**LAB2: Setting Up the Database Context for a Retail Store**

**Product.cs:**

namespace RetailStore.Models

{

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; }

}

}

**Category.cs:**

namespace RetailStore.Models

{

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; }

}

}

**AppDbContext.cs:**

using Microsoft.EntityFrameworkCore;

using RetailStore.Models;

public class AppDbContext : DbContext

{

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

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

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

// SQLite database file path (creates RetailStore.db in project folder)

optionsBuilder.UseSqlite("Data Source=RetailStore.db");

}

}

**Program.cs:**

using System;

using System.Linq;

using Microsoft.EntityFrameworkCore;

using RetailStore.Models;

class Program

{

static void Main()

{

using var context = new AppDbContext();

context.Database.EnsureCreated();

if (!context.Categories.Any())

{

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

category.Products.Add(new Product { Name = "Laptop", Price = 1200m });

category.Products.Add(new Product { Name = "Smartphone", Price = 800m });

context.Categories.Add(category);

context.SaveChanges();

}

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

foreach (var product in products)

{

Console.WriteLine($"{product.Name} ({product.Category?.Name}) - ${product.Price}");

}

}

}

**Output:**

