**WEEK-1**

**Data Structures Algorithms**

**Exercise 2: E-commerce Platform Search Function**

**Product.java**

public class Product {

    int productId;

    String productName;

    String category;

    public Product(int productId, String productName, String category) {

        this.productId = productId;

        this.productName = productName;

        this.category = category;

    }

    @Override

    public String toString() {

        return "Product ID: " + productId + ", Name: " + productName + ", Category: " + category;

    }

}

**SearchFunctions.java**

public class SearchFunctions {

    public static int linearSearch(Product[] products, int id) {

        for (int i = 0; i < products.length; i++) {

            if (products[i].productId == id) {

                return i;

            }

        }

        return -1;

    }

    public static int binarySearch(Product[] products, int id) {

        int left = 0, right = products.length - 1;

        while (left <= right) {

            int mid = (left + right) / 2;

            if (products[mid].productId == id)

                return mid;

            else if (products[mid].productId < id)

                left = mid + 1;

            else

                right = mid - 1;

        }

        return -1;

    }

}

**Main.java**

import java.util.Arrays;

import java.util.Comparator;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Product[] products = {

                new Product(105, "Running Shoes", "Footwear"),

                new Product(101, "Gaming Laptop", "Electronics"),

                new Product(103, "Sketchbook", "Stationery"),

                new Product(102, "Smartphone", "Electronics"),

                new Product(106, "Wireless Earbuds", "Accessories"),

                new Product(107, "Office Chair", "Furniture")

        };

        Arrays.sort(products, Comparator.comparingInt(p -> p.productId));

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter Product ID to search: ");

        int searchId = scanner.nextInt();

        int linearResult = SearchFunctions.linearSearch(products, searchId);

        int binaryResult = SearchFunctions.binarySearch(products, searchId);

        if (linearResult >= 0) {

            System.out.println("Linear Search Found: " + products[linearResult]);

        } else {

            System.out.println("Linear Search: Product not found.");

        }

        if (binaryResult >= 0) {

            System.out.println("Binary Search Found: " + products[binaryResult]);

        } else {

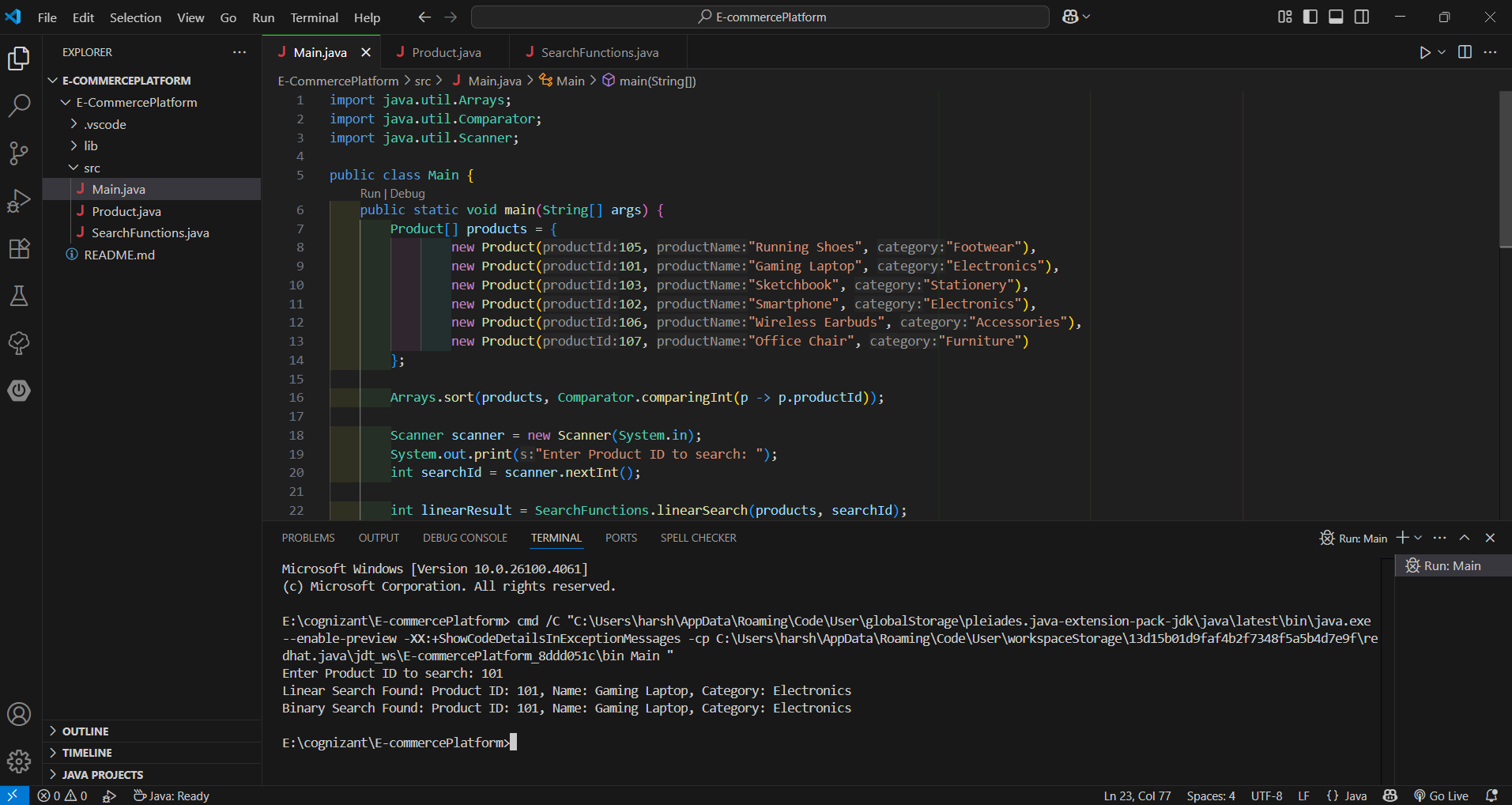
            System.out.println("Binary Search: Product not found.");

        }

        scanner.close();

    }

}



**Exercise 7: Financial Forecasting**

**FinancialForecasting.java**

import java.util.Scanner;

public class FinancialForecasting {

    public static double calculateFutureValue(double presentValue, double rate, int years) {

        if (years == 0) {

            return presentValue;

        }

        return calculateFutureValue(presentValue \* (1 + rate), rate, years - 1);

    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter present value (INR): ");

        double presentValue = scanner.nextDouble();

        System.out.print("Enter annual growth rate (in %): ");

        double rate = scanner.nextDouble() / 100;

        System.out.print("Enter number of years: ");

        int years = scanner.nextInt();

        double futureValue = calculateFutureValue(presentValue, rate, years);

        System.out.printf("Future Value after %d years: INR.%2f%n", years, futureValue);

        scanner.close();

    }

}

