**JAVA FISAC ASSIGNMENT - 1**





**Name:- Ashwini Giri**

**Reg No.:- 230970096**

**Section:- MCA (B)**

**Date:- 20-04-24**

**Submitted to:- Raghuram Holla Sir**

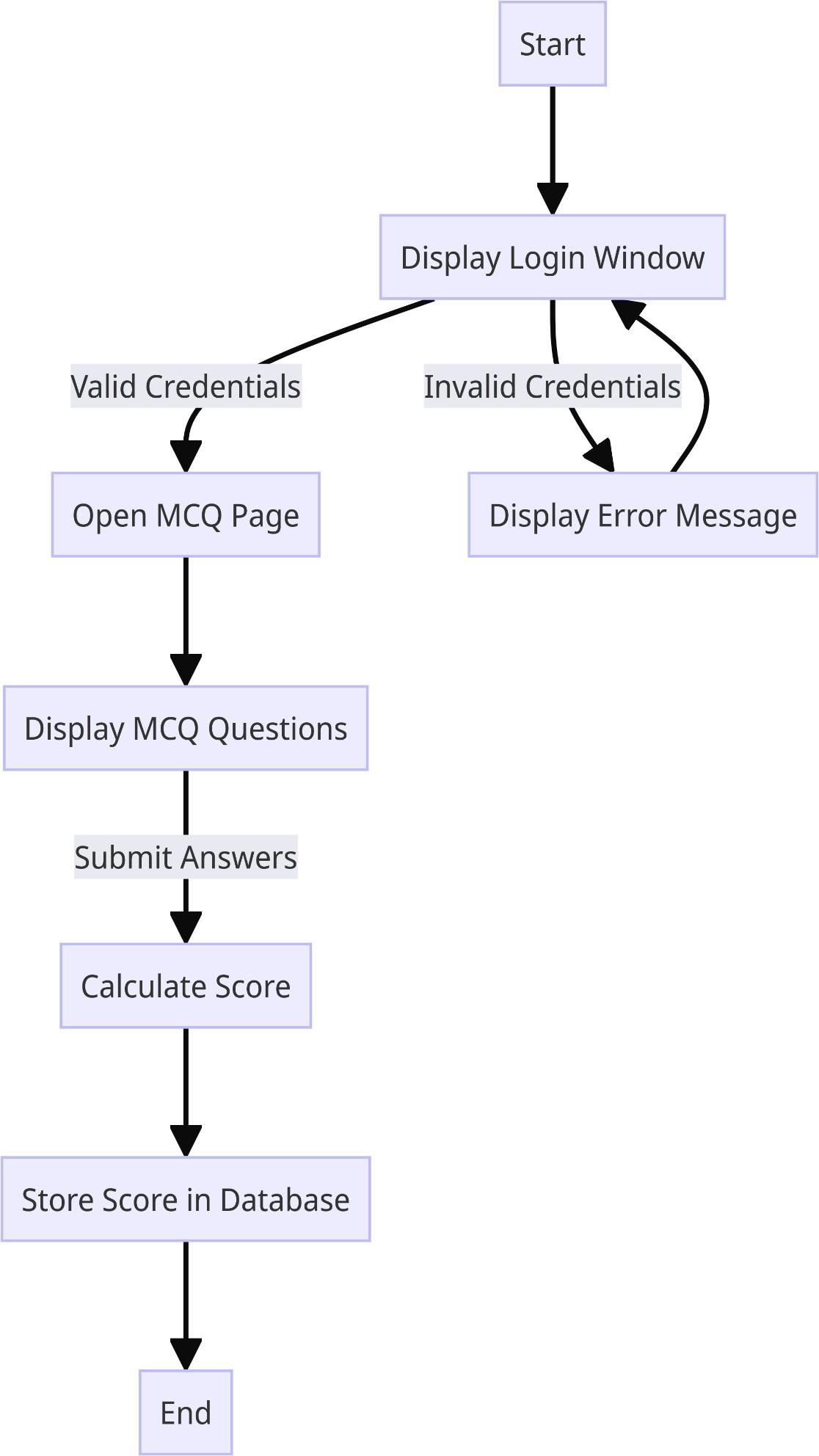
**Overview:**

Constructed a Java-based graphical user interface (GUI) application for administering an admission entrance exam comprising multiple-choice questions (MCQs), ensuring all specified requirements were met.

1. **Login Window:** 
   * The application opens a login window for user authentication.
   * Users need to enter their credentials (username and password) for validation.
   * After successful login, the application navigates to a new page.
   * This new page contains a set of multiple-choice questions (MCǪs).

1. **MCǪ Page:** 
   * Displaying 5 MCǪs on this page, each with a question and multiple radio buttons for choosing options.
   * Users can select one option per question.
   * Provided a "Submit" button for users to submit their answers.

1. **Result Display:** 
   * After the user has attempted all the questions and submitted their answers, the application calculates and displays the total score and stores the same in the database for each student.

**Flow-Chart :** 

**Swing components used :**

* + **JFrame:** Represents the main frame of the application.
  + **JLabel:** Displays text labels such as "Username:", "Password:", and question labels.
  + **JTextField:** Allows users to input text (e.g., username).
  + **JPasswordField:** Allows users to input passwords securely.
  + **JButton:** Represents buttons like "Login" and "Submit".
  + **JPanel:** Provides a container for organizing and grouping components.
  + **BoxLayout:** Used to arrange components in a vertical layout (MCǪ options).
  + **ButtonGroup:** Groups radio buttons to ensure only one option is selected per question.
  + **JRadioButton:** Represents radio buttons for multiple-choice questions.
  + **JScrollPane:** Provides scrolling functionality for the MCǪ panel.

**Events and Actions in the program:**

**Login Button Action:**

* + **loginButton.addActionListener()**: When the "Login" button is clicked.
  + **usernameField.getText()** and **new String(passwordField.getPassword())**: It retrieves the username and password entered by the user
  + **username.equals("Lalit") && password.equals("4321"):** Check if the credentials are valid **openMCǪPage():** It opens the MCǪ page
  + **JOptionPane.showMessageDialog**(): If the credentials are invalid, it shows an error message using.

**MCǪ Submit Button Action:**

* + **submitButton.addActionListener():** When the "Submit" button on the MCǪ page is clicked.
  + **calculateScore(mcqPanel):** It calculates the user's score by calling based on the selected answers.
  + **usernameField.getText():** It retrieves the username.
  + **JOptionPane.showMessageDialog()** and **storeScoreInDatabase(username, score)**: It shows the score using and stores the score in the database.

**Radio Button Selection:**

* + **JRadioButton** and **ButtonGroup**: Radio buttons for MCǪ options to ensure only one option is selected per question.
  + **isCorrectAnswer(radioButton.getText())**: The selected answer is checked when calculating the score.

**Database Interaction:**

**Connection, Prepared Statement,** and **Result Set**: methods uses JDBC to interact with the MySQL database and It checks if the "scores" table exists and creates it if necessary and inserts the username and score into the database using SǪL statements (INSERT INTO scores (username, score) VALUES (?,?)).

**Code:-**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

public class javaquiz extends JFrame {

    private JTextField usernameField;

    private JPasswordField passwordField;

    private String[] correctAnswers = {

            "false",

            "2HelloWorld",

            "public void calculate(int a);",

            "Only one thread at a time can execute the method on a given instance of a class",

            "Sorts the list in ascending order based on natural ordering or using a comparator provided"

    };

    public javaquiz() {

        setTitle("Admission Entrance Test");

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setSize(400, 200);

        setLocationRelativeTo(null);

        JLabel titleLabel = new JLabel("Manipal Institute Of Technology");

        titleLabel.setFont(new Font("Arial", Font.BOLD, 18));

        titleLabel.setForeground(Color.BLUE);

        titleLabel.setHorizontalAlignment(SwingConstants.CENTER);

        add(titleLabel, BorderLayout.NORTH);

        JPanel loginPanel = new JPanel();

        loginPanel.setLayout(new GridLayout(3, 2));

        loginPanel.setBackground(Color.LIGHT\_GRAY);

        JLabel usernameLabel = new JLabel("Username:");

        JLabel passwordLabel = new JLabel("Password:");

        usernameField = new JTextField(20);

        passwordField = new JPasswordField(20);

        JButton loginButton = new JButton("Login");

        loginButton.setBackground(Color.BLUE);

        loginButton.setForeground(Color.GREEN);

        loginPanel.add(usernameLabel);

        loginPanel.add(usernameField);

        loginPanel.add(passwordLabel);

        loginPanel.add(passwordField);

        loginPanel.add(new JLabel());

        loginPanel.add(loginButton);

        add(loginPanel, BorderLayout.CENTER);

        loginButton.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                String username = usernameField.getText();

                String password = new String(passwordField.getPassword());

                if (username.equals("ashwini") && password.equals("ashwini@22")) {

                    openMCQPage();

                } else {

                    JOptionPane.showMessageDialog(null, "Invalid credentials. Try again.");

                }

            }

        });

        setVisible(true);

    }

    private void openMCQPage() {

        JFrame mcqFrame = new JFrame("Java Quiz");

        mcqFrame.setSize(700, 500);

        mcqFrame.setLocationRelativeTo(null);

        mcqFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        JPanel mcqPanel = new JPanel();

        mcqPanel.setLayout(new BoxLayout(mcqPanel, BoxLayout.Y\_AXIS));

        JLabel titleLabel = new JLabel("Java Quiz");

        titleLabel.setFont(new Font("Arial", Font.BOLD, 18));

        titleLabel.setForeground(Color.ORANGE);

        titleLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

        mcqPanel.add(titleLabel);

        addMCQ(mcqPanel,

                "Question 1:What is the default value of a boolean variable in Java ?",

                new String[] {

                        "true",

                        "false",

                        "null",

                        "0" });

        addMCQ(mcqPanel, "Question 2:Which of the following is not a valid Java identifier?",

                new String[] {

                        "\_\_helloWorld",

                        "$helloWorld",

                        "2HelloWorld",

                        "hello\_World" });

        addMCQ(mcqPanel, "Question 3:Which method signature is correctly overridden from a superclass?",

                new String[] {

                        "public void calculate(int a);",

                        "protected void calculate(int a);",

                        "public int calculate(int a)",

                        "public void calculate(double a);"});

        addMCQ(mcqPanel, "Question 4: What does the synchronized keyword ensure when used in a Java method?",

                new String[] {

                        "The method can be accessed by multiple threads simultaneously",

                        "The method compiles into synchronized bytecode instructions only",

                        "Only one thread at a time can execute the method on a given instance of a class",

                        "The method execution will be faster as it uses optimized threading" });

        addMCQ(mcqPanel, "Question 5:What does the Collections API Collections.sort() method do when applied to a List?",

                new String[] {

                        "Reverses the order of elements in the list",

                        "Randomizes the order of elements in the list",

                        "Sorts the list in descending order",

                        " Sorts the list in ascending order based on natural ordering or using a comparator provided" });

        JScrollPane scrollPane = new JScrollPane(mcqPanel);

        mcqFrame.add(scrollPane);

        JButton submitButton = new JButton("Submit");

        submitButton.setAlignmentX(Component.CENTER\_ALIGNMENT);

        submitButton.setBackground(Color.YELLOW);

        submitButton.setForeground(Color.WHITE);

        mcqPanel.add(submitButton);

        submitButton.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                int score = calculateScore(mcqPanel);

                String username = usernameField.getText();

                JOptionPane.showMessageDialog(null,username + " Your score: " + score);

                storeScoreInDatabase(username, score);

            }

        });

        mcqFrame.setVisible(true);

        setVisible(false);

    }

    private void addMCQ(JPanel panel, String question, String[] options) {

        JLabel questionLabel = new JLabel(question);

        questionLabel.setFont(new Font("Arial", Font.PLAIN, 14));

        panel.add(questionLabel);

        ButtonGroup group = new ButtonGroup();

        for (String option : options) {

            JRadioButton radioButton = new JRadioButton(option);

            radioButton.setFont(new Font("Arial", Font.PLAIN, 15));

            group.add(radioButton);

            panel.add(radioButton);

        }

    }

    private int calculateScore(JPanel panel) {

        int score = 0;

        Component[] components = panel.getComponents();

        for (Component component : components) {

            if (component instanceof JRadioButton) {

                JRadioButton radioButton = (JRadioButton) component;

                if (radioButton.isSelected() && isCorrectAnswer(radioButton.getText())) {

                    score++;

                }

            }

        }

        return score;

    }

    private boolean isCorrectAnswer(String selectedAnswer) {

        for (String correctAnswer : correctAnswers) {

            if (selectedAnswer.equals(correctAnswer)) {

                return true;

            }

        }

        return false;

    }

    private void storeScoreInDatabase(String username, int score) {

        String url = "jdbc:mysql://localhost:3306/Scores";

        String user = "root";

        String password = "ashwini@123";

        try {

            Connection connection = DriverManager.getConnection(url, user, password);

            String checkTableExistsSQL = "SHOW TABLES LIKE 'scores'";

            PreparedStatement checkTableExistsStatement = connection.prepareStatement(checkTableExistsSQL);

            ResultSet resultSet = checkTableExistsStatement.executeQuery();

            if (!resultSet.next()) {

                String createTableSQL = "CREATE TABLE scores (id INT AUTO\_INCREMENT PRIMARY KEY, username VARCHAR(255), score INT)";

                Statement createTableStatement = connection.createStatement();

                createTableStatement.executeUpdate(createTableSQL);

            }

            String insertScoreSQL = "INSERT INTO scores (username, score) VALUES (?, ?)";

            PreparedStatement insertStatement = connection.prepareStatement(insertScoreSQL);

            insertStatement.setString(1, username);

            insertStatement.setInt(2, score);

            insertStatement.executeUpdate();

            JOptionPane.showMessageDialog(null, username + " Your score has successfully submitted", "System Message",

                  JOptionPane.INFORMATION\_MESSAGE);

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

    public static void main(String[] args) {

        SwingUtilities.invokeLater(new Runnable() {

            @Override

            public void run() {

                new javaquiz();

            }

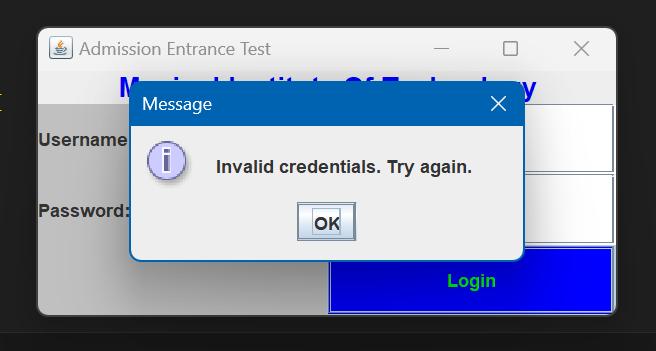
        });

    }

}

**Outputs:-**

* **When user enters wrong credentials**



* **When user enters right credentials**

A screen shot of a computer

Description automatically generated

**Display quiz window:-**

A screenshot of a computer

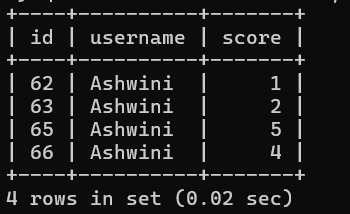
Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer error

Description automatically generated



**References:**

* **Swings Tutorial by Bro**

**Code:(https://www.youtube.com/watch?v=Kmgo00avvEw&t=15676s&ab\_ channel=BroCode)**

* **Ǫuiz App tutorial by Geeks for Geeks:** (**https://www.youtube.com/watch?v=utC- 8xeEǪǪA&ab\_channel=GeeksforGeeks)**

* **Java Point for swings: (https://www.javatpoint.com/java-swing)**

* **Geeks for Geeks for Swings:(**[**https://www.geeksforgeeks.org/introduction-to-javaswing/)**](https://www.geeksforgeeks.org/introduction-to-java-swing/)

* **Geeks for Geeks for JDBC:(https://www.geeksforgeeks.org/introduction-to-jdbc/)**

* **Indian Programmer JDBC tutorial:(https://www.youtube.com/watch?v=TcJZǪvDE1ow&ab\_channel=In dianProgrammer)**