**Source Code:** Full Project Link: https://github.com/khushi-TEK/BuyLaptop/

Program.cs

using Microsoft.AspNetCore.Hosting;

using Microsoft.Extensions.Configuration;

using Microsoft.Extensions.Hosting;

using Microsoft.Extensions.Logging;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace LaptopStore

{

public class Program

{

public static void Main(string[] args)

{

CreateHostBuilder(args).Build().Run();

}

public static IHostBuilder CreateHostBuilder(string[] args) =>

Host.CreateDefaultBuilder(args)

.ConfigureLogging(logging =>

{

logging.AddConsole();

})

.ConfigureWebHostDefaults(webBuilder =>

{

webBuilder.UseStartup<Startup>();

});

}

}

Startup.cs

using LaptopStore.Repositories;

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Hosting;

using Microsoft.Extensions.Configuration;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.EntityFrameworkCore;

using LaptopStore.Data;

namespace LaptopStore

{

public class Startup

{

public Startup(IConfiguration configuration)

{

Configuration = configuration;

}

public IConfiguration Configuration { get; }

// This method gets called by the runtime. Use this method to add services to the container.

public void ConfigureServices(IServiceCollection services)

{

services.AddControllersWithViews();

/\*

services.Add(

new ServiceDescriptor(typeof(IProductsRepository),

typeof(InMemoryProductsRepository),

ServiceLifetime.Scoped));

\*/

services.Add(new ServiceDescriptor(typeof(IProductsRepository), typeof(EntityFrameworkProductsRepository), ServiceLifetime.Scoped));

services.AddDbContext<LaptopStoreContext>(options =>

options.UseSqlServer(@"Server=.\SQLExpress;Database=LaptopStore;User Id=sa;Password=Sairam123$;"));

services.AddSession(options =>

{

options.IdleTimeout = TimeSpan.FromMinutes(10);

options.Cookie.HttpOnly = true;

options.Cookie.IsEssential = true;

});

}

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

else

{

app.UseExceptionHandler("/Home/Error");

}

app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.UseSession();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

});

}

}

}

Cart.cs

using Newtonsoft.Json;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace LaptopStore.Models

{

public class Cart

{

public int ID { get; set; }

public int CustomerID { get; set; }

public List<CartItem> Items { get; set; } = new List<CartItem>();

public CartItem FindItemByProductId(int productId)

{

return Items.FirstOrDefault(item => item.Product.ID == productId);

}

public void Add(CartItem item)

{

Items.Add(item);

}

public void Remove(CartItem item)

{

Items.Remove(item);

}

}

}

Product.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace LaptopStore.Models

{

public class Product

{

public int ID { get; set; }

public string Name { get; set; }

public string Brand { get; set; }

public decimal Price { get; set; }

}

}