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

**//code:**

namespace EcommerceSearch

{

    public class Product

    {

        public int ProductId { get; set; }

        public string ProductName { get; set; }

        public string Category { get; set; }

        public Product(int id, string name, string category)

        {

            ProductId = id;

            ProductName = name;

            Category = category;

        }

    }

}

using System;

using System.Collections.Generic;

namespace EcommerceSearch

{

    class Program

    {

        static void Main(string[] args)

        {

            Product[] products = new Product[]

            {

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

                new Product(102, "Shoes", "Fashion"),

                new Product(103, "Watch", "Accessories"),

                new Product(104, "Phone", "Electronics"),

                new Product(105, "Book", "Books")

            };

            Console.WriteLine("🔍 Linear Search for 'Phone':");

            int indexLinear = LinearSearch(products, "Phone");

            Console.WriteLine(indexLinear != -1 ? $"Product found at index {indexLinear}" : "Product not found");

            Array.Sort(products, (p1, p2) => p1.ProductName.CompareTo(p2.ProductName));

            Console.WriteLine("\n🔍 Binary Search for 'Phone':");

            int indexBinary = BinarySearch(products, "Phone");

            Console.WriteLine(indexBinary != -1 ? $"Product found at index {indexBinary}" : "Product not found");

        }

        static int LinearSearch(Product[] products, string name)

        {

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

            {

                if (products[i].ProductName.Equals(name, StringComparison.OrdinalIgnoreCase))

                    return i;

            }

            return -1;

        }

        static int BinarySearch(Product[] products, string name)

        {

            int left = 0;

            int right = products.Length - 1;

            while (left <= right)

            {

                int mid = (left + right) / 2;

                int comparison = string.Compare(products[mid].ProductName, name, StringComparison.OrdinalIgnoreCase);

                if (comparison == 0)

                    return mid;

                else if (comparison < 0)

                    left = mid + 1;

                else

                    right = mid - 1;

            }

            return -1;

        }

    }

}

**//output:**

****