**Tải package:** Microsoft.EntityFrameworkCore.Tools, Microsoft.EntityFrameworkCore, Microsoft.EntityFrameworkCore.SqlServer

**Kết nối database**: Tools -> Connect to Database

**Chạy câu lệnh sinh Model:** Scaffold-DbContext “Data Source=V\SQLEXPRESS;Initial Catalog=QLHangHoa;Integrated Security=True;Trust Server Certificate=True” Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models

**Sang Program.cs thêm vào dòng 11** builder.Services.AddDbContext<QlhangHoaContext>(options => options.UseSqlServer("Data Source=V\\SQLEXPRESS;Initial Catalog=QLHangHoa;Integrated Security=True;Trust Server Certificate=True"));

**Cho file css, js, image vào wwwroot**

**Tạo view Layout trong folder Views\Shared, copy nội dung index.html vào layout, sửa đường dẫn css, js**

**Tạo các partial view (Header, Footer, Main)**

**Sửa layout trong ViewStart thành NguyenDucVuong\_Layout**

**Tạo controller, view index**

**Vào Program.cs sửa pattern controller = (tên controller vừa tạo)**

**Tọa thư mục ViewComponents, trong thư mục tạo class RenderViewComponent.cs**

public class RenderViewComponent : ViewComponent

{

private List<LoaiHang> loaihangs;

private readonly QlhangHoaContext \_context;

public RenderViewComponent(QlhangHoaContext context)

{

\_context = context;

loaihangs = \_context.LoaiHangs.ToList();

}

public async Task<IViewComponentResult> InvokeAsync()

{

return View(“RenderNav”, loaihangs);

}

}

**Tạo thư mục Shared/Components/Render**

**Tạo View RenderNav trong thư mục Render**

@model IEnumerable<LoaiHang>

<div class="header\_bottom">

<div class="menu">

<ul>

<li class="active"><a href="index.html">Nguyễn Đức Vượng</a></li>

@foreach(var i in Model)

{

<li><a href="about.html">@i.TenLoai</a></li>

}

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

</ul>

</div>

<div class="search\_box">

<form>

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

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

</form>

</div>

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

</div>

**Trong layout ở phần bị cắt thêm vào dòng chữ :** @await Component.InvokeAsync("Render")

**Trong Controller sửa lại thành:** public async Task<IActionResult> Index()

{

List<HangHoa> hanghoas = \_context.HangHoas.Where(x=>x.Gia>=100).ToList();

return View(hanghoas);

}

**Vào view Create xóa phần Scripts**

**Annotation**

[Range(100, 5000, ErrorMessage = "Giá phải nằm trong khoảng từ 100 đến 5000.")]

public decimal? Gia { get; set; }

[RegularExpression(@".\***\.**(jpg|png|gif|tiff)$", ErrorMessage = "Tên file ảnh phải có đuôi .jpg, .png, .gif, hoặc .tiff.")]

public string? Anh { get; set; }

public virtual LoaiHang? MaLoaiNavigation { get; set; } = null!;

[Required(ErrorMessage = "Tên sản phẩm không được để trống.")]

[Range(1, 100, ErrorMessage = "Số lượng phải nằm trong khoảng từ 1 đến 100.")]

[StringLength(50, ErrorMessage = "Tên sản phẩm không được vượt quá 50 ký tự.")]

[StringLength(20, MinimumLength = 5, ErrorMessage = "Tên sản phẩm phải từ 5 đến 20 ký tự.")]

[MaxLength(100)]

[MinLength(3, ErrorMessage = "Mã sản phẩm phải có ít nhất 3 ký tự.")]

[RegularExpression(@"^[A-Z]{3}\d{4}$", ErrorMessage = "Mã sản phẩm phải có 3 chữ cái in hoa theo sau là 4 chữ số.")]

[EmailAddress(ErrorMessage = "Email không hợp lệ.")]

[Phone(ErrorMessage = "Số điện thoại không hợp lệ.")]

[Url(ErrorMessage = "Địa chỉ URL không hợp lệ.")]

[Compare("Password", ErrorMessage = "Mật khẩu xác nhận không khớp.")]