EF Core 8.0 Hands-On Labs 1–5

**Student:** Arya Vats  
 **Date:** July 4, 2025

**Superset-ID** : 6358118

## **1. Project Setup**

**Commands run in terminal:**

****dotnet new console -n RetailInventory

cd RetailInventory

dotnet add package Microsoft.EntityFrameworkCore.InMemory

*Explanation:* Created a new console app and added the in-memory EF Core provider to bypass external database setup.

## **2. Models and DbContext**

### **2.1 Models.cs**

****using System.Collections.Generic;

public class Category

{

public int Id { get; set; }

public string Name { get; set; }

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

}

public class Product

{

public int Id { get; set; }

public string Name { get; set; }

public decimal Price { get; set; }

public int CategoryId { get; set; }

public Category Category { get; set; }

}

### **2.2 AppDbContext.cs**

****using Microsoft.EntityFrameworkCore;

public class AppDbContext : DbContext

{

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

public DbSet<Category> Categories { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseInMemoryDatabase("RetailInventory");

}

}

*Explanation:* We used the in-memory database to simplify labs and avoid connection issues.

## **3. Combined Program.cs for Labs 4 & 5**

**Program.cs**

****using System;

using System.Threading.Tasks;

using Microsoft.EntityFrameworkCore;

class Program

{

static async Task Main(string[] args)

{

using var context = new AppDbContext();

// Ensure the in-memory database is created

context.Database.EnsureCreated();

// --- Lab 4: Seed Data ---

var electronics = new Category { Name = "Electronics" };

var groceries = new Category { Name = "Groceries" };

await context.Categories.AddRangeAsync(electronics, groceries);

var laptop = new Product { Name = "Laptop", Price = 75\_000, Category = electronics };

var riceBag = new Product { Name = "Rice Bag", Price = 1\_200, Category = groceries };

await context.Products.AddRangeAsync(laptop, riceBag);

await context.SaveChangesAsync();

Console.WriteLine("✅ Seed complete.\n");

// --- Lab 5: Retrieve Data ---

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

Console.WriteLine("📦 All Products:");

foreach (var p in products)

Console.WriteLine($" • {p.Name} - ₹{p.Price}");

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

var expensive = await context.Products.FirstOrDefaultAsync(p => p.Price > 50\_000);

Console.WriteLine($"\n🔍 Found by ID (1): {found?.Name}");

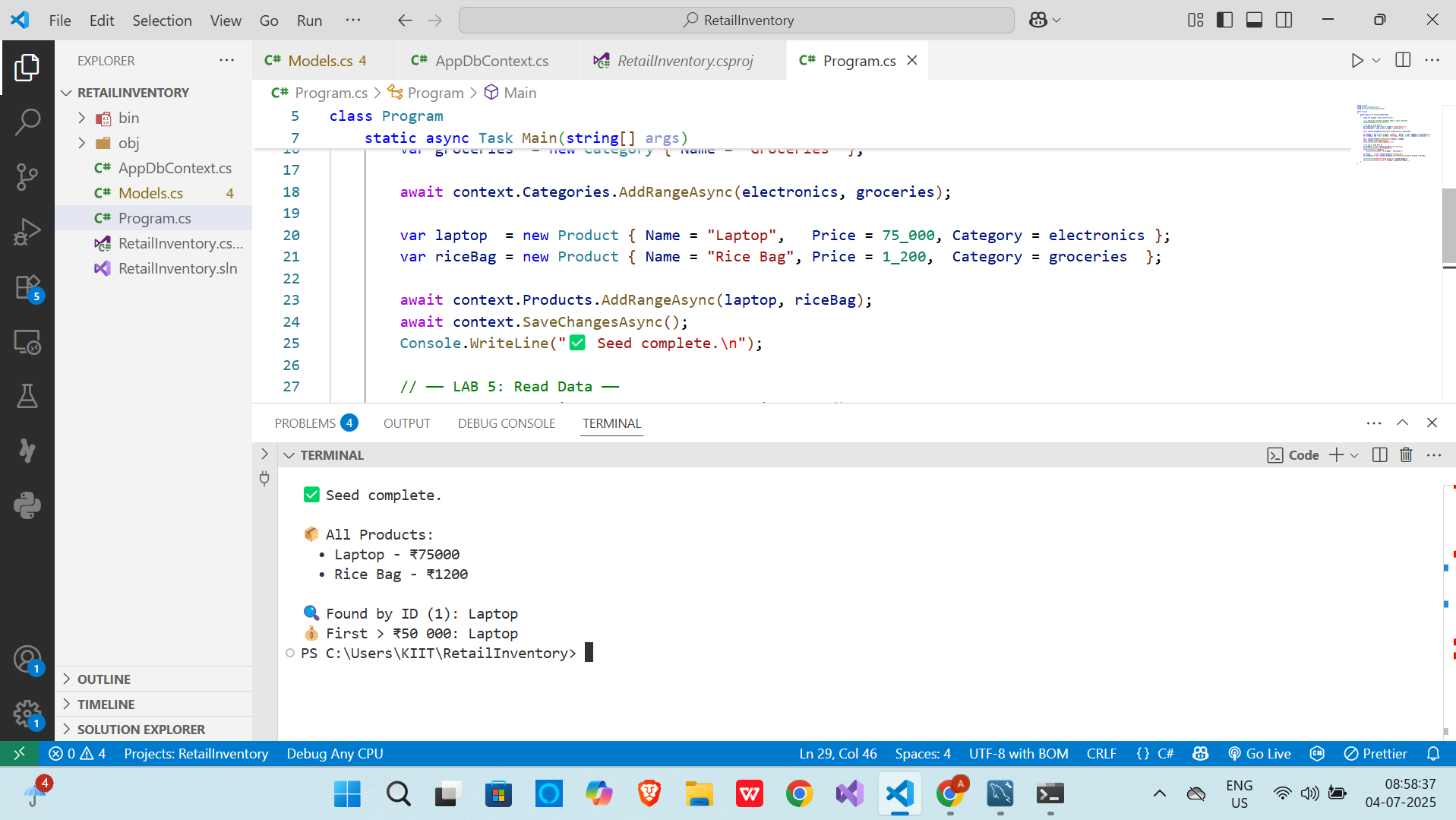
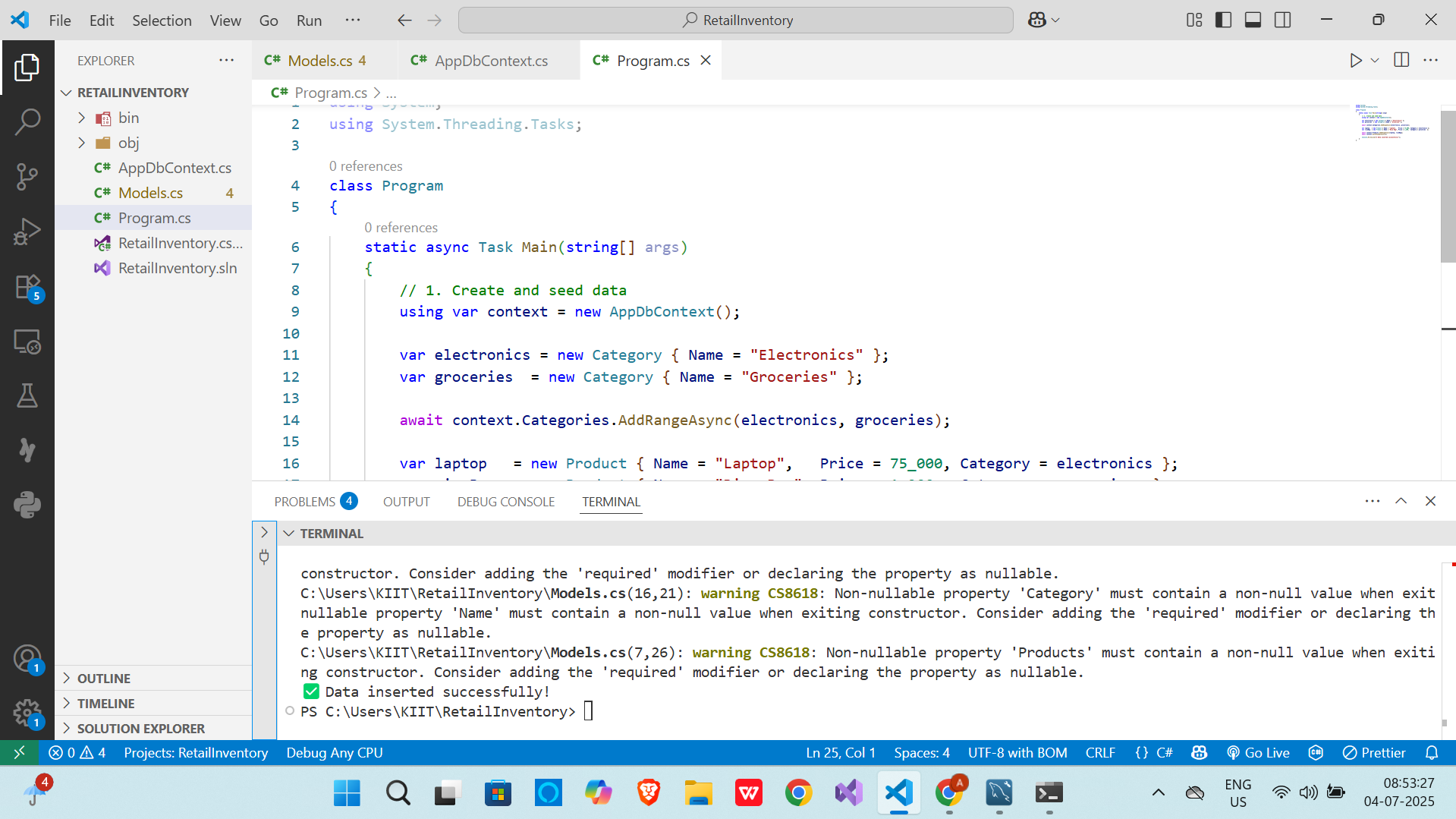
Console.WriteLine($"💰 First > ₹50 000: {expensive?.Name}");

}

}



## **4.Console Output:**



*End of Report.*