AAPP012-4-2-RWDD

RESPONSIVE WEB DESIGN & DEVELOPMENT

GROUP ASSIGNMENT

GTSPM: Interactive Web Quiz for SPM students

Student ID	Student Name	Intake
TP073318	Ejjaz Hakimi Bin Mohamad Azan	UCDF2307ICT(SE)
TP073631	Elianna Catrina Herrera	UCDF2307ICT(SE)
TP074762	Suchitra Nambiar A/P Mahandran	UCDF2307ICT(SE)
TP073088	Ng Vin Ee	UCDF2307ICT(SE)
TP076723	Danesh A/L Sellasamy	UCDF2307ICT(SE)
TP076282	Kwan Chun Hoe	UCDF2307ICT(SE)

Table of Contents

1.0 Introduction

1.1 Project Introduction

The Sijil Pelajaran Malaysia (SPM) examination is the main national examination for Malaysia, it is administered centrally by the Ministry of Education (KPM) and is conducted at the end of secondary education. The awarding of SPM certification is pivotal for the students as it serves multiple purposes: basic eligibility to enter the program at a higher level, basic criteria in obtaining scholarships, apply for a job (My Government, 2024). However, in SMK Puteri, we found out that not all students are able to find sources for SPM exercises which causes them to potentially miss out on the key topics or trends and leads to poor time management, especially when focusing on certain topics that are likely to appear in future exams. Other than that, for the students to excel in the SPM, it requires consistent practices to enhance the knowledge that they have learned, traditional methods, however, rely heavily on printed material, which is not eco-friendly and is difficult to find the source. Therefore, these challenges led to the development of "GTSPM", a digital platform specifically designed for SMK Puteri fifth-form students who are preparing for their Sijil Pelajaran Malaysia (SPM) examinations. Acknowledging the importance of SPM for the student's academic and professional future, GTSPM is designed to provide comprehensive learning materials that aim to aid the students in fully grasping the subject knowledge and having a clear understanding of each topic. By transitioning to a digital platform, GTSPM reduces the reliance on physical resources, promotes a more eco-friendly approach to education and allows the student to access the quiz at anywhere and anytime. Furthermore, GTSPM also includes quizzes for each SPM subject that allow the students to assess their learning progress and identify areas where they need to improve. This can help students further enhance their knowledge about the topics and encourages continuous revision to address knowledge gaps effectively. Additionally, by centralising all resources, SMK Puteri students can access the resources easily anywhere and anytime empowering them with a comprehensive tool in preparing for SPM examinations.

1.2 Objectives

The objectives of this project are to develop an online assessment web application with certain key features as follows:

1. Quiz Assessment

- To develop a web application that allows the student to assess their knowledge by taking multiple-choice quizzes.

2. Access Learning materials

- To develop a web application that allows students to view the uploaded learning materials such as PDF files or videos that are relevant to their SPM subjects.

3. Progress Tracking

- To develop a web application that allows students to track their learning progress and view their overall performance.

4. Quiz Management

- To allow the educators to manage the quiz such as creating and deleting a quiz, ensuring the quiz assessments are up to the syllabus.

5. Learning Materials Management

- To enable educators to upload and delete learning materials, providing up-to-date resources for the student to revise.

6. Student Monitoring

- To allow educators to monitor students' progress which can help in providing useful guidance for the student to keep up their studies

1.3 Project Scope

GTSPM website is catered for the needs of SMK Puteri students and educators of the school.

1.4 End-User Specifications

Target Users

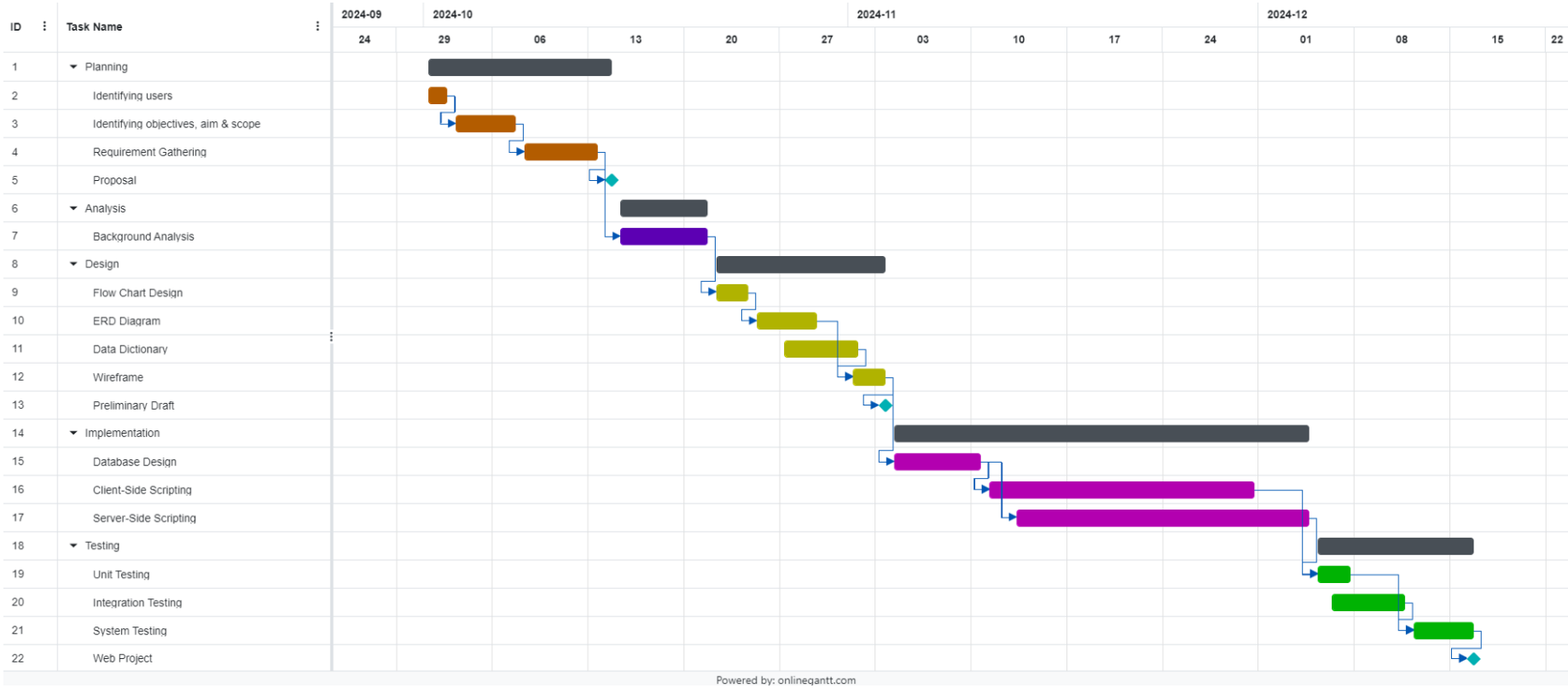
1. SPM Students (Form Five Students of SMK Puteri)
    - Students who are preparing for the Sijil Pelajaran Malaysia (SPM) examinations.
    - Require access to comprehensive, eco-friendly, and easily available learning materials to enhance their subject knowledge and exam preparation.
    - Need consistent practice to improve time management, identify knowledge gaps, and focus on potential key topics.
2. Educators (Teachers of SMK Puteri)
    - Teachers who guide students in their preparation for the SPM exams.
    - Require tools to track students' progress, assign quizzes, and identify areas where students need improvement.
    - Need centralized and accessible resources to enhance teaching efficiency.

1.5 Major Functions

The major function of GTSPM includes:

- A student portal that allows students to log in or register for a new account.
- Provide multiple choice quiz assessments on SPM subjects that the educators manage as exercises for upper-form students who preparing for SPM examinations
- A progress tracking dashboard allows the students to keep track of their performance and educators to identify the student's areas of improvement.
- A platform where all resources are centralised and can be accessed digitally such as downloading the learning materials uploaded by the educators.
- A role-based access system to ensure the platform is secured from unauthorised users.

1.6 Gantt Chart



2.0 Background Analysis & Requirement Gathering

In order to capture the requirements of our system (GTSPM) analysis must first be done in order to fully grasp and comprehend the scope of what defines a successful and ideal Online Assessment Application. To do so, we can study the general operations of existing websites which ideally is well known and used. Based on this criterion, we managed to identify a website which suit the bill which is Quizizz and Google Classroom. If more detailed analysis is to be performed, studying and analysing student projects or scholastic journals can provide us with a deeper understanding concerning the implementation of an Online Assessment Application as well as clue us into identifying best practices and potential challenges.

To begin our analysis, let us first look at the general operations of Quizizz. Quizizz can be described as a tool for online assessment particularly through the usage of user created quizzes. What makes Quizizz stand out from competitors however is its focus on facilitating a fun and gameable environment through their quizzes (Nugraha et. al., 2021). This can be seen through the utilisation of power-ups in the quiz which may grant users additional points for answering correctly, allow users to eliminate 2 incorrect choices in a Multiple-Choice Question (MCQ) in order to increase their odds of guessing correctly should the need arise, and even grant users an additional chance to correctly answer a question should they answer incorrectly (Shamil, 2024). The gamification of quizzes is further enhanced by Quizizz direct emphasis on time and quick responses. To further elaborate, each question in quiz has a predetermined time limit where the faster you answer, the more points you are awarded; essentially, the relationship between time and points are inversely proportional to one another. This emphasis of time and rewarding quick responses also introduces an element of risk-and-reward where a user rushing to answer as quickly as possible will increase their odds of answering incorrectly due to not properly analysing the question. This trend of gamifying the quiz to help facilitate a more interactive learning experience is not just isolated in Quizizz alone as other websites such as Kahoot, Quizlet and Gimkit also prides themselves on their fun learning environments.

While this current trend may have its fair share of advantages, its disadvantages cannot be ignored which are as follows (Krisbiantoro, 2021):

- Quizzes after a while can become predictable and stale if no effort is made to update or enrich gaming experience (such as introducing new power-ups).

- The gamification of the quizzes can potentially overpower the actual learning outcome of the quizzes as users may lose sight of actually learning something in favour of prioritising a high score
- While the rewards gained in the quiz can act as an external motivation to some students, it will all be in vain if students lack the internal motivation to continue studying

Besides the quizzes, Quizizz also offers a selection of learning materials mainly books or novels where users can attempt vocabulary quizzes based on the selected book. Though it is worth mentioning that this feature is generally underseen and underutilised by users which is reflected in the simplicity of the feature when compared to the complexity and intricacy seen in their quiz feature.

Moving onto Google Classroom however reveals to us that the differences between those two could not be any clearer. To elaborate, Google Classroom diverts from other websites by not just focusing on a single learning medium, but instead focuses on simulating an entire digital classroom. For example, in Google Classroom users will have the ability to:

- Upload learning materials for students to access
- Set assignments with deadlines for students to complete
- Allow students to perform quizzes through Google Forms
- Hold meetings / online classes with students through Google Meet

Furthermore, another diversion in Google Classroom can be seen in where Quizizz focuses on gamification, Google Classroom divers its focus on professionalism. The colourful and quirky UI seen in Quizizz is instead replaced with a more sterile and professional UI which emphasises usability and clarity. All in all, Google Classroom aims to capture the entire classroom experience rather than a specific element. This focus on capturing the classroom experience digitally is certainly the platforms biggest strength as many students and teachers can attest to the fact that using Google Classroom is certainly the best alternative for facilitating a complete learning environment when the physical counterpart may not be available which was especially the case when the platform saw its peak in usership during the COVID-19 pandemic.

From these two examples we can conclude that aligning our platform with Google Classroom's approach would be the better choice as it better suits our projects needs. To elaborate, GTSPM is specifically made for students preparing for their upcoming SPM examination. Therefore, gamifying the platform may cheapen the overall learning experience leading to one that is less

serious and less effective. Furthermore, our platform similarly expands beyond just providing quizzes as teachers will be able to upload learning materials for students to learn. It is important to note that while our platform may generally align with Google Classroom, that is not to say that we can't take anything from Quizizz. For example, our websites could employ a UI that is friendlier and more inviting compared to the sterility of Google Classroom; without sacrificing the usability and clarity.

Now moving onto a deeper analysis, studying scholastic journals will help us identify best practices as well as challenges seen in developing an Online Assessment Application. In doing so, the following papers were studied:

- [1] Developing a Compact and Practical Online Quiz System (Kajitori et.al., 2014)
- [2] Android based Online Quiz Application (Shoyeb, 2015)

From these two papers we could identify the following:

- Both applications contain the ability to:
    - Handle user registration [1] [2]
    - Update and set quizzes [1] [2]
    - Answer quizzes and gain instant results [1] [2]
    - Upload files for teaching purposes [1]
    - View previous results [2]

- Both applications utilise the following for development:
    - Client-Side Scripting – HTML, JavaScript, CSS
    - Server-Side Scripting – PHP
    - Database – MySQL

- Challenges seen in development:
    - Ensuring that the platform is kept simple and too complex to ensure that usability is not sacrificed in favour of arbitrary complexity [1]
    - Ensuring that the platform is easily accessible and understandable to even those that lack computer literacy [1]
    - Ensuring that the platforms UI is intuitive and effective in a mobile view [2]

Finally, it is also worth mentioning that team members also performed causal interviews with team family members that are preparing / undergoing SPM to further extract user requirements for an ideal learning application.

All in all, from the extensive analysis done, we can summarise the requirements of our system to be as follows:

Functional Requirements:

- System must be able to handle user registration
- System must allow educators to set quizzes
- System should be able to allow educators to upload learning materials
- System should allow students to access the uploaded learning materials
- System must allow students to answer quizzes and gain results
- System should allow students to view previous quiz results
- System must store all data into the database

Non-Functional Requirements:

- System should have UI that is responsive to both mobile and desktop view

User Requirements:

Students

- Student can login and register into the system
- Student can edit their user details as well as customise their profile
- Student can access learning material by filtering the materials through stream and subject choice
- Student can attempt quiz
- Student can view their previous quiz attempts

Educators

- Educator can login and register into the system
- Educator can create a quiz and specify quiz details such as allocated time
- Educator can monitor student progress on quizzes
- Educator can upload learning materials as well as specify the category of material by subject and stream

Technical Requirements:

- Client-Side Scripting (Used to facilitate user interaction and navigation) – HTML, JavaScript, CSS
- Server-Side Scripting (Used to handle the collection and gathering of data) – PHP
- Database (Used to store the data) – MySQL

// **Changes Made from Proposal / Preliminary Draft**

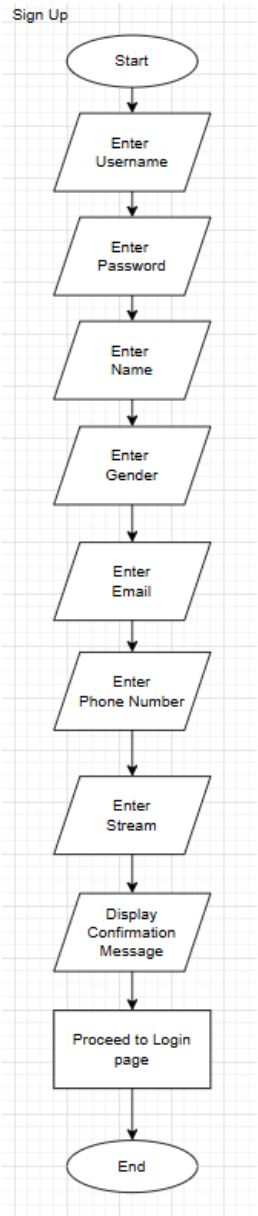
- Removed the ability for students to send questions to educators
- Removed the ability for educators to view and respond to student questions
- Removed admin role and associated functionalities entirely as we found it unnecessary

3.0 Design

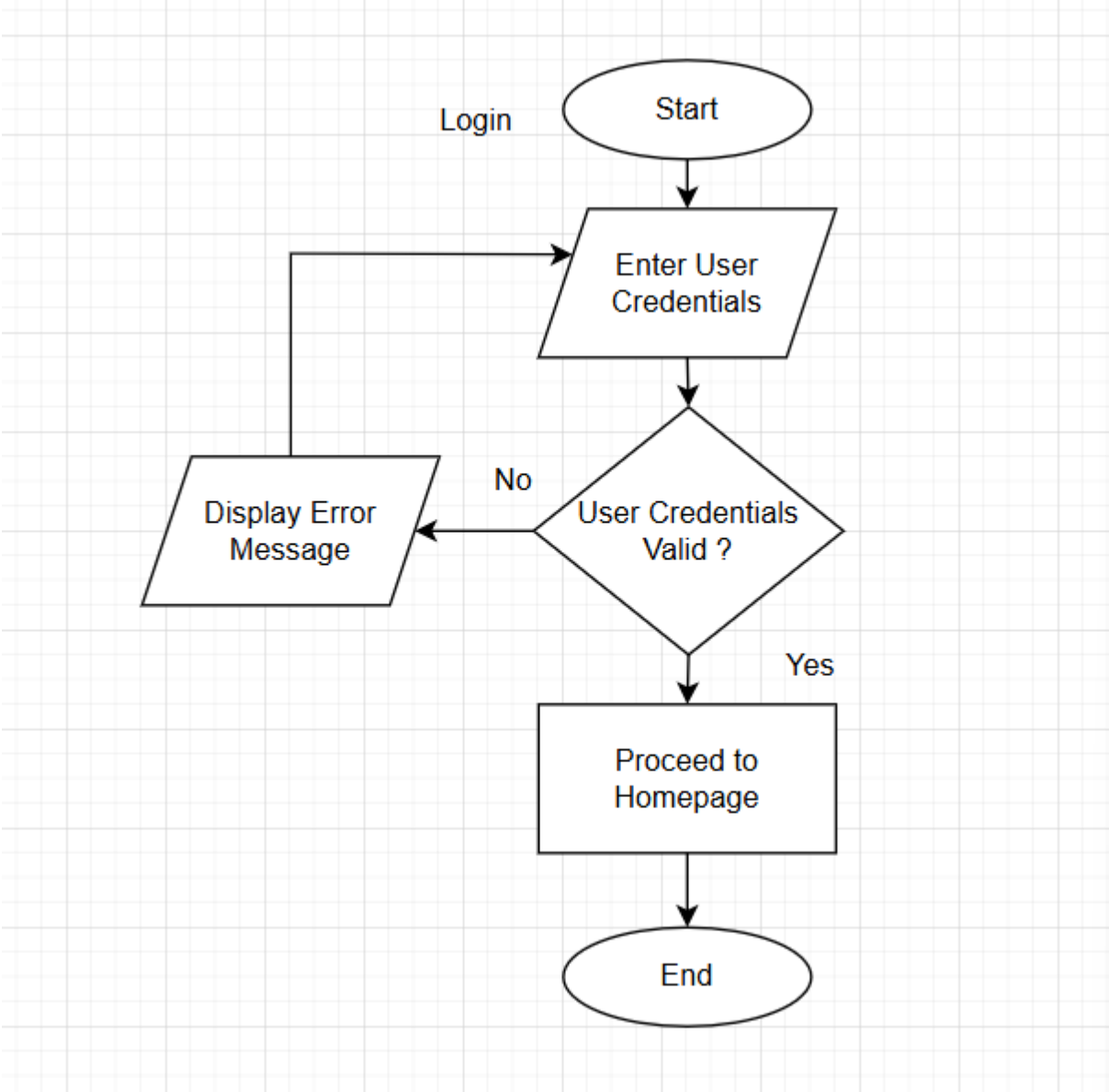
3.1 Flowchart

Student

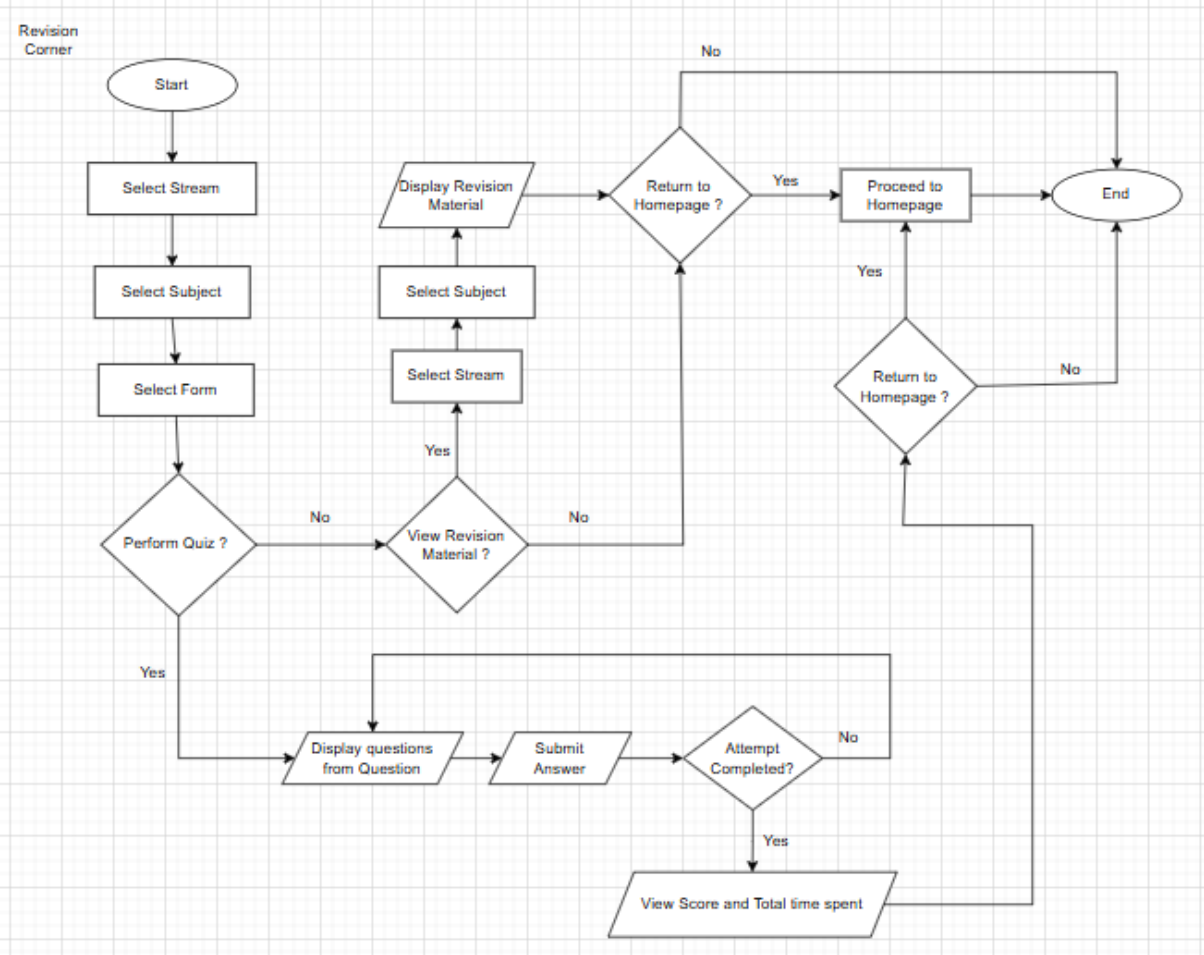
User Role	Student
Functionality Name	Sign Up process for new users
Group Member in Charge	Ng Vin Ee
Description	Create a new account by filling in details such as username, email, password, and additional information like grade and school year. Once registered, access features designed specifically for students.



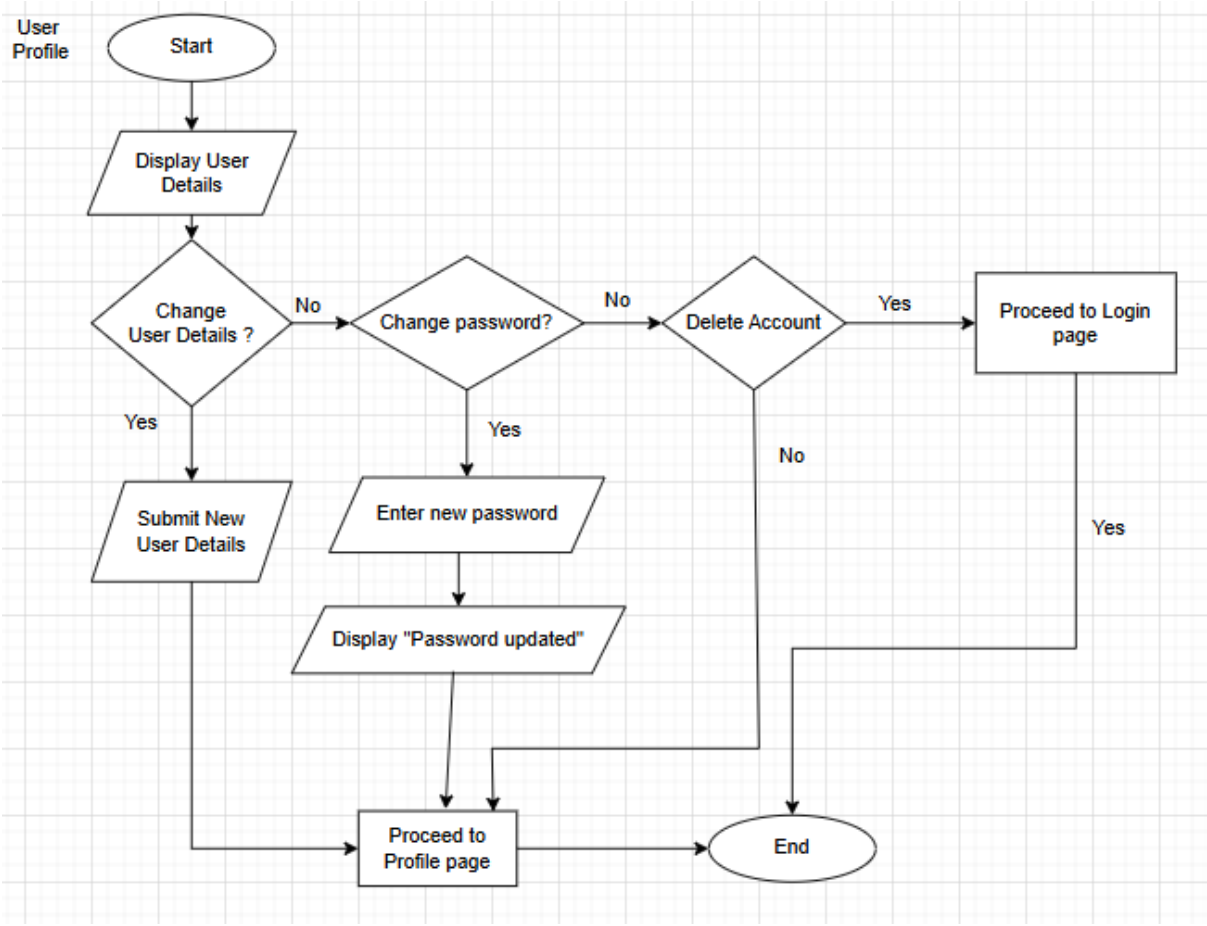
User Role	Student
Functionality Name	Login process for registered users
Group Member in Charge	Ng Vin Ee
Description	Log in using registered credentials to access quizzes, study materials. After authentication, user will land on the homepage.



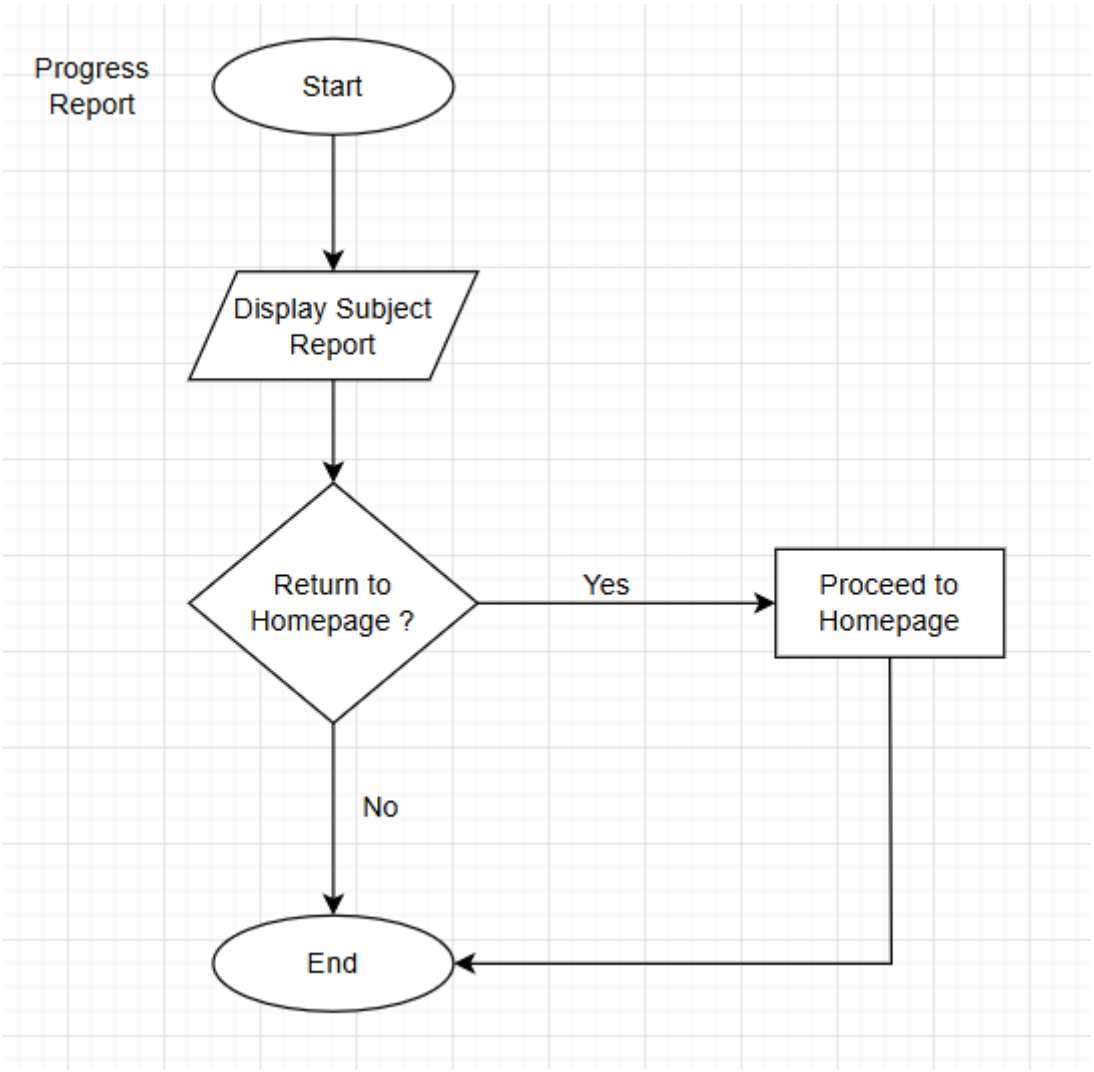
User Role	Student
Functionality Name	Revision Corner where users will be able to access quizzes as well as study materials for the subject and chapter of their liking
Group Member in Charge	Ejjaz Hakimi bin Mohamad Azan
Description	Browse and access subject-specific study materials and quizzes. Look into chapters and topics to prepare for exams or self-assess understanding.



User Role	Student
Functionality Name	User Profile where users will be able to view and modify their user details as well as switch accounts / log out
Group Member in Charge	Suchitra Nambiar A/P Mahandran
Description	View personal information, such as grade, stream, and account details. Easily update or modify the profile, change the password, or switch between accounts if needed.

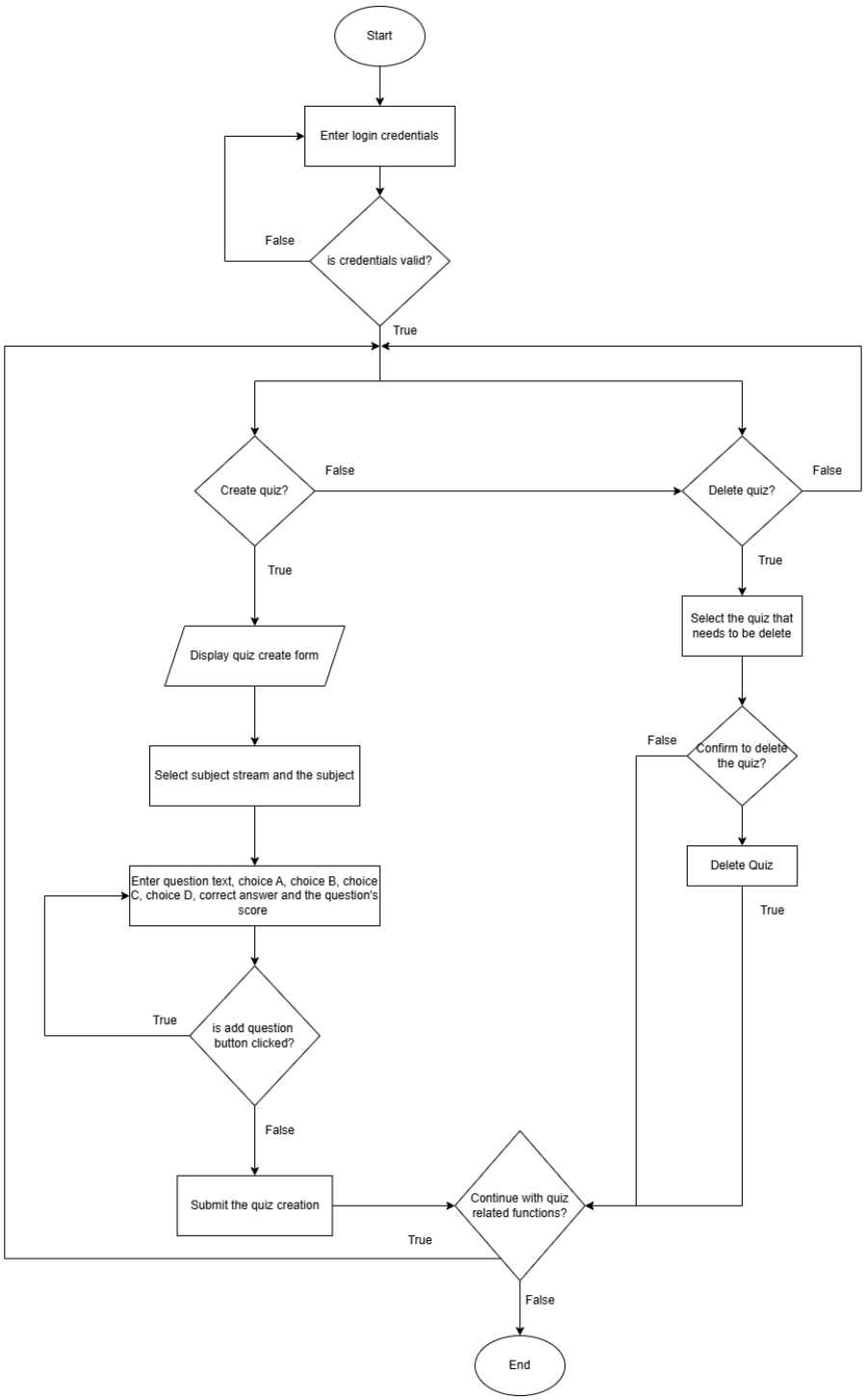


User Role	Student
Functionality Name	Progress Report where users can view various reports such as their completion percentage, overall grades and scores
Group Member in Charge	Ng Vin Ee
Description	Keep track of overall progress, including quiz completion rates, accuracy, and scores, to monitor academic growth over time.

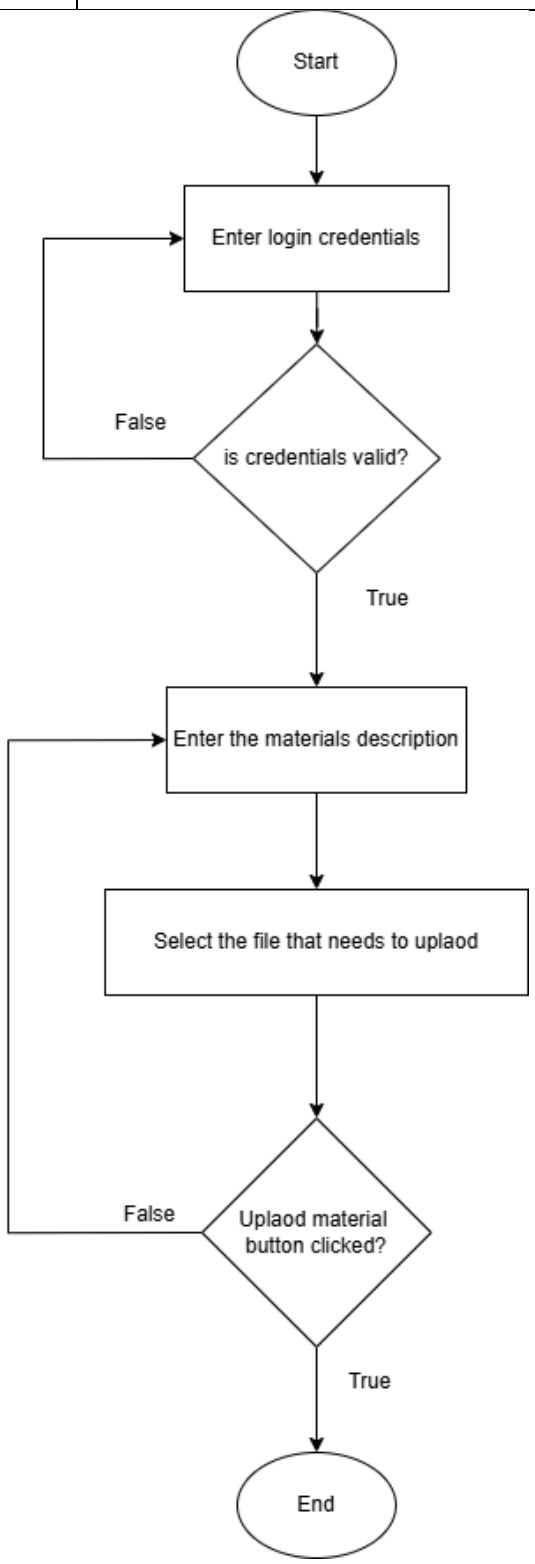


Educator

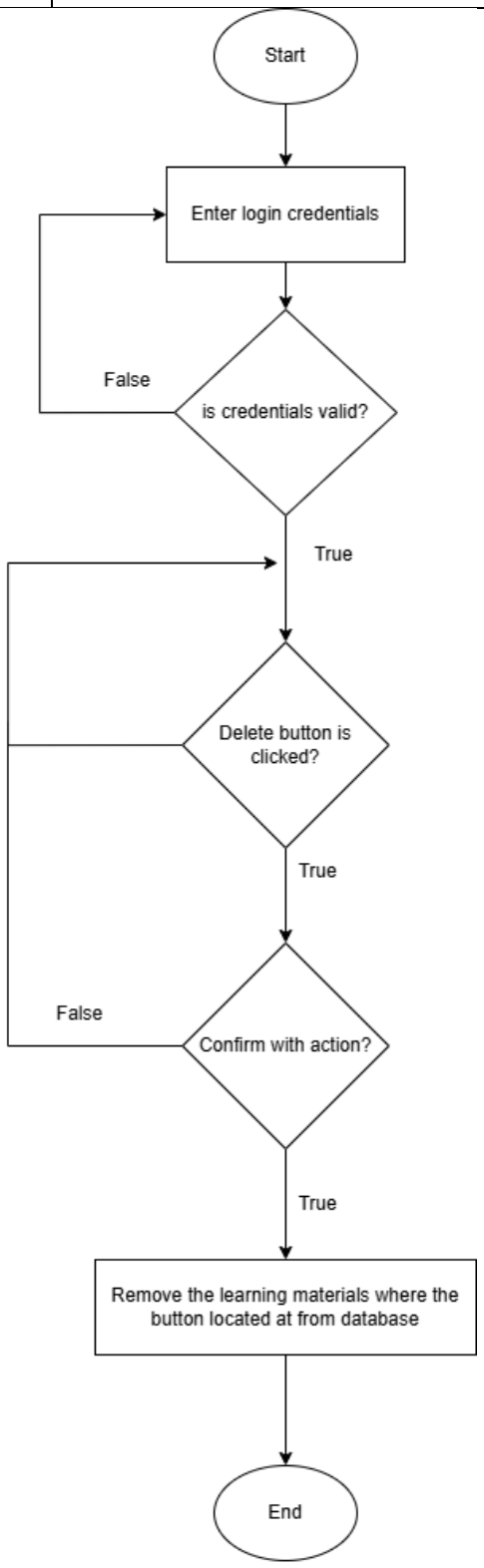
User Role	Educator
Functionality Name	Create new quiz and delete quiz
Group Member in Charge	Kwan Chun Hoe
Description	Create new quiz by selecting the subject and fill in the questions or delete an existing quiz



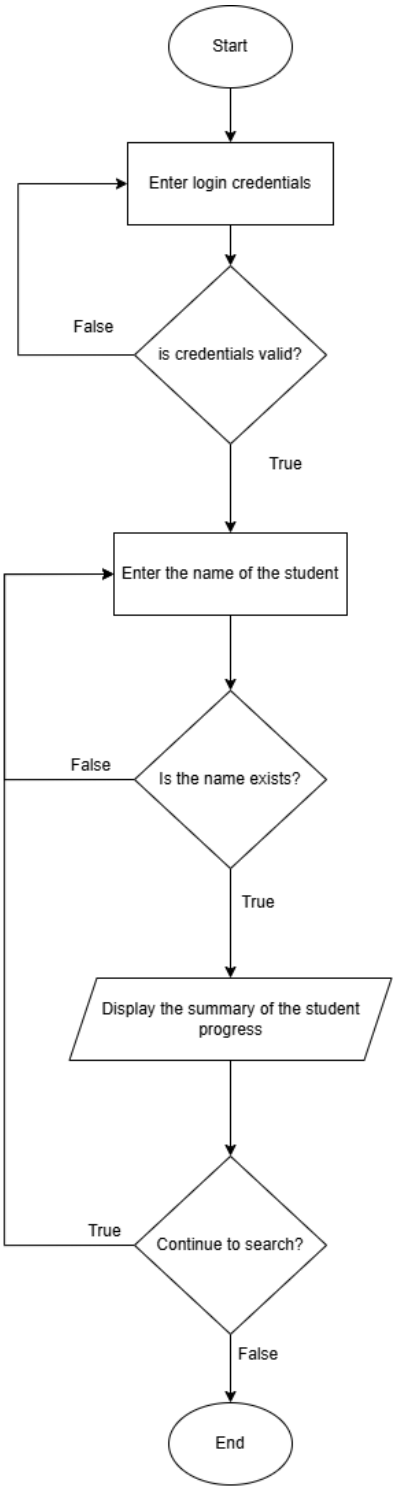
User Role	Educator
Functionality Name	Upload Learning Materials
Group Member in Charge	Kwan Chun Hoe
Description	Educator can upload the learning materials for the students from their computer



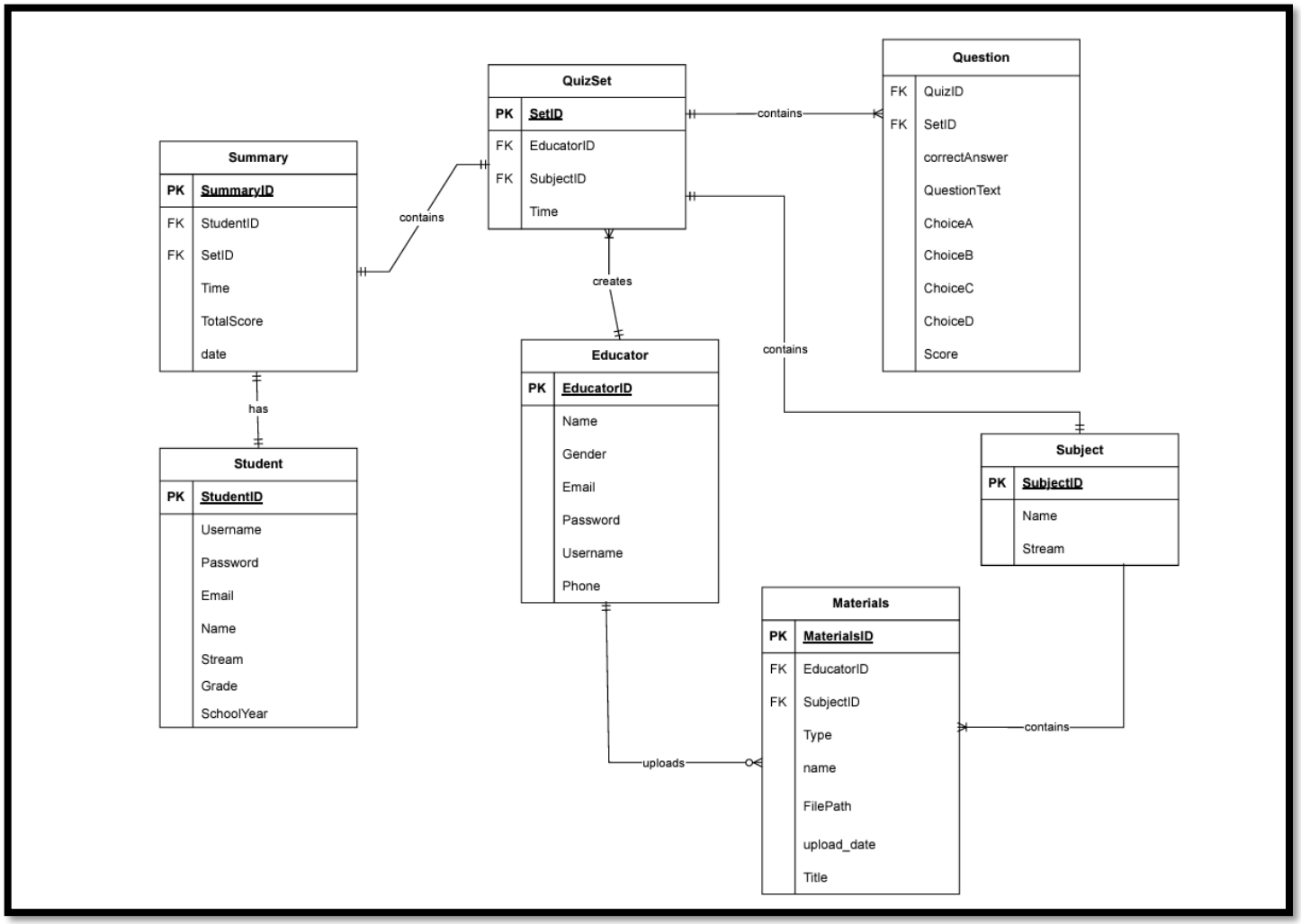
User Role	Educator
Functionality Name	Delete Learning Materials
Group Member in Charge	Kwan Chun Hoe
Description	Educator can delete the learning materials that is outdated



User Role	Educator
Functionality Name	Track Student Progress
Group Member in Charge	Kwan Chun Hoe
Description	Educator can track student's progress such as their time taken for the quiz and the points they scored



3.2 Entity Relationship Diagram (ERD)



1. Educator and QuizSet

- **Relationship**: One-to-Many
- **Description**: An Educator can create multiple QuizSets. Each QuizSet is linked to a specific Educator through the EducatorID foreign key.

2. QuizSet and Question

- **Relationship**: One-to-Many
- **Description**: A QuizSet contains multiple Questions. Each Question references a QuizSet using the SetID foreign key.

3. Student and Summary

- **Relationship**: One-to-One
- **Description**: A Student can will have one summary. The Summary table references a Student using the StudentID foreign key.

4. QuizSet and Summary

- **Relationship**: One-to-One
- **Description**: Each **QuizSet** will contain one summary that stores student performance on the quiz. The SetID in the Summary table acts as a foreign key.

5. Educator and Materials

- **Relationship**: **One-to-Many**
- **Description**: An **Educator** can zero or many materials. The EducatorID foreign key in the Materials table establishes this link.

6. Subject and QuizSet

- **Relationship**: One-to-One
- **Description**: Each **Subject** contains a QuizSet. The SubjectID foreign key in the QuizSet table references the Subject table.

7. Subject and Materials

- **Relationship**: One-to-Many
- **Description**: A **Subject** can contain multiple **Materials**. The SubjectID foreign key in the Materials table connects each material to its respective subject.

//Changes Made from **Proposal / Preliminary Draft**

- SUBJECTCHAPTER table was dropped to more streamline the quiz experience by encapsulating the entire subject into a single quiz rather than multiple
- ADMINFORUM, REPLYBOARD & QUESTIONBOARD tables were dropped due to our decision to forgo the feature
- HISTORY was dropped as we determined that it was redundant and unnecessary as it can be combined with SUMMARY
- ADMINISTRATOR was table was dropped because we decided the role to be unnecessary

3.3 Data Dictionary

Member that responsible	Entity	Attribute	Data Type	Length	Description	Example input
Ejjaz Hakimi bin Mohamad Azan	STUDENT	StudentID	String	12	Primary Key, unique identifier for each student	STU123
		Username	String	50	Username for student login	Slayer25
		Password	String	255	Password for student login	Sgtz35@4
		Email	String	100	Email address of the student	Fiona12@gmail.com
		Name	string	50	Name to address student	Jonathan
		Stream	string	10	Type of stream of student	Arts stream
		Grade	int	2	The grade of student	4
		SchoolYear	int	4	The student school year	2024
Kwan Chun Hoe	EDUCATOR	EducatorID	string	12	Primary key, unique identifier for each educator	ED0001
		Name	string	50	Name of the student	Mary
		Gender	string	50	The gender of the educator	Female
		Username	string	50	Username for educator to login	mary@56
		Password	string	50	Password for educator to login	MaRy1656
		Email	string	50	Email of educator	mary123@gmail.com
		Phone	string	50	Phone number of educator	012-3721128

Suchitra Nambiar A/P Mahandran	SUBJECT	SubjectID	int	N/A	Primary key, unique identifier or each subject	SUB172
		Name	string	50	Name of the subject	Biology
		Stream	string	50	Academic stream of the subject	Science
Ejjaz Hakimi bin Mohammad Azan	QUIZSET	SetID	int	10	Primary key, unique identifier for each quiz	ST4578
		EducatorID	string	12	Foreign key, references EducatorID in EDUCATOR	ED1
		SubjectID	int	N/A	Foreign key, references SubjectID in SUBJECT	CHP1
		Time	time	N/A	Time taken to answer the quiz	01:20:02
Suchitra Nambiar A/P Mahandran	QUESTION	QuizID	int	N/A	Primary key, unique identifier for each quiz question	QQ5
		SetID	int	N/A	Foreign key, references SetID in QUIZSET	ST4578
		QuestionText	string	255	Text of the question	What is chlorophyll?
		ChoiceA	string	255	Text for choice A	Pigment of plant that absorbs sunlight
		ChoiceB	string	255	Text for choice B	Protein that helps plant grow
		ChoiceC	string	255	Text for choice C	Essential mineral for plant growth
		ChoiceD	string	255	Text for choice D	Gas that plant release to atmosphere

		CorrectAnswer	string	50	The correct answer of a question	Pigment of plant that absorbs sunlight
		score	string	255	Score that student can obtained from a question	20%
Elianna Catrina Herrera	MATERIALS	MaterialsID	int	100	Primary key, unique identifier for each material	MT156
		EducatorID	string	12	Foreign key, references EducatorID in EDUCATOR	ED0001
		SubjectID	int	N/A	Foreign key, references SubjectID in SUBJECTCHAPTER	CHP1
		Type	string	100	The type of the materials	video
		Title	string	255	The learning materials description	Video about how microorganisms move.
		name	string	255	The access link to the material	www.googledrive.com
		upload_date	timestamp	N/A	The date and time when the material uploads	2024-12-16 14:06:26
		FilePath	string	255	Shows the location of the file stored	uploads/gtspm.pdf
Ng Vin Ee	SUMMARY	SummaryID	int	12	Primary key, unique identifier for each summary	HIS1
		StudentID	string	10	Foreign key, references StudentID in STUDENTS	80%
		SetID	int	5	Foreign key, references SetID in QUIZSET	95%
		Time	time	N/A	Students can view the total time taken to attempt a quiz.	14:06:26

		TotalScore	string	255	Students can view the total score after attempting a quiz.	100
		date	date	N/A	Students can view the date they have attempted the quiz.	2024-12-16

1) **STUDENT**:

- Attributes like StudentID, Username, Password, Email, and more define unique identifiers, login details, and personal information.

2) **EDUCATOR**:

- Includes EducatorID, Name, Username, and Email, specifying login credentials and contact information for educators.

3) **ADMINISTRATOR**:

- Attributes include AdminID, Name, and Gender, focusing on admin-specific identifiers and login details.

4) **SUBJECT**:

- Defined by attributes like SubjectID and Stream, identifying subjects and their academic stream.

5) **QUIZSET**:

- Manages quizzes with SetID, Time, and foreign keys referencing educators and subjects.

6) **QUESTION**:

- Attributes include QuizID, QuestionText, Choices, and CorrectAnswer, detailing quiz content and scoring.

7) **MATERIALS**:

- Contains MaterialsID, Type, Title, and FilePath, handling learning resources for subjects.

8) **SUMMARY**:

- Attributes like SummaryID, TotalScore, and Time provide quiz performance metrics for students.

//Changes Made from **Proposal / Preliminary Draft**

- TotalScore, Time & date were added to summary to more accurately capture user quiz attempts
- Name, FilePath, UploadDate & Title were added to material to more accurately define the material uploaded to the database
- Time was added to quizset to allow educators to set the allocated time for quizzes

3.4 Wireframe

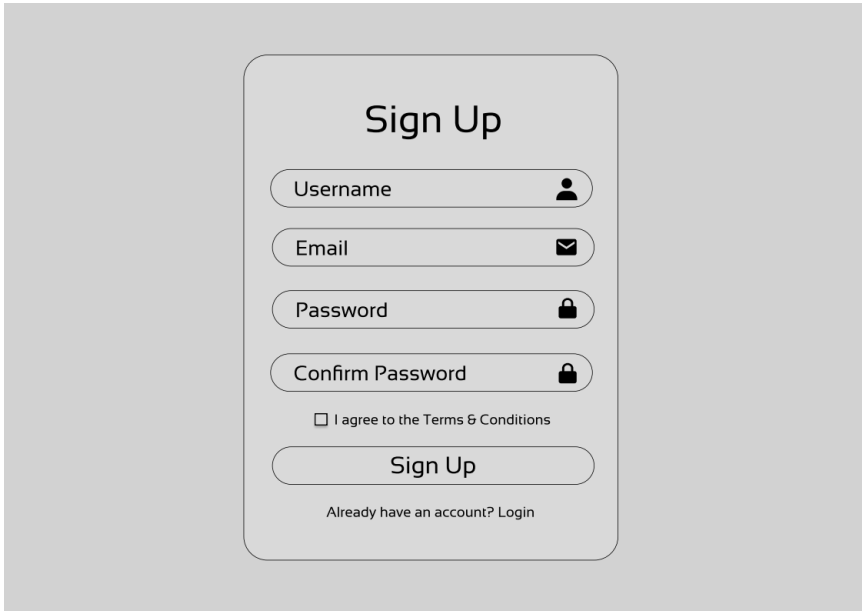
User role	Student, Educator, and Administrator
Functionality name	Log in
Group member in charge	Ng Vin Ee
Table from the database that produces output or process the form input	STUDENT, EDUCATOR, ADMINISTRATOR (username and password from these three are needed to verify their role and credentials)
Description	Sign in to the system using your username and password. This is to access to the right features based on your role like student, educator or administrator.

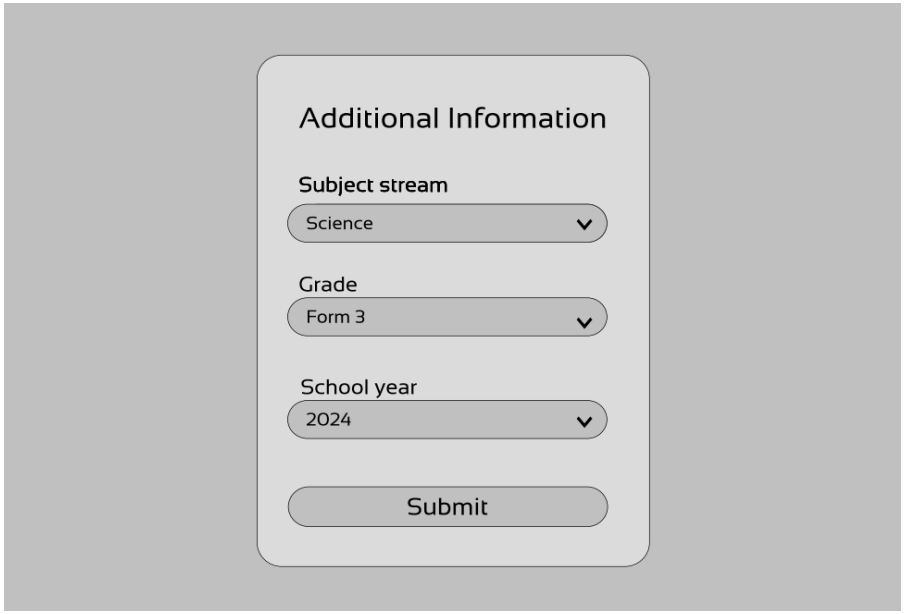
Desktop

Mobile

User role	Student
Functionality name	Sign up
Group member in charge	Ng Vin Ee
Table from the database that produces output, process form input	STUDENT (Username, email, password, stream, grade, schoolyear is needed to sign up)
Description	Create a new account by providing your username, email, password, grade, school year, and stream. This will allow you to begin using the platform for your studies.

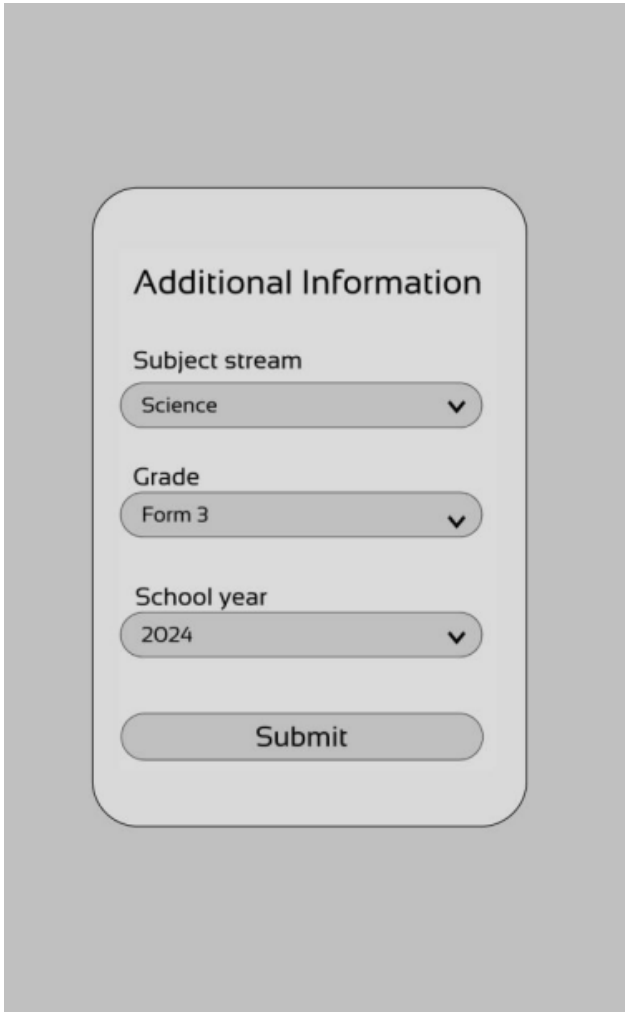
Desktop





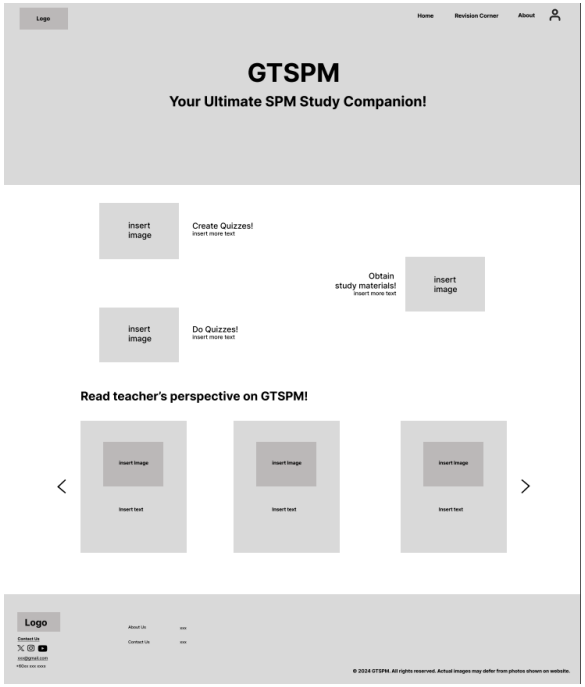
Mobile


Sign Up
Username
Email
Password
Confirm Password
I agree to the Terms & Conditions
Sign Up
Already have an account? Login

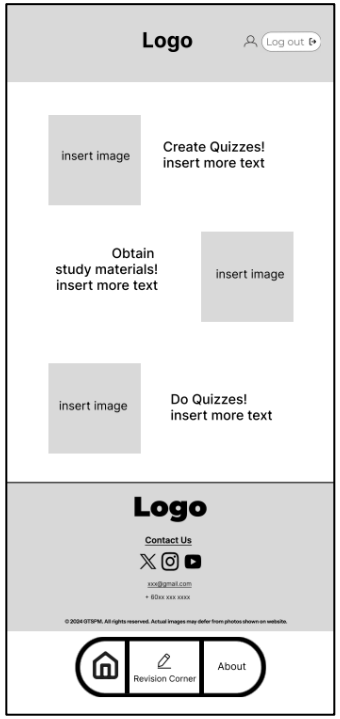

Additional Information
Subject stream
Science
Grade
Form 3
School year
2024
Submit

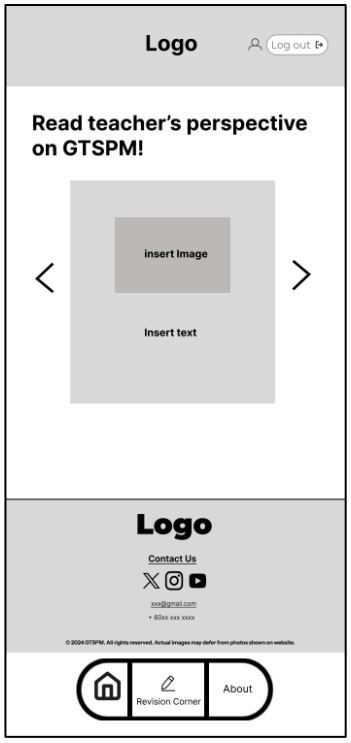
User role	Student
Functionality name	Homepage
Group member in charge	Elianna
Table from the database that produces output, process form input	-
Description	Once logged in, the homepage is your starting point where you can easily access quizzes.Students can study materials and track overall progress.

Desktop



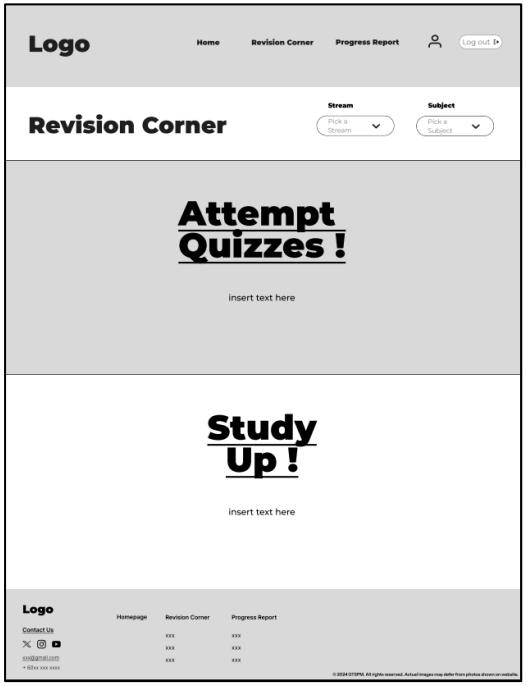
Mobile



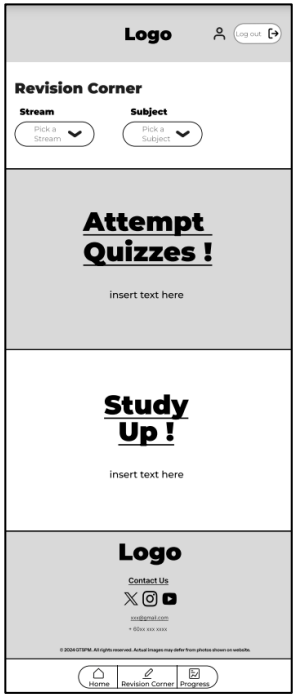


User role	Student
Functionality name	Revision corner (Main Page)
Group member in charge	Ejjaz Hakimi bin Mohamad Azan
Table from the database that produces output, process form input	-
Description	This is where students can access quizzes and study materials, organized by subjects and chapters for easy navigation. Students can use the drop down text to find what they need.

Desktop

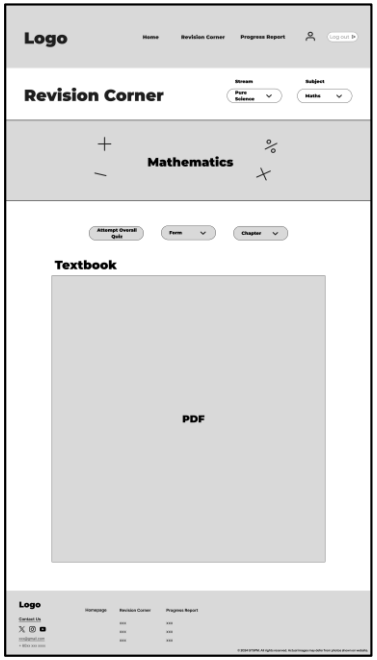


Mobile

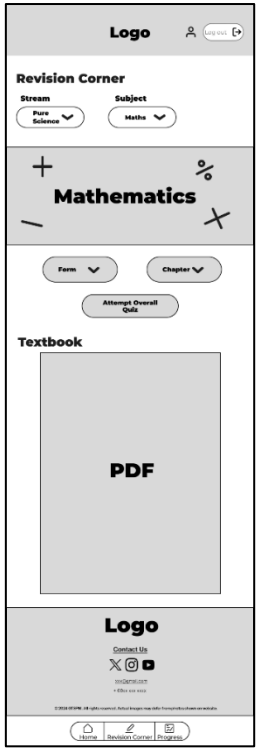


User role	Student
Functionality name	Revision corner (Select Subject)
Group member in charge	Ejjaz Hakimi bin Mohamad Azan
Table from the database that produces output, process form input	SUBJECT (to display the subject choice and subject stream)
Description	Choose the subject that student would like to focus on with options tailored to their grade and stream. Textbook is viewable in this page in a pdf format.

Desktop

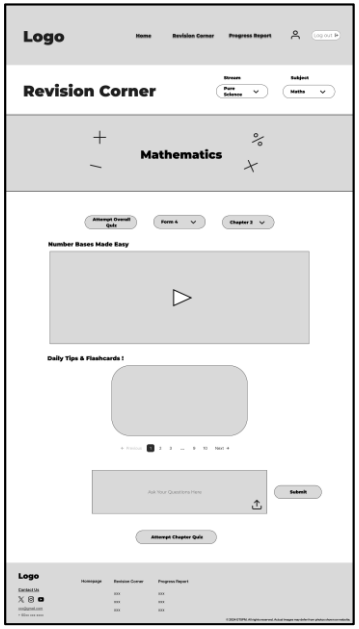


Mobile

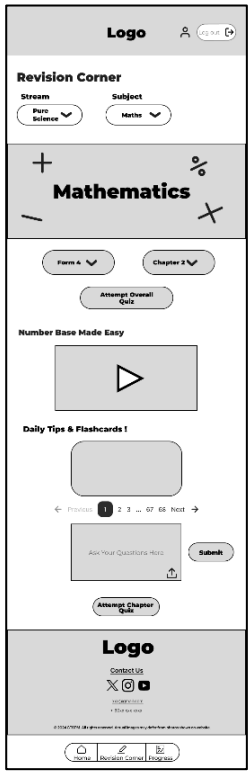


User Role	Student
Functionality Name	Revision Corner (Chapter Select)
Group Member in Charge	Ejjaz Hakimi bin Mohamad Azan
Table from the database that produces output, process form input	SUBEJCTCHAPTER (to display grade and chapter choices), QUIZSET (to display different quiz that had been created)
Description	After selecting a subject, pick a chapter to view related content is a feature added. Next, student can also take quizzes specific to that chapter.

Desktop

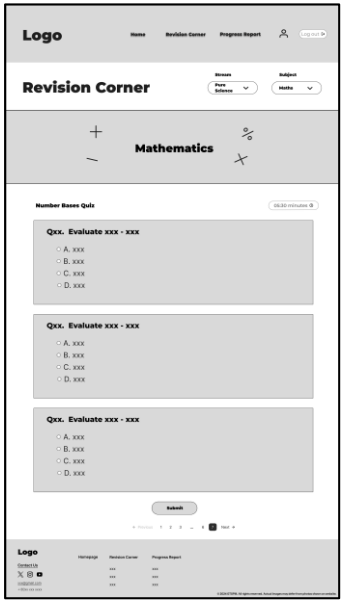


Mobile

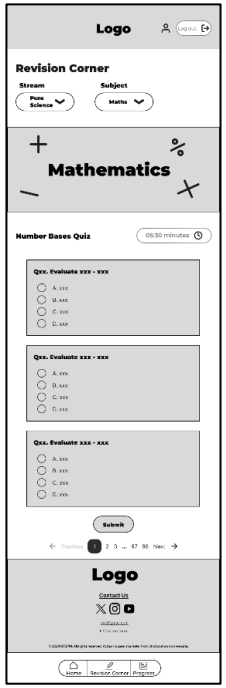


User Role	Student
Functionality Name	Revision Corner (Quiz)
Group Member in Charge	Ejjaz Hakimi bin Mohamad Azan
Table from the database that produces output, process form input	QUESTION (to display the question in the selected quiz)
Description	Take a quiz on the chapter that the student selected. It helps with understanding and improving student knowledge.

Desktop



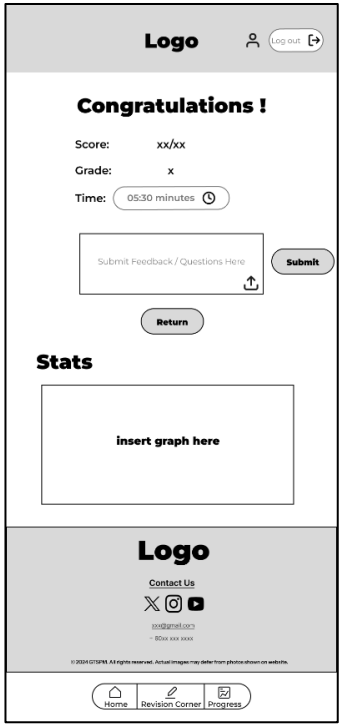
Mobile



User Role	Student
Functionality Name	Revision Corner (Quiz Results)
Group Member in Charge	Ejjaz Hakimi bin Mohamad Azan
Table from the database that produces output, process form input	QUESTIONBOARD (to insert a data entry when the student leave a question)
Description	After completing a quiz, review your results to see your scores and areas for improvement. Students can also submit feedbacks.

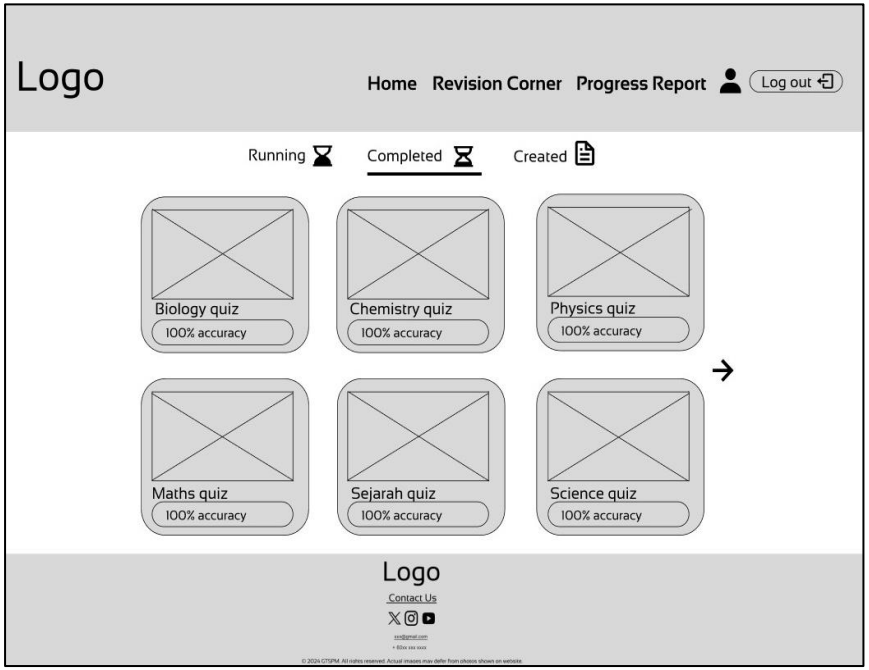
Desktop

Mobile

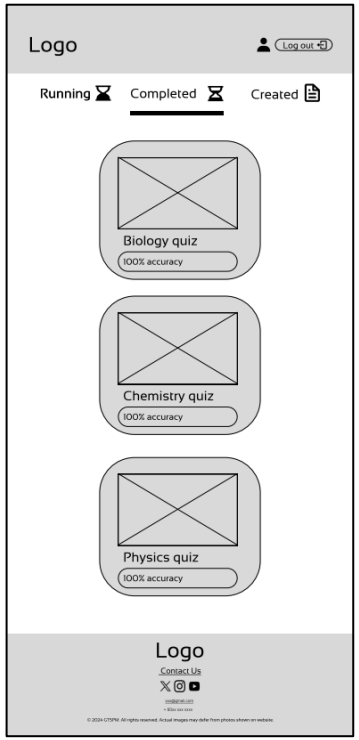


User role	Student
Functionality name	Progress report
Group member in charge	Ng Vin Ee
Table from the database that produces output, process form input	SUMMARY (Progress and accuracy is to track student progress of the quiz)
Description	View detailed reports of your progress, including completed quizzes, grade and overall accuracy to track how student is advancing.

Desktop

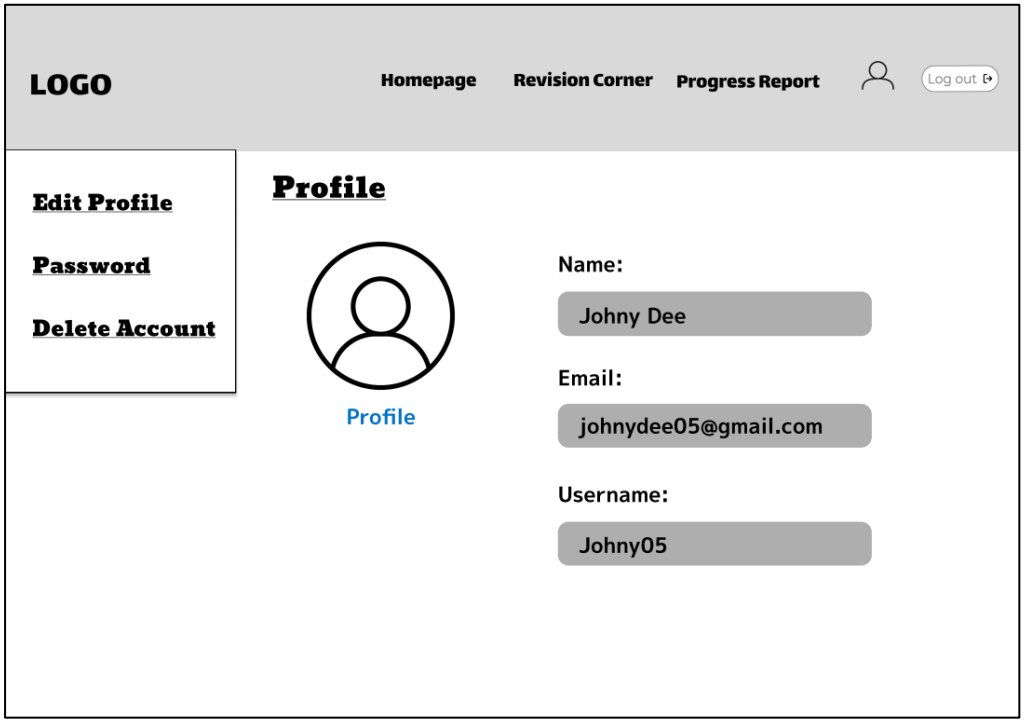


Mobile



User role	Student
Functionality name	User profile (current profile)
Group member in charge	Suchitra Nambiar A/P Mahandran
Table from the database that produces output, process form input	STUDENT (to display user profile details on the profile page)
Description	View current profile details.

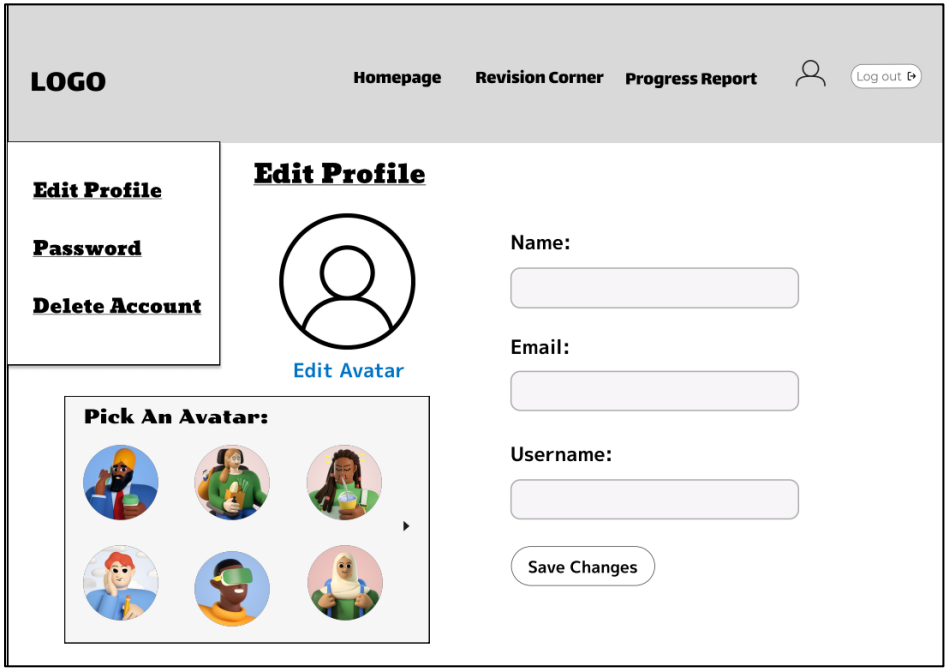
Desktop



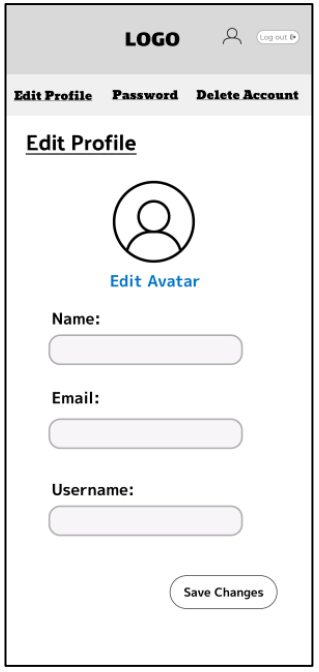
Mobile

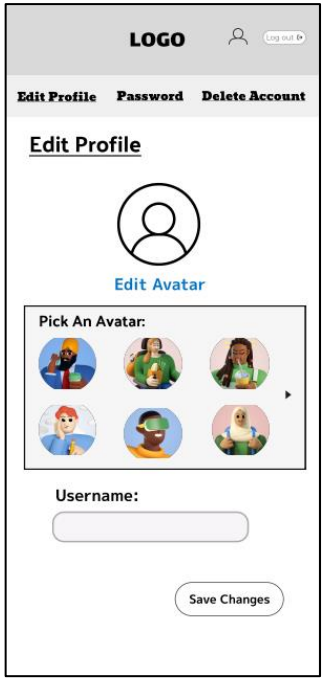
User role	Student
Functionality name	User profile (edit profile)
Group member in charge	Suchitra Nambiar A/P Mahandran
Table from the database that produces output, process form input	STUDENT (to edit user profile details on the profile page)
Description	Edit personal profile which includes all the information entered such as your name, email and username. Student can also edit their profile by picking the avatars provided under the profile picture.

Desktop



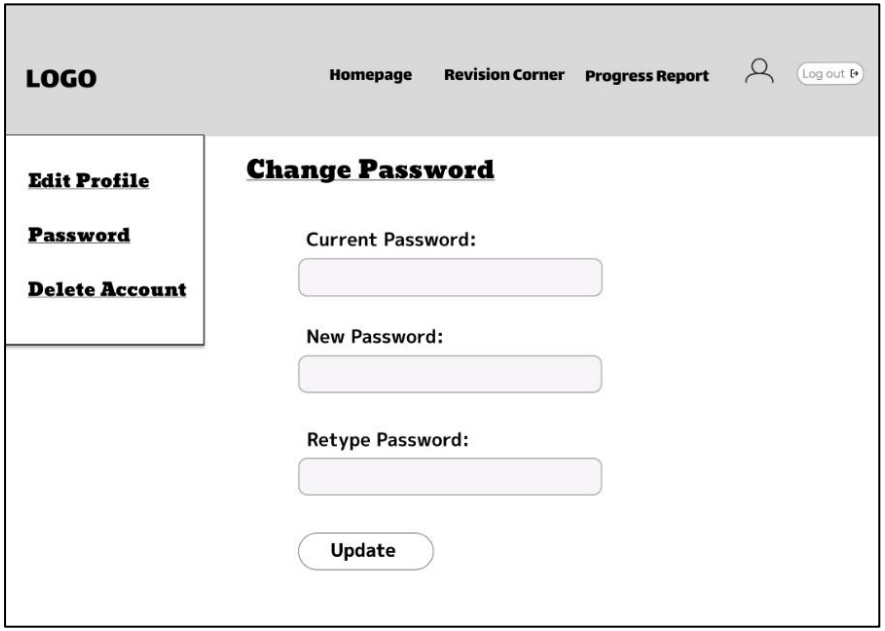
Mobile





User role	Student
Functionality name	User profile (change password)
Group member in charge	Suchitra Nambiar A/P Mahandran
Table from the database that produces output, process form input	STUDENT (to verify current password matched the database and update to new password if validation succeeds)
Description	Change your password by entering your current password and then updating it to a new one for security purposes.

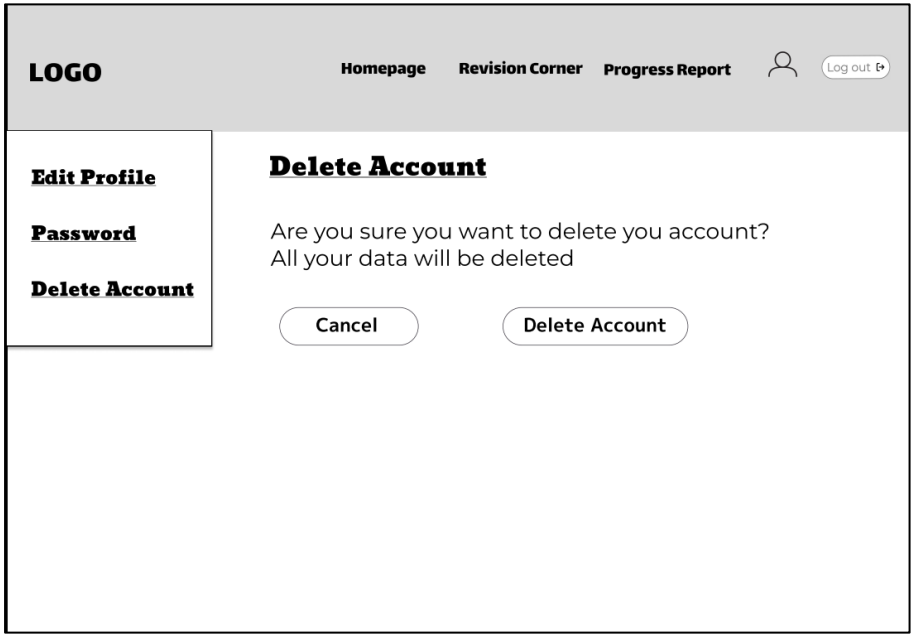
Desktop



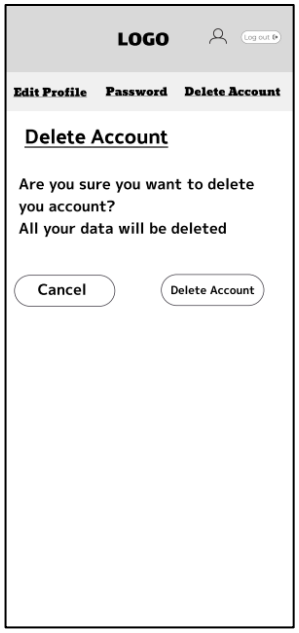
Mobile

User role	Student
Functionality name	User profile (delete account)
Group member in charge	Suchitra Nambiar A/P Mahandran
Table from the database that produces output, process form input	STUDENT (to delete user profile details on the profile page)
Description	If student decide to leave the platform, student can delete your account. Any personal information will be removed from the system.

Desktop

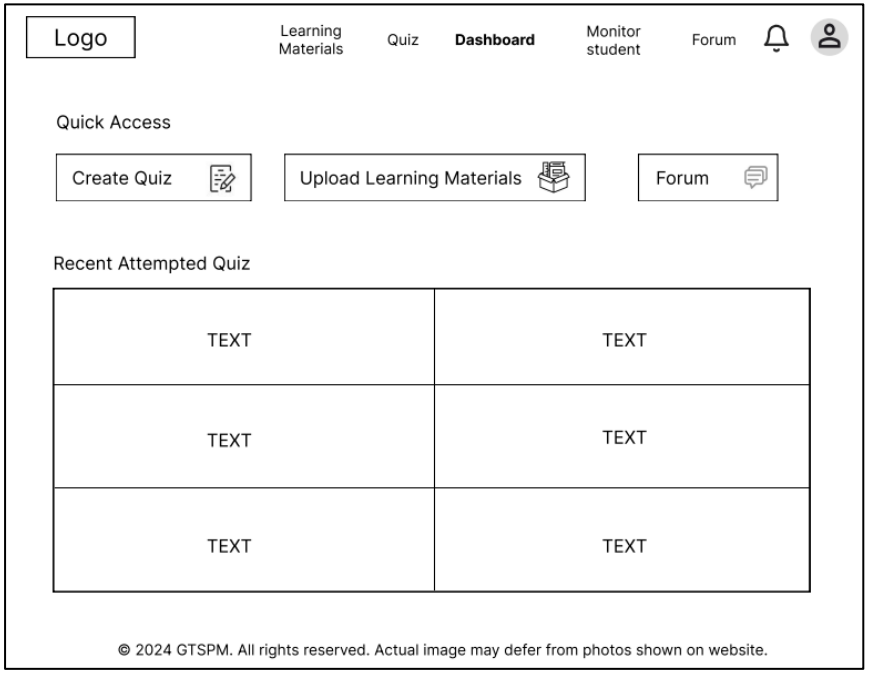


Mobile

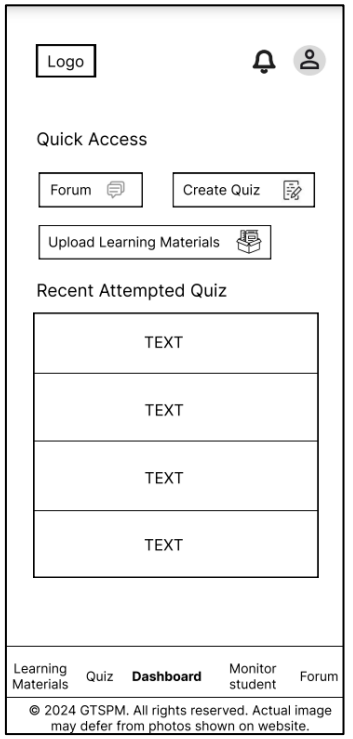


User role	Educator
Functionality name	Homepage
Group member in charge	Kwan Chun Hoe
Table from the database that produces output, process form input	HISTORY (to get the record for recent attempted quiz)
Description	The educator's homepage shows recent quizzes, student progress, and allows easy access to teaching materials and tools for managing classes.

Desktop

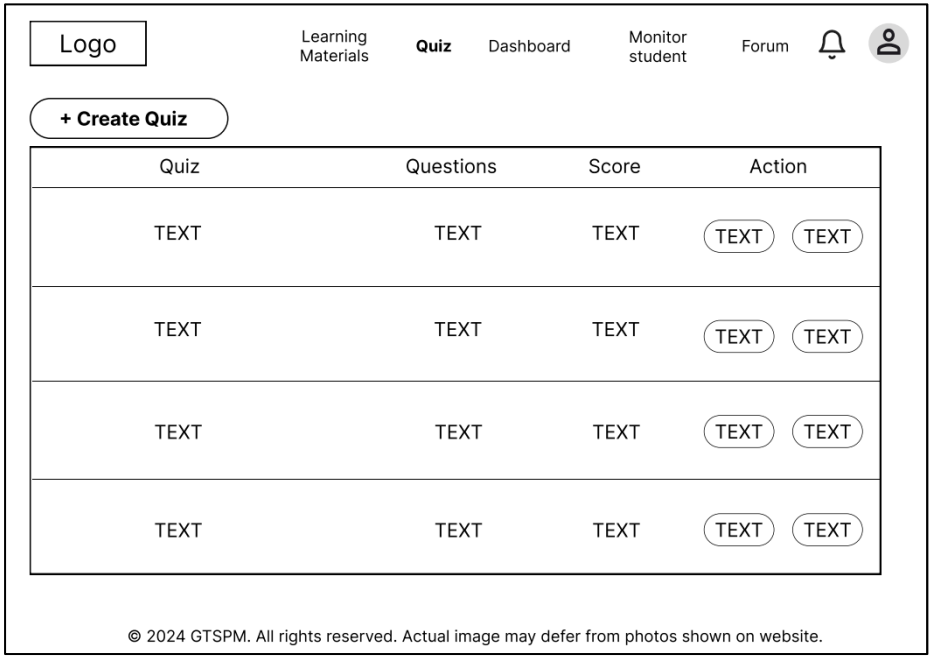


Mobile

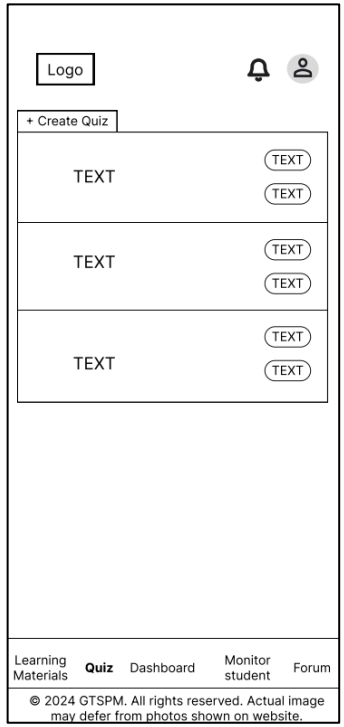


User role	Educator
Functionality name	Quiz dashboard
Group member in charge	Kwan Chun Hoe
Table from the database that produces output, process form input	QUIZSET (to get the information about the quiz that needs to be displayed)
Description	View, edit, and manage your created quizzes here. You can also delete outdated quizzes or create new ones to share with your students.

Desktop

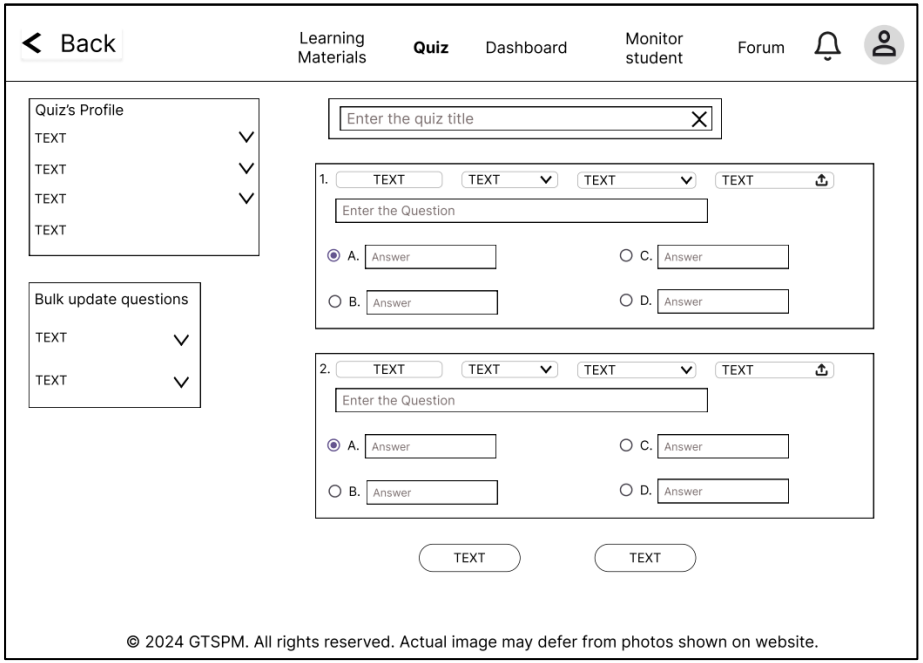


Mobile

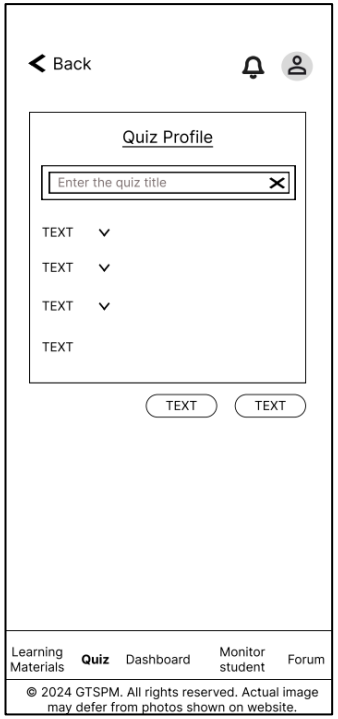


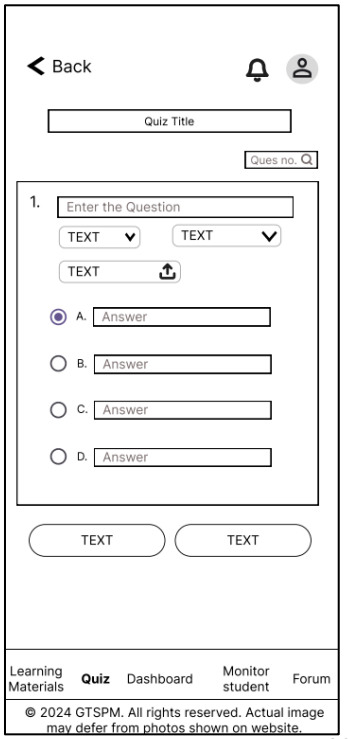
User role	Educator
Functionality name	Create quiz
Group member in charge	Kwan Chun Hoe
Table from the database that produces output, process form input	QUIZSET (to insert new data entry for quiz creation), QUESTION (to insert new entry for every question created)
Description	Create custom quizzes by adding questions, setting time limits, and tailoring them to the topics you teach. This helps engage students in meaningful assessments.

Desktop



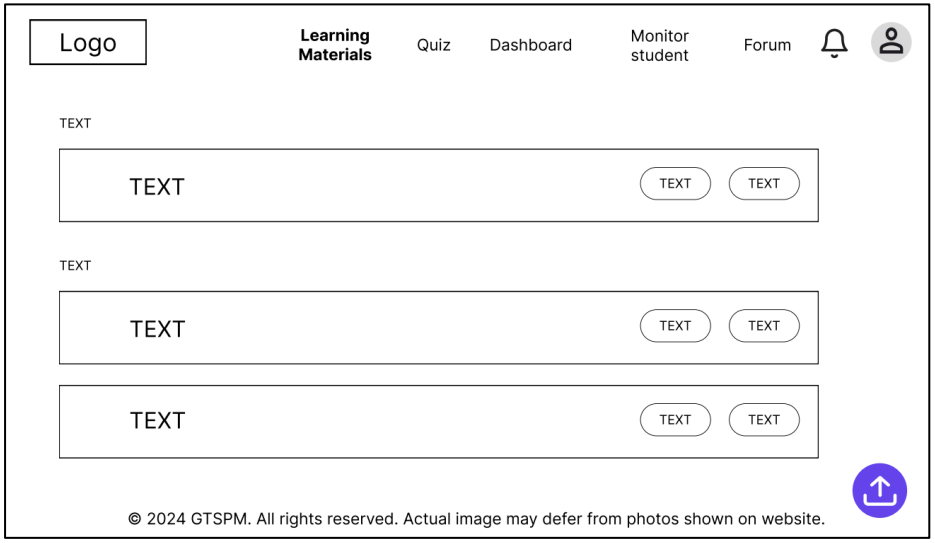
Mobile



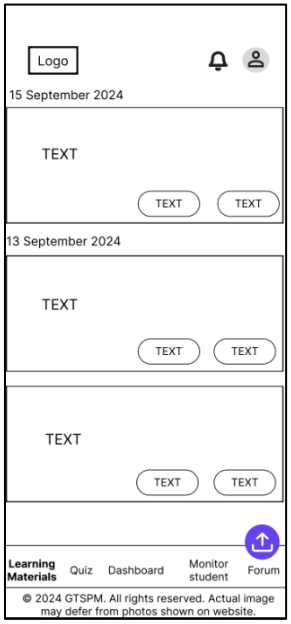


User role	Educator
Functionality name	Learning material dashboard
Group member in charge	Kwan Chun Hoe
Table from the database that produces output, process form input	MATERIALS (to get the information that needs to be displayed)
Description	This page displays the learning materials you've uploaded, giving you an overview of the content available to your students.

Desktop

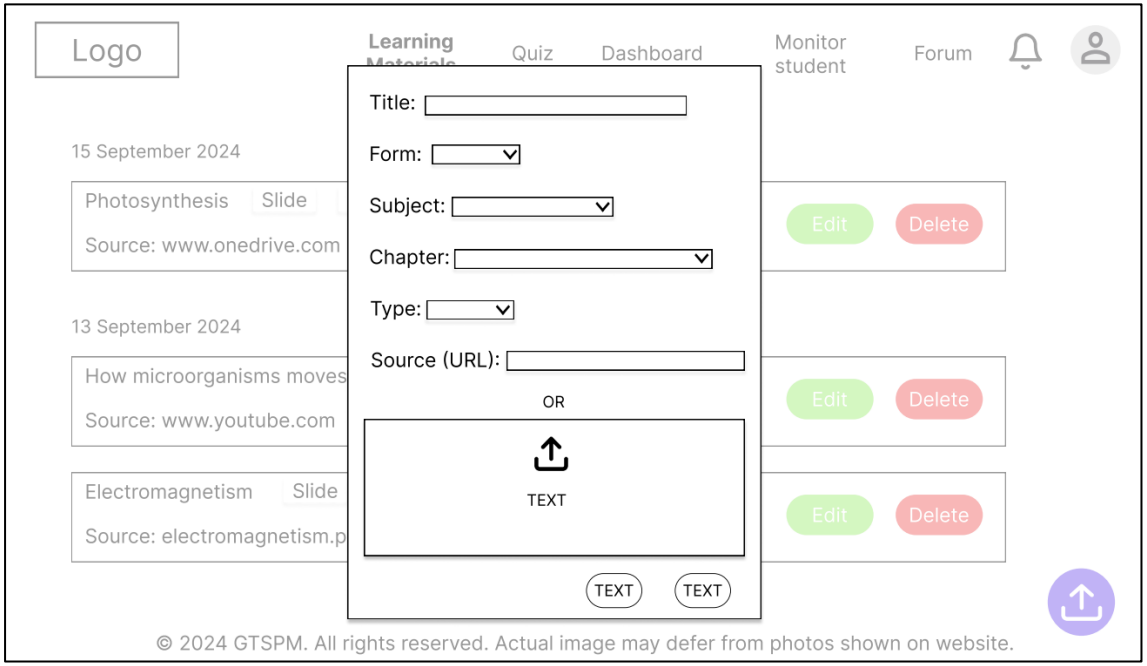


Mobile

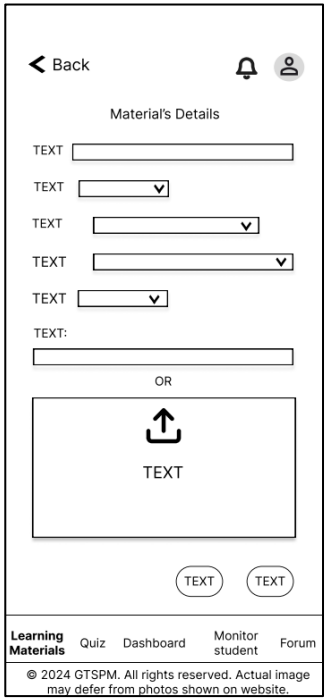


User role	Educator
Functionality name	Upload learning material
Group member in charge	Kwan Chun Hoe
Table from the database that produces output, process form input	MATERIALS (to insert new data entry when a new material is uploaded)
Description	Upload new study materials like textbooks, articles, or videos for students so that they can access and use them when they are learning.

Desktop

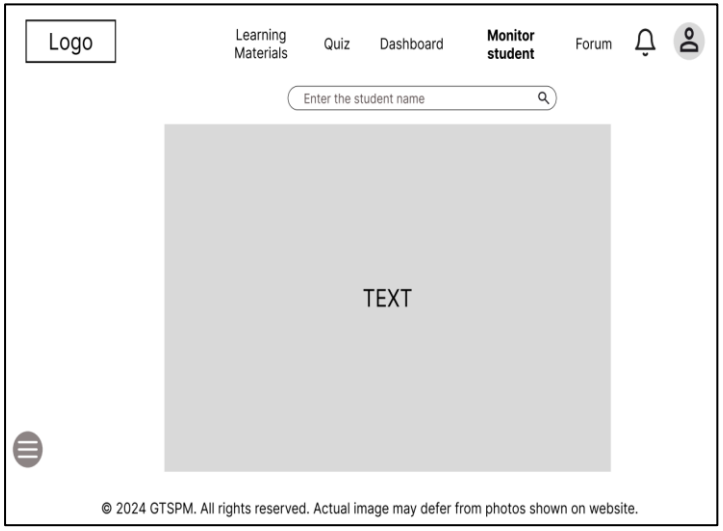


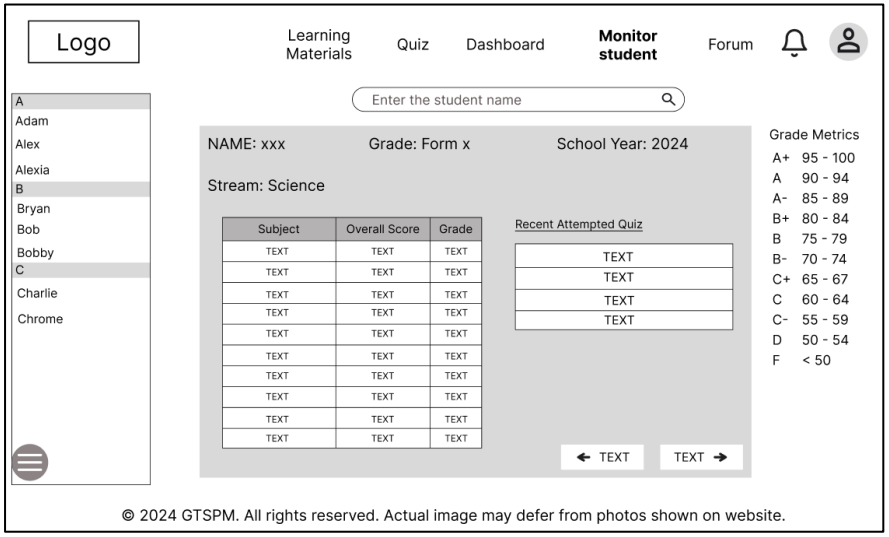
Mobile



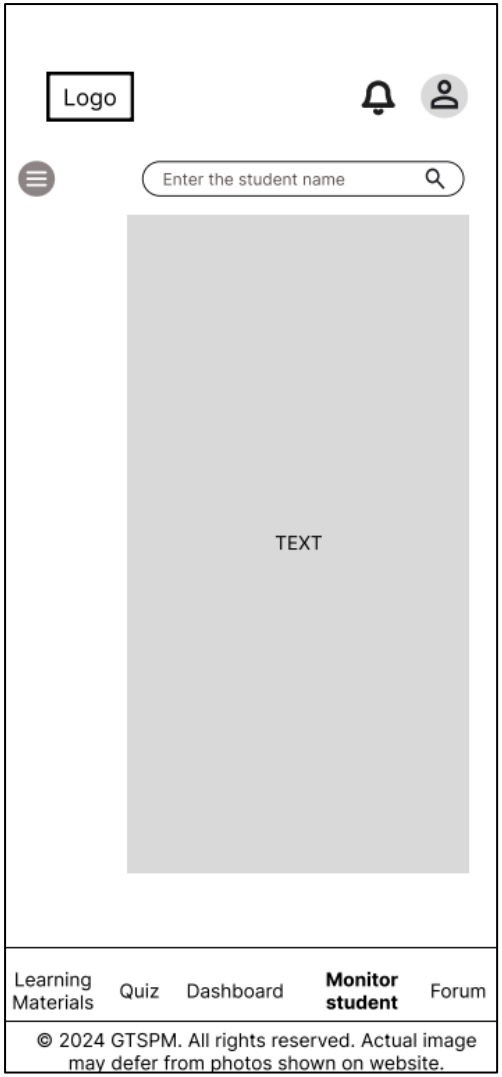
User role	Educator
Functionality name	Monitor student progress
Group member in charge	Kwan Chun Hoe
Table from the database that produces output, process form input	SUMMARY (to get the student summary to display)
Description	Track the progress of your students through their quiz scores and completed materials. This allows you to identify areas where students may need extra help.

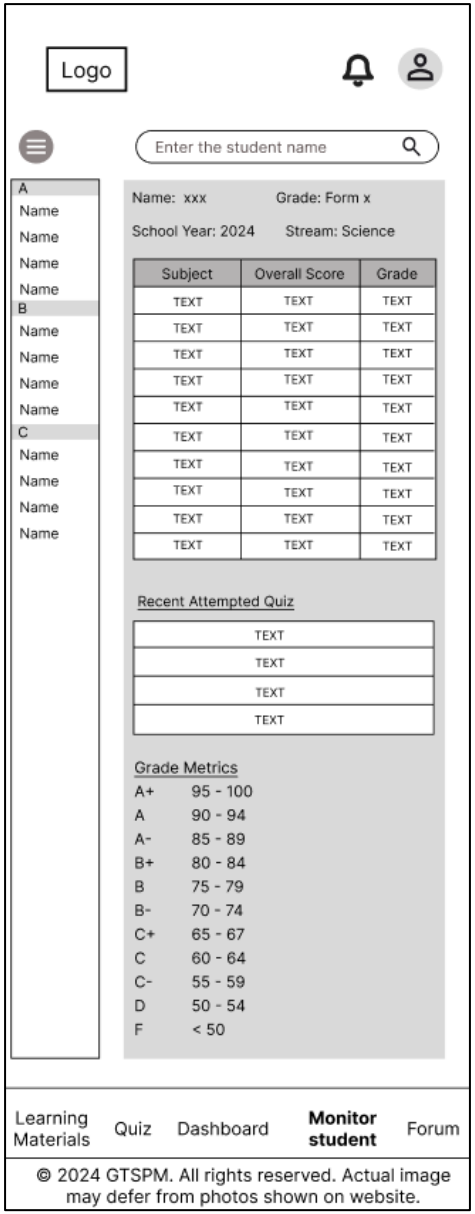
Desktop





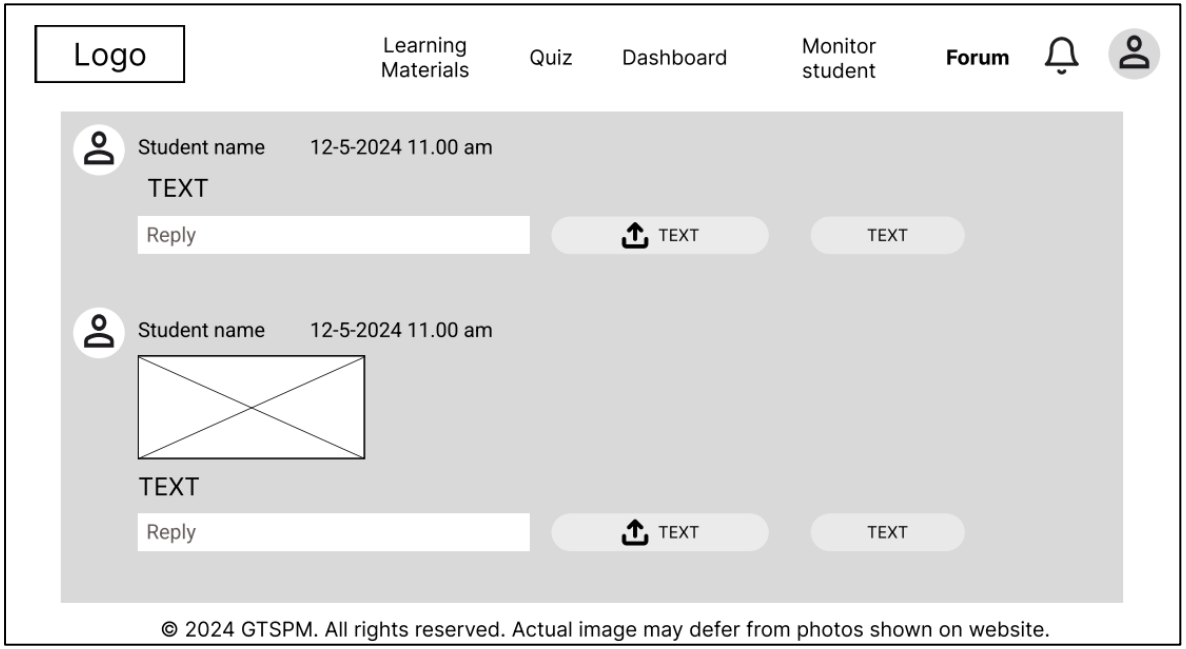
Mobile


Logo
Enter the student name
TEXT
Learning Materials
Quiz
Dashboard
Monitor student
Forum
© 2024 GTSPM. All rights reserved. Actual image may defer from photos shown on website.

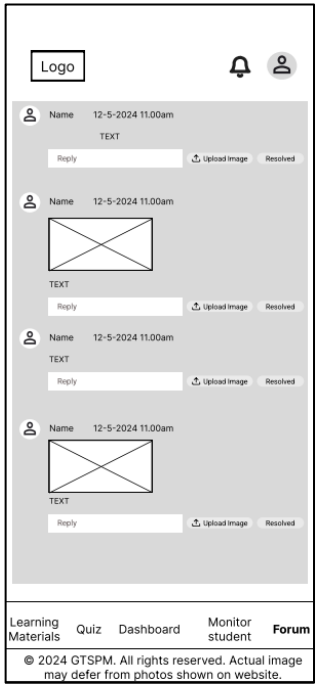

Logo
Enter the student name
A
Name
Name
Name
Name
B
Name
Name
Name
Name
C
Name
Name
Name
Name
Name: xxx
Grade: Form x
School Year: 2024
Stream: Science
Subject
Overall Score
Grade
TEXT
Recent Attempted Quiz
TEXT
Grade Metrics
A+ 95 - 100
A 90 - 94
A- 85 - 89
B+ 80 - 84
B 75 - 79
B- 70 - 74
C+ 65 - 67
C 60 - 64
C- 55 - 59
D 50 - 54
F < 50
Learning Materials
Quiz
Dashboard
Monitor student
Forum
© 2024 GTSPM. All rights reserved. Actual image may defer from photos shown on website.

User role	Educator
Functionality name	Reply student question
Group member in charge	Kwan Chun Hoe
Table from the database that produces output, process form input	QUESTIONBOARD (to get the question to be displayed), REPLYBOARD (to insert new data entry that answered the questions)
Description	Respond to any questions or concerns students have about lessons, assignments, or quizzes, offering additional guidance as needed. Text file can also be uploaded if there is any extra documentation needed to be reviewed.

Desktop



Mobile



4.0 Implementation

4.1 System Developing Steps

1. Understanding Requirements

- The key features required for **students**:
- Signup and login page.
- Homepage navigation.
- Accessing the Revision Corner.
- Managing their user profile.
- Reviewing progress reports.
- The required functionalities for **educators**:
- Creating, updating, and deleting quizzes.
- Uploading educational materials.
- Monitoring students' progress.

2. Planning and System Design

- Developed the project design by:
- Creating wireframes for the system's user interface in Figma.
- Developing flowcharts with draw.io to represent system workflows and procedures.

3. Development

- Key steps taken during the coding phase:
- Created tables in the **gtspm** database using phpMyAdmin, including:

- adminforum, administrator, answerquiz, educator, history, materials, question, questionboard, quizset, replyboard, signup, student, subject, subjectchapter, and summary.
- Designed and implemented entities in these tables to efficiently store and manage data.
- Used Visual Studio Code as the primary development tool for authoring and debugging code.

4. Testing

- Testing was performed during development to discover and correct faults. - Utilized error detection techniques and debugging tools.
- Conducted functionality tests to ensure features like user registration, quiz management, and progress tracking functioned properly.

5. Deployment

- Hosted the system locally using WampServer64 and a local server configuration.
- Demonstrated the system for review and evaluation.

6. Maintenance

- Planned for ongoing maintenance, including:
- Regular updates to fix bugs and improve system performance.

4.2 Code Implementation

Ejjaz Hakimi bin Mohamad Azan (TP073318)

1) File: RevisionCorner.php

```php
<?php
// Code Done By: Ejjaz Hakimi bin Mohamad Azan TP073318
session_start();  // Start the session

// Check if user is logged in
if (!isset($_SESSION['user_id'])){
    echo  "<script> alert('Please login to continue !')
    window.location.href='homepage.php'
    </script>";
}

// Define both time and remaining time variables
$time = isset($_SESSION['time']) ? $_SESSION['time'] : "00:00:00";
$remaining_time = isset($_POST['remaining_time']) ? $_POST['remaining_time'] :null;

// If remaining time exists, declare remaining time as a session variable
if (isset($_POST['remaining_time'])) {
    $_SESSION['remaining_time'] = $remaining_time;
}

// Initialise the question array for later use
$questions = [];

// Store the selected values in session on form submission
if ($_SERVER["REQUEST_METHOD"] == "POST" ) {
    // Save the selected Stream and Subject in the session
    if (isset($_POST['Stream']) && isset($_POST['Subject'])) {
        $_SESSION['Stream'] = $_POST['Stream'];
        $_SESSION['Subject'] = $_POST['Subject'];
    }
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Loop through the posted answers and store them in the session
    foreach ($_POST as $key => $value) {
        if (strpos($key, 'question-') === 0) {
            // Store the selected answer in the session using the question index as key
            $questionIndex = (int)substr($key, 9); // Extract the question index
            $_SESSION['answers'][$questionIndex] = $value;
        }
    }
}
```

RevisionCorner.php first begins with authenticating whether the user has logged in by checking the $_SESSION[‘user_id’] value. If the value is null / not set, they will be redirected back to the homepage alongside with a prompt alerting them to log in. The reason this is done is because our system mandates that all quiz attempts are recorded in our database which is impossible without gathering user data. The code then continues by defining the session variables for both the allocated timer as well as the remaining time. This was implemented due to the fact that users will be jumping though pages when answering the quiz as our system only displays 3 questions (maximum) at a time. With that being said, it is

crucial that the value on the timer is maintained when shifting through the pagination to ensure accurate data tracking. The same can also be said with block of code at the bottom which functions to store the selected answers by utilising $key. This is made possible due to the fact that all questions labelled with the same pattern being ‘question-x’.

```php
// Retrieve selected values from session
$selectedStream = isset($_SESSION['Stream']) ? $_SESSION['Stream'] : (isset($_GET['Stream']) ? $_GET['Stream'] : '');
$selectedSubject = isset($_SESSION['Subject']) ? $_SESSION['Subject'] : (isset($_GET['Subject']) ? $_GET['Subject'] : '');


if ($_SERVER["REQUEST_METHOD"] == "POST" || isset($_GET['Stream']) && isset($_GET['Subject'])) {
    include("conn.php");

    $stream = isset($_POST["Stream"]) ? $_POST["Stream"] : (isset($_GET['Stream']) ? $_GET['Stream'] : null);
    $subject = isset($_POST['Subject']) ? $_POST['Subject'] : (isset($_GET['Subject']) ? $_GET['Subject'] : null);

    if ($stream && $subject) {
        // Get the SubjectID based on the selected subject
        $subjectQuery = "SELECT SubjectID FROM subject WHERE Name = ?";
        $stmt = $conn->prepare($subjectQuery);
        $stmt->bind_param("s", $subject);
        $stmt->execute();
        $result = $stmt->get_result();

        if ($result->num_rows > 0) {
            $subjectData = $result->fetch_assoc();
            $subjectID = $subjectData['SubjectID'];

            // Fetch the timer based on the selected SubjectID
            $timerQuery = "SELECT Time FROM quizset WHERE SubjectID = ?";
            $stmt2 = $conn->prepare($timerQuery);
            $stmt2->bind_param("i", $subjectID);
            $stmt2->execute();
            $timerResult = $stmt2->get_result();

        if ($timerResult->num_rows > 0) {
            $timerData = $timerResult->fetch_assoc();
            $time = $timerData['Time']; // Store the time in the session
            $_SESSION['time'] = $time; // Save the time value in session

        }
```

The following code snippet essentially entails the gathering of the timer data based on the $_SESSION[‘SubjectID’] value set in the previous page

```php
            // Fetch SetID from quizset table based on SubjectID
            $quizsetQuery = "SELECT SetID FROM quizset WHERE SubjectID = ?";
            $stmt2 = $conn->prepare($quizsetQuery);
            $stmt2->bind_param("i", $subjectID);
            $stmt2->execute();
            $quizsetResult = $stmt2->get_result();

            if ($quizsetResult->num_rows > 0) {
                $quizsetData = $quizsetResult->fetch_assoc();
                $setID = $quizsetData['SetID'];

                // Fetch questions related to the SetID
                $questionQuery = "SELECT QuestionText, ChoiceA, ChoiceB, ChoiceC, ChoiceD, CorrectAnswer, Score FROM question WHERE SetID = ?";
                $stmt3 = $conn->prepare($questionQuery);
                $stmt3->bind_param("i", $setID);
                $stmt3->execute();
                $questionResult = $stmt3->get_result();

                while ($row = $questionResult->fetch_assoc()) {
                    $questions[] = $row; // Store the questions in the array
                }
            }
        }

        $conn->close();
    }
```

Moving on, the next code snipped just showcases the proses of gathering the question and score data. Once the data has been gathered, they are then stored inside an array called $questions []

```
<div class="question-box">
<?php if (!empty($questionsToDisplay)): ?>
    <form method="POST"  id="question_submit" >

        <?php foreach ($questionsToDisplay as $index => $question): ?>
            <div class="question-box<?php echo ($index + 1); ?>">
            <h3 id = "quesnum">Question <?php echo ($startIndex + $index + 1); ?>:</h3>
            <p id = "questext"><?php echo htmlspecialchars($question['QuestionText']); ?></p>

            <!-- Add a hidden input for the default value -->
            <input type="hidden" name="question-<?php echo $startIndex + $index; ?>" value="null">

                <ul>
                    <li>
                        <input type="radio" id = "ques" name="question-<?php echo $startIndex + $index; ?>" value="A"
                            <?php echo (isset($_SESSION['answers'][$startIndex + $index]) && $_SESSION['answers'][$startIndex + $index] == 'A') ? 'checked' : ''; ?>>
                        <?php echo htmlspecialchars($question['ChoiceA']); ?>
                    </li>
                    <li>
                        <input type="radio" id ="ques" name="question-<?php echo $startIndex + $index; ?>" value="B"
                            <?php echo (isset($_SESSION['answers'][$startIndex + $index]) && $_SESSION['answers'][$startIndex + $index] == 'B') ? 'checked' : ''; ?>>
                        <?php echo htmlspecialchars($question['ChoiceB']); ?>
                    </li>
                    <li>
                        <input type="radio" id = "ques" name="question-<?php echo $startIndex + $index; ?>" value="C"
                            <?php echo (isset($_SESSION['answers'][$startIndex + $index]) && $_SESSION['answers'][$startIndex + $index] == 'C') ? 'checked' : ''; ?>>
                        <?php echo htmlspecialchars($question['ChoiceC']); ?>
                    </li>
                    <li>
                        <input type="radio" id = "ques" name="question-<?php echo $startIndex + $index; ?>" value="D"
                            <?php echo (isset($_SESSION['answers'][$startIndex + $index]) && $_SESSION['answers'][$startIndex + $index] == 'D') ? 'checked' : ''; ?>>
                        <?php echo htmlspecialchars($question['ChoiceD']); ?>
                    </li>
                </ul>
            </div>
        <?php endforeach; ?>

        <!-- Hidden input to track the next page number -->
        <input type="hidden" name="next_page" value="<?php echo $currentPage + 1; ?>">
        <input type="hidden" name="previous_page" value="<?php echo $currentPage - 1; ?>">
        <input type="hidden" name="remaining_time" id="remaining_time" value="">
```

As for displaying the actual data, this is done by utilising a for loop to cover each individual index inside the $question [] array which is then displayed inside a <li> element. It is worth noting that each choice (A, B, C, D) are checked if the $_SESSION[‘answer’] value exists; if not, it is defaulted to being unanswered. Multiple hidden inputs were also utilised to help with the quiz logic. Firstly, there is a hidden input called ‘question’ which checks if there is an answer in all the indexes of $question []. If a question is found to be unanswered, they will automatically be assigned as null. Beyond that, 3 other hidden inputs were used which are ’next-page’, ‘previous-page’ and ‘remaining-time’. ‘next-page’ and ‘previous-page’ are mainly used just to keep track of what page the user is on to help with the pagination logic. ‘remaining-time’ however is utilised to track the value of the timer when a user shifts pages and updating the $_SESSION[‘remaining-time’] value accordingly

```
let timeValue = "<?php
    if (isset($_SESSION['remaining_time'])) {
        echo $_SESSION['remaining_time'];
} else {
    echo $_SESSION['time'];
}?>";
console.log("Initial Time Value: " + timeValue);  // Debug: Check the value loaded from session

if (timeValue !== "00:00:00" && timeValue !== "") {
    let [hours, minutes, seconds] = timeValue.split(":").map(Number);
    let timeLeft = hours * 3600 + minutes * 60 + seconds;

    function formatTime(seconds) {
        let hours = Math.floor(seconds / 3600);
        let minutes = Math.floor((seconds % 3600) / 60);
        let remainingSeconds = seconds % 60;
        return `${hours.toString().padStart(2, '0')}:${minutes.toString().padStart(2, '0')}:${remainingSeconds.toString().padStart(2, '0')}`;
    }

    const timerInterval = setInterval(function () {
        timeLeft--;  // Decrease time left by 1 second
        console.log("Time left: " + timeLeft);  // Debug: Check remaining time

        const timerElement = document.getElementById("timer");
        if (timerElement) {
            timerElement.textContent = formatTime(timeLeft);
        }

        // Update the remaining time in the hidden input
        document.getElementById('question_submit').addEventListener('submit', function(event) {
        const remainingTimeInput = document.getElementById("remaining_time");
        if (remainingTimeInput) {
            remainingTimeInput.value = formatTime(timeLeft);
            console.log("Updated Remaining Time Value: " + remainingTimeInput.value);  // Debug: Log the value
            console.log('Hidden field value: ', document.getElementById("remaining_time").value);

        }
    });

        // If time runs out, stop the timer and alert the user
        if (timeLeft <= 0) {
            clearInterval(timerInterval);
            alert("Time's up!");
            header("Location: 'RevisionCornerResult.php'")
```

Finally, the code above showcases the JavaScript employed to handle the timer function. First, the code checks if $_SESSION[‘remaining-time’] has been set; if that’s the case, the variable timeValue is defined as so; if not, it is defined as $_SESSION[‘timer’]. Once this is done, the code will proceed to gather the values of hours, minutes and seconds by splitting the value by the “:” character as the timer is formatted to be in HH:MM:SS format. Once it has been split, some arithmetic is performed to gather the value of the timer in seconds. Finally, the code will then begin to subtract the value by 1 second every second and once it has reached 0, the user will be redirected back to the Homepage.

2) File: RevisionCornerResults.php

```php
if (!isset($_SESSION['answers']) || empty($_SESSION['answers'])) {
    die("No answers submitted. Please complete the quiz.");
}

if (!isset($_SESSION['Subject']) || !isset($_SESSION['Stream'])) {
    die("Subject or Stream information is missing.");
}

$userID = $_SESSION['user_id'];
$username = $_SESSION['user_name'];
$selectedStream = isset($_SESSION['Stream']) ? $_SESSION['Stream'] : (isset($_GET['Stream']) ? $_GET['Stream'] : '');
$selectedSubject = isset($_SESSION['Subject']) ? $_SESSION['Subject'] : (isset($_GET['Subject']) ? $_GET['Subject'] : '');

// Gathering SubjectID,from the subject table based on both subject and stream session variable
$sql = "SELECT SubjectId FROM subject WHERE Name = ? AND Stream = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("ss", $selectedSubject, $selectedStream);
$stmt->execute();
$subjectResult = $stmt->get_result();

if ($subjectResult->num_rows == 0) {
    die("No matching subject found for the selected stream and subject.");
}

$subjectData = $subjectResult->fetch_assoc();
$subjectID = $subjectData['SubjectId'];


// Gathering SetID, Time from the quizset table based on the SubjectID gathered earlier
$sql = "SELECT SetID, Time FROM quizset WHERE SubjectID = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("i", $subjectID);
$stmt->execute();
$setResult = $stmt->get_result();

if ($setResult->num_rows == 0) {
    die("No quiz set found for the selected subject.");
}
```

The code for RevisionCornerResults.php is mostly comprised of php coding to gather the following values (student name, quiz subject, user score [which is converted to percentage] and time taken). Though these values are in different tables in our database, we can slowly but surely extract all of them by utilising foreign keys. First, we gather the SubjectID based on the $_SESSION[‘Subject’] and $_SESSION[‘Stream’] values. Once this is done, we gather the SetID and Time value by utilising SubjectID.

```php
$setData = $setResult->fetch_assoc();
$setID = $setData['SetID'];
$allocatedTime = $setData['Time'];

// Gathering QuizID, CorrectAnswer & Score from the question table based on the SetID gathered earlier
$sql = "SELECT QuizID, CorrectAnswer, Score FROM question WHERE SetID = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("i", $setID);
$stmt->execute();
$questionResult = $stmt->get_result();

if ($questionResult->num_rows == 0) {
    die("No questions found for the quiz set.");
}

// Storing all answers and scores into an array
$correctAnswers = [];
$questionScores = [];
while ($row = $questionResult->fetch_assoc()) {
    $correctAnswers[$row['QuizID']] = $row['CorrectAnswer'];
    $questionScores[$row['QuizID']] = $row['Score'];
}

// Comparing student answers with correct answers
$userAnswers = $_SESSION['answers'];
$correctCount = 0;
$totalScore = 0;

$quizIDs = array_keys($correctAnswers); // Get the QuizIDs in order from $correctAnswers

foreach ($quizIDs as $index => $quizID) {
    // Check if there's a user answer at the current index
    if (isset($userAnswers[$index]) && $userAnswers[$index] !== null && $userAnswers[$index] !== "null") {
        $userAnswer = $userAnswers[$index]; // Get the user's answer
        $correctAnswer = $correctAnswers[$quizID]; // Get the correct answer for the QuizID

        if ($userAnswer === $correctAnswer) {
            $correctCount++; // Increment correct count
            $totalScore += $questionScores[$quizID]; // Add the score for the correct answer
        }
    }
}
```

Next, we gather the QuizID, CorrectAnswer value and Score value from the SetID we obtained. The CorrectAnswer value is then placed in an array called $correctAnswers [] which is then compared with the $_SESSION[‘answers’] array which houses all of the user’s answers. For every matching value, they are awarded points based on the corresponding point value.

```php
// Generating score
$totalQuestions = count($correctAnswers);
$maxPossibleScore = array_sum($questionScores);
$scorePercentage = round(($totalScore / $maxPossibleScore) * 100, 2);

// Convert the quiz time from HH:MM:SS to seconds
list($hours, $minutes, $seconds) = explode(":", $allocatedTime);
$totalAllocatedTimeInSeconds = ($hours * 3600) + ($minutes * 60) + $seconds;

$remainingTime = isset($_SESSION['remaining_time']) ? $_SESSION['remaining_time'] : "00:00:00";

//Convert the remaining time from HH:MM:SS to seconds
list($remHours, $remMinutes, $remSeconds) = explode(":", $remainingTime);
$remainingTimeInSeconds = ($remHours * 3600) + ($remMinutes * 60) + $remSeconds;

// Calculate the time spent to complete quiz
$timeSpentInSeconds = $totalAllocatedTimeInSeconds - $remainingTimeInSeconds;

// Format the time spent into HH:MM:SS
$hoursSpent = floor($timeSpentInSeconds / 3600);
$minutesSpent = floor(($timeSpentInSeconds % 3600) / 60);
$secondsSpent = $timeSpentInSeconds % 60;

$timeSpentFormatted = sprintf("%02d:%02d:%02d", $hoursSpent, $minutesSpent, $secondsSpent);

$currentDate = date('Y-m-d');
```

Once the total score has been gathered, it is converted into a percentage. Beyond that, the time spent to answer the quiz is also gathered by performing the arithmetic seen above.

```php
$sql = "INSERT INTO summary(StudentID, SetID, Time, TotalScore, Date) VALUES ('$userID',$setID,'$timeSpentFormatted',$totalScore, '$currentDate')";

$conn->query($sql);

$conn->close();
```

The values are then inserted into the summary table using the query seen above

```php
<section class = "results">
    <form>
    <h1> Nice Try <h1> <br>
    <h3> Username: <?php echo $username ?> <h3> <br>
    <h3> Quiz: <?php echo $selectedSubject ?> <h3> <br>
    <h3> Score: <?php echo $scorePercentage ?>% <h3> <br>
    <h3> Time: <?php echo $timeSpentFormatted ?> <h3> <br>
    <button type="submit" name="return" formaction="RevisionCornerMainPage.php" id = "returnbut"> Return </button>
    </form>
</section>
<section class="footer">
```

Finally, the values are then displayed in html alongside a button which will return users to RevisionCornerMainPage.php when clicked.

Elianna Catrina Herrera (TP073631)

File: upload.php

```php
// Check for file upload
if (isset($_FILES["file"])) {
    if ($_FILES["file"]["error"] == 0) {
        $filename = basename(path: $_FILES["file"]["name"]);
        $file_type = strtolower(string: pathinfo(path: $filename, flags: PATHINFO_EXTENSION));
        $file_title = $_POST['title'];

        // Allowed file types
        $allowed_types = array("jpg", "jpeg", "png", "gif", "pdf");
        if (!in_array(needle: $file_type, haystack: $allowed_types)) {
            echo "Sorry, only JPG, JPEG, PNG, GIF, and PDF files are allowed.";
        } else {
            // Use hardcoded test values for EducatorID and SubjectID
            $educatorID = $_SESSION['EducatorID']; // Hardcoded EducatorID
            $subjectID =  $_POST['SubjectID'];  // Hardcoded SubjectID

            // Define paths
            $relative_path = $target_dir . $filename; // File path to store in the database
            $absolute_path = $absolute_dir . $filename; // Full file path to move the file

            // Move the uploaded file to the specified directory
            if (move_uploaded_file(from: $_FILES["file"]["tmp_name"], to: $absolute_path)) {
                $filesize = $_FILES["file"]["size"];
                $filetype = $_FILES["file"]["type"];

                // Insert file data into database
                $sql = "INSERT INTO materials (name, upload_date, Type, SubjectID, EducatorID, FilePath, Title)
                        VALUES (?, NOW(), ?, ?, ?, ?, ?)";
                $stmt = $conn->prepare(query: $sql);
                $stmt->bind_param(types: "ssssss", var: &$filename, vars: &$filetype, $subjectID, $educatorID, $relative_path, $file_title);

                if ($stmt->execute()) {
                    echo "<script type='text/javascript'> alert('The file " . htmlspecialchars(string: $filename) . " has been uploaded successfully.');
                    window.location.href = 'learningMaterials.php';
                    </script>";
                } else {
                    echo "Database error: " . $stmt->error;
                }
            } else {
                echo "Error: Unable to move the uploaded file.";
            }
        }
```

In the file upload php, its function is to allow educators to upload study materials for students to view and utilize. for the script will begin by checking if a file has been uploaded. After verification, the uploaded file's information will be extracted. The file name will be extracted, followed by retrieving the file extension and then obtaining the title of the file from a form field. The script allows specific file types such as pdf, jpg, png. Since only educators are allowed to access this functionality, $educatorID would be utilize to take from the current session. Moreover, $subjectID is fetched from the form submission. After that, $relative_path is used to store the path in the database while $absolute_path is used to physically save the file on the server. Then, the uploaded file will be moved to its intended directory and if it is successful, the file details will be added into the database. The information of the file will be stored in the materials table of our database. Lastly, the script will execute the query and respond to the user accordingly.

File: download.php

```php
if (isset($_GET['file'])) {
    $file = basename(path: $_GET['file']); // Prevent directory traversal
    $filePath = __DIR__ . "\\Educator\\fileupload\\" . $file; // Use absolute path

    if (file_exists(filename: $filePath)) {
        // Set headers for file download
        header(header: 'Content-Description: File Transfer');
        header(header: 'Content-Type: application/octet-stream');
        header(header: 'Content-Disposition: attachment; filename="' . $file . '"');
        header(header: 'Expires: 0');
        header(header: 'Cache-Control: must-revalidate');
        header(header: 'Pragma: public');
        header(header: 'Content-Length: ' . filesize(filename: $filePath));

        // Send the file
        readfile(filename: $filePath);
        exit;
    } else {
        echo "Error: File not found at $filePath.";
    }
} else {
    echo "";
}

$conn->close();
?>
```

In this download.php, its function is to allow students to download the materials that has been uploaded by educators. The script will check if file name is provided in GET request and basename() is used to ensure that only the file name is extracted. Next, the absolute file path for the file would be constructed. The script will then check the physical file existence at the given path set which is “Educator\fileupload\” as concatenation would take place. A list of headers are set for download, followed by file reading and the file’s content would be ready to be display to the browser.

Suchitra Nambiar A/P Mahandran (TP074762)

1) Select an Avatar and Update Profile Image

```
// Select an Avatar and Update Profile Image
function selectAvatar(avatarFileName) {
    const avatarPath = `images/${avatarFileName}`;
    document.getElementById('profileAvatar').src = avatarPath;  // Update profile image in View Mode
    document.getElementById('editProfileAvatar').src = avatarPath;  // Update profile image in Edit Mode
    document.getElementById('avatarInput').value = avatarFileName; // Update hidden input value
    console.log("Avatar Path Is:" + avatarPath);
    console.log ("Avatar filename = " + avatarFileName);
    const avatarValue = document.getElementById('avatarInput').value;
    console.log("Avatar Input Value Before Submission:", avatarValue);
    closeAvatarPopup();  // Close the Avatar Popup
}
```

The "**selectAvatar**" function allows a user to choose a new profile picture (avatar). When an avatar is selected, the function updates the profile image shown on the page, both in the "View Mode" and the "Edit Mode." It does this by changing the image source ("**src**") to the path of the selected avatar image file. Additionally, the function saves the name of the selected avatar image in a hidden input field, which will be sent to the server when the form is submitted. The function also logs information about the selected avatar, such as its path and filename, in the browser's console for debugging purposes. Once the avatar is selected and updated, the function calls another function ("**closeAvatarPopup()**") to close the avatar selection popup, completing the process.

2) Change Password Section

```
<!-- Change Password Section -->
<div id="changePassword" class="profile-section" style="display: none;">
    <h2>Change Password</h2>
    <form id="changePasswordForm" action="update_password.php" method="POST">
        <label for="currentPassword">Current Password:</label><br>
        <input type="password" id="currentPassword" name="currentPassword" required><br><br>

        <label for="newPassword">New Password:</label><br>
        <input type="password" id="newPassword" name="newPassword" required><br><br>

        <label for="retypePassword">Retype New Password:</label><br>
        <input type="password" id="retypePassword" name="retypePassword" required><br><br>

        <button type="submit" class="update-btn">Update</button>
    </form>
</div>
```

This HTML code defines a "Change Password" section within a user's profile page. Initially hidden (**style="display: none;"**), the section contains a form for updating the user's password. The form has three input fields: one for the current password ("**currentPassword**"), one for

the new password ("**newPassword**"), and one for retyping the new password ("**retypePassword**"). All fields are required. When the user submits the form, it sends a POST request to the server via the **update_password.php** file to process the password change. There is also a button labelled "**Update**" to submit the form and update the password.

3) Change Password JavaScript

```
document.getElementById("changePasswordForm").addEventListener("submit", function(event) {
// Get form data
var currentPassword = document.getElementById("currentPassword").value;
var newPassword = document.getElementById("newPassword").value;
var retypePassword = document.getElementById("retypePassword").value;

// Basic frontend validation
if (newPassword === currentPassword) {
    alert("The new password cannot be the same as the current password.");
    event.preventDefault();
    return false;
}

if (newPassword !== retypePassword) {
    alert("The retyped password does not match the new password.");
    event.preventDefault();
    return false;
}
});
```

This JavaScript code works alongside the profile picture update function, allowing users to change both their avatar and password on the same page. The JavaScript validation is connected to the "Change Password" section by listening for the form submission event (**submit**) on the "**changePasswordForm**". The password change form is validated to ensure the new password isn't the same as the current one and that the retyped password matches the new one. If either condition fails, the form submission is prevented, and an alert is shown. This ensures a secure, user-friendly experience by preventing invalid data submission. Together, these features create a seamless profile management system where the HTML form collects user input, and the JavaScript validates it before submission.

Ng Vin Ee (TP073088)

1. Checking User Credentials in Multiple Tables

```php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $user_email = $_POST['email'] ?? '';
    $user_password = $_POST['password'] ?? '';

    $admin_query = "SELECT 'Administrator' AS role, AdminID AS ID, Name, Email, Password FROM administrator WHERE Email = ?";
    $stmt_admin = $conn->prepare($admin_query);
    $stmt_admin->bind_param("s", $user_email);
    $stmt_admin->execute();
    $admin_result = $stmt_admin->get_result();

    $educator_query = "SELECT 'Educator' AS role, EducatorID AS ID, Name, Email, Password FROM educator WHERE Email = ?";
    $stmt_educator = $conn->prepare($educator_query);
    $stmt_educator->bind_param("s", $user_email);
    $stmt_educator->execute();
    $educator_result = $stmt_educator->get_result();

    $student_query = "SELECT 'Student' AS role, StudentID AS ID, Name, Email, Password FROM student WHERE Email = ?";
    $stmt_student = $conn->prepare($student_query);
    $stmt_student->bind_param("s", $user_email);
    $stmt_student->execute();
    $student_result = $stmt_student->get_result();

    $user_found = false;
```

When the user hits "submit" on the login form, the code grabs the email and password they typed in. It then checks if that email exists in three different tables: `administrator`, `educator`, and `student`. Each of these checks is done using a prepared query to make sure the email is safely passed into the database (to avoid any security issues). The goal is to find a match for the email and get details like the user's role, name, ID, and password. After checking all three tables, if any of them return a result, the code sets a flag (`$user_found`) to `true`, meaning a user was found. If no matches are found, the flag stays `false`, which helps ensure that the password is only verified if a valid user exists.

2. **Displaying Quiz Results for Students**

```
<div class="summary-container">
    <?php
    if (!empty($quizzes)) {
        foreach ($quizzes as $quiz) {
            echo "
            <div class='quiz-card'>
                <h3>Student: $username </h3>
                <p>Grade: {$quiz['StudentGrade']}</p>
                <h4>Subject: {$quiz['SubjectName']} ({$quiz['SubjectStream']})</h4>
                <p>Total Questions: {$quiz['TotalQuestions']}</p>
                <p>Correct Answers: {$quiz['TotalScore']}</p>
                <p>Accuracy: {$quiz['Accuracy']}%</p>
                <p>Date: {$quiz['Date']}</p>
            </div>";
        }
    } else {
        echo "<p>No quiz data to display.</p>";
    }
    ?>
</div>
```

This part of the code checks if there's any quiz data available in the $quizzes array. If there is, it loops through each quiz and displays a card with the details, such as the student's name, grade, subject, number of questions, correct answers, accuracy, and the date the quiz was taken. Each quiz's information is inserted into the HTML dynamically using PHP. If no quiz data is found, it simply shows a message saying, "No quiz data to display." The layout is structured to show the quiz results clearly for each student.

3. **Password Validation Script**

```
<script>
    function validatePassword() {
        var password = document.getElementById("password").value;
        var confirmPassword = document.getElementById("confirmPassword").value;

        if (password !== confirmPassword) {
            alert("Passwords do not match. Please try again.");
            return false;
        }
        return true;
    }
</script>
```

This JavaScript function checks if the password entered by the user matches the confirmation password. It grabs the values of both the password fields (password and confirmPassword) and compares them. If the passwords don't match, an alert is showed telling the user to try again, and the function returns false, preventing the form from being submitted. If the passwords match, it returns true, allowing the form to proceed.

Kwan Chun Hoe (TP076282)

1. File: getSubject.php

```
function getSelectedStream(dropdown) {
    var selectedStream = dropdown.value; // Get the selected stream value

    // Create an XMLHttpRequest object to send the selected stream to the PHP file
    var xhr = new XMLHttpRequest();
    xhr.open('GET', 'getSubject.php?stream=' + selectedStream, true);
    xhr.send(); // Send the request to the server
    // When the server response is received, update the subject dropdown
    xhr.onload = function () {
        if (xhr.status == 200) {
            document.getElementById('subject-select').innerHTML = xhr.responseText;
        } else {
            console.error("Failed to fetch subjects");
        }
    };
}
```

The getSubject.php functions to retrieve subjects based on the selected stream, however, it can only starts after this JavaScript function has been called to it, at first it will get the value from the dropdown option that is selected by the users and store into the variable selectedStream. Next, a XMLHttpRequest is created to request data from the localhost database. First, it will create a connection to the targeted PHP file that interacts with the database to retrieve the required data, in this case will be the getSubejct.php where it functions to populate the subject dropdown based on the subject stream selected. Then, it will send the request with the selectedStream value being passed as arguments. Once the data is returned from the database and the status property is 200, which indicates the response is ready, the responseText property returns the server response as a text string and will be used to update our webpage.

```php
<?php

1 reference
function getSubject($stream): void{
    include("conn.php");

    // Sanitize the stream to prevent SQL injection
    $stream = mysqli_real_escape_string(mysql: $con, string: $stream);

    // Proper SQL query with quotes around the stream value
    $sql = "SELECT SubjectID, Name FROM subject WHERE Stream = '$stream'";

    // Execute the query and handle errors
    $result = mysqli_query(mysql: $con, query: $sql);
    if (!$result) {
        echo "Error: " . mysqli_error(mysql: $con);
        return;
    }

    // Initialize the $options variable to store the generated HTML
    $options = "";
    while ($row = mysqli_fetch_assoc(result: $result)) {
        $subjectID = $row['SubjectID'];
        $options .= "<option value='".$subjectID."'>" . htmlspecialchars(string: $row['Name']) . "</option>";
    }

    // Output the options
    echo $options;
}

if (isset($_GET['stream'])) {
    $stream = $_GET['stream']; // Get the stream value passed via AJAX
    getSubject(stream: $stream); // Call the function with the selected stream
}
?>
```

When this PHP file is called, it first checks if the super global method is set and if true, it will retrieve the passed arguments which indicate the selected subject stream then it will the PHP function to retrieve the subjects that match with the stream. This is done using SQL query to select the name and the SubjectID of the subject that matches the stream value. The name is for display purposes for the user to understand the option offers easily and the SubjectID is assigned to the respective subject inside the dropdown for future purposes. After the result is retrieved it will run in a while loop and concatenate the values we want into the $options variable, the concatenation is done by the concatenation assignment operator '.='. After the while loop has ended, the concatenated options will be going to populate the subject dropdown menu.

2. File: deleteQuiz.php

```php
<?php
include("conn.php");
// $educatorID = $_SESSION['EducatorID'];
$educatorID = "ED0001"; //test data
$sql = "SELECT s.Name, qs.Time, qs.SetID FROM quizSet qs INNER JOIN
        subject s  on qs.SubjectID = s.SubjectID where EducatorID = '$educatorID'";
$result = mysqli_query(mysql: $con, query: $sql);
if (!$result) {
    echo "Error: " . mysqli_error(mysql: $con);
    return;
}
while ($row = mysqli_fetch_assoc(result: $result)) {
    $SetID = $row['SetID'];
    echo '
    <div class="quiz-item">
        <div class="quiz-content">
            <p>' . htmlspecialchars(string: $row["Name"]) . ' </p>
            <p>Time Allocated: ' . htmlspecialchars(string: $row["Time"]) . ' </p>
        </div>
        <div class="btn-group">
            <button class="delete" id = "deleteBtn" value="' . $SetID . '" type="button" onclick="confirmDelete()">Delete</button>
        </div>
    </div>';
}
?>
```

The flow start from this PHP code that dynamically displays the quizzes created by specific educator which will based on the value get from $_SESSION[‘EducatorID’], and each quizzes that displayed with have a delete button with the SetID assigned to it that uniquely identified the quiz and each button is added with an onclick listener that will trigger the confirmDelete() function below.

```html
<script>
    function confirmDelete(){
        if (confirm("Are you sure want to delete this quiz, it is an irreversible action!")){
            var setID = document.getElementById('deleteBtn').value;
            window.location.href = 'deleteQuiz.php?setID=' +setID;
        }
    }

</script>
```

After a button is clicked and this function is triggered, it will first reconfirm the user as the action next cannot be reverted back, this created a sense of caution for the educator to consider their action thoroughly. If the educator chooses to continue with the deletion, a variable setID will be assigned to the value data from the delete button clicked and pass that value to deleteQuiz.php as an argument.

```php
<?php
    if (isset($_GET['setID'])){
        include ("conn.php");
        $SetID = $_GET['setID'];
        $deleteQuestion = "DELETE FROM question WHERE SetID = '$SetID'";
        mysqli_query(mysql: $con, query: $deleteQuestion);
        $deleteQuizSet = "DELETE FROM quizset WHERE SetID = '$SetID'";
        mysqli_query(mysql: $con, query: $deleteQuizSet);
        if (!mysqli_error(mysql: $con)){
            echo "<script>alert('Quiz set and questions deleted successfully.');</script>";
            echo "<script>window.location.href = 'quiz-page.php';</script>";
        }else{
            echo "<script>alert('Error when delete');</script>";
        }
    }
?>
```

After the argument setID is assigned to the PHP side variable, $SetID using the $_GET method, we are executing two deletions, one is for the question table which stores the questions for certain quizzes and the another table is quizset that stores the main info for each quiz. We start deleting the questions first because it is referenced the SetID from the quizset table therefore the child dependency have to be remove first before removing the corresponding data from the parent table. Once the query has executed with no error which is denoted by '!mysqli_error($con)' it will display an alert message to inform the educator that the quiz is deleted successfully using JavaScript alert function. After that it will direct the user back to the quiz-page and reload it to update the display.

5.0 Conclusion

Looking back at our work we can conclude that the project was for the most part, handled smoothly with us only suffering from schedule difficulties forcing us to reevaluate and reconsider our proposed project requirements. Beyond that, we managed to successfully achieve the objectives we sought to reach with our project. While the objectives were successfully met, that does not mean that improvements cannot be made. For example, our application does not have the ability for students to send questions to educator and for educators to respond back. This was initially planned in our preliminary draft for the application however time constraints have forced us to drop the feature to allow us to prioritise features that we deem to be more important and crucial. Perhaps if granted more time, this feature could have been implemented by erecting a forum page on both student and educator side allowing for gathering and storing of user questions / responses.

Furthermore, another limitation in our project is the inability for educators to edit either uploaded materials or quizzes. The only available options for educators are to delete the entire material / quiz. This is an obvious setback as it causes inconvenience towards the user which negatively impacts overall user experience. Time constraint was the biggest factor in this feature not being included as the complexity of enabling educators to edit quizzes (which could very well be lengthy) was too much for us to handle in the time frame we were given.

Finally, another limitation that our project suffers from is the lack of variety in our quiz options. To elaborate, our quiz only allows for questions to be in multiple choice format; any other format such as subjective / open ended questions are not permitted by our system. This is mainly due to the fact that we want our system to create a seamless learning environment and part of that is implementing automated learning and evaluating with is impossible with open ended questions. With all that being said, we do recognise the limitations in the effectiveness of learning through MCQ quizzes. Perhaps in the future, we could employ a feature that sends open-ended responses to educators for manual marking leaving the system to auto-evaluate the MCQ answers.

References

Kazuaki Kajitori, Aoki, K., & Ito, S. (2014). Developing a Compact and Practical Online Quiz System. *International Journal of Modern Education and Computer Science*, *6*(9), 1–7. https://doi.org/10.5815/ijmecs.2014.09.01

Krisbiantoro, B. (2021). THE EFFECTIVENESS OF GAMIFICATION TO IMPROVE STUDENTS' TENSES MASTERY. *International Conference on Education of Suryakancana (IConnects Proceedings)*, *0*. https://doi.org/10.35194/cp.v0i0.1337

MyGOV - The Government of Malaysia's Official Portal. (n.d.). Www.malaysia.gov.my. https://www.malaysia.gov.my/portal/content/29552

Nugraha, E. N. L., Salsabila, & Ramadhiani, T. S. (2021). Implementing Online Quiz Application in EFL Classroom [Review of *Implementing Online Quiz Application in EFL Classroom*]. *Proceedings International Conference on Education of Suryakancana 2021*, 308–314.

Shamil (2024) *Power-Ups & Their Types*. (2024, August 2). Help Center. https://support.quizizz.com/hc/en-us/articles/360035742872-Power-Ups-Their-Types

Shoyeb, I. H. (2015). *Android based Online Quiz Application* (pp. 1–92) [Review of *Android based Online Quiz Application*].

Appendix

Workload Matrix

Group Number: Group 7

Group Leader Name: EJJAZ HAKIMI BIN MOHAMAD AZAN

Member Name and Student ID (including Group Leader)	Task Completed				Overall Contribution %
	Documentation	**Web App (Client-side)**	**Web (Server Side)**	**Database**	
EJJAZ HAKIMI BIN MOHAMAD AZAN	• Background Analysis & Requirement Gathering • Implementation • Conclusion	• Designed the UI for Revision Corner (Student Side) where students can attempt quizzes and view their results	• Implemented the quiz functionality to retrieve quiz questions from the database as well as upload results to the database • Debugging of member's work	• Redesigned the summary, quiz set table	22%

DANESH A/L SELLASAMY	• Wireframe	• -	• -	• -	1%
ELIANNA CATRINA HERRERA	• Introduction • Implementation • Student Flow Chart • Design	• Designed the UI for View / Download Materials (Student Side) as well as Upload Materials (Educator Side) • Designed the UI for the Student Side homepage	• Implemented the functionality to allow students to download materials that was uploaded by educators	• Designed the entire initial database	18%
KWAN CHUN HOE	• Introduction • Educator Flow Chart • Implementation	• Designed the UI for the entire Administrator Side with the exception of Upload Material	• Implemented the backend for the entire Administrator Side with the exception of Upload Materials • Debugging of member's work	• Redesigned the question and administrator table	24%
NG VIN EE	• Implementation (Steps involved in developing the system)	• Designed the UI for the login & sign-up process to allow all user no matter the role to access the system	• Implemented the login and sign-up feature alongside user verification	• Redesigned the student table	17%

	• Design	• Designed the UI to showcase student progress report on attempted quizzes	• Implemented the progress report functionality which displays the summary data		
SUCHITRA NAMBIAR A/P MAHANDRAN	• Introduction • Implementation • Student Flow Chart	• Designed the UI to allow users to edit their profile details • Designed the UI for the Student Side homepage	• Implemented the user customisation feature allowing students to edit their user details as well as select their avatar	• Designed the entire initial database	18%
Total					100%