using System;

using System.Collections.Generic;

using System.Linq;

// Lab 3

public class Category

{

public int Id { get; set; }

public string Name { get; set; }

public List<Product> Products { get; set; } = new List<Product>();

}

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; }

}

public class Program

{

public static void Main()

{

// Simulated database

var categories = new List<Category>();

var products = new List<Product>();

// Lab 4: Insert Initial Data

var electronics = new Category { Id = 1, Name = "Electronics" };

var groceries = new Category { Id = 2, Name = "Groceries" };

categories.Add(electronics);

categories.Add(groceries);

var product1 = new Product { Id = 1, Name = "Laptop", Price = 75000, Category = electronics, CategoryId = electronics.Id };

var product2 = new Product { Id = 2, Name = "Rice Bag", Price = 1200, Category = groceries, CategoryId = groceries.Id };

products.Add(product1);

products.Add(product2);

// Lab 5: Retrieve

Console.WriteLine("All Products:");

foreach (var p in products)

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

var found = products.FirstOrDefault(p => p.Id == 1);

Console.WriteLine($"\nFound by ID: {found?.Name}");

var expensive = products.FirstOrDefault(p => p.Price > 50000);

Console.WriteLine($"Expensive Product: {expensive?.Name}");

// Lab 6: Update + Delete

var toUpdate = products.FirstOrDefault(p => p.Name == "Laptop");

if (toUpdate != null)

{

toUpdate.Price = 70000;

Console.WriteLine("Updated Laptop Price.");

}

var toDelete = products.FirstOrDefault(p => p.Name == "Rice Bag");

if (toDelete != null)

{

products.Remove(toDelete);

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

}

// Lab 7: LINQ Queries

Console.WriteLine("\nFiltered & Sorted Products:");

var filtered = products.Where(p => p.Price > 1000).OrderByDescending(p => p.Price);

foreach (var p in filtered)

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

Console.WriteLine("\nProjected DTOs:");

var productDTOs = products.Select(p => new { p.Name, p.Price });

foreach (var dto in productDTOs)

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

}

}

