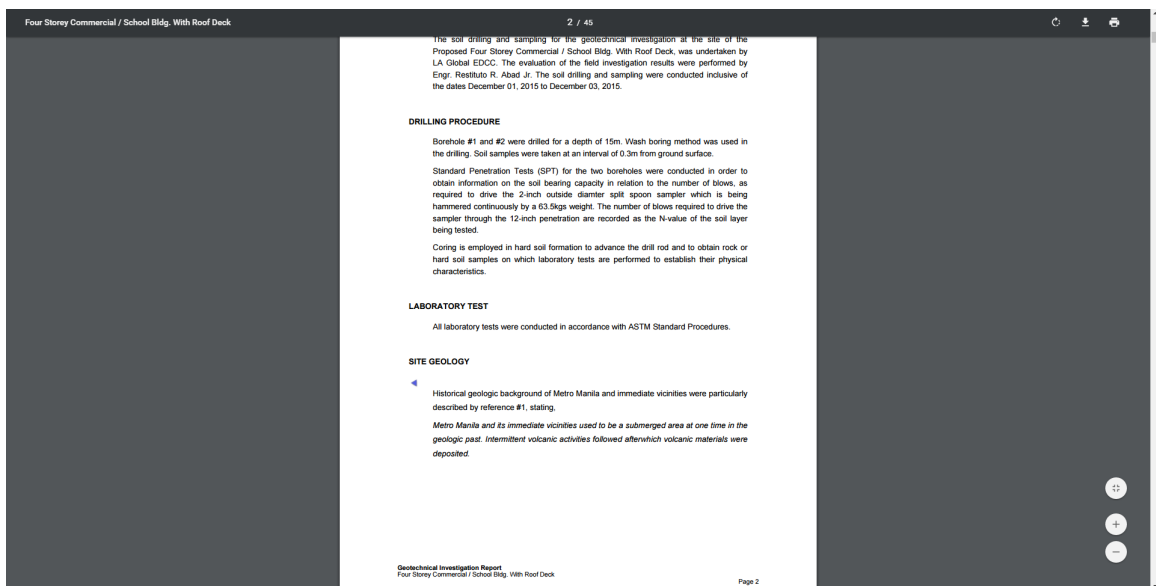


Figure 24: View Compiled Report

B. Admin View

The following figures show the pages and functionalities that only the administrator can access.

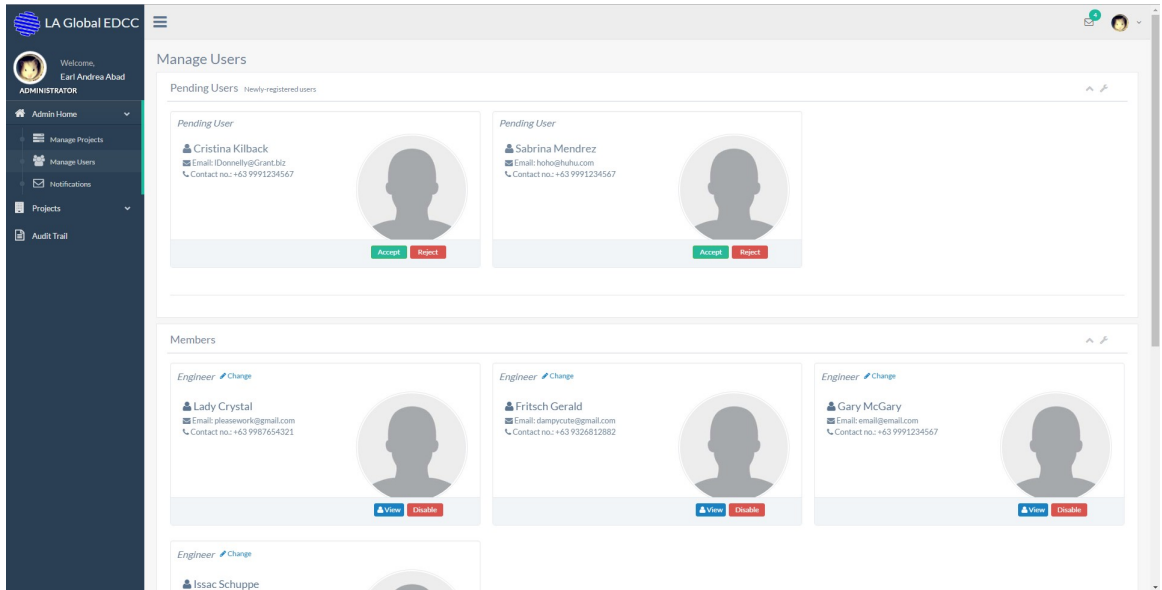


Figure 25: Manage Users

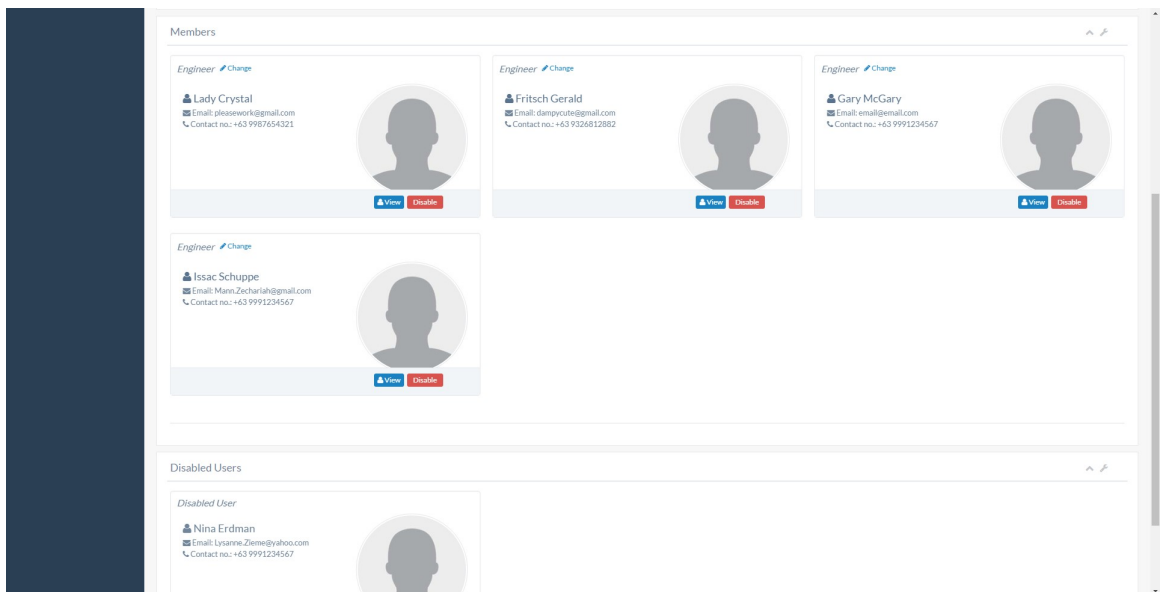


Figure 26: Manage Users

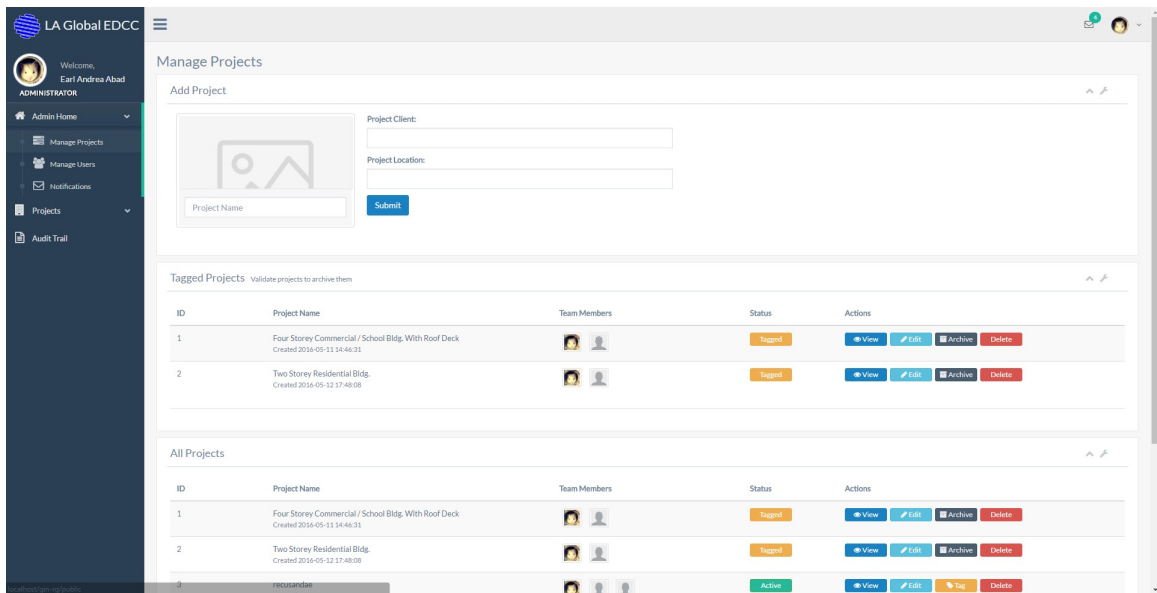


Figure 27: Manage Projects

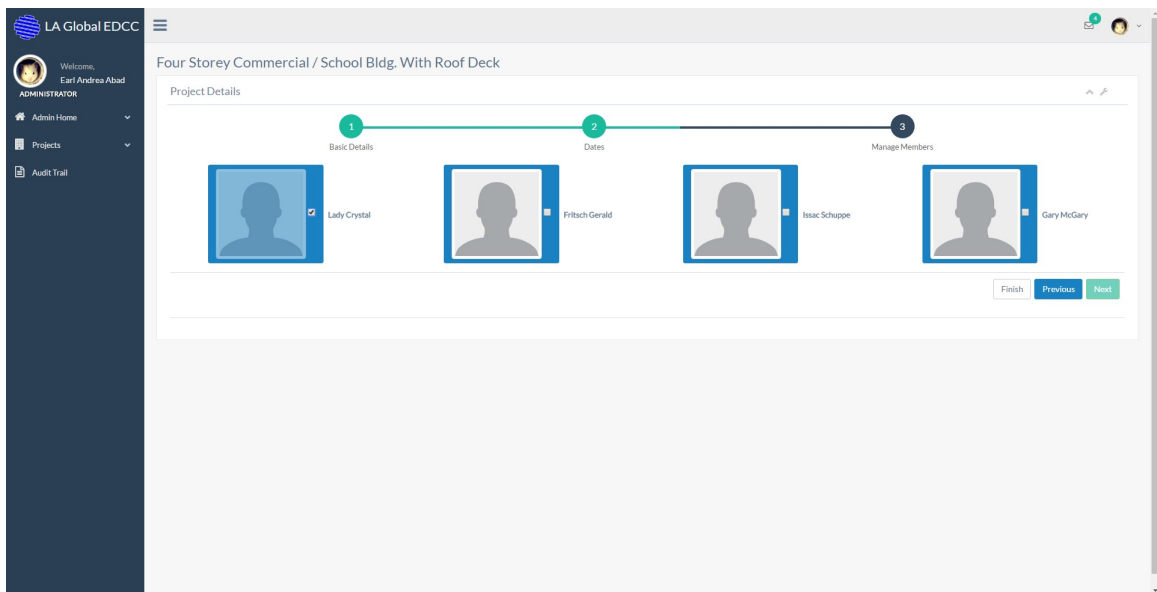


Figure 28: Add or Remove Engineers from Project

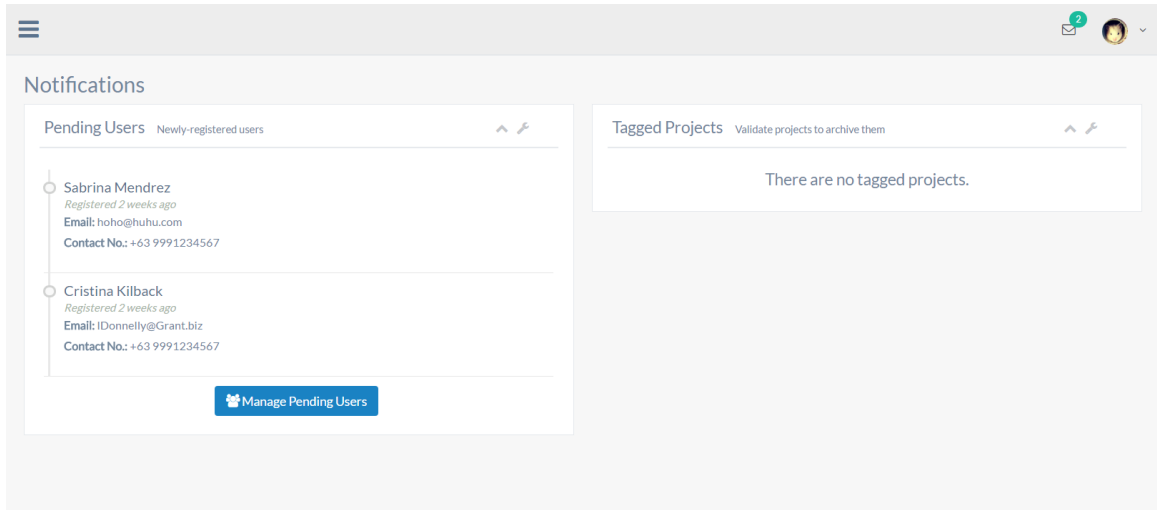


Figure 29: Receive Notifications

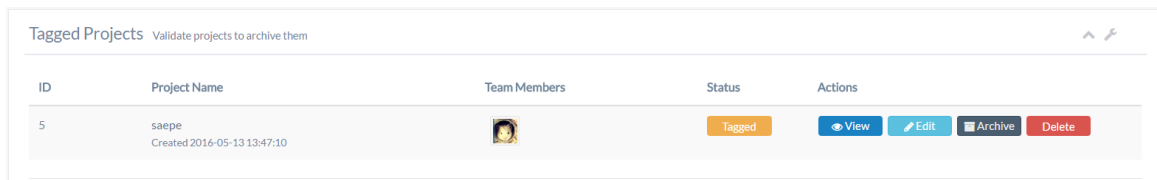


Figure 30: Archive Tagged Project

An email containing the attached compiled report can be created if the project is already tagged.

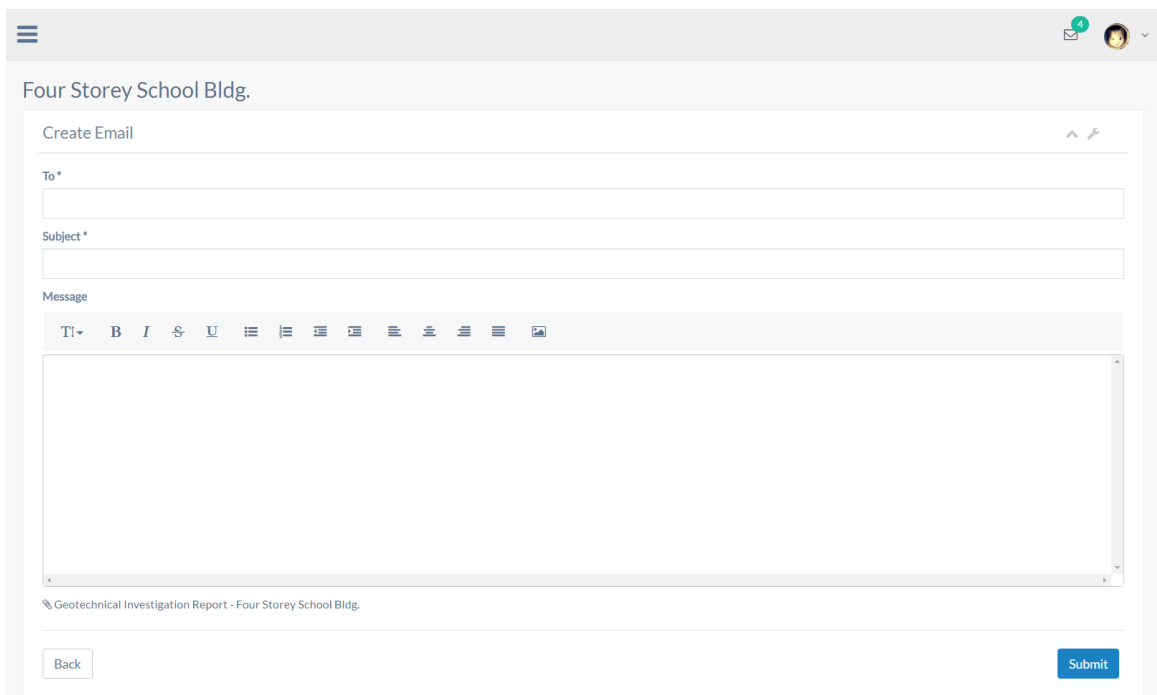


Figure 31: Create Email (with attached compiled report)

After adding complete boreholes and samples information, the administrator can manage (create, edit, view, and delete) data for computing bearing capacities. The following figures show the edit pages of the module.

The screenshot shows the 'Edit - Shallow Foundation' page for 'Four Storey School Bldg.'. The page title is 'Edit - Shallow Foundation' with a subtitle 'Compute soil bearing capacity for shallow foundation'. The main section is titled 'Borehole 1 (ID#1)'. It contains two tables of data.

B	ϕ	N_c	N_v	N_q	γ_1	γ_2	FS
2	12	5	1	3	16	16	3

D_f	SPT N-value	Soil Type	c	q_{ult}	q	Choose
0.8	17	silt	163.2	870.4	290.13	<input type="checkbox"/>
1.1	17	silt	163.2	884.8	294.93	<input type="checkbox"/>
1.4	7	silt	67.2	419.2	139.73	<input type="checkbox"/>
1.7	6	silt	57.6	385.6	128.53	<input type="checkbox"/>
2	10	silt	96	592	197.33	<input type="checkbox"/>
2.3	10	silt	96	606.4	202.13	<input type="checkbox"/>
2.6	11	silt	105.6	668.8	222.93	<input type="checkbox"/>

Figure 32: Edit Shallow Foundation

The screenshot shows the 'Edit - Deep Foundation' page for 'Four Storey School Bldg.'. The page title is 'Edit - Deep Foundation' with a subtitle 'Compute soil bearing capacity for deep foundation'. The main section is titled 'Borehole 1 (ID#1)'. It contains two tables of data.

Pile Size	K	γ	$\tan\delta$	FS	ρ	f_y	f_c	Concrete Strength (kN)	Concrete Strength (ton)
0.35	0.75	12	0.45	2	0.01	227	27	2226.36	227.51

D_f	f	$A_{surface}$	q	A_{tip}	Q_{total}	Q (kN)	Q (ton)	Choose
0.8	3.24	1.12	290.13	0.12	39.17	19.58	2	<input type="checkbox"/>
1.1	4.46	1.54	294.93	0.12	42.99	21.49	2.2	<input type="checkbox"/>
1.4	5.67	1.96	139.73	0.12	28.23	14.12	1.44	<input type="checkbox"/>
1.7	6.89	2.38	128.53	0.12	32.13	16.07	1.64	<input type="checkbox"/>
2	8.1	2.8	197.33	0.12	46.85	23.43	2.39	<input type="checkbox"/>
2.3	9.32	3.22	202.13	0.12	54.76	27.38	2.8	<input type="checkbox"/>
2.6	10.53	3.64	222.93	0.12	65.64	32.82	3.35	<input type="checkbox"/>

Figure 33: Edit Deep Foundation

Four Storey School Bldg.

Edit - Drilled Shaft Foundation Compute soil bearing capacity for drilled shaft foundation

Borehole 1 (ID#1)

Pile Size	N _c	F Coeff	f _c	f _y	SB Diameter	FS	Tension Capacity	Concrete Strength (kN)	Concrete Strength (ton)
0.25	8.2	0.3	20	227	16	3	45.64	629.78	64.36

D _f	SPT N-value	Soil Type	c	Q _{base}	f	Q _{friction}	Q _{total} (kN)	Q _{total} (ton)
0.8	17	clay	204	82.11	61.2	38.45	40.19	4.11
1.1	17	clay	204	82.11	61.2	52.87	45	4.6
1.4	7	silt	67.2	27.05	20.16	22.17	16.41	1.68
1.7	6	clay	72	28.98	21.6	28.84	19.27	1.97
2	10	clay	120	48.3	36	56.55	34.95	3.57
2.3	10	clay	120	48.3	36	65.03	37.78	3.86
2.6	11	clay	132	53.13	39.6	80.86	44.67	4.56

Figure 34: Edit Drilled Shaft Foundation

VI. Discussions

Geotechnical Investigation Reports Generator is a web-based computational and information system that computes necessary calculations and generates geotechnical investigation reports for LA Global EDCC's site investigation projects.

The system is designed for two types of users: (1) the system administrator, which is also the lead engineer of the whole company, and (2) the regular users, i.e. the rest of the engineers. The system administrator can use all the functionalities of the system, whereas the regular users can only access limited functionalities.

The system administrator is able to register his account before deploying the system to a live host. The regular users, on the other hand, are able to register their own accounts after deploying the system. They will be able to login once they have been accepted by the administrator.

Once logged in, both the regular users and the administrator are able to do the following: edit their user information, edit project information, manage borehole and sample information, manage subreports and sections, tag projects, and view and download compiled reports. Only the administrator is allowed to do the following: approve or reject pending user accounts, disable user account (if the user is already accepted), add and remove projects, add and remove engineers to and from projects, calculate bearing capacities to be used as basis for the recommendations, receive notifications regarding newly registered users and tagged projects, archive tagged projects, and send tagged reports through email.

VII. Conclusions

The geotechnical investigation reports generator (GeoTest) allows users to generate reports without the use of multiple applications. It can automatically generate the required graphs and logs, as the company needed. It also allows the lead engineer to compute for soil bearing capacities which are to be used as basis for their recommendations on which foundation to use.

Since the system is web-based, it also enables users to collaborate and view audit trail, thus attaining accountability in every project.

Moreover, since the users do not need to individually input and create everything (even the graphs and logs), reports are reduced of incorrect outputs.

VIII. Recommendations

The system can be extended by adding more capability to its user account management. Users can have different roles (e.g. lead engineer, associate engineer, etc.) in different projects. The system administrator and the lead engineer are to be separated in this design. Thus, the system administrator will only be the one who manages the operation of the system and its database.

Moreover, future developers can also add maps in order to view the site using coordinates that are to be inputted by the users. More preferably, they can add the top view of the investigation site, in order to have a better look at the boreholes' locations.

For better visualization, graphs and logs can be improved aesthetically. More borehole or sample details can also be added so that the users (or anyone will read the document) can analyze the graphs and logs more effectively.

IX. Bibliography

- [1] “Geotechnical investigations & testing methods.” <http://www.geogroup.co.za/geotechnical-investigations/geotechnical-investigations-testing>. Accessed: 2015-11-29.
- [2] J. B. Adeyeri, ed., *Technology and Practice in Geotechnical Engineering*. IGI Global, 2015.
- [3] British Geological Survey, “What is a borehole?.” <https://www.bgs.ac.uk/data/boreholescans/docs/borehole.pdf>. Accessed: 2015-11-27.
- [4] C. Zhang, A. Li, and L. Zhou, “An extensible borehole log model,” in *2011 19th International Conference on Geoinformatics*, (Shanghai), pp. 1–5, IEEE, 2011.
- [5] D. F. McCarthy, *Essentials of Soil Mechanics and Foundations: Basic Geotechnics*. Virginia, USA: Reston Publishing Company, Inc., 1986.
- [6] J. G. Atat, I. O. Akpabio, and N. J. George, “Allowable bearing capacity for shallow foundation in eket local government area, akwa ibom state, southern nigeria,” *International Journal of Geosciences*, vol. 4, 2013.
- [7] “La global edcc: About.” <http://www.laglobaledcc.com>. Accessed: 2015-10-29.
- [8] New York, *New York City Building Code*. 2008.
- [9] B. Abubakar and O. Adekola, “Assessment of borehole water quality in yola-jimeta metropolis, nigeria,” *International Journal of Water Resources and Environmental Engineering*, vol. 4, 2012.
- [10] I. Fubara-Manuel and R. B. Jumbo, “The effect of septic tank locations on borehole water quality in port harcourt, nigeria,” *International Journal of Engineering and Technology*, vol. 4, 2014.

- [11] J. O. Jeje and K. T. Oladepo, “Assessment of heavy metals of boreholes and hand dug wells in ife north local government area of osun state, nigeria,” *International Journal of Science and Technology*, vol. 3, 2014.
- [12] L. Palamuleni and M. Akoth, “Physico-chemical and microbial analysis of selected borehole water in mahikeng, south africa,” *International Journal of Environmental Research and Public Health*, vol. 12, 2015.
- [13] S. Gaci, “Regularity analysis of oil exploration borehole logs,” in *2013 8th International Symposium on Image and Signal Processing and Analysis (ISPA)*, (Trieste, Italy), pp. 456–461, IEEE, 2013.
- [14] S. Imran and N. Khan, “Experimental Study of Borehole Logs for Oil and Gas Exploration,” in *2015 29th International Conference on Advanced Information Networking and Applications Workshops*, (Gwangju), pp. 117–122, IEEE, 2015.
- [15] O. K. Festus, Adeoti, Lukman, O. Sunday, and K. Akintunde, “Investigation of a proposed four storey building sites using geophysical and laboratory engineering testing methods in lagos, nigeria,” *International Journal of Scientific Research in Knowledge*, vol. 2, 2014.
- [16] N. Ebehikhalu and P. Dawam, “Spatial analysis of building collapse in nigeria: A study of the causes and problems,” *Journal of Economics and Sustainable Development*, vol. 5, 2014.
- [17] H. O. Nwankwoala, A. N. Amadi, F. A. Ushie, T. Warmate, and C. J. Eze, “Determination of subsurface geotechnical properties for foundation design and construction in akenfa community, bayelsa state, nigeria,” *American Journal of Civil Engineering and Architecture*, vol. 2, 2014.
- [18] K. O. Yusuf, A. M. Ibrahim, and J. O. Famakinwa, “Productivity and quality analysis of selected boreholes in osun and kwara states, nigeria,” *Agricultural Engineering International: CIGR Journal*, vol. 14, 2012.

- [19] P. O. Youdeowei and H. O. Nwankwoala, "Studies on sub-soil characteristics of sand deposits in some parts of bayelsa state, eastern niger delta, nigeria," *Journal of Soil Science and Environmental Management*, vol. 2, 2011.
- [20] H. O. Nwankwoala and E. Oborie, "Geo-technical investigation and characterization of sub-soils in yenagoa, bayelsa state, central niger delta, nigeria," *Civil and Environmental Research*, vol. 6, 2014.
- [21] "gint - geotechnical and geoenvironmental software." <https://www.bentley.com/en/products/brands/gint>. Accessed: 2015-11-27.
- [22] Strater, "Strater[®] 4." <http://www.goldensoftware.com/products/strater>. Accessed: 2015-11-27.
- [23] "Logplot7 borehole log software: Features." <https://www.rockware.com/product/featuresLobby.php?id=176&category=337>. Accessed: 2015-11-27.
- [24] "Logplot cost/pricing. how to purchase/buy logplot." <https://www.rockware.com/product/pricing.php?id=176>. Accessed: 2015-11-27.
- [25] "Borehole log/boring log/well installation log package." <http://www.geosystemsoftware.com/products/ld5/>. Accessed: 2015-11-27.
- [26] Z. Xunguo, "Dalian Ikea Mall Work Geotechnical Investigation Conclusions Analyzed And Recommendations," in *2011 International Conference on Electric Technology and Civil Engineering (ICETCE)*, (Lushan), pp. 201–204, IEEE, 2011.
- [27] A. N. Amadi, W. G. Akande, I. A. Okunlola, M. O. Jimoh, and F. D. G., "Assessment of the geotechnical properties of lateritic soils in minna, north central nigeria for road design and construction," *American Journal of Mining and Metallurgy*, vol. 3, 2015.

- [28] S. Logsdon, D. Clay, D. Moore, and T. Tsegaye, eds., *Soil Science: Step-by-Step Field Analysis*. Wisconsin, USA: Soil Science Society of America, 2008.
- [29] R. F. Craig, *Craig's Soil Mechanics*. New York, USA: Taylor & Francis, 2004.
- [30] G. M. Reeves, I. Sims, and J. C. Cripps, eds., *Clay Materials Used in Construction*. London, England: The Geological Society, 2006.
- [31] I. Watson, *Hydrology: An Environmental Approach*. USA: CRC Press LLC, 1993.
- [32] B. M. Das, *Shallow Foundations: Bearing Capacity and Settlement*. USA: CRC Press LLC, 1999.
- [33] D. L. Nguyen, S. Ohtsuka, T. Hoshina, K. Isobe, and K. Kaneda, "Ultimate bearing capacity analysis of ground against inclined load by taking account of nonlinear property of shear strength," *International Journal of GEOMATE*, vol. 5, 2013.
- [34] S. K. Vanapalli and F. M. O. Mohamed, "Bearing capacity and settlement of footings in unsaturated sands," *International Journal of GEOMATE*, vol. 5, 2013.
- [35] Laravel, "Laravel - the php framework for web artisans." <http://www.laravel.com>. Accessed: 2015-12-1.
- [36] D. Gafitescu, "Goodbye codeigniter, hello laravel." <http://www.sitepoint.com/goodbye-codeigniter-hello-laravel/>, May 2013. Accessed: 2015-12-1.
- [37] B. Skvorc, "The best php framework for 2015: Sitepoint survey results." <http://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>, March 2015. Accessed: 2015-12-1.

- [38] Google Trends, “Google trends - web search interest: laravel, codeigniter, cakephp, zend framework, yii - worldwide, jan 2010 - jul 2015.” <http://www.google.com/trends/explore/>, 2015. Accessed: 2015-12-1.

X. Appendix

Listing 1: composer.json

```
{
  "name": "laravel/laravel",
  "description": "The Laravel Framework.",
  "keywords": ["framework", "laravel"],
  "license": "MIT",
  "type": "project",
  "require": {
    "php": ">=5.5.9",
    "laravel/framework": "5.1.*",
    "laravelcollective/html": "5.1.*",
    "spipu/html2pdf": "^4.5",
    "guzzlehttp/guzzle": "~4.0"
  },
  "require-dev": {
    "fzaninotto/faker": "~1.4",
    "mockery/mockery": "0.9.*",
    "phpunit/phpunit": "5.2",
    "phpspec/phpspec": "~2.1"
  },
  "autoload": {
    "classmap": [
      "database"
    ],
    "psr-4": {
      "App\\": "app/"
    }
  },
  "autoload-dev": {
    "classmap": [
      "tests/TestCase.php"
    ]
  },
  "scripts": {
    "post-install-cmd": [
      "php_artisan_clear_compiled",
      "php_artisan_optimize"
    ],
    "pre-update-cmd": [
      "php_artisan_clear_compiled"
    ],
    "post-update-cmd": [
      "php_artisan_optimize"
    ],
    "post-root-package-install": [
      "php_r_\"copy('env.example', 'env');\""
    ],
    "post-create-project-cmd": [
      "php_artisan_key_generate"
    ]
  },
  "config": {
    "preferred-install": "dist"
  }
}
```

Listing 2: AuthController.php

```
<?php
namespace App\Http\Controllers\Auth;

use Auth;
use Session;
use App\Activity;
use App\User;
use Validator;
use App\Http\Controllers\Controller;
use Illuminate\Foundation\Auth\ThrottlesLogins;
use Illuminate\Foundation\Auth\
    AuthenticatesAndRegistersUsers;

class AuthController extends Controller
{
    /*
     | Registration & Login Controller
     | This controller handles the registration
     | of new users, as well as the
     | authentication of existing users. By
     | default, this controller uses
     | a simple trait to add these behaviors.
     | Why don't you explore it?
     | */
}
```

```
use AuthenticatesAndRegistersUsers,
    ThrottlesLogins;

private $redirectTo = '/';
private $maxLoginAttempts = 6; //if you
    want maxLoginAttempts to be other
    than 5

/**
 * Create a new authentication controller
    instance.
 *
 * @return void
 */
public function __construct()
{
    $this->middleware('guest', ['except'
        => 'getLogout']);
}

/**
 * Get a validator for an incoming
    registration request.
 *
 * @param array $data
 * @return \Illuminate\Contracts\
    Validation\Validator
 */
protected function validator(array $data)
{
    return Validator::make($data, [
        'firstname' => 'required|max:255',
        'middlename' => 'max:255',
        'lastname' => 'required|max:255',
        'email' => 'required|email|max:
            255|unique:users',
        'contact' => 'required|max:14',
        'password' => 'required|confirmed|
            min:6',
    ]);
}

/**
 * Create a new user instance after a
    valid registration.
 *
 * @param array $data
 * @return User
 */
protected function create(array $data)
{
    $user = User::create([
        'firstname' => $data['firstname'],
        'middlename' => $data['middlename'],
        'lastname' => $data['lastname'],
        'email' => $data['email'],
        'contact' => $data['contact'],
        'password' => bcrypt($data['
            password'])
    ]);

    $user_name = $data['firstname'] . " " .
        $data['lastname'];
    // Add the activity to activity table
    Activity::create([
        'proj_id' => null,
        'user_id' => $user->id,
        'notify_admin' => 1,
        'activity_type' => "User_<i>
            $user_name</i>_has_registered
            _an_account."
    ]);

    return $user;
}
}
```

Listing 3: PasswordController

```
<?php
namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use Illuminate\Foundation\Auth\ResetsPasswords;

class PasswordController extends Controller
```

```

{
    /*
    | Password Reset Controller
    | This controller is responsible for
    | handling password reset requests
    | and uses a simple trait to include this
    | behavior. You're free to
    | explore this trait and override any
    | methods you wish to tweak.
    |
    */
    use ResetsPasswords;

    /**
     * Create a new password controller
     * instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('guest');
    }
}

```

Listing 4: ActivityController.php

```

<?php
namespace App\Http\Controllers;

use DB;
use Gate;
use App\Activity;
use Illuminate\Http\Request;
use App\Http\Requests;
use App\Http\Controllers\Controller;

class ActivityController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        if (Gate::denies('show-activities')) {
            abort(403);
        }

        $activities = DB::table('activities')
            ->leftJoin('users', 'activities.
                user_id', '=', 'users.id')
            ->select('activities.id', '
                activities.proj_id', '
                activities.user_id', 'users.
                firstname', 'users.lastname',
                'activities.activity_type',
                'activities.created_at')
            ->orderBy('activities.id', 'desc')
            ->paginate(20);

        return view('activities.index', array(
            'activities' => $activities));
    }

    /**
     * Display all notifications.
     *
     * @return \Illuminate\Http\Response
     */
    public function showNotifications()
    {
        if (Gate::denies('show-notifications')) {
            abort(403);
        }

        $pending_users = DB::table('users')->
            where('status', 0)->orderBy('
                created_at', 'desc')->get();
        $staged_projects = DB::table('projects')->
            where('proj-status', 1)->
            orderBy('created_at', 'desc')->
            get();

        return view('activities.notifications',
            array('pending-users' =>
                $pending_users, 'tagged-projects'
                => $staged_projects));
    }
}

```

Listing 5: BoreholeController.php

```

<?php
namespace App\Http\Controllers;

use Auth;
use DB;
use Helper;
use Gate;
use App\Activity;
use App\Borehole;
use App\Project;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests\Borehole\
    StoreBoreholeRequest;
use App\Http\Requests\Borehole\
    UpdateBoreholeRequest;

class BoreholeController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        //
    }

    /**
     * Show the form for creating a new
     * resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create($id)
    {
        $project = Project::findOrFail($id);

        if (Gate::denies('borehole', $project)) {
            abort(403);
        }

        return view('boreholes.create', array(
            'project' => $project));
    }

    /**
     * Store a newly created resource in
     * storage.
     *
     * @param \Illuminate\Http\Request
     * $request
     * @return \Illuminate\Http\Response
     */
    public function store(StoreBoreholeRequest
        $request)
    {
        // Create borehole
        $borehole = Borehole::create([
            'proj_id' => $request->proj_id,
            'bh_start_date' => $request->
                bh_start_date,
            'bh_end_date' => $request->
                bh_end_date,
            'bh_num' => $request->bh_num,
            'bh_max_depth' => $request->
                bh_max_depth,
            'bh_weather' => $request->
                bh_weather,
            'bh_operator' => $request->
                bh_operator,
            'bh_inspector' => $request->
                bh_inspector,
            'bh_drill_rig' => (empty(trim(
                $request->bh_drill_rig))) ? ,
                YBM' : $request->bh_drill_rig
            ,
            'bh_drill_pump' => (empty(trim(
                $request->bh_drill_pump))) ?
                'Shark' : $request->
                bh_drill_pump,
            'bh_hammer_weight' => (empty(trim(
                $request->bh_hammer_weight)))
                ? 63.5 : $request->
                bh_hammer_weight,
            'bh_height_of_drop' => (empty(trim(
                $request->bh_height_of_drop))
                ) ? 1.6 : $request->
                bh_height_of_drop,
            'bh_casing' => $request->bh_casing
            ,
            'bh_rod' => $request->bh_rod,
            'bh_sampler' => $request->
                bh_sampler,

```

```

        'bh_water_elevation' => $request->
            bh_water_elevation,
        'bh_surface_elevation' => $request
            ->bh_surface_elevation
    });

    // Increment the number of boreholes
    in project table
    DB::table('projects')->whereId(
        $request->proj_id)->increment('
        proj_num_of_boreholes');

    $project = Project::findOrFail(
        $request->proj_id);

    // Add the activity to activity table
    Activity::create([
        'proj_id' => $request->proj_id,
        'user_id' => Auth::user()->id,
        'activity_type' => "Borehole_<i>
        $request->bh_num</i>-in-
        project_<i>$project->
        proj_name</i>-has-been-
        created."
    ]);

    $flash_message = "Borehole_<i>$request->
    bh_num-in-project_<i>$project->
    proj_name-has-been-successfully-
    created.";
    return redirect()->route('boreholes.
    show', $borehole->id)->with('
    success', $flash_message);
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    $borehole = Borehole::findOrFail($id);
    $project = Borehole::find($borehole->
        id)->project;

    if (Gate::denies('show-borehole',
        $project)) {
        abort(403);
    }

    $samples = Borehole::find($borehole->
        id)->samples;

    return view('boreholes.show', array('
        borehole' => $borehole, 'project'
        => $project, 'samples' =>
        $samples));
}

/**
 * Show the form for editing the specified
    resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    $borehole = Borehole::findOrFail($id);
    $project = Borehole::find($borehole->
        id)->project;

    if (Gate::denies('borehole', $project)
        ) {
        abort(403);
    }

    return view('boreholes.edit', array('
        borehole' => $borehole, 'project'
        => $project));
}

/**
 * Update the specified resource in
    storage.
 *
 * @param \Illuminate\Http\Request
        $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(
    UpdateBoreholeRequest $request, $id)
{
    $borehole = Borehole::findOrFail($id);
    $project = Project::findOrFail(
        $request->proj_id);

    // Update borehole information
    $borehole->update([
        'bh_start_date' => $request->
            bh_start_date,
        'bh_end_date' => $request->
            bh_end_date,
        'bh_num' => $request->bh_num,
        'bh_max_depth' => $request->
            bh_max_depth,
        'bh_weather' => $request->
            bh_weather,
        'bh_operator' => $request->
            bh_operator,
        'bh_drill_rig' => $request->
            bh_drill_rig,
        'bh_drill_pump' => $request->
            bh_drill_pump,
        'bh_hammer_weight' => $request->
            bh_hammer_weight,
        'bh_height_of_drop' => $request->
            bh_height_of_drop,
        'bh_casing' => $request->bh_casing,
        'bh_rod' => $request->bh_rod,
        'bh_sampler' => $request->
            bh_sampler,
        'bh_water_elevation' => $request->
            bh_water_elevation,
        'bh_surface_elevation' => $request
            ->bh_surface_elevation
    ]);

    // Add the activity to activity table
    Activity::create([
        'proj_id' => $project->id,
        'user_id' => Auth::user()->id,
        'activity_type' => "Borehole_<i>
        $request->bh_num</i>-in-
        project_<i>$project->
        proj_name</i>-has-been-
        modified."
    ]);

    $flash_message = "Borehole_<i>$request->
    bh_num-in-project_<i>$project->
    proj_name-has-been-successfully-
    modified.";
    return redirect()->route('boreholes.
    show', $borehole->id)->with('
    success', $flash_message);
}

/**
 * Remove the specified resource from
    storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    $borehole = Borehole::findOrFail($id);

    if (Gate::denies('borehole', $borehole
        ->project)) {
        abort(403);
    }

    $bh_id = $borehole->id;
    $bh_num = $borehole->bh_num;
    $proj_id = $borehole->proj_id;

    $borehole->delete();

    return BoreholeController::
        recordDeleteActivity($proj_id,
            $bh_num);
}

public function recordDeleteActivity(
    $proj_id, $bh_num)
{
    // Decrement the number of boreholes
    in project table
    DB::table('projects')->whereId(
        $proj_id)->decrement('
        proj_num_of_boreholes');

    // Change borehole number of the rest
    of the boreholes accordingly
    $boreholes = Borehole::where('proj_id',
        $proj_id)
        ->where('bh_num', '>', $bh_num)
        ->get();

    foreach($boreholes as $borehole) {
        DB::table('boreholes')->whereId(
            $borehole->id)->decrement('

```

```

        bh.num');
    }

    $project = Project::findOrFail(
        $proj_id);
    // Add the activity to activity table
    Activity::create([
        'proj_id' => null,
        'user_id' => Auth::user()->id,
        'activity_type' => "Borehole_<i>
        $bh_num</i>-in_<i>$project->
        proj_name</i>-has-been-
        deleted."
    ]);

    $flash_message = "Borehole_<i>$bh_num-in-
    project_<i>$project->proj_name-has-
    been_successfully_deleted.";
    return redirect()->route('projects.
    show', $proj_id)->with('success',
    $flash_message);
}
}
}

```

Listing 6: Controller.php

```

<?php
namespace App\Http\Controllers;

use Illuminate\Foundation\Bus\DispatchesJobs;
use Illuminate\Routing\Controller as
    BaseController;
use Illuminate\Foundation\Validation\
    ValidatesRequests;
use Illuminate\Foundation\Auth\Access\
    AuthorizesRequests;

abstract class Controller extends
    BaseController
{
    use AuthorizesRequests, DispatchesJobs,
        ValidatesRequests;
}

```

Listing 7: DataPointController.php

```

<?php
namespace App\Http\Controllers;

use Auth;
use DB;
use Input;
use Gate;
use App\Activity;
use App\DataPoint;
use App\Project;
use App\Sample;
use App\Subreport;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests\DataPoint\
    StoreDataPointRequest;
use App\Http\Requests\DataPoint\
    UpdateDataPointRequest;

class DataPointController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index($id)
    {
        $project = Project::findOrFail($id);

        if (Gate::denies('index-data-point',
            $project)) {
            abort(403);
        }

        $boreholes = Project::findOrFail(
            $project->id)->boreholes;
        $samples = Sample::join('boreholes', '
            samples.bh_id', '=', 'boreholes.
            id')
            ->where('boreholes.proj_id',
                $project->id)
            ->select('samples.id', 'samples.
                bh_id', 'samples.sample_num')
            ->get();

        $matchThese = ['proj_id' => $project->
            id, 'sb_key' => 0];
    }
}

```

```

$subreport = Subreport::where(
    $matchThese)->first();
$data_points = DataPoint::where('
    subreport_id', $subreport->id)->
    get();

DataPointController::isUpdated(
    $project->id, $samples,
    $data_points);

return view('datapoints.index', array(
    'project' => $project, 'boreholes'
    => $boreholes, 'samples' =>
    $samples, 'subreport' =>
    $subreport, 'data_points' =>
    $data_points));
}

/**
 * Show the form for creating a new
 * resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create($id)
{
    $project = Project::findOrFail($id);

    if (Gate::denies('data-point',
        $project)) {
        abort(403);
    }

    // Check if subreport is already in
    // the database
    $matchThese = ['proj_id' => $project->
        id, 'sb_key' => 0];
    $match = DB::table('subreports')->
        where($matchThese)->first();

    if (is_null($match)) {
        $boreholes = DB::table('boreholes'
        )
            ->join('projects', 'boreholes.
                proj_id', '=', 'projects.
                id')
            ->where('boreholes.proj_id',
                $id)
            ->select('boreholes.id')
            ->get();

        if (sizeof($boreholes) > 0) { //
            // If there are boreholes in
            // project
            $i = 0;
            foreach($boreholes as
                $borehole) {
                $samples = DB::table('
                samples')
                    ->join('boreholes', '
                    samples.bh_id', '
                    =', 'boreholes.id'
                    )
                    ->where('samples.bh_id'
                    , $borehole->id)
                    ->select('samples.id')
                    ->get();

                $boreholes[$i] = $borehole
                    ->id;
                $j = 0;

                if (sizeof($samples) > 0) {
                    // If there are
                    // samples in current
                    // borehole
                    foreach($samples as
                        $sample) {
                            $counts[$i][$j] =
                                $sample->id;
                            $j++;
                        }
                    }
                }
            }
            $i++;
        }
        else {
            $counts = 0;
        }

        if (!isset($counts)) {
            $counts = 0;
        }

        DataPointController::
            checkPrerequisites($counts);

        return view('datapoints.create',
            array('project' => $project,

```

```

        'boreholes' => $boreholes, '
        counts' => $counts));
    }
    else {
        $flash_message = "Grain_Soil_
        Analysis_in_project_{$project
        ->proj_name_already_exists_}
        You_cannot_create_another_
        instance_of_the_subreport.";
        return redirect()->route('projects
        .show', $project->id)->with('
        error', $flash_message);
    }
}

/**
 * Store a newly created resource in
 * storage.
 *
 * @param \Illuminate\Http\Request
 * $request
 * @return \Illuminate\Http\Response
 */
public function store(
    StoreDataPointRequest $request)
{
    $project = Project::findOrFail(
        $request->proj_id);

    // Check if subreport is already in
    // the database
    $matchThese = ['proj_id' => $project->
        id, 'sb_key' => 0];
    $match = DB::table('subreports')->
        where($matchThese)->first();

    if (is_null($match)) {
        // Store data to subreports table
        $input_sb['sb_key'] = 0;
        $input_sb['proj_id'] = $project->
            id;
        $subreport = Subreport::create(
            $input_sb);

        // Store data to data_points table
        for($i = 0; $i < sizeof($request->
            sample_id); $i++) {
            DataPoint::create([
                'sample_id' => $request->
                    sample_id[$i],
                'subreport_id' =>
                    $subreport->id,
                'x_int' => $request->x_int
                    [$i],
                'y_int' => $request->y_int
                    [$i]
            ]);
        }

        // Add the activity to activity
        // table
        Activity::create([
            'proj_id' => $project->id,
            'user_id' => Auth::user()->id,
            'activity_type' => "Data_
            points_for_the_Grain_Soil_
            Analysis_in_project_<i>
            $project->proj_name</i>_
            has_been_created."
        ]);

        $flash_message = "Data_points_for_
        the_Grain_Soil_Analysis_in_
        project_{$project->proj_name_}
        has_been_successfully_created
        .";
        return redirect()->route('
        subreports.showGSA',
            $subreport->id)->with('
            success', $flash_message);
    }
    else {
        $flash_message = "Grain_Soil_
        Analysis_in_project_{$project
        ->proj_name_already_exists_}
        You_cannot_create_another_
        instance_of_the_subreport.";
        return redirect()->route('projects
        .show', $project->id)->with('
        error', $flash_message);
    }
}

/**
 * Display the specified resource.
 *
 * @param int $id

```

```

 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    //
}

/**
 * Show the form for editing the specified
 * resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    $project = Project::findOrFail($id);

    if (Gate::denies('data-point',
        $project)) {
        abort(403);
    }

    $boreholes = Project::findOrFail(
        $project->id)->boreholes;
    $samples = Sample::join('boreholes', '
        samples.bh_id', '=', 'boreholes.
        id')
        ->where('boreholes.proj_id',
            $project->id)
        ->select('samples.id', 'samples.
            bh_id', 'samples.sample.num')
        ->get();

    $matchThese = ['proj_id' => $project->
        id, 'sb_key' => 0];
    $subreport = Subreport::where(
        $matchThese)->first();
    $data_points = DataPoint::where('
        subreport_id', $subreport->id)->
        get();

    DataPointController::isUpdated(
        $project->id, $samples,
        $data_points);

    return view('datapoints.edit', array('
        project' => $project, 'boreholes'
        => $boreholes, 'samples' =>
        $samples, 'subreport' =>
        $subreport, 'data_points' =>
        $data_points));
}

/**
 * Update the specified resource in
 * storage.
 *
 * @param \Illuminate\Http\Request
 * $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(
    UpdateDataPointRequest $request, $id)
{
    // Update data in data_points table
    for($i = 0; $i < sizeof($request->
        data_point_id); $i++) {
        $data_point = DataPoint::
            findOrFail($request->
                data_point_id[$i]);

        $data_point->update([
            'x_int' => $request->x_int[$i]
                ],
            'y_int' => $request->y_int[$i]
        ]);
    }

    $project = Project::findOrFail(
        $request->proj_id);
    // Add the activity to activity table
    Activity::create([
        'proj_id' => $project->id,
        'user_id' => Auth::user()->id,
        'activity_type' => "Data_points_
        for_the_Grain_Soil_Analysis_
        in_project_<i>$project->
        proj_name</i>_has_been_
        modified."
    ]);

    $flash_message = "Data_points_for_the_
    Grain_Soil_Analysis_in_project_
    $project->proj_name_has_been_
    successfully_modified.";
    return redirect()->route('subreports.
    showGSA', $id)->with('success',

```



```

    $samples = Sample::join('boreholes', '
        samples.bh_id', '=', 'boreholes.
        id')
    ->where('boreholes.proj_id',
        $project->id)
    ->select('samples.id', 'samples.
        bh_id', 'samples.sample_num',
        'samples.sample_to', '
        samples.sample_num_of_blows')
    ->groupBy('samples.id')
    ->orderBy('samples.bh_id')
    ->orderBy('samples.id')
    ->get();
    $shallows = Shallow::where('proj_id',
        $project->id)->get();
    $deeps = Deep::where('proj_id',
        $project->id)->get();

    DeepController::isUpdated($project->id
        , sizeof($samples), sizeof($deeps
        ), sizeof($shallows));

    return view('deeps.index', array('
        project' => $project, 'boreholes'
        => $boreholes, 'samples' =>
        $samples, 'deeps' => $deeps));
}

/**
 * Show the form for creating a new
 * resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create($id)
{
    $project = Project::findOrFail($id);

    if (Gate::denies('deep', $project)) {
        abort(403);
    }

    $deeps = $project->deeps;

    if (sizeof($deeps) == 0) {
        $boreholes = Project::findOrFail(
            $project->id)->boreholes;
        $samples = Sample::join('boreholes
            ', 'samples.bh_id', '=', '
            boreholes.id')
            ->where('boreholes.proj_id',
                $project->id)
            ->select('samples.id', '
                samples.bh_id', 'samples.
                sample_num', 'samples.
                sample_to')
            ->get();
        $shallows = Shallow::where('
            proj_id', $project->id)->get
            ();

        DeepController::checkPrerequisites
            ($samples, $shallows,
            $project);

        return view('deeps.create', array(
            'project' => $project, '
            boreholes' => $boreholes, '
            samples' => $samples, '
            shallows' => $shallows));
    }
    else {
        $flash_message = "SBC_data_for_
            deep_foundation_in_project_
            $project->proj_name_already_
            exists.You_cannot_create_
            another_instance_of_the_data.
            ";
        return redirect()->route('projects
            .show', $project->id)->with('
            error', $flash_message);
    }
}

/**
 * Store a newly created resource in
 * storage.
 *
 * @param \Illuminate\Http\Request
 * $request
 * @return \Illuminate\Http\Response
 */
public function store(StoreDeepRequest
    $request)
{
    $project = Project::findOrFail(
        $request->proj_id);
    $j = $k = 0;

    for($i = 0; $i < $project->
        proj_num_of_boreholes; $i++) {
        while($j < $request->
            while($j < $request->
                num_of_samples[$i]) {
                    $deep = Deep::create([
                        'proj_id' => $project->id,
                        'bh_id' => $request->bh_id
                            [$i],
                        'sample_id' => $request->
                            sample_id[$k],
                        'deep_pile_size' =>
                            $request->
                            deep_pile_size[$i],
                        'deep_K' => $request->
                            deep_K[$i],
                        'deep_gamma' => $request->
                            deep_gamma[$i],
                        'deep_tan' => $request->
                            deep_tan[$i],
                        'deep_fs' => $request->
                            deep_fs[$i],
                        'deep_rho' => $request->
                            deep_rho[$i],
                        'deep_fy' => $request->
                            deep_fy[$i],
                        'deep_fc' => $request->
                            deep_fc[$i],
                        'deep_concrete_strength_kn
                            ' => $request->
                            deep_concrete_strength_kn
                            [$i],
                        '
                            deep_concrete_strength_ton
                            ' => $request->
                            deep_concrete_strength_ton
                            [$i],
                        '
                            deep_f' => $request->
                            deep_f[$k],
                        'deep_Asurface' =>
                            $request->
                            deep_Asurface[$k],
                        'deep_q' => $request->
                            deep_q[$k],
                        'deep_Atip' => $request->
                            deep_Atip[$k],
                        'deep_Qtotal' => $request
                            ->deep_Qtotal[$k],
                        'deep_Q.kn' => $request->
                            deep_Q.kn[$k],
                        'deep_Q.ton' => $request->
                            deep_Q.ton[$k]
                    ]);
                    $j++;
                    $k++;
                }
                $j = 0;
            }

            if((sizeof($request->deep_is_chosen) >
                0) && ($request->deep_is_chosen
                [0] > -1)) {
                    for($i = 0; $i < sizeof($request->
                        deep_is_chosen); $i++) {
                        for($j = 0; $j < sizeof(
                            $request->sample_id); $j
                            ++){
                            if($request->
                                deep_is_chosen[$i] ==
                                $request->sample_id[
                                    $j]) {
                                    $deep = Sample::find(
                                        $request->
                                        sample_id[$j])->
                                        deep;
                                    $deep->update(['
                                        deep_is_chosen'
                                        => 1]);
                                }
                            }
                        }
                    }

                    // Add the activity to activity table
                    Activity::create([
                        'proj_id' => $project->id,
                        'user_id' => Auth::user()->id,
                        'activity_type' => "Data_for_SBC_
                            for_deep_foundation_in_
                            project_<i>$project->
                            proj_name</i>_has_been_
                            created."
                    ]);

                    $flash_message = "Data_for_SBC_for_
                        deep_foundation_in_project_
                        $project->proj_name_has_been_
                        successfully_saved.";
                    return redirect()->route('deeps.index',

```



```

        , $project->id)->with('success',
        $flash_message);
    }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }

    /**
     * Show the form for editing the specified
     * resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        $project = Project::findOrFail($id);

        if (Gate::denies('deep', $project)) {
            abort(403);
        }

        $boreholes = Project::find($project->
            id)->boreholes;
        $samples = Sample::join('boreholes', '
            samples.bh_id', '=', 'boreholes.
            id')
            ->where('boreholes.proj_id',
                $project->id)
            ->select('samples.id', 'samples.
                bh_id', 'samples.sample_num',
                'samples.sample_to', '
                samples.sample_num_of_blows')
            ->groupBy('samples.id')
            ->orderBy('samples.bh_id')
            ->orderBy('samples.id')
            ->get();
        $shallows = Shallow::where('proj_id',
            $project->id)->get();
        $deeps = Deep::where('proj_id',
            $project->id)->get();

        DeepController::isUpdated($project->id
            , sizeof($samples), sizeof($deeps
            ), sizeof($shallows));

        return view('deeps.edit', array('
            project' => $project, 'boreholes'
            => $boreholes, 'samples' =>
            $samples, 'shallows' => $shallows
            , 'deeps' => $deeps));
    }

    /**
     * Update the specified resource in
     * storage.
     *
     * @param \Illuminate\Http\Request
     * $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(UpdateDeepRequest
        $request, $id)
    {
        // Update information
        $project = Project::findOrFail($id);
        $j = $k = 0;
        for($i = 0; $i < $project->
            proj_num_of_boreholes; $i++) {
            while($j < $request->
                num_of_samples[$i]) {
                $deep = Deep::findOrFail(
                    $request->deep_id[$k]);

                $deep->update([
                    'deep_pile_size' =>
                        $request->
                            deep_pile_size[$i],
                    'deep_K' => $request->
                        deep_K[$i],
                    'deep_gamma' => $request->
                        deep_gamma[$i],
                    'deep_tan' => $request->
                        deep_tan[$i],
                    'deep_fs' => $request->
                        deep_fs[$i],
                    'deep_rho' => $request->
                        deep_rho[$i],
                    'deep_fy' => $request->
                        deep_fy[$i],
                    'deep_fc' => $request->
                        deep_fc[$i],
                    'deep_concrete_strength_kn'
                    => $request->
                        deep_concrete_strength_kn
                        [$i],
                    'deep_concrete_strength_ton'
                    => $request->
                        deep_concrete_strength_ton
                        [$i],
                    'deep_f' => $request->
                        deep_f[$k],
                    'deep_Asurface' =>
                        $request->
                            deep_Asurface[$k],
                    'deep_q' => $request->
                        deep_q[$k],
                    'deep_At' => $request->
                        deep_At[$k],
                    'deep_Qtotal' => $request
                        ->deep_Qtotal[$k],
                    'deep_Q_kn' => $request->
                        deep_Q_kn[$k],
                    'deep_Q_ton' => $request->
                        deep_Q_ton[$k],
                    'deep_is_chosen' => 0
                ]);
                $j++;
                $k++;
            }
            $j = 0;
        }

        if((sizeof($request->deep_is_chosen) >
            0) && ($request->deep_is_chosen
            [0] > -1)) {
            for($i = 0; $i < sizeof($request->
                deep_is_chosen); $i++) {
                for($j = 0; $j < sizeof(
                    $request->sample_id); $j
                    ++){
                    if($request->
                        deep_is_chosen[$i] ==
                        $request->sample_id[
                            $j]) {
                        $deep = Sample::find(
                            $request->
                                sample_id[$j])->
                            deep;
                        $deep->update(['
                            deep_is_chosen'
                            => 1]);
                    }
                }
            }
        }

        // Add the activity to activity table
        Activity::create([
            'proj_id' => $request->proj_id,
            'user_id' => Auth::user()->id,
            'activity_type' => "Data_for_SBC_
                for_deep_foundation_in_
                project_<i>$project->
                proj_name</i>-has_been_
                modified."
        ]);

        $flash_message = "Data_for_SBC_for_
            deep_foundation_in_project_
            $project->proj_name_has_been_
            successfully_modified.";
        return redirect()->route('deeps.index'
            , $project->id)->with('success',
            $flash_message);
    }

    /**
     * Remove the specified resource from
     * storage.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        $deeps = Project::findOrFail($id)->
            deeps;

        if (Gate::denies('deep', Project::find
            ($id))) {
            abort(403);
        }

        foreach($deeps as $deep) {
            $deep->delete();
        }
    }

```

```

return DeepController::
    recordDeleteActivity($id);
}

public function recordDeleteActivity(
    $proj-id)
{
    $project = Project::findOrFail(
        $proj-id);

    // Add the activity to activity table
    Activity::create([
        'proj-id' => $project->id,
        'user-id' => Auth::user()->id,
        'activity-type' => "Data_for_SBC_
        for_deep_foundation_in_
        project_<i>$project->
        proj-name</i>-has-been-
        deleted."
    ]);

    $flash-message = "Data_for_SBC_for_
    deep_foundation_in_project_
    $project->proj_name_has_been_
    successfully_deleted.";
    return redirect()->route('projects.
    show', $project->id)->with('
    success', $flash-message);
}

public function isUpdated($proj-id,
    $size-of-samples, $size-of-deeps,
    $size-of-shallows) {
    $is-updated = false;

    // Check if data is not yet created
    if($size-of-deeps == 0) {
        $route = route('deeps.create',
            $proj-id);
        $flash-message = "SBC_data_for_
        shallow_foundation_is_missing_
        .<a_href=$route_class='btn_
        btn-link_btn-sm'>Add_one_here_
        </a>.";
        \Session::flash('warning_e',
            $flash-message);
        return;
    }

    // Check if a new sample has been
    created
    else if($size-of-samples !=
        $size-of-deeps) {
        $is-updated = true;
    }

    // Check if a new instance of shallows
    has been created
    else if($size-of-deeps !=
        $size-of-shallows) {
        $is-updated = true;
    }

    if($is-updated) {
        $route = route('deeps.
        deleteAndCreate', $proj-id);
        $flash-message = "Samples_were_
        recently_modified._Please_
        update_the_data_by_<a_href=
        $route_class='btn_btn-link_
        btn-sm'>creating_a_new_one</a_
        >-to_avoid_errors.";
        \Session::flash('warning_e',
            $flash-message);
    }
    return;
}

public function deleteAndCreate($proj-id)
{
    $matchThese = ['proj-id' => $proj-id,
        'enr-id' => Auth::user()->id];
    $project = DB::table('
    project_engineers')->where(
        $matchThese)->get();

    if(sizeof($project) == 0) {
        return view('errors.403');
    }

    DeepController::destroy($proj-id);
    return DeepController::create($proj-id
    );
}

public function checkPrerequisites(
    $samples, $shallows, $project) {
    $areThereSamples = true;

```

```

    $areThereShallows = true;
    if (sizeof($samples) == 0) {
        $areThereSamples = false;
    }
    if (sizeof($shallows) == 0) {
        $areThereShallows = false;
    }

    if(!$areThereSamples) {
        $flash-message = "Boreholes/
        samples_are_missing._Please_
        add_them_first_in_order_to_
        proceed_smoothly.";
        \Session::flash('warning',
            $flash-message);
    }
    if(!$areThereShallows) {
        $route = route('deeps.create',
            $project->id);
        $flash-message = "SBC_data_for_
        shallow_foundation_is_missing_
        .Please_<a_href=$route_class_
        ='btn_btn-link_btn-sm'>create_
        one_first_</a>-in_order_to_
        proceed_smoothly.";
        \Session::flash('warning_e',
            $flash-message);
    }
}
}

```

Listing 9: DrilledShaftController.php

```

<?php
namespace App\Http\Controllers;

use Auth;
use DB;
use Input;
use Gate;
use App\Activity;
use App\DrilledShaft;
use App\Project;
use App\Sample;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests\DrilledShaft\
    StoreDrilledShaftRequest;
use App\Http\Requests\DrilledShaft\
    UpdateDrilledShaftRequest;

class DrilledShaftController extends
    Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index($id)
    {
        $project = Project::findOrFail($id);

        if (Gate::denies('index-drilled-shaft',
            $project)) {
            abort(403);
        }

        $boreholes = Project::find($project->
            id)->boreholes;
        $samples = Sample::join('boreholes', '
            samples.bh_id', '=', 'boreholes.
            id')
            ->where('boreholes.proj_id',
                $project->id)
            ->select('samples.id', 'samples.
                bh_id', 'samples.sample_num',
                'samples.sample_to', '
                samples.sample_num_of.blows')
            ->groupBy('samples.id')
            ->orderBy('samples.bh_id')
            ->orderBy('samples.id')
            ->get();
        $drilled_shafts = DrilledShaft::where(
            'proj-id', $id)->get();

        DrilledShaftController::isUpdated(
            $project->id, sizeof($samples),
            sizeof($drilled_shafts));

        return view('drilledshafts.index',
            array('project' => $project, '
            boreholes' => $boreholes, '
            samples' => $samples, '
            drilled_shafts' =>

```

```

        $drilled_shafts));
    }

    /**
     * Show the form for creating a new
     * resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create($id)
    {
        $project = Project::findOrFail($id);

        if (Gate::denies('drilled-shaft',
            $project)) {
            abort(403);
        }

        $drilled_shafts = $project->
            drilledShafts;

        if (sizeof($drilled_shafts) == 0) {
            $boreholes = Project::findOrFail(
                $project->id)->boreholes;
            $samples = Sample::join('boreholes',
                'samples.bh_id', '=', 'boreholes.id')
                ->where('boreholes.proj_id',
                    $project->id)
                ->select('samples.id', 'samples.bh_id', 'samples.sample_num', 'samples.sample_to', 'samples.sample_num_of_blows')
                ->get();

            DrilledShaftController::
                checkPrerequisites($boreholes,
                    $samples);

            return view('drilledshafts.create',
                array('project' => $project,
                    'boreholes' => $boreholes,
                    'samples' => $samples));
        }
        else {
            $flash_message = "SBC_data_for_drilled_shaft_foundation_in_project_{$project->proj_name}_already_exists_You_cannot_create_another_instance_of_the_data.";
            return redirect()->route('projects.show', $project->id)->with('error', $flash_message);
        }
    }

    /**
     * Store a newly created resource in
     * storage.
     *
     * @param \Illuminate\Http\Request
     * $request
     * @return \Illuminate\Http\Response
     */
    public function store(
        StoreDrilledShaftRequest $request)
    {
        $project = Project::findOrFail(
            $request->proj_id);
        $j = $k = 0;

        for($i = 0; $i < $project->
            proj_num_of_boreholes; $i++) {
            while($j < $request->
                num_of_samples[$i]) {
                DrilledShaft::create([
                    'proj_id' => $project->id,
                    'bh_id' => $request->bh_id[$i],
                    'sample_id' => $request->
                        sample_id[$k],
                    'ds_pile_size' => $request->
                        ds_pile_size[$i],
                    'ds_Nc' => $request->ds_Nc[$i],
                    'ds_f_coeff' => $request->
                        ds_f_coeff[$i],
                    'ds_fc' => $request->ds_fc[$i],
                    'ds_fy' => $request->ds_fy[$i],
                    'ds_steel_bar_diameter' =>
                        $request->
                            ds_steel_bar_diameter[$i],
                    'ds_fs' => $request->ds_fs
                ]);
                $j++;
                $k++;
            }
            $j = 0;
        }

        if((sizeof($request->ds_is_chosen) >
            0) && ($request->ds_is_chosen[0] >
                -1)) {
            for($i = 0; $i < sizeof($request->
                ds_is_chosen); $i++) {
                for($j = 0; $j < sizeof(
                    $request->sample_id); $j
                    ++){
                    if($request->ds_is_chosen[
                        $i] == $request->
                            sample_id[$j]) {
                        $drilled_shaft =
                            Sample::find(
                                $request->
                                    sample_id[$j])->
                                drilledShaft;
                        $drilled_shaft->update(
                            (['ds_is_chosen'
                                => 1]));
                    }
                }
            }
        }

        // Add the activity to activity table
        Activity::create([
            'proj_id' => $project->id,
            'user_id' => Auth::user()->id,
            'activity_type' => "Data_for_SBC_for_drilled_shaft_foundation_in_project_<i>{$project->proj_name}</i>-has_been_created."
        ]);

        $flash_message = "Data_for_SBC_for_drilled_shaft_foundation_in_project_{$project->proj_name}_has_been_successfully_saved.";
        return redirect()->route('drilledshafts.index', $project->id)->with('success', $flash_message);
    }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }

    /**
     * Show the form for editing the specified
     * resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)

```

```

{
    $project = Project::findOrFail($id);

    if (Gate::denies('drilled-shaft',
        $project)) {
        abort(403);
    }

    $boreholes = Project::findOrFail(
        $project->id)->boreholes;
    $samples = Sample::join('boreholes', '
        samples.bh_id', '=', 'boreholes.
        id')
        ->where('boreholes.proj_id',
            $project->id)
        ->select('samples.id', 'samples.
            bh_id', 'samples.sample_num',
            'samples.sample_to', '
            samples.sample_num_of_blows')
        ->groupBy('samples.id')
        ->orderBy('samples.bh_id')
        ->orderBy('samples.id')
        ->get();
    $drilled_shafts = DrilledShaft::where(
        'proj_id', $id)->get();

    DrilledShaftController::isUpdated(
        $project->id, sizeof($samples),
        sizeof($drilled_shafts));

    return view('drilledshafts.edit',
        array('project' => $project, '
        boreholes' => $boreholes, '
        samples' => $samples, '
        drilled_shafts' =>
        $drilled_shafts));
}

/**
 * Update the specified resource in
 * storage.
 *
 * @param \Illuminate\Http\Request
 * $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(
    UpdateDrilledShaftRequest $request,
    $id)
{
    // Update information
    $project = Project::findOrFail($id);
    $j = $k = 0;
    for($i = 0; $i < $project->
        proj_num_of_boreholes; $i++) {
        while($j < $request->
            num_of_samples[$i]) {
            $drilled_shaft = DrilledShaft
                ::findOrFail($request->
                    drilled_id[$k]);

            $drilled_shaft->update([
                'ds_pile_size' => $request
                    ->ds_pile_size[$i],
                'ds_Nc' => $request->ds_Nc
                    [$i],
                'ds_f.coeff' => $request->
                    ds_f.coeff[$i],
                'ds_fc' => $request->ds_fc
                    [$i],
                'ds_fy' => $request->ds_fy
                    [$i],
                'ds_steel_bar_diameter' =>
                    $request->
                    ds_steel_bar_diameter
                    [$i],
                'ds_fs' => $request->ds_fs
                    [$i],
                'ds_tension_cap' =>
                    $request->
                    ds_tension_cap[$i],
                'ds_concrete_strength_kn'
                    => $request->
                    ds_concrete_strength_kn
                    [$i],
                'ds_concrete_strength_ton'
                    => $request->
                    ds_concrete_strength_ton
                    [$i],
                'ds_soil' => $request->
                    ds_soil[$k],
                'ds_c' => $request->ds_c[
                    $k],
                'ds_Qbase' => $request->
                    ds_Qbase[$k],
                'ds_f' => $request->ds_f[
                    $k],
                'ds_Qfriction' => $request
                    ->ds_Qfriction[$k],
                'ds_Qtotal_kn' => $request
                    ->ds_Qtotal_kn[$k],
                'ds_Qtotal_ton' =>
                    $request->
                    ds_Qtotal_ton[$k],
                'ds_is_chosen' => 0
            ]);
            $j++;
            $k++;
        }
        $j = 0;
    }

    if((sizeof($request->ds_is_chosen) >
        0) && ($request->ds_is_chosen[0]
        > -1)) {
        for($i = 0; $i < sizeof($request->
            ds_is_chosen); $i++) {
            for($j = 0; $j < sizeof(
                $request->sample_id); $j
                ++){
                if($request->ds_is_chosen[
                    $i] == $request->
                    sample_id[$j]) {
                    $drilled_shaft =
                        Sample::find(
                            $request->
                            sample_id[$j])->
                            drilledShaft;
                    $drilled_shaft->update(
                        ([ 'ds_is_chosen'
                            => 1]));
                }
            }
        }
    }

    // Add the activity to activity table
    Activity::create([
        'proj_id' => $request->proj_id,
        'user_id' => Auth::user()->id,
        'activity_type' => "Data_for_SBC_
        for_drilled_shaft_foundation_
        in_project_<i>$project->
        proj_name</i>_has_been_
        modified."
    ]);

    $flash_message = "Data_for_SBC_for_
        drilled_shaft_foundation_in_
        project_<math>proj\_name\_has\_
        been\_successfully\_modified.</math>";
    return redirect()->route('
        drilledshafts.index', $project->
        id)->with('success',
        $flash_message);
}

/**
 * Remove the specified resource from
 * storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    $drilled_shafts = Project::findOrFail(
        $id)->drilledShafts;

    if (Gate::denies('drilled-shaft',
        Project::find($id))) {
        abort(403);
    }

    foreach($drilled_shafts as
        $drilled_shaft) {
        $drilled_shaft->delete();
    }

    return DrilledShaftController::
        recordDeleteActivity($id);
}

public function recordDeleteActivity(
    $proj_id)
{
    $project = Project::findOrFail(
        $proj_id);

    // Add the activity to activity table
    Activity::create([
        'proj_id' => $project->id,
        'user_id' => Auth::user()->id,
        'activity_type' => "Data_for_SBC_
        for_drilled_shaft_foundation_
        in_project_<i>$project->

```



```

use App\Activity;
use App\Limit;
use App\Project;
use App\Sample;
use App\Subreport;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class LimitController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index($id)
    {
        $project = Project::findOrFail($id);

        if (Gate::denies('index-limit',
            $project)) {
            abort(403);
        }

        $boreholes = Project::findOrFail(
            $project->id)->boreholes;
        $samples = Sample::join('boreholes', '
            samples.bh_id', '=', 'boreholes.
            id')
            ->where('boreholes.proj_id',
                $project->id)
            ->select('samples.id', 'samples.
                bh_id', 'samples.sample_num')
            ->get();

        $matchThese = ['proj_id' => $project->
            id, 'sb_key' => 1];
        $subreport = Subreport::where(
            $matchThese)->first();

        $limits = Limit::where('subreport_id',
            $subreport->id)->get();

        return view('limits.index', array('
            project' => $project, 'boreholes'
            => $boreholes, 'samples' =>
            $samples, 'subreport' =>
            $subreport, 'limits' => $limits))
            ;
    }

    /**
     * Show the form for creating a new
     * resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create($id)
    {
        $project = Project::findOrFail($id);

        if (Gate::denies('limit', $project)) {
            abort(403);
        }

        // Check if subreport is already in
        // the database
        $matchThese = ['proj_id' => $project->
            id, 'sb_key' => 1];
        $match = DB::table('subreports')->
            where($matchThese)->first();

        if(is_null($match)) {
            $boreholes = DB::table('boreholes'
            )
            ->join('projects', 'boreholes.
            proj_id', '=', 'projects.
            id')
            ->where('boreholes.proj_id',
                $id)
            ->select('boreholes.id')
            ->get();

            if (sizeof($boreholes) > 0) { //
                // If there are boreholes in
                // project
                $i = 0;
                foreach($boreholes as
                    $borehole) {
                    $samples = DB::table('
                    samples')
                    ->join('boreholes', '
                    samples.bh_id', '
                    =', 'boreholes.id'
                    )
                    ->where('samples.bh_id
                    ', $borehole->id)
                    ->select('samples.id')
                    ->get();

                    $boreholes[$i] = $borehole
                    ->id;
                    $j = 0;

                    if(sizeof($samples) > 0) {
                        // If there are
                        // samples in current
                        // borehole
                        foreach($samples as
                            $sample) {
                            $counts[$i][$j] =
                                $sample->id;
                            $j++;
                        }
                    }
                    $i++;
                }
            }
            else {
                $counts = 0;
            }

            if(!isset($counts)) {
                $counts = 0;
            }

            LimitController::
                checkPrerequisites($counts);

            return view('limits.create', array
                ('project' => $project, '
                boreholes' => $boreholes, '
                counts' => $counts));
        }
        else {
            $flash_message = "Liquid_and_
            Plastic_Limits_Test_Report_in
            _project_-$project->proj_name_
            already_exists_..You_cannot_
            create_another_instance_of_
            the_subreport.";
            return redirect()->route('projects
            .show', $project->id)->with('
            error', $flash_message);
        }
    }

    /**
     * Store a newly created resource in
     * storage.
     *
     * @param \Illuminate\Http\Request
     * $request
     *
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        // Validate the request
        $this->validate($request, [
            'proj_id' => 'required',
            'sample_id' => 'required|array',
            'lim_test' => 'required|array',
            'lim_water_content' => 'required|
            array',
            'lim_num_of_blows' => 'required|
            array'
        ]);

        $project = Project::findOrFail(
            $request->proj_id);

        // Check if subreport is already in
        // the database
        $matchThese = ['proj_id' => $project->
            id, 'sb_key' => 1];
        $match = DB::table('subreports')->
            where($matchThese)->first();

        if(is_null($match)) {
            // Store data to subreports table
            $input_sb['sb_key'] = 1;
            $input_sb['proj_id'] = $project->
                id;
            $subreport = Subreport::create(
                $input_sb);

            // Store data to limits table
            $j = 0;
            $lim_run_num = 0;
            $prev_test = 0;
            for($i = 0; $i < sizeof($request->
                sample_id); $i++) {

                ($prev_test == $request->
                    lim_test[$i]) ?
                    $lim_run_num++ :
                    $lim_run_num = 1;
                if ($request->lim_test[$i] ==
    
```

```

        1) {
            $lim_num_of_blows = -1;
        }
    else {
        $lim_num_of_blows =
            $request->
                lim_num_of_blows[$j];
        $j++;
    }

    Limit::create([
        'sample_id' => $request->
            sample_id[$i],
        'subreport_id' =>
            $subreport->id,
        'lim_run_num' =>
            $lim_run_num,
        'lim_test' => $request->
            lim_test[$i],
        'lim_num_of_blows' =>
            $lim_num_of_blows,
        'lim_water_content' =>
            $request->
                lim_water_content[$i]
    ]);

    $prev_test = $request->
        lim_test[$i];
}

// Add the activity to activity
// table
Activity::create([
    'proj_id' => $project->id,
    'user_id' => Auth::user()->id,
    'activity_type' => "Data_for_
the_Liquids_and_Plastic_
Limits_Test_Report_in_
project_<i>$project->
proj_name</i>_has_been_
created."
]);

$flash_message = "Data_for_the_
Liquids_and_Plastic_Limits_
Test_Report_in_project_
$project->proj_name_has_been_
successfully_created.";
return redirect()->route('
subreports.showLPTR',
$subreport->id)->with('
success', $flash_message);
}
else {
    $flash_message = "Liquids_and_
Plastic_Limits_Test_Report_in_
project_<i>$project->proj_name_
already_exists_You_cannot_
create_another_instance_of_
the_subreport.";
    return redirect()->route('limits.
create', $project->id)->with(
        'error', $flash_message);
}
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    //
}

/**
 * Show the form for editing the specified
 * resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    $project = Project::findOrFail($id);

    if (Gate::denies('limit', $project)) {
        abort(403);
    }

    $boreholes = Project::findOrFail(
        $project->id)->boreholes;
    $samples = Sample::join('boreholes', '
samples.bh.id', '=', 'boreholes.
id')
        ->where('boreholes.proj_id',
            $project->id)
        ->select('samples.id', 'samples.
bh.id', 'samples.sample_num')
        ->get();

    $matchThese = ['proj_id' => $project->
        id, 'sb_key' => 1];
    $subreport = Subreport::where(
        $matchThese)->first();

    $limits = Limit::where('subreport_id',
        $subreport->id)->get();

    return view('limits.edit', array('
project' => $project, 'boreholes'
=> $boreholes, 'samples' =>
        $samples, 'subreport' =>
        $subreport, 'limits' => $limits))
        ;
}

/**
 * Update the specified resource in
 * storage.
 *
 * @param \Illuminate\Http\Request
 * $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request,
    $id)
{
    // Validate the request
    $this->validate($request, [
        'limit_id' => 'required|array',
        'sample_id' => 'required|array',
        'lim_test' => 'required|array',
        'lim_water_content' => 'required|
array',
        'lim_num_of_blows' => 'required|
array',
        'proj_id' => 'required',
    ]);

    // Update data in data_points table
    for($i = 0; $i < sizeof($request->
        limit_id); $i++) {
        $limit = Limit::findOrFail(
            $request->limit_id[$i]);

        $limit->update([
            'lim_water_content' =>
                $request->
                    lim_water_content[$i],
            'lim_num_of_blows' => $request
                ->lim_num_of_blows[$i]
        ]);
    }

    $project = Project::findOrFail(
        $request->proj_id);
    // Add the activity to activity table
    Activity::create([
        'proj_id' => $project->id,
        'user_id' => Auth::user()->id,
        'activity_type' => "Data_for_the_
Liquid_and_Plastic_Limits_
Test_Report_in_project_<i>
$project->proj_name</i>_has_
been_modified."
    ]);

    $flash_message = "Data_for_the_Liquid_
and_Plastic_Limits_Test_Report_in_
project_<i>$project->proj_name_
has_been_successfully_modified.";
    return redirect()->route('subreports.
showLPTR', $id)->with('success',
        $flash_message);
}

/**
 * Remove the specified resource from
 * storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    $subreport = Subreport::findOrFail($id
    );

    if (Gate::denies('limit', $subreport->
        project)) {
        abort(403);
    }
}

```

```

$limits = Subreport::find($subreport->
id)->limits;

$previous_sample_id = -1;
foreach($limits as $limit) {
    $sample = Limit::find($limit->id)
->samples;
    if($previous_sample_id != $sample
->id) {
        $file_path = "images/
subreports/LPTR/sample-
$sample->id.png";
        if(file_exists($file_path)) {
            if (!unlink($file_path)) {
                $status = "warning";
            }
            else {
                $status = "success";
            }
        }
    }
    $previous_sample_id = $sample->id;
}

$limit->delete();

$proj_id = $subreport->proj_id;
$subreport->delete();

return LimitController::
recordDeleteActivity($proj_id);
}

public function recordDeleteActivity(
    $proj_id)
{
    $project = Project::findOrFail(
        $proj_id);

    // Add the activity to activity table
    Activity::create([
        'proj_id' => $project->id,
        'user_id' => Auth::user()->id,
        'activity_type' => "Liquid_and_
Plastic_Limits_Test_Report_in_
project_<i>$project->
proj_name</i>_has_been_
deleted."
    ]);

    $flash_message = "Liquid_and_Plastic_
Limits_Test_Report_in_
project_
$project->proj_name_has_been_
successfully_deleted.";
    return redirect()->route('projects.
show', $project->id)->with('
success', $flash_message);
}

public function checkPrerequisites($counts
) {
    $arePrerequisitesOkay = true;
    if ($counts == 0) {
        $arePrerequisitesOkay = false;
    }

    if (!$arePrerequisitesOkay) {
        $flash_message = "Some_
prerequisites_are_missing_(
boreholes_or_samples)_Please_
add_them_first_in_order_to_
proceed_smoothly.";
        \Session::flash('warning',
            $flash_message);
    }
}
}
}

use App\Subreport;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests\Project\
StoreProjectRequest;
use App\Http\Requests\Project\
UpdateProjectRequest;

class ProjectController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $projects = Project::all();

        if (Gate::denies('index-project')) {
            abort(403);
        }

        return view('projects.index', array('
projects' => $projects));
    }

    /**
     * Show the form for creating a new
     resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        //
    }

    /**
     * Store a newly created resource in
     storage.
     *
     * @param \Illuminate\Http\Request
     $request
     * @return \Illuminate\Http\Response
     */
    public function store(StoreProjectRequest
        $request)
    {
        $project = Project::create($request->
            all());

        // Add the admin to project_engineers
        table
        DB::table('project_engineers')->insert
            ([['proj_id' => $project->id, '
            engr_id' => 1]]);

        // Add the activity to activity table
        Activity::create([
            'proj_id' => $project->id,
            'user_id' => Auth::user()->id,
            'activity_type' => "Project_<i>
            $project->proj_name</i>_has_
            been_created."
        ]);

        $flash_message = "Project_-$project->
            proj_name_has_been_successfully_
            created.";
        return redirect()->route('projects.
            show', $project->id)->with('
            success', $flash_message);
    }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        $project = Project::findOrFail($id);

        /* if (Gate::denies('show-project',
            $project)) {
            abort(403);
        } */

        $boreholes = $project->boreholes;
        $sections = $project->sections;
        $subreports = $project->subreports;
        $shallows = $project->shallows;
        $drilledshafts = $project->
            drilledshafts;
        $deeps = $project->deeps;

        return view('projects.show', array('

```

Listing 12: ProjectController.php

```
<?php
```

```

namespace App\Http\Controllers;

use Auth;
use DB;
use File;
use Gate;
use Session;
use Validator;
use Helper;
use TemplateHelper;
use App\Activity;
use App\Borehole;
use App\Project;
use App\Sample;
use App\Section;

```



```

        project' => $project, 'boreholes'
        => $boreholes, 'sections' =>
        $sections, 'subreports' =>
        $subreports, 'shallows' =>
        $shallows, 'drilled_shafts' =>
        $drilled_shafts, 'deeps' =>
        $deeps));
    }

    /**
     * Show the form for editing the specified
     * resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        $project = Project::findOrFail($id);

        if (Gate::denies('edit-project',
            $project)) {
            abort(403);
        }

        return view('projects.edit', array('
            project' => $project));
    }

    /**
     * Update the specified resource in
     * storage.
     *
     * @param \Illuminate\Http\Request
     * $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(
        UpdateProjectRequest $request, $id)
    {
        $project = Project::findOrFail($id);

        // Update project information
        $project->update(['proj_name' =>
            $request->proj_name,
            'proj_location' => $request->
            proj_location,
            'proj_client' => $request->
            proj_client,
            'proj_description' => $request->
            proj_description,
            'proj_start_date' => $request->
            proj_start_date,
            'proj_end_date' => $request->
            proj_end_date,
            'proj_date_signed' => $request->
            proj_date_signed
        ]);

        if (Auth::user()->isAdmin()) {
            if (isset($request->project_members
                )) {
                // Add engineer to project
                for ($i = 0; $i < sizeof(
                    $request->project_members
                ); $i++) {
                    $matchThese = ['proj_id'
                        => $project->id, '
                    engr_id' => $request
                        ->project_members[$i
                    ]];
                    $match = DB::table('
                        project_engineers')->
                        where($matchThese)->
                        first();

                    if (is_null($match)) {
                        DB::table('
                            project_engineers
                        ')->insert(['
                            proj_id' =>
                            $project->id, '
                            engr_id' =>
                            $request->
                            project_members[
                                $i
                            ]]);

                        $user_name = Helper::
                            getUser_name(
                                $request->
                                project_members[
                                    $i
                                ]);

                        // Add the activity to
                        activity table
                        Activity::create(['
                            'proj_id' =>
                            $project->id,
                            'user_id' => Auth
                                ::user()->id,
                            'activity_type' => "Project-<i>
                                has_been_
                                removed_from_
                                project_<i>
                                $project->
                                proj_name</i>
                                >."
                        ]);
                    }
                }
            }
        }

        // Add the activity to activity table
        Activity::create(['
            'proj_id' => $project->id,
            'user_id' => Auth::user()->id,
            'activity_type' => "Project-<i>
                $project->proj_name</i>_has_
                been_modified."
        ]);

        $flash_message = "Project_<i>
            $project->
            proj_name_has_been_successfully_
            modified.";

        return redirect()->route('projects.
            show', $project->id)->with('
            success', $flash_message);
    }

    /**
     * Remove the specified resource from
     * storage.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        if (Gate::denies('destroy-project')) {
            abort(403);
        }

        $project = Project::findOrFail($id);
        $proj_name = $project->proj_name;
        $project->delete();

        return ProjectController::
            ::user()->id,
            'activity_type' => "
            User_<i>
            $user_name</i>
            >_has_been_
            added_to_
            project_<i>
            $project->
            proj_name</i>
            >."
        ]);
    }

    // Remove engineer from
    project
    $engineers = Helper::
        getEngineers($project->id
    );
    foreach ($engineers as
        $engineer) {
        if (!in_array($engineer->id
            , $request->
            project_members)) {
            if ($engineer->id != 1)
            {
                $matchThese = ['
                    proj_id' =>
                    $project->id, '
                    engr_id' =>
                    $engineer->id];
                DB::table('
                    project_engineers
                ')->where(
                    $matchThese)->
                    delete();

                // Add the activity to
                activity table
                Activity::create(['
                    'proj_id' =>
                    $project->id,
                    'user_id' => Auth
                        ::user()->id,
                    'activity_type' => "
                    User_<i>
                    $engineer->
                    first_name_
                    $engineer->
                    lastname</i>_
                    has_been_
                    removed_from_
                    project_<i>
                    $project->
                    proj_name</i>
                    >."
                ]);
            }
        }
    }

    // Add the activity to activity table
    Activity::create(['
        'proj_id' => $project->id,
        'user_id' => Auth::user()->id,
        'activity_type' => "Project-<i>
            $project->proj_name</i>_has_
            been_modified."
    ]);

    $flash_message = "Project_<i>
        $project->
        proj_name_has_been_successfully_
        modified.";

    return redirect()->route('projects.
        show', $project->id)->with('
        success', $flash_message);
}

/**
 * Remove the specified resource from
 * storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    if (Gate::denies('destroy-project')) {
        abort(403);
    }

    $project = Project::findOrFail($id);
    $proj_name = $project->proj_name;
    $project->delete();

    return ProjectController::

```

```

        recordDeleteActivity($proj_name);
    }

    public function recordDeleteActivity(
        $proj_name)
    {
        // Add the activity to activity table
        Activity::create([
            'proj_id' => null,
            'user_id' => Auth::user()->id,
            'activity_type' => "Project<i>_</i>_has_been_deleted."
        ]);

        $flash_message = "Project_{$proj_name}_has_been_successfully_deleted.";
        return redirect()->route('projects.index')->with('success', $flash_message);
    }

    public function archive($id)
    {
        $project = Project::findOrFail($id);

        if (Gate::denies('archive-project')) {
            abort(403);
        }

        if ($project->isTagged()) {
            $project->update([
                'proj_status' => 2
            ]);

            // Add the activity to activity table
            Activity::create([
                'proj_id' => $project->id,
                'user_id' => Auth::user()->id,
                'activity_type' => "Project<i>_</i>_has_been_archived."
            ]);

            $flash_message = "Project_{$project->proj_name}_has_been_successfully_archived.";
            $status = "success";
        } else {
            $flash_message = "Project_{$project->proj_name}_cannot_be_archived_because_it_is_not_yet_tagged.";
            $status = "error";
        }

        return redirect()->route('projects.index')->with($status, $flash_message);
    }

    public function tag($id)
    {
        $project = Project::findOrFail($id);

        if (Gate::denies('tag-project', $project)) {
            abort(403);
        }

        if ($project->isActive()) {
            $project->update([
                'proj_status' => 1
            ]);

            // Add the activity to activity table
            Activity::create([
                'proj_id' => $project->id,
                'user_id' => Auth::user()->id,
                'activity_type' => "Project<i>_</i>_has_been_tagged_for_validation."
            ]);

            $flash_message = "Project_{$project->proj_name}_has_been_successfully_tagged_for_validation.";
            $status = "success";
        } else {
            $flash_message = "Project_{$project->proj_name}_cannot_be_tagged_because_it_is_not_active.";
            $status = "error";
        }
    }

```

```

        return redirect()->route('projects.index')->with($status, $flash_message);
    }

    /**
     * Display a listing of the user's resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function viewActive($id)
    {
        if (Gate::denies('view-active-project', $id)) {
            abort(403);
        }

        $projects = DB::select('SELECT _pe.proj_id, _p.proj_name, _p.proj_image FROM projects _p, project_engineers _pe WHERE _pe.id = _pe.proj_id AND _pe.engr_id = ?_ AND _p.proj_status = 0', [$id]);

        return view('projects.viewActive', array('projects' => $projects));
    }

    public function viewArchived()
    {
        if (Gate::denies('view-archived-project')) {
            abort(403);
        }

        $projects = Project::where('proj_status', 2)->get();

        return view('projects.viewArchived', array('projects' => $projects));
    }

    public function viewCompiledReport($id)
    {
        $project = Project::findOrFail($id);
        if (Gate::denies('view-compiled-report', $project)) {
            abort(403);
        }

        ob_start();
        list($boreholes, $samples, $sections, $section_bodies, $subreports, $liquid_limits, $plastic_limits) = TemplateHelper::getVariables($id);

        $content = view('includes.template', [
            'project' => $project,
            'boreholes' => $boreholes,
            'samples' => $samples,
            'sections' => $sections,
            'section_bodies' => $section_bodies,
            'subreports' => $subreports,
            'liquid_limits' => $liquid_limits,
            'plastic_limits' => $plastic_limits,
        ])->render();

        $html2pdf_path = base_path() . '\vendor\spipu\html2pdf\html2pdf.class.php';
        File::requireOnce($html2pdf_path);

        try {
            $html2pdf = new \HTML2PDF('P', 'Legal', 'en', true, 'UTF-8', array(25.4, 20.4, 25.4, 20.4));

            $html2pdf->pdf->SetAuthor('Earl_Area_Abad');
            $html2pdf->pdf->SetTitle($project->proj_name);

            $html2pdf->WriteHTML($content);

            $filename = 'Report-' . $project->id . '.pdf';
            $output_path = base_path() . '\public\reports\\' . $filename;

            $pdf = $html2pdf->Output('', 'S');
            $html2pdf->Output($output_path, 'F');
            return response($pdf)
                ->header('Content-Type', '');
        }
    }

```

```

        application/pdf')
        ->header('Content-Length',
            strlen($pdf))
        ->header('Content-Disposition',
            'inline;_filename="'.
                $filename.'"');
    }
    catch (HTML2PDF_exception $e) {
        echo $e;
        exit;
    }
}
}
}

```

Listing 13: SampleController.php

<?php

```

namespace App\Http\Controllers;

use Auth;
use DB;
use Gate;
use App\Activity;
use App\Sample;
use App\Borehole;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests\Sample\
    StoreSampleRequest;
use App\Http\Requests\Sample\
    UpdateSampleRequest;

class SampleController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        //
    }

    /**
     * Show the form for creating a new
     * resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create($id)
    {
        $borehole = Borehole::findOrFail($id);

        if (Gate::denies('sample', $borehole))
        {
            abort(403);
        }

        $project = $borehole->project;
        $uscs = DB::table('uscs')->orderBy('id')->get();
        $previous_sample = Borehole::find(
            $borehole->id->samples->sortBy('
                sample_num')->last());

        return view('samples.create', array('
            borehole' => $borehole, 'project'
            => $project, 'uscs' => $uscs, '
            previous_sample' =>
            $previous_sample));
    }

    /**
     * Store a newly created resource in
     * storage.
     *
     * @param \Illuminate\Http\Request
     * $request
     * @return \Illuminate\Http\Response
     */
    public function store(StoreSampleRequest
        $request)
    {
        $previous_sample = Borehole::find(
            $request->bh_id->samples->sortBy(
                'sample_num')->last());
        if ($previous_sample != null) {
            $from = $previous_sample->
                sample_to;
            $to = $from + $previous_sample->
                sample_run_length;
            $run_length_default =
                $previous_sample->
                sample_run_length;
        }
    }
}

```

```

else {
    $from = $to = "";
    $run_length_default = 0.30;
}

// Create sample
$sample = Sample::create([
    'bh_id' => $request->bh_id,
    'uscs_id' => $request->uscs_id,
    'sample_num' => $request->
        sample_num,
    'sample_from' => (empty(trim(
        $request->sample_from))) ?
        $from : $request->sample_from
    ),
    'sample_to' => (empty(trim(
        $request->sample_to))) ? $to
        : $request->sample_to,
    'sample_contract_num' => $request
        ->sample_contract_num,
    'sample_job_num' => $request->
        sample_job_num,
    'sample_color' => $request->
        sample_color,
    'sample_run_length' => (empty(trim(
        $request->sample_run_length))
        ) ? $run_length_default :
        $request->sample_run_length,
    'sample_sample_length' => $request
        ->sample_sample_length,
    'sample_rec_percent' => $request->
        sample_rec_percent,
    'sample_nmc_percent' => $request->
        sample_nmc_percent,
    'sample_num_of_blows' => $request
        ->sample_num_of_blows,
    'sample_liquid_limit' => $request
        ->sample_liquid_limit,
    'sample_plastic_limit' => $request
        ->sample_plastic_limit,
    'sample_plasticity_index' =>
        $request->
        sample_plasticity_index,
    'sample_plasticity_remark' =>
        $request->
        sample_plasticity_remark,
    'sample_tester' => $request->
        sample_tester,
    'sample_type_of_sampling' =>
        $request->
        sample_type_of_sampling,
    'sample_test_date' => $request->
        sample_test_date
]);

// Increment the number of boreholes
in project table
DB::table('boreholes')->whereId(
    $request->bh_id->increment('
    bh_num_of_samples');

$project = Borehole::findOrFail(
    $request->bh_id->project;
$borehole = Borehole::findOrFail(
    $request->bh_id);

// Add the activity to activity table
Activity::create([
    'proj_id' => $project->id,
    'user_id' => Auth::user()->id,
    'activity_type' => "Sample<i>
        $sample->sample_num</i>_in_
        borehole<i>$borehole->bh_num
        </i>,_project<i>$project->
        proj_name</i>-has_been_
        created."
]);

$flash_message = "Sample_{$sample->
    sample_num}_in_borehole_{$borehole
    ->bh_num},_project_{$project->
    proj_name}_has_been_created.";
return redirect()->route('boreholes.
    show', $borehole->id->with('
    success', $flash_message);
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    //
}

```

```

* Show the form for editing the specified
  resource.
*
* @param int $id
* @return \Illuminate\Http\Response
*/
public function edit($id)
{
    $sample = Sample::findOrFail($id);
    $borehole = Sample::find($sample->id)
        ->borehole;

    if (Gate::denies('sample', $borehole))
    {
        abort(403);
    }

    $project = Borehole::find($borehole->
        id)->project;
    $uscs = DB::table('uscs')->orderBy('id')
        ->get();

    return view('samples.edit', array('
        sample' => $sample, 'borehole' =>
        $borehole, 'project' => $project
        , 'uscs' => $uscs));
}

/**
 * Update the specified resource in
  storage.
 *
 * @param \Illuminate\Http\Request
  $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(UpdateSampleRequest
  $request, $id)
{
    $sample = Sample::findOrFail($id);
    $borehole = Borehole::findOrFail(
        $request->bh_id);
    $project = Borehole::findOrFail(
        $request->bh_id)->project;

    // Update sample information
    $sample->update([
        'uscs_id' => $request->uscs_id,
        'sample_num' => $request->
            sample_num,
        'sample_from' => $request->
            sample_from,
        'sample_to' => $request->sample_to
            ,
        'sample_contract_num' => $request
            ->sample_contract_num,
        'sample_job_num' => $request->
            sample_job_num,
        'sample_color' => $request->
            sample_color,
        'sample_run_length' => (empty(trim
            ($request->sample_run_length)
            )) ? 0.30 : $request->
            sample_run_length,
        'sample_sample_length' => $request
            ->sample_sample_length,
        'sample_rec_percent' => $request->
            sample_rec_percent,
        'sample_nmc_percent' => $request->
            sample_nmc_percent,
        'sample_num_of_blows' => $request
            ->sample_num_of_blows,
        'sample_liquid_limit' => $request
            ->sample_liquid_limit,
        'sample_plastic_limit' => $request
            ->sample_plastic_limit,
        'sample_plasticity_index' =>
            $request->
            sample_plasticity_index,
        'sample_plasticity_remark' =>
            $request->
            sample_plasticity_remark,
        'sample_tester' => $request->
            sample_tester,
        'sample_type_of_sampling' =>
            $request->
            sample_type_of_sampling,
        'sample_test_date' => $request->
            sample_test_date
    ]);

    // Add the activity to activity table
    Activity::create([
        'proj_id' => $project->id,
        'user_id' => Auth::user()->id,
        'activity_type' => "Sample<i>
        $sample->sample_num</i>-in-
        borehole<i>$borehole->bh_num

```

Listing 14: SectionController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```

use Auth;
use DB;
use Input;
use Helper;
use Gate;
use App\Activity;
use App\Project;
use App\Section;
use App\SectionBody;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests\Section\
    StoreSectionRequest;
use App\Http\Requests\Section\
    UpdateSectionRequest;

class SectionController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        //
    }

    /**
     * Show the form for creating a new
     * resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        $project = Project::findOrFail($POST[
            "proj_id"]);

        if (Gate::denies('section', $project))
        {
            abort(403);
        }

        return view('sections.create', array('
            project' => $project));
    }

    /**
     * Store a newly created resource in
     * storage.
     *
     * @param \Illuminate\Http\Request
     * $request
     * @return \Illuminate\Http\Response
     */
    public function store(StoreSectionRequest
        $request)
    {
        // Store data to sections table
        $input = $request->only(['proj_id', '
            section_key']);
        $section = Section::create($input);
        $section_id = $section->id;

        // Store data to section_bodies table
        $input = $request->only(['section_text
            ']);
        $input['section_id'] = $section_id;
        SectionBody::create($input);

        $project = Section::findOrFail(
            $section_id)->project;
        $section_name = Helper::getSectionName
            ($request->section_key);
        // Add the activity to activity table
        Activity::create([
            'proj_id' => $project->id,
            'user_id' => Auth::user()->id,
            'activity_type' => "Report_section
                <i>$section_name</i>_in_
                project <i>$project->
                proj_name</i>_has_been_
                created."
        ]);

        $flash_message = "Report_section_
            $section_name_in_project_<i>$project
            ->proj_name</i>_has_been_
            successfully
            _created.";
        return redirect()->route('sections.
            show', $section_id)->with('
            success', $flash_message);
    }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        $section = Section::findOrFail($id);
        $project = $section->project;

        if (Gate::denies('show-section',
            $project)) {
            abort(403);
        }

        return view('sections.show', array('
            section' => $section, 'project'
            => $project));
    }

    /**
     * Show the form for editing the specified
     * resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        $section = Section::findOrFail($id);
        $project = $section->project;

        if (Gate::denies('section', $project))
        {
            abort(403);
        }

        return view('sections.edit', array('
            section' => $section, 'project'
            => $project));
    }

    /**
     * Update the specified resource in
     * storage.
     *
     * @param \Illuminate\Http\Request
     * $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(
        UpdateSectionRequest $request, $id)
    {
        $section_body = Section::findOrFail(
            $id)->sectionBody;
        $section_body->section_text = $request
            ->input('section_text');
        $section_body->save();

        $project = Section::find($id)->project
            ;
        $section_name = Helper::getSectionName
            ($request->section_key);
        // Add the activity to activity table
        Activity::create([
            'proj_id' => $project->id,
            'user_id' => Auth::user()->id,
            'activity_type' => "Report_section
                <i>$section_name</i>_in_
                project <i>$project->
                proj_name</i>_has_been_
                modified."
        ]);

        $flash_message = "Report_section_
            $section_name_in_project_<i>$project
            ->proj_name</i>_has_been_
            successfully
            _modified.";
        return redirect()->route('sections.
            show', $section_body->section_id)
            ->with('success', $flash_message)
            ;
    }

    /**
     * Remove the specified resource from
     * storage.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        $section = Section::findOrFail($id);

        if (Gate::denies('section', Project::
            find($section->proj_id))) {
            abort(403);
        }

        $section_key = $section->section_key;

```

```

        $proj_id = $section->proj_id;
        $section->delete();

        return SectionController::
            recordDeleteActivity($proj_id,
                $section_key);
    }

    public function recordDeleteActivity(
        $proj_id, $section_key)
    {
        $project = Project::findOrFail(
            $proj_id)->select('proj_name')->
            first();
        $section_name = Helper::getSectionName(
            $section_key);
        var_dump($section_name);
        // Add the activity to activity table
        Activity::create([
            'proj_id' => $proj_id,
            'user_id' => Auth::user()->id,
            'activity_type' => "Report_section_
                _$section_name_in_project_<i>
                _$project->proj_name_</i>_has_
                been_deleted."
        ]);

        $flash_message = "Report_section_
            _$section_name_in_project_
            _>proj_name_has_been_successfully_
            _deleted.";
        return redirect()->route('projects.
            show', $proj_id)->with('success',
                $flash_message);
    }
}

```

Listing 15: ShallowController.php

```

<?php
namespace App\Http\Controllers;

use Auth;
use DB;
use Input;
use Gate;
use App\Activity;
use App\Project;
use App\Sample;
use App\Shallow;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests\Shallow\
    StoreShallowRequest;
use App\Http\Requests\Shallow\
    UpdateShallowRequest;

class ShallowController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index($id)
    {
        $project = Project::findOrFail($id);

        if (Gate::denies('index-shallow',
            $project)) {
            abort(403);
        }

        $boreholes = Project::find($project->
            id)->boreholes;
        $samples = Sample::join('boreholes', '
            samples.bh.id', '=', 'boreholes.
            id')
            ->where('boreholes.proj_id',
                $project->id)
            ->select('samples.id', 'samples.
                bh.id', 'samples.sample_num',
                'samples.sample_to',
                'samples.sample_num_of_blows')
            ->groupBy('samples.id')
            ->orderBy('samples.bh.id')
            ->orderBy('samples.id')
            ->get();
        $shallows = Shallow::where('proj_id',
            $project->id)->get();

        ShallowController::isUpdated($project
            ->id, sizeof($samples), sizeof(

```

```

        $shallows));

        return view('shallows.index', array('
            project' => $project, 'boreholes'
            => $boreholes, 'samples' =>
            $samples, 'shallows' => $shallows
        ));
    }

    /**
     * Show the form for creating a new
     * resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create($id)
    {
        $project = Project::findOrFail($id);

        if (Gate::denies('shallow', $project))
        {
            abort(403);
        }

        $shallows = $project->shallows;

        if(sizeof($shallows) == 0) {
            $boreholes = Project::findOrFail(
                $project->id)->boreholes;
            $samples = Sample::join('boreholes',
                'samples.bh.id', '=', '
                boreholes.id')
                ->where('boreholes.proj_id',
                    $project->id)
                ->select('samples.id', '
                    samples.bh.id', 'samples.
                    sample_num', 'samples.
                    sample_to', 'samples.
                    sample_num_of_blows')
                ->get();

            ShallowController::
                checkPrerequisites($boreholes
                    , $samples);

            return view('shallows.create',
                array('project' => $project,
                    'boreholes' => $boreholes,
                    'samples' => $samples));
        }
        else {
            $flash_message = "SBC_data_for_
                shallow_foundation_in_project_
                _$project->proj_name_already_
                exists_._You_cannot_create_
                another_instance_of_the_data.
                ";
            return redirect()->route('projects.
                show', $project->id)->with('
                error', $flash_message);
        }
    }

    /**
     * Store a newly created resource in
     * storage.
     *
     * @param \Illuminate\Http\Request
     * $request
     * @return \Illuminate\Http\Response
     */
    public function store(StoreShallowRequest
        $request)
    {
        $project = Project::findOrFail(
            $request->proj_id);
        $j = $k = 0;

        for($i = 0; $i < $project->
            proj_num_of_boreholes; $i++) {
            while($j < $request->
                num_of_samples[$i]) {
                Shallow::create([
                    'proj_id' => $project->id,
                    'bh.id' => $request->bh.id
                        [$i],
                    'sample.id' => $request->
                        sample.id[$k],
                    'shallow_B' => $request->
                        shallow_B[$i],
                    'shallow_phi' => $request
                        ->shallow_phi[$i],
                    'shallow_Nc' => $request->
                        shallow_Nc[$i],
                    'shallow_Nc' => $request->
                        shallow_Nc[$i],
                    'shallow_Ngamma' =>
                        $request->
                        shallow_Ngamma[$i],
                    'shallow_Nq' => $request->
                        shallow_Nq[$i],

```

```

        'shallow_gamma1' =>
            $request->
                shallow_gamma1[$i],
        'shallow_gamma2' =>
            $request->
                shallow_gamma2[$i],
        'shallow_fs' => $request->
            shallow_fs[$i],
        'shallow_soil' => $request
            ->shallow_soil[$k],
        'shallow_c' => $request->
            shallow_c[$k],
        'shallow_qult' => $request
            ->shallow_qult[$k],
        'shallow_q' => $request->
            shallow_q[$k]
    });

    $j++;
    $k++;
}
$j = 0;
}

if((sizeof($request->shallow_is_chosen
) > 0) && ($request->
shallow_is_chosen[0] > -1)) {
    for($i = 0; $i < sizeof($request->
shallow_is_chosen); $i++) {
        for($j = 0; $j < sizeof(
$request->sample_id); $j
        ++){
            if($request->
shallow_is_chosen[$i]
            == $request->
sample_id[$j]) {
                $shallow = Sample::
                    find($request->
sample_id[$j])->
shallow;
                $shallow->update(['
shallow_is_chosen
' => 1]);
            }
        }
    }

    // Add the activity to activity table
    Activity::create([
        'proj_id' => $project->id,
        'user_id' => Auth::user()->id,
        'activity_type' => "Data_for_SBC_
for_shallow_foundation_in_
project_<i>$project->
proj_name</i>_has_been_
created."
    ]);

    $flash_message = "Data_for_SBC_for_
shallow_foundation_in_
project_
$project->proj_name_has_been_
successfully_saved.";
    return redirect()->route('shallows.
index', $project->id)->with(['
success', $flash_message]);
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    //
}

/**
 * Show the form for editing the specified
resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    $project = Project::findOrFail($id);

    if (Gate::denies('shallow', $project))
    {
        abort(403);
    }

    $boreholes = Project::findOrFail(
        $project->id)->boreholes;
    $samples = Sample::join('boreholes', '
samples.bh_id', '=', 'boreholes.
id')
        ->where('boreholes.proj_id',
            $project->id)
        ->select('samples.id', 'samples.
bh_id', 'samples.sample_num',
            'samples.sample_to',
            'samples.sample_num.of_blow')
        ->groupBy('samples.id')
        ->orderBy('samples.bh_id')
        ->orderBy('samples.id')
        ->get();

    $shallows = Shallow::where('proj_id',
        $id)->get();

    ShallowController::isUpdated($project
        ->id, sizeof($samples), sizeof(
        $shallows));

    return view('shallows.edit', array('
project' => $project, 'boreholes'
=> $boreholes, 'samples' =>
        $samples, 'shallows' => $shallows
    ));
}

/**
 * Update the specified resource in
storage.
 *
 * @param \Illuminate\Http\Request
$request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(
    UpdateShallowRequest $request, $id)
{
    // Update information
    $project = Project::findOrFail($id);
    $j = $k = 0;
    for($i = 0; $i < $project->
proj_num_of_boreholes; $i++) {
        while($j < $request->
num_of_samples[$i]) {
            $shallow = Shallow::findOrFail(
                $request->shallow_id[$k
                ]);

            $shallow->update([
                'shallow_B' => $request->
shallow_B[$i],
                'shallow_phi' => $request
                ->shallow_phi[$i],
                'shallow_Nc' => $request->
shallow_Nc[$i],
                'shallow_Ngamma' =>
                $request->
shallow_Ngamma[$i],
                'shallow_Nq' => $request->
shallow_Nq[$i],
                'shallow_gamma1' =>
                $request->
shallow_gamma1[$i],
                'shallow_gamma2' =>
                $request->
shallow_gamma2[$i],
                'shallow_fs' => $request->
shallow_fs[$i],
                'shallow_soil' => $request
                ->shallow_soil[$k],
                'shallow_c' => $request->
shallow_c[$k],
                'shallow_qult' => $request
                ->shallow_qult[$k],
                'shallow_q' => $request->
shallow_q[$k],
                'shallow_is_chosen' => 0
            ]);

            $j++;
            $k++;
        }
        $j = 0;
    }

    if((sizeof($request->shallow_is_chosen
) > 0) && ($request->
shallow_is_chosen[0] > -1)) {
        for($i = 0; $i < sizeof($request->
shallow_is_chosen); $i++) {
            for($j = 0; $j < sizeof(
$request->sample_id); $j
            ++){
                if($request->
shallow_is_chosen[$i]
                == $request->
sample_id[$j]) {
                    $shallow = Sample::
                        find($request->

```

```

        sample_id[$j]->
        shallow;
        $shallow->update(['
        shallow_is_chosen
        ' => 1]);
    }
}
}

// Add the activity to activity table
Activity::create([
    'proj_id' => $request->proj_id,
    'user_id' => Auth::user()->id,
    'activity_type' => "Data_for_SBC_
    for_shallow_foundation_in_
    project_<i>$project->
    proj_name</i>_has_been_
    modified."
]);

$flash_message = "Data_for_SBC_for_
shallow_foundation_in_project_
$project->proj_name_has_been_
successfully_modified.";
return redirect()->route('shallows.
index', $project->id)->with('
success', $flash_message);
}

/**
 * Remove the specified resource from
 * storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    $shallows = Project::findOrFail($id)->
    shallows;

    if (Gate::denies('shallow', Project::
    find($id))) {
        abort(403);
    }

    foreach($shallows as $shallow) {
        $shallow->delete();
    }

    return ShallowController::
    recordDeleteActivity($id);
}

public function recordDeleteActivity(
    $proj_id)
{
    $project = Project::findOrFail(
    $proj_id);

    // Add the activity to activity table
    Activity::create([
        'proj_id' => $project->id,
        'user_id' => Auth::user()->id,
        'activity_type' => "Data_for_SBC_
        for_shallow_foundation_in_
        project_<i>$project->
        proj_name</i>_has_been_
        deleted."
    ]);

    $flash_message = "Data_for_SBC_for_
    shallow_foundation_in_project_
    $project->proj_name_has_been_
    successfully_deleted.";
    return redirect()->route('projects.
    show', $project->id)->with('
    success', $flash_message);
}

public function isUpdated($proj_id,
    $size_of_samples, $size_of_shallows)
{
    $is_updated = false;

    // Check if data is not yet created
    if($size_of_shallows == 0) {
        $route = route('shallows.create',
        $proj_id);
        $flash_message = "SBC_data_for_
        shallow_foundation_is_missing_
        .<a_href=$route_class='btn_
        btn-link_btn-sm'>Add_one_here_
        </a>.";
        \Session::flash('warning_e',
        $flash_message);
        return;
    }

    // Check if a new sample has been
    created
    else if($size_of_samples !=
    $size_of_shallows) {
        $is_updated = true;
    }

    if($is_updated) {
        $route = route('shallows.
        deleteAndCreate', $proj_id);
        $flash_message = "Samples_were_
        recently_modified._Please_
        update_the_data_by_<a_href=
        $route_class='btn_btn-link_
        btn-sm'>creating_a_new_one</a_
        >_to_avoid_errors.";
        \Session::flash('warning_e',
        $flash_message);
    }
    return;
}

public function deleteAndCreate($proj_id)
{
    $matchThese = ['proj_id' => $proj_id,
    'enr_id' => Auth::user()->id];
    $project = DB::table('
    project_engineers')->where(
    $matchThese)->get();

    if(sizeof($project) == 0) {
        return view('errors.403');
    }

    ShallowController::destroy($proj_id);
    return ShallowController::create(
    $proj_id);
}

public function checkPrerequisites(
    $boreholes, $samples) {
    $arePrerequisitesOkay = true;
    if(sizeof($boreholes) == 0) {
        $arePrerequisitesOkay = false;
    }
    if (sizeof($samples) == 0) {
        $arePrerequisitesOkay = false;
    }

    if(!$arePrerequisitesOkay) {
        $flash_message = "Some_
        prerequisites_are_missing_(
        boreholes_or_samples)._Please_
        add_them_first_in_order_to_
        proceed_smoothly.";
        \Session::flash('warning',
        $flash_message);
    }
}
}
}
}

Listing 16: SubreportController.php
<?php
namespace App\Http\Controllers;

use Auth;
use DB;
use Input;
use Gate;
use App\Activity;
use App\DataPoint;
use App\Limit;
use App\Sample;
use App\Project;
use App\Subreport;
use App\SubreportBody;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class SubreportController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        //
    }

    /**
     * Show the form for creating a new
     * resource.

```



```

* @return \Illuminate\Http\Response
*/
public function create()
{
    //
}
/**
 * Store a newly created resource in
 * storage.
 *
 * @param \Illuminate\Http\Request
 * $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    //
}
/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    //
}
/**
 * Show the form for editing the specified
 * resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit()
{
    //
}
/**
 * Update the specified resource in
 * storage.
 *
 * @param \Illuminate\Http\Request
 * $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request,
                        $id)
{
    //
}
/**
 * Remove the specified resource from
 * storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    $subreport = Subreport::findOrFail($id
    );
    $project = Project::find($subreport->
    proj_id);

    if (Gate::denies('subreport', $project
    )) {
        abort(403);
    }

    switch($subreport->sb_key) {
        case 2: // Borehole Log
            $sb_name = "Borehole_Log";
            $boreholes = Project::find(
            ($subreport->proj_id)->
            boreholes;
            foreach($boreholes as
            $borehole) {
                $file_path = "images/
                subreports/BL/bh-
                $borehole->id.png";
                if (file_exists($file_path
                )) {
                    if (!unlink($file_path
                    )) {
                        $status = "warning
                    ";
                    }
                    else {
                        $status = "success
                    ";
                    }
                }
            }
        }
    }

    }
}
/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function showGSA($id)
{
    $subreport = Subreport::findOrFail($id
    );

    if (Gate::denies('show-subreport',
    $subreport->project)) {
        abort(403);
    }

    $project = DB::table('projects')
    ->join('subreports', 'projects.id'
    , '=', 'subreports.proj_id')
    ->where('projects.id', $subreport
    ->proj_id)
    ->select('projects.id', 'projects.
    proj_name', 'projects.
    proj_location', 'projects.
    proj_client',
    'projects.
    proj_num_of_boreholes')
    ->first();
    $samples = DB::table('samples')
    ->join('data_points', 'samples.id'
    , '=', 'data_points.sample_id

```

```

    )
->join('uscs', 'samples.uscs_id',
      '=', 'uscs.id')
->where('data_points.subreport_id',
      $subreport->id)
->select('samples.id', 'samples.
bh_id', 'samples.uscs_id',
uscs.uscs_description',
samples.sample_num', 'samples.
.sample_from', 'samples.
.sample_to', 'samples.
.sample_job_num', 'samples.
.sample_test_date', 'samples.
.sample_color', 'samples.
.sample_nmc_percent', 'samples.
.sample_liquid_limit',
samples.sample_plastic_limit',
'samples.
.sample_plasticity_index')
->groupBy('samples.id')
->orderBy('samples.bh_id')
->orderBy('samples.id')
->get();

return view('subreports.showGSA',
array('subreport' => $subreport,
'project' => $project, 'samples'
=> $samples));
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function showLPTR($id)
{
    $subreport = Subreport::findOrFail($id
    );

    if (Gate::denies('show-subreport',
        $subreport->project)) {
        abort(403);
    }

    $project = DB::table('projects')
->join('subreports', 'projects.id',
      '=', 'subreports.proj_id')
->where('projects.id', $subreport
->proj_id)
->select('projects.id', 'projects.
proj_name', 'projects.
proj_location', 'projects.
proj_client',

'projects.
proj_num_of_boreholes')
->first();
    $samples = DB::table('samples')
->join('limits', 'samples.id', '=',
      'limits.sample_id')
->where('limits.subreport_id',
      $subreport->id)
->select('samples.id', 'samples.
bh_id', 'sample_job_num',
'sample_contract_num',
'sample_from', 'sample_to',
'sample_test_date',
'sample_tester',
'sample_liquid_limit',
'sample_plastic_limit',
'sample_plasticity_index',
'sample_plasticity_remark')
->groupBy('samples.id')
->orderBy('samples.bh_id')
->orderBy('samples.id')
->get();

    $matchThese = ['subreport_id' =>
        $subreport->id, 'lim_test' => 0];
    $liquid_limits = Limit::where(
        $matchThese->get());

    $matchThese = ['subreport_id' =>
        $subreport->id, 'lim_test' => 1];
    $plastic_limits = Limit::where(
        $matchThese->get());

    return view('subreports.showLPTR',
array('subreport' => $subreport,
'project' => $project, 'samples'
=> $samples, 'liquid_limits' =>
    $liquid_limits, 'plastic_limits'
=> $plastic_limits));
}

/**
 * Show the form for creating a new

```

```

resource.
 *
 * @return \Illuminate\Http\Response
 */
public function createBL($id)
{
    $matchThese = ['proj_id' => $id,
        engr_id' => Auth::user()->id];
    $project = DB::table('
project_engineers')->where(
        $matchThese->get());

    if(sizeof($project) > 0) {
        list($project, $boreholes, $counts
        ) = SubreportController::
            getData($id);

        $boreholes_info = DB::table('
boreholes')
->join('projects', 'boreholes.
proj_id', '=', 'projects.
id')
->where('boreholes.proj_id',
        $id)
->select('boreholes.id',
        bh_start_date',
        bh_end_date', 'bh_num',
        bh_max_depth',
        bh_weather', 'bh_operator',
        'bh_inspector',
        bh_drill_rig',
        bh_drill_pump',
        bh_hammer_weight',
        bh_height_of_drop',
        bh_casing', 'bh_rod',
        bh_sampler',
        bh_num_of_samples',
        bh_water_elevation',
        bh_surface_elevation')
->get();

        $samples_info = DB::select('SELECT
        _s.id AS _sample_id, _b.id AS _
        bh_id, _uscs.id, _uscs.
        uscs_description, _sample_num,
        _sample_from, _sample_to, _
        sample_color, _
        sample_run_length, _
        sample_sample_length, _
        sample_rec_percent, _
        sample_num_of_blows, _
        sample_type_of_sampling, _
        sample_test_date, _
        sample_other_results FROM _
        samples _s, _boreholes _b, _
        projects _p, _uscs _WHERE _s.
        bh_id = _b.id AND _b.proj_id = _
        p.id AND _uscs.id = _s.uscs_id
        AND _p.id = ? ORDER BY _
        sample_from', [$id]);

        // Add the activity to activity
        table
        Activity::create([
            'proj_id' => $project->id,
            'user_id' => Auth::user()->id,
            'activity_type' => "Borehole_
            Log_in_project<i>
            $project->proj_name</i>_
            has_been_created/updated.
            "
        ]);

        $flash_message = "Borehole_Log_in_
        project_-$project->proj_name_
        has_been_successfully_created
        /updated.";
        \Session::flash('success',
            $flash_message);
        return view('subreports.createBL',
array('project' => $project,
'boreholes_info' =>
        $boreholes_info,
'samples_info' => $samples_info,
'counts' => $counts));
    }
    else {
        return view('errors.403');
    }
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function showBL($id)
{

```

```

$subreport = Subreport::findOrFail($id
);

if (Gate::denies('show-subreport',
    $subreport->project)) {
    abort(403);
}

$project = DB::table('projects')
->join('subreports', 'projects.id',
    '=', 'subreports.proj_id')
->where('projects.id', $subreport
->proj_id)
->select('projects.id', 'projects.
    proj_name')
->first();

$boreholes_info = DB::table('boreholes
')
->join('projects', 'boreholes.
    proj_id', '=', 'projects.id')
->where('boreholes.proj_id',
    $project->id)
->select('boreholes.id', '
    bh_start_date', 'bh_end_date',
    'bh_num', 'bh_max_depth', '
    bh_weather', 'bh_operator', '
    bh_inspector', 'bh_drill_rig',
    'bh_drill_pump', '
    bh_hammer_weight', '
    bh_height_of_drop', '
    bh_casing', 'bh_rod', '
    bh_sampler', '
    bh_num_of_samples', '
    bh_water_elevation', '
    bh_surface_elevation')
->get();

return view('subreports.showBL', array
('project' => $project, '
    boreholes_info' => $boreholes_info
));
}

/**
 * Show the form for creating a new
    resource.
 *
 * @return \Illuminate\Http\Response
 */
public function createSP($id)
{
    $matchThese = ['proj_id' => $id, '
    engr_id' => Auth::user()->id];
    $project = DB::table('
    project_engineers')->where(
    $matchThese)->get();

    if(sizeof($project) > 0) {
        list($project, $boreholes, $counts
        ) = SubreportController::
        getData($id);

        $boreholes_info = DB::table('
        boreholes')
        ->join('projects', 'boreholes.
        proj_id', '=', 'projects.
        id')
        ->where('boreholes.proj_id',
        $id)
        ->select('boreholes.id', '
        bh_num', 'bh_max_depth', '
        bh_num_of_samples', '
        bh_water_elevation', '
        bh_surface_elevation')
        ->get();

        $samples_info = DB::select('SELECT
        _s.id AS _sample_id, _b.id AS _
        bh_id, _uscs_id, _sample_num, _
        sample_from, _sample_to, _
        sample_num_of_blows, _
        sample_type_of_sampling, _
        sample_run_length FROM _
        samples _s, _boreholes _b, _
        projects _p, _uscs _WHERE _s.
        bh_id = _b.id AND _b. proj_id = _
        p.id AND _uscs.id = _s.uscs_id
        AND _p.id = ?_ORDER_BY_
        sample_from', [$id]);

        // Add the activity to activity
        table
        Activity::create([
            'proj_id' => $project->id,
            'user_id' => Auth::user()->id,
            'activity_type' => "Soil_
                Profile_in_project_<i>
                    $project->proj_name</i>-
                    was_created/updated."
        ]);

        $flash_message = "Soil_Profile_in_
        project_<_proj_name_
        has_been_successfully_created
        /updated.";
        \Session::flash('success',
        $flash_message);
        return view('subreports.createSP',
        array('project' => $project,
            'boreholes_info' =>
            $boreholes_info, '
            samples_info' => $samples_info
            , 'counts' => $counts));
    }
    else {
        return view('errors.403');
    }
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function showSP($id)
{
    $subreport = Subreport::findOrFail($id
);
    $project = DB::table('projects')
->join('subreports', 'projects.id',
    '=', 'subreports.proj_id')
->where('projects.id', $subreport
->proj_id)
->select('projects.id', 'projects.
    proj_name', 'projects.
    proj_num_of_boreholes')
->first();

    return view('subreports.showSP', array
('project' => $project));
}

/**
 * Store an image generated in the
    subreports
 *
 * @param
 * @return
 */
public function storeImage()
{
    if(isset($_POST['image-uri'])) {
        if($_POST['sb_type'] == 'GSA') {
            // Variables for saving image
            to folder
            $sample_id = $_POST['
            sample_number'];
            $file_path = "images/
            subreports/GSA/sample-
            $sample_id.png";
        }
        else if($_POST['sb_type'] == '
        LPTR') {
            // Variables for saving image
            to folder
            $sample_id = $_POST['
            sample_number'];
            $file_path = "images/
            subreports/LPTR/sample-
            $sample_id.png";
        }
    }
    else if($_POST['sb_type'] == 'BL')
    {
        // Check if subreport is
        already in the database
        $matchThese = ['proj_id' =>
        $_POST['proj_id'], '
        sb_key' => 2];
        $match = DB::table('subreports
        ')->where($matchThese)->
        first();

        if(is_null($match)) {
            // Store data to
            subreports table
            $input_sb['sb_key'] = 2;
            $input_sb['proj_id'] =
            $_POST['proj_id'];
            $subreport = Subreport::
            create($input_sb);
        }

        // Variables for saving image
        to folder

```

```

        $bh_id = $_POST['bh_number'];
        $file_path = "images/
        subreports/BL/bh-$bh_id.
        png";
    }
    else if($_POST['sb-type'] == 'SP')
    {
        // Check if subreport is
        // already in the database
        $matchThese = ['proj_id' =>
        $_POST['proj_id'], '
        sb_key' => 3];
        $match = DB::table('subreports
        ')->where($matchThese)->
        first();

        if(is_null($match)) {
            // Store data to
            // subreports table
            $input_sb['sb_key'] = 3;
            $input_sb['proj_id'] =
            $_POST['proj_id'];
            $subreport = Subreport::
            create($input_sb);
        }

        // Variables for saving image
        // to folder
        $proj_id = $_POST['proj_id'];
        $canvas_num = $_POST['
        canvas_num'];
        $file_path = "images/
        subreports/SP/proj-
        $proj_id-$canvas_num.png"
        ;
    }

    $encoded_data = $_POST['image-uri'
    ];
    $encoded_data = str_replace('data:
    image/png;base64,', '',
    $encoded_data);
    $encoded_data = str_replace('_', '
    +', $encoded_data);
    $decoded_data = base64_decode(
    $encoded_data);
    $success = file_put_contents(
    $file_path, $decoded_data);
    echo $success ? $file_path :
    Unable_to_save_the_file.;
}

public function getData($id)
{
    $project = Project::findOrFail($id);

    $boreholes = DB::table('boreholes')
    ->join('projects', 'boreholes.
    proj_id', '=', 'projects.id')
    ->where('boreholes.proj_id', $id)
    ->select('boreholes.id')
    ->get();

    if (sizeof($boreholes) > 0) { // If
    there are boreholes in project
    $i = 0;
    foreach($boreholes as $borehole) {
        $samples = DB::table('samples'
        )
        ->join('boreholes', '
        samples.bh_id', '=',
        'boreholes.id')
        ->where('samples.bh_id',
        $borehole->id)
        ->select('samples.id')
        ->get();

        $boreholes[$i] = $borehole->id
        ;
        $j = 0;

        if(sizeof($samples) > 0) { //
        If there are samples in
        current borehole
        foreach($samples as
        $sample) {
            $counts[$i][$j] =
            $sample->id;
            $j++;
        }
        $i++;
    }
    }
    else {
        $counts = 0;
    }
}

if(!isset($counts)) {
    $counts = 0;
}

return array($project, $boreholes,
    $counts);
}
}

Listing 17: UserController.php
<?php
namespace App\Http\Controllers;

use Auth;
use DB;
use Gate;
use App\Activity;
use App\User;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class UserController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $users = User::orderBy('lastname')->
        get();

        if (Gate::denies('manage-user')) {
            abort(403);
        }

        return view('users.index', array('
        users' => $users));
    }

    /**
     * Show the form for creating a new
     * resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        //
    }

    /**
     * Store a newly created resource in
     * storage.
     *
     * @param \Illuminate\Http\Request
     * $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        //
    }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        $user = User::findOrFail($id);

        if (Gate::denies('show-user')) {
            abort(403);
        }

        $activities = DB::table('activities')
        ->leftJoin('projects', 'activities
        .proj_id', '=', 'projects.id'
        )
        ->where('activities.user_id',
        $user->id)
        ->select('activity_type', 'proj_id
        ', 'proj_name', 'proj_image',
        'activities.created_at')
        ->orderBy('created_at', 'desc')
        ->take(10)
        ->get();

        return view('users.show', array('user'
        => $user, 'activities' =>
        $activities));
    }
}

```

```

}

/**
 * Show the form for editing the specified
 * resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    $user = User::findOrFail($id);

    if (Gate::denies('edit-user', $user->id)) {
        abort(403);
    }

    return view('users.edit', array('user' => $user));
}

/**
 * Update the specified resource in
 * storage.
 *
 * @param \Illuminate\Http\Request
 * $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(UpdateUserRequest
    $request, $id)
{
    $user = User::findOrFail($id);

    // Validate the request
    $this->validate($request, [
        'firstname' => 'required|max:255',
        'middlename' => 'max:255',
        'lastname' => 'required|max:255',
        'email' => 'required|email|max:255|unique:users,email',
        $user->id,
        'contact' => 'required|max:14',
        'sex' => 'max:1',
        'birthdate' => 'date|before:yesterday',
        'about' => 'max:255'
    ]);

    // Update user information
    $user->update([
        'firstname' => $request->firstname,
        'middlename' => $request->
            middlename,
        'lastname' => $request->lastname,
        'email' => $request->email,
        'contact' => $request->contact,
        'sex' => $request->sex,
        'birthdate' => $request->birthdate,
        'about' => $request->about
    ]);

    // Add the activity to activity table
    Activity::create([
        'proj_id' => null,
        'user_id' => Auth::user()->id,
        'activity_type' => "<i>$user->
            firstname_{$user->lastname}</i>
            >'s_user_profile_has_been_
            modified."
    ]);

    $flash_message = "Your_profile_has_
        been_successfully_modified.";
    return redirect()->route('users.show',
        $user->id)->with('success',
        $flash_message);
}

/**
 * Remove the specified resource from
 * storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    if (Gate::denies('manage-user')) {
        abort(403);
    }

    $user = User::findOrFail($id);
    $user->delete();

    $user_name = $user->firstname . "_" .
        $user->lastname;

    return UserController::
        recordDeleteActivity($user_name);
}

public function recordDeleteActivity(
    $user_name) {
    // Add the activity to activity table
    Activity::create([
        'proj_id' => null,
        'user_id' => Auth::user()->id,
        'activity_type' => "<i>$user_name
            </i>_has_been_deleted."
    ]);

    $flash_message = "$user_name_has_been_
        successfully_deleted.";
    return redirect()->route('users.index')
        ->with('success', $flash_message);
}

/**
 * Disable the user.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function disableUser(Request
    $request)
{
    if (Gate::denies('manage-user')) {
        abort(403);
    }

    $user = User::findOrFail($request->
        user_id);

    $user->update([
        'status' => 3,
        'position' => 'Disabled_User'
    ]);

    $user_name = $user->firstname . "_" .
        $user->lastname;

    // Add the activity to activity table
    Activity::create([
        'proj_id' => null,
        'user_id' => Auth::user()->id,
        'activity_type' => "<i>$user_name
            </i>_has_been_disabled."
    ]);

    $flash_message = "$user_name_has_been_
        successfully_disabled.";
    return redirect()->route('users.index')
        ->with('success', $flash_message);
}

/**
 * Accept the pending user.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function acceptUser(Request
    $request)
{
    if (Gate::denies('manage-user')) {
        abort(403);
    }

    $user = User::findOrFail($request->
        user_id);

    $user->update([
        'status' => 1,
        'position' => 'Engineer'
    ]);

    $user_name = $user->firstname . "_" .
        $user->lastname;

    // Add the activity to activity table
    Activity::create([
        'proj_id' => null,
        'user_id' => Auth::user()->id,
        'activity_type' => "<i>$user_name
            </i>_has_been_accepted."
    ]);

    $flash_message = "$user_name_has_been_
        successfully_accepted.";
    return redirect()->route('users.index')
        ->with('success', $flash_message);
}

```

```

}

/**
 * Update user position.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function changeUserPosition()
{
    if (Gate::denies('manage-user')) {
        abort(403);
    }

    $id = $_POST['id'];
    $position = $_POST['position'];
    $user = User::findOrFail($id);

    // Update user position
    $user->update([
        'position' => $position
    ]);

    $user_name = $user->firstname . " " .
        $user->lastname;
    // Add the activity to activity table
    Activity::create([
        'proj_id' => null,
        'user_id' => Auth::user()->id,
        'activity_type' => "User<i>
            $user_name</i>'s<i>position_has
            _been_changed_to_<i>$position."
    ]);

    $flash_message = "User_name's_
        position_has_been_changed_to_
        $position.";
    return redirect()->route('users.index',
        )->with('success', $flash_message
    );
}
}
}

```

Listing 18: Authenticate.php

```

<?php
namespace App\Http\Middleware;

use Closure;
use Illuminate\Contracts\Auth\Guard;

class Authenticate
{
    /**
     * The Guard implementation.
     *
     * @var Guard
     */
    protected $auth;

    /**
     * Create a new filter instance.
     *
     * @param Guard $auth
     * @return void
     */
    public function __construct(Guard $auth)
    {
        $this->auth = $auth;
    }

    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request
     * $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure
        $next)
    {
        if ($this->auth->guest()) {
            if ($request->ajax()) {
                return response('Unauthorized.',
                    401);
            } else {
                return redirect()->guest('auth
                    /login');
            }
        }

        return $next($request);
    }
}
}

```

Listing 19: EncryptCookies.php

```

<?php
namespace App\Http\Middleware;

use Illuminate\Cookie\Middleware\
    EncryptCookies as BaseEncrypter;

class EncryptCookies extends BaseEncrypter
{
    /**
     * The names of the cookies that should
     * not be encrypted.
     *
     * @var array
     */
    protected $except = [
        //
    ];
}

```

Listing 20: RedirectIfAuthenticated.php

```

<?php
namespace App\Http\Middleware;

use Closure;
use Illuminate\Contracts\Auth\Guard;

class RedirectIfAuthenticated
{
    /**
     * The Guard implementation.
     *
     * @var Guard
     */
    protected $auth;

    /**
     * Create a new filter instance.
     *
     * @param Guard $auth
     * @return void
     */
    public function __construct(Guard $auth)
    {
        $this->auth = $auth;
    }

    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request
     * $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure
        $next)
    {
        if ($this->auth->check()) {
            return redirect('/home');
        }

        return $next($request);
    }
}

```

Listing 21: VerifyCsrfToken.php

```

<?php
namespace App\Http\Middleware;

use Illuminate\Foundation\Http\Middleware\
    VerifyCsrfToken as BaseVerifier;

class VerifyCsrfToken extends BaseVerifier
{
    /**
     * The URIs that should be excluded from
     * CSRF verification.
     *
     * @var array
     */
    protected $except = [
        //
    ];
}

```

Listing 22: StoreBoreholeRequest.php

```
<?php
namespace App\Http\Requests\Borehole;

use DB;
use Auth;
use Input;
use Response;
use Illuminate\Foundation\Http\FormRequest;

class StoreBoreholeRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to
     * make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        if (Auth::check()) {
            return true;
        }

        $proj_id = Input::get('proj_id');

        if (Auth::user()->isInProject($proj_id)) {
            return true;
        }

        return false;
    }

    /**
     * Get the validation rules that apply to
     * the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'proj_id' => 'required',
            'bh_start_date' => 'required|date',
            'bh_end_date' => 'required|date',
            'bh_num' => 'required|numeric',
            'bh_max_depth' => 'required|numeric',
            'bh_weather' => 'string',
            'bh_operator' => 'required',
            'bh_inspector' => 'string',
            'bh_drill_rig' => 'string',
            'bh_drill_pump' => 'string',
            'bh_hammer_weight' => 'numeric',
            'bh_height_of_drop' => 'numeric',
            'bh_casing' => 'required',
            'bh_rod' => 'required',
            'bh_sampler' => 'required',
            'bh_water_elevation' => 'required|numeric',
            'bh_surface_elevation' => 'required|numeric'
        ];
    }

    /**
     * Send a custom response on authorize
     * failure
     *
     */
    public function forbiddenResponse()
    {
        return response()->view('errors.403');
    }
}

```

Listing 23: UpdateBoreholeRequest.php

```
<?php
namespace App\Http\Requests\Borehole;

use DB;
use Auth;
use Input;
use Response;
use Illuminate\Foundation\Http\FormRequest;

class UpdateBoreholeRequest extends FormRequest
{

```

```
/**
 * Determine if the user is authorized to
 * make this request.
 *
 * @return bool
 */
public function authorize()
{
    if (Auth::check()) {
        return true;
    }

    $proj_id = Input::get('proj_id');

    if (Auth::user()->isInProject($proj_id)) {
        return true;
    }

    return false;
}

/**
 * Get the validation rules that apply to
 * the request.
 *
 * @return array
 */
public function rules()
{
    return [
        'proj_id' => 'required',
        'bh_start_date' => 'required|date',
        'bh_end_date' => 'required|date',
        'bh_num' => 'required|numeric',
        'bh_max_depth' => 'required|numeric',
        'bh_weather' => 'string',
        'bh_operator' => 'required',
        'bh_inspector' => 'string',
        'bh_drill_rig' => 'required|string',
        'bh_drill_pump' => 'required|string',
        'bh_hammer_weight' => 'required|numeric',
        'bh_height_of_drop' => 'required|numeric',
        'bh_casing' => 'required',
        'bh_rod' => 'required',
        'bh_sampler' => 'required',
        'bh_water_elevation' => 'required|numeric',
        'bh_surface_elevation' => 'required|numeric'
    ];
}

/**
 * Send a custom response on authorize
 * failure
 *
 */
public function forbiddenResponse()
{
    return response()->view('errors.403');
}
}

```

Listing 24: StoreDataPointRequest.php

```
<?php
namespace App\Http\Requests\DataPoint;

use DB;
use Auth;
use Input;
use Response;
use Illuminate\Foundation\Http\FormRequest;

class StoreDataPointRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to
     * make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        if (Auth::check()) {
            return true;
        }
    }
}

```

```

        $proj_id = Input::get('proj_id');

        if (Auth::user()->isInProject($proj_id)) {
            return true;
        }

        return false;
    }

    /**
     * Get the validation rules that apply to
     * the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'proj_id' => 'required',
            'sample_id' => 'required|array',
            'x_int' => 'required|array',
            'y_int' => 'required|array',
        ];
    }

    /**
     * Send a custom response on authorize
     * failure
     *
     */
    public function forbiddenResponse()
    {
        return response()->view('errors.403');
    }
}

```

Listing 25: UpdateDataPointRequest.php

```

<?php

namespace App\Http\Requests\DataPoint;

use DB;
use Auth;
use Input;
use Response;
use Illuminate\Foundation\Http\FormRequest;

class UpdateDataPointRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to
     * make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        if (Auth::check()) {
            return true;
        }

        $proj_id = Input::get('proj_id');

        if (Auth::user()->isInProject($proj_id)) {
            return true;
        }

        return false;
    }

    /**
     * Get the validation rules that apply to
     * the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'data_point_id' => 'required|array',
            'sample_id' => 'required|array',
            'x_int' => 'required|array',
            'y_int' => 'required|array',
            'proj_id' => 'required'
        ];
    }

    /**
     * Send a custom response on authorize
     * failure
     *
     */

```

```

    */
    public function forbiddenResponse()
    {
        return response()->view('errors.403');
    }
}

```

Listing 26: StoreDeepRequest.php

```

<?php

namespace App\Http\Requests\Deep;

use DB;
use Auth;
use Input;
use Response;
use Illuminate\Foundation\Http\FormRequest;

class StoreDeepRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to
     * make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        if (Auth::check()) {
            return true;
        }

        $proj_id = Input::get('proj_id');

        if (Auth::user()->isInProject($proj_id)) {
            return true;
        }

        return false;
    }

    /**
     * Get the validation rules that apply to
     * the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'proj_id' => 'required|numeric',
            'bh_id' => 'required|array',
            'sample_id' => 'required|array',
            'num_of_samples' => 'required|
                array',
            'deep_pile_size' => 'required|
                array',
            'deep_K' => 'required|array',
            'deep_gamma' => 'required|array',
            'deep_tan' => 'required|array',
            'deep_fs' => 'required|array',
            'deep_rho' => 'required|array',
            'deep_fy' => 'required|array',
            'deep_fc' => 'required|array',
            'deep_concrete_strength_kn' => '
                required|array',
            'deep_concrete_strength_ton' => '
                required|array',
            'deep_f' => 'required|array',
            'deep_Asurface' => 'required|array',
            'deep_q' => 'required|array',
            'deep_Atip' => 'required|array',
            'deep_Qtotal' => 'required|array',
            'deep_Q_kn' => 'required|array',
            'deep_Q_ton' => 'required|array',
            'deep_is_chosen' => 'required'
        ];
    }

    /**
     * Send a custom response on authorize
     * failure
     *
     */
    public function forbiddenResponse()
    {
        return response()->view('errors.403');
    }
}

```

Listing 27: UpdateDeepRequest.php


```

<?php
namespace App\Http\Requests\Deep;

use DB;
use Auth;
use Input;
use Response;
use Illuminate\Foundation\Http\FormRequest;

class UpdateDeepRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to
     * make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        if (Auth::check()) {
            return true;
        }

        $proj_id = Input::get('proj_id');

        if (Auth::user()->isInProject($proj_id
        )) {
            return true;
        }

        return false;
    }

    /**
     * Get the validation rules that apply to
     * the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'proj_id' => 'required',
            'bh_id' => 'required|array',
            'sample_id' => 'required|array',
            'deep_id' => 'required|array',
            'num_of_samples' => 'required|
            array',
            'deep_pile_size' => 'required|
            array',
            'deep_K' => 'required|array',
            'deep_gamma' => 'required|array',
            'deep_tan' => 'required|array',
            'deep_fs' => 'required|array',
            'deep_rho' => 'required|array',
            'deep_fy' => 'required|array',
            'deep_fc' => 'required|array',
            'deep_concrete_strength_kn' => '
            required|array',
            'deep_concrete_strength_ton' => '
            required|array',
            'deep_f' => 'required|array',
            'deep_Asurface' => 'required|array',
            'deep_q' => 'required|array',
            'deep_Atip' => 'required|array',
            'deep_Qtotal' => 'required|array',
            'deep_Q_kn' => 'required|array',
            'deep_Q_ton' => 'required|array',
            'deep_is_chosen' => 'required|
            array'
        ];
    }

    /**
     * Send a custom response on authorize
     * failure
     *
     */
    public function forbiddenResponse()
    {
        return response()->view('errors.403');
    }
}

```

Listing 28: StoreDrilledShaftRequest.php

```

<?php
namespace App\Http\Requests\DrilledShaft;

use DB;
use Auth;
use Input;

```

```

use Response;
use Illuminate\Foundation\Http\FormRequest;

class StoreDrilledShaftRequest extends
    FormRequest
{
    /**
     * Determine if the user is authorized to
     * make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        if (Auth::check()) {
            return true;
        }

        $proj_id = Input::get('proj_id');

        if (Auth::user()->isInProject($proj_id
        )) {
            return true;
        }

        return false;
    }

    /**
     * Get the validation rules that apply to
     * the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'proj_id' => 'required|numeric',
            'bh_id' => 'required|array',
            'sample_id' => 'required|array',
            'num_of_samples' => 'required|
            array',
            'ds_pile_size' => 'required|array',
            'ds_Nc' => 'required|array',
            'ds_f_coeff' => 'required|array',
            'ds_fc' => 'required|array',
            'ds_fy' => 'required|array',
            'ds_steel_bar_diameter' => '
            required|array',
            'ds_fs' => 'required|array',
            'ds_tension_cap' => 'required|
            array',
            'ds_concrete_strength_kn' => '
            required|array',
            'ds_concrete_strength_ton' => '
            required|array',
            'ds_soil' => 'required|array',
            'ds_c' => 'required|array',
            'ds_Qbase' => 'required|array',
            'ds_f' => 'required|array',
            'ds_Qfriction' => 'required|array',
            'ds_Qtotal_kn' => 'required|array',
            'ds_Qtotal_ton' => 'required|array',
            'ds_is_chosen' => 'required|array'
        ];
    }

    /**
     * Send a custom response on authorize
     * failure
     *
     */
    public function forbiddenResponse()
    {
        return response()->view('errors.403');
    }
}

```

Listing 29: UpdateDrilledShaftRequest.php

```

<?php
namespace App\Http\Requests\DrilledShaft;

use DB;
use Auth;
use Input;
use Response;
use Illuminate\Foundation\Http\FormRequest;

class UpdateDrilledShaftRequest extends
    FormRequest

```

```

{
/**
 * Determine if the user is authorized to
 * make this request.
 *
 * @return bool
 */
public function authorize()
{
    if (Auth::check()) {
        return true;
    }

    $proj_id = Input::get('proj_id');

    if (Auth::user()->isInProject($proj_id)) {
        return true;
    }

    return false;
}

/**
 * Get the validation rules that apply to
 * the request.
 *
 * @return array
 */
public function rules()
{
    return [
        'proj_id' => 'required',
        'bh_id' => 'required|array',
        'sample_id' => 'required|array',
        'drilled_id' => 'required|array',
        'num_of_samples' => 'required|
            array',
        'ds_pile_size' => 'required|array',
        'ds_Nc' => 'required|array',
        'ds_f_coeff' => 'required|array',
        'ds_fc' => 'required|array',
        'ds_fy' => 'required|array',
        'ds_steel_bar_diameter' => '
            required|array',
        'ds_fs' => 'required|array',
        'ds_tension_cap' => 'required|
            array',
        'ds_concrete_strength_kn' => '
            required|array',
        'ds_concrete_strength_ton' => '
            required|array',
        'ds_soil' => 'required|array',
        'ds_c' => 'required|array',
        'ds_Qbase' => 'required|array',
        'ds_f' => 'required|array',
        'ds_Qfriction' => 'required|array',
        'ds_Qtotal_kn' => 'required|array',
        'ds_Qtotal_ton' => 'required|array',
        'ds_is_chosen' => 'required|array'
    ];
}

/**
 * Send a custom response on authorize
 * failure
 *
 */
public function forbiddenResponse()
{
    return response()->view('errors.403');
}
}

```

Listing 30: EmailRequest.php

```

<?php
namespace App\Http\Requests\Email;

use Auth;
use Input;
use Response;
use App\Project;
use Illuminate\Foundation\Http\FormRequest;

class EmailRequest extends FormRequest
{
/**
 * Determine if the user is authorized to
 * make this request.
 *
 * @return bool
 */

```

```

/**
public function authorize()
{
    $proj_id = Input::get('proj_id');
    $is_tagged = Project::findOrFail(
        $proj_id)->isTagged();

    if (Auth::check() && Auth::user()->
        isAdmin() && $is_tagged) {
        return true;
    }

    return false;
}

/**
 * Get the validation rules that apply to
 * the request.
 *
 * @return array
 */
public function rules()
{
    return [
        'to' => 'required|email',
        'subject' => 'required',
        'message' => 'string',
        'file_path' => 'string',
        'report_exists' => 'min:1|max:1'
    ];
}

/**
 * Get the validation messages that apply
 * to the request.
 *
 * @return array
 */
public function messages()
{
    return [
        'report_exists.min' => 'Report_
            has_not_been_generated.',
        'report_exists.max' => 'Report_
            has_not_been_generated.'
    ];
}

/**
 * Send a custom response on authorize
 * failure
 *
 */
public function forbiddenResponse()
{
    return response()->view('errors.403');
}
}

Listing 31: StoreProjectRequest.php
<?php
namespace App\Http\Requests\Project;

use DB;
use Auth;
use Input;
use Response;
use Illuminate\Foundation\Http\FormRequest;

class StoreProjectRequest extends FormRequest
{
/**
 * Determine if the user is authorized to
 * make this request.
 *
 * @return bool
 */
public function authorize()
{
    if (Auth::check() && Auth::user()->
        isAdmin()) {
        return true;
    }

    return false;
}

/**
 * Get the validation rules that apply to
 * the request.
 *
 * @return array
 */
public function rules()
{

```

```

        return [
            'proj_name' => 'required',
            'proj_location' => 'required',
            'proj_client' => 'required'
        ];
    }

    /**
     * Send a custom response on authorize
     * failure
     */
    public function forbiddenResponse()
    {
        return response()->view('errors.403');
    }
}

```

Listing 32: UpdateProjectRequest.php

```

<?php
namespace App\Http\Requests\Project;

use DB;
use Auth;
use Input;
use Response;
use Illuminate\Foundation\Http\FormRequest;

class UpdateProjectRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to
     * make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        if (Auth::check()) {
            if (Auth::user()->isAdmin() ||
                Auth::user()->isMember()) {
                return true;
            }
        }

        return false;
    }

    /**
     * Get the validation rules that apply to
     * the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'proj_name' => 'required',
            'proj_location' => 'required',
            'proj_client' => 'required',
            'proj_start_date' => 'required|
                date',
            'proj_end_date' => 'required|date|
                after:proj_start_date',
            'proj_date_signed' => 'required|
                date',

            // 'project_members' => '
                required-with:is_admin|array',
            // 'is_admin' => 'required-with:
                project_members'
        ];
    }

    /**
     * Send a custom response on authorize
     * failure
     */
    public function forbiddenResponse()
    {
        return response()->view('errors.403');
    }
}

```

Listing 33: StoreSampleRequest.php

```

<?php
namespace App\Http\Requests\Sample;

use DB;

```

```

use Auth;
use Input;
use Response;
use Illuminate\Foundation\Http\FormRequest;

class StoreSampleRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to
     * make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        if (Auth::check()) {
            return true;
        }

        $bh_id = Input::get('bh_id');

        if (Auth::user()->isInProject2($bh_id)
        ) {
            return true;
        }

        return false;
    }

    /**
     * Get the validation rules that apply to
     * the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'bh_id' => 'required',
            'uscs_id' => 'required',
            'sample_num' => 'required',
            'sample_from' => 'numeric',
            'sample_to' => 'numeric',
            'sample_contract_num' => 'string',
            'sample_job_num' => 'string',
            'sample_color' => 'required',
            'sample_run_length' => 'numeric|
                between:0.30,0.60',
            'sample_sample_length' => '
                required|numeric',
            'sample_rec_percent' => 'required|
                numeric|between:0,100',
            'sample_nmc_percent' => 'numeric|
                between:0,100',
            'sample_num_of_blows' => 'required
                |numeric|between:0,60',
            'sample_liquid_limit' => 'required
                |numeric|between:0,100',
            'sample_plastic_limit' => '
                required|numeric|between
                :0,100',
            'sample_plasticity_index' => '
                required|numeric|between
                :0,100',
            'sample_plasticity_remark' => '
                string',
            'sample_tester' => 'string',
            'sample_type_of_sampling' => '
                string',
            'sample_test_date' => 'date'
        ];
    }

    /**
     * Send a custom response on authorize
     * failure
     */
    public function forbiddenResponse()
    {
        return response()->view('errors.403');
    }
}

```

Listing 34: UpdateSampleRequest.php

```

<?php
namespace App\Http\Requests\Sample;

use DB;
use Auth;
use Input;
use Response;
use Illuminate\Foundation\Http\FormRequest;

class UpdateSampleRequest extends FormRequest

```

```

{
/**
 * Determine if the user is authorized to
 * make this request.
 *
 * @return bool
 */
public function authorize()
{
    if (Auth::check()) {
        return true;
    }

    $bh_id = Input::get('bh_id');

    if (Auth::user()->isInProject2($bh_id)
    ) {
        return true;
    }

    return false;
}

/**
 * Get the validation rules that apply to
 * the request.
 *
 * @return array
 */
public function rules()
{
    return [
        'bh_id' => 'required',
        'uscs_id' => 'required',
        'sample_num' => 'required',
        'sample_from' => 'required|numeric',
        'sample_to' => 'required|numeric',
        'sample_contract_num' => 'string',
        'sample_job_num' => 'string',
        'sample_color' => 'required',
        'sample_run_length' => 'required|
        numeric|between:0.30,0.60',
        'sample_sample_length' => '
        required|numeric',
        'sample_rec_percent' => 'required|
        numeric|between:0,100',
        'sample_nmc_percent' => 'numeric|
        between:0,100',
        'sample_num_of_blow' => 'required
        |numeric|between:0,60',
        'sample_liquid_limit' => 'required
        |numeric|between:0,100',
        'sample_plastic_limit' => '
        required|numeric|between
        :0,100',
        'sample_plasticity_index' => '
        required|numeric|between
        :0,100',
        'sample_plasticity_remark' => '
        string',
        'sample_tester' => 'string',
        'sample_type_of_sampling' => '
        string',
        'sample_test_date' => 'date'
    ];
}

/**
 * Send a custom response on authorize
 * failure
 */
public function forbiddenResponse()
{
    return response()->view('errors.403');
}
}

```

Listing 35: Section.php

```

<?php
namespace App\Http\Requests\Section;

use DB;
use Auth;
use Input;
use Response;
use Illuminate\Foundation\Http\FormRequest;

class StoreSectionRequest extends FormRequest
{
/**
 * Determine if the user is authorized to
 * make this request.
 *
 *
 */

```

```

* @return bool
 */
public function authorize()
{
    if (Auth::check()) {
        return true;
    }

    $proj_id = Input::get('proj_id');

    if (Auth::user()->isInProject($proj_id)
    ) {
        return true;
    }

    return false;
}

/**
 * Get the validation rules that apply to
 * the request.
 *
 * @return array
 */
public function rules()
{
    return [
        'proj_id' => 'required',
        'section_key' => 'required'
    ];
}

/**
 * Send a custom response on authorize
 * failure
 */
public function forbiddenResponse()
{
    return response()->view('errors.403');
}
}

Listing 36: UpdateSectionRequest.php
<?php
namespace App\Http\Requests\Section;

use DB;
use Auth;
use Input;
use Response;
use Illuminate\Foundation\Http\FormRequest;

class UpdateSectionRequest extends FormRequest
{
/**
 * Determine if the user is authorized to
 * make this request.
 *
 * @return bool
 */
public function authorize()
{
    if (Auth::check()) {
        return true;
    }

    $proj_id = Input::get('proj_id');

    if (Auth::user()->isInProject($proj_id)
    ) {
        return true;
    }

    return false;
}

/**
 * Get the validation rules that apply to
 * the request.
 *
 * @return array
 */
public function rules()
{
    return [
        'section_text' => 'required'
    ];
}

/**
 * Send a custom response on authorize
 * failure
 *
 */

```

```

    public function forbiddenResponse()
    {
        return response()->view('errors.403');
    }
}

```

Listing 37: StoreShallowRequest.php

```

<?php
namespace App\Http\Requests\Shallow;

use DB;
use Auth;
use Input;
use Response;
use Illuminate\Foundation\Http\FormRequest;

class StoreShallowRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to
     * make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        if (Auth::check()) {
            return true;
        }

        $proj_id = Input::get('proj_id');

        if (Auth::user()->isInProject($proj_id)) {
            return true;
        }

        return false;
    }

    /**
     * Get the validation rules that apply to
     * the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'proj_id' => 'required|numeric',
            'bh_id' => 'required|array',
            'sample_id' => 'required|array',
            'num_of_samples' => 'required|
                array',
            'shallow_B' => 'required|array',
            'shallow_phi' => 'required|array',
            'shallow_Nc' => 'required|array',
            'shallow_Ngamma' => 'required|
                array',
            'shallow_Nq' => 'required|array',
            'shallow_gammal' => 'required|
                array',
            'shallow_gamma2' => 'required|
                array',
            'shallow_fs' => 'required|array',
            'shallow_soil' => 'required|array',
            'shallow_c' => 'required|array',
            'shallow_qult' => 'required|array',
            'shallow_q' => 'required|array',
            'shallow_is_chosen' => 'required|
                array'
        ];
    }

    /**
     * Send a custom response on authorize
     * failure
     */
    public function forbiddenResponse()
    {
        return response()->view('errors.403');
    }
}

```

Listing 38: UpdateShallowRequest.php

```

<?php
namespace App\Http\Requests\Shallow;

```

```

use DB;
use Auth;
use Input;
use Response;
use Illuminate\Foundation\Http\FormRequest;

class UpdateShallowRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to
     * make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        if (Auth::check()) {
            return true;
        }

        $proj_id = Input::get('proj_id');

        if (Auth::user()->isInProject($proj_id)) {
            return true;
        }

        return false;
    }

    /**
     * Get the validation rules that apply to
     * the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'proj_id' => 'required',
            'bh_id' => 'required|array',
            'sample_id' => 'required|array',
            'shallow_id' => 'required|array',
            'num_of_samples' => 'required|
                array',
            'shallow_B' => 'required|array',
            'shallow_phi' => 'required|array',
            'shallow_Nc' => 'required|array',
            'shallow_Ngamma' => 'required|
                array',
            'shallow_Nq' => 'required|array',
            'shallow_gammal' => 'required|
                array',
            'shallow_gamma2' => 'required|
                array',
            'shallow_fs' => 'required|array',
            'shallow_soil' => 'required|array',
            'shallow_c' => 'required|array',
            'shallow_qult' => 'required|array',
            'shallow_q' => 'required|array',
            'shallow_is_chosen' => 'required|
                array'
        ];
    }

    /**
     * Send a custom response on authorize
     * failure
     */
    public function forbiddenResponse()
    {
        return response()->view('errors.403');
    }
}

```

Listing 39: Request.php

```

<?php
namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

abstract class Request extends FormRequest
{
    //
}

```

Listing 40: Kernel.php

```

<?php

```

```

namespace App\Http;

use Illuminate\Foundation\Http\Kernel as
    HttpKernel;

class Kernel extends HttpKernel
{
    /**
     * The application's global HTTP
     * middleware stack.
     */
    /**
     * @var array
     */
    protected $middleware = [
        \Illuminate\Foundation\Http\Middleware\
            \CheckForMaintenanceMode::class,
        \App\Http\Middleware\EncryptCookies::
            class,
        \Illuminate\Cookie\Middleware\
            AddQueuedCookiesToResponse::class,
        \Illuminate\Session\Middleware\
            StartSession::class,
        \Illuminate\View\Middleware\
            ShareErrorsFromSession::class,
        \App\Http\Middleware\VerifyCsrfToken::
            class,
    ];

    /**
     * The application's route middleware.
     */
    /**
     * @var array
     */
    protected $routeMiddleware = [
        'auth' => \App\Http\Middleware\
            Authenticate::class,
        'auth.basic' => \Illuminate\Auth\
            Middleware\
            AuthenticateWithBasicAuth::class,
        'guest' => \App\Http\Middleware\
            RedirectIfAuthenticated::class
    ];
}

<?php
/*
|-----
| Application Routes
|-----
|
| Here is where you can register all of the
| routes for an application.
| It's a breeze. Simply tell Laravel the URIs
| it should respond to
| and give it the controller to call when that
| URI is requested.
|
| */

Route::get('/', function () {
    if(Auth::check()){
        if(Auth::user()->isPending()) {
            Auth::logout();
            return redirect('auth/login')->
                with('info', 'Your account
                needs admin confirmation to
                be able to login. ');
        }
        else if(Auth::user()->isDisabled()) {
            Auth::logout();
            return redirect('auth/login')->
                with('info', 'Your account is
                disabled. ');
        }
        return view('welcome');
    }
    return redirect('auth/login');
});

// Activity routes
Route::get('activities', ['as' => 'activities.
index', 'uses' =>
    'ActivityController@index']);
Route::get('notifications', ['as' =>
    'activities.notifications', 'uses' =>
    'ActivityController@showNotifications']);

// Project routes
Route::resource('projects', 'ProjectController
');
Route::get('archive-{id}', ['as' => 'projects.
archive', 'uses' =>
    'ProjectController@archive']);
Route::get('tag-{id}', ['as' => 'projects.tag',
    'uses' => 'ProjectController@tag']);
Route::get('active-{id}', ['as' => 'projects.
viewActive', 'uses' =>
    'ProjectController@viewActive']);
Route::get('archived', ['as' => 'projects.
viewArchived', 'uses' =>
    'ProjectController@viewArchived']);
Route::get('compiled-report-{id}', ['as' =>
    'projects.viewCompiledReport', 'uses' =>
    'ProjectController@viewCompiledReport']);

// User routes
Route::resource('users', 'UserController');
Route::post('users/accept', ['as' => 'users.
acceptUser', 'uses' =>
    'UserController@acceptUser']);
Route::post('users/disable', ['as' => 'users.
disableUser', 'uses' =>
    'UserController@disableUser']);
Route::post('users/change-position', ['as' =>
    'users.changeUserPosition', 'uses' =>
    'UserController@changeUserPosition']);

// Borehole routes
Route::resource('boreholes', '
    BoreholeController', ['except' => ['
    create']]);
Route::get('projects/{id}/create', ['as' =>
    'boreholes.create', 'uses' =>
    'BoreholeController@create']);

// Sample routes
Route::resource('samples', 'SampleController',
    ['except' => ['create']]);
Route::get('samples/{id}/create', ['as' =>
    'samples.create', 'uses' =>
    'SampleController@create']);

// Email routes
Route::get('emails/{id}/create', ['as' =>
    'emails.create', 'uses' =>
    'EmailController@create']);
Route::post('emails/send', ['as' => 'emails.
send', 'uses' => 'EmailController@send']);

// Shallow (foundation) routes
Route::resource('shallows', 'ShallowController
', ['except' => ['create', 'index', 'show
']]);
Route::get('shallows/{id}/create', ['as' =>
    'shallows.create', 'uses' =>
    'ShallowController@create']);
Route::get('shallows/{id}', ['as' => 'shallows.
index', 'uses' =>
    'ShallowController@index']);
Route::get('shallows/{id}/delete-create', ['as'
=> 'shallows.deleteAndCreate', 'uses'
=> 'ShallowController@deleteAndCreate']);

// Drilled shaft (foundation) routes
Route::resource('drilledshafts', '
    DrilledShaftController', ['except' => ['
    create', 'index', 'show']]);
Route::get('drilledshafts/{id}/create', ['as'
=> 'drilledshafts.create', 'uses' =>
    'DrilledShaftController@create']);
Route::get('drilledshafts/{id}', ['as' =>
    'drilledshafts.index', 'uses' =>
    'DrilledShaftController@index']);
Route::get('drilledshafts/{id}/delete-create',
    ['as' => 'drilledshafts.deleteAndCreate',
    'uses' =>
    'DrilledShaftController@deleteAndCreate']);

// Deep (foundation) routes
Route::resource('deeps', 'DeepController', ['
except' => ['create', 'index', 'show']]);
Route::get('deeps/{id}/create', ['as' =>
    'deeps.create', 'uses' =>
    'DeepController@create']);
Route::get('deeps/{id}', ['as' => 'deeps.index',
    'uses' => 'DeepController@index']);
Route::get('deeps/{id}/delete-create', ['as'
=> 'deeps.deleteAndCreate', 'uses' =>
    'DeepController@deleteAndCreate']);

// Section routes
Route::resource('sections', 'SectionController
', ['except' => ['create']]);
Route::post('sections/create', ['as' =>
    'sections.create', 'uses' =>
    'SectionController@create']);

// Subreport routes

```

Listing 41: routes.php

```

Route::resource('subreports', '
    SubreportController');
Route::post('subreports/{id}/store-image', '
    SubreportController@storeImage');

// GSA routes
Route::get('subreports/{id}/createGSA', ['as' => '
    subreports.createGSA', 'uses' => '
    SubreportController@createGSA']);
Route::get('subreports/{id}/showGSA', ['as' => '
    subreports.showGSA', 'uses' => '
    SubreportController@showGSA']);
Route::get('subreports/{id}/editGSA', ['as' => '
    subreports.editGSA', 'uses' => '
    SubreportController@editGSA']);
Route::post('subreports/storeGSA', ['as' => '
    subreports.storeGSA', 'uses' => '
    SubreportController@storeGSA']);

// Data point routes
Route::resource('datapoints', '
    DataPointController', ['except' => ['
    create', 'index', 'show']]);
Route::get('datapoints/{id}/create', ['as' => '
    datapoints.create', 'uses' => '
    DataPointController@create']);
Route::get('datapoints/{id}', ['as' => '
    datapoints.index', 'uses' => '
    DataPointController@index']);
Route::get('datapoints/{id}/delete-create', ['
    as' => 'datapoints.deleteAndCreate', '
    uses' => '
    DataPointController@deleteAndCreate']);

// LPTR routes
Route::get('subreports/{id}/createLPTR', ['as' => '
    subreports.createLPTR', 'uses' => '
    SubreportController@createLPTR']);
Route::get('subreports/{id}/showLPTR', ['as' => '
    subreports.showLPTR', 'uses' => '
    SubreportController@showLPTR']);

// Limit routes
Route::resource('limits', 'LimitController', [
    'except' => ['create', 'index', 'show']]);
Route::get('limits/{id}/create', ['as' => '
    limits.create', 'uses' => '
    LimitController@create']);
Route::get('limits/{id}', ['as' => 'limits.
    index', 'uses' => 'LimitController@index'
    ]);

// BL routes
Route::get('subreports/{id}/createBL', ['as' => '
    subreports.createBL', 'uses' => '
    SubreportController@createBL']);
Route::get('subreports/{id}/showBL', ['as' => '
    subreports.showBL', 'uses' => '
    SubreportController@showBL']);

// SP routes
Route::get('subreports/{id}/createSP', ['as' => '
    subreports.createSP', 'uses' => '
    SubreportController@createSP']);
Route::get('subreports/{id}/showSP', ['as' => '
    subreports.showSP', 'uses' => '
    SubreportController@showSP']);

// Authentication routes...
Route::get('auth/login', ['as' => 'auth.login',
    'uses' => 'Auth\AuthController@getLogin'
    ]);
Route::post('auth/login', ['as' => 'auth.login',
    'uses' => 'Auth\
    AuthController@postLogin']);
Route::get('auth/logout', ['as' => 'auth.
    logout', 'uses' => 'Auth\
    AuthController@getLogout']);

// Registration routes...
Route::get('auth/register', ['as' => 'auth.
    register', 'uses' => 'Auth\
    AuthController@getRegister']);
Route::post('auth/register', ['as' => 'auth.
    register', 'uses' => 'Auth\
    AuthController@postRegister']);

Route::controllers([
    'password' => 'Auth\PasswordController',
    ]);

/*Route::group(array('prefix' => 'admin'),
    function() {
    // main page for the admin section (app/
    views/admin/dashboard.blade.php)
    Route::get('/', function() {
        return View::make('admin.dashboard');
    });
}

```

```

// subpage for the posts found at /admin/
posts (app/views/admin/posts.blade.
php)
Route::get('posts', function() {
    return View::make('admin.posts');
});

// subpage to create a post found at /
admin/posts/create (app/views/admin/
posts-create.blade.php)
Route::get('posts/create', function() {
    return View::make('admin.posts-create
    ');
});
})*

```

Listing 42: Helper.php

```

<?php
namespace App\CustomClasses;

use DB;
use DateTime;
use App\DataPoint;
use App\Limit;
use App\Project;
use App\Sample;

class Helper {

    public static function getValue($old_value
        , $original_value) {
        if($old_value == null)
            $value = $original_value;
        else
            $value = $old_value;

        return $value;
    }

    public static function getNotifications()
    {
        $pending_users = DB::table('users')->
            where('status', 0)->orderBy('
            created_at', 'desc')->get();
        $tagged_projects = DB::table('projects'
            )->where('proj-status', 1)->
            orderBy('created_at', 'desc')->
            get();

        return array($pending_users,
            $tagged_projects);
    }

    public static function getEngineers(
        $proj_id) {
        $engineers = DB::select('SELECT_u.id, _
            u.firstname, _u.lastname, _u.image_
            FROM_users_u_LEFT_JOIN_
            project_engineers_pe_ON_u.id _=pe
            .enr_id _WHERE_pe.proj_id _=?', [
            $proj_id]);
        return $engineers;
    }

    public static function getMembers() {
        $members = DB::table('users')->select(
            'id', 'firstname', 'lastname', '
            image')->where('status', 1)->get
            ();
        return $members;
    }

    public static function getUserName($id) {
        $user = DB::table('users')->select('
            firstname', 'lastname')->whereId(
            $id)->first();
        $name = $user->firstname . "-" . $user
            ->lastname;
        return $name;
    }

    public static function getProjects(
        $sengr_id) {
        $projects = DB::select('SELECT_p.id, _p
            .proj_name, _p.proj_client _FROM_
            projects_p_LEFT_JOIN_
            project_engineers_pe_ON_p.id _=pe
            .proj_id _WHERE_pe.enr_id _=?', [
            $sengr_id]);
        return $projects;
    }

    public static function getSectionText(
        $section_id) {

```

```

$section = DB::select('SELECT se_b.
    section_text FROM section_bodies_
    se_b LEFT JOIN sections_se ON se_
    id = se_b.section_id WHERE se_b.
    section_id = ? ', [$section_id]);
return $section;
}

public static function getBoreholeNum(
    $bh_id) {
    $bh_num = DB::table('boreholes')->
        where('id', $bh_id)->select('
            bh_num')->first();
    return $bh_num->bh_num;
}

public static function getSectionName(
    $section_key) {
    switch($section_key) {
        case 0: $section_name = "Title_
            Page"; break;
        case 1: $section_name = "
            Introduction"; break;
        case 2: $section_name = "Drilling_
            Procedure"; break;
        case 3: $section_name = "
            Laboratory_Test"; break;
        case 4: $section_name = "Site_
            Geology"; break;
        case 5: $section_name = "Findings"
            ; break;
        case 6: $section_name = "
            Conclusions_and_
            Recommendations"; break;
        case 7: $section_name = "
            Limitations/Comments"; break;
        case 8: $section_name = "Other_
            Section"; break;
        default: $section_name = ""; break
    }
    return $section_name;
}

public static function getSectionKey(
    $section_name) {
    switch($section_name) {
        case "Title_Page": $section_key =
            0; break;
        case "Introduction": $section_key
            = 1; break;
        case "Drilling_Procedure":
            $section_key = 2; break;
        case "Laboratory_Test":
            $section_key = 3; break;
        case "Site_Geology": $section_key
            = 4; break;
        case "Findings": $section_key = 5;
            break;
        case "Conclusions_and_
            Recommendations":
            $section_key = 6; break;
        case "Limitations/Comments":
            $section_key = 7; break;
        case "Other_Section": $section_key
            = 8; break;
        default: $section_key = -1; break;
    }
    return $section_key;
}

public static function getSubreportName(
    $sb_key) {
    switch($sb_key) {
        case 0: $sb_name = "Grain_Size_
            Analysis"; break;
        case 1: $sb_name = "Liquid_and_
            Plastic_Limits_Test_Report";
            break;
        case 2: $sb_name = "Borehole_Log";
            break;
        case 3: $sb_name = "Soil_Profile";
            break;
        default: $sb_name = ""; break;
    }
    return $sb_name;
}

public static function getSectionTemplate(
    $section_key, $proj_id) {
    $project = Project::findOrFail(
        $proj_id);
    $boreholes = Project::find($project->
        id)->boreholes;

    $stext = "<div_style='text-align:_
        justify;'><span_style='font-size:_
        medium;line-height:_1.471;'><b>
            " . strtoupper($section_key) . "
            </b></div></span></div><br/>";

    switch($section_key) {
        case "Title_Page":
            $stext = "<div_style='text-
                align:_center;'><b>
                    GEOTECHNICAL_
                    INVESTIGATION_REPORT</b>
                    </div><br/>
                    <div_style='text-align:_center_
                    ;'><b>FOR_THE_PROPOSED_ " .
                    strtoupper($project->proj_name) .
                    "</b></div><br/><br/>
                    <div_style='text-align:_center_
                    ;'><b>PROJECT_LOCATION:</b></div><br/>
                    <div_style='text-align:_center_
                    ;'><b> " .
                    strtoupper($project->
                    proj_location) .
                    "</b></div><br/><br/>
                    <div_style='text-align:_center_
                    ;'><b>AS_PREPARED_FOR</b></div><br/>
                    <div_style='text-align:_center_
                    ;'><b> " .
                    strtoupper($project->
                    proj_client) .
                    "</b></div><br/><br/><br/>
                    <div_style='text-align:_center_
                    ;'><b>PREPARED_BY</b></div><br/>
                    <div_style='text-align:_center_
                    ;'><b>ENGR. RESTITUTO.R. ABAD.JR.</b></
                    div>
                    <div_style='text-align:_center_
                    ;'><b>CIVIL/FOUNDATION/STRUCTURAL_ENGR.</
                    b></div>
                    <div_style='text-align:_center_
                    ;'>PRC_62817</div>
                    <div_style='text-align:_center_
                    ;'>STRUCTURAL_ENG'_G_SPECIALIST_142</div><
                    br/><br/><br/>
                    <div_style='text-align:_center_
                    ;'><b>Member:_The_Institution_of_
                    Specialist_Structural_Engr._of_the_
                    Philippines_(ISSEP),_Inc.</b></div><br/><
                    br/><br/><br/><br/>
                    <div_style='text-align:_center_
                    ;'>Mobile_Nos.:_09235000984_-/_09985326206
                    _/_09062838860</div>
                    <div_style='text-align:_center_
                    ;'>Email_Address:_rabadjr@yahoo.com</div>
                    <div_style='text-align:_center_
                    ;'>Website:_laglobaledcc.com</div>
                    <div_style='text-align:_center_
                    ;'>Office_Address:_336_Majestic_st.,_
                    Monte_Vista_Park_subd.,_Cainta,_Rizal</
                    div>";
            break;
        case "Introduction":
            $myDateTime = DateTime::
                createFromFormat('Y-m-d',
                    $project->
                    proj_start_date);
            $start_date = $myDateTime->
                format('F.d,_Y');
            $myDateTime = DateTime::
                createFromFormat('Y-m-d',
                    $project->proj_end_date);
            $end_date = $myDateTime->
                format('F.d,_Y');
            $stext = $stext . "<blockquote_
                style='margin:_0_0_0_40px_
                ;border:_none;padding:_0_
                0px;'><div_style='text-
                align:_justify;'><span_
                style='font-size:_medium;
                line-height:_1.471;'>The
                soil_drilling_and_
                sampling_for_the_
                geotechnical_
                investigation_at_the_site_
                of_the_$project->
                proj_name,_was_undertaken_
                by_LA_Global_EDCC._The_
                evaluation_of_the_field_
                investigation_results_
                were_performed_by_Engr._
                Restituto.R._Abad_Jr._
                The_soil_drilling_and_
                sampling_were_conducted_
                inclusive_of_the_dates_
                $start_date_to_$end_date
                .</span></div></
                blockquote>";
            break;
        case "Drilling_Procedure":
            // num of boreholes,
            bh_max_depth, boring
            method, interval,
            bh_hammer_weight,

```



```

        bh_type_of_sampling
    $text = $text . "<blockquote
    style='margin: 0 0 0 40px
    ;_border: none;_padding: 0
    0px;_><div style='text-
    align: justify;_><span
    style='font-size: medium;
    _line-height: 1.471;_>&lt;
    ;_boreholes_&gt;_were_
    drilled_for_a_depth_of_&
    lt;_depth_&gt;_m_&lt;
    ;_boringmethod_&gt;_was_used
    _in_the_drilling_._Soil_
    samples_were_taken_at_an_
    interval_of_0.3m_from_
    ground_surface.</span></
    div></blockquote><br/>
    <blockquote style='margin: 0 0
    0 40px;_border: none;_padding: 0px;_><
    div style='text-align: justify;_><span
    style='font-size: medium;_line-height: 1.
    471;_> Standard Penetration Tests (SPT)
    for the two boreholes were conducted in
    order to obtain information on the soil
    bearing capacity in relation to the
    number of blows, as required to drive the
    2-inch outside diameter split spoon
    sampler which is being hammered
    continuously by a &lt;_hammerweight_&gt;_kg_
    weight. The number of blows required to
    drive the sampler through the 12-inch
    penetration are recorded as the N-value
    of the soil layer being tested.</span></
    div></blockquote><br/>
    <blockquote style='margin: 0 0
    0 40px;_border: none;_padding: 0px;_><
    div style='text-align: justify;_><span
    style='font-size: medium;_line-height: 1.
    471;_>&lt;_typeofsampling_&gt;_is_
    employed_in_hard_soil_formation_to_
    advance the drill rod and to obtain rock
    or hard soil samples on which laboratory
    tests are performed to establish their
    physical characteristics.</span></div></
    blockquote>";
    break;
    case "Laboratory Test":
    $text = $text . "<blockquote
    style='margin: 0 0 0 40px
    ;_border: none;_padding: 0
    0px;_><div style='text-
    align: justify;_><span
    style='font-size: medium;
    _line-height: 1.471;_> All
    laboratory tests were
    conducted in accordance
    with ASTM Standard
    Procedures.</span></div
    ></blockquote>";
    break;
    case "Site Geology":
    $text = $text . "<blockquote
    style='margin: 0 0 0 40px
    ;_border: none;_padding: 0
    0px;_><div style='text-
    align: justify;_><font
    size='3'><span style='
    line-height: 23.536px;_>
    Historical geologic
    background of Metro
    Manila and immediate
    vicinities were
    particularly described by
    reference #1, stating,</
    span></font></div></
    blockquote><br/>
    <blockquote style='margin: 0 0
    0 40px;_border: none;_padding: 0px;_><
    div style='text-align: justify;_><font
    size='3'><span style='line-height: 23.536
    px;_><i>Metro Manila and its immediate
    vicinities used to be a submerged area at
    one time in the geologic past.
    Intermittent volcanic activities followed
    after which volcanic materials were
    deposited.</i></span></font></div></
    blockquote><br/>
    <blockquote style='margin: 0 0
    0 40px;_border: none;_padding: 0px;_><
    div style='text-align: justify;_><font
    size='3'><span style='line-height: 23.536
    px;_><i>The islands of the Philippines
    were developed through volcanic activity
    in the sea floor millions of years ago.
    The Philippines first started to form in
    the mid-Oligocene, around 30 million
    years ago, when the land of mass that is
    now Australia started to drift northward,
    causing an uplift in the Pacific sea
    floor. At the same time, a part of the
    Asian continental shelf drifted southward,
    creating the landform which now
    consists of modern day Palawan and
    Mindoro Islands. The years of the Miocene
    epoch saw a rise in a series of volcanic
    islands in shallow waters, and further
    subduction of converging plates in the
    Pacific formed other larger islands such
    as Visayas and Mindanao. It was only
    after the Pliocene that the smaller
    islands emerged above sea level, forming
    the chain of 7,000 islands that make up
    the country today. The tectonic movements
    caused complex deformation in the crust
    and developed movement along with the
    rest of the Philippine Sea Plate caused
    the formation of large thrust components
    in the northern Philippines (Rimando et
    al., 2006).</i></span></font></div></
    blockquote><br/>
    <blockquote style='margin: 0 0
    0 40px;_border: none;_padding: 0px;_><
    div style='text-align: justify;_><font
    size='3'><span style='line-height: 23.536
    px;_><i>The city of Manila rests upon
    alluvial deposits from Pasig River, which
    is a tidal estuary that connects Laguna
    de Bay to Manila Bay. The direction of
    flow of water depends on the relative
    water levels of Laguna de Bay and Manila
    Bay.</i></span></font></div></blockquote>
    <br/>
    <blockquote style='margin: 0 0
    0 40px;_border: none;_padding: 0px;_><
    div style='text-align: justify;_><span
    style='line-height: 23.536px;_><font size
    =1'>Reference 1, Soil report for 4-
    Storey Sto. Nino Church, Pasig Green Park
    Village, Mangahan, Pasig City</font></
    span></div></blockquote>
    <blockquote style='margin: 0 0
    0 40px;_border: none;_padding: 0px;_><
    div style='text-align: justify;_><span
    style='line-height: 23.536px;_><font size
    =1'>Reference 2, Essentials of Soil
    Mechanics and Foundations, by David F.
    McCarthy</font></span></div></blockquote>
    <blockquote style='margin: 0 0
    0 40px;_border: none;_padding: 0px;_><
    div style='text-align: justify;_><span
    style='line-height: 23.536px;_><font size
    =1'>Reference 3, Foundation Analysis and
    Design, by Joseph E. Bowles</font></span
    ></div></blockquote>
    <blockquote style='margin: 0 0
    0 40px;_border: none;_padding: 0px;_><
    div style='text-align: justify;_><span
    style='line-height: 23.536px;_><font size
    =1'>Reference 4, Pile Foundation,
    Analysis, and Design</font></span></div
    ></blockquote>
    <blockquote style='margin: 0 0
    0 40px;_border: none;_padding: 0px;_><
    div style='text-align: justify;_><span
    style='line-height: 23.536px;_><font size
    =1'>Reference 5, National Structural
    Code of the Philippines, 6th ed.</font></
    span></div></blockquote>";
    break;
    case "Findings":
    $text = $text . "<blockquote
    style='margin: 0 0 0 40px
    ;_border: none;_padding: 0
    0px;_><div style='text-
    align: justify;_><span
    style='font-size: medium;
    _line-height: 1.471;_> The
    location of the
    boreholes are reflected
    in the attachment with
    the accompanying soil
    profile plan and borehole
    logs.</span></div></
    blockquote><br/>";
    foreach ($boreholes as $bh) {
    $text = $text . "<
    blockquote style='
    margin: 0 0 0 40px;_
    border: none;_padding:
    0px;_><div style='
    text-align: justify
  
```



```

<?php
namespace App\CustomClasses;

use DB;
use App\Borehole;
use App\Limit;
use App\Project;
use App\Sample;
use App\Section;
use App\Subreport;

class TemplateHelper {

    public static function getVariables(
        $proj_id) {
        $boreholes = Project::find($proj_id)->
            boreholes;
        $samples = Sample::join('boreholes', '
            samples.bh_id', '=', 'boreholes.
            id')
            ->join('uscs', 'samples.uscs_id', '
            =', 'uscs.id')
            ->where('boreholes.proj_id',
                $proj_id)
            ->select('samples.id', 'samples.
                bh_id', 'samples.uscs_id', '
                uscs.uscs_description', '
                samples.sample_num', 'samples.
                sample_from', 'samples.
                sample_to', 'samples.
                sample_job_num', 'samples.
                sample_test_date', 'samples.
                sample_color', 'samples.
                sample_nmc_percent', 'samples.
                sample_liquid_limit', '
                samples.sample_plastic_limit'
                , 'samples.
                sample_plasticity_index')
            ->get();
        $sections = Section::where('proj_id',
            $proj_id)
            ->orderBy('section_key', 'asc')
            ->get();
        $section_bodies = Section::join('
            section_bodies', 'sections.id', '
            =', 'section_bodies.section_id')
            ->where('sections.proj_id',
                $proj_id)
            ->orderBy('sections.section_key',
                'asc')
            ->get();
        $subreports = Subreport::where('
            proj_id', $proj_id)
            ->orderBy('sb_key', 'asc')
            ->get();

        $sb = $subreports->where('sb_key', 1)
            ->first();
        $matchThese = ['subreport_id' => $sb->
            id, 'lim_test' => 0];
        $liquid_limits = Limit::where(
            $matchThese->get());

        $matchThese = ['subreport_id' => $sb->
            id, 'lim_test' => 1];
        $plastic_limits = Limit::where(
            $matchThese->get());

        return array($boreholes, $samples,
            $sections, $section_bodies,
            $subreports, $liquid_limits,
            $plastic_limits);
    }
}
?>

```

Listing 44: AppServiceProvider.php

```

<?php
namespace App\Providers;

use Illuminate\Support\ServiceProviders;

class AppServiceProvider extends
    ServiceProvider
{
    /**
     * Bootstrap any application services.
     *
     * @return void
     */
    public function boot()
    {
        //
    }
}

```

```

}

/**
 * Register any application services.
 *
 * @return void
 */
public function register()
{
    //
}

```

Listing 45: AuthServiceProvider.php

```

<?php
namespace App\Providers;

use Auth;
use Illuminate\Contracts\Auth\Access\Gate as
    GateContract;
use Illuminate\Foundation\Support\Providers\
    AuthServiceProvider as ServiceProvider;

class AuthServiceProvider extends
    ServiceProvider
{
    /**
     * The policy mappings for the application
     *
     * @var array
     */
    protected $policies = [
        'App\Model' => 'App\Policies\
            ModelPolicy',
    ];

    /**
     * Register any application authentication
     * / authorization services.
     *
     * @param \Illuminate\Contracts\Auth\
        Access\Gate $gate
     * @return void
     */
    public function boot(GateContract $gate)
    {
        parent::registerPolicies($gate);

        // Projects
        $gate->define('show-project', function
            ($user, $project) {
            if($project->isArchived()) {
                return true;
            }
            return $user->isInProject($project
                ->id);
        });

        $gate->define('index-project',
            function ($user) {
            return $user->isAdmin();
        });

        $gate->define('edit-project', function
            ($user, $project) {
            if($project->isArchived()) {
                return false;
            }
            return $user->isInProject($project
                ->id);
        });

        $gate->define('destroy-project',
            function ($user) {
            return $user->isAdmin();
        });

        $gate->define('archive-project',
            function ($user) {
            return $user->isAdmin();
        });

        $gate->define('tag-project', function
            ($user, $project) {
            return $user->isInProject($project
                ->id);
        });

        $gate->define('view-active-project',
            function ($user, $user_id) {
            return $user->id == $user_id;
        });

        $gate->define('view-archived-project',

```

```

        function ($user) {
            return $user->isAdmin() || $user->
                isAdminMember();
        });
    $gate->define('view-compiled-report',
        function ($user, $project) {
            if($project->isArchived()) {
                return true;
            }
            return $user->isInProject($project
                ->id);
        });
    // Boreholes
    $gate->define('borehole', function (
        $user, $project) {
            if($project->isArchived()) {
                return false;
            }
            return $user->isInProject($project
                ->id);
        });
    $gate->define('show-borehole',
        function ($user, $project) {
            if($project->isArchived()) {
                return true;
            }
            return $user->isInProject($project
                ->id);
        });
    // Samples
    $gate->define('sample', function (
        $user, $borehole) {
            if($borehole->project->isArchived
                ()) {
                return false;
            }
            return $user->isInProject2(
                $borehole->id);
        });
    // Data points
    $gate->define('data-point', function (
        $user, $project) {
            if($project->isArchived()) {
                return false;
            }
            return $user->isInProject($project
                ->id);
        });
    $gate->define('index-data-point',
        function ($user, $project) {
            if($project->isArchived()) {
                return true;
            }
            return $user->isInProject($project
                ->id);
        });
    // Limits
    $gate->define('limit', function ($user
        , $project) {
            if($project->isArchived()) {
                return false;
            }
            return $user->isInProject($project
                ->id);
        });
    $gate->define('index-limit', function
        ($user, $project) {
            if($project->isArchived()) {
                return true;
            }
            return $user->isInProject($project
                ->id);
        });
    // Subreports
    $gate->define('subreport', function (
        $user, $project) {
            if($project->isArchived()) {
                return false;
            }
            return $user->isInProject($project
                ->id);
        });
    $gate->define('show-subreport',
        function ($user, $project) {
            if($project->isArchived()) {
                return true;
            }
            return $user->isInProject($project
                ->id);
        });
    });
    // Deeps
    $gate->define('deep', function ($user,
        $project) {
            if($project->isArchived()) {
                return false;
            }
            return $user->isInProject($project
                ->id);
        });
    $gate->define('index-deep', function (
        $user, $project) {
            if($project->isArchived()) {
                return true;
            }
            return $user->isInProject($project
                ->id);
        });
    // Drilled Shafts
    $gate->define('drilled-shaft',
        function ($user, $project) {
            if($project->isArchived()) {
                return false;
            }
            return $user->isInProject($project
                ->id);
        });
    $gate->define('index-drilled-shaft',
        function ($user, $project) {
            if($project->isArchived()) {
                return true;
            }
            return $user->isInProject($project
                ->id);
        });
    // Shallows
    $gate->define('shallow', function (
        $user, $project) {
            if($project->isArchived()) {
                return false;
            }
            return $user->isInProject($project
                ->id);
        });
    $gate->define('index-shallow',
        function ($user, $project) {
            if($project->isArchived()) {
                return true;
            }
            return $user->isInProject($project
                ->id);
        });
    // Sections
    $gate->define('section', function (
        $user, $project) {
            if($project->isArchived()) {
                return false;
            }
            return $user->isInProject($project
                ->id);
        });
    $gate->define('show-section', function
        ($user, $project) {
            if($project->isArchived()) {
                return true;
            }
            return $user->isInProject($project
                ->id);
        });
    // Users
    $gate->define('manage-user', function
        ($user) {
            return $user->isAdmin();
        });
    $gate->define('edit-user', function (
        $user, $user_id) {
            return $user->id == $user_id;
        });
    $gate->define('show-user', function (
        $user) {
            return $user->isAdmin() || $user->
                isAdminMember();
        });
    // Emails
    $gate->define('email', function ($user
        , $project) {
            if($project->isTagged()) {

```

```

        return true;
    }
    return $user->isAdmin();
});

// Activities
$gate->define('show-activities',
    function ($user) {
        return Auth::check();
    });

$gate->define('show-notifications',
    function ($user) {
        return $user->isAdmin();
    });
}
}

```

Listing 46: EventServiceProvider.php

```

<?php

namespace App\Providers;

use Illuminate\Contracts\Events\Dispatcher as
    DispatcherContract;
use Illuminate\Foundation\Support\Providers\
    EventServiceProvider as ServiceProvider;

class EventServiceProvider extends
    ServiceProvider
{
    /**
     * The event listener mappings for the
     * application.
     *
     * @var array
     */
    protected $listen = [
        'App\Events\SomeEvent' => [
            'App\Listeners\EventListener',
        ],
    ];

    /**
     * Register any other events for your
     * application.
     *
     * @param \Illuminate\Contracts\Events\
     *     Dispatcher $events
     * @return void
     */
    public function boot(DispatcherContract
        $events)
    {
        parent::boot($events);
    }
}

```

Listing 47: RouteServiceProvider.php

```

<?php

namespace App\Providers;

use Illuminate\Routing\Router;
use Illuminate\Foundation\Support\Providers\
    RouteServiceProvider as ServiceProvider;

class RouteServiceProvider extends
    ServiceProvider
{
    /**
     * This namespace is applied to the
     * controller routes in your routes
     * file.
     *
     * In addition, it is set as the URL
     * generator's root namespace.
     *
     * @var string
     */
    protected $namespace = 'App\Http\
        Controllers';

    /**
     * Define your route model bindings,
     * pattern filters, etc.
     *
     * @param \Illuminate\Routing\Router
     *     $router
     * @return void
     */

```

```

public function boot(Router $router)
{
    //
    parent::boot($router);
}

/**
 * Define the routes for the application.
 *
 * @param \Illuminate\Routing\Router
     *     $router
 * @return void
 */
public function map(Router $router)
{
    $router->group(['namespace' => $this->
        namespace], function ($router) {
        require app_path('Http/routes.php'
            );
    });
}
}

```

Listing 48: app.php

```

<?php

return [

    /*
     * Application Debug Mode
     *
     * When your application is in debug mode,
     * detailed error messages with
     * stack traces will be shown on every
     * error that occurs within your
     * application. If disabled, a simple
     * generic error page is shown.
     */
    'debug' => env('APP_DEBUG', true),

    /*
     * Application URL
     *
     * This URL is used by the console to
     * properly generate URLs when using
     * the Artisan command line tool. You
     * should set this to the root of
     * your application so that it is used when
     * running Artisan tasks.
     */
    'url' => 'http://localhost',

    /*
     * Application Timezone
     *
     * Here you may specify the default
     * timezone for your application, which
     * will be used by the PHP date and date-
     * time functions. We have gone
     * ahead and set this to a sensible default
     * for you out of the box.
     */
    'timezone' => 'Asia/Manila',

    /*
     * Application Locale Configuration
     *
     * The application locale determines the
     * default locale that will be used
     * by the translation service provider. You
     * are free to set this value
     * to any of the locales which will be
     * supported by the application.
     */

```

```

|
*/
'locale' => 'en',
/*
| Application Fallback Locale
|-----|
| The fallback locale determines the
| locale to use when the current one
| is not available. You may change the
| value to correspond to any of
| the language folders that are provided
| through your application.
|
*/
'fallback_locale' => 'en',
/*
| Encryption Key
|-----|
| This key is used by the Illuminate
| crypter service and should be set
| to a random, 32 character string,
| otherwise these encrypted strings
| will not be safe. Please do this before
| deploying an application!
|
*/
'key' => env('APP_KEY', 'SomeRandomString'
),
'cipher' => 'AES-256-CBC',
/*
| Logging Configuration
|-----|
| Here you may configure the log settings
| for your application. Out of
| the box, Laravel uses the Monolog PHP
| logging library. This gives
| you a variety of powerful log handlers /
| formatters to utilize.
|
| Available Settings: "single", "daily", "
| syslog", "errorlog"
|
*/
'log' => 'single',
/*
| Autoloaded Service Providers
|-----|
| The service providers listed here will
| be automatically loaded on the
| request to your application. Feel free
| to add your own services to
| this array to grant expanded
| functionality to your applications.
|
*/
'providers' => [
    /*
    * Laravel Framework Service Providers
    ...
    */
    Illuminate\Foundation\Providers\
        ArtisanServiceProvider::class,
    Illuminate\Auth\AuthServiceProvider::
        class,
    Illuminate\Broadcasting\
        BroadcastServiceProvider::class,
    Illuminate\Bus\BusServiceProvider::
        class,
    Illuminate\Cache\CacheServiceProvider
        ::class,
    Illuminate\Foundation\Providers\
        ConsoleSupportServiceProvider::
class,
    Illuminate\Cookie\
        CookieServiceProvider::class,
    Illuminate\Database\
        DatabaseServiceProvider::class,
    Illuminate\Encryption\
        EncryptionServiceProvider::class,
    Illuminate\Filesystem\
        FilesystemServiceProvider::class,
    Illuminate\Foundation\Providers\
        FoundationServiceProvider::class,
    Illuminate\Hashing\HashServiceProvider
        ::class,
    Illuminate\Mail\MailServiceProvider::
        class,
    Illuminate\Pagination\
        PaginationServiceProvider::class,
    Illuminate\Pipeline\
        PipelineServiceProvider::class,
    Illuminate\Queue\QueueServiceProvider
        ::class,
    Illuminate\Redis\RedisServiceProvider
        ::class,
    Illuminate\Auth\Passwords\
        PasswordResetServiceProvider::
class,
    Illuminate\Session\
        SessionServiceProvider::class,
    Illuminate\Translation\
        TranslationServiceProvider::class
    ,
    Illuminate\Validation\
        ValidationServiceProvider::class,
    Illuminate\View\ViewServiceProvider::
        class,
],
/*
| Class Aliases
|-----|
| This array of class aliases will be
| registered when this application
| is started. However, feel free to
| register as many as you wish as
| the aliases are "lazy" loaded so they
| don't hinder performance.
|
*/
'aliases' => [
    'App' => Illuminate\Support\
        Facades\App::class,
    'Artisan' => Illuminate\Support\
        Facades\Artisan::class,
    'Auth' => Illuminate\Support\
        Facades\Auth::class,
    'Blade' => Illuminate\Support\
        Facades\Blade::class,
    'Bus' => Illuminate\Support\
        Facades\Bus::class,
    'Cache' => Illuminate\Support\
        Facades\Cache::class,
    'Config' => Illuminate\Support\
        Facades\Config::class,
    'Cookie' => Illuminate\Support\
        Facades\Cookie::class,
    'Crypt' => Illuminate\Support\
        Facades\Crypt::class,
    'DB' => Illuminate\Support\
        Facades\DB::class,
    'Eloquent' => Illuminate\Database\
        Eloquent\Model::class,
    'Event' => Illuminate\Support\
        Facades\Event::class,
    'File' => Illuminate\Support\
        Facades\File::class,

```

```

'Gate' => Illuminate\Support\
    Facades\Gate::class,
'Hash' => Illuminate\Support\
    Facades\Hash::class,
'Input' => Illuminate\Support\
    Facades\Input::class,
'Inspiring' => Illuminate\Foundation\
    Inspiring::class,
'Lang' => Illuminate\Support\
    Facades\Lang::class,
'Log' => Illuminate\Support\
    Facades\Log::class,
'Mail' => Illuminate\Support\
    Facades\Mail::class,
'Password' => Illuminate\Support\
    Facades>Password::class,
'Queue' => Illuminate\Support\
    Facades\Queue::class,
'Redirect' => Illuminate\Support\
    Facades\Redirect::class,
'Redis' => Illuminate\Support\
    Facades\Redis::class,
'Request' => Illuminate\Support\
    Facades\Request::class,
'Response' => Illuminate\Support\
    Facades\Response::class,
'Route' => Illuminate\Support\
    Facades\Route::class,
'Schema' => Illuminate\Support\
    Facades\Schema::class,
'Session' => Illuminate\Support\
    Facades\Session::class,
'Storage' => Illuminate\Support\
    Facades\Storage::class,
'URL' => Illuminate\Support\
    Facades\URL::class,
'Validator' => Illuminate\Support\
    Facades\Validator::class,
'View' => Illuminate\Support\
    Facades\View::class,

'Form' => Collective\Html\
    FormFacade::class,
'Html' => Collective\Html\
    HtmlFacade::class,
'Helper' => App\CustomClasses\
    Helper::class,
'TemplateHelper' => App\
    CustomClasses\TemplateHelper::
        class
    ],
];

```

Listing 49: database.php

```
<?php
```

```

return [
    /*
    |-----|
    | PDO Fetch Style
    |-----|
    |
    | By default, database results will be
    | returned as instances of the PHP
    | stdClass object; however, you may desire
    | to retrieve records in an
    | array format for simplicity. Here you
    | can tweak the fetch style.
    |
    */
    'fetch' => PDO::FETCH_CLASS,
    /*
    |-----|
    | Default Database Connection Name
    |-----|
    |
    | Here you may specify which of the
    | database connections below you wish
    | to use as your default connection for
    | all database work. Of course
    | you may use many connections at once
    | using the Database library.
    |
    */
    'default' => env('DB.CONNECTION', 'mysql')
];

```

```

    /*
    |-----|
    | Database Connections
    |-----|
    |
    | Here are each of the database
    | connections setup for your
    | application.
    | Of course, examples of configuring each
    | database platform that is
    | supported by Laravel is shown below to
    | make development simple.
    |
    | All database work in Laravel is done
    | through the PHP PDO facilities
    | so make sure you have the driver for
    | your particular database of
    | choice installed on your machine before
    | you begin development.
    |
    */
    'connections' => [
        'sqlite' => [
            'driver' => 'sqlite',
            'database' => storage_path('
                database.sqlite'),
            'prefix' => '',
        ],
        'mysql' => [
            'driver' => 'mysql',
            'host' => 'localhost',
            'database' => 'gin-rg',
            'username' => 'root',
            'password' => '',
            'charset' => 'utf8',
            'collation' => 'utf8_unicode_ci',
            'prefix' => '',
            'strict' => false,
        ],
        'pgsql' => [
            'driver' => 'pgsql',
            'host' => env('DB.HOST', '
                localhost'),
            'database' => env('DB.DATABASE', '
                forge'),
            'username' => env('DB.USERNAME', '
                forge'),
            'password' => env('DB.PASSWORD', '
                '),
            'charset' => 'utf8',
            'prefix' => '',
            'schema' => 'public',
        ],
        'sqlsrv' => [
            'driver' => 'sqlsrv',
            'host' => env('DB.HOST', '
                localhost'),
            'database' => env('DB.DATABASE', '
                forge'),
            'username' => env('DB.USERNAME', '
                forge'),
            'password' => env('DB.PASSWORD', '
                '),
            'charset' => 'utf8',
            'prefix' => '',
        ],
    ],
    /*
    |-----|
    | Migration Repository Table
    |-----|
    |
    | This table keeps track of all the
    | migrations that have already run for
    | your application. Using this information
    | , we can determine which of
    | the migrations on disk haven't actually
    | been run in the database.
    |
    */
    'migrations' => 'migrations',
    /*
    |-----|

```

```

| Redis Databases
|-----
|
| Redis is an open source, fast, and
| advanced key-value store that also
| provides a richer set of commands than a
| typical key-value systems
| such as APC or Memcached. Laravel makes
| it easy to dig right in.
|
| */
'redis' => [
    'cluster' => false,
    'default' => [
        'host' => '127.0.0.1',
        'port' => 6379,
        'database' => 0,
    ],
],
];

```

Listing 50: Activity.php

```

<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
class Activity extends Model
{
    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'activities';

    /**
     * The attributes that are mass assignable
     *
     * @var array
     */
    protected $fillable = ['proj_id', 'user_id',
        'activity_type'];

    /**
     * The attributes that are guarded.
     *
     * @var array
     */
    protected $guarded = ['id'];
}

```

Listing 51: Borehole.php

```

<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
class Borehole extends Model
{
    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'boreholes';

    /**
     * The attributes that are mass assignable
     *
     * @var array
     */
    protected $fillable = ['proj_id', 'bh_start_date', 'bh_end_date', 'bh_num',
        'bh_num_of_samples', 'bh_max_depth', 'bh_weather', 'bh_operator',
        'bh_inspector', 'bh_drill_rig', 'bh_drill_pump', 'bh_hammer_weight',
        'bh_height_of_drop', 'bh_casing', 'bh_rod', 'bh_sampler', 'bh_water_elevation',
        'bh_surface_elevation'];
}

```

```

/**
 * The attributes that are guarded.
 *
 * @var array
 */
protected $guarded = ['id'];

/**
 * Get the project that owns the borehole.
 */
public function project()
{
    return $this->belongsTo('App\Project', 'proj_id');
}

/**
 * Get the samples of the borehole.
 */
public function samples()
{
    return $this->hasMany('App\Sample', 'bh_id');
}

/**
 * Get the SBC for shallow foundation of the borehole.
 */
public function shallows()
{
    return $this->hasMany('App\Shallow', 'bh_id');
}

/**
 * Get the SBC for deep foundation of the borehole.
 */
public function deeps()
{
    return $this->hasMany('App\Deep', 'bh_id');
}

/**
 * Get the SBC for drilled shaft foundation of the borehole.
 */
public function drilledShafts()
{
    return $this->hasMany('App\DrilledShaft', 'bh_id');
}
}

```

Listing 52: DataPoint.php

```

<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
class DataPoint extends Model
{
    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'data_points';

    /**
     * The attributes that are mass assignable
     *
     * @var array
     */
    protected $fillable = ['sample_id', 'subreport_id', 'x_int', 'y_int'];

    /**
     * The attributes that are guarded.
     *
     * @var array
     */
    protected $guarded = ['id'];

    /**
     * Get the sample that owns the data.
     */
    public function sample()
    {
        return $this->belongsTo('App\Sample',

```



```

        'sample_id');
    }
}

Listing 53: Deep.php

<?php
namespace App;

use Illuminate\Database\Eloquent\Model;

class Deep extends Model
{
    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'deeps';

    /**
     * The attributes that are mass assignable
     *
     * @var array
     */
    protected $fillable = ['proj_id', 'bh_id',
        'sample_id', 'deep_pile_size',
        'deep_K', 'deep_gamma', 'deep_tan',
        'deep_fs', 'deep_rho', 'deep_fy',
        'deep_fc', 'deep_f', 'deep_Asurface',
        'deep_q', 'deep_Atip', 'deep_Qtotal',
        'deep_Q_kn', 'deep_Q_ton',
        'deep_concrete_strength_kn',
        'deep_concrete_strength_ton',
        'deep_is_chosen'];

    /**
     * The attributes that are guarded.
     *
     * @var array
     */
    protected $guarded = ['id'];

    /**
     * Get the project that owns the data.
     */
    public function project()
    {
        return $this->belongsTo('App\Project',
            'proj_id');
    }

    /**
     * Get the borehole that owns the data.
     */
    public function borehole()
    {
        return $this->belongsTo('App\Borehole',
            'bh_id');
    }

    /**
     * Get the sample that owns the data.
     */
    public function samples()
    {
        return $this->belongsTo('App\Sample',
            'sample_id');
    }
}

```

Listing 54: DrilledShaft.php

```

<?php
namespace App;

use Illuminate\Database\Eloquent\Model;

class DrilledShaft extends Model
{
    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'drilled_shafts';

    /**
     * The attributes that are mass assignable
     *
     * @var array
     */
    protected $fillable = ['proj_id', 'bh_id',
        'sample_id', 'ds_soil',
        'ds_pile_size', 'ds_Nc', 'ds_f_coeff',
        'ds_fc', 'ds_fy',
        'ds_steel_bar_diameter', 'ds_fs',
        'ds_tension_cap', 'ds_c', 'ds_Qbase',
        'ds_f', 'ds_Qfriction', 'ds_Qtotal_kn',
        'ds_Qtotal_ton',
        'ds_concrete_strength_kn',
        'ds_concrete_strength_ton',
        'ds_is_chosen'];

    /**
     * The attributes that are guarded.
     *
     * @var array
     */
    protected $guarded = ['id'];

    /**
     * Get the project that owns the data.
     */
    public function project()
    {
        return $this->belongsTo('App\Project',
            'proj_id');
    }

    /**
     * Get the borehole that owns the data.
     */
    public function borehole()
    {
        return $this->belongsTo('App\Borehole',
            'bh_id');
    }

    /**
     * Get the sample that owns the data.
     */
    public function samples()
    {
        return $this->belongsTo('App\Sample',
            'sample_id');
    }
}

```

```

    /**
     * The attributes that are guarded.
     *
     * @var array
     */
    protected $guarded = ['id'];

    /**
     * Get the project that owns the data.
     */
    public function project()
    {
        return $this->belongsTo('App\Project',
            'proj_id');
    }

    /**
     * Get the borehole that owns the data.
     */
    public function borehole()
    {
        return $this->belongsTo('App\Borehole',
            'bh_id');
    }

    /**
     * Get the sample that owns the data.
     */
    public function samples()
    {
        return $this->belongsTo('App\Sample',
            'sample_id');
    }
}

```

Listing 55: Limit.php

```

<?php
namespace App;

use Illuminate\Database\Eloquent\Model;

class Limit extends Model
{
    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'limits';

    /**
     * The attributes that are mass assignable
     *
     * @var array
     */
    protected $fillable = ['sample_id',
        'subreport_id', 'lim_run_num',
        'lim_test', 'lim_num_of_blows',
        'lim_water_content'];

    /**
     * The attributes that are guarded.
     *
     * @var array
     */
    protected $guarded = ['id'];

    /**
     * Get the sample that owns the data.
     */
    public function samples()
    {
        return $this->belongsTo('App\Sample',
            'sample_id');
    }
}

```

Listing 56: Project.php

```

<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
class Project extends Model
{
    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'projects';

    /**
     * The attributes that are mass assignable
     *
     * @var array
     */
    protected $fillable = ['proj-name', 'proj-description', 'proj-location', 'proj-client', 'proj-num-of-boreholes', 'proj-start-date', 'proj-end-date', 'proj-date-signed', 'proj-status'];

    /**
     * The attributes that are guarded.
     *
     * @var array
     */
    protected $guarded = ['id'];

    /**
     * Checks if project is active
     */
    public function isActive()
    {
        if ($this->proj_status == 0) {
            return true;
        }

        return false;
    }

    /**
     * Checks if project is tagged
     */
    public function isTagged()
    {
        if ($this->proj_status == 1) {
            return true;
        }

        return false;
    }

    /**
     * Checks if project is archived
     */
    public function isArchived()
    {
        if ($this->proj_status == 2) {
            return true;
        }

        return false;
    }

    /**
     * Get the boreholes of the project.
     */
    public function boreholes()
    {
        return $this->hasMany('App\Borehole', 'proj-id');
    }

    /**
     * Get the sections of the project.
     */
    public function sections()
    {
        return $this->hasMany('App\Section', 'proj-id');
    }

    /**
     * Get the subreports of the project.
     */
    public function subreports()
    {
        return $this->hasMany('App\Subreport', 'proj-id');
    }

    /**

```

```

     * Get the SBC for shallow foundation of the project.
     */
    public function shallows()
    {
        return $this->hasMany('App\Shallow', 'proj-id');
    }

    /**
     * Get the SBC for deep foundation of the project.
     */
    public function deeps()
    {
        return $this->hasMany('App\Deep', 'proj-id');
    }

    /**
     * Get the SBC for drilled shaft foundation of the project.
     */
    public function drilledShafts()
    {
        return $this->hasMany('App\DrilledShaft', 'proj-id');
    }
}

```

Listing 57: Sample.php

```

<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
class Sample extends Model
{
    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'samples';

    /**
     * The attributes that are mass assignable
     *
     * @var array
     */
    protected $fillable = ['bh_id', 'uscs_id', 'sample_num', 'sample_from', 'sample_to', 'sample_contract_num', 'sample_job_num', 'sample_color', 'sample_run_length', 'sample_sample_length', 'sample_rec_percent', 'sample_nmc_percent', 'sample_num_of_blows', 'sample_liquid_limit', 'sample_plastic_limit', 'sample_plasticity_index', 'sample_plasticity_remark', 'sample_tester', 'sample_type_of_sampling', 'sample_test_date', 'sample_other_results'];

    /**
     * The attributes that are guarded.
     *
     * @var array
     */
    protected $guarded = ['id'];

    /**
     * Get the borehole that owns the sample.
     */
    public function borehole()
    {
        return $this->belongsTo('App\Borehole', 'bh_id');
    }

    /**
     * Get the data points of the sample.
     */
    public function dataPoints()
    {
        return $this->hasMany('App\DataPoint',

```

```

        'sample_id');
    }

    /**
     * Get the SBC for shallow foundation of
     * the sample.
     */
    public function shallow()
    {
        return $this->hasOne('App\Shallow', '
            sample_id');
    }

    /**
     * Get the SBC for deep foundation of the
     * sample.
     */
    public function deep()
    {
        return $this->hasOne('App\Deep', '
            sample_id');
    }

    /**
     * Get the SBC for drilled shaft
     * foundation of the sample.
     */
    public function drilledShaft()
    {
        return $this->hasOne('App\DrilledShaft
            ', 'sample_id');
    }
}

```

Listing 58: Section.php

```

<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
class Section extends Model
{
    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'sections';

    /**
     * The attributes that are mass assignable
     *
     * @var array
     */
    protected $fillable = ['proj_id', '
        section_key'];

    /**
     * The attributes that are guarded.
     *
     * @var array
     */
    protected $guarded = ['id'];

    /**
     * Get the project that owns the section.
     */
    public function project()
    {
        return $this->belongsTo('App\Project', '
            proj_id');
    }

    /**
     * Get the (body) sections of the project.
     */
    public function sectionBody()
    {
        return $this->hasOne('App\SectionBody'
            , 'section_id');
    }
}

```

Listing 59: SectionBody.php

```

<?php
namespace App;
use Illuminate\Database\Eloquent\Model;

```

```

class SectionBody extends Model
{
    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'section_bodies';

    /**
     * The attributes that are mass assignable
     *
     * @var array
     */
    protected $fillable = ['section_id', '
        section_text'];

    /**
     * The attributes that are guarded.
     *
     * @var array
     */
    protected $guarded = ['id'];
}

```

Listing 60: Shallow.php

```

<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
class Shallow extends Model
{
    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'shallows';

    /**
     * The attributes that are mass assignable
     *
     * @var array
     */
    protected $fillable = ['proj_id', 'bh_id',
        'sample_id', 'shallow_soil', '
        shallow_c', 'shallow_B', 'shallow_phi
        ', 'shallow_Nc', 'shallow_Ngamma', '
        shallow_Nq', 'shallow_gammal', '
        shallow_gamma2', 'shallow_fs', '
        shallow_qult', 'shallow-q', '
        shallow_is_chosen'];

    /**
     * The attributes that are guarded.
     *
     * @var array
     */
    protected $guarded = ['id'];

    /**
     * Get the project that owns the data.
     */
    public function project()
    {
        return $this->belongsTo('App\Project', '
            proj_id');
    }

    /**
     * Get the borehole that owns the data.
     */
    public function borehole()
    {
        return $this->belongsTo('App\Borehole'
            , 'bh_id');
    }

    /**
     * Get the sample that owns the data.
     */
    public function samples()
    {
        return $this->belongsTo('App\Sample', '
            sample_id');
    }
}

```

Listing 61: Subreport.php

```

<?php

```

```

namespace App;

use Illuminate\Database\Eloquent\Model;

class Subreport extends Model
{
    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'subreports';

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = ['proj_id', 'sb_key'
    ];

    /**
     * The attributes that are guarded.
     *
     * @var array
     */
    protected $guarded = ['id'];

    /**
     * Get the project that owns the section.
     */
    public function project()
    {
        return $this->belongsTo('App\Project',
            'proj_id');
    }

    /**
     * Get the data points of the subreport.
     */
    public function dataPoints()
    {
        return $this->hasMany('App\DataPoint',
            'subreport_id');
    }

    /**
     * Get the data points of the subreport.
     */
    public function limits()
    {
        return $this->hasMany('App\Limit',
            'subreport_id');
    }
}

```

Listing 62: User.php

```

<?php

namespace App;

use DB;
use Illuminate\Auth\Authenticated;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Auth\Passwords\CanResetPassword;
;
use Illuminate\Foundation\Auth\Access\
    Authorizable;
use Illuminate\Contracts\Auth\Authenticated
    as AuthenticatedContract;
use Illuminate\Contracts\Auth\Access\
    Authorizable as AuthorizableContract;
use Illuminate\Contracts\Auth\CanResetPassword
    as CanResetPasswordContract;

class User extends Model implements
    AuthenticatedContract,
    AuthorizableContract,
    CanResetPasswordContract
{
    use Authenticated, Authorizable,
        CanResetPassword;

    /**
     * The database table used by the model.
     *
     * @var string
     */
    protected $table = 'users';
}

```

```

/**
 * The attributes that are mass assignable.
 *
 * @var array
 */
protected $fillable = ['firstname', '
    middlename', 'lastname', 'email', '
    contact', 'position', 'status', 'sex',
    'birthdate', 'about', 'image', '
    password'];

/**
 * The attributes that are guarded.
 *
 * @var array
 */
protected $guarded = ['id'];

/**
 * The attributes excluded from the model's
    JSON form.
 *
 * @var array
 */
protected $hidden = ['password', '
    remember_token'];

public function isAdmin()
{
    if ($this->status == 2) {
        return true;
    }

    return false;
}

public function isMember()
{
    if ($this->status == 1) {
        return true;
    }

    return false;
}

public function isPending()
{
    if ($this->status == 0) {
        return true;
    }

    return false;
}

public function isDisabled()
{
    if ($this->status == 3) {
        return true;
    }

    return false;
}

/**
 * Check if user is in a project using
    project id.
 */
public function isInProject($proj_id)
{
    $matchThese = ['proj_id' => $proj_id,
        'enr_id' => $this->id];
    $project = DB::table('
        project_engineers')->where(
        $matchThese)->get();

    if (sizeof($project) > 0) {
        return true;
    }

    return false;
}

/**
 * Check if user is in a project using
    borehole id.
 */
public function isInProject2($bh_id)
{
    $matchThese = ['boreholes.id' =>
        $bh_id, 'project_engineers.
        enr_id' => $this->id];
    $project = DB::table('
        project_engineers')
        ->join('boreholes', '
            project_engineers.proj_id', '
            =', 'boreholes.proj_id')
        ->where($matchThese)->get();
}

```

```

    if(sizeof($project) > 0) {
        return true;
    }

    return false;
}
}

```

Listing 63: activities/index.blade.php

```

@extends('layouts/master')
@section('title', 'Audit Trail')
@section('content')
<div class="row">
    <div class="col-md-12 col-sm-12 col-xs-12">
        <div class="x_panel">
            <div class="x_title">
                <h2> Audit Trail <small>View
                user activities </small>
                </h2>
                <ul class="nav navbar-right
                panel_toolbox">
                    <li><a class="collapse-
                    link"><i class="fa fa
                    -chevron-up"></i></a>
                    </li>
                    <li class="dropdown">
                        <a href="#" class="
                        dropdown-toggle"
                        data-toggle="
                        dropdown" role="
                        button" aria-
                        expanded="false"
                        ><i class="fa fa-
                        wrench"></i></a>
                        <ul class="dropdown-
                        menu" role="menu"
                        >
                            <li><a href="#">
                                Settings 1</a>
                            </li>
                            <li><a href="#">
                                Settings 2</a>
                            </li>
                        </ul>
                    </li>
                </ul>
            </div>
            <div class="clearfix"></div>
        </div>
        <div class="x_content">
            @if(sizeof($activities) == 0)
            <center><h2>There are no
            activities.</h2></center>
            @else
            <div class="dashboard-widget-
            content">
                <ul class="list-unstyled
                timeline_widget">
                    @foreach($activities
                    as $activity)
                    <li>
                        <div class="block">
                            <div class="
                            block_content
                            ">
                                <h2 class=
                                "
                                title
                                ">
                                    <?php
                                    $user_name
                                    =
                                    $activity
                                    ->
                                    firstname
                                    .
                                    "
                                    </div>
                                    @endsection
                                </div>
                            </li>
                        </ul>
                    </div>
                </div>
            @endif
        </div>
    </div>
</div>

```

```

?>
{!!
    $activity
    ->
    activity_type
    !!}
</h2>
<div class
    =
    "
    byline
    ">
<span>
    {
        Helper
        ::
        getTimeElapsedStrin
        (
            $activity
            ->
            created_at
        )
    }
</
span
>
    by
    @if(
        $activity
        ->
        user_id
        !=
        NULL
        )
    {!!
        link_to_route
        (
            users
            .
            show
            ,
            $user_name
            ,
            $activity
            ->
            user_id
            )
        !!}
    @else
    Deactivated
    user
    @endif
</div>
<p class="
excerpt
"></p>
</div>
</li>
</div>
@endforeach
</ul>
<center>{!! $activities->
render() !!}</center>
</div>
@endif
$activity
->
firstname
</div>
</div>
"
</div>
"
@endsection
"

```

Listing 64: activities/notifications.blade.php

```

@extends('layouts/master')
@section('title', 'Notifications')

```

```

@section('content')
<!-- pending users -->
<div class="row">
  <div class="col-md-6_col-sm-6_col-xs-12">
    <div class="x_panel">
      <div class="x_title">
        <h2> Pending Users <small>
          Newly-registered users </small></h2>
        <ul class="nav navbar-right _
          panel_toolbox">
          <li><a class="collapse-
            link"><i class="fa_fa
              -chevron-up"></i></a>
          </li>
          <li class="dropdown">
            <a href="#" class="
              dropdown-toggle"
              data-toggle="
              dropdown" role="
              button" aria-
              expanded="false"
              ><i class="fa_fa-
                wrench"></i></a>
            <ul class="dropdown-
              menu" role="menu"
              >
              <li><a href="#">
                Settings 1</a>
              >
              </li>
              <li><a href="#">
                Settings 2</a>
              >
              </li>
            </ul>
          </li>
        </ul>
      </div>
      <div class="clearfix"></div>
    </div>
    <div class="x_content">
      @if(sizeof($pending-users) ==
        0)
      <center><h2>There are no
        pending users.</h2></center>
      @else
      <div class="dashboard-widget-
        content">
        <ul class="list-unstyled _
          timeline_widget">
          @foreach(
            $pending-users as
            $user)
          <li>
            <div class="block">
              <div class="
                block_content
                ">
                <h2 class="
                  "
                  title
                  "><a
                    href="
                    "{{_
                      route
                      (
                        'users
                          .show
                          ',
                          $user
                          ->id)
                        }}"
                    >{{
                      $user
                      ->
                      firstname
                      . " "
                      $user
                      ->
                      lastname
                    }}</a></h2>
                <div class="
                  "
                  byline
                  ">
                  Registered
                <span
                >{{
                  Helper
                    ::
                    getTimeElapsedStrin
                    (
                      $user
                      ->
                      created_at
                    )
                    }}</span
                >
                </div>
                <p class="
                  excerpt
                  ">
                  <label
                  >
                  Email
                  :</label
                  >
                  <label
                  >
                  <div class="
                    {}
                    $user
                    ->
                    email
                    {}
                    </div>
                  <br
                  />
                  <label
                  >
                  Contact
                  No
                  :</label
                  >
                  <div class="
                    {}
                    $user
                    ->
                    contact
                    {}
                    </div>
                </p>
              </div>
            </li>
          @endforeach
        </ul>
      </div>
      </div>
      @endif
      <center><a href="{{_route('
        users.index')}}_#
        pending_users'_" class="
        btn btn-primary"><i
          class="fa_fa-users"></i>
          Manage Pending Users </a>
      </center>
    </div>
  </div>
</div>
<!-- /pending users -->
<!-- tagged projects -->
<div class="col-md-6_col-sm-6_col-xs-12">
  <div class="x_panel">
    <div class="x_title">
      <h2> Tagged Projects <small>
        Validate projects to
        archive them</small> </h2>
      <ul class="nav navbar-right _
        panel_toolbox">
      <li><a class="collapse-
        link"><i class="fa_fa
          -chevron-up"></i></a>
      </li>
      <li class="dropdown">
        <a href="#" class="
          dropdown-toggle"
          data-toggle="
          dropdown" role="
          button" aria-
          expanded="false"
          ><i class="fa_fa-
            wrench"></i></a>
        <ul class="dropdown-
          menu" role="menu"
          >
          <li><a href="#">
        </li>
      </ul>
      </li>
    </ul>
  </div>
  <div class="
    "
    byline
    ">
    Registered
  <span
  >{{
    Helper
      ::
      getTimeElapsedStrin
      (
        $user
        ->
        created_at
      )
      }}</span
  >
  </div>
  <p class="
    excerpt
    ">
    <label
    >
    Email
    :</label
    >
    <label
    >
    <div class="
      {}
      $user
      ->
      email
      {}
      </div>
    <br
    />
    <label
    >
    Contact
    No
    :</label
    >
    <div class="
      {}
      $user
      ->
      contact
      {}
      </div>
  </p>
  </div>
  </div>
  @endif
  <center><a href="{{_route('
    users.index')}}_#
    pending_users'_" class="
    btn btn-primary"><i
      class="fa_fa-users"></i>
      Manage Pending Users </a>
  </center>
</div>
</div>

```



```

<link href="../../css/editor/index.css" rel="
stylesheet">
<!-- select2 -->
<link href="../../css/select/select2.min.css"
rel="stylesheet">
<!-- switchery -->
<link href="../../css/switchery/switchery.min
.css" rel="stylesheet"/>

<script src="../../js/jquery.min.js"></script
>
</head>
<body style="background:#F7F7F7;">
  <div class="">
    @if (count($errors) > 0)
    <div class="alert alert-danger"
role="alert">
      @foreach ($errors->all() as $error
)
        <span class="glyphicon_
glyphicon-exclamation-
sign" aria-hidden="true"
></span>
        {{ $error }}
      <br />
    @endforeach
  </div>
  @endif

  <div id="wrapper">
    <div class="animate_form">
      <section class="login_content">
        >
        {!! Form::open(array('
method' => 'POST', '
route' => 'auth.
register')) !!}
        {!! Form::hidden('_token',
csrf_token()) !!}
        <h1>Create Account</h1
>
      </section>

      <div class="form-group"
">
        {!! Form::label('
firstname', '
First_Name_*'
) !!}
        {!! Form::text('
firstname',
old('
firstname'),
['class' => '
form-control',
'required'
=> 'true']) !!}
      </div>

      <div class="form-group"
">
        {!! Form::label('
middlename', '
Middle_Name_*'
) !!}
        {!! Form::text('
middlename',
old('
middlename'),
['class' => '
form-control
']) !!}
      </div>

      <div class="form-group"
">
        {!! Form::label('
lastname', '
Last_Name_*'
) !!}
        {!! Form::text('
lastname',
old('lastname'
), ['class'
=> 'form-
control', '
required' =>
'true']) !!}
      </div>

      <div class="form-group"
">
        {!! Form::label('
email', '
Email_*') !!}
        {!! Form::text('
email', old('
email'), ['
class' => '
form-control',
'required'
=> 'true']) !!}
      </div>

      <div class="form-group"
">
        {!! Form::label('
contact', '
Contact_
Number_*')
        !!}
        {!! Form::text('
contact', old(
'contact'),
['class' => '
form-control',
'placeholder'
=> '+63_9
xxxxxxxx', '
required' =>
'true']) !!}
      </div>

      <div class="form-group"
">
        {!! Form::label('
password', '
Password_*')
        !!}
        {!! Form::password(
'password',
['class' => '
form-control',
'required'
=> 'true'])
        !!}
      </div>

      <div class="form-group"
">
        {!! Form::label('
password_confirmation',
'Confirm_
Password_*')
        !!}
        {!! Form::password(
'password_confirmation',
['class'
=> 'form-
control', '
required' =>
'true']) !!}
      </div>

      <section class="
login_content">
      <div>
        {!! Form::submit('
Submit', ['
class' => '
btn btn-
primary'], ['
style' => '
margin-right:
_15px;']) !!}
      </div>
      <div class="clearfix"
"></div>
      <div class="separator"
">
        <p class="
change_link">
          Already a
          member?
          {!!
            link_to_route
            ('auth.
login', '
Log_in',
null,
array('
class' =>
'
to_login'
)) !!}
        </p>
      </div>
    </div>
  </div>

```

```

        div>
        <br />
        <br />
        <div>
            <p> 2015 All
            Rights
            Reserved.

            Gentilella
            Alela!
            is a
            Bootstrap
            3
            template
            .</p>
        </div>
        </div>
        {!! Form::close() !!}
    </section>
</div>
</div>
</div>
</body>
</html>

```

Listing 67: boreholes/create.blade.php

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
    <div class="col-md-12_col-sm-12_col-xs-12">
        >
        <div class="x_panel">
            <div class="x_title">
                <h2> Add Borehole </h2>
                <ul class="nav navbar-right
                panel_toolbox">
                    <li><a class="collapse-
                    link"><i class="fa fa
                    -chevron-up"></i></a>
                </li>
                <li class="dropdown">
                    <a href="#" class="
                    dropdown-toggle"
                    data-toggle="
                    dropdown" role="
                    button" aria-
                    expanded="false"
                    ><i class="fa fa-
                    wrench"></i></a>
                    <ul class="dropdown-
                    menu" role="menu"
                    >
                        <li><a href="#">
                            Settings 1</a>
                        </li>
                        <li><a href="#">
                            Settings 2</a>
                        </li>
                    </ul>
                </li>
                </ul>
            </div>
            <div class="clearfix"></div>
            <div class="x_content">
                {!! Form::open(['name' => '
                store_form', 'method' =>
                'POST', 'route' => '
                boreholes.store', 'class'
                => 'form-horizontal_form
                -label-left']) !!}
                <div class="container">
                    <div class="form-group">
                        {!! Form::label('
                        bh_max_depth', '
                        Maximum_Depth(m)
                        _*', array('class'
                        => 'control-
                        label_col-md-3_
                        col-sm-3_col-xs
                        -12')) !!}
                    <div class="col-md-6_
                    col-sm-6_col-xs
                    -12">
                        {!! Form::number('
                        bh_max_depth'
                        , old('
                        bh_max_depth'
                        ), ['class'
                        => 'form-
                        control_col-
                        md-7_col-xs
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

-12', '
required' =>
'true']) !!}
</div>
</div>
<div class="form-group">
    {!! Form::label('
    bh_weather', '
    Weather', array('
    class' => '
    control-label_col
    -md-3_col-sm-3_
    col-xs-12')) !!}
    <div class="col-md-6_
    col-sm-6_col-xs
    -12">
        {!! Form::text('
        bh_weather',
        old('
        bh_weather'),
        ['class' =>
        'form-control
        _col-md-7_col
        -xs-12']) !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label('
    bh_operator', '
    Operator_*',
    array('class' =>
    'control-label_
    col-md-3_col-sm-3_
    col-xs-12')) !!}
    <div class="col-md-6_
    col-sm-6_col-xs
    -12">
        {!! Form::text('
        bh_operator',
        old('
        bh_operator'),
        ['class' =>
        'form-
        control_col-
        md-7_col-xs
        -12', '
        required' =>
        'true']) !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label('
    bh_inspector', '
    Inspector', array
    ('class' => '
    control-label_col
    -md-3_col-sm-3_
    col-xs-12')) !!}
    <div class="col-md-6_
    col-sm-6_col-xs
    -12">
        {!! Form::text('
        bh_inspector',
        old('
        bh_inspector'
        ), ['class'
        => 'form-
        control_col-
        md-7_col-xs
        -12']) !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label('
    bh_drill_rig', '
    Drill_Rig', array
    ('class' => '
    control-label_col
    -md-3_col-sm-3_
    col-xs-12')) !!}
    <div class="col-md-6_
    col-sm-6_col-xs
    -12">
        {!! Form::text('
        bh_drill_rig',
        old('
        bh_drill_rig'
        ), ['class'
        => 'form-
        control_col-
        md-7_col-xs
        -12', '
        placeholder'
        => 'YBM'])
        !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label('

```

```

        bh_drill_pump', '
        Drill_Pump',
        array('class' =>
        'control-label_
        col-md-3_col-sm-3
        _col-xs-12')) !!}
<div class="col-md-6_
        col-sm-6_col-xs
        -12">
        {!! Form::text('
        bh_drill_pump
        ', old('
        bh_drill_pump
        '), ['class'
        => 'form-
        control_col-
        md-7_col-xs
        -12',
        'placeholder'
        => 'Shark']) !!}
    </div>
</div>
<div class="form-group">
        {!! Form::label('
        bh_hammer_weight',
        'Hammer_Weight_
        (kg)', array('
        class' =>
        'control-label_col
        -md-3_col-sm-3_
        col-xs-12')) !!}
<div class="col-md-6_
        col-sm-6_col-xs
        -12">
        {!! Form::number('
        bh_hammer_weight
        ', old('
        bh_hammer_weight
        '), ['class'
        => 'form-
        control_col-
        md-7_col-xs
        -12', 'step'
        => 'any',
        'placeholder'
        => '63.5']) !!}
    </div>
</div>
<div class="form-group">
        {!! Form::label('
        bh_height_of_drop',
        'Height_of_
        Drop_(m)', array(
        'class' =>
        'control-label_col
        -md-3_col-sm-3_
        col-xs-12')) !!}
<div class="col-md-6_
        col-sm-6_col-xs
        -12">
        {!! Form::number('
        bh_height_of_drop
        ', old('
        bh_height_of_drop
        '), ['class'
        => 'form-
        control_col-
        md-7_col-xs
        -12', 'step'
        => 'any',
        'placeholder'
        => '1.6']) !!}
    </div>
</div>
<div class="form-group">
        {!! Form::label('
        bh_casing',
        'Casing_*', array(
        'class' =>
        'control-label_col
        -md-3_col-sm-3_
        col-xs-12')) !!}
<div class="col-md-6_
        col-sm-6_col-xs
        -12">
        {!! Form::text('
        bh_casing',
        old('
        bh_casing'),
        ['class' =>
        'form-control_
        col-md-7_col-
        xs-12',
        'required' =>
        'true']) !!}
    </div>
</div>

```

```

</div>
<div class="form-group">
        {!! Form::label('
        bh_rod', 'Rod_*',
        array('class' =>
        'control-label_
        col-md-3_col-sm-3
        _col-xs-12')) !!}
<div class="col-md-6_
        col-sm-6_col-xs
        -12">
        {!! Form::text('
        bh_rod', old(
        'bh_rod'), ['
        class' =>
        'form-control_
        col-md-7_col-
        xs-12',
        'required' =>
        'true']) !!}
    </div>
</div>
<div class="form-group">
        {!! Form::label('
        bh_sampler',
        'Sampler_*', array
        ('class' =>
        'control-label_col
        -md-3_col-sm-3_
        col-xs-12')) !!}
<div class="col-md-6_
        col-sm-6_col-xs
        -12">
        {!! Form::text('
        bh_sampler',
        old('
        bh_sampler'),
        ['class' =>
        'form-control_
        col-md-7_col
        -xs-12',
        'required' =>
        'true']) !!}
    </div>
</div>
<div class="form-group">
        {!! Form::label('
        bh_water_elevation',
        'Water_
        Elevation_(m)_*',
        array('class' =>
        'control-label_
        col-md-3_col-sm-3
        _col-__xs-12'))
        !!}
<div class="col-md-6_
        col-sm-6_col-xs
        -12">
        {!! Form::number('
        bh_water_elevation
        ', old('
        bh_water_elevation
        '), ['class'
        => 'form-
        control_col-
        md-7_col-xs
        -12',
        'required' =>
        'true', 'step'
        => 'any'])
        !!}
    </div>
</div>
<div class="form-group">
        {!! Form::label('
        bh_surface_elevation',
        'Surface_
        Elevation_*',
        array('class' =>
        'control-label_
        col-md-3_col-sm-3
        _col-xs-12')) !!}
<div class="col-md-6_
        col-sm-6_col-xs
        -12">
        {!! Form::number('
        bh_surface_elevation
        ', old('
        bh_surface_elevation
        '), ['class'
        => 'form-
        control_col-
        md-7_col-xs
        -12',
        'required' =>
        'true', 'step'
        => 'any'])
        !!}
    </div>
</div>

```

```

</div>
</div>
<div class="form-group">
  {!! Form::label('
    bh_start_date', '
    Start Date - End
    Date_*', ['class'
    => 'control-
    label_col-md-3_
    col-sm-3_col-xs
    -12']) !!}
  <div class="col-md-3_
    col-sm-6_col-xs
    -12_form-group">
    {!! Form::date('
      bh_start_date
      ', old('
      bh_start_date
      '), ['class'
      => 'form-
      control', '
      required' =>
      'true']) !!}
  </div>
  <div class="col-md-3_
    col-sm-6_col-xs
    -12_form-group">
    {!! Form::date('
      bh_end_date',
      old('
      bh_start_date
      '), ['class'
      => 'form-
      control', '
      required' =>
      'true']) !!}
  </div>
</div>
<div class="ln_solid"></div>
<?php $bh_num = $project->
proj_num_of_boreholes + 1
?>
{!! Form::hidden('bh_num',
  $bh_num) !!}
{!! Form::hidden('proj_id',
  $project->id) !!}
{!! link_to(URL::previous(), '
  Back', array('class' => '
  btn_btn-default')) !!}
{!! Form::submit('Submit', ['
  class' => 'btn_btn-
  primary', 'style' => '
  float:right']) !!}
{!! Form::close() !!}
</div>
</div>
</div>
</div>
@endsection

```

Listing 68: boreholes/edit.blade.php

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
  <div class="col-md-12_col-sm-12_col-xs-12">
    <div class="x_panel">
      <div class="x_title">
        <h2> Edit Borehole {!!
          $borehole->bh_num !!} <
          small>(ID#{ { $borehole->
          id }})</small></h2>
        <ul class="nav navbar-right_
          panel_toolbox">
          <li><a class="collapse-
            link"><i class="fa_fa-
            chevron-up"></i></a>
          </li>
          <li class="dropdown">
            <a href="#" class="
            dropdown-toggle"
            data-toggle="
            dropdown" role="
            button" aria-
            expanded="false"
            ><i class="fa_fa-
            wrench"></i></a>
            <ul class="dropdown-
            menu" role="menu"
            >
              <li><a href="#">
                Settings 1</a>
              >
            </ul>
          </li>
        </ul>
      </div>
    </div>
  </div>

```

```

</li>
<li><a href="#">
  Settings 2</a>
</li>
</ul>
</div>
<div class="clearfix"></div>
<div class="x_content">
  {!! Form::open(['name' => '
    update_form', 'method' =>
    'PUT', 'route' => ['
    boreholes.update',
    $borehole->id], 'class'
    => 'form-horizontal_
    form-label-left', 'id' =>
    'update_form']) !!}
  <div class="container">
    <div class="form-group">
      {!! Form::label('
        bh_max_depth', '
        Maximum_Depth_(m)
        _*', array('class'
        => 'control-
        label_col-md-3_
        col-sm-3_col-xs
        -12')) !!}
      <div class="col-md-6_
        col-sm-6_col-xs
        -12">
        <?php
          $bh_max_depth
          = Helper::
          getValue(old(
            'bh_max_depth
            '), $borehole
            ->
            bh_max_depth)
            ?>
        {!! Form::number('
          bh_max_depth',
          $bh_max_depth
          , ['class' => '
          form-
          control_col-
          md-7_col-xs
          -12', '
          required' =>
          'true']) !!}
      </div>
    </div>
    <div class="form-group">
      {!! Form::label('
        bh_weather', '
        Weather', array('
        class' => '
        control-label_col-
        md-3_col-sm-3_
        col-xs-12')) !!}
      <div class="col-md-6_
        col-sm-6_col-xs
        -12">
        <?php $bh_weather
          = Helper::
          getValue(old(
            'bh_weather->
            $borehole->
            bh_weather)
            ?>
        {!! Form::text('
          bh_weather',
          $bh_weather,
          ['class' => '
          form-control_
          col-md-7_col-
          xs-12']) !!}
      </div>
    </div>
    <div class="form-group">
      {!! Form::label('
        bh_operator', '
        Operator_*', array('class' =>
        'control-label_
        col-md-3_col-sm-3_
        col-xs-12')) !!}
      <div class="col-md-6_
        col-sm-6_col-xs
        -12">
        <?php $bh_operator
          = Helper::
          getValue(old(
            'bh_operator'
            ), $borehole
            ->bh_operator
            ) ?>
      </div>
    </div>
  </div>

```

```

        {!! Form::text(
            ,
            bh_operator',
            $bh_operator
            , ['class' =>
                'form-
                control_col-
                md-7_col-xs
                -12',
                '
                required' =>
                'true']) !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label(
        ,
        bh_inspector',
        Inspector', array(
            'class' =>
            ,
            control_label_col-
            md-3_col-sm-3_
            col-xs-12')) !!}
    <div class="col-md-6_
        col-sm-6_col-xs
        -12">
        <?php
            $bh_inspector
            = Helper::
            getValue(old(
                'bh_inspector
                '), $borehole
            ->
            bh_inspector)
            ?>
        {!! Form::text(
            ,
            bh_inspector',
            $bh_inspector
            , ['class' =>
                'form-
                control_col-
                md-7_col-xs
                -12']) !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label(
        ,
        bh_drill_rig',
        Drill_Rig_*,
        array('class' =>
            ,
            control_label_
            col-md-3_col-sm-3_
            col-xs-12')) !!}
    <div class="col-md-6_
        col-sm-6_col-xs
        -12">
        <?php
            $bh_drill_rig
            = Helper::
            getValue(old(
                'bh_drill_rig
                '), $borehole
            ->
            bh_drill_rig)
            ?>
        {!! Form::text(
            ,
            bh_drill_rig',
            $bh_drill_rig
            , ['class' =>
                'form-
                control_col-
                md-7_col-xs
                -12',
                '
                required' =>
                'true']) !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label(
        ,
        bh_drill_pump',
        Drill_Pump_*,
        array('class' =>
            ,
            control_label_
            col-md-3_col-sm-3_
            col-xs-12')) !!}
    <div class="col-md-6_
        col-sm-6_col-xs
        -12">
        <?php
            $bh_drill_pump
            = Helper::
            getValue(old(
                'bh_drill_pump
                '), $borehole
            ->
            bh_drill_pump
            ) ?>
        {!! Form::text(
            ,
            bh_drill_pump
            ,
            $bh_drill_pump
            , ['class' =>
                'form-
                control_col-
                md-7_col-xs
                -12',
                '
                required' =>
                'true']) !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label(
        ,
        bh_hammer_weight',
        Hammer_Weight_
        (kg)_*, array(
            'class' =>
            ,
            control_label_col-
            md-3_col-sm-3_
            col-xs-12')) !!}
    <div class="col-md-6_
        col-sm-6_col-xs
        -12">
        <?php
            $bh_hammer_weight
            = Helper::
            getValue(old(
                'bh_hammer_weight
                '), $borehole
            ->
            bh_hammer_weight
            ) ?>
        {!! Form::number(
            ,
            bh_hammer_weight',
            $bh_hammer_weight
            , ['class' =>
                'form-
                control_col-
                md-7_col-xs
                -12',
                '
                required' =>
                'true',
                'step'
                => 'any'])
            !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label(
        ,
        bh_height_of_drop',
        Height_of_
        Drop_(m)_*,
        array('class' =>
            ,
            control_label_
            col-md-3_col-sm-3_
            col-xs-12')) !!}
    <div class="col-md-6_
        col-sm-6_col-xs
        -12">
        <?php
            $bh_height_of_drop
            = Helper::
            getValue(old(
                'bh_height_of_drop
                '), $borehole
            ->
            bh_height_of_drop
            ) ?>
        {!! Form::number(
            ,
            bh_height_of_drop',
            $bh_height_of_drop
            , ['class' =>
                'form-
                control_col-
                md-7_col-xs
                -12',
                '
                required' =>
                'true',
                'step'
                => 'any'])
            !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label(
        ,
        bh_casing',
        Casing_*, array(
            'class' =>
            ,
            control_label_col-
            md-3_col-sm-3_
            col-xs-12')) !!}
    <div class="col-md-6_
        col-sm-6_col-xs
        -12">
        <?php $bh_casing =
            Helper::
            getValue(old(
                'bh_casing',
                $borehole
            ->
            bh_casing)
            ?>
        {!! Form::text(
            ,
            bh_casing',
            $bh_casing
            , ['class' =>
                'form-
                control_col-
                md-7_col-xs
                -12',
                '
                required' =>
                'true']) !!}
    </div>
</div>

```

```

        'bh_casing'),
        $borehole->
        bh_casing) ?>
    {!! Form::text('
        bh_casing',
        $bh_casing, [
        'class' => '
        form-control_
        col-md-7_col-
        xs-12', '
        required' =>
        'true']) !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label('
        bh_rod', 'Rod_*',
        array('class' =>
        'control-label_
        col-md-3_col-sm-3
        _col-xs-12')) !!}
    <div class="col-md-6_
        col-sm-6_col-xs
        -12">
        <?php $bh_rod =
        Helper::
        getValue(old(
        'bh_rod'),
        $borehole->
        bh_rod) ?>
        {!! Form::text('
        bh_rod',
        $bh_rod, [ '
        class' => '
        form-control_
        col-md-7_col-
        xs-12', '
        required' =>
        'true']) !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label('
        bh_sampler', '
        Sampler_*', array
        ('class' => '
        control-label_col
        -md-3_col-sm-3_
        col-xs-12')) !!}
    <div class="col-md-6_
        col-sm-6_col-xs
        -12">
        <?php $bh_sampler
        = Helper::
        getValue(old(
        'bh_sampler'),
        $borehole->
        bh_sampler)
        ?>
        {!! Form::text('
        bh_sampler',
        $bh_sampler,
        [ 'class' => '
        form-control_
        col-md-7_col-
        xs-12', '
        required' =>
        'true']) !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label('
        bh_water_elevation
        ', 'Water_
        Elevation_(m)_*',
        array('class' =>
        'control-label_
        col-md-3_col-sm-3
        _col-xs-12')) !!}
    <div class="col-md-6_
        col-sm-6_col-xs
        -12">
        <?php
        $bh_water_elevation
        = Helper::
        getValue(old(
        'bh_water_elevation
        '), $borehole
        ->
        bh_water_elevation
        ) ?>
        {!! Form::number('
        bh_water_elevation
        ',
        $bh_water_elevation
        , [ 'class' =>
        'form-
        control_col-
        md-7_col-xs
        -12', '
        required' =>
        'true', '
        step' => 'any'])
        !!}
    </div>
</div>
    md-7_col-xs
    -12', '
    required' =>
    'true', 'step
    ' => 'any'])
    !!}
</div>
<div class="form-group">
    {!! Form::label('
        bh_surface_elevation
        ', 'Surface_
        Elevation_*',
        array('class' =>
        'control-label_
        col-md-3_col-sm-3
        _col-xs-12')) !!}
    <div class="col-md-6_
        col-sm-6_col-xs
        -12">
        <?php
        $bh_surface_elevation
        = Helper::
        getValue(old(
        'bh_surface_elevation
        '), $borehole
        ->
        bh_surface_elevation
        ) ?>
        {!! Form::number('
        bh_surface_elevation
        ',
        $bh_surface_elevation
        , [ 'class' =>
        'form-
        control_col-
        md-7_col-xs
        -12', '
        required' =>
        'true', 'step
        ' => 'any'])
        !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label('
        bh_start_date', '
        Start_Date_End_
        Date_*', [ 'class'
        => 'control-
        label_col-md-3_
        col-sm-3_col-xs
        -12']) !!}
    <div class="col-md-3_
        col-sm-6_col-xs
        -12_form-group">
        <?php
        $bh_start_date
        = Helper::
        getValue(old(
        'bh_start_date
        '), $borehole
        ->
        bh_start_date
        ) ?>
        {!! Form::date('
        bh_start_date
        ',
        $bh_start_date
        , [ 'class' =>
        'form-
        control', '
        required' =>
        'true', '
        placeholder'
        => 'Start_
        Date']) !!}
    </div>
</div>
<div class="col-md-3_
        col-sm-6_col-xs
        -12_form-group">
        <?php $bh_end_date
        = Helper::
        getValue(old(
        'bh_end_date'
        ), $borehole
        ->bh_end_date
        ) ?>
        {!! Form::date('
        bh_end_date',
        $bh_end_date
        , [ 'class' =>
        'form-
        control', '
        required' =>
        'true', '
        placeholder'
        => 'End_
        Date']) !!}
    </div>
</div>

```

```

=> 'End_Date'
]) !!}
</div>
</div>
<div class="ln_solid"></div>
{!! Form::hidden('bh_num',
$borehole->bh_num) !!}
{!! Form::hidden('proj_id',
$project->id) !!}
{!! link_to(URL::previous(), '
Back', array('class' => '
btn_btn-default')) !!}
{!! Form::submit('Submit', ['
class' => 'btn_btn-
primary', 'style' => '
float:right']) !!}
{!! Form::close() !!}
</div>
</div>
</div>
</div>
@endsection

```

Listing 69: boreholes/show.blade.php

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<!-- borehole details -->
<div class="row">
<div class="col-md-12">
<div class="x_panel">
<div class="x_title">
<h2> Borehole {{ $borehole->
bh_num }} <small>(ID#{{
$borehole->id }})</small>
</h2>
<ul class="nav navbar-right
panel_toolbox">
<li><a class="collapse-
link"><i class="fa fa
-chevron-up"></i></a>
</li>
<li class="dropdown">
<a href="#" class="
dropdown-toggle"
data-toggle="
dropdown" role="
button" aria-
expanded="false"
><i class="fa fa-
wrench"></i></a>
<ul class="dropdown-
menu" role="menu"
>
<li><a href="#">
Settings 1</a>
</li>
<li><a href="#">
Settings 2</a>
</li>
</ul>
</li>
</ul>
</div>
<div class="clearfix"></div>
</div>
<div class="x_content">
<div class="col-md-6_col-md-6_
col-xs-12">
<table class="table_table-
striped_table-hover"
cellpadding="10">
<tr>
<?php
$myDateTime =
DateTime::
createFromFormat
('Y-m-d',
$borehole->
bh_start_date
);
$start_date =
$myDateTime->
format('F_d,_
Y');
$myDateTime =
DateTime::
createFromFormat
('Y-m-d',
$borehole->
bh_end_date);
$end_date =

```

```

$myDateTime->
format('F_d,_
Y');
?>
<th scope="row">
Start Date -
End Date</th>
<td>{{ $start_date
}} - {{
$end_date
}}</td>
</tr>
<tr>
<th scope="row">
Weather</th>
<td>{{ $borehole->
bh_weather
}}</td>
</tr>
<tr>
<th scope="row">
Maximum
Borehole
Depth</th>
<td>{{ $borehole->
bh_max_depth
}} m.</td>
</tr>
<tr>
<th scope="row">
Hammer Weight
</th>
<td>{{ $borehole->
bh_hammer_weight
}} kg.</td>
</tr>
<tr>
<th scope="row">
Height of
Drop</th>
<td>{{ $borehole->
bh_height_of_drop
}} m.</td>
</tr>
<tr>
<th scope="row">
Water
Elevation</th>
<td>{{ $borehole->
bh_water_elevation
}} m.</td>
</tr>
<tr>
<th scope="row">
Surface
Elevation</th>
<td>{{ $borehole->
bh_surface_elevation
}} m.</td>
</tr>
</table>
<div class="ln_solid"></div>
</div>
<div class="col-md-6_col-md-6_
col-xs-12">
<table class="table_table-
striped_table-hover"
cellpadding="10">
<tr>
<th scope="row">
Operator</th>
<td>{{ $borehole->
bh_operator
}}</td>
</tr>
<tr>
<th scope="row">
Inspector</th>
<td>{{ $borehole->
bh_inspector
}}</td>
</tr>
<tr>
<th scope="row">
Drill Rig</th>
<td>{{ $borehole->
bh_drill_rig
}}</td>
</tr>
<tr>
<th scope="row">
Drill Pump</th>
<td>{{ $borehole->

```

```

                bh_drill_pump
            }}</td>
        </tr>
        <tr>
            <th scope="row">
                Casing</th>
            <td>{{ $borehole->
                bh_casing
            }}</td>
        </tr>
        <tr>
            <th scope="row">
                Rod</th>
            <td>{{ $borehole->
                bh_rod }}</td>
        </tr>
        <tr>
            <th scope="row">
                Sampler</th>
            <td>{{ $borehole->
                bh_sampler
            }}</td>
        </tr>
    </table>
    <div class="ln_solid"></div>
</div>
<a href="{{_route('boreholes.
edit',_.$borehole->id)_}}"
class="btn btn-info"
style="float:right"><i
class="fa fa-edit_m-right
-xs"></i> Edit Borehole
{{ $borehole->bh_num }} (
ID#{{ $borehole->id }})</a>
<a href="{{_route('samples.
create',_.$borehole->id)_}}"
class="btn btn-
primary" style="float:
right"><i class="fa fa-
plus_m-right-xs"></i> Add
Sample</a>
{!! Form::open(['method' => '
DELETE', 'route' => ['
boreholes.destroy',
$borehole->id], 'onsubmit'
=> 'return_confirm("
Delete borehole '.
$borehole->bh_num . '_in_
project_'. $project->
proj_name . '_permanently
?")') !!}
{!! Form::submit('Delete', ['
class' => 'btn btn-danger
']) !!}
{!! Form::close() !!}
<div class="ln_solid"></div>
</div>
</div>
</div>
<!-- /borehole details -->
<!-- sample details -->
<div class="row">
    <div class="col-md-12">
        <div class="x_panel">
            <div class="x_title">
                <h2> Samples </h2>
                <ul class="nav navbar-right
panel_toolbox">
                    <li><a class="collapse-
link"><i class="fa fa-
chevron-up"></i></a>
                </li>
                    <li class="dropdown">
                        <a href="#" class="
dropdown-toggle"
data-toggle="
dropdown" role="
button" aria-
expanded="false"
><i class="fa fa-
wrench" </i></a>
                        <ul class="dropdown-
menu" role="menu"
>
                            <li><a href="#">
                                Settings 1</a>
                            </li>
                            <li><a href="#">
                                Settings 2</a>
                            </li>
                        </ul>
                    </li>
                </ul>
            </div>

```



```

        info_btn-
        xs"><i
        class=" fa
        _fa-edit _
        m-right _
        xs"></i>
        Edit</a>
    </td>
    <td>
        {!! Form::open
        ([ 'method
        ' => '
        DELETE',
        'route'
        => [ '
        samples.
        destroy',
        $sample
        ->id ], '
        onsubmit'
        => '
        return _
        confirm("
        Delete _
        sample _'.
        $sample
        ->
        sample_num
        . '_in _
        project _'
        .
        $project
        ->
        proj_name
        . '_ _
        permanently
        ?" ')]
        !!}
        {!! Form::
        submit('
        Delete',
        [ 'class'
        => 'btn _
        btn-
        danger _
        btn-xs' ])
        !!}
        {!! Form::
        close ()
        !!}
    </td>
</tr>
endforeach
</tbody>
</table>
<div class="modal_fade_bs-
example-modal-lg"
tabindex="-1" role="
dialog" aria-hidden="true
">
<div class="modal-dialog _
modal-lg">
<div class="modal-
content">
<div class="modal-
header">
<button type="
button"
class="
close"
data-
dismiss="
modal"><
span aria-
hidden="
true">
</span>
</button>
<h4 class="
modal-
title" id
="
myModalLabel
"><span
class="
sample_title
"></span
></h4>
</div>
<div class="modal-
body_edit _
content">
<div class="
col-md-6 _
col-md-6 _
col-xs-12
">

```

```

<
  td
  <
  span
  >
  class
  =
  "
  uscs_description
  ></
  span
  ></
  td
  >
  </tr>
<tr>
<
  th
  scope
  =
  "
  row
  "
  >
  Contract
  No
  .</
  th
  >
  <
  td
  <
  span
  >
  class
  =
  "
  sample_contract_num
  "
  ></
  span
  ></
  td
  >
  </tr>
<tr>
<
  th
  scope
  =
  "
  row
  "
  >
  Job
  No
  .</
  th
  >
  <
  td
  <
  span
  >
  class
  =
  "
  sample_job_num
  "
  ></
  span
  ></
  td
  >
  </tr>
<tr>
<
  th
  scope
  =
  "
  row
  "
  >
  Soil
  Color
  </
  th
  >
  </tr>
  <
  td
  <
  span
  >
  class
  =
  "
  sample_color
  "
  ></
  span
  ></
  td
  >
  </tr>
  <
  th
  scope
  =
  "
  row
  "
  >
  Length
  of
  Run
  </
  th
  >
  <
  td
  <
  span
  >
  class
  =
  "
  sample_run_len
  "
  ></
  span
  ></
  td
  >
  </tr>
  <
  th
  scope
  =
  "
  row
  "
  >
  Length
  of
  Sample
  </
  th
  >
  <
  td
  <
  span
  >
  class
  =
  "
  sample_sample_
  "
  ></
  span
  ></
  td
  >
  </tr>
  <
  th
  scope
  =
  "

```

```

row
"
>
Rec
.</
th
"
>
<
td
<
span
"
class
"
sample_rec_percent
"
></
span
></
td
"
>
</tr>
</table>
<div class
"
ln_solid
"></
div>
</div>
<div class="
col-md-6_
col-md-6_
col-xs-12
">
<table
class
"
table
-
table
-
striped
-
table
-
hover
"
cellpadding
=" 10"
">
<tr>
<
th
"
scope
"
row
"
">
No
.
of
Blows
(
SPT
N
-
Value
)
">
th
"
">
td
<
span
"
class
"
sample_num_of_blows
"
"></
span
"></
td
"
">
</tr>
<tr>

```

```

</tr>
<tr>
<
th
scope
=
"
row
"
>
Plastic
Limit
</
th
>
<
td
<
span
class
=
"
sample_plastic_limit
"
></
span
></
td
>
</tr>
<tr>
<
th
scope
=
"
row
"
>
Plasticity
Index
</
th
>
<
td
<
span
class
=
"
sample_plasticity_index
"
></
span
></
td
>
</tr>
<tr>
<
th
scope
=
"
row
"
>
Plasticity
Remark
</
th
>
<
td
<
span
class
=
"
sample_plasticity_remark
"
></
span
></
td
>
></
td
>
</tr>
<tr>
<
th
scope
=
"
row
"
>
Liquid
and
Plastic
Limits
Tester
</
th
>
<
td
<
span
class
=
"
sample_tester
"
></
span
></
td
>
</tr>
<tr>
<
th
scope
=
"
row
"
>
Other
Test
Results
</
th
>
<
td
<
span
class
=
"
sample_other_r
"
></
span
></
td
>
</tr>
<tr>
<
th
scope
=
"
row
"
>
Test
Date
</
th
>
<
td
<
span
class
=
"
sample_date
"
></
span
></
td
>

```



```

<table class="table_table-
  hover" id="table{{$_$i
  -}}">
  <thead>
    <tr>
      <th>Sample</th>
      <th>Grain
        Diameter
        (mm)</th>
      <th>Percentage
        Finer</
        th>
      <th width="18%
        ">Data
        Points
        Options</
        th>
    </tr>
  </thead>
  <tbody>
    @for($j = 0; $j <
      sizeof(
        $counts[$i]);
        $j++)
      <tr>
        <th scope="
          row"
          style
            ="
            border
            -top
            :1px_
            solid
            ;">
          {{ $j
            +
              1
            }}
          <input
            type
              ="
              hidden
            "
            name
              ="
              sample_id
            ["
            "
            value
              ="
              {{
                $counts
                [
                $i
                ]{
                $j
                }
                }
              "
            >
          <small
            >{!!
              link_to_route
                (
                  samples
                  .
                  show
                  ,
                  ,
                  (
                    ID
                    #
                    ,
                    .
                    $counts
                    [
                    $i
                    ]{
                    $j
                    }

```

```

]
.
,
)
,
,
$counts
[
$i
]{
$j
},
array
(
,
class
,
=>
,
to_login
,
)
)
!!}</
small
>
</th>
<td style="
border
-top
:1px_
solid
;"><
input
type
="
number
"
name="
x_int
["
class
="
form-
control
-
to_clear
_col-
md-7_
col-
xs-12
"
step=
"any"
required
></td
>
<td style="
border
-top
:1px_
solid
;"><
input
type
="
number
"
name="
y_int
["
class
="
form-
control
-
to_clear
_col-
md-7_
col-
xs-12
"
step=
"any"
required
></td

```



```

step
=,
any
'></
td
>
"
+
"
<
td
>
input
-
type
=,
number
,
-
name
=,
y_int
[],
-
class
=,
form
-
control
-
to_clear
-
col
-
md
-7
-
col
-
xs
-12'
-
step
=,
any
'></
td
>
"
+
"
<
td
>
button
-
type
=,
button
,
-
onclick
=,
addRow
(
this
,
-
"
+ "<button_type=
button'_onclick_=
removeRow(this ,
" + class_num +
" , " + sample_id
+ " );'_class='btn
_btn-danger_btn-
sm'_style='float :
center;_width:70
px;'><i_class='fa
_fa-trash-o_m-
right-sm'></i>-
Delete_</button
></td>"
+ "</
tr
>
"
class_num
)
;
+
"
// Delete previous add button
$(".previous").closest("button
").remove();
-
}
+
function removeRow (parameter ,
class_num , sample_id) {
sample_id var current_row = $(parameter)
.closest("tr");
+ var current_row_td =
current_row.prev("tr").
find("td:last");
)
;
// Append add button to
previous row if current
row is last row of sample
id
"
if ($(parameter).closest("tr")

```



```

        .find("button").hasClass(
            "addBtn") {
            current_row_td.append("<
                button_type='button'
                onclick='addRow(this,
                _ + class_num + ",
                + sample_id + ");'
                class=' + class_num
                + "_btn_btn-info_btn-
                sm_addBtn'
                style='
                float:left; width:70
                px;'
                <i class='fa-fa-
                plus_m-right-xs'></i>
                _Add</button>");
        }

        // Remove row
        current_row.remove();
    }

    //});
</script>
</endsection>

```

Listing 71: datapoints/edit.blade.php

```

<@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
    <div class="col-md-12">
        <div class="x_panel">
            <div class="x_title">
                <h2>Edit Grain Size Analysis</h2>
            <ul class="nav navbar-right panel-toolbox">
                <li><a class="collapse-link"
                    <i class="fa-fa-chevron-up"></i></a>
                </li>
                <li class="dropdown">
                    <a href="#" class="dropdown-toggle"
                        data-toggle="dropdown"
                        role="button"
                        aria-expanded="false"
                        <i class="fa-fa-wrench"></i></a>
                    <ul class="dropdown-menu"
                        role="menu">
                        <li><a href="#">
                            Settings 1</a>
                        </li>
                        <li><a href="#">
                            Settings 2</a>
                        </li>
                    </ul>
                </li>
            </ul>
            <div class="clearfix"></div>
            <div class="x_content">
                @if(sizeof($boreholes) == 0)
                <h4 align="center">There are no existing samples.</h4>
                @else
                {!! Form::open(['name' => 'update_form', 'method' => 'PUT', 'route' => ['datapoints.update', $subreport->id], 'class' => 'form-horizontal_form-label-left', 'id' => 'update_form']) !!}
                <?php
                    $class_num = 0;
                    $i = 0;
                    ?>
                @foreach($boreholes as $borehole)
                <h4 align="center">
                    Borehole {{ $borehole->bh_num }}
                <small>{!! link_to_route('boreholes.show', '(ID #' . $borehole->id . ')', $borehole->id, array('class' => 'to_login')) !!}</small>
            </div>
        </div>
    </div>
</endsection>

```

```

</h4>
<table class="table table-hover" id="table_{{$_i}}">
    <thead>
        <tr>
            <th>Sample</th>
            <th>Grain Diameter (mm)</th>
            <th>Percentage Finer</th>
        </tr>
    </thead>
    <tbody>
        <?php $prev_sample_id = 0 ?>
        @foreach($samples as $sample)
        @if($sample->bh_id == $borehole->id)
        @foreach($data_points as $data_point)
        @if($data_point->sample_id == $sample->id)
        @if($prev_sample_id != $sample->id)
        <?php $td_style = "border-top:1px_solid;" ?>
        @else
        <?php $td_style = "" ?>
        @endif
        <tr>
            <th scope="row" style="{{$_td_style}}">
                @if($prev_sample_id != $sample->id)
                {!! $sample->sample_num !!}
            <small>{!! link_to_route('samples.show', '(ID#' . $sample->id . ')', $sample->id, array('class' => 'to_login')) !!}</small>
            {!! Form::hidden('sample_id[]', $sample->id) !!}
            {!! Form::hidden('data_point_id[]', $data_point->id) !!}
        @else
        {!! Form::hidden('sample_id[]', $sample->id) !!}
        {!! Form::hidden('data_point_id[]', $data_point->id) !!}
        @endif
        </th>
            <td style="{{$_td_style}}">
                <input type="number" name="x_int[]" class="form-control" to_clear_col-md-7_col-xs-12" step="

```

```

any" value="
{{-
$data_point->
x_int_}}
required></td
>
<td style="{{-
$td_style_}}"
><input type="
number" name
=" y_int []"
class="form-
control_
to_clear_col-
md-7_col-xs
-12" step="
any" value="
{{-
$data_point->
y_int_}}
required></td
>
</tr>
<?php $prev_sample_id
= $sample->id ?>
@endif
@endforeach
@endif
<?php $class_num++ ?>
@endforeach
</tbody>
</table>
<?php $i++ ?>
@endforeach
<div class="ln_solid"></div>
{!! Form::hidden('proj_id',
$project->id) !!}
{!! link_to(URL::previous(), '
Back', array('class' => '
btn_btn-default')) !!}
{!! Form::button('Clear', ['
class' => 'btn_btn-danger',
'onclick' => '
clearForm()']) !!}
{!! Form::submit('Submit', ['
class' => 'btn_btn-
primary', 'style' => '
float:right']) !!}
{!! Form::close() !!}
@endif
</div>
</div>
</div>
</div>
@endsection
@section('js')
@parent
@include('includes/clearform')
@endsection

```

Listing 72: datapoints/index.blade.php

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
<div class="col-md-12">
<div class="x_panel">
<div class="x_title">
<h2>Grain Size Analysis - Data
Points</h2>
<ul class="nav navbar-right_
panel_toolbox">
<li><a class="collapse-
link"><i class="fa_fa-
-chevron-up"></i></a>
</li>
<li class="dropdown">
<a href="#" class="
dropdown-toggle"
data-toggle="
dropdown" role="
button" aria-
expanded="false"
><i class="fa_fa-
wrench"></i></a>
<ul class="dropdown-
menu" role="menu"
>
<li><a href="#">
Settings 1</a>
</li>
<li><a href="#">
Settings 2</a>
</li>
</ul>
</li>
</ul>

```

```

</ul>
</li>
</ul>
<div class="clearfix"></div>
</div>
<div class="x_content">
@if(sizeof($boreholes) == 0)
<h4 align="center">There are
no existing samples.</h4>
@else
<?php $i = 0 ?>
@foreach($boreholes as
$borehole)
<h4 align="center">
Borehole {{ $borehole->
bh_num }}
<small>{!! link_to_route(
'boreholes.show', '(ID
#' . $borehole->id .
')', $borehole->id,
array('class' => '
to_login')) !!}</
small>
</h4>
<table class="table_table-
hover" id="table{{- $i_}}"
>
<thead>
<tr>
<th>Sample</th>
<th>Grain Diameter
(mm)</th>
<th>Percentage
Finer</th>
</tr>
</thead>
<tbody>
<?php $prev_sample_id
= 0 ?>
@foreach($samples as
$sample)
@if($sample->bh_id ==
$borehole->id)
@foreach($data_points
as $data_point)
@if($data_point->
sample_id ==
$sample->id)
@if($prev_sample_id !=
$sample->id)
<?php $td_style = "
border-top:1px_
solid;" ?>
@else
<?php $td_style = ""
?>
@endif
<tr>
<th scope="row"
style="{{-
$td_style_}}"
>
@if(
$prev_sample_id
!=
$sample->
id)
{{ $sample->
sample_num
}}
<small>{!!
link_to_route(
'samples
.show', '(ID#',
$sample->
id . ')',
$sample
->id,
array('
class' =>
,
to_login'
)) !!}</
small>
@endif
</th>
<td style="{{-
$td_style_}}"
>{{
$data_point->
x_int }}</td>
<td style="{{-
$td_style_}}"
>{{
$data_point->
y_int }}</td>
</tr>
<?php $prev_sample_id

```

```

                = $sample->id ?>
                @endif
            @endforeach
            @endif
            @endforeach
        </tbody>
    </table>
    <?php $i++ ?>
    @endforeach
    <div class="ln_solid"></div>
    {!! Form::open(['method' => '
        DELETE', 'route' => ['
            datapoints.destroy',
            $subreport->id], '
            onsubmit' => 'return _
            confirm(" Delete _grain _
            size _analysis _permanently
            ?") ']) !!}
    {!! Form::submit('Delete', ['
        class' => 'btn btn-danger
        ']) !!}
    <a href="{{{_route('datapoints.
        edit', _$project->id)_}}}"
        class="btn btn-info"
        style="float:right"><i
        class="fa fa-edit _m-right
        -xs"></i> Edit </a>
    {!! Form::close() !!}
    @endif
    </div>
</div>
</div>
</div>
@endsection

```

Listing 73: deeps/calculate.blade.php

```

    @include('includes/clearform')

    <!-- tooltip -->
    <script>
    $(document).ready(function(){
        $('[data-toggle="tooltip"]').tooltip()
    });
    </script>
    <!-- /tooltip -->

    <!-- calculate -->
    <script type="text/javascript">
    var num_of_boreholes = <?php echo
        sizeof($boreholes) ?>;
    var num_of_samples;
    var multiplier, df, k, gamma, tan, f;
    var pile_size, asurface, atip;
    var q, qtotal;
    var fs, q_kn, q_ton;
    var fc, rho, fy, cs_kn, cs_ton;

    const ls = 3.5; // 10x dia of pile for
        loose sand
    const dcs = 7; // 20x dia of pie for
        dense compact sand

    function calculate() {
        for(i = 1; i <= num_of_boreholes;
            i++) {
            num_of_samples = $(''
                deep_depth_' + i).length;
            for(j = 1; j <= num_of_samples
                ; j++) {

                // Compute f
                df = $(''#deep_depth_' + i
                    + '_' + j).val();
                k = $(''#deep_K_' + i).val
                    ();
                gamma = $(''#deep_gamma_' +
                    i).val();
                tan = $(''#deep_tan_' + i).
                    val();

                if (df < dcs) {
                    multiplier = df;
                }
                else {
                    multiplier = dcs;
                }

                f = multiplier*k*gamma*tan
                    ;
                $(''#deep_f_' + i + '_' + j
                    ).val(parseFloat(f.
                        toFixed(4)));

                // Compute Asurface
                pile_size = $(''#

```

```

                deep_pile_size_' + i)
                .val();

                asurface = 4*pile_size*df;
                $(''#deep_Asurface_' + i +
                    '_' + j).val(
                    parseFloat(asurface.
                        toFixed(4)));

                // Compute Atip
                atip = pile_size*pile_size
                    ;
                $(''#deep_Atip_' + i + '_'
                    + j).val(parseFloat(
                        atip.toFixed(4)));

                // Compute Qtotal
                q = $(''#deep_q_' + i + '_'
                    + j).val();

                qtotal = f*asurface+q*atip
                    ;
                $(''#deep_Qtotal_' + i + '_'
                    + j).val(parseFloat(
                        qtotal.toFixed(4)));

                // Compute Q (kN)
                fs = $(''#deep_fs_' + i).
                    val();

                q_kn = qtotal/fs;
                $(''#deep_Q_kn_' + i + '_'
                    + j).val(parseFloat(
                        q_kn.toFixed(4)));

                // Compute Q (ton)
                //Q_kn*0.10219

                q_ton = q_kn*0.10219;
                $(''#deep_Q_ton_' + i + '_'
                    + j).val(parseFloat(
                        q_ton.toFixed(4)));

                // Compute concrete
                strength (kN)
                fc = $(''#deep_fc_' + i).
                    val();
                rho = $(''#deep_rho_' + i).
                    val();
                fy = $(''#deep_fy_' + i).
                    val();

                cs_kn = (((pile_size*
                    pile_size)
                    *(1000*1000)
                    *0.85*0.7*fc)+((
                    pile_size*pile_size)
                    *(1000*1000)*rho*fy)
                    -(0.85*0.7*(pile_size
                    *pile_size)
                    *(1000*1000)*rho*fc))
                    /1000;
                $(''#
                    deep_concrete_strength_kn-
                    ' + i).val(parseFloat(
                        cs_kn.toFixed(4)));

                // Compute concrete
                strength (ton)
                cs_ton = cs_kn*0.10219;
                $(''#
                    deep_concrete_strength_ton-
                    ' + i).val(parseFloat(
                        cs_ton.toFixed(4)));

            }
        }
    }
    </script>
    <!-- /calculate -->

```

Listing 74: deeps/create.blade.php

```

    @extends('layouts/master')
    @section('title', $project->proj_name)
    @section('content')
    <div class="row">
        <div class="col-md-12">
            <div class="x_panel">
                <div class="x_title">
                    <h2>Deep Foundation <small>
                        Compute soil bearing
                        capacity for deep
                        foundation </small></h2>
                <ul class="nav navbar-right _
                    panel_toolbox">
                    <li><a class="collapse _
                        link"><i class="fa fa

```

```

        -chevron-up"></i></a>
</li>
<li class="dropdown">
  <a href="#" class="
    dropdown-toggle"
    data-toggle="
    dropdown" role="
    button" aria-
    expanded="false"
    ><i class="fa_fa-
    wrench"></i></a>
  <ul class="dropdown-
    menu" role="menu"
    >
    <li><a href="#">
      Settings 1</a>
    </li>
    <li><a href="#">
      Settings 2</a>
    </li>
  </ul>
</li>
</ul>
<div class="clearfix"></div>
<div class="x_content">
  @if(sizeof($samples) > 0)
  <?php $soil = ["clay", "silt"]
  ?>
  {!! Form::open(['name' => '
    store_form', 'method' =>
    'POST', 'route' => 'deeps
    .store', 'class' => 'form
    -horizontal_form-label-
    left']) !!}
  <?php $i = 0 ?>
  @foreach($boreholes as
    $borehole)
    {!! Form::hidden('bh.id[]',
    $borehole->id) !!}
    {!! Form::hidden('
    num_of_samples[]',
    $borehole->
    bh_num_of_samples) !!}
  <center>
    <h2>Borehole {{ $borehole
    ->bh_num }} <small>(
    ID#{{ $borehole->id
    }})</small></h2>
  </center>
  <table class="table_table-
    striped_table-hover">
    <thead>
    <tr>
    <th><a data-toggle
    ="tooltip"
    title=" ">Pile
    Size</a></th>
    <th><a data-toggle
    ="tooltip"
    title=" ">K</a>
    </th>
    <th><a data-toggle
    ="tooltip"
    title=" ">&
    gamma</a></
    th>
    <th><a data-toggle
    ="tooltip"
    title=" ">tan&
    delta</a></
    th>
    <th><a data-toggle
    ="tooltip"
    title=" ">Factor
    _of_safety">
    FS</a></th>
    <th><a data-toggle
    ="tooltip"
    title=" ">&rho
    ;</a></th>
    <th><a data-toggle
    ="tooltip"
    title=" ">fy</
    a></th>
    <th><a data-toggle
    ="tooltip"
    title=" ">fc</
    a></th>
    <th><a data-toggle
    ="tooltip"
    title=" ">
    Concrete
    Strength (kN)
    </a></th>
    <th><a data-toggle
    ="tooltip"
    title=" ">
    Concrete
    Strength (ton
    )</a></th>
  </thead>
  <tbody>
  <tr>
  <td><input type="
  number" name="
  deep_pile_size
  []" class="form-
  control_
  to_clear"
  step="any"
  value="{{_old
  ('
  deep_pile_size
  ')_}}" id="
  deep_pile_size_
  {{_borehole
  ->bh_num_}}">
  </td>
  <td><input type="
  number" name="
  deep-K[]"
  class="form-
  control_
  to_clear"
  step="any"
  value="{{_old
  ('deep_K')_}}
  " id="deep_K_
  {{_borehole
  ->bh_num_}}">
  </td>
  <td><input type="
  number" name="
  deep_gamma[]"
  class="form-
  control_
  to_clear"
  step="any"
  value="{{_old
  ('deep_gamma
  ')_}}" id="
  deep_gamma_{{
  _borehole->
  bh_num_}}">
  </td>
  <td><input type="
  number" name="
  deep_tan[]"
  class="form-
  control_
  to_clear"
  step="any"
  value="{{_old
  ('deep_tan')_
  }}" id="
  deep_tan_{{_
  $borehole->
  bh_num_}}">
  </td>
  <td><input type="
  number" name="
  deep_fs[]"
  class="form-
  control_
  to_clear"
  step="any"
  value="{{_old
  ('deep_fs')_
  }}" id="
  deep_fs_{{_
  $borehole->
  bh_num_}}">
  </td>
  <td><input type="
  number" name="
  deep_rho[]"
  class="form-
  control_
  to_clear"
  step="any"
  value="{{_old
  ('deep_rho')_
  }}" id="
  deep_rho_{{_
  $borehole->
  bh_num_}}">
  </td>
  </tr>
  </tbody>
  </table>
  </div>

```

```

>
<td><input type="
number" name=
" deep_fy [] "
class="form-
control_
to_clear_
step="any"
value="{{_old
(' deep_fy ' )_
}}" id="
deep_fy_{{_
$borehole->
bh_num_}}"
required></td
>
<td><input type="
number" name=
" deep_fc [] "
class="form-
control_
to_clear_
step="any"
value="{{_old
(' deep_fc ' )_
}}" id="
deep_fc_{{_
$borehole->
bh_num_}}"
required></td
>
<td><input type="
number" name=
"
deep_concrete_strength_kn
[]" class="
form-control_
to_clear_
deep_concrete_strength_kn_
{{_ $borehole
->bh_num_}}"
step="any" id
="
deep_concrete_strength_kn_
{{_ $borehole
->bh_num_}}"
readonly
required></td
>
<td><input type="
number" name=
"
deep_concrete_strength_ton
[]" class="
form-control_
to_clear_
deep_concrete_strength_ton_
{{_ $borehole
->bh_num_}}"
step="any" id
="
deep_concrete_strength_ton_
{{_ $borehole
->bh_num_}}"
readonly
required></td
>
</tr>
</tbody>
</table>
<table class="table_table-
striped_table-hover">
<thead>
<tr>
<th><a data-toggle
=" tooltip"
title="Depth_
of_footing_
below_lowest_
adjacent_soil_
_surface">D<
sub>f</sub></a></th>
<th><a data-toggle
=" tooltip"
title="">f</a>
</th>
<th><a data-toggle
=" tooltip"
title="">A<
sub>surface</
sub></a></th>
<th><a data-toggle
=" tooltip"
title="">q</a>
</th>
<th><a data-toggle
=" tooltip"
title="">A<
sub>tip</sub></a></th>
</thead>
<tbody>
<tr>
<td><input type="
number" name=
"
deep_Asurface
[]" class="
form-control_
to_clear_
deep_Asurface_
{{_ $borehole
->bh_num_}}"
step="any" id
="
deep_Asurface_
{{_ $borehole
->bh_num_}}_
{{_ $sample->
sample_num_}}"
readonly
required></td
>
<td><input type="
number" name=
" deep_q [] "
class="form-
control_
to_clear_
deep_q_{{_
$borehole->
bh_num_}}_{{_
$sample->
sample_num_}}
" id="
deep_q_{{_
$borehole->
bh_num_}}_{{_
$sample->
sample_num_}}
"></td>
</tr>
</tbody>
</table>
</tr>
</thead>
<tbody>
@foreach( $samples as
$sample)
@if( $sample->bh_id ==
$borehole->id)
{!! Form::hidden('
sample_id [] ',
$sample->id) !!}
<?php (sizeof(
$shallows) > 0) ?
$q = $shallows[
$i]->shallow_q :
$q = "" ?>
<tr>
<td><input type="
number" name=
" deep_depth [] "
class="form-
control_
deep_depth_{{
_ $borehole->
bh_num_}}"
step="any"
value="{{_
$sample->
sample_to_}"
id="
deep_depth_{{
_ $borehole->
bh_num_}}_{{_
$sample->
sample_num_}}
" readonly></
td>
<td><input type="
number" name=
" deep_f [] "
class="form-
control_
to_clear_
deep_f_{{_
$borehole->
bh_num_}}_{{_
$sample->
sample_num_}}
" id="
deep_f_{{_
$borehole->
bh_num_}}_{{_
$sample->
sample_num_}}
"></td>
<td><input type="
number" name=
"
deep_Asurface
[]" class="
form-control_
to_clear_
deep_Asurface_
{{_ $borehole
->bh_num_}}"
step="any" id
="
deep_Asurface_
{{_ $borehole
->bh_num_}}_
{{_ $sample->
sample_num_}}
" readonly
required></td
>
<td><input type="
number" name=
" deep_q [] "
class="form-
control_
to_clear_
deep_q_{{_
$borehole->
bh_num_}}_{{_
$sample->
sample_num_}}
" id="
deep_q_{{_
$borehole->
bh_num_}}_{{_
$sample->
sample_num_}}
"></td>
</tr>
</tbody>
</table>

```

```

        deep-q-{{-
        $borehole->
        bh_num_}}"
        step="any"
        value="{{-$q-
        }}" id="
        deep-q-{{-
        $borehole->
        bh_num_}}-{{-
        $sample->
        sample_num_}}
        " readonly
        required></td
        >
<td><input type="
        number" name=
        "deep-Atip[]"
        class="form-
        control-
        to-clear-
        deep-Atip-{{-
        $borehole->
        bh_num_}}"
        step="any" id
        ="deep-Atip-
        {{-$borehole
        ->bh_num_}}-
        {{-$sample->
        sample_num_}}
        " readonly
        required></td
        >
<td><input type="
        number" name=
        "deep-Qtotal
        []" class="
        form-control-
        to-clear-
        deep-Qtotal-
        {{-$borehole
        ->bh_num_}}"
        step="any" id
        ="
        deep-Qtotal-
        {{-$borehole
        ->bh_num_}}-
        {{-$sample->
        sample_num_}}
        " readonly
        required></td
        >
<td><input type="
        number" name=
        "deep-Q-qn[]"
        class="form-
        control-
        to-clear-
        deep-Q-qn-{{-
        $borehole->
        bh_num_}}"
        step="any" id
        ="deep-Q-qn-
        {{-$borehole
        ->bh_num_}}-
        {{-$sample->
        sample_num_}}
        " readonly
        required></td
        >
<td><input type="
        number" name=
        "deep-Q-ton[]"
        class="form-
        control-
        to-clear-
        deep-Q-ton-{{
        -$borehole->
        bh_num_}}"
        step="any" id
        ="deep-Q-ton-
        {{-$borehole
        ->bh_num_}}-
        {{-$sample->
        sample_num_}}
        " readonly
        required></td
        >
<td style="text-
        align:center;
        "><input type
        ="checkbox"
        name="
        deep-is-chosen
        []" value="{{
        -$sample->id-
        }}"></td>
</tr>
<?php $i++ ?>
@endif
    @endforeach
</tbody>
</table>
<br/>
@foreach
<div class="ln_solid"></div>
{!! Form::hidden('proj_id',
        $project->id) !!}
{!! Form::hidden('
        deep_is_chosen[]', -1)
        !!}
{!! link_to(URL::previous(), '
        Back', array('class' => '
        btn btn-default')) !!}
{!! Form::button('Clear', ['
        class' => 'btn btn-danger'
        , 'onclick' => '
        clearForm()']) !!}
{!! Form::submit('Submit', ['
        class' => 'btn btn-
        primary', 'style' => '
        float:right', 'onclick'
        => 'calculate()']) !!}
{!! Form::button('Calculate',
        ['class' => 'btn btn-
        success', 'style' => '
        float:right', 'onclick'
        => 'calculate()']) !!}
{!! Form::close() !!}
@endif
</div>
</div>
</div>
@endsection
@section('js')
@parent
@include('deeps/calculate')
@endsection

```

Listing 75: deeps/edit.blade.php

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
    <div class="col-md-12">
        <div class="x_panel">
            <div class="x_title">
                <h2>Edit - Deep Foundation <
                small>Compute soil
                bearing capacity for deep
                foundation</small></h2>
                <ul class="nav navbar-right-
                panel_toolbox">
                    <li><a class="collapse-
                    link"><i class="fa fa-
                    chevron-up"></i></a>
                    </li>
                    <li class="dropdown">
                        <a href="#" class="
                        dropdown-toggle"
                        data-toggle="
                        dropdown" role="
                        button" aria-
                        expanded="false"
                        ><i class="fa fa-
                        wrench"></i></a>
                        <ul class="dropdown-
                        menu" role="menu"
                        >
                            <li><a href="#">
                            Settings 1</a>
                            </li>
                            <li><a href="#">
                            Settings 2</a>
                            </li>
                        </ul>
                    </li>
                </ul>
            </div>
            <div class="clearfix"></div>
            <div class="x_content">
                @if(sizeof($deeps) > 0)
                <?php $soil = ["clay", "silt"]
                ?>
                {!! Form::open(['name' => '
                update_form', 'method' =>
                'PUT', 'route' => [
                'deeps.update', $project->
                id], 'class' => 'form-
                horizontal form-label-
                left', 'id' => '
                update_form']) !!}
                @foreach($boreholes as

```

```

    $borehole)
<center>
  <h2>Borehole {{ $borehole
    ->bh_num }} <small>(
    ID#{{ $borehole->id
    }})</small></h2>
</center>
<table class="table table-
  striped table-hover">
  <thead>
    <tr>
      <th><a data-toggle
        =" tooltip"
        title=" ">Pile
        Size </a></th>
      <th><a data-toggle
        =" tooltip"
        title=" ">K</a>
      </th>
      <th><a data-toggle
        =" tooltip"
        title=" ">&
        gamma;</a></th>
      <th><a data-toggle
        =" tooltip"
        title=" ">tan&
        delta;</a></th>
      <th><a data-toggle
        =" tooltip"
        title=" ">Factor
        _of_safety">
        FS</a></th>
      <th><a data-toggle
        =" tooltip"
        title=" ">&rho;
      </a></th>
      <th><a data-toggle
        =" tooltip"
        title=" ">fy</a>
      </th>
      <th><a data-toggle
        =" tooltip"
        title=" ">fc</a>
      </th>
      <th><a data-toggle
        =" tooltip"
        title=" ">
        Concrete
        Strength (kN)
      </a></th>
      <th><a data-toggle
        =" tooltip"
        title=" ">
        Concrete
        Strength (ton
        )</a></th>
    </tr>
  </thead>
  <tbody>
    <?php $i = 0 ?>
    @while($borehole->id
      != $deeps[$i]->
      bh_id)
    <?php $i++ ?>
    @endwhile
    <tr>
      <td>
        <?php
          $deep_pile_size
          = Helper
          ::
          getValue(
            old('
            deep_pile_size
            '),
            $deeps[$i
            ]->
            deep_pile_size
            ) ?>
          <input type="
            number"
            name="
            deep_pile_size
            []" class
            =" form-
            control_
            to_clear"
            step="
            any"
            value="{{
            _(float)
            $deep_pile_size
            _}}" id="
            deep_pile_size_
            {{-
            $borehole
            ->bh_num_
            }}"
            required>
      </td>
      ->bh_num_
      }}"
      required>
    </td>
    <?php $deep_K
    = Helper
    ::
    getValue(
      old('
      deep_K'),
      $deeps[
      $i]->
      deep_K)
    ?>
    <input type="
    number"
    name="
    deep_K[]"
    class="
    form-
    control_
    to_clear"
    step="
    any"
    value="{{
    _(float)
    $deep_K_
    }}" id="
    deep_K-{{
    -
    $borehole
    ->bh_num_
    }}"
    required>
    </td>
    <?php
      $deep_gamma
      = Helper
      ::
      getValue(
        old('
        deep_gamma
        '),
        $deeps[$i
        ]->
        deep_gamma
        ) ?>
    <input type="
    number"
    name="
    deep_gamma
    []" class
    =" form-
    control_
    to_clear"
    step="
    any"
    value="{{
    _(float)
    $deep_gamma
    _}}" id="
    deep_gamma_
    {{-
    $borehole
    ->bh_num_
    }}"
    required>
    </td>
    <?php
      $deep_tan
      = Helper
      ::
      getValue(
        old('
        deep_tan
        '), $deeps
        [$i]->
        deep_tan)
      ?>
    <input type="
    number"
    name="
    deep_tan
    []" class
    =" form-
    control_
    to_clear"
    step="
    any"
    value="{{
    _(float)
    $deep_tan
    _}}" id="
    deep_tan_
    {{-
    $borehole
    ->bh_num_
    }}"
    required>
    </td>
  </tr>
</tbody>
</table>

```

```

->bh.num_
})"
required>
</td>
<td>
<?php $deep_fs
= Helper
::
getValue(
old('
deep_fs ')
, $deeps [
$i]->
deep_fs)
?>
<input type="
number"
name="
deep_fs []"
class="
form-
control_
to_clear"
step="
any"
value="{ {
_(float)
$deep_fs_
}" id="
deep_fs_
{ { _
$borehole
->bh.num_
}" }"
required>
</td>
<td>
<?php
$deep_rho
= Helper
::
getValue(
old('
deep_rho '
), $deeps
[$i]->
deep_rho)
?>
<input type="
number"
name="
deep_rho
[]" class="
form-
control_
to_clear"
step="
any"
value="{ {
_(float)
$deep_rho
_}" id="
deep_rho_
{ { _
$borehole
->bh.num_
}" }"
required>
</td>
<td>
<?php $deep_fy
= Helper
::
getValue(
old('
deep_fy ')
, $deeps [
$i]->
deep_fy)
?>
<input type="
number"
name="
deep_fy []"
class="
form-
control_
to_clear"
step="
any"
value="{ {
_(float)
$deep_fy_
}" id="
deep_fy_
{ { _
$borehole
->bh.num_
}" }"
required>
</td>
required>
</td>
<td>
<?php $deep_fc
= Helper
::
getValue(
old('
deep_fc ')
, $deeps [
$i]->
deep_fc)
?>
<input type="
number"
name="
deep_fc []"
class="
form-
control_
to_clear"
step="
any"
value="{ {
_(float)
$deep_fc_
}" id="
deep_fc_
{ { _
$borehole
->bh.num_
}" }"
required>
</td>
<td><input type="
number" name="
"
deep_concrete_strength_kn
[]" class="
form-control_
to_clear_
deep_concrete_strength_kn_
{ { _$borehole
->bh.num_}" }"
step="any"
value="{ {_(
float) $deeps [
$i]->
deep_concrete_strength_kn
_}" id="
deep_concrete_strength_kn_
{ { _$borehole
->bh.num_}" }"
readonly
required></td>
<td><input type="
number" name="
"
deep_concrete_strength_ton
[]" class="
form-control_
to_clear_
deep_concrete_strength_ton_
{ { _$borehole
->bh.num_}" }"
step="any"
value="{ {_(
float) $deeps [
$i]->
deep_concrete_strength_ton
_}" id="
deep_concrete_strength_ton_
{ { _$borehole
->bh.num_}" }"
readonly
required></td>
>
</tr>
</tbody>
</table>
<table class="table_table-
striped_table-hover">
<thead>
<tr>
<th><a data-toggle
="tooltip"
title="Depth_
of_footing_
below_lowest_
adjacent_soil_
_surface">D<
sub>f</sub></a></th>
<th><a data-toggle
="tooltip"
title="">f</a>
</th>
<th><a data-toggle

```



```

        = " tooltip"
        title=" ">A<
        sub>surface</
        sub></a></th>
<th><a data-toggle
= " tooltip"
title=" ">q</a
></th>
<th><a data-toggle
= " tooltip"
title=" ">A<
sub>tip</sub
></a></th>
<th><a data-toggle
= " tooltip"
title=" ">Q<
sub>total</
sub></a></th>
<th><a data-toggle
= " tooltip"
title=" ">Q (
kN)</a></th>
<th><a data-toggle
= " tooltip"
title=" ">Q (
ton)</a></th>
<th><a data-toggle
= " tooltip"
title=" ">
Choose</a></
th>
</tr>
</thead>
<tbody>
<?php $j = 0; $k = 1;
?>
@foreach ($deeps as
$deep)
@if($deep->bh_id ==
$borehole->id)
{!! Form::hidden('
sample_id[]',
$deep->sample_id)
!!}
{!! Form::hidden('
deep_id[]', $deep
->id) !!}
<?php ($sizeof(
$shallows) > 0) ?
$q = $shallows[
$j]->shallow_q :
$q = " " ?>
<tr>
<td><input type="
number" name="
deep_depth[]"
class="form-
control_
deep_depth-{{
_ $borehole->
bh_num_ }}"
step="any"
value="{{-(
float)
$deep->
$samples[ $j
]->sample_to_
 }}" id="
deep_depth-{{
_ $borehole->
bh_num_ }}-{{-
$k_ }}"
readonly></td
>
<td><input type="
number" name="
deep_f[]"
class="form-
control_
to_clear_
deep_f-{{_
$borehole->
bh_num_ }}"
step="any"
value="{{-(
float) $deep->
deep_f_ }}"
id="deep_f-{{
_ $borehole->
bh_num_ }}-{{-
$k_ }}"
readonly
required></td
>
<td><input type="
number" name="
deep_Asurface
[]" class="
form-control_
to_clear_
deep_Asurface_
{{_ $borehole
->bh_num_ }}"
step="any"
value="{{-(
float) $deep->
deep_Asurface_
 }}" id="
deep_Asurface_
{{_ $borehole
->bh_num_ }}-
{{_ $k_ }}"
readonly
required></td
>
<td><input type="
number" name="
deep_q[]"
class="form-
control_
deep_q-{{_
$borehole->
bh_num_ }}"
step="any"
value="{{-_
$deep->
deep_q- }}" id="
deep_q-{{_
$borehole->
bh_num_ }}-{{-
$k_ }}"
readonly
required></td
>
<td><input type="
number" name="
deep_At[]"
class="form-
control_
to_clear_
deep_At[]-{{_
$borehole->
bh_num_ }}"
step="any"
value="{{-(
float) $deep->
deep_At[]- }}"
id="
deep_At[]-{{_
$borehole->
bh_num_ }}-
{{_ $k_ }}"
readonly
required></td
>
<td><input type="
number" name="
deep_Qtotal
[]" class="
form-control_
to_clear_
deep_Qtotal_
{{_ $borehole
->bh_num_ }}"
step="any"
value="{{-(
float) $deep->
deep_Qtotal_
 }}" id="
deep_Qtotal_
{{_ $borehole
->bh_num_ }}-
{{_ $k_ }}"
readonly
required></td
>
<td><input type="
number" name="
deep_Q_kn[]"
class="form-
control_
to_clear_
deep_Q_kn-{{_
$borehole->
bh_num_ }}"
step="any"
value="{{-(
float) $deep->
deep_Q_kn- }}"
id="
deep_Q_kn-{{_
$borehole->
bh_num_ }}-{{-
$k_ }}"
readonly
required></td
>
<td><input type="
number" name="
deep_Asurface
[]" class="
form-control_

```

Listing 76: deeps/index.blade.php

```

"deep_Q_ton []
" class="form
-control_
to_clear_
deep_Q_ton_{{
_ $borehole->
bh_num_}}}"
step="any"
value="{{{(
float) $deep->
deep_Q_ton_}}}"
" id="
deep_Q_ton_{{
_ $borehole->
bh_num_}}}-{{(
$`k_}}}"
readonly
required></td
>
<td style="text-
align:center;
">
<?php ($deep->
deep_is_chosen
== 1) ?
$checked
= "
checked"
:
$checked
= "" ?>
<input type="
checkbox"
name="
deep_is_chosen
[]" value
=" {{(
$deep->
sample_id
_}})"
style="
text-
align:
center"
{{(
$checked
)}}>
</td>
</tr>
<?php $k++ ?>
@endif
<?php $j++ ?>
@endforeach
</tbody>
</table>
<br/>
{!! Form::hidden('bh_id []',
_ $borehole->id) !!}
{!! Form::hidden('
num_of_samples []', $k-1)
!!}
@endforeach
<div class="ln_solid"></div>
{!! Form::hidden('proj_id',
_ $project->id) !!}
{!! Form::hidden('
deep_is_chosen []', -1)
!!}
{!! link_to(URL::previous(), '
Back', array('class' => '
btn_btn-default')) !!}
{!! Form::button('Clear', ['
class' => 'btn_btn-danger',
'onclick' => '
clearForm()']) !!}
{!! Form::submit('Submit', ['
class' => 'btn_btn-
primary', 'style' => '
float:right', 'onclick'
=> 'calculate()']) !!}
{!! Form::button('Calculate',
['class' => 'btn_btn-
success', 'style' => '
float:right', 'onclick'
=> 'calculate()']) !!}
{!! Form::close() !!}
@endif
</div>
</div>
</div>
</div>
@endsection
@section('js')
@parent
@include('deeps/calculate')
@endsection

```

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
<div class="col-md-12"
>
<div class="
x_panel">
<div class="
x_title">
<h2>Deep
Foundation
<
small
>
Compute
soil
bearing
capacity
for
deep
foundation
</
small
></h2
>
<ul class=
"nav_
navbar
-
right
_
panel-toolbox
">
<li><a
class
=
"
collapse
-
link
"
><
i
class
=
"
dropdown
"
>
<a
href
=
"
#
"
class
=
"
dropdown
-
toggle
"
data
-
toggle
=
"
dropdown
"
role

```

```

=
"
button
"
aria
-
expanded
=
"
false
"
<<
i
class
=
"
fa
-
fa
-
wrench
"
></
i
></
a
>
<
ul
class
=
"
dropdown
-
menu
"
role
=
"
menu
"
>
<
li
<<
a
href
=
"
#
"
>
Settings
1</
a
>
</
li
>
<
li
<<
a
href
=
"
#
"
>
Settings
2</
a
>
</
li
>
</
ul
>
</li>
<div class
=
"
clearfix
"
>
</div>
"></
div>
<div class="
x_content
">
@if(sizeof
(
$deeps
) >
0)
@foreach
(
$boreholes
as
$borehole
)
<
center
<<
h2
>
Borehole
{{
$borehole
->
bh_num
}}
<
small
>(
ID
#
{{
$borehole
->
id
}})
</
small
><<
h2
><<
center
>
<
table
class
=
"
table
-
table
-
hover
-
table
-
bordered
"
>
<
thead
>
<
tr
>
<
th
<<
a
"
da
-
to
"
to
"
"
ti
=
"
"
">

```

```

Pile
Size
</a
></th
>
<
th
</a
data
toggle
="
"
tooltip
"
title
="
"
"
>
K
</a
></th
>
<
th
</a
data
toggle
="
"
tooltip
"
title
="
"
"
>&
gamma
:</a
></th
>
<
th
</a
data
toggle
="
"
tooltip
"
title
="
"
"
>
tan
&
delta
:</a
></th
>
<
th
</a
data
toggle
="
"
tooltip
"

```

```

toggle
=
"
tooltip
"

title
=
"
"
>
Concrete
Strength

(
kN
)
</
a
></
th
>

<
th
<
a

data
=
toggle
=
"
tooltip
"

title
=
"
"
"
>
Concrete
Strength

(
ton
)
</
a
></
th
>

</
tr
>

</
thead
>

<
tbody
>

<?
php
    $i
    =
    0
    ?>

    @while
    (
    $borehole
    ->
    id
    !=
    $deeps
    [
    $i
    ]->
    bh_id
    )
    <?
    php
        $i
        ++

```

```

)
$deeps
[
$i
]->
deep_rho
}}</
td
>
<
td
>{{
(
float
)
$deeps
[
$i
]->
deep_fy
}}</
td
>
<
td
>{{
(
float
)
$deeps
[
$i
]->
deep_fc
}}</
td
>
<
td
>{{
(
float
)
$deeps
[
$i
]->
deep_concrete_strength_kn
}}</
td
>
<
td
>{{
(
float
)
$deeps
[
$i
]->
deep_concrete_strength_ton
}}</
td
>
</
tr
>
</
tbody
>
</
table
>
<
table
class
=
"
table

```

```

"
tooltip
"
title
=
"
"
>
A
<sub
>
surface
</sub
sub
</a
a
</th
>
<
th
</a
data
-
toggle
=
"
tooltip
"
title
=
"
"
>
q
</a
a
</th
>
<
th
</a
data
-
toggle
=
"
tooltip
"
title
=
"
"
>
A
<sub
>
tip
</sub
</a
a
</th
>
<
th
</a
data
-
toggle
=
"
tooltip
"
title
=
"
"
>
Q
<sub

```

```

@if
(
  $deep
  ->
  bh_id
  ==
  $borehole
  ->
  id
  )
<?
  php
  (
    $deep
    ->
    deep.is.chosen
    ==
    1)
  ?
  $class
  =
  "
  chosen
  "
  :
  $class
  =
  "
  "
  ?>
<
  tr
  class
  =
  "{
  _
  $class
  _
  }"
  >
  <
  td
  >{{
    (
      float
    )
    $samples
    [
    $j
    ]->
    sample-to
  }}</
  td
  >
  <
  td
  >{{
    (
      float
    )
    $deep
    ->
    deep-f
  }}</
  td
  >
  <
  td
  >{{
    (
      float
    @endforeach
  }
  @endif
  <?
  php
  $j
  ++
  ?>
  @endforeach

```



```

        </tbody>
    </table>
    <br />
@endforeach
<div class="ln-solid">
{!!
    Form
    ::
    open
    (
    ,
    method
    ,
    =>
    ,
    DELETE
    ,
    ,
    route
    ,
    =>
    [
    ,
    deeps
    .
    destroy
    ,
    ,
    $project
    ->
    id
    ],
    ,
    onsubmit
    ,
    =>
    ,
    return
    -
    confirm
    ("
    -
    Delete
    -
    SBC
    -
    data
    -
    for
    -
    deep
    -
    foundation
    -
    permanently
    ?")
    ,
    ]
    !!}
{!!
    Form
    ::
    submit
    (
    ,
    Delete
    ,
    ,
    @endsection
}
[
,
class
,
=>
,
btn
-
btn
-
danger
,
])
!!}
<a href="
"
{{
-
route
(
,
deeps
.
edit
,
-
$project
->
id
)
}}
"
class
=
"
btn
-
btn
-
info
"
style
=
"
float
:
right
"
">
<<
i
class
=
"
"
fa
-
fa
-
edit
-
m
-
right
-
xs
"
">
</i
>
Edit
</a
>
{!!
    Form
    ::
    close
    ()
    !!}
}
@endif
</div>
</div>
</div>

```

Listing 77: drilledshafts/calculate.blade.php

```

@include('includes/clearform')

<!-- tooltip -->
<script>
$(document).ready(function(){
    $('[data-toggle="tooltip"]').tooltip()
});
</script>
<!-- /tooltip -->

<!-- calculate -->
<script type="text/javascript">
    var num_of_boreholes = <?php echo
        sizeof($boreholes) ?>;
    var num_of_samples;
    var fy, sb_diameter;
    var soil_type, num_of_blows,
        c_denominator, c;
    var nc, pile_size, qbase;
    var f_coeff, f;
    var df, qfriction;
    var fs, qtotal_kn, qtotal_ton;
    var fc, cs_kn, cs_ton;

    function calculate() {
        for(i = 1; i <= num_of_boreholes;
            i++) {
            // Compute tension capacity
            fy = $('#ds_fy_' + i).val();
            sb_diameter = $('#
                ds_steel_bar_diameter_' +
                i).val();

            tension_cap = (3.1416*fy*((
                sb_diameter*sb_diameter)
                /4))/1000;
            $('#ds_tension_cap_' + i).val(
                parseFloat(tension_cap.
                toFixed(4)));

            num_of_samples = $('#ds_depth_'
                + i).length;
            for(j = 1; j <= num_of_samples
                ; j++) {

                // Compute c
                soil_type = $('#ds_soil_'
                    + i + '_' + j + '_'
                    + option:selected').
                    text();
                num_of_blows = $('#
                    ds_num_of_blows_' + i
                    + '_' + j).val();

                switch(soil_type) {
                    case 'clay':
                        c_denominator =
                            4; break;
                    case 'silt':
                        c_denominator =
                            5; break;
                    default: c_denominator
                        = null; break;
                }

                c = (num_of_blows*48)/
                    c_denominator;
                $('#ds_c_' + i + '_' + j).
                    val(parseFloat(c.
                    toFixed(4)));

                // Compute Qbase
                nc = $('#ds_Nc_' + i).val
                    ();
                pile_size = $('#
                    ds_pile_size_' + i).
                    val();

                qbase = c*nc*(Math.PI*((
                    pile_size*pile_size)
                    /4));
                $('#ds_Qbase_' + i + '_' +
                    j).val(parseFloat(
                    qbase.toFixed(4)));

                // Compute f
                f_coeff = $('#ds_f_coeff_'
                    + i).val();

                f = c*f_coeff;
                $('#ds_f_' + i + '_' + j).
                    val(parseFloat(f.
                    toFixed(4)));
            }
        }
    }
</script>
<!-- /calculate -->

```

```

// Compute Qfriction
df = $('#ds_depth_' + i +
    '_' + j).val();

qfriction = Math.PI*f*
    pile_size*df;
$('#ds_Qfriction_' + i +
    '_' + j).val(
    parseFloat(qfriction.
    toFixed(4)));

// Compute Qtotal (kN)
fs = $('#ds_fs_' + i).val
    ();

qtotal_kn = (qbase+
    qfriction)/fs;
$('#ds_Qtotal_kn_' + i +
    '_' + j).val(
    parseFloat(qtotal_kn.
    toFixed(4)));

// Compute Qtotal (ton)
qtotal_ton = qtotal_kn
    *0.10219;
$('#ds_Qtotal_ton_' + i +
    '_' + j).val(
    parseFloat(qtotal_ton
    .toFixed(4)));

// Compute concrete
    strength (kN)
fc = $('#ds_fc_' + i).val
    ();
cs_kn = (0.85*0.7*(Math.PI
    *pile_size*pile_size
    *1000*1000/4)*fc+(
    Math.PI*sb_diameter*
    sb_diameter/4*fy))
    /1000;
$('#
    ds_concrete_strength_kn_'
    + i).val(parseFloat(
    cs_kn.toFixed(4)));

// Compute concrete
    strength (ton)
cs_ton = cs_kn*0.10219;
$('#
    ds_concrete_strength_ton_'
    + i).val(parseFloat(
    cs_ton.toFixed(4)));
}
}
</script>
<!-- /calculate -->

```

Listing 78: drilledshafts/create.blade.php

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
    <div class="col-md-12">
        <div class="x_panel">
            <div class="x_title">
                <h2>Create - Drilled Shaft
                    Foundation <small>Compute
                    soil bearing capacity
                    for drilled shaft
                    foundation </small></h2>
            </div>
            <ul class="nav navbar-right
                panel_toolbox">
                <li><a class="collapse-
                    link"><i class="fa fa
                    -chevron-up"></i></a>
            </li>
            <li class="dropdown">
                <a href="#" class="
                    dropdown-toggle"
                    data-toggle="
                    dropdown" role="
                    button" aria-
                    expanded="false"
                    ><i class="fa fa-
                    wrench"></i></a>
                <ul class="dropdown-
                    menu" role="menu"
                    >
                    <li><a href="#">
                        Settings 1</a>
                    </li>
                </ul>
            </li>

```

```

                <li><a href="#">
                    Settings 2</a>
                </li>
            </ul>
        </li>
    </ul>
</div>
<div class="clearfix"></div>
</div>
<div class="x_content">
    <div class="x_content">
        @if(sizeof($samples) > 0)
        <?php $soil = ["clay", "silt"]
        ?>
        {!! Form::open(['name' => '
            store_form', 'method' =>
            'POST', 'route' => '
            drilledshafts.store', '
            class' => 'form-
            horizontal_form-label-
            left']) !!}
        @foreach($boreholes as
            $borehole)
        {!! Form::hidden('bh_id[]',
            $borehole->id) !!}
        {!! Form::hidden('
            num_of_samples[]',
            $borehole->
            bh_num_of_samples) !!}
        <center>
            <h2>Borehole {{ $borehole
            ->bh_num }} <small>(
            ID#{{ $borehole->id
            }})</small></h2>
        </center>
        <table class="table table-
            striped table-hover">
            <thead>
                <tr>
                    <th><a data-toggle
                    =" tooltip"
                    title=" ">Pile
                    Size</a></th>
                    <th><a data-toggle
                    =" tooltip"
                    title=" ">N<
                    sub>c</sub></a></th>
                    <th><a data-toggle
                    =" tooltip"
                    title=" ">F
                    Coeff</a></th>
                    <th><a data-toggle
                    =" tooltip"
                    title=" "
                    Compressive_
                    strenght_of_
                    concrete">fc
                    </a></th>
                    <th><a data-toggle
                    =" tooltip"
                    title=" "Yield_
                    strenght_of_
                    reinforcing_
                    bar">fy</a></th>
                    <th><a data-toggle
                    =" tooltip"
                    title=" "Steel_
                    bar_diameter"
                    >SB Diameter
                    </a></th>
                    <th><a data-toggle
                    =" tooltip"
                    title=" "Factor
                    _of_
                    safety">
                    FS</a></th>
                    <th><a data-toggle
                    =" tooltip"
                    title=" "
                    Tension_
                    capacity_of_
                    steel">
                    Tension
                    Capacity</a
                    ></th>
                    <th><a data-toggle
                    =" tooltip"
                    title=" ">
                    Concrete
                    Strength (kN)
                    </a></th>
                    <th><a data-toggle
                    =" tooltip"
                    title=" ">
                    Concrete
                    Strength (ton
                    )</a></th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td><input type="
                    number" name=
                    " ds_pile_size
                    []" class="
                    form-control_
                    to_clear"
                    step="any"
                    value="{{_old
                    ('
                    ds_pile_size
                    ')_}}" id="
                    ds_pile_size_
                    {{_ $borehole
                    ->bh_num_}}
                    " required></td>
                    <td><input type="
                    number" name=
                    " ds_Nc []"
                    class="form-
                    control_
                    to_clear"
                    step="any"
                    value="{{_old
                    (' ds_Nc ')_}}
                    " id="
                    ds_Nc_{{
                    _ $borehole->
                    bh_num_}}
                    " required></td>
                    <td><input type="
                    number" name=
                    " ds_f_coeff []"
                    class="form-
                    control_
                    to_clear"
                    step="any"
                    value="{{_old
                    (' ds_f_coeff
                    ')_}}" id="
                    ds_f_coeff_{{
                    _ $borehole->
                    bh_num_}}
                    " required></td>
                    <td><input type="
                    number" name=
                    " ds_fc []"
                    class="form-
                    control_
                    to_clear"
                    step="any"
                    value="{{_old
                    (' ds_fc ')_}}
                    " id=" ds_fc_{{
                    _ $borehole->
                    bh_num_}}
                    " required></td>
                    <td><input type="
                    number" name=
                    " ds_fy []"
                    class="form-
                    control_
                    to_clear"
                    step="any"
                    value="{{_old
                    (' ds_fy ')_}}
                    " id=" ds_fy_{{
                    _ $borehole->
                    bh_num_}}
                    " required></td>
                    <td><input type="
                    number" name=
                    " ds_steel_bar_diameter
                    []" class="
                    form-control_
                    to_clear"
                    step="any"
                    value="{{_old
                    ('
                    ds_steel_bar_diameter
                    ')_}}" id="
                    ds_steel_bar_diameter_
                    {{_ $borehole
                    ->bh_num_}}
                    " required></td>
                    <td><input type="
                    number" name=
                    " ds_fs []"
                    class="form-
                    control_

```

```

        to_clear"
        step="any"
        value="{{_old
(' ds_fs '_)}}
        id=" ds_fs_{{
_ $borehole->
bh_num_ }}"
        required></td
>
<td><input type="
number" name="
"
ds_tension_cap
[]" class="
form-control_
to_clear_
step="any"
value="{{_old
('
ds_tension_cap
')_ }}" id="
ds_tension_cap_
{{_ $borehole
->bh_num_ }}"
readonly></td
>
<td><input type="
number" name="
"
ds_concrete_strength_kn
[]" class="
form-control_
to_clear_
ds_concrete_strength_kn_
{{_ $borehole
->bh_num_ }}"
step="any" id
="
ds_concrete_strength_kn_
{{_ $borehole
->bh_num_ }}"
readonly
required></td
>
<td><input type="
number" name="
"
ds_concrete_strength_ton
[]" class="
form-control_
to_clear_
ds_concrete_strength_ton_
{{_ $borehole
->bh_num_ }}"
step="any" id
="
ds_concrete_strength_ton_
{{_ $borehole
->bh_num_ }}"
readonly
required></td
>
</tr>
</tbody>
</table>
<table class="table_table-
striped_table-hover">
<thead>
<tr>
<th><a data-toggle
=" tooltip"
title="Depth_
of_ footing_
below_ lowest_
adjacent_ soil_
surface">D<
sub>f</sub></a></th>
<th><a data-toggle
=" tooltip"
title="Number
_of_ blows">
SPT N-value</a></th>
<th>Soil Type</th>
<th><a data-toggle
=" tooltip"
title="">c</a>
</th>
<th><a data-toggle
=" tooltip"
title="">Q<
sub>base</sub>
</a></th>
<th><a data-toggle
=" tooltip"
title="">f</a>
</th>
<th><a data-toggle
=" tooltip"
title="">
=" tooltip"
title="">Q<
sub>friction
</sub></a></th>
<th><a data-toggle
=" tooltip"
title="">Q<
sub>total</a>
</th>
<th><a data-toggle
=" tooltip"
title="">Q<
sub>total</a>
</th>
<th><a data-toggle
=" tooltip"
title="">
Choose</a></th>
</tr>
</thead>
<tbody>
@foreach( $samples as
$sample)
@if( $sample->bh_id ==
$borehole->id)
{!! Form:: hidden('
sample_id []',
$sample->id) !!}
<tr>
<td><input type="
number" name="
ds_depth []"
class="form-
control_
ds_depth_{{_
$borehole->
bh_num_ }}"
step="any"
value="{{_
$sample->
sample_to_ }}"
id="
ds_depth_{{_
$borehole->
bh_num_ }}-{{_
$sample->
sample_num_ }}"
readonly></td>
<td><input type="
number" name="
ds_num_of_blows
[]" class="
form-control_
ds_num_of_blows_
{{_ $borehole
->bh_num_ }}"
step="any"
value="{{_
$sample->
sample_num_of_blows_
 }}" id="
ds_num_of_blows_
{{_ $borehole
->bh_num_ }}-
{{_ $sample->
sample_num_ }}"
readonly></td>
<td>
<?php
ds_soil_select_id
= '
ds_soil_'
.
$borehole
->bh_num
.
'_'
.
$sample->
sample_num
?>
{!! Form::
select('
ds_soil []
', $soil,
old('
ds_soil []
'), array
('class'
=> 'btn-
btn-
default_
ds_soil_
{{_

```

```

        $borehole
        ->bh.num_
    }}, '
    placeholder
    , => ,
    Soil_type
    ... , '
    required '
    => 'true
    , 'id '
    =>
    $ds_soil_select_id
    )) !!}
</td>
<td><input type="
number" name=
" ds_c [] "
class="form-
control_
to_clear_
ds_c-{{_
$borehole->
bh.num_}} "
step="any" id
=" ds_c-{{_
$borehole->
bh.num_}}-{{_
$sample->
sample.num_}}
" readonly
required></td
>
<td><input type="
number" name=
" ds_Qbase [] "
class="form-
control_
to_clear_
ds_Qbase-{{_
$borehole->
bh.num_}} "
step="any" id
=" ds_Qbase-{{
_ $borehole->
bh.num_}}-{{_
$sample->
sample.num_}}
" readonly
required></td
>
<td><input type="
number" name=
" ds_f [] "
class="form-
control_
to_clear_
ds_f-{{_
$borehole->
bh.num_}} "
step="any" id
=" ds_f-{{_
$borehole->
bh.num_}}-{{_
$sample->
sample.num_}}
" readonly
required></td
>
<td><input type="
number" name=
" ds_Qfriction
[] " class="
form-control_
to_clear_
ds_Qfriction-
{{_ $borehole
->bh.num_}} "
step="any" id
="
ds_Qfriction-
{{_ $borehole
->bh.num_}}-
{{_ $sample->
sample.num_}}
" readonly
required></td
>
<td><input type="
number" name=
" ds_Qtotal.kn
[] " class="
form-control_
to_clear_
ds_Qtotal.kn-
{{_ $borehole
->bh.num_}} "
step="any" id
="
ds_Qtotal.kn_

```

```

        {{_ $borehole
        ->bh.num_}}-
        {{_ $sample->
        sample.num_}}
        " readonly
        required></td
        >
        <td><input type="
        number" name=
        " ds_Qtotal.ton
        [] " class="
        form-control_
        to_clear_
        ds_Qtotal.ton_
        {{_ $borehole
        ->bh.num_}} "
        step="any" id
        =
        " ds_Qtotal.ton_
        {{_ $borehole
        ->bh.num_}}-
        {{_ $sample->
        sample.num_}}
        " readonly
        required></td
        >
        <td style="text-
        align:center;
        "><input type
        ="checkbox"
        name="
        ds_is_chosen
        [] " value="{{
        _ $sample->id_
        }}"></td>
    </tr>
    @endif
    @endforeach
</tbody>
</table>
<br/>
@endforeach
<div class="ln_solid"></div>
{!! Form::hidden('proj.id',
    $project->id) !!}
{!! Form::hidden('ds_is_chosen
[]', -1) !!}
{!! link_to(URL::previous(), '
Back', array('class' => '
btn_btn-default')) !!}
{!! Form::button('Clear', [
    'class' => 'btn_btn-danger
', 'onclick' => '
clearForm()' ]) !!}
{!! Form::submit('Submit', [
    'class' => 'btn_btn-
primary', 'style' => '
float:right', 'onclick'
=> 'calculate()' ]) !!}
{!! Form::button('Calculate',
    ['class' => 'btn_btn-
success', 'style' => '
float:right', 'onclick'
=> 'calculate()' ]) !!}
{!! Form::close() !!}
@endif
</div>
</div>
</div>
</div>
@endsection
@section('js')
@parent
@include('drilledshafts/calculate')
@endsection

```

```

@extends('layouts/master')
@section('title', $project->proj.name)
@section('content')
<div class="row">
    <div class="col-md-12">
        <div class="x_panel">
            <div class="x_title">
                <h2>Edit - Drilled Shaft
                Foundation <small>Compute
                soil bearing capacity
                for drilled shaft
                foundation</small></h2>
            <ul class="nav navbar-right_
            panel_toolbox">
                <li><a class="collapse-
                link"><i class="fa_fa_
                -chevron-up"></i></a>
            </li>

```

Listing 79: drilledshafts/edit.blade.php

```

<li class="dropdown">
  <a href="#" class="
    dropdown-toggle"
    data-toggle="
    dropdown" role="
    button" aria-
    expanded="false"
    ><i class="fa_fa-
    wrench"></i></a>
  <ul class="dropdown-
    menu" role="menu"
    >
    <li><a href="#">
      Settings 1</a>
    </li>
    <li><a href="#">
      Settings 2</a>
    </li>
  </ul>
</li>
</div>
<div class="clearfix"></div>
<div class="x_content">
  @if(sizeof($drilled shafts) >
    0)
  <?php $soil = ["clay", "silt"]
  >
  {!! Form::open(['name' => '
    update_form', 'method' =>
    'PUT', 'route' => ['
    drilled shafts.update',
    $project->id], 'class' =>
    'form-horizontal form-
    label-left', 'id' => '
    update_form']) !!}
  @foreach($boreholes as
    $borehole)
  <center>
    <h2>Borehole {{ $borehole
      ->bh_num }} <small>(
      ID#{{ $borehole->id
      }})</small></h2>
  </center>
  <table class="table table-
    striped table-hover">
    <thead>
      <tr>
        <th><a data-toggle
          ="tooltip"
          title=">Pile
          Size</a></th>
        <th><a data-toggle
          ="tooltip"
          title=">N<
          sub>c</sub></a></th>
        <th><a data-toggle
          ="tooltip"
          title=">F
          Coeff</a></th>
        <th><a data-toggle
          ="tooltip"
          title="
          Compressive_
          strenght_of_
          concrete">fc
          </a></th>
        <th><a data-toggle
          ="tooltip"
          title="Yield_
          strenght_of_
          reinforcing_
          bar">fy</a></
          th>
        <th><a data-toggle
          ="tooltip"
          title="Steel_
          bar_diameter"
          >SB Diameter
          </a></th>
        <th><a data-toggle
          ="tooltip"
          title="Factor
          _of_safety">
          FS</a></th>
        <th><a data-toggle
          ="tooltip"
          title="
          Tension_
          capacity_of_
          steel">
          Tension
          Capacity</a
          ></th>

```



```

        [ $i ]->
        ds_concrete_strength_ton
        -}}" id="
        ds_concrete_strength_ton_
        {{_ $borehole
        ->bh_num_}}"
        readonly
        required></td
        >
    </tr>
</tbody>
</table>
<table class="table table-
striped table-hover">
<thead>
<tr>
<th><a data-toggle
=" tooltip"
title="Depth_
of_footing_
below_lowest_
adjacent_soil_
_surface">D<
sub>f</sub></a></th>
<th><a data-toggle
=" tooltip"
title="Number
_of_blows">
SPT N-value</
a></th>
<th>Soil Type</th>
<th><a data-toggle
=" tooltip"
title="">c</a
></th>
<th><a data-toggle
=" tooltip"
title="">Q<
sub>base</sub
></a></th>
<th><a data-toggle
=" tooltip"
title="">f</a
></th>
<th><a data-toggle
=" tooltip"
title="">Q<
sub>friction
</sub></a></
th>
<th><a data-toggle
=" tooltip"
title="">Q<
sub>total</
sub> (kN)</a
></th>
<th><a data-toggle
=" tooltip"
title="">Q<
sub>total</
sub> (ton)</a
></th>
</tr>
</thead>
<tbody>
<?php $j = 0; $k = 1;
?>
@foreach(
$drilled_shafts
as $drilled_shaft
)
@if($drilled_shaft->
bh_id ==
$borehole->id)
{!! Form::hidden('
sample_id[]',
$drilled_shaft->
sample_id) !!}
{!! Form::hidden('
drilled_id[]',
$drilled_shaft->
id) !!}
<tr>
<td><input type="
number" name="
ds_depth[]"
class="form-
control_
ds_depth-{{_
$borehole->
bh_num_}}"
step="any"
value="{{_(
float)
$drilled_shaft
->ds_c_}}" id
="ds_c-{{_
$borehole->
bh_num_}}-{{_
$k_}}"
readonly
required></td
>
        $borehole->
        bh_num_}}-{{_
        $k_}}"
        readonly></td
        >
<td><input type="
number" name="
ds_c[]"
class="form-
control_
to_clear_
ds_c-{{_
$borehole->
bh_num_}}"
step="any"
value="{{_(
float)
$drilled_shaft
->ds_c_}}" id
="ds_c-{{_
$borehole->
bh_num_}}-{{_
$k_}}"
readonly
required></td
>
<?php
$ds_soil_select_id
=
ds_soil_
.
$borehole
->
bh_num
.
'-'
.$k;
$ds_soil =
Helper
::
getValue
(old(
ds_soil
),
$drilled_shaft
->
ds_soil
);
?>
{!! Form::
select('
ds_soil[]
', $soil,
$ds_soil
, array('
class' =>
'btn btn-
default_
ds_soil_
{{_
$borehole
->bh_num_
}}',
placeholder
=>
Soil_type
...',
required
=> 'true
', 'id'
=>
$ds_soil_select_id
)) !!}
</td>
<td><input type="
number" name="
ds_c[]"
class="form-
control_
to_clear_
ds_c-{{_
$borehole->
bh_num_}}"
step="any"
value="{{_(
float)
$drilled_shaft
->ds_c_}}" id
="ds_c-{{_
$borehole->
bh_num_}}-{{_
$k_}}"
readonly
required></td
>

```



```

<td><input type="
  number" name=
    " ds_Qbase [ ] "
  class="form-
  control_
  to_clear_
  ds_Qbase-{{_
  $borehole->
  bh_num_}}}"
  step="any"
  value="{{_ (
  float)
  $drilled_shaft
  ->ds_Qbase_}}"
  id="
  ds_Qbase-{{_
  $borehole->
  bh_num_}}-{{_
  $k_}}}"
  readonly
  required></td
>
<td><input type="
  number" name=
    " ds_f [ ] "
  class="form-
  control_
  to_clear_
  ds_f-{{_
  $borehole->
  bh_num_}}}"
  step="any"
  value="{{_ (
  float)
  $drilled_shaft
  ->ds_f_}}" id
  =" ds_f-{{_
  $borehole->
  bh_num_}}-{{_
  $k_}}}"
  readonly
  required></td
>
<td><input type="
  number" name=
    " ds_Qfriction
    [ ] " class="
    form-control_
    to_clear_
    ds_Qfriction-
    {{_ $borehole
    ->bh_num_}}}"
    step="any"
    value="{{_ (
    float)
    $drilled_shaft
    ->
    ds_Qfriction_
    }}" id="
    ds_Qfriction-
    {{_ $borehole
    ->bh_num_}}-
    {{_ $k_}}}"
    readonly
    required></td
    >
<td><input type="
  number" name=
    " ds_Qtotal_kn
    [ ] " class="
    form-control_
    to_clear_
    ds_Qtotal_kn-
    {{_ $borehole
    ->bh_num_}}}"
    step="any"
    value="{{_ (
    float)
    $drilled_shaft
    ->
    ds_Qtotal_kn_
    }}" id="
    ds_Qtotal_kn-
    {{_ $borehole
    ->bh_num_}}-
    {{_ $k_}}}"
    readonly
    required></td
    >
<td><input type="
  number" name=
    "
    ds_Qtotal_ton
    [ ] " class="
    form-control_
    to_clear_
    ds_Qtotal_ton-
    {{_ $borehole
    ->bh_num_}}}"
    step="any"
    value="{{_ (
    float)
    $drilled_shaft
    ->
    ds_Qtotal_ton_
    }}" id="
    ds_Qtotal_ton-
    {{_ $borehole
    ->bh_num_}}-
    {{_ $k_}}}"
    readonly
    required></td
    >
<td style="text-
  align:center;
  ">
  <?php (
    $drilled_shaft
    ->
    ds_is_chosen
    = 1) ?
    $checked
    = "
    checked"
    :
    $checked
    = "" ?>
    <input type="
    checkbox"
    name="
    ds_is_chosen
    [ ] " value
    =" {{_
    $drilled_shaft
    ->
    sample_id
    _}}}"
    style="
    text-
    align:
    center"
    {{
    $checked
    }}>
  </td>
</tr>
<?php $k++ ?>
@endif
<?php $j++ ?>
@endforeach
</tbody>
</table>
<br/>
{!! Form::hidden('bh_id [ ] ',
  $borehole->id) !!}
{!! Form::hidden('
  num_of_samples [ ] ', $k-1)
  !!}
@endforeach
<div class="ln_solid"></div>
{!! Form::hidden('proj_id ',
  $project->id) !!}
{!! Form::hidden('ds_is_chosen
  [ ] ', -1) !!}
{!! link_to(URL::previous(), '
  Back', array('class' => '
  btn_btn-default')) !!}
{!! Form::button('Clear', ['
  class' => 'btn_btn-danger',
  'onclick' => '
  clearForm()']) !!}
{!! Form::submit('Submit', ['
  class' => 'btn_btn-
  primary', 'style' => '
  float:right', 'onclick'
  => 'calculate()']) !!}
{!! Form::button('Calculate',
  ['class' => 'btn_btn-
  success', 'style' => '
  float:right', 'onclick'
  => 'calculate()']) !!}
{!! Form::close() !!}
@endif
</div>
</div>
</div>
</div>
<div>
  @endsection
  @section('js')
  @parent
  @include('drilledshafts/calculate')
  @endsection
</div>
<td><input type="
  number" name=
    "
    ds_Qtotal_ton
    [ ] " class="
    form-control_
    to_clear_
    ds_Qtotal_ton-
    {{_ $borehole
    ->bh_num_}}}"
    step="any"
    value="{{_ (
    float)
    $drilled_shaft
    ->
    ds_Qtotal_ton_
    }}" id="
    ds_Qtotal_ton-
    {{_ $borehole
    ->bh_num_}}-
    {{_ $k_}}}"
    readonly
    required></td
    >
<td style="text-
  align:center;
  ">
  <?php (
    $drilled_shaft
    ->
    ds_is_chosen
    = 1) ?
    $checked
    = "
    checked"
    :
    $checked
    = "" ?>
    <input type="
    checkbox"
    name="
    ds_is_chosen
    [ ] " value
    =" {{_
    $drilled_shaft
    ->
    sample_id
    _}}}"
    style="
    text-
    align:
    center"
    {{
    $checked
    }}>
  </td>
</tr>
<?php $k++ ?>
@endif
<?php $j++ ?>
@endforeach
</tbody>
</table>
<br/>
{!! Form::hidden('bh_id [ ] ',
  $borehole->id) !!}
{!! Form::hidden('
  num_of_samples [ ] ', $k-1)
  !!}
@endforeach
<div class="ln_solid"></div>
{!! Form::hidden('proj_id ',
  $project->id) !!}
{!! Form::hidden('ds_is_chosen
  [ ] ', -1) !!}
{!! link_to(URL::previous(), '
  Back', array('class' => '
  btn_btn-default')) !!}
{!! Form::button('Clear', ['
  class' => 'btn_btn-danger',
  'onclick' => '
  clearForm()']) !!}
{!! Form::submit('Submit', ['
  class' => 'btn_btn-
  primary', 'style' => '
  float:right', 'onclick'
  => 'calculate()']) !!}
{!! Form::button('Calculate',
  ['class' => 'btn_btn-
  success', 'style' => '
  float:right', 'onclick'
  => 'calculate()']) !!}
{!! Form::close() !!}
@endif
</div>
</div>
</div>
</div>
<div>
  @endsection
  @section('js')
  @parent
  @include('drilledshafts/calculate')
  @endsection
</div>

```

Listing 80: drilledshafts/index.blade.php

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
  <div class="col-md-12">
    <div class="x_panel">
      <div class="x_title">
        <h2>Drilled Shaft Foundation <
          <small>Compute soil
            bearing capacity for
            drilled shaft foundation
          </small></h2>
        <ul class="nav navbar-right
          panel_toolbox">
          <li><a class="collapse-
            link"><i class="fa fa
            -chevron-up"></i></a>
          </li>
          <li class="dropdown">
            <a href="#" class="
              dropdown-toggle"
              data-toggle="
              dropdown" role="
              button" aria-
              expanded="false"
              ><i class="fa fa-
              wrench"></i></a>
            <ul class="dropdown-
              menu" role="menu"
              >
              <li><a href="#">
                Settings 1</a>
              </li>
              <li><a href="#">
                Settings 2</a>
              </li>
            </ul>
          </li>
        </ul>
      </div>
      <div class="clearfix"></div>
      </div>
      <div class="x_content">
        @if(sizeof($drilled_shafts) >
          0)
          @foreach($boreholes as
            $borehole)
          <center>
            <h2>Borehole {{ $borehole
              ->bh_num }} <small>(<
              ID#{{ $borehole->id
              }})</small></h2>
          </center>
          <table class="table table-
            striped table-hover">
            <thead>
              <tr>
                <th><a data-toggle
                  =" tooltip"
                  title=">Pile
                  Size</a></th>
                <th><a data-toggle
                  =" tooltip"
                  title=">N<
                  sub>c</sub></a></th>
                <th><a data-toggle
                  =" tooltip"
                  title=">F<
                  Coeff</a></th>
                <th><a data-toggle
                  =" tooltip"
                  title=">Compressive_
                  strenght_of_
                  concrete">fc
                  </a></th>
                <th><a data-toggle
                  =" tooltip"
                  title=">Yield_
                  strenght_of_
                  reinforcing_
                  bar">fy</a></th>
                <th><a data-toggle
                  =" tooltip"
                  title=">Steel_
                  bar_diameter"
                  >SB Diameter
                  </a></th>
                <th><a data-toggle
                  =" tooltip"
                  title=">Factor
                  of_safety">
                  FS</a></th>
                <th><a data-toggle
                  =" tooltip"
                  title=">Tension_
                  capacity_of_
                  steel">
                  Tension
                  Capacity</a>
                <th><a data-toggle
                  =" tooltip"
                  title=">Concrete
                  Strength (kN)
                </a></th>
                <th><a data-toggle
                  =" tooltip"
                  title=">Concrete
                  Strength (ton)
                </a></th>
              </tr>
            </thead>
            <tbody>
              <?php $i = 0 ?>
              @while($borehole->id
                !=
                $drilled_shafts[
                $i]->bh_id)
              <?php $i++ ?>
              @endwhile
              <tr>
                <td>{{ (float)
                  $drilled_shafts
                  [$i]->
                  ds_pile_size
                }}</td>
                <td>{{ (float)
                  $drilled_shafts
                  [$i]->ds_Nc
                }}</td>
                <td>{{ (float)
                  $drilled_shafts
                  [$i]->ds_f_coeff
                }}</td>
                <td>{{ (float)
                  $drilled_shafts
                  [$i]->ds_fc
                }}</td>
                <td>{{ (float)
                  $drilled_shafts
                  [$i]->ds_fy
                }}</td>
                <td>{{ (float)
                  $drilled_shafts
                  [$i]->
                  ds_steel_bar_diameter
                }}</td>
                <td>{{ (float)
                  $drilled_shafts
                  [$i]->ds_fs
                }}</td>
                <td>{{ (float)
                  $drilled_shafts
                  [$i]->
                  ds_tension_cap
                }}</td>
                <td>{{ (float)
                  $drilled_shafts
                  [$i]->
                  ds_concrete_strength_kn
                }}</td>
                <td>{{ (float)
                  $drilled_shafts
                  [$i]->
                  ds_concrete_strength_ton
                }}</td>
              </tr>
            </tbody>
          </table>
          <table class="table table-
            striped table-hover">
            <thead>
              <tr>
                <th><a data-toggle
                  =" tooltip"
                  title=">Depth_
                  of_footing_
                  below_lowest_
                  adjacent_soil_
                  surface">D<
                  sub>i</sub></a></th>
                <th><a data-toggle
                  =" tooltip"
                  title=">Number
                  of_blows">
                  SPT N-value</a>
                </th>
            </thead>

```



```

{!! Form::email('to', old('to'), ['class' => 'form-control', 'required' => 'true']) !!}
:11
px
"
>
Small
</
p
>
</div>
<div class="form-group">
{!! Form::label('subject', 'Subject_*') !!}
{!! Form::text('subject', old('subject'), ['class' => 'form-control', 'required' => 'true']) !!}
</li>
</ul>
</div>
<div class="form-group">
{!! Form::label('', 'Message') !!}
<div id="alerts"></div>
<div class="btn-toolbar editor" data-role="editor-toolbar" data-target="#editor">
<div class="btn-group">
<a class="btn dropdown-toggle" data-toggle="dropdown" title="Font Size"><i class="icon-text-height"></i>&nbsp;<b class="caret"></b></a>
<ul class="dropdown-menu">
<li>
<a data-edit="" font-size="5">
<p style="font-size: 17px; text-align: center; font-weight: bold; margin: 0; padding: 0;">Huge
</p>
</li>
</a>
</li>
<li>
<a data-edit="" font-size="3">
<p style="font-size: 14px; text-align: center; font-weight: normal; margin: 0; padding: 0;">Normal
</p>
</li>
</a>
</li>
<li>
<a data-edit="" font-size="1">
<p style="font-size: 11px; text-align: center; font-weight: normal; margin: 0; padding: 0;">Small
</p>
</li>
</a>
</li>
</ul>
</div>
</div>
<div class="btn-group">
<a class="btn" data-edit="bold" title="Bold_(Ctrl/Cmd+B)"><i class="icon-bold"></i></a>
<a class="btn" data-edit="italic" title="Italic_(Ctrl/Cmd+I)"><i class="icon-italic"></i></a>
<a class="btn" data-edit="strikethrough" title="Strikethrough"><i class="icon-strikethrough"></i></a>
<a class="btn" data-edit="underline" title="Underline_(Ctrl/Cmd+U)"><i class="icon-underline"></i></a>
</div>
<div class="btn-group">
<a class="btn" data-edit="insertunorderedlist" title="Bullet_list"><i class="icon-list-ul"></i></a>
<a class="btn" data-edit="insertorderedlist" title="Number_list"><i class="icon-list-ol"></i></a>
<a class="btn" data-edit="outdent" title="Reduce_indent_(Shift+Tab)"><i class="icon-indent-left"></i></a>
<a class="btn" data-edit="indent" title="Indent_(Tab)"><i class="icon-indent-right"></i></a>
</div>
<div class="btn-group">
<a class="btn" data-edit="justifyleft" title="Align_Left_(Ctrl/Cmd+L)"><i class="icon-align-left"></i></a>
<a class="btn" data-edit="

```

```

        justifycenter
        " title="
        Center_(Ctrl/
        Cmd+E)"><i
        class="icon-
        align-center"
        ></i></a>
        <a class="btn"
        data-edit="
        justifyright"
        title="Align
        _Right_(Ctrl/
        Cmd+R)"><i
        class="icon-
        align-right"
        ></i></a>
        <a class="btn"
        data-edit="
        justifyfull"
        title="
        Justify_(Ctrl
        /Cmd+J)"><i
        class="icon-
        align-justify"
        ></i></a>
    </div>
    <div class="btn-group"
    >
        <a class="btn"
        title="Insert
        _picture_(or_
        just_drag_&_
        drop)" id="
        pictureBtn"><
        i class="icon-
        -picture"></i>
        </a>
        <input type="file"
        data-role="
        magic-overlay"
        data-target
        ="#pictureBtn"
        data-edit="
        insertImage"
        />
    </div>
</div>
<textarea name="message"
id="message" style="
display:none;"></
textarea>
<div id="editor"
contenteditable="true"
"></div>
</div>
<?php $file_path = "reports/
Report->$project->id.pdf"
?>
@if($file_exists($file_path))
<?php $file_path = URL::asset(
"reports/Report->$project
->id.pdf") ?>
<i class="fa_fa-paperclip"></i>
> {!! link_to_asset(
$file_path, 'Geotechnical
_Investigation_Report_-_'
. $project->proj_name)
!!}
{!! Form::hidden('
report_exists', 1) !!}
{!! Form::hidden('file_path',
$file_path) !!}
@else
<i class="fa_fa-times"></i>
Report has not been
generated.
{!! Form::hidden('
report_exists', 0) !!}
{!! Form::hidden('file_path',
0) !!}
@endif
<div class="ln_solid"></div>
{!! Form::hidden('proj-id',
$project->id) !!}
{!! link_to(URL::previous(), '
Back', array('class' => '
btn_btn-default')) !!}
{!! Form::submit('Submit', ['
class' => 'btn_btn-
primary', 'style' => '
float:right']) !!}
{!! Form::close() !!}
</div>
</div>
</div>
</div>
@endsection
@section('js')
@parent
        <!-- tags -->
        <script src="{{_URL::asset('js/tags/jquery.
        tagsinput.min.js')}}"></script>
        <!-- switchery -->
        <script src="{{_URL::asset('js/switchery/
        switchery.min.js')}}"></script>
        <!-- daterangepicker -->
        <script src="{{_URL::asset('js/moment.min2.js
        ')}" type="text/javascript"></script>
        <script src="{{_URL::asset('js/datepicker/
        daterangepicker.js')}}" type="text/
        javascript"></script>
        <!-- richtext editor -->
        <script src="{{_URL::asset('js/editor/
        bootstrap-wysiwyg.js')}}"></script>
        <script src="{{_URL::asset('js/editor/external
        /jquery.hotkeys.js')}}"></script>
        <script src="{{_URL::asset('js/editor/external
        /google-code-prettify/prettify.js')}}"
        ></script>
        <!-- select2 -->
        <script src="{{_URL::asset('js/select/select2.
        full.js')}}"></script>
        <!-- form validation -->
        <script src="{{_URL::asset('js/parsley/parsley
        .min.js')}}" type="text/javascript" ></
        script>
        <!-- textarea resize -->
        <script src="{{_URL::asset('js/textarea/
        autosize.min.js')}}"></script>
        <!-- Autocomplete -->
        <script src="{{_URL::asset('js/autocomplete/
        countries.js')}}" type="text/javascript"
        ></script>
        <script src="{{_URL::asset('js/autocomplete/
        jquery.autocomplete.js')}}"></script>
        <!-- form validation -->
        <script type="text/javascript">
        $(document).ready(function () {
            $.listen('parsley:field:validate',
            function () {
                validateFront();
            });
            $('#demo-form_1.btn').on('click',
            function () {
                $('#demo-form').parsley().validate
                ();
                validateFront();
            });
            validateFront = function () {
                if (true === $('#demo-form').
                parsley().isValid()) {
                    $('.bs-callout-info').
                    removeClass('hidden');
                    $('.bs-callout-warning').
                    addClass('hidden');
                } else {
                    $('.bs-callout-info').addClass
                    ('hidden');
                    $('.bs-callout-warning').
                    removeClass('hidden');
                }
            };
        });
        $(document).ready(function () {
            $.listen('parsley:field:validate',
            function () {
                validateFront();
            });
            $('#demo-form2_1.btn').on('click',
            function () {
                $('#demo-form2').parsley().
                validate();
                validateFront();
            });
            validateFront = function () {
                if (true === $('#demo-form2').
                parsley().isValid()) {
                    $('.bs-callout-info').
                    removeClass('hidden');
                    $('.bs-callout-warning').
                    addClass('hidden');
                } else {
                    $('.bs-callout-info').addClass
                    ('hidden');
                    $('.bs-callout-warning').
                    removeClass('hidden');
                }
            };
        });
        try {
            hljs.initHighlightingOnLoad();
        } catch (err) {}
    </script>
    <!-- /form validation -->
    <!-- editor -->
    <script>
    $(document).ready(function () {

```

```

$('.xexc').click(function () {
    $('#section_text').val($('#editor')
        .html());
});
});

$(function () {
    function initToolbarBootstrapBindings
    () {
        var fonts = ['Arial', 'Arial_Black',
            'Calibri', 'Cambria',
            'Comic_Sans_MS', 'Courier',
            'Courier_New', 'Helvetica', 'Impact',
            'Lucida_Grande', 'Lucida_Sans',
            'Sans', 'Serif', 'Tahoma', 'Times',
            'Times_New_Roman', 'Verdana'];
        fontTarget = $('[title=Font]')
            .siblings('.dropdown-menu');
        $.each(fonts, function (idx,
            fontName) {
                fontTarget.append($('- <a_
                    data-edit="fontName_' +
                    fontName + '"_style="font_
                    -family:\'' + fontName +
                    '\''>' + fontName + '</a_
                    ></li>'));
            });
        $('[title]').tooltip({
            container: 'body'
        });
        $('[data-role=magic-overlay]').click(
            function () {
                return false;
            })
            .change(function () {
                $(this).parent('.dropdown-
                    menu').siblings('.
                    dropdown-toggle').
                    dropdown('toggle');
            })
            .keydown('esc', function () {
                this.value = '';
                $(this).change();
            });

        $('[data-role=magic-overlay]').
            each(function () {
                var overlay = $(this);
                target = $(overlay.data('
                    target'));
                overlay.css('opacity', 0).css(
                    'position', 'absolute').
                    offset(target.offset()).
                    width(target.outerWidth()
                    ).height(target.
                    outerHeight());
            });
        if ("onwebkitspeechchange" in
            document.createElement("input
            ")) {
            var editorOffset = $('#editor')
                .offset();
            $('#voiceBtn').css('position',
                'absolute').offset({
                    top: editorOffset.top,
                    left: editorOffset.left +
                    $('#editor').
                    innerWidth() - 35
                });
        } else {
            $('#voiceBtn').hide();
        }
    };

    function showAlert(reason, detail)
    () {
        var msg = '';
        if (reason === 'unsupported-file-
            type') {
            msg = "Unsupported_format_" +
                detail;
        } else {
            console.log("error_uploading_
                file", reason, detail);
        }
        $('[data-dismiss="alert"]><button_
            type="button" class="close" _
            data-dismiss="alert">&times
            </button>' +
            '<strong>File_upload_error</
            strong>' + msg + '</div_
            >').prependTo('#alerts');
    };

    initToolbarBootstrapBindings();
    $('#editor').wysiwyg({
        fileUploadError: showAlert
    });
}

```

```

        window.prettyPrint && prettyPrint();
    });
</script>
<!-- /editor -->
@endsection

```

Listing 82: emails/send.blade.php

```
<p>{!! $body !!}</p>
```

Listing 83: errors/403.blade.php

```

@extends('layouts/errormaster')
@section('error_name', '403')
@section('error_description1', 'Forbidden')
@section('error_description2', 'Sorry, you are
    not permitted to perform the requested_
    operation.')

```

Listing 84: errors/404.blade.php

```

@extends('layouts/errormaster')
@section('error_name', '404')
@section('error_description1', 'Not_Found')
@section('error_description2', 'Sorry, the_
    page_you_are_looking_for_is_not_found.')

```

Listing 85: errors/503.blade.php

```

@extends('layouts/errormaster')
@section('error_name', '404')
@section('error_description1', 'Service_
    Unavailable')
@section('error_description2', 'Be_right_back.
    ')

```

Listing 86: includes/clear- form.blade.php

```

<!-- clear -->
<script type="text/javascript">
    function clearForm() {
        $('[to_clear]').val('');
        $('[input:checkbox:checked]').prop('
            checked', false);
    }
</script>
<!-- /clear -->

```

Listing 87: includes/header.blade.php

```

<!-- Top navigation -->
<div class="top-nav">
    <div class="nav-menu">
        <nav class="" role="navigation">
            <div class="nav-toggle">
                <a id="menu_toggle"><i class=""
                    fa_fa-bars"></i></a>
            </div>
            <ul class="nav_navbar-nav_navbar-
                right">
                <li class="">
                    <a href="javascript:;"
                        class="user-profile_
                        dropdown-toggle" data_
                        -toggle="dropdown"
                        aria-expanded="false">
                        {{ Auth::
                            user()->name }}
                        <span class="fa_fa-
                            angle-down"></
                            span>
                    </a>
                <ul class="dropdown-menu_
                    dropdown-usermenu_
                    animated_fadeInDown_
                    pull-right">
                    <li><a href="{{_route
                        ('users.show', _
                        Auth::user()->id
                        -)}}"> Profile</a>
                    </li>
                </ul>
            </li>
        </ul>
    </div>

```



```

< span
  class
  =
  "
  message
  "
  >
  < span
    class
    =
    "
    time
    "
    >{{
      Helper
      ::
      getTimeElapsedString
      (
      $user
      ->
      created_at
      )
    }}</
    span
  >
</
span
>
</a>
</li>
<?php
  $notif_ctr
  ++ ?>
@endforeach
@if(sizeof(
  $notifications
  [0]) > 2)
<li>
  <span
    class
    =
    "
    message
    "
    <<
    center
    >...</
    center
    ></
    span
  >
  </li>
  <?php
    $isSizeGreaterThan2
    =
    true
    ?>
  @endif
  <?php
    $notif_ctr
    = 0 ?>
  @foreach(
    $notifications
    [1] as
    $project)
    @if(
      $notif_ctr
      ==
      2)
      <?php
        break
        ?>
    @endif
    <li>
      <a>
        <
          span
            class
            =
            "
            image
            "
            >
          <
            img
            src
            =
            "
            {{
              URL
              ::
              asset
              (
              $project
              ->
              proj_image
              )
            }}
            alt
            =
            "
            Project
            -
            Image
            "
            />
          </
            span
          >
        <
          span
          >
          <
            span
            <
              span
            <
              a
            =
              href
            =
            "
            {{
              route
              (
              '
              projects
              .
              index
              '
              )
              .
              .
              .
              #
              tagged_pro
              :
            }}
            <
              class
            =
            "
            btn
            -
            btn
            -
            link
            -
            btn
            -
            sm
            "
            >
            Tagged
            Project
            :
            {{
              $project
              ->
              proj_name
            }}</
            a
            ></
            span
          >
        </
          span
        >
      <
    <
  <

```



```

        <span>LA Global EDCC</span>
    </a>
</div>
<div class="clearfix"></div>
<!-- /Logo and title -->

<!-- Menu profile quick info -->
<div class="profile">
    <div class="profile-pic">
        <a href="{$_route('users.show', $_Auth::user()->id)}"
            >name}" class="img-circle profile_img"></a>
    </div>
    <div class="profile-info">
        <span>Welcome,</span>
        <h2>
            <?php
                if (Auth::check()) {
                    echo Auth::user()
                        ->firstname .
                        " " . Auth::
                            user()->
                                lastname;
                }
            </h2>
        </div>
    </div>
<!-- /Menu profile quick info -->

<br />

<!-- Sidebar menu -->
<div id="sidebar-menu" class="main_menu_side_hidden-print-main-menu">
    <div class="menu_section">
        <h3> {{ Auth::user()->position }} </h3>
        <ul class="nav_side-menu">
            @if (Auth::user()->isAdmin())
                <li><a><i class="fa fa-home"></i> Admin Home
                    <span class="fa fa-chevron-down"></span>
                </a>
                <ul class="nav_child_menu" style="display:none">
                    <li><a href="{$_route('projects.index')}"><i class="fa fa-tasks"></i> Manage Projects </a>
                    </li>
                    <li><a href="{$_route('users.index')}"><i class="fa fa-users"></i> Manage Users </a>
                    </li>
                    <li><a href="{$_route('activities.notifications')}"><i class="fa fa-envelope-o"></i> Notifications
                    </a></li>
                </ul>
            @endif
            <li><a><i class="fa fa-building"></i> Projects <span class="fa fa-chevron-down"></span>
                <ul class="nav_child_menu" style="display:none">
                    <li><a href="{$_route('projects.viewActive', $_Auth::user()->id)}"><i class="fa fa-

```

```

edit"></i> My
Active
Projects </a
></li>
<li><a href="{$_route('projects.viewArchived')}"><i class="fa fa-archive"></i> Archived
Projects </a
></li>
</ul>
</li>
<li><a href="{$_route('activities.index')}"><i class="fa fa-file-text-o"></i>
Audit Trail </a>
</li>
</ul>
</div>
<!-- /Sidebar menu -->
</div>
</div>

```

Listing 90: includes/stylesheets.blade.php

```

<!-- Bootstrap core CSS -->
<link href="{$_URL::asset('css/bootstrap.min.css')}" rel="stylesheet">

<link href="{$_URL::asset('fonts/css/font-awesome.min.css')}" rel="stylesheet">
<link href="{$_URL::asset('css/animate.min.css')}" rel="stylesheet">

<!-- Custom styling plus plugins -->
<link href="{$_URL::asset('css/custom.css')}" rel="stylesheet">
<link href="{$_URL::asset('css/ichack/flat/green.css')}" rel="stylesheet" />
<link href="{$_URL::asset('css/maps/jquery-jvectormap-2.0.1.css')}" rel="stylesheet" type="text/css" />
<link href="{$_URL::asset('css/floatexamples.css')}" rel="stylesheet" type="text/css" />
<link href="{$_URL::asset('css/progressbar/pace-theme-center-simple.css')}" rel="stylesheet" type="text/css" />

```

Listing 91: includes/template.blade.php

```

<?php ini_set('max_execution_time', 300); ?>
<!-- title page -->
<page>
    <br/><br/><br/><br/>
    <br/><br/><br/><br/>
    {!! $section_bodies->first()->section_text !!}
</page>
<!-- /title page -->

<!-- other sections -->
<page backleft="5mm" backtop="5mm" backright="5mm" backbottom="5mm">
    <page_header>
    </page_header>
    <page_footer>
        <div style="text-align:left;font-size:.8pt">
            <b>Geotechnical Investigation Report</b><br>
            {!! $project->proj_name !!}
        </div>
        <div style="text-align:right;font-size:.8pt"> Page [[page_cu]] </div>
    </page_footer>
    @foreach($section_bodies as $section_body)
        @if($section_body->section_key != 0)
            {!! $section_body->section_text !!} <br/><br/><br/>
        @endif
    @endforeach
</page>
<!-- /other sections -->

```

```

<!-- appendices -->
<page>
  <br/><br/><br/><br/><br/>
  <div style="text-align:center;"><b>
    APPENDICES</b><br/><br/>
    @foreach($subreports as $subreport)
      {!! Helper::getSubreportName(
        $subreport->sb.key) !!}<br/><br/>
    @endforeach
  </div>

  <page_footer>
    <div style="text-align:left;_font-size:_8pt">
      <b>Geotechnical Investigation
      Report</b><br/>
      {{ $project->proj.name }}
    </div>
    <div style="text-align:right;_font-size:_8pt"> Page [[ page_cu]] </div>
  </page_footer>
</page>
<!-- /appendices -->

<!-- grain size analysis -->
@if($subreports->contains('sb.key', 0))
<page>
  @foreach($samples as $sample)
    <table align="center" style="width:100%;_font-size:9pt;_border:1px_solid;">
      <tr>
        <td style="width:70%;"><b>Project
          :</b> {{ $project->proj.name
        }}</td>
        <td style="width:30%;"><b>Job No
          .:</b> {{ $sample->
          sample.job_num }}</td>
      </tr>
      <tr>
        <td style="width:70%;"><b>Location
          :</b> {{ $project->
          proj.location }}</td>
        <td style="width:30%;"><b>Date
          Tested:</b> {{ $sample->
          sample.test_date }}</td>
      </tr>
    </table>
    <div style="text-align:center;">
      <?php
        $img_path = "images/subreports/GSA
          /sample-{$sample->id}.png";
      ?>
      
    </div>
    <table align="center" style="width
      :100%;_border:1px_solid;_border-
      collapse:collapse;_padding:0;_
      margin:10;_text-align:center;">
      <thead>
        <tr>
          <th style="width:20%;_
            border:1px_solid;">
            Borehole No.</th>
          <th style="width:20%;_
            border:1px_solid;">
            Sample No.</th>
          <th style="width:20%;_
            border:1px_solid;">
            Depth (m)</th>
          <th style="width:40%;_
            border:1px_solid;">
            Soil Description</th>
        </tr>
      </thead>
      <tbody>
        <tr style="border-top:1px_
          solid_black;">
          <th style="border:1px_
            solid;">{{ $sample->
            bh.id }}</th>
          <td style="border:1px_
            solid;">{{ $sample->
            sample.num }}</td>
          <td style="border:1px_
            solid;">{{ $sample->
            sample.from . "_-_" .
            $sample->sample_to
          }}</td>
          <td style="border:1px_
            solid;">{{ $sample->
            uscs.description }}</td>
        </tr>
      </tbody>
    </table>
  @endforeach
</page>
@endif
<!-- /grain size analysis -->

<!-- liquid and plastic limits test report -->
@if($subreports->contains('sb.key', 1))
<?php $i = 0 ?>
@foreach($samples as $sample)
<page>
  <br/><br/><br/><br/><br/>
  <table align="center" style="width:100%;_
    border:1px_solid;_border-collapse:
    collapse;_padding:0;_margin:10;_text-
    align:center;_font-size:8pt">
    <tr>
      <td style="width:40%;_border:1px_
        solid;"><b>Project:</b>{{
        $project->proj.name }}</td>
      <td style="width:30%;_border:1px_
        solid;"><b>Borehole-Sample No
        .:</b>{{ $sample->bh.id }} -
        {{ $sample->id }}</td>
      <td style="width:30%;_border:1px_
        solid;"><b>Depth:</b>{{
        $sample->sample_from }} m. -
        {{ $sample->sample_to }} m.</td>
    </tr>
    <tr>
      <td style="width:40%;_border:1px_
        solid;"><b>Job No.:</b> {{
        $sample->sample.job_num }}</td>
      <td style="width:30%;_border:1px_
        solid;"><b>Contract No.:</b>
        {{ $sample->
        sample.contract_num }}</td>
      <td style="width:30%;_border:1px_
        solid;"><b>Tested By:</b> {{
        $sample->sample_tester }}</td>
    </tr>
    <tr>
      <td style="width:40%;_border:1px_
        solid;"><b>Location:</b> {{
        $project->proj.location }}</td>
  </tr>
  </table>
</table>
</table>
</thead>
<tbody>
<tr>
<td style="border:1px_solid;">USCS</td>
<td style="border:1px_solid;">NMC (%)</td>
<td style="border:1px_solid;">>LL</td>
<td style="border:1px_solid;">>PL</td>
<td style="border:1px_solid;">>PI</td>
</tr>
</tbody>
</table>
</table>
</table>
</thead>
<tbody>
<tr>
<td style="border:1px_
solid;">{{ $sample->
uscs.id }}</td>
<td style="border:1px_
solid;">{{ $sample->
sample.nmc.percent
}}</td>
<td style="border:1px_
solid;">{{ $sample->
sample.liquid.limit
}}</td>
<td style="border:1px_
solid;">{{ $sample->
sample.plastic.limit
}}</td>
<td style="border:1px_
solid;">{{ $sample->
sample.plasticity.index
}}</td>
</tr>
</tbody>
</table>
<br/><br/><br/><br/>
@endif
<!-- /liquid and plastic limits test report -->

```

```

        <td style="width:30%;border:1px_
        solid;"><b>Test Date:</b> {{
        $sample->sample_test_date
        }}</td>
        <td style="width:30%;border:1px_
        solid;"></td>
    </tr>
</table>
<div style="text-align:center;">
    <?php $img_path = "images/subreports/
    LPTR/sample-{$sample->id}.png"; ?>
    
</div>
<table align="center" style="width:100%;_
border:1px_solid;_border-collapse:
collapse;_padding:0;_margin:10;_text-
align:center;">
<thead>
<tr>
<td colspan="3" style="text-
align:center;_border:1px_
solid;">Liquid Limit
Tests</td>
<td colspan="2" style="text-
align:center;_border:1px_
solid;">Plastic Limit
Tests</td>
</tr>
<tr>
<td style="text-align:center;_
border:1px_solid;">Run No
.</td>
<td style="text-align:center;_
border:1px_solid;">No. of
Blows</td>
<td style="text-align:center;_
border:1px_solid;">Water
Content</td>
<td style="text-align:center;_
border:1px_solid;">Run No
.</td>
<td style="text-align:center;_
border:1px_solid;">Water
Content</td>
</tr>
</thead>
<tbody>
    @foreach($liquid_limits as
    $liquid_limit)
        @if($liquid_limit->sample_id
        == $sample->id)
            <tr>
                <th style="width:20%;_
                border:1px_solid;">{{
                $liquid_limit->
                lim_run_num }}</th>
                <td style="width:20%;_
                border:1px_solid;">{{
                $liquid_limit->
                lim_num_of_blows }}</
                td>
                <td style="width:20%;_
                border:1px_solid;">{{
                $liquid_limit->
                lim_water_content
                }}</td>
                @if(sizeof($plastic_limits
                ) > $i &&
                $plastic_limits[$i]->
                sample_id == $sample
                ->id)
                    <th style="width:20%;_
                    border:1px_solid;
                    ">{{
                    $plastic_limits[
                    $i]->lim_run_num
                    }}</th>
                    <td style="width:20%;_
                    border:1px_solid;
                    ">{{
                    $plastic_limits[
                    $i]->
                    lim_water_content
                    }}</td>
                    <?php $i++ ?>
                @else
                    <td style="width:20%;_
                    border:1px_solid;
                    "></td>
                    <td style="width:20%;_
                    border:1px_solid;
                    "></td>
                @endif
            </tr>
        @endif
    @endforeach

```

```

    </tbody>
</table>
<table align="center" style="width:100%;_
border:1px_solid;_border-collapse:
collapse;_padding:0;_margin:10;_text-
align:center;">
<tbody>
<tr>
<td style="width:33%;border:1
px_solid;">Liquid Limit
(%) = {{ $sample->
sample_liquid_limit }}</
td>
<td style="width:33%;border:1
px_solid;">Plastic Limit
(%) = {{ $sample->
sample_plastic_limit }}</
td>
<td style="width:33%;border:1
px_solid;">Plasticity
Index = {{ $sample->
sample_plasticity_index
}}</td>
</tr>
<tr>
<td style="width:100%;border
:1px_solid;" colspan="3">
Remarks: {{ $sample->
sample_plasticity_remark
}}</td>
</tr>
</tbody>
</table>
</page>
@endforeach
@endif
<!-- /liquid and plastic limits test report
-->
<!-- borehole log -->
@if($subreports->contains('sb_key', 2))
@foreach($boreholes as $borehole)
<page>
<br/><br/><br/><br/><br/>
<?php $img_path = "images/subreports/BL/bh
-{$borehole->id}.png" ?>

</page>
@endforeach
@endif
<!-- /borehole log -->
<!-- soil profile -->
@if($subreports->contains('sb_key', 3))
<page>
<?php
    $num_of_bh = $project->
    proj_num_of_boreholes;
    $num_of_canvas = ceil($num_of_bh/2);
    ?>
    @for($i = 0; $i < $num_of_canvas; $i++)
        <br/><br/><br/><br/><br/>
        <?php $img_path = "images/subreports/
        SP/proj-{$project->id-$i}.png" ?>
        
    @endfor
</page>
@endif
<!-- /soil profile -->

```

Listing 92: includes/welcomejs.blade.php

```

<div id="custom_notifications" class="
custom_notifications_dsp_none">
<ul class="list-unstyled_notifications
_clearfix" data-
tabbed_notifications="notif-group">
</ul>
<div class="clearfix"></div>
<div id="notif-group" class="
tabbed_notifications"></div>
</div>
<script src="{{_URL::asset('js/bootstrap.
min.js')_}}"></script>
<!-- gauge js -->
<script src="{{_URL::asset('js/gauge/gauge
.min.js')_}}" type="text/javascript">
</script>

```

```

<script src="{_URL::asset('js/gauge/
gauge_demo.js ')}" type="text/
javascript"></script>
<!-- chart js -->
<script src="{_URL::asset('js/chartjs/
chart.min.js ')}"></script>
<!-- bootstrap progress js -->
<script src="{_URL::asset('js/progressbar
/bootstrap-progressbar.min.js ')}"
></script>
<script src="{_URL::asset('js/nicescroll/
jquery.nicescroll.min.js ')}"></
script>
<!-- icheck -->
<script src="{_URL::asset('js/icheck/
icheck.min.js ')}"></script>
<!-- daterangepicker -->
<script src="{_URL::asset('js/moment.min.
js ')}" type="text/javascript"></
script>
<script src="{_URL::asset('js/datepicker/
daterangepicker.js ')}" type="text/
javascript"></script>

<script src="{_URL::asset('js/custom.js ')}"
"></script>

<!-- flot js -->
<!--[if lte IE 8]><script type="text/
javascript" src="js/excanvas.min.js"
"></script><![endif]>
<script src="{_URL::asset('js/flot/jquery
.flot.js ')}" type="text/javascript"
"></script>
<script src="{_URL::asset('js/flot/jquery
.flot.pie.js ')}" type="text/
javascript"></script>
<script src="{_URL::asset('js/flot/jquery
.flot.orderBars.js ')}" type="text/
javascript"></script>
<script src="{_URL::asset('js/flot/jquery
.flot.time.min.js ')}" type="text/
javascript"></script>
<script src="{_URL::asset('js/flot/date.
js ')}" type="text/javascript"></
script>
<script src="{_URL::asset('js/flot/jquery
.flot.spline.js ')}" type="text/
javascript"></script>
<script src="{_URL::asset('js/flot/jquery
.flot.stack.js ')}" type="text/
javascript"></script>
<script src="{_URL::asset('js/flot/
curvedLines.js ')}" type="text/
javascript"></script>
<script src="{_URL::asset('js/flot/jquery
.flot.resize.js ')}" type="text/
javascript"></script>
<script>
$(document).ready(function () {
// [17, 74, 6, 39, 20, 85, 7]
//[82, 23, 66, 9, 99, 6, 2]
var data1 = [[gd(2012, 1, 1), 17],
[gd(2012, 1, 2), 74], [gd(
2012, 1, 3), 6], [gd(2012,
1, 4), 39], [gd(2012, 1, 5),
20], [gd(2012, 1, 6), 85], [
gd(2012, 1, 7), 7]];

var data2 = [[gd(2012, 1, 1), 82],
[gd(2012, 1, 2), 23], [gd(
2012, 1, 3), 66], [gd(2012,
1, 4), 9], [gd(2012, 1, 5),
119], [gd(2012, 1, 6), 6], [
gd(2012, 1, 7), 9]];
$("##canvas_dahs").length && $.plot
($("##canvas_dahs"), [
data1, data2
], {
series: {
lines: {
show: false,
fill: true
},
splines: {
show: true,
tension: 0.4,
lineWidth: 1,
fill: 0.4
},
points: {
radius: 0,
show: true
},
shadowSize: 2
},
grid: {
verticalLines: true,
hoverable: true,
clickable: true,
tickColor: "#d5d5d5",
borderWidth: 1,
color: '#fff'
},
colors: ["rgba(38,185,154,0.38)", "rgba(3,88,106,0.38)"],
xaxis: {
tickColor: "rgba(51,51,51,0.06)",
mode: "time",
tickSize: [1, "day"],
//tickLength: 10,
axisLabel: "Date",
axisLabelUseCanvas: true,
axisLabelFontSizePixels: 12,
axisLabelFontFamily: '
Verdana, Arial',
axisLabelPadding: 10
//mode: "time",
timeformat: "%m/%
d/%y",
minTickSize: [1,
"day"]
},
yaxis: {
ticks: 8,
tickColor: "rgba(51,51,51,0.06)",
},
tooltip: false
});

function gd(year, month, day) {
return new Date(year, month -
1, day).getTime();
}
});
</script>

<!-- worldmap -->
<script type="text/javascript" src="js/
maps/jquery-jvectormap-2.0.1.min.js"
"></script>
<script type="text/javascript" src="js/
maps/gdp-data.js"></script>
<script type="text/javascript" src="js/
maps/jquery-jvectormap-world-mill-en.
js"></script>
<script type="text/javascript" src="js/
maps/jquery-jvectormap-us-aea-en.js"
"></script>
<script>
$(function () {
$('#world-map-gdp').vectorMap({
map: 'world_mill_en',
backgroundColor: 'transparent',
zoomOnScroll: false,
series: {
regions: [{
values: gdpData,
scale: ['#E6F2F0', '#149B7E'],
normalizeFunction: '
polynomial'
}],
onRegionTipShow: function (e,
el, code) {
el.html(el.html() + '_(GDP
_' + gdpData[code]
+ ')');
}
});
});
</script>
<!-- skycons -->
<script src="js/skycons/skycons.js"></
script>
<script>
var icons = new Skycons({
"color": "#73879C"
}),
list = [
"clear-day", "clear-night", "
partly-cloudy-day",
"partly-cloudy-night", "cloudy
", "rain", "sleet", "snow
", "wind",
"fog"
],
i;
for (i = list.length; i--;)
icons.set(list[i], list[i]);

```

```

        icons.play();
    </script>

    <!-- dashbord linegraph -->
    <script>
        var doughnutData = [
            {
                value: 30,
                color: "#455C73"
            },
            {
                value: 30,
                color: "#9B59B6"
            },
            {
                value: 60,
                color: "#BDC3C7"
            },
            {
                value: 100,
                color: "#26B99A"
            },
            {
                value: 120,
                color: "#3498DB"
            }
        ];

        var myDoughnut = new Chart(document.
            getElementById("canvas1").
            getContext("2d")).Doughnut(
                doughnutData);
    </script>
    <!-- /dashbord linegraph -->
    <!-- datepicker -->
    <script type="text/javascript">
        $(document).ready(function () {

            var cb = function (start, end,
                label) {
                console.log(start.toISOString(),
                    end.toISOString(),
                    label);
                $('#reportrange-span').html(
                    start.format('MMMLD, -
                    YYYY') + ' - ' + end.
                    format('MMMLD, YYYY'));
                //alert("Callback has fired:
                [" + start.format('MMMM
                , YYYY') + " to " + end.
                format('MMMM D, YYYY') +
                ", label = " + label +
                ""]);
            }

            var optionSet1 = {
                startDate: moment().subtract
                    (29, 'days'),
                endDate: moment(),
                minDate: '01/01/2012',
                maxDate: '12/31/2015',
                dateLimit: {
                    days: 60
                },
                showDropdowns: true,
                showWeekNumbers: true,
                timePicker: false,
                timePickerIncrement: 1,
                timePicker12Hour: true,
                ranges: {
                    'Today': [moment(), moment
                        ()],
                    'Yesterday': [moment().
                        subtract(1, 'days'),
                        moment().subtract(1,
                            'days')],
                    'Last_7_Days': [moment().
                        subtract(6, 'days'),
                        moment()],
                    'Last_30_Days': [moment().
                        subtract(29, 'days'),
                        moment()],
                    'This_Month': [moment().
                        startOf('month'),
                        moment().endOf('month
                            ')],
                    'Last_Month': [moment().
                        subtract(1, 'month').
                        startOf('month'),
                        moment().subtract(1,
                            'month').endOf('month
                                ')]
                },
                opens: 'left',
                buttonClasses: ['btn btn-
                    default'],
                applyClass: 'btn-small btn-
                    primary',
                cancelClass: 'btn-small',
                format: 'MM/DD/YYYY',

```

```

                separator: '_to_',
                locale: {
                    applyLabel: 'Submit',
                    cancelLabel: 'Clear',
                    fromLabel: 'From',
                    toLabel: 'To',
                    customRangeLabel: 'Custom',

                    daysOfWeek: ['Su', 'Mo', '
                        Tu', 'We', 'Th', 'Fr',
                        'Sa'],
                    monthNames: ['January', '
                        February', 'March', '
                        April', 'May', 'June',
                        'July', 'August', '
                        September', 'October',
                        'November', '
                        December'],
                    firstDay: 1
                }
            };
            $('#reportrange-span').html(moment
                ().subtract(29, 'days').
                format('MMMLD, YYYY') + ' - ' +
                moment().format('MMMLD,
                YYYY'));
            $('#reportrange').daterangepicker(
                optionSet1, cb);
            $('#reportrange').on('show.
                daterangepicker', function ()
                {
                    console.log("show_event_fired"
                        );
                });
            $('#reportrange').on('hide.
                daterangepicker', function ()
                {
                    console.log("hide_event_fired"
                        );
                });
            $('#reportrange').on('apply.
                daterangepicker', function (
                    ev, picker) {
                console.log("apply_event_fired
                    , start/end_dates_are_" +
                    picker.startDate.format(
                        'MMMLD, YYYY') + "_to_"
                    + picker.endDate.format(
                        'MMMLD, YYYY'));
            });
            $('#reportrange').on('cancel.
                daterangepicker', function (
                    ev, picker) {
                console.log("cancel_event_
                    fired");
            });
            $('#options1').click(function () {
                $('#reportrange').data('
                    daterangepicker').
                    setOptions(optionSet1, cb
                    );
            });
            $('#options2').click(function () {
                $('#reportrange').data('
                    daterangepicker').
                    setOptions(optionSet2, cb
                    );
            });
            $('#destroy').click(function () {
                $('#reportrange').data('
                    daterangepicker').remove
                    ();
            });
        });
    </script>
    <script>
        NProgress.done();
    </script>
    <!-- /datepicker -->
    <!-- /footer content -->

```

Listing 93: layouts/errormas-
ter.blade.php

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta name="csrf-token" content="{ { _
        csrf_token() } }">
    <meta http-equiv="Content-Type" content="
        text/html; charset=UTF-8">

    <!-- Meta, title, CSS, favicons, etc. -->
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content
        ="IE=edge">

```

```

<meta name="viewport" content="width=device-width, _initial-scale=1">
<title>GeoTest | @yield('title')</title>
<!-- Bootstrap core CSS -->
<link href="{{_URL::asset('css/bootstrap.min.css')}}" rel="stylesheet">
<link href="{{_URL::asset('fonts/css/font-awesome.min.css')}}" rel="stylesheet">
<link href="{{_URL::asset('css/animate.min.css')}}" rel="stylesheet">
<!-- Custom styling plus plugins -->
<link href="{{_URL::asset('css/custom.css')}}" rel="stylesheet">
<link href="{{_URL::asset('css/ichack/flat/green.css')}}" rel="stylesheet" />
</head>
<body class="nav-md">
    <div class="container_body">
        <div class="main_container">
            <!-- page content -->
            <div class="col-md-12">
                <div class="col-middle">
                    <div class="text-center text-center">
                        <h1 class="error-number">@yield('error_name')</h1>
                        <h2>@yield('error_description1')</h2>
                        <p>@yield('error_description2')</p>
                        <p>{!! link_to(URL::previous(), 'Go back', array('class' => 'btn btn-link')) !!}</p>
                    </div>
                </div>
            </div>
            <!-- /page content -->
        </div>
        <!-- footer content -->
    </div>
    <div id="custom_notifications" class="custom-notifications_dsp_none">
        <ul class="list-unstyled_notifications_clearfix" data-tabbed_notifications="notif-group">
        </ul>
        <div class="clearfix"></div>
        <div id="notif-group" class="tabbed_notifications"></div>
    </div>
    <script type="text/javascript">
        $.ajaxSetup({
            headers: { 'X-CSRF-Token' : $('meta[name=_token']).attr('content') }
        });
    </script>
    <!-- /footer content -->
</body>
</html>

<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, _initial-scale=1">
<title>GeoTest | @yield('title')</title>
@section('stylesheets')
    @include('includes/stylesheets')
@show
</head>
<body class="nav-md">
    <div class="container_body">
        <div class="main_container">
            @include('includes/sidebar')
            @include('includes/header')
            <!-- page content -->
            <div class="right_col" role="main">
                <div class="">
                    <div class="page-title">
                        <div class="title_left">
                            @if(isset($project) && !isset($projects))
                                <h3> {!! link_to_route('projects.show', $project->proj_name, $project->id, array('class' => 'to_login')) !!} </h3>
                            @else
                                <h3> @yield('title') </h3>
                            @endif
                        </div>
                    </div>
                    <div class="clearfix"></div>
                    @section('flash_messages')
                        @if (session('success'))
                            <div class="x_content_bs-example-popovers">
                                <div class="alert alert-success alert-dismissible fade_in" role="alert">
                                    <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true"></span>
                                </div>
                                {!! session('success') !!}
                            </div>
                        @endif
                    @if (session('error'))
                        <div class="x_content_bs-example-popovers">
                            <div class="alert alert-danger

```

Listing 94: layouts/master.blade.php

```

        alert-
        dismissible_
        fade_in" role
        =" alert">
        <button type="
        button"
        class="
        close"
        data-
        dismiss="
        alert"
        aria-
        label="
        Close"><
        span aria
        -hidden="
        true">
        </span>
        </button>
        {{ session('
        error')}
        }}
    </div>
</div>
@endif
@if (session('warning'
))
<div class="x_content_
bs-example-
popovers">
<div class="alert_
alert-warning
_alert-
dismissible_
fade_in" role
=" alert">
<button type="
button"
class="
close"
data-
dismiss="
alert"
aria-
label="
Close"><
span aria
-hidden="
true">
</span>
</button>
{{ session('
warning')}
}}
</div>
</div>
@endif
@if (session('
warning-e'))
<div class="x_content_
bs-example-
popovers">
<div class="alert_
alert-warning
_alert-
dismissible_
fade_in" role
=" alert">
<button type="
button"
class="
close"
data-
dismiss="
alert"
aria-
label="
Close"><
span aria
-hidden="
true">
</span>
</button>
{!! session('
warning-e
') !!}
</div>
</div>
@endif
@if (count($errors) >
0)
<div class="alert_
alert-danger"
role=" alert"
>
@foreach ($errors
=>all() as
$error)
<span class="
        glyphicon
        -
        glyphicon
        -
        exclamation
        -sign"
        aria-
        hidden="
        true"></
        span>
        {{ $error }}
        <br/>
    @endforeach
</div>
@endif
@show
@yield('content')
</div>
</div>
</div>
<div id="custom_notifications" class="
custom_notifications_dsp_none">
<ul class="list-unstyled_notifications
_clearfix" data-
tabbed_notifications="notif-group"
">
</ul>
<div class="clearfix"></div>
<div id="notif-group" class="
tabbed_notifications"></div>
</div>
@section('js')
@include('includes/js')
@show
</body>
</html>

```

Listing 95: limits/create.blade.php

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
<div class="col-md-12">
<div class="x_panel">
<div class="x_title">
<h2>Add Liquid and Plastic
Limits Test Report</h2>
<ul class="nav navbar-right_
panel_toolbox">
<li><a class="collapse-
link"><i class="fa_fa
-chevron-up"></i></a>
</li>
<li class="dropdown">
<a href="#" class="
dropdown-toggle"
data-toggle="
dropdown" role="
button" aria-
expanded=" false"
><i class="fa_fa-
wrench"></i></a>
<ul class="dropdown-
menu" role=" menu"
>
<li><a href="#">
Settings 1</a>
</li>
<li><a href="#">
Settings 2</a>
</li>
</ul>
</li>
</ul>
</div>
<div class="clearfix"></div>
</div>
<div class="x_content">
@if($counts > 0)
{!! Form::open(['name' => '
update_form', 'method' => '
POST', 'route' => ['
limits.store', $project->
id], 'class' => 'form-
horizontal_label-
left', 'id' => '
update_form']) !!}
<?php $class_num = 0 ?>
@for($i = 0; $i < sizeof(

```



```

    $counts); $i++)
<h4 align="center">
    Borehole {{ $i + 1 }}
    <small>{!! link_to_route('
    boreholes.show', '(ID
    #' . $boreholes[$i] .
    ')', $boreholes[$i],
    array('class' => '
    to_login')) !!}</
    small>
</h4>
<table class="table table-
    hover" id="table{{$_$i_}}">
    >
    <thead>
    <tr>
    <th>Sample</th>
    <th>Test</th>
    <th>Water Content
    </th>
    <th>No. of Blows</
    th>
    <th width="18%">
    Options</th>
    </tr>
    </thead>
    <tbody>
    @for($j = 0; $j <
    sizeof($counts[$i
    ]); $j++)
    <tr>
    <th scope="row"
    style="border
    -top:1px_
    solid;">
    {!! Form::
    hidden('
    sample_id
    []',
    $counts[
    $i][$j])
    !!}
    <small>{!!
    link_to_route
    ('samples
    .show', '(ID#
    #' .
    $counts[
    $i][$j] .
    ')',
    $counts[
    $i][$j],
    array('
    class' =>
    '
    to_login'
    )) !!}</
    small>
    </th>
    <td style="border-
    top:1px_solid
    ;">
    Liquid Limit
    Test
    {!! Form::
    hidden('
    lim_test
    []', 0)
    !!}
    </td>
    <td style="border-
    top:1px_solid
    ;"><input
    type="number"
    name="
    lim_water_content
    []" class="
    form-control_
    to_clear_col-
    md-7_col-xs
    -12" step="
    any" required
    ></td>
    <td style="border-
    top:1px_solid
    ;"><input
    type="number"
    name="
    lim_num_of_blow
    s[]" class="
    form-control_
    to_clear_col-
    md-7_col-xs
    -12" step="
    any" required
    ></td>
    <td style="border-
    top:1px_solid
    ;">
    <button type =
    "button"
    onclick =
    ;">
    </tr>
    </tbody>
    </table>
    <?php $class_num++ ?>
    <tr>
    <th scope="row"
    style="border
    -top:1px_
    solid;">
    {!! Form::
    hidden('
    sample_id
    []',
    $counts[
    $i][$j])
    !!}
    <small>{!!
    link_to_route
    ('samples
    .show', '(ID#
    #' .
    $counts[
    $i][$j] .
    ')',
    $counts[
    $i][$j],
    array('
    class' =>
    '
    to_login'
    )) !!}</
    small>
    </th>
    <td style="border-
    top:1px_solid
    ;">
    Plastic Limit
    Test
    {!! Form::
    hidden('
    lim_test
    []', 1)
    !!}
    </td>
    <td style="border-
    top:1px_solid
    ;"><input
    type="number"
    name="
    lim_water_content
    []" class="
    form-control_
    to_clear_col-
    md-7_col-xs
    -12" step="
    any" required
    ></td>
    <td style="border-
    top:1px_solid
    ;"><input
    type="number"
    name="
    lim_num_of_blow
    s[]" class="
    form-control_
    to_clear_col-
    md-7_col-xs
    -12" step="
    any" required
    ></td>
    <td style="border-
    top:1px_solid
    ;">
    <button type =
    "button"
    onclick =
    ;">
    </tr>
    </tbody>
    </table>

```

```

        "addRow(
        this ,_{-
        $class_num
        }},_{-
        $counts[
        $i][$j]-
        }},_{-1}"
        class="){
        -
        $class_num
        }-btn-
        btn-sm-
        btn-info-
        addBtn"
        style="
        width:70
        px;"<i
        class="fa
        -fa-plus-
        m-right-
        sm"></i>
        Add </
        button>
        </td>
        </tr>
        <?php $class_num++ ?>
        @endfor
        </tbody>
        </table>
        @endfor
        <div class="ln_solid"></div>
        {!! Form::hidden('proj-id',
        $project->id) !!}
        {!! link_to(URL::previous(), '
        Back', array('class' => '
        btn btn-default')) !!}
        {!! Form::submit('Submit', ['
        class' => 'btn btn-
        primary', 'style' => '
        float:right']) !!}
        {!! Form::close() !!}
        @else
        {!! link_to(URL::previous(), '
        Back', array('class' => '
        btn btn-default')) !!}
        @endif
        </div>
        </div>
        </div>
        @endsection
        @section('js')
        @parent
        <script type="text/javascript">
        //$(document).ready(function () {
        addRow = function (parameter,
        class_num, sample_id, test) {
        $("." + class_num + "").addClass("
        previous");
        // Add row
        if (test == 0) {
        var td_text = "<input type='
        number' _name='
        lim_num_of_blows []' _class
        ='form-control-to-clear-
        col-md-7_col-xs-12' _step
        ='any'>";
        }
        else {
        var td_text = "N/A";
        }
        var current_row = $(parameter).
        closest("tr");
        current_row.after("<tr>
        + "<td><input type='hidden'
        _name='sample_id []' _
        value='" + sample_id
        + "'></td>"
        + "<td><input type='hidden'
        _name='lim_test []' _
        value='" + test + "
        '></td>"
        + "<td><input type='number' _name='
        lim_water_content []' _class='form-
        control-to-clear-
        col-md-7_col-xs-12' _
        step='any'></td>"
        + "<td>" + td_text + "</td>"
        + "<td><button type_='button' _onclick_='
        addRow(this ,_{-
        class_num + ",_{-
        sample_id + ",_{-
        test + ")'; _class
        =' + class_num + "_btn_btn-
        info_btn-
        sm_addBtn' _style='float:left;_width-
        :70px;'><i class='fa-fa-plus-m-right-
        sm"></i>Add</button>"
        + "<button type_='button' _
        onclick_='removeRow(
        this ,_{-
        class_num +
        ",_{-
        sample_id +
        ",_{-
        test + ")'; _
        class='btn btn-danger
        btn-sm' _style='float
        :center;_width:70px
        ;'><i class='fa-fa-
        trash-o_m-right-sm
        '></i>Delete</
        button></td>"
        + "</tr>");
        // Delete previous add button
        $("." + "previous").closest("button").
        remove();
        }
        removeRow = function (parameter,
        class_num, sample_id, test) {
        var current_row = $(parameter).
        closest("tr");
        var current_row_td = current_row.
        prev("tr").find("td:last");
        // Append add button to previous
        row if current row is last
        row of sample id
        if ($ (parameter).closest("tr").
        find("button").hasClass("
        addBtn")) {
        current_row_td.append("<button
        _type_='button' _onclick_='
        addRow(this ,_{-
        class_num + ",_{-
        sample_id + ",_{-
        test +
        "); _class='" +
        class_num + "_btn_btn-
        info_btn-sm_addBtn' _style
        ='float:left;_width:70px
        ;'><i class='fa-fa-plus-m-
        right-xs'></i>Add</
        button>");
        }
        // Remove row
        current_row.remove();
        }
        //});
        </script>
        <!-- form wizard -->
        <script src="{{_URL::asset('js/wizard/jquery.
        smartWizard.js')}}" type="text/
        javascript"></script>
        <script type="text/javascript">
        $(document).ready(function () {
        // Smart Wizard
        $('#wizard').smartWizard();
        function onFinishCallback() {
        $('#wizard').smartWizard('
        showMessage', 'Finish_Clicked
        ');
        }
        }
        $(document).ready(function () {
        // Smart Wizard
        $('#wizard-verticle').smartWizard({
        transitionEffect: 'slide'
        });
        });
        </script>
        <!-- /form wizard -->
        <!-- input mask -->
        <script src="{{_URL::asset('js/input_mask/
        jquery.inputmask.js')}}"></script>
        <script>
        $(document).ready(function () {
        $(':input').inputmask();
        });
        </script>
        <!-- /input mask -->
        @endsection
        @extends('layouts/master')
        @section('title', $project->proj_name)
        @section('content')
        <div class="row">
        <div class="col-md-12">

```

Listing 96: limits/edit.blade.php

```

<div class="x_panel">
  <div class="x_title">
    <h2>Edit Liquid and Plastic
      Limits Test Report</h2>
    <ul class="nav navbar-right
      panel_toolbox">
      <li><a class="collapse-
        link"><i class="fa fa
        -chevron-up"></i></a>
      </li>
      <li class="dropdown">
        <a href="#" class="
          dropdown-toggle"
          data-toggle="
            dropdown" role="
              button" aria-
                expanded="false"
                ><i class="fa fa
                wrench"></i></a>
        <ul class="dropdown-
          menu" role="menu"
          >
          <li><a href="#">
            Settings 1</a>
          </li>
          <li><a href="#">
            Settings 2</a>
          </li>
        </ul>
      </li>
    </ul>
  </div>
  <div class="clearfix"></div>
</div>
<div class="x_content">
  @if(sizeof($boreholes) == 0)
  <h4 align="center">There are
    no existing samples.</h4>
  @else
  {!! Form::open(['name' => '
    update_form', 'method' =>
      'PUT', 'route' => [
        limits.update',
        $subreport->id], 'class'
        => 'form-horizontal_
        label-left', 'id' => '
        update_form']) !!}
  <?php
    $class_num = 0;
    $i = 0;
    ?>
  @foreach($boreholes as
    $borehole)
  <h4 align="center">
    Borehole {{ $borehole->
      bh.num }}
    <small>{!! link_to_route('
      boreholes.show', '(ID
      #' . $borehole->id .
      ')', $borehole->id,
      array('class' => '
        to_login')) !!}</
      small>
  </h4>
  <table class="table table-
    hover" id="table{{$_$i-
    }}"
  >
  <thead>
  <tr>
  <th>Sample</th>
  <th>Test</th>
  <th>Water Content
    </th>
  <th>No. of Blows</
    th>
  </tr>
  </thead>
  <tbody>
  <?php $prev_sample_id
    = 0 ?>
  @foreach($samples as
    $sample)
  @if($sample->bh.id ==
    $borehole->id)
  @foreach($limits as
    $limit)
  @if($limit->sample_id
    == $sample->id)
  @if($prev_sample_id !=
    $sample->id)
  <?php $td_style = "
    border-top:1px_
    solid;" ?>
  @else
  <?php $td_style = ""
    ?>
  @endif
  @if($limit->lim_test
    == 1 && $limit->
    lim.run_num == 1)
  <?php $td_style = "
    border-top:1px_
    solid;" ?>
  @endif
  <tr>
  <th scope="row"
    style="{{$_
    $td_style-
    _}}">
  >
  @if(
    $prev_sample_id
    !=
    $sample->
    id)
  {{ $sample->
    sample.num
    }}
  <small>{!!
    link_to_route
    ('samples
    .show', '(ID#
    #' .
    $sample->
    id .
    ')',
    $sample
    ->id,
    array('
    class' =>
    '
    to_login'
    )) !!}</
    small>
  {!! Form::
    hidden('
    sample_id
    []',
    $sample->
    id) !!}
  {!! Form::
    hidden('
    limit_id
    []',
    $limit->
    id) !!}
  @else
  {!! Form::
    hidden('
    sample_id
    []',
    $sample->
    id) !!}
  {!! Form::
    hidden('
    limit_id
    []',
    $limit->
    id) !!}
  @endif
  </th>
  <td style="{{$_
    $td_style-
    _}}">
  >
  {{ ($limit->
    lim_test
    == 0) ? '
    Liquid_
    Limit_
    Test' : '
    Plastic_
    Limit_
    Test' }}
  {!! Form::
    hidden('
    lim_test
    []',
    $limit->
    lim_test)
    !!}
  </td>
  <td style="{{$_
    $td_style-
    _}}">
  <input type=
    "number" name
    =
    lim_water_content
    []" class="
    form-control_
    to_clear_col-
    md-7_col-xs
    -12" step="
    any" value="
    {{{_
    $limit->
    lim_water_content
    _}}}" required
  ></td>
  @if($limit->
    lim.num_of_blow
  >

```

```

        != -1)
        <td style="{{_
        $td_style_}}"
        ><input type=
        "number" name
        =
        lim_num_of_blows
        []" class="
        form-control_
        to_clear_col_
        md-7_col_xs
        -12" step="
        any" value="
        {{_ $limit ->
        lim_num_of_blows
        ->}}" required
        ></td>
        @else
        <td style="{{_
        $td_style_}}"
        >N/A</td>
        {!! Form::hidden('
        lim_num_of_blows
        []', -1) !!}
        @endif
        </tr>
        <?php $prev_sample_id
        = $sample->id ?>
        @endif
        @endforeach
        @endif
        <?php $class_num++ ?>
        @endforeach
        </tbody>
        </table>
        <?php $i++ ?>
        @endforeach
        <div class="ln_solid"></div>
        {!! Form::hidden('proj_id',
        $project->id) !!}
        {!! link_to(URL::previous(), '
        Back', array('class' => '
        btn btn-default')) !!}
        {!! Form::button('Clear', ['
        class' => 'btn btn-danger',
        'onclick' => '
        clearForm()']) !!}
        {!! Form::submit('Submit', ['
        class' => 'btn btn-
        primary', 'style' => '
        float:right']) !!}
        {!! Form::close() !!}
        @endif
        <div class="ln_solid"></div>
        </div>
        </div>
        </div>
        @endsection
        @section('js')
        @parent
        @include('includes/clearform')
        @endsection

```

Listing 97: limits/index.blade.php

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
    <div class="col-md-12">
        <div class="x_panel">
            <div class="x_title">
                <h2>Liquid and Plastic Limits
                Test Report - Liquid and
                Plastic Limits</h2>
                <ul class="nav navbar-right
                panel_toolbox">
                    <li><a class="collapse-
                    link"><i class="fa fa-
                    -chevron-up"></i></a>
                    </li>
                    <li class="dropdown">
                        <a href="#" class="
                        dropdown-toggle"
                        data-toggle="
                        dropdown" role="
                        button" aria-
                        expanded="false"
                        ><i class="fa fa-
                        wrench"></i></a>
                        <ul class="dropdown-
                        menu" role="menu"
                        >
                            <li><a href="#">
                                Settings 1</a>
                            >
                        </ul>
                    </li>
                </ul>
            </div>
            <table border="1" class="table_
            table-hover" id="table{{_ $i_}}">
                <thead>
                    <tr>
                        <th>Sample</th>
                        <th>Test</th>
                        <th>Water Content</th>
                        <th>No. of Blows</th>
                    </tr>
                </thead>
                <tbody>
                    <?php $prev_sample_id
                    = 0 ?>
                    @foreach($samples as
                    $sample)
                        @if($sample->bh_id ==
                        $borehole->id)
                            @foreach($limits as
                            $limit)
                                @if($limit->sample_id
                                == $sample->id)
                                    @if($prev_sample_id !=
                                    $sample->id)
                                        <?php $td_style = "
                                        border-top:1px_
                                        solid;" ?>
                                    @else
                                        <?php $td_style = ""
                                        ?>
                                    @endif
                                    @if($limit->lim_test
                                    == 1 && $limit->
                                    lim_run_num == 1)
                                        <?php $td_style = "
                                        border-top:1px_
                                        solid;" ?>
                                    @endif
                                    <tr>
                                        <th scope="row"
                                        style="{{_
                                        $td_style_}}">
                                            @if(
                                                $prev_sample_id
                                                !=
                                                $sample->
                                                id)
                                                {!! $sample->
                                                sample_num
                                                !!}
                                            <small>{!!
                                            link_to.route(
                                            'samples
                                            .show', '(ID#
                                            ' .
                                            $sample->
                                            id . ')',
                                            $sample
                                            ->id,
                                            array('
                                            class' =>
                                            '
                                            to_login'
                                            )) !!}</small>
                                            @endif
                                        </th>

```

```

<td style="{_{-
  $td_style_}}"}
  >
  {{ ($limit->
    lim_test
    == 0) ?
    "Liquid_
    Limit_
    Test" : "
    Plastic_
    Limit_
    Test" }}
</td>
<td style="{_{-
  $td_style_}}"}
  >{{ $limit->
    lim_water_content
  }}</td>
<td style="{_{-
  $td_style_}}"}
  >
  {{ ($limit->
    lim_num_of_blows
    == -1)
    ? "N/A" :
    $limit->
    lim_num_of_blows
  }}
</td>
</tr>
<?php $prev_sample_id
  = $sample->id ?>
  @endif
  @endforeach
  @endif
  @endforeach
</tbody>
</table>
<?php $i++ ?>
  @endforeach
<div class="ln_solid"></div>
{!! Form::open(['method' => '
  DELETE', 'route' => [ '
  limits_destroy',
  $subreport->id], '
  onsubmit' => 'return_
  confirm(" Delete_limits_
  and_plastic_limits_test_
  report_permanently?")'])
  !!}
{!! Form::submit('Delete', [ '
  class' => 'btn_btn-danger
  ']) !!}
<a href="{_{-route('limits.edit
  ',-$project->id)-}}"}"
  class="btn_btn-info"
  style="float:right"><i
  class="fa_fa-edit_m-right
  -xs"></i> Edit </a>
  {!! Form::close() !!}
  @endif
  <div class="ln_solid"></div>
</div>
</div>
</div>
@endsection

```

```

dropdown-toggle"
data-toggle="
dropdown" role="
button" aria-
expanded="false"
><i class="fa_fa-
wrench"></i></a>
<ul class="dropdown-
menu" role="menu"
>
<li><a href="#">
  Settings 1</a>
  >
</li>
<li><a href="#">
  Settings 2</a>
  >
</li>
</ul>
</li>
</ul>
</div>
<div class="clearfix"></div>
</div>
<div class="x_content">
  <!-- Smart Wizard -->
  <div id="wizard" class="
  form_wizard_
  wizard_horizontal">
  <ul class="wizard_steps">
  <li>
  <a href="#step-1">
  <span class="
  step_no">1</
  span>
  <span class="
  step_descr">
  Basic Details<br
  />
  </span>
  </a>
  </li>
  <li>
  <a href="#step-2">
  <span class="
  step_no">2</
  span>
  <span class="
  step_descr">
  Dates<br />
  </span>
  </a>
  </li>
  <li>
  <a href="#step-3">
  <span class="
  step_no">3</
  span>
  <span class="
  step_descr">
  Manage Members<br
  />
  </span>
  </a>
  </li>
  </ul>
  <!-- step 1 -->
  <div id="step-1">
  {!! Form::open(['name'
  => 'update_form'
  ', 'method' => '
  PUT', 'route' => [ '
  projects.update
  ', $project->id],
  'class' => 'form
  -horizontal_form-
  label-left', 'id'
  => 'update_form'
  ']) !!}
  <div class="form-group"
  >
  {!! Form::label('
  proj_name', '
  Project_Name_
  *', array('
  class' => '
  control-label
  _col-md-3_col
  -sm-3_col-xs
  -12')) !!}
  <div class="col-md
  -6_col-sm-6_
  col-xs-12">
  {!! Form::text
  ('
  proj_name
  ',
  $project
  ->
  proj_name

```

Listing 98: projects/edit.blade.php

```

@extends('layouts/master')
@section('title', 'Edit-' . $project->
  proj_name)
@section('stylesheets')
  @parent
<style>
  .check {
  opacity:0.5;
  color:#996;
  }
</style>
@endsection
@section('content')
<div class="row">
  <div class="col-md-12_col-sm-12_col-xs-12">
  >
  <div class="x_panel">
  <div class="x_title">
  <h2> Project Details </h2>
  <ul class="nav_navbar-right_
  panel_toolbox">
  <li><a class="collapse-
  link"><i class="fa_fa
  -chevron-up"></i></a>
  </li>
  <li class="dropdown">
  <a href="#" class="

```

```

        , ['class
        , => ,
        form-
        control_
        col-md-7_
        col-xs-12
        , ,
        required'
        => 'true
        ']) !!}
    </div>
</div>
<div class="form-group
">
    {!! Form::label('
    proj_location
    , 'Location_
    *', array('
    class' => ,
    control-label_
    _col-md-3_col
    -sm-3_col-xs
    -12')) !!}
    <div class="col-md
    -6_col-sm-6_
    col-xs-12">
        {!! Form::text
        (
        ,
        proj_location
        ,
        $project
        ->
        proj_location
        , ['class
        ' => ,
        form-
        control_
        col-md-7_
        col-xs-12
        , ,
        required'
        => 'true
        ']) !!}
    </div>
</div>
<div class="form-group
">
    {!! Form::label('
    proj_client' ,
    'Client_*' ,
    array('class'
    => 'control-
    label_col-md
    -3_col-sm-3_
    col-xs-12'))
    !!}
    <div class="col-md
    -6_col-sm-6_
    col-xs-12">
        {!! Form::text
        (
        ,
        proj_client
        ,
        $project
        ->
        proj_client
        , ['class
        ' => ,
        form-
        control_
        col-md-7_
        col-xs-12
        , ,
        required'
        => 'true
        ']) !!}
    </div>
</div>
<div class="form-group
">
    {!! Form::label('
    proj_description
    ,
    Description' ,
    array('class
    ' => 'control
    -label_col-md
    -3_col-sm-3_
    col-xs-12'))
    !!}
    <div class="col-md
    -6_col-sm-6_
    col-xs-12">
        {!! Form::
        textarea(
        ,
        proj_description
        ,
        $project
        ,
        proj_description
        , ['class
        ' => ,
        form-
        control_
        col-md-7_
        col-xs-12
        , ,
        required'
        => 'true
        ']) !!}
    </div>
</div>
<div class="form-
horizontal_form-
label-left">
    <div class="form-
    group">
        {!! Form::
        label('
        proj_start_date
        , 'Start
        _Date_*' ,
        array('
        class' =>
        'control
        -label_
        col-md-3_
        col-sm-3_
        col-xs-12
        ')) !!}
    <div class="
    col-md-6_
    col-sm-6_
    col-xs-12
    ">
        {!! Form::
        date(
        ,
        proj_start_date
        ,
        $project
        ->
        proj_start_date
        , ['
        class
        ' =>
        'form
        -
        control
        _col-
        md-7_
        col-
        xs-12
        ' ,
        required
        ' =>
        'true
        ' ,
        form'
        => ,
        update_form
        '])
        !!}
    </div>
</div>
<div class="form-
group">
    {!! Form::
    label('
    proj_end_date
    , 'End_
    Date_*' ,
    array('
    class' =>
    'control
    -label_
    col-md-3_
    col-sm-3_
    col-xs-12
    ')) !!}
    <div class="
    col-md-6_
    col-sm-6_
    col-xs-12
    ">
        {!! Form::
        date(
        ,
        proj_end_date
        ,
        $project
        ->
        proj_end_date
        , ['
        class
        ' =>

```

```

        'form
        -
        control
        _col-
        md-7-
        col-
        xs-12
        , ,
        required
        , =>
        'true
        , ,
        form'
        => '
        update_form
        ')}
        !!}
    </div>
</div>
<div class="form-
group">
    {!! Form::
    label(
    'proj_date_signed
    , 'Date_
    Signed_*'
    , array(
    'class' =>
    'control
    -label_
    col-md-3_
    col-sm-3_
    col-xs-12
    ') !!}
    <div class="
    col-md-6_
    col-sm-6_
    col-xs-12
    ">
        {!! Form::
        date(
        ,
        proj_date_signed
        ,
        $project
        ->
        proj_date_signed
        , [
        'class
        ' =>
        'form
        -
        control
        _col-
        md-7-
        col-
        xs-12
        , ,
        required
        ' =>
        'true
        , ,
        form'
        => '
        update_form
        ')}
        !!}
    </div>
    </div>
</div>
<!-- /step 2 -->
<!-- step 3 -->
<div id="step-3">
    <div class="form-
    horizontal_form-
    label-left">
        <div class="form-
        group" align=
        "center">
            @if(Auth::user
            ()->
            isAdmin()
            )
            <?php
            $members =

                Helper
                ::
                getMembers
                ();
            $engineers
            =
            Helper
            ::
            getEngineers
            (
            $project
            ->id)
        ;

        for($i =
        0; $i
        <
        sizeof
        (
        $engineers
        ); $i
        ++){
            $engineer_id_array
            [
            $i
            ]
            =
            $engineers
            [
            $i
            ]->
            id
            ;
        }
        ?>
        @foreach(
        $members
        as
        $member)
        <?php
        $checkbox_class
        = ""
        ;
        $is_checked
        =
        null;
        ?>
        @if(in_array(
        $member->
        id,
        $engineer_id_array
        ))
        <?php
        $checkbox_class
        = "
        check
        ";
        $is_checked
        =
        true;
        ?>
        @endif
        <div class="
        col-md-3"
        >
            <label
            class
            ="btn
            _btn-
            primary
            ">
                
                lastname
                -}}"
                class
                ="img
                -
                thumbnail
                _img-
                check
                -{{-
                $checkbox_class
                -}}"
                width
                ="150
                px">
                {!! Form::
                checkbox
                (
                'project_members
                []' ,
                $member
                ->id ,
                $is_checked
                , [
                'class

```



```

class
=
"
avatar
"

alt
=
"
{{
-
$engr
->
}
}
}

</li>
@endforeach

</ul>
</td>
<td>
<button type="
button"
class="
btn btn-
warning btn-
xs"
style="
width:70
px">
Tagged </
button>

</td>
<td>
<table border=
"0">
<tr><td>
<a

href
=
"
{{
-
route
(
'
projects
.
show
,
,
$proj
->
id
)
}
}
"

class
=
"
btn
-
btn
-
info
-
btn
-
xs
"

style
=
"
width
:70
px
"
">
</i
i

class
=
"
fa
-
fa
-
pencil
"
">
</i
i

Edit

</a
a
">

</td>
<td>
<a

href
=
"
{{
-
route
(
'
projects
.
archive
,
,
$proj
->

```

```

id
)
}}
"

class
=
"
btn
_
btn
-
dark
_
btn
-
xs
"

style
=
"
width
:70
px
"
<<
i
class
=
"
fa
_
fa
-
archive
"
></
i
>
Archive

</
a
>
</td>
<td>
{!!

Form
::
open
([
,
method
,
=>
,
DELETE
,
,
route
,
=>
[
,
projects
,
destroy
,
,
$proj
->
id
],
onsubmit
,
=>
</div>
,
</div>
</div>
<!-- /tagged projects -->
<!-- /tagged projects -->
<div class="row">
Delete<div class="col-md-12">
-
project
-
,
.
$proj
->
proj-name
,
_
permanently
?)
,
])
!!}

{!!
Form
::
submit
(
,
Delete
,
,
[
,
class
,
=>
,
btn
_
btn
-
danger
_
btn
-
xs
,
,
style
,
=>
,
width
:70
px
,
])
!!}

{!!
Form
::
close
()
!!}
</td>
</tr>
</tbody>
@endif
@endforeach
$proj
->
id
@else
<center>
<h2>There are no tagged
projects.</h2>
</center>
@endif
<div class="ln_solid"></div>
=>
</div>
</div>
</div>
</div>
<!-- /tagged projects -->
<!-- /tagged projects -->
<div class="row">
Delete<div class="col-md-12">

```

```

<div class="x_panel">
  <div class="x_title">
    <h2>All Projects </h2>
    <ul class="nav navbar-right _
      panel_toolbox">
      <li><a class="collapse-
        link"><i class="fa fa-
          -chevron-up"></i></a>
      </li>
      <li class="dropdown">
        <a href="#" class="
          dropdown-toggle"
          data-toggle="
            dropdown" role="
              button" aria-
                expanded="false"
                ><i class="fa fa-
                  wrench"></i></a>
        <ul class="dropdown-
          menu" role="menu"
            >
          <li><a href="#">
            Settings 1</a>
          </li>
          <li><a href="#">
            Settings 2</a>
          </li>
        </ul>
      </li>
    </ul>
  </div>
  <div class="clearfix"></div>
</div>
<div class="x_content">
  @if( $projects->isEmpty() )
    <center>
      <h2>There are no projects
        .</h2>
    </center>
  @else
    <table class="table table-
      striped table-hover _
        projects">
      <thead>
        <tr>
          <th style="width:_
            10px">ID</th>
          <th style="width:_
            30%">Project
            Name</th>
          <th style="width:_
            20%">Team
            Members</th>
          <th style="width:_
            10%">Status</
            th>
          <th style="width:_
            30%">Actions
          </th>
        </tr>
      </thead>
      <tbody>
        @foreach( $projects as
          $proj )
          <tr>
            <td>{{ $proj->id
              }}</td>
            <td>
              <a href="{{_
                route('
                  projects.
                    show', _
                      $proj->id
                    )-}}">{{
                      $proj->
                        proj_name
                      }}</a>
              <br />
              <small>Created
                {{ $proj
                  ->
                    created_at
                  }}</
                  small>
            </td>
            <td>
              <ul class="
                list-
                  inline">
                <?php
                  $engineers
                    =
                      Helper
                        ::
                          getEngineers
                            (
                              $proj
                                ->id)

```

```

        btn
        ~
        btn
        ~
        warning
        ~
        btn
        ~
        xs
        ~
        ;
        break
        ;

    case 2:
        $status
        = "
        Archived
        ";
        $btn

        =
        "
        btn
        ~
        btn
        ~
        dark
        ~
        btn
        ~
        xs
        ~
        ;
        break
        ;

        default:
            break
            ;
    }
}
<td>
<button type="
button"
class="{{
~$btn~}}"
style="
width:70
px">{{
$status
}}</
button>
</td>
<td>
<table border=
"0">
<tr>
<td>
<a
href
=
"
{{
~
route
('
projects
.
edit
',
~
$proj
->
id
)
}}
"
class
=
"
btn
~
btn
~
info
~
btn
~
xs
"
style
=
"
width
:70
px
"
></
i
"
class
=
"
fa
~
fa
~
pencil
"
></
i
>
Edit
</
a

```

```

>
</td>
@endif
@if(
    $proj
    ->
    isTagged
    ()
)
<td>
<a
    href
    =
    "
    {{
    -
    route
    (
    '
    projects
    .
    archive
    ,
    -
    $proj
    ->
    id
    )
    -
    }}
    "
    class
    =
    "
    btn
    -
    btn
    -
    dark
    -
    btn
    -
    xs
    "
    style
    =
    "
    width
    :70
    px
    "
    <
    i
    class
    =
    "
    fa
    -
    fa
    -
    tag
    "
    ></
    i
    >
    Tag
    </
    a
    >
</td>
@endif
<td>
{!!
Form
::
open
([
,
method
,
=>
,
DELETE
,
,
,
route
,
=>
[
,
projects
.
destroy
,
,
$proj
->
id
],
,
onsubmit
,
=>
,
return
-

```

```

@extend('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
  <div class="col-md-12">
    <div class="x_panel">
      <div class="x_title">
        <h2> Project Information </h2>
        <ul class="nav navbar-right panel_toolbox">
          <li><a class="collapse-link" <i class="fa fa-chevron-up"></i></a>
          <li class="dropdown">
            <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-expanded="false" <i class="fa fa-wrench"></i></a>
            <ul class="dropdown-menu" role="menu">
              <li><a href="#"> Settings 1</a>
              <li><a href="#"> Settings 2</a>
            </ul>
          </li>
        </ul>
      </div>
      <div class="clearfix"></div>
      <div class="x_content">
        <!-- project image -->
        <div class="col-md-3_col-sm-3_col-xs-12_profile_left">
          <div class="profile_img">
            <!-- end of image cropping -->
            <div id="crop-avatar">
              <!-- Current avatar -->
              <div class="avatar-view" title="Change the avatar">
                
              </div>
              <!-- Cropping modal -->
              <div class="modal-fade" id="modal-avatar-modal" aria-hidden="true" aria-labelledby="modal-label" role="dialog" tabindex="-1">
                <div class="modal-dialog modal-lg">
                  <div class="modal-content">
                    <form class="avatar-form" action="">

```

Listing 100: projects/show.blade.php

```

.php
"
enctype
=
"
multipart
/
form
-
data
"

method
=
"
post
"
>
<
    div
        class
        =
        "
        modal
        -
        header
        "
        >
    <
        button
            class
            =
            "
            close
            "

            data
            -
            dismiss
            =
            "
            modal
            "

            type
            =
            "
            button
            "
            >&
            times
            ;</
            button
            >

    <
        h4
            class
            =
            "
            modal
            -
            title
            "

            id
            =
            "
            avatar
            -
            modal
            -
            label
            "
            >
            Change
            Image
            </
            h4
            >
</
div
>

<
    div
        class
        =
        "
        modal

```

```

-
body
"
>
<
    div
        class
        =
        "
        avatar
        -
        body
        "
        >
    <!--
        Upload
        image
        and
        data
        -->
    <
        div
            class
            =
            "
            avatar
            -
            upload
            "
            >
    <
        in
            cl
            =
            "
            av
            -
            sr
            "
            nar
            =
            "
            av
            "
            ty
            =
            "
            hi
            "
            >
    <
        in
            cl
            =
            "
            av
            -
            da
            "
            nar
            =
            "
            av
            "
            ty
            =
            "
            hi
            "
            >
    <
        la
        fo
        =
        "
        av
        "
        >
        Lo

```



```

        upload
      </label>
    >
  <
    <input
      class
      =
      "
      avatar
      -
      input
      "

      id
      =
      "
      avatarInput
      "

      name
      =
      "
      avatar_file
      "

      type
      =
      "
      file
      "
    >
  </div>
<!--
  Crop
  and
  preview
-->
<
  <div
    class
    =
    "
    row
    "
  >
    <
      <div
        class
        =
        "
        col
        -
        md
        -9
        "
      >
        <
          <div
            class
            =
            "
            avatar
            -
            wrapper
            "
          ></div>
        </div>
      </div>
    <
      <div
        class
        =
        "
        col
        -

```

```

"
  data
  -
  method
  =
  "
  rotate
"

  data
  -
  option
  =
  "
  -90
"

  type
  =
  "
  button
"

  title
  =
  "
  Rotate
  -
  -90
  -
  degrees
"
>
Rotate
Left
</
button
>
<
  button
  class
  =
  "
  btn
  -
  btn
  -
  primary
"

  data
  -
  method
  =
  "
  rotate
"

  data
  -
  option
  =
  "
  -15
"

  type
  =
  "
  button
"
  >-15
  deg
  </
  button
  >
<
  button
  class
  =
  "
  btn
  -
  btn
  -
  primary
"

  data
  -
  method
  =
  "

```

</

<

```

data
-
option
=
"
90
"

type
=
"
button
"

title
=
"
Rotate
-
90
-
degrees
"
>
Rotate
Right
</
button
>
<
button
class
=
"
btn
-
btn
-
primary
"
data
-
method
=
"
rotate
"
data
-
option
=
"
15
"
type
=
"
button
"
>15
deg
</
button
>
<
button
class
=
"
btn
-
btn
-
primary
"
data
-
method
=
"
rotate
"
data
-
option
=
"
30
"

```

```

</
di
>
<
di
cl
=
"
co
-
md
-3
"
>
<

```

```

    </div>
    <!-- project info -->
    <div class="col-md-9 col-sm-9 col-xs-12">
      <div class="" role="tabpanel" data-example-id="toggleable-tabs">
        <ul id="myTab" class="nav nav-tabs bar-tabs" role="tablist">
          <li role="presentation" class="active"><a href="#" tab_content1" role="tab" id="home-tab" data-toggle="tab" aria-expanded="false">Project Details</a>
        </ul>
        <div id="myTabContent" class="tab-content">
          <!-- project details -->
          <div role="tabpanel" class="tab-pane fade active_in" id="tab_content1" aria-labelledby="home-tab">
            <div class="col-md-5 col-sm-5 col-xs-12">
              <div class="project-detail">
                <p class="title">Client</p>
                <p>{<!-- project -->proj_client}</p>
                <p>{<!-- project -->proj_location}</p>
                <p>{<!-- project -->proj_description}</p>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
  <!-- end of image cropping -->
</div>
<br />
<a href="{<!-- route('projects.viewCompiledReport', $project->id) -->}" class="btn btn-primary btn-sm" style="width:220px"><i class="fa fa-eye_m-right-xs"></i> View Compiled Report </a>
<!-- if($project->isActive()) -->
<a href="{<!-- route('projects.edit', $project->id) -->}" class="btn btn-info btn-sm" style="width:150px"><i class="fa fa-edit_m-right-xs"></i> Edit Project Details </a>
<a href="{<!-- route('projects.tag', $project->id) -->}" class="btn btn-warning btn-sm" style="width:65px"><i class="fa fa-tag_m-right-xs"></i> Tag </a>
<!-- elseif($project->isTagged()) -->
<a href="{<!-- route('emails.create', $project->id) -->}" class="btn btn-primary btn-sm" style="width:220px"><i class="fa fa-send_m-right-xs"></i> Create Email </a>
<a href="{<!-- route('projects.edit', $project->id) -->}" class="btn btn-info btn-sm" style="width:140px"><i class="fa fa-edit_m-right-xs"></i> Edit Project Details </a>
<a href="{<!-- route('projects.archive', $project->id) -->}" class="btn btn-dark btn-sm" style="width:75px"><i class="fa fa-archive_m-right-xs"></i> Archive </a>
</endif>

```

```

)
)
>
0)
<p
  class
  =
  "
  title
  "
  >
  Description
  </
  p
  >
<p>{{
  $project
  ->
  proj_description
  }}</
  p
  >
@endif
<p
  class
  =
  "
  title
  "
  >
  Project
  Members
  </
  p
  >
<ul
  style
  =
  "
  list
  -
  style
  -
  type
  :
  -
  none
  ;
  -
  padding
  -
  left
  :0;
  "
  >
  <?
  php
  $engineers
  =
  Helper
  ::
  getEngineers
  (
  $project
  ->
  id
  )
  @endforeach
  ?>
  </ul>
@foreach
  (
  $engineers
  as
  $engr
  )
  <
  li
  >
  <a
  href
  =
  "
  {{
  -
  route
  (
  ?
  users
  .
  show
  ',
  -
  $engr
  ->
  id
  )
  -
  }}
  class
  =
  "
  btn
  -
  btn
  -
  link
  -
  btn
  -
  xs
  "
  ></
  i
  class
  =
  "
  fa
  -
  fa
  -
  user
  -
  m
  -
  right
  -
  xs
  "
  ></
  i
  >
  {{
  $engr
  ->
  firstname
  .
  "
  -
  "
  .
  $engr
  ->
  lastname
  }}
  </
  a
  ></
  li
  >
  @endforeach
  </div>
  </div>
  <div class="
  col-md-5_
  col-md-5_
  col-xs-12
  ">
  <div class
  =
  "
  project_detail
  ">
  </div>
  </div>
  </div>
  <!-- /project
  details -->
  </div>
  </div>
  <!-- /project info -->
  <div class="ln.solid"></div>
  </div>

```



```

        -pencil"></i>
        Edit </a></td>
<td>
    {!! Form::open(['
    method' => '
    DELETE', '
    route' => ['
    shallows.
    destroy',
    $project->id
    ], 'onsubmit'
    => 'return_
    confirm("
    Delete_SBC_
    data_for_
    shallow_
    foundation_
    permanently
    ?")')] !!}
    {!! Form::submit('
    Delete', ['
    class' => '
    btn_btn_
    danger_btn_xs
    ', 'style' =>
    'width:70px'
    ]) !!}
    {!! Form::close()
    !!}
</td>
<td></td>
@else
<td><a href="{{_route
('shallows.create
',_,$project->id)-
}}" class="btn_
btn_btn_
primary_btn_xs"
style="width:
70px"><i class="
fa_fa_
plus_m_
right_xs"></i>
Compute </a></td>
<td></td>
<td></td>
<td></td>
@endif
</tr>
<tr>
<td><h2>Deep Foundation</
td>
@if(sizeof($deeps) >
0)
<td><a href="{{_route
('deeps.index',_
,$project->id)-
}}" class="btn_
btn_btn_
primary_btn_xs"
style="width:70px
"><i class="fa_fa_
-eye"></i> View
</a></td>
<td><a href="{{_route
('deeps.edit',_
,$project->id)-
}}" class="btn_
btn_btn_
info_btn_xs"
style="width:70px
"><i class="fa_fa_
-pencil"></i> Edit
</a></td>
<td>
    {!! Form::open(['
    method' => '
    DELETE', '
    route' => ['
    deeps.destroy
    ', $project->
    id], '
    onsubmit' =>
    'return_
    confirm("
    Delete_SBC_
    data_for_deep_
    foundation_
    permanently
    ?")')] !!}
    {!! Form::submit('
    Delete', ['
    class' => '
    btn_btn_
    danger_btn_xs
    ', 'style' =>
    'width:70px'
    ]) !!}
    {!! Form::close()
    !!}
</td>
<td></td>
@else
<td><h2>Drilled Shaft
Foundation</td>
@if(sizeof(
$drilledshafts)
> 0)
<td><a href="{{_route
('drilledshafts.
index',_,$project
->id)-
}}" class="
btn_btn_
primary_
btn_xs" style="
width:70px"><i
class="fa_fa_
-eye"
"></i> View </a></
td>
<td><a href="{{_route
('drilledshafts.
edit',_,$project->
id)-
}}" class="
btn_btn_
info_btn_
xs" style="
width:70px"><i
class="fa_fa_
-pencil"></i> Edit
</a></td>
<td>
    {!! Form::open(['
    method' => '
    DELETE', '
    route' => ['
    drilledshafts.
    destroy',
    $project->id
    ], 'onsubmit'
    => 'return_
    confirm("
    Delete_SBC_
    data_for_
    drilled_shaft_
    foundation_
    permanently
    ?")')] !!}
    {!! Form::submit('
    Delete', ['
    class' => '
    btn_btn_
    danger_btn_xs
    ', 'style' =>
    'width:70px'
    ]) !!}
    {!! Form::close()
    !!}
</td>
<td></td>
@else
<td><a href="{{_route
('drilledshafts.
create',_,$project
->id)-
}}" class="
btn_btn_
primary_
btn_xs" style="
width:70px"><i
class="fa_fa_
plus_
m_
right_xs"></i>
Compute </a></td>
<td></td>
<td></td>
<td></td>
@endif
</tr>
</table>
@else
<center>
<h2>You are not permitted
to modify this
resource.</h2>
</center>
@endif
<div class="ln_solid"></div>
</div>
</div>
</div>

```

```

<!-- /soil bearing capacities -->
</div>
<div class="row">
  <!-- report sections -->
  <div class="col-md-6_col-sm-6_col-xs-12">
    <div class="x_panel">
      <div class="x_title">
        <h2> Report Sections </h2>
        <ul class="nav navbar-right _
panel_toolbox">
          <li><a class="collapse-
link"><i class="fa_fa
-chevron-up"></i></a>
        </li>
          <li class="dropdown">
            <a href="#" class="
dropdown-toggle"
data-toggle="
dropdown" role="
button" aria-
expanded="false"
><i class="fa_fa-
wrench"></i></a>
            <ul class="dropdown-
menu" role="menu"
>
              <li><a href="#">
                Settings 1</a>
              </li>
              <li><a href="#">
                Settings 2</a>
              </li>
            </ul>
          </li>
          <li><a class="close-link"
><i class="fa_fa-
close"></i></a>
        </li>
      </ul>
    </div>
    <div class="clearfix"></div>
    <div class="x_content">
      <?php
        $section_list = array("
          Title_Page",
            "
            Introduction
            "
            ,
            "
            Drilling
            _
            Procedure
            "
            ,
            "
            Laboratory
            _
            Test
            "
            ,
            "
            Site
            _
            Geology
            "
            ,
            "
            Findings
            "
            ,
            "
            Conclusions
            _
            and
            _
            Recommendations
            "
            ,
            "
            Limitations
            /
            Comments
            "
            ,
            "
            Other
            _
            Section
            "
          )
          $sarr = range(0,8);
          ?>
          @if(sizeof($sections) > 0)
          <?php $ctr2 = -1; ?>
          @foreach($sections as $section
          )
          <?php
            $ctr2++;
            $temp_array[$ctr2] =
              $section->section_key
                . "|" . $section->id
              ;
          ?>
          @endforeach
          <?php sort($temp_array) ?>
          <table class="table table-
striped table-hover"
cellpadding="10">
            @for($x = 0; $x < sizeof(
              $temp_array); $x++)
            <?php $exploded_section =
              explode("|",
                $temp_array[$x]); ?>
            <tr>
              <td>{{ $section_list [
                $exploded_section
                [0]] }}</td>
              <td><a href="{{ $_route
                ('sections.show',
                  _
                    $exploded_section
                    [1])_ }}" class="
                    btn_btn-primary-
                    btn-xs" style="
                    width:70px"><i
                    class="fa_fa-eye"
                    ></i> View </a></
                    td>
              <td><a href="{{ $_route
                ('sections.edit',
                  _
                    $exploded_section
                    [1])_ }}" class="
                    btn_btn-info_btn-
                    xs" style="width
                    :70px"><i class="
                    fa_fa-pencil"></i>
                    > Edit </a></td>
            </tr>
            <td>
              {!! Form::open(['
                method' =>
                'DELETE',
                'route' => ['
                sections.
                destroy',
                $exploded_section
                [1]],
                'onsubmit' =>
                'return _
                confirm("
                Delete _
                section _'.
                $section_list
                [
                $exploded_section
                [0]] . ' _
                permanently
                ?")' ) !!}
              {!! Form::submit('
                Delete', ['
                class' =>
                '
                btn_btn-
                danger_btn-xs
                ', 'style' =>
                'width:70px'
                ]) !!}
              {!! Form::close()
                !!}
            </td>
          </tr>
          </table>
          @if($exploded_section[0]
            != 8)
          <?php $sarr[
            $exploded_section[0]]
            = -1 ?>
          @endif
          @endif
        </table>
        @else
        <center>
          <h2>This project currently
            has no report
            sections.</h2>
        </center>
        @endif
      </div>
    </div>
  </div>
</div>
)
;

```



```

, 1))
<?php $subreport =
$subreports->
where('sb_key',
1)->first() ?>
<td><a href="{$_route
('limits.index',
-$project->id)}"
class="btn_btn-
primary_btn-xs"
style="width:70px
"><i class="fa_fa
-eye"></i> Limits
</a></td>
<td><a href="{$_route
('subreports.
showLPTR',
-$subreport->id)-
}" class="btn_
btn-primary_btn-
xs" style="width:
70px"><i class="
fa_fa-eye"></i>
View </a></td>
<td><a href="{$_route
('limits.edit',
-$project->id)}"
class="btn_btn-
info_btn-xs"
style="width:70px
"><i class="fa_fa
-pencil"></i>
Edit </a></td>
<td>
{!! Form::open(['
method' => '
DELETE', '
route' => ['
subreports.
destroy',
$subreport->
id], '
onsubmit' =>
'return_
confirm("
Delete_liquid_
and_plastic_
limits_test_
report_
permanently
?")' ] !!}
{!! Form::submit('
Delete', ['
class' => '
btn_btn-
danger_btn-xs
', 'style' =>
'width:70px'
]) !!}
{!! Form::close()
!!}
</td>
@else
<td><a href="{$_route
('limits.create',
-$project->id)}"
class="btn_btn-
primary_btn-xs"
style="width:70px
"><i class="fa_fa
-plus_m-right-xs"
"></i> Create </a
"></td>
<td></td>
<td></td>
<td></td>
@endif
</tr>
<tr>
<td>Borehole Log</td>
@if($subreports->
contains('sb_key',
2))
<?php $subreport =
$subreports->
where('sb_key',
2)->first() ?>
<td></td>
<td><a href="{$_route
('subreports.
showBL',
-$subreport->id)-
}" class="btn_
btn-primary_btn-
xs" style="width:
70px"><i class="
fa_fa-eye"></i>
View </a></td>
<td><a href="{$_route
('subreports.
createBL',
-$project->id)-
}" class="btn_btn-
info_btn-xs"
style="width:70px
"><i class="fa_fa
-pencil"></i>
Update </a></td>
<td>
{!! Form::open(['
method' => '
DELETE', '
route' => ['
subreports.
destroy',
$subreport->
id], '
onsubmit' =>
'return_
confirm("
Delete_soil_

```

```

        profile_
        report_
        permanently
        ?"')') !!}
    {!! Form::submit('
    Delete', ['
    class' => '
    btn_btn-
    danger_btn-xs
    ', 'style' =>
    'width:70px'
    ]) !!}
    {!! Form::close ()
    !!}
</td>
@else
<td><a href="{{_route
('subreports.
createSP', _
$project->id) }}"
class="btn_btn-
primary_btn-xs"
style="width:70px
"><i class="fa_fa
-plus_m-right-xs"
"></i> Create </a
></td>
<td></td>
<td></td>
<td></td>
<td></td>
@endif
</tr>
</table>
@endif
<div class="ln_solid"></div>
</div>
</div>
<!-- /subreports -->
</div>
<div class="clearfix"></div>
@endsection
@section('js')
@parent
<!-- image cropping -->
<script src="{{_URL::asset('js/cropping/
cropper.min.js') }}"></script>
<script src="{{_URL::asset('js/cropping/main.
js') }}"></script>
@endsection

```

Listing 101: projects/viewActive.blade.php

```

@extends('layouts/master')
@section('title', 'Projects')
@section('content')
<div class="row">
    <div class="col-md-12">
        <div class="x_panel">
            <div class="x_title">
                <h2> Active Projects <small>
                View and manage current
                projects </small> </h2>
                <ul class="nav_navbar-right_
                panel_toolbox">
                    <li><a class="collapse-
                    link"><i class="fa_fa
                    -chevron-up"></i></a>
                </li>
                    <li class="dropdown">
                        <a href="#" class="
                        dropdown-toggle"
                        data-toggle="
                        dropdown" role="
                        button" aria-
                        expanded="false"
                        ><i class="fa_fa-
                        wrench"></i></a>
                        <ul class="dropdown-
                        menu" role="menu"
                        >
                            <li><a href="#">
                                Settings 1</a>
                            </li>
                            <li><a href="#">
                                Settings 2</a>
                            </li>
                        </ul>
                    </li>
                </ul>
            </div>
            <div class="clearfix"></div>
        </div>
    </div>

```

```

class
=
"
fa
-
fa
-
pencil
"></i>
</div>
</div>
</div>
<div class="
caption">
<p>
<center>{{
$proj->
proj_name
}}</center>
</p>
</div>
</div>
</div>
<div>
@empty
<center>
<h2>You have no active
projects.</h2>
</center>
</div>
</div>
<div class="ln_solid"></div>
</div>
</div>
</div>
@endsection

```

Listing 102: projects/viewActive.blade.php

```

@extends('layouts/master')
@section('title', 'Projects')
@section('content')
<div class="row">
<div class="col-md-12">
<div class="x_panel">
<div class="x_title">
<h2> Archived Projects <small>
View approved (past)
projects </small> </h2>
<ul class="nav navbar-right
panel_toolbox">
<li><a class="collapse-
link"><i class="fa fa-
chevron-up"></i></a>
</li>
<li class="dropdown">
<a href="#" class="
dropdown-toggle"
data-toggle="
dropdown" role="
button" aria-
expanded="false"
><i class="fa fa-
wrench"></i></a>
<ul class="dropdown-
menu" role="menu"
>
<li><a href="#">
Settings 1</a>
</li>
<li><a href="#">
Settings 2</a>
</li>
</ul>
</li>
</ul>
</div>
<div class="clearfix"></div>
<div class="x_content">
<div class="row">
@forelse($projects as
$proj)
<div class="col-md-55">
<div class="thumbnail">
<div class="image_
view_view-
first">

```

```


<div class="
mask">
<p>{{
$proj
->
proj_id
}}</p>
<div class
="
tools
-
tools
-
bottom
">
<a
href
=
"
{{
_
route
(
'
projects
.
show
',
_
$proj
->
proj_id
)
_
}}
">
</div>
</div>
</div>
<div class="
caption">
<p>
<center>{{
$proj->
proj_name
}}</center>
</p>
</div>
</div>
</div>
<div>
@empty
<center>
<h2>There are no
archived projects
.</h2>
</center>
</div>
</div>
<div class="ln_solid"></div>
</div>
</div>
</div>
@endsection

```

Listing 103: samples/create.blade.php

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
    <div class="col-md-12_col-sm-12_col-xs-12">
        <div class="x_panel">
            <div class="x_title">
                <h2> Add Sample </h2>
                <ul class="nav navbar-right panel_toolbox">
                    <li><a class="collapse-link" <i class="fa fa-chevron-up"></i></a></li>
                    <li class="dropdown">
                        <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-expanded="false"><i class="fa fa-wrench"></i></a>
                        <ul class="dropdown-menu" role="menu">
                            <li><a href="#">Settings 1</a></li>
                            <li><a href="#">Settings 2</a></li>
                        </ul>
                    </li>
                </ul>
            </div>
            <div class="clearfix"></div>
            <div class="x_content">
                {!! Form::open(['name' => 'store_form', 'method' => 'POST', 'route' => 'samples.store', 'class' => 'form-horizontal_form-label-left']) !!}
                <div class="container">
                    <div class="form-group">
                        <?php $uscs_ids = [] ?>
                        @for($i = 0; $i < sizeof($uscs); $i++)
                            <?php $uscs_ids[$uscs[$i]->id] = $uscs[$i]->id ?>
                        @endfor
                        {!! Form::label('uscs_id', 'Soil Classification_*', ['class' => 'control-label_col-md-3_col-sm-3_col-xs-12']) !!}
                        <div class="col-md-3_col-sm-6_col-xs-12_form-group">
                            {!! Form::select('uscs_id', $uscs_ids, old('uscs_id'), array('class' => 'btn btn-default', 'placeholder' => 'Choose soil classification...', 'required' => 'true')) !!}
                        </div>
                    </div>
                    <div class="form-group">
                        @if($previous_sample == null)
                            <?php $from_placeholder = "From";
                            $to_placeholder = "To";
                            $depth_text = "Depth_(m)_*";
                            $run_length_default = 0.30;
                        ?>
                        @else
                            <?php $from_placeholder = "From:_" . (float) $previous_sample->sample_to;
                            $to_placeholder = $previous_sample->sample_to + $previous_sample->sample_run_length;
                            $depth_text = "From:_" . $to_placeholder;
                            $run_length_default = $previous_sample->sample_run_length;
                        ?>
                    @endif
                    {!! Form::label('sample_from', $depth_text, ['class' => 'control-label_col-md-3_col-sm-3_col-xs-12']) !!}
                    <div class="col-md-3_col-sm-6_col-xs-12_form-group">
                        @if($previous_sample == null)
                            {!! Form::number('sample_from', old('sample_from'), ['class' => 'form-control', 'placeholder' => $from_placeholder, 'step' => 'any', 'required' => 'true']) !!}
                        @else
                            {!! Form::number('sample_from', old('sample_from'), ['class' => 'form-control', 'placeholder' => $from_placeholder, 'step' => 'any']) !!}
                        @endif
                    </div>
                    <div class="col-md-3_col-sm-6_col-xs-12_form-group">
                        @if($previous_sample == null)
                            {!! Form::number('sample_to', old('sample_to'), ['class' => 'form-control', 'placeholder' => $to_placeholder, 'step' => 'any', 'required' => 'true']) !!}
                        @else
                            {!! Form::number('sample_to', old('sample_to'), ['class' => 'form-control', 'placeholder' => $to_placeholder, 'step' => 'any', 'required' => 'true']) !!}
                        @endif
                    </div>
                </div>
            </div>
        </div>
    </div>
</section>
</php

```

```

=>
    $to_placeholder
    , 'step' => '
    any']) !!}
    @endif
</div>
</div>
<div class="form-group">
  {!! Form::label('
  sample_contract_num
  ', 'Contract_No.'
  , ['class' => '
  control-label_col
  -md-3_col-sm-3_col
  -xs-12']) !!}
  <div class="col-md-6_
  col-sm-6_col-xs
  -12">
    {!! Form::text('
    sample_contract_num
    ', old('
    sample_contract_num
    '), ['class'
    => 'form-
    control_col-
    md-7_col-xs
    -12']) !!}
  </div>
</div>
<div class="form-group">
  {!! Form::label('
  sample_job_num',
  'Job_No.', ['
  class' => '
  control-label_col
  -md-3_col-sm-3_col
  -xs-12']) !!}
  <div class="col-md-6_
  col-sm-6_col-xs
  -12">
    {!! Form::text('
    sample_job_num
    ', old('
    sample_job_num
    '), ['class'
    => 'form-
    control_col-
    md-7_col-xs
    -12']) !!}
  </div>
</div>
<br/><br/>
<div class="form-group">
  {!! Form::label('
  sample_color', '
  Sample_Color_*',
  ['class' => '
  control-label_col
  -md-3_col-sm-3_col
  -xs-12']) !!}
  <div class="col-md-6_
  col-sm-6_col-xs
  -12">
    {!! Form::text('
    sample_color',
    old('
    sample_color'
    ), ['class'
    => 'form-
    control_col-
    md-7_col-xs
    -12', '
    required' =>
    'true']) !!}
  </div>
</div>
<div class="form-group">
  {!! Form::label('
  sample_run_length
  ', 'Length_of_Run
  (m)', ['class'
  => 'control-label
  _col-md-3_col-sm
  -3_col-xs-12'])
  !!}
  <div class="col-md-6_
  col-sm-6_col-xs
  -12">
    {!! Form::number('
    sample_run_length
    ', old('
    sample_run_length
    '), ['class'
    => 'form-
    control_col-
    md-7_col-xs
    -12', 'step'
    => 'any', '
    placeholder'

```

```

=>
    $run_length_default
    ]) !!}
  </div>
</div>
<div class="form-group">
  {!! Form::label('
  sample_sample_length
  ', 'Length_of_Sample
  (m)_*', ['class'
  => '
  control-label_col
  -md-3_col-sm-3_col
  -xs-12']) !!}
  <div class="col-md-6_
  col-sm-6_col-xs
  -12">
    {!! Form::number('
    sample_sample_length
    ', old('
    sample_sample_length
    '), ['class'
    => 'form-
    control_col-
    md-7_col-xs
    -12', '
    required' =>
    'true', 'step'
    => 'any'])
    !!}
  </div>
</div>
<div class="form-group">
  {!! Form::label('
  sample_rec_percent
  ', 'Rec. (%)_*',
  ['class' => '
  control-label_col
  -md-3_col-sm-3_col
  -xs-12']) !!}
  <div class="col-md-6_
  col-sm-6_col-xs
  -12">
    {!! Form::number('
    sample_rec_percent
    ', old('
    sample_rec_percent
    '), ['class'
    => 'form-
    control_col-
    md-7_col-xs
    -12', '
    required' =>
    'true', 'step'
    => 'any'])
    !!}
  </div>
</div>
<div class="form-group">
  {!! Form::label('
  sample_num_of_blows
  ', 'No. of Blows_
  (SPT_N-Value)_*',
  ['class' => '
  control-label_col
  -md-3_col-sm-3_col
  -xs-12']) !!}
  <div class="col-md-6_
  col-sm-6_col-xs
  -12">
    {!! Form::number('
    sample_num_of_blows
    ', old('
    sample_num_of_blows
    '), ['class'
    => 'form-
    control_col-
    md-7_col-xs
    -12', '
    required' =>
    'true', 'step'
    => 'any'])
    !!}
  </div>
</div>
<div class="form-group">
  {!! Form::label('
  sample_type_of_sampling
  ', 'Type_of_Sampling',
  ['class' => '
  control-label_col
  -md-3_col-sm-3_col
  -xs-12']) !!}
  <div class="col-md-6_
  col-sm-6_col-xs
  -12">
    {!! Form::text('
    sample_type_of_sampling

```

```

', old('
sample_tester
'), ['class'
=> 'form-
control_col-
md-7_col-xs
-12']) !!}
</div>
</div>
<br/><br/>
<div class="form-group">
  {!! Form::label('
sample_nmc_percent
', 'NMC(%)', ['
class' => '
control_label_col-
md-3_col-sm-3_
col-xs-12']) !!}
  <div class="col-md-6_
col-sm-6_col-xs
-12">
    {!! Form::number('
sample_nmc_percent
', old('
sample_nmc_percent
'), ['class'
=> 'form-
control_col-
md-7_col-xs
-12', 'step'
=> 'any'])
    !!}
  </div>
</div>
<div class="form-group">
  {!! Form::label('
sample_liquid_limit
', 'Liquid_Limit_
*', ['class' => '
control_label_col-
md-3_col-sm-3_
col-xs-12']) !!}
  <div class="col-md-6_
col-sm-6_col-xs
-12">
    {!! Form::number('
sample_liquid_limit
', old('
sample_liquid_limit
'), ['class'
=> 'form-
control_col-
md-7_col-xs
-12', '
required' =>
'true', 'step'
=> 'any'])
    !!}
  </div>
</div>
<div class="form-group">
  {!! Form::label('
sample_plastic_limit
', 'Plastic_Limit_
*', ['class' => '
control_label_
col-md-3_col-sm-3_
_col-xs-12']) !!}
  <div class="col-md-6_
col-sm-6_col-xs
-12">
    {!! Form::number('
sample_plastic_limit
', old('
sample_plastic_limit
'), ['class'
=> 'form-
control_col-
md-7_col-xs
-12', '
required' =>
'true', 'step'
=> 'any'])
    !!}
  </div>
</div>
<div class="form-group">
  {!! Form::label('
sample_plasticity_index
', 'Plasticity_
Index_*', ['class'
=> 'control_
label_col-md-3_
col-sm-3_col-xs
-12']) !!}
  <div class="col-md-6_
col-sm-6_col-xs
-12">
    {!! Form::number('
sample_plasticity_index
', old('
sample_plasticity_index
'), ['class'
=> 'form-
control_col-
md-7_col-xs
-12', '
required' =>
'true', 'step'
=> 'any'])
    !!}
  </div>
</div>
<div class="form-group">
  {!! Form::label('
sample_plasticity_remark
', 'Plasticity_
Remark', ['class'
=> 'control_
label_col-md-3_
col-sm-3_col-xs
-12']) !!}
  <div class="col-md-6_
col-sm-6_col-xs
-12">
    {!! Form::text('
sample_plasticity_remark
', old('
sample_plasticity_remark
'), ['class'
=> 'form-
control_col-
md-7_col-xs
-12']) !!}
  </div>
</div>
<div class="form-group">
  {!! Form::label('
sample_tester', '
Tester', ['class'
=> 'control_
label_col-md-3_
col-sm-3_col-xs
-12']) !!}
  <div class="col-md-6_
col-sm-6_col-xs
-12">
    {!! Form::text('
sample_tester
', old('
sample_tester
'), ['class'
=> 'form-
control_col-
md-7_col-xs
-12']) !!}
  </div>
</div>
<div class="form-group">
  {!! Form::label('
sample_other_results
', 'Other_Test_
Results', ['class'
=> 'control_
label_col-md-3_
col-sm-3_col-xs
-12']) !!}
  <div class="col-md-6_
col-sm-6_col-xs
-12">
    {!! Form::text('
sample_other_results
', old('
sample_other_results
'), ['class'
=> 'form-
control_col-
md-7_col-xs
-12']) !!}
  </div>
</div>
<div class="form-group">
  {!! Form::label('
sample_test_date',
'Test_Date', ['
class' => '
control_label_col-
md-3_col-sm-3_
col-xs-12']) !!}
  <div class="col-md-6_
col-sm-6_col-xs
-12">
    {!! Form::date('
sample_test_date
', old('
sample_test_date
'), ['class'
=> 'form-

```



```

        ), $sample->
        sample_contract_num
        ) ?>
    {!! Form::text('
    sample_contract_num
    ',
    $sample_contract_num
    , ['class' =>
    'form-
    control_col-
    md-7_col-xs
    -12']) !!}
</div>
</div>
<div class="form-group">
    {!! Form::label('
    sample_job_num ',
    'Job_No.', ['
    class' => '
    control_label_col
    -md-3_col-sm-3_
    col-xs-12']) !!}
    <div class="col-md-6_
    col-sm-6_col-xs
    -12">
        <?php
            $sample_job_num
            = Helper::
            getValue(old(
            ,
            sample_job_num
            ), $sample->
            sample_job_num
            ) ?>
        {!! Form::text('
        sample_job_num
        ',
        $sample_job_num
        , ['class' =>
        'form-
        control_col-
        md-7_col-xs
        -12']) !!}
    </div>
</div>
<br/><br/>
<div class="form-group">
    {!! Form::label('
    sample_color ',
    'Sample_Color_*',
    ['class' => '
    control_label_col
    -md-3_col-sm-3_
    col-xs-12']) !!}
    <div class="col-md-6_
    col-sm-6_col-xs
    -12">
        <?php
            $sample_color
            = Helper::
            getValue(old(
            'sample_color
            '), $sample->
            sample_color)
            ?>
        {!! Form::text('
        sample_color ',
        ,
        $sample_color
        , ['class' =>
        'form-
        control_col-
        md-7_col-xs
        -12', '
        required' =>
        'true']) !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label('
    sample_run_length
    ', 'Length_of_Run
    -(m)_*', ['class'
    => 'control-
    label_col-md-3_
    col-sm-3_col-xs
    -12']) !!}
    <div class="col-md-6_
    col-sm-6_col-xs
    -12">
        <?php
            $sample_run_length
            = Helper::
            getValue(old(
            ,
            sample_run_length
            ), $sample->
            sample_run_length
            ) ?>
        {!! Form::number('
        sample_run_length
        ',
        $sample_run_length
        , ['class' =>
        'form-
        control_col-
        md-7_col-xs
        -12', 'step'
        => '0.30', '
        required' =>
        'true']) !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label('
    sample_sample_length
    ', 'Length_of_
    Sample_(m)_*', ['
    class' => '
    control_label_col
    -md-3_col-sm-3_
    col-xs-12']) !!}
    <div class="col-md-6_
    col-sm-6_col-xs
    -12">
        <?php
            $sample_sample_length
            = Helper::
            getValue(old(
            ,
            sample_sample_length
            ), $sample->
            sample_sample_length
            ) ?>
        {!! Form::number('
        sample_sample_length
        ',
        $sample_sample_length
        , ['class' =>
        'form-
        control_col-
        md-7_col-xs
        -12', '
        required' =>
        'true', 'step'
        => 'any'])
        !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label('
    sample_rec_percent
    ', 'Rec._(%)_*',
    ['class' => '
    control_label_col
    -md-3_col-sm-3_
    col-xs-12']) !!}
    <div class="col-md-6_
    col-sm-6_col-xs
    -12">
        <?php
            $sample_rec_percent
            = Helper::
            getValue(old(
            ,
            sample_rec_percent
            ), $sample->
            sample_rec_percent
            ) ?>
        {!! Form::number('
        sample_rec_percent
        ',
        $sample_rec_percent
        , ['class' =>
        'form-
        control_col-
        md-7_col-xs
        -12', '
        required' =>
        'true', 'step'
        => 'any'])
        !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label('
    sample_num_of_blows
    ', 'No._of_Blows_
    (SPT-N-Value)_*',
    ['class' => '
    control_label_col
    -md-3_col-sm-3_
    col-xs-12']) !!}
    <div class="col-md-6_
    col-sm-6_col-xs
    -12">
        <?php
            $sample_num_of_blows
            = Helper::
            getValue(old(
            ,
            sample_num_of_blows
            ), $sample->
            sample_num_of_blows
            ) ?>
        {!! Form::number('
        sample_num_of_blows
        ',
        $sample_num_of_blows
        , ['class' =>
        'form-
        control_col-
        md-7_col-xs
        -12', 'step'
        => '0.5', '
        required' =>
        'true'])
        !!}
    </div>
</div>

```

```

<?php
    $sample_num_of_blows
    = Helper::
    getValue(old(
        ,
        sample_num_of_blows
        ), $sample->
        sample_num_of_blows
        ) ?>
    {!! Form::number('
    sample_num_of_blows
    ',
    $sample_num_of_blows
    , ['class' =>
    'form-
    control_col-
    md-7_col-xs
    -12', '
    required' =>
    'true', 'step
    ' => 'any'])
    !!}
</div>
</div>
<div class="form-group">
    {!! Form::label('
    sample_type_of_sampl
    ing', 'Type_of_
    Sampling', ['
    class' => ,
    control_label_col
    -md-3_col-sm-3_
    col-xs-12']) !!}
    <div class="col-md-6_
    col-sm-6_col-xs
    -12">
        <?php
            $sample_type_of_sampl
            ing
            = Helper::
            getValue(old(
                ,
                sample_type_of_sampl
                ing
                ), $sample->
                sample_type_of_sampl
                ing
                ) ?>
            {!! Form::text('
            sample_type_of_sampl
            ing
            ',
            $sample_type_of_sampl
            ing
            , ['class' =>
            'form-
            control_col-
            md-7_col-xs
            -12']) !!}
        </div>
    </div>
<br/><br/>
<div class="form-group">
    {!! Form::label('
    sample_nmc_percent
    ', 'NMC(%)', ['
    class' => ,
    control_label_col
    -md-3_col-sm-3_
    col-xs-12']) !!}
    <div class="col-md-6_
    col-sm-6_col-xs
    -12">
        <?php
            $sample_nmc_percent
            = Helper::
            getValue(old(
                ,
                sample_nmc_percent
                ), $sample->
                sample_nmc_percent
                ) ?>
            {!! Form::number('
            sample_nmc_percent
            ',
            $sample_nmc_percent
            , ['class' =>
            'form-
            control_col-
            md-7_col-xs
            -12', 'step
            ' => 'any'])
            !!}
        </div>
    </div>
<div class="form-group">
    {!! Form::label('
    sample_liquid_limit
    ', 'Liquid_Limit_
    *', ['class' => ,
    control_label_col
    -md-3_col-sm-3_
    col-xs-12']) !!}
<div class="col-md-6_
col-sm-6_col-xs
-12">
    <?php
        $sample_liquid_limit
        = Helper::
        getValue(old(
            ,
            sample_liquid_limit
            ), $sample->
            sample_liquid_limit
            ) ?>
        {!! Form::number('
        sample_liquid_limit
        ',
        $sample_liquid_limit
        , ['class' =>
        'form-
        control_col-
        md-7_col-xs
        -12', '
        required' =>
        'true', 'step
        ' => 'any'])
        !!}
    </div>
</div>
<div class="form-group">
    {!! Form::label('
    sample_plastic_limit
    ', 'Plastic_Limit
    *', ['class' => ,
    control_label_
    col-md-3_col-sm-3
    col-xs-12']) !!}
    <div class="col-md-6_
    col-sm-6_col-xs
    -12">
        <?php
            $sample_plastic_limit
            = Helper::
            getValue(old(
                ,
                sample_plastic_limit
                ), $sample->
                sample_plastic_limit
                ) ?>
            {!! Form::number('
            sample_plastic_limit
            ',
            $sample_plastic_limit
            , ['class' =>
            'form-
            control_col-
            md-7_col-xs
            -12', '
            required' =>
            'true', 'step
            ' => 'any'])
            !!}
        </div>
    </div>
<div class="form-group">
    {!! Form::label('
    sample_plasticity_index
    ', 'Plasticity_
    Index_*', ['class
    ' => ,
    control_
    label_col-md-3_
    col-sm-3_col-xs
    -12']) !!}
    <div class="col-md-6_
    col-sm-6_col-xs
    -12">
        <?php
            $sample_plasticity_index
            = Helper::
            getValue(old(
                ,
                sample_plasticity_index
                ), $sample->
                sample_plasticity_index
                ) ?>
            {!! Form::number('
            sample_plasticity_index
            ',
            $sample_plasticity_index
            , ['class' =>
            'form-
            control_col-
            md-7_col-xs
            -12', '
            required' =>
            'true', 'step
            ' => 'any'])
            !!}
        </div>
    </div>
<div class="form-group">
    {!! Form::label('

```



```

        </li>
        <li><a href="#">
            Settings 2</a>
        </li>
    </ul>
</div>
<div class="clearfix"></div>
</div>
<div class="x_content">
    <div id="alerts"></div>
    <div class="btn-toolbar-editor"
        data-role="editor-
        toolbar" data-target="#"
        editor">
        <div class="btn-group">
            <a class="btn_dropdown-
            toggle" data-
            toggle="dropdown"
            title="Font_Size"
            ><i class="icon-
            text-height"></i>
            &nbsp;<b class="
            caret"></b></a>
            <ul class="dropdown-
            menu">
                <li>
                    <a data-edit="
                    fontSize_
                    5">
                        <p style="
                        font-
                        size
                        :17px
                        ">
                            Huge
                        </p>
                    </a>
                </li>
                <li>
                    <a data-edit="
                    fontSize_
                    3">
                        <p style="
                        font-
                        size
                        :14px
                        ">
                            Normal
                        </p>
                    </a>
                </li>
                <li>
                    <a data-edit="
                    fontSize_
                    1">
                        <p style="
                        font-
                        size
                        :11px
                        ">
                            Small
                        </p>
                    </a>
                </li>
            </ul>
        </div>
        <div class="btn-group">
            <a class="btn" data-
            edit="bold" title
            =" Bold_( Ctrl/ Cmd+
            B)"><i class="
            icon-bold"></i></a>
            <a class="btn" data-
            edit="italic"
            title=" Italic_(
            Ctrl/ Cmd+ I)"><i
            class="icon-
            italic"></i></a>
            <a class="btn" data-
            edit="
            strikethrough"
            title="
            Strikethrough"><i
            class="icon-
            strikethrough"></i></a>
            <a class="btn" data-
            edit="underline"
            title=" Underline_(
            Ctrl/ Cmd+ U)"><i
            class="icon-
            underline"></i></a>
        </div>
    </div class="btn-group">
        <a class="btn" data-
        edit="
        insertunorderedlist"
        title=" Bullet_
        list"><i class="
        icon-list-ul"></i>
        </a>
        <a class="btn" data-
        edit="
        insertorderedlist"
        title=" Number_
        list"><i class="
        icon-list-ol"></i>
        </a>
        <a class="btn" data-
        edit="outdent"
        title=" Reduce_
        indent_( Shift+ Tab)"><i class="icon-
        indent-left"></i>
        </a>
        <a class="btn" data-
        edit="indent"
        title=" Indent_(
        Tab)"><i class="
        icon-indent-right"
        ></i></a>
    </div>
    <div class="btn-group">
        <a class="btn" data-
        edit="justifyleft"
        title=" Align_
        Left_( Ctrl/ Cmd+ L)"
        ><i class="icon-
        align-left"></i>
        </a>
        <a class="btn" data-
        edit="
        justifycenter"
        title=" Center_(
        Ctrl/ Cmd+ E)"><i
        class="icon-align-
        center"></i></a>
        <a class="btn" data-
        edit="
        justifyright"
        title=" Align_
        Right_( Ctrl/ Cmd+ R)"
        ><i class="icon-
        align-right"></i>
        </a>
        <a class="btn" data-
        edit="justifyfull"
        title=" Justify_(
        Ctrl/ Cmd+ J)"><i
        class="icon-align-
        justify"></i></a>
    </div>
    <div class="btn-group">
        <a class="btn" title="
        Insert_picture_(
        or_ just_drag_&_
        drop)" id="
        pictureBtn"><i
        class="icon-
        picture"></i></a>
        <input type="file"
        data-role="magic-
        overlay" data-
        target="#"
        pictureBtn" data-
        edit="insertImage"
        />
    </div>
</div>
</div>
{!! Form::open(['name' => '
store_form', 'method' => '
POST', 'route' => '
sections.store', '
onsubmit' => 'document.
store_form.section_text.
value._.$("#editor").html
()')] !!}
<?php
$text = Helper::
getSectionTemplate(
    $section_name ,
    $project->id);
$section_key = Helper::
getSectionKey(
    $section_name);
?>
<textarea name="section_text"
id="section_text" style="
display:none;">

```

```

    <div id="editor"
    contenteditable="true">
    {!! $text !!}
    </div>
    <br />
    <div class="ln_solid"></div>
    {!! Form::hidden('proj_id',
    $project->id) !!}
    {!! Form::hidden('section_key',
    $section_key) !!}
    {!! link_to(URL::previous(), '
    Back', array('class' => '
    btn btn-default')) !!}
    {!! Form::submit('Submit', ['
    class' => 'btn btn-
    primary', 'style' => '
    float:right']) !!}
    {!! Form::close() !!}
    </div>
    </div>
    </div>
    @endsection
    @section('js')
    @parent
    <!-- tags -->
    <script src="{{_URL::asset('js/tags/jquery.
    tagsinput.min.js')}}"></script>
    <!-- switchery -->
    <script src="{{_URL::asset('js/switchery/
    switchery.min.js')}}"></script>
    <!-- daterangepicker -->
    <script src="{{_URL::asset('js/moment.min2.js
   ')}}" type="text/javascript"></script>
    <script src="{{_URL::asset('js/daterangepicker/
    daterangepicker.js')}}" type="text/
    javascript"></script>
    <!-- richtext editor -->
    <script src="{{_URL::asset('js/editor/
    bootstrap-wysiwyg.js')}}"></script>
    <script src="{{_URL::asset('js/editor/external
    /jquery.hotkeys.js')}}"></script>
    <script src="{{_URL::asset('js/editor/external
    /google-code-prettify/prettify.js')}}"
    ></script>
    <!-- select2 -->
    <script src="{{_URL::asset('js/select/select2.
    full.js')}}"></script>
    <!-- form validation -->
    <script src="{{_URL::asset('js/parsley/parsley
    .min.js')}}" type="text/javascript" ></
    script>
    <!-- textarea resize -->
    <script src="{{_URL::asset('js/textarea/
    autosize.min.js')}}"></script>
    <!-- Autocomplete -->
    <script src="{{_URL::asset('js/autocomplete/
    countries.js')}}" type="text/javascript"
    ></script>
    <script src="{{_URL::asset('js/autocomplete/
    jquery.autocomplete.js')}}"></script>
    <!-- form validation -->
    <script type="text/javascript">
    $(document).ready(function () {
    $.listen('parsley:field:validate',
    function () {
    validateFront();
    });
    $('#demo-form-.btn').on('click',
    function () {
    $('#demo-form').parsley().validate
    ();
    validateFront();
    });
    var validateFront = function () {
    if (true === $('#demo-form').
    parsley().isValid()) {
    $('.bs-callout-info').
    removeClass('hidden');
    $('.bs-callout-warning').
    addClass('hidden');
    } else {
    $('.bs-callout-info').addClass
    ('hidden');
    $('.bs-callout-warning').
    removeClass('hidden');
    }
    };
    });
    $(document).ready(function () {
    $.listen('parsley:field:validate',
    function () {
    validateFront();
    });
    $('#demo-form2-.btn').on('click',
    function () {
    $('#demo-form2').parsley().

```

```

    validate();
    validateFront();
    });
    var validateFront = function () {
    if (true === $('#demo-form2').
    parsley().isValid()) {
    $('.bs-callout-info').
    removeClass('hidden');
    $('.bs-callout-warning').
    addClass('hidden');
    } else {
    $('.bs-callout-info').addClass
    ('hidden');
    $('.bs-callout-warning').
    removeClass('hidden');
    }
    };
    try {
    hljs.initHighlightingOnLoad();
    } catch (err) {}
    </script>
    <!-- /form validation -->
    <!-- editor -->
    <script>
    $(document).ready(function () {
    $('#cxc').click(function () {
    $('#section_text').val($('#editor')
    ).html();
    });
    });
    $(function () {
    function initToolbarBootstrapBindings
    () {
    var fonts = ['Arial', 'Arial_Black',
    'Calibri', 'Cambria', '
    Comic_Sans_MS', 'Courier',
    'Courier_New', 'Helvetica', 'Impact',
    'Lucida_Grande', 'Lucida_Sans', '
    Sans', 'Serif', 'Tahoma', 'Times',
    'Times_New_Roman', 'Verdana'];
    fontTarget = $('[title=Font]')
    .siblings('.dropdown-menu
    ');
    $.each(fonts, function (idx,
    fontName) {
    fontTarget.append('<li><a_
    data-edit="fontName_' +
    fontName + '"_style="font
    -family:\'' + fontName +
    \''>' + fontName + '</a
    ></li>');
    });
    $('a[title]').tooltip({
    container: 'body'
    });
    $('.dropdown-menu_input').click(
    function () {
    return false;
    })
    .change(function () {
    $(this).parent('.dropdown-
    menu').siblings('.
    dropdown-toggle').
    dropdown('toggle');
    })
    .keydown('esc', function () {
    this.value = '';
    $(this).change();
    });
    $('[data-role=magic-overlay]').
    each(function () {
    var overlay = $(this),
    target = $(overlay.data('
    target'));
    overlay.css('opacity', 0).css(
    'position', 'absolute').
    offset(target.offset()).
    width(target.outerWidth()
    ).height(target.
    outerHeight());
    });
    if ("onwebkitspeechchange" in
    document.createElement("input
    ")) {
    var editorOffset = $('#editor')
    .offset();
    $('#voiceBtn').css('position',
    'absolute').offset({
    top: editorOffset.top,
    left: editorOffset.left +
    $('#editor').
    innerWidth() - 35
    });
    } else {
    $('#voiceBtn').hide();

```

```

    }
};

function showErrorAlert(reason, detail)
{
    var msg = '';
    if (reason === 'unsupported-file-type') {
        msg = "Unsupported_format_" +
            detail;
    } else {
        console.log("error_uploading_file", reason, detail);
    }
    $('<div_class="alert"><button_type="button" _class="close" _data-dismiss="alert">&times</button>' +
        '<strong>File_upload_error</strong>' + msg + '</div>')
        .prependTo('#alerts');
};

initToolbarBootstrapBindings();
$('#editor').wysiwyg({
    fileUploadError: showErrorAlert
});

window.prettyPrint && prettyPrint();
});
</script>
<!-- /editor -->
@endsection

```

Listing 106: sections/edit.blade.php

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('stylesheets')
@parent
<!-- editor -->
<link href="{{_URL::asset('css/font-awesome.css')}}" rel="stylesheet">
<link href="{{_URL::asset('css/editor/external/google-code-prettify/prettify.css')}}" rel="stylesheet">
<link href="{{_URL::asset('css/editor/index.css')}}" rel="stylesheet">
<!-- select2 -->
<link href="{{_URL::asset('css/select/select2.min.css')}}" rel="stylesheet">
<!-- switchery -->
<link href="{{_URL::asset('css/switchery/switchery.min.css')}}" rel="stylesheet" />
@endsection
@section('content')
<div class="row">
    <div class="col-md-12">
        <div class="x_panel">
            <div class="x_title">
                <h2>Edit {{ Helper::getSectionName($section->section_key) }}</h2>
            <ul class="nav navbar-right panel_toolbox">
                <li><a class="collapse-link"><i class="fa fa-chevron-up"></i></a>
            </li>
                <li class="dropdown">
                    <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-expanded="false"><i class="fa fa-wrench"></i></a>
                    <ul class="dropdown-menu" role="menu">
                        <li><a href="#">Settings 1</a>
                    </li>
                        <li><a href="#">Settings 2</a>
                    </li>
                    </ul>
                </li>
            </ul>
            <div class="clearfix"></div>
            <div class="x_content">
                <div id="alerts"></div>
                <div class="btn-toolbar-editor" data-role="editor-

```

```

toolbar" data-target="#
editor">
<div class="btn-group">
    <a class="btn_dropdown-toggle" data-toggle="dropdown" title="Font_Size"><i class="icon-text-height"></i>&nbsp;<b class="caret"></b></a>
    <ul class="dropdown-menu">
        <li>
            <a data-edit="fontSize_5">
                <p style="font-size:17px">
                    Huge
                </p>
            </a>
        </li>
        <li>
            <a data-edit="fontSize_3">
                <p style="font-size:14px">
                    Normal
                </p>
            </a>
        </li>
        <li>
            <a data-edit="fontSize_1">
                <p style="font-size:11px">
                    Small
                </p>
            </a>
        </li>
    </ul>
</div>
<div class="btn-group">
    <a class="btn" data-edit="bold" title="Bold_(Ctrl/Cmd+B)"><i class="icon-bold"></i></a>
    <a class="btn" data-edit="italic" title="Italic_(Ctrl/Cmd+I)"><i class="icon-italic"></i></a>
    <a class="btn" data-edit="strikethrough" title="Strikethrough"><i class="icon-strikethrough"></i></a>
    <a class="btn" data-edit="underline" title="Underline_(Ctrl/Cmd+U)"><i class="icon-underline"></i></a>
</div>
<div class="btn-group">
    <a class="btn" data-edit="insertunorderedlist" title="Bullet_list"><i class="icon-list-ul"></i></a>
    <a class="btn" data-edit="insertorderedlist" title="Number_list"><i class="icon-list-ol"></i></a>
    <a class="btn" data-

```

```

edit="outdent"
title="Reduce_
indent_(Shift+Tab)"><i class="icon
-indent-left"></i>
</a>
<a class="btn" data-
edit="indent"
title="Indent_(
Tab)"><i class="
icon-indent-right"
"></i></a>
</div>
<div class="btn-group">
<a class="btn" data-
edit="justifyleft"
title="Align_
Left_(Ctrl/Cmd+L)"
"><i class="icon-
align-left"></i>
"></a>
<a class="btn" data-
edit="
justifycenter"
title="Center_(
Ctrl/Cmd+E)"><i
class="icon-align-
center"></i></a>
<a class="btn" data-
edit="
justifyright"
title="Align_
Right_(Ctrl/Cmd+R)"
"><i class="icon-
-align-right"></i>
"></a>
<a class="btn" data-
edit="justifyfull"
title="Justify_(
Ctrl/Cmd+J)"><i
class="icon-align-
-justify"></i></a>
">
</div>
<div class="btn-group">
<a class="btn" title="
Insert_picture_(
or_just_drag_&_
drop)" id="
pictureBtn"><i
class="icon-
picture"></i></a>
<input type="file"
data-role="magic-
overlay" data-
target="#"
pictureBtn" data-
edit="insertImage"
"/>
</div>
</div>
{!! Form::open(['name' => '
update_form', 'method' =>
'PUT', 'route' => [ '
sections.update',
$section->id], 'onsubmit'
=> 'document.update_form
.section_text.value_=$
("#editor").html()']) !!}
{!! Form::hidden('proj_id',
$project->id) !!}
<?php $text = Helper::
getSectionText($section->
id) ?>
<textarea name="section_text"
id="section_text" style="
display:none;">
</
textarea
>
<div id="editor">
{!! $text[0]->section_text
!!}
</div>
<br />
<div class="ln_solid"></div>
{!! link_to(URL::previous(), '
Back', array('class' => '
btn_btn-default')) !!}
{!! Form::submit('Submit', [ '
class' => 'btn_btn-
primary', 'style' => '
float:right']) !!}
{!! Form::close() !!}
</div>
</div>
</div>
</div>
@endsection
@section('js')
@parent
<!-- tags -->
<script src="{{_URL::asset('js/tags/jquery.
tagsinput.min.js')}}"></script>
<!-- switchery -->
<script src="{{_URL::asset('js/switchery/
switchery.min.js')}}"></script>
<!-- daterangepicker -->
<script src="{{_URL::asset('js/moment.min2.js
')}}" type="text/javascript"></script>
<script src="{{_URL::asset('js/datepicker/
daterangepicker.js')}}" type="text/
javascript"></script>
<!-- richtext editor -->
<script src="{{_URL::asset('js/editor/
bootstrap-wysiwyg.js')}}"></script>
<script src="{{_URL::asset('js/editor/external
/jquery.hotkeys.js')}}"></script>
<script src="{{_URL::asset('js/editor/external
/google-code-prettify/prettify.js')}}"
"></script>
<!-- select2 -->
<script src="{{_URL::asset('js/select/select2.
full.js')}}"></script>
<!-- form validation -->
<script src="{{_URL::asset('js/parsley/parsley
.min.js')}}" type="text/javascript" ></
script>
<!-- textarea resize -->
<script src="{{_URL::asset('js/textarea/
autosize.min.js')}}"></script>
<!-- Autocomplete -->
<script src="{{_URL::asset('js/autocomplete/
countries.js')}}" type="text/javascript"
"></script>
<script src="{{_URL::asset('js/autocomplete/
jquery.autocomplete.js')}}"></script>
<!-- form validation -->
<script type="text/javascript">
$(document).ready(function () {
$.listen('parsley:field:validate',
function () {
validateFront();
});
$('#demo-form_btn').on('click',
function () {
$('#demo-form').parsley().validate
();
validateFront();
});
var validateFront = function () {
if (true === $('#demo-form').
parsley().isValid()) {
$('.bs-callout-info').
removeClass('hidden');
$('.bs-callout-warning').
addClass('hidden');
} else {
$('.bs-callout-info').addClass
('hidden');
$('.bs-callout-warning').
removeClass('hidden');
}
});
});
$(document).ready(function () {
$.listen('parsley:field:validate',
function () {
validateFront();
});
$('#demo-form2_btn').on('click',
function () {
$('#demo-form2').parsley().
validate();
validateFront();
});
var validateFront = function () {
if (true === $('#demo-form2').
parsley().isValid()) {
$('.bs-callout-info').
removeClass('hidden');
$('.bs-callout-warning').
addClass('hidden');
} else {
$('.bs-callout-info').addClass
('hidden');
$('.bs-callout-warning').
removeClass('hidden');
}
});
});
try {
hljs.initHighlightingOnLoad();
} catch (err) {}
</script>
<!-- /form validation -->

```

```

<!-- editor -->
<script>
$(document).ready(function () {
    $('#xexc').click(function () {
        $('#section_text').val($('#editor')
        ).html());
    });
});

$(function () {
    function initToolbarBootstrapBindings
    () {
        var fonts = ['Arial', 'Arial_Black',
            'Calibri', 'Cambria',
            'Comic_Sans_MS', 'Courier',
            'Courier_New', 'Helvetica', 'Impact',
            'Lucida_Grande', 'Lucida_Sans',
            'Sans', 'Serif', 'Tahoma', 'Times',
            'Times_New_Roman', 'Verdana'];
        fontTarget = $(''[title=Font]')
            .siblings('.dropdown-menu');
        $.each(fonts, function (idx,
            fontName) {
            fontTarget.append($('- <a
            data-edit=fontName_ +
            fontName + "_style=font
            -family:\'' + fontName +
            '\''>' + fontName + '</a
            ></li>'));
        });
        $('a[title]').tooltip({
            container: 'body'
        });
        $(''.dropdown-menu_input').click(
            function () {
                return false;
            })
            .change(function () {
                $(this).parent('.dropdown-
                menu').siblings('.
                dropdown-toggle').
                dropdown('toggle');
            })
            .keydown('esc', function () {
                this.value = '';
                $(this).change();
            });
        $('[data-role=magic-overlay]').
            each(function () {
                var overlay = $(this),
                    target = $(overlay.data('
                    target'));
                overlay.css('opacity', 0).css(
                    'position', 'absolute').
                    offset(target.offset()).
                    width(target.outerWidth())
                    .height(target.
                    outerHeight());
            });
        if ("onwebkitspeechchange" in
            document.createElement("input
            ")) {
            var editorOffset = $('#editor')
                .offset();
            $('voiceBtn').css('position',
                'absolute').offset({
                top: editorOffset.top,
                left: editorOffset.left +
                $('#editor').
                innerWidth() - 35
            });
        } else {
            $('voiceBtn').hide();
        }
    };

    function showErrorAlert(reason, detail)
    {
        var msg = '';
        if (reason === 'unsupported-file-
        type') {
            msg = "Unsupported_format_" +
            detail;
        } else {
            console.log("error_uploading_
            file", reason, detail);
        }
        $('div_class=alert"><button_
        type=button">close">
        data-dismiss=alert">&times
        ;</button>' +
        '<strong>File_upload_error</
        strong>' + msg + '</div
        >').prependTo('#alerts');
    };
    initToolbarBootstrapBindings();
}

```

```

$('#editor').wysiwyg({
    fileUploadError: showErrorAlert
});
window.prettyPrint && prettyPrint();
});
</script>
<!-- /editor -->
@endsection

Listing 107: sections/show.blade.php

@extends('layouts/master')
@section('title', $project->proj-name)
@section('content')
<?php $section_name = Helper::getSectionName(
    $section->section-key) ?>
<div class=row">
    <div class=col-md-12">
        <div class=x_panel">
            <div class=x_title">
                <h2>{{ $section_name }}</h2>
                <ul class=nav_navbar-right_
                panel_toolbox">
                    <li><a class=collapse_
                    link"><i class=fa_fa_
                    -chevron-up"></i></a>
                </li>
                    <li class=dropdown">
                        <a href=#" class=
                        dropdown-toggle"
                        data-toggle=
                        dropdown" role=
                        button" aria-
                        expanded=false"
                        ><i class=fa_fa_
                        wrench"></i></a>
                        <ul class=dropdown-
                        menu" role=menu"
                        >
                            <li><a href=#">
                                Settings 1</a>
                            </li>
                            <li><a href=#">
                                Settings 2</a>
                            </li>
                        </ul>
                    </li>
                </ul>
            </div>
            <div class=clearfix"></div>
            <div class=x_content">
                <?php $text = Helper::
                getSectionText($section->
                id) ?>
                {!! $text[0]->section_text !!}
                <div class=ln_solid"></div>
                {!! Form::open(['method' => '
                DELETE', 'route' => ['
                deeps.destroy', $project
                ->id], 'onsubmit' => '
                return_confirm("Delete_
                SBC_data_for_deep_
                foundation_permanently?")
                ']) !!}
                {!! Form::submit('Delete', ['
                class' => 'btn_btn-danger
                ']) !!}
                <a href="{{_route('sections.
                edit', _$section->id)_}}"
                class=btn_btn-info"
                style=float:right"><i
                class=fa_fa-edit_m-right
                -xs"></i> Edit </a>
                {!! Form::close() !!}
            </div>
        </div>
    </div>
@endsection

Listing 108: shallows/calcu-
late.blade.php

@include('includes/clearform')

<!-- tooltip -->
<script>
$(document).ready(function () {
    $('[data-toggle=tooltip]').tooltip()
};
</script>

```



```

<!-- /tooltip -->

<!-- calculate -->
<script type="text/javascript">
    var num_of_boreholes = <?php echo
        sizeof($boreholes) ?>;
    var num_of_samples;
    var soil_type, num_of_blows,
        c_denominator, c;
    var df, nc, ngamma, nq, b, gammad,
        gamma2, fs, qult, q;

    function calculate() {
        for(i = 1; i <= num_of_boreholes;
            i++) {
            num_of_samples = $(''
                shallow_depth_' + i)
                .length;
            console.log(num_of_samples);
            for(j = 1; j <= num_of_samples
                ; j++) {

                // Calculate shallow_c
                soil_type = $(''
                    shallow_soil_' + i +
                    '_' + j + '_option:
                    selected').text();
                num_of_blows = $(''
                    shallow_num_of_blows_'
                    + i + '_' + j).val
                    ();

                switch(soil_type) {
                    case 'clay':
                        c_denominator =
                            4; break;
                    case 'silt':
                        c_denominator =
                            5; break;
                    default: c_denominator
                        = null; break;
                }

                c = (num_of_blows*48)/
                    c_denominator;
                $(''
                    '#shallow_c_' + i + '_'
                    + j).val(parseFloat(c
                    .toFixed(4)));

                // Calculate shallow_qult
                df = $(''
                    '#shallow_depth_' +
                    i + '_' + j).val();
                nc = $(''
                    '#shallow_Nc_' + i
                    ).val();
                ngamma = $(''
                    '#shallow_Ngamma_' + i
                    ).val();
                nq = $(''
                    '#shallow_Nq_' + i
                    ).val();
                b = $(''
                    '#shallow_B_' + i)
                    .val();
                gammad = $(''
                    '#shallow-gamma1_' + i
                    ).val();
                gamma2 = $(''
                    '#shallow-gamma2_' + i
                    ).val();
                fs = $(''
                    '#shallow_fs_' + i
                    ).val();

                qult = c*nc + 0.5*b*gammad
                    *ngamma + gamma2*df*
                    nq;
                $(''
                    '#shallow_qult_' + i + '_'
                    + j).val(
                    parseFloat(qult
                    .toFixed(4)));

                // Calculate shallow_q
                q = qult/fs;
                $(''
                    '#shallow_q_' + i + '_'
                    + j).val(parseFloat(q
                    .toFixed(4)));
            }
        }
    }
</script>
<!-- /calculate -->

```

```

<div class="x_panel">
    <div class="x_title">
        <h2>Create - Shallow
            Foundation <small>Compute
            soil bearing capacity
            for shallow foundation</
            small></h2>
        <ul class="nav navbar-right
            panel_toolbox">
            <li><a class="collapse-
                link"><i class="fa fa
                -chevron-up"></i></a>
            </li>
            <li class="dropdown">
                <a href="#" class="
                dropdown-toggle"
                data-toggle="
                dropdown" role="
                button" aria-
                expanded="false"
                ><i class="fa fa-
                wrench"></i></a>
                <ul class="dropdown-
                menu" role="menu"
                >
                    <li><a href="#">
                        Settings 1</a>
                    </li>
                    <li><a href="#">
                        Settings 2</a>
                    </li>
                </ul>
            </li>
        </ul>
    </div>
    <div class="clearfix"></div>
</div>
<div class="x_content">
    @if(sizeof($samples) > 0)
    <?php $soil = ["clay", "silt"]
    ?>
    {!! Form::open(['name' => '
        store_form', 'method' =>
        'POST', 'route' => '
        shallows.store', 'class'
        => 'form-horizontal form-
        label-left']) !!}
    @foreach($boreholes as
        $borehole)
    {!! Form::hidden('bh_id[]',
        $borehole->id) !!}
    {!! Form::hidden('
        num_of_samples[]',
        $borehole->
        bh_num_of_samples) !!}
    <center>
        <h2>Borehole {{ $borehole
            ->bh_num }} <small>(
            ID#{{ $borehole->id
            }})</small></h2>
    </center>
    <table class="table table-
        striped table-hover">
        <thead>
            <tr>
                <th><a data-toggle
                    ="tooltip"
                    title="
                    Footing_width
                    ">B</a></th>
                <th><a data-toggle
                    ="tooltip"
                    title="Angle_
                    of_internal_
                    friction">&
                    Phi;</a></th>
                <th><a data-toggle
                    ="tooltip"
                    title="Soil_
                    bearing_
                    capacity_
                    factors_whose
                    _values_
                    depend_on_the
                    _angle_of_
                    internal_
                    friction">N<
                    sub>c</sub></a></th>
                <th><a data-toggle
                    ="tooltip"
                    title="Soil_
                    bearing_
                    capacity_
                    factors_whose
                    _values_
                    depend_on_the
                    _angle_of_

```

Listing 109: shallows/create.blade.php

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
    <div class="col-md-12">

```

```

        internal_
        friction">N<
        sub>&gamma;</
        sub></a></th>
<th><a data-toggle
=" tooltip"
title=" Soil_
bearing_
capacity_
factors_whose
_values_
depend_on_the
_angle_of_
internal_
friction">N<
sub>q</sub></
a></th>
<th><a data-toggle
=" tooltip"
title="
Effective_
unit_weight_
of_soil_below_
 footing_
level">&gamma
; <sub>1</sub>
</a></th>
<th><a data-toggle
=" tooltip"
title="
Effective_
unit_weight_
of_soil_above_
 footing_
level">&gamma
; <sub>2</sub>
</a></th>
<th><a data-toggle
=" tooltip"
title=" Factor
_of_safety">
FS</a></th>
</tr>
</thead>
<tbody>
<tr>
<td><input type="
number" name="
shallow_B []"
class="form-
control_
to_clear"
step="any"
value="{ { _old
(' shallow_B '
) _} }" id="
shallow_B_{{
_ $borehole ->
bh_num_ }}"
required ></td>
<td><input type="
number" name="
shallow_phi
[]" class="
form-control_
to_clear"
step="any"
value="{ { _old
(' shallow_phi
') _} }" id="
shallow_phi_
{{ _ $borehole
->bh_num_ }}"
required ></td>
<td><input type="
number" name="
shallow_Nc []"
class="form-
control_
to_clear"
step="any"
value="{ { _old
(' shallow_Nc
') _} }" id="
shallow_Nc_{{
_ $borehole ->
bh_num_ }}"
required ></td>
<td><input type="
number" name="
shallow_Ngamma
[]" class="
form-control_
to_clear"
step="any"
value="{ { _old
('
shallow_Ngamma
') _} }" id="
shallow_Ngamma_
{{ _ $borehole
->bh_num_ }}"
required ></td>
<td><input type="
number" name="
shallow_Nq []"
class="form-
control_
to_clear"
step="any"
value="{ { _old
(' shallow_Nq
') _} }" id="
shallow_Nq_{{
_ $borehole ->
bh_num_ }}"
required ></td>
<td><input type="
number" name="
shallow_gamma1
[]" class="
form-control_
to_clear"
step="any"
value="{ { _old
('
shallow_gamma1
') _} }" id="
shallow_gamma1_
{{ _ $borehole
->bh_num_ }}"
required ></td>
<td><input type="
number" name="
shallow_gamma2
[]" class="
form-control_
to_clear"
step="any"
value="{ { _old
('
shallow_gamma2
') _} }" id="
shallow_gamma2_
{{ _ $borehole
->bh_num_ }}"
required ></td>
<td><input type="
number" name="
shallow_fs []"
class="form-
control_
to_clear"
step="any"
value="{ { _old
(' shallow_fs
') _} }" id="
shallow_fs_{{
_ $borehole ->
bh_num_ }}"
required ></td>
</tr>
</tbody>
</table>
<table class="table_table-
striped_table-hover">
<thead>
<tr>
<th><a data-toggle
=" tooltip"
title=" Depth_
of_ footing_
below_lowest_
adjacent_soil_
surface">D<
sub>f</sub></a></th>
<th><a data-toggle
=" tooltip"
title=" Number
_of_blows">
SPT N-value</
a></th>
<th>Soil Type</th>
<th><a data-toggle
=" tooltip"
title="

```

```

        Cohesion_of_
        the_soil_
        below_footing
        _level">c</a
    </th>
<th><a data-toggle
=" tooltip"
title="
Ultimate_
gross_bearing
_capacity_or_
soil_bearing_
pressure">q<
sub>ult</sub
></a></th>
<th><a data-toggle
=" tooltip"
title="Design
_bearing_
_capacity">q<
a></th>
<th><a data-toggle
=" tooltip"
title="Design
_bearing_
_capacity">
Choose</a></
th>
</tr>
</thead>
<tbody>
@foreach($samples as
$sample)
@if($sample->bh_id ==
$borehole->id)
{!! Form::hidden('
sample_id[]',
$sample->id) !!}
<tr>
<td><input type="
number" name="
shallow_depth
[]" class="
form-control_
shallow_depth_
{{_ $borehole
->bh_num_}}
step="any"
value="{{_
$sample->
sample.to_}}
id="
shallow_depth_
{{_ $borehole
->bh_num_}} -
{{_ $sample->
sample.num_}}
" readonly></
td>
<td><input type="
number" name="
shallow_num_of_blows
[]" class="
form-control_
shallow_num_of_blows_
{{_ $borehole
->bh_num_}}
step="any"
value="{{_
$sample->
sample.num_of_blows
_}}
id="
shallow_num_of_blows_
{{_ $borehole
->bh_num_}} -
{{_ $sample->
sample.num_}}
" readonly></
td>
<td>
<?php
$shallow_soil_select_id
=
shallow_soil_
.
$borehole
->bh_num
.
'_' .
$sample->
sample_num
?>
{!! Form::
select('
shallow_soil
[]',
$soil,
old('
shallow_soil
[]'),
array('
btn_btn
default_
shallow_soil_
{{_
$borehole
->bh_num_
}}',
placeholder
', =>
Soil_type
...',
required'
=> 'true
', 'id'
=>
$shallow_soil_select_id
)) !!}
</td>
<td><input type="
number" name="
shallow_c[]"
class="form-
control_
to_clear_
shallow_c_{{_
$borehole->
bh_num_}}
step="any" id
="shallow_c_
{{_ $borehole
->bh_num_}} -
{{_ $sample->
sample.num_}}
" readonly
required></td
>
<td><input type="
number" name="
shallow_qult
[]" class="
form-control_
to_clear_
shallow_qult_
{{_ $borehole
->bh_num_}}
step="any" id
="
shallow_qult_
{{_ $borehole
->bh_num_}} -
{{_ $sample->
sample.num_}}
" readonly
required></td
>
<td><input type="
number" name="
shallow_q[]"
class="form-
control_
to_clear_
shallow_q_{{_
$borehole->
bh_num_}}
step="any" id
="shallow_q_
{{_ $borehole
->bh_num_}} -
{{_ $sample->
sample.num_}}
" readonly
required></td
>
<td style="text-
align:center;
"><input type
="checkbox"
name="
shallow_is_chosen
[]" value="{{
_ $sample->id_
}}"></td>
</tr>
@endif
@endforeach
</tbody>
</table>
<br/>
@endforeach
<div class="ln_solid"></div>
{!! Form::hidden('proj_id',
$project->id) !!}
{!! Form::hidden('
shallow_is_chosen[]', -1)
!!}

```

```

        {!! link_to(URL::previous(), '
        Back', array('class' => '
        btn_btn-default')) !!}
    {!! Form::button('Clear', ['
        class' => 'btn_btn-danger',
        'onclick' => 'clearForm()']) !!}
    {!! Form::submit('Submit', ['
        class' => 'btn_btn-primary',
        'style' => 'float:right',
        'onclick' => 'calculate()']) !!}
    {!! Form::button('Calculate',
        ['class' => 'btn_btn-success',
        'style' => 'float:right',
        'onclick' => 'calculate()']) !!}
    {!! Form::close() !!}
    @endif
</div>
</div>
</div>
</div>
@endsection
@section('js')
@parent
@include('shallows/calculate')
@endsection

```

Listing 110: shallows/edit.blade.php

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
    <div class="col-md-12">
        <div class="x_panel">
            <div class="x_title">
                <h2>Edit - Shallow Foundation
                <small>Compute soil
                bearing capacity for
                shallow foundation</small>
                </h2>
            <ul class="nav navbar-right
            panel_toolbox">
                <li><a class="collapse-
                link"><i class="fa fa-
                chevron-up"></i></a>
            </li>
                <li class="dropdown">
                    <a href="#" class="
                    dropdown-toggle"
                    data-toggle="
                    dropdown" role="
                    button" aria-
                    expanded="false"
                    ><i class="fa fa-
                    wrench"></i></a>
                    <ul class="dropdown-
                    menu" role="menu"
                    >
                        <li><a href="#">
                            Settings 1</a>
                        </li>
                        <li><a href="#">
                            Settings 2</a>
                        </li>
                    </ul>
                </li>
            </ul>
            <div class="clearfix"></div>
            <div class="x_content">
                @if(sizeof($shallows) > 0)
                <?php $soil = ["clay", "silt"]
                ?>
                {!! Form::open(['name' => '
                update_form', 'method' =>
                'PUT', 'route' => ['
                shallows.update',
                $project->id], 'class' =>
                'form-horizontal_form-
                label-left', 'id' => '
                update_form']) !!}
                @foreach($boreholes as
                $borehole)
                <center>
                    <h2>Borehole {{ $borehole
                    ->bh_num }} <small>(
                    ID#{{ $borehole->id
                    }})</small></h2>
                </center>
                <table class="table table-
                striped_table-hover">
                    <thead>
                        <tr>
                            <th><a data-toggle
                            ="tooltip"
                            title="
                            Footing_width
                            ">B</a></th>
                            <th><a data-toggle
                            ="tooltip"
                            title="Angle_
                            of_internal_
                            friction">&
                            Phi;</a></th>
                            <th><a data-toggle
                            ="tooltip"
                            title="Soil_
                            bearing_
                            capacity_
                            factors_whose
                            _values_
                            depend_on_the
                            _angle_of_
                            internal_
                            friction">N<
                            sub>c</sub></a></th>
                            <th><a data-toggle
                            ="tooltip"
                            title="Soil_
                            bearing_
                            capacity_
                            factors_whose
                            _values_
                            depend_on_the
                            _angle_of_
                            internal_
                            friction">N<
                            sub>c</sub></a></th>
                            <th><a data-toggle
                            ="tooltip"
                            title="Soil_
                            bearing_
                            capacity_
                            factors_whose
                            _values_
                            depend_on_the
                            _angle_of_
                            internal_
                            friction">N<
                            sub>q</sub></a></th>
                            <th><a data-toggle
                            ="tooltip"
                            title="
                            Effective_
                            unit_weight_
                            of_soil_below
                            _footing_
                            level">&gamma
                            ;<sub>1</sub></a></th>
                            <th><a data-toggle
                            ="tooltip"
                            title="
                            Effective_
                            unit_weight_
                            of_soil_above
                            _footing_
                            level">&gamma
                            ;<sub>2</sub></a></th>
                            <th><a data-toggle
                            ="tooltip"
                            title="Factor
                            of_safety">
                            FS</a></th>
                        </tr>
                    </thead>
                    <tbody>
                        <?php $i = 0 ?>
                        @while($borehole->id
                        != $shallows[$i
                        ]->bh_id)
                        <?php $i++ ?>
                        @endwhile
                        <tr>
                            <td>
                                <?php
                                $shallow_B
                                = Helper
                                ::
                                getValue(
                                old('
                                shallow_B
                                '),
                                $shallows
                                [$i]->
                                shallow_B
                                ) ?>
                            </td>
                        </tr>
                    </tbody>
                </table>
            </div>
        </div>
    </div>
</div>

```

<pre> <input type=" number" name=" shallow_B []" class =" form- control_ to_clear" step=" any" value="{{ _(float) \$shallow_B _}}" id=" shallow_B_ {{_ \$borehole ->bh.num_ }}" required> </td> <td> <?php \$shallow_phi = Helper :: getValue(old(' shallow_phi '), \$shallows [\$i]-> shallow_phi) ?> <input type=" number" name=" shallow_phi []" class =" form- control_ to_clear" step=" any" value="{{ _(float) \$shallow_phi _}}" id=" shallow_phi_ {{_ \$borehole ->bh.num_ }}" required> </td> <td> <?php \$shallow_Nc = Helper :: getValue(old(' shallow_Nc '), \$shallows [\$i]-> shallow_Nc) ?> <input type=" number" name=" shallow_Nc []" class =" form- control_ to_clear" step=" any" value="{{ _(float) \$shallow_Nc _}}" id=" shallow_Nc_ {{_ \$borehole ->bh.num_ }}" required> </td> <td> <?php \$shallow_Ngamma = Helper :: getValue(old(' shallow_Ngamma '), \$shallows [\$i]-> shallow_Ngamma) ?> </pre>	<pre> [\$i]-> shallow_Ngamma) ?> <input type=" number" name=" shallow_Ngamma []" class =" form- control_ to_clear" step=" any" value="{{ _(float) \$shallow_Ngamma _}}" id=" shallow_Ngamma_ {{_ \$borehole ->bh.num_ }}" required> </td> <td> <?php \$shallow_Nq = Helper :: getValue(old(' shallow_Nq '), \$shallows [\$i]-> shallow_Nq) ?> <input type=" number" name=" shallow_Nq []" class =" form- control_ to_clear" step=" any" value="{{ _(float) \$shallow_Nq _}}" id=" shallow_Nq_ {{_ \$borehole ->bh.num_ }}" required> </td> <td> <?php \$shallow_gamma1 = Helper :: getValue(old(' shallow_gamma1 '), \$shallows [\$i]-> shallow_gamma1) ?> <input type=" number" name=" shallow_gamma1 []" class =" form- control_ to_clear" step=" any" value="{{ _(float) \$shallow_gamma1 _}}" id=" shallow_gamma1_ {{_ \$borehole ->bh.num_ }}" required> </td> <td> <?php \$shallow_gamma2 = Helper :: getValue(old(' shallow_gamma2 '), \$shallows </pre>
--	---

```

        shallow_gamma2
        '),
        $shallows
        [ $i]->
        shallow_gamma2
        ) ?>
<input type="
number"
name="
shallow_gamma2
[]" class
="form-
control_
to_clear"
step="
any"
value="{{
_(float)
$shallow_gamma2
-}}" id="
shallow_gamma2_
{{_
$borehole
->bh.num_
}}}"
required>
</td>
<td>
<?php
    $shallow_fs
    = Helper
    ::
    getValue(
    old('
    shallow_fs
    '),
    $shallows
    [ $i]->
    shallow_fs
    ) ?>
<input type="
number"
name="
shallow_fs
[]" class
="form-
control_
to_clear"
step="
any"
value="{{
_(float)
$shallow_fs
-}}" id="
shallow_fs_
{{_
$borehole
->bh.num_
}}}"
required>
</td>
</tr>
</tbody>
</table>
<table class="table table-
striped table-hover">
<thead>
<tr>
<th><a data-toggle
="tooltip"
title="Depth_
of_footing_
below_lowest_
adjacent_soil_
surface">D<
sub>f</sub></a></th>
<th><a data-toggle
="tooltip"
title="Number
_of_blows">
SPT N-value</
a></th>
<th>Soil Type</th>
<th><a data-toggle
="tooltip"
title="
Cohesion_of_
the_soil_
below_footing_
_level">c</a
></th>
<th><a data-toggle
="tooltip"
title="
Ultimate_
gross_bearing_
_capacity_or_
soil-bearing_
pressure">q<
sub>ult</sub
></a></th>
</thead>
<tbody>
<?php $j = 0; $k = 1;
?>
@foreach($shallows as
$shallow)
@if($shallow->bh.id
== $borehole->id
)
{!! Form::hidden('
sample_id [ ] ',
$shallow->
sample_id) !!}
{!! Form::hidden('
shallow_id [ ] ',
$shallow->id) !!}
<tr>
<td><input type="
number" name="
shallow_depth
[]" class="
form-control_
shallow_depth_
{{_ $borehole
->bh.num_}}}"
step="any"
value="{{_(
float)
$samples[ $j
]->sample.to_
-}}" id="
shallow_depth_
{{_ $borehole
->bh.num_}}-
{{_ $k_}}}"
readonly></td>
<td><input type="
number" name="
shallow_num_of_blows
[]" class="
form-control_
shallow_num_of_blows_
{{_ $borehole
->bh.num_}}}"
step="any"
value="{{_(
float)
$samples[ $j
]->
sample.num_of_blows
-}}" id="
shallow_num_of_blows_
{{_ $borehole
->bh.num_}}-
{{_ $k_}}}"
readonly></td>
<td>
<?php
    $shallow_soil_select_id
    =
    $shallow_soil_
    .
    $borehole
    ->
    bh.num
    .
    ' -
    ' .
    $k;
    $shallow_soil
    =
    Helper
    ::
    getValue(
    old(
    '
    shallow_soil
    '),
    $shallow
    ->
    shallow_soil

```



```

dropdown-toggle"
data-toggle="
dropdown" role="
button" aria-
expanded="false"
><i class="fa fa-
wrench"></i></a>
<ul class="dropdown-
menu" role="menu"
>
<li><a href="#">
Settings 1</a>
</li>
<li><a href="#">
Settings 2</a>
</li>
</ul>
</li>
</div>
<div class="clearfix"></div>
</div>
<div class="x_content">
@if(sizeof($shallows) > 0)
@foreach($boreholes as
$borehole)
<center>
<h2>Borehole {{ $borehole
->bh_num }} <small>(<br>
ID#{{ $borehole->id
}})</small></h2>
</center>
<table class="table table-
striped table-hover">
<thead>
<tr>
<th><a data-toggle
="tooltip"
title="
Footing_width
">B</a></th>
<th><a data-toggle
="tooltip"
title="Angle_
of_internal_
friction">&
Phi;</a></th>
<th><a data-toggle
="tooltip"
title="Soil_
bearing_
capacity_
factors_whose
_values_
depend_on_the
_angle_of_
internal_
friction">N<
sub>c</sub></a></th>
<th><a data-toggle
="tooltip"
title="Soil_
bearing_
capacity_
factors_whose
_values_
depend_on_the
_angle_of_
internal_
friction">N<
sub>q</sub></a></th>
<th><a data-toggle
="tooltip"
title="Soil_
bearing_
capacity_
factors_whose
_values_
depend_on_the
_angle_of_
internal_
friction">N<
sub>q</sub></a></th>
<th><a data-toggle
="tooltip"
title="
Effective_
unit_weight_
of_soil_below
_footing_
level">&gamma
; <sub>1</sub></a></th>
<th><a data-toggle
="tooltip"
title="
Effective_
unit_weight_
of_soil_above
_footing_
level">&gamma
; <sub>2</sub></a></th>
</thead>
<tbody>
<tr>
<td>{{ (float)
$shallows[ $i
]->shallow_B
}}</td>
<td>{{ (float)
$shallows[ $i
]->
shallow_phi
}}</td>
<td>{{ (float)
$shallows[ $i
]->shallow_Nc
}}</td>
<td>{{ (float)
$shallows[ $i
]->
shallow_Ngamma
}}</td>
<td>{{ (float)
$shallows[ $i
]->shallow_Nq
}}</td>
<td>{{ (float)
$shallows[ $i
]->
shallow_gamma1
}}</td>
<td>{{ (float)
$shallows[ $i
]->
shallow_gamma2
}}</td>
<td>{{ (float)
$shallows[ $i
]->shallow_fs
}}</td>
</tr>
</tbody>
</table>
<table class="table table-
striped table-hover">
<thead>
<tr>
<th><a data-toggle
="tooltip"
title="Depth_
of_footing_
below_lowest_
adjacent_soil
_surface">D<
sub>f</sub></a></th>
<th><a data-toggle
="tooltip"
title="Number
_of_blows">
SPT N-value</a></th>
<th>Soil Type</th>
<th><a data-toggle
="tooltip"
title="
Cohesion_of_
the_soil_
below_footing_
_level">c</a>
</th>
<th><a data-toggle
="tooltip"
title="
Ultimate_
gross_bearing_
_capacity_or_
soil_bearing_
pressure">q<
sub>ult</sub>
</th>

```



```

        </a></th>
        <th><a data-toggle="tooltip" title="Design bearing capacity">q</a></th>
    </tr>
</thead>
<tbody>
<?php $j = 0 ?>
@foreach( $shallows as $shallow)
@if($shallow->bh_id == $borehole->id)
<?php ($shallow->shallow_is_chosen == 1) ? $class = "chosen" : $class = "" ?>
<tr class="{$_$class_}">
<td>{{ (float) $samples[$j] ->sample_to }}</td>
<td>{{ (float) $samples[$j] -> sample_num_of_blows }}</td>
<td>{{ ($shallow->shallow_soil == 0)? 'clay': 'silt' }}</td>
<td>{{ (float) $shallow->shallow_c }}</td>
<td>{{ (float) $shallow->shallow_qult }}</td>
<td>{{ (float) $shallow->shallow_q }}</td>
</tr>
@elseif
<?php $j++ ?>
@endforeach
</tbody>
</table>
<br/>
@endforeach
<div class="ln_solid"></div>
{!! Form::open(['method' => 'DELETE', 'route' => ['shallows.destroy', $project->id], 'onsubmit' => 'return_confirm("Delete_SBC_data_for_shallow_foundation_permanently?")']) !!}
{!! Form::submit('Delete', ['class' => 'btn btn-danger']) !!}
<a href="{$_route('shallows.edit', $_$project->id)}" class="btn btn-info" style="float:right"><i class="fa fa-edit" style="float:right"> Edit </a>
{!! Form::close() !!}
@endif
</div>
</div>
</div>
</div>
@endsection

Listing 112: subreports/createBL.blade.php

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
<div class="col-md-12">
<div class="x_panel">
<div class="x_title">
<h2>Add/Update Borehole Log</h2>
<ul class="nav navbar-right panel_toolbox">
<li><a class="collapse-link" href="#">i class="fa fa-chevron-up"></a></li>
<li class="dropdown">
<a href="#"> class="dropdown-toggle" data-toggle="dropdown" role="button" aria-expanded="false"><i class="fa fa-wrench"></i></a>
<ul class="dropdown-menu" role="menu">
<li><a href="#"> Settings 1</a>
</li>
<li><a href="#"> Settings 2</a>
</li>
</ul>
</li>
</ul>
</div>
<div class="clearfix"></div>
<div class="x_content">
@if($counts == 0)
<h4 align="center">There are no existing samples.</h4>
@else
<div id="canvas_div">
@for($i = 0; $i < sizeof($boreholes_info); $i++)
<center>
<canvas id="bh_canvas" {{ $i }}" width="1000" height="1225" style="margin-top:15px;"></canvas>
</center>
<br />
@endfor
</div>
@endif
<div class="ln_solid"></div>
</div>
</div>
</div>
@endsection
@parent
<script type="text/javascript">
var project = <?php echo json_encode($project) ?>;
var counts = <?php echo json_encode($counts) ?>;
var samples_info = <?php echo json_encode($samples_info) ?>;
var boreholes_info = <?php echo json_encode($boreholes_info) ?>;

function drawTable(bh_info, cnv) {
var gridOptions = {
constants: {
// General constants
text_padding: 3,
canvas_width: cnv.width,
canvas_height: cnv.height,

// Header constants
header_cols_width: 1000/3,
header_cols_num: 3,
header_cols_separation: cnv.width/3, // header_cols_width / header_cols_num
header_rows_num: 7,
header_rows_separation: 23,

// Table header constants
table_start_y: 275,
table_header_start_y: 161,
table_header_width: 114, // table_start_y - table_header_start_y

// Column constants
scale: 200/60, // colsb_width / 60; colsb scale
colsa_num: 8,
colsb_num: 12,
colsc_num: 3,

```

```

        colsa_width: 45 * 8, //
            colsa_separation *
            colsa_num
        colsb_width: 200,
        colsc_width: 440,
        colsa_separation: 45,
        colsb_separation: 200/12, //
            colsb_width / colsb_num
        colsc_separation: 440/3, //
            colsc_width / colsc_num
        colsb_header_num: 7,

        // Color constants
        major_lines_color: '#000000',
        minor_lines_color: '#D3D3D3'
    };

var ctx = cnv.getContext('2d');
var num_of_samples = bh_info['
    bh_num_of_samples'];

var table_headers = ["Depth_(m)", "
    Type_of_Sampling", "From_(m)", "To
    _(m)", "Len._of_Run_(m)", "Len._of_
    Sample_(m)", "Rec._%", "USCS", "
    Color", "Soil_Description", "
    Other_Test_Results"];
var headers = ["Project:_", "project['
    proj_name'], "Location:_", "
    project['proj_location'], "
    Borehole_No.:", "bh_info['bh_num
    '], "Date_Started:", "bh_info['
    bh_start_date'], "Date_Finished:_
    " + bh_info['bh_end_date'], "
    Weather:", "bh_info['bh_weather'
    ], "Surface_Elevation:", "
    bh_info['bh_surface_elevation'],
    "Water_Elevation:", "bh_info['
    bh_water_elevation'], "Total_
    Boring_Depth:", "bh_info['
    bh_max_depth'], "Height_of_Drop:_
    " + bh_info['bh_height_of_drop'],
    "Hammer_Weight:", "bh_info['
    bh_hammer_weight'], "Drill_Rig:_",
    " + bh_info['bh_drill_rig'], "
    Drill_Pump:", "bh_info['
    bh_drill_pump'], "Operator:", "
    bh_info['bh_operator'], "
    Inspector:", "bh_info['
    bh_inspector'], "Casing:", "
    bh_info['bh_casing'] + " |Rod:", "
    bh_info['bh_rod'] + " |Sampler:",
    " + bh_info['bh_sampler'], "Type_of
    _Sampling:"];

var d_from = [];
var d_to = [];
var run_len = [];
var sample_len = [];
var rec_percent = [];
var uscs = [];
var color = [];
var descr = [];
var other = [];
var spt = [0];

for (i = 0; i < samples_info.length; i
    ++){
    if (samples_info[i]['bh_id'] ==
        bh_info['id']) {
        d_from.push(samples_info[i]['
            sample_from']);
        d_to.push(samples_info[i]['
            sample_to']);
        spt.push(samples_info[i]['
            sample_num_of_blows']);
        run_len.push(samples_info[i]['
            sample_run_length']);
        sample_len.push(samples_info[i]
            ['sample_sample_length'
            ]);
        rec_percent.push(samples_info[
            i]['sample_rec_percent'])
            ;
        uscs.push(samples_info[i]['
            uscs_id']);
        color.push(samples_info[i]['
            sample_color']);
        descr.push(samples_info[i]['
            uscs_description']);
        other.push(samples_info[i]['
            sample_other_results']);
    }
}

var data = {
    d_from: d_from,
    d_to: d_to,
    run_len: run_len,
    sample_len: sample_len,
    rec_percent: rec_percent,
    uscs: uscs,
    spt: spt,
    color: color,
    descr: descr,
    other: other
};

drawLines(ctx, gridOptions.constants,
    num_of_samples);
addHeaders(ctx, gridOptions.constants,
    num_of_samples, headers,
    table_headers);
if (samples_info.length > 0) {
    addData(ctx, gridOptions.constants
        , num_of_samples, data);
}

return;
}

function drawLines(ctx, grid_options,
    num_of_samples) {
    var i = null;
    var x = null;
    var y = null;
    var ctr = null;

    ctx.strokeStyle = grid_options.
        major_lines_color;
    ctx.beginPath();

    // Draw header vertical lines
    for (i = 0; i <= grid_options.
        header_cols_separation; i++) {
        x = i * grid_options.
            header_cols_separation;
        switch(i) {
            case 0: ctx.moveTo(x, 0);
                break;
            case 1: ctx.moveTo(x,
                grid_options.
                header_rows_separation);
                break;
            case 2: ctx.moveTo(x, 2*
                grid_options.
                header_rows_separation);
                break;
            case 3: ctx.moveTo(x, 0);
                break;
        }

        ctx.lineTo(x, grid_options.
            table_header_start_y);
    }

    // Draw header horizontal lines
    for (i = 0; i < grid_options.
        header_rows_num; i++) {
        y = i * grid_options.
            header_rows_separation;
        ctx.moveTo(0, y);
        ctx.lineTo(grid_options.
            canvas_width, y);
    }

    // Draw main table lines
    // Draw major vertical lines
    // Draw colsa borders
    for (i = 0; i <= grid_options.
        colsa_num; i++) {
        x = i * grid_options.
            colsa_separation;
        ctx.moveTo(x, grid_options.
            table_header_start_y);
        ctx.lineTo(x, grid_options.
            canvas_height);
    }

    // Draw colsc borders
    for (i = 0; i <= grid_options.
        colsc_num; i++) {
        x = grid_options.colsa_width +
            grid_options.colsb_width + (i
            * grid_options.
            colsc_separation);
        ctx.moveTo(x, grid_options.
            table_header_start_y);
        ctx.lineTo(x, grid_options.
            canvas_height);
    }

    // Draw major horizontal lines
    ctx.moveTo(0, grid_options.
        table_header_start_y);
    ctx.lineTo(grid_options.canvas_width,
        grid_options.table_header_start_y
    );
}

```

```

    );
    for (i = 0; i <= num_of_samples; i++)
    {
        y = i * grid_options.
            colsa_separation +
            grid_options.table_start_y;
        ctx.moveTo(0, y);
        ctx.lineTo(grid_options.
            canvas_width, y);
    }

    ctx.moveTo(0, grid_options.
        canvas_height);
    ctx.lineTo(grid_options.canvas_width,
        grid_options.canvas_height);

    ctx.lineWidth = 2;
    ctx.stroke();
    ctx.closePath();

    // Draw minor vertical lines
    ctx.strokeStyle = grid_options.
        minor_lines_color;
    ctx.beginPath();
    for (i = 1; i < grid_options.colsb_num
        ; i++) {
        x = grid_options.colsa_width + (i
            * grid_options.
            colsb_separation);
        ctx.moveTo(x, grid_options.
            table_start_y);
        ctx.lineTo(x, (num_of_samples) *
            grid_options.colsa_separation
            + grid_options.table_start_y
            );
    }

    ctx.lineWidth = 2;
    ctx.stroke();
    ctx.closePath();

    return;
}

function addHeaders(ctx, grid_options,
    num_of_samples, headers,
    table_headers) {
    ctx.font = "11pt_Arial";
    ctx.textBaseline = "bottom";
    ctx.textAlign = "center";
    ctx.fillText("BOREHOLE_LOG",
        grid_options.canvas_width/2,
        grid_options.
        header_rows_separation);

    ctx.textAlign = "start";

    y = 2 * grid_options.
        header_rows_separation;
    x = grid_options.text_padding;
    ctx.fillText(headers[0], x, y);
    ctx.fillText(headers[1], grid_options.
        header_cols_separation + x, y);

    // Add top header
    ctr = 0;
    for (i = 2; i < headers.length; i = i
        + 3) {
        y = (ctr+3) * grid_options.
            header_rows_separation;
        x = grid_options.text_padding;
        ctx.fillText(headers[i], x, y);

        if (i + 1 == 15) {
            var res = headers[i+1].split("
                ");
            x = grid_options.
                header_cols_separation +
                grid_options.text_padding
                ;
            ctx.fillText(res[0], x, y);

            x = x + grid_options.
                header_cols_separation/3;
            ctx.fillText(res[1], x, y);

            x = x + (grid_options.
                header_cols_separation/3)
                ;
            ctx.fillText(res[2], x, y);
        }
        else {
            x = grid_options.
                header_cols_separation +
                grid_options.text_padding
                ;
            ctx.fillText(headers[i+1], x,
                y);
        }
    }

    for (i = 0; i <= num_of_samples; i++)
    {
        x = 2*grid_options.
            header_cols_separation +
            grid_options.text_padding;
        ctx.fillText(headers[i+2], x, y);

        ctr++;
    }

    // Add table header
    // Add colsa header
    ctx.font = "9pt_Arial";
    ctx.textBaseline = "middle";
    y = grid_options.table_start_y -
        grid_options.text_padding;
    for (i = 0; i < grid_options.colsa_num
        ; i++) {
        ctx.save();
        x = (i*grid_options.
            colsa_separation) + (
            grid_options.colsa_separation
            /2);
        ctx.translate(x, y);
        ctx.rotate(-90*Math.PI/180);
        ctx.translate(-x,-y);
        ctx.fillText(table_headers[i], x,
            y);
        ctx.restore();
    }

    // Add colsb header
    ctx.font = "10pt_Arial";
    ctx.textAlign = "center";
    ctx.textBaseline = "middle";
    x = grid_options.colsa_width + (
        grid_options.colsb_width/2);
    y = (grid_options.table_start_y -
        grid_options.text_padding) - (
        grid_options.table_header_width
        /2);
    ctx.fillText("SPT_N-Value", x, y);

    ctx.textAlign = "center";
    ctx.textBaseline = "bottom";
    y = grid_options.table_start_y;
    for (i = 0; i < grid_options.
        colsb_header_num; i++) {
        x = grid_options.colsa_width + (i
            *2*grid_options.
            colsb_separation);
        if (i == 0) {
            ctx.textAlign = "left";
        }
        else if (i == 6) {
            ctx.textAlign = "right";
        }
        else {
            ctx.textAlign = "center";
        }
        ctx.fillText((i)*10, x, y);
    }

    // Add colsc header
    ctx.textAlign = "center";
    ctx.textBaseline = "middle";
    y = (grid_options.table_start_y -
        grid_options.text_padding) - (
        grid_options.table_header_width
        /2);
    for (i = 0; i <= grid_options.
        colsc_num; i++) {
        x = grid_options.colsa_width +
            grid_options.colsb_width + (i
            * grid_options.
            colsc_separation) + (
            grid_options.colsc_separation
            /2);
        ctx.fillText(table_headers[i + 8],
            x, y);
    }

    return;
}

function addData(ctx, grid_options,
    num_of_samples, data) {
    ctx.font = "11pt_Arial";
    ctx.textAlign = "start";
    ctx.textBaseline = "bottom";
    ctx.strokeStyle = grid_options.
        major_lines_color;
    ctx.beginPath();

    ctr = 0;
    for (i = 1; i <= num_of_samples; i++)
    {
        // Depth column

```

```

x = grid_options.text_padding;
y = i * grid_options.
  colsa_separation +
  grid_options.table_start_y;
ctx.fillText(data.d_to[ctr], x, y)
;

// Type of sampling column
x = grid_options.colsa_separation;
y = (i-1) * grid_options.
  colsa_separation +
  grid_options.table_start_y;
ctx.moveTo(x,y);
ctx.lineTo(x+grid_options.
  colsa_separation,y+
  grid_options.colsa_separation
);
ctx.moveTo(x,y+grid_options.
  colsa_separation);
ctx.lineTo(x+grid_options.
  colsa_separation,y);

// From column
x = 2*grid_options.
  colsa_separation +
  grid_options.text_padding;
y = i * grid_options.
  colsa_separation +
  grid_options.table_start_y;
ctx.fillText(data.d_from[ctr], x,
  y);

// To column
x = 3*grid_options.
  colsa_separation +
  grid_options.text_padding;
ctx.fillText(data.d_to[ctr], x, y)
;

// Length of run column
x = 4*grid_options.
  colsa_separation +
  grid_options.text_padding;
ctx.fillText(data.run_len[ctr], x,
  y);

// Length of sample column
x = 5*grid_options.
  colsa_separation +
  grid_options.text_padding;
ctx.fillText(parseFloat(data.
  sample_len[ctr]).toFixed(
  2), x, y);

// Rec percent column
x = 6*grid_options.
  colsa_separation +
  grid_options.text_padding;
ctx.fillText(parseFloat(data.
  rec_percent[ctr]).toFixed(
  3) + '%', x, y);

// USCS column
x = 7*grid_options.
  colsa_separation +
  grid_options.text_padding;
ctx.fillText(data.uscs[ctr], x, y)
;

// SPT N-value column
x = 8*grid_options.
  colsa_separation;
y = (i-1) * grid_options.
  colsa_separation +
  grid_options.table_start_y;
ctx.moveTo(x + data.spt[ctr]*
  grid_options.scale, y);
ctx.lineTo(x + data.spt[ctr+1]*
  grid_options.scale, y +
  grid_options.colsa_separation
);
ctx.arc(x + data.spt[ctr+1]*
  grid_options.scale, y +
  grid_options.colsa_separation
, 5, 0, 2 * Math.PI, true);
ctx.fillText(data.spt[ctr+1], x +
  data.spt[ctr+1]*grid_options.
  scale + grid_options.
  text_padding, y +
  grid_options.colsa_separation
);
//ctx.arc(colsa_width,
  grid_options.colsa_separation
, 5, 0, 2 * Math.PI, true);
//ctx.fill();

// Color column
x = grid_options.colsa_width +
  grid_options.colsb_width +
  grid_options.text_padding;
y = i * grid_options.
  colsa_separation +
  grid_options.table_start_y;
ctx.fillText(data.color[ctr], x, y
);

// Description column
x = grid_options.colsa_width +
  grid_options.colsb_width +
  grid_options.colsc_separation
  + grid_options.text_padding;
ctx.fillText(data.descr[ctr], x, y
);

// Remarks column
x = grid_options.colsa_width +
  grid_options.colsb_width +
  (2*grid_options.
  colsc_separation) +
  grid_options.text_padding;
ctx.fillText(data.other[ctr], x, y
);
ctx++;
}

ctx.lineWidth = 2;
ctx.stroke();
ctx.closePath();

return;
}

var canvas_div = document.getElementById('
  canvas_div');

if (canvas_div !== null) {
  for (z = 0; z < boreholes_info.length;
    z++) {
    var bh_canvas_id = 'bh_canvas' + z
    ;
    var cnv = document.getElementById(
      bh_canvas_id);

    // Draw borehole log
    drawTable(boreholes_info[z], cnv);
  }
}

window.onload = function() {
  for (z = 0; z < boreholes_info.length;
    z++) {
    var bh_canvas_id = 'bh_canvas' + z
    ;
    var cnv = document.getElementById(
      bh_canvas_id);

    var img_uri = cnv.toDataURL();
    var bh_id = boreholes_info[z]['id'
    ];
    var proj_id = project['id'];
    var csrf_token = $('meta[name="
      csrf-token"]').attr('content'
    );

    // Save image
    $.ajax({
      type: "POST",
      url: "store-image",
      data: {
        image_uri: img_uri,
        bh_number: bh_id,
        proj_id: proj_id,
        sb_type: 'BL',
        _token: csrf_token
      },
      success: function(data){
        //alert(data);
      },
      error: function(xhr, error) {
        alert(error.Message);
      }
    });
  }
};
</script>
<@endsection

```

Listing 113: subreports/createSP.blade.php

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">

```

```

<div class="col-md-12">
  <div class="x_panel">
    <div class="x_title">
      <h2>Add Soil Profile </h2>
      <ul class="nav navbar-right panel_toolbox">
        <li><a class="collapse-link" <i class="fa fa-chevron-up"></i></a></li>
      </ul>
      <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-expanded="false" <i class="fa fa-wrench"></i></a>
        <ul class="dropdown-menu" role="menu">
          <li><a href="#">Settings 1</a></li>
          <li><a href="#">Settings 2</a></li>
        </ul>
      </li>
    </ul>
    <div class="clearfix"></div>
  </div>
  <div class="x_content">
    <if($counts == 0)>
      <h4 align="center">There are no existing samples.</h4>
    <else>
      <div id="canvas_div">
        <?php
          $num_of_bh = $project
            ->
              proj_num_of_boreholes
            ;
          $num_of_canvas = ceil(
            $num_of_bh/2);
          ?>
          @for($i = 0; $i <
            $num_of_canvas; $i++)
            <center>
              <canvas id="bh_canvas"
                {{$i}} width="
                1000" height="
                1225" style="
                margin-top:15px;"
                ></canvas>
            </center>
          <br />
          @endfor
        </div>
        @endif
        <div class="ln_solid"></div>
      </div>
    </div>
  </div>
</div>
<endsection>
@section('js')
@parent
<script type="text/javascript">
  var project = <?php echo json_encode(
    $project) ?>;
  var counts = <?php echo json_encode(
    $counts) ?>;
  var samples_info = <?php echo json_encode(
    $samples_info) ?>;
  var boreholes_info = <?php echo
    json_encode($boreholes_info) ?>;
  var num_of_canvas = <?php echo json_encode(
    $num_of_canvas) ?>;

  function drawProfile(cnv, grid_options,
    bh_info, num, min_depth) {

    var ctx = cnv.getContext('2d');
    var max_depth = bh_info['bh_max_depth']
      ];
    var water_elevation = bh_info['
      bh_water_elevation'];
    var num_of_samples = bh_info['
      bh_num_of_samples'];
    var headers = ["Borehole_No." +
      bh_info['bh_num'], "Ground_
      Elevation", "Depth:" + max_depth
      + "m."];
    var table_headers = ["Depth_(m)", "
      Symbol", "USCS"];

    var d_from = [];
    var d_to = [];
    var run_length = [];
    var uscs = [];
    var spt = [0];

    for (i = 0; i < samples_info.length; i
      ++){
      if (samples_info[i]['bh_id'] ==
        bh_info['id']) {
        d_from.push(samples_info[i]['
          sample_from']);
        d_to.push(samples_info[i]['
          sample_to']);
        run_length.push(samples_info[i]
          ['sample_run_length']);
        uscs.push(samples_info[i]['
          uscs_id']);
        spt.push(samples_info[i]['
          sample_num_of_blows']);
      }
    }

    var bh_min_depth = Math.min.apply(null
      , d_from);

    var data = {
      d_from: d_from,
      d_to: d_to,
      run_length: run_length,
      uscs: uscs,
      spt: spt,

      min_depth: bh_min_depth,
      max_depth: max_depth,
      water_elevation: water_elevation,
      num_of_samples: num_of_samples,
    };

    if (num == 1) {
      start_x = grid_options.bh1_start_x
        ;
    }
    else {
      start_x = grid_options.bh2_start_x
        ;
    }

    var add_y = drawLines(ctx,
      grid_options, data, start_x,
      min_depth);
    addHeaders(ctx, grid_options, data,
      headers, table_headers, start_x,
      min_depth, add_y);
    if (samples_info.length > 0) {
      addData(ctx, grid_options,
        num_of_samples, data, start_x
        , min_depth, add_y);
    }

    return;
  }

  function drawLines(ctx, grid_options, data
    , start_x, min_depth) {
    ctx.strokeStyle = grid_options.
      major_lines_color;
    ctx.beginPath();

    // Draw vertical lines
    for (i = 0; i <= grid_options.
      colsa_num; i++) {
      x = start_x + (i * grid_options.
        colsa_separation);
      ctx.moveTo(x, grid_options.
        table_header_start_y);
      ctx.lineTo(x, grid_options.
        rows_end);
    }

    x = x + grid_options.colsb_width;
    ctx.moveTo(x, grid_options.
      table_header_start_y);
    ctx.lineTo(x, grid_options.rows_end);

    // Draw header horizontal lines
    var add_y = 0;
    if (data.min_depth > min_depth) {
      x = start_x;
      add_y = (data.min_depth -
        min_depth) * (grid_options.
        rows_separation/0.3);
      y = grid_options.min_depth_y +
        add_y;
      ctx.moveTo(x - (3*grid_options.
        extra_line_x), y);
      ctx.lineTo(x - (2*grid_options.
        extra_line_x), y);
    }
  }

```

```

        ctx.moveTo(x, y);
        ctx.lineTo(x + grid_options.
            cols_width, y);
    }

    x = start_x;
    y = grid_options.table_header_start_y;
    ctx.moveTo(x, y);
    ctx.lineTo(x + grid_options.cols_width
        , y);

    y = grid_options.table_start_y;
    ctx.moveTo(x, y);
    ctx.lineTo(x + grid_options.cols_width
        , y);

    y = grid_options.min_depth_y +
        grid_options.rows_height;
    ctx.moveTo(x, y);
    ctx.lineTo(x + grid_options.cols_width
        , y);

    ctx.lineWidth = 3;
    ctx.stroke();
    ctx.closePath();

    // Draw water elevation line
    ctx.beginPath();

    x = start_x;
    var add_y_min_depth = min_depth * (
        grid_options.rows_separation/0.3)
        ;
    y = grid_options.min_depth_y + ((data.
        water_elevation) * (grid_options.
        rows_separation/0.3)) -
        add_y_min_depth;
    ctx.moveTo(x - (3*grid_options.
        extra_line_x), y);
    ctx.lineTo(x - (2*grid_options.
        extra_line_x), y);
    ctx.moveTo(x, y);
    ctx.lineTo(x + grid_options.cols_width
        + grid_options.extra_line_x, y);

    ctx.lineWidth = 3;
    ctx.stroke();
    ctx.closePath();

    // Draw minor vertical lines
    ctx.strokeStyle = grid_options.
        minor_lines_color;
    ctx.beginPath();

    for (i = 1; i < grid_options.colsb_num
        ; i++) {
        x = start_x + grid_options.
            colsa_width + (i *
                grid_options.colsb_separation
            );
        ctx.moveTo(x, grid_options.
            table_start_y);
        ctx.lineTo(x, grid_options.
            rows_end);
    }

    ctx.lineWidth = 2;
    ctx.stroke();
    ctx.closePath();

    return add_y;
}

function addHeaders(ctx, grid_options,
    data, headers, table_headers, start_x
    , min_depth, add_y) {
    ctx.font = "11pt_Arial";
    ctx.textBaseline = "bottom";
    ctx.textAlign = "start";

    // Add top header
    x = start_x;
    for (i = 1; i <= grid_options.
        header_row_num; i++) {
        y = i * grid_options.
            header_row_separation;
        ctx.fillText(headers[i-1], x, y);
    }

    // Minimum depth label
    ctx.font = "8pt_Arial";
    ctx.textBaseline = "middle";
    x = x + grid_options.text_padding -
        (2*grid_options.extra_line_x);
    y = grid_options.min_depth_y;
    ctx.fillText(min_depth.toPrecision(2),
        x, y);

    if (data.min_depth > min_depth) {
        y = grid_options.min_depth_y +
            add_y;
        ctx.fillText(data.min_depth.
            toPrecision(2), x, y);
    }

    // Water elevation label
    x = start_x + grid_options.
        text_padding - (2*grid_options.
            extra_line_x);
    var add_y_min_depth = min_depth * (
        grid_options.rows_separation/0.3)
        ;
    y = grid_options.min_depth_y + ((data.
        water_elevation) * (grid_options.
        rows_separation/0.3)) -
        add_y_min_depth;
    ctx.fillText(data.water_elevation, x,
        y);

    // Add table header
    // Add first three headers
    ctx.font = "11pt_Arial";
    y = grid_options.table_start_y -
        grid_options.text_padding;
    for (i = 0; i < 3; i++) {
        ctx.save();
        x = start_x + (i*grid_options.
            colsa_separation) + (
                grid_options.colsa_separation
                /2);
        ctx.translate(x, y);
        ctx.rotate(-90*Math.PI/180);
        ctx.translate(-x,-y);
        ctx.fillText(table_headers[i], x,
            y);
        ctx.restore();
    }

    // Add last header
    ctx.textAlign = "center";
    ctx.textBaseline = "middle";
    x = start_x + grid_options.colsa_width
        + (grid_options.colsb_width/2);
    y = (grid_options.table_start_y -
        grid_options.text_padding) - (
        grid_options.table_header_width
        /2);
    ctx.fillText("SPT_N-Value", x, y);

    ctx.font = "10pt_Arial";
    ctx.textAlign = "center";
    ctx.textBaseline = "bottom";
    y = grid_options.table_start_y;
    for (i = 0; i < grid_options.
        colsb_header_num; i++) {
        x = start_x + grid_options.
            colsa_width + (i*2*
                grid_options.colsb_separation
            );
        if (i == 0) {
            ctx.textAlign = "left";
        }
        else if (i == 6) {
            ctx.textAlign = "right";
        }
        else {
            ctx.textAlign = "center";
        }
        ctx.fillText((i)*10, x, y);
    }

    return;
}

function addData(ctx, grid_options,
    num_of_samples, data, start_x,
    min_depth, add_y) {
    ctx.font = "11pt_Arial";
    ctx.textAlign = "start";

    ctx.strokeStyle = grid_options.
        major_lines_color;
    ctx.beginPath();

    ctx.textBaseline = "bottom";

    // Symbol colum
    var sources = {};
    for (i = 0; i < num_of_samples; i++) {
        var source_var = 'sample' + (i+1);
        sources[source_var] = '../././
            images/uscs/' + data.uscs[i]
            + '.png';
    }

    rows_sep = 0;
    // Draw images for symbol column
    loadImages(sources, function(images) {

```

```

for (i = 0; i < num_of_samples; i
  ++ ) {
  source_var = 'sample' + (i+1);
  x = start_x + grid_options.
    cols_separation;
  //rows_sep = rows_sep +
    grid_options.
    rows_separation*(data.
    run_length[ctr+1]/0.3);
  //y = rows_sep + grid_options.
    min_depth_y + add_y;
  y = (i * grid_options.
    rows_separation) +
    grid_options.min_depth_y
    + add_y;
  ctx.drawImage(images [
    source_var], x, y,
    grid_options.
    cols_separation,
    grid_options.
    rows_separation);
}
});
ctr = 0;
rows_sep = 0;
for (i = 1; i <= num_of_samples; i++)
{
  // Depth column
  x = start_x + grid_options.
    text_padding;
  rows_sep = rows_sep + grid_options.
    rows_separation*(data.
    run_length[ctr]/0.3);
  y = rows_sep + grid_options.
    min_depth_y + add_y;
  ctx.fillText(data.d_to[ctr], x, y)
  ;

  // Draw horizontal lines
  ctx.moveTo(x - grid_options.
    text_padding, y);
  ctx.lineTo(x - grid_options.
    text_padding + grid_options.
    cols_width, y);

  // USCS column
  x = start_x + 2*grid_options.
    cols_separation +
    grid_options.text_padding;
  y = rows_sep + grid_options.
    min_depth_y + add_y;
  ctx.fillText(data.uscs[ctr], x, y)
  ;

  // SPT N-value column
  x = start_x + 3*grid_options.
    cols_separation;
  //y = (i-1) * grid_options.
    rows_separation +
    grid_options.min_depth_y +
    add_y;
  y = rows_sep + grid_options.
    min_depth_y + add_y -
    grid_options.rows_separation;
  ctx.moveTo(x + data.spt[ctr]*
    grid_options.scale, y);
  ctx.lineTo(x + data.spt[ctr+1]*
    grid_options.scale, y +
    grid_options.rows_separation)
  ;
  //ctx.lineTo(x + data.spt[ctr+1]*
    grid_options.scale, y +
    grid_options.rows_separation
    *(data.run_length[ctr]/0.3));
  ctx.arc(x + data.spt[ctr+1]*
    grid_options.scale, y +
    grid_options.rows_separation,
    5, 0, 2 * Math.PI, true);
  ctx.fillText(data.spt[ctr+1], x +
    data.spt[ctr+1]*grid_options.
    scale + grid_options.
    text_padding, y +
    grid_options.rows_separation)
  ;
  ctr++;
}

// Coring text
ctx.textAlign = "center";
x = start_x + (grid_options.cols_width
  /2);
y = grid_options.rows_end +
  grid_options.coring_text_add_y;
var coring_text = 'Coring_up_to_' +
  data.max_depth + 'm.';
ctx.fillText(coring_text, x, y);

  ctx.stroke();
  ctx.closePath();
  return;
}

function drawCanvasLines(cnv, grid_options
  , bh1_num, bh2_num, min_depth) {
  var ctx = cnv.getContext('2d');

  ctx.font = "13pt_Arial";
  ctx.textAlign = "start";
  ctx.textBaseline = "bottom";
  ctx.strokeStyle = grid_options.
    major_lines_color;
  ctx.beginPath();

  // Draw minimum depth line
  y = grid_options.min_depth_y;
  ctx.moveTo(0, y);
  ctx.lineTo(grid_options.bh1_start_x -
    (2*grid_options.extra_line_x), y)
  ;
  ctx.moveTo(grid_options.bh1_start_x, y
  );
  ctx.lineTo(grid_options.bh2_start_x -
    (2*grid_options.extra_line_x), y)
  ;
  ctx.moveTo(grid_options.bh2_start_x, y
  );
  ctx.lineTo(grid_options.canvas_width,
    y);

  ctx.lineWidth = 3;
  ctx.stroke();
  ctx.closePath();

  ctx.beginPath();

  // Draw footer
  x = grid_options.footer_start_x;
  y = grid_options.footer_start_y;
  ctx.moveTo(x, y);
  ctx.arc(x + 50, y, 50, 0, 2 * Math.PI,
    true);
  ctx.lineTo(x + grid_options.
    footer_width, y);

  if (bh1_num == 0) {
    bh1_num = '';
  }
  if (bh2_num == 0) {
    bh2_num = '';
  }
  var footer_text = 'SOIL_PROFILE_ALONG_
    BH-' + bh1_num + '&_BH-' +
    bh2_num;
  ctx.fillText(footer_text, x +
    grid_options.text_padding + 100,
    y);

  ctx.lineWidth = 2;
  ctx.stroke();
  ctx.closePath();

  return;
}

function loadImages(sources, callback) {
  var images = {};
  var loadedImages = 0;
  var numImages = 0;

  // Get number of sources
  for (var src in sources) {
    numImages++;
  }

  for (var src in sources) {
    images[src] = new Image();
    images[src].onload = function() {
      if(++loadedImages >= numImages)
        {
          callback(images);
        }
    };
    images[src].src = sources[src];
  }
}

var canvas_div = document.getElementById('
  canvas_div');

if (canvas_div !== null) {
  var gridOptions = {
    constants: {
      // General constants
      scale: 230/60, //cols_width /

```

```

        60
        row_scale: 5,
        text_padding: 3,
        canvas_width: 1000, //cnv.
            width
        canvas_height: 1225, //cnv.
            height
        bh1_start_x: 100,
        bh2_start_x: 550,

        // Header constants
        header_row_num: 3,
        header_row_separation: 75/3,
            //table_header_start_y/3

        // Footer constants
        footer_start_x: 300,
        footer_start_y: 1100,
        footer_width: 400,
        coring_text_add_y: 20,

        // Table header and line
        constants
        table_start_y: 150,
        table_header_start_y: 75,
        table_header_width: 75, //
            table_start_y -
            table_header_start_y
        min_depth_y: 160, //
            table_start_y + 10
        extra_line_x: 15,

        // Column constants
        colsa_num: 3,
        colsb_num: 12,
        colsa_width: 120,
        colsb_width: 230,
        colsa_separation: 120/3, //
            colsa_width / colsa_num
        colsb_separation: 230/12, //
            colsb_width / colsb_num
        colsb_header_num: 7,
        colsx_width: 350, //sum of
            colsx_width; table width

        // Row constants
        rows_separation: 120/3, //same
            with colsa_separation
        rows_height: 840, //table body
            height
        rows_end: 160 + 840, //
            min_depth_y + rows_height

        // Color constants
        major_lines_color: '#000000',
        minor_lines_color: '#D3D3D3'
    };
}

y = 0;
for (z = 0; z < num_of_canvas; z++) {
    var bh_canvas_id = 'bh_canvas' + z
    ;
    var cnv = document.getElementById(
        bh_canvas_id);

    var min_depth = Number.
        MAX_SAFE_INTEGER;
    var bh1 = null;
    var bh2 = null;
    var bh1_num = 0;
    var bh2_num = 0;

    // First borehole to be drawn in
    zth canvas;
    if (typeof(boreholes_info[y]) !==
        'undefined') {
        bh1 = boreholes_info[y];
        bh1_num = boreholes_info[y]['
            bh_num'];
        for (i = 0; i < samples_info.
            length; i++) {
            if (samples_info[i]['bh_id
                '] == bh1['id']) {
                min_depth = Math.min(
                    min_depth,
                    samples_info[i]['
                        sample_from']);
            }
        }
    }
    y++;

    // Second borehole to be drawn in
    zth canvas;
    if (typeof(boreholes_info[y]) !==
        'undefined') {
        bh2 = boreholes_info[y];
        bh2_num = boreholes_info[y]['
            bh_num'];
        for (i = 0; i < samples_info.
            length; i++) {
            if (samples_info[i]['bh_id
                '] == bh2['id']) {
                min_depth = Math.min(
                    min_depth,
                    samples_info[i]['
                        sample_from']);
            }
        }
    }
    y++;

    if (min_depth !== Number.
        MAX_SAFE_INTEGER) {
        drawCanvasLines(cnv,
            gridOptions.constants,
            bh1_num, bh2_num,
            min_depth);

        // Draw soil profile
        if (bh1 !== null) {
            drawProfile(cnv,
                gridOptions.constants
                , bh1, 1, min_depth);
        }
        if (bh2 !== null) {
            drawProfile(cnv,
                gridOptions.constants
                , bh2, 2, min_depth);
        }
    }
}

window.onload = function() {
    for (z = 0; z < num_of_canvas; z++) {
        var bh_canvas_id = 'bh_canvas' + z
        ;
        var cnv = document.getElementById(
            bh_canvas_id);

        var img_uri = cnv.toDataURL();
        var proj_id = project['id'];
        var csrf_token = $('meta[name="
            csrf-token"]').attr('content'
            );

        // Save image
        $.ajax({
            type: "POST",
            url: "store-image",
            data: {
                image_uri: img_uri,
                proj_id: proj_id,
                canvas_num: z,
                sb_type: 'SP',
                _token: csrf_token
            },
            success: function(data){
                //alert(data);
            },
            error: function(xhr, error) {
                alert(error.Message);
            }
        });
    }
};
</script>
@endsection

```

Listing 114: subreports/showBL.blade.php

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
    <div class="col-md-12">
        <div class="x_panel">
            <div class="x_title">
                <h2>Borehole Log</h2>
            <ul class="nav navbar-right
                panel_toolbox">
                <li><a class="collapse-
                    link"><i class="fa fa
                    -chevron-up"></i></a>
                </li>
                <li class="dropdown">
                    <a href="#" class="
                        dropdown-toggle"
                        data-toggle="
                            dropdown"
                            role="

```



```

        button" aria-
        expanded=" false"
        ><i class=" fa_fa-
        wrench"></i></a>
    <ul class=" dropdown-
        menu" role=" menu"
        >
        <li><a href="#">
            Settings 1</a>
        </li>
        <li><a href="#">
            Settings 2</a>
        </li>
    </ul>
</li>
</ul>
<div class=" clearfix"></div>
<div class=" x_content">
    <div id=" canvas_div">
        <?php $i = 0 ?>
        @foreach ($boreholes_info
            as $bh)
            <?php $path = " images/
                subreports/BL/bh-$bh
                ->id.png" ?>
            <center></
                center>
            <br />
            <?php $i++ ?>
        @endforeach
    </div>
    <div class=" ln_solid"></div>
</div>
</div>
</div>
@endsection

```

```

        <p><label>Location:</label>
        > {{ $project->
        proj_location }}</p>
    </div>
    <div class=" col-md-4_col-md-4_
        col-xs-12_normal-font"
        style=" margin-left :50px">
        <p><label>Job No. :</label>
        {{ $sample->
        sample_job_num }}</p>
        <p><label>Date Tested:</
        label> {{ $sample->
        sample_test_date }}</
        p>
    </div>
    <?php
        $bh_num = Helper::
        getBoreholeNum(
            $sample->bh_id);
        $table[$ctr] = Helper::
        getDataPoints($sample
            ->id);
        $sample_ids[$ctr] =
            $sample->id;
        json_encode($table[$ctr]);
        ?>
    <center>
        <div id=" chart_div{{ $ctr }}"
            style=" margin-top
            :75px"></div>
    </center>
    <br />
    <center><img id=" chart_img{{
        $ctr }}" /></center>
    <table class=" table_table-
        nonfluid" align=" center"
        border=" 1">
        <thead>
            <tr>
                <th>Borehole No.</
                th>
                <th>Sample No.</th>
                <th>Depth (m)</th>
                <th>Soil
                    Description</
                    th>
                <th>USCS</th>
                <th>NMC (%)</th>
                <th>LL</th>
                <th>PL</th>
                <th>PI</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <th scope=" row">{{
                    $bh_num }}</
                    th>
                <td>{{ $sample->
                    sample_num
                    }}</td>
                <td>{{ $sample->
                    sample_from .
                    "_." .
                    $sample->
                    sample_to
                    }}</td>
                <td>{{ $sample->
                    uscs_description
                    }}</td>
                <td>{{ $sample->
                    uscs_id }}</
                    td>
                <td>{{ $sample->
                    sample_nmc_percent
                    }}</td>
                <td>{{ $sample->
                    sample_liquid_limit
                    }}</td>
                <td>{{ $sample->
                    sample_plastic_limit
                    }}</td>
                <td>{{ $sample->
                    sample_plasticity_index
                    }}</td>
            </tr>
        </tbody>
    </table>
    <div class=" ln_solid"></div>
    <br /><br /><br /><br /><br />
    <?php $ctr++ ?>
    @endforeach
</div>
</div>
</div>
</div>
</div>
@endsection
@section(' js ')

```

Listing 115: subreport-s/showGSA.blade.php

```

@parent
<!--Load the AJAX API-->
<script type="text/javascript" src="https://
www.google.com/jsapi"></script>
<script type="text/javascript">
// Load the Visualization API and the
piechart package.
//google.charts.load('current', {'packages
':['line']});
google.load('visualization', '1.0', {'
packages':['corechart', 'line']});
//google.charts.load('current', {'packages:
['corechart', 'line']});

// Set a callback to run when the Google
Visualization API is loaded.
google.setOnLoadCallback(drawChart);

// Callback that creates and populates a
data table,
// instantiates the pie chart, passes in
the data and
// draws it.
function drawChart() {

var table_dp = <?php echo json_encode(
$table) ?>;
var samples = <?php echo json_encode(
$sample_ids) ?>;
var sample_size = <?php echo sizeof(
$samples) ?>;

for(ctr = 0; ctr < sample_size; ctr++)
{
// Create the data table.
var data = new google.
visualization.DataTable(
table_dp[ctr]);

// Set chart options
var options = {
title: "Grain-Size-Analysis",
//titlePosition: 'none',
width: 900,
height: 400,
hAxis: {
title: 'Grain-Size-(mm)',
scaleType: 'log',
//logScale: 'true',
//ticks: [0.001, 0.01,
0.1, 1, 10, 100]
ticks: [0.001, 0.002,
0.006, 0.01, 0.02,
0.06, 0.1, 0.2, 0.6,
1, 2, 6, 10, 20, 60,
100],
gridlines: { color: '
#000000' }
},
vAxis: {
title: 'Percentage-Smaller
',
ticks: [0, 20, 40, 60, 80,
100],
gridlines: { color: '
#000000' }
},
pointSize: 10,
pointShape: 'circle'
};

var table_id = "chart_div" + ctr;
var table_id_img = "chart_img" +
ctr;
// Instantiate and draw our chart,
passing in some options.
var chart = new google.
visualization.LineChart(
document.getElementById(
table_id));
//var chart = new google.charts.
Line(document.getElementById(
table_id));

google.visualization.events.
addListener(chart, 'ready',
function () {
var img_uri = chart.
getImageURI();
var sample_id = samples[ctr];
var csrf_token = $('meta[name
="csrf-token"]').attr('
content');

$.ajax({
type: "POST",
url: "store-image",
data: {
image_uri: img_uri,

```

```

sample_number:
sample_id,
sb_type: 'GSA',
_token: csrf_token
},
success: function(data){
//alert(data);
},
error: function(xhr, error
) {
alert('Error-saving-
images');
}
});
});
chart.draw(data, options);
}
}
</script>
@endsection

```

Listing 116: subreport- s/showLPTR.blade.php

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
<div class="col-md-12">
<div class="x_panel">
<div class="x_title">
<h2>Liquid and Plastic Limits
Test Report</h2>
<ul class="nav navbar-right
panel_toolbox">
<li><a class="collapse-
link"><i class="fa_fa
-chevron-up"></i></a>
</li>
<li class="dropdown">
<a href="#" class="
dropdown-toggle"
data-toggle="
dropdown" role="
button" aria-
expanded="false"
><i class="fa_fa-
wrench"></i></a>
<ul class="dropdown-
menu" role="menu"
>
<li><a href="#">
Settings 1</a>
</li>
<li><a href="#">
Settings 2</a>
</li>
</ul>
</li>
</ul>
</div>
<div class="clearfix"></div>
</div>
<div class="x_content">
<?php $ctr = 0; $i = 0; ?>
@foreach($samples as $sample)
<div class="col-md-4_col-md-4-
col-xs-12_normal-font">
<p><label>Project:</label>
{{ $project->
proj_name }}</p>
<p><label>Job No.:</label>
{{ $sample->
sample_job_num }}</p>
<p><label>Location:</label>
{{ $project->
proj_location }}</p>
</div>
<div class="col-md-4_col-md-4-
col-xs-12_normal-font">
<p><label>Borehole-Sample
No.:</label> {{
$sample->bh_id }} -
{{ $sample->id }}</p>
<p><label>Contract No.:</
label> {{ $sample->
sample_contract_num
}}</p>
<p><label>Date:</label> {{
$sample->
sample_test_date }}</
p>
</div>

```

```

<div class="col-md-4_col-md-4_col-xs-12_normal-font">
  <p><label>Depth:</label>
    {{ $sample->
      sample_from }} m. -
    {{ $sample->sample_to
      }} m.</p>
  <p><label>Tested By:</
    label> {{ $sample->
      sample_tester }}</p>
</div>
<?php
  $bh_num = Helper::
    getBoreholeNum(
      $sample->bh_id);
  $table[$ctr] = Helper::
    getLimits($sample->id
    );
  $sample_ids[$ctr] =
    $sample->id;
  json_encode($table[$ctr]);
  ?>
<br />
<center>
  <div id="chart_div{{_.$ctr_
    }}" style="margin-top
    :75px"></div>
</center>
<br />
<center><img id="chart_img{{_
  $ctr_}}" /></center>
<table class="table-table-
  nonfluid" align="center"
  border="1">
  <thead>
    <tr>
      <td colspan="3"
        style="text-align:center">
        >Liquid Limit
        Tests</td>
      <td colspan="2"
        style="text-align:center">
        >Plastic
        Limit Tests</
        td>
    </tr>
    <tr>
      <td style="text-align:center">
        >Run No.</td>
      <td style="text-align:center">
        >No. of Blows
        </td>
      <td style="text-align:center">
        >Water
        Content</td>
      <td style="text-align:center">
        >Run No.</td>
      <td style="text-align:center">
        >Water
        Content</td>
    </tr>
  </thead>
  <tbody>
    <!-- assumes that
      sizeof
      plastic_limits =<
      sizeof
      liquid_limits -->
    @foreach(
      $liquid_limits as
      $liquid_limit)
    @if($liquid_limit->
      sample_id ==
      $sample->id)
    <tr>
      <th scope="row">{{
        $liquid_limit
        ->lim_run_num
        }}</th>
      <td>{{
        $liquid_limit
        ->
        lim_num_of_blows
        }}</td>
      <td>{{
        $liquid_limit
        ->
        lim_water_content
        }}</td>
    @if(sizeof(
      $plastic_limits
    ) > $i &&
      $plastic_limits
      [$i]->
      sample_id ==
      $sample->id)
    <th scope="row">{{
      $plastic_limits
      [$i]->
      lim_run_num
      }}</th>
    <td>{{
      $plastic_limits
      [$i]->
      lim_water_content
      }}</td>
    <?php $i++ ?>
    @else
    <td></td>
    <td></td>
    @endif
  </tr>
  @endif
  @endforeach
</tbody>
</table>
<table class="table-table-
  nonfluid" align="center"
  border="1">
  <tbody>
    <tr>
      <td>Liquid Limit
      (%) = {{
        $sample->
        sample_liquid_limit
        }}</td>
      <td>Plastic Limit
      (%) = {{
        $sample->
        sample_plastic_limit
        }}</td>
      <td>Plasticity
      Index = {{
        $sample->
        sample_plasticity_index
        }}</td>
    </tr>
    <tr>
      <td colspan="3">
        Remarks: {{
          $sample->
          sample_plasticity_remark
          }}</td>
    </tr>
  </tbody>
</table>
<div class="ln_solid"></div>
<br /><br /><br /><br />
<?php $ctr++ ?>
  @endforeach
</div>
</div>
@endsection
@section('js')
@parent
<!--Load the AJAX API-->
<script type="text/javascript" src="https://
  www.google.com/jsapi"></script>
<script type="text/javascript">
  // Load the Visualization API and the
  piechart package.
  //google.charts.load('current', {'packages
  ':'line'});
  google.load('visualization', '1.0', {'
  packages':['corechart', 'line']});
  //google.charts.load('current', {'packages:
  ['corechart', 'line']});

  // Set a callback to run when the Google
  Visualization API is loaded.
  google.setOnLoadCallback(drawChart);

  // Callback that creates and populates a
  data table,
  // instantiates the pie chart, passes in
  the data and
  // draws it.
  function drawChart() {
    var table_dp = <?php echo json_encode(
      $table) ?>;
    var samples = <?php echo json_encode(
      $sample_ids) ?>;
    var sample_size = <?php echo sizeof(
      $samples) ?>;
    for(ctr = 0; ctr < sample_size; ctr++)

```

```

    {
    // Create the data table.
    var data = new google.
        visualization.DataTable(
            table_dp[ctr]);

    // Set chart options
    var options = {
        title: "Liquid_and_Plastic_
            Limits_Test_Report",
        //titlePosition: 'none',
        width: 900,
        height: 400,
        hAxis: {
            title: 'No._of_Blows',
            scaleType: 'log',
            ticks: [3, 5, 10, 15, 20,
                30, 40, 50],
            gridlines: { color: '
                #000000' }
        },
        vAxis: {
            title: 'Water_Content_(%)',
            gridlines: { color: '
                #000000' }
        },
        pointSize: 10,
        pointShape: 'circle'
    };

    var table_id = "chart_div" + ctr;
    var table_id_img = "chart_img" +
        ctr;
    // Instantiate and draw our chart,
    // passing in some options.
    var chart = new google.
        visualization.LineChart(
            document.getElementById(
                table_id));
    //var chart = new google.charts.
        Line(document.getElementById(
            table_id));

    google.visualization.events.
        addListener(chart, 'ready',
            function () {
                var img_uri = chart.
                    getImageURI();
                var sample_id = samples[ctr];
                var csrf_token = $('meta[name
                    ="csrf-token"]').attr('
                    content');

                $.ajax({
                    type: "POST",
                    url: "store-image",
                    data: {
                        image_uri: img_uri,
                        sample_number:
                            sample_id,
                        sb_type: 'LPTR',
                        _token: csrf_token
                    },
                    success: function(data){
                        //alert(data);
                    },
                    error: function(xhr, error
                        ) {
                            alert('Error_saving_
                                images');
                        }
                });
            });

    chart.draw(data, options);
    }
}
</script>
@endsection

```

Listing 117: subreport-s/showSP.blade.php

```

@extends('layouts/master')
@section('title', $project->proj_name)
@section('content')
<div class="row">
    <div class="col-md-12">
        <div class="x_panel">
            <div class="x_title">
                <h2>Soil Profile </h2>
            <ul class="nav navbar-right_
                panel_toolbox">
                <li><a class="collapse-

```

```

                link"><i class="fa_fa
                -chevron-up"></i></a>
            </li>
            <li class="dropdown">
                <a href="#" class="
                dropdown-toggle"
                data-toggle="
                dropdown" role="
                button" aria-
                expanded="false"
                ><i class="fa_fa-
                wrench"></i></a>
            <ul class="dropdown-
                menu" role="menu"
                >
                <li><a href="#">
                    Settings 1</a>
                </li>
                <li><a href="#">
                    Settings 2</a>
                </li>
            </ul>
            </li>
        </ul>
    </div>
<div class="clearfix"></div>
</div>
<div class="x_content">
    <div id="canvas_div">
        <?php
            $num_of_bh = $project
                ->
                proj_num_of_boreholes
                ;
            $num_of_canvas = ceil(
                $num_of_bh/2);
            ?>
            @for($i = 0; $i <
                $num_of_canvas; $i++)
            <?php $path = "images/
                subreports/SP/proj-
                $project->id-$i.png"
                ?>
            <center></
                center>
            <br />
            @endfor
        </div>
    <div class="ln_solid"></div>
</div>
</div>
</div>
@endsection

```

Listing 118: users/create.blade.php

```

@extends('layouts/master')
@section('title', $user->firstname . '_' .
    $user->lastname)
@section('content')
<div class="row">
    <div class="col-md-12_col-sm-12_col-xs-12">
        <div class="x_panel">
            <div class="x_title">
                <h2> Edit Profile </h2>
            <ul class="nav navbar-right_
                panel_toolbox">
                <li><a class="collapse-
                link"><i class="fa_fa
                -chevron-up"></i></a>
            </li>
            <li class="dropdown">
                <a href="#" class="
                dropdown-toggle"
                data-toggle="
                dropdown" role="
                button" aria-
                expanded="false"
                ><i class="fa_fa-
                wrench"></i></a>
            <ul class="dropdown-
                menu" role="menu"
                >
                <li><a href="#">
                    Settings 1</a>
                </li>
                <li><a href="#">
                    Settings 2</a>
                </li>
            </ul>
            </li>
        </ul>
    </div>

```

```

        </ul>
      </li>
    </ul>
  </div class="clearfix"></div>
</div>
<div class="x_content">
  <div class="container">
    {!! Form::open(['name' =>
      'update_form',
      'method' => 'PUT',
      'route' => ['users.update', $user->id],
      'class' => 'form-horizontal_form-label-left',
      'id' => 'update_form']) !!}
    <div class="form-group">
      {!! Form::label('first_name', 'First Name',
        array('class' => 'control-label_col-md-3_col-sm-3_col-xs-12')) !!}
    <div class="col-md-6_col-sm-6_col-xs-12">
      <?php $firstname =
        Helper::
          getValue(old('first_name'),
            $user->
              firstname) ?>
      {!! Form::text('first_name',
        $firstname, [
          'class' => 'form-control_col-md-7_col-xs-12',
          'required' => 'true']) !!}
    </div>
  </div>
  <div class="form-group">
    {!! Form::label('middle_name', 'Middle Name',
      array('class' => 'control-label_col-md-3_col-sm-3_col-xs-12')) !!}
    <div class="col-md-6_col-sm-6_col-xs-12">
      <?php $middlename =
        Helper::
          getValue(old('middle_name'),
            $user->
              middlename) ?>
      {!! Form::text('middle_name',
        $middlename, [
          'class' => 'form-control_col-md-7_col-xs-12']) !!}
    </div>
  </div>
  <div class="form-group">
    {!! Form::label('last_name', 'Last Name',
      array('class' => 'control-label_col-md-3_col-sm-3_col-xs-12')) !!}
    <div class="col-md-6_col-sm-6_col-xs-12">
      <?php $lastname =
        Helper::
          getValue(old('last_name'),
            $user->
              lastname) ?>
      {!! Form::text('last_name',
        $lastname, [
          'class' => 'form-control_col-md-7_col-xs-12',
          'required' => 'true']) !!}
    </div>
  </div>
</div>

```

```

<div class="form-group">
  {!! Form::label('email', 'Email',
    array('class' => 'control-label_col-md-3_col-sm-3_col-xs-12')) !!}
  <div class="col-md-6_col-sm-6_col-xs-12">
    <?php $email =
      Helper::
        getValue(old('email'),
          $user->email) ?>
    {!! Form::text('email',
      $email, [
        'class' => 'form-control_col-md-7_col-xs-12',
        'required' => 'true']) !!}
  </div>
</div>
<div class="form-group">
  {!! Form::label('contact', 'Contact Number',
    array('class' => 'control-label_col-md-3_col-sm-3_col-xs-12')) !!}
  <div class="col-md-6_col-sm-6_col-xs-12">
    <?php $contact =
      Helper::
        getValue(old('contact'),
          $user->
            contact) ?>
    {!! Form::text('contact',
      $contact, [
        'class' => 'form-control_col-md-7_col-xs-12',
        'required' => 'true']) !!}
  </div>
</div>
<div class="form-group">
  {!! Form::label('sex', 'Sex',
    array('class' => 'control-label_col-md-3_col-sm-3_col-xs-12')) !!}
  <div class="radio">
    <?php $sex =
      Helper::
        getValue(old('sex'),
          $user->sex) ?>
    <label >{!! Form::radio('sex',
      'M', ($sex == 'M')?true:
      false, [
        'class' => 'flat']); !!}
    Male</label>
    <label >{!! Form::radio('sex',
      'F', ($sex == 'F')?true:
      false, [
        'class' => 'flat']); !!}
    Female</label>
  </div>
</div>
<div class="form-group">
  {!! Form::label('birthdate', 'Birthdate',
    array('class' => 'control-label_col-md-3_col-sm-3_col-xs-12')) !!}
  <div class="col-md-6_col-sm-6_col-xs-12">

```



```

        i
      >
      Contact
    no
    ..
    {{
      $user
    ->
    contact
    }}
  </li
  >
</ul>
</div>
<div class="
  right_col
  -xs-5_
  text-
  center">
  
  </div>
</div>
<div class="col-xs
  -12_bottom_
  text-center">
  <div class="
    col-xs-12
    _col-sm-6
    _emphasis
    ">
  </div>
  <div class="
    col-xs-12
    _col-sm-6
    _emphasis
    ">
    <center>
      <table
        border
        =
        "
        0
        "
        style
        =
        "
        margin
        -
        right
        :-25
        px
        ">
      <
        tr
        >
      <
        td
        >
        {!!
          Form
          ::
          open
          ([
            ,
            name
            ,
            =>
          Form
          ::

```

```

        close
        ()
        {!!}
    </td
    >
    <td
    >
    {!!
    =>
        Form
        ::
        open
        (
        {
        ,
        method
        ,
        =>
        ,
        DELETE
        ,
        ,
        route
        ,
        =>
        =>
        [
        ,
        users
        ,
        destroy
        ,
        ,
        $user
        ->
        id
        ],
        ,
        onsubmit
        ,
        =>
        =>
        ,
        return
        -
        confirm
        (
        "
        Reject
        -
        pending
        -
        user
        ,
        .
        </div>
        @endif
        @endforeach
        $user_name
        @if(!$hasPending)
        .
        <center>
        ?
        <h2>There are no
        -
        pending users.</
        -
        h2>
        This
        </center>
        -
        @endif
        user
        </div>
        -
        <div class="ln_solid"></div>
        will</div>
        </div>
    </div>
    </div>
    <!-- /pendingusers -->
    <!-- members -->
    <div class="row">
    <div class="col-md-12">
    <div class="x_panel">
    <div class="x_title">
    database<h2>Members</h2>
    .")
    <ul class="nav navbar-right
    -
    panel_toolbox">
    <li><a class="collapse-
    -
    link"><i class="fa_fa
    -
    chevron-up"></i></a>
    {!!}
    </li>
    {!!
    <li class="dropdown">
    <a href="#" class="
    Form
    dropdown-toggle"

```



```

        data-toggle="
        dropdown" role="
        button" aria-
        expanded="false"
    ><i class="fa_fa-
    wrench"></i></a>
<ul class="dropdown-
    menu" role="menu"
    >
    <li><a href="#">
        Settings 1</a>
    </li>
    <li><a href="#">
        Settings 2</a>
    </li>
</ul>
</li>
</ul>
<div class="clearfix"></div>
<div class="x_content">
    <div class="row">
        @foreach($users as $user)
        @if($user->status == 1)
        <div class="col-md-4_col-
        sm-4_col-xs-12_
        animated_fadeInDown">
            <div class="well_
            profile_view">
                <div class="col-sm
                -12">
                    <h4 class="
                    brief">
                        <i> {{
                            $user
                            ->
                            position
                        }}</i>
                    <button
                    type=
                    "
                    button
                    "
                    class
                    =
                    "
                    open-
                    modal_
                    _btn_
                    btn_
                    link_
                    btn_
                    xs"
                    data-
                    toggle
                    =
                    "
                    modal
                    "
                    data-
                    target
                    =
                    ".bs
                    -
                    example
                    -
                    modal
                    -sm"
                    data-
                    id="
                    {{
                        $user
                        ->id_
                    }}">
                        <i
                        class
                        =
                        "fa_
                        fa_
                        pencil
                        "></i>
                    >
                    Change
                    </
                    button
                    >
                </h4>
                <div class="
                left_col-
                xs-7">
                    <a href="
                    {{
                        _
                        route
                        (
                        '
                        users
                        .show
                        '
                        ,
                        $user
                        ->id)
                    }}">

```



```

Form
::
hidden
(
,
user_id
,
,
$user
->
id
)
!!}
{!!
Form
::
submit
(
,
Disable
,
,
[
,
class
,
=>
,
,
btn
_
btn
_
danger
_
btn
_
xs
,
,
,
style
,
=>
,
width
:60
px
])
!!}
{!!
Form
::
close
()
!!}
</
td
>
</
tr
>
</
table
>
</center>
</div>
</div>
</div>
</div>
</div>
</div>
@endif
@endforeach
<div class="modal_fade_bs-
example-modal-sm"
tabindex="-1" role="
dialog" aria-hidden="
true">
<div class="modal-
dialog_modal-sm">
<div class="modal-
content">
<div class="
Form
::
hidden
(
,
user_id
,
,
$user
->
id
)
!!}
{!!
Form
::
submit
(
,
Disable
,
,
[
,
class
,
=>
,
,
btn
_
btn
_
danger
_
btn
_
xs
,
,
,
style
,
=>
,
width
:60
px
])
!!}
{!!
Form
::
close
()
!!}
</
td
>
</
tr
>
</
table
>
</center>
</div>
</div>
</div>
</div>
</div>
</div>
@endif
@endforeach
<div class="modal_fade_bs-
example-modal-sm"
tabindex="-1" role="
dialog" aria-hidden="
true">
<div class="modal-
dialog_modal-sm">
<div class="modal-
content">
<div class="
modal-
header">
<button
type=
"
button
"
class
=
"
close
"
data-
dismiss
=
"
modal
"
aria-
label
=
"
Close
"><
span
aria-
hidden
=
"
true"
> </
span
></
button
>
<h4 class=
"
modal
_
title
" id=
"
"
myModalLabel2
">
Change
user
position
</h4>
</div>
<div class="
modal-
body">
{!! Form::
open
(['
name'
=>
'update_form'
,
,
method'
=>
'POST'
,
,
route'
=>
'users
_
changeUserPosition'
,
,
class'
=>
'form
_
horizontal
_form
_
label
_left'
])
!!}
{!! Form::
text(
,
position
,
old('
position'
), [
,
class'
=>
'form
_
control
_col-
md-2'
,
,
placeholder'
=>
'New_
user_

```

```

position                                     button
',',                                         >
required                                     {!!
',=>                                        Form
',true                                       ::
',))                                         close
{!!}                                         ()
{!! Form::                                   {!!}
hidden                                     ( 'id'
',
null,                                       </div>
['id',                                     </div>
',=>                                       </div>
',
user_id                                     </div>
',))                                       <div class="ln-solid"></div>
{!!}                                       </div>
<div class                                 </div>
="
modal </div>
- <!-- /members -->
footer <!-- disabled users -->
"> <div class="row">
{!! <div class="col-md-12">
Form <div class="x_panel">
:: <div class="x_title">
submit <h2>Disabled Users</h2>
( <ul class="nav navbar-right
Submit <li><a class="collapse-
panel_toolbox">
link"><i class="fa fa
-chevron-up"></i></a>
', </li>
[ <li class="dropdown">
class <a href="#" class="
', dropdown-toggle"
=> data-toggle="
', button" aria-
', expanded="false"
btn <i class="fa fa-
_ wrench"></i></a>
btn <ul class="dropdown-
primary <li><a href="#">
', <div class="clearfix"></div>
', </li>
margin <li><a href="#">
- </li>
top <div class="x_content">
: <div class="row">
_ <?php $hasDisabled = false
5 ; ?>
px @foreach($users as $user)
; @if($user->status == 3)
; <?php $hasDisabled = true;
] ?>
!!} <div class="col-md-4 col-
sm-4 col-xs-12
animated_fadeInDown">
<div class="well_
button profile_view">
type <div class="col-sm
= -12">
" <h4 class="
button brief"><i
" > {{
class $user->
= position
_ }} </i></
_ h4>
btn <div class="
_ left_col-
_ xs-7">
default <h2><i
" class
= " fa
fa-
user"
data ></i>
_ dismiss {{
= $user_name
= $user
modal ->
" firstname
" . "_
" .
Close $user
</

```

```

->
  lastname
  }}
</h2>
<ul class=
" list
-
unstyled
">
<li><i>
  class
  =
  "
  fa
  -
  fa
  -
  envelope
  "
  ></
  i
  >
  Email
  :
  {{
  $user
  ->
  email
  }}
  </
  li
  >
<li><i>
  class
  =
  "
  fa
  -
  fa
  -
  phone
  "
  ></
  i
  >
  Contact
  no
  .:
  {{
  $user
  ->
  contact
  }}
  </
  li
  >
</ul>
</div>
<div class="
right_col
-xs-5_
text-
center">

  alt=
  ""
  class
  ="img
  -
  circle
  _img-
  responsive
  ">
</div>
<div class="col-xs
-12_bottom_
text-center">
<div class="
col-xs-12
_col-sm-6
_emphasis
">
</div>
<div class="
col-xs-12
_col-sm-6
_emphasis
">
<center>
  <table
    border
    =
    "
    0
    "
    style
    =
    "
    margin
    -
    right
    :-25
    px
    ">
    <tr
    >
    <td
    >
    <a
    href
    =
    "
    {{
    _
    route
    (
    '
    users
    .
    show
    '
    ,
    _
    $user
    ->
    id
    )
    _
    }}
    "
    class
    =
    "
    btn
    -
    btn
    -
    primary
    _
    btn
    -
    xs
    "
    style
    =
    "
    width
    :50
    px
    ;
    ">
    </i
    >
    class
    =
    "
    "
    fa
    -
    fa
    -
    user
    "
    ></
    i
    >

```



```

class
=
"
modal
-
body
"
>
<
  div
    class
    =
    "
    avatar
    -
    body
    "
    >
    <!--
      Upload
      image
      and
      data
      -->
    <
      div
        class
        =
        "
        avatar
        -
        upload
        "
        >
        <
          input
            class
            =
            "
            avatar
            -
            src
            "
            name
            =
            "
            avatar_src
            "
            type
            =
            "
            hidden
            "
            >
        <
          input
            class
            =
            "
            avatar
            -
            data
            "
            name
            =
            "
            avatar_data
            "
            type
            =
            "
            hidden
            "
            >
        <
          label
            for
            =
            "
            av
            "
            >
            Lo
            up
            </
            la
            >
            in:
            <
            cl
            =
            "
            av
            -
            in:
            "
            id
            =
            "
            av
            "
            na
            =
            "
            av
            "
            ty
            =
            "
            fi
            "
            >
            </
            div
            >
            <!--
            Crop
            and
            preview
            -->
            <
            div
            class
            =
            "
            row
            "
            >
            <
            di
            cl
            =
            "
            co
            -
            md
            -
            9
            "
            >
            <
            di
            >
            di
            cl

```



```

=
"
col
-
md
-3
"
>
<
div
class
=
"
avatar
-
preview
-
preview
-
lg
"
></
div
>
<
div
class
=
"
avatar
-
preview
-
preview
-
md
"
></
div
>
<
div
class
=
"
avatar
-
preview
-
preview
-
sm
"
></
div
>
</
div
>
</
div
>
<
div
class
=
"
row
-
avatar
-
btns
"
>
<
div
class
=
"
col
-
md
-9
"
>
<

```

```

        </
        button
        >
    <
        button
        class
        =
        "
        btn
        -
        btn
        -
        primary
        "

        data
        -
        method
        =
        "
        rotate
        "

        data
        -
        option
        =
        "
        -30
        "

        type
        =
        "
        button
        "
        >-30
        deg
        </
        button
        >
    <
        button
        class
        =
        "
        btn
        -
        btn
        -
        primary
        "

        data
        -
        method
        =
        "
        rotate
        "

        data
        -
        option
        =
        "
        -45
        "

        type
        =
        "
        button
        "
        >-45
        deg
        </
        button
        >
</
div
>
<
div
class
=
"
btn
-
group
"
>

```

```

        btn
        -
        btn
        -
        primary
        "

        data
        -
        method
        =
        "
        rotate
        "

        data
        -
        option
        =
        "
        30
        "

        type
        =
        "
        button
        "
        >30
        deg
        </
        button
        >

<
        button
        =
        class
        "
        btn
        -
        btn
        -
        primary
        "
        </form
        >

        data
        -
        method
        =
        "
        rotate
        "
        data
        -
        option
        =
        "
        45
        "
        </div>
        </div>
        </div>
        <!-- /modal -->
        <!-- Loading state -->
        <div class="
        loading" aria
        -label="
        Loading" role
        ="img"
        tabindex="-1"
        ></div>
        45 <!-- /image cropping -->
        </div>
        </div>
        </div>
        @if(Auth::user()->id ==
        $user->id)
        <button href="{route('users.
        edit', $user->id)}"
        >45 class="btn btn-info
        btn-sm"><i class="fa
        -fa-edit_m-right-xs"
        >
        </div>
        </div>
        </div>
        </div>
        </div>
        <!-- /profile image -->
        <!-- user info -->
        <div class="col-md-9_col-sm-9_
        col-xs-12">
        <div class="" role="
        tabpanel" data-
        example-id="toggleable
        -tabs">
        <div
        <ul id="myTab" class="
        nav.nav-tabs_
        bar_tabs" role="
        tablist">
        <li role="
        presentation"
        class="
        active"><a
        href="#
        tab_content1"
        role="tab"
        id="home-tab"
        >
        <div class="
        data-toggle=
        "tab" aria-
        expanded="

```


XI. Acknowledgement

I want to express my sincerest gratitude to the people who supported and helped me throughout this major requirement.

First of all, I would like to thank my parents, *Mommy* and *Daddy* for the overwhelming support especially during the most difficult times in my academic life. Thank you for providing me all my needs, for understanding my limitations and for loving me as I am. All my achievements are for you. I love you.

To my siblings, *Adrian*, thank you for your corny jokes as well as for encouraging me especially before my SP defense; *Princess*, thank you for all the sister bonding moments and for taking care of Dampy when I am not home; and *Andrei*, thank you for your tight hugs and for your 'kakulitan' (home will not be as lively without it).

To *Ma'am Sheila*, thank you for guiding me throughout this project, for encouraging me and for pushing me to do my best even when sometimes I think I cannot do it.

To *Sir Marvin John Ignacio*, thank you for giving us advices and tips on how to survive ComSci life in UPM. Thank you for being a responsible and caring professor to all of us. I salute you.

To the *DPSM faculty* and to my professors, instructors, and teachers, thank you for sharing your knowledge. I have learned so much from all of you.

To *Evangeline Carandang* and *Mark Nagrampa*, thank you for lifting my spirits up especially during the past few weeks. Thank you for listening to my problems and making me feel better.

To *Aaron Custodio*, thank you for being my mentor (master!) and for encouraging me so much when I was doubting myself. I will always remember the things you have taught me.

To *Joseph Tuazon*, *Krizia Ilaga*, *Rafael Ferrer*, *Ruahden Dang-awan*, *Nathan dela Rosa Reinier Maristela*, *Cristine Caedo*, and all my other college friends who, at one point or another, listened to my troubles, gave me advices and helped me

feel better, thank you.

To *Block 12 Hotties*, thank you for being a wonderful group of friends. My college life is happier because of you all.

To *my elementary and high school friends*, I know we don't see each other often but thank you for always making plans. I miss you all so much. Thank you for all the memories.

To *Dampy*, thank you so much for being so cute and for making me happy everytime I look at you. I miss you.

To *Little Mix, Parachute, and One Direction*, whose songs I've been listening to during my stressful days and nights, thank you.

To all the others I forgot to mention, I'm sorry and thank you for being a part of my life.

And last but not the least, I express my gratitude to our *God Almighty*. Thank You for always being there especially during the toughest times. Everything would not be possible without You.