Giả sử đề bài là :

Sử dụng CSDL Vali , hiển thị MENU theo 1 tiêu chí bất kỳ. Sử dụng template karma.

Đã cho tất cả folder css,js,fonts,img vào trong Template. Thay thế toàn bộ theo quy tắc :

~/ là địa chỉ folder root của các file , ví dụ như Views , Shared

../ là địa chỉ folder wwwroot

Mẫu này tương thích với ASP-ROUTE . Code vẫn chạy kể cả nếu không có biến trong link. Tuy nhiên sẽ không hiển thị gì cả, nhưng có thể kiểm tra tay bằng cách thêm đoạn sau:?tieuchi=thông tin gì đấy

Cần sử dụng 5 mẫu sau , chia thành 5 nhóm :

Nhóm 1: Class và Interface Class: Folder : Repository;

Làm Menu thì chỉ cần Get và GetAll , nên mình chỉ sửa code tại hai phần này.

Nhóm 2: ViewComponent để hiển thị: Folder : ViewComponents;

Chú ý thêm chữ s ở cuối để tránh trùng tên với Class

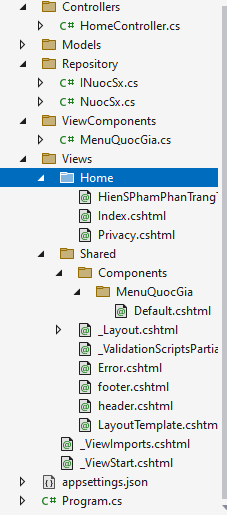
Nhóm 3: Shared/View/Component/Tên Layout cần hiển thị/Default.cshtml;

Ở đây làm VD với trường hợp menu theo mã nước SX, nên đặt tên Layout là MenuQuocGia

Nhóm 4: Program.cs ở ngoài cùng

Nhóm 5: Layout

Cấu trúc cuối cùng của project:



Nhóm 1: Class và Interface Class: Folder : Repository;

Tạo 1 class và interface theo mẫu sau, nếu nhỡ tạo class thì chỉ cần thêm chữ interface sau public là được :

Interface:

using KIEMTRA.Models;

namespace KIEMTRA.Repository

{

public interface INuocSx

{

TQuocGia Add(TQuocGia QuocGia);

//Để thêm vào CSDL

TQuocGia Update(TQuocGia QuocGia);

//Để update CSDL

TQuocGia Delete(String MaQgia);

//Để xóa CSDL

TQuocGia GetQuocGia(String MaQgia);

//Để tìm trong CSDL

IEnumerable<TQuocGia> GetAllQuocGia();

//Để lấy toàn bộ CSDL

}

}

Class:

using KIEMTRA.Models;

namespace KIEMTRA.Repository

{

public class NuocSx : INuocSx

//Để sử dụng Interface

{

private readonly QlbanVaLiContext csdl=new QlbanVaLiContext();

//Khai báo CSDL

public NuocSx(QlbanVaLiContext csdl)

{

this.csdl = csdl;

}

//Để truy cập CSDL

public TQuocGia Add(TQuocGia QuocGia)

{

throw new NotImplementedException();

}

public TQuocGia Delete(string MaQgia)

{

throw new NotImplementedException();

}

public IEnumerable<TQuocGia> GetAllQuocGia()

{

return csdl.TQuocGia;

}

public TQuocGia GetQuocGia(string MaQgia)

{

return csdl.TQuocGia.Find(MaQgia);

}

public TQuocGia Update(TQuocGia QuocGia)

{

throw new NotImplementedException();

}

}

}

Nhóm 2: ViewComponent để hiển thị: Folder : ViewComponents;

ViewComponent, chú ý phải đặt tên class trong file có chứa chữ ViewComponent, nếu không ASP.NET sẽ không nhận dạng được. Tên ngoài file đặt là gì cũng được .

Ở đây ASP.NET sẽ nhận diện MenuQuocGia là 1 ViewComponent vì có chứa cụm chữ này trong tên class, mặc dù tên file không có chữ ViewComponent.

using KIEMTRA.Repository;

using Microsoft.AspNetCore.Mvc;

namespace KIEMTRA.ViewComponents

{

public class MenuQuocGiaViewComponent:ViewComponent

{

private readonly INuocSx nuocSx;

//Gọi ra interface để chạy lệnh

public MenuQuocGiaViewComponent(INuocSx nuocSx)

{

this.nuocSx = nuocSx;

}

public IViewComponentResult Invoke()

{

var allQuocGia = nuocSx.GetAllQuocGia().OrderBy(x => x.MaNuoc).ToList();

return View(allQuocGia);

}

//Invoke là bắt buộc để InvokeAsync

}

}

Nhóm 3: Shared/View/Component/Tên Layout cần hiển thị/Default.cshtml, ở đây mình sửa và dùng luôn Controller của HienSanPham để hiển thị theo tiêu chí nước Sx;

Defaut.cshtml:

@model IEnumerable<TQuocGia>

<h1>Đây là Menu</h1>

@foreach (var tieuchi in Model)

{

<li class="nav-item"><**a** **asp-controller**="Home" **asp-action**="HienSPhamPhanTrangTChiBatKy" **asp-route-tieuchi**="@tieuchi.MaNuoc">Hãng : @tieuchi.TenNuoc</**a**></li>

}

@\*Mình đang chèn vào Menu drop down của template , nhưng các bạn để chỗ nào cũng được\*@

Controller:

public IActionResult HienSPhamPhanTrangTChiBatKy(string TieuChi,int? page)

{

int pageSize = 8;//Số đối tượng hiển thị tối đa trên 1 trang

int pageNumber = page == null || page < 0 ? 1 : page.Value;

//Số trang hiện tại đang truy cập

var ds\_san\_pham = csdl.TDanhMucSps.AsNoTracking().Where(x => x.MaNuocSx == TieuChi).OrderBy(x => x.TenSp);

//Khởi tạo 1 danh sách tự do, có tiêu chí là có MaNuocSx trùng với tiêu chí

PagedList<TDanhMucSp> paged\_ds\_san\_pham = new PagedList<TDanhMucSp>(ds\_san\_pham, pageNumber, pageSize);

//Khai báo model 1 danh sách đã được phân trang, có mẫu là TDanhMucSp

ViewBag.tieuchi = TieuChi;

//Tạo 1 ViewBag để gửi thông tin sang View

return View(paged\_ds\_san\_pham);

//Trả về View chạy theo model đã được đưa ra ở trên

}

Nhóm 4: Program.cs:

Thêm đoạn lệnh sau vào trong program .cs, trước đoạn

var app = builder.Build();

Đoạn lệnh:

var connectionString = builder.Configuration.GetConnectionString("QlBanValiContext");

builder.Services.AddDbContext<QlbanVaLiContext>(x => x.UseSqlServer(connectionString));

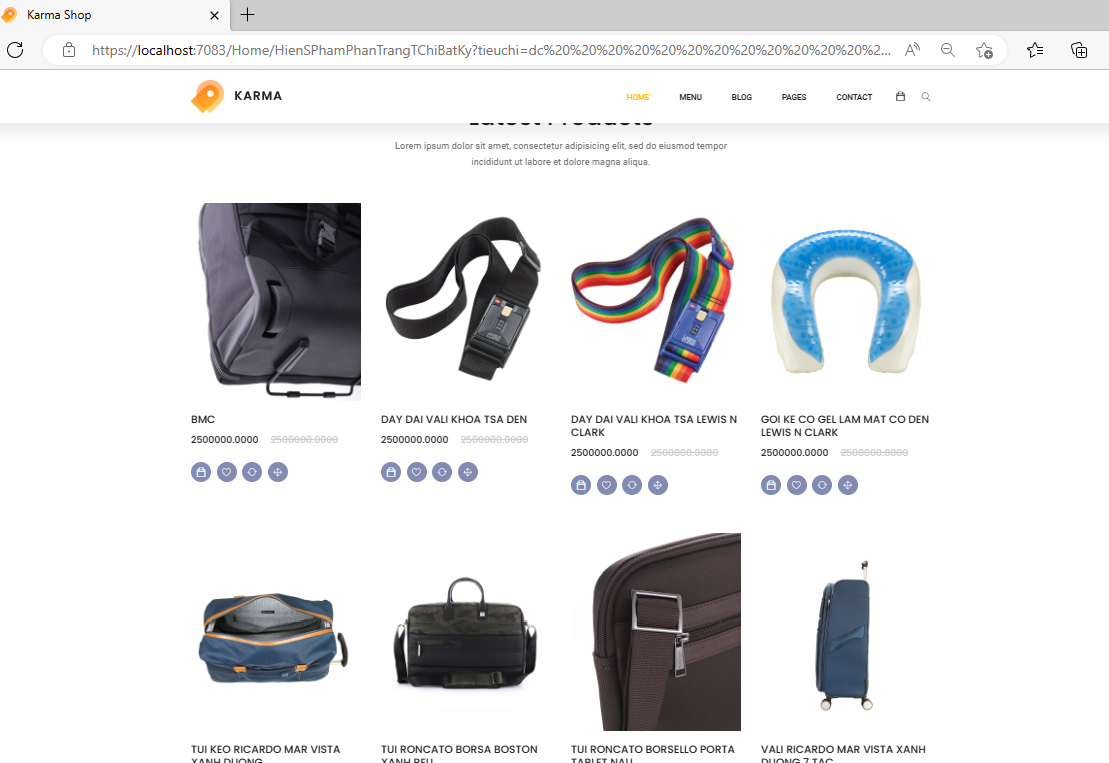
builder.Services.AddScoped<INuocSx, NuocSx>();

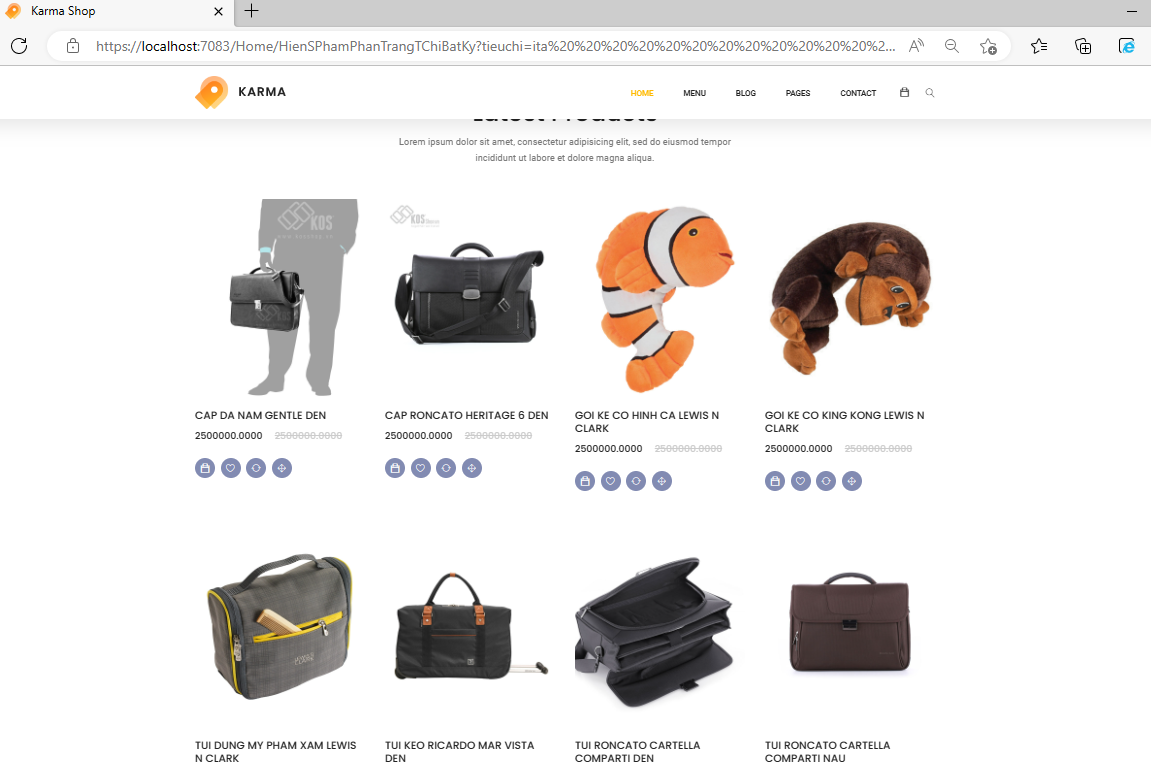
//gắn interface vào class, nhớ đặt tên đúng

Nhóm 5: Layout :

Sử dụng @await Component.InvokeAsync("MenuQuocGia");

Vì tên folder trong component là MenuQuocGia

Kết quả chạy : tieuchi=dc

Tieuchi=ita

Hình ảnh menu thực :

