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

**Main.java**

import java.util.\*;

public class Main {

    public static void main(String[] args) {

        SearchEngine engine = new SearchEngine();

        engine.addProduct(new Product("Blue Running Shoes", "Footwear", "Comfortable shoes for running", 1299));

        engine.addProduct(new Product("Adidas Sports Shoes", "Footwear", "High-quality sportswear", 2499));

        engine.addProduct(new Product("Red T-Shirt", "Apparel", "Cotton casual t-shirt", 599));

        engine.addProduct(new Product("Blue Jeans", "Apparel", "Slim-fit jeans for men", 1199));

        Scanner sc = new Scanner(System.in);

        System.out.print("Search (type something): ");

        String input = sc.nextLine();

        List<String> suggestions = engine.getSuggestions(input);

        System.out.println("Autocomplete Suggestions:");

        for (String s : suggestions) {

            System.out.println("  → " + s);

        }

        List<Product> results = engine.search(input);

        System.out.println("\nSearch Results:");

        for (Product p : results) {

            System.out.println("  - " + p);

        }

    }

}

**Product.java**

public class Product {

    String name;

    String category;

    String description;

    double price;

    public Product(String name, String category, String description, double price) {

        this.name = name;

        this.category = category;

        this.description = description;

        this.price = price;

    }

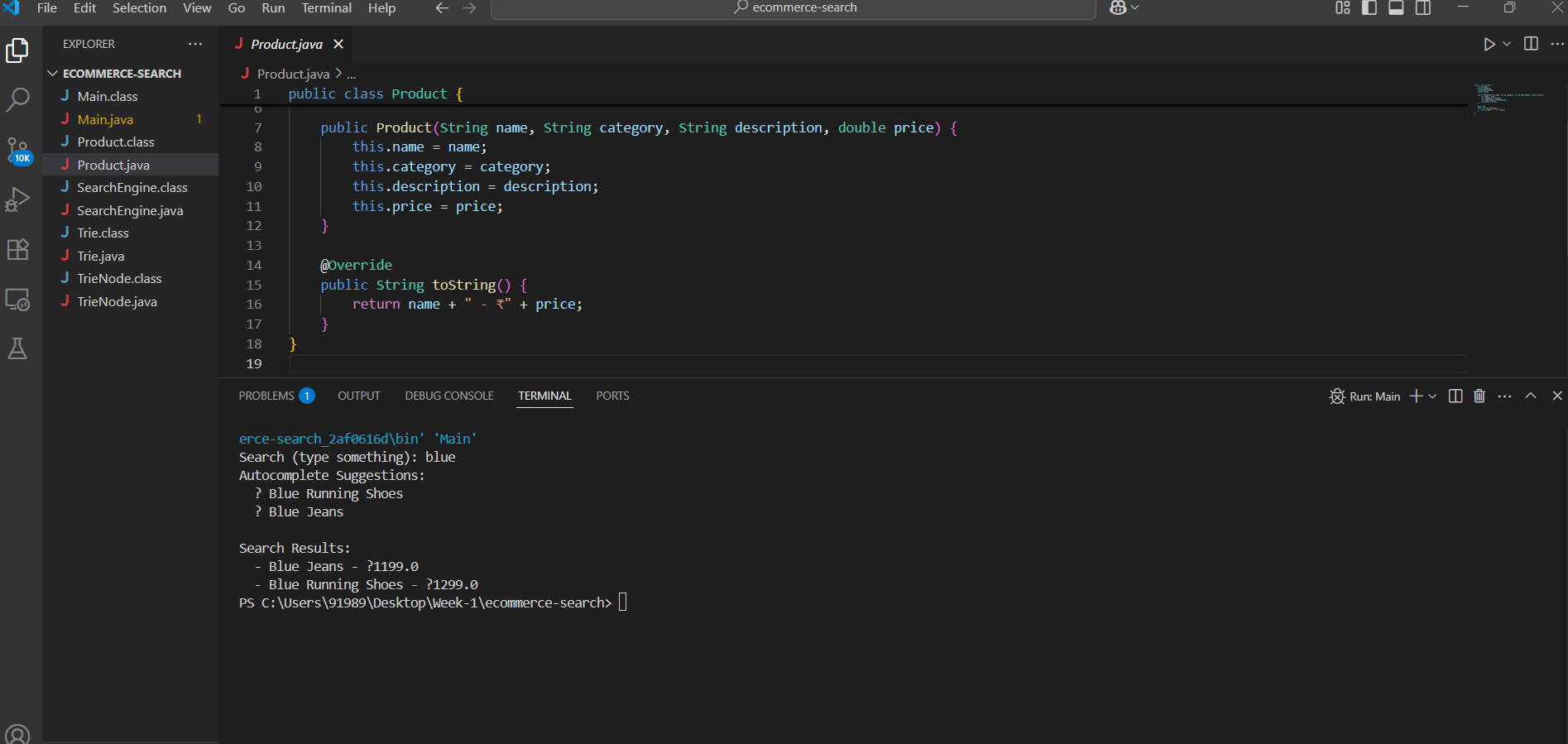
    @Override

    public String toString() {

        return name + " - ₹" + price;

    }

}

**Output:**

**Exercise 7: Financial Forecasting**

**Main.java**

import java.util.\*;

public class Main {

    public static void main(String[] args) {

        int[] revenue = {1200, 1300, 1250, 1400, 1500, 1600, 1550, 1650};

        System.out.println("📊 Past Revenue (Last 8 Days):");

        System.out.println(Arrays.toString(revenue));

        System.out.println("\n📈 3-Day Moving Averages:");

        List<Double> movingAvg = MovingAverage.calculate(revenue, 3);

        for (double avg : movingAvg) {

            System.out.println("  → " + avg);

        }

        double nextPrediction = LinearTrendPredictor.predictNext(revenue);

        System.out.println("\n🔮 Predicted Revenue for Next Day (Using Linear Regression): ₹" + nextPrediction);

    }

}

**MovingAverage.java**

import java.util.\*;

public class MovingAverage {

    public static List<Double> calculate(int[] data, int k) {

        List<Double> result = new ArrayList<>();

        double sum = 0;

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

            sum += data[i];

            if (i >= k) {

                sum -= data[i - k];

            }

            if (i >= k - 1) {

                result.add(Math.round((sum / k) \* 100.0) / 100.0); // round to 2 decimal places

            }

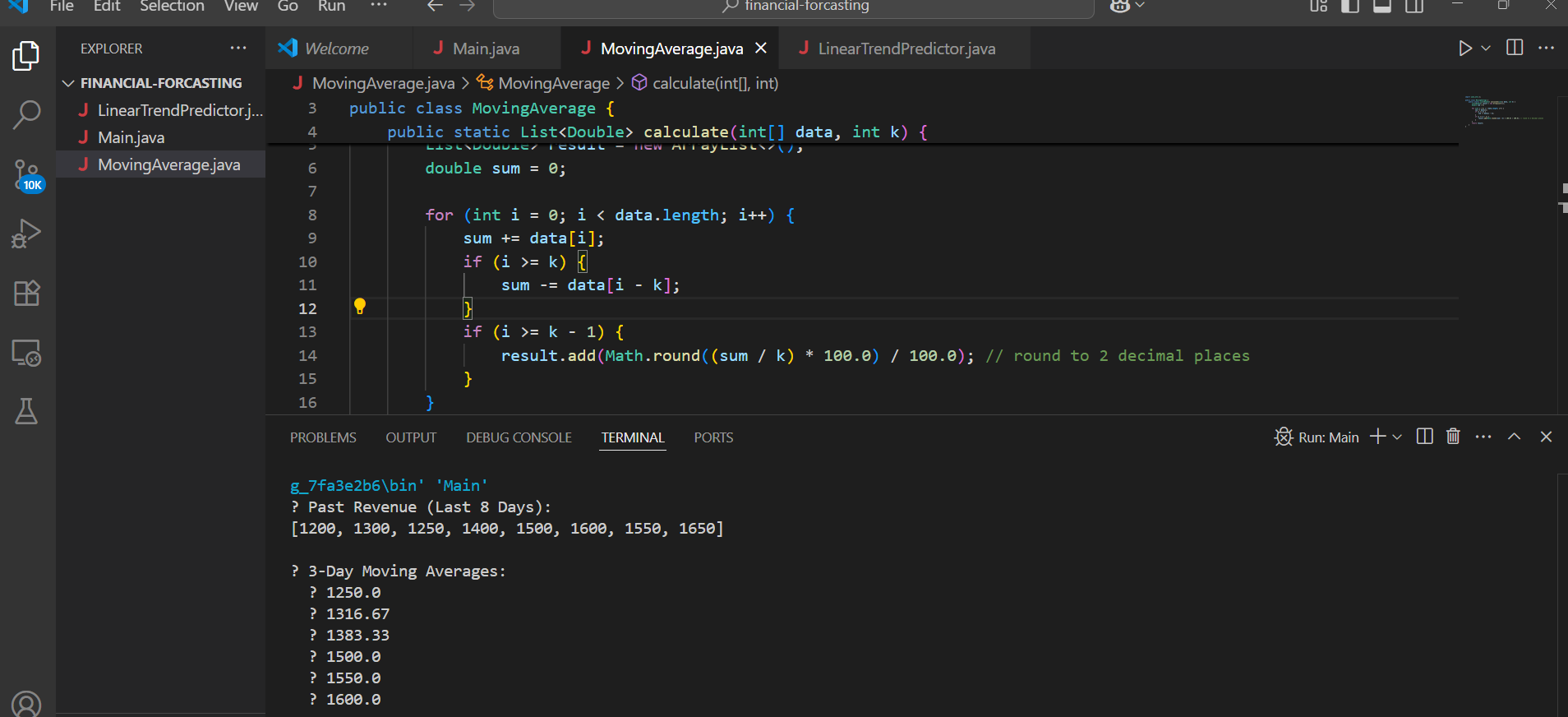
        }

        return result;

    }

}

**Output:**

****