Manipal Institute of Technology

FISAC (mini project): OOP-II [MCA 4221]

Department of Data Science & Computer Applications

MCA-II Semester

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Institution of Manipal University

Name: Nishan Hegde

Reg. No: 230970139

Section: C

Problem Statement and Overview:

The educational institution wants to simplify its admission entrance test process by automating the creation of Multiple-Choice Questions (MCQs) question papers and store all score to database. To achieve this, they require a Java GUI application.

The application have three main components:

Login Window:

- Users will log in by entering their username and password.
- After successful login, the application will move to the MCQ page.

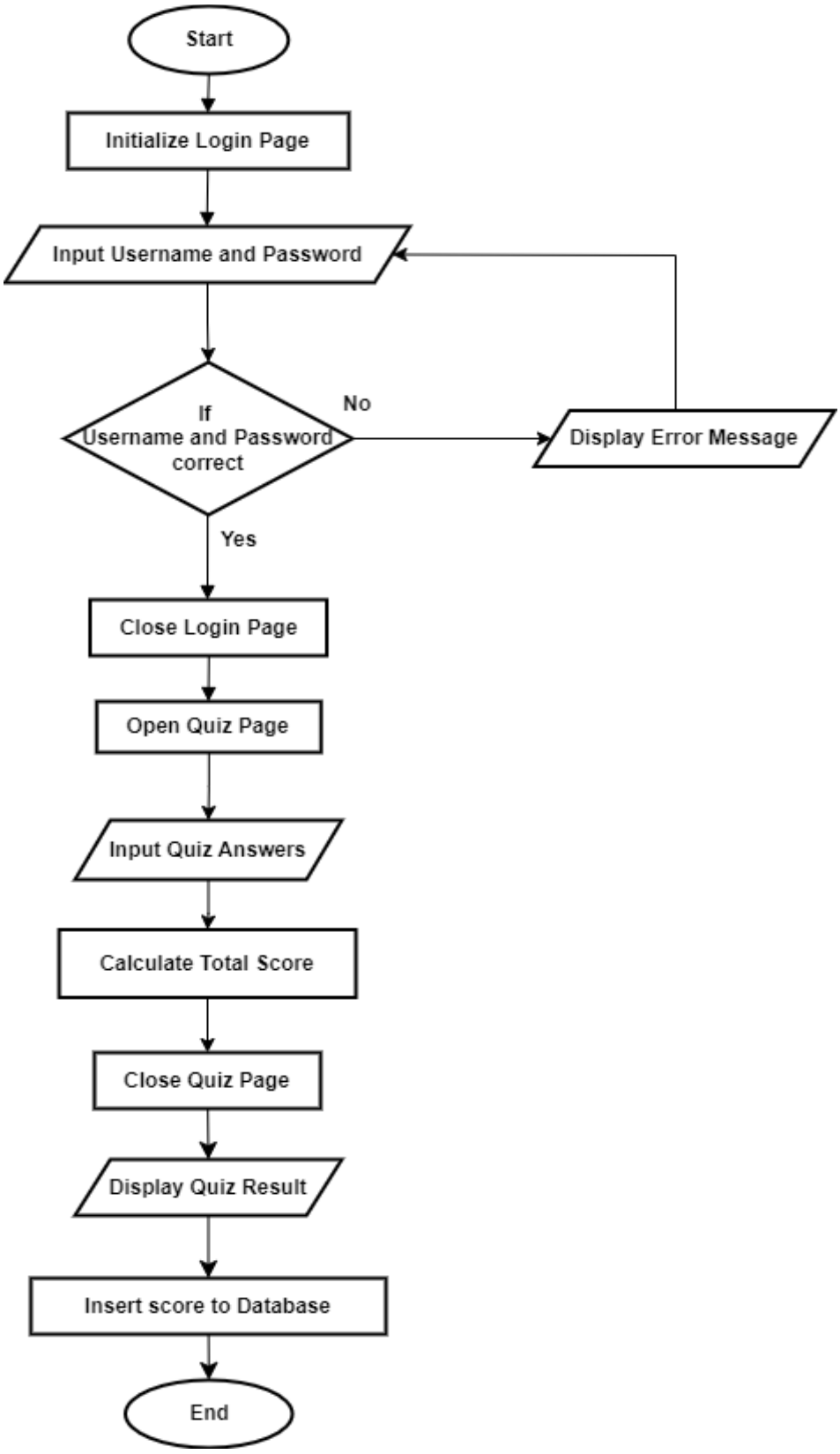
MCQ Page:

- This page will display 5 MCQs, each with a question and multiple radio buttons for selecting options.
- Users can select only one option per question.
- A "Submit" button will be provided for users to submit their answers.

Result Display:

- After the user has answered all the questions and submitted them, the application will calculate and display the total score.
- The score will be stored in the database for each student.

Flow diagram:



Swings Components used:

JFrame:
Main window of the application where all Swing components are added.

JLabel:
Displays text or images. In this application, it shows labels like "Username", "Password", etc.

JTextField:
Accepts single-line text input from the user, used for the username.

JButton:
Creates a button that triggers an action when clicked. Here, the "Submit" button is used to submit login credentials.

JPanel:
Lightweight container to hold other components. In this application, it holds the quiz questions and options.

JRadioButton:
Creates radio buttons for single selection. They represent options for each question.

ButtonGroup:
Groups together a set of radio buttons, allowing only one selection at a time for each question.

JScrollPane:
Provides a scrollable view of a component when its size exceeds the visible area. Enables scrolling for the quiz questions and options panel.

JOptionPane:
Displays standard dialog boxes for messages or input. In this application, it shows an error message for invalid credential.

Events, action:

ActionListener is an interface used to manage action events. It consists of a single method, actionPerformed(ActionEvent e), which is called when an action occurs.

In the Login class, the actionPerformed(ActionEvent e) method is used to handle the action event generated when the user clicks the "Submit" button. It retrieves the username and password entered by the user, validates them against the database, and opens the quiz frame if the credentials are correct.

In the QuizFrame class, the actionPerformed(ActionEvent e) method is used to handle the action event generated when the user clicks the "Submit" button. It calculates the total score based on the user's selected answers, updates the database with the score, and presents the quiz result in a new frame.

Program code with comments:

```
//Name: Nishan Hegde
//Reg. No: 230970139
//Section: C
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.*;
import javax.swing.border.EmptyBorder;

// Login class
public class Login implements ActionListener {

    JFrame frame;
    JTextField nameField, passwordField;
    JLabel nameLabel, passwordLabel, titleLabel;
    JButton submitButton;
```

```
    Connection connection;

    // Constructor
    Login() {
        // Creating and setting up the login frame
        frame = new JFrame("Login Page");
        frame.setSize(1000, 500);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(null);

        // Creating GUI components
        titleLabel = new JLabel("Candidate Page");
        titleLabel.setFont(new Font("Arial", Font.BOLD, 28));
        nameField = new JTextField();
        passwordField = new JPasswordField(); // Changed to
JPasswordField for password input
        nameLabel = new JLabel("Username");
        passwordLabel = new JLabel("Password");
        submitButton = new JButton("Submit");
        submitButton.addActionListener(this);

        // Setting the position and size of GUI components
        titleLabel.setBounds(400, 50, 300, 30);
        nameLabel.setBounds(400, 100, 100, 20);
        nameField.setBounds(500, 100, 150, 20);
        passwordLabel.setBounds(400, 150, 100, 20);
        passwordField.setBounds(500, 150, 150, 20);
        submitButton.setBounds(530, 200, 95, 30);

        // Adding GUI components to the frame
        frame.add(titleLabel);
        frame.add(nameLabel);
        frame.add(nameField);
        frame.add(passwordLabel);
        frame.add(passwordField);
        frame.add(submitButton);
        frame.setVisible(true);

        // Database connection initialization
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            connection = DriverManager.getConnection(
                    "jdbc:mysql://localhost:3306/jdbcdemo", "root", ""
            );
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
    }

    // ActionListener implementation
    public void actionPerformed(ActionEvent e) {
        String name = nameField.getText();
        String password = passwordField.getText();

        try {
            // Query to check if the username and password exist in the
```

```
database
            Statement statement = connection.createStatement();
            ResultSet resultSet = statement.executeQuery("SELECT * FROM
student WHERE Name='" + name + "' AND Password='" + password + "'");

            if (resultSet.next()) {
                System.out.println("Login Successful");
                // Close the login window after successful login
                frame.dispose();
                // Open the quiz window
                SwingUtilities.invokeLater(new Runnable() {
                    public void run() {
                        new QuizFrame(connection, name); // Passing the
name to quiz frame
                    }
                });
            } else {
                System.out.println("Invalid Username or Password");
                JOptionPane.showMessageDialog(frame, "Invalid Username or
Password", "Error", JOptionPane.ERROR_MESSAGE);
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    // Main method
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new Login();
            }
        });
    }
}

// QuizFrame class
class QuizFrame implements ActionListener {

    JFrame quizFrame;
    JPanel panel;
    JLabel lblTitle;
    JLabel[] lblQuestions;
    JRadioButton[][] radioButtons;
    ButtonGroup[] buttonGroups;
    JButton btnSubmit;
    Connection connection;
    String name;

    // Constructor
    QuizFrame(Connection connection, String name) {
        this.connection = connection;
        this.name = name;

        // Creating and setting up the quiz frame
        quizFrame = new JFrame("Quiz");
```

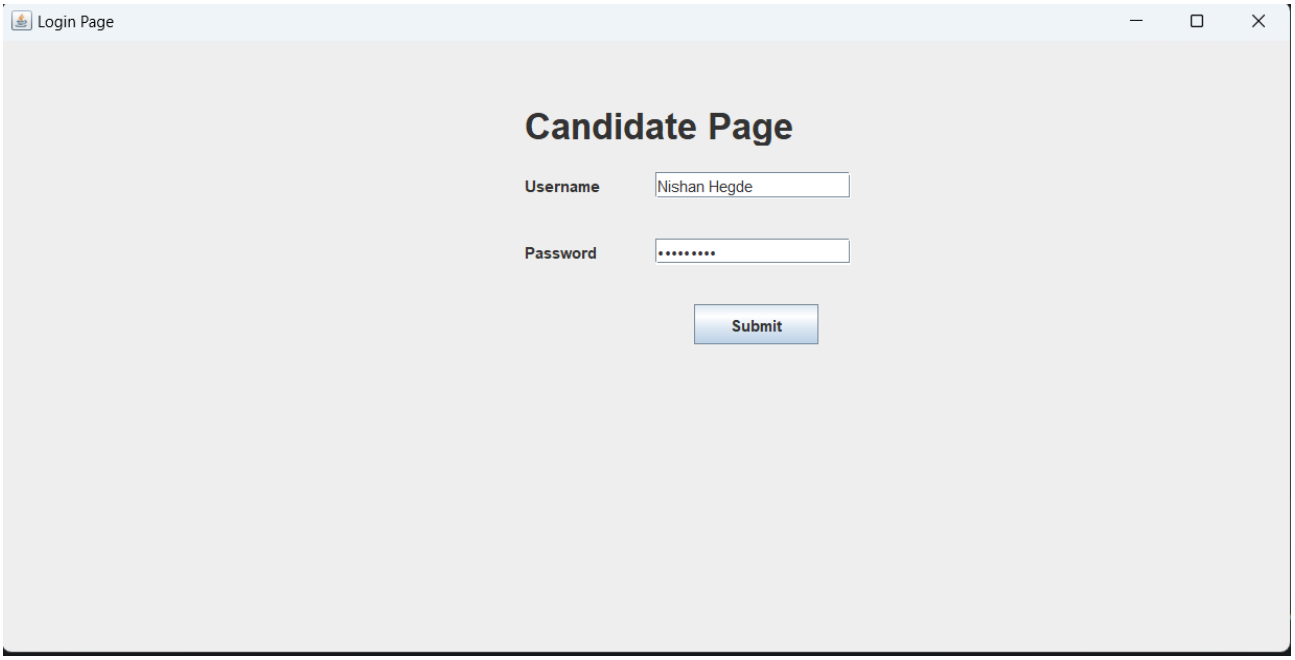
```
        quizFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        quizFrame.setSize(1000, 700); // Set the size of the quiz frame
        quizFrame.setLocationRelativeTo(null);
        quizFrame.setLayout(null);

        // Title label
        lblTitle = new JLabel("Quiz");
        lblTitle.setBounds(350, 10, 300, 30); // Adjusted the position
and size of the title
        lblTitle.setFont(new Font("Arial", Font.BOLD, 28));
        lblTitle.setOpaque(true);
        lblTitle.setBorder(new EmptyBorder(0, 10, 0, 0));

        // Panel to hold questions and options
        panel = new JPanel();
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));

        // Array to hold question texts and options
        String[] questionTexts = {
                "Which of the following is NOT a valid declaration of the
main method in Java?",
                "Which keyword is used to define a constant in Java?",
                "Which of the following data types is used to store a
single character in Java?",
                "What does the StringBuilder class in Java provide?",
                "What is the purpose of the break statement in Java?"
        };
        String[][] options = {
                {"a) public static void main(String[] args)", "b) public
static void main(String args[])", "c) public void main(String[] args)",
"d) static public void main(String[] args)"},
                {"a) constant", "b) var", "c) final", "d) static"},
                {"a) char" , "b) string", "c) Character", "d)
CharSequence"},
                {"a) Mutable sequence of characters", "b) Immutable
sequence of characters", "c) Synchronized sequence of characters", "d)
Sequential sequence of characters"},
                {"a) To terminate the loop or switch statement and
transfer control to the next statement.", "b) To skip the execution of
the current iteration and move to the next iteration in a loop.," ,"c) To
throw an exception and exit the program.,," ,"d) To define a named block
of code that can be exited with a goto statement."}
        };

        // Creating and adding question labels and radio buttons
        lblQuestions = new JLabel[5];
        radioButtons = new JRadioButton[5][4];
        buttonGroups = new ButtonGroup[5];

        for (int i = 0; i < 5; i++) {
            lblQuestions[i] = new JLabel((i + 1) + ") " +
questionTexts[i]);
            lblQuestions[i].setFont(new Font("Arial", Font.BOLD, 20));
            lblQuestions[i].setBorder(new EmptyBorder(10, 10, 0, 0));
            panel.add(lblQuestions[i]);
```

```
            buttonGroups[i] = new ButtonGroup();

            for (int j = 0; j < 4; j++) {
                radioButtons[i][j] = new JRadioButton(options[i][j]);
                buttonGroups[i].add(radioButtons[i][j]);
                panel.add(radioButtons[i][j]);
            }
        }

        // Submit button
        btnSubmit = new JButton("SUBMIT");
        btnSubmit.addActionListener(this);
        panel.add(btnSubmit);

        // Scroll pane to scroll through the questions and options
        JScrollPane scrollPane = new JScrollPane(panel);
        scrollPane.setBounds(50, 50, 900, 500); // Adjusted the position
and size of the scroll pane
        quizFrame.add(scrollPane);

        // Adding title label to the quiz frame
        quizFrame.add(lblTitle);
        quizFrame.setVisible(true);
    }

    // ActionListener implementation
    @Override
    public void actionPerformed(ActionEvent e) {
        int totalMarks = 0;
        String[] correctAnswers = {
                "c) public void main(String[] args)",
                "a) constant",
                "a) char",
                "a) Mutable sequence of characters",
                "a) To terminate the loop or switch statement and
transfer control to the next statement."
        };

        // Calculating total marks
        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 4; j++) {
                if (radioButtons[i][j].isSelected() &&
radioButtons[i][j].getText().equals(correctAnswers[i])) {
                    totalMarks++;
                    break;
                }
            }
        }

        // Closing the quiz frame
        quizFrame.dispose();

        try {
            // Insert marks in the database
            Statement statement = connection.createStatement();
            statement.executeUpdate("UPDATE student SET Mark=" +
```

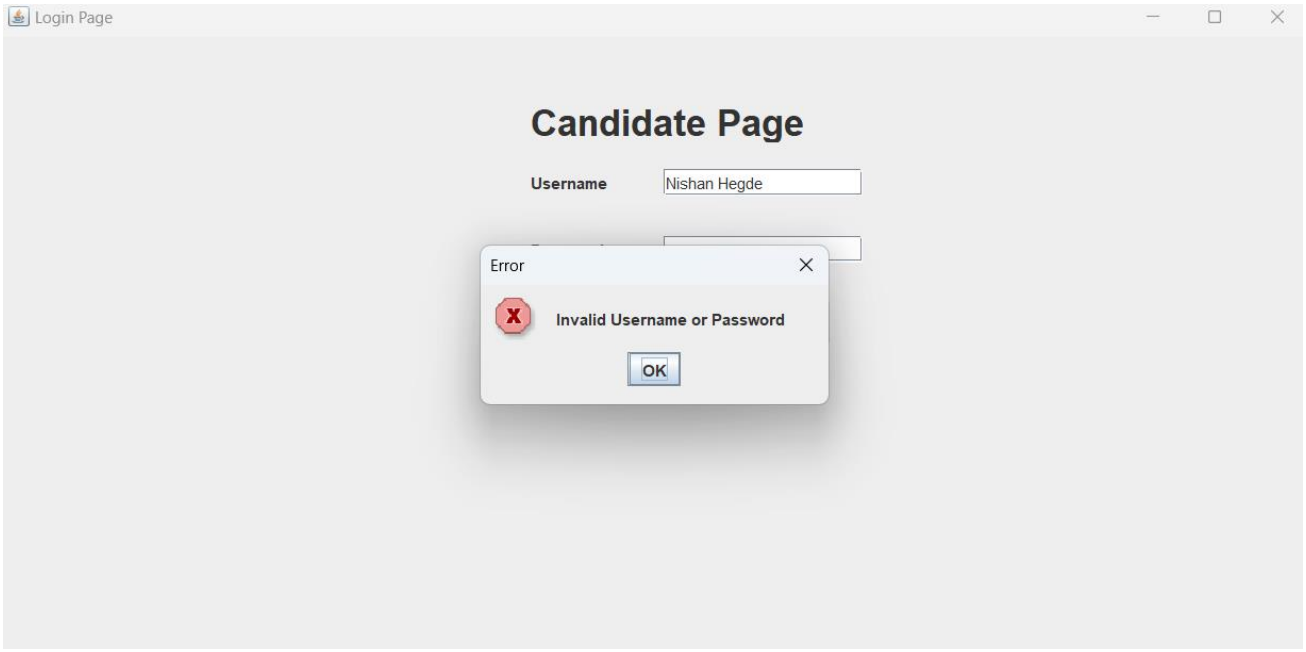
```
totalMarks + " WHERE Name='" + name + "'");
            statement.close();
        } catch (SQLException ex) {
            ex.printStackTrace();
        }

        // Creating and setting up the result frame
        JFrame resultFrame = new JFrame("Quiz Result");
        resultFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        resultFrame.setSize(300, 200);
        resultFrame.setLocationRelativeTo(null);

        // Displaying total score
        JLabel resultLabel = new JLabel("Total Score: " + totalMarks);
        resultLabel.setFont(new Font("Arial", Font.BOLD, 20));
        resultLabel.setHorizontalAlignment(SwingConstants.CENTER);
        resultFrame.add(resultLabel);

        resultFrame.setVisible(true);
    }
}
```

Screenshot of output:

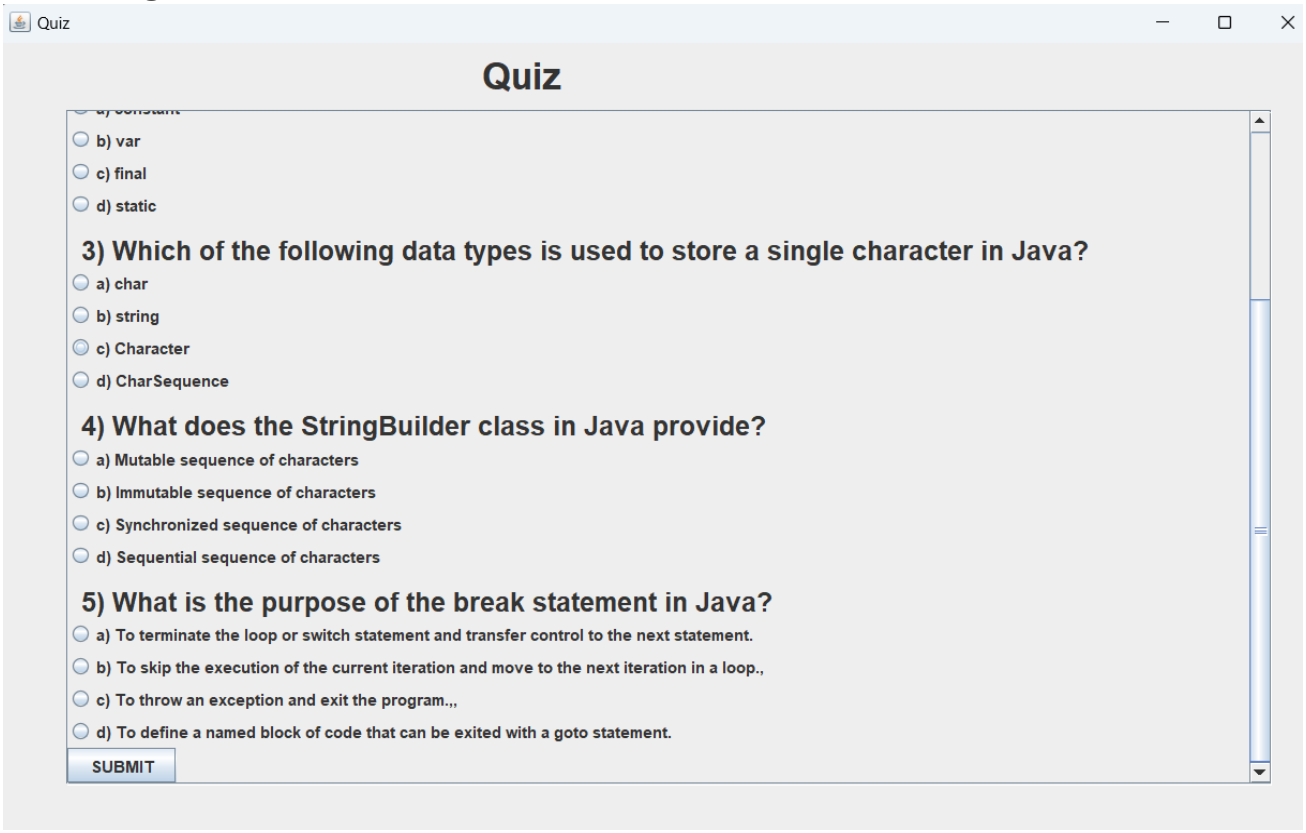
Login Page



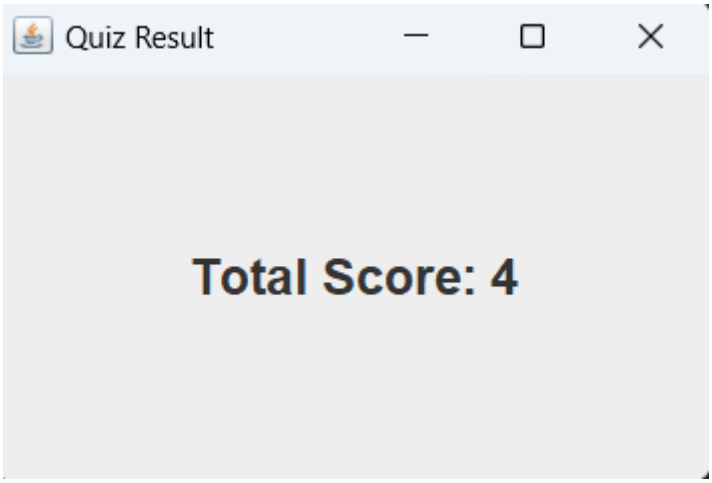
When login credential is wrong



Quiz Page



Score page



Database

Table Columns

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	Name	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
☐	2	Password	varchar(15)	utf8mb4_general_ci		No	None			Change Drop More
☐	3	Mark	int(10)			No	None			Change Drop More

Records

Name	Password	Mark 1
Jay	230970139	5
Adhi	Adhi@123	5
Yathish	yathish@123	5
Nishan Hegde	abcd@123	4
Rohan	Rohan@123	3
Surya	123456789	0
Sujay	Sujay2123	0

References:

Swing Form Creation
Textbook: Java The Complete Reference, Thirteenth Edition
https://www.javatpoint.com/java-swing
https://youtu.be/Hiv3gwJC5kw?si=FR9aD9GcX3ZOybc6
https://youtu.be/6K9OpqDM_RM?si=6--hdZZdq6OiUnNK

XAMPP database connection:
https://youtu.be/AHFBPxWebFQ?si=mfRC39v2hWUKhfJd
https://youtu.be/7v2OnUti2eM?si=I58rKfGO0U09gVqW