Digital Assessment and Learning Platform for Educational Institutes

by

Asim Baidya
20301239

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2024

Declaration

It is hereby declared that

1. The report submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Asim Baidya
20301239

Approval

The thesis/project titled "Digital Assessment and Learning Platform for Teachers and Students" submitted by

1. Asim Baidya(20301239)

Of Summer, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 2024.

Examining Committee:

Supervisor:
(Member)

Arif Shakil
Lecturer
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

Abstract

This project is about an online learning platform that would facilitate teachers and students in gaining a comprehensive understanding of individual students' aptitudes and deficiencies in a specific course or topic. It will provide valuable insights for analyzing students' performance and progress. Teachers can efficiently create a wide range of question sets for quizzes in many subjects using this platform, using several provided question formats. In addition, teachers can observe students' real-time activity, such as tracking the duration of each student's work and gaining insight into individual performance on specific questions or tests. Students can monitor their progress on quiz and test through thorough performance reports. Additionally, users can assess their expertise in specific topics using tag-based analysis of questions, allowing them to concentrate on the areas that require improvement. In addition, they can create rich toggle based notes to enhance their preparation. Essentially, this proposed site would offer adaptable interfaces to serve the changing requirements of educators and learners, including both teachers and students, to establish a conducive environment for efficient learning and evaluation.

Table of Contents

Declaration i

Approval ii

Abstract iii

Table of Contents iv

List of Figures vi

1 Introduction 1
- 1.1 Projects Objective . . . 1
- 1.2 About The Project . . . 1
- 1.3 Learning Scope . . . 2
- 1.4 Problem Statement . . . 2
  - 1.4.1 Lack of a Platform . . . 2
  - 1.4.2 Timeline of AI and Technology . . . 2
  - 1.4.3 Personalized Analysis . . . 3
- 1.5 Outcome-Based Education . . . 3

2 Literature Reviews(Similar Platforms That Already Exist) 5
- 2.1 Related Projects . . . 5
- 2.2 Comparison With Existence Applications . . . 6

3 Design Analysis 7
- 3.1 Data Model . . . 7
  - 3.1.1 Schema Validation . . . 8
- 3.2 Primary Entity . . . 8
  - 3.2.1 User . . . 8
  - 3.2.2 Course And Enrollment . . . 8
  - 3.2.3 Test and Quiz . . . 8
  - 3.2.4 Question and Tag . . . 9
  - 3.2.5 Student's Activity . . . 9
- 3.3 Marking and Evaluation . . . 10

4 Work in Details 11
- 4.1 Code Base . . . 11
- 4.2 API development . . . 12
- 4.3 Backend Validation . . . 13
- 4.4 Data Fetching . . . 15

4.5 Image Upload 16

5 Application Demonstration **17**
5.1 Teacher's UI I 17
5.2 Course Creation 17
5.3 Quiz and Test Creation 19
5.4 Question Creation and validation 20
5.5 Student - Enrollment, Take Quiz and Test 20
5.6 Teacher's UI II: Report 23
5.7 Student: Stand Alone Note Taker 23
5.8 Admin's Role 26

6 Learning Growth and Challenges **27**

7 Conclusion **28**

List of Figures

1.1 Blooms Taxonomy 4

3.1 Database Schema 7
3.2 Logical Flow of Test Question Access for Student 9
3.3 Marking and Evaluation Algorithmic Process 10

4.1 Project's Codes Base as a Mono Repository in Github 12
4.2 FastAPI's Automatic API Documentation (lock icon) indicate authorized routes 13
4.3 Auto Generated Validation Error Response from Backend 14
4.4 Database Schema Overview: An example of the schema structure for the Question, with its varying JSON formats and constraints. 14
4.5 React Query Debugger: A tool to see how the state of the application changes with the data 15
4.6 The process of uploading an image to filesystem and fetching the identifier that is used to fetch the image 16

5.1 Teachers UI to Create New Course(server successful or error responses are shown using a custom toast at bottom right) 18
5.2 List of Enrolled Students 18
5.3 Teachers Ui to Create Quiz 19
5.4 Teachers UI to Create Test 19
5.5 Question Validation: Backend validation for ensuring question structure and type integrity. 20
5.6 View Test: A UI component for students to view and start tests assigned to them. 21
5.7 View Test Question, Remaining Time is shown at top 21
5.8 View Quiz Questions and submission 22
5.9 Quiz Report: A UI for students to see detailed quiz results and statistics. 22
5.10 a tabular view of the quiz progress of all students 23
5.11 Note with Image: A UI for students to attach images to their notes. . 24
5.12 Flagged Notes: A UI for students to flag important notes that need further attention. 24
5.13 Note List: A UI for students to view and organize their notes. 25
5.14 Raw Note in Database 25
5.15 Admin UI to Delete Image 26

Chapter 1

Introduction

1.1 Projects Objective

In education today, the integration of technology is a crucial part of enhancing the teaching and learning experience. By utilizing the potential of innovative technologies individual students can get more personalized insight into their understanding. To integrate technology into learning this project intends to create a bridge between student and teacher for smoothly evaluating a student's knowledge and comprehension of a particular subject via a quiz. Furthermore, by collecting assessment records of a student, we can predict where individual students need to give more concentration and effort to understand the big picture of a subject. Providing this feature will eliminate the repeated revision of topics that they are already familiar with thereby students will have a more personalized learning experience rather than a generic curriculum that is usually followed in traditional curriculum.
On the other hand, Teachers will be able to use this intended platform for creating quizzes, organizing questions for quizzes, and by sharing the questing they can receive and provide feedback on the effectiveness of question sets. Allowing teachers to share the set of questions and enabling them to provide feedback can produce more coherent and effective questions that will be more accurate at evaluating students' performance. Furthermore, providing a template and an easy-to-use interface can help teachers be more productive in creating test sets and organizing questions which improves the productivity of teachers.
In short, using technology to provide feedback to students based on their performance and helping teachers create test sets and assess students in an easy platform can enhance the learning and teaching process in this technology-heavy world.

1.2 About The Project

As the requirement for the Undergraduate Program at Brac University, every student must complete an internship, thesis, or project that spans three consecutive semesters. Traditionally, 90% of the students go for the Thesis as this is crucial for higher studies, but the other 10% student usually opts for internships or projects(Based on the Registered List for Project/Internship/Thesis). Initially, I wanted to do an internship but the internship culture is not widely embraced in Bangladesh, in the case of Developing Western countries, a student can do both part-time internships and continue their academic semester, but that opportunity

is very distant in Bangladesh. *(But in hindsight, I think I am regretting not doing an internship because doing project alone has lots of drawback like taking all the responsibility and procrastination etc. on the other hand I think working in a company woulds have much better alternative to gain some real world experience)* Presently, I am not interested in writing a thesis consequently I intend to do this project. Initially, I brainstormed various topics for this project and after evaluating my understanding of technology I decided to do this project using web technologies. Another major factor in choosing this particular project is, at the start of COVID-19, almost every student from high school to University is familiar with the ed-tech platform thus building an effective platform can become helpful for many people. Finally, another major reason is, that after graduation, I want to pursue a career in software development as a web developer thereby building a project that can improve my resume for getting better opportunities after graduation. Also, being able to utilize all the technologies that are used in software development industries will help me to become a better developer thereby I have chosen this project and this particular technology.

1.3 Learning Scope

Instead of just building a website with listed features, I have tried to follow all the conventions like tracking features with git repository, utilizing type in codebase to minimize bugs in production(when running). Firstly, I have completed the back-end of the project with initialization of database where I have created apis for each feature. This gave me an opportunity to learn about the back-end concept like authorization with JWT, creating a database using ORM, and creating a RESTful API. Similarly, to make the application usable, I have build the front-end using React and Chakra-Ui library. This gave me an opportunity to learn about the front-end concept like state management using react query, routing with tanstack router, and and principles of web design like responsive design and many more.

1.4 Problem Statement

1.4.1 Lack of a Platform

In the current educational system, the traditional method of teaching and learning is undergoing a significant transformation driven by technological advancement. However, there remains a critical gap in the seamless integration of technology to create comprehensive and iterative learning. There needs to be a dedicated platform for teachers and students to engage in learning and assessment.

1.4.2 Timeline of AI and Technology

At the beginning of COVID-19, many institutions and students were unsure about academic curriculum but as time went on, institutions adopted technology to remotely facilitate learning experiences. For example, Brac University launched a brand new platform called buX to smoothly integrate learning experiences via technology. The majority of the universities did the same and the demand for Zoom and

Google Meet skyrocketed. So, it became a requirement for everyone to use digital platforms and take assessments online via quizzes and writing assignments of tests. As Time passed, the opportunities for plagiarism also increased. Furthermore, in 2023 those universities were operating offline, but the rise of generative AI poses another challenge to the traditional take-home assessment. Popular Generative AI, Chatgpt happened to be an expert in all the academic tests from Chinese Gaokao to complicated Medical exam tests. Thereby, the credibility of home assessments and essays is losing its value so a more interactive and observable platform is now essential to ensure fair assessment and learning enhancement for students and teachers. Therefore, a platform is needed like the one we needed during the COVID-19 period to tackle the issue of increased opportunities for plagiarism.

1.4.3 Personalized Analysis

Another major issue that we can address is personalized learning feedback based on test participation and through analysis based on tests. In academic or job training, we need to fulfill certain requirements and training for qualification, but the utilization of technology for a personalized analysis of where one needs to improve more focused study is not quite there. So, being able to track quiz performance progress in a synchronous platform for a variety of subjects will enable us to generate in-depth insight into understanding.

1.5 Outcome-Based Education

Outcome-based education(OBE) is new in the education system that emphasizes more on active learning and the outcome of learning rather than the passive learning that traditional academics used to follow. Even the University Grants Commission has shown considerable emphasis on OBE but the adoption of OBE is not there yet. Nevertheless, in OBE there is one saying, "All students can learn, but not in the same way and same day", which means, individuals learn differently and need different periods to comprehend something, but there is no known platform or practice in academic that provide through feedback to a student to achieve OBE practically. Therefore, providing custom feedback on each student's performance can help students learn a topic for the outcome of it rather than only for getting grades.
In OBE, Bloom's Taxonomy is often present as a reference that provides 6 fundamental hierarchies. The top-down view of Bloom's Taxonomy looks like the diagram below. This taxonomy can be used to define different levels of human cognition like thinking, learning, and understanding. This classification model is often used by educators for the assessment of tests and other evaluations. Thereby, if intended features are incorporated into the project, the desired platform will be able to provide thorough feedback and a good overview of the understanding of individuals.

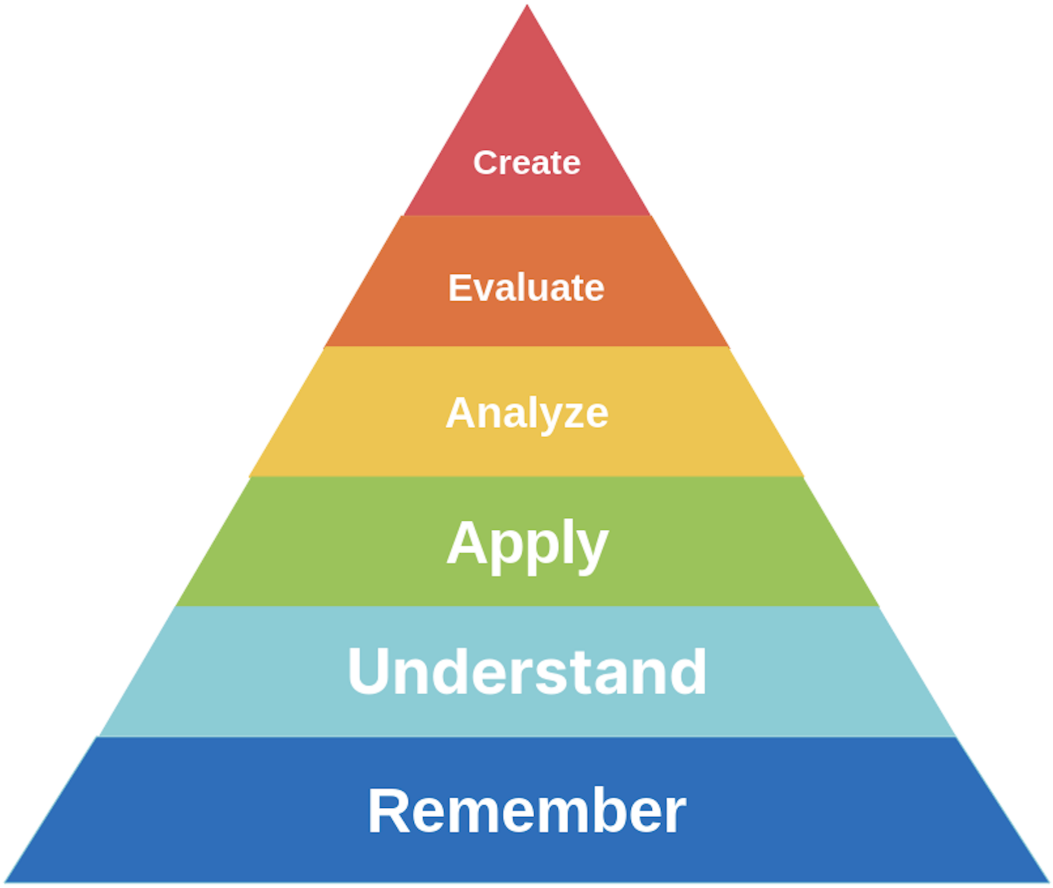

Figure 1.1: Blooms Taxonomy

Chapter 2

Literature Reviews(Similar Platforms That Already Exist)

2.1 Related Projects

In recent years, the integration of digital technologies into education has become a norm, particularly after the COVID-19 lockdown period, it became an inevitable part of academics around the globe. This extensive reliance created a dependency on technology for many academics and as time passed many tools emerged to make the experience of learning more convenient. But there remains some scope for new features that are not present in current software.

In terms of Formal surveys like Quiz assessments, Google Forms is a widely recognized and used tool. Google Forms allows users to create short questions, quizzes, checkboxes, and various other types of questions. For many cases, Google Forms is ideal, but it is not extensible in terms of providing history and feedback based on response. Hence, for a time final examination, Google Forms can be used but for feedback-based questions and interactions Google Form is not the ideal option as it lacks the necessary features for giving a learning experience to students.

Further, in terms of gamification of the learning experience, Kahoot! Is an excellent platform that makes learning much more interactive and engaging. However, Kahoot! Exceptionally good in many aspects, but it does not provide any comprehensive personalized feedback to individual students. Also, the interface of Kahoot! is more focused on gamification and leaderboard rather than giving a smooth experience for both students and teachers. One Principle of OBE is, "All students can learn, but not in the same way and same day". So, if we make the platform for learning purposes also, we should allow some regressive checking of students' performance so teachers can monitor if students learned from iterative attempts of quizzes and tests.

Now in terms of regressive attempts, many MOOC providers like Coursera, edX, and buX allow retakes(or multiple attempts) of quizzes and assessments. In terms of learning, those are more refined as it based on repetitive attempts and give students scope to learn the material while progressing further into the topics. However, those are not stand-alone assessment platforms but rather MOOC platforms. Therefore, in this project, I intend to build a platform like the assessment and quiz taker that platform but more suitable for academic institutions.

2.2 Comparison With Existence Applications

Although there exists a vast number of platforms for similar purposes, there are still tons of requirements for new features thus this application intends to contribute a few. Among those platforms, almost none has any features to provide personalized feedback. Apart from that, This platform will also allow students to take notes that will be helpful for students. Additionally, this proposed platform also focuses on the relationship between repetition and retaining information. Therefore, using an algorithm to make students retake quizzes that they failed on the first attempt will reinforce the content which I believe can create a more effective learning experience.

Chapter 3

Design Analysis

3.1 Data Model

Data is a crucial part of any application. One of the Primary focuses of this project is to make informative data, like when someone participates in a quiz or test, that response data is saved and a collection of that can be used to analyze overall statistical information, for example, based on question tag and correctness, the difficulty level can be estimated. In this application, to asses someone's knowledge and learning, the designed application is based on the following database schema.

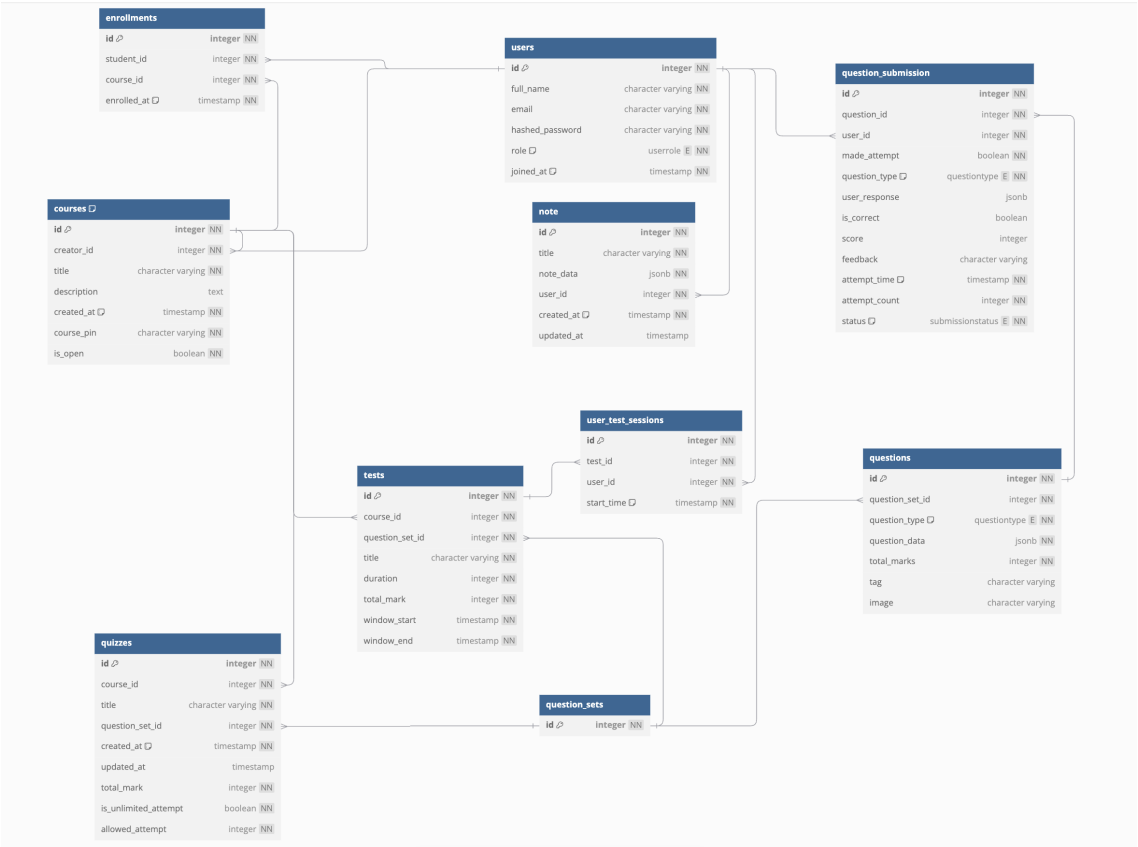

Figure 3.1: Database Schema

To implement this desired ERD I have used the PostgreSQL database to seamlessly integrate the need for complex data structure for different types of questions by using

the JSON. For the structured data, the ERD model was designed and implemented as regular tables. Still, as questions often have varying structures, therefore the information about the question is stored as a JSON in the PostgreSQL database table. Similarly, for User Notes, to give more flexibility and reduce joins JSON is used to store individual Note documents.

3.1.1 Schema Validation

Now as the structure of questions is stored in a JSON field, and without proper validation, the data can get corrupted. To avoid this, I have used Pydantic to validate each JSON field data before storing it in the database. For the Demonstration of validation, **4.3** shows an example of question schema validation when someone submits the data. There you can see, that the data does not conform to the schema and thus the request is rejected by the server with a 422 status code.

3.2 Primary Entity

3.2.1 User

This application has two primary users with distinct roles and disjoint actions. Users with teacher roles will have the flexibility to create courses, quizzes, tests, and questions in quizzes and tests. On the other hand, students will be able to participate in those quizzes and tests. Moreover, they can also add notes in an intuitive format. Now along with these two users, I have also added an Admin role to delete unused images from the folder regardless of who uploaded.

3.2.2 Course And Enrollment

The primary assessment tool will be a Test and Quiz consisting of Questions of four different formats. But to control the access to those tests and quizzes, a Course Entity will be used. Now the Course will have a PIN to enroll, and the PIN is either auto-generated using a UUID or the user can supply a customized PIN. So when creating a course, it's not required to supply the pin, in that case, a custom UUID will be showing in the course card in the course list. The idea is that when this pin is shared along with the course title, a user with a student role can enroll and participate in all quizzes and tests depending on the test window.

3.2.3 Test and Quiz

In this Application, tests and quizzes are designed for two different purposes. Quizzes are designed for more flexible use cases and have no time window but when creating, the creator can control how many attempts can a user make on each question on the Quiz. Even can set unlimited attempts. On the other hand, Tests are more restrictive, and each test will have a time window and duration and students can only participate during that time window. Also unlike Quizzes test scores are released after the test window ends. To design this feature in the application. I am using a new table UserTestSessions that stores start time and that data is used to implement necessary logic to restrict unauthorized access to questions. Also if a

student fails to participate in a test within the designated window, then the test won't be visible to that student. Following diagram illustrates the test access follow for student

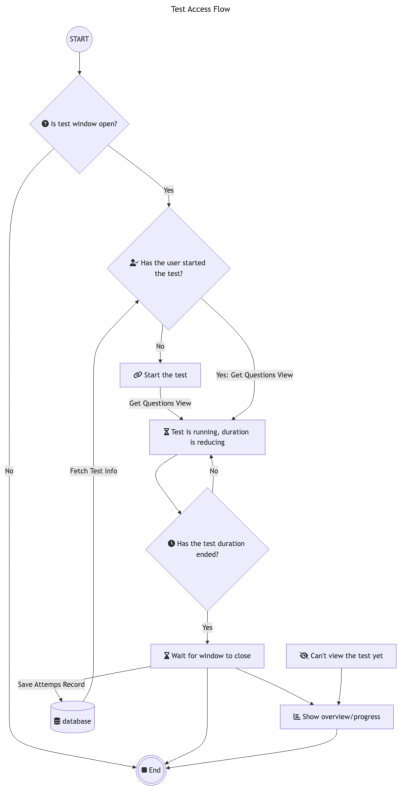

Figure 3.2: Logical Flow of Test Question Access for Student

3.2.4 Question and Tag

From the assessment perspective, questions are the heart of this application. The question is the primary Entity that students and teachers interact with. The teacher creates questions and students attempt to assess their understanding. Now to provide insight about a topic, each question can have a tag, which can be used to filter questions based on the topic from teachers' UI for balancing question of multiple topic. On the other hand students get a hints about the topic based on tag.

3.2.5 Student's Activity

As mentioned, the student will have the ability to enroll in courses and participate in quizzes and tests under a course. Apart from participation in assessment, students can log in themselves and use this platform for day-to-day note-taking in an intuitive format which would enable them to review their notes effectively. Intuitive in the sense that, traditional notes are the plain text of information but here notes are inspired by Notion's Toggle note, thus each note is like a flashcard of information for the study. As this is primarily focused on learning, each note is represented as a document that consists of multiple notes. Each note can be thought of as a memory chunk each note in the document can have a different state and states are synced in the database and also hinted with color in the user's. The design of these features

addresses study topics that require memorization. This is more effective for pushing information into long-term memory than traditional review or reading.

3.3 Marking and Evaluation

When a student submits the quiz, a customized algorithm is developed to handle the marking and feedback calculation process. In order to give feedback about their response, an automatically generated feedback is stored while marking the submission along with the number of attempts for validating limited attempts in the quiz. The whole process of marking is shown in the following diagram.

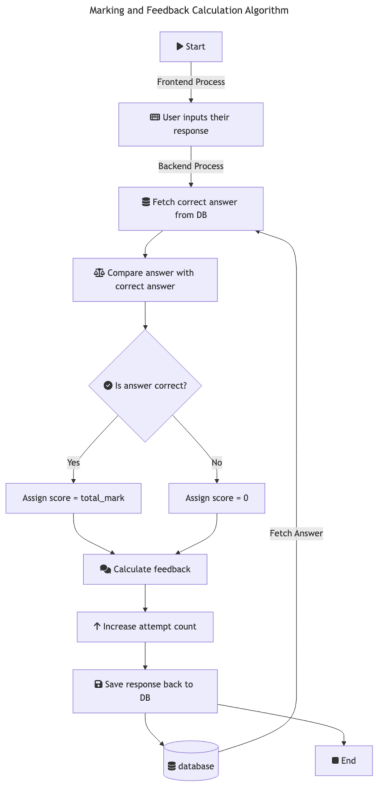

Figure 3.3: Marking and Evaluation Algorithmic Process

Chapter 4

Work in Details

To build the **ERD** in practice using web technologies, I have used contemporary technology. For the separation of concerns, I opted to build the project in two parts. Firstly, a backend that will interact with the database and expose the necessary API endpoints to perform application logic. I chose this approach because it follows contemporary web development practices and offers benefits such as API reusability across different platforms. By using the same backend, another application can connect to it with authorized access.
To implement the backend, I have chosen **FastAPI**, and for the front, I opted for **ReactJS**. The primary reason behind choosing **FastAPI** is my familiarity with Python, and among other Python libraries, **FastAPI** offers modern Python support along with the integration of **Pydantic** for data validation. Since I am using **JSON** as a field in the database table, the application needed to ensure data consistency to scale effectively. On the front end, **ReactJS** is used to provide a user-friendly, engaging, and incremental user interface by leveraging reusable components. In the UI part instead of creating a custom UI for each component, I used the Chakra-UI library, which provides a set of components that can be used to build a modern web application faster. It also makes it easier to implement dark mode support.

4.1 Code Base

To keep the project organized, I have used a mono repository to store both frontend and backend at the same folder. This reduces management complexity and makes it easy to track changes in one place rather than using two separate repositories. However, as both frontend and backend are scoped under different directories, and changes are committed with separate branches, this approach avoided possible conflict when trying to revert to a previous commit. On when merged to the main branch, temporary branches are deleted to keep the repository commit history linear.

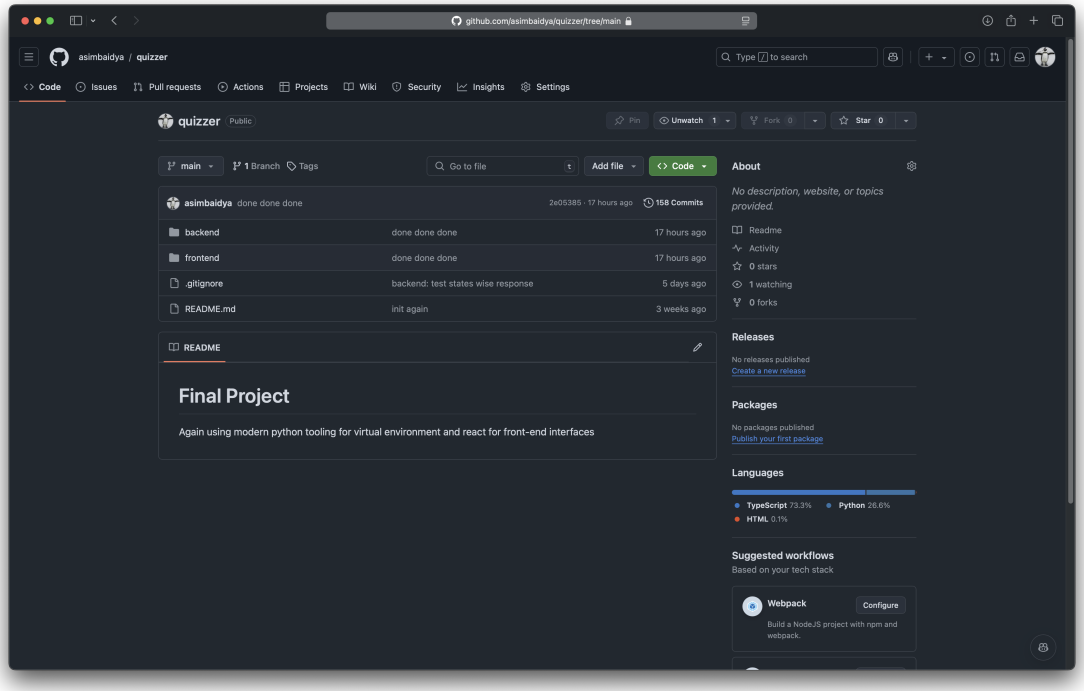

Figure 4.1: Project's Codes Base as a Mono Repository in Github

I have completed the project and kept every major changes recorded in around **157+ commits**. Using git for version control allows to revert back to any previous state of the application for debugging and pivoting purposes.

4.2 API development

For the Backend, I have created APIs to interact with the main application. These APIs are the primary gateway to interact with the application. When creating an API is important to ensure that the API is working as designed, for that, an interface is required. I am using FastAPI, which has automatic API documentation generation that gives an overview and testing capabilities for all the API endpoints. This was important because to test whether the backend logic was working or not, I could simply use the documentation to test. On top of that, this also allows authorized manual API testing. Below is an example of the automatically generated documentation.

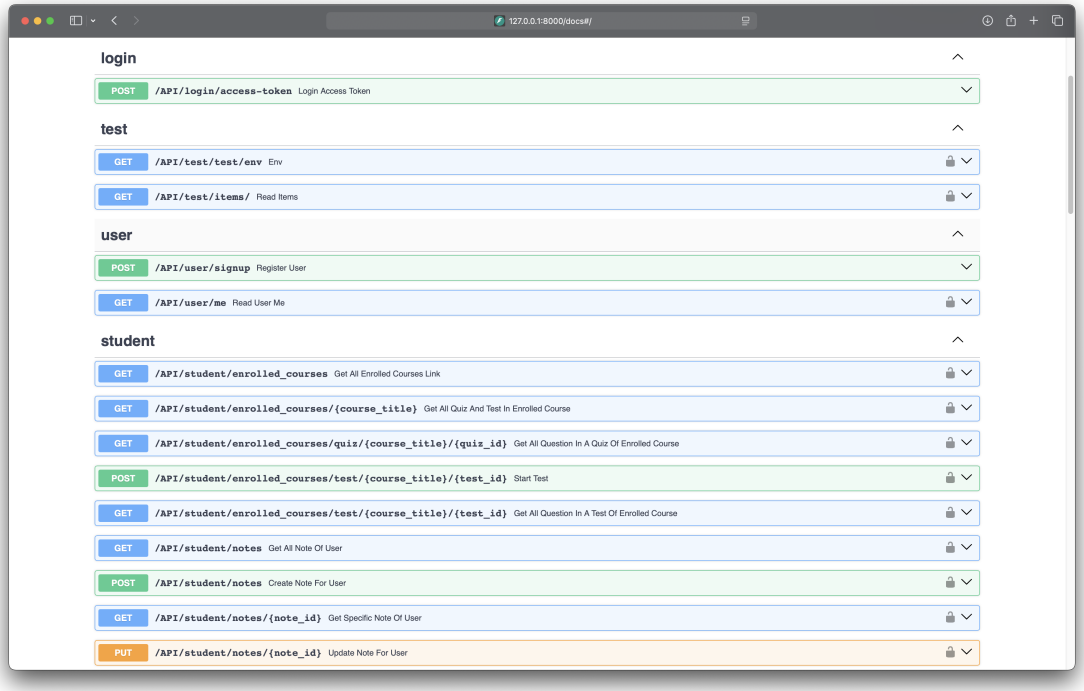

Figure 4.2: FastAPI's Automatic API Documentation (lock icon) indicate authorized routes

4.3 Backend Validation

In this application, I validate the data in both the backend and frontend. Backend data validation is crucial because if further front-ends like mobile apps want to incorporate these APIs, they don't necessarily need to implement the validation themselves. On the front-ends, the validation is useful because it can provide instant feedback to users which provides a more interactive and engaging experience than waiting for server validation and showing that response. Since FastAPI provides built-in support for validation using Pydantic on the server side. It also provides information about data validation in the documentation and, in case of unprocessable or invalid data, returns a response with a 422 (Error: Unprocessable Entity) along with precise feedback about what was wrong with the data. Here is an example document that simplifies it all, here the validation error is wrapped in a response that tells exactly what went wrong with the data.

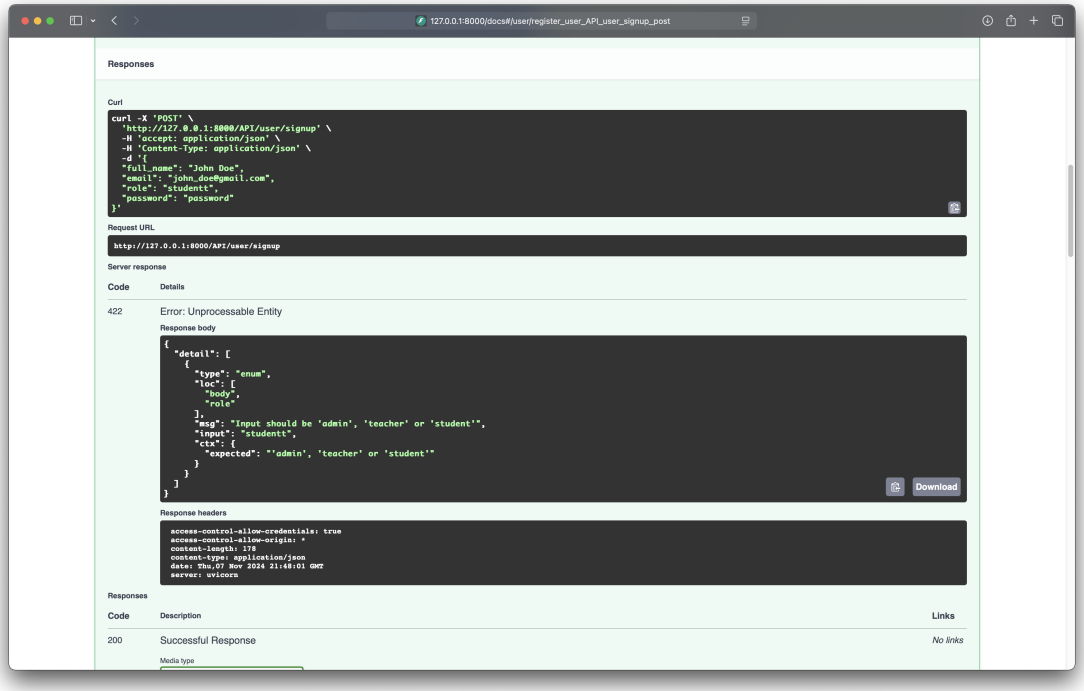

Figure 4.3: Auto Generated Validation Error Response from Backend

For example, a typo is detected for a role that has only 3 allowed values. Moreover, the documentation also lists all the database schema (validation model) to give a proper overview of the allowed data structure. This is crucial for an application where the backend and frontend are managed by separate teams because the frontend team can get useful information out of the box. Here is the allowed document structure for the Question, which has a JSON field with 4 varying structures.

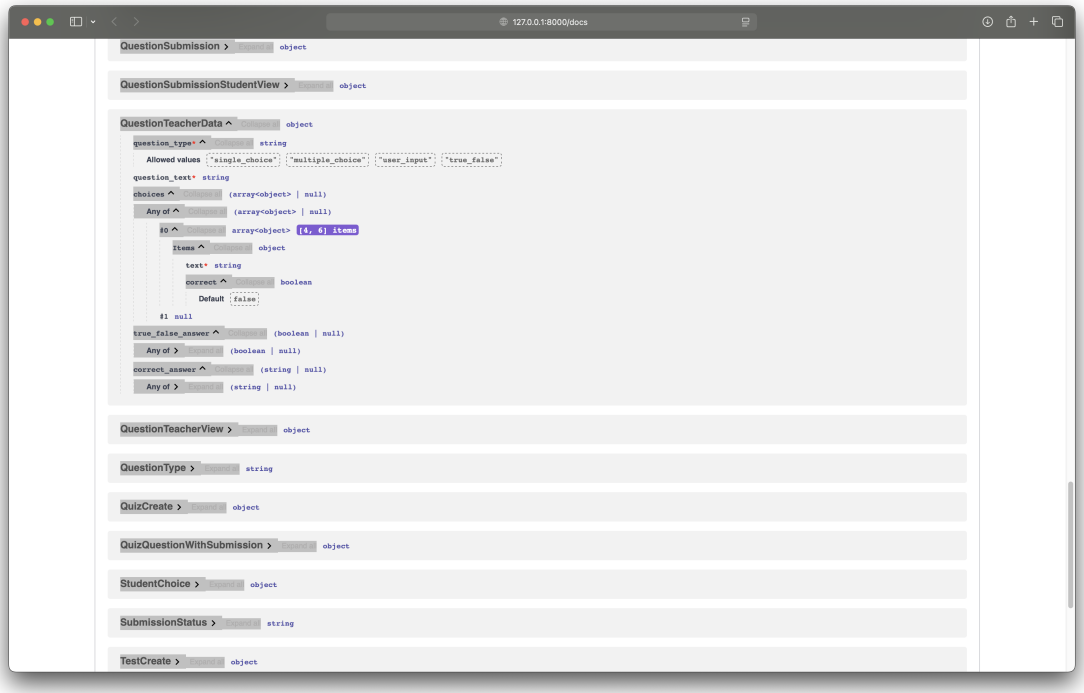

Figure 4.4: Database Schema Overview: An example of the schema structure for the Question, with its varying JSON formats and constraints.

The question type is a enum with 4 possible value(visible in the previous documentation figure)

4.4 Data Fetching

Now in the front end, I have used a few technologies like Zod for data validation for form submission and, tanstack router for organizing each component and application actions in a single place. However, the most important part of the frontend is the data fetching. This is important because when any data changes, some parts of the UI may need to change as well. For example, when creating questions or quizzes, the data in the server gets updated, and to show that on the user's end without reloading, I have used react-query. React-query is a library that provides a set of hooks to fetch data from the server and cache the fetched data using a key. This is important because it reduces the number of requests to the server and also provides a way to invalidate specific cache using the key when the data changes, therefore on successful post request, I can tell react query which data needs to be re-fetched. Using this library has helped me to build a stateful react application without the need for an external library to manage the state. Also, react query has a debugger that can help to change the state of API data and see how the UI is changing.

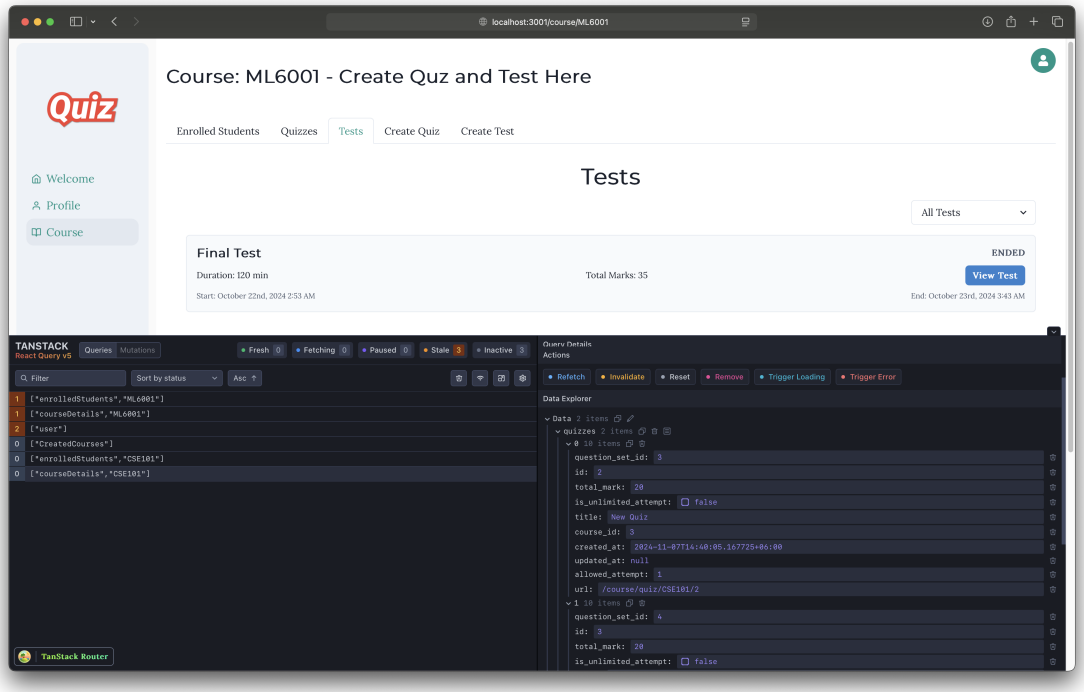

Figure 4.5: React Query Debugger: A tool to see how the state of the application changes with the data

4.5 Image Upload

To allow image uploads, in question and notes, I have used two custom hooks that save the image on a static folder with a unique file name as identifier. This is just a minimal implementation of what a storage bucket can do but it is less restrictive. Also as the image upload/fetch API endpoints are hardcoded in the hook, it can cause issues when a note is deleted or after upload, the image is changed or removed, I have addressed that at the end of the next chapter, but here is a process of how image upload work in note and question
As soon as the user selects the add image button, it opens a file dialog, and after selecting the image, the image is uploaded to the server and the server immediately responds with the identifier. On form submission, that identifier is stored in the database simultaneously the hook use another API endpoint to fetches the image using the responded identifier(or note's image identifier for note). The image upload links are hard-coded, only authenticated users can upload images. Below is a diagram that illustrates the process of image upload in the application.

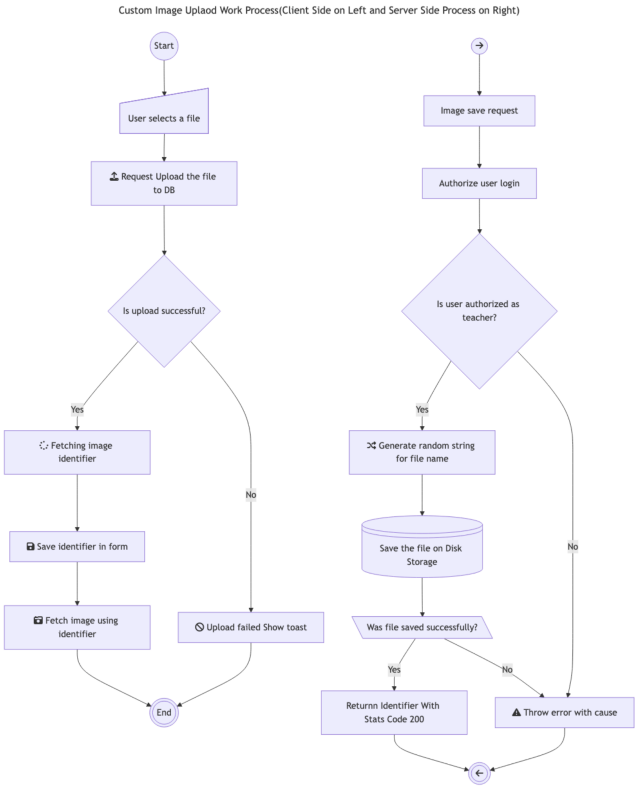

Figure 4.6: The process of uploading an image to filesystem and fetching the identifier that is used to fetch the image

Chapter 5

Application Demonstration

To use the application, one needs to authenticate using the login form, or one can enter the system via the sign-up form. After signing up, based on the role, different views and actions will appear in the sidebar. To distinguish the user role, I have used light mode for teachers demo and dark mode for student's demos.

5.1 Teacher's UI I

If the user logged in is a teacher, then they will have a set of limited actions. The available action will be listed in the Primary Sidebar on the left and they can then navigate to nested routes to perform the following actions.

5.2 Course Creation

The teacher will have this UI to create new courses and view all the courses under the course section. On the "Add New" tab, the user can add a new course. The pin here is used to let students enroll in a course.

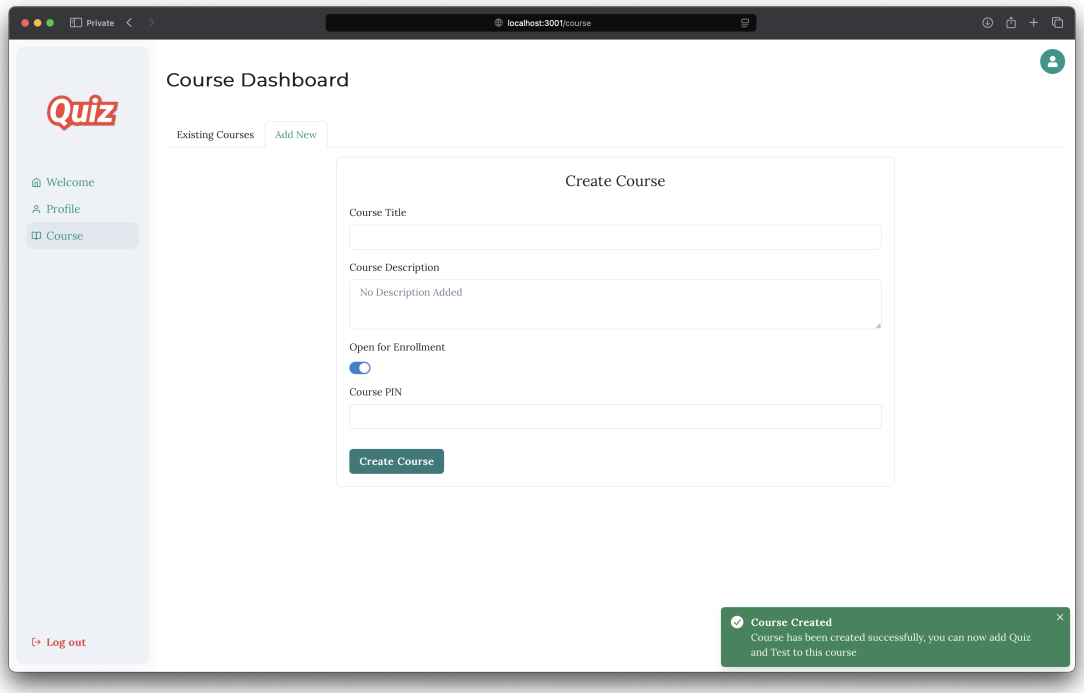

Figure 5.1: Teachers UI to Create New Course(server successful or error responses are shown using a custom toast at bottom right)

Also all the Enrolled students will be listed in the "Enrolled Students" tab. here teacher can see current students for a particular course.

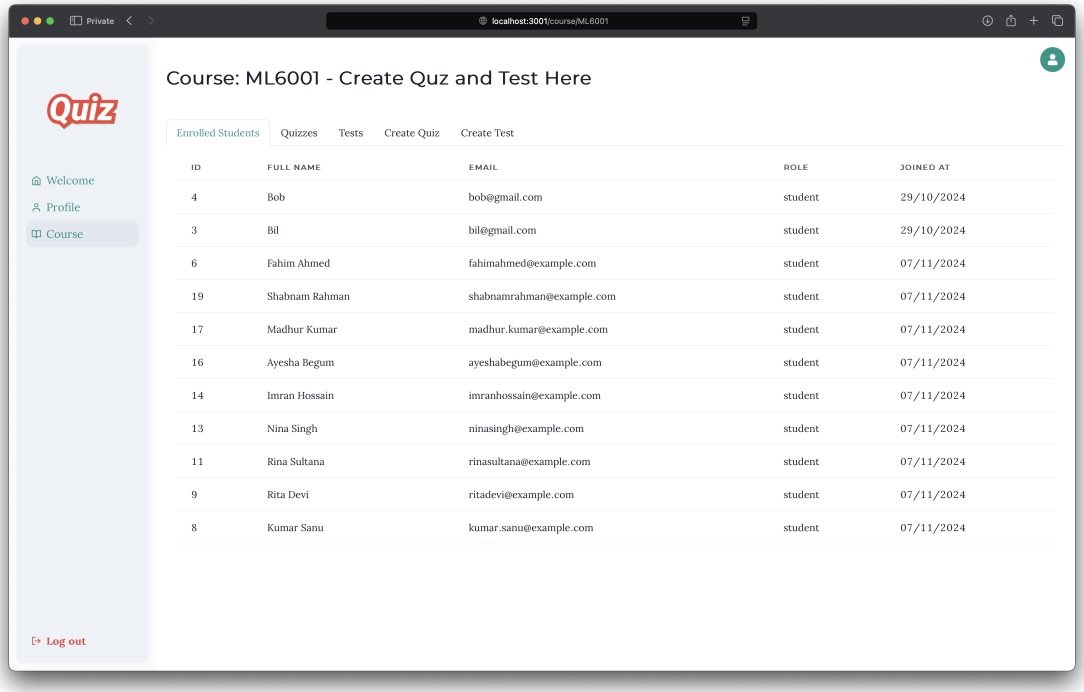

Figure 5.2: List of Enrolled Students

5.3 Quiz and Test Creation

On the "Quiz" tab, teachers can create quizzes and set various parameters like whether a quiz has unlimited attempts or limited attempts.

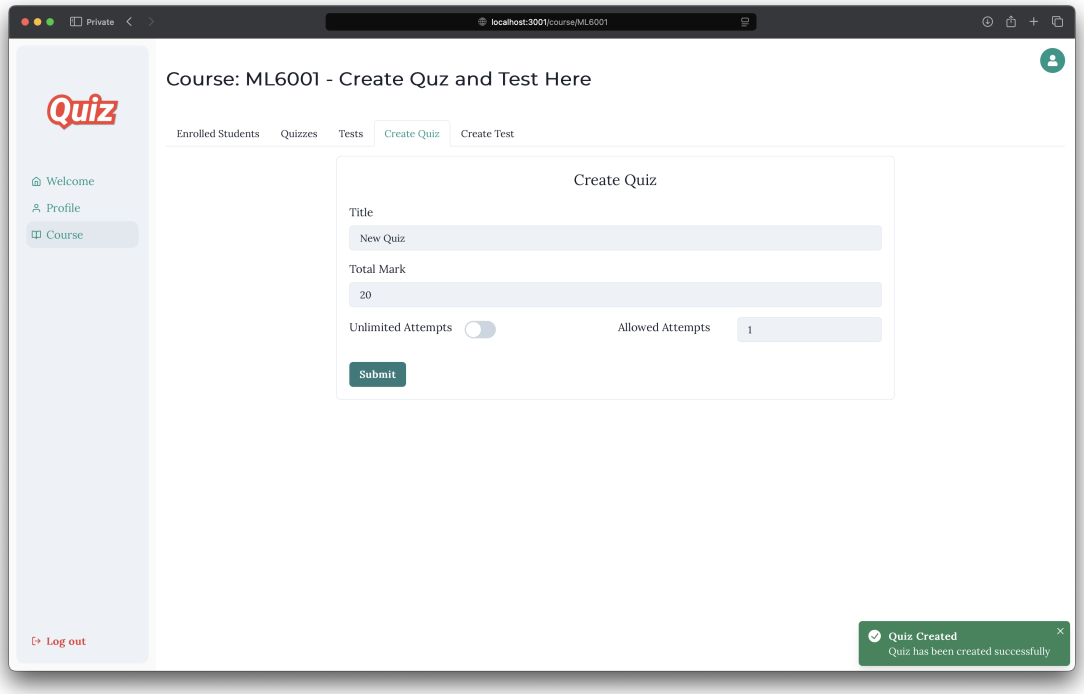

Figure 5.3: Teachers Ui to Create Quizzes

Now for the Test, the form needs a valid time window and the duration of the test in minutes.

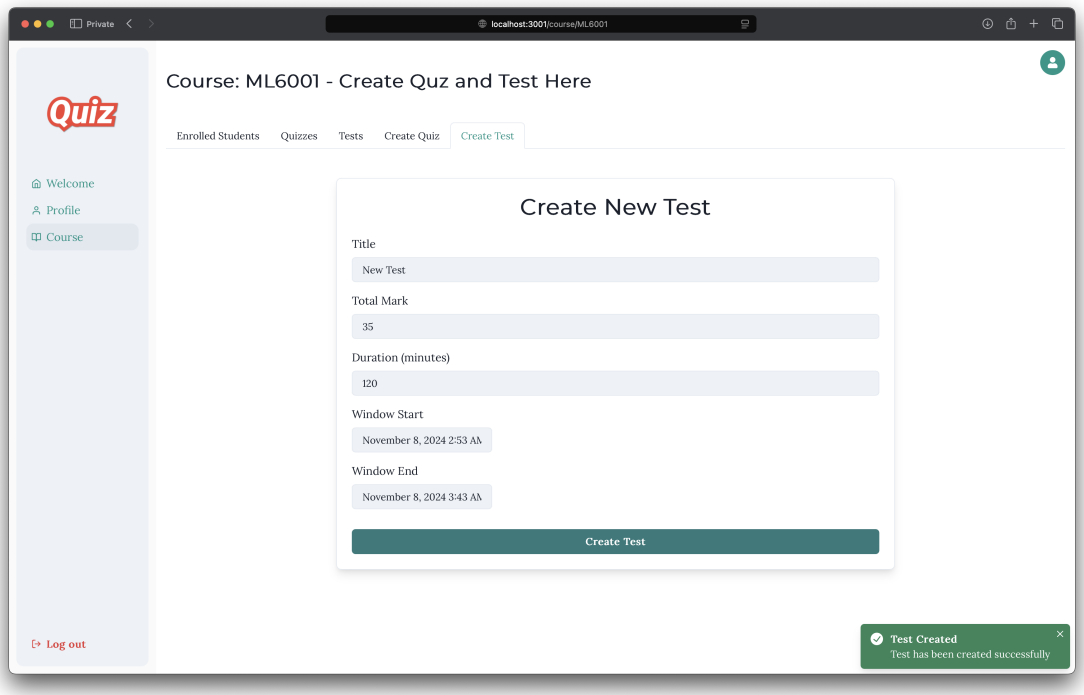

Figure 5.4: Teachers UI to Create Tests

5.4 Question Creation and validation

Finally, for the assessments, teachers can add questions to already created quizzes and tests. However, in the case of a test, the question can be added before the start window. To give instant feedback on validation errors, the Zod library is used to validate the form data before in client's browser before submission.

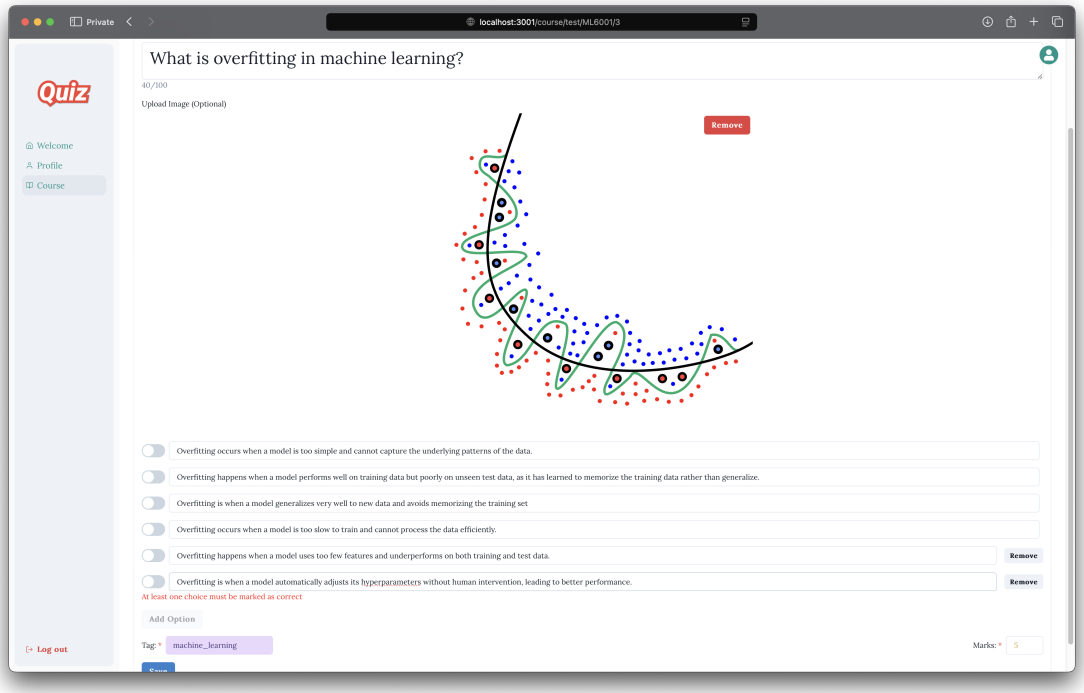

Figure 5.5: Question Validation: Backend validation for ensuring question structure and type integrity.

5.5 Student - Enrollment, Take Quiz and Test

The student's user interface allows them to view and take tests under a course within the allowed time window. In the given example, time window is open and current user has not attempted thus it show an button to start the test which will redirect to the test

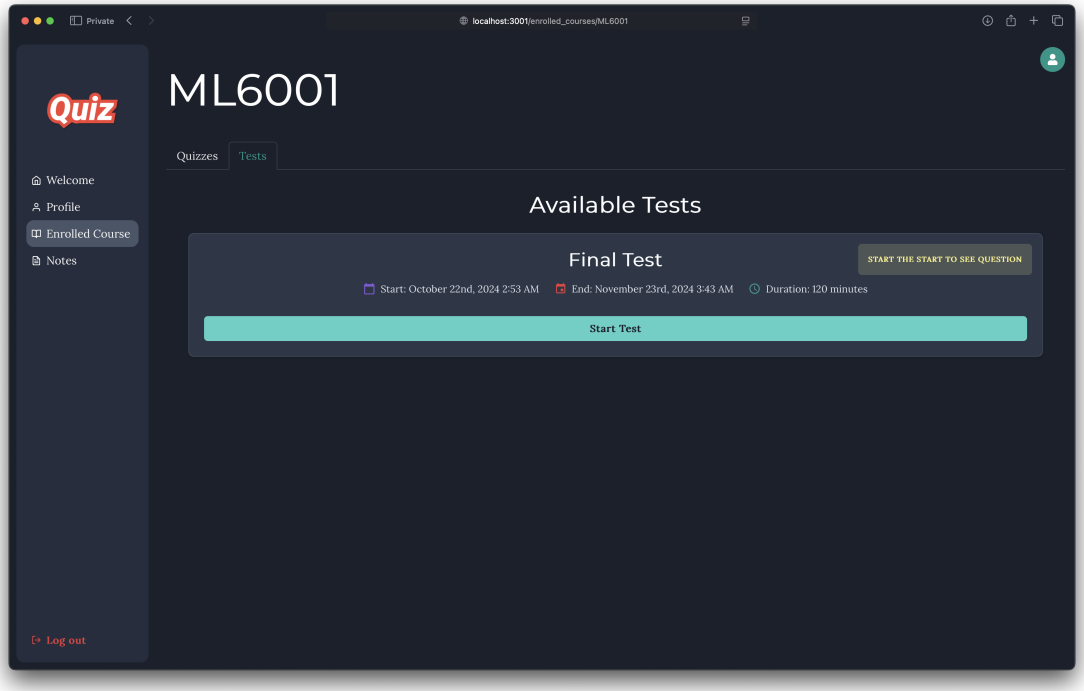

Figure 5.6: View Test: A UI component for students to view and start tests assigned to them.

UI for taking test

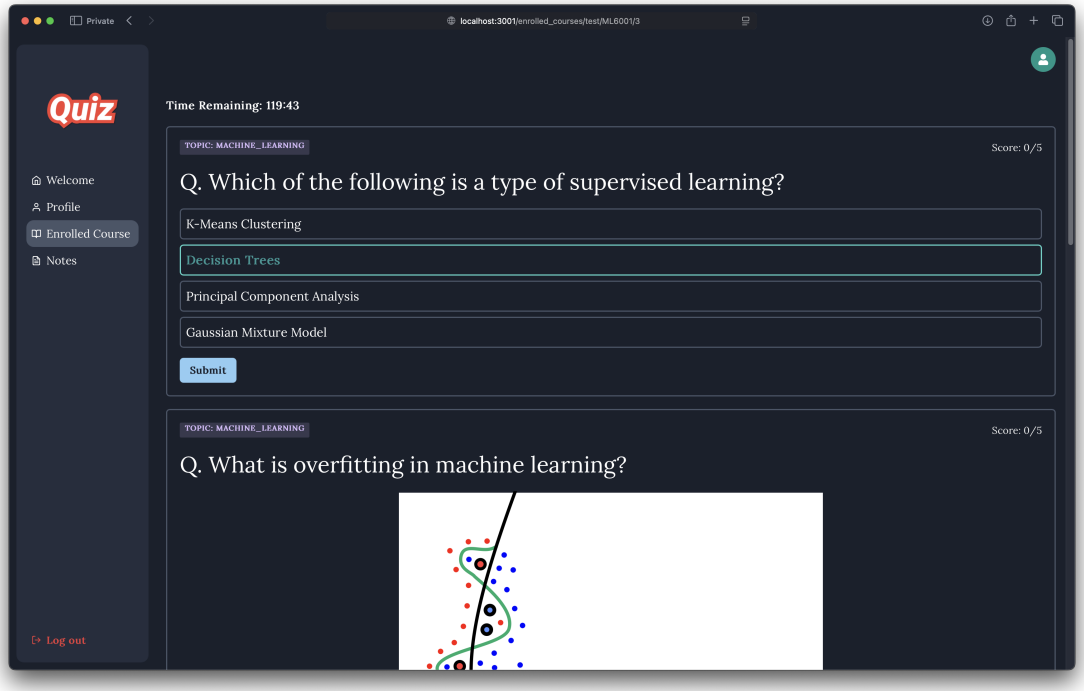

Figure 5.7: View Test Question, Remaining Time is shown at top

UI for giving a quiz

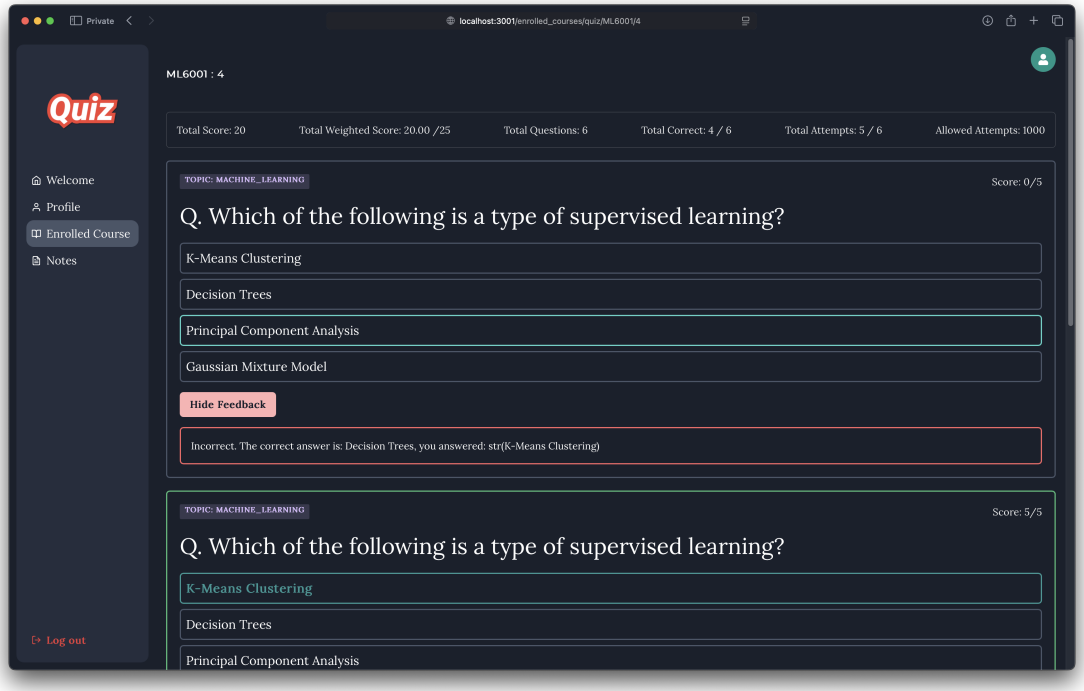

Figure 5.8: View Quiz Questions and submission

Now for taking the test, students can see the question and submit the answer only during the window and only for the allowed duration. In the case of the quiz, the student can take the quiz at any time and if multiple attempts are allowed, they can retake the quiz. Also to enhance the learning, after the attempt, the feedback will be hidden so they can test again to check their understanding.

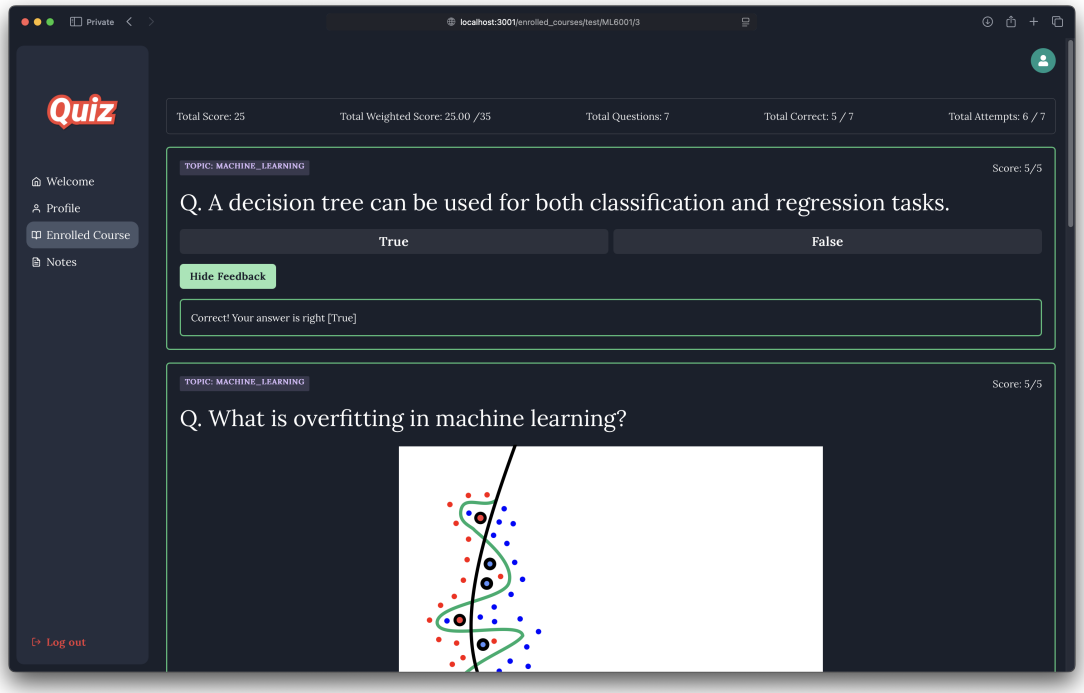

Figure 5.9: Quiz Report: A UI for students to see detailed quiz results and statistics.

5.6 Teacher's UI II: Report

Now in order to track who has attended the test and quiz, teachers can view the progress of all students under quiz and tests. Here many useful information is generated to to help teachers.

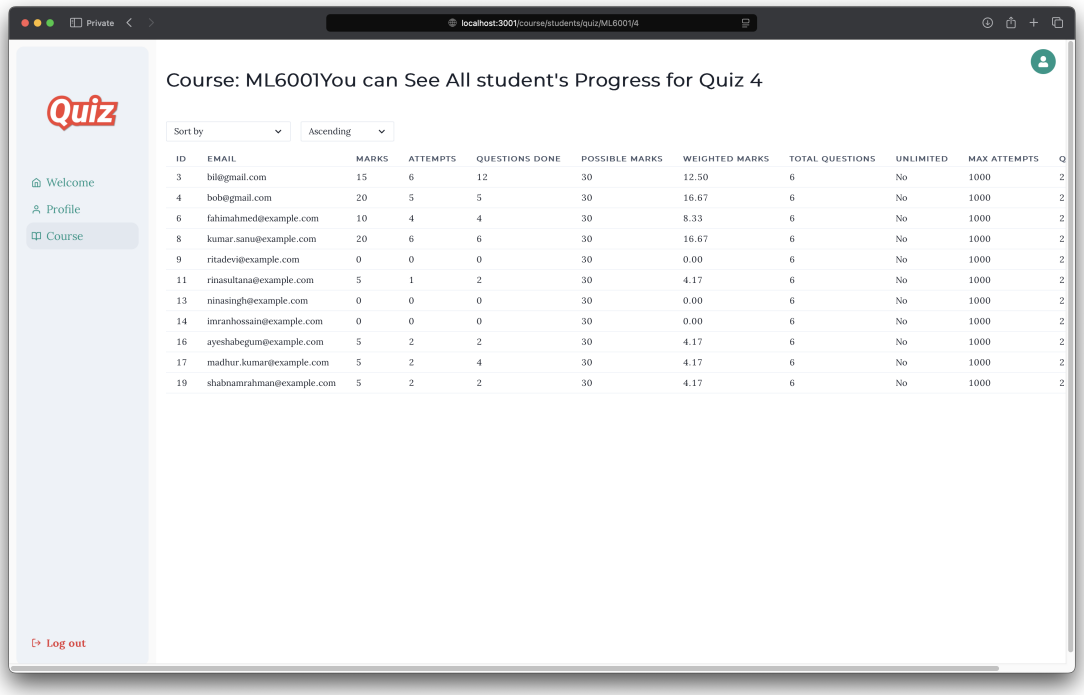

Figure 5.10: a tabular view of the quiz progress of all students

Firstly, a weighted mark is calculated with the idea that each question can have a varying number of marks depending on the importance of a topic and the total score is the sum of all marks in a test or quiz. However, the weighted mark is a fixed number what contributes to grade(if applicable) and will be used to calculate the final mark. Also, teachers can see students' metadata like how many attempts they have made, etc.

5.7 Student: Stand Alone Note Taker

Now to help students learn a topic, this application also incorporates an intuitive note-taking system, where students can take notes in a flashcard-like format with support for images.

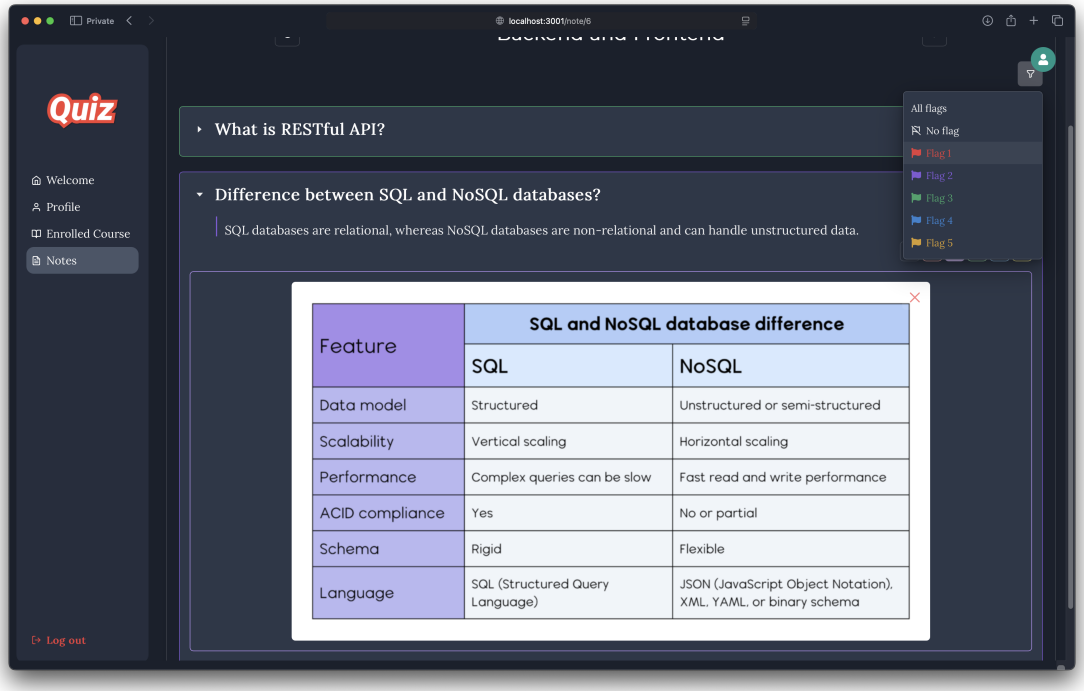

Figure 5.11: Note with Image: A UI for students to attach images to their notes.

A single Note is a structured list of information chunks, and inside each Note, students can have some chunks that are important and some that are hard for hard for particular student, they can filter the notes by selecting a tag to indicate state/difficulty/grouping etc.
Students can flag notes for review, which will highlight important or unclear points.

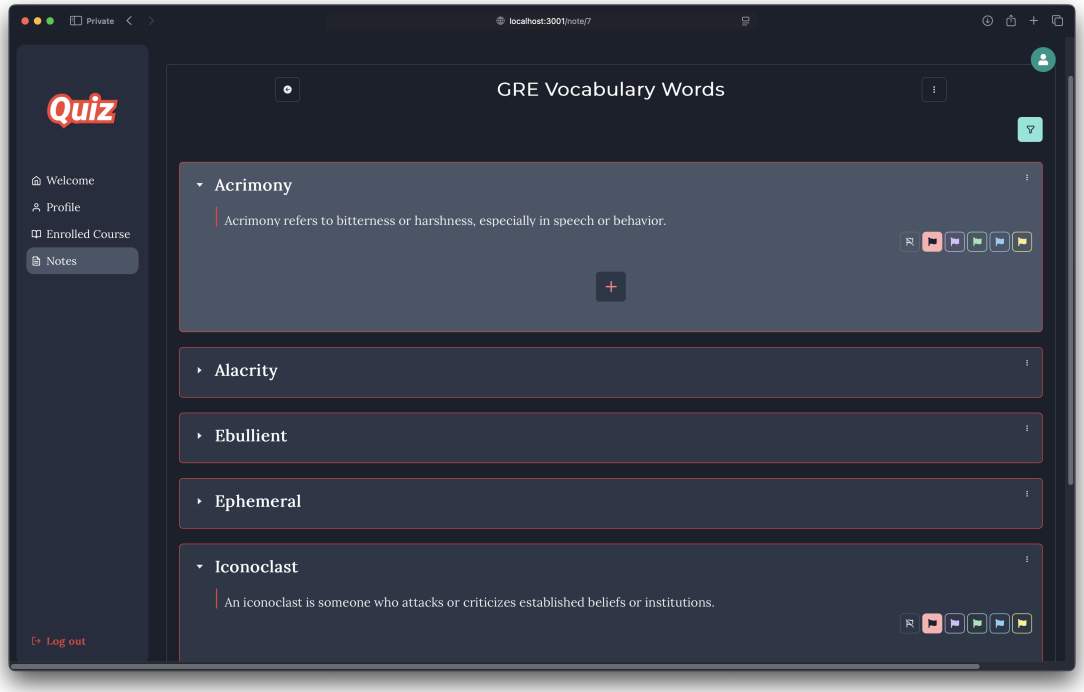

Figure 5.12: Flagged Notes: A UI for students to flag important notes that need further attention.

Also, they can organize all of their note under the note list in the sidebar along with the ability to delete or edit the note.

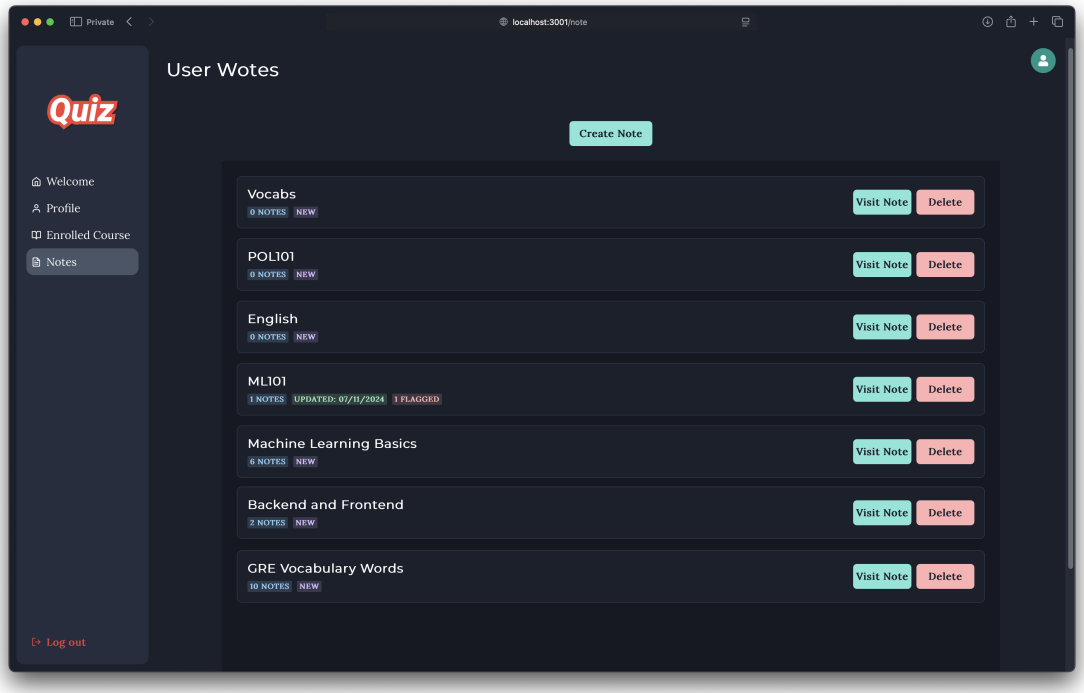

Figure 5.13: Note List: A UI for students to view and organize their notes.

The structure of the note store in a JSON field here is the actual note without any UI in the database, in the case of a note, the UI does the work to make tiny snippets of text useful

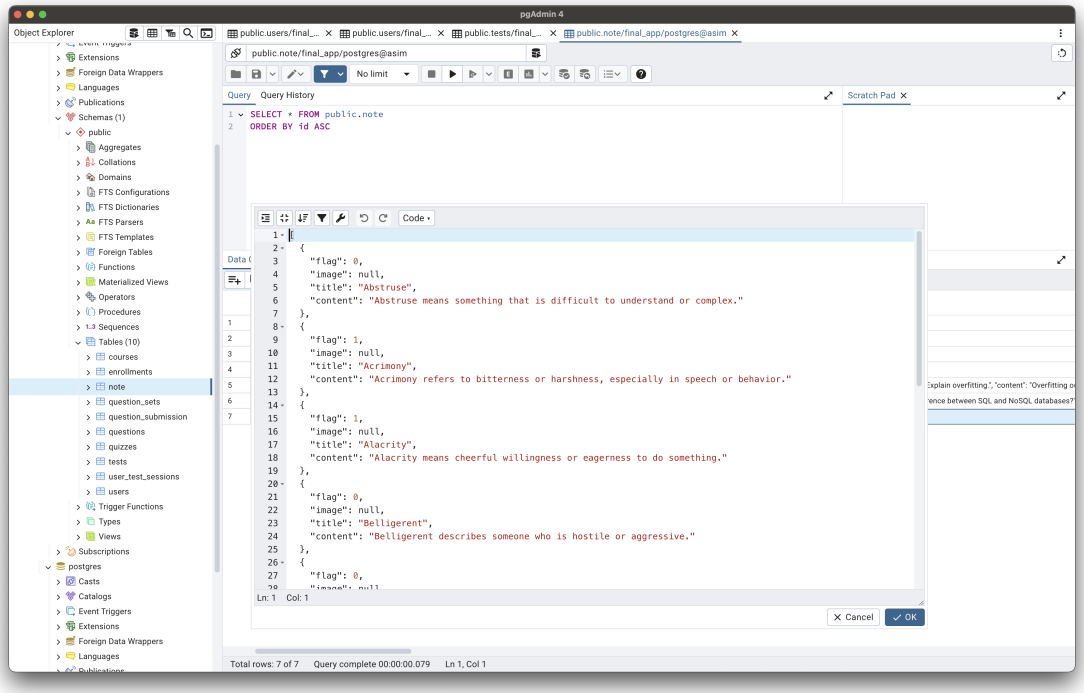

Figure 5.14: Raw Note in Database

5.8 Admin's Role

As mentioned in the previous chapter, image uploads are handled via a react hook, after row deletion in the database, images are not deleted as they are not stored in a database. Admin can delete the image from the server. For this action admin tracks all the image files in the database and all the file names, if a file only exists in the folder and not in the database, it will be deleted.

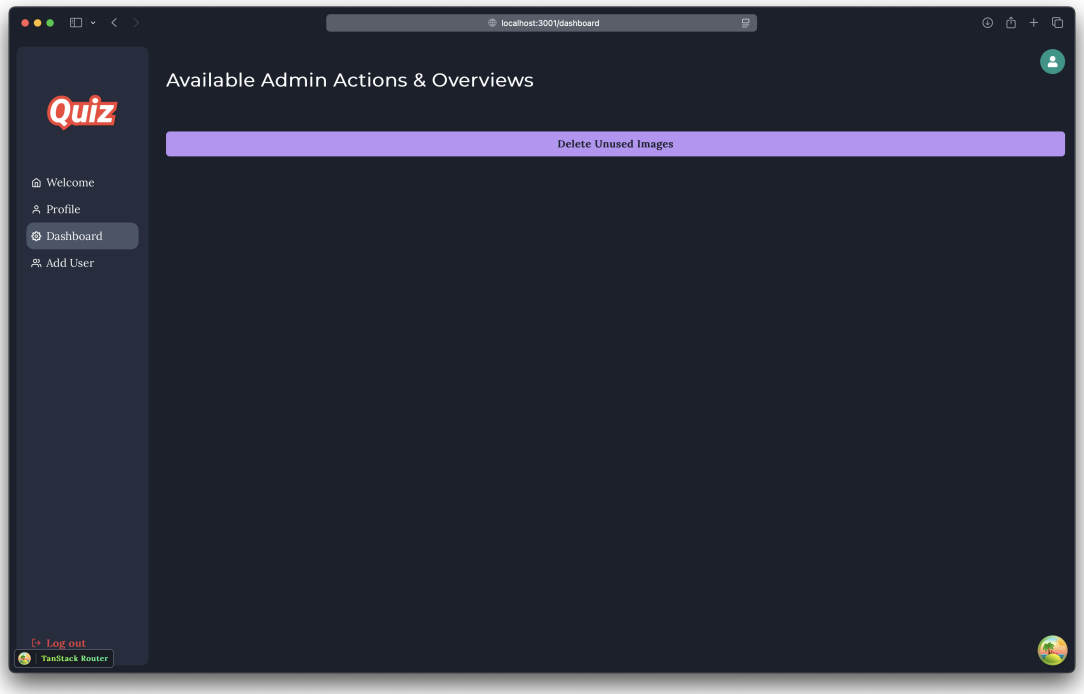

Figure 5.15: Admin UI to Delete Image

Chapter 6

Learning Growth and Challenges

One of the primary reasons behind doing a software project instead of a thesis was that I wanted to learn a few sets of technology. This is because my goal is to work as a software developer. After exploring various fields, I found web technology interesting, however, I did not know many things, like the set of technology that is common in the web, because there are thousands of libraries to build the same thing on the web. So it was hard to follow a fixed path or fixed set of technology. Secondly, I found it extremely hard to handle edge cases on an application. Especially, when edge cases are not designed beforehand, like what happens if the same user logs in from two devices, and then starts to submit questions from both devices, should the server be concerned about it, etc. Moreover, working with two different repositories under a mono repository was confusing as there were too many files. Also for relational databases, different join relation management was quite challenging. Lastly, the most difficult challenge was to follow a schedule for project completion in a solo project.

Chapter 7

Conclusion

In conclusion, this final-year project aims to create a user-friendly interface for teachers and students, enhancing technology integration in academic settings. Simultaneously, it was a practical opportunity to advance and acquire my skills as a web developer. The project promises to deliver features that contribute to an improved learning experience. Throughout its development, I have learned various concepts and technologies that I wish to use in my future projects.