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

**Program.cs**

**using System;**

**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;**

**}**

**public override string ToString()**

**{**

**return $"{ProductId} - {ProductName} ({Category})";**

**}**

**}**

**class Program**

**{**

**static void Main()**

**{**

**Console.Write("Enter number of products: ");**

**int n = Convert.ToInt32(Console.ReadLine());**

**Product[] products = new Product[n];**

**Console.WriteLine("Enter product details:");**

**for (int i = 0; i < n; i++)**

**{**

**Console.Write($"Product {i + 1} ID: ");**

**int id = Convert.ToInt32(Console.ReadLine());**

**Console.Write($"Product {i + 1} Name: ");**

**string name = Console.ReadLine();**

**Console.Write($"Product {i + 1} Category: ");**

**string category = Console.ReadLine();**

**products[i] = new Product(id, name, category);**

**}**

**Array.Sort(products, (a, b) => a.ProductId.CompareTo(b.ProductId));**

**Console.Write("\nEnter Product ID to search: ");**

**int targetId = Convert.ToInt32(Console.ReadLine());**

**Console.WriteLine("\nLinear Search Result:");**

**LinearSearch(products, targetId);**

**Console.WriteLine("\nBinary Search Result:");**

**BinarySearch(products, targetId);**

**}**

**static void LinearSearch(Product[] products, int targetId)**

**{**

**foreach (var product in products)**

**{**

**if (product.ProductId == targetId)**

**{**

**Console.WriteLine("Product found: " + product);**

**return;**

**}**

**}**

**Console.WriteLine("Product not found.");**

**}**

**static void BinarySearch(Product[] products, int targetId)**

**{**

**int low = 0;**

**int high = products.Length - 1;**

**while (low <= high)**

**{**

**int mid = (low + high) / 2;**

**if (products[mid].ProductId == targetId)**

**{**

**Console.WriteLine("Product found: " + products[mid]);**

**return;**

**}**

**else if (products[mid].ProductId < targetId)**

**low = mid + 1;**

**else**

**high = mid - 1;**

**}**

**Console.WriteLine("Product not found.");**

**}**

**}**

**EcommerceSearchApp.csproj**

**<Project Sdk="Microsoft.NET.Sdk">**

**<PropertyGroup>**

**<OutputType>Exe</OutputType>**

**<TargetFramework>net9.0</TargetFramework>**

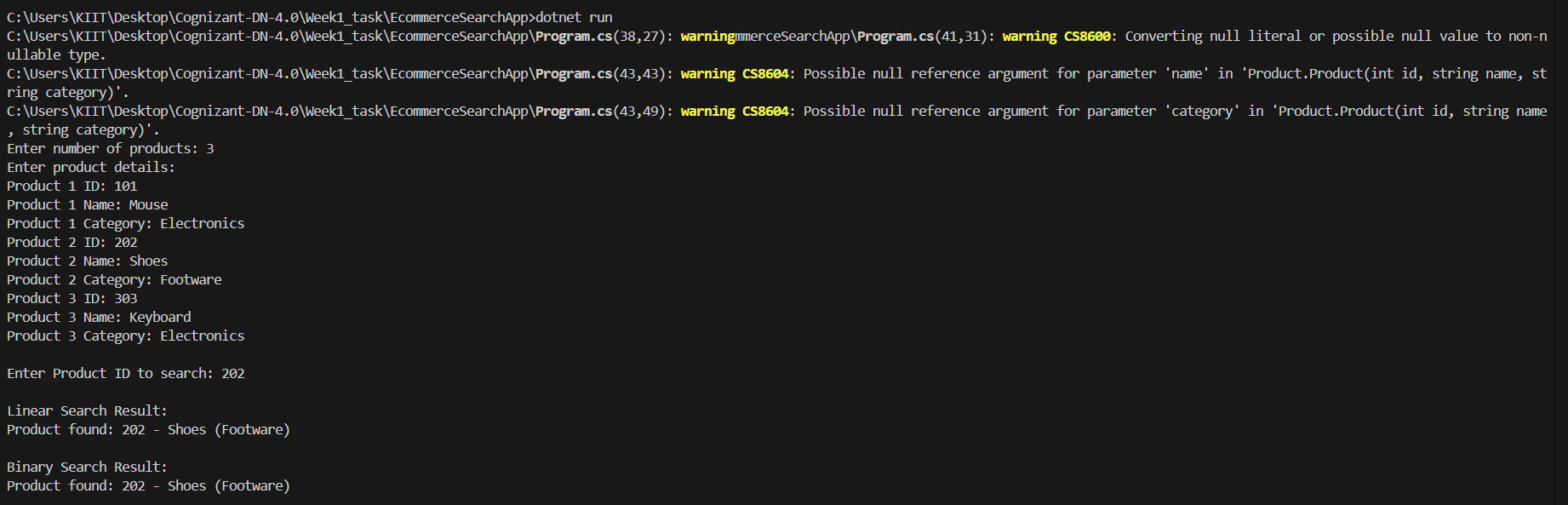
**<ImplicitUsings>enable</ImplicitUsings>**

**<Nullable>enable</Nullable>**

**</PropertyGroup>**

**</Project>**

**Output:-**

****

**Exercise 7: Financial Forecasting**

**Program.cs**

**using System;**

**class FinancialForecast**

**{**

**static double PredictFutureValue(double amount, double growthRate, int years)**

**{**

**for (int i = 0; i < years; i++)**

**{**

**amount \*= (1 + growthRate);**

**}**

**return amount;**

**}**

**static void Main(string[] args)**

**{**

**Console.WriteLine("==== Financial Forecasting Tool ====\n");**

**Console.Write("Enter initial investment amount: ");**

**double initialAmount = Convert.ToDouble(Console.ReadLine());**

**Console.Write("Enter annual growth rate (e.g., 0.10 for 10%): ");**

**double growthRate = Convert.ToDouble(Console.ReadLine());**

**Console.Write("Enter number of years to forecast: ");**

**int years = Convert.ToInt32(Console.ReadLine());**

**double futureValue = PredictFutureValue(initialAmount, growthRate, years);**

**Console.WriteLine($"\n📈 Forecasted Value after {years} years: ₹{futureValue:F2}");**

**}**

**}**

**Financial Forecasting.csproj**

**<Project Sdk="Microsoft.NET.Sdk">**

**<PropertyGroup>**

**<OutputType>Exe</OutputType>**

**<TargetFramework>net9.0</TargetFramework>**

**<RootNamespace>Financial\_Forecasting</RootNamespace>**

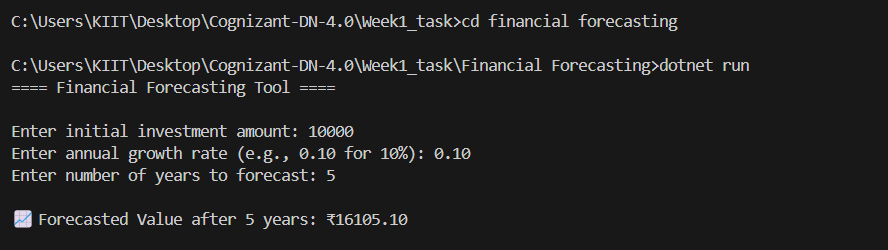
**<ImplicitUsings>enable</ImplicitUsings>**

**<Nullable>enable</Nullable>**

**</PropertyGroup>**

**</Project>**

**Output:-**

****