****

Online Exam Management System

Project submitted to the

SRM University – AP , Andhra Pradesh

Submitted in the partial fulfillment of the requirement for the award of the degree of

**Bachelor of Technology**

**in**

**Computer Science and Engineering**

School of Engineering and Sciences

Submitted By

AP21110010608 : M.Pradeepika

AP21110010612 : CH.Rushendra Sai

AP21110010626 : P.Sai Kiran

AP21110010631: B.Uday Krishna

AP21110010649 : T.Chaitanya

Under the guidance of

Dr.L.Srinivas Rao Sir

**Department of Computer Science and Engineering**

**SRM University**

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh - 522502

**a) Aim of the project:**

The aim of the Online Exam Management System is to provide a platform for conducting online exams. It allows users to register, login, select an exam, answer questions, and view their exam results.

**b) Description of the project:**

The Online Exam Management System is implemented in Java and consists of several classes to manage user profiles, exams, questions, and exam results. Here is a breakdown of the key components:

1. UserProfile class:

* Stores information about a user, including name, age, and email.
* Provides methods to validate the user's name, age, and email.

1. ExamResult class:

* Represents the result of an exam taken by a user.
* Contains references to the user, exam, and the score obtained.
* Overrides the toString() method to display the result information.

1. Question class:

* Represents a single question in an exam.
* Stores the question, options, and the index of the correct answer.

1. Exam class:

* Represents an exam for a specific subject.
* Stores the subject and a list of questions.
* Provides methods to add questions and start the exam.

1. OnlineExamManagementSystem class:

* Contains the main method and serves as the entry point for the program.
* Manages user registration, login, exam selection, and result viewing.
* Implements various validation methods for name, age, and email.
* Creates instances of exams and populates them with questions.
* Keeps track of user profiles, exams, and exam results using ArrayLists.

**c)Individual roles and responsibilities:**

AP21110010631

-------[B.Uday Krishna](TEAM LEADER ) :

* Overall coordination and management of the project
* Ensuring project deadlines are met
* Tracking project progress
* Implemeting the functions

AP21110010608

------- [M.Pradeepika](TEAM MEMBER) :

* Algorithm implemetation
* Assisting the implementation of functions
* Collaborating with Uday and Chaitanya to ensure smooth integration of code
* Documentation

AP21110010612

------- [Ch.Rushendra Sai] (TEAM MEMBER):

* Designing and preparing test cases
* Analysing the results
* Identifying and reporting the issues,bugs,or unexpected behaviours

AP21110010649

------- [T.Chaitanya] (TEAM MEMBER):

* Implementation of main function
* Assisting other team members with necessary documentation
* Preparing required details (questions for each subject)

AP21110010625

------- [P.Sai Kiran] (TEAM MEMBER):

* Assisting In the implementation of main function
* Gathering references
* Collaborating with all the team mates to ensure proper resolution

**d) Code:**

import java.util.\*;

import java.util.regex.Pattern;

class UserProfile {

String name;

int age;

String email;

public UserProfile(String name, int age, String email) {

this.name = name;

this.age = age;

this.email = email;

}

@Override

public String toString() {

return "\t\tName: " + name + "\n\t\tAge: " + age + "\n\t\tEmail: " + email;

}

}

class ExamResult {

UserProfile user;

Exam exam;

int score;

public ExamResult(UserProfile user, Exam exam, int score) {

this.user = user;

this.exam = exam;

this.score = score;

}

@Override

public String toString() {

return "\n\n\n\t\t\t\t\t\t\t\tUser: " + user.name + "\n\n\t\t\t\t\t\t\t\tScore: " + score + " \tout of " + exam.questions.size();

}

}

class Question {

String question;

String[] options;

int correctAnswer;

public Question(String question, String[] options, int correctAnswer) {

this.question = question;

this.options = options;

this.correctAnswer = correctAnswer;

}

}

class Exam {

String subject;

List<Question> questions = new ArrayList<>();

public Exam(String subject) {

this.subject = subject;

}

public void addQuestion(Question question) {

questions.add(question);

}

public int startExam() {

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((i + 1) + ". " + question.question);

for (int j = 0; j < question.options.length; j++) {

System.out.println((j + 1) + ". " + question.options[j]);

}

System.out.print("\t\t\t\t\t\t\t\tEnter your answer (1-" + question.options.length + "): ");

int answer = scanner.nextInt();

if (answer == question.correctAnswer) {

score++;

}

}

return score;

}

}

public class OnlineExamManagementSystem {

public static boolean isValidName(String name) {

return Pattern.matches("^[a-zA-Z]{1,20}$", name);

}

public static boolean isValidEmail(String email) {

return Pattern.matches("^[a-zA-Z0-9.\_]+@[a-zA-Z0-9]+\\.[a-zA-Z]{2,}$", email);

}

public static boolean isValidAge(int age) {

return age >= 15 && age <= 99;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

List<Exam> exams = new ArrayList<>();

List<ExamResult> examResults = new ArrayList<>();

List<UserProfile> userProfiles = new ArrayList<>();

UserProfile currentUserProfile = null;

// Create exams

Exam javaExam = new Exam("Java");

javaExam.addQuestion(new Question("Who invented Java Programming?\n", new String[]{"Guido van Rossum","James Gosling","Dennis Ritchie","Bjarne Stroustrup"}, 2));

javaExam.addQuestion(new Question("Which statement is true about Java?\n", new String[]{"Java is a sequence-dependent programming language", "Java is a code dependent programming language", "Java is a platform-dependent programming language", "Java is a platform-independent programming language"}, 4));

javaExam.addQuestion(new Question("Which keyword is used to define a class in Java?\n", new String[]{"class", "public", "void", "static"}, 1));

javaExam.addQuestion(new Question("Which data type is used to represent a single character in Java?\n", new String[]{"char", "int", "String", "boolean"}, 1));

javaExam.addQuestion(new Question("What is the output of the following code?\nSystem.out.println(5 + 2 \* 3);\n", new String[]{"11", "21", "17", "15"}, 3));

exams.add(javaExam);

Exam htmlExam = new Exam("HTML");

htmlExam.addQuestion(new Question("What does HTML stand for?", new String[]{"Hyperlink and Text Markup Language", "Hyper Text Markup Language", "Hyper Transfer Markup Language", "Home Tool Markup Language"}, 2));

htmlExam.addQuestion(new Question("Which tag is used to define a hyperlink in HTML?", new String[]{"<a>", "<h1>", "<p>", "<div>"}, 1));

htmlExam.addQuestion(new Question("What is the correct HTML element for the largest heading?", new String[]{"<h1>", "<h6>", "<head>", "<heading>"}, 1));

htmlExam.addQuestion(new Question("Which HTML tag is used to define a table?", new String[]{"<table>", "<tr>", "<td>", "<th>"}, 1));

htmlExam.addQuestion(new Question("What is the correct HTML for inserting an image?", new String[]{"<img href='image.jpg' alt='MyImage'>", "<image src='image.jpg' alt='MyImage'>", "<img src='image.jpg' alt='MyImage'>", "<image href='image.jpg' alt='MyImage'>"}, 3));

exams.add(htmlExam);

Exam cssExam = new Exam("CSS");

cssExam.addQuestion(new Question("What does CSS stand for?", new String[]{"Cascading Style Sheet", "Colorful Style Sheet", "Computer Style Sheet", "Creative Style Sheet"}, 1));

cssExam.addQuestion(new Question("Which property is used to change the background color of an element?", new String[]{"color", "background-color", "background", "bgcolor"}, 2));

cssExam.addQuestion(new Question("Which CSS property is used to control the text size?", new String[]{"text-size", "font-style", "font-size", "text-style"}, 3));

cssExam.addQuestion(new Question("Which CSS property is used to control the spacing between lines of text?", new String[]{"line-height", "text-spacing", "letter-spacing", "word-spacing"}, 1));

cssExam.addQuestion(new Question("Which CSS property is used to add a background image to an element?", new String[]{"background-image", "image-url", "background-url", "image-source"}, 1));

exams.add(cssExam);

Exam javascriptExam = new Exam("JavaScript");

javascriptExam.addQuestion(new Question("Inside which HTML element do we put the JavaScript?", new String[]{"<js>", "<javascript>", "<script>", "<scripting>"}, 3));

javascriptExam.addQuestion(new Question("How do you write 'Hello World' in an alert box?", new String[]{"alert('Hello World');", "msgBox('Hello World');", "msg('Hello World');", "alertBox('Hello World');"}, 1));

javascriptExam.addQuestion(new Question("What is the correct way to write a JavaScript array?", new String[]{"var colors = 'red', 'green', 'blue';", "var colors = (1:'red', 2:'green', 3:'blue');", "var colors = ['red', 'green', 'blue'];", "var colors = 1 = ('red'), 2 = ('green'), 3 = ('blue');"}, 3));

javascriptExam.addQuestion(new Question("What is the correct way to write a JavaScript array?", new String[]{"[1, 2, 3, 4]", "{1, 2, 3, 4}", "(1, 2, 3, 4)", "1, 2, 3, 4"}, 1));

javascriptExam.addQuestion(new Question("Which operator is used to compare two values for equality in JavaScript?", new String[]{"==", "=", "===", "!=="}, 3));

exams.add(javascriptExam);

Exam cppExam = new Exam("C++");

cppExam.addQuestion(new Question("Which of the following is the correct syntax to declare a pointer?", new String[]{"int \*ptr;", "int ptr;", "int $ptr;", "int #ptr;"}, 1));

cppExam.addQuestion(new Question("Which of the following is not a fundamental data type in C++?", new String[]{"int", "float", "char", "string"}, 4));

cppExam.addQuestion(new Question("Which operator is used to allocate memory dynamically in C++?", new String[]{"new", "malloc", "allocate", "alloc"}, 1));

cppExam.addQuestion(new Question("What is the result of the following expression?\n5 / 2", new String[]{"2.5", "2", "2.0", "2.25"}, 2));

cppExam.addQuestion(new Question("What is the output of the following code?\nint x = 5;\nint y = ++x;\ncout << y;", new String[]{"5", "6", "0", "1"}, 2));

exams.add(cppExam);

Exam sqlExam = new Exam("SQL");

sqlExam.addQuestion(new Question("What does SQL stand for?", new String[]{"Structured Query Language", "Structured Question Language", "Simple Query Language", "Standard Query Language"}, 1));

sqlExam.addQuestion(new Question("Which keyword is used to retrieve data from a SQL database?", new String[]{"SELECT", "UPDATE", "DELETE", "INSERT"}, 1));

sqlExam.addQuestion(new Question("Which statement is used to add data to a SQL table?", new String[]{"ADD", "INSERT", "UPDATE", "CREATE"}, 2));

sqlExam.addQuestion(new Question("Which SQL clause is used to filter records based on a specified condition?", new String[]{"WHERE", "GROUP BY", "HAVING", "ORDER BY"}, 1));

sqlExam.addQuestion(new Question("What is the result of the following SQL query?\nSELECT MAX(Price) FROM Products;", new String[]{"Maximum value in the 'Price' column of the 'Products' table", "Minimum value in the 'Price' column of the 'Products' table", "Average value in the 'Price' column of the 'Products' table", "Total number of rows in the 'Price' column of the 'Products' table"}, 1));

exams.add(sqlExam);

while (true) {

System.out.println("\n\n\n\n\n\n\t\t\t\t\t------------------ Online Exam Management System ------------------ \n\n\n");

System.out.println("\t\t\t\t\t\t\t\t1. Register");

System.out.println("\t\t\t\t\t\t\t\t2. Login");

System.out.println("\t\t\t\t\t\t\t\t3. Start Exam");

System.out.println("\t\t\t\t\t\t\t\t4. View Exam Result");

System.out.println("\t\t\t\t\t\t\t\t5. Exit");

System.out.print("\t\t\t\t\t\t\t\tEnter your choice: ");

int choice = scanner.nextInt();

switch (choice) {

case 1:

scanner.nextLine(); // Clear the newline character

System.out.print("\t\t\t\t\t\t\t\tEnter your name: ");

String name = scanner.nextLine();

while (!isValidName(name)) {

System.out.println("\t\t\t\t\t\t\t\tInvalid name! Name should contain only alphabets and have a maximum length of 20 characters.");

System.out.print("\t\t\t\t\t\t\t\tEnter your name: ");

name = scanner.nextLine();

}

System.out.print("\t\t\t\t\t\t\t\tEnter your age: ");

int age = scanner.nextInt();

while (!isValidAge(age)) {

System.out.println("\t\t\t\t\t\t\t\tInvalid age! Age should be between 15 and 99.");

System.out.print("\t\t\t\t\t\t\t\tEnter your age: ");

age = scanner.nextInt();

}

scanner.nextLine(); // Clear the newline character

System.out.print("\t\t\t\t\t\t\t\tEnter your email: ");

String email = scanner.nextLine();

while (!isValidEmail(email)) {

System.out.println("\t\t\t\t\t\t\t\tInvalid email address!");

System.out.print("\t\t\t\t\t\t\t\tEnter your email: ");

email = scanner.nextLine();

}

UserProfile userProfile = new UserProfile(name, age, email);

userProfiles.add(userProfile);

System.out.println("\t\t\t\t\t\t\t\tRegistration successful!");

break;

case 2:

scanner.nextLine(); // Clear the newline character

System.out.print("\t\t\t\t\t\t\t\tEnter your email: ");

String loginEmail = scanner.nextLine();

boolean userFound = false;

for (UserProfile profile : userProfiles) {

if (profile.email.equals(loginEmail)) {

currentUserProfile = profile;

userFound = true;

break;

}

}

if (userFound) {

System.out.println("\t\t\t\t\t\t\t\tLogin successful!");

} else {

System.out.println("\t\t\t\t\t\t\t\tUser not found! Please register first.");

}

break;

case 3:

if (currentUserProfile == null) {

System.out.println("\t\t\t\t\t\t\t\tYou need to login first!");

} else {

System.out.println("\n\n\n\n\n\n\t\t\t\t\t------------------ Select an exam to start ------------------\n\n\n");

for (int i = 0; i < exams.size(); i++) {

System.out.println("\t\t\t\t\t\t\t\t"+(i + 1) + ". " + exams.get(i).subject);

}

System.out.print("\t\t\t\t\t\t\t\tEnter your choice: ");

int examChoice = scanner.nextInt();

if (examChoice >= 1 && examChoice <= exams.size()) {

Exam selectedExam = exams.get(examChoice - 1);

int score = selectedExam.startExam();

ExamResult examResult = new ExamResult(currentUserProfile, selectedExam, score);

examResults.add(examResult);

System.out.println("\t\t\t\t\t\t\t\tExam completed!");

} else {

System.out.println("\t\t\t\t\t\t\t\tInvalid choice!");

}

}

break;

case 4:

if (examResults.isEmpty()) {

System.out.println("\t\t\t\t\t\t\t\tNo exam results found!");

} else {

for (ExamResult result : examResults) {

System.out.println("\n -------------------------- Exam Result --------------------------");

System.out.println(result);

}

}

break;

case 5:

System.out.println("\t\t\t\t\t\t\t\tThank you for using the Online Exam Management System!");

System.exit(0);

break;

default:

System.out.println("\t\t\t\t\t\t\t\tInvalid choice!");

break;

}

}

}

}

**e) Explanation about the code:**

The provided code is an implementation of an Online Exam Management System in Java. It allows users to register, login, take exams, and view their exam results. Here is a complete explanation of the code:

1. The code begins by importing necessary Java classes and packages.
2. It defines a class called **“UserProfile,”** which represents a user's profile. It has three fields**: ”name”** , **”age”** , and **“email”** . The class also provides a constructor to initialize the fields and an overridden **“toString()”** method to display the profile information.
3. The next class”, is **“ExamResult “** which represents the result of an exam for a particular user. It contains three fields**:” user”** (of type **“UserProfile”**), **“exam”** (of type**” Exam”**), and “score”. It also has a constructor and an overridden **“toString()”** method to display the result information.
4. The **“Question”** class represents a single question in an exam. It has three fields: **“question”** **,” options”**, and **“correctAnswer”.** The constructor is used to initialize these fields.
5. The**” Exam”** class represents an exam for a specific subject. It has two fields: **“subject”** and a list of **” Question”** objects called **“questions”** . It provides methods to add questions to the exam and start the exam by displaying the questions and receiving user input for answers.
6. The **“OnlineExamManagementSystem”** class is the main class that contains the **“main()”** method. It handles the overall functionality of the exam management system.
7. The class provides several utility methods:

* **“isValidName(String name)”**: Validates a name by checking if it contains only alphabets and has a maximum length of 20 characters.
* **“isValidEmail(String email)”**: Validates an email address using a regular expression pattern.
* **“isValidAge(int age)”:** Validates an age by checking if it falls within the range of 15 to 99.

1. In the **“main()”** method, the code creates instances of different exams (Java, HTML, CSS, etc.) and adds questions to each exam.
2. It uses an infinite loop to display a menu to the user and handle user choices.

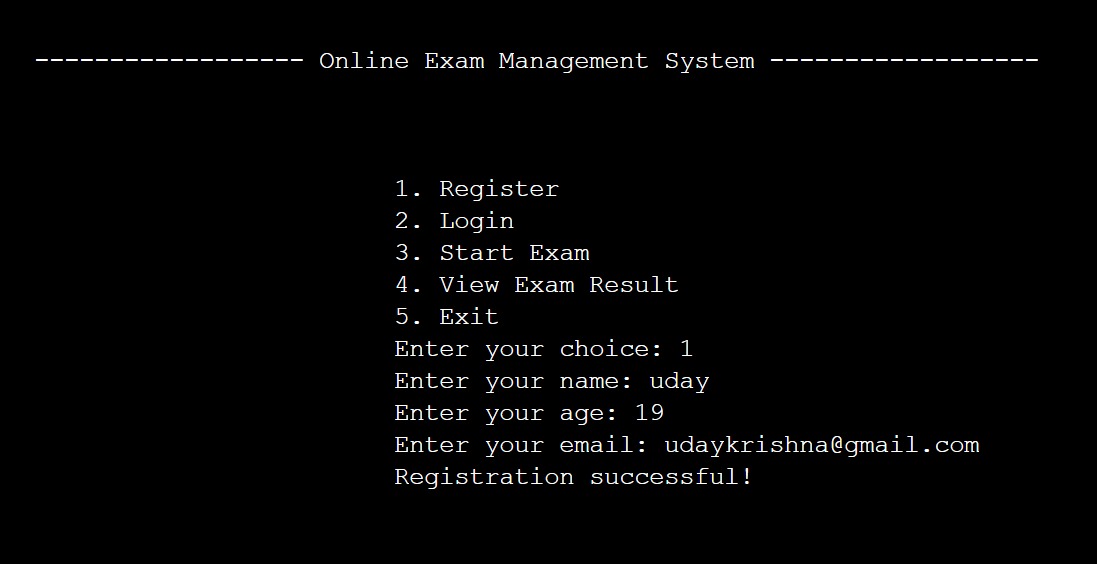
* Option 1: Register a new user by entering name, age, and email. User input is validated before creating a new **“UserProfile”** object.
* Option 2: Login with an existing email. The system searches for a matching user profile and sets the current user profile.
* Option 3: Start an exam. If a user is logged in, it displays the available exams and prompts the user to select one. The chosen exam is started, and the user's score is recorded.
* Option 4: View exam results. It displays the results of all exams taken so far.
* Option 5: Exit the program.

1. The code makes use of various input/output operations with the help of the **”Scanner”** class to read user input and display information on the console.
2. The main method also includes a sample setup of exams with questions and demonstrates the functionality of the exam management system.

Overall, this code provides a basic implementation of an online exam management system, allowing users to register, login, take exams, and view their results. It utilizes classes and objects to organize and represent different components of the system.

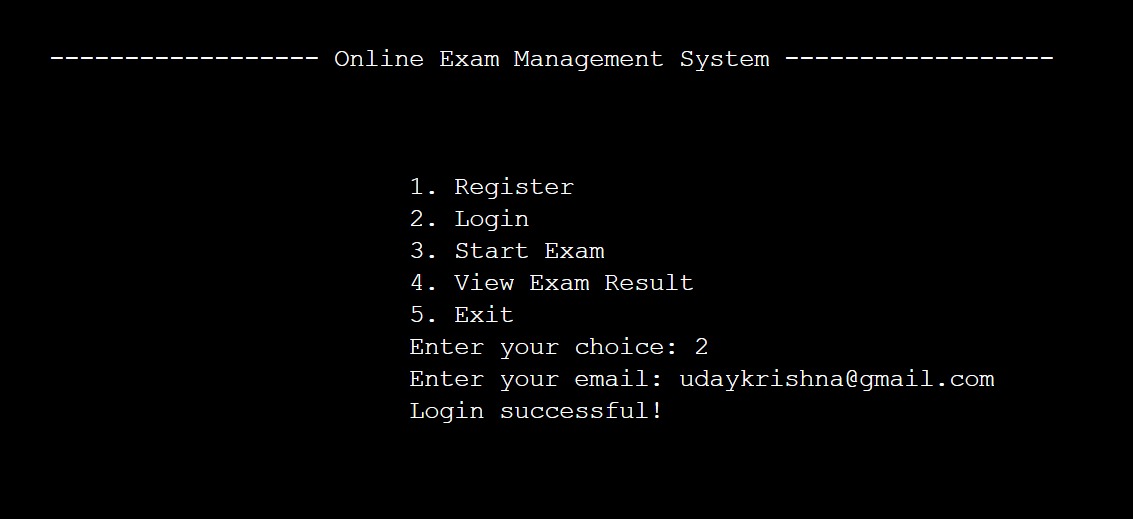
**f) Test cases and their outputs:**

Test Case 1: Registration

* Input:
  + Name: John Doe
  + Age: 25
  + Email: johndoe@example.com
* Output:
* Registration successful!
* 

Test Case 2: Login

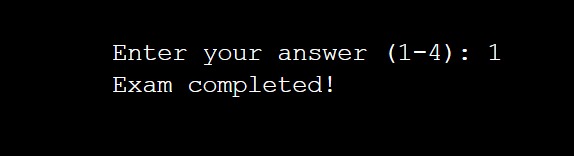
* Input:
* Email: johndoe@example.com
* Output:
* Login successful!



Test Case 3: Start Exam

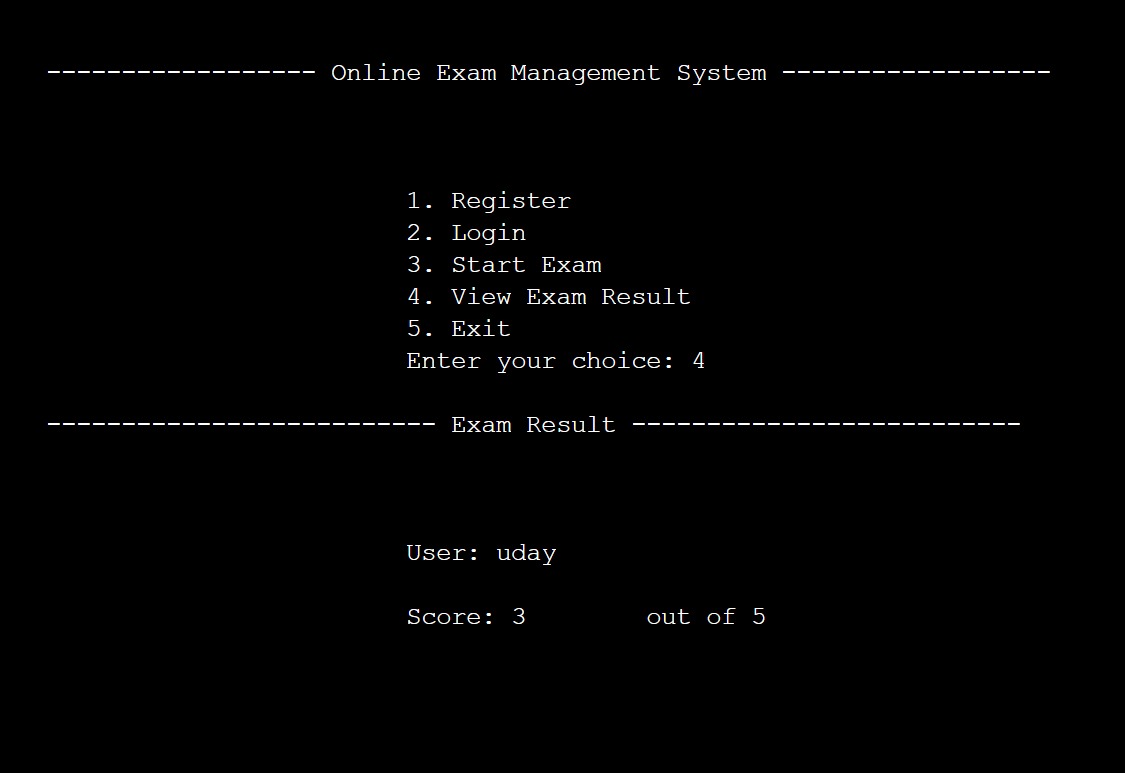
* Input:
* Exam choice: 1 (Java)
* Answers to the Java exam questions
* Output:
* Exam completed!





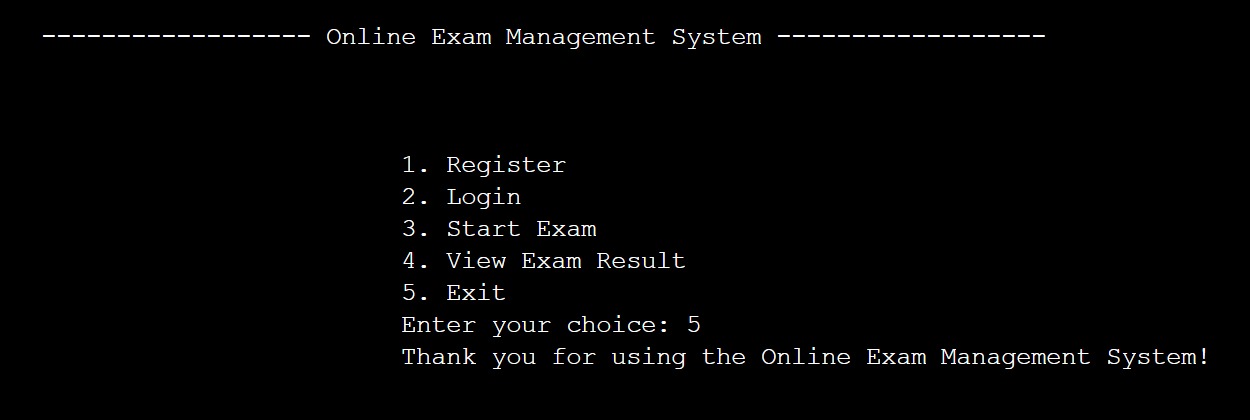
Test Case 4: View Exam Result

* Output:
* Exam Result:
* User: John Doe
* Score: X out of X



Test Case 5: Exit

* Output:
* Thank you for using the Online Exam Management System!



Note: The actual test cases would involve providing inputs during program execution and verifying the displayed outputs. The exact inputs and outputs depend on the specific actions taken during program execution, such as registering, logging in, selecting exams, answering questions, and viewing results.

**g) Conclusion:**

In conclusion, the Online Exam Management System is a Java-based project aimed at providing a platform for conducting online exams. It allows users to register, login, select exams, answer questions, and view their exam results. The system is implemented using various classes such as UserProfile, ExamResult, Question, Exam, and OnlineExamManagementSystem. These classes work together to manage user profiles, exams, questions, and exam results.

The project demonstrates the functionalities of user registration, login, exam selection, and result viewing. It also includes validation methods to ensure the correctness of user inputs such as name, age, and email. The system enables users to take exams by answering questions and calculates their scores based on the correct answers.

Overall, the Online Exam Management System provides a convenient and efficient way to conduct online exams, saving time and resources compared to traditional paper-based exams. It can be further enhanced with additional features such as multiple exam categories, and more comprehensive result analysis.

**\*\*\***