ASSIGNMENT 4

Name: RAMAVATH SANTHOSH

Roll No: 22MCF1R40

2nd Semester, MCA

Course: OOPS

Qu-1 **Online book recommendation system**

Recommendation systems were evolved as intelligent algorithms, which can generate results in the form of recommendations to users. They provide benefit to both the consumer and the manufacturer, by suggesting items to consumers, which can't be demanded until the recommendations. Every recommender system comprises of two entities, one is user and other is item. A user can be any customer or consumer of any product or items, who get the suggestions. Input to recommendation algorithm can be a database of user and items and output obliviously will be the recommendations. Input for this system is customers and book data and output of this book denotes the book recommendations. This project presents a new approach for recommending books to the buyers. This system consists of content filtering, collaborative filtering to produce efficient recommendations.

Modules and their Description

The system comprises of 2 major modules with their sub-modules as follows

User Modules:

· Registration / Login: User or customer should register with basic details to make an account for buying the books online. Registered User can login into the system with valid username and password.

· View Book Gallery: After Successful Login user can see all the available books to buy online. The book gallery contains front image of book, price, name etc. Clicking on particular book name it will show brief overview about that book.

· Recommendations: When User comes to website again the system will show the recommendation of books using collaborative filtering.

· Add to Cart/Buy Now: User can buy book with two options like add to cart and buy now. Within cart option User can add or remove items and after confirmation payment procedure starts.

· Online Payment: User can buy the book by doing online Payment with credit/debit card, while doing payment all the card details mandatory like card number, CVV, name on card etc.

· Order Details: With this option user can view which books he/she buy online with basic details of that book

Admin Modules:

· Login: Admin can login with valid username and password into system.

· Add Books: Admin have authority to add new books to sell online with image, price and basic details of it.

· View Books: Admin can see all the available books with its details like name, price etc.

· View Order Details: All the successful order details are visible to the admin.

· View Users: Can view all the registered users with their details.

Give the pseudo code for all the mentioned modules in this problem along with their functionality.

Qu-2 Consider a list of holidays that are given every year by the Govt.of India. Now, list out 4 such holidays that have occurred in the past 3 years. Sort this list based on their chronological occurrence. Next, hope you remember your date of birth! Now insert your DOB in this sorted list (not to mention that your b'day is also one of the holidays you wish to have).

For example consider the year 2019: 25-12-2019 Wednesday Christmas, 15-01-2019 Tuesday Makar sanktranti, 26-01-2019 Saturday Republic day. If your b'day falls on 01-04-2019, then your new sorted list would be 15-01-2019 Tuesday Makar sanktranti, 26-01-2019 Saturday Republic day, 01-04-2019 B'day 25-12-2019 Wednesday Christmas. Make sure you take at least 4 holidays per year and perform the sorting operation based on the event first occurred.

CODE:

```
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class HolidayList {
    public static void main(String[] args) {
        // Create a list of holidays for the past 3 years
        List<LocalDate> holidays = new ArrayList<>();
        holidays.add(LocalDate.parse("2018-01-26"));
        holidays.add(LocalDate.parse("2018-08-15"));
        holidays.add(LocalDate.parse("2018-10-02"));
        holidays.add(LocalDate.parse("2018-12-25"));
        holidays.add(LocalDate.parse("2019-01-26"));
        holidays.add(LocalDate.parse("2019-03-04"));
        holidays.add(LocalDate.parse("2019-04-14"));
        holidays.add(LocalDate.parse("2019-08-15"));
        holidays.add(LocalDate.parse("2019-10-02"));
        holidays.add(LocalDate.parse("2019-12-25"));
        holidays.add(LocalDate.parse("2020-01-26"));
        holidays.add(LocalDate.parse("2020-08-15"));
        holidays.add(LocalDate.parse("2020-10-02"));
        holidays.add(LocalDate.parse("2020-12-25"));
        holidays.add(LocalDate.parse("2021-01-26"));
        holidays.add(LocalDate.parse("2021-08-15"));
        holidays.add(LocalDate.parse("2021-10-02"));
        holidays.add(LocalDate.parse("2021-12-25"));

        // Sort the list by chronological occurrence
        Collections.sort(holidays);

        // Insert birthday into the list
        LocalDate myBirthday = LocalDate.parse("2000-09-16");
        for (int i = 0; i < holidays.size(); i++) {
            if (myBirthday.isBefore(holidays.get(i))) {
```

```
                holidays.add(i, myBirthday);
                break;
            }
        }

        // Print the sorted list
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-yyyy
EEEE");
        for (LocalDate holiday : holidays) {
            System.out.println(formatter.format(holiday));
        }
    }
}
```

Qu-3 Implement the famous Tic-Tac-Toe game using 3x3 grid

CODE:

```
import java.util.Scanner;

public class TicTacToe {
    private char[][] board;
    private char currentPlayerMark;

    public TicTacToe() {
        board = new char[3][3];
        currentPlayerMark = 'X';
        initializeBoard();
    }

    public void initializeBoard() {
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                board[i][j] = '-';
            }
        }
    }

    public void printBoard() {
        System.out.println("-------------");

        for (int i = 0; i < 3; i++) {
            System.out.print("| ");
            for (int j = 0; j < 3; j++) {
                System.out.print(board[i][j] + " | ");
            }
            System.out.println();
            System.out.println("-------------");
        }
    }
```

```
    public boolean isBoardFull() {
        boolean isFull = true;

        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                if (board[i][j] == '-') {
                    isFull = false;
                }
            }
        }

        return isFull;
    }

    public boolean checkForWin() {
        return (checkRowsForWin() || checkColumnsForWin() ||
checkDiagonalsForWin());
    }

    private boolean checkRowsForWin() {
        for (int i = 0; i < 3; i++) {
            if (checkRowCol(board[i][0], board[i][1], board[i][2]) == true) {
                return true;
            }
        }
        return false;
    }

    private boolean checkColumnsForWin() {
        for (int i = 0; i < 3; i++) {
            if (checkRowCol(board[0][i], board[1][i], board[2][i]) == true) {
                return true;
            }
        }
        return false;
    }

    private boolean checkDiagonalsForWin() {
        return ((checkRowCol(board[0][0], board[1][1], board[2][2]) == true) ||
(checkRowCol(board[0][2], board[1][1], board[2][0]) == true));
    }

    private boolean checkRowCol(char c1, char c2, char c3) {
        return ((c1 != '-') && (c1 == c2) && (c2 == c3));
    }

    public void changePlayer() {
        if (currentPlayerMark == 'X') {
            currentPlayerMark = 'O';
        } else {
            currentPlayerMark = 'X';
        }
    }

    public boolean placeMark(int row, int col) {
        if (row >= 0 && row < 3 && col >= 0 && col < 3) {
            if (board[row][col] == '-') {
                board[row][col] = currentPlayerMark;
```

```
                return true;
            }
        }
        return false;
    }

    public static void main(String[] args) {
        TicTacToe game = new TicTacToe();

        Scanner scanner = new Scanner(System.in);

        System.out.println("Tic Tac Toe game");
        System.out.println("----------------");

        int row, col;
        while (true) {
            System.out.println("Current board:");
            game.printBoard();

            System.out.print("Player " + game.currentPlayerMark + ", enter a row
(0-2): ");
            row = scanner.nextInt();

            System.out.print("Player " + game.currentPlayerMark + ", enter a
column (0-2): ");
            col = scanner.nextInt();

            if (game.placeMark(row, col)) {
                if (game.checkForWin()) {
                    System.out.println("Current board:");
                    game.printBoard();
                    System.out.println("Congratulations! Player " +
game.currentPlayerMark + " wins!");
                    break;
                } else if (game.isBoardFull()) {
                    System.out.println("Current board:");
                    game.printBoard();
                    System.out.println("It's a tie!");
                    break;
                } else {
                    game.changePlayer();
                }
            } else {
                System.out.println("Invalid move. Please try again.");
            }
        }

        scanner.close();
    }

}
```

Qu-4 program to TO CHECK IF A TRIANGLE IS EQUILATERAL.

CODE:

```java
import java.util.Scanner;

public class EquilateralTriangleChecker {

    public static void main(String[] args) {
        try (// Create a scanner object to read user input
        Scanner scanner = new Scanner(System.in)) {
          // Read the length of the three sides of the triangle
          System.out.print("Enter the length of side 1: ");
          double side1 = scanner.nextDouble();

          System.out.print("Enter the length of side 2: ");
          double side2 = scanner.nextDouble();

          System.out.print("Enter the length of side 3: ");
          double side3 = scanner.nextDouble();

          // Check if the triangle is equilateral
          if (side1 == side2 && side2 == side3 && side1 == side3) {
              System.out.println("The triangle is equilateral.");
          } else {
              System.out.println("The triangle is not equilateral.");
          }
        }
    }
}
```

Qu-5 program TO CHECK IF A TRIANGLE IS ISOSCELES.
CODE:

```java
import java.util.Scanner;


public class IsoscelesTriangleChecker {

    public static void main(String[] args) {
        try (// Create a scanner object to read user input
        Scanner scanner = new Scanner(System.in)) {
          // Read the length of the three sides of the triangle
          System.out.print("Enter the length of side 1: ");
          double side1 = scanner.nextDouble();

          System.out.print("Enter the length of side 2: ");
          double side2 = scanner.nextDouble();

          System.out.print("Enter the length of side 3: ");
          double side3 = scanner.nextDouble();

          // Check if the triangle is isosceles
          if (side1 == side2 || side1 == side3 || side2 == side3) {
              System.out.println("The triangle is isosceles.");
          } else {
```

```
            System.out.println("The triangle is not isosceles.");
        }
      }
    }
}
```

Qu 6. program to CHECK IF A TRIANGLE IS RIGHT ANGLED.

CODE:

```
import java.util.Scanner;

public class RightAngleTriangleChecker {

    public static void main(String[] args) {
        try (// Create a scanner object to read user input
        Scanner scanner = new Scanner(System.in)) {
          // Read the angle of the three sides of the triangle
          System.out.print("Enter the angle of side 1: ");
          double side1 = scanner.nextDouble();

          System.out.print("Enter the angle of side 2: ");
          double side2 = scanner.nextDouble();

          System.out.print("Enter the angle of side 3: ");
          double side3 = scanner.nextDouble();

          // Check if the triangle is right angled
          if((side1+side2+side3)==180){
            if(side1==90 || side2==90 || side3==90){
                if (side1 == side2 || side2 == side3 || side1 == side3) {
                    System.out.println("The triangle is Right Angled.");
                } else {
                    System.out.println("The triangle is not Right Angled.");
                }
            }
          }
          else {
            System.out.println("Entered wrong angles!!!");
          }
        }
    }
}
```

Qu 7. Program TO MODEL A TETRAHEDRON USING TRIANGLES.

Let us first model a triangular pyramid using triangle structure. A triangular pyramid can be thought of as a combination of four triangles including the base.
enum TypeOfTriangularPyramid{NOTTETRAHEDRON,TETRAHEDRON};

typedef struct TriangularPyramid {
Triangle Side1; Triangle Side2; Triangle Side3; Triangle Base;
Point Centroid;//Centroid of the Pyramid enum TypeOfTriangularPyramid type;
}TriangularPyramid;
Any triangular pyramid can be of two types. Either it is a tetrahedron or it is not. To make the code more readable enum variable ﷺTypeOfTriangularPyramidﷺ is used.
Now a method is written that accepts a TriangularPyramid structure as input, and checks whether the pyramid is a tetrahedron or not. Here is the code.
int isTetrahedron(TriangularPyramid tp) {
if(istriangleequilateral(tp.Base)==EQUILATERAL && istriangleequilateral(tp.Side1) == EQUILATERAL && istriangleequilateral(tp.Side2) == EQUILATERAL && istriangleequilateral(tp.Side3) == EQUILATERAL)
tp.type = TETRAHEDRON; else
tp.type = NOTTETRAHEDRON;
return tp.type;
}
This method returns 1 when the triangular pyramid is a tetrahedron, otherwise it returns 0. Thus it gives more conceptual look to the code. As we know that tetrahedron is used extensively in 3D modeling, we can use this structure to model more complex 3D solids

CODE:

```
import java.util.Scanner;

class Triangle {
    double a, b, c;

    public Triangle(double a, double b, double c) {
        this.a = a;
        this.b = b;
        this.c = c;
    }

    public boolean isEquilateral() {
        return a == b && b == c;
    }
}

class Tetrahedron {
    Triangle side1, side2, side3, base;
    double[] centroid = new double[3];

    public Tetrahedron(Triangle side1, Triangle side2, Triangle side3, Triangle
base) {
        this.side1 = side1;
        this.side2 = side2;
        this.side3 = side3;
        this.base = base;
        calculateCentroid();
    }

    private void calculateCentroid() {
```

```
        centroid[0] = (base.a / 3.0 + side1.a / 6.0 + side2.a / 6.0 + side3.a /
6.0);
        centroid[1] = (base.b / 3.0 + side1.b / 6.0 + side2.b / 6.0 + side3.b /
6.0);
        centroid[2] = (base.c / 3.0 + side1.c / 6.0 + side2.c / 6.0 + side3.c /
6.0);
    }

    public boolean isTetrahedron() {
        return base.isEquilateral() && side1.isEquilateral() &&
side2.isEquilateral() && side3.isEquilateral();
    }
}

// MODEL A TETRAHEDRON USING TRIANGLES
public class TetrahedronModel {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the three sides of the first triangle: ");
        double a1 = scanner.nextDouble();
        double b1 = scanner.nextDouble();
        double c1 = scanner.nextDouble();
        Triangle t1 = new Triangle(a1, b1, c1);

        System.out.println("Enter the three sides of the second triangle: ");
        double a2 = scanner.nextDouble();
        double b2 = scanner.nextDouble();
        double c2 = scanner.nextDouble();
        Triangle t2 = new Triangle(a2, b2, c2);

        System.out.println("Enter the three sides of the third triangle: ");
        double a3 = scanner.nextDouble();
        double b3 = scanner.nextDouble();
        double c3 = scanner.nextDouble();
        Triangle t3 = new Triangle(a3, b3, c3);

        System.out.println("Enter the three sides of the base triangle: ");
        double ab = scanner.nextDouble();
        double bb = scanner.nextDouble();
        double cb = scanner.nextDouble();
        Triangle base = new Triangle(ab, bb, cb);

        Tetrahedron tetrahedron = new Tetrahedron(t1, t2, t3, base);

        if (tetrahedron.isTetrahedron()) {
            System.out.println("The given triangles form a tetrahedron.");
        } else {
            System.out.println("The given triangles do not form a tetrahedron.");
        }

        scanner.close();
    }
}
```