EF Core 8.0 Guided Hands-On Exercises

Lab 2: Setting Up the Database Context for a Retail Store

CODE:

using System;

using System.Collections.Generic;

using Microsoft.EntityFrameworkCore;

namespace InventoryManagementSystem

{

    // Represents a single item that can be sold

    public class StoreItem

    {

        public int StoreItemId { get; set; }                            // Primary Key

        public required string ItemName { get; set; }                   // Item name (required)

        public decimal ItemPrice { get; set; }                          // Item price

        public int DepartmentId { get; set; }                           // Foreign key

        public required Department RelatedDepartment { get; set; }      // Navigation property (required)

    }

    // Represents a department or category in the store

    public class Department

    {

        public int DepartmentId { get; set; }                           // Primary Key

        public required string DepartmentName { get; set; }             // Department name (required)

        public List<StoreItem> ItemsInDepartment { get; set; } = new(); // Navigation list

    }

    // The database context that manages access to the database

    public class InventoryDbContext : DbContext

    {

        public DbSet<StoreItem> StoreItems { get; set; } = null!;

        public DbSet<Department> Departments { get; set; } = null!;

        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

        {

            // Connection string for SQL Server LocalDB instance (v11.0)

            optionsBuilder.UseSqlServer("Server=(localdb)\\v11.0;Database=RetailInventoryDb;Trusted\_Connection=True;");

        }

    }

    // Program entry point

    class Startup

    {

        static void Main()

        {

            using var dbContext = new InventoryDbContext();

            // Create database and tables if not already existing

            dbContext.Database.EnsureCreated();

            // Create a new department and item

            var electronicsDepartment = new Department { DepartmentName = "Electronics" };

            var newItem = new StoreItem

            {

                ItemName = "Laptop",

                ItemPrice = 80000,

                RelatedDepartment = electronicsDepartment

            };

            // Save the new data

            dbContext.Departments.Add(electronicsDepartment);

            dbContext.StoreItems.Add(newItem);

            dbContext.SaveChanges();

            Console.WriteLine("Inventory data successfully saved to the database.");

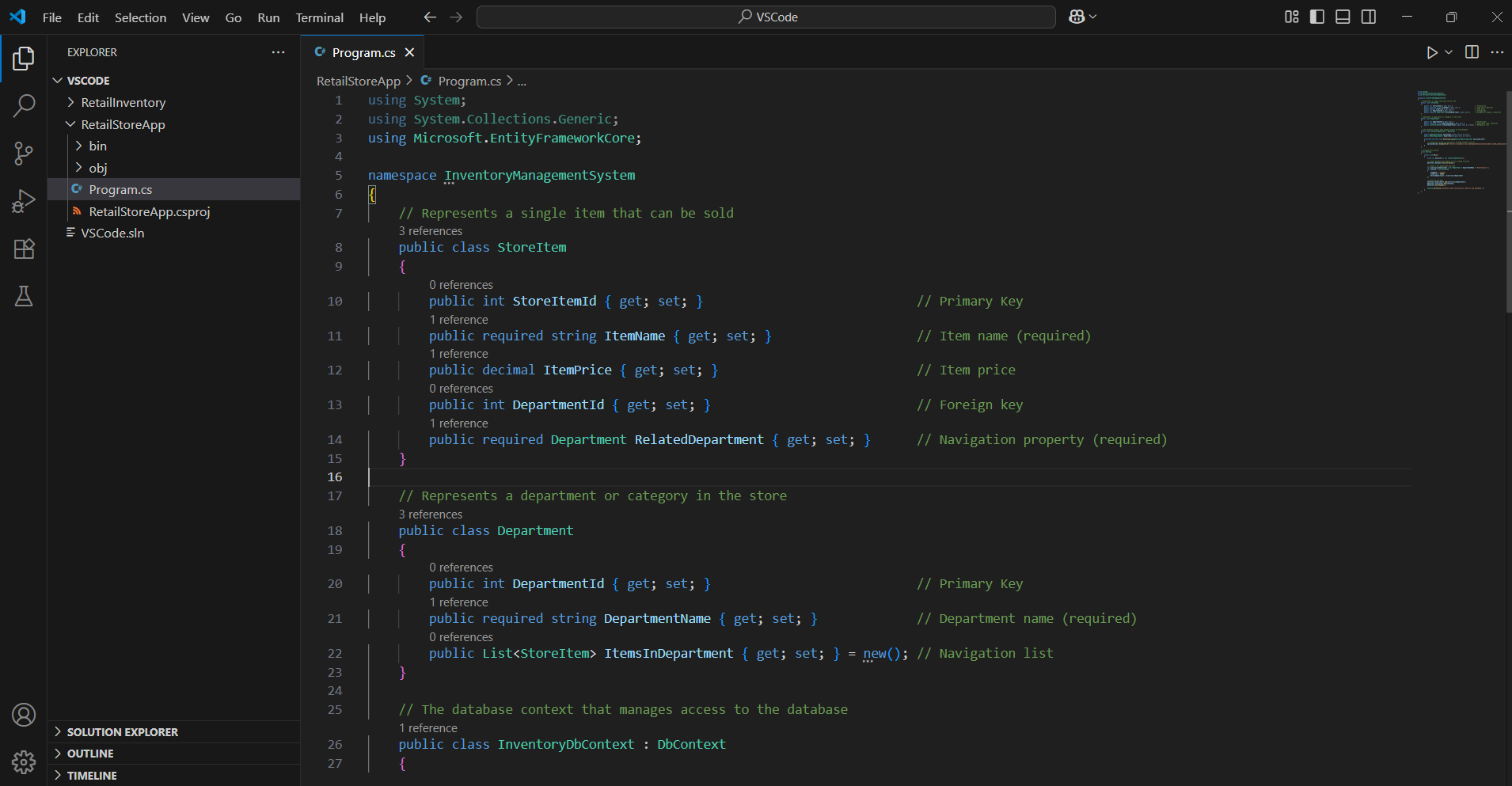
        }

    }

}

OUTPUT:

Inventory data successfully saved to the database.

ScreenShots

