

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

Dee Hwa Liong Academy Virtual Classroom

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

Submitted by:

Marvin Sherwin O. Tejada

June 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 “Dee Hwa Liong Academy Virtual Classroom” prepared and submitted by Marvin Sherwin O. Tejada in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

Gregorio B. Baes, Ph.D. (candidate)
Adviser

EXAMINERS:

	Approved	Disapproved
1. Avegail D. Carpio, M.S.	_____	_____
2. Richard Bryann L. Chua, Ph.D. (candidate)	_____	_____
3. Perlita E. Gasmien, M.S. (candidate)	_____	_____
4. Marvin John C. Ignacio, M.S. (candidate)	_____	_____
5. Ma. Sheila A. Magboo, M.Sc.	_____	_____
6. Vincent Peter C. Magboo, M.D., M.S.	_____	_____

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

Ma. Sheila A. Magboo, M.Sc.
Unit Head
Mathematical Computing Science Unit
Department of Physical Sciences
and Mathematics

Marcelina B. Lizaran, Ph.D.
Chair
Department of Physical Sciences
and Mathematics

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

Abstract

The Dee Hwa Liong Academy Virtual Classroom is an online (although also accessible offline) student and faculty learning portal. The aim of the system is to create a bridge between students and faculty whenever direct confrontation is not applicable. The system also patterns existing learning portals such as Moodle, Genyo, and Blackboard, but is customized to suit the requirements of the client, Dee Hwa Liong Academy.

Keywords: Virtual classroom, learning portals, Dee Hwa Liong Academy, examination creator

Table of Contents

Acceptance Sheet	i
Abstract	ii
I. Introduction	1
A. Background of the Study.....	1
B. Statement of the Problem.....	3
C. Objectives of the Study.....	4
D. Significance of the Study.....	6
E. Scope and Limitations.....	7
F. Assumptions.....	7
II. Review of Related Literature	8
III. Theoretical Framework	14
A. Dee Hwa Liong Academy.....	14
B. Information System.....	14
C. Database Management System.....	15
D. Laravel.....	15
E. Scripting Language.....	16
IV. Design and Implementation	17
A. Entity-Relationship Diagram.....	17
B. Data Flow.....	21
C. Data Dictionary.....	26
D. System Architecture.....	40

E.	Technical Architecture.....	42
V.	Results	43
VI.	Discussion	59
VII.	Conclusions	60
VIII.	Recommendations	62
IX.	Bibliography	63
X.	Appendix	65
A.	Source Code.....	65
XI.	Acknowledgement	

I. Introduction

A. Background of the Study

Because of the level of technology today, people are no longer limited to just physical contact in terms of communication. The internet has been successful in making its way into the lives of the public and in being an effective medium for communication. The success of numerous social networking sites attests to how effective using the internet as medium for communication can be.

One of the greatest perks of communicating online is that one does not need to be near someone to talk to another person. People can be miles apart, yet they can share information. People do not even need being online at the same time to share information. With today's technology, storing data in the cloud has become more and more convenient. Many people are utilizing this technology for their data management.

The education sector is adapting to these said advancements in technology. Online learning portals are now being developed across the whole world. Online learning is defined as courses in which at least 80 percent of the course content is delivered online [1]. Basically, online learning portals act as secondary, although virtual, classrooms to students.

Learning experience does not degrade just because the students and the teachers do not meet eye to eye. In a study done by Anna Yan Ni, she concluded that student performance is independent of the mode of instruction. In fact, quality and quantity of interaction is said to be better in online classes [2]. The main reason why online learning is effective as a substitute to traditional learning is because of increasing student access [3]. Students who may have missed

their class can keep up with the rest of the class provided that the teachers upload lesson readings that the student may use.

Dee Hwa Liong Academy is named after its founder, Mr. Cao Pek. Mr. Cao Pek founded the Deeco Electronics Manufacturing, a manufacturing company that specializes in assembly and manufacture of transformers, battery chargers, welding transformers, and public address systems, on February 2, 1948.

To date, over 80 commercial radio stations and 12 telephone exchanges are operating with equipment supplied by DEECO. Behind this wide range of broadcast and telecommunications equipment are fully established enterprises Deeco Electronics Manufacturing, and Dee Hwa Liong Electronics Equipment Corporation.

In 1983, Mr. Cao Pek “resigned” from active participation in DEECO’s affairs and devoted all his time to philanthropic work. He took over the Principalship of the 200-student Cheng Kuang School and transformed it into the 1000-student Cheng Kuang High School, with two new 3-storey building erected under his supervision.

As of now, DHLA is currently offering courses from kindergarten to high school education. Since its establishment, most of DHLA’s processes are performed manually, including but not limited to the process of students taking examinations. The school also has no online portal in use and relies solely on emails and their Facebook page for feedback. [4]

B. Statement of the Problem

Dee Hwa Liong has been doing all of their examinations manually. Since an exam is answered through handwriting, there may be some answers that are illegible to the checker. This may contribute to an error on the checker's part. It should also be noted that checking the examinations take up much time and effort. With the volume of students a faculty member in Dee Hwa Liong is teaching, checking all of the exams would take days. After the process of checking the exams, the teacher would still need to record all of the scores of the students he/she handles and this process will also take time and effort.

Another platform that Dee Hwa Liong Academy needs is an online portal. With the upcoming implementation of the K-12 program, DHLA's volume of students is expected to increase. It would be harder to keep up with the increasing amount of students. There could be difficulty in handling all the requirements being submitted by all those students.

Although there are many existing online learning portals, they all cater to the specifications submitted by their clients. DHLA's processes certainly differ from those of other educational institutions. Aside from conforming to DHLA's records, the system must also comply with the upcoming K-12 grading system.

It is because of these reasons that a new system will be created to cater to Dee Hwa Liong Academy's specifications. The said system should be able to speed up the process of taking examinations because checking of exams will be automated, and human intervention will be minimized making the examinations more reliable. The system would also create a way for teachers and students to communicate more easily and make submission of requirements easier.

C. Objectives of the Study

The aim of this study is to create a web-based learning platform that can process examinations online as well as provide a virtual classroom. This study aims to create a user-friendly interface that can provide the users ease of access while covering much of what happens in a real classroom.

The specific functionalities of this study are as follows:

1. The faculty can do the following:
 - a. Login and logout of the system
 - b. Change his/her password
 - c. View all classes he/she is handling in a given school year
 - d. Create new categories, topics, and subtopics for questions
 - e. Create questions and add it in the question bank
 - f. Create a post in a given class that will be viewable by all students in the class
 - g. Create a requirement in a given class that will be viewable by all students in the class
 - h. Create an examination by giving the parameters of the examination, including the time interval when the examination would be accessible to the students
 - i. View the submissions of the students in the class
 - j. View the results of the examinations of students in the class
 - k. Upload files that can be downloaded by his/her students
 - l. View the list of users who have downloaded an uploaded file

- m. View the class record of the students in the class
2. The students can do the following:
- a. Login and logout of the system
 - b. Change his/her password
 - c. View the list of classes he/she is enrolled to
 - d. View the posts made by the faculty and the system administrator
 - e. View the requirements made by the faculty
 - f. Submit a file in response to a requirement
 - g. Answer an examination the faculty has prepared
3. The school management can do the following:
- a. Login and logout of the system
 - b. Change his/her password
 - c. View site statistics
4. The system administrator can do the following:
- a. Login and logout of the system
 - b. Change his/her password
 - c. Create new users of the role: faculty, student, school management
 - d. Create a new system administrator, which will lead to the current system administrator's account to be disabled.
 - e. Create new sections
 - f. Create new school year
 - g. Enlist students in a grade section
 - h. Specify what subjects a given grade section would take

- i. Assign a faculty who will be handling a subject of a particular grade section
- j. Edit other users' information or disable their account completely

D. Significance of the Study

The system would be able to produce examinations that is composed of multiple choice, fill in the blanks, true or false, and matching type (which are classified as “match column a with column b” and “select from the wordbox”) formats. Automated checking of the answers of the students would make the process of examination checking faster. The system would produce exams randomly by picking a group of questions from a given question bank. This would help in creating multiple exams, significantly lowering the chances of a student cheating. The scores of the students will also be automatically stored in the class record of the teacher. The system also uses a user-friendly interface that will help the teachers adjust easily to the new implementation of examinations.

The online portal can act as a site where all the subject related files can be stored. Just like most online learning systems, this will act as a virtual dropbox for the faculty and students. The system is also designed in such a way that user without the proper permission cannot access certain files. The portal can also act as news feed from which users can see updates posted by others.

The system would also keep track of all the downloads and uploads being made. Every time a file was either downloaded or uploaded, the system would log the date and time of that process.

The system is expected to greatly reduce the time and effort used in creating, checking, and recording exams. This will also help students in having a centralized portal to access subject related files from their different subjects.

E. Scope and Limitations

1. Although the chances are very slim, the possibility of two students getting exactly the same exam is possible.
2. The types of questions are limited to the following: multiple choice, true or false, select from the wordbox, match column a with column b, and fill in the blanks. Other types of exams are excluded from the said system.
3. The system is not responsible for any misspelled or incorrectly written questions by teachers, nor is the system responsible for any incorrect questions made by the faculty.
4. The file upload size limit is 16mb, whether the user is a faculty or a student.
5. The maximum size of images that the faculty can attach to questions is 200x200 pixels.

F. Assumptions

1. Teachers creating the examinations should be given proper training on how to construct exam questions.
2. The faculty will be responsible enough in creating questions, and will not cause actions that may affect the generation of the exams.
3. The system time and date should be accurate.

II. Review of Related Literature

The implementation of K-12 in several school systems has been popular all over the world for the past decade. This system is still being adapted by many other academic learning institutions all over the world, and the Philippines is not an exception. For a long time, the Philippines has been using a 10-year public educational system. As a result, the Philippines, along with three other countries in the whole world, have been at a disadvantage especially when it came to working in other countries. The Philippines is generally looked down upon by countries implementing a 12-year public educational system.

Dr. Calderon described our educational system as congested, as each student's time to learn pre-college courses are pushed in 10 years rather than 12 years. A biggest problem this brings up is the fact that most high school students graduate without being fully taught on the basics to move on to a higher educational level. This also makes high school graduates less competent at work.

Dr. Calderon stated the importance of K-12 system being implemented in our country. As she stated on her paper, "the K to 12 is aimed at addressing the deficiency of the Philippine educational system particularly in the basic education, the elementary and high school, in order to meet the standards of the international education criteria and for the students to be at par with the students in neighboring countries." [5] Of course, all these numbers may vary, from each area in the country, but in general these facts point out that a reform is much needed if we wish to compete with other countries globally.

A paper covering the K-12 educational system published by De Justo, Digal, and Lagura concluded the opposite of what Dr. Calderon has stated. In their paper, they stated that the ideal

course of action is to impede the implementation of the policy. They insisted that the Philippine education sector is still unprepared for such change, and the program will yield more negative outcomes than positive ones.

However, Lagura et. al. agreed that our educational system is faulty in a sense that the time spent on academic learning institutes may be insufficient in acquiring all the necessary sets of skills and knowledge to move to a higher level of education. They have also stated that enhancing the education system of the Philippines is very crucial as this would, in the long term, affect the economy of the Philippines as a whole. [6]

As such, it is inconclusive whether or not the K-12 would be ideal right now. There are many aspects that we have not considered, including but not limited to its proper implementation. Of course, this problem has not only arisen in the the Philippines but also in other countries that has implemented or would be implementing the new education system.

A journal published by DiPietro, Ferdig, Black, and Preston, from the United States of America, tackles how K-12 study could be implemented using virtual schools. Though they have stated that more research needs to be conducted for more accurate data, they have stated that the introduction of virtual classrooms to the K-12 education system has provided a “hybrid learning environments where aspects of online learning are integrated into traditional, face-to-face settings.” [7] As they have stated, these results are inconclusive for now, but with further research on virtual classrooms and the K-12 system, more conclusive data could be drawn.

In a study conducted by J. Lynn McBrien and Phyllis Jones, they stated that using systems that promote online interaction in classrooms are inefficient. They said that such systems lack interaction between the student and the teachers, and thus “improved teaching and learning strategies are needed to provide students not only the convenience of distance education but also

the kind of access to real-time interaction with the instructor and class peers that simulates the traditional classroom setting.”

At the end, they concluded that such results are predictable with today’s technology, but can be expected to be improved upon as technology grows. There may come a time when the lack of interaction between the students and the teachers would be so minimal that the online learning experience would not be degraded. [8]

However, two writers, Palloff and Pratt, begged to disagree with McBrien and Jones’ conclusion. In a book they’ve published, they’ve found out that students are much more willing to participate in a discussion outside of a classroom. They stated that this may be due to two primary reasons: (1) student has more time to stop and think of an answer, while an answer is immediately required of a student in a class, and (2) “students can discuss openly and not have to worry about failure as much.” [9] With their results, we can assume that implementing an online virtual classroom may not be as counterproductive as once thought.

In an analysis conducted by Sitzmann, Kraiger, Stewart, and Wisher, the effectiveness of using WBI, or web-based instruction, and CI, or classroom instruction, were compared. In their paper, they talked about how WBI is favored in being used in academic learning institutes. This became the center of their analysis, as they have stated “it is important to understand whether or not this delivery medium is effective, whether it is more effective than other delivery media, and what contextual or methodological factors moderate its effectiveness.” In their analysis, they have concluded that, despite having several results that favor WBI over CI, CI was more effective in teaching declarative knowledge to respondents, yet are (Placeholder1) equally effective when using the same instructional methods to deliver training. However, they have stated that “designing long training courses and providing trainees with control, practice, and

feedback during WBI will maximize learning declarative knowledge from WBI relative to CI". [10] In other words, we can assume that web-based instruction can be improved upon to match the results of classroom instruction, or maybe even surpass the standards.

A study conducted by Chang and Smith "determined whether there was a relationship between students' perceptions of how effective course-related interaction was and their level of course satisfaction". In the fall of 2005, 949 students and 25 teaching assistants participated in their study. They found out that several attributes affect whether or not a classroom environment, whether face-to-face or online, would be successful in reaching out to students and making sure they understand the course. As for online classrooms, making sure that course quality is examined and checked, and assessing learner satisfaction are very crucial components in distance education courses. [11]

Merchant, Goetz, Cifuentes, Keeney-Kennicutt, and Davis took a different approach covering virtual learning. In their study, not only did they tackle online learning, but they also tackled about virtual reality being used in the present education system, especially in K-12 system. In their article, they described how financial feasibility hampered the widespread dissemination of virtual reality in K-12 and higher education systems. Here they explain that the cost of procuring the necessary devices (e.g. head-mounted display gear, body suits, and data gloves) for even the most basic functionalities of virtual reality would cost a lot, not to mention the cost of maintenance these devices would require. However, over time, as the development of said devices has improved their cost decreasing and internet connection improving, the implementation of virtual reality to education systems has begun to grow.

In the end, they have concluded in their study that using VR was quite effective, but the results may be inconclusive as the findings have several limitations. They also emphasized the

fact that many studies had provided incomplete information on the variables for their study. [12] However, in accordance with their own study, they concluded that using virtual reality in education systems like K-12 would be effective if not slightly costly.

Another group of researchers, Ju-Ling Shih, Chien-Wen Chuang, and Gwo-Jen Hwang, took a different approach at looking at online classrooms. In their study, they looked upon whether using mobile devices and wireless communications would help students learn in a social science activity. Their results showed that the student's performances have improved overall, whether it is their cognitive learning achievements, their satisfaction level, or their physical participation and affective involvement in the respective activities, as well as the appraisal they received from their teachers, compared to traditional classroom lectures. [13] This further supports the case that online learning isn't necessarily worse than traditional learning, and, in some cases, it might even surpass traditional learning.

A previous study was conducted by Mark Manalang, a UP Manila Alumni. In his work, he created a web-based examination system and tutorial scheduling system for the UP Manila Learning Resource Center. He called this project TOES (Tutorial and Online Examination System), who cater to the following users: student, tutor, staff, admin, and director. He focused on the problem of the LRC about giving mock examinations, which is done manually. He stated that some written answers are not easily readable and may be corrected improperly. He also addressed the problem of how the files were stored; stating that keeping the data in paper and storing them in folders slows down the process of retrieval of questions. Lastly, he noted the system of how the schedules are posted which, as he stated, "fail to get the attention of students".

In his work, he recommended that it would be more efficient if TOES had its own question editor. He also stated that registration and reactivation of student account is frustrating

as it would involve encoding over thousand student information, and it be wise if the system could access an existing UP Manila database and extract the required records to avoid hassle.

[14]

III. Theoretical Framework

A. Dee Hwa Liong Academy

Dee Hwa Liong Academy will be implementing the new K-12 system next school year. Because of this, several protocols have been changed in the school, and the school itself is in the process of adjusting to the new system. Because of their massive student population size, the management of the school could find it challenging to organize and oversee every student along the way. There is also the issue of keeping student records and the like, which are important not only for the school, but also for the students.

One of the solutions proposed is creating an online learning portal, an online learning community. With the said system, the load of the faculty will be definitely lighter, and since most of the work will be done by computers, errors made by the faculty will be significantly reduced.

B. Information System

An IS (information system) is the collection of technical and human resources that provide the storage, computing, distribution, and communication for the information required by all or some part of an enterprise. [15] It is made up of the database, application programs and manual and machine procedures. It also encompasses the computer systems that do the processing. [16]

C. Database Management System

MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use. MySQL is an essential part of almost every open source PHP application. Good examples for PHP/MySQL-based scripts are phpBB, osCommerce and Joomla.

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. [17]

D. Laravel

Laravel is a web application framework with expressive, elegant syntax. The developers believe development must be an enjoyable, creative experience to be truly fulfilling. Laravel attempts to take the pain out of development by easing common tasks used in the majority of web projects, such as authentication, routing, sessions, and caching.

Laravel aims to make the development process a pleasing one for the developer without sacrificing application functionality. Happy developers make the best code. To this end, the developers have attempted to combine the very best of what we have seen in other web frameworks, including frameworks implemented in other languages, such as Ruby on Rails, ASP.NET MVC, and Sinatra.

Laravel is accessible, yet powerful, providing powerful tools needed for large, robust applications. A superb inversion of control container, expressive migration system, and tightly

integrated unit testing support give you the tools you need to build any application with which you are tasked. [18]

E. Scripting Language

PHP, which stands for “PHP: Hypertext Processor”, is a web-based server-side scripting language that is often used in web development. It can be embedded into HTML. PHP can be used on all major operating systems, including Linux, many UNIX variants (including HP-UX, Solaris and OpenBSD), Microsoft Windows, Mac OS X, RISC OS, and others. The greatest advantage of using PHP is having the freedom of choosing an operating system and a web server. Furthermore, you also have the choice of using procedural programming or object oriented programming (OOP), or a mixture of them both. [19]

JavaScript (often shortened to JS) is a lightweight, interpreted, object-oriented language with first-class functions, most known as the scripting language for Web pages, but used in many non-browser environments as well. It is a prototype-based, multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming styles. Javascript can be classified as client-side scripting, and can respond to mouse events, form inputs, and page navigation when embedded on HTML pages. [20]

IV. Design and Implementation

A. Entity-Relationship Diagram

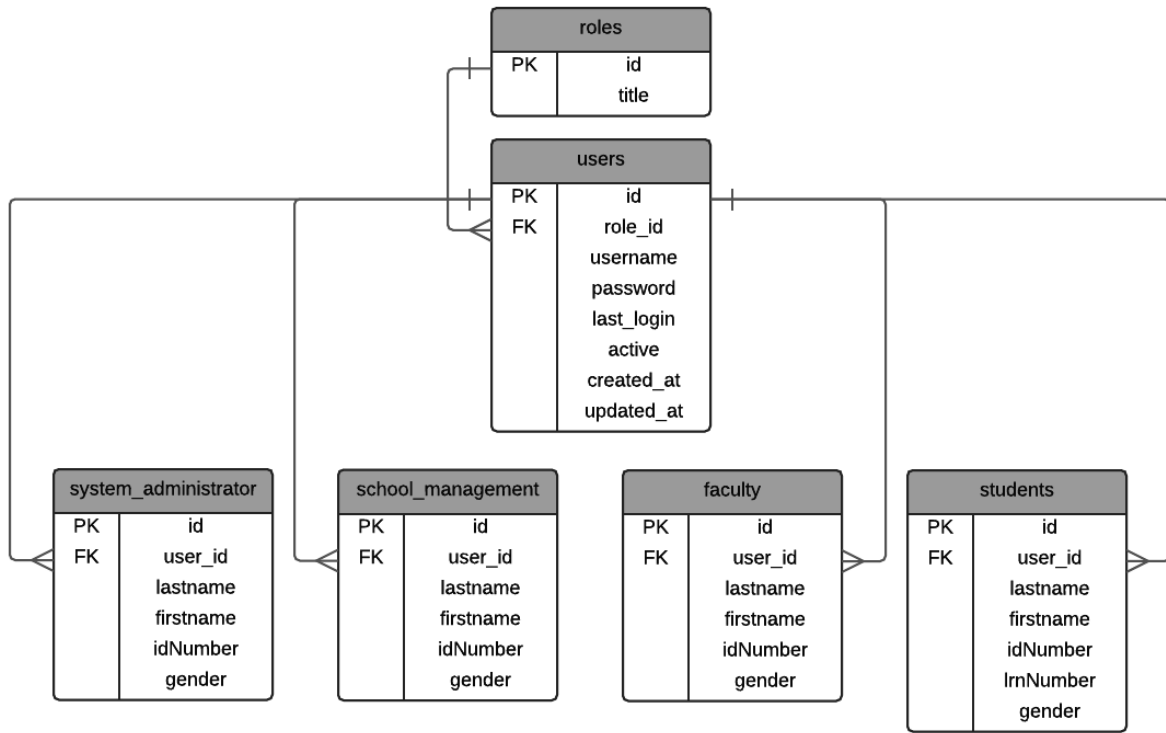


Figure 1: Users and their details

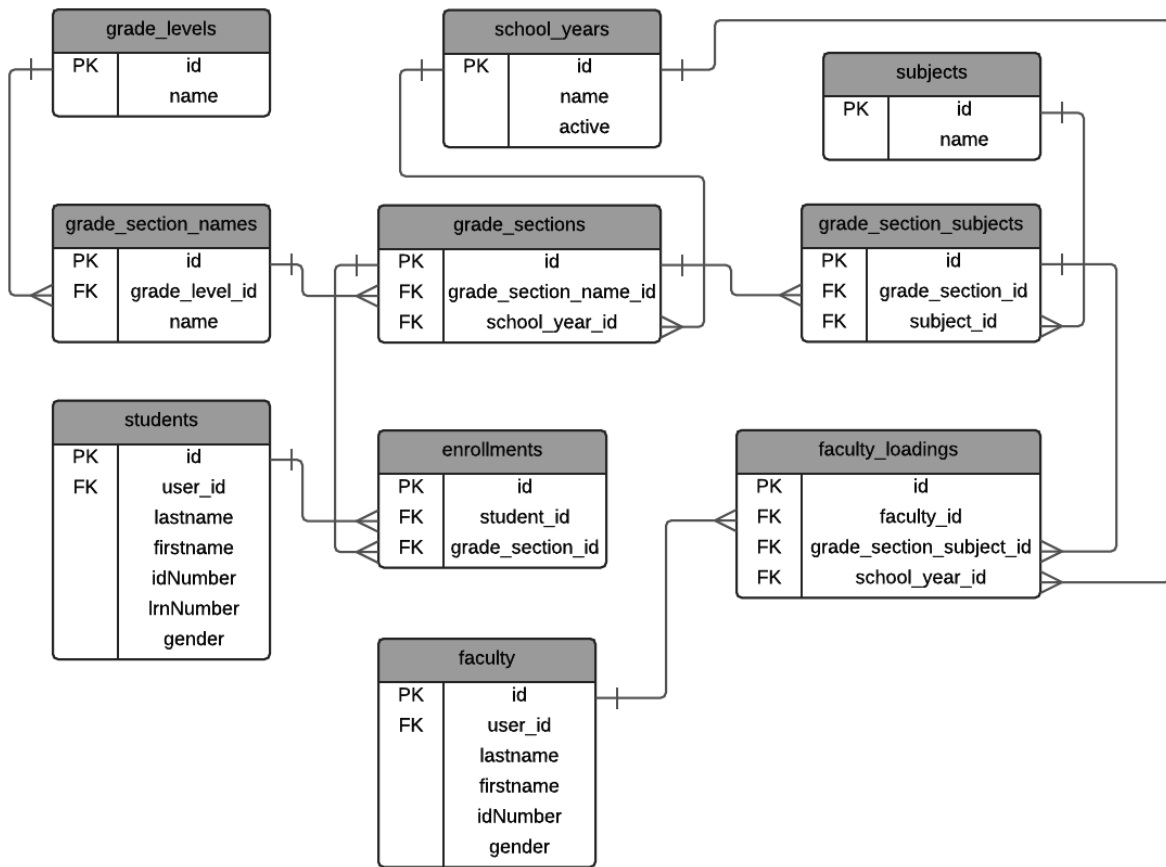


Figure 2: Management of grade subject sections, enrollment of users, assignment of subjects, and assignment of faculty to grade subject section

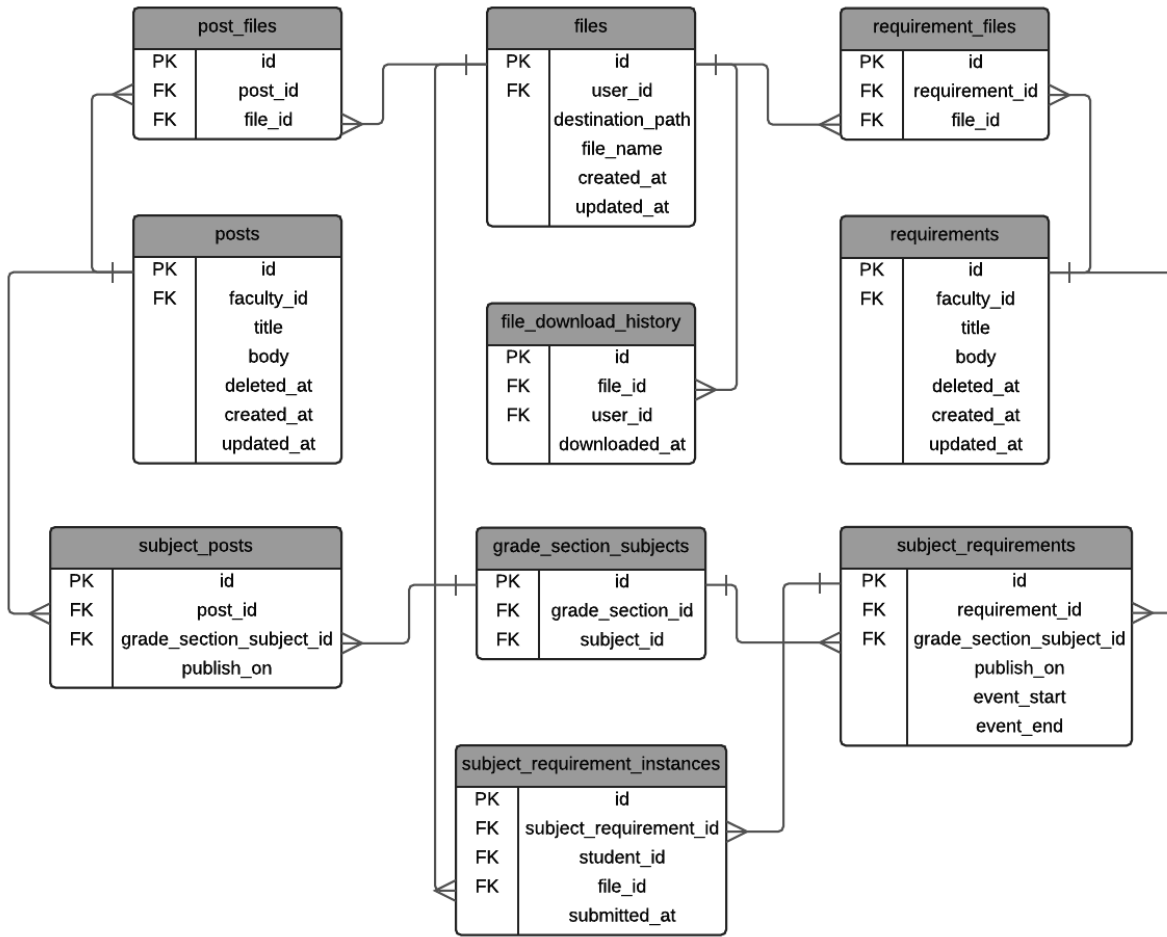


Figure 3: Creation of subject posts and requirements, handling user submissions for a particular requirement, and management of files

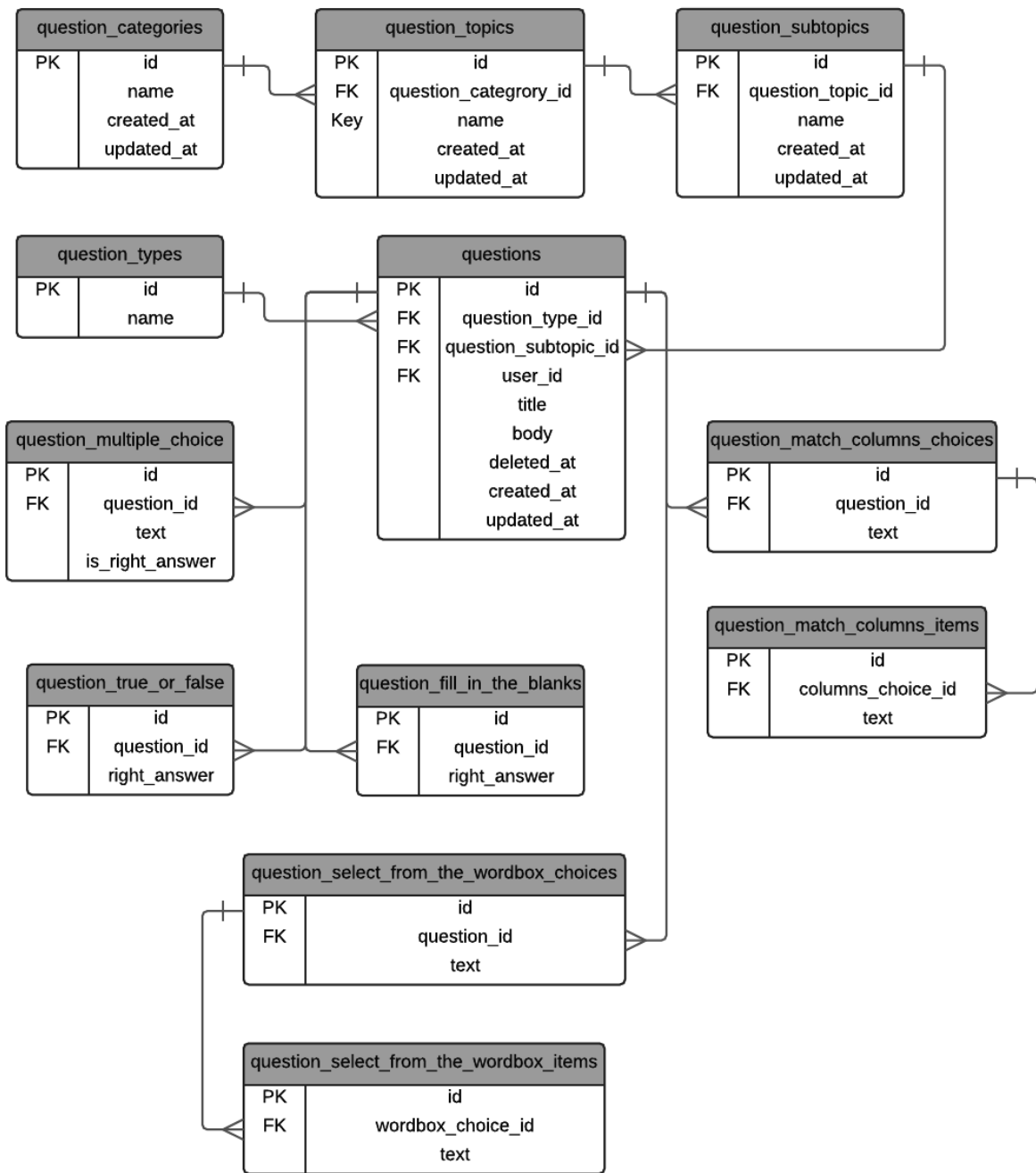


Figure 4: Questions and its different types

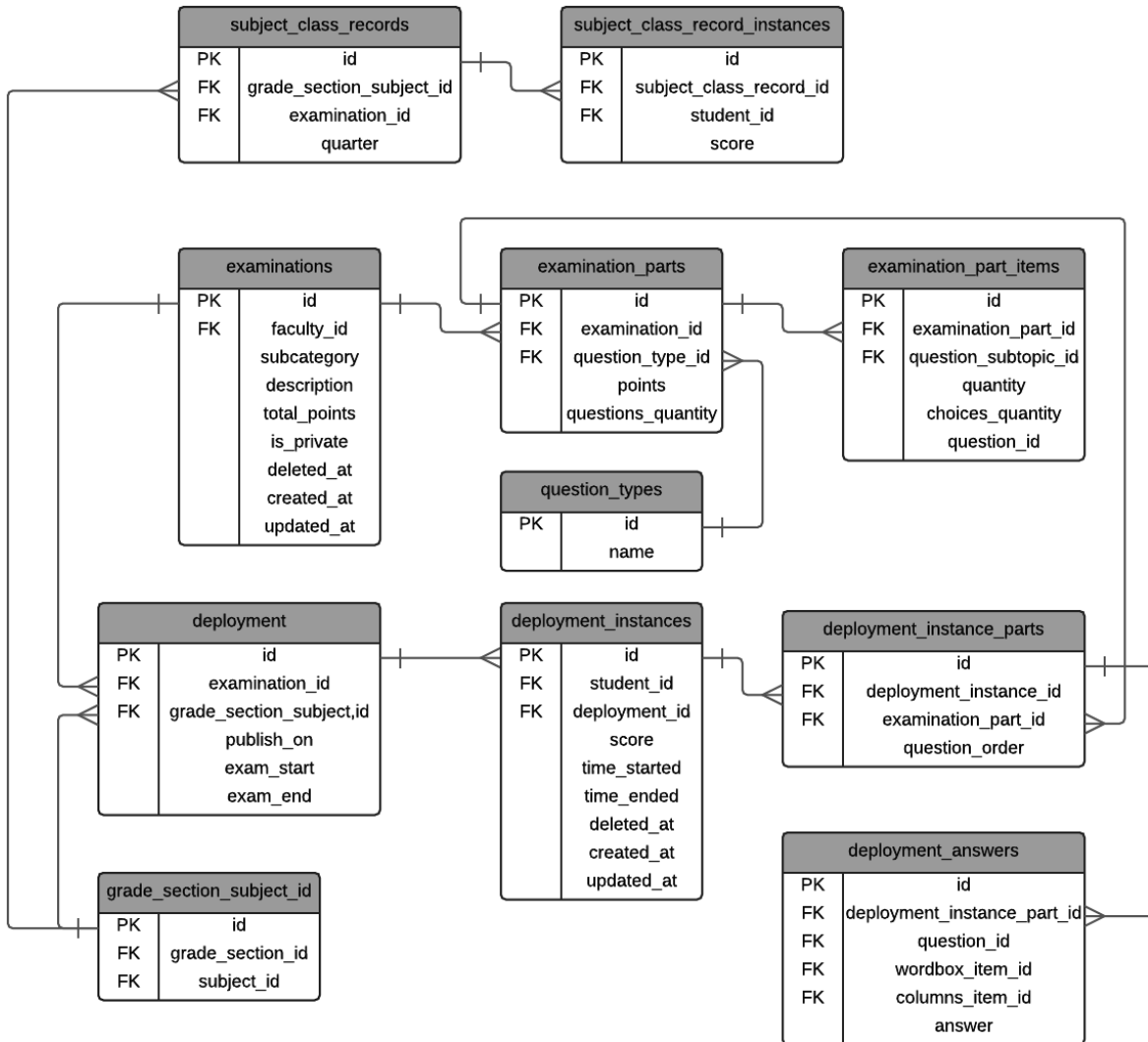


Figure 5: Creating examinations and recording it to the subject class record

B. Data Flow

The system can be classified into two parts. One module is about the online examination, and the other module is about the online portal.

For the online examination, here is how the process goes:

1. Uploading of questions in the question bank

- a. The faculty navigates through the categories, topics, and subtopics. The faculty will have to select from what subtopic the question will be added.
- b. In case the faculty wishes to create new categories, topics, or subtopics, he/she may do so. The faculty would just need to provide a name for the category. The faculty would provide a name for the topic, as well as which category it falls under. The faculty would provide a name for the subtopic, as well as which topic it falls under.
- c. The faculty decides on what type of question to be created. They can select from five different types of exam: multiple choice, match column a with column b, true or false, fill in the blanks, and select from the wordbox.
- d. Depending on the type of question, the faculty will provide the necessary details of the question.
- e. This process can be done repeatedly until the desired number of questions has been created.

2. Creating an examination

- a. The faculty logs in into his account and sees the examination tab on his home page.
- b. The faculty then proceeds to fill out the header, or the details to be used in a class record.
- c. The faculty then proceeds to create parts for the exam. These part require a given question type parameter as well as the maximum quantity of questions that could be added in a part.

- d. Once a part has been created, the faculty then will create items. The way items are created is by randomly selecting questions from a given subtopic a faculty has chosen.
- e. The faculty also specifies the quantity of questions that will be taken from a given subtopic.
- f. The faculty repeats the process of creating parts and items until he/she is satisfied.
- g. The exams are then given out to the selected grade sections. The faculty will also specify other parameters upon assigning an examination, which includes publish date and time.

3. Taking the exam

- a. A student, given that he/she has permission to answer the exam (which means that they are enrolled in the section that is taking the examination) can start answering it as soon as the examination is published, and the time allotted for the exam is still sufficient.
- b. The exam questions are shown one part at a time. The questions are grouped together by what their question types are. The user can answer the current question, skip the question, or go back to the previous questions.
- c. Upon answering a page, the exams are automatically recorded. The student may then go back to his/her previous answers to review and/or change them.
- d. If the student is satisfied with his/her answers, he/she may finish the exam. The results of the exam will then be shown to the user.

4. Checking the exam

- a. The system automatically checks the exam, records the score, stores it in the records database, then prevents the student from taking the exam again.
- b. A copy of the exam questions, the answers chosen by the student, the correct answers for the question, and the total points the student got will be recorded. This data can be viewed later on by the faculty only.

For the online portal, here is how the process goes:

1. Creation of tables

- a. The system administrator can create new users who can use the system. The system administrator would also have to specify certain details about that user, including their last name, first name, etc.
- b. The system administrator would also be able to enroll a given set of students to a given grade section.
- c. Subject would also be assigned to a grade section.
- d. The system administrator then proceeds to assign faculty to each respective grade section subject.
- e. This process is repeated until all students have been enrolled, all sections have been assigned all their respective subjects, and all subjects studied by each section have an assigned faculty to it.

2. User login

- If a student has logged in, he/she will immediately see the list of subjects that his/her section is currently taking. Upon clicking on one of his/her subjects, he

would see articles the faculty for that subject has posted. These articles include posts, requirements, and examinations.

- If a faculty has logged in, he/she would see the links in creating and managing posts, requirements, and examinations. The list of grade section subjects he/she is handling will also be listed as well.
- If a school management has logged in, there would be links to different parts of the system that he/she may view. This includes the events in any given grade section subject.
- If a system administrator has logged in, he/she will see the main menu of creating user accounts, subjects, grade sections, as well as changing the school year.

3. Creating subject requirements

- a. Upon login, a faculty will see the requirements tab, which will enable him/her to see all requirements he/she has created, as well as create new ones.
- b. The faculty provides the necessary information for the requirement.
- c. Before the faculty could submit the form, he/she will have to specify which grade sections would be able to receive the requirement (can be edited later).
- d. The faculty sets a time and date when the event would start and end, or in other words when the students will be able to see the requirement and when the deadline will be. This is done for each grade section that the faculty has selected.

4. Submitting a subject requirement

- a. The student opens up the class he has a requirement on
- b. The student selects the requirement post
- c. The student uploads the requirement needed

- d. The date and time when the upload was done will be recorded
- e. As long as the requirement period is not over yet, the student may upload a new file to submit. Take note that the previous one will be deleted.

5. Creating subject posts

- a. Upon login, a faculty will see the posts tab, which will enable him/her to see all posts he/she has created, as well as create new ones.
- b. The faculty provides the necessary information for the post.
- c. Before the faculty could submit the form, he/she will have to specify which grade sections would be able to receive the post (can be edited later).
- d. The faculty sets a time and date when the post will be publish on each student’s wall. This is done for each grade section that the faculty has selected.

6. Downloading of subject files

- a. The students view any requirement/post that has files attached to it.
- b. The user click on the file name, then the file will be downloaded.
- c. The date and time when the download was made will be recorded.

C. Data Dictionary

deployments

Field	Type	Description
id	bigInteger	Unique identifier of the table
examination_id	bigInteger	Foreign key to “examinations” table
grade_section_subject_id	bigInteger	Foreign key to “grade_section_subjects” table
publish_on	dateTime	Marks when the examination will be published

exam_start	dateTime	Marks when the examination would be answerable by the students
exam_end	dateTime	Marks the end of the examination

deployment_answers

Field	Type	Description
id	bigInteger	Unique identifier of the table
deployment_instance_part_id	bigInteger	Foreign key to “deployment_instance_parts” table
question_id	bigInteger	Foreign key to “questions” table
wordbox_item_id	bigInteger	Nullable except when the question type is “Select from the Wordbox”
columns_item_id	bigInteger	Nullable except when the question type is “Match Column A with Column B”
answer	string	The answer of the student

deployment_instances

Field	Type	Description
id	bigInteger	Unique identifier of the table
deployment_id	bigInteger	Foreign key to “deployments” table
user_id	bigInteger	Foreign key to “users” table
score	integer	The final score of the student
time_started	timestamp	Marks when the student has begun answering the examination
time_ended	timestamp	Marks when the student has finished answering the examination
is_finished	boolean	Determines if the student is finished answering the examination

deleted_at	timestamp	Marks when the instance has been deleted (usually for faculty)
created_at	timestamp	Marks the time the instance is created
updated_at	timestamp	Marks the time the instance was last updated

deployment_instance_parts

Field	Type	Description
id	bigInteger	Unique identifier of the table
deployment_instance_id	bigInteger	Foreign key to “deployment_instances” table
examination_part_id	bigInteger	Foreign key to “examination_parts” table
question_order	string	A sequence of question id separated by a “ ” character. These questions are the questions selected for the given deployment part of the instance.

enrollments

Field	Type	Description
id	bigInteger	Unique identifier of the table
user_id	bigInteger	Foreign key to “users” table
grade_section_id	bigInteger	Foreign key to “grade_sections” table

examinations

Field	Type	Description
id	bigInteger	Unique identifier of the table
user_id	bigInteger	Foreign key to “users” table
subcategory	string	Determines in what subcategory of the class record the examination will be recorded to

description	string	A short description about the examination
total_points	integer	Total points of the examination
deleted_at	timestamp	Marks when the examination has been deleted
created_at	timestamp	Marks when the examination has been created
updated_at	timestamp	Marks when the examination was last updated

examination_parts

Field	Type	Description
id	bigInteger	Unique identifier of the table
examination_id	bigInteger	Foreign key to “examinations” table
question_type_id	bigInteger	Foreign key to “question_types” table
points	integer	Show how much each correct answer is worth in score
questions_quantity	integer	The exact number of questions this part can have

examination_part_items

Field	Type	Description
id	bigInteger	Unique identifier of the table
examination_part_id	bigInteger	Foreign key to “examination_parts” table
question_subtopic_id	bigInteger	Foreign key to “question_subtopics” table
quantity	integer	The number of questions the system will get from the chosen subtopic
choices_quantity	integer	Null for all other question types except for “Match Column A with Column B”. Determines the number of wrong choices to be thrown in the mix
question_id	bigInteger	Nullable except for “Match Column A with Column B” and “Select from the Wordbox”. Determines the question chosen for the given

		examination part
--	--	------------------

faculty

Field	Type	Description
id	bigInteger	Unique identifier of the table
user_id	bigInteger	Foreign key to “users” table
lastname	string	Lastname of user
firstname	string	Firstname of user
idNumber	string	A unique identifier used in DHLA
gender	boolean	Determines the gender of the user

faculty-loadings

Field	Type	Description
id	bigInteger	Unique identifier of the table
grade_section_subject_id	bigInteger	Foreign key to “grade_section_subject_id” table
user_id	bigInteger	Foreign key to “users” table
school_year_id	bigInteger	Foreign key to “school_years” table

files

Field	Type	Description
id	bigInteger	Unique identifier of the table
user_id	bigInteger	Foreign key to “users” table
destination_path	string	Determines the destination path where the file is located
file_name	string	Name of the file

file_download_history

Field	Type	Description
id	bigInteger	Unique identifier of the table
file_id	bigInteger	Foreign key to “files” table
user_id	bigInteger	Foreign key to “users” table
downloaded_at	timestamp	Marks when the file has been downloaded by the given user

grade_levels

Field	Type	Description
id	bigInteger	Unique identifier of the table
name	string	The name of the grade level

grade_sections

Field	Type	Description
id	bigInteger	Unique identifier of the table
grade_section_name_id	bigInteger	Foreign key to “grade_section_names” table
school_year_id	bigInteger	Foreign key to “school_years” table

grade_section_names

Field	Type	Description
id	bigInteger	Unique identifier of the table
grade_level_id	bigInteger	Foreign key to “grade_levels” table
name	string	Name of grade section

grade_section_subjects

Field	Type	Description
id	bigInteger	Unique identifier of the table
grade_section_id	bigInteger	Foreign key to “grade_sections” table
subject_id	bigInteger	Foreign key to “subjects” table

posts

Field	Type	Description
id	bigInteger	Unique identifier of the table
user_id	bigInteger	Foreign key to “users” table
title	string	Title of the post
body	textarea	Contains the message of the post
deleted_at	timestamp	Marks the time when the post was deleted
created_at	timestamp	Marks the time when the post was created
updated_at	timestamp	Marks the time when the post was last updated

post_files

Field	Type	Description
id	bigInteger	Unique identifier of the table
post_id	bigInteger	Foreign key to “posts” table
file_id	bigInteger	Foreign key to “files” table

questions

Field	Type	Description
id	bigInteger	Unique identifier of the table

question_subtopic_id	bigInteger	Foreign key to “question_subtopics” table
question_type_id	bigInteger	Foreign key to “question_types” table
title	string	Title of the question
body	string	Contains the message of the question
deleted_at	timestamp	Marks the time when the question was deleted
created_at	timestamp	Marks the time when the question was created
updated_at	timestamp	Marks the time when the question was last updated

question_categories

Field	Type	Description
id	bigInteger	Unique identifier of the table
name	string	Name of the category
created_at	timestamp	Marks the time when the category was created
updated_at	timestamp	Marks the time when the category was last updated

question_fill_in_the_blanks

Field	Type	Description
id	bigInteger	Unique identifier of the table
question_id	bigInteger	Foreign key to “questions” table
right_answer	string	The correct answer

question_match_columns_choices

Field	Type	Description
id	bigInteger	Unique identifier of the table

question_id	bigInteger	Foreign key to “questions” table
text	string	The text message of the choice

question_match_columns_items

Field	Type	Description
id	bigInteger	Unique identifier of the table
columns_choice_id	bigInteger	Foreign key to “question_match_columns_choice” table
text	string	The text message of the item

question_multiple_choice

Field	Type	Description
id	bigInteger	Unique identifier of the table
question_id	bigInteger	Foreign key to “questions” table
text	string	The text message of the choice
is_right_answer	boolean	The correct answer

question_select_from_the_wordbox_choices

Field	Type	Description
id	bigInteger	Unique identifier of the table
question_id	bigInteger	Foreign key to “questions” table
text	string	The text message of the choice

question_select_from_the_wordbox_item

Field	Type	Description
-------	------	-------------

id	bigInteger	Unique identifier of the table
wordbox_choice_id	bigInteger	Foreign key to “question_select_from_the_wordbox_choices” table
text	string	The text message of the choice

question_subtopics

Field	Type	Description
id	bigInteger	Unique identifier of the table
question_topic_id	bigInteger	Foreign key to “question_topics” table
name	string	Name of the topic
created_at	timestamp	Marks the time when the subtopic was created
updated_at	timestamp	Marks the time when the subtopic was last updated

question_topics

Field	Type	Description
id	bigInteger	Unique identifier of the table
question_category_id	bigInteger	Foreign key to “question_categories” table
name	string	Name of the topic
created_at	timestamp	Marks the time when the topic was created
updated_at	timestamp	Marks the time when the topic was last updated

question_true_or_false

Field	Type	Description
id	bigInteger	Unique identifier of the table

question_id	bigInteger	Foreign key to “questions” table
right_answer	boolean	The correct answer

question_types

Field	Type	Description
id	bigInteger	Unique identifier of the table
name	string	Name of the question type

requirements

Field	Type	Description
id	bigInteger	Unique identifier of the table
user_id	bigInteger	Foreign key to “users” table
title	string	Title of the requirement
body	textarea	Contains the message of the requirement
deleted_at	timestamp	Marks the time when the requirement was deleted
created_at	timestamp	Marks the time when the requirement was created
updated_at	timestamp	Marks the time when the requirement was last updated

requirement_files

Field	Type	Description
id	bigInteger	Unique identifier of the table
requirement_id	bigInteger	Foreign key to “requirements” table
file_id	bigInteger	Foreign key to “files” table

roles

Field	Type	Description
id	bigInteger	Unique identifier of the table
name	string	The name of the role

school_management

Field	Type	Description
id	bigInteger	Unique identifier of the table
user_id	bigInteger	Foreign key to “users” table
lastname	string	Lastname of user
firstname	string	Firstname of user
idNumber	string	A unique identifier used in DHLA
gender	boolean	Determines the gender of the user

school_years

Field	Type	Description
id	bigInteger	Unique identifier of the table
name	string	The name of the school year
active	boolean	Determines the current school year

students

Field	Type	Description
id	bigInteger	Unique identifier of the table
user_id	bigInteger	Foreign key to “users” table
lastname	string	Lastname of user

firstname	string	Firstname of user
idNumber	string	A unique identifier used in DHLA
lnrNumber	string	An identifier unique for every student in the country
gender	boolean	Determines the gender of the user

subjects

Field	Type	Description
id	bigInteger	Unique identifier of the table
name	string	Name of the subject

subject_class_records

Field	Type	Description
id	bigInteger	Unique identifier of the table
grade_section_subject_id	bigInteger	Foreign key to “grade_section_subjects” table
examination_id	bigInteger	Foreign key to “examinations” table
quarter	integer	Tells in what quarter the examination belongs to

subject_class_record_instances

Field	Type	Description
id	bigInteger	Unique identifier of the table
subject_class_record_id	bigInteger	Foreign key to “subject_class_records” table
user_id	bigInteger	Foreign key to “users” table
score	integer	The score the student got from the examination

subject_post

Field	Type	Description
id	bigInteger	Unique identifier of the table
post_id	bigInteger	Foreign key to “posts” table
grade_section_subjet_id	bigInteger	Foreign key to “grade_section_subjects” table
publish_on	dateTime	Marks the time when the post will be published

subject_requirements

Field	Type	Description
id	bigInteger	Unique identifier of the table
requirement_id	bigInteger	Foreign key to “requirements” table
grade_section_subjet_id	bigInteger	Foreign key to “grade_section_subjects” table
publish_on	dateTime	Marks the time when the requirement will be published
event_start	dateTime	Marks the time when the requirement article will be open for submission
event_end	dateTime	Marks the time when the requirement article will no longer accept submissions

subject_requirement_instances

Field	Type	Description
id	bigInteger	Unique identifier of the table
subject_requirement_id	bigInteger	Foreign key to “subject_requirments”
file_id	bigInteger	Foreign key to “files” table
user_id	bigInteger	Foreign key to “users’ table
downloaded_at	timestamp	Marks when the file has been downloaded by the given user

system_administrator

Field	Type	Description
id	bigInteger	Unique identifier of the table
user_id	bigInteger	Foreign key to “users” table
lastname	string	Lastname of user
firstname	string	Firstname of user
idNumber	string	A unique identifier used in DHLA
gender	boolean	Determines the gender of the user

users

Field	Type	Description
id	bigInteger	Unique identifier of the table
username	string	Account name used for logging in
password	string	String for used for login verification
role_id	bigInteger	Foreign key to “roles” table
last_login	timestamp	Marks the last time the user has logged in
active	boolean	Determines whether the account is active
created_at	timestamp	Marks when the account was created
updated_at	timestamp	Marks when the account was last updated

D. System Architecture

The system uses Laravel as its base framework. Laravel uses an MVC (Model-View-Controller) architectural pattern. As defined by Margaret Rouse, “The MVC pattern has been

heralded by many developers as a useful pattern for the reuse of object code and a pattern that allows them to significantly reduce the time it takes to develop applications with user interfaces.”

[21]

The system also uses the following libraries/software:

1. Composer

The developer also recommends the use of composer. Although technically not required, installing Composer is highly recommended because it makes installing Laravel a lot easier, as well as other PHP libraries or frameworks you may desire to use.

2. Laravel Collective

Laravel Collective is a library recommended by Laravel developers. Laravel Collective contains functions that enable the creation of forms with ease. It is completely optional in creating Laravel systems, but is highly recommended because of its usability.

3. Laracasts Flash

Laracasts Flash provides an easy way to insert flash messaging into your system. Flash messages are a great way to convey messages to users, especially notifications on whether or not an action by the user is successful or has failed.

4. nicEdit WYSIWYG Content Editor

The system requires fully customizable messages to be created, and nicEdit editor provides a great way to implement this. Installing the library can enable selected normal form text areas to be converted to nicEdit editors which has functionalities that allow formatting of texts, including but not limited to paragraph alignment, fonts, bold, italics, underlines, lists, and inserting images and links the in the text.

It should be noted, however, that the developer has customized the nicEditor to allow storage of images locally. The images were originally meant to be uploaded to an external site, but as per client request the location has been changed.

5. Laravel Random

Laravel Random is an external PHP library made for the Laravel framework which, as the name suggests, lets you generate random strings consisting of either alphanumeric characters, special characters, or both. This is useful in generating temporary password values, generating short keys, and the like.

E. Technical Architecture

1. PHP 7

Laravel entirely depends on PHP 7. PHP 7 has new functionality that is vital on making controller, view, middleware, database migrations, and almost any other Laravel objects.

Aside from installing PHP 7, the extensions OpenSSL, PDO, Mbstring, and Tokenizer must also be enabled.

2. Apache 2.4

The system requires installing at least version 2.4 of Apache. The software is free to download on the Apache website.

3. Microsoft Visual C++ 2015

The system also requires the 2015 version of Microsoft Visual C++. The software is free to download on the Microsoft website and is used to install both the Apache and the Laravel Framework itself.

V. Results

The Dee Hwa Liong Academy Virtual Classroom is an online (although also accessible offline) student and faculty learning portal. The aim of the system is to create a bridge between students and faculty whenever direct confrontation is not applicable. The system has many functionalities, as well as web pages, sorted by user.

A. Welcome pages

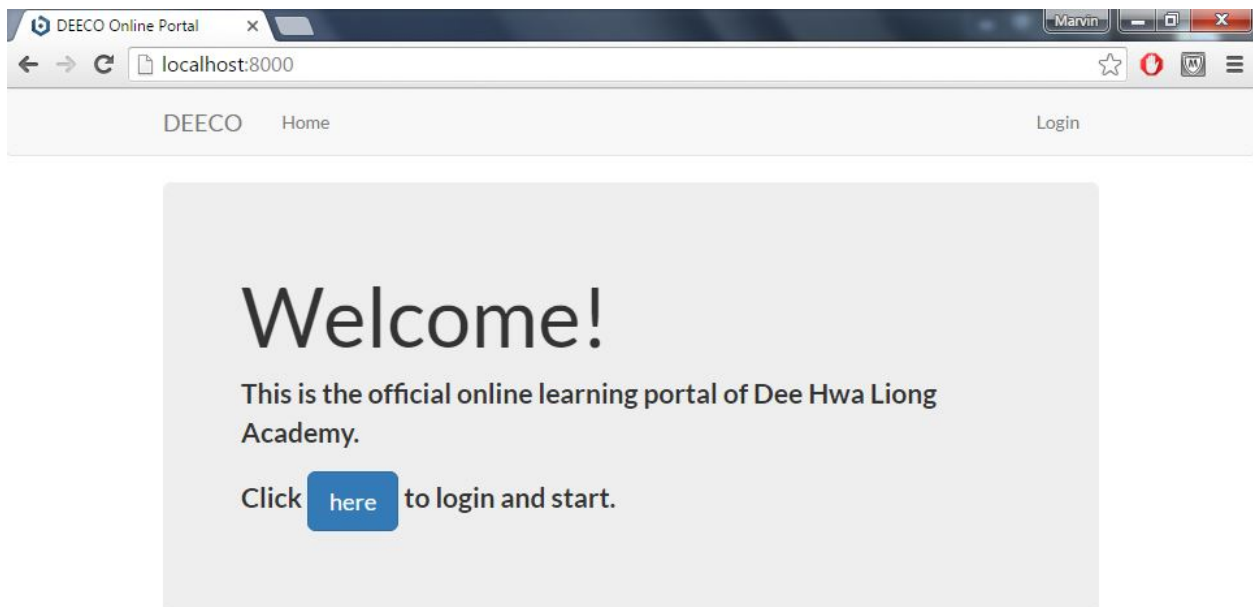


Fig. 6: Showing the welcome page of the site

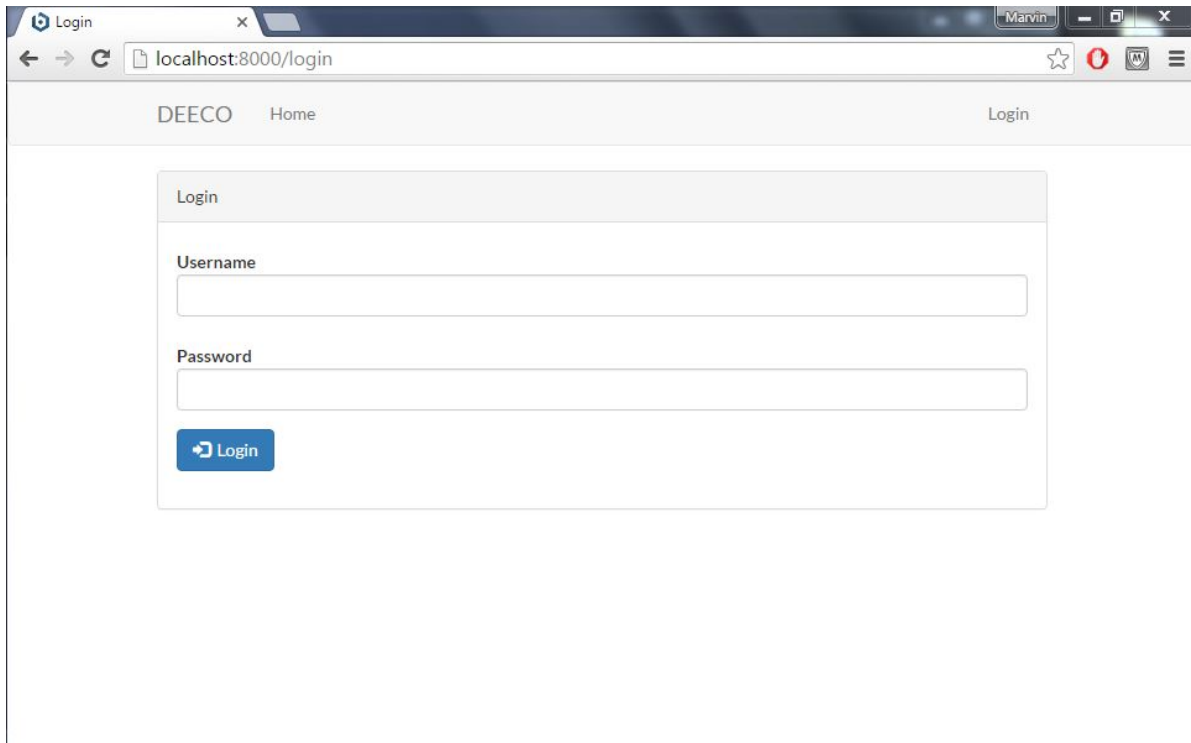


Figure 7: Showing the login page

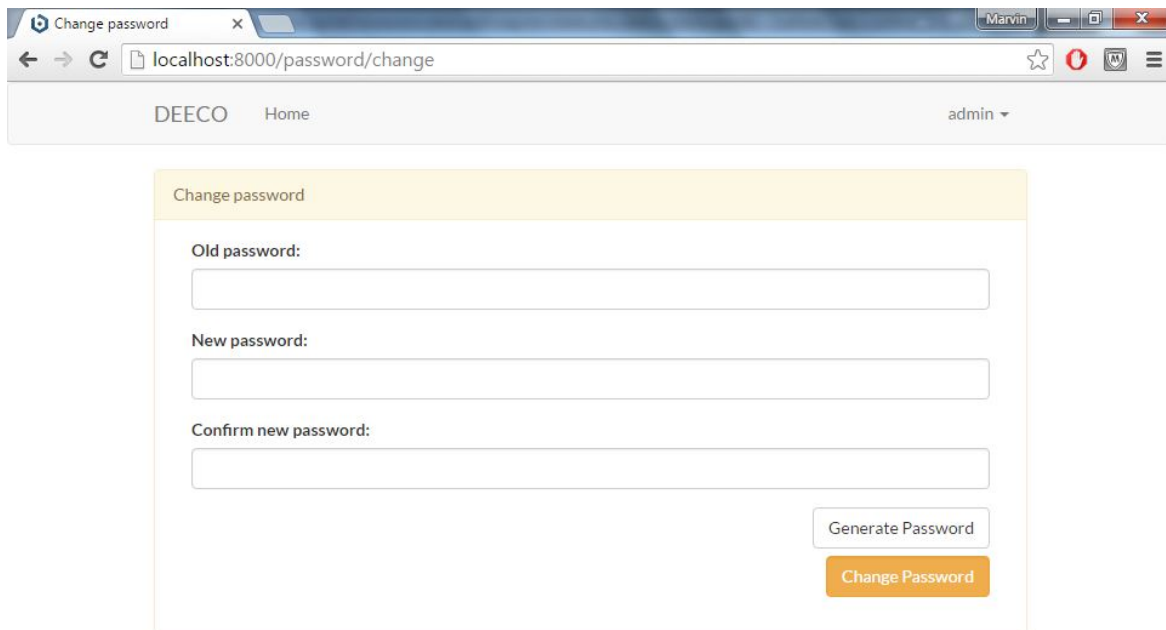


Figure 8: Showing the password change form. This is, of course, only available to registered users.

B. System Administrator pages

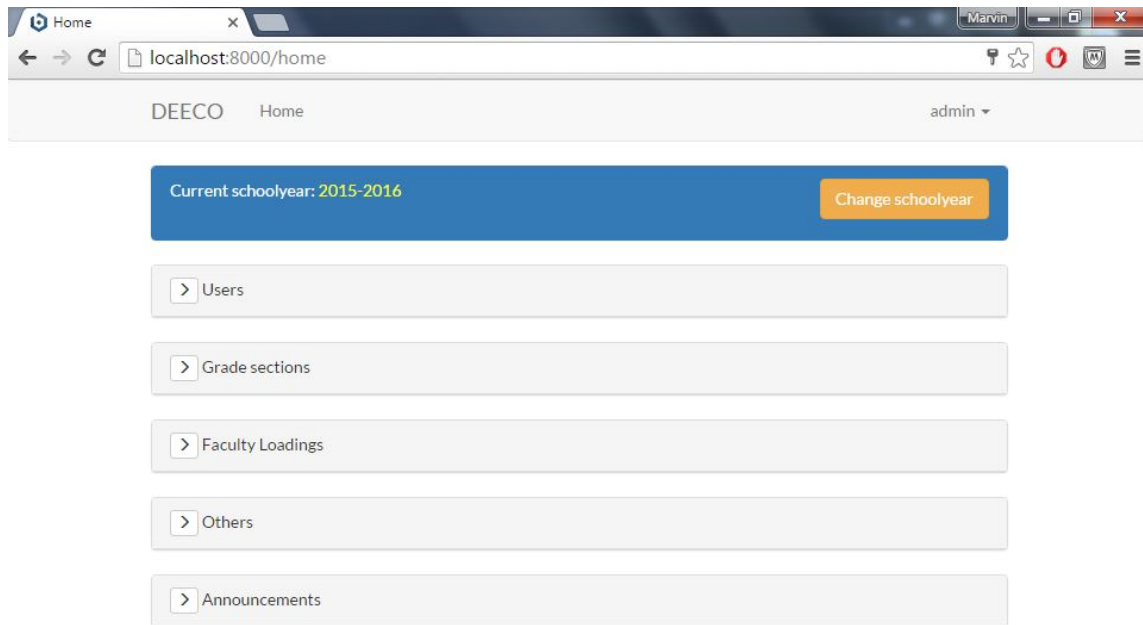


Figure 9: Home page of the system administrator upon login

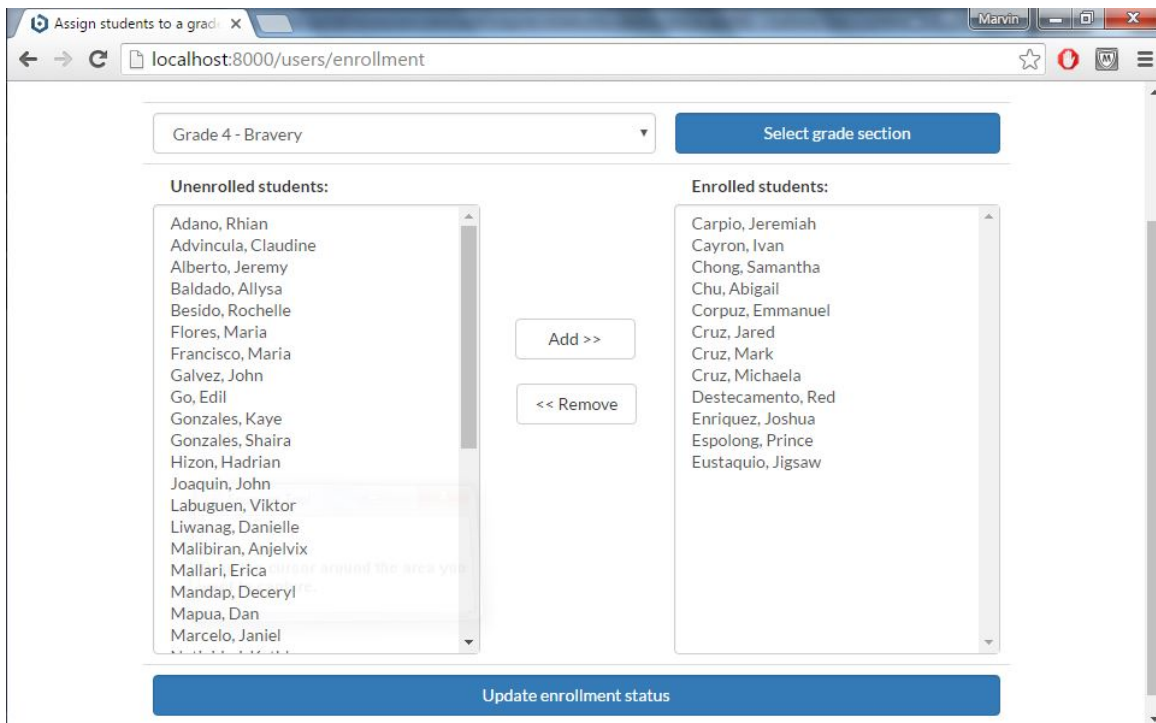


Figure 10: Enrolling students to a selected grade section

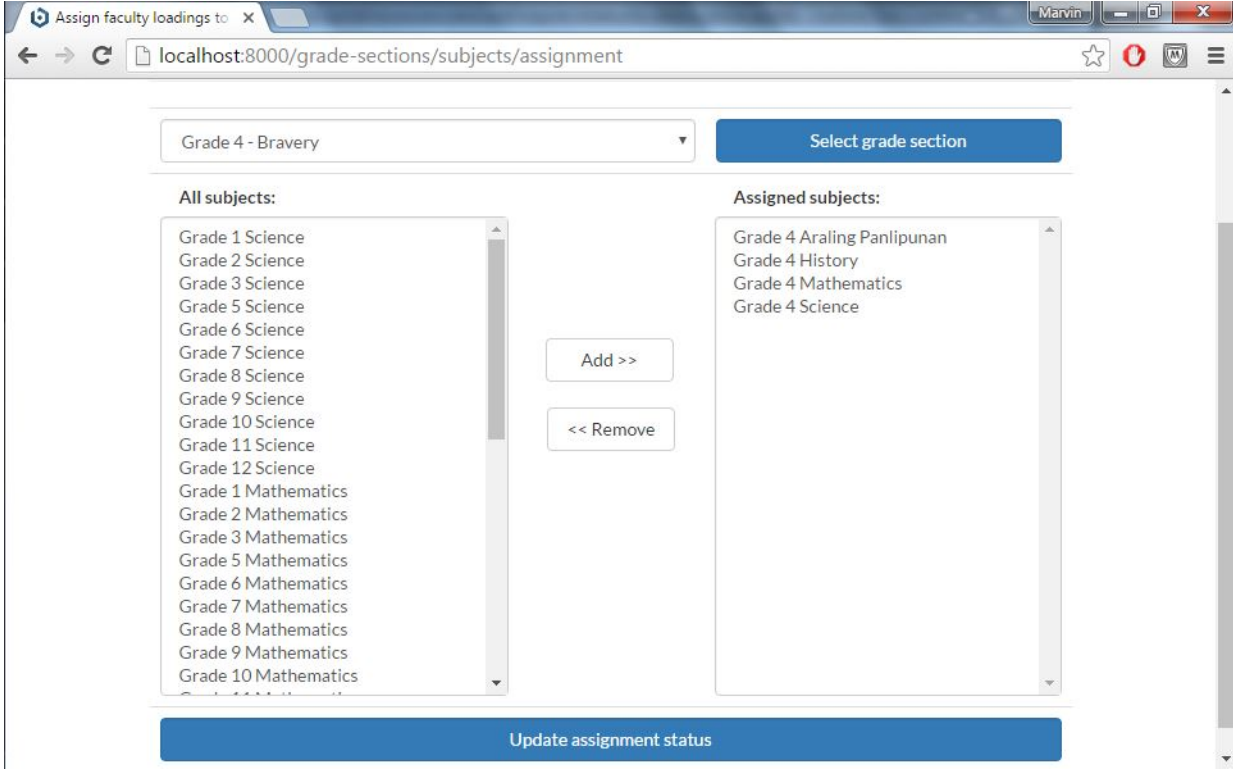


Figure 11: Assigning subjects to a grade section

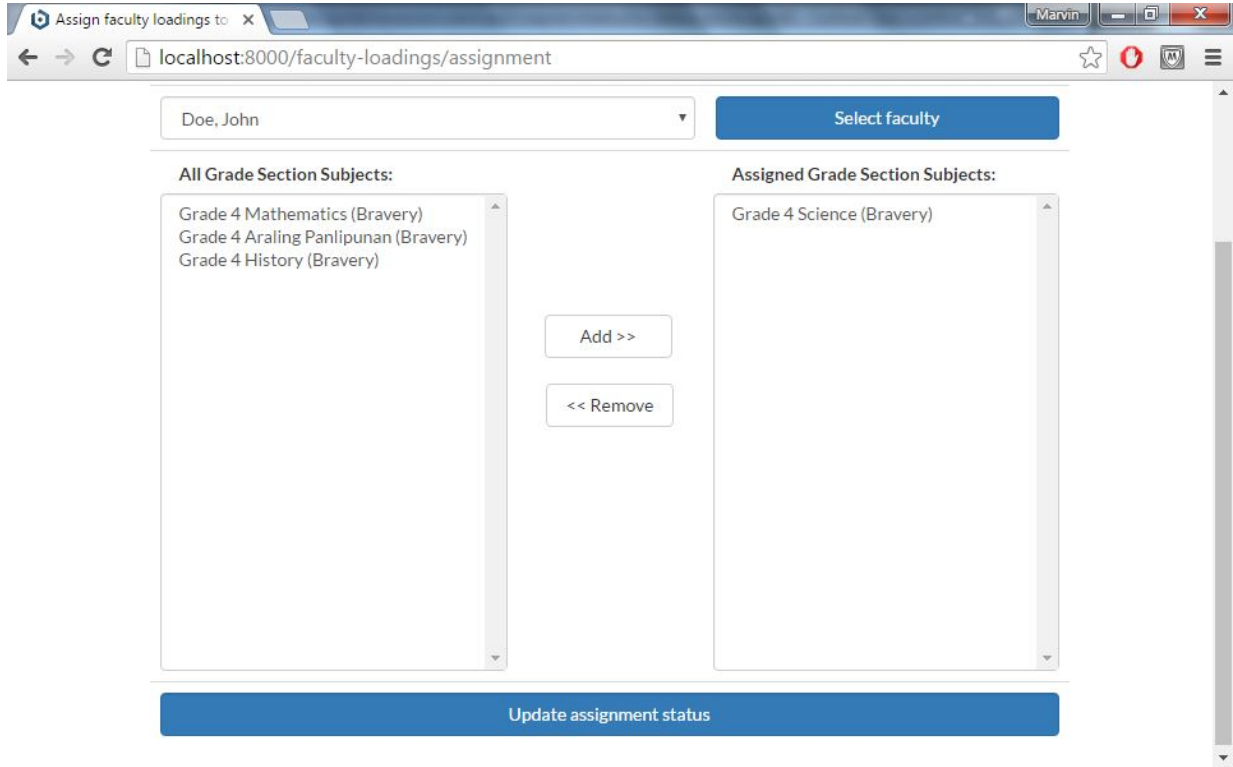


Figure 12: Assigning grade section subjects to a faculty

C. Faculty Pages

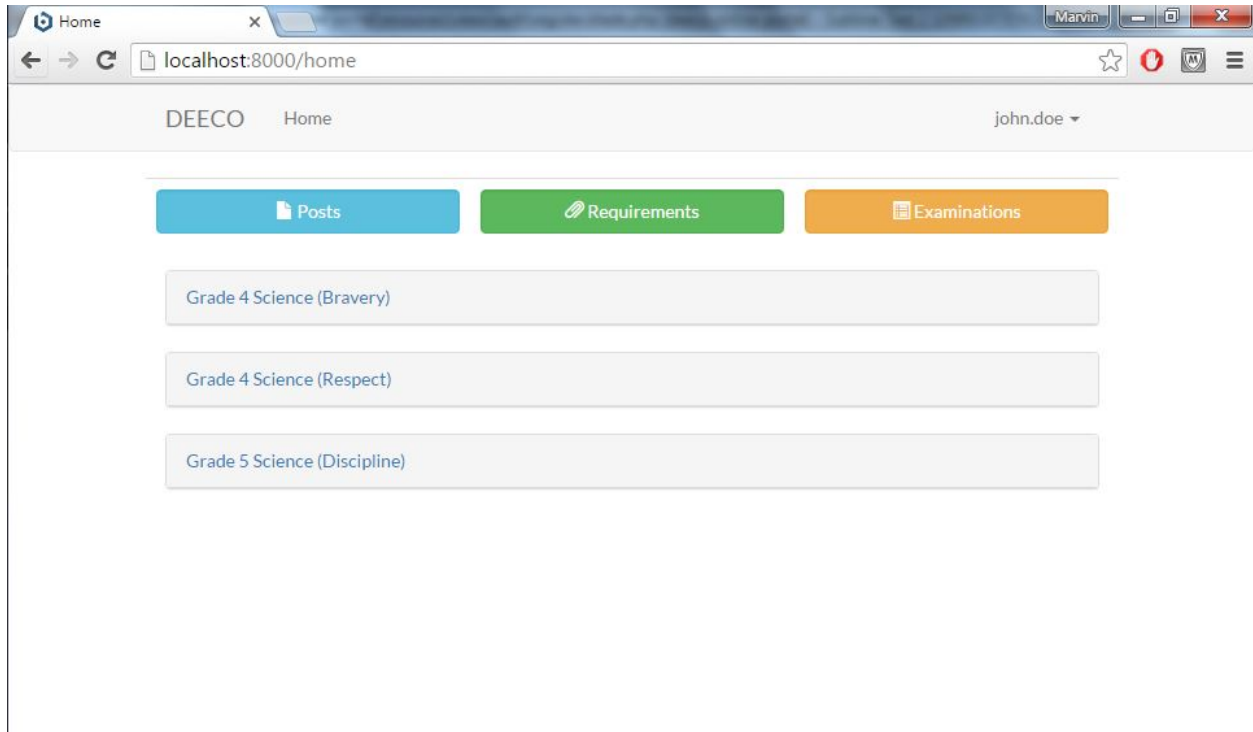


Figure 13: Home page of the faculty upon login

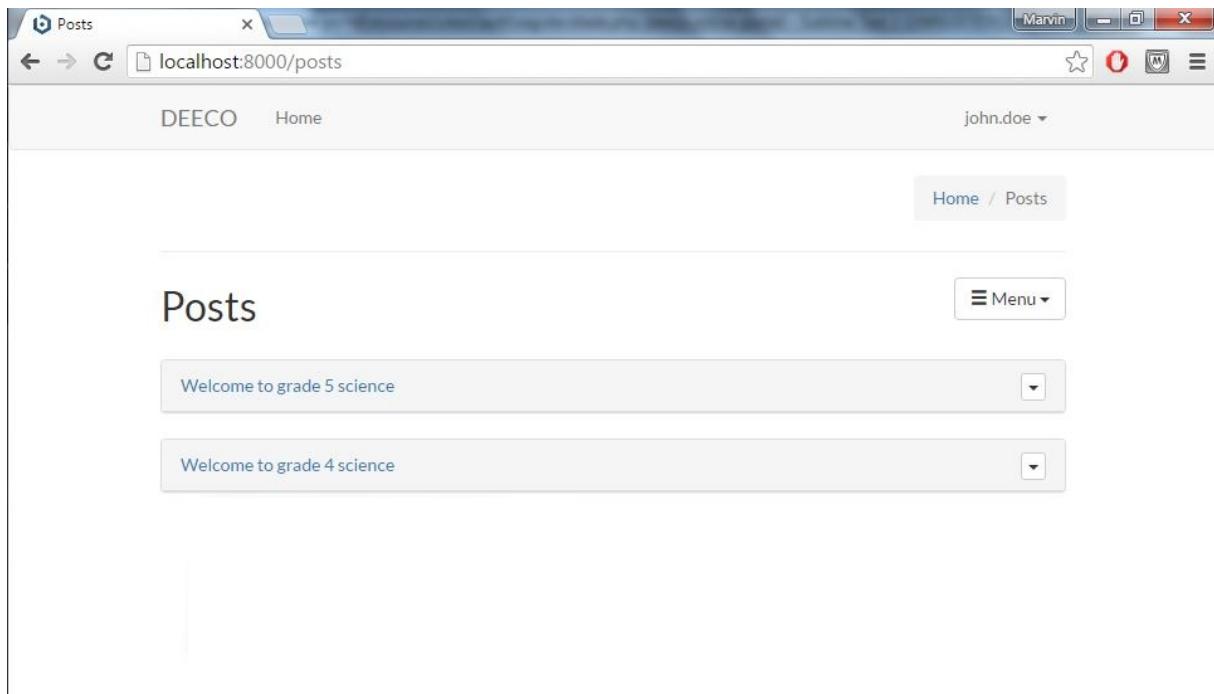


Figure 14: List of posts available for publishing

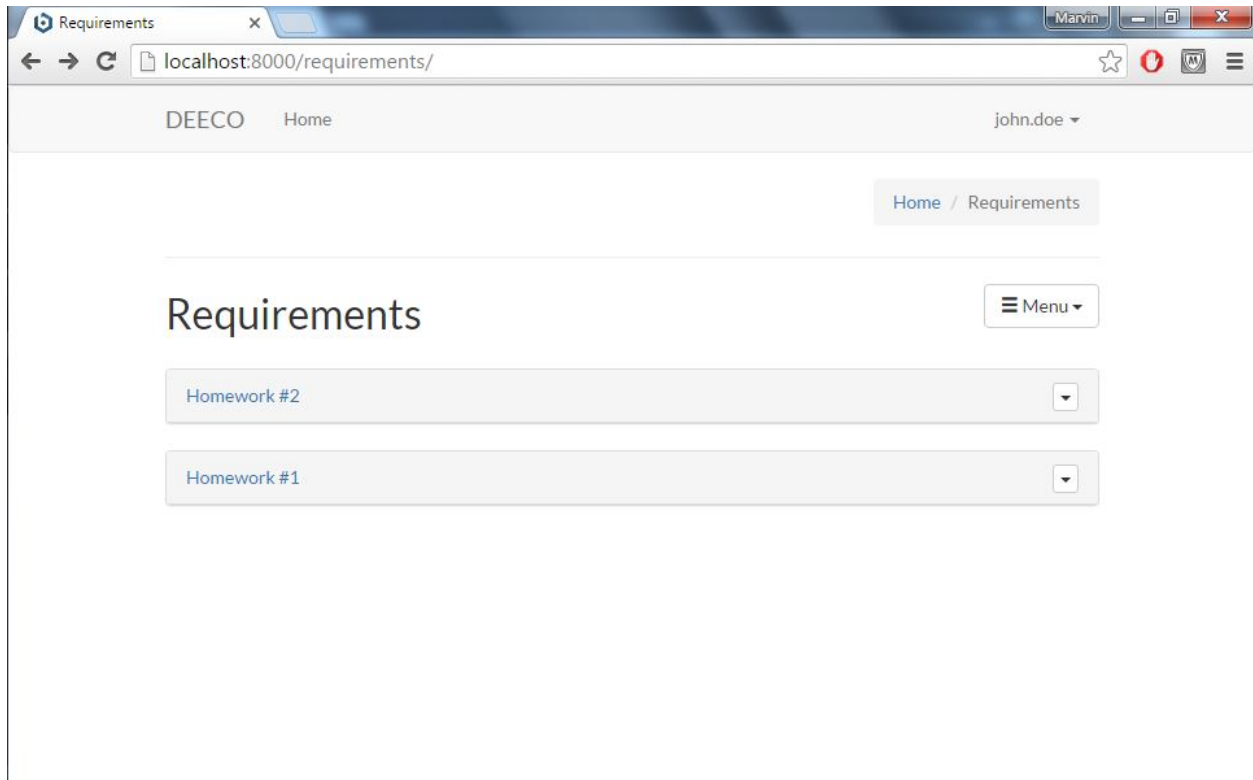


Figure 15: List of requirements available for publishing

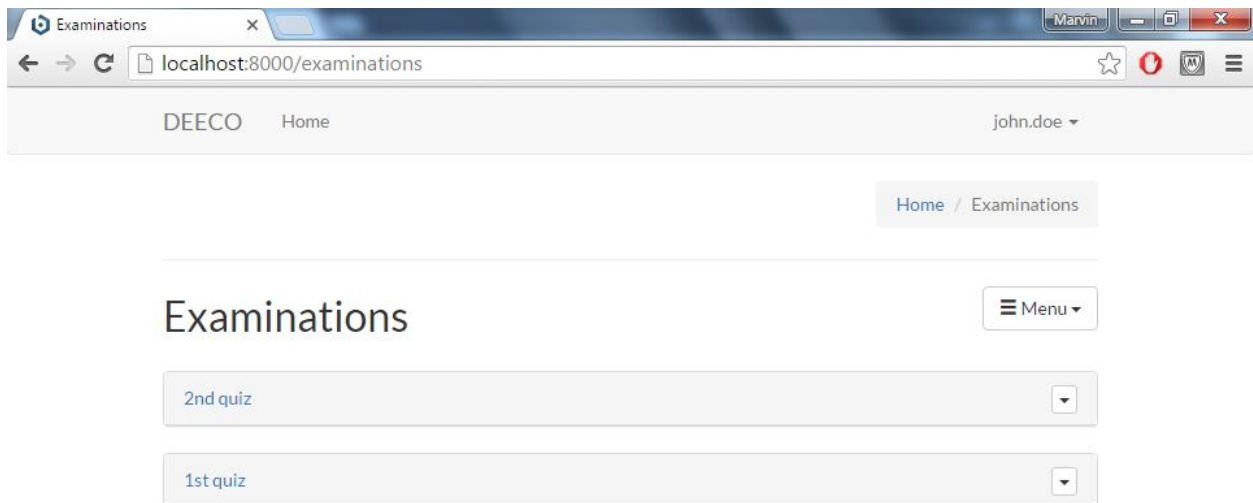


Figure 16: List of examinations available for publishing

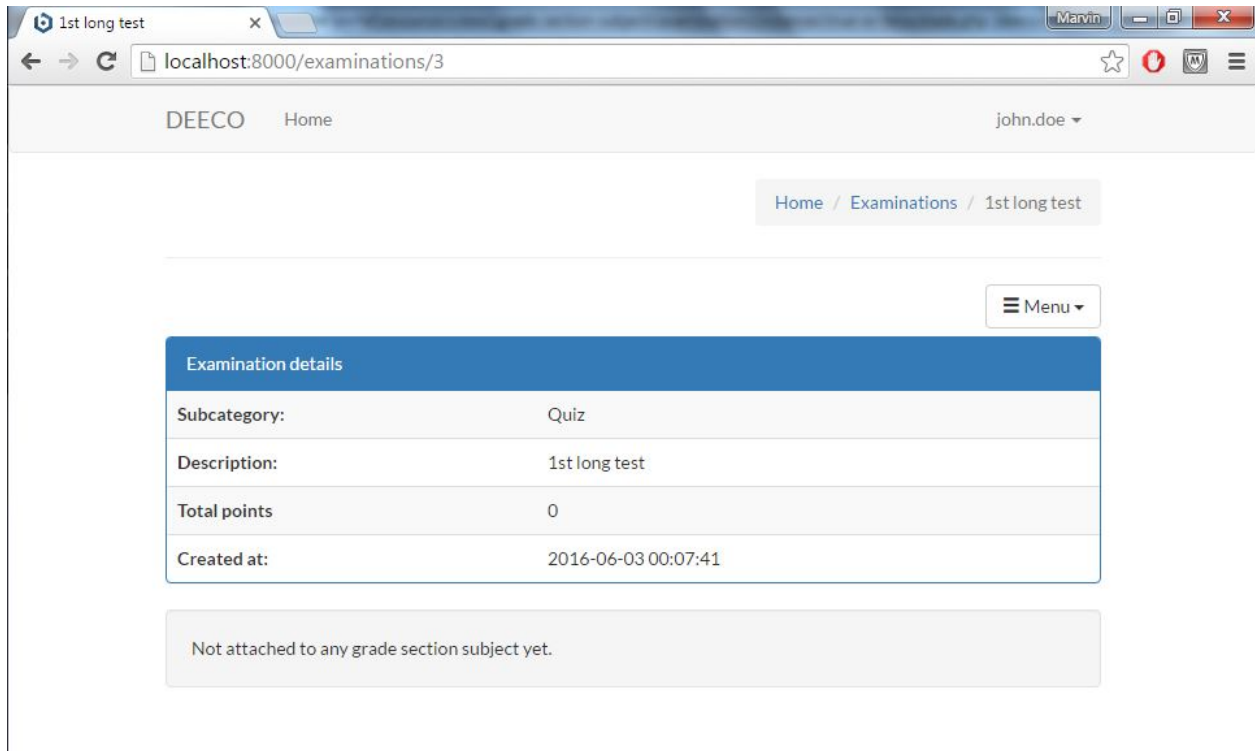


Figure 17: Showing the details of an examination

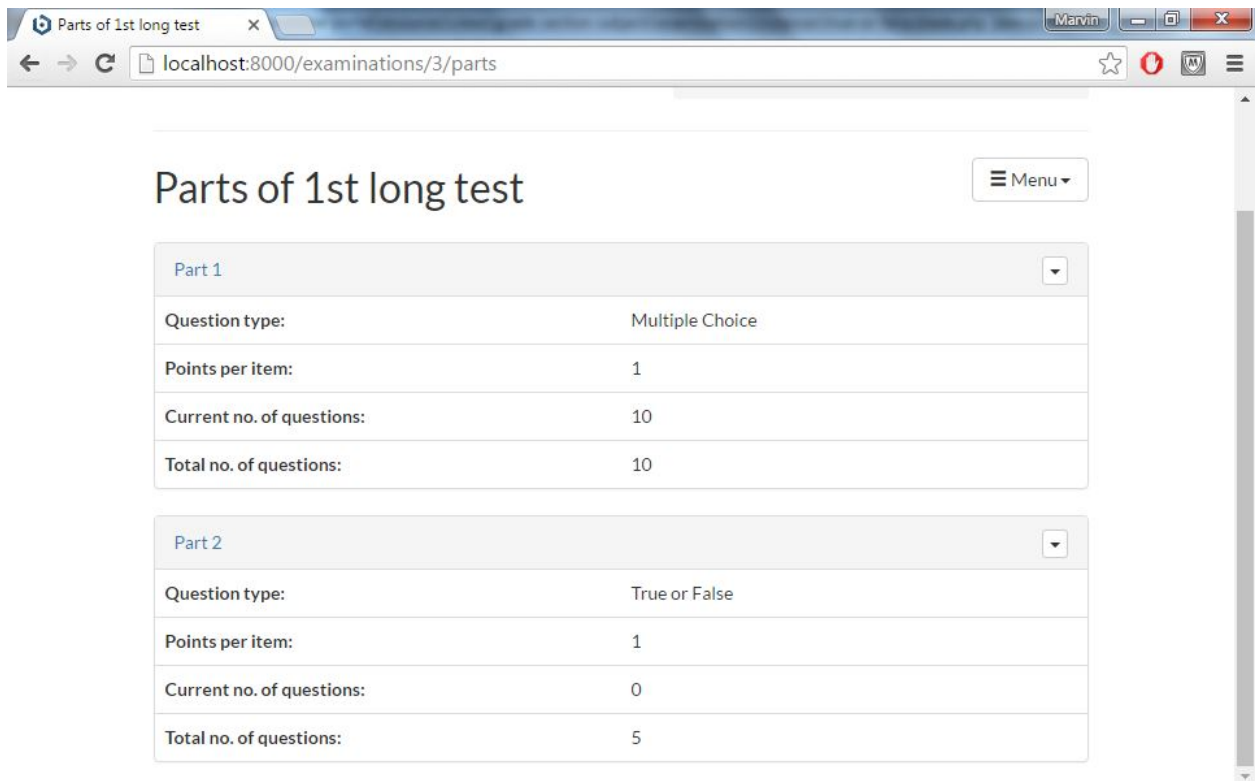


Figure 18: Showing parts of an examination

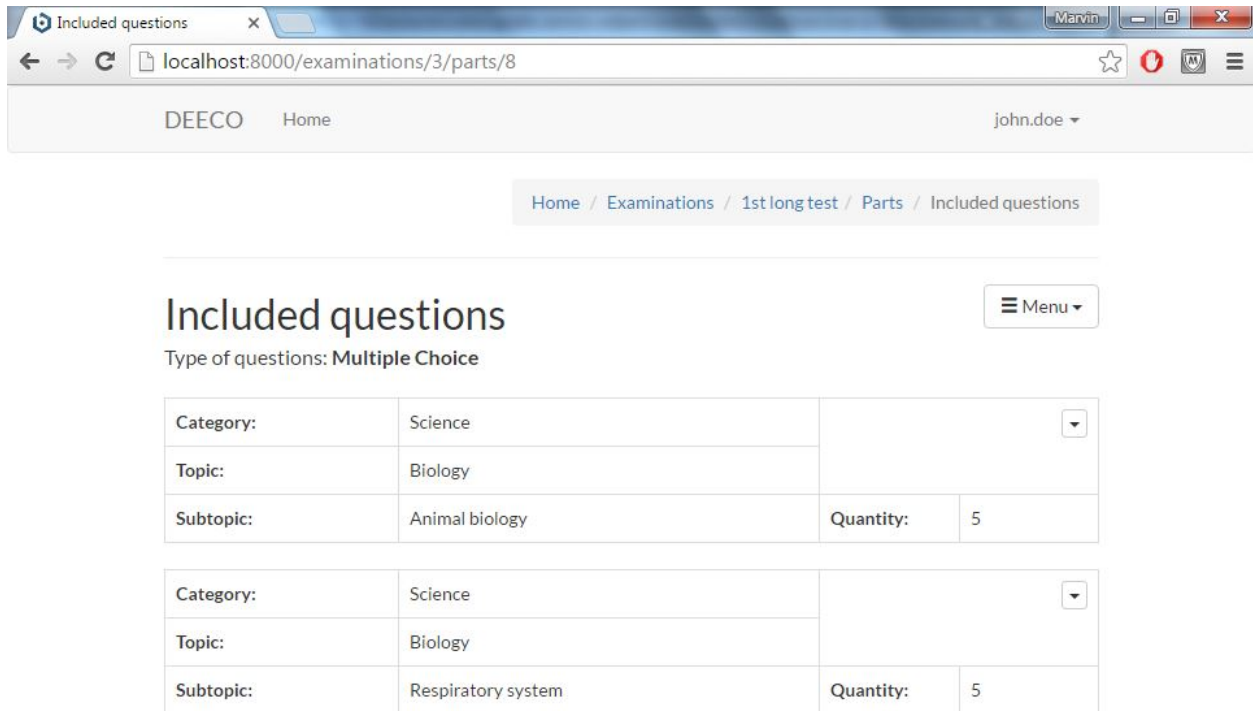


Figure 19: Showing the items included in the examination part

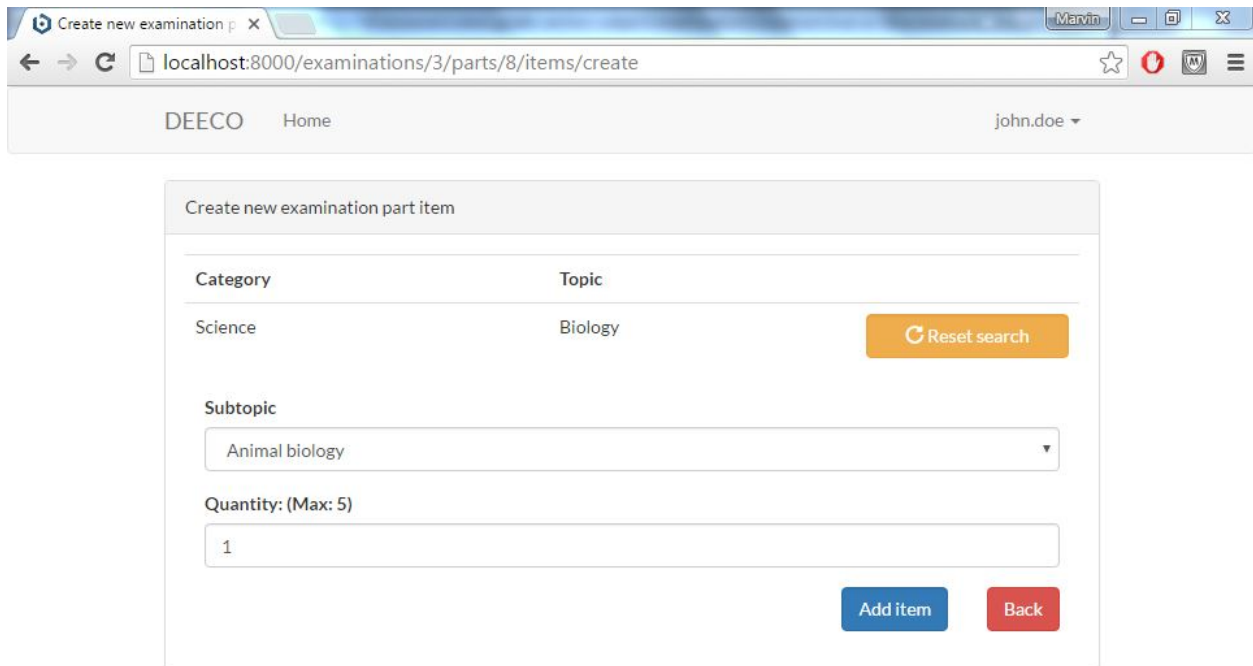


Figure 20: Adding an item to an examination part

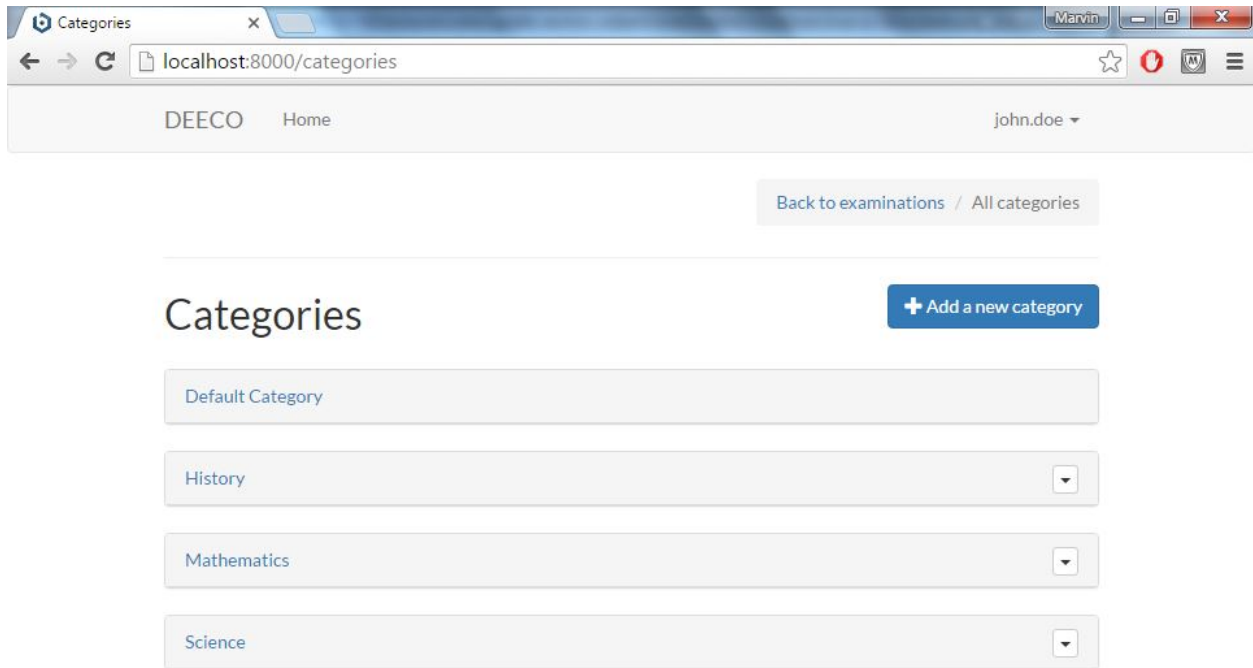


Figure 21: List of question categories

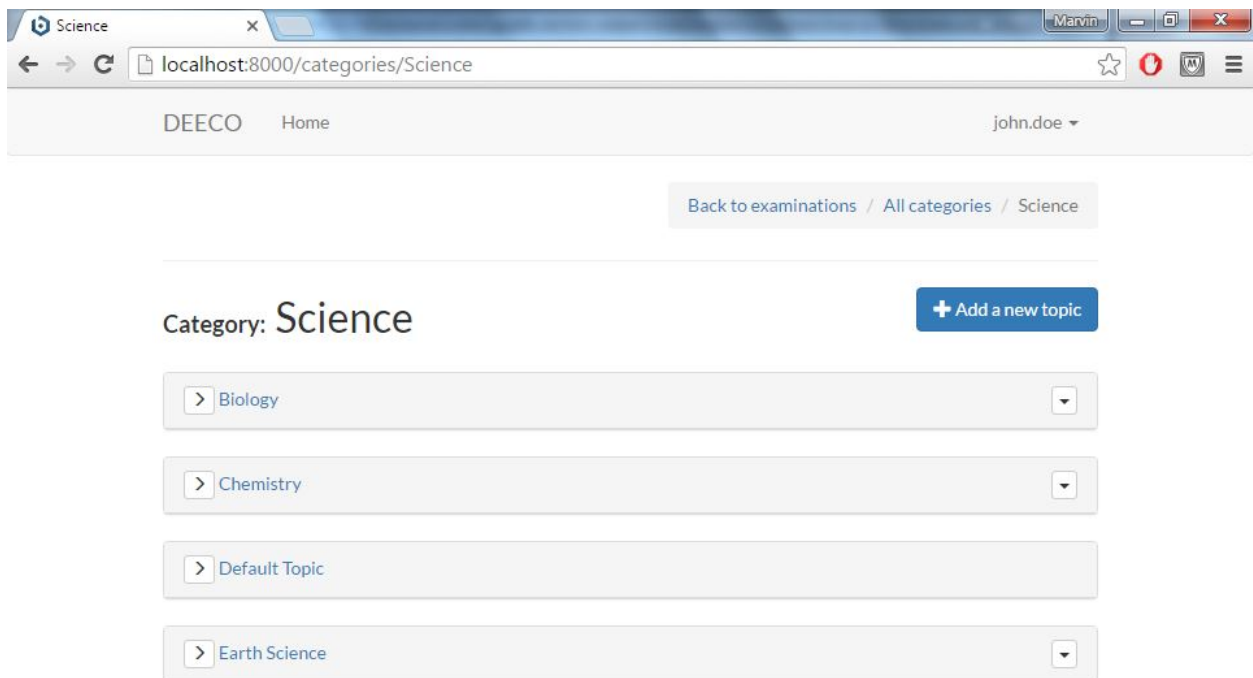


Figure 22: List of question topics

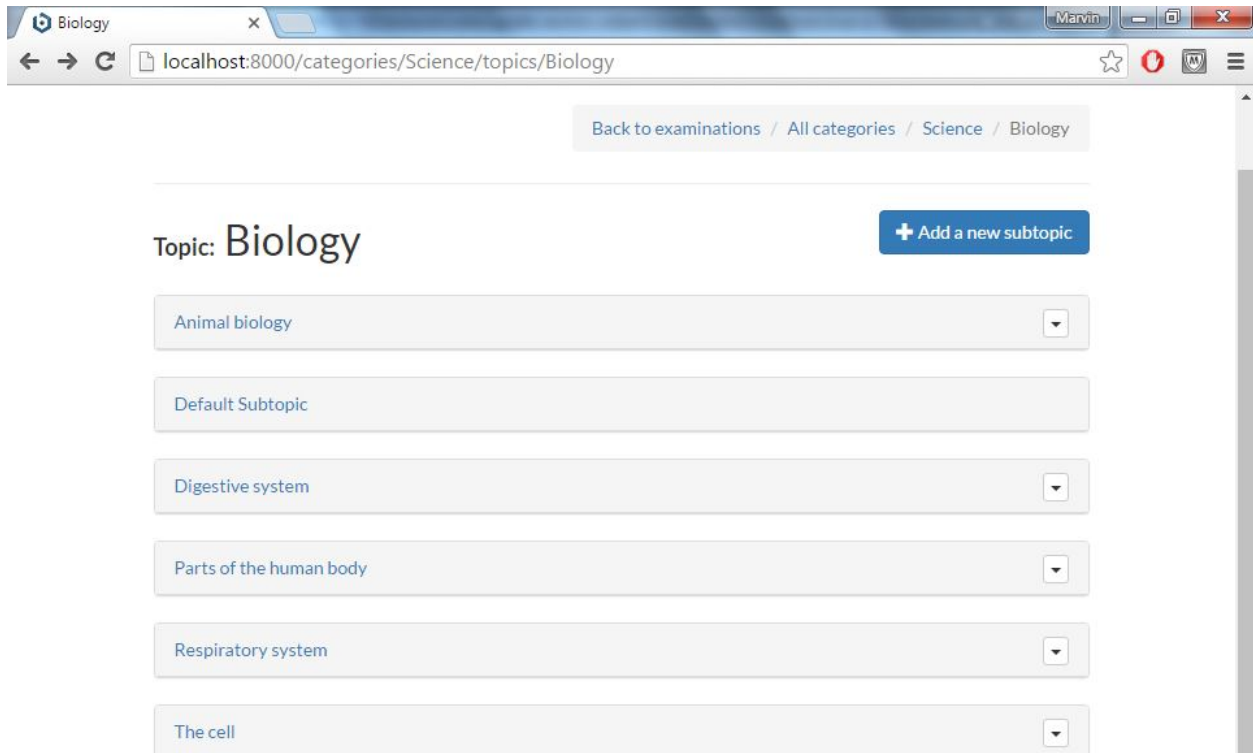


Figure 23: List of question subtopics

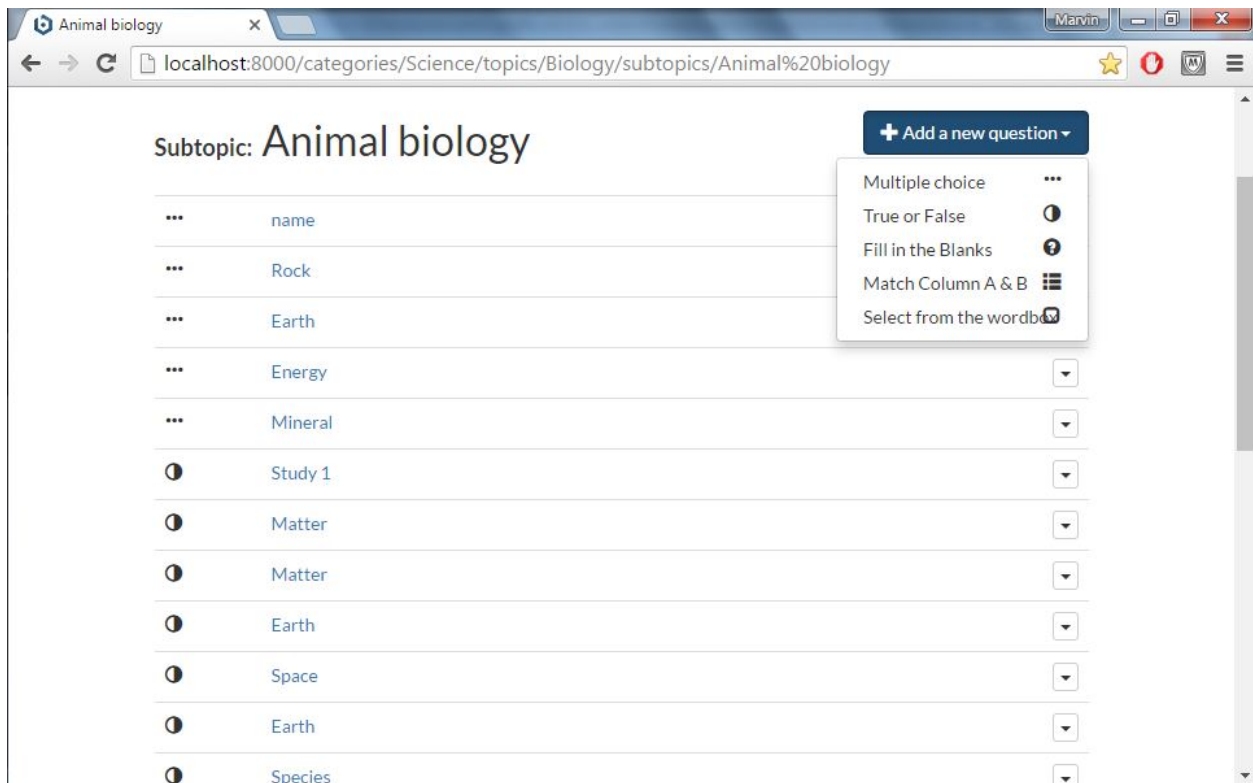


Figure 24: List of questions under the selected subtopic

D. Student Pages



Figure 25: Home page of the student upon login

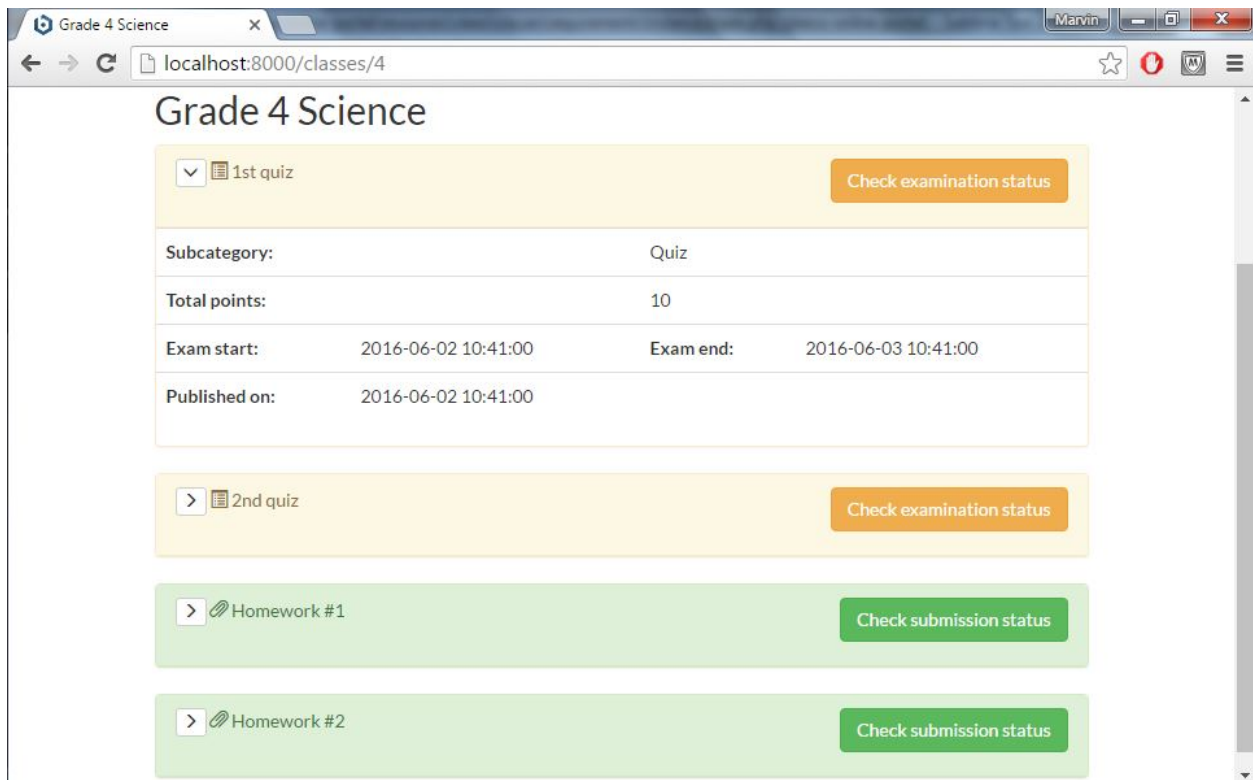


Figure 26: Showing the content of the grade section subject

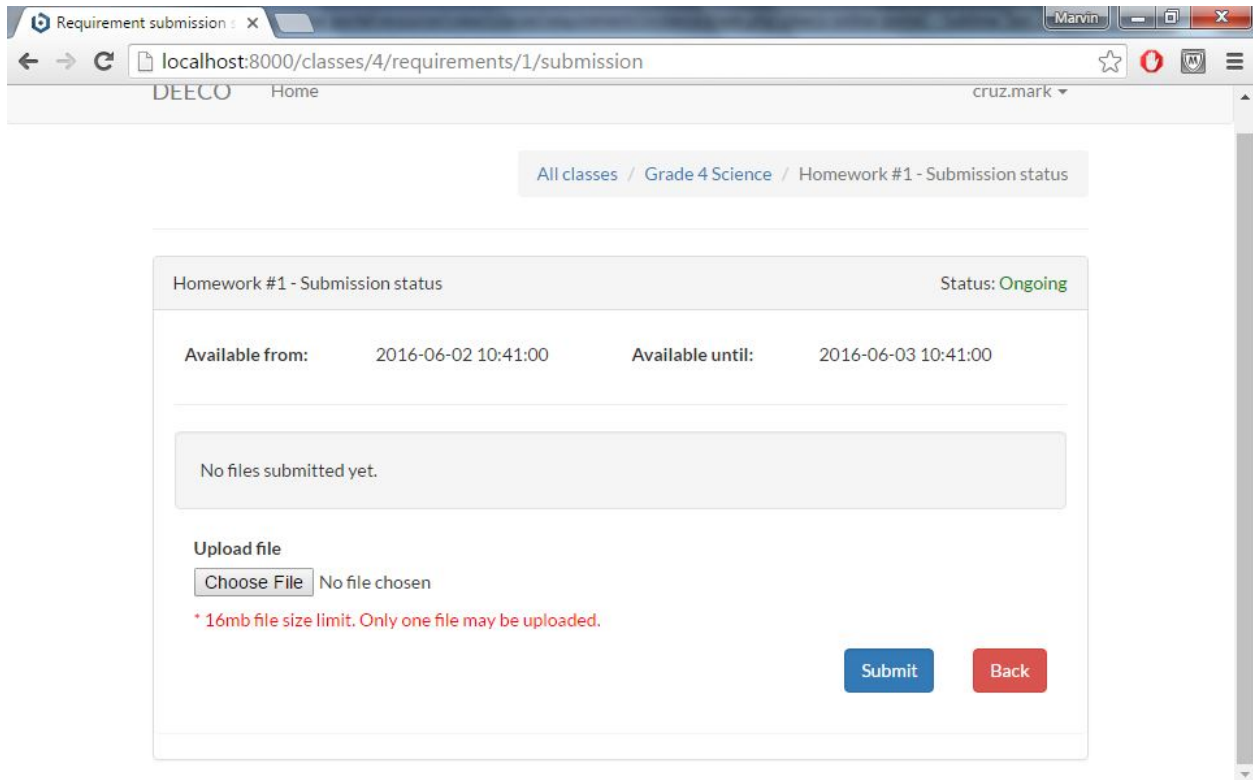


Figure 27: A student submitting a file in response to a faculty’s demand

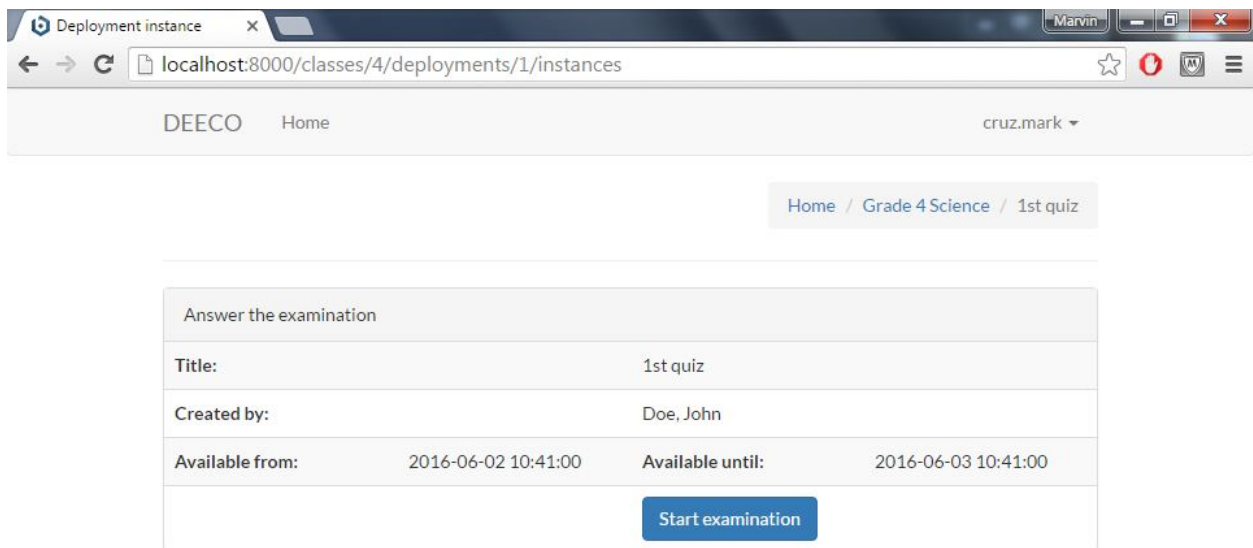


Figure 28: Confirmation before taking an examination

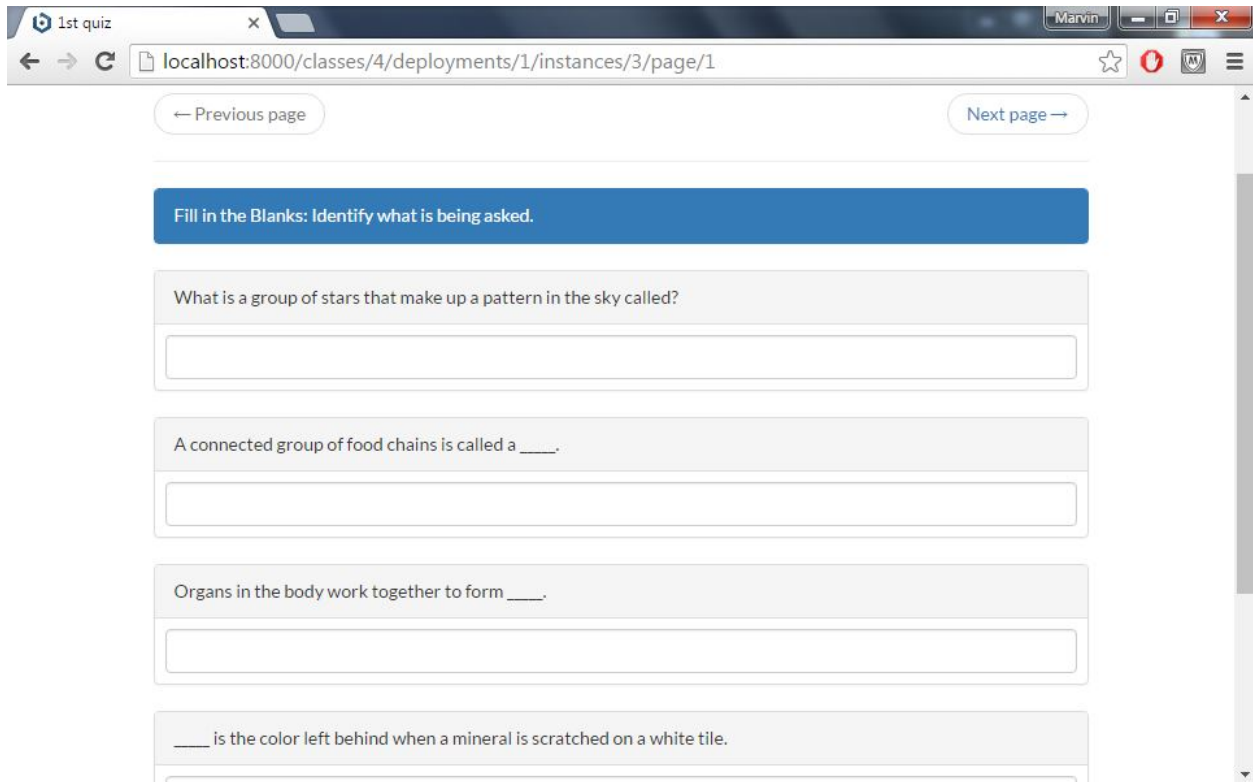


Figure 29: Answering “fill in the blanks” questions

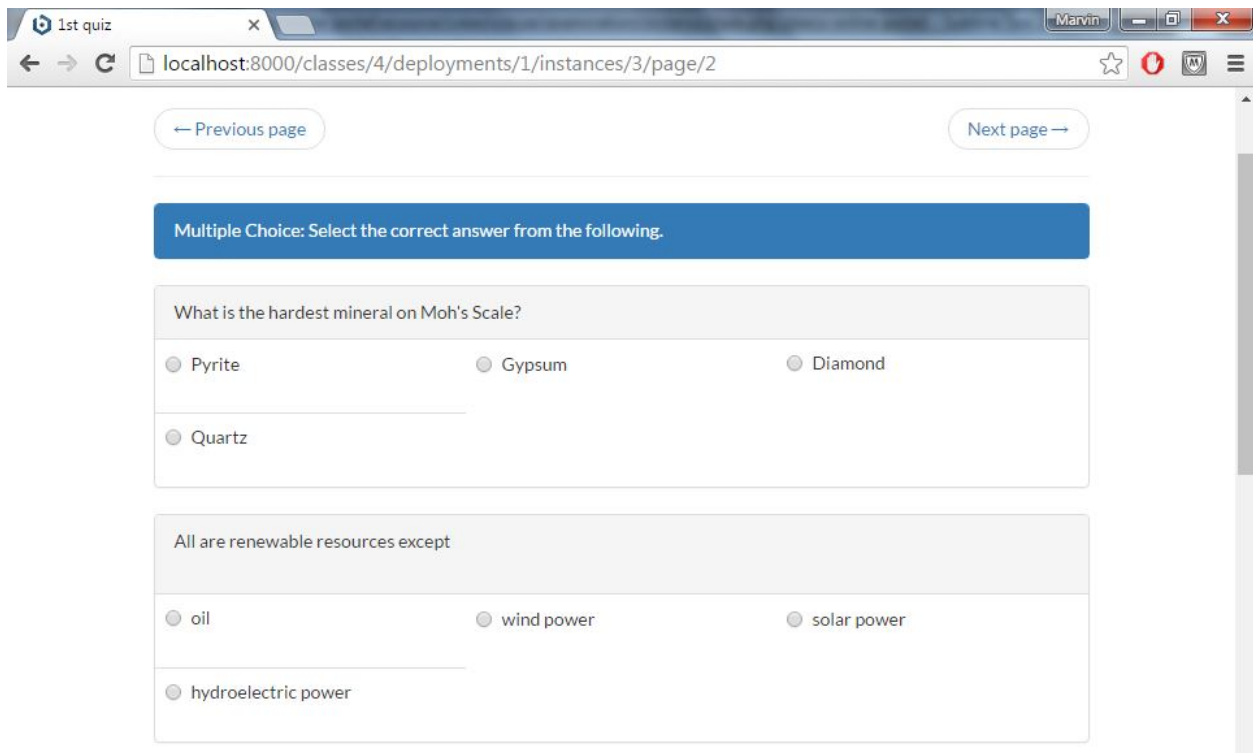


Figure 30: Answering “multiple choice” questions

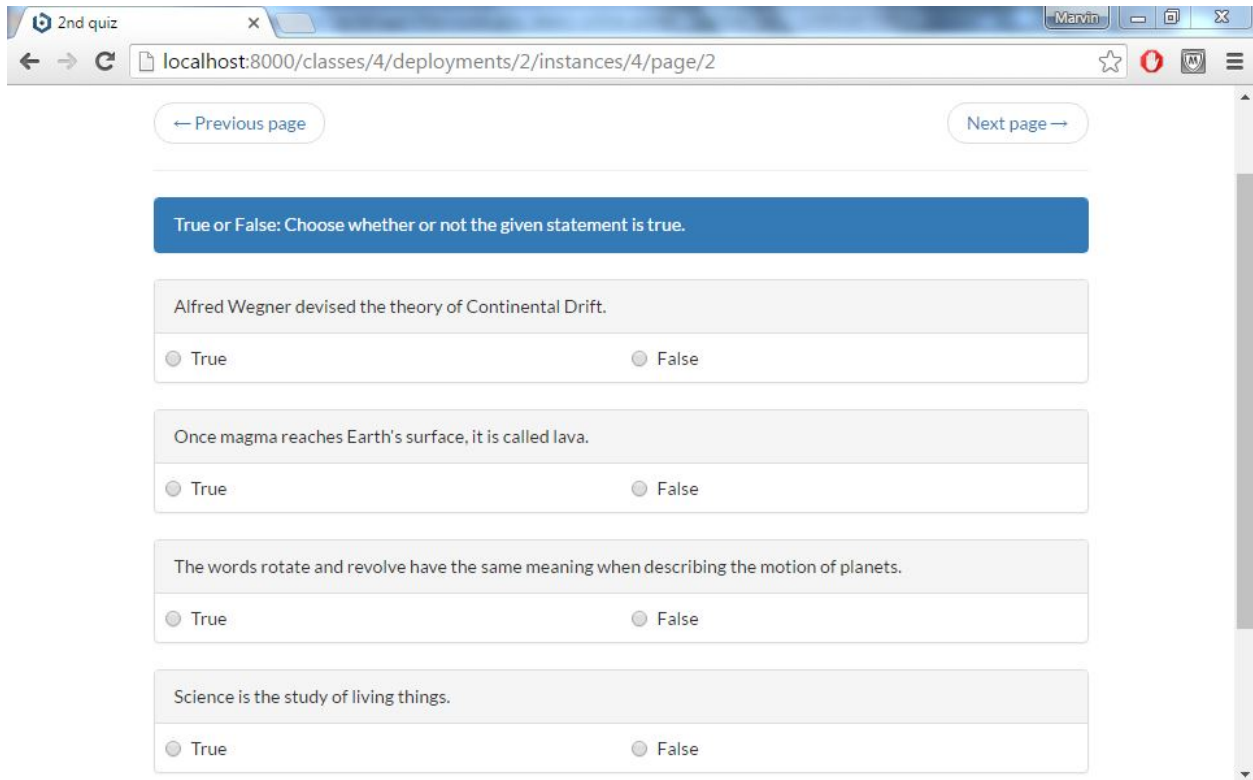


Figure 31: Answering “true or false” questions

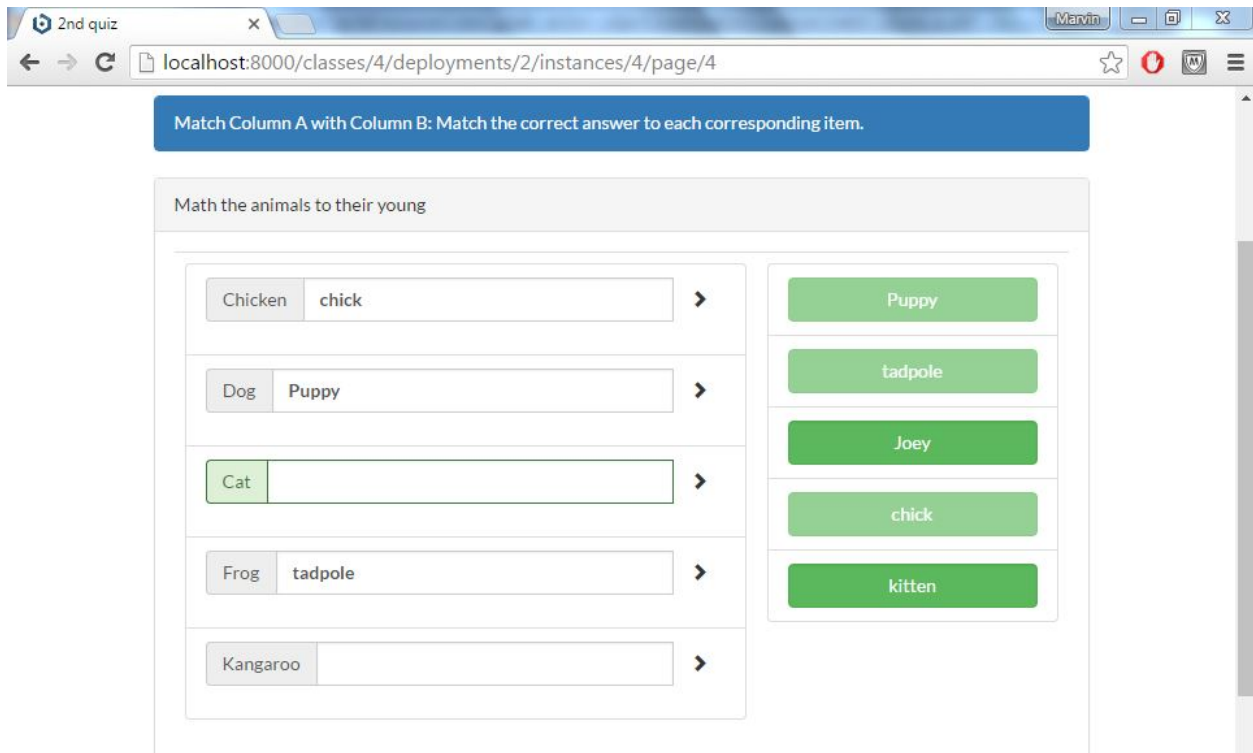


Figure 32: Answering “match column a with column b” questions

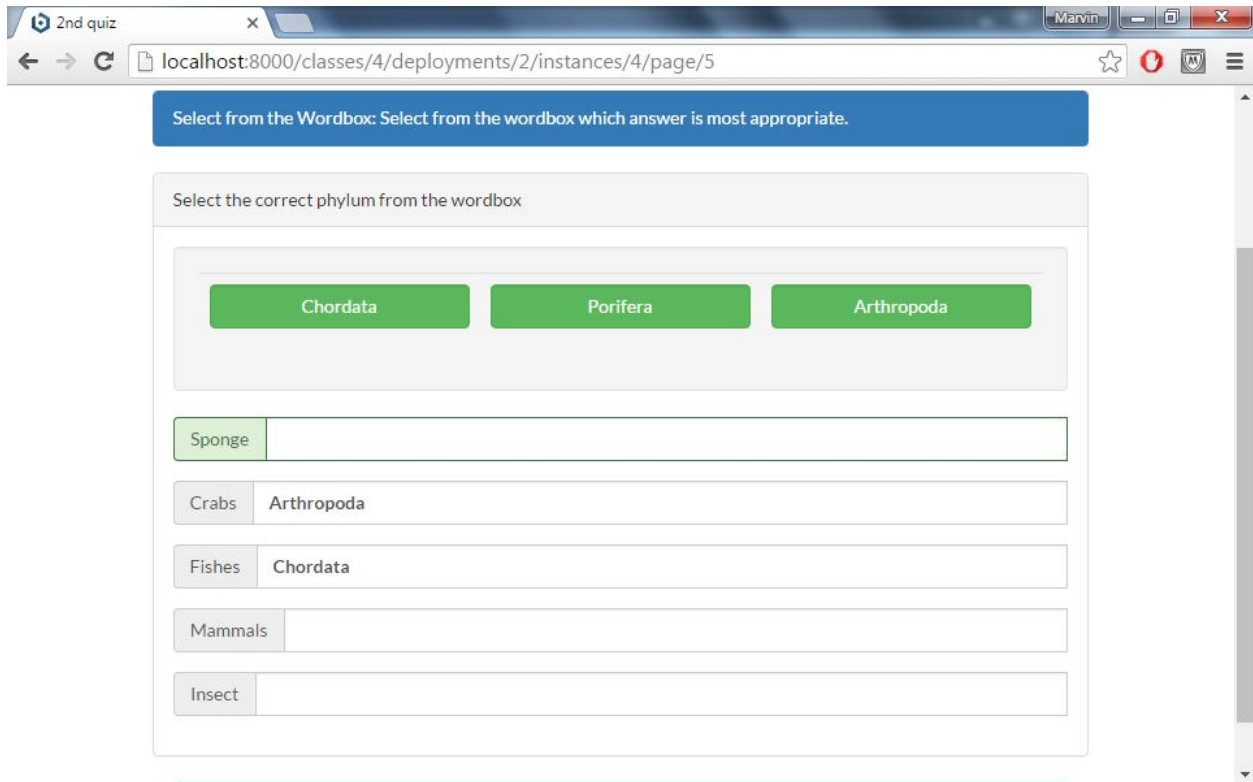


Figure 33: Answering “select from the wordbox” questions

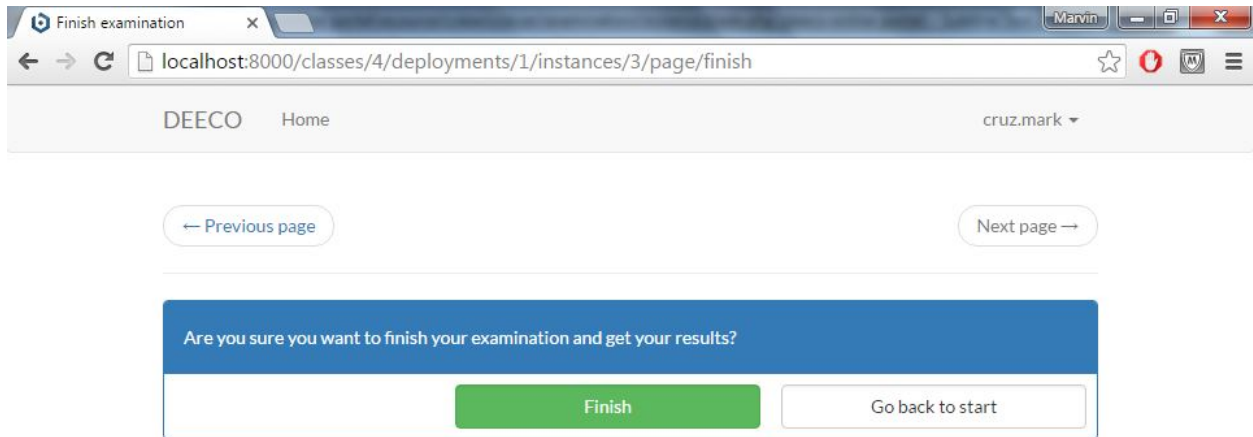


Figure 34: Confirmation before finishing the examination

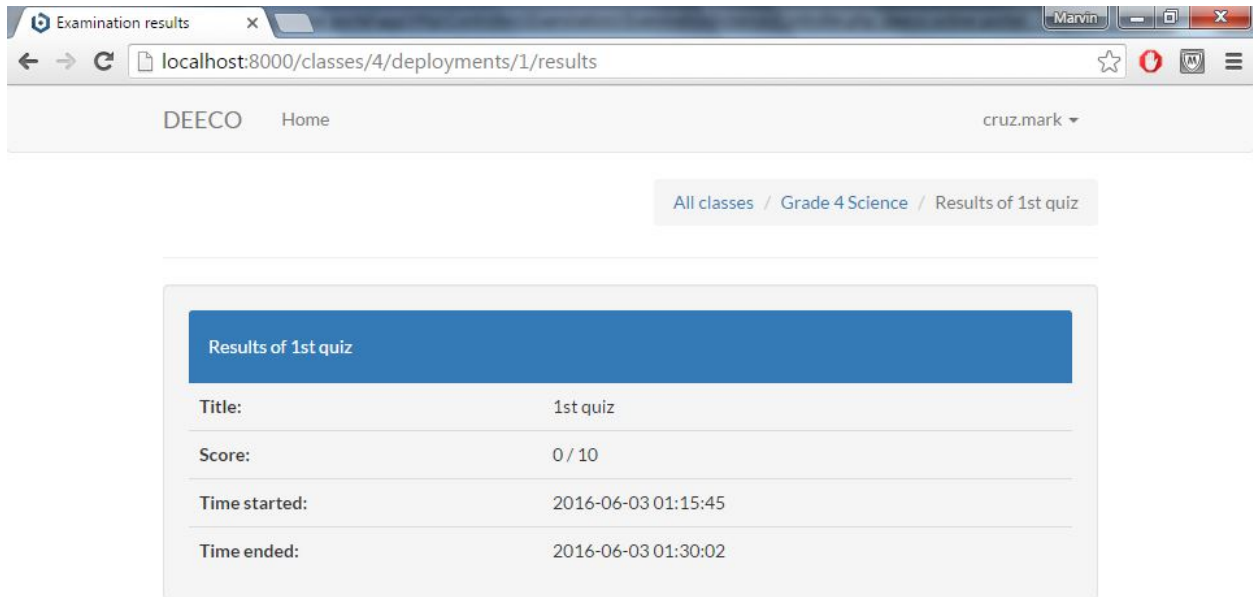


Figure 35: Showing the result to the student

The screenshot shows a web browser window with the address bar containing 'localhost:8000/subjects/4/class-record'. The page title is 'View class record'. Below the browser window, there is a table with the following structure:

	Quizes	Long test	Others
	<input type="text"/>	<input type="text"/>	
Carpio, Jeremiah (00-0349)			
Cayron, Ivan (00-1125)			
Chong, Samantha (00-1124)			
Chu, Abigail (00-0633)	12		
Corpuz, Emmanuel (00-0366)			
Cruz, Jared (00-0344)			
Cruz, Mark (00-1180)	0		
Cruz, Michaela			

The score '0' for Cruz, Mark is circled in red.

Figure 36: Showing the subject class record. The encircled score is the newly inserted score from the previous examination

VI. Discussion

The Dee Hwa Liong Academy Virtual Classroom is an online (although also accessible offline) student and faculty learning portal. The aim of the system is to create a bridge between students and faculty whenever direct confrontation is not applicable. The system also patterns existing learning portals such as Moodle, Genyo, and Blackboard, but is customized to suit the requirements of the client, Dee Hwa Liong Academy.

Like most learning portals already in existence, the system focuses on the faculty-student relationship, particularly when taking examinations. In real life, the faculty would create different kinds of examinations with varying levels of difficulty, as well as different percentages allotted to them when tallying the final grade. The system has implemented the said relationship, as we can see on the system.

The system could act as an online dropbox for both students and faculty. The system would greatly improve dissemination of lectures, syllabus, and other files the faculty would like to share with the class. This would also be a great alternative in requiring submissions from students.

The system would also be able to create examinations for students to answer. The system has achieved its goal of making sure that randomness plays a key part in generation of exams to minimize chances of students cheating on a test. The system has also achieved its goal of recording the scores of students in a given class record, to minimize human error in updating the class record.

VII. Conclusion

The objectives have been implemented in the system, although there had been a lot of changes from the original proposal. This is due to the fact that the developer has to change from using Moodle as its base framework to using Laravel.

The design has also been changed based on what the client has required. Nothing major has been deviated from the original design. The color scheme would also be customized based on the client's requirements.

The functionalities of the system administrator were also dramatically increased. The system administrator would now be able to add data to several tables which are essential especially during enrollment. This is to avoid the original assumption that the system would assume that some other data acquired during enrollment were already present. Adding said functionalities to the system administrator would enable the system to be usable without any external data to back it up.

The functionalities of the school management, although a bit simple, were also implemented. The school management would be able to view statistics of the website, as well as view other academic related data including but not limited to list of examinations the faculty has created, and the list of requirement on a given subject.

The system was most successful in implementing the functionalities of the faculty. The faculty was able to create posts, requirements, and most importantly, examinations. The faculty can also view the answers submitted by the user to double check the validity of their scores. The faculty can also view the requirements that are submitted by the students in each respective class.

Furthermore, the faculty can also contribute to the question bank, where all other faculty members can access the same questions to use in their respective exams.

The process of creating questions was also successfully made, although a few parameters have to be changed to fulfill the requirements of the user. What originally was four types of questions was changed to five types of questions, where the type “Matching Type” was split into two types of question: “Select from the Wordbox” and “Match Column A with Column B”.

The system, which was focused on the implementation of the examinations, was successful in doing so. The questions that would make up the examinations would randomly be selected, to fulfill the requirement of the client that exams would have to be as unique per users as possible, while at the same time having the guarantee that the examinations created are reliable and credible. The examinations were automatically checked and recorded in the subject class record, as originally proposed.

The functionalities of the user were also implemented. They can upload requirements demanded by the faculty, and they can only access the subjects they are enrolled in. They can answer the examinations and their scores would automatically be recorded on the class record of the faculty. The students would also be prevented from changing their uploaded files in the faculty’s requirements as well as their answer for a given examination whenever they exceeded the time limit.

VIII. Recommendations

Since the system only uses five forms of question, it is recommended to future developers who will improve the system to add more question types. Question types involving calculation, like problem solving, can be created, as well as question types that would not be immediately checked by the system but would require the direct interaction with the faculty, like essay types.

Also, since Laravel is used, one can also maximize the other objects that are not used in this project. Laravel has much custom functionality that can be implemented to further customize the system according to the specifications required by the user. Any future developers may also rewrite the way the controllers handle requests and calls other objects like services, middlewares, or models, when the said developer has any plan to expand the system.

Lastly, the developer may also improve on the design of the website. The current developer has not covered much in CSS and form designs, so the future developer may use CSS aside from basic bootstrap. They can also utilize using node.js, which could be very useful in creating custom designs for the website.

IX. Bibliography

- [1] I. Elaine Allen and Jeff Seaman, “Five Years of Growth in Online Learning”, p. 4, 2007
- [2] A. Ya Ni, “Comparing the Effectiveness of Classroom and Online Learning: Teaching Research Methods”, California State University–San Bernardino, *Journal of Public Affairs Education*, p. 199-215, 2013
- [3] I. Elaine Allen and Jeff Seaman, “Five Years of Growth in Online Learning”, p. 17, 2007
- [4] DEECO’s official Facebook page: <https://www.facebook.com/Deecoacademy/>
- [5] Maria Teresa F. Calderon, Ph.D., Arellano University, Manila, Philippines, “A Critique of K-12 Philippine Education System“, 2014
- [6] De Justo, H.A., Digal, E.J., Lagura, G., University of Southeastern Philippines, “An Analysis of the Policy: K-12 Education Program”, p. 2, 9, March 2012
- [7] DiPietro, M., Ferdig, R.E., Black, E.W., Preston, M., University of Florida, Gainesville, “Best practices in teaching K-12 online: Lessons learned from Michigan Virtual School teachers”, 2008
- [8] J. Lynn McBrien and Phyllis Jones, “Virtual Spaces: Employing a Synchronous Online Classroom to Facilitate Student Engagement in Online Learning”, June 2009
- [9] Rena M. Palloff and Keith Pratt, “Building Online Learning Communities: Effective Strategies for the Virtual Classroom, Second Edition of Building Learning Communities in Cyberspace”, 2007
- [10] Sitzmann, T, Kraiger, K., Stewart, D., Wisher, R., “The Comparative Effectiveness of Web-based and Classroom Instruction: A Meta-Analysis”, June 2006

- [11] Shu-Hui Hsieh Chang and Roger A. Smith, Iowa State University, “Effectiveness of Personal Interaction in a Learner-Centered Paradigm Distance Education Class Based on Student Satisfaction”, February 2014
- [12] Merchant, Z., Goetz, E.T., Cifuentes, L., Keeney-Kennicutt, W., Davis, T.J., “Effectiveness of virtual reality-based instruction on students’ learning outcomes in K-12 and higher education: A Meta-Analysis”, July 2013
- [13] Ju-Ling Shih , Chien-Wen Chuang² and Gwo-Jen Hwang, “An Inquiry-based Mobile Learning Approach to Enhancing Social Science Learning”, 2010
- [14] Manalang, M.A., “Tutorial and Online Examination System”, October 2006
- [15] Definition of Information Systems by techtarget.com:
<http://whatis.techtarget.com/definition/IS-information-system-or-information-services>
- [16] Definition of Information Systems by pcmag.com:
<http://www.pcmag.com/encyclopedia/term/44963/information-system>
- [17] Definition of MySQL by mysql.com: <https://dev.mysql.com/doc/refman/4.1/en/what-is-mysql.html>
- [18] Laravel official documentation site:
<https://laravel.com/docs/4.2/introduction>
- [19] PHP Documentation website: <http://php.net/docs.php>
- [20] JavaScript Definition by Mozilla: https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript
- [21] MVC Framework definition: <http://whatis.techtarget.com/definition/model-view-controller-MVC>

X. Appendix

App\Http\Controllers\AdminFunctionalities\FacultyLoadingsController.php

```
<?php
```

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

```
use Illuminate\Http\Request;
```

```
use App\Http\Controllers\Controller;
```

```
use App\Http\Requests;
```

```
use App\Models\User;
```

```
use App\Models\GradeSections\FacultyLoading;
```

```
use App\Models\GradeSections\Subject;
```

```
use App\Models\GradeSectionSubjects\GradeSectionSubject;
```

```
use DB;
```

```
class FacultyLoadingsController extends Controller
```

```
{
```

```
    public function index(){
```

```
        $facultyList = User::where('role_id', '3')->where('active', '1')->get();
```

```
        return view('grade-sections.faculty-loadings.index', compact('facultyList'));
```

```
    }
```

```
    public function show(User $user){
```

```
        $activeSchoolYear = DB::table('school_years')->where('active', '1')->first();
```

```
        $facultyLoadings = FacultyLoading::where('user_id', $user->id)->where('school_year_id',
```

```
$activeSchoolYear->id)->get();
```

```
        return view('grade-sections.faculty-loadings.show', compact('user', 'facultyLoadings'));
```

```
    }
```

```
    public function assignment($facultySelected = null, $gradeSectionSubjectsAssignedToFaculty = []){
```

```
        $activeSchoolYear = DB::table('school_years')->where('active', '1')->first();
```

```
        $gradeSectionSubjectList = GradeSectionSubject::orderBy('subject_id')->get();
```

```
        $gradeSectionSubjects = [];
```

```
        foreach ($gradeSectionSubjectList as $gradeSectionSubject){
```

```
            $facultyLoading = FacultyLoading::where('grade_section_subject_id', $gradeSectionSubject->id)->where('school_year_id', $activeSchoolYear->id)->first();
```

```
            if ($facultyLoading == null){
```

```
                $gradeSectionSubjects[] = $gradeSectionSubject;
```

```
            }
```

```
        }
```

```
        $facultyList = User::where('role_id', '3')->where('active', '1')->get();
```

```

        return view('grade-sections.faculty-loadings.assignment', compact('gradeSectionSubjects',
'facultyList', 'facultySelected', 'gradeSectionSubjectsAssignedToFaculty'));
    }

    public function postSearch(Request $request){
        $activeSchoolYear = DB::table('school_years')->where('active','1')->first();
        $faculty = User::find($request->input('facultyId'));
        $gradeSectionSubjectIds = FacultyLoading::where('user_id', $faculty->id)->where('school_year_id',
$activeSchoolYear->id)->pluck('grade_section_subject_id');
        $gradeSectionSubjects = [];
        foreach ($gradeSectionSubjectIds as $id){
            $gradeSectionSubjects[] = GradeSectionSubject::find($id);
        }
        return $this->assignment($faculty, $gradeSectionSubjects);
    }

    public function assign(Request $request){
        $activeSchoolYear = DB::table('school_years')->where('active','1')->first();
        $faculty = User::find($request->input('assignFaculty'));
        $selectedIds = explode(',', $request->input('assignList'));
        if ($selectedIds[0] == ""){
            $faculty->gradeSectionSubjects()->detach();
        } else {
            $faculty->gradeSectionSubjects()->sync($selectedIds);
        }
        $facultyLoadings = FacultyLoading::whereIn('grade_section_subject_id', $selectedIds)-
>where('user_id', $faculty->id)->get();
        if (!$facultyLoadings->isEmpty()){
            foreach ($facultyLoadings as $facultyLoading){
                $facultyLoading->school_year_id = $activeSchoolYear->id;
                $facultyLoading->update();
            }
        }
        return redirect('faculty-loadings/' . $faculty->id);
    }
}
}

```

App\Http\Controllers\AdminFunctionalities\GradeSectionsController.php

<?php

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

```
use Illuminate\Http\Request;
```

```
use App\Http\Controllers\Controller;
```

```

use App\Http\Requests;
use App\Models\GradeSections\GradeSection;
use App\Models\GradeSections\Subject;
use DB;

class GradeSectionsController extends Controller
{
    public function index(){
        $gradeSections = GradeSection::all();
        return view('grade-sections.index', compact('gradeSections'));
    }

    public function show(GradeSection $gradeSection){
        return view('grade-sections.show', compact('gradeSection'));
    }

    public function assignment($gradeSectionSelected = null, $subjectsAssignedInSection = []){
        $activeSchoolYear = DB::table('school_years')->where('active','1')->first();
        $subjectIdsAssignedInSection = [];
        foreach ($subjectsAssignedInSection as $subject){
            $subjectIdsAssignedInSection[] = $subject->id;
        }
        $subjects = Subject::get();
        $gradeSections = GradeSection::where('school_year_id', $activeSchoolYear->id)->get();
        return view('grade-sections.subjects.assignment', compact('subjects', 'gradeSections',
'gradeSectionSelected', 'subjectIdsAssignedInSection', 'subjectsAssignedInSection'));
    }

    public function postSearch(Request $request){
        $gradeSection = GradeSection::find($request->input('gradeSectionId'));
        $subjects = $gradeSection->subjects;
        return $this->assignment($gradeSection, $subjects);
    }

    public function assign(Request $request){
        $gradeSection = GradeSection::find($request->input('assignSection'));
        $selectedIds = explode(',', $request->input('assignList'));
        $gradeSection->subjects()->sync($selectedIds);
        return redirect('/grade-sections/' . $gradeSection->id);
    }
}

```


App\Http\Controllers\AdminFunctionalities\SchoolYearsController.php

<?php

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

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests;
use App\Models\GradeSections\SchoolYear;
use App\Models\GradeSections\GradeSection;
use DB;

class SchoolYearsController extends Controller
{
    public function create(){
        return view('grade-sections.school-years.create');
    }

    public function store(Request $request){
        $schoolYear = SchoolYear::create([
            'name' => $request->input('name'),
            'active' => '1'
        ]);
        $gradeSectionNames = DB::table('grade_section_names')->get();
        foreach ($gradeSectionNames as $gradeSectionName){
            GradeSection::create([
                'grade_section_name_id' => $gradeSectionName->id,
                'school_year_id' => $schoolYear->id,
            ]);
        }
        return redirect('/home');
    }
}
```

App\Http\Controllers\SubjectsController.php

<?php

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

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests;
use App\Models\GradeSections\Subject;
```

```

class SubjectsController extends Controller
{
    public function index(){
        $subjects = Subject::orderBy('name', 'asc')->get();
        return view('grade-sections.subjects.index', compact('subjects'));
    }

    public function create(){
        return view('grade-sections.subjects.create');
    }

    public function store(Request $request){
        Subject::create([
            'name' => $request->input('name'),
        ]);
        return redirect('/others/subjects');
    }
}

```

app\Http\Controllers\Auth\AuthController.php

```
<?php
```

```

namespace App\Http\Controllers\Auth;

use App\Models\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;

```

```

/**
 * Where to redirect users after login / registration.
 *
 * @var string
 */
protected $redirectTo = '/home';

/**
 * Create a new authentication controller instance.
 *
 * @return void
 */
public function __construct()
{
    //$this->middleware('role:System Administrator', ['only' => ['getRegister', 'showRegistrationForm',
'register']]);
    $this->middleware('guest', ['except' => ['logout', 'getRegister', 'showRegistrationForm', 'register']]);
}

/**
 * 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, [
        'username' => 'required|max:100',
        'password' => 'required|confirmed|min:6|max:255',
        'role_id' => 'required',
    ], $messages = [
        'role_id.required' => 'Please specify role of user.',
    ]);
}

/**
 * Create a new user instance after a valid registration.
 *
 * @param array $data
 * @return User
 */
protected function create(array $data)

```

```

    {
        return \App\Models\User::create([
            'username' => $data['username'],
            'password' => bcrypt($data['password']),
            'role_id' => $data['role_id'],
        ]);
    }
}

```

App\Http\Controllers\Auth

<?php

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

```

use App\Http\Controllers\Controller;
use App\Http\Requests>PasswordChangeRequest;
use Illuminate\Foundation\Auth\ResetsPasswords;
use Auth;
use Hash;
use Random;

```

```
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('auth', ['only' => ['showPasswordChangeForm', 'changePassword']]);

```

```

        $this->middleware('guest', ['except' => ['showPasswordChangeForm', 'changePassword']]);
    }

    public function showPasswordChangeForm(){
        $randKey = Random::generateString(16,
'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ');
        return view('auth.passwords.change', compact('randKey'));
    }

    public function changePassword(PasswordChangeRequest $request){
        $user = Auth::user();
        $oldPassword = $request->input('oldPassword');
        if (Hash::check($oldPassword, $user->password)){
            $user->password = bcrypt($request->input('newPassword'));
            $user->update();
            flash()->success('Your password has been successfully changed. ');
            return redirect('/home');
        } else {
            return redirect('password/change')->withErrors(['oldPassword' => 'The value you have entered
does not match your current password.']);
        }
    }
}
}
}

```

App\Http\Controllers\Examinations\ExaminationInstancesController.php

<?php

```

namespace App\Http\Controllers\Examinations;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests;
use App\Http\Services\QuestionsService;
use App\Http\Services\ExaminationsService;
use App\Http\Services\ClassRecordService;
use App\Models\Questions\Question;
use App\Models\Questions\Types\QuestionMultipleChoice;
use App\Models\Questions\Types\QuestionTrueOrFalse;
use App\Models\Questions\Types\QuestionFillInTheBlanks;
use App\Models\Questions\Types\QuestionSelectFromTheWordboxChoice;
use App\Models\Questions\Types\QuestionSelectFromTheWordboxItem;
use App\Models\Questions\Types\QuestionMatchColumnsChoice;
use App\Models\Questions\Types\QuestionMatchColumnsItem;
use App\Models\GradeSectionSubjects\GradeSectionSubject;

```

```

use App\Models\Examinations\Examination;
use App\Models\Examinations\Deployment;
use App\Models\Examinations\DeploymentInstance;
use App\Models\Examinations\DeploymentAnswer;
use Carbon\Carbon;
use Auth;

class ExaminationInstancesController extends Controller
{
    private $questionsService;
    private $examinationsService;
    private $classRecordService;

    public function __construct(QuestionsService $questionsService, ExaminationsService
    $examinationsService, ClassRecordService $classRecordService){
        $this->questionsService = $questionsService;
        $this->examinationsService = $examinationsService;
        $this->classRecordService = $classRecordService;
        $this->middleware(['examFinished', ['only' => ['showExamPage', 'processExamPage',
'showExamFinishPage', 'finish']]);
    }

    public function showInstanceConfirmation(GradeSectionSubject $gradeSectionSubject = null,
    Deployment $deployment){
        $instance = DeploymentInstance::where('user_id', Auth::user()->id)->where('deployment_id',
    $deployment->id)->first();
        if ($instance == null){
            return view('classes.examinations.instance', compact('gradeSectionSubject', 'deployment'));
        } else if ($instance->is_finished){
            return redirect('/classes/' . $gradeSectionSubject->id . '/deployments/' . $deployment->id .
'/results');
        } else if (!$instance->is_finished){
            return redirect('/classes/' . $gradeSectionSubject->id . '/deployments/' . $deployment->id .
'/instances/' . $instance->id . '/page/1');
        } else {
            flash()->error('Some error occurred. Please try again. ');
            return redirect('/home');
        }
    }

    public function createExaminationInstance(GradeSectionSubject $gradeSectionSubject, Deployment
    $deployment, Request $request){
        if (!$this->examinationsService->verifyQuestionQuantity($deployment->examination)){
            dd('questions not enough');
        }
    }
}

```

```

} else{
    $deploymentInstance = $this->createInstance($deployment);
    foreach ($deployment->examination->parts as $examPart){
        $questionTypeId = $examPart->question_type_id;
        $first = true;
        $questionsOrder = "";
        if (!strcmp($examPart->getQuestionType(), 'Select from the Wordbox')){
            $questionItem = $examPart->items()->first();
            $question = Question::find($questionItem->question_id);
            $wordboxChoiceIds = QuestionSelectFromTheWordboxChoice::get()->pluck('id');
            $wordboxItems = QuestionSelectFromTheWordboxItem::whereIn('wordbox_choice_id',
$wordboxChoiceIds)->get()->shuffle();
            for ($i = 0; $i < $examPart->items->first()->quantity; $i++){
                if (!$first){
                    $questionsOrder = $questionsOrder . '|' . $wordboxItems[$i]->id;
                } else {
                    $questionsOrder = " . $wordboxItems[$i]->id;
                    $first = false;
                }
            }
        }
        } else if (!strcmp($examPart->getQuestionType(), 'Match Column A with Column B')){
            $questionItem = $examPart->items()->first();
            $question = Question::find($questionItem->question_id);
            $columnsChoiceIds = QuestionMatchColumnsChoice::get()->pluck('id');
            $columnsItems = QuestionMatchColumnsItem::whereIn('columns_choice_id',
$columnsChoiceIds)->get()->shuffle();
            for ($i = 0; $i < ($questionItem->quantity + $questionItem->choices_quantity); $i++){
                if (!$first){
                    $questionsOrder = $questionsOrder . '|' . $columnsItems[$i]->id;
                } else {
                    $questionsOrder = " . $columnsItems[$i]->id;
                    $first = false;
                }
            }
        }
        } else {
            foreach ($examPart->items as $item){
                $questions = Question::where('question_type_id', $questionTypeId)-
>where('question_subtopic_id', $item->question_subtopic_id)->get()->shuffle();
                for ($i = 0; $i < $item->quantity; $i++){
                    if (!$first){
                        $questionsOrder = $questionsOrder . '|' . $questions[$i]->id;
                    } else {
                        $questionsOrder = " . $questions[$i]->id;
                        $first = false;
                    }
                }
            }
        }
    }
}

```

```

        }
    }
}
$deploymentInstance->parts()->create([
    'examination_part_id' => $examPart->id,
    'question_order' => $questionsOrder,
]);
}
}
return redirect('/classes/' . $gradeSectionSubject->id . '/deployments/' . $deployment->id .
'/instances/' . $deploymentInstance->id . '/page/1');
}

public function showExamPage(GradeSectionSubject $gradeSectionSubject, Deployment
$deployment, DeploymentInstance $instance, $page){
    if ($page > count($instance->parts)){
        return redirect('/classes/' . $gradeSectionSubject->id . '/deployments/' . $deployment->id .
'/instances/' . $instance->id . '/page/finish');
    } else {
        $instancePart = $instance->parts[$page-1];
        $questionIds = explode('|', $instancePart->question_order);
        $choices_quantity = 0;
        foreach ($questionIds as $id){
            if (!strcmp($instancePart->examinationPart->getQuestionType(), 'Select from the
Wordbox')){
                $questions[$id] = QuestionSelectFromTheWordboxItem::find($id);
            } else if (!strcmp($instancePart->examinationPart->getQuestionType(), 'Match Column
A with Column B')){
                $questions[$id] = QuestionMatchColumnsItem::find($id);
                $choices_quantity = $deployment->examination->parts[$page-1]->items()-
>first()->choices_quantity;
            } else {
                $questions[$id] = Question::find($id);
            }
        }
        $navbar = $this->createNavbar($gradeSectionSubject, $deployment, $instance, $page);
        $submitUrl = '/classes/' . $gradeSectionSubject->id . '/deployments/' . $deployment->id .
'/instances/' . $instance->id . '/page/' . $page;
        return $this->examinationsService->generatePage($instancePart, $questions, $navbar,
$submitUrl, $choices_quantity);
    }
}
}

```



```

public function processExamPage(GradeSectionSubject $gradeSectionSubject, Deployment
$deployment, DeploymentInstance $instance, $page, Request $request){
    $instancePart = $instance->parts[$page-1];
    $questionIds = explode('|', $instancePart->question_order);
    foreach ($questionIds as $id){
        $questions[$id] = Question::find($id);
    }
    $nextUrl = '/classes/' . $gradeSectionSubject->id . '/deployments/' . $deployment->id .
'/instances/' . $instance->id . '/page/' . ($page + 1);
    return $this->examinationsService->processPage($instancePart, $questions, $nextUrl, $page,
$request);
}

```

```

public function showExamFinishPage(GradeSectionSubject $gradeSectionSubject, Deployment
$deployment, DeploymentInstance $instance){
    $navbar = $this->createNavbar($gradeSectionSubject, $deployment, $instance, 'finish');
    $nextUrl = '/classes/' . $gradeSectionSubject->id . '/deployments/' . $deployment->id . '/instances/' .
$instance->id . '/page/finish';
    $startUrl = '/classes/' . $gradeSectionSubject->id . '/deployments/' . $deployment->id . '/instances/' .
$instance->id . '/page/1';
    return view('grade-section-subjects.examinations.instances.finish', compact('navbar', 'nextUrl',
'startUrl'));
}

```

```

public function finish(GradeSectionSubject $gradeSectionSubject, Deployment $deployment,
DeploymentInstance $instance){
    $score = 0;
    foreach ($instance->parts as $instancePart){
        $questionType = $instancePart->examinationPart->getQuestionType();
        $questionIds = explode('|', $instancePart->question_order);
        if (!strcmp($questionType, 'Match Column A with Column B')) {
            $sch_quantity = $instancePart->examinationPart->items()->first()->choices_quantity;
            $questionIds = array_slice($questionIds, 0, count($questionIds) - $sch_quantity);
        }
        $pointsPerItem = $instancePart->examinationPart->points;
        foreach ($questionIds as $questionId){
            switch ($questionType){
                case 'Multiple Choice':
                    $question = Question::find($questionId);
                    if ($this->examinationsService->fetchAnswer($instancePart->id, $question->id) != null){
                        $answer = $this->examinationsService->fetchAnswer($instancePart->id, $question-
>id)->answer;
                        $correctAnswer = QuestionMultipleChoice::where('question_id', $question->id)-
>where('is_right_answer', '1')->first()->text;

```

```

        $score += (!strcmp($answer, $correctAnswer)) ? $pointsPerItem : 0;
    }
    break;
case 'True or False':
    $question = Question::find($questionId);
    if ($this->examinationsService->fetchAnswer($instancePart->id, $question->id) != null){
        $answer = $this->examinationsService->fetchAnswer($instancePart->id, $question-
>id)->answer;
        $correctAnswer = QuestionTrueOrFalse::where('question_id', $question->id)->first()-
>right_answer;
        $score += (!strcmp($answer, $correctAnswer)) ? $pointsPerItem : 0;
    }
    break;
case 'Fill in the Blanks':
    $question = Question::find($questionId);
    if ($this->examinationsService->fetchAnswer($instancePart->id, $question->id) != null){
        $answer = $this->examinationsService->fetchAnswer($instancePart->id, $question-
>id)->answer;
        $correctAnswer = QuestionFillInTheBlanks::where('question_id', $question->id)-
>first()->right_answer;
        $score += (!strcmp(strtolower($answer), strtolower($correctAnswer))) ? $pointsPerItem
: 0;
    }
    break;
case 'Select from the Wordbox':
    $wordboxItemId = $questionId;
    if (DeploymentAnswer::where('deployment_instance_part_id', $instancePart->id)-
>where('wordbox_item_id', $wordboxItemId)->first() != null){
        $answer = DeploymentAnswer::where('deployment_instance_part_id', $instancePart-
>id)->where('wordbox_item_id', $wordboxItemId)->first()->answer;
        $item = QuestionSelectFromTheWordboxItem::find($wordboxItemId);
        $correctAnswer = $item->choice->text;
        $score += (!strcmp($answer, $correctAnswer)) ? $pointsPerItem : 0;
    }
    break;
case 'Match Column A with Column B':
    $columnsItemId = $questionId;
    if (DeploymentAnswer::where('deployment_instance_part_id', $instancePart->id)-
>where('columns_item_id', $columnsItemId)->first()){
        $answer = DeploymentAnswer::where('deployment_instance_part_id', $instancePart-
>id)->where('columns_item_id', $columnsItemId)->first()->answer;
        $item = QuestionMatchColumnsItem::find($columnsItemId);
        $correctAnswer = $item->choice->text;
        $score += (!strcmp($answer, $correctAnswer)) ? $pointsPerItem : 0;
    }

```

```

        }
        break;
    default:
        flash()->error('Some error occurred. Please try again.');
```

return redirect('/home');

```

    }
}
}
$instance->time_ended = Carbon::now();
$instance->score = $score;
$instance->is_finished = true;
$instance->update();

$this->classRecordService->storeResultsInClassRecord($gradeSectionSubject, $deployment,
$instance);
return redirect('/classes/' . $gradeSectionSubject->id . '/deployments/' . $deployment->id . '/results');
```

}

```

private function createInstance($deployment){
    $deploymentInstance = new DeploymentInstance();
    $deploymentInstance->user_id = Auth::user()->id;
    $deploymentInstance->deployment_id = $deployment->id;
    $deploymentInstance->score = null;
    $deploymentInstance->time_started = Carbon::now();
    $deploymentInstance->time_ended = null;
    $deploymentInstance->is_finished = false;
    $deploymentInstance->save();
    return $deploymentInstance;
}

public function createNavbar($gradeSectionSubject, $deployment, $instance, $page){
    $navbar = "<nav><ul class='pager'>";
    if ($page == 1){
        $navbar = $navbar . "<li class='previous disabled'><a href='#'><span aria-
hidden='true'>&larr;</span> Previous page</a></li>";
    } else if ($page == 'finish'){
        $navbar = $navbar . "<li class='previous'><a href='/classes/" . $gradeSectionSubject->id .
"/deployments/" . $deployment->id . "/instances/" . $instance->id . "/page/" . count($instance->parts) .
"><span aria-hidden='true'>&larr;</span> Previous page</a></li>";
    } else {
        $navbar = $navbar . "<li class='previous'><a href='/classes/" .
$gradeSectionSubject->id . "/deployments/" . $deployment->id . "/instances/" . $instance->id . "/page/" .
($page - 1) . "><span aria-hidden='true'>&larr;</span> Previous page</a></li>";
    }
}

```

```

        if ($page == 'finish'){
            $navbar = $navbar . "<li class='next disabled'><a href='#>Next page <span aria-
hidden='true'>&rarr;</span></a></li>";
        } else {
            $navbar = $navbar . "<li class='next'><a href='/classes/" . $gradeSectionSubject-
>id . "/deployments/" . $deployment->id . "/instances/" . $instance->id . "/page/" . ($page + 1) . "'>
            Next page <span aria-
hidden='true'>&rarr;</span></a></li>";
        }
        $navbar = $navbar . "</ul></nav>";
    }
    return $navbar;
}
}
}

```

app\Http\Controllers\Examinations\ExaminationPartItemsController.php

<?php

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

```

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests;
use App\Http\Services\QuestionsService;
use App\Models\Questions\Question;
use App\Models\Questions\QuestionCategory;
use App\Models\Questions\QuestionTopic;
use App\Models\Questions\QuestionSubtopic;
use App\Models\Examinations\Examination;
use App\Models\Examinations\ExaminationPart;
use App\Models\Examinations\ExaminationPartItem;

```

```
class ExaminationPartItemsController extends Controller
```

```
{
```

```
    private $questionsService;
```

```
    public function __construct(QuestionsService $questionsService){
```

```
        $this->questionsService = $questionsService;
```

```
    }
```

```

    public function create(Examination $examination, ExaminationPart $part, $subtopics = [], $questions =
[], $selectedCategory = null, $selectedTopic = null, $questionCount = []){

```

```
        $categories = QuestionCategory::where('name', '<>', 'Default Category')->get();
```

```
        $topics = [];
```

```
        foreach($categories as $category){
```

```

        $topic_arr = [];
        foreach($category->questionTopics as $topic){
            if($topic->name != 'Default Topic'){
                $topic_arr[] = $topic;
            }
        }
        $topics[$category->id] = $topic_arr;
    }
    return view('grade-section-subjects.examinations.parts.items.create', compact('examination', 'part',
'categories', 'topics', 'subtopics', 'questions', 'selectedCategory', 'selectedTopic', 'questionCount'));
}

```

```

public function postCreateSearch(Examination $examination, ExaminationPart $part, Request
$request){
    $subtopics = QuestionSubtopic::where('name', '<>', 'Default Subtopic')->where('question_topic_id',
$request->input('topic'))->lists('name', 'id');
    $subtopicList = QuestionSubtopic::where('question_topic_id', $request->input('topic'))->get();
    $questions = [];
    $questionCount = [];
    foreach($subtopicList as $subtopic){
        $question_arr = [];
        $questionList = $subtopic->questions()->where('question_type_id', $part->question_type_id)-
>get();
        $questionCount[$subtopic->id] = count($questionList);
        foreach($questionList as $question){
            $question_arr[] = $question;
        }
        $questions[$subtopic->id] = $question_arr;
    }
    $selectedCategory = QuestionCategory::find($request->input('category'));
    $selectedTopic = QuestionTopic::find($request->input('topic'));
    return $this->create($examination, $part, $subtopics, $questions, $selectedCategory, $selectedTopic,
$questionCount);
}

```

```

public function store(Examination $examination, ExaminationPart $part, Request $request){
    $questionTypeId = $part->question_type_id;
    $subtopicId = $request->input('question_subtopic_id');
    $quantity = $request->input('quantity');
    if(!strcmp($part->getQuestionType(), 'Select from the Wordbox') || !strcmp($part-
>getQuestionType(), 'Match Column A with Column B')){
        $currentItem = $part->items()->first();
        if ($currentItem != null){
            $currentItem->delete();
        }
    }
}

```

```

    }
    $choices_quantity = $request->input('choices_quantity');
    $question_id = $request->input('question_id');
    $part->items()->create($request->all());
    return redirect('/examinations/' . $examination->id . '/parts/' . $part->id);
} else if($this->questionsService->verifyQuestionQuantityInSubtopic($questionTypeId, $subtopicId,
$quantity)){
    $part->items()->create($request->all());
    return redirect('/examinations/' . $examination->id . '/parts/' . $part->id);
} else {
    flash()->error('Cannot create this item. The number of questions in the database
satisfying the conditions is less than the quantity you demand.
Create more questions first or choose another subtopic. ');
    return redirect('/examinations/' . $examination->id . '/parts/' . $part->id . '/items/create');
}
}
}

```

```

public function showDeleteConfirmation(Examination $examination, ExaminationPart $part,
$item_id){
    $item = ExaminationPartItem::find($item_id);
    return view('grade-section-subjects.examinations.parts.items.delete', compact('examination', 'part',
'item'));
}

```

```

public function delete(Examination $examination, ExaminationPart $part, $item_id){
    $item = ExaminationPartItem::find($item_id);
    $item->delete();
    return redirect('/examinations/' . $examination->id . '/parts/' . $part->id);
}
}

```

app\Http\Controllers\Examinations\ExaminationPartsController.php

```
<?php
```

```

namespace App\Http\Controllers\Examinations;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests;
use App\Models\Questions\Question;
use App\Models\Examinations\Examination;
use App\Models\Examinations\ExaminationPart;
use DB;

```

```

class ExaminationPartsController extends Controller
{
    public function index(Examination $examination){
        return view('grade-section-subjects.examinations.parts.content.index', compact('examination'));
    }

    public function show(Examination $examination, ExaminationPart $part){
        if (!strcmp($part->getQuestionType(), 'Select from the Wordbox') ||
            !strcmp($part->getQuestionType(), 'Match Column A with Column B')){
            if ($part->items->first() == null){
                $question = null;
            } else {
                $question = Question::find($part->items()->first()->question_id);
            }
            return view('grade-section-subjects.examinations.parts.content.show-one', compact('subject',
'examination', 'part', 'question'));
        } else {
            return view('grade-section-subjects.examinations.parts.content.show', compact('subject',
'examination', 'part'));
        }
    }

    public function create(Examination $examination){
        $types = DB::table('question_types')->lists('name', 'id');
        return view('grade-section-subjects.examinations.parts.content.create', compact('subject',
'examination', 'types'));
    }

    public function store(Examination $examination, Request $request){
        $part = $examination->parts()->create($request->all());
        $examination->total_points = $examination->computeTotalPoints();
        $examination->update();
        return redirect('/examinations/' . $examination->id . '/parts/' . $part->id);
    }

    public function edit(Examination $examination, ExaminationPart $part){
        $types = DB::table('question_types')->lists('name', 'id');
        return view('grade-section-subjects.examinations.parts.content.edit', compact('subject',
'examination', 'part', 'types'));
    }

    public function update(Examination $examination, ExaminationPart $part, Request $request){
        if ($part->question_type_id != $request->input('question_type_id')){
            foreach ($part->items as $item){

```

```

        $item->delete();
    }
}
$part->update($request->all());
$examination->total_points = $examination->computeTotalPoints();
$examination->update();
return redirect('/examinations/' . $examination->id . '/parts');
}

public function showDeleteConfirmation(Examination $examination, ExaminationPart $part) {
    return view('grade-section-subjects.examinations.parts.content.delete', compact('subject',
'examination', 'part'));
}

public function delete(Examination $examination, ExaminationPart $part, Request $request) {
    $examination->total_points = $examination->computeTotalPoints();
    $examination->update();
    $part->delete();
    return redirect('/examinations/' . $examination->id . '/parts');
}
}
}

```

app\Http\Controllers\Examinations\ExaminationsController.php

<?php

```

namespace App\Http\Controllers\Examinations;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests;
use App\Http\Services\QuestionsService;
use App\Http\Services\ExaminationsService;
use App\Models\Questions\Question;
use App\Models\Examinations\Examination;
use App\Models\Examinations\Deployment;

use App\Models\GradeSectionSubjects\GradeSectionSubject;
use App\Models\Examinations\DeploymentInstance;
// use App\Models\Subjects\SubjectExaminationInstance;
// use App\Models\Subjects\SubjectExaminationAnswer;
use Carbon\Carbon;
use Auth;

class ExaminationsController extends Controller

```



```

{
  private $questionsService;
  private $examinationsService;

  public function __construct(QuestionsService $questionsService, ExaminationsService
$examinationsService){
    $this->questionsService = $questionsService;
    $this->examinationsService = $examinationsService;
  }

  public function index(){
    $examinations = Examination::where('user_id', Auth::user()->id)->orderBy('created_at', 'desc')-
>get();
    return view('grade-section-subjects.examinations.content.index', compact('examinations'));
  }

  public function show(Examination $examination){
    $attachments = Deployment::where('examination_id', $examination->id)->get();
    return view('grade-section-subjects.examinations.content.show', compact('examination',
'attachments'));
  }

  public function create(){
    $subcategories = $this->fetchExamSubcategories();
    return view('grade-section-subjects.examinations.content.create', compact('subcategories'));
  }

  public function store(Request $request){
    $examRequest = $request->all();
    $examRequest['total_points'] = 0;
    $examRequest['user_id'] = Auth::user()->id;
    $exam = Examination::create($examRequest);
    return redirect('/examinations/' . $exam->id);
  }

  public function edit(Examination $examination){
    $subcategories = $this->fetchExamSubcategories();
    return view('grade-section-subjects.examinations.content.edit', compact('subject', 'examination',
'subcategories'));
  }

  public function update(Examination $examination, Request $request){
    $examination->update($request->all());
    return redirect('/examinations');
  }

```

```

}

public function assignment(Examination $examination)

public function showExamResults(GradeSectionSubject $gradeSectionSubject, Deployment
$deployment){
    $instance = DeploymentInstance::where('user_id', Auth::user()->id)->where('deployment_id',
$deployment->id)->first();
    if (!strcmp(Auth::user()->getRole(), 'Student') && $instance != null && $instance->is_finished){
        return view('classes.examinations.results', compact('gradeSectionSubject', 'deployment',
$instance'));
    } else {
        return redirect('/classes/' . $gradeSectionSubject->id . '/deployments/' . $deployment->id .
'/instances/');
    }
}

public function showResultsIndex(Subject $subject, SubjectExamination $examination){
    $exams = [];
    foreach ($subject->section->students as $student){
        $instance = SubjectExaminationInstance::where('examination_id', $examination->id)-
>where('user_id', $student->id)->first();
        $exams[$student->id] = $instance;
    }
    return view('grade-section-subjects.examinations.students.index', compact('subject', 'examination',
'exams'));
}

public function showStudentResults(Subject $subject, SubjectExamination $examination,
SubjectExaminationInstance $instance){
    return $this->examinationsService->showResults($subject, $examination, $instance);
}

private function fetchExamSubcategories(){
    $subcategories['Seatwork'] = 'Seatwork';
    $subcategories['Homework'] = 'Homework';
    $subcategories['Quiz'] = 'Quiz';
    $subcategories['Long test'] = 'Long test';
    $subcategories['Others'] = 'Others';
    return $subcategories;
}
}

```

App\Http\Controllers\Examinations\FacultyExaminationInstancesController.php

```
<?php
```

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

```
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Services\QuestionsService;
use App\Http\Services\ExaminationsService;
use App\Http\Requests;
use App\Models\Questions\Question;
use App\Models\Questions\Types\QuestionMultipleChoice;
use App\Models\Questions\Types\QuestionTrueOrFalse;
use App\Models\Questions\Types\QuestionFillInTheBlanks;
use App\Models\Questions\Types\QuestionSelectFromTheWordboxChoice;
use App\Models\Questions\Types\QuestionSelectFromTheWordboxItem;
use App\Models\Questions\Types\QuestionMatchColumnsChoice;
use App\Models\Questions\Types\QuestionMatchColumnsItem;
use App\Models\Examinations\Examination;
use App\Models\Examinations\DeploymentInstance;
use App\Models\Examinations\DeploymentInstancePart;
use Auth;
```

```
class FacultyExaminationInstancesController extends Controller
```

```
{
    private $questionsService;
    private $examinationsService;

    public function __construct(QuestionsService $questionsService, ExaminationsService
$examinationsService){
        $this->questionsService = $questionsService;
        $this->examinationsService = $examinationsService;
    }

    public function createExaminationInstance(Examination $examination, Request $request){
        if (!$this->examinationsService->verifyQuestionQuantity($examination)){
            return redirect('/examinations/' . $examination->id . '/parts');
        } else{
            $deploymentInstance = $this->createDeploymentInstance($examination);
            foreach ($examination->parts as $examPart){
                $questionTypeId = $examPart->question_type_id;
                $first = true;
                $questionsOrder = "";
                if (!strcmp($examPart->getQuestionType(), 'Select from the Wordbox')){
                    $questionItem = $examPart->items()->first();
                }
            }
        }
    }
}
```

```

    $question = Question::find($questionItem->question_id);
    $swordboxChoiceIds = QuestionSelectFromTheWordboxChoice::get()->pluck('id');
    $swordboxItems = QuestionSelectFromTheWordboxItem::whereIn('wordbox_choice_id',
$ SwordboxChoiceIds)->get()->shuffle();
    for ($i = 0; $i < $examPart->items->first()->quantity; $i++){
        if (!$first){
            $questionsOrder = $questionsOrder . '|' . $swordboxItems[$i]->id;
        } else {
            $questionsOrder = " . $swordboxItems[$i]->id;
            $first = false;
        }
    }
} else if (!strcmp($examPart->getQuestionType(), 'Match Column A with Column B')){
    $questionItem = $examPart->items()->first();
    $question = Question::find($questionItem->question_id);
    $columnsChoiceIds = QuestionMatchColumnsChoice::get()->pluck('id');
    $columnsItems = QuestionMatchColumnsItem::whereIn('columns_choice_id',
$ ColumnsChoiceIds)->get()->shuffle();
    for ($i = 0; $i < ($questionItem->quantity + $questionItem->choices_quantity); $i++){
        if (!$first){
            $questionsOrder = $questionsOrder . '|' . $columnsItems[$i]->id;
        } else {
            $questionsOrder = " . $columnsItems[$i]->id;
            $first = false;
        }
    }
} else {
    foreach ($examPart->items as $item){
        $questions = Question::where('question_type_id', $questionTypeId)-
>where('question_subtopic_id', $item->question_subtopic_id)->get()->shuffle();
        for ($i = 0; $i < $item->quantity; $i++){
            if (!$first){
                $questionsOrder = $questionsOrder . '|' . $questions[$i]->id;
            } else {
                $questionsOrder = " . $questions[$i]->id;
                $first = false;
            }
        }
    }
}
$deploymentInstance->parts()->create([
    'examination_part_id' => $examPart->id,
    'question_order' => $questionsOrder,
]);

```

```

    }
    }
    return redirect('/examinations/' . $examination->id . '/faculty/instances/' . $deploymentInstance->id .
'/page/1');
}

public function showExamPage(Examination $examination, DeploymentInstance
$deploymentInstance, $page){
    if ($page > count($deploymentInstance->parts)){
        return redirect('/examinations/' . $examination->id . '/faculty/instances/' . $deploymentInstance->id
. '/page/finish');
    } else {
        $deploymentInstancePart = $deploymentInstance->parts[$page-1];
        $questionIds = explode('|', $deploymentInstancePart->question_order);
        $choices_quantity = 0;
        foreach ($questionIds as $id){
            if (!strcmp($deploymentInstancePart->examinationPart->getQuestionType(), 'Select from
the Wordbox')){
                $questions[$id] = QuestionSelectFromTheWordboxItem::find($id);
            } else if (!strcmp($deploymentInstancePart->examinationPart->getQuestionType(),
'Match Column A with Column B')){
                $questions[$id] = QuestionMatchColumnsItem::find($id);
                $choices_quantity = $examination->parts[$page-1]->items()->first()-
>choices_quantity;
            } else {
                $questions[$id] = Question::find($id);
            }
        }
        $navbar = $this->createNavbar($examination, $deploymentInstance, $page);
        $submitUrl = '/examinations/' . $examination->id . '/faculty/instances/' . $deploymentInstance->id
. '/page/' . $page;
        return $this->examinationsService->generatePage($deploymentInstancePart, $questions,
$navbar, $submitUrl, $choices_quantity);
    }
}

public function processExamPage(Examination $examination, DeploymentInstance
$deploymentInstance, $page, Request $request){
    $deploymentInstancePart = $deploymentInstance->parts[$page-1];
    $questionIds = explode('|', $deploymentInstancePart->question_order);
    foreach ($questionIds as $id){
        $questions[$id] = Question::find($id);
    }
}

```

```

        $nextUrl = '/examinations/' . $examination->id . '/faculty/instances/' . $deploymentInstance->id
        . '/page/' . ($page + 1);
        return $this->examinationsService->processPage($deploymentInstancePart, $questions, $nextUrl,
        $page, $request);
    }

    public function showExamFinishPage(Examination $examination, DeploymentInstance
    $deploymentInstance){
        $navbar = $this->createNavbar($examination, $deploymentInstance, 'finish');
        $nextUrl = '/examinations/' . $examination->id . '/faculty/instances/' . $deploymentInstance->id .
        '/page/finish';
        $startUrl = '/examinations/' . $examination->id . '/faculty/instances/' . $deploymentInstance->id .
        '/page/1';
        return view('grade-section-subjects.examinations.instances.finish', compact('navbar', 'nextUrl',
        'startUrl'));
    }

    public function finish(Examination $examination, DeploymentInstance $deploymentInstance){
        $score = 0;
        foreach ($deploymentInstance->parts as $instancePart){
            $questionType = $instancePart->examinationPart->getQuestionType();
            $questionIds = explode('|', $instancePart->question_order);
            if (!strcmp($questionType, 'Match Column A with Column B')) {
                $sch_quantity = $instancePart->examinationPart->items()->first()->choices_quantity;
                $questionIds = array_slice($questionIds, 0, count($questionIds) - $sch_quantity);
            }
            $pointsPerItem = $instancePart->examinationPart->points;
            foreach ($questionIds as $questionId){
                switch ($questionType){
                    case 'Multiple Choice':
                        $question = Question::find($questionId);
                        $answer = $this->examinationsService->fetchAnswer($instancePart->id, $question->id)-
                        >answer;
                        $correctAnswer = QuestionMultipleChoice::where('question_id', $question->id)-
                        >where('is_right_answer', '1')->first()->text;
                        $score += (!strcmp($answer, $correctAnswer)) ? $pointsPerItem : 0;
                        break;
                    case 'True or False':
                        $question = Question::find($questionId);
                        $answer = $this->examinationsService->fetchAnswer($instancePart->id, $question->id)-
                        >answer;
                        $correctAnswer = QuestionTrueOrFalse::where('question_id', $question->id)->first()-
                        >right_answer;
                        $score += (!strcmp($answer, $correctAnswer)) ? $pointsPerItem : 0;

```

```

        break;
    case 'Fill in the Blanks':
        $question = Question::find($questionId);
        $answer = $this->examinationsService->fetchAnswer($instancePart->id, $question->id)-
>answer;
        $correctAnswer = QuestionFillInTheBlanks::where('question_id', $question->id)->first()-
>right_answer;
        $score += (!strcmp(strtolower($answer), strtolower($correctAnswer))) ? $pointsPerItem :
0;
        break;
    case 'Select from the Wordbox':
        $wordboxItemId = $questionId;
        $answer = ExaminationAnswer::where('examination_instance_part_id', $instancePart-
>id)->where('wordbox_item_id', $wordboxItemId)->first()->answer;
        $item = QuestionSelectFromTheWordboxItem::find($wordboxItemId);
        $correctAnswer = $item->choice->text;
        $score += (!strcmp($answer, $correctAnswer)) ? $pointsPerItem : 0;
        break;
    case 'Match Column A with Column B':
        $columnsItemId = $questionId;
        $answer = ExaminationAnswer::where('examination_instance_part_id', $instancePart-
>id)->where('columns_item_id', $columnsItemId)->first()->answer;
        $item = QuestionMatchColumnsItem::find($columnsItemId);
        $correctAnswer = $item->choice->text;
        $score += (!strcmp($answer, $correctAnswer)) ? $pointsPerItem : 0;
        break;
    default:
        flash()->error('Some error occurred. Please try again.');
```

```

        return redirect('/home');
    }
}
}
$deploymentInstance->score = $score;
$deploymentInstance->is_finished = true;
$deploymentInstance->update();
return redirect('/examinations/' . $examination->id . '/faculty/instances/' . $deploymentInstance->id .
'/results');
```

```

}

public function showExamResults(Examination $examination, DeploymentInstance
$deploymentInstance){
    return view('grade-section-subjects.examinations.instances.results', compact('examination',
'deploymentInstance'));
}

```

```

private function createDeploymentInstance($examination){
    $deploymentInstance = new DeploymentInstance();
    $deploymentInstance->user_id = Auth::user()->id;
    $deploymentInstance->deployment_id = null;
    $deploymentInstance->is_finished = false;
    $deploymentInstance->score = null;
    $deploymentInstance->time_started = null;
    $deploymentInstance->time_ended = null;
    $deploymentInstance->save();
    return $deploymentInstance;
}

public function createNavbar($examination, $deploymentInstance, $page){
    $navbar = "<nav><ul class='pager'>";
    if ($page == 1){
        $navbar = $navbar . "<li class='previous disabled'><a href='#'><span aria-
hidden='true'>&larr;</span> Previous page</a></li>";
        } else if ($page == 'finish'){
        $navbar = $navbar . "<li class='previous'><a href='/examinations/" . $examination->id .
"/instances/" . $deploymentInstance->id . "/page/" . count($deploymentInstance->parts) . "'><span aria-
hidden='true'>&larr;</span> Previous page</a></li>";
        } else {
        $navbar = $navbar . "<li class='previous'><a href='/examinations/" .
$examination->id . "/instances/" . $deploymentInstance->id . "/page/" . ($page - 1) . "'><span aria-
hidden='true'>&larr;</span> Previous page</a></li>";
        }
        if ($page == 'finish'){
        $navbar = $navbar . "<li class='next disabled'><a href='#'>Next page <span aria-
hidden='true'>&rarr;</span></a></li>";
        } else {
        $navbar = $navbar . "<li class='next'><a href='/examinations/" . $examination-
>id . "/instances/" . $deploymentInstance->id . "/page/" . ($page + 1) . "'>
        Next page <span aria-
hidden='true'>&rarr;</span></a></li>";
        }
        $navbar = $navbar . "</ul></nav>";
    return $navbar;
}
}

```

app\Http\Controllers\GradeSectionSubjectContents\PostsController.php

<?php


```

namespace App\Http\Controllers\GradeSectionSubjectContents;

use Illuminate\Http\Request;
use App\Http\Requests;
use App\Http\Controllers\Controller;
use App\Http\Requests\GradeSectionSubjects\PostRequest;
use App\Http\Services\FilesService;
use App\Models\File;
use App\Models\PostsAndRequirements\Post;
use App\Models\GradeSectionSubjects\GradeSectionSubject;
use App\Models\GradeSectionSubjects\GradeSectionSubjectPost;
use Auth;
use Carbon\Carbon;

class PostsController extends Controller
{
    private $filesService;

    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct(FilesService $filesService){
        $this->filesService = $filesService;
    }

    /**
     * Show list of all posts
     *
     * @return \Illuminate\Http\Response
     */
    public function index(){
        $posts = Post::where('user_id', Auth::user()->id)->orderBy('created_at', 'desc')->get();
        return view('grade-section-subjects.posts.index', compact('posts'));
    }

    /**
     * Show the contents of the given post
     *
     * @param Post $post
     * @return \Illuminate\Http\Response
     */

```

```

public function show(Post $post){
    $is_teacher = true;
    $attachments = GradeSectionSubjectPost::where('post_id', $post->id)->get();
    return view('grade-section-subjects.posts.show', compact('post', 'attachments', 'is_teacher'));
}

/**
 * Show the form in creating a subject post
 *
 * @param Subject $subject
 * @return \Illuminate\Http\Response
 */
public function create(){
    $timeNow = Carbon::parse(Carbon::now()->format('Y-m-d\\TH:i'));
    $gradeSectionSubjectsList = Auth::user()->gradeSectionSubjects;
    $gradeSectionSubjects = [];
    $gradeSectionSubjects[0] = "--Disable--";
    foreach ($gradeSectionSubjectsList as $gradeSectionSubject){
        $gradeSectionSubjects[$gradeSectionSubject->id] = $gradeSectionSubject->subject->name . ' (' .
$gradeSectionSubject->gradeSection->getName->name . ')';
    }
    return view('grade-section-subjects.posts.create', compact('gradeSectionSubjects', 'timeNow'));
}

/**
 * Store the subject post in the database
 *
 * @param Subject $subject
 * @param PostRequest $request
 * @return \Illuminate\Http\Response
 */
public function store(PostRequest $request){
    $data = $request->all();
    $data['user_id'] = Auth::user()->id;
    $post = Post::create($data);

    $destinationPath = 'documents/' . Auth::user()->username . '/posts/' . $post->id;
    $files = $request->file('files');
    $file_ids = $this->uploadFiles($files, $destinationPath);
    $post->files()->sync($file_ids);

    $gradeSectionSubjectIdList = $request->input('grade_section_subject_id');
    $publishOnList = $request->input('publish_on');
    for ($i = 0; $i < count($gradeSectionSubjectIdList); $i++){

```

```

    if ($gradeSectionSubjectIdList[$i] != '0'){
        GradeSectionSubjectPost::create([
            'post_id' => $post->id,
            'grade_section_subject_id' => $gradeSectionSubjectIdList[$i],
            'publish_on' => Carbon::parse($publishOnList[$i]),
        ]);
    }
}
flash()->success('Post has been created and attached. It will be published on the specified date.');
```

```

return redirect('/posts/' . $post->id);
}

/**
 * Show the form in editing a subject post
 *
 * @param Subject $subject
 * @param Post $post
 * @return \Illuminate\Http\Response
 */
public function edit(Post $post){
    return view('grade-section-subjects.posts.edit', compact('post'));
}

/**
 * Update the subject post in the database
 *
 * @param Subject $subject
 * @param Post $post
 * @param PostRequest $request
 * @return \Illuminate\Http\Response
 */
public function update(Post $post, PostRequest $request){
    $files = $request->file('files');
    $destinationPath = 'documents/' . Auth::user()->username . '/posts/' . $post->id;
    $new_file_ids = $this->uploadFiles($files, $destinationPath);
    $old_file_ids = ($request->input('old_files') == null ? [] : $request->input('old_files'));
    $file_ids = array_merge($old_file_ids, $new_file_ids);

    $post->update($request->all());
    $post->files()->sync($file_ids);
    flash()->success('Post successfully updated');
    return redirect('/posts/' . $post->id);
}

```

```

/**
 * Store into the database the files uploaded by the faculty
 *
 * @param array $files
 * @param string $destinationPath
 * @return array
 */
private function uploadFiles($files, $destinationPath){
    if($files[0]){
        return $this->filesService->upload($files, $destinationPath);
    } else {
        return [];
    }
}
}

```

app\Http\Controllers\GradeSectionSubjectContents\RequirementsController.php

<?php

```

namespace App\Http\Controllers\GradeSectionSubjectContents;

use Illuminate\Http\Request;
use App\Http\Requests;
use App\Http\Controllers\Controller;
use App\Http\Requests\GradeSectionSubjects\RequirementRequest;
use App\Http\Services\FilesService;
use App\Models\File;
use App\Models\PostsAndRequirements\Requirement;
use App\Models\GradeSectionSubjects\GradeSectionSubject;
use App\Models\GradeSectionSubjects\GradeSectionSubjectRequirement;
use App\Models\GradeSectionSubjects\GradeSectionSubjectRequirementInstance;
use Auth;
use Carbon\Carbon;

class RequirementsController extends Controller
{
    private $filesService;

    /**
     * Create a new controller instance.
     *
     * @return void
     */
}

```

```

*/
public function __construct(FilesService $filesService){
    $this->filesService = $filesService;
}

/**
 * Show list of all subject requirements
 *
 * @return \Illuminate\Http\Response
 */
public function index(){
    $requirements = Requirement::where('user_id', Auth::user()->id)->orderBy('created_at', 'desc')-
>get();
    return view('grade-section-subjects.requirements.index', compact('requirements'));
}

/**
 * Show the contents of the given subject requirement
 *
 * @param Requirement $requirement
 * @return \Illuminate\Http\Response
 */
public function show(Requirement $requirement){
    $is_teacher = true;
    $attachments = GradeSectionSubjectRequirement::where('requirement_id', $requirement->id)-
>get();
    return view('grade-section-subjects.requirements.show', compact('requirement', 'attachments',
'is_teacher'));
}

/**
 * Show the form in creating a subject requirement
 *
 * @return \Illuminate\Http\Response
 */
public function create(){
    $timeNow = Carbon::parse(Carbon::now()->format('Y-m-d\\TH:i'));
    $timeEnd = Carbon::parse(Carbon::now()->addDays(1)->format('Y-m-d\\TH:i'));
    $gradeSectionSubjectsList = Auth::user()->gradeSectionSubjects;
    $gradeSectionSubjects = [];
    $gradeSectionSubjects[0] = "--Disable--";
    foreach ($gradeSectionSubjectsList as $gradeSectionSubject){
        $gradeSectionSubjects[$gradeSectionSubject->id] = $gradeSectionSubject->subject->name . ' (' .
$gradeSectionSubject->gradeSection->getName->name . ')';
    }
}

```

```

    }
    return view('grade-section-subjects.requirements.create', compact('gradeSectionSubjects', 'timeNow',
'timeEnd'));
}

/**
 * Store the subject requirement in the database
 *
 * @param RequirementRequest $request
 * @return \Illuminate\Http\Response
 */
public function store(RequirementRequest $request){
    $data = $request->all();
    $data['user_id'] = Auth::user()->id;
    $requirement = Requirement::create($data);

    $destinationPath = 'documents/' . Auth::user()->username . '/requirements/' . $requirement->id;
    $files = $request->file('files');
    $file_ids = $this->uploadFiles($files, $destinationPath);
    $requirement->files()->sync($file_ids);

    $gradeSectionSubjectIdList = $request->input('grade_section_subject_id');
    $publishOnList = $request->input('publish_on');
    $eventStartList = $request->input('event_start');
    $eventEndList = $request->input('event_end');
    for ($i = 0; $i < count($gradeSectionSubjectIdList); $i++){
        if ($gradeSectionSubjectIdList[$i] != '0'){
            GradeSectionSubjectRequirement::create([
                'requirement_id' => $requirement->id,
                'grade_section_subject_id' => $gradeSectionSubjectIdList[$i],
                'publish_on' => Carbon::parse($publishOnList[$i]),
                'event_start' => Carbon::parse($eventStartList[$i]),
                'event_end' => Carbon::parse($eventEndList[$i]),
            ]);
        }
    }
    flash()->success('Requirement has been created and attached. It will be published on the specified
date. ');
    return redirect('/requirements/' . $requirement->id);
}

/**
 * Show the form in editing a subject requirement
 *

```

```

* @param Requirement $requirement
* @return \Illuminate\Http\Response
*/
public function edit(Requirement $requirement){
    return view('grade-section-subjects.requirements.edit', compact('requirement'));
}

/**
 * Update the subject requirement in the database
 *
 * @param Requirement $requirement
 * @param RequirementRequest $request
 * @return \Illuminate\Http\Response
 */
public function update(Requirement $requirement, RequirementRequest $request){
    $files = $request->file('files');
    $destinationPath = 'documents/' . Auth::user()->username . '/requirements' . $requirement->id;
    $new_file_ids = $this->uploadFiles($files, $destinationPath);
    $old_file_ids = ($request->input('old_files') == null ? [] : $request->input('old_files'));
    $file_ids = array_merge($old_file_ids, $new_file_ids);

    $requirement->update($request->all());
    $requirement->files()->sync($file_ids);
    flash()->success('Requirement successfully updated');
    return redirect('/requirements');
}

public function showStudentSubmissions(GradeSectionSubject $gradeSectionSubject,
GradeSectionSubjectRequirement $requirement){
    $submissions = [];
    foreach ($gradeSectionSubject->gradeSection->students as $student){
        $submissions[$student->id] = GradeSectionSubjectRequirementInstance::where('user_id',
        $student->id)->where('subject_requirement_id', $requirement->id)->first();
    }
    return view('grade-section-subjects.requirements.student-submissions',
compact('gradeSectionSubject', 'requirement', 'submissions'));
}

/**
 * Store into the database the files uploaded by the faculty
 *
 * @param array $files
 * @param string $destinationPath
 * @return array

```

```

    */
private function uploadFiles($files, $destinationPath){
    if($files[0]){
        return $this->filesService->upload($files, $destinationPath);
    } else {
        return [];
    }
}
}
}

```

app\Http\Controllers\Questions\QuestionCategoriesController.php

<?php

```

namespace App\Http\Controllers\Questions;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests;
use App\Http\Requests\Questions\QuestionCategoryRequest;
use App\Models\Questions\QuestionCategory;
use URL;

class QuestionCategoriesController extends Controller
{
    /**
     * Show list of all categories
     *
     * @param string $url
     * @return \Illuminate\Http\Response
     */
    public function index(){
        $categories = QuestionCategory::orderBy('name', 'asc')->get();
        return view('questions.categories.index', compact('categories', 'url'));
    }

    /**
     * Show the contents of a given category
     *
     * @param QuestionCategory $category
     * @return \Illuminate\Http\Response
     */
    public function show(QuestionCategory $category){
        $topics = $category->questionTopics()->get();
        return view('questions.categories.show', compact('category', 'topics'));
    }
}

```



```

    }

/**
 * Show the form in creating a category
 *
 * @return \Illuminate\Http\Response
 */
public function create(){
    return view('questions.categories.create');
}

/**
 * Store the category in the database
 *
 * @param QuestionCategoryRequest $request
 * @return \Illuminate\Http\Response
 */
public function store(QuestionCategoryRequest $request){
    $category = QuestionCategory::create($request->all());
    $topic = $category->questionTopics()->create(['name' => 'Default Topic']);
    $topic->questionSubtopics()->create(['name' => 'Default Subtopic']);
    flash()->success('New category has been added. ');
    return redirect('/categories');
}

/**
 * Show the form in editing a category
 *
 * @param QuestionCategory $category
 * @return \Illuminate\Http\Response
 */
public function edit(QuestionCategory $category){
    return view('questions.categories.edit', compact('category'));
}

/**
 * Update the category in the database
 *
 * @param QuestionCategory $category
 * @param QuestionCategoryRequest $request
 * @return \Illuminate\Http\Response
 */
public function update(QuestionCategory $category, QuestionCategoryRequest $request){
    $category->update($request->all());
}

```

```

        flash()->success('Category successfully updated');
        return redirect('/categories');
    }

    /**
     * Show the form in deleting a category
     *
     * @param QuestionCategory $category
     * @return \Illuminate\Http\Response
     */
    public function showDeleteConfirmation(QuestionCategory $category){
        return view('questions.categories.delete', compact('category'));
    }

    /**
     * Delete the category in the database
     *
     * @param QuestionCategory $category
     * @return \Illuminate\Http\Response
     */
    public function delete(QuestionCategory $category){
        foreach ($category->questionTopics as $topic){
            foreach ($topic->questionSubtopics as $subtopic){
                foreach ($subtopic->questions as $question){
                    $question->move('Default Category', 'Default Topic', 'Default Subtopic');
                }
            }
        }
        $category->delete();
        return redirect('/categories');
    }
}

```

app\Http\Controllers\Questions\QuestionMatchColumnsController.php

```
<?php
```

```

namespace App\Http\Controllers\Questions;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests;
use App\Models\Questions\QuestionCategory;
use App\Models\Questions\QuestionTopic;

```

```

use App\Models\Questions\QuestionSubtopic;
use App\Models\Questions\Question;
use App\Models\Questions\Types\QuestionMatchColumnsChoice;
use App\Models\Questions\Types\QuestionMatchColumnsItem;

class QuestionMatchColumnsController extends Controller
{
    public function createItem(QuestionCategory $category, $url_topic, $url_subtopic, Question
    $question){
        $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id', '=',
    $category->id)->first();
        $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
    $topic->id)->first();
        return view('questions.content.create-type.match-column-a-with-column-b-item',
    compact('category', 'topic', 'subtopic', 'question', 'choices'));
    }

    public function storeItem(QuestionCategory $category, $url_topic, $url_subtopic, Question $question,
    Request $request){
        $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id', '=',
    $category->id)->first();
        $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
    $topic->id)->first();

        $choice = $question->matchColumnsChoices()->create(['text' => $request->input('choice')]);
        $choice->item()->create(['text' => $request->input('item')]);
        return redirect('/categories/' . $category->name . '/topics/' . $topic->name . '/subtopics/' .
    $subtopic->name . '/questions/' . $question->id);
    }
}

```

app\Http\Controllers\Questions\QuestionMultipleChoiceController.php

```
<?php
```

```

namespace App\Http\Controllers\Questions;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests;
use App\Models\Questions\Question;
use App\Models\Questions\QuestionCategory;
use App\Models\Questions\QuestionTopic;
use App\Models\Questions\QuestionSubtopic;
use App\Models\Questions\Types\QuestionMultipleChoice;

```

use URL;

```
class QuestionMultipleChoiceController extends Controller
{
    public function createChoice(QuestionCategory $category, $url_topic, $url_subtopic, Question
    $question){
        $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id', '=',
    $category->id)->first();
        $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
    $topic->id)->first();
        return view('questions.content.create-type.multiple-choice-choice', compact('category', 'topic',
    'subtopic', 'question', 'choices'));
    }

    public function storeChoice(QuestionCategory $category, $url_topic, $url_subtopic, Question
    $question, Request $request){
        $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id', '=',
    $category->id)->first();
        $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
    $topic->id)->first();

        if ($request->input('text') == "<br>" || $request->input('text') == ""){
            flash()->error('Please fill up all necessary fields');
            return redirect('/categories/' . $category->name . '/topics/' . $topic->name . '/subtopics/' . $subtopic-
    >name . '/questions/create/' . $url_type)
                ->withInput($request->all());
        }
        $choice = $question->multipleChoice()->create(['text' => $request->input('text'), 'is_right_answer'
    => '0']);
        return redirect('/categories/' . $category->name . '/topics/' . $topic->name . '/subtopics/' .
    $subtopic->name . '/questions/' . $question->id);
    }

    public function removeChoice($id){
        $choice = QuestionMultipleChoice::find($id);
        $choice->delete();
        return redirect(URL::previous());
    }
}
```

app\Http\Controllers\Questions\QuestionsController.php

<?php

namespace App\Http\Controllers\Questions;

```

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests;
use App\Models\Questions\QuestionCategory;
use App\Models\Questions\QuestionTopic;
use App\Models\Questions\QuestionSubtopic;
use App\Models\Questions\Question;
use App\Http\Services\QuestionsService;
use DB;
use Auth;

class QuestionsController extends Controller
{
    private $questionsService;

    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct(QuestionsService $questionsService){
        $this->questionsService = $questionsService;
    }

    /**
     * Show the contents of the given question
     *
     * @param QuestionCategory $category
     * @param string $url_topic
     * @param string $url_subtopic
     * @param Question $question
     * @return \Illuminate\Http\Response
     */
    public function show(QuestionCategory $category, $url_topic, $url_subtopic, Question $question){
        $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id', '=',
$category->id)->first();
        $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
$topic->id)->first();

        $backUrl = "/categories/" . $category->name . "/topics/" . $topic->name . "/subtopics/" . $subtopic-
>name;
        $generateUrl = "/categories/" . $category->name . "/topics/" . $topic->name . "/subtopics/" .
$subtopic->name . "/questions/" . $question->id . "/instance";

```

```

        return $this->questionsService->showByType($question, $generateUrl, $backUrl);
    }

    /**
     * Show the form in creating questions
     *
     * @param QuestionCategory $category
     * @param string $url_topic
     * @param string $url_subtopic
     * @param string $url_type
     * @return \Illuminate\Http\Response
     */
    public function create(QuestionCategory $category, $url_topic, $url_subtopic, $url_type){
        $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id', '=',
$category->id)->first();
        $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
$topic->id)->first();

        switch($url_type){
            case 'multiple-choice':
                $type = 'Multiple Choice';
                //return view('questions.content.create-type.multiple-choice', compact('url_type', 'category',
'topic', 'subtopic', 'type'));
                return view('questions.content.create-type.multiple-choice-2', compact('url_type', 'category',
'topic', 'subtopic', 'type'));
                break;
            case 'true-or-false':
                $type = 'True or False';
                return view('questions.content.create-type.true-or-false', compact('url_type', 'category', 'topic',
'subtopic', 'type'));
                break;
            case 'fill-in-the-blanks':
                $type = 'Fill in the Blanks';
                return view('questions.content.create-type.fill-in-the-blanks', compact('url_type',
'category', 'topic', 'subtopic', 'type'));
                break;
            case 'match-column-a-with-column-b':
                $type = 'Match Column A with Column B';
                return view('questions.content.create', compact('url_type', 'category', 'topic', 'subtopic', 'type'));
                break;
            case 'select-from-the-wordbox':
                $type = 'Select from the Wordbox';
                return view('questions.content.create', compact('url_type', 'category', 'topic', 'subtopic', 'type'));
                break;
        }
    }

```

```

        default:
            return redirect('/home');
    }
}

/**
 * Store the question in the database
 *
 * @param QuestionCategory $category
 * @param string $url_topic
 * @param string $url_subtopic
 * @param string $url_type
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(QuestionCategory $category, $url_topic, $url_subtopic, $url_type, Request
$request){
    $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id', '=',
$category->id)->first();
    $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
$topic->id)->first();

    $this->validate($request, $this->fetchRules('common'));
    $type_id = DB::table('question_types')->where('name', $request->input('type'))->first()->id;
    $rules = $this->fetchRules($url_type);
    switch($url_type){
        case 'multiple-choice':
            // $this->validate($request, $rules);
            // $wrongAnswers = explode("|", $request->input('wrong_answers'));
            // if($wrongAnswers[0] == ""){
            //     return redirect('/categories/' . $category->name . '/topics/' . $topic->name . '/subtopics/' .
$subtopic->name . '/questions/create/' . $url_type)
            //         ->withInput($request->all())->withErrors(['wrong_answers' => 'Must have at least one
wrong choice.']);
            // } else {
            //     $question = $this->createQuestion($request, $type_id, $subtopic->id);

            //     $question->multipleChoice()->create(['text' => $request->input('right_answer'),
'is_right_answer' => '1']);
            //     foreach ($wrongAnswers as $wrongAnswer){
            //         $question->multipleChoice()->create(['text' => $wrongAnswer, 'is_right_answer' =>
'0']);
            //     }
            // }

```

```

        if ($request->input('right_answer') == "<br>" || $request->input('wrong_answer') == "<br>" ||
        $request->input('right_answer') == "" || $request->input('wrong_answer') == ""){
            flash()->error('Please fill up all necessary fields');
            return redirect('/categories/' . $category->name . '/topics/' . $topic->name . '/subtopics/' .
        $subtopic->name . '/questions/create/' . $url_type)
                ->withInput($request->all());
        }
        $question = $this->createQuestion($request, $type_id, $subtopic->id);
        $question->multipleChoice()->create(['text' => $request->input('right_answer'),
        'is_right_answer' => '1']);
        $question->multipleChoice()->create(['text' => $request->input('wrong_answer'),
        'is_right_answer' => '0']);
        flash()->success("Question successfully added");
        return redirect('/categories/' . $category->name . '/topics/' . $topic->name . '/subtopics/' .
        $subtopic->name . '/questions/' . $question->id);
        break;
    case 'true-or-false':
        $this->validate($request, $rules);
        $question = $this->createQuestion($request, $type_id, $subtopic->id);
        $question->trueOrFalse()->create($request->all());
        break;
    case 'fill-in-the-blanks':
        $this->validate($request, $rules);
        $question = $this->createQuestion($request, $type_id, $subtopic->id);
        $question->fillInTheBlanks()->create($request->all());
        break;
    case 'match-column-a-with-column-b':
    case 'select-from-the-wordbox':
        $question = $this->createQuestion($request, $type_id, $subtopic->id);
        flash()->success("Question successfully added");
        return redirect('/categories/' . $category->name . '/topics/' . $topic->name . '/subtopics/' .
        $subtopic->name . '/questions/' . $question->id);
        break;
    default:
        return redirect('/home');
    }
    flash()->success("Question successfully added");
    return redirect('/categories/' . $category->name . '/topics/' . $topic->name . '/subtopics/' . $subtopic-
    >name);
}

/**
 * Show the form in deleting a question
 */

```



```

* @param QuestionCategory $category
* @param string $url_topic
* @param string $url_subtopic
* @param Question $question
* @return \Illuminate\Http\Response
*/

public function showDeleteConfirmation(QuestionCategory $category, $url_topic, $url_subtopic,
Question $question){
    $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id', '=',
$category->id)->first();
    $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
$topic->id)->first();

    if ($question->user_id != Auth::user()->id){
        flash()->error("You cannot delete a question you haven't made!");
        return redirect('/categories/' . $category->name . '/topics/' . $topic->name . '/subtopics/' . $subtopic-
>name);
    } else {
        return view('questions.content.delete', compact('category', 'topic', 'subtopic', 'question'));
    }
}

/**
 * Delete the question in the database
 *
 * @param QuestionCategory $category
 * @param string $url_topic
 * @param string $url_subtopic
 * @param Question $question
 * @return \Illuminate\Http\Response
 */

public function delete(QuestionCategory $category, $url_topic, $url_subtopic, Question $question){
    $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id', '=',
$category->id)->first();
    $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
$topic->id)->first();
    $question->delete();
    flash()->success('Question successfully deleted');
    return redirect('/categories/' . $category->name . '/topics/' . $topic->name . '/subtopics/' . $subtopic-
>name);
}

/**
 * Show the form in creating questions

```

```

*
* @param string $type
* @return array
*/
private function fetchRules($type){
    switch($type){
        case 'common':
            return [
                'title' => 'required',
                'body' => 'required',
            ];
            break;
        case 'multiple-choice':
            break;
        case 'fill-in-the-blanks':
            return [
                'right_answer' => 'required|max:255',
            ];
            break;
        case 'true-or-false':
            return [
                'right_answer' => 'required|boolean',
            ];
            break;
        default:
            return [];
    }
}

/**
 * Creates a Question instance and stores it in the database
 *
 * @param \Illuminate\Http\Request $request
 * @param int $type_id
 * @param int $subtopic_id
 * @return Question $question
 */
private function createQuestion($request, $type_id, $subtopic_id){
    $question = new Question($request->all());
    $question->question_type_id = $type_id;
    $question->question_subtopic_id = $subtopic_id;
    $question->user_id = Auth::user()->id;
    $question->save();
}

```

```

        return $question;
    }
}

```

app\Http\Controllers\Questions\QuestionSelectFromTheWordboxController.php

```
<?php
```

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

```

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests;
use App\Models\Questions\QuestionCategory;
use App\Models\Questions\QuestionTopic;
use App\Models\Questions\QuestionSubtopic;
use App\Models\Questions\Question;
use App\Models\Questions\Types\QuestionSelectFromTheWordboxChoice;

```

```
class QuestionSelectFromTheWordboxController extends Controller
```

```

{
    public function createChoice(QuestionCategory $category, $url_topic, $url_subtopic, Question
    $question){
        $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id', '=',
    $category->id)->first();
        $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
    $topic->id)->first();
        return view('questions.content.create-type.select-from-the-wordbox-choice', compact('category',
    'topic', 'subtopic', 'question'));
    }

```

```

    public function storeChoice(QuestionCategory $category, $url_topic, $url_subtopic, Question
    $question, Request $request){
        $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id', '=',
    $category->id)->first();
        $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
    $topic->id)->first();

```

```

        $question->selectFromTheWordboxChoices()->create($request->all());
        return redirect('/categories/' . $category->name . '/topics/' . $topic->name . '/subtopics/' .
    $subtopic->name . '/questions/' . $question->id);
    }

```

```

    public function createItem(QuestionCategory $category, $url_topic, $url_subtopic, Question
    $question){

```

```

        $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id', '=',
$category->id)->first();
        $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
$topic->id)->first();

        $choices = QuestionSelectFromTheWordboxChoice::where('question_id', $question->id)-
>lists('text', 'id');
        return view('questions.content.create-type.select-from-the-wordbox-item', compact('category',
'topic', 'subtopic', 'question', 'choices'));
    }

    public function storeItem(QuestionCategory $category, $url_topic, $url_subtopic, Question $question,
Request $request) {
        $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id', '=',
$category->id)->first();
        $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
$topic->id)->first();

        $choice = $question->selectFromTheWordboxChoices()->find($request-
>input('wordbox_choice_id'));
        $choice->items()->create(['text' => $request->input('text')]);
        return redirect('/categories/' . $category->name . '/topics/' . $topic->name . '/subtopics/' .
$subtopic->name . '/questions/' . $question->id);
    }
}

```

app\Http\Controllers\Questions\QuestionSubtopicsController.php

```
<?php
```

```

namespace App\Http\Controllers\Questions;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests;
use App\Http\Requests\Questions\QuestionSubtopicRequest;
use App\Models\Questions\QuestionCategory;
use App\Models\Questions\QuestionTopic;
use App\Models\Questions\QuestionSubtopic;

class QuestionSubtopicsController extends Controller
{
    /**
     * Show the contents of a given subtopic
     *

```

```

* @param QuestionCategory $category
* @param string $url_topic
* @param string $url_subtopic
* @return \Illuminate\Http\Response
*/
public function show(QuestionCategory $category, $url_topic, $url_subtopic){
    $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id', '=',
$category->id)->first();
    $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
$topic->id)->first();
    return view('questions.subtopics.show', compact('category', 'topic', 'subtopic'));
    }

/**
* Show the form in creating a subtopic
*
* @param QuestionCategory $category
* @param string $url_topic
* @return \Illuminate\Http\Response
*/
public function create(QuestionCategory $category, $url_topic){
    $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id', '=',
$category->id)->first();
    return view('questions.subtopics.create', compact('category', 'topic'));
}

/**
* Store the subtopic in the database
*
* @param QuestionCategory $category
* @param string $url_topic
* @param QuestionSubtopicRequest $request
* @return \Illuminate\Http\Response
*/
public function store(QuestionCategory $category, $url_topic, QuestionSubtopicRequest
$request){
    $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id', '=',
$category->id)->first();
    $name = $request->input('name');
    if($this->checkIfAvailable($topic, $name)){
        $topic->questionSubtopics()->create($request->all());
        flash()->success('New subtopic has been added. ');
        return redirect('/categories/' . $category->name . '/topics/' . $topic->name);
    } else {

```

```

        return redirect('/categories/' . $category->name . '/topics/' . $topic->name .
'/subtopics/create')->withErrors(['name' => 'Subtopic name already exists for this topic.']);
    }
}

/**
 * Show the form in editing a subtopic
 *
 * @param QuestionCategory $category
 * @param string $url_topic
 * @param string $url_subtopic
 * @return \Illuminate\Http\Response
 */
public function edit(QuestionCategory $category, $url_topic, $url_subtopic){
    $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id',
'=', $category->id)->first();
    $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
$topic->id)->first();
    return view('questions.subtopics.edit', compact('category', 'topic', 'subtopic'));
}

/**
 * Update the subtopic in the database
 *
 * @param QuestionCategory $category
 * @param string $url_topic
 * @param string $url_subtopic
 * @return \Illuminate\Http\Response
 */
public function update(QuestionCategory $category, $url_topic, $url_subtopic,
QuestionSubtopicRequest $request){
    $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id',
'=', $category->id)->first();
    $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
$topic->id)->first();
    $name = $request->input('name');
    if($this->checkIfAvailable($topic, $name)){
        $subtopic->update($request->all());
        flash()->success('Subtopic successfully updated');
        return redirect('/categories/' . $category->name . '/topics/' . $topic->name);
    } else {
        return redirect('/categories/' . $category->name . '/topics/' . $topic->name .
'/subtopics/' . $subtopic->name . '/edit')->withErrors(['name' => 'Subtopic name already exists for this
topic.']);
    }
}

```

```

    }
}

/**
 * Show the form in deleting a subtopic
 *
 * @param QuestionCategory $category
 * @param string $url_topic
 * @param string $url_subtopic
 * @return \Illuminate\Http\Response
 */
public function showDeleteConfirmation(QuestionCategory $category, $url_topic,
$url_subtopic){
    $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id',
'=', $category->id)->first();
    $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
$topic->id)->first();

    return view('questions.subtopics.delete', compact('category', 'topic', 'subtopic'));
}

/**
 * Delete the subtopic in the database
 *
 * @param QuestionCategory $category
 * @param string $url_topic
 * @param string $url_subtopic
 * @return \Illuminate\Http\Response
 */
public function delete(QuestionCategory $category, $url_topic, $url_subtopic){
    $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id',
'=', $category->id)->first();
    $subtopic = QuestionSubtopic::where('name', '=', $url_subtopic)->where('question_topic_id', '=',
$topic->id)->first();

    foreach ($subtopic->questions as $question){
        $question->move($category->name, $topic->name, 'Default Subtopic');
    }

    $subtopic->delete();
    return redirect('/categories/' . $category->name . '/topics/' . $topic->name);
}

/**
 * Check if the submitted subtopic name is available for the given topic

```

```

*
* @param int $topic
* @param string $name
* @return boolean
*/
private function checkIfAvailable($topic, $name){
    $result = QuestionSubtopic::where('name', '=', $name)->where('question_topic_id', '=',
$topic->id)->first();
    if(empty($result)){
        return true;
    } else {
        return false;
    }
}
}

```

App\Http\Controllers\Questions\QuestionTopicsController.php

```
<?php
```

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

```

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests;
use App\Http\Requests\Questions\QuestionTopicRequest;
use App\Models\Questions\QuestionCategory;
use App\Models\Questions\QuestionTopic;
use Validator;

```

```
class QuestionTopicsController extends Controller
```

```

{
    /**
     * Show the contents of a given topic
     *
     * @param QuestionCategory $category
     * @param string $url_topic
     * @return \Illuminate\Http\Response
     */
    public function show(QuestionCategory $category, $url_topic){
        $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id', '=',
$category->id)->first();
        return view('questions.topics.show', compact('category', 'topic'));
    }
}

```



```

/**
 * Show the form in creating a topic
 *
 * @param QuestionCategory $category
 * @return \Illuminate\Http\Response
 */
    public function create(QuestionCategory $category){
        return view('questions.topics.create', compact('category'));
    }

    /**
 * Store the topic in the database
 *
 * @param QuestionCategory $category
 * @param QuestionTopicRequest $request
 * @return \Illuminate\Http\Response
 */
    public function store(QuestionCategory $category, QuestionTopicRequest $request){
        $name = $request->input('name');
        if($this->checkIfAvailable($category, $name)){
            $topic = $category->questionTopics()->create($request->all());
            $topic->questionSubtopics()->create(['name' => 'Default Subtopic']);
            flash()->success('New topic has been added.');
```

return redirect('/categories/' . \$category->name);

```
        } else {
            return redirect('/categories/' . $category->name . '/topics/create')->withErrors(['name' => 'Topic
name already exists for this category.']);
        }
    }

    /**
 * Show the form in editing a topic
 *
 * @param QuestionCategory $category
 * @param string $url_topic
 * @return \Illuminate\Http\Response
 */
    public function edit(QuestionCategory $category, $url_topic){
        $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id',
        '=', $category->id)->first();
        return view('questions.topics.edit', compact('category', 'topic'));
    }

    /**

```

```

* Update the topic in the database
*
* @param QuestionCategory $category
* @param string $url_topic
* @param QuestionTopicRequest $request
* @return \Illuminate\Http\Response
*/
    public function update(QuestionCategory $category, $url_topic, QuestionTopicRequest
$request){
        $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id',
'=', $category->id)->first();
        $name = $request->input('name');
        if($this->checkIfAvailable($category, $name)){
            $topic->update($request->all());
            flash()->success('Topic successfully updated');
            return redirect('/categories/' . $category->name);
        } else {
            return redirect('/categories/' . $category->name . '/topics/' . $topic->name . '/edit')-
>withErrors(['name' => 'Topic name already exists for this category.']);
        }
    }

/**
* Show the form in deleting a topic
*
* @param QuestionCategory $category
* @param string $url_topic
* @return \Illuminate\Http\Response
*/
    public function showDeleteConfirmation(QuestionCategory $category, $url_topic){
        $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id',
'=', $category->id)->first();
        return view('questions.topics.delete', compact('category', 'topic'));
    }

/**
* Delete the topic in the database
*
* @param QuestionCategory $category
* @param string $url_topic
* @return \Illuminate\Http\Response
*/
    public function delete(QuestionCategory $category, $url_topic){

```

```

        $topic = QuestionTopic::where('name', '=', $url_topic)->where('question_category_id',
        '=', $category->id)->first();

        foreach ($topic->questionSubtopics as $subtopic){
            foreach ($subtopic->questions as $question){
                $question->move($category->name, 'Default Topic', 'Default Subtopic');
            }
        }

        $topic->delete();
        return redirect('/categories/' . $category->name);
    }

    /**
     * Check if the submitted topic name is available for the given category
     *
     * @param int $category
     * @param string $name
     * @return boolean
     */
    private function checkIfAvailable($category, $name){
        $result = QuestionTopic::where('name', '=', $name)->where('question_category_id', '=',
        $category->id)->first();
        if(empty($result)){
            return true;
        } else {
            return false;
        }
    }
}

```

App\Http\Controllers\ClassesControllers.php

```
<?php
```

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

```
use Illuminate\Http\Request;
```

```
use App\Http\Requests;
```

```
use App\Http\Services\SubjectArticlesService;
```

```
use App\Http\Services\FilesService;
```

```
use App\Http\Services\QuestionsService;
```

```
use App\Models\GradeSectionSubjects\GradeSectionSubject;
```

```
use App\Models\GradeSectionSubjects\GradeSectionSubjectRequirement;
```

```
use App\Models\GradeSectionSubjects\GradeSectionSubjectRequirementInstance;
```

```

// use App\Models\Subjects\Subject;
// use App\Models\Subjects\SubjectExamination;
// use App\Models\Subjects\SubjectExaminationInstance;
use Carbon\Carbon;
use Auth;

class ClassesController extends Controller
{
    private $subjectArticlesService;
    private $filesService;
    private $questionsService;

    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct(SubjectArticlesService $subjectArticlesService, FilesService $filesService,
QuestionsService $questionsService){
        $this->subjectArticlesService = $subjectArticlesService;
        $this->filesService = $filesService;
        $this->questionsService = $questionsService;
    }

    /**
     * Show the contents of a given class
     *
     * @param GradeSectionSubject $gradeSectionSubject
     * @return \Illuminate\Http\Response
     */
    public function show(GradeSectionSubject $gradeSectionSubject){
        $articles = $this->subjectArticlesService->sortArticles($gradeSectionSubject);
        $types = session()->pull('types');
        $is_teacher = false;
        return view('classes.content.show', compact('gradeSectionSubject', 'articles', 'types', 'is_teacher'));
    }

    /**
     * Show the page with details of students submission to a given requirement
     *
     * @param GradeSectionSubject $gradeSectionSubject
     * @param GradeSectionSubjectRequirement $subjectRequirement
     * @return \Illuminate\Http\Response
     */
}

```

```

public function showRequirementStatus(GradeSectionSubject $gradeSectionSubject,
GradeSectionSubjectRequirement $subjectRequirement){
    $timeNow = strtotime(Carbon::now());
    $timeStart = strtotime($subjectRequirement->event_start);
    $timeEnd = strtotime($subjectRequirement->event_end);
    if ($timeStart <= $timeNow && $timeNow <= $timeEnd){
        $status = '<font color="green">Ongoing</font>';
        $ongoing = true;
    } else {
        $status = '<font color="red">Finished</font>';
        $ongoing = false;
    }
    $instance = $subjectRequirement->instances()->where('user_id', Auth::user()->id)->first();
    $file = (($instance) ? $instance->getFile() : null);
    $is_teacher = false;
    return view('classes.requirements.instance', compact('gradeSectionSubject', 'subjectRequirement',
'status', 'ongoing', 'instance', 'file', 'is_teacher'));
}

```

```
/**
```

```
* Verifies and uploads any file uploaded to the requirement
```

```
*
```

```
* @param GradeSectionSubject $gradeSectionSubject
```

```
* @param GradeSectionSubjectRequirement $requirement
```

```
* @param \Illuminate\Http\Request $request
```

```
* @return \Illuminate\Http\Response
```

```
*/
```

```

public function submitRequirement(GradeSectionSubject $gradeSectionSubject,
GradeSectionSubjectRequirement $requirement, Request $request){
    $this->validate($request, ['file' => 'required|max:16384'], ['file.required' => 'File required']);
    $destinationPath = 'documents/' . Auth::user()->username . '/' . $gradeSectionSubject->subject-
>name . '/' . $requirement->id;
    $file_ids = $this->filesService->upload([$request->file('file')], $destinationPath);
    $request['file_id'] = $file_ids[0];
    $request['user_id'] = Auth::user()->id;
    $request['submitted_at'] = Carbon::now();

    $instance = $requirement->instances()->where('user_id', Auth::user()->id)->first();
    if ($instance){
        $instance->delete();
    }
    $requirement->instances()->create($request->all());
    flash()->success('Your submission is successfully updated.');
```

```

        return redirect('/classes/' . $gradeSectionSubject->id . '/requirements/' . $requirement->id .
'/submission');
    }
}

```

\App\Http\Controllers\FilesController.php

```
<?php
```

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

```
use Illuminate\Http\Request;
```

```
use App\Http\Requests;
```

```
use App\Http\Requests\FileRequest;
```

```
use App\Http\Controllers\Controller;
```

```
use App\Models\File;
```

```
use App\Models\FileDownloadHistory;
```

```
use Auth;
```

```
use Response;
```

```
use Carbon\Carbon;
```

```
class FilesController extends Controller
```

```
{
```

```
/**
```

```
 * Create a new controller instance.
```

```
 *
```

```
 * @return void
```

```
 */
```

```
public function __construct(){
```

```
    $this->middleware('fileOwner', ['only' => ['viewDownloadHistory']]);
```

```
    $this->middleware('role:Faculty', ['only' => ['viewDownloadHistory']]);
```

```
}
```

```
/**
```

```
 * Request a file download
```

```
 *
```

```
 * @param \Illuminate\Http\Request $request
```

```
 * @return Illuminate\Http\Response
```

```
 */
```

```
public function download(Request $request){
```

```
    $fileId = $request->input('fileId');
```

```
    $file = File::find($fileId);
```

```
    $destinationPath = $file->destination_path;
```

```
    $fileName = $file->file_name;
```

```

    $filePath = $destinationPath . '/' . $fileName;

    $fileDownloadHistory = new FileDownloadHistory;
    $fileDownloadHistory->user_id = Auth::user()->id;
    $fileDownloadHistory->file_id = $file->id;
    $fileDownloadHistory->downloaded_at = Carbon::now();
    $fileDownloadHistory->save();
    return Response::download($filePath);
}

/**
 * Returns the list of students who downloaded the given file
 *
 * @param File $file
 * @return array $list
 */
public function viewDownloadHistory($file_id){
    $file = File::find($file_id);
    $list = FileDownloadHistory::where('file_id', $file->id)->orderBy('downloaded_at', 'desc')->get();
    return view('download-history', compact('file', 'list'));
}
}

```

App\Http\Controllers\GradeSectionSubjectsController.php

<?php

```

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Requests;
use App\Http\Services\SubjectArticlesService;
use App\Models\GradeSectionSubjects\GradeSectionSubject;
use App\Models\GradeSectionSubjects\SubjectClassRecord;
use App\Models\GradeSectionSubjects\SubjectClassRecordInstance;

class GradeSectionSubjectsController extends Controller
{
    private $subjectArticlesService;

    /**
     * Create a new controller instance.
     *
     * @return void
     */
}

```

```

*/
public function __construct(SubjectArticlesService $subjectArticlesService){
    $this->subjectArticlesService = $subjectArticlesService;
}

public function show(GradeSectionSubject $gradeSectionSubject){
    $articles = $this->subjectArticlesService->sortArticles($gradeSectionSubject);
    $types = session()->pull('types');
    $is_teacher = true;
    return view('grade-section-subjects.content.show', compact('gradeSectionSubject', 'articles',
'types', 'is_teacher'));
}

/**
 * Show the details of the given class
 *
 * @param GradeSectionSubject $gradeSectionSubject
 * @return \Illuminate\Http\Response
 */
public function showDetails(GradeSectionSubject $gradeSectionSubject){
    return view('grade-section-subjects.content.details', compact('gradeSectionSubject'));
}

public function showClassRecord(GradeSectionSubject $gradeSectionSubject){
    $quizClassRecords = [];
    $longTestClassRecords = [];
    $othersClassRecords = [];
    $classRecords = SubjectClassRecord::where('grade_section_subject_id', $gradeSectionSubject->id)-
>get();
    foreach ($classRecords as $classRecord){
        $subcategory = $classRecord->examination->subcategory;
        if (!strcmp($subcategory, 'Quiz')){
            $quizClassRecords[] = $classRecord;
        } else if (!strcmp($subcategory, 'Long test')){
            $longTestClassRecords[] = $classRecord;
        } else if (!strcmp($subcategory, 'Others')){
            $othersClassRecords[] = $classRecord;
        }
    }
}

$quizCount = count($quizClassRecords);
$longTestCount = count($longTestClassRecords);
$othersCount = count($othersClassRecords);
$students = $gradeSectionSubject->gradeSection->students;

```



```

        return view('grade-section-subjects.content.class-record', compact('gradeSectionSubject',
'quizClassRecords', 'longTestClassRecords', 'othersClassRecords',
        'classRecords', 'quizCount', 'longTestCount', 'othersCount', 'students'));
    }
}

```

App\Http\Controllers\HomeController.php

```
<?php
```

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

```

use App\Http\Requests;
use Illuminate\Http\Request;
use App\Models\Article;
use App\Models\GradeSectionSubjects\GradeSectionSubject;
use Auth;
use DB;

```

```
class HomeController extends Controller
```

```

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

    /**
     * Show the application dashboard based on role of user
     *
     * @param \Illuminate\Http\Request $request
     */
    public function index(Request $request){
        $role = $request->user()->getRole();
        switch($role){
            case 'System Administrator':
                return $this->redirectToSystemAdminHome();
                break;
            case 'School Management':
                return $this->redirectToSchoolManagementHome();
            case 'Faculty':

```

```

        return $this->redirectToFacultyHome();
        break;
    case 'Student':
        return $this->redirectToStudentHome();
        break;
    default:
        return redirect('/');
    }
}

/**
 * Redirects to system administrator home page with necessary data
 *
 * @return \Illuminate\Http\Response
 */
protected function redirectToSystemAdminHome(){
    $activeSchoolYear = DB::table('school_years')->where('active', '1')->first();
    return view('dashboards.system-administrator-home', compact('activeSchoolYear'));
}

/**
 * Redirects to system administrator home page with necessary data
 *
 * @return \Illuminate\Http\Response
 */
protected function redirectToSchoolManagementHome(){
    return view('dashboards.system-administrator-home');
}

/**
 * Redirects to faculty home page with necessary data
 *
 * @return \Illuminate\Http\Response
 */
protected function redirectToFacultyHome(){
    // $facultyLoadings = Auth::user()->facultyLoadings->pluck('id');
    // $gradeSectionSubjects = GradeSectionSubject::whereIn('faculty_loading_id', $facultyLoadings)-
>get();
    $gradeSectionSubjects = Auth::user()->gradeSectionSubjects;
    return view('dashboards.faculty-home', compact('gradeSectionSubjects'));
}

/**
 * Redirects to student home page with necessary data

```

```

*
* @return \Illuminate\Http\Response
*/
protected function redirectToStudentHome(){
    $activeSchoolYear = DB::table('school_years')->where('active', '1')->first();
    $gradeSection = Auth::user()->gradeSections->where('school_year_id', $activeSchoolYear->id)-
>first();
    $gradeSectionSubjects = GradeSectionSubject::where('grade_section_id', $gradeSection->id)-
>get();
    return view('dashboards.student-home', compact('gradeSectionSubjects'));
}
}

```

App\Http\Controllers\UsersController.php

```
<?php
```

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

```
use Illuminate\Http\Request;
```

```
use App\Http\Requests;
```

```
use App\Http\Requests\UserRequest;
```

```
use App\Http\Requests\UserDetailRequest;
```

```
use App\Models\User;
```

```
use App\Models\GradeSections\GradeSection;
```

```
use DB;
```

```
use Random;
```

```
use Auth;
```

```
class UsersController extends Controller
```

```
{
```

```
    public function index(){
```

```
        $users = User::get();
```

```
        return view('users.index', compact('users'));
    }
```

```
    public function indexofSchoolManagement(){
```

```
        $users = User::where('role_id', '2')->orderBy('username', 'asc')->get();
```

```
        return view('users.school-management', compact('users'));
    }
```

```
    public function indexofFaculty(){
```

```
        $users = User::where('role_id', '3')->orderBy('username', 'asc')->get();
```

```
        return view('users.faculty', compact('users'));
    }
```

```

}

public function indexOfStudents(){
    $activeSchoolYear = DB::table('school_years')->where('active', '1')->first();
    $users = User::where('role_id', '4')->orderBy('username', 'asc')->get();
    return view('users.students', compact('users', 'activeSchoolYear'));
}

public function create(){
    $roles = DB::table('roles')->lists('title', 'id');
    $randKey = Random::generateString(16,
'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ');
    return view('auth.register', compact('users', 'roles', 'randKey'));
}

public function store(UserRequest $request){
    $user = User::create([
        'username' => $request->input('username'),
        'password' => bcrypt($request->input('password')),
        'role_id' => $request->input('role_id')
    ]);
    switch($request->input('role_id')){
    case '1':
        $this->deactivate(Auth::user());
        return redirect('/logout');
        break;
        case '2':
            DB::table('school_management')->insert([
                'user_id' => $user->id,
                'lastname' => $request->input('lastname'),
                'firstname' => $request->input('firstname'),
                'idNumber' => $request->input('idNumberPre') . '-' . $request->input('idNumberPost'),
                'gender' => $request->input('gender'),
            ]);
                return redirect('/users/school-management');
                break;
                case '3':
                    DB::table('faculty')->insert([
                        'user_id' => $user->id,
                        'lastname' => $request->input('lastname'),
                        'firstname' => $request->input('firstname'),
                        'idNumber' => $request->input('idNumberPre') . '-' . $request->input('idNumberPost'),
                        'gender' => $request->input('gender'),
                    ]);
    }
}

```

```

return redirect('/users/faculty');
        break;
    case '4':
DB::table('students')->insert([
    'user_id' => $user->id,
    'lastname' => $request->input('lastname'),
    'firstname' => $request->input('firstname'),
    'idNumber' => $request->input('idNumberPre') . '-' . $request->input('idNumberPost'),
    'lrmNumber' => $request->input('lrmNumber'),
    'gender' => $request->input('gender'),
]);
        return redirect('/users/students');
        break;
    }
}

public function edit(User $user){
    $idNumberParts = explode('-', $user->getInfo()->idNumber);
    return view('users.edit', compact('user', 'idNumberParts'));
}

public function update(User $user, UserDetailRequest $request){
    switch($user->role_id){
        case '2':
            DB::table('school_management')->where('user_id', $user->id)->update([
                'lastname' => $request->input('lastname'),
                'firstname' => $request->input('firstname'),
                'idNumber' => $request->input('idNumberPre') . '-' . $request->input('idNumberPost')
            ]);
            return redirect('/users/school-management');
            break;
        case '3':
            DB::table('faculty')->where('user_id', $user->id)->update([
                'lastname' => $request->input('lastname'),
                'firstname' => $request->input('firstname'),
                'idNumber' => $request->input('idNumberPre') . '-' . $request->input('idNumberPost')
            ]);
            return redirect('/users/faculty');
            break;
        case '4':
            DB::table('students')->where('user_id', $user->id)->update([
                'lastname' => $request->input('lastname'),
                'firstname' => $request->input('firstname'),
                'idNumber' => $request->input('idNumberPre') . '-' . $request->input('idNumberPost'),
            ]);
            return redirect('/users/students');
            break;
    }
}

```

```

        'lrnNumber' => $request->input('lrnNumber'),
    ]);
    return redirect('/users/students');
    break;
}
}

```

```

public function activateUser(User $user){
    $user->active = 1;
    $user->update();
    switch ($user->role_id){
        case '2':
            return redirect('/users/school-management');
            break;
        case '3':
            return redirect('/users/faculty');
            break;
        case '4':
            return redirect('/users/students');
            break;
    }
}

```

```

public function deactivateUser(User $user){
    $this->deactivate($user);
    switch ($user->role_id){
        case '2':
            return redirect('/users/school-management');
            break;
        case '3':
            return redirect('/users/faculty');
            break;
        case '4':
            return redirect('/users/students');
            break;
    }
}

```

```

public function enrollment($gradeSectionSelected = null, $studentsEnrolledInGradeSection = null){
    $activeSchoolYear = DB::table('school_years')->where('active','1')->first();
    $studentsList = User::where('role_id', '4')->orderBy('username', 'asc')->where('active', '1')->get();
    $students = [];
    foreach ($studentsList as $student){

```

```

        $gradeSection = $student->gradeSections->where('school_year_id', $activeSchoolYear->id)-
>first();
        if ($gradeSection == null){
            $students[] = $student;
        }
    }
    $gradeSections = GradeSection::get();
    return view('users.enrollment', compact('students', 'gradeSections', 'gradeSectionSelected',
'studentsEnrolledInGradeSection'));
}

```

```

public function postSearch(Request $request){
    $gradeSection = GradeSection::find($request->input('gradeSectionId'));
    $students = $gradeSection->students;
    return $this->enrollment($gradeSection, $students);
}

```

```

public function enroll(Request $request){
    $gradeSection = GradeSection::find($request->input('enrollSection'));
    $selectedIds = explode(',', $request->input('enrollList'));
    $gradeSection->students()->sync($selectedIds);
    return redirect('/grade-sections/' . $gradeSection->id);
}

```

```

private function deactivate(User $user){
    $user->active = 0;
    $user->update();
}
}

```

App\Http\Controllers>WelcomeController.php
<?php

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

```
use Illuminate\Http\Request;
```

```
use App\Http\Requests;
```

```
use App\Http\Controllers\Controller;
```

```
use App\Http\Controllers\ArticlesController;
```

```
use App\Models\Article;
```

```
class WelcomeController extends Controller
{

```

```

/**
 * Show the main page, the starting point of the site. Also retrieve articles
 * to be shown in main page
 *
 * @return \Illuminate\Http\Response
 */
public function index(){
    return view('welcome');
}
}

```

App\Models\Examinations\Deployment.php

```
<?php
```

```

namespace App\Models\Examinations;

use Illuminate\Database\Eloquent\Model;
use Carbon\Carbon;
use DB;

class Deployment extends Model
{
    protected $fillable = [
        'examination_id',
        'grade_section_subject_id',
        'publish_on',
        'exam_start',
        'exam_end'
    ];

    protected $table = 'deployments';
    public $timestamps = false;

    public function scopePublished($query){
        $query->where('publish_on', '<=', Carbon::now());
    }

    public function getPublishOnAttribute($date){
        return Carbon::parse($date)->format('Y-m-d\TH:i');
    }

    public function setPublishOnAttribute($date){
        $this->attributes['publish_on'] = Carbon::parse($date);
    }
}

```



```

public function getExamStartAttribute($date){
    return Carbon::parse($date)->format('Y-m-d\\TH:i');
}

public function setExamStartAttribute($date){
    $this->attributes['exam_start'] = Carbon::parse($date);
}

public function getExamEndAttribute($date){
    return Carbon::parse($date)->format('Y-m-d\\TH:i');
}

public function setExamEndAttribute($date){
    $this->attributes['exam_end'] = Carbon::parse($date);
}

public function getUnformattedDate($attribute){
    return DB::table('deployments')->where('id', $this->id)->first()->$attribute;
}

public function examination(){
    return $this->belongsTo('App\Models\Examinations\Examination');
}

public function instances(){
    return $this->hasMany('App\Models\Examinations\DeploymentInstance');
}
}

```

App\Models\Examinations\DeploymentAnswer.php

```

<?php

namespace App\Models\Examinations;

use Illuminate\Database\Eloquent\Model;

class DeploymentAnswer extends Model
{
    protected $fillable = [
        'deployment_instance_part_id',
        'question_id',
        'wordbox_item_id',

```

```

        'columns_item_id',
        'answer',
    ];

    protected $table = 'deployment_answers';
    public $timestamps = false;
}

```

App\Models\Examinations\DeploymentInstance.php

```
<?php
```

```

namespace App\Models\Examinations;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;
use DB;

class DeploymentInstance extends Model
{
    use SoftDeletes;

    protected $fillable = [
        'user_id',
        'deployment_id',
        'score',
        'time_started',
        'time_ended',
        'is_finished',
    ];

    protected $table = 'deployment_instances';

    public function getUnformattedDate($attribute){
        return DB::table('deployment_instances')->where('id', $this->id)->first()->$attribute;
    }

    public function deployment(){
        return $this->belongsTo('App\Models\Examinations\Deployment');
    }

    public function user(){
        return $this->belongsTo('App\Models\User');
    }
}

```

```

    public function parts(){
        return $this->hasMany('App\Models\Examinations\DeploymentInstancePart',
'deployment_instance_id');
    }
}

```

App\Http\Models\Examinations\DeploymentInstancePart.php

```
<?php
```

```
namespace App\Models\Examinations;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
use DB;
```

```
class DeploymentInstancePart extends Model
```

```
{
```

```
    protected $fillable = [
        'examination_part_id',
        'question_order',

```

```
];
```

```
    protected $table = 'deployment_instance_parts';
```

```
    public $timestamps = false;
```

```
    public function instance(){
```

```
        return $this->belongsTo('App\Models\Examinations\DeploymentInstance',
'deployment_instance_id');
```

```
    }
```

```
    public function examinationPart(){
```

```
        return $this->belongsTo('App\Models\Examinations\ExaminationPart');
```

```
    }
```

```
    public function getQuestionOrder(){
```

```
        return (explode("|", $this->question_order));
```

```
    }
```

```
}
```

App\Models\Examinations\Examination.php

```
<?php
```

```
namespace App\Models\Examinations;
```

```

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;
use Carbon\Carbon;
use DB;

class Examination extends Model
{
    use SoftDeletes;

    protected $fillable = [
        'user_id',
        'subcategory',
        'description',
        'total_points',
        'is_private',
    ];

    protected $table = 'examinations';

    public function faculty(){
        return $this->belongsTo('App\Models\User', 'user_id');
    }

    public function computeTotalPoints(){
        $totalPoints = 0;
        foreach ($this->parts as $part){
            $quantity = $part->getQuestionsCount();
            $pointsPerItem = $part->points;
            $totalPoints += ($quantity * $pointsPerItem);
        }
        return $totalPoints;
    }

    public function parts(){
        return $this->hasMany('App\Models\Examinations\ExaminationPart', 'examination_id');
    }

    public function deployments(){
        return $this->hasMany('App\Models\Examinations\Deployment');
    }
}

```

App\Http\Models\ExaminationPart.php

<?php

```

namespace App\Models\Examinations;

use Illuminate\Database\Eloquent\Model;
use DB;

class ExaminationPart extends Model
{
    protected $fillable = [
        'question_type_id',
        'points',
        'questions_quantity'
    ];

    protected $table = 'examination_parts';
    public $timestamps = false;

    public function getQuestionType(){
        return DB::table('question_types')->where('id', $this->question_type_id)->first()->name;
    }

    public function getQuestionsCount(){
        $itemsQuantity = DB::table('examination_part_items')->where('examination_part_id', $this->id)-
        >pluck('quantity');
        return array_sum($itemsQuantity);
    }

    public function examination(){
        return $this->belongsTo('App\Models\Examinations\Examination');
    }

    public function items(){
        return $this->hasMany('App\Models\Examinations\ExaminationPartItem', 'examination_part_id');
    }

    public function instances(){
        return $this->hasMany('App\Models\Examinations\DeploymentInstancePart', 'examination_part_id');
    }
}

```

App\Http\Models\Examinations\ExaminationPartItem.php

```
<?php
```

```
namespace App\Models\Examinations;
```

```

use Illuminate\Database\Eloquent\Model;

class ExaminationPartItem extends Model
{
    protected $fillable = [
        'question_subtopic_id',
        'quantity',
        'choices_quantity',
        'question_id'
    ];

    protected $table = 'examination_part_items';
    public $timestamps = false;

    public function part(){
        return $this->belongsTo('App\Models\Examinations\ExaminationPart', 'examination_part_id');
    }

    public function questionSubtopic(){
        return $this->belongsTo('App\Models\Questions\QuestionSubtopic');
    }
}

```

App\Models\GradeSections\FacultyLoading.php

```
<?php
```

```

namespace App\Models\GradeSections;

use Illuminate\Database\Eloquent\Model;

class FacultyLoading extends Model
{
    protected $fillable = [
        'subject_id',
        'user_id',
        'school_year_id'
    ];

    protected $table = 'faculty_loadings';
    public $timestamps = false;

    public function faculty(){
        return $this->belongsTo('App\Models\User', 'user_id');
    }
}

```

```

    }

    public function gradeSectionSubject(){
        return $this->belongsTo('App\Models\GradeSectionSubjects\GradeSectionSubject');
    }

    public function schoolYear(){
        return $this->belongsTo('App\Models\GradeSections\SchoolYear');
    }
}

```

app\Models\GradeSections\GradeSection

<?php

```

namespace App\Models\GradeSections;

use Illuminate\Database\Eloquent\Model;
use App\Models\GradeSectionSubjects\GradeSectionSubject;
use DB;

class GradeSection extends Model
{
    protected $fillable = [
        'grade_section_name_id',
        'school_year_id'
    ];

    protected $table = 'grade_sections';
    public $timestamps = false;

    public function getName(){
        return $this->belongsTo('App\Models\GradeSections\GradeSectionName',
'grade_section_name_id');
    }

    public function students(){
        return $this->belongsToMany('App\Models\User', 'enrollments', 'grade_section_id', 'user_id');
    }

    public function subjects(){
        return $this->belongsToMany('App\Models\GradeSections\Subject', 'grade_section_subjects');
    }

    public function schoolYear(){

```

```

        return $this->belongsTo('App\Models\GradeSections\SchoolYear');
    }

    public function getGradeSectionSubject($subject){
        return GradeSectionSubject::where('subject_id', $subject->id)->where('grade_section_id', $this->id)-
>first();
    }
}

```

\App\Models\GradeSections\GradeSectionName

<?php

```

namespace App\Models\GradeSections;

use Illuminate\Database\Eloquent\Model;
use DB;

class GradeSectionName extends Model
{
    protected $table = 'grade_section_names';
    public $timestamps = false;

    public function getGradeLevel(){
        return DB::table('grade_levels')->find($this->grade_level_id)->name;
    }

    public function gradeSection(){
        return $this->hasMany('App\Models\GradeSections\GradeSection');
    }
}

```

\App\ModelsGradeSections\SchoolYear.php

<?php

```

namespace App\Models\GradeSections;

use Illuminate\Database\Eloquent\Model;

class SchoolYear extends Model
{
    protected $fillable = ['name', 'active'];
    protected $table = 'school_years';
    public $timestamps = false;
}

```


App\Models\GradeSectionSubjects\GradeSectionSubject.php

<?php

```
namespace App\Models\GradeSectionSubjects;

use Illuminate\Database\Eloquent\Model;
use App\Models\GradeSections\FacultyLoading;
use DB;

class GradeSectionSubject extends Model
{
    protected $table = 'grade_section_subjects';
    public $timestamps = false;

    public function faculty(){
        $activeSchoolYear = DB::table('school_years')->where('active','1')->first();
        return $this->belongsToMany('App\Models\User', 'faculty_loadings')->where('school_year_id',
        $activeSchoolYear->id);
    }

    public function gradeSection(){
        return $this->belongsTo('App\Models\GradeSections\GradeSection');
    }

    public function subject(){
        return $this->belongsTo('App\Models\GradeSections\Subject');
    }
}
```

App\Models\GradeSectionSubjectPost.php

<?php

```
namespace App\Models\GradeSectionSubjects;

use Illuminate\Database\Eloquent\Model;
use Carbon\Carbon;
use DB;

class GradeSectionSubjectPost extends Model
{
    protected $fillable = ['post_id', 'grade_section_subject_id', 'publish_on'];
    protected $table = 'subject_posts';
    public $timestamps = false;
}
```

```

protected $dates = ['publish_on'];

public function scopePublished($query){
    $query->where('publish_on', '<=', Carbon::now());
}

public function getPublishOnAttribute($date){
    return Carbon::parse($date)->format('Y-m-d\\TH:i');
}

public function setPublishOnAttribute($date){
    $this->attributes['publish_on'] = Carbon::parse($date);
}

public function getUnformattedDate($attribute){
    return DB::table('subject_posts')->find($this->id)->$attribute;
}

public function gradeSectionSubject(){
    return $this->belongsTo('App\Models\GradeSectionSubjects\GradeSectionSubject');
}

public function post(){
    return $this->belongsTo('App\Models\PostsAndRequirements\Post');
}
}

```

GradeSectionSubjectRequirement.php

```
<?php
```

```

namespace App\Models\GradeSectionSubjects;

use Illuminate\Database\Eloquent\Model;
use Carbon\Carbon;
use DB;

class GradeSectionSubjectRequirement extends Model
{
    protected $fillable = [
        'requirement_id',
        'grade_section_subject_id',
        'publish_on',
        'event_start',
        'event_end'
    ]
}

```

```

];

protected $table = 'subject_requirements';
public $timestamps = false;

protected $dates = ['publish_on', 'event_start', 'event_end'];

public function scopePublished($query){
    $query->where('publish_on', '<=', Carbon::now());
}

public function scopeActive($query){
    $query->where('event_start', '<=', Carbon::now());
    $query->where('event_end', '>=', Carbon::now());
}

public function getPublishOnAttribute($date){
    return Carbon::parse($date)->format('Y-m-d\\TH:i');
}

public function setPublishOnAttribute($date){
    $this->attributes['publish_on'] = Carbon::parse($date);
}

public function getEventStartAttribute($date){
    return Carbon::parse($date)->format('Y-m-d\\TH:i');
}

public function setEventStartAttribute($date){
    $this->attributes['event_start'] = Carbon::parse($date);
}

public function getEventEndAttribute($date){
    return Carbon::parse($date)->format('Y-m-d\\TH:i');
}

public function setEventEndAttribute($date){
    $this->attributes['event_end'] = Carbon::parse($date);
}

public function getUnformattedDate($attribute){
    return DB::table('subject_requirements')->find($this->id)->$attribute;
}

```

```

public function gradeSectionSubject(){
    return $this->belongsTo('App\Models\GradeSectionSubjects\GradeSectionSubject');
}

public function requirement(){
    return $this->belongsTo('App\Models\PostsAndRequirements\Requirement');
}

public function instances(){
    return $this-
>hasMany('App\Models\GradeSectionSubjects\GradeSectionSubjectRequirementInstance',
'subject_requirement_id');
}
}

```

GradeSectionSubjectRequirementInstance.php

```
<?php
```

```
namespace App\Models\GradeSectionSubjects;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
use DB;
```

```
class GradeSectionSubjectRequirementInstance extends Model
```

```
{
    protected $fillable = [
        'user_id',
        'file_id',
        'submitted_at',
    ];

    protected $table = 'subject_requirement_instances';
    protected $dates = ['submitted_at'];
    public $timestamps = false;

    public function subjectRequirement(){
        return $this-
>belongsTo('App\Models\GradeSectionSubjects\GradeSectionSubjectRequirement',
'subject_requirement_id');
    }

    public function user(){
        return $this->belongsTo('App\Models\User');
    }
}

```

```

        public function getFile(){
            return DB::table('files')->where('id', $this->file_id)->first();
        }
    }
}

```

App\Models\SubjectClassRecord.php

```
<?php
```

```
namespace App\Models\GradeSectionSubjects;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class SubjectClassRecord extends Model
```

```

{
    protected $fillable = [
        'grade_section_subject_id',
        'examination_id',
    ];

    protected $table = "subject_class_records";
    public $timestamps = false;

    public function gradeSectionSubject(){
        return $this->belongsTo('App\Models\GradeSectionSubjects\GradeSectionSubject');
    }

    public function examination(){
        return $this->belongsTo('App\Models\Examinations\Examination');
    }

    public function instances(){
        return $this->hasMany('App\Models\GradeSectionSubjects\SubjectClassRecordInstance',
'subject_class_record_id');
    }
}

```

App\Models\GradeSectionSubjects\SubjectClassRecordInstance.php

```
<?php
```

```
namespace App\Models\GradeSectionSubjects;
```

```
use Illuminate\Database\Eloquent\Model;
```

```

class SubjectClassRecordInstance extends Model
{
    protected $fillable = [
        'subject_class_record_id',
        'user_id',
        'score',
    ];

    protected $table = "subject_class_record_instances";
    public $timestamps = false;

    public function classRecord(){
        return $this->belongsTo('App\Models\Subjects\SubjectClassRecord',
'subject_class_record_id');
    }
}

```

App\Models\PostsAndRequirements\Posts.php

```
<?php
```

```
namespace App\Models\PostsAndRequirements;
```

```
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;
use DB;
```

```
class Post extends Model
```

```

{
    use SoftDeletes;

    protected $fillable = [
        'user_id',
        'title',
        'body',
    ];

    public function gradeSectionSubjects(){
        return $this->belongsToMany('App\Models\GradeSectionSubjects\GradeSectionSubject',
'subject_posts');
    }

    public function faculty(){

```

```

        return $this->belongsTo('App\Models\User', 'user_id');
    }

    public function files(){
        return $this->belongsToMany('App\Models\File', 'post_files');
    }
}

```

App\Models\PostsAndRequirements\Requirement.php

<?php

```
namespace App\Models\PostsAndRequirements;
```

```
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;
use DB;
```

```
class Requirement extends Model
```

```
{
```

```
    use SoftDeletes;
```

```
    protected $fillable = [
```

```
        'user_id',
```

```
        'title',
```

```
        'body',
```

```
    ];
```

```
    public function gradeSectionSubjects(){
```

```
        return $this->belongsToMany('App\Models\GradeSectionSubjects\GradeSectionSubject',
'subject_posts');
```

```
    }
```

```
    public function faculty(){
```

```
        return $this->belongsTo('App\Models\User', 'user_id');
```

```
    }
```

```
    public function files(){
```

```
        return $this->belongsToMany('App\Models\File', 'requirement_files');
```

```
    }
```

```
    // public function instances(){
```

```
        // return $this->hasMany('App\Models\Subjects\SubjectRequirementInstance');
```

```
    // }
```

```
}
```

App\Models\File.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class File extends Model
{
    protected $fillable = [
        'destination_path',
        'file_name',
    ];

    public function posts(){
        return $this->belongsToMany('App\Models\PostsAndRequirements\Post', 'post_files');
    }

    public function requirements(){
        return $this->belongsToMany('App\Models\PostsAndRequirements\Requirement',
'requirement_files');
    }

    public function user(){
        return $this->belongsTo('App\Models\User');
    }

    public function downloadHistory(){
        return $this->hasMany('App\Models\FileDownloadHistory');
    }
}
```

App\Models\FileDownloadHistory

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use DB;

class FileDownloadHistory extends Model
{
    protected $fillable = [
        'user_id',
```



```

        'downloaded_at',
    ];

    protected $table = 'file_download_history';
    public $timestamps = false;

    public function file(){
        return $this->belongsTo('App\Models\File');
    }

    public function getUser(){
        return DB::table('users')->where('id', $this->user_id)->first();
    }
}

```

App\Models\Questions\Types\QuestionFillintheblanks.php

```
<?php
```

```

namespace App\Models\Questions\Types;

use Illuminate\Database\Eloquent\Model;

class QuestionFillInTheBlanks extends Model
{
    protected $fillable = ['right_answer'];
    protected $table = 'question_fill_in_the_blanks';

    public function question(){
        return $this->belongsTo('App\Models\Questions\Question');
    }
}

```

App\Models\Questions\Types\QuestionMatchColumnsChoice.php

```
<?php
```

```

namespace App\Models\Questions\Types;

use Illuminate\Database\Eloquent\Model;

class QuestionMatchColumnsChoice extends Model
{
    protected $fillable = ['text'];
    protected $table = 'question_match_columns_choices';
    public $timestamps = false;
}

```

```

        public function question(){
            return $this->belongsTo('App\Models\Questions\Question');
        }

        public function item(){
            return $this->hasOne('App\Models\Questions\Types\QuestionMatchColumnsItem',
'columns_choice_id');
        }
    }
}

```

App\Models\Questions\Types\QuestionMatchColumnsItem.php

```
<?php
```

```

namespace App\Models\Questions\Types;

use Illuminate\Database\Eloquent\Model;

class QuestionMatchColumnsItem extends Model
{
    protected $fillable = ['text', 'columns_choice_id'];
    protected $table = 'question_match_columns_items';
    public $timestamps = false;

    public function choice(){
        return $this->belongsTo('App\Models\Questions\Types\QuestionMatchColumnsChoice',
'columns_choice_id');
    }
}

```

App\Models\Questions\Types\QuestionMultipleChoice.php

```
<?php
```

```

namespace App\Models\Questions\Types;

use Illuminate\Database\Eloquent\Model;
use DB;

class QuestionMultipleChoice extends Model
{
    protected $fillable = [
        'text',
        'is_right_answer',
    ];
}

```

```

protected $table = 'question_multiple_choice';

public function question(){
    return $this->belongsTo('App\Models\Questions\Question');
}
}

```

App\Models\Questions\Types\QuestionSelectFromTheWordboxChoice.php <?php

```

namespace App\Models\Questions\Types;

use Illuminate\Database\Eloquent\Model;

class QuestionSelectFromTheWordboxChoice extends Model
{
    protected $fillable = ['text'];
    protected $table = 'question_select_from_the_wordbox_choices';
    public $timestamps = false;

    public function question(){
        return $this->belongsTo('App\Models\Questions\Question');
    }

    public function items(){
        return $this->hasMany('App\Models\Questions\Types\QuestionSelectFromTheWordboxItem',
'wordbox_choice_id');
    }
}

```

App\Models\Questions\Types\QuestionSelectFromTheWordboxItem.php <?php

```

namespace App\Models\Questions\Types;

use Illuminate\Database\Eloquent\Model;

class QuestionSelectFromTheWordboxItem extends Model
{
    protected $fillable = ['text', 'wordbox_choice_id'];
    protected $table = 'question_select_from_the_wordbox_items';
    public $timestamps = false;
}

```

```

        public function choice(){
            return $this->
>belongsTo('App\Models\Questions\Types\QuestionSelectFromTheWordboxChoice',
'wordbox_choice_id');
        }
    }
}

```

App\Models\Questions\Types\QuestionTrueOrFalse.php

```
<?php
```

```

namespace App\Models\Questions\Types;

use Illuminate\Database\Eloquent\Model;

class QuestionTrueOrFalse extends Model
{
    protected $fillable = ['right_answer'];
    protected $table = 'question_true_or_false';

    public function question(){
        return $this->belongsTo('App\Models\Questions\Question');
    }
}

```

App\Models\Questions

```
<?php
```

```

namespace App\Models\Questions;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;
use DB;

class Question extends Model
{
    use SoftDeletes;

    protected $fillable = [
        'question_type_id',
        'question_subtopic_id',
        'title',
        'body',
        'points',
        'user_id',
    ]
}

```

```

];

public function move($categoryName, $topicName, $subtopicName){
    $category = QuestionCategory::where('name', '=', $categoryName)->first();
    $topic = QuestionTopic::where('name', '=', $topicName)->where('question_category_id', '=',
$category->id)->first();
    $subtopic = QuestionSubtopic::where('name', '=', $subtopicName)->where('question_topic_id', '=',
$topic->id)->first();

    if (empty($category) || empty($topic) || empty($subtopic)){
        return redirect('home');
    } else {
        $this->question_subtopic_id = $subtopic->id;
        $this->save();
    }
}

public function getQuestionType(){
    return DB::table('question_types')->where('id', $this->question_type_id)->first()->name;
}

public function getAuthor(){
    return DB::table('users')->where('id', $this->user_id)->first()->username;
}

public function questionSubtopic(){
    return $this->belongsTo('App\Models\Questions\QuestionSubtopic');
}

public function subjectExaminations(){
    return $this->belongsToMany('App\Models\Subjects\SubjectExamination', 'examination_questions',
'examination_id', 'question_id');
}

public function trueOrFalse(){
    return $this->hasOne('App\Models\Questions\Types\QuestionTrueOrFalse');
}

public function fillInTheBlanks(){
    return $this->hasOne('App\Models\Questions\Types\QuestionFillInTheBlanks');
}

public function multipleChoice(){
    return $this->hasMany('App\Models\Questions\Types\QuestionMultipleChoice');
}

```

```

    }

    public function matchColumnsChoices(){
        return $this->hasMany('App\Models\Questions\Types\QuestionMatchColumnsChoice');
    }

    public function selectFromTheWordboxChoices(){
        return $this->hasMany('App\Models\Questions\Types\QuestionSelectFromTheWordboxChoice');
    }
}

```

App\Models\QuestionCategory.php

<?php

```

namespace App\Models\Questions;

use Illuminate\Database\Eloquent\Model;

class QuestionCategory extends Model
{
    protected $fillable = ['name'];
    protected $table = 'question_categories';

    public function questionTopics(){
        return $this->hasMany('App\Models\Questions\QuestionTopic')->orderBy('name', 'asc');
    }
}

```

App\Models\QuestionSubtopic.php

<?php

```

namespace App\Models\Questions;

use Illuminate\Database\Eloquent\Model;

class QuestionSubtopic extends Model
{
    protected $fillable = ['name'];

    public function questionTopic(){
        return $this->belongsTo('App\Models\Questions\QuestionTopic');
    }

    public function questions(){

```

```

        return $this->hasMany('App\Models\Questions\Question')->orderBy('question_type_id',
'asc');
    }

    public function examinationPartItems(){
        return $this->hasMany('App\Models\Subjects\SubjectExaminationPartItem',
'question_subtopic_id');
    }
}

```

App\Models\Questions\QuestionTopic.php

```
<?php
```

```
namespace App\Models\Questions;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class QuestionTopic extends Model
```

```
{
```

```
    protected $fillable = ['name'];
```

```
    public function questionCategory(){
```

```
        return $this->belongsTo('App\Models\Questions\QuestionCategory',
'question_category_id');
```

```
    }
```

```
    public function questionSubtopics(){
```

```
        return $this->hasMany('App\Models\Questions\QuestionSubtopic')->orderBy('name', 'asc');
```

```
    }
}
```

Resources\Views\Auth\Emails\password.blade

Click here to reset your password: getEmailForPasswordReset()) }} }}"> {{ \$link }}

Resources\Views\Auth\Passwords\Change.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Change password
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    <div class="row">
```

```
        <div class="col-md-8 col-md-offset-2">
```

```
            <div class="panel panel-warning">
```

```
                <div class="panel-heading">Change password</div>
```

```
                <div class="panel-body">
```

```
                    {!! Form::open(['url' => 'password/change']) !!} 
```

```
                    <div class="form-group{{ $errors->has('oldPassword') ?
```

```
' has-error' : " }}">
```

```
                        {!! Form::label('oldPassword', 'Old password:',
```

```
                        ['class' => 'col-md-4 control-label']) !!} 
```

```
                        <div class="col-md-6">
```

```
                            <input type="password" class="form-
```

```
                            control" name="oldPassword"></input>
```

```
                            @if ($errors->has('oldPassword'))
```

```
                                <span class="help-block">
```

```
                                    <strong>{{ $errors->first('oldPassword')
```

```
                                    }}</strong>
```

```
                                </span>
```

```
                            @endif
```

```
                            </div>
```

```
                        </div>
```

```
                    <div class="form-group{{ $errors->has('newPassword')
```

```
? ' has-error' : " }}">
```

```
                        {!! Form::label('newPassword', 'New password:',
```

```
                        ['class' => 'col-md-4 control-label']) !!} 
```

```
                        <div class="col-md-6">
```

```
                            {!! Form::text('newPassword', null,
```

```
                            ['class' => 'form-control', 'maxlength' => '255']) !!} 
```



```

        @if ($errors->has('newPassword'))
        <span class="help-block">
            <strong>{{ $errors->first('newPassword')
}}</strong>

        </span>
        @endif
    </div>

    <div class="form-group{{ $errors-
>has('newPassword_confirmation') ? ' has-error' : " }}">
        {!! Form::label('newPassword_confirmation',
'Confirm new password:', ['class' => 'col-md-4 control-label']) !!}
        <div class="col-md-6">
            {!!
Form::text('newPassword_confirmation', null, ['class' => 'form-control', 'maxlength' => '255']) !!}

            @if ($errors-
>has('newPassword_confirmation'))

            <span class="help-block">
                <strong>{{ $errors-
>first('newPassword_confirmation') }}</strong>

            </span>
            @endif
        </div>

    </div>

    <div class="form-group">
        <div class="col-md-6 pull-right">
            <input class="btn btn-default" type="button" onclick="generatePassword()"
value="Generate Password">
        </div>
    </div>
    <br></br>

    <div class="form-group pull-right">
        <div class="col-md-6 col-md-offset-4">
            {!! Form::submit('Change Password',
[class' => 'btn btn-warning form-control']) !!}
        </div>
    </div>
    {!! Form::close() !!}
</div>
</div>

```

```

        </div>
    </div>
</div>
<script>
function generatePassword(){
    var string = "<?php echo $randKey ?>";
    document.getElementById("newPassword").value = string;
    document.getElementById("newPassword_confirmation").value = string;
}
</script>
@endsection

```

Resources\Views\Auth\Passwords\Email.blade

```

@extends('layouts.app')

<!-- Main Content -->
@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-8 col-md-offset-2">
            <div class="panel panel-default">
                <div class="panel-heading">Reset Password</div>
                <div class="panel-body">
                    @if (session('status'))
                        <div class="alert alert-success">
                            {{ session('status') }}
                        </div>
                    @endif

                    <form class="form-horizontal" role="form" method="POST" action="{{
url('/password/email') }}">
                        {!! csrf_field() !!}

                        <div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">
                            <label class="col-md-4 control-label">E-Mail Address</label>

                            <div class="col-md-6">
                                <input type="email" class="form-control" name="email" value="{{ old('email') }}">

                                @if ($errors->has('email'))
                                    <span class="help-block">
                                        <strong>{{ $errors->first('email') }}</strong>
                                    </span>
                                @endif
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>

```

```

        </div>
    </div>

    <div class="form-group">
        <div class="col-md-6 col-md-offset-4">
            <button type="submit" class="btn btn-primary">
                <i class="fa fa-btn fa-envelope"></i>Send Password Reset Link
            </button>
        </div>
    </div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
@endsection

```

Resources\Views\Auth\Passwords\Reset.blade

```

@extends('layouts.app')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-8 col-md-offset-2">
            <div class="panel panel-default">
                <div class="panel-heading">Reset Password</div>

                <div class="panel-body">
                    <form class="form-horizontal" role="form" method="POST" action="{{
url('/password/reset') }}">
                        {!! csrf_field() !!}

                        <input type="hidden" name="token" value="{{ $token }}">

                        <div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">
                            <label class="col-md-4 control-label">E-Mail Address</label>

                            <div class="col-md-6">
                                <input type="email" class="form-control" name="email" value="{{ $email or
old('email') }}">

                                @if ($errors->has('email'))
                                    <span class="help-block">

```

```

        <strong>{{ $errors->first('email') }}</strong>
    </span>
    @endif
</div>
</div>

<div class="form-group"{{ $errors->has('password') ? ' has-error' : " }}">
    <label class="col-md-4 control-label">Password</label>

    <div class="col-md-6">
        <input type="password" class="form-control" name="password">

        @if ($errors->has('password'))
            <span class="help-block">
                <strong>{{ $errors->first('password') }}</strong>
            </span>
        @endif
    </div>
</div>

<div class="form-group"{{ $errors->has('password_confirmation') ? ' has-error' : " }}">
    <label class="col-md-4 control-label">Confirm Password</label>
    <div class="col-md-6">
        <input type="password" class="form-control" name="password_confirmation">

        @if ($errors->has('password_confirmation'))
            <span class="help-block">
                <strong>{{ $errors->first('password_confirmation') }}</strong>
            </span>
        @endif
    </div>
</div>

<div class="form-group">
    <div class="col-md-6 col-md-offset-4">
        <button type="submit" class="btn btn-primary">
            <i class="fa fa-btn fa-refresh"></i>Reset Password
        </button>
    </div>
</div>
</form>
</div>
</div>
</div>

```

```
</div>
</div>
@endsection
```

Resources\Views\Auth\Login.blade

```
@extends('layouts.app')
```

```
@section('title')
    Login
@endsection
```

```
@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-8 col-md-offset-2">
            @include('flash::message')
            <div class="panel panel-default">
                <div class="panel-heading">Login</div>
                <div class="panel-body">
                    <form class="form-horizontal" role="form" method="POST" action="{{ url('/login') }}">
                        {!! csrf_field() !!}

                        <div class="form-group{{ $errors->has('username') ? ' has-error' : "" }}">
                            <label class="col-md-4 control-label">Username</label>

                            <div class="col-md-6">
                                <input type="username" class="form-control" name="username" value="{{
empty($user) ? old('username') : $user->username }}">

                                @if ($errors->has('username'))
                                    <span class="help-block">
                                        <strong>{{ $errors->first('username') }}</strong>
                                    </span>
                                @endif
                            </div>
                        </div>

                        <div class="form-group{{ $errors->has('password') ? ' has-error' : "" }}">
                            <label class="col-md-4 control-label">Password</label>

                            <div class="col-md-6">
                                <input type="password" class="form-control" name="password" value="{{
empty($user) ? "" : $user->password }}">

```

```

        @if ($errors->has('password'))
            <span class="help-block">
                <strong>{{ $errors->first('password') }}</strong>
            </span>
        @endif
    </div>
</div>

<div class="form-group">
    <div class="col-md-6 col-md-offset-4">
        <button type="submit" class="btn btn-primary">
            <span class="glyphicon glyphicon-log-in"></span>&nbsp; Login
        </button>
        <!--
        <a class="btn btn-link" href="{{ url('/password/reset') }}">Forgot Your
Password?</a>
        -->
    </div>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
@endsection

```

Resources\Views\Auth\Register.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Register
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    <ol class="breadcrumb pull-right">
```

```
        <li><a href="/home">Home</a></li>
```

```
        <li class="active">Register</li>
```

```
    </ol>
```

```
    <br></br><hr/>
```

```
    <div class="row">
```

```
        <div class="col-md-8 col-md-offset-2">
```

```
            <div class="panel panel-default">
```

```

<div class="panel-heading">Register</div>
<div class="panel-body">
  {!! Form::model($user = new \App\Models\User, ['url' => '/users']) !!}
  {!! csrf_field() !!}

  <div class="form-group"{{ $errors->has('role_id') ? ' has-error' : '' }}">
    {!! Form::label('role_id', 'Role', ['class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
      {!! Form::select('role_id', $roles, null, ['class' => 'form-control', 'id' => 'role_id',
'placeholder' => 'Select from the following...', 'onchange' => 'showLRNField()']) !!}

      @if ($errors->has('role_id'))
        <span class="help-block">
          <strong>{{ $errors->first('role_id') }}</strong>
        </span>
      @endif
    </div>
  </div>

  <div class="form-group"{{ $errors->has('username') ? ' has-error' : '' }}">
    <label class="col-md-4 control-label">Username</label>

    <div class="col-md-6">
      <input type="text" class="form-control" name="username" value="{{ old('username')
}}">

      @if ($errors->has('username'))
        <span class="help-block">
          <strong>{{ $errors->first('username') }}</strong>
        </span>
      @endif
    </div>
  </div>

  <div class="form-group"{{ $errors->has('password') ? ' has-error' : '' }}">
    <label class="col-md-4 control-label">Password</label>

    <div class="col-md-6">
      <input type="text" id="password" class="form-control" name="password">

      @if ($errors->has('password'))
        <span class="help-block">
          <strong>{{ $errors->first('password') }}</strong>
        </span>
      @endif
    </div>
  </div>

```

```

        @endif
    </div>
</div>

<div class="form-group"{{ $Errors->has('password_confirmation') ? ' has-error' : " }}">
    <label class="col-md-4 control-label">Confirm Password</label>

    <div class="col-md-6">
        <input type="text" id="password_confirmation" class="form-control"
name="password_confirmation">

        @if ($Errors->has('password_confirmation'))
            <span class="help-block">
                <strong>{{ $Errors->first('password_confirmation') }}</strong>
            </span>
        @endif
    </div>
</div>

<div class="form-group">
    <div class="col-md-6 pull-right">
        <input class="btn btn-default" type="button" onclick="generatePassword()"
value="Generate Password">
    </div>
</div>
<br></br>

<div class="form-group"{{ $Errors->has('lastname') ? ' has-error' : " }}">
    <label class="col-md-4 control-label">Lastname</label>

    <div class="col-md-6">
        <input type="text" class="form-control" name="lastname" value="{{ old('lastname')
}}">

        @if ($Errors->has('lastname'))
            <span class="help-block">
                <strong>{{ $Errors->first('lastname') }}</strong>
            </span>
        @endif
    </div>
</div>

<div class="form-group"{{ $Errors->has('firstname') ? ' has-error' : " }}">
    <label class="col-md-4 control-label">Firstname</label>

```



```

<div class="col-md-6">
  <input type="text" class="form-control" name="firstname" value="{{ old('firstname')
}}">

```

```

    @if ($errors->has('firstname'))
      <span class="help-block">
        <strong>{{ $errors->first('firstname') }}</strong>
      </span>
    @endif
  </div>
</div>

```

```

<div class="form-group{{ $errors->has('gender') ? ' has-error' : '' }}">
  <div class="col-md-6">
    <label class="radio-inline">
      {!! Form::radio('gender', '1') !!} Male
    </label>
    <label class="radio-inline">
      {!! Form::radio('gender', '0') !!} Female
    </label>
    @if ($errors->has('gender'))
      <span class="help-block">
        <strong>{{ $errors->first('gender') }}</strong>
      </span>
    @endif
  </div>
</div>

```

```

<div id="lrnDiv" style="display:none">
  <div class="form-group{{ $errors->has('lrnNumber') ? ' has-error' : '' }}">
    <label class="col-md-4 control-label">LRN</label>

    <div class="col-md-6">
      <input id="lrnField" pattern=".{12,}" type="text" class="form-control"
maxlength="12" name="lrnNumber" required value="{{ old('lrnNumber') }}" disabled>

      @if ($errors->has('lrnNumber'))
        <span class="help-block">
          <strong>{{ $errors->first('lrnNumber') }}</strong>
        </span>
      @endif
    </div>
  </div>

```



```

var lrnDiv = document.getElementById('lrnDiv');
var lrnField = document.getElementById('lrnField');
var role_id = document.getElementById('role_id').value;

if (role_id == '4'){
    lrnDiv.style = "display:";
    lrnField.disabled = false;
} else {
    lrnDiv.style = "display:none";
    lrnField.disabled = true;
}
}
</script>
@endsection

```

Resources\Views\Classes\Content\Partials\ examination.blade

```

<?php $deployment = $article ?>
<div class="panel panel-warning">
    <div class="panel-heading">
        <button id='span_right_{{ $i }}' type="button" class="btn btn-default btn-xs"
onclick="showhide('{{ $i }}', 1)" style="display:;>
            <span class="glyphicon glyphicon-menu-right"></span>
        </button>
        <button id='span_down_{{ $i }}' type="button" class="btn btn-default btn-xs"
onclick="showhide('{{ $i }}', 0)" style="display: none;>
            <span class="glyphicon glyphicon-menu-down"></span>
        </button>
        <span class="glyphicon glyphicon-list-alt"></span>
        {{ $deployment->examination->description }}
        <button type="button" class="btn btn-warning pull-right" onclick="location.href='/classes/{{
$gradeSectionSubject->id }}/deployments/{{ $deployment->id }}/instances'">
            Check examination status
        </button>
        <br></br>
    </div>
    <div id="{{ $i }}" style="display:none;>
        <table class="table table-responsive">
            <tr>
                <th colspan="2">Subcategory:</th><td colspan="2">{{ $deployment->examination-
>subcategory }}</td>
            </tr>
            <tr>
                <th colspan="2">Total points:</th><td colspan="2">{{ $deployment->examination-
>computeTotalPoints() }}</td>

```

```

</tr>
<tr>
  <th>Exam start:</th><td>{{ $deployment->getUnformattedDate('exam_start') }}</td>
  <th>Exam end:</th><td>{{ $deployment->getUnformattedDate('exam_end') }}</td>
</tr>
<tr>
  <th>Published on:</th><td>{{ $deployment->getUnformattedDate('publish_on') }}</td>
  <td colspan="2"></td>
</tr>
</table>
</div>
</div>

```

Resources\Views\Classes\Content\Partials\ post.blade

```

<?php $subjectPost = $article ?>
<div class="panel panel-info">
  <div class="panel-heading">
    <button id='span_right_{{ $i }}' type="button" class="btn btn-default btn-xs"
onlick="showhide('{{ $i }}', 1)" style="display:;>
      <span class="glyphicon glyphicon-menu-right"></span>
    </button>
    <button id='span_down_{{ $i }}' type="button" class="btn btn-default btn-xs"
onlick="showhide('{{ $i }}', 0)" style="display: none;>
      <span class="glyphicon glyphicon-menu-down"></span>
    </button>
    <span class="glyphicon glyphicon-file"></span>
    {{ $subjectPost->post->title }}
  </div>
  <div id="{{ $i }}" style="display:none;>
    <table class="table table-responsive">
      <tr>
        <td colspan="2">{!! $subjectPost->post->body !!</td>
      </tr>
      <tr>
        <th>Published on:</th><td>{{ $subjectPost->getUnformattedDate('publish_on') }}</td>
      </tr>
    </table>
  </div class="well">
    @if (count($subjectPost->post->files) == 0)
      No files attached
    @else
      Files attached:
      <ul>
        @foreach ($subjectPost->post->files as $file)

```

```

        @include('layouts.file')
    @endforeach
</ul>
    @endif
</div>
</div>
</div>

```

Resources\Views\Classes\Content\Partials\ requirement.blade

```

<?php $subjectRequirement = $article ?>
<div class="panel panel-success">
    <div class="panel-heading">
        <button id='span_right_{{ $i }}' type="button" class="btn btn-default btn-xs"
onclick="showhide('{{ $i }}', 1)" style="display:;>
            <span class="glyphicon glyphicon-menu-right"></span>
        </button>
        <button id='span_down_{{ $i }}' type="button" class="btn btn-default btn-xs"
onclick="showhide('{{ $i }}', 0)" style="display: none;>
            <span class="glyphicon glyphicon-menu-down"></span>
        </button>
        <span class="glyphicon glyphicon-paperclip"></span>
        {{ $subjectRequirement->requirement->title }}
        <button type="button" class="btn btn-success pull-right" onclick="location.href='/classes/{{
$gradeSectionSubject->id }}/requirements/{{ $subjectRequirement->id }}/submission'">
            Check submission status
        </button>
        <br></br>
    </div>
    <div id="{{ $i }}" style="display:none;>
        <table class="table table-responsive">
            <tr>
                <td colspan="4">{!! $subjectRequirement->requirement->body !!}</td>
            </tr>
            <tr>
                <th>Students can upload from:</th><td>{{ $subjectRequirement-
>getUnformattedDate('event_start') }}</td>
                <th>Students can upload until:</th><td>{{ $subjectRequirement-
>getUnformattedDate('event_end') }}</td>
            </tr>
            <tr>
                <th>Published on:</th><td>{{ $subjectRequirement->getUnformattedDate('publish_on')
}}</td>
                <td colspan="2"></td>
            </tr>
        </table>
    </div>

```

```

</table>
<div class="well">
  @if (count($subjectRequirement->requirement->files) == 0)
    No files attached
  @else
    Files attached:
    <ul>
      @foreach ($subjectRequirement->requirement->files as $file)
        @include('layouts.file')
      @endforeach
    </ul>
  @endif
</div>
</div>
</div>

```

Resources\Views\Classes\Content\show.blade

```

@extends('layouts.app')

@section('title')
  {{ $gradeSectionSubject->subject->name }}
@endsection

@section('content')
<div class="container">
  @include('flash::message')
  <ol class="breadcrumb pull-right">
    <li><a href="/home">Home</a></li>
    <li class="active">{{ $gradeSectionSubject->subject->name }}</li>
  </ol>
  <br></br><hr/>
  <div class="row">
    <div class="col-md-10 col-md-offset-1">
      <h2>{{ $gradeSectionSubject->subject->name }}</h2>
      @if (empty($articles))
        <div class="well">
          Nothing posted yet.
        </div>
      @else
        <?php $i = 0 ?>
        @foreach ($articles as $article)
          @if (!strcmp($types[$i], 'P'))
            @include('classes.content.partials.post')
          @elseif (!strcmp($types[$i], 'R'))

```

```

        @include('classes.content.partials.requirement')
    @elseif (!strcmp($types[$i], 'E'))
        @include('classes.content.partials.examination')
    @elseif (!strcmp($types[$i], 'A'))
        @include('subjects.content.partials.admin-article')
    @endif
    <?php $i++ ?>
@endforeach
@endif
</div>
</div>
</div>
<script type="text/javascript">
function showhide(id, counter) {
    var e = document.getElementById(id);
    e.style.display = (e.style.display == 'block') ? 'none' : 'block';

    if(counter){
        var spanId = 'span_right_' + id;
        document.getElementById(spanId).style.display = 'none';
        var spanId2 = 'span_down_' + id;
        document.getElementById(spanId2).style.display = "";
    } else {
        var spanId = 'span_down_' + id;
        document.getElementById(spanId).style.display = 'none';
        var spanId2 = 'span_right_' + id;
        document.getElementById(spanId2).style.display = "";
    }
}
</script>
@stop

```

Resources\Views\Classes\Examinations\instance.blade

```

@extends('layouts.app')

@section('title')
    Deployment instance
@endsection

@section('content')
<div class="container">
    @include('flash::message')
    <ol class="breadcrumb pull-right">
        <li><a href="/home">Home</a></li>

```

```

    <li><a href="/classes/{{ $gradeSectionSubject->id }}">{{ $gradeSectionSubject->subject->name
}}</a></li>
    <li class="active">{{ $deployment->examination->description }}</li>
</ol>
<br></br><hr/>
<div class="row">
    <div class="col-md-10 col-md-offset-1">
        <div class="panel panel-default">
            <div class="panel-heading">
                Answer the examination
            </div>
            <table class="table table-hover">
                <tr>
                    <th style="width:50%" colspan="2">Title:</th><td style="width:50%" colspan="2">{{
$deployment->examination->description }}</td>
                </tr>
                <tr>
                    <th colspan="2">Created by:</th><td colspan="2">{{ $deployment->examination-
>faculty->getFullName() }}</td>
                </tr>
                <tr>
                    <th style="width:25%">Available from:</th><td style="width:25%">{{ $deployment-
>getUnformattedDate('exam_start') }}</td>
                    <th style="width:25%">Available until:</th><td style="width:25%">{{ $deployment-
>getUnformattedDate('exam_end') }}</td>
                </tr>
                <tr>
                    <td colspan="2"></td>
                    <td colspan="2">
                        {!! Form::open(['url' => '/classes/' . $gradeSectionSubject->id . '/deployments/' .
$deployment->id . '/instances']) !!}
                        <button type="submit" class="btn btn-primary">
                            Start examination
                        </button>
                        {!! Form::close() !!}
                    </td>
                </tr>
            </table>
        </div>
    </div>
</div>
</div>
</div>
@stop

```


Resources\Views\Classes\Examinations\results.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Examination results
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    <ol class="breadcrumb pull-right">
```

```
        <li><a href="/home">All classes</a></li>
```

```
        <li><a href="/classes/{{ $gradeSectionSubject->id }}">{{ $gradeSectionSubject->subject->name }}</a></li>
```

```
        <li class="active">Results of {{ $deployment->examination->description }}</li>
```

```
    </ol>
```

```
    <br></br><hr/>
```

```
    <div class="row">
```

```
        <div class="col-md-8 col-md-offset-2">
```

```
            <div class="panel panel-primary well">
```

```
                <div class="panel-heading">
```

```
                    <h5>Results of {{ $deployment->examination->description }}</h5>
```

```
                </div>
```

```
                <table class="table table-responsive table-hover">
```

```
                    <tr>
```

```
                        <th>Title:</th><td>{{ $deployment->examination->description }}</td>
```

```
                    </tr>
```

```
                    <tr>
```

```
                        <th>Score:</th>
```

```
                        <td>{{ $instance->score }} / {{ $deployment->examination->total_points }}</td>
```

```
                    </tr>
```

```
                    <tr>
```

```
                        <th>Time started:</th>
```

```
                        <td>{{ $instance->getUnformattedDate('time_started') }}</td>
```

```
                    </tr>
```

```
                    <tr>
```

```
                        <th>Time ended:</th>
```

```
                        <td>{{ $instance->getUnformattedDate('time_ended') }}</td>
```

```
                    </tr>
```

```
                </table>
```

```
            </div>
```

```
        </div>
```

```
    </div>
```

```
</div>
```

```
@endsection
```

Resources\Views\Classes\Requirements\instance.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Requirement submission status
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    @include('flash::message')
```

```
    <ol class="breadcrumb pull-right">
```

```
        <li><a href="/home">All classes</a></li>
```

```
        <li><a href="/classes/{{ $gradeSectionSubject->id }}">{{ $gradeSectionSubject->subject->name }}</a></li>
```

```
        <li class="active">{{ $subjectRequirement->requirement->title }} - Submission status</li>
```

```
    </ol>
```

```
    <br></br><hr/>
```

```
    <div class="row">
```

```
        <div class="col-md-10 col-md-offset-1">
```

```
            <div class="panel panel-default">
```

```
                <div class="panel-heading">
```

```
                    {{ $subjectRequirement->requirement->title }} - Submission status
```

```
                <div class="pull-right">
```

```
                    Status: {!! $status !!}
```

```
                </div>
```

```
            </div>
```

```
            <div class="panel panel-body">
```

```
                <table class="table table-responsive">
```

```
                    <tr>
```

```
                        <th>Available from:</th><td>{{ $subjectRequirement->getUnformattedDate('event_start') }}</td>
```

```
                        <th>Available until:</th><td>{{ $subjectRequirement->getUnformattedDate('event_end') }}</td>
```

```
                    </tr>
```

```
                </table>
```

```
            <hr/>
```

```
            <div class="well">
```

```
                @if ($instance)
```

```
                    Your submitted file:
```

```
                    @include('layouts.file')
```

```
                    <h4><label class="label label-primary">Submitted at: {{ $instance->submitted_at }}</label></h4>
```

```
                @else
```

```
                    No files submitted yet.
```

```

        @endif
    </div>
    @if ($ongoing)
        {!! Form::open(['url' => 'classes/' . $gradeSectionSubject->id . '/requirements/' .
$subjectRequirement->id . '/submission', 'files'=>true]) !!}
        <div class="form-group"{{ $errors->has('file') ? ' has-error' : " }}">
            {!! Form::label('file', 'Upload file', ['class' => 'col-md-4 control-label']) !!}
            <div class="col-md-6">
                {!! Form::file('file') !!}

                <h5><font color='red'>* 16mb file size limit. Only one file may be
uploaded.</font></h5>
                @if ($errors->has('file'))
                    <span class="help-block">
                        <strong>{{ $errors->first('file') }}</strong>
                    </span>
                @endif
            </div>
        </div>
    @endif
    @endif
    <div class="form-group pull-right">
        <div class="col-md-6 col-md-offset-4">
            <button type="button" class="btn btn-danger" onclick="location.href='/classes/{
$gradeSectionSubject->id }'">
                Back
            </button>
        </div>
    </div>
    @if ($ongoing)
    <div class="form-group pull-right">
        <div class="col-md-6 col-md-offset-4">
            {!! Form::submit('Submit', ['class' => 'btn btn-primary form-control']) !!}
        </div>
    </div>
    {!! Form::close() !!}
    @endif
</div>
</div>
</div>
</div>
@stop

```

Resources\Views\Dashboards\ faculty-home.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Home
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    @include('flash::message')
```

```
    <div class="row">
```

```
        <table class="table">
```

```
            <tr>
```

```
                <td style="width:33%">
```

```
                    <button type="button" class="btn btn-info form-control" onclick="location.href='/posts'">
```

```
                        <span class="glyphicon glyphicon-file" aria-hidden="true"></span> Posts
```

```
                    </button>
```

```
                </td>
```

```
                <td style="width:33%">
```

```
                    <button type="button" class="btn btn-success form-control"
```

```
onclick="location.href='/requirements'">
```

```
                        <span class="glyphicon glyphicon-paperclip" aria-hidden="true"></span> Requirements
```

```
                    </button>
```

```
                </td>
```

```
                <td style="width:33%">
```

```
                    <button type="button" class="btn btn-warning form-control"
```

```
onclick="location.href='/examinations'">
```

```
                        <span class="glyphicon glyphicon-list-alt" aria-hidden="true"></span> Examinations
```

```
                    </button>
```

```
                </td>
```

```
            </tr>
```

```
        </table>
```

```
    </div>
```

```
    <div class="row">
```

```
        <div class="col-md-10 col-md-offset-1">
```

```
            @if(count($gradeSectionSubjects) == 0)
```

```
                <div class="panel panel-default">
```

```
                    <div class="panel-body">Nothing to show.</div>
```

```
                </div>
```

```
            @else
```

```
                @foreach ($gradeSectionSubjects as $gradeSectionSubject)
```

```
                    <div class="panel panel-default">
```

```
                        <div class="panel-heading">
```

```
                            <a href="/subjects/{{ $gradeSectionSubject->id }}">
```


Resources\Views\Dashboards\system-administrator-home.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Home
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    @include('flash::message')
```

```
    <div class="row">
```

```
        <div class="col-md-10 col-md-offset-1">
```

```
            <div class="panel panel-primary">
```

```
                <div class="panel-heading">
```

```
                    Current schoolyear:
```

```
                    @if(is_null($activeSchoolYear))
```

```
                        <font color="orange">Undefined</font>
```

```
                    @else
```

```
                        <font color="yellow">{{ $activeSchoolYear->name }}</font>
```

```
                    @endif
```

```
                    <button type="button" class="btn btn-warning pull-right" onclick="">
```

```
                        Change schoolyear
```

```
                    </button>
```

```
                    <br></br>
```

```
                </div>
```

```
            </div>
```

```
            <div class="panel panel-default">
```

```
                <div class="panel-heading">
```

```
                    <button id='span_right_users_body' type="button" class="btn btn-default btn-xs"
onlick="showhide('users_body', 1)" style="display:; ">
```

```
                        <span class="glyphicon glyphicon-menu-right"></span>
```

```
                    </button>
```

```
                    <button id='span_down_users_body' type="button" class="btn btn-default btn-xs"
onlick="showhide('users_body', 0)" style="display: none; ">
```

```
                        <span class="glyphicon glyphicon-menu-down"></span>
```

```
                    </button>
```

```
                    Users
```

```
                </div>
```

```
                <div class="panel-body" id="users_body" style="display:none; ">
```

```
                    <ul>
```

```
                        <li><a href="/users/create">Create a new user</a></li>
```

```
                        <hr/>
```

```
                        <li><a href="/users/school-management">View list of school management</a></li>
```

```
                        <li><a href="/users/faculty">View list of faculty</a></li>
```

```

        <li><a href="/users/students">View list of students</a></li>
        <hr/>
        <li><a href="/users/enrollment">Assign students to a grade section</a></li>
    </ul>
</div>
</div>
<div class="panel panel-default">
    <div class="panel-heading">
        <button id='span_right_grade_sections_body' type="button" class="btn btn-default btn-xs"
onclick="showhide('grade_sections_body', 1)" style="display:;>
            <span class="glyphicon glyphicon-menu-right"></span>
        </button>
        <button id='span_down_grade_sections_body' type="button" class="btn btn-default btn-xs"
onclick="showhide('grade_sections_body', 0)" style="display: none; ">
            <span class="glyphicon glyphicon-menu-down"></span>
        </button>
        Grade sections
    </div>
    <div class="panel-body" id="grade_sections_body" style="display:none;">
        <ul>
            <li><a href="/grade-sections">View list of grade sections</a></li>
            <hr/>
            <li><a href="/users/enrollment">Assign students to a grade section</a></li>
            <li><a href="/grade-sections/subjects/assignment">Assign subjects to a grade
section</a></li>
        </ul>
    </div>
</div>
</div>
<div class="panel panel-default">
    <div class="panel-heading">
        <button id='span_right_faculty_loadings_body' type="button" class="btn btn-default btn-xs"
onclick="showhide('faculty_loadings_body', 1)" style="display:;>
            <span class="glyphicon glyphicon-menu-right"></span>
        </button>
        <button id='span_down_faculty_loadings_body' type="button" class="btn btn-default btn-xs"
onclick="showhide('faculty_loadings_body', 0)" style="display: none; ">
            <span class="glyphicon glyphicon-menu-down"></span>
        </button>
        Faculty Loadings
    </div>
    <div class="panel-body" id="faculty_loadings_body" style="display:none;">
        <ul>
            <li><a href="/faculty-loadings">View list of faculty loadings</a></li>
            <hr/>

```

```

        <li><a href="/faculty-loadings/assignment">Assign grade section subjects to a
faculty</a></li>
    </ul>
</div>
</div>
<div class="panel panel-default">
    <div class="panel-heading">
        <button id='span_right_others_body' type="button" class="btn btn-default btn-xs"
onclick="showhide('others_body', 1)" style="display:; ">
            <span class="glyphicon glyphicon-menu-right"></span>
        </button>
        <button id='span_down_others_body' type="button" class="btn btn-default btn-xs"
onclick="showhide('others_body', 0)" style="display: none; ">
            <span class="glyphicon glyphicon-menu-down"></span>
        </button>
        Others
    </div>
    <div class="panel-body" id="others_body" style="display:none; ">
        <ul>
            <li><a href="/others/subjects/create">Create a new subject</a></li>
            <li><a href="/others/subjects">View list of subjects</a></li>
            <hr/>
            <li><a href="/others/school-years/create">Create a new school year</a></li>
        </ul>
    </div>
</div>
</div>
<div class="panel panel-default">
    <div class="panel-heading">
        <button id='span_right_announcements_body' type="button" class="btn btn-default btn-xs"
onclick="showhide('announcements_body', 1)" style="display:; ">
            <span class="glyphicon glyphicon-menu-right"></span>
        </button>
        <button id='span_down_announcements_body' type="button" class="btn btn-default btn-xs"
onclick="showhide('announcements_body', 0)" style="display: none; ">
            <span class="glyphicon glyphicon-menu-down"></span>
        </button>
        Announcements
    </div>
    <div class="panel-body" id="announcements_body" style="display:none; ">
        <ul>
            <li><a href="#">Create a new announcement</a></li>
            <hr/>
            <li><a href="#">See list of announcements</a></li>
        </ul>
    </div>

```



```

        </div>
    </div>
</div>
</div>
</div>
<script type="text/javascript">
function showhide(id, counter) {
    var e = document.getElementById(id);
    e.style.display = (e.style.display == 'block') ? 'none' : 'block';

    if(counter){
        var spanId = 'span_right_' + id;
        document.getElementById(spanId).style.display = 'none';
        var spanId2 = 'span_down_' + id;
        document.getElementById(spanId2).style.display = "";
    } else {
        var spanId = 'span_down_' + id;
        document.getElementById(spanId).style.display = 'none';
        var spanId2 = 'span_right_' + id;
        document.getElementById(spanId2).style.display = "";
    }
}
</script>
@stop

```

Resources\Views\Errors\ 503.blade

```

<!DOCTYPE html>
<html>
    <head>
        <title>Be right back.</title>

        <link href="https://fonts.googleapis.com/css?family=Lato:100" rel="stylesheet"
type="text/css">

        <style>
            html, body {
                height: 100%;
            }

            body {
                margin: 0;
                padding: 0;

```

```

        width: 100%;
        color: #B0BEC5;
        display: table;
        font-weight: 100;
        font-family: 'Lato';
    }

    .container {
        text-align: center;
        display: table-cell;
        vertical-align: middle;
    }

    .content {
        text-align: center;
        display: inline-block;
    }

    .title {
        font-size: 72px;
        margin-bottom: 40px;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="content">
            <div class="title">Be right back.</div>
        </div>
    </div>
</body>
</html>

```

Resources\Views\Grade-Sections\ faculty-loadings\assignment.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Assign faculty loadings to a grade section
```

```
@endsection
```

```
@section('content')
```

```

<div class="container">
  <ol class="breadcrumb pull-right">
    <li><a href="/home">Home</a></li>
    <li class="active">Assignment</li>
  </ol>
  <br></br><hr/>
  <div class="row">
    <div class="col-md-8 col-md-offset-2">
      <script language="Javascript">
        function SelectMoveRows(SS1,SS2){
          var SelID="";
          var SelText="";
          // Move rows from SS1 to SS2 from bottom to top
          for (i=SS1.options.length - 1; i>=0; i--){
            if (SS1.options[i].selected == true){
              SelID=SS1.options[i].value;
              SelText=SS1.options[i].text;
              var newRow = new Option(SelText,SelID);
              SS2.options[SS2.length]=newRow;
              SS1.options[i]=null;
            }
          }
          SelectSort(SS2);
        }

        function SelectSort(SelList){
          var ID="";
          var someText="";
          for (x=0; x < SelList.length - 1; x++){
            for (y=x + 1; y < SelList.length; y++){
              if (SelList[x].text > SelList[y].text){
                // Swap rows
                ID=SelList[x].value;
                someText=SelList[x].text;
                SelList[x].value=SelList[y].value;
                SelList[x].text=SelList[y].text;
                SelList[y].value=ID;
                SelList[y].text=someText;
              }
            }
          }
        }

        function getSubjectList(){

```

```

var select = document.getElementById('FeatureCodes');
var result = [];
var options = select.options;
var opt;

for (var i=0, iLen=options.length; i<iLen; i++) {
    opt = options[i];
    result.push(opt.value || opt.text);
}
var hiddenDiv = document.getElementById('assignList');
hiddenDiv.value = result;
}
</script>
<table class="table">
<tr>
    {!! Form::open(['url' => '/faculty-loadings/assignment']) !!}
<td colspan="4">
    <select name="facultyId" class="form-control"
required>
        <option value="" selected disabled>Select a
faculty...</option>
        @foreach ($facultyList as $faculty)
            @if (!is_null($facultySelected) &&
$faculty->id == $facultySelected->id)
                <option value="{{ $faculty->id }}"
selected>{{ $faculty->getInfo()->lastname }}, {{ $faculty->getInfo()->firstname }}</option>
            @else
                <option value="{{ $faculty->id }}">{{
$faculty->getInfo()->lastname }}, {{ $faculty->getInfo()->firstname }}</option>
            @endif
        @endforeach
    </select>
</td>
<td>
    <button type="submit" class="btn btn-primary form-
control">
        Select faculty
    </button>
</td>
    {!! Form::close() !!}
</tr>
@if (isset($facultySelected))
    {!! Form::open(['url' => '/faculty-loadings/assignment/assign', 'name' =>
'Assignment']) !!}

```

```

<tr>
  <td colspan="2" style="width:40%">
    {!! Form::label('Features', 'All Grade Section Subjects:', ['class' => 'col-
md-4 control-label']) !!}
    <select name="Features" size="20" MULTIPLE class="form-control">
      @if (!empty($gradeSectionSubjects))
        @foreach ($gradeSectionSubjects as $gradeSectionSubject)
          <option value="{{ $gradeSectionSubject->id }}">{{
$gradeSectionSubject->subject->name }} ({{ $gradeSectionSubject->gradeSection->getName->name
}})</option>
        @endforeach
      @endif
    </select>
  </td>
  <td colspan="1" align="center">
    <br><br><br><br><br><br>
    <input type="Button" class="btn btn-default" value="Add >>"
style="width:100px"
onClick="SelectMoveRows(document.Assignment.Features,document.Assignment.FeatureCodes)" {{
isset($facultySelected) ? " : 'disabled' }}>
    <br><br>
    <input type="Button" class="btn btn-default" value="<< Remove"
style="width:100px"
onClick="SelectMoveRows(document.Assignment.FeatureCodes,document.Assignment.Features)" {{
isset($facultySelected) ? " : 'disabled' }}>
  </td>
  <td colspan="2" style="width:40%">
    {!! Form::label('FeatureCodes', 'Assigned Grade Section Subjects:',
[class' => 'col-md-4 control-label']) !!}
    <select name="FeatureCodes" id="FeatureCodes" size="20" MULTIPLE
class="form-control">
      @if (!empty($gradeSectionSubjectsAssignedToFaculty))
        @foreach ($gradeSectionSubjectsAssignedToFaculty as
$gradeSectionSubject)
          <option value="{{ $gradeSectionSubject->id }}">{{
$gradeSectionSubject->subject->name }} ({{ $gradeSectionSubject->gradeSection->getName->name
}})</option>
        @endforeach
      @endif
    </select>
  </td>
</tr>
<tr>
  <td colspan="5">

```

```

                <button type="submit" class="btn btn-primary form-control"
onclick="getSubjectList()" {{ isset($facultySelected) ? " : 'disabled' }}">
                    Update assignment status
                </button>
            </td>
        </tr>
    </tr>
    <tr>
        <input type="hidden" id="assignList" name="assignList" value="">
        <input type="hidden" id="assignFaculty" name="assignFaculty"
value="{{ $facultySelected->id or " }}">
    </tr>
    {!! Form::close() !!}
    @endif
</table>
</div>
</div>
</div>
@endsection

```

Resources\Views\Grade-Sections\ faculty-loadings\index.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Faculty list
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    @include('flash::message')
```

```
    <ol class="breadcrumb pull-right">
```

```
        <li><a href="/home">Home</a></li>
```

```
        <li class="active">Faculty list</li>
```

```
    </ol>
```

```
    <br></br><hr>
```

```
    <div class="row">
```

```
        <div class="col-md-10 col-md-offset-1">
```

```
            <div class="panel panel-primary">
```

```
                <div class="panel-heading">
```

```
                    Faculty list
```

```
                </div>
```

```
                @if ($facultyList->isEmpty())
```

```
                    <div class="panel-body">
```

```
                        No faculty found.
```

```
                    </div>
```

```

    @else
    <table class="table table-responsive table-bordered table-striped">
      <tr>
        <td></td>
        <th>Faculty name</th>
        <th>ID number</th>
      </tr>
      <?php $count = 1 ?>
      @foreach ($facultyList as $faculty)
      <tr>
        <td>{{ $count }}</td>
        <td><a href="faculty-loadings/{{ $faculty->id }}">{{ $faculty->getInfo()->lastname }} ,
        {{ $faculty->getInfo()->firstname }}</a></td>
        <td>{{ $faculty->getInfo()->idNumber }}</td>
      </tr>
      <?php $count++ ?>
      @endforeach
    </table>
  @endif
</div>
</div>
</div>
</div>
@stop

```

Resources\Views\Grade-Sections\ faculty-loadings\show.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Faculty loadings
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    @include('flash::message')
```

```
    <ol class="breadcrumb pull-right">
```

```
        <li><a href="/home">Home</a></li>
```

```
        <li><a href="/faculty-loadings">Faculty list</a></li>
```

```
        <li class="active">Faculty loadings</li>
```

```
    </ol>
```

```
    <br></br><hr/>
```

```
    <div class="row">
```

```
        <div class="col-md-10 col-md-offset-1">
```

```
            <div class="panel panel-primary">
```

```

<div class="panel-heading">
    Faculty loadings of {{ $user->getInfo()->lastname }}, {{ $user->getInfo()->firstname }}
</div>
@if ($facultyLoadings->isEmpty())
<div class="panel-body">
    No faculty loadings found.
</div>
@else
<table class="table table-responsive table-bordered table-striped">
    <tr>
        <td></td>
        <th>Subject name</th>
        <th>Section</th>
    </tr>
    <?php $count = 1 ?>
    @foreach ($facultyLoadings as $facultyLoading)
    <tr>
        <td>{{ $count }}</td>
        <td>{{ $facultyLoading->gradeSectionSubject->subject->name }}</td>
        <td>{{ $facultyLoading->gradeSectionSubject->gradeSection->getName->name }}</td>
    </tr>
    <?php $count++ ?>
    @endforeach
</table>
@endif
</div>
</div>
</div>
</div>
@stop

```

Resources\Views\Grade-Sections\ school-years\create.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Create a new school year
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    <div class="row">
```

```
        <div class="col-md-8 col-md-offset-2">
```

```
    <div class="panel panel-default">
```

```
        <div class="panel-heading">Create a new school year</div>
```



```

</li><a href="/home">Home</a></li>
<li class="active">Assignment</li>
</ol>
<br></br><hr/>
<div class="row">
  <div class="col-md-8 col-md-offset-2">
    <script language="Javascript">
      function SelectMoveRows(SS1,SS2){
        var SelID="";
        var SelText="";
        // Move rows from SS1 to SS2 from bottom to top
        for (i=SS1.options.length - 1; i>=0; i--){
          if (SS1.options[i].selected == true){
            SelID=SS1.options[i].value;
            SelText=SS1.options[i].text;
            var newRow = new Option(SelText,SelID);
            SS2.options[SS2.length]=newRow;
            SS1.options[i]=null;
          }
        }
        SelectSort(SS2);
      }

      function SelectSort(SelList){
        var ID="";
        var someText="";
        for (x=0; x < SelList.length - 1; x++){
          for (y=x + 1; y < SelList.length; y++){
            if (SelList[x].text > SelList[y].text){
              // Swap rows
              ID=SelList[x].value;
              someText=SelList[x].text;
              SelList[x].value=SelList[y].value;
              SelList[x].text=SelList[y].text;
              SelList[y].value=ID;
              SelList[y].text=someText;
            }
          }
        }
      }

      function getSubjectList(){
        var select = document.getElementById('FeatureCodes');
        var result = [];

```

```

        var options = select.options;
        var opt;

        for (var i=0, iLen=options.length; i<iLen; i++) {
            opt = options[i];
            result.push(opt.value || opt.text);
        }
        var hiddenDiv = document.getElementById('assignList');
        hiddenDiv.value = result;
    }
</script>
<table class="table">
    <tr>
        {!! Form::open(['url' => '/grade-sections/subjects/assignment'])
!!}
        <td colspan="4">
            <select name="gradeSectionId" class="form-control"
required>
                <option value="" selected disabled>Select a
grade section...</option>
                @foreach ($gradeSections as $gradeSection)
                    @if (!is_null($gradeSectionSelected)
&& $gradeSection->id == $gradeSectionSelected->id)
                        <option value="{{ $gradeSection->id
}}" selected>{{ $gradeSection->getName->getGradeLevel() }} - {{ $gradeSection->getName->name
}}</option>
                    @else
                        <option value="{{ $gradeSection->id
}}">{{ $gradeSection->getName->getGradeLevel() }} - {{ $gradeSection->getName->name
}}</option>
                    @endif
                @endforeach
            </select>
        </td>
        <td>
            <button type="submit" class="btn btn-primary form-
control">
                Select grade section
            </button>
        </td>
        {!! Form::close() !!}
    </tr>
    @if (isset($gradeSectionSelected))

```

```

        {!! Form::open(['url' => '/grade-sections/subjects/assignment/assign',
'name' => 'Assignment']) !!}
        <tr>
            <td colspan="2" style="width:40%">
                {!! Form::label('Features', 'All subjects:', ['class' => 'col-md-4 control-
label']) !!}

                <select name="Features" size="20" MULTIPLE class="form-control">
                    @foreach ($subjects as $subject)
                        @if (!in_array($subject->id, $subjectIdsAssignedInSection))
                            <option value="{{ $subject->id }}">{{ $subject->name }}</option>
                        @endif
                    @endforeach
                </select>
            </td>
            <td colspan="1" align="center">
                <br><br><br><br><br><br>
                <input type="Button" class="btn btn-default" value="Add >>"
style="width:100px"
onClick="SelectMoveRows(document.Assignment.Features,document.Assignment.FeatureCodes)" {{
isset($gradeSectionSelected) ? " : 'disabled' }}>
                <br><br>
                <input type="Button" class="btn btn-default" value="<< Remove"
style="width:100px"
onClick="SelectMoveRows(document.Assignment.FeatureCodes,document.Assignment.Features)" {{
isset($gradeSectionSelected) ? " : 'disabled' }}>
            </td>
            <td colspan="2" style="width:40%">
                {!! Form::label('FeatureCodes', 'Assigned subjects:', ['class' => 'col-md-4
control-label']) !!}

                <select name="FeatureCodes" id="FeatureCodes" size="20" MULTIPLE
class="form-control">

                    @if (!empty($subjectsAssignedInSection))
                        @foreach ($subjectsAssignedInSection as $subject)
                            <option value="{{ $subject->id }}">{{ $subject->name
}}</option>
                        @endforeach
                    @endif
                </select>
            </td>
        </tr>
        <tr>
            <td colspan="5">
                <button type="submit" class="btn btn-primary form-control"
onClick="getSubjectList()" {{ isset($gradeSectionSelected) ? " : 'disabled' }}>

```

```

                Update assignment status
            </button>
        </td>
    </tr>
</tr>
<tr>
    <input type="hidden" id="assignList" name="assignList" value="">
    <input type="hidden" id="assignSection" name="assignSection"
value="{{ $gradeSectionSelected->id or " }}">
</tr>
    {!! Form::close() !!}
    @endif
</table>
</div>
</div>
</div>
@endsection

```

Resources\Views\Grade-Sectionss\subjects\create.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Create a new subject
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    <div class="row">
```

```
        <div class="col-md-8 col-md-offset-2">
```

```
        <div class="panel panel-default">
```

```
            <div class="panel-heading">Create a new subject</div>
```

```
            <div class="panel-body">
```

```
                {!! Form::open(['url' => '/others/subjects']) !!}

```

```
                <div class="form-group">
```

```
                    {!! Form::label('name', 'Name:', ['class' => 'col-
md-4 control-label']) !!}

```

```
                    <div class="col-md-6">
```

```
                        {!! Form::text('name', null, ['class' =>
'form-control', 'required']) !!}

```

```
                    </div>
```

```
                </div>
```

```
            <div class="form-group pull-right">
```

```
                <div class="col-md-6 col-md-offset-4">
```

```

danger" onclick="location.href='/home'">
                                <button type="button" class="btn btn-
                                danger">
                                    Back
                                </button>
                            </div>
                        </div>
                    </div>
                <div class="form-group pull-right">
                    <div class="col-md-6 col-md-offset-4">
                        {!! Form::submit('Create subject',
                        ['class' => 'btn btn-primary form-control']) !!}
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
@endsection

```

Resources\Views\Grade-Sectionss\subjects\index.blade

```

@extends('layouts.app')

@section('title')
    Subjects
@endsection

@section('content')
<div class="container">
    @include('flash::message')
    <ol class="breadcrumb pull-right">
        <li><a href="/home">Home</a></li>
        <li class="active">Subjects</li>
    </ol>
    <br></br><hr/>
    <div class="row">
        <div class="col-md-10 col-md-offset-1">
            <div class="panel panel-default">
                <div class="panel-heading">
                    Subjects
                </div>
                @if ($subjects->isEmpty())
                <div class="panel-body">

```

```

        No subjects found.
    </div>
    @else
    <table class="table table-responsive table-bordered table-striped">
        <tr>
            <td></td>
            <th>Name</th>
        </tr>
        <?php $count = 1 ?>
        @foreach ($subjects as $subject)
        <tr>
            <td>{{ $count }}</td>
            <td>{{ $subject->name }}</td>
        </tr>
        <?php $count++ ?>
        @endforeach
    </table>
    @endif
</div>
</div>
</div>
</div>
</div>
@stop

```

Resources\Views\Grade-Sections\index.blade

```

@extends('layouts.app')

@section('title')
    Grade sections
@endsection

@section('content')
<div class="container">
    @include('flash::message')
    <ol class="breadcrumb pull-right">
        <li><a href="/home">Home</a></li>
        <li class="active">Grade sections</li>
    </ol>
    <br></br><hr/>
    <div class="row">
        <div class="col-md-10 col-md-offset-1">
            <div class="panel panel-default">
                <div class="panel-heading">
                    Grade sections

```

```

</div>
@if (empty($gradeSections))
<div class="panel-body">
    No grade sections found.
</div>
@else
<table class="table table-responsive table-bordered table-striped">
    <tr>
        <td></td>
        <th>Name</th>
        <th>Number of students</th>
        <th>School year</th>
    </tr>
    <?php $count = 1 ?>
    @foreach ($gradeSections as $gradeSection)
    <tr>
        <td>{{ $count }}</td>
        <td><a href="/grade-sections/{{ $gradeSection->id }}">{{ $gradeSection->getName->getGradeLevel() }} - {{ $gradeSection->getName->name }}</a></td>
        <td>{{ count($gradeSection->students) }}</td>
        <td>{{ $gradeSection->schoolYear->name }}</td>
    </tr>
    <?php $count++ ?>
    @endforeach
</table>
@endif
</div>
</div>
</div>
</div>
</div>
@stop

```

Resources\Views\Grade-Sections\show.blade

```

@extends('layouts.app')

@section('title')
    {{ $gradeSection->getName->getGradeLevel() }} - {{ $gradeSection->getName->name }}
@endsection

@section('content')
<div class="container">
    @include('flash::message')
    <ol class="breadcrumb pull-right">
        <li><a href="/home">Home</a></li>

```



```

<li><a href="/grade-sections">Grade sections</a></li>
<li class="active">{{ $gradeSection->getName->getGradeLevel() }} - {{ $gradeSection->getName-
>name }}</li>
</ol>
<br></br><hr/>
<div class="row">
  <div class="col-md-10 col-md-offset-1">
    <div class="panel panel-primary">
      <div class="panel-heading">
        Details
      </div>
      <table class="table">
        <tr>
          <th style="width:50%">Name</th>
          <td style="width:50%">{{ $gradeSection->getName->getGradeLevel() }} - {{
$gradeSection->getName->name }}</td>
        </tr>
        <tr>
          <th style="width:50%">School year</th>
          <td style="width:50%">{{ $gradeSection->schoolYear->name }}</td>
        </tr>
      </table>
    </div>
    <div class="panel panel-primary">
      <div class="panel-heading">
        Students
      </div>
      <@if ($gradeSection->students->isEmpty())>
      <div class="panel-body">
        No students found. Click <a href="/users/enrollment">here</a> to start enrolling students.
      </div>
      <@else>
      <table class="table table-responsive table-bordered table-striped">
        <tr>
          <td></td>
          <th>Username</th>
          <th>Lastname</th>
          <th>Firstname</th>
        </tr>
        <?php $count = 1 ?>
        <@foreach ($gradeSection->students as $student)>
        <tr>
          <td>{{ $count }}</td>
          <td>{{ $student->username }}</td>

```

```

        <td>{{ $student->getInfo()->lastname }}</td>
        <td>{{ $student->getInfo()->firstname }}</td>
    </tr>
    <?php $count++ ?>
    @endforeach
</table>
@endif
</div>
<div class="panel panel-primary">
    <div class="panel-heading">
        Subjects
    </div>
    @if ($gradeSection->subjects->isEmpty())
    <div class="panel-body">
        No subjects found. Click <a href="/grade-sections/subjects/assignment">here</a> to start
        assigning subjects.
    </div>
    @else
    <table class="table table-responsive table-bordered table-striped">
        <tr>
            <td></td>
            <th>Name</th>
            <th>Faculty</th>
        </tr>
        <?php $count = 1 ?>
        @foreach ($gradeSection->subjects as $subject)
        <tr>
            <td>{{ $count }}</td>
            <td>{{ $subject->name }}</td>
            <?php
                $faculty = $gradeSection->getGradeSectionSubject($subject)->faculty->first();
            ?>
            @if (is_null($faculty))
            <td class="bg-danger">No faculty assigned yet.</td>
            @else
            <td>
                {{ $faculty->getInfo()->lastname }}, {{ $faculty->getInfo()->firstname }}
            </td>
            @endif
        </tr>
        <?php $count++ ?>
    @endforeach
</table>
@endif

```

```

        </div>
    </div>
</div>
</div>
@stop

```

Resources\Views\Grade-Section-Subjects\content\partials\admin-article.blade

```

<div class="panel panel-default">
    <div class="panel-heading">
        {{ $article->title }}
        <div class="pull-right">
            Published at: {{ $article->getUnformattedDate('publish_on') }}
        </div>
    </div>
    <div class="panel-body">
        {!! $article->body !!}
    </div>
</div>

```

Resources\Views\Grade-Section-Subjects\content\partials\examination.blade

```

<?php $deployment = $article ?>
<div class="panel panel-warning">
    <div class="panel-heading">
        <button id='span_right_{{ $i }}' type="button" class="btn btn-default btn-xs"
onclick="showhide('{{ $i }}', 1)" style="display:;>
            <span class="glyphicon glyphicon-menu-right"></span>
        </button>
        <button id='span_down_{{ $i }}' type="button" class="btn btn-default btn-xs"
onclick="showhide('{{ $i }}', 0)" style="display: none;>
            <span class="glyphicon glyphicon-menu-down"></span>
        </button>
        <span class="glyphicon glyphicon-list-alt"></span>
        {{ $deployment->examination->description }}
        <button type="button" class="btn btn-warning pull-right" onclick="location.href='#">
            Check examination status of students
        </button>
        <br></br>
    </div>
    <div id="{{ $i }}" style="display:none;>
        <table class="table table-responsive">
            <tr>
                <th colspan="2">Subcategory:</th><td colspan="2">{{ $deployment->examination-
>subcategory }}</td>
            </tr>

```

```

        <tr>
            <th colspan="2">Total points:</th><td colspan="2">{{ $deployment->examination-
>computeTotalPoints() }}</td>
        </tr>
        <tr>
            <th>Exam start:</th><td>{{ $deployment->getUnformattedDate('exam_start') }}</td>
            <th>Exam end:</th><td>{{ $deployment->getUnformattedDate('exam_end') }}</td>
        </tr>
        <tr>
            <th>Published on:</th><td>{{ $deployment->getUnformattedDate('publish_on') }}</td>
            <td colspan="2"></td>
        </tr>
    </table>
</div>
</div>

```

Resources\Views\Grade-Section-Subjects\content\partials\post.blade

```

<?php $subjectPost = $article ?>
<div class="panel panel-info">
    <div class="panel-heading">
        <button id='span_right_{{ $i }}' type="button" class="btn btn-default btn-xs"
onclick="showhide('{{ $i }}', 1)" style="display:; ">
            <span class="glyphicon glyphicon-menu-right"></span>
        </button>
        <button id='span_down_{{ $i }}' type="button" class="btn btn-default btn-xs"
onclick="showhide('{{ $i }}', 0)" style="display: none; ">
            <span class="glyphicon glyphicon-menu-down"></span>
        </button>
        <span class="glyphicon glyphicon-file"></span>
        {{ $subjectPost->post->title }}
    </div>
    <div id="{{ $i }}" style="display:none; ">
        <table class="table table-responsive">
            <tr>
                <td colspan="2">{!! $subjectPost->post->body !!</td>
            </tr>
            <tr>
                <th>Published on:</th><td>{{ $subjectPost->getUnformattedDate('publish_on') }}</td>
            </tr>
        </table>
    <div class="well">
        @if (count($subjectPost->post->files) == 0)
            No files attached
        @else

```

Files attached:

```
<ul>
  @foreach ($subjectPost->post->files as $file)
    @include('layouts.file')
  @endforeach
</ul>
@endif
</div>
</div>
</div>
```

Resources\Views\Grade-Section-Subjects\content\partials\requirement.blade

```
<?php $subjectRequirement = $article ?>
<div class="panel panel-success">
  <div class="panel-heading">
    <button id='span_right_{{ $i }}' type="button" class="btn btn-default btn-xs"
onclick="showhide('{{ $i }}', 1)" style="display:; ">
      <span class="glyphicon glyphicon-menu-right"></span>
    </button>
    <button id='span_down_{{ $i }}' type="button" class="btn btn-default btn-xs"
onclick="showhide('{{ $i }}', 0)" style="display: none; ">
      <span class="glyphicon glyphicon-menu-down"></span>
    </button>
    <span class="glyphicon glyphicon-paperclip"></span>
    {{ $subjectRequirement->requirement->title }}
    <button type="button" class="btn btn-success pull-right" onclick="location.href='/subjects/{{
$gradeSectionSubject->id }}/requirements/{{ $subjectRequirement->id }}/submissions'">
      Check submission status of students
    </button>
    <br></br>
  </div>
  <div id="{{ $i }}" style="display:none;">
    <table class="table table-responsive">
      <tr>
        <td colspan="4">{!! $subjectRequirement->requirement->body !!</td>
      </tr>
      <tr>
        <th>Students can upload from:</th><td>{{ $subjectRequirement-
>getUnformattedDate('event_start') }}</td>
        <th>Students can upload until:</th><td>{{ $subjectRequirement-
>getUnformattedDate('event_end') }}</td>
      </tr>
      <tr>
```

```

        <th>Published on:</th><td>{{ $subjectRequirement->getUnformattedDate('publish_on')
}}</td>
        <td colspan="2"></td>
    </tr>
</table>
<div class="well">
    @if (count($subjectRequirement->requirement->files) == 0)
        No files attached
    @else
        Files attached:
        <ul>
            @foreach ($subjectRequirement->requirement->files as $file)
                @include('layouts.file')
            @endforeach
        </ul>
    @endif
</div>
</div>
</div>

```

Resources\Views\Grade-Section-Subjects\content\class-record.blade

```

@extends('layouts.app')

@section('title')
    View class record
@endsection

@section('content')
<div class="container">
    <ol class="breadcrumb pull-right">
        <li><a href="/home">All subjects</a></li>
        <li><a href="/subjects/{{ $gradeSectionSubject->id }}">{{ $gradeSectionSubject->subject->name
}} ({{ $gradeSectionSubject->gradeSection->getName->name }})</a></li>
        <li class="active">Class record</a></li>
    </ol>
    <br></br><hr/>
    <div class="row">
        <table class="table table-responsive table-hover table-bordered">
            <tr>
                <th style="width:20%"></th>
                <th colspan="{{ $quizCount }}">Quizes</th>
                <th colspan="{{ $longTestCount }}">Long test</th>
                <th colspan="{{ $othersCount }}">Others</th>
            </tr>

```

```

<tr>
  <td style="width:20%"></td>

  @if ($quizCount == 0)
    <td></td>
  @endif
  @foreach ($quizClassRecords as $quizClassRecord)
    <td>
      <div class="dropdown">
        <button class="btn btn-default btn-xs dropdown-toggle" id="dLabel"
data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">
          <span class="caret"></span>
        </button>
        <ul class="dropdown-menu" aria-labelledby="dLabel">
          <li>Description: {{ $quizClassRecord->examination-
>description }}</li>
          <li class="divider"></li>
          <li>Date deployed: {{ $quizClassRecord->examination-
>deployments()->where('grade_section_subject_id', $gradeSectionSubject->id)->first()->publish_on
}}</li>
          <li class="divider"></li>
          <li>Total points: {{ $quizClassRecord->examination-
>total_points }}</li>
        </ul>
      </div>
    </td>
  @endforeach

  @if ($longTestCount == 0)
    <td></td>
  @endif
  @foreach ($longTestClassRecords as $longTestClassRecord)
    <td>
      <div class="dropdown">
        <button class="btn btn-default btn-xs dropdown-toggle" id="dLabel"
data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">
          <span class="caret"></span>
        </button>
        <ul class="dropdown-menu" aria-labelledby="dLabel">
          <li>Description: {{ $longTestClassRecord->examination-
>description }}</li>
          <li class="divider"></li>
          <li>Date deployed: {{ $longTestClassRecord->examination-
>deployments()->where('grade_section_subject_id', $gradeSectionSubject->id)->publish_on }}</li>

```

```

        <li class="divider"></li>
        <li>Total points: {{ $longTestClassRecord->examination-
>total_points }}</li>
    </ul>
</div>
</td>
@endforeach

@if ($othersCount == 0)
    <td></td>
@endif
@foreach ($othersClassRecords as $othersClassRecord)
    <td>
        <div class="dropdown">
            <button class="btn btn-default btn-xs dropdown-toggle" id="dLabel"
data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">
                <span class="caret"></span>
            </button>
            <ul class="dropdown-menu" aria-labelledby="dLabel">
                <li>Description: {{ $othersClassRecord->examination-
>description }}</li>
                <li class="divider"></li>
                <li>Date deployed: {{ $othersClassRecord->examination-
>deployments()->where('grade_section_subject_id', $gradeSectionSubject->id)->publish_on }}</li>
                <li class="divider"></li>
                <li>Total points: {{ $othersClassRecord->examination-
>total_points }}</li>
            </ul>
        </div>
    </td>
@endforeach
</tr>

@foreach ($students as $student)
<tr>
    <td>{{ $student->getInfo()->lastname }}, {{ $student->getInfo()-
>firstname }} <br>
        ({{ $student->getInfo()->idNumber }})
    </td>

    @if ($quizCount == 0)
        <td></td>
    @endif
    @foreach ($quizClassRecords as $quizClassRecord)

```



```

        <td><?php $record = $quizClassRecord->instances()->where('user_id',
$student->id)->first();
                echo (($record == null) ? " : $record->score);
?></td>
@endforeach

@if ($longTestCount == 0)
        <td></td>
@endif
@foreach ($longTestClassRecords as $longTestClassRecord)
        <td><?php $record = $longTestClassRecord->instances()-
>where('user_id', $student->id)->first();
                echo (($record == null) ? " : $record->score);
?></td>
@endforeach

@if ($othersCount == 0)
        <td></td>
@endif
@foreach ($othersClassRecords as $othersClassRecord)
        <td><?php $record = $othersClassRecord->instances()->where('user_id',
$student->id)->first();
                echo (($record == null) ? " : $record->score);
?></td>
@endforeach
</tr>
@endforeach
</table>
</div>
</div>
@endsection

```

Resources\Views\Grade-Section-Subjects\content\details.blade

```

@extends('layouts.app')

@section('title')
    {{ $gradeSectionSubject->subject->name }}
@endsection

@section('content')
<div class="container">
    <ol class="breadcrumb pull-right">
        <li><a href="/home">Home</a></li>

```

```

    <li><a href="/subjects/{ { $gradeSectionSubject->id } }">{ { $gradeSectionSubject->subject->name
}} ( { { $gradeSectionSubject->gradeSection->getName->name } } )</a></li>
    <li class="active">Details</li>
</ol>
<br></br><hr/>
<div class="row">
    <div class="col-md-10 col-md-offset-1">
        <div class="panel panel-primary">
            <div class="panel-heading">
                Details
            </div>
            <table class="table table-responsive">
                <tr>
                    <th style="width:35%">Title:</th>
                    <td>{ { $gradeSectionSubject->subject->name } }</td>
                </tr>
                <tr>
                    <th>Section:</th>
                    <td>{ { $gradeSectionSubject->gradeSection->getName->name } }</td>
                </tr>
                <tr>
                    <th>Number of students:</th>
                    <td>{ { count($gradeSectionSubject->gradeSection->students) } }</td>
                </tr>
            </table>
        </div>
        <div class="panel panel-primary">
            <div class="panel-heading">
                Students enrolled
            </div>
            @if (count($gradeSectionSubject->gradeSection->students) == 0)
            <div class="panel-body">
                No students enrolled.
            </div>
            @else
            <table class="table">
                <tr>
                    <td></td>
                    <th>Last name</th>
                    <th>First name</th>
                    <th>ID number</th>
                </tr>
                <?php $count = 1 ?>
                @foreach ($gradeSectionSubject->gradeSection->students as $student)

```

```

        <tr>
            <td>{{ $count++ }}</td>
            <td>{{ $student->getInfo()->lastname }}</td>
            <td>{{ $student->getInfo()->firstname }}</td>
            <td>{{ $student->getInfo()->idNumber }}</td>
        </tr>
    @endforeach
</table>
    @endif
</div>
</div>
</div>
</div>
@stop

```

Resources\Views\Grade-Section-Subjects\content\show.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    {{ $gradeSectionSubject->subject->name }} ({{ $gradeSectionSubject->gradeSection->getName->name }})
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    @include('flash::message')
```

```
    <ol class="breadcrumb pull-right">
```

```
        <li><a href="/home">Home</a></li>
```

```
        <li class="active">{{ $gradeSectionSubject->subject->name }} ({{ $gradeSectionSubject->gradeSection->getName->name }})</li>
```

```
    </ol>
```

```
    <br></br><hr/>
```

```
    <div class="row">
```

```
        <div class="col-md-10 col-md-offset-1">
```

```
            <font size='6'>{{ $gradeSectionSubject->subject->name }} ({{ $gradeSectionSubject->gradeSection->getName->name }})</font>
```

```
            <div class="btn-group pull-right">
```

```
                <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
```

```
                    <span class="glyphicon glyphicon-menu-hamburger" aria-hidden="true"></span> Menu  
                    <span class="caret"></span>
```

```
                </button>
```

```
                <ul class="dropdown-menu">
```

```

        <li><a href="/subjects/{ { $gradeSectionSubject->id } }/class-record">View class
record</a></li>
        <li class="divider">
        <li><a href="/subjects/{ { $gradeSectionSubject->id } }/details">View details</a></li>
    </ul>
</div>
<br></br>
<hr/>
@if (empty($articles))
    <div class="well">
        Nothing posted yet.
    </div>
@else
    <?php $i = 0 ?>
    @foreach ($articles as $article)
        @if (!strcmp($types[$i], 'P'))
            @include('grade-section-subjects.content.partials.post')
        @elseif (!strcmp($types[$i], 'R'))
            @include('grade-section-subjects.content.partials.requirement')
        @elseif (!strcmp($types[$i], 'E'))
            @include('grade-section-subjects.content.partials.examination')
        @elseif (!strcmp($types[$i], 'A'))
            @include('grade-section-subjects.content.partials.admin-article')
        @endif
        <?php $i++ ?>
    @endforeach
@endif
</div>
</div>
</div>
<script type="text/javascript">
function showhide(id, counter) {
    var e = document.getElementById(id);
    e.style.display = (e.style.display == 'block') ? 'none' : 'block';

    if(counter){
        var spanId = 'span_right_' + id;
        document.getElementById(spanId).style.display = 'none';
        var spanId2 = 'span_down_' + id;
        document.getElementById(spanId2).style.display = "";
    } else {
        var spanId = 'span_down_' + id;
        document.getElementById(spanId).style.display = 'none';
        var spanId2 = 'span_right_' + id;

```

```

        document.getElementById(spanId2).style.display = "";
    }
}
</script>
@stop

```

Resources\Views\Grade-Section-Subjects\examinations\content\ create.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Create new examination header
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    <div class="row">
```

```
        <div class="col-md-8 col-md-offset-2">
```

```
        <div class="panel panel-default">
```

```
            <div class="panel-heading">Create a new examination header</div>
```

```
                <div class="panel-body">
```

```
                    {!! Form::model($examination = new
```

```
\App\Models\Examinations\Examination, ['url' => '/examinations']) !!}

```

```
                    @include('grade-section-
```

```
subjects.examinations.content.form', ['submitButtonText' => 'Create examination header'])

```

```
                    {!! Form::close() !!}

```

```
                </div>

```

```
            </div>

```

```
        </div>

```

```
    </div>

```

```
</div>
```

```
@endsection
```

Resources\Views\Grade-Section-Subjects\examinations\content\edit.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Edit examination header
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    <div class="row">
```

```
        <div class="col-md-8 col-md-offset-2">
```

```
        <div class="panel panel-default">
```

```

    <div class="panel-heading">Edit examination header</div>
        <div class="panel-body">
            {!! Form::model($examination, ['method' => 'PATCH', 'url' =>
'/examinations/' . $examination->id]) !!}
                @include('grade-section-
subjects.examinations.content.form', ['submitButtonText' => 'Edit examination header'])
                {!! Form::close() !!}
            </div>
        </div>
    </div>
</div>
@endsection

```

Resources\Views\Grade-Section-Subjects\examinations\content\form.blade

```

<div class="form-group">
    {!! Form::label('subcategory', 'Subcategory', ['class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
        {!! Form::select('subcategory', $subcategories, null, ['class' => 'form-control', 'placeholder' => 'Select
from the following...', 'required']) !!}
    </div>
</div>

```

```

<div class="form-group">
    {!! Form::label('description', 'Description:', ['class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
        {!! Form::text('description', null, ['class' => 'form-control', 'required', 'maxlength' =>
'255']) !!}
    </div>
</div>

```

```

<div class="form-group">
    {!! Form::label('is_private', 'Is this examination private?', ['class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
        <label class="radio-inline">
            {!! Form::radio('is_private', '1', ['required']) !!} True
        </label>
        <label class="radio-inline">
            {!! Form::radio('is_private', '0') !!} False
        </label>
    </div>
</div>

```

```

<div class="form-group pull-right">
    <div class="col-md-6 col-md-offset-4">
        <button type="button" class="btn btn-danger" onclick="location.href='/examinations'">
            Back
        </button>
    </div>
</div>

<div class="form-group pull-right">
    <div class="col-md-6 col-md-offset-4">
        {!! Form::submit($submitButtonText, ['class' => 'btn btn-primary form-control']) !!}
    </div>
</div>

```

Resources\Views\Grade-Section-Subjects\examinations\content\index.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Examinations
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    @include('flash::message')
```

```
    <ol class="breadcrumb pull-right">
```

```
        <li><a href="/home">Home</a></li>
```

```
        <li class="active">Examinations</li>
```

```
    </ol>
```

```
    <br></br><hr/>
```

```
    <font size='6'>Examinations</font>
```

```
    <div class="btn-group pull-right">
```

```
        <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown" aria-
haspopup="true" aria-expanded="false">
```

```
            <span class="glyphicon glyphicon-menu-hamburger" aria-hidden="true"></span> Menu <span
class="caret"></span>
```

```
        </button>
```

```
        <ul class="dropdown-menu">
```

```
            <li><a href="/examinations/create">
```

```
                <span class="glyphicon glyphicon-list-alt"></span> New examination
```

```
            </a></li>
```

```
            <li class="divider">
```

```
            <li><a class="bg-info" href="/categories">View all categories</a></li>
```

```
        </ul>
```

```
    </div>
```

```

<br></br>
<div class="row">
  @if ($examinations->isEmpty())
  <div class="col-md-8 col-md-offset-2 well">
    No examinations found.
  </div>
  @else
  <div class="col-md-10 col-md-offset-1">
    @foreach ($examinations as $examination)
    <div class="panel panel-default">
      <div class="panel-heading">
        <a href="/examinations/{{ $examination->id }}">{{ $examination->description }}</a>
        <div class="dropdown pull-right">
          <button class="btn btn-default btn-xs dropdown-toggle" id="dLabel" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">
            <span class="caret"></span>
          </button>
          <ul class="dropdown-menu" aria-labelledby="dLabel">
            <li><a href="/examinations/{{ $examination->id }}/edit">Edit examination
details</a></li>
            <li class="divider"></li>
            <li><a href="/examinations/{{ $examination->id }}/attach">Publish exam and attach to
grade section subjects</a></li>
          </ul>
        </div>
      </div>
    </div>
    @endforeach
  </div>
  @endif
</div>
@stop

```

Resources\Views\Grade-Section-Subjects\examinations\content\instance.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Examination instance
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    @include('flash::message')
```



```

<ol class="breadcrumb pull-right">
  <li><a href="/home">Home</a></li>
  <li><a href="/examinations">All examinations</a></li>
  <li><a href="/examinations/{ { $examination->id } }">{ { $examination->description } }</a></li>
  <li class="active">Examination instance</li>
</ol>
<br></br><hr/>
<div class="row">
  <div class="col-md-10 col-md-offset-1">
    <div class="panel panel-default">
      <div class="panel-heading">
        Answer a sample generated examination
      </div>
      <table class="table table-striped">
        <tr>
          <th style="width:50%" colspan="2">Subcategory:</th><td style="width:50%"
colspan="2">{ { $examination->subcategory } }</td>
        </tr>
        <tr>
          <th colspan="2">Description:</th><td colspan="2">{ { $examination->description
} }</td>
        </tr>
        <tr>
          <th colspan="2">Total points</th><td colspan="2">{ { $examination-
>computeTotalPoints() } }</td>
        </tr>
        <tr>
          <th colspan="2">Created by:</th><td colspan="2">{ { $examination->faculty->getInfo()-
>lastname } }, { { $examination->faculty->getInfo()->firstname } }</td>
        </tr>
        <tr>
          <td style="width:75%" colspan="3"></td>
          <td style="width:25%">
            {!! Form::open(['url' => '/examinations/' . $examination->id . '/faculty/instances']) !!}
            <button type="submit" class="btn btn-primary" {{ (count($examination->parts) == 0)
? 'disabled' : " }}>
              Start examination
            </button>
            {!! Form::close() !!}
          </td>
        </tr>
      </table>
    </div>
  </div>
</div>

```

```
</div>
</div>
@stop
```

Resources\Views\Grade-Section-Subjects\examinations\content\show.blade

```
@extends('layouts.app')
```

```
@section('title')
    {{ $examination->description }}
@endsection
```

```
@section('content')
<div class="container">
    @include('flash::message')
    <ol class="breadcrumb pull-right">
        <li><a href="/home">Home</a></li>
        <li><a href="/examinations">Examinations</a></li>
        <li class="active">{{ $examination->description }}</li>
    </ol>
    <br></br><hr/>
    <div class="btn-group pull-right">
        <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown" aria-
haspopup="true" aria-expanded="false">
            <span class="glyphicon glyphicon-menu-hamburger" aria-hidden="true"></span> Menu <span
class="caret"></span>
        </button>
        <ul class="dropdown-menu">
            <li><a href="/examinations/{{ $examination->id }}/parts">View all examination parts</a></li>
            <li class="divider">
            <li><a href="/examinations/{{ $examination->id }}/edit">Edit examination details</a></li>
            <li class="divider">
            <li><a href="/examinations/{{ $examination->id }}/instances">Generate sample exam</a></li>
            <li class="divider">
            <li><a href="/examinations/{{ $examination->id }}/assignment">Create examination
deployments</a></li>
            <li class="divider">
            <li><a class="bg-info" href="/categories">View all categories</a></li>
        </ul>
    </div>
<br></br>
<div class="row">
    <div class="col-md-10 col-md-offset-1">
        <div class="panel panel-primary">
            <div class="panel-heading">
```

```

    Examination details
</div>
<table class="table table-striped table-hover">
  <tr>
    <th>Subcategory:</th><td>{{ $examination->subcategory }}</td>
  </tr>
  <tr>
    <th>Description:</th><td>{{ $examination->description }}</td>
  </tr>
  <tr>
    <th>Total points</th><td>{{ $examination->computeTotalPoints() }}</td>
  </tr>
  <tr>
    <th>Created at:</th><td>{{ $examination->created_at }}</td>
  </tr>
</table>
</div>
@if($attachments->isEmpty())
<div class="well">
  Not attached to any grade section subject yet.
</div>
@else
<div class="panel panel-primary">
  <div class="panel-heading">
    Attached to the following grade section subjects
  </div>
  <table class="table table-striped">
    <tr>
      <td></td>
      <th>Subject name</th>
      <th>Section name</th>
      <th>Publish on</th>
    </tr>
    <?php $count = 1 ?>
    @foreach ($attachments as $attachment)
      <tr>
        <td>{{ $count++ }}</td>
        <td>{{ $attachment->gradeSectionSubject->subject->name }}</td>
        <td>{{ $attachment->gradeSectionSubject->gradeSection->getName->name }}</td>
        <td>{{ $attachment->getUnformattedDate('publish_on') }}</td>
      </tr>
    @endforeach
  </table>
</div>

```

```

        @endif
    </div>
</div>
@stop

```

Resources\Views\Grade-Section-Subjects\examinations\instances\fill-in-the-blanks.blade

```

@extends('grade-section-subjects.examinations.layout')

@section('instance-form')
    @foreach ($questions as $question)
        <div class="panel panel-default">
            <div class="panel-heading">{!! $question->body !!</div>
            <table class="table table-responsive">
                <tr>
                    <td style="width:100%">
                        {!! Form::text('answers[' . $question->id . ']', $answers[$question->id],
[ 'class' => 'form-control', 'required' ]) !!}
                    </td>
                </tr>
            </table>
        </div>
    @endforeach

    <div class="form-group">
        <div class="col-md-6 col-md-offset-4">
            {!! Form::submit('Submit answers', [ 'class' => 'btn btn-primary form-control' ]) !!}
        </div>
    </div>
@endsection

```

Resources\Views\Grade-Section-Subjects\examinations\instances\finish.blade

```

@extends('layouts.app')

```

```

@section('title')
    Finish examination
@endsection

```

```

@section('content')
<div class="container">
    {!! $navbar !!}
    <hr/>
    <div class="row">

```

```

<div class="col-md-8 col-md-offset-2">
  <div class="panel panel-primary">
    <div class="panel-heading">
      <h5>Are you sure you want to finish your examination and get your results?</h5>
    </div>
    <table class="table">
      <tr>
        <td></td>
        <td style="width:35%">
          {!! Form::open(['url' => $nextUrl ] !!}
            {!! Form::submit('Finish', ['class' => 'btn btn-success form-control']) !!}
            {!! Form::close() !!}
          </td>
        <td style="width:35%">
          <button type="button" class="btn btn-default form-control" onclick="location.href='{{
$startUrl }}'">
            Go back to start
          </button>
        </td>
      </tr>
    </table>
  </div>
</div>
</div>
</div>
@endsection

```

Resources\Views\Grade-Section-Subjects\examinations\instances\layout.blade

```

@extends('layouts.app')

@section('title')
  @if ($instancePart->instance->deployment == null)
    Sample generated examination
  @else
    {{ $instancePart->instance->deployment->examination->description }}
  @endif
@endsection

@section('content')
<div class="container">
  {!! $navbar !!}
  @include('flash::message')
  <hr/>
  <div class="row">

```

```

    <div class="col-md-8 col-md-offset-2">
        <div class="panel panel-primary">
            <div class="panel-heading">
                {{ $instancePart->examinationPart->getQuestionType() }}:
                @if (!strcmp($instancePart->examinationPart-
>getQuestionType(), 'Multiple Choice'))
                    Select the correct answer from the following.
                @elseif (!strcmp($instancePart->examinationPart-
>getQuestionType(), 'True or False'))
                    Choose whether or not the given statement is true.
                @elseif (!strcmp($instancePart->examinationPart-
>getQuestionType(), 'Fill in the Blanks'))
                    Identify what is being asked.
                @elseif (!strcmp($instancePart->examinationPart-
>getQuestionType(), 'Match Column A with Column B'))
                    Match the correct answer to each corresponding item.
                @elseif (!strcmp($instancePart->examinationPart-
>getQuestionType(), 'Select from the Wordbox'))
                    Select from the wordbox which answer is most
appropriate.
                @endif
            </div>
        </div>
        <div>
            {!! Form::open(['url' => $submitUrl]) !!}
                @yield('instance-form')
            {!! Form::close() !!}
        </div>
    </div>
</div>
@endsection

```

Resources\Views\Grade-Section-Subjects\examinations\instances\match-column-a-with-column-b.blade

```

@extends('grade-section-subjects.examinations.instances.layout')

@section('instance-form')
<div class="panel panel-default">
    <div class="panel-heading">{!! $question->body !!</div>
    <div class="panel-body">
        <table class="table table-responsive">
            <tr>
                <td style="width:65%">
                    <ul class="list-group">

```

```

        @foreach ($items as $item)
            <li class="list-group-item">
                <div class="form-group" id="item_div_{{ $item->id }}"
onclick="setToActive({{ $item->id }})">
                    <div class="input-group">
                        <span class="input-group-
addon">{{ $item->text }}</span>
                            <label class="form-control"
id="input_div_{{ $item->id }}">{{ $answers[$item->id] or " " }}</label>
                                <span class="input-group-btn">
                                    <label class="btn"
onclick="removeAnswer({{ $item->id }})">
                                        <span
class="glyphicon glyphicon-chevron-right"></span>
                                            </label>
                                                </span>
                                                    <input type="hidden"
id="hidden_div_{{ $item->id }}" value="{{ $answerIds[$item->id] or " " }}" name="answers[{{ $item-
>id }}]">
                                                        </div>
                                                            </div>
                                                                </li>
                                                                    @endforeach
                                                                </ul>
                                                                    </td>
                                                                        <td style="width:35%">
                                                                            <ul class="list-group">
                                                                                @foreach ($choices as $choice)
                                                                                    <li class="list-group-item">
                                                                                        @if (in_array($choice->id, $answerIds))
                                                                                            <label class="btn btn-success form-control disabled"
id="choice_div_{{ $choice->id }}" onclick="insertAnswer({{ $choice->id }}, '{{ $choice->text }}');">
                                                                                                {{ $choice->text }}
                                                                                            </label>
                                                                                                @else
                                                                                            <label class="btn btn-success form-control"
id="choice_div_{{ $choice->id }}" onclick="insertAnswer({{ $choice->id }}, '{{ $choice->text }}');">
                                                                                                {{ $choice->text }}
                                                                                            </label>
                                                                                                @endif
                                                                                    </li>
                                                                                @endforeach
                                                                            </ul>

```

```

                </td>
            </tr>
        </table>
    </div>
</div>

<div class="form-group">
    <div class="col-md-6 col-md-offset-4">
        {!! Form::submit('Submit answers', ['class' => 'btn btn-primary form-control']) !!}
    </div>
</div>

<script type="text/javascript">
    var activeDiv = "";
    var activeId = "";

    function setToActive(itemId){
        activeDiv.className = "form-group";
        activeDiv = document.getElementById('item_div_' + itemId);
        activeDiv.className = "form-group has-success";
        activeId = itemId;
    }

    function insertAnswer(choiceId, answer){
        choiceDiv = document.getElementById('choice_div_' + choiceId);
        if (choiceDiv.className == 'btn btn-success form-control'){
            hiddenDiv = document.getElementById('hidden_div_' + activeId);
            if (hiddenDiv.value != ""){
                prevChoiceDiv = document.getElementById('choice_div_' +
hiddenDiv.value);
                prevChoiceDiv.className = "btn btn-success form-control";
            }
            hiddenDiv.value = choiceId;
            labelDiv = document.getElementById('input_div_' + activeId);
            labelDiv.innerHTML = answer;

            deactivateOption(choiceId);
        }
    }

    function deactivateOption(choiceId){
        choiceDiv = document.getElementById('choice_div_' + choiceId);
        choiceDiv.className = "btn btn-success form-control disabled";
    }
}

```



```

function removeAnswer(itemId){
    hiddenDiv = document.getElementById('hidden_div_' + itemId);
    choiceId = hiddenDiv.value;
    hiddenDiv.value = "";
    labelDiv = document.getElementById('input_div_' + itemId);
    labelDiv.innerHTML = "";

    activateOption(choiceId);
}

function activateOption(choiceId){
    choiceDiv = document.getElementById('choice_div_' + choiceId);
    choiceDiv.className = "btn btn-success form-control";
}
</script>
@endsection

```

Resources\Views\Grade-Section-Subjects\examinations\instances\multiple-choice.blade

```

@extends('grade-section-subjects.examinations.instances.layout')

@section('instance-form')
    @foreach ($questions as $question)
        <div class="panel panel-default">
            <div class="panel-heading">{!! $question->body !!</div>
            <table class="table table-responsive">
                <?php $count = 0; ?>
                <tr>
                    @foreach ($choices[$question->id] as $choice)
                        <?php
                            if ($count == 3){
                                echo "</tr><tr>";
                                $count = 0;
                            }
                        ?>
                        <td style="width:33%">
                            <label class="radio-inline">
                                @if (!strcmp($choice->text, $answers[$question->id]))
                                    {!! Form::radio('answers[' . $question->id . ']', $choice->id,
[checked']) !!} {!! $choice->text !!}
                                @else
                                    {!! Form::radio('answers[' . $question->id . ']', $choice->id) !!}
                                {!! $choice->text !!}
                            @endif
                        </td>
                    @endforeach
                </tr>
            </table>
        </div>
    @endforeach

```

```

                </label>
            </td>
            <?php $count++; ?>
        @endforeach
    </tr>
</table>
</div>
@endforeach

<div class="form-group">
    <div class="col-md-6 col-md-offset-4">
        {!! Form::submit('Submit answers', ['class' => 'btn btn-primary form-control'])
!!}
    </div>
</div>
@endsection

```

Resources\Views\Grade-Section-Subjects\examinations\instances\results.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Examination results
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    <ol class="breadcrumb pull-right">
```

```
        <li><a href="/home">Home</a></li>
```

```
        <li><a href="/examinations">All Examinations</a></li>
```

```
        <li><a href="/examinations/{{ $examination->id }}">{{ $examination->description }}</a></li>
```

```
        <li class="active">Results</li>
```

```
    </ol>
```

```
    <br></br><hr/>
```

```
    <div class="row">
```

```
        <div class="col-md-8 col-md-offset-2">
```

```
            <div class="panel panel-primary well">
```

```
                <div class="panel-heading">
```

```
                    <h5>Results of {{ $examination->description }}</h5>
```

```
                </div>
```

```
                <table class="table table-responsive table-hover">
```

```
                    <tr>
```

```
                        <th>Subcategory:</th><td>{{ $examination->subcategory }}</td>
```

```
                    </tr>
```

```
                    <tr>
```

```

        <th>Title:</th><td>{{ $examination->description }}</td>
    </tr>
    <tr>
        <th>Score:</th>
        <td>{{ $deploymentInstance->score }} / {{ $examination->computeTotalPoints() }}</td>
    </tr>
</table>
</div>
</div>
</div>
</div>
@endsection

```

Resources\Views\Grade-Section-Subjects\examinations\instances\select-from-the-wordbox.blade

```

@extends('grade-section-subjects.examinations.instances.layout')

@section('instance-form')
<div class="panel panel-default">
    <div class="panel-heading">{!! $question->body !!</div>
    <div class="panel-body">
        <div class="well">
            <table class="table">
                <tr>
                    <?php $count = 0 ?>
                    @foreach ($choices as $choice)
                        <?php
                            if ($count == 3){
                                echo "</tr><tr>";
                                $count = 0;
                            }
                        ?>
                    <td style="width:33%">
                        <label class="btn btn-success form-control"
onlick="insertAnswer('{{ $choice->text }}');">
                            {{ $choice->text }}
                        </label>
                    </td>
                    <?php $count++; ?>
                @endforeach
            </tr>
        </table>
    </div>

```

```

    @foreach ($items as $item)
        <div class="form-group" id="item_div_{{ $item->id }}"
onclick="setToActive({{ $item->id }})">
            <div class="input-group">
                <span class="input-group-addon">{{ $item->text }}</span>
                <label class="form-control" id="input_div_{{ $item->id }}">{{
$answers[$item->id] or " " }}</label>
                <input type="hidden" id="hidden_div_{{ $item->id }}"
value="{{ $answers[$item->id] or " " }}" name="answers[{{ $item->id }}" required>
            </div>
        </div>
    @endforeach
</div>
</div>

<div class="form-group">
    <div class="col-md-6 col-md-offset-4">
        {!! Form::submit('Submit answers', ['class' => 'btn btn-primary form-control']) !!}
    </div>
</div>

<script type="text/javascript">
    var activeDiv = "";
    var activeId = "";

    function setToActive(divId){
        activeDiv.className = "form-group";
        activeDiv = document.getElementById('item_div_' + divId);
        activeDiv.className = "form-group has-success";
        activeId = divId;
    }

    function insertAnswer(answer){
        hiddenDiv = document.getElementById('hidden_div_' + activeId);
        hiddenDiv.value = answer;
        labelDiv = document.getElementById('input_div_' + activeId);
        labelDiv.innerHTML = answer;
    }
</script>
@endsection

```

Resources\Views\Grade-Section-Subjects\examinations\instances\true-or-false.blade
@extends('grade-section-subjects.examinations.instances.layout')

```

@section('instance-form')
    @foreach ($Questions as $Question)
    <div class="panel panel-default">
        <div class="panel-heading">{!! $Question->body !!</div>
        <table class="table table-responsive">
            <tr>
                <td style="width:50%">
                    <label class="radio-inline">
                        @if ($answers[$Question->id] == '1')
                            {!! Form::radio('answers[' . $Question->id . ']', '1', ['checked']) !!}
                        @else
                            {!! Form::radio('answers[' . $Question->id . ']', '1') !!} True
                        @endif
                    </label>
                </td>
                <td>
                    <label class="radio-inline">
                        @if ($answers[$Question->id] == '0')
                            {!! Form::radio('answers[' . $Question->id . ']', '0', ['checked']) !!}
                        @else
                            {!! Form::radio('answers[' . $Question->id . ']', '0') !!}
                        @endif
                    </label>
                </td>
            </tr>
        </table>
    </div>
    @endforeach

    <div class="form-group">
        <div class="col-md-6 col-md-offset-4">
            {!! Form::submit('Submit answers', ['class' => 'btn btn-primary form-control'])
        !!}
        </div>
    </div>
    @endsection

```

Resources\Views\Grade-Section-Subjects\examinations\parts\content\create.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```

        Create new examination part
    @endsection

    @section('content')
    <div class="container">
        <div class="row">
            <div class="col-md-8 col-md-offset-2">
                <div class="panel panel-default">
                    <div class="panel-heading">Create new examination part</div>
                    <div class="panel-body">
                        {!! Form::model($part = new
    \App\Models\Examinations\ExaminationPart, ['url' => '/examinations/' . $examination->id . '/parts']) !!}
                        @include('grade-section-
    subjects.examinations.parts.content.form', ['submitButtonText' => 'Add part'])
                        {!! Form::close() !!}
                    </div>
                </div>
            </div>
        </div>
    </div>
    @endsection

```

Resources\Views\Grade-Section-Subjects\examinations\parts\content\delete.blade

```

    @extends('layouts.app')

    @section('title')
        Delete examination part
    @endsection

    @section('content')
    <div class="container">
        <div class="row">
            <div class="col-md-8 col-md-offset-2">
                <div class="panel panel-default">
                    <div class="panel-heading">Delete examination part</div>
                    <div class="panel-body">
                        {!! Form::open(['url' => '/examinations/' . $examination->id .
    '/parts/' . $part->id . '/delete']) !!}
                        <div class="form-group">
                            {!! Form::label(null, 'Are you sure you want to
    delete this part?', ['class' => 'col-md-4 control-label']) !!}
                            <blockquote>
                                <article class="panel panel-default">
                                    <div class="panel-heading">

```

```

        Question type: {{ $part->getQuestionType() }}
    </div>
    <div class="panel-body">
        Points: {!! $part->points !!}
        <br>
        Total no. of questions: {!! $part->questions_quantity
!!}
    </div>
</article>
    </blockquote>
    {!! Form::submit('Delete part', ['class' => 'btn
btn-danger']) !!}
    <input class="btn btn-primary" type="button"
onclick="location.href='/examinations/{{ $examination->id }}
/parts'" value="Back">
    </div>
    {!! Form::close() !!}
    </div>
    </div>
    </div>
    </div>
</div>
@endsection

```

Resources\Views\Grade-Section-Subjects\examinations\parts\content\edit.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Edit examination part details
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    <div class="row">
```

```
        <div class="col-md-8 col-md-offset-2">
```

```
    <div class="panel panel-default">
```

```
        <div class="panel-heading">Edit examination part details</div>
```

```
        <div class="panel-body">
```

```
            {!! Form::model($part, ['method' => 'PATCH', 'url' =>
'/examinations/' . $examination->id . '/parts/' . $part->id]) !!}

```

```
                @include('grade-section-
subjects.examinations.parts.content.form', ['submitButtonText' => 'Edit part details'])

```

```
            {!! Form::close() !!}

```

```
        </div>

```

```
    </div>

```

```

        </div>
    </div>
</div>
@endsection

```

Resources\Views\Grade-Section-Subjects\examinations\parts\content\form.blade

```

<div class="form-group">
    {!! Form::label('question_type_id', 'Question Type', ['class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
        {!! Form::select('question_type_id', $types, null, ['class' => 'form-control', 'placeholder' => 'Select
from the following...', 'required']) !!}
    </div>
</div>

<div class="form-group">
    {!! Form::label('points', 'Points per item:', ['class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
        {!! Form::number('points', (isset($part->points) ? $part->points : 1), ['min' => '1', 'class'
=> 'form-control', 'required']) !!}
    </div>
</div>

<div class="form-group">
    {!! Form::label('questions_quantity', 'Number of questions:', ['class' => 'col-md-4 control-label'])
!!}
    <div class="col-md-6">
        {!! Form::number('questions_quantity', (isset($part->questions_quantity) ? $part-
>questions_quantity : 1), ['min' => '1', 'class' => 'form-control', 'required']) !!}
    </div>
</div>

<div class="form-group pull-right">
    <div class="col-md-6 col-md-offset-4">
        <button type="button" class="btn btn-danger" onclick="location.href=/examinations/{
$examination->id } }/parts">
            Back
        </button>
    </div>
</div>

<div class="form-group pull-right">
    <div class="col-md-6 col-md-offset-4">
        {!! Form::submit($submitButtonText, ['class' => 'btn btn-primary form-control']) !!}
    </div>

```



```
</div>
```

Resources\Views\Grade-Section-Subjects\examinations\parts\content\index.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Parts of {{ $examination->description }}
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    @include('flash::message')
```

```
    <ol class="breadcrumb pull-right">
```

```
        <li><a href="/home">Home</a></li>
```

```
        <li><a href="/examinations">Examinations</a></li>
```

```
        <li><a href="/examinations/{{ $examination->id }}">{{ $examination->description }}</a></li>
```

```
        <li class="active">Parts</li>
```

```
    </ol>
```

```
    <br></br><hr/>
```

```
    <div class="btn-group pull-right">
```

```
        <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown" aria-  
haspopup="true" aria-expanded="false">
```

```
            <span class="glyphicon glyphicon-menu-hamburger" aria-hidden="true"></span> Menu <span  
class="caret"></span>
```

```
        </button>
```

```
        <ul class="dropdown-menu">
```

```
            <li><a href="/examinations/{{ $examination->id }}/parts/create">Add a new examination  
part</a></li>
```

```
        </ul>
```

```
    </div>
```

```
    <font size='6'>Parts of {{ $examination->description }}</font><br></br>
```

```
    <div class="row">
```

```
        <div class="col-md-10 col-md-offset-1">
```

```
            @if ($examination->parts->isEmpty())
```

```
                <div class="col-md-8 col-md-offset-2 well">
```

```
                    No parts added yet. Click <a href="/examinations/{{ $examination->id  
}}/parts/create">here</a> to start adding examination parts.
```

```
                </div>
```

```
            @else
```

```
                <?php $count = 1 ?>
```

```
                @foreach ($examination->parts as $part)
```

```
                    <div class="panel panel-default">
```

```
                        <div class="panel-heading">
```

```

    <a href="/examinations/{ { $examination->id } }/parts/{ { $part->id } }">Part { { $count
}}</a>
    <div class="dropdown pull-right">
        <button class="btn btn-default btn-xs dropdown-toggle" id="dLabel" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">
            <span class="caret"></span>
        </button>
        <ul class="dropdown-menu" aria-labelledby="dLabel">
            <li><a href="/examinations/{ { $examination->id } }/parts/{ { $part->id } }/edit">Edit
examination part details</a></li>
            <li class="divider"></li>
            <li><a href="/examinations/{ { $examination->id } }/parts/{ { $part->id
}}/delete">Delete examination part</a></li>
        </ul>
    </div>
</div>
<table class="table">
    <tr>
        <th style="width:50%">Question type:</th>
        <td style="width:50%">{ { $part->getQuestionType() } }</td>
    </tr>
    <tr>
        <th>Points per item:</th>
        <td>{ { $part->points } }</td>
    </tr>
    <tr>
        <th>Current no. of questions:</th>
        <td>{ { $part->getQuestionsCount() } }</td>
    </tr>
    <tr>
        <th>Total no. of questions:</th>
        <td>{ { $part->questions_quantity } }</td>
    </tr>
</table>
</div>
<?php $count++ ?>
    @endforeach
    @endif
</div>
</div>
@stop

```

Resources\Views\Grade-Section-Subjects\examinations\parts\content\show.blade

```

@extends('layouts.app')

@section('title')
    Included questions
@endsection

@section('content')
<div class="container">
    <ol class="breadcrumb pull-right">
        <li><a href="/home">Home</a></li>
        <li><a href="/examinations">Examinations</a></li>
        <li><a href="/examinations/{{ $examination->id }}">{{ $examination->description }}</a></li>
        <li><a href="/examinations/{{ $examination->id }}/parts">Parts</a></li>
        <li class="active">Included questions</li>
    </ol>
    <br></br><hr>
    <div class="btn-group pull-right">
        <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown" aria-
haspopup="true" aria-expanded="false">
            <span class="glyphicon glyphicon-menu-hamburger" aria-hidden="true"></span> Menu <span
class="caret"></span>
        </button>
        <ul class="dropdown-menu">
            <li><a href="/examinations/{{ $examination->id }}/parts/{{ $part->id }}/items/create">Add a
new examination part item</a></li>
        </ul>
    </div>
    <font size='6'>Included questions</font><br>
    <font size='3'>Type of questions: <b>{{ $part->getQuestionType() }}</b></font><br><br>
    <div class="row">
        <div class="col-md-10 col-md-offset-1">
            @if ($part->items->isEmpty())
                <div class="col-md-8 col-md-offset-2 well">
                    No questions included yet. Click <a href="/examinations/{{ $examination->id }}/parts/{{
$part->id }}/items/create">here</a> to start including questions.
                </div>
            @else
                @foreach ($part->items as $item)
                    <table class="table table-bordered">
                        <tr>
                            <th style="width:25%">Category:</th>
                            <td style="width:45%">{{ $item->questionSubtopic->questionTopic->questionCategory-
>name }}</td>
                            <td style="width:30%" colspan="2" rowspan="2">

```

```

        <div class="dropdown pull-right">
            <button class="btn btn-default btn-xs dropdown-toggle" id="dLabel" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">
                <span class="caret"></span>
            </button>
            <ul class="dropdown-menu" aria-labelledby="dLabel">
                <li><a href="/examinations/{ { $examination->id } }/parts/{ { $part->id } }/items/{ {
$item->id } }/delete">Delete examination part item</a></li>
            </ul>
        </div>
    </td>
</tr>
<tr>
    <th>Topic:</th>
    <td>{ { $item->questionSubtopic->questionTopic->name } }</td>
</tr>
<tr>
    <th>Subtopic:</th>
    <td>{ { $item->questionSubtopic->name } }</td>
    <th style="width:15%">Quantity:</th>
    <td style="width:15%">{ { $item->quantity } }</td>
</tr>
</table>
    @endforeach
    @endif
</div>
</div>
@stop

```

Resources\Views\Grade-Section-Subjects\examinations\parts\content\show-one.blade

```
@extends("layouts.app")
```

```
@section('title')
```

```
    Included questions
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    <ol class="breadcrumb pull-right">
```

```
        <li><a href="/home">Home</a></li>
```

```
        <li><a href="/examinations">Examinations</a></li>
```

```
        <li><a href="/examinations/{ { $examination->id } }">{ { $examination->description } }</a></li>
```

```
        <li><a href="/examinations/{ { $examination->id } }/parts">Parts</a></li>
```

```

    <li class="active">Included questions</li>
</ol>
<br></br><hr/>
<font size='6'>Included question</font><br/>
<font size='3'>Type of questions: <b>{{ $part->getQuestionType() }}</b></font><br></br>
<div class="row">
    <div class="col-md-10 col-md-offset-1">
        @if ($part->items->isEmpty())
            <div class="col-md-8 col-md-offset-2 well">
                No specific question included yet. Click <a href="/examinations/{{ $examination->id
}}/parts/{{ $part->id }}/items/create">here</a> to specify the question.
            </div>
        @else
            <div class="col-md-8 col-md-offset-2 well">
                Click <a href="/examinations/{{ $examination->id }}/parts/{{ $part->id
}}/items/create">here</a> to change the specified question. The previous one will be removed.
            </div>
            @foreach ($part->items as $item)
                <table class="table table-bordered">
                    <tr>
                        <th style="width:25%">Question:</th>
                        <th style="width:75%" colspan="3">{{ $question->title }}</th>
                    </tr>
                    <tr>
                        <th>Category:</th>
                        <td style="width:45%">{{ $item->questionSubtopic->questionTopic->questionCategory-
>name }}</td>
                        <td style="width:30%" colspan="2" rowspan="2">
                            <div class="dropdown pull-right">
                                <button class="btn btn-default btn-xs dropdown-toggle" id="dLabel" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">
                                    <span class="caret"></span>
                                </button>
                                <ul class="dropdown-menu" aria-labelledby="dLabel">
                                    <li><a href="/examinations/{{ $examination->id }}/parts/{{ $part->id }}/items/{{
$item->id }}/delete">Delete examination part item</a></li>
                                </ul>
                            </div>
                        </td>
                    </tr>
                    <tr>
                        <th>Topic:</th>
                        <td>{{ $item->questionSubtopic->questionTopic->name }}</td>
                    </tr>
                </table>
            @endforeach
        </div>
    </div>

```

```

        <tr>
            <th>Subtopic:</th>
            <td>{{ $item->questionSubtopic->name }}</td>
            <th style="width:15%">Quantity:</th>
            <td style="width:15%">{{ $item->quantity }}</td>
        </tr>
    </table>
    @endforeach
@endif
</div>
</div>
</div>
@stop

```

Resources\Views\Grade-Section-Subjects\examinations\parts\items\create.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Create new examination part item
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    @include('flash::message')
```

```
    <div class="row">
```

```
        <div class="col-md-8 col-md-offset-2">
```

```
    <div class="panel panel-default">
```

```
        <div class="panel-heading">Create new examination part item</div>
```

```
        <div class="panel-body">
```

```
            @include('grade-section-
subjects.examinations.parts.items.search')
```

```
                {!! Form::model($item = new
```

```
\App\Models\Examinations\ExaminationPartItem, ['url' => '/examinations/' . $examination->id . '/parts/' .
$part->id . '/items']) !!}
```

```
                @include('grade-section-
subjects.examinations.parts.items.form', ['submitButtonText' => 'Add item'])
```

```
                {!! Form::close() !!}
```

```
            </div>
```

```
        </div>
```

```
    </div>
```

```
</div>
```

```
</div>
```

```
<script type="text/javascript">
```

```

function configureTopicDropDownLists() {
    ddl1 = document.getElementById('category_ddl');
    ddl2 = document.getElementById('topic_ddl');

    switch (ddl1.value) {
    <?php
        foreach ($categories as $category){
            echo "case " . $category->id . " : " .
                "ddl2.options.length = 0; " .
                "var opt0 = document.createElement('option'); ";
            foreach ($topics[$category->id] as $topic){
                echo "var opt = document.createElement('option'); " .
                    "opt.value = " . $topic->id . " ; " .
                    "opt.text = " . $topic->name . " ; " .
                    "ddl2.options.add(opt);";
            }
            echo "break;";
        }
    ?>
    default:
        ddl2.options.length = 0;
    }
}
</script>
@endsection

```

Resources\Views\Grade-Section-Subjects\examinations\parts\items\create-ext.blade

```

<div class="form-group">
    {!! Form::label('question_subtopic_id', 'Subtopic', ['class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
        <select name="question_subtopic_id" id="subtopic_ddl"
onchange="configureSubtopicDropDownLists()" class="form-control" required>
            <option value="" disabled selected>Select from the following...</option>
            @foreach ($subtopics as $key => $name)
                <option value="{{ $key }}">{{ $name }}</option>
            @endforeach
        </select>
    </div>
</div>

<div class="form-group">
    {!! Form::label('question_id', 'Question', ['class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
        <select name="question_id" id="question_ddl" class="form-control" required>

```

```

        <option value="" disabled selected>Select from the following...</option>
    </select>
</div>
</div>

@if (!strcmp($part->getQuestionType(), 'Match Column A with Column B'))
<div class="form-group">
    {!! Form::label('choices_quantity', 'No. of wrong choices:', ['class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
        {!! Form::number('choices_quantity', 0, ['min' => '0', 'class' => 'form-control', 'required']) !!}
    </div>
</div>
</div>
@endif

<script type="text/javascript">
    function configureSubtopicDropDownLists() {
        ddl1 = document.getElementById('subtopic_ddl');
        ddl2 = document.getElementById('question_ddl');
        switch (ddl1.value) {
            <?php
                foreach ($subtopics as $key => $name){
                    echo "case " . $key . ": " .
                        "ddl2.options.length = 0; " .
                        "var opt0 = document.createElement('option'); " .
                        "opt0.value = "; " .
                        "opt0.text = 'Select from the following...'; " .
                        "ddl2.options.add(opt0); ";
                    foreach ($questions[$key] as $question){
                        echo "var opt = document.createElement('option'); " .
                            "opt.value = " . $question->id . "; " .
                            "opt.text = " . $question->title . "; " .
                            "ddl2.options.add(opt);";
                    }
                    echo "break;";
                }
            ?>
            default:
                ddl2.options.length = 0;
        }
    }
</script>

```

Resources\Views\Grade-Section-Subjects\examinations\parts\items\delete.blade
@extends('layouts.app')


```

@section('title')
    Delete examination part item
@endsection

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-8 col-md-offset-2">
            <div class="panel panel-default">
                <div class="panel-heading">Delete examination part item</div>
                <div class="panel-body">
                    {!! Form::open(['url' => '/examinations/' . $examination->id .
'/parts/' . $part->id . '/items/' . $item->id . '/delete']) !!}
                    <div class="form-group">
                        {!! Form::label(null, 'Are you sure you want to
delete this item?', ['class' => 'col-md-4 control-label']) !!}
                        <blockquote>
                            <table class="table table-bordered">
                                <tr>
                                    <th>Category:</th>
                                    <td>{{ $item->questionSubtopic->questionTopic-
>questionCategory->name }}</td>
                                </tr>
                                <tr>
                                    <th>Topic:</th>
                                    <td>{{ $item->questionSubtopic->questionTopic->name
}}</td>
                                </tr>
                                <tr>
                                    <th>Subtopic:</th>
                                    <td>{{ $item->questionSubtopic->name }}</td>
                                </tr>
                                <tr>
                                    <th>Quantity:</th>
                                    <td>{{ $item->quantity }}</td>
                                </tr>
                            </table>
                        </blockquote>
                        {!! Form::submit('Delete item', ['class' => 'btn
btn-danger']) !!}
                        <input class="btn btn-primary" type="button"
onclick="location.href='/examinations/{{ $examination->id }}/parts/{{ $part->id }}" value="Back">
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        {!! Form::close() !!}
    </div>
</div>
</div>
</div>
</div>
</div>
@endsection

```

Resources\Views\Grade-Section-Subjects\examinations\parts\items\form.blade

```

@unless (!strcmp($part->getQuestionType(), 'Select from the Wordbox') || !strcmp($part-
>getQuestionType(), 'Match Column A with Column B'))
<div class="form-group">
    {!! Form::label('question_subtopic_id', 'Subtopic', ['class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
        @if (empty($subtopics))
            {!! Form::select('question_subtopic_id', $subtopics, null, ['id' => 'subtopic_div',
'class' => 'form-control', 'placeholder' => 'Please select a category and topic first...', 'required', 'onchange'
=> 'updateMaxQuantity()']) !!}
        @else
            {!! Form::select('question_subtopic_id', $subtopics, null, ['id' => 'subtopic_div', 'class' =>
'form-control', 'placeholder' => 'Select from the following...', 'required', 'onchange' =>
'updateMaxQuantity()']) !!}
        @endif
    </div>
</div>
@else
    @include('grade-section-subjects.examinations.parts.items.create-ext')
@endif

@if (!strcmp($part->getQuestionType(), 'Match Column A with Column B') || !strcmp($part-
>getQuestionType(), 'Select from the Wordbox'))
    {!! Form::hidden('quantity', $part->questions_quantity) !!}
@else
<div class="form-group">
    {!! Form::label('quantity', 'Quantity:', ['id' => 'quantity_label', 'class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
        {!! Form::number('quantity', 1, ['id' => 'quantity', 'min' => '1', 'class' => 'form-control', 'required']) !!}
    </div>
</div>
@endif

<div class="form-group pull-right">
    <div class="col-md-6 col-md-offset-4">

```

```

        <button type="button" class="btn btn-danger" onclick="location.href='/examinations/{ {
$examination->id } }/parts'">
            Back
        </button>
    </div>
</div>

<div class="form-group pull-right">
    <div class="col-md-6 col-md-offset-4">
        {!! Form::submit($submitButtonText, ['class' => 'btn btn-primary form-control']) !!}
    </div>
</div>

<script type="text/javascript">
    function updateMaxQuantity() {
        var quantityDiv = document.getElementById('quantity');
        var subtopicDiv = document.getElementById('subtopic_div');
        var quantityLabel = document.getElementById('quantity_label');

        switch (subtopicDiv.value) {
            @foreach ($questionCount as $subtopicId => $count)
                case '{{ $subtopicId }}':
                    quantityDiv.max = '{{ $count }}';
                    quantityLabel.innerHTML = 'Quantity: (Max: {{ $count }})';
                    break;
            @endforeach
        }
    }
</script>

```

Resources\Views\Grade-Section-Subjects\examinations\parts\items\search.blade

```

<table class="table">
    {!! Form::open(['url' => '/examinations/' . $examination->id . '/parts/' . $part->id . '/items/create']) !!}
    !!}
    <tr>
        <th>Category</th>
        <th>Topic</th>
        <td style="width:25%"></td>
    </tr>
    <tr>
        <td>
            @if ($selectedCategory)
                {{ $selectedCategory->name }}
            @else

```

```

                <select name="category" id="category_ddl"
onchange="configureTopicDropDownLists()" class="form-control" required>
                <option value="" disabled selected>Select from the
following...</option>
                @foreach ($categories as $category)
                    <option value="{{ $category->id }}">{{ $category->name
}}</option>
                @endforeach
            </select>
        @endif
    </td>
    <td>
        @if ($selectedTopic)
            {{ $selectedTopic->name }}
        @else
            <select name="topic" id="topic_ddl" class="form-control">
                <option value="" disabled selected>Select from the
following...</option>
            </select>
        @endif
    </td>
    <td>
        @if ($selectedTopic == null)
            <button type="submit" class="btn btn-primary form-control">
                <span class="glyphicon glyphicon-search"></span> Search
            </button>
        @else
            <label onclick="location.href="/examinations/{{ $examination->id }}/parts/{{ $part->id
}}/items/create" class="btn btn-warning form-control">
                <span class="glyphicon glyphicon-repeat"></span> Reset search
            </label>
        @endunless
    </td>
</tr>
<tr>
    <td colspan="3">
        {!! Form::close() !!}
    </td>
</tr>
</table>

```

Resources\Views\Grade-Section-Subjects\posts\attach.blade

```

<div class="form-group pull-right">
    <div class="col-md-6 col-md-offset-4">
        <label class="btn btn-primary" onclick="attachMore()">
            <span class="glyphicon glyphicon-plus"></span> Attach more grade section
subjects
        </label>
    </div>
</div>

```

```

        </div>
</div>

<table class="table" id="additional_attachments">
    <tr>
        <th>Grade section subject</th>
        <th>Publish on</th>
    </tr>
    <tr>
        <td>{!! Form::select('grade_section_subject_id[]', $gradeSectionSubjects, null, ['class' => 'form-control', 'required']) !!}</td>
        <td>{!! Form::input('datetime-local', 'publish_on[]', $timeNow, ['class' => 'form-control', 'required']) !!}</td>
    </tr>
    <tr id="additional_attachments">
    </tr>
</table>

<script type="text/javascript">
    function attachMore(){
        var table = document.getElementById('additional_attachments');
        var row = table.insertRow();
        var cell1 = row.insertCell(0);
        var cell2 = row.insertCell(1);

        cell1.innerHTML = '<select class="form-control"
name="grade_section_subject_id[]" placeholder="Select from the following..." required>' +
            @foreach ($gradeSectionSubjects as $id
=> $name)
                '<option value="{{ $id }}">{{ $name
}}</option>' +
            @endforeach
            '</select>';
        cell2.innerHTML = '<input type="datetime-local" name="publish_on[]"
class="form-control" value="{{ $timeNow }}" required></input>';
    }
</script>

```

Resources\Views\Grade-Section-Subjects\posts\create.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Create new post
```

```
@endsection
```

```

@section('content')
<div class="container">
  <div class="row">
    <div class="col-md-8 col-md-offset-2">
      <div class="panel panel-default">
        <div class="panel-heading">Create a new post</div>
        <div class="panel-body">
          {!! Form::model($post = new
\App\Models\PostsAndRequirements\Post, ['url' => '/posts', 'files'=>true]) !!}
          @include('grade-section-subjects.posts.form',
['submitButtonText' => 'Create post', 'filesButtonText' => 'Insert files:'])
          {!! Form::close() !!}
        </div>
      </div>
    </div>
  </div>
</div>
@endsection

```

Resources\Views\Grade-Section-Subjects\posts\edit.blade

```

@extends('layouts.app')

@section('title')
  Edit post details
@endsection

@section('content')
<div class="container">
  <div class="row">
    <div class="col-md-8 col-md-offset-2">
      <div class="panel panel-default">
        <div class="panel-heading">Edit post details</div>
        <div class="panel-body">
          {!! Form::model($post, ['method' => 'PATCH', 'url' => '/posts/' .
$post->id, 'files'=>true]) !!}
          @include('grade-section-subjects.posts.form',
['submitButtonText' => 'Edit post', 'filesButtonText' => 'Add additional files:'])
          {!! Form::close() !!}
        </div>
      </div>
    </div>
  </div>
</div>

```

@endsection

Resources\Views\Grade-Section-Subjects\posts\form.blade

```
<div class="form-group{{ $errors->has('title') ? ' has-error' : '' }}">
    {!! Form::label('title', 'Title:', ['class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
        {!! Form::text('title', null, ['class' => 'form-control', 'required']) !!}

        @if ($errors->has('title'))
        <span class="help-block">
            <strong>{{ $errors->first('title') }}</strong>
        </span>
        @endif
    </div>
</div>

<div class="form-group{{ $errors->has('body') ? ' has-error' : '' }}">
    {!! Form::label('body', 'Body:', ['class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
        {!! Form::textarea('body', null, ['class' => 'form-control']) !!}
        @if ($errors->has('body'))
        <span class="help-block">
            <strong>{{ $errors->first('body') }}</strong>
        </span>
        @endif
    </div>
</div>

<div class="form-group{{ $errors->has('files') ? ' has-error' : '' }}">
    {!! Form::label('files', $filesButtonText . ' (Optional)', ['class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
        {!! Form::file('files[]', ['multiple'=>true]) !!}

        @if ($errors->has('files'))
        <span class="help-block">
            <strong>{{ $errors->first('files') }}</strong>
        </span>
        @endif
        <h5><font color='red'>* 16mb file size limit</font></h5>

        @if (count($post->files))
        {!! Form::label("", 'Please uncheck if you wish to remove file', ['class' => 'col-md-
4 control-label']) !!}
    </div>
</div>
```

```

                <br/>
                @foreach ($post->files as $file)
                    {!! Form::checkbox('old_files[]', $file->id, true) !!} {!! $file->file_name
            }}<br/>
        @endforeach
    @endif
</div>
</div>
<hr/>
@include('grade-section-subjects.posts.attach')

<div class="form-group pull-right">
    <div class="col-md-6 col-md-offset-4">
        <button type="button" class="btn btn-danger" onclick="location.href='/posts'">
            Back
        </button>
    </div>
</div>

<div class="form-group pull-right">
    <div class="col-md-6 col-md-offset-4">
        {!! Form::submit($submitButtonText, ['class' => 'btn btn-primary form-control']) !!}
    </div>
</div>

```

Resources\Views\Grade-Section-Subjects\posts\index.blade

```

@extends('layouts.app')

@section('title')
    Posts
@endsection

@section('content')
<div class="container">
    @include('flash::message')
    <ol class="breadcrumb pull-right">
        <li><a href="/home">Home</a></li>
        <li class="active">Posts</li>
    </ol>
    <br></br><hr/>
    <font size='6'>Posts</font>
    <div class="btn-group pull-right">
        <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown" aria-
haspopup="true" aria-expanded="false">

```



```

        <span class="glyphicon glyphicon-menu-hamburger" aria-hidden="true"></span> Menu <span
class="caret"></span>
    </button>
    <ul class="dropdown-menu">
        <li><a href="/posts/create">
            <span class="glyphicon glyphicon-file"></span> New post
        </a></li>
    </ul>
</div>
<br></br>
<div class="row">
    @if ($posts->isEmpty())
    <div class="col-md-8 col-md-offset-2 well">
        No posts found.
    </div>
    @else
    <div class="col-md-10 col-md-offset-1">
        @foreach ($posts as $post)
        <div class="panel panel-default">
            <div class="panel-heading">
                <a href="/posts/{{ $post->id }}">{{ $post->title }}</a>
                <div class="dropdown pull-right">
                    <button class="btn btn-default btn-xs dropdown-toggle" id="dLabel" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">
                        <span class="caret"></span>
                    </button>
                    <ul class="dropdown-menu" aria-labelledby="dLabel">
                        <li><a href="/posts/{{ $post->id }}/edit">Edit post details</a></li>
                    </ul>
                </div>
            </div>
        </div>
        @endforeach
    </div>
    @endif
</div>
@stop

```

Resources\Views\Grade-Section-Subjects\posts\show.blade

```
@extends('layouts.app')
```

```
@section('title')
    {{ $post->title }}

```

```
@endsection
```

```
@section('content')
```

```
<div class="container">  
  @include('flash::message')  
  <ol class="breadcrumb pull-right">  
    <li><a href="/home">Home</a></li>  
    <li><a href="/posts">Posts</a></li>  
    <li class="active">{{ $post->title }}</li>  
  </ol>  
  <br></br><hr/>  
  <div class="btn-group pull-right">  
    <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown" aria-  
haspopup="true" aria-expanded="false">  
      <span class="glyphicon glyphicon-menu-hamburger" aria-hidden="true"></span> Menu <span  
class="caret"></span>  
    </button>  
    <ul class="dropdown-menu">  
      <li><a href="/posts/{{ $post->id }}/edit">Edit post details</a></li>  
    </ul>  
  </div>  
  <br></br>  
  <div class="row">  
    <div class="col-md-10 col-md-offset-1">  
      <div class="panel panel-primary">  
        <div class="panel-heading">  
          Post details  
        </div>  
        <table class="table table-striped table-hover">  
          <tr>  
            <th>Title:</th><td>{{ $post->title }}</td>  
          </tr>  
          <tr>  
            <td colspan="2">{!! $post->body !!}</td>  
          </tr>  
          <tr>  
            <th>Created at:</th><td>{{ $post->created_at }}</td>  
          </tr>  
        </table>  
        <div class="well">  
          @if (count($post->files) == 0)  
            No files attached  
          @else  
            Files attached:
```

```

        <ul>
            @foreach ($post->files as $file)
                @include('layouts.file')
            @endforeach
        </ul>
    @endif
</div>
</div>
@if($attachments->isEmpty())
<div class="well">
    Not attached to any grade section subject yet.
</div>
@else
<div class="panel panel-primary">
    <div class="panel-heading">
        Attached to the following grade section subjects
    </div>
    <table class="table table-striped">
        <tr>
            <td></td>
            <th>Subject name</th>
            <th>Section name</th>
            <th>Publish on</th>
        </tr>
        <?php $count = 1 ?>
        @foreach ($attachments as $attachment)
            <tr>
                <td>{{ $count++ }}</td>
                <td>{{ $attachment->gradeSectionSubject->subject->name }}</td>
                <td>{{ $attachment->gradeSectionSubject->gradeSection->getName->name }}</td>
                <td>{{ $attachment->getUnformattedDate('publish_on') }}</td>
            </tr>
        @endforeach
    </table>
</div>
@endif
</div>
</div>
</div>
@stop

```

Resources\Views\Grade-Section-Subjects\requirements\attach.blade

```

<div class="form-group pull-right">
    <div class="col-md-6 col-md-offset-4">

```

```

        <label class="btn btn-primary" onclick="attachMore()">
            <span class="glyphicon glyphicon-plus"></span> Attach more grade section
subjects
        </label>
    </div>
</div>

<table class="table" id="additional_attachments">
    <tr>
        <td>Grade section subject</td>
        <td>Publish on</td>
    </tr>
    <tr>
        <td>{!! Form::select('grade_section_subject_id[]', $gradeSectionSubjects, null, ['class' => 'form-
control', 'required']) !!}</td>
        <td>{!! Form::input('datetime-local', 'publish_on[]', $timeNow, ['class' => 'form-control',
'required']) !!}</td>
    </tr>
    <tr>
        <td>Available to students from</td>
        <td>Available to students until</td>
    </tr>
    <tr>
        <td>{!! Form::input('datetime-local', 'event_start[]', $timeNow, ['class' => 'form-control',
'required']) !!}</td>
        <td>{!! Form::input('datetime-local', 'event_end[]', $timeEnd, ['class' => 'form-control',
'required']) !!}</td>
    </tr>
    <tr id="additional_attachments">
    </tr>
</table>

<script type="text/javascript">
    function attachMore(){
        var table = document.getElementById('additional_attachments');
        var row0 = table.insertRow();
        var cell1 = row0.insertCell(0);
        var cell2 = row0.insertCell(1);
        cell1.innerHTML = '<hr/>' +
            'Grade section subject<br>' +
            '<select class="form-control"
name="grade_section_subject_id[]" placeholder="Select from the following..." required>' +
            '@foreach ($gradeSectionSubjects as $id
=> $name)

```

```

}}</option>' +
                                '<option value="{{ $id }}">{{ $name
                                @endforeach
                                '</select><br>' +
                                'Available to students from<br>' +
                                '<input type="datetime-local"
name="event_start[]" class="form-control" value="{{ $timeNow }}" required></input>';

                                cell2.innerHTML = '<hr/>' +
                                'Publish on<br>' +
                                '<input type="datetime-local"
name="publish_on[]" class="form-control" value="{{ $timeNow }}" required></input><br>' +
                                'Available to students until<br>' +
                                '<input type="datetime-local"
name="event_end[]" class="form-control" value="{{ $timeEnd }}" required></input>';
                                }
</script>

```

Resources\Views\Grade-Section-Subjects\requirements\create.blade

```

@extends('layouts.app')

@section('title')
    Create new requirement
@endsection

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-8 col-md-offset-2">
            <div class="panel panel-default">
                <div class="panel-heading">Create a new requirement</div>
                <div class="panel-body">
                    {!! Form::model($requirement = new
\App\Models\PostsAndRequirements\Requirement, ['url' => '/requirements', 'files'=>true]) !!}
                    @include('grade-section-subjects.requirements.form',
['submitButtonText' => 'Create requirement', 'filesButtonText' => 'Insert files:'])
                    {!! Form::close() !!}
                </div>
            </div>
        </div>
    </div>
</div>
</div>
@endsection

```

Resources\Views\Grade-Section-Subjects\requirements\edit.blade

```
@extends('layouts.app')

@section('title')
    Edit requirement details
@endsection

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-8 col-md-offset-2">
            <div class="panel panel-default">
                <div class="panel-heading">Edit requirement details</div>
                <div class="panel-body">
                    {!! Form::model($requirement, ['method' => 'PATCH', 'url' =>
'/requirements/' . $requirement->id, 'files'=>true]) !!}
                    @include('grade-section-subjects.requirements.form',
['submitButtonText' => 'Edit requirement', 'filesButtonText' => 'Add additional files:'])
                    {!! Form::close() !!}
                </div>
            </div>
        </div>
    </div>
</div>
@endsection
```

Resources\Views\Grade-Section-Subjects\requirements\form.blade

```
<div class="form-group{{ $errors->has('title') ? ' has-error' : '' }}">
    {!! Form::label('title', 'Title:', ['class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
        {!! Form::text('title', null, ['class' => 'form-control', 'required']) !!}

        @if ($errors->has('title'))
            <span class="help-block">
                <strong>{{ $errors->first('title') }}</strong>
            </span>
        @endif
    </div>
</div>

<div class="form-group{{ $errors->has('body') ? ' has-error' : '' }}">
    {!! Form::label('body', 'Body:', ['class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
```

```

        {!! Form::textarea('body', null, ['class' => 'form-control']) !!}

        @if ($errors->has('body'))
            <span class="help-block">
                <strong>{{ $errors->first('body') }}</strong>
            </span>
        @endif
    </div>
</div>

<div class="form-group{{ $errors->has('files') ? ' has-error' : '' }}">
    {!! Form::label('files', $filesButtonText . ' (Optional)', ['class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
        {!! Form::file('files[]', ['multiple'=>true]) !!}

        @if ($errors->has('files'))
            <span class="help-block">
                <strong>{{ $errors->first('files') }}</strong>
            </span>
        @endif
        <h5><font color='red'>* 16mb file size limit</font></h5>

        @if (count($requirement->files))
            {!! Form::label("", 'Please uncheck if you wish to remove file', ['class' => 'col-md-4 control-label']) !!}
            <br/>
            @foreach ($requirement->files as $file)
                {!! Form::checkbox('old_files[]', $file->id, true) !!} {{ $file->file_name
            }}<br/>
            @endforeach
        @endif
    </div>
</div>
@include('grade-section-subjects.requirements.attach')

<div class="form-group pull-right">
    <div class="col-md-6 col-md-offset-4">
        <button type="button" class="btn btn-danger" onclick="location.href='/requirements'">
            Back
        </button>
    </div>
</div>

<div class="form-group pull-right">

```

```

        <div class="col-md-6 col-md-offset-4">
            {!! Form::submit($submitButtonText, ['class' => 'btn btn-primary form-control']) !!}
        </div>
    </div>

```

Resources\Views\Grade-Section-Subjects\requirements\index.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Requirements
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    @include('flash::message')
```

```
    <ol class="breadcrumb pull-right">
```

```
        <li><a href="/home">Home</a></li>
```

```
        <li class="active">Requirements</li>
```

```
    </ol>
```

```
    <br></br><hr/>
```

```
    <font size='6'>Requirements</font>
```

```
    <div class="btn-group pull-right">
```

```
        <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown" aria-
haspopup="true" aria-expanded="false">
```

```
            <span class="glyphicon glyphicon-menu-hamburger" aria-hidden="true"></span> Menu <span
class="caret"></span>
```

```
        </button>
```

```
        <ul class="dropdown-menu">
```

```
            <li><a href="/requirements/create">
```

```
                <span class="glyphicon glyphicon-paperclip"></span> New requirement
```

```
            </a></li>
```

```
        </ul>
```

```
    </div>
```

```
    <br></br>
```

```
    <div class="row">
```

```
        @if($requirements->isEmpty())
```

```
            <div class="col-md-8 col-md-offset-2 well">
```

```
                No requirements found.
```

```
            </div>
```

```
        @else
```

```
            <div class="col-md-10 col-md-offset-1">
```

```
                @foreach ($requirements as $requirement)
```

```
                    <div class="panel panel-default">
```

```
                        <div class="panel-heading">
```



```

        <a href="/requirements/{{ $requirement->id }}">{{ $requirement->title }}</a>
        <div class="dropdown pull-right">
            <button class="btn btn-default btn-xs dropdown-toggle" id="dLabel" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">
                <span class="caret"></span>
            </button>
            <ul class="dropdown-menu" aria-labelledby="dLabel">
                <li><a href="/requirements/{{ $requirement->id }}/edit">Edit requirement
details</a></li>
                <li class="divider"></li>
                <li><a href="/requirements/{{ $requirement->id }}/attach">Attach grade section
subjects to requirement</a></li>
            </ul>
        </div>
    </div>
</div>
@endforeach
</div>
@endif
</div>
</div>
@stop

```

Resources\Views\Grade-Section-Subjects\requirements\show.blade

```
@extends('layouts.app')
```

```

@section('title')
    {{ $requirement->title }}
@endsection

```

```

@section('content')
<div class="container">
    @include('flash::message')
    <ol class="breadcrumb pull-right">
        <li><a href="/home">Home</a></li>
        <li><a href="/requirements">Requirements</a></li>
        <li class="active">{{ $requirement->title }}</li>
    </ol>
    <br></br><hr>
    <div class="btn-group pull-right">
        <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown" aria-
haspopup="true" aria-expanded="false">
            <span class="glyphicon glyphicon-menu-hamburger" aria-hidden="true"></span> Menu <span
class="caret"></span>
    </div>

```

```

</button>
<ul class="dropdown-menu">
  <li><a href="/requirements/{{ $requirement->id }}/edit">Edit requirement details</a></li>
  <li class="divider"></li>
  <li><a href="/requirements/{{ $requirement->id }}/attach">Attach grade section subjects to
requirement</a></li>
</ul>
</div>
<br></br>
<div class="row">
  <div class="col-md-10 col-md-offset-1">
    <div class="panel panel-primary">
      <div class="panel-heading">
        Requirement details
      </div>
      <table class="table table-striped table-hover">
        <tr>
          <th>Title:</th><td>{{ $requirement->title }}</td>
        </tr>
        <tr>
          <td colspan="2">{!! $requirement->body !!}</td>
        </tr>
        <tr>
          <th>Created at:</th><td>{{ $requirement->created_at }}</td>
        </tr>
      </table>
      <div class="well">
        @if (count($requirement->files) == 0)
          No files attached
        @else
          Files attached:
          <ul>
            @foreach ($requirement->files as $file)
              @include('layouts.file')
            @endforeach
          </ul>
        @endif
      </div>
    </div>
    @if($attachments->isEmpty())
      <div class="well">
        Not attached to any grade section subject yet.
      </div>
    @else

```

```

<div class="panel panel-primary">
  <div class="panel-heading">
    Attached to the following grade section subjects
  </div>
  <table class="table table-striped">
    <tr>
      <td></td>
      <th>Subject name</th>
      <th>Section name</th>
      <th>Publish on</th>
      <th>Available to students from</th>
      <th>until</th>
    </tr>
    <?php $count = 1 ?>
    @foreach ($attachments as $attachment)
      <tr>
        <td>{{ $count++ }}</td>
        <td>{{ $attachment->gradeSectionSubject->subject->name }}</td>
        <td>{{ $attachment->gradeSectionSubject->gradeSection->getName->name }}</td>
        <td>{{ $attachment->getUnformattedDate('publish_on') }}</td>
        <td>{{ $attachment->getUnformattedDate('event_start') }}</td>
        <td>{{ $attachment->getUnformattedDate('event_end') }}</td>
      </tr>
    @endforeach
  </table>
</div>
@endif
</div>
</div>
@stop

```

Resources\Views\Grade-Section-Subjects\requirements\student-submissions.blade

```
@extends('layouts.app')
```

```
@section('title')
```

```
    Students submissions
```

```
@endsection
```

```
@section('content')
```

```
<div class="container">
```

```
    <ol class="breadcrumb pull-right">
```

```
        <li><a href="/home">Home</a></li>
```

```

    <li><a href="/subjects/{ { $gradeSectionSubject->id } }">{ { $gradeSectionSubject->subject->name
}}</a></li>
    <li class="active">{ { $requirement->title } } - Student submissions</li>
</ol>
    <div class="row">
        <div class="col-md-8 col-md-offset-2">
            <div class="panel panel-primary">
                <div class="panel-heading">
                    Submissions of students
                </div>
                <table class="table table-hover">
                    <tr>
                        <th>Name</th>
                        <th>Student ID Number</th>
                        <th>File submitted</th>
                        <th>Submitted at</th>
                    </tr>
                    @foreach ($gradeSectionSubject->gradeSection->students as $student)
                    <tr>
                        <th>{ { $student->getInfo()->lastname } }, { { $student->getInfo()-
>firstname } }</th>
                        <th>{ { $student->getInfo()->idNumber } }</th>
                        <th>
                            @if ($submissions[$student->id] == null)
                                None
                            @else
                                <?php $file = $submissions[$student->id]->getFile() ?>
                                {!! Form::open(['url' => 'download']) !!}
                                {!! Form::hidden('fileId', $file->id) !!}
                                <input type="submit" class="submitLink" value="{ { $file-
>file_name } }">
                                    {!! Form::close() !!}
                                @endif
                            </th>
                        <th>{ { ($submissions[$student->id] == null) ? 'None' :
$submissions[$student->id]->submitted_at } }</th>
                    </tr>
                    @endforeach
                </table>
            </div>
        </div>
    </div>
</div>
@endsection

```

Resources\Views\Layouts\app.blade

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <title>@yield('title')</title>

  <!-- Fonts -->
  <link rel='stylesheet' type='text/css' href="/fonts/font-awesome.min.css">
  <link rel='stylesheet' type='text/css' href="/fonts/font-googleapis.css">

  <!-- Styles -->
  <link rel="stylesheet" href="/css/bootstrap.min.css">
  <link rel="stylesheet" href="/css/additional.css">
  <link rel="stylesheet" href="/css/selectize.bootstrap3.css">
  {{-- <link href="{{ elixir('css/app.css') }}" rel="stylesheet" --}}

  <style>
    body {
      font-family: 'Lato';
    }

    .fa-btn {
      margin-right: 6px;
    }
  </style>

  <!-- JavaScripts -->
  <script src="/js/jquery.min.js"></script>
  <script src="/js/standalone/selectize.js"></script>
  <script src="/js/nicEdit.js"></script>
  <script type="text/javascript">bkLib.onDomLoaded(nicEditors.allTextAreas);</script>
</head>
<body id="app-layout">
  @include('layouts.navbar')

  @yield('content')

  <!-- JavaScripts -->
  <script src="/js/bootstrap.min.js"></script>
```

```

    {{-- <script src="{{ elixir('js/app.js') }}"></script> --}}
</body>
</html>

```

Resources\Views\Layouts\file.blade

```

{!! Form::open(['url' => 'download']) !!}
    {!! Form::hidden('fileId', $file->id) !!}
    <span class='glyphicon glyphicon-download-alt'></span>
    <input type="submit" class="submitLink" value="{{ $file->file_name }}">
    @if ($is_teacher)
    <div class="dropdown pull-right">
        <button class="btn btn-default btn-xs dropdown-toggle" id="dLabel" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">
            <span class="caret"></span>
        </button>
        <ul class="dropdown-menu" aria-labelledby="dLabel">
            <li><a href="/files/download-history/{{ $file->id }}" target="_blank">View download
history</a></li>
        </ul>
    </div>
    @endif
{!! Form::close() !!}

```

Resources\Views\Layouts\navbar.blade

```

<nav class="navbar navbar-default">
    <div class="container">
        <div class="navbar-header">

            <!-- Collapsed Hamburger -->
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#app-
navbar-collapse">
                <span class="sr-only">Toggle Navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>

            <!-- Branding Image -->
            <a class="navbar-brand" href="{{ url('/') }}">DEECO</a>
        </div>

        <div class="collapse navbar-collapse" id="app-navbar-collapse">
            <!-- Left Side Of Navbar -->
            <ul class="nav navbar-nav">

```

```

        <li><a href="/home">Home</a></li>
        @yield('navbar-links')
    </ul>

    <!-- Right Side Of Navbar -->
    <ul class="nav navbar-nav navbar-right">
        <!-- Authentication Links -->
        @if (Auth::guest())
            <li><a href="{{ url('/login') }}">Login</a></li>
        @else
            <li class="dropdown">
                <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-
expanded="false">
                    {{ Auth::user()->username }} <span class="caret"></span>
                </a>
                <ul class="dropdown-menu" role="menu">
                    <li><a href="{{ url('/logout') }}"><span class="glyphicon glyphicon-log-out"></span>
Logout</a></li>
                    <li class="divider"></li>
                    <li><a href="/password/change">Change password</a></li>
                </ul>
            </li>
        @endif
    </ul>
</div>
</div>
</nav>

```

Resources\Views\Questions\Categories\create.blade

```

@extends('layouts.app')

@section('title')
    Add a new question category
@endsection

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-8 col-md-offset-2">
            <div class="panel panel-default">
                <div class="panel-heading">Add a new question category</div>
                <div class="panel-body">
                    {!! Form::model($category = new
\App\Models\Questions\QuestionCategory, ['url' => 'categories']) !!}

```



```

                                <input class="btn btn-primary" type="button"
onclick="location.href='/categories'" value="Back">
                                {!! Form::close() !!}
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        @endsection

```

Resources\Views\Questions\Categories\edit.blade

```

@extends('layouts.app')

@section('title')
    Edit question category
@endsection

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-8 col-md-offset-2">
            <div class="panel panel-default">
                <div class="panel-heading">Edit question category</div>
                <div class="panel-body">
                    {!! Form::model($category, ['method' => 'PATCH', 'url' =>
'categories/' . $category->name]) !!}
                    @include('questions.categories.form',
[submitButtonText' => 'Edit category'])
                    {!! Form::close() !!}
                </div>
            </div>
        </div>
    </div>
</div>
@endsection

```

Resources\Views\Questions\Categories\form.blade

```

<div class="form-group"{{ $errors->has('name') ? ' has-error' : " }}">
    {!! Form::label('name', 'Name:', ['class' => 'col-md-4 control-label']) !!}
    <div class="col-md-6">
        {!! Form::text('name', null, ['class' => 'form-control']) !!}

        @if ($errors->has('name'))
            <span class="help-block">

```

```

        <strong>{{ $errors->first('name') }}</strong>
    </span>
@endif
</div>
</div>

<div class="form-group pull-right">
    <div class="col-md-6 col-md-offset-4">
        <button type="button" class="btn btn-danger" onclick="location.href='/categories'">
            Back
        </button>
    </div>
</div>

<div class="form-group pull-right">
    <div class="col-md-6 col-md-offset-4">
        {!! Form::submit($submitButtonText, ['class' => 'btn btn-primary form-control']) !!}
    </div>
</div>

```

Resources\Views\Questions\Categories\index.blade

```

@extends('layouts.app')

@section('title')
    Categories
@endsection

@section('content')
<div class="container">
    @include('flash::message')
    <ol class="breadcrumb pull-right">
        <li><a href="/examinations">Back to examinations</a></li>
        <li class="active">All categories</li>
    </ol>
    <br></br><hr/>
    <font size='6'>Categories</font>
    <button type="button" class="btn btn-primary pull-right" onclick="location.href='/categories/create'">
        <span class="glyphicon glyphicon-plus" aria-hidden="true"></span> Add a new category
    </button>
    <br></br>
    @foreach ($categories as $category)
        <div class="row">
            <div class="col-md-10 col-md-offset-1">
                <div class="panel panel-default">

```

```

<div class="panel-heading">
  <a href="/categories/{{ $category->name }}">{{ $category->name }}</a>
  @if (strcmp($category->name, 'Default Category') != 0)
  <div class="dropdown pull-right">
    <button class="btn btn-default btn-xs dropdown-toggle" id="dLabel" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">
      <span class="caret"></span>
    </button>
    <ul class="dropdown-menu" aria-labelledby="dLabel">
      <li><a href="/categories/{{ $category->name }}/edit">Edit name</a></li>
      <li class="divider"></li>
      <li><a href="/categories/{{ $category->name }}/delete">Delete category</a></li>
    </ul>
  </div>
  @endif
</div>
</div>
</div>
</div>
</div>
@endforeach
</div>
@stop

```

Resources\Views\Questions\Categories\show.blade

```

@extends('layouts.app')

@section('title')
  {{ $category->name }}
@endsection

@section('content')
<div class="container">
  @include('flash::message')
  <ol class="breadcrumb pull-right">
    <li><a href="/examinations">Back to examinations</a></li>
    <li><a href="/categories">All categories</a></li>
    <li class="active">{{ $category->name }}</li>
  </ol>
  <br></br><hr>
  <font size='4'><b>Category: &nbsp;</b></font><font size='6'>{{ $category->name }}</font>
  <button type="button" class="btn btn-primary pull-right" onclick="location.href="/categories/{{
$category->name }}/topics/create">
    <span class="glyphicon glyphicon-plus" aria-hidden="true"></span> Add a new topic
  </button>

```

```

<br></br>
<?php $count = 0 ?>
@foreach ($topics as $topic)
    <div class="row">
        <div class="col-md-10 col-md-offset-1">
            <div class="panel panel-default">
                <div class="panel-heading">
                    <button id="span_right_{{ $count }}" type="button" class="btn btn-default btn-xs"
onclick="showhide('{{ $count }}', 1)" style="display:;>
                        <span class="glyphicon glyphicon-menu-right"></span>
                    </button>
                    <button id="span_down_{{ $count }}" type="button" class="btn btn-default btn-xs"
onclick="showhide('{{ $count }}', 0)" style="display: none;>
                        <span class="glyphicon glyphicon-menu-down"></span>
                    </button>
                    <a href="/categories/{{ $category->name }}/topics/{{ $topic->name }}">{{ $topic->name
}}</a>
                @if (strcmp($topic->name, 'Default Topic') != 0)
                    <div class="dropdown pull-right">
                        <button class="btn btn-default btn-xs dropdown-toggle" id="dLabel" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">
                            <span class="caret"></span>
                        </button>
                        <ul class="dropdown-menu" aria-labelledby="dLabel">
                            <li><a href="/categories/{{ $category->name }}/topics/{{ $topic->name }}/edit">Edit
name</a></li>
                            <li class="divider"></li>
                            <li><a href="/categories/{{ $category->name }}/topics/{{ $topic->name
}}/delete">Delete topic</a></li>
                        </ul>
                    </div>
                @endif
            </div>
            <div class="panel-body" id='{{ $count }}' style="display:none;>
                <button type="button" class="btn btn-primary btn-xs" onclick="location.href='/categories/{{
$category->name }}/topics/{{ $topic->name }}/subtopics/create">
                    <span class="glyphicon glyphicon-plus" aria-hidden="true"></span> Add a new subtopic
                </button>
                <br></br>
                @foreach ($topic->questionSubtopics as $subtopic)
                    <div class="panel panel-default">
                        <div class="panel-heading">
                            <a href="/categories/{{ $category->name }}/topics/{{ $topic->name }}/subtopics/{{
$subtopic->name }}">{{ $subtopic->name }}</a>

```

```

        @if (strcmp($subtopic->name, 'Default Subtopic') != 0)
        <div class="dropdown pull-right">
            <button class="btn btn-default btn-xs dropdown-toggle" id="dLabel" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">
                <span class="caret"></span>
            </button>
            <ul class="dropdown-menu" aria-labelledby="dLabel">
                <li><a href="/categories/{ { $category->name } }/topics/{ { $topic->name
}}/subtopics/{ { $subtopic->name } }/edit">Edit name</a></li>
                <li class="divider"></li>
                <li><a href="/categories/{ { $category->name } }/topics/{ { $topic->name
}}/subtopics/{ { $subtopic->name } }/delete">Delete subtopic</a></li>
            </ul>
        </div>
        @endif
    </div>
</div>
    @endforeach
</div>
</div>
</div>
</div>
<?php $count++ ?>
    @endforeach
</div>

```

```

<script type="text/javascript">
function showhide(id, counter) {
    var e = document.getElementById(id);
    e.style.display = (e.style.display == 'block') ? 'none' : 'block';

    if(counter){
        var spanId = 'span_right_' + id;
        document.getElementById(spanId).style.display = 'none';
        var spanId2 = 'span_down_' + id;
        document.getElementById(spanId2).style.display = "";
    } else {
        var spanId = 'span_down_' + id;
        document.getElementById(spanId).style.display = 'none';
        var spanId2 = 'span_right_' + id;
        document.getElementById(spanId2).style.display = "";
    }
}
</script>

```

@stop

Resources\Views\Questions\Content

Resources\Views\Questions\Content

Resources\Views\Questions\Content

Resources\Views\Questions\Content

Resources\Views\Questions\Content

Resources\Views\Questions\Content

Resources\Views\Questions\Subtopics

Resources\Views\Questions\Topics

XI. Acknowledgement

Starting with the most commonly thanked person in acknowledgements, I would like to thank God. This may seem so ordinary to most people, but to me, this means a lot. If there was anyone who I would like to thank the most, it really is God. God made all of this happen. I was at the point of giving up, yet it was His promise to me, in Jeremiah 29:11, which came through. That promise gave me hope when I felt doomed, strength when I felt weak, and wisdom when I lacked it the most. Thank you so much, Lord.

I would also like to thank my family. This also seems like something ordinary, but I cannot express my gratitude for these people. They provided help in every opportunity they can. They kept me sane when I was close to madness; they gave me hope when I didn't have any. They have kept on trusting me when even I can't trust myself. They saw in me someone who can push through, and that's what I became. To my family, and I know you will be reading this, and you're all probably laughing now because of these cheesy lines, thank you so much for the sacrifices you've endured for me. Know that from the bottom of my heart, I am really grateful.

To my friends who never gave up on me, to all of YOU, thank you. To one of my role models, Kuya Raymond, thank you for putting your trust in me and never giving up on my potential. Thank you for never being annoyed with me even when I don't show up. Thank you for taking time to help me in my struggles.

To my team, to Kuya Josh, Daren, Lloyd, Kuya Berce, Henry, Mark, Daniel and Daniel, Andre, and Kuya Jeff, thank you guys for your never ending support. I always know I can count on you guys during times of trouble. To my team leader, Kuya Josh. I'm sorry I never made it in

your despedida, and it will be a long time without you my friend, but I'll tell you all about it when I see you again.

To my blockmates, thank you guys for the four years (or should I say five?) that we've had. I know that some of you hated me at some point (and I know you are btw) but please accept my apology. Know that it's the good memories that will stick in my mind, not the bad ones.

To UPM-SONS, I can't even begin on how to say thanks to you guys. I know that there are times when we're not really ourselves, and there are even times when we have very emotional conversations *ahem* *ahem*, but those memories cannot compare to the all the great times we've had. Just a quick rundown of those people, thank you Ivan, Toni, Greame, Ghe, Ronnie, Ate Rudj, Ate Sharyl, Ate Kim, Kuya Robin, Dareen, Alex, Aaron, Michelle, Ate Sha. To those not in here, I still want to thank you.

To my adviser, Sir Baes, thank you for putting up with my incompetence. I know I may not be the best advisee you had, but know that I really did my best, and thank you for putting your faith in me.

To my cousins who somehow had something amazing happen to them this 2016, thank for that pressure you put me. No, seriously, that pressure helped gave me momentum in finishing my SP.

To my laptop, thank you so much for not breaking down when I needed you the most. You may be the most used item I ever had in my life, yet you kept strong for me and didn't crash and broke or something. Just don't scare me like crashing right before I have my SP defense again, cool?

To all others who I've probably forgot to include here, I still want to thank you, and sorry for not putting you here. Maybe next time...