QUIZ MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

DHARSHINI.H	231001035
DILSHATH SHAFANA	231001038

In partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

DEPARTMENT OF INFORMATION TECHNOLOGY

RAJALAKSHMI ENGINEERING COLLEGE

November 2024

BONAFIDE CERTIFICATE

Certified that this project titled "**Quiz management system**" is the bonafide work of **Dharshini.H (231001035), Dilshath Shafana (231001038)** who carried out the project work under my supervision.

SIGNATURE
Dr. P.Valarmathie
HEAD OF THE DEPARTMENT
Department of Information Technology
Rajalakshmi Engineering College,

SIGNATURE
Mrs.Usha S
COURSE INCHARGE
Department of Information Technology
Rajalakshmi Engineering College

This project is submitted for CS23333 –Object oriented programming using Java held on _____

INTERNAL EXAMINAR

EXTERNAL EXAMINAR

TABLE CONTENTS

1.1Abstract

The Quiz Management System (QMS) is designed to provide a seamless platform for creating, managing, and participating in quizzes. Utilizing Java and JDBC for database connectivity, the system ensures data reliability and security, allowing administrators to easily manage quizzes and track user performance. With two levels of quiz difficulty (Easy and Hard), the system is tailored for various user skill levels. A secure admin interface enables the addition and editing of questions, making QMS adaptable for ongoing educational needs and entertainment.

1.2 Introduction

QMS is a user friendly platform that simplifies quiz administration and user interaction. Administrators can securely log in to manage quiz content, add new questions, and monitor quiz participation. Users, in turn, can select their preferred quiz difficulty level, providing an accessible learning tool that supports both basic and advanced assessments. The system is built using Java for the frontend and SQL for backend data management, ensuring smooth data transactions and easy maintenance.

1.3 Purpose

The purpose of QMS is to streamline quiz creation and management, allowing:

- **Quiz Creation and Management**: Administrators can set up and adjust quizzes quickly.
- **Question Bank Expansion**: Admins can add new questions to keep the quizzes dynamic.
- **Difficulty Selection**: Users can choose between Easy and Hard levels based on their skill level.
- **Secure Access**: Admins and users have login protected access for data integrity.
- **Performance Tracking**: Users can track their scores, while admins can monitor quiz performance for future analysis.

1.4 Scope of the Project

The envisioned Quiz Management System (QMS) aims to provide seamless interaction for administrators and effective fulfilment of all core functionalities. Built using Java (JDBC) with a SQL database, the system ensures efficient management of quiz content and user data, reducing response times for quiz taking and administration tasks. This project addresses the challenges associated with traditional quiz management by offering a structured platform that stores comprehensive information on quizzes, questions, and user performance. With a focus on features like user verification, data validation, and an intuitive interface, the system aims to optimize the quiz taking experience for users while providing administrators with reliable tools for managing and analysing quiz data. The scope of QMS is performance tracking, making it suitable for educational and training environments.

1.5 Software Requirement Specification

Introduction

The Quiz Management System (QMS) is designed to handle all aspects of quiz administration and participation, including quiz creation, question management, and user performance tracking. QMS serves as an automated solution to traditional quiz processes, enhancing efficiency in educational and training settings.

Document Purpose

This SRS document outlines the software requirements for the Quiz Management System, detailing design decisions, architectural structure, and necessary specifications for successful implementation. It provides insight into the system's overall design and is a valuable reference for ongoing support and future development.

Product Scope

The Quiz Management System is developed for broad applicability in educational and training environments, aiming to replace manual quiz processes. QMS simplifies quiz management, delivering a complete solution for administrators. It provides versatility by allowing easy editing of quiz content, supports two levels of difficulty, and enhances the overall quiz experience for users.

References and Acknowledgement

[1] https://www.javatpoint.com/java awt

[2] https://www.javatpoint.com/java swing

Overall Description

The Quiz Management System provides an interactive platform for users to test their knowledge through quizzes of varying difficulty levels. The system features two distinct difficulty modes (Easy and Hard), allowing users to choose according to their comfort level. A secure admin portal enables authorized personnel to monitor quiz participation statistics and manage the question database. The system streamlines the entire quiz taking process while maintaining detailed records of user attempts and performance metrics.

Product Perspective

Utilizing a client/server architecture, the system is designed to be compatible with the Microsoft Windows Operating System. The front end is developed using Java AWT and SWING, featuring:

User Interface Components

- Quiz difficulty selection (Easy/Hard)
- Immediate score display
- User friendly navigation

Admin Interface Components

- Secure login portal
    - Performance metrics by difficulty level
    - Success rates for each difficulty

The backend leverages the SQL server for efficient data management, handling:

- Question bank for both difficulty levels
- User attempt records and scores
- Admin authentication credentials

This architecture ensures a seamless quiz taking experience for users while providing administrators with comprehensive oversight tools. The system's design allows for easy maintenance of question banks and efficient tracking of user performance across both difficulty levels.

Product Functionality

- Admin Login: Enables existing administrators to log in securely.
- User signup: Facilitates the addition of new users.
- user: Allows users to attempt the quiz in two difficulty levels
- Admin interface: displays the phone number, name and the difficulty level

User and Characteristics

Qualification: Users should have basic interface navigating knowledge

Experience: Familiarity with interfaces such a google forms

Technical Experience: Users are expected to have elementary knowledge of computers for optimal system interaction.

Operating Environment:

Hardware Requirements

- Processor: Any Processor over i3
- Operating System: Windows 8, 10, 11
- Processor Speed: 2.0 GHz
- RAM: 4GB
- Hard Disk: 500GB

Software Requirements

- Database: SQL
- Frontend: Java (SWING, AWT)
- Technology: Java (JDBC)

Constraints

- Access to system administration features is limited to authorized administrators.
- The delete function is restricted to administrators and lacks additional confirmation prompts for simplicity, placing responsibility on administrators to avoid unintended deletions.
- Administrators must exercise caution when deleting questions to maintain the consistency and integrity of the question bank.

Assumptions and Dependencies

- System administrators will securely create and communicate login IDs and passwords to users as necessary.

User Interface

the Quiz Management System provides a menu driven interface that includes:

1. **Admin Login**: Allows existing administrators to log in.
2. **Add Question**: Permits administrators to add new questions to the quiz database.
3. **View Questions**: Enables admins to view and modify existing questions.
4. **Delete Question**: Allows administrators to delete outdated or irrelevant questions.
5. **View User Results**: Admins can view performance metrics and quiz results by difficulty level.

performance records. **Hardware Interface**

- Requires a screen resolution of at least 640 x 480.
- Compatible with Windows OS (8, 10, and 11).

Software Interface

- MS Windows Operating System
- Java AWT and SWING for designing the front end
- SQL for the backend
- Platform: Java Language
- Integrated Development Environment (IDE):VS Code

Functional Requirements

- Login **Module (LM)**
  - Provides secure access for administrators via a username and password.
  - Passwords are masked for confidentiality.
  - Successful login verification is required to gain system access.
- Question **Management Module (QMM)**
  - Allows administrators to add, view, update, and delete questions.
  - Supports two levels of difficulty (Easy and Hard) to categorize questions accordingly.
  - Ensures consistency in data entry to maintain quiz quality and relevance.
- Administrator **Module (AM)**
  - Upon login, administrators gain access to core administrative functions.
  - Includes adding new questions, updating existing ones, and removing irrelevant entries.
  - Each action triggers communication with the Server Module (SM) for database updates.
- Server **Module (SM)**
  - Acts as an intermediary between the user interface and the database.
  - Processes and validates requests from various modules.
  - Ensures data consistency and integrity in managing question banks and quiz

Non-functional Requirements

- Performance
  - The system should handle quiz question retrieval and performance tracking efficiently, maintaining a response time of under 2 seconds.

- Reliability
  - The system must be robust, with immediate measures in place for system recovery in case of failure.
- Availability
  - Under normal conditions, requests for accessing and updating quiz content should be processed within 2 seconds for a seamless experience.
- Security
  - A strong security protocol is required to prevent unauthorized access, protecting user and quiz data.
- Maintainability
  - Design documentation should be provided for ongoing maintenance, ensuring system adaptability and allowing for updates on both the frontend and backend.

2.System Flow Diagrams

2.1 Use Case Diagrams

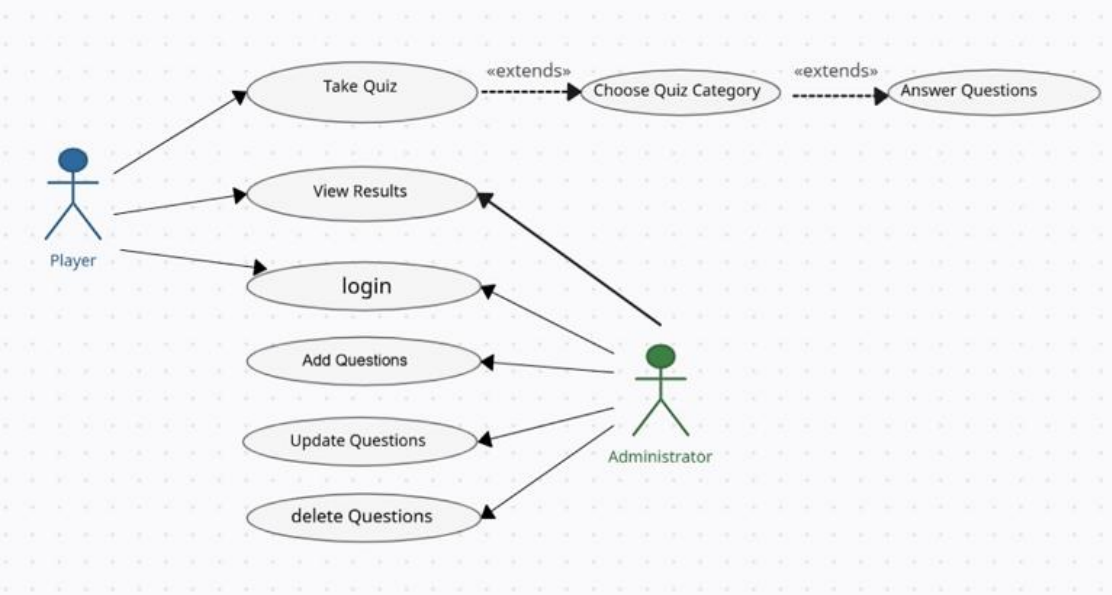

Figure 2.1.1 Use Case Diagrams

2.2 Entity relationship diagram

ER (Entity Relationship) Diagram is used to represents the relationship between entities in the table.

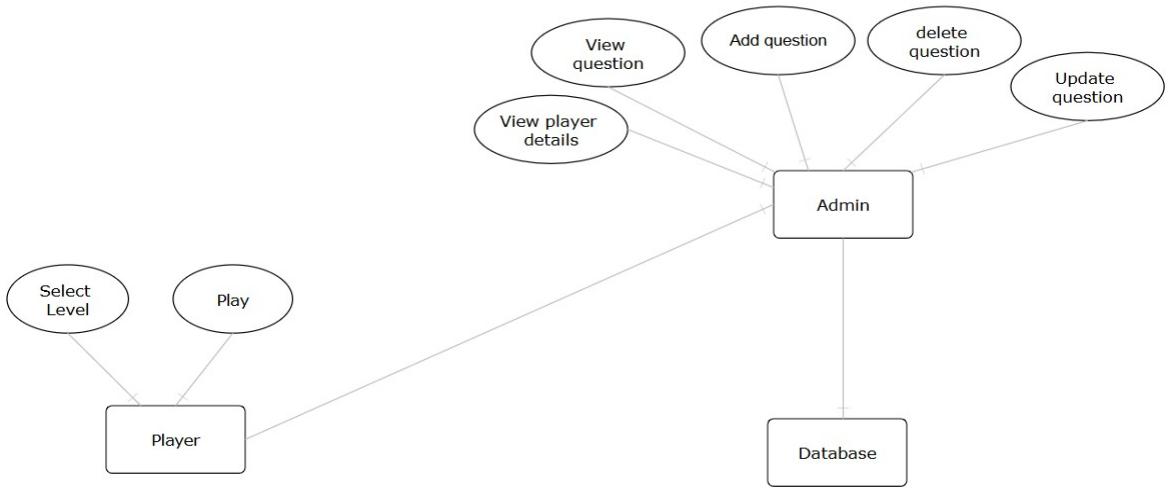

Figure 2.2.1 Entity relationship diagram

2.3 Data flow diagram

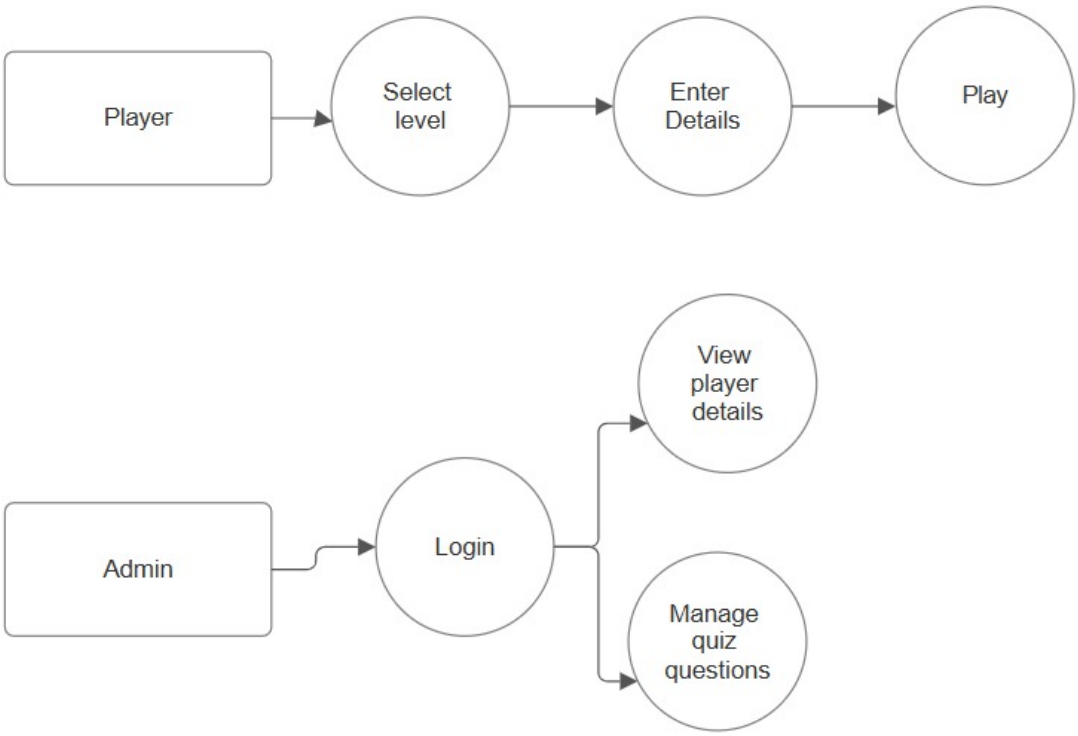

Figure 2.3.1 Data flow diagram

3.Module description

Admin:

1. **Login:**
    - Admins can log in using their registered username and password to access the system.
2. **After Login:**
    - Once logged in, the admin has access to the following functionalities:
    - **Add Question:**
        - Admins can add new quiz questions to the system, specifying the question text, answer options, and the correct answer. Questions can also be assigned a difficulty level (Easy or Hard) to provide a varied quiz experience.
    - **View Questions:**
        - Admins can view the list of existing quiz questions, organized by difficulty level. They have the ability to update question details, including modifying answer choices or changing the correct answer.
    - **Delete Question:**
        - Admins can delete questions that are outdated or irrelevant, ensuring the quiz content stays up to date and relevant to users.
    - **View User Results:**
        - Admins can access performance metrics for each user, reviewing scores and completion times. This helps in monitoring quiz difficulty effectiveness and user engagement.

4. Design

Home Page

Figure 4.1 Home page

User login

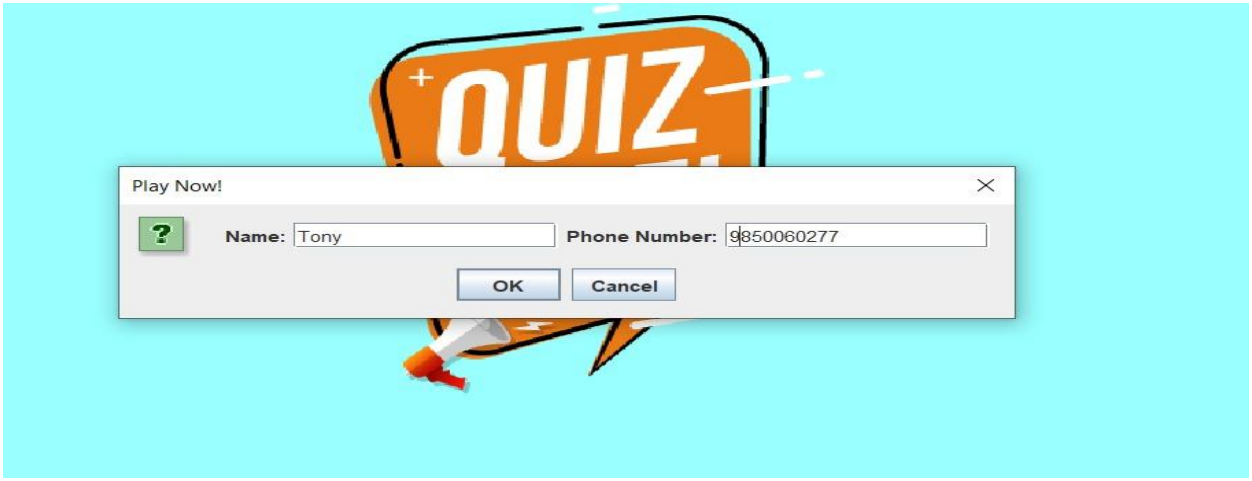

Figure 4.2 User login

Correct Answer

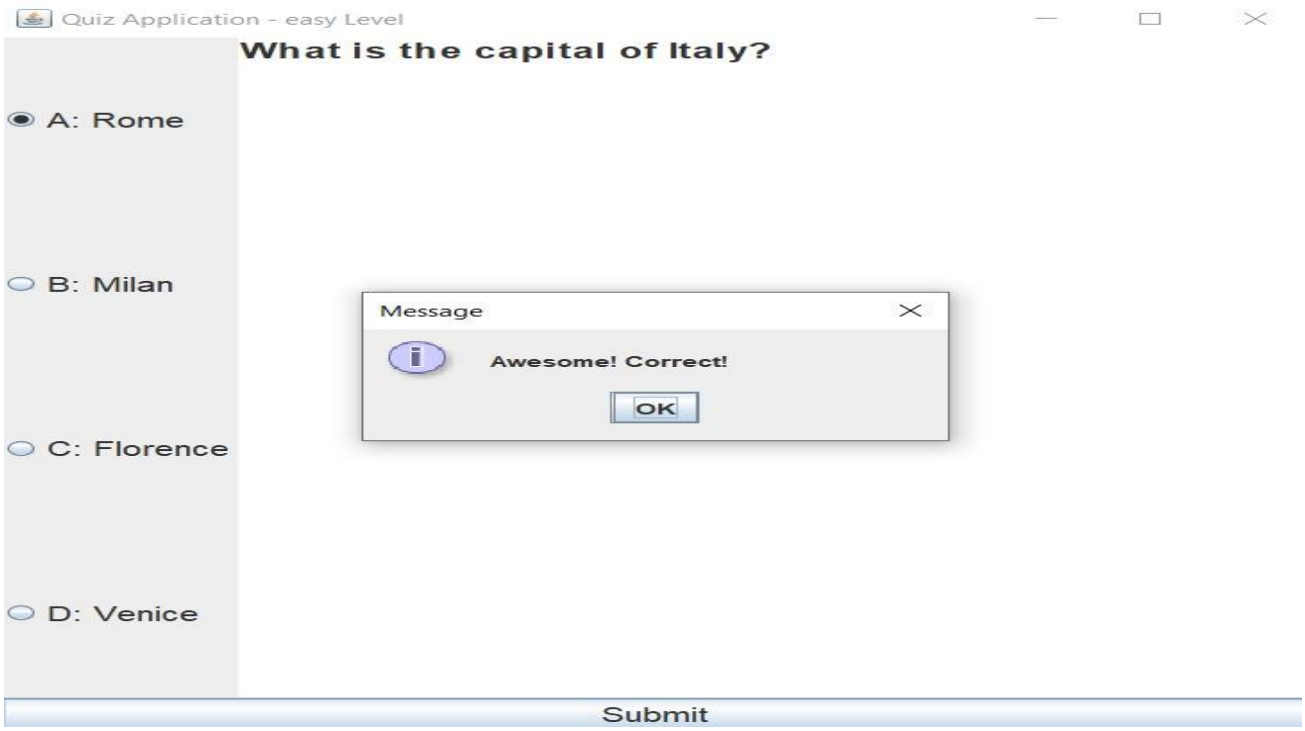

Figure 4.3 Correct Answer

Wrong Answer

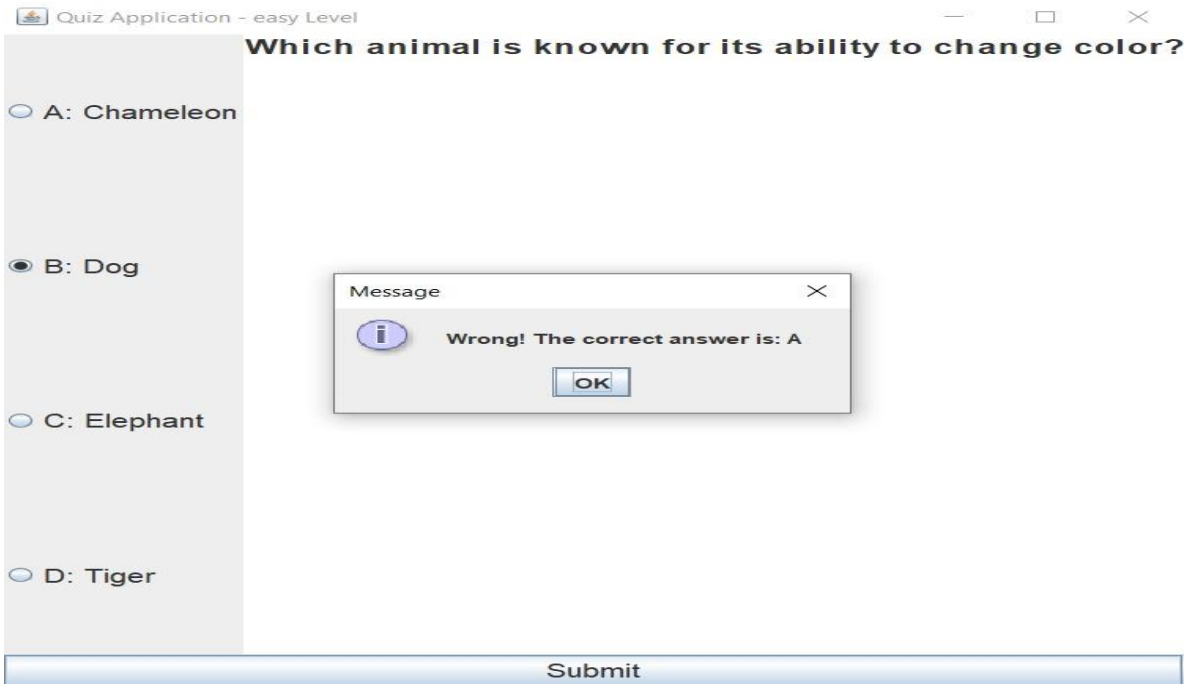

Figure 4.4 Wrong Answer

Quiz Results

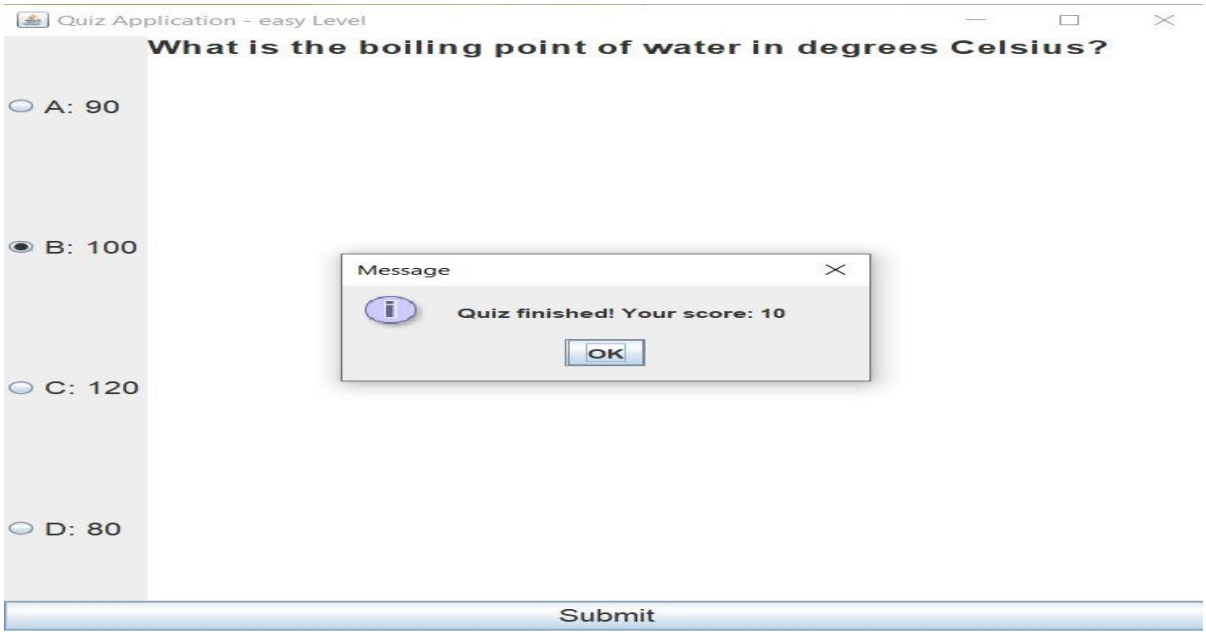

Figure 4.5 Quiz Results

Admin login

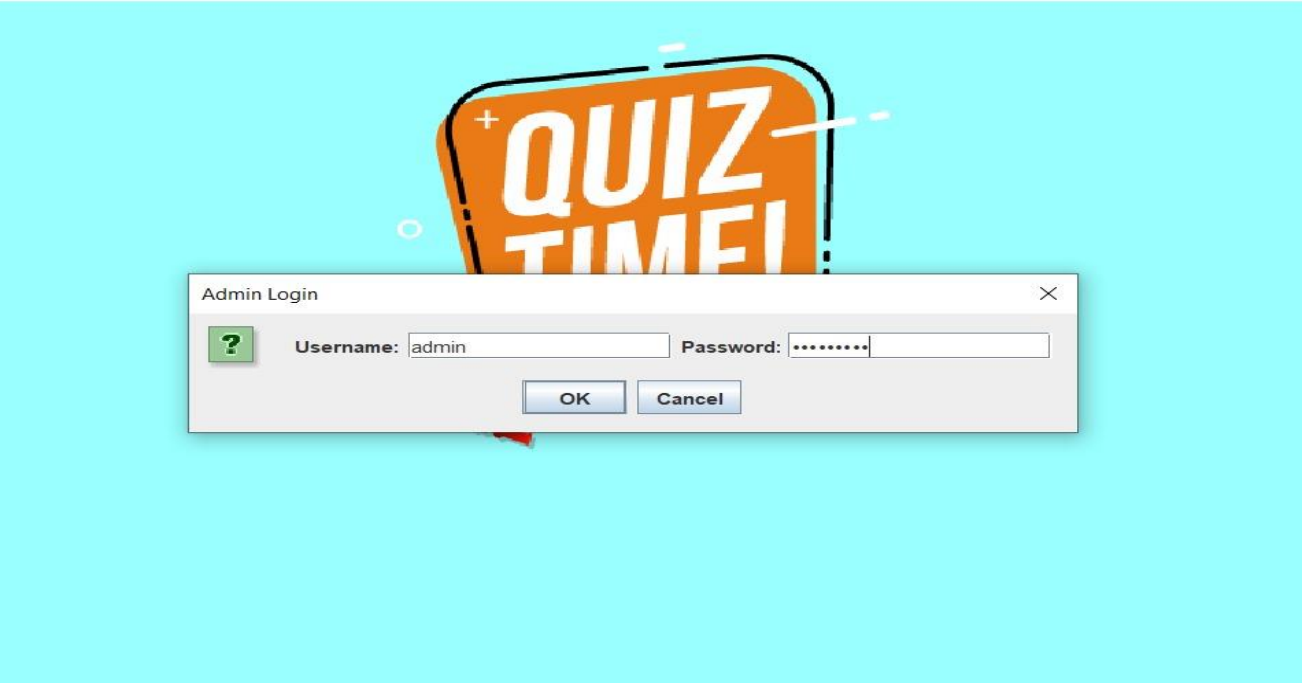

Figure 4.6 Admin login

Admin interface

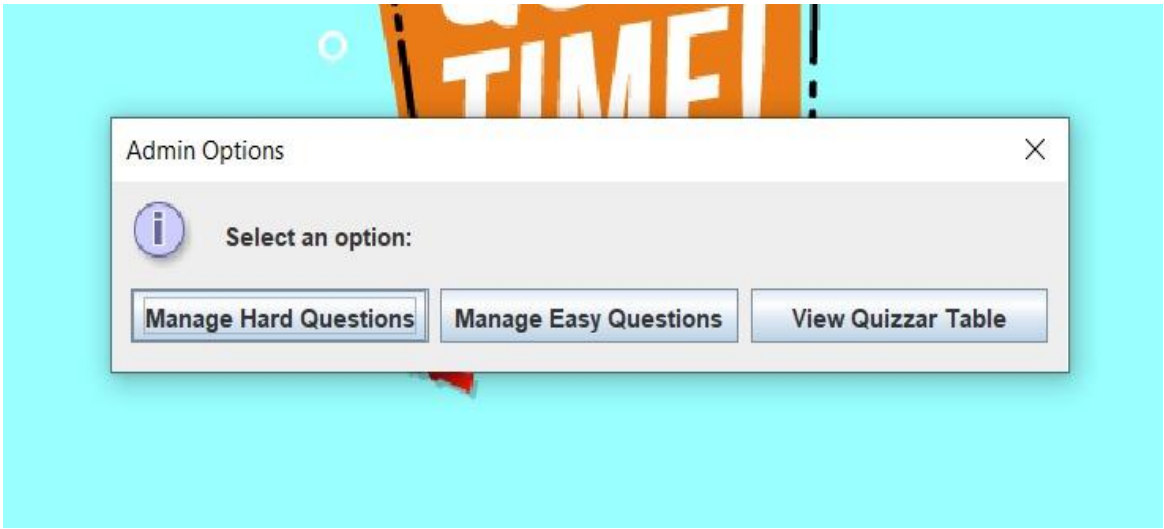

Figure 4.7 Admin login

Admin Manage interface

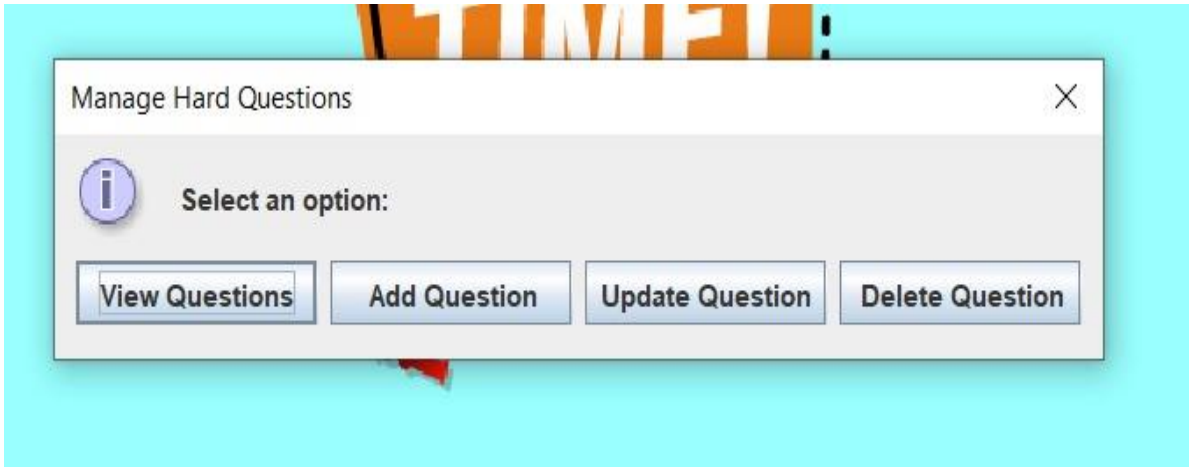

Figure 4.8 Admin Manage interface

Add Question

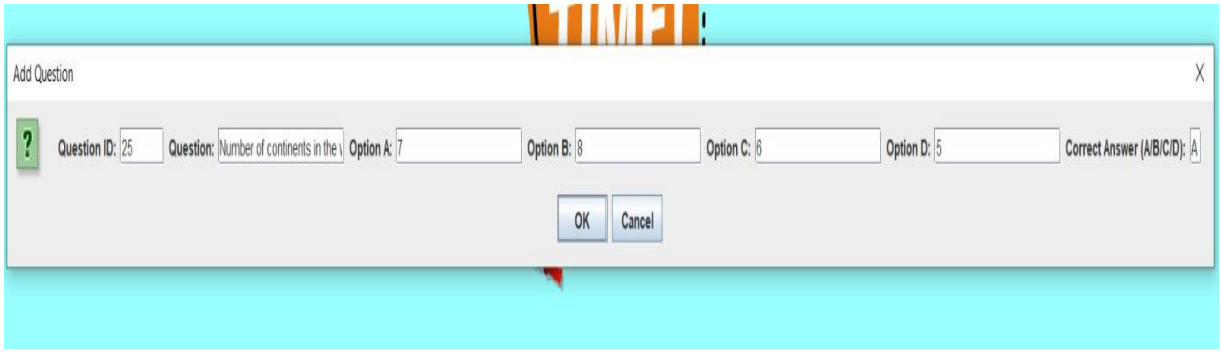

Figure 4.9 Add Question

Add Question Result

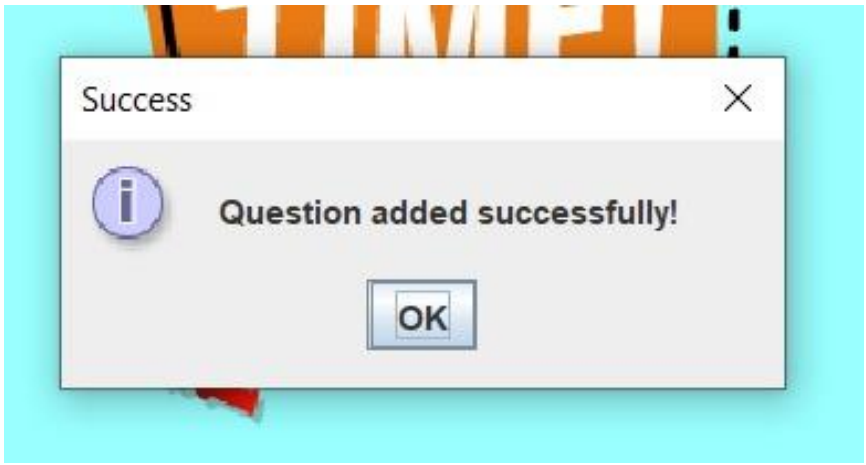

Figure 4.10 Add Question Result

Delete Question

Figure 4.11 Delete Question

Delete Question Result

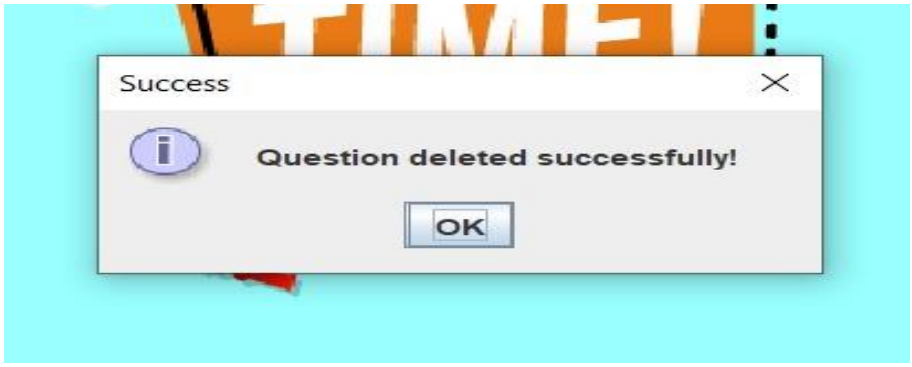

Figure 4.12 Delete Question Result

4.2 Database Design:

The data for the Quiz Management System will be stored and retrieved from a relational database. The database design phase is crucial in ensuring efficient and effective data handling. The data elements and structures necessary for the system have been identified during the analysis phase and organized accordingly for optimal storage and retrieval and efficient access for multiple users. The objective is to design the database so that it provides fast, cost effective, and flexible access to data. Relationships between the data items are established to ensure logical connections between the various components of the quiz, such as questions, users, and performance records.

QID	Question	A	B	C	D	Correct Ans
1	What is the main function of the chloroplast in plant cells?	Cellular respiration	Photosynthesis	Energy production	Protein synthesis	Photosynthesis
2	Who is known as the father of modern physics?	Isaac Newton	Albert Einstein	Niels Bohr	Galileo Galilei	Albert Einstein
3	What is the chemical formula for table salt?	NaCl	KCl	CaCl2	MgCl2	NaCl
4	What type of bond involves the sharing of electron pairs between atoms?	Ionic bond	Covalent bond	Metallic bond	Hydrogen bond	Covalent bond
5	In which year did the Berlin Wall fall?	1987	1989	1991	1993	1989
6	What is the capital city of Japan?	Tokyo	Kyoto	Osaka	Hiroshima	Tokyo
7	Which law states that for every action, there is an equal and opposite reaction?	Newton's First Law	Newton's Second Law	Newton's Third Law	Law of Conservation of Energy	Newton's Third Law
8	What is the process by which plants lose water vapour through small openings called stomata?	Photosynthesis	Transpiration	Respiration	Evaporation	Transpiration
9	What is the largest organ in the human body?	Liver	Heart	Skin	Brain	Skin
10	What economic theory advocates for minimal government intervention in the economy?	Keynesianism	Mercantilism	Laissez-faire	Socialism	Laissez-faire

Table 4.2 Database Design

4.3 IMPLEMENTATION CODE:

```
//JDBC Connection to Java
import java.sql.*;
public class DBcon {
private static final String URL =
"jdbc:oracle:thin:@localhost:1521:xe";
private static final String USER = "SYSTEM";
private static final String PASSWORD = "**********";
// Method to establish and return a database connection
public static Connection getConnection() {
Connection conn = null;
try {
Class.forName("oracle.jdbc.driver.OracleDriver");
conn = DriverManager.getConnection(URL, USER, PASSWORD);
} catch (ClassNotFoundException | SQLException e) {
e.printStackTrace();
}
return conn;
}
}
//Main welcome page
import javax.swing.*;
import java.awt.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
```

```
import java.util.List;
public class WelcomePage extends JFrame {
public WelcomePage() {
setTitle("Welcome to the Quiz Application");
setSize(1500, 600);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new BorderLayout());
getContentPane().setBackground(new Color(153, 255, 255));
// Load the image
JLabel imageLabel = new JLabel(new
ImageIcon(getClass().getResource("/quiz.png")));
imageLabel.setHorizontalAlignment(SwingConstants.CENTER);
add(imageLabel, BorderLayout.NORTH);
JPanel buttonPanel = new JPanel();
buttonPanel.setLayout(new FlowLayout());
JLabel welcomeLabel = new JLabel("Welcome to the Quiz! This is a
simple quiz for kids to check their GK. Don't worry kids, no one's judging
you! ;)");
welcomeLabel.setFont(new Font("Arial", Font.BOLD, 20));
buttonPanel.add(welcomeLabel);
JButton easyButton = new JButton("Easy Quiz");
JButton hardButton = new JButton("Hard Quiz");
JButton adminButton = new JButton("Admin");
easyButton.addActionListener(e  > showInputDialog("easy"));
hardButton.addActionListener(e  > showInputDialog("hard"));
adminButton.addActionListener(e  > showAdminLoginDialog());
buttonPanel.add(welcomeLabel);
buttonPanel.add(easyButton);
buttonPanel.add(hardButton);
buttonPanel.add(adminButton);
add(buttonPanel, BorderLayout.SOUTH);
}
private void showInputDialog(String quizLevel) {
JPanel panel = new JPanel();
JTextField nameField = new JTextField(15);
```

```
JTextField phoneField = new JTextField(15);
panel.add(new JLabel("Name:"));
panel.add(nameField);
panel.add(new JLabel("Phone Number:"));
panel.add(phoneField);
int result = JOptionPane.showConfirmDialog(this, panel, "Play Now!",
JOptionPane.OK_CANCEL_OPTION);
if (result == JOptionPane.OK_OPTION) {
String name = nameField.getText();
String phoneNumber = phoneField.getText();
// Store details in the database
if (storeUserDetails(name, phoneNumber, quizLevel)) {
QuizApp quizApp = new QuizApp(quizLevel);
quizApp.setVisible(true);
this.dispose(); // Close the welcome page
} else {
JOptionPane.showMessageDialog(this, "Error storing details.",
"Error", JOptionPane.ERROR_MESSAGE);
}
}
}
private boolean storeUserDetails(String name, String phoneNumber, String
quizLevel) {
String url = "jdbc:oracle:thin:@localhost:1521:xe";
 String user = "SYSTEM";
 String pass = "**********";
String query = "INSERT INTO SYS.Quizzar (PhoneNumber, UserName,
QuizLevel) VALUES (?, ?, ?)";
try (Connection conn = DriverManager.getConnection(url, user, pass);
PreparedStatement pstmt = conn.prepareStatement(query)) {
pstmt.setString(1, phoneNumber);
pstmt.setString(2, name);
pstmt.setString(3, quizLevel);
pstmt.executeUpdate();
return true;
```

```
} catch (Exception e) {
e.printStackTrace();
return false;
}
}
private void showAdminLoginDialog() {
JPanel panel = new JPanel();
JTextField usernameField = new JTextField(15);
JPasswordField passwordField = new JPasswordField(15);
panel.add(new JLabel("Username:"));
panel.add(usernameField);
panel.add(new JLabel("Password:"));
panel.add(passwordField);
int result = JOptionPane.showConfirmDialog(this, panel, "Admin Login",
JOptionPane.OK_CANCEL_OPTION);
if (result == JOptionPane.OK_OPTION) {
String username = usernameField.getText();
String password = new String(passwordField.getPassword());
if (username.equals("admin") && password.equals("adminpass")) {
showAdminOptions();
} else {
JOptionPane.showMessageDialog(this, "Invalid credentials.",
"Error", JOptionPane.ERROR_MESSAGE);
}
}
}
private void showAdminOptions() {
String[] options = {"Manage Hard Questions", "Manage Easy Questions",
"View Quizzar Table"};
int choice = JOptionPane.showOptionDialog(this, "Select an option:",
"Admin Options",
JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, null,
options, options[0]);
switch (choice) {
case 0:
```

```
manageHardQuestions();
break;
case 1:
manageEasyQuestions();
break;
case 2:
displayQuizzarTable();
break;
default:
break;
}
}
private void manageHardQuestions() {
String[] options = {"View Questions", "Add Question" , "Update Question",
"Delete Question"};
int choice = JOptionPane.showOptionDialog(this, "Select an option:",
"Manage Hard Questions",
JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, null,
options, options[0]);
switch (choice) {
case 0:
viewQuestions("SYS.HardQuestions");
break;
case 1:
addQuestion("SYS.HardQuestions");
break;
case 2:
updateQuestion("SYS.HardQuestions");
break;
case 3:
deleteQuestion("SYS.HardQuestions");
break;
default:
break;
}
```

```
}
private void manageEasyQuestions() {
String[] options = {"View Questions", "Add Question", "Update Question",
"Delete Question"};
int choice = JOptionPane.showOptionDialog(this, "Select an option:",
"Manage Easy Questions",
JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, null,
options, options[0]);
switch (choice) {
case 0:
viewQuestions("SYS.EasyQuestions");
break;
case 1:
addQuestion("SYS.EasyQuestions");
break;
case 2:
updateQuestion("SYS.EasyQuestions");
break;
case 3:
deleteQuestion("SYS.EasyQuestions");
break;
default:
break;
}
}
private void viewQuestions(String tableName) {
String url = "jdbc:oracle:thin:@localhost:1521:xe";
 String user = "SYSTEM";
 String pass = "**********";
String query = "SELECT QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer FROM " + tableName;
StringBuilder tableData = new
StringBuilder("ID\tQuestion\tOptionA\tOptionB\tOptionC\tOptionD\tCorrect
Answer\n");
try (Connection conn = DriverManager.getConnection(url, user, pass);
```

```
PreparedStatement pstmt = conn.prepareStatement(query);
ResultSet rs = pstmt.executeQuery()) {
while (rs.next()) {
int questionID = rs.getInt("QuestionID");
String questionText = rs.getString("QuestionText");
String optionA = rs.getString("OptionA");
String optionB = rs.getString("OptionB");
String optionC = rs.getString("OptionC");
String optionD = rs.getString("OptionD");
String correctAnswer = rs.getString("CorrectAnswer");
tableData.append(questionID).append("\t")
.append(questionText).append("\t")
.append(optionA).append("\t")
.append(optionB).append("\t")
.append(optionC).append("\t")
.append(optionD).append("\t")
.append(correctAnswer).append("\n");
}
} catch (Exception e) {
e.printStackTrace();
}
JTextArea textArea = new JTextArea(tableData.toString());
textArea.setEditable(false);
JOptionPane.showMessageDialog(this, new JScrollPane(textArea), "Questions
in " + tableName, JOptionPane.INFORMATION_MESSAGE);
}
private void addQuestion(String tableName) {
JPanel panel = new JPanel();
JTextField idField = new JTextField(5);
JTextField questionField = new JTextField(15);
JTextField optionAField = new JTextField(15);
JTextField optionBField = new JTextField(15);
JTextField optionCField = new JTextField(15);
JTextField optionDField = new JTextField(15);
JTextField answerField = new JTextField(1);
```

```
panel.add(new JLabel("Question ID:"));
panel.add(idField);
panel.add(new JLabel("Question:"));
panel.add(questionField);
panel.add(new JLabel("Option A:"));
panel.add(optionAField);
panel.add(new JLabel("Option B:"));
panel.add(optionBField);
panel.add(new JLabel("Option C:"));
panel.add(optionCField);
panel.add(new JLabel("Option D:"));
panel.add(optionDField);
panel.add(new JLabel("Correct Answer (A/B/C/D):"));
panel.add(answerField);
int result = JOptionPane.showConfirmDialog(this, panel, "Add Question",
JOptionPane.OK_CANCEL_OPTION);
if (result == JOptionPane.OK_OPTION) {
try {
int questionID = Integer.parseInt(idField.getText());
String questionText = questionField.getText();
String optionA = optionAField.getText();
String optionB = optionBField.getText();
String optionC = optionCField.getText();
String optionD = optionDField.getText();
String correctAnswer = answerField.getText().toUpperCase();
String url = "jdbc:oracle:thin:@localhost:1521:xe";
String user = "SYSTEM";
String pass = "**********";
String query = "INSERT INTO " + tableName + " (QuestionID,
QuestionText, OptionA, OptionB, OptionC, OptionD, CorrectAnswer) VALUES (?, ?,
?, ?, ?, ?, ?)";
try (Connection conn = DriverManager.getConnection(url, user,
pass);
PreparedStatement pstmt = conn.prepareStatement(query)) {
pstmt.setInt(1, questionID);
```

```
pstmt.setString(2, questionText);
pstmt.setString(3, optionA);
pstmt.setString(4, optionB);
pstmt.setString(5, optionC);
pstmt.setString(6, optionD);
pstmt.setString(7, correctAnswer);
pstmt.executeUpdate();
JOptionPane.showMessageDialog(this, "Question added
successfully!", "Success", JOptionPane.INFORMATION_MESSAGE);
}
} catch (Exception e) {
e.printStackTrace();
JOptionPane.showMessageDialog(this, "Error adding question. Ensure
QuestionID is unique.", "Error", JOptionPane.ERROR_MESSAGE);
}
}
}
private void updateQuestion(String tableName) {
String questionToUpdate = JOptionPane.showInputDialog(this, "Enter the
question to update:");
if (questionToUpdate == null || questionToUpdate.isEmpty()) return;
String url = "jdbc:oracle:thin:@localhost:1521:xe";
String user = "SYSTEM";
String pass = "**********";
String query = "SELECT * FROM SYS." + tableName + " WHERE QuestionText =
?";
try (Connection conn = DriverManager.getConnection(url, user, pass);
PreparedStatement pstmt = conn.prepareStatement(query)) {
pstmt.setString(1, questionToUpdate);
ResultSet rs = pstmt.executeQuery();
if (rs.next()) {
String currentOptions = rs.getString("OptionA") + ", " +
rs.getString("OptionB") + ", "
+ rs.getString("OptionC") + ", " +
rs.getString("OptionD");
```

```
String currentCorrectAnswer = rs.getString("CorrectAnswer");
JPanel panel = new JPanel();
JTextField questionField = new
JTextField(rs.getString("QuestionText"), 15);
JTextField optionsField = new JTextField(currentOptions, 15);
JTextField answerField = new JTextField(currentCorrectAnswer, 15);
panel.add(new JLabel("Question:"));
panel.add(questionField);
panel.add(new JLabel("Options (comma separated):"));
panel.add(optionsField);
panel.add(new JLabel("Correct Answer (A/B/C/D):"));
panel.add(answerField);
int result = JOptionPane.showConfirmDialog(this, panel, "Update
Question", JOptionPane.OK_CANCEL_OPTION);
if (result == JOptionPane.OK_OPTION) {
String newQuestion = questionField.getText();
String newOptions = optionsField.getText();
String newCorrectAnswer = answerField.getText();
String updateQuery = "UPDATE SYS." + tableName + " SET
QuestionText = ?, OptionA = ?, OptionB = ?, OptionC = ?, OptionD = ?,
CorrectAnswer = ? WHERE QuestionText = ?";
try (PreparedStatement updatePstmt =
conn.prepareStatement(updateQuery)) {
String[] options = newOptions.split(",");
updatePstmt.setString(1, newQuestion);
updatePstmt.setString(2, options[0].trim());
updatePstmt.setString(3, options[1].trim());
updatePstmt.setString(4, options[2].trim());
updatePstmt.setString(5, options[3].trim());
updatePstmt.setString(6, newCorrectAnswer);
updatePstmt.setString(7, questionToUpdate);
updatePstmt.executeUpdate();
JOptionPane.showMessageDialog(this, "Question updated
successfully!", "Success", JOptionPane.INFORMATION_MESSAGE);
}
```

```
}
} else {
JOptionPane.showMessageDialog(this, "Question not found.",
"Error", JOptionPane.ERROR_MESSAGE);
}
} catch (Exception e) {
e.printStackTrace();
}
}
private void deleteQuestion(String tableName) {
String questionToDelete = JOptionPane.showInputDialog(this, "Enter the
questionID to delete:");
if (questionToDelete == null || questionToDelete.isEmpty()) return;
String url = "jdbc:oracle:thin:@localhost:1521:xe";
String user = "SYSTEM";
String pass = "**********";
String query = "DELETE FROM " + tableName + " WHERE QuestionID = ?";
try (Connection conn = DriverManager.getConnection(url, user, pass);
PreparedStatement pstmt = conn.prepareStatement(query)) {
pstmt.setString(1, questionToDelete);
int rowsAffected = pstmt.executeUpdate();
if (rowsAffected > 0) {
JOptionPane.showMessageDialog(this, "Question deleted
successfully!", "Success", JOptionPane.INFORMATION_MESSAGE);
} else {
JOptionPane.showMessageDialog(this, "Question not found.",
"Error", JOptionPane.ERROR_MESSAGE);
}
} catch (Exception e) {
e.printStackTrace();
}
}
private void displayQuizzarTable() {
String url = "jdbc:oracle:thin:@localhost:1521:xe";
String user = "SYSTEM";
```

```
String pass = "**********";
String query = "SELECT * FROM SYS.Quizzar";
StringBuilder tableData = new
StringBuilder("PhoneNumber\tUserName\tQuizLevel\n");
try (Connection conn = DriverManager.getConnection(url, user, pass);
PreparedStatement pstmt = conn.prepareStatement(query);
ResultSet rs = pstmt.executeQuery()) {
while (rs.next()) {
String phoneNumber = rs.getString("PhoneNumber");
String userName = rs.getString("UserName");
String quizLevel = rs.getString("QuizLevel");
tableData.append(phoneNumber).append("\t").append(userName).ap
pend("\t").append(quizLevel).append("\n");
}
} catch (Exception e) {
e.printStackTrace();
}
JTextArea textArea = new JTextArea(tableData.toString());
textArea.setEditable(false);
JOptionPane.showMessageDialog(this, new JScrollPane(textArea),
"Quizzar Table", JOptionPane.INFORMATION_MESSAGE);
}
public static void main(String[] args) {
SwingUtilities.invokeLater(() > {
WelcomePage welcomePage = new WelcomePage();
welcomePage.setVisible(true);
});
}
}
//Quiz Application interface
import javax.swing.*;
import java.awt.*;
import java.util.List;
public class QuizApp extends JFrame {
private JTextArea questionArea;
```

```
private JRadioButton optionA, optionB, optionC, optionD;
private JButton submitButton;
private int currentQuestionIndex = 0;
private int score = 0;
private List<Question> questions;
public QuizApp(String level) {
setTitle("Quiz Application   " + level + " Level");
setSize(600, 600);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new BorderLayout());
ImageIcon backgroundImage = new
ImageIcon(getClass().getResource("/question.png"));
JLabel backgroundLabel = new JLabel(backgroundImage);
backgroundLabel.setLayout(new BorderLayout());
setContentPane(backgroundLabel);
// Font settings
Font questionFont = new Font("Arial", Font.BOLD, 18);
Font optionFont = new Font("Arial", Font.PLAIN, 16);
Font buttonFont = new Font("Arial", Font.PLAIN, 16);
questionArea = new JTextArea();
questionArea.setEditable(false);
questionArea.setFont(questionFont);
add(questionArea, BorderLayout.CENTER);
optionA = new JRadioButton();
optionB = new JRadioButton();
optionC = new JRadioButton();
optionD = new JRadioButton();
optionA.setFont(optionFont); optionB.setFont(optionFont);
optionC.setFont(optionFont);
optionD.setFont(optionFont);
ButtonGroup optionsGroup = new ButtonGroup();
optionsGroup.add(optionA);
optionsGroup.add(optionB);
optionsGroup.add(optionC);
optionsGroup.add(optionD);
```

```
JPanel optionsPanel = new JPanel();
optionsPanel.setLayout(new GridLayout(4, 1));
optionsPanel.add(optionA);
optionsPanel.add(optionB);
optionsPanel.add(optionC);
optionsPanel.add(optionD);
add(optionsPanel, BorderLayout.WEST);
// Submit Button
submitButton = new JButton("Submit");
submitButton.setFont(buttonFont);
submitButton.addActionListener(e  > checkAnswer());
add(submitButton, BorderLayout.SOUTH);
// Load the quiz questions
loadQuiz(level);
}
private void loadQuiz(String level) {
QuizService quizService = new QuizService();
questions = quizService.getQuestions(level);
currentQuestionIndex = 0;
score = 0;
displayNextQuestion();
}
private void displayNextQuestion() {
if (currentQuestionIndex < questions.size()) {
Question question = questions.get(currentQuestionIndex);
questionArea.setText(question.getQuestionText());
optionA.setText("A: " + question.getOptionA());
optionB.setText("B: " + question.getOptionB());
optionC.setText("C: " + question.getOptionC());
optionD.setText("D: " + question.getOptionD());
optionA.setSelected(false);
optionB.setSelected(false);
optionC.setSelected(false);
optionD.setSelected(false);
} else {
```

```
JOptionPane.showMessageDialog(this, "Quiz finished! Your score: "
+ score);
System.exit(0);
}
}
private void checkAnswer() {
if (currentQuestionIndex < questions.size()) {
Question question = questions.get(currentQuestionIndex);
String selectedAnswer = getSelectedOption();
if (selectedAnswer != null) {
if
(selectedAnswer.equalsIgnoreCase(question.getCorrectAnswer())) {
score++;
JOptionPane.showMessageDialog(this, "Awesome! Correct!");
} else {
JOptionPane.showMessageDialog(this, "Wrong! The correct
answer is: " + question.getCorrectAnswer());
}
currentQuestionIndex++;
displayNextQuestion();
} else {
JOptionPane.showMessageDialog(this, "Please select an
option.");
}
}
}
private String getSelectedOption() {
if (optionA.isSelected()) return "A";
if (optionB.isSelected()) return "B";
if (optionC.isSelected()) return "C";
if (optionD.isSelected()) return "D";
return null; // No option selected
}
public static void main(String[] args) {
SwingUtilities.invokeLater(() > {
```

```
QuizApp app = new QuizApp("easy");
app.setVisible(true);
});
}
}
//Quiz Service
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
public class QuizService {
public List<Question> getQuestions(String level) {
List<Question> questions = new ArrayList<>();
String query = level.equals("easy") ?
"SELECT * FROM SYS.EasyQuestions" :
"SELECT * FROM SYS.HardQuestions";
try (Connection conn = DBcon.getConnection();
PreparedStatement pstmt = conn.prepareStatement(query);
ResultSet rs = pstmt.executeQuery()) {
while (rs.next()) {
questions.add(new Question(rs.getInt("QuestionID"),
rs.getString("QuestionText"),
rs.getString("OptionA"),
rs.getString("OptionB"),
rs.getString("OptionC"),
rs.getString("OptionD"),
rs.getString("CorrectAnswer")));
}
} catch (SQLException e) {
e.printStackTrace();
}
return questions;
}
```

```
}
//Question
public class Question {
private int questionID;
private String questionText;
private String optionA;
private String optionB;
private String optionC;
private String optionD;
private String correctAnswer;
public Question(int questionID, String questionText, String optionA,
String optionB, String optionC, String optionD, String correctAnswer) {
this.questionID = questionID;
this.questionText = questionText;
this.optionA = optionA;
this.optionB = optionB;
this.optionC = optionC;
this.optionD = optionD;
this.correctAnswer = correctAnswer;
}
// Getters
public int getQuestionID() { return questionID; }
public String getQuestionText() { return questionText; }
public String getOptionA() { return optionA; }
public String getOptionB() { return optionB; }
public String getOptionC() { return optionC; }
public String getOptionD() { return optionD; }
public String getCorrectAnswer() { return correctAnswer; }
}
//SQL : TABLES FOR QUESTION AND INFORMATION STORAGE
drop table HardQuestions;
CREATE TABLE HardQuestions (
QuestionID NUMBER PRIMARY KEY,
QuestionText VARCHAR2(255) NOT NULL,
OptionA VARCHAR2(255) NOT NULL,
```

```
OptionB VARCHAR2(255) NOT NULL,
OptionC VARCHAR2(255) NOT NULL,
OptionD VARCHAR2(255) NOT NULL,
CorrectAnswer CHAR(1) NOT NULL
);
CREATE TABLE EasyQuestions (
QuestionID NUMBER PRIMARY KEY,
QuestionText VARCHAR2(255) NOT NULL,
OptionA VARCHAR2(255) NOT NULL,
OptionB VARCHAR2(255) NOT NULL,
OptionC VARCHAR2(255) NOT NULL,
OptionD VARCHAR2(255) NOT NULL,
CorrectAnswer CHAR(1) NOT NULL
);
INSERT INTO EasyQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(1, 'What is the capital of Italy?', 'Rome', 'Milan', 'Florence', 'Venice',
'A');
INSERT INTO EasyQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(2, 'What color do you get by mixing blue and yellow?', 'Green', 'Purple',
'Orange', 'Brown', 'A');
INSERT INTO EasyQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(3, 'Which animal is known for its ability to change color?', 'Chameleon',
'Dog', 'Elephant', 'Tiger', 'A');
INSERT INTO EasyQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(4, 'What is the name of the fairy in Peter Pan?', 'Tinker Bell',
'Cinderella', 'Snow White', 'Ariel', 'A');
INSERT INTO EasyQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(5, 'How many legs does a spider have?', '6', '8', '10', '12', 'B');
INSERT INTO EasyQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
```

(6, 'What fruit is known for having seeds on the outside?', 'Banana',
'Strawberry', 'Kiwi', 'Apple', 'B');
INSERT INTO EasyQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(7, 'Which planet is known as the Red Planet?', 'Mars', 'Venus', 'Mercury',
'Jupiter', 'A');
INSERT INTO EasyQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(8, 'What is the main ingredient in bread?', 'Rice', 'Flour', 'Sugar', 'Salt',
'B');
INSERT INTO EasyQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(9, 'Which ocean is the largest?', 'Atlantic Ocean', 'Indian Ocean', 'Arctic
Ocean', 'Pacific Ocean', 'D');
INSERT INTO EasyQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(10, 'What is the largest mammal in the world?', 'African Elephant', 'Blue
Whale', 'Giraffe', 'Great White Shark', 'B');
INSERT INTO EasyQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(11, 'In what year did the first man land on the moon?', '1965', '1969',
'1972', '1975', 'B');
INSERT INTO EasyQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(12, 'Which cartoon character is known for saying “What’s up, Doc?”', 'Daffy
Duck', 'Bugs Bunny', 'Mickey Mouse', 'Donald Duck', 'B');
INSERT INTO EasyQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(13, 'What is the smallest continent by land area?', 'Europe', 'Australia',
'Antarctica', 'South America', 'B');
INSERT INTO EasyQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(14, 'What do you call a baby kangaroo?', 'Calf', 'Cub', 'Joey', 'Kit', 'C');
INSERT INTO EasyQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES

```
(15, 'What is the boiling point of water in degrees Celsius?', '90', '100',
'120', '80', 'B');
SELECT * from EASYQUESTIONS;
INSERT INTO HardQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(1, 'What is the main function of the chloroplast in plant cells?', 'Cellular
respiration', 'Photosynthesis', 'Energy production', 'Protein synthesis',
'B');
INSERT INTO HardQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(2, 'Who is known as the father of modern physics?', 'Isaac Newton', 'Albert
Einstein', 'Niels Bohr', 'Galileo Galilei', 'B');
INSERT INTO HardQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(3, 'What is the chemical formula for table salt?', 'NaCl', 'KCl', 'CaCl2',
'MgCl2', 'A');
INSERT INTO HardQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(4, 'What type of bond involves the sharing of electron pairs between atoms?',
'Ionic bond', 'Covalent bond', 'Metallic bond', 'Hydrogen bond', 'B');
INSERT INTO HardQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(5, 'In which year did the Berlin Wall fall?', '1987', '1989', '1991', '1993',
'B');
INSERT INTO HardQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(6, 'What is the capital city of Japan?', 'Tokyo', 'Kyoto', 'Osaka',
'Hiroshima', 'A');
INSERT INTO HardQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(7, 'Which law states that for every action, there is an equal and opposite
reaction?', 'Newtons First Law', 'Newtons Second Law', 'Newtons Third Law',
'Law of Conservation of Energy', 'C');
INSERT INTO HardQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
```

```
(8, 'What is the process by which plants lose water vapor through small
openings called stomata?', 'Photosynthesis', 'Transpiration', 'Respiration',
'Evaporation', 'B');
INSERT INTO HardQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(9, 'What is the largest organ in the human body?', 'Liver', 'Heart', 'Skin',
'Brain', 'C');
INSERT INTO HardQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(10, 'What economic theory advocates for minimal government intervention in
the economy?', 'Keynesianism', 'Mercantilism', 'Laissez faire', 'Socialism',
'C');
INSERT INTO HardQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(11, 'Which element has the atomic number 1?', 'Helium', 'Oxygen', 'Hydrogen',
'Lithium', 'C');
INSERT INTO HardQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(12, 'What is the main language spoken in Brazil?', 'Spanish', 'English',
'Portuguese', 'French', 'C');
INSERT INTO HardQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(13, 'What term describes the genetic makeup of an organism?', 'Phenotype',
'Genotype', 'Allele', 'Chromosome', 'B');
INSERT INTO HardQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(14, 'What is the name of the largest desert in the world?', 'Sahara Desert',
'Arabian Desert', 'Gobi Desert', 'Antarctic Desert', 'D');
INSERT INTO HardQuestions (QuestionID, QuestionText, OptionA, OptionB,
OptionC, OptionD, CorrectAnswer) VALUES
(15, 'Which scientist is famous for his work on the theory of evolution?',
'Gregor Mendel', 'Charles Darwin', 'Louis Pasteur', 'Albert Einstein', 'B');
SELECT * from HARDQUESTIONS;
GRANT ALL PRIVILEGES ON SYS.HARDQUESTIONS TO SYSTEM;
GRANT ALL PRIVILEGES ON SYS.EASYQUESTIONS TO SYSTEM;
```

```
CREATE TABLE Quizzard (
PhoneNumber VARCHAR(15) PRIMARY KEY,
UserName VARCHAR(50) NOT NULL
QuizLevel VARCHAR(10) NOT NULL,
MarksObtained INT
);
GRANT ALL PRIVILEGES ON SYS.Quizzard TO SYSTEM;
CREATE TABLE Quizzar (
PhoneNumber VARCHAR(15) PRIMARY KEY,    Stores the user's phone number as
the primary key
UserName VARCHAR(50) NOT NULL,    Stores the user's name
QuizLevel VARCHAR(10) NOT NULL    Stores the quiz level, e.g., 'easy' or
'hard'
);
GRANT ALL PRIVILEGES ON SYS.Quizzar TO SYSTEM;
```

Conclusion

The Quiz Management System project, developed under experienced guidance, demonstrates a meticulous approach to both design and implementation. With user friendly functionalities such as adding/viewing quiz questions, managing difficulty levels, and tracking user performance, the system provides an engaging and efficient quiz taking experience. Robust security measures, particularly in the administrator login and question management modules, underscore a commitment to data integrity and system protection. This project serves as a comprehensive solution that meets current educational and training needs while providing a flexible foundation for future enhancements scalability.

References

1. **https://www.javatpoint.com/java awt**
2. **https://www.javatpoint.com/java swing**
3. **http://www.codeproject.com**
4. **http://www.udemy.com**
5. **http://www.support.microsoft.com**
6. **http://www.codeacademy.com**