QUIZ MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

AVISHNAVI P -231001024

GOKULA SRINITHI -231001048

in partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

RAJALAKSHMI ENGINEERING COLLAGE

NOVEMBER 2024

BONAFIDE CERTIFICATE

Certified that this project report titled **"QUIZ MANAGEMENT SYSTEM"** is the bonafide work of **AVISHNAVI P (231001024), GOKULA SRINITHI (231001048)** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Dr.P.Valarmathie	**Mrs.Usha S**
HEAD OF THE DEPARTMENT	**COURSE INCHARGE**
Department of Information Technology	Department of Information Technology
Rajalakshmi Engineering College	Rajalakshmi Engineering College

Submitted to Project Viva Voce Examination held On_____

Internal Examiner External Examiner

ABSTRACT

The Quiz Management System (QMS) is a web-based application designed to streamline the process of creating, administering, and evaluating quizzes. It enables instructors or administrators to easily create custom quizzes by defining multiple-choice questions, true/false statements, and other question formats. The system allows for the scheduling of quizzes, automatic grading, and real-time result tracking. Users (students) can attempt quizzes, receive instant feedback on their performance, and review detailed reports on their results. The system includes features such as user authentication, role-based access control, a database for storing quiz data, and reporting tools for both users and administrators. By automating quiz grading and result analysis, the QMS reduces the administrative burden on educators while enhancing the learning experience for students through interactive assessments. This system can be used in a variety of educational settings, from schools to online learning platforms, improving the efficiency and effectiveness of evaluation processes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

In traditional educational environments, the process of creating, administering, and grading quizzes is often time-consuming and prone to human error. Teachers and administrators must manually design quizzes, grade them, and provide feedback, which can be inefficient and cumbersome, especially when dealing with large groups of students. Additionally, manually tracking student progress and analyzing quiz results can be a challenging task that consumes significant time and resources.

Moreover, students may face difficulties in receiving immediate feedback on their performance, leading to delayed understanding of their strengths and weaknesses. In the case of online learning platforms, the lack of a centralized, automated quiz management system further complicates quiz distribution, grading, and result analysis.

Thus, there is a need for an automated and efficient **Quiz Management System (QMS)** that can simplify the process of quiz creation, scheduling, grading, and reporting. The system should provide a user-friendly interface for both instructors and students, automate the grading process, provide instant feedback, and allow for real-time tracking of student progress. Additionally, it should offer robust security features to ensure data privacy and integrity. By addressing these challenges, the system will improve the overall learning experience, reduce administrative workload, and enhance the quality of assessments.

1.2 OBJECTIVE OF THE PROJECT

The objective of the Quiz Management System is to create an automated platform that simplifies the creation, administration, grading, and analysis of quizzes. The system aims to reduce the manual workload for educators by providing an efficient way to design quizzes, automatically grade student responses, and generate real-time performance reports. It seeks to improve the learning experience by offering students instant feedback on their quiz results, allowing them to track their progress and areas for improvement. Additionally, the system will provide a secure and user-friendly environment for both instructors and students, ensuring ease of use, data privacy, and access control. Ultimately,

the goal is to enhance the educational process by automating time-consuming tasks and providing accurate, real-time assessments to both instructors and students.

1.3 ORGANIZATION OF THE PROJECT

CHAPTER 1: INTRODUCTION

CHAPTER 2: SYSTEMFLOW DIAGRAM

CHAPTER 3: SYSTEMDESIGN

CHAPTER 4: CONCLUSION

CHAPTER 2

2 SYSTEM FLOW DIAGRAM

2.1 USE CASE DIAGRAM

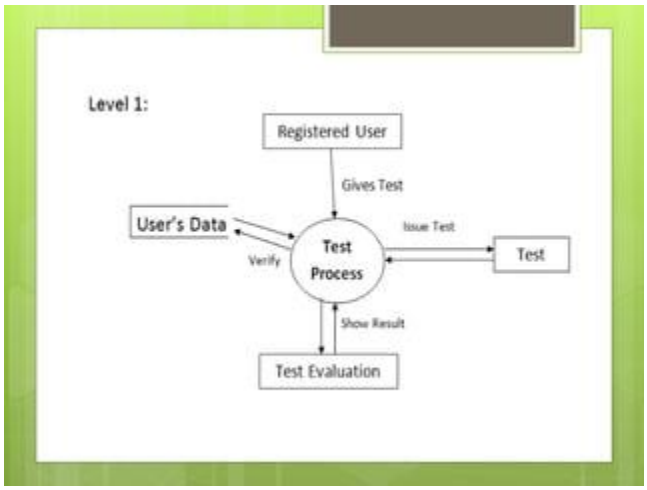


Figure 2.1 Use Case Diagram

2.2 ENTITY RELATIONSHIP DIAGRAM

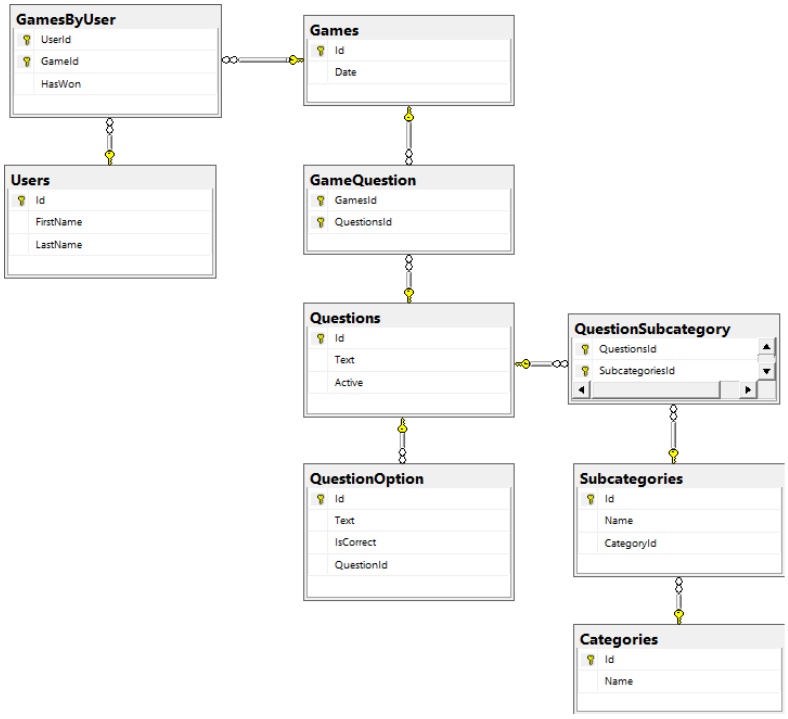


Figure 2.2 Entity Relationship Diagram

2.3 DATA FLOW DIAGRAM

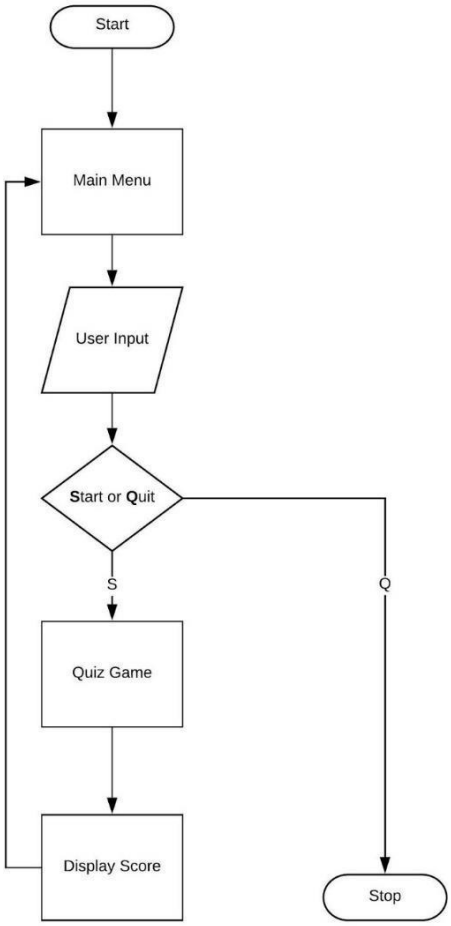


Figure 2.3 Data Flow Diagram

2.4 SYSTEM SPECIFICATION

The Quiz Management System will feature a web-based interface for quiz creation, scheduling, and grading. It will support multiple question types (e.g., multiple-choice, true/false), automatic grading, real-time performance tracking, and detailed reports. The system will provide secure login, role-based access, and ensure data privacy and system scalability.

2.4.1 SOFTWARE REQUIREMENTS

1. **Operating System**: Windows, macOS, or Linux.
2. **Web Server**: Apache Tomcat.
3. **Database**: MySQL, PostgreSQL, or Oracle.
4. **Programming Languages**: Java (with JDBC for database connection), HTML, CSS, JavaScript.
5. **Frameworks**: Java Servlets, JSP, JDBC.
6. **IDE**: Eclipse, IntelliJ IDEA, or NetBeans.
7. **Version Control**: Git.
8. **Security**: SSL/TLS, Prepared Statements (for SQL injection prevention), CAPTCHA.
9. **Testing Tools**: JUnit, Postman.

2.4.2 FUNCTIONAL REQUIREMENTS

1. User Authentication
2. Quiz Creation
3. Question Management
4. Quiz Scheduling
5. Student Quiz Attempt
6. Results
7. Reports Generation
8. Admin Management

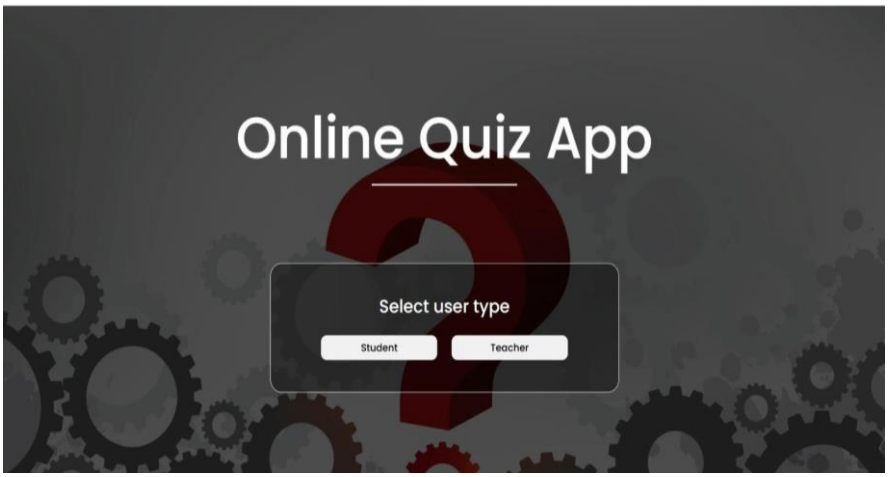
2.4.3 ADDITIONAL FEATURES

1. **Question Bank**:
   The system can maintain a repository of reusable questions that instructors can select for multiple quizzes.
2. **Timed Quizzes**:
   The system allows setting time limits for each quiz and provides countdown timers during student attempts.
3. **Randomized Questions and Answers**:
   Randomization ensures different question orders and shuffled answer choices to minimize cheating.
4. **Review and Retake Option**:
   Students can review their quiz attempts and retake quizzes if permitted by the instructor.

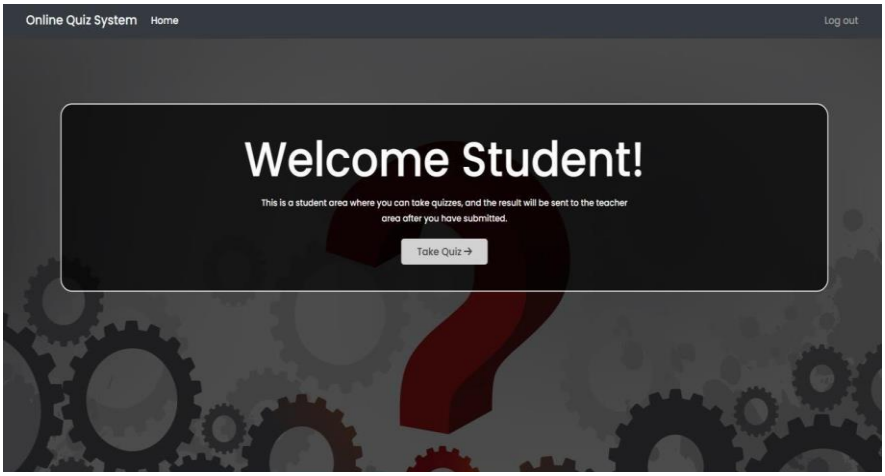
CHAPTER 3

SYSTEM DESIGN

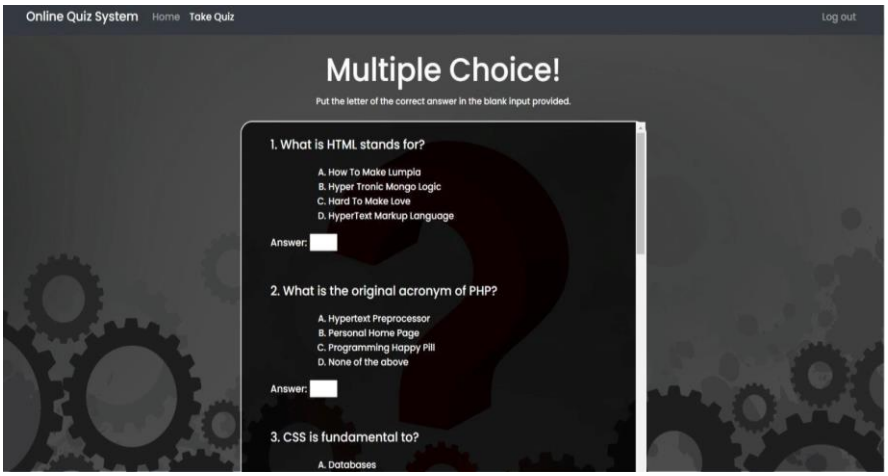
3.1 DESIGN



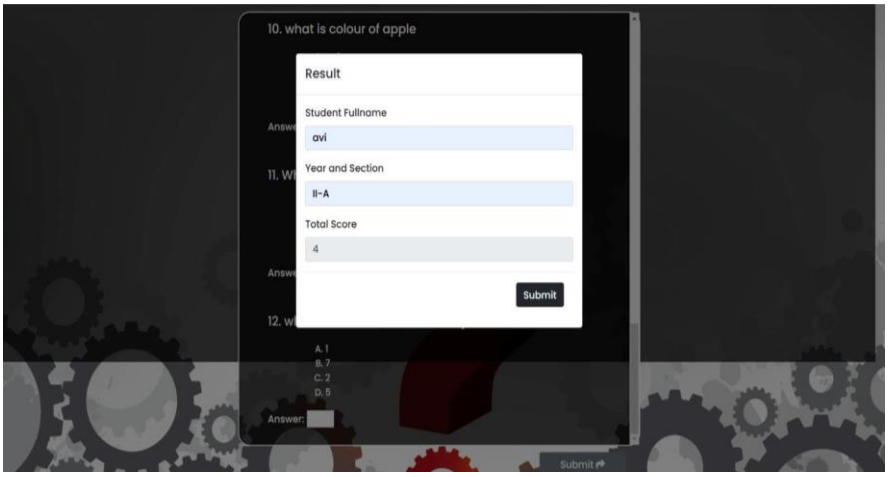
3.1.1 Home Page



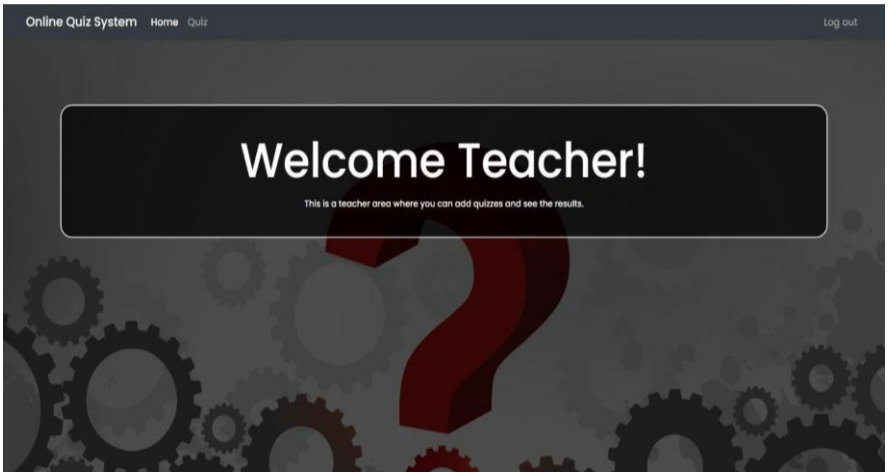
3.1.2 Welcome page for student



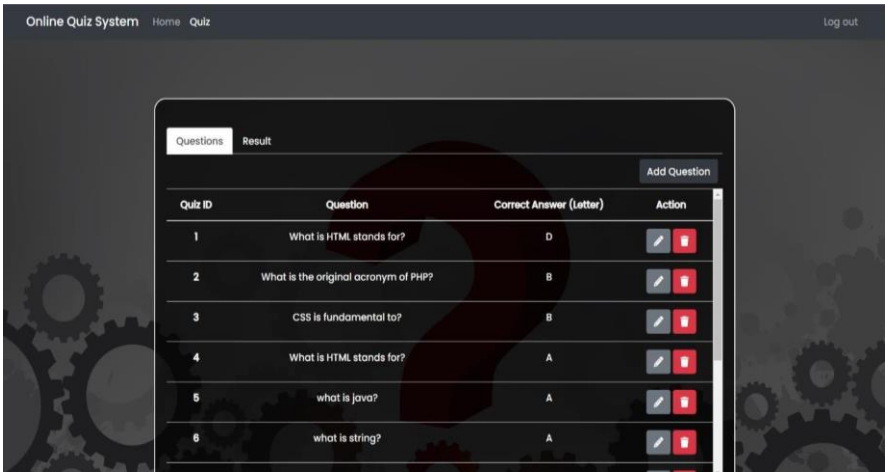
3.1.3 Question Page



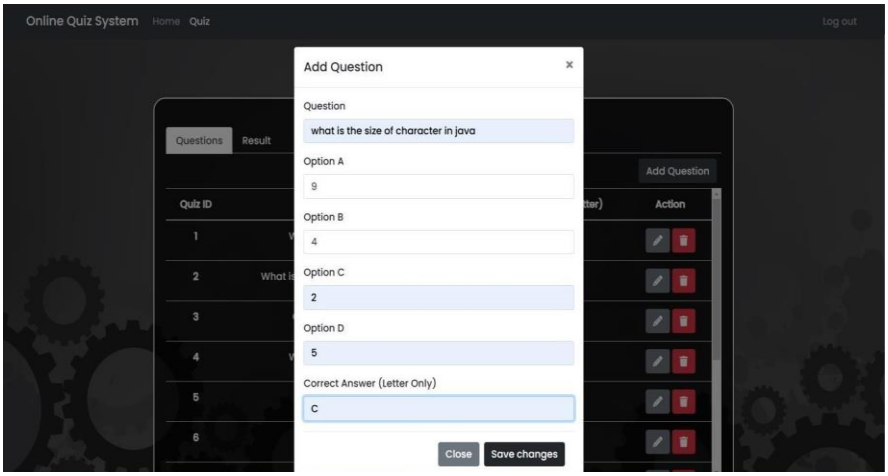
3.1.4 Result Page



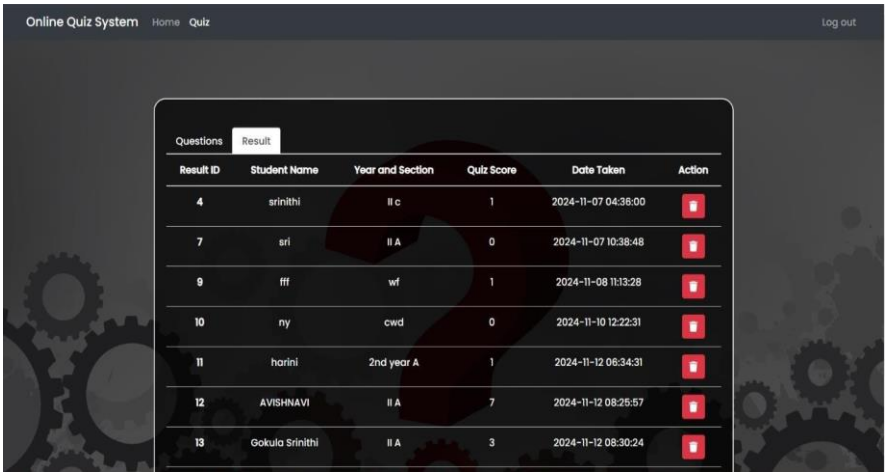
3.1.5 Welcome Page For The Teacher



3.1.6 Question Handler



3.1.7 Add Question



3.1.8 Result View By Teacher

3.2 DATABASE DESIGN

	tbl_quiz_id	quiz_question	option_a	option_b	option_c	option_d	correct_answer
Edit Copy Delete	1	What is HTML stands for?	How To Make Lumpia	Hyper Tronic Mongo Logic	Hard To Make Love	HyperText Markup Language	D
Edit Copy Delete	2	What is the original acronym of PHP?	Hypertext Preprocessor	Personal Home Page	Programming Happy Pill	None of the above	B
Edit Copy Delete	3	CSS is fundamental to?	Databases	Web design	Server-side	None of the above	B
Edit Copy Delete	4	What is HTML stands for?	hyper text markup language	Hyper Tronic Mongo Logic	Hard To Make Love	HyperText Markup Language	A
Edit Copy Delete	5	what is java?	prg language	mangodb	apple	banana	A
Edit Copy Delete	6	what is string?	sequence of characters	sequence of numbers	sequence of special chaarcters	mutable	A
Edit Copy Delete	7	WHAT IS COLOUR OF BANANA	yellow	blue	orange	purple	A
Edit Copy Delete	9	what is array	it stores same data type values	it stores same variable type	it stores different data type value	it stores different variable type value	A
Edit Copy Delete	10	what is colour of apple	red	green	blue	yellow	A
Edit Copy Delete	11	What is IT	Information Technology	Info Tech	Internet things	IMO	A
Edit Copy Delete	12	what is the size of character in java	1	7	2	5	C

3.2.1 Question Table

	tbl_result_id	quiz_taker	year_section	total_score	date_taken
Edit Copy Delete	4	srinithi	II c	1	2024-11-07 04:36:00
Edit Copy Delete	7	sri	II A	0	2024-11-07 10:38:48
Edit Copy Delete	9	fff	wf	1	2024-11-08 11:13:28
Edit Copy Delete	10	ny	cwd	0	2024-11-10 12:22:31
Edit Copy Delete	11	harini	2nd year A	1	2024-11-12 06:34:31
Edit Copy Delete	12	AVISHNAVI	II A	7	2024-11-12 08:25:57
Edit Copy Delete	13	Gokula Srinithi	II A	3	2024-11-12 08:30:24
Edit Copy Delete	14			1	2024-11-12 08:59:59
Edit Copy Delete	15			0	2024-11-19 09:41:08

3.2.2 Result Table

CHAPTER 4

IMPLEMENTATION

4.1 Login Page

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;


public class LoginPage {


   public static void main(String[] args) {

      // Create the main frame

      JFrame frame = new JFrame("Online Quiz App");

      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      frame.setSize(600, 400);

      frame.setLayout(new BorderLayout());


      // Main container panel

      JPanel mainPanel = new JPanel();

      mainPanel.setLayout(new BorderLayout());


      // Title of the app

      JLabel titleLabel = new JLabel("Online Quiz App", JLabel.CENTER);

      titleLabel.setFont(new Font("Arial", Font.BOLD, 30));

      mainPanel.add(titleLabel, BorderLayout.NORTH);


      // Border line (using a JSeparator)
```

```
JSeparator separator = new JSeparator();
separator.setPreferredSize(new Dimension(600, 2));
mainPanel.add(separator, BorderLayout.CENTER);

// Panel for user selection
JPanel selectionPanel = new JPanel();
selectionPanel.setLayout(new BoxLayout(selectionPanel, BoxLayout.Y_AXIS));
selectionPanel.setAlignmentX(Component.CENTER_ALIGNMENT);

// Heading
JLabel selectLabel = new JLabel("Select user type");
selectLabel.setFont(new Font("Arial", Font.PLAIN, 20));
selectionPanel.add(selectLabel);
selectionPanel.add(Box.createVerticalStrut(20)); // Spacer

// Button Panel for Student and Teacher
JPanel buttonPanel = new JPanel();
buttonPanel.setLayout(new FlowLayout());

JButton studentButton = new JButton("Student");
studentButton.setPreferredSize(new Dimension(150, 40));
studentButton.setBackground(Color.BLUE);
studentButton.setForeground(Color.WHITE);
studentButton.addActionListener(new ActionListener() {
   @Override
   public void actionPerformed(ActionEvent e) {
      // Open the Student page when Student button is clicked
      frame.setVisible(false); // Hide the LoginPage
```

```
        new Student(); // Open the Student quiz page
    }
});

JButton teacherButton = new JButton("Teacher");
teacherButton.setPreferredSize(new Dimension(150, 40));
teacherButton.setBackground(Color.GREEN);
teacherButton.setForeground(Color.WHITE);
teacherButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // Open TeacherPage when Teacher button is clicked
        frame.setVisible(false); // Hide the LoginPage
        new TeacherPage(); // Open TeacherPage
    }
});

buttonPanel.add(studentButton);
buttonPanel.add(teacherButton);

// Add the button panel to the selection panel
selectionPanel.add(buttonPanel);

// Add the selection panel to the main panel
mainPanel.add(selectionPanel, BorderLayout.CENTER);

// Add the main panel to the frame
frame.add(mainPanel);
```

```
        // Display the frame
        frame.setVisible(true);
    }
}
```

4.2 For Student Page

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class StudentPage extends JFrame {

    private JButton btnViewQuizzes, btnStartQuiz, btnViewResults, btnLogout;
    private JList<String> quizList;
    private DefaultListModel<String> quizListModel;
    private JPanel panel;

    public StudentPage() {
        // Set up the frame
        setTitle("Student Dashboard");
        setSize(400, 400);
        setLocationRelativeTo(null); // Center the window
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Initialize components
        panel = new JPanel();
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));
```

```
quizListModel = new DefaultListModel<>();
quizList = new JList<>(quizListModel);
quizList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
quizList.setBorder(BorderFactory.createTitledBorder("Available Quizzes"));

btnViewQuizzes = new JButton("View Quizzes");
btnStartQuiz = new JButton("Start Quiz");
btnViewResults = new JButton("View Results");
btnLogout = new JButton("Logout");

// Add sample quizzes to the list
quizListModel.addElement("Quiz 1: Java Basics");
quizListModel.addElement("Quiz 2: OOP Concepts");
quizListModel.addElement("Quiz 3: Data Structures");

// Set up the layout and add components to the panel
panel.add(new JScrollPane(quizList));
panel.add(btnViewQuizzes);
panel.add(btnStartQuiz);
panel.add(btnViewResults);
panel.add(btnLogout);

// Add the panel to the frame
add(panel);

// Button actions
btnViewQuizzes.addActionListener(new ActionListener() {
```

```
    @Override
    public void actionPerformed(ActionEvent e) {
        // Display available quizzes (for now, this is handled by the JList)
        JOptionPane.showMessageDialog(null, "Quizzes displayed here.");
    }
});

btnStartQuiz.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String selectedQuiz = quizList.getSelectedValue();
        if (selectedQuiz != null) {
            // In real application, launch the quiz interface
            JOptionPane.showMessageDialog(null, "Starting: " + selectedQuiz);
        } else {
            JOptionPane.showMessageDialog(null, "Please select a quiz to start.");
        }
    }
});

btnViewResults.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // In real application, fetch and display student results
        JOptionPane.showMessageDialog(null, "Displaying results.");
    }
});
```

```
        btnLogout.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // Log out the student (close the application)
                System.exit(0);
            }
        });
    }

    public static void main(String[] args) {
        // Launch the student page GUI
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new StudentPage().setVisible(true);
            }
        });
    }
}
```

4.3 Teacher Page

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class TeacherPage extends JFrame {

    private JButton btnCreateQuiz, btnViewQuizzes, btnViewResults, btnLogout;
```

```
private JList<String> quizList;
private DefaultListModel<String> quizListModel;
private JPanel panel;

public TeacherPage() {
    // Set up the frame
    setTitle("Teacher Dashboard");
    setSize(400, 400);
    setLocationRelativeTo(null); // Center the window
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    // Initialize components
    panel = new JPanel();
    panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));

    quizListModel = new DefaultListModel<>();
    quizList = new JList<>(quizListModel);
    quizList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
    quizList.setBorder(BorderFactory.createTitledBorder("Created Quizzes"));

    btnCreateQuiz = new JButton("Create New Quiz");
    btnViewQuizzes = new JButton("View Quizzes");
    btnViewResults = new JButton("View Quiz Results");
    btnLogout = new JButton("Logout");

    // Add sample quizzes to the list (in a real app, this would be fetched from a database)
    quizListModel.addElement("Quiz 1: Java Basics");
    quizListModel.addElement("Quiz 2: Object-Oriented Programming");
```

```
quizListModel.addElement("Quiz 3: Data Structures");

// Set up the layout and add components to the panel
panel.add(new JScrollPane(quizList));
panel.add(btnCreateQuiz);
panel.add(btnViewQuizzes);
panel.add(btnViewResults);
panel.add(btnLogout);

// Add the panel to the frame
add(panel);

// Button actions
btnCreateQuiz.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // Launch quiz creation form (this is a placeholder for actual quiz creation logic)
        JOptionPane.showMessageDialog(null, "Opening quiz creation form...");
    }
});

btnViewQuizzes.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // Display list of created quizzes (this would be dynamically fetched from a database)
        JOptionPane.showMessageDialog(null, "Quizzes listed here.");
    }
});
```

```
    btnViewResults.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // View results for the selected quiz
            String selectedQuiz = quizList.getSelectedValue();
            if (selectedQuiz != null) {
                // In a real application, fetch and display quiz results
                JOptionPane.showMessageDialog(null, "Viewing results for: " + selectedQuiz);
            } else {
                JOptionPane.showMessageDialog(null, "Please select a quiz to view results.");
            }
        }
    });

    btnLogout.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // Log out the teacher (close the application)
            System.exit(0);
        }
    });
}

public static void main(String[] args) {
    // Launch the teacher page GUI
    SwingUtilities.invokeLater(new Runnable() {
        @Override
```

```
        public void run() {
            new TeacherPage().setVisible(true);
        }
    });
}
```

CONCLUSION

The Quiz Management System provides an efficient and automated platform for creating, administering, and evaluating quizzes in educational environments. It streamlines the process of quiz creation, grading, and result analysis, reducing manual effort and errors. By offering real-time feedback, performance tracking, and secure data management, it enhances both the learning experience for students and the administrative workflow for educators, making the assessment process more effective, scalable, and user-friendly.

REFERENCE

1. https://www.freecodecamp.org/news/how-to-build-an-online-quiz-system-python-oop-tutorial/

2. https://www.geeksforgeeks.org/courses/Java-backend-live

3. https://www.javatpoint.com/java-awt

4. https://www.javatpoint.com/java-swing