



## **UE22CS352B - Object Oriented Analysis & Design**

### **Mini Project Report**

**Title: QuizNest – Online Quiz Application**

*Submitted by:*

**Trishita Umapathi: PES2UG22CS631**  
**Deepak Velmurugan: PES2UG22CS639**  
**Vandana J: PES2UG22CS643**  
**Vignesh Palanirajan: PES2UG22CS661**

Semester: 6

Section: K

**Faculty Name: Dr.Nandhi Kesavan**

**January - May 2025**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**FACULTY OF ENGINEERING**  
**PES UNIVERSITY**

(Established under Karnataka Act No.  
16 of 2013) 100ft Ring Road, Bengaluru  
– 560 085, Karnataka, India

## **Problem Statement:**

The objective of this project is to develop a Java-based desktop application for a Quiz Management System that facilitates seamless quiz participation for users and efficient administration for system administrators. The application supports user authentication, multiple quiz types, and score tracking, while providing administrative capabilities such as user management and question creation. The system is developed using Java Swing and adheres to the Model-View-Controller (MVC) architecture. In addition, it incorporates key Object-Oriented Design Principles and Design Patterns to ensure maintainability, scalability, and modularity.

## **Key Features**

### **User Authentication and Role-Based Access**

- Secure login and signup functionality.
- Role-based redirection for "Admin" and "Participant" accounts.

### **Quiz Participation (Participant Module)**

- Users can participate in quizzes categorized into three scoring strategies:
  - *Simple Scoring*: 1 point for a correct answer, 0 for incorrect.
  - *Negative Scoring*: 1 point for correct, -1 for incorrect.
  - *Time-Based Scoring*: Score influenced by accuracy and response time.
- Each quiz session consists of 10 randomly selected questions.
- Display of total score and detailed analysis of correct/incorrect responses upon completion.

### **Score History and Scoreboard**

- Participants can view their quiz attempt history along with scoring type and timestamp.
- A global scoreboard sorted by score for each scoring type is available.
- Implementation of the *Observer Pattern* ensures real-time scoreboard updates.

### **Administrative Dashboard (Admin Module)**

- View the highest score achieved by each participant.
- Delete participant accounts from the system.
- Add new quiz questions by specifying the question, options (A-D), and the correct answer.
- All administrative actions update the database in real-time.

## Models

### 1. User

Represents a user of the system (admin or participant).

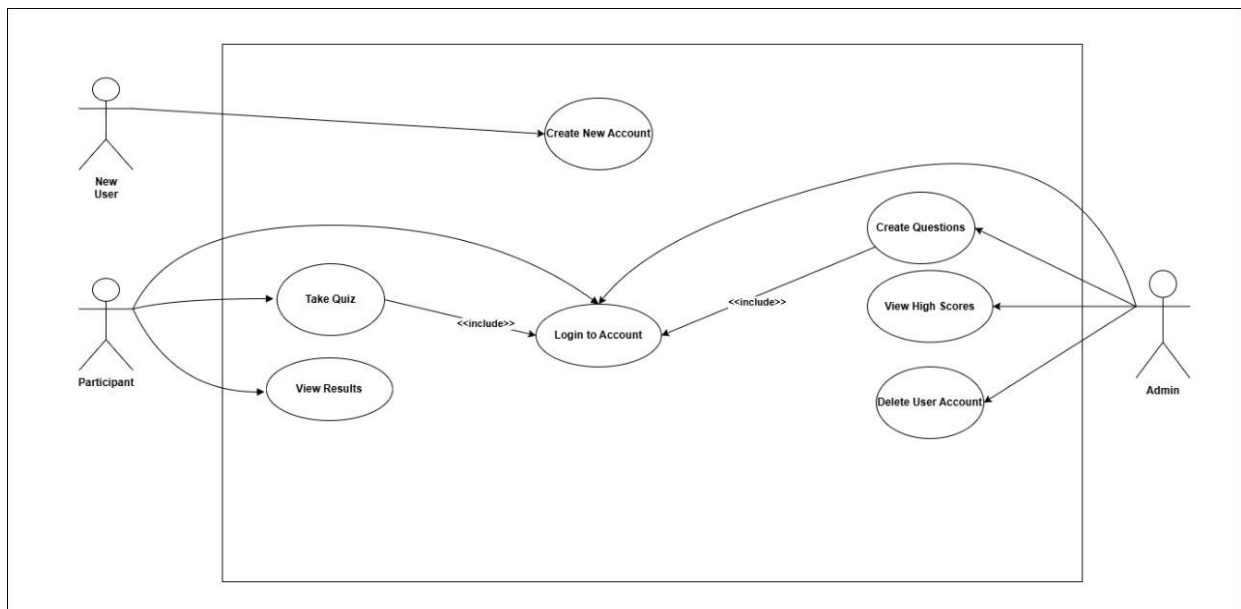
- **Fields:**
  - id: Unique user ID
  - username: Username of the user
  - password: Encrypted password
  - role: User role (Admin or Participant)

### 2. Question

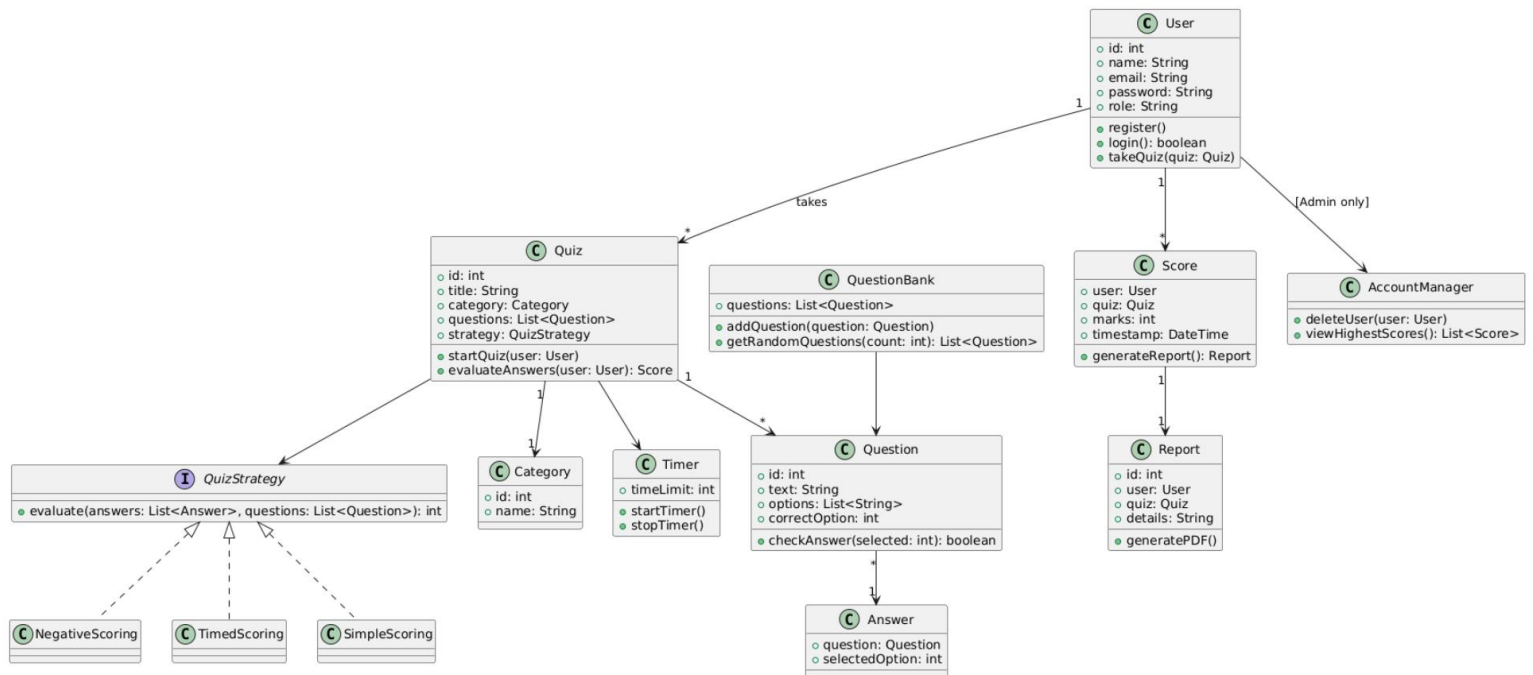
Represents a quiz question.

- **Fields:**
  - id: Unique question ID
  - question: Question text
  - optionA to optionD: Multiple-choice options
  - correctOption: Correct answer (A/B/C/D)

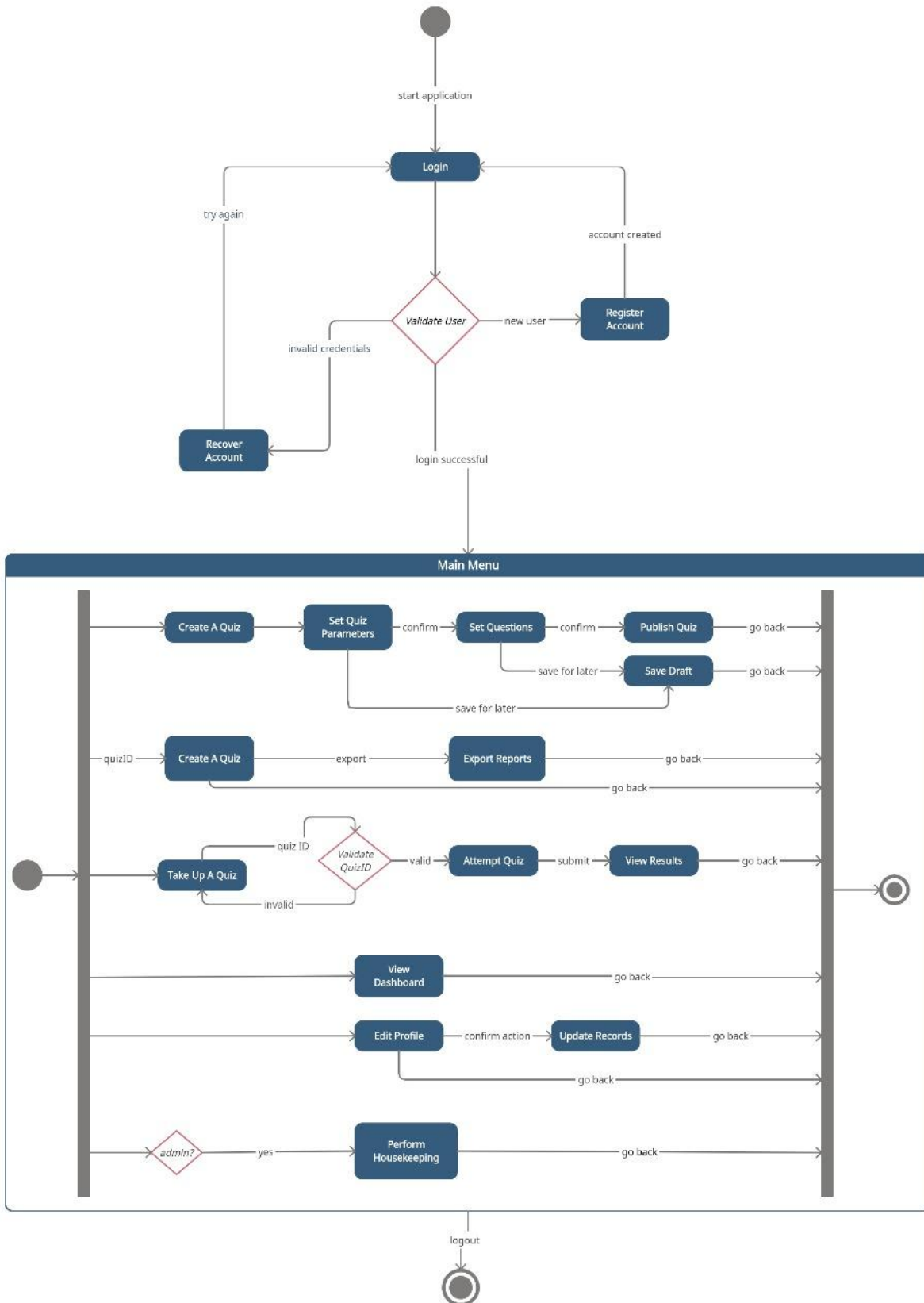
### Use Case Diagram:



## Class Diagram:



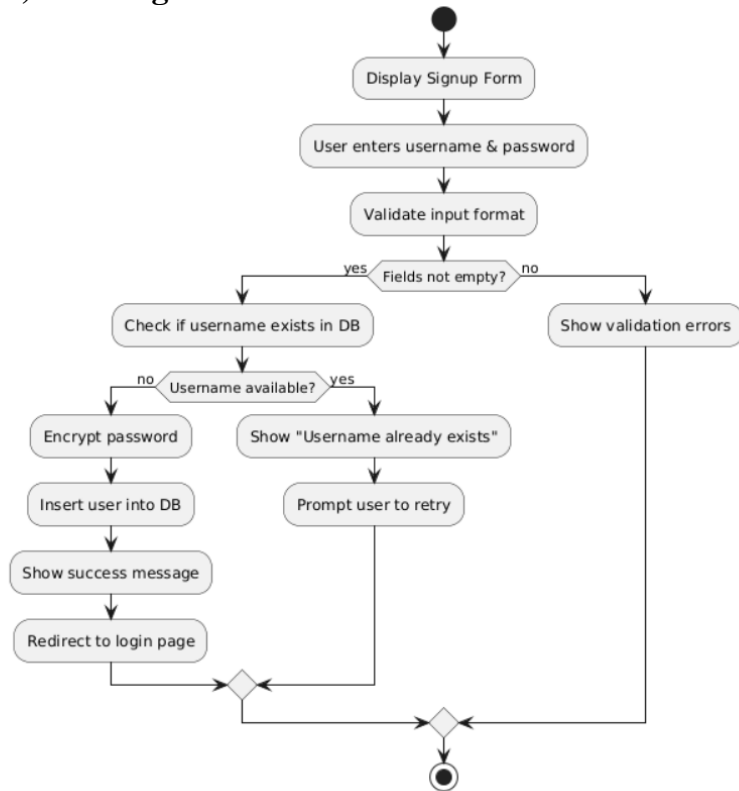
## State Diagram:



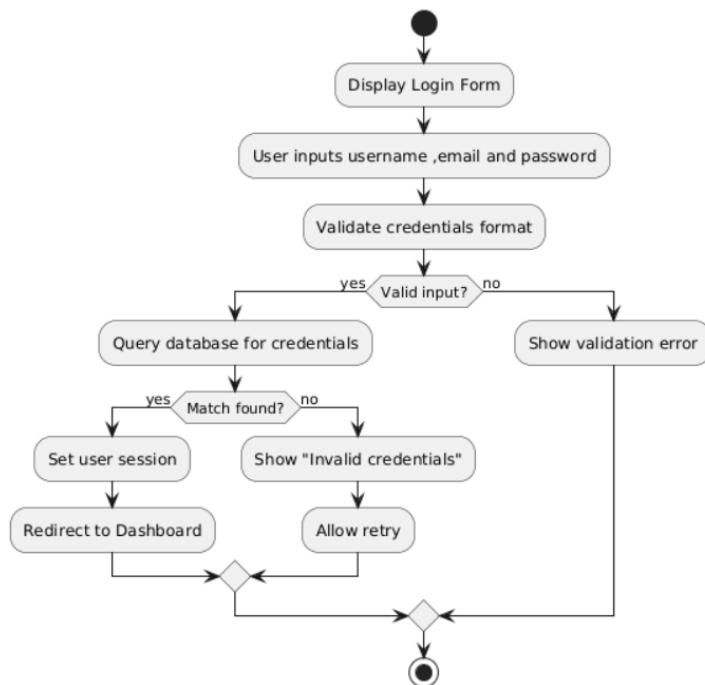
## Activity Diagrams:

### 1. Major Usecase

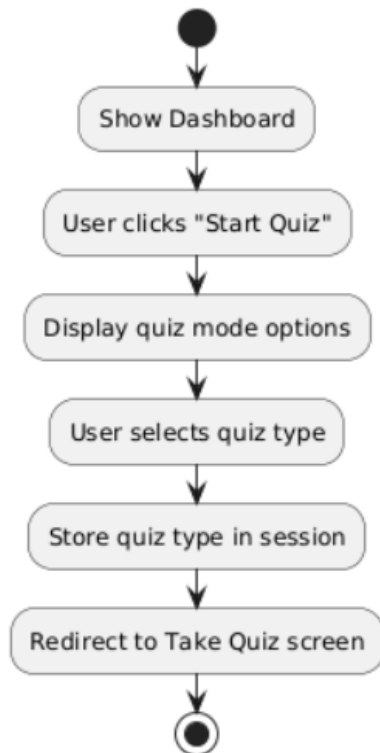
#### 1) User Registration



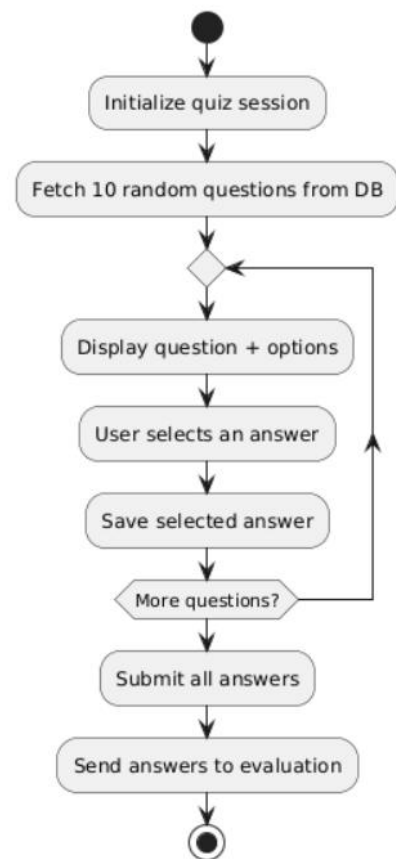
#### 2) User Login



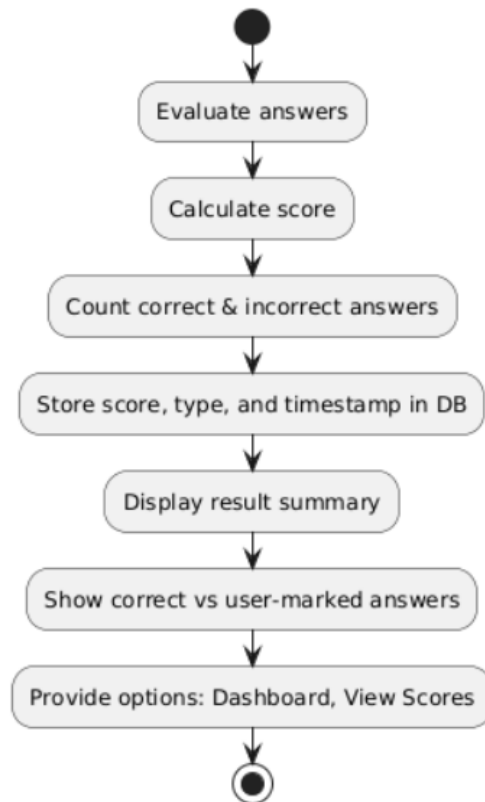
### 3)Start Quiz



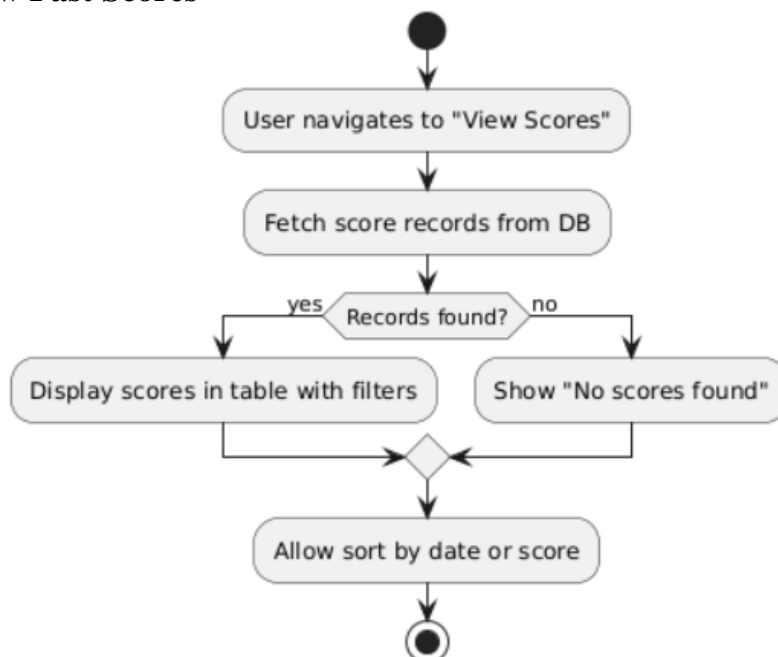
### 4)Take Quiz



### 5) View Score After Quiz



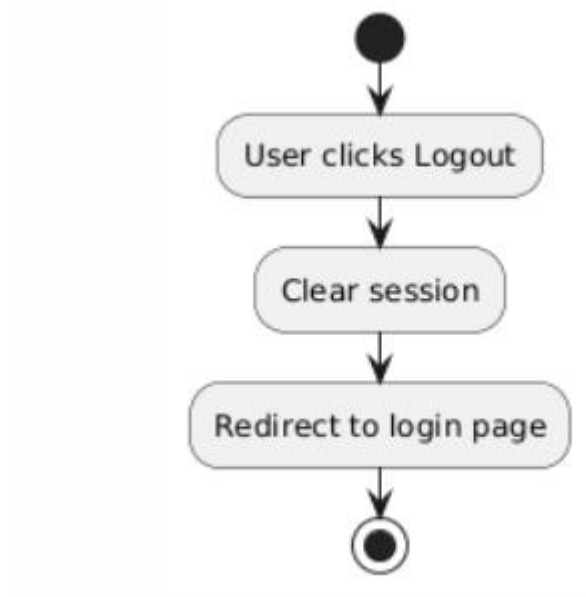
### 6) View Past Scores



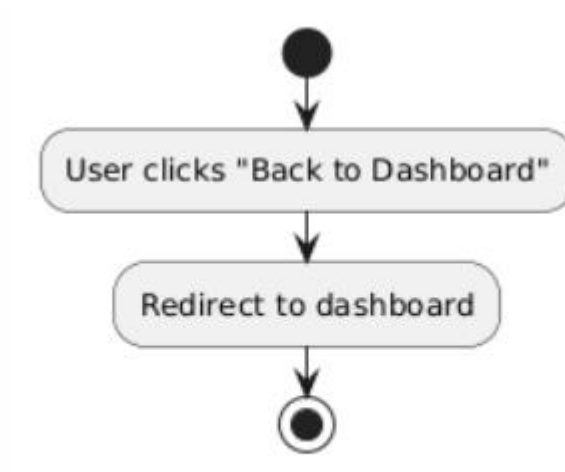


## 2. Minor Use case

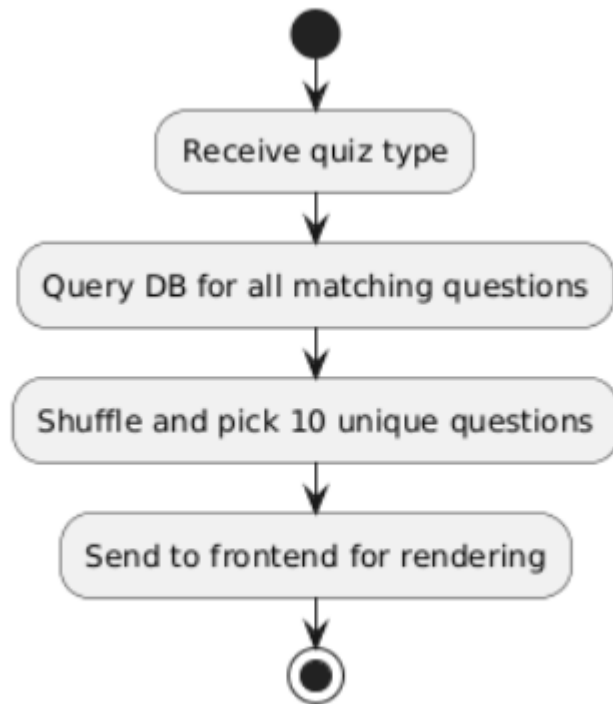
### 1)Logout



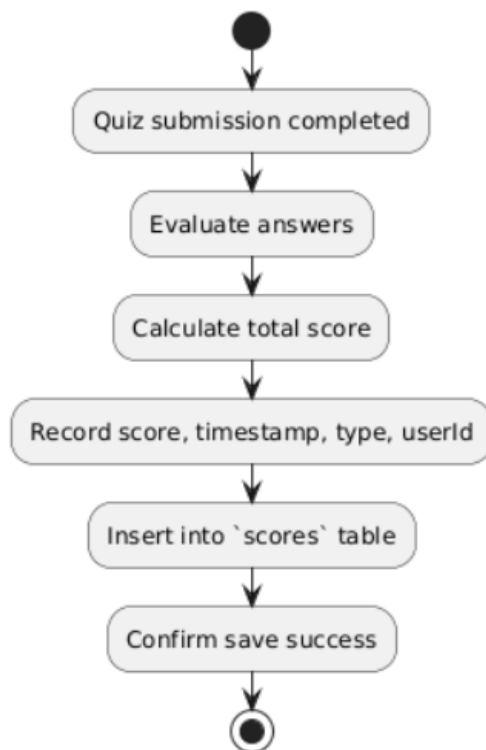
### 2)Return to Dashboard



### 3) Fetching Random Questions



### 4) Score Persistence



## **Architecture Patterns, Design Principles, and Design Patterns:**

### **Architecture Pattern**

#### **Model – View – Controller (MVC)**

- **Model:** Represents core data structures such as User and Question. Responsible for storing and managing quiz-related data.
- **View:** Java Swing-based GUI components like LoginForm, AdminDashboard, QuizScreen, and ScoreScreen for user interaction.
- **Controller:** Mediates input from views and updates the model or view accordingly (e.g., handling login, starting quizzes, loading questions).

#### **Benefits:**

- Clear separation of concerns
- Enhanced modularity and maintainability
- Easier testing and debugging

### **Design Principles**

#### **1. Single Responsibility Principle (SRP)**

- Each class serves a single purpose (e.g., UserDao handles DB operations related to users; QuizScreen manages quiz UI).

#### **2. Open/Closed Principle (OCP)**

- New quiz types (e.g., time-based, negative scoring) can be added using the Strategy pattern without modifying existing quiz logic.

#### **3. Liskov Substitution Principle (LSP)**

- Different scoring strategies (Simple, Negative, Time-based) implement a common ScoringStrategy interface and can be used interchangeably.

#### **4. Interface Segregation Principle (ISP)**

- Interfaces and classes are designed with focused responsibilities (e.g., scoring logic, question management).

#### **5. Dependency Inversion Principle (DIP)**

- High-level modules depend on abstractions like ScoringStrategy, not concrete classes. Promotes loose coupling and testability.

## **Design Patterns**

### **1. Singleton**

- Database class ensures a single, shared connection instance to the database. Prevents redundant connections and ensures thread safety.

### **2. Factory**

- ScoringStrategyFactory dynamically creates scoring strategies (Simple, Negative, Time-Based) based on quiz type, decoupling creation logic.

### **3. Strategy**

- Used to encapsulate different scoring logics in separate classes implementing the ScoringStrategy interface.

### **4. Observer**

- ScoreboardScreen observes quiz results and updates the scoreboard display accordingly, enabling dynamic tracking

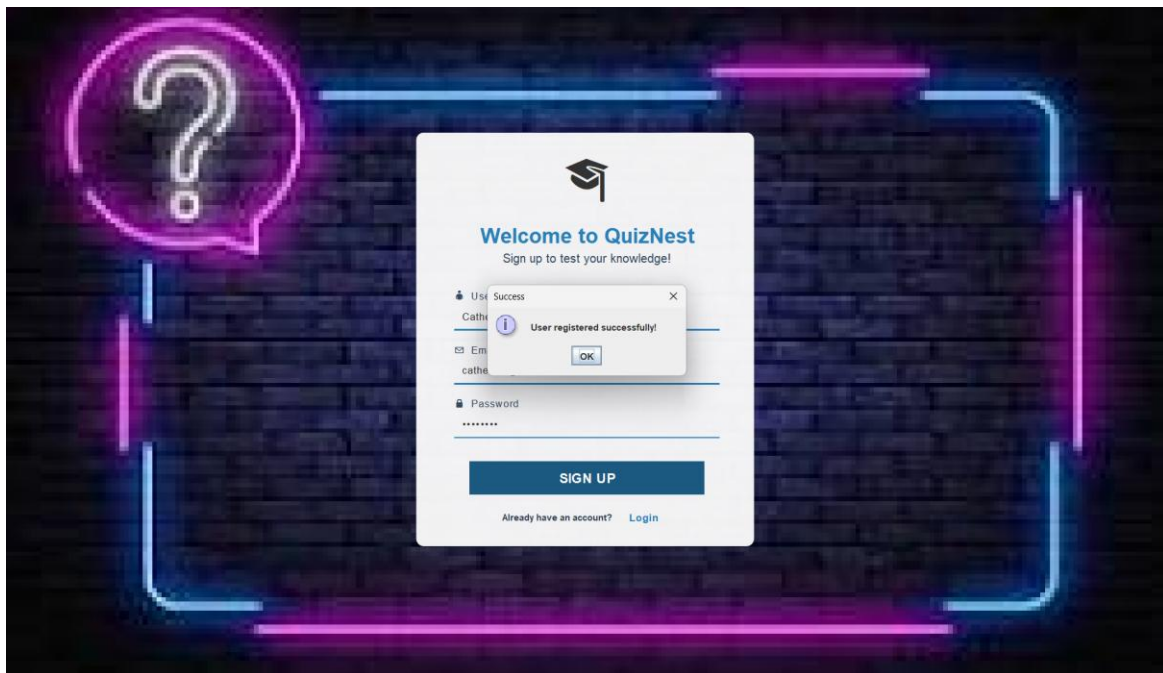
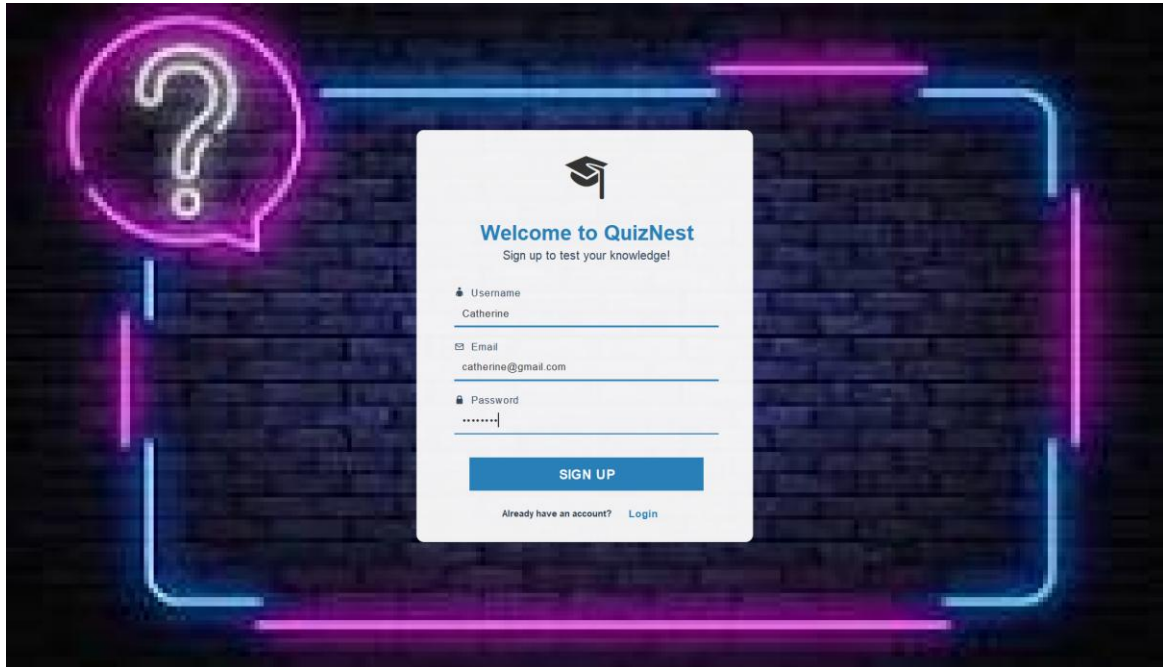
## **Github link to the Codebase:**

<https://github.com/vignesh-palanirajan/ooad-quizapp>

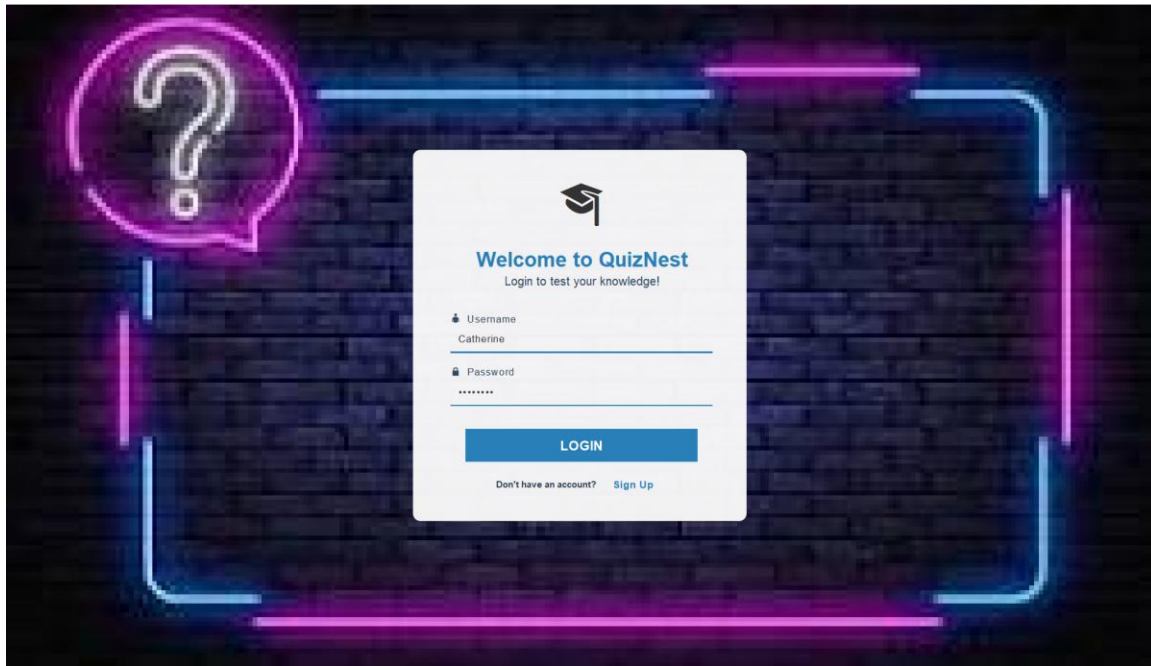
## Screenshots

UI:


SIGNUP



## PARTICIPANT LOGIN



The login form is centered on a dark brick wall background. To the left is a glowing pink speech bubble with a white question mark. The form itself is a white card with a blue border. At the top left of the card is a black graduation cap icon. Below it, the text 'Welcome to QuizNest' is in blue, followed by 'Login to test your knowledge!' in grey. There are two input fields: 'Username' with the text 'Catherine' and 'Password' with masked characters '\*\*\*\*\*'. A blue 'LOGIN' button is below the fields. At the bottom, there is a link 'Don't have an account? Sign Up'.



### Welcome to QuizNest

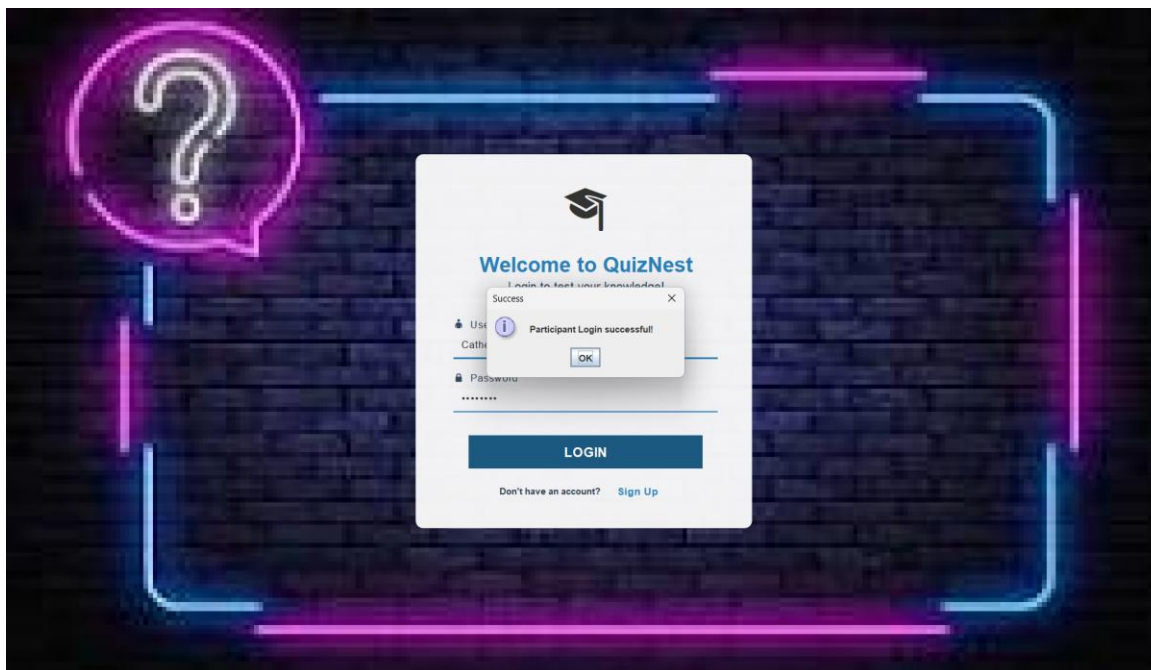
Login to test your knowledge!

Username  
Catherine


Password  
\*\*\*\*\*

**LOGIN**

[Don't have an account? Sign Up](#)



This image shows the same login form as above, but with a success message overlay. The overlay is a small white box with a blue border, containing a blue information icon, the text 'Participant Login successfull', and an 'OK' button. The background and form elements are identical to the previous image.



### Welcome to QuizNest

Login to test your knowledge!

Username  
Catherine

Password  
\*\*\*\*\*

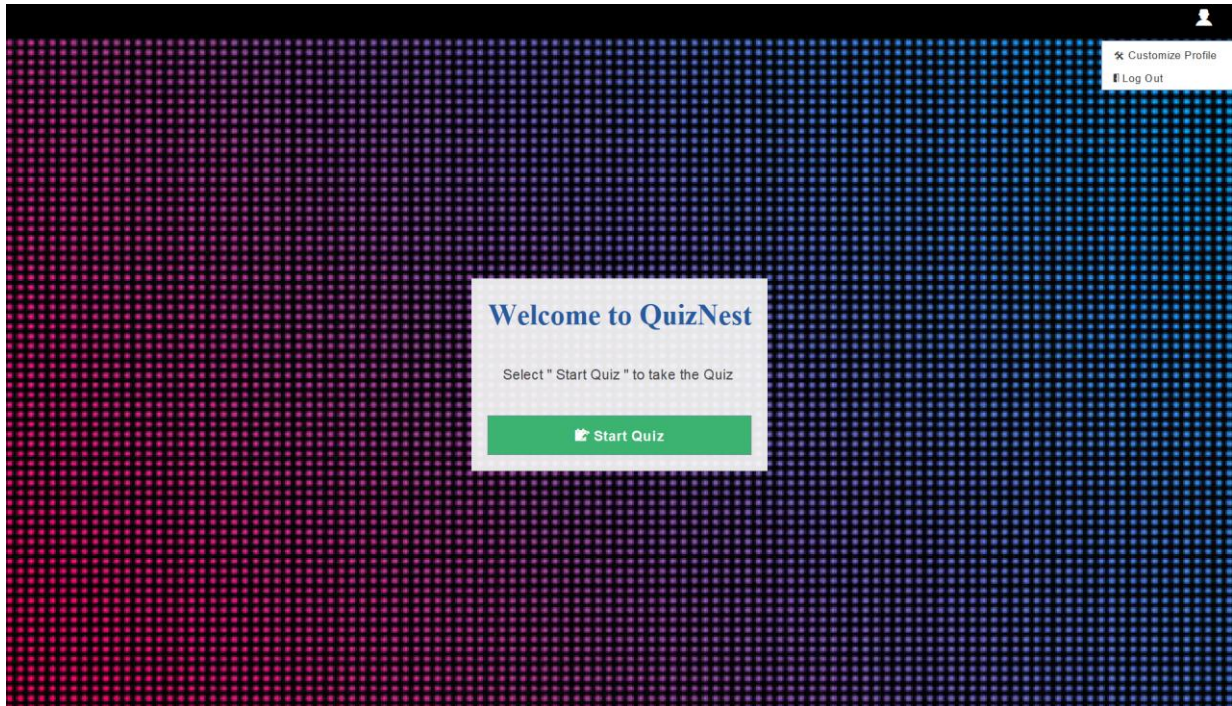
**LOGIN**

[Don't have an account? Sign Up](#)

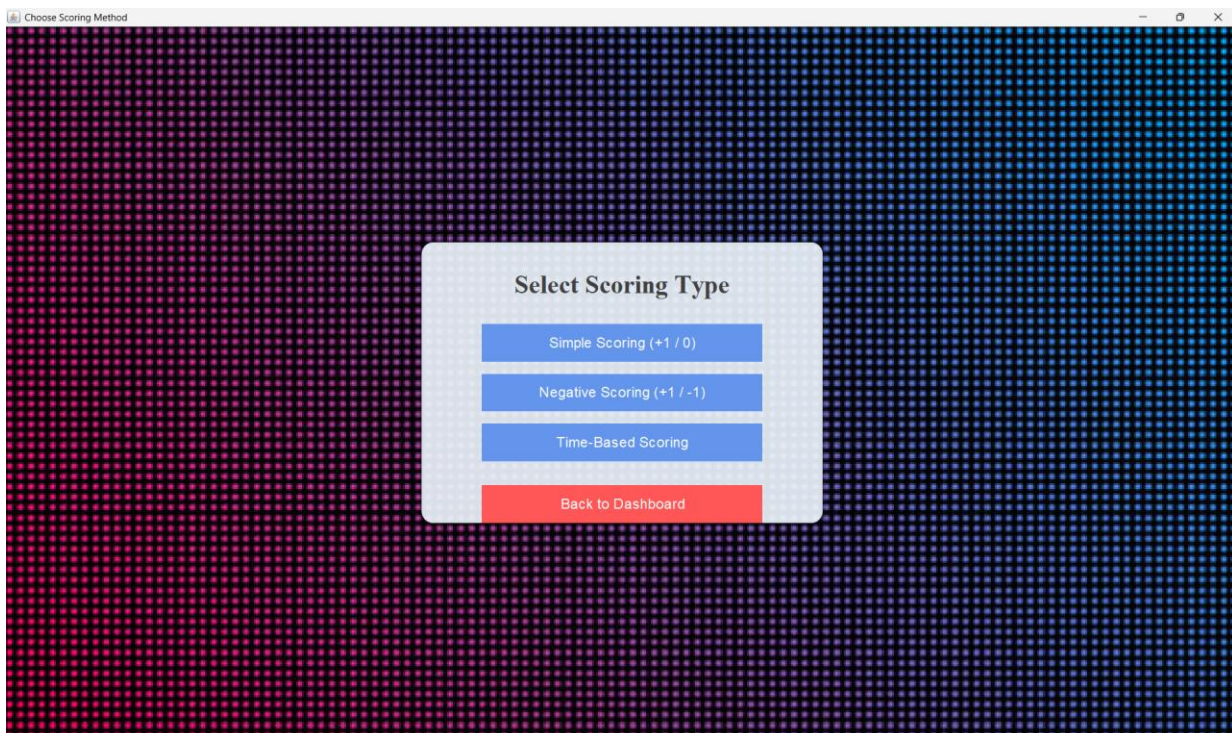
**Success**  
Participant Login successfull  
**OK**



## QUIZ DASHBOARD

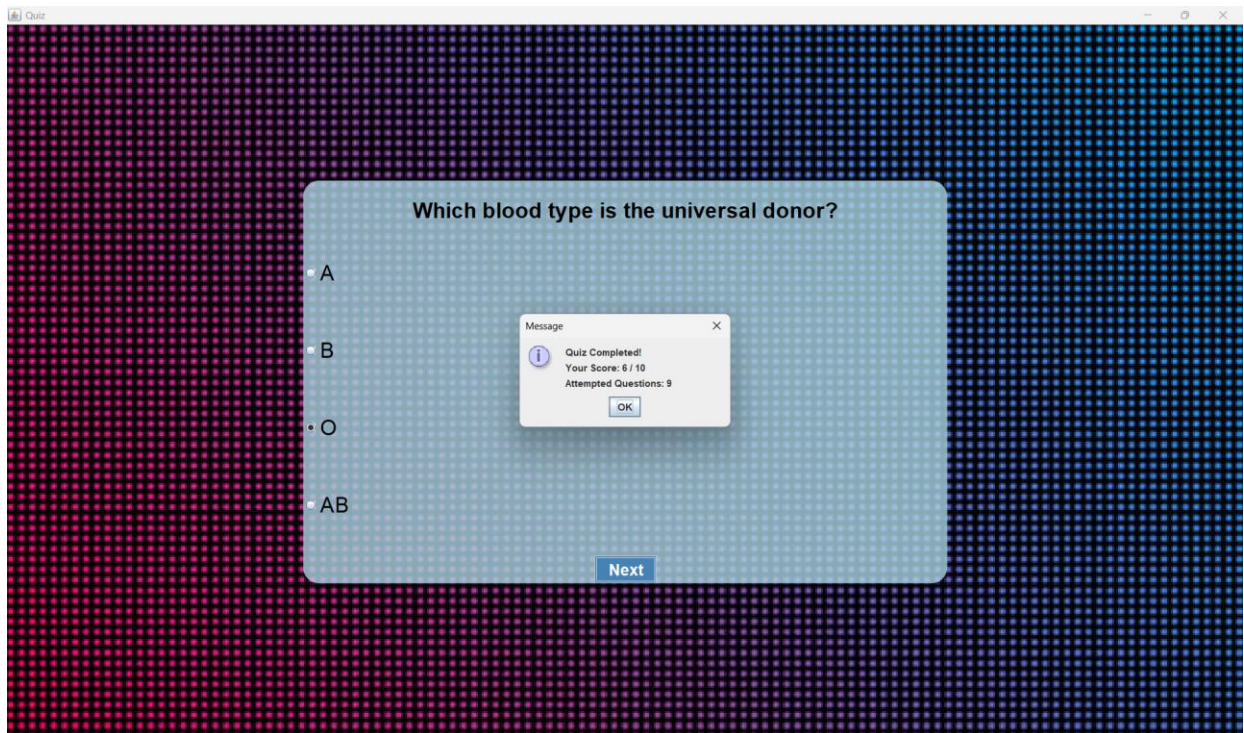
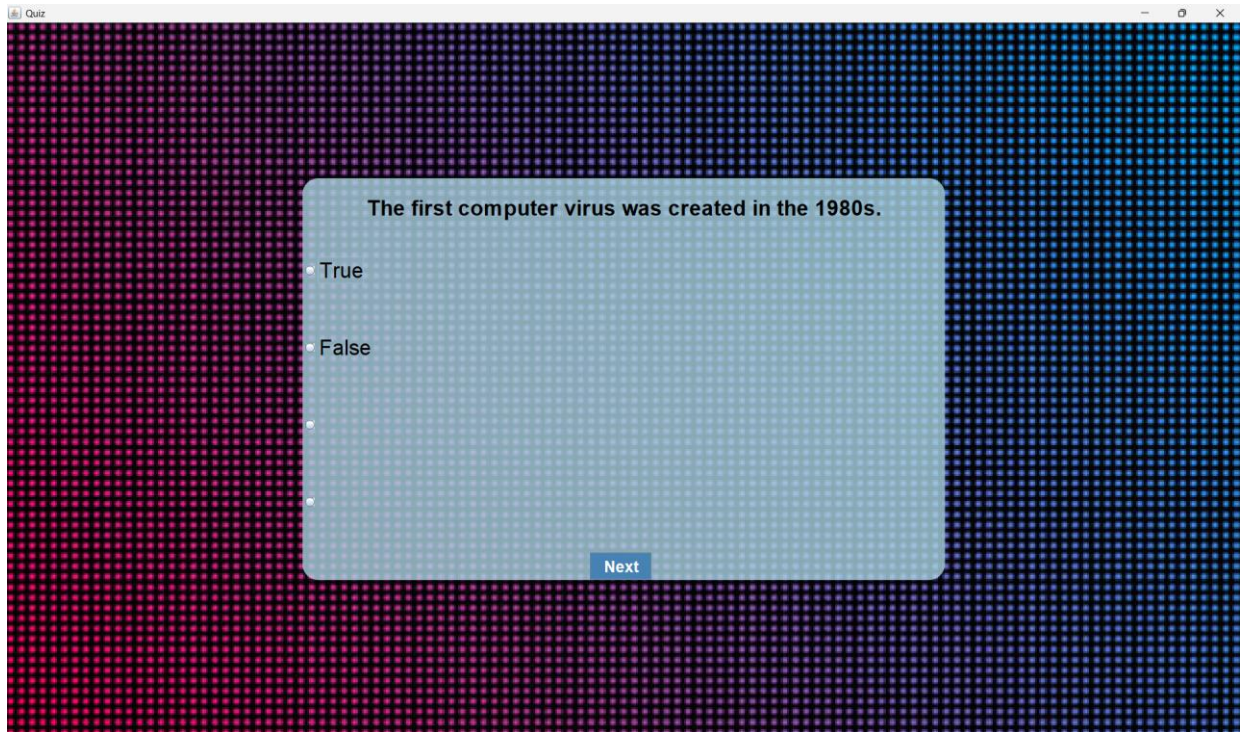


## SCORING TYPES



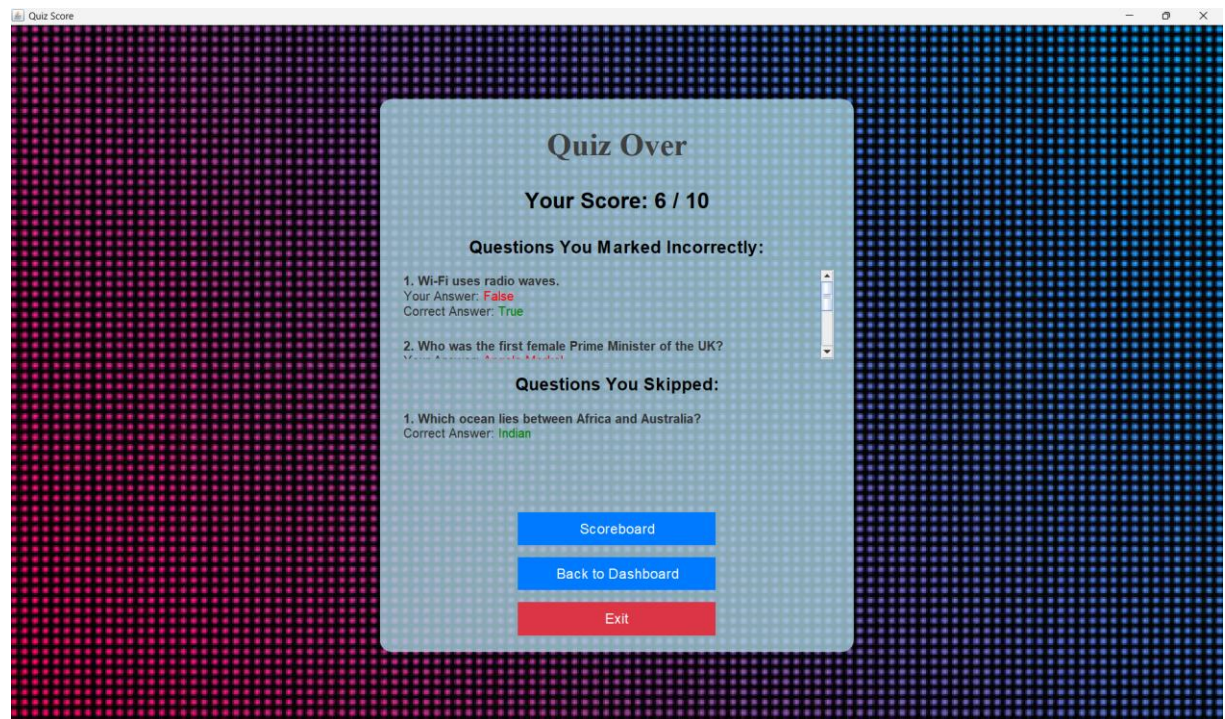


## QUIZ WINDOW

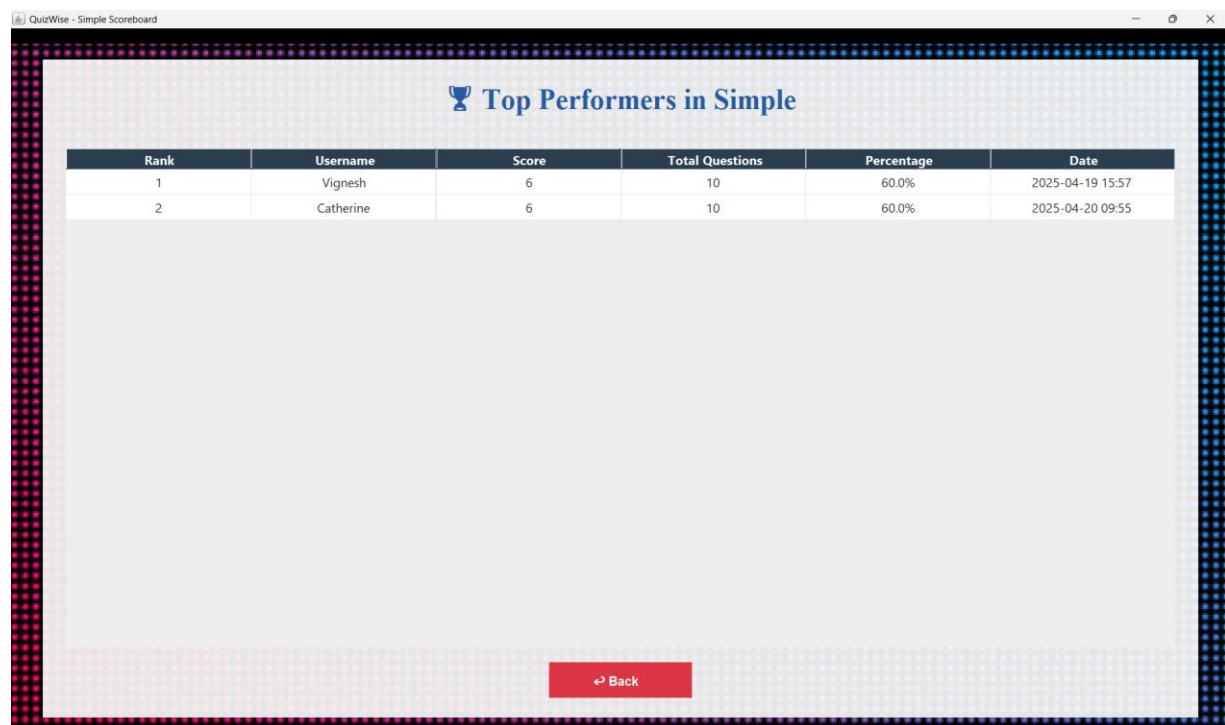




## QUIZ SCORE



## SCOREBOARD



## EDIT PROFILE

The screenshot shows a web browser window titled "QuizWise - Edit Profile". The background is a dark blue and purple grid pattern. In the center is a light blue rectangular form titled "Edit Your Profile". The form contains three input fields: "Username:" with the text "CatherineI", "Password:" with masked characters "\*\*\*\*\*", and "Email:" with the text "catherine@gmail.com". Below the input fields are three buttons: a blue "Save" button, a red "Back" button, and a green "View Scores" button.

## VIEW SCORES

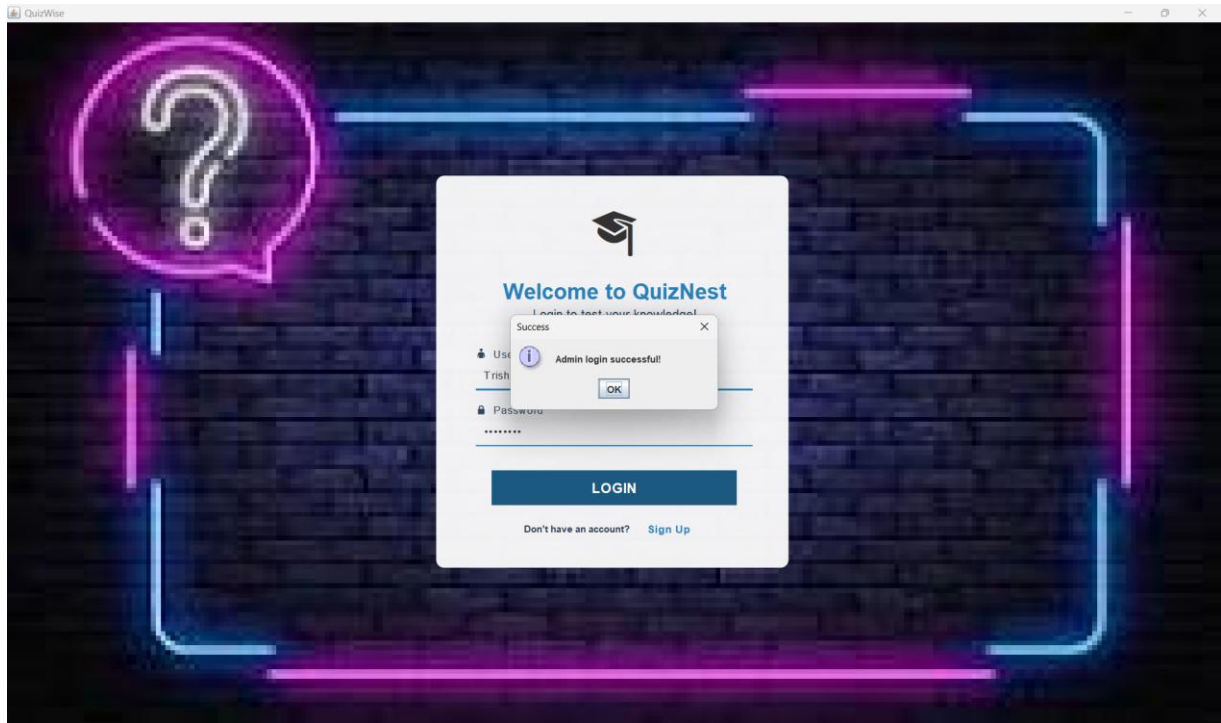
The screenshot shows a web browser window titled "Quiz Score History". The page has a blue header bar with the text "Your Quiz Performance History" and a link "View your past quiz results". Below the header is a table with the following data:

Type	Score	Total Questi...	Percentage	Date & Time
simple	6	10	60.0%	April 20, 2025 at 09:55 am

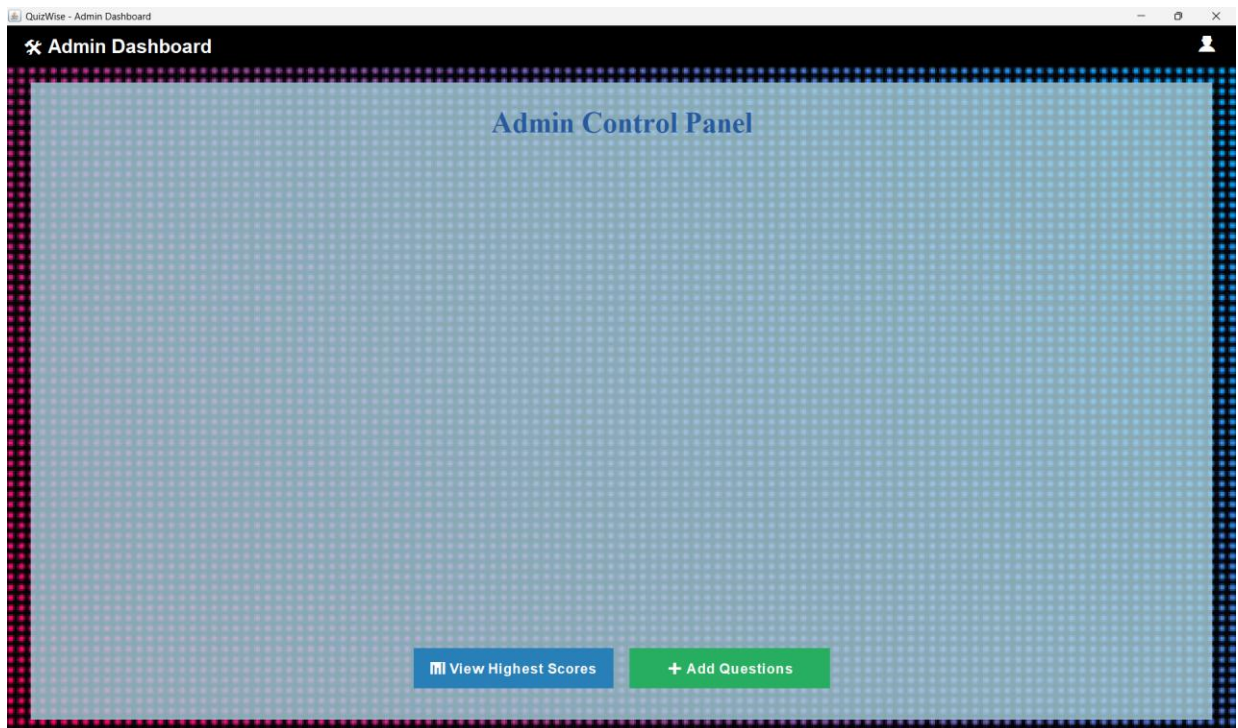
Below the table is a large empty rectangular box. At the bottom of the page is a blue button labeled "Back to Dashboard".



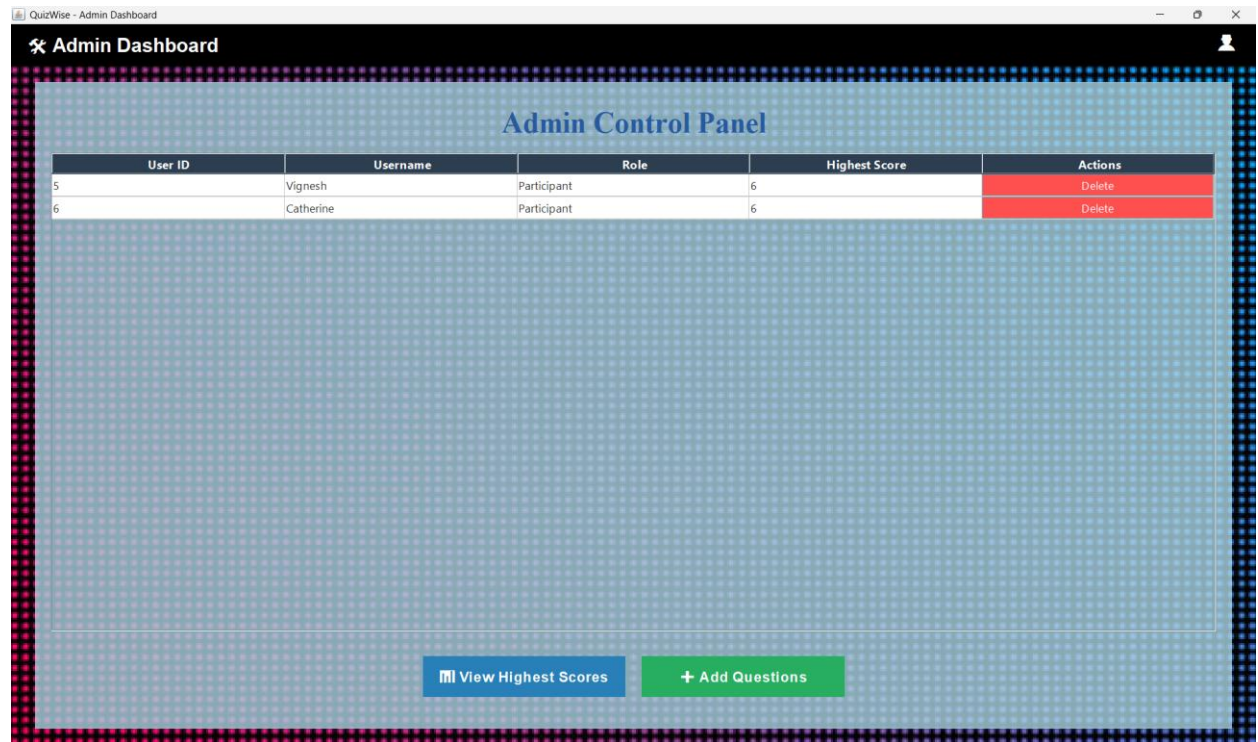
## ADMIN LOGIN



## ADMIN DASHBOARD



## VIEW HIGHEST SCORES

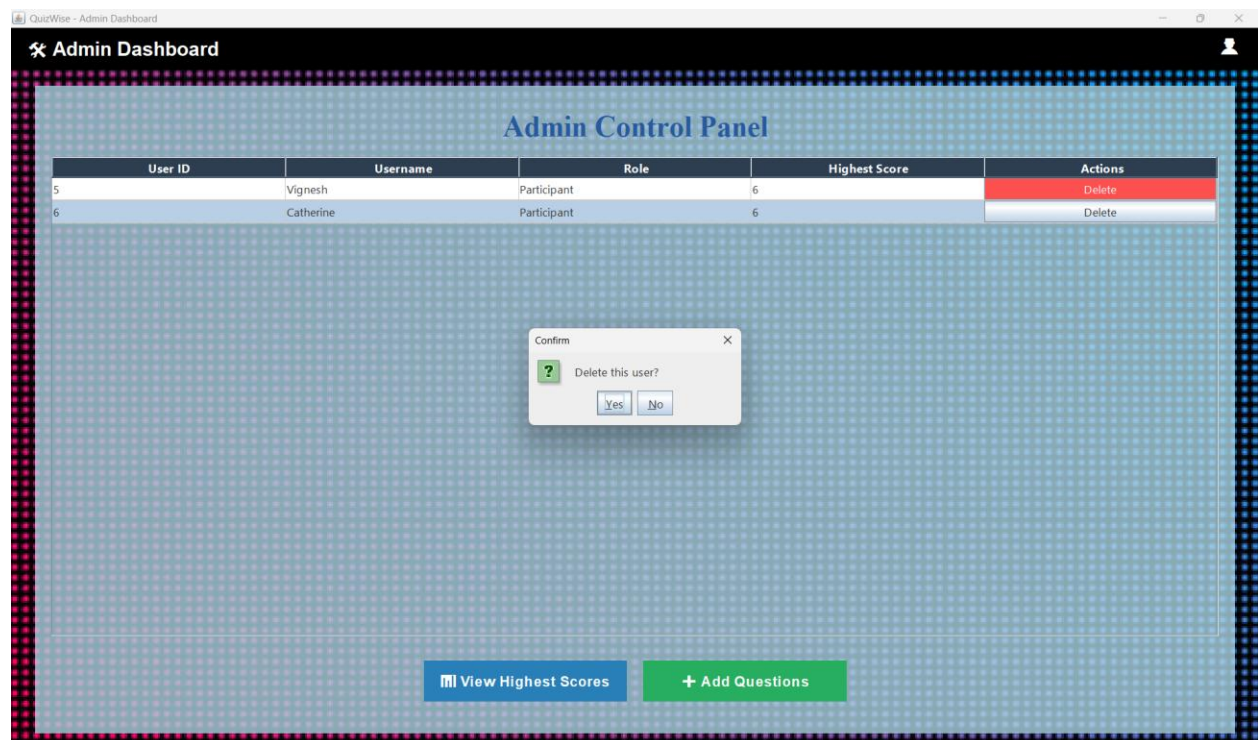


The screenshot shows the QuizWise Admin Dashboard. At the top, there's a header with the QuizWise logo and the text "Admin Dashboard". Below this is a large blue area titled "Admin Control Panel". Inside this panel is a table with the following data:

User ID	Username	Role	Highest Score	Actions
5	Vignesh	Participant	6	Delete
6	Catherine	Participant	6	Delete

Below the table, there are two buttons: "View Highest Scores" and "+ Add Questions".

## DELETE USER



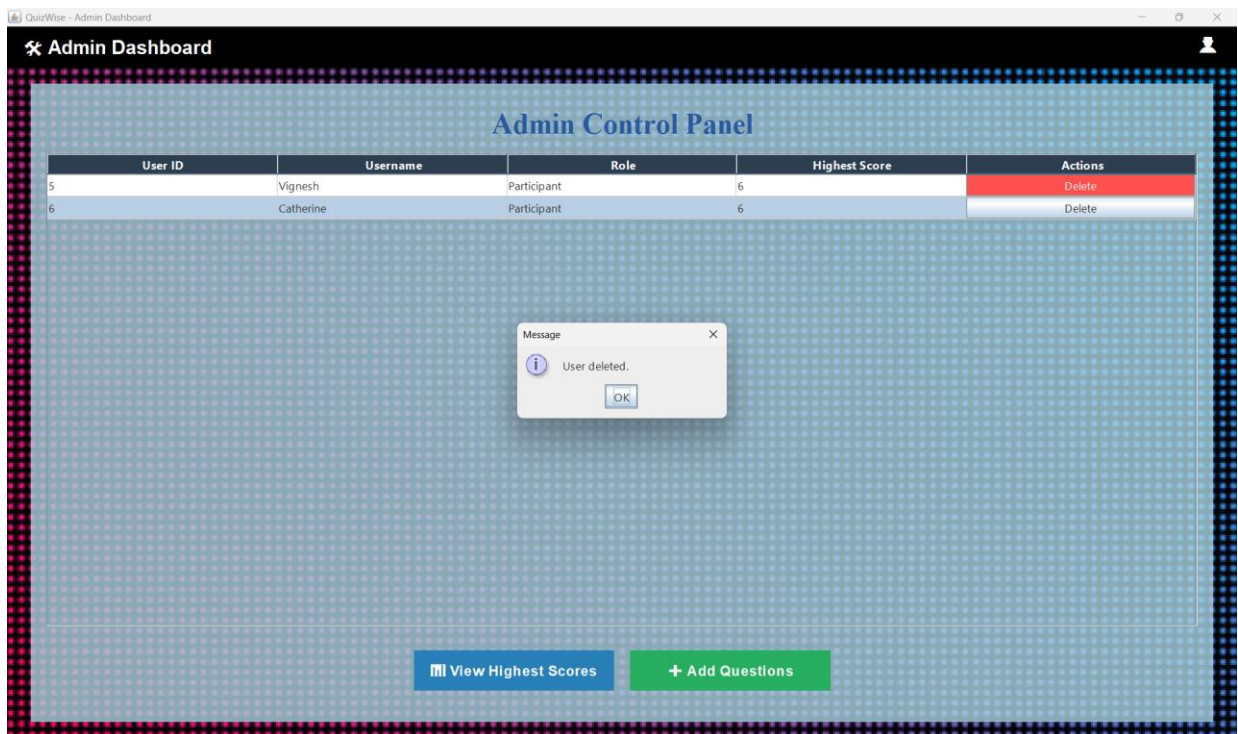
The screenshot shows the QuizWise Admin Dashboard. At the top, there's a header with the QuizWise logo and the text "Admin Dashboard". Below this is a large blue area titled "Admin Control Panel". Inside this panel is a table with the following data:

User ID	Username	Role	Highest Score	Actions
5	Vignesh	Participant	6	Delete
6	Catherine	Participant	6	Delete

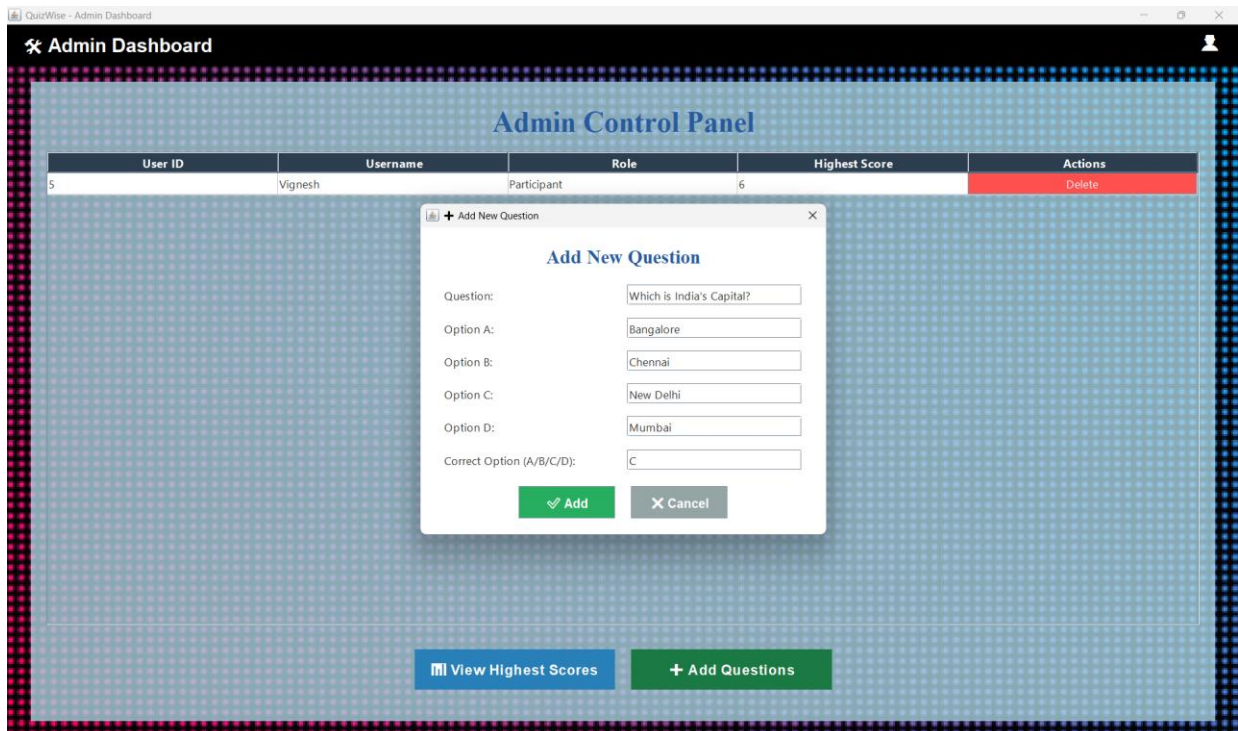
Below the table, there are two buttons: "View Highest Scores" and "+ Add Questions".

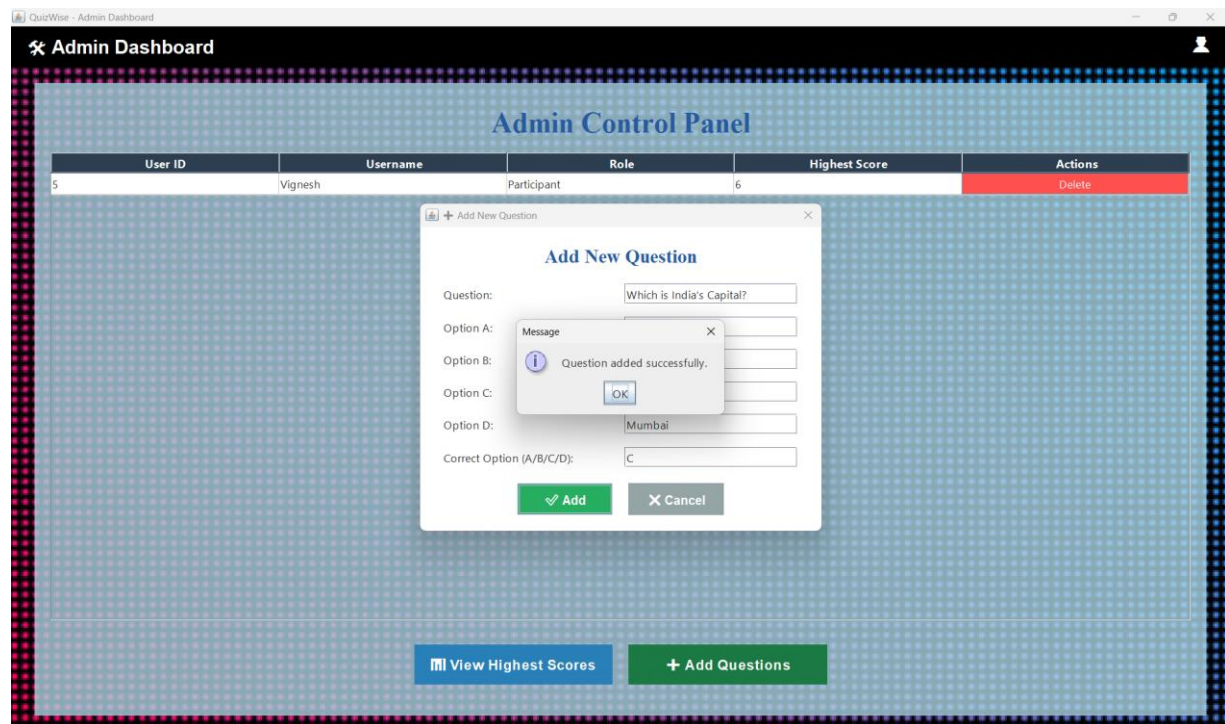
A confirmation dialog box is open in the center of the screen. It has a title "Confirm" and a question "Delete this user?". There are two buttons: "Yes" and "No".





## ADD NEW QUESTION





Individual contributions of the team members:

Name	Module worked on
Deepak	User Authentication and Role based Login
Trishita	Quiz Module with Different Scoring Types
Vandana	View Scores and Scoreboard
Vignesh	Admin Dashboard and Question Management