Cần tải những cái này trước trong Tools/ Nuget Packet / For Solution

Install-Package Microsoft.EntityFrameworkCore.Design -v 6.0.21

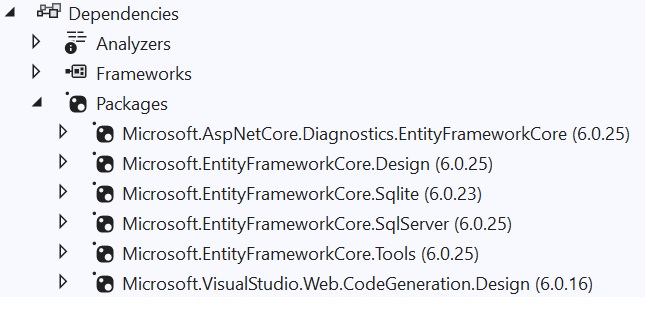
Install-Package Microsoft.AspNetCore.Diagnostics.EntityFrameworkCore -v 6.0.21

Install-Package Microsoft.EntityFrameworkCore.SqlServer -v 6.0.21

Install-Package Microsoft.EntityFrameworkCore.Sqlite -v 6.0.21

Install-Package Microsoft.EntityFrameworkCore.Tools

Install-Microsoft.VisualStudio.Web.CodeGeneration.Design



B0: KẾT NỐI SQL . Trong Connected Services. Chọn Add/ SQLSeverDatabase

* Đặt tên là QLHangHoaContext.



* Dòng dưới bấm vào dấu ...

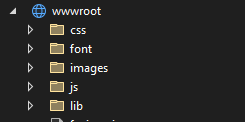
Phải dùng sa để add DB vào

Phải lấy đường dẫn ( lấy trong secrete.json) để copy vào đường dẫn bên dưới

Tools/Nuget Packet/ Packet Manager Console

Scaffold-DbContext "Data Source=LAMVU\SQLEXPRESS;Initial Catalog=QLHangHoa;Persist Security Info=True;User ID=sa;Password=123;" Microsoft.EntityFrameworkCore.SqlServer -Project Test03 -OutputDir Models

B1: copy css , font img và js vào



* Dòng code sau để lấy js, img, css vào:

<**link** href="~/css/style.css" rel="stylesheet" type="text/css" media="all">

<**link** href="~/css/slider.css" rel="stylesheet" type="text/css" media="all">

<**script** src="~/lib/jquery/dist/jquery.min.js"></**script**>

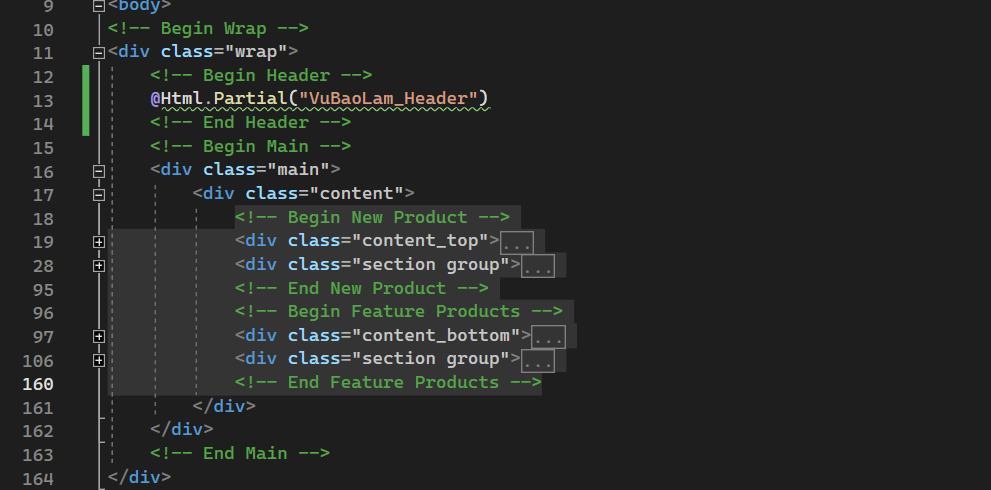
<**script** src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></**script**>

<**script** **src**="~/js/site.js" **asp-append-version**="true"></**script**>

B2: Tạo VuBaoLam\_Header. Cắt toàn bộ phần header của VuBaoLam\_Layout sang.

Tạo VuBaoLam\_MainContent. Cắt toàn bộ phần main của VuBaoLam\_Layout sang

VuBaoLam\_Main. Ở bên Layout thêm @RenderBody()



Tương tự tạo VuBaoLam\_ Footer rồi cắt Footer bên VubaoLam\_Layout sang và bên Layout thêm <**partial** **name**="VuBaoLam\_Footer" />



B3 : Trong Home/Index.cshtml thêm để hiển thị Layout lên

Layout = "~/Views/Shared/VuBaoLam\_Layout.cshtml";

B5: Thêm folder ViewComponent và thêm Class LoaiHangViewComponent

public async Task<IViewComponentResult> InvokeAsync()

{

QlhangHoaContext db = new QlhangHoaContext();

var loaihang = db.LoaiHangs.ToList();

return View(loaihang);

}

B6: Trong Shared tạo thư mục Components trong đó tạo tiếp LoaiHang ( ên phải giống trong View Components là LoaiHangViewComponent) . Trong LoaiHang tạo Partial View : Default

Trong Default thêm :

@model IEnumerable<LoaiHang>

<ul>

@\* Thay đổi tên cho menu Home được thay thế bằng họ tên sinh \*@

<li class="active"><a id=" null" href="#">Vũ Bảo Lâm</a></li>

@foreach (var item in Model)

{

<li><a href="#" id="@item.MaLoai">@item.TenLoai</a></li> @\* Load tên các loại hàng mã hàng \*@

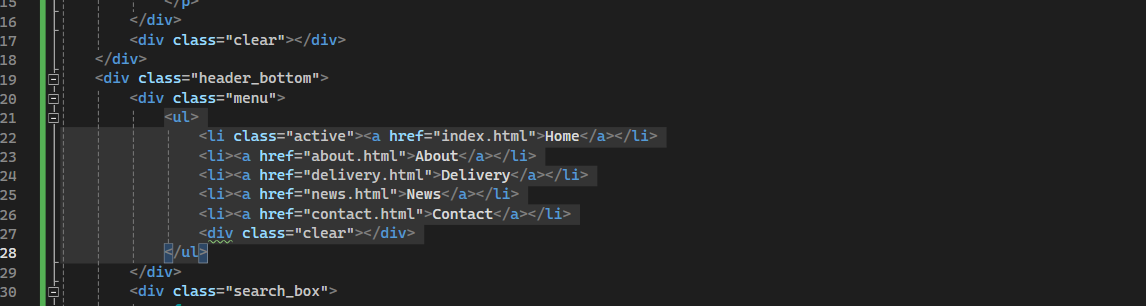
}

<div class="clear"></div>

</ul>

Tiếp tục : sửa trong Header Xoá cái này và thêm

@await Component.InvokeAsync("LoaiHang"); Cái này phải trùng với ViewComponent



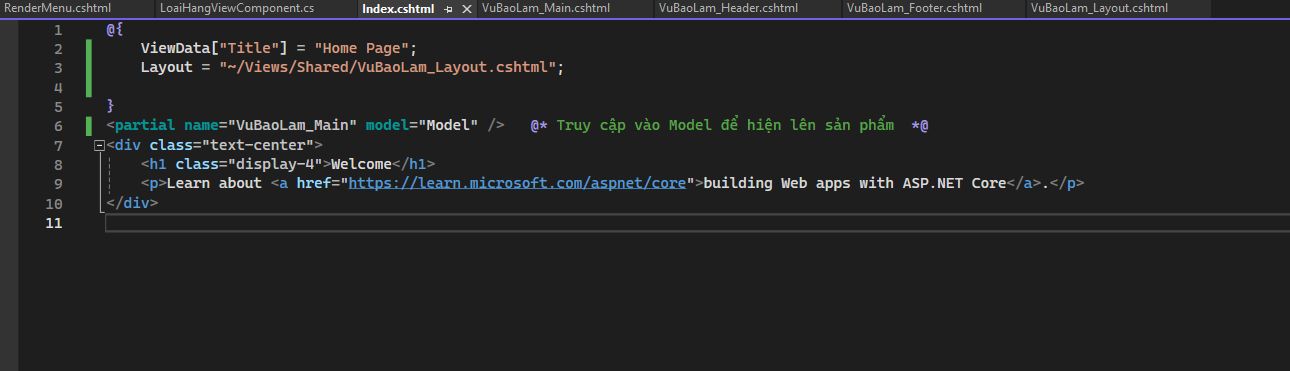
---Đến đây là đc 5 điểm rồi ----

Câu 3: B7 : **Thêm cái này ở đầu :**

QlhangHoaContext db = new QlhangHoaContext();

Tiếp trong Index : thêm để đẩy toàn bộ dữ liệu hàng lên

<**partial** **name**="VuBaoLam\_Main" **model**="Model" />



B8: Trong HomeController sửa để cập nhật giá hàng hoá lớn hơn 100

public IActionResult HangHoaTheoLoai(int? maloaihang)

{

if (maloaihang == null)

{

var hanghoa = db.HangHoas.ToList();

return PartialView("HangHoa", hanghoa);

}

else

{

var hanghoa = db.HangHoas.Where(p => p.MaLoai == maloaihang).ToList();

return PartialView("HangHoa", hanghoa);

}

}

Câu 4: Tạo Action trong HomeController

public IActionResult HangHoaTheoLoai(int? maloaihang)

{

var hanghoa = db.HangHoas.Where(p => p.MaLoai == maloaihang).ToList();

return PartialView("HangHoa", hanghoa);

}

B9 : Tiếp tục trong Home tạo Partial View : HangHoa

* Và thêm code ( phần div là lấy từ 1 sản phẩm bất kì )

@model IEnumerable<HangHoa>

@foreach (var item in Model)

{

<div class="grid\_1\_of\_4 images\_1\_of\_4">

<a href="#"><img src="/images/@item.Anh" alt="@item.TenHang" /></a>

<h2>@item.TenHang </h2>

<div class="price-details">

<div class="price-number">

<p><span class="rupees">$@item.Gia</span></p>

</div>

<div class="add-cart">

<h4><a href="preview.html">Add to Cart</a></h4>

</div>

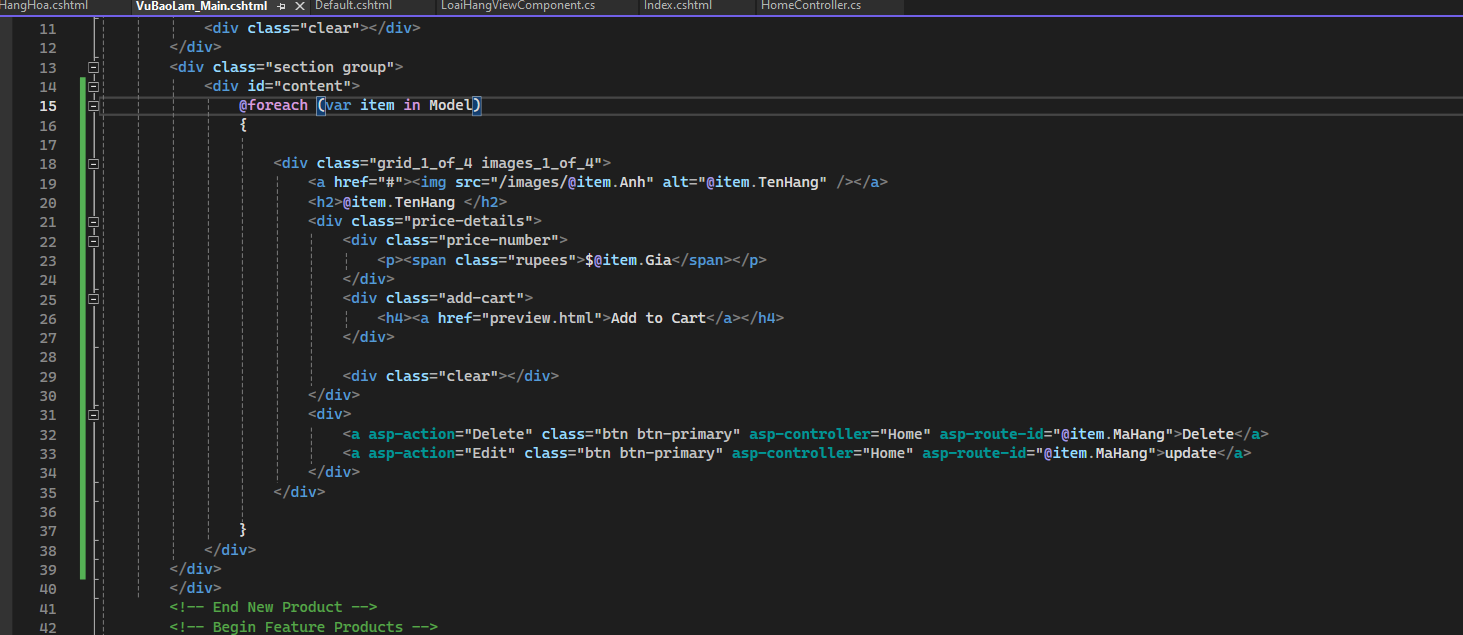
<div class="clear"></div>

</div>

</div>

}

* Sửa tiếp trong VubaoLam\_Main ở thẻ section group xoá hết đến end new product thêm code này :



<div id="content">

@foreach (var item in Model)

{

<div class="grid\_1\_of\_4 images\_1\_of\_4">

<a href="#"><img src="/images/@item.Anh" alt="@item.TenHang" /></a>

<h2>@item.TenHang </h2>

<div class="price-details">

<div class="price-number">

<p><span class="rupees">$@item.Gia</span></p>

</div>

<div class="add-cart">

<h4><a href="preview.html">Add to Cart</a></h4>

</div>

<div class="clear"></div>

</div>

<div>

<**a** **asp-action**="Delete" class="btn btn-primary" **asp-controller**="Home" **asp-route-id**="@item.MaHang">Delete</**a**>

<**a** **asp-action**="Edit" class="btn btn-primary" **asp-controller**="Home" **asp-route-id**="@item.MaHang">update</**a**>

</div>

</div>

}

</div>

* Và tiếp tục trong Views/Home/Index thêm

@section Scripts {

<script>

$("ul li a").on('click', function () {

var id = $(this).attr("id")

$.ajax({

url: "/Home/HangHoaTheoLoai",

data: { maloaihang: id },

success: function (res) {

$("div#content").html(res)

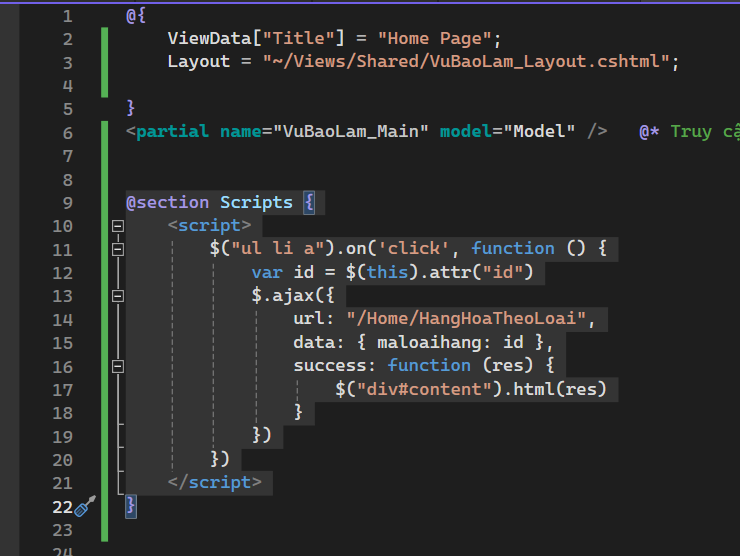
}

})

})

</script>

}



---- XONG câu 4 ---

NÚT THÊM

public IActionResult Create()

{

ViewBag.MaLoai = new SelectList(db.LoaiHangs, "MaLoai", "TenLoai");

return View();

}

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Create([Bind("MaLoai", "TenHang", "Gia", "Anh")] HangHoa hanghoa)

{

if (ModelState.IsValid)

{

db.Add(hanghoa);

await db.SaveChangesAsync();

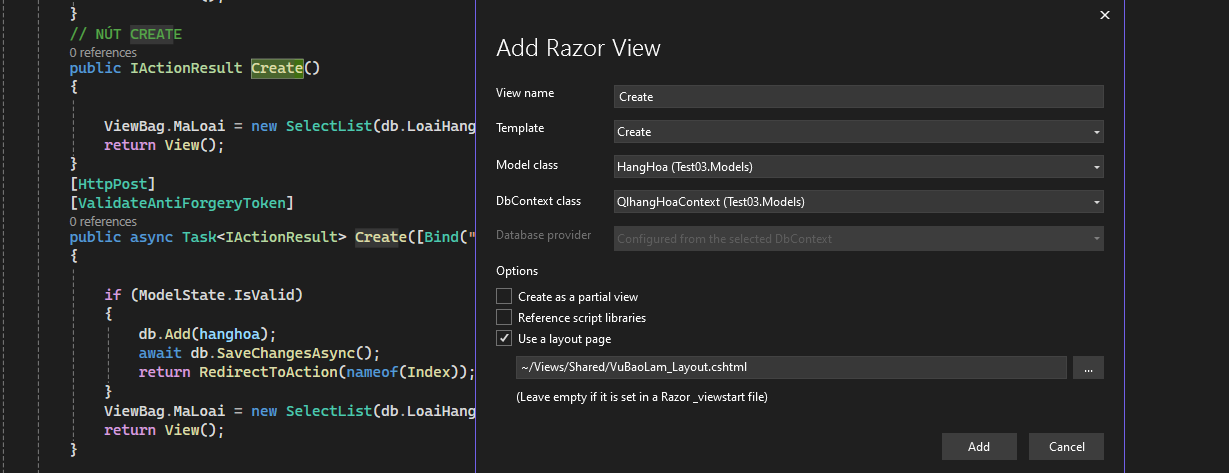
return RedirectToAction(nameof(Index));

}

ViewBag.MaLoai = new SelectList(db.LoaiHangs, "MaLoai", "TenLoai");

return View();

}



* Trong Views/Home/Index thêm :

<**a** **asp-action**="Create" class="btn btn-primary">Create</**a**>

* Trong Models/HangHoa thêm dấu ?



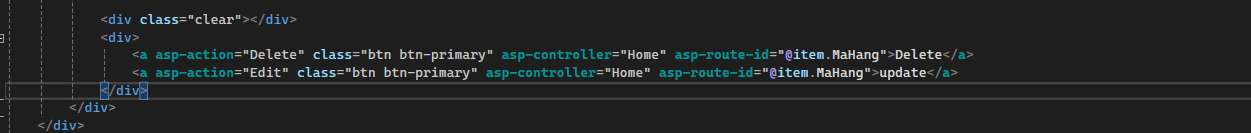
* Trong Views/Home/HangHoa thêm sau thẻ clear

<div>

<**a** **asp-action**="Delete" class="btn btn-primary" **asp-controller**="Home" **asp-route-id**="@item.MaHang">Delete</**a**>

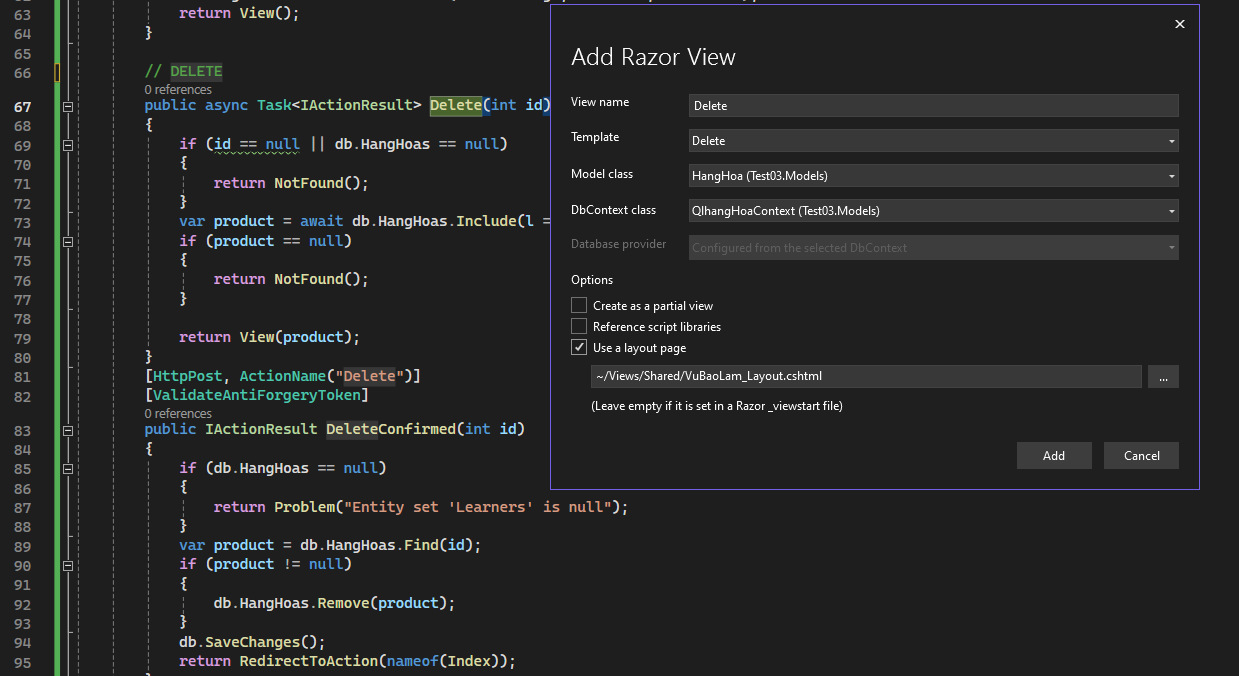
<**a** **asp-action**="Edit" class="btn btn-primary" **asp-controller**="Home" **asp-route-id**="@item.MaHang">update</**a**>

</div>



NÚT XOÁ VÀ SỬA

LÀM TƯƠNG TỰ NHƯ CREATE



* Lưu ý trong Main thêm

<**a** **asp-action**="Delete" class="btn btn-primary" **asp-controller**="Home" **asp-route-id**="@item.MaHang">Delete</**a**>

<**a** **asp-action**="Edit" class="btn btn-primary" **asp-controller**="Home" **asp-route-id**="@item.MaHang">update</**a**>



**-----------------TÌM KIẾM-----------------**

* Trong Views/Home/Index thêm

$("input#btnSearch").click(function (e) {

var keyword = $("#keyword").val();

var data = { keyword: keyword };

$.ajax({

url: "/Home/tk", //Đường dẫn gửi request

data: data, //Tham số được gửi kèm request lên server

success: function (response) { // callback function khi nhận response từ server

$("div#contentView").html(response);

}

});

});

* Trong HomeController thêm

public IActionResult tk(string keyword)

{

var hanghoa = db.HangHoas.Where(l => l.TenHang.ToLower().Contains(keyword.ToLower()));

return PartialView("HangHoa", hanghoa);

}

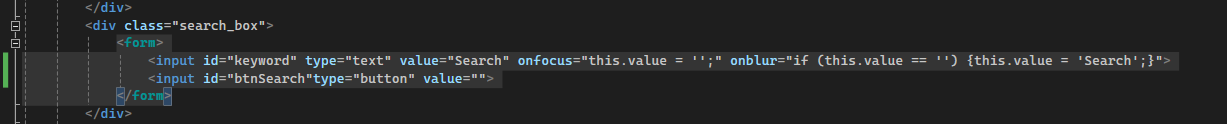
* **Trong** VuBaoLam\_Header tìm dòng <div class="search\_box">

<**form**>

<input id="keyword" type="text" value="Search" onfocus="this.value = '';" onblur="if (this.value == '') {this.value = 'Search';}">

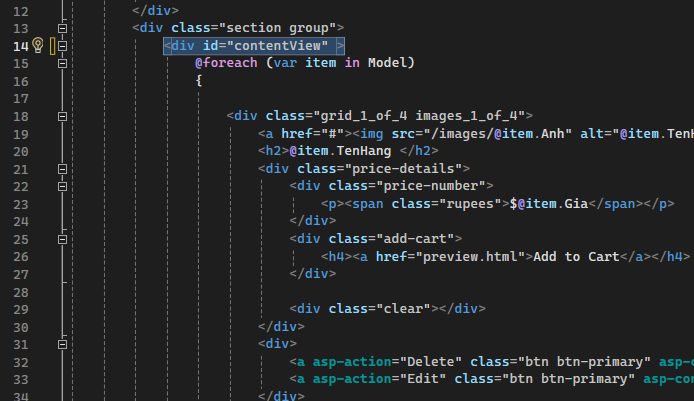
<input id="btnSearch"type="button" value="">

</**form**>

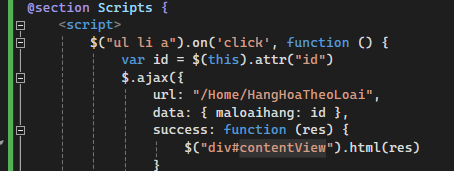


* Note sửa ( ko quan trọng lắm) dùng để load đúng

Trong VuBaoLam\_Main sửa content 🡺 contentView



Tương tự bên Index



NOTE :

var hanghoa = db.HangHoas.OrderBy(p => p.Gia).Take(4).ToList(); // LOAD top 4 san pham co gia thap nhat tang dan

var hanghoa = db.HangHoas.OrderByDescending(p =>p.Gia).Take(3).ToList(); //LOAD top 3 có giá cao nhất giảm dần

var products = db.Products.Where(p => p.Available == true && p => p.Gia >= 100 ).ToList(); // LOAD sp có available = true và UnitPrice <=1000

VALIDATE :

* Email :

[Required(ErrorMessage = "Email bắt buộc phải được nhập")]

[RegularExpression(@"[A-Za-z0-9.\_%+-]+@gmail.com", ErrorMessage = "Email phải có gmail.com")]

public string? Email { get; set; }

* Mật Khẩu

[Required(ErrorMessage = "Bắt buộc phải nhập mật khẩu")]

[StringLength(100, MinimumLength = 8)]

[RegularExpression(@"^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*\W).{8,}$", ErrorMessage = "Mật khẩu từ 8 ký tự trở lên, có ký tự viết hoa, viết thường, chữ số và ký tự đặc biệt")]

public string? Password { get; set; }

* Mật khẩu nhập lần thứ 2 phải trùng

[Required(ErrorMessage = "Password is required")]

public string Password { get; set; }

[Required(ErrorMessage = "RePassword is required")]

[Compare("Password", ErrorMessage = "RePassword must match Password")]

public string RePassword { get; set; }

* Giá

[Required(ErrorMessage = "Price is required")]

[Range(100, 5000, ErrorMessage = "Price must be between 100 and 5000")]

* Ảnh

[Required(ErrorMessage = "Image file is required")]

[RegularExpression(@"^.+\.(jpg|png|gif|tiff)$", ErrorMessage = "Invalid file format. Supported formats: jpg, png, gif, tiff")]

public string? Anh { get; set; }