**1.E-commerce Platform Search Function**

**Main.java**

package search;

import java.util.Arrays;

import java.util.Comparator;

public class Main {

    public static void main(String[] args) {

        Product[] products = {

            new Product(3, "Laptop", "Electronics"),

            new Product(1, "Shirt", "Clothing"),

            new Product(5, "Book", "Education"),

            new Product(2, "Phone", "Electronics"),

            new Product(4, "Shoes", "Footwear")

        };

System.out.println("Linear Search for ID 4:");

        Product foundLinear = SearchAlgorithms.linearSearch(products, 4);

        System.out.println(foundLinear != null ? foundLinear : "Not found");

System.out.println("\nSorting products by productId for Binary Search...");

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

System.out.println("Binary Search for ID 4:");

        Product foundBinary = SearchAlgorithms.binarySearch(products, 4);

        System.out.println(foundBinary != null ? foundBinary : "Not found");

    }}

**Product.java**

package search;

public class Product {

    public int productId;

    public String productName;

    public String category;

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

        this.productId = productId;

        this.productName = productName;

        this.category = category;

 }

public String toString() {

        return "[" + productId + "] " + productName + " - " + category;

    }}

**Searchalgorithm.java**

package search;

public class SearchAlgorithms {

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

        for (Product p : products) {

            if (p.productId == id) {

                return p;

            } }

        return null;

    }

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

        int low = 0;

        int high = products.length - 1;

while (low <= high) {

            int mid = (low + high) / 2;

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

                return products[mid];

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

                low = mid + 1;

            } else {

                high = mid - 1;

            }

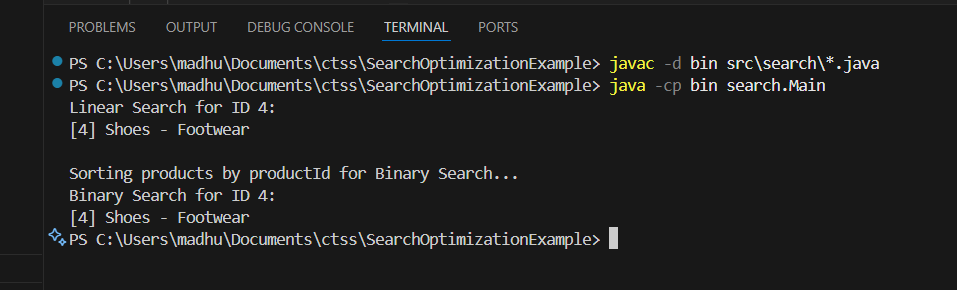
        }

        return null;

    }

}

**Output:**



**2.Financial Forecasting**

public class FinancialForecast {

public static double forecast(int years, double currentValue, double rate) {

        if (years == 0) {

            return currentValue;

        } else {

            return forecast(years - 1, currentValue, rate) \* (1 + rate);

        }

    }

public static void main(String[] args) {

        double currentValue = 10000;

        double rate = 0.05;

        int years = 5;

double result = forecast(years, currentValue, rate);

        System.out.printf("Predicted value after %d years: ₹%.2f\n", years, result);

    }

}

