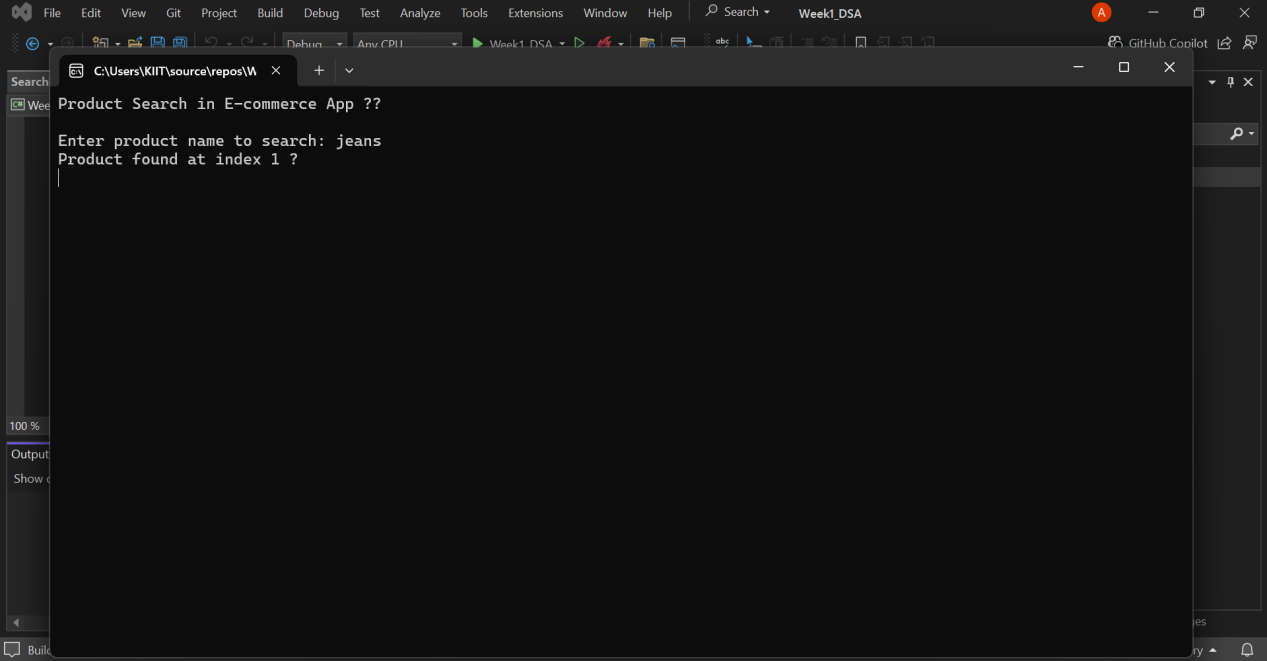
**Exercise 2: E-commerce Platform Search**

**FunctionScreenshots of OUTPUT:**

****

**CODES :**

**//program.cs**

using System;

namespace EcommerceSearch

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Product Search in E-commerce App 🛒\n");

string[] products = { "shoes", "shirt", "jeans", "watch", "bag" };

Array.Sort(products); // search works better when sorted

Console.Write("Enter product name to search: ");

string input = Console.ReadLine().ToLower(); // lowercase for uniformity

int index = SearchHelper.BinarySearch(products, input);

if (index != -1)

Console.WriteLine($"Product found at index {index} ✅");

else

Console.WriteLine("sorry, product not found ❌");

Console.ReadLine(); // pause

}

}

}

**//SearchHelper.cs**

using System;

namespace EcommerceSearch

{

public class SearchHelper

{

public static int BinarySearch(string[] arr, string target)

{

int low = 0, high = arr.Length - 1;

while (low <= high)

{

int mid = (low + high) / 2;

int res = string.Compare(arr[mid], target);

if (res == 0)

return mid;

else if (res < 0)

low = mid + 1;

else

high = mid - 1;

}

return -1; // not found

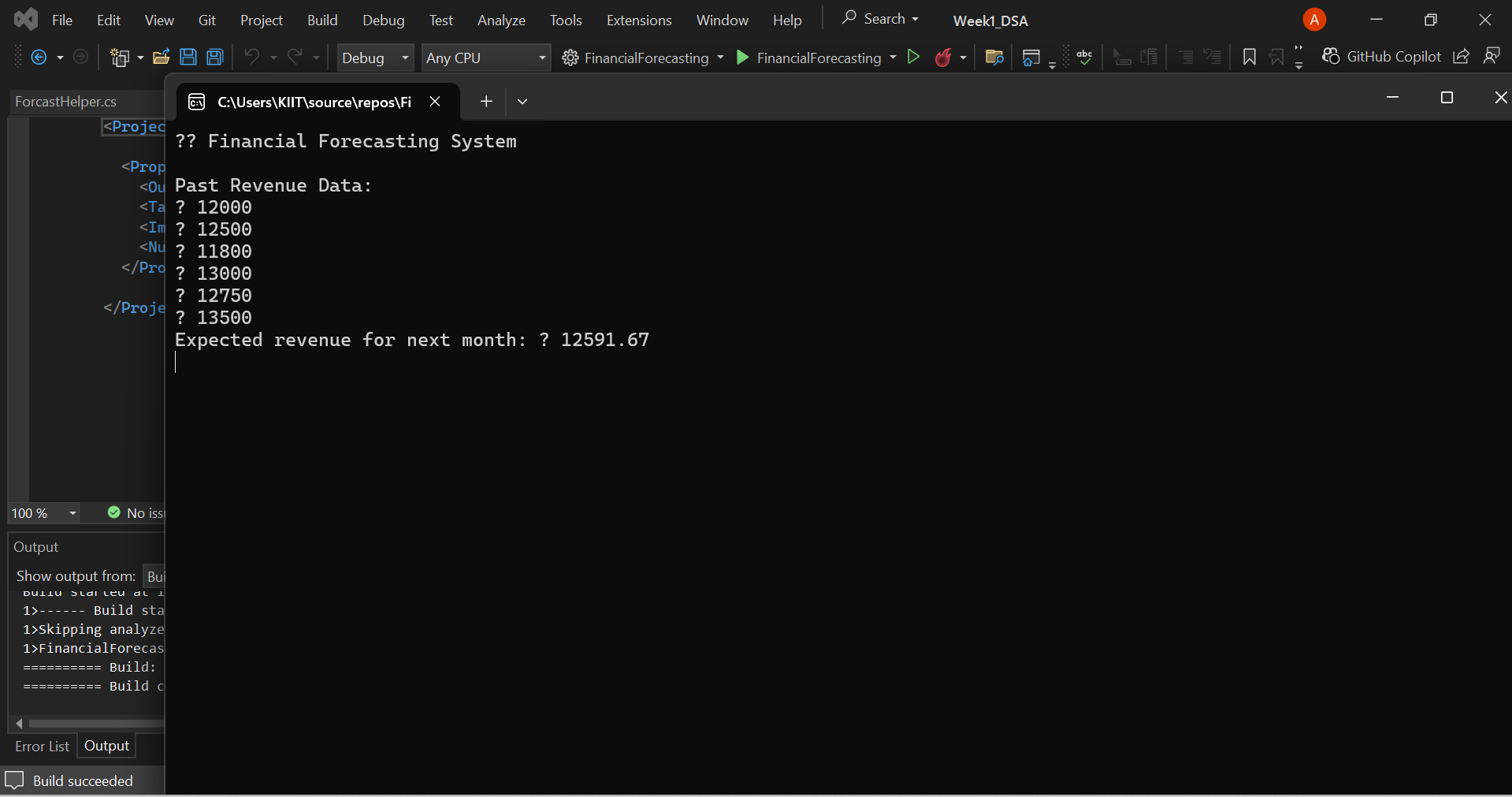
}

}

}

**Exercise 7:Financial Forecasting**

**Screenshots of OUTPUT:**



**CODES :**

**//program.cs**

using System;

using static System.Net.Mime.MediaTypeNames;

namespace FinancialForecasting

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("📊 Financial Forecasting System\n");

// dummy revenue data for last 6 months

double[] pastRevenue = { 12000, 12500, 11800, 13000, 12750, 13500 };

Console.WriteLine("Past Revenue Data:");

foreach (var r in pastRevenue)

{

Console.WriteLine("₹ " + r);

}

// forecasting next month based on average

double forecast = ForecastHelper.GetNextMonthForecast(pastRevenue);

Console.WriteLine($"Expected revenue for next month: ₹ { forecast}");

Console.ReadLine(); // pause to view output

}

}

}

**//ForecastHelper.cs**

using System;

namespace FinancialForecasting

{

public class ForecastHelper

{

public static double GetNextMonthForecast(double[] revenues)

{

// avg logic — basic but works

double sum = 0;

foreach (var val in revenues)

{

sum += val;

}

return Math.Round(sum / revenues.Length, 2); // round to 2 decimal places

}

}

}