CSCE 5350: Fundamentals of Database System

Quiz Management System

Project Report

Instructor: Dr. Nirosha Dinayadura

Date: 19 April 2024

Team Members:

Name	EUID
Kishan Kumar Zalavadia	11685261
Uday Bhaskar Valapadasu	11696364
Sapthagiri Naik Bhukya	11699072

Table of Contents:

1. Abstract:

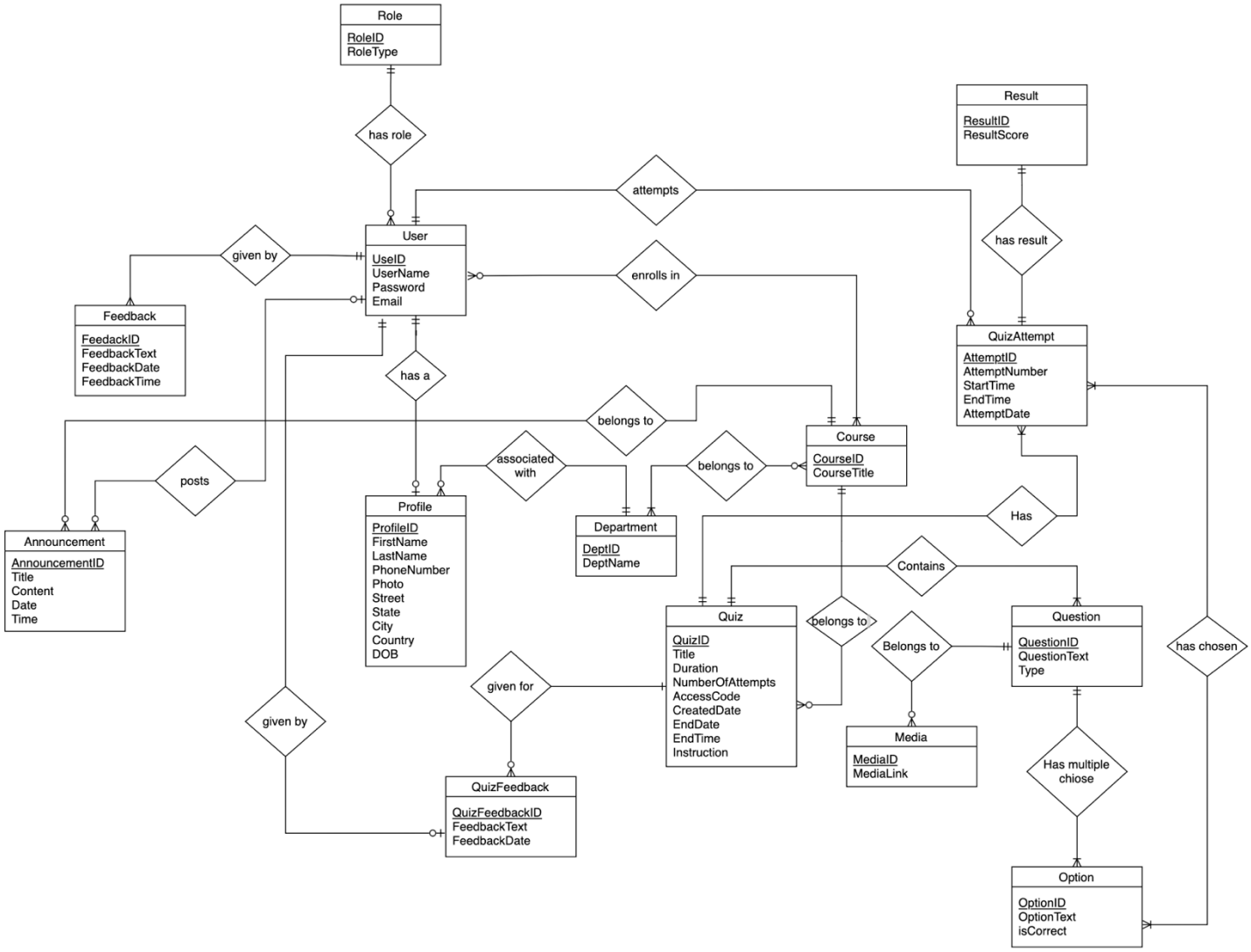
Our main objective in developing this project is to gain a deep understanding of database schema structures and apply this knowledge to create a real-time quiz management system application. By designing and implementing an interconnected **Entity-Relationship (ER)** diagram within the system's database, we aim to address practical challenges in educational assessments. This hands-on experience will enhance our comprehension of how databases manage complex relationships and transactions in an actual application environment, while also providing a user-friendly platform for both educators and students to conduct and participate in quizzes seamlessly.

2. Introduction:

The quiz management system is designed to improve the assessment process. This system is easy to use and has benefits like tracking the quiz history. Professors can easily conduct quizzes with multiple attempts. And the students can take quizzes conveniently. This system's database consists of 17 tables which are interconnected. Our system gives a very convenient UI which makes it attractive to use.

3. System Analysis:

3.1 Database design (Entity Relationship Diagram):



3.2. Relational Schema

Below are the entities of the Quiz Management System

1. User
2. Profile
3. Role
4. Feedback
5. Department
6. Course
7. User_Course
8. Department_Course
9. Quiz
10. Quiz_Feedback
11. Question
12. Option
13. Media

14. Attempts
15. Quiz_Attempt_Option
16. Result
17. Announcement

Tables_in_qms
announcement
attempts
course
department
department_course
feedback
media
options
profile
question
quiz
quiz_attempt_option
quiz_feedback
result
role
user
user_course

1. **User**

Field	Type	Null	Key	Default	Extra
userid	int	NO	PRI	NULL	auto_increment
emailid	varchar(255)	YES		NULL	
password	varchar(255)	YES		NULL	
user_name	varchar(255)	YES		NULL	
profileid	int	YES	UNI	NULL	
roleid	int	YES	MUL	NULL	

2. **Profile**

Field	Type	Null	Key	Default	Extra
profileid	int	NO	PRI	NULL	auto_increment
age	int	NO		NULL	
apt_number	varchar(255)	YES		NULL	
city	varchar(255)	YES		NULL	
country	varchar(255)	YES		NULL	
dob	date	YES		NULL	
first_name	varchar(255)	YES		NULL	
last_name	varchar(255)	YES		NULL	
phone_number	bigint	NO		NULL	
state	varchar(255)	YES		NULL	
street_name	varchar(255)	YES		NULL	
street_number	varchar(255)	YES		NULL	
zip	varchar(255)	YES		NULL	
departmentid	int	YES	MUL	NULL	

3. **Role**

Field	Type	Null	Key	Default	Extra
roleid	int	NO	PRI	NULL	auto_increment
role_type	varchar(255)	YES		NULL	

4. **Feedback**

Field	Type	Null	Key	Default	Extra
feedbackid	int	NO	PRI	NULL	auto_increment
feedback_date	datetime(6)	YES		NULL	
feedback_text	varchar(255)	YES		NULL	
feedback_time	time(6)	YES		NULL	
userid	int	YES	MUL	NULL	

5. **Department**

Field	Type	Null	Key	Default	Extra
departmentid	int	NO	PRI	NULL	auto_increment
department_name	varchar(255)	YES		NULL	

6. **Course**

Field	Type	Null	Key	Default	Extra
course_id	int	NO	PRI	NULL	auto_increment
course_name	varchar(255)	YES	UNI	NULL	
professor_userid	int	YES	MUL	NULL	

7. **User_Course**

Field	Type	Null	Key	Default	Extra
user_id	int	NO	MUL	NULL	
course_id	int	NO	MUL	NULL	

8. **Department_Course**

Field	Type	Null	Key	Default	Extra
course_id	int	NO	MUL	NULL	
department_id	int	NO	MUL	NULL	

9. **Quiz**

Field	Type	Null	Key	Default	Extra
quiz_id	int	NO	PRI	NULL	auto_increment
access_code	varchar(255)	YES		NULL	
created_date	date	YES		NULL	
end_date	date	YES		NULL	
end_time	time(6)	YES		NULL	
instruction	varchar(255)	YES		NULL	
quiz_duration	int	NO		NULL	
quiz_number_of_attempts	int	NO		NULL	
quiz_title	varchar(255)	YES		NULL	
course_course_id	int	YES	MUL	NULL	

10. **Quiz_Feedback**

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
date	date	YES		NULL	
text	varchar(255)	YES		NULL	
time	time(6)	YES		NULL	
quiz_quiz_id	int	YES	MUL	NULL	
user_userid	int	YES	MUL	NULL	

11. **Question**

Field	Type	Null	Key	Default	Extra
question_id	int	NO	PRI	NULL	auto_increment
question_marks	int	NO		NULL	
question_text	varchar(255)	YES		NULL	
quiz_id	int	YES	MUL	NULL	

12. Options

Field	Type	Null	Key	Default	Extra
option_id	int	NO	PRI	NULL	auto_increment
correct	bit(1)	NO		NULL	
option_name	varchar(255)	YES		NULL	
question_id	int	YES	MUL	NULL	

13. Media

Field	Type	Null	Key	Default	Extra
media_id	int	NO	PRI	NULL	auto_increment
media_link	varchar(255)	YES		NULL	
question_id	int	YES	MUL	NULL	

14. Attempts

Field	Type	Null	Key	Default	Extra
quiz_attempt_id	int	NO	PRI	NULL	auto_increment
attempt_date	date	YES		NULL	
attempt_number	int	NO		NULL	
end_time	time(6)	YES		NULL	
quiz_quiz_id	int	YES	MUL	NULL	
result_result_id	int	YES	UNI	NULL	
user_userid	int	YES	MUL	NULL	

15. Quiz_Attempt_Option

Field	Type	Null	Key	Default	Extra
quiz_attempt_id	int	NO	MUL	NULL	
option_id	int	NO	MUL	NULL	

16. Result

Field	Type	Null	Key	Default	Extra
result_id	int	NO	PRI	NULL	auto_increment
result_score	int	NO		NULL	

17. **Announcement**

Field	Type	Null	Key	Default	Extra
announcement_id	int	NO	PRI	NULL	auto_increment
content	varchar(255)	YES		NULL	
date	date	YES		NULL	
time	time(6)	YES		NULL	
course_id	int	YES	MUL	NULL	
user_userid	int	YES	MUL	NULL	

3.3. Normalization:

In the context of normalization:

1. First Normal Form (1NF):

- Each table in our relational schema has a unique primary key attribute, ensuring that each record can be identified uniquely.
- Every piece of data in our tables is atomic, meaning it cannot be further divided. Therefore, our tables are already in 1st normal form.

2. Second Normal Form (2NF):

- All non-prime attributes (attributes not part of the primary key) are fully dependent on the primary key.
- We have eliminated partial dependencies in our relations, ensuring that each non-prime attribute is directly related to the primary key. Hence, our tables are already in 2nd normal form.

3. Third Normal Form (3NF):

- Our relations do not contain non-prime attributes that depend on other non-prime attributes, which, in turn, are dependent on the primary key.
- There are no transitive dependencies; all non-prime attributes are fully dependent on the primary key.
- Therefore, our relations are already in 3rd normal form.

In summary, our database design meets the criteria of 1NF, 2NF, and 3NF, indicating a high level of normalization, and no further normalization is deemed necessary.

4. Implementation:

4.1. Technologies Used:

- **Frontend:** Angular, HTML, SCSS, TypeScript
- **Backend:** Spring Boot, Java
- **Database:** MySQL
- **IDE:** Visual Studio Code (frontend), IntelliJ IDE (backend), and MySQL Workbench (Database)

4.2. How to run?

Below are the instructions to run the project.

Step 1: Download the project

The project can be found in the following places

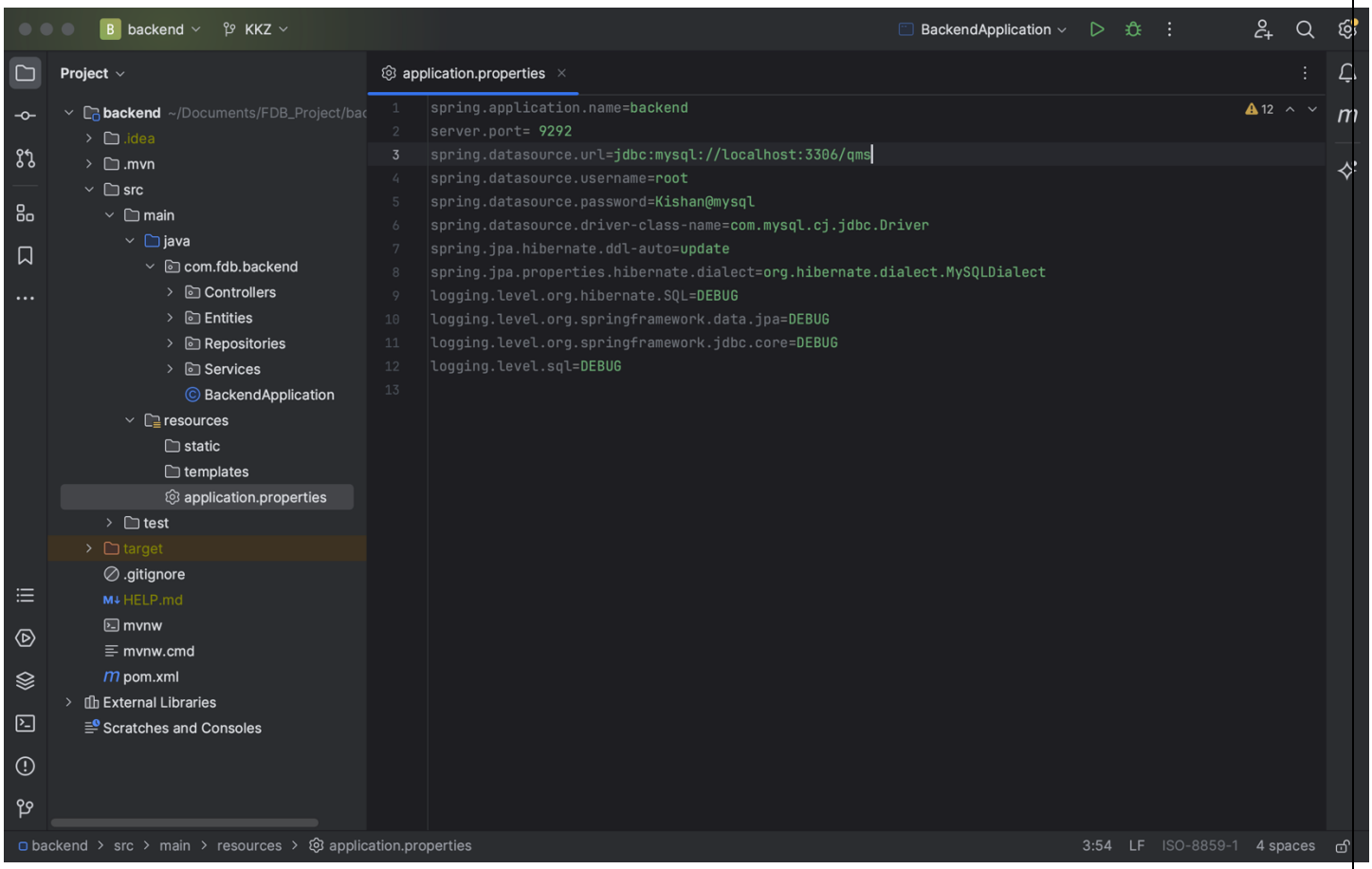
- Canvas Submission
- GitHub: https://github.com/Kishan-Kumar-Zalavadia/FDB_Quiz_Management_System

Step 2: Open the project

- The main folder contains 2 subfolders – frontend and backend.
- Open the frontend folder in VS Code and the backend folder in IntelliJ or Eclipse IDE's

Step 3: Setup the project

- In the application.properties file change the username and password according to your local SQL credentials. (As shown below)

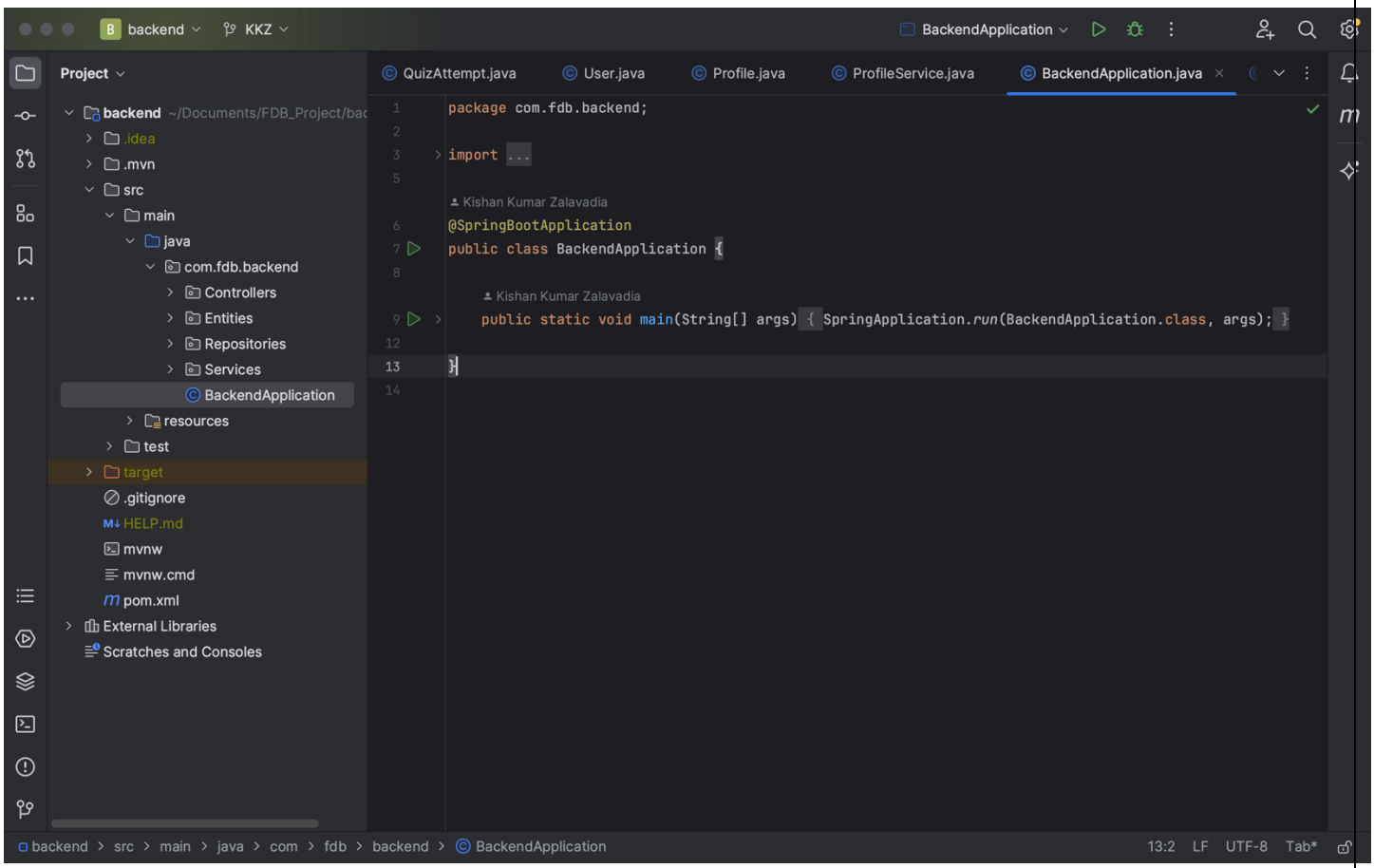


Step 4: Setup Database

- Create a MySQL database named '*qms*' in your local system.
- Use the command 'create database QMS' to do so.

Step 5: Run the backend

- Run the '*BackendApplication.java*' file.
- Click on the triangle sign beside '*public class BackendApplication*' to run the Java file. (As shown below)



Step 6: Load sample data

- Run the commands given in the '*SampleData.sql*' file.
- NOTE: Make sure to have the Role and Department table filled with data present to *'SmapleData.sql'*.
- NOTE: The queries will not run until the backend is running without any errors.

Step 7: Run fronted

- In the terminal type the command *'ng serve'* to run the frontend.
- Make sure to be inside the 'frontend' folder while entering the command'.

Step 8: View the project

- To view the project, go to any browser and enter the following URL. (Project URL: http://localhost:4200/login)
- NOTE: For these credentials to work, you must have loaded the SQL data as shown in step 6.

4.3. Code Snippets

Below is a complex query used in the project:

- SELECT MAX(qa.attemptNumber) FROM QuizAttempt qa WHERE qa.user.userID = :userId AND qa.quiz.quizId = :quizId"

5. Results and Discussions:

5.1. Functionalities:

- Instructors and Students can create accounts.
- Instructors and Students can log in to their respective accounts using email and password.
- Instructors can create their own courses.
- Students can enroll in courses created by instructors.
- Instructors and Students can give feedback about this application.
- Instructors can create several quizzes for their courses.
- For each quiz, the instructor can add multiple questions each with a different weightage (marks)
- For each question, the instructor can add multiple options. There is no limit to how many options can a question have.
- For each question, the instructor can also add multiple media (Images)
- Students can take quizzes for those courses in which they have enrolled.
- Students can see the number of attempts left for each quiz.
- Students can view their marks after each quiz.
- Students can see the attempt histories which include marks, time, and date of attempt.
- Instructors can see the score of each student for the quiz that they have created. Instructors can also see each score of different attempts of all the students.
- Students can give feedback for each quiz separately.

- Instructors can see all the feedback given by students for the quiz that they have created.
- Instructors can create announcements for courses separately.
- Students can see announcements for the course separately including data and time of announcement.

5.2. Assumptions:

- Role and Department are already mentioned.
- Only one profile is associated with each user.
- Users can give one or more feedback.
- Each user has only one role.
- One course can belong to multiple departments and vice versa.
- Users can give one or more feedback on a quiz.
- Each profile is associated with only one department.
- Each Quiz belongs to a course.

5.3. Challenges:

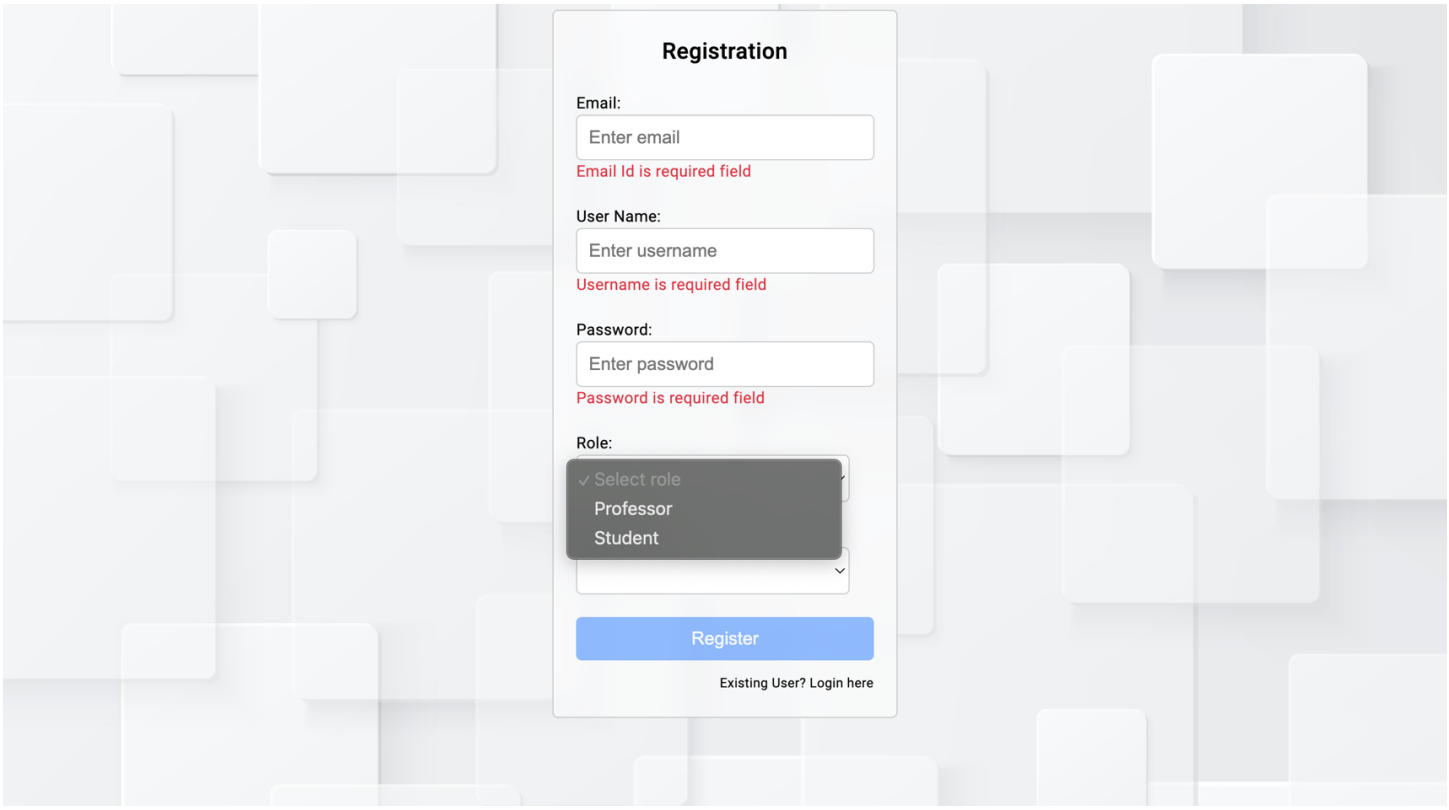
One of the major challenges faced during the project is the infinite repetition of data during cardinalities.

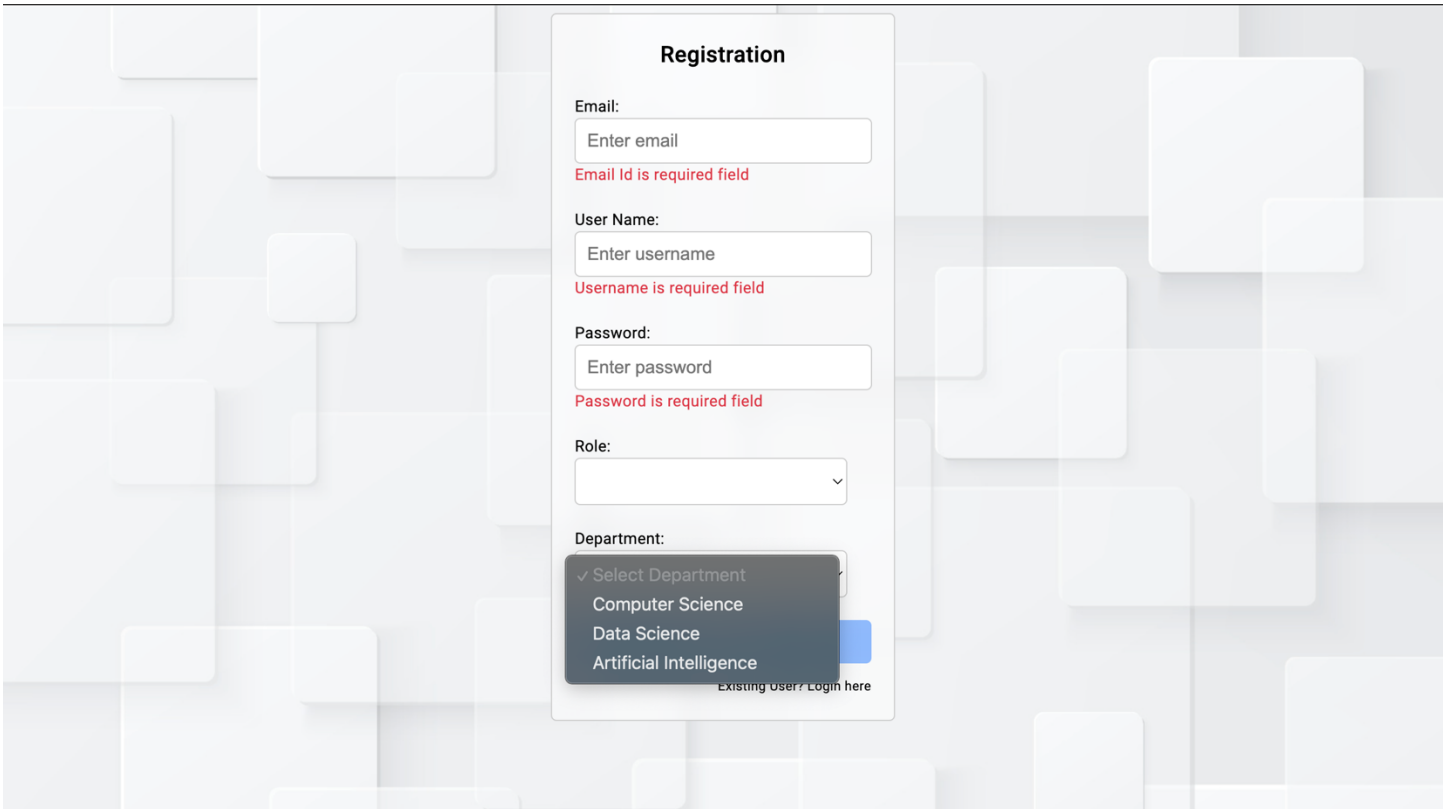
For example, the user table has a one-to-one relationship with the profile table. Because we used spring boot, we can have a profile object in the user object and a user object in the profile object. In that case, the data gets trapped infinitely when reading the data from the database.

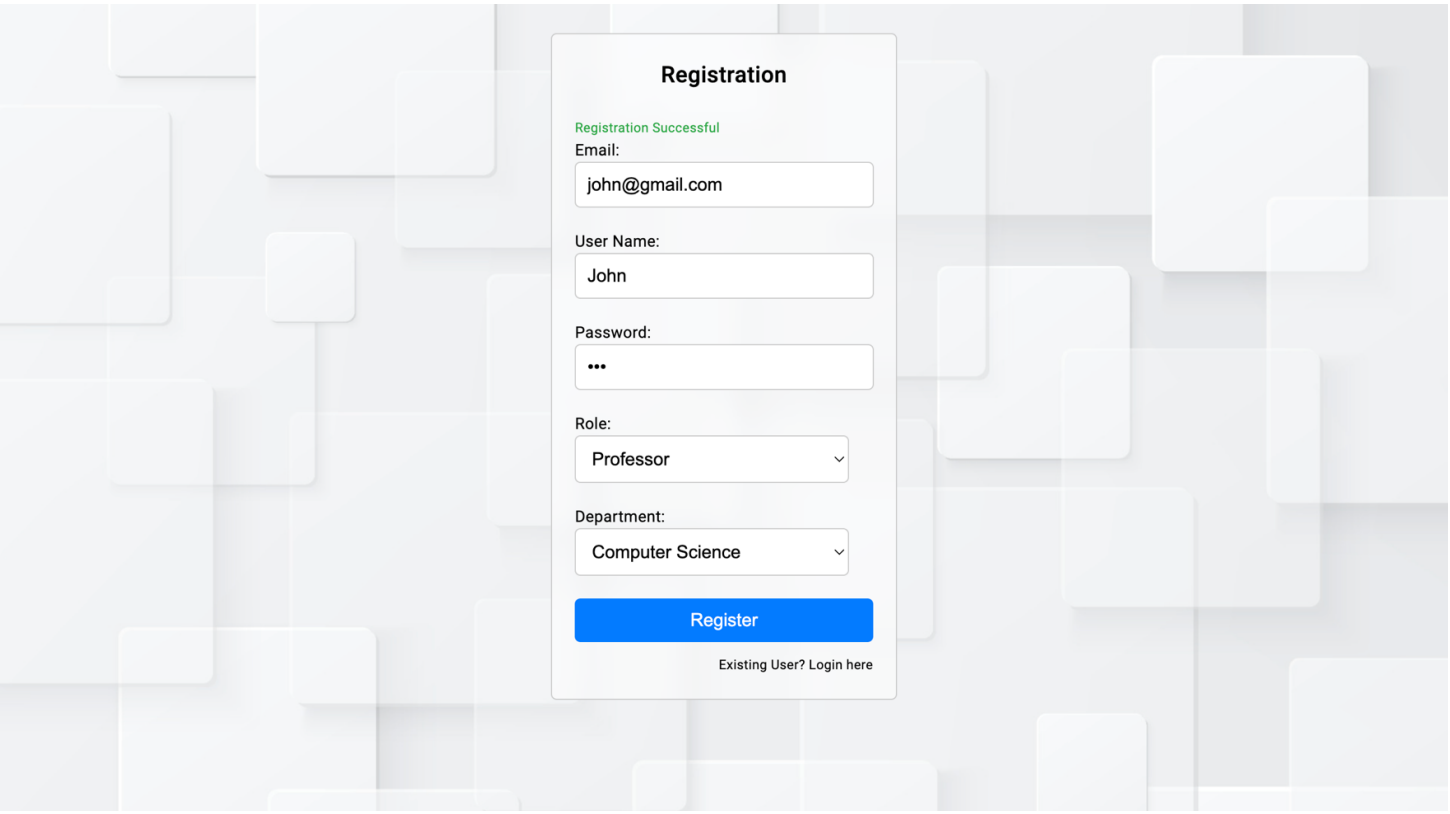
To overcome this challenge we have used a spring boot annotation '@JSONIgnore' which ignores one of the objects to avoid an infinite loop.

5.4. Application Walkthrough:

Register Page:

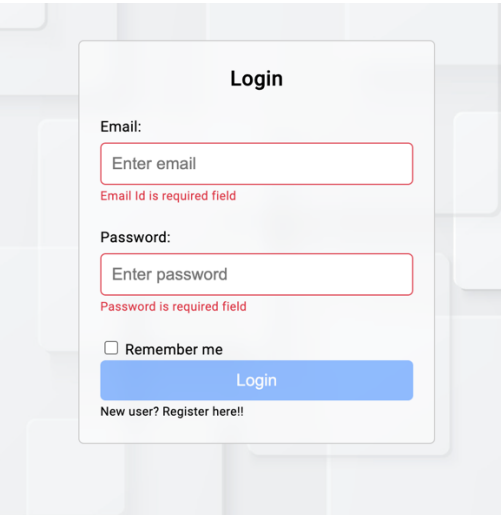

Registration
Email:
Enter email
Email Id is required field
User Name:
Enter username
Username is required field
Password:
Enter password
Password is required field
Role:
Select role
Professor
Student
Register
Existing User? Login here


Registration
Email:
Enter email
Email Id is required field
User Name:
Enter username
Username is required field
Password:
Enter password
Password is required field
Role:
Department:
Select Department
Computer Science
Data Science
Artificial Intelligence
Existing User? Login here



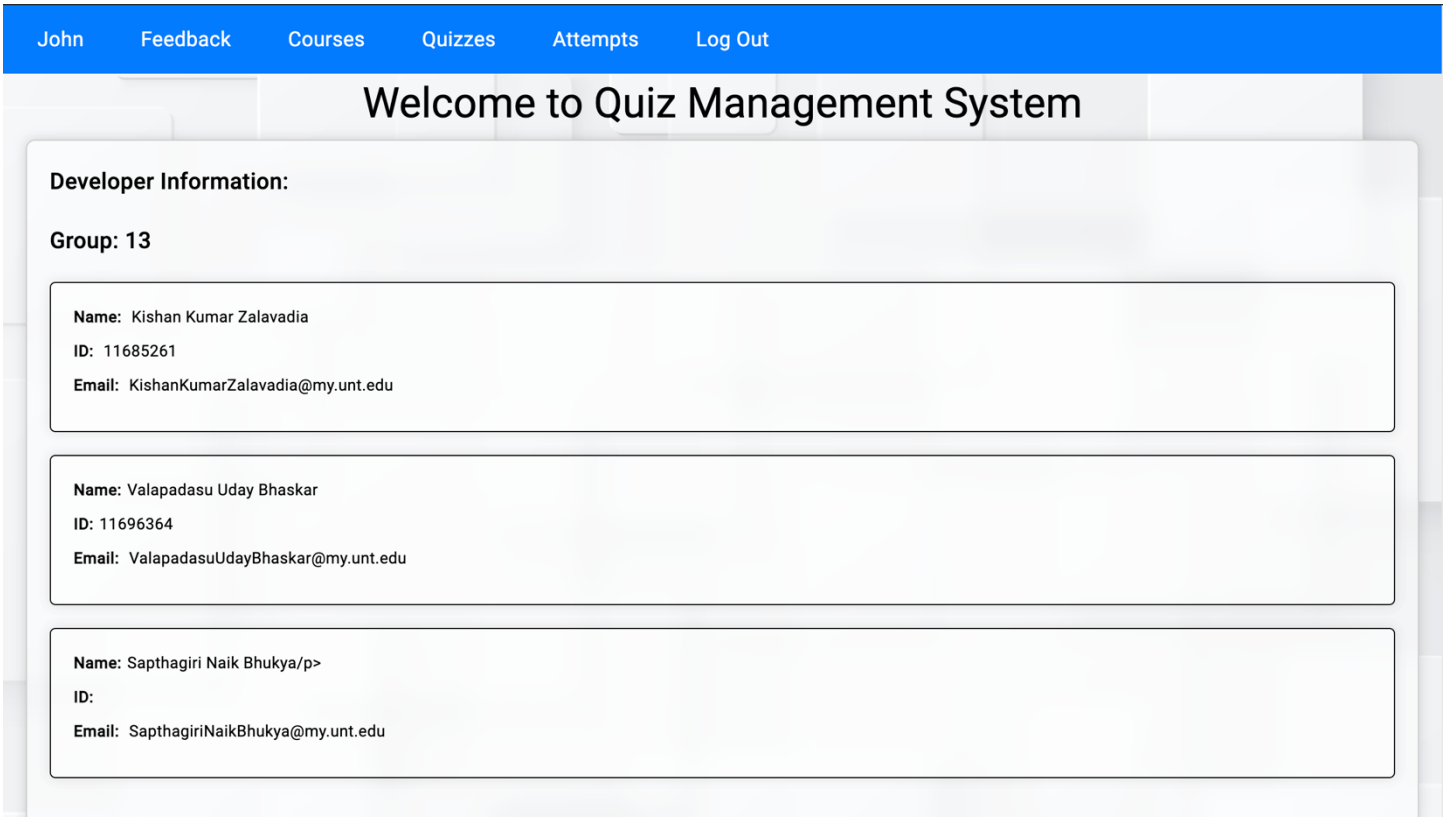
- Users can register as different roles and select the department he/she is in.
- Used is required to fill in all the details, and once registration is successful "Registration Successful" message will be shown in green color.

Login Page:



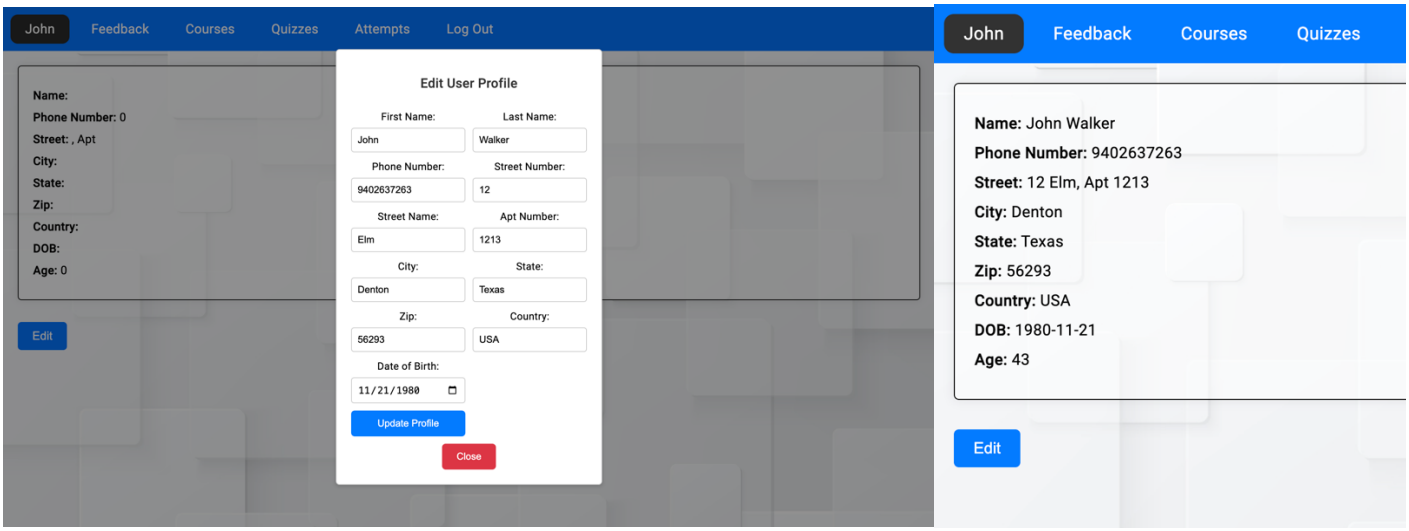
- Once the user has successfully registered, he can log in using the email and password that he used while registering.

Home Page:



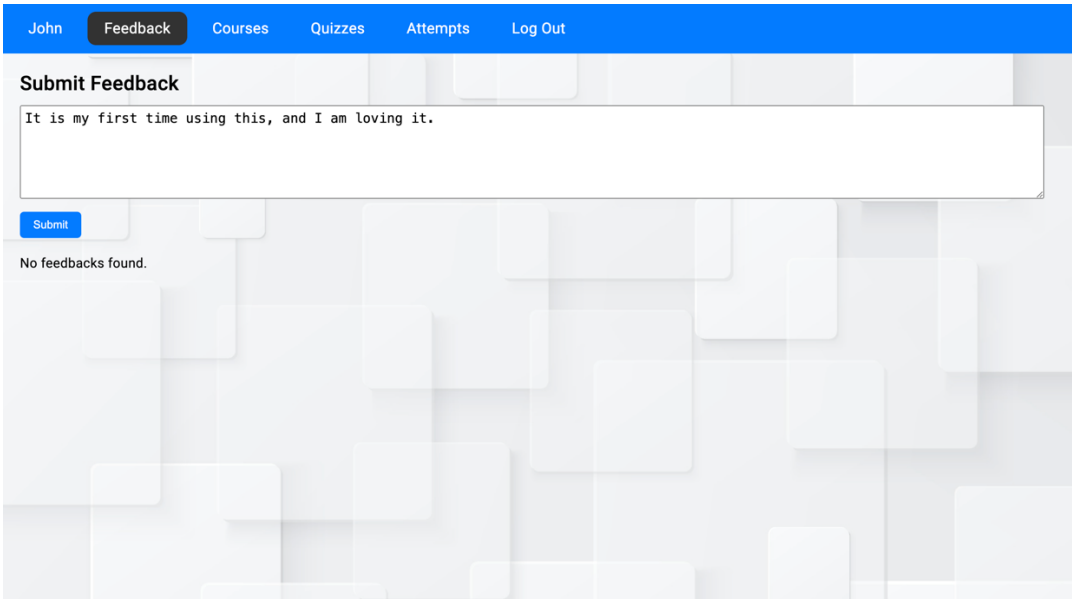
- On the home page user can see his username in the top-left corner.

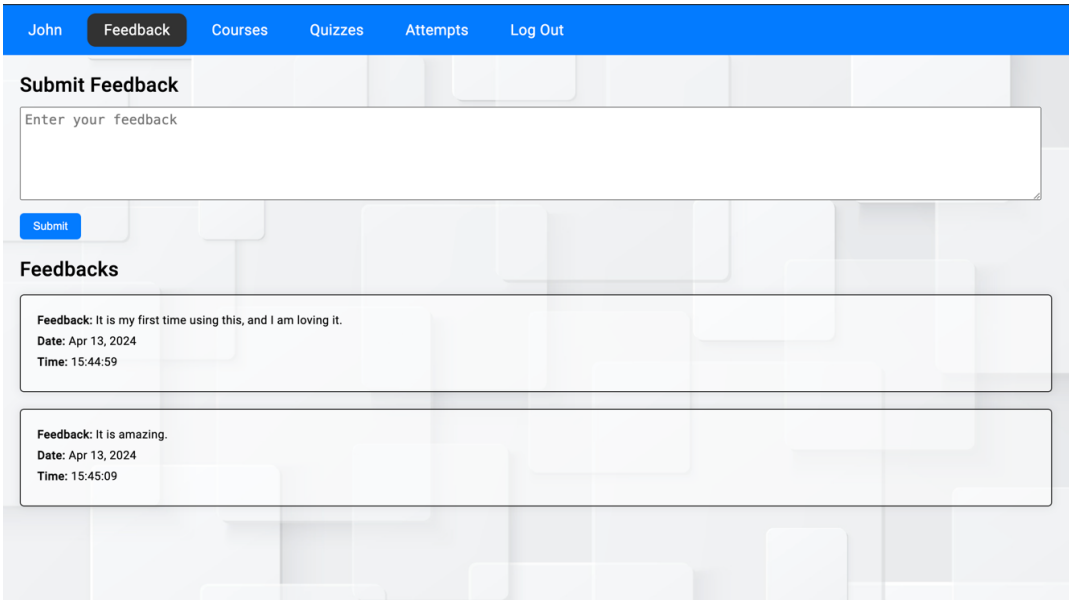
Profile Page:



- Users can edit or update their profile at any time.

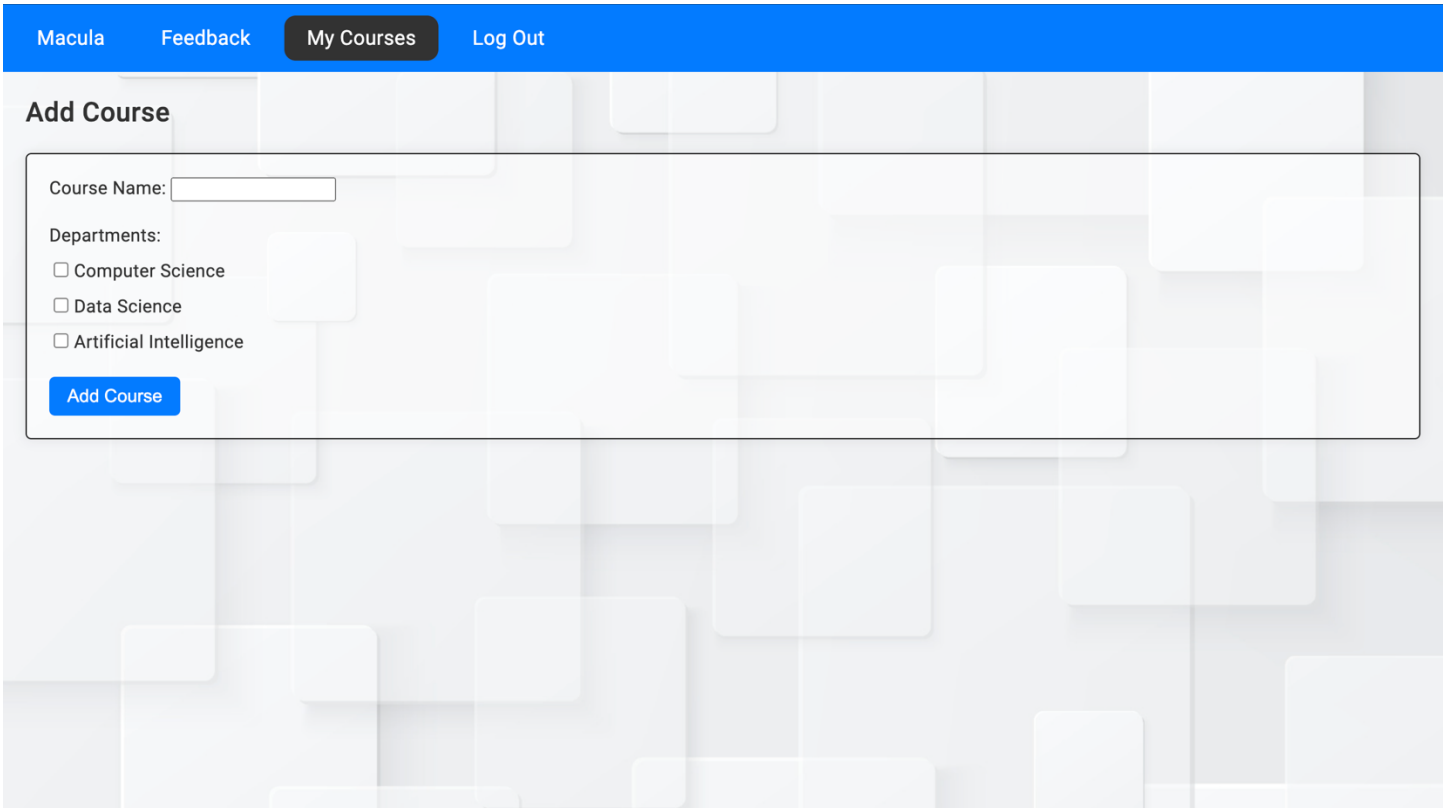
Feedback Page:



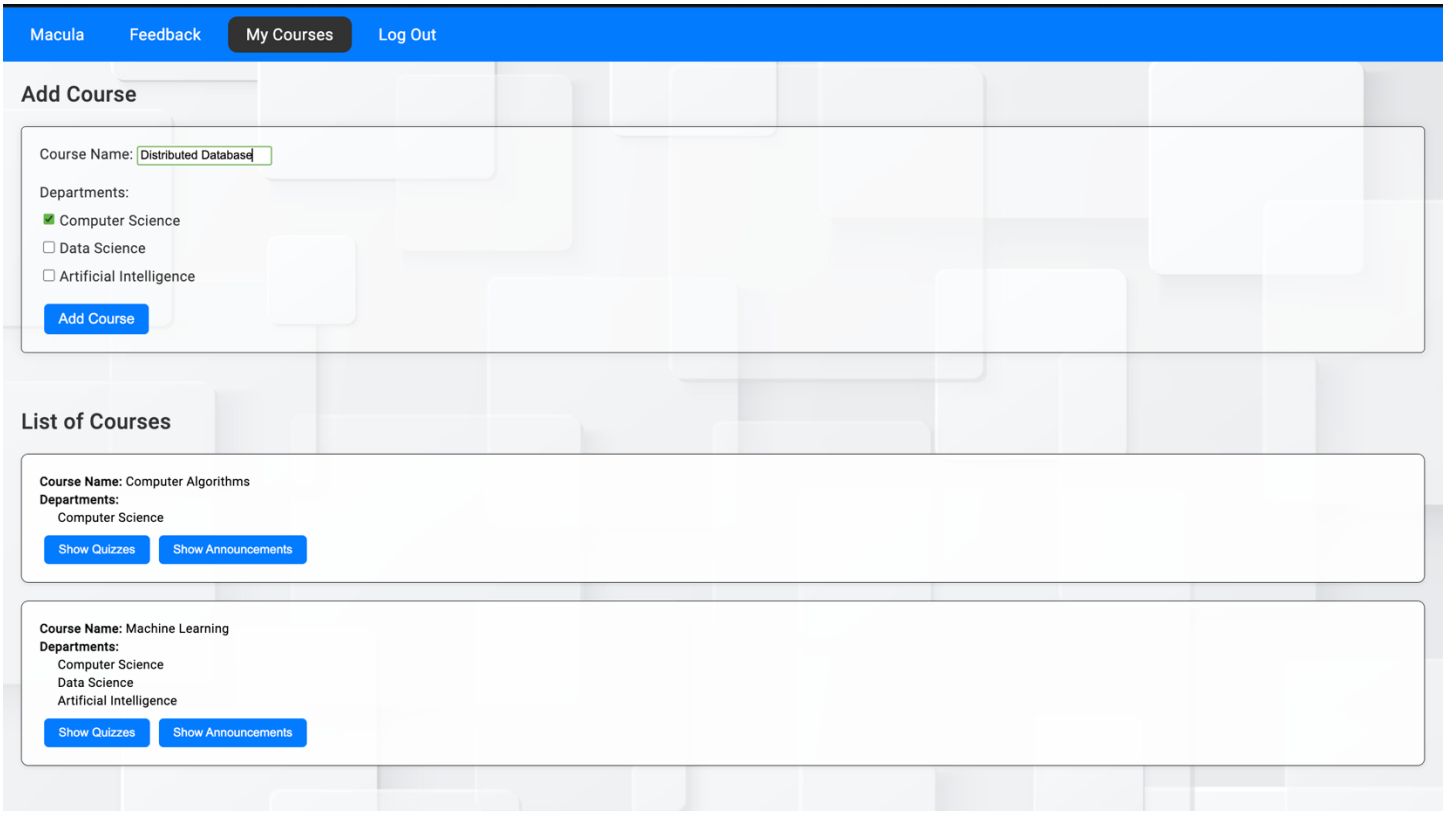


- Users can give feedback and can see all the feedback given by them along with the date and time.

Add Courses (Professor)

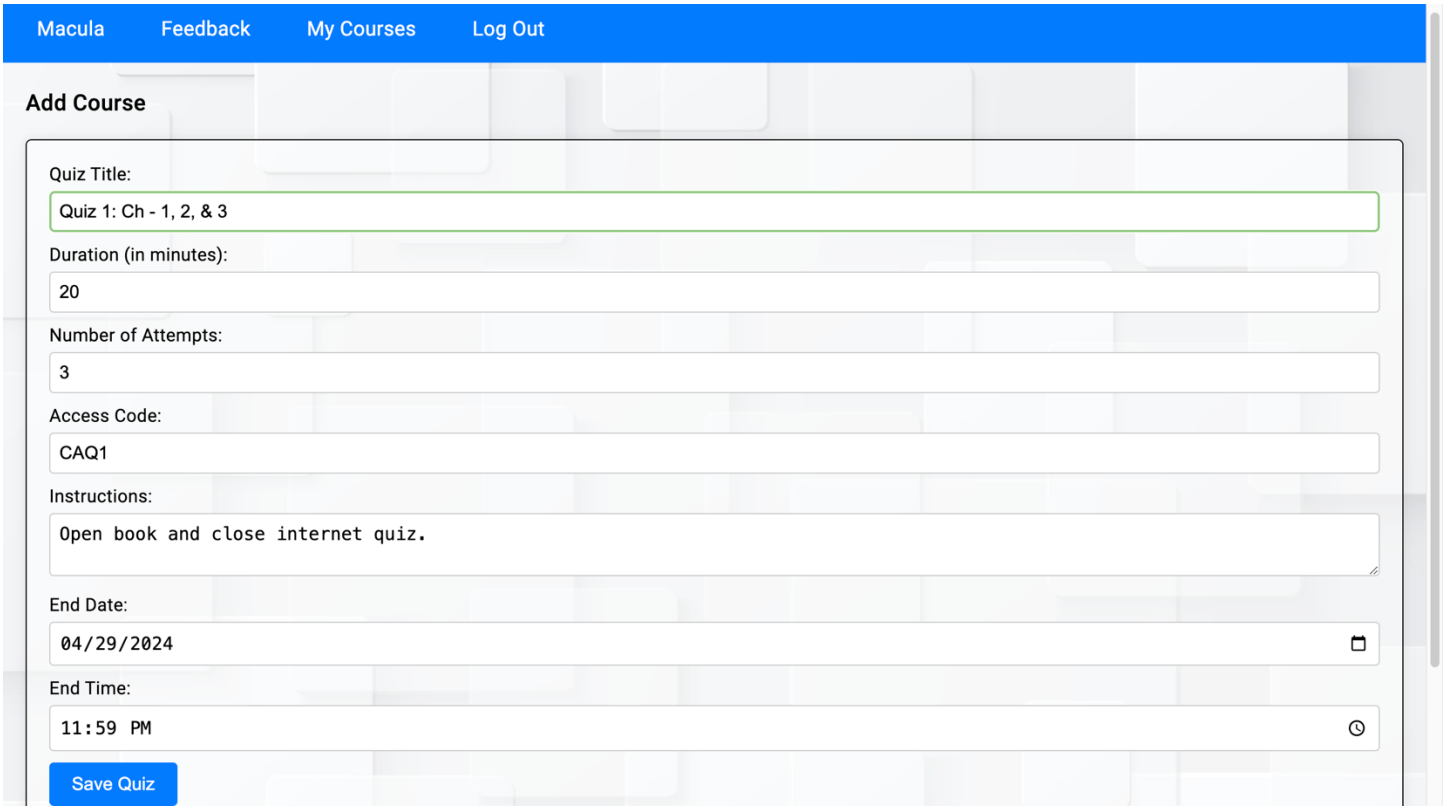


- If the user is a professor, then he/she can add the course and can also choose which department students can take the course.

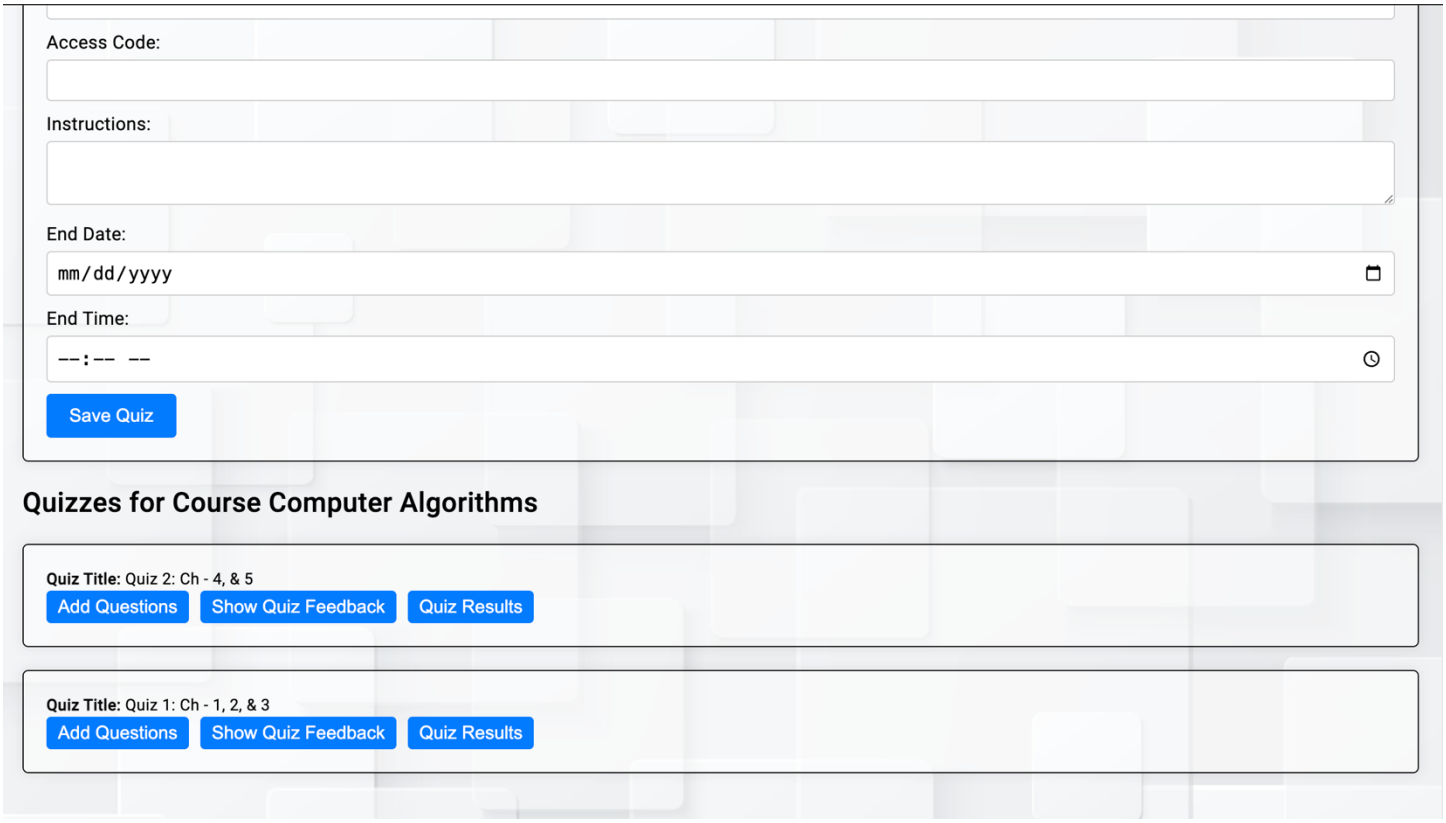


- All the courses which the professor has created can be visible.

Add Quizzes

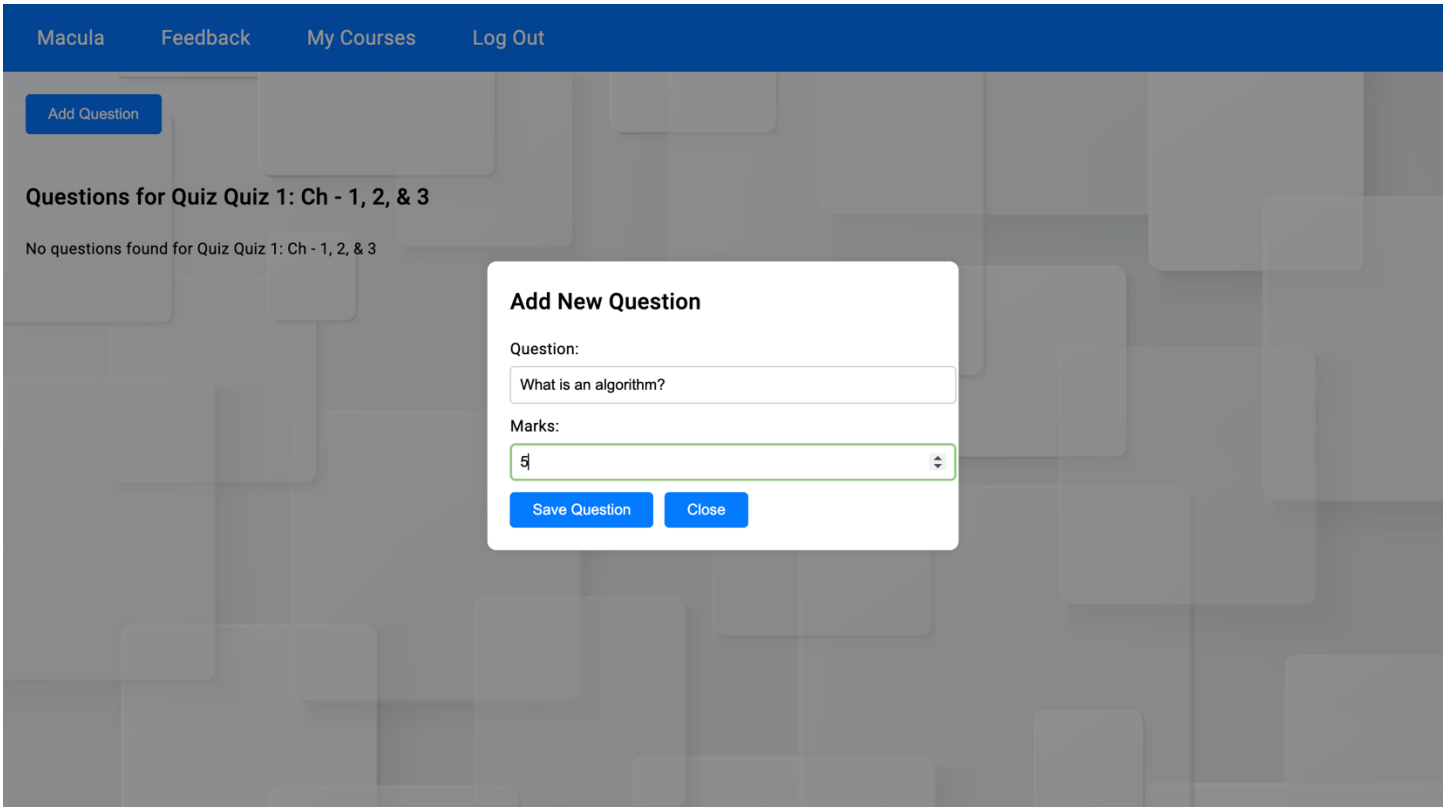


- The professor can create several quizzes for each course.



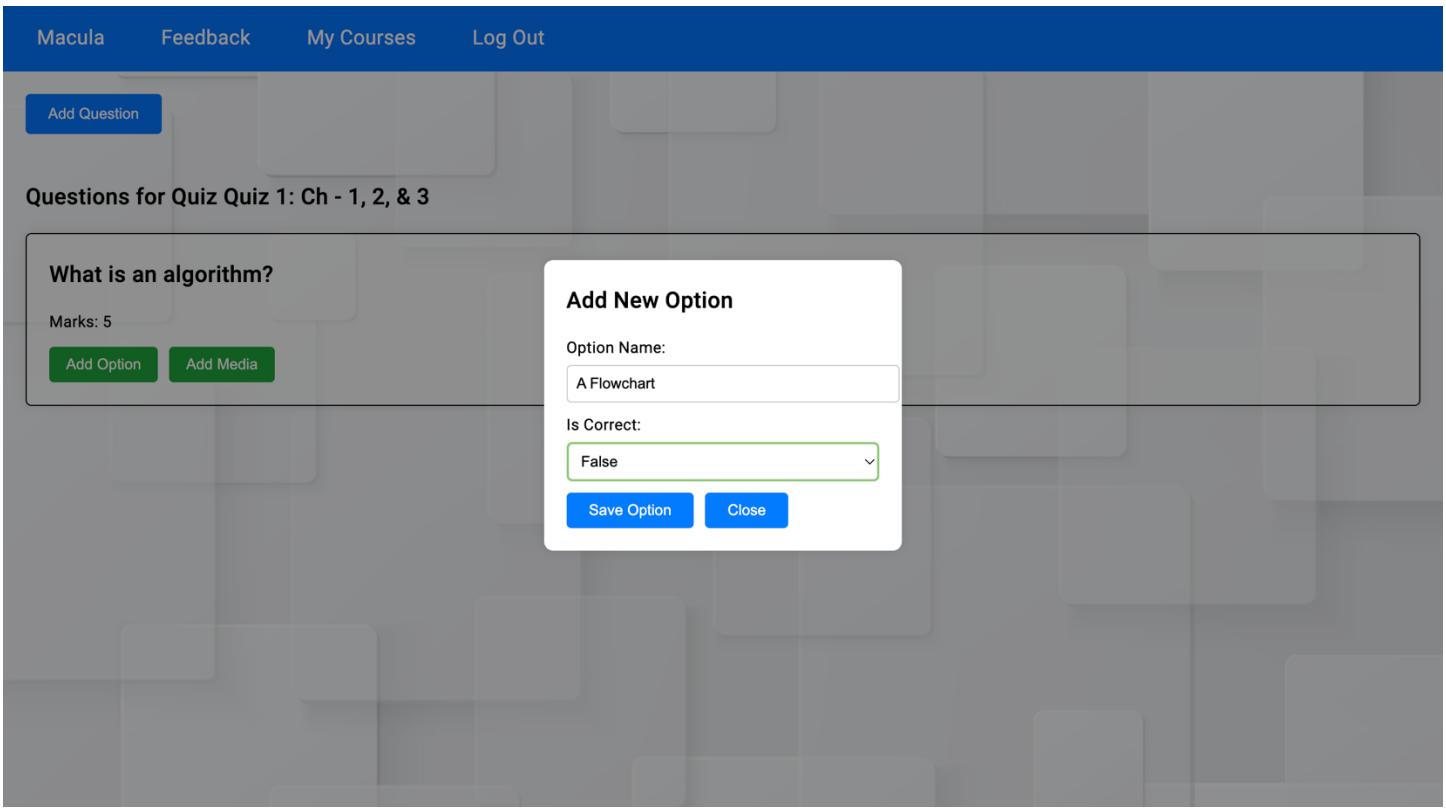
- The professor can see all the quizzes that he created

Add Questions Page



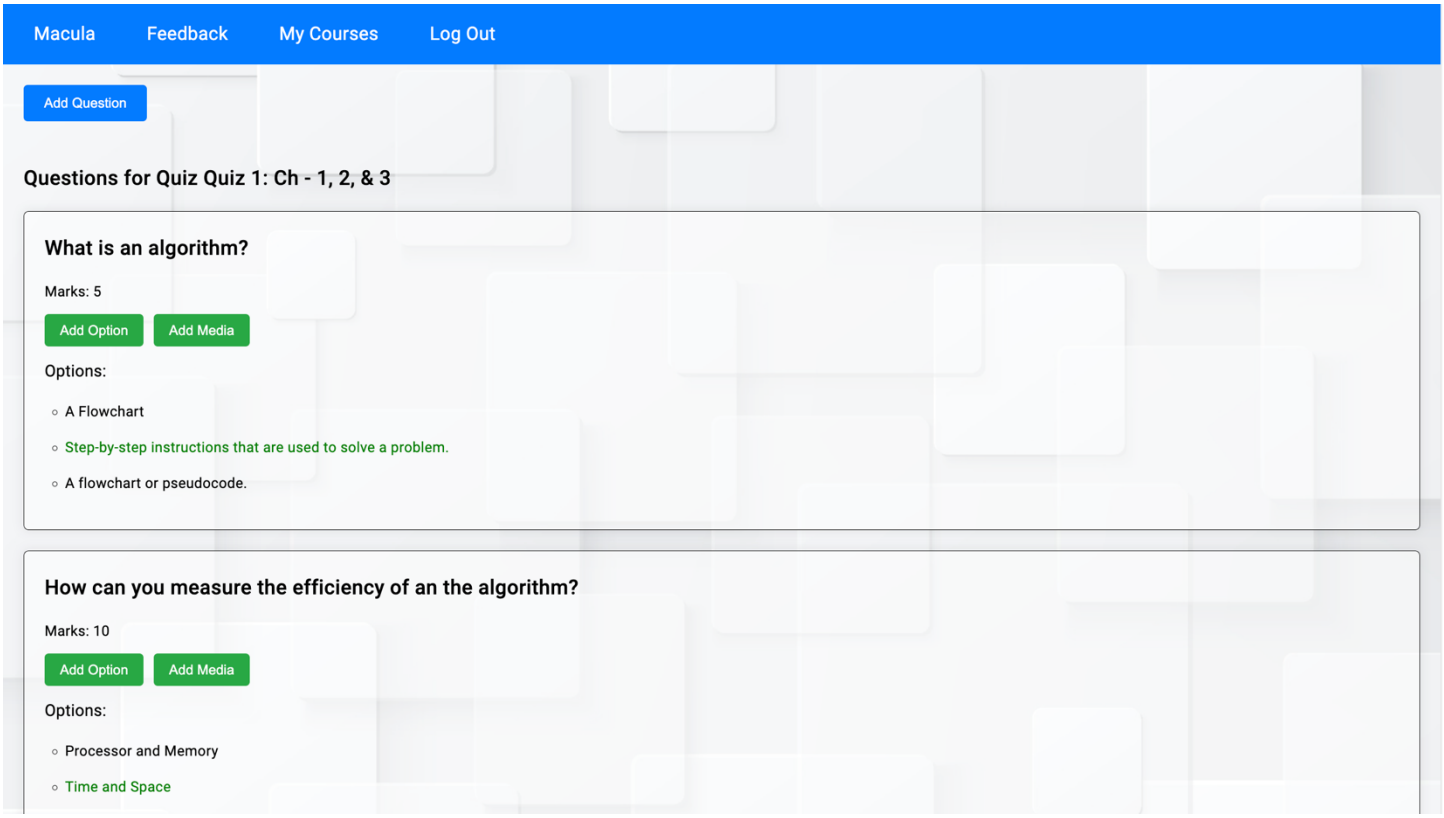
- The professor can add questions and assign marks to each question.

Add Options



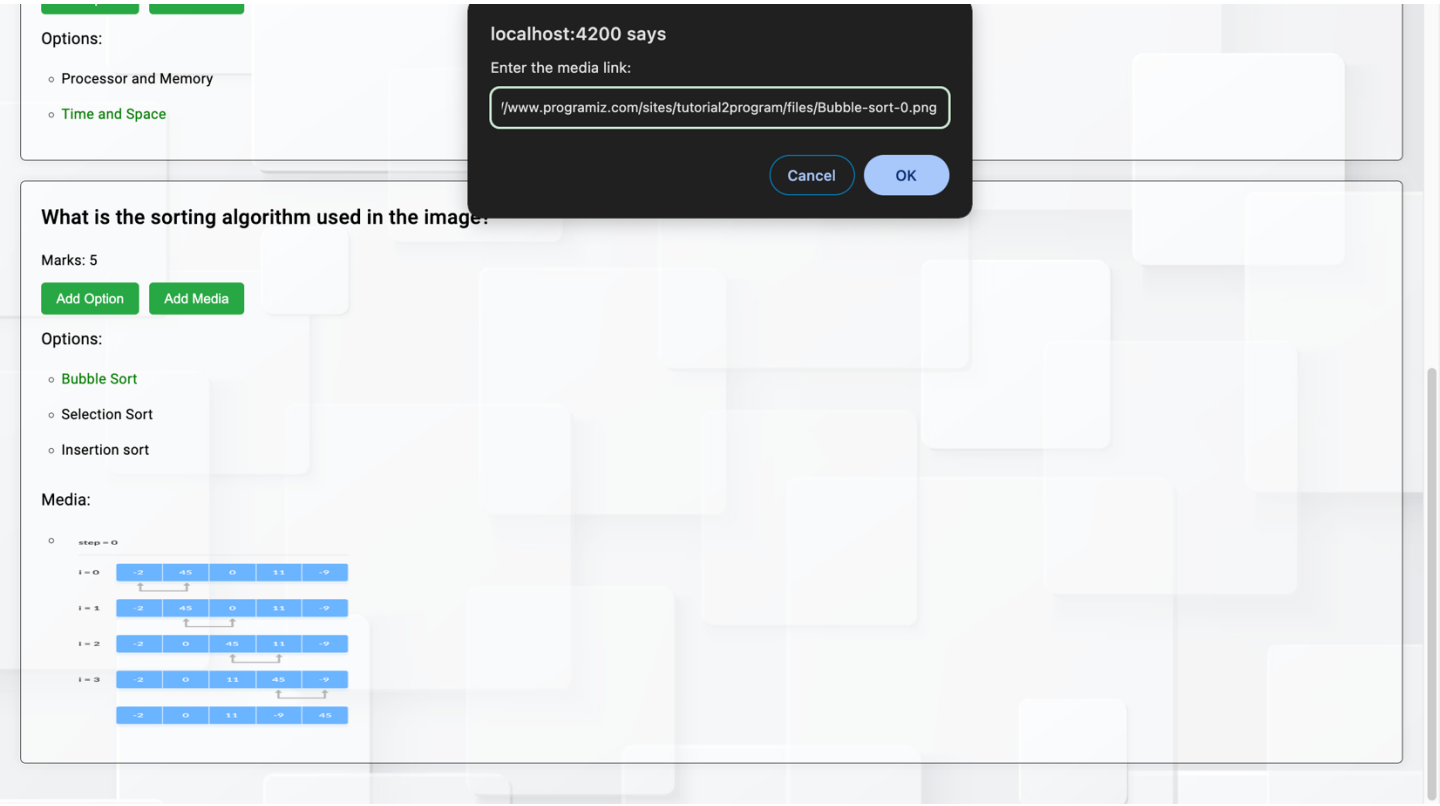
- The professor can add options to the questions.

- There is no limit on how many options to add for each question, that depends on the professor.
- For each option, the professor can decide whether it's the correct answer or not.



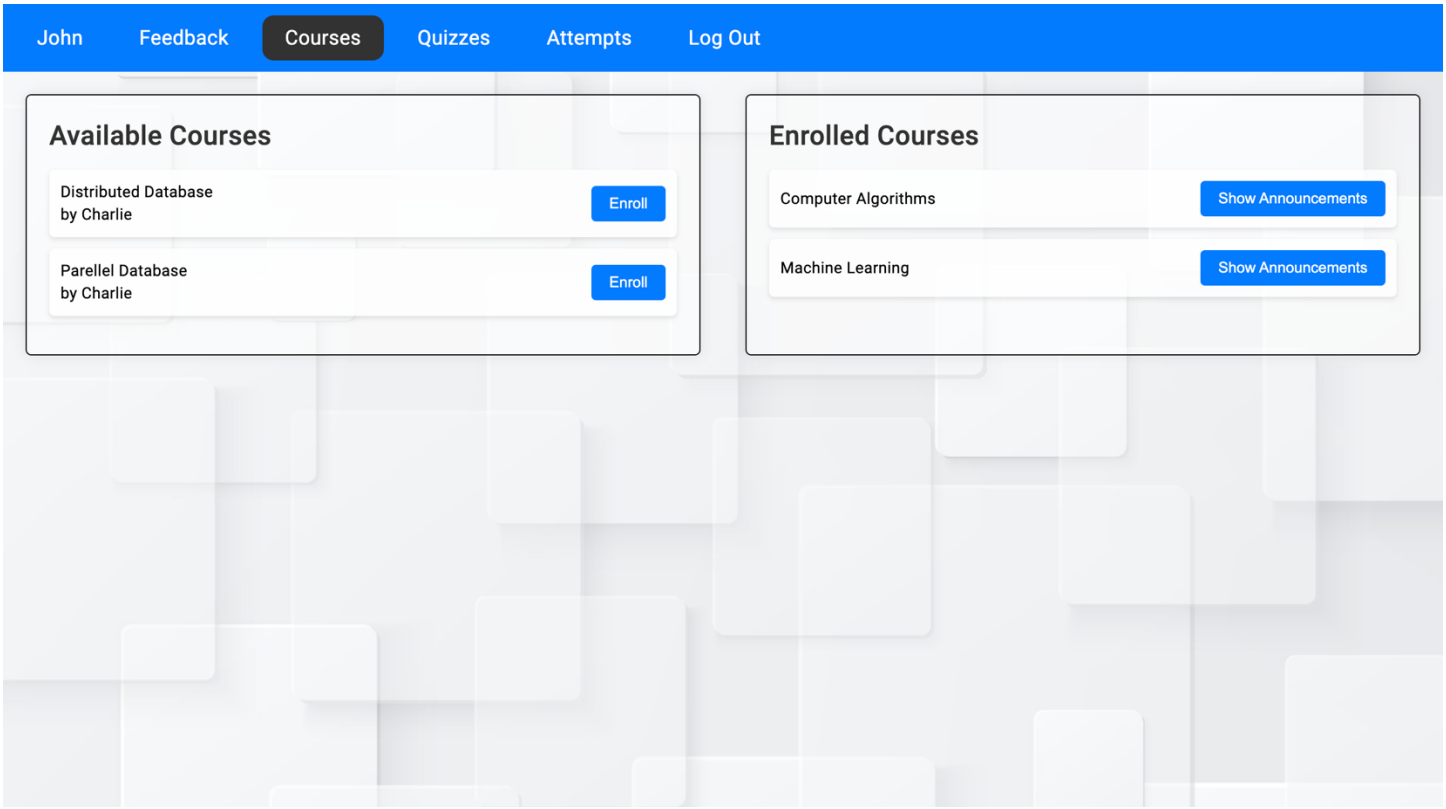
- All the correct answers are visible as green only to the professor.

Add Media



- For each question, the professor can add images.
- There can also be more than 1 image for a question.

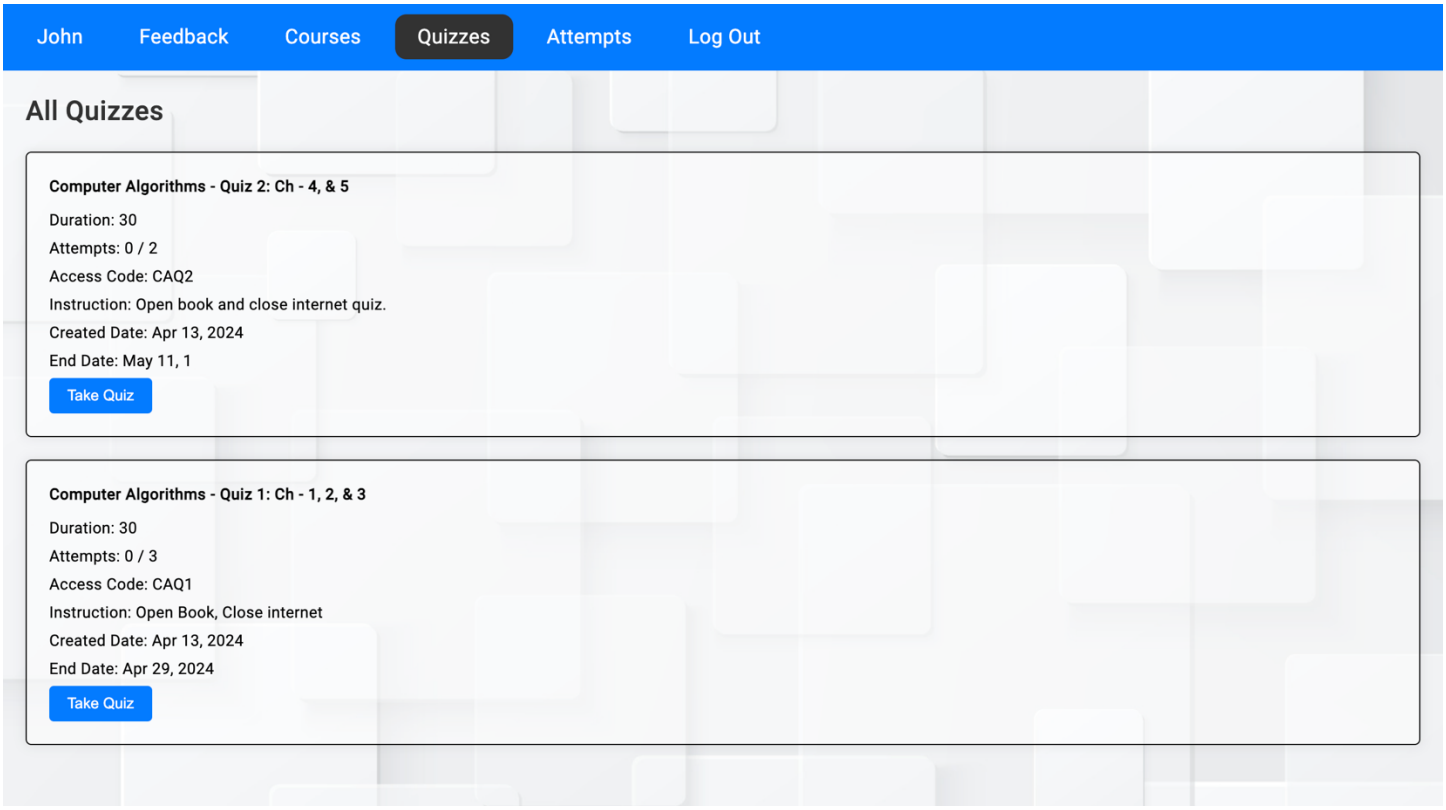
Course Page



- Students can see and enroll in courses.

- The name of the professor is also visible who is taking that particular course.
- All the enrolled courses are displayed on the right, and all courses that can be enrolled are listed on the left.

Quiz Page



- All the quizzes are displayed on the quiz page.
- Only quizzes for the course that the user has enrolled in are visible.

Take Quiz Page

Quiz 1: Ch - 1, 2, & 3

Duration: 30
Number of Attempts: 3
Access Code: CAQ1

What is an algorithm?

Marks: 5

- (●) A Flowchart
- ( ) Step-by-step instructions that are used to solve a problem.
- ( ) A flowchart or pseudocode.

How can you measure the efficiency of an the algorithm?

Marks: 10

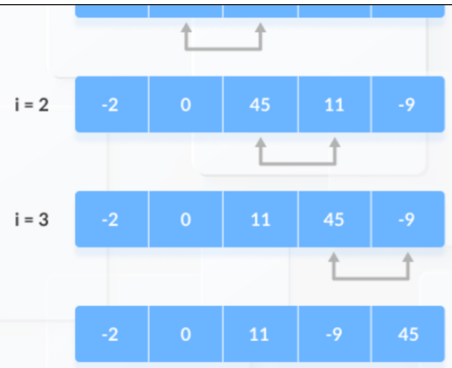
- ( ) Processor and Memory
- (●) Time and Space

What is the sorting algorithm used in the image?

Marks: 5

step = 0

- Students can take the quiz



- ( ) Bubble Sort
- (●) Selection Sort
- ( ) Insertion sort

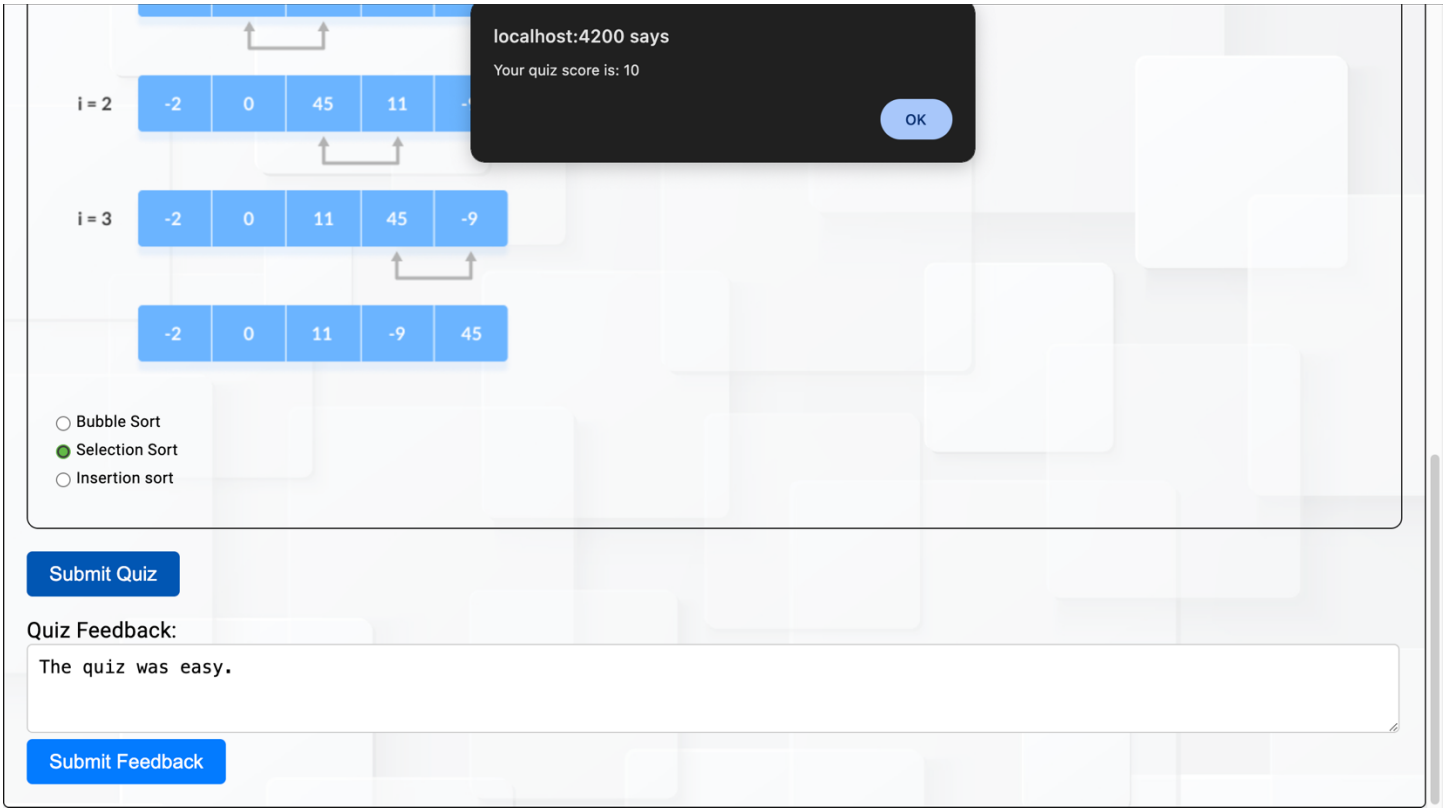
Submit Quiz

Quiz Feedback:

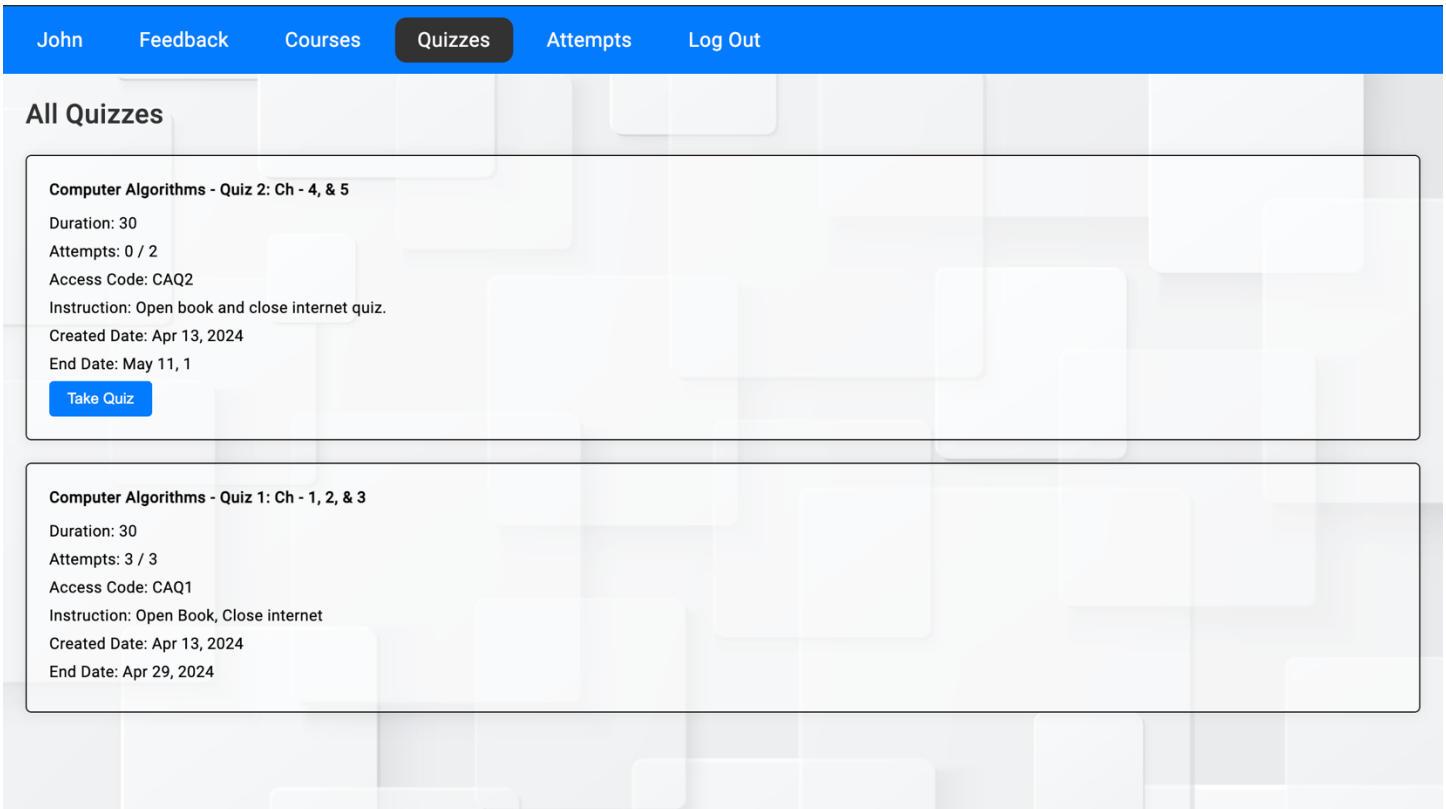
The quiz was easy.

Submit Feedback

- Each student can also give feedback for a particular quiz before submitting the quiz.

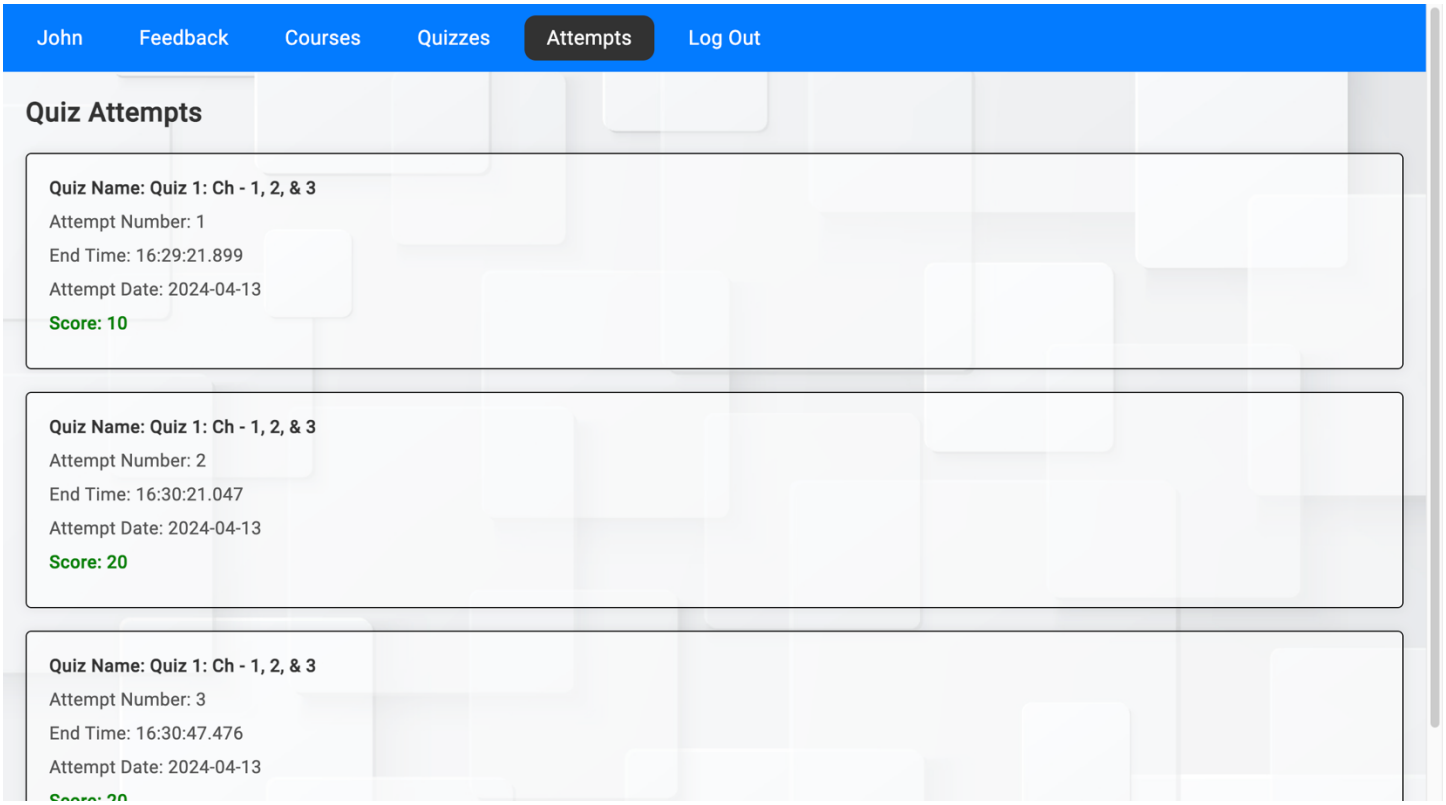


- Once the student submits the quiz, he/she can see the score.



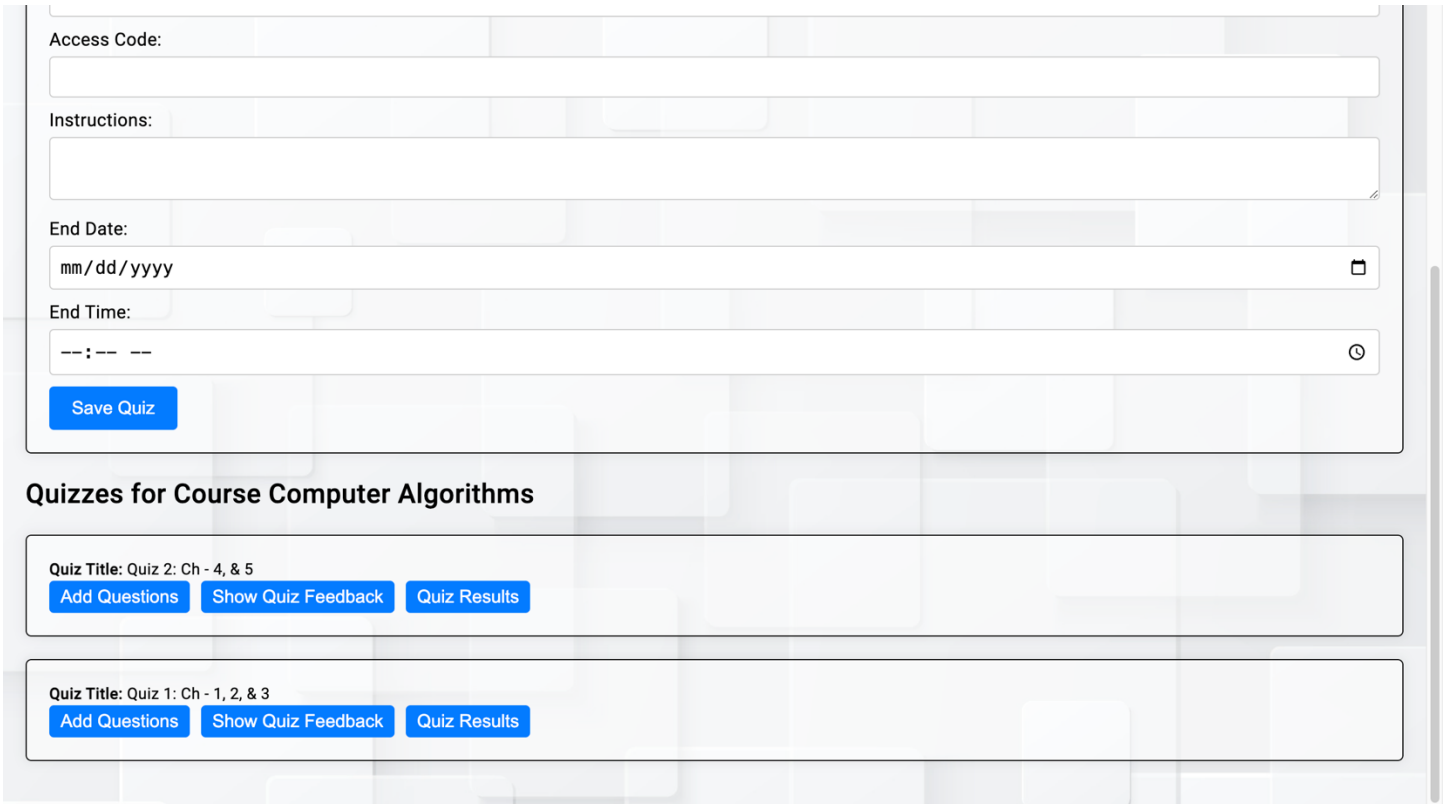
- Students can also see the number of attempts remaining for the quiz.
- If all the attempts are completed then the "Take Quiz" button is not visible, which means the student cannot attempt that quiz.

Attempt History Page

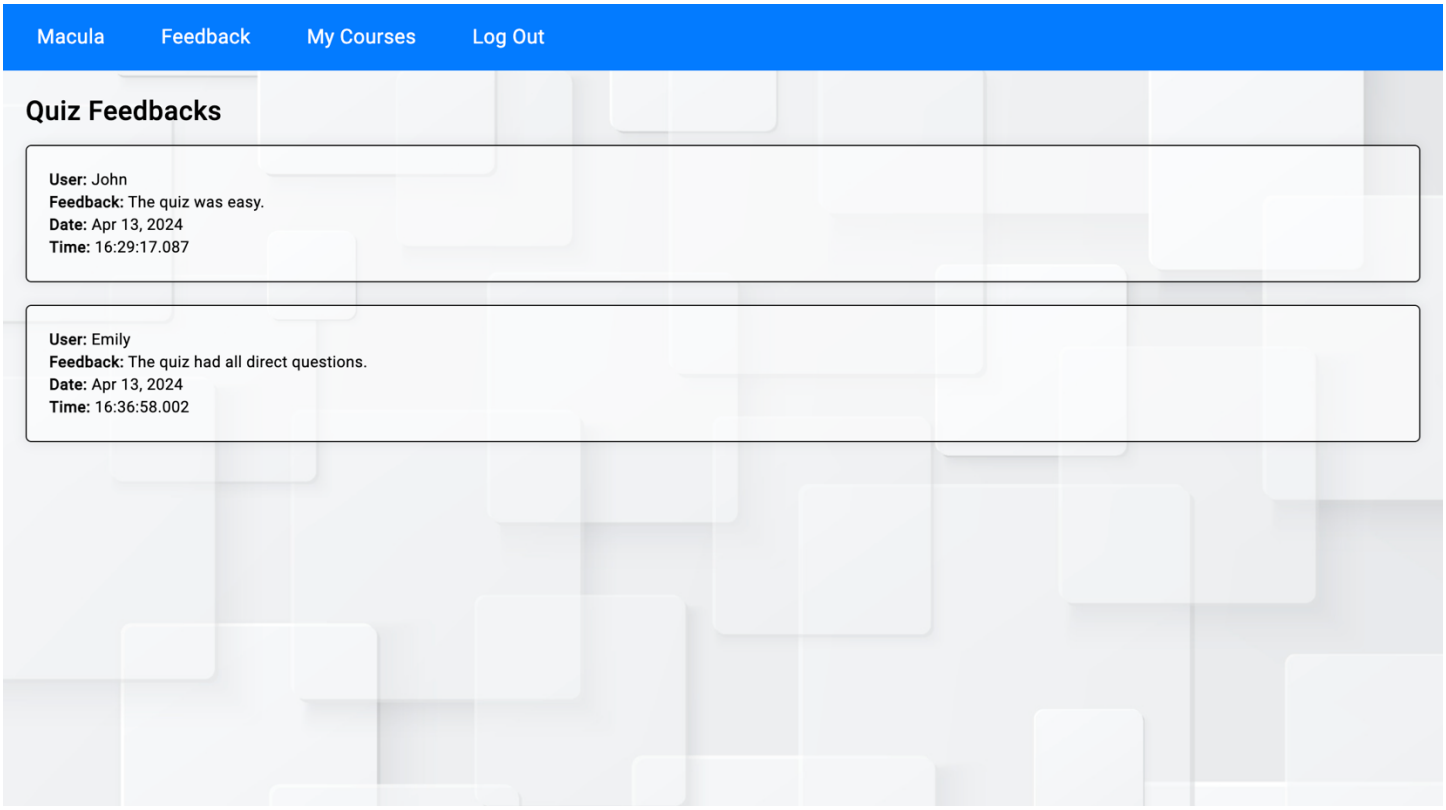


- All the attempts made by the student are visible on the attempts page.
- Data, time, and score of that particular attempt are displayed.

View Quiz Feedbacks

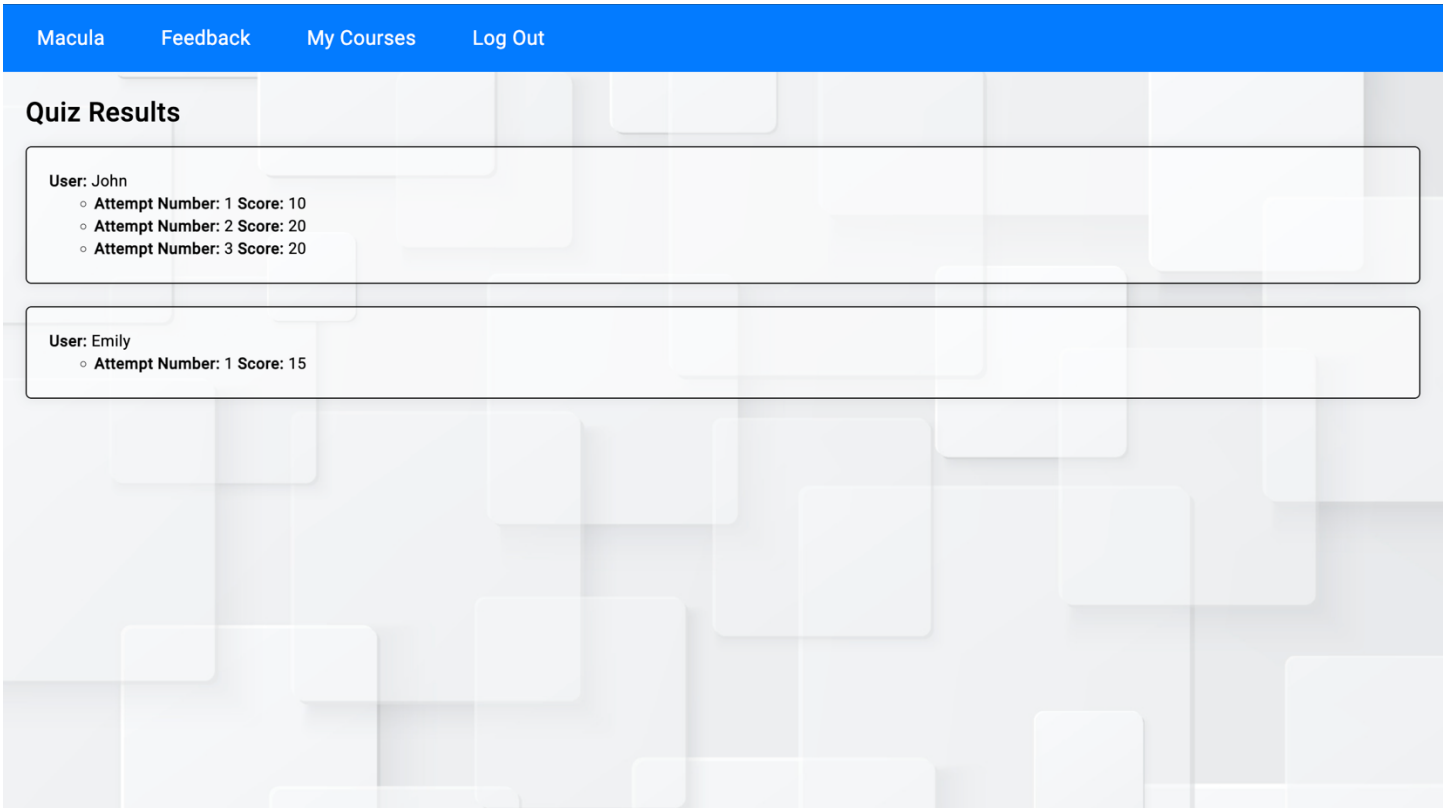


- The professor can click on the "Show Quiz Feedback" button to view all the feedback given by different students as shown below.



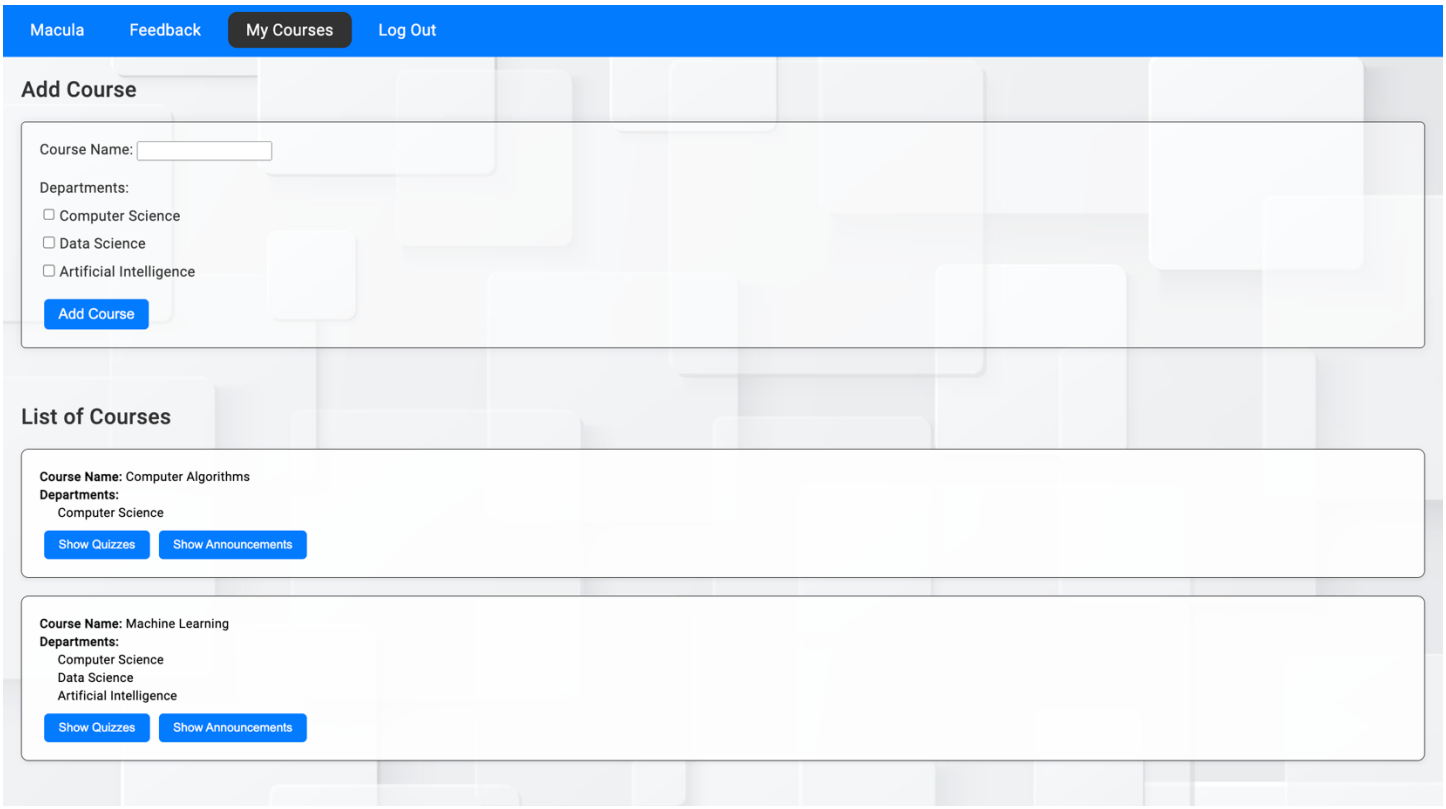
- The professor can see the feedback, who has given that feedback, and also the date and time on which the feedback is given.

Quiz Results

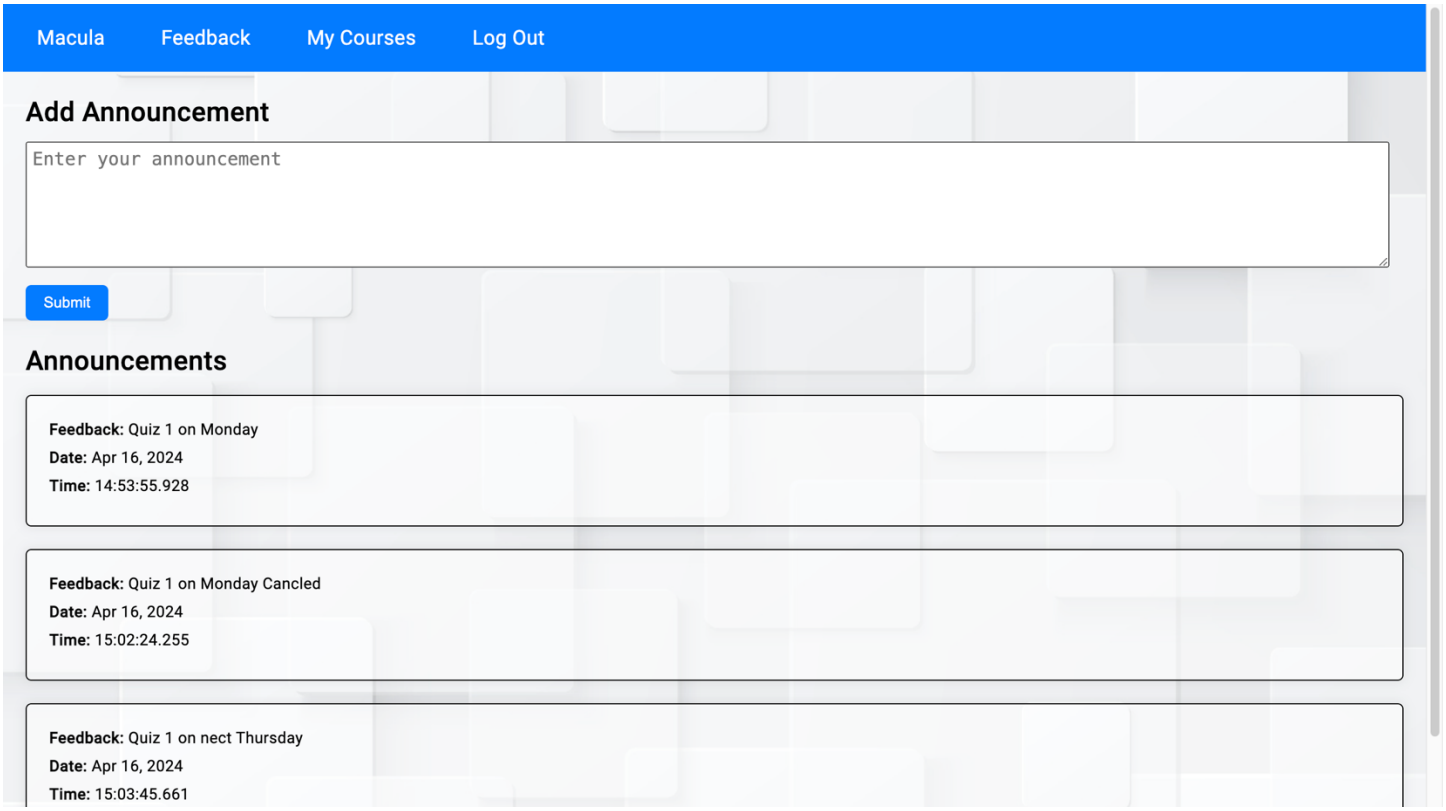


- The professor can also see all the student's quiz results along with all the attempt scores separately.

Add announcement

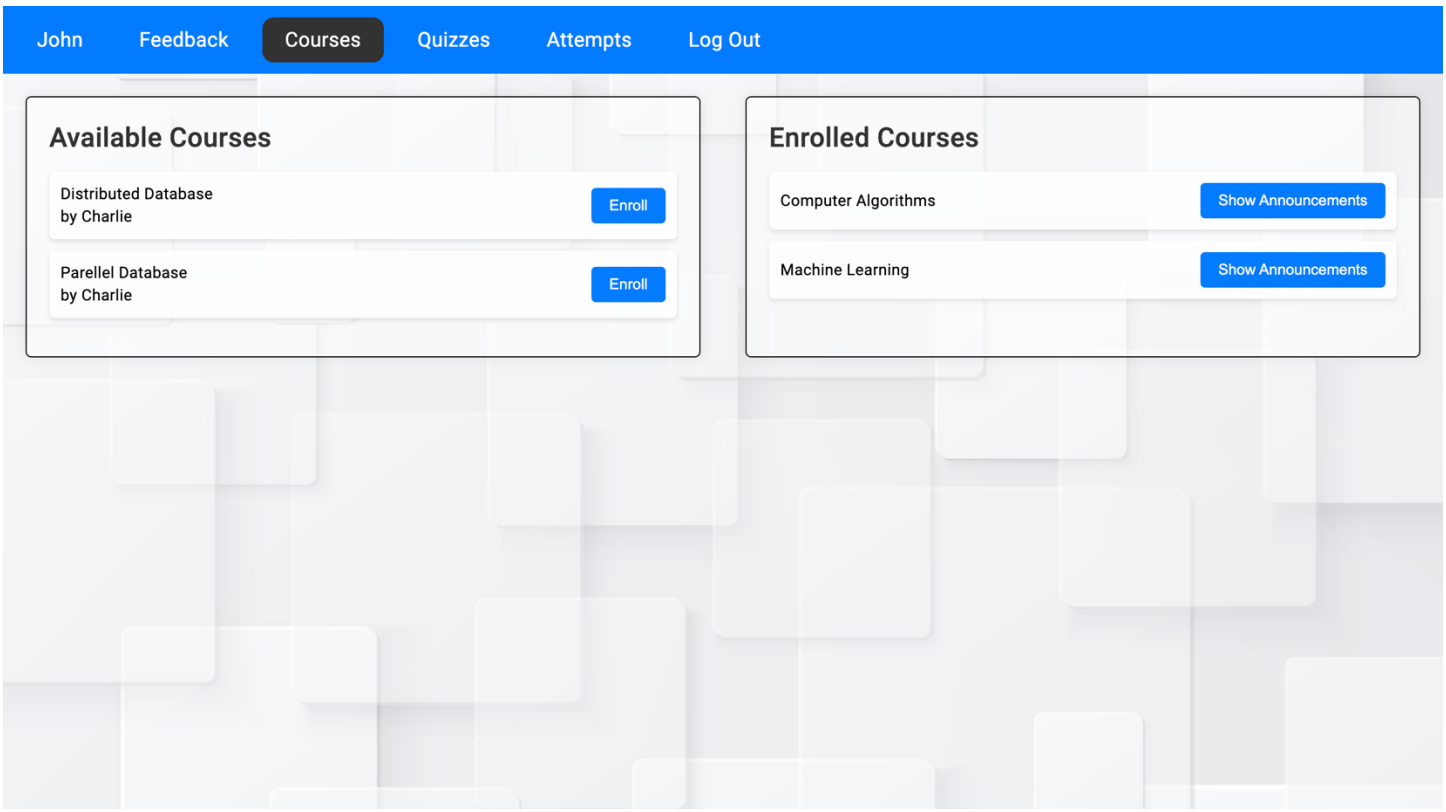


- Each course has its announcements.

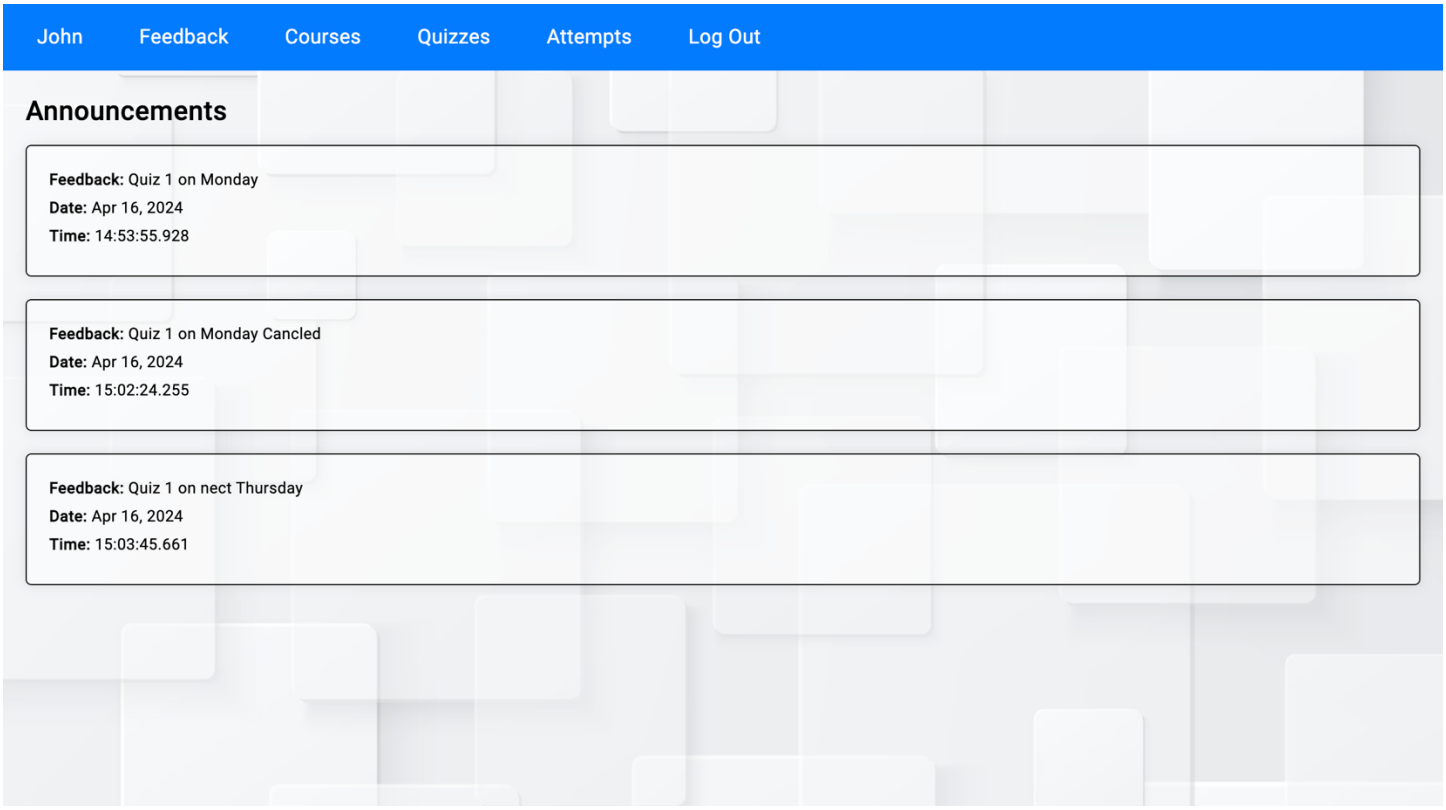


- The professor can add announcements for that particular course.

View Announcement



- Each course has a separate announcement.



- Students can see announcements for courses including data and time.

6. Conclusion:

This project has achieved its objectives by providing a solution that improves the assessment process in educational institutes. This application is developed with interconnected database schema structures and the implementation of user-friendly interface interaction between professors and students.

This application allows instructors to create courses, quizzes, and announcements and students can enroll in courses, take quizzes, and view their results, and attempt history.

To maintain integrity and efficiency this system meets 1NF, 2NF, and 3NF.

One potential limitation could be scalability with a large number of users or concurrent quiz sessions.

There are many other potential applications of this database system like online testing platforms, employee training modules, or certification program tests. The complex and expandable database design makes the system indispensable in today's digital age which offers solutions to a wide range of educational and organizational challenges.