

# **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:

S.No.	Title	Page No.
1	Abstract	3
2	Introduction	3
3	System Analysis	3
3.1	Database Design	3
3.2	Relational Scheme	4
3.3	Normalization	9
4	Implementation	9
4.1	Technologies Used	9
4.2	How to run?	10
4.3	Code Snippet	12
5	Results and Discussions	12
5.1	Functionalities	12
5.2	Assumptions	13
5.3	Challenges	13
5.4	Application Walk-through	13
6	Conclusion	30

## **1. Abstract:**

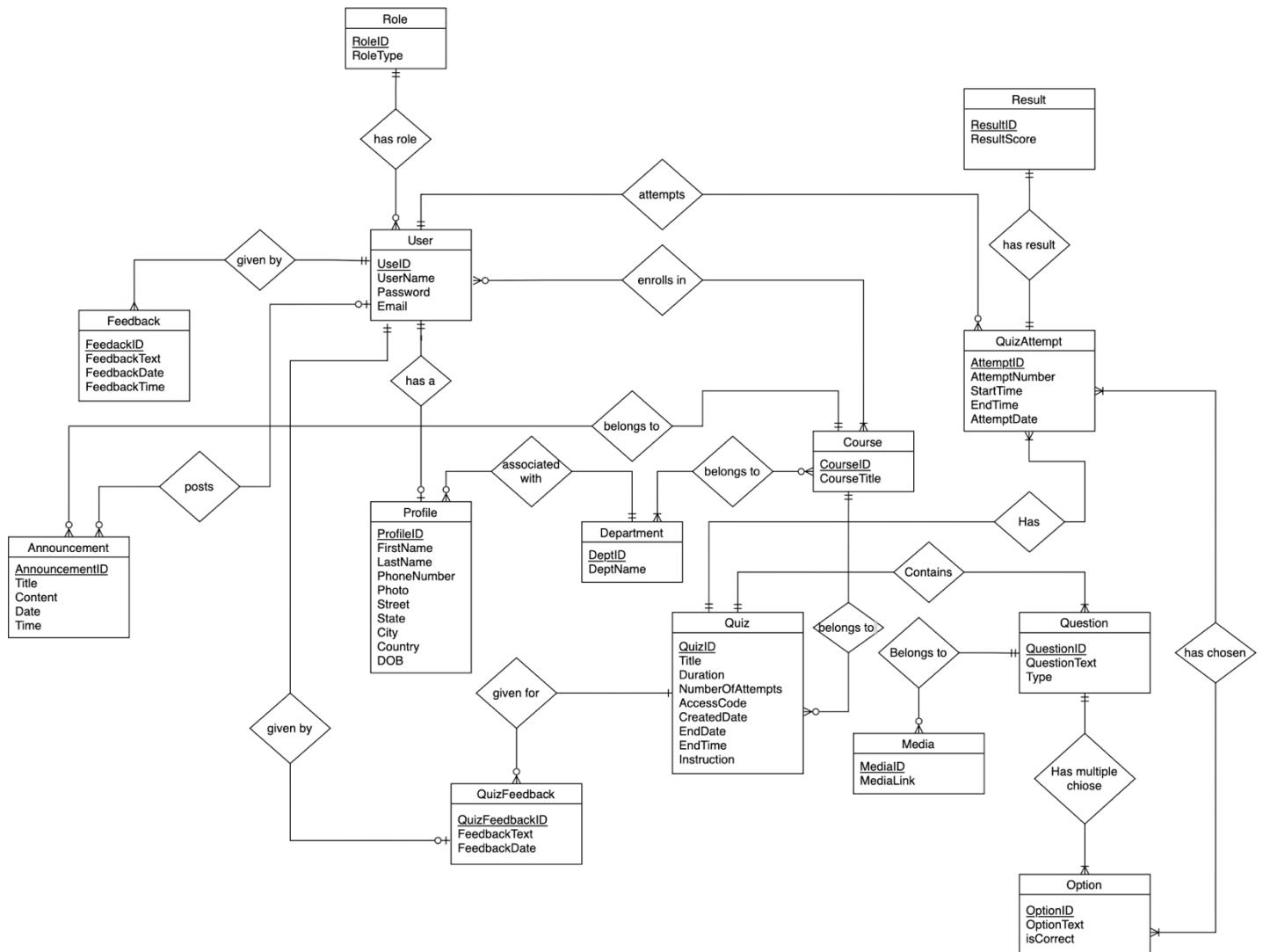
Our main objective in developing this project is to gain a deep understanding of database schema structures and apply this knowledge to create a real-time quiz management system application. By designing and implementing an interconnected **Entity-Relationship (ER)** diagram within the system's database, 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

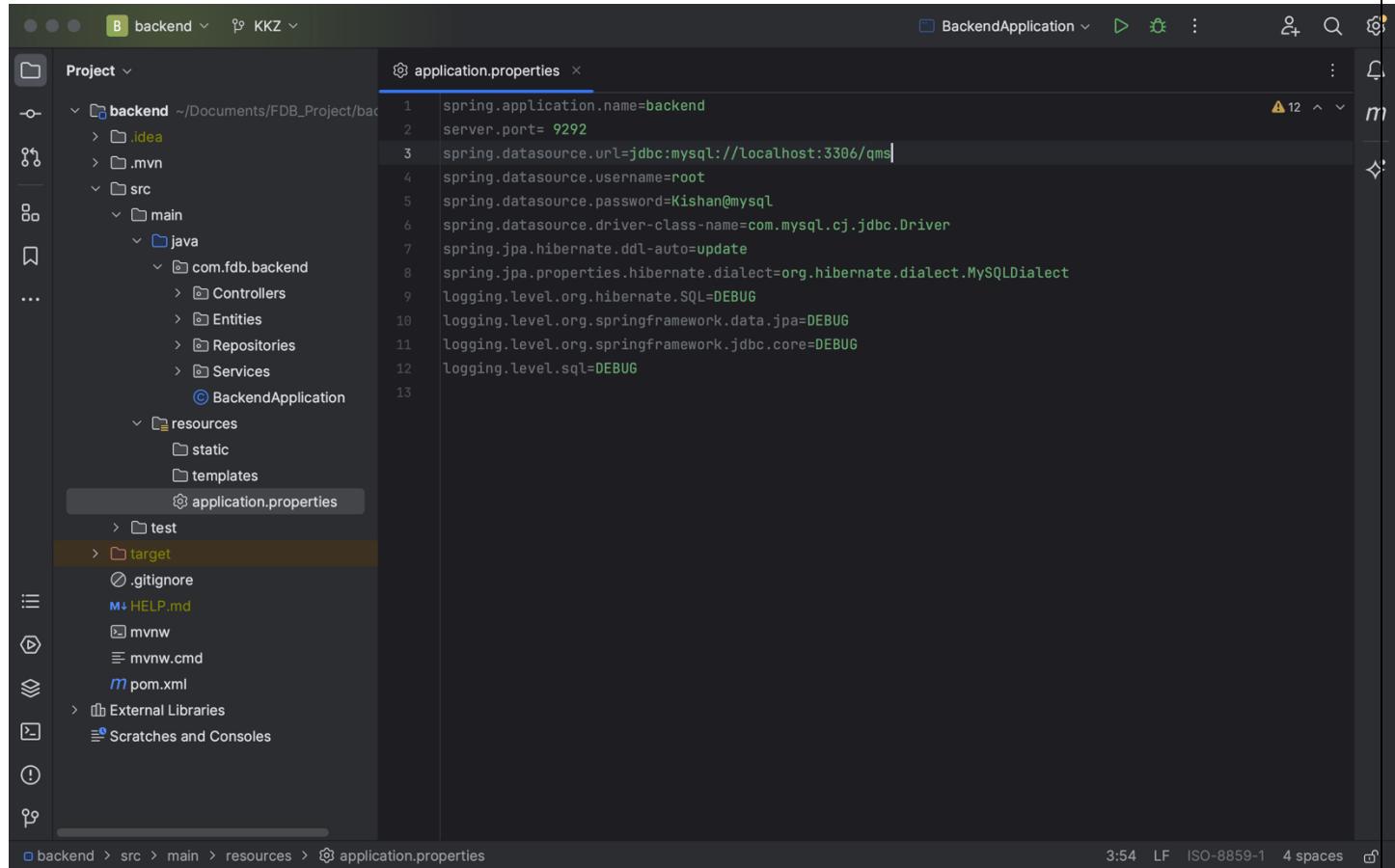
- Canvas Submission
- GitHub: [https://github.com/Kishan-Kumar-Zalavadia/FDB\\_Quiz\\_Management\\_System](https://github.com/Kishan-Kumar-Zalavadia/FDB_Quiz_Management_System)

### 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)



The screenshot shows the IntelliJ IDEA interface with the 'backend' project selected. The left sidebar displays the project structure, including the 'backend' module which contains '.idea', '.mvn', 'src' (with 'main' and 'java' subfolders), 'resources' (with 'static' and 'templates' subfolders), and 'target'. The 'application.properties' file is open in the main editor window. The code in the file is as follows:

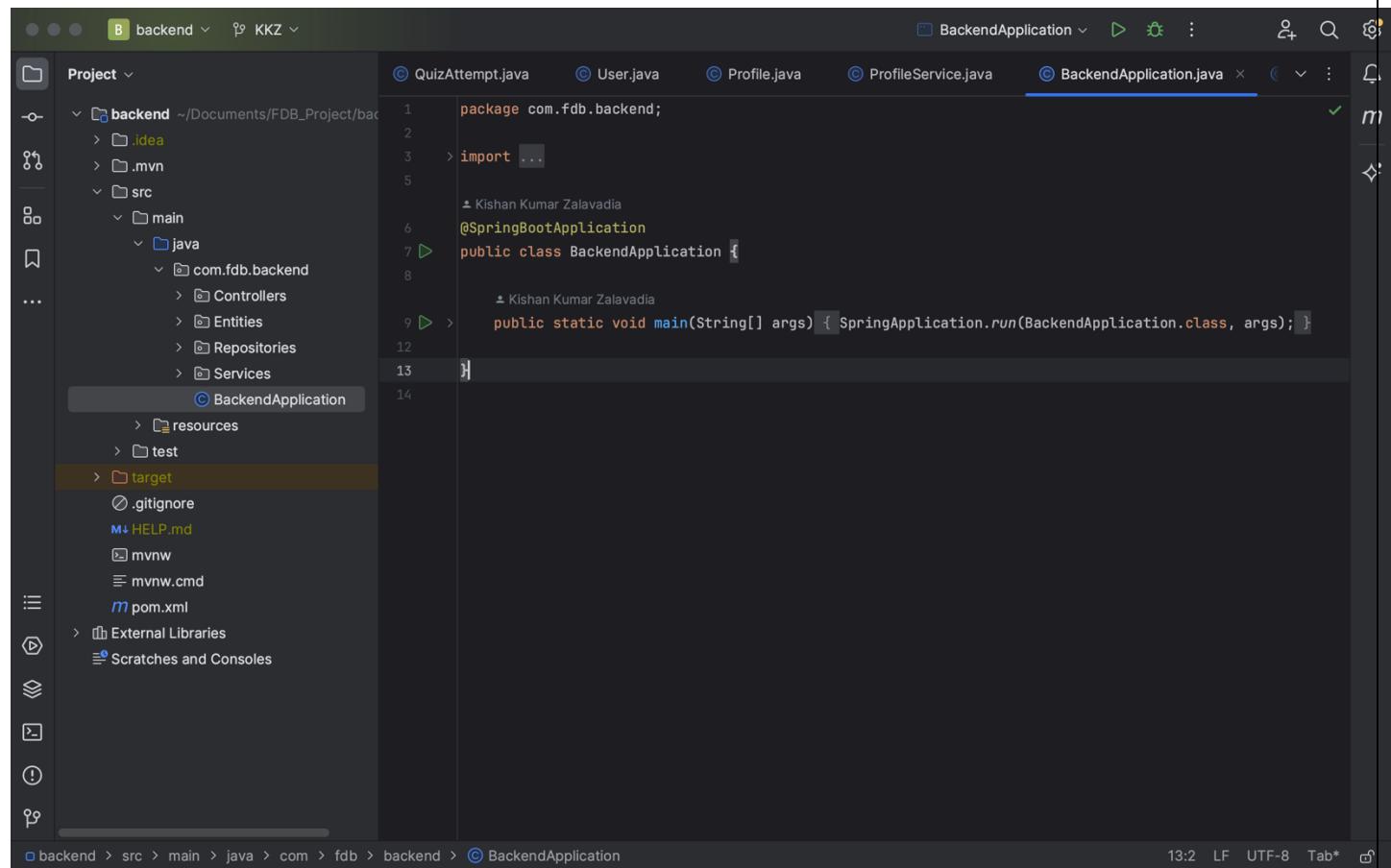
```
1 spring.application.name=backend
2 server.port= 9292
3 spring.datasource.url=jdbc:mysql://localhost:3306/qms
4 spring.datasource.username=root
5 spring.datasource.password=Kishan@mysql
6 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
7 spring.jpa.hibernate.ddl-auto=update
8 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
9 logging.level.org.hibernate.SQL=DEBUG
10 logging.level.org.springframework.data.jpa=DEBUG
11 logging.level.org.springframework.jdbc.core=DEBUG
12 logging.level.sql=DEBUG
13
```

## 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)



```
1 package com.fdb.backend;
2
3 > import ...
4
5
6 < Kishan Kumar Zalavadia
7 < @SpringBootApplication
8 < public class BackendApplication {
9 > < Kishan Kumar Zalavadia
10 > public static void main(String[] args) { SpringApplication.run(BackendApplication.class, args); }
11
12
13 }
14
```

## 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 ‘SampleData.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:  
  
Email Id is required field

User Name:  
  
Username is required field

Password:  
  
Password is required field

Role:  
✓ Select role  
Professor  
Student

**Register**

Existing User? [Login here](#)

### Registration

Email:  
  
Email Id is required field

User Name:  
  
Username is required field

Password:  
  
Password is required field

Role:

Department:  
✓ Select Department  
Computer Science  
Data Science  
Artificial Intelligence

Existing User? [Login here](#)

**Registration**

Registration Successful

Email:

User Name:

Password:

Role:

Department:

**Register**

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:**

**Login**

Email:

Email Id is required field

Password:

Password is required field

Remember me

**Login**

New user? Register here!!

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

### **Home Page:**

# Welcome to Quiz Management System

## Developer Information:

**Group: 13**

**Name:** Kishan Kumar Zalavadia

**ID:** 11685261

**Email:** KishanKumarZalavadia@my.unt.edu

**Name:** Valapadasu Uday Bhaskar

**ID:** 11696364

**Email:** ValapadasuUdayBhaskar@my.unt.edu

**Name:** Sapthagiri Naik Bhukya/p>

**ID:**

**Email:** SapthagiriNaikBhukya@my.unt.edu

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

## Profile Page:

The screenshot shows a user profile editing interface. At the top, there's a navigation bar with links for John, Feedback, Courses, Quizzes, Attempts, and Log Out. Below the navigation bar, the user's current profile information is displayed on the left, including fields for Name, Phone Number, Street, City, State, Zip, Country, DOB, and Age. An 'Edit' button is located below these fields. In the center, a modal window titled 'Edit User Profile' contains input fields for First Name (John), Last Name (Walker), Phone Number (9402637263), Street Number (12), Street Name (Elm), Apt Number (1213), City (Denton), State (Texas), Zip (56293), Country (USA), and Date of Birth (11/21/1980). There are 'Update Profile' and 'Close' buttons at the bottom of the modal. On the right side of the main content area, the updated profile information is shown: Name (John Walker), Phone Number (9402637263), Street (12 Elm, Apt 1213), City (Denton), State (Texas), Zip (56293), Country (USA), DOB (1980-11-21), and Age (43). An 'Edit' button is also present here.

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

## Feedback Page:

John Feedback Courses Quizzes Attempts Log Out

**Submit Feedback**

It is my first time using this, and I am loving it.

No feedbacks found.

John Feedback Courses Quizzes Attempts Log Out

**Submit Feedback**

Enter your feedback

**Feedbacks**

Feedback: It is my first time using this, and I am loving it.  
Date: Apr 13, 2024  
Time: 15:44:59

Feedback: It is amazing.  
Date: Apr 13, 2024  
Time: 15:45:09

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

## Add Courses (Professor)

## Add Course

Course Name:

Departments:

- Computer Science
- Data Science
- Artificial Intelligence

[Add Course](#)

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

## Add Course

Course Name:

Departments:

- Computer Science
- Data Science
- Artificial Intelligence

[Add Course](#)

## List of Courses

Course Name: Computer Algorithms

Departments:

Computer Science

[Show Quizzes](#) [Show Announcements](#)

Course Name: Machine Learning

Departments:

Computer Science  
Data Science  
Artificial Intelligence

[Show Quizzes](#) [Show Announcements](#)

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

## Add Quizzes

**Add Course**

Quiz Title:

Quiz 1: Ch - 1, 2, &amp; 3

Duration (in minutes):

20

Number of Attempts:

3

Access Code:

CAQ1

Instructions:

Open book and close internet quiz.

End Date:

04/29/2024



End Time:

11:59 PM

**Save Quiz**

- The professor can create several quizzes for each course.

Access Code:

Instructions:

End Date:

mm/dd/yyyy



End Time:

-- : -- --

**Save Quiz****Quizzes for Course Computer Algorithms**

Quiz Title: Quiz 2: Ch - 4, &amp; 5

**Add Questions****Show Quiz Feedback****Quiz Results**

Quiz Title: Quiz 1: Ch - 1, 2, &amp; 3

**Add Questions****Show Quiz Feedback****Quiz Results**

- The professor can see all the quizzes that he created

**Add Questions Page**

The screenshot shows a user interface for creating quizzes. At the top, there is a blue header bar with links for 'Macula', 'Feedback', 'My Courses', and 'Log Out'. Below the header, a button labeled 'Add Question' is visible. The main content area displays the message 'Questions for Quiz Quiz 1: Ch - 1, 2, & 3' and 'No questions found for Quiz Quiz 1: Ch - 1, 2, & 3'. A central modal window titled 'Add New Question' is open. It contains fields for 'Question' (containing 'What is an algorithm?') and 'Marks' (set to 5). There are 'Save Question' and 'Close' buttons at the bottom of the modal.

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

## Add Options

The screenshot shows a user interface for creating quizzes. At the top, there is a blue header bar with links for 'Macula', 'Feedback', 'My Courses', and 'Log Out'. Below the header, a button labeled 'Add Question' is visible. The main content area displays the message 'Questions for Quiz Quiz 1: Ch - 1, 2, & 3'. On the left, a question card for 'What is an algorithm?' is shown with a mark of 5, and buttons for 'Add Option' and 'Add Media'. A central modal window titled 'Add New Option' is open. It contains fields for 'Option Name' (containing 'A Flowchart') and 'Is Correct' (set to 'False'). There are 'Save Option' and 'Close' buttons at the bottom of the modal.

- 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.

The screenshot shows a user interface for creating quizzes. At the top, there is a blue header bar with navigation links: 'Macula', 'Feedback', 'My Courses', and 'Log Out'. Below the header, a button labeled 'Add Question' is visible. The main content area displays two questions:

**Questions for Quiz Quiz 1: Ch - 1, 2, & 3**

**What is an algorithm?**

Marks: 5

**Add Option** **Add Media**

Options:

- 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

**Add Option** **Add Media**

Options:

- Processor and Memory
- Time and Space

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

## Add Media

localhost:4200 says

Enter the media link:

/www.programiz.com/sites/tutorial2program/files/Bubble-sort-0.png

Cancel OK

**What is the sorting algorithm used in the image?**

Marks: 5

Add Option Add Media

Options:

- Processor and Memory
- Time and Space

Options:

- Bubble Sort
- Selection Sort
- Insertion sort

Media:

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

## Course Page

John Feedback Courses Quizzes Attempts Log Out

**Available Courses**

Distributed Database by Charlie [Enroll](#)

Parallel Database by Charlie [Enroll](#)

**Enrolled Courses**

Computer Algorithms [Show Announcements](#)

Machine Learning [Show Announcements](#)

- 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

The screenshot shows a user interface for a quiz system. At the top, there is a navigation bar with links: John, Feedback, Courses, Quizzes (which is highlighted in black), Attempts, and Log Out. Below the navigation bar, the title "All Quizzes" is displayed. There are two quiz entries, each in its own box:

**Computer Algorithms - Quiz 2: Ch - 4, & 5**

- Duration: 30
- Attempts: 0 / 2
- Access Code: CAQ2
- Instruction: Open book and close internet quiz.
- Created Date: Apr 13, 2024
- End Date: May 11, 1

[Take Quiz](#)

**Computer Algorithms - Quiz 1: Ch - 1, 2, & 3**

- Duration: 30
- Attempts: 0 / 3
- Access Code: CAQ1
- Instruction: Open Book, Close internet
- Created Date: Apr 13, 2024
- End Date: Apr 29, 2024

[Take Quiz](#)

- 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.

localhost:4200 says  
Your quiz score is: 10

i = 2    -2    0    45    11    -9

i = 3    -2    0    11    45    -9

-2    0    11    -9    45

Bubble Sort  
 Selection Sort  
 Insertion sort

**Submit Quiz**

**Quiz Feedback:**  
The quiz was easy.

**Submit Feedback**

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

John    Feedback    Courses    **Quizzes**    Attempts    Log Out

### All Quizzes

**Computer Algorithms - Quiz 2: Ch - 4, & 5**

Duration: 30  
Attempts: 0 / 2  
Access Code: CAQ2  
Instruction: Open book and close internet quiz.  
Created Date: Apr 13, 2024  
End Date: May 11, 1

**Take Quiz**

**Computer Algorithms - Quiz 1: Ch - 1, 2, & 3**

Duration: 30  
Attempts: 3 / 3  
Access Code: CAQ1  
Instruction: Open Book, Close internet  
Created Date: Apr 13, 2024  
End Date: Apr 29, 2024

- 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

John    Feedback    Courses    Quizzes    Attempts    Log Out

### Quiz Attempts

Quiz Name: Quiz 1: Ch - 1, 2, & 3

Attempt Number: 1

End Time: 16:29:21.899

Attempt Date: 2024-04-13

Score: 10

Quiz Name: Quiz 1: Ch - 1, 2, & 3

Attempt Number: 2

End Time: 16:30:21.047

Attempt Date: 2024-04-13

Score: 20

Quiz Name: Quiz 1: Ch - 1, 2, & 3

Attempt Number: 3

End Time: 16:30:47.476

Attempt Date: 2024-04-13

Score: 20

- 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

Access Code:

Instructions:

End Date:

mm/dd/yyyy

End Time:

-- : -- : --

Save Quiz

### Quizzes for Course Computer Algorithms

Quiz Title: Quiz 2: Ch - 4, & 5

Add Questions

Show Quiz Feedback

Quiz Results

Quiz Title: Quiz 1: Ch - 1, 2, & 3

Add Questions

Show Quiz Feedback

Quiz Results

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

The screenshot shows a user interface for viewing quiz feedback. At the top, there is a blue header bar with four items: "Macula", "Feedback", "My Courses", and "Log Out". Below the header, the title "Quiz Feedbacks" is displayed. There are two separate feedback entries, each enclosed in a light gray box:

**User: John**  
Feedback: The quiz was easy.  
Date: Apr 13, 2024  
Time: 16:29:17.087

**User: Emily**  
Feedback: The quiz had all direct questions.  
Date: Apr 13, 2024  
Time: 16:36:58.002

- 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

## Quiz Results

User: John

- Attempt Number: 1 Score: 10
- Attempt Number: 2 Score: 20
- Attempt Number: 3 Score: 20

User: Emily

- Attempt Number: 1 Score: 15

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

## Add announcement

### Add Course

Course Name:

Departments:

- Computer Science
- Data Science
- Artificial Intelligence

[Add Course](#)

### List of Courses

Course Name: Computer Algorithms

Departments:  
Computer Science

[Show Quizzes](#) [Show Announcements](#)

Course Name: Machine Learning

Departments:  
Computer Science  
Data Science  
Artificial Intelligence

[Show Quizzes](#) [Show Announcements](#)

- Each course has its announcements.

## Add Announcement

Enter your announcement

[Submit](#)

## Announcements

**Feedback:** Quiz 1 on Monday

**Date:** Apr 16, 2024

**Time:** 14:53:55.928

**Feedback:** Quiz 1 on Monday Canceled

**Date:** Apr 16, 2024

**Time:** 15:02:24.255

**Feedback:** Quiz 1 on next Thursday

**Date:** Apr 16, 2024

**Time:** 15:03:45.661

- The professor can add announcements for that particular course.

## View Announcement

### Available Courses

Distributed Database  
by Charlie

[Enroll](#)

Parallel Database  
by Charlie

[Enroll](#)

### Enrolled Courses

Computer Algorithms

[Show Announcements](#)

Machine Learning

[Show Announcements](#)

- Each course has a separate announcement.

## Announcements

Feedback: Quiz 1 on Monday

Date: Apr 16, 2024

Time: 14:53:55.928

Feedback: Quiz 1 on Monday Canceled

Date: Apr 16, 2024

Time: 15:02:24.255

Feedback: Quiz 1 on next Thursday

Date: Apr 16, 2024

Time: 15:03:45.661

- 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.