Lab 1: Understanding ORM with a Retail Inventory System

Required NuGet Packages:

// - Microsoft.EntityFrameworkCore

// - Microsoft.EntityFrameworkCore.Sqlite

using System;

using System.Collections.Generic;

using System.Linq;

using Microsoft.EntityFrameworkCore;

namespace RetailInventoryApp

{

// Model: Category

public class Category

{

public int CategoryId { get; set; }

public string Name { get; set; }

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

}

// Model: Product

public class Product

{

public int ProductId { get; set; }

public string Name { get; set; }

public int Quantity { get; set; }

public decimal Price { get; set; }

// Foreign key

public int CategoryId { get; set; }

// Navigation property

public Category Category { get; set; }

}

// DB Context

public class RetailContext : DbContext

{

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

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

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

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

}

}

// Main Program

class Program

{

static void Main()

{

using (var context = new RetailContext())

{

context.Database.EnsureCreated(); // Create database if not exists

// CREATE Category

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

context.Categories.Add(category);

context.SaveChanges();

// CREATE Product

var product = new Product

{

Name = "Smartphone",

Quantity = 50,

Price = 29999,

CategoryId = category.CategoryId

};

context.Products.Add(product);

context.SaveChanges();

// READ Products

Console.WriteLine("Products:");

foreach (var p in context.Products.Include(p => p.Category))

{

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

}

// UPDATE Product

product.Quantity += 10;

context.SaveChanges();

// DELETE Product

context.Products.Remove(product);

context.SaveChanges();

// Final output

Console.WriteLine("\nAfter Deletion:");

foreach (var p in context.Products.Include(p => p.Category))

{

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

}

}

}

}

}

Out put

