

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

ELECTRONIC CLASS RECORD INQUIRY SYSTEM

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

Submitted by:

Dan Chester D. Francisco

May 2018

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 "Electronic Class Record Inquiry System" prepared and submitted by Dan Chester D. Francisco in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

Avegail D. Carpio, M.S.
Adviser

EXAMINERS:

	Approved	Disapproved
1. Gregorio B. Baes, Ph.D. (<i>candidate</i>)	_____	_____
2. Richard Bryann L. Chua, Ph.D. (<i>candidate</i>)	_____	_____
3. Perlita E. Gasmien, M.S. (<i>candidate</i>)	_____	_____
4. Marvin John C. Ignacio, M.S. (<i>candidate</i>)	_____	_____
5. Ma. Sheila A. Magboo, M.S.	_____	_____
6. Vincent Peter C. Magboo, M.D., M.S.	_____	_____
7. Geoffrey A. Solano, Ph.D. (<i>candidate</i>)	_____	_____

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

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

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

Abstract

Student class record serves as a documented evidence for tracking the performance of the students. It contains numerical values that represents the growth and progress of the students. With the enormous amount of information, the process of managing these documents causes problems to the teachers. This system aims to systematically record, store, and update student class records through the use of spreadsheet application. The project will also serve as a primary tool for schools on monitoring the growth and development of the students.

Keywords: online spreadsheet, grades management, student class record

Contents

Acceptance Sheet	i
Abstract	ii
List of Figures	v
List of Tables	vii
I. Introduction	1
A. Background of the Study	1
B. Statement of the Problem	2
C. Objectives of the Study	2
D. Significance of the Project	4
E. Scope and Limitations	4
F. Assumptions	5
II. Review of Related Literature	6
III. Theoretical Framework	11
A. Spreadsheet	11
B. Class Record	14
C. Grading System	15
D. Grade Aggregation	16
E. Information System	17
F. Database Management System (DBMS)	18
G. CodeIgniter	18
H. Bootstrap	19
IV. Design and Implementation	20
A. Use Case Diagram	20
B. Activity Diagrams	21

C. Entity Relationship Diagram	25
D. Database Design	26
V. Results	35
VI. Discussions	57
VII. Conclusions	59
VIII. Recommendations	61
IX. Bibliography	62
X. Appendix	64
A. Source Codes	64
XI. Acknowledgement	214

List of Figures

1	An example of relative reference	12
2	An example of absolute reference	13
3	Spreadsheet window displaying results of computation	13
4	Graph-oriented view of cell dependency	14
5	PUP Grading Scale	15
6	DLSU Grading System	16
7	Natural Aggregation Method	17
8	Top-level use case diagram of ECRIS	20
9	Activity Diagram - ECRIS View Student Record	21
10	Activity Diagram - ECRIS Modify Course Record	22
11	Activity Diagram - Assign Grading Scale	23
12	Activity Diagram - Assign Grading Scale	24
13	Entity Relationship Diagram - ECRIS	25
14	Login page	35
15	Register School Admin page	36
16	Account Verification e-mail	36
17	Verify e-mail Form	37
18	Verify e-mail Success	38
19	First Login - Forced Change Password Form	38
20	Home Page - System Administrator	39
21	Register User Account - System Administrator	40
22	List of user accounts - System Administrator	40
23	Upload multiple accounts - System Administrator	41
24	CSV File Preview - System Administrator	42
25	List of Student Groups - System Administrator	42
26	Create Student Groups - System Administrator	43
27	Create Course Form - System Administrator	43
28	List of Courses - System Administrator	44

29	List of assigned students in a course - System Administrator	45
30	Assigning students to course - System Administrator	45
31	Creating Course Category - System Administrator	46
32	Updating Course Category - System Administrator	47
33	Dashboard - Department Head/ Teacher	47
34	Update Class Record - Department Head/ Teacher	48
35	Modifying Course Categories - Department Head	49
36	Modifying Gradebook Setup - Department Head/Teacher	49
37	Modifying Gradebook Categories/Items - Department Head/Teacher	50
38	Modifying Gradebook Subitems - Department Head/Teacher	51
39	Update Grading Scale - Department Head/Teacher	51
40	Save/Use Gradebook Scale - Department Head/Teacher	52
41	Student Grade Stats - Teacher	53
42	Course Announcement - Teacher	53
43	Student Record - Student	54
44	View Announcement - Student	54
45	Announcement e-mail - Student	55
46	Account Settings - Users	56

List of Tables

1	announcement table	26
2	category members table	26
3	class record table	27
4	course table	27
5	course category table	28
6	course name table	28
7	course students table	28
8	final grade table	29
9	grade item table	29
10	gradebook setup details table	30
11	gradebook scale table	30
12	main category table	31
13	mapping table	31
14	scale template table	32
15	scale template members table	32
16	scale values table	32
17	school table	32
18	student table	33
19	student group table	33
20	student group member table	33
21	teacher table	34
22	user table	34
23	user details table	34

I. Introduction

A. Background of the Study

The integration of information and communication technology into education has been one of the most efficient solutions in the field of teaching and learning processes [1]. As it continues to improve, these issues in the education sector have been tried to resolve by introducing new alternative methods. One of the greatest benefits of technology that teachers, and especially students, are enjoying is the access to computers and the Internet. It has been revolutionizing the way how educators teach and students learn. This privilege has been helpful to the educators in which they can be able to share information in an efficient, non-traditional way. Aside from its capability for communication, the teachers can also learn new methods for teaching and record handling.

With the huge amount of information that schools are storing, the teacher plays a vital role in providing complete and accurate records of the students. As a basis for student progress, schools record the development of students by representing student performance with numerical values or grades. The records can help the teachers and parents on their observation about the students' learning needs. Compared to the traditional use of ledgers, teachers are now able to record the student grades digitally with the use of Microsoft Excel and other spreadsheet software.

In software organizations, spreadsheets are implemented to gather data from different systems, to perform sequence of operations for improving data, to store increasing amounts of data, and to describe information and mathematical formulas in tabular form [2]. Various tasks such as simple organizing lists of addresses to complex simulations of biological data sets can be performed through the use of spreadsheets [3].

B. Statement of the Problem

As the student records that are need to be provided and maintained gradually increase, recording student grades manually can cause problems—such as errors in computation and improper file handling—to the teachers. In addition, improper record handling can also affect the observation of the teacher in monitoring the performance of the students.

The process of grading consists of several factors or criteria that need to be recorded in order to produce complete and accurate data representation of the student record. Unclear grading criteria or grading system is also an issue that can affect in monitoring the student’s performance.

At present, some teachers are able to record student grades through the use of electronic and online spreadsheet. However, this can also cause an issue about disclosing student information since the record is easily accessible. Most online spreadsheets, by default, display the whole record in which the one student can view the records of the other one.

C. Objectives of the Study

The project aims to design and develop the Electronic Class Record Inquiry System (ECRIS), a generic web-based grades management system, that will help the schools to provide and manage accurate student grade records. The system will implement a spreadsheet library that will support the teachers in recording the grades of the students.

1. Users:
 - (a) Log in with their accounts
 - (b) Change their account password
2. School Administrator:
 - (a) Sign up for the school

- (b) Create accounts for students, teachers and department heads.
- (c) Update or delete accounts of the students, teachers and department heads.
- (d) Create, update or remove courses
- (e) Clear the course record data after the duration of the course
- (f) Create, update or delete course categories
- (g) Assign department heads to course categories
- (h) Assign students and teachers to courses

3. Teacher:

- (a) View or update student course records
- (b) Add, Edit or Delete gradebook categories/items/sub-items (with approval of the department head).
- (c) Generate mean, maximum score, minimum score, histogram of the student grades
- (d) Add, edit, or delete course announcements

4. Department Head:

- (a) View or update class records
- (b) Add, edit or delete gradebook categories/items/sub-items.
- (c) Generate statistical report of the student records
- (d) Add, edit or delete course announcements
- (e) Update grading system
- (f) Allow the teachers to modify gradebook categories/items/sub-items

5. Student:

- (a) View student record
- (b) View course announcements

D. Significance of the Project

The Electronic Class Record Inquiry System (ECRIS) will help the teachers on monitoring and maintaining accurate student class records. This type of approach that focuses on helping the system users, especially the teachers, on managing the spreadsheet can reduce the number of errors in computation and improper file management. The system will also solve the issue of unclear grading criteria being implemented by the teacher. Moreover, the system will also resolve the problem of disclosing student records by filtering the information that will be displayed to every student.

The use of technology has made gathering of data, updating and presenting records more efficient that will be helpful for both the teachers and the students involved. With the system, the students can now access their records online to verify their performance in a certain subject. The teachers can also monitor the performance of the students. These records will serve as a basis for the teachers to focus on the students who are not performing well or focus on the topics that the students find difficult to comprehend.

In addition, the system will send an e-mail message to the students every time announcements are made by the teachers. This feature will be useful for the students upon inquiring for the grades that they need and receiving announcements from their respective teachers.

E. Scope and Limitations

1. The system is not capable of managing other tasks that will affect the grades of the students involved.
2. The school administrators are the only ones allowed to assign/reassign the department heads, teachers, and students in class.
3. The school administrators are the only ones allowed to clear the course information and student records.

4. The system uses 3 grading methods: natural aggregation, mean of grades, weighted mean of grades.
5. The system also allows 2 levels per gradebook setup.

F. Assumptions

1. The teachers are held responsible for the errors during the encoding of the student grades.
2. The school administrator should be updated with any changes in the enrolment information.

II. Review of Related Literature

Spreadsheet errors have been a problem to the developers since the start of end-user programming. Several studies have been discussed for the improvement of spreadsheet systems in order to provide better support to the end users. Many researchers focused on presenting methodologies and developing software for improved spreadsheet implementation.

At present, Microsoft Excel and OpenOffice Calc have presented the illusion of simplicity because of its free-form structure. These tools have been popular for end users. As users continued to create and modify more complex spreadsheets, the need for self-imposed structure in order to conform to the needs of the users is an indication that free-form structure has its shortcomings [4]. A study by **Mireault** (2012) has proposed a spreadsheet development methodology named Structured Spreadsheet Modeling and Implementation (SSMI). This methodology utilizes proven techniques in the field of Information Systems and Software Engineering to develop rules in spreadsheet implementation.

With the number of users continue to increase, different developers (end users) are concerned with different functionalities of the system. However, while modularity mechanisms and abstraction are offered in modern programming languages, spreadsheet systems do not offer such support to the users [5, 6]. This problem of lack in modularity and abstraction has led to the development of specification languages such as ViTSL (Visual Template Specification Language) and ClassSheet [6]. In order to resolve the lack of modularity, a study about Aspect Oriented Programming has been discussed. In this style of programming, the end users do not intrusively modify the original program (the spreadsheet), but instead, a new code fragment is defined in order to determine the changes in the program (changes in the spreadsheet) that needs concern [5].

Aside from instructing the end users on efficient methods of spreadsheet implementation, some researchers are inclined on designing and implementing software tools for spreadsheet systems. One of these has considered the application of mod-

els for guiding the users in introducing correct data. These spreadsheet models, also known as ClassSheet models, are used to define the business logic of the underlying spreadsheet data. ClassSheet uses high-level, object-oriented formalism to determine the business rules within spreadsheets [2]. The design of ClassSheet enables the users to define spreadsheet tables and give them names; it is also possible to define labels for the columns or rows, and to specify the types of values the cells may contain. ClassSheet is also capable of automatic gathering of data instance once a model is defined, or to produce a model from a particular data spreadsheet. These models were used to generate spreadsheet instances following a specific ClassSheet models. This spreadsheet mechanism guarantees that the end users always produce a spreadsheet instance that corresponds to the following model.

In addition to the studies presented above, University of Copenhagen has published a book manuscript regarding spreadsheet technology. The manuscript includes the summary of the spreadsheet computation model, challenges for efficient recalculation, and description of the Corecalc. Corecalc is a spreadsheet system that is designed and implemented in C#. The system is small and simple enough that enable users to conduct experiments with design decisions, yet it contains all essential functions a spreadsheet can offer [3].

As stated earlier, existing researches that improve spreadsheets has focused on extending it with abstraction mechanisms. Another study in enriching spreadsheets is implementing user-defined functions where it allows the users to input their own functions using cells, formulas and references, and no external programming languages [7, 8]. Funcalc is a prototype spreadsheet implementation of sheet-defined/user-defined functions written in C# which also serves as an extension of Corecalc. This software contains some features such as interpretative spreadsheet implementation, compilation of sheet-defined (user-defined) functions, and recursive functions and higher-order functions. As a spreadsheet model becomes more complex, it may be observed that the formulas used in the data usually appear

again and again. The Funcalc aims to simplify the function call by clustering similar formulas and replacing it with a single call to the function. With this idea, it simplifies the spreadsheet model and makes the spreadsheet more flexible, because any changes or improvement to the function will automatically affect all of its uses, unlike the traditional situation in which the users are tasked to change every instance of the function.

Spreadsheet programming is the most successful and well-known form of programming today [8]. According to researches, spreadsheet has a characteristic called “liveness” which is important to its users. Moreover, the liveness characteristic of spreadsheets can be divided into two categories; first is the direct manipulation that allows the user to edit the spreadsheet itself instead of using another program to achieve the result that the users needed— using spreadsheets will make it easier for the users because there is no difference between the actual data and the code of the spreadsheet—the second characteristic is defined as “immediate feedback”, where the users can observe immediately the change in the spreadsheet.

Despite these exciting features of spreadsheets, this does not mean that this program is exempted in experiencing problems—when in fact spreadsheets are notoriously known to be fault-prone [8]. Along with the other problems regarding spreadsheets, researchers propose features alternatives to manage these errors without diminishing directness. One of these examples is the XanaSheet which employs origin tracking techniques to maintain connection between sources and destination of copy paste actions [8]. The XanaSheet is capable of transforming and relaying the changes in the original and the other pages whenever a copied formula is modified.

Though implementation of spreadsheets in business processes has its minor drawbacks, it is still been widely used to develop for managing complex records in finance, science, engineering and education.

Spreadsheets are an extremely popular data management tool, enabling the

end users to accomplish several data tasks. The researchers sought the problem of integrating data from multiple spreadsheets. Hence, they propose Senbazuru, a prototype spreadsheet database management system, to extract information from multiple spreadsheets [9]. The system allows users to search information from a large bunch of data, to create a new version of the obtained data, and other operations such as combining relational information from several spreadsheets. The framework of the system is divided into three modules, the search, extract, and query. The search module helps the user to gather datasets via Web searchstyle relevance ranking. Similar to other search-and-rank tools, the user will type the keyword and obtains a list of spreadsheets relevant to the keyword. After selecting the most relevant, the user can use extract to convert the spreadsheet data into relational table. The last function is the query in which the user is allowed to select (filter) a recovered relation or join two spreadsheet-related relations. With the great availability of spreadsheet data, this system featuring easy spreadsheet integration will be helpful for many users.

In the field of agriculture, spreadsheet systems are still useful in managing quantitative data. PiroPinus was conceived for functional support to line-ignited, hazard reduction burning in maritime pine forests [10]. According to the study, prescribed burning is a regular component of forest management in temperate and Mediterranean regions. The efficient use of prescribed burning is based on quantitative application guidelines. The use of spreadsheet system facilitates improvement, expansion and custom modification that can be used as a modeling tool for surface fire behavior.

Educational institutions also are now utilizing spreadsheet systems to handle the records of the students compared to the traditional mode of using ledgers. University of Malta has implemented a Student Information Management System (SIMS) to provide and maintain records of the students data [11]. The system enables the students to view information regarding the course details and other online tasks. Annually, the system is responsible for facilitating spreadsheet re-

ports such as Incorrect SMO Status Report, Annual Average Marks, Student Registered Modules, and Student Module Results. The administrator is only allowed to modify the documents for documentation purposes.

Another study has focused on developing an Excel-based workbook that is easily customizable by the teachers in order to automate the scoring procedure for reports or presentations in any particular course [12]. The workbook includes an editable rubric for the teachers, a macro that creates an individual, labeled worksheet for each student, drop-down menus to select the level for each report component, displaying the total grades with an option for late submission and a complete list of students for submission to a course management system.

Nowadays, online spreadsheets are becoming a trend to most end users, especially to teachers, since it is capable of managing the student records and attendance through the Internet. A study in Malaysia has discussed the potentials of the Google Spreadsheets, some of which is providing and maintaining the grades of the students [13]. The online characteristic of Google Spreadsheets removes the fear of the teachers in losing the document because the document is not stored locally. Another advantage of online spreadsheets, compared to offline forms of spreadsheets, is their capabilities to publish the data directly through generated URL or embed it in a blog or Web site.

III. Theoretical Framework

A. Spreadsheet

At the most basic level of spreadsheet implementation, a spreadsheet is simply defined as a composition of cells [5]. It is visually represented as a two-dimensional grid of cells in which the users are enabled to store and process information. These cells may contain specified values (numbers and strings), references to other cells, or a formula [3]. The cells are classified according to the purpose of its value and described as follows:

- Cells with constant values that are not referenced by other cells are essentially labels. The purpose of these cells is mainly used for documentation, and to provide information about the contents of the spreadsheets.
- On the contrary, cells that contain values that are referenced by other cells serve as the inputs for processing the information in the spreadsheet.
- Cells that contain formulas can be considered as intermediate computations or output cells. Intermediate computation cells contain formulas that are referenced by other cell. Output cells contain formulas which are not referenced by other cells thus serving as an output of the spreadsheet.

As mentioned, the system also makes use of formulas for more efficient operation on the spreadsheet data. A formula can involve constants, basic arithmetic operations, functions, and references to other cells. With complex spreadsheets, these formulas that consist of functions can be defined over cell ranges [6]. For example, the formula `SUM(A1:A5)` adds the values in cells A1, A2, A3, A4, and A5. In addition, formulas that can be utilized has increased since spreadsheet systems have offered hundred of predefined functions and some of those also allows user to create their own functions based on their needs. Most of these programs implement standard mathematical functions such as `EXP`, `LOG` and `SIN`, statistical functions such as `MEDIAN` and probability distributions, functions to generate pseudo-random

number, functions to manipulate time and data, and much more. In this type of software where processing information is utmost important, a method of reference is needed to obtain the contents of the cells. Most spreadsheet systems have implemented several types of cell addresses [3]. Some of the most common cell addressing schemes is A1-style addressing and R1C1- style addressing. In A1-style addressing scheme, the user can reference a cell by its row and column position within the spreadsheet file. For example, a cell at row 8 and column A can be referenced with the address A8. In R1C1-style addressing, the cell address consists of the letter R, followed by the row number, and letter C and its corresponding column number. A cell at row 8 and column 1 is addressed as R8C1. Types of cell referencing can be also distinguished by absolute, relative or the combination of two. By default, relative reference has been used in copying formulas from one cell to another. In this type of reference, the formula changes as it is copied to a new cell. For example, the formula SUM(B5:B8), changes to SUM(C5:C8) when copied across to the next column.

	A	B	C
1	Smith's Bo		
2	Septemb		
3			
4	Region	Books	Periodicals
5	North	\$15,678	\$8,796
6	South	\$13,865	\$9,776
7	East	\$25,401	\$11,392
8	West	\$18,552	\$4,928
9	Product Total	\$73,496	

	A	B	C
1	Smith's Bo		
2	Septemb		
3			
4	Region	Books	Periodicals
5	North	\$15,678	\$8,796
6	South	\$13,865	\$9,776
7	East	\$25,401	\$11,392
8	West	\$18,552	\$4,928
9	Product Total	\$73,496	\$34,892

Figure 1: An example of relative reference

On the contrary, absolute referencing is used if there is a value that must remain constant in the formula. Dollar signs (\$) are used to indicate a column/row reference constant. In some cases like computing for commissions, the user will input the commission rate and it should remain the same throughout the computation. As an example, the commission rate is at column B and row 10. In absolute referencing, the cell can be addressed as B10. Copying formulas with Figure 1.1 An example of Relative Reference absolute references will not change

the value of the row or column unlike in relative reference.

	A	B	C
1	Commissions - November 2000		
2			
3		Total Sales	Commission
4	Bob	26,000	=B4*\$B\$10
5	Sally	35,350	
6	Joseph	42,000	
7	Celia	28,800	
8			
9			
10	Rate:	10%	

	A	B	C
1	Commissions - November 2000		
2			
3		Tot:	
4	Bob	26,000	2600
5	Sally	35,350	3535
6	Joseph	42,000	4200
7	Celia	28,800	2880
8			
9			
10	Rate:	10%	

Figure 2: An example of absolute reference

Obviously, implementation of spreadsheet relies on the concept of dependencies between the cells. A cell may exactly depend on values of the other cells and it can also directly support other cells. Figure 3 shows an example of a spreadsheet using basic arithmetic operation.

	A	B	C	D	E	F
1		Count	Weight	Total weight	Count %	
2	Crowbars	5	7	35	22.73	
3	Screwdrivers	11	2	22	50	
4	Hammers	6	3	18	27.27	
5	Sum	22		75		
6						
7		Average	3.41			
8						

Figure 3: Spreadsheet window displaying results of computation

Figure 4 shows that the values of cells E2, E3, and E4 directly depend on the values of cells B2, B3, B4, and B5. Also with the values of cells D2, D3, and D4 depends on both the values from column B and C. Figure 1.4 shows the basic process of the spreadsheet example in figure 1.3. In the graph, the nodes represent the cells and the edges represent the dependencies between the cells.

Another important process in the implementation of spreadsheet is the method of recalculation. When the value of a specified cell is modified, all other cells that are supported by the edited cell must be recalculated. The design of the recalculation mechanism is important to the efficiency and reliability of the spreadsheet implementation [3]. An approach on the recalculation mechanism is to visit every

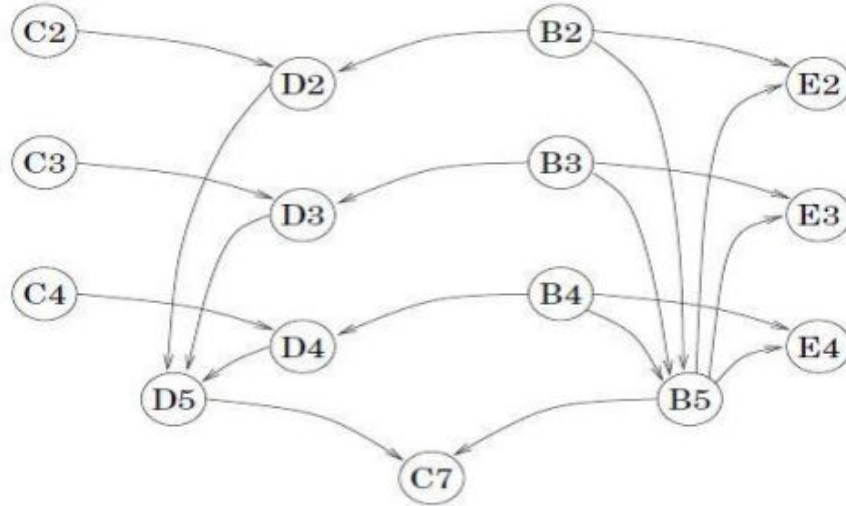


Figure 4: Graph-oriented view of cell dependency

active cell and evaluate its formula if needed. This approach has been applied in Corecalc to ensure that each formula is evaluated at most every recalculation

B. Class Record

Class record is a document that keeps the record of the class performance of the students. It serves as a basis of the teacher for the students progress specifically on monitoring the students projects, assignments, and test scores. Developing a good class record will be helpful in organizing, processing, and communicating the students understanding of the curriculum. With the improvement of technology, manual class recording have been improved into an electronic form of handling the student records [14]. Such spreadsheet applications like Microsoft Excel, OpenOffice Calc have been popular for the teachers on monitoring the performance of the students. These forms of electronic spreadsheets are capable of storing, processing and presenting records more simple and efficient compared to its traditional method.

C. Grading System

A grading system is the scale used by a university or an educational institution in measuring the performance of the students throughout the assigned class duration. Various schools and universities in the Philippines have followed different grading scales. Most of the universities in the country, like University of the Philippines (UP) [15] and Polytechnic University of the Philippines (PUP) [16], set 1.00 as the highest grade with 5.00 as the lowest one. Other schools like Ateneo De Manila University (ADMU) [17] and De La Salle University (DLSU) [18] adheres to the grading system patterned to the United States grading standard with 4.00 as the highest mark and 1.00 as the lowest.

Grade/Mark	Percentage/Equivalent	Description
1.0	97-100	Excellent
1.25	94-96	Excellent
1.5	91-93	Very Good
1.75	88-90	Very Good
2.0	85-87	Good
2.25	82-84	Good
2.5	79-81	Satisfactory
2.75	76-78	Satisfactory
3.0	75	Passing
5.0	65-74	Failure
Inc.		Incomplete
W		Withdrawn
D		Dropped

Figure 5: PUP Grading Scale

Grade Point	Description
4.0	Excellent
3.5	Superior
3.0	Very Good
2.5	Good
2.0	Satisfactory
1.5	Fair
1.0	Passed
0.0	Failed
Audit	Audit
6.5	Withdrawn
7.0	Passed (For Pass/Fail Courses)
8.0	Failed (For Pass/Fail Courses)
9.9	Incomplete/Deferred (for Thesis/Practicum courses only)

Figure 6: DLSU Grading System

The percentage equivalent of the grading scale can be different depending on the course set by the university or departments in which the class belongs.

D. Grade Aggregation

Grade aggregation is the method of how scores are computed to produce the final grades of the students [19]. The system uses three types of grading methods. The three methods are natural aggregate, mean of grades, and weighted mean of grades.

- Natural Aggregate - This method is implemented by adding the score divided by the maximum score possible.

A1: 70/100
 A2: 20/80
 A3: 10/10
 Earned: 70 + 20 + 10 = 100
 Maximum: 100 + 80 + 10 = 190

 Aggregated grade: 100/190

Figure 7: Natural Aggregation Method

- Mean of Grades - The final score is obtained by adding the score of the n items divided by the total number of items.

$$\frac{1}{n} \sum_{i=1}^n a_i = \frac{1}{n} (a_1 + a_2 + \cdots + a_n)$$

- Weighted Mean of Grades - The final score is obtained by the summation of product of weights (w) and scores (x) divided by the summation of weights (w).

$$\frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i} = \frac{w_1 x_1 + w_2 x_2 + \cdots + w_n x_n}{w_1 + w_2 + \cdots + w_n}$$

E. Information System

An information system is defined as the software that is helpful in managing and interpreting data. It is a set of components for collecting, storing and processing data. This type of software aims to deliver information, knowledge, and digital products in which it also follows a specified business rules. Another purpose of information system is to transform raw data into useful information that can be used for decision making in organizations [20].

F. Database Management System (DBMS)

Database management system or DBMS refers to the system of storing and retrieving users data with maximum efficiency along with appropriate security measures. The system allows users to create, retrieve, update, and manage data in a more systematic manner [21]. It also serves as an interface between the database, and the application programs, ensuring that the data remains organized and easily accessible.

The use of DBMS for storing and managing data has its benefits. One of the advantages of utilizing DBMS is that it let the end users and programmers to access the data while managing data integrity. Instead of creating multiple copies of the data in case of several applications accessing it, the DBMS is capable of providing a central storage of data that can be accessed by multiple users. Another advantage of utilizing DBMS is that it can be used to impose a logical organization of data [22].

G. CodeIgniter

Created by Rick Ellis in 2006, CodeIgniter is an open-source PHP framework. The framework implemented a modified version of the Model-View-Controller (MVC) design pattern. It is a simple and elegant toolkit enabling rapid development of both Web sites and applications. It provides an extensive set of PHP libraries, as well as a simple interface and logical structure in accessing these libraries [23]. This open-source framework has capabilities, from compatibility to almost all hosting platforms, simple and efficient database connection, stable configuration, clear technical documentation, addition of third-party tools or plugins, and much more has attracted thousands of PHP developers

H. Bootstrap

Bootstrap is currently the most popular front end framework for developing responsive, mobile first projects on the web. It is an HTML, CSS and JS framework that contains extensive and beautiful documentation for common HTML elements. It also consists of numerous HTML and CSS components with additional jQuery plug-ins to improve front end web development faster and easier. With CSS media queries, this framework has been useful for the developers in creating responsive Web sites for the users can able to view the sites in any Web browsers or devices [24].

IV. Design and Implementation

A. Use Case Diagram

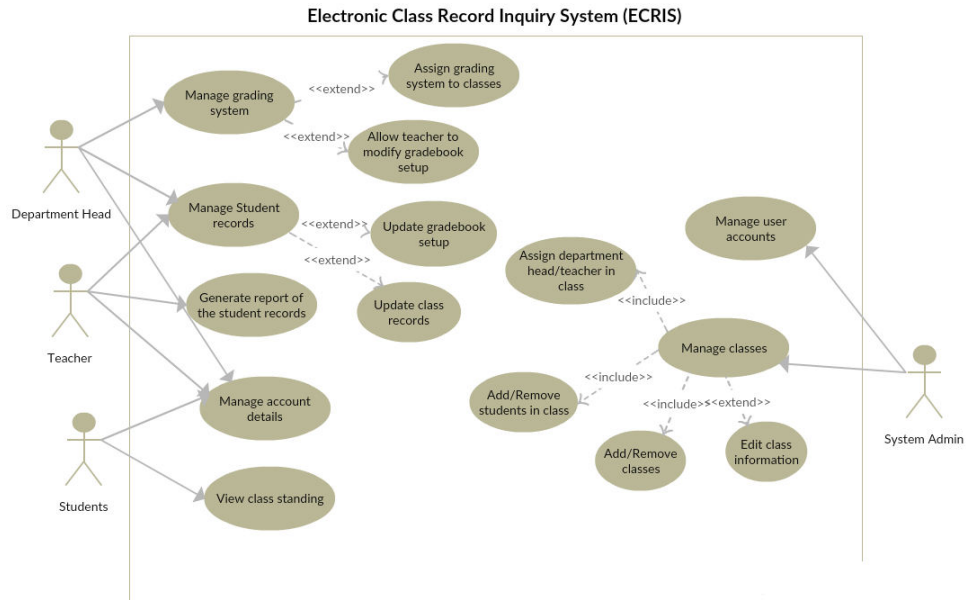


Figure 8: Top-level use case diagram of ECRIS

Primarily, the school administrator of every school is responsible for the management of the system. Before students, teachers, and the department heads are allowed to access the system, e-mail verification will be sent to their respective e-mail addresses for account validation. The system administrator should ensure that the students are enrolled. It is also the responsibility of the system administrator to provide the class details to the teachers and students. Upon approval of the account request, the teachers can update and modify the class record. The department heads are allowed to modify the gradebook setup that will be used for the class record. In addition, they also allow the teachers to modify the gradebook setup. Students enlisted in the course can view their student class records as updated by their teachers.

B. Activity Diagrams

Figure 9 shows that students are allowed to view his/her class standing in a specific course. The system will display all of the courses in which the student is enrolled and allows the student to view the selected class record.

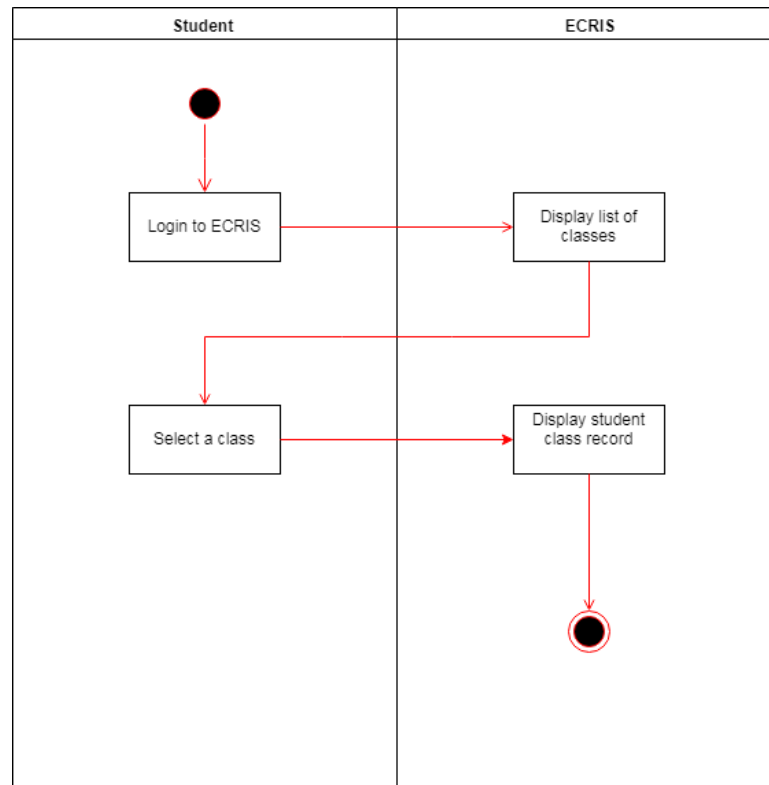


Figure 9: Activity Diagram - ECRIS View Student Record

Figure 10 shows the activity diagram for updating the student class records. The teachers can select from their assigned courses and modify the student class records.

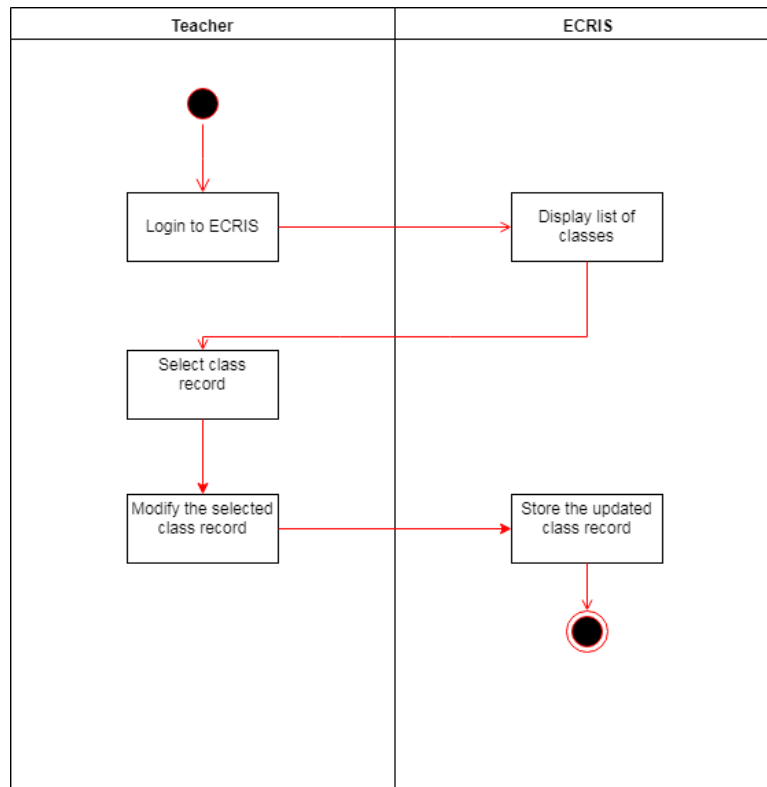


Figure 10: Activity Diagram - ECRIS Modify Course Record

The department head is assigned to set the grading scale for the gradebook setup. Figure 11 shows that the assigned department head can update the grading scale of the selected course.

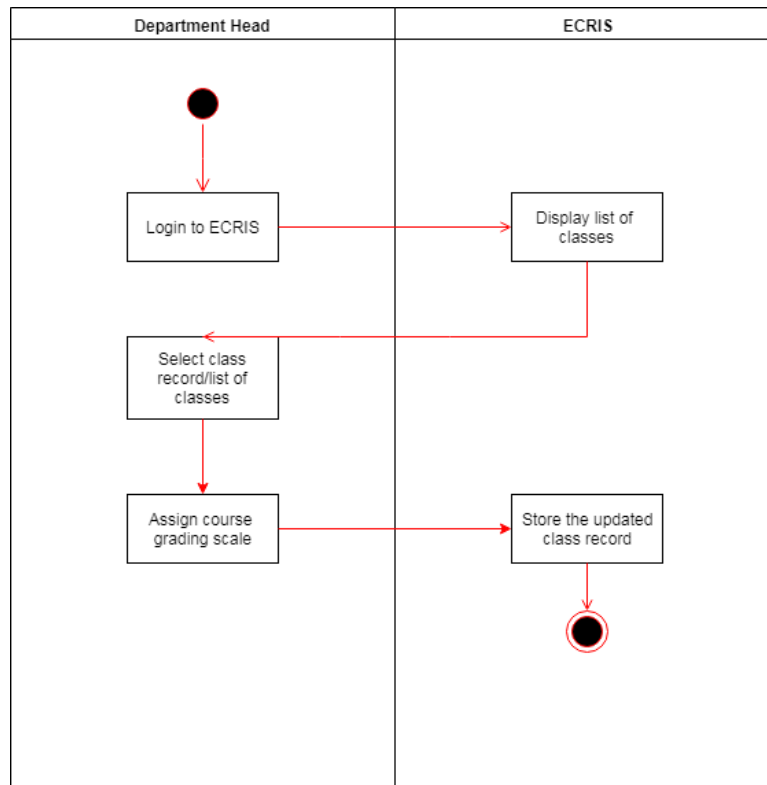


Figure 11: Activity Diagram - Assign Grading Scale

The activity diagram (figure 12) shows that the system administrator is allowed to modify the course information such as course name and section. In addition, they can also reassign department heads, teachers and students in a class record if needed.

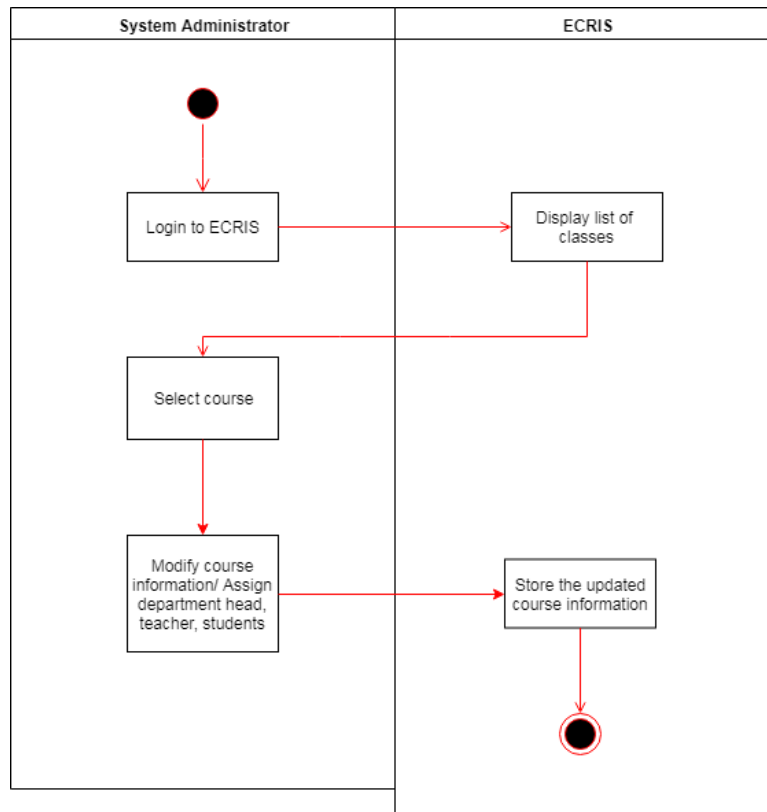


Figure 12: Activity Diagram - Assign Grading Scale

C. Entity Relationship Diagram

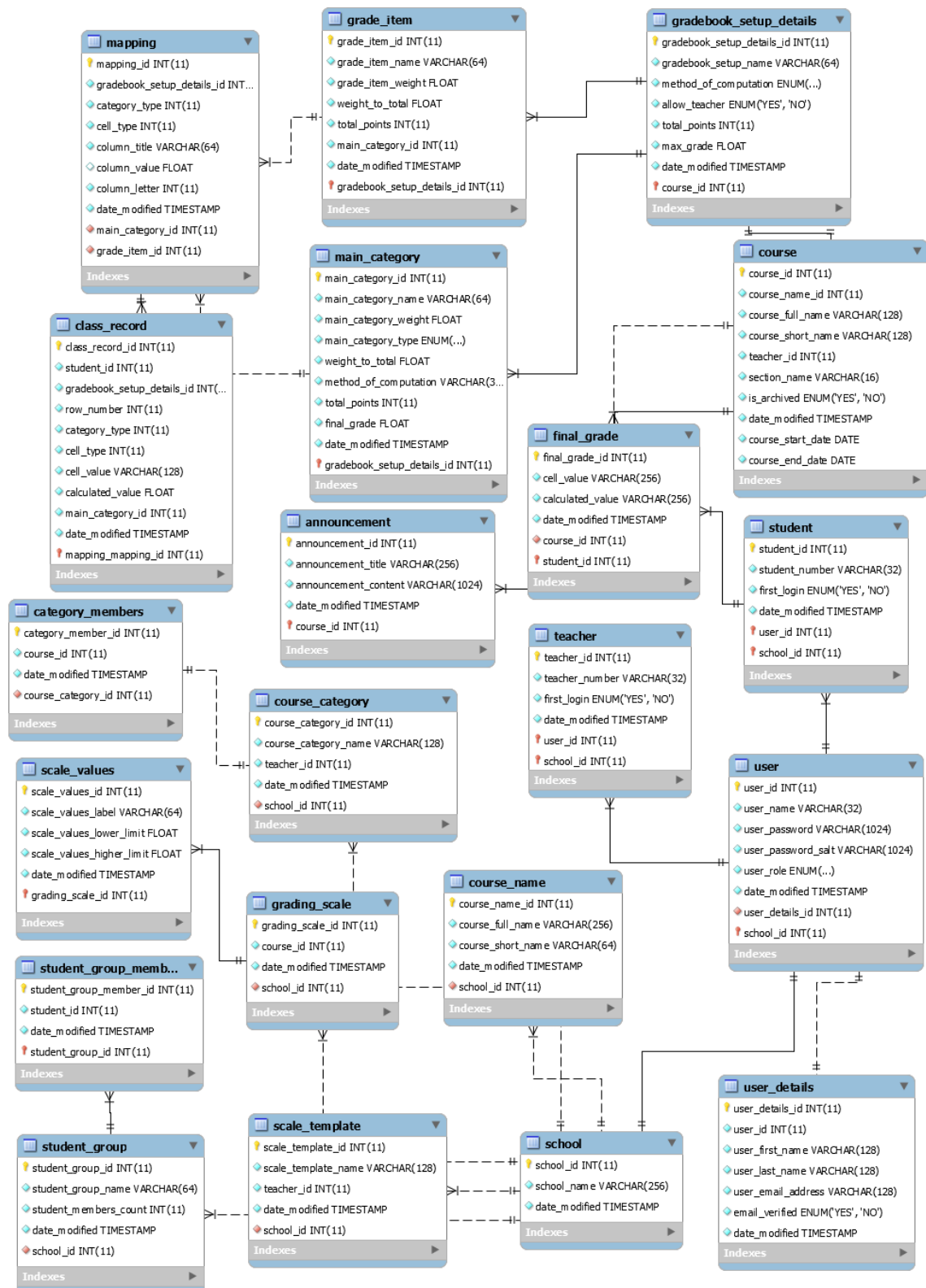


Figure 13: Entity Relationship Diagram - ECRIS

Figure 13 shows the entity relationship diagram of ECRIS. All of the information regarding the user accounts and courses are stored in a database. The school table contains the names of the school that are registered in the system. From that, every school has different users and courses.

In order to fully implement an electronic class record, the system also needs to store information about the course and the gradebook setup that will be used. The course/class details such as the course name, section name are stored in the course table.

Other information such as the gradebook scale and announcements is also stored in the database. The values in the spreadsheet is stored in the class record table and mapping table.

D. Database Design

Column	Type	Description
announcement_id	int	unique id of the announcement
course_id	int	unique id of the course
announcement_title	varchar	title of the announcement
announcement_content	varchar	content of the announcement
date_modified	timestamp	time stamp of the last modification

Table 1: announcement table

Column	Type	Description
category_member_id	int	unique id of the category member
course_id	int	unique id of the course
course_category_id	int	unique id of the course category
date_modified	timestamp	time stamp of the last modification

Table 2: category members table

Column	Type	Description
class_record_id	int	unique id of the class record
student_id	int	unique id of the student
gradebook_setup_details_id	int	unique id of the gradebook setup
row_number	int	order number of the student in the class record
mapping_id	int	unique id of the mapping
category_type	int	represent whether gradebook items or categories; 1 for category, 2 for items
cell_type	int	represent whether the cell type is a single value or computer by formula; 1 for sub, 2 for total
cell_value	varchar	the value assigned to a cell
calculated_value	float	result of the formulas assigned to the cell
main_category_id	int	unique id of the main category
date_modified	timestamp	time stamp of the last modification

Table 3: class record table

Column	Type	Description
course_id	int	unique id of the course
school_id	int	unique id of the school
course_name_id	int	unique id of the course name
course_full_name	varchar	the full name given to a course
course_short_name	varchar	short name assigned to a course
teacher_id	int	unique id of the teacher
section_name	varchar	unique section name for course
course_start_date	date	start day of the course
course_end_date	date	end date of the course
date_modified	timestamp	time stamp of the last modification

Table 4: course table

Column	Type	Description
course_category_id	int	unique id of the course category
school_id	int	unique id of the school
course_category_name	varchar	name of the course category
teacher_id	int	unique id of the teacher
date_modified	timestamp	time stamp of the last modification

Table 5: course category table

Column	Type	Description
course_name_id	int	unique id of the course name
school_id	int	unique id of the school
course_full_name	varchar	full name of the course
course_short_name	varchar	short name of the course
date_modified	timestamp	time stamp of the last modification

Table 6: course name table

Column	Type	Description
course_student_id	int	unique id of the course student
student_id	int	unique id of the student
course_id	int	unique id of the course
is_dropped	enum('YES', 'NO')	status of the student in the course
order_number	int	number assigned to a student based on alphabetical manner
date_modified	timestamp	time stamp of the last modification

Table 7: course students table

Column	Type	Description
final_grade_id	int	unique id of the final grade
student_id	int	unique id of the student
course_id	int	unique id of the course
cell_value	varchar	value of the cell in the record
calculated_value	varchar	values based on formulas assigned to the cell
date_modified	timestamp	time stamp of the last modification

Table 8: final grade table

Column	Type	Description
grade_item_id	int	unique id of the grade item
gradebook_setup_details_id	int	unique id of the gradebook setup
grade_item_name	varchar	name assigned to a grade item
grade_item_weight	float	percentage of score in the category
weight_to_total	float	percentage of score in the gradebook setup
main_category_id	int	unique id of the main category
date_modified	timestamp	time stamp of the last modification

Table 9: grade item table

Column	Type	Description
gradebook_setup_details_id	int	unique id of the gradebook setup
course_id	int	unique id of the course
gradebook_setup_name	varchar	name of the gradebook setup
method_of_computation	enum('Mean of Grades', 'Weighted Mean of Grades', 'Natural Aggregate')	grading method for the whole gradebook
allow_teacher	enum('YES', 'NO')	setting that allow the teacher to modify the gradebook
total_points	int	maximum score of the items
max_grade	float	maximum grade that can be given to the students
date_modified	timestamp	time stamp of the last modification

Table 10: gradebook setup details table

Column	Type	Description
gradebook_scale_id	int	unique id of the gradebook scale
school_id	int	unique id of the school
course_id	int	unique id of the course
date_modified	timestamp	time stamp of the last modification

Table 11: gradebook scale table

Column	Type	Description
main_category_id	int	unique id of the main category
gradebook_setup_details_id	int	unique id of the gradebook setup
main_category_name	varchar	name of the main category
main_category_weight	float	weight of the category as assigned by the gradebook setup
main_category_type	enum('CATEGORY', 'ITEM')	category types; categories with 'ITEM' types do not have subitems.
weight_to_total	float	percentage of the weight to the whole gradebook setup
method_of_computation	varchar	grading method used in the category
total_points	int	total points of the category
final_grade	float	maximum grade that can be obtained from the category
date_modified	timestamp	time stamp of the last modification

Table 12: main category table

Column	Type	Description
mapping_id	int	unique id of the mapping table
gradebook_setup_details_id	int	unique id of the gradebook setup
category_type	int	represent whether gradebook items or categories; 1 for category, 2 for items
cell_type	int	represent whether the cell type is a single value or computer by formula; 1 for sub, 2 for total
main_category_id	int	unique id for the main category
grade_item_id	int	unique id for the grade item
column_title	varchar	header of the column in the spreadsheet
column_value	int	the value of the cell in the spreadsheet
column_letter	float	column assigned to the cell
date_modified	timestamp	time stamp of the last modification

Table 13: mapping table

Column	Type	Description
scale_template_id	int	unique id of the scale template
school_id	int	unique id of the school
scale_template_name	varchar	name of the scale template
teacher_id	int	unique id for the teacher
date_modified	timestamp	time stamp of the last modification

Table 14: scale template table

Column	Type	Description
scale_template_members_id	int	unique id of the scale template members
scale_template_id	int	unique id of the scale template
scale_values_id	int	unique id of the scale values
date_modified	timestamp	time stamp of the last modification

Table 15: scale template members table

Column	Type	Description
scale_values_id	int	unique id of the scale values
grading_scale_id	int	unique id of the grading scale
scale_values_label	varchar	name set for the scale value
scale_values_lower_limit	float	lower limit of the scale value
scale_values_higher_limit	float	higher limit of the scale value
date_modified	timestamp	time stamp of the last modification

Table 16: scale values table

Column	Type	Description
school_id	int	unique id of the school
school_name	varchar	name of the school
date_modified	timestamp	time stamp of the last modification

Table 17: school table

Column	Type	Description
student_id	int	unique id of the student
school_id	int	unique id of the school
user_id	int	unique id of the user
student_number	varchar	assigned student number
first_login	enum('YES', 'NO')	checker if the account has logged in for the first time
date_modified	timestamp	time stamp of the last modification

Table 18: student table

Column	Type	Description
student_group_id	int	unique id of the student group
school_id	int	unique id of the school
student_group_name	varchar	name of the student group
student_members_count	int	number of student assigned in the student group
date_modified	timestamp	time stamp of the last modification

Table 19: student group table

Column	Type	Description
student_group_member_id	int	unique id of the student group member
student_group_id	int	unique id of the student group
student_id	int	unique id of the student
date_modified	timestamp	time stamp of the last modification

Table 20: student group member table

Column	Type	Description
teacher_id	int	unique id of the teacher
school_id	int	unique id of the school
user_id	int	unique id of the user
teacher_number	varchar	assigned teacher number
first_login	enum('YES', 'NO')	checker if the account has logged in for the first time
date_modified	timestamp	time stamp of the last modification

Table 21: teacher table

Column	Type	Description
user_id	int	unique id of the user
school_id	int	unique id of the school
user_name	varchar	user name of the account
user_password	varchar	password of the account
user_password_salt	varchar	password salt
user_role	enum('TEACHER', 'ADMIN', 'STUDENT')	user role of the account
date_modified	timestamp	time stamp of the last modification

Table 22: user table

Column	Type	Description
user_details_id	int	unique id of the user details
user_id	int	unique id of the user
user_first_name	varchar	user's first name
user_last_name	varchar	user's last name
user_email_address	varchar	user's email address
email_verified	enum('YES', 'NO')	checker if the email address is verified
date_modified	timestamp	time stamp of the last modification

Table 23: user details table

V. Results

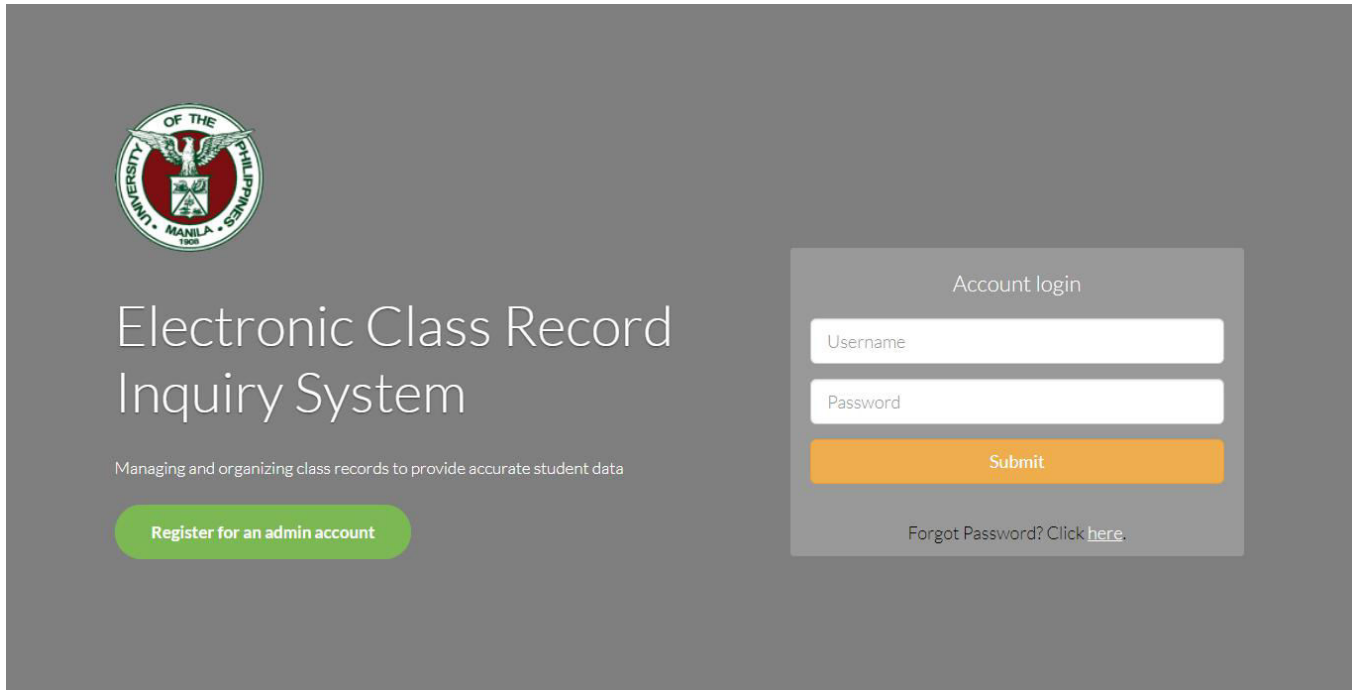


Figure 14: Login page

In order to use ECRIS, all system users must navigate through the login page of the site. For creation of system administrator accounts, navigate to the register school admin page by clicking the green button which shown in figure 14.

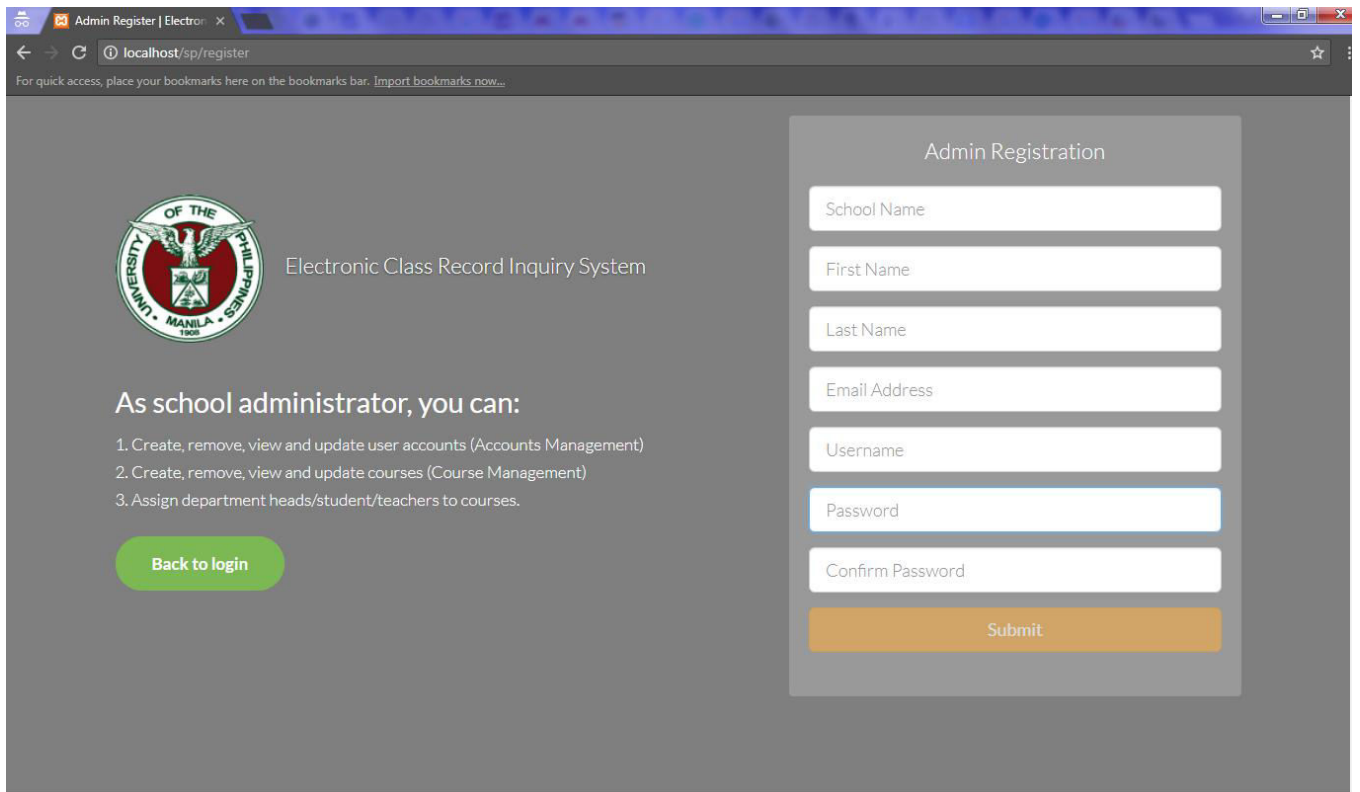


Figure 15: Register School Admin page

System administrators will register through the form as shown in figure 15. The user has to complete the form that consists of the school name, first name, last name, e-mail address, desired username, and desired password.



Figure 16: Account Verification e-mail

Upon successful registration, the system administrator will receive an e-mail to verify his/her account in the system as shown in figure 16 . System user accounts that will be created by the system administrators be required to submit their e-mail addresses for account verification.

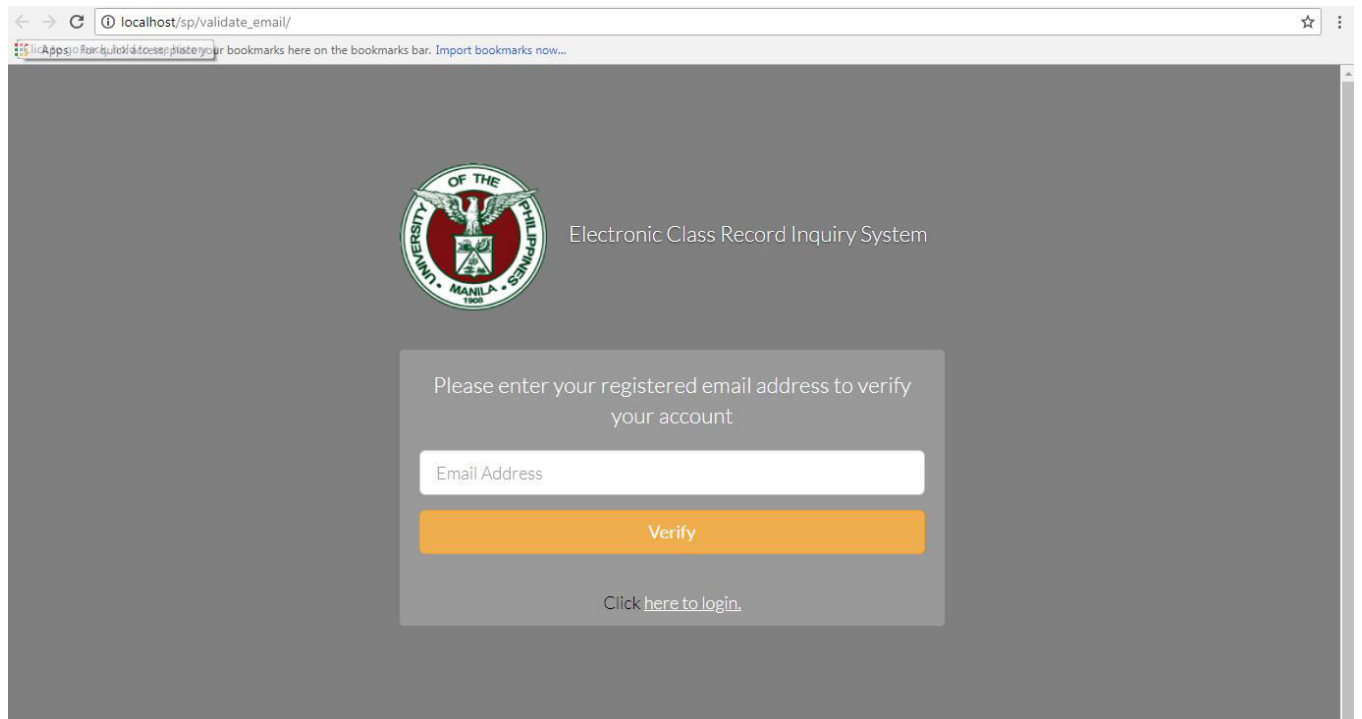


Figure 17: Verify e-mail Form

Clicking the link in the account verification e-mail will open a new tab and redirect the users to an input form in which they are required to submit the e-mail addresses that they have used for the creation of account.

System administrators can now directly log in to the system with the username and password that they have submitted in the “Register Admin” page. For students and teachers, they need to submit the required e-mail address and they will be redirected to the “verify success” page which displays the default username and password.

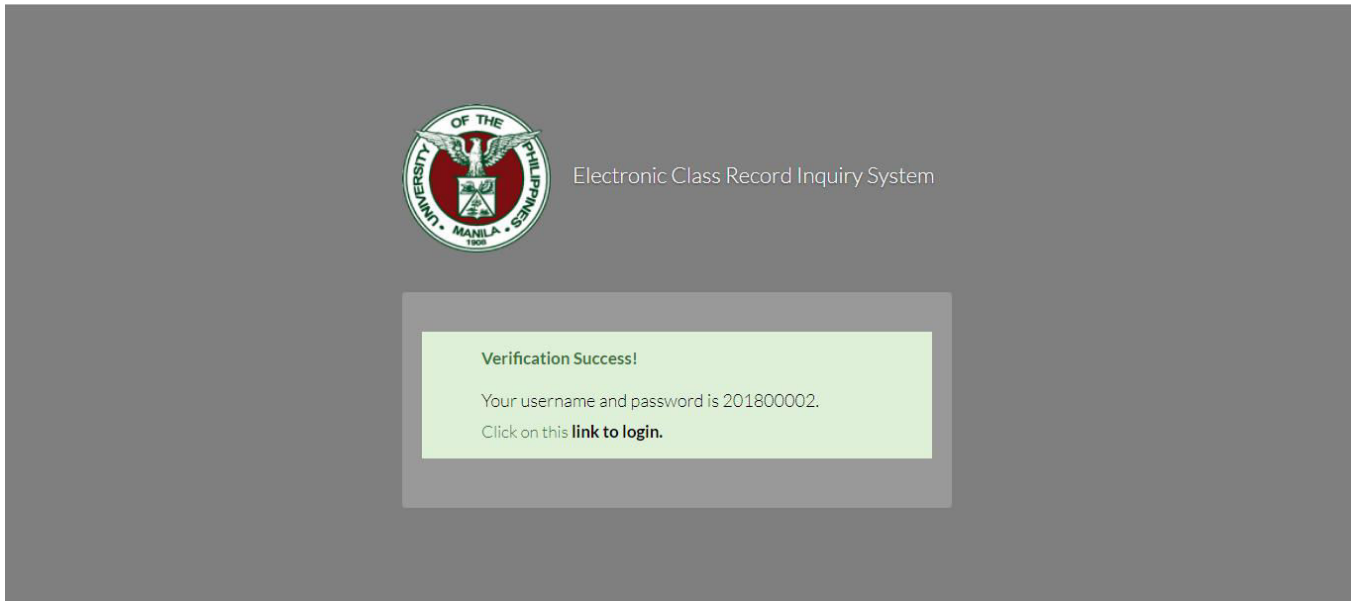


Figure 18: Verify e-mail Success

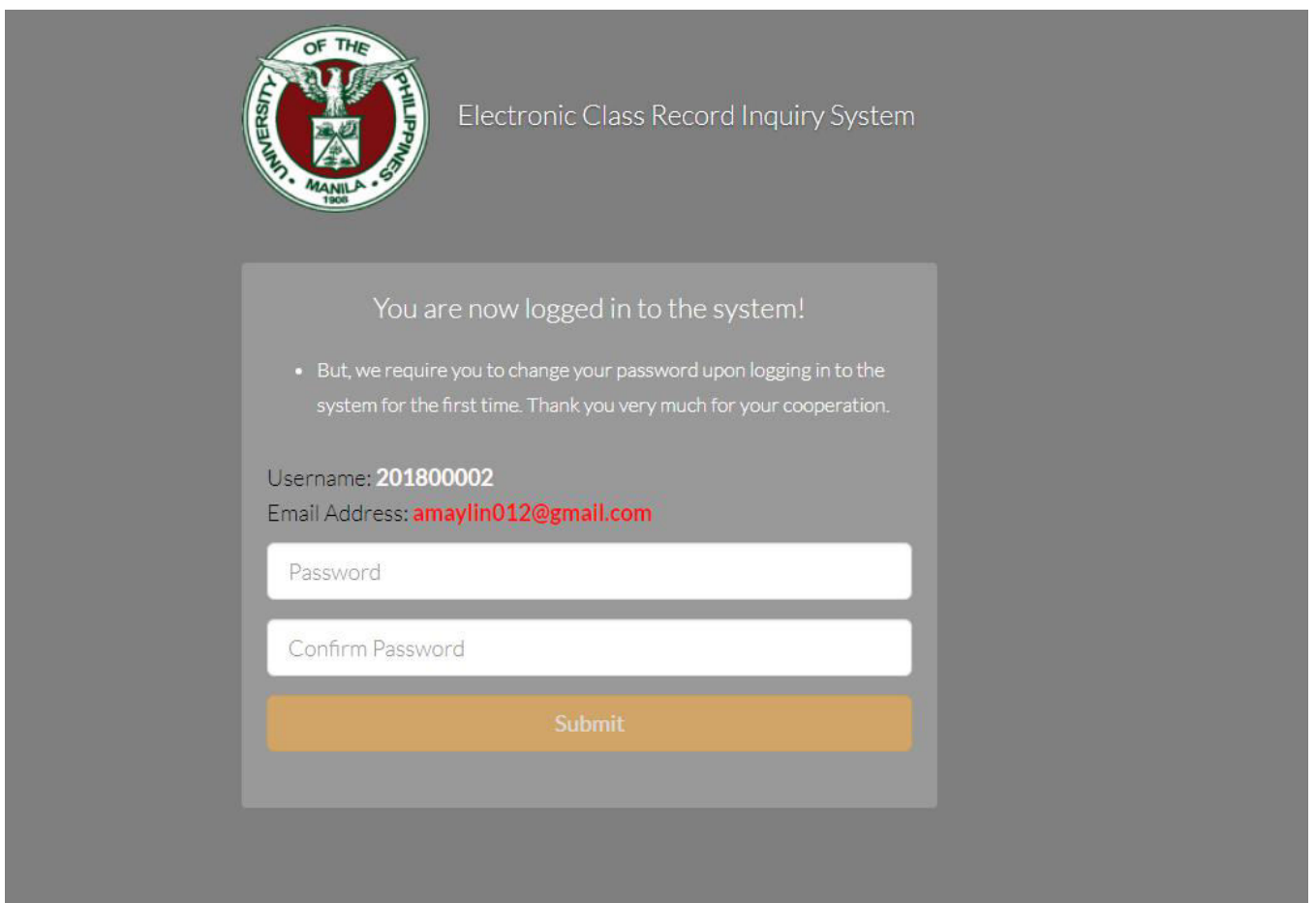


Figure 19: First Login - Forced Change Password Form

Upon the creation of account, the system will set the username and password

with the same value. After figure 18, the teachers and students are now able to log in to the system with the displayed username and password. To improve the account security, the system forces the teachers and students to change their password upon logging in to the site for the first time as shown in figure 19.

In the home page/dashboard for system administrators, there are several links that allows him to navigate throughout the system. As a system administrator, he/she is responsible for the creation, modification and deletion of the user accounts and courses in the system.

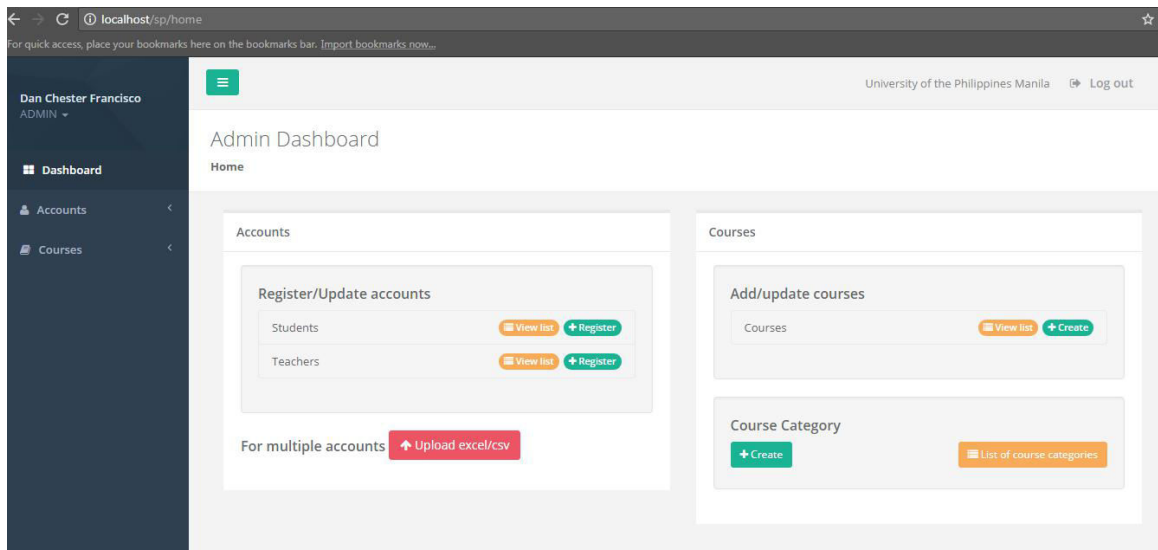


Figure 20: Home Page - System Administrator

As shown in figure 20, the system administrator can easily register user accounts and create new courses by clicking the green “+ Register” and “+ Create” link. After clicking the “+ Register” link in either of the teacher row or student row, the system administrator will be redirected to the register user account page for teacher or student.

The screenshot shows the 'Add student/s' page. It features a dark sidebar with navigation options: Dashboard, Accounts (with sub-options for Student, Add new student/s, View list of students, Student Groups, Teacher, and Upload accounts), and Courses. The main content area has a breadcrumb trail: Home / Accounts / Students / Add new student/s. The form contains three input fields: First Name, Last Name, and Email Address. Below these is a section titled 'Select courses from the list on the left table in which the student should be assigned'. It includes 'Select All' and 'Deselect All' buttons. A 'List of courses' box contains two items: 'CMSC 11 Lab1A - Course ID: 2' and 'CMSC 11 Lect1A - Course ID: 1'. An arrow points to an empty 'Selected courses' box. At the bottom are 'Cancel' and 'Submit' buttons.

Figure 21: Register User Account - System Administrator

As stated earlier, an e-mail will be sent to the e-mail address of the student or teacher after the account has been registered to verify the account. The student or teacher should verify their user account before logging in to the system.

The screenshot shows the 'View list of students' page. The header includes the user name 'Dan Chester Francisco ADMIN' and 'University of the Philippines Manila' with a 'Log out' button. The breadcrumb trail is Home / Student / View list of students. Below the header is a table titled 'List of all students in University of the Philippines Manila'. The table has columns for Student Last Name, Student First Name, Email Address, Verified?, and Actions. There are 10 rows of student data, all with a 'Not yet verified' status. The 'Actions' column contains 'Edit' and 'Delete' buttons for each student.

Student Last Name	Student First Name	Email Address	Verified?	Actions
Amiranda	Tadeo	tamirandac@plala.or.jp	Not yet verified	Edit Delete
Antonellini	Gard	gantonnellinie@myspace.com	Not yet verified	Edit Delete
Baldcock	Rudolph	rbaldcock@yellowbook.com	Not yet verified	Edit Delete
Baseley	Danila	dbaseleyl@alexa.com	Not yet verified	Edit Delete
Belin	Ashton	abelinp@msu.edu	Not yet verified	Edit Delete
Besnard	Westley	wbesnard1b@ox.ac.uk	Not yet verified	Edit Delete
Bett	Harrison	hbett19@devhub.com	Not yet verified	Edit Delete
Bitterton	Tyrone	tbitterton9@hexun.com	Not yet verified	Edit Delete
Borell	Jeri	jborellx@trellian.com	Not yet verified	Edit Delete

Figure 22: List of user accounts - System Administrator

Upon successful registration by the system administration, he/she will be redirected to the list of registered teacher/student accounts. As shown in figure 22, the information regarding the users such as first name, last name, and e-mail address. In this page, the system also allows the administrator to edit or remove the user accounts.

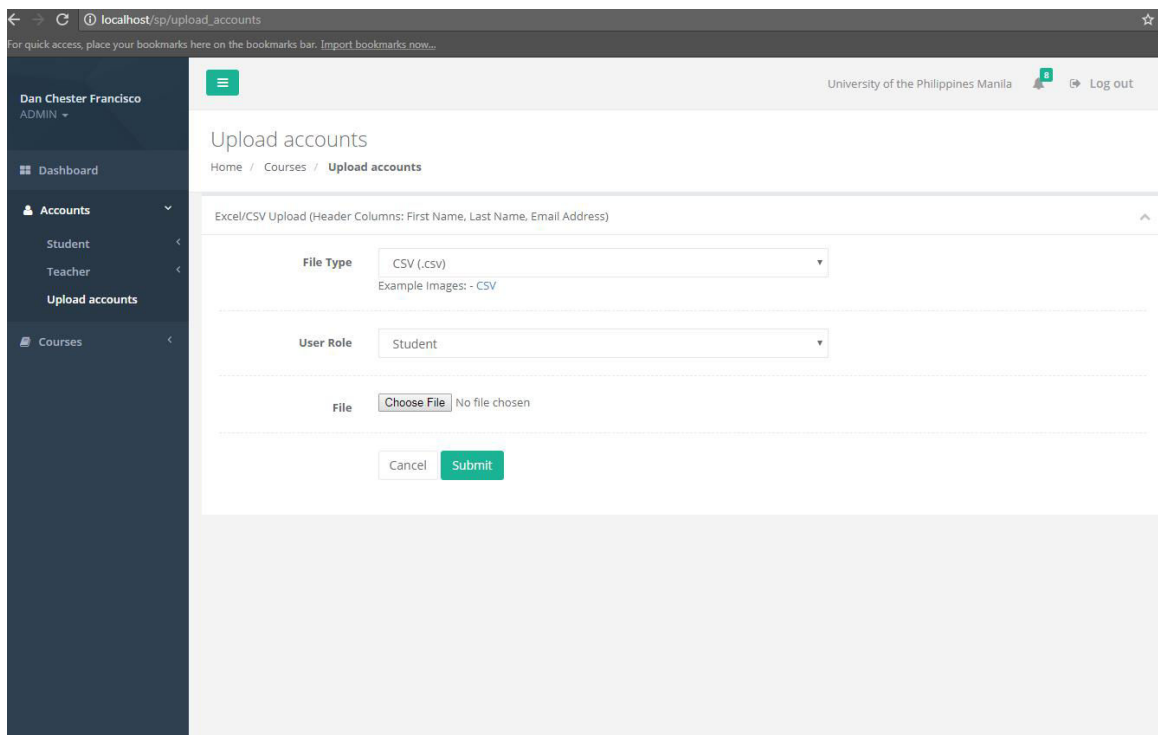


Figure 23: Upload multiple accounts - System Administrator

The system also allows bulk insert of user accounts in the database. By clicking the red "Upload csv" button as shows in figure 20, the system admin will be able to upload multiple user accounts by uploading csv files. By selecting whether the user is a student or teacher, the system admin can easily register multiple user accounts.

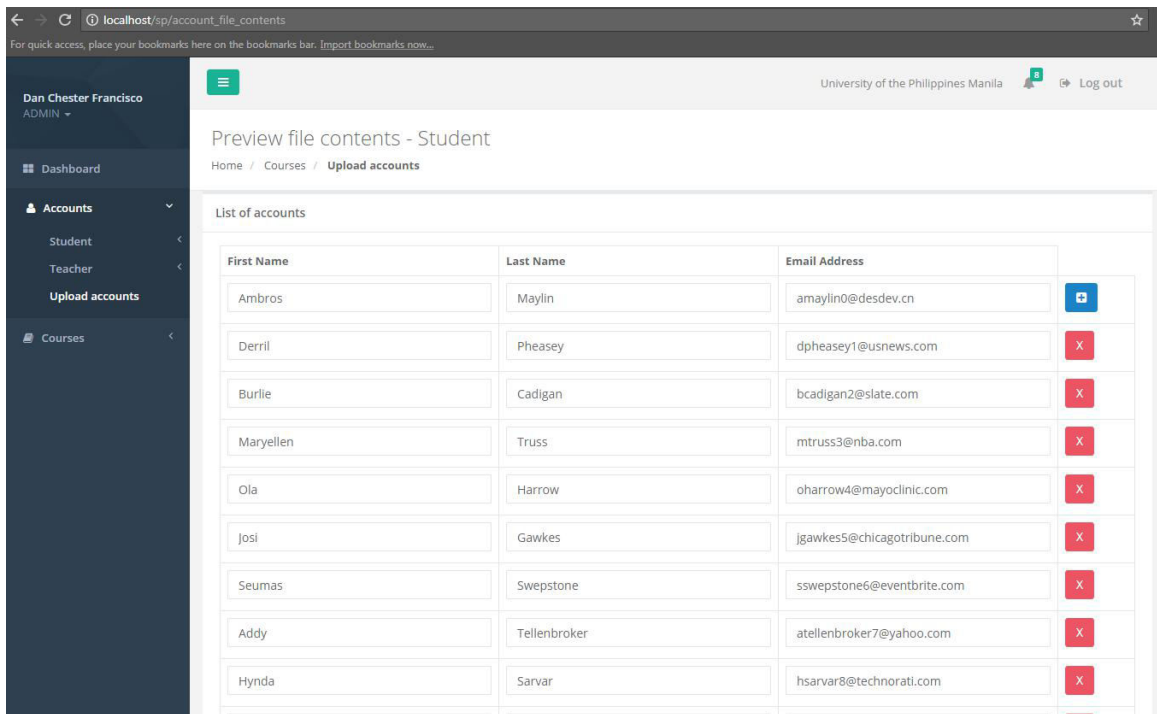


Figure 24: CSV File Preview - System Administrator

The system admin can inspect the information of the csv file before being inserted into the database as shown in figure 24. e-mail addresses that are already existing in the database will not be inserted. An e-mail message will be sent to the e-mail addresses for the verification of accounts.

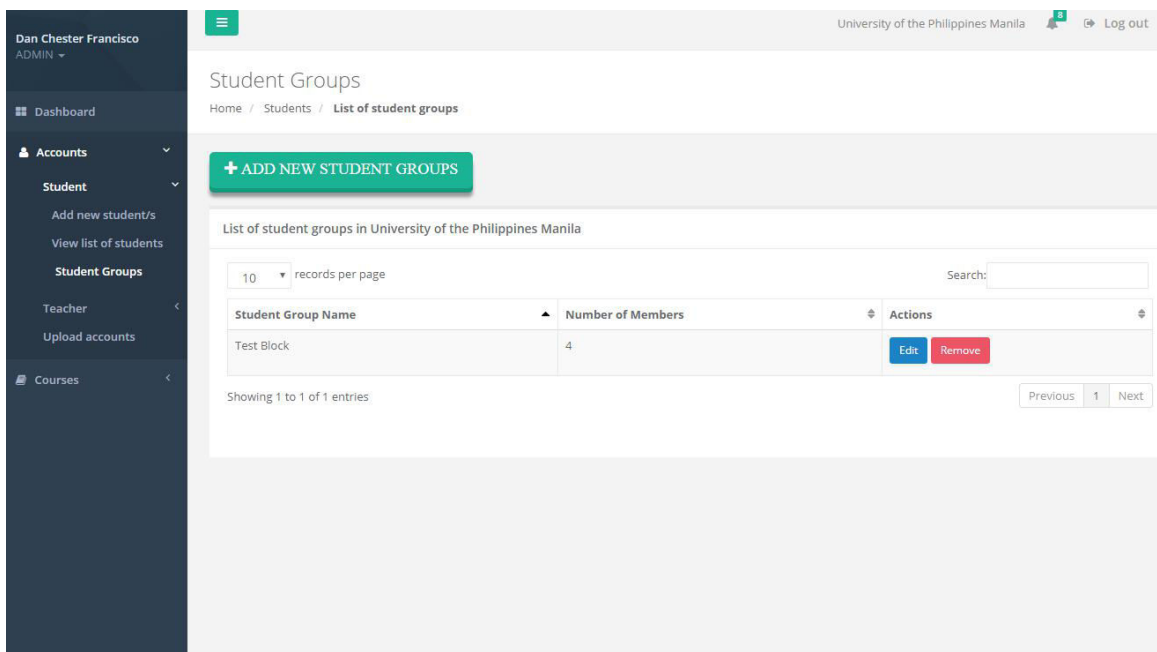


Figure 25: List of Student Groups - System Administrator

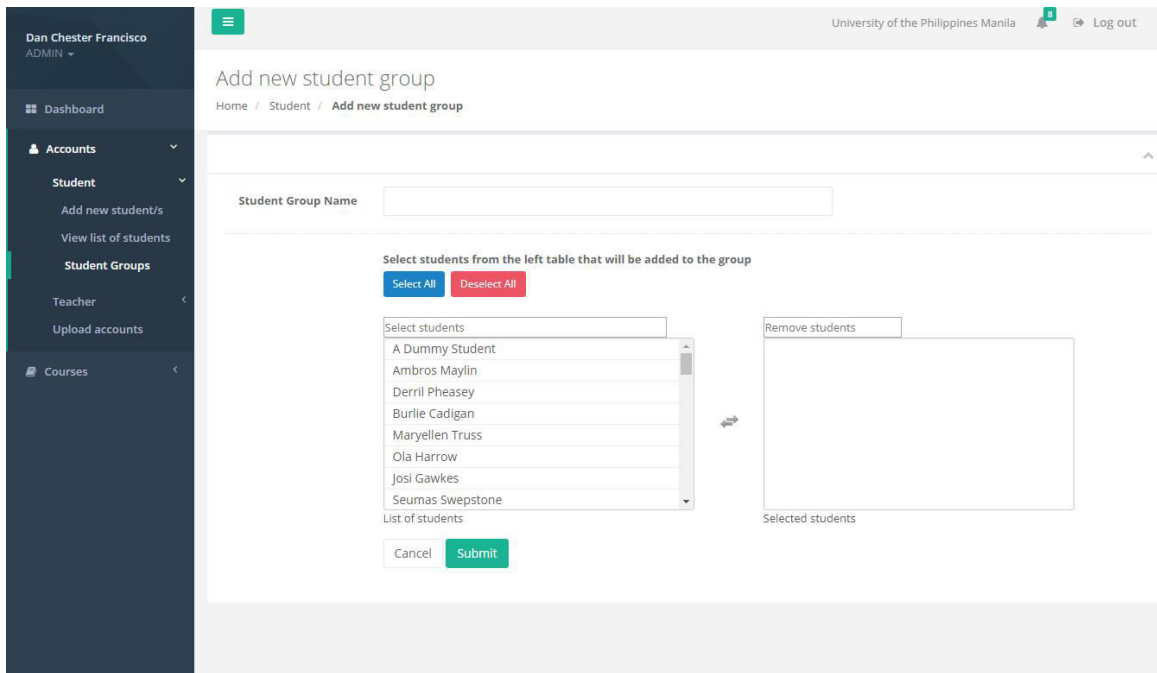


Figure 26: Create Student Groups - System Administrator

Registered students can also be grouped into student groups to hasten the assigning of students in courses. The system administrator is allowed to create student groups and assign students to it. By creating student groups, multiple students can be assigned to a course in an instance.

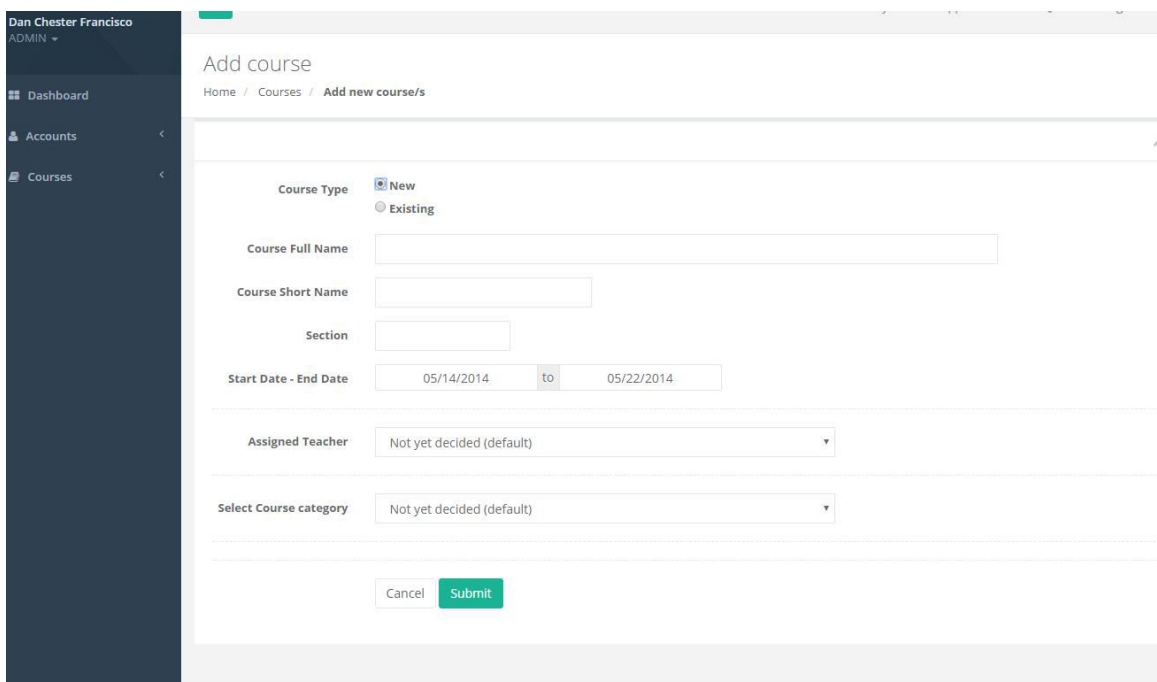


Figure 27: Create Course Form - System Administrator

Aside from user accounts management, the system administrator is capable of handling course information. The system allows him/her to create, modify, and delete courses from database. He/she is also responsible for assigning the department heads, teachers and students in courses.

In creating a new course, the system admin will choose whether the course is new or the course is existing already. A new course means that the course will be inserted without any existing section. The existing course means that it will just be another section of the same course. Inserting an existing course with an existing section name will be invalid.

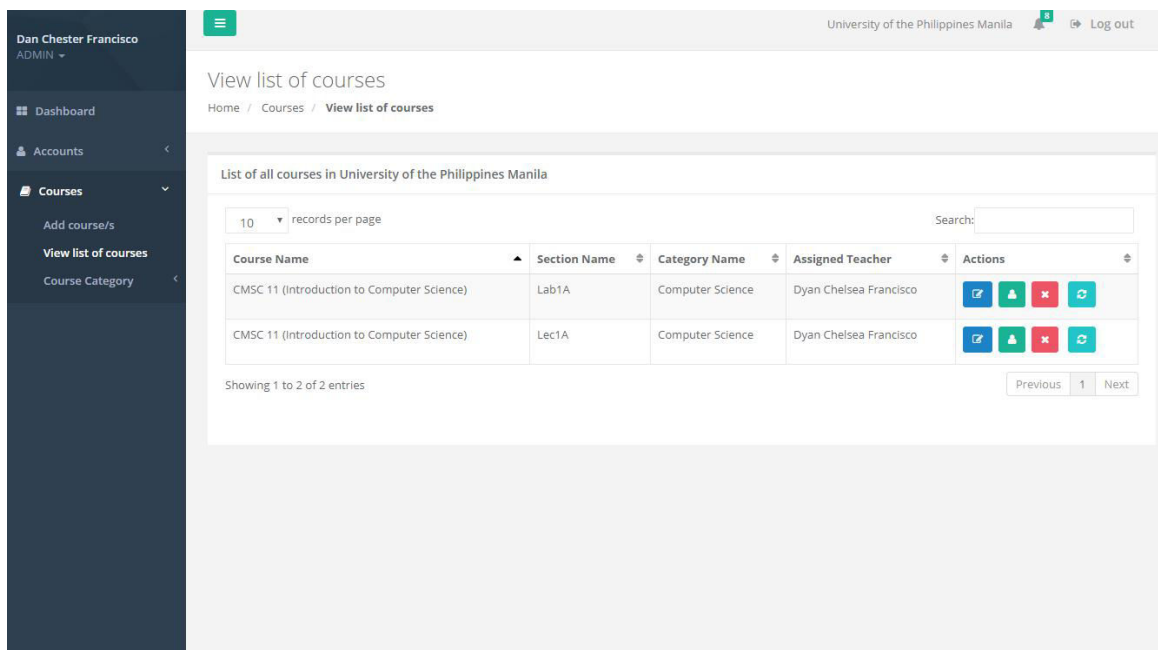


Figure 28: List of Courses - System Administrator

Upon creation of new course, the system administrator is also allowed to modify course information (clicking the blue edit button) and deletion of course (clicking the red x button). They are also allowed to clear the course record after the course duration (clicking the light blue cycle button).

By clicking the green “person” button as shown in figure 28, the system administrator can assign students to the courses. Student groups can also be assigned for bulk assignment of students in courses.

University of the Philippines Manila | Log out

Course students list - CMSC 11 Lec1A (Introduction to Computer Science)

Home / Course / **Course students list**

[+ ENROLL STUDENT](#) [+ ENROLL STUDENT GROUPS](#)

List of students - CMSC 11 Lec1A

10 records per page | Search:

Student Last Name	Student First Name	Email Address	Verified?	Actions
Cadigan	Burle	bcadigan2@slate.com	Not yet verified	Edit Remove
Maylin	Ambros	amaylin012@gmail.com	Verified	Edit Remove
Pheasey	Derril	dpheasey012@gmail.com	Not yet verified	Edit Remove
Student	A Dummy	dummystudent@gmail.com	Not yet verified	Edit Remove
Truss	Maryellen	mtruss3@nba.com	Not yet verified	Edit Remove

Showing 1 to 5 of 5 entries | [Previous](#) 1 [Next](#)

[<< Go back to list of courses](#)

Figure 29: List of assigned students in a course - System Administrator

University of the Philippines Manila | Log out

Enrol students - CMSC 11 (Introduction to Computer Science)

Home / Course / **Enrol students**

Select students from the left table that will be added to the group

[Select All](#) [Deselect All](#)

Select students

- Hynda Sarvar
- Tyrone Bitterton
- Gus Lafrentz
- Brendon Hugonneau
- Tadeo Amiranda
- Palm Joicey
- Gard Antonellini
- Nadia Cadman

List of students

[Go back to list](#) [Enrol](#)

Remove students

Selected students

Figure 30: Assigning students to course - System Administrator

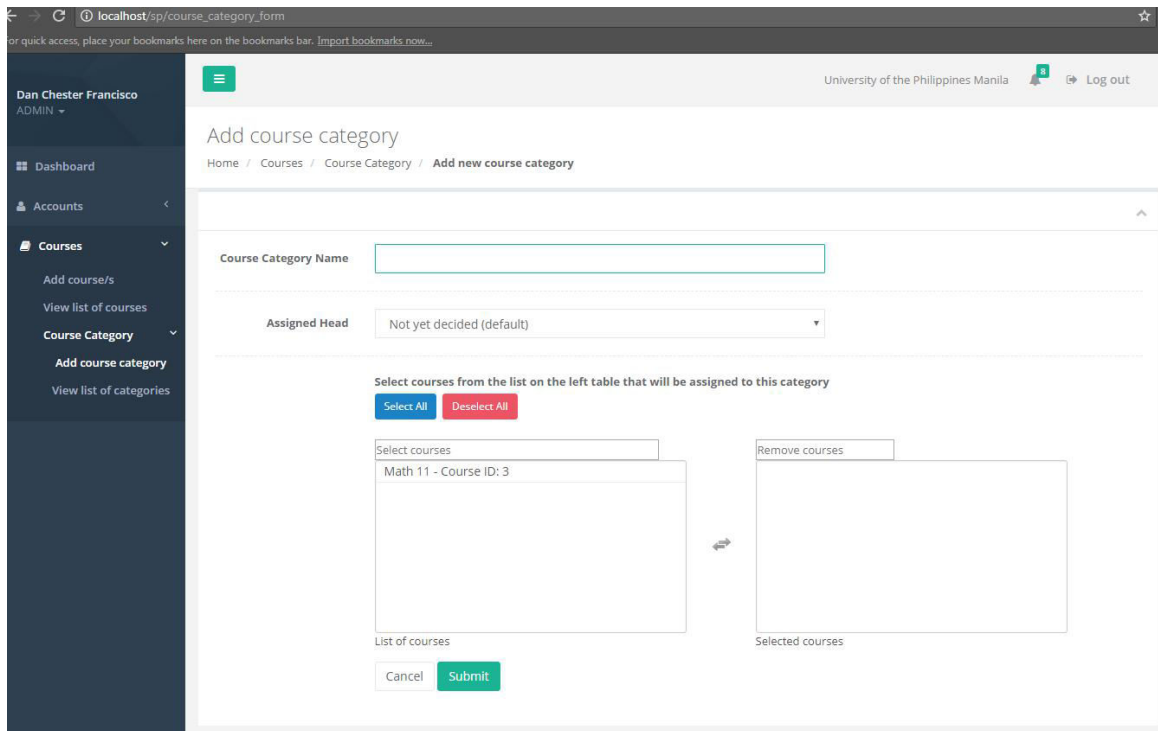


Figure 31: Creating Course Category - System Administrator

Similar to student groups, the system implements course category to apply changes to several courses in an instance. Every course category will be headed by a department/course category head. As shown in figure 31, the system administrator needs to insert a unique course category name and assign a category head.

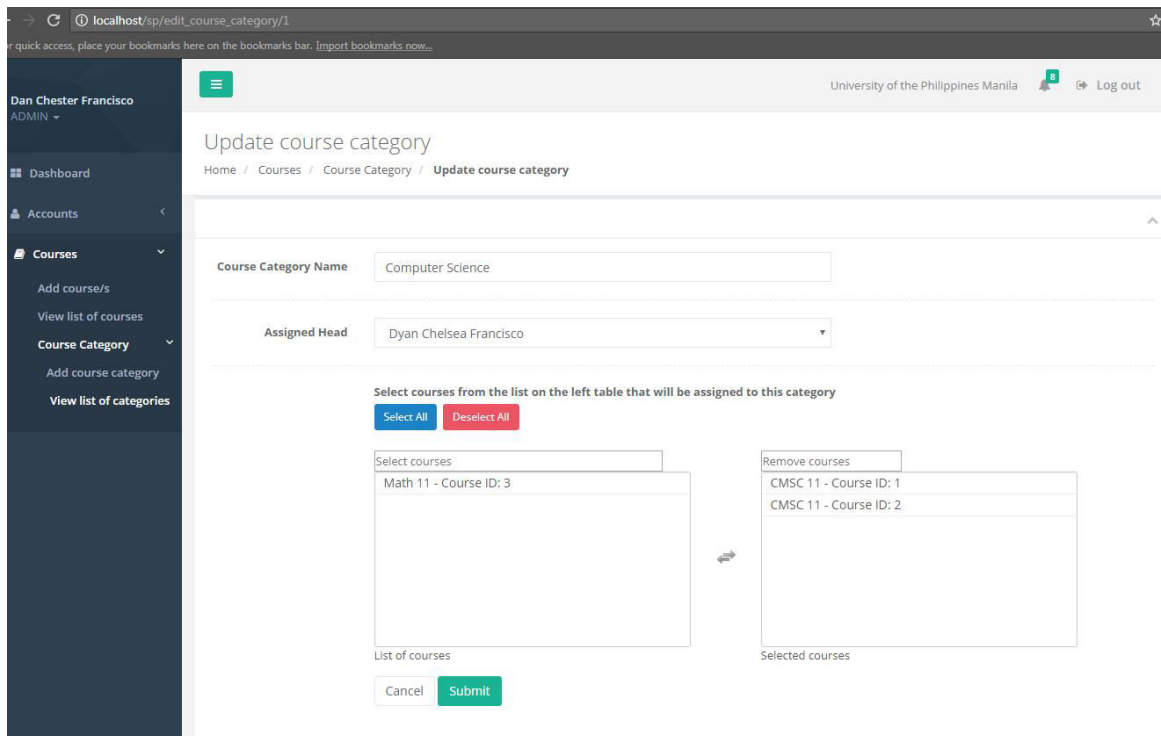


Figure 32: Updating Course Category - System Administrator

The system administrator can also modify the course category details by choosing a new category head, changing the course category name, and assigning/reassigning of courses (Figure 32).

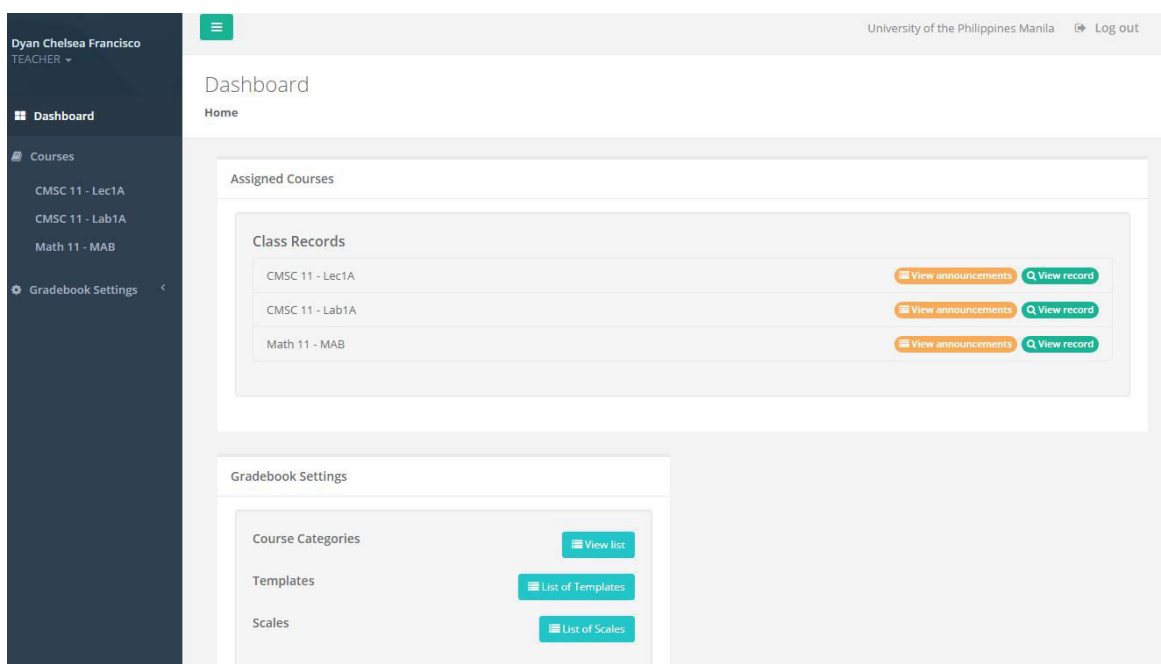


Figure 33: Dashboard - Department Head/ Teacher

The next type of users, the department head and the teacher, are similar to one another. The only difference between them is that the department head is capable of modifying the course categories. Inserting gradebook categories, items and sub-items into courses are some of the functionalities of the department head. The department head can also change the grading scale and allow other teachers to modify the gradebook setup of the course. (Figure 33)

Record - CMSC 11 Lec1A

		Quiz 1	Quiz 2	Total (33.33333%)	Activity 1	Activity 2	Activity 3	Total (33.33333%)	Machine Problem	Total (33.33333%)	Overall Total (100%)	Final Grade
		35	45		25	30	45		100			
5	Cadigan, Burlie	35	40	31.48	25	45	33	35.93	89	29.67	97.08	A
3	Maylin, Ambros	33	42	31.27	20	45	31	33.21	86	28.67	93.15	A
4	Pheasey, Derril	12	40	20.53	20	45	44	36.42	60	20	76.95	C
1	Student, A Dummy	10	30	15.87	12	30	45	27.56	70	23.33	66.76	D
6	Truss, Maryellen	25	25	21.16	22	30	37	30.02	45	15	66.18	D

Figure 34: Update Class Record - Department Head/ Teacher

One of the most important features of the system. The teachers can modify the student grades and store it into the database. (Figure 34)

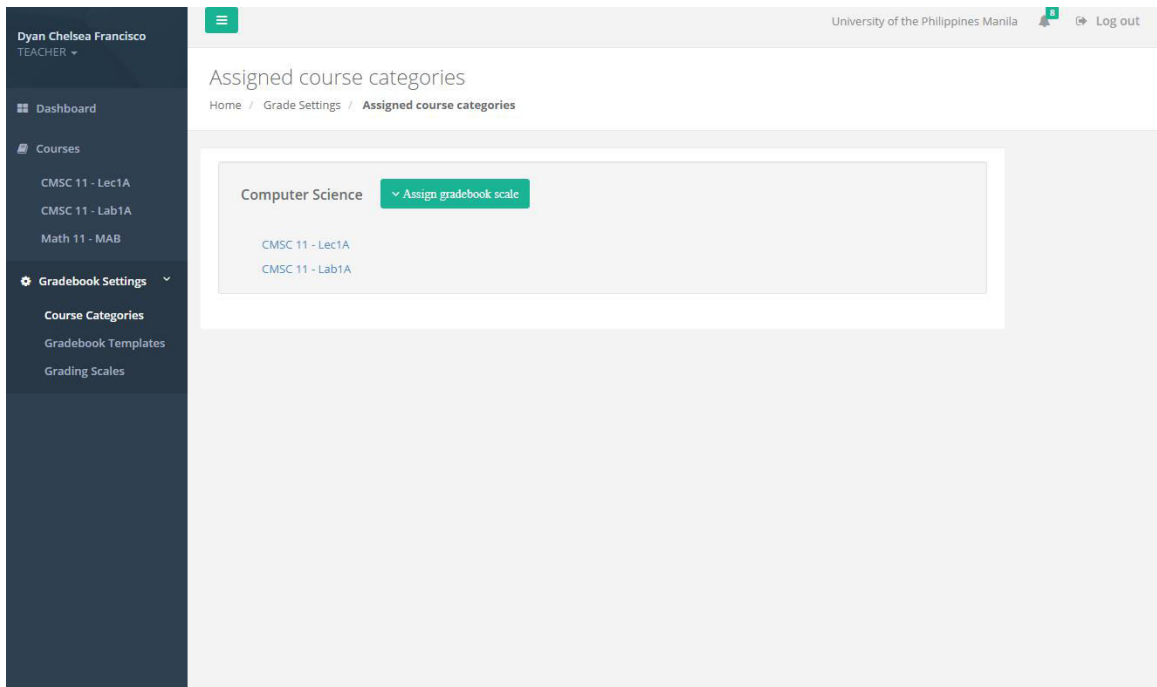


Figure 35: Modifying Course Categories - Department Head

Figure 35 shows that department heads can modify the courses by applying specific gradebook scale through the course category.

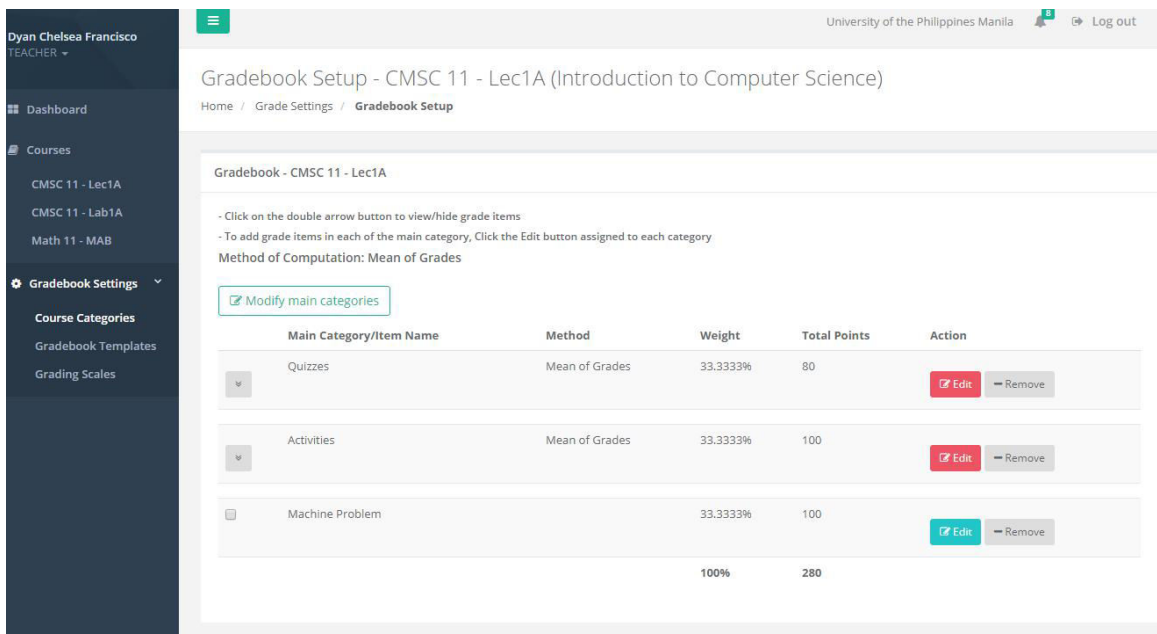


Figure 36: Modifying Gradebook Setup - Department Head/Teacher

The department head can edit the gradebook setup of the course. The teacher can also edit the gradebook setup given that the department head

allowed his/her to modify the gradebook. (Figure 36)

Home / Grade Settings / Add/modify gradebook categories

Method of Computation
Mean of Grades

How to update main categories/items

- No empty/blank input fields
- No duplicate main category/item names
- For weighted mean of grades, the sum of the weights must be less than or equal to 100

+ Add new category + Add new item

Category/Item Name	Weight Value (for weighted mean of grades)	Total Score (for grade items)	
Quizzes	33.3333 %	80	- Delete Category
Activities	33.3333 %	100	- Delete Category
Machine Problem	33.3333 %	100	- Delete Grade Item

← Go back to list Update

Figure 37: Modifying Gradebook Categories/Items - Department Head/Teacher

Figure 37 shows how to update gradebook categories. By selecting the method of computation and inserting/updating the gradebook category names, weights, and points.

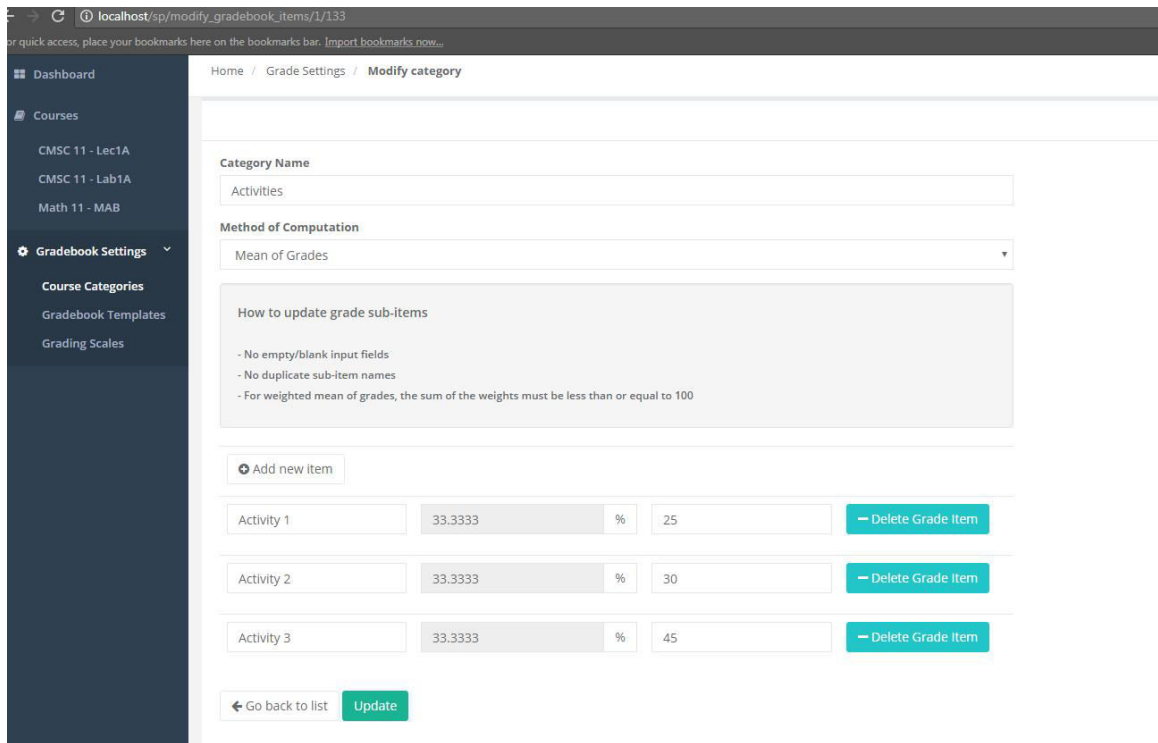


Figure 38: Modifying Gradebook Subitems - Department Head/Teacher

Similar to Figure 37, it shows how to update gradebook subitems. By selecting the method of computation and inserting/updating the gradebook subitem names, weights, and points.

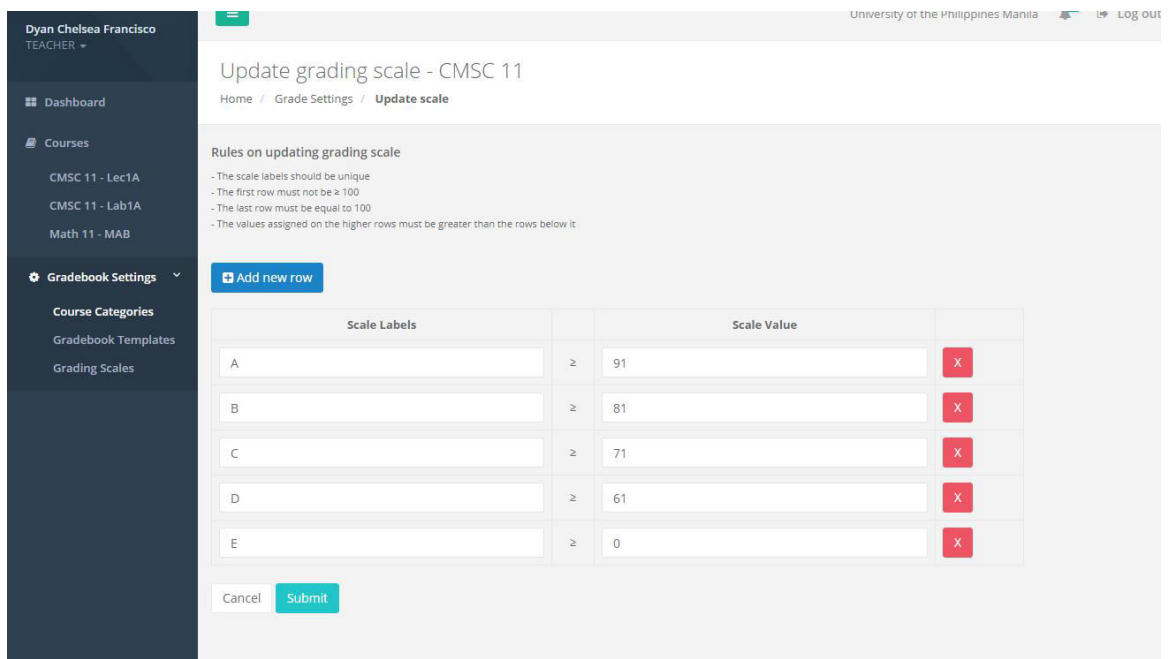


Figure 39: Update Grading Scale - Department Head/Teacher

The system also allows to represent final grades into scales or letters for better representation of the data (Figure 39).

The screenshot shows a web interface titled "Grading Scale". At the top, there are two buttons: "Save current" (with a floppy disk icon) and "Use existing" (with a downward arrow icon). Below these is a table with two main columns: "Scale Labels" and "Scale Values". The "Scale Values" column is further divided into two sub-columns by a hyphen, representing the start and end percentages for each grade. The table lists grades A through E with their respective percentage ranges. At the bottom of the interface is a large green button labeled "Update scale" with a list icon.

Scale Labels	Scale Values	
A	91.00%	- 100.00%
B	81.00%	- 90.00%
C	71.00%	- 80.00%
D	61.00%	- 70.00%
E	0.00%	- 60.00%

Figure 40: Save/Use Gradebook Scale - Department Head/Teacher

The scales on the grading system can also be stored as a template. The scale template will easily update the gradebook scale without entering the same scale values (Figure 40).

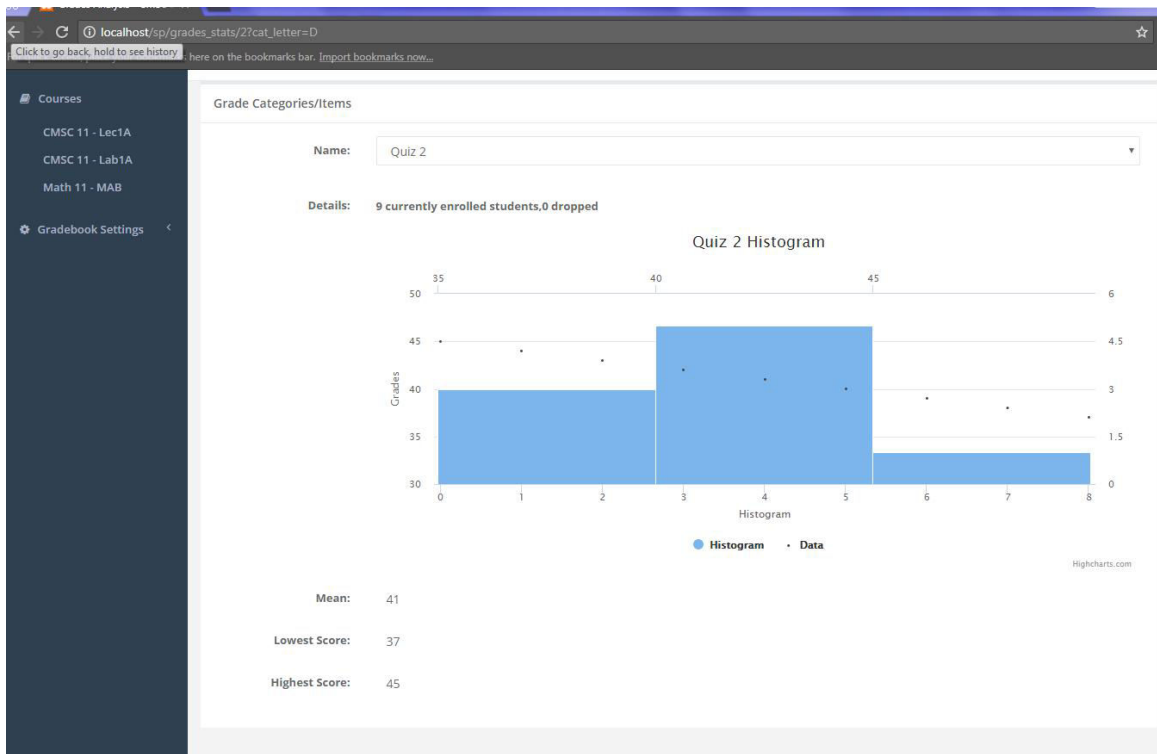


Figure 41: Student Grade Stats - Teacher

The system allows the teacher to view the statistical data of the students per gradebook subitems. This serves as an evidence for monitoring the student performance and activities. (Figure 41).

Dyan Chelsea Francisco
TEACHER

University of the Philippines Manila Log out

Course Announcements - CMSC 11 Lec1A
Home / CMSC 11

+ ADD ANNOUNCEMENT

Announcements

10 records per page Search:

Subject	Date/Time Modified	Action
Exam Next Friday	2018-05-23 05:43:12	VIEW DELETE

Showing 1 to 1 of 1 entries Previous 1 Next

Figure 42: Course Announcement - Teacher

Figure 42 shows that teachers are able to submit, update, and delete an announcement in their respective classes. e-mail will be sent to the members of the class regarding the announcement.

Grade Categories	Max Points	Score
Quiz 1	35	34
Quiz 2	45	44
Quiz 3	10	0
Quiz 4	10	0
Total (100pts)		= 48.73
Activity 1	25	25
Activity 2	30	28
Activity 3	45	44
Total (100pts)		= 97
Machine Problem	100	98
Total (100pts)		= 98
Final Grade		= 81.24
Scaled Grade		= B

Figure 43: Student Record - Student

Subject	Date/Time Modified	Action
Exam Next Friday	2018-05-23 05:43:12	VIEW

Figure 44: View Announcement - Student

The last user, students, are allowed to view their complete student record in the class where they belong (Figure 43). Figure 44 and 45 show that the students can view the announcements from their teachers.

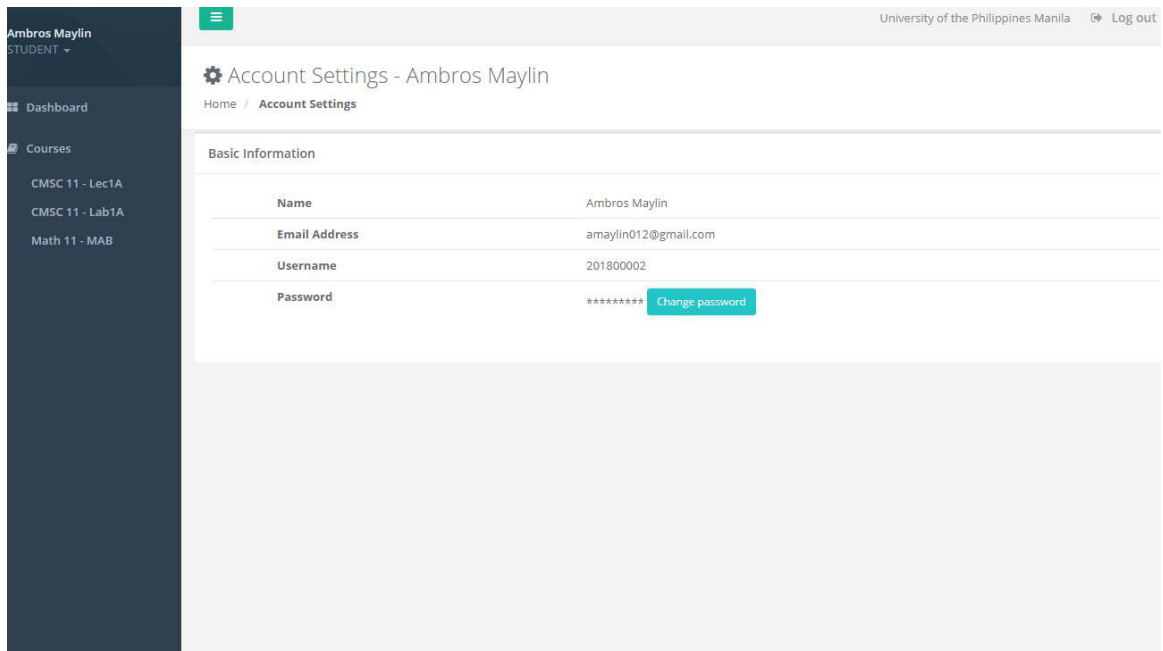


Figure 46: Account Settings - Users

All of the system users are allowed to change their password upon successfully logging in to the system. Changing other information such as e-mail address is the responsibility of the system administration (Figure 46).

VI. Discussions

The Electronic Class Record Inquiry System is a Web-based grades management system that allows the schools, especially the teachers, to manage the course records of the students. By implementing the spreadsheet library, the system is able to produce a tabular format grading tool, similar to the paper-based class records. In addition to that, the library allows the input of formulas in cells for efficient calculation of the grades.

Aside from the teacher and student users, the addition of the department head and the system administrator really helps on managing the student records and course information. The system administrator can easily register user accounts, create courses and assign people to different courses. The system also allows the administrator to upload multiple user accounts at once by uploading CSV file that contains the information of the users. For added security, the system also sends an e-mail of verification to the users for validation.

Most online gradebooks heavily rely on the teacher on managing information regarding the student and the courses. In the system, the role of the department head is to help the teachers in modifying the gradebook setup of the course. To ensure that the gradebook setup adheres to the standard set by the school, modification of gradebook setup, by default, can only be performed by the department head. He/she can allow the teachers to modify the gradebook setup of the course if needed.

Most schools also follows a standard grading scale to represent the grades of the students. The system allows the department head, and teachers if allowed by the department head, to modify the grading scale to use to represent the final grades of the student. The teachers can save a grading scale as a template into the database. In result, grading scale in other courses can be easily updated by using the stored template.

Another advantage of the system is monitoring the student's performance. From the course record, the teacher can observe the changes in the learning capability of the student. The system also can help in identifying difficult school activities from the easier ones.

However, there are also some disadvantages in the system. One of these is the possibility of fabricating the student grades affected by grading leniency. The system only accepts the numerical outcomes of the activities and represents the data in the spreadsheet. With the system having the capability of grades assessment like online submission of activities or online exams, the system can accurately record the student grades—improves the grading reliability.

Despite the disadvantages and minimal functionalities compared to other online gradebook, the system can still produce accurate results. Compared to other online gradebooks, the department head and the system administrator is an aid on managing school records.

VII. Conclusions

The Electronic Class Record Inquiry System has been designed and developed to help the schools in providing and managing student grade records. With the implementation of spreadsheet, the system is able to produce a tabular format grading tool similar to traditional class records.

Upon registration of the system administrator, he/she can register user accounts and course. The system administrator can also modify the user details and course details, and assign teachers/students/department heads to respective classes. In order to hasten assignment of students and course, he/she can create student groups and course categories to apply changes into several entities at once.

User accounts created by the system administrator will receive an e-mail for account validation. Students and teachers require to verify their accounts before they can log in to the system.

In the system, the teacher plays a vital role in which he/she can update the class record of the students. With the approval of the department head, the teacher are allowed to modify the gradebook setup of the course. In addition, course announcements can be created by the teachers with e-mails being sent to the e-mail addresses of the students.

The assigned department head, on the other hand, can modify the gradebook setup and gradebook scale of the course. He/she can also save gradebook scale for future uses. One of the most important roles of the department head is to allow the teachers to modify gradebook setup if needed.

To conclude, the Electronic Class Record Inquiry System has improved the management of student grades by spreadsheet implementation. From the grades statistics, the teacher can identify which items should be focused on— affecting the teaching strategies of the teacher. Aside from the teachers, the students can also monitor his/her performance by viewing the course record.

This can also change the study habits and behavior of the students. The system has also succeeded in disclosing student record without compromising student information privacy.

VIII. Recommendations

The inclusion of grading assessments in the system, such as assignment submission and online examinations, are recommended to produce accurate student records. Without grading assessments, errors in inserting student grades may happen and affect the grading reliability.

Aside from spreadsheet grade calculations, it is also recommended to include other functions that can be implemented in a gradebook such as attendance tracking, lesson plan management, individualized grading assessments, and collaboration with other teachers to improve the performance monitoring on the students.

IX. Bibliography

- [1] J. Bingcang, “Ict competency of teachers in science education in the philippines,” n.d.
- [2] J. Cunha, J. P. Fernandes, J. Mendes, and J. Saraiva, “Extension and implementation of ClassSheet models,” in *2012 IEEE Symposium on Visual Languages and Human-Centric Computing*, (Innsbruck, Austria), pp. 19–22, IEEE, 2012.
- [3] P. Sestoft, *Spreadsheet technology*. IT University of Copenhagen, 0.12 ed., December 2011.
- [4] P. Mireault, “Structured spreadsheet modeling and implementation,” *arXiv preprint arXiv:1503.03122*, 2015.
- [5] P. Maia, J. Cunha, J. Mendes, J. Saraiva, and H. Rebelo, “Extension and implementation of ClassSheet models,” in *In Proceedings of the 2nd Workshop on Software Engineering Methods in Spreadsheets*, 2015.
- [6] R. Abraham, M. Burnett, and M. Erwig, *Spreadsheet Programming*, pp. 2804–2810. American Cancer Society, 2009.
- [7] P. Sestoft, J. D. Rask, and S. Eikeland, “End-user development via sheet-defined functions,” *Software Engineering Methods in Spreadsheets*, p. 8, 2014.
- [8] F. Hermans and T. Van Der Storm, “Copy-paste tracking: Fixing spreadsheets without breaking them,” in *Proceedings of the 1st International Conference on Live Coding (ICLC)*, Citeseer, 2015.
- [9] Z. Chen, M. Cafarella, J. Chen, D. Prevo, and J. Zhuang, “Senbazuru: a prototype spreadsheet database management system,” *Proceedings of the VLDB Endowment*, vol. 6, no. 12, pp. 1202–1205, 2013.
- [10] P. M. Fernandes, C. Loureiro, and H. Botelho, “Piripinus: a spreadsheet application to guide prescribed burning operations in maritime

- pine forest,” *Computers and Electronics in Agriculture*, vol. 81, pp. 58–61, 2012.
- [11] U. of Malta, “Student records reporting facility.” <https://www.genome.gov/11511417>, 2015.
- [12] S. S. Hunnicutt, “An automated Grading Spreadsheet for reports or presentations,” *Journal of Chemical Education*, vol. 93, no. 1, pp. 210–212, 2016.
- [13] A. Z. Mansor, “Managing Student’s Grades and attendance Records using google forms and google Spreadsheets,” *Procedia - Social and Behavioral Sciences*, vol. 59, p. 420428, 10 2012.
- [14] S. Springer, B. Alexander, and K. Becker-Persiani, “Teacher Record keeping,” October 2011.
- [15] U. of the Philippines, “University of the philippines system code,” 2006.
- [16] P. U. of the Philippines, “The pup student handbook,” 2014.
- [17] A. de Manila University, “Undergraduate student handbook,” 2012.
- [18] D. la Salle University, “De la salle university: Office of the university registrar : Grades.”
- [19] “Grade aggregation - information technology.” Retrieved from St.Olaf College website.
- [20] V. Zwass, “Information system.” Retrived from Encyclopedia Britannica.
- [21] “Dbms tutorial.” Retrieved from Tutorials Point Web site.
- [22] “What is database management system?.” Retrieved from SearchSQL Server - Tech Target.
- [23] “Codeigniter.” Retrieved from Ellis Lab Website.
- [24] “Bootstrap - the world’s most popular mobile-first and responsive front-end framework.” Retrieved from Bootstrap website.

X. Appendix

A. Source Codes

Admin Controller

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Admin extends CI_Controller {

public function register_account(){
if(!$this->session->userdata('logged_in')){
$this->form_validation->set_rules('school_name','SchoolName', 'required|
trim|min_length[2]|max_length[128]|callback_schoolname_check');
$this->form_validation->set_rules('user_first_name','UserFirstName', '
required|trim|min_length[2]|max_length[64]');
$this->form_validation->set_rules('user_last_name','UserLastName', '
required|trim|min_length[2]|max_length[64]');
$this->form_validation->set_rules('user_email_address','UserEmailAddress
', 'required|trim|valid_email|callback_emailaddress_check');
$this->form_validation->set_rules('user_name','UserName', 'required|trim|
callback_username_check|min_length[6]|max_length[20]', array('
min_length' => '{field}_must_have_{param}-20_characters.', '
max_length' => '{field}_must_have_{param}_characters.'));
$this->form_validation->set_rules('user_password','UserPassword', '
required|trim|alpha_numeric|min_length[6]|max_length[20]');
if ($this->form_validation->run() === FALSE){ // inputs didn't pass
validation

$this->session->set_flashdata('errors', validation_errors());
redirect('register','refresh');
}else{
$this->user_model->register_admin();
$this->user_model->send_verify_mail(array($this->input->post('
user_email_address')));
$this->session->set_flashdata('succeed', 'Before you are allowed to log-in
, you need to verify your email!');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function verify_email(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "NO"){
$data['title'] = 'Email Verification';
$data['user_email_address'] = $this->user_model->get_email_address(
$_SESSION['logged_in']['user_id']);
$this->load->view('templates/header', $data);
$this->load->view('verify_email', $data);
$this->load->view('templates/footer', $data);
} else {
redirect('home', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

}

}

public function student_list(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'View list of students';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['students'] = $this->student_model->get_list_of_students();
$this->load->view('templates/admin_header', $data);
$this->load->view('templates/admin_sidebar', $data);
$this->load->view('student_list', $data);
$this->load->view('templates/JSscripts', $data);
$this->load->view('templates/admin_footer', $data);
} else {
$this->session->set_flashdata('error_404', 'True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function student_form(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'Add student/s';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['courses'] = $this->course_model->get_list_of_courses();
$this->load->view('templates/admin_header', $data);
$this->load->view('templates/admin_sidebar', $data);
$this->load->view('student_form', $data);
$this->load->view('templates/JSscripts', $data);
$this->load->view('templates/admin_footer', $data);
} else {
$this->session->set_flashdata('error_404', 'True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function add_student(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
email_verified'] === "YES"){
```

```

$this->form_validation->set_rules('student_first_name','User_FirstName',
    'required|trim|min_length[2]|max_length[64]');
$this->form_validation->set_rules('student_last_name','User_LastName',
    'required|trim|min_length[2]|max_length[64]');
$this->form_validation->set_rules('student_email_address','User_Email_
    Address', 'required|trim|valid_email|callback_emailaddress_check');
if ($this->form_validation->run() === FALSE){ // inputs didn't pass
    validation
    $this->session->set_flashdata('student_registry_error', validation_errors
        ());
    redirect('student_form','refresh');
}else{
    $this->student_model->add_student();
    $this->user_model->send_verify_mail(array($this->input->post('
        student_email_address')));
    $this->session->set_flashdata('student_registry_succeed', 'Success!');
    redirect('student_list','refresh');
}
} else {
    redirect('', 'refresh');
}
}

public function add_multiple_accounts(){
    if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
        email_verified'] === "YES"){
        if ($this->input->post('upload_user_role') === "Student"){ // inputs didn'
            t pass validation
            $emails = $this->student_model->add_multiple_student();
            $this->user_model->send_verify_mail($emails[0]);
            $this->session->set_flashdata('mult_student_registry_succeed', 'Success!')
                ;
            $this->session->set_flashdata('mult_student_registry_succeed1', $emails
                [1]);
            $this->session->set_flashdata('mult_student_registry_succeed2', $emails
                [2]);
            $this->session->set_flashdata('mult_student_registry_succeed3', $emails
                [3]);
            redirect('student_list','refresh');
        }else{
            $emails = $this->teacher_model->add_multiple_teacher();
            $this->user_model->send_verify_mail($emails[0]);
            $this->session->set_flashdata('mult_teacher_registry_succeed', 'Success!')
                ;
            $this->session->set_flashdata('mult_teacher_registry_succeed1', $emails
                [1]);
            $this->session->set_flashdata('mult_teacher_registry_succeed2', $emails
                [2]);
            $this->session->set_flashdata('mult_teacher_registry_succeed3', $emails
                [3]);
            redirect('teacher_list','refresh');
        }
    } else {
        redirect('', 'refresh');
    }
}

public function add_multiple_courses(){
    if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
        email_verified'] === "YES"){
        $this->course_model->add_multiple_courses();
        $this->session->set_flashdata('mult_course_registry_succeed', 'Success!');
        redirect('course_list','refresh');
    } else {
        redirect('', 'refresh');
    }
}

public function teacher_list(){
    if($this->session->userdata('logged_in')){
        if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
            logged_in']['user_role'] === "ADMIN"){
            $data['title'] = 'View_list_of_teachers';
            $data['user_details'] = $this->user_model->get_user_details_by_id(
                $_SESSION['logged_in']['user_id']);
            $data['school_name'] = $this->user_model->get_school_name($data['
                user_details']['school_id']);
            $data['teachers'] = $this->teacher_model->get_list_of_teachers();
            $this->load->view('templates/admin_header',$data);
            $this->load->view('templates/admin_sidebar',$data);
            $this->load->view('teacher_list',$data);
            $this->load->view('templates/JSscripts',$data);
            $this->load->view('templates/admin_footer',$data);
        } else {
            $this->session->set_flashdata('error_404','True');
            redirect('', 'refresh');
        }
    } else {
        redirect('', 'refresh');
    }
}

public function teacher_form(){
    if($this->session->userdata('logged_in')){
        if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
            logged_in']['user_role'] === "ADMIN"){
            $data['title'] = 'Add_teacher/s';
            $data['user_details'] = $this->user_model->get_user_details_by_id(
                $_SESSION['logged_in']['user_id']);
            $data['school_name'] = $this->user_model->get_school_name($data['
                user_details']['school_id']);
            $data['courses'] = $this->course_model->get_list_of_unassigned_courses();
            $this->load->view('templates/admin_header',$data);
            $this->load->view('templates/admin_sidebar',$data);
            $this->load->view('teacher_form',$data);
            $this->load->view('templates/JSscripts',$data);
            $this->load->view('templates/admin_footer',$data);
        } else {
            $this->session->set_flashdata('error_404','True');
            redirect('', 'refresh');
        }
    } else {
        redirect('', 'refresh');
    }
}

public function add_teacher(){
    if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
        email_verified'] === "YES"){
        $this->form_validation->set_rules('teacher_first_name','User_FirstName',
            'required|trim|min_length[2]|max_length[64]');
        $this->form_validation->set_rules('teacher_last_name','User_LastName',
            'required|trim|min_length[2]|max_length[64]');
        $this->form_validation->set_rules('teacher_email_address','User_Email_
            Address', 'required|trim|valid_email');
    }
}

```



```

if ($this->form_validation->run() === FALSE){ // inputs didn't pass
    validation
$this->session->set_flashdata('teacher_registry_error', validation_errors
());
redirect('teacher_form','refresh');
}else{
$this->teacher_model->add_teacher();
$this->user_model->send_verify_mail(array($this->input->post('
teacher_email_address')));
$this->session->set_flashdata('teacher_registry_succeed', 'Success!');
redirect('teacher_list','refresh');
}
} else {
redirect('', 'refresh');
}
}

public function student_groups_list(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'Student_Groups';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['student_groups'] = $this->student_model->get_student_groups();
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/admin_sidebar',$data);
$this->load->view('student_groups_list',$data);
$this->load->view('templates/JSscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_flashdata('error_404','True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function student_groups_form(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'Add_new_student_group';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['students'] = $this->student_model->get_list_of_students();
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/admin_sidebar',$data);
$this->load->view('student_groups_form',$data);
$this->load->view('templates/JSscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_flashdata('error_404','True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

```

```

}
}

public function add_student_group(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
email_verified'] === "YES"){
$this->form_validation->set_rules('student_group_name','Student_Group_Name
', 'required|trim|min_length[3]|max_length[64]|
callback_groupname_check',array('min_length' => '{field}_must_have_
at_least_{param}_characters.','max_length' => '{field}_must_not_
exceed_{param}_characters.));
if ($this->form_validation->run() === FALSE){ // inputs didn't pass
    validation
$this->session->set_flashdata('student_group_error', validation_errors());
redirect('student_groups_form','refresh');
}else{
$this->student_model->add_student_group($this->input->post('
student_group_name'));
$this->session->set_flashdata('student_group_succeed', 'Success!');
redirect('student_groups_list','refresh');
}
} else {
redirect('', 'refresh');
}
}

public function upload_courses(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'Upload_courses';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/admin_sidebar',$data);
$this->load->view('upload_courses',$data);
$this->load->view('templates/JSscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_flashdata('error_404','True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function upload_accounts(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'Upload_accounts';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/admin_sidebar',$data);
$this->load->view('upload_accounts',$data);
$this->load->view('templates/JSscripts',$data);
$this->load->view('templates/admin_footer',$data);
}
}
}

```

```

} else {
$this->session->set_tempdata('error_404','True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}
}

public function course_file_contents(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] == "YES"){
$config['upload_path'] = './uploads/';
$config['max_size'] = 64000;
$config['file_ext_tolower'] = true;
if($this->input->post('upload_file_type') == "CSV"){
$config['allowed_types'] = 'csv';
} else {
$config['allowed_types'] = 'xls|xlsx';
}
$this->load->library('upload');
$this->upload->initialize($config);
if (!$this->upload->do_upload('userfile'))
{
$error = array('error' => $this->upload->display_errors());
$this->session->set_flashdata('upload_account_error',$error['error']);
redirect('upload_courses','refresh');
} else {
$upload_data = $this->upload->data();
if($this->input->post('upload_file_type') == "CSV"){
$headers = array();
$values = array();
$row = 0;
if (($handle = fopen(base_url('uploads/'.$upload_data['file_name']), "r"))
    !== FALSE) {
while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
$num = count($data);
if($num == 2){ // limiting the fields to only 3 values
for ($c=0; $c < $num; $c++){
if($row == 0){
if($data[$c] == "Course_Full_Name" || $data[$c] == "Course_Short_Name"){
$headers[$c] = $data[$c];
} else {
$this->session->set_flashdata('upload_header_error','Invalid
header values!');
redirect('upload_courses','refresh');
}
} else {
$values[$row-1][$c] = $data[$c];
}
}
$row++;
} else {
$this->session->set_flashdata('upload_count_error','The
headers/values in
the file contains blank/excess values!');
redirect('upload_courses','refresh');
}
}
fclose($handle);
}
}
}
}
}

$clean_array = array_unique($values, SORT_REGULAR);

```

```

$data['headers2'] = $headers;
$data['values'] = $clean_array;
} else {
$array = $this->display_excel_course($upload_data['file_name'],$this->
input->post('upload_user_role'));
$data['headers2'] = $array['headers'][0];
$data['values'] = $array['values'];
}
}
}

$data['title'] = 'Preview_file_contents_Courses';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['students'] = $this->student_model->get_list_of_students();
$data['course_categories'] = $this->course_model->get_list_of_categories()
;
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/admin_sidebar',$data);
$this->load->view('course_file_contents',$data);
$this->load->view('templates/JSscripts',$data);
$this->load->view('templates/admin_footer',$data);
}
} else {
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}
}

public function account_file_contents(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] == "YES"){
$config['upload_path'] = './uploads/';
$config['max_size'] = 64000;
$config['file_ext_tolower'] = true;
if($this->input->post('upload_file_type') == "CSV"){
$config['allowed_types'] = 'csv';
} else {
$config['allowed_types'] = 'xls|xlsx';
}
$this->load->library('upload');
$this->upload->initialize($config);
if (!$this->upload->do_upload('userfile'))
{
$error = array('error' => $this->upload->display_errors());
$this->session->set_flashdata('upload_account_error',$error['error']);
redirect('upload_accounts','refresh');
} else {
$upload_data = $this->upload->data();
if($this->input->post('upload_file_type') == "CSV"){
$headers = array();
$values = array();
$row = 0;
if (($handle = fopen(base_url('uploads/'.$upload_data['file_name']), "r"))
    !== FALSE) {
while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
$num = count($data);
if($num == 3){ // limiting the fields to only 3 values
for ($c=0; $c < $num; $c++){
if($row == 0){

```

```

if($data[$c] === "First_Name" || $data[$c] === "Last_Name" || $data[$c]
    === "Email_Address"){
$headers[$c] = $data[$c];
} else {
    $this->session->set_flashdata('upload_header_error','Invalid_
        header_values!');
    redirect('upload_accounts','refresh');
}
} else {
$values[$row-1][$c] = $data[$c];
}
}
$row++;
} else {
$this->session->set_flashdata('upload_count_error','The_headers/values_in_
    the_file_contains_blank/excess_values!');
redirect('upload_accounts','refresh');
}
}
fclose($handle);
}
$clean_array = array_unique($values, SORT_REGULAR);
$data['headers'] = $headers;
$data['values'] = $clean_array;
} else {
$array = $this->display_excel_account($upload_data['file_name'],$this->
    input->post('upload_user_role'));
$data['headers'] = $array['headers'][0];
$data['values'] = $array['values'];
}
$data['title'] = 'Preview_file_contents_'.$this->input->post('
    upload_user_role');
$data['upload_user_role'] = $this->input->post('upload_user_role');
$data['user_details'] = $this->user_model->get_user_details_by_id(
    $_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
    user_details']['school_id']);
$data['students'] = $this->student_model->get_list_of_students();
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/admin_sidebar',$data);
$this->load->view('account_file_contents',$data);
$this->load->view('templates/JSscripts',$data);
$this->load->view('templates/admin_footer',$data);
}
} else {
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}
public function display_excel_account($filename,$user_role){
$file = 'uploads/'.$filename;
//load the excel library
$this->load->library('Excel');
//read file from path
$objPHPExcel = PHPEXCEL_IOFactory::load($file);
//get only the Cell Collection
$cell_collection = $objPHPExcel->getActiveSheet()->getCellCollection();
$count = 0;
$max_row = 0;
$count_blanks = 0;
$prev_row = 0;
foreach ($cell_collection as $cell) {
$column = $objPHPExcel->getActiveSheet()->getCell($cell)->getColumn();
$row = $objPHPExcel->getActiveSheet()->getCell($cell)->getRow();
$data_value = $objPHPExcel->getActiveSheet()->getCell($cell)->getValue();
//header will/should be in row 1 only. of course this can be modified to
    suit your need.
$colNumber = PHPEXCEL_Cell::columnIndexFromString($column);
if($colNumber <= 3){
if($row == 1){
if($data_value === "Last_Name" || $data_value === "First_Name" ||
    $data_value === "Email_Address"){
$header[$row-1][$colNumber-1] = $data_value;
} else {
$this->session->set_flashdata('upload_excel_error','The_headers/values_in_
        the_file_contains_blank/excess_values!');
redirect('upload_accounts','refresh');
}
} else {
$arr_data[$row-2][$colNumber-1] = $data_value;
}
}
}
$clean_array = $this->remove_blank_arrays($arr_data);
$clean_array = array_unique($clean_array, SORT_REGULAR);
return array('headers' => $header,'values' => $clean_array);
}
public function display_excel_course($filename,$user_role){
$file = 'uploads/'.$filename;
//load the excel library
$this->load->library('Excel');
//read file from path
$objPHPExcel = PHPEXCEL_IOFactory::load($file);
//get only the Cell Collection
$cell_collection = $objPHPExcel->getActiveSheet()->getCellCollection();
$count = 0;
$max_row = 0;
$count_blanks = 0;
$prev_row = 0;
foreach ($cell_collection as $cell) {
$column = $objPHPExcel->getActiveSheet()->getCell($cell)->getColumn();
$row = $objPHPExcel->getActiveSheet()->getCell($cell)->getRow();
$data_value = $objPHPExcel->getActiveSheet()->getCell($cell)->getValue();
//header will/should be in row 1 only. of course this can be modified to
    suit your need.
$colNumber = PHPEXCEL_Cell::columnIndexFromString($column);
if($colNumber <= 2){
if($row == 1){
if($data_value === "Course_Full_Name" || $data_value === "Course_Short_
    Name"){
$header[$row-1][$colNumber-1] = $data_value;
} else {
$this->session->set_flashdata('upload_excel_error','The_headers/values_in_
        the_file_contains_blank/excess_values!');
redirect('upload_courses','refresh');
}
}
} else {
$arr_data[$row-2][$colNumber-1] = $data_value;
}
}
}
}

```

```

}
}
}
$clean_array = $this->remove_blank_arrays($arr_data);
$clean_array = array_unique($clean_array, SORT_REGULAR);
return array('headers' => $header, 'values' => $clean_array);
}

public function remove_blank_arrays($array){
return array_filter($array, function($v){return array_filter($v) != array
();});
}

public function course_list(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'View list of courses';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['courses'] = $this->course_model->get_list_of_courses();
$this->session->unset_userdata('go_to_category');
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/admin_sidebar',$data);
$this->load->view('course_list',$data);
$this->load->view('templates/JSscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_userdata('error_404','True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function category_courses($course_category_id){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
logged_in']['user_role'] === "ADMIN"){
$data['category_details'] = $this->course_model->get_category_details(
$course_category_id);
$data['title'] = 'List of courses in Category ' . $data['category_details']['
course_category_name'];
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['courses'] = $this->course_model->get_list_of_courses_by_category(
$course_category_id);
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/admin_sidebar',$data);
$this->load->view('category_courses_list',$data);
$this->load->view('templates/JSscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_userdata('error_404','True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}
}

```

```

}
}
public function course_form(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'Add course';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['existing_course'] = $this->course_model->get_existing_courses($data
['user_details']['school_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['teacher'] = $this->teacher_model->get_list_of_teachers();
$data['course_category'] = $this->course_model->get_list_of_categories();
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/admin_sidebar',$data);
$this->load->view('course_form',$data);
$this->load->view('templates/JSscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_userdata('error_404','True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function add_course(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
email_verified'] === "YES"){
$type = $this->input->post('optionsRadios');
if($type === "new"){
$this->form_validation->set_rules('course_short_name','Course Short Name',
'required|callback_course_shortcode_check|trim|min_length[3]|
max_length[20]',array('min_length' => '{field} must have at least {
param} characters.', 'max_length' => '{field} must not exceed {param}
characters. ');
$this->form_validation->set_rules('course_full_name','Course Full Name',
'required|callback_course_fullname_check|trim|min_length[3]|
max_length[64]',array('min_length' => '{field} must have at least {
param} characters.', 'max_length' => '{field} must not exceed {param}
characters. ');
$this->form_validation->set_rules('section_name','Section Name', 'required
|trim|callback_section_check|min_length[3]|max_length[64]',array('
min_length' => '{field} must have at least {param} characters.',
'max_length' => '{field} must not exceed {param} characters. ');
} else {
$this->form_validation->set_rules('section_name','Section Name', 'required
|callback_section_check|trim|min_length[3]|max_length[64]',array('
min_length' => '{field} must have at least {param} characters.',
'max_length' => '{field} must not exceed {param} characters. ');
}
}
if ($this->form_validation->run() === FALSE) { // inputs didn't pass
validation
$this->session->set_flashdata('course_registry_error', validation_errors()
);
redirect('course_form','refresh');
} else {
if($type === "new"){

```

```

$this->course_model->add_course($type,$this->input->post('
    course_short_name'));
} else {
$this->course_model->add_course($type,$this->input->post('
    course_short_name'),$this->input->post('existing_course'));
}
$this->session->set_flashdata('course_registry_succeed', 'Success!');
redirect('course_list','refresh');
}
} else {
redirect('', 'refresh');
}
}

public function section_check($str)
{
if ($this->course_model->check_section_name($str,$this->input->post('
    optionsRadios'))
{
$this->form_validation->set_message('section_check', 'The section already
    exists. ');
return FALSE;
}
else
{
return TRUE;
}
}

public function course_fullname_check($str)
{
if ($this->course_model->check_course_fullname($str))
{
$this->form_validation->set_message('course_fullname_check', 'The {field}
    already exists. Try entering a new section by clicking the existing
    button and selecting your desired course. ');
return FALSE;
}
else
{
return TRUE;
}
}

public function course_shortcode_check($str)
{
if ($this->course_model->check_course_shortcode($str))
{
$this->form_validation->set_message('course_shortcode_check', 'The {field}
    already exists. Try entering a new section by clicking the existing
    button and selecting your desired course. ');
return FALSE;
}
else
{
return TRUE;
}
}

public function course_category_list(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
    logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'View course categories';
}

$data['user_details'] = $this->user_model->get_user_details_by_id(
    $_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
    user_details']['school_id']);
$data['categories'] = $this->course_model->get_list_of_categories();
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/admin_sidebar',$data);
$this->load->view('course_category_list',$data);
$this->load->view('templates/JScscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_flashdata('error_404','True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function course_category_form(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
    logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'Add course category';
$data['user_details'] = $this->user_model->get_user_details_by_id(
    $_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
    user_details']['school_id']);
$data['courses'] = $this->course_model->get_list_of_uncategorized_courses
    ();
$data['teacher'] = $this->teacher_model->get_list_of_teachers();
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/admin_sidebar',$data);
$this->load->view('course_category_form',$data);
$this->load->view('templates/JScscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_flashdata('error_404','True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function add_course_category(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
    email_verified'] === "YES"){
$this->form_validation->set_rules('course_category_name','Course
    Category
    Name', 'required|trim|callback_coursecat_check|min_length[3]
    max_length[64],array('min_length' => '{field}must_have_at_least{
    param}characters.','max_length' => '{field}must_not_exceed{param}
    characters. ');
if ($this->form_validation->run() === FALSE) { // inputs didn't pass
    validation
    $this->session->set_flashdata('course_cat_registry_error',
        validation_errors());
    redirect('course_category_form','refresh');
} else {
$this->course_model->add_course_category();
$this->session->set_flashdata('course_cat_registry_succeed', 'Course
    category added!');
}
}
}
}

```

```

redirect('course_category_list','refresh');
}
} else {
redirect('', 'refresh');
}
}

public function validate_email_check(){
if(!$this->session->userdata('logged_in')){
$this->form_validation->set_rules('email_address','Email_Address', '
    required|trim|min_length[1]|max_length[64]|
    callback_email_exists_first');
$user_email = $this->input->post('email_address');
if ($this->form_validation->run() === FALSE){ // inputs didn't pass
    validation

$this->session->set_flashdata('search_error1', validation_errors());
redirect('validate_email','refresh');
}else{
redirect('verify_success/'.md5($user_email), 'refresh');
//$this->session->set_flashdata('search_succeed', 'The reset password link
    has been sent to your email linked to the account. Check your email
    to reset your password.');
```

```

//redirect('forgot_pass','refresh');
}
} else {
redirect('', 'refresh');
}
}

public function verify_success($user_email_address){
$this->session->unset_userdata('logged_in');
session_destroy();
if(!$this->session->userdata('logged_in')){
$accounts = $this->user_model->get_users();
foreach($accounts as $user){
if(md5($user['user_email_address']) === $user_email_address){
$verified_email_address = $user['user_email_address'];
break;
}
}

$row = $this->user_model->check_email_verification($verified_email_address
    );
if($row['email_verified'] === "YES"){
$data['username'] = $row['user_name'];
$data['userrole'] = $row['user_role'];
$data['title'] = 'Verification_Success!';
$this->load->view('templates/header',$data);
$this->load->view('verify_success',$data);
$this->load->view('templates/footer',$data);
} else {
redirect('', 'refresh');
}
} else {
redirect('home','refresh');
}
}

public function reset_password($user_email_address){
if(!$this->session->userdata('logged_in')){
$accounts = $this->user_model->get_users();
foreach($accounts as $user){
if(md5($user['user_email_address']) === $user_email_address){
$verified_email_address = $user['user_email_address'];
break;
}
}

$row = $this->user_model->get_user_details_by_email(
    $verified_email_address);
if($row['email_verified'] === "YES"){
$data['details'] = $row;
$data['title'] = 'Reset_Password';
$this->load->view('templates/header',$data);
$this->load->view('reset_password',$data);
$this->load->view('templates/footer',$data);
} else {
$this->session->set_flashdata('email_not_verified','Account_should_be_
    verified_through_email_before_requesting_for_reset_password_mail.');
```

```

redirect('', 'refresh');
}
} else {
redirect('home','refresh');
}
}

public function reset_success(){
if(!$this->session->userdata('logged_in')){
$data['title'] = 'Reset_Password_Successful!';
$this->load->view('templates/header',$data);
$this->load->view('reset_success',$data);
$this->load->view('templates/footer',$data);
} else {
redirect('home','refresh');
}
}

public function update_password(){
if(!$this->session->userdata('logged_in')){
$this->form_validation->set_rules('user_password','User_Password', '
    required|trim|alpha_numeric|min_length[6]|max_length[20]', array('
    min_length' => '{field}_must_have_{param}-20_characters.', '
    max_length' => '{field}_must_have_{param}_characters.');
```

```

if ($this->form_validation->run() === FALSE){ // inputs didn't pass
    validation

$this->session->set_flashdata('reset_error', validation_errors());
redirect('reset_password/'.md5($this->input->post('user_email_address')), '
    refresh');
```

```

}else{
$this->user_model->update_new_password($this->input->post('user_name'),
    $this->input->post('user_password'));
$this->session->set_flashdata('reset_succeed', 'You_can_now_login_with_
    your_new_password!');
```

```

redirect('reset_success','refresh');
}
} else {
redirect('', 'refresh');
}
}

public function send_reset_mail(){
if(!$this->session->userdata('logged_in')){
$this->form_validation->set_rules('search_user_filter','Search_Filter', '
    required|trim');
```

```

$search_filter = $this->input->post('search_user_filter');
if($search_filter === "user_name"){
$this->form_validation->set_rules('details_value','Details_Value', '
    required|trim|min_length[1]|max_length[64]|callback_username_exists'
```

```

    );
    $user_id = $this->user_model->get_user_by_name($this->input->post('
        details_value'));
    $user_email = $this->user_model->get_email_address($user_id);
} else {
    $this->form_validation->set_rules('details_value', 'Details Value', '
        required|trim|min_length[1]|max_length[64]|callback_email_exists');
    $user_email = $this->input->post('details_value');
}
if ($this->form_validation->run() == FALSE){ // inputs didn't pass
    validation
    $this->session->set_flashdata('search_error', validation_errors());
    redirect('forgot_pass', 'refresh');
} else {
    $this->user_model->send_reset_email($user_email);
    // $this->session->set_flashdata('search_succeed', 'The reset password link
        has been sent to your email linked to the account. Check your email
        to reset your password. ');
    // redirect('forgot_pass', 'refresh');
}
} else {
    redirect('', 'refresh');
}
}
public function username_check($str)
{
    if ($this->user_model->check_user_name($str))
    {
        $this->form_validation->set_message('username_check', 'The {field} already
            exists. ');
        return FALSE;
    }
    else
    {
        return TRUE;
    }
}
public function schoolname_check($str)
{
    if ($this->user_model->check_school_name($str))
    {
        $this->form_validation->set_message('schoolname_check', 'The {field}
            already exists. ');
        return FALSE;
    }
    else
    {
        return TRUE;
    }
}
public function emailaddress_check($str)
{
    if ($this->user_model->check_email_address($str))
    {
        $this->form_validation->set_message('emailaddress_check', 'The {field}
            already exists. ');
        return FALSE;
    }
    else
    {

```

```

        return TRUE;
    }
}
public function coursecat_check($str)
{
    if ($this->course_model->check_course_category_name($str))
    {
        $this->form_validation->set_message('coursecat_check', 'The {field}
            already exists. ');
        return FALSE;
    }
    else
    {
        return TRUE;
    }
}
public function update_coursecat_check($str)
{
    if ($this->course_model->check_update_course_category_name($str, $this->
        input->post('course_category_id'))
    {
        $this->form_validation->set_message('update_coursecat_check', 'The {field}
            already exists. ');
        return FALSE;
    }
    else
    {
        return TRUE;
    }
}
public function groupname_check($str)
{
    if ($this->student_model->check_student_group_name($str))
    {
        $this->form_validation->set_message('groupname_check', 'The {field}
            already exists. ');
        return FALSE;
    }
    else
    {
        return TRUE;
    }
}
public function update_groupname_check($str)
{
    if ($this->student_model->check_update_student_group_name($str, $this->
        input->post('student_group_id'))
    {
        $this->form_validation->set_message('update_groupname_check', 'The {field}
            already exists. ');
        return FALSE;
    }
    else
    {
        return TRUE;
    }
}
public function username_exists($str)
{
    if ($this->user_model->check_user_name($str))

```

```

{
return TRUE;
}
else
{
$this->form_validation->set_message('username_exists', 'The username is
not stored in the database. ');
return FALSE;
}
}

public function email_exists($str)
{
if ($this->user_model->check_email_address($str))
{
return TRUE;
}
else
{
$this->form_validation->set_message('email_exists', 'The email address is
not stored in the database. ');
return FALSE;
}
}

public function email_exists_first($str)
{
if ($this->user_model->check_email_address_first($str))
{
return TRUE;
}
else
{
$this->form_validation->set_message('email_exists_first', 'The email
address is already verified. ');
return FALSE;
}
}

public function edit_student_details($student_id){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'Edit Student Details';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['student_details'] = $this->student_model->get_details_by_id(
$student_id);
$data['assigned_courses'] = $this->student_model->get_assigned_courses(
$student_id);
$data['courses'] = $this->course_model->get_list_of_courses();
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/admin_sidebar',$data);
$this->load->view('edit_student_details',$data);
$this->load->view('templates/JScscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_flashdata('error_404','True');
redirect('','refresh');
}
} else {
}
}

} else {
redirect('','refresh');
}
}

public function update_student() {
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
email_verified'] === "YES"){
$this->form_validation->set_rules('student_first_name','User First Name',
'required|trim|min_length[2]|max_length[64]');
$this->form_validation->set_rules('student_last_name','User Last Name',
'required|trim|min_length[2]|max_length[64]');
$this->form_validation->set_rules('student_email_address','User Email
Address', 'required|trim|valid_email');
if ($this->form_validation->run() === FALSE){ // inputs didn't pass
validation
$this->session->set_flashdata('update_student_error', validation_errors())
;
redirect('edit_student_details/'.$this->input->post('student_id'),'refresh
');
} else {
$array = array('test','test');
$prev_assigned_categories = $this->student_model->get_assigned_courses(
$this->input->post('student_id'));
$this->student_model->update_student($this->input->post('student_id'),
$prev_assigned_categories);
$this->user_model->send_verify_mail(array($this->input->post('
student_email_address')));
$this->session->set_flashdata('update_student_succeed', $array);
redirect('edit_student_details/'.$this->input->post('student_id'),'refresh
');
}
} else {
redirect('','refresh');
}
}

public function edit_teacher_details($teacher_id){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'Edit Teacher Details';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['teacher_details'] = $this->teacher_model->get_details_by_id(
$teacher_id);
$data['assigned_courses'] = $this->teacher_model->get_assigned_courses(
$teacher_id);
$data['courses'] = $this->course_model->get_list_of_courses();
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/admin_sidebar',$data);
$this->load->view('edit_teacher_details',$data);
$this->load->view('templates/JScscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_flashdata('error_404','True');
redirect('','refresh');
}
} else {
redirect('','refresh');
}
}
}

```



```

}
public function update_teacher() {
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
email_verified'] === "YES"){
$this->form_validation->set_rules('teacher_first_name','User_FirstName',
'required|trim|min_length[2]|max_length[64]');
$this->form_validation->set_rules('teacher_last_name','User_LastName',
'required|trim|min_length[2]|max_length[64]');
$this->form_validation->set_rules('teacher_email_address','User_Email
Address', 'required|trim|valid_email');
if ($this->form_validation->run() === FALSE){ // inputs didn't pass
validation
$this->session->set_flashdata('update_teacher_error', validation_errors())
;
redirect('edit_teacher_details/'.$this->input->post('teacher_id'),'refresh
');
}else{
$prev_assigned_categories = $this->teacher_model->get_assigned_courses(
$this->input->post('teacher_id'));
$this->teacher_model->update_teacher($this->input->post('teacher_id'),
$prev_assigned_categories);
$this->user_model->send_verify_mail(array($this->input->post('
teacher_email_address')));
$this->session->set_flashdata('update_teacher_succeed', 'Success!');
redirect('edit_teacher_details/'.$this->input->post('teacher_id'),'refresh
');
}
} else {
redirect('', 'refresh');
}
}
public function edit_course_details($course_id){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'Edit_Course_Details';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['course_details'] = $this->course_model->get_details_by_id(
$course_id);
$data['teacher'] = $this->teacher_model->get_list_of_teachers();
$data['course_category'] = $this->course_model->get_list_of_categories();
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/admin_sidebar',$data);
$this->load->view('edit_course_details',$data);
$this->load->view('templates/JScscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_flashdata('error_404','True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}
public function update_course(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
email_verified'] === "YES"){
$this->form_validation->set_rules('course_short_name','Course_ShortName',
'required|trim|min_length[3]|max_length[20]',array('min_length' =>
'{field}_must_have_at_least_{param}_characters.','max_length' => '{
field}_must_not_exceed_{param}_characters.'));
$this->form_validation->set_rules('course_full_name','Course_FullName',
'required|trim|min_length[3]|max_length[64]',array('min_length' => '{
field}_must_have_at_least_{param}_characters.','max_length' => '{
field}_must_not_exceed_{param}_characters.'));
$this->form_validation->set_rules('section_name','SectionName', 'required
|trim|callback_section_check|min_length[3]|max_length[64]',array('
min_length' => '{field}_must_have_at_least_{param}_characters.','
max_length' => '{field}_must_not_exceed_{param}_characters.'));
if ($this->form_validation->run() === FALSE){ // inputs didn't pass
validation
$this->session->set_flashdata('course_update_error', validation_errors());
redirect('edit_course_details/'.$this->input->post('course_id'),'refresh')
;
}else{
$this->course_model->update_course($this->input->post('course_id'));
$this->session->set_flashdata('course_update_succeed', 'Success!');
redirect('edit_course_details/'.$this->input->post('course_id'),'refresh')
;
}
} else {
redirect('', 'refresh');
}
}
public function edit_course_category($course_category_id){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'Update_Course_Category';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['category_details'] = $this->course_model->get_category_details(
$course_category_id);
$data['assigned_courses'] = $this->course_model->
get_assigned_courses_by_category($course_category_id);
$data['courses'] = $this->course_model->get_list_of_courses();
$data['teacher'] = $this->teacher_model->get_list_of_teachers();
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/admin_sidebar',$data);
$this->load->view('edit_course_category',$data);
$this->load->view('templates/JScscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_flashdata('error_404','True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}
public function update_course_category(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
email_verified'] === "YES"){
$this->form_validation->set_rules('course_category_name','Course_Category
Name', 'required|trim|callback_update_coursecat_check|min_length[3]|

```

```

        max_length[64]', array('min_length' => '{field}_must_have_at_least_{
        param}_characters.', 'max_length' => '{field}_must_not_exceed_{param}
        _characters.');
```

```

if ($this->form_validation->run() == FALSE) { // inputs didn't pass
    validation

$this->session->set_flashdata('course_cat_update_error', validation_errors
());
redirect('edit_course_category/'. $this->input->post('course_category_id'),
'refresh');
} else {
$assigned_courses = $this->course_model->get_assigned_courses_by_category(
$this->input->post('course_category_id'));
$this->course_model->update_course_category($this->input->post('
course_category_id'), $assigned_courses);
$this->session->set_flashdata('course_cat_update_succeed', 'Success!');
redirect('edit_course_category/'. $this->input->post('course_category_id'),
'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function edit_student_group($student_group_id){
if($this->session->userdata('logged_in')){
if($SESSION['logged_in']['email_verified'] === "YES" && $SESSION['
logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'Update_Student_Group';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['group_details'] = $this->student_model->get_group_details(
$student_group_id);
$data['assigned_students'] = $this->student_model->
get_assigned_students_by_group($student_group_id);
$data['students'] = $this->student_model->get_list_of_students();
$this->load->view('templates/admin_header', $data);
$this->load->view('templates/admin_sidebar', $data);
$this->load->view('edit_student_group', $data);
$this->load->view('templates/JSscripts', $data);
$this->load->view('templates/admin_footer', $data);
} else {
$this->session->set_flashdata('error_404', 'True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function update_student_group(){
if($this->session->userdata('logged_in') && $SESSION['logged_in']['
email_verified'] === "YES"){
$this->form_validation->set_rules('student_group_name', 'Student_Group_Name
', 'required|trim|min_length[3]|max_length[64]|
callback_update_groupname_check', array('min_length' => '{field}_must
_have_at_least_{param}_characters.', 'max_length' => '{field}_must_
not_exceed_{param}_characters.');
```

```

if ($this->form_validation->run() == FALSE) { // inputs didn't pass
    validation

$this->session->set_flashdata('update_student_group_error',
validation_errors());
redirect('edit_student_group/'. $this->input->post('student_group_id'),
'refresh');
} else {
$assigned_students = $this->student_model->get_assigned_students_by_group(
$this->input->post('student_group_id'));
$this->student_model->update_student_group($this->input->post('
student_group_id'), $assigned_students);
$this->session->set_flashdata('update_student_group_succeed', 'Success!');
redirect('edit_student_group/'. $this->input->post('student_group_id'),
'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function course_students($course_id){
if($this->session->userdata('logged_in')){
if($SESSION['logged_in']['email_verified'] === "YES" && $SESSION['
logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'Course_Students_List';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['course_details'] = $this->course_model->get_details_by_id(
$course_id);
$data['students'] = $this->student_model->get_list_of_students_by_course(
$course_id);
$this->load->view('templates/admin_header', $data);
$this->load->view('templates/admin_sidebar', $data);
$this->load->view('course_students_list', $data);
$this->load->view('templates/JSscripts', $data);
$this->load->view('templates/admin_footer', $data);
} else {
$this->session->set_flashdata('error_404', 'True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function enrol_students($course_id){
if($this->session->userdata('logged_in')){
if($SESSION['logged_in']['email_verified'] === "YES" && $SESSION['
logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'Enrol_Students';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['course_details'] = $this->course_model->get_details_by_id(
$course_id);
$data['assigned_students'] = $this->student_model->
get_list_of_students_id_by_course($course_id);
$data['students'] = $this->student_model->get_list_of_students();
$this->load->view('templates/admin_header', $data);
$this->load->view('templates/admin_sidebar', $data);
$this->load->view('enrol_students', $data);
}
}
}
}

```

```

$this->load->view('templates/JSscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_tempdata('error_404','True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function enrol_student_groups($course_id){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'Enrol student groups';
$data['user_details'] = $this->user_model->get_user_details_by_id($_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['user_details']['school_id']);
$data['course_details'] = $this->course_model->get_details_by_id($course_id);
$assigned_students = $this->student_model->get_list_of_students_id_by_course($course_id);
if(count($assigned_students) == 0) {
$assigned_students = array(0);
}

$data['student_groups'] = $this->student_model->get_student_group_members($assigned_students);
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/admin_sidebar',$data);
$this->load->view('enrol_student_groups',$data);
$this->load->view('templates/JSscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_tempdata('error_404','True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function assign_student_groups(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['email_verified'] === "YES"){
if(count($this->input->post('student_groups')) == 0){ // inputs didn't pass validation
redirect('course_students/'.$this->input->post('course_id'),'refresh');
}else{
$assigned_students = $this->student_model->get_list_of_students_id_by_course($this->input->post('course_id'));
if(count($assigned_students) == 0) {
$assigned_students = array(0);
}

$input = $this->input->post('student_groups');
$students = $this->student_model->get_student_group_members_by_ids($input, $assigned_students);
$this->course_model->assign_students_to_course($students, $this->input->post('course_id'));
$this->session->set_flashdata('enrol_student_group_succeed', 'Success!');
redirect('course_students/'.$this->input->post('course_id'),'refresh');
}
}
}

}
} else {
redirect('', 'refresh');
}
}

public function assign_students(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['email_verified'] === "YES"){
if(count($this->input->post('selected_students')) == 0){ // inputs didn't pass validation
redirect('course_students/'.$this->input->post('course_id'),'refresh');
}else{
$assigned_students = $this->student_model->get_list_of_students_id_by_course($this->input->post('course_id'));
if(count($assigned_students) == 0) {
$assigned_students = array(0);
}

$input = $this->input->post('selected_students');
$this->course_model->assign_students_to_course($input, $this->input->post('course_id'));
$this->grade_model->update_students_in_class_record($this->input->post('course_id'));
$this->session->set_flashdata('enrol_student_succeed', 'Success!');
redirect('course_students/'.$this->input->post('course_id'),'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function delete_student(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['email_verified'] === "YES"){
$this->student_model->remove_student_from_database($this->input->post('student_id'));
$this->session->set_flashdata('delete_student_succeed', 'Success!');
redirect('student_list','refresh');
} else {
redirect('', 'refresh');
}
}

public function delete_teacher(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['email_verified'] === "YES"){
$this->teacher_model->remove_teacher_from_database($this->input->post('teacher_id'));
$this->session->set_flashdata('delete_teacher_succeed', 'Success!');
redirect('teacher_list','refresh');
} else {
redirect('', 'refresh');
}
}

public function delete_course_category(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['email_verified'] === "YES"){
$this->course_model->remove_course_category($this->input->post('cc_id'));
$this->session->set_flashdata('delete_course_category_succeed', 'Success!');
};
redirect('course_category_list','refresh');
} else {
redirect('', 'refresh');
}
}
}

```

```

}
}
public function delete_course(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['email_verified'] === "YES"){
$this->course_model->remove_course($this->input->post('course_id'));
$this->session->set_flashdata('delete_course_succeed', 'Success!');
redirect('course_list','refresh');
} else {
redirect('', 'refresh');
}
}
public function clear_course_record(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['email_verified'] === "YES"){
$this->course_model->clear_course_record($this->input->post('course_id'));
$this->session->set_flashdata('clear_course_record_succeed', 'Success!');
redirect('edit_course_details/'.$this->input->post('course_id'),'refresh');
};
} else {
redirect('', 'refresh');
}
}
public function remove_student(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['email_verified'] === "YES"){
$this->course_model->remove_student_from_course($this->input->post('student_id'),$this->input->post('course_id'));
$this->course_model->update_order_in_course($this->input->post('course_id'));
$this->grade_model->update_students_in_class_record($this->input->post('course_id'));
$this->session->set_flashdata('remove_student_succeed', 'Success!');
redirect('course_students/'.$this->input->post('course_id'),'refresh');
} else {
redirect('', 'refresh');
}
}
public function remove_course_from_category(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['email_verified'] === "YES"){
$this->course_model->remove_course_from_category($this->input->post('course_id'),$this->input->post('category_id'));
$this->session->set_flashdata('remove_course_succeed', 'Success!');
redirect('category_courses/'.$this->input->post('category_id'),'refresh');
} else {
redirect('', 'refresh');
}
}
public function remove_student_group(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['email_verified'] === "YES"){
$this->student_model->remove_student_group($this->input->post('student_group_id'),$_SESSION['logged_in']['school_id']);
$this->session->set_flashdata('remove_student_group_succeed', 'Success!');
redirect('student_groups_list','refresh');
} else {
redirect('', 'refresh');
}
}
}
}

```

Ajax Controller

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Ajax extends CI_Controller {
public function resend_email()
{
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "NO"){
$email = $this->input->post('email');
$this->user_model->send_verify_mail($email);
$this->session->set_flashdata('resend_succeed', 'Email reset!');
} else {
redirect('home','refresh');
}
} else {
redirect('', 'refresh');
}
}
public function get_class_record($course_id){
$data['record'] = $this->student_model->
get_class_record_details_by_course_id($course_id);
echo json_encode($data['record']);
}
public function save_class_record(){
$this->grade_model->save_class_record($_POST['record'], $_POST['course_id']);
echo '<pre>';
print_r($_POST['record']);
echo '</pre>';
//echo stripslashes(json_encode($_POST['record']));
}
}
}

```

Navigation Controller

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Navigation extends CI_Controller {
public function index()
{
if(!$this->session->userdata('logged_in')){
$data['title'] = 'Welcome';
$this->load->view('templates/header',$data);
$this->load->view('login',$data);
$this->load->view('templates/footer',$data);
} else {
redirect('home','refresh');
}
}
public function register(){
if(!$this->session->userdata('logged_in')){
$data['title'] = 'Admin Register';
$this->load->view('templates/header',$data);
$this->load->view('register',$data);
$this->load->view('templates/footer',$data);
} else {
}
}
}

```

```

redirect('home','refresh');
}
}
public function forgot_pass(){
if(!$this->session->userdata('logged_in')){
$data['title'] = 'Forgot_Password?';
$this->load->view('templates/header',$data);
$this->load->view('forgot_password',$data);
$this->load->view('templates/footer',$data);
} else {
redirect('home','refresh');
}
}
public function validate_email(){
if(!$this->session->userdata('logged_in')){
$data['title'] = 'Validate_Email';
$this->load->view('templates/header',$data);
$this->load->view('validate_email',$data);
$this->load->view('templates/footer',$data);
} else {
redirect('home','refresh');
}
}
public function home(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "NO"){
redirect('verify_email','refresh');
} else {
if($_SESSION['logged_in']['user_role'] === "ADMIN"){
$data['title'] = 'Admin_Dashboard';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/admin_sidebar',$data);
$this->load->view('admin_home',$data);
$this->load->view('templates/JSscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else if($_SESSION['logged_in']['user_role'] === "TEACHER") {
$teacher_data = $this->teacher_model->get_details_by_user_id($_SESSION['
logged_in']['user_id']);
if($teacher_data['first_login'] == "YES"){
redirect('force_password_change','refresh');
} else {
$data['title'] = 'Dashboard';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['teacher'] = $this->teacher_model->get_details_by_user_id($data['
user_details']['user_id']);
$data['my_courses'] = $this->course_model->get_all_courses_by_teacher(
$data['teacher']['teacher_id']);
$data['categories'] = $this->course_model->
get_list_of_categories_by_teacher($data['teacher']['teacher_id']);
//$data['course_list'] = $this->course_model->get_course_by_teacher($data
['teacher']['teacher_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/teacher_sidebar',$data);

```

```

$this->load->view('teacher_home',$data);
$this->load->view('templates/JSscripts',$data);
$this->load->view('templates/admin_footer',$data);
}
} else if($_SESSION['logged_in']['user_role'] === "STUDENT") {
$student_data = $this->student_model->get_details_by_user_id($_SESSION['
logged_in']['user_id']);
if($student_data['first_login'] == "YES"){
redirect('force_password_change','refresh');
} else {
$data['title'] = 'Student_Dashboard';
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['student'] = $this->student_model->get_details_by_user_id($data['
user_details']['user_id']);
$data['my_courses'] = $this->course_model->get_all_courses_by_student(
$data['student']['student_id']);
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/student_sidebar',$data);
$this->load->view('student_home',$data);
$this->load->view('templates/JSscripts',$data);
$this->load->view('templates/admin_footer',$data);
}
}
} else {
redirect('', 'refresh');
}
}
public function logout(){
$this->session->unset_userdata('logged_in');
session_destroy();
redirect('');
}
}
} else {

```

Student Controller

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Student extends CI_Controller {
public function force_password_change(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES"){
$data['title'] = 'Change_password_(for_first-time_system_login)';
if($_SESSION['logged_in']['user_role'] === "TEACHER") {
$data['user_details'] = $this->teacher_model->get_details_by_user_id(
$_SESSION['logged_in']['user_id']);
} else if($_SESSION['logged_in']['user_role'] === "STUDENT"){
$data['user_details'] = $this->student_model->get_details_by_user_id(
$_SESSION['logged_in']['user_id']);
}
}
if($data['user_details']['first_login'] === "YES") {
$this->load->view('templates/header',$data);
$this->load->view('force_password',$data);
$this->load->view('templates/footer',$data);
}
} else {

```

```

redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function update_force_password(){
if($this->session->userdata('logged_in')){
$this->form_validation->set_rules('user_password', 'User Password', '
    required|trim|alpha_numeric|min_length[6]|max_length[20]', array('
    min_length' => '{field}_must_have_{param}-20_characters.', '
    max_length' => '{field}_must_have_{param}_characters.'));
if ($this->form_validation->run() === FALSE){ // inputs didn't pass
    validation

$this->session->set_flashdata('force_update_error', validation_errors());
redirect('force_password_change', 'refresh');
}else{
$this->user_model->update_force_password($this->input->post('user_name'),
    $this->input->post('user_password'), $SESSION['logged_in']['
    user_role'], $SESSION['logged_in']['user_id']);
$this->session->set_flashdata('login_succeed', 'Success!');
redirect('home', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function my_course_record($course_id){
if($this->session->userdata('logged_in')){
if($SESSION['logged_in']['email_verified'] === "YES" && $SESSION['
    logged_in']['user_role'] === "STUDENT"){
$data['user_details'] = $this->user_model->get_user_details_by_id(
    $SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
    user_details']['school_id']);
$data['student'] = $this->student_model->get_details_by_user_id($data['
    user_details']['user_id']);
$data['my_courses'] = $this->course_model->get_all_courses_by_student(
    $data['student']['student_id']);
$data['category_total'] = $this->grade_model->get_category_total_mapping(
    $course_id);
// $data['course_list'] = $this->course_model->get_course_by_teacher($data
    ['teacher']['teacher_id']);
$data['course_details'] = $this->course_model->get_course_details_by_id(
    $course_id);
$data['title'] = 'My Course Record - ' . $data['course_details']['
    course_short_name'];
$data['class_record'] = $this->student_model->
    get_grades_stats_by_course_id($course_id);
$data['order_number'] = $this->student_model->get_order_number_in_course(
    $course_id, $data['student']['student_id']);
$data['trimmed_index'] = $this->grade_model->get_trimmed_indices(
    $course_id, $data['student']['student_id']);
$this->load->view('templates/admin_header', $data);
$this->load->view('templates/student_sidebar', $data);
$this->load->view('my_course_record', $data);
$this->load->view('templates/JSScripts', $data);
$this->load->view('templates/admin_footer', $data);
} else {
$this->session->set_tempdata('error_404', 'True');

```

```

redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Teacher extends CI_Controller {
public function force_password_change(){
if($this->session->userdata('logged_in')){
if($SESSION['logged_in']['email_verified'] === "YES"){
$data['title'] = 'Change Password (for first-time system login)';
if($SESSION['logged_in']['user_role'] === "TEACHER") {
$data['user_details'] = $this->teacher_model->get_details_by_user_id(
    $SESSION['logged_in']['user_id']);
} else if($SESSION['logged_in']['user_role'] === "STUDENT"){
$data['user_details'] = $this->student_model->get_details_by_user_id(
    $SESSION['logged_in']['user_id']);
}
}
if($data['user_details']['first_login'] === "YES") {
$this->load->view('templates/header', $data);
$this->load->view('force_password', $data);
$this->load->view('templates/footer', $data);
}
} else {
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function update_force_password(){
if($this->session->userdata('logged_in')){
$this->form_validation->set_rules('user_password', 'User Password', '
    required|trim|alpha_numeric|min_length[6]|max_length[20]', array('
    min_length' => '{field}_must_have_{param}-20_characters.', '
    max_length' => '{field}_must_have_{param}_characters.'));
if ($this->form_validation->run() === FALSE){ // inputs didn't pass
    validation

$this->session->set_flashdata('force_update_error', validation_errors());
redirect('force_password_change', 'refresh');
}else{
$this->user_model->update_force_password($this->input->post('user_name'),
    $this->input->post('user_password'), $SESSION['logged_in']['
    user_role'], $SESSION['logged_in']['user_id']);
$this->session->set_flashdata('login_succeed', 'Success!');
redirect('home', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function gradebook_setup_list(){
if($this->session->userdata('logged_in')){

```

Teacher Controller

```

if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['logged_in']['user_role'] === "TEACHER"){
    $data['title'] = 'Assigned Course Categories';
    $data['user_details'] = $this->user_model->get_user_details_by_id($_SESSION['logged_in']['user_id']);
    $data['school_name'] = $this->user_model->get_school_name($data['user_details']['school_id']);
    $data['teacher_details'] = $this->teacher_model->get_details_by_user_id($data['user_details']['user_id']);
    $data['my_courses'] = $this->course_model->get_all_courses_by_teacher($data['teacher_details']['teacher_id']);
    $data['categories'] = $this->course_model->get_list_of_categories_by_teacher($data['teacher_details']['teacher_id']);
    $array = array();
    $index = 0;
    foreach($data['categories'] as $row){
        $array[$index] = $row['course_category_id'];
        $index++;
    }
    $data['courses'] = $this->course_model->get_list_of_courses_by_teacher($array);
    $data['scale_templates'] = $this->grade_model->get_scale_templates($_SESSION['logged_in']['school_id'], $data['teacher_details']['teacher_id']);
    $this->load->view('templates/admin_header', $data);
    $this->load->view('templates/teacher_sidebar', $data);
    $this->load->view('gradebook_setup_list', $data);
    $this->load->view('templates/JSscripts', $data);
    $this->load->view('templates/admin_footer', $data);
} else {
    $this->session->set_tempdata('error_404', 'True');
    redirect('', 'refresh');
}
} else {
    redirect('', 'refresh');
}
}

public function gradebook_setup($course_id){
    if($this->session->userdata('logged_in')){
        if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['logged_in']['user_role'] === "TEACHER"){
            $data['title'] = 'Gradebook Setup';
            $data['user_details'] = $this->user_model->get_user_details_by_id($_SESSION['logged_in']['user_id']);
            $data['teacher_details'] = $this->teacher_model->get_details_by_user_id($data['user_details']['user_id']);
            $data['my_courses'] = $this->course_model->get_all_courses_by_teacher($data['teacher_details']['teacher_id']);
            $data['school_name'] = $this->user_model->get_school_name($data['user_details']['school_id']);
            $data['course_details'] = $this->course_model->get_details_by_id($course_id);
            $data['gradebook_details'] = $this->course_model->get_gradebook_by_course($course_id);
            $data['gradebook_categories'] = $this->grade_model->get_gradebook_main_category_details($data['gradebook_details']['gradebook_setup_details_id']);
            $main_category_ids = $this->grade_model->get_gradebook_main_categories($data['gradebook_details']['gradebook_setup_details_id']);

```

```

if(count($main_category_ids) == 0){
    $main_category_ids = array(0);
}
}
    $data['category_items'] = $this->grade_model->get_category_items_of_gradebook($main_category_ids, $data['gradebook_details']['gradebook_setup_details_id']);
    $data['grade_item_names'] = $this->grade_model->get_grade_item_names($data['gradebook_details']['gradebook_setup_details_id']);
    $data['grade_item_categories'] = $this->grade_model->get_grade_item_categories($data['gradebook_details']['gradebook_setup_details_id']);
    $data['category_names'] = $this->grade_model->get_category_names($data['gradebook_details']['gradebook_setup_details_id']);
    $data['gradebook_templates'] = $this->grade_model->get_existing_templates($_SESSION['logged_in']['school_id'], $data['teacher_details']['teacher_id']);
    $data['grading_scales'] = $this->grade_model->get_grading_scales($course_id);
    $data['scale_templates'] = $this->grade_model->get_scale_templates($_SESSION['logged_in']['school_id'], $data['teacher_details']['teacher_id']);
    $data['categories'] = $this->course_model->get_list_of_categories_by_teacher($data['teacher_details']['teacher_id']);
    $this->session->unset_userdata('move_to_category');
    $this->load->view('templates/admin_header', $data);
    $this->load->view('templates/teacher_sidebar', $data);
    $this->load->view('gradebook_setup', $data);
    $this->load->view('templates/JSscripts', $data);
    $this->load->view('templates/admin_footer', $data);
} else {
    $this->session->set_tempdata('error_404', 'True');
    redirect('', 'refresh');
}
} else {
    redirect('', 'refresh');
}
}

public function update_gradebook_settings(){
    if($this->session->userdata('logged_in') && $_SESSION['logged_in']['email_verified'] === "YES"){
        $this->form_validation->set_rules('allow_teacher', 'Allow Teacher to Update Gradebook', 'required|trim');
        if ($this->form_validation->run() === FALSE) { // inputs didn't pass validation
            $this->session->set_flashdata('update_gradebook_settings_error', validation_errors());
            //redirect('gradebook_setup/'.$this->input->post('course_id'), 'refresh');
        }else{
            $this->grade_model->update_gradebook_settings($this->input->post('course_id'));
            $this->session->set_flashdata('update_gradebook_settings_succeed', 'Success!');
            redirect('gradebook_setup/'.$this->input->post('course_id'), 'refresh');
        }
    } else {
        redirect('', 'refresh');
    }
}

public function drop_student(){

```

```

if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
    email_verified'] === "YES"){
$this->form_validation->set_rules('student_id','Student_ID', 'required|
    trim');
$this->form_validation->set_rules('course_id','Course_ID', 'required|trim
    ');
if ($this->form_validation->run() === FALSE){ // inputs didn't pass
    validation
$this->session->set_flashdata('drop_student_error', validation_errors());
//redirect('gradebook_setup/'.$this->input->post('course_id'),'refresh');
}else{
$this->student_model->drop_student($this->input->post('course_id'), $this
->input->post('student_id'));
$this->session->set_flashdata('drop_student_succeed', 'Success!');
redirect('course_record_info/'.$this->input->post('course_id'),'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function modify_gradebook_categories($setup_id){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
logged_in']['user_role'] === "TEACHER"){
$data['title'] = 'Modify_Gradebook_Categories';
$data['user_details'] = $this->user_model->get_user_details_by_id(
    $_SESSION['logged_in']['user_id']);
$data['teacher_details'] = $this->teacher_model->get_details_by_user_id(
    $data['user_details']['user_id']);
$data['my_courses'] = $this->course_model->get_all_courses_by_teacher(
    $data['teacher_details']['teacher_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['gradebook_details'] = $this->course_model->
    get_gradebook_by_setup_id($setup_id);
$data['course_details'] = $this->course_model->get_details_by_id($data['
gradebook_details']['course_id']);
$data['gradebook_categories'] = $this->grade_model->
    get_gradebook_main_category_details($data['gradebook_details']['
    gradebook_setup_details_id']);
$data['gradebook_count'] = count($data['gradebook_categories']);
$data['categories'] = $this->course_model->
    get_list_of_categories_by_teacher($data['teacher_details']['
    teacher_id']);
if($this->session->userdata('move_to_category')){
$items = $this->session->userdata('move_to_category');
$grade_items = array();
foreach($items as $key => $value){
$row_exp = explode('_', $value);
if($row_exp[1] == "I")
$grade_items[] = $row_exp[0];
}
if(count($grade_items) == 0){
$grade_items = array(0);
}
}
}
}
}
}

$this->load->view('templates/admin_header', $data);
$this->load->view('templates/teacher_sidebar', $data);

$this->load->view('modify_gradebook_categories', $data);
$this->load->view('templates/JSscripts', $data);
$this->load->view('templates/admin_footer', $data);
} else {
$this->session->set_flashdata('error_404', 'True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function modify_grade_category_template($gradebook_id, $method,
    $category_ids, $category_type, $category_weight, $category_total,
    $category_names){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
    email_verified'] === "YES"){
$this->form_validation->set_rules('category_name[]', 'Category_Name', 'trim
    ');
if ($this->form_validation->run() === FALSE){ // inputs didn't pass
    validation
$this->session->set_flashdata('modify_categories_error', validation_errors
    ());
redirect('modify_gradebook_categories/'.$this->input->post('gradebook_id')
    , 'refresh');
}else{
$main_category_ids = $this->grade_model->get_gradebook_main_categories(
    $gradebook_id);
$main_item_ids = $this->grade_model->get_gradebook_main_items(
    $gradebook_id);
$gradebook_count = count($main_category_ids); // existing
//echo ' ===== <br/>';
//print_r($this->input->post('category_ids'));
//echo ' <br/> ===== <br/>';
$stest = $category_ids;
$stest_item = $category_ids;
// input
if(count($this->input->post('category_ids')) > 0){
$input_category_index = array_keys(array_filter($category_type, function(
    $type){
return $type === "CATEGORY";
}));
$input_item_index = array_keys(array_filter($category_type, function($type)
    {
return $type === "ITEM";
}));
/*echo 'Index Ids of Category Input: ';
print_r($input_category_index);
echo '<br/>';
echo 'Index Ids of Item Input: ';
print_r($input_item_index);
echo '<br/>';*/
foreach($input_item_index as $id){
unset($stest[$id]);
}
foreach($input_category_index as $id2){
unset($stest_item[$id2]);
}
}
}
}
}

$stest_count = count($stest);
/*echo 'Existing Item Ids: ';

```



```

print_r($main_item_ids);
echo '<br/>';
echo 'Existing Category Ids: ';
print_r($main_category_ids);
echo '<br/>';
echo 'Input Category Ids: ';
print_r($test);
echo '<br/>';*/
$item_check = 0;
$new_inputs_array = array();
$sold_inputs_array = array();
if($test_count > 0) {
    $new_inputs_array = array_keys($test);
    //echo 'New Input Array Keys: ';
    //print_r($new_inputs_array); // new categories / update for existing if
    // necessary
    //echo '<br/>';
    if($gradebook_count > 0) {
        $sold_inputs_array = array_keys(array_filter($test,function($item){
            return $item != 0;
        }));
        //echo 'Old Input Array Keys: ';
        //print_r($sold_inputs_array); // existing categories for update
        //echo '<br/>';
    }
    } else {
        $test = array(0);
    }
    if(count($test_item) == 0){
        $test_item = array(0);
        $item_check = 1;
    }
    if(count($sold_inputs_array) < $gradebook_count){
        $removed_inputs = array_diff($main_category_ids,$test);
        //echo 'Removed Inputs: ';
        //print_r($removed_inputs);
        //echo '<br/>';
        $remove_ids = array_keys($removed_inputs);
    } else {
        $remove_ids = array(0);
    }
    //echo 'Removed Ids: ';
    //print_r($remove_ids);
    //echo '<br/>';
    if($test_count == 0){
        //Case 1: No Input Delete All
        $this->grade_model->delete_gradebook_categories($gradebook_id);
    } else if($gradebook_count == 0 && $test_count > 0){
        //Case 2: No existing categories/ Insert All
        $array_categories = $this->grade_model->
            insert_gradebook_categories_template($gradebook_id,$test_count,
                $new_inputs_array, $category_names, $category_type);
    } else if(count($sold_inputs_array) == 0 && $test_count > 0) {
        //Case 3: Delete All then add ALL New Categories
        $this->grade_model->delete_gradebook_categories($gradebook_id);
        $this->grade_model->insert_gradebook_categories($gradebook_id,$test_count,
            $new_inputs_array);
    } else if(count($sold_inputs_array) < $gradebook_count && count(
        $new_inputs_array) == 0){
        //Case 4: Delete/Update Some with No New Categories
        $this->grade_model->delete_gradebook_categories($gradebook_id,count(
            $removed_inputs),$removed_inputs);
        $this->grade_model->update_gradebook_categories($gradebook_id,count(
            $sold_inputs_array),$sold_inputs_array);
    } else if(count($sold_inputs_array) == $gradebook_count && count(
        $new_inputs_array) == 0) {
        //Case 5: Update all with NO Delete - No Insert
        $this->grade_model->update_gradebook_categories($gradebook_id,count(
            $sold_inputs_array),$sold_inputs_array);
    } else if(count($sold_inputs_array) == $gradebook_count && count(
        $new_inputs_array) > 0) {
        //Case 6: Update All Existing and Add New Ones
        $this->grade_model->update_gradebook_categories($gradebook_id,count(
            $sold_inputs_array),$sold_inputs_array);
        $this->grade_model->insert_gradebook_categories($gradebook_id,count(
            $new_inputs_array),$new_inputs_array);
    } else if(count($sold_inputs_array) < $gradebook_count && count(
        $new_inputs_array) > 0){
        //Case 7: Delete/Update Some with New Categories
        $this->grade_model->delete_gradebook_categories($gradebook_id,count(
            $removed_inputs),$removed_inputs);
        $this->grade_model->update_gradebook_categories($gradebook_id,count(
            $sold_inputs_array),$sold_inputs_array);
        $this->grade_model->insert_gradebook_categories($gradebook_id,count(
            $new_inputs_array),$new_inputs_array);
    }
    $this->modify_grade_items_template($gradebook_id,$test_item,$item_check,
        $category_names, $category_total, $category_type);
    if($this->session->userdata('move_to_category')){
        $array = $this->input->post('grade_item_ids');
        $categories = $this->input->post('grade_item_categories');
        $categories = array_unique($categories);
        $this->grade_model->remove_transferred_items($array);
        $this->grade_model->update_categories_with_removed_items($categories,
            $gradebook_id);
        $this->grade_model->update_category_weights_with_removed_items(
            $gradebook_id,$categories);
        $this->session->unset_userdata('move_to_category');
    }
    $total_points = $this->grade_model->update_total_points($gradebook_id);
    $array_ids = $this->grade_model->get_gradebook_parameters($gradebook_id);
    $this->grade_model->update_method($gradebook_id,$method,$array_ids,
        $category_weight,$category_total,$total_points);
    $new_category_ids = $this->grade_model->get_gradebook_main_categories(
        $gradebook_id);
    $this->grade_model->update_category_weights_with_removed_items(
        $gradebook_id,$new_category_ids);
    $this->grade_model->update_final_grade($gradebook_id);
    //update weight depending on method
    $this->grade_model->update_mapping($gradebook_id);
    $this->grade_model->update_class_record($gradebook_id);
    return $array_categories;
}
} else {
    redirect('','refresh');
}
}
function modify_grade_items_template($gradebook_id,$test,$item_check,
    $names, $total, $type){

```

```

$main_item_ids = $this->grade_model->get_gradebook_main_items(
    $gradebook_id);
$gradebook_count = count($main_item_ids); // existing
if($item_check == 1) {
    $stest_count = 0;
} else {
    $stest_count = count($stest); // input
}
//echo 'Existing Item Ids: ';
//print_r($main_item_ids);
//echo '<br/>';
//echo 'Input Item Ids: ';
//print_r($stest);
//echo '<br/>';
$new_inputs_array = array();
$sold_inputs_array = array();
if($stest_count > 0) {
    $new_inputs_array = array_keys($stest);
    //echo 'New Input Array Keys: ';
    //print_r($new_inputs_array); // new categories / update for existing if
    // necessary
    //echo '<br/>';
}
if($gradebook_count > 0) {
    $sold_inputs_array = array_keys(array_filter($stest,function($item){
    return $item != 0;
}));
    //echo 'Old Input Array Keys: ';
    //print_r($sold_inputs_array); // existing categories for update
    //echo '<br/>';
}
if(count($sold_inputs_array) < $gradebook_count){
    $removed_inputs = array_diff($main_item_ids,$stest);
    //echo 'Removed Inputs: ';
    //print_r($removed_inputs);
    //echo '<br/>';
    $remove_ids = array_keys($removed_inputs);
} else {
    $remove_ids = array(0);
}
//echo 'Removed Ids: ';
//print_r($remove_ids);
//echo '<br/>';
//print_r($this->input->post('category_total'));
if($stest_count == 0){
    //Case 1: No Input Delete All
    $this->grade_model->delete_gradebook_items($gradebook_id);
} else if($gradebook_count == 0 && $stest_count > 0){
    //Case 2: No existing categories/ Insert All
    $this->grade_model->insert_gradebook_items_template($gradebook_id,
        $stest_count,$new_inputs_array, $names, $total, $type);
} else if(count($sold_inputs_array) == 0 && $stest_count > 0) {
    //Case 3: Delete All then add ALL New Categories
    $this->grade_model->delete_gradebook_items($gradebook_id);
    $this->grade_model->insert_gradebook_items($gradebook_id,$stest_count,
        $new_inputs_array);
} else if(count($sold_inputs_array) < $gradebook_count && count(
    $new_inputs_array) == 0){
    //Case 4: Delete/Update Some with No New Categories
    $this->grade_model->delete_gradebook_items($gradebook_id,count(
        $removed_inputs),$removed_inputs);
    $this->grade_model->update_gradebook_items($gradebook_id,count(
        $sold_inputs_array),$sold_inputs_array);
} else if(count($sold_inputs_array) == $gradebook_count && count(
    $new_inputs_array) == 0) {
    //Case 5: Update all with NO Delete - No Insert
    $this->grade_model->update_gradebook_items($gradebook_id,count(
        $sold_inputs_array),$sold_inputs_array);
} else if(count($sold_inputs_array) == $gradebook_count && count(
    $new_inputs_array) > 0) {
    //Case 6: Update All Existing and Add New Ones
    $this->grade_model->update_gradebook_items($gradebook_id,count(
        $sold_inputs_array),$sold_inputs_array);
    $this->grade_model->insert_gradebook_items($gradebook_id,count(
        $new_inputs_array),$new_inputs_array);
} else if(count($sold_inputs_array) < $gradebook_count && count(
    $new_inputs_array) > 0){
    //Case 7: Delete/Update Some with New Categories
    $this->grade_model->delete_gradebook_items($gradebook_id,count(
        $removed_inputs),$removed_inputs);
    $this->grade_model->update_gradebook_items($gradebook_id,count(
        $sold_inputs_array),$sold_inputs_array);
    $this->grade_model->insert_gradebook_items($gradebook_id,count(
        $new_inputs_array),$new_inputs_array);
}
}
}
public function modify_grade_category(){
    if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
        email_verified'] == "YES"){
    $this->form_validation->set_rules('category_name[]','Category_Name', 'trim
        ');
    if ($this->form_validation->run() == FALSE){ // inputs didn't pass
        validation
    $this->session->set_flashdata('modify_categories_error', validation_errors
        ());
    redirect('modify_gradebook_categories/'.$this->input->post('gradebook_id')
        ,'refresh');
    }else{
    $gradebook_id = $this->input->post('gradebook_id');
    $method = $this->input->post('method_of_computation');
    $main_category_ids = $this->grade_model->get_gradebook_main_categories(
        $this->input->post('gradebook_id'));
    $main_item_ids = $this->grade_model->get_gradebook_main_items($this->input
        ->post('gradebook_id'));
    $gradebook_count = count($main_category_ids); // existing
    //echo '==== <br/>';
    //print_r($this->input->post('category_ids'));
    //echo ' <br/> ==== <br/>';
    $stest = $this->input->post('category_ids');
    $stest_item = $this->input->post('category_ids');
    // input
    if(count($this->input->post('category_ids')) > 0){
    $input_category_index = array_keys(array_filter($this->input->post('
        category_type'),function($type){
    return $type == "CATEGORY";
    }));
    $input_item_index = array_keys(array_filter($this->input->post('
        category_type'),function($type){
    return $type == "ITEM";
    }));
    }
}
}

```

```

));
/*echo 'Index Ids of Category Input: ';
print_r($input_category_index);
echo '<br/>';
echo 'Index Ids of Item Input: ';
print_r($input_item_index);
echo '<br/>';*/
foreach($input_item_index as $id){
unset($test[$id]);
}
foreach($input_category_index as $id2){
unset($test_item[$id2]);
}
}
$test_count = count($test);
/*echo 'Existing Item Ids: ';
print_r($main_item_ids);
echo '<br/>';
echo 'Existing Category Ids: ';
print_r($main_category_ids);
echo '<br/>';
echo 'Input Category Ids: ';
print_r($test);
echo '<br/>';*/
$item_check = 0;
$new_inputs_array = array();
$sold_inputs_array = array();
if($test_count > 0) {
$new_inputs_array = array_keys($test,0);
// echo 'New Input Array Keys: ';
//print_r($new_inputs_array); // new categories / update for existing if
// necessary
// echo '<br/>';
if($gradebook_count > 0) {
$sold_inputs_array = array_keys(array_filter($test,function($item){
return $item != 0;
}));
//echo 'Old Input Array Keys: ';
//print_r($sold_inputs_array); // existing categories for update
//echo '<br/>';
}
} else {
$test = array(0);
}
if(count($test_item) == 0){
$test_item = array(0);
$item_check = 1;
}
if(count($sold_inputs_array) < $gradebook_count){
$removed_inputs = array_diff($main_category_ids,$test);
//echo 'Removed Inputs: ';
//print_r($removed_inputs);
//echo '<br/>';
$remove_ids = array_keys($removed_inputs);
} else {
$remove_ids = array(0);
}
//echo 'Removed Ids: ';
//print_r($remove_ids);
//echo '<br/>';
if($test_count == 0){
//Case 1: No Input Delete All
$this->grade_model->delete_gradebook_categories($gradebook_id);
} else if($gradebook_count == 0 && $test_count > 0){
//Case 2: No existing categories/ Insert All
//print_r($test_count);
//print_r($new_inputs_array);
$this->grade_model->insert_gradebook_categories($gradebook_id,$test_count,
$new_inputs_array);
} else if(count($sold_inputs_array) == 0 && $test_count > 0) {
//Case 3: Delete All then add ALL New Categories
$this->grade_model->delete_gradebook_categories($gradebook_id);
$this->grade_model->insert_gradebook_categories($gradebook_id,$test_count,
$new_inputs_array);
} else if(count($sold_inputs_array) < $gradebook_count && count(
$new_inputs_array) == 0){
//Case 4: Delete/Update Some with No New Categories
$this->grade_model->delete_gradebook_categories($gradebook_id,count(
$removed_inputs),$removed_inputs);
$this->grade_model->update_gradebook_categories($gradebook_id,count(
$sold_inputs_array),$sold_inputs_array);
} else if(count($sold_inputs_array) == $gradebook_count && count(
$new_inputs_array) == 0) {
//Case 5: Update all with NO Delete - No Insert
$this->grade_model->update_gradebook_categories($gradebook_id,count(
$sold_inputs_array),$sold_inputs_array);
} else if(count($sold_inputs_array) == $gradebook_count && count(
$new_inputs_array) > 0) {
//Case 6: Update All Existing and Add New Ones
$this->grade_model->update_gradebook_categories($gradebook_id,count(
$sold_inputs_array),$sold_inputs_array);
$this->grade_model->insert_gradebook_categories($gradebook_id,count(
$new_inputs_array),$new_inputs_array);
} else if(count($sold_inputs_array) < $gradebook_count && count(
$new_inputs_array) > 0){
//Case 7: Delete/Update Some with New Categories
$this->grade_model->delete_gradebook_categories($gradebook_id,count(
$removed_inputs),$removed_inputs);
$this->grade_model->update_gradebook_categories($gradebook_id,count(
$sold_inputs_array),$sold_inputs_array);
$this->grade_model->insert_gradebook_categories($gradebook_id,count(
$new_inputs_array),$new_inputs_array);
}
}
$this->modify_grade_items($gradebook_id,$test_item,$item_check);
if($this->session->userdata('move_to_category')){
$array = $this->input->post('grade_item_ids');
$categories = $this->input->post('grade_item_categories');
$categories = array_unique($categories);
$this->grade_model->remove_transferred_items($array);
$this->grade_model->update_categories_with_removed_items($categories,
$gradebook_id);
$this->grade_model->update_category_weights_with_removed_items(
$gradebook_id,$categories);
$this->session->unset_userdata('move_to_category');
}
$total_points = $this->grade_model->update_total_points($gradebook_id);
$array_ids = $this->grade_model->get_gradebook_parameters($gradebook_id);
$this->grade_model->update_method($gradebook_id,$method,$array_ids,$this->
input->post('category_weight'),$this->input->post('category_total'),
$total_points);

```

```

$new_category_ids = $this->grade_model->get_gradebook_main_categories(
    $this->input->post('gradebook_id'));
$this->grade_model->update_category_weights_with_removed_items(
    $gradebook_id,$new_category_ids);
$this->grade_model->update_final_grade($gradebook_id);
//update weight depending on method
$this->grade_model->update_mapping($gradebook_id);
$this->grade_model->update_class_record($gradebook_id);
$this->session->set_flashdata('modify_categories_succeed', 'Success!');
redirect('modify_gradebook_categories/'.$this->input->post('gradebook_id')
    , 'refresh');
}
} else {
redirect('', 'refresh');
}
}
function modify_grade_items($gradebook_id,$test,$item_check){
$main_item_ids = $this->grade_model->get_gradebook_main_items(
    $gradebook_id);
$gradebook_count = count($main_item_ids); // existing
if($item_check == 1) {
$test_count = 0;
} else {
$test_count = count($test); // input
}
//echo 'Existing Item Ids: ';
//print_r($main_item_ids);
//echo '<br/>';
//echo 'Input Item Ids: ';
//print_r($test);
//echo '<br/>';
$new_inputs_array = array();
$sold_inputs_array = array();
if($test_count > 0) {
$new_inputs_array = array_keys($test,0);
//echo 'New Input Array Keys: ';
//print_r($new_inputs_array); // new categories / update for existing if
    necessary
//echo '<br/>';
}
if($gradebook_count > 0) {
$sold_inputs_array = array_keys(array_filter($test,function($item){
return $item != 0;
}));
//echo 'Old Input Array Keys: ';
//print_r($sold_inputs_array); // existing categories for update
//echo '<br/>';
}
if(count($sold_inputs_array) < $gradebook_count){
$removed_inputs = array_diff($main_item_ids,$test);
//echo 'Removed Inputs: ';
//print_r($removed_inputs);
//echo '<br/>';
$remove_ids = array_keys($removed_inputs);
} else {
$remove_ids = array(0);
}
//echo 'Removed Ids: ';
//print_r($remove_ids);
//echo '<br/>';
}
//print_r($this->input->post('category_total'));
if($test_count == 0){
//Case 1: No Input Delete All
$this->grade_model->delete_gradebook_items($gradebook_id);
} else if($gradebook_count == 0 && $test_count > 0){
//Case 2: No existing categories/ Insert All
$this->grade_model->insert_gradebook_items($gradebook_id,$test_count,
    $new_inputs_array);
} else if(count($sold_inputs_array) == 0 && $test_count > 0) {
//Case 3: Delete All then add ALL New Categories
$this->grade_model->delete_gradebook_items($gradebook_id);
$this->grade_model->insert_gradebook_items($gradebook_id,$test_count,
    $new_inputs_array);
} else if(count($sold_inputs_array) < $gradebook_count && count(
    $new_inputs_array) == 0){
//Case 4: Delete/Update Some with No New Categories
$this->grade_model->delete_gradebook_items($gradebook_id,count(
    $removed_inputs),$removed_inputs);
$this->grade_model->update_gradebook_items($gradebook_id,count(
    $sold_inputs_array),$sold_inputs_array);
} else if(count($sold_inputs_array) == $gradebook_count && count(
    $new_inputs_array) == 0) {
//Case 5: Update all with NO Delete - No Insert
$this->grade_model->update_gradebook_items($gradebook_id,count(
    $sold_inputs_array),$sold_inputs_array);
} else if(count($sold_inputs_array) == $gradebook_count && count(
    $new_inputs_array) > 0) {
//Case 6: Update All Existing and Add New Ones
$this->grade_model->update_gradebook_items($gradebook_id,count(
    $sold_inputs_array),$sold_inputs_array);
$this->grade_model->insert_gradebook_items($gradebook_id,count(
    $new_inputs_array),$new_inputs_array);
} else if(count($sold_inputs_array) < $gradebook_count && count(
    $new_inputs_array) > 0){
//Case 7: Delete/Update Some with New Categories
$this->grade_model->delete_gradebook_items($gradebook_id,count(
    $removed_inputs),$removed_inputs);
$this->grade_model->update_gradebook_items($gradebook_id,count(
    $sold_inputs_array),$sold_inputs_array);
$this->grade_model->insert_gradebook_items($gradebook_id,count(
    $new_inputs_array),$new_inputs_array);
}
}
public function modify_gradebook_items($setup_id,$category_id){
if($this->session->userdata('logged_in')){
if($this->session->userdata('logged_in')['email_verified'] == "YES" && $this->session->
    userdata('logged_in')['user_role'] == "TEACHER"){
$data['title'] = 'Modify_<u>category_</u>';
$data['user_details'] = $this->user_model->get_user_details_by_id(
    $this->session->userdata('logged_in')['user_id']);
$data['teacher_details'] = $this->teacher_model->get_details_by_user_id(
    $data['user_details']['user_id']);
$data['my_courses'] = $this->course_model->get_all_courses_by_teacher(
    $data['teacher_details']['teacher_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
    user_details']['school_id']);
$data['gradebook_details'] = $this->course_model->
    get_gradebook_by_setup_id($setup_id);
$data['course_details'] = $this->course_model->get_details_by_id($data['
    gradebook_details']['course_id']);
}
}
}

```

```

$data['category_details'] = $this->grade_model->
    get_category_details_by_setup($setup_id,$category_id);
$result_array = $this->grade_model->get_category_items_by_id($setup_id,
    $category_id);
$data['gradebook_items'] = $result_array;
$data['gradebook_count'] = count($data['gradebook_items']);
$data['categories'] = $this->course_model->
    get_list_of_categories_by_teacher($data['teacher_details']['
        teacher_id']);
$array = array();
$index = 0;
foreach($result_array as $row){
    $array[$index] = $row['grade_item_id'];
    $index++;
}
if($this->session->userdata('move_to_category')){
    $items = $this->session->userdata('move_to_category');
    $grade_items = array();
    $category_items = array();
    foreach($items as $key => $value){
        $row_exp = explode('_', $value);
        if($row_exp[1] == "I"){
            $grade_items[] = $row_exp[0];
        }
        else
            $category_items[] = $row_exp[0];
    }
    if(count($category_items) == 0){
        $category_items = array(0);
    }
    $grade_items = array_diff($grade_items,$array);
    if(count($grade_items) == 0){
        $grade_items = array(0);
    }
    $data['new_items'] = $this->grade_model->move_category_items($grade_items
        ,$data['gradebook_details']['gradebook_setup_details_id']);
    $data['new_main_items'] = $this->grade_model->move_main_category_items(
        $category_items,$data['gradebook_details']['
            gradebook_setup_details_id']);
}
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/teacher_sidebar',$data);
$this->load->view('modify_gradebook_item',$data);
$this->load->view('templates/JSscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
    $this->session->set_tempdata('error_404','True');
    redirect('','refresh');
}
} else {
    redirect('','refresh');
}
}
public function modify_category_items(){
    if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
        email_verified'] == "YES"){
        $this->form_validation->set_rules('category_name[]','Category_Name', 'trim
            ');
        $this->form_validation->set_rules('gradebook_category_name','Gradebook_
            Category_Name', 'trim|required|callback_category_name_check');
        if ($this->form_validation->run() == FALSE){ // inputs didn't pass
            validation
            $this->session->set_flashdata('update_category_items_error',
                validation_errors());
            redirect('modify_gradebook_items/'.$this->input->post('gradebook_id').'/'.
                $this->input->post('category_id'),'refresh');
        }else{
            $gradebook_id = $this->input->post('gradebook_id');
            $method = $this->input->post('method_of_computation');
            $category_id = $this->input->post('category_id');
            $main_category_ids = $this->grade_model->get_category_items($category_id,
                $gradebook_id);
            $gradebook_count = count($main_category_ids); // existing
            //echo ' ===== <br/>';
            //print_r($this->input->post('category_ids'));
            //echo ' <br/> ===== <br/>';
            $test = $this->input->post('category_ids');
            $test_item = $this->input->post('category_ids');
            $test_count = count($test); // input
            /*echo 'Existing Item Ids: ';
            print_r($main_item_ids);
            echo '<br/>';
            echo 'Existing Category Ids: ';
            print_r($main_category_ids);
            echo '<br/>';
            echo 'Input Category Ids: ';
            print_r($test);
            echo '<br/>';
            */
            $item_check = 0;
            $new_inputs_array = array();
            $old_inputs_array = array();
            if($test_count > 0) {
                $new_inputs_array = array_keys($test,0);
                // echo 'New Input Array Keys: ';
                //print_r($new_inputs_array); // new categories / update for existing if
                    necessary
                // echo '<br/>';
                if($gradebook_count > 0) {
                    $old_inputs_array = array_keys(array_filter($test,function($item){
                        return $item != 0;
                    }));
                    //echo 'Old Input Array Keys: ';
                    //print_r($old_inputs_array); // existing categories for update
                    //echo '<br/>';
                }
            } else {
                $test = array(0);
            }
            if(count($old_inputs_array) < $gradebook_count){
                $removed_inputs = array_diff($main_category_ids,$test);
                //echo 'Removed Inputs: ';
                //print_r($removed_inputs);
                //echo '<br/>';
                $remove_ids = array_keys($removed_inputs);
            } else {
                $remove_ids = array(0);
            }
            //echo 'Removed Ids: ';
            //print_r($remove_ids);

```

```

//echo '<br/>';
if($test_count == 0){
//Case 1: No Input Delete All
$this->grade_model->delete_category_items($gradebook_id,$category_id);
} else if($gradebook_count == 0 && $test_count > 0){
//Case 2: No existing categories/ Insert All
$this->grade_model->insert_category_items($gradebook_id,$category_id,
    $test_count,$new_inputs_array);
} else if(count($old_inputs_array) == 0 && $test_count > 0) {
//Case 3: Delete All then add ALL New Categories
$this->grade_model->delete_category_items($gradebook_id,$category_id);
$this->grade_model->insert_category_items($gradebook_id,$category_id,
    $test_count,$new_inputs_array);
} else if(count($old_inputs_array) < $gradebook_count && count(
    $new_inputs_array) == 0){
//Case 4: Delete/Update Some with No New Categories
$this->grade_model->delete_category_items($gradebook_id,$category_id,count
    ($removed_inputs),$removed_inputs);
$this->grade_model->update_category_items($gradebook_id,$category_id,count
    ($old_inputs_array),$old_inputs_array);
} else if(count($old_inputs_array) == $gradebook_count && count(
    $new_inputs_array) == 0) {
//Case 5: Update all with NO Delete - No Insert
$this->grade_model->update_category_items($gradebook_id,$category_id,count
    ($old_inputs_array),$old_inputs_array);
} else if(count($old_inputs_array) == $gradebook_count && count(
    $new_inputs_array) > 0) {
//Case 6: Update All Existing and Add New Ones
$this->grade_model->update_category_items($gradebook_id,$category_id,count
    ($old_inputs_array),$old_inputs_array);
$this->grade_model->insert_category_items($gradebook_id,$category_id,count
    ($new_inputs_array),$new_inputs_array);
} else if(count($old_inputs_array) < $gradebook_count && count(
    $new_inputs_array) > 0){
//Case 7: Delete/Update Some with New Categories
$this->grade_model->delete_category_items($gradebook_id,$category_id,count
    ($removed_inputs),$removed_inputs);
$this->grade_model->update_category_items($gradebook_id,$category_id,count
    ($old_inputs_array),$old_inputs_array);
$this->grade_model->insert_category_items($gradebook_id,$category_id,count
    ($new_inputs_array),$new_inputs_array);
}
if($this->session->userdata('move_to_category')){
if(count($this->input->post('grade_item_ids')) > 0) {
$array = $this->input->post('grade_item_ids');
$categories = $this->input->post('grade_item_categories');
$categories = array_unique($categories);
$this->grade_model->remove_transferred_items($array);
$this->grade_model->update_categories_with_removed_items($categories,
    $gradebook_id);
$this->grade_model->update_category_weights_with_removed_items(
    $gradebook_id,$categories);
}
if(count($this->input->post('category_item_ids')) > 0) {
$array = $this->input->post('category_item_ids');
$this->grade_model->remove_transferred_category_items($array);
$this->grade_model->update_main_category_with_remove($gradebook_id);
}
$this->session->unset_userdata('move_to_category');
}

$array_ids = $this->grade_model->get_category_parameters($gradebook_id,
    $category_id);
$total_category_points = $this->grade_model->update_total_category_points(
    $category_id);
$this->grade_model->update_category_method($gradebook_id,$method,
    $category_id,$array_ids,$this->input->post('category_weight'),$this
->input->post('category_total'),$total_category_points);
$total_points = $this->grade_model->update_total_points($gradebook_id);
$array2_ids = $this->grade_model->get_gradebook_parameters($gradebook_id);
$array3_ids = $this->grade_model->get_gradebook_parameters_weight(
    $gradebook_id);
$array4_ids = $this->grade_model->get_gradebook_parameters_points(
    $gradebook_id);
$details = $this->course_model->get_gradebook_by_setup_id($gradebook_id);
$this->grade_model->update_method($gradebook_id,$details['
    method_of_computation'],$array2_ids,$array3_ids,$array4_ids,
    $total_points);
$new_category_ids = $this->grade_model->get_gradebook_main_categories(
    $this->input->post('gradebook_id'));
$this->grade_model->update_category_weights_with_removed_items(
    $gradebook_id,$new_category_ids);
$this->grade_model->update_final_grade($gradebook_id);
//update weight depending on method
$this->grade_model->update_mapping($gradebook_id);
$this->grade_model->update_class_record($gradebook_id);
$this->session->set_flashdata('modify_categories_succeed', 'Success!');
redirect('modify_gradebook_items/'.$this->input->post('gradebook_id').'/'.
    $this->input->post('category_id'),'refresh');
}
} else {
redirect('','refresh');
}
}

public function modify_category_items_template($gradebook_id, $cat_name,
    $method, $category_id, $category_ids, $name, $weight, $points){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
    email_verified'] == "YES"){
if ($this->form_validation->run() == FALSE){ // inputs didn't pass
    validation
$this->session->set_flashdata('update_category_items_error',
    validation_errors());
redirect('modify_gradebook_items/'.$this->input->post('gradebook_id').'/'.
    $this->input->post('category_id'),'refresh');
}
}
}
}

$main_category_ids = $this->grade_model->get_category_items($category_id,
    $gradebook_id);
$gradebook_count = count($main_category_ids); // existing
//echo ' ===== <br/>';
//print_r($this->input->post('category_ids'));
//echo ' <br/> ===== <br/>';
$test = $category_ids;
$test_item = $category_ids;
$test_count = count($test); // input
/*echo 'Existing Item Ids: ';
print_r($main_item_ids);
echo '<br/>';
echo 'Existing Category Ids: ';
print_r($main_category_ids);
echo '<br/>';
echo 'Input Category Ids: ';

```

```

print_r($test);
echo '<br/>';
*/
$item_check = 0;
$new_inputs_array = array();
$sold_inputs_array = array();
if($test_count > 0) {
    $new_inputs_array = array_keys($test);
    // echo 'New Input Array Keys: ';
    //print_r($new_inputs_array); // new categories / update for existing if
    // necessary
    // echo '<br/>';
    if($gradebook_count > 0) {
        $sold_inputs_array = array_keys(array_filter($test,function($item){
            return $item != 0;
        }));
        //echo 'Old Input Array Keys: ';
        //print_r($sold_inputs_array); // existing categories for update
        //echo '<br/>';
    }
    } else {
        $test = array(0);
    }
    if(count($sold_inputs_array) < $gradebook_count){
        $removed_inputs = array_diff($main_category_ids,$test);
        //echo 'Removed Inputs: ';
        //print_r($removed_inputs);
        //echo '<br/>';
        $remove_ids = array_keys($removed_inputs);
    } else {
        $remove_ids = array(0);
    }
    //echo 'Removed Ids: ';
    //print_r($remove_ids);
    //echo '<br/>';
    if($test_count == 0){
        //Case 1: No Input Delete All
        $this->grade_model->delete_category_items($gradebook_id,$category_id);
    } else if($gradebook_count == 0 && $test_count > 0){
        //Case 2: No existing categories/ Insert All
        // print_r($test_count);
        // print_r($new_inputs_array);
        $this->grade_model->insert_category_items_template($gradebook_id,
            $category_id,$test_count,$new_inputs_array,$name, $points);
    }
    if($this->session->userdata('move_to_category')){
        if(count($this->input->post('grade_item_ids')) > 0) {
            $array = $this->input->post('grade_item_ids');
            $categories = $this->input->post('grade_item_categories');
            $categories = array_unique($categories);
            $this->grade_model->remove_transferred_items($array);
            $this->grade_model->update_categories_with_removed_items($categories,
                $gradebook_id);
            $this->grade_model->update_category_weights_with_removed_items(
                $gradebook_id,$categories);
        }
        if(count($this->input->post('category_item_ids')) > 0) {
            $array = $this->input->post('category_item_ids');
            $this->grade_model->remove_transferred_category_items($array);
            $this->grade_model->update_main_category_with_remove($gradebook_id);
        }
    }
    $this->session->unset_userdata('move_to_category');
}
$array_ids = $this->grade_model->get_category_parameters($gradebook_id,
    $category_id);
$total_category_points = $this->grade_model->update_total_category_points(
    $category_id);
$this->grade_model->update_category_method_template($gradebook_id,
    $cat_name,$method,$category_id,$array_ids,$weight,$points,
    $total_category_points);
$total_points = $this->grade_model->update_total_points($gradebook_id);
$array2_ids = $this->grade_model->get_gradebook_parameters($gradebook_id);
$array3_ids = $this->grade_model->get_gradebook_parameters_weight(
    $gradebook_id);
$array4_ids = $this->grade_model->get_gradebook_parameters_points(
    $gradebook_id);
$details = $this->course_model->get_gradebook_by_setup_id($gradebook_id);
$this->grade_model->update_method($gradebook_id,$details['
    method_of_computation'],$array2_ids,$array3_ids,$array4_ids,
    $total_points);
$new_category_ids = $this->grade_model->get_gradebook_main_categories(
    $gradebook_id);
$this->grade_model->update_category_weights_with_removed_items(
    $gradebook_id,$new_category_ids);
$this->grade_model->update_final_grade($gradebook_id);
//update weight depending on method
$this->grade_model->update_mapping($gradebook_id);
$this->grade_model->update_class_record($gradebook_id);
//$this->session->set_flashdata('modify_categories_succeed', 'Success!');
//redirect('modify_gradebook_items/'.$this->input->post('gradebook_id')
    .'/'.$this->input->post('category_id'),'refresh');
}
} else {
    redirect('', 'refresh');
}
}
public function category_name_check($str)
{
    if ($this->grade_model->check_category_name($str,$this->input->post('
        gradebook_id'),$this->input->post('category_id'))
    {
        $this->form_validation->set_message('category_name_check', 'The{field}
            already exists. ');
        return FALSE;
    }
    else
    {
        return TRUE;
    }
}
public function move_gradebook_items(){
    if($this->session->userdata('logged_in')){
        if($SESSION['logged_in']['email_verified'] == "YES" && $SESSION['
            logged_in']['user_role'] == "TEACHER"){
            $result = $this->input->post('move_items');
            if($this->input->post('move_to_category') == 0) {
                $this->session->set_userdata('move_to_category',$result);
                $this->session->set_userdata('move_to_category_toast','1');
                redirect('modify_gradebook_categories/'.$this->input->post('gradebook_id')
                    );
            }
        }
    }
}

```

```

} else {
$this->session->set_userdata('move_to_category',$result);
$this->session->set_userdata('move_to_category_toast','1');
redirect('modify_gradebook_items/'.$this->input->post('gradebook_id').'/'.
    $this->input->post('move_to_category'));
}
} else {
$this->session->set_tempdata('error_404','True');
redirect('','refresh');
}
} else {
redirect('','refresh');
}
}

public function remove_grade_item(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
    email_verified'] === "YES"){
$this->form_validation->set_rules('grade_item_id','GradeItemID', '
    required|trim');
$this->form_validation->set_rules('category_id','CategoryID', 'required|
    trim');
if ($this->form_validation->run() === FALSE){ // inputs didn't pass
    validation
$this->session->set_flashdata('remove_grade_item_error', validation_errors
    ());
redirect('gradebook_setup/'.$this->input->post('course_id'),'refresh');
}else{
$gradebook_id = $this->input->post('setup_id');
$category_id = $this->input->post('category_id');
$this->grade_model->remove_grade_item($this->input->post('grade_item_id')
    );
$array = array($this->input->post('category_id'));
$this->grade_model->update_categories_with_removed_items($array,$this->
    input->post('setup_id'));
$this->grade_model->update_category_weights_with_removed_items(
    $gradebook_id,$array);
$array2_ids = $this->grade_model->get_gradebook_parameters($gradebook_id);
$array3_ids = $this->grade_model->get_gradebook_parameters_weight(
    $gradebook_id);
$array4_ids = $this->grade_model->get_gradebook_parameters_points(
    $gradebook_id);
$details = $this->course_model->get_gradebook_by_setup_id($gradebook_id);
$total_points = $this->grade_model->update_total_points($gradebook_id);
$this->grade_model->update_method($gradebook_id,$details['
    method_of_computation'],$array2_ids,$array3_ids,$array4_ids,
    $total_points);
$new_category_ids = $this->grade_model->get_gradebook_main_categories(
    $gradebook_id);
$this->grade_model->update_category_weights_with_removed_items(
    $gradebook_id,$new_category_ids);
$this->grade_model->update_final_grade($gradebook_id);
$this->grade_model->update_mapping($gradebook_id);
$this->grade_model->update_class_record($gradebook_id);
$this->session->set_flashdata('remove_grade_item_succeed', 'Success!');
redirect('gradebook_setup/'.$this->input->post('course_id'),'refresh');
}
} else {
redirect('','refresh');
}
}

public function remove_gradebook_item(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
    email_verified'] === "YES"){
$this->form_validation->set_rules('setup_id','SetupID', 'required|trim');
$this->form_validation->set_rules('category_id','CategoryID', 'required|
    trim');
if ($this->form_validation->run() === FALSE){ // inputs didn't pass
    validation
$this->session->set_flashdata('remove_gradebook_item_error',
    validation_errors());
redirect('gradebook_setup/'.$this->input->post('course_id'),'refresh');
}else{
$gradebook_id = $this->input->post('setup_id');
$category_id = $this->input->post('category_id');
$this->grade_model->remove_gradebook_item($this->input->post('category_id'
    ));
$this->grade_model->update_main_category_with_remove($this->input->post('
    setup_id'));
$array2_ids = $this->grade_model->get_gradebook_parameters($gradebook_id);
$array3_ids = $this->grade_model->get_gradebook_parameters_weight(
    $gradebook_id);
$array4_ids = $this->grade_model->get_gradebook_parameters_points(
    $gradebook_id);
$details = $this->course_model->get_gradebook_by_setup_id($gradebook_id);
$total_points = $this->grade_model->update_total_points($gradebook_id);
$this->grade_model->update_method($gradebook_id,$details['
    method_of_computation'],$array2_ids,$array3_ids,$array4_ids,
    $total_points);
$this->grade_model->update_final_grade($gradebook_id);
$this->grade_model->update_mapping($gradebook_id);
$this->grade_model->update_class_record($gradebook_id);
$this->session->set_flashdata('remove_gradebook_item_succeed', 'Success!')
    ;
redirect('gradebook_setup/'.$this->input->post('course_id'),'refresh');
}
} else {
redirect('','refresh');
}
}

public function remove_gradebook_category(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
    email_verified'] === "YES"){
$this->form_validation->set_rules('setup_id','SetupID', 'required|trim');
$this->form_validation->set_rules('category_id','CategoryID', 'required|
    trim');
if ($this->form_validation->run() === FALSE){ // inputs didn't pass
    validation
$this->session->set_flashdata('remove_gradebook_category_error',
    validation_errors());
redirect('gradebook_setup/'.$this->input->post('course_id'),'refresh');
}else{
$gradebook_id = $this->input->post('setup_id');
$category_id = $this->input->post('category_id');
$this->grade_model->remove_gradebook_category($this->input->post('
    category_id'));
$this->grade_model->update_main_category_with_remove($this->input->post('
    setup_id'));
$array2_ids = $this->grade_model->get_gradebook_parameters($gradebook_id);
$array3_ids = $this->grade_model->get_gradebook_parameters_weight(
    $gradebook_id);

```



```

$array4_ids = $this->grade_model->get_gradebook_parameters_points(
    $gradebook_id);
$details = $this->course_model->get_gradebook_by_setup_id($gradebook_id);
$total_points = $this->grade_model->update_total_points($gradebook_id);
$this->grade_model->update_method($gradebook_id,$details['
    method_of_computation'],$array2_ids,$array3_ids,$array4_ids,
    $total_points);
$this->grade_model->update_final_grade($gradebook_id);
$this->grade_model->update_mapping($gradebook_id);
$this->grade_model->update_class_record($gradebook_id);
$this->session->set_flashdata('remove_gradebook_category_succeed', '
    Success!');
redirect('gradebook_setup/'. $this->input->post('course_id'),'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function edit_grade_item(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
    email_verified'] === "YES"){
$this->form_validation->set_rules('grade_item_name','GradeItemName', '
    required|trim');
$this->form_validation->set_rules('total_points','TotalPoints', 'required
    |trim');
if ($this->form_validation->run() === FALSE){ // inputs didn't pass
    validation
$this->session->set_flashdata('update_grade_item_error', validation_errors
    ());
redirect('gradebook_setup/'. $this->input->post('course_id'),'refresh');
}else{
$gradebook_id = $this->input->post('setup_id');
$category_id = $this->input->post('category_id');
$this->grade_model->update_grade_item($this->input->post('grade_item_id')
    );
$array = array($this->input->post('category_id'));
$this->grade_model->update_categories_with_removed_items($array,$this->
    input->post('setup_id'));
$this->grade_model->update_category_weights_with_removed_items(
    $gradebook_id,$array);
$array2_ids = $this->grade_model->get_gradebook_parameters($gradebook_id);
$array3_ids = $this->grade_model->get_gradebook_parameters_weight(
    $gradebook_id);
$array4_ids = $this->grade_model->get_gradebook_parameters_points(
    $gradebook_id);
$details = $this->course_model->get_gradebook_by_setup_id($gradebook_id);
$total_points = $this->grade_model->update_total_points($gradebook_id);
$this->grade_model->update_method($gradebook_id,$details['
    method_of_computation'],$array2_ids,$array3_ids,$array4_ids,
    $total_points);
$new_category_ids = $this->grade_model->get_gradebook_main_categories(
    $gradebook_id);
$this->grade_model->update_category_weights_with_removed_items(
    $gradebook_id,$new_category_ids);
$this->grade_model->update_final_grade($gradebook_id);
$this->grade_model->update_mapping($gradebook_id);
$this->grade_model->update_class_record($gradebook_id);
$this->session->set_flashdata('update_gradebook_category_succeed', '
    Success!');
redirect('gradebook_setup/'. $this->input->post('course_id'),'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function save_gradebook_template(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
    email_verified'] === "YES"){
$options = 0;
if($this->session->flashdata('insert')){
$options = 1;
$this->form_validation->set_rules('gradebook_template_name','Gradebook_
    TemplateName', 'trim|callback_template_name_check');
}
if ($this->form_validation->run() === FALSE && $this->session->flashdata('
    insert') === '1'){
$this->session->set_flashdata('save_gradebook_template_error', 'Error!');
}
}
}

```

```

redirect('gradebook_setup/'. $this->input->post('course_id'),'refresh');
}else{
$gradebook_id = $this->input->post('setup_id');
if($option == 1){
$template_id = $this->grade_model->insert_gradebook_template($this->input
->post('gradebook_template_name'),$this->input->post('teacher_id'));
$main_category = $this->grade_model->
get_gradebook_main_category_for_template($gradebook_id);
$main_grade_item = $this->grade_model->get_grade_item_for_template(
$gradebook_id);
$this->grade_model->insert_template_elements($template_id,$main_category,
$main_grade_item);
} else {
$template_id = $this->input->post('existing_templates_name');
$this->grade_model->remove_template_elements($template_id);
$main_category = $this->grade_model->
get_gradebook_main_category_for_template($gradebook_id);
$main_grade_item = $this->grade_model->get_grade_item_for_template(
$gradebook_id);
$this->grade_model->insert_template_elements($template_id,$main_category,
$main_grade_item);
}
$this->session->set_flashdata('save_gradebook_template_succeed', 'Success!
');
redirect('gradebook_setup/'. $this->input->post('course_id'),'refresh');
}
} else {
redirect('', 'refresh');
}
}
public function template_name_check($str)
{
if ($this->grade_model->check_template_name($str,$_SESSION['logged_in']['
school_id'],$this->input->post('teacher_id'))
{
$this->form_validation->set_message('template_name_check', 'The_ field_
value_ already_ exists. ');
return FALSE;
}
else
{
return TRUE;
}
}
public function use_gradebook_template(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
email_verified'] === "YES"){
$this->form_validation->set_rules('existing_templates','Template_Name', '
required|trim');
if ($this->form_validation->run() === FALSE){
redirect('gradebook_setup/'. $this->input->post('course_id'),'refresh');
}else{
$gradebook_id = $this->input->post('setup_id');
$this->grade_model->remove_all_setup_elements($gradebook_id);
$template_id = $this->input->post('existing_templates');
$category_members = $this->grade_model->get_template_category_members(
$template_id);
$item_members = $this->grade_model->get_template_item_members($template_id
);
$sold_cat_ids = array();
$sold_item_ids = array();
if(count($category_members) > 0) {
foreach($category_members as $cat){
$sold_cat_ids[] = $cat['member_id'];
}
} else {
$sold_cat_ids = array(0);
}
if(count($item_members) > 0) {
foreach($item_members as $item){
$sold_item_ids[] = $item['member_id'];
}
} else {
$sold_item_ids = array(0);
}
$categories = $this->grade_model->get_main_categories_by_ids($sold_cat_ids)
;
$items = $this->grade_model->get_grade_items_by_ids($sold_item_ids);
$array2_ids = $this->grade_model->get_gradebook_parameters_template(
$template_id);
$array3_ids = $this->grade_model->get_gradebook_parameters_weight_template
($template_id);
$array4_ids = $this->grade_model->get_gradebook_parameters_points_template
($template_id);
$array5_ids = $this->grade_model->get_gradebook_parameters_types_template(
$template_id);
$array6_ids = $this->grade_model->get_gradebook_parameters_names_template(
$template_id);
$details = $this->course_model->get_gradebook_by_setup_id($gradebook_id);
$array = $this->modify_grade_category_template($gradebook_id,$details['
method_of_computation'],$array2_ids, $array5_ids, $array3_ids,
$array4_ids, $array6_ids);
$template_category_ids = $this->grade_model->get_template_category_ids(
$template_id);
$count = 0;
foreach($template_category_ids as $row){
$array2_ids = $this->grade_model->get_gradebook_items_parameters_template
($template_id,$row['template_category_id']);
$array3_ids = $this->grade_model->
get_gradebook_items_parameters_weight_template($template_id,$row['
template_category_id']);
$array4_ids = $this->grade_model->
get_gradebook_items_parameters_points_template($template_id,$row['
template_category_id']);
$array6_ids = $this->grade_model->
get_gradebook_items_parameters_names_template($template_id,$row['
template_category_id']);
$this->modify_category_items_template($gradebook_id,$row['
template_category_name'],$row['method_of_computation'], $array[
$count],$array2_ids, $array6_ids, $array3_ids, $array4_ids);
$count++;
}
$this->grade_model->update_final_grade($gradebook_id);
$this->grade_model->update_mapping($gradebook_id);
$this->grade_model->update_class_record($gradebook_id);
$this->session->set_flashdata('use_existing_template_succeed', 'Success!')
;
redirect('gradebook_setup/'. $this->input->post('course_id'),'refresh');
}
} else {
}
}

```

```

redirect('', 'refresh');
}
}
public function gradebook_templates_list(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['logged_in']['user_role'] === "TEACHER"){
$data['title'] = 'List of gradebook templates';
$data['user_details'] = $this->user_model->get_user_details_by_id($_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['user_details']['school_id']);
$data['teacher_details'] = $this->teacher_model->get_details_by_user_id($data['user_details']['user_id']);
$data['my_courses'] = $this->course_model->get_all_courses_by_teacher($data['teacher_details']['teacher_id']);
$data['templates'] = $this->grade_model->get_gradebook_templates($_SESSION['logged_in']['school_id'],$data['teacher_details']['teacher_id']);
$data['categories'] = $this->course_model->get_list_of_categories_by_teacher($data['teacher_details']['teacher_id']);
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/teacher_sidebar',$data);
$this->load->view('gradebook_template_list',$data);
$this->load->view('templates/JSscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_flashdata('error_404','True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}
public function remove_gradebook_template(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['email_verified'] === "YES"){
$this->form_validation->set_rules('template_id','Template Id', 'required|trim');
if ($this->form_validation->run() === FALSE){
redirect('templates_list','refresh');
}else{
$template_id = $this->input->post('template_id');
$this->grade_model->delete_gradebook_template($template_id);
$this->session->set_flashdata('remove_template_succeed', 'Success!');
redirect('templates_list','refresh');
}
} else {
redirect('', 'refresh');
}
}
public function change_template_name(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['email_verified'] === "YES"){
$this->form_validation->set_rules('template_name','Template Name', 'required|trim|callback_update_template_name_check');
if ($this->form_validation->run() === FALSE){
$this->session->set_flashdata('change_template_name_error', 'Error!');
redirect('templates_list','refresh');
}else{
$template_id = $this->input->post('template_id');
$this->grade_model->update_template_name($template_id,$this->input->post('template_name'));
$this->session->set_flashdata('change_template_name_succeed', 'Success!');
redirect('templates_list','refresh');
}
} else {
redirect('', 'refresh');
}
}
public function update_template_name_check($str)
{
if ($this->grade_model->update_check_template_name($str,$_SESSION['logged_in']['school_id'],$this->input->post('template_id'))
{
$this->form_validation->set_message('update_template_name_check', 'The field value already exists. ');
return FALSE;
}
else
{
return TRUE;
}
}
}
}
/*public function assign_gradebook_template($template_id){
}*/
public function update_grading_scale($course_id){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['logged_in']['user_role'] === "TEACHER"){
$data['title'] = 'Update grading scale';
$data['user_details'] = $this->user_model->get_user_details_by_id($_SESSION['logged_in']['user_id']);
$data['teacher_details'] = $this->teacher_model->get_details_by_user_id($data['user_details']['user_id']);
$data['my_courses'] = $this->course_model->get_all_courses_by_teacher($data['teacher_details']['teacher_id']);
$data['school_name'] = $this->user_model->get_school_name($data['user_details']['school_id']);
$data['gradebook_details'] = $this->course_model->get_gradebook_by_course($course_id);
$data['course_details'] = $this->course_model->get_details_by_id($data['gradebook_details']['course_id']);
$data['grading_scales'] = $this->grade_model->get_grading_scales($course_id);
$data['categories'] = $this->course_model->get_list_of_categories_by_teacher($data['teacher_details']['teacher_id']);
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/teacher_sidebar',$data);
$this->load->view('update_grading_scale',$data);
$this->load->view('templates/JSscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_flashdata('error_404','True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}
}

```

```

}

public function modify_grading_scale(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
    email_verified'] === "YES"){
$this->form_validation->set_rules('scale_values_labels[]','Category_Name',
    'trim');
if ($this->form_validation->run() === FALSE){ // inputs didn't pass
    validation
    redirect('update_scale/'.$this->input->post('course_id'),'refresh');
} else {
    $course_id = $this->input->post('course_id');
    $gradebook_scales = $this->grade_model->get_grading_scales($course_id);
    if(count($gradebook_scales) != 0){
        $gradebook_scale_id = $gradebook_scales[0]['grading_scale_id'];
    } else {
        $gradebook_scale_id = $this->grade_model->get_grading_scale_id($course_id)
        ;
    }
    $scale_ids = $this->grade_model->get_grading_scale_ids($course_id);
    $scale_count = count($gradebook_scales); // existing
    //echo '==== <br/>';
    //print_r($this->input->post('category_ids'));
    //echo ' <br/> ===== <br/>';
    $test = $this->input->post('scale_values_id');
    // input
    $test_count = count($test);
    /*echo 'Existing Item Ids: ';
    print_r($main_item_ids);
    echo '<br/>';
    echo 'Existing Category Ids: ';
    print_r($main_category_ids);
    echo '<br/>';
    echo 'Input Category Ids: ';
    print_r($test);
    echo '<br/>';*/
    $higher_limit = array();
    $lower_limit = $this->input->post('scale_values_lower_limit');
    $count = count($lower_limit);
    $higher_limit = $this->input->post('scale_values_higher_limit');
    $item_check = 0;
    $new_inputs_array = array();
    $old_inputs_array = array();
    if($test_count > 0) {
        $new_inputs_array = array_keys($test,0);
        // echo 'New Input Array Keys: ';
        //print_r($new_inputs_array); // new categories / update for existing if
        // necessary
        // echo '<br/>';
        if($gradebook_scales > 0) {
            $old_inputs_array = array_keys(array_filter($test,function($item){
                return $item != 0;
            }));
            //echo 'Old Input Array Keys: ';
            //print_r($old_inputs_array); // existing categories for update
            //echo '<br/>';
        }
    } else {
        $test_item = array(0);
        $item_check = 1;
        $test = array(0);
    }
}

}

if(count($old_inputs_array) < $scale_count){
    $removed_inputs = array_diff($scale_ids,$test);
    //echo 'Removed Inputs: ';
    //print_r($removed_inputs);
    //echo '<br/>';
    $remove_ids = array_keys($removed_inputs);
} else {
    $remove_ids = array(0);
}
//echo 'Removed Ids: ';
//print_r($remove_ids);
//echo '<br/>';
if($test_count == 0){
    //Case 1: No Input Delete All
    //echo 1;
    $this->grade_model->delete_gradebook_scales($gradebook_id);
} else if($scale_count == 0 && $test_count > 0){
    //Case 2: No existing categories/ Insert All
    //echo 2;
    $this->grade_model->insert_gradebook_scales($gradebook_scale_id,
        $test_count,$new_inputs_array,$higher_limit);
} else if(count($old_inputs_array) == 0 && $test_count > 0) {
    //Case 3: Delete All then add ALL New Categories
    //echo 3;
    $this->grade_model->delete_gradebook_scales($gradebook_scale_id);
    $this->grade_model->insert_gradebook_scales($gradebook_scale_id,
        $test_count,$new_inputs_array,$higher_limit);
} else if(count($old_inputs_array) < $scale_count && count(
    $new_inputs_array) == 0){
    //Case 4: Delete/Update Some with No New Categories
    //echo 4;
    $this->grade_model->delete_gradebook_scales($gradebook_scale_id,count(
        $removed_inputs),$removed_inputs);
    $this->grade_model->update_gradebook_scales($gradebook_scale_id,count(
        $old_inputs_array),$old_inputs_array,$higher_limit);
} else if(count($old_inputs_array) == $scale_count && count(
    $new_inputs_array) == 0) {
    //Case 5: Update all with NO Delete - No Insert
    //echo 5;
    $this->grade_model->update_gradebook_scales($gradebook_scale_id,count(
        $old_inputs_array),$old_inputs_array,$higher_limit);
} else if(count($old_inputs_array) == $scale_count && count(
    $new_inputs_array) > 0) {
    //Case 6: Update All Existing and Add New Ones
    //echo 6;
    $this->grade_model->update_gradebook_scales($gradebook_scale_id,count(
        $old_inputs_array),$old_inputs_array,$higher_limit);
    $this->grade_model->insert_gradebook_scales($gradebook_scale_id,count(
        $new_inputs_array),$new_inputs_array,$higher_limit);
} else if(count($old_inputs_array) < $scale_count && count(
    $new_inputs_array) > 0){
    //Case 7: Delete/Update Some with New Categories
    //echo 7;
    $this->grade_model->delete_gradebook_scales($gradebook_scale_id,count(
        $removed_inputs),$removed_inputs);
    $this->grade_model->update_gradebook_scales($gradebook_scale_id,count(
        $old_inputs_array),$old_inputs_array,$higher_limit);
    $this->grade_model->insert_gradebook_scales($gradebook_scale_id,count(
        $new_inputs_array),$new_inputs_array,$higher_limit);
}
}

```

```

}
//update weight depending on method
$this->grade_model->update_students_in_class_record($course_id);
$this->session->set_flashdata('modify_categories_succeed', 'Success!');
redirect('update_scale/'.$this->input->post('course_id'),'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function save_gradebook_scale(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['email_verified'] === "YES"){
$option = 0;
if($this->session->flashdata('insert_scale')){
$option = 1;
$this->form_validation->set_rules('gradebook_scale_name', 'Gradebook Scale Name', 'trim|callback_scale_name_check');
}
if ($this->form_validation->run() === FALSE && $this->session->flashdata('insert_scale') === '1'){
$this->session->set_flashdata('save_gradebook_scale_error', 'Error!');
redirect('gradebook_setup/'.$this->input->post('course_id'),'refresh');
}else{
$course_id = $this->input->post('course_id');
if($option == 1){
$template_id = $this->grade_model->insert_gradebook_scale($this->input->post('gradebook_scale_name'), $this->input->post('teacher_id'));
$scale_template_ids = $this->grade_model->get_scale_template_elements($this->input->post('course_id'));
$this->grade_model->insert_scale_template_elements($template_id, $scale_template_ids);
} else {
$template_id = $this->input->post('existing_scales_name');
$this->grade_model->remove_scale_template_elements($template_id);
$scale_template_ids = $this->grade_model->get_scale_template_elements($this->input->post('course_id'));
$this->grade_model->insert_scale_template_elements($template_id, $scale_template_ids);
}
$this->session->set_flashdata('save_gradebook_scale_succeed', 'Success!');
redirect('gradebook_setup/'.$this->input->post('course_id'),'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function scale_name_check($str)
{
if ($this->grade_model->check_scale_name($str, $_SESSION['logged_in']['school_id'], $this->input->post('teacher_id'))
{
$this->form_validation->set_message('scale_name_check', 'The {field} value already exists. ');
return FALSE;
}
else
{
return TRUE;
}
}
}

}

public function use_gradebook_scale(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['email_verified'] === "YES"){
$this->form_validation->set_rules('existing_templates', 'Template Name', 'required|trim');
if ($this->form_validation->run() === FALSE){
redirect('gradebook_setup/'.$this->input->post('course_id'),'refresh');
}else{
$gradebook_id = $this->input->post('setup_id');
$course_id = $this->input->post('course_id');
$scale_template_id = $this->input->post('existing_templates');
$scale_id = $this->grade_model->get_grading_scale_by_course_id($course_id);
;
$this->grade_model->delete_gradebook_scales($scale_id);
$scale_values_ids = $this->grade_model->get_scale_template_elements_by_scale_id($scale_template_id);
$scale_values = $this->grade_model->get_scale_values_by_ids($scale_values_ids);
$this->grade_model->insert_new_scale_template_elements($scale_id, $scale_values);
$this->grade_model->update_class_record($gradebook_id);
$this->session->set_flashdata('use_existing_scale_template_succeed', 'Success!');
redirect('gradebook_setup/'.$this->input->post('course_id'),'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function gradebook_scales_list(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['logged_in']['user_role'] === "TEACHER"){
$data['title'] = 'List of gradebook scales';
$data['user_details'] = $this->user_model->get_user_details_by_id($_SESSION['logged_in']['user_id']);
$data['school_name'] = $this->user_model->get_school_name($data['user_details']['school_id']);
$data['teacher_details'] = $this->teacher_model->get_details_by_user_id($data['user_details']['user_id']);
$data['my_courses'] = $this->course_model->get_all_courses_by_teacher($data['teacher_details']['teacher_id']);
$data['scales'] = $this->grade_model->get_gradebook_scales($_SESSION['logged_in']['school_id'], $data['teacher_details']['teacher_id']);
$data['categories'] = $this->course_model->get_list_of_categories_by_teacher($data['teacher_details']['teacher_id']);
$this->load->view('templates/admin_header', $data);
$this->load->view('templates/teacher_sidebar', $data);
$this->load->view('gradebook_scale_list', $data);
$this->load->view('templates/JSscripts', $data);
$this->load->view('templates/admin_footer', $data);
} else {
$this->session->set_flashdata('error_404', 'True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}
}

```

```

}
public function remove_gradebook_scale(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
email_verified'] === "YES"){
$this->form_validation->set_rules('template_id','Template_Id', 'required|
trim');
if ($this->form_validation->run() === FALSE){
redirect('scales_list','refresh');
}else{
$template_id = $this->input->post('template_id');
$this->grade_model->delete_gradebook_scale($template_id);
$this->session->set_flashdata('remove_scale_succeed', 'Success!');
redirect('scales_list','refresh');
}
} else {
redirect('', 'refresh');
}
}
public function change_scale_name(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
email_verified'] === "YES"){
$this->form_validation->set_rules('template_name','Template_Name', '
required|trim|callback_update_scale_name_check');
if ($this->form_validation->run() === FALSE){
$this->session->set_flashdata('change_template_name_error', 'Error');
redirect('scales_list','refresh');
}else{
$template_id = $this->input->post('template_id');
$this->grade_model->update_scale_name($template_id,$this->input->post('
template_name'));
$this->session->set_flashdata('change_scale_name_succeed', 'Success!');
redirect('scales_list','refresh');
}
} else {
redirect('', 'refresh');
}
}
public function update_scale_name_check($str)
{
if ($this->grade_model->update_check_scale_name($str,$_SESSION['logged_in'
]['school_id'],$this->input->post('template_id'))
{
$this->form_validation->set_message('update_scale_name_check', 'The_
value_already_exists. ');
return FALSE;
}
else
{
return TRUE;
}
}
public function assign_category_scale(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
email_verified'] === "YES"){
$this->form_validation->set_rules('existing_templates','Template_Name', '
required|trim');
if ($this->form_validation->run() === FALSE){
redirect('gradebook_setup_list','refresh');
}else{
$category_id = $this->input->post('category_id');
$scale_template_id = $this->input->post('existing_templates');
$category_courses_ids = $this->course_model->
get_assigned_courses_by_category($category_id);
foreach ($category_courses_ids as $course_id) {
$scale_id = $this->grade_model->get_grading_scale_by_course_id($course_id)
;
$this->grade_model->delete_gradebook_scales($scale_id);
$scale_values_ids = $this->grade_model->
get_scale_template_elements_by_scale_id($scale_template_id);
$scale_values = $this->grade_model->get_scale_values_by_ids(
$scale_values_ids);
$this->grade_model->insert_new_scale_template_elements($scale_id,
$scale_values);
}
$this->session->set_flashdata('assign_category_scale_succeed', 'Success!')
;
redirect('gradebook_setup_list','refresh');
}
} else {
redirect('', 'refresh');
}
}
public function view_course_record_info($course_id){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['
logged_in']['user_role'] === "TEACHER"){
$data['user_details'] = $this->user_model->get_user_details_by_id(
$_SESSION['logged_in']['user_id']);
$data['teacher'] = $this->teacher_model->get_details_by_user_id($data['
user_details']['user_id']);
$data['my_courses'] = $this->course_model->get_all_courses_by_teacher(
$data['teacher']['teacher_id']);
//$data['course_list'] = $this->course_model->get_course_by_teacher($data
['teacher']['teacher_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
user_details']['school_id']);
$data['course_details'] = $this->course_model->get_course_details_by_id(
$course_id);
$data['students'] = $this->student_model->get_student_details_by_course_id
($course_id);
$data['gradebook_details'] = $this->course_model->get_gradebook_by_course(
$course_id);
$data['categories'] = $this->course_model->
get_list_of_categories_by_teacher($data['teacher']['teacher_id']);
$data['title'] = 'Course_Information';
$this->load->view('templates/admin_header',$data);
$this->load->view('templates/teacher_sidebar',$data);
$this->load->view('course_record_info',$data);
$this->load->view('templates/JScscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_flashdata('error_404','True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}
public function class_record($course_id){
if($this->session->userdata('logged_in')){

```

```

if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['logged_in']['user_role'] === "TEACHER"){
    $data['user_details'] = $this->user_model->get_user_details_by_id($_SESSION['logged_in']['user_id']);
    $data['teacher'] = $this->teacher_model->get_details_by_user_id($data['user_details']['user_id']);
    $data['my_courses'] = $this->course_model->get_all_courses_by_teacher($data['teacher']['teacher_id']);
    $data['category_total'] = $this->grade_model->get_category_total_mapping($course_id);
    // $data['course_list'] = $this->course_model->get_course_by_teacher($data['teacher']['teacher_id']);
    $data['school_name'] = $this->user_model->get_school_name($data['user_details']['school_id']);
    $data['course_details'] = $this->course_model->get_course_details_by_id($course_id);
    $data['students'] = $this->student_model->get_student_details_by_course_id($course_id);
    $data['title'] = 'Student_Record_'.$data['course_details']['course_short_name'];
    $data['class_record'] = $this->student_model->get_class_record_details_by_course_id($course_id);
    $data['dropped_rows'] = $this->grade_model->get_dropped_students($course_id);
    $data['categories'] = $this->course_model->get_list_of_categories_by_teacher($data['teacher']['teacher_id']);
    // print_r($data['class_record']);
    $this->load->view('templates/admin_header', $data);
    $this->load->view('templates/teacher_sidebar', $data);
    $this->load->view('class_record', $data);
    $this->load->view('templates/JSscripts', $data);
    $this->load->view('templates/admin_footer', $data);
} else {
    $this->session->set_flashdata('error_404', 'True');
    redirect('', 'refresh');
}
} else {
    redirect('', 'refresh');
}
}

public function grades_stats($course_id){
    if($this->session->userdata('logged_in')){
        if($_SESSION['logged_in']['email_verified'] === "YES" && $_SESSION['logged_in']['user_role'] === "TEACHER"){
            $data['user_details'] = $this->user_model->get_user_details_by_id($_SESSION['logged_in']['user_id']);
            $data['teacher'] = $this->teacher_model->get_details_by_user_id($data['user_details']['user_id']);
            $data['my_courses'] = $this->course_model->get_all_courses_by_teacher($data['teacher']['teacher_id']);
            $data['category_total'] = $this->grade_model->get_category_total_mapping($course_id);
            // $data['course_list'] = $this->course_model->get_course_by_teacher($data['teacher']['teacher_id']);
            $data['school_name'] = $this->user_model->get_school_name($data['user_details']['school_id']);
            $data['course_details'] = $this->course_model->get_course_details_by_id($course_id);
            $data['students'] = $this->student_model->get_student_details_by_course_id($course_id);

```

```

        $data['title'] = 'Grades_Analysis_'.$data['course_details']['course_short_name'];
        $data['title2'] = 'Student_grades_data_'.$data['course_details']['course_short_name'];
        $data['class_record'] = $this->student_model->get_grades_stats_by_course_id($course_id);
        $data['dropped_rows'] = $this->grade_model->get_dropped_students($course_id);
        $data['categories'] = $this->course_model->get_list_of_categories_by_teacher($data['teacher']['teacher_id']);
        $data['category_names'] = $this->grade_model->get_category_stats_names($course_id);
        $data['enrolled_students'] = $this->grade_model->get_enrolled_student_list($course_id);
        $data['dropped_students'] = $this->grade_model->get_dropped_student_list($course_id);
        // $data['category_names'] = $this->grade_model->get_category_names($course_id);
        $data['course_id'] = $course_id;
        // print_r(count($data['enrolled_students']));
        // print_r(count($data['class_record']));
        $this->load->view('templates/admin_header', $data);
        $this->load->view('templates/teacher_sidebar', $data);
        $this->load->view('grades_stats', $data);
        $this->load->view('templates/JSscripts', $data);
        $this->load->view('templates/admin_footer', $data);
    } else {
        $this->session->set_flashdata('error_404', 'True');
        redirect('', 'refresh');
    }
} else {
    redirect('', 'refresh');
}
}
}

```

User Controller

```

[<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class User extends CI_Controller {

    public function login_user(){
        if($this->session->userdata('logged_in')){
            redirect('home', 'refresh');
        }else{
            $data['title'] = "Login";
            $this->form_validation->set_rules('user_name', 'UserName', 'required|trim|min_length[1]');
            $this->form_validation->set_rules('user_password', 'Password', 'required|trim|callback_check_database', array('callback_check_database' => 'Invalid_username_or_password'));
            $password = $this->input->post('user_password');
            $username = $this->input->post('user_name');
            if ($this->form_validation->run() === FALSE){ // inputs didn't pass validation
                $this->session->set_flashdata('login_error', 'Invalid_username_or_password');
            };
            redirect('', 'refresh');
        }
    }
}

```

```

$this->session->set_flashdata('login_succeed','Account_unlocked!');
redirect('home','refresh');
}
}
function check_database($password){
$username = $this->input->post('user_name');
$user_details = $this->user_model->get_password_salt($username);
$new_password = '';
$user_id = 0;
if(isset($user_details) && count($user_details) > 0){
$salt = $user_details['user_password_salt'];
$new_password = $salt.$password;
$new_password = hash('sha256',$new_password);
$user_id = $user_details['user_id'];
}
$result = $this->user_model->login_details($user_id,$username,
    $new_password);
if($result && $user_details)
{
    $sess_array = array();
    foreach($result as $row)
    {
        $sess_array = array(
            'user_id' => $row->user_id,
            'school_id' => $row->school_id,
            'user_name' => $row->user_name,
            'user_role' => $row->user_role,
            'email_verified' => $row->email_verified
        );
        $this->session->set_userdata('logged_in', $sess_array);
    }
    return TRUE;
}
else
{
    $this->form_validation->set_message('check_database', 'Invalid_username_or_password');
    return FALSE;
}
}
public function account_settings(){
if($this->session->userdata('logged_in')){
if($_SESSION['logged_in']['email_verified'] === "YES"){
$data['user_details'] = $this->user_model->get_user_details_by_id(
    $_SESSION['logged_in']['user_id']);
$user_details = $data['user_details'];
if($_SESSION['logged_in']['user_role'] === "TEACHER"){
$data['teacher'] = $this->teacher_model->get_details_by_user_id($data['
    user_details']['user_id']);
$data['my_courses'] = $this->course_model->get_all_courses_by_teacher(
    $data['teacher']['teacher_id']);
$data['categories'] = $this->course_model->
    get_list_of_categories_by_teacher($data['teacher']['teacher_id']);
} else {
$data['student'] = $this->student_model->get_details_by_user_id($data['
    user_details']['user_id']);
$data['my_courses'] = $this->course_model->get_all_courses_by_student(
    $data['student']['student_id']);
}
}
}
}

```

```

$data['title'] = 'Account_Settings_'.$user_details['user_first_name'].'_
    '.$user_details['user_last_name'];
// $data['course_list'] = $this->course_model->get_course_by_teacher($data
    ['teacher']['teacher_id']);
$data['school_name'] = $this->user_model->get_school_name($data['
    user_details']['school_id']);
$this->load->view('templates/admin_header',$data);
if($_SESSION['logged_in']['user_role'] === "TEACHER"){
$this->load->view('templates/teacher_sidebar',$data);
} else {
$this->load->view('templates/student_sidebar',$data);
}
$this->load->view('account_settings',$data);
$this->load->view('templates/JScscripts',$data);
$this->load->view('templates/admin_footer',$data);
} else {
$this->session->set_tempdata('error_404','True');
redirect('', 'refresh');
}
} else {
redirect('', 'refresh');
}
}
public function change_pass(){
if($this->session->userdata('logged_in')){
$this->form_validation->set_rules('old_password','Old_Password', 'required
    |trim|callback_check_pass');
$this->form_validation->set_rules('password','New_Password', 'required|
    trim|min_length[8]|max_length[20]');
$this->form_validation->set_rules('confirm_password','Confirm_New_Password
    ', 'required|trim|min_length[8]|max_length[20]');
if ($this->form_validation->run() === FALSE) { // inputs didn't pass
    validation
    $this->session->set_flashdata('change_password_error', 'Error!');
    redirect('account_settings','refresh');
} else {
    $salt = $this->user_model->get_password_salt_by_id($this->input->post('
        user_id'));
    $new_password = $salt.$this->input->post('confirm_password');
    $new_password = hash("sha256", $new_password);
    $this->user_model->change_password($new_password, $this->input->post('
        user_id'));
    $this->session->set_flashdata('change_password_succeed', 'Success!');
    redirect('account_settings','refresh');
}
} else {
    redirect('', 'refresh');
}
}
}
public function check_pass($old_password){
if($this->user_model->check_password($old_password, $_SESSION['logged_in'
    ]['user_id'])){
return TRUE;
} else {
return FALSE;
}
}
}
public function course_announcements($course_id){
if($this->session->userdata('logged_in')){

```



```

if($_SESSION['logged_in']['email_verified'] === "YES" && ($_SESSION['logged_in']['user_role'] === "TEACHER" || $_SESSION['logged_in']['user_role'] === "STUDENT")){
    $data['user_details'] = $this->user_model->get_user_details_by_id($_SESSION['logged_in']['user_id']);
    if($_SESSION['logged_in']['user_role'] === "TEACHER"){
        $data['teacher'] = $this->teacher_model->get_details_by_user_id($data['user_details']['user_id']);
        $data['my_courses'] = $this->course_model->get_all_courses_by_teacher($data['teacher']['teacher_id']);
        $data['categories'] = $this->course_model->get_list_of_categories_by_teacher($data['teacher']['teacher_id']);
    } else {
        $data['student'] = $this->student_model->get_details_by_user_id($data['user_details']['user_id']);
        $data['my_courses'] = $this->course_model->get_all_courses_by_student($data['student']['student_id']);
    }
    // $data['course_list'] = $this->course_model->get_course_by_teacher($data['teacher']['teacher_id']);
    $data['school_name'] = $this->user_model->get_school_name($data['user_details']['school_id']);
    $data['course_details'] = $this->course_model->get_course_details_by_id($course_id);
    $data['announcements'] = $this->course_model->get_announcements_by_course($course_id);
    $data['title'] = 'Course Announcements';
    $this->load->view('templates/admin_header',$data);
    if($_SESSION['logged_in']['user_role'] === "TEACHER"){
        $this->load->view('templates/teacher_sidebar',$data);
    } else {
        $this->load->view('templates/student_sidebar',$data);
    }
    $this->load->view('course_announcements',$data);
    $this->load->view('templates/JScscripts',$data);
    $this->load->view('templates/admin_footer',$data);
} else {
    $this->session->set_tempdata('error_404','True');
    redirect('', 'refresh');
}
} else {
    redirect('', 'refresh');
}
}

public function view_announcements($course_id, $announcement_id){
    if($this->session->userdata('logged_in')){
        if($_SESSION['logged_in']['email_verified'] === "YES" && ($_SESSION['logged_in']['user_role'] === "TEACHER" || $_SESSION['logged_in']['user_role'] === "STUDENT")){
            $data['user_details'] = $this->user_model->get_user_details_by_id($_SESSION['logged_in']['user_id']);
            if($_SESSION['logged_in']['user_role'] === "TEACHER"){
                $data['teacher'] = $this->teacher_model->get_details_by_user_id($data['user_details']['user_id']);
                $data['my_courses'] = $this->course_model->get_all_courses_by_teacher($data['teacher']['teacher_id']);
                $data['categories'] = $this->course_model->get_list_of_categories_by_teacher($data['teacher']['teacher_id']);
            } else {
                $data['student'] = $this->student_model->get_details_by_user_id($data['user_details']['user_id']);
                $data['my_courses'] = $this->course_model->get_all_courses_by_student($data['student']['student_id']);
            }
            // $data['course_list'] = $this->course_model->get_course_by_teacher($data['teacher']['teacher_id']);
            $data['school_name'] = $this->user_model->get_school_name($data['user_details']['school_id']);
            $data['course_details'] = $this->course_model->get_course_details_by_id($course_id);
            $data['announcements'] = $this->course_model->get_announcements_by_course($course_id);
            $data['title'] = 'View Announcement';
            $this->load->view('templates/admin_header',$data);
            if($_SESSION['logged_in']['user_role'] === "TEACHER"){
                $this->load->view('templates/teacher_sidebar',$data);
            } else {
                $this->load->view('templates/student_sidebar',$data);
            }
            $this->load->view('view_announcement',$data);
            $this->load->view('templates/JScscripts',$data);
            $this->load->view('templates/admin_footer',$data);
        } else {
            $this->session->set_tempdata('error_404','True');
            redirect('', 'refresh');
        }
    } else {
        redirect('', 'refresh');
    }
}

public function submit_announcements(){
    if($this->session->userdata('logged_in') && $_SESSION['logged_in']['email_verified'] === "YES"){
        $this->form_validation->set_rules('announcement_subject','Announcement Subject', 'required|trim|min_length[1]|max_length[256]');
        $this->form_validation->set_rules('announcement_content','Announcement Message', 'required|trim|min_length[1]|max_length[1024]');
        if ($this->form_validation->run() === FALSE){
            $this->session->set_flashdata('submit_announcement_error', 'Error!');
            redirect('course_announcements/'.$this->input->post('course_id'),'refresh');
        } else {
            $this->course_model->set_announcement($this->input->post('course_id'));
            $data['course_details'] = $this->course_model->get_details_by_id($this->input->post('course_id'));
            $email_addresses = $this->student_model->get_email_addresses_by_course($this->input->post('course_id'));
            $this->user_model->send_announcement_mails($email_addresses, $data['course_details']);
            $this->session->set_flashdata('submit_announcement_succeed', 'Success!');
            redirect('course_announcements/'.$this->input->post('course_id'),'refresh');
        }
    } else {
        redirect('', 'refresh');
    }
}

public function remove_announcements(){

```

```

if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
    email_verified'] === "YES"){
$this->form_validation->set_rules('course_id','Course_ID', 'required|trim'
);
$this->form_validation->set_rules('announcement_id','Announcement_ID', '
    required|trim');
if ($this->form_validation->run() === FALSE){
redirect('course_announcements/'.$course_id,'refresh');
}else{
$course_id = $this->input->post('course_id');
$ann_id = $this->input->post('announcement_id');
$this->course_model->delete_announcement($course_id, $ann_id);
$this->session->set_flashdata('remove_announcement_succeed', 'Success!');
redirect('course_announcements/'.$course_id,'refresh');
}
} else {
redirect('', 'refresh');
}
}

public function update_announcements(){
if($this->session->userdata('logged_in') && $_SESSION['logged_in']['
    email_verified'] === "YES"){
$this->form_validation->set_rules('announcement_subject','Announcement_
    Subject', 'required|trim|min_length[1]|max_length[256]');
$this->form_validation->set_rules('announcement_content','Announcement_
    Message', 'required|trim|min_length[1]|max_length[1024]');
if ($this->form_validation->run() === FALSE){
$this->session->set_flashdata('update_announcement_error', 'Error!');
redirect('view_announcements/'.$this->input->post('course_id').'/'.$this->
    input->post('announcement_id'),'refresh');
}else{
$this->course_model->update_announcement($this->input->post('course_id'),
    $this->input->post('announcement_id'));
$this->session->set_flashdata('update_announcement_succeed', 'Success!');
redirect('view_announcements/'.$this->input->post('course_id').'/'.$this->
    input->post('announcement_id'),'refresh');
}
} else {
redirect('', 'refresh');
}
}
}
}
}
}
}

```

Course Model

```

<?php
if (!defined('BASEPATH')) exit('No direct script access allowed');

class Course_model extends CI_Model {

public function __construct(){
$this->load->database();
$this->load->helper('url_helper');
$this->load->library('session');
}

public function add_course($course_type, $course_name = NULL,
    $course_name_id = NULL){
$label_array = array('A','B','C','D','E');
$lower_limit_array = array('91','81','71','61','0');
$higher_limit_array = array('100','90','80','70','60');
$school_id = $_SESSION['logged_in']['school_id'];

```

```

$start_date = $this->input->post('course_start');
$end_date = $this->input->post('course_end');
$edited_start_date = date("Y-m-d", strtotime($start_date));
$edited_end_date = date("Y-m-d", strtotime($end_date));
if($course_type === "new"){
$data = array(
'school_id' => $school_id,
'course_full_name' => $this->input->post('course_full_name'),
'course_short_name' => $course_name
);
$this->db->insert('course_name',$data);
$course_name_id = $this->db->insert_id();
$this->db->select('*');
$this->db->from('course_name');
$this->db->where('course_name_id',$course_name_id);
$query = $this->db->get();
$course_full_name = $query->row_array()['course_full_name'];
$course_short_name = $query->row_array()['course_short_name'];
} else {
$this->db->select('*');
$this->db->from('course_name');
$this->db->where('course_name_id',$course_name_id);
$query = $this->db->get();
$course_name_id = $query->row_array()['course_name_id'];
$course_full_name = $query->row_array()['course_full_name'];
$course_short_name = $query->row_array()['course_short_name'];
}
$data = array(
'school_id' => $school_id,
'course_name_id' => $course_name_id,
'course_full_name' => $course_full_name,
'course_short_name' => $course_short_name,
'teacher_id' => $this->input->post('assigned_teacher'),
'is_archived' => "NO",
'section_name' => $this->input->post('section_name'),
'course_start_date' => $edited_start_date,
'course_end_date' => $edited_end_date
);
$this->db->insert('course',$data);
$course_id = $this->db->insert_id();
$data2 = array(
'course_category_id' => $this->input->post('course_category'),
'course_id' => $course_id
);
$this->db->insert('category_members',$data2);
$data3 = array(
'course_id' => $course_id,
'gradebook_setup_name' => $course_short_name.'_'.$this->input->post('
    section_name').'Gradebook',
'adjust_grade_settings' => "NO",
'show_percentage_to_total' => "YES",
'method_of_computation' => 'Mean_of_Grades',
'allow_teacher' => "NO"
);
$this->db->insert('gradebook_setup_details', $data3);
$data4 = array(
'school_id' => $_SESSION['logged_in']['school_id'],
'course_id' => $course_id
);
$this->db->insert('grading_scale',$data4);

```

```

$this->db->select('*');
$this->db->from('grading_scale');
$this->db->where('course_id',$course_id);
$query2 = $this->db->get();
$grading_scale_id = $query2->row_array()['grading_scale_id'];
$index_count = 0;
foreach($label_array as $label){
    $data5 = array(
        'grading_scale_id' => $grading_scale_id,
        'scale_values_label' => $label,
        'scale_values_lower_limit' => $lower_limit_array[$index_count],
        'scale_values_higher_limit' => $higher_limit_array[$index_count]
    );
    $this->db->insert('scale_values',$data5);
    $index_count++;
}
}

public function add_multiple_courses(){
    $school_id = $_SESSION['logged_in']['school_id'];
    $course_short_name = $this->input->post('course_short_name');
    $course_full_name = $this->input->post('course_full_name');
    $course_category = $this->input->post('course_category_ids');
    $label_array = array('A','B','C','D','E');
    $lower_limit_array = array('91','81','71','61','0');
    $higher_limit_array = array('100','90','80','70','60');
    $size = count($course_short_name);
    for($index = 0; $index < $size ; $index++){
        $data = array(
            'school_id' => $school_id,
            'course_full_name' => $course_full_name[$index],
            'course_short_name' => $course_short_name[$index],
            'teacher_id' => 0,
            'is_archived' => "NO"
        );
        $this->db->insert('course',$data);
        $this->db->select('*');
        $this->db->from('course');
        $this->db->where('course_short_name',$course_short_name[$index]);
        $query = $this->db->get();
        $course_id = $query->row_array()['course_id'];
        $data2 = array(
            'course_category_id' => $course_category[$index],
            'course_id' => $course_id
        );
        $this->db->insert('category_members',$data2);
        $data3 = array(
            'course_id' => $course_id,
            'gradebook_setup_name' => $course_short_name[$index].'_Gradebook',
            'adjust_grade_settings' => "NO",
            'show_percentage_to_total' => "YES",
            'method_of_computation' => 'Mean_of_Grades',
            'allow_teacher' => "NO"
        );
        $this->db->insert('gradebook_setup_details', $data3);
        $data4 = array(
            'school_id' => $_SESSION['logged_in']['school_id'],
            'course_id' => $course_id
        );
        $this->db->insert('grading_scale',$data4);
        $this->db->select('*');

        $this->db->from('grading_scale');
        $this->db->where('course_id',$course_id);
        $query2 = $this->db->get();
        $grading_scale_id = $query2->row_array()['grading_scale_id'];
        $index_count = 0;
        foreach($label_array as $label){
            $data5 = array(
                'grading_scale_id' => $grading_scale_id,
                'scale_values_label' => $label,
                'scale_values_lower_limit' => $lower_limit_array[$index_count],
                'scale_values_higher_limit' => $higher_limit_array[$index_count]
            );
            $this->db->insert('scale_values',$data5);
            $index_count++;
        }
    }

    public function add_course_category(){
        $data = array(
            'school_id' => $_SESSION['logged_in']['school_id'],
            'course_category_name' => $this->input->post('course_category_name'),
            'teacher_id' => $this->input->post('assigned_head')
        );
        $this->db->insert('course_category',$data);
        $this->db->select('*');
        $this->db->from('course_category');
        $this->db->where('course_category_name',$this->input->post('course_category_name'));
        $query = $this->db->get();
        $course_category_id = $query->row_array()['course_category_id'];
        $array = $this->input->post('category_courses');
        $size = count($array);
        if($size > 0){
            foreach($array as $id){
                $data2 = array(
                    'course_category_id' => $course_category_id
                );
                $this->db->where('course_id',$id);
                $this->db->update('category_members',$data2);
            }
        }

        public function get_list_of_courses(){
            $this->db->select('*');
            $this->db->from('course_as_c');
            $this->db->where('c.school_id', $_SESSION['logged_in']['school_id']);
            $this->db->join('category_members_as_cm','c.course_id=cm.course_id','left');
            $this->db->join('course_category_as_cc','cm.course_category_id=cc.course_category_id','left');
            $this->db->join('teacher_as_t','c.teacher_id=t.teacher_id','left');
            $this->db->join('user_as_u','t.user_id=u.user_id','left');
            $this->db->join('user_details_as_ud','u.user_id=ud.user_id','left');
            $this->db->order_by('cm.course_category_id');
            $query = $this->db->get();
            return $query->result_array();
        }

        public function get_list_of_uncategorized_courses(){
            $this->db->select('*');
            $this->db->from('course_as_c');

```

```

$this->db->where('c.school_id', $_SESSION['logged_in']['school_id']);
$this->db->join('category_members_uas_cm','c.course_id=u.cm.course_id','
    left');
$this->db->where('cm.course_category_id', 0);
$this->db->join('course_category_uas_cc','cm.course_category_id=u.cc.
    course_category_id','left');
$query = $this->db->get();
return $query->result_array();
}

public function get_list_of_unassigned_courses(){
$this->db->select('*');
$this->db->from('course_uas_c');
$this->db->where('c.school_id', $_SESSION['logged_in']['school_id']);
$this->db->where('c.teacher_id', 0);
$this->db->join('category_members_uas_cm','c.course_id=u.cm.course_id','
    left');
$this->db->join('course_category_uas_cc','cm.course_category_id=u.cc.
    course_category_id','left');
$query = $this->db->get();
return $query->result_array();
}

function get_last_student_index(){
$this->db->select_max('student_id','latest_id');
$this->db->from('student');
$query = $this->db->get();
return $query->row_array();
}

public function get_list_of_categories(){
$this->db->select('*');
$this->db->from('course_category');
$this->db->where('school_id',$_SESSION['logged_in']['school_id']);
$query = $this->db->get();
return $query->result_array();
}

public function get_list_of_categories_by_teacher($teacher_id){
$this->db->select('*');
$this->db->from('course_category');
$this->db->where('school_id',$_SESSION['logged_in']['school_id']);
$this->db->order_by('course_category_id','ASC');
$this->db->where('teacher_id',$teacher_id);
$query = $this->db->get();
return $query->result_array();
}

public function get_category_details($category_id){
$this->db->select('*');
$this->db->from('course_category_uas_cc');
$this->db->where('cc.course_category_id',$category_id);
$this->db->join('teacher_uas_t','cc.teacher_id=u.t.teacher_id','left');
$this->db->join('user_uas_u','t.user_id=u.u.user_id','left');
$this->db->join('user_details_uas_ud','u.user_id=u.ud.user_id','left');
$query = $this->db->get();
return $query->row_array();
}

public function get_list_of_courses_by_category($category_id){
$this->db->select('*');
$this->db->from('course_uas_c');
$this->db->where('c.school_id', $_SESSION['logged_in']['school_id']);
$this->db->join('category_members_uas_cm','c.course_id=u.cm.course_id','
    left');
$this->db->where('cm.course_category_id',$category_id);
$this->db->join('course_category_uas_cc','cm.course_category_id=u.cc.
    course_category_id','left');
$this->db->join('teacher_uas_t','c.teacher_id=u.t.teacher_id','left');
$this->db->join('user_uas_u','t.user_id=u.u.user_id','left');
$this->db->join('user_details_uas_ud','u.user_id=u.ud.user_id','left');
$query = $this->db->get();
return $query->result_array();
}

public function get_all_courses_by_teacher($teacher_id){
$sql = "SELECT course_short_name, course_id, section_name FROM course_uas_c
    WHERE teacher_id=".$this->db->escape($teacher_id).";";
$query = $this->db->query($sql);
return $query->result_array();
}

public function get_all_courses_by_student($student_id){
$sql = "SELECT course_id FROM course_students WHERE student_id=".$this->db->
    escape($student_id)."and is_dropped='NO'";
$query = $this->db->query($sql);
$result = $query->result_array();
$id_array = array_column($result, 'course_id');
if(count($id_array) > 0){
$this->db->select('*');
$this->db->from('course_uas_c');
$this->db->where_in('c.course_id',$id_array);
$query = $this->db->get();
return $query->result_array();
}
}

public function remove_course_category($cc_id){
$sql = "SELECT course_id FROM category_members WHERE course_category_id="
    .$this->db->escape($cc_id);
$query = $this->db->query($sql);
$result = $query->result_array();
$id_array = array_column($result, 'course_id');
$this->db->trans_start();
$this->db->where('course_category_id', $cc_id);
$this->db->delete('course_category');
$this->db->trans_complete();
if(count($id_array) > 0){
foreach($id_array as $key => $value){
$data = array('course_category_id' => 0);
$this->db->where('course_id',$value);
$this->db->update('category_members',$data);
}
}
}

```

```

}
}
}
public function get_course_details_by_id($course_id){
    $sql = "SELECT_*_FROM_*WHERE_*course_id='". $this->db->escape(
        $course_id);
    $query = $this->db->query($sql);
    return $query->row_array();
}
public function get_details_by_id($course_id){
    $this->db->select('*');
    $this->db->from('course_*');
    $this->db->where('c.course_id',$course_id);
    $this->db->join('category_members_*as_*cm','c.course_id=*cm.course_id','
        left');
    $query = $this->db->get();
    return $query->row_array();
}
public function check_course_category_name($course_category_name){
    $this->db->select('*');
    $this->db->from('course_category');
    $this->db->where('course_category_name',$course_category_name);
    $query = $this->db->get();
    if($query -> num_rows() == 1)
    {
        return true;
    }
    else
    {
        return false;
    }
}
public function check_section_name($section_name, $course_type){
    if($course_type == "existing"){
        $course_name_id = $this->input->post('existing_course');
        $this->db->select('*');
        $this->db->from('course');
        $this->db->where('section_name',$section_name);
        $this->db->where('course_name_id',$course_name_id);
        $query = $this->db->get();
        if($query -> num_rows() == 1)
        {
            return true;
        }
        else
        {
            return false;
        }
    } else {
        return false;
    }
}
public function check_update_course_category_name($course_category_name,
    $course_category_id){
    $this->db->select('*');
    $this->db->from('course_category');
    $this->db->where('course_category_name',$course_category_name);
    $this->db->where('course_category_id!='.$course_category_id);
    $query = $this->db->get();
    if($query -> num_rows() == 1)
    {
        return true;
    }
    else
    {
        return false;
    }
}
public function check_course_fullname($course_fullname){
    $this->db->select('*');
    $this->db->from('course_name');
    $this->db->where('course_full_name',$course_fullname);
    $query = $this->db->get();
    if($query -> num_rows() == 1)
    {
        return true;
    }
    else
    {
        return false;
    }
}
public function check_course_shortcode($course_shortcode){
    $this->db->select('*');
    $this->db->from('course_shortcode');
    $this->db->where('course_shortcode',$course_shortcode);
    $query = $this->db->get();
    if($query -> num_rows() == 1)
    {
        return true;
    }
    else
    {
        return false;
    }
}
public function update_course($course_id){
    $start_date = $this->input->post('course_start');
    $end_date = $this->input->post('course_end');
    $edited_start_date = date("Y-m-d", strtotime($start_date));
    $edited_end_date = date("Y-m-d", strtotime($end_date));
    $data = array(
        'course_full_name' => $this->input->post('course_full_name'),
        'course_shortcode' => $this->input->post('course_shortcode'),
        'teacher_id' => $this->input->post('assigned_teacher'),
        'section_name' => $this->input->post('section_name'),
        'course_start_date' => $edited_start_date,
        'course_end_date' => $edited_end_date
    );
    $this->db->where('course_id',$course_id);
    $this->db->update('course',$data);
    $data2 = array(
        'course_category_id' => $this->input->post('course_category')
    );
    $this->db->where('course_id',$course_id);
    $this->db->update('category_members',$data2);
}
public function get_assigned_courses_by_category($category_id){
    $this->db->select('*');
    $this->db->from('category_members');
}

```

```

$this->db->where('course_category_id',$category_id);
$query = $this->db->get();
$result = $query->result_array();
$array = array();
$index = 0;
foreach($result as $row){
    $array[$index] = $row['course_id'];
    $index++;
}
return $array;
}

public function update_course_category($course_category_id,$array_courses)
{
    $data = array(
        'course_category_name' => $this->input->post('course_category_name'),
        'teacher_id' => $this->input->post('assigned_head')
    );
    $this->db->where('course_category_id',$course_category_id);
    $this->db->update('course_category',$data);
    $array = $this->input->post('category_courses');
    $new_array_size = count($array);
    $old_array_size = count($array_courses);
    if($new_array_size > $old_array_size){
        $result_array = array_diff($array,$array_courses);
        foreach($result_array as $id){
            $data2 = array(
                'course_category_id' => $course_category_id
            );
            $this->db->where('course_id',$id);
            $this->db->update('category_members',$data2);
        }
    } else if($new_array_size < $old_array_size) {
        $result_array = array_diff($array_courses,$array);
        foreach($result_array as $id){
            $data2 = array(
                'course_category_id' => 0
            );
            $this->db->where('course_id',$id);
            $this->db->update('category_members',$data2);
        }
    }
}

public function assign_students_to_course($student_ids,$course_id){
    foreach($student_ids as $id){
        $data = array(
            'student_id' => $id,
            'course_id' => $course_id,
            'is_dropped' => "NO",
            'order_number' => 0
        );
        $this->db->insert('course_students',$data);
        $data2 = array(
            'student_id' => $id,
            'course_id' => $course_id,
            'cell_value' => '',
            'calculated_value' => 0
        );
        $this->db->insert('final_grade',$data2);
    }
    $this->update_order_in_course($course_id);
}

public function update_order_in_course($course_id){
    $sql = "SELECT student_id FROM course_students WHERE course_id=". $this->db->escape($course_id);
    $query = $this->db->query($sql);
    $student_ids = array_column($query->result_array(),'student_id');
    $this->db->select('s.student_id,ud.user_last_name,ud.user_first_name');
    $this->db->from('student as s');
    $this->db->where_in('s.student_id',$student_ids);
    $this->db->join('user as u','s.user_id=u.user_id','inner');
    $this->db->join('user_details as ud','u.user_id=ud.user_id','inner');
    $this->db->order_by('ud.user_last_name','ud.user_first_name');
    $query = $this->db->get();
    $result = $query->result_array();
    $count = 0;
    foreach($result as &$row){
        $count++;
        $row['order_number'] = $count;
    }
    foreach($result as &$row){
        $data = array('order_number' => $row['order_number']);
        $this->db->where('student_id', $row['student_id']);
        $this->db->update('course_students',$data);
    }
}

public function remove_student_from_course($student_id,$course_id){
    $this->db->where('student_id',$student_id);
    $this->db->where('course_id',$course_id);
    $this->db->delete('course_students');
    $this->db->where('student_id',$student_id);
    $this->db->where('course_id',$course_id);
    $this->db->delete('final_grade');
    $sql = "SELECT gradebook_setup_details_id FROM gradebook_setup_details WHERE course_id=". $this->db->escape($course_id);
    $query = $this->db->query($sql);
    $res = $query->row_array();
    $gradebook_id = $res['gradebook_setup_details_id'];
    $this->db->where('student_id',$student_id);
    $this->db->where('gradebook_setup_details_id',$gradebook_id);
    $this->db->delete('class_record');
}

public function remove_course_from_category($course_id,$category_id){
    $data = array('course_category_id' => 0);
    $this->db->where('course_id',$course_id);
    $this->db->update('category_members',$data);
}

public function get_gradebook_by_course($course_id){
    $this->db->select('*');
    $this->db->from('gradebook_setup_details');
    $this->db->where('course_id',$course_id);
    $query = $this->db->get();
    return $query->row_array();
}

public function get_gradebook_by_setup_id($setup_id){
    $this->db->select('*');
    $this->db->from('gradebook_setup_details');
    $this->db->where('gradebook_setup_details_id',$setup_id);
    $query = $this->db->get();
    return $query->row_array();
}
}

```

```

public function get_gradebook_categories($setup_id){
$this->db->select('*');
$this->db->from('main_category');
$this->db->where('gradebook_setup_details_id',$setup_id);
$query = $this->db->get();
return $query->result_array();
}

public function get_existing_courses($school_id){
$this->db->select('*');
$this->db->from('course_name');
$this->db->where('school_id',$school_id);
$query = $this->db->get();
return $query->result_array();
}

public function get_announcements_by_course($course_id){
$sql = 'SELECT * FROM announcement WHERE course_id=' . $this->db->escape(
    $course_id) . ' ORDER BY date_modified DESC';
$query = $this->db->query($sql);
return $query->result_array();
}

public function get_announcement_by_id($announcement_id){
$sql = 'SELECT * FROM announcement WHERE announcement_id=' . $this->db->
    escape($announcement_id);
$query = $this->db->query($sql);
return $query->row_array();
}

public function set_announcement($course_id){
$data = array('course_id' => $course_id,
'announcement_title' => $this->input->post('announcement_subject'),
'announcement_content' => $this->input->post('announcement_content')
);
$this->db->insert('announcement', $data);
}

public function delete_announcement($course_id, $ann_id){
$this->db->where('announcement_id', $ann_id);
$this->db->where('course_id', $course_id);
$this->db->delete('announcement');
}

public function update_announcement($course_id, $ann_id){
$data = array('announcement_title' => $this->input->post('
    announcement_subject'),
'announcement_content' => $this->input->post('announcement_content')
);
$this->db->where('announcement_id', $ann_id);
$this->db->where('course_id', $course_id);
$this->db->update('announcement', $data);
}

public function remove_course($course_id){
$sql = 'SELECT gradebook_setup_details_id FROM gradebook_setup_details
    WHERE course_id=' . $this->db->escape($course_id);
$query = $this->db->query($sql);
$result = $query->row_array();
$gradebook_id = $result['gradebook_setup_details_id'];
$this->db->trans_start();
$this->db->where('course_id', $course_id);
$this->db->delete('announcement');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('course_id', $course_id);
$this->db->delete('category_members');
}

$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('gradebook_setup_details_id', $gradebook_id);
$this->db->delete('class_record');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('course_id', $course_id);
$this->db->delete('course');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('course_id', $course_id);
$this->db->delete('course_students');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('course_id', $course_id);
$this->db->delete('final_grade');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('course_id', $course_id);
$this->db->delete('grading_scale');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('gradebook_setup_details_id', $gradebook_id);
$this->db->delete('gradebook_setup_details');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('gradebook_setup_details_id', $gradebook_id);
$this->db->delete('grade_item');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('gradebook_setup_details_id', $gradebook_id);
$this->db->delete('gradebook_setup_details');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('gradebook_setup_details_id', $gradebook_id);
$this->db->delete('main_category');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('gradebook_setup_details_id', $gradebook_id);
$this->db->delete('mapping');
$this->db->trans_complete();
}

public function clear_course_record($course_id){
$sql = 'SELECT gradebook_setup_details_id FROM gradebook_setup_details
    WHERE course_id=' . $this->db->escape($course_id);
$query = $this->db->query($sql);
$result = $query->row_array();
$gradebook_id = $result['gradebook_setup_details_id'];
$data = array('total_points' => 0, 'max_grade' => 0);
$this->db->trans_start();
$this->db->where('gradebook_setup_details_id', $gradebook_id);
$this->db->update('gradebook_setup_details', $data);
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('course_id', $course_id);
$this->db->delete('announcement');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('gradebook_setup_details_id', $gradebook_id);
$this->db->delete('class_record');
}

```

```

$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('course_id', $course_id);
$this->db->delete('course_students');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('course_id', $course_id);
$this->db->delete('final_grade');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('gradebook_setup_details_id', $gradebook_id);
$this->db->delete('grade_item');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('gradebook_setup_details_id', $gradebook_id);
$this->db->delete('main_category');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('gradebook_setup_details_id', $gradebook_id);
$this->db->delete('mapping');
$this->db->trans_complete();
}
}

```

Grade Model

```

<?php
if (!defined('BASEPATH')) exit('No direct script access allowed');
class Grade_model extends CI_Model {
public function __construct(){
$this->load->database();
$this->load->helper('url_helper');
$this->load->library('session');
}
public function update_gradebook_settings($course_id){
$data = array(
'allow_teacher' => $this->input->post('allow_teacher')
);
$this->db->where('course_id', $course_id);
$this->db->update('gradebook_setup_details', $data);
}
public function get_gradebook_main_category_details($setup_id){
$this->db->select('*');
$this->db->from('main_category');
$this->db->order_by('main_category_type', 'ASC');
$this->db->order_by('main_category_id', 'ASC');
$this->db->where('gradebook_setup_details_id', $setup_id);
$query = $this->db->get();
return $query->result_array();
}
public function get_gradebook_main_categories($setup_id){
$this->db->select('*');
$this->db->from('main_category');
$this->db->where('main_category_type', 'CATEGORY');
$this->db->where('gradebook_setup_details_id', $setup_id);
$query = $this->db->get();
$result = $query->result_array();
$array = array();
$index = 0;
foreach($result as $row){

```

```

$array[$index] = $row['main_category_id'];
$index++;
}
return $array;
}
public function get_gradebook_main_items($setup_id){
$this->db->select('*');
$this->db->from('main_category');
$this->db->where('main_category_type', 'ITEM');
$this->db->where('gradebook_setup_details_id', $setup_id);
$query = $this->db->get();
$result = $query->result_array();
$array = array();
$index = 0;
foreach($result as $row){
$array[$index] = $row['main_category_id'];
$index++;
}
return $array;
}
public function get_category_template_member($template_id){
$this->db->select('*');
$this->db->from('template_members');
$this->db->order_by('member_id', 'ASC');
$this->db->where('gradebook_template_id', $template_id);
$query = $this->db->get();
$result = $query->result_array();
$array = array();
$index = 0;
foreach($result as $row){
$array[$index] = $row['member_id'];
$index++;
}
return $array;
}
public function get_gradebook_parameters($setup_id){
$this->db->select('*');
$this->db->from('main_category');
$this->db->order_by('main_category_type', 'ASC');
$this->db->order_by('main_category_id', 'ASC');
$this->db->where('gradebook_setup_details_id', $setup_id);
$query = $this->db->get();
$result = $query->result_array();
$array = array();
$index = 0;
foreach($result as $row){
$array[$index] = $row['main_category_id'];
$index++;
}
return $array;
}
public function get_gradebook_parameters_weight_template($template_id){
$this->db->select('*');
$this->db->from('template_category');
$this->db->order_by('template_category_type', 'ASC');
$this->db->order_by('template_category_id', 'ASC');
$this->db->where('gradebook_template_id', $template_id);
$query = $this->db->get();
$result = $query->result_array();
$array = array();

```



```

    $index = 0;
    foreach($result as $row){
    $array[$index] = $row['template_category_weight'];
    $index++;
    }
    return $array;
}

public function get_gradebook_parameters_points_template($template_id){
    $this->db->select('*');
    $this->db->from('template_category');
    $this->db->order_by('template_category_type','ASC');
    $this->db->order_by('template_category_id','ASC');
    $this->db->where('gradebook_template_id',$template_id);
    $query = $this->db->get();
    $result = $query->result_array();
    $array = array();
    $index = 0;
    foreach($result as $row){
    $array[$index] = $row['total_points'];
    $index++;
    }
    return $array;
}

public function get_gradebook_parameters_names_template($template_id){
    $this->db->select('*');
    $this->db->from('template_category');
    $this->db->order_by('template_category_type','ASC');
    $this->db->order_by('template_category_id','ASC');
    $this->db->where('gradebook_template_id',$template_id);
    $query = $this->db->get();
    $result = $query->result_array();
    $array = array();
    $index = 0;
    foreach($result as $row){
    $array[$index] = $row['template_category_name'];
    $index++;
    }
    return $array;
}

public function get_gradebook_parameters_types_template($template_id){
    $this->db->select('*');
    $this->db->from('template_category');
    $this->db->order_by('template_category_type','ASC');
    $this->db->order_by('template_category_id','ASC');
    $this->db->where('gradebook_template_id',$template_id);
    $query = $this->db->get();
    $result = $query->result_array();
    $array = array();
    $index = 0;
    foreach($result as $row){
    $array[$index] = $row['template_category_type'];
    $index++;
    }
    return $array;
}

public function get_gradebook_parameters_template($template_id){
    $this->db->select('*');
    $this->db->from('template_category');
    $this->db->order_by('template_category_type','ASC');
    $this->db->order_by('template_category_id','ASC');
}

$this->db->where('gradebook_template_id',$template_id);
$query = $this->db->get();
$result = $query->result_array();
$array = array();
$index = 0;
foreach($result as $row){
    $array[$index] = $row['template_category_id'];
    $index++;
}
return $array;
}

public function get_gradebook_items_parameters_template($template_id,
    $category_id){
    $this->db->select('*');
    $this->db->from('template_item');
    $this->db->order_by('template_item_id','ASC');
    $this->db->where('gradebook_template_id',$template_id);
    $this->db->where('template_category_id',$category_id);
    $query = $this->db->get();
    $result = $query->result_array();
    $array = array();
    $index = 0;
    foreach($result as $row){
    $array[$index] = $row['template_item_id'];
    $index++;
    }
    return $array;
}

public function get_gradebook_items_parameters_weight_template(
    $template_id, $category_id){
    $this->db->select('*');
    $this->db->from('template_item');
    $this->db->order_by('template_item_id','ASC');
    $this->db->where('gradebook_template_id',$template_id);
    $this->db->where('template_category_id',$category_id);
    $query = $this->db->get();
    $result = $query->result_array();
    $array = array();
    $index = 0;
    foreach($result as $row){
    $array[$index] = $row['template_item_weight'];
    $index++;
    }
    return $array;
}

public function get_gradebook_items_parameters_points_template(
    $template_id, $category_id){
    $this->db->select('*');
    $this->db->from('template_item');
    $this->db->order_by('template_item_id','ASC');
    $this->db->where('gradebook_template_id',$template_id);
    $this->db->where('template_category_id',$category_id);
    $query = $this->db->get();
    $result = $query->result_array();
    $array = array();
    $index = 0;
    foreach($result as $row){
    $array[$index] = $row['total_points'];
    $index++;
    }
}

```

```

return $array;
}

public function get_gradebook_items_parameters_names_template($template_id
    , $category_id){
$this->db->select('*');
$this->db->from('template_item');
$this->db->order_by('template_item_id','ASC');
$this->db->where('gradebook_template_id',$template_id);
$this->db->where('template_category_id',$category_id);
$query = $this->db->get();
$result = $query->result_array();
$array = array();
$index = 0;
foreach($result as $row){
$array[$index] = $row['template_item_name'];
$index++;
}
return $array;
}

public function delete_gradebook_categories($gradebook_id,$count = NULL,
    $remove_ids = NULL){
if($count == NULL){
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_type',"CATEGORY");
$this->db->delete('main_category');
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->delete('grade_item');
} else {
foreach($remove_ids as $id){
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_type',"CATEGORY");
$this->db->where('main_category_id',$id);
$this->db->delete('main_category');
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$id);
$this->db->delete('grade_item');
}
}
}

public function update_gradebook_categories($gradebook_id,$count = NULL,
    $update_ids = NULL){
$test = 0;
for($index = 0; $index < $count; $index++){
$real_index = $update_ids[$test];
$data = array(
    'main_category_name' => $this->input->post('category_name')[$real_index],
    'main_category_weight' => 0,
);
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_type',"CATEGORY");
$this->db->where('main_category_id', $this->input->post('category_ids')[$real_index]);
$this->db->update('main_category',$data);
$test++;
}
}

public function insert_gradebook_categories($gradebook_id,$count ,
    $index_array){
$test = 0;
for($index = 0; $index < $count; $index++){
$real_index = $index_array[$test];
$data = array(
    'gradebook_setup_details_id' => $gradebook_id,
    'main_category_name' => $this->input->post('category_name')[$real_index],
    'main_category_weight' => 0,
    'main_category_type' => "CATEGORY",
    'weight_to_total' => 0,
    'method_of_computation' => "Mean_of_Grades",
    'total_points' => 0,
    'final_grade' => 100
);
$this->db->insert('main_category',$data);
$test++;
}
}

public function insert_gradebook_categories_template($gradebook_id,$count ,
    $index_array, $category_names, $category_types){
$new_cat_ids = array();
$test = 0;
for($index = 0; $index < $count; $index++){
$real_index = $index_array[$test];
if($category_types[$real_index] == "CATEGORY"){
$data = array(
    'gradebook_setup_details_id' => $gradebook_id,
    'main_category_name' => $category_names[$real_index],
    'main_category_weight' => 0,
    'main_category_type' => "CATEGORY",
    'weight_to_total' => 0,
    'method_of_computation' => "Mean_of_Grades",
    'total_points' => 0,
    'final_grade' => 100
);
$this->db->insert('main_category',$data);
$new_cat_ids[] = $this->db->insert_id();
}
$test++;
}
return $new_cat_ids;
}

public function delete_gradebook_items($gradebook_id,$count = NULL,
    $remove_ids = NULL){
if($count == NULL){
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_type',"ITEM");
$this->db->delete('main_category');
} else {
foreach($remove_ids as $id){
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_type',"ITEM");
$this->db->where('main_category_id',$id);
$this->db->delete('main_category');
}
}
}

public function update_gradebook_items($gradebook_id,$count = NULL,
    $update_ids = NULL){
$test = 0;
for($index = 0; $index < $count; $index++){
$real_index = $update_ids[$test];
$data = array(

```

```

'main_category_name' => $this->input->post('category_name')[$real_index],
'main_category_weight' => 0,
'total_points' => $this->input->post('category_total')[$real_index]
);
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_type',"ITEM");
$this->db->where('main_category_id', $this->input->post('category_ids')[$real_index]);
$this->db->update('main_category',$data);
$test++;
}
}
public function insert_gradebook_items($gradebook_id,$count,$index_array){
$test = 0;
for($index = 0; $index < $count; $index++){
$real_index = $index_array[$test];
$data = array(
'gradebook_setup_details_id' => $gradebook_id,
'main_category_name' => $this->input->post('category_name')[$real_index],
'main_category_weight' => 0,
'main_category_type' => "ITEM",
'weight_to_total' => 0,
'method_of_computation' => "NO_METHOD",
'total_points' => $this->input->post('category_total')[$real_index],
'final_grade' => 0
);
$this->db->insert('main_category',$data);
$test++;
}
}
public function insert_gradebook_items_template($gradebook_id,$count,
$index_array,$category_names, $category_points, $type){
$test = 0;
for($index = 0; $index < $count; $index++){
$real_index = $index_array[$test];
if($type[$real_index] === "ITEM"){
$data = array(
'gradebook_setup_details_id' => $gradebook_id,
'main_category_name' => $category_names[$real_index],
'main_category_weight' => 0,
'main_category_type' => "ITEM",
'weight_to_total' => 0,
'method_of_computation' => "NO_METHOD",
'total_points' => $category_points[$real_index],
'final_grade' => 0
);
$this->db->insert('main_category',$data);
}
$test++;
}
}
public function get_template_category_ids($template_id){
$this->db->select('template_category_id,method_of_computation,
template_category_name');
$this->db->from('template_category');
$this->db->order_by('template_category_type','ASC');
$this->db->order_by('template_category_id','ASC');
$this->db->where('template_category_type','CATEGORY');
$this->db->where('gradebook_template_id',$template_id);
$query = $this->db->get();
$result = $query->result_array();
return $result;
}
public function update_method($gradebook_id,$method,$array_ids = NULL,
$array_weights = NULL,$array_points = NULL,$total_points = NULL){
$data = array('method_of_computation' => $method);
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->update('gradebook_setup_details',$data);
if($method === "Weighted_Mean_of_Grades"){
if(count($array_ids) > 0){
$index = 0;
foreach($array_ids as $id){
$data = array(
'main_category_weight' => $array_weights[$index]
);
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$id);
$this->db->update('main_category',$data);
$index++;
}
}
} else if ($method === "Mean_of_Grades") {
if(count($array_ids) > 0){
$index = 0;
foreach($array_ids as $id){
$data = array(
'main_category_weight' => (100 / count($array_ids))
);
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$id);
$this->db->update('main_category',$data);
$index++;
}
}
} else {
if(count($array_ids) > 0){
$index = 0;
foreach($array_ids as $id){
$data = array(
'main_category_weight' => ($array_points[$index]/$total_points) * 100
);
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$id);
$this->db->update('main_category',$data);
$index++;
}
}
}
}
public function get_category_details_by_setup($setup_id,$category_id){
$this->db->select('*');
$this->db->from('main_category');
$this->db->where('gradebook_setup_details_id',$setup_id);
$this->db->where('main_category_id',$category_id);
$query = $this->db->get();
return $query->row_array();
}
}
public function get_category_items_by_id($setup_id,$category_id){
$this->db->select('*');
$this->db->from('grade_item');
}

```

```

$this->db->where('gradebook_setup_details_id',$setup_id);
$this->db->where('main_category_id',$category_id);
$query = $this->db->get();
return $query->result_array();
}

public function get_category_items($category_id,$setup_id){
$this->db->select('*');
$this->db->from('grade_item');
$this->db->where('main_category_id',$category_id);
$this->db->where('gradebook_setup_details_id',$setup_id);
$this->db->order_by('grade_item_id');
$query = $this->db->get();
$result = $query->result_array();
$array = array();
$index = 0;
foreach($result as $row){
$array[$index] = $row['grade_item_id'];
$index++;
}
return $array;
}

public function delete_category_items($gradebook_id,$category_id,$count =
NULL,$remove_ids = NULL){
if($count == NULL){
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$category_id);
$this->db->delete('grade_item');
} else {
foreach($remove_ids as $id){
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$category_id);
$this->db->where('grade_item_id',$id);
$this->db->delete('grade_item');
}
}

public function update_category_items($gradebook_id,$category_id,$count =
NULL,$update_ids = NULL){
$test = 0;
for($index = 0; $index < $count; $index++){
$real_index = $update_ids[$test];
$data = array(
'grade_item_name' => $this->input->post('category_name')[$real_index],
'grade_item_weight' => 0,
'total_points' => $this->input->post('category_total')[$real_index]
);
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$category_id);
$this->db->where('grade_item_id', $this->input->post('category_ids')[$real_index]);
$this->db->update('grade_item',$data);
$test++;
}
}

public function insert_category_items($gradebook_id,$category_id,$count ,
$index_array){
$test = 0;
for($index = 0; $index < $count; $index++){
$real_index = $index_array[$test];
$data = array(
'gradebook_setup_details_id' => $gradebook_id,
'grade_item_name' => $this->input->post('category_name')[$real_index],
'grade_item_weight' => 0,
'weight_to_total' => 0,
'total_points' => $this->input->post('category_total')[$real_index],
'main_category_id' => $category_id
);
$this->db->insert('grade_item',$data);
$test++;
}

public function insert_category_items_template($gradebook_id,$category_id,
$count,$index_array, $names, $points){
$test = 0;
for($index = 0; $index < $count; $index++){
$real_index = $index_array[$test];
$data = array(
'gradebook_setup_details_id' => $gradebook_id,
'grade_item_name' => $names[$real_index],
'grade_item_weight' => 0,
'weight_to_total' => 0,
'total_points' => $points[$real_index],
'main_category_id' => $category_id
);
$this->db->insert('grade_item',$data);
$test++;
}

public function update_category_method_template($gradebook_id,$cat_name,
$method,$category_id,$array_ids = NULL,$array_weights = NULL,
$array_points = NULL,$total_category_points = NULL){
$data = array('method_of_computation' => $method,'main_category_name'=>
$cat_name);
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$category_id);
$this->db->update('main_category',$data);
$sum = 0;
$max_grade = 0;
if($method === "Weighted_Mean_of_Grades"){
if(count($array_ids) > 0){
$index = 0;
foreach($array_ids as $id){
$data = array(
'grade_item_weight' => $array_weights[$index]
);
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$category_id);
$this->db->where('grade_item_id',$id);
$this->db->update('grade_item',$data);
$sum += $array_points[$index];
$max_grade += $array_weights[$index];
$index++;
}
} else if ($method === "Mean_of_Grades") {
if(count($array_ids) > 0){
$index = 0;
foreach($array_ids as $id){
$data = array(
'grade_item_weight' => (100 / count($array_ids))

```

```

);
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$category_id);
$this->db->where('grade_item_id',$id);
$this->db->update('grade_item',$data);
$sum += $array_points[$index];
$index++;
}
}
} else {
if(count($array_ids) > 0){
$index = 0;
foreach($array_ids as $id){
$data = array(
'grade_item_weight' => ($array_points[$index]/$total_category_points) *
100
);
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$category_id);
$this->db->where('grade_item_id',$id);
$this->db->update('grade_item',$data);
$sum += $array_points[$index];
$index++;
}
}
}

if($method === "Weighted_Mean_of_Grades"){
$data = array(
'total_points' => $sum,
'final_grade' => $max_grade
);
} else {
$data = array('total_points' => $sum,'final_grade' => 100);
}

$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$category_id);
$this->db->update('main_category',$data);
}

public function update_category_method($gradebook_id,$method,$category_id,
$array_ids = NULL,$array_weights = NULL,$array_points = NULL,
$total_category_points = NULL){
$data = array('method_of_computation' => $method,'main_category_name'=>
$this->input->post('gradebook_category_name'));
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$category_id);
$this->db->update('main_category',$data);
$sum = 0;
$max_grade = 0;
if($method === "Weighted_Mean_of_Grades"){
if(count($array_ids) > 0){
$index = 0;
foreach($array_ids as $id){
$data = array(
'grade_item_weight' => $array_weights[$index]
);
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$category_id);
$this->db->where('grade_item_id',$id);
$this->db->update('grade_item',$data);
$sum += $this->input->post('category_total')[$index];
}
}
} else if ($method === "Mean_of_Grades") {
if(count($array_ids) > 0){
$index = 0;
foreach($array_ids as $id){
$data = array(
'grade_item_weight' => (100 / count($array_ids))
);
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$category_id);
$this->db->where('grade_item_id',$id);
$this->db->update('grade_item',$data);
$sum += $this->input->post('category_total')[$index];
$index++;
}
}
} else {
if(count($array_ids) > 0){
$index = 0;
foreach($array_ids as $id){
$data = array(
'grade_item_weight' => ($array_points[$index]/$total_category_points) *
100
);
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$category_id);
$this->db->where('grade_item_id',$id);
$this->db->update('grade_item',$data);
$sum += $this->input->post('category_total')[$index];
$index++;
}
}
}

if($method === "Weighted_Mean_of_Grades"){
$data = array(
'total_points' => $sum,
'final_grade' => $max_grade
);
} else {
$data = array('total_points' => $sum,'final_grade' => 100);
}

$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$category_id);
$this->db->update('main_category',$data);
}

public function update_category_weights($gradebook_id,$category_id){
$this->db->select('*');
$this->db->from('main_category');
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$category_id);
$query2 = $this->db->get();
$result2 = $query2->row_array();
$current_category_weight = $result2['main_category_weight'];
$this->db->select('*');
$this->db->from('grade_item');
$this->db->where('gradebook_setup_details_id',$gradebook_id);
$this->db->where('main_category_id',$category_id);
}

```

```

$query = $this->db->get();
$result = $query->result_array();
foreach($result as $row){
    $data = array('weight_to_total' => (($current_category_weight/100) * $row[
        'grade_item_weight']));
    $this->db->where('gradebook_setup_details_id', $gradebook_id);
    $this->db->where('main_category_id', $category_id);
    $this->db->where('grade_item_id', $row['grade_item_id']);
    $this->db->update('grade_item', $data);
}
}

public function update_category_weights_with_removed_items($gradebook_id,
    $category_ids){
    foreach($category_ids as $id){
        $this->db->select('*');
        $this->db->from('main_category');
        $this->db->where('gradebook_setup_details_id', $gradebook_id);
        $this->db->where('main_category_id', $id);
        $query2 = $this->db->get();
        $result2 = $query2->row_array();
        $current_category_weight = $result2['main_category_weight'];
        $this->db->select('*');
        $this->db->from('grade_item');
        $this->db->where('gradebook_setup_details_id', $gradebook_id);
        $this->db->where('main_category_id', $id);
        $query = $this->db->get();
        $result = $query->result_array();
        foreach($result as $row){
            $data = array('weight_to_total' => (($current_category_weight/100) * $row[
                'grade_item_weight']));
            $this->db->where('gradebook_setup_details_id', $gradebook_id);
            $this->db->where('main_category_id', $id);
            $this->db->where('grade_item_id', $row['grade_item_id']);
            $this->db->update('grade_item', $data);
        }
    }
}

public function get_category_parameters($setup_id, $category_id){
    $this->db->select('*');
    $this->db->from('grade_item');
    $this->db->order_by('grade_item_id', 'ASC');
    $this->db->where('gradebook_setup_details_id', $setup_id);
    $this->db->where('main_category_id', $category_id);
    $query = $this->db->get();
    $result = $query->result_array();
    $array = array();
    $index = 0;
    foreach($result as $row){
        $array[$index] = $row['grade_item_id'];
        $index++;
    }
    return $array;
}

public function get_category_items_of_gradebook($category_ids, $setup_id){
    $this->db->select('*');
    $this->db->from('grade_item');
    $this->db->where_in('main_category_id', $category_ids);
    $this->db->order_by('main_category_id', 'ASC');
    $this->db->order_by('grade_item_id', 'ASC');
    $this->db->where('gradebook_setup_details_id', $setup_id);
    $query = $this->db->get();
    return $query->result_array();
}

public function check_category_name($category_name, $gradebook_id,
    $category_id){
    $this->db->select('*');
    $this->db->from('main_category');
    $this->db->where('main_category_name', $category_name);
    $this->db->where('gradebook_setup_details_id', $gradebook_id);
    $this->db->where('main_category_id !=', $category_id);
    $query = $this->db->get();
    if($query -> num_rows() == 1)
    {
        return true;
    }
    else
    {
        return false;
    }
}

public function move_category_items($grade_items, $setup_id){
    $this->db->select('*');
    $this->db->from('grade_item');
    $this->db->where('gradebook_setup_details_id', $setup_id);
    $this->db->where_in('grade_item_id', $grade_items);
    $query = $this->db->get();
    $array = $query->result_array();
    return $array;
}

public function move_main_category_items($grade_items, $setup_id){
    $this->db->select('*');
    $this->db->from('main_category');
    $this->db->where('main_category_type', 'ITEM');
    $this->db->where('gradebook_setup_details_id', $setup_id);
    $this->db->where_in('main_category_id', $grade_items);
    $query = $this->db->get();
    $array = $query->result_array();
    return $array;
}

public function remove_transferred_items($array){
    $this->db->where_in('grade_item_id', $array);
    $this->db->delete('grade_item');
}

public function remove_transferred_category_items($array){
    $this->db->where_in('main_category_id', $array);
    $this->db->where('main_category_type', 'ITEM');
    $this->db->delete('main_category');
}

public function remove_grade_item($grade_item_id){
    $this->db->where('grade_item_id', $grade_item_id);
    $this->db->delete('grade_item');
}

public function remove_gradebook_item($category_id){
    $this->db->where('main_category_id', $category_id);
    $this->db->where('main_category_type', "ITEM");
    $this->db->delete('main_category');
}

public function remove_gradebook_category($category_id){
    $this->db->where('main_category_id', $category_id);
    $this->db->where('main_category_type', "CATEGORY");
    $this->db->delete('main_category');
}

```

```

$this->db->where('main_category_id', $category_id);
$this->db->delete('grade_item');
$this->db->where('main_category_id', $category_id);
$this->db->delete('class_record');
}

public function update_categories_with_removed_items($category_ids,
    $gradebook_id){
    foreach($category_ids as $id){
        $this->db->select('*');
        $this->db->from('main_category');
        $this->db->where('gradebook_setup_details_id', $gradebook_id);
        $this->db->where('main_category_id', $id);
        $query2 = $this->db->get();
        $result2 = $query2->row_array();
        $this->db->select('*');
        $this->db->from('grade_item');
        $this->db->where('gradebook_setup_details_id', $gradebook_id);
        $this->db->where('main_category_id', $id);
        $query = $this->db->get();
        $result = $query->result_array();
        $sum = 0;
        $max_grade = 0;
        foreach($result as $row){
            $sum += $row['total_points'];
            if($result2['method_of_computation'] === "Weighted_Mean_of_Grades"){
                $data = array(
                    'grade_item_weight' => $row['grade_item_weight']
                );
            }
            $this->db->where('gradebook_setup_details_id', $gradebook_id);
            $this->db->where('main_category_id', $row['main_category_id']);
            $this->db->update('main_category', $data);
            } else if ($result2['method_of_computation'] === "Mean_of_Grades" ) {
                $data = array(
                    'main_category_weight' => (100 / count($result2))
                );
            }
            $this->db->where('gradebook_setup_details_id', $gradebook_id);
            $this->db->where('main_category_id', $row['main_category_id']);
            $this->db->update('main_category', $data);
            } else {
                $data = array(
                    'main_category_weight' => ($row['total_points']/$result['total_points']) *
                        100
                );
            }
            $this->db->where('gradebook_setup_details_id', $gradebook_id);
            $this->db->where('main_category_id', $row['main_category_id']);
            $this->db->update('main_category', $data);
            }
        }
    }

    public function update_grade_item($grade_item_id){
        $data = array(
            'grade_item_name' => $this->input->post('grade_item_name'),
            'total_points' => $this->input->post('total_points'),
        );
        $this->db->where('grade_item_id', $grade_item_id);
        $this->db->update('grade_item', $data);
    }

    public function update_gradebook_item($category_id){
        $data = array(
            'main_category_name' => $this->input->post('category_name'),
            'total_points' => $this->input->post('total_points'),
        );
        $this->db->where('main_category_id', $category_id);
        $this->db->update('main_category', $data);
    }

    public function get_grade_item_names($setup_id){
        $this->db->select('*');
        $this->db->from('grade_item');

```

```

$this->db->order_by('grade_item_id');
$query = $this->db->get();
$result = $query->result_array();
$array = array();
foreach($result as $row){
$array[] = $row['grade_item_name'];
}
return $array;
}

public function get_grade_item_categories($setup_id){
$this->db->select('*');
$this->db->from('grade_item');
$this->db->order_by('grade_item_id');
$query = $this->db->get();
$result = $query->result_array();
$array = array();
foreach($result as $row){
$array[] = $row['main_category_id'];
}
return $array;
}

public function get_category_names($setup_id){
$this->db->select('*');
$this->db->from('main_category');
$query = $this->db->get();
$result = $query->result_array();
$array = array();
foreach($result as $row){
$array[] = $row['main_category_name'];
}
return $array;
}

public function update_total_points($setup_id){
$this->db->select_sum('total_points');
$this->db->where('gradebook_setup_details_id', $setup_id);
$this->db->from('main_category');
$query = $this->db->get();
$row = $query->row_array();
$value = $row['total_points'];
$data = array('total_points' => $value);
$this->db->where('gradebook_setup_details_id', $setup_id);
$this->db->update('gradebook_setup_details', $data);
return $value;
}

public function update_final_grade($setup_id){
$this->db->select_sum('main_category_weight');
$this->db->where('gradebook_setup_details_id', $setup_id);
$this->db->from('main_category');
$query = $this->db->get();
$row = $query->row_array();
$value = $row['main_category_weight'];
$data = array('max_grade' => $value);
$this->db->where('gradebook_setup_details_id', $setup_id);
$this->db->update('gradebook_setup_details', $data);
return $value;
}

public function update_total_category_points($category_id){
$this->db->select_sum('total_points');
$this->db->where('main_category_id', $category_id);
$this->db->from('grade_item');

$query = $this->db->get();
$row = $query->row_array();
$value = $row['total_points'];
$data = array('total_points' => $value);
$this->db->where('main_category_id', $category_id);
$this->db->update('main_category', $data);
return $value;
}

public function get_gradebook_parameters_weight($setup_id){
$this->db->select('*');
$this->db->from('main_category');
$this->db->order_by('main_category_type', 'ASC');
$this->db->order_by('main_category_id', 'ASC');
$this->db->where('gradebook_setup_details_id', $setup_id);
$query = $this->db->get();
$result = $query->result_array();
$array = array();
$index = 0;
foreach($result as $row){
$array[$index] = $row['main_category_weight'];
$index++;
}
return $array;
}

public function get_gradebook_parameters_types($setup_id){
$this->db->select('*');
$this->db->from('main_category');
$this->db->order_by('main_category_type', 'ASC');
$this->db->order_by('main_category_id', 'ASC');
$this->db->where('gradebook_setup_details_id', $setup_id);
$query = $this->db->get();
$result = $query->result_array();
$array = array();
$index = 0;
foreach($result as $row){
$array[$index] = $row['main_category_type'];
$index++;
}
return $array;
}

public function get_gradebook_parameters_points($setup_id){
$this->db->select('*');
$this->db->from('main_category');
$this->db->order_by('main_category_type', 'ASC');
$this->db->order_by('main_category_id', 'ASC');
$this->db->where('gradebook_setup_details_id', $setup_id);
$query = $this->db->get();
$result = $query->result_array();
$array = array();
$index = 0;
foreach($result as $row){
$array[$index] = $row['total_points'];
$index++;
}
return $array;
}

public function get_existing_templates($school_id, $teacher_id){
$this->db->select('*');
$this->db->from('gradebook_template');
$this->db->where('school_id', $school_id);

```



```

$this->db->where('teacher_id',$teacher_id);
$query = $this->db->get();
return $query->result_array();
}

public function check_template_name($template_name,$school_id,$teacher_id)
{
$this->db->select('*');
$this->db->from('gradebook_template');
$this->db->where('gradebook_template_name',$template_name);
$this->db->where('school_id',$school_id);
$this->db->where('teacher_id',$teacher_id);
$query = $this->db->get();
if($query -> num_rows() == 1)
{
return true;
}
else
{
return false;
}
}

public function update_check_template_name($template_name,$school_id,
$template_id){
$this->db->select('*');
$this->db->from('gradebook_template');
$this->db->where('school_id',$school_id);
$this->db->where('gradebook_template_name',$template_name);
$this->db->where('gradebook_template_id!',$template_id);
$query = $this->db->get();
if($query -> num_rows() == 1)
{
return true;
}
else
{
return false;
}
}

public function insert_gradebook_template($name,$teacher_id){
$data = array(
'school_id' => $_SESSION['logged_in']['school_id'],
'teacher_id' => $teacher_id,
'gradebook_template_name' => $name
);
$this->db->insert('gradebook_template',$data);
$this->db->select('*');
$this->db->from('gradebook_template');
$this->db->where('gradebook_template_name',$name);
$query = $this->db->get();
$result = $query->row_array();
return $result['gradebook_template_id'];
}

public function get_gradebook_main_category_ids($setup_id){
$this->db->select('*');
$this->db->from('main_category');
$this->db->order_by('main_category_id','ASC');
$this->db->where('gradebook_setup_details_id',$setup_id);
$query = $this->db->get();
$result = $query->result_array();
$array = array();

$index = 0;
foreach($result as $row){
$array[$index] = $row['main_category_id'];
$index++;
}

return $array;
}

public function get_gradebook_main_category_for_template($setup_id){
$this->db->select('*');
$this->db->from('main_category');
$this->db->order_by('main_category_id','ASC');
$this->db->where('gradebook_setup_details_id',$setup_id);
$query = $this->db->get();
$result = $query->result_array();
return $result;
}

public function get_grade_item_for_template($setup_id){
$this->db->select('*');
$this->db->from('grade_item');
$this->db->order_by('grade_item_id','ASC');
$this->db->where('gradebook_setup_details_id',$setup_id);
$query = $this->db->get();
$result = $query->result_array();
return $result;
}

public function get_grade_item_ids_by_setup($setup_id){
$this->db->select('*');
$this->db->from('grade_item');
$this->db->order_by('grade_item_id','ASC');
$this->db->where('gradebook_setup_details_id',$setup_id);
$query = $this->db->get();
$result = $query->result_array();
$array = array();

$index = 0;
foreach($result as $row){
$array[$index] = $row['grade_item_id'];
$index++;
}

return $array;
}

public function insert_template_elements($template_id,$category,$item){
if(count($category) > 0){
foreach($category as $row){
$data = array(
'gradebook_template_id' => $template_id,
'main_category_id' => $row['main_category_id'],
'template_category_name' => $row['main_category_name'],
'template_category_type' => $row['main_category_type'],
'template_category_weight' => $row['main_category_weight'],
'weight_to_total' => 0,
'method_of_computation' => $row['method_of_computation'],
'total_points' => $row['total_points'],
'final_grade' => $row['final_grade']
);
$this->db->insert('template_category',$data);
}

if(count($item) > 0){
foreach($item as $row){
$this->db->select('*');
}
}
}
}

```

```

$this->db->from('template_category');
$this->db->where('main_category_id', $row['main_category_id']);
$query = $this->db->get();
$result = $query->row_array();
$tc_id = $result['template_category_id'];
$data = array(
    'gradebook_template_id' => $template_id,
    'template_item_name' => $row['grade_item_name'],
    'template_item_weight' => $row['grade_item_weight'],
    'weight_to_total' => 0,
    'total_points' => $row['total_points'],
    'template_category_id' => $tc_id
);
$this->db->insert('template_item',$data);
}
}

public function remove_template_elements($template_id){
    $this->db->trans_start();
    $this->db->where('gradebook_template_id',$template_id);
    $this->db->delete('template_category');
    $this->db->trans_complete();
    $this->db->trans_start();
    $this->db->where('gradebook_template_id',$template_id);
    $this->db->delete('template_item');
    $this->db->trans_complete();
}

public function remove_all_setup_elements($setup_id){
    $this->db->where('gradebook_setup_details_id',$setup_id);
    $this->db->delete('main_category');
    $this->db->where('gradebook_setup_details_id',$setup_id);
    $this->db->delete('grade_item');
}

public function get_template_category_members($template_id){
    $this->db->select('*');
    $this->db->from('template_members');
    $this->db->where('gradebook_template_id',$template_id);
    $this->db->where('member_type','CATEGORY');
    $this->db->order_by('member_id','ASC');
    $query = $this->db->get();
    return $query->result_array();
}

public function get_template_item_members($template_id){
    $this->db->select('*');
    $this->db->from('template_members');
    $this->db->where('gradebook_template_id',$template_id);
    $this->db->where('member_type','ITEM');
    $this->db->order_by('member_id','ASC');
    $query = $this->db->get();
    return $query->result_array();
}

public function get_main_categories_by_ids($category_ids){
    $this->db->select('*');
    $this->db->from('main_category');
    $this->db->where_in('main_category_id',$category_ids);
    $this->db->order_by('main_category_id','ASC');
    $query = $this->db->get();
    return $query->result_array();
}

public function get_grade_items_by_ids($item_ids){
    $this->db->select('*');
    $this->db->from('grade_item');
    $this->db->where_in('grade_item_id',$item_ids);
    $this->db->order_by('grade_item_id','ASC');
    $query = $this->db->get();
    return $query->result_array();
}

}

public function insert_categories_from_templates($categories,$setup_id){
    $new_cat_ids = array();
    foreach($categories as $row){
        $data = array(
            'gradebook_setup_details_id' => $setup_id,
            'main_category_name' => $row['main_category_name'],
            'main_category_type' => $row['main_category_type'],
            'main_category_weight' => $row['main_category_weight'],
            'weight_to_total' => 0,
            'method_of_computation' => $row['method_of_computation'],
            'total_points' => $row['total_points'],
            'final_grade' => $row['final_grade']
        );
        $this->db->insert('main_category',$data);
        $new_cat_ids[] = $this->db->insert_id();
    }
    return $new_cat_ids;
}

public function insert_items_from_templates($items,$new_cat_ids,
    $old_cat_ids,$setup_id){
    foreach($items as $row){
        $index = array_search($row['main_category_id'],$old_cat_ids);
        $new_id = $new_cat_ids[$index];
        $data = array(
            'gradebook_setup_details_id' => $setup_id,
            'grade_item_name' => $row['grade_item_name'],
            'grade_item_weight' => $row['grade_item_weight'],
            'weight_to_total' => 0,
            'total_points' => $row['total_points'],
            'main_category_id' => $new_id
        );
        $this->db->insert('grade_item',$data);
        $new_cat_ids[] = $this->db->insert_id();
    }
}

public function get_gradebook_templates($school_id,$teacher_id){
    $this->db->select('*');
    $this->db->from('gradebook_template');
    $this->db->where('school_id',$school_id);
    $this->db->where('teacher_id',$teacher_id);
    $query = $this->db->get();
    return $query->result_array();
}

public function get_gradebook_scales($school_id,$teacher_id){
    $this->db->select('*');
    $this->db->from('scale_template');
    $this->db->where('school_id',$school_id);
    $this->db->where('teacher_id',$teacher_id);
    $query = $this->db->get();
    return $query->result_array();
}

}

public function delete_gradebook_template($template_id){
    $this->db->where('gradebook_template_id',$template_id);

```

```

$this->db->delete('template_members');
$this->db->where('gradebook_template_id',$template_id);
$this->db->delete('gradebook_template');
}
public function delete_gradebook_scale($template_id){
$this->db->where('scale_template_id',$template_id);
$this->db->delete('scale_template_members');
$this->db->where('scale_template_id',$template_id);
$this->db->delete('scale_template');
}
public function update_template_name($template_id,$template_name){
$data = array(
'gradebook_template_name' => $template_name
);
$this->db->where('gradebook_template_id',$template_id);
$this->db->update('gradebook_template',$data);
}
public function update_scale_name($template_id,$template_name){
$data = array(
'scale_template_name' => $template_name
);
$this->db->where('scale_template_id',$template_id);
$this->db->update('scale_template',$data);
}
public function get_grading_scales($course_id){
$this->db->select('*');
$this->db->from('grading_scale');
$this->db->where('course_id',$course_id);
$query = $this->db->get();
$array = $query->row_array();
$scale_id = $array['grading_scale_id'];
$this->db->select('*');
$this->db->from('scale_values');
$this->db->where('grading_scale_id',$scale_id);
$this->db->order_by('scale_values_higher_limit','DESC');
$query2 = $this->db->get();
return $query2->result_array();
}
public function get_grading_scale_id($course_id){
$this->db->select('*');
$this->db->from('grading_scale');
$this->db->where('course_id',$course_id);
$query = $this->db->get();
$array = $query->row_array();
$scale_id = $array['grading_scale_id'];
return $scale_id;
}
public function get_grading_scale_ids($course_id){
$this->db->select('*');
$this->db->from('grading_scale');
$this->db->where('course_id',$course_id);
$query = $this->db->get();
$array = $query->row_array();
$scale_id = $array['grading_scale_id'];
$this->db->select('*');
$this->db->from('scale_values');
$this->db->where('grading_scale_id',$scale_id);
$this->db->order_by('scale_values_higher_limit','DESC');
$query2 = $this->db->get();
$result = $query2->result_array();
}
}

$array2 = array();
$index = 0;
foreach($result as $row){
$array2[$index] = $row['scale_values_id'];
$index++;
}
return $array2;
}
public function delete_gradebook_scales($scale_id,$count = NULL,
$remove_ids = NULL){
if($count == NULL){
$this->db->where('grading_scale_id',$scale_id);
$this->db->delete('scale_values');
} else {
foreach($remove_ids as $id){
$this->db->where('grading_scale_id',$scale_id);
$this->db->where('scale_values_id',$id);
$this->db->delete('scale_values');
}
}
}
public function update_gradebook_scales($scale_id,$count = NULL,
$update_ids = NULL,$higher_limit = NULL){
$test = 0;
for($index = 0; $index < $count; $index++){
$real_index = $update_ids[$test];
$data = array(
'scale_values_label' => $this->input->post('scale_values_label')[
$real_index],
'scale_values_lower_limit' => $this->input->post('scale_values_lower_limit
')[$real_index],
'scale_values_higher_limit' => $higher_limit[$real_index]
);
$this->db->where('grading_scale_id',$scale_id);
$this->db->where('scale_values_id', $this->input->post('scale_values_id')[
$real_index]);
$this->db->update('scale_values',$data);
$test++;
}
}
public function insert_gradebook_scales($scale_id,$count,$index_array,
$higher_limit = NULL){
$test = 0;
for($index = 0; $index < $count; $index++){
$real_index = $index_array[$test];
$data = array(
'grading_scale_id' => $scale_id,
'scale_values_label' => $this->input->post('scale_values_label')[
$real_index],
'scale_values_lower_limit' => $this->input->post('scale_values_lower_limit
')[$real_index],
'scale_values_higher_limit' => $higher_limit[$real_index]
);
$this->db->insert('scale_values',$data);
$test++;
}
}
public function insert_gradebook_scale($name,$teacher_id){
$data = array(
'school_id' => $_SESSION['logged_in']['school_id'],

```

```

'teacher_id' => $teacher_id,
'scale_template_name' => $name
);
$this->db->insert('scale_template',$data);
$this->db->select('*');
$this->db->from('scale_template');
$this->db->where('scale_template_name',$name);
$query = $this->db->get();
$result = $query->row_array();
return $result['scale_template_id'];
}

public function get_scale_template_elements($course_id){
$this->db->select('*');
$this->db->from('grading_scale');
$this->db->where('course_id',$course_id);
$query2 = $this->db->get();
$result2 = $query2->row_array();
$scale_id = $result2['grading_scale_id'];
$this->db->select('*');
$this->db->from('scale_values');
$this->db->where('grading_scale_id',$scale_id);
$this->db->order_by('scale_values_higher_limit','DESC');
$query = $this->db->get();
$result = $query->result_array();
$array = array();
$index = 0;
foreach($result as $row){
$array[$index] = $row['scale_values_id'];
$index++;
}
return $array;
}

public function insert_scale_template_elements($template_id,$category_ids)
{
if(count($category_ids) > 0){
foreach($category_ids as $key => $value){
$data = array(
'scale_template_id' => $template_id,
'scale_values_id' => $value,
);
$this->db->insert('scale_template_members',$data);
}
}
}

public function check_scale_name($template_name,$school_id,$teacher_id){
$this->db->select('*');
$this->db->from('scale_template');
$this->db->where('scale_template_name',$template_name);
$this->db->where('school_id',$school_id);
$this->db->where('teacher_id',$teacher_id);
$query = $this->db->get();
if($query -> num_rows() == 1)
{
return true;
}
else
{
return false;
}
}

public function update_check_scale_name($template_name,$school_id,
$template_id){
$this->db->select('*');
$this->db->from('scale_template');
$this->db->where('scale_template_name',$template_name);
$this->db->where('school_id',$school_id);
$this->db->where('scale_template_id',$template_id);
$query = $this->db->get();
if($query -> num_rows() == 1)
{
return true;
}
else
{
return false;
}
}

public function get_scale_templates($school_id,$teacher_id){
$this->db->select('*');
$this->db->from('scale_template');
$this->db->where('school_id',$school_id);
$this->db->or_where('teacher_id',$teacher_id);
$this->db->or_where('teacher_id', 0);
$query = $this->db->get();
return $query->result_array();
}

public function remove_scale_template_elements($template_id){
$this->db->where('scale_template_id',$template_id);
$this->db->delete('scale_template_members');
}

public function get_grading_scale_by_course_id($course_id){
$this->db->select('*');
$this->db->from('grading_scale');
$this->db->where('course_id',$course_id);
$query = $this->db->get();
$result = $query->row_array();
return $result['grading_scale_id'];
}

public function get_scale_template_elements_by_scale_id($template_id){
$this->db->select('*');
$this->db->from('scale_template_members');
$this->db->where('scale_template_id',$template_id);
$this->db->order_by('scale_values_id','ASC');
$query = $this->db->get();
$result = $query->result_array();
$array = array();
$index = 0;
foreach($result as $row){
$array[$index] = $row['scale_values_id'];
$index++;
}
return $array;
}

public function get_scale_values_by_ids($scale_values_ids){
$this->db->select('*');
$this->db->from('scale_values');
$this->db->where_in('scale_values_id',$scale_values_ids);
$this->db->order_by('scale_values_higher_limit','DESC');
$query = $this->db->get();
}

```

```

return $query->result_array();
}

public function insert_new_scale_template_elements($scale_id,$scale_values
){
foreach($scale_values as $row){
$data = array(
'grading_scale_id' => $scale_id,
'scale_values_label' => $row['scale_values_label'],
'scale_values_lower_limit' => $row['scale_values_lower_limit'],
'scale_values_higher_limit' => $row['scale_values_higher_limit']
);
$this->db->insert('scale_values',$data);
}
}

public function update_class_record($gradebook_id, $assign_new_students =
NULL){
$sql = 'SELECT course_id FROM gradebook_setup_details WHERE
gradebook_setup_details_id=' . $this->db->escape($gradebook_id);
$query = $this->db->query($sql);
$result = $query->row_array();
$course_id = $result['course_id'];
$alphabet = array('B','C','D','E','F','G','H','I','J','K','L','M','N','O',
'P','Q','R','S','T','U','V','W','X','Y','Z','AA','AB','AC','AD','AE',
'AF','AG','AH','AJ','AK','AL','AM','AN','AD','AP','AQ','AR','AS',
'AT','AU','AV','AW','AX','AY','AZ');
$sql = 'SELECT main_category_id FROM class_record WHERE
gradebook_setup_details_id=' . $this->db->escape($gradebook_id) . '
GROUP BY main_category_id';
$query = $this->db->query($sql);
$existing_id_array = array_column($query->result_array(),"main_category_id
");
$existing_cat_count = count($existing_id_array);
$sql = 'SELECT COUNT(*) as count_cat FROM main_category WHERE
gradebook_setup_details_id=' . $this->db->escape($gradebook_id) . ' AND
main_category_type="CATEGORY"';
$query = $this->db->query($sql);
$result = $query->row_array();
$main_category_count = $result['count_cat'];
$sql = 'SELECT COUNT(*) as count_item FROM main_category WHERE
gradebook_setup_details_id=' . $this->db->escape($gradebook_id) . ' AND
main_category_type="ITEM"';
$query = $this->db->query($sql);
$result = $query->row_array();
$main_item_count = $result['count_item'];
$total_category_count = $main_category_count + $main_item_count;
$sql = 'SELECT student_id, order_number FROM course_students WHERE
course_id=' . $this->db->escape($course_id) . ' ORDER BY order_number';
$query = $this->db->query($sql);
$student_details = $query->result_array();
if($existing_cat_count < $total_category_count){
$sql = 'SELECT main_category_id FROM main_category WHERE
gradebook_setup_details_id=' . $this->db->escape($gradebook_id);
$query = $this->db->query($sql);
$current_id_array = array_column($query->result_array(),"main_category_id
");
$result = array_diff($current_id_array,$existing_id_array);
foreach($result as $id){
$sql = 'SELECT main_category_type,main_category_weight,total_points,
main_category_name FROM main_category WHERE
gradebook_setup_details_id=' . $this->db->escape($gradebook_id) . ' AND
main_category_id=' . $this->db->escape($id);
$query = $this->db->query($sql);
$result = $query->row_array();
$cat_type = $result['main_category_type'];
$cat_weight = $result['main_category_weight'] / 100;
$cat_total_points = $result['total_points'];
$cat_name = $result['main_category_name'];
if($cat_type == "CATEGORY"){
$data = array(
'gradebook_setup_details_id' => $gradebook_id,
'column_letter' => 0,
'category_type' => 1,
'cell_type' => 2,
'grade_item_id' => 0,
'column_value' => $cat_weight,
'column_title' => "Total_(" . $result['main_category_weight'] . "%)",
'main_category_id' => $id
);
$this->db->insert('mapping',$data);
$totalcat_map_id = $this->db->insert_id();
} else if ($cat_type == "ITEM"){
$data = array(
'gradebook_setup_details_id' => $gradebook_id,
'column_letter' => 0,
'category_type' => 2,
'cell_type' => 1,
'grade_item_id' => 0,
'column_value' => $cat_total_points,
'column_title' => $cat_name,
'main_category_id' => $id
);
$this->db->insert('mapping',$data);
$subitem_map_id = $this->db->insert_id();
$data = array(
'gradebook_setup_details_id' => $gradebook_id,
'column_letter' => 0,
'category_type' => 2,
'cell_type' => 2,
'grade_item_id' => 0,
'column_value' => $cat_weight,
'column_title' => "Total_(" . $result['main_category_weight'] . "%)",
'main_category_id' => $id
);
$this->db->insert('mapping',$data);
$totalitem_map_id = $this->db->insert_id();
}
foreach($student_details as $row){
if($cat_type == "CATEGORY"){
$data = array('student_id' => $row['student_id'],
'gradebook_setup_details_id' => $gradebook_id,
'row_number' => $row['order_number'],
'mapping_id' => $totalcat_map_id,
'category_type' => 1,
'cell_type' => 2,
'cell_value' => 0,
'calculated_value' => 0,
'main_category_id' => $id
);
} else if ($cat_type == "ITEM"){
}
}
}
}
}

```

```

$data = array('student_id' => $row['student_id'],
'gradebook_setup_details_id' => $gradebook_id,
'row_number' => $row['order_number'],
'mapping_id' => $subitem_map_id,
'category_type' => 2,
'cell_type' => 1,
'cell_value' => 0,
'calculated_value' => 0,
'main_category_id' => $id
);
$this->db->insert('class_record', $data);
$data = array('student_id' => $row['student_id'],
'gradebook_setup_details_id' => $gradebook_id,
'row_number' => $row['order_number'],
'mapping_id' => $totalitem_map_id,
'category_type' => 2,
'cell_type' => 2,
'cell_value' => 0,
'calculated_value' => 0,
'main_category_id' => $id
);
$this->db->insert('class_record', $data);
}
}
} else if ($existing_cat_count > $total_category_count){
$sql = 'SELECT main_category_id FROM main_category WHERE
gradebook_setup_details_id='. $this->db->escape($gradebook_id);
$query = $this->db->query($sql);
$current_id_array = array_column($query->result_array(), "main_category_id"
);
$result = array_diff($existing_id_array, $current_id_array);
foreach($result as $id){
$this->db->where('main_category_id', $id);
$this->db->delete('class_record');
$this->db->where('main_category_id', $id);
$this->db->delete('mapping');
}
}
//-----
$sql = 'SELECT grade_item_id FROM mapping WHERE gradebook_setup_details_id
='. $this->db->escape($gradebook_id). ' AND category_type=1 AND
cell_type=1';
$query = $this->db->query($sql);
$existing_grade_item_id_array = array_column($query->result_array(), "
grade_item_id");
$existing_grade_item_count = count($existing_grade_item_id_array);
$sql = 'SELECT COUNT(*) as count_item FROM grade_item WHERE
gradebook_setup_details_id='. $this->db->escape($gradebook_id);
$query = $this->db->query($sql);
$result = $query->row_array();
$sub_item_count = $result['count_item'];
if($existing_grade_item_count < $sub_item_count){
$sql = 'SELECT grade_item_id FROM grade_item WHERE
gradebook_setup_details_id='. $this->db->escape($gradebook_id);
$query = $this->db->query($sql);
$current_grade_item_id_array = array_column($query->result_array(), "
grade_item_id");
$result = array_diff($current_grade_item_id_array,
$existing_grade_item_id_array);
foreach($result as $id){
$sql = 'SELECT main_category_id, grade_item_name, total_points FROM
grade_item WHERE grade_item_id='. $this->db->escape($id);
$query = $this->db->query($sql);
$res2 = $query->row_array();
$cat_id = $res2['main_category_id'];
$cat_title = $res2['grade_item_name'];
$cat_points = $res2['total_points'];
$data = array(
'gradebook_setup_details_id' => $gradebook_id,
'column_letter' => 0,
'category_type' => 1,
'cell_type' => 1,
'column_value' => $cat_points,
'column_title' => $cat_title,
'grade_item_id' => $id,
'main_category_id' => $cat_id
);
$this->db->insert('mapping', $data);
$gradeitem_map_id = $this->db->insert_id();
foreach($student_details as $row){
$data = array('student_id' => $row['student_id'],
'gradebook_setup_details_id' => $gradebook_id,
'row_number' => $row['order_number'],
'mapping_id' => $gradeitem_map_id,
'category_type' => 1,
'cell_type' => 1,
'cell_value' => 0,
'calculated_value' => 0,
'main_category_id' => $cat_id
);
$this->db->insert('class_record', $data);
}
} else if ($existing_grade_item_count > $sub_item_count){
$sql = 'SELECT grade_item_id FROM grade_item WHERE
gradebook_setup_details_id='. $this->db->escape($gradebook_id);
$query = $this->db->query($sql);
$current_grade_item_id_array = array_column($query->result_array(), "
main_category_id");
$result = array_diff($existing_grade_item_id_array,
$current_grade_item_id_array);
foreach($result as $id){
$sql = 'SELECT mapping_id FROM mapping WHERE grade_item_id='. $this->db->
escape($id);
$query = $this->db->query($sql);
$res = $query->row_array();
$mapping_id = $res['mapping_id'];
$this->db->where('mapping_id', $mapping_id);
$this->db->delete('class_record');
$this->db->where('grade_item_id', $id);
$this->db->delete('mapping');
}
}
$this->db->trans_start();
$this->db->query('SET @x=0');
$this->db->query('UPDATE mapping SET column_letter=@x:=@x+1 WHERE
gradebook_setup_details_id='. $this->db->escape($gradebook_id). '
ORDER BY main_category_id, category_type, cell_type');
$this->db->trans_complete();

```

```

$sql = "SELECT mapping_id,main_category_id,column_letter FROM mapping
WHERE gradebook_setup_details_id=".$this->db->escape($gradebook_id).
" AND cell_type=2";
$query = $this->db->query($sql);
$result = $query->result_array();
$sql = 'SELECT * FROM gradebook_setup_details WHERE
gradebook_setup_details_id='.$this->db->escape($gradebook_id);
$query = $this->db->query($sql);
$r2 = $query->row_array();
$method = $r2['method_of_computation'];
foreach($result as $row){
//print_r($row);
$sql = 'SELECT * FROM main_category WHERE main_category_id='.$this->db->
escape($row['main_category_id']);
$query = $this->db->query($sql);
$category = $query->row_array();
if($category['main_category_type'] === "CATEGORY"){
$sql = 'SELECT * FROM grade_item as gi INNER JOIN mapping as m ON (gi.
grade_item_id=m.grade_item_id) WHERE gi.main_category_id='.$this->db->escape($row['main_category_id']);
$query = $this->db->query($sql);
$items = $query->result_array();
$count = count($items);
$row_number = (int) $row['column_letter'];
$mapping_id = $row['mapping_id'];
if(count($items) == 0){
$data = array('cell_value' => 0);
$this->db->where('mapping_id', $row['mapping_id']);
$this->db->update('class_record', $data);
} else {
if($method === "NaturalAggregate" && $category['method_of_computation']
=== "NaturalAggregate"){
foreach($student_details as $stud){
$index = $row_number - $count;
$row = $stud['order_number'] + 2;
$cell_value = 'ROUND(';
$numerator = '';
foreach($items as $total){
$numerator .= $alphabet[$index].$row;
$numerator .= '+';
$index++;
}
$numerator = rtrim($numerator, "+");
$cell_value .= $numerator.'';
$cell_value .= ',2)';
$data = array('cell_value' => $cell_value);
$this->db->where('student_id', $stud['student_id']);
$this->db->where('gradebook_setup_details_id', $gradebook_id);
$this->db->where('mapping_id', $mapping_id);
$this->db->update('class_record', $data);
}
} else {
foreach($student_details as $stud){
$index = $row_number - $count;
$row = $stud['order_number'] + 2;
if($category['method_of_computation'] === "NaturalAggregate"){
$cell_value = 'ROUND(';
$numerator = '';
$denominator = '';
} else if ($category['method_of_computation'] === "Mean_of_Grades" &&
$method === "NaturalAggregate"){
$cell_value = 'ROUND(';
$numerator = '';
$denominator = '';
} else if ($category['method_of_computation'] === "Weighted_Mean_of_Grades"
&& $method === "NaturalAggregate"){
$cell_value = 'ROUND(';
$numerator = '';
$denominator = '';
$grade_weight = 0;
} else if ($category['method_of_computation'] === "Weighted_Mean_of_Grades"
){
$cell_value = 'ROUND(';
$numerator = '';
$denominator = '';
$grade_weight = 0;
} else if ($method === "NaturalAggregate"){
$cell_value = 'ROUND(';
$numerator = '';
}
foreach($items as $total){
if($category['method_of_computation'] === "NaturalAggregate"){
$numerator .= $alphabet[$index].$row;
$numerator .= '+';
$denominator .= $alphabet[$index].'^2';
$index++;
} else if ($category['method_of_computation'] === "Mean_of_Grades" &&
$method === "NaturalAggregate"){
$numerator .= '('. $alphabet[$index].$row.'/'.'$total['column_value'].')';
$numerator .= '+';
$index++;
} else if ($category['method_of_computation'] === "Mean_of_Grades"){
$numerator .= '('. $alphabet[$index].$row.'/'.'$total['column_value'].')';
$numerator .= '+';
$index++;
} else if ($category['method_of_computation'] === "Weighted_Mean_of_Grades"
&& $method === "NaturalAggregate"){
$grade_weight += $total['grade_item_weight'];
$numerator .= '('. $alphabet[$index].$row.'/'.'$total['column_value'].')'*'.
$total['grade_item_weight'].')';
$numerator .= '+';
$index++;
} else if ($category['method_of_computation'] === "Weighted_Mean_of_Grades"
){
$grade_weight += $total['grade_item_weight'];
$numerator .= '('. $alphabet[$index].$row.'/'.'$total['column_value'].')'*'.
$total['grade_item_weight'].')';
$numerator .= '+';
$index++;
} else if ($method === "NaturalAggregate"){
$numerator .= $alphabet[$index].$row;
$numerator .= '+';
$index++;
}
}
}
}

```

```

$enumerator = rtrim($enumerator, "+");
$denominator = rtrim($denominator, "+");
if($category['method_of_computation'] === "NaturalAggregate"){
$cell_value .= $enumerator.'/'./.$denominator.'')*(.'.$category['
    main_category_weight'.')')';
} else if ($category['method_of_computation'] === "MeanofGrades" &&
    $method === "NaturalAggregate"){
$cell_value .= $enumerator.'/'./.$count.'')*(.'.$category['total_points'.')')';
} else if ($category['method_of_computation'] === "MeanofGrades"){
$cell_value .= $enumerator.'/'./.$count.'')*(.'.$category['main_category_weight
    '.')')';
} else if ($category['method_of_computation'] === "WeightedMeanofGrades
    " && $method === "NaturalAggregate"){
$cell_value .= $enumerator.'/'./.$grade_weight.'')*(.'.$category['total_points'
    '.')')';
} else if ($category['method_of_computation'] === "WeightedMeanofGrades
    "){
$cell_value .= $enumerator.'/'./.$grade_weight.'')*(.'.$category['
    main_category_weight'.')')';
} else if ($method === "NaturalAggregate"){
$cell_value .= $enumerator.'')';
}
}
$cell_value .= ',2)';
$data = array('cell_value' => $cell_value);
$this->db->where('student_id', $stud['student_id']);
$this->db->where('gradebook_setup_details_id', $gradebook_id);
$this->db->where('mapping_id', $mapping_id);
$this->db->update('class_record', $data);
}
}
}
} else {
$sql = 'SELECT*FROMmain_categoryasmcINNERJOINmappingasmON(mc.
    main_category_id=m.main_category_id)WHEREmc.main_category_id='.
    $this->db->escape($row['main_category_id']).'ANDcell_type=2';
$query = $this->db->query($sql);
$item_cat = $query->row_array();
$row_number = (int) $row['column_letter'];
$mapping_id = $row['mapping_id'];
foreach($student_details as $stud){
$index = $row_number - 1;
$row = $stud['order_number'] + 2;
if($method === "NaturalAggregate"){
$cell_value = "ROUND(";
$cell_value .= $alphabet[$index].$row;
} else {
$cell_value = "ROUND((((";
$cell_value .= $alphabet[$index].$row.'/'./.$item_cat['total_points'.')')*(.
    $item_cat['main_category_weight'.')')';
}
}
$cell_value .= ',2)';
$data = array('cell_value' => $cell_value);
$this->db->where('student_id', $stud['student_id']);
$this->db->where('gradebook_setup_details_id', $gradebook_id);
$this->db->where('mapping_id', $mapping_id);
$this->db->update('class_record', $data);
}
}
}
}

$sql = 'SELECTm.column_letter,mc.main_category_weightFROMmappingasm
    INNERJOINmain_categoryasmcON(m.main_category_id=mc.
    main_category_id)WHEREmc.gradebook_setup_details_id='.$this->db->
    escape($gradebook_id).'ANDcell_type=2';
$q = $this->db->query($sql);
$res = $q->result_array();
$count = count($res);
$sql = 'SELECT*FROMgradebook_setup_detailsWHERE
    gradebook_setup_details_id='.$this->db->escape($gradebook_id);
$query = $this->db->query($sql);
$gradebook = $query->row_array();
$sql = 'SELECTgrading_scale_idFROMgrading_scaleWHEREcourse_id='.
    $this->db->escape($course_id);
$query2 = $this->db->query($sql);
$result2 = $query2->row_array();
$grading_scale_id = $result2['grading_scale_id'];
$sql = 'SELECT*FROMscale_valuesWHEREgrading_scale_id='.$this->db->
    escape($grading_scale_id).'ORDERBYscale_values_higher_limitDESC';
$query3 = $this->db->query($sql);
$result3 = $query3->result_array();
$scale_value_count = count($result3);
$sql = 'SELECTCOUNT(*)asmap_countFROMmappingWHERE
    gradebook_setup_details_id='.$this->db->escape($gradebook_id);
$query2 = $this->db->query($sql);
$result2 = $query2->row_array();
$mapping_count = $result2['map_count'];
$total_column = $alphabet[$mapping_count + 1];
foreach($student_details as $stud){
$cat_weight = 0;
$row = $stud['order_number'] + 2;
if ($gradebook['method_of_computation'] === "NaturalAggregate"){
$cell_value = 'ROUND((((';
} else {
$cell_value = 'ROUND(';
}
$enumerator = '';
$calculated_value = '';
foreach($res as $ids){
$index = $ids['column_letter'];
if ($gradebook['method_of_computation'] === "MeanofGrades"){
$enumerator .= '('.$alphabet[$index].$row.'')';
$enumerator .= '+';
} else if ($gradebook['method_of_computation'] === "WeightedMeanof
    Grades"){
$cat_weight += $ids['main_category_weight'];
$enumerator .= '('.$alphabet[$index].$row.'')';
$enumerator .= '+';
} else if ($gradebook['method_of_computation'] === "NaturalAggregate"){
$enumerator .= '('.$alphabet[$index].$row.'')';
$enumerator .= '+';
}
}
$enumerator = rtrim($enumerator, "+");
//if ($gradebook['method_of_computation'] === "Mean of Grades"){
$cell_value .= $enumerator;
//} else if ($gradebook['method_of_computation'] === "Weighted Mean of
    Grades"){
//    $cell_value .= $enumerator.'/'./.$cat_weight.'')*(.'.$gradebook['
        max_grade'.')')';
//}

```



```

if ($gradebook['method_of_computation'] == "NaturalAggregate"){
$cell_value .= ')/'. $gradebook['total_points'].')*100';
}
$cell_value .= ',2)';
$calculated_value = '=';
for($a = 0; $a < $scale_value_count - 1; $a++){
$calculated_value .= 'IF('.$total_column.$row.>'>'. $result3[$a]['
scale_values_lower_limit'].',".'. $result3[$a]['scale_values_label'].',
',';
}
$calculated_value .= "'". $result3[$scale_value_count - 1]['
scale_values_label']."'";
for($a = 0; $a < $scale_value_count - 1; $a++){
$calculated_value .= ')';
}
$data = array('cell_value' => $cell_value, 'calculated_value' =>
$calculated_value);
$this->db->where('student_id', $stud['student_id']);
$this->db->where('course_id', $course_id);
$this->db->update('final_grade', $data);
}
//echo '<pre>';
//print_r($student_details);
//echo '</pre>';
}

public function update_students_in_class_record($course_id){
$sql = 'SELECT gradebook_setup_details_id FROM gradebook_setup_details
WHERE course_id=' . $this->db->escape($course_id);
$query = $this->db->query($sql);
$result = $query->row_array();
$gradebook_id = $result['gradebook_setup_details_id'];
$alphabet = array('B','C','D','E','F','G','H','I','J','K','L','M','N','O',
'P','Q','R','S','T','U','V','W','X','Y','Z','AA','AB','AC','AD','AE',
'AF','AG','AH','AJ','AK','AL','AM','AN','AO','AP','AQ','AR','AS',
'AT','AU','AV','AW','AX','AY','AZ');
$sql = 'SELECT student_id, order_number FROM course_students WHERE
course_id=' . $this->db->escape($course_id). ' ORDER BY order_number';
$query = $this->db->query($sql);
$student_details = $query->result_array();
$student_id_array = array_column($student_details, "student_id");
$sql2 = 'SELECT DISTINCT (student_id) FROM class_record WHERE
gradebook_setup_details_id=' . $this->db->escape($gradebook_id). '
ORDER BY student_id';
$query = $this->db->query($sql2);
$record_students = $query->result_array();
$record_stud_id_array = array_column($record_students, "student_id");
if(count($record_stud_id_array) == 0){
$this->update_class_record($gradebook_id);
return;
}
$difference = array_diff($student_id_array, $record_stud_id_array);
if (count($difference) > 0) {
$this->db->select('student_id, order_number');
$this->db->from('course_students');
$this->db->where('course_id', $course_id);
$this->db->where_in('student_id', $difference);
$this->db->order_by('order_number');
$query = $this->db->get();
$stud_details = $query->result_array();
// -----
$sql = 'SELECT grade_item_id, mapping_id FROM mapping WHERE
gradebook_setup_details_id=' . $this->db->escape($gradebook_id). ' AND
(category_type=1 AND cell_type=1)';
$query = $this->db->query($sql);
$result = $query->result_array();
foreach($result as $row){
$sql = 'SELECT main_category_id, grade_item_name, total_points FROM
grade_item WHERE grade_item_id=' . $this->db->escape($row['
grade_item_id']);
$query = $this->db->query($sql);
$res2 = $query->row_array();
$cat_id = $res2['main_category_id'];
$cat_title = $res2['grade_item_name'];
$cat_points = $res2['total_points'];
$gradeitem_map_id = $row['mapping_id'];
foreach($stud_details as $row2){
$data = array('student_id' => $row2['student_id'],
'gradebook_setup_details_id' => $gradebook_id,
'row_number' => $row2['order_number'],
'mapping_id' => $gradeitem_map_id,
'category_type' => 1,
'cell_type' => 1,
'cell_value' => 0,
'calculated_value' => 0,
'main_category_id' => $cat_id
);
$this->db->insert('class_record', $data);
}
// -----
$sql = 'SELECT DISTINCT main_category_id FROM mapping WHERE
gradebook_setup_details_id=' . $this->db->escape($gradebook_id). ' AND
NOT (cell_type=1 AND category_type=1)';
$query = $this->db->query($sql);
$result = $query->result_array();
$main_cat_id = array_column($result, 'main_category_id');
$this->db->select('*');
$this->db->from('mapping');
$this->db->where_in('main_category_id', $main_cat_id);
$query = $this->db->get();
$r = $query->result_array();
foreach($stud_details as $row){
foreach($r as $row2){
$sql = 'SELECT main_category_type, main_category_weight, total_points,
main_category_name FROM main_category WHERE
gradebook_setup_details_id=' . $this->db->escape($gradebook_id). ' AND
main_category_id=' . $this->db->escape($row2['main_category_id']);
$query = $this->db->query($sql);
$result = $query->row_array();
$cat_type = $result['main_category_type'];
$cat_weight = $result['main_category_weight'] / 100;
$cat_total_points = $result['total_points'];
$cat_name = $result['main_category_name'];
if($cat_type == "CATEGORY"){
if($row2['cell_type'] == 2 && $row2['category_type'] == 1){
$data = array('student_id' => $row['student_id'],
'gradebook_setup_details_id' => $gradebook_id,
'row_number' => $row['order_number'],
'mapping_id' => $row2['mapping_id'],
'category_type' => 1,

```

```

'cell_type' => 2,
'cell_value' => 0,
'calculated_value' => 0,
'main_category_id' => $row2['main_category_id']
);
$this->db->insert('class_record', $data);
}
} else if ($cat_type === "ITEM"){
if($row2['cell_type'] == 1 && $row2['category_type'] == 2){
$data = array('student_id' => $row['student_id'],
'gradebook_setup_details_id' => $gradebook_id,
'row_number' => $row['order_number'],
'mapping_id' => $row2['mapping_id'],
'category_type' => 2,
'cell_type' => 1,
'cell_value' => 0,
'calculated_value' => 0,
'main_category_id' => $row2['main_category_id']
);
$this->db->insert('class_record', $data);
}
if($row2['cell_type'] == 2 && $row2['category_type'] == 2){
$data2 = array('student_id' => $row['student_id'],
'gradebook_setup_details_id' => $gradebook_id,
'row_number' => $row['order_number'],
'mapping_id' => $row2['mapping_id'],
'category_type' => 2,
'cell_type' => 2,
'cell_value' => 0,
'calculated_value' => 0,
'main_category_id' => $row2['main_category_id']
);
$this->db->insert('class_record', $data2);
}
}
}
}
}
// -----
$sql = "SELECT mapping_id, main_category_id, column_letter FROM mapping
WHERE gradebook_setup_details_id = ".$this->db->escape($gradebook_id).
" AND cell_type = 2";
$query = $this->db->query($sql);
$result = $query->result_array();
$sql = 'SELECT * FROM gradebook_setup_details WHERE
gradebook_setup_details_id = '.$this->db->escape($gradebook_id);
$query = $this->db->query($sql);
$r2 = $query->row_array();
$method = $r2['method_of_computation'];
foreach($result as $row){
//print_r($row);
$sql = 'SELECT * FROM main_category WHERE main_category_id = '.$this->db->
escape($row['main_category_id']);
$query = $this->db->query($sql);
$category = $query->row_array();
if($category['main_category_type'] === "CATEGORY"){
$sql = 'SELECT * FROM grade_item as gi INNER JOIN mapping as m ON (gi.
grade_item_id = m.grade_item_id) WHERE gi.main_category_id = '.$this->
db->escape($row['main_category_id']);
$query = $this->db->query($sql);
}

$items = $query->result_array();
$count = count($items);
$row_number = (int) $row['column_letter'];
$mapping_id = $row['mapping_id'];
if(count($items) == 0){
$data = array('cell_value' => 0);
$this->db->where('mapping_id', $row['mapping_id']);
$this->db->update('class_record', $data);
} else {
if($method === "NaturalAggregate"){
foreach($student_details as $stud){
$index = $row_number - $count;
$row = $stud['order_number'] + 2;
$cell_value = 'ROUND(';
$numerator = '';
foreach($items as $total){
$numerator .= $alphabet[$index].$row;
$numerator .= '+';
$index++;
}
$numerator = rtrim($numerator, "+");
$cell_value .= $numerator.'';
$cell_value .= ', 2)';
$data = array('cell_value' => $cell_value);
$this->db->where('student_id', $stud['student_id']);
$this->db->where('gradebook_setup_details_id', $gradebook_id);
$this->db->where('mapping_id', $mapping_id);
$this->db->update('class_record', $data);
}
} else {
foreach($student_details as $stud){
$index = $row_number - $count;
$row = $stud['order_number'] + 2;
if($category['method_of_computation'] === "NaturalAggregate"){
$cell_value = 'ROUND((((';
$numerator = '';
$denominator = '';
} else if ($category['method_of_computation'] === "Mean_of_Grades"){
$cell_value = 'ROUND((((';
$numerator = '';
$denominator = '';
} else if ($category['method_of_computation'] === "Weighted_Mean_of_Grades
"){
$cell_value = 'ROUND((((';
$numerator = '';
$denominator = '';
$grade_weight = 0;
}
foreach($items as $total){
if($category['method_of_computation'] === "NaturalAggregate"){
$numerator .= $alphabet[$index].$row;
$numerator .= '+';
$denominator .= $alphabet[$index].'^2+';
$index++;
} else if ($category['method_of_computation'] === "Mean_of_Grades"){
$numerator .= '('. $alphabet[$index].$row.' / '. $total['column_value']. ')';
$numerator .= '+';
$index++;
} else if ($category['method_of_computation'] === "Weighted_Mean_of_Grades
"){

```

```

$grade_weight += $total['grade_item_weight'];
$enumerator .= '('. $alphabet[$index]. $row.'/'. $total['column_value'].')*'.
    $total['grade_item_weight'].')';
$enumerator .= '+';
$index++;
}
}
$enumerator = rtrim($enumerator, "+");
$denominator = rtrim($denominator, "+");
if($category['method_of_computation'] === "NaturalAggregate"){
$cell_value .= $enumerator.'/'.'$denominator.'*('.$category['
    main_category_weight'].')';
} else if ($category['method_of_computation'] === "Mean_of_Grades"){
$cell_value .= $enumerator.'/'.'$count.'*'. $category['main_category_weight
    '].')';
} else if ($category['method_of_computation'] === "Weighted_Mean_of_Grades
    "){
$cell_value .= $enumerator.'/'.'$grade_weight.'*'. $category['
    main_category_weight'].')';
}
$cell_value .= ',2)';
$data = array('cell_value' => $cell_value);
$this->db->where('student_id', $stud['student_id']);
$this->db->where('gradebook_setup_details_id', $gradebook_id);
$this->db->where('mapping_id', $mapping_id);
$this->db->update('class_record', $data);
}
}
} else {
$sql = 'SELECT * FROM main_category AS mc INNER JOIN mapping AS m ON (mc.
    main_category_id = m.main_category_id) WHERE mc.main_category_id = ' .
    $this->db->escape($row['main_category_id']). ' AND cell_type=2';
$query = $this->db->query($sql);
$item_cat = $query->row_array();
$row_number = (int) $row['column_letter'];
$mapping_id = $row['mapping_id'];
foreach($student_details as $stud){
$index = $row_number - 1;
$row = $stud['order_number'] + 2;
if($method === "NaturalAggregate"){
$cell_value = "ROUND(";
$cell_value .= $alphabet[$index]. $row;
} else {
$cell_value = "ROUND((((";
$cell_value .= $alphabet[$index]. $row.'/'.'$item_cat['total_points'].')*'.
    $item_cat['main_category_weight'].')';
}
$cell_value .= ',2)';
$data = array('cell_value' => $cell_value);
$this->db->where('student_id', $stud['student_id']);
$this->db->where('gradebook_setup_details_id', $gradebook_id);
$this->db->where('mapping_id', $mapping_id);
$this->db->update('class_record', $data);
}
}
}
$sql = 'SELECT m.column_letter, mc.main_category_weight FROM mapping AS m
    INNER JOIN main_category AS mc ON (m.main_category_id = mc.
    main_category_id) WHERE mc.gradebook_setup_details_id = ' . $this->db->
    escape($gradebook_id). ' AND cell_type=2';
$query = $this->db->query($sql);
$res = $query->result_array();
$count = count($res);
$sql = 'SELECT * FROM gradebook_setup_details WHERE
    gradebook_setup_details_id = ' . $this->db->escape($gradebook_id);
$query = $this->db->query($sql);
$gradebook = $query->row_array();
$sql = 'SELECT grading_scale_id FROM grading_scale WHERE course_id = ' .
    $this->db->escape($course_id);
$query2 = $this->db->query($sql);
$result2 = $query2->row_array();
$grading_scale_id = $result2['grading_scale_id'];
$sql = 'SELECT * FROM scale_values WHERE grading_scale_id = ' . $this->db->
    escape($grading_scale_id). ' ORDER BY scale_values_lower_limit DESC';
$query3 = $this->db->query($sql);
$result3 = $query3->result_array();
$scale_value_count = count($result3);
$sql = 'SELECT COUNT(*) AS map_count FROM mapping WHERE
    gradebook_setup_details_id = ' . $this->db->escape($gradebook_id);
$query2 = $this->db->query($sql);
$result2 = $query2->row_array();
$mapping_count = $result2['map_count'];
$total_column = $alphabet[$mapping_count + 1];
foreach($student_details as $stud){
$cat_weight = 0;
$row = $stud['order_number'] + 2;
if ($gradebook['method_of_computation'] === "NaturalAggregate"){
$cell_value = "ROUND((((";
} else {
$cell_value = "ROUND(";
}
$enumerator = '';
$calculated_value = '';
foreach($res as $ids){
$index = $ids['column_letter'];
if ($gradebook['method_of_computation'] === "Mean_of_Grades"){
$enumerator .= '('. $alphabet[$index]. $row.')' . $gradebook['method_of_computation'] === "Weighted_Mean_of_Grades"
    Grades){
$cat_weight += $ids['main_category_weight'];
$enumerator .= '('. $alphabet[$index]. $row.')' . $gradebook['method_of_computation'] === "NaturalAggregate"
    NaturalAggregate){
$enumerator .= '('. $alphabet[$index]. $row.')' . $gradebook['method_of_computation'] === "Weighted_Mean_of_Grades"
    Weighted_Mean_of_Grades){
$cell_value .= $enumerator.'/'.'$cat_weight.'*'. $gradebook['
    max_grade'].')';
}
}
$enumerator = rtrim($enumerator, "+");
//if ($gradebook['method_of_computation'] === "Mean of Grades"){
$cell_value .= $enumerator;
//} else if ($gradebook['method_of_computation'] === "Weighted Mean of
    Grades"){
//    $cell_value .= $enumerator.'/'.'$cat_weight.'*'. $gradebook['
        max_grade'].')';
//}
if ($gradebook['method_of_computation'] === "NaturalAggregate"){
$cell_value .= ')'.'$gradebook['total_points'].')*100)';
}
}
}

```

```

$cell_value .= ',2)';
$calculated_value = '';
for($a = 0; $a < $scale_value_count - 1; $a++){
$calculated_value .= 'IF(('$total_column.$row.'>='.$result3[$a]['
    scale_values_lower_limit'].','.$result3[$a]['scale_values_label'].
    ',';
}
$calculated_value .= ''.$result3[$scale_value_count - 1]['
    scale_values_label'].'';
for($a = 0; $a < $scale_value_count - 1; $a++){
$calculated_value .= ''';
}
$data = array('cell_value' => $cell_value, 'calculated_value' =>
    $calculated_value);
$this->db->where('student_id', $stud['student_id']);
$this->db->where('course_id', $course_id);
$this->db->update('final_grade', $data);
}
//echo '<pre>';
//print_r($student_details);
//echo '</pre>';
}
public function update_mapping($gradebook_id){
$sql = 'SELECT * FROM mapping WHERE gradebook_setup_details_id=' . $this->db
    ->escape($gradebook_id);
$query = $this->db->query($sql);
$result = $query->result_array();
$sql = 'SELECT * FROM gradebook_setup_details WHERE
    gradebook_setup_details_id=' . $this->db->escape($gradebook_id);
$query = $this->db->query($sql);
$result2 = $query->row_array();
$method = $result2['method_of_computation'];
foreach($result as $row){
if($row['cell_type'] == 1 && $row['category_type'] == 1){
$sql = 'SELECT * FROM grade_item WHERE grade_item_id=' . $this->db->escape(
    $row['grade_item_id']);
$query = $this->db->query($sql);
$result2 = $query->row_array();
$data = array('column_title' => $result2['grade_item_name'], 'column_value
    ' => $result2['total_points']);
$this->db->where('grade_item_id', $row['grade_item_id']);
$this->db->update('mapping', $data);
} else if ($row['cell_type'] == 1 && $row['category_type'] == 2){
$sql = 'SELECT * FROM main_category WHERE main_category_id=' . $this->db->
    escape($row['main_category_id']);
$query = $this->db->query($sql);
$result2 = $query->row_array();
$data = array('column_title' => $result2['main_category_name'], '
    column_value' => $result2['total_points']);
$this->db->where('main_category_id', $row['main_category_id']);
$this->db->where('cell_type', 1);
$this->db->where('category_type', 2);
$this->db->update('mapping', $data);
} else if ($row['cell_type'] == 2 && $row['category_type'] == 2){
$sql = 'SELECT * FROM main_category WHERE main_category_id=' . $this->db->
    escape($row['main_category_id']);
$query = $this->db->query($sql);
$result2 = $query->row_array();
if($method === "NaturalAggregate"){
$data = array('column_title' => 'Total('.$result2['total_points'].'.pts)',
    'column_value' => NULL);
} else {
$data = array('column_title' => 'Total('.$result2['main_category_weight
    ].'%)', 'column_value' => NULL);
}
$this->db->where('main_category_id', $row['main_category_id']);
$this->db->where('cell_type', 2);
$this->db->where('category_type', 2);
$this->db->update('mapping', $data);
} else if ($row['cell_type'] == 2 && $row['category_type'] == 1){
$sql = 'SELECT * FROM main_category WHERE main_category_id=' . $this->db->
    escape($row['main_category_id']);
$query = $this->db->query($sql);
$result2 = $query->row_array();
if($method === "NaturalAggregate"){
$data = array('column_title' => 'Total('.$result2['total_points'].'.pts)',
    'column_value' => NULL);
} else {
$data = array('column_title' => 'Total('.$result2['main_category_weight
    ].'%)', 'column_value' => NULL);
}
$this->db->where('main_category_id', $row['main_category_id']);
$this->db->where('cell_type', 2);
$this->db->where('category_type', 1);
$this->db->update('mapping', $data);
}
}
public function get_category_total_mapping($course_id){
$sql = "SELECT gradebook_setup_details_id FROM gradebook_setup_details
    WHERE course_id=" . $this->db->escape($course_id);
$query = $this->db->query($sql);
$res = $query->row_array();
$gradebook_id = $res['gradebook_setup_details_id'];
$sql = "SELECT column_letter FROM mapping WHERE gradebook_setup_details_id
    =" . $this->db->escape($gradebook_id) . " AND cell_type=2";
$query = $this->db->query($sql);
$res2 = $query->result_array();
//return array_column($res2, "column_letter");
return array_map(function($val){ return $val+1; }, array_map('intval',
    array_column($res2, "column_letter")));
}
public function get_dropped_students($course_id){
$sql = "SELECT order_number FROM course_students WHERE course_id=" . $this->
    db->escape($course_id) . " AND is_dropped=" . "YES";
$query = $this->db->query($sql);
$res2 = $query->result_array();
//return array_column($res2, "column_letter");
return array_map(function($val){ return $val+1; }, array_map('intval',
    array_column($res2, "order_number")));
}
public function get_trimmed_indices($course_id, $student_id){
$sql = 'SELECT order_number FROM course_students WHERE course_id=' . $this->
    db->escape($course_id) . " AND student_id!=" . $this->db->escape(
    $student_id);
$q = $this->db->query($sql);
$res = $q->result_array();
$array = array_column($res, 'order_number');
foreach ($array as &$row){

```

```

$row = $row + 1;
}
return $array;
}

public function save_class_record ($data, $course_id){
    $sql = "SELECT gradebook_setup_details_id FROM gradebook_setup_details
        WHERE course_id=". $this->db->escape($course_id);
    $query = $this->db->query($sql);
    $res = $query->row_array();
    $gradebook_id = $res['gradebook_setup_details_id'];
    $sql = "SELECT column_letter, mapping_id FROM mapping WHERE
        gradebook_setup_details_id=". $this->db->escape($gradebook_id);
    $query2 = $this->db->query($sql);
    $result2 = $query2->result_array();
    $count = count($result2);
    $alphabet = range('B','Z');
    foreach($data as $row){
        for($x = 0 ; $x < $count; $x++){
            if(isset($row[$alphabet[$x]])){
                $key = array_search($x, array_column($result2, 'column_letter'));
                $mapping_id = $result2[$key]['mapping_id'];
                $data = array('cell_value' => $row[$alphabet[$x]]);
                $this->db->where('student_id', $row['student_id']);
                $this->db->where('mapping_id', $mapping_id);
                $this->db->update('class_record', $data);
            }
        }
    }
    $data2 = array('cell_value' => $row['final_grade'],
        'calculated_value' => $row['scaled_grade']
    );
    $this->db->where('student_id', $row['student_id']);
    $this->db->where('course_id', $course_id);
    $this->db->update('final_grade', $data2);
}

public function get_category_stats_names($course_id){
    $sql = "SELECT gradebook_setup_details_id FROM gradebook_setup_details
        WHERE course_id=". $this->db->escape($course_id);
    $query = $this->db->query($sql);
    $res = $query->row_array();
    $gradebook_id = $res['gradebook_setup_details_id'];
    $sql2 = "SELECT * FROM mapping AS m INNER JOIN main_category AS mc ON (mc.
        main_category_id = m.main_category_id) WHERE m.
        gradebook_setup_details_id=". $this->db->escape($gradebook_id). "
        ORDER BY column_letter";
    $query2 = $this->db->query($sql2);
    $result = $query2->result_array();
    return $result;
}

public function get_enrolled_student_list($course_id){
    $sql2 = "SELECT order_number FROM course_students WHERE is_dropped='NO'
        AND course_id=". $this->db->escape($course_id);
    $query2 = $this->db->query($sql2);
    $result = $query2->result_array();
    return $result;
}

public function get_dropped_student_list($course_id){
    $sql2 = "SELECT * FROM course_students WHERE is_dropped='YES' AND
        course_id=". $this->db->escape($course_id);
    $query2 = $this->db->query($sql2);

```

```

$result = $query2->result_array();
$result = array_map('intval', array_column($result, 'order_number'));
for($a = 0 ; $a < count($result); $a++){
    $result[$a] += 2;
}
return $result;
}
}

```

Student Model

```

<?php
if (!defined('BASEPATH')) exit('No direct script access allowed');
class Student_model extends CI_Model {
    public function __construct(){
        $this->load->database();
        $this->load->helper('url_helper');
        $this->load->library('session');
    }
    public function add_student(){
        $max = $this->get_last_student_index();
        if($max['latest_id'] > 0){
            $array = $this->generate_student_number($max['latest_id'] + 1, 1);
        } else {
            $array = $this->generate_student_number(1,1);
        }
        $salt = uniqid(mt_rand(), true);
        $password = $salt.$array[0];
        $data2 = array(
            'user_name' => $array[0],
            'school_id' => $_SESSION['logged_in']['school_id'],
            'user_password' => hash('sha256', $password),
            'user_password_salt' => $salt,
            'user_role' => "STUDENT"
        );
        $this->db->insert('user', $data2);
        $this->db->select('*');
        $this->db->from('user');
        $this->db->where('user_name', $array[0]);
        $query = $this->db->get();
        $user_id = $query->row_array()['user_id'];
        $data = array(
            'user_id' => $user_id,
            'user_first_name' => $this->input->post('student_first_name'),
            'user_last_name' => $this->input->post('student_last_name'),
            'user_email_address' => $this->input->post('student_email_address'),
            'email_verified' => "NO"
        );
        $this->db->insert('user_details', $data);
        $data3 = array(
            'user_id' => $user_id,
            'student_number' => $array[0],
            'school_id' => $_SESSION['logged_in']['school_id'],
            'first_login' => "YES"
        );
        $this->db->insert('student', $data3);
        $this->db->select('*');
        $this->db->from('student');
        $this->db->where('student_number', $array[0]);
        $query = $this->db->get();

```

```

$student_id = $query->row_array()['student_id'];
$array2 = $this->input->post('available_courses');
if(count($array2) > 0){
foreach($array2 as $row){
$data = array(
'student_id' => $student_id,
'course_id' => $row,
'is_dropped' => "NO"
);
$this->db->insert('course_students',$data);
}
$data2 = array(
'student_id' => $student_id,
'course_id' => $row,
'cell_value' => '',
'calculated_value' => 0
);
$this->db->insert('final_grade',$data2);
$this->course_model->update_order_in_course($row);
$this->grade_model->update_students_in_class_record($row);
}
}

public function add_multiple_student(){
$first_name = $this->input->post('first_name');
$last_name = $this->input->post('last_name');
$email_address = $this->input->post('email_address');
$array_set = array();
$array_first_names = array();
$array_last_names = array();
$array_email_dups = array();
$count = 0;
$array_emails = array();
$size = count($first_name);
for($index = 0; $index < $size; $index++){
if($this->user_model->check_email_address($email_address[$index])){
$array_first_names[$count] = $first_name[$index];
$array_last_names[$count] = $last_name[$index];
$array_email_dups[$count] = $email_address[$index];
$count++;
} else {
$max = $this->get_last_student_index();
if($max['latest_id'] > 0){
$array = $this->generate_student_number($max['latest_id'] + 1, 1);
} else {
$array = $this->generate_student_number(1,1);
}
$salt = uniqid(mt_rand(), true);
$password = $salt.$array[0];
$data2 = array(
'user_name' => $array[0],
'school_id' => $_SESSION['logged_in']['school_id'],
'user_password' => hash('sha256',$password),
'user_password_salt' => $salt,
'user_role' => "STUDENT"
);
$this->db->insert('user',$data2);
$this->db->select('*');
$this->db->from('user');
$this->db->where('user_name', $array[0]);
$query = $this->db->get();
}

$user_id = $query->row_array()['user_id'];
$data = array(
'user_id' => $user_id,
'user_first_name' => $first_name[$index],
'user_last_name' => $last_name[$index],
'user_email_address' => $email_address[$index],
'email_verified' => "NO"
);
$this->db->insert('user_details',$data);
array_push($array_emails, $email_address[$index]);
$data3 = array(
'user_id' => $user_id,
'student_number' => $array[0],
'first_login' => "YES",
'school_id' => $_SESSION['logged_in']['school_id']
);
$this->db->insert('student',$data3);
}
}

array_push($array_set, $array_emails);
array_push($array_set, $array_first_names);
array_push($array_set, $array_last_names);
array_push($array_set, $array_email_dups);
return $array_set;
}

function get_last_student_index(){
$this->db->select_max('student_id','latest_id');
$this->db->from('student');
$query = $this->db->get();
return $query->row_array();
}

public function generate_student_number($start, $count){
$data = array();
for($i = 0; $i < $count; $i++){
$data[$i] = date("Y").str_pad($start,5,0,STR_PAD_LEFT);
$start++;
}
return $data;
}

public function get_list_of_students(){
$this->db->select('*');
$this->db->from('student_asus');
$this->db->where('s.school_id',$_SESSION['logged_in']['school_id']);
$this->db->join('user_details_asud','s.user_id=ud.user_id','left');
$query = $this->db->get();
return $query->result_array();
}

public function get_student_groups(){
$this->db->select('*');
$this->db->from('student_group_asug');
$this->db->where('sg.school_id',$_SESSION['logged_in']['school_id']);
$query = $this->db->get();
return $query->result_array();
}

public function check_student_group_name($group_name){
$this->db->select('*');
$this->db->from('student_group');
$this->db->where('student_group_name',$group_name);
$query = $this->db->get();
if($query->num_rows() == 1)

```

```

{
return true;
}
else
{
return false;
}
}

public function check_update_student_group_name($group_name,
    $student_group_id){
    $this->db->select('*');
    $this->db->from('student_group');
    $this->db->where('student_group_name', $group_name);
    $this->db->where('student_group_id!=' , $student_group_id);
    $query = $this->db->get();
    if($query -> num_rows() == 1)
    {
return true;
}
else
{
return false;
}
}

public function add_student_group($group_name){
    $size = count($this->input->post('group_members'));
    $data = array(
    'school_id' => $_SESSION['logged_in']['school_id'],
    'student_group_name' => $group_name,
    'student_members_count' => $size
);
    $this->db->insert('student_group', $data);
    $this->db->select('*');
    $this->db->from('student_group');
    $this->db->where('student_group_name', $group_name);
    $query = $this->db->get();
    $array = $query->row_array();
    $group_id = $array['student_group_id'];
    $size = count($this->input->post('group_members'));
    if($size > 0){
    $student_id = $this->input->post('group_members');
    for($a = 0; $a < $size; $a++){
    $data2 = array(
    'student_group_id' => $group_id,
    'student_id' => $student_id[$a]
);
    $this->db->insert('student_group_members', $data2);
}
}
}

public function get_details_by_id($student_id){
    $this->db->select('*');
    $this->db->from('student_id as s');
    $this->db->where('s.student_id', $student_id);
    $this->db->where('s.school_id', $_SESSION['logged_in']['school_id']);
    $this->db->join('user as u', 's.user_id = u.user_id', 'left');
    $this->db->join('user_details as ud', 'u.user_id = ud.user_id', 'left');
    $query = $this->db->get();
    return $query->row_array();
}
}

public function get_assigned_courses($student_id){
    $this->db->select('*');
    $this->db->from('course_students');
    $this->db->where('student_id', $student_id);
    $query = $this->db->get();
    $result = $query->result_array();
    $array = array();
    $index = 0;
    foreach($result as $row){
    $array[$index] = $row['course_id'];
    $index++;
}
}

return $array;
}

public function update_student($student_id, $array_courses) {
    $this->db->select('*');
    $this->db->from('student');
    $this->db->where('student_id', $student_id);
    $query = $this->db->get();
    $array = $query->row_array();
    $user_id = $array['user_id'];
    $data = array(
    'user_last_name' => $this->input->post('student_last_name'),
    'user_first_name' => $this->input->post('student_first_name'),
    'user_email_address' => $this->input->post('student_email_address')
);
    $this->db->where('user_id', $user_id);
    $this->db->update('user_details', $data);
    $array = $this->input->post('available_courses');
    $new_array_size = count($array);
    $old_array_size = count($array_courses);
    if($new_array_size > $old_array_size){
    $result_array = array_diff($array, $array_courses);
    foreach($result_array as $id){
    $data2 = array(
    'student_id' => $student_id,
    'course_id' => $id,
    'is_dropped' => "NO"
);
    $this->db->insert('course_students', $data2);
    $data3 = array(
    'student_id' => $student_id,
    'course_id' => $id,
    'cell_value' => '',
    'calculated_value' => 0
);
    $this->db->insert('final_grade', $data3);
    $this->course_model->update_order_in_course($id);
    $this->grade_model->update_students_in_class_record($id);
}
}
} else if($new_array_size < $old_array_size) {
    $result_array = array_diff($array_courses, $array);
    foreach($result_array as $id){
    $this->db->where('student_id', $student_id);
    $this->db->where('course_id', $id);
    $this->db->delete('course_students');
    $this->db->where('student_id', $student_id);
    $this->db->where('course_id', $id);
    $this->db->delete('final_grade');
    $this->course_model->update_order_in_course($id);
}
}
}

```

```

$this->grade_model->update_students_in_class_record($id);
}
} else {
$result_array = $array_courses;
foreach($result_array as $id){
$this->course_model->update_order_in_course($id);
$this->grade_model->update_students_in_class_record($id);
}
}
}

public function get_group_details($student_group_id){
$this->db->select('*');
$this->db->from('student_group');
$this->db->where('student_group_id',$student_group_id);
$query = $this->db->get();
return $query->row_array();
}

public function get_assigned_students_by_group($student_group_id){
$this->db->select('*');
$this->db->from('student_group_members');
$this->db->where('student_group_id',$student_group_id);
$query = $this->db->get();
$result = $query->result_array();
$array = array();
$index = 0;
foreach($result as $row){
$array[$index] = $row['student_id'];
$index++;
}
return $array;
}

public function update_student_group($group_id,$array_students){
$size = count($this->input->post('group_members'));
$data = array(
'student_group_name' => $this->input->post('student_group_name'),
'student_members_count' => $size
);
$this->db->where('student_group_id',$group_id);
$this->db->update('student_group',$data);
$array = $this->input->post('group_members');
$new_array_size = count($array);
$old_array_size = count($array_students);
if($new_array_size > $old_array_size){
$result_array = array_diff($array,$array_students);
foreach($result_array as $id){
$data2 = array(
'student_id' => $id,
'student_group_id' => $group_id,
);
$this->db->insert('student_group_members',$data2);
}
} else if($new_array_size < $old_array_size) {
$result_array = array_diff($array_students,$array);
foreach($result_array as $id){
$this->db->where('student_id', $id);
$this->db->delete('student_group_members');
}
}
}

public function get_list_of_students_by_course($course_id){
$this->db->select('*');
$this->db->from('student_as_us');
$this->db->where('s.school_id',$SESSION['logged_in']['school_id']);
$this->db->join('course_students_as_us','s.student_id=us.student_id','left');
$this->db->where('cs.course_id',$course_id);
$this->db->join('user_details_as_ud','s.user_id=ud.user_id','left');
$query = $this->db->get();
return $query->result_array();
}

public function get_list_of_students_id_by_course($course_id){
$this->db->select('*');
$this->db->from('student_as_us');
$this->db->where('s.school_id',$SESSION['logged_in']['school_id']);
$this->db->join('course_students_as_us','s.student_id=us.student_id','left');
$this->db->where('cs.course_id',$course_id);
$this->db->join('user_details_as_ud','s.user_id=ud.user_id','left');
$query = $this->db->get();
$result = $query->result_array();
$array = array();
$index = 0;
foreach($result as $row){
$array[$index] = $row['student_id'];
$index++;
}
return $array;
}

public function get_unassigned_students_by_course($course_id){
$this->db->select('*');
$this->db->from('student_as_us');
$this->db->where('s.school_id',$SESSION['logged_in']['school_id']);
$this->db->join('course_students_as_us','s.student_id=us.student_id','left');
$this->db->join('user_details_as_ud','s.user_id=ud.user_id','left');
$query = $this->db->get();
return $query->result_array();
}

public function get_student_group_members($array){
$this->db->select('sgm.student_group_id,COUNT(sgm.student_id)as student_members,us.student_group_name');
$this->db->from('student_group_members_as_sgm');
$this->db->where_not_in('sgm.student_id',$array);
$this->db->group_by('sgm.student_group_id');
$this->db->join('student_group_as_sg','sgm.student_group_id=sg.student_group_id','left');
$query = $this->db->get();
return $query->result_array();
}

public function get_student_group_members_by_ids($array,$assigned_students){
}

$this->db->select('*');
$this->db->from('student_group_members_as_sgm');
$this->db->distinct();
$this->db->where_in('sgm.student_group_id',$array);
$this->db->where_not_in('student_id', $assigned_students);
$this->db->join('student_group_as_sg','sgm.student_group_id=sg.student_group_id','left');
$query = $this->db->get();
$array = $query->result_array();
}

```



```

$new_array = array();
$index = 0;
foreach($array as $row){
    $new_array[$index] = $row['student_id'];
    $index++;
}
return array_unique($new_array);
}

public function remove_student_group($student_group_id,$school_id){
    $this->db->where('student_group_id',$student_group_id);
    $this->db->where('school_id',$school_id);
    $this->db->delete('student_group');
    $this->db->where('student_group_id',$student_group_id);
    $this->db->delete('student_group_members');
}

public function get_details_by_user_id($user_id){
    $this->db->select('*');
    $this->db->from('user_as_u');
    $this->db->where('u.user_id',$user_id);
    $this->db->where('u.school_id',$SESSION['logged_in']['school_id']);
    $this->db->join('student_as_s','u.user_id=u.s.user_id','left');
    $this->db->join('user_details_as_ud','u.user_id=u.ud.user_id','left');
    $query = $this->db->get();
    return $query->row_array();
}

public function get_student_details_by_course_id($course_id){
    $sql = "SELECT * FROM course_students AS cs INNER JOIN student AS s ON (cs
        .student_id=s.student_id) INNER JOIN user_details AS ud ON (s.
        user_id=ud.user_id) WHERE cs.course_id=". $this->db->escape(
        $course_id);
    $query = $this->db->query($sql);
    return $query->result_array();
}

public function get_order_number_in_course($course_id, $student_id){
    $sql = "SELECT order_number FROM course_students AS cs INNER JOIN student
        AS s ON (cs.student_id=s.student_id) WHERE cs.course_id=". $this->
        db->escape($course_id)." AND s.student_id=". $this->db->escape(
        $student_id);
    $query = $this->db->query($sql);
    return $query->row_array()['order_number'];
}

public function get_class_record_details_by_course_id($course_id){
    $sql = "SELECT * FROM gradebook_setup_details WHERE course_id=". $this->db
        ->escape($course_id);
    $query = $this->db->query($sql);
    $res = $query->row_array();
    $gradebook_id = $res['gradebook_setup_details_id'];
    $method = $res['method_of_computation'];
    $alphabet = array('B','C','D','E','F','G','H','I','J','K','L','M','N','O',
        'P','Q','R','S','T','U','V','W','X','Y','Z','AA','AB','AC','AD','AE',
        'AF','AG','AH','AJ','AK','AL','AM','AN','AD','AP','AQ','AR','AS',
        'AT','AU','AV','AW','AX','AY','AZ');
    $sql = "SELECT cs.is_dropped,cs.student_id,CONCAT(ud.user_last_name,',',
        ud.user_first_name) AS student_name FROM course_students AS cs INNER
        JOIN student AS s ON (cs.student_id=s.student_id) INNER JOIN
        user_details AS ud ON (s.user_id=u.ud.user_id) WHERE cs.course_id="
        $this->db->escape($course_id)." ORDER BY cs.order_number";
    $query = $this->db->query($sql);
    $current_array = $query->result_array();
    foreach($current_array as $key => $value){
        if($value['is_dropped'] == "YES"){
            $current_array[$key]['student_name'] .= ' (DRP)';
        }
    }
    $sql = "SELECT cr.student_id,cr.cell_value,m.column_letter,m.cell_type,cr.
        row_number FROM class_record AS cr INNER JOIN mapping AS m ON (cr.
        mapping_id=m.mapping_id) WHERE cr.gradebook_setup_details_id="
        $this->db->escape($gradebook_id)." ORDER BY m.column_letter,cr.
        row_number";
    $query = $this->db->query($sql);
    $current_array2 = $query->result_array();
    //echo '<pre>';
    //print_r($current_array);
    //echo '</pre>';
    $sql = 'SELECT order_number FROM course_students WHERE course_id=' . $this->
        db->escape($course_id) . ' AND is_dropped="YES"';
    $query = $this->db->query($sql);
    $result = $query->result_array();
    $order_numbers = array_column($result, 'order_number');
    $index = 0;
    $start_num = 0;
    foreach($current_array as $key => $value){
        foreach($current_array2 as $value2){
            if($value['student_id'] == $value2['student_id']){
                $current_array[$key]['alphabet'][$value2['column_letter']] = $value2['
                    cell_value'];
            }
        }
        $count_stud = $value2['row_number'];
    }
    if(count($current_array2) > 0){
        $start_num = $count_stud + 2;
        $count_stud -= 1;
        $start_num -= $count_stud;
    }
    $sql = "SELECT * FROM final_grade WHERE course_id=". $this->db->escape(
        $course_id);
    $query = $this->db->query($sql);
    $current_array3 = $query->result_array();
    $index = 0;
    foreach($current_array as $key => $value){
        foreach($current_array3 as $value2){
            if($value['student_id'] == $value2['student_id']){
                $current_array[$key]['final_grade'] = $value2['cell_value'];
                $current_array[$key]['scaled_grade'] = $value2['calculated_value'];
            }
        }
        $index++;
    }
    $student_count = $index - 1;
    $end_num = $start_num + $student_count;
    foreach($current_array2 as $value2){
        $current_array[$index]['alphabet'][$value2['column_letter']] = '';
        $current_array[$index]['final_grade'] = '';
        $current_array[$index]['scaled_grade'] = '';
    }
    $sql = "SELECT column_title, column_value, column_letter, cell_type FROM
        mapping WHERE gradebook_setup_details_id=" . $this->db->escape(
        $gradebook_id) . " ORDER BY column_letter";
    $q = $this->db->query($sql);

```

```

$res_array = $q->result_array();
$title_array = array('student_id' => '', 'student_name' => "");
$value_array = array('student_id' => '', 'student_name' => "");
$overall = 0;
foreach($res_array as $row){
    $title_array[$alphabet[$row['column_letter']]] = $row['column_title'];
    $value_array[$alphabet[$row['column_letter']]] = $row['column_value'];
}
if($method === "NaturalAggregate"){
    $overall = $res['total_points'];
    $title_array['final_grade'] = 'OverallTotal('.$overall.'.pts)';
} else {
    $overall = $res['max_grade'];
    $title_array['final_grade'] = 'OverallTotal('.$overall.'%)';
}
$value_array['final_grade'] = '';
$title_array['scaled_grade'] = 'FinalGrade';
$value_array['scaled_grade'] = '';
array_unshift($current_array, $value_array);
array_unshift($current_array, $title_array);
return $current_array;
}

public function get_student_record_details($course_id){
    $sql = "SELECT*FROMgradebook_setup_detailsWHEREcourse_id=".$this->db
        ->escape($course_id);
    $query = $this->db->query($sql);
    $res = $query->row_array();
    $gradebook_id = $res['gradebook_setup_details_id'];
    $method = $res['method_of_computation'];
    $alphabet = range('B','Z');
    $sql = "SELECTcs.student_id,CONCAT(ud.user_last_name,',',ud.
        user_first_name)ASstudent_nameFROMcourse_studentsAScsINNER
        JOINstudentASsON(cs.student_id=s.student_id)INNERJOIN
        user_detailsASudON(s.user_id=ud.user_id)WHEREcs.course_id="
        $this->db->escape($course_id)."ORDERBYcs.order_number";
    $query = $this->db->query($sql);
    $current_array = $query->result_array();
    $sql = "SELECTcr.student_id,cr.cell_value,m.column_letterFROM
        class_recordAScrINNERJOINmappingASmON(cr.mapping_id=m.
        mapping_id)WHEREcr.gradebook_setup_details_id=".$this->db->escape(
        $gradebook_id)."ORDERBYm.column_letter,cr.row_number";
    $query = $this->db->query($sql);
    $current_array2 = $query->result_array();
    //echo '<pre>';
    //print_r($current_array);
    //echo '</pre>';
    foreach($current_array as $key => $value){
        foreach($current_array2 as $value2){
            if($value['student_id'] === $value2['student_id']){
                $current_array[$key][$alphabet[$value2['column_letter']]] = $value2['
                    cell_value'];
            }
        }
    }
    $sql = "SELECT*FROMfinal_gradeWHEREcourse_id=".$this->db->escape(
        $course_id);
    $query = $this->db->query($sql);
    $current_array2 = $query->result_array();
    foreach($current_array as $key => $value){
        foreach($current_array2 as $value2){
            if($value['student_id'] === $value2['student_id']){
                $current_array[$key]['final_grade'] = $value2['cell_value'];
                $current_array[$key]['scaled_grade'] = $value2['calculated_value'];
            }
        }
    }
    $sql = "SELECTcolumn_title,column_value,column_letter,cell_typeFROM
        mappingWHEREgradebook_setup_details_id=".$this->db->escape(
        $gradebook_id)."ORDERBYcolumn_letter";
    $q = $this->db->query($sql);
    $res_array = $q->result_array();
    $title_array = array('student_id' => '', 'student_name' => "");
    $value_array = array('student_id' => '', 'student_name' => "");
    $overall = 0;
    foreach($res_array as $row){
        $title_array[$alphabet[$row['column_letter']]] = $row['column_title'];
        $value_array[$alphabet[$row['column_letter']]] = $row['column_value'];
    }
    if($method === "NaturalAggregate"){
        $overall = $res['total_points'];
        $title_array['final_grade'] = 'OverallTotal('.$overall.'.pts)';
    } else {
        $overall = $res['max_grade'];
        $title_array['final_grade'] = 'OverallTotal('.$overall.'%)';
    }
    $value_array['final_grade'] = '';
    $title_array['scaled_grade'] = 'FinalGrade';
    $value_array['scaled_grade'] = '';
    array_unshift($current_array, $value_array);
    array_unshift($current_array, $title_array);
    return $current_array;
}

public function drop_student($course_id, $student_id){
    $data = array('is_dropped' => 'YES');
    $this->db->where('course_id', $course_id);
    $this->db->where('student_id', $student_id);
    $this->db->update('course_students', $data);
    $data2 = array('cell_value' => 'DRP', 'calculated_value' => 'DRP');
    $this->db->where('course_id', $course_id);
    $this->db->where('student_id', $student_id);
    $this->db->update('final_grade', $data2);
}

public function get_grades_stats_by_course_id($course_id){
    $sql = "SELECT*FROMgradebook_setup_detailsWHEREcourse_id=".$this->db
        ->escape($course_id);
    $query = $this->db->query($sql);
    $res = $query->row_array();
    $gradebook_id = $res['gradebook_setup_details_id'];
    $method = $res['method_of_computation'];
    $alphabet = array('B','C','D','E','F','G','H','I','J','K','L','M','N','O',
        'P','Q','R','S','T','U','V','W','X','Y','Z','AA','AB','AC','AD','AE',
        'AF','AG','AH','AJ','AK','AL','AM','AN','AO','AP','AQ','AR','AS',
        'AT','AU','AV','AW','AX','AY','AZ');
    $sql = "SELECTcs.student_id,CONCAT(ud.user_last_name,',',ud.
        user_first_name)ASstudent_nameFROMcourse_studentsAScsINNER
        JOINstudentASsON(cs.student_id=s.student_id)INNERJOIN
        user_detailsASudON(s.user_id=ud.user_id)WHEREcs.course_id="
        $this->db->escape($course_id)."ORDERBYcs.order_number";
    $query = $this->db->query($sql);
    $current_array = $query->result_array();
}

```

```

$sql = "SELECT cr.student_id,cr.cell_value,m.column_letter,m.cell_type,cr.
row_number FROM class_record as cr INNER JOIN mapping as m ON (cr.
mapping_id=m.mapping_id) WHERE cr.gradebook_setup_details_id=" .
$this->db->escape($gradebook_id)." ORDER BY m.column_letter ,cr.
row_number";

$query = $this->db->query($sql);
$current_array2 = $query->result_array();
//echo '<pre>';
//print_r($current_array);
//echo '</pre>';

$sql = 'SELECT order_number from course_students WHERE course_id=' . $this->
db->escape($course_id)." and is_dropped='YES'";
$query = $this->db->query($sql);
$result = $query->result_array();
$order_numbers = array_column($result, 'order_number');
$index = 0;
$start_num = 0;
foreach($current_array as $key => $value){
foreach($current_array2 as $value2){
if($value['student_id'] === $value2['student_id']){
$current_array[$key][alphabet[$value2['column_letter']]] = $value2['
cell_value'];
}
$count_stud = $value2['row_number'];
}
}

if(count($current_array2) > 0){
$start_num = $count_stud + 2;
$count_stud -- 1;
$start_num -- $count_stud;
}

$sql = "SELECT * FROM final_grade WHERE course_id=" . $this->db->escape(
$course_id);
$query = $this->db->query($sql);
$current_array3 = $query->result_array();
$sql3 = "SELECT MAX(column_letter) AS column_letter FROM mapping WHERE
gradebook_setup_details_id=" . $this->db->escape($gradebook_id);
$special_query = $this->db->query($sql3);
$special_query_array = $special_query->row_array();
$max_column = $special_query_array['column_letter'];
$index = 0;
foreach($current_array as $key => $value){
foreach($current_array3 as $value2){
if($value['student_id'] === $value2['student_id']){
$current_array[$key][alphabet[$max_column + 1]] = $value2['cell_value'];
$current_array[$key][alphabet[$max_column + 2]] = $value2['
calculated_value'];
}
}

$index++;
}

$student_count = $index - 1;
$end_num = $start_num + $student_count;
foreach($current_array2 as $value2){
$current_array[$index][alphabet[$value2['column_letter']]] = '';
$current_array[$index]['final_grade'] = '';
$current_array[$index]['scaled_grade'] = '';
}

$sql = "SELECT column_title ,column_value ,column_letter ,cell_type FROM
mapping WHERE gradebook_setup_details_id=" . $this->db->escape(
$gradebook_id)." ORDER BY column_letter";

$q = $this->db->query($sql);
$res_array = $q->result_array();
$title_array = array('student_id' => '', 'student_name' => "");
$value_array = array('student_id' => '', 'student_name' => "");
$overall = 0;
foreach($res_array as $row){
$title_array[alphabet[$row['column_letter']]] = $row['column_title'];
$value_array[alphabet[$row['column_letter']]] = $row['column_value'];
}

if($method === "NaturalAggregate"){
$overall = $res['total_points'];
$title_array['final_grade'] = 'Overall Total (' . $overall . 'pts)';
} else {
$overall = $res['max_grade'];
$title_array['final_grade'] = 'Overall Total (' . $overall . '%)';
}

$value_array['final_grade'] = '';
$title_array['scaled_grade'] = 'Final Grade';
$value_array['scaled_grade'] = '';
array_unshift($current_array, $value_array);
array_unshift($current_array, $title_array);
return $current_array;
}

public function get_email_addresses_by_course($course_id){
$sql = "SELECT ud.user_email_address FROM user_details as ud INNER JOIN
user as u ON (u.user_id=ud.user_id) INNER JOIN student as s ON (s.
user_id=u.user_id) INNER JOIN course_students as cs ON (cs.
student_id=s.student_id) WHERE cs.course_id=" . $this->db->escape(
$course_id)." AND ud.email_verified='YES'";
$query = $this->db->query($sql);
$current_array2 = $query->result_array();
$res_array = array_column($current_array2, 'user_email_address');
return $res_array;
}

public function remove_student_from_database($student_id){
$sql = "SELECT course_id FROM course_students WHERE student_id=" . $this->db
->escape($student_id)." and is_dropped='NO'";
$query = $this->db->query($sql);
$result = $query->result_array();
$id_array = array_column($result, 'course_id');
$sql2 = "SELECT user_id FROM student WHERE student_id=" . $this->db->escape(
$student_id);
$query2 = $this->db->query($sql2);
$result2 = $query2->row_array();
$user_id = $result2['user_id'];
$this->db->trans_start();
$this->db->where('student_id', $student_id);
$this->db->delete('course_students');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('student_id', $student_id);
$this->db->delete('final_grade');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('student_id', $student_id);
$this->db->delete('class_record');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('student_id', $student_id);

```

```

$this->db->delete('student');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('user_id', $user_id);
$this->db->delete('user_details');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('user_id', $user_id);
$this->db->delete('user');
$this->db->trans_complete();
if(count($id_array) > 0){
foreach($id_array as $key => $value){
$this->course_model->update_order_in_course($value);
$this->grade_model->update_students_in_class_record($value);
}
}
}
}

```

Teacher Model

```

<?php
if (!defined('BASEPATH')) exit('No direct script access allowed');
class Teacher_model extends CI_Model {
public function __construct(){
$this->load->database();
$this->load->helper('url_helper');
$this->load->library('session');
}
public function add_teacher(){
$max = $this->get_last_teacher_index();
if($max['latest_id'] > 0){
$array = $this->generate_teacher_number($max['latest_id'] + 1, 1);
} else {
$array = $this->generate_teacher_number(1,1);
}
$salt = uniqid(mt_rand(), true);
$password = $salt.$array[0];
$data2 = array(
'user_name' => $array[0],
'school_id' => $_SESSION['logged_in']['school_id'],
'user_password' => hash('sha256', $password),
'user_password_salt' => $salt,
'user_role' => "TEACHER"
);
$this->db->insert('user', $data2);
$this->db->select('*');
$this->db->from('user');
$this->db->where('user_name', $array[0]);
$query = $this->db->get();
$user_id = $query->row_array()['user_id'];
$data = array(
'user_id' => $user_id,
'user_first_name' => $this->input->post('teacher_first_name'),
'user_last_name' => $this->input->post('teacher_last_name'),
'user_email_address' => $this->input->post('teacher_email_address'),
'email_verified' => "NO"
);
$this->db->insert('user_details', $data);
$data3 = array(

```

```

'user_id' => $user_id,
'teacher_number' => $array[0],
'school_id' => $_SESSION['logged_in']['school_id'],
'first_login' => "YES"
);
$this->db->insert('teacher', $data3);
$this->db->select('*');
$this->db->from('teacher');
$this->db->where('teacher_number', $array[0]);
$query = $this->db->get();
$teacher_id = $query->row_array()['teacher_id'];
$array2 = $this->input->post('available_courses');
if(count($array2) > 0){
foreach($array2 as $row){
$data = array(
'teacher_id' => $teacher_id,
);
$this->db->where('course_id', $row);
$this->db->update('course', $data);
}
}
}
public function add_multiple_teacher(){
$first_name = $this->input->post('first_name');
$last_name = $this->input->post('last_name');
$email_address = $this->input->post('email_address');
$array_set = array();
$array_first_names = array();
$array_last_names = array();
$array_email_dups = array();
$count = 0;
$array_emails = array();
$size = count($first_name);
for($index = 0; $index < $size; $index++){
if($this->user_model->check_email_address($email_address[$index])){
$array_first_names[$count] = $first_name[$index];
$array_last_names[$count] = $last_name[$index];
$array_email_dups[$count] = $email_address[$index];
$count++;
} else {
$max = $this->get_last_teacher_index();
if($max['latest_id'] > 0){
$array = $this->generate_teacher_number($max['latest_id'] + 1, 1);
} else {
$array = $this->generate_teacher_number(1,1);
}
$salt = uniqid(mt_rand(), true);
$password = $salt.$array[0];
$data2 = array(
'user_name' => $array[0],
'school_id' => $_SESSION['logged_in']['school_id'],
'user_password' => hash('sha256', $password),
'user_password_salt' => $salt,
'user_role' => "TEACHER"
);
$this->db->insert('user', $data2);
$this->db->select('*');
$this->db->from('user');
$this->db->where('user_name', $array[0]);
$query = $this->db->get();

```

```

$user_id = $query->row_array()['user_id'];
$data = array(
    'user_id' => $user_id,
    'user_first_name' => $first_name[$index],
    'user_last_name' => $last_name[$index],
    'user_email_address' => $email_address[$index],
    'email_verified' => "NO"
);
$this->db->insert('user_details',$data);
array_push($array_emails, $email_address[$index]);
$data3 = array(
    'user_id' => $user_id,
    'teacher_number' => $array[0],
    'first_login' => "YES",
    'school_id' => $_SESSION['logged_in']['school_id']
);
$this->db->insert('teacher',$data3);
}
}
array_push($array_set, $array_emails);
array_push($array_set, $array_first_names);
array_push($array_set, $array_last_names);
array_push($array_set, $array_email_dups);
return $array_set;
}

function get_last_teacher_index(){
    $this->db->select_max('teacher_id','latest_id');
    $this->db->from('teacher');
    $query = $this->db->get();
    return $query->row_array();
}

public function generate_teacher_number($start, $count){
    $data = array();
    for($i = 0; $i < $count; $i++){
        $data[$i] = "E".date("Y").str_pad($start,5,0,STR_PAD_LEFT);
        $start++;
    }
    return $data;
}

public function get_list_of_teachers(){
    $this->db->select('*');
    $this->db->from('teacher_⊔as⊔t');
    $this->db->where('t.school_id',$_SESSION['logged_in']['school_id']);
    $this->db->join('user_details_⊔as⊔ud','t.user_id=⊔ud.user_id','left');
    $query = $this->db->get();
    return $query->result_array();
}

public function get_details_by_id($teacher_id){
    $this->db->select('*');
    $this->db->from('teacher_⊔as⊔t');
    $this->db->where('t.teacher_id',$teacher_id);
    $this->db->where('t.school_id',$_SESSION['logged_in']['school_id']);
    $this->db->join('user_⊔as⊔u','t.user_id=⊔u.user_id','left');
    $this->db->join('user_details_⊔as⊔ud','u.user_id=⊔ud.user_id','left');
    $query = $this->db->get();
    return $query->row_array();
}

public function get_details_by_user_id($user_id){
    $this->db->select('*');
    $this->db->from('user_⊔as⊔u');
    $this->db->where('u.user_id',$user_id);
    $this->db->where('u.school_id',$_SESSION['logged_in']['school_id']);
    $this->db->join('teacher_⊔as⊔t','u.user_id=⊔t.user_id','left');
    $this->db->join('user_details_⊔as⊔ud','u.user_id=⊔ud.user_id','left');
    $query = $this->db->get();
    return $query->row_array();
}

public function get_assigned_courses($teacher_id){
    $this->db->select('*');
    $this->db->from('course');
    $this->db->where('teacher_id',$teacher_id);
    $query = $this->db->get();
    $result = $query->result_array();
    $array = array();
    $index = 0;
    foreach($result as $row){
        $array[$index] = $row['course_id'];
        $index++;
    }
    return $array;
}

public function update_teacher($teacher_id, $array_courses) {
    $this->db->select('*');
    $this->db->from('teacher');
    $this->db->where('teacher_id',$teacher_id);
    $query = $this->db->get();
    $array = $query->row_array();
    $user_id = $array['user_id'];
    $data = array(
        'user_last_name' => $this->input->post('teacher_last_name'),
        'user_first_name' => $this->input->post('teacher_first_name'),
        'user_email_address' => $this->input->post('teacher_email_address')
    );
    $this->db->where('user_id',$user_id);
    $this->db->update('user_details',$data);
    $array = $this->input->post('available_courses');
    $new_array_size = count($array);
    $old_array_size = count($array_courses);
    if($new_array_size > $old_array_size){
        $result_array = array_diff($array,$array_courses);
        foreach($result_array as $id){
            $data2 = array(
                'teacher_id' => $teacher_id
            );
            $this->db->where('course_id',$id);
            $this->db->update('course',$data2);
        }
    } else if($new_array_size < $old_array_size) {
        $result_array = array_diff($array_courses,$array);
        foreach($result_array as $id){
            $data2 = array(
                'teacher_id' => 0
            );
            $this->db->where('course_id',$id);
            $this->db->update('course',$data2);
        }
    }
}

public function remove_teacher_from_database($teacher_id){

```

```

$sql1 = "SELECT course_id FROM course WHERE teacher_id=". $this->db->escape(
    $teacher_id);
$query = $this->db->query($sql);
$result = $query->result_array();
$id_array = array_column($result, 'course_id');
$sql12 = "SELECT user_id FROM teacher WHERE teacher_id=". $this->db->escape(
    $teacher_id);
$query2 = $this->db->query($sql12);
$result2 = $query2->row_array();
$user_id = $result2['user_id'];
$sql14 = "SELECT gradebook_template_id FROM gradebook_template WHERE
    teacher_id=". $this->db->escape($teacher_id);
$query4 = $this->db->query($sql14);
$result4 = $query4->row_array();
$gt_id = $result4['gradebook_template_id'];
$sql13 = "SELECT course_category_id FROM course_category WHERE teacher_id="
    . $this->db->escape($teacher_id);
$query3 = $this->db->query($sql13);
$result3 = $query3->result_array();
$id_cat_array = array_column($result3, 'course_category_id');
$this->db->trans_start();
$this->db->where('teacher_id', $teacher_id);
$this->db->delete('teacher');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('teacher_id', $teacher_id);
$this->db->delete('gradebook_template');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('teacher_id', $teacher_id);
$this->db->delete('gradebook_template');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('gradebook_template_id', $gt_id);
$this->db->delete('template_members');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('user_id', $user_id);
$this->db->delete('user_details');
$this->db->trans_complete();
$this->db->trans_start();
$this->db->where('user_id', $user_id);
$this->db->delete('user');
$this->db->trans_complete();
if(count($id_array) > 0){
    foreach($id_array as $key => $value){
        $data = array(
            'teacher_id' => 0
        );
        $this->db->where('course_id', $value);
        $this->db->update('course', $data);
    }
}
if(count($id_cat_array) > 0){
    foreach($id_cat_array as $key => $value){
        $data = array(
            'teacher_id' => 0
        );
        $this->db->where('course_category_id', $value);
        $this->db->update('course_category', $data);
    }
}

```

User Model

```

<?php
if (!defined('BASEPATH')) exit('No direct script access allowed');
class User_model extends CI_Model {
    public function __construct(){
        $this->load->database();
        $this->load->helper('url_helper');
        $this->load->library('session');
    }
    public function register_admin(){
        $data = array(
            'school_name' => $this->input->post('school_name')
        );
        $this->db->insert('school', $data);
        $this->db->select('*');
        $this->db->from('school');
        $this->db->where('school_name', $this->input->post('school_name'));
        $query = $this->db->get();
        $array = $query->row_array();
        $school_id = $array['school_id'];
        $salt = uniqid(mt_rand(), true);
        $password = $salt.$this->input->post('user_password');
        $password = hash('sha256', $password);
        $data2 = array(
            'school_id' => $school_id,
            'user_name' => $this->input->post('user_name'),
            'user_password' => $password,
            'user_password_salt' => $salt,
            'user_role' => "ADMIN"
        );
        $this->db->insert('user', $data2);
        $this->db->select('*');
        $this->db->from('user');
        $this->db->where('user_name', $this->input->post('user_name'));
        $query = $this->db->get();
        $array = $query->row_array();
        $user_id = $array['user_id'];
        $data3 = array(
            'user_id' => $user_id,
            'user_first_name' => $this->input->post('user_first_name'),
            'user_last_name' => $this->input->post('user_last_name'),
            'user_email_address' => $this->input->post('user_email_address'),
            'email_verified' => "NO"
        );
        $this->db->insert('user_details', $data3);
    }
    public function check_user_name($user_name){
        $this->db->select('*');
        $this->db->from('user');
        $this->db->where('user_name', $user_name);
        $query = $this->db->get();
        if($query->num_rows() == 1)
        {
            return true;
        }
    }
}

```

```

}
else
{
return false;
}
}

public function check_school_name($school_name){
$this->db->select('*');
$this->db->from('school');
$this->db->where('school_name', $school_name);
$query = $this->db->get();
if($query -> num_rows() == 1)
{
return true;
}
else
{
return false;
}
}

public function check_email_address($email_address){
$this->db->select('*');
$this->db->from('user_details');
$this->db->where('user_email_address', $email_address);
$query = $this->db->get();
if($query -> num_rows() == 1)
{
return true;
}
else
{
return false;
}
}

public function check_email_address_first($email_address){
$this->db->select('*');
$this->db->from('user_details');
$this->db->where('user_email_address', $email_address);
$this->db->where('email_verified', 'NO');
$query = $this->db->get();
if($query -> num_rows() == 1)
{
return true;
}
else
{
return false;
}
}

public function get_password_salt($user_name){
$this->db->select('*');
$this->db->from('user');
$this->db->where('user_name', $user_name);
$query = $this->db->get();
if($query->num_rows() == 1){
return $query->row_array();
} else {
return false;
}
}

}

public function login_details($user_id, $user_name, $user_password){
$this->db->select('*');
$this->db->from('user_as_u');
$this->db->where('u.user_name', $user_name);
$this->db->where('u.user_password', $user_password);
$this->db->where('u.user_id', $user_id);
$this->db->join('user_details_as_ud', 'u.user_id=ud.user_id', 'left');
$query = $this->db->get();
if($query -> num_rows() == 1)
{
return $query->result();
}
else
{
return false;
}
}

public function get_users(){
$this->db->select('*');
$this->db->from('user_details');
$query = $this->db->get();
return $query->result_array();
}

public function check_email_verification($email_address){
$data = array ('email_verified' => "YES");
$this->db->where('user_email_address', $email_address);
$this->db->update('user_details', $data);
$this->db->select('*');
$this->db->from('user_details_as_ud');
$this->db->where('ud.user_email_address', $email_address);
$this->db->join('user_as_u', 'ud.user_id=u.user_id', 'left');
$query = $this->db->get();
return $query->row_array();
}

public function get_user_details_by_email($email_address){
$this->db->select('*');
$this->db->from('user_details_as_ud');
$this->db->where('ud.user_email_address', $email_address);
$this->db->join('user_as_u', 'ud.user_id=u.user_id', 'left');
$query = $this->db->get();
return $query->row_array();
}

public function get_user_details_by_id($user_id){
$this->db->select('*');
$this->db->from('user_details_as_ud');
$this->db->where('ud.user_id', $user_id);
$this->db->join('user_as_u', 'ud.user_id=u.user_id', 'left');
$query = $this->db->get();
return $query->row_array();
}
}

public function get_email_address($user_id){
$this->db->select('*');
$this->db->from('user_details');
$this->db->where('user_id', $user_id);
$query = $this->db->get();
$array = $query->row_array();
return $array['user_email_address'];
}

public function get_user_by_name($user_name){
$this->db->select('*');

```

```

$this->db->from('user');
$this->db->where('user_name', $user_name);
$query = $this->db->get();
$array = $query->row_array();
return $array['user_id'];
}

public function get_school_name($school_id){
$this->db->select('*');
$this->db->from('school');
$this->db->where('school_id', $school_id);
$query = $this->db->get();
$array = $query->row_array();
return $array['school_name'];
}

public function update_new_password($user_name,$user_password){
$salt = uniqid(mt_rand(), true);
$password = $salt.$user_password;
$password = hash('sha256',$password);
$data = array ('user_password' => $password,'user_password_salt' => $salt)
;
$this->db->where('user_name',$user_name);
$this->db->update('user',$data);
}

public function check_password($password, $user_id){
$this->db->select('user_password_salt');
$this->db->from('user');
$this->db->where('user_id',$user_id);
$query = $this->db->get();
$res = $query->row_array();
$salt = $res['user_password_salt'];
$new_pass = $salt.$password;
$new_pass = hash("sha256", $new_pass);
$this->db->select('*');
$this->db->where('user_password',$new_pass);
$this->db->where('user_id',$user_id);
$this->db->from('user');
$query = $this->db->get();
if ($query->num_rows() > 0){
return true;
}
else{
return false;
}
}

public function get_password_salt_by_id($user_id){
$this->db->select('user_password_salt');
$this->db->from('user');
$this->db->where('user_id',$user_id);
$query = $this->db->get();
$res = $query->row_array();
$salt = $res['user_password_salt'];
return $salt;
}

public function change_password($password, $user_id){
$data = array('user_password' => $password);
$this->db->where('user_id',$user_id);
$this->db->update('user', $data);
}

public function update_force_password($user_name,$user_password,$user_role
,$user_id){
if($user_role === "TEACHER") {
$data = array ('first_login' => 'NO');
$this->db->where('user_id',$user_id);
$this->db->update('teacher',$data);
} else if ($user_role === "STUDENT") {
$data = array ('first_login' => 'NO');
$this->db->where('user_id',$user_id);
$this->db->update('student',$data);
}

$salt = uniqid(mt_rand(), true);
$password = $salt.$user_password;
$password = hash('sha256',$password);
$data2 = array ('user_password' => $password,'user_password_salt' => $salt
);
$this->db->where('user_id',$user_id);
$this->db->update('user',$data2);
}

function send_verify_mail($email_addresses){
$subject = 'Electronic_Class_Record_Inquiry_System_Email_Verification';
// Also, for getting full html you may use the following internal method:
// $body = $this->email->full_html($subject, $message);
$mail = new PHPMailer(); // create a new object
$mail->isSMTP(); // enable SMTP
// debugging: 1 = errors and messages, 2 = messages only
$mail->SMTPAuth = true; // authentication enabled
$mail->SMTPSecure = 'ssl'; // secure transfer enabled REQUIRED for Gmail
$mail->Host = "ssl://smtp.gmail.com:465";
$mail->Port = 465; // or 587
$mail->isHTML(true);
$mail->Username = "ecrisapplication@gmail.com";
$mail->Password = "ecris1234";
$mail->SetFrom("ecrisapplication@gmail.com");
$mail->Subject = $subject;
$mail->Body = $message;
foreach($email_addresses as $key => $value){
$mail->AddAddress($value);
}
$mail->SMTPOptions = array(
'ssl' => array(
'verify_peer' => false,
'verify_peer_name' => false,
'allow_self_signed' => true
)
);
);
$mail->Send();
return true;
}

function send_reset_email($user_email_address){
$subject = 'Electronic_Class_Record_Inquiry_System_Reset_Password';
// Also, for getting full html you may use the following internal method:
// $body = $this->email->full_html($subject, $message);
$mail = new PHPMailer(); // create a new object
$mail->isSMTP(); // enable SMTP
// debugging: 1 = errors and messages, 2 = messages only
$mail->SMTPAuth = true; // authentication enabled
$mail->SMTPSecure = 'ssl'; // secure transfer enabled REQUIRED for Gmail
$mail->Host = "ssl://smtp.gmail.com:465";
$mail->Port = 465; // or 587
$mail->isHTML(true);
$mail->Username = "ecrisapplication@gmail.com";
}

```



```

$mail->Password = "ecris1234";
$mail->SetFrom("ecrisapplication@gmail.com");
$mail->Subject = $subject;
$mail->Body = $message;
$mail->AddAddress($user_email_address);
$mail->SMTPOptions = array(
    'ssl' => array(
        'verify_peer' => false,
        'verify_peer_name' => false,
        'allow_self_signed' => true
    )
);
$mail->Send();
return true;
}

function send_announcement_mails($email_addresses, $course_details){
    $subject = '[ECRIS]['. $course_details['course_short_name'].'] ['.
        $course_details['section_name'].']_Announcement';
    // Also, for getting full html you may use the following internal method:
    // $body = $this->email->full_html($subject, $message);
    $mail = new PHPMailer(); // create a new object
    $mail->IsSMTP(); // enable SMTP
    // debugging: 1 = errors and messages, 2 = messages only
    $mail->SMTPAuth = true; // authentication enabled
    $mail->SMTPSecure = 'ssl'; // secure transfer enabled REQUIRED for Gmail
    $mail->Host = "ssl://smtp.gmail.com:465";
    $mail->Port = 465; // or 587
    $mail->IsHTML(true);
    $mail->Username = "ecrisapplication@gmail.com";
    $mail->Password = "ecris1234";
    $mail->SetFrom("ecrisapplication@gmail.com");
    $mail->Subject = $subject;
    $mail->Body = $message;
    foreach($email_addresses as $key => $value){
        $mail->AddAddress($value);
    }
    $mail->SMTPOptions = array(
        'ssl' => array(
            'verify_peer' => false,
            'verify_peer_name' => false,
            'allow_self_signed' => true
        )
    );
    $mail->Send();
    return true;
}
}

```

Account File Contents View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
    bottom: 0">
<div class="navbar-header">
<a class="navbar-minimalize minimalize-styl-2 btn btn-primary" href="#"><
    i class="fa fa-bars"></i> </a>
</div>
<ul class="navbar-nav navbar-top-links navbar-right">
<li>

```

```

<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
    ;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailbox.html">
<div>
<i class="fa fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-4">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li class="active">
<a href="<?php echo base_url('')?>">Home</a>
</li>

```

```

</li>
<a>Courses</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('upload_accounts');?>">Upload
    accounts</a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<h5>List of accounts</h5>
</div>
<div class="ibox-content">
<div class="form-group">
<?php echo form_open('add_multiple_accounts','','array('upload_user_role'
=> $upload_user_role));?>
<div class="table-responsive">
<table class="table table-bordered" id="dynamic_field">
<tr>
<?php if(isset($headers)) {?>
<?php foreach($headers as $label) {?>
<th><?php echo $label;?></th>
<?php } ?>
<?php } ?>
</tr>
<?php $index = 0;?>
<?php if(isset($values)) {?>
<?php $size = count($values);?>
<?php foreach($values as $label) {?>
<?php if ($index == 0) {?>
<tr>
<td><input type="text" name="first_name[]" value="<?php echo $label[0];?>"
    placeholder="First Name" class="form-control name_list" required="
    required" /></td>
<td><input type="text" name="last_name[]" value="<?php echo $label[1];?>"
    placeholder="Last Name" class="form-control name_list" required="
    required" /></td>
<td><input type="email" name="email_address[]" value="<?php echo $label
[2];?>" placeholder="Email Address" class="form-control name_list"
    required="required" /></td>
<td><button type="button" name="add" id="add" class="btn btn-success"><i
class="fa fa-plus-square"></i></button></td>
</tr>
<?php } else { ?>
<tr id="row<?php echo $index;?>">
<td><input type="text" name="first_name[]" value="<?php echo $label[0];?>"
    placeholder="First Name" class="form-control name_list" required="
    required" /></td>
<td><input type="text" name="last_name[]" value="<?php echo $label[1];?>"
    placeholder="Last Name" class="form-control name_list" required="
    required" /></td>
<td><input type="email" name="email_address[]" value="<?php echo $label
[2];?>" placeholder="Email Address" class="form-control name_list"
    required="required" /></td>
<td><button type="button" name="remove" id="<?php echo $index;?>" class="
    btn btn-danger btn_remove">X</button></td>

```

```

</tr>
<?php } ?>
<?php $index++; ?>
<?php } ?>
<?php } ?>
</table>
<a class="btn btn-white" href="<?php echo base_url('upload_accounts');?>"
/>Cancel</a>
<input type="submit" name="submit" id="submit" class="btn btn-info" value=
"Submit" />
</div>
</div>
</div>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>
</div>

```

Account Settings

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
bottom: 0">
<div class="navbar-header">
<a class="navbar-minimalize minimalize-styl-2 btn btn-primary" href="#"><i
class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
;?></span>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-6">
<h2><i class="fa fa-gear"></i> <?php echo $title;?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('')?>">Home</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('account_settings');?>">Account
    Settings</a></strong>
</li>
</ol>
</div>

```



```

<span class="m-r-sm_text-muted_welcome-message"><?php echo $school_name
;?></span>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row_wrapper border-bottom white-bg page-heading">
<div class="col-sm-4">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li class="active">
<strong><a href="<?php echo base_url('')?>">Home</a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="wrapper_wrapper-content">
<div class="col-sm-6">
<div class="ibox_float-e-margins">
<div class="ibox-title">
<h5>Accounts</h5>
</div>
<div class="ibox-content">
<div class="well">
<h3>
Register/Update accounts
</h3>
<ul class="list-group">
<li class="list-group-item">
<span class="badge badge-primary"><a href="<?php echo base_url('
student_form')?>" style="color:white;" onMouseOver="this.style.color
='red'" onMouseOut="this.style.color='white'"><i class='fa fa-plus'
></i> Register</a></span>
<span class="badge badge-warning"><a href="<?php echo base_url('
student_list')?>" style="color:white;" onMouseOver="this.style.color
='red'" onMouseOut="this.style.color='white'"><i class='fa fa-list'
></i> View list</a></span>
Students
</li>
<li class="list-group-item">
<span class="badge badge-primary"><a href="<?php echo base_url('
teacher_form')?>" style="color:white;" onMouseOver="this.style.color
='red'" onMouseOut="this.style.color='white'"><i class='fa fa-plus'
></i> Register</a></span>
<span class="badge badge-warning"><a href="<?php echo base_url('
teacher_list')?>" style="color:white;" onMouseOver="this.style.color
='red'" onMouseOut="this.style.color='white'"><i class='fa fa-list'
></i> View list</a></span>
Teachers
</li>
</ul>
</div>
<h3>For multiple accounts

```

```

<a type="button" href="<?php echo base_url('upload_accounts')?>" class="
btn btn-danger"><i class='fa fa-arrow-up'></i> Upload csv</a>
</h3>
</div>
</div>
<div class="col-sm-6">
<div class="ibox_float-e-margins">
<div class="ibox-title">
<h5>Courses</h5>
</div>
<div class="ibox-content">
<div class="well">
<h3>
Add/update courses
</h3>
<ul class="list-group">
<li class="list-group-item">
<span class="badge badge-primary"><a href="<?php echo base_url('
course_form')?>" style="color:white;" onMouseOver="this.style.color
='red'" onMouseOut="this.style.color='white'"><i class='fa fa-plus'
></i> Create</a></span>
<span class="badge badge-warning"><a href="<?php echo base_url('
course_list')?>" style="color:white;" onMouseOver="this.style.color
='red'" onMouseOut="this.style.color='white'"><i class='fa fa-list'
></i> View list</a></span>
Courses
</li>
</ul>
</div>
<div class="well">
<h3>
Course Category
</h3>
<a type="button" href="<?php echo base_url('course_category_form')?>"
class="btn btn-sm btn-primary"><i class='fa fa-plus'></i> Create</a>
<a type="button" href="<?php echo base_url('course_category_list')?>"
class="btn btn-sm btn-warning pull-right"><i class='fa fa-list'></i>
List of course categories</a>
</div>
</div>
</div>
<div class="well">
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>
Category Courses List View
<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
bottom:0">

```

```

<div class="navbar-header">
<a class="navbar-minimalize_minimalize-styl-2_btn btn-primary" href="#">
    i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm_text-muted_welcome-message"><?php echo $school_name
;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle_count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label_label-primary">8</span>
</a>
<ul class="dropdown-menu_dropdown-alerts">
<li>
<a href="mailbox.html">
<div>
<i class="fa fa-envelope_fa-fw"></i> You have 16 messages
<span class="pull-right_text-muted_small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter_fa-fw"></i> 3 New Followers
<span class="pull-right_text-muted_small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload_fa-fw"></i> Server Rebooted
<span class="pull-right_text-muted_small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center_link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row_wrapper border-bottom_white-bg_page-heading">
<div class="col-sm-8">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li class>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li class>
<a href="#">Courses</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('category_courses/'. $category_details
['course_category_id']);?>">Courses - <?php echo $category_details['
course_category_name'];?></a></strong>
</li>
</ol>
</div>
</div>
<div class="wrapper_wrapper-content">
<div class="row">
<div class="col-lg-12">
<?php if($this->session->flashdata('remove_course_succeed')) { ?>
<div class="alert_alert-danger"><strong>Success!</strong>
<p>The course is completely removed from this category.</p>
</div>
<?php } ?>
<div class="ibox_float-e-margins">
<div class="ibox-title">
<h5>Assigned Person:
<?php if(strlen(utf8_decode($category_details['user_first_name'])) > 0) {?>
<?php echo $category_details['user_last_name'].', '. $category_details['
user_first_name'];?>
<?php } else { ?>
Not yet assigned.
<?php } ?>
</h5>
</div>
<div class="ibox-content">
<table class="table_table-striped_table-bordered_table-hover_dataTables-
example" >
<thead>
<tr>
<th>Course ID</th>
<th>Course Name</th>
<th>Assigned Teacher</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<?php $size = count($courses);?>
<?php if(isset($courses) && count($courses) > 0) {?>
<?php for($a = 0; $a < $size; $a++){ ?>
<tr>
<td><?php echo $courses[$a]['course_id'];?></td>
<td><?php echo $courses[$a]['course_short_name'].', '. $courses[$a]['
course_full_name'].', '?></td>
<?php if(strlen(utf8_decode($courses[$a]['user_first_name'])) > 0) {?>
<td>
<?php echo $courses[$a]['user_first_name'].', '. $courses[$a]['
user_last_name'];?>
</td>

```

```

<?php } else { ?>
<td><span class="label_label-danger">No teacher assigned</span></td>
<?php } ?>
<?php $this->session->set_userdata('go_to_category', $category_details['
course_category_id'], 300); ?>
<td><a class="btn btn-sm btn-success" href="<?php echo base_url('
edit_course_details/'. $courses[$a]['course_id'])?">Edit</a>&nbsp;<
button data-toggle="modal" data-target="#removeCourseModal" data-
whatever=<?php echo $courses[$a]['course_id'];?>" class="btn btn-sm
btn-danger">Remove</button></td>
</tr>
<?php } ?>
<?php } ?>
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
<div class="modal fade" id="removeCourseModal" tabindex="-1" role="dialog"
aria-labelledby="exampleModallabel" aria-hidden="true">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModallabel">Remove course from category
?</h5>
<button type="button" class="close" data-dismiss="modal" aria-label="Close
">
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<p>Are you sure you want to remove the course from this category?. Press
proceed to continue.</p>
<?php echo form_open('remove_course_from_category');?>
<input type="hidden" class="form-control" name="course_id" id="course_id">
<input type="hidden" class="form-control" name="category_id" id="
category_id" value="<?php echo $category_details['course_category_id
'];?>">
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary" data-dismiss="modal">Close
</button>
<input type="submit" class="btn btn-primary" value="Proceed"/></button>
</div>
<?php echo form_close();?>
</div>
</div>
</div>

```

Class Record View

```
<div id="page-wrapper" class="gray-bg">
```

```

<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
bottom: 0">
<div class="navbar-header">
<a class="navbar-minimalize minimize-styl-2 btn btn-primary" href="#"><i
class="fa fa-bars"></i></a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
;?></span>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-6">
<h2><?php echo $title.'_'.$course_details['section_name'];?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('')?>">Home</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('course_record_info'). $course_details
['course_id'];?>"><?php echo $course_details['course_short_name'].'_'
.$course_details['section_name'];?></a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="wrapper wrapper-content">
<div class="animated fadeInRightBig">
<div class="row">
<div class="col-lg-4 tooltip-demo">
<button class="btn btn-success btn-lg dim" type="button" data-toggle="
tooltip" id="saveTable" data-placement="right" title="" data-
original-title="Save Class Record"><i id="save_icon" class="fa fa-
save"></i></button>
<button onclick="location.href='<?php echo base_url('grades_stats/'.
$course_details['course_id']);?>' " class="btn btn-info btn-lg dim"
type="button" data-toggle="tooltip" data-placement="right" title=""
data-original-title="Stats"><i class="fa fa-bar-chart"></i></button>
</div>
<div class="col-lg-12">
<div class="ibox">
<div class="ibox-title">
<h5>Record - <?php echo $course_details['course_short_name'].'_'
.$course_details['section_name'];?>
</h5>
</div>
<div class="ibox-content" style="overflow: hidden">
<?php if(count($category_total) < 1){ ?>
<div class="well">

```

```

<h3>
No grading categories assigned yet
</h3>
</div>

<?php } else if (count($students) < 1) { ?>
<div class="well">
<h3>
No student assigned yet
</h3>
</div>

<?php } else { ?>
<?php echo form_open();?>
<div id="example">
</div>

<?php echo form_close(); ?>
<?php } ?>

<button style="visibility:hidden;" id="loadTable" name="loadTable"></
    button>

</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>
</div>

```

Course Announcements View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
    bottom:0">
<div class="navbar-header">
<a class="navbar-minimize minimize-styl-2 btn btn-primary" href="#"><i
    class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
    ;?></span>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>

<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-12">

```

```

<h2><?php echo $title.'_'.'_'.$course_details['course_short_name'].'_'.'
    $course_details['section_name'];?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('')?>">Home</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('course_record_info').'_'.
    $course_details
    ['course_id']?>"><?php echo $course_details['course_short_name'
    ];?></a></strong>
</li>
</ol>
</div>
</div>

<div class="row">
<div class="col-lg-12">
<div class="wrapper wrapper-content">
<div class="animated fadeInRightBig">
<div class="row">
<div class="col-lg-4 tooltip-demo">
<?php if($SESSION['logged_in']['user_role'] === "TEACHER") {?>
<button class="btn btn-primary" type="button" data-toggle="modal" data
    -target="#addAnnouncementModal"><i class="fa fa-plus"></i> Add
    announcement</button>
<?php } else if ($SESSION['logged_in']['user_role'] === "STUDENT") { ?>
<button class="btn btn-default dim btn-sm" type="button" onclick='location
    .href="<?php echo base_url('my_course_record/'.
    $course_details['
    course_id']);?>'> << Class Record</button>
<?php } ?>
</div>
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<h5>Announcements</h5>
</div>
<div class="ibox-content">
<table class="table table-striped table-bordered table-hover dataTable
    example">
<thead>
<tr>
<th>Subject</th>
<th>Date/Time Modified</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<?php $size = count($announcements);?>
<?php if(isset($announcements) && count($announcements) > 0) {?>
<?php for($a = 0; $a < $size; $a++){ ?>
<tr>
<td><?php echo $announcements[$a]['announcement_title'];?></td>
<td><span class="label label-primary"><?php echo $announcements[$a]['
    date_modified'];?></span></td>
<td><button type="button" onclick="location.href='<?php echo base_url('
    view_announcements/'.
    $course_details['course_id'].
    '/'.'_'.$announcements
    [$a]['announcement_id'];?>';" class="btn btn-sm btn-info dim m-t">
    View</button>
<?php if($SESSION['logged_in']['user_role'] === "TEACHER") {?>
<button class="btn btn-sm btn-danger dim m-t" data-toggle="modal" type="
    button" data-target="#deleteAnnouncementModal" data-ann_id='<?php_

```



```

<li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-6">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li>
<a>Courses</a>
</li>
<li>
<a>Course Category</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('course_category_form');?>">Add new
course category</a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<div class="ibox-tools">
<a class="collapse-link">
<i class="fa fa-chevron-up"></i>
</a>
</div>
</div>
<div class="ibox-content">
<?php if($this->session->flashdata('course_cat_registry_error')) { ?>
<div class="alert alert-danger"><strong>Error!</strong>
<p><?php echo $this->session->flashdata('course_cat_registry_error');?></p>
}
</div>
<?php } ?>
<?php echo form_open('add_course_category',array('class' => 'form-
horizontal'));?>
<input type="hidden" name="school_id" value="<?php echo $_SESSION['
logged_in']['school_id'];?>">
<div class="form-group"><label class="col-sm-2 control-label">Course
Category Name</label>
<div class="col-sm-6"><input type="text" name="course_category_name" class
="form-control" required="required"/></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group"><label class="col-sm-2 control-label">Assigned
Head</label>
<div class="col-sm-6">
<select class="form-control" name="assigned_head">
<?php if (isset($teacher)){ ?>
<?php for($a = 0; $a < count($teacher); $a++){ ?>
<option value='<?php echo $teacher[$a]['teacher_id'];?>'><?php echo
$teacher[$a]['user_first_name'].' '.$teacher[$a]['user_last_name'];
?></option>
<?php } ?>
<?php } ?>
<option value='0' selected> Not yet decided (default)</option>
</select>
</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group">
<label class="col-sm-2 control-label"></label>
<div class="col-sm-7">
<label>Select courses from the list on the left table that will be
assigned to this category</label><br/>
<a class="btn btn-sm btn-success" href="#" id="select_all">Select All</a>&
nbsp;
<a class="btn btn-sm btn-danger" href="#" id="deselect_all">Deselect All</
a>
<br/><br/>
<select id='searchable' multiple='multiple' name="category_courses[]">
<?php if (isset($courses)){ ?>
<?php for($i = 0; $i < count($courses); $i++){ ?>
<option value="<?php echo $courses[$i]['course_id'];?>">
<?php echo $courses[$i]['course_short_name'].' - Course ID: '.$courses[$i
]['course_id'];?>
</option>
<?php } ?>
<?php } else { ?>
<option value="0" disabled>No courses available.</option>

```

```

<?php } ?>
</select>
</div>
</div>
<div class="form-group">
<div class="col-sm-4 col-sm-offset-2">
<a class="btn btn-white" href="<?php echo base_url('course_category_list')
;?>">Cancel<a>
<input class="btn btn-primary" type="submit" value="Submit"/>
</div>
</div>
<?php echo form_close('');?>
</div>
</div>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>
</div>

```

Course Category List View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
bottom: 0">
<div class="navbar-header">
<a class="navbar-minimalize minimalize-styl-2 btn btn-primary" href="#"><
i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailbox.html">
<div>
<i class="fa fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>

```

```

</li>
</li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
</li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout')?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-6">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li class="active">
<a href="<?php echo base_url('')?>">Home</a>
</li>
<li>
Courses
</li>
<li class="active">
<strong><a href="<?php echo base_url('course_category_list')?>">View
course categories</a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<br/>
<div class="col-lg-10">
<?php if($this->session->flashdata('course_cat_registry_succeed')) { ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>New course category added!</p>
</div>
<?php } else if($this->session->flashdata('delete_course_category_succeed'
)) { ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>Course category deleted!</p>
</div>
<?php } ?>
</div>

```

```

<div class="col-lg-10">
<div class="ibox float-e-margins">
<div class="ibox-content">
<?php if(count($categories) > 0) {?>
<?php foreach($categories as $row){?>
<div class="well well-sm">
<p>
<h3>
&nbsp;<?php echo $row['course_category_name'];?>&nbsp;  
<a class="btn btn-white btn-sm" style="font-size:15px;" href="<?php echo
base_url('category_courses/'. $row['course_category_id']);?>"><i
class="fa fa-list"></i> See full list</a>
<a class="btn btn-white btn-sm" style="font-size:15px;" href="<?php echo
base_url('edit_course_category/'. $row['course_category_id']);?>"><i
class="fa fa-wrench"></i> Edit</a>
<a class="btn btn-white btn-sm" style="font-size:15px;" data-toggle="modal
" type="button" data-target="#deleteCourseCategoryModal" data-cc_id=
'<?php echo $row['course_category_id'];?>'><i class="fa fa-minus"></
i> Delete</a>
</h3>
<p>
</div>
<?php } ?>
<?php } ?>
</div>
</div>
</div>
</div>
<div class="footer">
<div>
<strong><i class="fa fa-copyright"></i> 2017</strong> Electronic Class
Record Inquiry System
</div>
</div>
</div>
</div>
<div class="modal fade" id="deleteCourseCategoryModal" tabindex="-1" role=
"dialog" aria-labelledby="addGradeItemModalLabel" aria-hidden="true"
>
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel">
Remove Course Category
<button type="button" class="close" data-dismiss="modal" aria-label="Close
">
<span aria-hidden="true">&times;</span>
</button>
</h5>
</div>
<div class="modal-body">
<?php $attributes = array('class' => 'form-horizontal'); ?>
<?php echo form_open('delete_course_category',$attributes);?>
<input type="hidden" name="cc_id" id="cc_id"/>
<p>Are you sure you want to delete this course category? Click confirm to
continue. </p>
<p></p>
<div class="modal-footer">
<input type="submit" class="btn btn-primary" value="Confirm"></input>

```

```

<button type="button" class="btn btn-default" id="disagreeButton" data-
dismiss="modal">Close</button>
<?php echo form_close();?>
</div>
</div>
</div>

```

Course Form View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
bottom:0">
<div class="navbar-header">
<a class="navbar-minimalize minimalize-styl-2 btn btn-primary" href="#"><
i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailbox.html">
<div>
<i class="fa fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</li>
</a>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>

```

```

</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-6">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li>
<a>Courses</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('course_form');?>">Add new course/s</a>
</strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<div class="ibox-tools">
<a class="collapse-link">
<i class="fa fa-chevron-up"></i>
</a>
</div>
</div>
<div class="ibox-content">
<?php if($this->session->flashdata('course_registry_error')) { ?>
<div class="alert alert-danger"><strong>Error!</strong>
<p><?php echo $this->session->flashdata('course_registry_error');?></p>
</div>
<?php } ?>
<?php echo form_open('add_course', array('class' => 'form-horizontal'));?>
<input type="hidden" name="school_id" value="<?php echo $_SESSION['logged_in']['school_id'];?>">
<div class="form-group"><label class="col-sm-2 control-label">Course Type
</label>
<div class="col-sm-10">
<div><label> <input type="radio" checked value="new" class="optionsRadios"
name="optionsRadios"> New </label></div>
<?php if(count($existing_course) > 0) {?>
<div><label> <input type="radio" value="existing" class="optionsRadios"
name="optionsRadios"> Existing </label></div>
<?php } else { ?>
<div><label> <input type="radio" value="existing" class="optionsRadios"
name="optionsRadios" disabled=""> Existing </label></div>
</div>
</div>
<div class="col-sm-2"><input type="text" name="course_full_name" class="form-control course_fields" required="required"/></div>
</div>
<div class="form-group"><label class="col-sm-2 control-label">Course Short Name</label>
<div class="col-sm-3"><input type="text" name="course_short_name" class="form-control course_fields" required="required"/></div>
</div>
</div>
<div class="existing_course">
<div class="form-group"><label class="col-sm-2 control-label">Select course:</label>
<div class="col-sm-8"><select class="form-control" name="existing_course">
<?php if(count($existing_course) > 0) {?>
<?php for($a = 0 ; $a < count($existing_course); $a++) {?>
<option value="<?=$existing_course[$a]['course_name_id'];?>"><?=$existing_course[$a]['course_full_name'].'_'. $existing_course[$a]['course_short_name'];?></option>
<?php } ?>
<?php } ?>
</select></div>
</div>
</div>
<div class="form-group"><label class="col-sm-2 control-label">Section</label>
<div class="col-sm-2"><input type="text" name="section_name" class="form-control" required="required"/></div>
</div>
<div class="form-group" id="data_5">
<label class="font-normal col-sm-2 control-label">Start Date - End Date</label>
<div class="col-sm-8">
<div class="input-daterange input-group" id="datepicker">
<input type="text" class="input-sm form-control" name="course_start" value="05/14/2014">
<span class="input-group-addon">to</span>
<input type="text" class="input-sm form-control" name="course_end" value="05/22/2014">
</div>
</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group"><label class="col-sm-2 control-label">Assigned Teacher</label>
<div class="col-sm-6">
<select class="form-control" name="assigned_teacher">
<?php if (isset($teacher)){ ?>
<?php $size = count($teacher);?>
<?php for($a = 0; $a < $size; $a++){ ?>
<option value='<?php echo $teacher[$a]['teacher_id'];?>'><?php echo $teacher[$a]['user_first_name'].'_'. $teacher[$a]['user_last_name'];?></option>
<?php } ?>
<?php } ?>
</div>
</div>

```

```

<option value='0' selected> Not yet decided (default)</option>
</select>
</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group"><label class="col-sm-2 control-label">Select
    Course category</label>
<div class="col-sm-6">
<select class="form-control" name="course_category">
<?php if (isset($course_category)){ ?>
<?php $size = count($course_category);?>
<?php for($a = 0; $a < $size; $a++){ ?>
<option value='<?php echo $course_category[$a]['course_category_id'];?>'
    <?php echo $course_category[$a]['course_category_name'];?></option>
<?php } ?>
<?php } ?>
<option value='0' selected> Not yet decided (default)</option>
</select>
</div>
</div>
<div class="hr-line-dashed"></div>
<div class="hr-line-dashed"></div>
<div class="form-group">
<div class="col-sm-4 col-sm-offset-2">
<a class="btn btn-white" href="<?php echo base_url('course_list');?>">
    Cancel<a>
<input class="btn btn-primary" type="submit" value="Submit"/>
</div>
</div>
</div>
<?php echo form_close('');?>
</div>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>

```

Course List View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-bottom: 0">
<div class="navbar-header">
<a class="navbar-minimalize minimalize-styl-2 btn btn-primary" href="#">
    i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>

```

```

</a>
<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailbox.html">
<div>
<i class="fa fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-4">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li class">
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li class">
<a href="#">Courses</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('course_list');?>">View list of
    courses</a></strong>

```

```

</li>
</ol>
</div>
</div>
<div class="wrapper_wrapper-content">
<div class="row">
<div class="col-lg-8">
<?php if($this->session->flashdata('course_registry_succeed')) { ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>New course added!</p>
</div>
<?php } else if($this->session->flashdata('mult_course_registry_succeed'))
    { ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>Multiple courses added complete!</p>
</div>
<?php } else if($this->session->flashdata('delete_course_succeed')) { ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>Delete course complete!</p>
</div>
<?php } ?>
</div>
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<h5>List of all courses in <?php echo $school_name;?></h5>
</div>
<div class="ibox-content tooltip-demo">
<table class="table table-striped table-bordered table-hover dataTable example" >
<thead>
<tr>
<th>Course Name</th>
<th>Section Name</th>
<th>Category Name</th>
<th>Assigned Teacher</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<?php $size = count($courses);?>
<?php if(isset($courses) && count($courses) > 0) {?>
<?php for($a = 0; $a < $size; $a++){ ?>
<tr>
<td><?php echo $courses[$a]['course_short_name'].','.$courses[$a]['course_full_name'].','?></td>
<td><?php echo $courses[$a]['section_name']?></td>
<?php if(strlen(utf8_decode($courses[$a]['course_category_name']))> 0) {?>
<td>
<?php echo $courses[$a]['course_category_name'];?>
</td>
<?php } else { ?>
<td><span class="label label-danger">No category assigned</span></td>
<?php } ?>
<?php if(strlen(utf8_decode($courses[$a]['user_first_name']))> 0) {?>
<td>
<?php echo $courses[$a]['user_first_name'].','.$courses[$a]['user_last_name'];?>
</td>
<?php } else { ?>
<td><span class="label label-danger">No teacher assigned</span></td>

```

```

<?php } ?>
<td>
<a class="btn btn-sm btn-success" data-toggle="tooltip" data-placement="top" title="" data-original-title="Update course information" href="#">
<?php echo base_url('edit_course_details/'. $courses[$a]['course_id'])?><i class="fa fa-edit"></i></a>&nbsp;&nbsp;&nbsp;<a class="btn btn-sm btn-primary" href="#">
<?php echo base_url('course_students/'. $courses[$a]['course_id'])?>" data-toggle="tooltip" data-placement="top" title="" data-original-title="Manage students"><i class="fa fa-user"></i></a>
&nbsp;&nbsp;&nbsp;<a class="btn btn-sm btn-danger" data-toggle="modal" data-target="#deleteCourseModal" data-course_id="#">
<?php echo $courses[$a]['course_id']?>"><i data-placement="top" data-toggle="tooltip" data-original-title="Delete Course" class="fa fa-times"></i></a>
&nbsp;&nbsp;&nbsp;<?php
$today = date("Y-m-d", strtotime("today"));
$end_date = date('Y-m-d', strtotime($courses[$a]['course_end_date']));
?>
<?php if($today > $end_date) { ?>
<a class="btn btn-sm btn-info" data-toggle="modal" data-target="#clearCourseRecordModal" data-course_id="#">
<?php echo $courses[$a]['course_id']?>"><i data-placement="top" data-toggle="tooltip" data-original-title="Clear class record" class="fa fa-refresh"></i></a>
<?php } ?>
</td>
</tr>
</tbody>
</table>
</div>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
<div class="modal fade id="deleteCourseModal" tabindex="-1" role="dialog" aria-labelledby="addGradeItemModalLabel" aria-hidden="true">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel">
Remove Course
<button type="button" class="close" data-dismiss="modal" aria-label="Close">
"
<span aria-hidden="true">&times;</span>
</button>
</h5>
</div>
<div class="modal-body">
<?php $attributes = array('class' => 'form-horizontal'); ?>
<?php echo form_open('delete_course',$attributes);?>
<input type="hidden" name="course_id" id="course_id" />

```

```

<p>Are you sure you want to delete this course? Click confirm to continue.
    </p>
<p></p>
<div class="modal-footer">
<input type="submit" class="btn btn-primary" value="Confirm"></input>
<button type="button" class="btn btn-default" id="disagreeButton" data-
    dismiss="modal">Close</button>
<?php echo form_close();?>
</div>
</div>
</div>
</div>
<div class="modal_fade" id="clearCourseRecordModal" tabindex="-1" role="
    dialog" aria-labelledby="addGradeItemModalLabel" aria-hidden="true">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel">
Clear Course Record
<button type="button" class="close" data-dismiss="modal" aria-label="Close
    ">
<span aria-hidden="true">&times;</span>
</button>
</h5>
</div>
<div class="modal-body">
<?php $attributes = array('class' => 'form-horizontal'); ?>
<?php echo form_open('clear_course_record',$attributes);?>
<input type="hidden" name="course_id" id="course_id" />
<p>Are you sure you want to clear the records of this course? All of the
    student grades and the grading categories/setting will be removed.
    Click confirm to continue. </p>
<p></p>
<div class="modal-footer">
<input type="submit" class="btn btn-primary" value="Confirm"></input>
<button type="button" class="btn btn-default" id="disagreeButton" data-
    dismiss="modal">Close</button>
<?php echo form_close();?>
</div>
</div>
</div>
</div>

```

Course Record Info View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
    bottom:0">
<div class="navbar-header">
<a class="navbar-minimize minimize-styl-2 btn btn-primary" href="#"><i
    class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
    ;?></span>
</li>
<li>
<a href="<?php echo base_url('logout');?>">

```

```

<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-12">
<h2><?php echo $title.'-'. $course_details['course_short_name'].'-'.
    $course_details['section_name'];?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('')?>">Home</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('course_record_info'). $course_details
    ['course_id'];?>"><?php echo $course_details['course_short_name']
    ;?></a></strong>
</li>
</ol>
</div>
<div class="row">
<div class="col-lg-12">
<div class="wrapper wrapper-content">
<div class="animated_fadeInRightBig">
<div class="row">
<div class="col-lg-4 tooltip-demo">
<button onclick="location.href='<?php echo base_url('class_record/').
    $course_details['course_id'];?>'" class="btn btn-primary btn-lg dim
    " type="button" data-toggle="tooltip" data-placement="right" title="
    " data-original-title="Modify class record"><i class="fa fa-th"></i>
</button>
<?php if($gradebook_details['allow_teacher'] === "YES" ){ ?>
<button onclick="location.href='<?php echo base_url('gradebook_setup/').
    $course_details['course_id'];?>'" class="btn btn-lg btn-danger dim
    " type="button" data-toggle="tooltip" data-placement="right" title="
    " data-original-title="Gradebook settings"><i class="fa fa-gear"></i>
</button>
<?php } ?>
<button onclick="location.href='<?php echo base_url('course_announcements
    /'). $course_details['course_id'];?>'" class="btn btn-warning btn-lg
    " type="button" data-toggle="tooltip" data-placement="right"
    title="" data-original-title="Course Announcements"><i class="fa fa-
    bullhorn"></i></button>
</div>
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<h5>Students</h5>
</div>
<div class="ibox-content">
<table class="table table-striped table-bordered table-hover dataTable">
<thead>
<tr>
<th>Student Last Name</th>
<th>Student First Name</th>
<th>Email Address</th>
<th>Email Verification</th>

```



```

<div class="text-center_link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row_wrapper border-bottom white-bg page-heading">
<div class="col-sm-12">
<h2><?php echo $title.'-'. $course_details['course_short_name'].'. ' .
    $course_details['section_name'].'. ' . $course_details['
    course_full_name'].'?></h2>
<ol class="breadcrumb">
<li class>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li class>
<a href="#">Course</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('course_students/'. $course_details['
    course_id']);?>">Course students list</a></strong>
</li>
</ol>
</div>
</div>
<div class="wrapper_wrapper-content">
<div class="row">
<div class="col-lg-8">
<?php if($this->session->flashdata('enrol_student_succeed')) { ??
<div class="alert alert-success"><strong>Success!</strong>
<p>New student added!</p>
</div>
<?php } else if($this->session->flashdata('enrol_student_group_succeed'))
    { ??
<div class="alert alert-success"><strong>Success!</strong>
<p>Upload students data complete!</p>
</div>
<?php } else if($this->session->flashdata('remove_student_succeed')) { ??
<div class="alert alert-danger"><strong>Success!</strong>
<p>The student is completely removed from this course!</p>
</div>
<?php } ??
<a href="<?php echo base_url('enrol_students/'. $course_details['course_id
    '])?>"><button class="btn btn-success btn-outline dim" type="button"
    ><i class="fa fa-plus"> Enroll student</i></button></a>
<a href="<?php echo base_url('enrol_student_groups/'. $course_details['
    course_id']);?>"><button class="btn btn-danger btn-outline dim" type=
    "button"><i class="fa fa-plus"> Enroll student groups</i></button></
    a>

```

```

<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel">Remove student from course
?</h5>
<button type="button" class="close" data-dismiss="modal" aria-label="Close
">
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<p>Are you sure you want to remove the student from this course? The
grades of the student in this course will be deleted. Press proceed
to continue.</p>
<?php echo form_open('remove_student');?>
<input type="hidden" class="form-control" name="student_id" id="student_id
">
<input type="hidden" class="form-control" name="course_id" id="course_id"
value="<?php echo $course_details['course_id'];?>">
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary" data-dismiss="modal">Close
</button>
<input type="submit" class="btn btn-primary" value="Proceed"/></button>
</div>
<?php echo form_close();?>
</div>
</div>
</div>

```

Edit Course Category View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
bottom:0">
<div class="navbar-header">
<a class="navbar-minimalize minimalize-styl-2 btn btn-primary" href="#">
i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailbox.html">
<div>
<i class="fa fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>

```

```

<a href="profile.html">
<div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-6">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li>
<a>Courses</a>
</li>
<li>
<a>Course Category</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('edit_course_category/'.
$category_details['course_category_id'];?>">Update course category
</a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">

```

```

<div class="ibox-tools">
<a class="collapse-link">
<i class="fa fa-chevron-up"></i>
</a>
</div>
</div>
<div class="ibox-content">
<?php if($this->session->flashdata('course_cat_update_error')) { ?>
<div class="alert alert-danger"><strong>Error!</strong>
<p><?php echo $this->session->flashdata('course_cat_update_error');?></p>
</div>
<?php } else if($this->session->flashdata('course_cat_update_succeed')) {
?>
<div class="alert alert-success"><strong>Success!</strong>
<p>Course category data updated!</p>
</div>
<?php } ?>
<?php echo form_open('update_course_category', array('class' => 'form-
horizontal'), array('course_category_id'=>$category_details['
course_category_id']));?>
<input type="hidden" name="school_id" value="<?php echo $_SESSION['
logged_in']['school_id'];?>">
<div class="form-group"><label class="col-sm-2 control-label">Course
Category Name</label>
<div class="col-sm-6"><input type="text" value="<?php echo $_
$category_details['course_category_name'];?>" name="
course_category_name" class="form-control" required="required"/></
div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group"><label class="col-sm-2 control-label">Assigned
Head</label>
<div class="col-sm-6">
<select class="form-control" name="assigned_head">
<?php if (isset($teacher)){ ?>
<?php $size = count($teacher);?>
<?php for($a = 0; $a < $size; $a++){ ?>
<?php if($teacher[$a]['teacher_id'] == $category_details['teacher_id'] &&
$category_details['teacher_id'] != 0) {?>
<option value='<?php echo $_teacher[$a]['teacher_id'];?>' selected><?php
echo $teacher[$a]['user_first_name']. ' ' . $teacher[$a]['
user_last_name']; ?></option>
<?php } else { ?>
<option value='<?php echo $_teacher[$a]['teacher_id'];?>'><?php echo
$teacher[$a]['user_first_name']. ' ' . $teacher[$a]['user_last_name'];
?></option>
<?php } ?>
<?php } ?>
<?php } ?>
<?php if($category_details['teacher_id'] == 0){ ?>
<option value='0' selected> Not yet decided (default)</option>
<?php } else { ?>
<option value='0'> Not yet decided (default)</option>
<?php } ?>
</select>
</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group">
<label class="col-sm-2 control-label"></label>

```

```

<div class="col-sm-7">
<label>Select courses from the list on the left table that will be
assigned to this category</label><br/>
<a class="btn btn-sm btn-success" href="#" id="select_all">Select All</a>&
nbsp;
<a class="btn btn-sm btn-danger" href="#" id="deselect_all">Deselect All</
a>
<br/><br/>
<select id='searchable' multiple='multiple' name="category_courses[]">
<?php if (isset($courses)){ ?>
<?php for($i = 0; $i < count($courses); $i++){ ?>
<?php if($courses[$i]['course_category_id'] != 0){ ?>
<?php if(in_array($courses[$i]['course_id'], $assigned_courses)){ ?>
<option value="<?php echo $_courses[$i]['course_id'];?>" selected>
<?php echo $courses[$i]['course_short_name']. ' - ' . $_courseID: ' . $courses [
$i]['course_id'];?>
</option>
<?php } ?>
<?php } else { ?>
<option value="<?php echo $_courses[$i]['course_id'];?>">
<?php echo $courses[$i]['course_short_name']. ' - ' . $_courseID: ' . $courses [
$i]['course_id'];?>
</option>
<?php } ?>
<?php } ?>
<option value="0" disabled>No courses available.</option>
<?php } ?>
</select>
</div>
</div>
<div class="form-group">
<div class="col-sm-4 col-sm-offset-2">
<a class="btn btn-white" href="<?php echo $_base_url('course_category_list')
;?>">Cancel<a>
<input class="btn btn-primary" type="submit" value="Submit"/>
</div>
<?php echo form_close('');?>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>

```

Edit Course Details View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
bottom: 0">
<div class="navbar-header">
<a class="navbar-minimize minimize-styl-2 btn btn-primary" href="#"><
i class="fa fa-bars"></i> </a>

```

```

</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailbox.html">
<div>
<i class="fa fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-6">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li>
<a>Courses</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('edit_course_details/' .
$course_details['course_id']);?>">Edit course details</a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<div class="ibox-tools">
<a class="collapse-link">
<i class="fa fa-chevron-up"></i>
</a>
</div>
</div>
<div class="ibox-content">
<?php if($this->session->flashdata('course_update_error')) { ?>
<div class="alert alert-danger"><strong>Error!</strong>
<p><?php echo $this->session->flashdata('course_update_error');?></p>
</div>
<?php } else if($this->session->flashdata('course_update_succeed')) { ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>Updating course data complete!</p>
</div>
<?php } else if($this->session->flashdata('clear_course_record_succeed'))
{ ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>Clearing Student Class record Succeed!</p>
</div>
<?php } ?>
<?php echo form_open('update_course', array('class' => 'form-horizontal',
array('course_id'=>$course_details['course_id']));?>
<input type="hidden" name="school_id" value="<?php echo $_SESSION['
logged_in']['school_id'];?>">
<div class="form-group"><label class="col-sm-2 control-label">Course Full
Name</label>
<div class="col-sm-8"><input type="text" value="<?php echo $course_details
['course_full_name'];?>" name="course_full_name" class="form-control
" required="required"/></div>
</div>
<div class="form-group"><label class="col-sm-2 control-label">Course Short
Name</label>
<div class="col-sm-3"><input type="text" value="<?php echo $course_details
['course_short_name'];?>" name="course_short_name" class="form-
control" required="required"/></div>
</div>
<div class="form-group"><label class="col-sm-2 control-label">Section</
label>
<div class="col-sm-2"><input type="text" value="<?php echo $course_details
['section_name'];?>" name="section_name" class="form-control"
required="required"/></div>
</div>

```

```

<?php
$start_date = date("m/d/Y", strtotime($course_details['course_start_date']
));
$end_date = date("m/d/Y", strtotime($course_details['course_end_date']));
?>
<div class="form-group" id="data_5">
<label class="font-normal,col-sm-2,control-label">Start Date - End Date</
label>
<div class="col-sm-8">
<div class="input-daterange,input-group" id="datepicker">
<input type="text" class="input-sm,form-control" name="course_start" value
="<?php echo $start_date;?>">
<span class="input-group-addon">to</span>
<input type="text" class="input-sm,form-control" name="course_end" value="
<?php echo $end_date;?>">
</div>
</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group"><label class="col-sm-2,control-label">Assigned
Teacher</label>
<div class="col-sm-6">
<select class="form-control" name="assigned_teacher">
<?php if (isset($teacher)){ ?>
<?php $size = count($teacher);?>
<?php for($a = 0; $a < $size; $a++){ ?>
<?php if($teacher[$a]['teacher_id'] == $course_details['teacher_id'] &&
$course_details['teacher_id'] != 0) {?>
<option value='<?php echo $teacher[$a]['teacher_id'];?>' selected><?php
echo $teacher[$a]['user_first_name'].'.'. $teacher[$a]['
user_last_name']; ?></option>
<?php } else { ?>
<option value='<?php echo $teacher[$a]['teacher_id'];?>'><?php echo
$teacher[$a]['user_first_name'].'.'. $teacher[$a]['user_last_name'];
?></option>
<?php } ?>
<?php } ?>
<?php } ?>
<?php if($course_details['teacher_id'] == 0){ ?>
<option value='0' selected> Not yet decided (default)</option>
<?php } else { ?>
<option value='0'> Not yet decided (default)</option>
<?php } ?>
</select>
</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group"><label class="col-sm-2,control-label">Select
Course category</label>
<div class="col-sm-6">
<select class="form-control" name="course_category">
<?php if (isset($course_category)){ ?>
<?php $size = count($course_category);?>
<?php for($a = 0; $a < $size; $a++){ ?>
<?php if($course_category[$a]['course_category_id'] == $course_details['
course_category_id'] && $course_details['course_category_id'] != 0)
{?>
<option value='<?php echo $course_category[$a]['course_category_id'];?>'
selected><?php echo $course_category[$a]['course_category_name'
];?></option>

```

```

<?php } else { ?>
<option value='<?php echo $course_category[$a]['course_category_id'];?>'
><?php echo $course_category[$a]['course_category_name'];?></option>
<?php } ?>
<?php } ?>
<?php } ?>
<?php if($course_details['course_category_id'] == 0){ ?>
<option value='0' selected> Not yet decided (default)</option>
<?php } else { ?>
<option value='0'> Not yet decided (default)</option>
<?php } ?>
</select>
</div>
</div>
<div class="hr-line-dashed"></div>
<div class="hr-line-dashed"></div>
<div class="form-group">
<div class="col-sm-4,col-sm-offset-2">
<?php if($this->session->tempdata('go_to_category')) { ?>
<a class="btn,btn-white" href="<?php echo base_url('category_courses/'.
$this->session->tempdata('go_to_category'));?>">Go Back to list<a>
<?php } else { ?>
<a class="btn,btn-white" href="<?php echo base_url('course_list');?>">Go
Back to list<a>
<?php } ?>
<input class="btn,btn-primary" type="submit" value="Submit"/>
</div>
</div>
<?php echo form_close('');?>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>

```

Edit Student Details View

```

<div id="page-wrapper" class="gray-bg">
<div class="row,border-bottom">
<nav class="navbar,navbar-static-top,navbar" role="navigation" style="margin-
bottom:0">
<div class="navbar-header">
<a class="navbar-minimize,minimize-styl-2,btn,btn-primary" href="#"><
i class="fa,fa-bars"></i> </a>
</div>
<ul class="nav,navbar-top-links,navbar-right">
<li>
<span class="m-r-sm,text-muted,welcome-message"><?php echo $school_name
;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle,info" data-toggle="dropdown" href="#">
<i class="fa,fa-bell"></i> <span class="label,label-primary">8</span>
</a>

```

```

<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailto:mail.html">
<div>
<i class="fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-6">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li>
<a>Accounts</a>
</li>
<li>
<a>Students</a>
</li>
<li class="active">

```

```

<strong><a href="<?php echo base_url('edit_student_details/'.
    $student_details['student_id'];?>">Edit student details</a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<div class="ibox-tools">
<a class="collapse-link">
<i class="fa-chevron-up"></i>
</a>
</div>
</div>
<div class="ibox-content">
<?php if($this->session->flashdata('update_student_error')) { ?>
<div class="alert alert-danger"><strong>Error!</strong>
<p><?php echo $this->session->flashdata('update_student_error');?></p>
</div>
<?php } else if($this->session->flashdata('update_student_succeed')) { ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>Update student data complete!</p>
</div>
<?php } ?>
<?php echo form_open('update_student',array('class' => 'form-horizontal'),
    array('student_id' => $student_details['student_id'],'school_id' =>
    $_SESSION['logged_in']['school_id']));?>
<div class="form-group"><label class="col-sm-2 control-label">First Name</
    label>
<div class="col-sm-6"><input type="text" value="<?php echo
    $student_details['user_first_name'];?>" name="student_first_name"
    class="form-control" required="required"/></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group"><label class="col-sm-2 control-label">Last Name</
    label>
<div class="col-sm-6"><input type="text" value="<?php echo
    $student_details['user_last_name'];?>" name="student_last_name" class
    ="form-control" required="required"/></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group"><label class="col-sm-2 control-label">Email
    Address</label>
<div class="col-sm-6"><input type="text" value="<?php echo
    $student_details['user_email_address'];?>" name="
    student_email_address" class="form-control" required="required"/></
    div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group">
<label class="col-sm-2 control-label"></label>
<div class="col-sm-7">
<label>Select courses from the list on the left table in which the student
    should be assigned</label><br/>
<a class="btn btn-sm btn-success" href="#" id="select_all">Select All</a>&
    nbsp;

```

```

<a class="btn btn-sm btn-danger" href="#" id="deselect_all">Deselect All</a>
<br/><br/>
<select id='searchable' multiple='multiple' name="available_courses[]">
<?php if (isset($courses)){ ?>
<?php for($i = 0; $i < count($courses); $i++){ ?>
<?php if (in_array($courses[$i]['course_id'],$assigned_courses)){ ?>
<option value="<?php echo $courses[$i]['course_id'];>" selected>
<?php echo $courses[$i]['course_short_name'].'.'. $courses[$i]['
section_name'].'_CourseID:.'. $courses[$i]['course_id'];?>
</option>
<?php } else { ?>
<option value="<?php echo $courses[$i]['course_id'];>">
<?php echo $courses[$i]['course_short_name'].'.'. $courses[$i]['
section_name'].'_CourseID:.'. $courses[$i]['course_id'];?>
</option>
<?php } ?>
<?php } ?>
<?php } else { ?>
<option value="0" disabled>No courses available.</option>
<?php } ?>
</select>
</div>
</div>
<div class="form-group">
<div class="col-sm-4 col-sm-offset-2">
<a class="btn btn-white" href="<?php echo base_url('student_list');?>">Go
back to List<a>
<input class="btn btn-primary" type="submit" value="Update"/>
</div>
</div>
<?php echo form_close('');?>
</div>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>

```

Edit Student Group View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-bottom:0">
<div class="navbar-header">
<a class="navbar-minimalize minimalize-styl-2 btn btn-primary" href="#">
<i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
;?></span>
</li>
<li class="dropdown">

```

```

<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailbox.html">
<div>
<i class="fa fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-6">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li>
<a>Student</a>
</li>
<li class="active">

```

```

<strong><a href="<?php echo base_url('edit_student_group/'.$group_details
    ['student_group_id']);?>">Update student group</a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<div class="ibox-tools">
<a class="collapse-link">
<i class="fa fa-chevron-up"></i>
</a>
</div>
</div>
<div class="ibox-content">
<?php if($this->session->flashdata('update_student_group_error')) { ?>
<div class="alert alert-danger"><strong>Error!</strong>
<p><?php echo $this->session->flashdata('update_student_group_error');?></p>
</div>
<?php } else if($this->session->flashdata('update_student_group_succeed'))
    { ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>Student group data updated!</p>
</div>
<?php } ?>
<?php echo form_open('update_student_group', array('class' => 'form-
    horizontal'), array('student_group_id'=>$group_details['
    student_group_id']));?>
<input type="hidden" name="school_id" value="<?php echo $SESSION['
    logged_in']['school_id'];?>">
<div class="form-group"><label class="col-sm-2 control-label">Student
    Group Name</label>
<div class="col-sm-6"><input type="text" value="<?php echo $group_details
    ['student_group_name'];?>" name="student_group_name" class="form-
    control" required="required"/></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group">
<label class="col-sm-2 control-label"></label>
<div class="col-sm-7">
<label>Select students from the left table that will be added to the group
    </label><br/>
<a class="btn btn-sm btn-success" href="#" id="select_all">Select All</a>&
    nbsp;
<a class="btn btn-sm btn-danger" href="#" id="deselect_all">Deselect All</
    a>
<br/><br/>
<select id='searchable' multiple='multiple' name="group_members[]">
<?php if (isset($students)){ ?>
<?php for($i = 0; $i < count($students); $i++){ ?>
<?php if(in_array($students[$i]['student_id'],$assigned_students)){ ?>
<option value="<?php echo $students[$i]['student_id'];?>" selected>
<?php echo $students[$i]['user_first_name'].' ' . $students[$i]['
    user_last_name'];?>
</option>
<?php } else { ?>
<option value="<?php echo $students[$i]['student_id'];?>">

```

```

<?php echo $students[$i]['user_first_name'].' ' . $students[$i]['
    user_last_name'];?>
</option>
<?php } ?>
<?php } ?>
<?php } else { ?>
<option value="0" disabled>No students available.</option>
<?php } ?>
</select>
</div>
</div>
<div class="form-group">
<div class="col-sm-4 col-sm-offset-2">
<a class="btn btn-white" href="<?php echo base_url('student_groups_list')
    ;?>">Go back to list<a>
<input class="btn btn-primary" type="submit" value="Update"/>
</div>
</div>
<?php echo form_close('');?>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>

```

Edit Teacher Details View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
    bottom: 0">
<div class="navbar-header">
<a class="navbar-minimize minimize-styl-2 btn btn-primary" href="#"><
    i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
    ;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailbox.html">
<div>
<i class="fa fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>

```



```

<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-6">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li>
<a>Accounts</a>
</li>
<li>
<a>Teachers</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('edit_teacher_details/'.
    $teacher_details['teacher_id']);?>">Edit teacher details</a></strong>
    >
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<div class="ibox-tools">
<a class="collapse-link">
<i class="fa fa-chevron-up"></i>
</a>
</div>
</div>
<div class="ibox-content">
<?php if($this->session->flashdata('update_teacher_error')) { ?>
<div class="alert alert-danger"><strong>Error!</strong>
<p><?php echo $this->session->flashdata('update_teacher_error');?></p>
</div>
<?php } else if($this->session->flashdata('update_teacher_succeed')) { ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>Update teacher data complete!</p>
</div>
<?php } ?>
<?php echo form_open('update_teacher',array('class' => 'form-horizontal'),
    array('teacher_id'=> $teacher_details['teacher_id'],'school_id'=>
    $_SESSION['logged_in']['school_id']));?>
<div class="form-group"><label class="col-sm-2 control-label">First Name</label>
<div class="col-sm-6"><input type="text" value="<?php echo $teacher_details['user_first_name'];?>" name="teacher_first_name"
    class="form-control" required="required"/></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group"><label class="col-sm-2 control-label">Last Name</label>
<div class="col-sm-6"><input type="text" value="<?php echo $teacher_details['user_last_name'];?>" name="teacher_last_name"
    class="form-control" required="required"/></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group"><label class="col-sm-2 control-label">Email
    Address</label>
<div class="col-sm-6"><input type="text" value="<?php echo $teacher_details['user_email_address'];?>" name="
    teacher_email_address" class="form-control" required="required"/></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group">
<label class="col-sm-2 control-label"></label>
<div class="col-sm-7">
<label>Select courses from the list on the left table in which the teacher
    should be assigned</label><br>
<a class="btn btn-sm btn-success" href="#" id="select_all">Select All</a>&
    nbsp;
<a class="btn btn-sm btn-danger" href="#" id="deselect_all">Deselect All</a>
<br><br>
<select id='searchable' multiple='multiple' name="available_courses[]">
<?php if (isset($courses)){ ?>
<?php for($i = 0; $i < count($courses); $i++){ ?>
<?php if($courses[$i]['teacher_id'] != 0){ ?>
<?php if(in_array($courses[$i]['course_id'],$assigned_courses)){ ?>
<option value="<?php echo $courses[$i]['course_id'];?>" selected>
<?php echo $courses[$i]['course_short_name'].' - Course ID: '.$courses[$i]['course_id'];?>

```

```

</option>
<?php } ?>
<?php } else { ?>
<option value="<?php echo $courses[$i]['course_id'];>">
<?php echo $courses[$i]['course_short_name'].'_CourseID:'. $courses[
    $i]['course_id'];?>
</option>
<?php } ?>
<?php } ?>
<?php } else { ?>
<option value="0" disabled>No courses available.</option>
<?php } ?>
</select>
</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group">
<div class="col-sm-4 col-sm-offset-2">
<a class="btn btn-white" href="<?php echo base_url('teacher_list');?>">Go
    back to List<a>
<input class="btn btn-primary" type="submit" value="Submit"/>
</div>
</div>
<?php echo form_close('');?>
</div>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>
</div>

```

```

<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-12">
<h2><?php echo $title.'_'. $course_details['course_short_name'].'_(' .
    $course_details['course_full_name'].')';?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li>
<a>Course</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('enrol_student_groups/'.
    $course_details['course_id'].')';?>">Enrol students</a></strong>
</li>
</ol>
</div>
</div>
<div class="row">

```

Enrol Student Group View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
    bottom: 0">
<div class="navbar-header">
<a class="navbar-minimalize minimalize-styl-2 btn btn-primary" href="#"><
    i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
    ;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailbox.html">
<div>
<i class="fa fa-envelope fa-fw"></i> You have 16 messages

```

```

<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<div class="ibox-tools">
<a class="collapse-link">
<i class="fa fa-chevron-up"></i>
</a>
</div>
</div>
<div class="ibox-content">
<?php echo form_open('assign_student_groups',array('class' => 'form-
horizontal'),array('course_id'=>$course_details['course_id']));?>
<input type="hidden" name="school_id" value="<?php echo $_SESSION['
logged_in']['school_id']?>">
<div class="form-group">
<label class="col-sm-2 control-label"></label>
<div class="col-sm-7">
<label>Select student groups from the left table that will be enrolled in
this course</label><br/>
<a class="btn btn-sm btn-success" href="#" id="select_all">Select All</a>&
nbsp;
<a class="btn btn-sm btn-danger" href="#" id="deselect_all">Deselect All</
a>
<br/><br/>
<select id='searchable' multiple='multiple' name="student_groups[]">
<?php if (isset($student_groups)){ ?>
<?php for($i = 0; $i < count($student_groups); $i++){ ?>
<option value="<?php echo $student_groups[$i]['student_group_id'];>">
<?php echo $student_groups[$i]['student_group_name'].'_>
$student_groups[$i]['student_members'].'_members';?>
</option>
<?php } ?>
<?php } else { ?>
<option value="0" disabled>No students available.</option>
<?php } ?>
</select>
</div>
</div>
<div class="form-group">
<div class="col-sm-4 col-sm-offset-2">
<a class="btn btn-white" href="<?php echo base_url('course_students/' .
$course_details['course_id']);?>">Go back to list<a>
<input class="btn btn-primary" type="submit" value="Enrol"/>
</div>
</div>
<?php echo form_close('');?>
</div>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>

```

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
bottom:0">
<div class="navbar-header">
<a class="navbar-minimalize minimize-styl-2 btn btn-primary" href="#"><
i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailbox.html">
<div>
<i class="fa fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>

```

Enrol Student View

```

</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-12">
<h2><?php echo $title.' - '. $course_details['course_short_name']. ' - ' .
    $course_details['course_full_name']. ' ';?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li>
<a>Course</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('enrol_students/'. $course_details['
    course_id'];?>">Enrol students</a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<div class="ibox-tools">
<a class="collapse-link">
<i class="fa fa-chevron-up"></i>
</a>
</div>
</div>
<div class="ibox-content">
<?php echo form_open('assign_students', array('class' => 'form-horizontal'
    , array('course_id' => $course_details['course_id']));?>
<input type="hidden" name="school_id" value="<?php echo $_SESSION['
    logged_in']['school_id'];?>">
<div class="form-group">
<label class="col-sm-2 control-label"></label>
<div class="col-sm-7">
<label>Select students from the left table that will be added to the group
</label><br/>
<a class="btn btn-sm btn-success" href="#" id="select_all">Select All</a>&
    nbsp;
<a class="btn btn-sm btn-danger" href="#" id="deselect_all">Deselect All</
    a>
<br/><br/>
<select id='searchable' multiple='multiple' name="selected_students[]">
<?php if (isset($students)){ ?>
<?php for($i = 0; $i < count($students); $i++){ ?>
<?php if (!(in_array($students[$i]['student_id'], $assigned_students))){ ?>
<option value="<?php echo $students[$i]['student_id'];?>">
<?php echo $students[$i]['user_first_name']. ' - ' . $students[$i]['
    user_last_name'];?>
</option>
<?php } ?>
<?php } ?>
<?php } else { ?>
<option value="0" disabled>No students available.</option>
<?php } ?>
</select>

```

```

</div>
</div>
<div class="form-group">
<div class="col-sm-4 col-sm-offset-2">
<a class="btn btn-white" href="<?php echo base_url('course_students/'.
    $course_details['course_id']);?>">Go back to list<a>
<input class="btn btn-primary" type="submit" value="Enrol"/>
</div>
</div>
<?php echo form_close('');?>
</div>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>

```

Force Password View

```

<div id="preloader"></div>
<div id="top"></div>
<div class="landing">
<div class="overlay">
<div class="container">
<div class="row">
<!-- /.signup form -->
<div class="col-md-offset-3 col-md-6">
<div class="logo wow"><a href="">&nbsp;Electronic Class
    Record Inquiry System</a></div>
<div class="signup-header wow">
<h3 class="form-title text-center">You are now logged in to the system!</
    h3>
<ul><li style="color:white;">But, we require you to change your password
    upon logging in to the system for the first time. Thank you very
    much for your cooperation.</li></ul>
<?php if($this->session->flashdata('force_update_error')) { ?>
<div class="alert alert-danger" role="alert">
<strong>Error!</strong> <?php echo $this->session->flashdata('
    force_update_error'); ?>
</div>
<?php } ?>
<?php echo form_open('update_force_password');?>
<p>Username: <a><?php echo $user_details['user_name'];?></a><br/>
<input name="user_name" type="hidden" value="<?php echo $user_details['
    user_name'];?>" />
Email Address: <a><?php echo $user_details['user_email_address'];?></a><p>
<input name="user_email_address" type="hidden" value="<?php echo $
    $user_details['user_email_address'];?>" />
<div class="form-group">
<input class="form-control input-lg" onkeyup="checkPass(); return false;"
    name="user_password" id="password" type="password" placeholder="
    Password" required />
</div>

```



```

<div class="row_border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
    bottom:0">
<div class="navbar-header">
<a class="navbar-minimize minimize-styl-2 btn btn-primary" href="#"><
    i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
    ;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailbox.html">
<div>
<i class="fa fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-12">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li class>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li class>
<a href="#">Grade Settings</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('scales_list');?>">List of gradebook
    scales</a></strong>
</li>
</ol>
</div>
</div>
<div class="wrapper wrapper-content">
<div class="row">
<div class="col-lg-4">
<div class="ibox float-e-margins">
<div class="ibox-content">
<table class="table table-striped table-bordered table-hover dataTables-
    example" >
<thead>
<tr>
<th>Scale Name</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<?php if(isset($scales) && count($scales) > 0) {?>
<?php $size = count($scales);?>
<?php for($a = 0; $a < $size; $a++){ ?>
<tr>
<td class="col-md-6"><?php echo $scales[$a]['scale_template_name']?></td>
<td class="col-md-6"><button type="button" class="btn btn-sm btn-primary"
    data-toggle="modal" data-target="#changeScaleNameModal" data-
    template_id="<?php echo $scales[$a]['scale_template_id'];?>" data-
    template_name="<?php echo $scales[$a]['scale_template_name'];?>"><i
    class="fa fa-edit"></i> Change Name </button>&nbsp;<button type="
    button" class="btn btn-sm btn-danger" data-toggle="modal" data-
    target="#removeScaleModal" data-template_id="<?php echo $scales[$a
    ]['scale_template_id'];?>"><i class="fa fa-minus"></i> Remove</
    button></td>
</tr>
<?php } ?>
<?php } else { ?>
<tr>
<td>No grading scales found</td>
<td></td>
</tr>
<?php } ?>
</tbody>
</table>
</div>
</div>

```



```

</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>

<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-12">
<h2><?php echo $title.' - '. $course_details['course_short_name'].' - '.
    $course_details['section_name'].' - '. $course_details['
        course_full_name'].'?></h2>
<ol class="breadcrumb">
<li class="active">
<a href="<?php echo base_url('')?>">Home</a>
</li>
<li class="active">
Grade Settings
</li>
<li class="active">
<strong><a href="<?php echo base_url('gradebook_setup/'. $course_details['
        course_id'])?>">Gradebook Setup</a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="wrapper wrapper-content">
<div class="row">
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<h5>Gradebook - <?php echo $course_details['course_short_name'].' - '.
    $course_details['section_name'];?></h5>
</div>
<div class="ibox-content">
<div>
<h5> - Click on the double arrow button to view/hide grade items </h5>
<h5> - To add grade items in each of the main category, Click the Edit
    button assigned to each category</h5>
<h4> Method of Computation: <?php echo $gradebook_details['
    method_of_computation'];?></h4>
<a href="<?php echo base_url('modify_gradebook_categories/'.
    $gradebook_details['gradebook_setup_details_id'])?>" class="btn btn
    -primary btn-outline"><i class="fa fa-edit"></i> Modify main
    categories </a>
</div>
<div class="table-responsive">
<?php echo form_open('move_gradebook_items');?>
<table class="table table-striped" style="border-collapse: collapse;">
<thead>
<tr><th>&nbsp;</th>
<th>Main Category/Item Name</th>

```



```

<td class="col-md-3 text-center"><?php echo $category_items[$index2]['
    grade_item_name']?></td>
    main_category_id'];?>" data-setup_id="<?php echo $gradebook_details
    ['gradebook_setup_details_id']?>" class="btn btn-sm"><i class="
<td class="col-md-3 text-center"><?php echo $category_items[$index2]['
    grade_item_weight']. '%'. ($category_items[$index2]['weight_to_total'
    ] . '%')?></td>
    fa fa-minus"></i> Remove</button></td>
</tr>
<td class="col-md-3 text-center"><?php echo $category_items[$index2]['
    total_points']?></td>
    <?php } ?>
<td class="col-md-3 text-center"><?php echo $category_items[$index2]['
    total_points']?></td>
    <?php } ?>
<td class="col-md-3 text-center"><a class="btn btn-sm btn-info" data-
    toggle="modal" data-target="#editGradeItemModal" data-category_name=
    "<?php echo $gradebook_categories[$index]['main_category_name'];?>"
    data-category_id=" <?php echo $gradebook_categories[$index]['
    main_category_id'];?>" data-grade_item_id="<?php echo
    $category_items[$index2]['grade_item_id']?>" data-grade_item_name="
    <?php echo $category_items[$index2]['grade_item_name']?>" data-
    points="<?php echo $category_items[$index2]['total_points']?>" data-
    grade_item_names='<?php echo json_encode($grade_item_names)?>' data-
    grade_item_categories = '<?php echo json_encode(
    $grade_item_categories)?>'>
    <tr>
    <th &nbsp;</th>
    <th></th>
    <th></th>
    <th><?php echo $gradebook_details['max_grade']. '%';?></th>
    <th><?php echo $gradebook_details['total_points'];?></th>
    <th></th>
    </tr>
    </tbody>
    </table>
    <input type="hidden" name="gradebook_id" value="<?php echo
    $gradebook_details['gradebook_setup_details_id'];?>"/>
    <div class="form-group">
    <label id="move_to_label">Move to: &nbsp;</label><select name="
    move_to_category" id="move_to_category">
    <?php for($index = 0; $index < $size; $index++){ ?>
    <?php if($gradebook_categories[$index]['main_category_type'] == "CATEGORY"
    ){ ?>
    <option value="<?php echo $gradebook_categories[$index]['main_category_id
    ']?>"><?php echo $gradebook_categories[$index]['main_category_name'
    ]?></option>
    <?php } ?>
    <?php } ?>
    <option value="0"><?php echo $gradebook_details['gradebook_setup_name'
    ]?></option>
    </select>
    </div>
    <div class="form-group">
    <input class="btn btn-info" type="submit" name="submit" id="
    move_item_submit" value="Save Changes"/>
    </div>
    <?php echo form_close();?>
    </div>
    </div>
    <div class="col-lg-12">
    <div class="wrapper wrapper-content">
    <div class="row">
    <div class="col-lg-5">
    <div class="ibox float-e-margins">
    <div class="ibox-title">
    <h5><i class="fa fa-list"></i> Grading Scale</h5>
    <div class="ibox-tools">
    </div>
    <div class="ibox-content">
    <div>
    <table class="table table-sm">
    <thead>
    <th colspan="4"><button type="button" class="btn btn-sm btn-outline btn-
    success" data-toggle="modal" data-target="#saveCurrentScaleModal"><i
    class="fa fa-floppy-o"></i> Save current</button>&nbsp;<button type

```

```

        ="button" class="btn btn-sm btn-outline btn-danger" data-toggle="
        modal" data-target="#useExistingScaleModal"><i class="fa fa-arrow-
        down"></i> Use existing</button></th>
</thead>
<tbody>
<tr>
<th>Scale Labels</th>
<th colspan="3">Scale Values</th>
</tr>
<?php foreach($grading_scales as $scale){?>
<tr>
<td><?php echo $scale['scale_values_label']?></td>
<td><?php echo number_format($scale['scale_values_lower_limit'],2, '.', ' ');?></td>
<td>-</td>
<td><?php echo number_format($scale['scale_values_higher_limit'],2, '.', ' ');?></td>
</tr>
<?php } ?>
</tbody>
</table>
<a class="btn btn-primary btn-outline btn-block m-t" href=" <?php echo
        base_url('update_scale/'. $gradebook_details['course_id'] )?>"><i
        class="fa fa-list-alt"></i> Update scale</a>
</div>
</div>
</div>
</div>
<div class="col-lg-offset-1 col-lg-6">
<div class="ibox float-e-margins">
<div class="ibox-title">
<h5><i class="fa fa-gear"></i> Settings</h5>
<div class="ibox-tools">
</div>
</div>
<div class="ibox-content">
<div>
<ul class="list-group">
<li class="list-group-item">
<?php if($gradebook_details['allow_teacher'] === "YES"){ ?>
<span class="badge badge-info">Allowed</span>
<?php } else { ?>
<span class="badge badge-danger">Not Allowed</span>
<?php } ?>
Allow teachers to update?</li>
</ul>
<button data-toggle="modal" data-target="#updateSettingsModal" data-
        whatever=<?php echo $course_details['course_id']?> class="btn btn-
        primary btn-outline btn-block m-t"><i class="fa fa-gear"></i> Update
        settings</button>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="footer">
</div>

```

```

<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
<div class="modal fade" id="editGradeItemModal" tabindex="-1" role="dialog"
        aria-labelledby="addGradeItemModalLabel" aria-hidden="true">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel">Edit Grade Item
<button type="button" class="close" data-dismiss="modal" aria-label="Close"
        ">
<span aria-hidden="true">&times;</span>
</button>
</h5>
</div>
<div class="modal-body">
<div class="container-fluid">
<div class="col-md-12">
<?php $attributes = array('class' => 'form-horizontal'); ?>
<?php echo form_open('edit_grade_item', $attributes);?>
<input type="hidden" name="course_id" value=" <?php echo $gradebook_details
        ['course_id']?>" />
<input type="hidden" name="setup_id" value=" <?php echo $gradebook_details
        ['gradebook_setup_details_id']?>" />
<input type="hidden" name="grade_item_id" id="grade_item_id" />
<input type="hidden" name="category_id" id="category_id" />
<div class="form-group">
<label for="recipient-name" class="form-control-label">Grade Item Name: </
        label>
<input type="text" class="form-control" required="required" name="
        grade_item_name" id="grade_item_name">
</div>
<div class="form-group">
<label for="recipient-name" class="form-control-label">Item Points: </label
        >
<input type="text" class="form-control" required="required" name="
        total_points" id="total_points">
</div>
</div>
</div>
</div>
<div class="modal-footer">
<input type="submit" id="edit_grade_item_button" class="btn btn-primary"
        value="Update"></input>
<button type="button" class="btn btn-default" data-dismiss="modal">Close
        </button>
<?php echo form_close();?>
</div>
</div>
</div>
</div>
<div class="modal fade" id="editGradeCategoryItemModal" tabindex="-1" role
        ="dialog" aria-labelledby="addGradeItemModalLabel" aria-hidden="true"
        ">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel">
<button type="button" class="close" data-dismiss="modal" aria-label="Close"
        ">

```

```

<span aria-hidden="true">&times;</span>
</button>
</h5>
</div>
<div class="modal-body">
<div class = "container-fluid">
<div class = "col-md-12">
<?php $attributes = array('class' => 'form-horizontal'); ?>
<?php echo form_open('edit_gradebook_item',$attributes);?>
<input type="hidden" name="course_id" value="<?php echo $gradebook_details
['course_id']?>"/>
<input type="hidden" name="category_id" id="category_id"/>
<input type="hidden" name="setup_id" id="setup_id"/>
<div class="form-group">
<label for="recipient-name" class="form-control-label">Grade Item Name:</
label>
<input type="text" class="form-control" required="required" name="
category_name" id="category_name">
</div>
<div class="form-group">
<label for="recipient-name" class="form-control-label">Item Points:</label
>
<input type="text" class="form-control" required="required" name="
total_points" id="total_points2">
</div>
</div>
</div>
<div class="modal-footer">
<input type="submit" id="edit_grade_item_button1" class ="btn btn-primary
" value = "Update"></input>
<button type="button" class="btn btn-default" id="disagreeButton" data-
dismiss="modal">Close</button>
<?php echo form_close();?>
</div>
</div>
</div>
<div class="modal_fade" id="removeGradeItemModal" tabindex="-1" role="
dialog" aria-labelledby="addGradeItemModalLabel" aria-hidden="true">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel">
<button type="button" class="close" data-dismiss="modal" aria-label="Close
">
<span aria-hidden="true">&times;</span>
</button>
</h5>
</div>
<div class="modal-body">
<?php $attributes = array('class' => 'form-horizontal'); ?>
<?php echo form_open('remove_grade_item',$attributes);?>
<input type="hidden" name="course_id" value="<?php echo $gradebook_details
['course_id']?>"/>
<input type="hidden" name="setup_id" value="<?php echo $gradebook_details
['gradebook_setup_details_id']?>"/>
<input type="hidden" name="grade_item_id" id="grade_item_id"/>
<input type="hidden" name="category_id" id="category_id"/>
<p>Are you sure you want to remove this gradebook item?</p>
<p></p>

```

```

<div class="modal-footer">
<input type="submit" class ="btn btn-primary" value = "Remove"></input>
<button type="button" class="btn btn-default" id="disagreeButton" data-
dismiss="modal">Close</button>
<?php echo form_close();?>
</div>
</div>
</div>
<div class="modal_fade" id="removeGradebookItemModal" tabindex="-1" role="
dialog" aria-labelledby="addGradeItemModalLabel" aria-hidden="true">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel">
<button type="button" class="close" data-dismiss="modal" aria-label="Close
">
<span aria-hidden="true">&times;</span>
</button>
</h5>
</div>
<div class="modal-body">
<?php $attributes = array('class' => 'form-horizontal'); ?>
<?php echo form_open('remove_gradebook_item',$attributes);?>
<input type="hidden" name="course_id" value="<?php echo $gradebook_details
['course_id']?>"/>
<input type="hidden" name="setup_id" id="setup_id"/>
<input type="hidden" name="category_id" id="category_id"/>
<p>Are you sure you want to remove this gradebook item?</p>
<p></p>
<div class="modal-footer">
<input type="submit" class ="btn btn-primary" value = "Remove"></input>
<button type="button" class="btn btn-default" id="disagreeButton" data-
dismiss="modal">Close</button>
<?php echo form_close();?>
</div>
</div>
</div>
<div class="modal_fade" id="removeGradebookCategoryModal" tabindex="-1"
role="dialog" aria-labelledby="addGradeItemModalLabel" aria-hidden="
true">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel">
<button type="button" class="close" data-dismiss="modal" aria-label="Close
">
<span aria-hidden="true">&times;</span>
</button>
</h5>
</div>
<div class="modal-body">
<?php $attributes = array('class' => 'form-horizontal'); ?>
<?php echo form_open('remove_gradebook_category',$attributes);?>
<input type="hidden" name="course_id" value="<?php echo $gradebook_details
['course_id']?>"/>
<input type="hidden" name="setup_id" id="setup_id"/>
<input type="hidden" name="category_id" id="category_id"/>

```



```

</button>
</h5>
</div>
<div class="modal-body">
<?php $attributes = array('class' => 'form-horizontal'); ?>
<?php echo form_open('use_gradebook_scale',$attributes);?>
<input type="hidden" name="setup_id" value="<?php echo $gradebook_details
['gradebook_setup_details_id']?>"/>;
<input type="hidden" name="course_id" value="<?php echo $gradebook_details
['course_id']?>"/>
<p>Note: Using an existing gradebook scale will overwrite the current
scale being used in this course. </p>
<div class="form-group">
Existing Templates:
<select id="existing_templates" class="form-control" name="
existing_templates">
<?php $counter = count($scale_templates);?>
<?php if($counter > 0) {?>
<?php for($a = 0; $a < $counter; $a++) {?>
<option value="<?php echo $scale_templates[$a]['scale_template_id'];?>"><?
php echo $scale_templates[$a]['scale_template_name'];?></option>
<?php } ?>
<?php } else { ?>
<option>No existing templates yet</option>
<?php } ?>
</select>
</div>
<p></p>
<div class="modal-footer">
<input type="submit" class="btn btn-primary" value="Submit"></input>
<button type="button" class="btn btn-default" data-dismiss="modal">Close</
button>
<?php echo form_close();?>
</div>
</div>
</div>
</div>

```

```

<div>
<i class="fa fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-6">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li class="active">
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li>
Grade Settings
</li>
<li class="active">
<strong><a href="<?php echo base_url('gradebook_setup_list');?>">Assigned
course categories</a></strong>
</li>
</ol>
</div>
</div>

```

Gradebook Setup List View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
bottom: 0">
<div class="navbar-header">
<a class="navbar-minimalize minimalize-styl-2 btn btn-primary" href="#"><
i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailbox.html">

```

```

<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-6">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li class="active">
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li>
Grade Settings
</li>
<li class="active">
<strong><a href="<?php echo base_url('gradebook_setup_list');?>">Assigned
course categories</a></strong>
</li>
</ol>
</div>
</div>

```

```

<div class="row">
<br/>
<div class="col-lg-10">
<?php if($this->session->flashdata('course_cat_registry_succeed')) { ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>New course category added!</p>
</div>
<?php } ?>
</div>
<div class="col-lg-10">
<div class="ibox float-e-margins">
<div class="ibox-content">
<?php if(count($categories) > 0) {?>
<?php foreach($categories as $row){?>
<div class="well well-sm">
<p>
<h3>
&nbsp;<?php echo $row['course_category_name'];?>&nbsp;<button type="
button" class="btn btn-primary" data-toggle="modal" data-target="#
assignCategoryScaleModal" data-category_id="<?php echo $row['
course_category_id'];?>"><i class="fa fa-angle-down"> Assign
gradebook scale</i></button><br/><br/></h3>
<?php if(count($courses) > 0) { ?>
<?php foreach($courses as $class) {?>
<?php if($class['course_category_id'] == $row['course_category_id']) { ?>
<ul class="unstyled">
<li>
<a href="<?php echo base_url('gradebook_setup/'. $class['course_id']);?>">
large<?php echo $class['course_short_name'].'_'. $class['
section_name'];?></a></li>
</ul>
<?php } ?>
<?php } ?>
<?php } ?>
<p>
</div>
<?php } ?>
<?php } ?>
</div>
</div>
</div>
</div>
<div class="footer">
<div>
<strong><i class="fa fa-copyright"></i> 2017</strong> Electronic Class
Record Inquiry System
</div>
</div>
</div>
</div>
<div class="modal fade" id="assignCategoryScaleModal" tabindex="-1" role="
dialog" aria-labelledby="addGradeItemModalLabel" aria-hidden="true">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel">Assign Gradebook Scale for
this Course Category
<button type="button" class="close" data-dismiss="modal" aria-label="Close
">

```

```

<span aria-hidden="true">&times;</span>
</button>
</h5>
</div>
<div class="modal-body">
<?php $attributes = array('class' => 'form-horizontal'); ?>
<?php echo form_open('assign_category_scale',$attributes);?>
<p>Note: The selected grading scale will be used by all of the courses
assigned to this category.</p>
<input type="hidden" name="category_id" id="category_id"/>
<div class="form-group">
Existing Templates:
<select id="existing_templates" class="form-control" name="
existing_templates">
<?php $counter = count($scale_templates);?>
<?php if($counter > 0) {?>
<?php for($a = 0; $a < $counter; $a++) {?>
<?php if($scale_templates[$a]['school_id'] != 0) {?>
<option value="<?php echo $scale_templates[$a]['scale_template_id'];?>"><
?php echo $scale_templates[$a]['scale_template_name'];?></option>
<?php } ?>
<?php } ?>
<?php } else { ?>
<option>No existing templates yet</option>
<?php } ?>
</select>
</div>
<p></p>
<div class="modal-footer">
<input type="submit" class="btn btn-primary" value="Submit"></input>
<button type="button" class="btn btn-default" data-dismiss="modal">Close</
button>
<?php echo form_close();?>
</div>
</div>
</div>
</div>

```

Grades Stats View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
bottom:0">
<div class="navbar-header">
<a class="navbar-minimize minimize-styl-2 btn btn-primary" href="#"><i
class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
;?></span>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>

```

```

</div>
<div class="row_wrapper border-bottom white-bg page-heading">
<div class="col-sm-4">
<h2><?php echo $title2;></h2>
<ol class="breadcrumb">
<li class="active">
<strong><a href="<?php echo base_url('')?>">Home</a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<h5>Grade Categories/Items</h5>
<?php $letters = array('B','C','D','E','F','G','H','I','J','K','L','M','N',
,'O','P','Q','R','S','T','U','V','W','X','Y','Z','AA','AB','AC','AD',
,'AE','AF','AG','AH','AJ','AK','AL','AM','AN','AO','AP','AQ','AR','',
,'AS','AT','AU','AV','AW','AX','AY','AZ');
error_reporting(0);
?>
</div>
<div class="ibox-content">
<form method="get" id="select_category" class="form-horizontal">
<div class="form-group">
<label class="col-sm-2 control-label">Name:</label>
<div class="col-sm-10">
<select class="form-control" name="category_names" id="category_names"
>
<?php for($a = 0; $a < count($category_names); $a++) {?>
<?php if($category_names[$a]['cell_type'] == 1 && $category_names[$a]['
category_type'] == 1) {?>
<option value="<?=$letters[$category_names[$a]['column_letter']]?>"><?=$
$category_names[$a]['column_title']?></option>
<?php } else if($category_names[$a]['cell_type'] == 2 && $category_names[
$a]['category_type'] == 1) { ?>
<option value="<?=$letters[$category_names[$a]['column_letter']]?>"><?=$
$category_names[$a]['main_category_name'].'_Total'?></option>
<?php } else if($category_names[$a]['cell_type'] == 1 && $category_names[
$a]['category_type'] == 2) { ?>
<option value="<?=$letters[$category_names[$a]['column_letter']]?>"><?=$
$category_names[$a]['column_title']?></option>
<?php } ?>
<?php } ?>
<option value="<?=$letters[$a+1]?>">Final Grade</option>
</select>
</div>
</div>
</form>
<form id="sheet" method="get" class="form-horizontal">
<?php $letters = array('C','D','E','F','G','H','I','J','K','L','M','N','O',
,'P','Q','R','S','T','U','V','W','X','Y','Z','AA','AB','AC','AD','AE',
,'AF','AG','AH','AJ','AK','AL','AM','AN','AO','AP','AQ','AR','AS',
,'AT','AU','AV','AW','AX','AY','AZ');
error_reporting(0);
?>
<div class="form-group"><label class="col-sm-2 control-label">Details:</
label>
<div class="col-sm-10">

```

```

<?php for($a = 0; $a < count($class_record); $a++) {?>
<?php if($a > -1) { ?>
<?php for($b = 0; $b < count($class_record[$a]); $b++) {?>
<?php if($class_record[$a][$letters[$b]][0] == '=' ) {?>
<?php ?>
<input type="hidden" data-cell="<?=$letters[$b].($a+1)?>" data-formula='
<?=$substr($class_record[$a][$letters[$b]],1)?>' />
<?php } else { ?>
<input type="hidden" data-cell="<?=$letters[$b].($a+1)?>" value='<?=$
$class_record[$a][$letters[$b]]?>' />
<?php } ?>
<?php } ?>
<?php } ?>
</div>
<div class="col-sm-10">
<label class="control-label"><?=$count($enrolled_students). 'currently
enrolled students.' . $count($dropped_students). 'dropped' ?></label>
</div>
<div class="form-group"><label class="col-sm-2 control-label"></label>
<div class="col-sm-10" id="histograph">
</div>
<?php $letter = $_GET['cat_letter'];?>
<?php $number = (3 + count($enrolled_students)) - 1;?>
<div class="form-group"><label class="col-sm-2 control-label">Mean:</label
>
<div class="col-sm-10">
<input type="text" class="form-control" style="border:none;background:
transparent;" data-formula='AVERAGE(<?php echo $letter;?>3;<?php
echo $letter.$number;?>)' />
</div>
</div>
<div class="form-group"><label class="col-sm-2 control-label">Lowest Score
:</label>
<div class="col-sm-10">
<input type="text" class="form-control" style="border:none;background:
transparent;" data-formula='MIN(<?php echo $letter;?>3;<?php echo
$letter.$number;?>)' />
</div>
</div>
<div class="form-group"><label class="col-sm-2 control-label">Highest
Score:</label>
<div class="col-sm-10">
<input type="text" class="form-control" style="border:none;background:
transparent;" data-formula='MAX(<?php echo $letter;?>3;<?php echo
$letter.$number;?>)' />
</div>
</div>
</form>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>

```



```
</div>
</div>
```

Login View

```

<div id="preloader"></div>
<div id="top"></div>
<div class="landing">
<div class="overlay">
<div class="container">
<div class="row">
<div class="col-md-7">
<!-- /.logo -->
<div class="logo_wow"> <a href=""><img src=""><?php echo base_url('assets/
backyard/images/uplogol.png');?>" alt="logo"></a></div>
<!-- /.main title -->
<h1 class="wow">
Electronic Class Record Inquiry System
</h1>
<!-- /.header paragraph -->
<div class="landing-text_wow">
<p>Managing and organizing class records to provide accurate student data
</p>
</div>
<!-- /.header button -->
<div class="head-btn_wow">
<a class="btn-primary" href=""><?php echo base_url('register');?>">Register
for an admin account</a>
</div>
</div>
<!-- /.signup form -->
<div class="col-md-5 vert-offset-top-6">
<div class="signup-header_wow" id="login">
<h3 class="form-title text-center">Account login</h3>
<?php if($this->session->flashdata('succeed')) { ?>
<div class="alert alert-success" role="alert">
<strong>Registration Success!</strong> Verify your email address before
logging in.
</div>
<?php } else if($this->session->flashdata('login_error')) { ?>
<div class="alert alert-danger" role="alert">
<strong>Login Error!</strong> <?php echo $this->session->flashdata('
login_error'); ?>
</div>
<?php } else if($this->session->flashdata('login_succeed')) {?>3
<div class="alert alert-success" role="alert">
<strong>Login Success!</strong> <?php echo $this->session->flashdata('
login_succeed'); ?>
</div>
<?php } else if($this->session->flashdata('email_not_verified')) {?>3
<div class="alert alert-danger" role="alert">
<strong>Request Error!</strong> <?php echo $this->session->flashdata('
email_not_verified'); ?>
</div>
<?php } ?>
<?php echo form_open('login');?>
<div class="form-group">
<input class="form-control input-lg" name="user_name" id="name" type="text
" placeholder="Username" required/>
</div>

```

```

<div class="form-group">
<input class="form-control input-lg" name="user_password" id="password"
type="password" placeholder="Password" required/>
</div>
<div class="form-group last">
<input type="submit" class="btn btn-warning btn-block btn-lg" value="
Submit">
</div>
<p class="privacy text-center">Forgot Password? Click <a href=""><?php echo
base_url('forgot_pass');?>">here</a>. </p>
<?php echo form_close(); ?>
</div>
</div>
</div>
</div>
<div id="footer">
<div class="container">
<div class="col-sm-4 col-sm-offset-4">
<!-- /.social links -->
<div class="social text-center">
<ul>
<li><a class="wow fadeInUp" href="https://twitter.com/d4nchester"><i class
="fa fa-twitter"></i></a></li>
<li><a class="wow fadeInUp" href="https://www.facebook.com/d4nchester"
data-wow-delay="0.2s"><i class="fa fa-facebook"></i></a></li>
<li><a class="wow fadeInUp" href="https://instagram.com/d4nchester" data-
wow-delay="0.6s"><i class="fa fa-instagram"></i></a></li>
</ul>
</div>
<div class="text-center wow fadeInUp" style="font-size: 14px;">Electronic
Class Record Inquiry System - <br/> Dan Chester D. Francisco (CMSC
198)</div>
<a href="#" class="scrollToTop"><i class="pe-7s-up-arrow pe-va"></i></a>
</div>
</div>
</div>

```

Modify Category Item View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
bottom: 0">
<div class="navbar-header">
<a class="navbar-minimalize minimize-styl-2 btn btn-primary" href="#"><
i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu dropdown-alerts">
<li>

```

```

<a href="mailbox.html">
</a>
</div>
<i class="fa fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
</a>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
</a>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-12">
<h2><?php echo $title.' - '. $course_details['course_short_name'].' - ' .
    $course_details['course_full_name'].'';?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li>
<a>Grade Settings</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('modify_gradebook_categories/' .
    $course_details['course_id']);?>">Add/modify gradebook categories</a>
</strong>
</li>
</ol>
</div>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<div class="ibox-tools">
<a class="collapse-link">
<i class="fa fa-chevron-up"></i>
</a>
</div>
</div>
</div>
<div class="ibox-content">
<?php echo form_open('modify_grade_category');?>
<input type="hidden" class="form-control" value="<?php echo
    $gradebook_details['course_id'];?>" name="course_id" id="course_id">
<input type="hidden" class="form-control" value="<?php echo
    $gradebook_details['gradebook_setup_details_id'];?>" name="
    gradebook_id" id="gradebook_id">
</div>
<div class="row">
<div class="col-lg-8">
<div class="form-group">
<label>Method of Computation</label>
<select class="form-control" name="method_of_computation" id="
    computation_methods">
<?php if($gradebook_details['method_of_computation'] === "Mean_of_Grades")
    { ?>
<option value="Mean_of_Grades" selected>Mean of Grades</option>
<option value="Weighted_Mean_of_Grades">Weighted Mean of Grades</option>
<option value="Natural_Aggregate">Natural Aggregate</option>
<?php } else if($gradebook_details['method_of_computation'] === "Weighted_
    Mean_of_Grades") { ?>
<option value="Mean_of_Grades">Mean of Grades</option>
<option value="Weighted_Mean_of_Grades" selected>Weighted Mean of Grades</
    option>
<option value="Natural_Aggregate">Natural Aggregate</option>
<?php } else { ?>
<option value="Mean_of_Grades">Mean of Grades</option>
<option value="Weighted_Mean_of_Grades">Weighted Mean of Grades</option>
<option value="Natural_Aggregate" selected>Natural Aggregate</option>
<?php } ?>
</select>
</div>
<div class="table-responsive">
<h5>Insert/update categories</h5>
<small>- The item names should be unique</small><br/>
<small>- For weighted mean of grades, the sum of the weights must be 100</
    small><br/>
<table class="table table-condensed" id="dynamic_field">
<tbody>
<tr><td><button type="button" class="btn btn-white add"><i class="fa fa-
    plus-circle"></i> Add new category</button></td><td></td><td></td></tr>
<tr>
<td><input type="hidden" name="category_ids[]" value="<?php echo
    $gradebook_details['course_id'];?>">
</td>
</tr>
</tbody>
</table>
</div>

```

```

<td><input type="text" name="category_name[]" value="<?php_<br>
    $gradebook_categories[$index]['main_category_name'];?>" placeholder="<br>
    Name" class="form-control_name_list_category_names" required="<br>
    required" /></td>
<td><input type="text" name="category_weight[]" value="<?php_<br>
    $gradebook_categories[$index]['main_category_weight'];?>"<br>
    placeholder="Weight" class="form-control_name_list_<br>
    main_category_weight" required="required" /></td>
<td><button type="button" name="remove" id="<?php_<br>
    echo_<br>
    $index;?>" class="<br>
    btn btn-danger btn_remove">X</button></td>
</tr>
<?php } ?>
<?php } ?>
</tbody>
</table>
</div>
<div class="form-group">
<a href="<?php_<br>
    echo_base_url('gradebook_setup/'.<br>
    $gradebook_details['<br>
    course_id']?>" class="btn btn-white" /><i class="fa fa-arrow-left"<br>
    ></i> Go back to list</a>
<input type="submit" id="button2" class="btn btn-primary" value="Update"/>
</div>
<?php echo form_close();?>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>
</div>

```

Modify Gradebook Category View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
    bottom:0">
<div class="navbar-header">
<a class="navbar-minimize minimize-styl-2 btn btn-primary" href="#"><br>
    <i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name<br>
    ;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailbox.html">
</li>
</ul>
</div>

```

```

<i class="fa fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
</div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
</div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php_<br>
    echo_base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-12">
<h2><?php echo $title.'_'<br>
    $course_details['course_short_name'].<br>
    $course_details['course_full_name'].?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php_<br>
    echo_base_url('');?>">Home</a>
</li>
<li>
<a>Grade Settings</a>
</li>
<li class="active">
<strong><a href="<?php_<br>
    echo_base_url('modify_gradebook_categories/'<br>
    $course_details['course_id']?>">Add/modify gradebook categories</a>
</strong>
</li>
</ol>
</div>

```

```

</div>
<div class="row">
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<div class="ibox-tools">
<a class="collapse-link">
<i class="fa fa-chevron-up"></i>
</a>
</div>
</div>
<div class="ibox-content">
<?php echo form_open('modify_grade_category');?>
<input type="hidden" class="form-control" value="<?php echo
    $gradebook_details['course_id'];?>" name="course_id" id="course_id">
<input type="hidden" class="form-control" value="<?php echo
    $gradebook_details['gradebook_setup_details_id'];?>" name="
    gradebook_id" id="gradebook_id">
<div class="row">
<div class="col-lg-10">
<div class="form-group">
<label>Method of Computation</label>
<select class="form-control" name="method_of_computation" id="
    computation_methods">
<?php if($gradebook_details['method_of_computation'] === "Mean_of_Grades")
    { ?>
<option value="Mean_of_Grades" selected>Mean of Grades</option>
<option value="Weighted_Mean_of_Grades">Weighted Mean of Grades</option>
<option value="Natural_Aggregate">Natural Aggregate</option>
<?php } else if($gradebook_details['method_of_computation'] === "Weighted_
    Mean_of_Grades") { ?>
<option value="Mean_of_Grades">Mean of Grades</option>
<option value="Weighted_Mean_of_Grades" selected>Weighted Mean of Grades</
    option>
<option value="Natural_Aggregate">Natural Aggregate</option>
<?php } else { ?>
<option value="Mean_of_Grades">Mean of Grades</option>
<option value="Weighted_Mean_of_Grades">Weighted Mean of Grades</option>
<option value="Natural_Aggregate" selected>Natural Aggregate</option>
<?php } ?>
</select>
</div>
<div class="table-responsive">
<div class="well">
<h4>How to update main categories/items</h4>
<br/>
<h5> - No empty/blank input fields</h5>
<h5> - No duplicate main category/item names</h5>
<h5> - For weighted mean of grades, the sum of the weights must be less
    than or equal to 100</h5>
</div>
<table class="table table-condensed" id="dynamic_field">
<tbody>
<tr><td class="col-md-5"><button type="button" class="btn btn-white add"><
    i class="fa fa-plus-circle"></i> Add new category</button>&nbsp;<
    button type="button" class="btn btn-white add_item"><i class="fa fa-
    plus-circle"></i> Add new item</button></td>
<td class="col-md-1"></td>
<td></td>
</tr>
</tbody>
</table>
</div>
</div>
<tr>
<td class="col-md-3">Category/Item Name</td>
<td class="col-md-3">Weight Value (for weighted mean of grades)</td>
<td class="col-md-3">Total Score (for grade items)</td>
<td class="col-md-3"></td>
</tr>
<?php if($gradebook_count > 0) {?>
<?php $real_index = 0; ?>
<?php for($index = 0; $index < $gradebook_count; $index++){ ?>
<?php if($gradebook_categories[$real_index]['main_category_type'] === "
    CATEGORY"){ ?>
<tr id="row<?php echo $index;?>">
<input type="hidden" name="category_ids[]" value="<?php echo
    $gradebook_categories[$real_index]['main_category_id'];?>"/>
<input type="hidden" name="category_type[]" value="<?php echo
    $gradebook_categories[$real_index]['main_category_type'];?>"/>
<td class="col-md-3"><input type="text" name="category_name[]" value="<?
    php echo $gradebook_categories[$real_index]['main_category_name'];?>
    " placeholder="CategoryName" class="form-control name_list_
    category_names" required="required" /></td>
<td class="col-md
-3"><div class="input-group m-b"><input type="text" name="
    category_weight[]" value="<?php echo $gradebook_categories[
    $real_index]['main_category_weight'];?>" placeholder="Category_
    Weight" class="form-control name_list_main_category_weight"
    required="required" /> <span class="input-group-addon">%</span></div
    ></td>
<td class="col-md-3"><input type="text" name="category_total[]" value="<?
    php echo $gradebook_categories[$real_index]['total_points'];?>"
    placeholder="CategoryTotal" class="form-control name_list_
    total_points" required="required" readonly/></td>
<td class="col-md-3"><button type="button" name="remove" id="<?php echo
    $index;?>" class="btn btn-danger btn_remove"><i class="fa fa-minus"
    ></i> Delete Category</button></td>
</tr>
<?php $real_index++; ?>
<?php } ?>
<?php } ?>
<?php for($index = $real_index; $index < $gradebook_count; $index++){ ?>
<?php if($gradebook_categories[$index]['main_category_type'] === "ITEM"){
    ?>
<tr id="row<?php echo $index;?>">
<input type="hidden" name="category_ids[]" value="<?php echo
    $gradebook_categories[$index]['main_category_id'];?>"/>
<input type="hidden" name="category_type[]" value="<?php echo
    $gradebook_categories[$index]['main_category_type'];?>"/>
<td class="col-md-3"><input type="text" name="category_name[]" value="<?
    php echo $gradebook_categories[$index]['main_category_name'];?>"
    placeholder="ItemName" class="form-control name_list_
    category_names" required="required" /></td>
<td class="col-md-3"><div class="input-group m-b"><input type="text" name="
    category_weight[]" value="<?php echo $gradebook_categories[
    $index
    ]['main_category_weight'];?>" placeholder="CategoryWeight" class="
    form-control name_list_main_category_weight" required="required" />
    <span class="input-group-addon">%</span></div></td>
<td class="col-md-3">
<?php if($gradebook_categories[$index]['total_points'] == 0) {?>
<input type="text" name="category_total[]" placeholder="ItemTotal" class="
    form-control name_list_total_points" required="required"/>
<?php } else { ?>

```



```

<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-12">
<h2><?php echo $title.$category_details['main_category_name'].'<br>
    $course_details['course_short_name'].?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li>
<a>Grade Settings</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('modify_gradebook_items/' .
    $gradebook_details['gradebook_setup_details_id'].?>'.
    $category_details['main_category_id'];?>">Modify category</a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<div class="ibox-tools">
<a class="collapse-link">
<i class="fa fa-chevron-up"></i>
</a>
</div>
</div>
<div class="ibox-content">
<?php echo form_open('modify_category_items');?>
<input type="hidden" class="form-control" value="<?php echo
    $category_details['main_category_id'];?>" name="category_id" id="
    category_id">
<input type="hidden" class="form-control" value="<?php echo
    $gradebook_details['gradebook_setup_details_id'];?>" name="
    gradebook_id" id="gradebook_id">
<div class="row">
<div class="col-lg-10">
<div class="form-group">
<label class="form-control-label">Category Name</label>
<input type="text" class="form-control" name="gradebook_category_name"
    value="<?php echo $category_details['main_category_name'];?>" />

```

```

<?php } ?>
</td>
<td>
<button type="button" name="remove" id="<?php echo $index2;" class="btn btn-info btn_remove"><i class="fa fa-minus"></i> Delete Grade Item</button></td>
</tr>
<?php } ?>
<?php } ?>
<?php if($this->session->userdata('move_to_category')){ ?>
<?php $new_size = count($new_items);?>
<?php for($index2 = 0; $index2 < $new_size; $index2++){ ?>
<tr id="row<?php echo $index2;">
<input type="hidden" name="category_ids[]" value="0"/>
<input type="hidden" name="grade_item_ids[]" value="<?php echo $new_items[
    $index2]['grade_item_id'];?>"/>
<input type="hidden" name="grade_item_categories[]" value="<?php echo
    $new_items[$index2]['main_category_id'];?>"/>
<td class="col-md-3"><input type="text" name="category_name[]" value="<?
    php echo $new_items[$index2]['grade_item_name'];?>" placeholder="
    Item Name" class="form-control name_list category_names" required="
    required" /></td>
<td class="col-md-3"><div class="input-group m-b"><input type="text" name=
    "category_weight[]" placeholder="Item Weight" class="form-control
    name_list main_category_weight" required="required" /><span class="
    input-group-addon">%</span></div></td>
<td class="col-md-3">
<?php if($new_items[$index2]['total_points'] == 0) {?>
<input type="text" name="category_total[]" placeholder="Item Total" class=
    "form-control name_list total_points" required="required"/>
<?php } else { ?>
<input type="text" name="category_total[]" value="<?php echo $new_items[
    $index2]['total_points'];?>" placeholder="Item Total" class="form-
    control name_list total_points" required="required"/>
<?php } ?>
</td>
<td class="col-md-3"><button type="button" name="remove" id="<?php echo
    $index2;" class="btn btn-info btn_remove"><i class="fa fa-minus"
    ></i> Delete Grade Item</button></td>
</tr>
<?php } ?>
<?php $new_size = count($new_main_items);?>
<?php for($index2 = 0; $index2 < $new_size; $index2++){ ?>
<tr id="row<?php echo $index2;">
<input type="hidden" name="category_ids[]" value="0"/>
<input type="hidden" name="category_item_ids[]" value="<?php echo
    $new_main_items[$index2]['main_category_id'];?>"/>
<td class="col-md-3"><input type="text" name="category_name[]" value="<?
    php echo $new_main_items[$index2]['main_category_name'];?>"
    placeholder="Item Name" class="form-control name_list category_names
    " required="required" /></td>
<td class="col-md-3"><div class="input-group m-b"><input type="text" name=
    "category_weight[]" placeholder="Item Weight" class="form-control
    name_list main_category_weight" required="required" /><span class="
    input-group-addon">%</span></div></td>
<td class="col-md-3">
<?php if($new_main_items[$index2]['total_points'] == 0) {?>
<input type="text" name="category_total[]" placeholder="Item Total" class=
    "form-control name_list total_points" required="required"/>
<?php } else { ?>

```

```

<input type="text" name="category_total[]" value="<?php echo
    $new_main_items[$index2]['total_points'];?>" placeholder="Item Total
    " class="form-control name_list total_points" required="required"/>
<?php } ?>
</td>
<td class="col-md-3"><button type="button" name="remove" id="<?php echo
    $index2;" class="btn btn-info btn_remove"><i class="fa fa-minus"
    ></i> Delete Grade Item</button></td>
</tr>
<?php } ?>
<?php } ?>
</tbody>
</table>
</div>
<div class="form-group">
<a href="<?php echo base_url('gradebook_setup/'.$gradebook_details['
    course_id']);?>" class="btn btn-white" /><i class="fa fa-arrow-left"
    >></i> Go back to list</a>
<input type="submit" id="button2" class="btn btn-primary" value="Update"/>
</div>
<?php echo form_close();?>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>
</div>

```

Student Course Record View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
    bottom: 0">
<div class="navbar-header">
<a class="navbar-minimize minimize-styl-2 btn btn-primary" href="#"><i
    class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
    ;?></span>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-6">

```

```

<h2><?php echo $title.'_'.$course_details['section_name'];?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('')>">Home</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('course_record_info').$course_details
['course_id'];?>"><?php echo $course_details['course_short_name'].'_'
.$course_details['section_name'];?></a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="wrapper-wrapper-content">
<div class="animated_fadeInRightBig">
<div class="row">
<div class="col-lg-4_tooltip-demo">
<button onclick="location.href='<?php echo base_url('course_announcements
/').$course_details['course_id'];?>';" class="btn btn-warning btn-lg
dim" type="button" data-toggle="tooltip" data-placement="right"
title="" data-original-title="Course Announcements"><i class="fa fa-
exclamation"></i><i class="fa fa-exclamation"></i></button>
</div>
<div class="col-lg-12">
<div class="ibox">
<div class="ibox-title">
<h5>Record - <?php echo $course_details['course_short_name'].'_'
.$course_details['section_name'];?>
</h5>
</div>
<div class="ibox-content" style="overflow: hidden;">
<div class="well">
<form id="sheet" method="get" class="form-horizontal">
<?php $letters = array('C','D','E','F','G','H','I','J','K','L','M','N','O'
,'P','Q','R','S','T','U','V','W','X','Y','Z','AA','AB','AC','AD','AE'
,'AF','AG','AH','AJ','AK','AL','AM','AN','AO','AP','AQ','AR','AS','
AT','AU','AV','AW','AX','AY','AZ');
error_reporting(0);
?>
<?php $count = count($category_total); ?>
<?php $value = $category_total[$count - 1]; ?>.
<?php $class_record[0][$letters[$value - 1]] = 'Final_Grade'?>
<?php $class_record[0][$letters[$value]] = 'Scaled_Grade';
$order = $order_number + 1;
$limit = $order + 1;
?>
<div class="form-group">
<div class="col-sm-2">
<ul class="list-group">
<li class="list-group-item" >
<span style="font-weight: bold;"></span></div>
</li>
<?php for($a = 0; $a < 1; $a++) {?>
<?php for($b = 0; $b < count($class_record[$a]); $b++) {?>
<li class="list-group-item">
<?php if(strpos($class_record[$a][$letters[$b]], 'Total') != false) {?>
<input type="text" disabled style="background:transparent;border:none;font
-weight: bold;" data-cell="<?=$letters[$b].($a+1)?>" value="<?=$

```

```

$class_record[$a][$letters[$b]]?>'/><br/>
<?php } else if(strpos($class_record[$a][$letters[$b]], 'Final_Grade') !=
false || strpos($class_record[$a][$letters[$b]], 'Scaled_Grade')
!= false) {?>
<input type="text" disabled style="background:transparent;border:none;font
-weight: bold;padding-left: 2em;text-decoration: underline;" data-
cell="<?=$letters[$b].($a+1)?>" value="<?=$class_record[$a][$letters
[$b]]?>'/><br/>
<?php } else { ?>
<input type="text" disabled style="background:transparent;border:none;"
data-cell="<?=$letters[$b].($a+1)?>" value="<?=$class_record[$a][
$letters[$b]]?>'/><br/>
<?php } ?>
</li>
<?php } ?>
</ul>
</div>
<div class="col-sm-2">
<li class="list-group-item" >
<span style="font-weight: bold;"></span></div>
</li>
<?php for($a = 1; $a < 2; $a++) {?>
<?php for($b = 0; $b < count($class_record[$a]); $b++) {?>
<li class="list-group-item">
<?php if($class_record[$a][$letters[$b]][0] == '=') {?>
<?php ?>
<input type="text" disabled style="background:transparent;border:none;"
data-cell="<?=$letters[$b].($a+1)?>" data-formula="<?=$substr(
$class_record[$a][$letters[$b]], 1)?>'/><br/>
<?php } else { ?>
<input type="text" disabled style="background:transparent;border:none;"
data-cell="<?=$letters[$b].($a+1)?>" value="<?=$class_record[$a][
$letters[$b]]?>'/><br/>
<?php } ?>
</li>
<?php } ?>
</div>
<div class="col-sm-4">
<ul class="list-group">
<li class="list-group-item" >
<span style="font-weight: bold;"></span></div>
</li>
<?php for($a = $order; $a < $limit; $a++) {?>
<?php for($b = 0; $b < count($class_record[$a]); $b++) {?>
<li class="list-group-item">
<?php if($class_record[$a][$letters[$b]][0] == '=') {?>
<?php ?>
=
<input type="text" disabled style="background:transparent;border:none;font
-weight: bold;padding-left: 2em;" data-cell="<?=$letters[$b].($a+1)
?>" data-formula="<?=$substr($class_record[$a][$letters[$b]], 1)?>'/
/><br/>
<?php } else { ?>
<input type="text" disabled style="background:transparent;border:none;"
data-cell="<?=$letters[$b].($a+1)?>" value="<?=$class_record[$a][
$letters[$b]]?>'/><br/>
<?php } ?>
</li>

```



```

</ul>
</div>
<div class="text-center_wow_fadeInUp" style="font-size:14px;">Electronic
    Class Record Inquiry System - <br/> Dan Chester D. Francisco (CMSC
    198)</div>
<a href="#" class="scrollTop"><i class="pe-7s-up-arrow_upe_va"></i></a>
</div>
</div>
</footer>

```

Student Form View

```

<div id="page-wrapper" class="gray-bg">
<div class="row_border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
    bottom:0">
<div class="navbar-header">
<a class="navbar-minimize_minimize-styl-2_btn btn-primary" href="#">
    <i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted_welcome-message"><?php echo $school_name
    ;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle_count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu_dropdown-alerts">
<li>
<a href="mailbox.html">
<div>
<i class="fa fa-envelope_fa-fw"></i> You have 16 messages
<span class="pull-right_text-muted_small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter_fa-fw"></i> 3 New Followers
<span class="pull-right_text-muted_small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload_fa-fw"></i> Server Rebooted
<span class="pull-right_text-muted_small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center_link-block">
<a href="notifications.html">

```

```

<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row_wrapper_border-bottom_white-bg_page-heading">
<div class="col-sm-6">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li>
<a>Accounts</a>
</li>
<li>
<a>Students</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('student_form');?>">Add new student/s
    </a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="ibox_float-e-margins">
<div class="ibox-title">
<div class="ibox-tools">
<a class="collapse-link">
<i class="fa fa-chevron-up"></i>
</a>
</div>
</div>
<div class="ibox-content">
<?php if($this->session->flashdata('student_registry_error')) { ?>
<div class="alert_alert-danger"><strong>Error!</strong>
<p><?php echo $this->session->flashdata('student_registry_error');?></p>
</div>
<?php } ?>
<?php echo form_open('add_student', array('class' => 'form-horizontal'));?>
<input type="hidden" name="school_id" value="<?php echo $_SESSION['
    logged_in']['school_id'];?>">
<div class="form-group"><label class="col-sm-2_control-label">First Name</
    label>
<div class="col-sm-6"><input type="text" name="student_first_name" class="
    form-control" required="required"/></div>
</div>
<div class="hr-line-dashed"></div>

```

```

<div class="form-group"><label class="col-sm-2 control-label">Last Name</label>
<div class="col-sm-6"><input type="text" name="student_last_name" class="form-control" required="required"/></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group"><label class="col-sm-2 control-label">Email Address</label>
<div class="col-sm-6"><input type="text" name="student_email_address" class="form-control" required="required"/></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group">
<div class="col-sm-4 col-sm-offset-2">
<a class="btn btn-white" href="<?php echo base_url('student_list');?>">Cancel<a>
<input class="btn btn-primary" type="submit" value="Submit"/>
</div>
</div>
<?php echo form_close('');?>
</div>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>
</div>

```

Student Groups Form View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-bottom: 0">
<div class="navbar-header">
<a class="navbar-minimalize minimalize-styl-2 btn btn-primary" href="#">
< i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name ;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailbox.html">
<div>
<i class="fa fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>

```

```

</li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-6">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li>
<a>Student</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('student_groups_form');?>">Add new student group</a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<div class="ibox-tools">
<a class="collapse-link">

```

```

<i class="fa fa-chevron-up"></i>
</a>
</div>
</div>
<div class="ibox-content">
<?php if($this->session->flashdata('student_group_error')) { ?>
<div class="alert alert-danger"><strong>Error!</strong>
<p><?php echo $this->session->flashdata('student_group_error');?></p>
</div>
<?php } ?>
<?php echo form_open('add_student_group',array('class' => 'form-horizontal
'));?>
<input type="hidden" name="school_id" value="<?php echo $_SESSION['
logged_in']['school_id']?>">
<div class="form-group"><label class="col-sm-2 control-label">Student
Group Name</label>
<div class="col-sm-6"><input type="text" name="student_group_name" class="
form-control" required="required"/></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group">
<label class="col-sm-2 control-label"></label>
<div class="col-sm-7">
<label>Select students from the left table that will be added to the group
</label><br/>
<a class="btn btn-sm btn-success" href="#" id="select_all">Select All</a>&
nbsp;
<a class="btn btn-sm btn-danger" href="#" id="deselect_all">Deselect All</
a>
<br/><br/>
<select id='searchable' multiple='multiple' name="group_members[]">
<?php if (isset($students)){ ?>
<?php for($i = 0; $i < count($students); $i++){ ?>
<option value="<?php echo $_students[$i]['student_id'];>">
<?php echo $students[$i]['user_first_name']. ' ' . $students[$i]['
user_last_name'];?>
</option>
<?php } ?>
<?php } else { ?>
<option value="0" disabled>No students available.</option>
<?php } ?>
</select>
</div>
</div>
<div class="form-group">
<div class="col-sm-4 col-sm-offset-2">
<a class="btn btn-white" href="<?php echo base_url('student_groups_list')
;?>">Cancel<a>
<input class="btn btn-primary" type="submit" value="Submit"/>
</div>
</div>
<?php echo form_close('');?>
</div>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>

```

```

</div>
</div>
</div>
Student Groups List View
<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
bottom:0">
<div class="navbar-header">
<a class="navbar-minimize minimize-styl-2 btn btn-primary" href="#"><
i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailbox.html">
<div>
<i class="fa fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>

```

```

</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-4">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li class>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li class>
<a href="#">Students</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('student_groups_list');?>">List of
student groups</a></strong>
</li>
</ol>
</div>
</div>
<div class="wrapper wrapper-content">
<div class="row">
<div class="col-lg-4">
<?php if($this->session->flashdata('student_group_succeed')) { ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>New student group added!</p>
</div>
<?php } else if($this->session->flashdata('remove_student_group_succeed'))
{ ?>
<div class="alert alert-danger"><strong>Success!</strong>
<p>The student group has been removed!</p>
</div>
<?php } ?>
<a style="color:white;" href="<?php echo base_url('student_groups_form')
;?>"><button class="btn btn-lg btn-primary" type="button"><i
class="fa fa-plus"> Add new student groups</i></button></a>
</div>
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<h5>List of student groups in <?php echo $school_name;?></h5>
</div>
<div class="ibox-content">
<table class="table table-striped table-bordered table-hover" dataTables-
example" >
<thead>
<tr>
<th>Student Group Name</th>
<th>Number of Members</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<?php if(isset($student_groups) && count($student_groups) > 0) {?>

```

```

<?php $size = count($student_groups);?>
<?php for($a = 0; $a < $size; $a++){ ?>
<tr>
<td><?php echo $student_groups[$a]['student_group_name'];?></td>
<td><?php echo $student_groups[$a]['student_members_count'];?></td>
<td><a href="<?php echo base_url('edit_student_group/'.$student_groups[$a
]['student_group_id']);?>" class="btn btn-sm btn-success">Edit</a>&
nbsp;<button data-toggle="modal" data-target="#
removeStudentGroupModal" data-whatever="<?php echo $student_groups[$a
]['student_group_id'];?>" class="btn btn-sm btn-danger">Remove</
button></td>
</tr>
<?php } ?>
<?php } ?>
</tbody>
</table>
</div>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
<div class="modal fade" id="removeStudentGroupModal" tabindex="-1" role="
dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel">Remove student group?</h5>
<button type="button" class="close" data-dismiss="modal" aria-label="Close
">
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<p>Are you sure you want to remove the student group? All the information
regarding the group will also be deleted. Press proceed to continue
.</p>
<?php echo form_open('remove_student_group');?>
<input type="hidden" class="form-control" name="student_group_id" id="
student_group_id">
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary" data-dismiss="modal">Close
</button>
<input type="submit" class="btn btn-primary" value="Proceed"/></button>
</div>
<?php echo form_close();?>
</div>
</div>
</div>
<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">

```

Student Home View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">

```



```

<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-4">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li class>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li class>
<a href="#">Student</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('student_list');?>">View list of
students</a></strong>
</li>
</ol>
</div>
</div>
<div class="wrapper wrapper-content">
<div class="row">
<div class="col-lg-8">
<?php if($this->session->flashdata('student_registry_succeed')) { ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>New student added!</p>
</div>
<?php } else if($this->session->flashdata('mult_student_registry_succeed')
) { ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>Upload students data complete!</p>
</div>
<div class="alert alert-danger"><strong></strong>
<?php
$first_names = $this->session->flashdata('mult_student_registry_succeed1')
;
$last_names = $this->session->flashdata('mult_student_registry_succeed2');
$email_dups = $this->session->flashdata('mult_student_registry_succeed3');
?>
<?php if(count($first_names) > 0) {?>

```

```

<p>These accounts are not registered since email addresses are already
existing in the system:</p>
<ul class="unstyled">
<?php for($a = 0; $a < count($first_names); $a++) {?>
<li>
<?php echo $first_names[$a].', '.$last_names[$a].', '.$email_dups[$a]?>
</li>
<?php } ?>
</ul>
<?php } ?>
</div>
<?php } else if($this->session->flashdata('delete_student_succeed')) { ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>Delete student account succeed!</p>
</div>
<?php } ?>
</div>
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<h5>List of all students in <?php echo $school_name;?></h5>
</div>
<div class="ibox-content">
<table class="table table-striped table-bordered table-hover dataTable example" >
<thead>
<tr>
<th>Student Last Name</th>
<th>Student First Name</th>
<th>Email Address</th>
<th>Verified?</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<?php $size = count($students);?>
<?php if(isset($students) && count($students) > 0) {?>
<?php for($a = 0; $a < $size; $a++){ ?>
<tr>
<td><?php echo $students[$a]['user_last_name'];?></td>
<td><?php echo $students[$a]['user_first_name'];?></td>
<td><?php echo $students[$a]['user_email_address'];?></td>
<?php if($students[$a]['email_verified'] === "YES") {?>
<td><span class="label label-primary">Verified</span></td>
<?php } else { ?>
<td><span class="label label-danger">Not yet verified</span></td>
<?php } ?>
<td><a href="<?php echo base_url('edit_student_details/'.$students[$a]['
student_id'];?>" class="btn btn-sm btn-success">Edit</a>&nbsp;&nbsp;<
button class="btn btn-sm btn-danger" data-toggle="modal" type="
button" data-target="#removeStudentModal" data-stud_id='<?php echo
$students[$a]['student_id'];?>'> Delete</button></td>
</tr>
<?php } ?>
<?php } ?>
</tbody>
</table>
</div>
</div>
</div>

```

```

</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
<div class="modal_fade" id="removeStudentModal" tabindex="-1" role="dialog"
  " aria-labelledby="addGradeItemModalLabel" aria-hidden="true">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel">
Remove Student
<button type="button" class="close" data-dismiss="modal" aria-label="Close"
  ">
<span aria-hidden="true">&times;</span>
</button>
</h5>
</div>
<div class="modal-body">
<?php $attributes = array('class' => 'form-horizontal'); ?>
<?php echo form_open('delete_student',$attributes);?>
<input type="hidden" name="student_id" id="student_id"/>
<p>Are you sure you want to delete this account? Click confirm to continue
  . </p>
<p></p>
<div class="modal-footer">
<input type="submit" class="btn btn-primary" value="Confirm"></input>
<button type="button" class="btn btn-default" id="disagreeButton" data-
  dismiss="modal">Close</button>
<?php echo form_close();?>
</div>
</div>
</div>
</div>

```

Teacher Form View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-
  bottom:0">
<div class="navbar-header">
<a class="navbar-minimalize minimize-styl-2 btn btn-primary" href="#"><
  i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
  ;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailbox.html">

```

```

<div>
<i class="fa fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-6">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php echo base_url('');?>">Home</a>
</li>
<li>
<a>Accounts</a>
</li>
<li>
<a>Teachers</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('teacher_form');?>">Add new teacher/s
  </a></strong>
</li>

```



```

</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="ibox_float-e-margins">
<div class="ibox-title">
<div class="ibox-tools">
<a class="collapse-link">
<i class="fa fa-chevron-up"></i>
</a>
</div>
</div>
</div>
<div class="ibox-content">
<?php if($this->session->flashdata('teacher_registry_error')) { ?>
<div class="alert alert-danger"><strong>Error!</strong>
<p><?php echo $this->session->flashdata('teacher_registry_error');?></p>
</div>
<?php } ?>
<?php echo form_open('add_teacher',array('class' => 'form-horizontal'));?>
<input type="hidden" name="school_id" value="<?php echo $_SESSION['logged_in']['school_id']?>">
<div class="form-group"><label class="col-sm-2 control-label">First Name</label>
<div class="col-sm-6"><input type="text" name="teacher_first_name" class="form-control" required="required"/></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group"><label class="col-sm-2 control-label">Last Name</label>
<div class="col-sm-6"><input type="text" name="teacher_last_name" class="form-control" required="required"/></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group"><label class="col-sm-2 control-label">Email Address</label>
<div class="col-sm-6"><input type="text" name="teacher_email_address" class="form-control" required="required"/></div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group">
<div class="col-sm-4 col-sm-offset-2">
<a class="btn btn-white" href="<?php echo base_url('teacher_list');?>">
Cancel <a>
<input class="btn btn-primary" type="submit" value="Submit"/>
</div>
</div>
<?php echo form_close('');?>
</div>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>

```

Teacher Home View

```

<div id="page-wrapper" class="gray-bg">
<div class="row border-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-bottom: 0">
<div class="navbar-header">
<a class="navbar-minimalize minimalize-styl-2 btn btn-primary" href="#"><i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name;?></span>
</li>
<li>
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-4">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li class="active">
<strong><a href="<?php echo base_url('')?>">Home</a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="wrapper wrapper-content">
<div class="col-sm-12">
<div class="ibox_float-e-margins">
<div class="ibox-title">
<h5>Assigned Courses</h5>
</div>
<div class="ibox-content">
<div class="well">
<h3>
Class Records
</h3>
<ul class="list-group">
<?php foreach($my_courses as $key => $value){ ?>
<li class="list-group-item">
<span class="badge badge-primary"><a href="<?php echo base_url('class_record/' . $value['course_id']);?>" style="color:white;" onMouseOver="this.style.color='red'" onMouseOut="this.style.color='white'"><i class='fa fa-search'></i> View record</a></span>
<span class="badge badge-warning"><a href="<?php echo base_url('course_announcements/' . $value['course_id']);?>" style="color:white;" onMouseOver="this.style.color='red'" onMouseOut="this.style.color='white'"><i class='fa fa-list'></i> View announcements</a></span>
<?php echo $value['course_short_name'] . ' - ' . $value['section_name']; ?>
</li>
<?php } ?>
</ul>

```



```

</li>
</ol>
</div>
</div>
<div class="wrapper_wrapper-content">
<div class="row">
<div class="col-lg-4">
<?php if($this->session->flashdata('teacher_registry_succeed')) { ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>New teacher added!</p>
</div>
<?php } else if($this->session->flashdata('mult_teacher_registry_succeed')
) { ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>Upload teachers data complete!</p>
</div>
<div class="alert alert-danger"><strong></strong>
<?php
$first_names = $this->session->flashdata('mult_teacher_registry_succeed1')
;
$last_names = $this->session->flashdata('mult_teacher_registry_succeed2');
$email_dups = $this->session->flashdata('mult_teacher_registry_succeed3');
?>
<?php if(count($first_names) > 0) {?>
<p>These accounts are not registered since email addresses are already
existing in the system:</p>
<ul class="unstyled">
<?php for($a = 0; $a < count($first_names); $a++) {?>
<li>
<?php echo $first_names[$a].', '. $last_names[$a].', '. $email_dups[$a]?>
</li>
<?php } ?>
</ul>
<?php } ?>
</div>
<?php } else if($this->session->flashdata('delete_teacher_succeed')) { ?>
<div class="alert alert-success"><strong>Success!</strong>
<p>Delete teacher account succeed!</p>
</div>
<?php } ?>
</div>
<div class="col-lg-12">
<div class="ibox float-e-margins">
<div class="ibox-title">
<h5>List of all teachers in <?php echo $school_name;?></h5>
</div>
<div class="ibox-content">
<table class="table table-striped table-bordered table-hover dataTable example" >
<thead>
<tr>
<th>Teacher Last Name</th>
<th>Teacher First Name</th>
<th>Email Address</th>
<th>Verified?</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<?php $size = count($teachers);?>
<?php if(isset($teachers) && count($teachers) > 0) {?>
<?php for($a = 0; $a < $size; $a++){ ?>
<tr>
<td><?php echo $teachers[$a]['user_last_name']?></td>
<td><?php echo $teachers[$a]['user_first_name']?></td>
<td><?php echo $teachers[$a]['user_email_address']?></td>
<?php if($teachers[$a]['email_verified'] === "YES") {?>
<td><span class="label label-primary">Verified</span></td>
<?php } else { ?>
<td><span class="label label-danger">Not yet verified</span></td>
<?php } ?>
<td><a class="btn btn-sm btn-success" href="<?php echo base_url('
edit_teacher_details/' . $teachers[$a]['teacher_id']?>)">Edit</a>&nbsp;
<button class="btn btn-sm btn-danger" data-toggle="modal" type="
button" data-target="#removeTeacherModal" data-teacher_id='<?php
echo $teachers[$a]['teacher_id'];?>'> Delete</button></td>
</tr>
<?php } ?>
</tbody>
</table>
</div>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
<div class="modal fade id="removeTeacherModal" tabindex="-1" role="dialog"
aria-labelledby="addGradeItemModalLabel" aria-hidden="true">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModalLabel">
Remove Teacher
<button type="button" class="close" data-dismiss="modal" aria-label="Close"
">
<span aria-hidden="true">&times;</span>
</button>
</h5>
</div>
<div class="modal-body">
<?php $attributes = array('class' => 'form-horizontal'); ?>
<?php echo form_open('delete_teacher', $attributes);?>
<input type="hidden" name="teacher_id" id="teacher_id"/>
<p>Are you sure you want to remove this account? Click confirm to continue
.</p>
<p></p>
<div class="modal-footer">
<input type="submit" class="btn btn-primary" value="Confirm"></input>
<button type="button" class="btn btn-default" id="disagreeButton" data-
dismiss="modal">Close</button>
<?php echo form_close();?>
</div>
</div>
</div>

```

```
</div>
```

Update Grading Scale View

```
<div id="page-wrapper" class="gray-bg">
<div class="rowborder-bottom">
<nav class="navbar navbar-static-top" role="navigation" style="margin-bottom: 0">
<div class="navbar-header">
<a class="navbar-minimalize minimalize-styl-2 btn btn-primary" href="#">
<i class="fa fa-bars"></i> </a>
</div>
<ul class="nav navbar-top-links navbar-right">
<li>
<span class="m-r-sm text-muted welcome-message"><?php echo $school_name
;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle count-info" data-toggle="dropdown" href="#">
<i class="fa fa-bell"></i> <span class="label label-primary">8</span>
</a>
<ul class="dropdown-menu dropdown-alerts">
<li>
<a href="mailbox.html">
<div>
<i class="fa fa-envelope fa-fw"></i> You have 16 messages
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa fa-twitter fa-fw"></i> 3 New Followers
<span class="pull-right text-muted small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa fa-upload fa-fw"></i> Server Rebooted
<span class="pull-right text-muted small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
</li>
```

```
<a href="<?php echo base_url('logout');?>">
<i class="fa fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row wrapper border-bottom white-bg page-heading">
<div class="col-sm-6">
<h2><?php echo $title.' - '. $course_details['course_short_name'];?></h2>
<ol class="breadcrumb">
<li class="active">
<a href="<?php echo base_url('')?>">Home</a>
</li>
<li>
<a>Grade Settings</a>
</li>
<li class="active">
<strong><a href="<?php echo base_url('update_scale/'. $gradebook_details['
course_id'];?>">Update scale</a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-10">
<div class="form-group">
<?php echo form_open('modify_grading_scale');?>
<br/>
<h4>Rules on updating grading scale</h4>
<small>- The scale labels should be unique</small><br/>
<small>- The last row on the lower limit column must be equal to 0</small>
<br/>
<small>- The values assigned on the higher rows must be greater than the
rows below it</small><br/><br/>
<div class="table-responsive">
<input type="hidden" name="course_id" value="<?php echo $gradebook_details
['course_id'];?>" />
<br/>
<button type="button" class="btn btn-success" id="add"><i class="fa fa-
plus-square"></i> Add new row</button>
<br/><br/>
<table class="table table-bordered" id="dynamic_field">
<tr>
<th class="text-center" id="greaterthanorequalto">Scale Labels</th>
<th class="text-center" id="greaterthanorequalto">Scale Lower Limit</th>
<td></td>
<th class="text-center" id="greaterthanorequalto">Scale Higher Limit</th>
<td></td>
</tr>
</tr>
<?php $index = 0;?>
<?php if (isset($grading_scales)) {?>
<?php $size = count($grading_scales);?>
<?php foreach ($grading_scales as $label) {?>
<tr id="row<?php echo $index;?>">
<input type="hidden" value="<?php echo $label['scale_values_id'];?>" name="
scale_values_id[]" />
<td><input type="text" name="scale_values_label[]" value="<?php echo
$label['scale_values_label'];?>" placeholder="Label" class="form-
control label_list" required="required" /></td>
```

```

<td><input type="text" name="scale_values_lower_limit[]" value="<?php_echo
    _$label['scale_values_lower_limit'];?>" placeholder="LowerLimit_
    Value" class="form-control_value_list" required="required" /></td>
<td class="text-center" id="greaterthanorequalto"></td>
<td><input type="text" name="scale_values_higher_limit[]" value="<?php_
    echo_$label['scale_values_higher_limit'];?>" placeholder="Lower_
    Limit_Value" class="form-control_value_list" required="required"
    /></td>
<td><button type="button" name="remove" id="<?php_echo_$index;?>" class="
    btn_btn-danger_btn_remove">X</button></td>
</tr>
<?php $index++; ?>
<?php } ?>
<?php } ?>
</table>
<a class="btn_btn-white" href="<?php_echo_base_url('gradebook_setup/'.
    _$gradebook_details['gradebook_setup_details_id']);?>" />Cancel</a>
<input type="submit" name="submit" id="submit" class="btn_btn-info" value=
    "Submit" />
</div>
<?php echo form_close();?>
</div>
</div>
</div>
<div class="footer">
<div>
<strong>&#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>

```

Upload Accounts View

```

<div id="page-wrapper" class="gray-bg">
<div class="row_border-bottom">
<nav class="navbar_navbar-static-top" role="navigation" style="margin-
    bottom:0">
<div class="navbar-header">
<a class="navbar-minimalize_minimalize-styl-2_btn_btn-primary" href="#">
    i class="fa_fa-bars"></i> </a>
</div>
<ul class="nav_navbar-top-links_navbar-right">
<li>
<span class="m-r-sm_text-muted_welcome-message"><?php echo $school_name
    ;?></span>
</li>
<li class="dropdown">
<a class="dropdown-toggle_count-info" data-toggle="dropdown" href="#">
<i class="fa_fa-bell"></i> <span class="label_label-primary">8</span>
</a>
<ul class="dropdown-menu_dropdown-alerts">
<li>
<a href="mailbox.html">
<div>
<i class="fa_fa-envelope_fa-fw"></i> You have 16 messages
<span class="pull-right_text-muted_small">4 minutes ago</span>
</div>
</a>
</li>

```

```

</li class="divider"></li>
<li>
<a href="profile.html">
<div>
<i class="fa_fa-twitter_fa-fw"></i> 3 New Followers
<span class="pull-right_text-muted_small">12 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<a href="grid_options.html">
<div>
<i class="fa_fa-upload_fa-fw"></i> Server Rebooted
<span class="pull-right_text-muted_small">4 minutes ago</span>
</div>
</a>
</li>
<li class="divider"></li>
<li>
<div class="text-center_link-block">
<a href="notifications.html">
<strong>See All Alerts</strong>
<i class="fa_fa-angle-right"></i>
</a>
</div>
</li>
</ul>
</li>
<li>
<a href="<?php_echo_base_url('logout');?>">
<i class="fa_fa-sign-out"></i> Log out
</a>
</li>
</ul>
</nav>
</div>
<div class="row_wrapper_border-bottom_white-bg_page-heading">
<div class="col-sm-6">
<h2><?php echo $title;?></h2>
<ol class="breadcrumb">
<li>
<a href="<?php_echo_base_url('');?>">Home</a>
</li>
<li>
<a>Courses</a>
</li>
<li class="active">
<strong><a href="<?php_echo_base_url('upload_accounts');?>">Upload
    accounts</a></strong>
</li>
</ol>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="ibox_float-e-margins">
<div class="ibox-title">
Excel/CSV Upload (Header Columns: First Name, Last Name, Email Address)
<div class="ibox-tools">

```

```

<a class="collapse-link">
<i class="fa fa-chevron-up"></i>
</a>
</div>
</div>
<div class="ibox-content">
<?php if($this->session->flashdata('upload_account_error')) { ?>
<div class="alert alert-danger"><strong>Error!</strong>
<p><?php print_r($this->session->flashdata('upload_account_error'));?</p>
</div>
<?php } else if ($this->session->flashdata('upload_header_error')) { ?>
<div class="alert alert-danger"><strong>Error!</strong>
<p><?php print_r($this->session->flashdata('upload_header_error'));?</p>
</div>
<?php } else if ($this->session->flashdata('upload_count_error')) { ?>
<div class="alert alert-danger"><strong>Error!</strong>
<p><?php print_r($this->session->flashdata('upload_count_error'));?</p>
</div>
<?php } else if ($this->session->flashdata('upload_excel_error')) { ?>
<div class="alert alert-danger"><strong>Error!</strong>
<p><?php print_r($this->session->flashdata('upload_excel_error'));?</p>
</div>
<?php } ?>
<?php echo form_open_multipart('account_file_contents', array('class' => '
form-horizontal'));?>
<input type="hidden" name="school_id" value="<?php echo $_SESSION['
logged_in']['school_id']?>">
<div class="form-group"><label class="col-sm-2 control-label">File Type</
label>
<div class="col-sm-6">
<select class="form-control" name="upload_file_type">
<option value="CSV">CSV (.csv)</option>
</select>
<div class="lightBoxGallery">Example Images: -
<a href="<?php echo base_url('assets/images/csv_image.jpg')?>" title="
Image from Unsplash" data-gallery="">CSV</a>
<!-- The Gallery as lightbox dialog, should be a child element of the
document body -->
<div id="blueimp-gallery" class="blueimp-gallery">
<div class="slides"></div>
<h3 class="title"></h3>
<a class="prev"> </a>
<a class="next"> </a>
<a class="close"> </a>
<a class="play-pause"></a>
<ol class="indicator"></ol>
</div>
</div>
</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group"><label class="col-sm-2 control-label">User Role</
label>
<div class="col-sm-6">
<select class="form-control" name="upload_user_role">
<option value="Student">Student</option>
<option value="Teacher">Teacher</option>
</select>
</div>
</div>

```

```

<div class="hr-line-dashed"></div>
<div class="form-group">
<label class="col-sm-2 control-label">File</label>
<div class="col-sm-6 top-margin">
<?PHP echo form_upload('userfile', '', 'id="user_file" class="user_file"')
?>
</div>
</div>
<div class="hr-line-dashed"></div>
<div class="form-group">
<div class="col-sm-4 col-sm-offset-2">
<a class="btn btn-white" href="<?php echo base_url('');?>">Cancel<a
<input class="btn btn-primary" type="submit" value="Submit"/>
</div>
</div>
<?php echo form_close();?>
</div>
</div>
<div class="footer">
<div>
<strong>#9400; 2017</strong> Electronic Class Record Inquiry System
</div>
</div>
</div>
</div>

```

Validate Email View

```

<div id="preloader"></div>
<div id="top"></div>
<div class="landing">
<div class="overlay">
<div class="container">
<div class="row">
<!-- /.signup form -->
<div class="col-md-offset-3 col-md-6">
<div class="logo wow"><a href="">&nbsp;Electronic Class
Record Inquiry System</a></div>
<div class="signup-header wow">
<h3 class="form-title text-center">Please enter your registered email
address to verify your account</h3>
<?php if($this->session->flashdata('search_error1') != '') { ?>
<div class="alert alert-danger" role="alert">
<strong>Error!</strong> <?php echo $this->session->flashdata('
search_error1'); ?>
</div>
<?php } ?>
<?php echo form_open('validate_email_check');?>
<div class="form-group">
<input class="form-control input-lg" name="email_address" id="name" type="
email" placeholder="Email Address" required/>
</div>
<div class="form-group last">
<input type="submit" class="btn btn-warning btn-block btn-lg" value="
Verify">
</div>

```



```

<div class="modal_fade" id="updateAnnouncementModal" tabindex="-1" role="
    dialog" aria-labelledby="addGradeItemModalLabel" aria-hidden="true">
<div class="modal-dialog modal-lg" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="exampleModallLabel">
Update current announcement
<button type="button" class="close" data-dismiss="modal" aria-label="Close
    ">
<span aria-hidden="true">&times;</span>
</button>
</h5>
</div>
<div class="modal-body">
<?php $attributes = array('class' => 'form-horizontal'); ?>
<?php echo form_open('update_announcements', $attributes);?>
<input type="hidden" name="course_id" value="<?php echo $course_details['
    course_id']?>">
<input type="hidden" name="announcement_id" value="<?php echo
    $announcement['announcement_id']?>">
<div class="form-group">
<label for="recipient-name" class="form-control-label">Subject:</label>
<input type="text" class="form-control" required="required" name="
    announcement_subject" id="announcement_subject">
</div>
<div class="form-group">
<label for="recipient-name" class="form-control-label">Message:</label>
<textarea class="form-control" required="required" name="
    announcement_content" id="announcement_content"></textarea>
</div>
<p></p>
<div class="modal-footer">
<input type="submit" class="btn btn-primary" value = "Submit"></input>
<button type="button" class="btn btn-default" id="disagreeButton" data-
    dismiss="modal">Close</button>
<?php echo form_close();?>
</div>
</div>
</div>
</div>

```

Admin Header View

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="description" content="">
<meta name="author" content="">
<title><?php echo $title;?> | Electronic Class Record Inquiry System</
    title>
<link href="<?php echo base_url('assets/backyard/css/bootstrap1.css');?>"
    rel="stylesheet" media="screen">
<link href="<?php echo base_url('assets/backyard/css/font-awesome.min.css
    ');?>" rel="stylesheet">
<?php if($title === "View_list_of_courses" || $title === "View_list_of
    students" || $title === "View_list_of_teachers" || isset(
    $category_details) || $title === "Student_Groups" || $title === "
    Course_students_list" || strpos($title, "Course") !== FALSE) {?>

```

```

<link href="<?php echo base_url('assets/inspinia/css/plugins/dataTables/
   /dataTables.bootstrap1.css');?>" rel="stylesheet">
<link href="<?php echo base_url('assets/inspinia/css/plugins/dataTables/
   /dataTables.responsive.css');?>" rel="stylesheet">
<link href="<?php echo base_url('assets/inspinia/css/plugins/dataTables/
   /dataTables.tableTools.min.css');?>" rel="stylesheet">
<?php ?>
<link href="<?php echo base_url('assets/backyard/css/animate.css');?>" rel
    ="stylesheet" media="screen">
<?php if($title === "Add_course_category" || $title === "Add_student/s" ||
    $title === "Add_teacher/s" || $title === "Add_new_student_group"
    || $title === "Edit_Student_Details" || $title === "Edit_Teacher_
    Details" || $title === "Update_course_category" || $title === "
    Update_Student_Group" || $title === "Enrol_students" || $title ===
    "Enrol_student_groups") {?>
<link href="<?php echo base_url('assets/multiselect/css/multi-select3.css
    ');?>" rel="stylesheet">
<?php ?>
<?php if($title === "Upload_accounts" || $title === "Upload_courses") {?>
<link href="<?php echo base_url('assets/inspinia/css/plugins/blueimp/css/
    blueimp-gallery.min.css');?>" rel="stylesheet">
<?php ?>
<link href="<?php echo base_url('assets/inspinia/css/plugins/datapicker/
    datapicker3.css');?>" rel="stylesheet">
<link href="<?php echo base_url('assets/inspinia/css/plugins/toastr/toastr
    .min.css');?>" rel="stylesheet">
<link href="<?php echo base_url('assets/inspinia/css/style.css');?>" rel="
    stylesheet">
<link href="<?php echo base_url('assets/padding3.css');?>" rel="stylesheet
    " media="screen">
<?php if($title === "Change_password_(for_first-time_system_login)") {?>
<script type="text/javascript" src="<?php echo base_url('assets/
    password_check1.js');?>"></script>
<?php ?>
<script type="text/javascript" src="<?php echo base_url('assets/
    password_check.js');?>"></script>
<?php ?>
<script type="text/javascript" src="<?php echo base_url('assets/calx2/
    jquery-1.9.1.min.js');?>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/backyard/
    js/jquery.js');?>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/backyard/
    js/bootstrap.min.js');?>"></script>
<?php if($title === "Add_course_category" || $title === "Add_student/s" ||
    $title === "Add_teacher/s" || $title === "Add_new_student_group"
    || $title === "Edit_Student_Details" || $title === "Edit_Teacher_
    Details" || $title === "Update_course_category" || $title === "
    Update_Student_Group" || $title === "Enrol_students" || $title ===
    "Enrol_student_groups") {?>
<script type="text/javascript" src="<?php echo base_url('assets/
    multiselect/js/jquery.quicksearch.js');?>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/
    multiselect/js/jquery.multi-select.js');?>"></script>
<?php ?>
<?php if(strpos($title, "My_Course_Record") !== FALSE){ ?>
<link href="<?php echo base_url('assets/handsontable-pro-master/dist/
    handsontable.full.css');?>" rel="stylesheet" type="text/css">
<?php ?>
<?php else if (strpos($title, "Student_Record") !== FALSE) { ?>
<link href="<?php echo base_url('assets/handsontable-ruleJS/
    bower_components/handsontable/dist/handsontable.full.css');?>" rel=

```

```

"stylesheet" type="text/css">
<link href="<?php echo base_url('assets/handsontable-ruleJS/src/
handsontable.formula.css');>" rel="stylesheet" type="text/css">
<link href="<?php echo base_url('assets/handsontable-ruleJS/lib/highlight/
styles/github.css');>" rel="stylesheet" type="text/css">
<link href="<?php echo base_url('assets/handsontable-ruleJS/css/font-
awesome/css/font-awesome.min.css');>" rel="stylesheet" type="text/
css">
<?php } ?>
<style type="text/css">
.top-scrollbar-wrapper {
height: 20px;
border: none;
overflow-x: scroll;
overflow-y: hidden;
}
.top-scrollbar-wrapper .top-scrollbar {
height: 20px;
}
</style>
<!--[if lt IE 9]>
<script src="js/html5shiv.js"></script>
<script src="js/respond.min.js"></script>
<![endif]-->
</head><!--/head-->
<body>

```

Admin Footer View

```

<script type="text/javascript" src="<?php echo base_url('assets/backyard/
js/jquery.sticky.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/backyard/
js/wow.min.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/inspinia/
js/plugins/metisMenu/jquery.metisMenu.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/inspinia/
js/plugins/slimscroll/jquery.slimscroll.min.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/inspinia/
js/plugins/datapicker/bootstrap-datepicker.js');>"></script>
<?php if($title=="View list of courses"||$title=="View list of
students"||$title=="View list of teachers"||isset(
$category_details)||$title=="Student Groups"||$title=="
Course students list"||strpos($title,"Course")!==FALSE){?>
<script type="text/javascript" src="<?php echo base_url('assets/inspinia/
js/plugins/dataTables/jquery.dataTables.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/inspinia/
js/plugins/dataTables/dataTables.bootstrap.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/inspinia/
js/plugins/dataTables/dataTables.responsive.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/inspinia/
js/plugins/dataTables/dataTables.tableTools.min.js');>"></script>
<?php }?>
<?php if($title=="Upload accounts"||$title=="Upload courses")?>
<script type="text/javascript" src="<?php echo base_url('assets/inspinia/
js/plugins/blueimp/jquery.blueimp-gallery.min.js');>"></script>
<?php }?>
<?php if(strpos($title,"Student Record")!==FALSE){?>
<script type="text/javascript" src="<?php echo base_url('assets/
handsontable-ruleJS/bower_components/handsontable/dist/handsontable.
full.js');>"></script>

```

```

<script type="text/javascript" src="<?php echo base_url('assets/
handsontable-ruleJS/dist/lib/lodash/lodash.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/
handsontable-ruleJS/dist/lib/underscore.string/underscore.string.js'
);>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/
handsontable-ruleJS/dist/lib/moment/moment.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/
handsontable-ruleJS/dist/lib/numeral/numeral.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/
handsontable-ruleJS/dist/lib/numericjs/numeric.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/
handsontable-ruleJS/dist/lib/js-md5/md5.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/
handsontable-ruleJS/dist/lib/jstat/jstat.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/
handsontable-ruleJS/dist/lib/formulajs/formula.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/
handsontable-ruleJS/dist/js/parser.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/
handsontable-ruleJS/dist/js/ruleJS.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/
handsontable-ruleJS/src/handsontable.formula.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/
handsontable-ruleJS/lib/samples.js');>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/
handsontable-ruleJS/lib/highlight/highlight.pack.js');>"></script>
<script type="text/javascript">
var $$=function(id){
return document.getElementById(id);
}
function isArray(value,array){
return array.indexOf(value)>=0;
}
var container=$$('.example');
var test=$($category_total);
max=Math.max.apply(null,test);
test.push(max+1);
var test2=$[max+2];
var id=$($course_details['course_id']);
var record=$($class_record);
var count=$($count);
var rows_drop=$($dropped_rows);
count=parseInt(count);
console.log(rows_drop);
function firstRowRenderer(instance,td,row,col,prop,value,cellProperties){
Handsontable.renderers.TextRenderer.apply(this,arguments);
td.style.fontWeight='bold';
}
function secondRowRenderer(instance,td,row,col,prop,value,cellProperties){
Handsontable.renderers.TextRenderer.apply(this,arguments);
td.style.fontWeight='bold';
td.style.borderBottom='1px solid #000';
td.style.borderTop='1px solid #000';
td.style.backgroundColor='EEE';
}
function finalGradeRenderer(instance,td,row,col,prop,value,cellProperties){

```

```

Handsontable.renderers.TextRenderer.apply(this, arguments);
td.style.fontWeight = 'bold';
td.style.color = 'green';
td.style.background = '#EEE';
}

function scaledGradeRenderer(instance, td, row, col, prop, value,
    cellProperties) {
Handsontable.renderers.TextRenderer.apply(this, arguments);
if (parseInt(value, 10) < 0) {
// add class "negative"
td.className = 'make-me-red';
}
if (!value || value === '') {
td.style.background = '#EEE';
}
else {
if (value === 'Nissan') {
td.style.fontStyle = 'italic';
}
td.style.background = '';
}
}

function hiddenColumnRenderer(instance, td, row, col, prop, value,
    cellProperties){
Handsontable.renderers.TextRenderer.apply(this, arguments);
if (col == 0 || col == 1){
td.hidden = true;
}
}

function droppedStudentRenderer(instance, td, row, col, prop, value,
    cellProperties) {
Handsontable.renderers.TextRenderer.apply(this, arguments);
td.style.fontWeight = 'bold';
td.style.color = 'red';
td.style.background = '#EEE';
}

Handsontable.renderers.registerRenderer('droppedStudentRenderer',
    droppedStudentRenderer);

Handsontable.renderers.registerRenderer('hiddenColumnRenderer',
    hiddenColumnRenderer);

Handsontable.renderers.registerRenderer('finalGradeRenderer',
    finalGradeRenderer);

Handsontable.renderers.registerRenderer('scaledGradeRenderer',
    scaledGradeRenderer);

Handsontable.renderers.registerRenderer('secondRowRenderer',
    secondRowRenderer);

console.log(record);
var hot = new Handsontable(container, {
data: record,
colHeaders: false,
rowHeaders: false,
contextMenu: false,
fixedColumnsLeft:2,
formulas: true,
manualColumnResize: true,
manualRowResize: true,
observeChanges: true,
className: "htCenter",
cells: function(row, col, prop) {
var cellProperties = {};
if (isArray(col, test)){
cellProperties.readOnly = 'true';
}
if (col == 0 || row == 0 || row == 1){
cellProperties.readOnly = 'true';
}
if (row == 1){
cellProperties.renderer = secondRowRenderer;
}
if (row == 0 && this.instance.getDataAtCell(row, col).indexOf("Total") !=
-1){
cellProperties.renderer = firstRowRenderer;
}
if (row == 0 && this.instance.getDataAtCell(row, col).indexOf("Final_Grade")
!= -1){
cellProperties.readOnly = 'true';
}
if (isArray(col, test) && row > 1){
cellProperties.readOnly = 'true';
}
if (isArray(col, test2) && row > 1){
cellProperties.readOnly = 'true';
}
if (col == 1)
{
cellProperties.readOnly = 'true';
cellProperties.renderer = 'finalGradeRenderer';
}
if (col == 1 && isArray(row, rows_drop))
{
cellProperties.readOnly = 'true';
cellProperties.renderer = 'droppedStudentRenderer';
}
if (row == (count - 1)){
cellProperties.readOnly = 'true';
}
return cellProperties;
},
beforeChange: function (changes, source) {
for (var i = changes.length - 1; i >= 0; i--) {
// gently don't accept the word "foo" (remove the change at index i)
if (isNaN(changes[i][3]) || changes[i][3] < 0 || changes[i][3] === '' ||
changes[i][3] === '\u') {
changes.splice(i, 1);
}
}
});
$(document).ready(function () {
// set an on click on the button
$("#loadTable1").click(function () {
// get the time if clicked via an ajax get query
$.ajax({
url: "<?php_echo_base_url('get_class_record/').$course_details['course_id
'];?>",
type: "POST",
data: {course_id: id},
dataType: "json",
contentType: "application/json, charset=UTF-8",
success: function(result) {

```

```

console.log(JSON.stringify(result));
console.log(result);
hot.loadData(result);
}
});
});
$("#loadTable").click();
});
$(document).ready(function () {
// set an on click on the button
$("#saveTable").click(function () {
// get the time if clicked via an ajax get query
$.ajax({
url: "<?php_echo_base_url('save_class_record');?>",
type: "POST",
data: {record: hot.getData(), course_id: id},
beforeSend: function(){
$('#save_icon').removeClass('fa_fa-save');
$('#save_icon').addClass('fa_fa-spinner_fa-spin');
},
success: function(result) {
console.log(result);
$('#save_icon').removeClass('fa_fa-spinner_fa-spin');
$('#save_icon').addClass('fa_fa-save');
$(document).ready(function() {
setTimeout(function() {
toastr.options = {
positionClass: 'toast-top-center',
closeButton: true,
progressBar: true,
showDuration: 0,
showMethod: 'slideDown',
timeOut: 5000
};
toastr.success('The class record has been saved.', 'Success!');
}, 1300);
});
},
error: function(result){
console.log('error');
}
});
});
});
</script>
<?php } ?>
<?php if(strpos($title, "MyCourseRecord") !== FALSE){ ?>
<script type="text/javascript" src="<?php_echo_base_url('assets/
handsontable-ruleJS/bower_components/handsontable/dist/handsontable.
full.js');>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/
handsontable-ruleJS/dist/lib/lodash/lodash.js');>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/
handsontable-ruleJS/dist/lib/underscore.string/underscore.string.js
');>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/
handsontable-ruleJS/dist/lib/moment/moment.js');>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/
handsontable-ruleJS/dist/lib/numeral/numeral.js');>"></script>

```

```

<script type="text/javascript" src="<?php_echo_base_url('assets/
handsontable-ruleJS/dist/lib/numericjs/numeric.js');>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/
handsontable-ruleJS/dist/lib/js-md5/md5.js');>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/
handsontable-ruleJS/dist/lib/jstat/jstat.js');>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/
handsontable-ruleJS/dist/lib/formulajs/formula.js');>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/
handsontable-ruleJS/dist/js/parser.js');>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/
handsontable-ruleJS/dist/js/ruleJS.js');>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/
handsontable-ruleJS/src/handsontable.formula.js');>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/
handsontable-ruleJS/lib/samples.js');>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/
handsontable-ruleJS/lib/highlight/highlight.pack.js');>"></script>
<script type="text/javascript">
window.TopScrollbar = window.TopScrollbar || {};
window.TopScrollbar = function (options) {
this.$fakeWrapper = getjQueryElement(options.fakeWrapper);
this.$fakeScrollableElement = getjQueryElement(options.
fakeScrollableElement);
this.$wrapper = getjQueryElement(options.wrapper);
this.$scrollableElement = getjQueryElement(options.scrollableElement);
this._addListeners();
function getjQueryElement(el) {
return el instanceof jQuery ? el : $(el);
}
};
window.TopScrollbar.prototype.refresh = function () {
if (this.$wrapper[0].scrollWidth <= this.$wrapper[0].clientWidth) {
this.$fakeWrapper.hide();
return;
}
this.$fakeWrapper.show();
this.$fakeWrapper.width(this.$wrapper.width());
this.$fakeScrollableElement.width(this.$scrollableElement.width());
};
window.TopScrollbar.prototype._addListeners = function () {
var self = this;
this.$wrapper.scroll(function () {
self.$fakeWrapper.scrollLeft(
$(this).scrollLeft()
);
});
this.$fakeWrapper.scroll(function () {
self.$wrapper.scrollLeft(
$(this).scrollLeft()
);
});
};
var $$ = function(id){
return document.getElementById(id);
}
function isArray(value, array) {
return array.indexOf(value) > -1;
}
var container = $$('example');

```

```

var test = <?php echo json_encode($category_total); ?>;
max = Math.max.apply(null, test);
test.push(max+1);
var test2 = [max+2];
var id= <?php echo json_encode($course_details['course_id']); ?>;
var record = <?php echo json_encode($class_record); ?>;
console.log(test2);
function firstRowRenderer(instance, td, row, col, prop, value,
    cellProperties) {
Handsontable.renderers.TextRenderer.apply(this, arguments);
td.style.fontWeight = 'bold';
}
function finalGradeRenderer(instance, td, row, col, prop, value,
    cellProperties) {
Handsontable.renderers.TextRenderer.apply(this, arguments);
td.style.fontWeight = 'bold';
td.style.color = 'green';
td.style.background = '#EEE';
}
function scaledGradeRenderer(instance, td, row, col, prop, value,
    cellProperties) {
Handsontable.renderers.TextRenderer.apply(this, arguments);
if (parseInt(value, 10) < 0) {
// add class "negative"
td.className = 'make-me-red';
}
if (!value || value === '') {
td.style.background = '#EEE';
}
else {
if (value === 'Nissan') {
td.style.fontStyle = 'italic';
}
td.style.background = '';
}
}
function hiddenRowsRenderer(instance, td, row, col, prop, value,
    cellProperties) {
Handsontable.renderers.TextRenderer.apply(this, arguments);
td.style.display = 'none';
}
function hiddenColumnRenderer(instance,td,row,col,prop,value,
    cellProperties){
Handsontable.renderers.TextRenderer.apply(this, arguments);
if(col == 0 || col == 1){
td.hidden = true;
}
}
Handsontable.renderers.registerRenderer('hiddenColumnRenderer',
    hiddenColumnRenderer)
Handsontable.renderers.registerRenderer('finalGradeRenderer',
    finalGradeRenderer);
Handsontable.renderers.registerRenderer('scaledGradeRenderer',
    scaledGradeRenderer);
Handsontable.renderers.registerRenderer('hiddenRowsRenderer',
    hiddenRowsRenderer);
var test = <?php echo json_encode($trimmed_index);?>;
var hot = new Handsontable(container, {
data: record,
colHeaders: false,
rowHeaders: false,
contextMenu: false,
fixedColumnsLeft: 2,
formulas: true,
manualColumnResize: true,
manualRowResize: true,
observeChanges: true,
readOnly: true,
className: "htCenter",
cells: function(row, col, prop) {
var cellProperties = {};
if(row == 0 && this.instance.getDataAtCell(row,col).indexOf("Total") !=
-1){
cellProperties.renderer = firstRowRenderer;
}
if(col == 1){
cellProperties.renderer = 'finalGradeRenderer';
}
if(isInArray(row,test)){
cellProperties.renderer = 'hiddenRowsRenderer';
}
return cellProperties;
}
});
var s = new window.TopScrollbar({
fakeWrapper: $('<div>.top-scrollbar-wrapper'),
fakeScrollableElement: $('<div>.top-scrollbar'),
wrapper: $('<div>#example>.div.ht_master.handsontable>.div'),
scrollableElement: $('<div>#example>.div.ht_master.handsontable>.div>.div'),
});
s.refresh();
</script>
<?php } ?>
<script type="text/javascript" src="<?php echo base_url('assets/inspinia/
js/inspinia.js');?>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/inspinia/
js/plugins/pace/pace.min.js');?>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/inspinia/
js/plugins/toastr/toastr.min.js');?>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/calx2/
jquery-calx-2.2.5.min.js');?>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/highcharts
/code/highcharts.js');?>"></script>
<script type="text/javascript" src="<?php echo base_url('assets/highcharts
/code/modules/histogram-bellcurve.js');?>"></script>
<script>
$('#sheet').calx();
<?php if(isset($title2)) { ?>
$('#category_names').on('change', function(){
var cat_value = $('#category_names').val();
window.location = '<?php echo base_url('grades_stats/'.$course_id.'?
cat_letter=');?>' + cat_value;
});
<?php } ?>
</script>
<?php if(isset($title2) && isset($_GET['cat_letter'])) {?>
<script type="text/javascript">
var cat_value = '<?=$_GET['cat_letter'];?>';
$('#category_names').find('option[value='+cat_value+']').attr('selected', '
selected');

```

```

var cat_name = $('#category_names_option:selected').text();
console.log(cat_name);
var array = <?=json_encode($dropped_students)?>;
console.log(array);
var drop_count = <?=count($dropped_students)?>;
var enrol_count = <?=count($enrolled_students)?>;
var total_count = drop_count + enrol_count;
var data2 = new Array();
var points = parseFloat($('#input[type=hidden][data-cell='+cat_value+'2']).val());
for(i = 0, j = 3; i < total_count; i++, j++){
if(!(array.includes(j))){
var value = parseFloat($('#input[type=hidden][data-cell='+cat_value+j+']').val());
console.log(cat_value + j);
data2.push(value);
}
}
console.log(data2);
var data = [3.5, 3, 3.2, 3.1, 3.6, 3.9, 3.4, 3.4, 2.9, 3.1, 3.7, 3.4, 3,
3, 4, 4.4, 3.9, 3.5, 3.8, 3.8, 3.4, 3.7, 3.6, 3.3, 3.4, 3, 3.4, 3.5,
3.4, 3.2, 3.1, 3.4, 4.1, 4.2, 3.1, 3.2, 3.5, 3.6, 3, 3.4, 3.5, 2.3,
3.2, 3.5, 3.8, 3, 3.8, 3.2, 3.7, 3.3, 3.2, 3.2, 3.1, 2.3, 2.8, 2.8,
3.3, 2.4, 2.9, 2.7, 2, 3, 2.2, 2.9, 2.9, 3.1, 3, 2.7, 2.2, 2.5,
3.2, 2.8, 2.5, 2.8, 2.9, 3, 2.8, 3, 2.9, 2.6, 2.4, 2.4, 2.7, 2.7, 3,
3.4, 3.1, 2.3, 3, 2.5, 2.6, 3, 2.6, 2.3, 2.7, 3, 2.9, 2.9, 2.5,
2.8, 3.3, 2.7, 3, 2.9, 3, 3, 2.5, 2.9, 2.5, 3.6, 3.2, 2.7, 3, 2.5,
2.8, 3.2, 3, 3.8, 2.6, 2.2, 3.2, 2.8, 2.8, 2.7, 3.3, 3.2, 2.8, 3,
2.8, 3, 2.8, 3.8, 2.8, 2.8, 2.6, 3, 3.4, 3.1, 3, 3.1, 3.1, 3.1, 2.7,
3.2, 3.3, 3, 2.5, 3, 3.4, 3];
Highcharts.chart('histogram', {
title: {
text: cat_name + ' Histogram'
},
xAxis: [{
title: { text: 'Histogram' },
alignTicks: false
}, {
alignTicks: false,
opposite: true
}],
yAxis: [{
title: { text: 'Grades' }
}, {
title: { text: ' ' },
opposite: true
}],
series: [{
name: 'Histogram',
type: 'histogram',
xAxis: 1,
yAxis: 1,
baseSeries: 's1',
zIndex: -1,
binWidth: 5
}, {
name: 'Data',
type: 'scatter',
data: data2,
id: 's1',

```

```

marker: {
radius: 1.5
}
}
});
</script>
<?php } ?>
<script type="text/javascript">
$('#data_50.input-daterange').datepicker({
keyboardNavigation: false,
forceParse: false,
autoclose: true
});
</script>
<script type="text/javascript" src="<?php echo base_url('assets/scripts1.
js');?>"></script>
</body>
</html>

```

Admin Sidebar View

```

<div id="wrapper">
<nav class="navbar-default navbar-static-side" role="navigation">
<div class="sidebar-collapse">
<ul class="nav" id="side-menu">
<li class="nav-header">
<div class="dropdown profile-element"> <span>
</span>
<a data-toggle="dropdown" class="dropdown-toggle" href="#">
<span class="clear"> <span class="block ui-t-xs"> <strong class="font-bold">
<?php echo $user_details['user_first_name'].' '.$user_details['
user_last_name'];?></strong>
</span> <span class="text-muted text-xs block"><?php echo $_SESSION['
logged_in']['user_role'];?> <b class="caret"></b></span> </span> </a>
</div>
<ul class="dropdown-menu animated fadeInRight ui-t-xs">
<li><a href="<?php echo base_url('account_settings');?>">Profile </a></li>
<li class="divider"></li>
<li><a href="<?php echo base_url('logout');?>">Logout </a></li>
</ul>
</div>
<div class="logo-element">
ECRIS
</div>
</li>
<?php if($title === "Admin Dashboard") { ?>
<li class="active">
<?php } else { ?>
<li>
<?php } ?>
<a href="<?php echo base_url('');?>"><i class="fa fa-th-large"></i> <span
class="nav-label">Dashboard </span> </a>
</li>
<?php if($title === "Add student/s" || $title === "View list of students"
|| $title === "Add teacher/s" || $title === "View list of teachers"
|| $title === "Upload accounts" || isset($headers) || $title === "
Student Groups" || $title === "Add new student group" || $title ===
"Edit Student Details") { ?>
<li class="active">
<?php } else { ?>

```

```

</li>
<?php } ?>
<a href="#"><i class="fa fa-user"></i> <span class="nav-label">Accounts</span><span class="fa arrow"></span></a>
<ul class="nav nav-second-level">
<?php if($title === "Addstudent/s" || $title === "Viewlistofstudents"
|| $title === "StudentGroups" || $title === "Addnewstudentgroup"
|| $title === "EditStudentDetails") { ?>
<li class="active">
<?php } else { ?>
<li>
<?php } ?>
<a href="#">Student<span class="fa arrow"></span></a>
<ul class="nav nav-third-level">
<?php if($title === "Addstudent/s") { ?>
<li class="active">
<?php } else { ?>
<li>
<?php } ?>
<a href="<?php echo base_url('student_form');?>">Add new student/s</a>
</li>
<?php if($title === "Viewlistofstudents" || $title === "EditStudent_
Details") { ?>
<li class="active">
<?php } else { ?>
<li>
<?php } ?>
<a href="<?php echo base_url('student_list');?>">View list of students</a>
</li>
<?php if($title === "StudentGroups" || $title === "Addnewstudentgroup
") { ?>
<li class="active">
<?php } else { ?>
<li>
<?php } ?>
<a href="<?php echo base_url('student_groups_list');?>">Student Groups</a>
</li>
</ul>
</li>
<?php if($title === "Addteacher/s" || $title === "Viewlistofteachers")
{ ?>
<li class="active">
<?php } else { ?>
<li>
<?php } ?>
<a href="#">Teacher<span class="fa arrow"></span></a>
<ul class="nav nav-third-level">
<?php if($title === "Addteacher/s") { ?>
<li class="active">
<?php } else { ?>
<li>
<?php } ?>
<a href="<?php echo base_url('teacher_form');?>">Add new teacher/s</a>
</li>
<?php if($title === "Viewlistofteachers") { ?>
<li class="active">
<?php } else { ?>
<li>
<?php } ?>
<a href="<?php echo base_url('teacher_list');?>">View list of teachers</a>

```

```

</li>
</ul>
</li>
<?php if($title === "Uploadaccounts" || isset($headers)) { ?>
<li class="active">
<?php } else { ?>
<li>
<?php } ?>
<a href="<?php echo base_url('upload_accounts');?>">Upload accounts</a></li>
</ul>
</li>
<?php if($title === "Addcourse/s" || $title === "Addcourse_category" ||
$title === "Viewlistofcourses" || $title === "Viewcourse_
categories" || isset($category_details) || $title === "Uploadcourses
" || isset($headers2) || $title === "EditCourseDetails" || $title
=== "Updatecourse_category" || $title === "Coursestudentslist"
|| $title === "Enrolstudents" || $title === "Enrolstudentgroups"
) { ?>
<li class="active">
<?php } else { ?>
<li>
<?php } ?>
<a href="#"><i class="fa fa-book"></i> <span class="nav-label">Courses</span><span class="fa arrow"></span></a>
<ul class="nav nav-second-level">
<?php if($title === "Addcourse/s") { ?>
<li class="active">
<?php } else { ?>
<li>
<?php } ?>
<a href="<?php echo base_url('course_form');?>">Add course/s</a></li>
<?php if($title === "Viewlistofcourses" || $title === "EditCourse_
Details" || $title === "Coursestudentslist" || $title === "Enrol_
students" || $title === "Enrolstudentgroups") { ?>
<li class="active">
<?php } else { ?>
<li>
<?php } ?><a href="<?php echo base_url('course_list');?>">View list of
courses</a></li>
<?php if($title === "Addcourse_category" || $title === "Viewcourse_
categories" || isset($category_details) || $title === "Updatecourse_
category") { ?>
<li class="active">
<?php } else { ?>
<li>
<?php } ?>
<a href="#">Course Category<span class="fa arrow"></span></a>
<ul class="nav nav-third-level">
<?php if($title === "Addcourse_category") { ?>
<li class="active">
<?php } else { ?>
<li>
<?php } ?>
<a href="<?php echo base_url('course_category_form');?>">Add course
category</a>
</li>
<?php if($title === "Viewcourse_categories" || isset($category_details)
|| $title === "Updatecourse_category") { ?>
<li class="active">

```

```

<?php } else { ?>
<li>
<?php } ?>
<a href="<?php_echo_base_url('course_category_list');?>">View list of
    categories</a>
</li>
</ul>
</li>
</ul>
</li>
</ul>
</div>
</nav>

```

Footer View

```

<script type="text/javascript" src="<?php_echo_base_url('assets/backyard/
    js/jquery.js');?>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/backyard/
    js/bootstrap.min.js');?>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/backyard/
    js/custom.js');?>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/backyard/
    js/jquery.sticky.js');?>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/backyard/
    js/wow.min.js');?>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/backyard/
    js/owl.carousel.min.js');?>"></script>
<script type="text/javascript" src="<?php_echo_base_url('assets/scripts.js
    ');?>"></script>
</body>
</html>

```

Header View

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <meta name="description" content="">
    <meta name="author" content="">
    <title><?php_echo $title;?> | Electronic Class Record Inquiry System</
        title>
    <link href="<?php_echo_base_url('assets/backyard/css/bootstrap1.
        css');?>" rel="stylesheet" media="screen">
    <link href="<?php_echo_base_url('assets/backyard/css/font-awesome.min
        .css');?>" rel="stylesheet">
    <link href="<?php_echo_base_url('assets/backyard/fonts/icon-7-stroke/
        css/pe-icon-7-stroke.css');?>" rel="stylesheet">
    <link href="<?php_echo_base_url('assets/backyard/css/animate.css');?>
        " rel="stylesheet" media="screen">
    <link href="<?php_echo_base_url('assets/backyard/css/owl.theme.css')
        ;?>" rel="stylesheet">
    <link href="<?php_echo_base_url('assets/backyard/css/owl.carousel.css
        ');?>" rel="stylesheet">
    <link href="<?php_echo_base_url('assets/backyard/css/index-green
        .css');?>" rel="stylesheet" media="screen">

```

```

<link href="<?php_echo_base_url('assets/padding.css');?>" rel="
    stylesheet" media="screen">
<link rel="stylesheet" href="http://fonts.googleapis.com/css?
    family=Lato:100,300,400,700,900,100italic,300italic,400
    italic,700italic,900italic" />
<?php if($title === "Change_password_(for_first-time_system_
    login)") {?>
    <script type="text/javascript" src="<?php_echo_base_url('
        assets/password_check1.js');?>"></script>
<?php } else { ?>
<script type="text/javascript" src="<?php_echo_base_url('
    assets/password_check.js');?>"></script>
<?php } ?>
<!--[if lt IE 9]>
    <script src="js/html5shiv.js"></script>
    <script src="js/respond.min.js"></script>
<![endif]-->
</head><!--/head-->
<body>

```

Student Sidebar View

```

<div id="wrapper">
<nav class="navbar-default navbar-static-side" role="navigation">
<div class="sidebar-collapse">
<ul class="nav" id="side-menu">
<li class="nav-header">
<div class="dropdown_profile-element"> <span>
</span>
<a data-toggle="dropdown" class="dropdown-toggle" href="#">
<span class="clear"> <span class="block m-t-xs"> <strong class="font-bold">
    <?php_echo $user_details['user_first_name'];?>.$user_details['
    user_last_name'];?></strong>
</span> <span class="text-muted text-xs block"><?php_echo $_SESSION['
    logged_in']['user_role'];?> <b class="caret"></b></span> </span> </a>
</div>
<ul class="dropdown-menu animated fadeInRight m-t-xs">
<li><a href="<?php_echo_base_url('account_settings');?>">Account Settings
    </a></li>
<li class="divider"></li>
<li><a href="<?php_echo_base_url('logout');?>">Logout </a></li>
</ul>
</div>
<div class="logo-element">
    ECRIS
</div>
</li>
<?php if($title === "Dashboard") { ?>
<li class="active">
<a href="<?php_echo_base_url('');?>"><i class="fa fa-th-large"></i> <span
    class="nav-label">Dashboard</span> </a>
</li>
<?php } else { ?>
<li>
<a href="<?php_echo_base_url('');?>"><i class="fa fa-th-large"></i> <span
    class="nav-label">Dashboard</span> </a>
</li>
<?php } ?>
<?php if(count($my_courses) > 0) { ?>

```



```

<?php if(strpos($title,"Course") === FALSE && strpos($title,"Student_
Record") === FALSE) { ?>
<li style="border-left:0;background-color:#2f4050;">
<a id="course_list_button" href="#" style="pointer-events:none;cursor:
default;"><i class="fa fa-book"></i> <span class="nav-label">Courses
</span></a>
<ul id="course_list_id" class="nav nav-second-level" style="display:block"
>
<?php foreach($my_courses as $key => $value){ ?>
<li><a href="<?php echo base_url('my_course_record/'.$value['course_id'])
;">">
<?php echo $value['course_short_name'].'_'. $value['section_name']; ?>
</a>
</li>
<?php } ?>
</ul>
</li>
<?php } else { ?>
<li class="active">
<a id="course_list_button" disabled href="#"><i class="fa fa-book"></i> <
span class="nav-label">Courses</span></a>
<ul id="course_list_id" class="nav nav-second-level" style="display:block"
>
<?php foreach($my_courses as $key => $value){ ?>
<li><a href="<?php echo base_url('my_course_record/'.$value['course_id'])
;">">
<?php echo $value['course_short_name'].'_'. $value['section_name']; ?>
</a>
</li>
<?php } ?>
</ul>
</li>
<?php }
} else { ?>
<li>
<a href="#"><i class="fa fa-book"></i> <span class="nav-label">Courses</
span></a>
</li>
<?php } ?>
</ul>
</div>
</nav>

```

Teacher Sidebar View

```

<div id="wrapper">
<nav class="navbar-default navbar-static-side" role="navigation">
<div class="sidebar-collapse">
<ul class="nav" id="side-menu">
<li class="nav-header">
<div class="dropdown profile-element"> <span>
</span>
<a data-toggle="dropdown" class="dropdown-toggle" href="#">
<span class="clear"> <span class="block m-t-xs"> <strong class="font-bold">
<?php echo $user_details['user_first_name'].'_'. $user_details['
user_last_name'];?></strong>
</span> <span class="text-muted text-xs block"><?php echo $_SESSION['
logged_in']['user_role'];?> <b class="caret"></b></span> </span> </a>
<ul class="dropdown-menu animated fadeInRight m-t-xs">

```

```

<li><a href="<?php echo base_url('account_settings');?>">Account Settings
</a></li>
<li class="divider"></li>
<li><a href="<?php echo base_url('logout');?>">Logout</a></li>
</ul>
</div>
<div class="logo-element">
ECRIS
</div>
</li>
<?php if($title === "Dashboard") { ?>
<li class="active">
<a href="<?php echo base_url('');?>"><i class="fa fa-th-large"></i> <span
class="nav-label">Dashboard</span> </a>
</li>
<?php } else { ?>
<li>
<a href="<?php echo base_url('');?>"><i class="fa fa-th-large"></i> <span
class="nav-label">Dashboard</span> </a>
</li>
<?php } ?>
<?php if(count($my_courses) > 0) { ?>
<?php if(strpos($title,"Course") === FALSE && strpos($title,"Student_
Record") === FALSE) { ?>
<li style="border-left:0;background-color:#2f4050;">
<a id="course_list_button" href="#" style="pointer-events:none;cursor:
default;"><i class="fa fa-book"></i> <span class="nav-label">Courses
</span></a>
<ul id="course_list_id" class="nav nav-second-level" style="display:block"
>
<?php foreach($my_courses as $key => $value){ ?>
<li><a href="<?php echo base_url('course_record_info/'.$value['course_id
']);?>">
<?php echo $value['course_short_name'].'_'. $value['section_name']; ?>
</a>
</li>
<?php } ?>
</ul>
</li>
<?php } else { ?>
<li class="active">
<a id="course_list_button" disabled href="#"><i class="fa fa-book"></i> <
span class="nav-label">Courses</span></a>
<ul id="course_list_id" class="nav nav-second-level" style="display:block"
>
<?php foreach($my_courses as $key => $value){ ?>
<li><a href="<?php echo base_url('course_record_info/'.$value['course_id
']);?>">
<?php echo $value['course_short_name'].'_'. $value['section_name']; ?>
</a>
</li>
<?php } ?>
</ul>
</li>
<?php }
} else { ?>
<li>
<a href="#"><i class="fa fa-book"></i> <span class="nav-label">Courses</
span></a>
</li>

```

```

<?php } ?>
<?php if(count($categories) > 0) {?>
<?php if($title === "Assigned_course_categories" || $title === "Gradebook_
Setup" || $title === "Modify_gradebook_categories" || $title === "
Modify_category_-" || $title === "List_of_gradebook_templates" ||
$title === "Update_grading_scale" || $title === "List_of_gradebook_
scales") { ?>
<li class="active">
<?php } else { ?>
<li>
<?php } ?>
<a href="#"><i class="fa fa-gear"></i> <span class="nav-label">Gradebook
Settings</span><span class="fa arrow"></span></a>
<ul class="nav nav-second-level">
<?php if($title === "Assigned_course_categories" || $title === "Gradebook_
Setup" || $title === "Modify_gradebook_categories" || $title === "
Modify_category_-" || $title === "Update_grading_scale") { ?>
<li class="active">
<?php } else { ?>
<li>
<?php } ?>
<a href="<?php echo base_url('gradebook_setup_list');?>">Course Categories
</a>
</li>
<?php if($title === "List_of_gradebook_scales") { ?>
<li class="active">
<?php } else { ?>
<li>
<?php } ?>
<a href="<?php echo base_url('scales_list');?>">Grading Scales</a>
</li>
</ul>
</li>
<?php } ?>
</ul>
</div>
</nav>

```

/ so that a different class/function is called than the one
/ corresponding to the URL.
/ Please see the user guide for complete details:
/ https://codeigniter.com/user_guide/general/routing.html
/ -----
/ RESERVED ROUTES
/ -----
/ There are three reserved routes:
/ \$route['default_controller'] = 'welcome';
/ This route indicates which controller class should be loaded if the
/ URI contains no data. In the above example, the "welcome" class
/ would be loaded.
/ \$route['404_override'] = 'errors/page_missing';
/ This route will tell the Router which controller/method to use if those
/ provided in the URL cannot be matched to a valid route.
/ \$route['translate_uri_dashes'] = FALSE;
/ This is not exactly a route, but allows you to automatically route
/ controller and method names that contain dashes. '-' isn't a valid
/ class or method name character, so it requires translation.
/ When you set this option to TRUE, it will replace ALL dashes in the
/ controller and method URI segments.
/ Examples: my-controller/index -> my_controller/index
/ my-controller/my-method -> my_controller/my_method
*/

Routes

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

/*
/
-----
/ URI ROUTING
/
-----
/ This file lets you re-map URI requests to specific controller functions.
/
/ Typically there is a one-to-one relationship between a URL string
/ and its corresponding controller class/method. The segments in a
/ URL normally follow this pattern:
/
/     example.com/class/method/id/
/
/ In some instances, however, you may want to remap this relationship

```

```

$route['default_controller'] = 'navigation';
$route['home'] = 'navigation/home';
$route['register'] = 'navigation/register';
$route['logout'] = 'navigation/logout';
$route['forgot_pass'] = 'navigation/forgot_pass';
$route['login'] = 'user/login_user';
$route['validate_email'] = 'navigation/validate_email';
$route['validate_email_check'] = 'admin/validate_email_check';
$route['register_admin'] = 'admin/register_account';
$route['verify_mail'] = 'admin/verify_mail';
$route['verify_email'] = 'admin/verify_email';
$route['verify_success/(.any)'] = 'admin/verify_success/$1';
$route['send_reset_mail'] = 'admin/send_reset_mail';
$route['reset_password/(.any)'] = 'admin/reset_password/$1';
$route['update_password'] = 'admin/update_password';
$route['reset_success'] = 'admin/reset_success';
$route['delete_student'] = 'admin/delete_student';
$route['delete_teacher'] = 'admin/delete_teacher';

```

```

$route['delete_course_category'] = 'admin/delete_course_category';
$route['delete_course'] = 'admin/delete_course';
$route['clear_course_record'] = 'admin/clear_course_record';

$route['student_form'] = 'admin/student_form';
$route['student_groups_form'] = 'admin/student_groups_form';
$route['teacher_form'] = 'admin/teacher_form';
$route['course_form'] = 'admin/course_form';
$route['course_category_form'] = 'admin/course_category_form';

$route['student_list'] = 'admin/student_list';
$route['student_groups_list'] = 'admin/student_groups_list';
$route['teacher_list'] = 'admin/teacher_list';
$route['course_list'] = 'admin/course_list';
$route['course_category_list'] = 'admin/course_category_list';

$route['category_courses/(:num)'] = 'admin/category_courses/$1';

$route['edit_student_details/(:num)'] = 'admin/edit_student_details/$1';
$route['edit_course_details/(:num)'] = 'admin/edit_course_details/$1';
$route['edit_teacher_details/(:num)'] = 'admin/edit_teacher_details/$1';
$route['edit_course_category/(:num)'] = 'admin/edit_course_category/$1';
$route['edit_student_group/(:num)'] = 'admin/edit_student_group/$1';

$route['course_students/(:num)'] = 'admin/course_students/$1';
$route['enrol_students/(:num)'] = 'admin/enrol_students/$1';
$route['enrol_student_groups/(:num)'] = 'admin/enrol_student_groups/$1';

$route['assign_students'] = 'admin/assign_students';
$route['assign_student_groups'] = 'admin/assign_student_groups';

$route['remove_student'] = 'admin/remove_student';
$route['remove_course_from_category'] = 'admin/remove_course_from_category
';
$route['remove_student_group'] = 'admin/remove_student_group';

$route['update_student'] = 'admin/update_student';
$route['update_teacher'] = 'admin/update_teacher';
$route['update_course'] = 'admin/update_course';
$route['update_course_category'] = 'admin/update_course_category';
$route['update_student_group'] = 'admin/update_student_group';

$route['upload_accounts'] = 'admin/upload_accounts';
$route['upload_courses'] = 'admin/upload_courses';

$route['account_file_contents'] = 'admin/account_file_contents';
$route['course_file_contents'] = 'admin/course_file_contents';

$route['add_multiple_accounts'] = 'admin/add_multiple_accounts';
$route['add_multiple_courses'] = 'admin/add_multiple_courses';

$route['add_course'] = 'admin/add_course';
$route['add_student'] = 'admin/add_student';
$route['add_student_group'] = 'admin/add_student_group';
$route['add_teacher'] = 'admin/add_teacher';
$route['add_course_category'] = 'admin/add_course_category';

$route['force_password_change'] = 'teacher/force_password_change';
$route['update_force_password'] = 'teacher/update_force_password';

$route['gradebook_setup_list'] = 'teacher/gradebook_setup_list';
$route['gradebook_setup/(:num)'] = 'teacher/gradebook_setup/$1';
$route['update_gradebook_settings'] = 'teacher/update_gradebook_settings';
$route['modify_gradebook_categories/(:num)'] = 'teacher/
modify_gradebook_categories/$1';
$route['modify_gradebook_items/(:num)/(:num)'] = 'teacher/
modify_gradebook_items/$1/$2';
$route['modify_grade_items'] = 'teacher/modify_grade_items';
$route['modify_grade_category'] = 'teacher/modify_grade_category';

$route['edit_grade_item'] = 'teacher/edit_grade_item';
$route['edit_gradebook_item'] = 'teacher/edit_gradebook_item';

$route['remove_grade_item'] = 'teacher/remove_grade_item';
$route['remove_gradebook_item'] = 'teacher/remove_gradebook_item';
$route['remove_gradebook_category'] = 'teacher/remove_gradebook_category';

$route['modify_category_items'] = 'teacher/modify_category_items';

$route['move_gradebook_items'] = 'teacher/move_gradebook_items';

$route['save_gradebook_template'] = 'teacher/save_gradebook_template';
$route['use_gradebook_template'] = 'teacher/use_gradebook_template';
$route['templates_list'] = 'teacher/gradebook_templates_list';

$route['remove_gradebook_template'] = 'teacher/remove_gradebook_template';
$route['change_template_name'] = 'teacher/change_template_name';

$route['update_scale/(:num)'] = 'teacher/update_grading_scale/$1';
$route['modify_grading_scale'] = 'teacher/modify_grading_scale';

$route['save_gradebook_scale'] = 'teacher/save_gradebook_scale';
$route['use_gradebook_scale'] = 'teacher/use_gradebook_scale';

$route['scales_list'] = 'teacher/gradebook_scales_list';
$route['remove_gradebook_scale'] = 'teacher/remove_gradebook_scale';
$route['change_scale_name'] = 'teacher/change_scale_name';

$route['assign_category_scale'] = 'teacher/assign_category_scale';

$route['course_record_info/(:num)'] = 'teacher/view_course_record_info/$1
';
$route['class_record/(:num)'] = 'teacher/class_record/$1';
$route['my_course_record/(:num)'] = 'student/my_course_record/$1';

$route['get_class_record/(:num)'] = 'ajax/get_class_record/$1';
$route['save_class_record'] = 'ajax/save_class_record';

$route['account_settings'] = 'user/account_settings';
$route['change_pass'] = 'user/change_pass';

$route['drop_student'] = 'teacher/drop_student';

$route['course_announcements/(:num)'] = 'user/course_announcements/$1';
$route['view_announcements/(:num)/(:num)'] = 'user/view_announcements/$1/
$2';
$route['remove_announcements'] = 'user/remove_announcements';
$route['submit_announcements'] = 'user/submit_announcements';
$route['update_announcements'] = 'user/update_announcements';

```

```
$route['grades_stats/(:num)'] = 'teacher/grades_stats/$1';
```

```
$route['404_override'] = '';
```

```
$route['translate_uri_dashes'] = FALSE;
```

XI. Acknowledgement

Six years ago noong una akong pumasok sa UPM. At ngayon na may pagkakataon na ko para isulat ito, papasalamatan ko na lahat ng taong nakasalamuha ko bago ako umabot sa puntong ito. Oo. Lahat kayo.

Unang-una sa lahat, sa Panginoong Diyos. Inabot man ako ng anim na taon na lagi kong tinatanong ang sarili ko kung magagawa ko ba ito, hindi ako nagduda at hindi ako nawalan ng tiwala sayo. Hindi magiging posible ito kung hindi dahil sayo.

Gusto ko rin pasalamatang ang pamilya ko sa walang sawang pagmahahal at pagtitiwala sa akin kahit nadelay na ako. Alam kong nagalit kayo sa akin noong nalaman niyo na madedelay ulit ako, pero nandiyan lang kayo para suportahan pa rin ako palagi. Kay mama, papa, lola at kapatid ko na naging inspirasyon ko para tapusin itong SP ko. Inaalay ko ito para sa inyo.

Gusto ko rin pasalamatang si Ma'am Avegail Carpio sa pagtitiwala sa akin na patuloy pa rin akong ginabayan kahit inabot ako ng 2 taon para matapos ang SP ko. Hanggang sa huli, nagtiwala pa rin kayo sa akin at pinayagan niyo akong makapagdefense kahit wala kayo. Thank you po at pasensya na din po sa abalang nagawa ko. Maraming salamat din kay Ma'am Perl Gasmen na naghanap ng oras para makapagdefense lang ako. Pasensya na po kung naabala ko kayo masyado noong araw na iyon.

Kay Diana Grace Ang, na nagtiis sa ugali ko at sinamahan ako sa lahat ng kalokohan ko. Maraming salamat sa pagmamahal na binigay mo, sa pagtitiis at pagmotivate mo sa akin kapag feeling ko na hindi ko na naman matatapos ito. Alam mong nandito lang ako para sayo.

Gusto ko rin magpasalamat kay Adriel (AD) na idol ko sa basketball. Isa siya sa mga naging inspirasyon ko para matapos ang SP na ito. Namotivate pa lalo ako na tapusin ito dahil nakita ko na binanggit niya rin ako sa acknowledgement niya. Ngayon, makakabawi na ako at malalagay ko na rin

yung pangalan mo dito. Salamat dreee.

Sa buong Block 12, Smokey's, at GH na naging dahilan para mabuo itong college life ko, Maraming maraming salamat sa inyo.

At higit sa lahat, ang UP Manila na nagturo sa akin kung paano maging mas matatag, hindi lang sa napakadaming exams at MPs, kundi pati na rin sa mga pagsubok sa buhay. Marmaing salamat.

Salamat, UP!