

APPENDICES

Sample Survey Questionnaire

Request Letter to Conduct Survey

Certification of Editing

Computation of Evaluation Results

Computation of Longitudinal Analysis

CONTROL				EXPERIMENTAL			
Time on Quiz	Time on Read	COMPE TENCY	PERFOR MANCE	Time on Quiz	Time on Read	COMPE TENCY	PERFOR MANCE
7	4	11	70	3	3	6	100
7	5	12	60	3	3	6	90
9	3	12	75	3	3	6	85
9	3	12	80	4	3	7	95
9	3	12	85	4	4	8	95
7	6	13	75	5	4	9	90
7	6	13	80	3	7	10	85
10	3	13	70	8	4	12	90
9	4	13	85	6	6	12	75
7	6	13	80	4	8	12	80
9	4	13	70	7	6	13	60
7	6	13	90	7	6	13	90
11	3	14	75	6	7	13	90
11	3	14	70	6	9	15	85
11	3	14	85	10	6	16	95
9	5	14	75	12	4	16	85
11	4	15	85	9	8	17	80
10	5	15	75	10	8	18	90
11	4	15	70	10	9	19	95
11	4	15	60	11	8	19	75
10	5	15	65	12	8	20	80
11	5	16	70	18	2	20	100
11	5	16	70	15	6	21	90
11	5	16	80	13	9	22	90
11	5	16	75	11	11	22	85
11	5	16	75	11	11	22	80
10	6	16	75	21	5	26	95
11	5	16	80	15	11	26	85
11	6	17	65	21	6	27	80
11	6	17	90	15	14	29	90

COMPETENCY

SS-TOTAL	SS-TREAT	SS-ERROR
17.2225	0.840277778	10.45444
9.9225	0.840277778	4.987778
9.9225		4.987778
9.9225		4.987778
9.9225		4.987778
4.6225		1.521111
4.6225		1.521111
4.6225		1.521111
4.6225		1.521111
4.6225		1.521111
4.6225		1.521111
1.3225		0.054444
1.3225		0.054444
1.3225		0.054444
1.3225		0.054444
0.0225		0.587778
0.0225		0.587778
0.0225		0.587778
0.0225		0.587778
0.0225		0.587778
0.7225		3.121111
0.7225		3.121111
0.7225		3.121111
0.7225		3.121111
0.7225		3.121111
0.7225		3.121111
0.7225		3.121111
3.4225		7.654444

3.4225	7.654444
83.7225	101.3378
83.7225	101.3378
83.7225	101.3378
66.4225	82.20444
51.1225	65.07111
37.8225	49.93778
26.5225	36.80444
9.9225	16.53778
9.9225	16.53778
9.9225	16.53778
4.6225	9.404444
4.6225	9.404444
4.6225	9.404444
0.0225	1.137778
0.7225	0.004444
0.7225	0.004444
3.4225	0.871111
8.1225	3.737778
14.8225	8.604444
14.8225	8.604444
23.5225	15.47111
23.5225	15.47111
34.2225	24.33778
46.9225	35.20444
46.9225	35.20444
46.9225	35.20444
117.7225	98.67111
117.7225	98.67111
140.4225	119.5378
191.8225	167.2711

COMPETENCY

Group Means		Grand Mean	15.15
Control	14.23333333	k	2
Experimental	16.06666667	n	30
		N(k * n)	60
		SS-Total	1415.65
		SS-Treat	50.41667
		SS-Error	1365.233

	df-total	59
	df-treat	1
	df-error	58
	MS-treat	50.41667
	MS-error	23.53851
	f-value	2.141881
	Crit f-value	4.01

PERFROMANCE

SS-TOTAL	SS-TREAT	SS-ERROR
122.8403	33.0625	28.44444
444.5069	33.0625	235.1111
37.00694		0.111111
1.173611		21.77778
15.34028		93.44444
37.00694		0.111111
1.173611		21.77778
122.8403		28.44444
15.34028		93.44444
1.173611		21.77778
122.8403		28.44444
79.50694		215.1111
37.00694		0.111111
122.8403		28.44444
15.34028		93.44444
37.00694		0.111111
15.34028		93.44444
37.00694		0.111111
122.8403		28.44444
444.5069		235.1111
258.6736		106.7778
122.8403		28.44444
122.8403		28.44444
1.173611		21.77778
37.00694		0.111111
37.00694		0.111111
37.00694		0.111111
1.173611		21.77778
258.6736		106.7778

79.50694	215.1111
357.8403	173.3611
79.50694	10.02778
15.34028	3.361111
193.6736	66.69444
193.6736	66.69444
79.50694	10.02778
15.34028	3.361111
79.50694	10.02778
37.00694	140.0278
1.173611	46.69444
444.5069	720.0278
79.50694	10.02778
79.50694	10.02778
15.34028	3.361111
193.6736	66.69444
15.34028	3.361111
1.173611	46.69444
79.50694	10.02778
193.6736	66.69444
37.00694	140.0278
1.173611	46.69444
357.8403	173.3611
79.50694	10.02778
79.50694	10.02778
15.34028	3.361111
1.173611	46.69444
193.6736	66.69444
15.34028	3.361111
1.173611	46.69444
79.50694	10.02778

PERFORMANCE

<i>Group Means</i>		Grand Mean	81.08333333
<i>Control</i>	75.33333333	k	2
<i>Experimental</i>	86.83333333	n	30
		N(k * n)	60
		SS-Total	5804.583333
		SS-Treat	1983.75
		SS-Error	3820.833333
		df-total	59
		df-treat	1
		df-error	58
		MS-treat	1983.75
		MS-error	65.87643678
		f-value	30.1131952
		Crit f-value	4.01

Note: FOR GRAPHING THE PERFORMANCE OVER COMPETENCY ONLY

CONTROL**EXPERIMENTAL**

<i>Competency</i>	Performance	Competency	Performance
11	70	6	91.66666667
12	71.66666667	7	95
13	78.57142857	8	95
14	76.25	9	90
15	71	10	85
16	75	12	81.66666667
17	77.5	13	80
		15	85
		16	90
		17	80
		18	90
		19	85
		20	90
		21	90
		22	85
		26	90
		27	80
		29	90

Competency t-Test Results

t-Test: Paired Two Sample for Means		
	Variable 1	Variable 2
Mean	14.23333333	16.06666667
Variance	2.805747126	44.27126437
Observations	30	30
Pearson Correlation	0.970064241	
Hypothesized Mean Difference	0	
df	29	
t Stat	-1.990324304	
P(T<=t) one-tail	0.028027635	
t Critical one-tail	1.699127027	
P(T<=t) two-tail	0.056055269	
t Critical two-tail	2.045229642	
Finding:		
There is no significant difference between traditional and proposed in terms of competency		

Performance t-Test Results

t-Test: Paired Two Sample for Means		
	Variable 1	Variable 2
Mean	75.33333333	86.83333333
Variance	61.95402299	69.79885057
Observations	30	30
Pearson Correlation	0.147699343	
Hypothesized Mean Difference	0	
df	29	
t Stat	-5.943130333	
P(T<=t) one-tail	9.30974E-07	
t Critical one-tail	1.699127027	
P(T<=t) two-tail	1.86195E-06	
t Critical two-tail	2.045229642	
Finding:		
There is a significant difference between traditional and proposed in terms of performance		

Traditional Method Competency and Performance Tabulation and Assessment Computation Results

Student Assigned Number	Time of Reading (mins.) (Started at 2:00 pm)	Time of Answering Quiz (mins.) (Started at 2:14 pm)	Quiz Score (Total: 10)	Score Percentage
#1	7	6	5	75%
#2	7	6	6	80%
#3	11	4	7	85%
#4	7	5	2	60%
#5	11	3	5	75%
#6	10	3	4	70%
#7	9	4	7	85%
#8	7	6	6	80%
#9	11	6	3	65%
#10	9	3	5	75%
#11	11	3	4	70%
#12	9	3	6	80%
#13	11	5	4	70%
#14	11	3	7	85%
#15	9	3	7	85%
#16	9	4	4	70%
#17	9	5	5	75%
#18	7	6	8	90%
#19	11	6	8	90%
#20	10	5	5	75%
#21	11	4	4	70%
#22	11	5	4	70%
#23	11	5	6	80%
#24	11	4	2	60%
#25	11	5	5	75%
#26	7	4	4	70%
#27	11	5	5	75%
#28	10	6	5	75%
#29	11	5	6	80%
#30	10	5	3	65%

Traditional Method Performance Assessment		
Score in Quiz	Frequency	Percentage
60	2	7%
65	2	7%
70	7	23%
75	8	27%
80	5	17%
85	4	13%
90	2	7%
TOTAL	30	100%
TOTAL		

TRADITIONAL METHOD TIME OF READING IN MINUTES		
Time of Reading in mins	Frequency	Percentage
7	6	20%
9	6	20%
10	4	13%
11	14	47%
TOTAL	30	100%
TOTAL		

TRADITIONAL METHOD TIME OF QUIZ IN MINUTES		
Time of answering quiz in mins	Frequency	Percentage
3	7	23%
4	6	20%
5	10	33%
6	7	23%
TOTAL	30	100%
TOTAL		

Proposed System Competency and Performance Tabulation and Assessment Computation Results

Student Assigned Number	Time of Reading (mins.)	Time of Answering Quiz (mins.)	Quiz Score (Total: 10)	Score Percentage
#1	6	9	7	85%
#2	9	8	6	80%
#3	7	6	2	60%
#4	10	9	9	95%
#5	7	6	8	90%
#6	15	6	8	90%
#7	12	8	6	80%
#8	6	7	8	90%
#9	8	4	8	90%
#10	6	6	5	75%
#11	11	8	5	75%
#12	21	5	9	95%
#13	10	8	8	90%
#14	4	3	9	95%
#15	3	3	10	100%
#16	13	9	8	90%
#17	10	6	9	95%
#18	4	4	9	95%
#19	15	11	7	85%
#20	21	6	6	80%
#21	3	3	8	90%
#22	4	8	6	80%
#23	3	7	7	85%
#24	18	2	10	100%
#25	12	4	7	85%
#26	5	4	8	90%
#27	3	3	7	85%
#28	11	11	7	85%
#29	15	14	8	90%
#30	11	11	6	80%

Proposed System Performance Assessment		
Score in Quiz	Frequency	Percentage
60	1	3%
75	2	7%
80	5	17%
85	6	20%
90	9	30%
95	5	17%
100	2	7%
TOTAL	30	100%
TOTAL		

PROPOSED SYSTEM TIME OF READING IN MINUTES		
Time of Reading in mins	Frequency	Percentage
3	4	13%
4	3	10%
5	1	3%
6	3	10%
7	2	7%
8	1	3%
9	1	3%
10	3	10%
11	3	10%
12	2	7%
13	1	3%
15	3	10%
18	1	3%
21	2	7%
TOTAL	30	100%
TOTAL		

PROPOSED SYSTEM TIME OF QUIZ IN MINUTES		
Time of answering quiz in mins	Frequency	Percentage
2	1	3%
3	4	13%
4	4	13%
5	1	3%
6	6	20%
7	2	7%
8	5	17%
9	3	10%
11	3	10%
14	1	3%
TOTAL	30	100%
TOTAL		

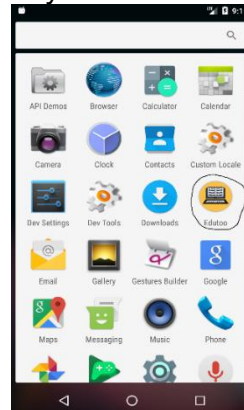
User's Manual

Student Module

Login/Register

Find the App?

Once installed, find the app on your mobile. It is a named “Edutoo”



How to Register?

Supply the needed information: First Name and Last Name, a valid e-mail address, and your desired password. After completing the form, click “Save” and a verification/confirmation will be sent to your email.

9:16

wjalbao@gmail.com

.....

REGISTER

[Already registered?](#)

9:34

Register Your Information

First Name

Juan

Last Name

Two

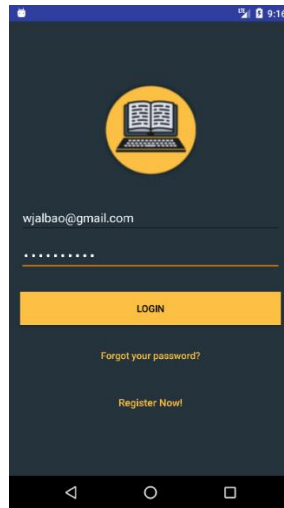
☐ Male

☒ Female

SAVE

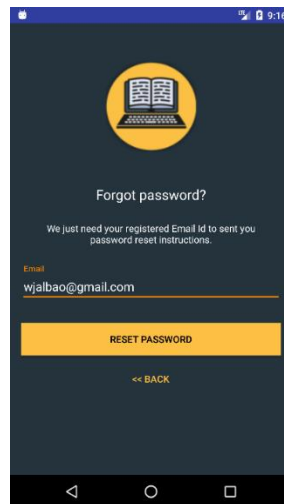
How to Log-in?

After you verified your account using email, you may start accessing the app. Click “LOGIN” to start.



How to reset my password in case I forgot?

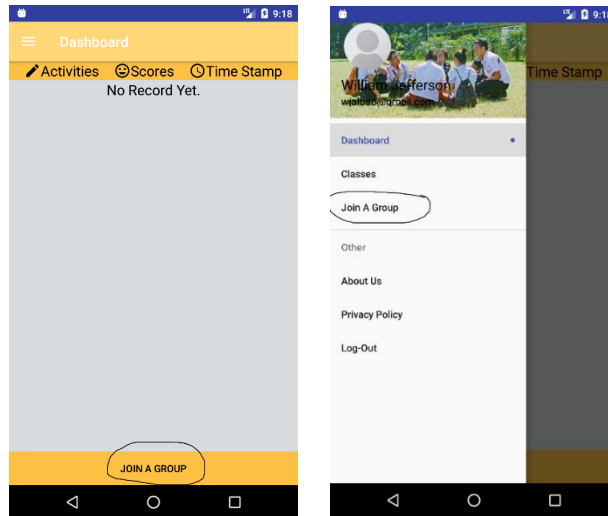
Supply your e-mail address and then click “RESET PASSWORD” button; after this an e-mail is sent to you. Click the link given in the e-mail to start creating a new password.



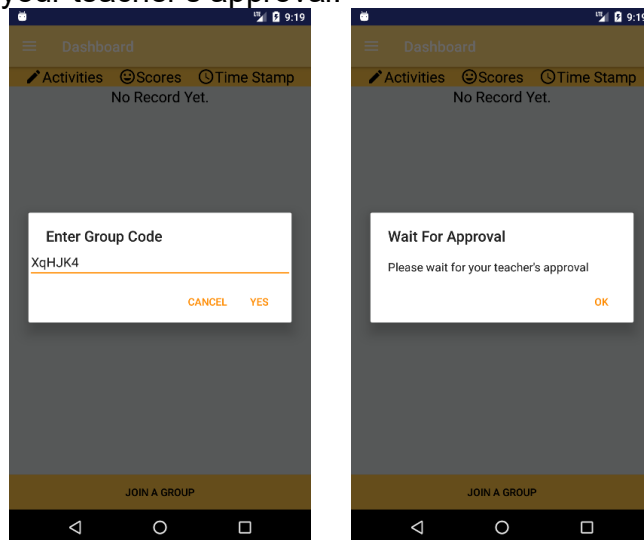
Joining a Group

How to join a group?

You have two options to join a group. If you're a new user, a “JOIN A GROUP” button is visible to the dashboard. If you're already a member of a group and wants to join another group, swipe the screen to the right, you'll see a “Join A Group” Button.

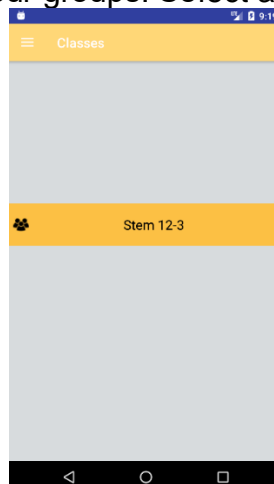


To join a group, you must have a group code provided by your teacher. Enter it then wait for your teacher's approval.

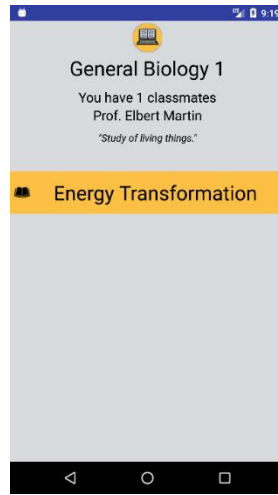


How to start studying?

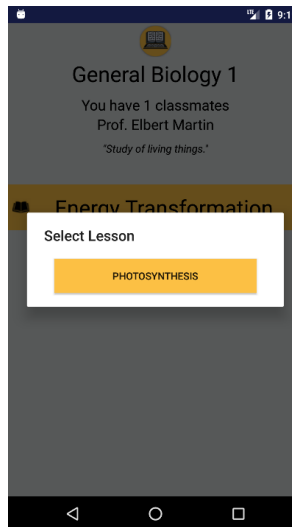
Once approved, swipe the screen to the right and you'll see a "Classes" button. Tap it then you'll see all your groups. Select a group to see all chapters.



Once you select a group, you'll see the Subject name, teacher, chapters, and description. You'll also see the number of classmates you have. Select a chapter.

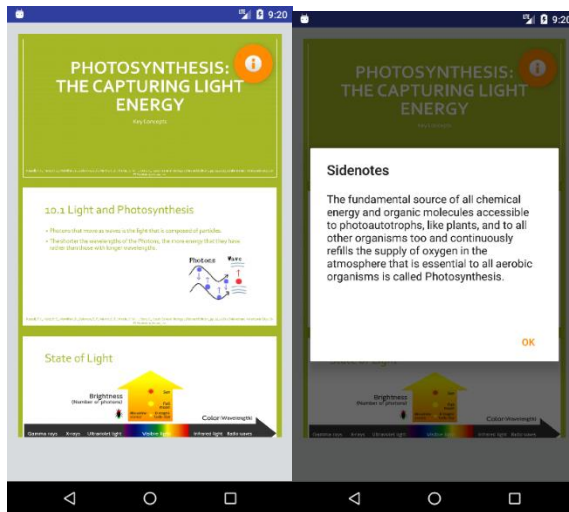


Once you select a chapter, you'll see the list of all available lessons and chapter quizzes. Select a lesson.

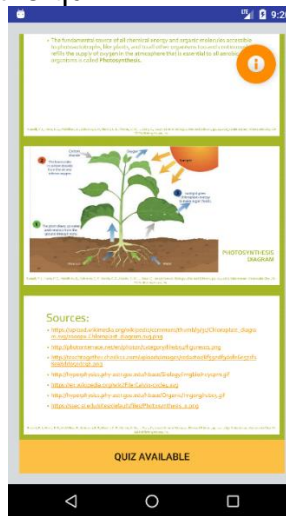


How to read a lesson?

Once you select a lesson, you will be redirected to the reading of lesson screen. You can swipe, double tap and zoom the lesson material. You can also read some side notes using the orange circle floating button above. You can take the quiz if you read all the slide or simply just slide down.



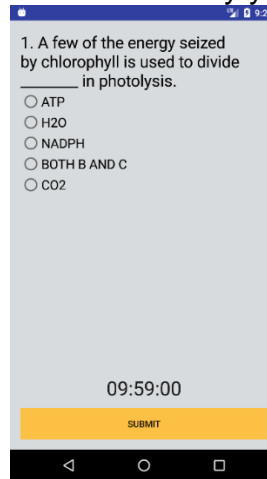
Once done reading, click “QUIZ AVAILABLE” to take the quiz. There is a confirmation before taking up the quiz.



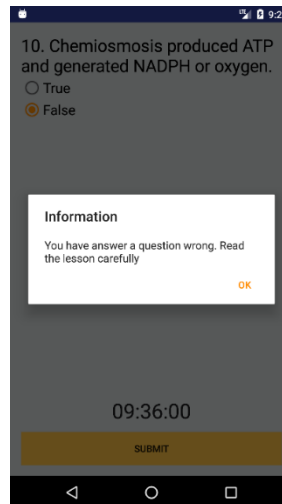
Note: You cannot go back to reading the lesson once you’ve decided to take the quiz unless you made a mistake at the quiz.

How to answer the quiz?

Once you decided to take the quiz, you will see the question, choices, a timer and the submit button. Questions are viewed using different pages. The questions, choices and the timer are created by your teacher. Answer the quiz.

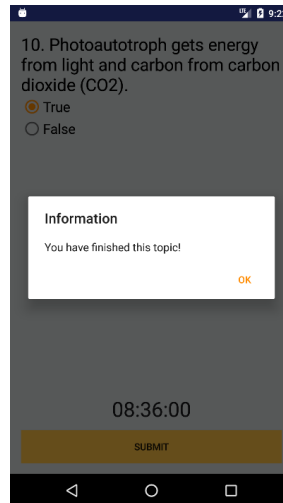


Once you've answered all the questions but committed a mistake. You will be redirected to the reading of lesson where you'll be able to further read the lesson the took up the quiz again.



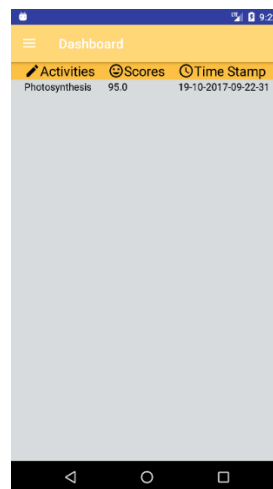
Note: You cannot disregard the quiz until you finish the quiz with no wrong answer/s. Every mistake will deduct 5 points in your total grade.

Once you've answered all the questions with no mistakes. You'll be redirected to the chapters screen.



How to view my scores?

Go back to the dashboard, if you already took atleast one quiz, your grade will be seen there.



What is “About Us”?

Swipe the screen to the right and click “About Us”. There you’ll see about the system.



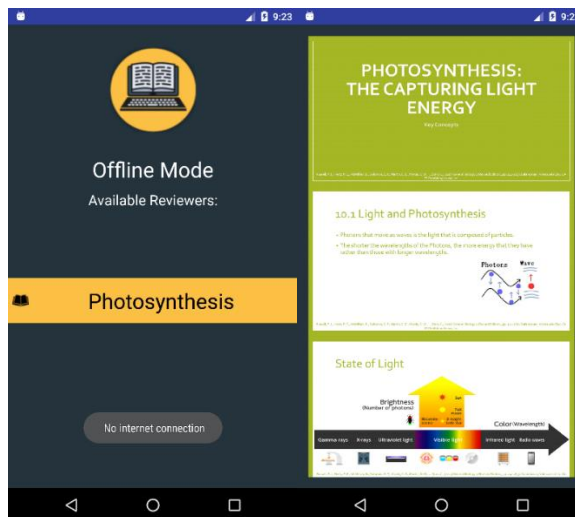
What is “Privacy Policy”?

Swipe the screen to the right and click “Privacy Policy”. There you’ll see about how the application/system protect your personal informations.



How to read a lesson without internet?

When there is no internet connection, the application will automatically go to offline mode. There you will see the lesson you have already saved, ready to be reviewed.

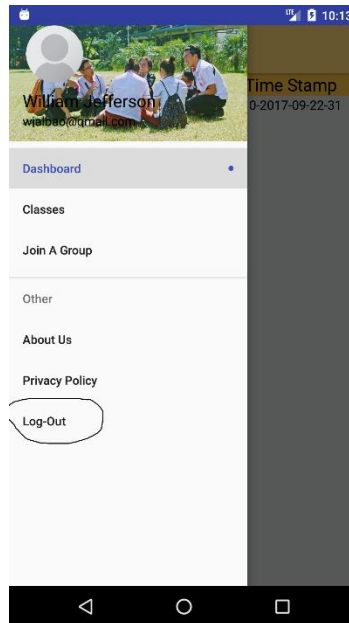


Note: Lessons are only available for offline mode if you have already read it atleast once.

Logout

How to logout on student's module?

Swipe the screen to the right then click “Logout” button on the bottom navigation drawer.

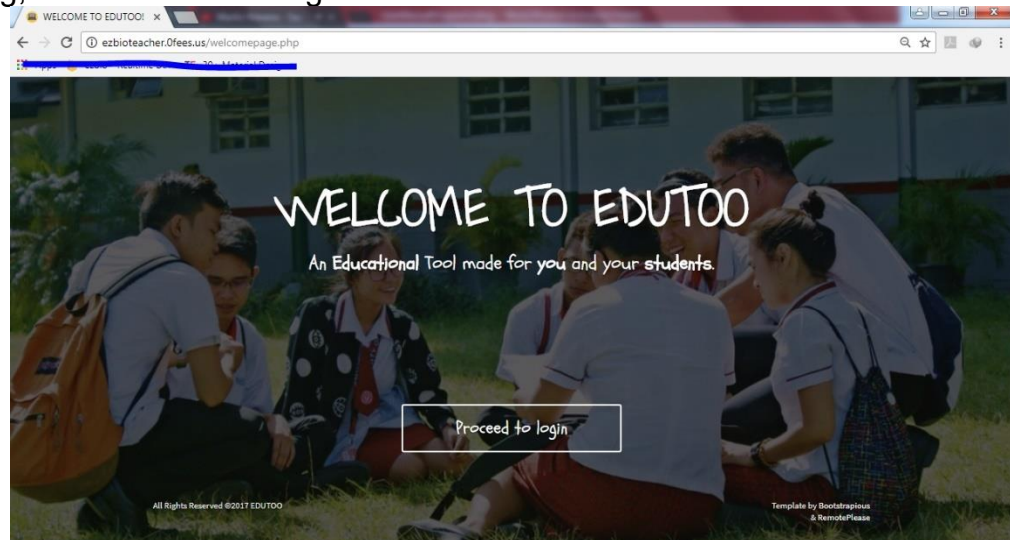


Teacher's Module

Login/Register

How to get to the teacher's module?

Type `ezbioteacher.Ofees.us` on your address bar. After the page started loading, click "Proceed to login".



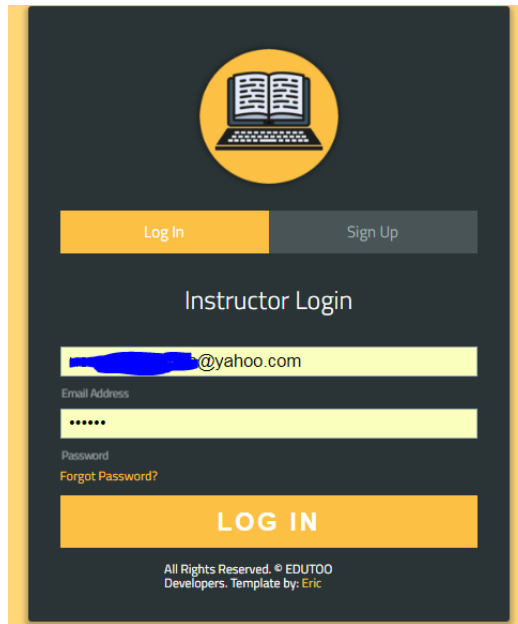
How to Register?

Supply the needed information: First Name and Last Name, a valid e-mail address, and your desired password. After completing the form, click "Get Started" and a verification/confirmation will be sent to your email.

A registration form with a dark gray background and yellow accents. At the top is a yellow circular icon containing a laptop with an open book. Below the icon are two buttons: 'Log In' (gray) and 'Sign Up' (yellow). The section is titled 'REGISTER HERE' in white. The form includes input fields for 'First Name' (containing 'Juan'), 'Last Name' (containing 'Peccato'), and 'Email Address' (containing 'juanpeccato@yahoo.com'). There is a password field with a yellow background and a masked password '*****'. Below the password field is a small link that says 'Set A Password'. A large yellow button at the bottom is labeled 'GET STARTED'. At the very bottom, in small white text, it says 'All Rights Reserved. © EDUTOO Developers. Template by: Eric'.

How to Log-in?

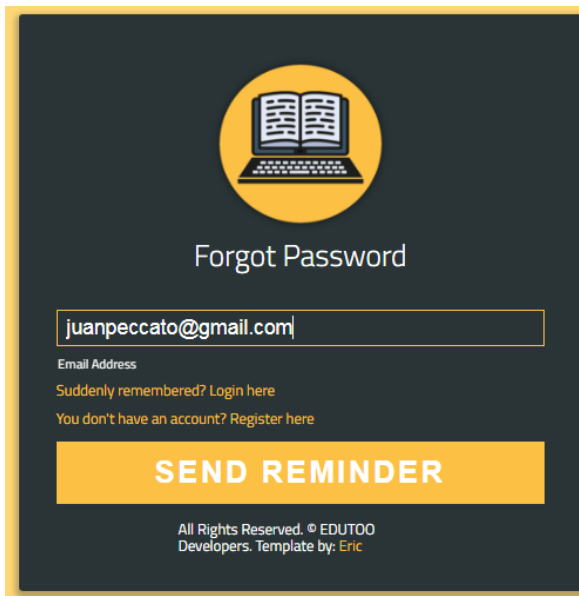
After you verified your account using email, you may start accessing the system. Click "Log In" to start.



The image shows a dark-themed login form for instructors. At the top is a circular logo with a laptop and an open book. Below the logo are two buttons: "Log In" (orange) and "Sign Up" (grey). The title "Instructor Login" is centered. There are two input fields: "Email Address" with the value "j...@yahoo.com" and "Password" with masked characters "*****". A link "Forgot Password?" is below the password field. A large orange "LOG IN" button is at the bottom. Footer text reads: "All Rights Reserved. © EDUTOO Developers. Template by: Eric".

How to reset my password in case I forgot?

Supply your e-mail address and then click “Send Reminder” button; after this an e-mail is sent to you. Click the link given in the e-mail to start creating a new password.

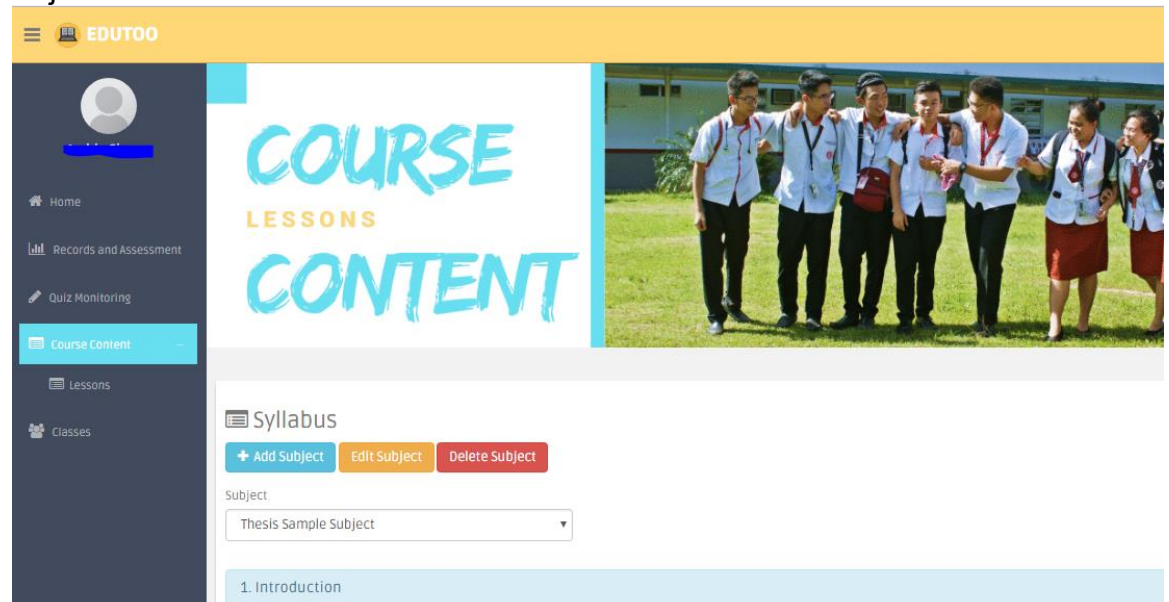


The image shows a dark-themed "Forgot Password" form. It features the same circular logo at the top. The title "Forgot Password" is centered. There is one input field for "Email Address" containing "juanpeccato@gmail.com". Below the field are two links: "Suddenly remembered? Login here" and "You don't have an account? Register here". A large orange "SEND REMINDER" button is at the bottom. Footer text reads: "All Rights Reserved. © EDUTOO Developers. Template by: Eric".

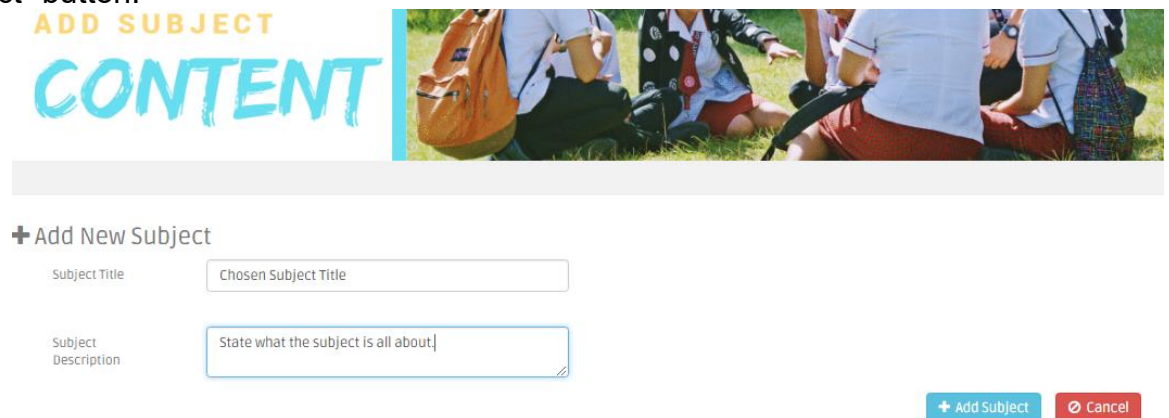
Subject Management

How to create a new subject?

Click “Course Content” then “Lessons” in the side navigation. Then, click “Add Subject” button.

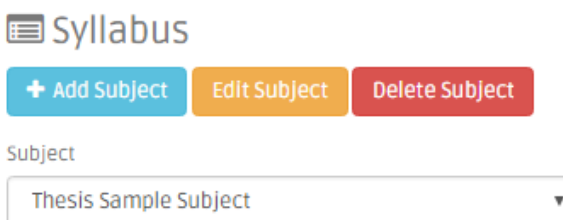


In adding a new subject, type your chosen subject title and in the subject description, state what the subject is all about. When you are done, click “Add Subject” button.



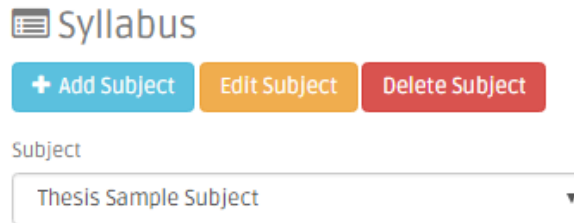
How to delete a subject?

Click “Delete Subject” button to remove the selected subject.



How to edit a subject?

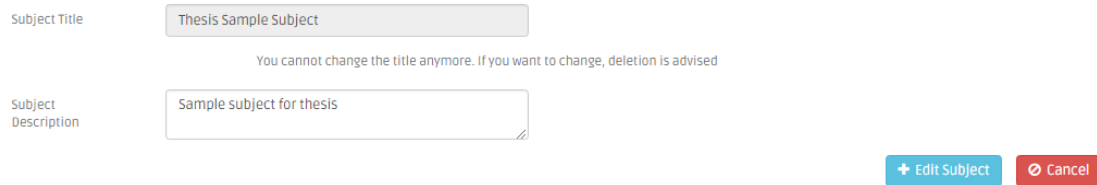
Click “Edit Subject” button to go to the edit subject panel.



The interface shows a header with a menu icon and the word "Syllabus". Below it are three buttons: "+ Add Subject" (blue), "Edit Subject" (orange), and "Delete Subject" (red). Underneath is a "Subject" label and a dropdown menu currently showing "Thesis Sample Subject".

Change the subject description. Click “Edit Subject” button when done.

+ Edit Subject

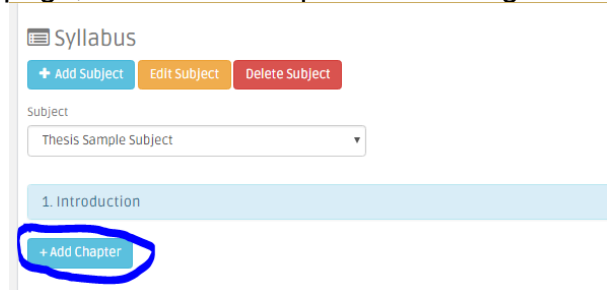


The "Edit Subject" panel has two main sections. The first is "Subject Title" with a text box containing "Thesis Sample Subject" and a message below it: "You cannot change the title anymore. If you want to change, deletion is advised". The second is "Subject Description" with a text box containing "Sample subject for thesis". At the bottom right are two buttons: "+ Edit Subject" (blue) and "Cancel" (red).

Note: You cannot change the subject title. Deletion is advised.

How to create a chapter?

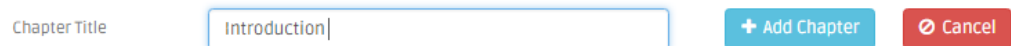
In the same page, Click “Add Chapter” button to go to the add chapter panel.



This view shows the "Syllabus" page with the "Thesis Sample Subject" selected. Below the subject dropdown, a new chapter titled "1. Introduction" has been added and is highlighted in light blue. A blue circle highlights the "+ Add Chapter" button located below the chapter list.

Fill in the chapter title and click “Add Chapter” when done.

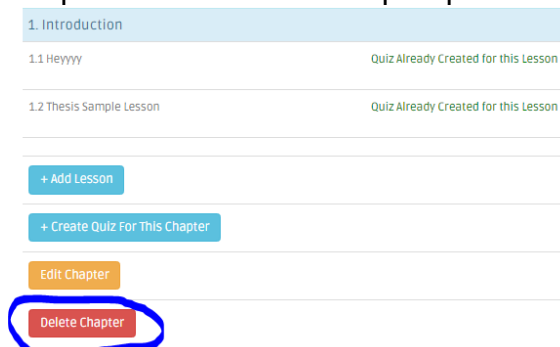
+ Add Chapter



The "Add Chapter" panel features a "Chapter Title" label and a text box containing "Introduction". To the right of the text box are two buttons: "+ Add Chapter" (blue) and "Cancel" (red).

How to delete a chapter?

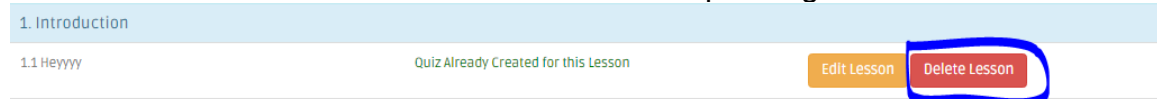
Click “Delete Chapter” button on the chapter panel.



The chapter panel displays a list of chapters. The first chapter is "1. Introduction", which is highlighted in light blue. Below it are two sub-chapters: "1.1 Heyyyy" and "1.2 Thesis Sample Lesson", both with a note "Quiz Already Created for this Lesson". At the bottom of the panel are four buttons: "+ Add Lesson" (blue), "+ Create Quiz For This Chapter" (blue), "Edit Chapter" (orange), and "Delete Chapter" (red). The "Delete Chapter" button is circled in blue.

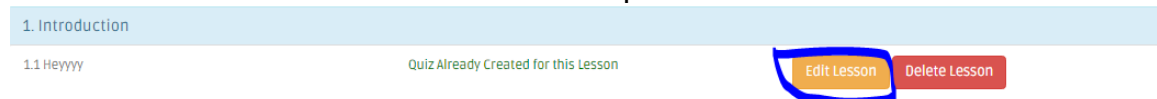
How to delete a lesson?

Click “Delete Lesson” button for the corresponding lesson.



How to edit a lesson?

Click “Edit Lesson” button for the respective lesson to start.

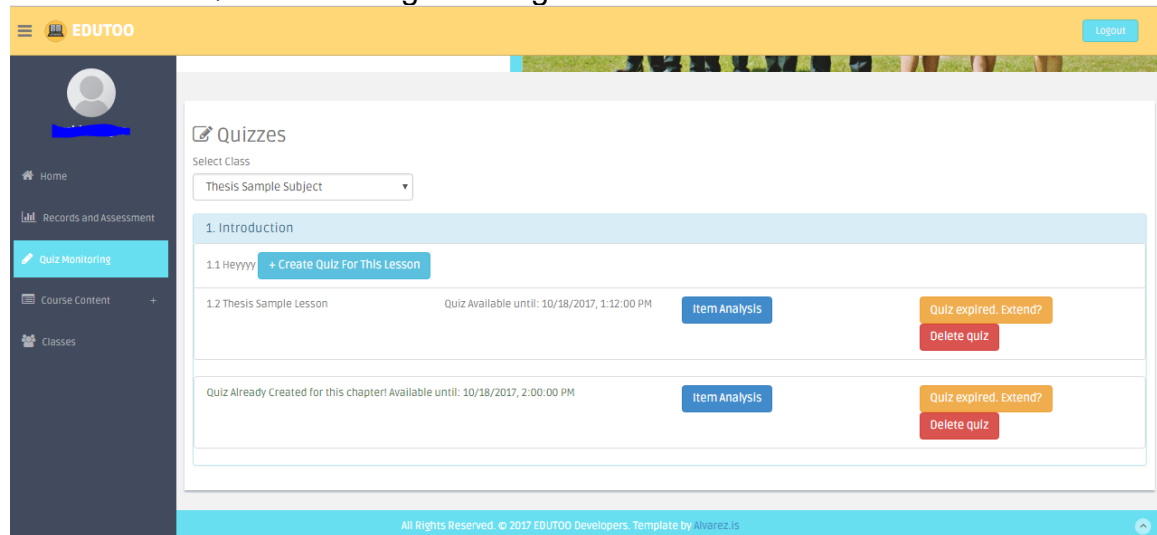


Supply the changes you decided. Whenever you are done, click “Save Changes”.

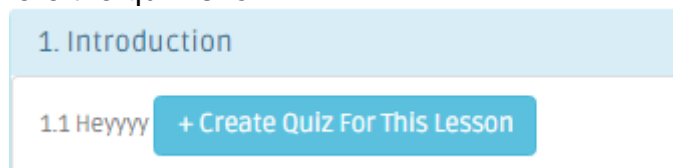
A screenshot of the 'Add Lesson' form. It has a title '+ Add Lesson'. Below it, there are three main sections: 'Lesson Title' with a text input field containing 'Heyyyy'; 'Lesson File/Material' with a 'Choose File' button and the text 'No file chosen'; and 'Sidenotes' with a rich text editor. The rich text editor has a toolbar with various icons (bold, italic, underline, link, unlink, list, etc.) and a text area below it. At the bottom right of the form, there are two buttons: 'Save Changes' (blue) and 'Cancel' (red).

How to create a quiz for a lesson/chapter?

Click “Quiz Monitoring” in navigation sidebar.



Click “Create Quiz for this Lesson” or “Create Quiz for This Chapter” depending on where the quiz is for.



Set the time limit in seconds, the number of items (must be equal to the number of questions), the expiration date and time, and whether the quiz is in random order. You may select a default question type so everytime you add a question that is the selected question type.

 Create Quiz

Quiz Properties

Quiz Title Heyyyy	Quiz Type After-Lesson Quiz	Time Limit(Seconds) 1	Number of Items 3	Expiration Date mm/dd/yyyy --	<input type="checkbox"/> Randomize
Default Question Type Multiple Choice					

Question 1

Question 2

Question 3

+ Add Question

+ Add Questions from File

Done

Cancel

Click “Add Question” to add a new question. Select the type of question and the right answer.

Question 1

Delete Question

Question

Some question?

Type

Multiple Choice

Right Answer

Yowwwwww

+Add Choice

Heyyyy

This is the right answer

Delete this choice

Yowwwwww

This is the right answer

Delete this choice

Question 2

Delete Question

Question

Apple starts with A?

Type

True/False

Answer

True

You may also choose “Add Questions from File”. Copy the specified format on the site and alter it in a .txt file as long as it matches the quiz properties. This is for offline processing.

Upload Questions File (.txt)

Copy paste Format

```
[
  {
    "q_answer": "True",
    "q_question": "a",
    "q_type": "True/False"
  },
  {
    "q_answer": "False",
    "q_question": "ab",
    "q_type": "True/False"
  },
  {
    "q_answer": "False",
    "q_question": "abc",
    "q_type": "True/False"
  },
  {
    "q_answer": "b",
    "q_question": "abcd",
    "q_a": "a",
    "q_b": "b",
    "q_c": "c",
    "q_d": "d",
    "q_type": "Multiple Choice"
  }
]
```

Choose File

quiz.txt

Add

Click “Done” after completing the quiz. Do not forget to start the quiz in Quiz Monitoring page with “Start Quiz” on the respective quiz. You may also end it with “End Quiz” if you want to.

```
[
  {
    "q_answer": "True",
    "q_question": "a",
    "q_type": "True/False"
  },
  {
    "q_answer": "False",
    "q_question": "ab",
    "q_type": "True/False"
  },
  {
    "q_answer": "False",
    "q_question": "abc",
    "q_type": "True/False"
  },
  {
    "q_answer": "b",
    "q_question": "abcd",
    "q_a": "a",
    "q_b": "b",
    "q_c": "c",
    "q_d": "d",
    "q_type": "Multiple Choice"
  }
]
```

+ Add Question

+ Add Questions from File

Done

Cancel

How to extend quiz expiration?

Click “Extend?” button.

1. Introduction

1.1 Heyyyy

+ Create Quiz For This Lesson

1.2 Thesis Sample Lesson

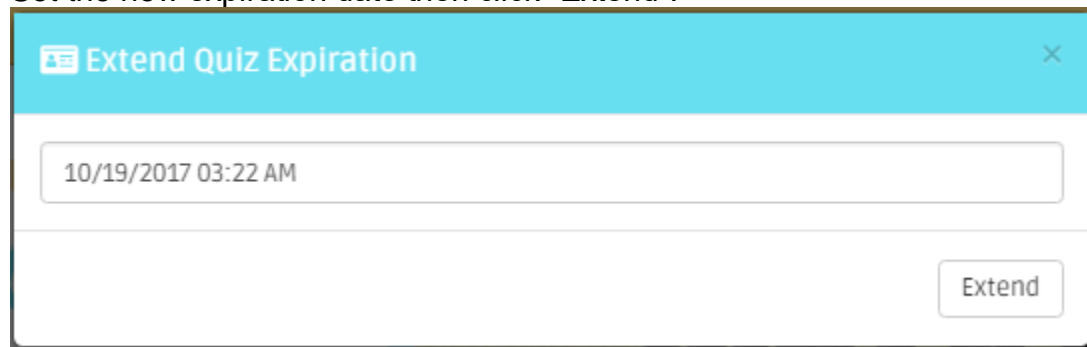
Quiz Available until: 10/18/2017, 1:12:00 PM

Item Analysis

Quiz expired. Extend?

Delete quiz

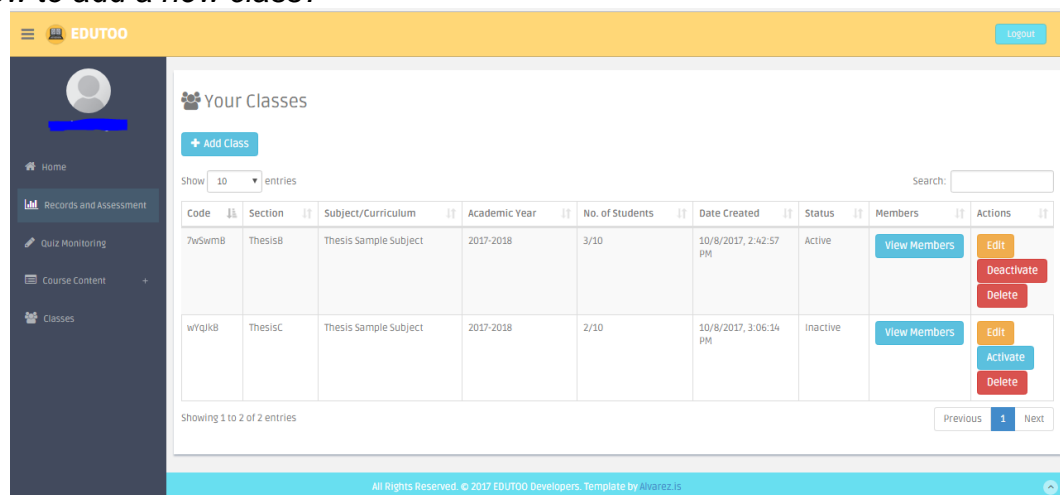
Set the new expiration date then click “Extend”.



The dialog box has a title bar "Extend Quiz Expiration" with a close button. Below the title bar is a text input field containing the date and time "10/19/2017 03:22 AM". At the bottom right of the dialog is an "Extend" button.

Class Management

How to add a new class?

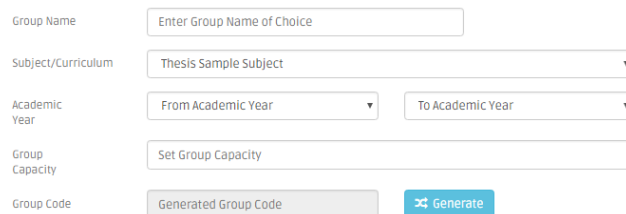


The interface shows a sidebar with navigation links: Home, Records and Assessment, Quiz Monitoring, Course Content, and Classes. The main area is titled "Your Classes" and contains a table of classes. The table has columns: Code, Section, Subject/Curriculum, Academic Year, No. of Students, Date Created, Status, Members, and Actions. There are two rows of class data. Below the table is a pagination bar showing "Showing 1 to 2 of 2 entries".

Code	Section	Subject/Curriculum	Academic Year	No. of Students	Date Created	Status	Members	Actions
7wSwmB	ThesisB	Thesis Sample Subject	2017-2018	3/10	10/8/2017, 2:42:57 PM	Active	View Members	Edit, Deactivate, Delete
wVqjkb	ThesisC	Thesis Sample Subject	2017-2018	2/10	10/8/2017, 3:06:14 PM	Inactive	View Members	Edit, Activate, Delete

Click “Classes” on the navigation sidebar. Click “Add Class” button.

+ Add Class



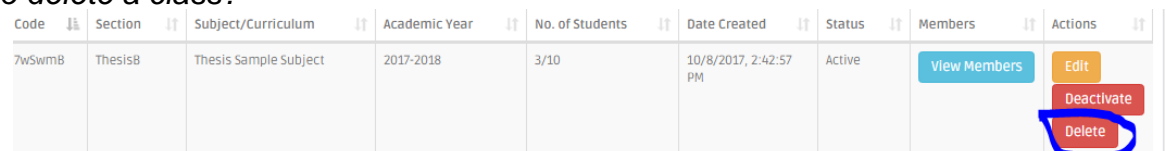
The form contains the following fields:

- Group Name: Text input field with placeholder "Enter Group Name of Choice".
- Subject/Curriculum: Dropdown menu with "Thesis Sample Subject" selected.
- Academic Year: Two dropdown menus labeled "From Academic Year" and "To Academic Year".
- Group Capacity: Text input field with placeholder "Set Group Capacity".
- Group Code: Text input field showing "Generated Group Code" and a "Generate" button.

+ Add Class **Cancel**

Supply the needed information. Click “Generate” to generate a unique group code for your class. Click “Add Class” when done.

How to delete a class?



Code	Section	Subject/Curriculum	Academic Year	No. of Students	Date Created	Status	Members	Actions
7wSwmB	ThesisB	Thesis Sample Subject	2017-2018	3/10	10/8/2017, 2:42:57 PM	Active	View Members	Edit, Deactivate, Delete

Click “Delete” button if you are decided.

How to edit a class?

Code	Section	Subject/Curriculum	Academic Year	No. of Students	Date Created	Status	Members	Actions
7wSwmB	ThesisB	Thesis Sample Subject	2017-2018	3/10	10/8/2017, 2:42:57 PM	Active	View Members	Edit Deactivate Delete

Click “Edit” button for the corresponding class.

+ Add Class

Group Name

ThesisB

Subject/Curriculum

Thesis Sample Subject

Academic Year

2017

2018

Group Capacity

10

Group Code

7wSwmB

Generate

+ Add Class

Cancel

Revise the needed information. Click “Add Class” when done.

Note: You cannot generate new group code.

How to view class members?

Code	Section	Subject/Curriculum	Academic Year	No. of Students	Date Created	Status	Members	Actions
7wSwmB	ThesisB	Thesis Sample Subject	2017-2018	3/10	10/8/2017, 2:42:57 PM	Active	View Members	Edit Deactivate Delete

Click “View Members”.

Class Members

Select Class

ThesisB(2017-2018)

Show 10 entries

Search:

	First Name	Last Name	Email
View Assessment			
View Assessment			
View Assessment	Pj	Fumera	@gmail.com

Showing 1 to 3 of 3 entries

[Previous](#) [1](#) [Next](#)

How to approve pending members of a class?

Pending Members

Show 10 entries

Search:

	First Name	Last Name	Email	Approve	Reject
No data available in table					

Showing 0 to 0 of 0 entries

[Previous](#) [Next](#)

Approve

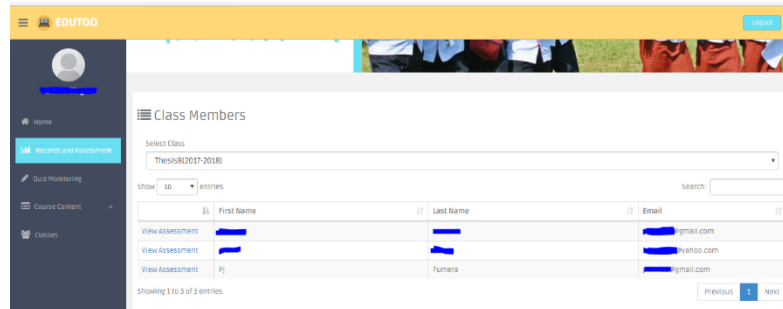
Reject

You may check all students and click “Approve” to do a batch approval. Another way is to click “Approve” for the corresponding student.

Records and Assessment

How to view the assessment of my students?

Click “Records and Assessment” on the sidebar. Click “View Assessment” for a student.



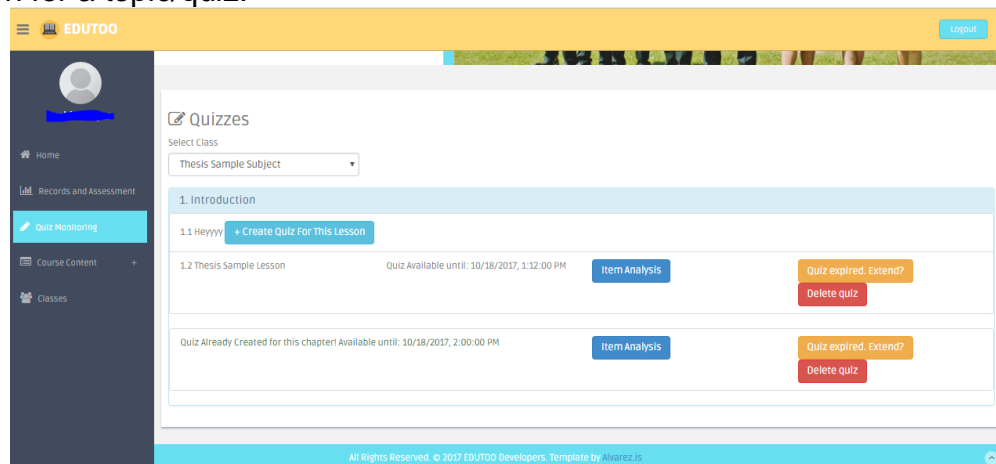
This contains the reading times, times on quiz, and scores of a student. Descriptive statistics for the data is also included.

Pj Fumera

Activity	Time to Read(minutes)	Time on Quiz(minutes)	Score	Timestamp
Heyyyy	0.10	0.13	100.0	18-10-2017-12-11-31
Time on Read Description				
Minimum Time To Read(minutes)		0.10		
Maximum Time To Read(minutes)		0.10		
Average Time To Read(minutes)		0.1		
Time on Quiz Description				
Minimum Time On Quiz(minutes)		0.13		
Maximum Time On Quiz(minutes)		0.13		
Average Time On Quiz(minutes)		0.13		
Performance Description				
Minimum Score		100		
Maximum Score		100		
Average Score		100		

How to view the item analysis for my quizzes?

Click “Quiz Monitoring” in navigation sidebar. Then click “Item Analysis” button for a topic/quiz.



The difficulty of each question is shown.

Item Analysis

Show 10 entries

Search:

Question	Difficulty
Apple starts with ?	20%
Ball starts with b?	20%
C word example	20%
Expected number of Examinees:	5
Formula(for each question)	Difficulty = Wrong/Total Number of Students in the Group

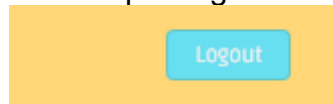
Showing 1 to 3 of 3 entries

Previous 1 Next

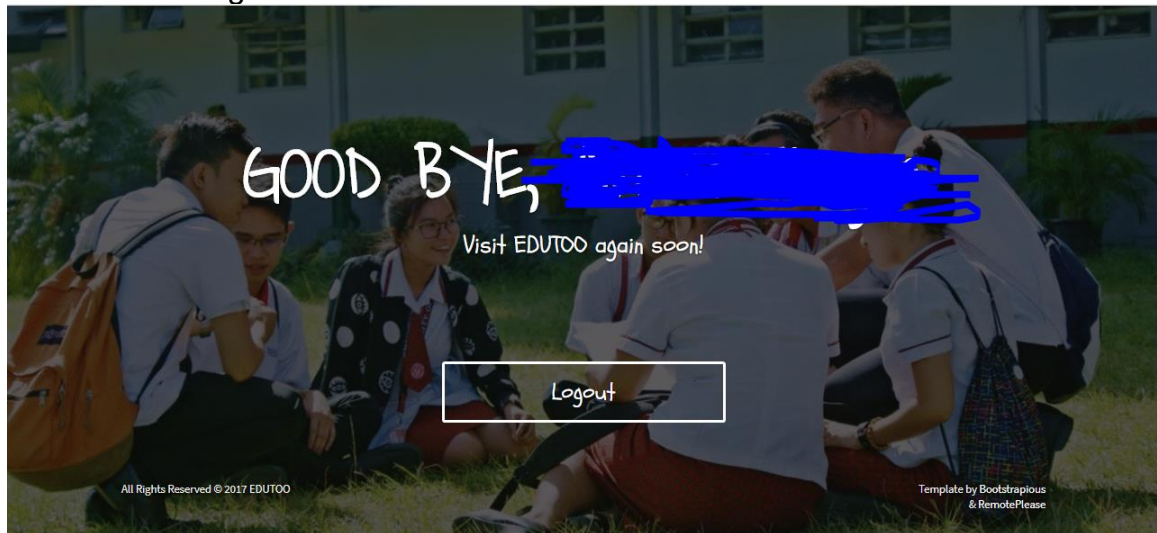
Logout

How to logout on teacher's module?

Click "Logout" button on the top navigation bar.



Click "Logout" button.



Project Cost

Program Listing (Sample Codes)

Student Module

LoginScreen.java

```
public class LoginScreen extends AppCompatActivity {

    private EditText inputEmail, inputPassword;
    private FirebaseAuth auth;
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    DatabaseReference dref;
    private ProgressBar progressBar;
    Button btnSignup, btnLogin, btnReset;
    Object email2, fname, type;
    String email;
    List<String> classes;
    FirebaseUser user;
    ImageView logo;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        auth = FirebaseAuth.getInstance();
        if (auth.getCurrentUser() != null) {
            startActivity(new Intent(LoginScreen.this, MainScreen.class));
            finish();
        }
        if (getIntent().getBooleanExtra("EXIT", false)) {
            finish();
        }
        setContentView(R.layout.activity_login_screen);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        classes = new ArrayList<>();
        logo = (ImageView) findViewById(R.id.logo);
        inputEmail = (EditText) findViewById(R.id.email);
        inputPassword = (EditText) findViewById(R.id.password);
        progressBar = (ProgressBar) findViewById(R.id.progressBar);
        btnSignup = (Button) findViewById(R.id.btn_signup);
        btnLogin = (Button) findViewById(R.id.btn_login);
        btnReset = (Button) findViewById(R.id.btn_reset_password);
        auth = FirebaseAuth.getInstance();
        btnSignup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(LoginScreen.this, RegisterScreen.class));
            }
        });

        btnReset.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(LoginScreen.this, ResetPasswordScreen.class));
            }
        });

        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                email = inputEmail.getText().toString();
                final String password = inputPassword.getText().toString();
            }
        });
    }
}
```

```

        if (TextUtils.isEmpty(email)) {
            LinearLayout layout=new LinearLayout(LoginScreen.this);
            layout.setBackgroundResource(R.color.darkorange);
            layout.setClipToOutline(true);
            TextView tv=new TextView(LoginScreen.this);
            tv.setTextColor(Color.BLACK);
            tv.setTextSize(25);
            tv.setGravity(Gravity.CENTER_HORIZONTAL);
            tv.setText("Enter Email Address");
            layout.addView(tv);
            Toast toast=new Toast(LoginScreen.this);
            toast.setView(layout);
            toast.setGravity(Gravity.CENTER, 0, 50);
            toast.show();
            return;
        }

        if (TextUtils.isEmpty(password)) {
            LinearLayout layout=new LinearLayout(LoginScreen.this);
            layout.setBackgroundResource(R.color.darkorange);
            layout.setClipToOutline(true);
            TextView tv=new TextView(LoginScreen.this);
            tv.setTextColor(Color.BLACK);
            tv.setTextSize(25);
            tv.setGravity(Gravity.CENTER_HORIZONTAL);
            tv.setText("Enter Password");
            layout.addView(tv);
            Toast toast=new Toast(LoginScreen.this);
            toast.setView(layout);
            toast.setGravity(Gravity.CENTER, 0, 50);
            toast.show();
            return;
        }
        btnLogin.setVisibility(View.GONE);
        inputEmail.setEnabled(false);
        inputPassword.setEnabled(false);
        progressBar.setVisibility(View.VISIBLE);
        //authenticate user
        auth.signInWithEmailAndPassword(email, password)
            .addOnCompleteListener(LoginScreen.this, new
OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                // If sign in fails, display a message to the user. If
sign in succeeds
                // the auth state listener will be notified and logic
to handle the
                // signed in user can be handled in the listener.
                if (!task.isSuccessful()) {
                    // there was an error
                    progressBar.setVisibility(View.GONE);
                    btnLogin.setVisibility(View.VISIBLE);
                    inputEmail.setEnabled(true);
                    inputPassword.setEnabled(true);
                    if (password.length() < 6) {

inputPassword.setError(getString(R.string.minimum_password));
                    } else {
                        LinearLayout layout=new
LinearLayout(LoginScreen.this);

```

[illegible]

```

TextView(LoginScreen.this);

tv.setGravity(Gravity.CENTER_HORIZONTAL);
tv.setText(getString(R.string.auth_failed3));

Toast(LoginScreen.this);

toast.setGravity(Gravity.CENTER, 0, 50);

progressBar.setVisibility(View.GONE);
btnLogin.setVisibility(View.VISIBLE);

snapshot.getKey();
newPost.get("firstname");
globalVariables.setname(fname.toString());
globalVariables.setEmail(email);
globalVariables.setstudentKey(studentkey);
FileCacher<String>(LoginScreen.this, "name");
FileCacher<Object>(LoginScreen.this, "email");
new FileCacher<String>(LoginScreen.this, "studkey");

name.writeCache(fname.toString());

studkey.writeCache(studentkey);

progressBar.setVisibility(View.GONE);
Intent(LoginScreen.this, MainScreen.class);

```

```

TextView tv=new

tv.setTextColor(Color.BLACK);
tv.setTextSize(25);

layout.addView(tv);
Toast toast=new

toast.setView(layout);

toast.show();
auth.signOut();

inputEmail.setEnabled(true);
inputPassword.setEnabled(true);
}else{
String studentkey =

fname =

FileCacher<String> name = new
FileCacher<Object> email = new
FileCacher<String> studkey =
try{

email.writeCache(email2);

}catch (IOException e) {
e.printStackTrace();
}

Intent intent = new

startActivity(intent);
finish();
}

```

```

}

```

```

dataSnapshot, String s) {

dataSnapshot) {

dataSnapshot, String s) {

databaseError) {

onChildChanged, onChildRemoved,
in later sections.

    });
}

    }
}

@Override
public void onBackPressed() {
    new AlertDialog.Builder(this)
        .setTitle("Really Exit?")
        .setMessage("Are you sure you want to exit?")
        .setNegativeButton(android.R.string.no, null)
        .setPositiveButton(android.R.string.yes, new
DialogInterface.OnClickListener() {

            public void onClick(DialogInterface arg0, int arg1) {
                Intent intent = new Intent(LoginScreen.this, LoginScreen.class);
                intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                intent.putExtra("EXIT", true);
                startActivity(intent);
            }
        })
        .create().show();
}}

RegisterScreen.java
public class RegisterScreen extends AppCompatActivity {
    private EditText inputEmail, inputPassword;

```



```

private Button btnSignIn, btnSignUp;
private ProgressBar progressBar;
private FirebaseAuth auth;
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference dref;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_register_screen);
    auth = FirebaseAuth.getInstance();
    dref = FirebaseDatabase.getInstance().getReference();
    btnSignIn = (Button) findViewById(R.id.sign_in_button);
    btnSignUp = (Button) findViewById(R.id.sign_up_button);
    inputEmail = (EditText) findViewById(R.id.email);
    inputPassword = (EditText) findViewById(R.id.password);
    progressBar = (ProgressBar) findViewById(R.id.progressBar);
    btnSignIn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(getBaseContext(), LoginScreen.class);
            startActivity(intent);
            finish();
        }
    });

    btnSignUp.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            final String email = inputEmail.getText().toString().trim();
            final String password = inputPassword.getText().toString().trim();

            if (TextUtils.isEmpty(email)) {
                LinearLayout layout=new LinearLayout(RegisterScreen.this);
                layout.setBackgroundResource(R.color.darkorange);
                layout.setClipToOutline(true);
                TextView tv=new TextView(RegisterScreen.this);
                tv.setTextColor(Color.BLACK);
                tv.setTextSize(25);
                tv.setGravity(Gravity.CENTER_HORIZONTAL);
                tv.setText("Enter Email Address");
                layout.addView(tv);
                Toast toast=new Toast(RegisterScreen.this);
                toast.setView(layout);
                toast.setGravity(Gravity.CENTER, 0, 50);
                toast.show();
                return;
            }

            if (TextUtils.isEmpty(password)) {
                LinearLayout layout=new LinearLayout(RegisterScreen.this);
                layout.setBackgroundResource(R.color.darkorange);
                layout.setClipToOutline(true);
                TextView tv=new TextView(RegisterScreen.this);
                tv.setTextColor(Color.BLACK);
                tv.setTextSize(25);
                tv.setGravity(Gravity.CENTER_HORIZONTAL);
                tv.setText("Enter Password");
                layout.addView(tv);
                Toast toast=new Toast(RegisterScreen.this);
                toast.setView(layout);
            }
        }
    });
}

```

```

        toast.setGravity(Gravity.CENTER, 0, 50);
        toast.show();
        return;
    }
    if (password.length() < 6) {
        LinearLayout layout=new LinearLayout(RegisterScreen.this);
        layout.setBackgroundResource(R.color.darkorange);
        layout.setClipToOutline(true);
        TextView tv=new TextView(RegisterScreen.this);
        tv.setTextColor(Color.BLACK);
        tv.setTextSize(25);
        tv.setGravity(Gravity.CENTER_HORIZONTAL);
        tv.setText("Password too short. Enter a minimum of 6 characters");
        layout.addView(tv);
        Toast toast=new Toast(RegisterScreen.this);
        toast.setView(layout);
        toast.setGravity(Gravity.CENTER, 0, 50);
        toast.show();
        return;
    }
    if(isEmailValid(email)){
        btnSignUp.setVisibility(View.INVISIBLE);
        progressBar.setVisibility(View.VISIBLE);
        inputEmail.setEnabled(false);
        inputPassword.setEnabled(false);

        auth.createUserWithEmailAndPassword(email,password).addOnCompleteListener(RegisterScreen.this,new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (!task.isSuccessful())
                {
                    try
                    {
                        throw task.getException();
                    }
                    catch (FirebaseAuthUserCollisionException existEmail)
                    {
                        LinearLayout layout=new
LinearLayout(RegisterScreen.this);
                        layout.setBackgroundResource(R.color.darkorange);
                        layout.setClipToOutline(true);
                        TextView tv=new TextView(RegisterScreen.this);
                        tv.setTextColor(Color.BLACK);
                        tv.setTextSize(25);
                        tv.setGravity(Gravity.CENTER_HORIZONTAL);
                        tv.setText("Email Already in Used.");
                        layout.addView(tv);
                        Toast toast=new Toast(RegisterScreen.this);
                        toast.setView(layout);
                        toast.setGravity(Gravity.CENTER, 0, 50);
                        toast.show();
                        btnSignUp.setVisibility(View.VISIBLE);
                        progressBar.setVisibility(View.INVISIBLE);
                        inputEmail.setEnabled(true);
                        inputPassword.setEnabled(true);
                    }
                    catch (Exception e)
                    {
                        Log.d("CHECK", "onComplete: " + e.getMessage());
                    }
                }
            }
        });
    }
}

```

```

        }else{
            auth.getCurrentUser().delete();
            Intent intent = new Intent(getApplicationContext(),
RegisterInfoScreen.class);

            Bundle extras = new Bundle();
            extras.putString("EXTRA_EMAIL",email);
            extras.putString("EXTRA_PASSWORD",password);
            intent.putExtras(extras);
            startActivity(intent);
            finish();
        }
    }
});

}else{
    LinearLayout layout=new LinearLayout(RegisterScreen.this);
    layout.setBackgroundResource(R.color.darkorange);
    layout.setClipToOutline(true);
    TextView tv=new TextView(RegisterScreen.this);
    tv.setTextColor(Color.BLACK);
    tv.setTextSize(25);
    tv.setGravity(Gravity.CENTER_HORIZONTAL);
    tv.setText("Invalid Email");
    layout.addView(tv);
    Toast toast=new Toast(RegisterScreen.this);
    toast.setView(layout);
    toast.setGravity(Gravity.CENTER, 0, 50);
    toast.show();
    btnSignUp.setVisibility(View.VISIBLE);
    progressBar.setVisibility(View.INVISIBLE);
    inputEmail.setEnabled(true);
    inputPassword.setEnabled(true);
}

});
}

public static boolean isValidEmail(String email) {
    String expression = "^[\\w\\.\\-]+@([\\w\\.\\-]+\\.)+[A-Z]{2,4}$";
    Pattern pattern = Pattern.compile(expression, Pattern.CASE_INSENSITIVE);
    Matcher matcher = pattern.matcher(email);
    return matcher.matches();
}

@Override
protected void onResume() {
    super.onResume();
    progressBar.setVisibility(View.GONE);
}

@Override
public void onBackPressed() {
    Intent intent = new Intent(RegisterScreen.this,LoginScreen.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    intent.putExtra("EXIT", true);
    startActivity(intent);
}
}

```

RegisterInfoScreen.java

```

public class RegisterInfoScreen extends AppCompatActivity {

    private static final String TAG = RegisterInfoScreen.class.getSimpleName();

```

```

private TextView txtDetails;
private EditText inputfirstName, inputlastName;
private Button btnSave;
RadioGroup rdgroup;
RadioButton rdbutton0, rdbutton1, rdbutton2;
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference dref;
FirebaseUser user;
private FirebaseAuth auth;
String fname, lname, email1, gender, utype, id, status;
String email, password;
private ProgressBar progressBar;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    auth = FirebaseAuth.getInstance();
    setContentView(R.layout.activity_register_info_screen);
    txtDetails = (TextView) findViewById(R.id.txt_user);
    inputfirstName = (EditText) findViewById(R.id.firstname);
    inputlastName = (EditText) findViewById(R.id.lastname);
    btnSave = (Button) findViewById(R.id.btn_save);
    auth = FirebaseAuth.getInstance();
    rdgroup = (RadioGroup) findViewById(R.id.radioGroup1);
    rdbutton0 = (RadioButton) findViewById(R.id.radio0);
    rdbutton1 = (RadioButton) findViewById(R.id.radio1);
    progressBar = (ProgressBar) findViewById(R.id.progressBar);
    dref = FirebaseDatabase.getInstance().getReference();
    Bundle extras = getIntent().getExtras();
    email = extras.getString("EXTRA_EMAIL");
    password = extras.getString("EXTRA_PASSWORD");
    btnSave.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            inputfirstName.setEnabled(false);
            inputlastName.setEnabled(false);
            btnSave.setVisibility(View.INVISIBLE);
            rdgroup.setEnabled(false);
            fname = inputfirstName.getText().toString();
            lname = inputlastName.getText().toString();
            utype = "Student";
            status = "firsttimelogin";
            email1 = email;
            int selectedId = rdgroup.getCheckedRadioButtonId();
            rdbutton2 = (RadioButton) findViewById(selectedId);
            gender = rdbutton2.getText().toString();
            if(fname.isEmpty() || lname.isEmpty() || gender.isEmpty()){
                LinearLayout layout=new LinearLayout(RegisterInfoScreen.this);
                layout.setBackgroundResource(R.color.darkorange);
                layout.setClipToOutline(true);
                TextView tv=new TextView(RegisterInfoScreen.this);
                tv.setTextColor(Color.BLACK);
                tv.setTextSize(25);
                tv.setGravity(Gravity.CENTER_HORIZONTAL);
                tv.setText("Empty Field/s");
                layout.addView(tv);
                Toast toast=new Toast(RegisterInfoScreen.this);
                toast.setView(layout);
                toast.setGravity(Gravity.CENTER, 0, 50);
                toast.show();
                inputfirstName.setEnabled(true);
                inputlastName.setEnabled(true);
            }
        }
    });
}

```

```

        btnSave.setVisibility(View.VISIBLE);
        rdgroup.setEnabled(true);
    }else{
        progressBar.setVisibility(View.VISIBLE);
        auth.createUserWithEmailAndPassword(email, password)
            .addOnCompleteListener(RegisterInfoScreen.this, new
OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task)
{
                progressBar.setVisibility(View.GONE);
                if (!task.isSuccessful()) {
                    inputfirstName.setEnabled(true);
                    inputlastName.setEnabled(true);
                    btnSave.setVisibility(View.VISIBLE);
                    rdgroup.setEnabled(true);
                    LinearLayout layout=new
LinearLayout(RegisterInfoScreen.this);

                    layout.setBackgroundResource(R.color.darkorange);
                    layout.setClipToOutline(true);
                    TextView tv=new
TextView(RegisterInfoScreen.this);

                    tv.setTextColor(Color.BLACK);
                    tv.setTextSize(25);
                    tv.setGravity(Gravity.CENTER_HORIZONTAL);
                    tv.setText("Registration Failed. Email is
already used.");

                    layout.addView(tv);
                    Toast toast=new Toast(RegisterInfoScreen.this);
                    toast.setView(layout);
                    toast.setGravity(Gravity.CENTER, 0, 50);
                    toast.show();
                    Intent intent = new
Intent(RegisterInfoScreen.this, RegisterScreen.class);
                    startActivity(intent);
                    finish();
                } else if(task.isSuccessful()) {
                    id = dref.push().getKey();
                    user = auth.getCurrentUser();
                    createUser(fname, lname, utype, email1, id,
gender);

                    finish();
                }
            }
        });
    }
}

});
});
});

}

private void createUser(String fname, String lname, String utype, String email1,
String id, String gender) {
    User user = new User(fname, lname, utype, email1, gender);
    dref.child("members").child(id).setValue(user);
    addUserChangeListener(id, fname, email1);
}

private void addUserChangeListener(String id, String fname, String email1) {
    final String studentkey = id;
    final String fname1 = fname;
    final String email2 = email1;

```

```

        dref.child("members").child(id).addValueEventListener(new ValueEventListener()
    {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            globalVariables.setname(fname1);
            globalVariables.setEmail(email2);
            globalVariables.setstudentKey(studentkey);
            user.sendEmailVerification().addOnCompleteListener(new
    OnCompleteListener<Void>() {
                @Override
                public void onComplete(@NonNull Task<Void> task) {
                    LinearLayout layout=new LinearLayout(RegisterInfoScreen.this);
                    layout.setBackgroundResource(R.color.darkorange);
                    layout.setClipToOutline(true);
                    TextView tv=new TextView(RegisterInfoScreen.this);
                    tv.setTextColor(Color.BLACK);
                    tv.setTextSize(25);
                    tv.setGravity(Gravity.CENTER_HORIZONTAL);
                    tv.setText(getString(R.string.verify_first));
                    layout.addView(tv);
                    Toast toast=new Toast(RegisterInfoScreen.this);
                    toast.setView(layout);
                    toast.setGravity(Gravity.CENTER, 0, 50);
                    toast.show();
                }
            });
            boolean isVerified = user.isEmailVerified();
            if(!isVerified){
                auth.signOut();
                Intent intent = new Intent(RegisterInfoScreen.this,
    LoginScreen.class);
                startActivity(intent);
                finish();
            }else{
                Intent intent = new Intent(RegisterInfoScreen.this,
    MainScreen.class);
                startActivity(intent);
                finish();
            }

            @Override
            public void onCancelled(DatabaseError error) {
                // Failed to read value
                Log.e(TAG, "Failed", error.toException());
            }
        });
    }
    @Override
    public void onBackPressed() {
        Intent intent = new Intent(RegisterInfoScreen.this,RegisterScreen.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        intent.putExtra("EXIT", true);
        startActivity(intent);
    }
}

```

MainScreen.java

```

public class MainScreen extends AppCompatActivity {
    private NavigationView navigationView;

```

```

private DrawerLayout drawer;
private View navHeader;
private ImageView imgNavHeaderBg, imgProfile;
private TextView txtName, txtWebsite;
private Toolbar toolbar;
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference dref, dref2;
private static final String urlNavHeaderBg =
"https://firebasestorage.googleapis.com/v0/b/ezbio-
98be1.appspot.com/o/forapp.png?alt=media&token=16045b5f-89f8-4a03-af20-44eac3829ea8";
private static final String urlProfileImg =
"https://firebasestorage.googleapis.com/v0/b/ezbio-
98be1.appspot.com/o/user_default.png?alt=media&token=5057ffd1-dd08-4f84-a13e-
499fc71a8a05";
public static int navItemIndex = 0;
private static final String TAG_DASHBOARD = "dashboard";
private static final String TAG_CLASSES = "classes";
public static String CURRENT_TAG = TAG_DASHBOARD;
String curr_email, id, grpCode;
String meronna;
List<String> groupcodes;
int cnt = 0;
private String[] activityTitles;
private FirebaseAuth auth;
private boolean shouldLoadCatFragOnBackPressed = true;
private Handler mHandler;
public static final String Name = "nameKey";
public static final String Email = "emailKey";
FileCacher<String> name = new FileCacher<String>(MainScreen.this, "name");
FileCacher<Object> email = new FileCacher<Object>(MainScreen.this, "email");
FileCacher<String> studkey = new FileCacher<String>(MainScreen.this, "studkey");
FileCacheManager manager = new FileCacheManager(MainScreen.this);
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    auth = FirebaseAuth.getInstance();
    if (auth.getCurrentUser() == null) {
        startActivity(new Intent(MainScreen.this, LoginScreen.class));
        finish();
    }
    isInternetOn();
    setContentView(R.layout.activity_main_screen);
    curr_email = auth.getCurrentUser().getEmail();
    try {
        globalVariables.setstudentKey(studkey.readCache());
    } catch (IOException e) {
        e.printStackTrace();
    }
    toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    mHandler = new Handler();
    drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    navigationView = (NavigationView) findViewById(R.id.nav_view);
    navHeader = navigationView.getHeaderView(0);
    txtName = (TextView) navHeader.findViewById(R.id.name);
    txtWebsite = (TextView) navHeader.findViewById(R.id.website);
    imgNavHeaderBg = (ImageView) navHeader.findViewById(R.id.img_header_bg);
    imgProfile = (ImageView) navHeader.findViewById(R.id.img_profile);
    activityTitles =
getResources().getStringArray(R.array.nav_item_activity_titles);
    loadNavHeader();
}

```

```

        setUpNavigationView();
        if (savedInstanceState == null) {
            navItemIndex = 0;
            CURRENT_TAG = TAG_DASHBOARD;
            loadHomeFragment();
        }
    }
    private void loadNavHeader() {
        // name, website
        if(name.hasCache()){
            try{
                txtName.setText(name.readCache());
            }catch(IOException e){
                e.printStackTrace();
            }
        }

        if(email.hasCache()){
            try{
                Object email2 = email.readCache();
                txtWebsite.setText(email2.toString());
            }catch(IOException e){
                e.printStackTrace();
            }
        }

        // loading header background image
        Glide.with(this).load(urlNavHeaderBg)
            .crossFade()
            .diskCacheStrategy(DiskCacheStrategy.ALL)
            .into(imgNavHeaderBg);
        // Loading profile image
        Glide.with(this).load(urlProfileImg)
            .crossFade()
            .thumbnail(0.5f)
            .bitmapTransform(new CircleTransform(this))
            .diskCacheStrategy(DiskCacheStrategy.ALL)
            .into(imgProfile);
        // showing dot next to notifications label
        navigationView.getMenu().getItem(0).setActionView(R.layout.menu_dot);
    }
    private void loadHomeFragment() {
        // selecting appropriate nav menu item
        selectNavMenu();
        // set toolbar title
        setToolbarTitle();
        // if user select the current navigation menu again, don't do anything
        // just close the navigation drawer
        Runnable mPendingRunnable = new Runnable() {
            @Override
            public void run() {
                // update the main content by replacing fragments
                Fragment fragment = getHomeFragment();
                FragmentTransaction fragmentTransaction =
getSupportFragmentManager().beginTransaction();
                fragmentTransaction.setCustomAnimations(android.R.anim.fade_in,
                    android.R.anim.fade_out);
                fragmentTransaction.replace(R.id.frame, fragment, CURRENT_TAG);
                fragmentTransaction.commitAllowingStateLoss();
            }
        };
    }

```



```

        // If mPendingRunnable is not null, then add to the message queue
        if (mPendingRunnable != null) {
            mHandler.post(mPendingRunnable);
        }
        // show or hide the fab button
        //Closing drawer on item click
        drawer.closeDrawers();
        // refresh toolbar menu
        invalidateOptionsMenu();
    }
    private Fragment getHomeFragment() {
        switch (navItemIndex) {
            case 0:
                // home
                CatalogFragment catalogFragment = new CatalogFragment();
                return catalogFragment;
            case 1:
                // photos
                PhotosFragment photosFragment = new PhotosFragment();
                return photosFragment;
            case 2:
                // notifications fragment
                NotificationsFragment notificationsFragment = new
NotificationsFragment();
                return notificationsFragment;
            case 3:
                // movies fragment
                ProfileFragment profileFragment = new ProfileFragment();
                return profileFragment;
            default:
                return new CatalogFragment();
        }
    }

    private void setToolbarTitle() {
        getSupportActionBar().setTitle(activityTitles[navItemIndex]);
    }
    private void selectNavMenu() {
        navigationView.getMenu().getItem(navItemIndex).setChecked(true);
    }
    private void setUpNavigationView() {
        //Setting Navigation View Item Selected Listener to handle the item click of
the navigation menu
        navigationView.setNavigationItemSelectedListener(new
NavigationView.OnNavigationItemSelectedListener() {

            // This method will trigger on item Click of navigation menu
            @Override
            public boolean onNavigationItemSelected(MenuItem menuItem) {
                //Check to see which item was being clicked and perform appropriate
action

                switch (menuItem.getItemId()) {
                    //Replacing the main content with ContentFragment Which is our
Inbox View;

                    case R.id.nav_catalog:
                        navItemIndex = 0;
                        CURRENT_TAG = TAG_DASHBOARD;
                        break;
                    case R.id.nav_classes:
                        navItemIndex = 1;
                        CURRENT_TAG = TAG_CLASSES;

```

```

        break;
    case R.id.nav_joininggroup:
        dref = FirebaseDatabase.getInstance().getReference();
        groupcodes = new ArrayList<String>();
        AlertDialog.Builder builder = new
AlertDialog.Builder(MainScreen.this);
        builder.setTitle("Enter Group Code");
        final EditText input = new EditText(MainScreen.this);

        // Specify the type of input expected; this, for example, sets
the input as a password, and will mask the text
        input.setInputType(InputType.TYPE_CLASS_TEXT);
        builder.setView(input);
        builder.setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                grpCode = input.getText().toString();
                dref2 = database.getReference().child("classes");
                dref2.addListenerForSingleValueEvent(new
ValueEventListener() {
                    @Override
                    public void onDataChange(DataSnapshot dataSnapshot) {
                        {
                            if(dataSnapshot.exists()){
                                for(DataSnapshot d :
dataSnapshot.getChildren()){
                                    groupcodes.add(d.getKey());
                                    if (Objects.equals(d.getKey(),
grpCode)) {
                                        cnt++;
                                    }
                                }
                            }
                            if(cnt == 0){
                                AlertDialog.Builder builder2 = new
                                builder2.setMessage("Group Code Not
                                Found")
                                .setCancelable(false)
                                .setNegativeButton("Ok", new
                                DialogInterface.OnClickListener() {
                                    public void
                                    onClick(DialogInterface dialog, int id) {
                                        dialog.cancel();
                                    }
                                })
                                //Set your icon here
                                .setTitle("Information")
                                .setIcon(R.mipmap.ic_logo);
                                AlertDialog alert = builder2.create();
                                alert.show();//showing the dialog
                            }else{
                                for(int k = 0 ; k<= groupcodes.size()-1
; k++){
                                    if
                                    (Objects.equals(groupcodes.get(k), grpCode)) {
                                        id =
                                        globalVariables.getstudentKey();
                                        joininggroup(id, grpCode);
                                    }
                                }
                            }
                        }
                    }
                }
            }
        });
    }
}

```

```

        }
    }
}

@Override
public void onCancelled(DatabaseError
databaseError) {

    }

});
});
builder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        dialog.cancel();
    }
});
builder.show();
drawer.closeDrawers();
break;
case R.id.nav_about_us:
    // launch new intent instead of loading fragment
    startActivity(new Intent(MainScreen.this,
AboutUsActivity.class));
    drawer.closeDrawers();
    break;
case R.id.nav_privacy_policy:
    // launch new intent instead of loading fragment
    startActivity(new Intent(MainScreen.this,
PrivacyPolicyActivity.class));
    drawer.closeDrawers();
    break;
case R.id.nav_log_out:
    // launch new intent instead of loading fragment
    auth.signOut();
    Intent intent = new Intent(MainScreen.this, LoginScreen.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    drawer.closeDrawers();
    if(manager.hasCacheFiles()){
        manager.deleteAllCaches();
    }
    startActivity(intent);
    finish();
    return true;
default:
    navItemIndex = 0;
}

//Checking if the item is in checked state or not, if not make it in
checked state
if (menuItem.isChecked()) {
    menuItem.setChecked(false);
} else {
    menuItem.setChecked(true);
}
menuItem.setChecked(true);

loadHomeFragment();

```

```

        return true;
    }
});

ActionBarDrawerToggle actionBarDrawerToggle = new ActionBarDrawerToggle(this,
drawer, toolbar, R.string.openDrawer, R.string.closeDrawer) {
    @Override
    public void onDrawerClosed(View drawerView) {
        // Code here will be triggered once the drawer closes as we dont want
        anything to happen so we leave this blank
        super.onDrawerClosed(drawerView);
    }
    @Override
    public void onDrawerOpened(View drawerView) {
        // Code here will be triggered once the drawer open as we dont want
        anything to happen so we leave this blank
        super.onDrawerOpened(drawerView);
    }
};
//Setting the actionBarToggle to drawer layout
drawer.setDrawerListener(actionBarDrawerToggle);
//calling sync state is necessary or else your hamburger icon wont show up
actionBarDrawerToggle.syncState();
}
@Override
public void onBackPressed() {
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawers();
        return;
    }
    // This code loads home fragment when back key is pressed
    // when user is in other fragment than home
    if (shouldLoadCatFragOnBackPressed) {
        // checking if user is on other navigation menu
        // rather than home
        if (navItemIndex != 0) {
            navItemIndex = 0;
            CURRENT_TAG = TAG_DASHBOARD;
            loadHomeFragment();
            return;
        }
    }
    super.onBackPressed();
}
@Override
protected void onStop(){
    super.onStop();
}
@Override
protected void onResume(){
    super.onResume();
    auth = FirebaseAuth.getInstance();
    if (auth.getCurrentUser() == null) {
        startActivity(new Intent(MainScreen.this, LoginScreen.class));
        finish();
    }
}
@Override
protected void onDestroy(){
    super.onDestroy();
}

```

```

    }
    public final boolean isInternetOn() {
        // get Connectivity Manager object to check connection
        ConnectivityManager connec =

        (ConnectivityManager) getSystemService(getBaseContext().CONNECTIVITY_SERVICE);
        // Check for network connections
        if ( connec.getNetworkInfo(0).getState() ==
        android.net.NetworkInfo.State.CONNECTED ||
            connec.getNetworkInfo(0).getState() ==
        android.net.NetworkInfo.State.CONNECTING ||
            connec.getNetworkInfo(1).getState() ==
        android.net.NetworkInfo.State.CONNECTING ||
            connec.getNetworkInfo(1).getState() ==
        android.net.NetworkInfo.State.CONNECTED ) {
            // if connected with internet
            return true;
        } else if (
            connec.getNetworkInfo(0).getState() ==
        android.net.NetworkInfo.State.DISCONNECTED ||
            connec.getNetworkInfo(1).getState() ==
        android.net.NetworkInfo.State.DISCONNECTED ) {
            Toast.makeText(this, " No internet connection", Toast.LENGTH_LONG).show();
            Intent intent = new Intent(MainScreen.this,OfflineScreen.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);
            finish();
            return false;
        }
        return false;
    }
    private void joingroup(final String id, final String grpCode) {
        dref2 = database.getReference().child("classes").child(grpCode);
        dref2.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                Map<String, Object> newPost = (Map<String, Object>)
                dataSnapshot.getValue();
                final String grpcapacity = newPost.get("grp_capacity").toString();
                dref2 =
                database.getReference().child("classes").child(grpCode).child("member");
                dref2.addListenerForSingleValueEvent(new ValueEventListener() {
                    @Override
                    public void onDataChange(DataSnapshot dataSnapshot) {
                        Log.e("Check", ""+dataSnapshot.getChildrenCount());
                        if(Integer.parseInt(grpcapacity) >
                        dataSnapshot.getChildrenCount()){
                            SimpleDateFormat simpleDateFormat = new
                            SimpleDateFormat("dd-MM-yyyy-hh-mm-ss");
                            String date_requested = simpleDateFormat.format(new
                            Date());
                            classes group = new classes(id, grpCode, date_requested);

                            dref.child("classes").child(grpCode).child("pending_members").child(id).setValue(group)
                            ;
                            AlertDialog.Builder builder = new
                            AlertDialog.Builder(MainScreen.this);
                            builder.setTitle("Wait For Approval");
                            builder.setMessage("Please wait for your teacher's
                            approval");

```

```

        builder.setPositiveButton("Ok", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which)
{
                dialog.cancel();
                recreate();
            }
        });
        builder.show();
    }else{
        AlertDialog.Builder builder = new
AlertDialog.Builder(MainScreen.this);
        builder.setTitle("Information");
        builder.setMessage("Group is already full");
        builder.setPositiveButton("Ok", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which)
{
                dialog.cancel();
                recreate();
            }
        });
        builder.show();
    }
}

@Override
public void onCancelled(DatabaseError databaseError) {
}

});
}

@Override
public void onCancelled(DatabaseError databaseError) {
}

});
}
}
}

```

CatalogFragment.java

```

public class CatalogFragment extends Fragment implements View.OnClickListener {
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";
    View view;
    private Button btnjoininggroup;
    // TODO: Rename and change types of parameters
    private String mParam1;
    private String mParam2;
    String grpCode;
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    DatabaseReference dref, dref2;
    String curr_email, id;
    int cnt = 0;
    int l;
    List<String> classes;
    List<String> groupcodes;
    List<String> activities;
}

```

```

private FirebaseAuth auth;
TableLayout assessment;
TextView norecord;
private OnFragmentInteractionListener mListener;
public CatalogFragment() {
    // Required empty public constructor
}

/**
 * Use this factory method to create a new instance of
 * this fragment using the provided parameters.
 *
 * @param param1 Parameter 1.
 * @param param2 Parameter 2.
 * @return A new instance of fragment CatalogFragment.
 */
// TODO: Rename and change types and number of parameters
public static CatalogFragment newInstance(String param1, String param2) {
    CatalogFragment fragment = new CatalogFragment();
    Bundle args = new Bundle();
    args.putString(ARG_PARAM1, param1);
    args.putString(ARG_PARAM2, param2);
    fragment.setArguments(args);
    return fragment;
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    auth = FirebaseAuth.getInstance();
    isInternetOn();
    if (getArguments() != null) {
        mParam1 = getArguments().getString(ARG_PARAM1);
        mParam2 = getArguments().getString(ARG_PARAM2);
    }
    curr_email = auth.getCurrentUser().getEmail();
    classes = new ArrayList<String>();
    activities = new ArrayList<String>();
    globalVariables.clearClasses();
    classes.clear();
    activities.clear();
    getgroup();
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    view = inflater.inflate(R.layout.fragment_catalog, container, false);
    btnjoingroup = (Button) view.findViewById(R.id.btn_joingroup);
    norecord = (TextView) view.findViewById(R.id.norecords);
    assessment = (TableLayout) view.findViewById(R.id.recordstable);
    btnjoingroup.setOnClickListener(this);
    return view;
}

@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.btn_joingroup:
            //Start activity one
            dref = FirebaseDatabase.getInstance().getReference();
            groupcodes = new ArrayList<String>();

```

```

        AlertDialog.Builder builder = new AlertDialog.Builder(v.getContext());
        builder.setTitle("Enter Group Code");
        final EditText input = new EditText(v.getContext());
        // Specify the type of input expected; this, for example, sets the
input as a password, and will mask the text
        input.setInputType(InputType.TYPE_CLASS_TEXT);
        builder.setView(input);
        builder.setPositiveButton("Yes", new DialogInterface.OnClickListener()
{
    @Override
    public void onClick(DialogInterface dialog, int which) {
        grpCode = input.getText().toString();
        dref2 = database.getReference().child("classes");
        dref2.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                if(dataSnapshot.exists()){
                    for(DataSnapshot d : dataSnapshot.getChildren()){
                        groupcodes.add(d.getKey());
                        if (Objects.equals(d.getKey(), grpCode)) {
                            cnt++;
                        }
                    }
                }
                if(cnt == 0){
                    AlertDialog.Builder builder2 = new
AlertDialog.Builder(getActivity());
                    builder2.setMessage("Group Code Not Found")
                        .setCancelable(false)
                        .setNegativeButton("Ok", new
DialogInterface.OnClickListener() {
                            public void onClick(DialogInterface
dialog, int id) {
                                dialog.cancel();
                            }
                        })
                    //Set your icon here
                    .setTitle("Warning")
                    .setIcon(R.mipmap.ic_logo);
                    AlertDialog alert = builder2.create();
                    alert.show();//showing the dialog
                }else{
                    for(int k = 0 ; k<= groupcodes.size()-1 ; k++){
                        if (Objects.equals(groupcodes.get(k),
grpCode)) {
                            id = globalVariables.getstudentKey();
                            //
                            joingroup(id, grpCode);
                        }
                    }
                }
            }
        });
    }
});
    @Override
    public void onCancelled(DatabaseError databaseError) {
    }
});

```



```

        }
    });

    builder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
        }
    });
    builder.show();
    break;
    // Do this for all buttons.
}
}
private void getyourclasses(int index){
    final int indexx = index;
    dref =
database.getReference().child("classes").child(globalVariables.getclasses(indexx));
    dref.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            Map<String, Object> newPost = (Map<String, Object>)
dataSnapshot.getValue();
            getrecords(globalVariables.getclasses(indexx),
newPost.get("grp_teacher_key"));
        }
        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });
}
private void getgroup(){
    classes.clear();
    dref = database.getReference().child("classes");
    dref.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            if(dataSnapshot.exists()){
                for(DataSnapshot d : dataSnapshot.getChildren()){
                    classes.add(d.getKey());
                }
            }
            for(int k = 0 ; k<=classes.size()-1 ; k++){
                getyourgroup(classes.get(k));
            }
        }

        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });
}
}

```

```

        private void getrecords(String groupcode, Object teacherkey){
            final String curr_group = groupcode;
            final String curr_teacherkey = teacherkey.toString();
            dref =
            database.getReference().child("assessment").child(curr_teacherkey).child(curr_group).child(globalVariables.getstudentKey());
            dref.addListenerForSingleValueEvent(new ValueEventListener() {
                @Override
                public void onDataChange(DataSnapshot dataSnapshot) {
                    if(dataSnapshot.exists()){
                        for(DataSnapshot d : dataSnapshot.getChildren()){
                            activities.add(d.getKey());
                        }
                    }
                    for(int k = 0 ; k<=activities.size()-1 ; k++){
                        getyourassessment(activities.get(k), curr_teacherkey,curr_group);
                    }
                }

                @Override
                public void onCancelled(DatabaseError databaseError) {

                }
            });
        }

        private void getyourassessment(String activities, String curr_teacherkey, String curr_group){
            final String group = curr_group;
            final String teacherkey = curr_teacherkey;
            final String curr_activities = activities;
            dref =
            database.getReference().child("assessment").child(teacherkey).child(group).child(globalVariables.getstudentKey()).child(curr_activities);
            dref.addListenerForSingleValueEvent(new ValueEventListener() {
                @Override
                public void onDataChange(DataSnapshot dataSnapshot) {
                    if(dataSnapshot.exists()) {
                        norecord.setVisibility(View.GONE);
                        Map<String, Object> newPost = (Map<String, Object>)
                        dataSnapshot.getValue();
                        Object score = newPost.get("score");
                        Object timestamp = newPost.get("date_quiztaken");
                        TableRow tr = new TableRow(getContext());
                        tr.setLayoutParams(new
                        TableRow.LayoutParams(TableRow.LayoutParams.MATCH_PARENT,
                        TableRow.LayoutParams.FILL_PARENT));
                        TextView youractivity = new TextView(getContext());
                        if (Objects.equals(curr_activities, "quiz")) {
                            youractivity.setText("Chapter Quiz");
                        } else {
                            youractivity.setText(curr_activities);
                        }
                        youractivity.setLayoutParams(new TableRow.LayoutParams(1));
                        youractivity.setTextSize(TypedValue.COMPLEX_UNIT_PX,
                        getResources().getDimension(R.dimen.records2));
                        youractivity.setTextColor(getResources().getColor(R.color.black));
                        TextView yourscore = new TextView(getContext());
                        yourscore.setText(score.toString());
                    }
                }
            });
        }

```

```

        yourscore.setLayoutParams(new TableRow.LayoutParams(1));
        yourscore.setTextSize(TypedValue.COMPLEX_UNIT_PX,
            getResources().getDimension(R.dimen.records2));
        yourscore.setTextColor(getResources().getColor(R.color.black));
        TextView yourtimestamp = new TextView(getContext());
        yourtimestamp.setText(timestamp.toString());
        yourtimestamp.setLayoutParams(new TableRow.LayoutParams(1));
        yourtimestamp.setTextSize(TypedValue.COMPLEX_UNIT_PX,
            getResources().getDimension(R.dimen.records2));
        yourtimestamp.setTextColor(getResources().getColor(R.color.black));
        tr.addView(youractivity);
        tr.addView(yourscore);
        tr.addView(yourtimestamp);
        assessment.addView(tr, new TableLayout.LayoutParams(0,
TableLayout.LayoutParams.FILL_PARENT, 1));
    }
}
@Override
public void onCancelled(DatabaseError databaseError) {

}

});

}

public final boolean isInternetOn() {
    // get Connectivity Manager object to check connection
    ConnectivityManager connec =
    (ConnectivityManager) getActivity().getSystemService(Context.CONNECTIVITY_SERVICE);
    // Check for network connections
    if ( connec.getNetworkInfo(0).getState() ==
    android.net.NetworkInfo.State.CONNECTED ||
        connec.getNetworkInfo(0).getState() ==
    android.net.NetworkInfo.State.CONNECTING ||
        connec.getNetworkInfo(1).getState() ==
    android.net.NetworkInfo.State.CONNECTING ||
        connec.getNetworkInfo(1).getState() ==
    android.net.NetworkInfo.State.CONNECTED ) {
        // if connected with internet
        return true;
    } else if (
        connec.getNetworkInfo(0).getState() ==
    android.net.NetworkInfo.State.DISCONNECTED ||
        connec.getNetworkInfo(1).getState() ==
    android.net.NetworkInfo.State.DISCONNECTED ) {
        Toast.makeText(getActivity(), " No internet connection",
        Toast.LENGTH_LONG).show();
        Intent intent = new Intent(getActivity(), OfflineScreen.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
        return false;
    }
    return false;
}

private void getyourgroup(String curr_group){
    final String group = curr_group;
    dref = database.getReference().child("classes").child(group).child("member");
    dref.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            if (dataSnapshot.exists()) {

```

```

        for (DataSnapshot d : dataSnapshot.getChildren()) {
globalVariables.setcurrclassmemcnt(dataSnapshot.getChildrenCount());
            if (Objects.equals(globalVariables.getstudentKey(),
d.getKey())) {
                btnjoingroup.setVisibility(view.INVISIBLE);
                globalVariables.setclasses(group);
                for(l = 0 ; l<=globalVariables.getclassessize()-1 ; l++) {
                    getyourclasses(l);
                }
            }
            @Override
            public void onCancelled(DatabaseError databaseError) {
            }
        });

    }
    // TODO: Rename method, update argument and hook method into UI event
    public void onPressed(Uri uri) {
        if (mListener != null) {
            mListener.onFragmentInteraction(uri);
        }
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        if (context instanceof OnFragmentInteractionListener) {
            mListener = (OnFragmentInteractionListener) context;
        }
        globalVariables.clearClasses();
    }

    @Override
    public void onDetach() {
        super.onDetach();
        mListener = null;
        classes.clear();
        activities.clear();
    }

    @Override
    public void onResume() {
        super.onResume();
        mListener = null;
        classes.clear();
        activities.clear();
    }

    private void joingroup(final String id, final String grpCode) {
        dref2 = database.getReference().child("classes").child(grpCode);
        dref2.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                Map<String, Object> newPost = (Map<String, Object>)
dataSnapshot.getValue();
                final String grpcapacity = newPost.get("grp_capacity").toString();
                dref2 =
database.getReference().child("classes").child(grpCode).child("member");
                dref2.addListenerForSingleValueEvent(new ValueEventListener() {
                    @Override
                    public void onDataChange(DataSnapshot dataSnapshot) {
                        if(Integer.parseInt(grpcapacity) >
dataSnapshot.getChildrenCount()){

```

```

        SimpleDateFormat simpleDateFormat = new
SimpleDateFormat("dd-MM-yyyy-hh-mm-ss");
        String date_requested = simpleDateFormat.format(new
Date());
        classes group = new classes(id, grpCode, date_requested);

dref.child("classes").child(grpCode).child("pending_members").child(id).setValue(group)
;
        AlertDialog.Builder builder = new
AlertDialog.Builder(getContext());
        builder.setTitle("Wait For Approval");
        builder.setMessage("Please wait for your teacher's
approval");
        builder.setPositiveButton("Ok", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which)
{
                dialog.cancel();
                getActivity().recreate();
            }
        });
        builder.show();
    }else{
        AlertDialog.Builder builder = new
AlertDialog.Builder(getContext());
        builder.setTitle("Warning");
        builder.setMessage("Group is already full");
        builder.setPositiveButton("Ok", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which)
{
                dialog.cancel();
                getActivity().recreate();
            }
        });
        builder.show();
    }
}

@Override
public void onCancelled(DatabaseError databaseError) {

}

});
}

@Override
public void onCancelled(DatabaseError databaseError) {

}

});
}

public interface OnFragmentInteractionListener {
    // TODO: Update argument type and name
    void onFragmentInteraction(Uri uri);
}
}

PhotoFragment.java
public class PhotosFragment extends Fragment {

```

```

// TODO: Rename parameter arguments, choose names that match
// the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
private static final String ARG_PARAM1 = "param1";
private static final String ARG_PARAM2 = "param2";
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference dref;
String curr_email, curr_group, subject;
Object grp_name;
int size;
Button[] btngroup;
List<String> sections;
List<String> fromdbsections;
List<String> grpname;
List<String> grpsubject;
List<String> grpteacherkey;
LinearLayout.LayoutParams params;
LinearLayout rl;
ListView listview;
ArrayList<Object> list = new ArrayList<>();
ArrayList<Object> classes;
ArrayAdapter<Object> adapter;
TextView nogroup;
int y = 0;
// TODO: Rename and change types of parameters
private String mParam1;
private String mParam2;
private FirebaseAuth auth;
private OnFragmentInteractionListener mListener;
public PhotosFragment() {
    // Required empty public constructor
}
/**
 * Use this factory method to create a new instance of
 * this fragment using the provided parameters.
 *
 * @param param1 Parameter 1.
 * @param param2 Parameter 2.
 * @return A new instance of fragment PhotosFragment.
 */
// TODO: Rename and change types and number of parameters
public static PhotosFragment newInstance(String param1, String param2) {
    PhotosFragment fragment = new PhotosFragment();
    Bundle args = new Bundle();
    args.putString(ARG_PARAM1, param1);
    args.putString(ARG_PARAM2, param2);
    fragment.setArguments(args);
    return fragment;
}
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    auth = FirebaseAuth.getInstance();
    if (getArguments() != null) {
        mParam1 = getArguments().getString(ARG_PARAM1);
        mParam2 = getArguments().getString(ARG_PARAM2);
    }
    curr_email = auth.getCurrentUser().getEmail();
    getgroup();
}

```

```

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                        Bundle savedInstanceState) {
    final View rootView = inflater.inflate(R.layout.fragment_photos, container,
false);
    params = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT);
    params.setMargins(0,10,0,0);
    rl = (LinearLayout) rootView.findViewById(R.id.fragment_photos_layout);
    nogroup = (TextView) rootView.findViewById(R.id.txt_nogroup);
    listview = (ListView) rootView.findViewById(R.id.listview);
    adapter = new ArrayAdapter<Object>(getContext(), R.layout.listview_text, list);
    listview.setAdapter(adapter);
    sections = new ArrayList<String>();
    fromdbsections = new ArrayList<String>();
    grpname = new ArrayList<String>();
    grpsubject = new ArrayList<String>();
    grpteacherkey = new ArrayList<String>();
    classes = new ArrayList<>();
    btngroup = new Button[size+1];
    if(globalVariables.getclassessize() == 0){
        nogroup.setVisibility(View.VISIBLE);
    }else {
        dref = database.getReference().child("classes");
        dref.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                if (dataSnapshot.exists()) {
                    for (DataSnapshot d : dataSnapshot.getChildren()) {
                        fromdbsections.add(d.getKey());
                    }
                    size = fromdbsections.size()-1;
                    for (int k = 0; k <= globalVariables.getclassessize() - 1; k++){
{
                        groupcheck(globalVariables.getclasses(k));
                    }
                }
            }
            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });
        // Inflate the layout for this fragment
    }
    listview.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
            final String pickedTopic =
String.valueOf(parent.getItemAtPosition(position));
            for(int j = 0 ; j<=globalVariables.getclassessize()-1 ; j++){
                dref =
database.getReference().child("classes").child(globalVariables.getclasses(j));
                dref.addListenerForSingleValueEvent(new ValueEventListener() {
                    @Override
                    public void onDataChange(DataSnapshot dataSnapshot) {
                        Map<String, Object> newPost = (Map<String, Object>)
dataSnapshot.getValue();
                        grp_name = newPost.get("grp_name");
                        if(Objects.equals(pickedTopic, grp_name.toString())){

```

```

globalVariables.setcurr_subject(newPost.get("grp_subject").toString());

globalVariables.setteacherkey(newPost.get("grp_teacher_key").toString());
globalVariables.setgrpCode(dataSnapshot.getKey());
Intent intent=new Intent(getActivity(),
SelectedCourseScreen.class);
startActivity(intent);
    }
}

@Override
public void onCancelled(DatabaseError databaseError) {

}

});
}
}
});
return rootView;
}
// TODO: Rename method, update argument and hook method into UI event
public void onPressed(Uri uri) {
    if (mListener != null) {
        mListener.onFragmentInteraction(uri);
    }
}

@Override
public void onAttach(Context context) {
    super.onAttach(context);
    if (context instanceof OnFragmentInteractionListener) {
        mListener = (OnFragmentInteractionListener) context;
    }
}

@Override
public void onDetach() {
    super.onDetach();
    mListener = null;
    fromdbsections.clear();
    grpname.clear();
    grpteacherkey.clear();
    grpsubject.clear();
    sections.clear();
    //globalVariables.clearClasses();
}

private void getgroup(){
    dref = database.getReference().child("classes");
    dref.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            if(dataSnapshot.exists()){
                for(DataSnapshot d : dataSnapshot.getChildren()){
                    classes.add(d.getKey());
                }
            }
            for(int k = 0 ; k<=classes.size()-1 ; k++){
                getyourgroup(classes.get(k).toString());
            }
        }
    }
}

```



```

        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });

}

private void getyourgroup(String curr_group){
    final String group = curr_group;
    dref = database.getReference().child("classes").child(group).child("member");
    dref.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            if (dataSnapshot.exists()) {
                for (DataSnapshot d : dataSnapshot.getChildren()) {

globalVariables.setcurrclassmemcnt(dataSnapshot.getChildrenCount());
                    if (Objects.equals(globalVariables.getstudentKey(),
d.getKey())) {
                        Log.e("Check", d.getKey());
                    }
                }
            }
        }
        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });

}

private void groupcheck(String yourgroups){
    curr_group = yourgroups;
    y = 0;
    do{
        if(Objects.equals(fromdbsections.get(y), curr_group)){
            dref = database.getReference().child("classes").child(curr_group);
            dref.addChildEventListener(new ChildEventListener() {
                @Override
                public void onChildAdded(DataSnapshot dataSnapshot, String s) {
                    if(Objects.equals(dataSnapshot.getKey(), "grp_name")){
                        subject = dataSnapshot.getValue().toString();
                        list.add(subject);
                        adapter.notifyDataSetChanged();
                    }
                }
                @Override
                public void onChildChanged(DataSnapshot dataSnapshot, String s) {

                }
                @Override
                public void onChildRemoved(DataSnapshot dataSnapshot) {

                }
                @Override
                public void onChildMoved(DataSnapshot dataSnapshot, String s) {

                }
                @Override
                public void onCancelled(DatabaseError databaseError) {

                }
            });
        }
    }
};

```

```

        }
        y++;
    }while(y<=size);
}

/**
 * This interface must be implemented by activities that contain this
 * fragment to allow an interaction in this fragment to be communicated
 * to the activity and potentially other fragments contained in that
 * activity.
 * <p>
 * See the Android Training lesson <a href=
 * "http://developer.android.com/training/basics/fragments/communicating.html"
 * >Communicating with Other Fragments</a> for more information.
 */
public interface OnFragmentInteractionListener {
    // TODO: Update argument type and name
    void onFragmentInteraction(Uri uri);
}
}

SelectedCourseScreen.java
public class SelectedCourseScreen extends AppCompatActivity {
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    boolean isenabled = true;
    DatabaseReference dref;
    ListView listview;
    ArrayList<Object> list = new ArrayList<>();
    ArrayAdapter<Object> adapter;
    FirebaseAuth auth;
    Object about_subject2, teacherfname, teacherlname;
    int size, x;
    String teacherkey, quizstatus;
    List<String> lessons;
    Button[] btngroup;
    LinearLayout.LayoutParams params;
    LinearLayout ll;
    TextView about_subject, subj_title, subj_members, subj_teacher;
    AlertDialog.Builder builder;
    AlertDialog builder2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        auth = FirebaseAuth.getInstance();
        if (auth.getCurrentUser() == null) {
            startActivity(new Intent(SelectedCourseScreen.this, LoginScreen.class));
            finish();
        }
        isInternetOn();
        setContentView(R.layout.activity_selected_course_screen);
        teacherkey = globalVariables.getteacherkey();
        builder = new AlertDialog.Builder(SelectedCourseScreen.this);
        listview = (ListView) findViewById(R.id.listview);
        about_subject = (TextView) findViewById(R.id.txt_aboutsubj);
        subj_title = (TextView) findViewById(R.id.txt_subj_title);
        subj_members = (TextView) findViewById(R.id.txt_subj_members_count);
        subj_teacher = (TextView) findViewById(R.id.txt_subj_teacher);
        subj_title.setText(globalVariables.getcurr_subject());
        int counter = (int) globalVariables.getcurrclassmemcnt() - 1;
        subj_members.setText("You have " + counter + " classmates");
        adapter = new ArrayAdapter<Object>(this, R.layout.listview_text2, list);
        listview.setAdapter(adapter);
    }
}

```

```

        dref =
database.getReference().child("assessment").child(teacherkey).child(globalVariables.get
grpCode()).child(globalVariables.getstudentKey()).child("quiz");
        dref.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                if(dataSnapshot.exists()){
                    quizstatus = "You've already taken the quiz.";
                    isenabled = false;
                }else{
                    quizstatus = "Chapter Quiz";
                }
            }
            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });
        dref = database.getReference().child("members").child(teacherkey);
        dref.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                Map<String, Object> newPost = (Map<String, Object>)
dataSnapshot.getValue();
                teacherfname = newPost.get("firstname");
                teacherlname = newPost.get("lastname");
                subj_teacher.setText("Prof. "+teacherfname.toString() + " " +
teacherlname.toString());
            }
            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });
        dref =
database.getReference().child("subjects").child(teacherkey).child(globalVariables.getcu
rr_subject());
        dref.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                Map<String, Object> newPost = (Map<String, Object>)
dataSnapshot.getValue();
                about_subject2 = newPost.get("subj_description");
                about_subject.setText("\"" + about_subject2 + "\"");
            }
            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });

        dref.addChildEventListener(new ChildEventListener() {
            // Retrieve new posts as they are added to Firebase
            @Override
            public void onChildAdded(DataSnapshot snapshot, String previousChildKey) {
                if(!Objects.equals(snapshot.getKey(), "subj_date_added") &&
!Objects.equals(snapshot.getKey(), "subj_description")){
                    list.add(snapshot.getKey());
                    globalVariables.setchapters(snapshot.getKey());
                    adapter.notifyDataSetChanged();
                }
            }
        });

```

```

    }
    @Override
    public void onChildChanged(DataSnapshot dataSnapshot, String s) {

    }
    @Override
    public void onChildRemoved(DataSnapshot dataSnapshot) {

    }
    @Override
    public void onChildMoved(DataSnapshot dataSnapshot, String s) {

    }
    @Override
    public void onCancelled(DatabaseError databaseError) {

    }
});

listview.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
        builder.setTitle("Select Lesson");
        ll = new LinearLayout(SelectedCourseScreen.this);
        ll.setOrientation(LinearLayout.VERTICAL);
        params = new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT);
        params.setMargins(100,50,100,50);
        String pickedTopic =
String.valueOf(parent.getItemAtPosition(position));
        globalVariables.setcurr_lesson(pickedTopic);
        lessons = new ArrayList<String>();
        dref =
database.getReference().child("subjects").child(teacherkey).child(globalVariables.getcu
rr_subject()).child(pickedTopic);
        dref.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                if(dataSnapshot.exists()) {
                    for(DataSnapshot d : dataSnapshot.getChildren()){
                        if(!Objects.equals(d.getKey(), "chap_date_added")){
                            lessons.add(d.getKey());
                        }
                    }
                    size = lessons.size();
                    btngroup = new Button[size];
                    Log.e("G", "" + size);
                    for(x = 0 ; x<=size-1 ; x++){
                        final int y = x;
                        btngroup[x] = new Button(SelectedCourseScreen.this);
                        if(Objects.equals(lessons.get(x), "quiz")){
                            btngroup[x].setText(quizstatus);
                            if(!isEnabled){
                                btngroup[x].setEnabled(isEnabled);
                            }else{
                                btngroup[x].setEnabled(isEnabled);
                            }
                        }
                    }
                }
            }
        });
    }
});

```

```

        }else{
            btngroup[x].setText(lessons.get(x));
        }

btngroup[x].setBackgroundColor(btngroup[x].getContext().getResources().getColor(R.color
    .darkorange));

        btngroup[x].setLayoutParams(params);
        btngroup[x].setId(x);
        Log.e("Checks", "" + btngroup[x].getId());
        btngroup[y].setOnClickListener(new

View.OnClickListener() {

            @Override
            public void onClick(View v) {
                if (btngroup[y].getId() == ((Button)

v).getId()){

                    int id = btngroup[y].getId();
                    if(btngroup[id].getText() == "Chapter

Quiz"){

                        Intent intent=new
                        Intent(SelectedCourseScreen.this, ChapterQuiz.class);
                        intent.putExtra("lesson",
                        lessons.get(id));

                        globalVariables.setcurr_topic(lessons.get(id));

                        builder2.dismiss();
                        startActivity(intent);
                    }else{
                        Intent intent=new
                        Intent(SelectedCourseScreen.this, Read_Lesson_Screen.class);
                        intent.putExtra("lesson",
                        lessons.get(id));

                        globalVariables.setcurr_topic(lessons.get(id));

                        builder2.dismiss();
                        startActivity(intent);
                    }
                }
            }
        });
        ll.addView(btngroup[x]);
    }
}

@Override
public void onCancelled(DatabaseError databaseError) {

}

});
builder.setView(ll);
builder2 = builder.show();
}
});

}

@Override
public void onStop(){
    super.onStop();
    if(builder2 != null){
        builder2.dismiss();
    }
}

```

```

        }
        this.recreate();
    }
    @Override
    public void onResume(){
        super.onResume();
    }

    @Override
    public void onBackPressed() {
        Intent intent=new Intent(SelectedCourseScreen.this, MainScreen.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
    }
    public final boolean isInternetOn() {
        // get Connectivity Manager object to check connection
        ConnectivityManager connec =
        (ConnectivityManager)getSystemService(getBaseContext().CONNECTIVITY_SERVICE);
        // Check for network connections
        if ( connec.getNetworkInfo(0).getState() ==
        android.net.NetworkInfo.State.CONNECTED ||
            connec.getNetworkInfo(0).getState() ==
        android.net.NetworkInfo.State.CONNECTING ||
            connec.getNetworkInfo(1).getState() ==
        android.net.NetworkInfo.State.CONNECTING ||
            connec.getNetworkInfo(1).getState() ==
        android.net.NetworkInfo.State.CONNECTED ) {
            // if connected with internet
            return true;
        } else if (
            connec.getNetworkInfo(0).getState() ==
        android.net.NetworkInfo.State.DISCONNECTED ||
            connec.getNetworkInfo(1).getState() ==
        android.net.NetworkInfo.State.DISCONNECTED ) {
            Toast.makeText(SelectedCourseScreen.this, " No internet connection",
            Toast.LENGTH_LONG).show();
            Intent intent = new Intent(SelectedCourseScreen.this,OfflineScreen.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);
            return false;
        }
        return false;
    }
}

```

Read_Lesson_Screen.java

```

public class Read_Lesson_Screen extends AppCompatActivity {
    FirebaseAuth auth;
    String s, teacherkey, curr_subject, curr_lesson, pdflink;
    Button takequiz;
    long startTime, endTime, spent;
    DatabaseReference dref;
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    Object lesson_material, lesson_sidenotes, quiz_started;
    private FloatingActionButton fab;
    LinearLayout ll;
    LinearLayout.LayoutParams params;
    Date date1, date2;
    int totalnumberofpages = 0;
    @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    auth = FirebaseAuth.getInstance();
    if (auth.getCurrentUser() == null) {
        startActivity(new Intent(Read_Lesson_Screen.this, LoginScreen.class));
        finish();
    }
    setContentView(R.layout.activity_read__lesson__screen);
    fab = (FloatingActionButton) findViewById(R.id.fab);
    isInternetOn();
    Bundle extras = getIntent().getExtras();
    s = extras.getString("lesson");
    takequiz = (Button) findViewById(R.id.btn_takequiz);
    takequiz.setVisibility(View.INVISIBLE);
    teacherkey = globalVariables.getteacherkey();
    curr_subject = globalVariables.getcurr_subject();
    curr_lesson = globalVariables.getcurr_lesson();
    dref =
database.getReference().child("subjects").child(teacherkey).child(curr_subject).child(c
urr_lesson).child(s);
    dref.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            Map<String, Object> newPost = (Map<String, Object>)
dataSnapshot.getValue();
            lesson_material = newPost.get("lesson_material");
            lesson_sidenotes = newPost.get("lesson_sidenotes");
            Log.e("Check", lesson_material.toString());
            downloadFile(lesson_material.toString());
        }
        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });
    dref =
database.getReference().child("subjects").child(teacherkey).child(curr_subject).child(c
urr_lesson).child(s).child("quiz");
    dref.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            if(!dataSnapshot.exists()){
                takequiz.setVisibility(View.VISIBLE);
                takequiz.setText("No Quiz Available");
                takequiz.setEnabled(false);
            }else{
                Map<String, Object> newPost = (Map<String, Object>)
dataSnapshot.getValue();
                quiz_started = newPost.get("quiz_started");
                if(Objects.equals(quiz_started.toString(), "no")){
                    takequiz.setVisibility(View.VISIBLE);
                    takequiz.setText("No Quiz Available");
                    takequiz.setEnabled(false);
                }else{
                    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("EEE,
d MMM yyyy HH:mm:ss z");
                    simpleDateFormat.setTimeZone(TimeZone.getTimeZone("GMT"));
                    String haha = newPost.get("quiz_expiration").toString();
                    try {
                        date1 = simpleDateFormat.parse(haha);

```

```

        date2 = simpleDateFormat.parse(simpleDateFormat.format(new
Date()));

        int checkExpired = date1.compareTo(date2);
        if(checkExpired == -1){
            takequiz.setVisibility(View.VISIBLE);
            takequiz.setText("Quiz Expired");
            takequiz.setEnabled(false);
        }else{
            takequiz.setText("Quiz Available");
            takequiz.setEnabled(true);
        }
    } catch (ParseException e) {
        e.printStackTrace();
    }
}

@Override
public void onCancelled(DatabaseError databaseError) {

}

});
dref =
database.getReference().child("assessment").child(teacherkey).child(globalVariables.get
grpCode()).child(globalVariables.getstudentKey()).child(s);
dref.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        if(dataSnapshot.exists()){
            takequiz.setVisibility(View.VISIBLE);
            takequiz.setText("You Have Already Taken The Quiz.");
            takequiz.setEnabled(false);
        }
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }

});
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        ll = new LinearLayout(Read_Lesson_Screen.this);
        ll.setOrientation(LinearLayout.VERTICAL);
        params = new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT);
        params.setMargins(100,50,100,50);
        AlertDialog.Builder builder = new
AlertDialog.Builder(Read_Lesson_Screen.this);
        builder.setTitle("Sidenotes");
        TextView htmltext = new TextView(Read_Lesson_Screen.this);
        String sidenote = "<body>" + lesson_sidenotes.toString() + "</body>";
        htmltext.setText(Html.fromHtml(sidenote));
        ll.addView(htmltext);
        builder.setMessage(htmltext.getText());
        builder.setPositiveButton("Ok", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {

```



```

        dialog.cancel();
    }
    });
    builder.show();
}
});
}
private void downloadFile(String pdfname) {
    pdflink = pdfname;
    FirebaseStorage storage = FirebaseStorage.getInstance();
    StorageReference storageRef = storage.getReferenceFromUrl("gs://ezbio-98be1.appspot.com");
    storageRef.child(pdflink).getBytes(Long.MAX_VALUE).addOnSuccessListener(new OnSuccessListener<byte[]>() {
        @Override
        public void onSuccess(byte[] bytes) {
            Log.e("CHEKING", s);
            FileCacher<byte[]> byteCacher = new FileCacher<>(Read_Lesson_Screen.this, s);
            List<String> topics = new ArrayList<String>();
            topics.add(s);
            FileCacher<List<String>> topic = new FileCacher<>(Read_Lesson_Screen.this, "topics");
            try{
                byteCacher.writeCache(bytes);
                topic.writeCache(topics);
            }catch (IOException e) {
                e.printStackTrace();
            }
            globalVariables.setpdf(bytes);
            PDFView pdfView = (PDFView) findViewById(R.id.pdfView);
            pdfView.fromBytes(bytes)
                .enableSwipe(true) // allows to block changing pages using
swipe
                .swipeHorizontal(false)
                .enableDoubletap(true)
                .defaultPage(0)
                .enableAnnotationRendering(false) // render annotations
(such as comments, colors or forms)
                .password(null)
                .scrollHandle(null)
                .onPageChange(new OnPageChangeListener() {
                    @Override
                    public void onPageChanged(int page, int pageCount) {
                        totalnumberofpages = pageCount;
                    }
                })
                .onPageScroll(new OnPageScrollListener() {
                    @Override
                    public void onPageScrolled(int page, float
positionOffset) {
positionOffset + " " + page);
                        Log.e("CHeck", " "+ totalnumberofpages + " " +
                        if(page == (totalnumberofpages-3)){
                            takequiz.setVisibility(View.VISIBLE);
                        }
                    }
                })
                .load();
        }
    });
}
}

```

```

        });
    }

    public void takequiz(View v)
    {
        AlertDialog.Builder builder = new AlertDialog.Builder(Read_Lesson_Screen.this);
        builder.setTitle("Are you ready to take the quiz?");
        builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                Intent intent = new Intent(getBaseContext(), Quiz.class);
                Bundle extras = new Bundle();
                extras.putString("lesson",s);
                intent.putExtras(extras);
                startActivity(intent);
            }
        });
        builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.cancel();
            }
        });
        builder.show();
    }

    @Override
    protected void onResume() {
        super.onResume();
        startTime = System.currentTimeMillis();
    }

    @Override
    protected void onPause() {
        super.onPause();
        endTime = System.currentTimeMillis();
        spent = endTime - startTime;
        spent = spent / 1000;
        globalVariables.setTimespentonRead(spent,"no");
        Log.d("Check", "" + spent + "seconds");
    }

    public void onBackPressed() {
        if(Objects.equals(globalVariables.getonquiz(), "yes")){
            new AlertDialog.Builder(this)
                .setTitle("Information")
                .setMessage("Sorry you can't disregard your quiz.")
                .setPositiveButton(android.R.string.yes, null).create().show();
        }else{
            Intent intent=new Intent(Read_Lesson_Screen.this,
SelectedCourseScreen.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);
        }
    }

    public final boolean isInternetOn() {
        // get Connectivity Manager object to check connection
        ConnectivityManager connec =
        (ConnectivityManager)getSystemService(getBaseContext().CONNECTIVITY_SERVICE);
        // Check for network connections
    }

```

```

        if ( connec.getNetworkInfo(0).getState() ==
android.net.NetworkInfo.State.CONNECTED ||
            connec.getNetworkInfo(0).getState() ==
android.net.NetworkInfo.State.CONNECTING ||
            connec.getNetworkInfo(1).getState() ==
android.net.NetworkInfo.State.CONNECTING ||
            connec.getNetworkInfo(1).getState() ==
android.net.NetworkInfo.State.CONNECTED ) {
            // if connected with internet
            return true;
        } else if (
            connec.getNetworkInfo(0).getState() ==
android.net.NetworkInfo.State.DISCONNECTED ||
            connec.getNetworkInfo(1).getState() ==
android.net.NetworkInfo.State.DISCONNECTED ) {

            Toast.makeText(Read_Lesson_Screen.this, " No internet connection",
Toast.LENGTH_LONG).show();
            Intent intent = new Intent(Read_Lesson_Screen.this,OfflineScreen.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);
            return false;
        }
        return false;
    }
}

```

Quiz.java

```

public class Quiz extends AppCompatActivity {
    String s, curr_subject,curr_lesson,curr_topic, findings, grpCode, studkey,
teacherkey, timelimit;
    Object question, canswer, q_type, q_randomize;
    FirebaseAuth auth;
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    DatabaseReference dref;
    Double totalscore = 0.0;
    int scored, numOfItems;
    RadioGroup[] rdgroup = new RadioGroup[100];
    RadioButton rdbtn;
    TextView questions, timer;
    String[] youranswers;
    String[] correctans;
    List<String> missed;
    long startTime, endTime, spent;
    List<String> questions_key;
    List<String> choices;
    int index = 0;
    int pointer = 1;
    LinearLayout quizlay;
    int tomillis;
    public int counter;
    AlertDialog ad;
    CountdownTimer timer2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        auth = FirebaseAuth.getInstance();
        if (auth.getCurrentUser() == null) {
            startActivity(new Intent(Quiz.this, LoginScreen.class));
            finish();
        }
        setContentView(R.layout.activity_quiz);
    }
}

```

```

scored = globalVariables.getscore();
curr_subject = globalVariables.getcurr_subject();
curr_lesson = globalVariables.getcurr_lesson();
curr_topic = globalVariables.getcurr_topic();
grpCode = globalVariables.getgrpCode();
studkey = globalVariables.getstudentKey();
teacherkey = globalVariables.getteacherkey();
Bundle extras = getIntent().getExtras();
s = extras.getString("lesson");
quizlay = (LinearLayout) findViewById(R.id.quiz_layout);
Button submit = (Button) findViewById(R.id.btn_submit);
timer = (TextView) findViewById(R.id.timer);
dref =
database.getReference().child("subjects").child(teacherkey).child(curr_subject).child(c
urr_lesson).child(curr_topic).child("quiz");
dref.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot snapshot) {
        Map<String, Object> newPost = (Map<String, Object>)
snapshot.getValue();
        String numOfItems1 = newPost.get("quiz_numItems").toString();
        q_randomize = newPost.get("quiz_randomize");
        timelimit = newPost.get("quiz_timelimit").toString();
        counter = Integer.parseInt(timelimit);
        tomillis = (counter * 1000) + 2000;
        timer2 = new CountdownTimer(tomillis,1000) {
            @Override
            public void onTick(long millisUntilFinished) {
                timer.setText(String.format("%02d:%02d:%02d", counter / 60,
counter % 60, 0));
                counter--;
            }
            @Override
            public void onFinish() {
                repeatquiz("timeexpired");
                timer2.cancel();
            }
        }.start();
        numOfItems = Integer.parseInt(numOfItems1);
        questions_key = new ArrayList<String>();
        missed = new ArrayList<String>();
        dref =
database.getReference().child("subjects").child(teacherkey).child(curr_subject).child(c
urr_lesson).child(curr_topic).child("quiz").child("questions");
        dref.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                if(dataSnapshot.exists()){
                    for(DataSnapshot d : dataSnapshot.getChildren()){
                        questions_key.add(d.getKey());
                    }
                    for(int m = 1; m<=numOfItems; m++) {
                        rdgroup[m] = new RadioGroup(Quiz.this);
                    }
                    if(Objects.equals(q_randomize.toString(), "yes")){
                        Collections.shuffle(questions_key,new
Random(System.nanoTime()));
                    }
                    for(int re = questions_key.size() ; re<numOfItems ; re-
-){
                        questions_key.remove(re);

```

```

        }
        correctans = new String[questions_key.size()];
        youranswers = new String[questions_key.size()];
        getquestions();
    }else{
        for(int re = questions_key.size() ; re<numOfItems ; re-
-){
            questions_key.remove(re);
        }
        correctans = new String[questions_key.size()];
        youranswers = new String[questions_key.size()];
        getquestions();
    }
}

@Override
public void onCancelled(DatabaseError databaseError) {

}

});
}
@Override
public void onCancelled(DatabaseError databaseError) {

}

});
submit.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        int selected = rdgroup[index+1].getCheckedRadioButtonId();
        if(selected == -1){
            android.app.AlertDialog.Builder builder = new
android.app.AlertDialog.Builder(Quiz.this);
            builder.setTitle("Information");
            builder.setMessage("Please choose your answer.")
                .setCancelable(false)
                .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {
                        ad.dismiss();
                    }
                });
            ad = builder.show();
        }else{
            RadioButton radioButton = (RadioButton) findViewById(selected);
            youranswers[index] = radioButton.getText().toString();
            if(pointer < questions_key.size()){
                rdgroup[index+1].setVisibility(View.INVISIBLE);
                questions.setVisibility(View.INVISIBLE);
                pointer++;
                index++;
                quizlay.removeAllViews();
                getquestions();
            }else{
                timer2.cancel();
                for(int l = 0 ; l<=questions_key.size()-1 ; l++){
                    if(!Objects.equals(youranswers[l], correctans[l])){
                        findings = "fail";
                        missed.add(questions_key.get(l));
                    }
                }
            }
        }
    }
});
}

```

```

    }
    if(Objects.equals(findings, "fail")){
        for(int y = 0 ; y<=missed.size()-1 ; y++){
            dref = FirebaseDatabase.getInstance().getReference();

updateitemanalysis(curr_subject,curr_lesson,curr_topic,grpCode,missed.get(y), studkey,
teacherkey);

            }
            repeatquiz("wronganswer");
        }else{
            onPause();
            totalscore = totalscore + scored;
            String curr_topic = globalVariables.getcurr_topic();
            String grpCode = globalVariables.getgrpCode();
            String score = totalscore.toString();
            dref = FirebaseDatabase.getInstance().getReference();
            long timeonRead = globalVariables.getTimespentonRead();
            String totaltimeonread = timeonRead+ "";
            long timeonQuiz = globalVariables.getTimespentonQuiz();
            String totaltimeonQuiz = timeonQuiz + "";
            String studentkey = globalVariables.getstudentKey();
            createAssessment(grpCode, score, curr_topic, teacherkey,
totaltimeonread, totaltimeonQuiz, studentkey);

        }
    }
}

});
}

private void repeatquiz(String method){
    String todo = method;
    scored = scored - 5;
    globalVariables.setscore(scored);
    globalVariables.setonquiz("yes");
    android.app.AlertDialog.Builder builder = new
android.app.AlertDialog.Builder(Quiz.this);
    builder.setTitle("Information");
    if(todo == "wronganswer"){
        todo = "You have answer a question wrong. Read the lesson carefully";
    }else{
        todo = "You have consume your allotted time. \n 5% will be deducted to your
percentage. \n Read the lesson carefully.";
    }
    builder.setMessage(todo)
        .setCancelable(false)
        .setPositiveButton("OK", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                Intent intent = new Intent(Quiz.this,
Read_Lesson_Screen.class);
                Bundle extras = new Bundle();
                extras.putString("lesson",s);
                intent.putExtras(extras);
                ad.dismiss();
                startActivity(intent);
                finish();
            }
        });
    ad = builder.show();
}

private void getquestions(){

```

```

        dref =
        database.getReference().child("subjects").child(teacherkey).child(curr_subject).child(c
        urre_lesson).child(curr_topic).child("quiz").child("questions").child(questions_key.get(
        index));
        dref.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                Map<String, Object> newPost = (Map<String, Object>)
                dataSnapshot.getValue();
                q_type = newPost.get("q_type");
                if(Objects.equals(q_type.toString(), "True/False")){
                    question = newPost.get("q_question");
                    canswer = newPost.get("q_answer");
                    correctans[index] = canswer.toString();
                    questions = new TextView(Quiz.this);
                    questions.setText(index+1 + ". " + question.toString());
                    questions.setTextSize(TypedValue.COMPLEX_UNIT_PX,
                        getResources().getDimension(R.dimen.questions));
                    questions.setTextColor(getResources().getColor(R.color.black));
                    quizlay.addView(questions);
                    quizlay.addView(rdgroup[index+1]);
                    RadioButton choices1 = new RadioButton(Quiz.this);
                    choices1.setText("True");
                    choices1.setTextSize(TypedValue.COMPLEX_UNIT_PX,
                        getResources().getDimension(R.dimen.choices));
                    rdgroup[index+1].addView(choices1);
                    RadioButton choices2 = new RadioButton(Quiz.this);
                    choices2.setText("False");
                    choices2.setTextSize(TypedValue.COMPLEX_UNIT_PX,
                        getResources().getDimension(R.dimen.choices));
                    rdgroup[index+1].addView(choices2);
                }else if(Objects.equals(q_type.toString(), "Multiple Choice")){
                    choices = new ArrayList<String>();
                    question = newPost.get("q_question");
                    canswer = newPost.get("q_answer");
                    correctans[index] = canswer.toString();
                    dref =
                    database.getReference().child("subjects").child(teacherkey).child(curr_subject).child(c
                    urre_lesson).child(curr_topic).child("quiz").child("questions").child(questions_key.get(
                    index));

                    dref.addListenerForSingleValueEvent(new ValueEventListener() {
                        @Override
                        public void onDataChange(DataSnapshot dataSnapshot) {
                            for(DataSnapshot d : dataSnapshot.getChildren()){
                                if(Objects.equals(d.getKey(), "q_answer") ||
                                Objects.equals(d.getKey(), "q_question") || Objects.equals(d.getKey(), "q_type")){

                                    }else{
                                        choices.add(d.getValue().toString());
                                    }
                                }
                            }
                            questions = new TextView(Quiz.this);
                            questions.setText(index+1 + ". " + question.toString());
                            questions.setTextSize(TypedValue.COMPLEX_UNIT_PX,
                                getResources().getDimension(R.dimen.questions));

                            questions.setTextColor(getResources().getColor(R.color.black));
                            quizlay.addView(questions);
                            quizlay.addView(rdgroup[index+1]);

                            for(int k = 0 ; k<=choices.size()-1 ; k++){

```

```

        rdbtn = new RadioButton(Quiz.this);
        rdbtn.setText(choices.get(k));
        rdbtn.setTextSize(TypedValue.COMPLEX_UNIT_PX,
            getResources().getDimension(R.dimen.choices));
        rdgroup[index+1].addView(rdbtn);
    }
}

@Override
public void onCancelled(DatabaseError databaseError) {

}

});

}

@Override
public void onCancelled(DatabaseError databaseError) {

}

});

private void createAssessment(String grpCode, String score, String curr_topic,
String teacherkey, String timeonRead, String timeonQuiz, String studentkey) {
    SimpleDateFormat simpleDateFormat = new SimpleDateFormat("dd-MM-yyyy-hh-mm-ss");
    String date_quiztaken = simpleDateFormat.format(new Date());
    Assessment user = new Assessment(score, timeonRead, timeonQuiz,
date_quiztaken);

    dref.child("assessment").child(teacherkey).child(grpCode).child(studentkey).child(curr_
topic).setValue(user);
    addUserChangeListener(studentkey, teacherkey,grpCode, curr_topic);

}

private void updateitemanalysis(String curr_subject, String curr_lesson, String
curr_topic, String grpCode, String missed_key, String studkey, String teacherkey){
    dref.child("subjects").child(teacherkey).child(curr_subject).child(curr_lesson)

.child(curr_topic).child("quiz").child("item_analysis").child(grpCode).child(missed_key
).child(studkey).setValue(1);

    addupdateanalysislistener(curr_subject,curr_lesson,curr_topic,grpCode,missed_key,studke
y, teacherkey);
}

private void addupdateanalysislistener(String curr_subject, String curr_lesson,
String curr_topic, String grpCode, String missed_key, String studkey, String
teacherkey) {
    dref.child("subjects").child(teacherkey).child(curr_subject).child(curr_lesson)

.child(curr_topic).child("quiz").child("item_analysis").child(grpCode).child(missed_key
).addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            Log.d("Check", "Success");
        }

        @Override
        public void onCancelled(DatabaseError error) {
            // Failed to read value
            Log.d("Check", "Failed", error.toException());
        }
    });
}

```



```

    }
    });
}
private void addUserChangeListener(String studentkey, String teacherkey, String
grpCode,String curr_topic) {

dref.child("assessment").child(teacherkey).child(grpCode).child(studentkey).child(curr_
topic).addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        Log.d("Check", "Success");
        globalVariables.setscore(100);
        globalVariables.setTimespentonRead(0,"yes");
        globalVariables.setTimespentonQuiz(0,"yes");
        android.app.AlertDialog.Builder builder = new
android.app.AlertDialog.Builder(Quiz.this);
        builder.setTitle("Information");
        builder.setMessage("You have finished this topic! ")
            .setCancelable(false)
            .setPositiveButton("OK", new DialogInterface.OnClickListener()
{
                public void onClick(DialogInterface dialog, int id) {
                    Intent intent = new Intent(Quiz.this,
SelectedCourseScreen.class);
                    ad.dismiss();
                    startActivity(intent);
                    finish();
                }
            });
        ad = builder.show();
    }

    @Override
    public void onCancelled(DatabaseError error) {
        // Failed to read value
        Log.d("Check", "Failed", error.toException());
    }
});
}
@Override
protected void onResume() {
    super.onResume();
    startTime = System.currentTimeMillis();
}
@Override
protected void onPause() {
    super.onPause();
    endTime = System.currentTimeMillis();
    spent = endTime - startTime;
    spent = spent / 1000;
    globalVariables.setTimespentonQuiz(spent,"no");
    timer2.cancel();
    if(ad != null){
        ad.dismiss();
    }
}
@Override
protected void onStop() {
    super.onStop();
    ad.dismiss();
}

```

```

    }
    public void onBackPressed() {
        new AlertDialog.Builder(this)
            .setTitle("Information")
            .setMessage("Sorry you can't disregard your quiz.")
            .setPositiveButton(android.R.string.yes, null).create().show();
    }
}
OfflineScreen.java
public class OfflineScreen extends AppCompatActivity {
    LinearLayout.LayoutParams params;
    LinearLayout rl;
    ListView listView;
    TextView nopdf;
    ArrayList<Object> list = new ArrayList<>();
    ArrayAdapter<Object> adapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_offline_screen);
        isInternetOn();
        params = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
            LinearLayout.LayoutParams.WRAP_CONTENT);
        params.setMargins(0,10,0,0);
        rl = (LinearLayout) findViewById(R.id.fragment_photos_layout);
        nopdf = (TextView) findViewById(R.id.txt_nogroup);
        listView = (ListView) findViewById(R.id.listView);
        adapter = new ArrayAdapter<Object>(OfflineScreen.this, R.layout.listView_text2,
list);
        listView.setAdapter(adapter);
        FileCacher cacher = new FileCacher(OfflineScreen.this, "topics");
        if(cacher.hasCache()){
            try {
                String topic = cacher.readCache().toString();
                StringBuilder sb = new StringBuilder(topic);
                sb.deleteCharAt(0);
                sb.deleteCharAt(sb.length()-1);
                list.add(sb.toString());
                adapter.notifyDataSetChanged();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }else{
            nopdf.setVisibility(View.VISIBLE);
        }
        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
                final String pickedTopic =
String.valueOf(parent.getItemAtPosition(position));
                Intent intent = new Intent(OfflineScreen.this,
ReviewLessonsScreen.class);
                Bundle extras = new Bundle();
                extras.putString("pickedTopic",pickedTopic);
                intent.putExtras(extras);
                startActivity(intent);
            }
        });
    }
    public final boolean isInternetOn() {
        // get Connectivity Manager object to check connection

```

```

        ConnectivityManager connec =
        (ConnectivityManager) getSystemService(getBaseContext().CONNECTIVITY_SERVICE);
        // Check for network connections
        if ( connec.getNetworkInfo(0).getState() ==
        android.net.NetworkInfo.State.CONNECTED ||
            connec.getNetworkInfo(0).getState() ==
        android.net.NetworkInfo.State.CONNECTING ||
            connec.getNetworkInfo(1).getState() ==
        android.net.NetworkInfo.State.CONNECTING ||
            connec.getNetworkInfo(1).getState() ==
        android.net.NetworkInfo.State.CONNECTED ) {
            // if connected with internet
            Toast.makeText(this, " Internet Connection Established",
            Toast.LENGTH_LONG).show();
            Intent intent = new Intent(OfflineScreen.this,LoginScreen.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);
            finish();
            return true;
        } else if (
            connec.getNetworkInfo(0).getState() ==
        android.net.NetworkInfo.State.DISCONNECTED ||
            connec.getNetworkInfo(1).getState() ==
        android.net.NetworkInfo.State.DISCONNECTED ) {
            return false;
        }
        return false;
    }
    public void onBackPressed() {
        if(isInternetOn()){

            Intent intent = new Intent(OfflineScreen.this,MainScreen.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            intent.putExtra("EXIT", true);
            startActivity(intent);

        }else{
            new AlertDialog.Builder(this)
                .setTitle("Information")
                .setMessage("Connection Not Yet Established")
                .setPositiveButton(android.R.string.yes, null).create().show();
        }
    }
}

```

ReviewLessonScreen.java

```

public class ReviewLessonsScreen extends AppCompatActivity {
    FirebaseAuth auth;
    String s;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_review_lessons_screen);
        Bundle extras = getIntent().getExtras();
        s = extras.getString("pickedTopic");
        FileCacher<byte[]> cacher = new FileCacher<>(ReviewLessonsScreen.this, s);
        try {
            byte[] bytes = cacher.readCache();
            PDFView pdfView = (PDFView) findViewById(R.id.pdfView);
            pdfView.fromBytes(bytes)
                .enableSwipe(true) // allows to block changing pages using swipe

```

```

        .swipeHorizontal(false)
        .enableDoubletap(true)
        .defaultPage(0)
        .enableAnnotationRendering(false) // render annotations (such as
comments, colors or forms)
        .password(null)
        .scrollHandle(null)
        .load();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

Teacher's Module

addnewclass.js

```

var test = localStorage.loggedin;
if(test != "true"){
    window.location.replace("login.php");
}
if (localStorage.locked == "true") {
    window.location.replace("lock_screen.php");
}
var config = {
    apiKey: "AIzaSyD0CsnZgDBGsFBVrtZH0HRBbVQWk__txe0",
    authDomain: "ezbio-98be1.firebaseio.com",
    databaseURL: "https://ezbio-98be1.firebaseio.com",
    storageBucket: "ezbio-98be1.appspot.com",
    messagingSenderId: "595946963410"
};
firebase.initializeApp(config);
$(document).ready(function () {
    $("#teachername").html(localStorage.teacher_name);
    var dbRef = firebase.database().ref();
    loadSubjects();
    for(var i = 2000 ; i < 2050 ; ++i){
        $("#cmb_fromyear").append("<option>" + i + "</option>");
        $("#cmb_toyear").append("<option>" + i + "</option>");
    }

    if (typeof GetParameterValues("action") !== 'undefined') {
        if(GetParameterValues("action") == 1){
            setTimeout(function() {

                firebase.database().ref("classes/"+Get("code")).once('value').then(function(snapshot){
                    $("#txt_name").val(snapshot.val().grp_name);
                    var dateseparated = (snapshot.val().grp_year).split('-');
                    $("#cmb_fromyear").val(dateseparated[0]);
                    $("#cmb_toyear").val(dateseparated[1]);
                    $("#cmb_curriculum
option[value='"+snapshot.val().grp_subject+"'").attr('selected', 'selected');
                    $("#num_capacity").val(snapshot.val().grp_capacity);
                    $("#txt_grpcode").val(snapshot.key);
                    $("#btn_gen_grpcode").prop('disabled', 'true');
                });
            }, 3000);
        }
    }
    $("#frm_newclass").on("submit", function(){

```

```

if(GetParameterValues("action") == 1){
    var gname = $("#txt_name").val();
    var gsubject = $("#cmb_curriculum").val();
    var gyear = $("#cmb_fromyear").val() + "-" + $("#cmb_toyear").val();
    var gcap = $("#num_capacity").val();
    var gcode = $("#txt_grpcode").val();
    var gteacher = localStorage.teacher;
    var gteacherkey = localStorage.memberkey;
    var dt = new Date();
    var added = dt.toUTCString();
    dbRef.child("classes/" + gcode).update({
        grp_name : gname,
        grp_subject : gsubject,
        grp_year : gyear,
        grp_capacity : gcap,
        grp_teacher: gteacher,
        grp_teacher_key: gteacherkey,
        grp_status: "Active",
        grp_date_created: added
    }).then(function(){
        alert("Class Edited");
        window.location.replace("classes.php");
    });
}else{
    var gname = $("#txt_name").val();
    var gsubject = $("#cmb_curriculum").val();
    var gyear = $("#cmb_fromyear").val() + "-" + $("#cmb_toyear").val();
    var gcap = $("#num_capacity").val();
    var gcode = $("#txt_grpcode").val();
    var gteacher = localStorage.teacher;
    var gteacherkey = localStorage.memberkey;
    var dt = new Date();
    var added = dt.toUTCString();
    dbRef.child("classes/" + gcode).set({
        grp_name : gname,
        grp_subject : gsubject,
        grp_year : gyear,
        grp_capacity : gcap,
        grp_teacher: gteacher,
        grp_teacher_key: gteacherkey,
        grp_status: "Active",
        grp_date_created: added
    }).then(function(){
        alert("Class Added");
        window.location.replace("classes.php");
    });
}
});
$("#btn_gen_grpcode").click(function(){
    var text = "";
    var possible = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
    for( var i=0; i < 6; i++ ){
        text += possible.charAt(Math.floor(Math.random() * possible.length));
    }
    var validate = false;
    firebase.database().ref("classes").once('value', function(snapshot) {
        if (snapshot.hasChild(text)) {
            validate = true;
        }
        else{
            validate = false;
        }
    });
}

```

```

    }
  }).then(function(){
    if(validate == false)
      $("#txt_grpcode").val(text);
  });
});
});
firebase.database().ref("classes").on('child_changed', function(){
  window.location.href = "addnewclass.php";
});
function loadSubjects(){
  firebase.database().ref("subjects/"
  localStorage.memberkey).orderByKey().once('value').then(function(snapshot) {
    snapshot.forEach(function(childSnapshot) {
      $("#cmb_curriculum").append("<option value = '"+childSnapshot.key+"'> " +
      childSnapshot.key + "</option>")
    });
  })
}
function GetParameterValues(param) {
  var url = window.location.href.slice(window.location.href.indexOf('?')
  1).split('&');
  for (var i = 0; i < url.length; i++) {
    var urlparam = url[i].split('=');
    if (urlparam[0] == param) {
      return urlparam[1];
    }
  }
}

```

addnewsbjeect.js

```

var test = localStorage.loggedin;
if(test != "true"){
  window.location.replace("welcomepage.php");
}
if (localStorage.locked == "true") {
  window.location.replace("lock_screen.php");
}
var config = {
  apiKey: "AIzaSyD0CsnZgDBGsFBVrtZH0HRBbVQWk__tXe0",
  authDomain: "ezbio-98be1.firebaseio.com",
  databaseURL: "https://ezbio-98be1.firebaseio.com",
  storageBucket: "ezbio-98be1.appspot.com",
  messagingSenderId: "595946963410"
};
firebase.initializeApp(config);
$(document).ready(function(){
  $("#teachername").html(localStorage.teacher_name);
});
var dbRef = firebase.database().ref();
if (typeof GetParameterValues("action") !== 'undefined') {
  if(Get("action") == 1){
    var subj_decrypted = CryptoJS.AES.decrypt(Get("title"), "Secret
    Passphrase").toString(CryptoJS.enc.Utf8);
    $("#btn_add").html("<i aria-hidden='true' class='fa fa-plus'></i>&nbsp; Edit
    Subject</button>");
    $("#win_title").html("<i aria-hidden='true' class='fa fa-plus'></i>&nbsp; Edit
    Subject");
    setTimeout(function() {

```

```

firebase.database().ref("subjects/"+localStorage.memberkey+"/"+subj_decrypted).once('value').then(function(snapshot){
    $("#txt_name").val(snapshot.key);
    $("#txt_desc").val(snapshot.val().subj_description);
    $("#txt_name").attr('readonly', 'true');
    $("#subj_title").html('You cannot change the title anymore. If you want to change, deletion is advised');
    });
    }, 2000);
}
}
$("#btn_add").on('click', function(){
    if(GetParameterValues("action") == 1){
        var subj_decrypted = CryptoJS.AES.decrypt(GetParameterValues("title"), "Secret Passphrase").toString(CryptoJS.enc.Utf8);
        var conf = confirm('Are you sure to edit the selected subject? The action is irreversible');
        if(conf == true){
            var gteacher = localStorage.memberkey;
            var desc = $("#txt_desc").val();
            firebase.database().ref("subjects/"+gteacher+"/"+subj_decrypted).update({
                subj_description: desc
            }).then(function(){
                alert("Subject successfully edited!");
                setTimeout(function() {window.location.href = "syllabus.php";}, 1000);
            });
        }
    }else{
        var name = $("#txt_name").val();
        var dt = new Date();
        var gadded = dt.toUTCString();
        var gteacher = localStorage.memberkey;
        var desc = $("#txt_desc").val();
        dbRef.child("subjects/" + gteacher + "/" + name).set({
            subj_date_added: gadded,
            subj_description: desc
        }).then(function(){
            alert("Subject successfully added!");
            window.location.replace("syllabus.php");
        });
    }
});
});
function GetParameterValues(param) {
    var url = window.location.href.slice(window.location.href.indexOf('?') + 1).split('&');
    for (var i = 0; i < url.length; i++) {
        var urlparam = url[i].split('=');
        if (urlparam[0] == param) {
            return urlparam[1];
        }
    }
}
}

```

assessment.js

```

var test = localStorage.loggedin;
if (test != "true") {
    window.location.replace("login.php");
}
if (localStorage.locked == "true") {

```

```

        window.location.replace("lock_screen.php");
    }
    var config = {
        apiKey: "AIzaSyD0CsnZgDBGsFBVrtZH0HRBbVQWk__txe0",
        authDomain: "ezbio-98be1.firebaseio.com",
        databaseURL: "https://ezbio-98be1.firebaseio.com",
        storageBucket: "ezbio-98be1.appspot.com",
        messagingSenderId: "595946963410"
    };
    firebase.initializeApp(config);
    var student_keys = new Array();
    var columns = new Array();
    var quizzes = new Array();
    $(document).ready(function() {
        $("#teachername").html(localStorage.teacher_name);
        $('#tbl_assess').dataTable();

        firebase.database().ref('classes').orderByKey().once('value').then(function(snapshot) {
            snapshot.forEach(function(childSnapshot) {
                var section = childSnapshot.val().grp_name;
                var year = childSnapshot.val().grp_year;
                var gteacher = childSnapshot.val().grp_teacher;
                var gteacherclass = childSnapshot.val().grp_subject;
                if (gteacher == localStorage.teacher) {
                    $('#cmb_class').append("<option value = '" + section + "' data-year = '"
+ year + "' data-code = '" + childSnapshot.key + "'> " + section + "(" + year + ")" +
"</option>")
                }
            });
        }).then(function() {
            if (typeof GetParameterValues("sec") !== 'undefined') { //WORKS LIKE THE ISSET
FUNCTION IN PHP
                var decrypted = CryptoJS.AES.decrypt(GetParameterValues("sec"), "Secret
Passphrase").toString(CryptoJS.enc.Utf8);
                $("#cmb_class option[value=\"" + decrypted + "\"]').attr('selected',
'selected');
            }
        }).then(function() {
            cancel();
        });
        $(this).on("change", "#cmb_class", function() {
            cancel();
        });
        if (typeof GetParameterValues("approved") !== 'undefined') {
            if (GetParameterValues("approved") == "true") {
                var dt = new Date();
                var added = dt.toUTCString();
                var oldRef = firebase.database().ref("classes/"
+ GetParameterValues("group") + "/pending_members/" + GetParameterValues("key"));
                var newRef = firebase.database().ref("classes/"
+ GetParameterValues("group") + "/member/" + GetParameterValues("key"));
                newRef.set({
                    date_joined: added
                }).then(function() {
                    oldRef.remove();
                    alert("Approved");
                    window.location.replace("assessment.php?sec="
+ CryptoJS.AES.encrypt($("#cmb_class").val(), "Secret Passphrase"));
                });
            } else {
                alert("Rejected");
            }
        }
    });

```



```

    }
  }
});
function cancel() {
  $('#tbl_assess').DataTable().clear().draw();
  var cansy = $("#cmb_class option:selected").data('code');

  firebase.database().ref('classes/' + cansy + "/member").orderByKey().once('value').then(function(snapshot) {
    snapshot.forEach(function(childSnapshot) {
      firebase.database().ref("members/" +
childSnapshot.key).orderByKey().once('value').then(function(snapshot) {
        var email = snapshot.val().email;
        var firstname = snapshot.val().firstname;
        var lastname = snapshot.val().lastname;
        $('#tbl_assess').DataTable().row.add(["<a href =
'studassessment.php?key=" + childSnapshot.key + "&sectionkey=" + cansy + "'>View
Assessment</a>", firstname, lastname, email]).draw();
      });
    });
  }).then(function(){
    pendingmembers();
  });
};

function pendingmembers() {
  $('#tbl_pending').DataTable().clear().draw();
  var cansy = $("#cmb_class option:selected").data('code');

  firebase.database().ref("classes/" + cansy).child("pending_members").once('value').then(function(snapshot){
    snapshot.forEach(function(childSnapshot){
      firebase.database().ref("members/" +
childSnapshot.key).orderByKey().once('value').then(function(snapshot) {
        var email = snapshot.val().email;
        var firstname = snapshot.val().firstname;
        var lastname = snapshot.val().lastname;
        var approved_code = "<a class = 'btn btn-info' href =
assessment.php?approved=true&key=" + childSnapshot.key + "&group=" + cansy +
">Approve</a>";
        var rejected_code = "<a class = 'btn btn-danger' href =
assessment.php?approved=false&key=" + childSnapshot.key + "&group=" + cansy +
">Reject</a>";
        var check_approve = "<input type = 'checkbox' name = 'chk_values[]' value
= '" + childSnapshot.key + "'>";
        $('#tbl_pending').DataTable().row.add([check_approve,
firstname,
lastname, email, approved_code, rejected_code]).draw();
      });
    });
  }).then(function(){
    loadQuizzes();
  });
}

$("#btn_batchApprove").on("click", function(){
  var checked = new Array();
  $("input[name='chk_values[]']:checked").each(function ()
  {
    checked.push($(this).val());
  });
  checked.forEach(function(vals){
    var dt = new Date();

```

```

        var added = dt.toUTCString();
        var cansy = $("#cmb_class option:selected").data('code');
        var oldRef = firebase.database().ref("classes/" + cansy + "/pending_members/" +
vals);
        var newRef = firebase.database().ref("classes/" + cansy + "/member/" + vals);
        newRef.set({
            date_joined: added
        });
    });
    checked.forEach(function(valy){
        var dt = new Date();
        var added = dt.toUTCString();
        var cansy = $("#cmb_class option:selected").data('code');
        var oldRef = firebase.database().ref("classes/" + cansy + "/pending_members/" +
vals);
        oldRef.remove();
    });
    alert("Approved");
    window.location.replace("assessment.php?sec="
CryptoJS.AES.encrypt($("#cmb_class").val(), "Secret Passphrase"));
});

function GetParameterValues(param) {
    var url = window.location.href.slice(window.location.href.indexOf('?')
1).split('&');
    for (var i = 0; i < url.length; i++) {
        var urlparam = url[i].split('=');
        if (urlparam[0] == param) {
            return urlparam[1];
        }
    }
}

function loadQuizzes(){
    $("#cmb_assessmentfor").empty();
    $("#cmb_assessmentfor").append("<option>None</option>");
    quizzes = [];
    var cansy = $("#cmb_class option:selected").data('code');

    firebase.database().ref("assessment/"+localStorage.memberkey+"/"+cansy).once('value').t
hen(function(snapshot){
        snapshot.forEach(function(childSnapshot){

            firebase.database().ref("assessment/"+localStorage.memberkey+"/"+cansy+"/"+childSnapsho
t.key).once('value').then(function(snappy){
                snappy.forEach(function(childSnappy){
                    if(quizzes.indexOf(childSnappy.key) == -1){
                        quizzes.push(childSnappy.key);
                    }
                });
            });
        });
    });
    quizzes.forEach(function(x){
        $("#cmb_assessmentfor").append("<option>"+x+"</option>");
    });
}

firebase.database().ref("classes").on('child_changed', function(){
    window.location.href = "assessment.php";
});
firebase.database().ref("assessment").on('child_changed', function(){
    window.location.href = "assessment.php";
});

```

```

});

classes.js

var test = localStorage.loggedin;
    if(test != "true"){
        window.location.replace("login.php");
    }
if (localStorage.locked == "true") {
    window.location.replace("lock_screen.php");
}
var config = {
    apiKey: "AIzaSyD0CsnZgDBGsFBVrtZH0HRBbVQWk__tXe0",
    authDomain: "ezbio-98be1.firebaseio.com",
    databaseURL: "https://ezbio-98be1.firebaseio.com",
    storageBucket: "ezbio-98be1.appspot.com",
    messagingSenderId: "595946963410"
};
firebase.initializeApp(config);
$(document).ready(function () {
    $("#teachername").html(localStorage.teacher_name);

    firebase.database().ref('classes').orderByKey().once('value').then(function(snapshot) {
        snapshot.forEach(function(childSnapshot) {
            var code = childSnapshot.key;
            var section = childSnapshot.val().grp_name;
            var year = childSnapshot.val().grp_year;
            var gteacher = childSnapshot.val().grp_teacher;
            var gsubject = childSnapshot.val().grp_subject;
            var gstatus = childSnapshot.val().grp_status;
            var gcap = childSnapshot.val().grp_capacity;
            var gcreated = childSnapshot.val().grp_date_created;
            var hey = "<a class = 'btn btn-info' href = \"assessment.php?sec=" +
                CryptoJS.AES.encrypt(section, "Secret Passphrase") + "\">View Members</a>";
            var currcount = 0;

            firebase.database().ref("classes/"+code+"/member").once('value').then(function(snapshot)
            ){
                currcount = snapshot.numChildren();
            }).then(function(){
                var cap = currcount + "/" + gcap;
                if(gteacher == localStorage.teacher && gstatus == "Active"){
                    $('#tbl_class').DataTable().row.add( [code, section, gsubject, year,
                    cap, gcreated, gstatus, hey, "<a class = 'btn btn-warning' href =
                    'addnewclass.php?action=1&code="+code+"'>Edit</a> <a class = 'btn btn-danger' onclick =
                    'deactivate(\""+code+"\");'>Deactivate</a> <a class = 'btn btn-danger' onclick =
                    'deleteClass(\""+code+"\");'>Delete</a>"] ).draw();
                }else if(gteacher == localStorage.teacher && gstatus == "Inactive"){
                    $('#tbl_class').DataTable().row.add( [code, section, gsubject, year,
                    cap, gcreated, gstatus, hey, "<a class = 'btn btn-warning' href =
                    'addnewclass.php?action=1&code="+code+"'>Edit</a><a class = 'btn btn-info' onclick =
                    'activate(\""+code+"\");'>Activate</a> <a class = 'btn btn-danger' onclick =
                    'deleteClass(\""+code+"\");'>Delete</a>"] ).draw();
                }
            });
        });
    });
});
function deactivate(code){
    var teacherkey = localStorage.memberkey;

```

```

        firebase.database().ref("classes/"+code).update({grp_status: "Inactive"});
        window.location.href = "classes.php";
    }
    function activate(code){
        var teacherkey = localStorage.memberkey;
        firebase.database().ref("classes/"+code).update({grp_status: "Active"});
        window.location.href = "classes.php";
    }
    function deleteClass(code){
        var conf = confirm("Delete this class?");
        if (conf == true){
            firebase.database().ref("classes/"+code).remove();
            window.location.href = "classes.php";
        }
    }
}

```

createquiz.js

```

var test = localStorage.loggedin;
if (test != "true") {
    window.location.replace("welcomepage.php");
}
if (localStorage.locked == "true") {
    window.location.replace("lock_screen.php");
}
var selected_chapter = "";
var config = {
    apiKey: "AIzaSyD0CsnZgDBGsFBVrtZH0HRBbVQWk__txe0",
    authDomain: "ezbio-98be1.firebaseio.com",
    databaseURL: "https://ezbio-98be1.firebaseio.com",
    storageBucket: "ezbio-98be1.appspot.com",
    messagingSenderId: "595946963410"
};
firebase.initializeApp(config);
var question_forms = new Array();
var panel_titles = new Array();
var questionnum = 0;
var trigger_fileadd = false;
var texthere;
$(document).ready(function() {
    InitialLoad();
    $("#teachername").html(localStorage.teacher_name);
    var today = new Date().toISOString().split('.')[0]+"Z";
    $("#dt_expiration").attr('min', today);
    $(this).on("click", "#btn_addquestion", function(e){
        e.preventDefault();
        questionnum++;
        $("#questions_container").append("<div class = 'row' id = 'cont_"+questionnum+"'" +
        "<div class = 'col-md-12'" +
        "<div class='panel-group'" +
        "<div class='panel panel-info'" +
        "<div class='panel-heading'" +
        "<h4 class='panel-title'" +
        "<a data-toggle='collapse' href='#q" + questionnum + "' id = 'titles_"+questionnum+"'" +
        ">Question " + questionnum + "</a> </h4> </div> <div id='q" + questionnum + "' class='panel-collapse collapse'" +
        "> <div class='panel-body'" +
        "><form id = \"frm_q" + questionnum + "\" name = \"frm_q" + questionnum + "\" class = 'form-horizontal'" +
        "><div class = 'form-group'" +
        "><div class = 'col-md-3 pull-right'" +
        "><button class = 'btn btn-danger' onclick = \"deleteQuestion(\" + questionnum + "\")\"><i class='fa fa-ban'" +
        "></i>&nbsp; Delete Question</button></div></div><div class = 'form-group'" +
        "><div class = 'col-md-12'" +
        ">Question<textarea class = 'form-control' id='q" + questionnum + "_question' name='q" + questionnum + "_question' style = 'resize:none;'></textarea></div></div><div class = 'form-group'" +
        "><div class = 'col-md-6'" +
        ">Type <select class = 'form-control' id='q" +

```

```

questionnum + "_type" name="q" + questionnum + "_type"><option>Multiple
Choice</option><option>True/False</option></select></div></div><div id = 'canc_' +
questionnum + "'></div></div></form></div> </div> </div> </div></div>");
    question_forms.push("#frm_q" + questionnum);
    panel_titles.push("#titles_" + questionnum);
    $("body").append("<script>$(\"#q"+questionnum+"_type\").on('change',
function(){ \n\t$(\"#canc_" + questionnum + "\").html(\" \");\n\tquestionType(\"frm_q" +
questionnum + "\", $(\"#q"+questionnum+"_type\").val(), \"q" + questionnum + "\", \"canc_"
+
questionnum
+ "\");
\n});$(\"#q"+questionnum+"_type\").val('\"+$(\"#cmb_defaultType\").val()+\"');</script>");
    questionType("frm_q" + questionnum, $(\"#q"+questionnum+"_type\").val(), "q" +
questionnum, "canc_" + questionnum);
    });
    $(this).on("submit", "#cancy", function(e){
        e.preventDefault();
        var dt = new Date();
        var added = dt.toUTCString();
        var qtype = $(\"#cmb_quizType\").val();
        var tlimit = $(\"#num_quizTimeLimit\").val();
        var items = $(\"#num_quizItems\").val();
        var rand = "no";
        var expiration = new Date($(\"#dt_expiration\").val()).toUTCString();
        var started = "no";
        if($(\"#chk_randomize\").is(':checked')){
            rand = "yes";
        }

        if(trigger_fileadd == true){
            var batchie = JSON.parse(texthere.replace(/\n/g, ""));
            if(batchie.length < items){
                alert('The number of question is less than the indicated number of items.
The questions < number of items. Add more question/s.')
            }else if(batchie.length > items){
                alert('The number of question is greater than the indicated number of
items. The questions > number of items. Remove question/s.')
            }else{
                if(qtype == "aq"){
                    var subj = CryptoJS.AES.decrypt(GetParameterValues("sj"), "Secret
Passphrase").toString(CryptoJS.enc.Utf8);
                    var chap = CryptoJS.AES.decrypt(GetParameterValues("chap"), "Secret
Passphrase").toString(CryptoJS.enc.Utf8);
                    var title = CryptoJS.AES.decrypt(GetParameterValues("title"),
"Secret Passphrase").toString(CryptoJS.enc.Utf8);
                    firebase.database().ref("subjects/" + localStorage.memberkey + "/" +
subj + "/" + chap + "/" + title + "/quiz/").set({
                        quiz_date_added: added,
                        quiz_type : qtype,
                        quiz_timelimit : tlimit,
                        quiz_numItems : items,
                        quiz_randomize: rand,
                        quiz_expiration: expiration,
                        quiz_started: started
                    }).then(function(){
                        batchie.forEach(function(x){
                            firebase.database().ref("subjects/"
+
localStorage.memberkey + "/" + subj + "/" + chap + "/" + title + "/quiz/questions/").push(
x
                        );
                    });
                });
            }).then(function(){
                window.alert("Quiz successfully created");
            });
        }
    });

```

```

        window.location.replace("quiz.php");
    });
    }else if(qtype == "ct"){
        var subj = CryptoJS.AES.decrypt(GetParameterValues("subj"), "Secret
Passphrase").toString(CryptoJS.enc.Utf8);
        var title = CryptoJS.AES.decrypt(GetParameterValues("title"),
"Secret Passphrase").toString(CryptoJS.enc.Utf8);
        firebase.database().ref("subjects/" + localStorage.memberkey + "/" +
subj + "/" + title + "/quiz/" ).set({
            quiz_date_added: added,
            quiz_type : qtype,
            quiz_timelimit : tlimit,
            quiz_randomize: rand,
            quiz_numItems: items,
            quiz_expiration: expiration,
            quiz_started: started
        }).then(function(){
            batchie.forEach(function(x){
                firebase.database().ref("subjects/" +
localStorage.memberkey + "/" + subj + "/" + title + "/quiz/questions/") +
                .push(
                    x
                );
            });
        }).then(function(){
            window.alert("Chapter Test successfully created");
            window.location.replace("quiz.php");
        });
    }
}
}else{
    if(question_forms.length < items){
        alert('The number of question is less than the indicated number of items.
The questions < number of items. Add more question/s.')
    }else if(question_forms.length > items){
        alert('The number of question is greater than the indicated number of
items. The questions > number of items. Remove question/s.')
    }else{
        if(qtype == "aq"){
            var subj = CryptoJS.AES.decrypt(GetParameterValues("sj"), "Secret
Passphrase").toString(CryptoJS.enc.Utf8);
            var chap = CryptoJS.AES.decrypt(GetParameterValues("chap"), "Secret
Passphrase").toString(CryptoJS.enc.Utf8);
            var title = CryptoJS.AES.decrypt(GetParameterValues("title"),
"Secret Passphrase").toString(CryptoJS.enc.Utf8);
            firebase.database().ref("subjects/" + localStorage.memberkey + "/" +
subj + "/" + chap + "/" + title + "/quiz/" ).set({
                quiz_date_added: added,
                quiz_type : qtype,
                quiz_timelimit : tlimit,
                quiz_numItems : items,
                quiz_randomize: rand,
                quiz_expiration: expiration,
                quiz_started: started
            }).then(function(){
                for (var i = 0 ; i < question_forms.length ; i++){
                    var hey =
objectifyForm($(question_forms[i]).serializeArray());
                    firebase.database().ref("subjects/" +
localStorage.memberkey + "/" + subj + "/" + chap + "/" + title + "/quiz/questions/") +
                    .push(
                        hey
                    );
                }
            });
        }
    }
}
}

```

```

    }
    }).then(function(){
        window.alert("Quiz successfully created");
        window.location.replace("quiz.php");
    });
    }else if(qtype == "ct"){
        var subj = CryptoJS.AES.decrypt(GetParameterValues("subj"), "Secret
Passphrase").toString(CryptoJS.enc.Utf8);
        var title = CryptoJS.AES.decrypt(GetParameterValues("title"),
"Secret Passphrase").toString(CryptoJS.enc.Utf8);
        firebase.database().ref("subjects/" + localStorage.memberkey + "/" +
subj + "/" + title + "/quiz/").set({
            quiz_date_added: added,
            quiz_type : qtype,
            quiz_timelimit : tlimit,
            quiz_randomize: rand,
            quiz_numItems: items,
            quiz_expiration: expiration,
            quiz_started: started
        }).then(function(){
            for (var i = 0 ; i < question_forms.length ; i++){
                var hey = objectifyForm($(question_forms[i]).serializeArray());
                firebase.database().ref("subjects/" + localStorage.memberkey +
"/" + subj + "/" + title + "/quiz/questions/").push(
                    hey
                );
            }
        }).then(function(){
            window.alert("Chapter Test successfully created");
            window.location.replace("quiz.php");
        });
    }
    }
    }
    });
    $("[type='number']").keypress(function (evt) {
        evt.preventDefault();
    });
    $("#num_quizItems").trigger( "change" );
    $( "#num_quizItems[title]" ).qtip();
    $("#btn_addquestionfile").click(function(){
        question_forms = [];
        panel_titles = [];
        questionnum = 0;
        $("#questions_container").empty();
        $("#uploadmodal").modal('show');
    });
});
$("#num_quizItems").on("change", function(e){
    question_forms = [];
    panel_titles = [];
    questionnum = 0;
    $("#questions_container").empty();
    var numm = $("#num_quizItems").val();
    for(var i = 0 ; i < numm ; ++i){
        $( "#btn_addquestion" ).trigger( "click" );
    }
});
$("#file_questions").on('change', function(e){
    var file = $(this).prop('files')[0];
    var textType = /text.*/;

```

```

        if (file.type.match(textType)) {
            var reader = new FileReader();

            reader.onload = function(e) {
                $("#show_format").text(reader.result);
                textthere = reader.result;
            }
            reader.readAsText(file);
        } else {
            $("#show_format").text("File not supported!");
        }
    });
    $("#btn_addqfile").on('click', function(e){
        $("#questions_container").empty();
        $("#questions_container").append("<pre>" + textthere + "</pre>");
        trigger_fileadd = true;
    });
    function GetParameterValues(param) {
        var url = window.location.href.slice(window.location.href.indexOf('?') +
1).split('&');
        for (var i = 0; i < url.length; i++) {
            var urlparam = url[i].split('=');
            if (urlparam[0] == param) {
                return urlparam[1];
            }
        }
    }
    var choiceCount = 0;
    function InitialLoad(){
        var quizTitle_decrypted = CryptoJS.AES.decrypt(GetParameterValues("title"), "Secret
Passphrase").toString(CryptoJS.enc.Utf8);
        $("#txt_quizTitle").val(quizTitle_decrypted);
        $("#cmb_quizType").val(GetParameterValues("type"));
    }
    function objectifyForm(formArray) { //serialize data function

        var returnArray = {};
        for (var i = 0; i < formArray.length; i++){
            if((formArray[i]['name']).match(/_type/))
                formArray[i]['name'] = "q_type";
            if((formArray[i]['name']).match(/_answer/))
                formArray[i]['name'] = "q_answer";
            if((formArray[i]['name']).match(/_question/))
                formArray[i]['name'] = "q_question";
            returnArray[formArray[i]['name']] = formArray[i]['value'];
        }
        return returnArray;
    }
    function questionType(formid, type, questionIndex, cancelIndex){
        if(type == "Multiple Choice"){
            $("#"+cancelIndex).append("<div class = 'form-group has-success'><div class = 'col-
md-3'>Right Answer<input class = 'form-control' type = 'text' id = '\" + questionIndex +
'_answer' name = '\" + questionIndex + '_answer' readonly></div></div><div class = 'form-
group'><div class = 'col-md-3'><button type = 'button' class = 'btn btn-info' id =
\""+questionIndex+"_addchoice\" onclick = \"addChoice(\" + formid + "\",\" + type +
\"', '\" + questionIndex + "\", '\" + cancelIndex + \"')\" required>+Add
Choice</button></div></div></div>");
            addChoice(formid, type, questionIndex, cancelIndex);
            addChoice(formid, type, questionIndex, cancelIndex);
        } else if(type == "True/False"){

```



```

        $("#"+cancIndex).append("<div class = 'row'><div class = 'col-md-3 has-  

success'>Answer<select class = 'form-control' id = '" + questionIndex + "_answer' name =  

'" + questionIndex +  

"_answer'><option>True</option><option>False</option></select></div></div>");  

    }  

}  

function addChoice(formid, type, questionIndex, cancIndex){  

    choiceCount++;  

    $("#"+ cancIndex).append("<div class = 'form-group' id = 'cont_" + questionIndex +  

"_" + choiceCount + "'><div class = 'col-md-6'><input class = 'form-control' type =  

'text' id = '" + questionIndex + "_" + choiceCount + "' name = '" + questionIndex + "_" +  

choiceCount + "'></div><div class = 'col-md-3'><button type = 'button' id = 'right_" +  

questionIndex + "_" + choiceCount + "' class = 'btn btn-info' onclick = \"right_ans(\""+  

questionIndex + "\", \"\" + choiceCount + "\");\">This is the right  

answer</button></div><div class = 'col-md-3'><button type = 'button' id = 'right_" +  

questionIndex + "_" + choiceCount + "' class = 'btn btn-danger' onclick =  

\"deleteChoice(\""+ questionIndex + "\", \"\" + choiceCount + "\");\">Delete this  

choice</button></div></div>");  

}  

function deleteQuestion(qNum){  

    $("#cont_"+qNum).remove();  

    question_forms.splice($.inArray("#frm_q" + qNum, question_forms),1);  

    panel_titles.splice($.inArray("#titles_" + qNum, panel_titles),1);  

    if (panel_titles.length == 0){  

        questionnum = 0;  

    }else{  

        for(var k = 0 ; k < panel_titles.length ; k++){  

            var cans = k+1;  

            $(panel_titles[k]).html("Question " + cans);  

        }  

    }  

}  

function deleteChoice(qI,cc){  

    $("#cont_"+qI+"_"+cc).remove();  

}  

function right_ans(qI, cc){  

    var ans = $("#"+qI+"_"+cc).val();  

    $("#"+qI+"_answer").val(ans);  

}  


```

itemanalysis.js

```

var config = {  

    apiKey: "AIzaSyD0CsnZgDBGSFBVrtZH0HRBbVQWk__txe0",  

    authDomain: "ezbio-98be1.firebaseio.com",  

    databaseURL: "https://ezbio-98be1.firebaseio.com",  

    storageBucket: "ezbio-98be1.appspot.com",  

    messagingSenderId: "595946963410"  

};  

firebase.initializeApp(config);  

var test = localStorage.loggedin;  

if(test != "true"){  

    window.location.replace("welcomepage.php");  

}  

if (localStorage.locked == "true") {  

    window.location.replace("lock_screen.php");  

}  

var member_count = 0;  

$(document).ready(function () {  

    countMembers();  


```

```

        itemAnalysis();
        validateCount();
    });
    function validateCount(){
        var loc = CryptoJS.AES.decrypt(GetParameterValues("subj"), "Secret
        Passphrase").toString(CryptoJS.enc.Utf8);
        var counted_questions;
        var numitems;
        setTimeout(function(){

        firebase.database().ref(loc+"/quiz/questions").once('value').then(function(snapshot){
            counted_questions = snapshot.numChildren();
        });
        }, 2000);
        setTimeout(function(){
            firebase.database().ref(loc+"/quiz").once('value').then(function(snapshot){
                numitems = snapshot.val().quiz_numItems;
                if(numitems != counted_questions){
                    $("#warning").html("Warning: The number of questions in the quiz is not enough
                    for the expected number of items.");
                }
            });
        }, 2000);
    }
    function countMembers(){
        var gkeys = new Array();
        var s = CryptoJS.AES.decrypt(GetParameterValues("s"), "Secret
        Passphrase").toString(CryptoJS.enc.Utf8)

        firebase.database().ref("classes").orderByKey().once('value').then(function(snapshot){
            snapshot.forEach(function(childSnapshot){
                var gteacherkey = childSnapshot.val().grp_teacher_key;
                var gsubject = childSnapshot.val().grp_subject;
                if(gteacherkey == localStorage.memberkey && gsubject == s){
                    gkeys.push(childSnapshot.key);
                }
            });
        }).then(function(){
            gkeys.forEach(function(canc){
                firebase.database().ref("classes/"+canc
                +"/").child("member").once('value').then(function(snapshot){
                    member_count = member_count + parseInt(snapshot.numChildren());
                });
            });
        });
    }
    function itemAnalysis(){
        var questions_array = new Array();
        var difficulty = new Array();
        var analysis = new Array();
        var keys = new Array();
        var loc = CryptoJS.AES.decrypt(GetParameterValues("subj"), "Secret
        Passphrase").toString(CryptoJS.enc.Utf8);
        firebase.database().ref(loc
        +"/quiz/questions").orderByKey().once('value').then(function(snapshot) {
            snapshot.forEach(function(childSnapshot){
                if(childSnapshot.key != "quiz_numItems"){
                    questions_array.push(childSnapshot.val().q_question);
                    keys.push(childSnapshot.key);
                }
            });
        });
    }

```

```

    }).then(function(){
        firebase.database().ref(loc
"/quiz/item_analysis").orderByKey().once('value').then(function(snapshot) {
            snapshot.forEach(function(childSnapshot){
                childSnapshot.forEach(function(grandchildSnapshot){
                    difficulty.push((grandchildSnapshot.numChildren() / member_count)*100);
                });
            });
        }).then(function(){
            for (var i = 0 ; i < questions_array.length ; ++i){
                parseFloat(difficulty[i]);
                analysis[questions_array[i]] = difficulty[i];
                var display;
                if(difficulty[i] >= 0 && difficulty[i] <= 33.32){
                    display = "<td><font color = 'green'>" + difficulty[i] + "%</font></td>";
                } else if(difficulty[i] >= 33.33 && difficulty[i] <= 66.65){
                    display = "<td><font color = 'orange'>" + difficulty[i] + "%</font></td>";
                } else if(difficulty[i] >= 66.66 && difficulty[i] <= 100.00){
                    display = "<td><font color = 'red'>" + difficulty[i] + "%</font></td>";
                } else{
                    display = "<td>Percentage of difficulty cannot be assessed.</td>";
                }
                $("#questions_container").append("<tr><td>" + questions_array[i] +
"</td>" + display + "</tr>");
            }
        }).then(function(){
            $("#exp_num").append(member_count);
            $("#tbl_analysis").DataTable();
        });
    });
}

function GetParameterValues(param) {
    var url = window.location.href.slice(window.location.href.indexOf('?')
1).split('&');
    for (var i = 0; i < url.length; i++) {
        var urlparam = url[i].split('=');
        if (urlparam[0] == param) {
            return urlparam[1];
        }
    }
}

```

js-login/index.js

```

var test = localStorage.loggedin;
if(test == "true"){
    window.location.replace("index.php");
}
var config = {
    apiKey: "AIzaSyD0CsnZgDBGSFBVrtZH0HRBbVQWk__txe0",
    authDomain: "ezbio-98be1.firebaseio.com",
    databaseURL: "https://ezbio-98be1.firebaseio.com",
    storageBucket: "ezbio-98be1.appspot.com",
    messagingSenderId: "595946963410"
};
firebase.initializeApp(config);
$('.form').find('input, textarea').on('keyup blur focus', function (e) {

    var $this = $(this),
        label = $this.prev('label');

```

```

        if (e.type === 'keyup') {
            if ($this.val() === '') {
                label.removeClass('active highlight');
            } else {
                label.addClass('active highlight');
            }
        } else if (e.type === 'blur') {
            if( $this.val() === '' ) {
                label.removeClass('active highlight');
            } else {
                label.removeClass('highlight');
            }
        } else if (e.type === 'focus') {

            if( $this.val() === '' ) {
                label.removeClass('highlight');
            }
            else if( $this.val() !== '' ) {
                label.addClass('highlight');
            }
        }
    });

    $('.tab a').on('click', function (e) {

        e.preventDefault();

        $(this).parent().addClass('active');
        $(this).parent().siblings().removeClass('active');

        target = $(this).attr('href');

        $('.tab-content > div').not(target).hide();

        $(target).fadeIn(600);

    });
    jQuery.validator.defaults({
        debug: true,
        success: "valid"
    });
    $("form").submit(function(e){
        e.preventDefault();
    });
    $("#frm_login").on("submit", function(){
        var xmlhttp = new XMLHttpRequest();
        var email = $("#txt_email").val();
        var password = $("#txt_pass").val();
        firebase.auth().signInWithEmailAndPassword(email, password).then(function(){
            var user = firebase.auth().currentUser;
            if(user.emailVerified){

                firebase.database().ref('members').orderByKey().once('value').then(function(snapshot) {
                    snapshot.forEach(function(childSnapshot) {
                        var search_type = childSnapshot.val().type;
                        var search_email = childSnapshot.val().email;
                        if(search_email == email){
                            if(search_type == "Teacher"){
                                localStorage.loggedin = "true";
                                localStorage.teacher = email;
                            }
                        }
                    });
                });
            }
        });
    });

```

```

        localStorage.memberkey = childSnapshot.key;
        localStorage.teacher_name = childSnapshot.val().firstname + " " +
childSnapshot.val().lastname;
        window.location.replace("index.php");
    }else{
        alert("You are not a teacher!");
        window.location.replace("login.php");
    }
    });
});
}else{
    alert("Email not yet confirmed!");
}
}).catch(function(error) {
    var errorCode = error.code;
    var errorMessage = error.message;
    alert(errorCode + ": " + errorMessage);
    window.location.replace("login.php");
});
});

```

logout.php

```

var test = localStorage.loggedin;
if (test != "true") {
    window.location.replace("welcomepage.php");
}
var config = {
    apiKey: "AIzaSyD0CsnZgDBGSFBVrtZH0HRBbVQWk__txe0",
    authDomain: "ezbio-98be1.firebaseio.com",
    databaseURL: "https://ezbio-98be1.firebaseio.com",
    storageBucket: "ezbio-98be1.appspot.com",
    messagingSenderId: "595946963410"
};
firebase.initializeApp(config);
$(document).ready(function(){
    $("#teachername").html(localStorage.teacher_name);
});
function logout() {
    firebase.auth().signOut().then(function() {
        localStorage.clear();
        window.location.replace("welcomepage.php");
    }, function(error) {
        alert('Sign Out Error', error);
    });
}

```

quizmonitoring.js

```

var test = localStorage.loggedin;
if (test != "true") {
    window.location.replace("welcomepage.php");
}
if (localStorage.locked == "true") {
    window.location.replace("lock_screen.php");
}
var config = {
    apiKey: "AIzaSyD0CsnZgDBGSFBVrtZH0HRBbVQWk__txe0",
    authDomain: "ezbio-98be1.firebaseio.com",

```



```

        var qadded = new
Date(childSnappy.val().quiz_date_added).toLocaleString();
        $("#tbl_customquizzes").append("<tr><td>" + qtitle +
"</td><td>Subject</td><td>" + qsubject + "</td><td>" + qadded + "</td></tr>");
    });
    });
    });
    });
    });
}

function loadSubject() {
    var selected_subject = $('#cmb_subjects').val();
    var chap_num = 0;
    firebase.database().ref("subjects/" + localStorage.memberkey + "/" +
selected_subject).orderByChild("chap_date_added").once('value').then(function(snapshot)
{
    if (snapshot.numChildren() <= 2) //1 because of the subject added timestamp
        $("#container_subjects").append("<h1>No quizzes. Create your course content
first.</h1><br>");
    snapshot.forEach(function(childSnapshot) {
        if (childSnapshot.key != "subj_date_added" && childSnapshot.key !=
"subj_description" && childSnapshot.key != "custom_quizzes") {
            chap_num++;
            var str = "subjects/" + localStorage.memberkey + "/" + selected_subject
+ "/" + childSnapshot.key;
            var subj = CryptoJS.AES.encrypt(str, "Secret Passphrase");
            var s = CryptoJS.AES.encrypt(selected_subject, "Secret Passphrase");
            var toggle = childSnapshot.key.replace(/\s+/g, ''); //used for names of
div
            var panel_title = childSnapshot.key;
            var isExpired1 = false;
            var checkStarted1 = childSnapshot.child("quiz/quiz_started").val();
            var datenow1 = new Date().getTime();
            if
Date(childSnapshot.child("quiz/quiz_expiration").val()).getTime() < datenow1) { (new
            isExpired1 = true;
        }
        if (childSnapshot.child("quiz").exists()) {
            var expirrr = new
Date(childSnapshot.child("quiz/quiz_expiration").val()).toLocaleString();
            if (isExpired1 == true) {
                $("#container_subjects").append("<div class='panel-group'> <div
class='panel panel-info'> <div class='panel-heading'> <h4 class='panel-title'> <a data-
toggle='collapse' href='#" + toggle + "'>" + chap_num + ". " + childSnapshot.key + "</a>
</h4> </div> <div id='" + toggle + "' class='panel-collapse collapse'><ul class='list-
group' id = list_" + toggle + "'> </ul> <ul class = 'list-group'><li class = 'list-group-
item'><div class = 'row'><div class = 'col-sm-6'><p class = 'text-success'>Quiz Already
Created for this chapter! Available until: " + expirrr + "</p></div><div class = 'col-
sm-3'><a class = 'btn btn-primary' href = \"itemanalysis.php?subj=" + subj + "&s=" + s +
\">Item Analysis</a></div><div class = 'col-sm-3'><button class = 'btn btn-warning'
onclick = 'extend(\"" + subj + "\");>Quiz expired. Extend?</button>&nbsp;<a class = 'btn
btn-danger' onclick = 'deleteQuiz(\"" + subj + "\"');>Delete
quiz</a></div></div></li></ul> </div> </div> </div>");
            } else {
                if (checkStarted1 == "no") {
                    $("#container_subjects").append("<div class='panel-group'>
<div class='panel panel-info'> <div class='panel-heading'> <h4 class='panel-title'> <a
data-toggle='collapse' href='#" + toggle + "'>" + chap_num + ". " + childSnapshot.key +
"</a> </h4> </div> <div id='" + toggle + "' class='panel-collapse collapse'><ul

```

```

class='list-group' id = list_" + toggle + "> </ul> <ul class = 'list-group'><li class =
'list-group-item'><div class = 'row'><div class = 'col-sm-6'><p class = 'text-
success'>Quiz Already Created for this chapter! Available until: " + expirrr +
"</p></div><div class = 'col-sm-3'><a class = 'btn btn-primary' href =
\"itemanalysis.php?subj=\" + subj + "&s=\" + s + "\">Item Analysis</a></div><div class =
'col-sm-3'><button class = 'btn btn-warning' onclick = 'startQuiz(\"" + subj +
"\");'>Start Quiz</button>&nbsp;<a class = 'btn btn-danger' onclick = 'deleteQuiz(\"" +
subj + "\");'>Delete quiz</a></div></li></ul> </div> </div> </div>");
    } else {
        $("#container_subjects").append("<div class='panel-group'>
<div class='panel panel-info'> <div class='panel-heading'> <h4 class='panel-title'> <a
data-toggle='collapse' href='#" + toggle + "'>" + chap_num + ". " + childSnapshot.key +
"</a> </h4> </div> <div id='" + toggle + "' class='panel-collapse collapse'><ul
class='list-group' id = list_" + toggle + "> </ul> <ul class = 'list-group'><li class =
'list-group-item'><div class = 'row'><div class = 'col-sm-6'><p class = 'text-
success'>Quiz Already Created for this chapter! Available until: " + expirrr +
"</p></div><div class = 'col-sm-3'><a class = 'btn btn-primary' href =
\"itemanalysis.php?subj=\" + subj + "&s=\" + s + "\">Item Analysis</a></div><div class =
'col-sm-3'><button class = 'btn btn-warning' onclick = 'endQuiz(\"" + subj + "\");'>End
Quiz</button>&nbsp;<a class = 'btn btn-danger' onclick = 'deleteQuiz(\"" + subj +
"\");'>Delete quiz</a></div></li></ul> </div> </div> </div>");
    }
    }
    } else {
        $("#container_subjects").append("<div class='panel-group'> <div
class='panel panel-info'> <div class='panel-heading'> <h4 class='panel-title'> <a
data-toggle='collapse' href='#" + toggle + "'>" + chap_num + ". " + childSnapshot.key + "</a>
</h4> </div> <div id='" + toggle + "' class='panel-collapse collapse'> <ul class='list-
group' id = list_" + toggle + "> </ul> <ul class = 'list-group'><li class = 'list-group-
item'><a class = 'btn btn-info' href = 'createquiz.php?title=" +
CryptoJS.AES.encrypt(childSnapshot.key, "Secret Passphrase") + "&type=ct&subj=" +
CryptoJS.AES.encrypt(selected_subject, "Secret Passphrase") + "'> + Create Quiz For This
Chapter</a></li></ul> </div> </div> </div>");
    }
    firebase.database().ref("subjects/" + localStorage.memberkey + "/" +
selected_subject
+
"/"
+
childSnapshot.key).orderByChild("lesson_date_added").once('value').then(function(snapsh
ot) {
        var lesson_num = 0;
        snapshot.forEach(function(childSnapshot) {
            if (childSnapshot.key != "chap_date_added" && childSnapshot.key
!= "quiz") {
                lesson_num++;
                var hey = CryptoJS.AES.encrypt(childSnapshot.key, "Secret
Passphrase");
                var str = "subjects/" + localStorage.memberkey + "/" +
selected_subject + "/" + panel_title + "/" + childSnapshot.key;
                var subj = CryptoJS.AES.encrypt(str, "Secret Passphrase");
                var s = CryptoJS.AES.encrypt(selected_subject, "Secret
Passphrase");
                if (childSnapshot.child("quiz").exists()) {
                    var checkStarted =
childSnapshot.child("quiz/quiz_started").val();
                    var isExpired = false;;
                    var datenow = new Date().getTime();
                    var exppp =
Date(childSnapshot.child("quiz/quiz_expiration").val()).getTime();
                    if (exppp < datenow) {
                        isExpired = true;
                    }
                    if (isExpired == true) {

```



```

    $("#list_" + toggle).append("<li class='list-group-item'><div class = 'row'><div class = 'col-sm-3'>" + chap_num + "." + lesson_num + " " + childSnapshot.key + "</div><div class = 'col-sm-3'>Quiz Available until: " + new Date(childSnapshot.child("quiz/quiz_expiration").val()).toLocaleString() + "</div><div class = 'col-sm-3'><a class = 'btn btn-primary' href = \"itemanalysis.php?subj=\" + subj + \"&s=\" + s + \"\">Item Analysis</a></div><div class = 'col-sm-3'><button class = 'btn btn-warning' onclick = 'extend(\"\" + subj + \"\")';>Quiz expired. Extend?</button>&nbsp;<a class = 'btn btn-danger' onclick = 'deleteQuiz(\"\" + subj + \"\")';>Delete quiz</a></div></div></li>");
        } else {
            if (checkStarted == "no") {
                $("#list_" + toggle).append("<li class='list-group-item'><div class = 'row'><div class = 'col-sm-3'>" + chap_num + "." + lesson_num + " " + childSnapshot.key + "</div><div class = 'col-sm-3'>Quiz Available until: " + new Date(childSnapshot.child("quiz/quiz_expiration").val()).toLocaleString() + "</div><div class = 'col-sm-3'><a class = 'btn btn-primary' href = \"itemanalysis.php?subj=\" + subj + \"&s=\" + s + \"\">Item Analysis</a></div><div class = 'col-sm-3'><button class = 'btn btn-success' onclick = 'startQuiz(\"\" + subj + \"\")';>Start Quiz</button>&nbsp;<a class = 'btn btn-danger' onclick = 'deleteQuiz(\"\" + subj + \"\")';>Delete quiz</a></div></div></li>");
            } else {
                $("#list_" + toggle).append("<li class='list-group-item'><div class = 'row'><div class = 'col-sm-3'>" + chap_num + "." + lesson_num + " " + childSnapshot.key + "</div><div class = 'col-sm-3'>Quiz Available until: " + new Date(childSnapshot.child("quiz/quiz_expiration").val()).toLocaleString() + "</div><div class = 'col-sm-3'><a class = 'btn btn-primary' href = \"itemanalysis.php?subj=\" + subj + \"&s=\" + s + \"\">Item Analysis</a></div><div class = 'col-sm-3'><button class = 'btn btn-warning' onclick = 'endQuiz(\"\" + subj + \"\")';>End Quiz</button>&nbsp;<a class = 'btn btn-danger' onclick = 'deleteQuiz(\"\" + subj + \"\")';>Delete quiz</a></div></div></li>");
            }
        }
    } else {
        $("#list_" + toggle).append("<li class='list-group-item'>" + chap_num + "." + lesson_num + " " + childSnapshot.key + " <a class = 'btn btn-info' href = 'createquiz.php?title=" + hey + "&type=aq&sj=" + CryptoJS.AES.encrypt(selected_subject, "Secret Passphrase") + "&chap=" + CryptoJS.AES.encrypt(panel_title, "Secret Passphrase") + "'> Create Quiz For This Lesson</a></li>");
    }
}

});
});
});
});

function deleteQuiz(loc) {
    var conf = confirm('Are you sure to delete the selected quiz? The action is irreversible');
    if (conf == true) {
        var location = CryptoJS.AES.decrypt(loc, "Secret Passphrase").toString(CryptoJS.enc.Utf8);
        firebase.database().ref(location + "/quiz").remove().then(function() {
            alert("Quiz successfully deleted!");
            setTimeout(function() {
                window.location.href = "quiz.php";
            }, 1000);
        });
    }
}

```

```

}

function startQuiz(loc) {
    var conf = confirm('Start the quiz?');
    if (conf == true) {
        var started = "yes";
        var location = CryptoJS.AES.decrypt(loc, "Secret
Passphrase").toString(CryptoJS.enc.Utf8);
        firebase.database().ref(location + "/quiz").update({
            quiz_started: started
        }).then(function() {
            setTimeout(function() {
                window.location.href = "quiz.php";
            }, 1000);
        });
    }
}

function endQuiz(loc) {
    var conf = confirm('End the quiz?');
    if (conf == true) {
        var started = "no";
        var location = CryptoJS.AES.decrypt(loc, "Secret
Passphrase").toString(CryptoJS.enc.Utf8);
        firebase.database().ref(location + "/quiz").update({
            quiz_started: started
        }).then(function() {
            setTimeout(function() {
                window.location.href = "quiz.php";
            }, 1000);
        });
    }
}

function extend(loc) {
    $("#extendmodal").modal('show');
    var datetoday = moment().format("YYYY-MM-DDTHH:mm");
    $("#dt_extend").attr("value", datetoday);
    $("#btn_extend").attr("onclick", "extendQuiz('" + loc + "')");
}

function extendQuiz(loc) {
    var started = "yes";
    var location = CryptoJS.AES.decrypt(loc, "Secret
Passphrase").toString(CryptoJS.enc.Utf8);
    var exp = new Date($("#dt_extend").val()).toUTCString();
    firebase.database().ref(location + "/quiz").update({
        quiz_started: started,
        quiz_expiration: exp
    }).then(function() {
        alert("Quiz expiration extended!");
        setTimeout(function() {
            window.location.href = "quiz.php";
        }, 1000);
    });
}

register.js

```

```

$(window, document, undefined).ready(function() {
    $("#frm_register").on("submit", function(e){

```

```

        e.preventDefault();
        var user_email = $("#txt_reg_email").val();
        var user_pass = $("#txt_reg_pass").val();
        var user_fname = $("#txt_reg_fname").val();
        var user_lname = $("#txt_reg_lname").val();
        var user_type = "Teacher";
        firebase.auth().createUserWithEmailAndPassword(user_email,
user_pass).then(function(e){
            firebase.database().ref().child("members").push({
                email: user_email,
                firstname: user_fname,
                lastname : user_lname,
                type: user_type,
            }).then(function(){
                var user = firebase.auth().currentUser;
                user.sendEmailVerification().then(function() {
                    alert("Check your email for confirmation. ");
                    location.replace("login.php");
                }).catch(function(error) {
                    alert("Error: " + error.message);
                });
            }).catch(function(error){
                var errorCode = error.code;
                var errorMessage = error.message;
                alert(errorCode + ": " + errorMessage);
            });
        }).catch(function(error) {
            // Handle Errors here.
            var errorCode = error.code;
            var errorMessage = error.message;
            alert(errorCode + ": " + errorMessage);
        });
    });
});

```

studassessment.js

```

var test = localStorage.loggedin;
if (test != "true") {
    window.location.replace("welcomepage.php");
}
if (localStorage.locked == "true") {
    window.location.replace("lock_screen.php");
}
var config = {
    apiKey: "AIzaSyD0CsnZgDBGSFBVrtZH0HRBbVQWk__txe0",
    authDomain: "ezbio-98be1.firebaseio.com",
    databaseURL: "https://ezbio-98be1.firebaseio.com",
    storageBucket: "ezbio-98be1.appspot.com",
    messagingSenderId: "595946963410"
};
firebase.initializeApp(config);
$(document).ready(function(){
    $("#teachername").html(localStorage.teacher_name);
    getStudentName();
    setTimeout(function() { getGrades(); }, 2000);
    setTimeout(function() { performStatistics(); }, 3000);
});
function getStudentName(){
    var key = GetParameterValues("key");

```

```

firebase.database().ref("members/"+key).orderByKey().once('value').then(function(snapshot) {
    $("#studentName").html(snapshot.val().firstname + " " + snapshot.val().lastname);
});
}
var timesToRead = new Array();
var timeOnQuiz = new Array();
var scores = new Array();
function getGrades(){
    var key = GetParameterValues("key");
    var sectionkey = GetParameterValues("sectionkey");
    var teacherkey = localStorage.memberkey;

    firebase.database().ref("assessment/"+teacherkey+"/"+sectionkey+"/"+key).orderByKey().once('value').then(function(snapshot) {
        snapshot.forEach(function(childSnapshot){

            $("#rows").append("<tr><td>" + childSnapshot.key + "</td><td>" + parseFloat(childSnapshot.val().timeonRead/60).toFixed(2) + "</td><td>" + parseFloat(childSnapshot.val().timeonQuiz/60).toFixed(2) + "</td><td>" + childSnapshot.val().score + "</td><td>" + childSnapshot.val().date_quiztaken + "</td></tr>");
            timesToRead.push(parseFloat(childSnapshot.val().timeonRead/60).toFixed(2));
            timeOnQuiz.push(parseFloat(childSnapshot.val().timeonQuiz/60).toFixed(2));
            scores.push(parseFloat(childSnapshot.val().score));
        });
    });
}
function performStatistics(){
    $("#minimetoread").html(ss.min(timesToRead));
    $("#maximetoread").html(ss.max(timesToRead));
    $("#avgtimetoread").html(ss.mean(timesToRead));
    $("#minimetoquiz").html(ss.min(timeOnQuiz));
    $("#maximetoquiz").html(ss.max(timeOnQuiz));
    $("#avgtimetoquiz").html(ss.mean(timeOnQuiz));
    $("#minperf").html(ss.min(scores));
    $("#maxperf").html(ss.max(scores));
    $("#avgperf").html(ss.mean(scores));
}
function GetParameterValues(param) {
    var url = window.location.href.slice(window.location.href.indexOf('?') + 1).split('&');
    for (var i = 0; i < url.length; i++) {
        var urlparam = url[i].split('=');
        if (urlparam[0] == param) {
            return urlparam[1];
        }
    }
}

```

syllabus.js

```

var test = localStorage.loggedin;
if (test != "true") {
    window.location.replace("welcomepage.php");
}
if (localStorage.locked == "true") {
    window.location.replace("lock_screen.php");
}
var selected_chapter = "";

```

```

var config = {
  apiKey: "AIzaSyD0CsnZgDBGSFBVrtZH0HRBbVQWk__txe0",
  authDomain: "ezbio-98be1.firebaseio.com",
  databaseURL: "https://ezbio-98be1.firebaseio.com",
  storageBucket: "ezbio-98be1.appspot.com",
  messagingSenderId: "595946963410"
};
firebase.initializeApp(config);
$(document).ready(function() {
  $("#teachername").html(localStorage.teacher_name);
  var lesson_titles = new Array();
  firebase.database().ref("subjects/"
  localStorage.memberkey).orderByKey().once('value').then(function(snapshot) {
    snapshot.forEach(function(childSnapshot) {
      $('#cmb_subjects').append("<option value = '" + childSnapshot.key + "'> " +
childSnapshot.key + "</option>");
    });
  }).then(function() {
    loadSubject();
  });
  $(window).bind('beforeunload', function(){
    sessionStorage.removeItem('editongoing');
    sessionStorage.removeItem('oldKey');
  });
  $("#txt_sidenotes").summernote();
  $(document).on("change", "#cmb_subjects", function() {
    $("#container_subjects").html("");
    loadSubject();
  });
  $("#frm_addlesson").hide();
  $("#frm_addnewchapter").hide();
  $("#up_progress").hide();
  $("#form_addlesson").validate();
  $("#form_addchap").validate();
  jQuery.validator.setDefaults({
    debug: true,
    success: "valid"
  });
  $("form").submit(function(e){
    e.preventDefault();
  });
  $(document).on("click", "#btn_addchap", function() {
    if($("#form_addchap").valid()){
      var alreadyexists = false;
      var selected_subject = $('#cmb_subjects').val();
      var chaptitle = $("#txt_name").val();
      firebase.database().ref("subjects/" + localStorage.memberkey + "/" +
selected_subject).once('value').then(function(snapshot){
        if(snapshot.child(chaptitle).exists()){
          alreadyexists = true;
        }
      });
      setTimeout(function(){
        if(alreadyexists == true){
          alert("Chapter already exists.");
        }else{
          if(sessionStorage.editongoing == 1){
            var conf = window.confirm('Edit this chapter? The action is
irreversible.');
```

```

        var newchap = $("#txt_name").val();
        var oldRef =
firebase.database().ref("subjects/"+localStorage.memberkey+"/"+cank+"/"+sessionStorage.
oldKey);
        var newRef=
firebase.database().ref("subjects/"+localStorage.memberkey+"/"+cank+"/"+newchap);
        moveChapterChildren(oldRef, newRef);
    }
}
}else{
    var dt = new Date();
    var added = dt.toUTCString();
    firebase.database().ref("subjects/" + localStorage.memberkey +
"/" + selected_subject + "/" + chaptitle).set({
        chap_date_added: added
    }).then(function() {
        window.alert("Chapter Added");
        window.location.replace("syllabus.php");
    });
}
    }
}, 1000);
}
});
$(document).on("click", "#btn_addlesson", function() {
    if($("#form_addlesson").valid()){
        var alreadyexists = false;
        var selected_subject = $("#cmb_subjects").val();
        var lessonTitle = $("#txt_lessonName").val();
        var dt = new Date();
        var added = dt.toUTCString();
        var sidenotes = $("#txt_sidenotes").summernote('code');
        var expiration = new Date($("#dt_expiration").val()).toUTCString();
        //var hashed_filename =
        //console.log(material_file);
        firebase.database().ref("subjects/" + localStorage.memberkey + "/" +
selected_subject + "/" + selected_chapter).once('value').then(function(snapshot){
            if(snapshot.child(lessonTitle).exists()){
                alreadyexists = true;
            }
        });
        setTimeout(function(){
            if(alreadyexists == true){
                alert("Cannot proceed. Lesson by that name already exists");
            }else{
                if(sessionStorage.editongoing == 2){
                    var conf = window.confirm('Edit this lesson? The action is
irreversible.');
```

```

    }else{
        var material_file = $("#file_lessonMaterial").prop("files")[0];
        var material = $("#file_lessonMaterial").val();
        var material_filename = material.split('\\').pop();
        var material_loc =
"files/"+localStorage.memberkey+"/"+selected_subject+"/"+selected_chapter+"/"+material_
filename;
        firebase.database().ref("subjects/" + localStorage.memberkey +
"/" + selected_subject + "/" + selected_chapter + "/" + lessonTitle).set({
            lesson_date_added: added,
            lesson_material : material_loc,
            lesson_sidenotes : sidenotes
        }).then(function() {
            var uploadFile =
firebase.storage().ref().child(material_loc).put(material_file);
            $("#up_progress").show();
            uploadFile.on(firebase.storage.TaskEvent.STATE_CHANGED, // or
'state_changed'
                function(snapshot) {
                    // Get task progress, including the number of bytes uploaded
                    and the total number of bytes to be uploaded
                    var progress = Math.floor((snapshot.bytesTransferred /
snapshot.totalBytes) * 100);
                    //console.log('Upload is ' + progress + '% done');
                    if(progress == 100){
                        $("#prog_upload").attr('aria-valuenow',
progress).css('width', progress+"%");
                        $("#prog_upload").html(progress + "% uploading");
                        setTimeout(lessonbye(), 10000);
                    }else{
                        $("#prog_upload").attr('aria-valuenow',
progress).css('width', progress+"%");
                        $("#prog_upload").html(progress + "% uploading");
                    }
                }
            );
        });
    }
}, 2000);
}
});
});
$("#txt_lessonName").on("keypress", function(event){
    var regex = new RegExp("^[a-zA-Z0-9\\s]+$");
    var key = String.fromCharCode(!event.charCode ? event.which : event.charCode);
    if (!regex.test(key)) {
        event.preventDefault();
        return false;
    }
});
function loadSubject() {
    var selected_subject = $('#cmb_subjects').val();
    var chap_num = 0;
    firebase.database().ref("subjects/" + localStorage.memberkey + "/" +
selected_subject).orderByChild("chap_date_added").once('value').then(function(snapshot)
    {
        if (snapshot.numChildren() <= 2) //1 because of the subject added timestamp
            $("#container_subjects").append("<h1>No chapters on this
subject.</h1><br>");
        snapshot.forEach(function(childSnapshot) {

```

```

        if (childSnapshot.key != "subj_date_added" && childSnapshot.key !=
"subj_description" && childSnapshot.key != "custom_quizzes") {
            chap_num++;
            var toggle = childSnapshot.key.replace(/\s+/g, ''); //used for names of
div
            var panel_title = childSnapshot.key;
            $("#container_subjects").append("<div class='panel-group'> <div
class='panel panel-info'> <div class='panel-heading'> <h4 class='panel-title'> <a data-
toggle='collapse' href='#" + toggle + "'>" + chap_num + ". " + childSnapshot.key + "</a>
</h4> </div> <div id='" + toggle + "' class='panel-collapse collapse'> <ul class='list-
group' id = list_" + toggle + "'> </ul> <ul class = 'list-group'><li class = 'list-group-
item'><a class = 'btn btn-info' onclick = 'addLesson(\"" + panel_title + "\")'> + Add
Lesson </a></li><li class = 'list-group-item'><a class = 'btn btn-info' href =
'createquiz.php?title=" + CryptoJS.AES.encrypt(childSnapshot.key, "Secret Passphrase")
+"&type=ct&subj=" + CryptoJS.AES.encrypt(selected_subject, "Secret Passphrase") + "'> +
Create Quiz For This Chapter</a></li><li class = 'list-group-item'><a class = 'btn btn-
warning' onclick = 'editChapter(\""+panel_title+"\"');>Edit Chapter</a></li><li class =
'list-group-item'> <a class = 'btn btn-danger' onclick =
'deleteChapter(\""+panel_title+"\"');>Delete Chapter</a></li></ul> </div> </div>
</div>");
            firebase.database().ref("subjects/" + localStorage.memberkey + "/" +
selected_subject + "/" +
childSnapshot.key).orderByChild("lesson_date_added").once('value').then(function(snapsho
t) {
                var lesson_num = 0;
                snapshot.forEach(function(childSnapshot) {
                    if (childSnapshot.key != "chap_date_added" && childSnapshot.key
!= "quiz") {
                        lesson_num++;
                        var hey = CryptoJS.AES.encrypt(childSnapshot.key, "Secret
Passphrase");
                        if(childSnapshot.child("quiz").exists()){
                            $("#list_" + toggle).append("<li class='list-group-
item'><div class = 'row'><div class = 'col-sm-4'>" + chap_num + "." + lesson_num + " "
+ childSnapshot.key + "</div><div class = 'col-sm-4'><font color = 'green'>Quiz Already
Created for this Lesson</font></div><div class = 'col-sm-4'><a class = 'btn btn-warning'
onclick = 'editLesson(\""+panel_title+"\", \""+childSnapshot.key+"\"');>Edit Lesson</a>
<a class = 'btn btn-danger' onclick =
'deleteLesson(\""+panel_title+"\", \""+childSnapshot.key+"\"');>Delete
Lesson</a></div></div></li>");
                        }else{
                            $("#list_" + toggle).append("<li class='list-group-
item'><div class = 'row'><div class = 'col-sm-4'>" + chap_num + "." + lesson_num + " "
+ childSnapshot.key + "</div><div class = 'col-sm-4'><a class = 'btn btn-info' href =
'createquiz.php?title=" + hey + "&type=aq&sj=" + CryptoJS.AES.encrypt(selected_subject,
"Secret Passphrase") + "&chap=" + CryptoJS.AES.encrypt(panel_title, "Secret Passphrase")
+ "'> + Create Quiz For This Lesson</a></div><div class = 'col-sm-4'><a class = 'btn btn-
warning' onclick = 'editLesson(\""+panel_title+"\", \""+childSnapshot.key+"\"');>Edit
Lesson</a> <a class = 'btn btn-danger' onclick =
'deleteLesson(\""+panel_title+"\", \""+childSnapshot.key+"\"');>Delete
Lesson</a></div></div></li>");
                        }
                    }
                });
            });
        });
    });
    $("#btn_deletesubject").on('click', function(e){
        e.preventDefault();

```



```

        var conf = confirm('Are you sure to delete the selected subject? All chapters,
lessons, and quizzes on the syllabus will be deleted. The action is irreversible');
        if(conf == true){
            var cank = $("#cmb_subjects").val();

firebase.database().ref("subjects/"+localStorage.memberkey+"/"+cank).remove().then(func
tion(){
            alert("Subject successfully deleted!");
            setTimeout(function() {window.location.href = "syllabus.php";}, 1000);
        });
    }
});
$("#btn_editsubject").on('click', function(e){
    e.preventDefault();
    var hey = CryptoJS.AES.encrypt($("#cmb_subjects").val(), "Secret Passphrase");
    window.location.replace("addnewsbjeect.php?action=1&title="+hey);
});
function deleteChapter(key){
    var conf = window.confirm('Delete this Chapter? The lessons and quizzes under the
selected chapter will be deleted. The action is irreversible.');
```

selected chapter will be deleted. The action is irreversible.');

```

        var cank = $("#cmb_subjects").val();
        if(conf == true){

firebase.database().ref("subjects/"+localStorage.memberkey+"/"+cank+"/"+key).remove().t
hen(function(){
            alert("Chapter successfully deleted!");
            setTimeout(function() {window.location.href = "syllabus.php";}, 1000);
        });
    }
}
function editChapter(key){
    sessionStorage.editongoing = 1;
    sessionStorage.oldKey = key;
    $("#frm_addnewchapter").show();
    $("#container_subjects").hide();
    $("#link_addchap").hide();
    $("#txt_name").val(key);
}
function deleteLesson(chapter, lesson){
    var conf = window.confirm('Delete this Lesson? The quizzes under the selected lesson
will be deleted. The action is irreversible.');
```

will be deleted. The action is irreversible.');

```

        var cank = $("#cmb_subjects").val();
        if(conf == true){

firebase.database().ref("subjects/"+localStorage.memberkey+"/"+cank+"/"+chapter+"/"+les
son).remove().then(function(){
            alert("Lesson successfully deleted!");
            setTimeout(function() {window.location.href = "syllabus.php";}, 1000);
        });
    }
}
function editLesson(chapter, lesson){
    sessionStorage.editongoing = 2;
    sessionStorage.oldKeyForLessonChapter = chapter;
    sessionStorage.oldKeyForLesson = lesson;
    var cank = $("#cmb_subjects").val();
    $("#frm_addlesson").show();
    $("#chaptertitle").append(chapter);
    selected_chapter = chapter;
    $("#container_subjects").hide();
    $("#link_addchap").hide();

```

```

firebase.database().ref("subjects/"+localStorage.memberkey+"/"+cank+"/"+chapter+"/"+lesson).once('value').then(function(snapshot){
    $("#txt_lessonName").val(snapshot.key);
    $("#txt_sidenotes").summernote('code', snapshot.val().lesson_sidenotes);
    /*$("#file_lessonMaterial").prop("files")[0];
    $("#file_lessonMaterial").val(snapshot.val().lesson_material);*/
});
}
function moveChapterChildren(oldRef, newRef) {
    oldRef.once('value', function(snap) {
        newRef.set( snap.val(), function(error) {
            if( !error ) { oldRef.remove(); alert("Chapter successfully edited!");
            setTimeout(function() {window.location.href = "syllabus.php";}, 1000);}
            else if( typeof(console) !== 'undefined' && console.error ) {
console.error(error); }
        });
    });
}
function moveLessonChildren(oldRef, newRef) {
    oldRef.once('value', function(snap) {
        newRef.set( snap.val(), function(error) {
            if( !error ) { oldRef.remove(); alert("Lesson successfully edited!");
            setTimeout(function() {window.location.href = "syllabus.php";}, 1000);}
            else if( typeof(console) !== 'undefined' && console.error ) {
console.error(error); }
        });
    });
}
function addNewChapter() {
    $("#frm_addnewchapter").show();
    $("#container_subjects").hide();
    $("#link_addchap").hide();
}
function addLesson(c) {
    $("#frm_addlesson").show();
    $("#chaptertitle").append(c);
    selected_chapter = c;
    $("#container_subjects").hide();
    $("#link_addchap").hide();
}
function lessonbye(){
    window.alert("Lesson Added");
    window.location.replace("syllabus.php");
}
function GetParameterValues(param) {
    var url = window.location.href.slice(window.location.href.indexOf('?') +
1).split('&');
    for (var i = 0; i < url.length; i++) {
        var urlparam = url[i].split('=');
        if (urlparam[0] == param) {
            return urlparam[1];
        }
    }
}
}

```

William Jefferson A. Albao

+639293358802
wjalbao@gmail.com
38-B Samson Road
Caloocan City



OBJECTIVE

To look for a suitable work where I could use my knowledge and skills as an IT professional and to develop my personality as a career person while utilizing my skills.

On-the Job Training history

Alorica Philippines – GF Three Cyberpod Centris, South Tower, ETON Centris, EDSA Cor. Quezon Ave., Brgy. Pinyahan, Diliman Dist., Quezon City, Philippines

IT Department – IT Support Trainee
April 2017 – June 2017

Tasks:

- Hardware and Network Troubleshooting
- Re-imaging PCs
- Installing softwares by remoting the PCs
- Troubleshooting of AVAYA phones
- Checking monitors for disposal
- Updating the floor/cube map of the company

EDUCATION

TERTIARY

COURSE : BACHELOR OF SCIENCE IN COMPUTER SCIENCE
SCHOOL : UNIVERSITY OF THE EAST CALOOCAN
YEAR : 2014– 2018
THESIS TITLE : **MOBILE LEARNING MANAGEMENT SYSTEM: A TOOL FOR BASIC EDUCATION**
ADVISOR : DR. DENNIS B. GONZALES

ACHIEVEMENTS

- COLLEGE SCHOLAR FOR 2ND SEMESTER 2014-2015
- COLLEGE SCHOLAR FOR 2ND SEMESTER 2015-2016

SECONDARY

SCHOOL : CALOOCAN HIGH SCHOOL
YEAR : 2010 – 2014

ACHIEVEMENT

- TOP 1 OF THE CLASS S.Y 2013-2014

ELEMENTARY

SCHOOL : GOMBURZA ELEMENTARY SCHOOL
YEAR : 2003 – 2010

PROFESSIONAL QUALIFICATIONS

▪ C++	2014
▪ Java	2015
▪ HTML/CSS/JavaScript/PHP/MySQL	2015
▪ Visual Basic	2015
▪ C#	2016
▪ Android	2016

SEMINARS ATTENDED

▪ C++	2014
▪ Python Programming	2016
▪ Photo Manipulation with Adobe Photoshop	2016
▪ Corel Video Studio Pro x6 – Basic Video Editing	2016
▪ Introduction to Macromedia Flash 8	2016
▪ Firebase	2016
▪ Node.js	2016
▪ Basic Ren'Py	2016
▪ Writing A+ Research Paper	2017

SKILLS AND QUALIFICATIONS

KNOWLEDGEABLE IN PROGRAMMING/MARKUP LANGUAGES

- C#, Visual Basic, JavaScript, HTML, CSS, PHP, Java, Android

KNOWLEDGEABLE IN DATABASE

- MySQL
- Firebase

KNOWLEDGEABLE IN APPLICATION SOFTWARE

- Visual Studio, Android Studio

PROFICIENT IN USING MICROSOFT OFFICE APPLICATIONS

- MS Word, MS Excel, MS Powerpoint

Christine Anne O. Edoño

+639499948709

christineedoo@yahoo.com

123 Capt. Cruz St., Barangay Parada Valenzuela City



OBJECTIVE

To look for a suitable work where I could use my knowledge and skills as an IT professional and to develop my personality as a career person while utilizing my skills.

On-the Job Training history

Alorica Philippines – Cybergate Tower 1, Pioneer St., Mandaluyong City

IT Department – Technical Support Trainee

April 2017 – May 2017

Tasks:

- Hardware and Network Troubleshooting
- Re-imaging PCs
- Installing software by remoting the PCs
- Troubleshooting of AVAYA phones
- Checking monitors for disposal
- Updating the floor/cube map of the company

EDUCATION

TERTIARY

COURSE : BACHELOR OF SCIENCE IN COMPUTER SCIENCE
SCHOOL : UNIVERSITY OF THE EAST CALOOCAN
YEAR : 2014– 2018
THESIS TITLE : **MOBILE LEARNING MANAGEMENT SYSTEM: A TOOL FOR BASIC EDUCATION**
ADVISOR : DR. DENNIS B. GONZALES

ACHIEVEMENTS

- Consistent scholar of MEADE (Metrobank Educational Assistance for Dependents of Employees)
- Highest earned GWA of 1.65 1st Year-2nd Semester
- PRO Internal Officer of ACSS (Association of Computer Studies and Systems Students) 2016-2017
- Vice President Internal Officer of ACSS (Association of Computer Studies and Systems Students) 2017-2018

SECONDARY

SCHOOL : ST. LOUIS COLLEGE VALENZUELA
YEAR : 2010 – 2014

ACHIEVEMENT

- Earned "Most Responsible" award at 3rd year Highschool Recognition Day
- Elected as Treasurer of the class SY. 2012-2013

ELEMENTARY

SCHOOL : CHRIST AMBASSADOR'S CHRISTIAN ACADEMY
YEAR : 2004 – 2010

ACHIEVEMENT

- Consistently in the Top 10 students of the class from Grade 1 to 6

PROFESSIONAL QUALIFICATIONS

▪ C++	2014
▪ Macromedia Flash	2015
▪ Learn Javascript (Codecademy)	2016
▪ JQuery (Codecademy)	2016
▪ Javascript (Codecademy)	2017
▪ ACSS: The Tech World Today	2017
▪ ACSS: PerDev (Personality Development)	2017

SEMINARS ATTENDED

▪ C++	2014
▪ Python Programming	2016
▪ Photo Manipulation with Adobe Photoshop	2016
▪ Corel Video Studio Pro x6 – Basic Video Editing	2016
▪ Introduction to Macromedia Flash 8	2016
▪ Firebase	2016
▪ Node.js	2016
▪ Basic Ren'Py	2016
▪ Writing A+ Research Paper	2017
▪ ACSS: The Tech World Today	2017
▪ ACSS: PerDev (Personality Development)	2017

SKILLS AND QUALIFICATIONS

KNOWLEDGEABLE IN PROGRAMMING/MARKUP LANGUAGES

- C#, Visual Basic, JavaScript, HTML, CSS, PHP

KNOWLEDGEABLE IN DATABASE

- MySQL

KNOWLEDGEABLE IN APPLICATION SOFTWARE

- Visual Studio

PROFICIENT IN USING MICROSOFT OFFICE APPLICATIONS

- MS Word, MS Excel, MS Powerpoint

Patrick John Dabi Fumera

+639279847306

fumerapj@gmail.com

Phase 1, Unit 12 Victoneta Townhomes, Barangay Bahay Toro

EDSA, Muñoz, Quezon City



OBJECTIVE

To look for a suitable work where I could use my knowledge and skills as an IT professional and to develop my personality as a career person while utilizing my skills.

On-the Job Training history

Dempsey Resource Management – Vicars Building 31 Visayas Avenue, Brgy. Vasra Quezon City, Philippines

HR Recruitment Department – Technical Support Trainee

April 2017 – May 2017

Tasks:

- Searching applicants for a job vacancy
- Ensuring security between the employer and the applicant
- Scheduling job interviews

EDUCATION

TERTIARY

COURSE : BACHELOR OF SCIENCE IN COMPUTER SCIENCE

SCHOOL : UNIVERSITY OF THE EAST CALOOCAN

YEAR : 2014– 2018

THESIS TITLE : **MOBILE LEARNING MANAGEMENT SYSTEM: A TOOL FOR BASIC EDUCATION**

ADVISOR : DR. DENNIS B. GONZALES

ACHIEVEMENTS

- Consistent scholar of Commission on Higher Education *Tulong-Dunong* Student Financial Assistance Program (CHED-StuFAP).
- College Scholar for 1st Semester 2014-2015
- University Scholar for 2nd Semester 2014-2015
- College Scholar for 1st Semester 2015-2016
- University Scholar for 2nd Semester 2016-2017
- University Scholar for 1st Semester 2016-2017
- University Scholar for 2nd Semester 2016-2017

SECONDARY

SCHOOL : MANILA CENTRAL UNIVERSITY

YEAR : 2010 – 2014

ACHIEVEMENT

- Top 1 of the class S.Y 2010-2011

ELEMENTARY

SCHOOL : SAN ANTONIO ELEMENTARY SCHOOL
YEAR : 2003 – 2010

PROFESSIONAL QUALIFICATIONS

▪ C++	2014
▪ Java	2015
▪ HTML/CSS/JavaScript/PHP/MySQL	2015
▪ Visual Basic	2015
▪ C#	2016

SEMINARS ATTENDED

▪ C++	2014
▪ Python Programming	2016
▪ Photo Manipulation with Adobe Photoshop	2016
▪ Corel Video Studio Pro x6 – Basic Video Editing	2016
▪ Introduction to Macromedia Flash 8	2016
▪ Firebase	2016
▪ Node.js	2016
▪ Basic Ren'Py	2016
▪ Writing A+ Research Paper	2017

SKILLS AND QUALIFICATIONS

KNOWLEDGEABLE IN PROGRAMMING/MARKUP LANGUAGES

- C#, Visual Basic, JavaScript, HTML, CSS, PHP

KNOWLEDGEABLE IN DATABASE

- MySQL
- Firebase

KNOWLEDGEABLE IN APPLICATION SOFTWARE

- Visual Studio

PROFICIENT IN USING MICROSOFT OFFICE APPLICATIONS

- MS Word, MS Excel, MS Powerpoint

Elaiza C. Malayto

+639759405975

elaizamalayto@yahoo.com

75 Tiangco St. Brgy. Ibaba Malabon City



OBJECTIVE

To acquire a position in the field of Computer Programming where I can utilize my skills as an IT professional and allow me the opportunity to gain additional skills and experience.

On-the Job Training history

Alorica Philippines – Cybergate Tower 1, Pioneer St., Mandaluyong City

IT Department – Technical Support Trainee

April 2017 – May 2017

Tasks:

- Hardware and Network Troubleshooting
- Re-imaging PCs
- Installing software by remoting the PCs
- Troubleshooting of AVAYA phones
- Checking monitors for disposal
- Updating the floor/cube map of the company

EDUCATION

TERTIARY

COURSE	: BACHELOR OF SCIENCE IN COMPUTER SCIENCE
SCHOOL	: UNIVERSITY OF THE EAST
YEAR	: 2014– 2018

SECONDARY

SCHOOL	: ARELLANO UNIVERSITY – ELISA ESGUERRA CAMPUS
YEAR	: 2010 – 2014

ELEMENTARY

SCHOOL	: SANTIAGO SYJUCO MEMORIAL SCHOOL
YEAR	: 2004 - 2010

PROFESSIONAL QUALIFICATIONS

- | | |
|--|------|
| ▪ C++ | 2014 |
| ▪ Macromedia Flash | 2015 |
| ▪ Learn Javascript (Codecademy) | 2016 |
| ▪ JQuery (Codecademy) | 2016 |
| ▪ Javascript (Codecademy) | 2017 |
| ▪ Android Summit | 2017 |
| ▪ ACSS: PerDev (Personality Development) | 2017 |

SEMINARS ATTENDED

▪ C++ (UE Caloocan)	2014
▪ Python Programming (UE Caloocan)	2016
▪ Photo Manipulation with Adobe Photoshop (UE Caloocan)	2016
▪ Corel Video Studio Pro x6 – Basic Video Editing (UE Caloocan)	2016
▪ Introduction to Macromedia Flash 8 (UE Caloocan)	2016
▪ Firebase (UE Caloocan)	2016
▪ Node.js (UE Caloocan)	2016
▪ Basic Ren'Py (UE Caloocan)	2016
▪ Writing A+ Research Paper (UE Caloocan)	2017
▪ Android Summit (FEU Tech)	2017
▪ ACSS: PerDev (UE Caloocan)	2017

SKILLS AND QUALIFICATIONS

KNOWLEDGEABLE IN PROGRAMMING/MARKUP LANGUAGES

- PHP, C#, Visual Basic, Javascript, HTML, CSS

KNOWLEDGEABLE IN DATABASE

- MySQL

KNOWLEDGEABLE IN APPLICATION SOFTWARE

- Visual Studio

PROFICIENT IN USING MICROSOFT OFFICE APPLICATIONS

- MS Word, MS Excel, MS Powerpoint

Elbert G. Martin

+639759406934

elbert.martin83@yahoo.com

177 8th Ave. West Grace Park, Caloocan City



OBJECTIVE

My objective is to enhance my skills and knowledge, to get a work experience which I can use after I graduate and to be trained in different types of field in the industry.

On-the Job Training History

University of the East: Caloocan – 105 Samson Road St. Caloocan City

Elementary and High School Department – Database Administrator

April 2017 – May 2017

Tasks:

- Encoding of soon to be enrolled students
- Compiling of Cards to their respective folders
- Revising of important documents

EDUCATION

TERTIARY

COURSE : BACHELOR OF SCIENCE IN COMPUTER SCIENCE

SCHOOL : UNIVERSITY OF THE EAST CALOOCAN

YEAR : 2014– 2018

THESIS TITLE : **MOBILE LEARNING MANAGEMENT SYSTEM: A TOOL FOR BASIC EDUCATION**

ADVISOR : DR. DENNIS B. GONZALES

ACHIEVEMENTS

- Consistent scholar of Cong. Egay Erice (Caloocan city)
- Best male performer of LAMPS (Lakas Mag-aaral ng Pamantasan ng Silangan)
- Freshman Representative of ACSS (Association Of Computer Studies And Systems Students) SY 2014-2015
- Pro-internal of ACSS (Association Of Computer Studies And Systems Students) SY. 2015-2016
- Vice President Internal Officer of ACSS (Association Of Computer Studies And Systems Students) SY. 2016-2017
- Communications Chairman of LAMPS (Lakas Mag-aaral ng Pamantasan ng Silangan) SY. 2015-2016
- Non-academic chairman LAMPS (Lakas Mag-aaral ng Pamantasan ng Silangan) SY. 2016-2017
- College of Engineering – Board of Director of LAMPS (Lakas Mag-aaral ng Pamantasan ng Silangan) SY. 2017-2018

SECONDARY

SCHOOL : CALOOCAN HIGH SCHOOL

YEAR : 2010 – 2014

ACHIEVEMENTS

- Best in Filipino SY. 2010-2011
- 5th Honor SY. 2010-2011
- Most Polite SY. 2011-2012
- 10th Honor SY. 2011-2012

- 9th Place In Araling Panlipunan Quiz Bee SY. 2011-2012
- 1st Place In Poem Making Contest SY. 2011-2012
- 10th Honor SY. 2012-2013
- Perfect Attendance SY. 2012-2013
- Overall Vice President Of 4th Year High School Regular Basic Education Curriculum SY. 2013-2014
- Representative Of "BERT" (Batang Emergency Response Team) SY. 2013-2014
- Auditor Of 2nd Year Sec. Zinnia SY. 2011-2012
- Vice President Of 3rd Year Sec. Teresa Magbanua SY. 2012-2013
- Vice President Of 4th Year Sec. Magnesium SY. 2013-2014

ELEMENTARY

SCHOOL : CALOOCAN ELEMENTARY SCHOOL - CENTRAL
YEAR : 2004 – 2010

ACHIEVEMENT

- 10TH HONOR SY. 2005 – 2006

PROFESSIONAL QUALIFICATIONS

▪ C++	2014
▪ Macromedia flash	2015
▪ Learn javascript (codecademy)	2016
▪ JQuery (codecademy)	2016
▪ Javascript (codecademy)	2017

SEMINARS ATTENDED

▪ C++	2014
▪ Python Programming	2016
▪ Photo Manipulation with Adobe Photoshop	2016
▪ Corel Video Studio Pro x6 – Basic Video Editing	2016
▪ Introduction to Macromedia Flash 8	2016
▪ Firebase	2016
▪ Node.js	2016
▪ Basic Ren'Py	2016
▪ Writing A+ Research Paper	2017

SKILLS AND QUALIFICATIONS

KNOWLEDGEABLE IN PROGRAMMING/MARKUP LANGUAGES

- C++, C#, Visual Basic, JavaScript, HTML, CSS, PHP

KNOWLEDGEABLE IN DATABASE

- MySQL

KNOWLEDGEABLE IN APPLICATION SOFTWARE

- Visual Studio, Dev C++

PROFICIENT IN USING MICROSOFT OFFICE APPLICATIONS

- MS Word, MS Excel, MS Powerpoint, MS Publisher