# Online Examination Portal

VJ Vasudharini

# Introduction

Most schools, colleges, universities, and even Government and Private offices have switched from traditional paper-based examinations to online examinations. Due to the wide-spread applications, this project might help you learn more about a wide variety of skills.

# Outline of the project

The project should be capable of doing the following

....Loading

# Features

## 01 User Authentication

User login, Updating profile and passwords

## 02 Examination creation

Choosing questions from a question-bank

## 03 Attempting Exam

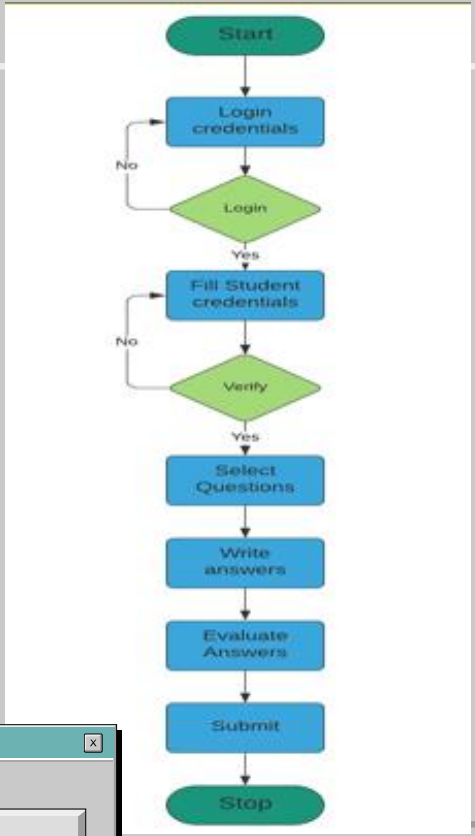
Allowing students to take exam.

## 04 Automatic Scoring

The system should score automatically and display it.

# Flowchart

Basic representation of the idea




A flowchart helps in easy understanding of the code’s workflow.

# Overview Of The Code:

1.The program starts by welcoming the user and attempting to authenticate them by asking for a username and password.
2. If the user provides the correct credentials, they are granted access to the exam.
3. The `generateQuestions` method creates a list of Java-related questions with answer choices and correct options.
4. The `conductExam` method presents these questions one by one to the user, collects their answers, and calculates the user's score.
5. After all questions have been answered, the program displays the user's score.

# Code Breakdown

Every class and method explained

# Class: OnlineExamSystem

```java
public class OnlineExamSystem {
    private static final String CORRECT_USERNAME = "user123";
    private static final String CORRECT_PASSWORD = "password123";

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Welcome to the Online Exam System");

        boolean isAuthenticated = authenticateUser(scanner);

        if (isAuthenticated) {
            System.out.println("Authentication successful. You can now attempt the questions.");
            List<Question> questions = generateQuestions();
            int score = conductExam(questions);
            System.out.println("Your score: " + score + " out of " + questions.size());
        } else {
            System.out.println("Authentication failed. Access denied.");
        }

        scanner.close();
    }
```

This is the main class of the program that serves as an entry point. It handles user authentication, question generation, and exam conduction.

Methods in this class:

- main Method
- authenticateUser Method
- generateQuestions Method
- conductExam Method
- createQuestion method

# Method: authenticateUser

```java
private static boolean authenticateUser(Scanner scanner) {
    System.out.print("Enter your username: ");
    String username = scanner.nextLine();

    System.out.print("Enter your password: ");
    String password = scanner.nextLine();

    return CORRECT_USERNAME.equals(username) && CORRECT_PASSWORD.equals(password);
}
```

- This method is responsible for authenticating the user by checking their username and password.
- It takes user input for the username and password, compares them with predefined correct values, and returns a boolean value indicating whether authentication was successful.

# Method: generateQuestion

```java
private static List<Question> generateQuestions() {
    List<Question> questions = new ArrayList<>();

    // Add 10 Java-related MCQs
    questions.add(createQuestion("What is a class in Java?", 0,
            "A blueprint for objects", "A type of car", "A programming language", "A software application"));
    questions.add(createQuestion("What is an interface in Java?", 0,
            "A contract for classes", "A kind of insect", "A musical instrument", "A type of food"));
    questions.add(createQuestion("Which package is directly available to our class without importing it?", 0,
            "swing", "applet", "net", "lang"));
    questions.add(createQuestion("String class is defined in which package?", 0,
            "lang", "Swing", "Applet", "awt"));
    questions.add(createQuestion("What does the acronym JRE stand for in the context of Java programming?", 2,
            "Java Runtime Environment", "Java Resource Editor", "Java Relational Engine", "Java Reference Engine"));
    questions.add(createQuestion("Which one among these is not a keyword?", 2,
            "class", "int", "get", "if"));
    questions.add(createQuestion("Which one among these is not a class?", 1,
            "Swing", "Actionperformed", "ActionEvent", "Button"));
    questions.add(createQuestion("Which one among these is not a function of Object class?", 3,
            "toString", "finalize", "equals", "getDocumentBase"));
    questions.add(createQuestion("Which function is not present in the Applet class?", 1,
            "init", "main", "start", "destroy"));
    questions.add(createQuestion("Which one among these is not a valid component?", 2,
            "JButton", "JList", "JButtonGroup", "JTextArea"));

    return questions;
}
```

- This method is used to create a list of questions for the exam. In this code, it generates Java-related multiple-choice questions (MCQs).
- Each question is represented as a `Question` object and added to the list.
- The questions are created with options and the correct option index.

# Method: conductExam

```java
private static int conductExam(List<Question> questions) {
    Scanner scanner = new Scanner(System.in);
    int score = 0;

    for (int i = 0; i < questions.size(); i++) {
        Question question = questions.get(i);
        System.out.println("Question " + (i + 1) + ": " + question.getQuestion());

        List<String> options = question.getOptions();
        for (int j = 0; j < options.size(); j++) {
            System.out.println((j + 1) + ". " + options.get(j));
        }

        System.out.print("Your answer (enter the option number): ");
        int userAnswerIndex = scanner.nextInt();

        if (userAnswerIndex >= 1 && userAnswerIndex <= options.size()) {
            if (question.isCorrect(userAnswerIndex - 1)) {
                System.out.println("Correct!\n");
                score++;
            } else {
                System.out.println("Incorrect. The correct answer is: "
                + options.get(question.getCorrectOptionIndex()) + "\n");
            }
        } else {
            System.out.println("Invalid input. Please enter a valid option number.");
            i--;
        }
    }

    return score;
}
```

- This method handles the exam process, where the user is presented with questions and is expected to select an answer.
- It uses a `for` loop to iterate through the list of questions and presents each question to the user.
- The user's answer is collected, evaluated for correctness, and the score is updated accordingly.
- At the end of the exam, the user's score is returned.

# Method:createQuestion

```java
    private static Question createQuestion(String question, int correctOptionIndex, String... options) {
        List<String> optionList = new ArrayList<>();
        for (String option : options) {
            optionList.add(option);
        }
        Question q = new Question(question, optionList);
        q.setCorrectOptionIndex(correctOptionIndex);
        return q;
    }
}
```

-This method is used to create a multiple-choice question.It takes the question text, the index of the correct answer, and a variable number of options.

- It converts the options into a list and sets the correct option index for the question.
- Finally, it returns the created Question object.

# Class: Question

```java
public class Question {
        private String question;
        private List<String> options;
        private int correctOptionIndex;

        public Question(String question, List<String> options) {
            this.question = question;
            this.options = options;
        }

        public String getQuestion() {
            return question;
        }

        public List<String> getOptions() {
            return options;
        }

        public int getCorrectOptionIndex() {
            return correctOptionIndex;
        }

        public void setCorrectOptionIndex(int correctOptionIndex) {
            this.correctOptionIndex = correctOptionIndex;
        }

        public boolean isCorrect(int userAnswerIndex) {
            return userAnswerIndex == correctOptionIndex;
        }
    }
```

The `Question` class represents an individual question in the exam.
Methods in this class:
- `isCorrect(int userAnswerIndex)`: This method checks if the user's answer (given by the index of the selected option) is correct by comparing it to the `correctOptionIndex`.
- `getQuestion()`: Returns the text of the question.
- `getOptions()`: Returns the list of answer choices.
- `getCorrectOptionIndex()`: Returns the index of the correct answer.

# Thanks!