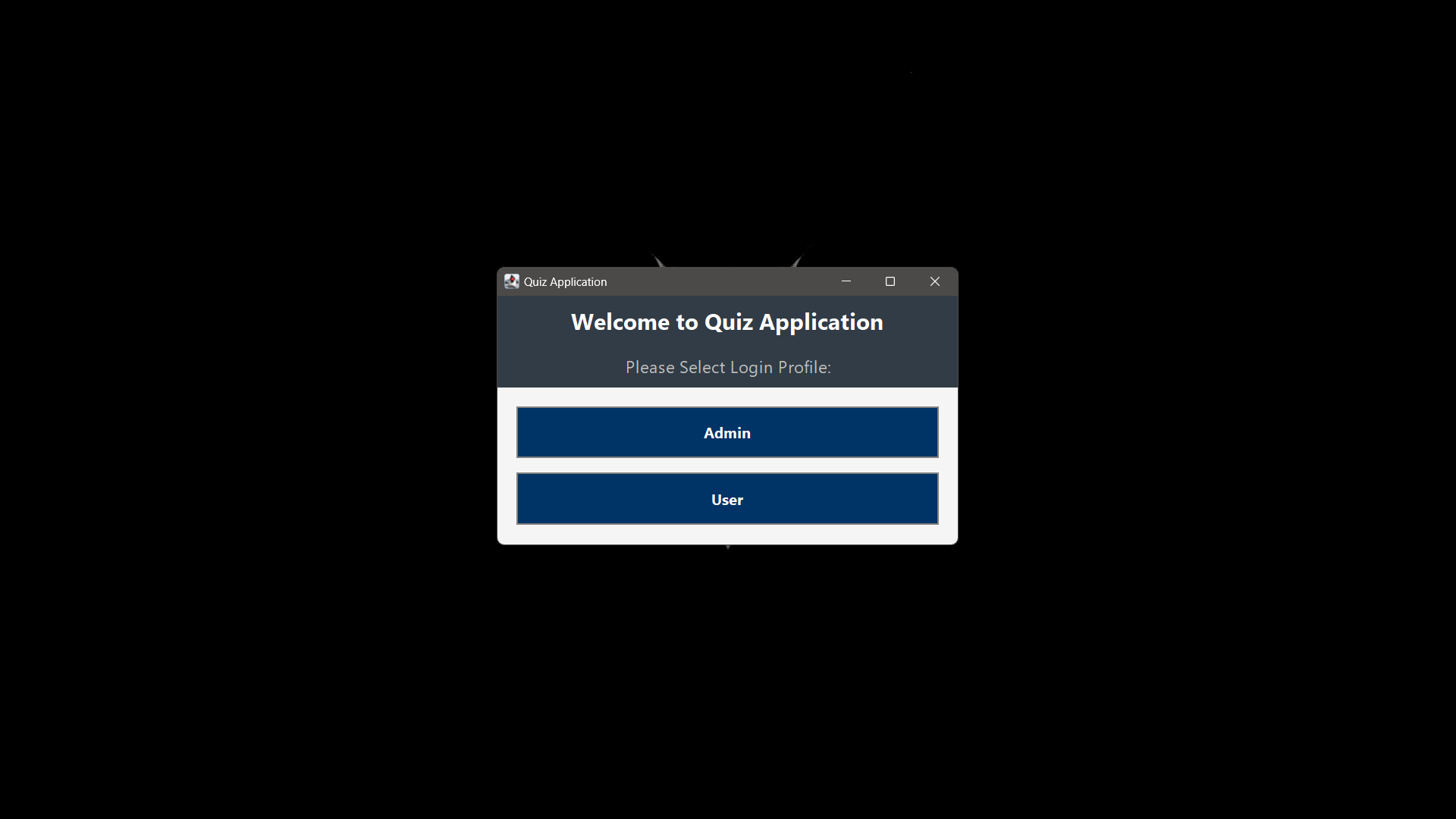
**Project Documentation : Quiz Application**

****

**Introduction**

**1.1. Project Overview**

The Quiz Application is a Java Swing-based GUI application designed for managing and administering quizzes. It allows administrators to add, delete, and view quiz questions, along with their options and answers. The application features a sophisticated and elegant user interface with functionalities to support quiz management efficiently.

**1.2. Objectives**

* Provide a user-friendly interface for managing quiz questions.
* Allow the addition of new questions with multiple options and a correct answer.
* Enable the deletion of questions based on unique identifiers.
* Display all questions in a readable format.

**1.3. Technologies Used**

* **Programming Language:** Java
* **GUI Framework:** Java Swing
* **Database:** SQL
* **IDE:** (Specify the IDE used, e.g., IntelliJ IDEA, Eclipse)

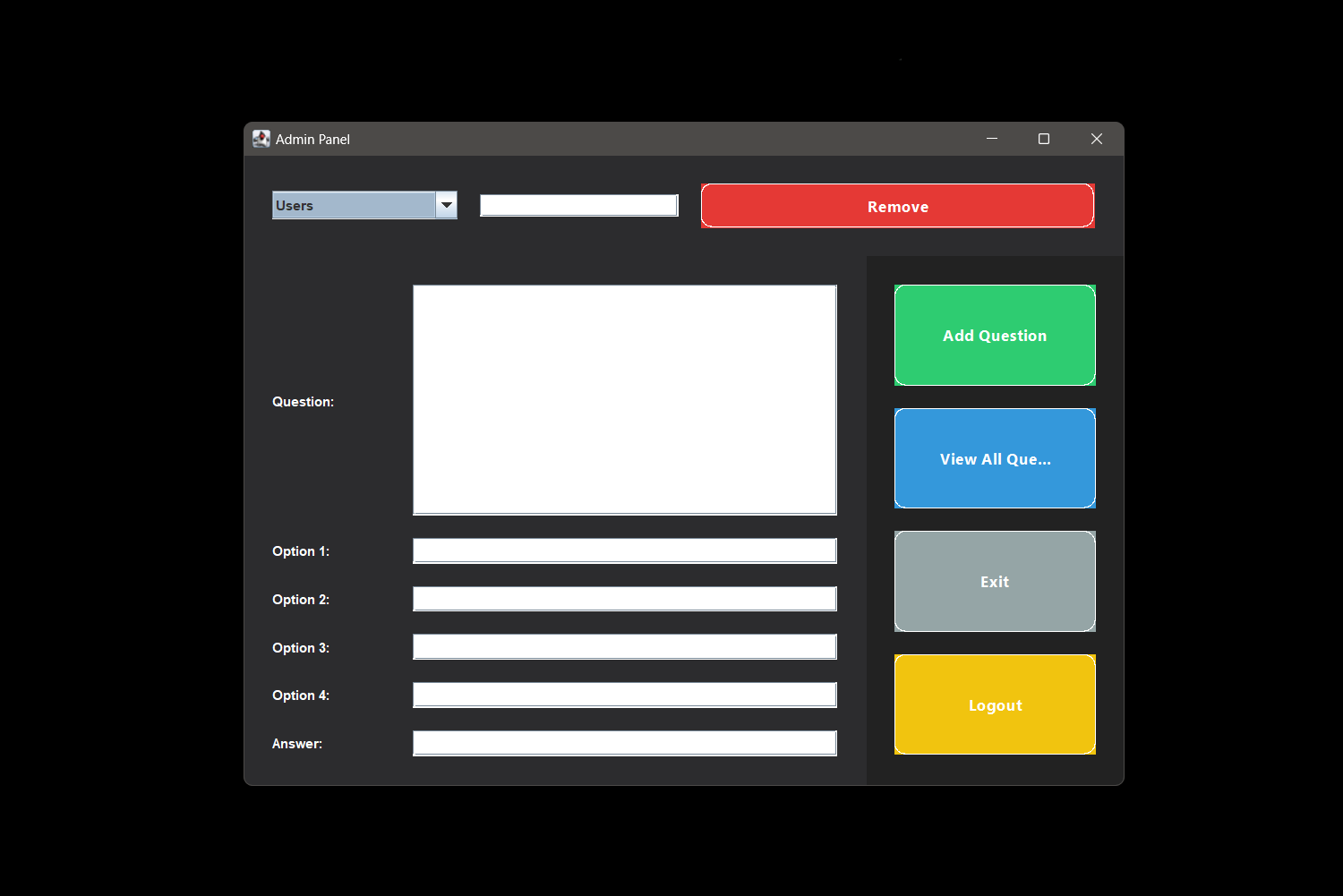
**2. System Architecture**

**2.1. Overview**

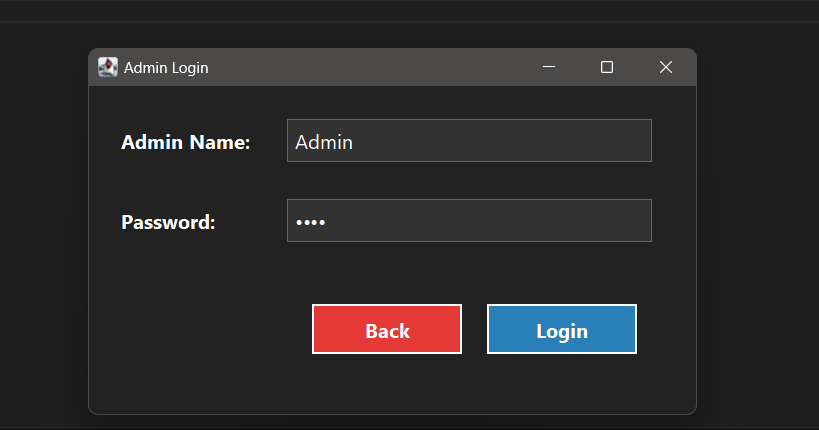
The system consists of a client-side application with a graphical user interface (GUI) built using Java Swing. It interacts with a database to perform CRUD (Create, Read, Update, Delete) operations on quiz questions.

**2.2. Components**

* **AdminPanel2:** The main interface for administrators, including functionality for adding, deleting, and viewing questions.



* **DataBase:** A class handling database operations for adding and deleting questions.
* **Question:** A class representing a quiz question with its options and answer.
* **AdminLogin2:** The login screen for administrators (mentioned in logout functionality).



**3. User Interface Design**

**3.1. Main Panel**

The AdminPanel2 JFrame serves as the main panel with a sophisticated design:

* **Background Color:** Light grey for main panel, dark grey for delete panel, and white for input and buttons panels.
* **Layouts:**
  + **BorderLayout** for overall structure.
  + **GridBagLayout** for precise control over component placement within panels.

**3.2. Panels**

* **Delete Panel:**
  + **Purpose:** Allows the removal of questions or users.
  + **Components:** JComboBox for selection, JTextField for ID input, and JButton for the delete action.
  + **Background Color:** Dark grey.
* **Input Panel:**
  + **Purpose:** Provides fields for entering new quiz questions and options.
  + **Components:** JTextArea for the question, JTextFields for options and answers.
  + **Background Color:** White.
* **Buttons Panel:**
  + **Purpose:** Contains action buttons for adding questions, viewing all questions, exiting the application, and logging out.
  + **Components:** JButton with different actions.
  + **Background Color:** White.

**3.3. Custom Styling**

* **Buttons:** Styled with rounded borders, color transitions on hover, and a modern look.
* **Text Areas:** Configured to handle text wrapping and word wrapping for readability.

**4. Functionality**

**4.1. Add Question**

* **Functionality:** Allows administrators to add a new quiz question with multiple options and the correct answer.
* **Components Involved:** JTextArea for question input, JTextFields for options, and JTextField for the answer.
* **Operation:** The input is validated and added to the database through DataBase.addQuestion().

**4.2. Delete Question**

* **Functionality:** Enables the removal of a quiz question based on its unique identifier.
* **Components Involved:** JComboBox for selection (Users/Question), JTextField for ID input, and JButton for delete action.
* **Operation:** The ID and type are passed to DataBase.delete() for deletion.

**4.3. View All Questions**

* **Functionality:** Displays all stored quiz questions in a separate window.
* **Components Involved:** JTextArea for displaying questions, JScrollPane for scrollable view.
* **Operation:** Questions are fetched from the database and displayed in a formatted manner.

**4.4. Logout and Exit**

* **Logout:** Closes the current frame and opens the login screen (AdminLogin2).
* **Exit:** Terminates the application.

**5. Database Interaction**

**5.1. DataBase Class**

Handles CRUD operations for quiz questions:

* **addQuestion(String question, String[] options, String answer)**
  + Adds a new question to the database.
* **delete(String id, String type)**
  + Deletes a record (question or user) based on the provided ID and type.
* **getQuestionAns()**
  + Retrieves all questions and their options from the database.

**5.2. Database Schema**

* **Table Name:** Questions (assumed)
* **Columns:** ID, QuestionText, Option1, Option2, Option3, Option4, Answer

**6. Error Handling**

**6.1. Exceptions**

* **SQLException:** Handled during database interactions with user-friendly messages displayed.
* **General Errors:** Displayed using JOptionPane for user feedback.

**7. User Guide**

**7.1. Starting the Application**

* Launch the application from the IDE or executable jar file.
* Log in using the provided credentials (if applicable).

**7.2. Using the Admin Panel**

* **Add Question:** Fill out the fields and click "Add Question".
* **Delete Question:** Select the item type, enter the ID, and click "Remove".
* **View Questions:** Click "View All Questions" to see a list of all questions.
* **Logout:** Click "Logout" to return to the login screen.
* **Exit:** Click "Exit" to close the application.

**8. Future Enhancements**

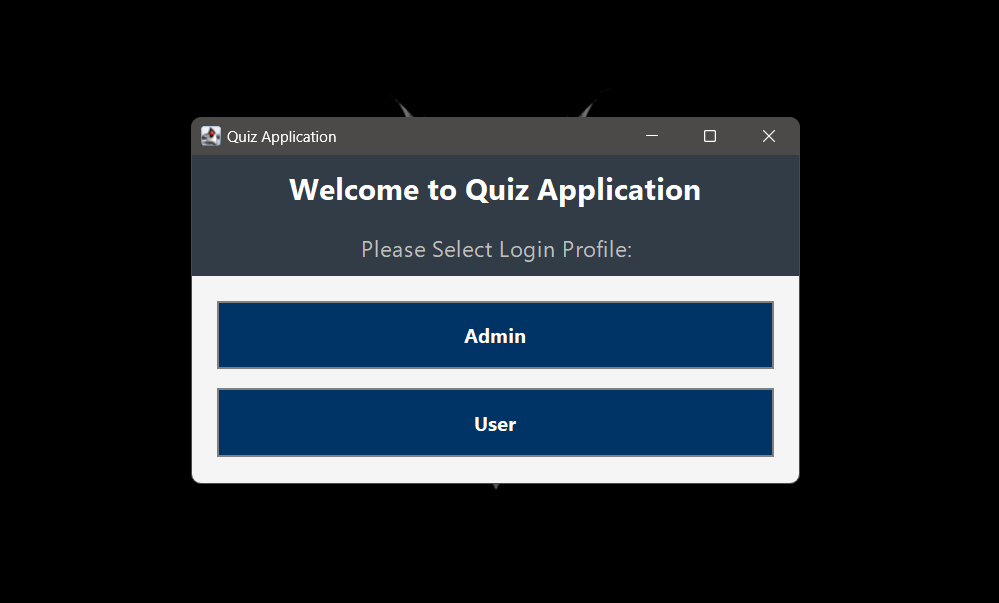
**8.1. Possible Improvements**

* **User Authentication:** Implement more robust authentication for admin users.
* **Question Categories:** Add support for different categories of questions.
* **Statistics and Reporting:** Include functionality to view statistics and reports on quizzes.

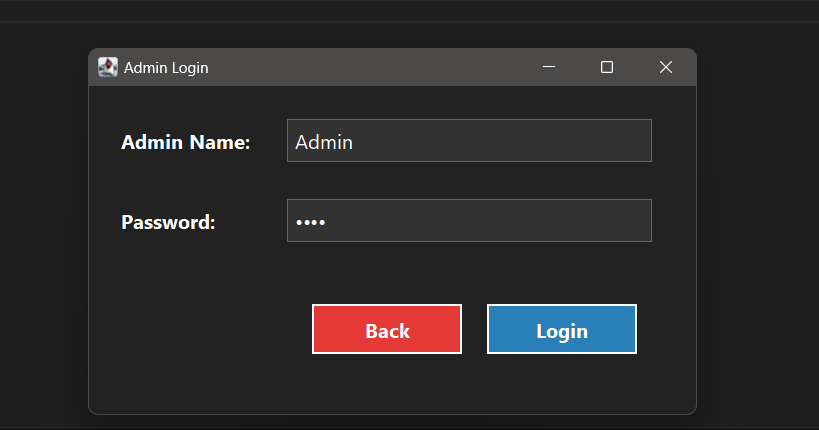
**8.2. Maintenance**

* Regular updates to fix bugs and improve performance.
* Periodic review of database schema and application functionality.

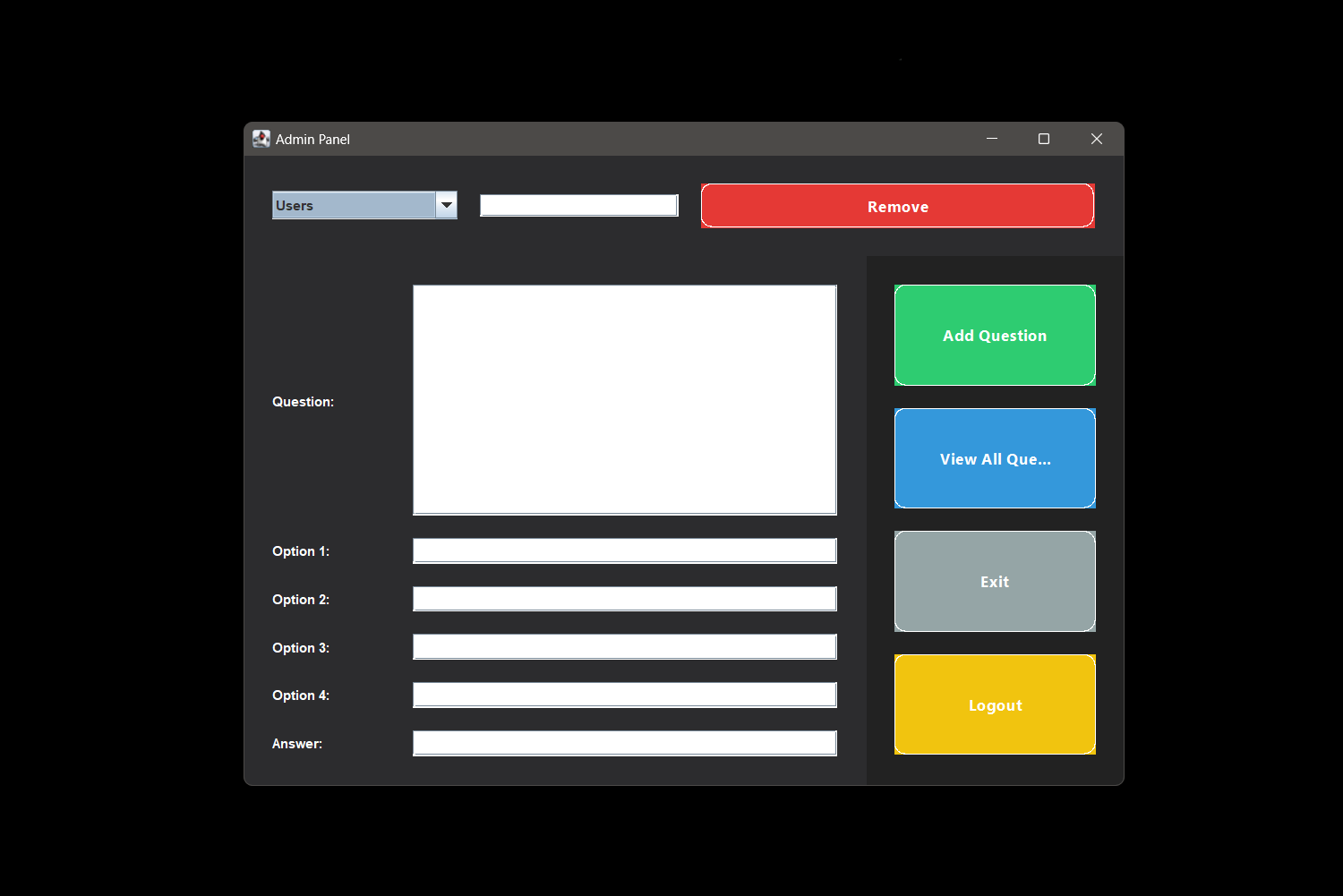
**Proof Of Work  
  
1. Welcome Screen**

****

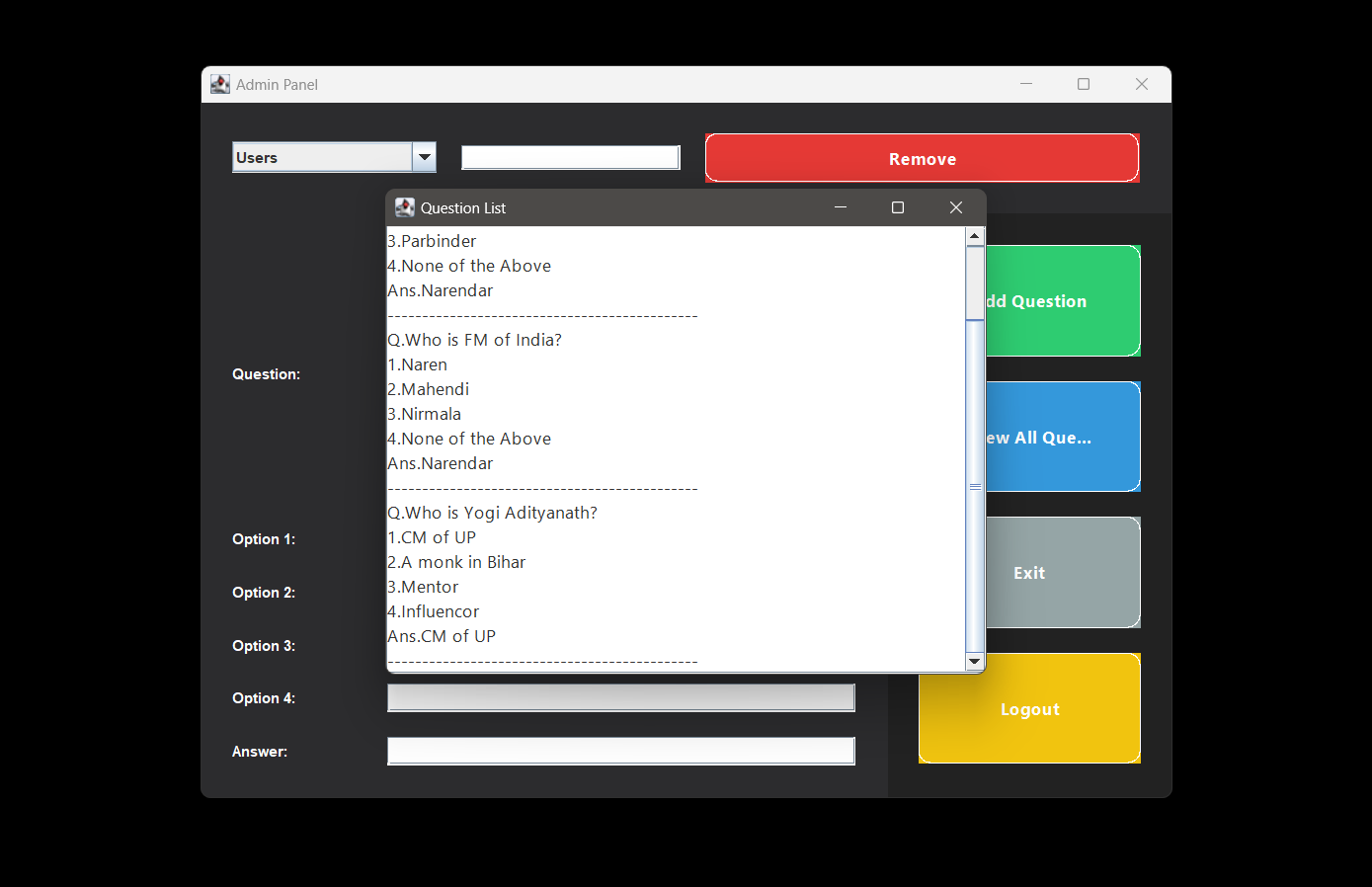
**2. Admin Login Screen (below)**

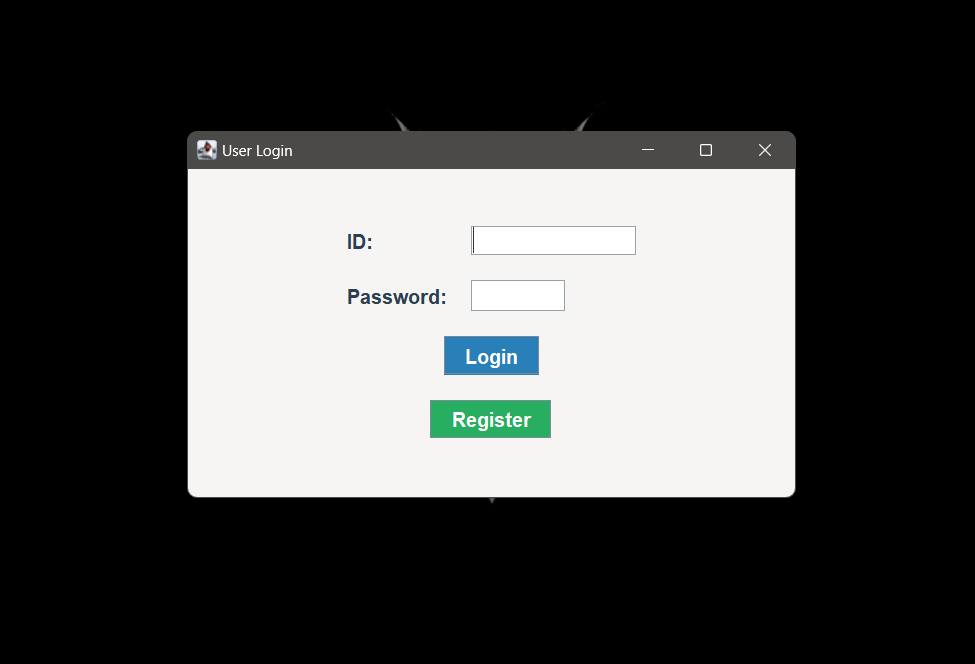


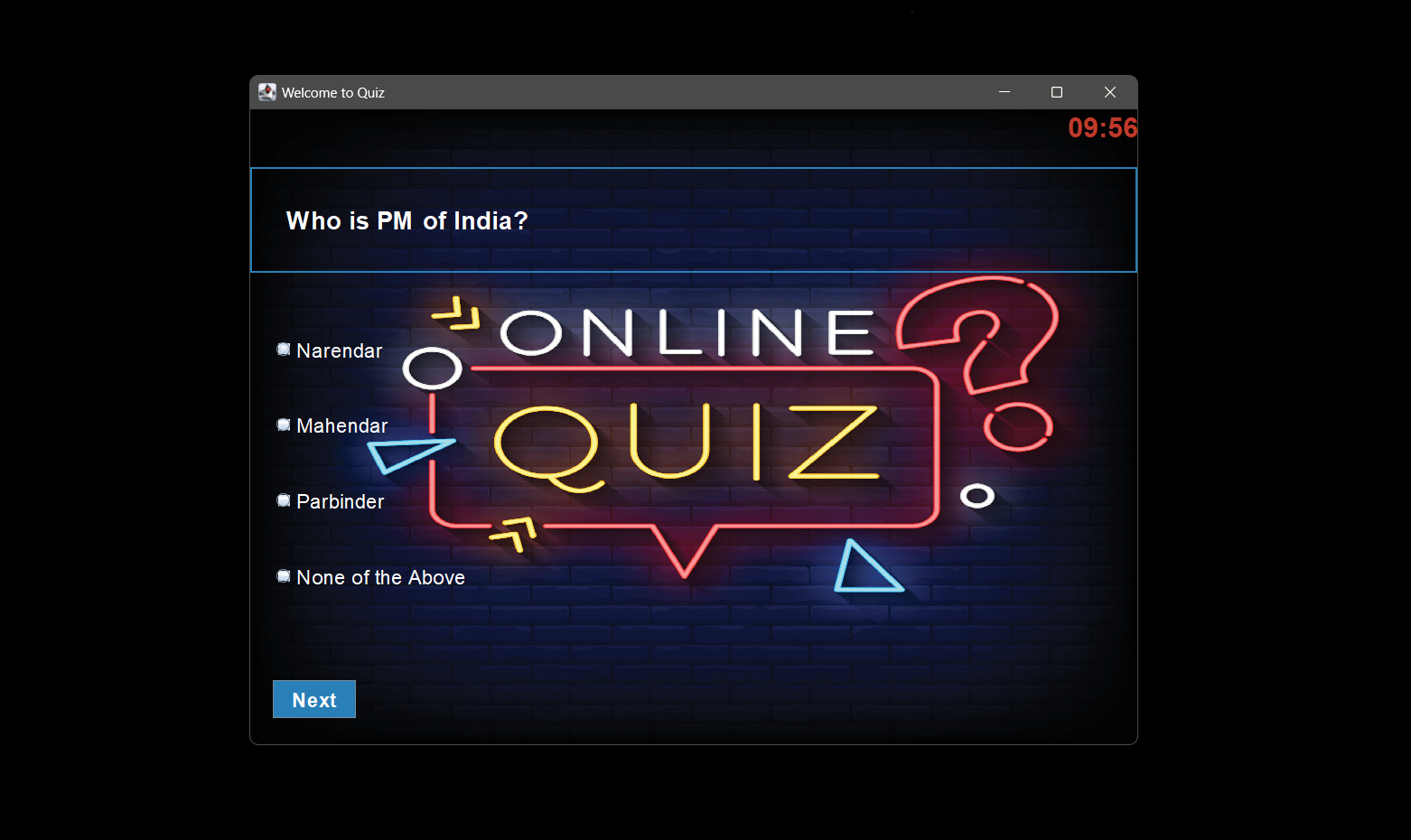
**3. Admin Panel**



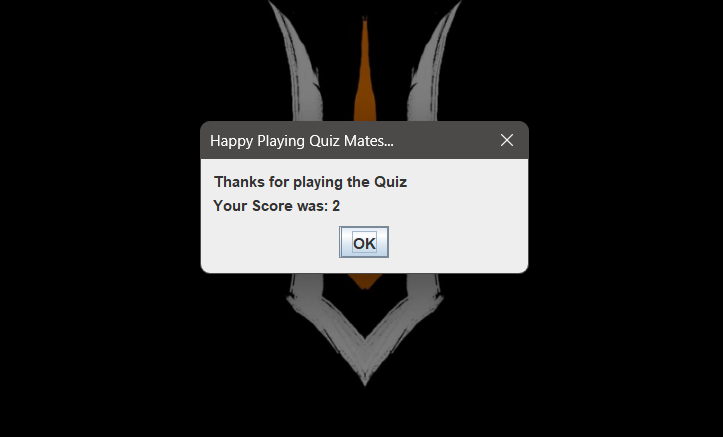
**4. Admin Panel [Questions List] (below)**



**5. User Login**  
  


**6. Quiz Zone**

**7. End**

****