**WEEK 3 ASSIGNMENT**

using System;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.EntityFrameworkCore;

using RetailInventory.Data;

using RetailInventory.Models;

class Program

{

    static async Task Main(string[] args)

    {

        Console.WriteLine("Retail Inventory System");

        using var context = new AppDbContext();

        await context.Database.EnsureCreatedAsync();

        // Lab 4: Seed initial data if not already present

        if (!context.Categories.Any())

        {

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

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

            var product1 = new Product { Name = "Laptop", Price = 75000, StockQuantity = 10, Category = electronics };

            var product2 = new Product { Name = "Rice Bag", Price = 1200, StockQuantity = 50, Category = groceries };

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

            await context.Products.AddRangeAsync(product1, product2);

            await context.SaveChangesAsync();

            Console.WriteLine("Initial data inserted.\n");

        }

        // Lab 5: Retrieve Data

        Console.WriteLine("=== All Products ===");

        var products = await context.Products.Include(p => p.Category).ToListAsync();

        foreach (var p in products)

        {

            Console.WriteLine($"{p.Name} - ₹{p.Price} - Stock: {p.StockQuantity} - Category: {p.Category.Name}");

        }

        Console.WriteLine("\n=== Find by ID ===");

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

        Console.WriteLine(productById != null ? $"Found: {productById.Name}" : "Not Found");

        Console.WriteLine("\n=== First Product > ₹50,000 ===");

        var expensiveProduct = await context.Products.FirstOrDefaultAsync(p => p.Price > 50000);

        Console.WriteLine(expensiveProduct != null ? $"Expensive: {expensiveProduct.Name}" : "None found");

        // Lab 6: Update and Delete

        Console.WriteLine("\n=== Update Product ===");

        var laptop = await context.Products.FirstOrDefaultAsync(p => p.Name == "Laptop");

        if (laptop != null)

        {

            laptop.Price = 70000;

            await context.SaveChangesAsync();

            Console.WriteLine("Laptop price updated.");

        }

        Console.WriteLine("\n=== Delete Product ===");

        var riceBag = await context.Products.FirstOrDefaultAsync(p => p.Name == "Rice Bag");

        if (riceBag != null)

        {

            context.Products.Remove(riceBag);

            await context.SaveChangesAsync();

            Console.WriteLine("Rice Bag deleted.");

        }

        // Lab 7: LINQ Queries

        Console.WriteLine("\n=== Filtered & Sorted Products (Price > 1000) ===");

        var filtered = await context.Products

            .Where(p => p.Price > 1000)

            .OrderByDescending(p => p.Price)

            .ToListAsync();

        foreach (var p in filtered)

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

        Console.WriteLine("\n=== Project to DTO (Name, Price) ===");

        var productDTOs = await context.Products

            .Select(p => new { p.Name, p.Price })

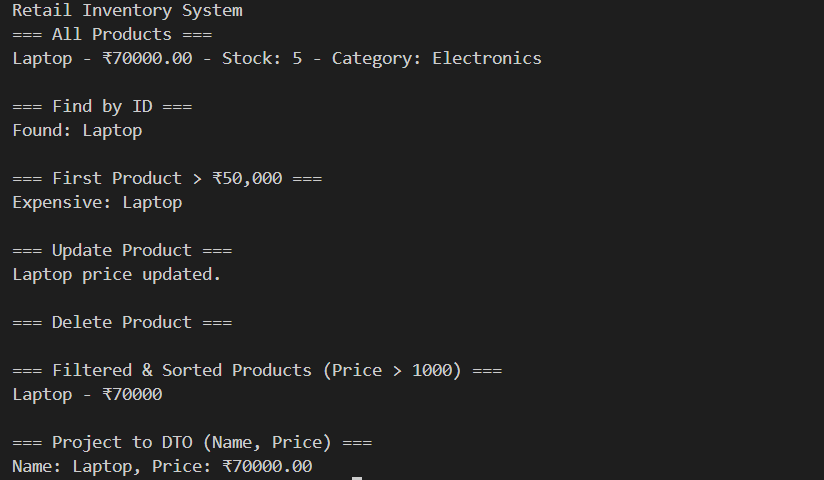
            .ToListAsync();

        foreach (var dto in productDTOs)

            Console.WriteLine($"Name: {dto.Name}, Price: ₹{dto.Price}");

    }

}

****