



Daffodil
International
University

Title of the Project

Quick Edu Live: An AI-Enhanced Educational Platform for Interactive
Classrooms

Course: Project – Spring'2024

Course Code: CIS499

Submitted By

Md. Minhajur Rahman

ID: 193-16-477

Department of Computing & Information System
Daffodil International University

Supervised By

Md. Mehedi Hassan

Lecturer

Department of Computing & Information System
Daffodil International University

Submission Date: 23 September, 2024

APPROVAL

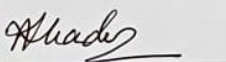
This Project titled “Quick Edu Live: An AI-Enhanced Educational Platform for Interactive Classrooms”, Submitted by Md. Minhajur Rahman, ID No: 193-16-477 to the Department of Computing & Information System, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computing & Information System and approved as to its style and contents. The presentation has been held on 23-09-2024.

BOARD OF EXAMINERS



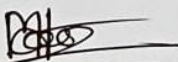
Md Sarwar Hossain Mollah
Associate Professor and Head
Department of Computing & Information System
Faculty of Science & Information Technology
Daffodil International University

Chairman



Md. Nasimul Kader
Assistant Professor
Department of Computing & Information System
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Md. Mehedi Hassan
Lecturer
Department of Computing & Information System
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Saifuddin Md. Tareeq
Professor
Department of Computer Science and Engineering
University of Dhaka, Dhaka

External Examiner

Declaration

I hereby declare that; this project has been done by me under supervision of **Md. Mehedi Hassan, Lecturer**, department of Computing and Information System (CIS) of Daffodil International University. I am also declaring that this project or any part of there has never been submitted anywhere else for the award of any educational degree like, B.Sc., M.Sc., Diploma or other qualifications.

Supervised By


23.09.2024

Md. Mehedi Hassan

Lecturer

Department of CIS

Daffodil International University

Submitted By

Md. Minhajur Rahman

Name: Md. Minhajur Rahman

ID: 193-16-477

Department of CIS

Daffodil International University

Acknowledgement

I wish to express my deepest thanks to all who helped me and gave advice during creating Quick Edu Live. There are a number of people without whose invaluable help and inspiration this project would not have been possible. First and far, I would like to thank my great supervisor, Md Mehedi Hasan, for his immense support, encouragement and direction. His invaluable advice, positive criticism and deep knowledge were more than helpful during the realization of this project. He was always very patient, accurate and aimed high standards all this encouraged me very much and helped raise the game in every respect. Finally, I must express my gratitude to my family and friends for their constant and understanding support during the course of the project. Their faith in my capabilities turned to be great motivation and power.

Abstract

Quick Edu Live is a groundbreaking educational technology designed to both simplify and enhance the teaching and learning processes. In doing so, I aim to solve the numerous existing challenges, such as the overload of administrative tasks associated with creating and grading quizzes and assignments for educators and a significant delay in feedback provision, as well as inefficient, time-consuming class management and insecure maintenance of student information. Quick Edu Live addresses these by delegating all administrative tasks to the intelligent platform, providing timely detailed automated feedback for students and ensuring the safe storage of data. I utilize AI Gemini-1.5-Flash API and Tesseract.js AI for creating questionnaires and assignments and for automatic grading with AI Paper Checker. These devices provide a big volume of consistent, relevant assignments and instant feedback which greatly increases learning efficiency. I provide teachers with easy and quick class and user management due to our learning and teaching management system, while students use classroom code to participate in the course. Node JS, React JS, MongoDB and Firebase constitute a light, fast modern tech stack that makes our platform responsive and compatible with different browsers. Moreover, I provide comprehensive training for teachers in the newly introduced technologies, roll out the platform for all users simultaneously and created measures for its timely scaling. Finally, being developed via the agile approach, Quick Edu Live is a result of our user feedback, which makes it a perfect utility for all users.

Table of Contents

Acknowledgement	iii
Abstract	iv
List of Figures	ix
List of Tables	xi
Chapter 1: Introduction	1
Chapter 2: Initial Study	2
2.1 Project Proposal	2
2.2 Background of the Project	3
2.3 Problem Area	3
2.4 Possible Solution.....	4
Chapter 3: Literature Review	6
3.1 Discussion on Problem Domain Based on Published Articles	6
3.2 Discussion on Problem Solutions Based on Published Articles	6
3.3 Comparison of Leading Solutions	7
3.4 Recommended Approach.....	8
Chapter 4: Methodology (DSDM)	9
4.1 What to Use.....	9
4.2 Why to Use	10
4.3 Sections of Methodology	10
4.4 Implementation Plans.....	12
Chapter 5: Planning.....	14
5.1 Project Plan	14
Work Breakdown Structure (WBS).....	14
Resource Allocation.....	15
Time Boxing	16
Gantt Chart.....	17

5.2 Risk Management	18
Risk Identification.....	18
Risk Assessment	18
Risk Action Plan	19
Steps Taken for Possible Risks	20
5.3 Change Management	20
Factors That Might Cause Change.....	20
DSDM Atern Welcomes Change.....	20
Change Workshop.....	21
Changes That Are Allowed.....	21
Key Decision Takers of Change	21
5.4 Quality Management.....	22
Rules Applied to Maintain Quality	22
DSDM Atern Standard Quality Measures	22
Quality Plan and Measuring Meter	22
Chapter 6: Feasibility	23
6.1 All Possible Types of Feasibility	23
6.2 DSDM – Good or Not for Quick Edu Live (PAQ).....	24
Chapter 7: Foundation.....	25
7.1 Problem Area Identification.....	25
i. Interview.....	25
ii. Observation.....	25
iii. Questionnaires	25
7.2 Specific Problem Area Identification and Description	25
7.3 Possible Solution.....	26
7.4 Overall Requirement List.....	27
7.5 Technology to Be Implemented.....	28

7.6 Recommendations and Justifications	28
Chapter 8: Exploration.....	29
8.1 Old System Use Case.....	29
8.2 Activity Diagram	30
8.3 Full System Use Case	31
8.4 Full System Activity Diagram	32
8.5 Prioritized Requirements List (PRL) MoSCoW prioritization technique:	33
Chapter 9: Engineering	34
9.1 New System Modules	34
9.2 Use Case.....	35
9.3 Class Diagram.....	36
9.4 Entity Relationship Diagram.....	37
9.5 Sequence Diagram	38
9.6 System Interface Design	39
Chapter 10: Deployment.....	50
10.1 Core Module Coding Samples	50
10.2 Possible Problem Breakdown	61
10.3 Prioritization While Developing.....	62
Chapter 11: Testing.....	63
11.1 Test Plan Acceptance.....	63
11.2 Test Case.....	63
11.3 Unit Testing	65
11.4 Module Testing	66
11.5 Integration Testing	67
11.6 Acceptance Testing.....	68
Chapter 12: Implementation	77
12.1 Training.....	77

12.2 Implementation Scheme.....	77
Big Bang	77
12.3 Scaling.....	77
12.4 Load Balancing	77
Chapter 13: Critical Appraisal and Evaluation	78
13.1 Objective that could be met	78
13.2 Objectives totally not touched in Quick Edu Live!	79
Chapter 14: Lessons Learned.....	80
14.1 Pre-Project Review – Closing.....	80
14.2 What Have I Learned?	80
14.3 What Problems I Have Faced	80
14.4 What Solutions Occurred.....	81
Chapter 15: Conclusion.....	82
15.1 Summary of the Project	82
15.2 Goal of the Project	82
15.3 Success of the Project	82
15.4 What I Have Done in the Documentation.....	83
15.5 Value of the Project	84
15.6 My Experience	84
Bibliography	85
Plagiarism Report.....	86

List of Figures

Figure 5.1: Gantt Chart	17
Figure 8.1: Use Case Diagram of Old System Classroom Managing.....	29
Figure 8.2: Activity Diagram Managing Classroom in the Old System.....	30
Figure 8.3: Use Case of Quick Edu Live	31
Figure 8.4: QEL Teachers Full System Activity Diagram	32
Figure 8.5: QEL Student Full System Activity Diagram.....	32
Figure 9.1: Use Case of Quick Edu Live	35
Figure 9.2: Quick Edu Live Class Diagram.....	36
Figure 9.3: Quick Edu Live Entity Relationship Diagram	37
Figure 9.4: Quick Edu Live Sequence Diagram	38
Figure 9.5: Home Page	39
Figure 9.6: Sign Up Page.....	40
Figure 9.7: Login Page.....	40
Figure 9.8: Teachers Home Page	41
Figure 9.9: Create Class modal.....	41
Figure 9.10: Teachers Home Page with a Class	41
Figure 9.11: Ai Classroom “Announcement” Page	42
Figure 9.12: Ai Classroom “Classwork (Quiz)” Page	42
Figure 9.13: Quiz Modal.....	42
Figure 9.14: Ai Classroom “Classwork (Quiz)” Page with created quiz.....	43
Figure 9.15: View Quiz Question Page	43
Figure 9.16: Print Quiz	43
Figure 9.17: Ai Classroom “Classwork (Assignment)” Page.....	43
Figure 9.18: Assignment Modal	43
Figure 9.19: Ai Classroom “Classwork (Assignment)” Page with created assignment	44
Figure 9.20: View Assignment Question Page	44
Figure 9.21: Print Assignment	44
Figure 9.22. View Classroom People	44
Figure 9.23: Buying Ai Paper Checker.....	44
Figure 9.24: SSLCOMMERZ Demo Payment Page	45
Figure 9.25: Payment History Page	45
Figure 9.26: Ai Paper Checker.....	45
Figure 9.27: Ai Paper Checker Result	45

Figure 9.28: Students Home Page.....	46
Figure 9.29: Enroll Class Modal.....	46
Figure 9.30: Students Home Page with a Enroll Class	46
Figure 9.31: Ai Classroom “Classwork (Undergoing Quiz)” Page	46
Figure 9.32: Performing Quiz.....	47
Figure 9.33: Completed Quiz.....	47
Figure 9.34: View Quiz Marks	47
Figure 9.35: Ai Classroom “Classwork (Assignment)” Page.....	47
Figure 9.36: Ai Classroom “Classwork (Undergoing Assignment)” Page.....	48
Figure 9.37: Assignment Submission	48
Figure 9.38: Assignment Submitting	48
Figure 9.39: Ai Classroom “Classwork (Assignment)” Submitted Page	48
Figure 9.40: Help Page	49
Figure 9.41: Feedback Page.....	49
Figure 9.42: Report a Problem Page	49
Figure 10.1: Signup Code Sample	53
Figure 10.2: Create Classroom Code Sample	57
Figure 10.3: Ai Quiz Generator Code Sample.....	61
Figure 11.1: Unit Test 1	65
Figure 11.2: Unit Test 2.....	65
Figure 11.3: Module Test 1.....	66
Figure 11.4: Module Test 2.....	66
Figure 11.5: Integration Test 1.....	67
Figure 11.6: Integration Test 2.....	67
Figure 11.7: Acceptance Test	68
Figure 11.8: Registered a new teacher and student.....	69
Figure 11.9: Logging as a teacher and creating a new class.....	70
Figure 11.10: Copying classroom code to share with students.....	70
Figure 11.11: Logging as a student and enrolling in the class using the class code.....	71
Figure 11.12: Creating a quiz using the AI Quiz Generator as a teacher.	72
Figure 11.13: Logging as a student and participating in the quiz.....	73
Figure 11.14: Verifying that the quiz is automatically graded and feedback is provided	74
Figure 11.15: Posting an announcement as a teacher and ensure as a student can interact.....	75
Figure 11.16: Removing a student from the class as a teacher.....	76

List of Tables

Table 5.1: Work Breakdown Structure	14
Table 5.2: Resource Allocation.....	15
Table 5.3: Time Boxing	16
Table 8.1: Prioritized Requirements List	33
Table 9.1: Module of the System.....	34
Table 12.1: Quick Edu Live User Training	77

Chapter 1: Introduction

Quick Edu Live is an education platform built on artificial technology that is used to determine the efficiency of teaching and learning as well as creating a dynamic classroom and interactive session for users. The system is designed to accommodate two users: Teachers and Students. These user types are assigned to give certain capabilities and roles that affect how education is seen and increase the efficiency of its processes using the platform. Teachers on Quick Edu Live can create an amazing class and manage a class. They have the ability to create a new class, delete class and get unique class code, which is given to students to enroll in their classes. This enables only the students with correct class code to enter the teacher's class. Instructors can also display text properly as an announcement. The system has designed two specific tools/classrooms for proper activity in the class.

The first one is AI quiz generator, a tool that can automatically create a quiz, where it picks a predefined set of options to create a quiz like subject, date, time, quiz number, duration, total mark, total question, difficulty and topic. The quiz uses the Gemini-1.5-Flash API to improve its efficiency. It also has an auto grading system, in which immediately after submission of the quiz, there have a quiz print out option for better offline class.

The second is AI assignment generator made by the same API as the first one for quick assignment. Note they are not yet auto graded, not only that, but the AI Paper Checker is not free for use as it cost \$10.99 or 500 BDT through SSLZCommerce for just the teachers only. It also converts student question and test script into an image, based on the test script, it sends the possible right answer. Teachers in Quick Edu Live can also grade the students.

The system is populated by students who enter classes using the code given to them by their teachers. Once a student is enrolled, they will be able to see an announcement, comment on them, quiz them on the assignment. It is important to take into account the students are aware of the submission of the quiz as seen from the instructor. Both parties can access Quick Edu Live website for update. They can also report any issues or give general feedback, but this will depend up the event. It can only login via email, password and Google by firebase. It has a Responsive Technology for every device with browser. The technology used on the website are JavaScript, React JS, Node JS and MongoDB. Only SSLZCommerce that can use the API features.

Chapter 2: Initial Study

2.1 Project Proposal

The main concept of the Project idea is to change the way people learn. While the whole world makes advances in science and technology's improvements, it is quite surprising that the educational system has not shown significant changes for more than 50 years. The project aims to implement cutting-edge technologies that will revolutionize the perception of the current system. The Quick Edu Live project aims to bring new Artificial Intelligence practices into everyday classroom life. It is evident that AI technologies can significantly increase the effectiveness of student learning. However, it is time to bring them to another level. It is planned to develop an educational platform based on modern AI technologies, suggesting the constant interaction between the teacher and students. First and foremost, the idea is to automate all the ongoing processes in education. Thus, effective learning implies not only classes in schools or universities. It is also necessary to take time preparing assignments, generating and checking quizzes and assignments and eventually, tracking students' grades and performance. However, the usage of AI tools might free the teacher from the continuous routine tasks and reprehend them for something more interesting and useful for students: contact, developing more unique courses, talking to them.

The main features will include standard AI Classroom, AI Quiz Generator, AI Assignment Generator. Obviously, every aspect of the learning process should be improved. The latter is designed to check papers upon different criteria and generate grades for each student based on the paper. Moreover, the project will be based on a reliable Firebase Authentication system. The platform will also be developed using React JS due to a broad variety of features and capabilities. Since the platform is intended to be used by all students and teachers, it will be created as responsive, equally nice on tablets and smartphones. In general, the platform's stack will include JavaScript, React JS, Node JS and MongoDB.

In conclusion, to make significant changes in the current system and help to borrow distance between the surge of technologies' progress. In addition, to increase the students' interest level, constant feedback is necessary and the only way is to use a responsive AI tool that will automate all the mentioned routine processes. Such usage will soon make the whole educational system more developed and the relationship between the teacher and students will change, making the latter more effective in their studying processes.

2.2 Background of the Project

Nowadays, the educational sector is changing significantly due to technology. In addition, the rapid shift towards online learning has highlighted how poorly set up the traditional system was. Under normal circumstances, the usual classroom settings forced the vocational and higher school teachers to do a lot of administrative work, such as creating quizzes, assignments and grading them. Moreover, the feedback usually comes late, which influences the learning process quite notably. During the COVID-19 pandemic, the situation with remote learning required from many schools and universities to come up with practical tools for online education, which created a new demand and demographic. The fact that this market expanded so quickly shows that there were vital features missing in the existing tools. One of the evident issues was the lack of interaction. The other problem that had been already existing in traditional settings was that teachers had to perform significant amounts of non-teaching work. In response to this, the idea for the Quick Edu Live platform emerged. The primary goal of the project is to create a highly efficient online education platform that would help teachers to spend more time teaching and connecting with their students by employing AI technologies. The AI Quiz Generator, AI Assignment Generator and AI Paper Checker together reduce weekly tasks of each teacher to twenty minutes. In addition to outstanding efficiency, the product provides an absolute positive user experience. It has responsive design so that it operates seamlessly everywhere. The platform has secure authentication to enable the users to sign in safely on any device, no matter their technical literacy level. From a more technical side, the selection of new technologies such as JavaScript, React JS, Node JS and MongoDB guarantees that the final product is reliable and most importantly, easy to scale when a growing number of users appear. The main concern is to provide instant feedback to every student. The current-methods do not ensure high levels of motivation and there is a risk that from now on, even when the feedback finally becomes available, the students will not accept it positively, analyzing their scores from the point of view of their previous experience. However, when one's accomplishments are given the evaluation not long after the completion of a particular task, people tend to be much more positive and might use this information to continue improving and do better next time.

2.3 Problem Area

Several hurdles to effective teaching and learning processes exist in the current educational landscape. An administrative burden on teachers is one of the first and foremost issues. In traditional educational systems, teachers consume a good portion of their time creating quizzes,

assignments, grading exams and maintaining data records. However, they are also tasked with some administrative duties which draws focus away from their primary function as educators and engagers of students. A second very serious concern: The delayed feedback back to students. Typically, these high-stakes assessments are completed under conventional systems in which it can take days or even weeks for students to receive any feedback. This delay has an effect on the learning process, as feedback is needed so you understand what mistakes you are making and can get better sooner. It can also reduce student motivation and engagement with the subject, as quick feedback leads to greater interest in a topic.

Initiated by the COVID-19 pandemic, moving to online transfer has simply compounded this problem. The majority of current online educational platforms only have passive and rigid structure, are not well interconnected to facilitate teachers-students interaction. The rapid change to remote learning has revealed the limitations of existing systems in delivering a frictionless and fun educational experience. Teachers have trouble providing online test content and more importantly with the management of these tests once completed; in addition to seeing testing scores is students' primary incentive for reading an article.

What is more, the current systems often feature weak authentication and security protocols opening them up to privacy leaks even if meant for secured use. This market has never seen a greater urgency for their product - many educators have an immediate requirement to transition online and the move is being driven by necessity in most cases.

Our objective is to solve these issues by employing sophisticated AI technologies, so I automate as many basic activities as possible and deliver instant feedback via Quick Edu Live. The platform is built to help reduce the admin load on teachers, boost student interactivity and drive better learning outcomes. The use of AI in Quick Edu Live is a complete package for meeting the existing barriers in the academic sector. It does this by engaging students with the learning process and giving them feedback, but most importantly, it is instantaneous.

2.4 Possible Solution

Quick Edu Live is a revolutionary tech platform that focuses on problematic aspects in the educational systems of nowadays. Powered by AI, it provides an unprecedented mixture of functionalities that simplify the mechanisms for educators and lighten their administrative loads as well as increase student engagement levels with improved efficacy in outcomes. Here are some of the unique solutions Quick Edu Live introduces to solve:

AI Quiz Generator: AI is rapidly transforming the way we create quizzes. Generates questions for a quiz automatically based on topic of the class and fixed parameters like Quiz Number, Date-Time, Duration (in minutes), Total Marks & Difficulty Level. The AI Quiz Generator gemini-1.5-Flash API It not only ensures consistency and accuracy but also has automatic grading function. Teachers receive real-time feedback with this approach which also helps in saving a lot of valuable time for students as well.

AI Assignment Generator: Teachers can create assignments with instructions and the tool will ensure they stay on topic for a given class subject or difficulty level. Though assignments have to be graded, the assignment creation is heavily abbreviated with this AI Assignment Generator.

Artificial Paper Checker: This is a premium feature for teachers where the images of question-and-answer script will be converted into text using tesseract.js. The following AI Paper Checker takes your writing and compares the words with The Gemini-1. 5-Flash API to get the answer given a question. The tool grades the paper and gives responsive feedback to students, therefore cutting down on time educators spend with administration.

User-friendly and Secure Platform: Quick Edu Live provides safe authorization by Firebase, supporting email verification (via Google Sign In), password change option. That makes it more secure, reliable and scalable using modern technologies such as JavaScript ReactJs Nodejs MongoDB to make the application responsive across devices and browsers available globally.

The Quick Edu Live has a great turnaround time because it is not just an online platform but also the way to join an active learning community which results in increased student participation. Automatic grading with immediate results keeps students engaged and motivated by providing timely feedback. The platform also allows students to see announcements, provide comments on them and interact with friends fostering a collaborative learning environment as well.

All in one Administration Tools: Quick Edu Live provides instructors with multiple tools that enable them to manage classes, monitor student enrollments and post announcements. Such tools are used to automate administrative work, thus facilitating faculty members in giving an increased focus on offering quality teaching experience and dealing with students.

Chapter 3: Literature Review

3.1 Discussion on Problem Domain Based on Published Articles

Since few decades, many researchers have studied about technology integration in education. This rapid evolution of AI affords instructors with new and dynamic ways to improve the learning process. While with the current educational landscape, several challenges continue to exist and are well documented and researched in academic literature.

First, a cumbersome problem is placed upon the shoulders of teachers. Teachers reportedly use a substantial amount of their time doing non-teaching activities such as preparing quizzes, lessons and examinations (Skaalvik & Skaalvik, 2018) [1]. This can ultimately cause teacher burnout and reduce job satisfaction, both of which impact the way students are taught. As (Darling-Hammond, 2010) [2] reminded us, lightening teachers' administrative loads can mean more effective teachers and better learning for students.

This points out that feedback is an important factor. Feedback at the right time is important to learning for two reasons: It tells students if they are on track and shows them what failed (Hattie & Timperley, 2007) [3]. But in legacy educational systems, feedback often has a delay of days or even weeks which undermines learning and reduces student engagement. Corrective feedback supporting the idea of immediate informational entailment has been recommended in several pieces of literature dating back to before works on automatization began (Shute, 2008) [4].

This already frail line is being broken even more with the advent of online learning in view to COVID-19. Most schools and universities were not ready to continue teaching with online educational tools, so it was a difficult period. Study also stated that the sudden shift to online learning was found to increase stress and anxiety among teachers as well as students (Besser, Zeigler-Hill & Flett, 2020) [5].

3.2 Discussion on Problem Solutions Based on Published Articles

The use of AI and other advanced technologies has also been proposed to provide some solutions with respect to the challenges in this arena. The literature reveals ideas and thoughts as to how these solutions may contribute toward an even more efficient higher educational processes. One potential remedy is the power of AI to automate administrative tasks. AI-enabled tools can cut down the time it takes for teachers to create quizzes, assignments and grading significantly.

For example, AI models could create questions for a quiz based on specified parameters or grade quizzes without human intervention (Chen, Xie & Hwang, 2018) [6]. This automation saves time and also makes the assessment process more consistent as well as accurate.

A second strategy is introducing real-time feedback mechanisms. With AI and machine learning algorithm, immediate feedbacks can be given to the students helping them understand their errors thereby improving a lot more. Studies show that timely feedback is associated with improved student learning performance and participation (Wang, 2019) [7]. Intelligent tutoring systems leveraging technologies like natural language processing (NLP) and computer vision are rapidly emerging to provide personalized feedback tailored for the individual students (Nye, 2015) [8]. The move to online learning has also driven the creation of interactivity and edutainment oriented educational tools.

3.3 Comparison of Leading Solutions

AI Quiz Generators

Best Features: Currently lacks automated grading, requiring manual evaluation by teachers. May face challenges in generating assignments for diverse subjects.

Limitations: Question generation effectiveness heavily relies on the AI accuracy That dependence massively reduces versatility.

AI Assignment Generators

Best Features: Automates the process to streamline address assignment creation.

Limitations: Missing Automated Grading - Teachers have to check answers manually. There may be difficulties in producing tasks for different topics.

AI Paper Checkers

Best Features: Grades Automatically grades images of answer-scrapes to text and creates the right response. Saves teachers of critical hours they might have spent marking papers.

Limitations: Some of the handwriting was hard to read so I had poor results in image-to-text conversion. Only for question types that can be autoscored

Interactive Online Learning Tools

Best Features: Enables classroom and announcements.

Limitations: Depends on stable internet and tech infrastructure Learning curve: May struggle to keep thousands of online students engaged.

3.4 Recommended Approach

Based on the literature review, the recommended approach for Quick Edu Live is to integrate a combination of AI-powered tools and interactive online learning features.:

AI-Powered Tools for Administrative Automation

Use AI Quiz Generators and AI Assignment generators for automatic assessment-generation. These tools help save time that is used in an administrative way by teachers and at the same time maintain consistency, relevance of quizzes as well maintaining originality while giving assignments.

Real-Time Feedback Mechanisms

Incorporate automatic grading features to provide instant feedback to students. This will enhance the learning process by helping students understand their mistakes and improve their performance in real time.

Interactive and Engaging Online Learning Tools

Develop virtual classrooms and discussion forums to support interaction between teachers and students. Adaptive learning platforms can be used to personalize the learning experience based on individual student needs.

Chapter 4: Methodology (DSDM)

4.1 What to Use

The Dynamic Systems Development Method (DSDM) will be used for Quick Edu Live development. DSDM is an agile project delivery framework that emphasizes on iterative development, user feedback and delivering working modules incrementally. Created to be scalable and flexible, it is well suited for very complex projects that need frequent changes in development or new features according to the user length.

Core Components of DSDM

Feasibility and Business Study: Concept validation and business discovery phase.

Functional Model Iteration: Prototype development to test user requirements and system functionality.

Design and Build Iteration: Build the system components incrementally based on feedback of users.

Implementation: Final Product Deployment and transitioning the product to User Domain.

Key Principles of DSDM

Active User Involvement: Frequently involving users in the development process to make sure that their needs and feedback are implemented.

Empowered Teams: Teams have the authority to decide working, open and responding development environment.

Frequent Delivery of Products: Early Delivery of Valuable Functional Components with Frequent Updates to Improve the end product.

Fitness for Business Purpose: Verifying the business requirements are satisfied and realize value from what developed.

Iterative and Incremental Development: Developing the system in bite-sized parts for tweaking and refining on a regular basis

4.2 Why to Use

Quick Edu Live chose DSDM for its additional flexibility, user-centered approach and iterative nature. All of these align well with project goals and the rapidly changing nature of educational technology development.

Advantages of Using DSDM

User-Centric Development: Active user participation provides a good understanding of what the users are expecting from the system and facilitates better acceptance rates.

Risk Management: This enables the team to recognize and counter were links at an early stage of development by splitting projects into smaller pieces.

Enhanced Collaboration: The cross-functional nature of DSDM encourages the collaboration between developers as users ensuring all perspectives are accommodated for.

Quality Assurance: Testing and validation should be done at intervals throughout the development process to keep quality high and avoid big surprises in a final product.

Application to Quick Edu Live

User Involvement: Throughout the production process, teachers and students also will be integrated into development through testing of prototypes.

Iterative Development: The platform will launch over iterations, with each iteration delivering capabilities such as the AI Quiz Generator and AI Assignment Generator.

Frequent Delivery: It enables us to deliver working parts of the system from your core functionality, each part adding more functions and features as higher levels in terms scope.

Risk Mitigation: Detection of possible problems like scalability, security threats etc. will occur at the beginning its iterations rather than on later stages and that serves mostly as risk mitigation practice.

4.3 Sections of Methodology

The DSDM methodology will be divided into several key sections to ensure a structured and comprehensive approach to the development of Quick Edu Live.

Feasibility Study

Objective: Evaluate if the idea is feasible, determine risks and create a business case.

Activities: Market analysis, stakeholder interviews, risk assessment & cost benefit analysis.

Deliverables: Business case, list of project risks, revised initial Project Plan.

Business Study

Objective: Get to know the business objectives and requirements, define project scope.

Activities: Stakeholder engagement workshops, defining requirements and mapping out processes.

Deliverables: Business Requirement Document, Project Scope Statement.

Functional Model Iteration

Objective: Developed prototypes iteratively in response to feedback from users who used system features.

Activities: Prototyping, user testing and feedback

Deliverables: Prototypes, use case feedbacks and updated requirements.

Design and Build Iteration

Objective: Systems components are iteratively designed and developed to increase in functionality based on engineering prototypes with user feedback.

Activities: System design, code, unit test case, integration testing.

Deliverables: Components, Design Docs and Test Reports.

Implementation

Objective: launch the last product begin to introduce in user environment take care of any handoff that need to be done.

Activities: Go-live, training the users and setting support.

Deliverables: APIs are imported already in production, training collateral and support information.

Post-Implementation Review

Objective: Analyze the success of your project and determine any lessons learned for future projects.

Activities: Feedback gathering from the users, analysis based upon performance reviews and project retrospective.

Deliverables: Lessons Learned Document, Post Implementation Review Report.

4.4 Implementation Plans

The implementation of Quick Edu Live will follow a structured plan based on the DSDM methodology. The plan will be divided into several phases, each corresponding to the sections of the methodology.

Phase 1: Feasibility Study

Timeline: 2 weeks

Activities: Market research for an AI-based education platform and look how this idea works or the demand is! Conduct interviews with the major stakeholders i.e., teachers, students and school administrators to understand what they need and also want Conduct a risk assessment to recognize the probable pitfalls and prepare remediation measures.

Deliverables: A feasibility report that discusses the viability of a project, risks identified and how to mitigate them. A more detailed overview of what is being undertaken, to include scope and timeline.

Phase 2: Business Study

Timeline: 3 weeks

Activities: Facilitate workshops with both business and technology stakeholder across the project timeline to understand specific details on their requirements. Document current educational workflows and determine where I can optimize. Business Requirement Document (BRD)Project Scope Statement.

Deliverables: Business requirements documentation of the desired features for Quick Edu Live. A detailed description of the project scope statement including what is and is not a part if the work to be delivered.

Phase 3: Functional Model Iteration

Timeline: 4 weeks

Activities: Develop initial prototypes of key components such as the AI Quiz Generator and AI Assignment Generator. Conduct user testing sessions to gather feedback on the prototypes. Refine the prototypes based on user feedback and update the requirements as necessary.

Deliverables: Functional prototypes demonstrating the core features of Quick Edu Live. User feedback reports summarizing the insights gathered from testing sessions. Refined requirements document incorporating user feedback.

Phase 4: Design and Build Iteration

Timeline: 8 weeks

Phase 5: Implementation

Timeline: 2 weeks

Activities: Develop design system components off of those finalized requirements and prototypes. Build components using React JS, Node JS and MongoDB technologies. Test each component to see if it is working as intended and can be combined without breaking.

Deliverables: Created system components AI Quiz Generator, Assignment, Paper Checker. Architecture, design diagram of the system Reports of testing done on the units and during implementations.

Activities: Deploy the final version of Quick Edu Live to the production environment. Conduct training sessions for teachers and students to ensure they understand how to use the platform. Set up support channels to assist users with any issues they may encounter.

Deliverables: Deployed system accessible to all users. Training materials, including user guides and video tutorials. Support documentation detailing common issues and troubleshooting steps.

Phase 6: Post-Implementation Review

Timeline: 1 week

Activities: Quick Edu Live Feedback from Users So that I can perform a performance monitoring and improvement exercise.

Deliverables: Post-Implementation Review report on user feedback and system performance.

Chapter 5: Planning

5.1 Project Plan

Work Breakdown Structure (WBS)

Work Plan for Quick Edu Live shall be a comprehensive work breakdown structure (WBS), so that all elements and deliveries are categorized, current and properly monitored.

Task Name	Duration (days)	Start Date	End Date
Introduction	10	1 Jan 2024	10 Jan 2024
Initial Study	7	11 Jan 2024	17 Jan 2024
Literature Review	5	18 Jan 2024	22 Jan 2024
Methodology	6	23 Jan 2024	28 Jan 2024
Project Plan	10	29 Jan 2024	7 Feb 2024
Feasibility Study	9	8 Feb 2024	16 Feb 2024
Foundation	9	17 Feb 2024	25 Feb 2024
Exploration	11	26 Feb 2024	7 Mar 2024
Engineering	14	8 Mar 2024	21 Mar 2024
UI/UX Development	7	22 Mar 2024	28 Mar 2024
Development	60	1 Apr 2024	30 May 2024
Testing	9	31 May 2024	8 Jun 2024
Implementation	7	9 Jun 2024	15 Jun 2024
Critical Appraisal & Evaluation	4	16 Jun 2024	19 Jun 2024
Lessons Learned	5	20 Jun 2024	24 Jun 2024
Conclusion	5	25 Jun 2024	29 Jun 2024
Total	178		

Table 5.1: Work Breakdown Structure

Resource Allocation

I need to allocate good resources for it in order to keep quick Edu Live running and on track with the objectives.

Task Name	Duration (days)	Resource
Introduction	10	Analyst, User
Initial Study	7	Analyst
Literature Review	5	Analyst, Team Leader
Methodology	6	Analyst, Developer, Project Manager
Project Plan	10	Analyst, Project Manager, Team Leader
Feasibility Study	9	Analyst, Project Manager, Team Leader, User
Foundation	9	Analyst, Team Leader
Exploration	11	Analyst, Developer, Team Leader, Designer
Engineering	14	Project Manager, Team Leader
UI/UX Development	7	Designer, Developer, Team Leader, User
Development	60	Developer, Analyst, Tester
Testing	9	Tester, Developer, Team Leader, User
Implementation	7	Project Manager, Developer, Tester
Critical Appraisal & Evaluation	4	Analyst, Developer, User
Lessons Learned	5	Developer, Analyst
Conclusion	5	Analyst
Total	178	

Table 5.2: Resource Allocation

Time Boxing

An important part of DSDM is that time boxing through which it ensures the contributions within each phase of a project are completed on schedule. I plan out the work in iterations, that each lasts a fixed amount of time.

Time Boxes	Task Name	Duration (days)	Resource
TB1	Introduction	10	Analyst, User
	Initial Study	7	Analyst
TB2	Literature Review	5	Analyst, Team Leader
	Methodology	6	Analyst, Developer, Project Manager
TB3	Project Plan	10	Analyst, Project Manager, Team Leader
	Feasibility Study	9	Analyst, Project Manager, Team Leader, User
	Foundation	9	Analyst, Team Leader
TB4	Exploration	11	Analyst, Developer, Team Leader, Designer
	Engineering	14	Project Manager, Team Leader
	UI/UX Development	7	Designer, Developer, Team Leader, User
TB5	Development	60	Developer, Analyst, Tester
TB6	Testing	9	Tester, Developer, Team Leader, User
	Implementation	7	Project Manager, Developer, Tester
TB7	Critical Appraisal & Evaluation	4	Analyst, Developer, User
	Lessons Learned	5	Developer, Analyst
	Conclusion	5	Analyst
Total		178	

Table 5.3: Time Boxing

Gantt Chart

A Gantt chart will be depicted here to show the project schedule visually. It would be a chart including all activities, along with their start and finish dates as well as dependencies on the part of responsible teams.

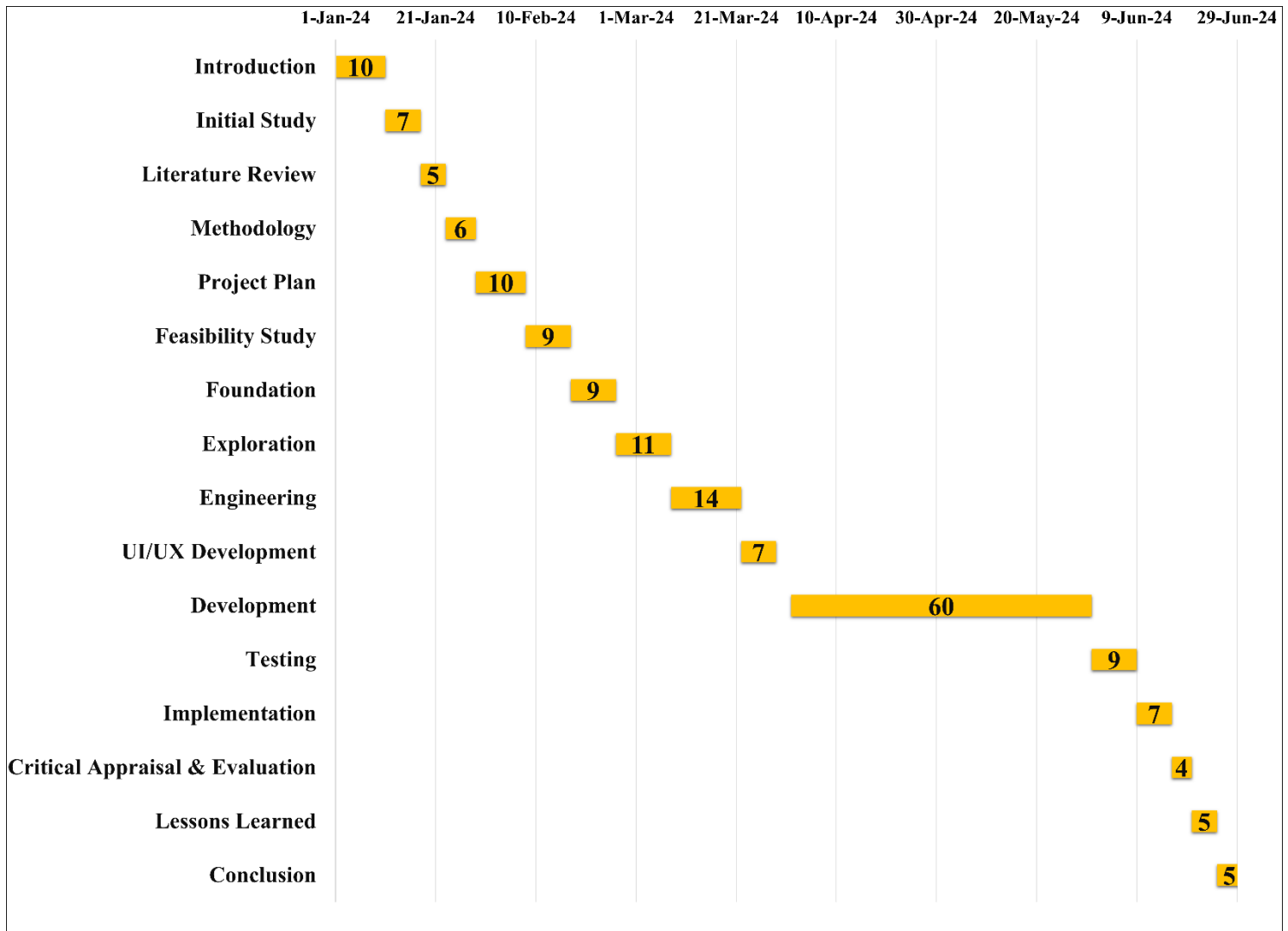


Figure 5.1: Gantt Chart

5.2 Risk Management

Risk Identification

It is very important to identify risks that present themselves in the path of successful implementation and operation of Quick Edu Live. Major risks identified Major risk for this project:

Technical Risks

- Problem with AI tools Integration (Gemini-1.5-Flash API and Tesseract.js).
- Lack of performance on high concurrent users.
- Exposure of sensitive data.

Operational Risks

- Resistance from teachers and students to change.
- Poor training and support resulting in misuse of the software
- Resource Constraints for Development and Maintenance.

Project Management Risks

- The classic scope creep that results in time and cost overruns.
- Mismatch of The Project Goals with Stakeholders Expectations
- Unforeseen technical problems which could lead to delays in the development process.

External Risks

- Regulatory requirements around data security and privacy continue to evolve;
- Competition of the market from another educational platform
- Depends on Third-party services like cloud hosting providers.

Risk Assessment

Once identified, these risks need to be assessed in terms of their likelihood and potential impact on the project. This involves both qualitative and quantitative analysis.

Likelihood Assessment:

- ❖ Technical Risks: Very likely because this system is so complex.
- ❖ Operational Risks: dependent on the effectiveness of user training programs.
- ❖ External Risks: Unlikely but high impact if they do happen.

- ❖ Security Risks: It is almost sure to happen due the rise in cyber security threats.
- ❖ Performance Risks: This is likely to happen if given enough room for the system to scale and load management techniques.

Impact Assessment:

- Technical Risks: High technical failures result in a stop to progress for the project.
- Operational Risks: Medium with possible increase using through user training programs.
- External Risks: Legal & regulatory changes in force, with high impact
- Security Risks: Severe reputation impact and loss of trust among clients as a result from data breaches.
- Performance Risks: Critical, user experience and system availability.

Risk Action Plan

In order to reduce these identified risks, a complete action plan is established:

Technical Risks

- The full integration of AI tools will be achievable after proper testing and prototyping.
- Improve the platform for performance and scalability plus More.
- Adopt security controls such as encryption and constantly monitor data.

Operational Risks

- Develop teachers training programs and student training program in totality.
- Keep up the support and resources as people transition to you through those platforms.
- Assign extra resources to stress development phases.

Project Management Risks

- Work with stakeholders to define project scope and objectives
- Employing practices like agile project management to control scope and order tasks
- Prepare buffer time and resources for unexpected tech problems.

External Risks

- Track regulatory changes and update the platform to adapt.
- Track the market trends and employ strategies to remain ahead.
- Create backup plans for 3rd party service outages.

Steps Taken for Possible Risks

Regular Risk Reviews

- Arrange Regular Risk Review Meetings to Track and Re-Evaluate the Risks.
- Regularly revising the risk management plan according to new data and project advancement.

Performance Monitoring

- Put monitor tools in place to measure performance and early detection of issues.
- Do performance testing and optimization regularly.

Training and Support

- Create and Distribute Training for All Users.
- Create a support group that can help users immediately and solve the problems.

5.3 Change Management

Factors That Might Cause Change

The Quick Edu Live project can be transformed for a number of reasons:

User Feedback: User feedback will also inform bug fixes and feature requests.

Technological Advancements: New features can be added by the latest tools and technology.

Regulatory Changes: Movements on data protection and privacy law could see the platform amended.

Market Dynamics: Competitive pressure and market trends require changes to remain competitive.

DSDM Atern Welcomes Change

DSDM Atern fundamentally accepts change as a healthy project development characteristic.

Key principles include:

Iterative Development: Iteratively construct the platform and check in with reviews of capabilities based on feedback and changing requirements.

Active User Involvement: Obtain feedback from users at every stage of the development to ensure you are on track.

Empowered Teams: Enable developers to take action and make changes fast, so they can keep the project moving.

Change Workshop

Perform Change Workshop to Evaluate & Plan for Changes:

Workshop Objectives

- Exam the effect of proposed changes on project
- Decide to make the most value for least effort changes.
- Create an implementation plan for approved changes.

Participants: Key stakeholders, project managers, development team and user representatives.

Outcomes: A ranked list of changes and detailed implementation plans with follow-up metrics.

Changes That Are Allowed

Only the ones that falls within scope of the project objectives providing significant business value are allowed. Examples include:

User-Requested Features: Deliver user stories for functionality that improves features and overall end-user experience.

Performance Enhancements: Enhance the performance and scalability of platform by feed-back from users, performance audits.

Regulatory Compliance: Modify platform to comply with the recently-held data protection and privacy regulations.

Key Decision Takers of Change

Those who are included in decision making process of changes:

Project Manager: Manages the change management process to ensure changes are aligned with project objectives.

Development Team: Supports technical analysis and conduct feasibility assessments for proposed changes.

User Representatives: Provide ideas regarding what the user wants and requires, serving in that way modifying satisfies user expectations.

5.4 Quality Management

Rules Applied to Maintain Quality

High-quality standards need to be maintained which is important for the success of Quick Edu

Live key rules include:

Adherence to Best Practices: Adhere to best practices for software development, including coding standards and documentation.

Continuous Testing: Install continuous testing right through development to identify issues as early as possible.

User-Centric Design: Our website is user-focused and usable to make a good experience.

Regular Reviews and Audits: Perform code reviews, security audits and performance tests on a regular basis to ensure quality.

DSDM Atern Standard Quality Measures

The following standard quality measures are adopted in the DSDM Atern methodology to deliver projects:

Fitness for Purpose: To meet user and stakeholder requirements

On-Time Delivery: Complete the project on time, so you will be able to deploy and start using them quicker

Right First Time: Commit our work from the start to produce higher quality outputs.

Quality Plan and Measuring Meter

This is the quality plan where I describe how will I keep and measure the level of our code:

Quality Objectives: Provide a secure, efficient and user-friendly platform.

Quality Assurance Activities: Perform regular testing, code reviews and performance checks to uphold the quality standards.

Quality Metrics: It's Define and monitor quality metrics for example, defects rate, user satisfaction scores, performance benchmarks & compliance with security standards.

Continuous Improvement: Feedback loops and iterative development create a continuous improvement of the quality of the platform.

Chapter 6: Feasibility

6.1 All Possible Types of Feasibility

Feasibility analysis is a critical component in determining whether Quick Edu Live can be successfully developed and implemented. Various types of feasibility will be assessed to provide a comprehensive evaluation:

Technical Feasibility

- Technical Feasibility covers questions on availability of technology, tools and resources in building Quick Edu Live The evaluation includes:
- Technology Stack: JavaScript, React JS, Node JS, MongoDB — Gemini-1 5-Flash API and Tesseract. The combination of server-side Node.js and client-side ReactJS is solid one that can work very well for a responsive and scalable system.
- Development Resources: I have a lot of code that requires AI, frontend and backend development patterns as well database management in turn developers skilled at this are important.
- Infrastructure: Scaling of servers in Firebase Hosting are fully reliable thus you have best performing, secure web server.

Economic Feasibility

- Economic Feasibility is to determine if the project can be justified in terms of each cost and benefits.
- Development Costs: Like payments for development resources, software licenses, hardware infrastructure costs etc.
- Operational Costs: As for after-deployment, operational costs will consist of servers hosting and keeping status quo as much as possible with potential overhead on maintaining resources.
- Revenue Model: It is an external revenue generation for Quick Edu Live as it comes under premium features and while using the AI Paper Checker, you will need to make a purchase.

Operational Feasibility

Operational feasibility checks whether Quick Edu Live can work within the existing organization aspects and if end users (teachers and students) are able to use it effectively.

- **User Acceptance:** For the success of this site, it needs to be accepted by teachers and students may need user-friendly interfaces, responsive design and intuitive functions.
- **Training and Support:** The effectiveness of platform use also depends on the level of user instructions, video tutorials and a quick support.
- **Scalability:** As more users and data are added, the platform needs to scale appropriately so that performance does not degrade.

Legal Feasibility

Ensuring the Quick Edu Live is compliant with all relevant laws and regulations.

- **Intellectual Property:** Make sure you are legally operating with third party APIs, using licensed and agreement technology.

Schedule Feasibility

- **Project Timeline:** It is a phased approach that breaks down the project into individual phases with associated deliverables and timelines.

6.2 DSDM – Good or Not for Quick Edu Live (PAQ)

- **Project Complexity**
 - **High Complexity:** The iterative approach of DSDM allows us to implement AI-based quiz generation and automated grading, with security based on federated authentication.
- **User Involvement**
 - **Critical User Involvement:** DSDM provides feedback from teachers and students throughout the project that continuously aligns with user needs.
- **Requirements Uncertainty**
 - **Evolving Requirements:** With a very stringent iterative process, DSDM can handle changing requirements to better create relevant and effective systems.
- **Time Constraints**
 - **Fixed Timeframes:** DSDM mostly does time-boxing approach that keeps your project to stay on track and deliverables in deadlines so far.
- **Risk Management**
 - **High Risk:** The early risk identification and mitigation practices of DSDM ensures significant technical as well as operational risks to be sufficiently managed.

Chapter 7: Foundation

7.1 Problem Area Identification

i. Interview

To identify key problems for Quick Edu Live, interviews were conducted with teachers, students and school administrators to gather insights into current educational challenges.

- Teachers: Worried about the time spent on creating quizzes, grading and record keeping looking for ways of automating these tools so you can concentrate more in teaching.
- Students: Noted that waiting long periods for feedback on quizzes and assignments proved detrimental to their learning, underscored a call for feedback immediacy.
- School Administrators: Had difficulties with ensuring standard of education over time and protecting large amounts data of students.

ii. Observation

Classroom observations: Key challenges

- Classroom Dynamics: Teachers dedicate hours to clerical work such as taking attendance, creating tests and grading papers which cuts down on time spent teaching interactively.
- Feedback Delays: Delays in feedback, students slow learning from their mistakes and iterations given they would have to wait till the course ends.
- Data Management: This is a lot of work and that student data management carries deep concerns over privacy.

iii. Questionnaires

Teachers, students and administrators completed quantitative questionnaires

- Administrative Workload: More than 30% of teachers' time is wasted on admin work, driving the necessity for automation tools.
- Feedback Mechanisms: The other side of the Big Insights coin was that 70% of students said if feedback is late, it really alters their performance.
- Data Security: 80% of administrators: Secure Data Management

7.2 Specific Problem Area Identification and Description

Based on the data collected, the following specific problem areas were identified:

High Administrative Workload for Teachers: Generation of tests, material submissions, quizzing results and grading are just a few hurdles some teachers face in another screen. It's an administrative nightmare. This results in less time of theirs that they can focus on teaching and more difficult to engagement with students.

Delayed Feedback for Students: Manually grading results in delays of student feedback. Students need timely feedback for learning and the delays prevent students from experiencing, understanding and capitalizing on their errors to perform better at school.

Data Security Concerns: Centrally, securely managing the huge amount of student data is a major challenge. The security of data is a must to keep the trust and comply with conditions from regulators on handling, maintaining or destroying it.

7.3 Possible Solution

Quick Edu Live offers a comprehensive solution to address these problem areas by integrating advanced AI technologies and modern web development frameworks.

AI-Powered Administrative Tools

- ❖ AI Quiz Generator: Enables teachers to create quizzes faster and more consistently.
- ❖ AI Assignment Generator: Helps teachers save hours on creating assignments so they can spend time teaching instead.
- ❖ AI Paper Checker: Grades automatically by scanning pictures of answer scripts into text and reflects correct answers according to questions.

Real-Time Feedback Mechanisms

- ❖ Automatic Grading: Gives students immediate feedback in order to learn from mistakes and perform better quicker.

Data Management Securely

- ❖ Firebase Authentication: Makes certain that Platform access is secure, thus safeguarding Users.
- ❖ Data Encryption: Protects student information and align with the data-protection regulations.
- ❖ Scalable Infrastructure: Friendly and easily-maintainable data repository with Firebase hosting.

7.4 Overall Requirement List

The general specifications for Quick Edu Live are divided into Functional and Non-Functional Requirements:

Functional Requirements

- Users are able register / login with email, password or Google account.
- Role management apply role-based access control to teacher and student
- Class creation teachers can create, edit and delete classes.
- Students can simply enroll in the class with a unique class code.
- Teacher will be able to create quizzes using the AI Quiz Generator.
- Teacher will be able to create assignment using the AI Assignment Generator.
- Auto grading quiz system.
- Teachers can post text-based announcements for their classes.
- Students may reply to announcements
- Users can be able to view and edit their profiles.
- Teachers are able to print quizzes for off-line use.
- Teachers can print the assignment for offline.
- Class Management on which Teachers can view and share enrollment key.
- Student Deletion: Teachers can delete students from their classes.
- Students can measure how well they are doing on the quizzes and assignments.
- AI Paper Checker Teachers can purchase & use this too
- Users have the ability to report issues using the platform.
- Users can submit comments about their experience

Non-Functional Requirements

- The platform has to support increasing user numbers and data: Scalability.
- Performance Keep load times mostly to a few milliseconds and interactions immediate.
- Put in place robust security measures to safeguard user data.
- Data encryption at-the data within things and on the server as well as in transit.
- Easy to use for end-users.
- Check for compatibility on different devices and browsers.
- Structure system should be easy to maintain and update.
- Error message and methods to handle errors.
- Scale integrated third-party tools and services

7.5 Technology to Be Implemented

Quick Edu Live has been developed as a web-app, utilizing the following technologies:

Language: JavaScript

Frontend: React JS, DaisyUI (Tailwind CSS components) for responsive and modern UIs.

Backend: Node JS, Express JS (Server-side Logic and API Development).

Database: For flexible and scalable data storage - MongoDB.

AI Tools: Gemini-1. 5-Flash API for AI-based quiz and assignment be built, Tesseract.js. But the main Tesseract.js functionality used to convert image-to-text in AI Paper Checker.

Authentication - Firebase Auth for safe and secure login of a user.

Hosting: Firebase Hosting that serves our app on a simple, scalable and reliable hosting infrastructure.

7.6 Recommendations and Justifications

Implement AI-Powered Tools:

AI Quiz Generator, AI Assignment Generator and AI Paper Checker can reduce the administrative burden on teachers while assuring consistency & providing immediate feedback to students

Focus on Security and Data Privacy:

Secure student data and respect regulation using Firebase Authentication for users' management coupled with encryption.

Develop a User-Friendly Interface:

Build a frontend experience with React JS and Tailwind CSS that works wonderfully across browsers.

Chapter 8: Exploration

8.1 Old System Use Case

The system of the past makes traditional things on paper managing classroom activities happen. Below is a use case scenario to illustrate the core processes and challenges of this legacy system:

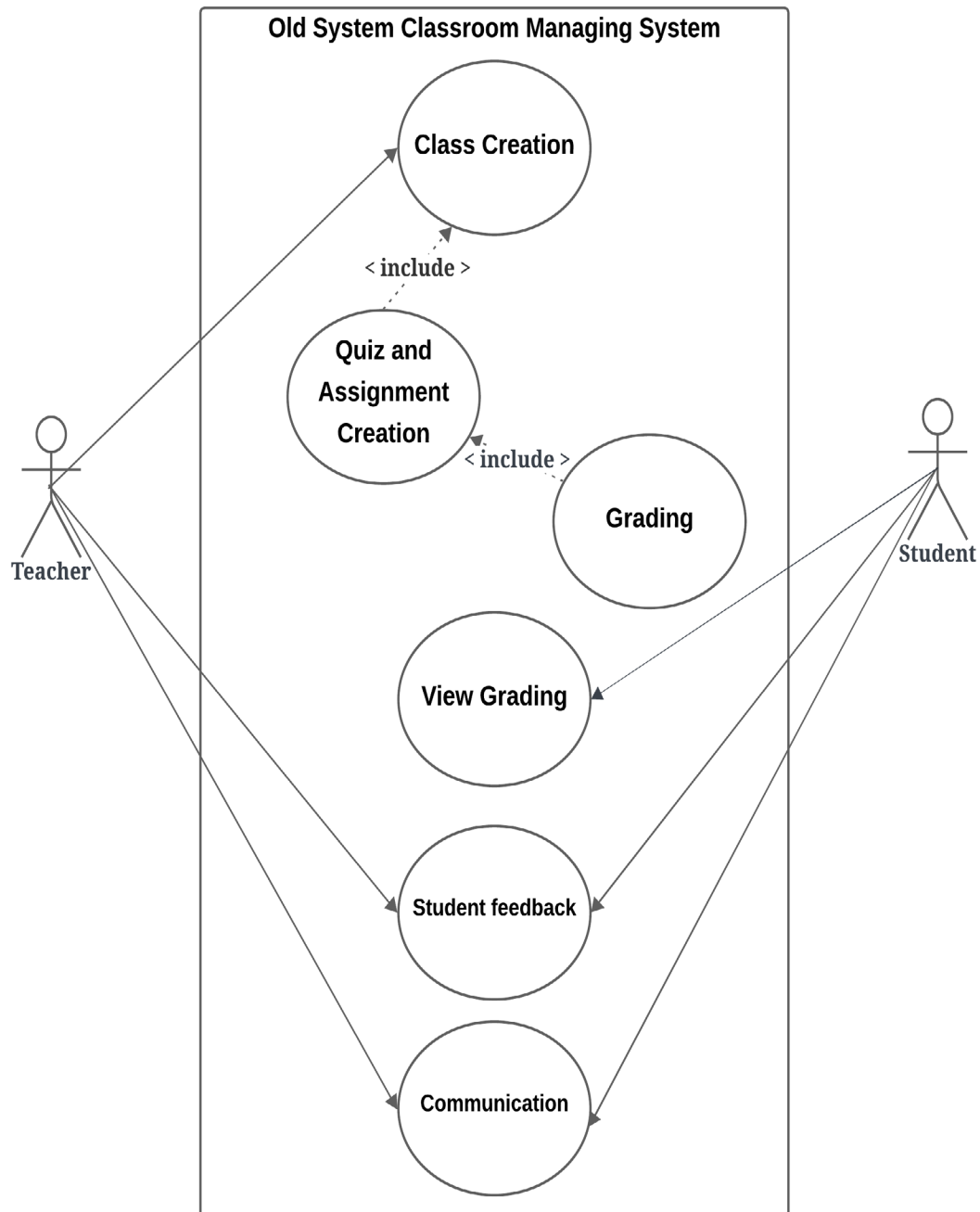


Figure 8.1: Use Case Diagram of Old System Classroom Managing

8.2 Activity Diagram

The activity diagram gives a graphical representation of workflow in the old system.

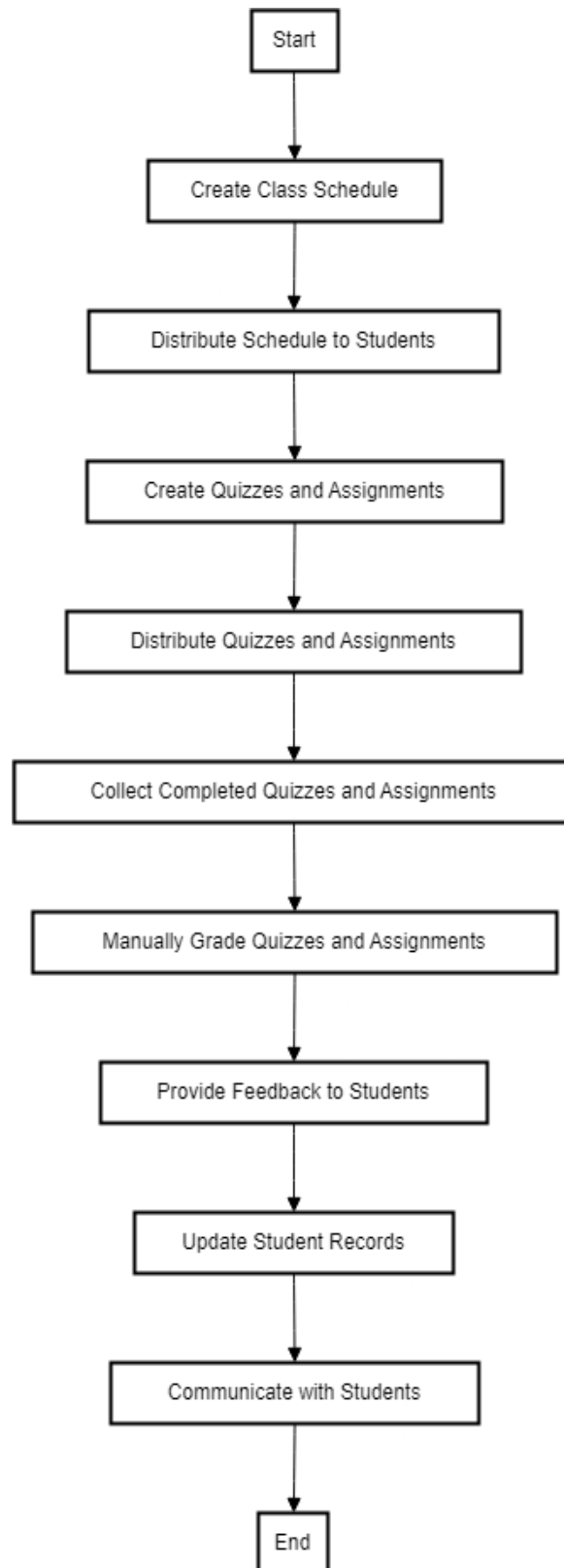


Figure 8.2: Activity Diagram Managing Classroom in the Old System

8.3 Full System Use Case

Quick Edu Live is the new learning and management system that uses AI and modern web technologies to quickly automate all classroom activities.

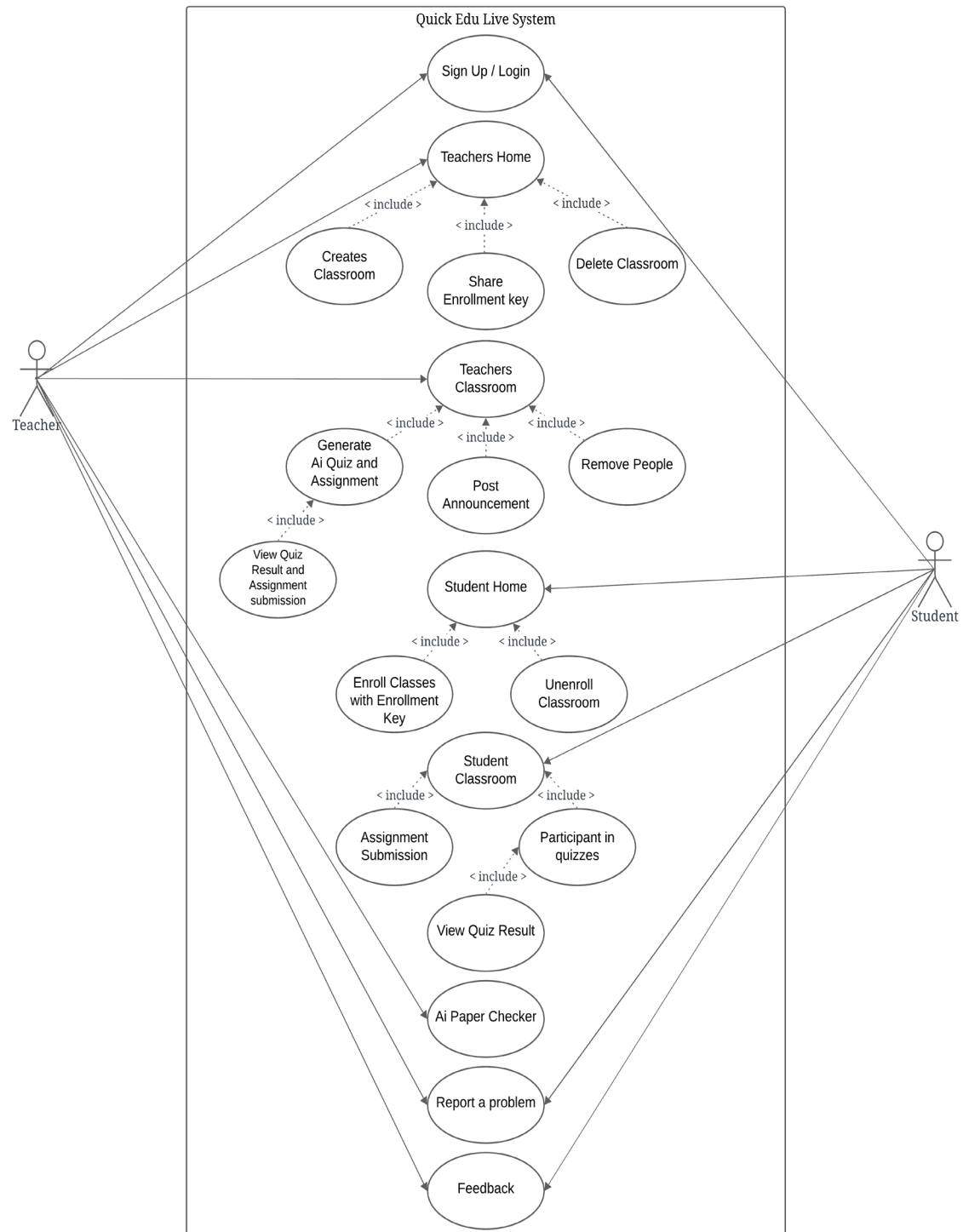


Figure 8.3: Use Case of Quick Edu Live

8.4 Full System Activity Diagram

The following activity diagram represents the simplified and automated workflow in Quick Edu Live.

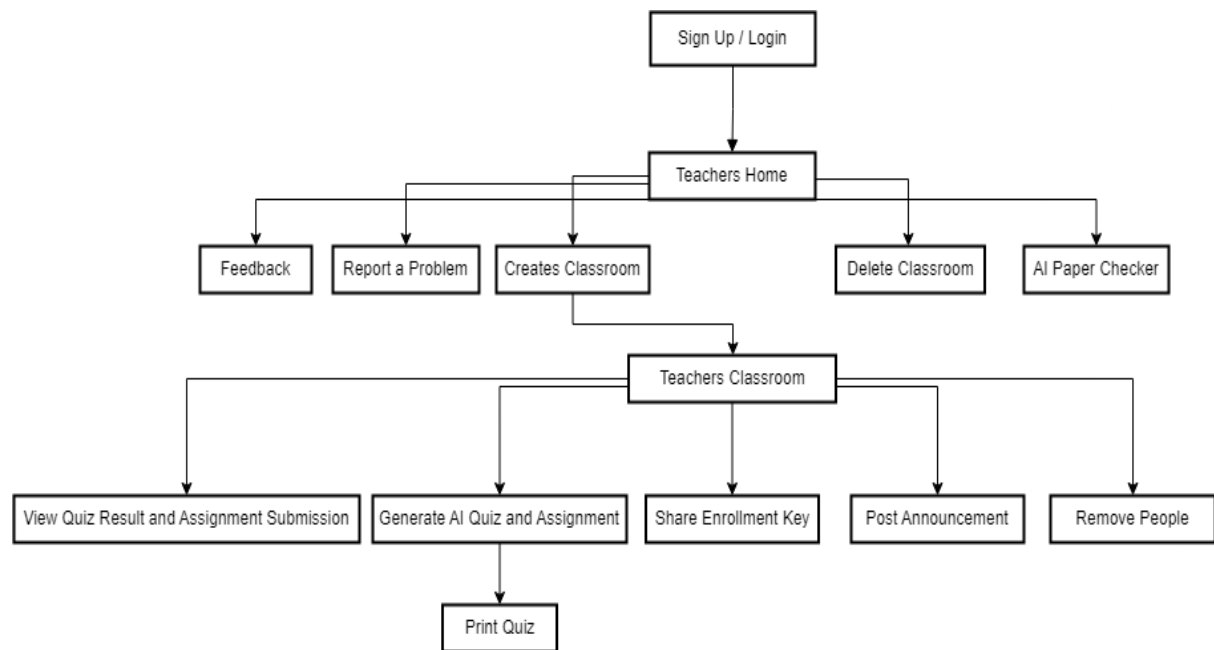


Figure 8.4: QEL Teachers Full System Activity Diagram

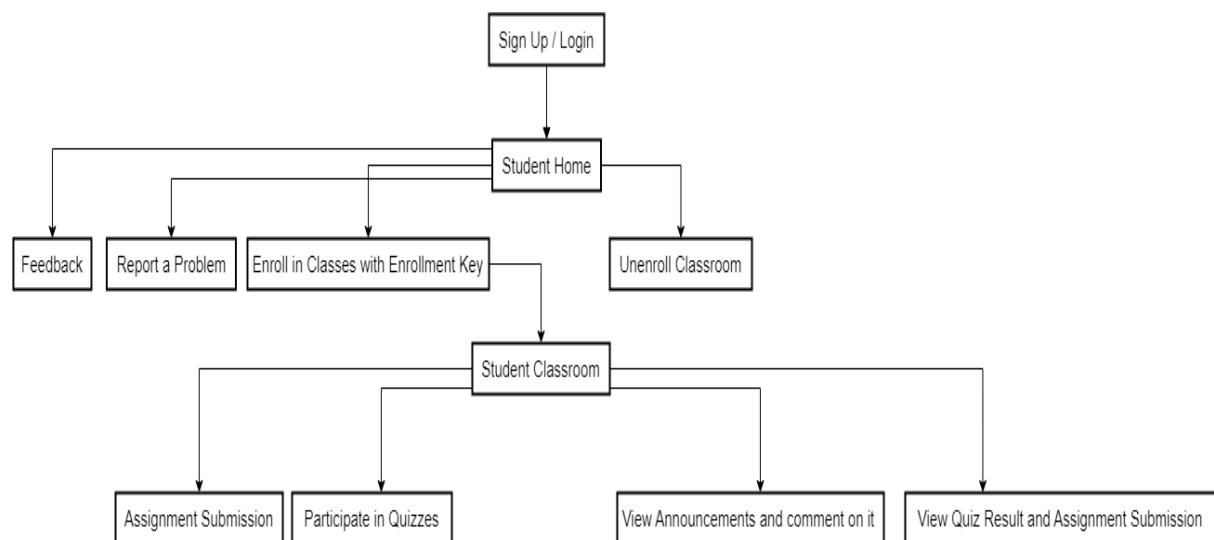


Figure 8.5: QEL Student Full System Activity Diagram

8.5 Prioritized Requirements List (PRL) MoSCoW prioritization technique:

R.NO.	Requirement Name	Prioritization
FR01	Registration and login for teachers and students	Must have
FR02	User authentication via email/password and Google	Must have
FR03	Create, manage and delete classrooms	Must have
FR04	Generate and manage AI quizzes and assignments	Must have
FR05	Post text announcements	Must have
FR06	Communicate with students via announcements	Must have
FR07	View list of enrolled students	Must have
FR08	Remove students from classes	Must have
FR09	Enroll students in classes using enrollment keys	Must have
FR10	View and manage quiz results and assignment submissions	Must have
FR11	Auto-grading for quizzes	Must have
FR12	Print quizzes for offline use	Must have
FR13	AI Paper Checker for grading assignments	Must have
FR14	Report a problem	Could have
FR15	Provide feedback	Should Have

Table 8.1: Prioritized Requirements List

Chapter 9: Engineering

9.1 New System Modules

Action	User Action	System Interaction
Teachers and Student Registration	Sign up for an account by providing necessary information.	Authorize user inputs, store user details securely.
Classroom Creation	Teacher creates a new classroom.	Validate classroom details, store classroom information securely and generate a unique classroom ID.
Enrollment Key Sharing	Teacher shares the enrollment key with students.	Generate and provide a unique enrollment key for the classroom.
Class Enrollment	Student enrolls in a class using the enrollment key.	Validate the enrollment key, add the student to the classroom and update classroom rosters.
Quiz and Assignment Generation	Teacher creates and generates AI-based quizzes and assignments.	Utilize the AI engine to generate quizzes and assignments, store them securely and associate them with the classroom.
Quiz and Assignment Submission	Student submits completed quizzes and assignments.	Validate submissions and store responses securely
Communication	Teacher and students communicate through the system.	Facilitate secure announcement in the classroom between teachers and students, store communication logs.

Table 9.1: Module of the System

9.2 Use Case

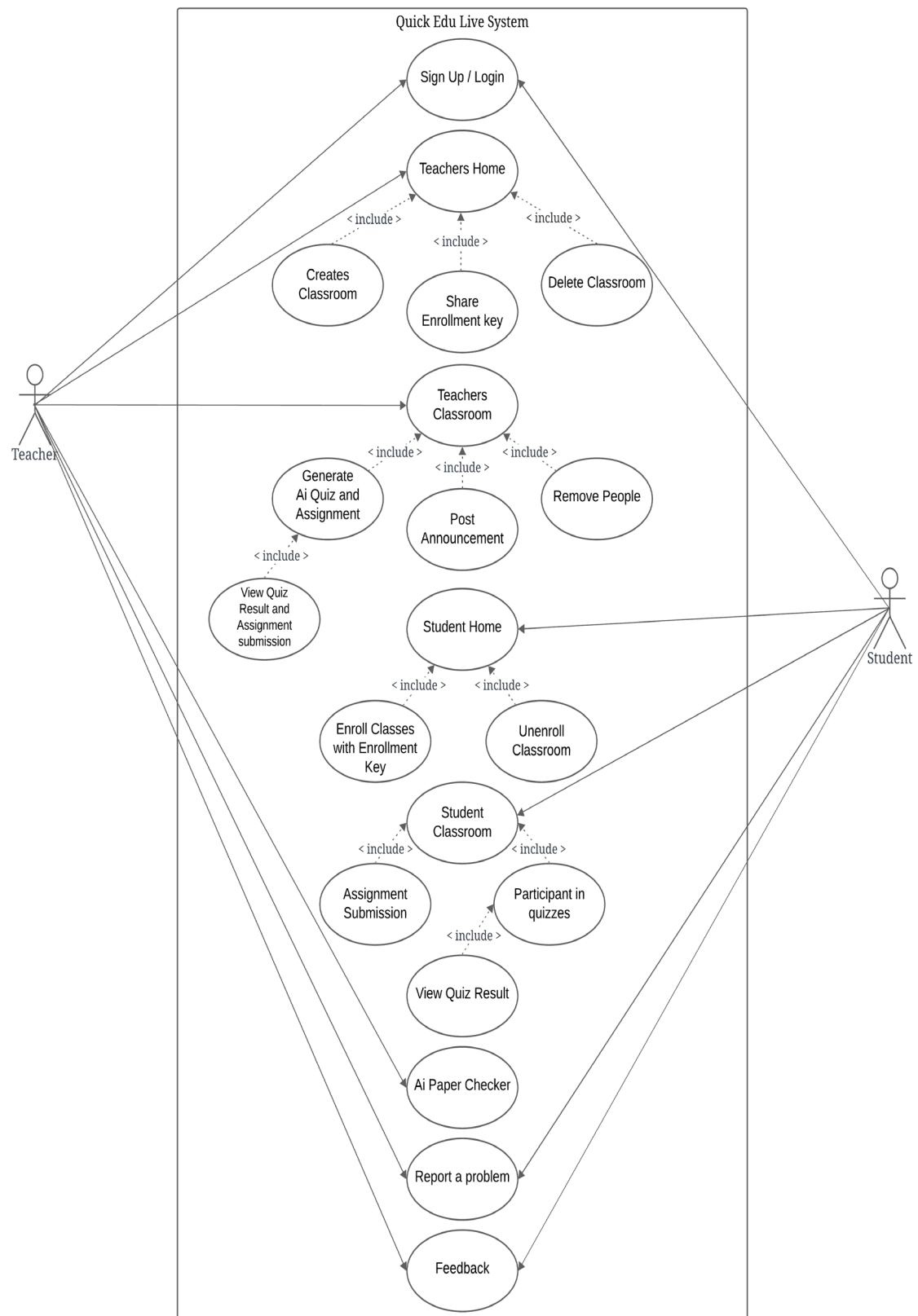


Figure 9.1: Use Case of Quick Edu Live

9.3 Class Diagram

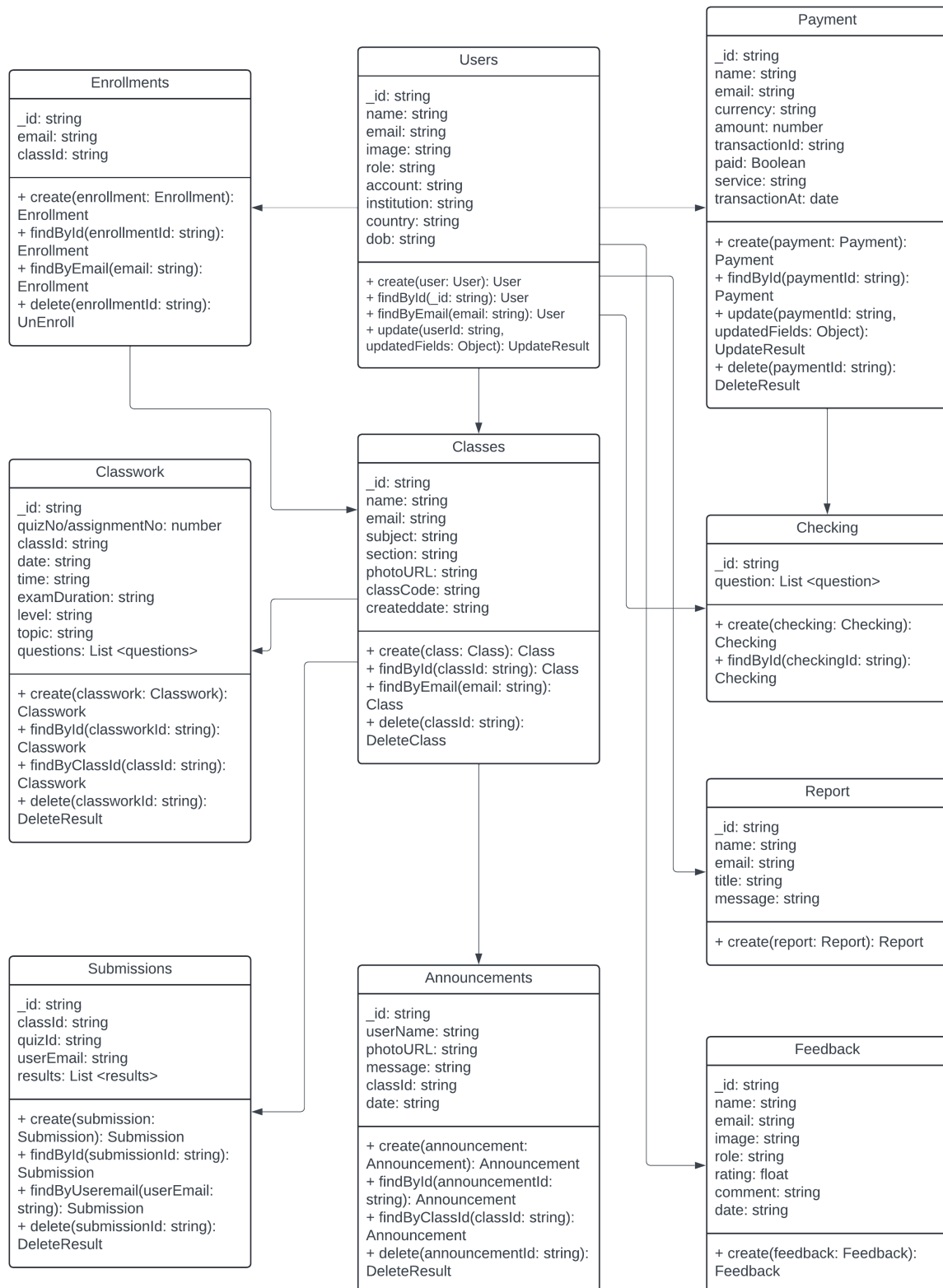


Figure 9.2: Quick Edu Live Class Diagram

9.4 Entity Relationship Diagram

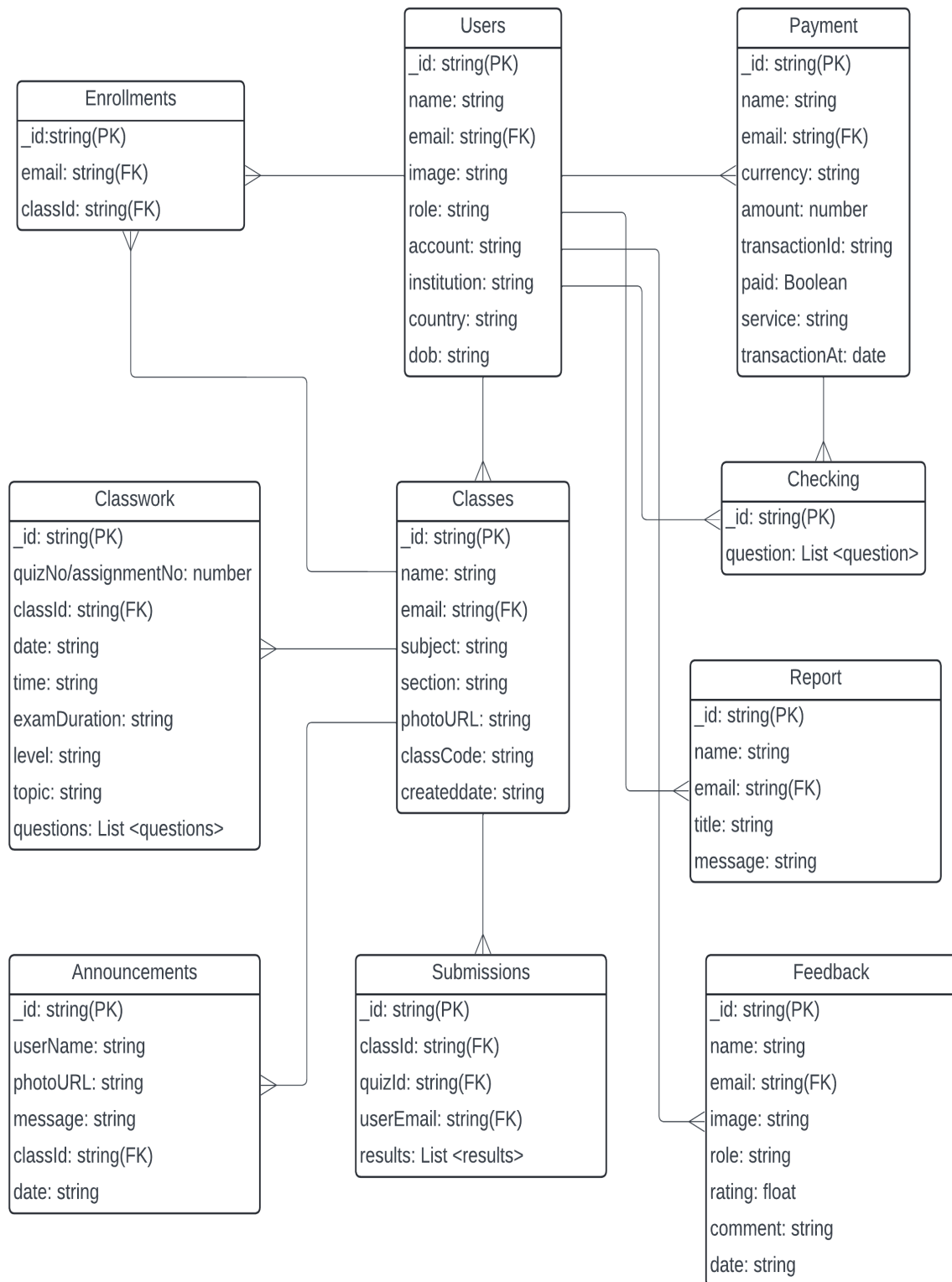


Figure 9.3: Quick Edu Live Entity Relationship Diagram

9.5 Sequence Diagram

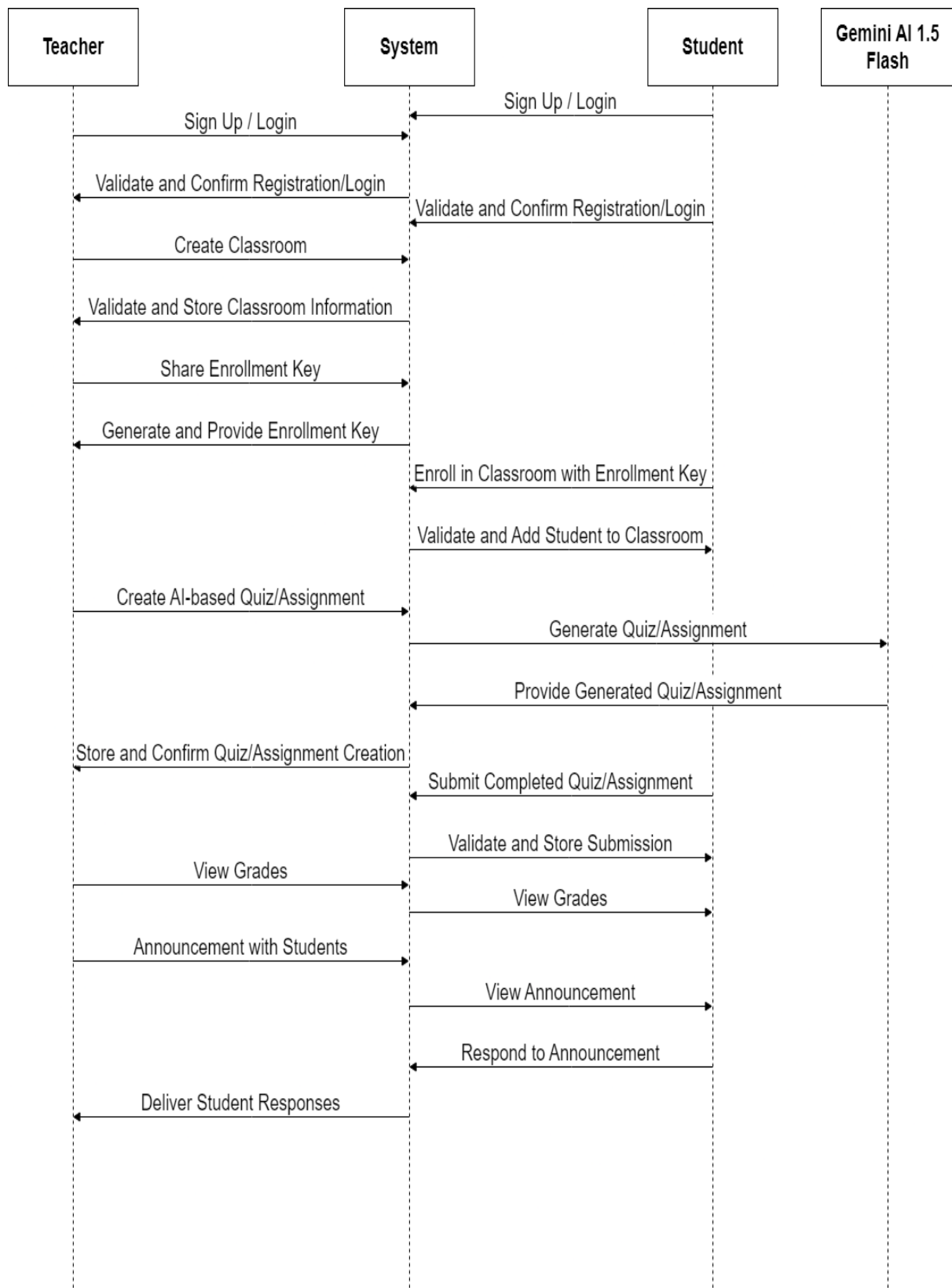


Figure 9.4: Quick Edu Live Sequence Diagram

9.6 System Interface Design

Interface Design of Home Page

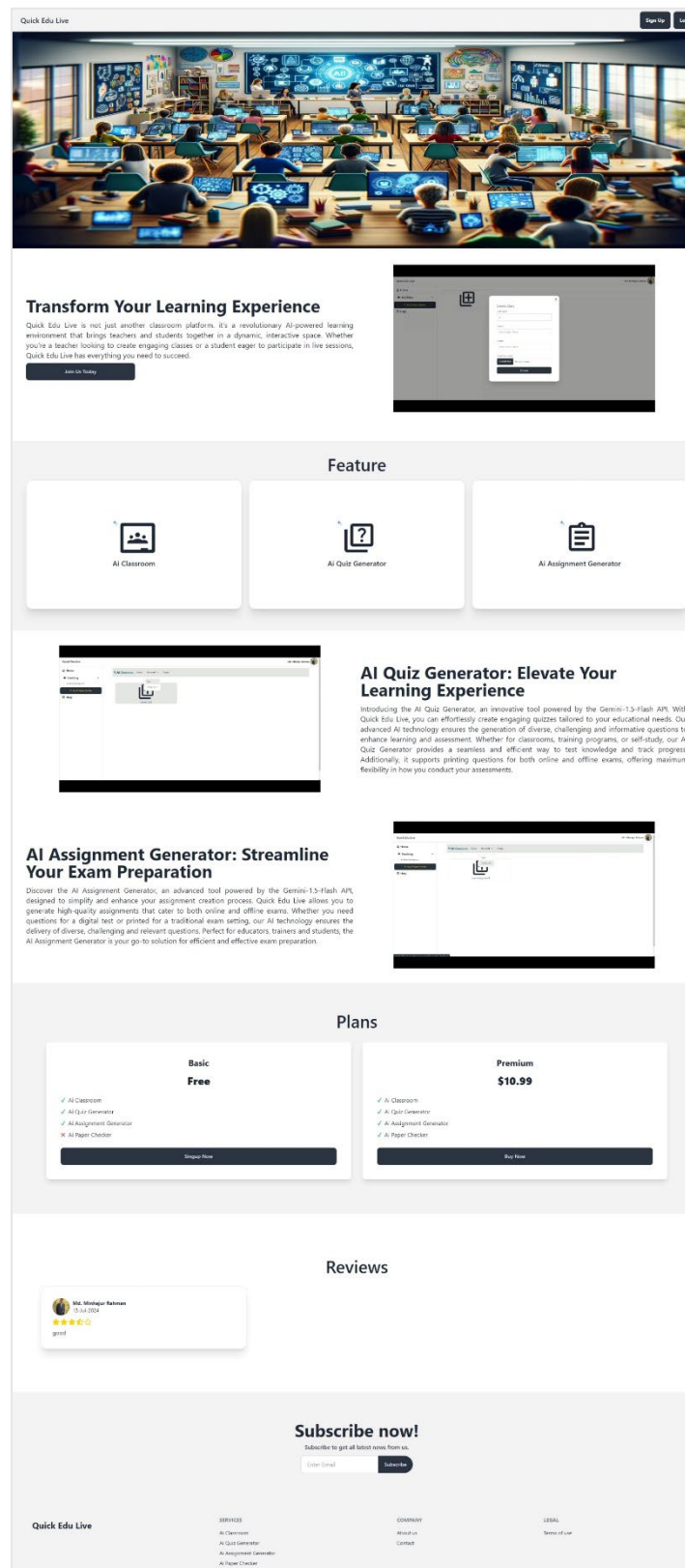


Figure 9.5: Home Page

Interface Design of Sign-Up Page

Quick Edu Live

[Sign Up](#)[Login](#)

Sign Up

Name

Mehedi Hasan

Institution

MBSTU

Country

Bangladesh

Date of Birth

01/01/1992

Role

☒ Teacher ☐ Student

Email

mehedihasan@gmail.com

Password

Confirm Password

Insert Your Photo

CHOOSE FILE

No file chosen

Sign Up

Google

Already Have Account! [Login](#)



Quick Edu Live

SERVICES

[AI Classroom](#)[AI Quiz Generator](#)[AI Assignment Generator](#)[AI Paper Checker](#)

COMPANY

[About us](#)[Contact](#)

LEGAL

[Terms of use](#)

Figure 9.6: Sign Up Page

Interface Design of Login Page

Quick Edu Live

[Sign Up](#)[Login](#)

Login

Email

mehedihasan@gmail.com

Password

Login

Google

Haven't an Account! [Signup](#)



Quick Edu Live

SERVICES

[AI Classroom](#)[AI Quiz Generator](#)[AI Assignment Generator](#)[AI Paper Checker](#)

COMPANY

[About us](#)[Contact](#)

LEGAL

[Terms of use](#)

Figure 9.7: Login Page

Interface Design of Teachers Home Page

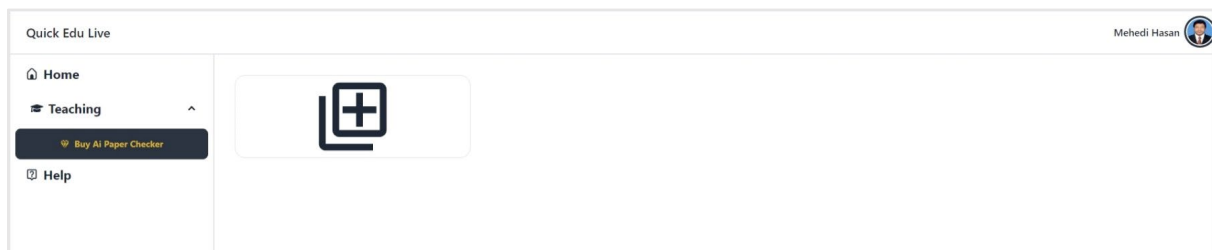


Figure 9.8: Teachers Home Page

Interface Design of Create Class Modal

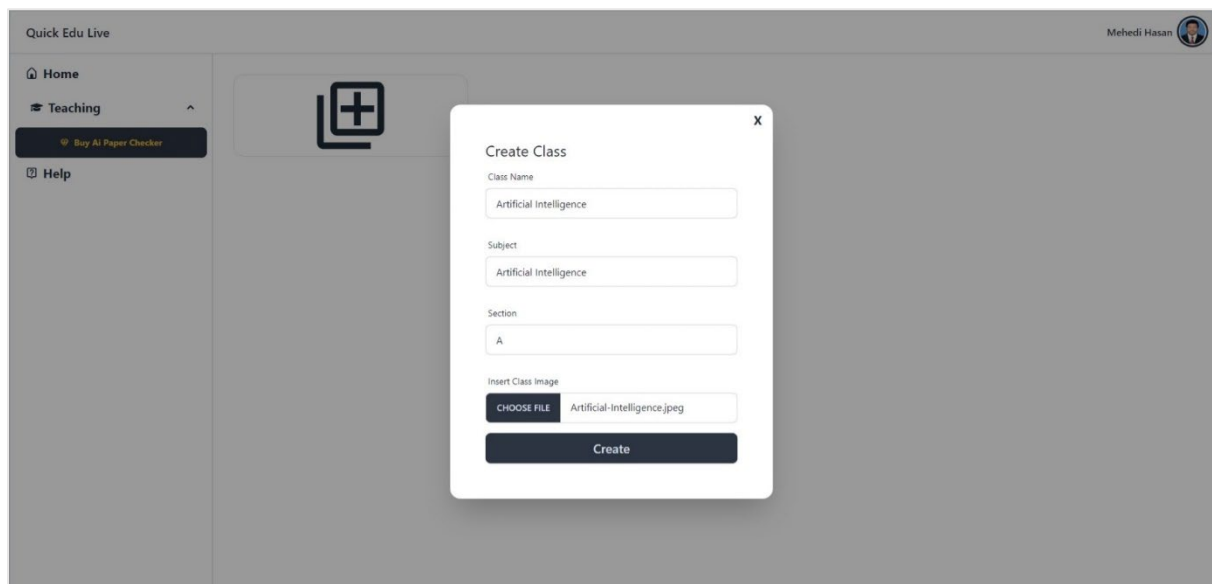


Figure 9.9: Create Class modal

Interface Design of Teachers Home Page with a Class

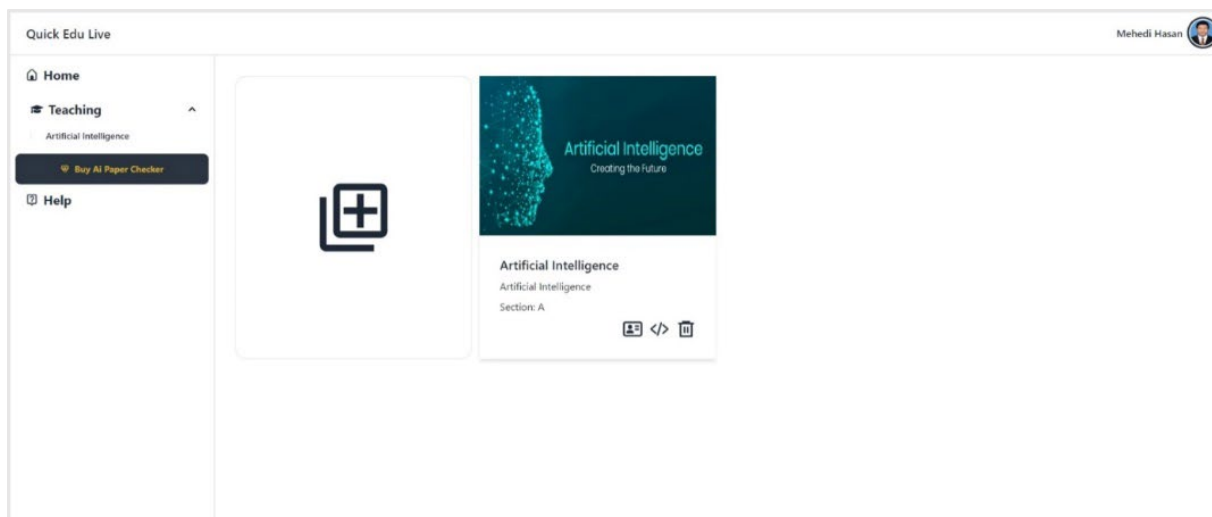


Figure 9.10: Teachers Home Page with a Class

Interface Design of Ai Classroom “Announcement” Page

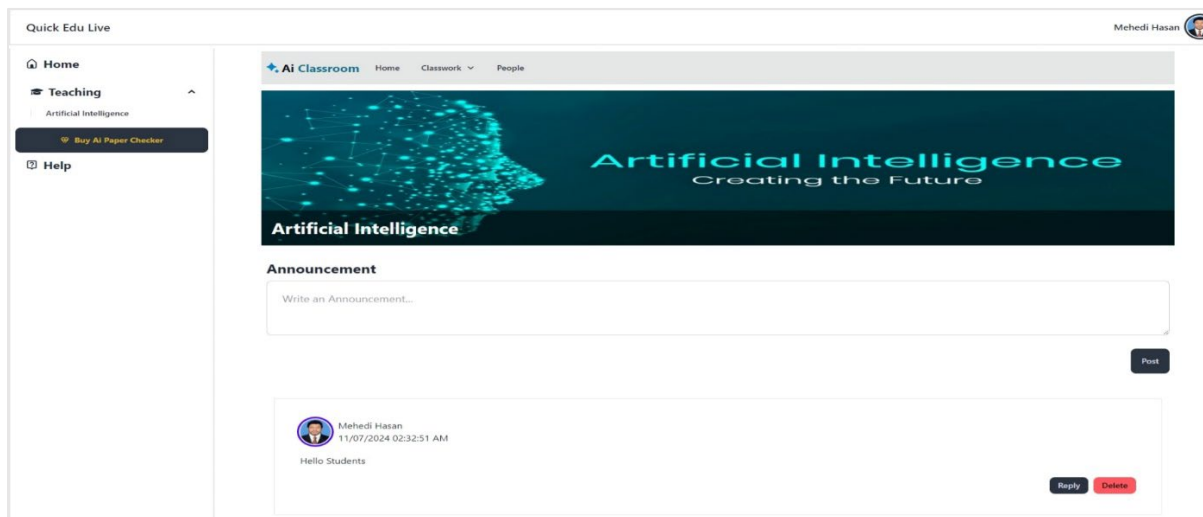


Figure 9.11: Ai Classroom “Announcement” Page

Interface Design of Ai Classroom “Classwork (Quiz)” Page

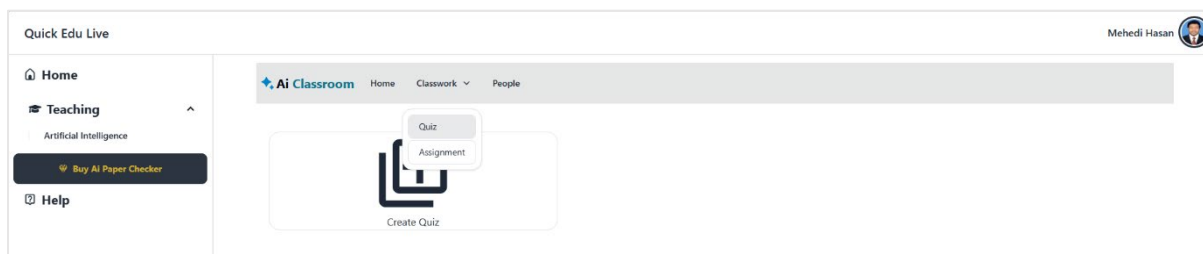


Figure 9.12: Ai Classroom “Classwork (Quiz)” Page

Interface Design of Create Quiz Modal

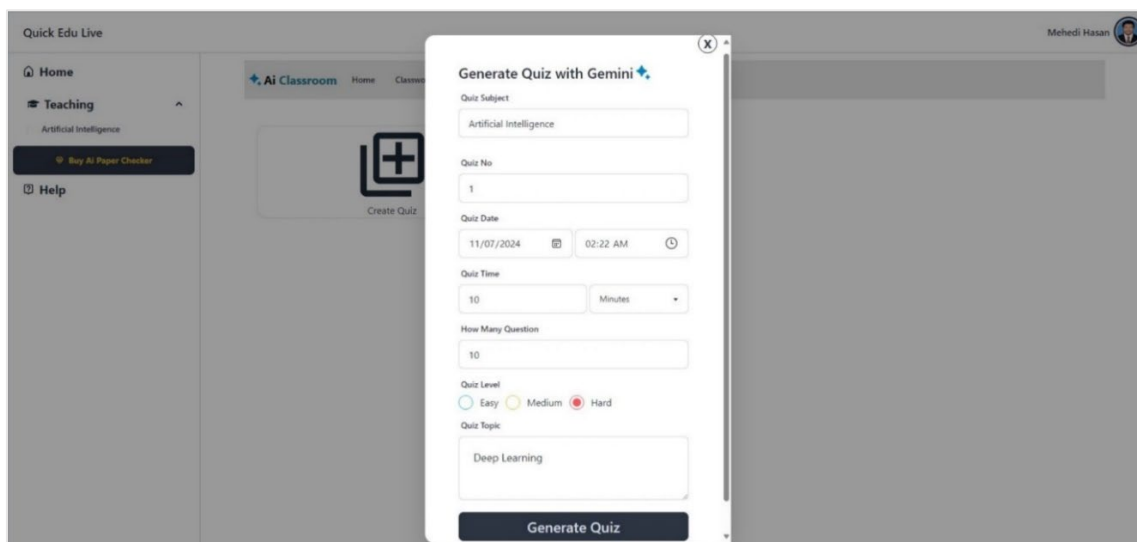


Figure 9.13: Quiz Modal

Interface Design of Ai Classroom “Classwork (Quiz)” Page with created quiz

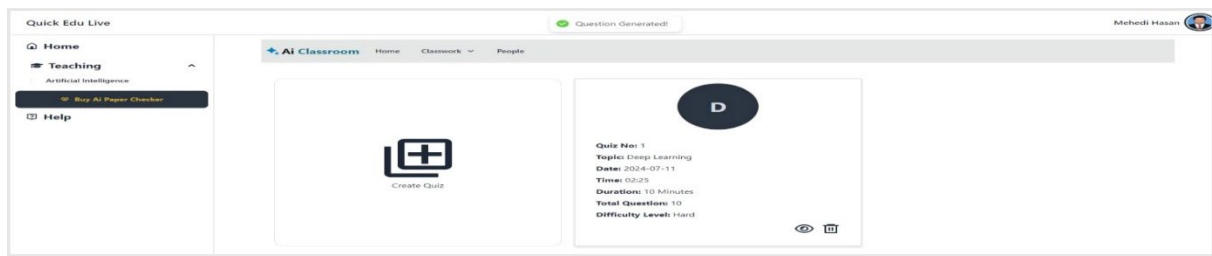


Figure 9.14: Ai Classroom “Classwork (Quiz)” Page with created quiz

Interface Design of View Quiz Question Page



Figure 9.15: View Quiz Question Page

Interface Design of Print Quiz / Assignment Question



Figure 9.16: Print Quiz

Interface Design of Ai Classroom “Classwork (Assignment)” Page

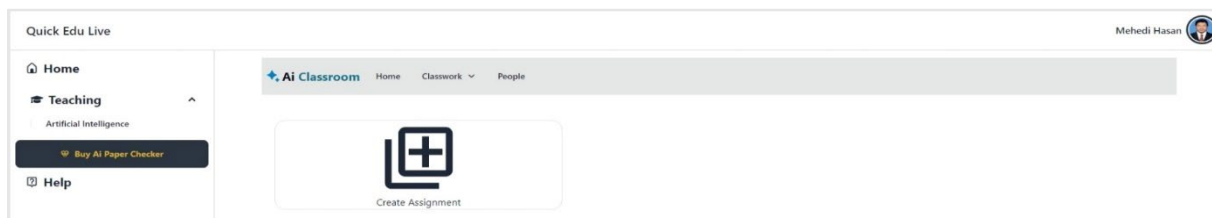


Figure 9.17: Ai Classroom “Classwork (Assignment)” Page

Interface Design of Assignment Modal



Figure 9.18: Assignment Modal

Interface Design of Ai Classroom “Classwork (Assignment)” Page with created assignment



Figure 9.19: Ai Classroom “Classwork (Assignment)” Page with created assignment

Interface Design of View Assignment Question Page

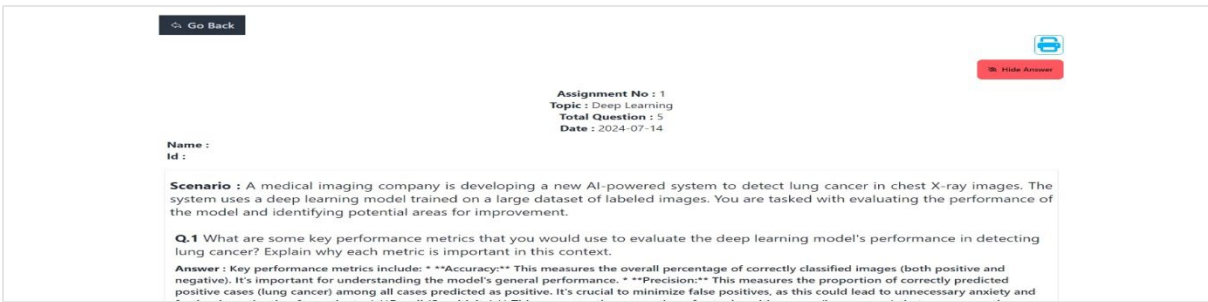


Figure 9.20: View Assignment Question Page

Interface Design of Print Assignment Question

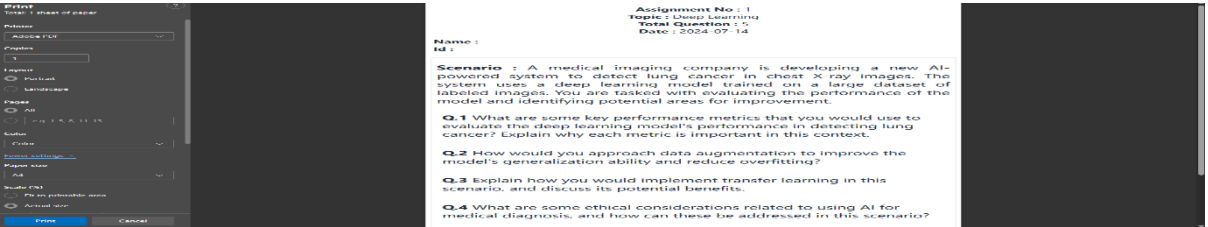


Figure 9.21: Print Assignment

Interface Design of View Classroom People Page



Figure 9.22. View Classroom People

Interface Design of Buying Ai Paper Checker



Figure 9.23: Buying Ai Paper Checker

Interface Design of SSLCOMMERZ Demo Payment Page

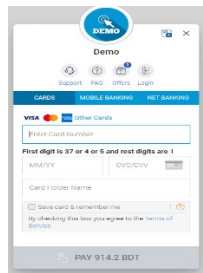


Figure 9.24: SSLCOMMERZ Demo Payment Page

Interface Design of Payment History Page

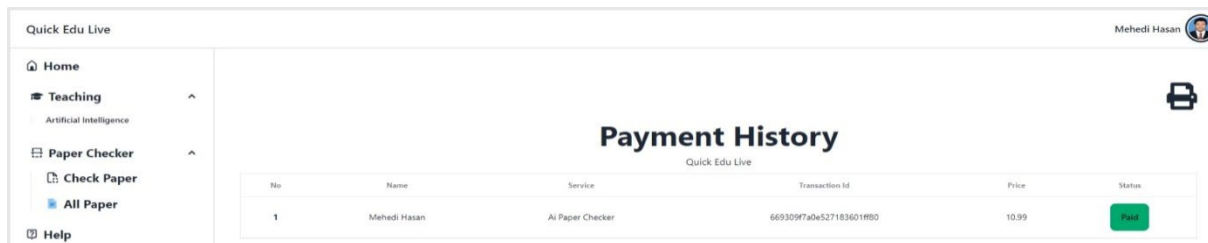


Figure 9.25: Payment History Page

Interface Design of Ai Paper Checker

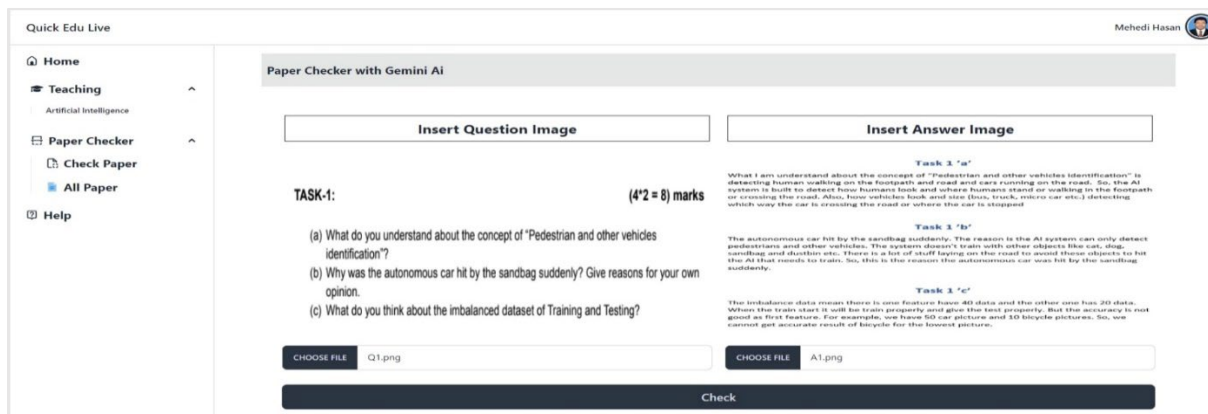


Figure 9.26: Ai Paper Checker

Interface Design of Ai Paper Checker Result

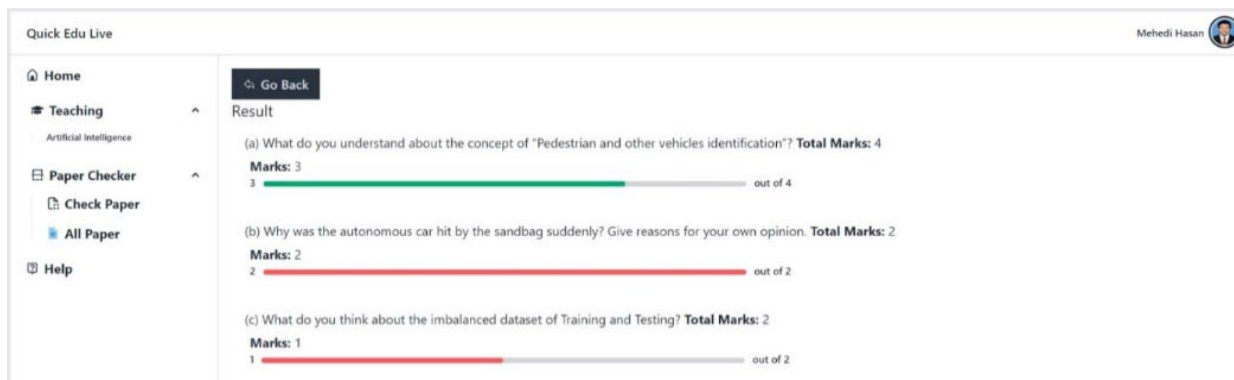


Figure 9.27: Ai Paper Checker Result

Interface Design of Students Home Page

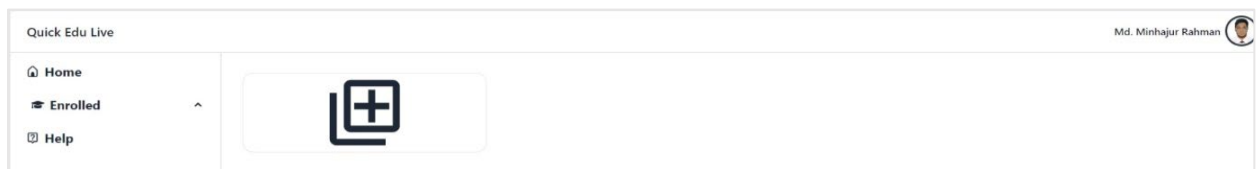


Figure 9.28: Students Home Page

Interface Design of Enroll Class Modal

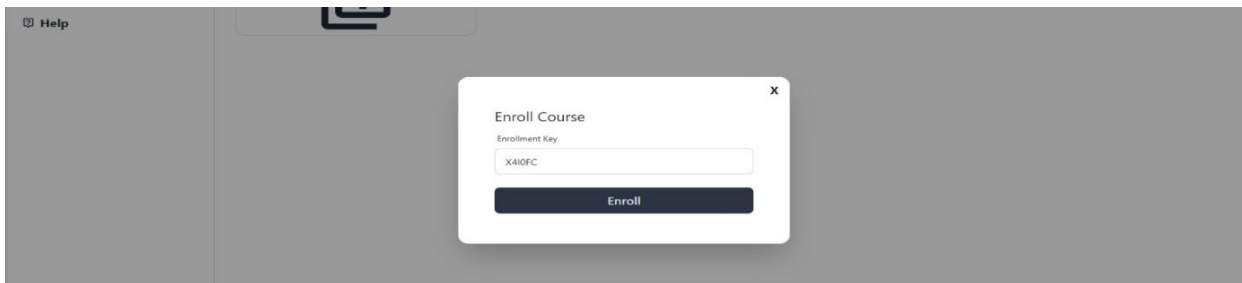


Figure 9.29: Enroll Class Modal

Interface Design of Students Home Page with a Enroll Class

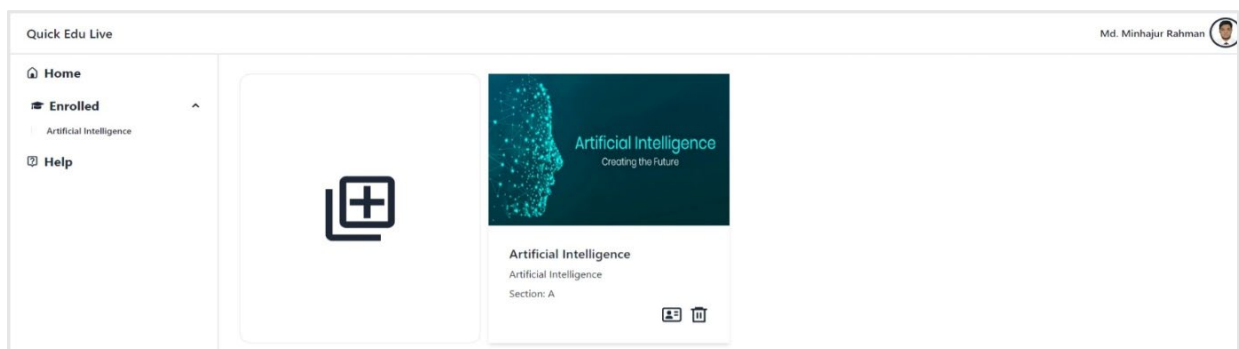


Figure 9.30: Students Home Page with a Enroll Class

Interface Design of Ai Classroom “Classwork (Undergoing Quiz)” Page

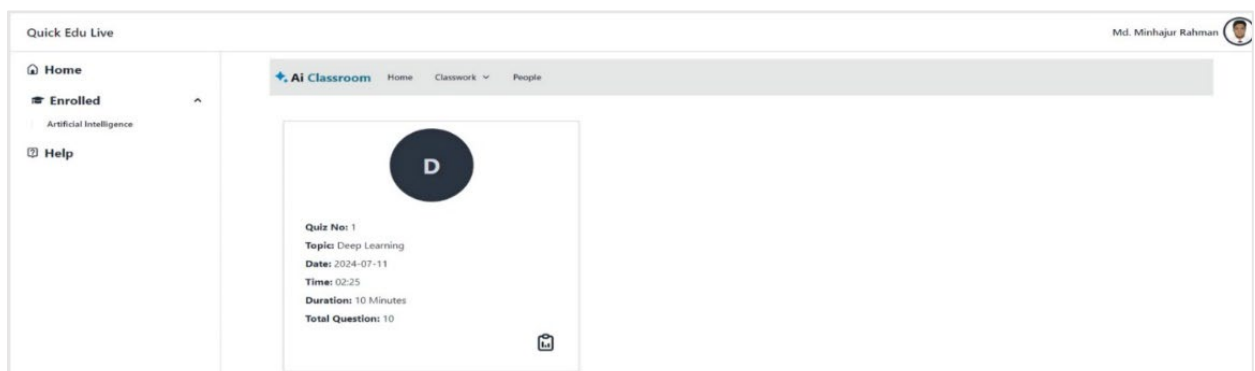


Figure 9.31: Ai Classroom “Classwork (Undergoing Quiz)” Page

Interface Design of Performing Quiz

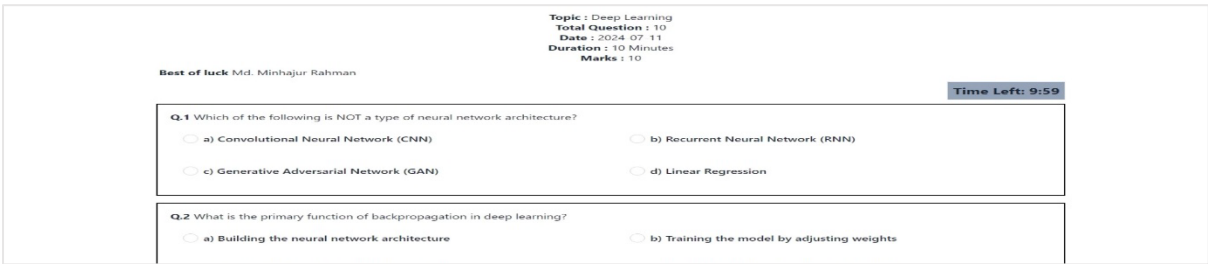


Figure 9.32: Performing Quiz

Interface Design of Completed Quiz

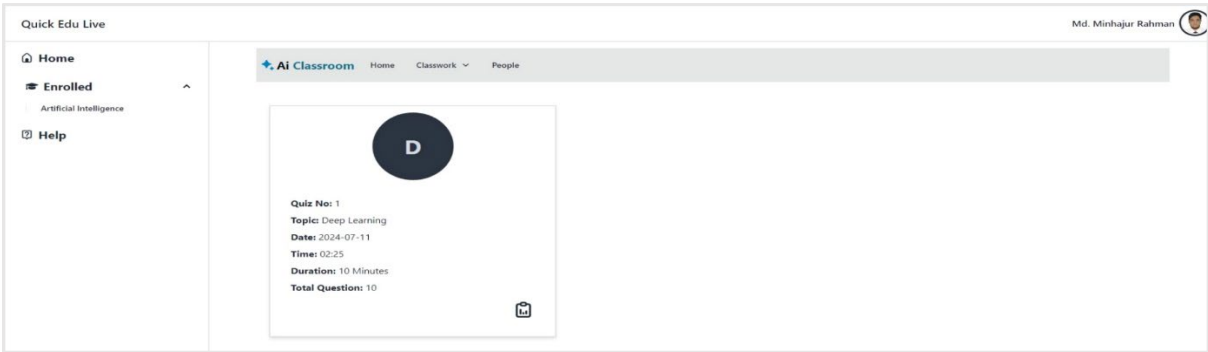


Figure 9.33: Completed Quiz

Interface Design of View Quiz Marks

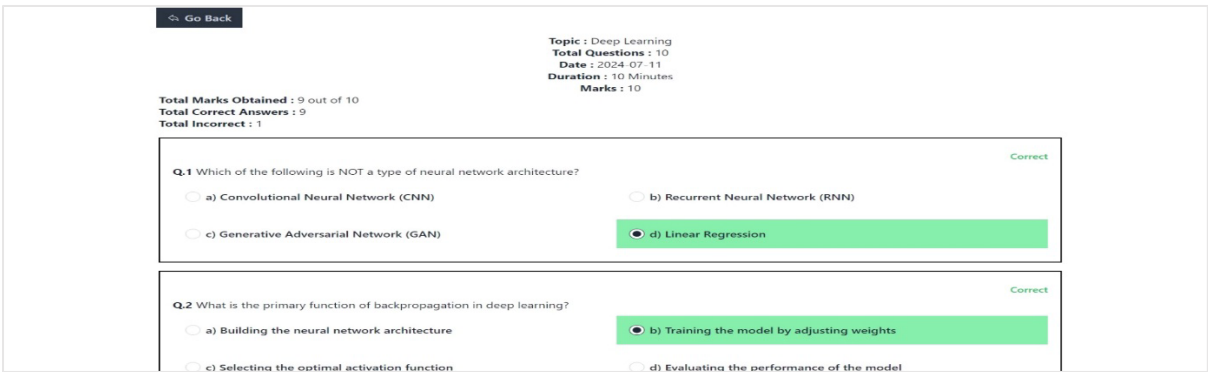


Figure 9.34: View Quiz Marks

Interface Design of Ai Classroom “Classwork (Assignment)” Page

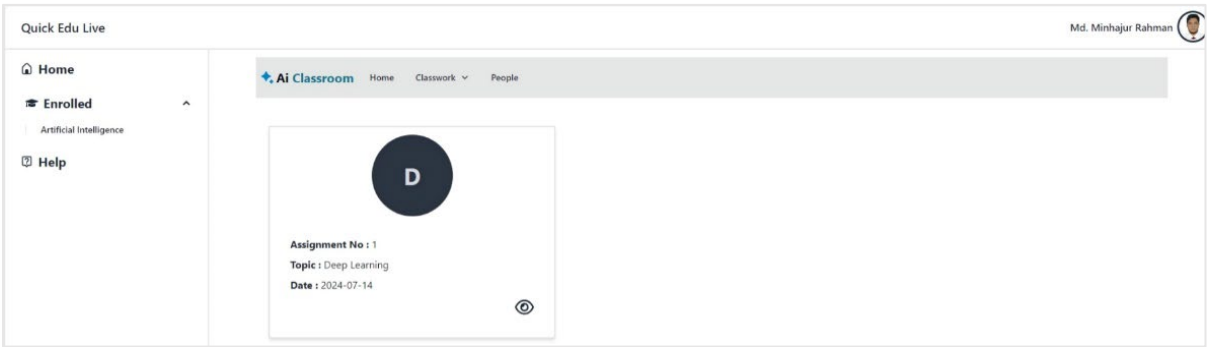


Figure 9.35: Ai Classroom “Classwork (Assignment)” Page

Interface Design of Ai Classroom “Classwork (Undergoing Assignment)” Page



Figure 9.36: Ai Classroom “Classwork (Undergoing Assignment)” Page

Interface Design of Ai Classroom “Classwork (Assignment)” Submission Page

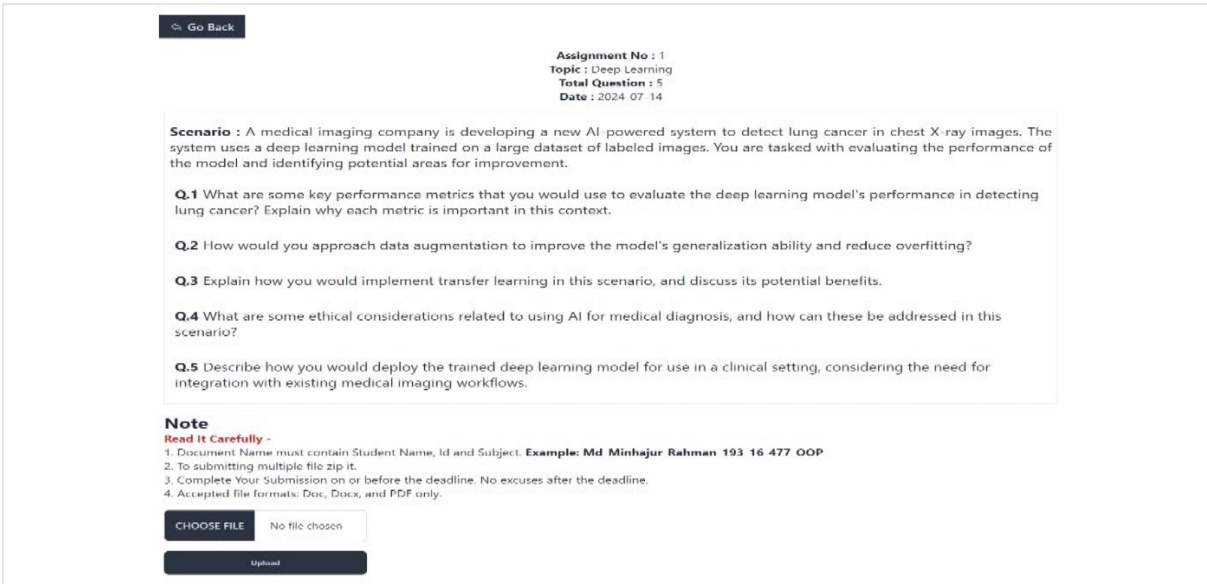


Figure 9.37: Assignment Submission

Interface Design of Ai Classroom “Classwork (Assignment)” Submission Page

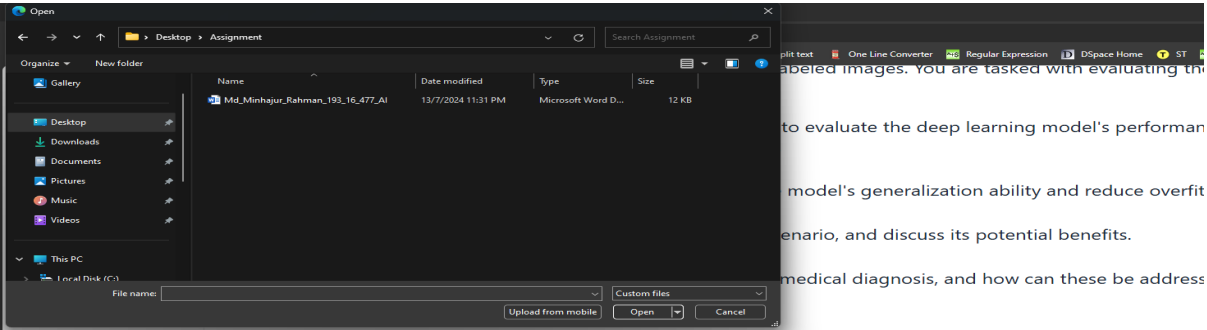


Figure 9.38: Assignment Submitting

Interface Design of Ai Classroom “Classwork (Assignment)” Submitted Page

Submission Done	
File Name	Action
Md_Minhajur_Rahman_193_16_477_AI.docx	 

Figure 9.39: Ai Classroom “Classwork (Assignment)” Submitted Page

Interface Design of Help Page

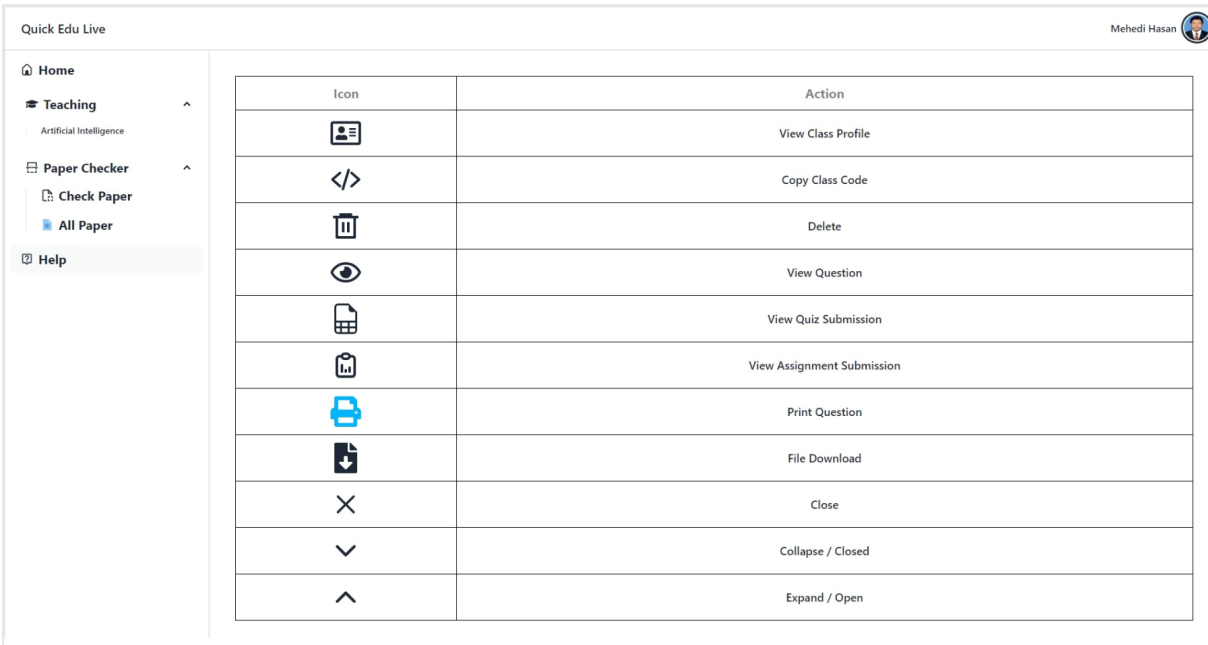


Figure 9.40: Help Page

Interface Design of Feedback Page

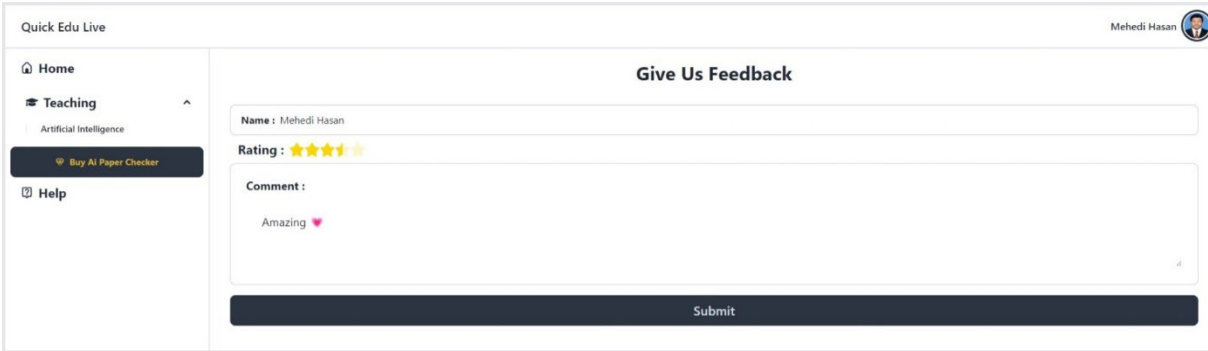


Figure 9.41: Feedback Page

Interface Design of Report a problem Page

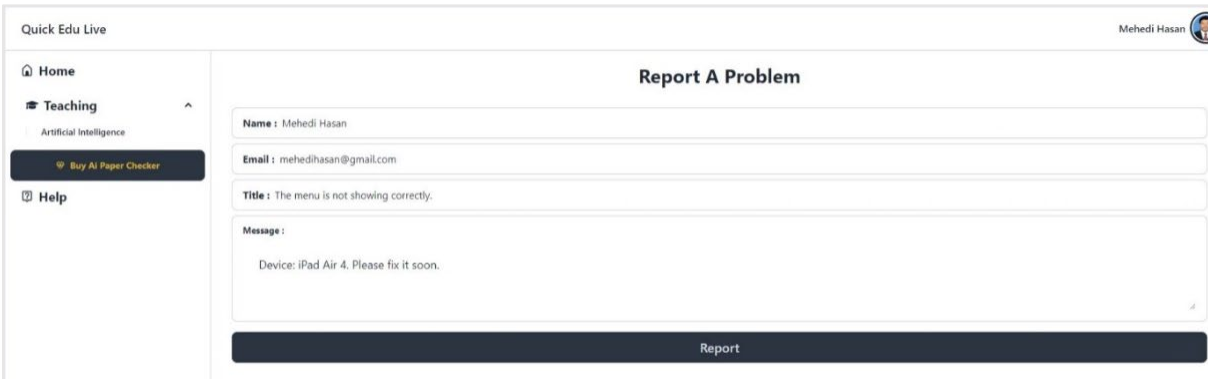


Figure 9.42: Report a Problem Page

Chapter 10: Deployment

10.1 Core Module Coding Samples

The technologies using to develop Quick Edu Live are ReactJS and Tailwind CSS components (DaisyUI) for the frontend and NodeJS and ExpressJS for the backend. For the database, I used MongoDBs.

Registration Sample for the users

```
1 // Import the functions you need from the SDKs you need
2 import { initializeApp } from "firebase/app";
3 // TODO: Add SDKs for Firebase products that you want to use
4 // https://firebase.google.com/docs/web/setup#available-libraries
5
6 // Your web app's Firebase configuration
7 const firebaseConfig = {
8   apiKey: process.env.REACT_APP_apiKey,
9   authDomain: process.env.REACT_APP_authDomain,
10  projectId: process.env.REACT_APP_projectId,
11  storageBucket: process.env.REACT_APP_storageBucket,
12  messagingSenderId: process.env.REACT_APP_messagingSenderId,
13  appId: process.env.REACT_APP_appId
14};
15
16// Initialize Firebase
17const app = initializeApp(firebaseConfig);
18export default app;
```

```
1 import React, { useEffect, useState } from 'react';
2 import { createUserWithEmailAndPassword, getAuth, onAuthStateChanged, signInWithEmailAndPassword, signInWithPopup, signOut, updateProfile } from 'firebase/auth';
3 import app from '../firebase/firebase.config';
4
5 // const auth = getAuth(app);
6
7 // const AuthProvider = ({ children }) => {
8   // const [user, setUser] = useState(null);
9   // const [loading, setLoading] = useState(true);
10  // const googleProvider = new GoogleAuthProvider();
11  //
12  // const signInWithGoogle = () => {
13  //   return signInWithPopup(auth, googleProvider);
14  // }
15  //
16  // const createUser = (email, password) => {
17  //   setLoading(true);
18  //   return createUserWithEmailAndPassword(auth, email, password);
19  // }
20  //
21  // const signIn = (email, password) => {
22  //   setLoading(true);
23  //   return signInWithEmailAndPassword(auth, email, password);
24  // }
25  //
26  // const updateUser = (userInfo) => {
27  //   return updateProfile(auth.currentUser, userInfo);
28  // }
29  //
30  // const logout = () => {
31  //   setLoading(true);
32  //   return signOut(auth);
33  // }
34  //
35  // useEffect(() => {
36  //   const unsubscribe = onAuthStateChanged(auth, currentUser => {
37  //     setUser(currentUser);
38  //     setLoading(false);
39  //   });
40  //   return () => unsubscribe();
41  // }, []);
42  //
43  // const authInfo = {
44  //   createUser,
45  //   signIn,
46  //   updateUser,
47  //   logout,
48  //   signInWithGoogle,
49  //   user,
50  //   loading
51  // }
52  //
53  // return (
54  //   <AuthContext.Provider value={authInfo}>
55  //     {children}
56  //   </AuthContext.Provider>
57  // );
58 // }
59 //
60 // export default AuthProvider;
```

```

1 import React, { useContext, useEffect, useState } from 'react';
2 import { Link, useLocation, useNavigate } from 'react-router-dom';
3 import signupImage from '../assets/signup/Signup.png';
4 import { useForm } from 'react-hook-form';
5 import { FcGoogle } from 'react-icons/fc';
6 import { AuthContext } from '../contexts/AuthProvider';
7 import toast from 'react-hot-toast';
8 import useToken from '../hooks/useToken';
9 import { Helmet } from 'react-helmet-async';
10 const Signup = () => {
11   const { register, formState: { errors }, handleSubmit, watch } = useForm();
12   const { createUser, updateUser, signInWithGoogle } = useContext(AuthContext);
13   const imageHostKey = process.env.REACT_APP_img8_key;
14   const [signupError, setSignupError] = useState("");
15   const [createdUserEmail, setCreatedUserEmail] = useState("");
16   const [token] = useToken(createdUserEmail);
17   const location = useLocation();
18   const navigate = useNavigate();
19   const from = location.state?.from?.pathname || "/home";
20
21   const handleSignup = (data, event) => {
22     setSignupError("");
23
24     // Create New User Function
25     createUser(data.email, data.password)
26       .then(result => {
27         const loggedUser = result.user;
28         const image = data.image[0];
29         const formData = new FormData();
30         formData.append('image', image);
31         fetch(`https://api.imgbb.com/1/upload?key=${imageHostKey}`,
32           {
33             method: 'POST',
34             body: formData
35           })
36           .then(res => res.json())
37           .then(imageData => {
38             const userInfo = {
39               displayName: data.name,
40               photoURL: imageData.data.url
41             };
42             updateUser(userInfo)
43               .then(() => {
44                 const user = {
45                   name: data.name,
46                   email: data.email,
47                   image: imageData.data.url,
48                   role: data.role,
49                   account: "",
50                   institution: data.institution,
51                   country: data.country,
52                   dob: data.dob
53                 };
54                 saveUser(user);
55                 event.target.reset();
56               })
57               .catch(error => {
58                 setSignupError(error.message);
59               })
60             })
61           .catch(error => {
62             setSignupError(error.message);
63           })
64         });
65   };
66
67   const handleGoogleSignIn = () => {
68     signInWithGoogle()
69       .then(result => {
70         const loggedUser = result.user;
71         const user = {
72           name: loggedUser.displayName,
73           email: loggedUser.email,
74           image: loggedUser.photoURL,
75           role: "",
76           account: "",
77           institution: "",
78           country: "",
79           dob: ""
80         };
81         saveUser(user);
82       })
83       .catch(error => console.error(error));
84   };
85
86   const saveUser = user => {
87     fetch('http://localhost:5000/users', {
88       method: 'POST',
89       headers: {
90         'content-type': 'application/json'
91       },
92       body: JSON.stringify(user)
93     })
94       .then(res => res.json())
95       .then(data => {
96         // console.log(data);
97         setCreatedUserEmail(user.email);
98         toast.success(data.message);
99       })
100     };
101
102   useEffect(() => {
103     if (token) {
104       toast.success("Account Created!");
105       navigate(from, { replace: true });
106     }
107   }, [token, from, navigate]);
108
109   return (
110     <div className="hero my-10">
111       <div>
112         <img alt="Signup Image" data-bbox="297 92 691 906" style="width: 100px; height: 100px; object-fit: cover; border-radius: 50%; border: 1px solid #ccc; margin-bottom: 10px;"/>
113         <h2 style="text-align: center; margin: 0; font-size: 24px; font-weight: bold;">Sign Up</h2>
114       </div>
115       <div className="hero-content grid md:grid-cols-2 gap-20">
116         <div className="card shadow-2xl border">
117           <form onSubmit={handleSubmit(handleSignup)} className="card-body">
118             <div className="text-2xl text-center font-semibold">Signup</div>
119             <div className="form-control">
120               <label className="label">
121                 <span className="label-text">Name</span>
122               </label>
123               <input {...register("name", {
124                 required: { value: true, message: "Name is Required" }
125               })} type="text" placeholder="Enter Your Name" className="input input-bordered" />
126             </div>
127             <div className="label">
128               <div style="display: flex; align-items: center; justify-content: space-between; font-size: 0.9em; font-weight: bold; margin-bottom: 5px;">
129                 <span>Errors</span>
130                 <span style="color: red; font-weight: normal;">(errors.name.message)</span>
131               </div>
132             </div>
133             <div className="form-control">
134               <label className="label">
135                 <span className="label-text">Institution</span>
136               </label>
137               <input {...register("institution", {
138                 required: { value: true, message: "Institution Name is Required" }
139               })} type="text" placeholder="Enter Your Institution Name" className="input input-bordered" />
140             </div>
141             <div className="label">
142               <div style="display: flex; align-items: center; justify-content: space-between; font-size: 0.9em; font-weight: bold; margin-bottom: 5px;">
143                 <span>Errors</span>
144                 <span style="color: red; font-weight: normal;">(errors.institution.message)</span>
145               </div>
146             </div>
147           </form>
148         </div>

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```



```

1 app.post('/users', async (req, res) => {
2   const data = req.body;
3   const userExist = await usersCollection.findOne({email: data.email});
4   if(userExist){
5     return res.send({
6       success: false,
7       message: "User Already Exits"
8     });
9   }
10  const user = postData(usersCollection, data);
11  user
12    .then(result => {
13      return res.send({
14        success: true,
15        message: "New User Added",
16        data: result
17      });
18    })
19    .catch(err => {
20      return res.send({
21        success: false,
22        message: err?.message
23      });
24    });
25});

```

Figure 10.1: Signup Code Sample

Create classroom Sample code for teachers

```
1 import React, { useContext, useState } from 'react';
2 import Classes from '../Classes/Classes';
3 import ClassModal from '../ClassModal/ClassModal';
4 import { AuthContext } from '../../contexts/AuthProvider';
5 import useClasses from '../../hooks/useClasses';
6 import useTeacher from '../../hooks/useTeacher';
7 import Loading from '../Shared/Loading/Loading';
8 import useLoadUser from '../../hooks/useLoadUser';
9 import Enroll from '../ClassEnroll/Enroll/Enroll';
10 import { Helmet } from 'react-helmet-async';
11 import { Link } from 'react-router-dom';
12
13 const MyHome = () => {
14   const { user } = useContext(AuthContext);
15   const { userInfo, userIsLoading } = useLoadUser(user);
16   const [isTeacher, isTeacherLoading] = useTeacher(user?.email);
17   const { classes, classLoading, refetch } = useClasses(user);
18   const [modal, setModal] = useState(null);
19
20   if (userIsLoading || isTeacherLoading) {
21     return <Loading></Loading>;
22   }
23   return (
24     <div>
25       <Helmet>
26         <title>
27           {isTeacher ?
28             "Teacher's Home"
29             :
30             "Students's Home"}
31         </title>
32       </Helmet>
33       {
34         isTeacher ?
35         <>
36           <Classes
37             classes={classes.data}
38             classLoading={classLoading}
39             setModal={setModal}
40             refetch={refetch}
41           >
42             </Classes>
43             {
44               modal &&
45               <ClassModal
46                 refetch={refetch}
47                 modal={modal}
48                 setModal={setModal}
49               >
50                 </ClassModal>
51             }
52           </>
53           :
54           <>
55             {
56               userInfo.data?.role &&
57               <Enroll>
58
59                 </Enroll>
60             }
61           </>
62         </>
63       }
64
65       <div className="flex justify-center items-center">
66         {
67           !userInfo.data?.role && <Link className="btn btn-neutral" to="profile">Please Select Role First</Link>
68         }
69       </div>
70
71     </div>
72   );
73 };
74
75 export default MyHome;
```

```

1 import React, { useContext } from 'react';
2 import { MdOutlineAddToPhotos } from "react-icons/md";
3 import Loading from '../Shared/Loading/Loading';
4 import { AuthContext } from '../contexts/AuthProvider';
5 import ClassCard from './ClassCard';
6 const Classes = ({ classes, classloading, setModal, refetch }) => {
7   const { user } = useContext(AuthContext);
8   if (classloading) {
9     return <Loading></Loading>
10  };
11
12  // if(!loading){
13  //   return <Loading></Loading>
14  // };
15  return (
16    <div className="card-body grid xl:grid-cols-4 lg:grid-cols-3 md:grid-cols-4 sm:grid-cols-1 gap-3 gap-y-10">
17      <label htmlFor="class-modal" className="card justify-center items-center bg-base-100 hover:bg-base-300 border cursor-pointer" onClick={() => setModal(user?.email)}>
18        <MdOutlineAddToPhotos className="text-9xl"></MdOutlineAddToPhotos>
19      </label>
20      {
21        classes &&
22        classes?.map(c =>
23          <ClassCard
24            key={c._id}
25            c={c}
26            refetch={refetch}
27          >
28            </ClassCard>
29          )
30      </div>
31  );
32};
33
34export default Classes;

```

```

1 import React from 'react';
2 import { FaRegAddressCard } from "react-icons/fa6";
3 import { IoCodeSlash } from "react-icons/io5";
4 import { RiDeleteBin6Line } from "react-icons/ri";
5 import toast from 'react-hot-toast';
6 import { Link } from 'react-router-dom';
7 import Swal from 'swalalert2';
8 const ClassCard = ({ c, refetch }) => {
9   const { _id, name, subject, section, photoURL, classCode } = c;
10   const handelCopyCode = classCode => {
11     navigator.clipboard.writeText(classCode)
12     .then(() => {
13       // console.log('Code copied to clipboard:', code);
14       toast.success('Enrollment Key copied to clipboard!');
15     })
16     .catch(err => {
17       // console.error('Failed to copy code to clipboard:', err);
18       toast.error('Failed to copy Enrollment Key to clipboard. Please try again.');
```

```

19     });
20   };
21
22   const handelDeleteClass = id => {
23     Swal.fire({
24       title: "Are you sure?",
25       text: "You won't be able to revert this Class!",
26       icon: "warning",
27       showCancelButton: true,
28       confirmButtonColor: "#3085d6",
29       cancelButtonColor: "#d33",
30       confirmButtonText: "Delete Class"
31     }).then((result) => {
32       if (result.isConfirmed) {
33         fetch('http://localhost:5000/classes/${id}', {
34           method: "DELETE",
35           headers: {
36             authorization: `bearer ${localStorage.getItem("quickEdu-token")}`
37           }
38         })
39         .then(res => res.json())
40         .then(data => {
41           console.log(data)
42           if(data.data.deletedCount > 0){
43             refetch();
44             Swal.fire({
45               title: "Class Deleted!",
46               text: "Your Class has been deleted.",
47               icon: "success"
48             });
49           }
50           else{
51             Swal.fire({
52               icon: "error",
53               title: "Oops...",
54               text: "Something went wrong! Please Try Again"
55             });
56           }
57         })
58       });
59     });
60   };
61
62   return (
63     <div className="card bg-base-100 shadow-md rounded-none hover:shadow-xl">
64       <Link to={`/myhome/classinfo/${_id}`}><img className="object-fill h-[250px] w-full cursor-pointer" src={photoURL} alt={name + " Class Image"} /></Link>
65       <div className="card-body">
66         <h2 className="card-title">
67           {name}
68         </h2>
69         <p>{subject}</p>
70         <p>{section}</p>
71         <div className="card-actions justify-end gap-3">
72           <Link to={`/myhome/classinfo/${_id}`} className="text-3xl hover:bg-slate-400 tooltip" data-tip={"Class Info"}><FaRegAddressCard></FaRegAddressCard></Link>
73
74           <div onClick={() => handelCopyCode(classCode)} className="text-3xl hover:bg-slate-400 tooltip" data-tip={classCode}><IoCodeSlash></IoCodeSlash></div>
75
76           <div onClick={() => handelDeleteClass(id)} className="text-3xl hover:bg-slate-400 tooltip" data-tip={"Delete Class"}><RiDeleteBin6Line></RiDeleteBin6Line></div>
77         </div>
78       </div>
79     </div>
80   );
81};
82
83export default ClassCard;

```

```

1 import React from 'react';
2 import { useForm } from 'react-hook-form';
3 import toast from 'react-hot-toast';
4 import { format } from 'date-fns';
5
6 const ClassModal = ({ refetch, modal, setModal }) => {
7   const { register, formState: { errors }, handleSubmit } = useForm();
8   const currentDate = new Date();
9   const imageHostKey = process.env.REACT_APP_img88_key;
10
11   const handelCreateClass = (data, event) => {
12     const image = data.image[0];
13     const formData = new FormData();
14     formData.append('image', image);
15     fetch('https://api.imgbb.com/1/upload?key=${imageHostKey}',
16       {
17         method: 'POST',
18         body: formData
19       })
20     .then(res => res.json())
21     .then(imageData => {
22       const classInfo = {
23         name: data.name,
24         email: modal,
25         subject: data.subject,
26         section: data.section,
27         photoURL: imageData.data.url,
28         classCode: Math.random().toString(36).substr(2, 6).toUpperCase(),
29         createdAt: format(currentDate, "d/MM/yyyy HH:mm:ss")
30       }
31
32       fetch("http://localhost:5000/classes", {
33         method: "POST",
34         headers: {
35           "content-type": "application/json",
36           authorization: `bearer ${localStorage.getItem("quickEdu-token")}`
37         },
38         body: JSON.stringify(classInfo)
39       })
40       .then(res => res.json())
41       .then(data => {
42         // console.log(data.data);
43         if (data.data.insertedId) {
44           toast.success("Class Created!");
45           setModal(null);
46           refetch();
47         }
48       })
49     });
50   };
51   return (
52     <div>
53       {/* The button to open modal */}
54       <input type="checkbox" id="class-modal" className="modal-toggle" />
55       <div className="modal" role="dialog">
56         <div className="modal-box">
57           <label htmlFor="class-modal" className="btn btn-sm btn-circle btn-ghost absolute right-2 top-2 text-xl font-bold">X</label>
58           <form onSubmit={handleSubmit(handelCreateClass)} className="card-body">
59             <h1 className="text-xl">Create Class</h1>
60
61             <div className="form-control">
62               <label className="label">
63                 <span className="label-text">Class Name</span>
64               </label>
65               <input {...register("name",
66                 {
67                   required: { value: true, message: "Class Name is Required" }
68                 })}
69               type="text" placeholder="Enter Class Name" className="input input-bordered" />
70               <div className="label">
71                 {errors.name && <p className="text-red-600">{errors.name.message}</p>}
72               </div>
73             </div>
74             <div className="form-control">
75               <label className="label">
76                 <span className="label-text">Subject</span>
77               </label>
78               <input {...register("subject",
79                 {
80                   required: { value: true, message: "Subject is Required" }
81                 })}
82               type="text" placeholder="Enter Subject Name" className="input input-bordered" />
83               <div className="label">
84                 {errors.subject && <p className="text-red-600">{errors.subject.message}</p>}
85               </div>
86             </div>
87             <div className="form-control">
88               <label className="label">
89                 <span className="label-text">Section</span>
90               </label>
91               <input {...register("section",
92                 {
93                   required: { value: true, message: "Section is Required" }
94                 })}
95               type="text" placeholder="Enter Section Name" className="input input-bordered" />
96               <div className="label">
97                 {errors.section && <p className="text-red-600">{errors.section.message}</p>}
98               </div>
99             </div>
100
101             <div className="form-control">
102               <label className="label">
103                 <span className="label-text">Insert Class Image</span>
104               </label>
105               <input {...register("image",
106                 {
107                   validate: (value) =>
108                     [image/png, 'image/jpeg', 'image/jpg', 'image/gif'].includes(value[0]?.type) ||
109                     'Invalid Photo format. Only PNG, JPEG and JPG files are allowed.',
110                 })}
111               type="file" accept=".png, .jpg, .jpeg" className="file-input file-input-bordered w-full" />
112               <div className="label">
113                 {errors.image && <p className="text-red-600">{errors.image.message}</p>}
114               </div>
115             </div>
116             <div className="form-control">
117               <input type="submit" value="Create" className="btn btn-neutral hover:bg-slate-600 text-xl font-semibold" />
118             </div>
119           </form>
120         </div>
121       </div>
122     </div>
123   );
124 };
125
126 export default ClassModal;

```

```
1 app.post('/classes', async (req, res) => {
2   const data = req.body;
3   const newClass = postData(classesCollection, data);
4   newClass
5     .then(result => {
6       return res.send({
7         success: true,
8         message: "New Class Created",
9         data: result,
10      })
11    })
12    .catch(err => {
13      return res.send({
14        success: false,
15        message: err?.message
16      })
17    });
18});
```

Figure 10.2: Create Classroom Code Sample

Ai Question generator Sample code for the teachers

```
1 import React, { useState } from 'react';
2 import { useForm } from 'react-hook-form';
3 import { SiGooglebard } from 'react-icons/si';
4 import toast from 'react-hot-toast';
5 const QuizModal = ({ modal, setModal, refetch }) => {
6   const [resultLoading, setResultLoading] = useState(false);
7   const { register, formState: { errors }, handleSubmit } = useForm();
8   const currentUTC = new Date();
9   currentUTC.setHours(currentUTC.getHours() + 6);
10  const currentDate = currentUTC.toISOString().slice(0, 10); // For date input
11
12  // State for date and time
13  const [selectedDate, setSelectedDate] = useState(currentDate);
14
15  const handleDateChange = (e) => {
16    setSelectedDate(e.target.value);
17  };
18
19  const handelGenerateQuestion = (data, e) => {
20    setResultLoading(true);
21    // console.log(data)
22    data.classId = modal._id;
23    fetch("http://localhost:5000/classwork", {
24      method: "POST",
25      headers: {
26        'content-type': "application/json",
27        authorization: 'bearer ${localStorage.getItem("quickEdu-token")}'
28      },
29      body: JSON.stringify(data)
30    })
31      .then(res => res.json())
32      .then(data => {
33        console.log(data);
34        if (data.data.insertedId) {
35          setResultLoading(false);
36          toast.success("Question Generated!");
37          e.target.reset();
38          setModal(null);
39          refetch();
40        }
41      })
42      .catch(error => {
43        toast.error("Please Try again! Something went wrong...");
44        e.target.reset();
45        setModal(null);
46      })
47      // setModal(null)
48    };
49    return (
50      <div>
51        {
52          resultLoading ?
53            <div>
54              { /* The button to open modal */ }
55              <input type="checkbox" id="quiz-modal" className="modal-toggle" />
56              <div className="modal" role="dialog">
57                <div className="modal-box">
58                  <label htmlFor="quiz-modal" className="btn btn-sm btn-circle btn-ghost absolute right-2 top-2 text-xl font-bold">X</label>
59                  <div className="flex flex-col justify-center items-center sm:text-sm md:text-xl p-7">
60                    <span>Generating Your Quiz</span>
61                    <span>Please Wait</span>
62                    <div>
63                      <span className="loading loading-ball loading-xs"></span>
64                      <span className="loading loading-ball loading-sm text-info"></span>
65                      <span className="loading loading-ball loading-md"></span>
66                      <span className="loading loading-ball loading-lg text-info"></span>
67                    </div>
68                  </div>
69                </div>
70              :
71              <div>
72                { /* The button to open modal */ }
73                <input type="checkbox" id="quiz-modal" className="modal-toggle" />
74                <div className="modal" role="dialog">
75                  <div className="modal-box">
76                    <label htmlFor="quiz-modal" className="btn btn-sm btn-circle btn-ghost absolute right-2 top-2 text-xl font-bold border border-black">X</label>
77                    <form onSubmit={handleSubmit(handelGenerateQuestion)} className="card-body">
78                      <div className="flex items-center gap-1 text-2xl">
79                        <h1 className="font-semibold">Generate Quiz with Gemini </h1>
80                        <SiGooglebard className="text-sky-600"></SiGooglebard>
81                      </div>
82                      <div className="form-control">
83                        <label className="label">
84                          <span className="label-text font-semibold">Quiz Subject</span>
85                        </label>
86                        <input {...register("subject")} type="text" placeholder="Example: Math101" className="input input-bordered" defaultValue={modal.subject} />
87                        <div className="label">
88                          {errors.subject && <p className="text-red-600">{errors.subject.message}</p>}
89                        </div>
90                      </div>
91                      <div className="form-control">
92                        <label className="label">
93                          <span className="label-text font-semibold">Quiz No</span>
94                        </label>
95                        <input {...register("quizNo",
96                          {
97                            required: { value: true, message: "Quiz No is Required" }
98                          })} type="number" placeholder="Example: 1/2" className="input input-bordered" min={1} />
99                      </div>
100
```

```

1 <div className="form-control">
2     <label className="label">
3         <span className="label-text font-semibold">Quiz Date</span>
4     </label>
5     <div className="flex gap-1">
6         <input {...register("date",
7             {
8                 required: { value: true, message: "Quiz No is Required" }
9             })}
10            type="date"
11            className="input input-bordered w-full"
12            value={selectedDate}
13            min={currentDate}
14            onChange={handleDateChange}
15        />
16        <input
17            {...register("time",
18                {
19                    required: { value: true, message: "Time is Required" }
20                })}
21            type="time"
22            className="input input-bordered w-full"
23        />
24    </div>
25 </div>
26
27 <div className="form-control">
28     <label className="label">
29         <span className="label-text font-semibold">Quiz Time</span>
30     </label>
31     <div className="flex gap-1">
32         <input {...register("duration",
33             {
34                 required: { value: true, message: "duration is Required" }
35             })} type="number" placeholder="10 M / 1 H / 1.30 HM" className="input input-bordered" min={1} />
36         <select {...register("timeUnit",
37             {
38                 required: { value: true, message: "Class Name is Required" }
39             })} className="select select-bordered w-full max-w-xs">
40             <option value={"Minutes"}>Minutes</option>
41             <option value={"Hour"}>Hour</option>
42             <option value={"Hour+Minutes"}>Hour+Minutes</option>
43         </select>
44     </div>
45 </div>
46
47 <div className="form-control">
48     <label className="label">
49         <span className="label-text font-semibold">How Many Question</span>
50     </label>
51     <input {...register("totalQuestions",
52         {
53             required: { value: true, message: "Class Name is Required" }
54         })} type="number" placeholder="Example: 1/2" className="input input-bordered" min={1} />
55 </div>
56
57 <div className="form-control">
58     <label className="label">
59         <span className="label-text font-semibold">Quiz Level</span>
60     </label>
61     <div className="flex items-center">
62         <input {...register("level",
63             {
64                 required: { value: true, message: "Class Name is Required" }
65             })} type="radio" name="level" value={"Easy"} className="radio radio-info" />
66         <span className="mx-3">Easy</span>
67         <input {...register("level",
68             {
69                 required: { value: true, message: "Class Name is Required" }
70             })} type="radio" name="level" value={"Medium"} className="radio radio-warning" />
71         <span className="mx-3">Medium</span>
72         <input {...register("level",
73             {
74                 required: { value: true, message: "Class Name is Required" }
75             })} type="radio" name="level" value={"Hard"} className="radio radio-error" />
76         <span className="mx-3">Hard</span>
77     </div>
78 </div>
79
80 <div className="form-control">
81     <label className="label">
82         <span className="label-text font-semibold">Quiz Topic</span>
83     </label>
84     <textarea {...register("topic",
85         {
86             required: { value: true, message: "Class Name is Required" }
87         })} placeholder="Describe Properly Course Topic, Example: Algebra, OOP" className="textarea textarea-bordered textarea-lg w-full" ></textarea>
88 </div>
89
90 <input className="btn btn-neutral mt-3 text-2xl" type="submit" value="Generate Quiz" />
91 </form>
92 </div>
93 </div>
94 ]
95
96 </div>
97 );
98 };
99
100 export default QuizModal;

```

```

1 //Requiring gemini AI
2 const {
3     GoogleGenerativeAI,
4     HarmCategory,
5     HarmBlockThreshold,
6 } = require("@google/generative-ai");
7
8 //Selected gemini AI Model and configuration
9 // const MODEL_NAME = "gemini-1.0-pro"; // old model
10const MODEL_NAME = "gemini-1.5-flash"; // new model
11const API_KEY = process.env.GEMINI_KEY;
12
13const genAI = new GoogleGenerativeAI(API_KEY);
14const model = genAI.getGenerativeModel({ model: MODEL_NAME });
15
16const generationConfig = {
17     temperature: 0.9,
18     topK: 1,
19     topP: 1,
20     maxOutputTokens: 2048,
21};
22
23const safetySettings = [
24    {
25        category: HarmCategory.HARM_CATEGORY_HARASSMENT,
26        threshold: HarmBlockThreshold.BLOCK_MEDIUM_AND_ABOVE,
27    },
28    {
29        category: HarmCategory.HARM_CATEGORY_HATE_SPEECH,
30        threshold: HarmBlockThreshold.BLOCK_MEDIUM_AND_ABOVE,
31    },
32    {
33        category: HarmCategory.HARM_CATEGORY_SEXUALLY_EXPLICIT,
34        threshold: HarmBlockThreshold.BLOCK_MEDIUM_AND_ABOVE,
35    },
36    {
37        category: HarmCategory.HARM_CATEGORY_DANGEROUS_CONTENT,
38        threshold: HarmBlockThreshold.BLOCK_MEDIUM_AND_ABOVE,
39    },
40];
41
42const chat = model.startChat({
43    generationConfig,
44    safetySettings,
45    history: [
46    ],
47});

```

```

1 // ...
2 // ...
3 // ...
4 // ...
5 // ...
6 // ...
7 // ...
8 // ...
9 // ...
10 // ...
11 // ...
12 // ...
13 // ...
14 // ...
15 // ...
16 // ...
17 // ...
18 // ...
19 // ...
20 // ...
21 // ...
22 // ...
23 // ...
24 // ...
25 // ...
26 // ...
27 // ...
28 // ...
29 // ...
30 // ...
31 // ...
32 // ...
33 // ...
34 // ...
35 // ...
36 // ...
37 // ...
38 // ...
39 // ...
40 // ...
41 // ...
42 // ...
43 // ...
44 // ...
45 // ...
46 // ...
47 // ...
48 // ...
49 // ...
50 // ...
51 // ...
52 // ...
53 // ...
54 // ...
55 // ...
56 // ...
57 // ...
58 // ...
59 // ...
60 // ...
61 // ...
62 // ...
63 // ...
64 // ...
65 // ...
66 // ...
67 // ...
68 // ...
69 // ...
70 // ...
71 // ...
72 // ...
73 // ...
74 // ...
75 // ...
76 // ...
77 // ...
78 // ...
79 // ...
80 // ...
81 // ...
82 // ...
83 // ...
84 // ...
85 // ...
86 // ...
87 // ...
88 // ...
89 // ...
90 // ...
91 // ...
92 // ...
93 // ...
94 // ...
95 // ...
96 // ...
97 // ...
98 // ...
99 // ...
100 // ...

```

Figure 10.3: Ai Quiz Generator Code Sample

10.2 Possible Problem Breakdown

There are several possible challenges for developing and deploying Quick Edu Live. It is important to identify these issues and solve them early for the predictable deployment:

AI Technologies Used

- ❖ Problem: Seamless Integration of AI Technologies Such as Gemini-1 5-Flash API and Tesseract.js can be complex.
- ✓ Solution: Undertake Research and have prototyping with the guidance of AI experts. Documentation and extensive plans of testing

Security and User Authentication

- ❖ Problem: The security of user authentication and sensitive data
- ✓ Solution: Firebase Authentication.

User Experience and Interface Design

- ❖ Problem: Provide an Interactive Interface
- ✓ Solution: Perform user research and testing, design with React JS and Tailwind CSS, adhere to accessibility standards for ease of use.

Data Management and Privacy

- ❖ Problem: Securely managing student data.
- ✓ Solution: MongoDB Data protection with JWT.

Performance Optimization

- ❖ Problem: To guarantee a quick response time and high-performance load times.
- ✓ Solution: Improve the code, try to access as quickly or cache if possible and measure in the metrics any question.

10.3 Prioritization While Developing

Phase 1: Core Features

- User Authentication: Build using Firebase, Multi Role (Teacher, Student) management.
- Class Management: Teachers make and assign classes. students join class using code.
- Quiz/Assignment Generation: Use Gemini-1. 5- AI Generated Content Flash API.

Phase 2: Data Management

- Security: Use JWT.

Phase 3:

- User Interface and Experience Design: Build a user-friendly experience with React JS and Tailwind CSS to polish the design using feedback from our users.

Phase 4: Communication

- Real-Time Communication: Allow students to interact with teachers and make announcements.

Phase 5: Improvement

- Feedback: Keep listening to the voice of your users.
- Testing: Standard unit, integration and UAT testing.

Chapter 11: Testing

11.1 Test Plan Acceptance

Quick Edu Live application comes a proper test plan with all targeted requirements and is in a working condition. Test plan acceptance takes place when all necessary stakeholders vest detailed test strategies, methodologies and schedules, including the project managers and developers' team as well as quality assurance alone or together to approve a test.

Testing Approach

- Unit Testing: Each individual component is tested for its correctness.
- Module Testing: Each module is tested in isolation.
- Integration Testing: Coordinate modules or services to work together
- System Testing: The distributed system is tested in its entirety
- Acceptance Testing: Verifies the system complies to acceptance criteria defined by end user.

11.2 Test Case

Example of a test case for the user sign up/ login functionality.

- Test Case Name: Unit Test
- Title: Verify User sign up / Login
- Explanation: Ensure that a sign-up user can log in with valid credentials.

Test Steps:

- Navigation to the sign-up page.
- Enter a authorize email and password.
- Click the "Sign up" button.
- Expected Result: The user should be navigating to the system home screen.
- Actual Result: Show the system home screen

Unit Testing

Unit testing verifies the working of a particular piece of module only. Below are 2-unit tests:

1. User Registration
 - Verify that a new user can sign up with valid credentials.
2. AI Quiz Generator

- Verify that the AI quiz generator function returns a quiz with the correct parameters.

Module Testing

Module testing involves testing of individual applications modules separately to check whether they are working as per the specified requirement. Some examples of module tests.

1. Class Management

- Verify that teachers can create and delete classes.

2. Announcement Posting

- Verify that teachers can post text announcements.

Integration Testing

Integration tests make sure that the various modules interact with each other in the proper way.

The following are examples of integration tests:

1. User Registration and Class Enrollment

- Verify that a new user can register and then enroll in a class.

2. AI Quiz Generating, live quiz and Result Checking

- Verify that a teacher can generate a quiz and student can participant in that quiz and then finally can check result

Acceptance Testing

Acceptance testing validates the system against user requirements and ensures it meets the acceptance criteria set by the end users.

Acceptance Test Steps

1. Register a new teacher and student.
2. Log in as the teacher and create a new class.
3. Share the class code with the student.
4. Log in as the student and enroll in the class using the class code.
5. Create a quiz using the AI Quiz Generator as the teacher.
6. Log in as the student and take the quiz.
7. Verify that the quiz is automatically graded and feedback is provided.
8. Post an announcement as the teacher and ensure the student can view and comment on it.
9. Remove the student from the class as the teacher.

11.3 Unit Testing

Test Case Name	Unit Test 1		
Test Class	User Registration		
Test Description	Verify that a new user can register with valid credentials		
Data Source	Test Steps	Expected Result	Actual Result
User (Teacher and Student)	1. Input User Details (User credential) 2. Click Signup Button	User successfully register and navigated to classroom	User successfully register and navigated to classroom

Figure 11.1: Unit Test 1

Test Case Name	Unit Test 2		
Test Class	AI Quiz Generator		
Test Description	Verify that the AI quiz generator function returns a quiz with the correct parameters.		
Data Source	Test Steps	Expected Result	Actual Result
Teacher	1. Input: Quiz Subject, Quiz No, Quiz Date, Quiz Time, How Many Questions, Quiz Level (easy, medium, hard), Quiz Topic 2. Click Generate Quiz Button	Teacher will successfully generate quizzes.	Teacher will successfully generate quizzes.

Figure 11.2: Unit Test 2

11.4 Module Testing

Test Case Name	Module Test 1		
Test Class	Classroom Management		
Test Description	Verify that teachers can create and delete classes		
Data Source	Test Steps	Expected Result	Actual Result
Teacher	1. Input: Class name, Subject Name, Section, Insert Class Image 2. Click Create Button	Teacher will successfully create class.	Teacher will successfully create class.

Figure 11.3: Module Test 1

Test Case Name	Module Test 2		
Test Class	Announcement Posting		
Test Description	Verify that teachers can post text announcements		
Data Source	Test Steps	Expected Result	Actual Result
Teacher	1. Input: announcement 2. Click Post Button	Teacher will successfully post an announcement.	Teacher will successfully post an announcement.

Figure 11.4: Module Test 2

11.5 Integration Testing

Test Case Name	Integration Test 1		
Test Class	User Registration and Class Enrollment		
Test Description	Verify that a new user can register and then enroll in a class		
Data Source	Test Steps	Expected Result	Actual Result
Student	1. Input User Details (User credential) 2. Click Signup Button 3. Input Enrollment Key 4. Click Enroll Button	User successfully registered and enrolled in a classroom.	User successfully registered and enrolled in a classroom.

Figure 11.5: Integration Test 1

Test Case Name	Integration Test 2		
Test Class	AI Quiz Generating, live quiz and Result Checking		
Test Description	Verify that a teacher can generate a quiz and that student can participant in that quiz and finally can check result		
Data Source	Test Steps	Expected Result	Actual Result
Users	1. Teacher generating Quiz 2. Student participating in quiz 3. Student Getting quiz result	Quiz generated successfully, student was able to participated in quiz and viewing their result	Quiz generated successfully, student was able to participated in quiz and viewing their result

Figure 11.6: Integration Test 2

11.6 Acceptance Testing

Test Case Name	Acceptance Test		
Test Class	Overall Test of users		
Test Description	Verify that a user can complete the entire process from registration to checking quiz result.		
Data Source	Test Steps	Expected Result	Actual Result
Users (Teacher and Student)	1. Register a new teacher and student. 2. Log in as the teacher and create a new class. 3. Share the class code with the student. 4. Log in as the student and enroll in the class using the class code. 5. Create a quiz using the AI Quiz Generator as the teacher. 6. Log in as the student and take the quiz. 7. Verify that the quiz is automatically graded and feedback is provided. 8. Post an announcement as the teacher and ensure the student can view and comment on it. 9. Remove the student from the class as the teacher.	Entire process should be successful without error	Entire process should be successful without error.

Figure 11.7: Acceptance Test

Quick Edu Live

Signup

Login

Signup

Name

Mehedi Hasan

Institution

MBSTU

Country

Bangladesh

Date of Birth

01/01/1994

Role

☒ Teacher
 ☐ Student

Email

mehedihasan@gmail.com

Password

Confirm Password

Insert Your Photo

CHOOSE FILE

photo.png

Signup

Google

Already Have Account! [Login](#)

Quick Edu Live

SERVICES

AI Classroom

AI Quiz Generator

AI Assignment Generator

AI Paper Checker

COMPANY

About us

Contact

LEGAL

Terms of use

Quick Edu Live

Signup

Login

Signup

Name

Md. Minhajur Rahman

Institution

DIU

Country

Bangladesh

Date of Birth

13/11/1999

Role

☐ Teacher
 ☒ Student

Email

minhajur16-477@diu.edu.bd

Password

Confirm Password

Insert Your Photo

CHOOSE FILE

Md. Minhajur Rahman.png

Signup

Google

Already Have Account! [Login](#)

Quick Edu Live

SERVICES

AI Classroom

AI Quiz Generator

AI Assignment Generator

AI Paper Checker

COMPANY

About us

Contact

LEGAL

Terms of use

Figure 11.8: Registered a new teacher and student

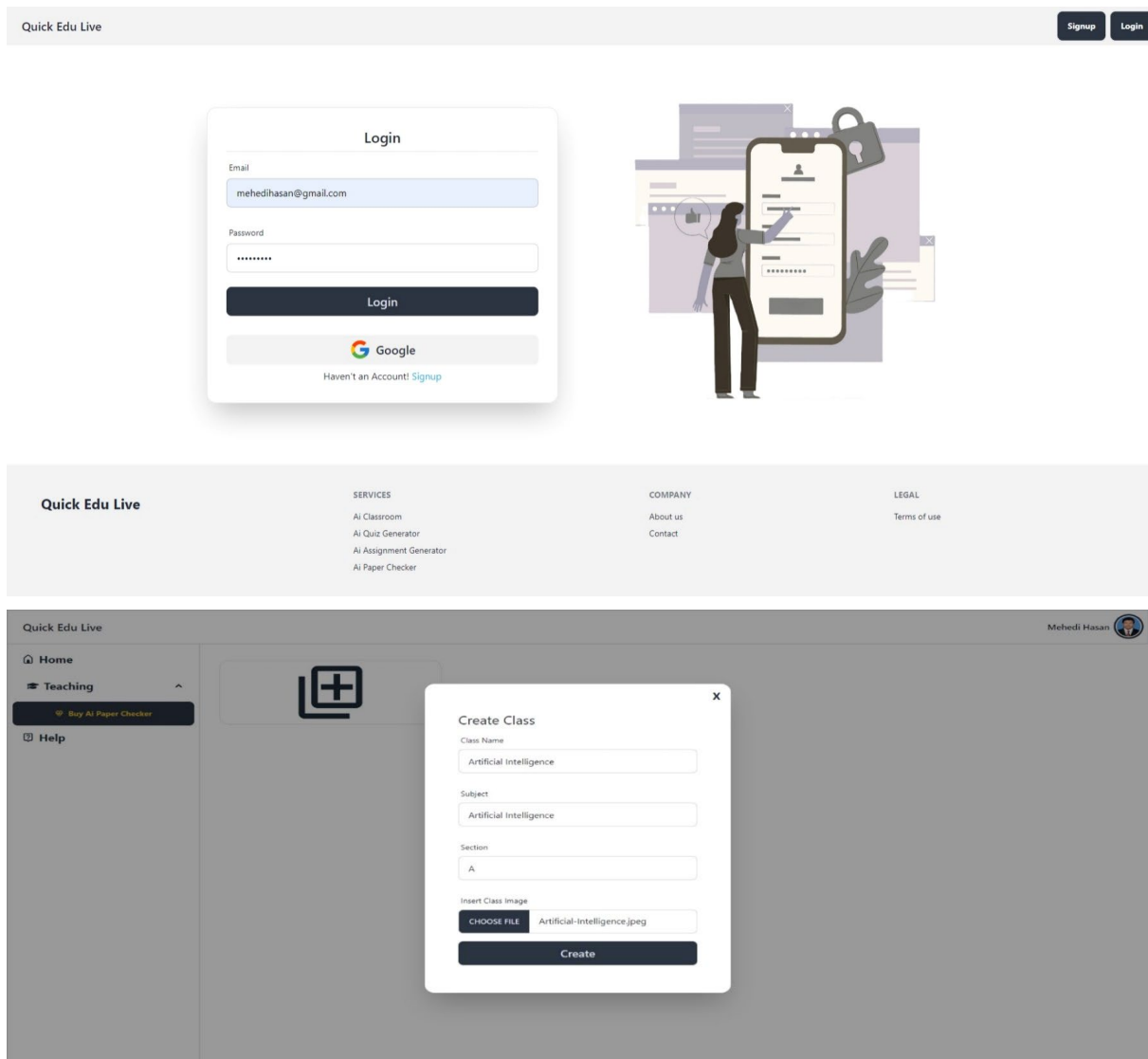


Figure 11.9: Logging as a teacher and creating a new class

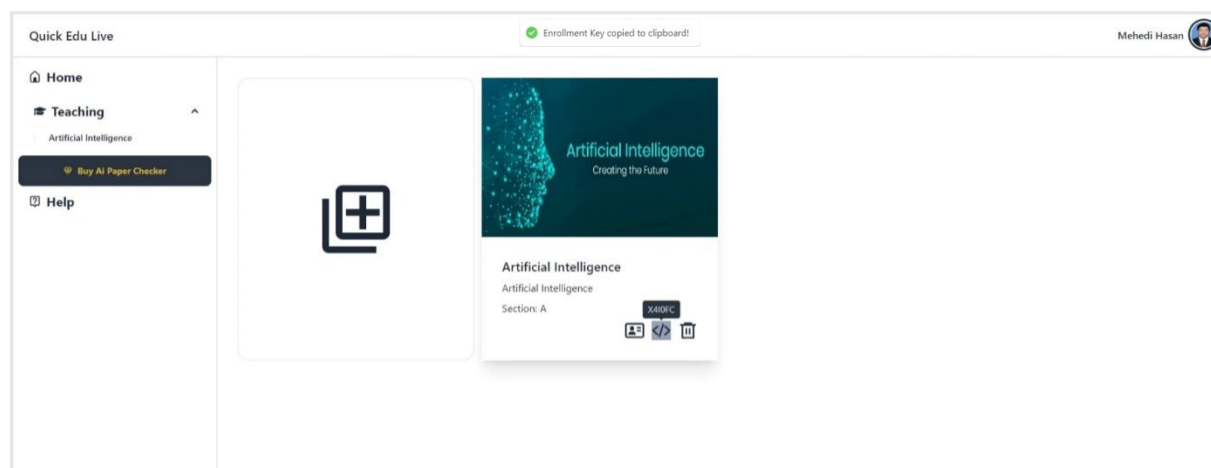


Figure 11.10: Copying classroom code to share with students

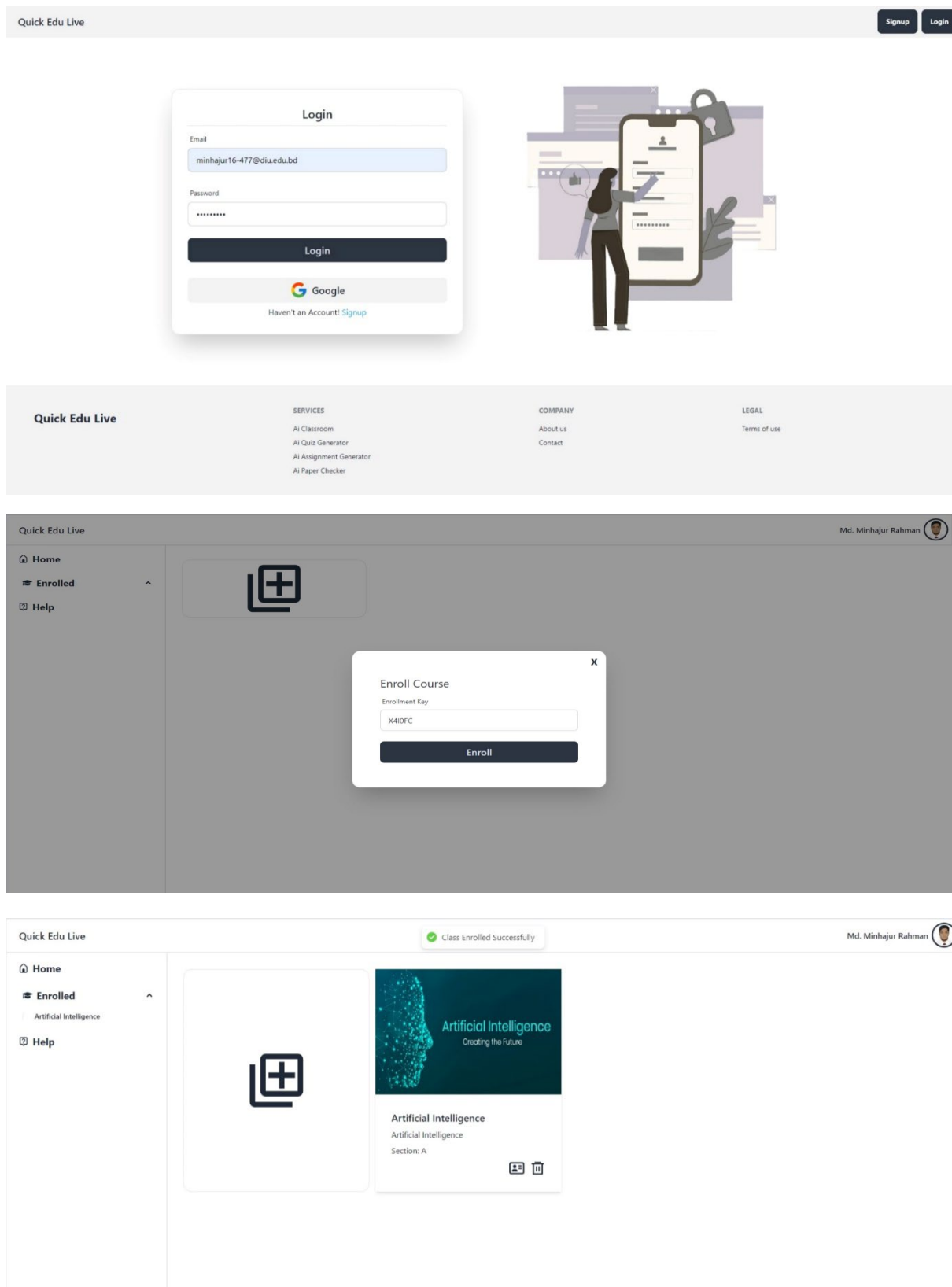


Figure 11.11: Logging as a student and enrolling in the class using the class code

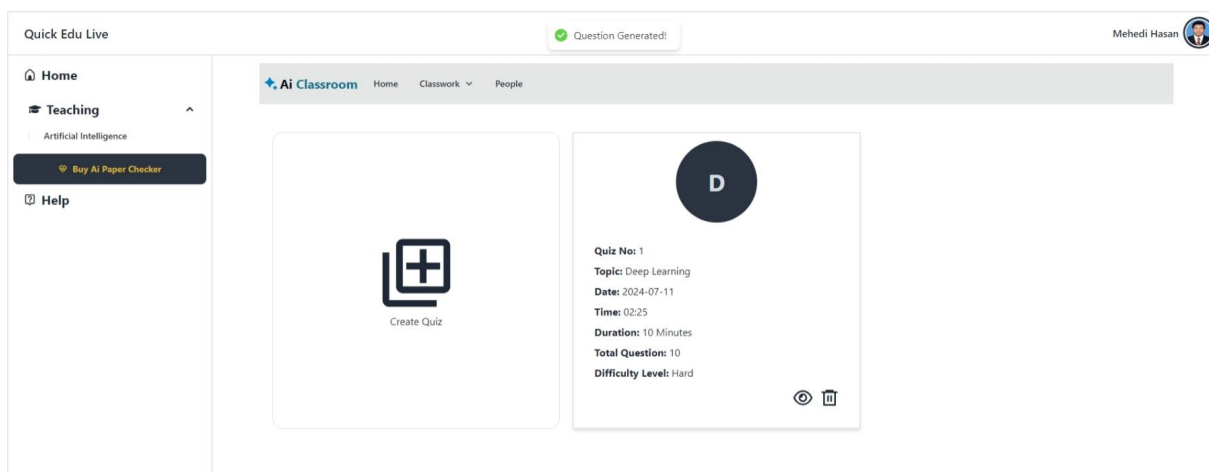
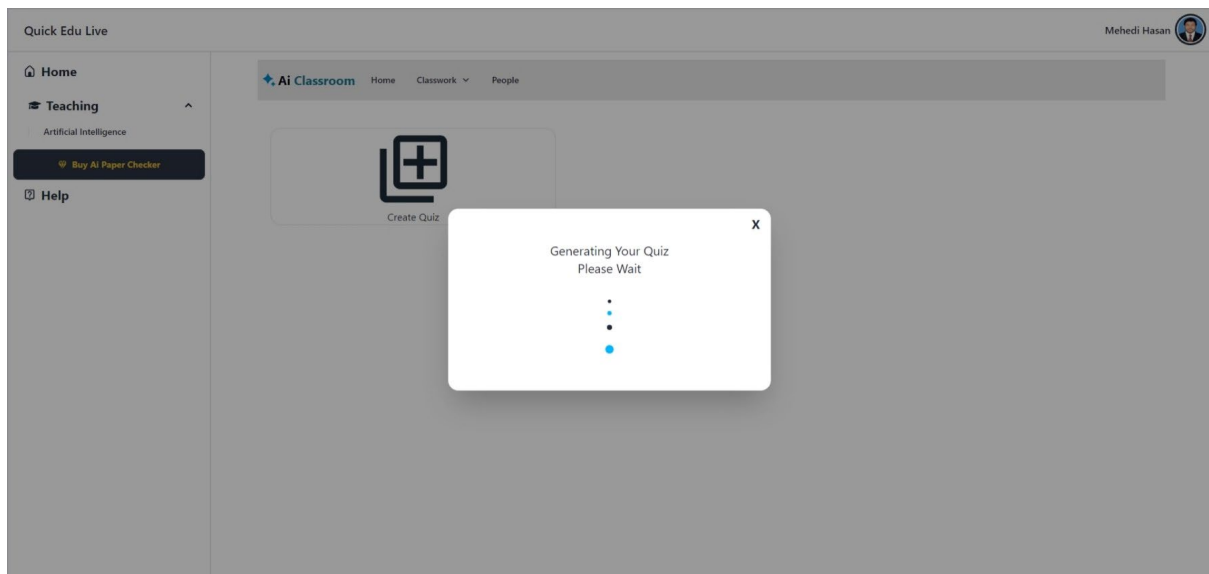
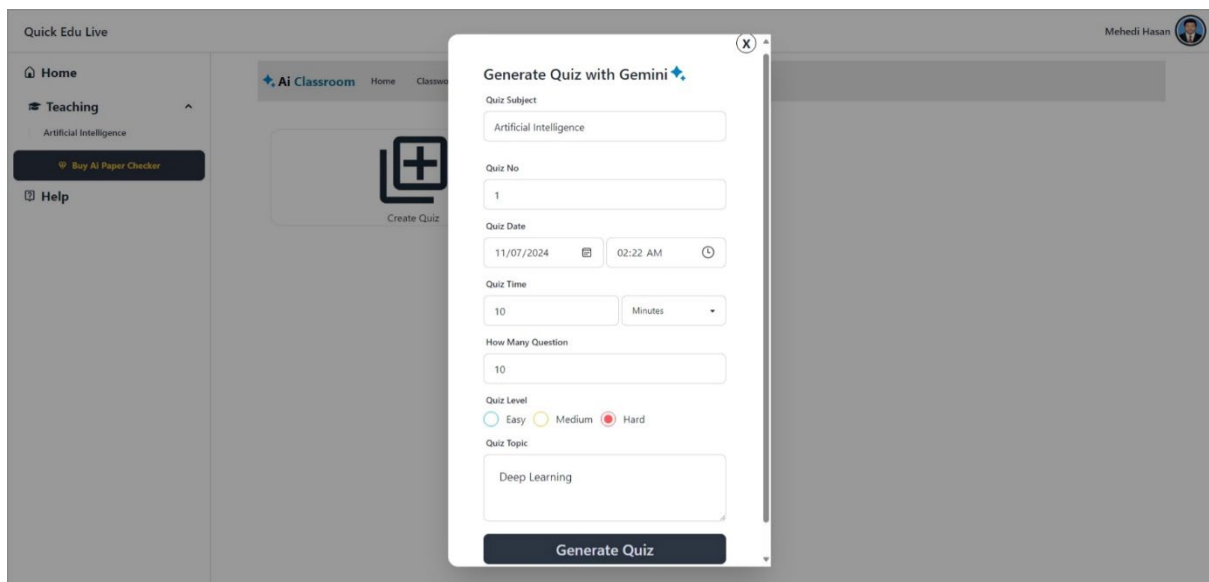
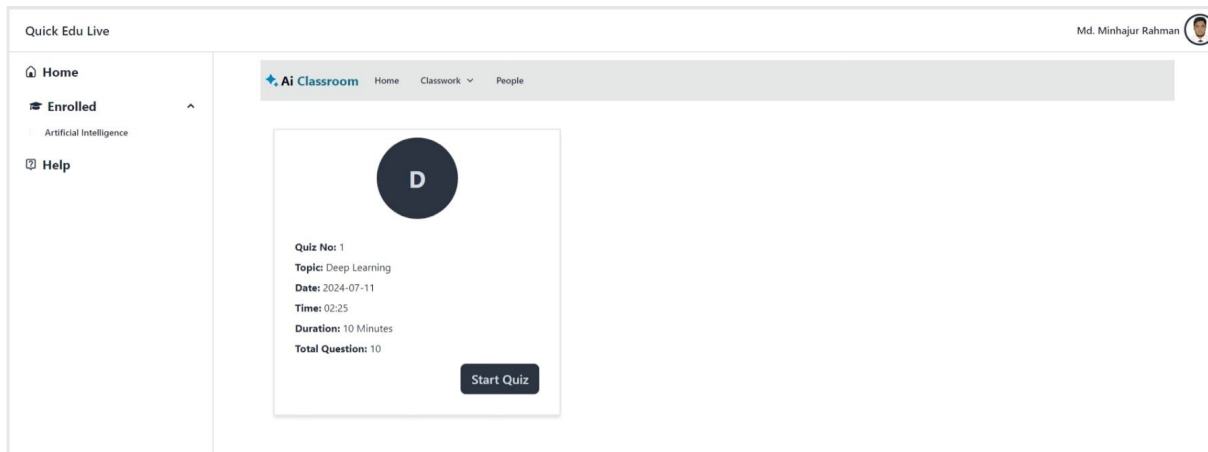


Figure 11.12: Creating a quiz using the AI Quiz Generator as a teacher.



Topic: Deep Learning
Total Questions: 10
Date: 2024-07-11
Duration: 10 Minutes
Marks: 10

Best of luck Md. Minhajur Rahman

Time Left: 9:59

Q.1 Which of the following is NOT a type of neural network architecture?

☐ a) Convolutional Neural Network (CNN) ☐ b) Recurrent Neural Network (RNN)

☐ c) Generative Adversarial Network (GAN) ☐ d) Linear Regression

Q.2 What is the primary function of backpropagation in deep learning?

☐ a) Building the neural network architecture ☐ b) Training the model by adjusting weights

☐ c) Selecting the optimal activation function ☐ d) Evaluating the performance of the model

Q.3 What is the purpose of an activation function in a neural network?

☐ a) To introduce non-linearity into the model ☐ b) To normalize the input data

☐ c) To determine the number of hidden layers ☐ d) To calculate the loss function

Q.4 What is overfitting in the context of deep learning?

☐ a) When the model performs well on the training data but poorly on unseen data ☐ b) When the model performs poorly on both training and testing data

☐ c) When the model is too complex for the given data ☐ d) When the model is too simple for the given data

Q.5 Which of the following is a common technique to prevent overfitting in deep learning?

☐ a) Dropout ☐ b) Early stopping

☐ c) Data augmentation ☐ d) All of the above

Q.6 What does the loss function in a deep learning model measure?

☐ a) The accuracy of the model ☐ b) The complexity of the model

☐ c) The error between the predicted output and the actual output ☐ d) The number of training epochs

Q.7 How does gradient descent work in the context of deep learning?

☐ a) It randomly assigns weights to the network ☐ b) It iteratively updates the weights to minimize the loss function

☐ c) It determines the optimal number of hidden layers ☐ d) It calculates the activation function

Q.8 What is the primary benefit of using batch normalization in deep learning?

☐ a) It reduces the training time ☐ b) It improves the stability of the training process

☐ c) It helps to prevent overfitting ☐ d) All of the above

Q.9 What is transfer learning in deep learning?

☐ a) Using a pre-trained model on a different dataset ☐ b) Training a model from scratch

☐ c) Using different activation functions ☐ d) Increasing the number of hidden layers

Q.10 What type of neural network is particularly well-suited for processing sequential data, such as text or time series?

☐ a) Convolutional Neural Network (CNN) ☐ b) Recurrent Neural Network (RNN)

☐ c) Generative Adversarial Network (GAN) ☐ d) Autoencoder

Submit

Figure 11.13: Logging as a student and participating in the quiz.

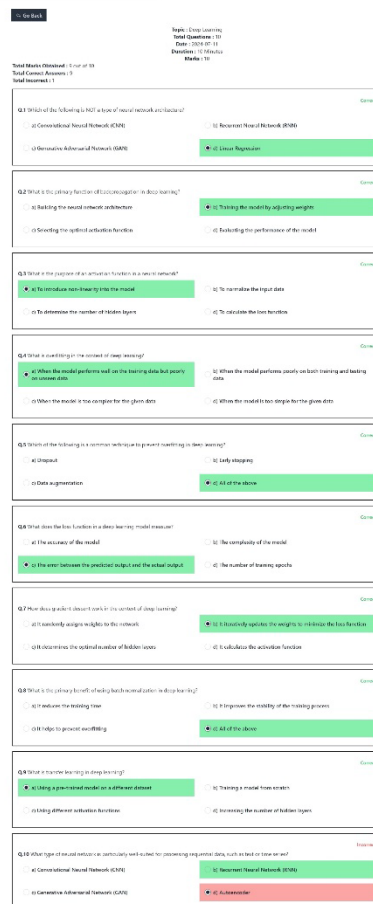
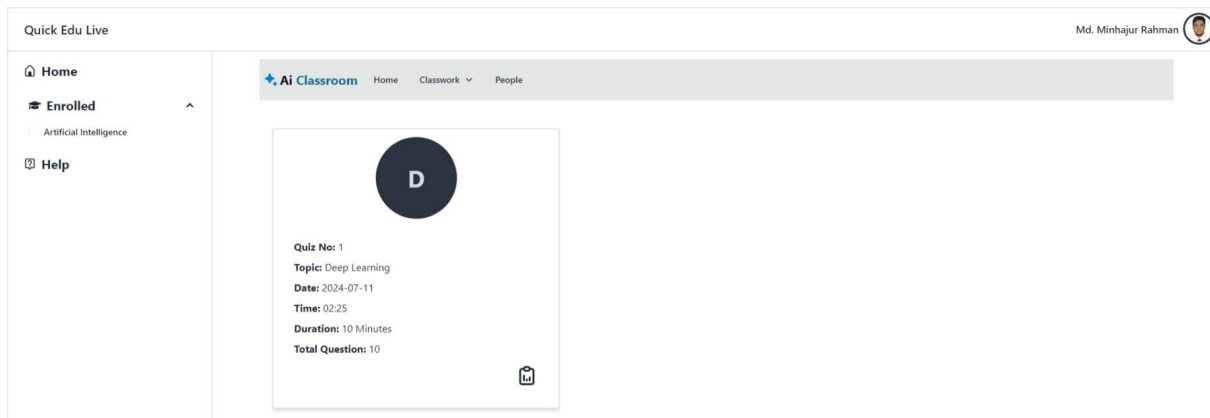
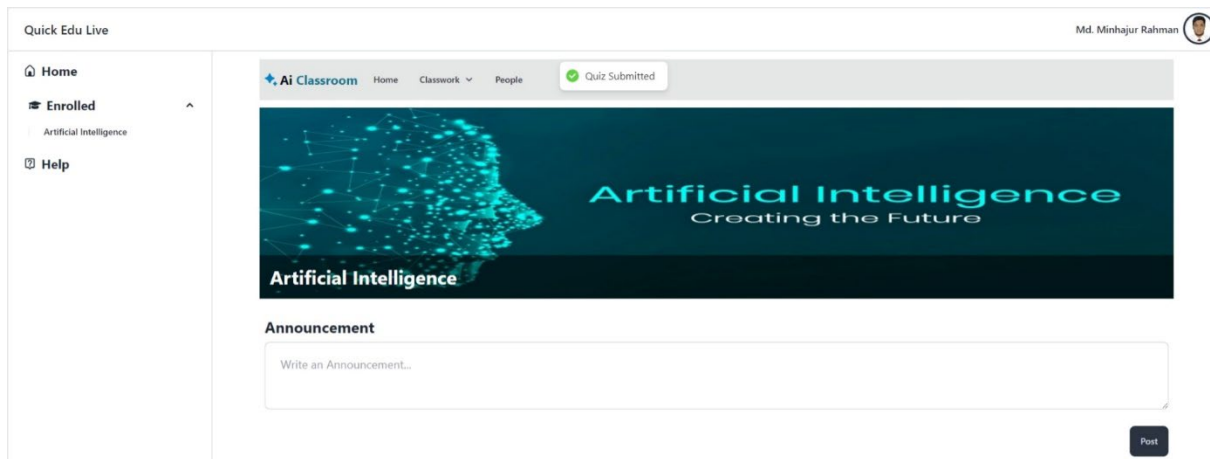


Figure 11.14: Verifying that the quiz is automatically graded and feedback is provided

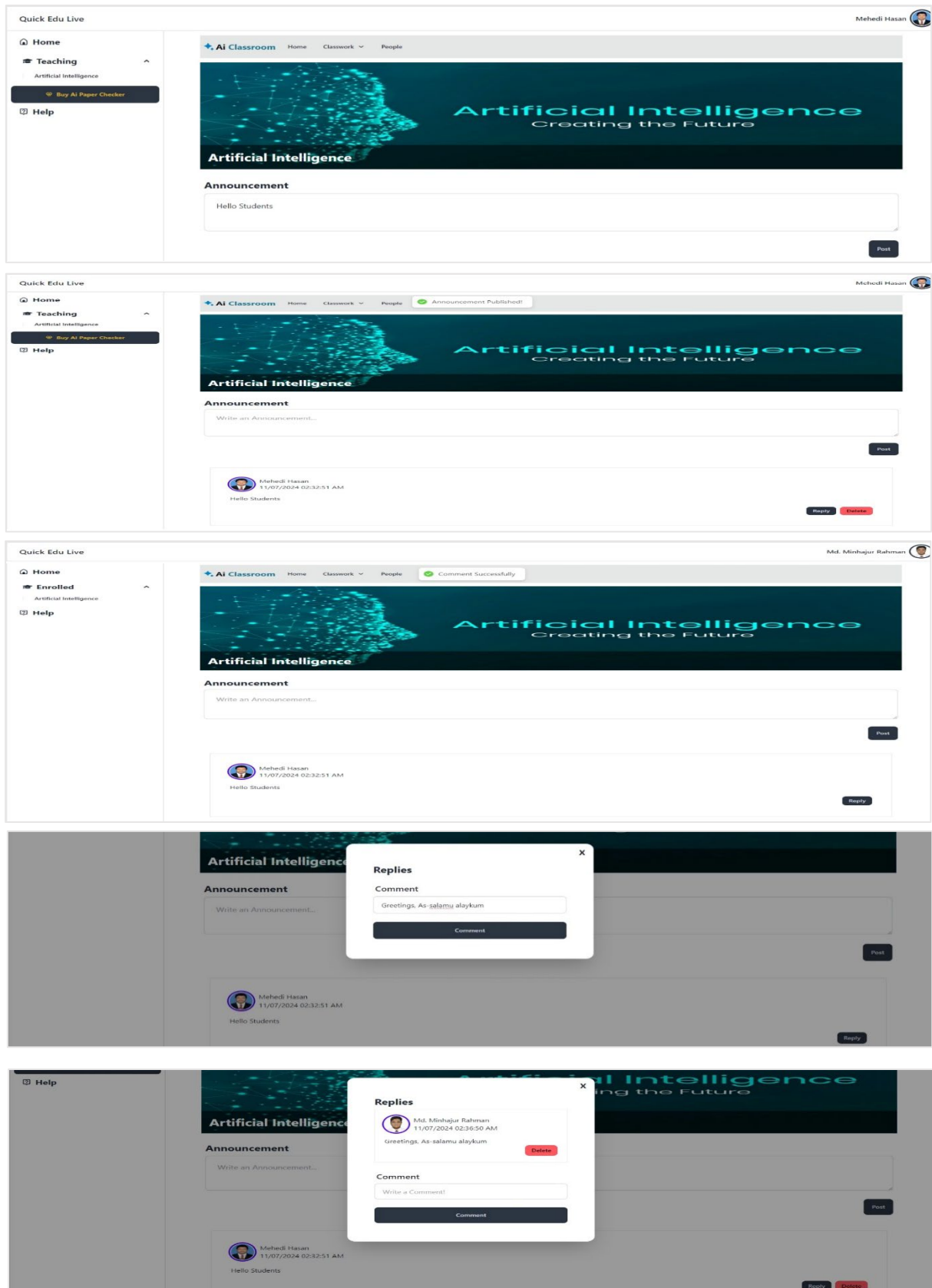


Figure 11.15: Posting an announcement as a teacher and ensure as a student can interact

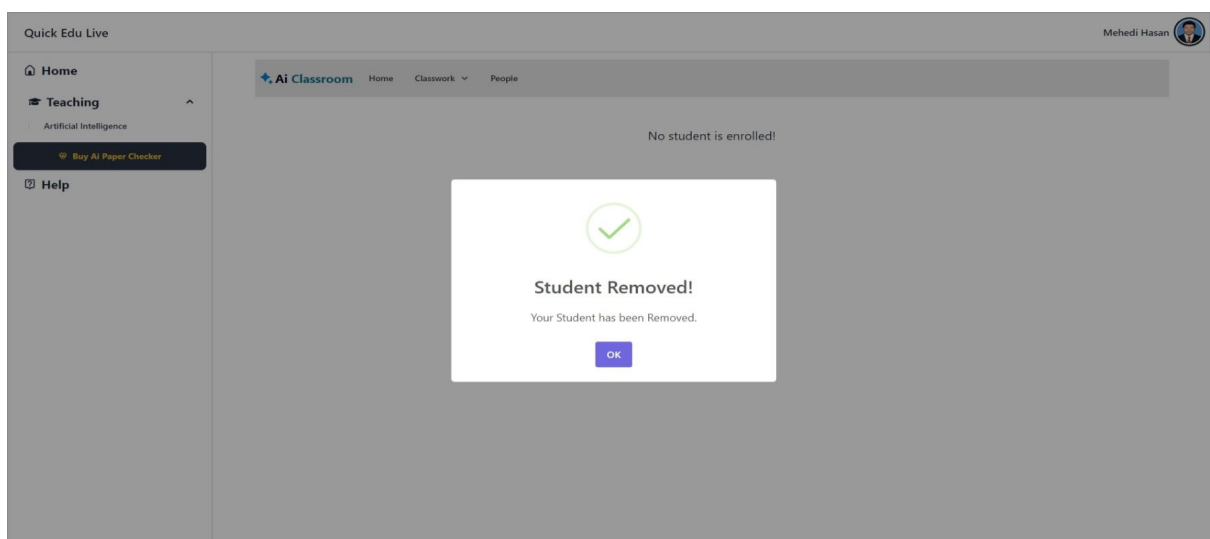
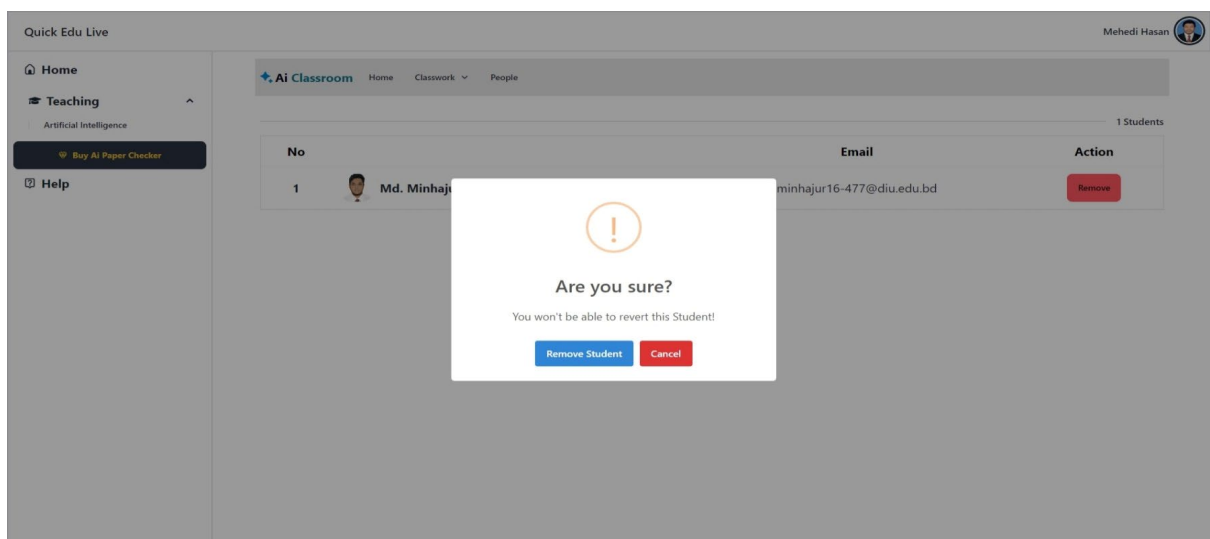
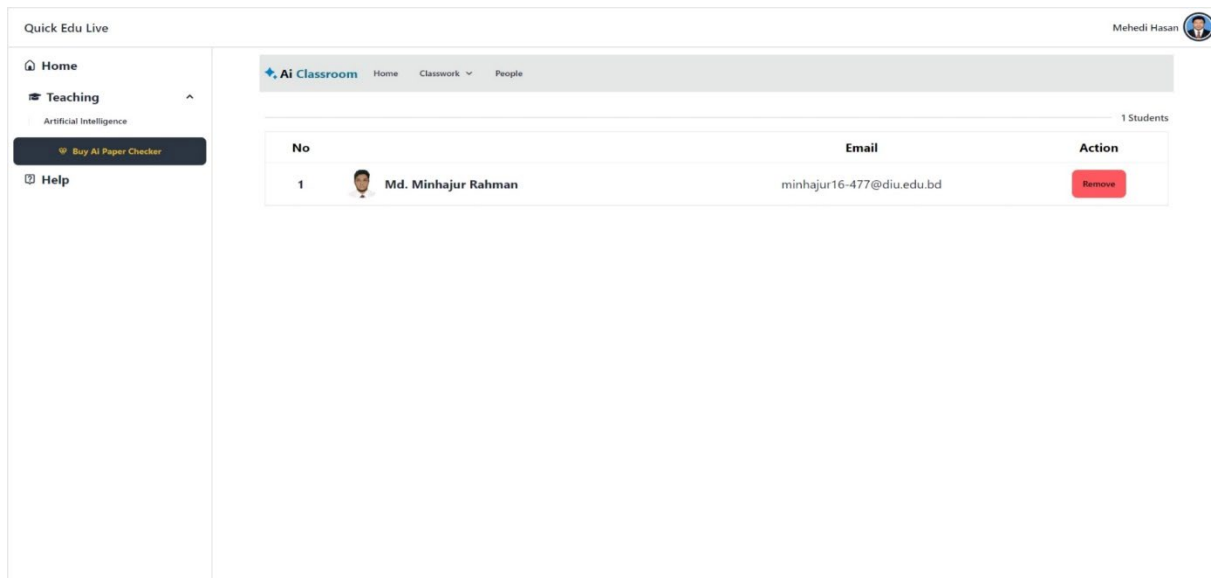


Figure 11.16: Removing a student from the class as a teacher

Chapter 12: Implementation

12.1 Training

Training is crucial to ensure that users are used to with the Quick Edu Live system. Once fully functional, the team must ensure that users are effectively prepared to handle processes through the system. A chart that defines the training actions is given below.

No	Users	Training Description	Duration	Comments
1	Teacher	How to create and manage classrooms, how they can generate quizzes / assignments, post announcements using AI.	3 Hour	The procedure is that the teachers are learning so efficiently.
2	Student	How to Joining classes, reading announcements and how to take their quizzes as well assignments here.	2 Hour	Their roles are completely understood by the students.

Table 12.1: Quick Edu Live User Training

12.2 Implementation Scheme

Big Bang

It refers to turning off the existing system and switching on immediately the new one. This is quicker than other methods, since a new implementation must be used right after being tested. On the other hand, this method is less reliable as data can fail to be transferred from the old system due to instability in which designation may cause a loss of data.

12.3 Scaling

Quick Edu Live is made as an academic project, so there are no plans to scale the application. This makes Scaling plans will come into play only if the project is successful and there are really users who need it.

12.4 Load Balancing

Load balancing helps to provide optimum system performance over distributing user impact regularly. I use multiple types of users who have many daily hits this demands us to create a solid load balancing strategy. I execute this strategy to avoid that the system may crash with many users.

Chapter 13: Critical Appraisal and Evaluation

13.1 Objective that could be met

- ❖ Success rate against each objective of Quick Edu Live!

The success ratio of Quick Edu Live can be determined by contrasting the achieved aims with the primary project objectives. The main needs were to develop an AI assisted classroom management platform with features such as class creation, announcements and user management, which functions included like AI Quiz Generator, AI Assignment Generator and Paper Checker. The solution is to measure your success by examining user feedback, system performance metrics and the satisfaction of functional requirements.

- ❖ How much better could have been done Quick Edu Live?

Although Quick Edu Live has accomplished its main goals, things could be better. Some improvements might consist of better UI, more AI features and being able to be integrated with other educational tools.

- ❖ Why it could not be done in Quick Edu Live?

Some improvements remain uncompleted simply for the lack of time and technology. For example, developing a production ready infrastructure can be out of the question for an academic project. Integrating more advanced AI capabilities similarly involve executing complex algorithms that cannot be achieved in the project with available resources.

- ❖ Which objectives have been missed in Quick Edu Live?

Might have missed out on few objectives such as advanced user analytics & personalized learning pathways. These features would take longer to build and require a degree of data science, machine learning expertise.

- ❖ Why these objectives have missed in Quick Edu Live?

Mostly resources and technical expertise are the reasons that do not allow achieving these objectives. Probably the core functionality was the focus and the secondary may be features took a second place.

- ❖ What could have been done to complete those objectives in Quick Edu Live?

A more detailed project plan with specific milestones could have been useful to complete the missed objectives.

❖ How better are the features of the solution in Quick Edu Live?

Quick Edu Live comes with many features, which are designed especially for teachers and students to get the best use of futuristic AI driven tools be it Classroom Management, Video Quizzing or Game-Based Assessment. These capabilities greatly improve efficiency and less the administrative load on teachers, improving education overall.

❖ Which features could not be touched in in Quick Edu Live?

Other higher-level capabilities, including real-time collaboration tools, robust user analytics and individualized learning suggestions were missing. They depend on complex algorithms and a large amount of user data to function.

❖ Why these features could not be touched in Quick Edu Live?

Time and resource constraints limited the development of these advanced features. Even though the additional features were necessary, implementing something like collaborating in real-time was a considerable amount of effort on both the backend and front-end which may not have been feasible due to project scope.

❖ What could be done to touch those features in Quick Edu Live?

Thinking even further, the next iterations of Quick Edu Live will be powered to build more extensive planning and resource allocation.

13.2 Objectives totally not touched in Quick Edu Live!

❖ Why it could not be touched

Some targets were untouched by a shortage of resources large scalability solution & Comprehensive accessibility features This would have required significant additional infrastructure with more developed technical expertise, which at the time were not available.

❖ What could have been done in Quick Edu Live?

Develop in phases, focusing on core functionality scalability and accessibility. Form strategic partnerships with ed-tech companies and receive grants for developmental purposes.

❖ Including software and documentation

Quick Edu Live remains reliable and user-friendly with periodic updates, complete instructions as well as consumer feedback. Quality is integral to all that I deliver.

Chapter 14: Lessons Learned

14.1 Pre-Project Review – Closing

Quick Edu Live done with a clear plan for building an AI Classroom on Quick Edu Live. I started from a project proposal, defended the idea and the developed & documented system. So, the core reason behind this is to bring AI tools in order to make regular management and learning improvement, which would ultimately benefit us being learner making fulfilling more education scenarios.

14.2 What Have I Learned?

Creating Quick Edu Live has been a fabulous experience & learning in terms of managing my projects, developing technically and interacting with users. Key lessons learned include:

- ✓ Comprehensive Planning: Necessary to have a proper base and plan for the project.
- ✓ Flexibility and Adaptability: Important for adapting to changes and the unexpected.
- ✓ Iterative Development: A path of continuous improvement could then lead to delivering a final product with the functionality that user wants, thanks to Agile ways.
- ✓ Risk Management: Proactive identification of the risks and updating to regular risk management plan update can help avoiding problems.
- ✓ User-Centric Design: Stress on user needs and feedback results in satisfaction so more adoption.

14.3 What Problems I Have Faced

Quick Edu Live had many challenges while development that needed to be solved properly and managed. Key problems faced included:

Requirement Ambiguities

- ❖ Problem: Since a few requirements were ambiguous initially.
- ✓ Solution: I did research on google and YouTube for requirements clarity.

Resource Constraints

- ❖ Problem: Project was affected due to limited time.
- ✓ Solution: Prioritized the must-have features and took a phased approach.

Technical Challenges

- ❖ Problem: Issues integrating advanced AI.

- ✓ Solution: Lear about advanced AI integration then read the documentation.

Performance Optimization

- ❖ Problem: It had to be able to function during high volume activities on the site.
- ✓ Solution: Extensive Performance Testing and Optimization Optimizations like caching were deployed to increase performance.

11.4 What Solutions Occurred

I solved the challenges of developing Quick Edu Live by introducing these key solutions:

Better Planning and Requirement Gathering: Held additional sessions to clarify & finalize the list of requirements.

Phased Development Approach: Prioritized characteristics based on criticality according to the business and timeline constraints

Training & Support: A Focus on the User Developed robust training programs and support mechanisms to help users convert into the new platform with a focus on continuous improvement.

The problems in developing Quick Edu Live were addressed by introducing: **Optimizing Techniques for Performance:** Extensively tested to find bottlenecks and mitigate the same using caching, load balancing etc. for better performance as well efficiency

Regular Communication and Feedback Loops: Continuously increased the quality of our platform by involving users and providing feedback.

Robust Risk Management: Created and maintained a risk management plan to anticipate and address risks early in the project process.

Chapter 15: Conclusion

15.1 Summary of the Project

Quick Edu Live remains revolutionary platform in the field of online education to facilitate teachers and students. The platform incorporates AI-powered features to create quizzes and assignments, grade automatically and store all the data in one place. A universal remedy to many problems that are typical for traditional educational systems. User authentication, class management system AI quiz & assignment generator Quiz grading automation Real-time feedback to alleviate administrative burden timeously the project underwent development, testing and planning with emphasis on user centered design and iterative-based changes per stakeholder feedback. I tested from feature level to performance, security, accessibility and usability levels of the app.

15.2 Goal of the Project

The motive of Quick Edu live is to develop AI (Artificial Intelligence) based learning and teaching portal. The objectives include:

- ❖ Reducing Administrative Workload: Allow teachers to automate quiz creation, grading and student records management.
- ❖ Providing Timely Feedback: Provide auto-grading and instant feedback to further support the learning process for students.
- ❖ Ensuring Data Security: Set up a full-proof mechanism to store and handle the student data where each record is secure, private using this system complies with all data protection laws.
- ❖ Enhancing User Experience: Build web-based user interface for teachers and students in order to provide an easy-to-understand on any devices with all popular browsers.

15.3 Success of the Project

Quick Edu Live project is a successful achievement and produce an educational platform. Key achievements include:

- ✓ AI Tools: Developed artificial intelligence-enabled quiz and assignment generators to save time and effort.
- ✓ Automatic Grading: Built an AI Paper Checker that grades and provides near instant feedback.
- ✓ Secure Data Management: Implemented data encryption and secure user authentication

- ✓ User-Friendly Interface: Created an intuitive UI with the help of modern frontend frameworks

15.4 What I Have Done in the Documentation

Quick Edu Live documents contains everything from initial project stages to specific activities and plans.

Key sections include –

- Introduction: Introducing the background, goals and problems of the project.
- Literature Review: Comparison with available solutions on the market and how this solution was chosen.
- Methodology: Detail the DSDM methodology chosen, rationale for selection and how it will be implemented.
- Planning: Project Planning, to include management plans resource engagements timelines activity networks critical paths and Gantt charts.
- Feasibility: Technical feasibility, economy and financial analysis, operational feasibility legal feasibility and schedule feasibility.
- Foundation: Problem areas, solutions, requirements and technology to be identified.
- Exploration: Analyze the system use cases, activity diagrams, requirement lists and prototypes.
- Engineering: About the new system modules, Use Case Diagrams, Class Diagram, EERD diagram and Sequence diagrams
- Deployment: Deployment Then the different steps in deployment like Training, Big bang implementation Scaling Load Balancing
- Testing: Specific test steps for unit, module, integration, acceptance performance security accessibility usability testing.
- Critical Appraisal and Evaluation: Criteria used to evaluate successful achievement of goals, those not met successful objectives and recommends for improvement.
- Lessons Learned: Lessons about these challenges, various solutions deployed and overall project learnings.

15.5 Value of the Project

Quick Edu Live is a solution that goes beyond its immediate functionalities and provides scalability to the overall educational ecosystem. Key value propositions:

Enhanced Efficiency: The platform automates daily operations, resulting in a significant reduction of administrative burden on teachers so they can dedicate more time to teaching and interacting with students.

Improved Learning Outcomes: Immediate feedback and in-depth analytics can encourage self-awareness among students, which then leads to targeted improvements for higher learning results.

Data Security and Compliance: The key to effectively managing student privacy in compliance with data protection regulations and maintaining the trust of your end users.

User Satisfaction: An educator facing interface and a full-service teacher support mechanism offers the best experience in terms of enhancing adoption and satisfaction.

15.6 My Experience

My experience with Quick Edu Live allowed me to work on a project, participate in technical development and be part of the responsible team. I learned about the following aspects:

Planning: Detailed planning and requirement analysis helped to reduce the number of technical issues.

Technical: Advanced AI was a valuable experience as it required expert cooperation and scaling, which the learning enabled.

Flexibility: The project taught me to be flexible, adapt to changes and respond to unforeseen challenges.

User-centered: A focus on user feedback allowed creating a user-friendly platform. I have used it for our project from the beginning.

Learning: Iterative development and continuous testing helped to understand when to release the final version thoroughly and when it was possible.

Bibliography

- [1] Skaalvik, & Skaalvik. (2017, August). Teacher Stress and Teacher Self-Efficacy: Relations and Consequences. Research Gate.
https://www.researchgate.net/publication/319325792_Teacher_Stress_and_Teacher_Self-Efficacy_Relations_and_Consequences
- [2] Darling Hammond. (2010, January). Teacher Education and the American Future. Research Gate.
https://www.researchgate.net/publication/249704766_Teacher_Education_and_the_American_Future
- [3] Hattie, & Timperley. (2007, March). The Power of Feedback. Research Gate.
https://www.researchgate.net/publication/258182775_The_Power_of_Feedback
- [4] Shute. (2008, March). Focus on Formative Feedback. Research Gate.
https://www.researchgate.net/publication/220017728_Focus_on_Formative_Feedback
- [5] Besser, Zeigler-Hill, & Flett. (2020, October). Adaptability to a Sudden Transition to Online Learning During the COVID-19 Pandemic: Understanding the Challenges for Students. Research Gate.
https://www.researchgate.net/publication/346280766_Adaptability_to_a_sudden_transition_to_online_learning_during_the_COVID-19_pandemic_Understanding_the_challenges_for_students
- [6] Chen, Xie, & Hwang. (2020, January). A multi-perspective study on Artificial Intelligence in Education: grants, conferences, journals, software tools, institutions, and researchers. Research Gate. https://www.researchgate.net/publication/346225567_A_multi-perspective_study_on_Artificial_Intelligence_in_Education_grants_conferences_journals_software_tools_institutions_and_researchers
- [7] Wang. (2019, April). Elaborated feedback and learning: Examining cognitive and motivational influences. Research Gate.
https://www.researchgate.net/publication/332175556_Elaborated_feedback_and_learning_Examining_cognitive_and_motivational_influences
- [8] Nye. (2015, June). Intelligent Tutoring Systems by and for the Developing World: A Review of Trends and Approaches for Educational Technology in a Global Context. Research Gate.
https://www.researchgate.net/publication/274737499_Intelligent_Tutoring_Systems_by_and_for_the_Developing_World_A_Review_of_Trends_and_Approaches_for_Educational_Technology_in_a_Global_Context

Plagiarism Report

193-16-477

ORIGINALITY REPORT

8%

SIMILARITY INDEX

7%

INTERNET SOURCES

1%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Daffodil International University

Student Paper

3%

2

[dspace.daffodilvarsity.edu.bd:8080](https://dspace.daffodilvarsity.edu.bd/8080)

Internet Source

2%

3

Submitted to University of Greenwich

Student Paper

1%

4

www.researchgate.net

Internet Source

<1%

5

Submitted to University of Leeds

Student Paper

<1%

6

Submitted to NCC Education

Student Paper

<1%

7

Submitted to TAFE NSW Higher Education

Student Paper

<1%

8

Submitted to Collin County Community College

Student Paper

<1%

9

repository.eduhk.hk

Internet Source

<1%