ONLINE LEARNING MANAGEMENT SYSTEM WITH ADVANCED QUIZ AND MARKING SYSTEM

BY

TAN SHI YONG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2025

COPYRIGHT STATEMENT

This Final Year Project proposal is submitted in partial fulfillment of the requirements for the degree of Bachelor of Information Systems (Honours) Information Systems Engineering at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project proposal represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project proposal may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ACKNOWLEDGEMENTS

First, I would like to express my deepest thanks to my supervisor, Mr Sun Teik Heng, for their guidance, patience, and valuable feedback throughout this project. Their expertise and encouragement helped me overcome challenges and improve my work.

I wish to express my profound appreciation to Universiti Tunku Abdul Rahman (UTAR) for the provision of resources and support essential to the completion of this research.

To my family and friends, thank you for your endless support, motivation, and understanding during this journey. Your encouragement kept me focused and determined.

Finally, I appreciate all the lecturers, colleagues, and participants who contributed directly or indirectly to this project. Your insights and assistance were invaluable.

ABSTRACT

This project aims to developing an Online Learning Management System (LMS) With Advanced Quiz and Marking System which focuses on improving the efficiency on making quizzes, grading, and feedback easier and better. Many current LMS platforms like Moodle, Google Classroom, and Frappe LMS help with teaching and assessments, but they still have some weaknesses. This project is going to compares those platforms, takes their best parts and tries to addressing their limitations. This proposed system is built by using ASP.NET Core and includes features like an easy-to-use quiz builder, auto-grading for objective questions, a real-time feedback for students, and a manual grading option with rubrics for essay questions and assignment. Thus, this system hopes to make online teaching and learning better for both teachers and students by saving their time and improving assessment quality.

Area of Study: Online Learning, Educational Technology

Keywords: Learning Management System (LMS), Online Assessment, Quiz Management, Automated Grading, Feedback Mechanism, Moodle, Google Classroom, ASP.NET Core.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In this chapter, this project would like to investigate to the objective, problem statement, and more about the Online Learning Management System that focuses on advance quiz and grading system. Online Learning Management Systems (LMS) have changed education by giving us tools to teach, test, and connect with students online in different learning situations. The existing LMS platforms like **Moodle, Google Classroom, and Frappe LMS** show how online learning has grown, and each has its own good features. As the demand for online education continues to grow, there is an increasing need to enhance the capabilities of learning management systems, especially in the area of assessment tools. Thus, it is important to improve how the quizzes work, how grading is done, and how feedback is given to fostering more effective learning experience and accurate performance assessments.

1.1 Problem Statement and Motivation

In today's online learning environment, many teachers find it hard to create and managing quizzes, grade assignments, and give students quick, helpful feedback, especially when they have many students [1]. While current online learning systems (LMS) help with many things, they still have weaknesses. For example, it can be difficult to set up different grading methods, lack of real-time feedback, and time-consuming quiz creation [1]. Research on automated grading and feedback has shown that traditional grading methods require teachers to spend a significant amount of time [2]. This has motivated the exploration of automated solutions to reduce teacher burden and potentially enhance student engagement and performance [2]. Furthermore, while the existing LMS platforms provide useful features, their limitations in assessment can create ongoing challenges. These include difficulties in offering a reliable, detailed feedback and supporting a diverse of question types for effectively measuring student learning [3].

This project is motivated by the need to fix these problems with existing LMS like Moodle, Google Classroom, and Frappe LMS when it comes to quizzes, feedback, and grading. The goal is to build a better online Learning Management System (LMS) that makes the teachers create and manage quizzes more easily, grade them fairly (both automatically and manually), and give students immediate and useful feedback. By combining the best features of existing LMS platforms and adding new tools, this system will help teachers save time and support students more effectively to improve the overall online learning experience for everyone.

1.2 Project Objectives

This project aims to develop an Online Learning Management System (LMS) with Advanced Quiz and Marking System to address limitations in existing tools like Google Classroom, Moodle, and Frappe LMS. The system will focus on three core improvements:

1. **Advanced Quiz System**
    - Design an intuitive interface for educators to create quizzes with diverse question types (multiple-choice, fill-in-the-blank, essay, etc.).
    - Support features like time limits and randomization to reduce cheating.
2. **Real-Time Feedback Mechanism**
    - Provide instant feedback to students after quiz submission, showing correct/incorrect answers with explanations.
    - Allow educators to add custom feedback for subjective questions also (e.g., essays).
3. **Automated and Customizable Grading**
    - Implement auto-grading for objective and fill in the blanks quiz (e.g., MCQs) to save teachers' time.
    - Implement customizable grading schemes and manual grading for essays using rubrics to ensure fair assessment across different question formats. (like a teacher can customize the score and grade for a test and customise the mark for each question).

1.3 Project Scope

This project will deliver a **web-based Learning Management System (LMS)** built with **ASP.NET Core** in Visual Studio 2022, focusing on advanced quiz creation and marking features. The system will include:

1. **A Functional LMS Web Application**
    - Developed using **C# and ASP.NET core** for backend logic.
    - **SQL Server** for database storage (e.g., user accounts, quizzes, grades).
    - Responsive frontend with **HTML5, CSS, and JavaScript**.
2. **Core Features**
    - **Quiz Module**: Educators can create quizzes with multiple question types (e.g., Essay, MCQ, fill-in-the-blank) and set time limits.
    - **Real-Time Feedback**: Students view results immediately after submission, with explanations for answers.
    - **Grading System**: Auto-grading for objective questions and manual rubric-based grading for essays and assignment.
3. **Deliverables**
    - Incomplete source code with documentation.

1.4 Contributions

This project aims to make a real difference in how teachers and students experience online learning. Right now, teachers spend a lot of time making quizzes, grading them, and giving feedback. This can take away from their time to teach and connect with students. By building an advanced Learning Management System (LMS) with better quiz and grading features, we will help teachers do these tasks much faster and more easily. At the same time, students can get quicker and clearer results, which can help them understand their mistakes and learn better. This makes learning more effective and less stressful for everyone involved.

Our system also solves real problems in popular platforms like Moodle, Google Classroom, and Frappe LMS. These systems are useful but still have gaps like slow feedback, hard-to-use quiz builders, and limited grading options [4]. By fixing these problems, teachers can save valuable time and focus more on teaching and helping students learn better. Students will also benefit because they will get their grades and feedback faster, which can help them understand what they need to improve right away. By making online quizzes and grading easier, this project can make online education more effective and less stressful for everyone involved. This could lead to a better learning experience for many students and a more manageable workload for teachers in the long run.

1.5 Report Organization

This report is divided into several chapters, each focusing on a different part of the project. **Chapter 1** gives a general introduction to the project, including the background, problem statement, project goals, scope, and contributions. It explains why the project is important and what it hopes to achieve.

Chapter 2 presents a literature review of existing Learning Management Systems (LMS) such as Moodle, Frappe LMS, and Google Classroom. It compares their features, especially in quiz creation, grading, and feedback. The chapter also points out their limitations and how our system will try to improve on them.

Chapter 3 is the proposed method/approach that describes the methods and technologies used to build the proposed system. It explains how the system was developed, the tools and programming languages used, and how each feature works.

Chapter 4 focuses on the overall system design of the proposed Learning Management System (LMS) with an advanced quiz and marking module. It describes the architecture of the system, the relationships between different entities, the UML class diagram, and the process flowcharts for both teachers and students.

Chapter 5 discusses the Simulation of the Work. It explains how the system works, including screenshots and sample tests. It also includes a user guide, and any challenges faced during the development process.

Chapter 6 ensure the reliability, usability, and functionality of the Learning Management System (LMS) with an advanced quiz and marking module, the system was evaluated through a series of structured test cases.

Chapter 7 conclude the report with a summary and recommendation of the project. This chapter summarizes the project information such as problem, motivation, and proposed solutions.

1.6 Project Background

In recent years, Learning Management Systems (LMS) have become a key part of education, whether students are learning online or in classrooms. These computer systems help teachers share lessons, see how students are doing, and talk to them online. Many teachers use platforms like Moodle, Google Classroom, and Frappe LMS. But even with these helpful tools, there are still some problems. It's not always easy for teachers to create quizzes, grade them, and give good feedback to students [4].

While the widespread adoption of LMS has accelerated more recently, particularly with the increased need for online learning, the history of these systems dates back several decades [5]. One of the earliest LMS software was "FirstClass," created in 1990, which offered features like online conferences and discussion boards [5]. Following this initial development, the late 1990s and early 2000s saw the emergence of other significant platforms such as Blackboard (launched in 1997) and Moodle (established in 2001) [5]. These early systems laid the groundwork for the comprehensive online learning environments we see today, initially focusing on content delivery and student record-keeping before evolving to include more interactive and assessment-focused features.

As more learning happens online, especially after the COVID-19 pandemic, the need for good online learning systems has become even more important [6]. When universities had to quickly move to online teaching, it showed that LMS need to be strong and easy to use for tests and for giving students quick, helpful feedback [7]. Teachers have felt the pressure to grade quickly, manage lots of online tests, and keep students interested [7]. This means we need better ways to make quizzes, grade them, and give feedback, which are areas where current LMS sometimes do not work as well as they could.

Moodle is one of the most widely used learning management systems in higher education. According to Gamage et al., Moodle had offers easy-to-use tools like quizzes, forums, databases, and workshops to help teachers create digital materials for teaching and assessments [8]. The platform is listed as one of the top 20 best learning systems based on what users say about their experience. Moodle quizzes are particularly useful for both formative and summative assessments, allowing educators to embed different types of multimedia such as text, images, audio, and video into questions [8]. These quizzes offer immediate feedback options, which help students understand their mistakes and improve their learning. Moodle also

provides built-in statistical tools like Facility Index (FI) and Discrimination Index (DI) that help teachers evaluate the effectiveness of quiz questions [8]. Moodle has many features that make it a good system for universities and colleges. It especially works well for courses where students need to solve complex problems and work with calculations.

Google Classroom is another widely used LMS, known for being easy to get started with [9]. Students find it helpful for things like getting assignments, turning in their work online without using paper, and talking with their teachers and classmates [9]. It also helps teachers keep their online classes organized [9]. However, students have also pointed out some challenges with Google Classroom. These include sometimes finding it hard to understand things online, getting feedback from teachers late, and issues with internet connection [9]. Some students also miss having face-to-face interactions and feel there is not enough support for learning online [9]. These challenges show that even though Google Classroom has many benefits, there are still areas where the online learning experience, including things like assessments and feedback, could be improved.

On the other hand, Frappe LMS is designed to be an easy-to-use and open-source learning system [7]. Its creators wanted to build something that was not complicated, with a simple design so teachers could focus on teaching instead of struggling with the technology [7]. Frappe LMS lets teachers create courses, add different kinds of lessons like articles, videos, and quizzes, and track how students are doing, all in one place [7]. While it aims to be user-friendly for creating courses, the blog post does not go into a lot of detail about its specific features for making and managing quizzes, or its grading system, compared to more established platforms like Moodle [7]. So, while Frappe LMS offers a simpler approach to managing online learning, there's still a need for LMS that make creating quizzes, grading, and giving feedback both easy and effective for teachers across different platforms.

CHAPTER 2

LITERATURE REVIEWS

This chapter explores existing Learning Management Systems (LMS) with a focus on their capabilities in quiz management, grading, and feedback mechanisms. By examining Moodle, Google Classroom, and Frappe LMS, we aim to identify their strengths, limitations, and how these systems attempt to address the challenges faced by educators in administering assessments. Additionally, this review outlines the gaps in these systems and presents how our project intends to overcome these weaknesses by developing a more advanced and integrated quiz and marking system.

2.1 Moodle

Moodle is a popular free learning system that gives teachers a lot of tools for teaching, testing, and managing courses online. One of its key features is the quiz functionality, quiz, which lets teachers make different kinds of questions and grade them automatically. Moodle also has ways to look at quiz results, such as the Facility Index (FI) and Discrimination Index (DI), to evaluate the effectiveness of quizzes. These tools help teachers see how hard and how useful the quiz questions are, so they can make their quizzes better [10].

2.1.1 Key Features of Moodle

Moodle has several important features that help with online learning. First, it lets teachers create different kinds of quizzes and grade them automatically. It also gives teachers tools to check how good the quiz questions are [10]. Second, Moodle lets teachers use text, pictures, and videos in lessons and quizzes, which makes learning more interesting [10]. Third, Moodle helps teachers give students feedback quickly, so students can learn and improve [10]. Fourth, Moodle provides tools for teachers to manage their classes, including giving assignments, using chat rooms and forums, and sharing learning materials [11]. Fifth, Moodle can be changed and adapted to fit what teachers need because it is open source [11]. Sixth, Moodle is designed to make students learn actively by doing things and talking to each other [11]. Finally, Moodle can be used to support different ways of learning, like when the system changes to fit each student or when students learn together [10].

2.1.2 What have other researchers/developers done to resolve the problem?

Researchers have been trying to make Moodle better, especially how it handles quizzes. One way is to make Moodle quizzes work better for both practice and final tests. Gamage and others (2019) used things like text, pictures, and videos in quiz questions to get students more interested and give them fast feedback [10]. They also used psychometric analytic methods which are Facility Index (FI) and Discrimination Index (DI) to check the quiz questions and make sure the tests were accurate [10]. This helps quizzes not just test students but also help them learn and remember things.

Gamage et al. (2022) also looked at many studies on how Moodle is used [10]. They found that Moodle is used a lot in science and math classes in colleges [10]. Studies show that Moodle helps students do better, feel happier with the course, and pay more attention [10]. This review also pointed out that Moodle is being used to support things like adaptive learning (where the system changes to fit each student) and working together in groups [10].

Athaya et al. (2021) also reviewed research on Moodle. They found that Moodle is a popular tool in schools and universities [11]. Moodle is designed to let students learn by doing and talking to each other [11]. This shows that Moodle is seen as a system that helps students learn actively.

2.1.3 Strengths

Moodle is strong because it has good quiz tools, including automatic grading, ways to check if the questions are good, and the ability to add things like videos to make tests more interesting [10]. It also gives feedback quickly, which helps students learn and do better in online classes [10]. The 2022 review also supports this, showing that Moodle is a popular and useful tool that helps students learn more and enjoy learning [10]. Athaya et al. (2021) add that Moodle has tools like assignments, quizzes, chat rooms, and ways to share materials, which makes it a strong system for learning [11].

2.1.4 Weaknesses/Limitation

Even though Moodle has good quiz tools, it takes a lot of time to set up and change. The grading part of Moodle is not always flexible enough for different teaching needs. And while Moodle gives quick feedback, it does not always give teachers detailed information about how students are learning, which could help them teach in a more personalized way [10]. The 2022 review by Gamage et al. focused more on the positive aspects of Moodle, but it did mention that there are still gaps in research, such as the need for more studies on how teachers use Moodle [10]. Athaya et al. (2021) pointed out that using Moodle can be hard for teachers and students who need to get used to it [11]. They also noted that it can be difficult to give good feedback to students online and that students can get confused by too much information.

Strengths	**Weaknesses/Limitation**
Robust Quiz Capabilities: Includes automated grading, statistical tools, and multimedia support.	**Manual Setup**: Requires extensive manual setup and customization.
Timely Feedback: Provides immediate feedback, increasing student engagement and performance.	**Limited Advanced Grading**: Lacks flexibility in advanced grading schemes and detailed insights.
Psychometric Tools: Uses Facility Index (FI) and Discrimination Index (DI) for question quality.	**Complex Customization**: Customization is complex and not user friendly.

Table 2.1. Strengths and Weaknesses/Limitation of Moodle

2.2 Google Classroom

Google Classroom is an online learning system that is made by Google [12]. It is part of a set of Google Workspace for Education (formerly G Suite for Education). It started in 2014 to make it easier to give, get, and grade homework without using paper [13]. The system integrates with other Google tools such as Google Docs, Google Drive, and Google Forms, creating a full set of learning tools.

In 2019, Sudarsana and others said that Google Classroom is one of the online tools from Google that helps teachers and students work better when learning online [14]. The application can download and used for free, but schools need to set it up, so it works best in organized learning places.

2.2.1 Key Features of Google Classroom

Google Classroom has several features that help teachers manage online learning. First, teachers can use it to create classes, add students, share announcements, and handle all their classes in one place [13]. Second, teachers can give out and collect assignments online, which means they do not need to use paper, and it makes the process easier [13]. Third, Google Classroom helps teachers and students communicate through announcements, comments, and private messages [15]. Fourth, it works well with other Google tools, like Google Drive for saving files, Google Forms for tests, Google Docs for writing together, and Google Meet for online meetings [13]. Fifth, teachers can use it to track how students are doing, give grades, and save the grade information [15]. In 2016, Iftakhar said that Google Classroom is a really good tool for teachers because it has many helpful features that make it easier for them to work with students [15].

2.2.2 Basic Usage and Implementation

Sudarsana et al. (2019) provide a practical overview of Google Classroom's basic functionality and implementation, particularly in the context of Indonesia's educational system [14]. The initial setup of Google Classroom involves establishing user roles (teacher or student), creating classes, and managing student enrollment. Teachers can add students to classes through three primary methods:

- Providing a class code for students to join independently
- Manually inviting students through the platform
- Managing student enrollment by adding or removing students as needed

The study also highlights the communication functionality of Google Classroom, noting that teachers can send emails directly to individual students or small groups from within the platform, while using the Announcement feature for class-wide communications [14].

2.2.3 Assessment and Evaluation Tools within Google Classroom

Google Forms for Assessment

Google Classroom uses Google Forms to let teachers create and give out tests, surveys, and quizzes. Mulatsih (2020) showed that Google Forms was used to test students' knowledge in chemistry classes [13]. Google Forms is helpful because:

1. Teachers can easily make quizzes with different kinds of questions, like multiple-choice and short answer (Ease of Creation).
2. Google Forms can automatically grade some questions, which saves teachers time and gives students fast feedback (Automatic Grading).
3. The answers from students are saved in a spreadsheet, so teachers can easily see how students did (Data Collection and Analysis).
4. Students can take the tests on any device with internet, which is convenient (Accessibility).
5.

2.2.4 Strengths of Google Classroom

Google Classroom has several good things about it. First, it works very well with other Google tools. This makes it easy for teachers and students to use, and it helps everything flow smoothly [13]. Second, it's easy to use as it has a user-friendly interface. Even if someone is not very good with computers, they can still use it [15]. Third, it does not cost much. Schools that use Google Workspace for Education can use Google Classroom for free, which is good for schools that do not have a lot of money [15]. Fourth, users can use it on phones and tablets. Google Classroom and the tools that go with it have apps, so they can use them on different devices [15].

2.2.5 Weaknesses/Limitations in Google Classroom

Google Classroom also has some downsides. First, even though Google Forms can be used for tests, it does not have all the advanced features that some special testing programs have [13]. Second, user cannot change and customize Google Classroom as much as they can with some other learning systems. Third, it can be hard to make sure students are doing their work by themselves without cheating. Fourth, teachers have to spend a lot of time grading work that students upload to Google Classroom, especially if they have a lot of students. Basically, grading can take up a lot of the teacher's time.

Strengths	Weaknesses/Limitations
Integration with Google Ecosystem: Works well with other Google tools.	**Assessment Limitations:** Google Forms does not have all the advanced testing features that some special testing programs have.
User-Friendly Interface: Easy to use.	**Limited Customization:** Can not be changed and customized too much.
Cost-Effectiveness: Can used for Free	**Assessment Integrity:** Hard to make sure students are not cheating
Mobile Accessibility: Can be used on phones and tablets	**Grading Workload**: Time-consuming for teachers, especially with large classes

Table 2.2 Strengths and Weaknesses/Limitation of Google Classroom

2.3 Frappe LMS

Frappe LMS is a relatively new, open-source platform designed to provide a flexible and customizable learning environment. It's built on the Frappe framework and offers basic LMS features like creating courses, managing learners, and tools for testing students. Frappe LMS is known for being easy to change and adapt, which makes it useful for different kinds of teaching (as mentioned in previous references). The Frappe website (Frappe, 2023) describes it as a "user-friendly learning management system (LMS)" that helps users to create and manage courses, track how students are doing, and talk to students all in one place. According to ITQlick [16], Frappe LMS is designed to meet the needs of modern businesses and schools, making learning simple and fun.

2.3.1 What have other researchers/developers done to resolve the problem?

It's important to understand why Frappe LMS was created. The Frappe website (Frappe, 2023) explains that it started when a group called FOSS United wanted to build a platform for teaching programming called mon.school. They looked at other open-source LMS systems like Moodle, but they found Moodle difficult to use, with a confusing design. So, they decided to create their own LMS that would be easier and less distracting. This shows that Frappe LMS was developed to address the problem of complex and hard-to-use LMS interfaces. ITQlick [16] also supports this by emphasizing that Frappe LMS aims to simplify the learning process.

2.3.2 Strengths

Frappe LMS's strength lies in its flexibility and how much it can be customized, which allows developers to change the platform to fit specific needs (as mentioned in previous references). It also has a simple interface, making it easier for teachers to manage courses and tests, even if they're not very technical. The Frappe website (Frappe, 2023) highlights that Frappe LMS is designed to be user-friendly, with easy-to-use controls for creating courses and clear guides to help new users get started. The open-source nature of Frappe LMS allows for quick improvements based on what users say they need, making it a good choice for those wanting a system that can grow with them. ITQlick [16] adds that Frappe LMS supports multimedia (like videos and audio) and gamification (adding game-like elements), which makes learning more

engaging. ITQlick [16] also notes that Frappe LMS is more affordable than some other LMS options.

2.3.3 Weakness/Limitation

Despite its good points, Frappe LMS is still being developed, and its testing and grading features are not as advanced as those in older LMS systems like Moodle (as mentioned in previous references). Also, Frappe LMS might not connect as well with other business programs, which could be a problem for companies that need a more integrated system. While it works on mobile phones, it may not have dedicated mobile apps or as many connections to other programs, which could be a limitation for users who rely heavily on mobile access. The Frappe website acknowledges that they are constantly adding new features and improvements (Frappe, 2023). ITQlick [16] confirms that Frappe LMS has limited integrations and mobile app support compared to competitors.

Strengths	**Weaknesses/Limitation**
Customization: Highly flexible and customizable to meet specific needs.	**Limited Assessment Features**: Less sophisticated quiz and grading capabilities compared to established platforms.
Ease of Use: Simple interface suitable for educators with varying technical expertise.	**Development Stage**: Still under development with limited advanced features and integrations.
Open Source: Allows rapid innovation and adaptation based on user feedback.	**Limited Mobile Support**: Limited support for mobile applications and fewer third-party content integrations.

Table 2.3 Strengths and Weaknesses/Limitation of Frappe LMS

2.4 Feature Comparison Table

Feature	Moodle	Google Classroom	Frappe LMS	Proposed System
Quiz Creation	Powerful but complex setup	Basic (via Google Forms)	Limited options	**Powerful but easy setup**
Question Types	Multiple types (MCQ, essay)	MCQs/short answers	Basic types	**All types + diagrams/file uploads**
Grading Flexibility	Auto + manual (limited customization)	Auto-grading (objective only)	Basic grading	**Auto-grading + custom rubrics**
Feedback Speed	Fast but generic	Delayed (manual review needed)	Basic	**Instant + personalized comments**
Anti-Cheating Tools	Time limits, randomization	Minimal features	Shuffle Questions	**Time limits + randomization**
Customization	High (requires technical skill)	Very limited	High (developer-friendly	**User-friendly customization**
Cost	Free (hosting costs may apply)	Free	Free	**Free**

Table 2.4 Features comparison

Our system offers significant improvements over current platforms like Moodle, Google Classroom, and Frappe LMS by addressing their key limitations. Moodle is powerful but can be hard to use, while Google Classroom is simple but limited. Frappe LMS has basic tools. My proposed system combines the best parts: it lets teachers create strong quizzes easily, supports many question types including file uploads, and gives fast, personalized feedback. It also has fair grading with custom rubrics, tools to prevent cheating, and is easy for teachers to customize without needing tech skills.

CHAPTER 3: System Methodology/Approach

3.1 System Design Diagram/Equation

This chapter explains how the Online Learning Management System (LMS) with an advanced quiz and marking system will be developed, including the tools used, the steps taken to build the system, and how each part works together. Simple diagrams will show how the system functions and how it meets user needs.

This project will use a modified **Agile methodology** for its development. **The Agile** was chosen because it is well-suited for my projects where requirements might evolve, and it allows for continuous feedback and improvement.

Unlike the traditional Waterfall model (a type of SDLC), this approach will be more suitable for me as its break the project down into a smaller, manageable part. Each sprint will last for a short period and will focus on building, testing, and delivering a specific set of working features.

3.2 System Architecture Diagram

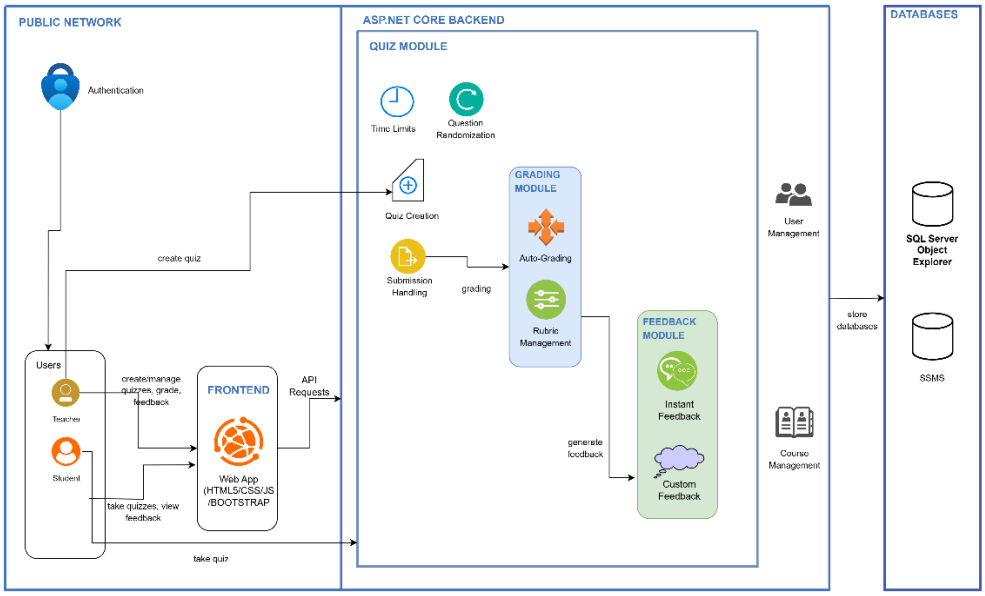


Figure 3.2 System Architecture Diagram

First, we have the **Users** on the left which are the Teacher and the Student– who connect through the **Public Network**, like the internet. To get in, they need to go through **Authentication** to prove they are who they say they are.

Then the users interact with the **Frontend**, which is like the face of the system, the website we can see. It is built using web languages like HTML, CSS, and JavaScript. When a user wants to do something, like create a quiz or take a quiz, the Frontend sends **API Requests** to the **ASP.NET Core Backend**, which is the brain of the system.

Inside the Backend, there is a **Quiz Module** that handles everything about quizzes. Teachers can set **Time Limits** and **Question Randomization** when they do **Quiz Creation**. When students **Submit Answer**, it goes to **Submission Handling**.

Then, there's a **Grading Module** that does the scoring. It can do **Auto-Grading** for objective questions, and it also manages **Rubric Management** for another type of questions.

After grading, the **Feedback Module** kicks in to **generate feedback**. This can be **Instant Feedback** right away for objective questions or more detailed **Custom Feedback** for another type of questions from the teacher.

All the information, like user accounts, quizzes, and grades, is stored in **Databases**, specifically **SQL Server Object Explorer** and **SSMS**. The system also has parts for **User Management** and **Course Management** to keep everything organized in these databases.

3.3 Use Case Diagram

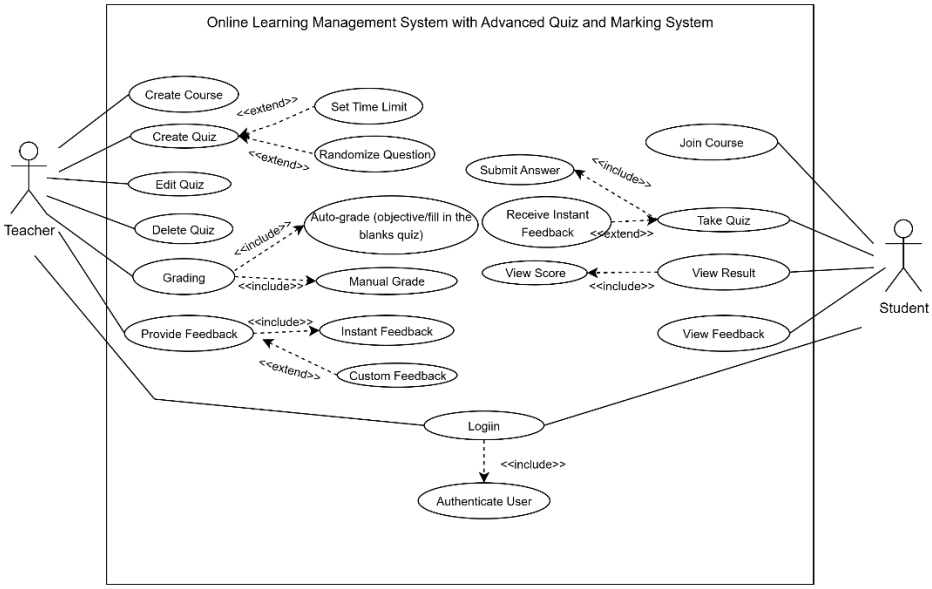


Figure 3.3 Use Case Diagram

First, everyone needs to **Login** to use the system, and to do that, they need to **Authenticate User** (prove who they are).

The teacher can do things related to quizzes, like **Create Quiz, Edit Quiz, and Delete Quiz**. When making a quiz, they can also **Set Time Limit and Randomize Question**. They also do Grading, which includes the system Auto-grade and the Manual Grade. Teachers also **Provide Feedback**, which can be **Instant Feedback** or more detailed **Custom Feedback**. They can also Create Course.

The student can **Join Course and then Take Quiz**. After taking a quiz and Submit Answer, they can **Receive Instant Feedback (Answer/Explaination)** for **objective/fill in the blanks quizzes**. They can also **View Result** which includes the **View Feedback** given by the teacher.

So, this Use Case Diagram simply shows all the different actions each person can take when using this online learning system for quizzes and grading.

3.4 Activity Diagram

Teacher's Activity Diagram

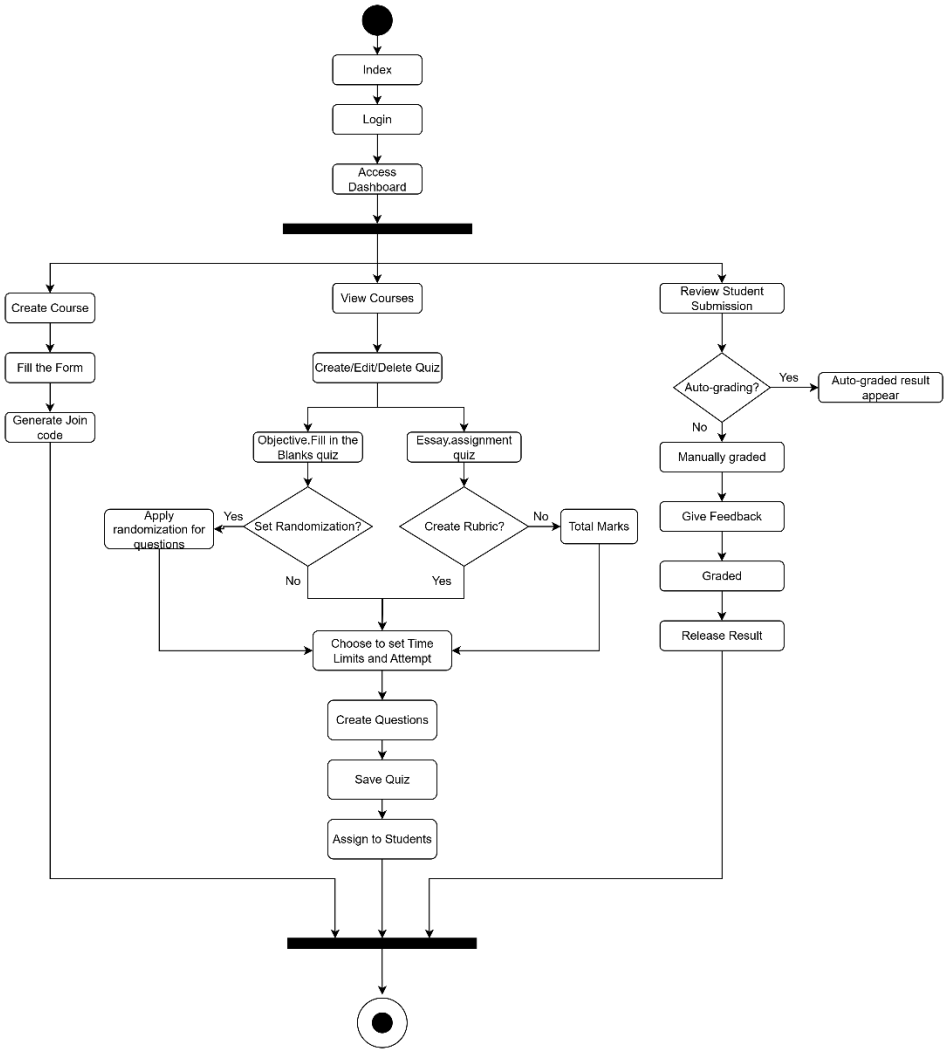


Figure 3.4.1 Activity Diagram of Teacher

This activity diagram starts with going to the **Index page** and then **Login**. After logging in, teacher can get to the **Dashboard**.

From the Dashboard, a teacher can **Create Course**. They need to **Fill in the form**, and if it's successful, the course is created and join code will be generated.

Also from the Dashboard, a teacher can **CreateQuiz/Edit Quiz/DeleteQuiz**. First, they are able to **Choose quiz types**, then **Fill out the quiz details**. They can also **Set time limit and randomization** for the questions. Then, they continue with **create the questions for quiz**. If there are **Objective/Fill in the Blanks**, the system can **Enable auto grading** and teacher can

Add instant feedback for each of the question. If there are **Subjective questions** (like essay/assignment), the teacher can choose to **Add manual grading rubric**. If they do, they need to **Fill out the Rubric form**. After adding all the questions, the teacher can **Assign Quiz** to the students.

When students take the quiz and **Review Student Submission**, the system checks if there are **Objective/Fill in the Blanks quiz**. If yes, the **Auto-graded results appear**. If no (meaning there are subjective questions), the teacher needs to **Manually grade subjective questions** and can **Add custom feedback if needed**. Finally, the teacher ccan **Release Results to Studen**

Student's Activity Diagram

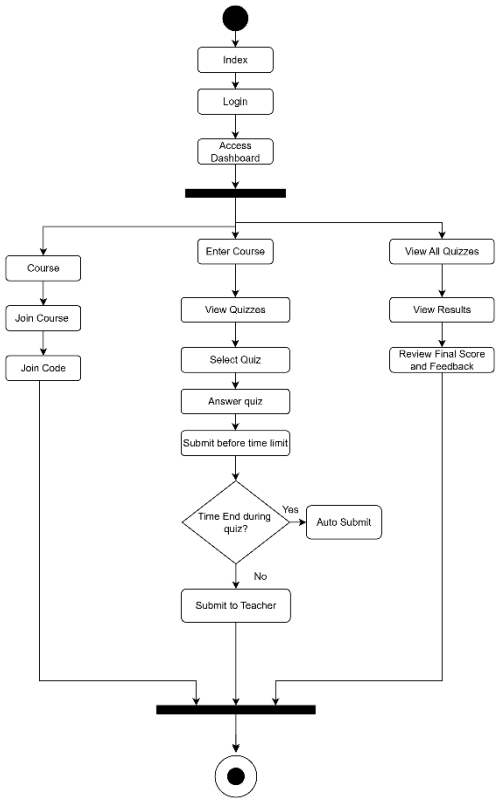


Figure 3.4.2 Activity Diagram of Student

This activity diagram starts with going to the **Index page** and then **Login** to get to their **Dashboard**.

From the Dashboard, a student has a few choices. They can look at **Course** information and **View Courses**. If they want to **Join Courses**, they can **Type Join Code**, and if it works, they **Join course successful**.

Another thing they can do from the Dashboard is **View Quizzes**, where they can view all the quiz and **Review Final Score and Feedback** for their completed work.

After submitting, if there were **Objective/Fill in the blanks quiz**, they will **receive auto-graded results more faster**. If there were **Subjective** questions, they will need to **Wait for manual grading** to **Review Final Score and Feedback**.

They can also **Select the Quiz assign by teacher** and then **Attempt Quiz**. Then **Answer the Questions**. They need to **Submit before time limit and limited attempt if set by teacher**. If

the **Time expired during quiz**, the system will **Auto-submit** their answers. If not, they can **Continue** taking the quiz and just need to **Submit before Deadline**.

CHAPTER 4: System Design

4.1 System Block Diagram

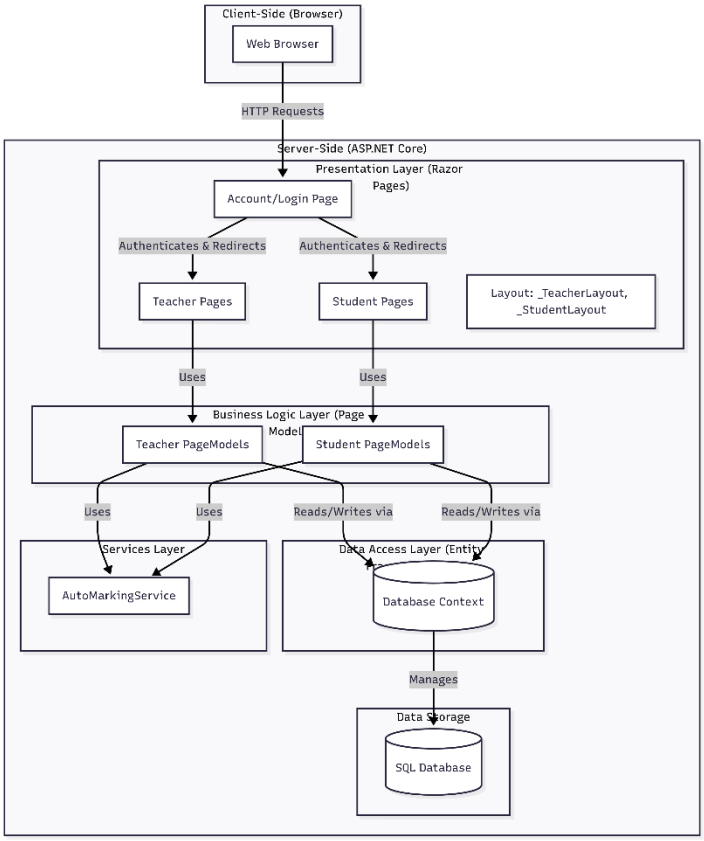


Figure 4.1.1 Block Diagram of Overall System

This Overall System Block Diagram showing the different components interacting with each other's.

1. **Client-Side (Browser):** This is the user's web browser (like Chrome or Firefox). It makes HTTP requests to the server and displays the HTML, CSS, and JavaScript.
2. **Server-Side (ASP.NET Core):** This is where the application's logic runs. It has divided into several logical layers:
    - **Presentation Layer (Razor Pages):** This layer will process the HTTP requests and generates the HTML views that are sent back to the browser. It is divided into the Teacher Pages, Student Pages, and a shared Account/Login page for the authentication.
    - **Business Logic Layer (Page Models):** The files, such as DashboardModel.cs, TakeQuiz.cs, contain the C# code that processes form data, handles user requests, and the core application logic for each page.

- **Data Access Layer (Entity Framework Core):** This layer is represented by the database context and is how this application talks with the database. It will translate C# code into SQL commands to read and write data.
- **Services Layer:** This contains reusable services with specific jobs. The AutoMarkingService containing the logic for automatically grading for the objective and fill-in-the-blank questions.
- **Data Storage (SQL Database):** This is the persistent database (FYPDB) that stores all the application's data, such as user, course, quiz, question, and quizsubmission.

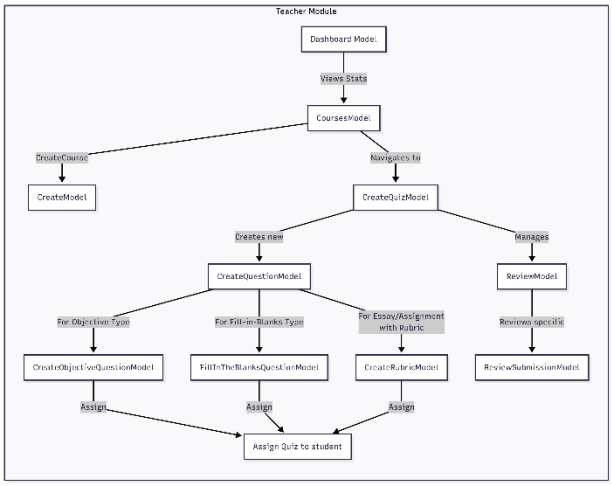


Figure 4.1.2 Block Diagram of Teacher

The Teacher Module is a management and grading dashboard. Its main process is centered around creating quizzes and reviewing student work.

1. **Dashboard:** The teacher can see an overview of their courses, quizzes, and pending grading tasks.
2. **Course Model:** Teacher can navigate to their course list to create a new course or select an existing one.
3. **Create Quiz**: Inside a course, the teacher can access the Quiz Management hub (CreateQuiz). From here, they can choose the type of quiz to create.
4. **Create Question:** They can set up the quiz basics (title, due date) in CreateQuestion and depending on the quiz type, the teacher can navigate to CreateObjectiveQuestion or FillInTheBlanksQuestion. For Essay and Assignment quiz, just stay on

CreateQuestion page and choose to use the Create Rubric to build a detailed grading guide for students.

5. **Grading:** Once the students submit attempts, the teacher can use the Review page to see all submissions and use the ReviewSubmission page to grade student individually, then providing the scores and feedback.

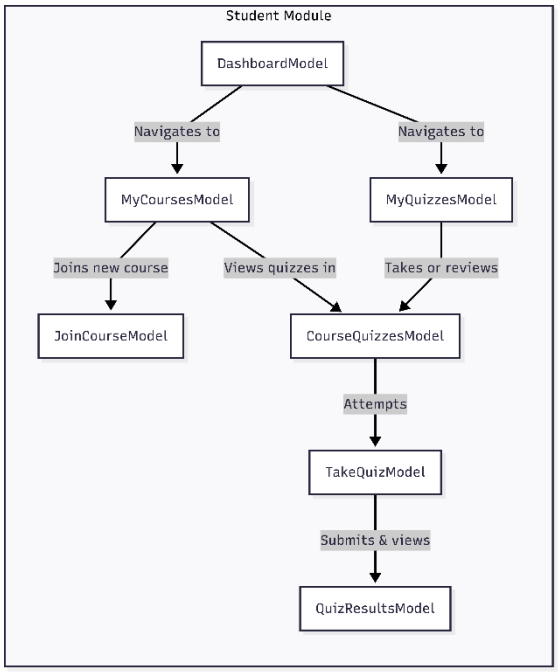


Figure 4.1.3 Block Diagram of Student

The Student Module is an interactive learning and assessment interface.

1. **Dashboard:** Students can view the courses theyare enrolled in, upcoming deadlines, and recent grades.
2. **Joining a Course:** They can use the JoinCourse page to enter a join code provided by their teacher to join into a new class.
3. **Accessing Quizzes:** The student can view all their quizzes across all courses (MyQuizzes) or drill down into quizzes for a specific course (CourseQuizzes).
4. **Taking Quiz:** When they select a quiz, they are taken to the TakeQuiz page. This is the testing environment where they need to answer all questions under a time limit.
5. **Result:** After the submission, student will be taken to the Quiz Results page. This page will shows their score, correct answers, and any feedback from their teacher after the teacher complete their marking. This can allow them to learn from their mistakes.

4.2 Entity-Relationship Diagram (ERD)

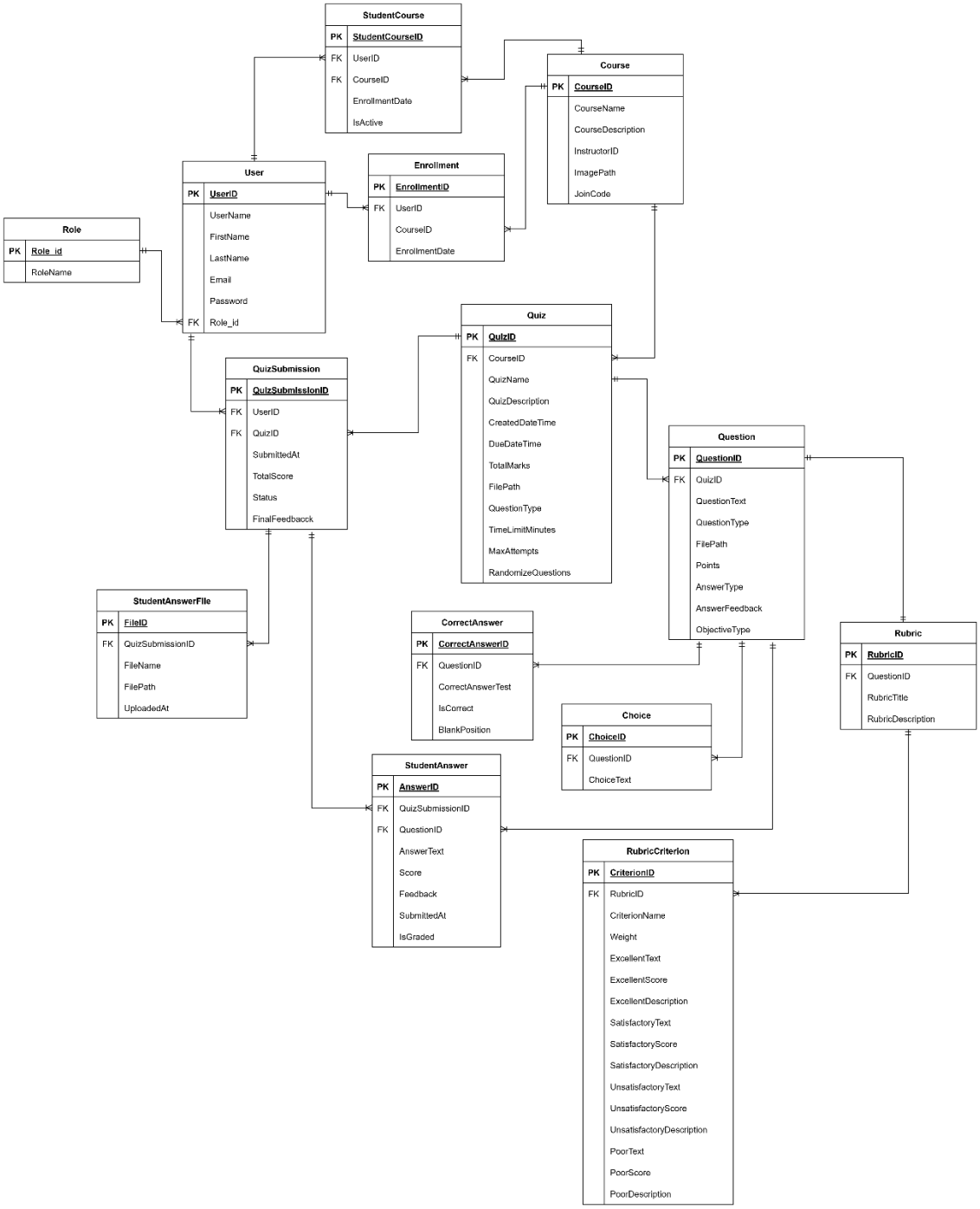


Figure 4.2.1 Entity-Relationship Diagram

Relationships between Entities:

1. **Role → User**
    - **One-to-many:** One Role (e.g., Student, Teacher) can be assigned to many Users, but each User can only have one Role at a time.
2. **User → Enrollment**
    - **One-to-many:** A User can enroll in multiple Courses, but each Enrollment entry links only one User to one Course.
3. **Course → Enrollment**
    - **One-to-many:** A Course multiple Enrollments, but each Enrollment belongs to only one Course.
4. **User → Course (Instructor)**
    - **One-to-many:** A User (Teacher) can instruct multiple Courses, but each Course has exactly one Instructor.
5. **Course → Quiz**
    - **One-to-many:** A Course can have multiple Quizzes, but each Quiz belongs to one Course.
6. **Quiz → Question**
    - **One-to-many:** A Quiz can have multiple Questions, but a Question can only belongs to a Quiz.
7. **Question → Choice**
    - **One-to-many:** A Question can have multiple Choices (for MCQs, True/False), but each Choice belongs to only one Question.
8. **Question → CorrectAnswer**
    - **One-to-many:** A Question can have multiple CorrectAnswers (e.g., multiple blanks or multiple accepted answers), but each CorrectAnswer belongs to one Question.

9. **Question → Rubric**
    - **One-to-one:** A Question can have one Rubric for grading, and each Rubric is linked to exactly one Question.

10. **Rubric → RubricCriterion**

- **One-to-many:** A Rubric can have multiple Criteria (different grading aspects), but each Criterion belongs to one Rubric.

11. **Quiz → QuizSubmission**

- **One-to-many:** A Quiz can have many Submissions (from different students), but each Submission belongs to one Quiz.

12. **User → QuizSubmission**

- **One-to-many:** A User (Student) can submit multiple Quiz attempts, but each Submission is made by exactly one User.

13. **QuizSubmission → StudentAnswer**

- **One-to-many:** A Submission can include multiple StudentAnswers (one per Question), but each StudentAnswer belongs to one Submission.

14. **Question → StudentAnswer**

- **One-to-many:** A Question can have many StudentAnswers (from different students), but each StudentAnswer is tied to exactly one Question.

15. **QuizSubmission → StudentAnswerFile**

- One-to-many: A Submission can have multiple uploaded files (for assignments/essays), but each file belongs to one Submission.

4.3 UML Diagram

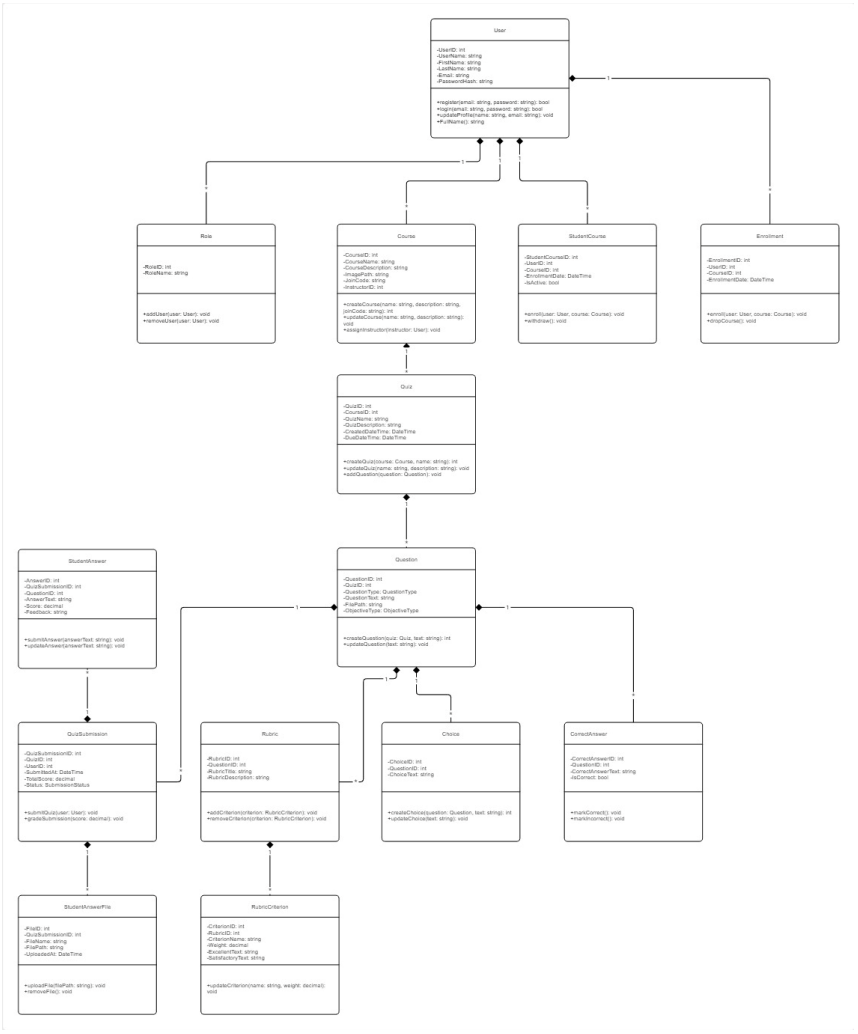


Figure 4.3.1 UML Diagram

This UML class diagram is showing how different parts of the application are connected.

1. User and Roles:
    - The User class is the central point of the system. It represents anyone who uses the application, storing their personal details like name, email, and a secure password.
    - Each User is linked to a **Role**, such as "Student" or "Teacher." This connection tells the system what permissions a user has.
2. Courses and Quizzes:
    - A Course is a subject or class, and it is created by one Instructor ("Teacher" role).

- Users ("Student" role) can join a course through an Enrollment using the join code give by teacher. This table tracks which students are in which courses.
- Each Course can have multiple Quizzes. A Quiz has details like a name, a due date, and how long students have to complete it.

3. Questions and Answers:
    - A Quiz is made up of many Questions. Questions can be different types, such as multiple-choice or true/false.
    - For multiple-choice questions, the Question is linked to a list of Choices.
    - The correct answers for each question are stored in the CorrectAnswer class.
    - For questions that are graded manually, like essays and assignment, a Rubric can be linked to the question. The Rubric has multiple RubricCriterion to help a teacher grade consistently.
4. Submissions and Grading:
    - When a student takes a quiz, their attempt is recorded as a QuizSubmission. This tracks the student, the quiz, and the time of submission.
    - Each QuizSubmission contains many StudentAnswer records. These records store what the student wrote or selected for each question.
    - If a student uploads files as part of their answer, a StudentAnswerFile is created to store the file details.

4.4 Flowchart Diagram

Registration Flowchart

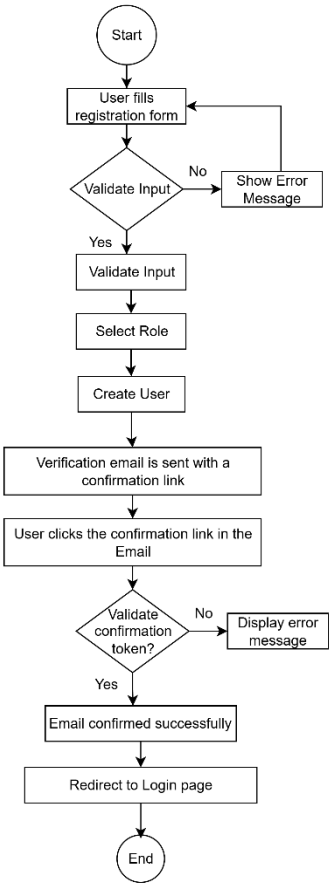


Figure 4.4.1 Register

1. Users fills out the form and select the role (Teacher/Student) and click Register.

2. The email confirmation link will send to the user email and user need to click on the link to complete the confirmation.

3. After registering, the user is will navigate to Login page.

Login Flowchart

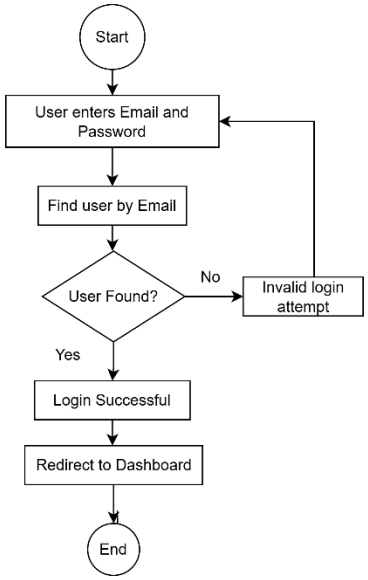


Figure 4.4.2 Login

1. Users enter their username/email and password to log in.

2. Based on the user's role, they are redirected to their specific dashboard (/Teacher/Dashboard or /Student/Dashboard).

Teacher FlowChart

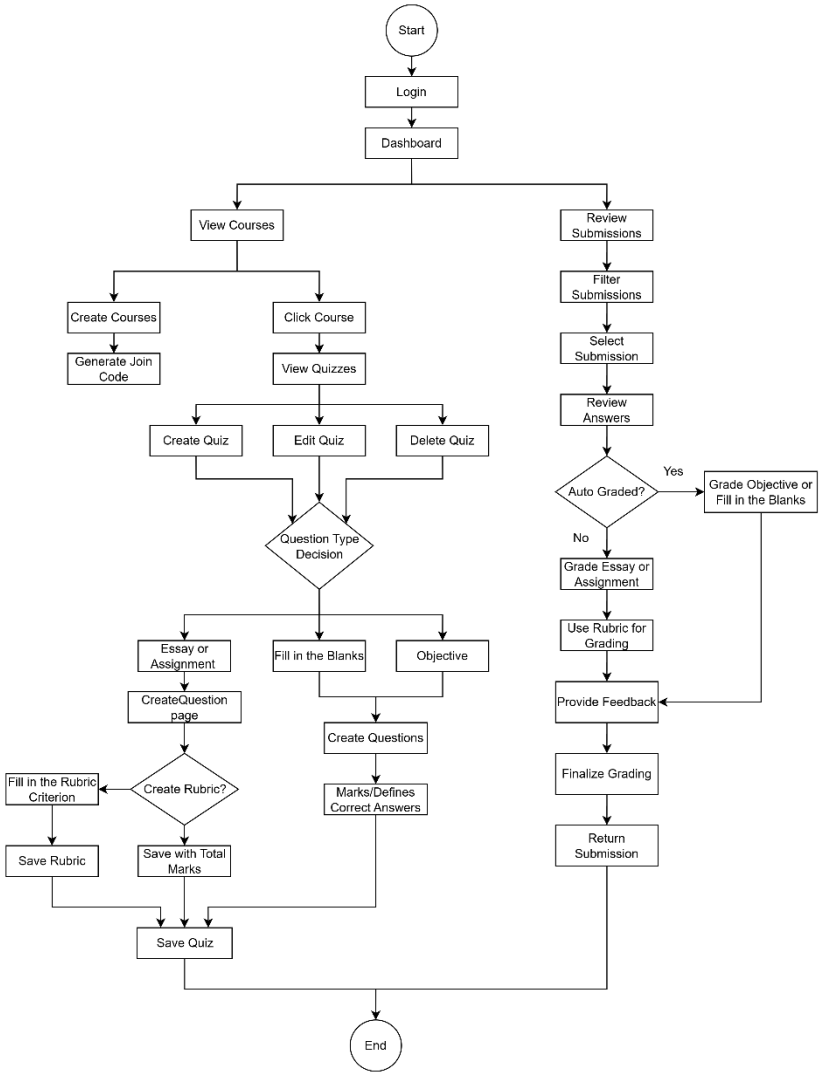


Figure 4.4.3 Teacher Model

1. After login, teachers are redirected to the teacher Dashboard.
2. From the dashboard, the teacher can choose different actions:
   - **View Courses → see the list of courses they manage.**
   - **Review Submissions → check student quiz submissions for grading.**
3. When viewing courses, the teacher has the options which are Create Course (Create new course) or click on the course to open a course and manage their quizzes.
4. Inside a course, the teacher can view, create, edit, and delete the quiz.
5. When creating a quiz, the teacher needs to fill in the detail such as Title, Description, Due Date, Time Limit and Attempts and navigate to next page due to the quiz type they created:

- **Essay/Assignment** → goes to the **CreateQuestion page and create quiz**.
- **Fill in the Blanks** → goes to **CreateQuestions** and then to **FillInTheBlanksQuestion** to create questions and defines correct answers.
    - **Objective (MCQ/True-False)** → goes to **Create Questions** and then to **CreateObjectiveQuestion** create questions by define choices and correct answers.

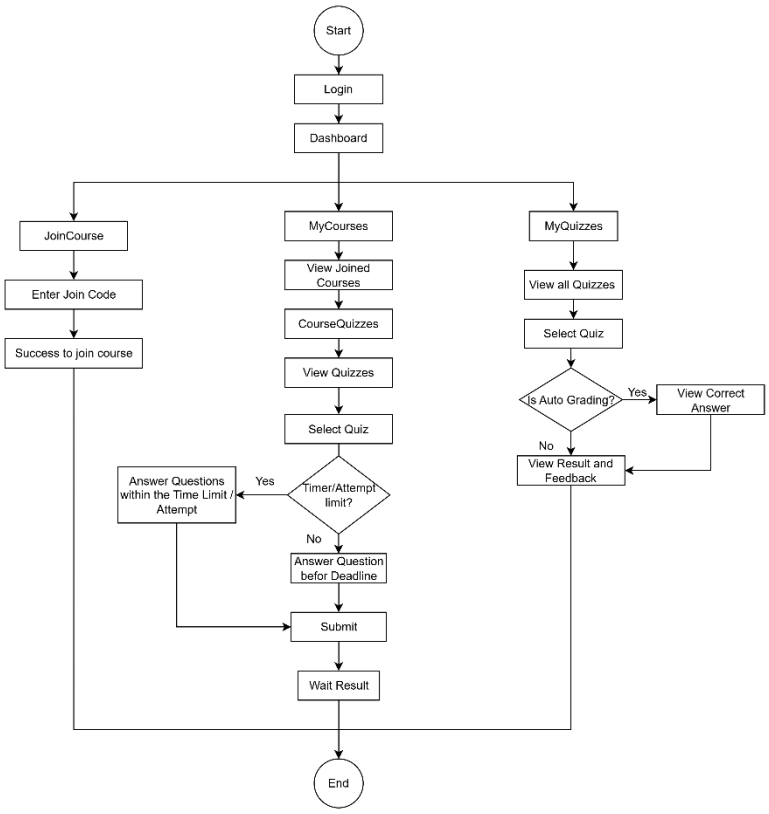
6. For the Essay/Assignment quiz, teacher can decide they want to CreateRubric or TotalMarks to do the marking.

7. In the Review page, the teacher can do the marking in ReviewSubmission page to grade the student submission answer.

8. If the quiz is Auto-Graded (Objective and Fill in the blanks quiz), marks are calculated automatically. The teacher just needs to give their final Feedback and returned it to students.

9. If the quiz is Manual Grading, the teacher may use the Rubric or TotalMarks to do the marking and provides feedback then finalizes the grading.

Student Flowchart



4.4.4 Student model

1. After login, the student will redirect to Student Dashboard.

2. Students can choose to navigate to JoinCourse, MyCourses or My Quizzes page.

3. In JoinCourse, student can use the Join Code given by teacher to join into a course.

4. In MyCourses, it display a simple list of all courses the student is currently enrolled in.

5. Then, student can click a course to navigate to CourseQuizzes page to view a quiz for this course.

6. After that, student can clink on a selected quiz to attempts a quiz and submit it before the time limit and attempt end, if not, the system will automatically submit the quiz.

7. After Submitting, student will navigate to the QuizResults page.

8. In QuizResults page, it displays the detailed results and feedback after a student submits a quiz (no score) or after a teacher grades it (score).

4.5 Wireframe

Figure 4.4.5 Register

-This is the wireframe of Register, which have the column to let user input their informations.

Figure 4.4.6 Log in

-This wireframe let the user can enter their email and password.

Figure 4.4.7 Form (Create Question)

-This wireframe is design for the question form types.

Figure 4.4.8 Course

-This wireframe is design for the teacher and student course page.

Figure 4.4.9 Review/Submission

-This wireframe is use in the review and quiz submission page to let the student and teacher can view the info of the quiz more details.

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 Hardware Setup

Table 3.1 shows the hardware of the laptop used in developing this project. This laptop has a good performance to handle programming tasks and testing. The laptop is an HP Victus Gaming 16-E1044AX. It runs on Windows 11 and has an AMD Ryzen 5 6600H processor, which helps to run software smoothly. It also has 8GB of RAM which helps the computer handle multiple tasks at once and a fast 512GB SSD for storing all the project files and programs. The laptop includes an NVIDIA GeForce RTX 3050 graphics card, which can helps with tasks that need better visuals or processing power. Therefore, this laptop had the necessary power and features to develop the online quiz and grading system.

Description	Specifications
Model	Hp Victus Gaming 16-E1044AX
Processor	AMD Ryzen™ 5 6600H
Operating System	Windows 11
Graphic	NVIDIA® GeForce RTX™ 3050 4GB GDDR6
Memory	8 GB DDR5-4800 MHz RAM (1 x 8 GB)
Storage	512 GB PCIe® NVMe™ TLC M.2 SSD

Table 5.1 Specifications of laptops

5.2 Software Setup

Before starting to develop this Online Learning Management System (LMS) with an advanced quiz and marking system, there are two software needed to be downloaded and installed into my laptop:

1. Visual Studio Enterprise 2022 v17.13.1, .NET SDK
2. SQL Server Management Studio (SSMS)

5.3 Setting and Configuration

To build this application, I installed several NuGet packages in Visual Studio 2022. These packages provide the essential libraries and tools that add critical functionality to the project.

1. SendGrid

This package is used to send the emails from the application. I used it to connects the application to the SendGrid email service to send automated emails, such as account confirmation links, password reset requests to users and teachers.

2. BCrypt.Net-Next

This is a security package used for hashing passwords. This library encrypts the password text into a secure, unreadable format (a hash) before saving them to the database to protects the user passwords.

3. HtmlAgilityPack

This package helps me to process and manipulate HTML content. It is used to analyze the HTML code created by the rich text editor. For example, it can find and extract all images embedded in a student's essay answer for display on the results page.

4. Microsoft.AspNetCore.Http.Features

This package will provides advanced control over HTTP requests and responses. I used it to configure settings for file uploads, such as increasing the maximum allowed file size when students upload documents for their assignments.

5. Microsoft.AspNetCore.Identity.UI

This package provides pre-built user interface (UI) pages for user management. It automatically provides ready-to-use web pages for common tasks like user registration, login, password reset, and profile management, help me saving significant development time.

6. Microsoft.EntityFrameworkCore

This is the main package for Entity Framework Core (EF Core), which is an Object-Relational Mapper (ORM). It help me to link between the C# code and the SQL Server

database to allows developers to work with database data using C# objects instead of writing complex SQL queries manually.

7. Microsoft.EntityFrameworkCore.Design

This is a design-time tool for EF Core. It provides the commands needed to create and apply database migrations.

8. Microsoft.EntityFrameworkCore.SqlServer

This is the database provider package for EF Core. It enables EF Core to communicate specifically with a Microsoft SQL Server database. It translates the C# operations into SQL Server commands.

9. Microsoft.EntityFrameworkCore.Tools

This package adds useful commands to Visual Studio to use the Package Manager Console to run commands like Add-Migration and Update-Database to create and update the database schema.

10. Microsoft.VisualStudio.Web.CodeGeneration.Design

This package contains scaffolding tools. Scaffolding is a technique used to automatically generate basic code for standard operations (Create, Read, Update, Delete - CRUD). This speeds up development by creating starter pages for managing data, like courses or quizzes.

5.4 System Operation

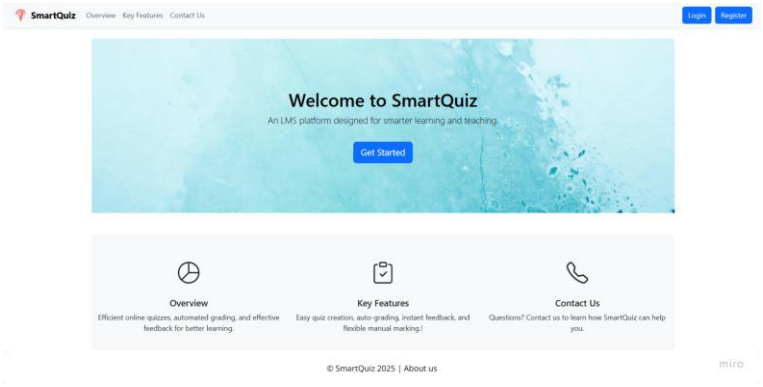


Figure 5.4.1 Index model

-This is the **homepage** of the SmartQuiz Learning Management System (LMS).
It welcomes users with a friendly message and a background image.

-There is a **"Get Started"** button that takes users to the **register page.**

-There are Login and Register button also at the top that takes users to the registrer page and login page.

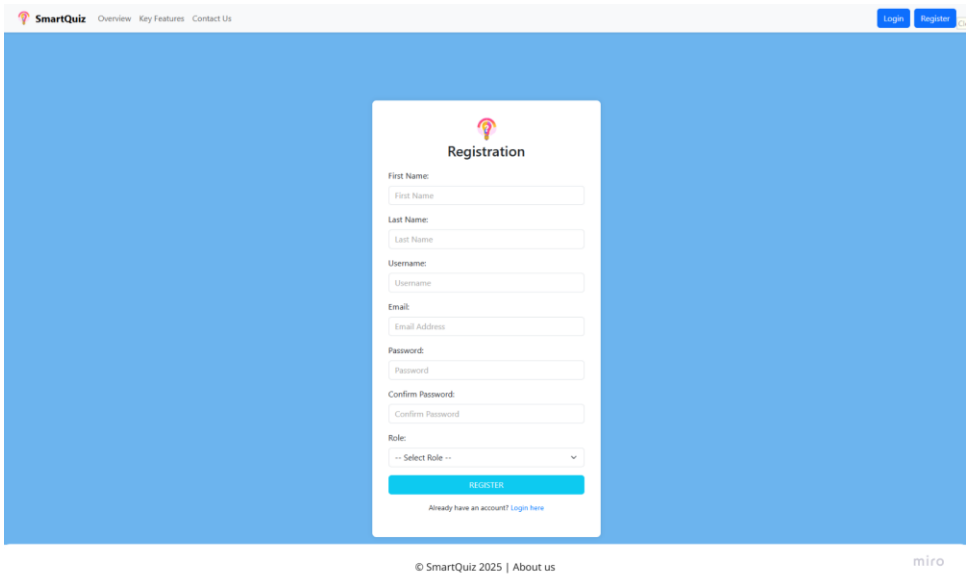


Figure 5.4.2 Register Model

- This page let new users to **create an account.**

-Users must fill in their **first name**, **last name**, **username**, **email**, **password**, and **role** (Teacher or Student).

-After that, the system will send the email confirmation link to the email account that registered just now, and then user click the link to confirm.

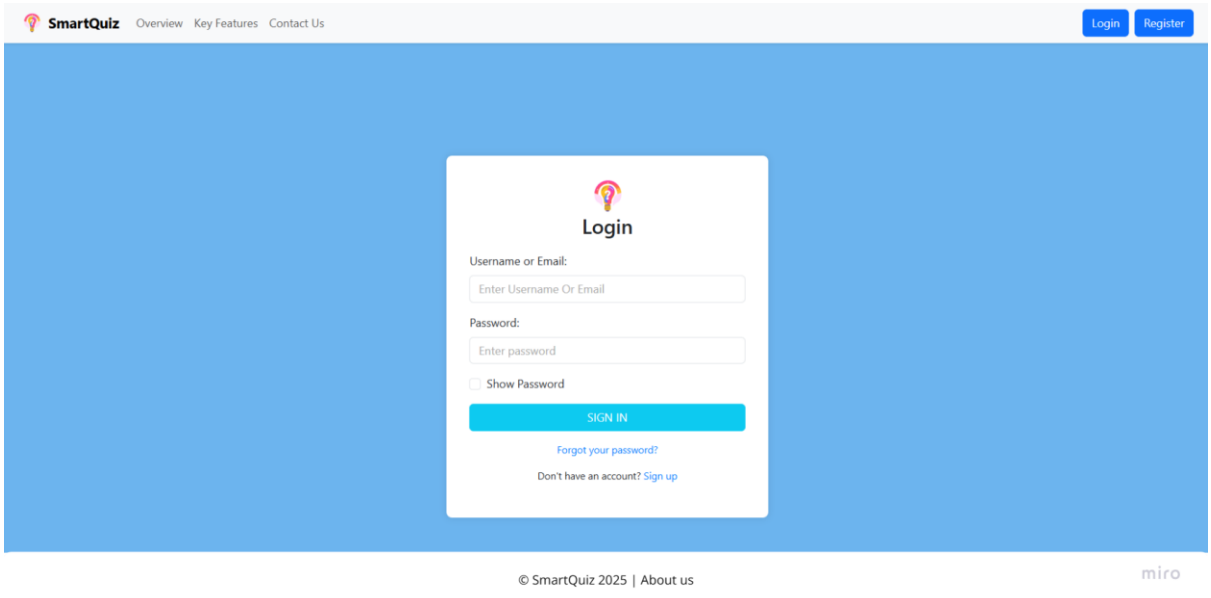


Figure 5.4.3 Login Model

-Figure 5.4.3 is a login page where users can sign in to SmartQuiz. The login form asks for Username or Email and Password (with an option to **show or hide** the password) to login.

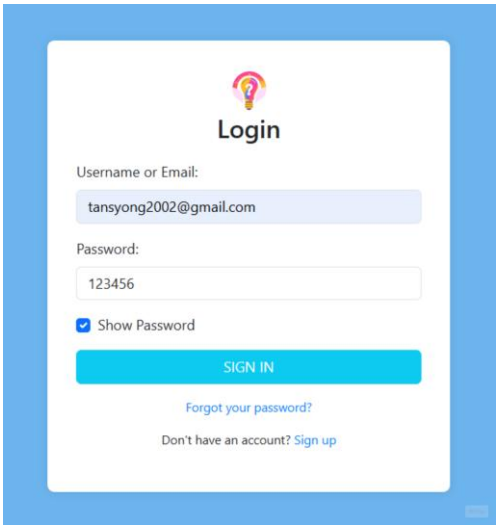


Figure 5.4.4 Login Test Case

-The user can either insert their Username or email address to login to the teacher's dashboard page.

-If a message (like “Account created successfully”) is available, it appears at the top.

-If login fails, **error messages** will appear to tell the user what went wrong.

-There is a forgot your password links on the bottom of login form to bring user to forgot password page when they forgot their password.

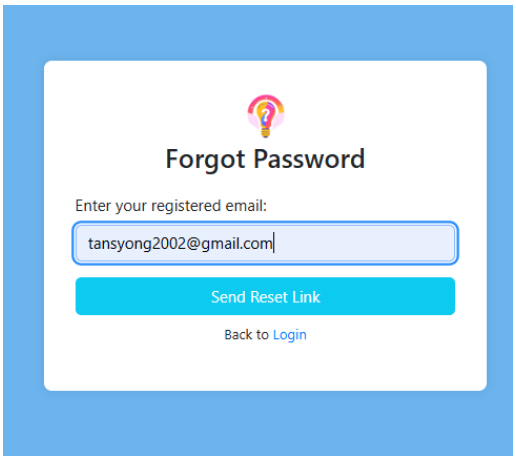


Figure 5.4.5 Forgot Password

- Figure 5.4.5 is the forgot password page that helps users **reset their password** if they forgot it.

- The form asks the user to **enter their registered email address**.

-Then, reset link will be sent to user email account.

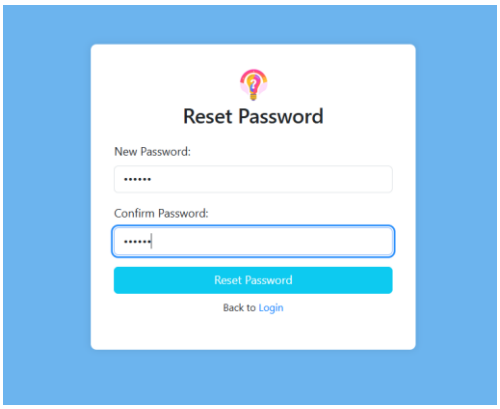


Figure 5.4.6 Reset Password

-The user needs to click on the reset link and navigate to Reset Password page to enter their new password.

- This page lets users **set a new password** after they request a password reset.

- The form asks the user to enter a **new password** and **confirm it.**

- When the user clicks "**Reset Password"**, the system saves the new password.

Teacher System Operation

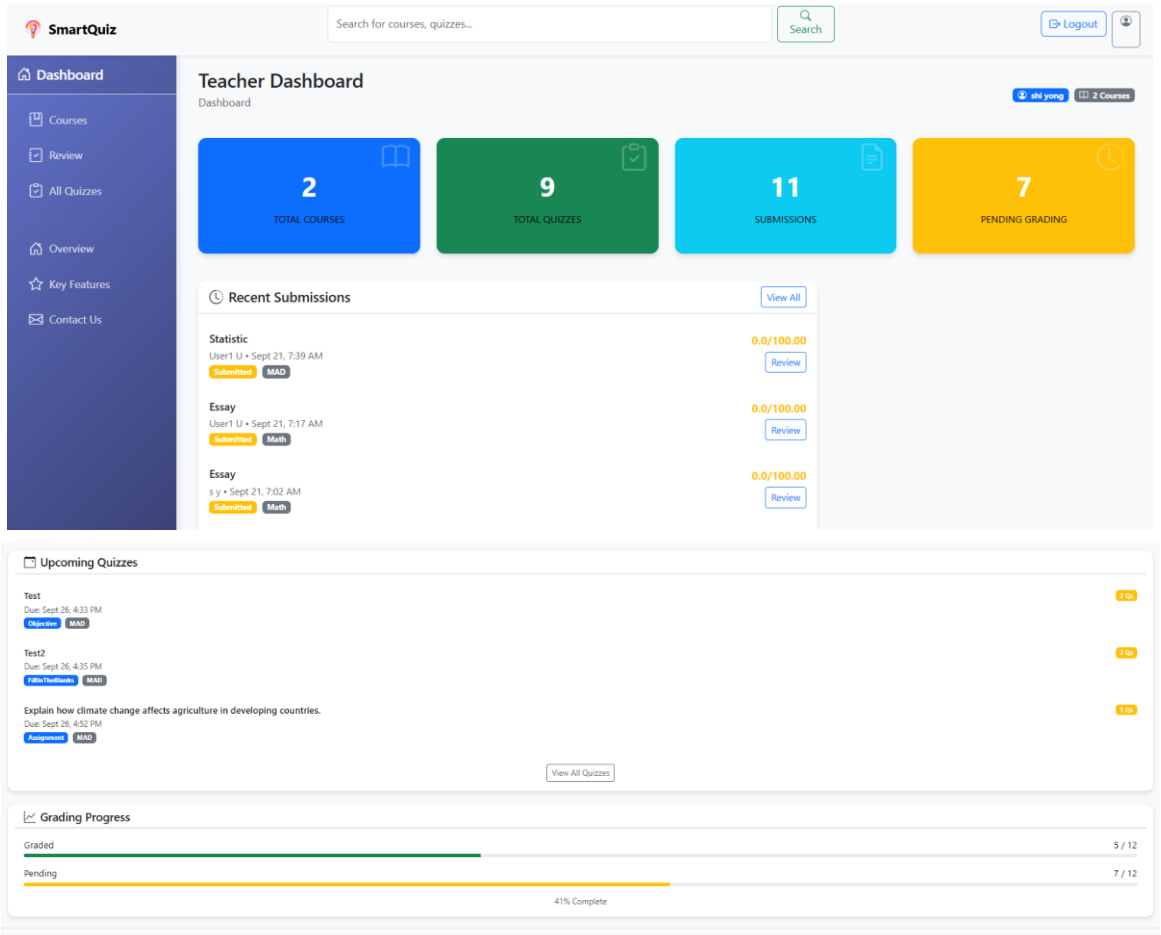


Figure 5.4.7 Teacher's Dashboard

- This page is the **main page** for teachers after they log in, it provides an overview of their workload, recent activity, Upcoming Quizzes and Grading Process.

-Teachers can choose where they would like to navigate to the Courses, Review, or Quizzes page by using the sidebar or buttons at the dashboard main page.

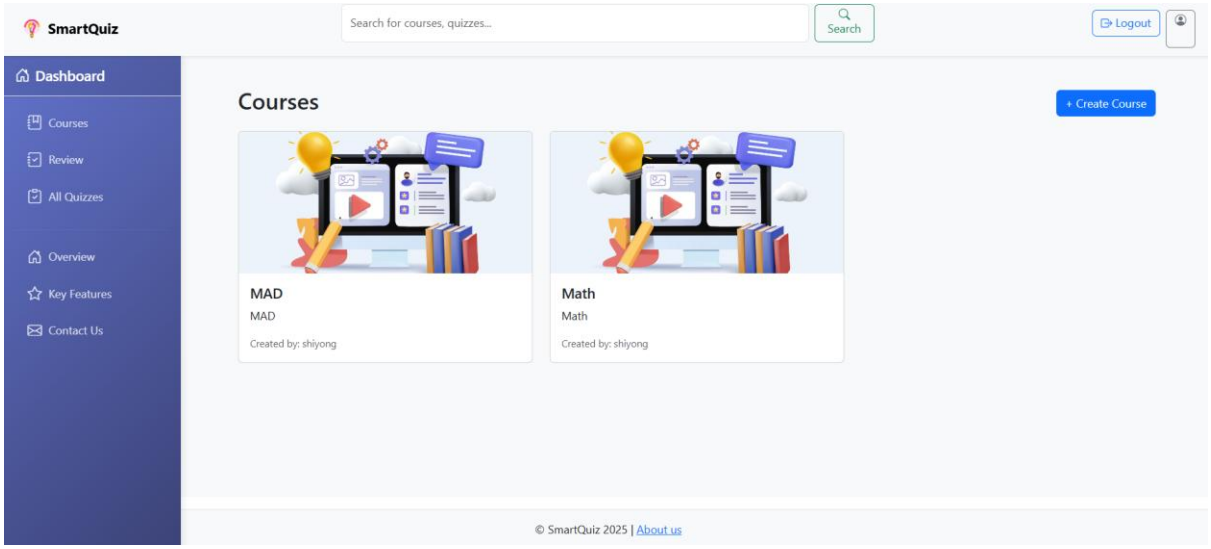


Figure 5.4.8 Courses model

-This page lets teachers **view, create, and manage their courses**.

-At the top, there's a **"Create Course"** button navigate to Create page to add a new course.

-Each course is shown in a card with a **picture**, **name**, **description**, and **creator's name**.

-If no courses exist, a message is shown saying no courses have been created yet.

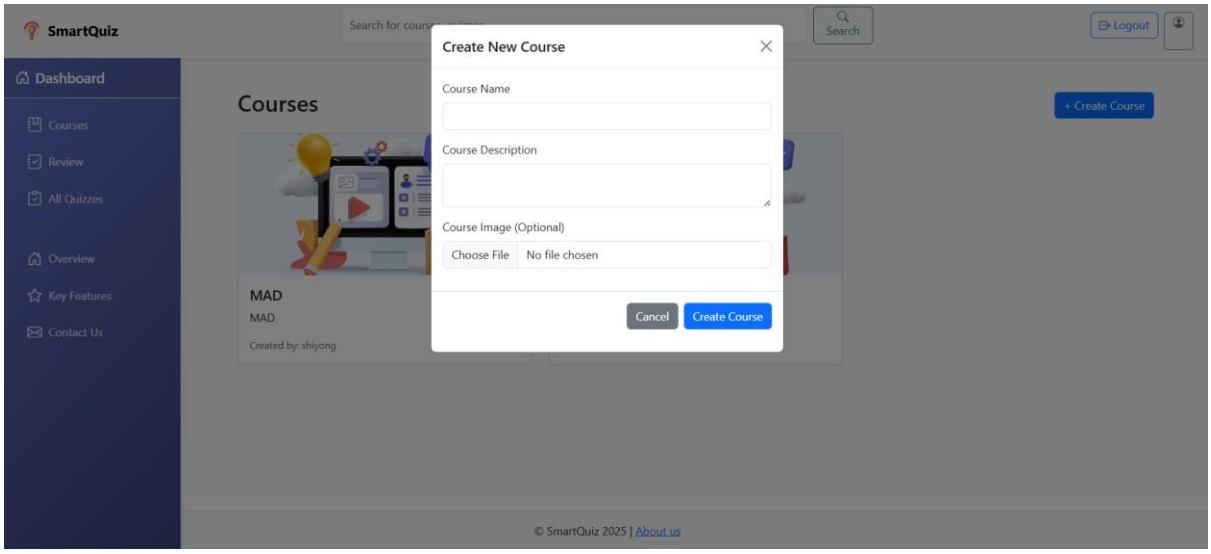


Figure 5.4.9 Create Courses form

-This page allows teachers to **add a new course** by inserting **course name** and a **description.**

-They can also upload an **optional image** for the course.

-After filling in the details, clicking the **"Create Course"** button will save it and there is also a **"Cancel"** button to go back without saving anything.

-After created a course, teacher can click on the course card and navigate to CreateQuiz page.

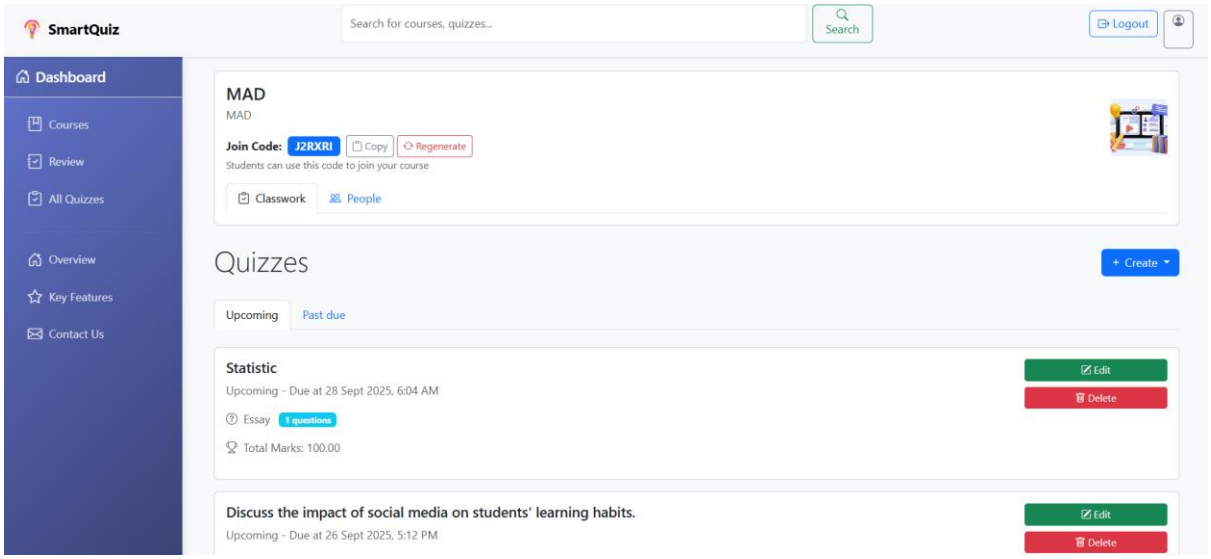


Figure 5.4.10 CreateQuiz Model

-This is a page where teachers can manage their quizzes in a Learning Management System.

There is a button (+Create) that lets teachers create different types of questions for their quiz, like **Objective, Assignment, Essay, and Fill-in-the-Blanks**.

-There is the auto generated join code for this course, the teacher can **share this join code** to the students to let them join into this course.

-The page has two tabs to display quizzes in different stages which is **Upcoming and Past Due**.

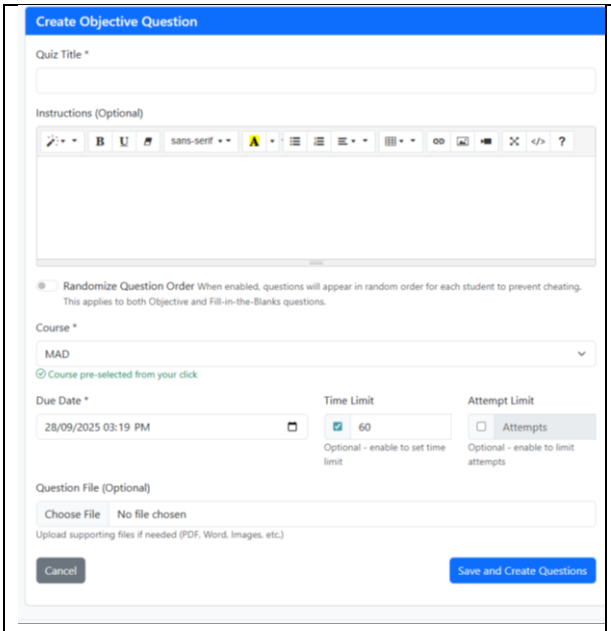


Figure 5.4.11 CreateQuestion (Objective)

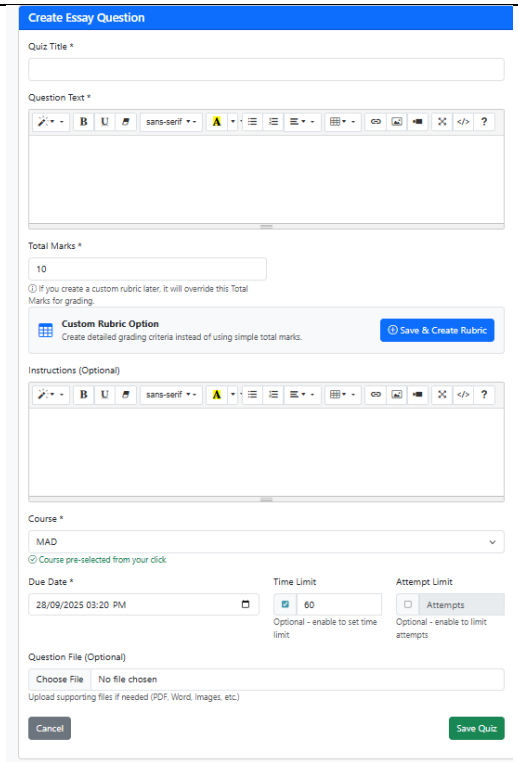


Figure 5.4.12 CreateQuestion (Essay)

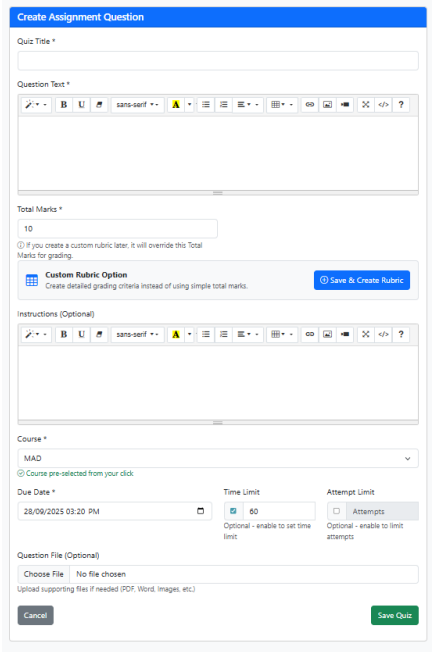


Figure 5.4.13 CreateQuestion (Assignment)

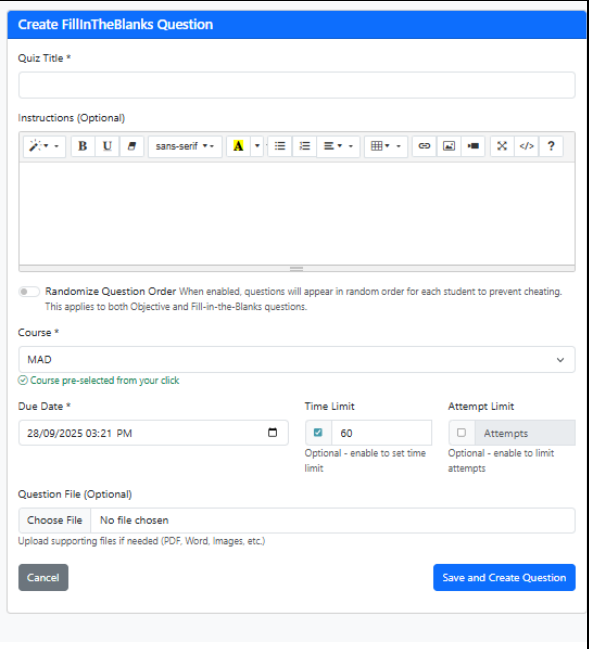


Figure 5.4.14 CreateQuestion (Fill in the Blanks)

-After that, the teacher can choose one of the quiz type (Assignment/Essay/Objective/Fill in the Blanks) and create the quiz for this course.

-For the CreateQuestion page of all type of quiz, the teacher can set the **Time Limits (1-480 minutes)and the Attempt Limit (1- 10 attempts)** to reduce the student cheating rate.

-The CreateQuestion page for the **Objective and Fill in the Blanks** quiz has the **randomize questions order option,** the teacher can choose whether to open this. If open, the order of the question for each of the students will be differenced to reduce the cheating rate.

-The CreateQuestion page for the **Essay and Assignment quiz** has the option of **CreateRubric,** teachers can attach a detailed rubric for grading instead of a simple point value if they want. If they choose to create the rubric, the **rubric** marking system will override the simple **"Total Marks"** value for grading.

-After click on Save and Create Question button for **Objective and Fill in the Blanks** quiz, it will navigate to the next section which is **CreateObjectiveQuestion** page and **FillInTheBlanksQuestion** page.

-For the **Assignment/Essay quiz**, the quiz will be created and assign to the students after click on the Save button.

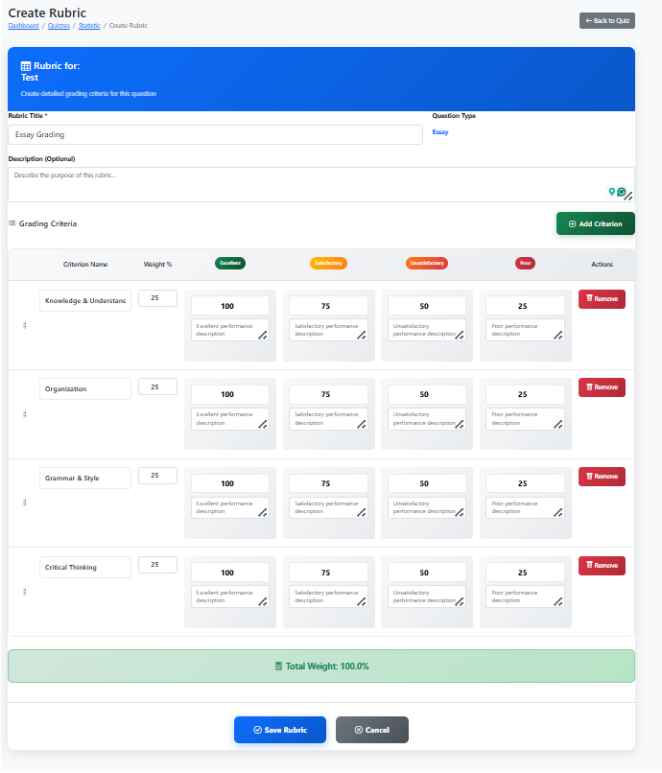


Figure 5.4.15 Create Rubric

-If the teacher clicks on **CreateRubric button** on the CreateQuestion page, it will bring to the CreateRubric page to **create detailed, multi-criteria grading rubrics** for essay and assignment questions.

-For each criterion, the **teacher defines levels** (e.g., Excellent, Satisfactory, Poor) with descriptions and point values.

-Each criterion can be assigned a **weight (percentage)**, and the system validates that the total weight should **be equals to 100%.**

-The teacher can choose to add or remove the criterion as long as the total weight is 100%.

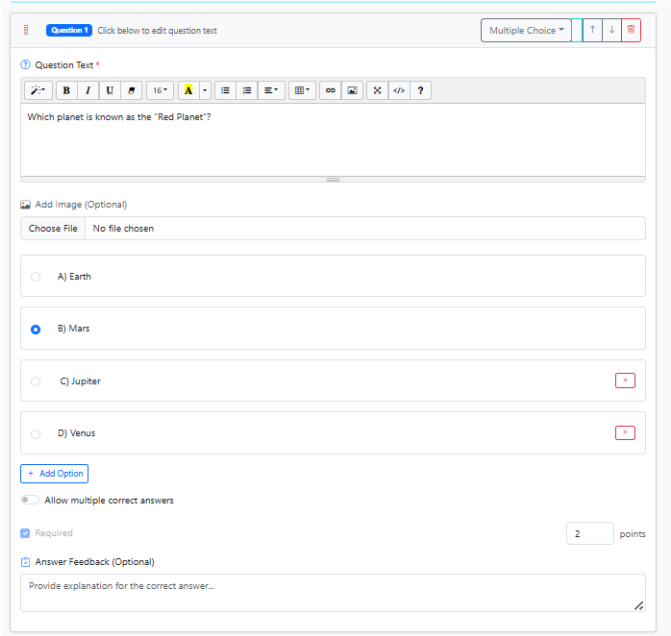


Figure 5.4.16 CreateObjectiveQuestion (Single Correct Answer)

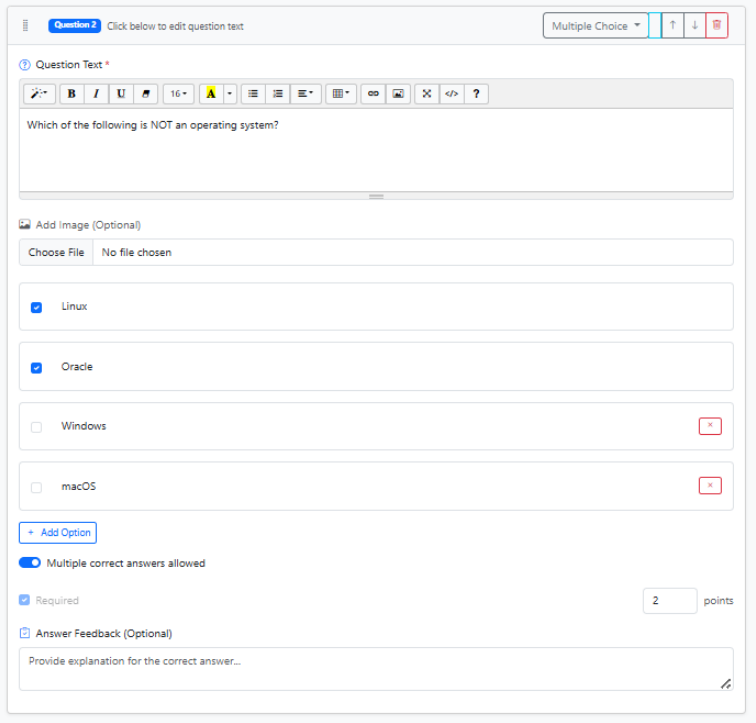


Figure 5.4.17 CreateObjectiveQuestion (Multiple Correct Answer)

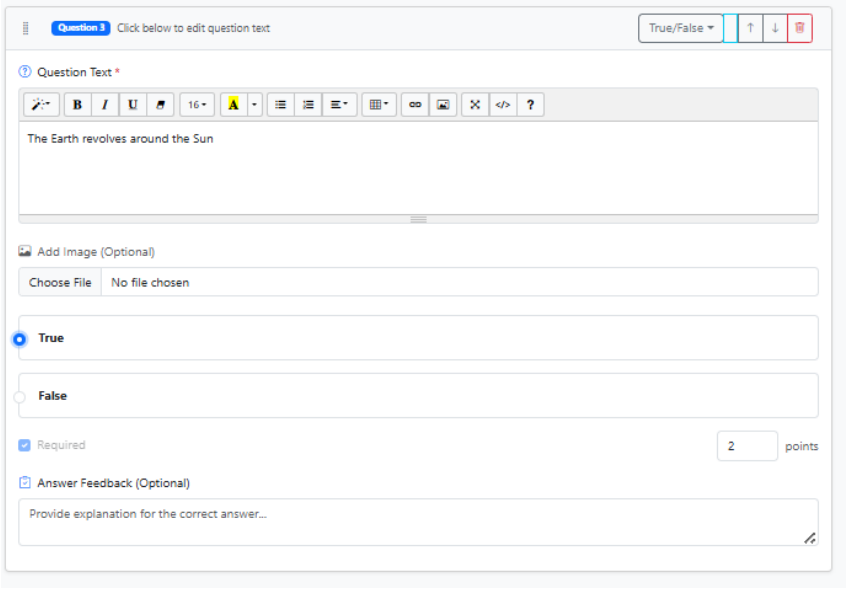


Figure 5.4.18 CreateObjectiveQuestion (True/False)

-From **Figure 5.4.16, 5.4.17, and 5.4.18,** teacher can create **Multiple Choice Question (MCQ) and True/False quizzes .**

-Teachers can **add, remove, duplicate, and reorder** questions.

-For each question, the teacher **inputs choices and marks the correct one(s).**

- It supports both **single and multiple correct answers**, for the multiple correct answers, just need to click on the " Allow multiple correct answers" option.

-The system later will use the auto-grade so that the teacher no need to manually marking each question one by one.

-The teacher can also give the **feedback and explanation** for each question that was created, and it will be display **after the student submit their quiz.**

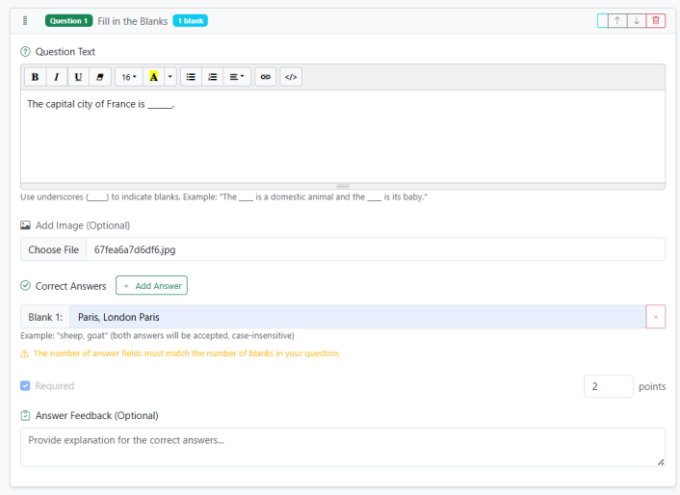


Figure 5.4.19 FillInTheBlanksQuestion (Multiple accepted answer)

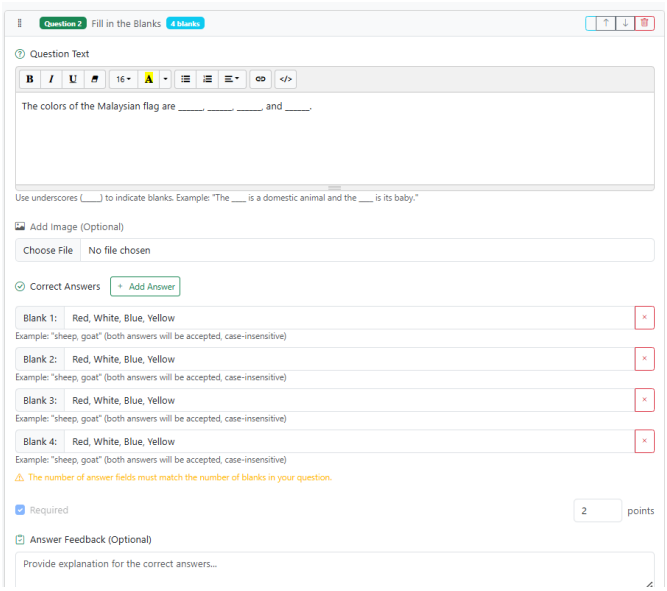


Figure 5.4.20 FillInTheBlanksQuestion (Multiple Blanks Answer)

- From **Figure 5.4.19 and 5.4.20,** the teacher can create questions where students fill in missing words/phrases and provide the **feedback** (explanation of a question) to the students.

- - **Blank Detection:** The system automatically detects blanks in the question text denoted by underscores (_____).
- **Answer Mapping:** The UI generates an input field for each blank, where the teacher enters the correct answer(s). Multiple acceptable answers can be separated by commas (e.g., "WWII, World War Two").

-It is similar to objective questions, it saves the question and correct answers. The system later uses this to auto-grade submissions by comparing the student's input (case-insensitively) to the accepted answers.

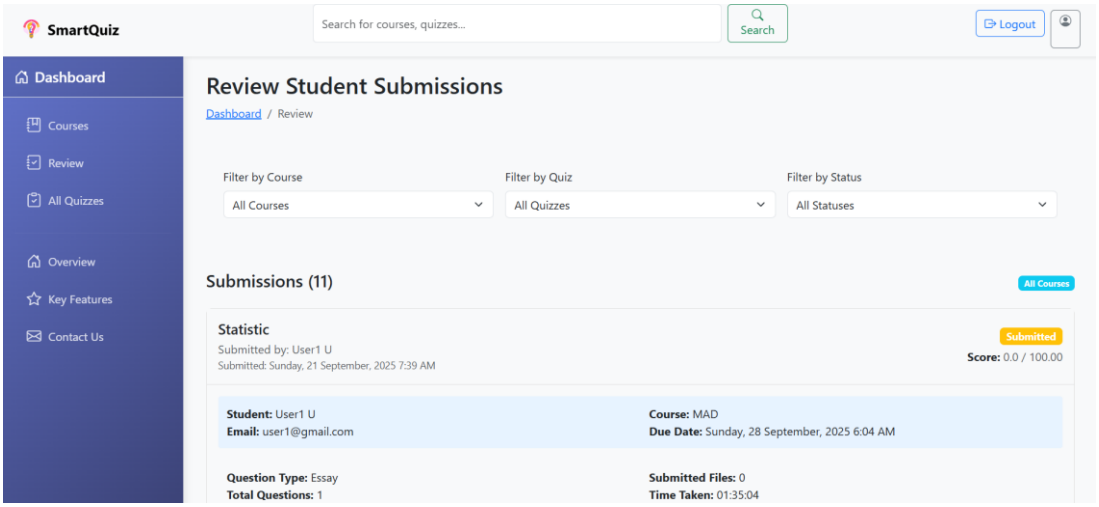


Figure 5.4.21 Review

-After that, In the Review page, the teacher can find, review, and grade student submissions.

-There is a filterable list (by Course, Quiz, Status) of all student submissions. It shows student info, quiz info, score, and status.

-The teacher can click on a quiz submitted by student to do the marking.

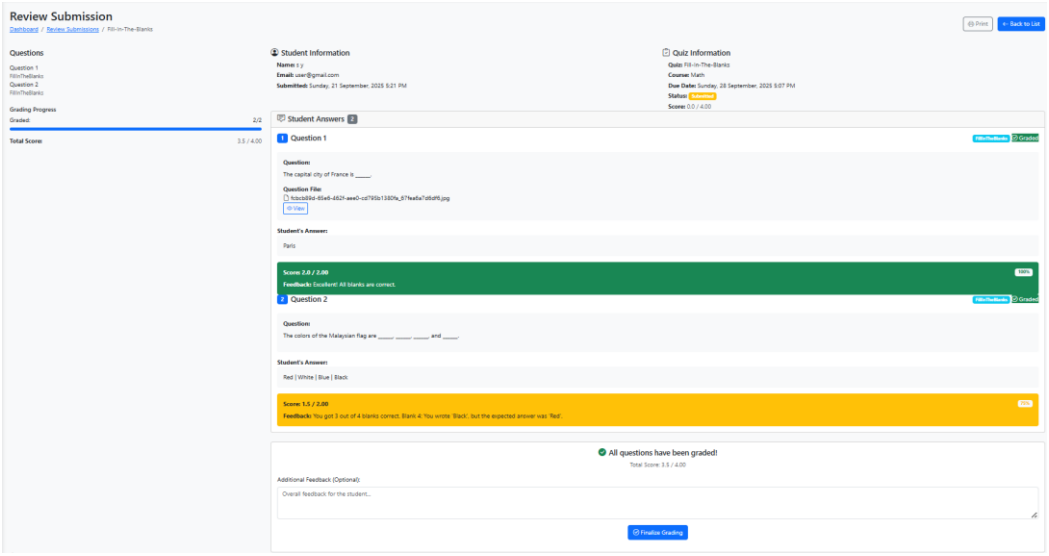


Figure 5.4.22 ReviewSubmission Submitted Status(Fill in the Blanks)

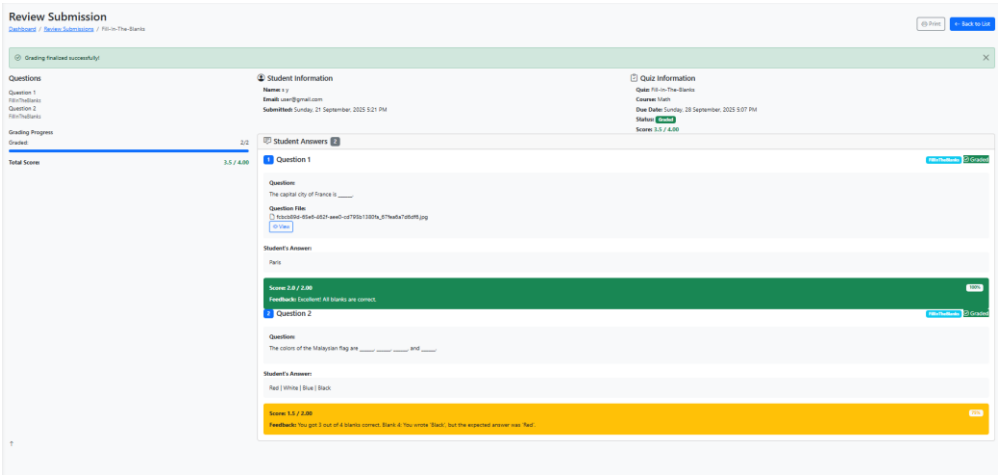


Figure 5.4.23 ReviewSubmission Graded Status (Fill in the Blanks)

- Objective and Fill-in-the-blanks questions are **automatically graded** upon submission. The teacher can see the result and just need to **give the Additional Feedback and click on the Finalize Grading** to change the status from **Submitted to Graded** (The score will be updated and return back to student).

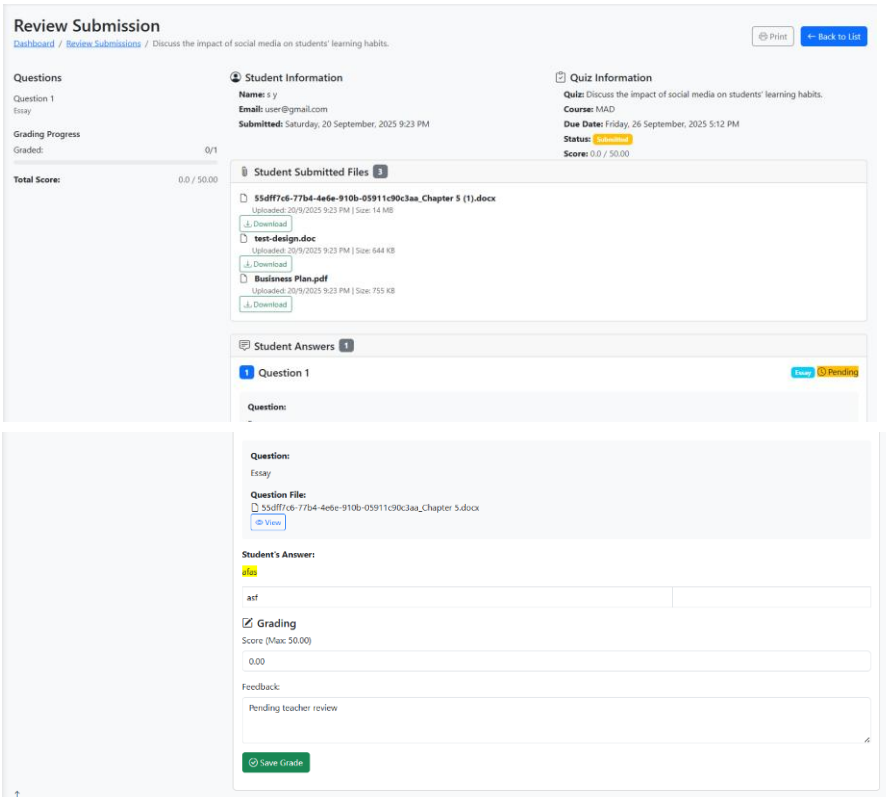


Figure 5.4.24 Review Submission (Essay)

-Teachers can download any files submitted by the student.

- For essays and assignments, the teacher can give the score manually.

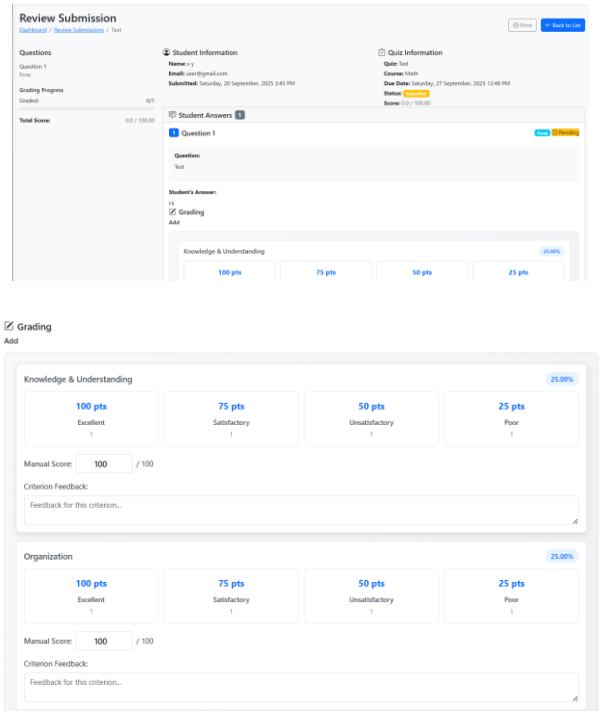


Figure 5.4.25 Review Submission (Essay)

- If a **rubric is attached, the** teacher can grade by selecting performance levels for each criterion, and the score is calculated automatically.

Student System Operation:

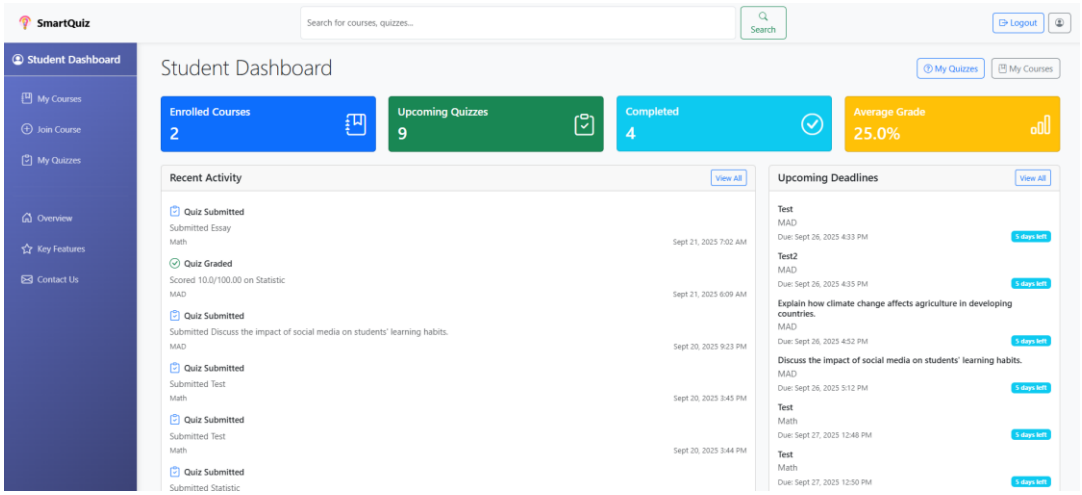


Figure 5.4.26 Student Dashboard

-After student login will navigate to student's Dashboard, which provides a personalized overview of their academic activity and performance.

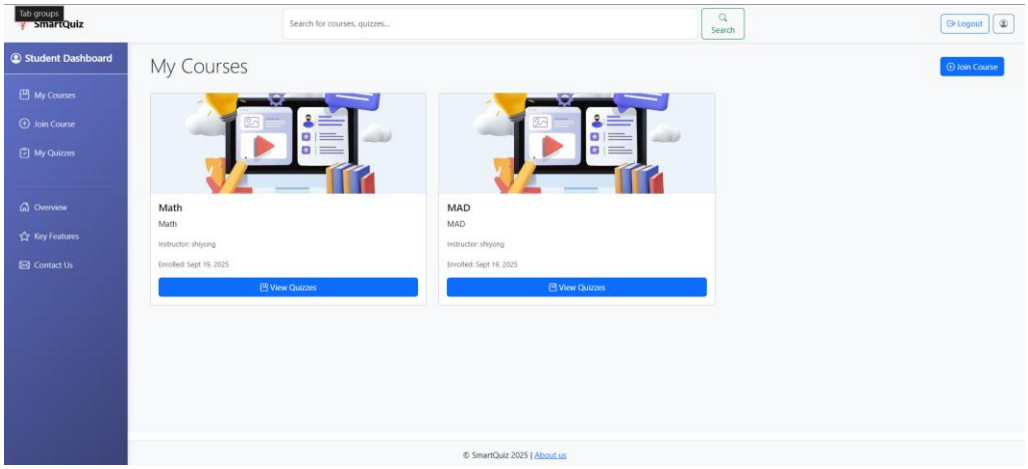


Figure 5.4.27 MyCourses

-This MyCourses page displays a simple list of all courses the student is currently joined in.

-Students can click on the MyCourses page Join Course button or navigate to Join Course page.

-The students **can click on the course to navigate to CourseQuizzes** page.

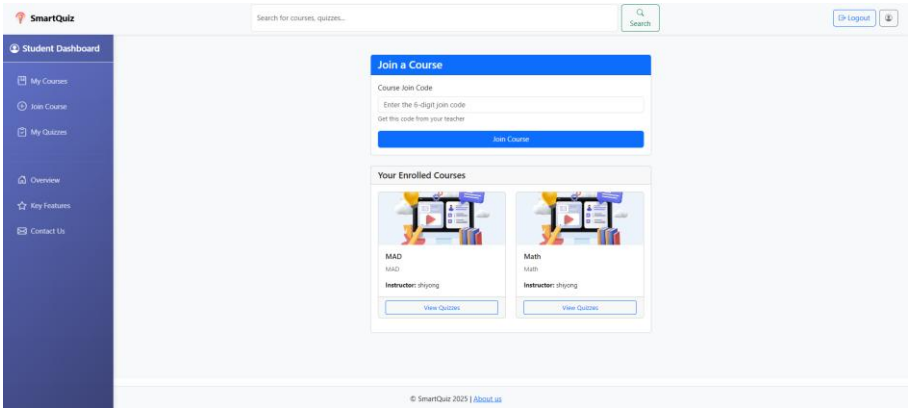


Figure 5.4.28 JoinCourse

-This page is to let the student **to join into a new course using the Join Code (6-digit)** given by teachers.

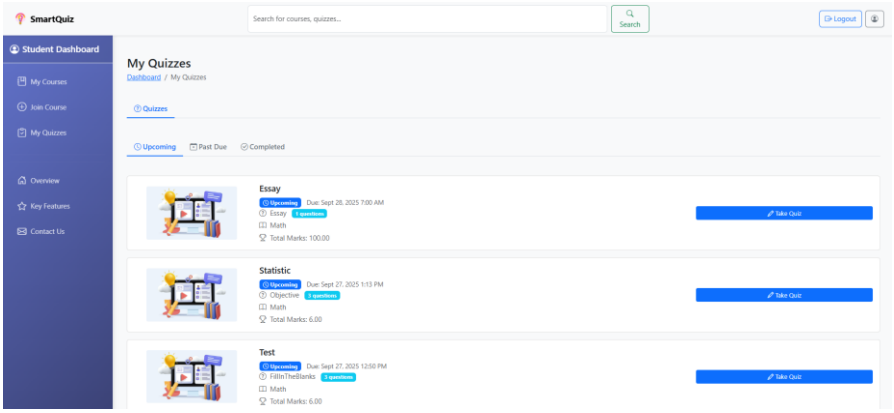


Figure 5.4.29 MyQuizzes

-This is a central page for students to see all quizzes across all their courses, categorized by status.

- Quizzes are organized into three tabs:

- **Upcoming:** Quizzes that are not yet due and the student hasn't submitted.
- **Past Due:** Quizzes that are past their due date and the student hasn't submitted.
- **Completed:** Quizzes that the student has submitted (graded or pending grading).

-It provides buttons like **"Take Quiz" (for upcoming), "View Results" (for completed), or a disabled "Quiz Closed" (for past due).**

-The students can take the quiz that was created by teacher in this course.

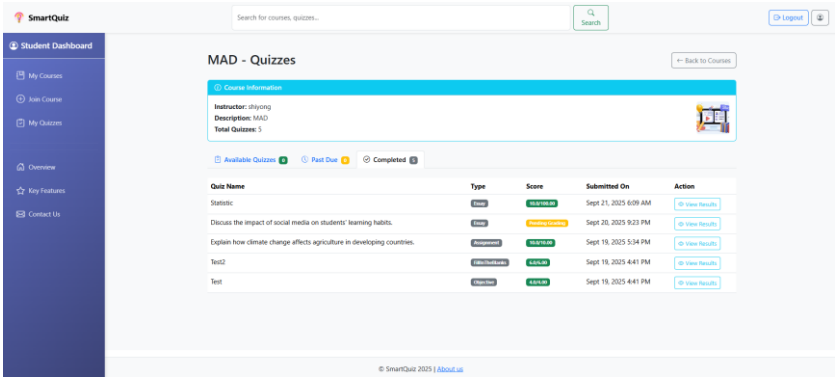


Figure 5.4.30 CourseQuizzes

-This page is a detailed view of quizzes within a **single, specific course**. This is a filtered version of "MyQuizzes.".

-It is almost same with the "MyQuizzes" but the is limited to the course selected from the "My Courses" page. It shows the same three-tab structure (Available, Past Due, Completed) but only for the selected course.

-The student also can attend their quiz at this page in the Available Tab.

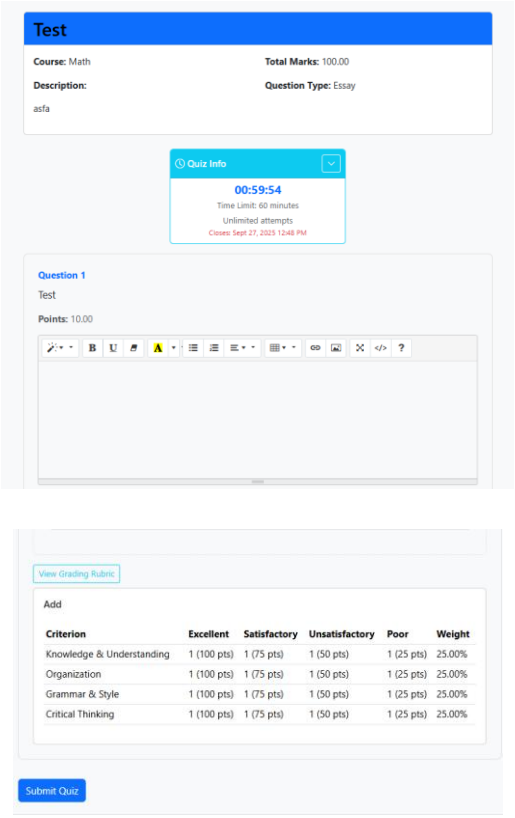


Figure 5.4.31 TakeQuiz (Essay/Assignment)

-In TakeQuiz page, it displays the quiz name, description, and a prominent timer counting down if a time limit is set and also the limit of the attempts.

-The quiz will automatically submit when the timer reaches zero.

-The student can view the grading rubric for essay questions before answering to let them have the idea to answer their questions.

-TakeQuiz page also allow students to upload supporting files for their answers.

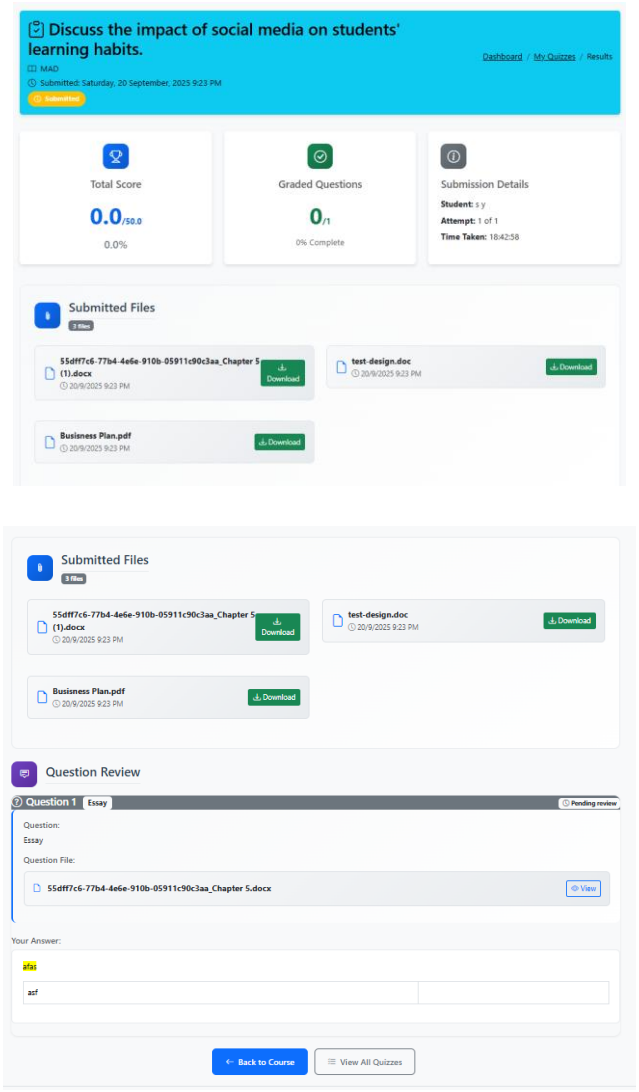


Figure 5.4.32 QuizResult Submitted Status (Assignment/Essay)

-The QuizResult page will display detailed results and feedback after a student submits (Submitted Status) a quiz or after a teacher grades it (Graded Status).

-In the Submitted Status, student will not see the total score as the teacher have not grade yet.

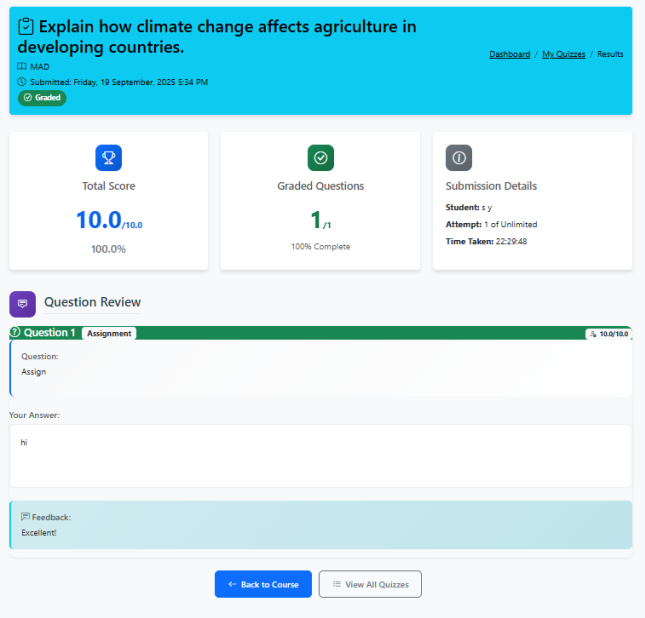


Figure 5.4.33 QuizResult Graded Status (Assignment/Essay)

-In the Graded Status, student will see the total score as the teacher have do the Finalize grading.

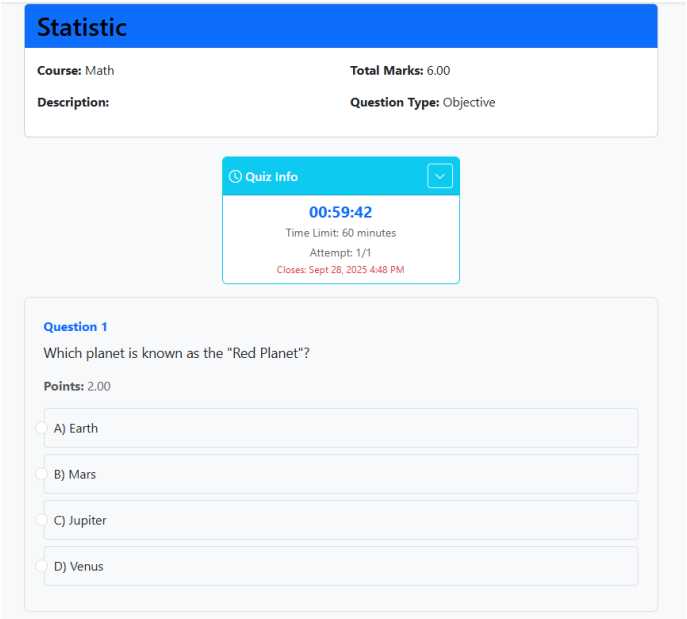


Figure 5.4.34 TakeQuiz (Objective)

-This is the TakeQuiz page for the Objective Questions which renders radio buttons or checkboxes to let students answer it.

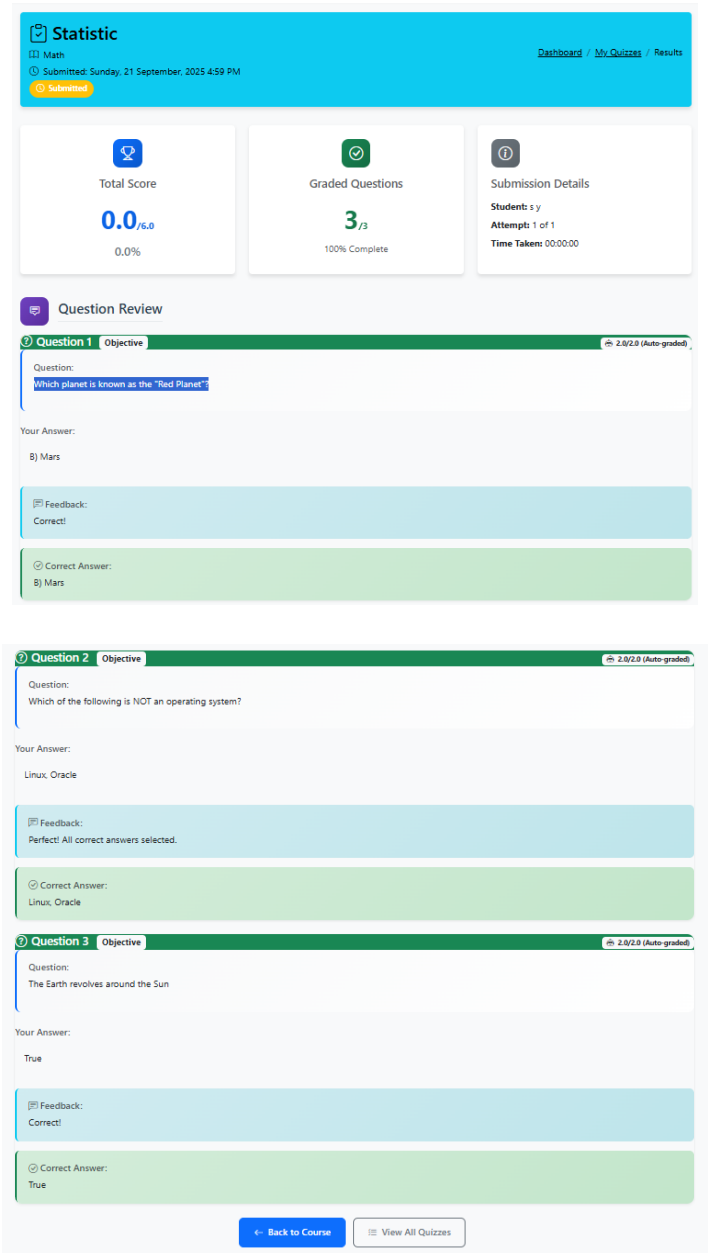


Figure 5.4.35 QuizResults Submitted Status (Objective)

-The Student can directly see the **result, feedback and the correct answer** of the Objective Questions but the Total Score needs to wait teacher to do finalize grading.

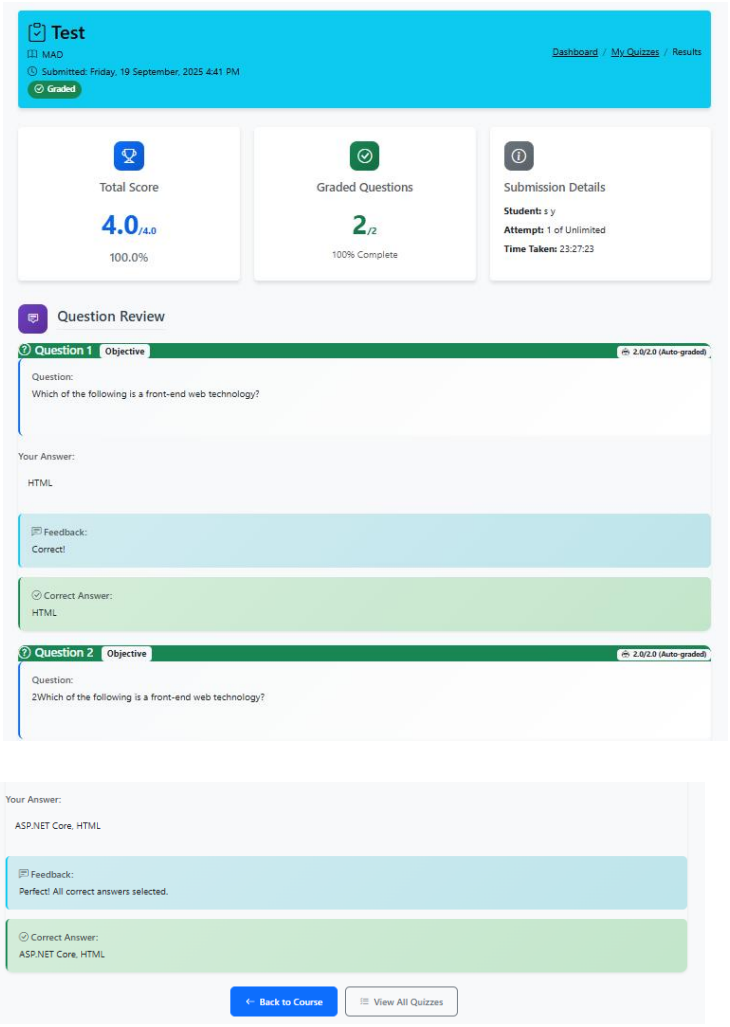


Figure 5.4.36 QuizResult Graded Status (Objective)

-After teacher do the finalize grading, student can see the total score.

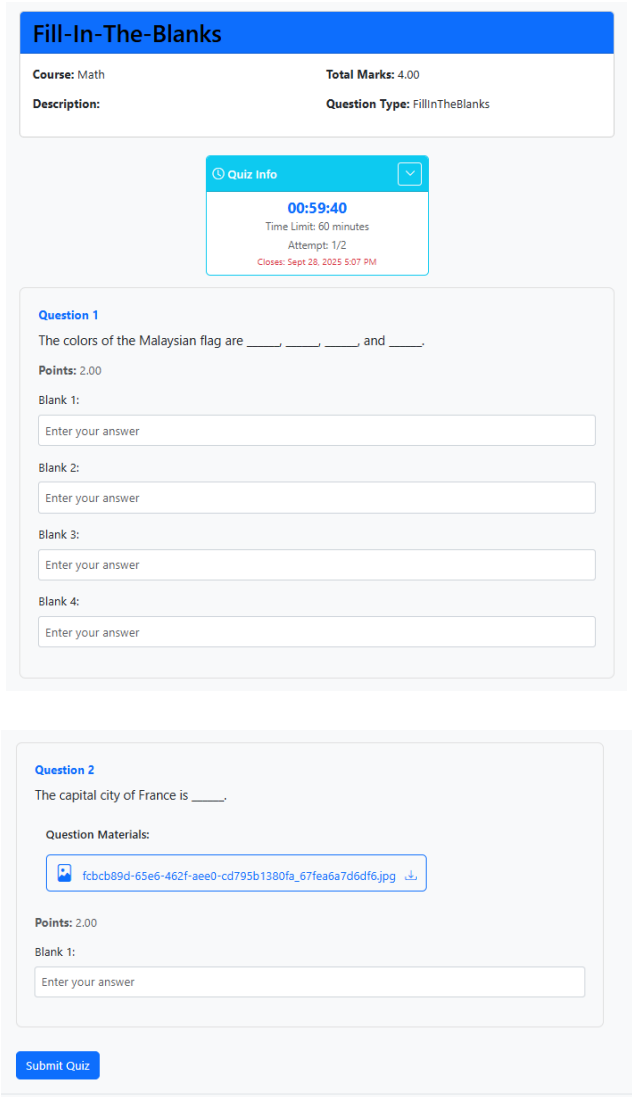


Figure 5.4.37 TakeQuiz (Fill in the Blanks)

-Compared with Figure 5.4.19 and 5.4.20, randomization has changed the order of the questions.

-The student just needs to enter the answer to fill in the blanks questions with no case sensitive.

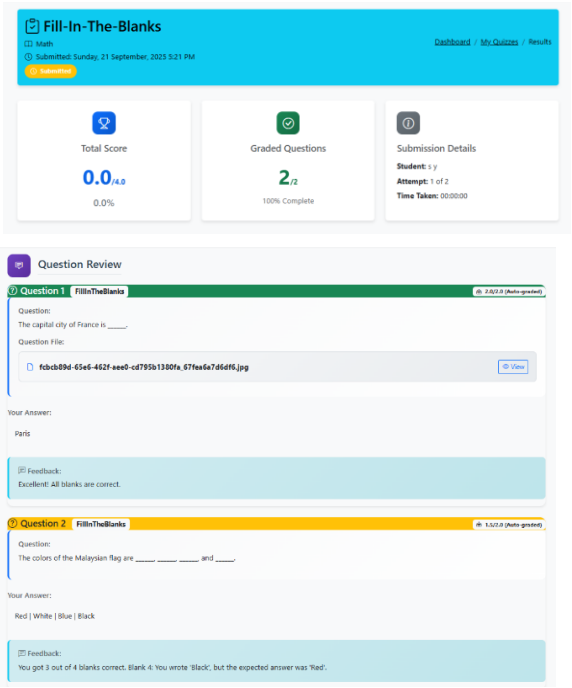


Figure 5.4.38 QuizResult Submitted Status (Fill in the Blanks)

-In submitted status, it also can see the result of the correct answer and feedback.

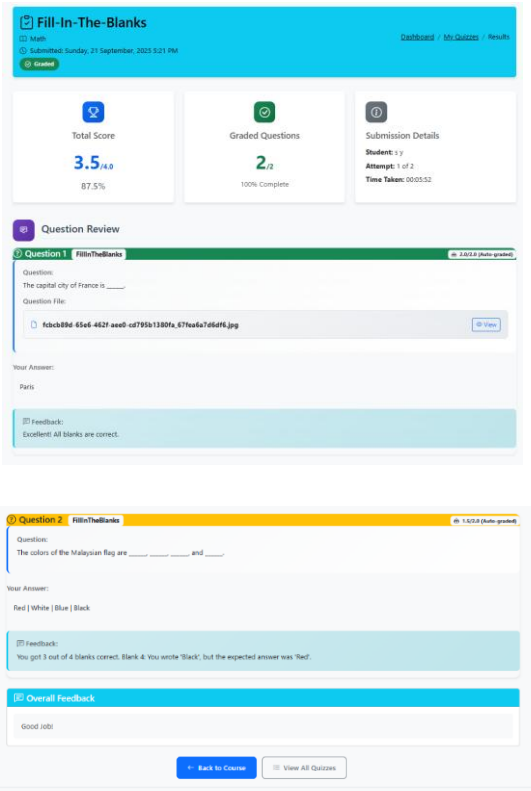


Figure 5.4.39 QuizResult Graded Status (Fill in the Blanks)

-The **Total Score and Addition Feedback** will be displayed after teacher do the **Finalize Grading.**

5.5 Implementation Issues and Challenges

For this project, there might be a few issues when we try to build it. One issue could be making the quiz creation tool easy for teachers to use, even if they are not very good with computers, while still having all the advanced features they need.

Another problem when building this system could be with the database. Sometimes, when we make changes to how we want to store this information like adding new sections or changing how things are organized, we need to update the database. It can be tricky to do these database updates smoothly. For example, if we want to add a new type of question to the quizzes, we need to make sure the database knows how to store the answers for this new question. If something goes wrong during this update, it could cause problems with the existing information or even stop the system from working correctly. Thus, making sure these databases updates happen without any hiccups can be a challenge during the project.

5.6 Concluding Remark

This chapter has detailed the successful implementation of the SmartQuiz Learning Management System. The development was carried out on a capable hardware setup, ensuring smooth performance. The necessary software and NuGet packages were installed and configured to provide essential functionalities like security, database management, and automated email services.

CHAPTER 6: SYSTEM EVALUATION AND DISCUSSION

6.1 System Testing and Performance Metrics

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Fill out the form with valid data.	**First Name:** Shi **Last Name:** Yong **Username:** ShiYong **Email:** tansyong2002@gmail.com **Password:** User@123 **Confirm Password:** User@123 **Role:** Student	User is redirected to the login page.	User is naviagte to the login page. and new account will record in the database.	Pass
2	Fill out the form with mismatched passwords.	**Password:** User@123 **Confirm Password:** User@321	An error message is displayed.	An error message "Password and confirmation do not match. " is displayed	Pass
3	Leave a required field empty.	Leave **Email** field empty **Other fields:** valid data	A validation error is displayed for the empty field.	A validation error is displayed for the empty field, e.g., "The Email field is required."	Pass
5	Fill out the form with an invalid email format.	**Email:** tansyong2002gmail.com **Other fields:** valid data	A validation error is displayed for the invalid email.	A validation error is displayed for the invalid email.	Pass

Table 6.1.1 Registration Test Case

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Log in with a valid username and password.	**Username:** ShiYong **Password:** User@123	User is navigating to dashboard by their role.	User navigate to the correct dashboard.	Pass
2	Log in with a valid email and password.	**Email:** tansyong2002@gmail.com **Password:** User@123	User is redirected to the correct dashboard.	User redirected to the correct dashboard.	Pass
3	Log in with an incorrect password.	**Username/Email:** Shi Yong **Password:** User123	An error message is displayed on the login page.	An error message "Invalid username/email or password." is displayed on the login page.	Pass
4	Log in with a non-existent username.	**Username:** Tan **Password:** User@123	A validation error " Invalid username/email or password." is displayed.	A validation error " Invalid username/email or password." is displayed.	Pass
5	Leave the username/email field empty.	**Username/Email:** (empty) **Password:** SecurePass123!	A validation error is displayed for the empty field.	A validation error is displayed for the empty field, "The Password field is required. " and "The UsernameOrEmail field is required."	Pass
6	Leave the password field empty.	**Username/Email: johndoe Password: (empty)**data	A validation error "The Password field is required." is displayed.	A validation error "The Password field is required." is displayed.	Pass
7	Verify "Show Password" functionality.	Click the **"Show Password"** checkbox.	The password field's input type changes from 'password' to 'text', revealing the entered characters.	The password User123 is showed.	Pass

Table 6.1.2 Login Test Cases

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Leave the Course Name field empty when create new course	**CourseName;** Empty	An error message is displayed on the Courses page.	An error message " The CourseName field is required.." is displayed on the Create course page.	Pass
2	Leave the Quiz title empties in Create Question page.	**Quiz Title:** Empty	An error message is displayed on the CreateQuestion page.	An error message " Quiz name is required..." is displayed on the CreateQuestion page.	Pass
3	Leave the Question Text empty in Create ObjectiveQuestion page	**Quiz Text:** Empty	An error message is displayed on the Create ObjectiveQuestion page.	An error message " Question text is required." is displayed on the Create ObjectiveQuestion page.	Pass
4	Do not select the correct answer for single correct answer choices.	**No select the radio button**	An error message is displayed on the Create ObjectiveQuestion page.	An error message " Please select at least one correct answer" is displayed on the Create ObjectiveQuestion page.	Pass
5	Enter a wrong format for Fill in the Blanks question test.	**Question Test;** Where is the London?	An error message is displayed on the FillInTheBlanksQuestion page.	An error message " Question should contain underscores (_) to indicate blanks." is displayed on the Create ObjectiveQuestion page.	Pass

Table 6.1.3 Teacher module Test Cases

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Enter a invalid and wrong JoinCode	**Join Code:** JJDIST	An error message is displayed on the JoinCourse page.	An error message " Invalid join code. Please check with your teacher.." is displayed on the JoinCourse page.	Pass
2	Join Course with Valid Code	**Quiz Title:** KSOFUS	The student is now enrolled in the new course.	The newly joined course appears in the "Your Enrolled Courses" list on the JoinCourse page	Pass
3	Join Already Enrolled Course	**Join Code:** 8YEIVV (already enrolled in)	An error message is displayed on the JoinCourse page.	An error message " You are already enrolled in this course." is displayed on the JoinCourse page.page.	Pass
4	View My Courses	**1. Navigate to the My Courses page** (/Student/MyCourses)**. 2. Verify the list of courses.**	The page displays a card for each course the student is enrolled in.	The page displays a card for each course the student is enrolled in.	Pass
5	My Courses with No Enrollments	**1. Log in with a student account that has no enrollments. 2. Navigate to the My Courses page** (/Student/MyCourses).	The page displays a message indicating that there are no enrolled courses.	The page displays a message"You haven't joined any courses yet Join a Course to get started! " indicating that there are no enrolled courses.	Pass

Table 6.1.4 Student module Test Cases

6.2 Objectives Evaluation

Objective 1: To develop an Advanced Quiz System with an intuitive interface for educators.

- **Evaluation:** This objective was successfully achieved. The system allows teachers to easily create four different types of quizzes: Multiple Choice (Objective), True/False, Fill-in-the-Blanks, and Essay/Assignment. The interface is a step-by-step form that guides the teacher through the process. Teachers can set time limits, attempt limits, and randomize the order of questions to reduce cheating, just as planned. The creation process for each question type is straightforward, making quiz building much easier than in many existing systems.

Objective 2: To implement a Real-Time Feedback Mechanism for students.

- **Evaluation:** This objective was also successfully met. For objective quizzes (Multiple Choice, True/False, Fill-in-the-Blanks), students receive their results immediately after submission. They can see which answers they got right or wrong and view the correct answers along with any explanation the teacher provided. For subjective quizzes (Essay/Assignment), while instant grading is not possible, the system is designed to show feedback as soon as the teacher completes the manual grading and finalizes the results. This provides students with clear and timely feedback on their performance.

Objective 3: To create an Automated and Customizable Grading system.

- **Evaluation:** This objective was fully accomplished. The system automatically grades all quiz (Multiple Choice, True/False, Fill-in-the-Blanks), which saves teachers a significant amount of time. For essays and assignments, teachers have two flexible options: they can grade using a simple total score or create a detailed, multi-criteria rubric. When a rubric is used, the system automatically calculates the total score based on the teacher's selections. This allows for fair, consistent, and customizable grading, exactly as intended.

6.3. Concluding Remark

The developed SmartQuiz LMS has successfully fulfilled all three primary project objectives. It provides students anduiz creation tool, delivers instant and detailed feedback to students, and combines both automated and highly customizable manual grading to meet the needs of modern online education.

CHAPTER 7 CONCLUSION AND RECOMMENDATION

7.1 Conclusion

This project set out to solve real problems that teachers and students face with current Online Learning Management Systems (LMS) like Moodle and Google Classroom. The main issues were that creating quizzes and grading assignments took too much time, and students often did not get feedback quickly enough.

To fix this, I successfully built a new web-based LMS called "SmartQuiz." This system focuses on making quizzes and grading easier and faster. The main goals of the project were all achieved:

- **Advanced Quiz System**: I created an easy-to-use tool that lets teachers make different types of quizzes (Assignment/Essay/Objective/Fill-in-the Blanks). Teachers can also add features like time limits and random question orders to make quizzes more secure.
- **Real-Time Feedback:** The system gives students immediate results for objective and fill in the blanks quizzes (like multiple-choice) right after they submit. For essays and assignment, students get their feedback as soon as the teacher finishes grading.
- **Automated and Customizable Grading:** The system automatically grades objective and fill in the blanks questions, this can save the teachers' times. For essays and assignment, teachers can grade with a simple score or use a detailed custom rubric for fair and consistent marking.

In summary, the SmartQuiz system effectively addresses the limitations found in existing platforms. It makes the process of creating quizzes, grading, and giving feedback much more efficient. This helps teachers save time and allows students to learn from their mistakes faster, making online education better for everyone.

7.2 Recommendation

While the SmartQuiz system works well and meets its main goals, there are always ways to make it even better in the future. The first recommendation is future versions could include more interactive question types, like drag-and-drop, matching, or diagram labeling. This would make quizzes more engaging for students.

The second recommendation is the teacher and student dashboards could be enhanced with more detailed charts and graphs. This would help the teachers to track student progress over time and help students see their performance more clearly.

The third recommendation is creating dedicated mobile apps for iOS and Android which will make it easier for users to access their courses and quizzes anytime, anywhere. This is because the current system is a website that works on mobile browsers.

The fourth recommendation is making sure the website is fully accessible for people with disabilities. For example, better screen reader support, keyboard navigation would make the system suitable for more people.

The last recommendation is adding a advanced anti-cheating tools in the future like locking the browser during a quiz, using the webcam to monitor students, or checking for plagiarism in essay answers to make quizzes even more secure.

By working on these recommendations, the SmartQuiz system could become an even more powerful and helpful tool for online education.

REFERENCES

[1] Cao, T. X. L. (2023). Benefits and challenges of using LMS in blended learning: Views from EFL teachers and students at a Vietnamese public university. International Journal of TESOL & Education, 3(3), 78-100.DOI: https://doi.org/10.54855/ijte.23335

[2] S. A. Langove and A. Khan, "Automated Grading and Feedback Systems: Reducing Teacher Workload and Improving Student Performance", *J. Asian Dev. Studies*, vol. 13, no. 4, pp. 202–212, Dec. 2024, doi: 10.62345/jads.2024.13.4.16.

[3] M. Sharifov, S. Safikhanova, and A. Mustafa, "Review of Prevailing Trends Barriers and Future Perspectives of Learning Management Systems (LMSs) in Higher Education Institutions," *International Journal of Education and Development using Information and Communication Technology*, vol. 17, issue 3, pp. 207-216, 2021.

[4] V. M. Bradley, "Learning management system (LMS) use with online instruction," *International Journal of Technology in Education*, vol. 4, no. 1, pp. 68–92, Dec. 2021, doi: https://doi.org/10.46328/ijte.36.

[5] F. T. Al-Dhief, A. A. Nasser, S. M. Tharikh, H. A. Nasser, A. A. A. Al-Mosleh, M. A. A. Albadr, and M. R. M. Alsemawi, "Review of learning management systems: history, types, advantages, and challenges," *Indonesian J. Elect. Eng. Comput. Sci.*, vol. 33, no. 1, pp. 350–360, Jan. 2024, doi: 10.11591/ijeecs.v33.i1.pp350-360.

[6] M. A. Almaiah, A. Al-Khasawneh, and A. Althunibat, "Exploring the critical challenges and factors influencing the E-learning system usage during COVID-19 pandemic," *Education and Information Technologies*, vol. 25, no. 6, pp. 5261–5280, May 2020, doi: https://doi.org/10.1007/s10639-020-10219-y.

[7] "Frappe LMS: The easy to use open-source learning management system | Frappe Blog," *Frappe.io*, 2023. https://frappe.io/blog/product-stories/frappe-lms-the-fun-and-flexible-open-source-lms (accessed Sep. 11, 2024).

[8] S. H. P. W. Gamage, J. R. Ayres, M. B. Behrend, and E. J. Smith, "Optimising Moodle quizzes for online assessments," *International Journal of STEM Education*, vol. 6, no. 1, Aug. 2019, doi: https://doi.org/10.1186/s40594-019-0181-4.

[9] M. Zakaria, K. A. Manaf, H. A. Bustaman, and R. A. Rahman, "Exploring Benefits and Challenges of Adopting Google Classroom in the Perspective of Higher Institution's Learners," *The Mattingly Publishing Co., Inc*, vol. 83, pp. 9739–9749, Mar.-Apr. 2020.

[10] S. H. P. W. Gamage, J. R. Ayres, and M. B. Behrend, "A systematic review on trends in using Moodle for teaching and learning," *International Journal of STEM Education*, vol. 9, no. 9, 2022. [Online]. Available: https://doi.org/10.1186/s40594-021-00323-x

[11] H. Athaya, R. D. A. Nadir, D. I. Sensuse, Kautsarina, and R. R. Suryono, "Moodle implementation for e-learning: A systematic review," in *Proc. SIET'21*, Malang, Indonesia, Sep. 13–14, 2021. doi: 10.1145/1234567890.

[12] A. Ali, "Getting Start with Google Classroom," *LinkedIn*, Jan. 1, 2023. [Online]. Available: https://www.linkedin.com/pulse/getting-start-google-classroom-ahmed-ali

[13] B. Mulatsih, "Application of Google Classroom, Google Form and Quizizz in chemical learning during the COVID-19 pandemic," *Ideguru: Jurnal Karya Ilmiah Guru*, vol. 5, no. 1, pp. 16–, Jun. 2020. [Online]. Available: https://doi.org/10.51169/ideguru.v5i1.129

[14] I. K. Sudarsana, I. B. M. A. Putra, I. N. T. Astawa, and I. W. L. Yogantara, "The use of Google classroom in the learning process," *J. Phys.: Conf. Ser.*, vol. 1175, no. 1, p. 012165, 2019, doi: 10.1088/1742-6596/1175/1/012165.

[15] S. Iftakhar, "Google Classroom: What Works and How?," *Journal of Education and Social Sciences*, vol. 3, Feb. 2016, pp. 12–14, ISSN 2289-9855.

[16] Frappe LMS, "Frappe LMS Review, Pros and Cons (Jun 2024)," *Itqlick.com*, Jun. 03, 2024. https://www.itqlick.com/frappe-lms#cons (accessed Sep. 11, 2024).

APPENDIX

A.1 Questionnaire Sample

1. Are you familiar with Learning Management Systems (LMS) like Moodle, Google Classroom, or Frappe LMS?

A. Yes

B. No

C. Heard of them, but don't really understand them.

2. Have you ever created or taken quizzes on an online Learning Management System?

A. Yes

B. No

C. Maybe

3. Do you know what an advanced quiz and marking system is in the context of online learning?

A. Yes

B. No

C. Heard of it, but don't really understand it.

4. Have you experienced difficulties with existing online quiz systems as a teacher or student?

A. Yes

B. No

C. Maybe

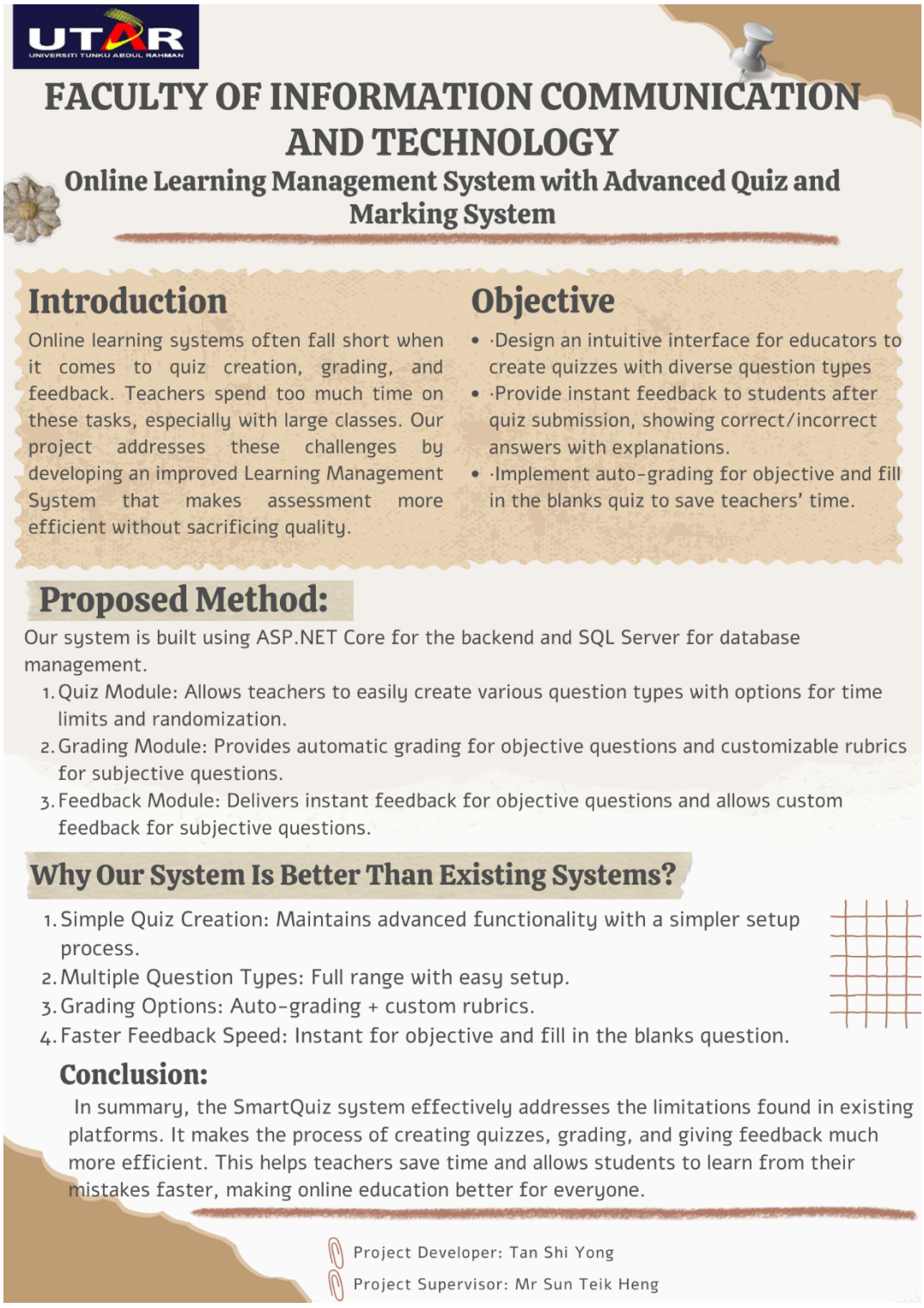
5. Do you think an LMS with advanced quiz features would improve the learning experience?

A. Yes

B. No

C. Maybe

POSTER


UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
FACULTY OF INFORMATION COMMUNICATION AND TECHNOLOGY
Online Learning Management System with Advanced Quiz and Marking System
Introduction
Online learning systems often fall short when it comes to quiz creation, grading, and feedback. Teachers spend too much time on these tasks, especially with large classes. Our project addresses these challenges by developing an improved Learning Management System that makes assessment more efficient without sacrificing quality.
Objective
·Design an intuitive interface for educators to create quizzes with diverse question types
·Provide instant feedback to students after quiz submission, showing correct/incorrect answers with explanations.
·Implement auto-grading for objective and fill in the blanks quiz to save teachers' time.
Proposed Method:
Our system is built using ASP.NET Core for the backend and SQL Server for database management.
1. Quiz Module: Allows teachers to easily create various question types with options for time limits and randomization.
2. Grading Module: Provides automatic grading for objective questions and customizable rubrics for subjective questions.
3. Feedback Module: Delivers instant feedback for objective questions and allows custom feedback for subjective questions.
Why Our System Is Better Than Existing Systems?
1. Simple Quiz Creation: Maintains advanced functionality with a simpler setup process.
2. Multiple Question Types: Full range with easy setup.
3. Grading Options: Auto-grading + custom rubrics.
4. Faster Feedback Speed: Instant for objective and fill in the blanks question.
Conclusion:
In summary, the SmartQuiz system effectively addresses the limitations found in existing platforms. It makes the process of creating quizzes, grading, and giving feedback much more efficient. This helps teachers save time and allows students to learn from their mistakes faster, making online education better for everyone.
Project Developer: Tan Shi Yong
Project Supervisor: Mr Sun Teik Heng