EF Core 8.0 Guided Hands-On Exercises

Lab 5: Retrieving Data from the Database

CODE:

using System;

using System.Threading.Tasks;

using System.Collections.Generic;

using Microsoft.EntityFrameworkCore;

namespace RetailStoreConsoleApp

{

// Represents a product in the store

public class Product

{

public int Id { get; set; } // Unique ID for the product

public string Name { get; set; } // Name of the product

public decimal Price { get; set; } // Price of the product in INR

}

// The application's database context

public class RetailDbContext : DbContext

{

// Table of products

public DbSet<Product> Products { get; set; }

// Configure the database connection

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlServer(

@"Server=(localdb)\MSSQLLocalDB;Database=RetailStoreDb\_New;Trusted\_Connection=True;"

);

}

}

class Program

{

static async Task Main(string[] args)

{

Console.WriteLine("=== Retail Store Dashboard ===\n");

using (var context = new RetailDbContext())

{

// 1. Get and display all products

var allProducts = await context.Products.ToListAsync();

Console.WriteLine("All Available Products:");

foreach (var product in allProducts)

{

Console.WriteLine($"- {product.Name} | Price: ₹{product.Price}");

}

// 2. Find a specific product by ID (e.g., ID = 1)

var productById = await context.Products.FindAsync(1);

Console.WriteLine($"\nProduct with ID 1: {productById?.Name ?? "Not Found"}");

// 3. Find the first product with price > ₹5000

var expensiveProduct = await context.Products.FirstOrDefaultAsync(p => p.Price > 5000);

Console.WriteLine($"First Expensive Product (> ₹5000): {expensiveProduct?.Name ?? "None"}");

}

Console.WriteLine("\nPress any key to exit...");

Console.ReadKey();

}

}

}

OUTPUT:

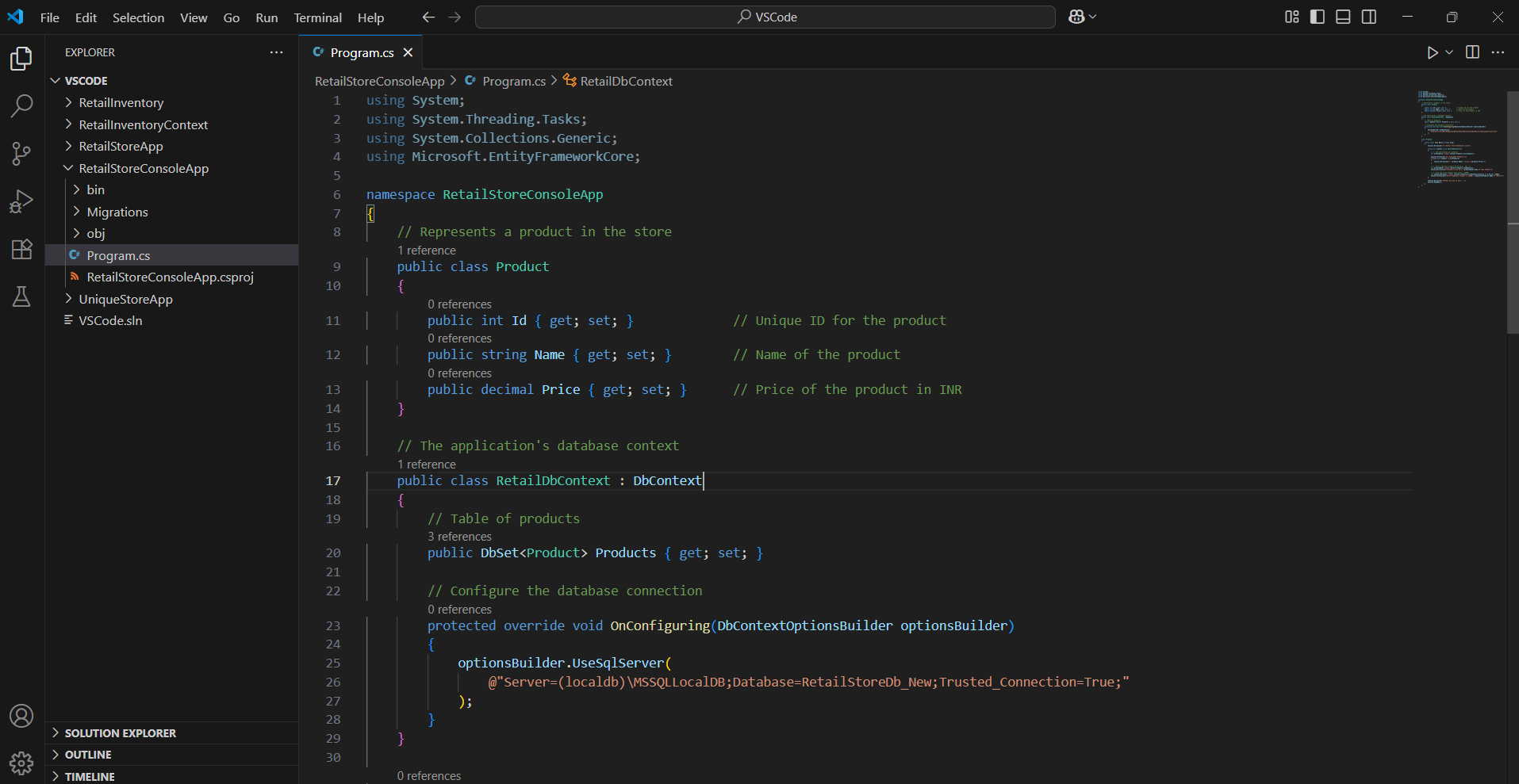
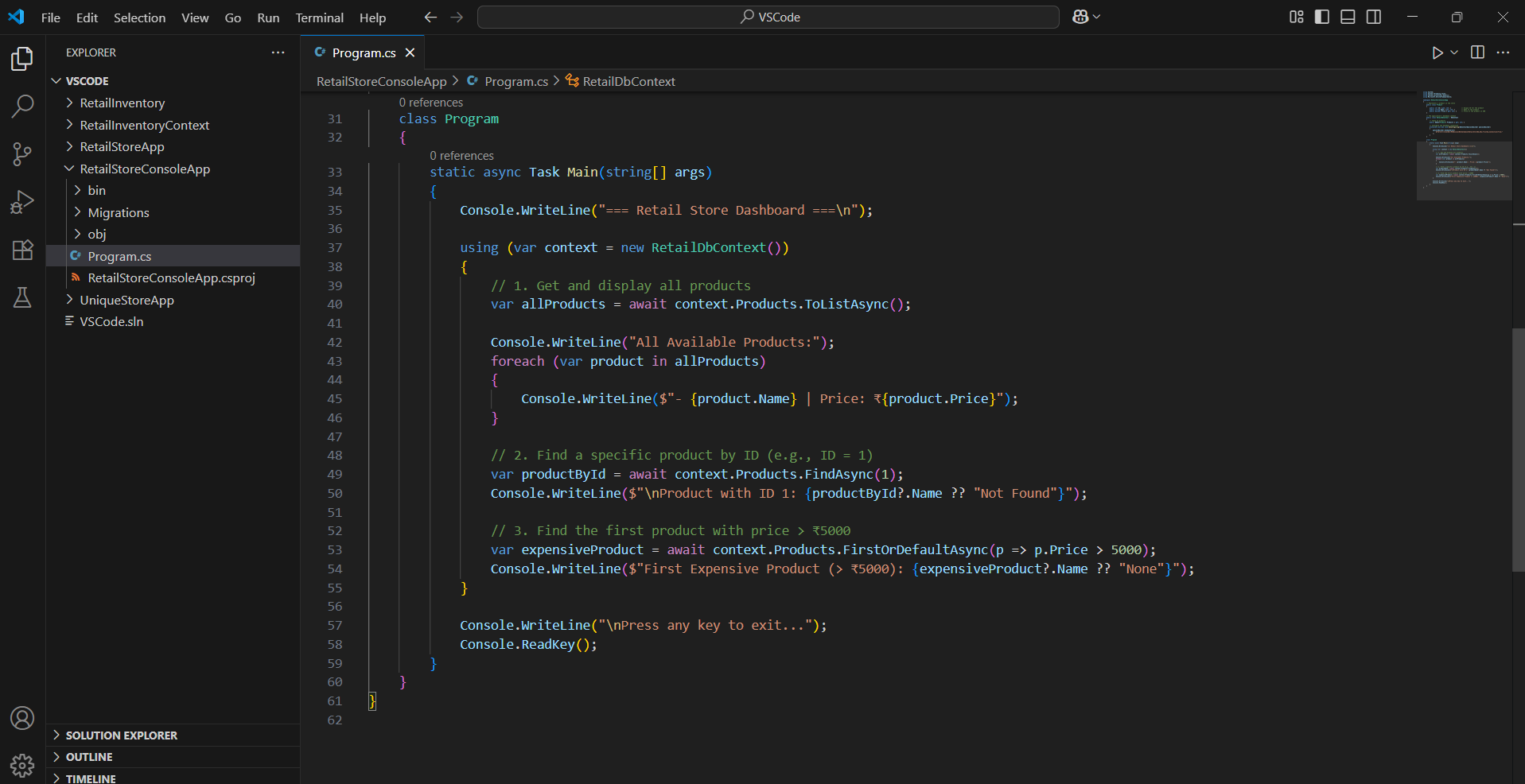
=== Retail Store Dashboard ===

All Available Products:

Product with ID 1: Not Found

First Expensive Product (> ₹5000): None

Press any key to exit...

ScreenShots:

