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

// ProductModel.cs

namespace RetailStoreApp.Models

{

public class ProductModel

{

public int ProductModelId { get; set; }

public string Name { get; set; } = string.Empty;

public decimal Price { get; set; }

public int Stock { get; set; }

public string Description { get; set; } = string.Empty;

}

}

// CustomerModel.cs

namespace RetailStoreApp.Models

{

public class CustomerModel

{

public int CustomerModelId { get; set; }

public string FullName { get; set; } = string.Empty;

public string Email { get; set; } = string.Empty;

public string PhoneNumber { get; set; } = string.Empty;

}

}

// RetailDbContext.cs

using Microsoft.EntityFrameworkCore;

using RetailStoreApp.Models;

namespace RetailStoreApp.Data

{

public class RetailDbContext : DbContext

{

public RetailDbContext(DbContextOptions<RetailDbContext> options) : base(options) { }

public DbSet<ProductModel> Products => Set<ProductModel>();

public DbSet<CustomerModel> Customers => Set<CustomerModel>();

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<ProductModel>()

.Property(p => p.Price)

.HasPrecision(18, 2);

}

}

}

// Program.cs

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

using Microsoft.Extensions.Configuration;

using Microsoft.EntityFrameworkCore;

using RetailStoreApp.Data;

using RetailStoreApp.Models;

var host = CreateHostBuilder(args).Build();

using var scope = host.Services.CreateScope();

var services = scope.ServiceProvider;

try

{

var dbContext = services.GetRequiredService<RetailDbContext>();

dbContext.Database.EnsureCreated();

Console.WriteLine("✅ Database successfully initialized.");

if (!dbContext.Products.Any())

{

dbContext.Products.AddRange(

new ProductModel { Name = "Laptop", Price = 1200.00m, Stock = 10, Description = "High-performance laptop" },

new ProductModel { Name = "Mouse", Price = 25.00m, Stock = 50, Description = "Wireless mouse" }

);

dbContext.SaveChanges();

}

var products = dbContext.Products.ToList();

Console.WriteLine("🛒 Current Products:");

foreach (var product in products)

{

Console.WriteLine($"- {product.Name} (${product.Price}) - {product.Stock} in stock - {product.Description}");

}

}

catch (Exception ex)

{

Console.WriteLine("❌ Error accessing database:");

Console.WriteLine(ex.Message);

}

static IHostBuilder CreateHostBuilder(string[] args) =>

Host.CreateDefaultBuilder(args)

.ConfigureAppConfiguration((context, config) =>

{

config.AddJsonFile("appsettings.json", optional: false, reloadOnChange: true);

})

.ConfigureServices((context, services) =>

{

var connectionString = context.Configuration.GetConnectionString("RetailDbConnection");

services.AddDbContext<RetailDbContext>(options =>

options.UseSqlite(connectionString));

});

// appsettings.json (Place in project root as a separate file)

{

"ConnectionStrings": {

"RetailDbConnection": "Data Source=retailstore.db"

}

}

Out put:

✅ Database successfully initialized.

🛒 Current Products:

- Laptop ($1200.00) - 10 in stock - High-performance laptop

- Mouse ($25.00) - 50 in stock - Wireless mouse