ANJUMAN - I - ISLAM'S
A. R. KALSEKAR
POLYTECHNIC, NEW PANVEL
ISO 9001:2015 CERTIFIED

Symbol of Secularism & National Integration

DTE Institute Code 3278 Raigad District

Approved by : All India Council for Technical Education (A.I.C.T.E.)

Recognized by : Directorate of Technical Education (D.T.E.) & Govt. of Maharashtra

Affiliated to : Maharashtra State Board of Technical Education (M.S.B.T.E.)

Vision : To be the most sought-after Polytechnic that others would wish to emulate.

Mission : Creating Exuberant Technical Professionals

A Micro-Project Report

On

"Quiz Application"

Submitted on "05-11-24"

By

1. 2205690362 – Arqam Qazi
2. 2205690348 - Furquan Saiyed

Under Guidance of

" Iffat Ma'am "

In

Three Years Diploma Program in Engineering & Technology of
Maharashtra State Board of Technical Education, Mumbai
(Autonomous)

ISO 9001:2015

At

Anjuman-I-Islam's Abdul Razzaq Kalsekar Polytechnic

Academic Year [2024 - 2025]

MAHARASHTRA STATE BOARD OFTECHNICAL EDUCATION, MUMBAI

Certificate

This is to certify that Mr. Arqam Qazi
Enrollment No: 2205690362 of 5th Semester of **Diploma in Computer Engineering** at **Anjuman I Islam's Abdul Razzak Kalsekar Polytechnic,** has completed the **Micro Project** satisfactorily in Subject _________AJP______ in the academic year 2024-2025 as per the MSBTE prescribed curriculum of I Scheme.

Place: Panvel

Enrollment No: 220560352

Date: 21/10/24

Exam Seat No:

Project Guide **Head of the Department** **Principal**

Head of Institute

MAHARASHTRA STATE BOARD OFTECHNICAL EDUCATION, MUMBAI

Certificate

This is to certify that Mr. Saiyed Furquanahmed Barkatali Enrollment No: 2205690348 of 5th Semester of **Diploma in Computer Engineering** at **Anjuman I Islam's Abdul Razzak Kalsekar Polytechnic,** has completed the **Micro Project** satisfactorily in Subject ________AJP________ in the academic year 2024-2025 as per the MSBTE prescribed curriculum of I Scheme.

Place: Panvel	Enrollment No: 220560348
Date: 21/10/24	Exam Seat No:

Project Guide **Head of the Department** **Principal**

Head of Institute

Annexure – I

Micro-Project On

Quiz Application

1.0 Aims/Benefits of the Micro-Project:

The micro-project on a Quiz Application using Advanced Java Programming (AJP) aims to build skills in creating interactive applications. Participants will gain experience in backend logic, session management, and dynamic content integration, enhancing their Java programming abilities.

2.0 Course Outcomes Addressed:

a) Develop programs using GUI Framework (AWT and Swing).
e) Develop programs using database.
f) Develop programs using Servlets.

3.0 Proposed Methodology:

- **Develop Application:** Build the quiz app with features like question bank, user login, and session management using Advanced Java.
- **Test Functionality:** Check for smooth performance and fix any issues.
- **Optimize Performance:** Improve backend processes for faster response and efficient resource use.

4.0 Action Plan:

Sr No.	Details of Activity	Planned Start Date	Planned Finish Date	Name of the Responsible Team Members
1	Quiz Application Development	28/10/24	31/10/24	Arqam Qazi & Furquan Saiyed
2	Report	01/10/24	02/10/24	Arqam Qazi & Furquan Saiyed

5.0 Resources Required:

Sr No.	Name of Resources/Material	Specifications	Qty.	Remarks
1	Laptop	8.00 GB Ram windows 11	2	
2	Microsoft Word & Eclipse	Software	2	

Names of Team Members with Enrollment No.:

1. 2205690362 – Arqam Qazi
2. 2205690348 - Furquan Saiyed

Annexure – II

Micro-Project On

Memory Management

1.0 Rationale (Importance of project):

A Quiz Application is vital for enhancing user engagement and learning outcomes. This project emphasizes the need for an interactive platform that supports efficient question management and user interaction. By implementing features like real-time scoring and dynamic question banks, the application aims to provide a seamless experience, improve user satisfaction, and facilitate effective learning. Exploring best practices in application design will ensure high performance and reliability, making the quiz experience enjoyable and informative.

2.0 Aims/Benefits of the Micro-Project:

The micro-project on a Quiz Application using Advanced Java Programming (AJP) aims to build skills in creating interactive applications. Participants will gain experience in backend logic, session management, and dynamic content integration, enhancing their Java programming abilities.

2.0 Course Outcomes Addressed:

a) Develop programs using GUI Framework (AWT and Swing).
e) Develop programs using database.
f) Develop programs using Servlets.

4.0 Literature Review:

The development of a Quiz Application requires efficient data management and user interaction to enhance performance and user experience. Research indicates that interactive applications benefit from effective question management systems and real-time feedback mechanisms, which significantly improve user engagement. Studies show that implementing a dynamic question bank and user-friendly interfaces leads to higher satisfaction rates among users. Furthermore, literature highlights the importance of data security and user authentication in maintaining user trust and application integrity. By exploring these aspects, the project aims to create a robust and reliable quiz application that fosters effective learning.

5.0 Actual Method Followed (Step wise execution):

The actual method followed for the project are as Follows:

- **Develop Application:** Build the quiz app with features like question bank, user login, and session management using Advanced Java.
- **Test Functionality:** Check for smooth performance and fix any issues.
- **Optimize Performance:** Improve backend processes for faster response and efficient resource use.

6.0 Actual Resources Used:

Sr No.	Name of Resources/Material	Specifications	Qty.	Remarks
1	Laptop	8.00 GB Ram windows 11	2	
2	Microsoft Word & Eclipse	Software	2	

Skill Developed:

1. Application Development Techniques
2. User Interface Design
3. Database Management
4. User Authentication and Security
5. Performance Testing and Optimization
6. Problem-Solving and Debugging

Application:

1. Interactive Learning and Assessment
2. User Engagement and Retention
3. Real-time Feedback and Analytics
4. Customizable Question Banks and Difficulty Levels

Names of Team Members with Enrollment No.:

1. 2205690362 – Arqam Qazi
2. 2205690348 - Furquan Saiyed

INDEX

1. Abstract
2. Introduction
3. Quiz Application
4. Code
5. Ouput
6. Conclusion
7. References

ABSTRACT

The Quiz Application is an interactive platform designed to enhance learning and assessment through user engagement. By implementing efficient resource allocation strategies, the application ensures users can easily access quiz content and receive immediate feedback. Features such as dynamic question banks and real-time scoring create a seamless experience for learners.

To accommodate multiple concurrent users and resource-intensive functionalities, the Quiz Application employs effective memory management techniques. These include caching frequently accessed questions and optimizing data retrieval processes to minimize latency and enhance responsiveness. Additionally, tracking user progress and integrating analytics enriches the learning experience and facilitates continuous improvement.

This paper highlights the significance of memory management in the Quiz Application, demonstrating its role in maximizing resource efficiency and maintaining system stability. The findings emphasize the importance of robust memory management strategies in developing effective educational tools that meet the demands of modern learners.

INTRODUCTION

The Quiz Application is designed to provide an engaging and interactive platform for learners to assess their knowledge and understanding of various subjects. In an era where technology plays a pivotal role in education, this application leverages user-friendly features and real-time feedback to enhance the learning experience. It allows users to participate in quizzes, track their progress, and receive instant results, making learning both effective and enjoyable.

Central to the functionality of the Quiz Application is efficient memory management, which ensures that the application can handle multiple users and resource-intensive features simultaneously. By optimizing memory allocation and retrieval processes, the application maintains smooth performance, minimizing delays and maximizing user satisfaction.

As educational needs evolve, the Quiz Application aims to provide a flexible and customizable experience, accommodating various question types and difficulty levels. This adaptability is crucial for catering to diverse learning styles and enhancing user engagement. Overall, the Quiz Application represents a significant step forward in utilizing technology to facilitate interactive learning and assessment in today's educational landscape.

Quiz Application

This quiz application uses **MVC** (Model View Controller).

MVC is an architectural pattern used for making application more manageable and scalable. It is useful for unit testing the application.

The project is divided into three parts

1. **Model**: It is used for managing the data from database.

2. **View**: It contains the UI of the application.

3. **Controller**: It act as a bridge between the UI and data. The data from the model can be displayed on the view using controller.

We have used **JDBC**, **swing** and **event handling.**

What our application does??

First there is a login screen and we have to enter the user name and password. And then it checks if the user name exists in a database and if the password is correct. If the password is correct then you are directed to the quiz otherwise not.

After the login you can give the quiz, after the quiz; the result is shown and it is stored in a database.

Code

Main.java

```
import View.*;
import Controller.*;

public class Main {
    public static void main(String[] args) {
        LoginPage loginView = new LoginPage();
        new LoginController(loginView);
    }
}
```

LoginModel.java

```
package Model;

public class LoginModel {
    private String userName;
    private String password;

    public LoginModel(String userName, String password) {
        this.userName = userName;
        this.password = password;
    }

    public String getUserName() { return userName; }
    public String getPassword() { return password; }
}
```

LoginDAO.java

```
package Model;

import java.sql.*;
import java.util.ArrayList;

public class LoginDAO {

    private static final String JDBC_URL = "jdbc:oracle:thin:@localhost:1521:xe";
    private static final String USERNAME = "system";
    private static final String PASSWORD = "123";

    public static ArrayList<LoginModel> getUsers() {
        ArrayList<LoginModel> users = new ArrayList<>();
        Connection connection = null;
        Statement statement = null;
        ResultSet resultSet = null;

        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");

            connection = DriverManager.getConnection(JDBC_URL, USERNAME, PASSWORD);

            statement = connection.createStatement();

            String query = "SELECT NAME, PASSWORD from users";

            resultSet = statement.executeQuery(query);

            while (resultSet.next()) {
                String userName = resultSet.getString("Name");
                String password = resultSet.getString("PASSWORD");

                users.add(new LoginModel(userName, password));
            }

        } catch (ClassNotFoundException e) {
            System.out.println("Oracle JDBC Driver not found. Add the ojdbc jar to your
classpath.");
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        } finally  {
            try {
                if (resultSet != null) { resultSet.close(); }
                if (connection != null) {connection.close(); }
                if (statement != null) {statement.close(); }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return users;
    }
}
```

LoginPage.java

```
package View;

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

public class LoginPage extends JFrame {
    private JTextField usernameField;
    private JPasswordField passwordField;
    private JButton loginButton;

    public LoginPage() {
        // Set frame properties
        setTitle("Quiz Application - Login");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        // Create a panel for the form
        JPanel panel = new JPanel();
        panel.setLayout(new GridBagLayout());
        panel.setBackground(new Color(240, 248, 255)); // Light background color

        // Create form elements
        JLabel titleLabel = new JLabel("Login");
        titleLabel.setFont(new Font("Arial", Font.BOLD, 24));
        titleLabel.setForeground(new Color(70, 130, 180)); // SteelBlue

        usernameField = new JTextField(15);
        usernameField.setFont(new Font("Arial", Font.PLAIN, 16));
        usernameField.setBorder(BorderFactory.createTitledBorder("Username"));

        passwordField = new JPasswordField(15);
        passwordField.setFont(new Font("Arial", Font.PLAIN, 16));
        passwordField.setBorder(BorderFactory.createTitledBorder("Password"));

        loginButton = new JButton("Login");
        loginButton.setFont(new Font("Arial", Font.BOLD, 16));
        loginButton.setBackground(new Color(70, 130, 180)); // SteelBlue
        loginButton.setForeground(Color.WHITE);
        loginButton.setFocusPainted(false);
        loginButton.setCursor(new Cursor(Cursor.HAND_CURSOR));

        // Add components to the panel with GridBagLayout
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10);
        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.gridwidth = 2;
        panel.add(titleLabel, gbc);

        gbc.gridx = 0;
        gbc.gridy = 1;
        gbc.gridwidth = 2;
        panel.add(usernameField, gbc);

        gbc.gridx = 0;
        gbc.gridy = 2;
        gbc.gridwidth = 2;
        panel.add(passwordField, gbc);

        gbc.gridx = 0;
        gbc.gridy = 3;
        gbc.gridwidth = 2;
        panel.add(loginButton, gbc);

        // Add the panel to the frame
        add(panel);
        setVisible(true);
    }

    public JTextField getUsernameField() {
        return usernameField;
    }
    public JTextField getPasswordField() {
        return passwordField;
    }
    public JButton getLoginButton() {
        return loginButton;
    }


    public static void main(String[] args) {
        // Run the LoginPage GUI
        new LoginPage();
    }
}
```

LoginController.java

```
package Controller;

import Model.LoginDAO;
import Model.LoginModel;
import View.LoginPage;
import View.QuizView;

import javax.swing.*;

import java.util.ArrayList;

public class LoginController {
    private LoginPage loginView;
    private ArrayList<LoginModel> userList;

    public LoginController(LoginPage view) {
        this.loginView = view;
        this.userList = LoginDAO.getUsers();

        loginView.getLoginButton().addActionListener(e -> {
            String userName = loginView.getUsernameField().getText();
            String password = loginView.getPasswordField().getText();
            boolean flag = false;
            for (LoginModel loginModel : userList) {
                String currentUserName = loginModel.getUserName();
                String currentPassword = loginModel.getPassword();

                if (currentUserName.equals(userName) && currentPassword.equals(password)) {
                    flag = true;
                    break;
                }
            }
            if (flag) {
                loginView.setVisible(false);
                JOptionPane.showMessageDialog(null, "Login Successful");
                QuizView quizView = new QuizView();
                System.out.println("quiz view initialized");
                QuizController quizController = new QuizController(quizView, userName);
                System.out.println("Quiz controller initialized");
            } else {
                JOptionPane.showMessageDialog(null, "Invalid Username or Password", "Error",
JOptionPane.ERROR_MESSAGE);
            }
        });

    }
}
```

QuestionModel.java

```
package Model;

public class QuestionModel {
    private String question;
    private String option1, option2, option3, option4;
    private int correctOption;

    public QuestionModel(String question, String option1, String option2, String option3, String
option4, int correctOption) {
        this.question = question;
        this.option1 = option1;
        this.option2 = option2;
        this.option3 = option3;
        this.option4 = option4;
        this.correctOption = correctOption;
    }

    public String getQuestion() { return question; }
    public String getOption1() { return option1; }
    public String getOption2() { return option2; }
    public String getOption3() { return option3; }
    public String getOption4() { return option4; }
    public int getCorrectOption() { return correctOption; }
}
```

QuestionDAO.java

```
package Model;

import java.sql.*;
import java.util.ArrayList;

public class QuestionDAO {
    private static final String JDBC_URL = "jdbc:oracle:thin:@localhost:1521:xe";
    private static final String USERNAME = "system";
    private static final String PASSWORD = "123";

    public static ArrayList<QuestionModel> getQuestionsFromDB() {
        ArrayList<QuestionModel> questions = new ArrayList<>();
        Connection connection = null;
        Statement statement = null;
        ResultSet resultSet = null;

        try {
            // Load Oracle JDBC Driver
            Class.forName("oracle.jdbc.driver.OracleDriver");

            // Establish the connection
            connection = DriverManager.getConnection(JDBC_URL, USERNAME, PASSWORD);
            // Create a statement
            statement = connection.createStatement();

            // SQL query to get all questions
            String query = "SELECT id, question_text, option1, option2, option3, option4,
correct_option FROM questions";

            // Execute the query and get the result set
            resultSet = statement.executeQuery(query);

            // Iterate through the result set and populate the questions list
            while (resultSet.next()) {
                int id = resultSet.getInt("id");
                String questionText = resultSet.getString("question_text");
                String option1 = resultSet.getString("option1");
                String option2 = resultSet.getString("option2");
                String option3 = resultSet.getString("option3");
                String option4 = resultSet.getString("option4");
                int correctOption = resultSet.getInt("correct_option");

                // Add the question to the list
                questions.add(new QuestionModel(questionText, option1, option2, option3, option4,
correctOption));
            }

        } catch (ClassNotFoundException e) {
            System.out.println("Oracle JDBC Driver not found. Add the ojdbc jar to your
classpath.");
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            // Close the resources
            try {
                if (resultSet != null) resultSet.close();
                if (statement != null) statement.close();
                if (connection != null) connection.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return questions;
    }

}
```

QuizView.java

```
package View;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class QuizView extends JFrame {
    JLabel question, option1, option2, option3, option4;
    JButton submit;

    public QuizView() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(null);
        setVisible(true);
        setBackground(Color.WHITE);
        setSize(600, 700);

        question = new JLabel("Hello this is the question");
        question.setFont(new Font("Times New Roman", Font.PLAIN, 22));
        question.setBounds(100, 100, 1100, 22);

        option1 = createMaterialLabel("Option1");
        option2 = createMaterialLabel("Option2");
        option3 = createMaterialLabel("Option3");
        option4 = createMaterialLabel("Option4");

        submit = createMaterialButton("SUBMIT");

        // Set bounds for all the options
        option1.setBounds(100, 150, 360, 60);
        option2.setBounds(100, 230, 360, 60);
        option3.setBounds(100, 310, 360, 60);
        option4.setBounds(100, 390, 360, 60);


        // Set padding for options
//        option1.setOpaque(true);
//        option3.setOpaque(true);
//
//        option4.setOpaque(true);

        submit.setBounds(100, 470, 360, 60);

        add(question);
        add(option1);
        add(option2);
        add(option3);
        add(option4);
        add(submit);
    }

    public static JButton createMaterialButton(String buttonText) {
        JButton button = new JButton(buttonText);
        button.setPreferredSize(new Dimension(260, 50));
        Color primaryColor = new Color(33, 150, 243);  // Material Design Blue
        button.setBackground(primaryColor);
        button.setForeground(Color.WHITE);
        button.setFont(new Font("Arial", Font.BOLD, 18));
        button.setFocusPainted(false);
        button.setMargin(new Insets(10, 10, 10, 10));
        button.setCursor(new Cursor(Cursor.HAND_CURSOR));

        button.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseEntered(MouseEvent e) {
                button.setBackground(primaryColor.darker());
            }

            @Override
            public void mouseExited(MouseEvent e) {
                button.setBackground(primaryColor);
            }
        });
        return button;
    }

        public static JLabel createMaterialLabel(String labelText) {
        JLabel label = new JLabel(labelText, SwingConstants.CENTER);
        label.setPreferredSize(new Dimension(360, 60));
        label.setFont(new Font("Arial", Font.PLAIN, 18));
        label.setHorizontalAlignment(SwingConstants.CENTER);
        label.setVerticalAlignment(SwingConstants.CENTER);
        label.setForeground(new Color(122, 128, 137));
        label.setBorder(BorderFactory.createEmptyBorder(15, 15, 15, 15));
        label.setOpaque(true);
        label.setBackground(Color.WHITE);
        label.setCursor(new Cursor(Cursor.HAND_CURSOR));
        return label;
    }

    public JButton getSubmitButton() {
        return submit;
    }

    public JLabel getOption1() {
        return option1;
    }

    public JLabel getOption2() {
        return option2;
    }

    public JLabel getOption3() {
        return option3;
    }

    public JLabel getOption4() {
        return option4;
    }

    public JLabel getQuestionLabel() {
        return question;
    }
    public static void main(String[] args) {
        new QuizView();
    }
}
```

QuizController.java

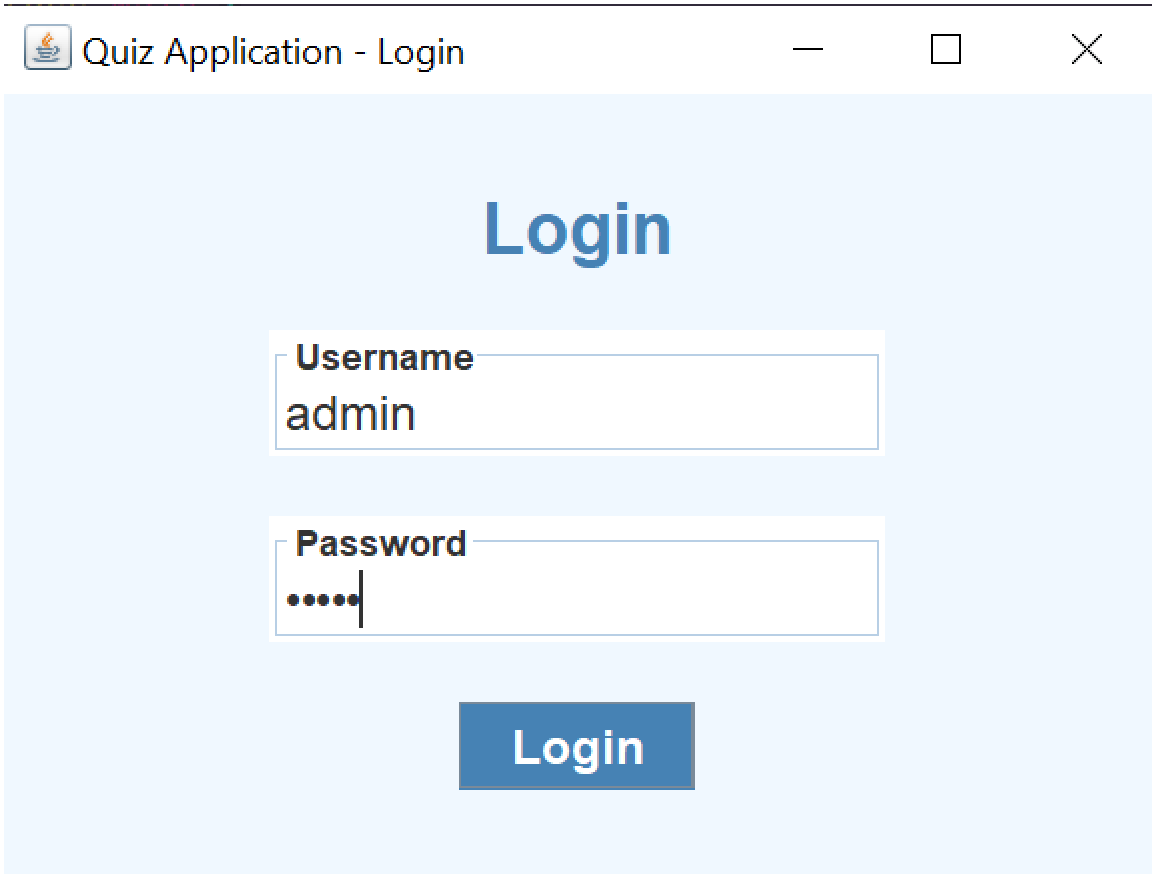
```
package Controller;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.util.ArrayList;

import Model.QuestionDAO;
import Model.QuestionModel;
import View.QuizView;
import View.ResultView;

public class QuizController {
    private QuizView quizView;
    private ArrayList<QuestionModel> questionList;
    private int selectedOptionPosition = 0;
    private int currentPosition = 1;
    private int correctAnswers = 0;

    public QuizController(QuizView view, String userName) {
        this.quizView = view;
        this.questionList = QuestionDAO.getQuestionsFromDB();

        setQuestion();

        quizView.getOption1().addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                selectedOptionView(quizView.getOption1(), 1);
            }
        });

        quizView.getOption2().addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                selectedOptionView(quizView.getOption2(), 2);
            }
        });

        quizView.getOption3().addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                selectedOptionView(quizView.getOption3(), 3);
            }
        });

        quizView.getOption4().addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                selectedOptionView(quizView.getOption4(), 4);
            }
        });

        quizView.getSubmitButton().addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                submitAnswer(userName);
            }
        });
    }

    public void setQuestion() {
        resetOptionView();
        QuestionModel question = questionList.get(currentPosition - 1);

        quizView.getQuestionLabel().setText(question.getQuestion());
        quizView.getOption1().setText(question.getOption1());
        quizView.getOption2().setText(question.getOption2());
        quizView.getOption3().setText(question.getOption3());
        quizView.getOption4().setText(question.getOption4());

        if (currentPosition == questionList.size()) {
            quizView.getSubmitButton().setText("FINISH");
        } else {
            quizView.getSubmitButton().setText("SUBMIT");
        }
    }

    private void selectedOptionView(JLabel option, int selectedNumber) {
        resetOptionView();
        selectedOptionPosition = selectedNumber;
        option.setBackground(Color.decode("#9013FE"));
        option.setForeground(Color.WHITE);
    }

    private void resetOptionView() {
        quizView.getOption1().setBackground(Color.WHITE);
        quizView.getOption1().setForeground(new Color(122, 128, 137));
        quizView.getOption2().setBackground(Color.WHITE);
        quizView.getOption2().setForeground(new Color(122, 128, 137));
        quizView.getOption3().setBackground(Color.WHITE);
        quizView.getOption3().setForeground(new Color(122, 128, 137));
        quizView.getOption4().setBackground(Color.WHITE);
        quizView.getOption4().setForeground(new Color(122, 128, 137));
    }

    private void submitAnswer(String userName) {
        if (selectedOptionPosition == 0) {
            currentPosition++;
            if (currentPosition <= questionList.size()) {
                setQuestion();
            } else {
                // Show result here
                new ResultView(userName, correctAnswers);
            }
        } else {
            QuestionModel question = questionList.get(currentPosition - 1);
            if (question.getCorrectOption() != selectedOptionPosition) {
                highlightAnswer(selectedOptionPosition, "#FF4444");
            } else {
                correctAnswers++;
            }
            highlightAnswer(question.getCorrectOption(), "#99CC00");
            if (currentPosition == questionList.size()) {
                quizView.getSubmitButton().setText("FINISH");
            } else {
                quizView.getSubmitButton().setText("NEXT");
            }
            selectedOptionPosition = 0;
        }
    }

    private void highlightAnswer(int option, String color) {
        switch (option) {
            case 1 -> {
                quizView.getOption1().setBackground(Color.decode(color));
                quizView.getOption1().setForeground(Color.WHITE);
            }
            case 2 -> {
                quizView.getOption2().setBackground(Color.decode(color));
                quizView.getOption2().setForeground(Color.WHITE);
            }
            case 3 -> {
                quizView.getOption3().setBackground(Color.decode(color));
                quizView.getOption3().setForeground(Color.WHITE);
            }
            case 4 -> {
                quizView.getOption4().setBackground(Color.decode(color));
                quizView.getOption4().setForeground(Color.WHITE);
            }
        }
    }
}
```

ResultView.java

```
package View;


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

public class ResultView extends JFrame {

    private JLabel scoreLabel;
    private JButton exitButton;
    Connection connection = null;
    Statement statement = null;
    private static final String JDBC_URL = "jdbc:oracle:thin:@localhost:1521:xe";
    private static final String USERNAME = "system";
    private static final String PASSWORD = "123";

    public ResultView(String name, int score) {
        setTitle("Quiz Application - Score");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        JPanel panel = new JPanel();
        panel.setLayout(new GridBagLayout());
        panel.setBackground(new Color(240, 248, 255)); // Light blueish background

        scoreLabel = new JLabel("Hello " + name + ", your score is: " + score);
        scoreLabel.setFont(new Font("Arial", Font.BOLD, 24));
        scoreLabel.setForeground(new Color(70, 130, 180)); // Steel Blue for text

        exitButton = new JButton("Exit");
        styleButton(exitButton);

        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10);

        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.gridwidth = 2;
        panel.add(scoreLabel, gbc);

        gbc.gridx = 0;
        gbc.gridy = 1;
        gbc.gridwidth = 2;
    panel.add(exitButton, gbc);

    exitButton.addActionListener(new ActionListener() {
      @Override
      public void actionPerformed(ActionEvent e) {
        System.exit(0);
      }
    });

    try {
      // Load Oracle JDBC Driver
      Class.forName("oracle.jdbc.driver.OracleDriver");

      // Establish the connection
      connection = DriverManager.getConnection(JDBC_URL, USERNAME, PASSWORD);

      // Create a statement
      statement = connection.createStatement();

      // SQL query to get all questions
      String query = "insert into scores values('" + name + "', " + score + ")";

      // Execute the query and get the result set
      statement.executeUpdate(query);

    } catch (ClassNotFoundException e) {
      System.out.println("Oracle JDBC Driver not found. Add the ojdbc jar to your
classpath.");
      e.printStackTrace();
    } catch (SQLException e) {
      e.printStackTrace();
    } finally {
      // Close the resources
      try {
        if (statement != null) statement.close();
        if (connection != null) connection.close();
      } catch (SQLException e) {
        e.printStackTrace();
      }
    }
```

```
        add(panel);
        setVisible(true);
    }

    // Method to style the Exit button
    private void styleButton(JButton button) {
        button.setFont(new Font("Arial", Font.BOLD, 14));
        button.setBackground(new Color(70, 130, 180)); // Steel Blue
        button.setForeground(Color.WHITE);
        button.setFocusPainted(false);
        button.setBorder(BorderFactory.createEmptyBorder(10, 20, 10, 20)); // Padding inside button
        button.setCursor(new Cursor(Cursor.HAND_CURSOR));
    }


    public static void main(String[] args) {

        new ResultView("Arqam", 85);
    }
}
```

Output



Quiz Application - Login

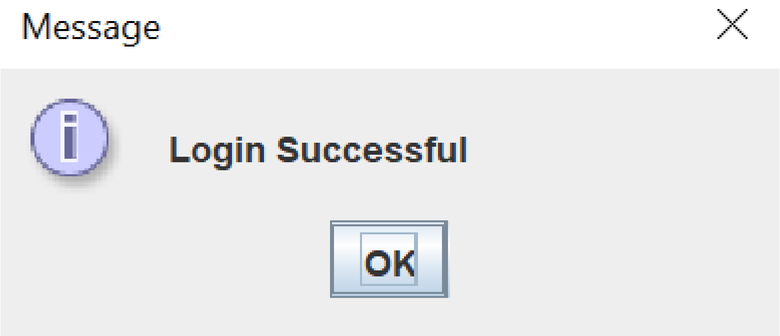
Login

Username

admin

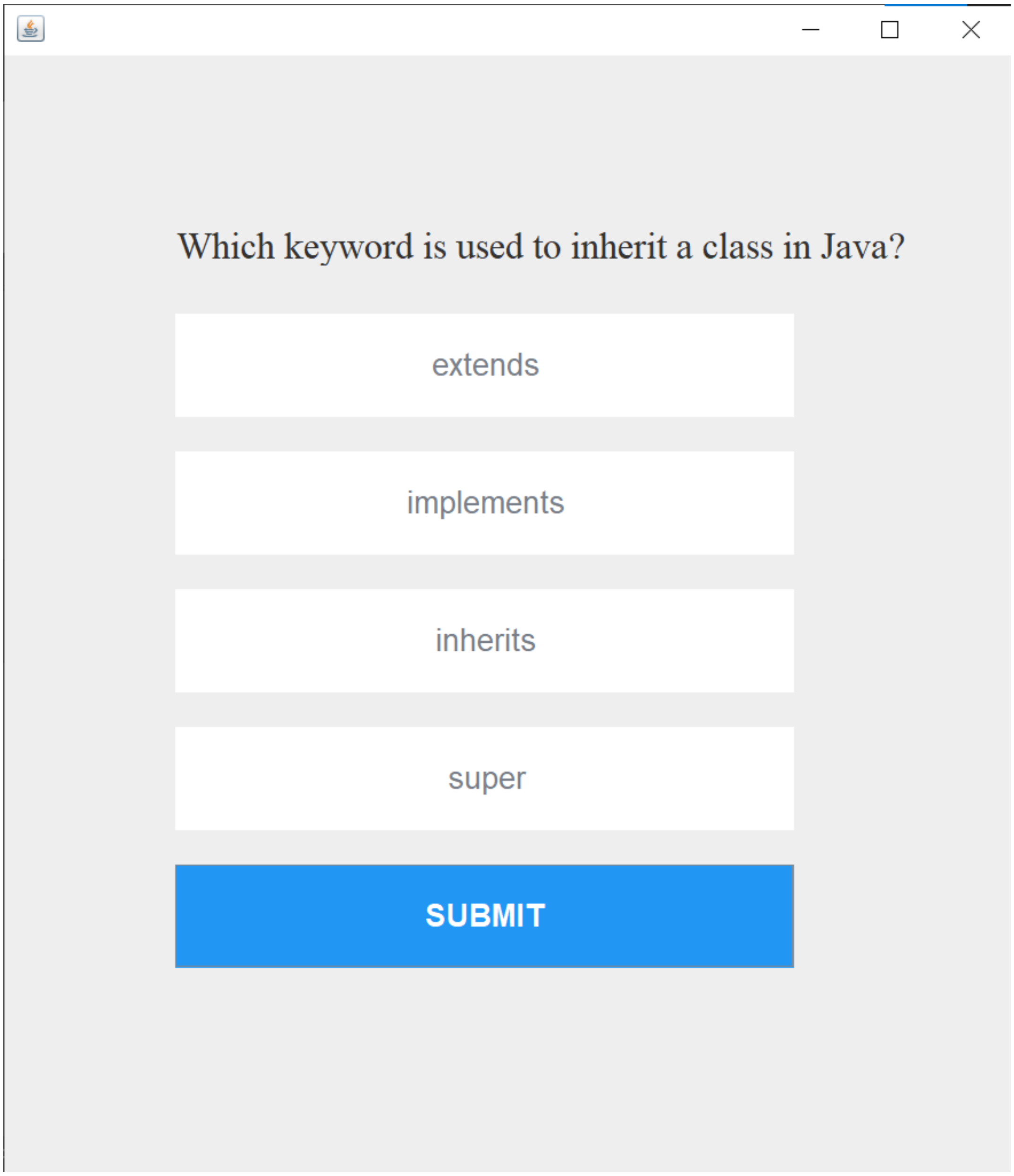
Password

•••••

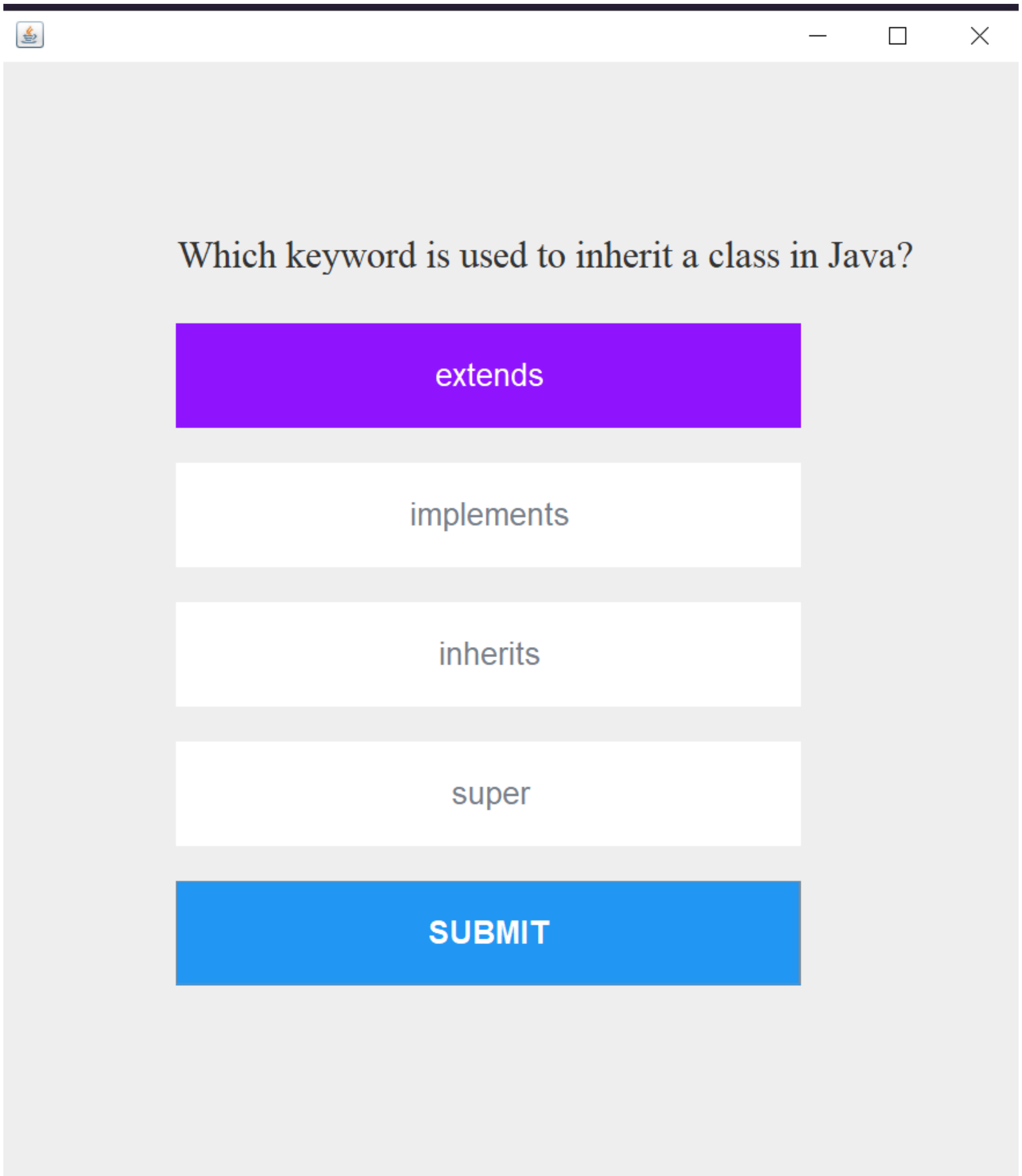


Message

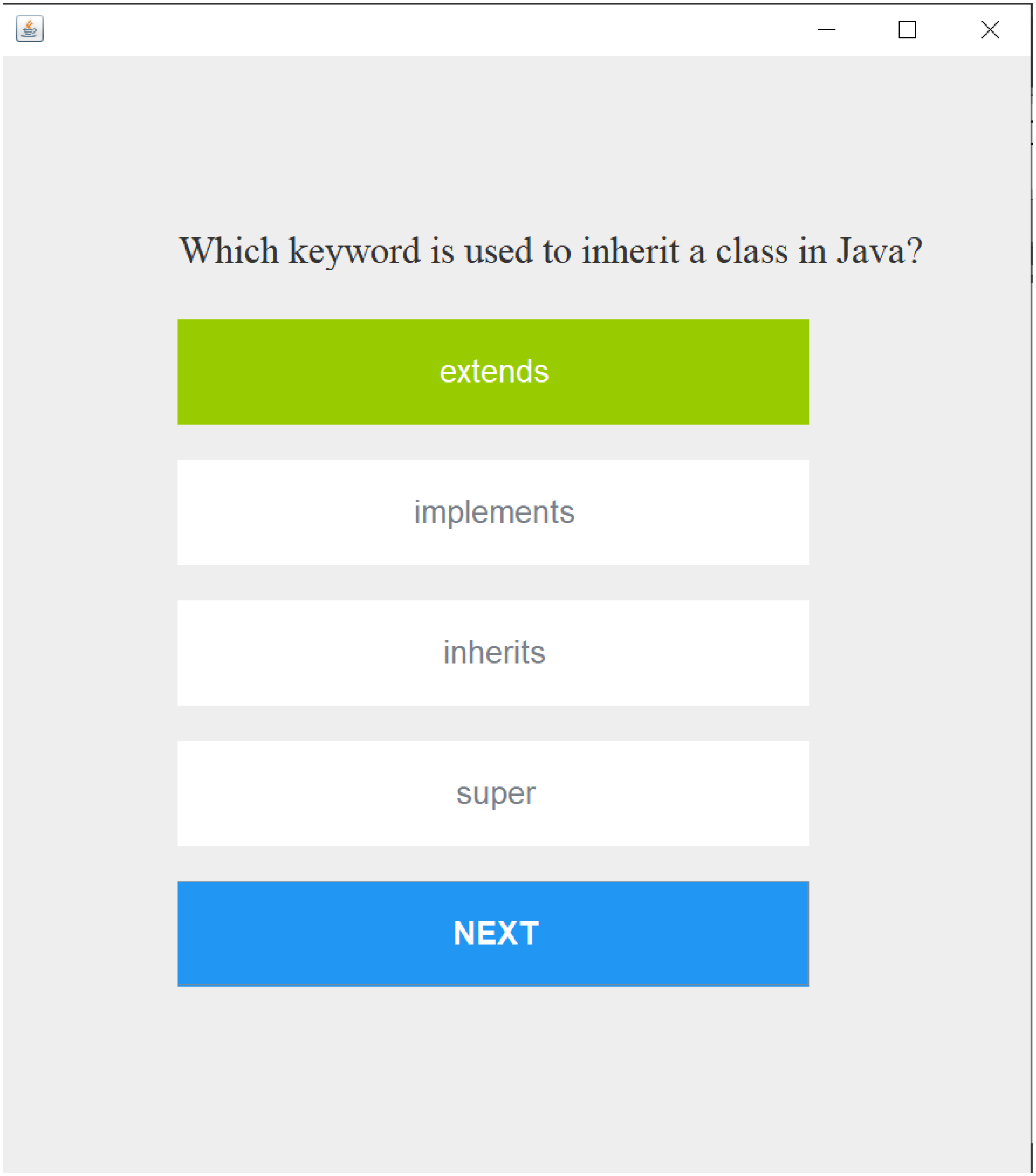
Login Successful



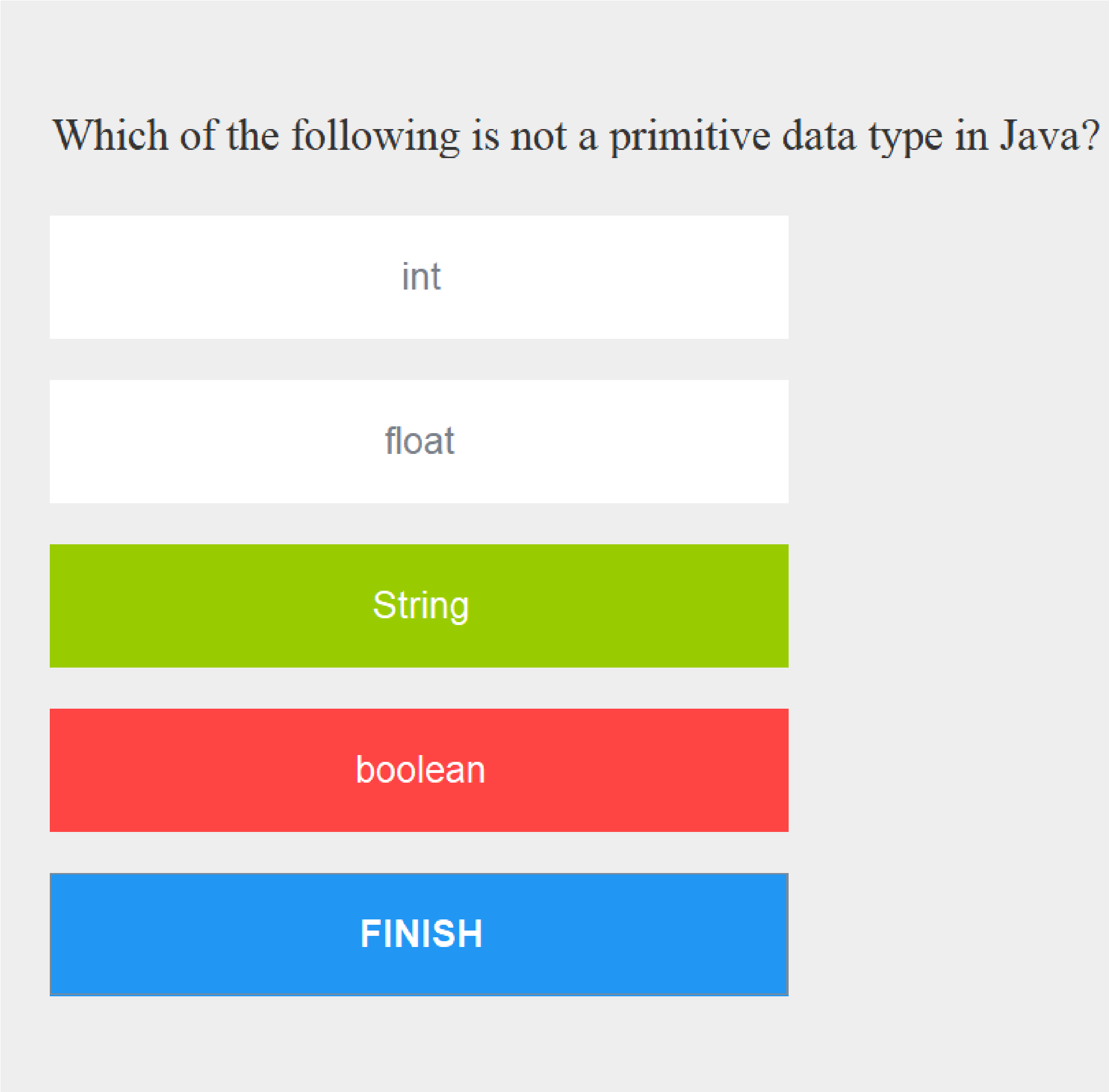
Which keyword is used to inherit a class in Java?



Which keyword is used to inherit a class in Java?



Which keyword is used to inherit a class in Java?



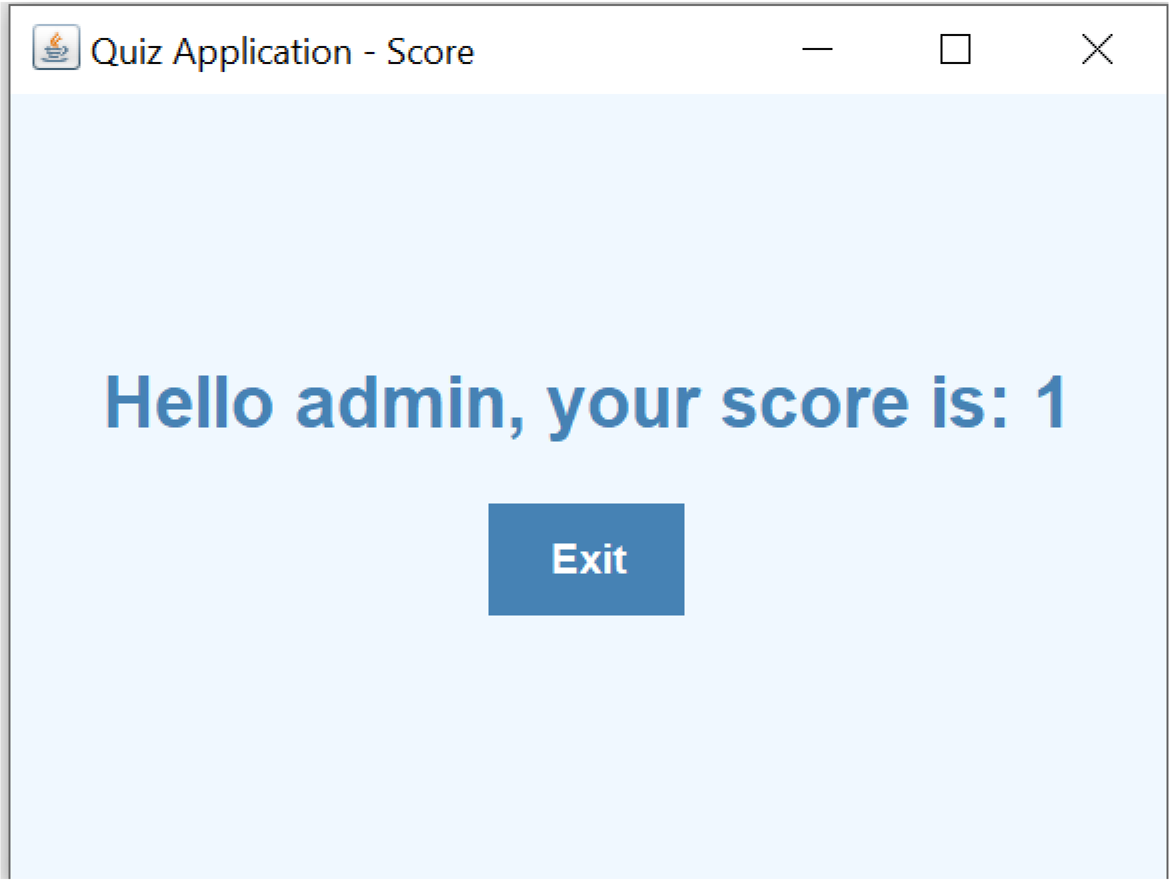
Which of the following is not a primitive data type in Java?

int

float

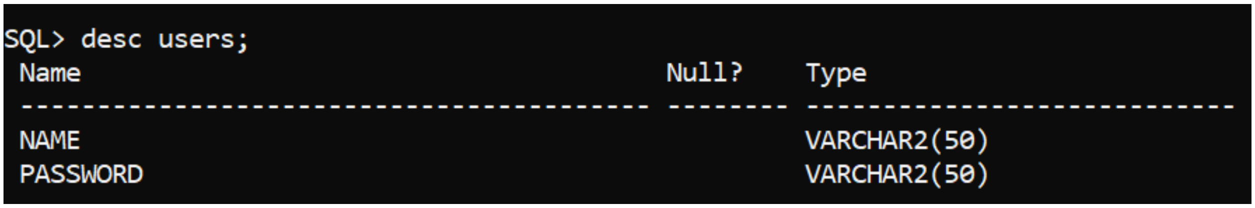
String

boolean

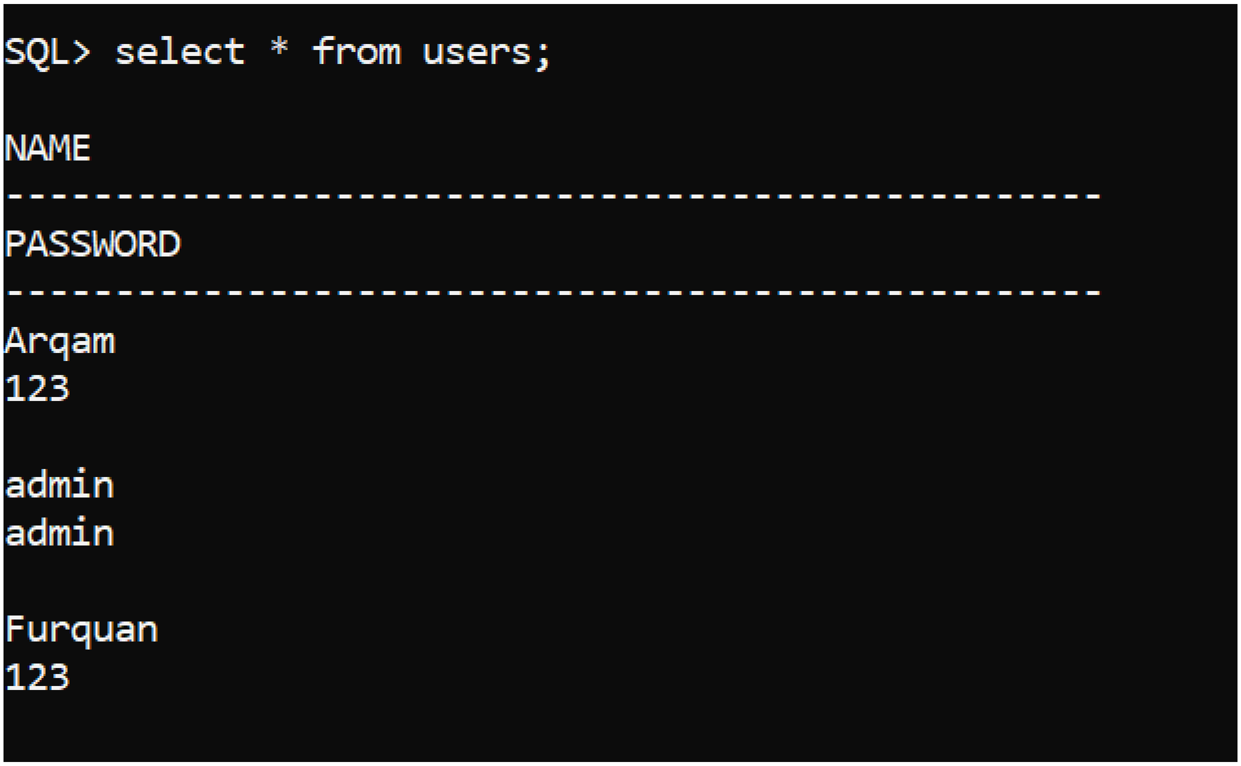


Hello admin, your score is: 1

Tables:

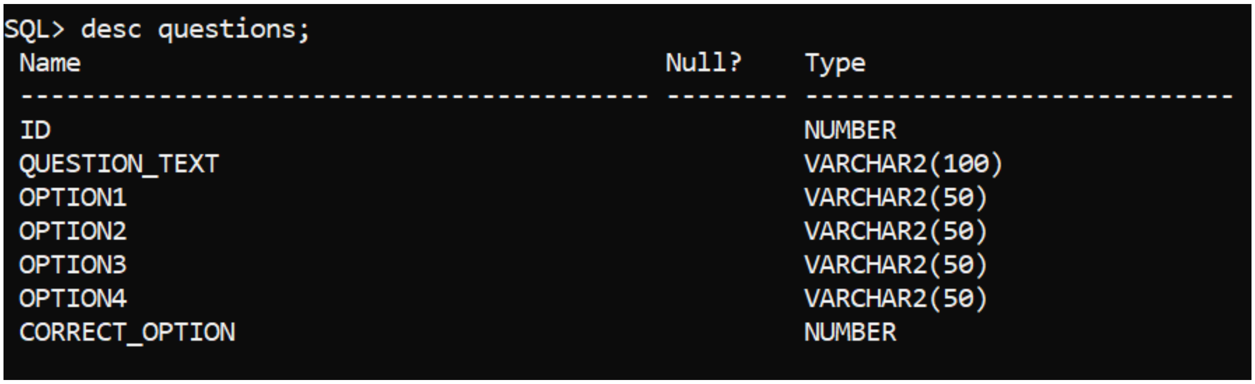


```
SQL> desc users;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NAME                                               VARCHAR2(50)
 PASSWORD                                           VARCHAR2(50)
```

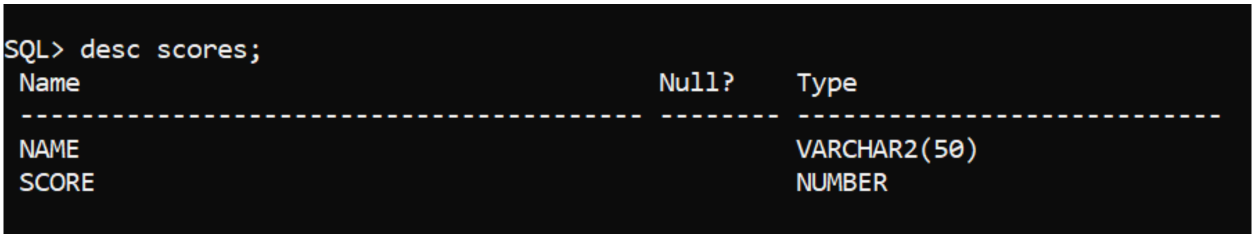


```
SQL> select * from users;

NAME
--------------------------------------------------
PASSWORD
--------------------------------------------------
Arqam
123

admin
admin

Furquan
123
```

Questions table



```
SQL> desc questions;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                                 NUMBER
 QUESTION_TEXT                                      VARCHAR2(100)
 OPTION1                                            VARCHAR2(50)
 OPTION2                                            VARCHAR2(50)
 OPTION3                                            VARCHAR2(50)
 OPTION4                                            VARCHAR2(50)
 CORRECT_OPTION                                     NUMBER
```



```
SQL> desc scores;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NAME                                               VARCHAR2(50)
 SCORE                                              NUMBER
```

Annexure IV

Micro Project Evaluation Sheet

Name of Student: ………………………………………… **Enrollment No:**…………..……………….

Name of Programme:………………………….……… **Semester:** …………………………………..

Course Title: ……………………………………………….. **Code:**……………………………..……..…

Title of the Micro-Project: ………………………………………………………………………………..

Course Outcomes Achieved:-

a) …….

b) ……..

c) …….

d) …….

Sr. No.	**Characteristic to be assessed**	**Poor (Marks 1 - 3)**	**Average (Marks 4 - 5)**	**Good (Marks 6 - 8)**	**Excellent (Marks 9- 10)**	**Sub Total**
(A) Process and Product Assessment (Convert above total marks out of 6 Marks)						
1	Relevance to the course					
2	Literature Review/information collection					
3	Completion of the Target as per project proposal					
4	Analysis of Data and representation					
5	Quality of Prototype/Model					
6	Report Preparation					
(B) Individual Presentation / Viva (Convert above total marks out of 4 Marks)						
7	Presentation					
8	Viva					

(A) **Process and Product Assessment** **(6 marks)**	**(B)** **Individual Presentation/Viva** **(4 marks)**	**Total Marks** **10**

Comments/Suggestions about team work/leadership/inter-personal communication (if any)

……

……

……

Name and designation of the Teacher……………………………………….

Dated Signature ……………………………………………………………….

Annexure IV

Micro Project Evaluation Sheet

Name of Student: ……………………………………… **Enrollment No:**……………………………

Name of Programme:………………………………… **Semester:** …………………………………..

Course Title: …………………………………………….. **Code:**……………………………………

Title of the Micro-Project: ……………………………………………………………………………..

Course Outcomes Achieved:-

a) ……………………………………………………………………………………………………

b) …………………………………………………………………………………………………...

c) ……………………………………………………………………………………………………

d) ……………………………………………………………………………………………………

Sr. No.	**Characteristic to be assessed**	**Poor (Marks 1 - 3)**	**Average (Marks 4 - 5)**	**Good (Marks 6 - 8)**	**Excellent (Marks 9- 10)**	**Sub Total**
(A) Process and Product Assessment (Convert above total marks out of 6 Marks)						
1	Relevance to the course					
2	Literature Review/information collection					
3	Completion of the Target as per project proposal					
4	Analysis of Data and representation					
5	Quality of Prototype/Model					
6	Report Preparation					
(B) Individual Presentation / Viva (Convert above total marks out of 4 Marks)						
7	Presentation					
8	Viva					

(A) **Process and Product Assessment** **(6 marks)**	**(B)** **Individual Presentation/Viva** **(4 marks)**	**Total Marks** **10**

Comments/Suggestions about team work/leadership/inter-personal communication (if any)

………

………

………

Name and designation of the Teacher……………………………………….

Dated Signature ……………………………………………………………….

ANNEXURE II

Evaluation Sheet for Micro Project

Academic Year: 2024-2025 **Name of Faculty:** Mrs. Iffat Salim

Course: AJP **Course Code:** 22517 **Semester:** 5th

Title of the Project: Quiz Application

COs Addressed by the Micro Project:

a) Develop programs using GUI Framework (AWT and Swing).
e) Develop programs using database.
f) Develop programs using Servlets.

Major Learning Outcomes achieved by the students during this Project:

(a) Practical Outcome:

- We implement memory management techniques in the Quiz Application to optimize performance and enhance user experience.

(b) Cognitive Domain Outcome:

- We gain a comprehensive understanding of application design principles and enhance our ability to analyze user engagement strategies and their impact on the Quiz Application's performance.

(c) Affective Domain Reflection:

- We cultivate a passion for creating engaging educational tools and develop a sense of responsibility toward enhancing the learning experience through thoughtful application design in the Quiz Application.

Comments/Suggestions about team work/leadership/inter-personal communication (if any)

…………………………………………………………………………………………………
…………………………………………………………………………………………………

Enrollment No.	**Students Name**	**Marks out of 6 for performance of group activity**	**Marks out of 4 for performance in oral/ presentation**	**Total out of 10**
2205690362	Arqam Qazi			
2205690348	Furquan Saiyed			

(Name and Signature of the Faculty)