

**TRƯỜNG ĐẠI HỌC HỒNG ĐỨC
KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



BÀI THỰC HÀNH

HỌC PHẦN: LẬP TRÌNH WEB

SỐ TÍN CHỈ: 3

DÙNG CHO: SINH VIÊN NGÀNH CNTT

THANH HÓA, 02/2023

HƯỚNG DẪN CHUNG

1. CHÍNH SÁCH CỦA MÔN HỌC

- Sinh viên "**bắt buộc**" in bài thực hành và mang theo tại các buổi thực hành. Sinh viên phải làm bài thực hành đầy đủ của các buổi thực hành, khuyến khích sinh viên mang máy tính cá nhân để thực hành. Sinh viên không có máy tính cá nhân thực hành trên máy tính nhà trường yêu cầu cuối buổi thực hành cần lưu lại code, CSDL của buổi thực hành ngày hôm đấy để phục vụ các buổi thực hành tiếp theo và kiểm tra. Sinh viên có thể sử dụng GIT để quản lý code hoặc có thể nén file lưu giữ lên Google drive, copy vào USB.
- Sinh viên vắng 1 (một) buổi thực hành sẽ bị đánh vắng tương đương 05 (năm) tiết học.
- Sinh viên mang thêm bài giảng lý thuyết, tài liệu tham khảo của học phần.
- Tổng số buổi thực hành của học phần là 06 buổi, tương đương 30 tiết.

2. PHẦN MỀM PHỤC VỤ

- Các bài tập thực hành trong học phần này được xây dựng thực hành với ASP.Net Web Application (.NET Framework) + C# + Mô hình MVC + Entity Framework
- Visual Studio phiên bản 2022 , Sql server 2019.
- Địa chỉ để tải phần mềm
 - + Link cài đặt SQL Server 2019 Express:
<https://www.microsoft.com/en-US/download/details.aspx?id=101064>
 - + Cài đặt Visual Studio 2022 Community:
<https://visualstudio.microsoft.com/downloads/>

3. TÀI LIỆU THAM KHẢO

- Internet:

<https://learn.microsoft.com/en-us/aspnet/mvc/>

https://www.tutorialspoint.com/asp.net_mvc/asp.net_mvc_getting_started.htm?fbcl

4. DỰ ÁN XÂY DỰNG

- Trong 6 buổi thực hành (30 tiết), sinh viên sẽ thực hành xây dựng website quản lý bán hàng.

MỤC LỤC

HƯỚNG DẪN CHUNG	1
1. CHÍNH SÁCH CỦA MÔN HỌC.....	1
2. PHẦN MỀM PHỤC VỤ	1
3. TÀI LIỆU THAM KHẢO	1
4. DỰ ÁN XÂY DỰNG	1
HƯỚNG DẪN THỰC HÀNH BUỔI SỐ 1	4
1. Hướng dẫn thực hành: Xây dựng ứng dụng Asp.net MVC – CSDL SQL Server ..4	
1.1. Phân tích hệ thống Website bán hàng và xây dựng cơ sở dữ liệu	4
1.2. Tạo mới ứng dụng Asp.net MVC	8
1.3. Kết nối ứng dụng Asp.net MVC với CSDL sử dụng Entity Framework	18
1.4. Xây dựng các trang quản lý dữ liệu.....	24
HƯỚNG DẪN THỰC HÀNH BUỔI SỐ 2	34
2. Hướng dẫn thực hành: Xây dựng chức năng nâng cao cho trang quản trị	34
2.1. Hoàn thiện các menu quản lý dữ liệu	34
2.2. Xây dựng chức năng tìm kiếm	39
2.2. Xây dựng kết hợp tìm kiếm + phân trang.....	41
HƯỚNG DẪN THỰC HÀNH BUỔI SỐ 3	47
3. Hướng dẫn thực hành: Xây dựng chức năng nâng cao cho trang quản trị (tiếp) .47	
3.1. Validate form khi nhập dữ liệu.....	47
3.2. Tích hợp CKeditor và Ckfinder vào dự án	49
3.3. Thực hiện thêm mới các sản phẩm có kèm hình ảnh sản phẩm và mô tả.	53
HƯỚNG DẪN THỰC HÀNH BUỔI SỐ 4	60
4. Hướng dẫn thực hành: Hoàn thiện trang quản trị	60
4.1. Xây dựng trang đăng nhập cho website quản trị sử dụng session.....	60
4.2. Xây dựng chức năng đăng xuất tài khoản hiện tại	64
4.3. Mã hoá mật khẩu khi lưu vào database	64

4.4. Hoàn thiện website quản lý	68
HƯỚNG DẪN THỰC HÀNH BUỔI SỐ 5	69
5. Xây dựng trang dành cho người dùng	69
5.1. Xây dựng trang dành cho người dùng	69
5.2. Sử dụng PartialView để hiển thị sản phẩm	71
5.3. Xây dựng trang chi tiết sản phẩm.....	77
HƯỚNG DẪN THỰC HÀNH BUỔI SỐ 6	80
6. Xây dựng chức năng giỏ hàng	80
6.1. Xây dựng class giỏ hàng, các phương thức xử lý giỏ hàng.....	80
6.2. Xây dựng trang giỏ hàng partial, giỏ hàng	84
6.3. Xây dựng chức năng chỉnh sửa giỏ hàng và đặt hàng	88

HƯỚNG DẪN THỰC HÀNH BUỔI SỐ 1

I. Mục tiêu

- Nắm được hệ thống website quản lý bán hàng
- Xây dựng được cơ sở dữ liệu website quản lý bán hàng
- Biết cách tạo Template ASP.NET MVC Project; cấu trúc một ASP.NET MVC Project và chạy ứng dụng trên môi trường Localhost.
- Biết cách kết nối CSDL với project thông qua Entity Framework
- Xây dựng và nắm được các phương thức Index, Create, Details, Edit và Delete khi thao tác với dữ liệu.

II. Nội dung

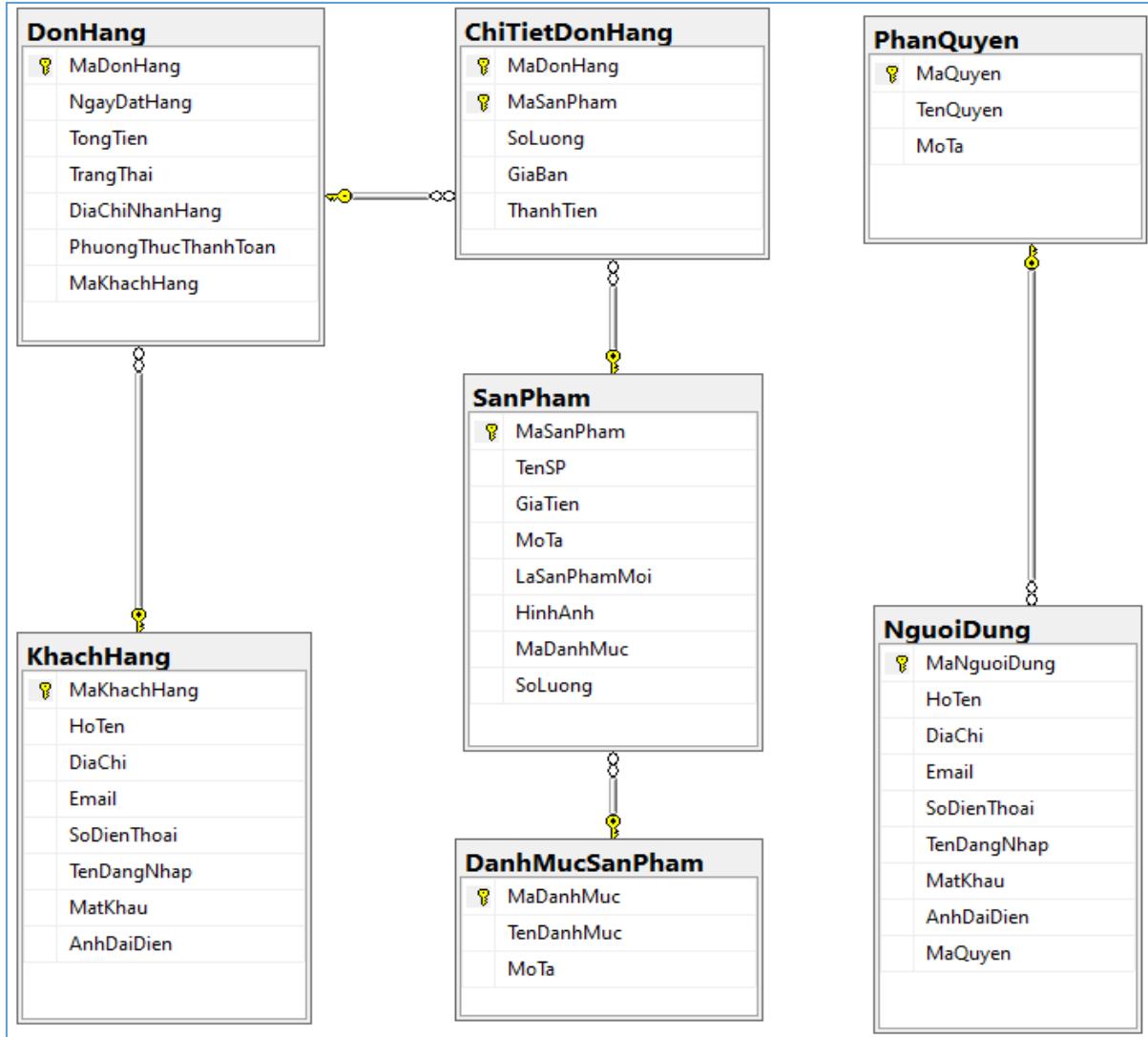
- Phân tích hệ thống,
- Tạo cơ sở dữ liệu sau khi phân tích
- Tạo project, phân chia quản lý trong project
- Kết nối CSDL với project
- Xây dựng các trang quản trị của website bán hàng:
 - + Quản lý Danh mục sản phẩm, Quản lý Sản phẩm , Quản lý phân quyền, Quản lý người dùng (Thêm, xóa, sửa và xem chi tiết dữ liệu)

1. Hướng dẫn thực hành: Xây dựng ứng dụng Asp.net MVC – CSDL SQL Server

1.1. Phân tích hệ thống Website bán hàng và xây dựng cơ sở dữ liệu

- CSDL mẫu cho dự án website bán hàng, gồm các bảng:
 - + NguoiDung: Bảng lưu trữ thông tin về người dùng trong hệ thống.
 - + PhanQuyen: Bảng lưu trữ thông tin về các quyền của người dùng trong hệ thống.
 - + DanhMucSanPham: Bảng lưu trữ thông tin về danh mục của các sản phẩm trong hệ thống.
 - + SanPham: Bảng lưu trữ thông tin về sản phẩm trong hệ thống
 - + KhachHang: Bảng lưu trữ thông tin về khách hàng tham gia mua hàng của hệ thống.
 - + DonHang: Bảng lưu trữ thông tin về đơn hàng của khách hàng.
 - + ChiTietDonHang: Bảng lưu trữ thông tin chi tiết đơn hàng của đơn hàng. 1 đơn hàng gồm nhiều chi tiết đơn hàng.

- Diagram:



- Mô tả chi tiết thiết kế của các bảng:

a. PhanQuyen

Tên cột	Kiểu dữ liệu	Mô tả
MaQuyen	INT	IDENTITY(1,1), PRIMARY KEY
TenQuyen	NVARCHAR(50)	NOT NULL
MoTa	NVARCHAR(200)	NULL

b. NguoiDung

Tên cột	Kiểu dữ liệu	Mô tả
MaNguoiDung	INT	IDENTITY(1,1), PRIMARY KEY
HoTen	NVARCHAR(50)	NOT NULL
DiaChi	NVARCHAR(100)	NULL
Email	NVARCHAR(50)	NOT NULL
SoDienThoai	NCHAR(10)	NOT NULL
TenDangNhap	NVARCHAR(50)	NOT NULL
MatKhau	NVARCHAR(50)	NOT NULL
AnhDaiDien	NVARCHAR(50)	NULL
MaQuyen	INT	FOREIGN KEY REFERENCES PhanQuyen(MaQuyen)

c. DanhMucSanPham

Tên cột	Kiểu dữ liệu	Mô tả
MaDanhMuc	INT	IDENTITY(1,1), PRIMARY KEY
TenDanhMuc	NVARCHAR(50)	NOT NULL
MoTa	NVARCHAR(200)	NULL

d. SanPham

Tên cột	Kiểu dữ liệu	Mô tả
MaSanPham	INT	IDENTITY(1,1), PRIMARY KEY
TenSP	NVARCHAR(50)	NOT NULL
GiaTien	DECIMAL(18, 0)	NOT NULL
MoTa	NVARCHAR(500)	NOT NULL
LaSanPhamMoi	BIT	DEFAULT 0
HinhAnh	NVARCHAR(50)	NULL
SoLuong	INT	NOT NULL
MaDanhMuc	INT	FOREIGN KEY REFERENCES DanhMucSanPham(MaDanhMuc)

e. DonHang

Tên cột	Kiểu dữ liệu	Mô tả
MaDonHang	INT	IDENTITY(1,1), PRIMARY KEY
NgayDatHang	DATETIME	NOT NULL
TongTien	DECIMAL(18, 0)	NOT NULL
TrangThai	NVARCHAR(50)	NOT NULL
DiaChiNhanHang	NVARCHAR(100)	NOT NULL
PhuongThucThanhToan	NVARCHAR(50)	NOT NULL
MaKhachHang	INT	FOREIGN KEY REFERENCES KhachHang(MaKhachHang)

f. ChiTietDonHang

Tên cột	Kiểu dữ liệu	Mô tả
MaDonHang	INT	FOREIGN KEY REFERENCES DonHang(MaDonHang)
MaSanPham	INT	INT FOREIGN KEY REFERENCES SanPham(MaSanPham)
SoLuong	INT	NOT NULL
GiaBan	DECIMAL(18, 0)	NOT NULL
ThanhTien	DECIMAL(18, 0)	NOT NULL
PRIMARY KEY(MaDonHang, MaSanPham)		

g. KhachHang

Tên cột	Kiểu dữ liệu	Mô tả
MaKhachHang	INT	IDENTITY(1,1), PRIMARY KEY
HoTen	NVARCHAR(50)	NOT NULL
DiaChi	NVARCHAR(100)	NULL
Email	NVARCHAR(50)	NOT NULL
SoDienThoai	NCHAR(10)	NOT NULL
TenDangNhap	NVARCHAR(50)	NOT NULL
MatKhau	NVARCHAR(50)	NOT NULL

- Tiến hành tạo CSDL gồm các bảng trên, CSDL Lưu trong thư mục HTenSinhVien_QLBanHangMVC\Database với tên là HTenSinhVien_QLBanHangMVC (ví dụ NVCuong_QLBanHangMVC)

B1. Tạo mới cơ sở dữ liệu (database)

B2. Thiết kế bảng (table)

- Tạo mới bảng (new table)
- Khai báo cột (Column)
- Chọn kiểu dữ liệu (Data Type) tương ứng cho cột.
 - o Char(), Nvarchar(), bigint, int, varchar(), datetime, bit, float,...
- Thiết lập khóa chính (primary key), khóa ngoại, check.....
- Thiết lập thuộc tính tự động tăng cho khóa chính của các bảng cần thiết
- Lưu bảng

B3. Thiết lập ràng buộc (diagram)

B4. Nhập dữ liệu mẫu

- Sau khi thực hiện thiết kế Database xong, thực hiện nhập dữ liệu mẫu cho các bảng:

+ PhanQuyen:

MaQuyen	TenQuyen	MoTa
1	Admin	NULL
2	Manager	NULL

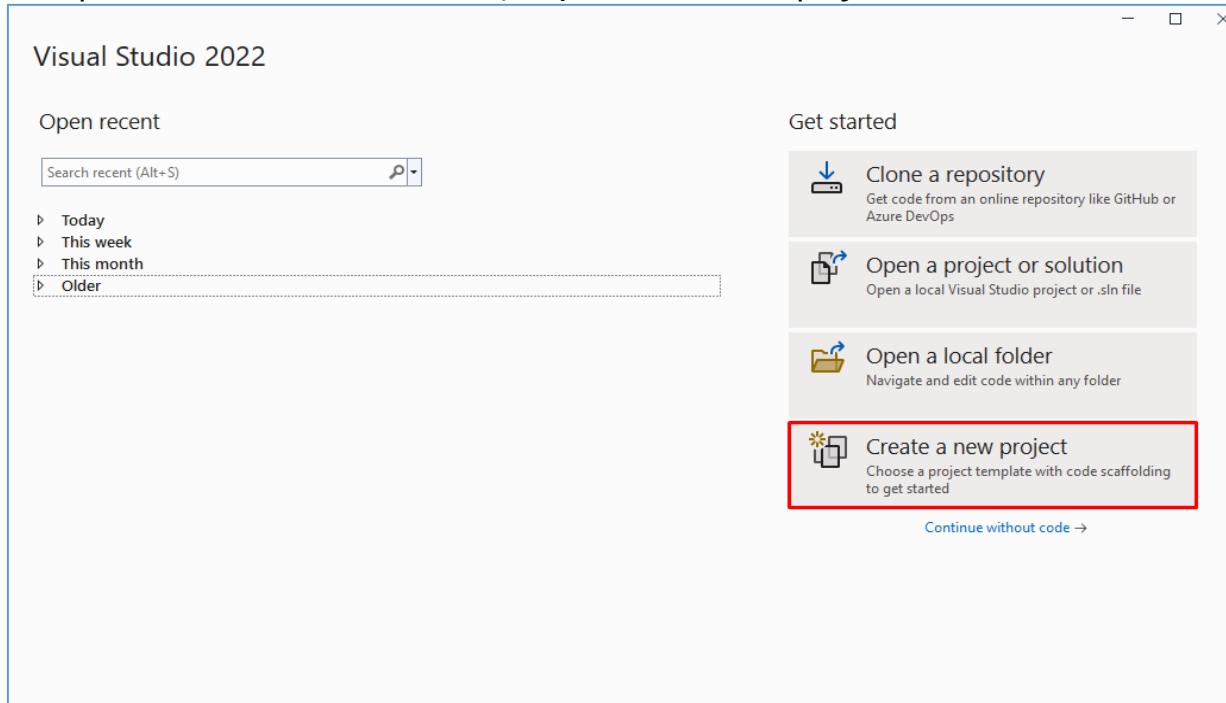
+ DanhMucSanPham

MaDanhMuc	TenDanhMuc	MoTa
1	Xiaomi	NULL
2	Iphone	NULL
3	Nokia	NULL
4	Samsung	NULL
5	LG	NULL
6	Oppo	NULL
7	Sony	NULL

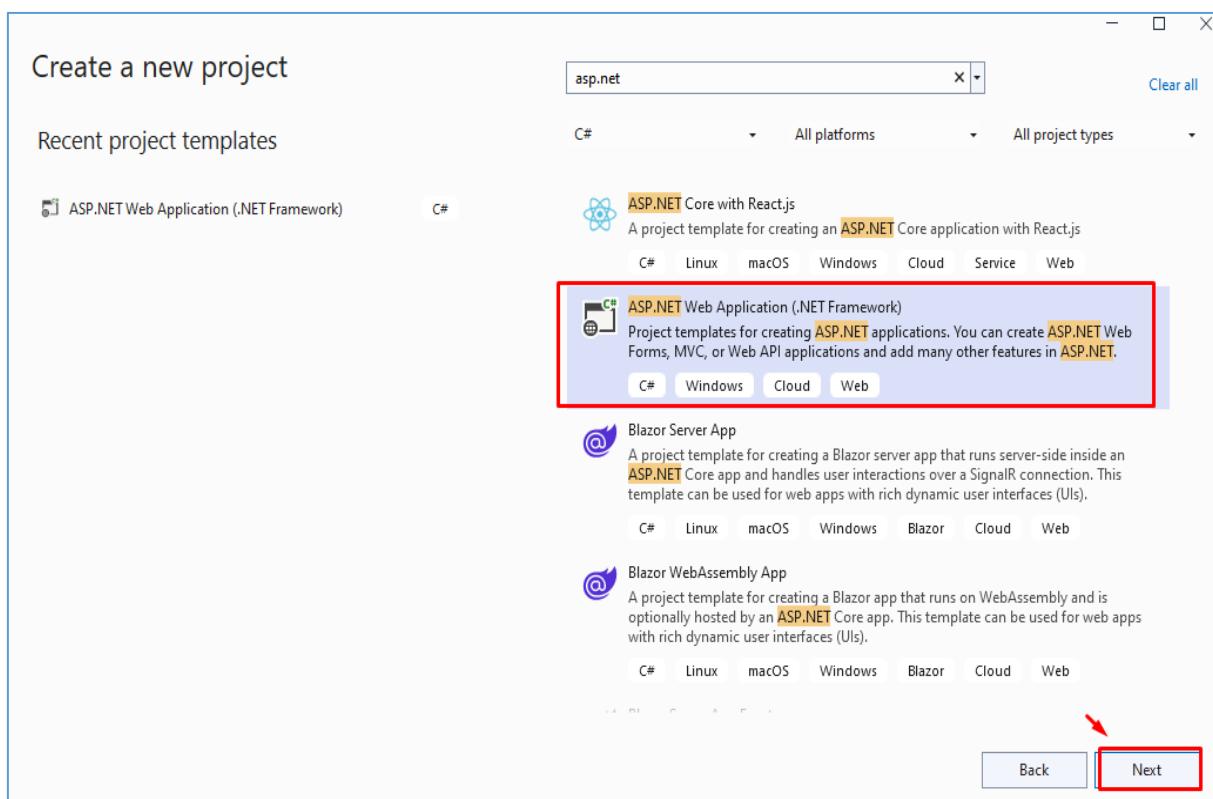
1.2. Tạo mới ứng dụng Asp.net MVC

Tạo project lưu trữ trong thư mục HTenSinhVien_QLBanHangMVC (ví dụ NVCuong_QLBanHangMVC) **trên máy tính:**

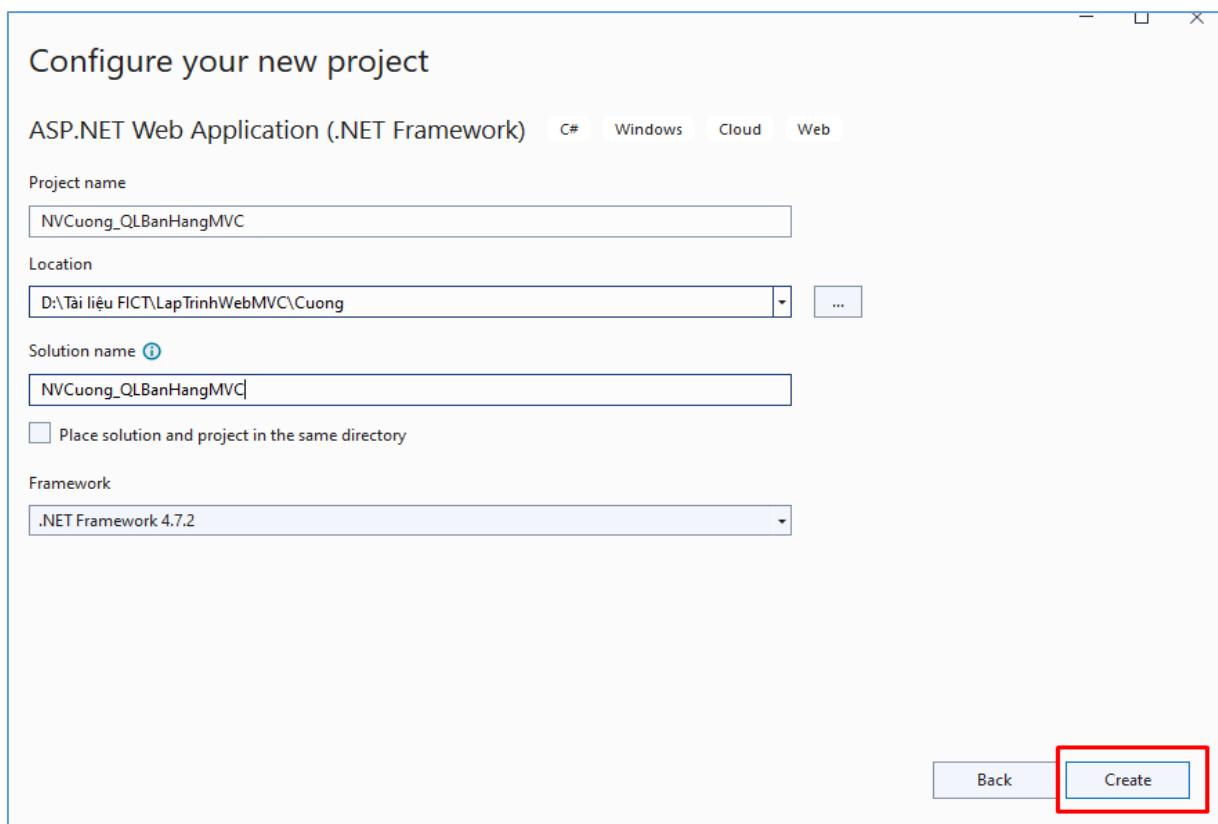
- Mở phần mềm Visual Studio 2022, chọn Create a new project:



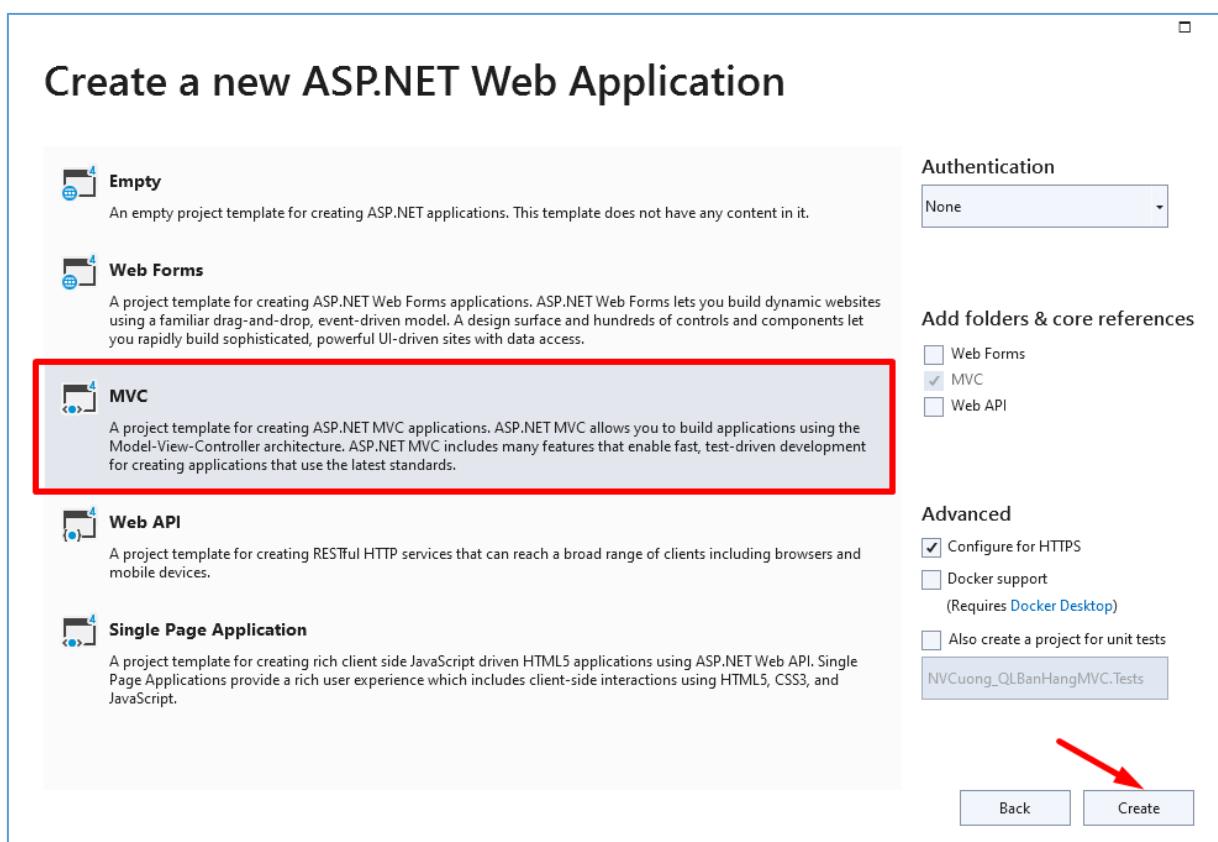
- Chọn ASP.NET Web Application (.NET Framework)



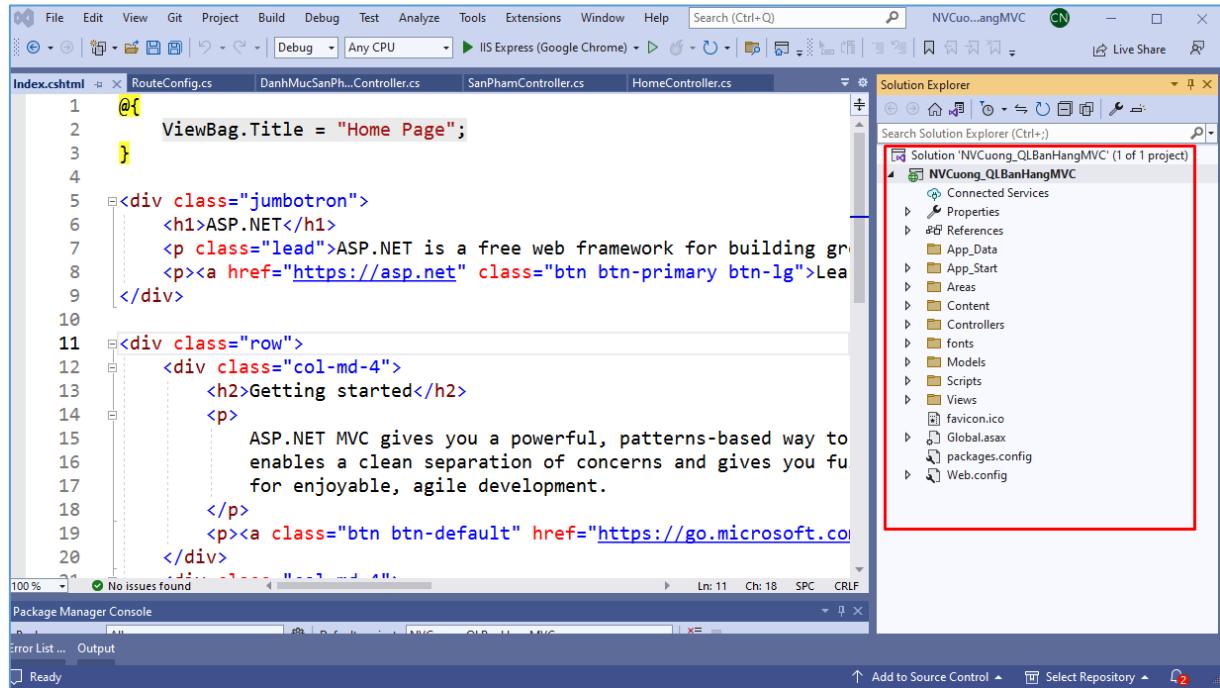
- Đặt tên theo cú pháp HTenSinhVien_QLBanHangMVC (ví dụ NVCuong_QLBanHangMVC), sau đó chọn Create.



- Chọn MVC sau đó Create:



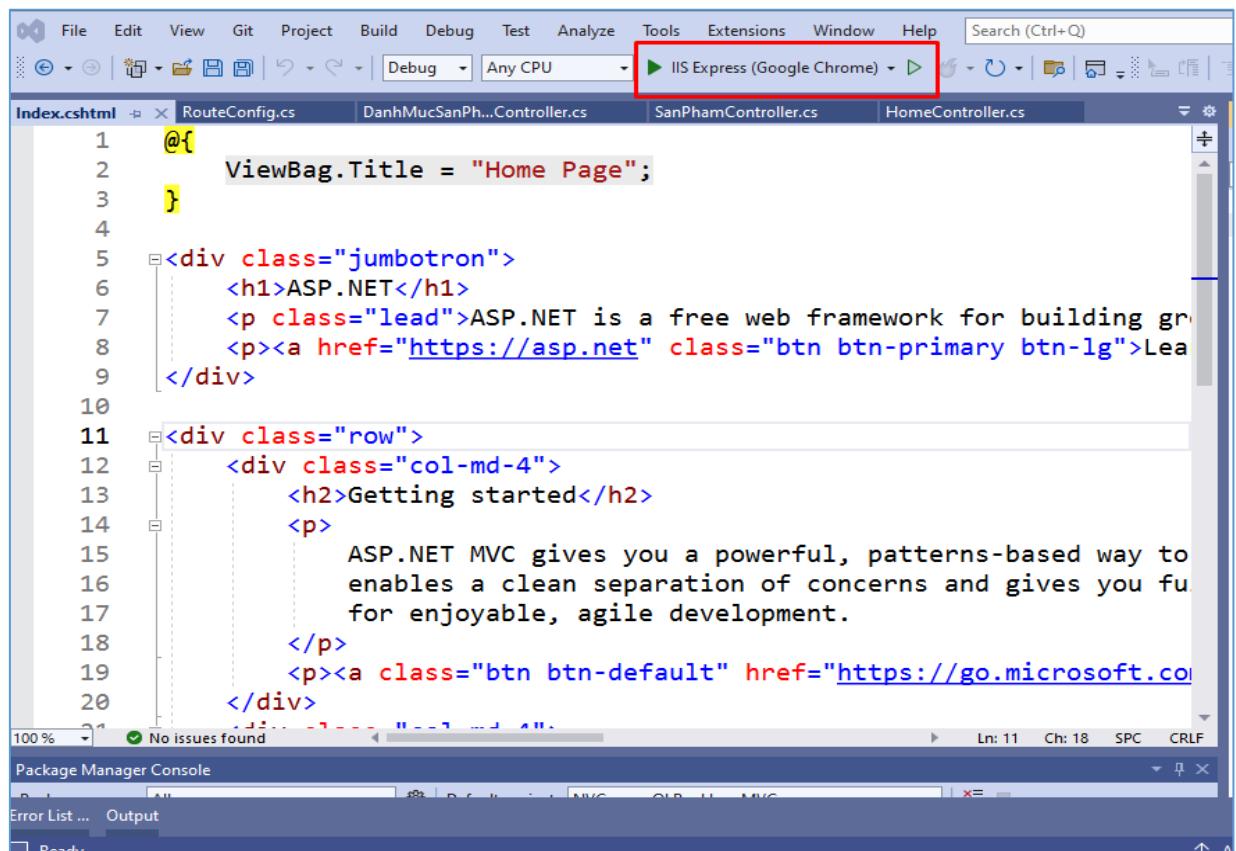
- Tạo project thành công:



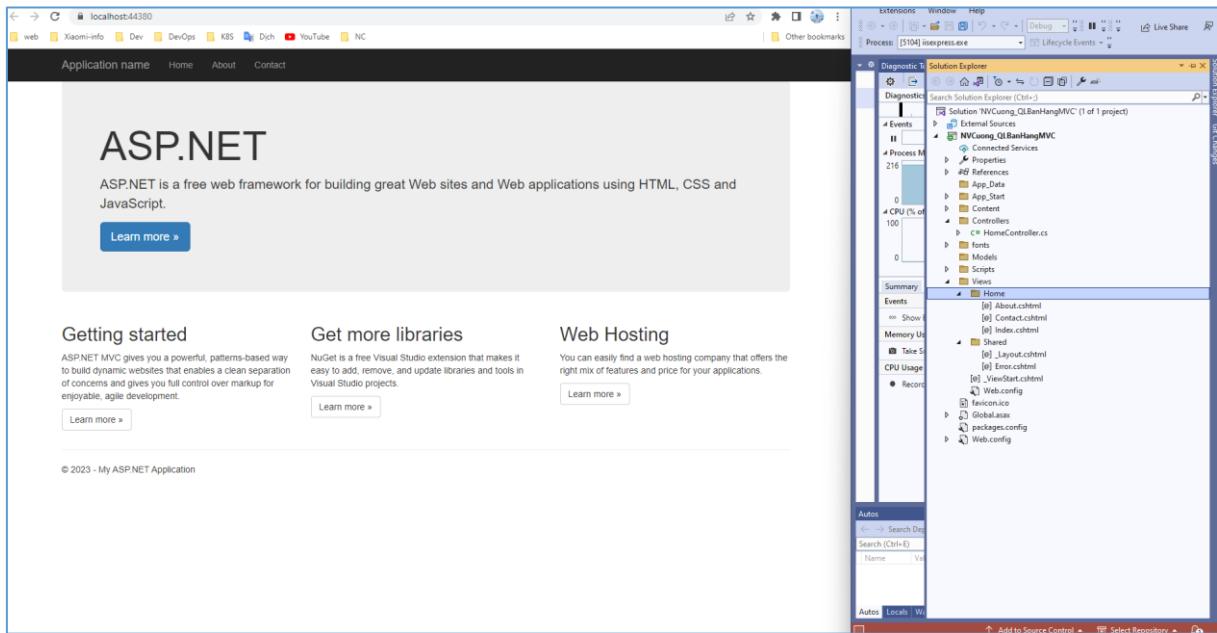
- Dự án sẽ gồm:

- + 1 trang dành cho quản trị viên dùng để quản trị website: Đăng sản phẩm, thêm người dùng, duyệt đơn hàng, xem thống kê...
- + 1 trang dành cho khách hàng: trang hiển thị các sản phẩm để khách hàng xem và mua hàng

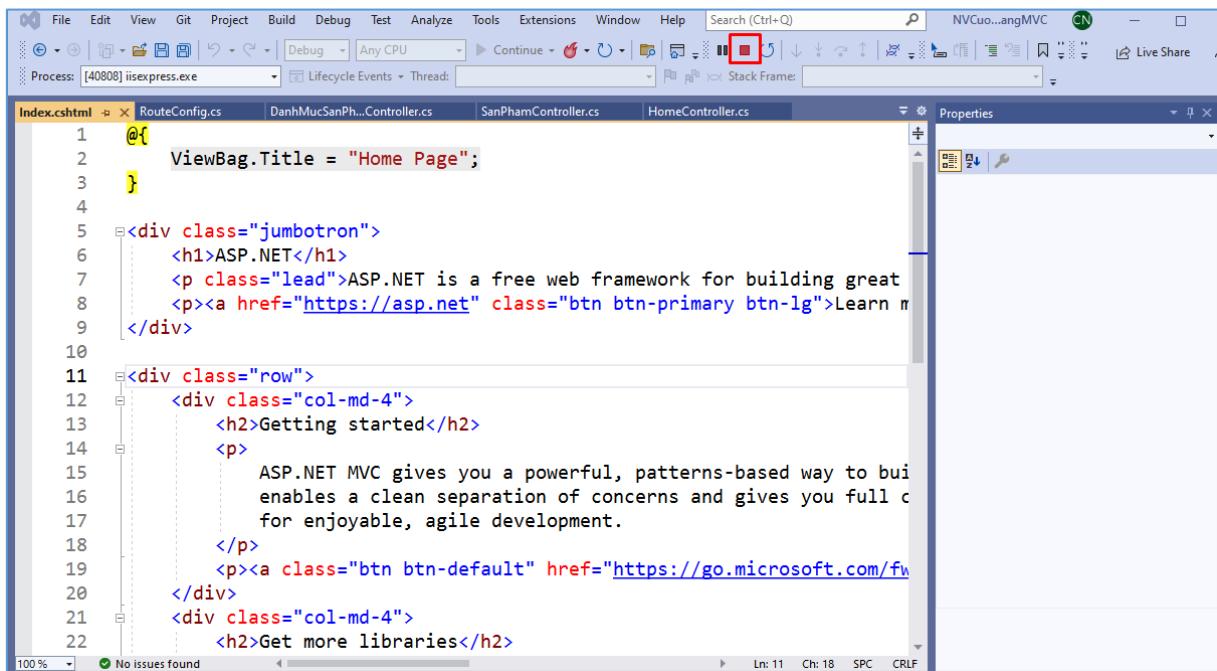
- Tiến hành chạy dự án:



- Đây sẽ là trang sử dụng để thiết kế dành cho khách hàng.

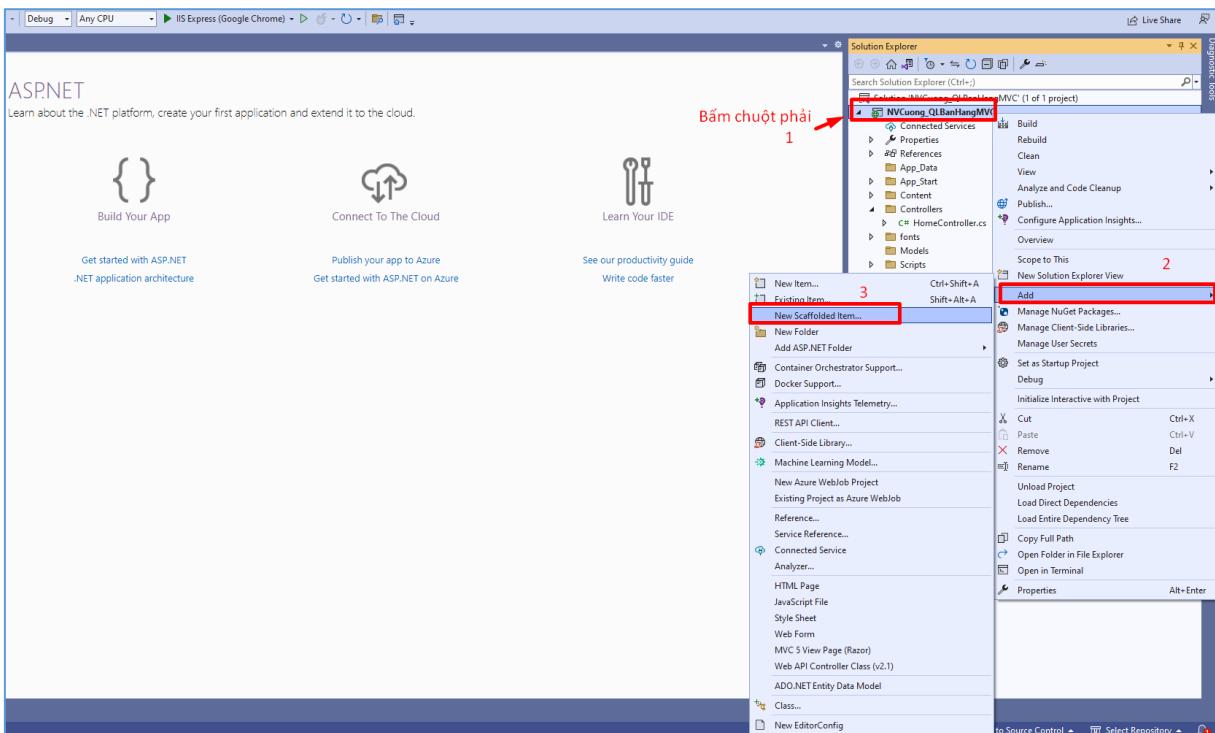


+ Dừng chạy website :

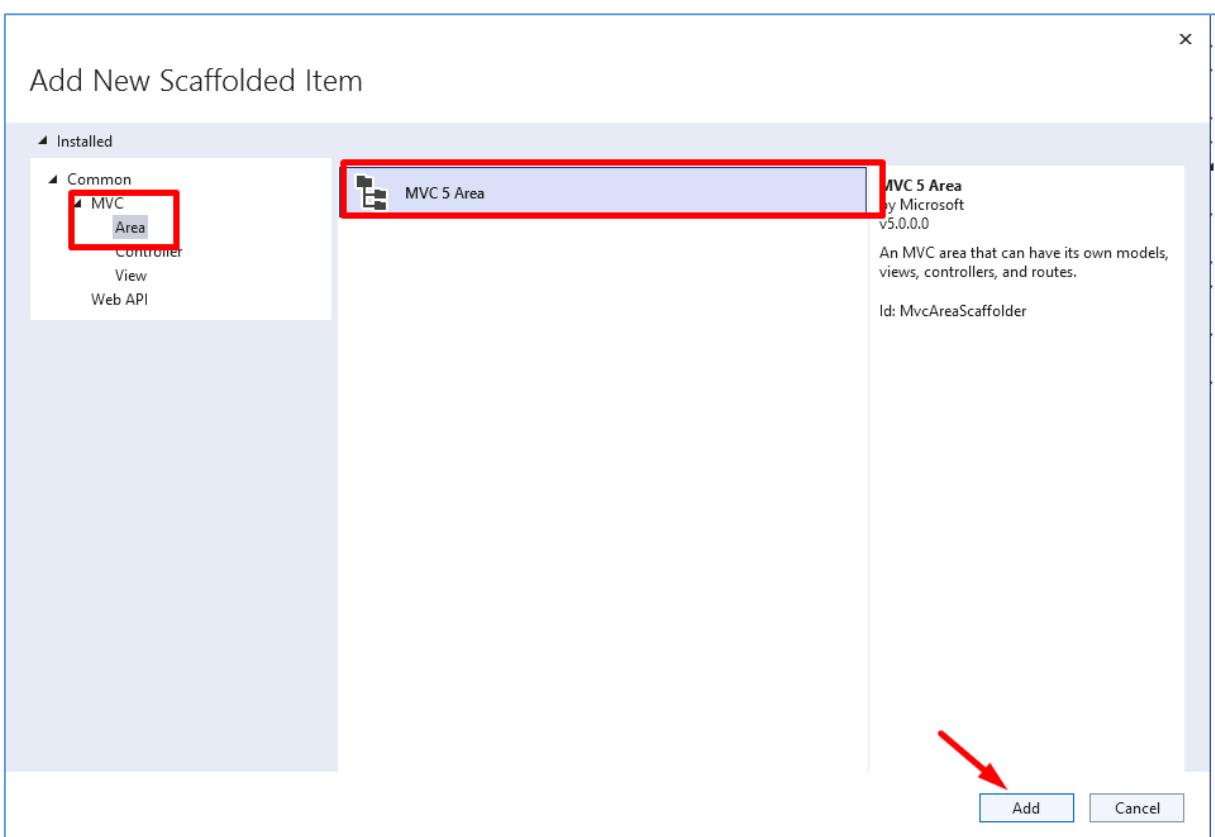


- Về trang dành cho quản trị sẽ tiến hành tạo Area:

+ Tại dự án: bấm chuột phải chọn Add => New Scaffolded item...



- Tiếp theo:



- Nhập Admin -> Add

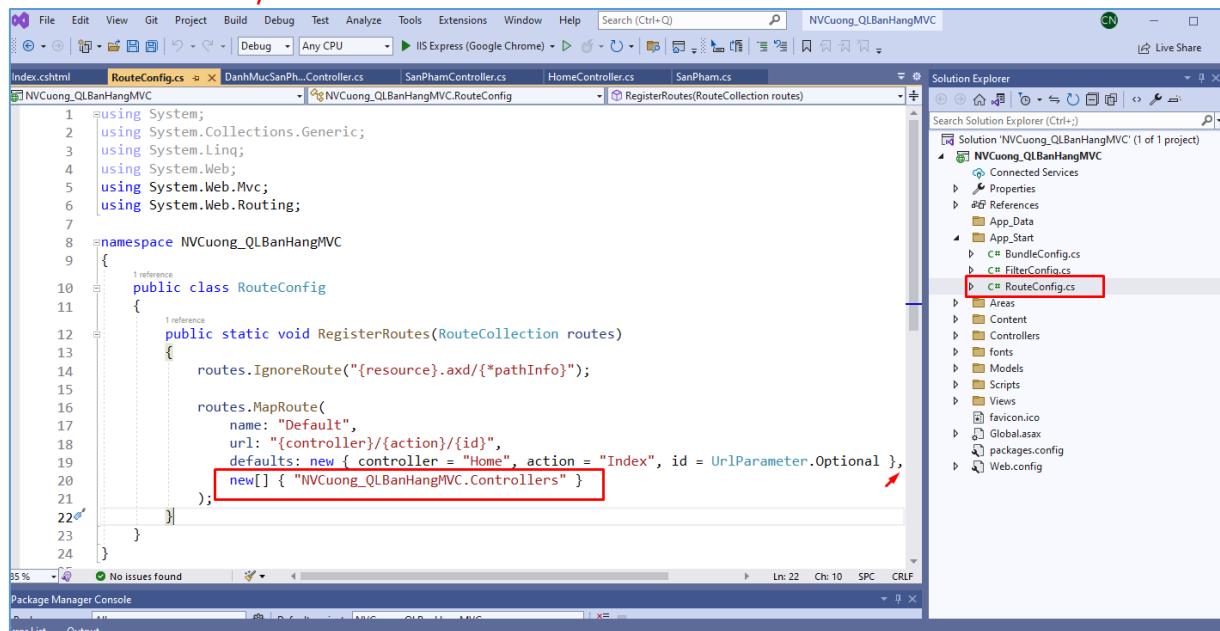


- Tạo Area Admin thành công, chúng ta sẽ thực hiện code trang quản trị tại thư mục /Area/Admin.

- Tiếp theo, Mở **App_Start -> RouteConfig.cs** thêm dòng sau:

`new[] { "NVCuong_QLBanHangMVC.Controllers" }`

Thay **NVCuong_QLBanHangMVC** bằng tên dự án đang xây dựng. Chú ý trước dòng mới thêm có dấu , :

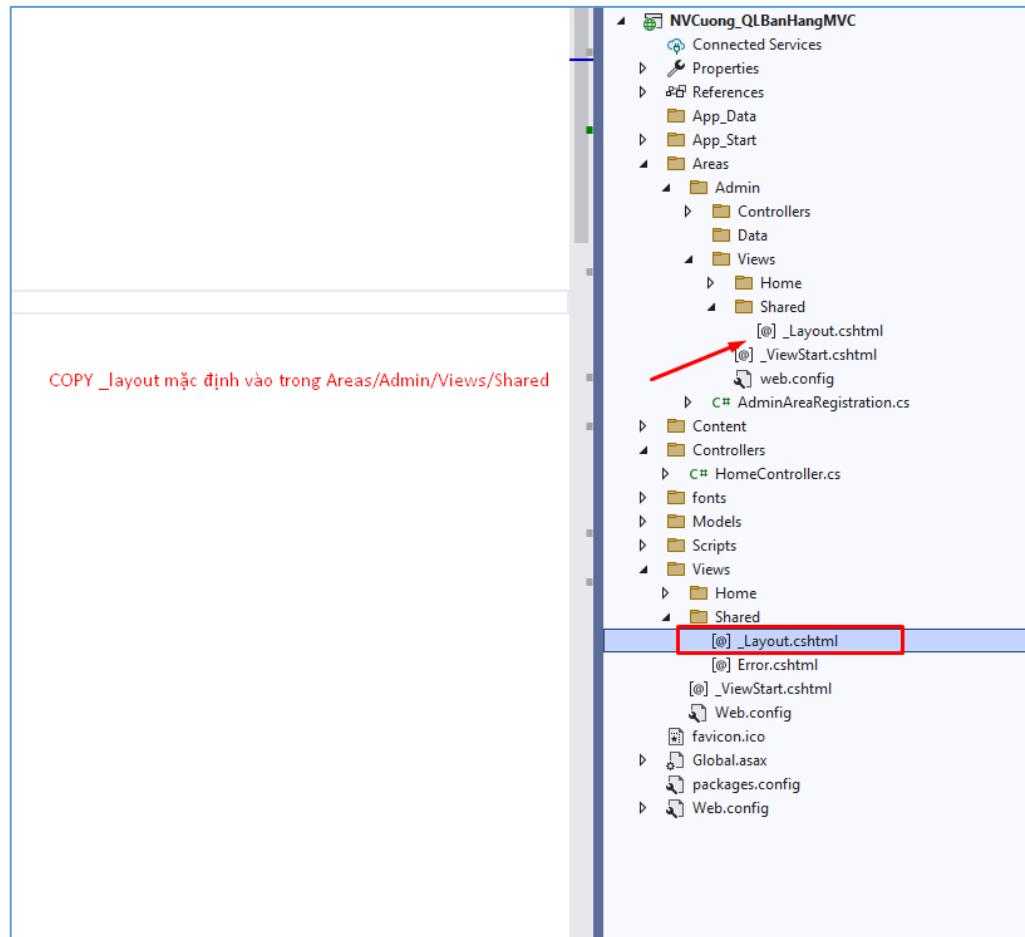


```
namespace NVCuong_QLBanHangMVC
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

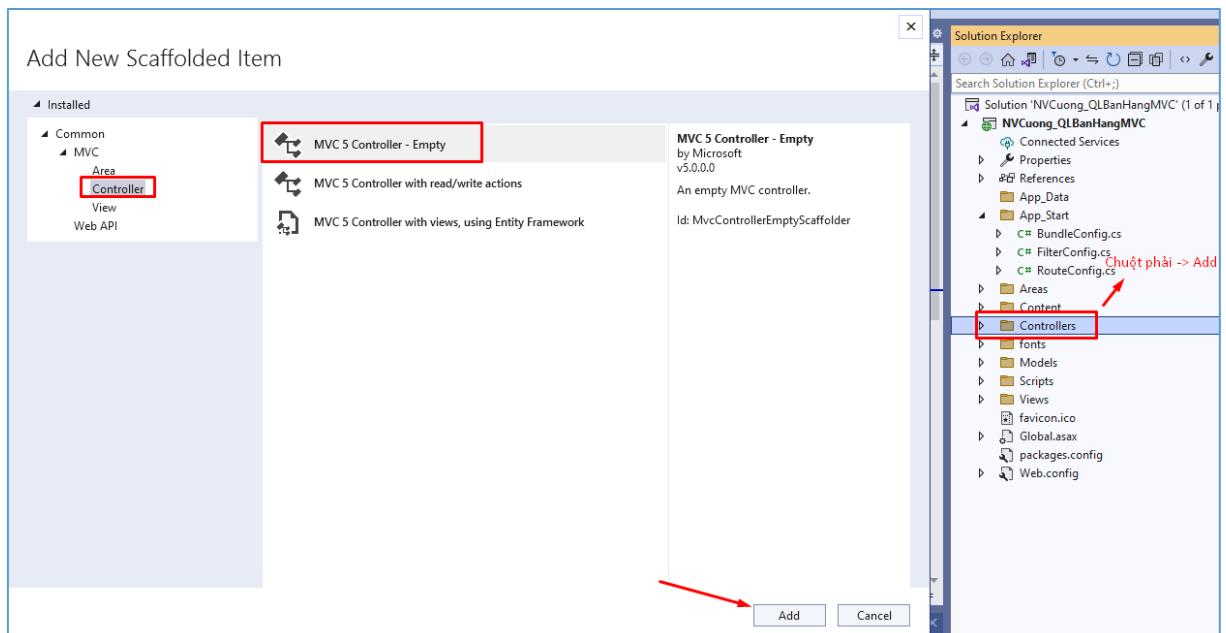
            routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{id}",
                defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional },
                new[] { "NVCuong_QLBanHangMVC.Controllers" }
            );
        }
    }
}
```

}

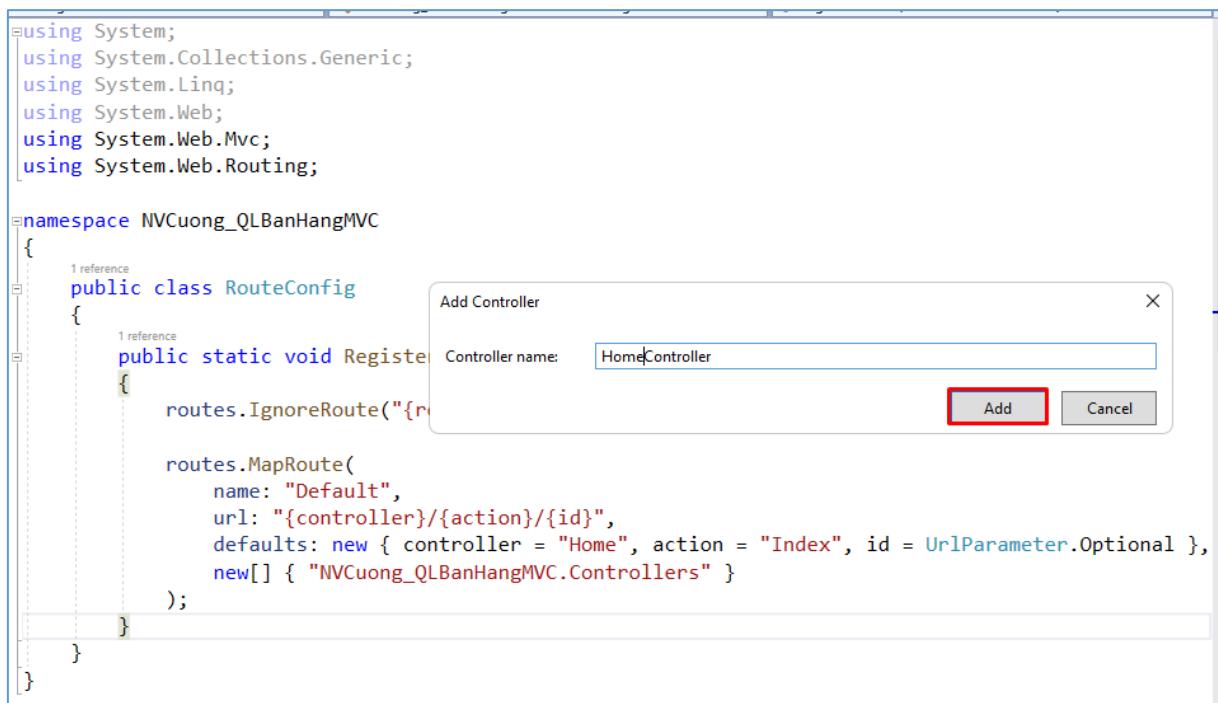
- Tạo layout dùng chung cho trang quản trị từ `_layout` mặc định:
+ Copy file `_layout` mặc định từ `Views/Shared/_Layout.cshtml` đến thư mục:
`Areas/Admin/Views/Shared/`



Tại: *Areas/Admin/Controllers/*. Thêm mới Controller Home.



- Controller Name: HomeController

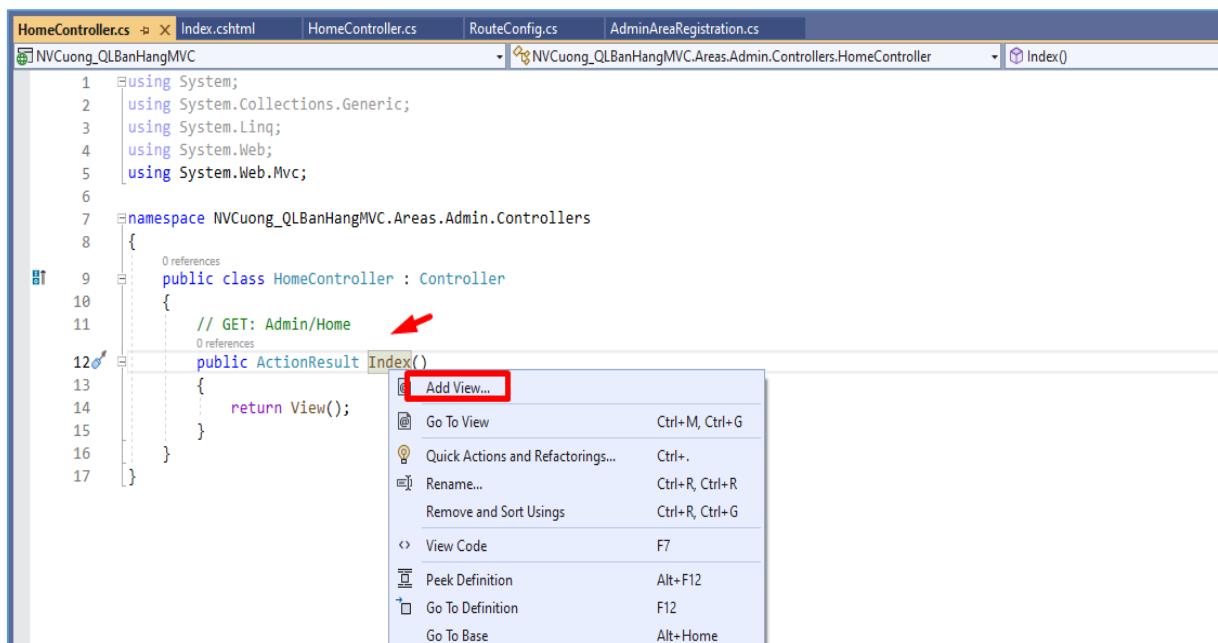


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Routing;

namespace NVCuong_QLBanHangMVC
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

            routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{id}",
                defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional },
                namespaces: new[] { "NVCuong_QLBanHangMVC.Controllers" }
            );
        }
    }
}
```

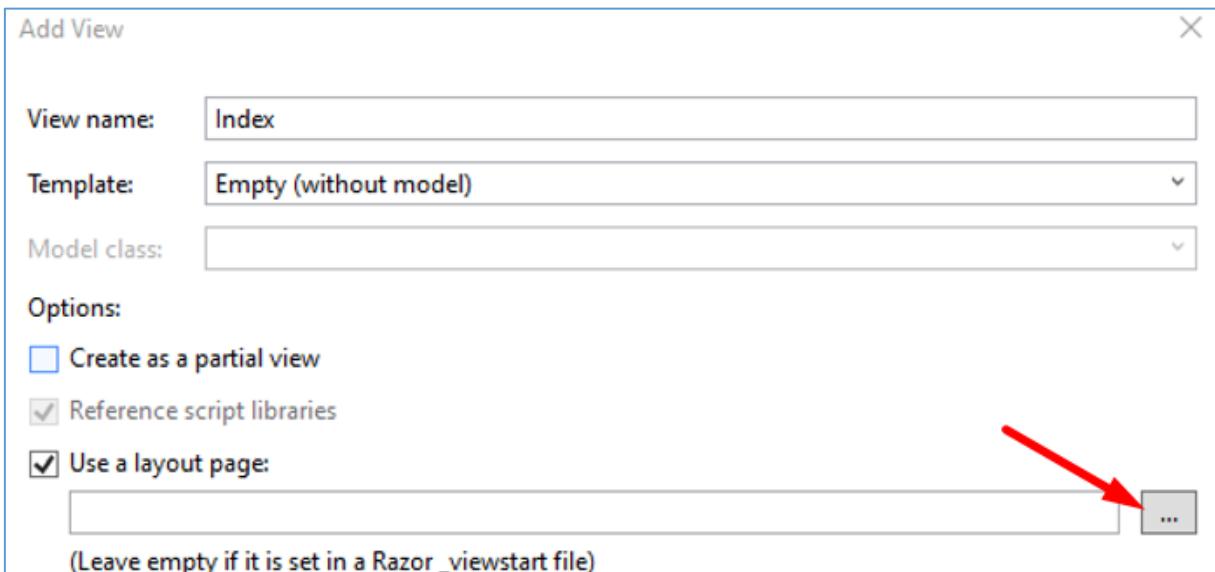
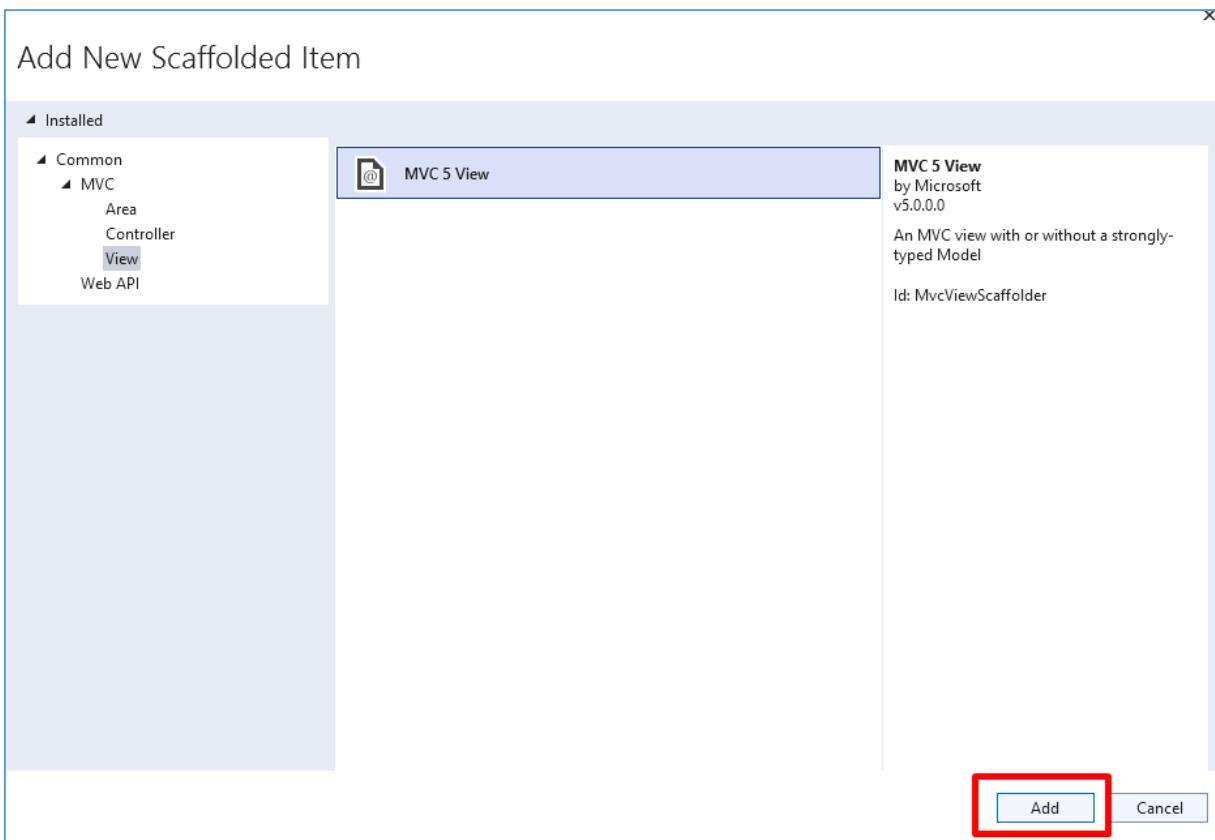
- Chuột phải tại Index chọn Add View:

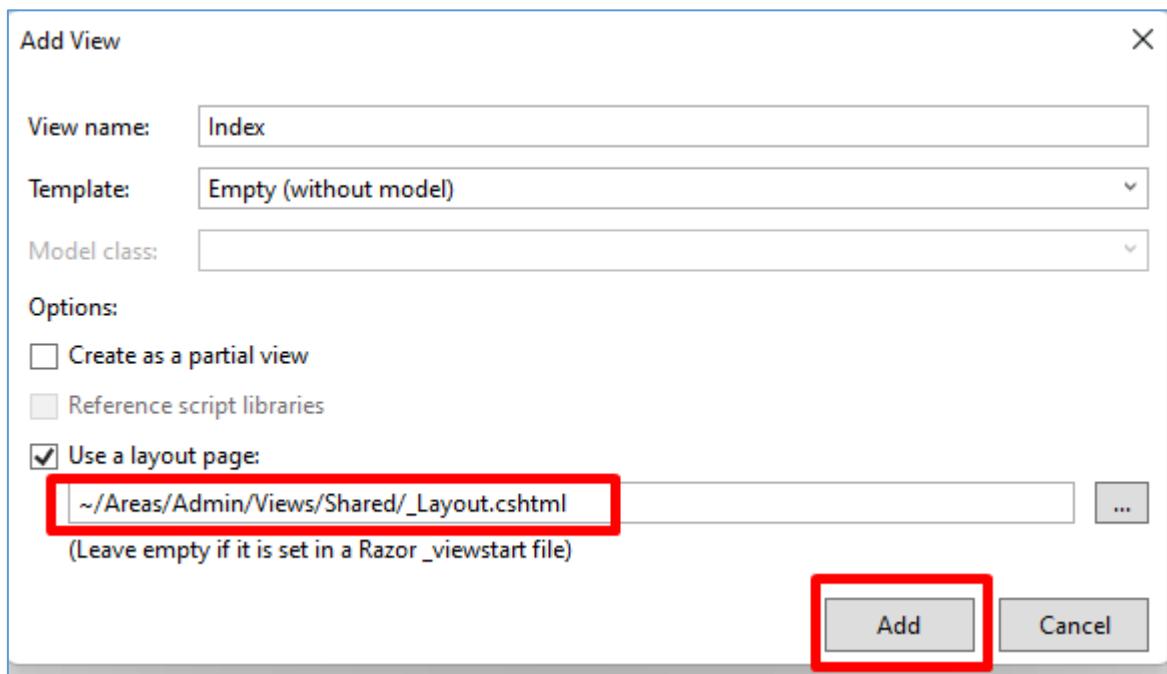
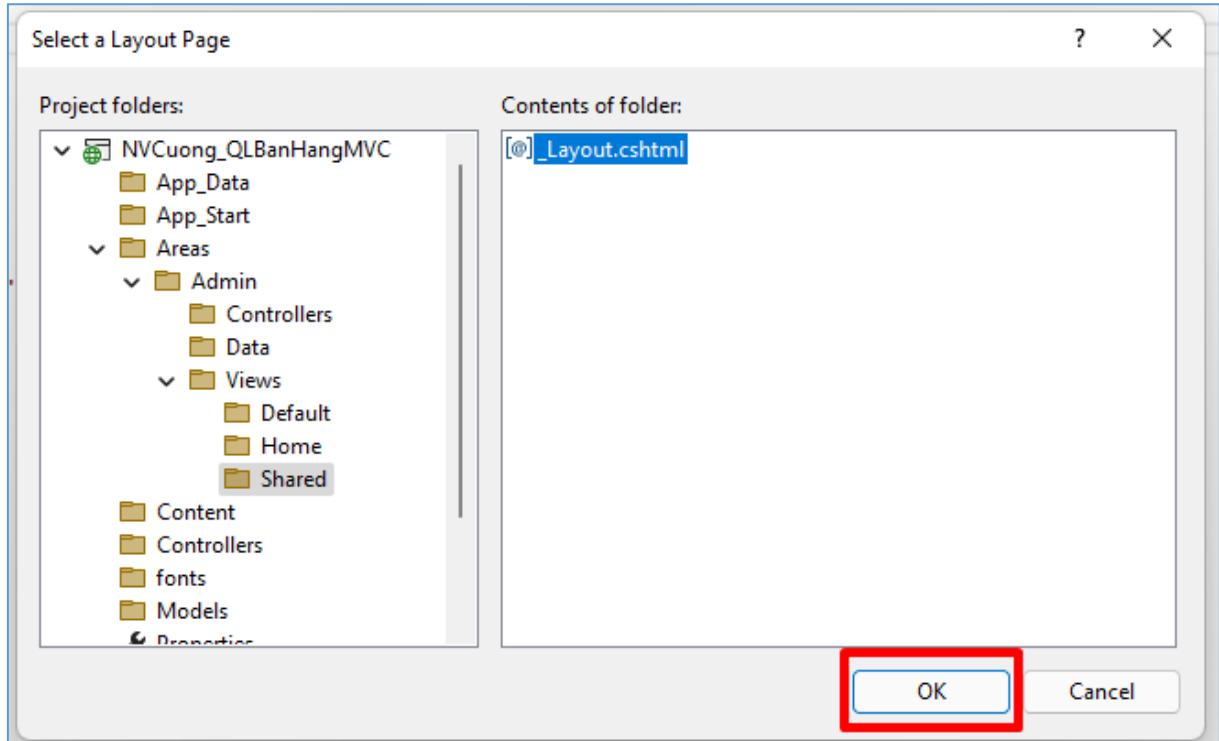


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

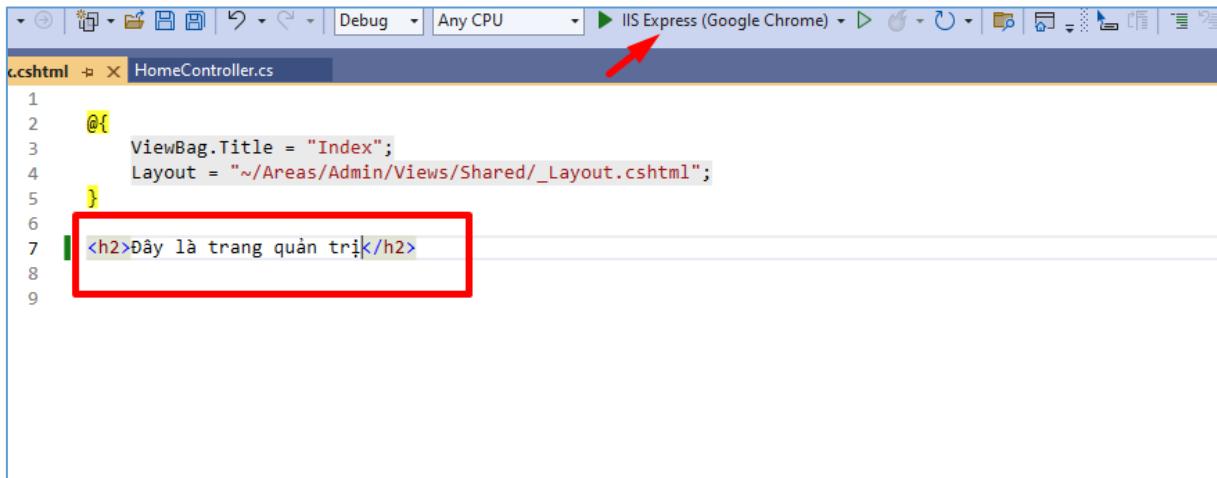
namespace NVCuong_QLBanHangMVC.Areas.Admin.Controllers
{
    public class HomeController : Controller
    {
        // GET: Admin/Home
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

Thực hành theo các bước sau:



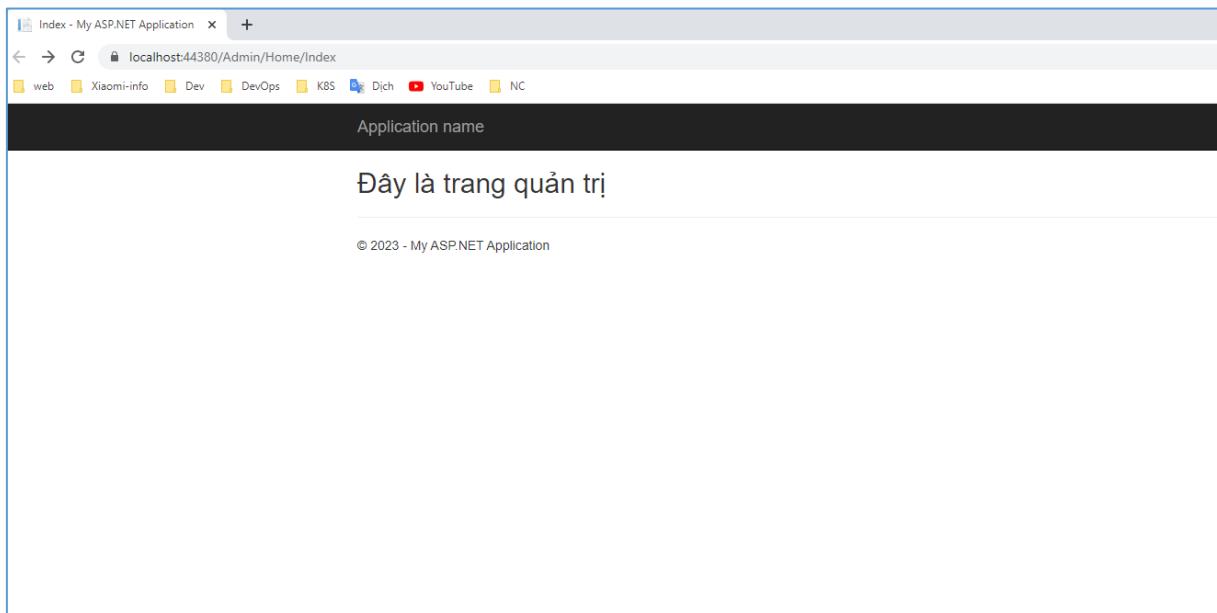


- Chạy website:



```
1
2  @{
3      ViewBag.Title = "Index";
4      Layout = "~/Areas/Admin/Views/Shared/_Layout.cshtml";
5  }
6
7  <h2>Đây là trang quản trị</h2>
8
9
```

- Hiển thị trang dành cho quản trị:



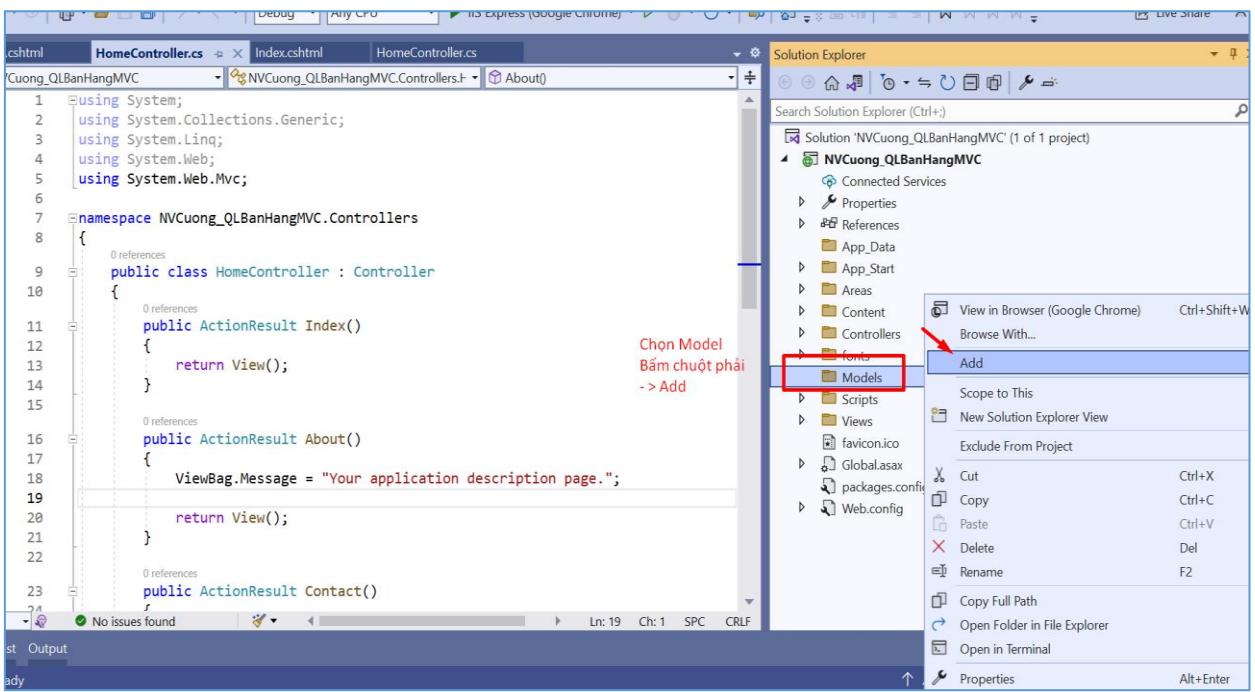
- Dừng chạy website:

- Bây giờ

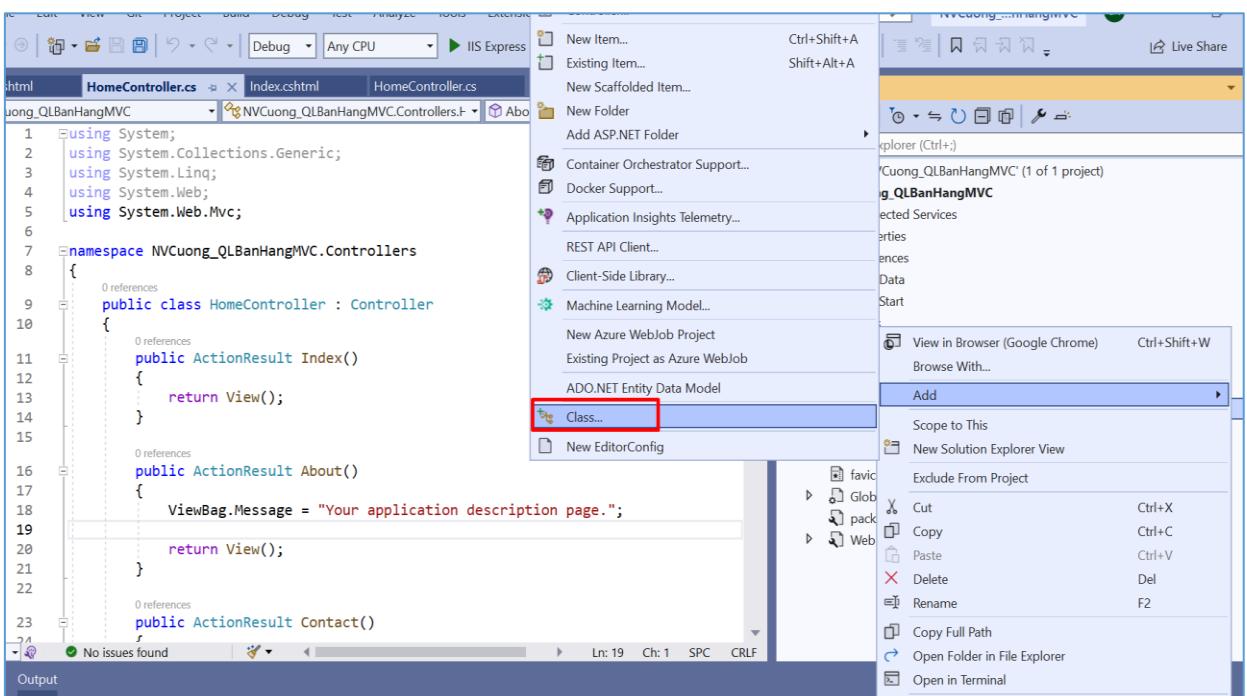
- + Trang chủ quản trị sẽ có địa chỉ là: **https://localhost:44380/Admin/Home**
 - + Trang chủ người dùng sẽ có địa chỉ là: **https://localhost:44380/Home**
- 44380 -> là port của website. Các máy tính có thể sẽ khác nhau.

1.3. Kết nối ứng dụng Asp.net MVC với CSDL sử dụng Entity Framework

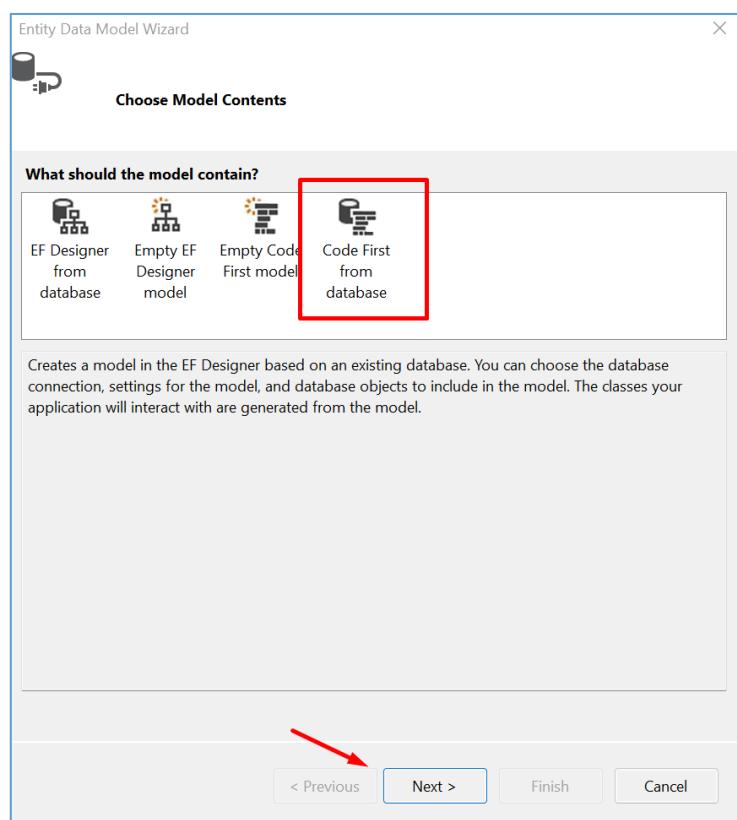
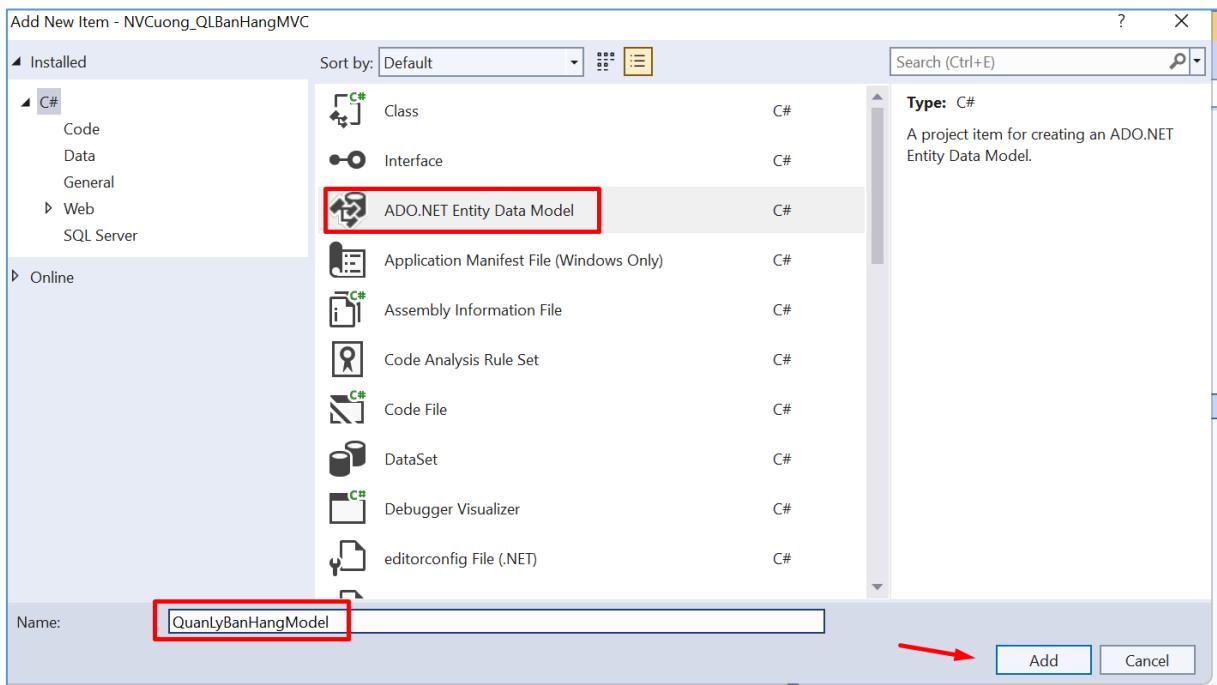
- Tiến hành sử dụng Entity Framework để kết nối tới Database đã xây dựng. Chọn Code first.
- Thực hiện như sau:
- + Chọn Models bấm chuột phải chọn Add:



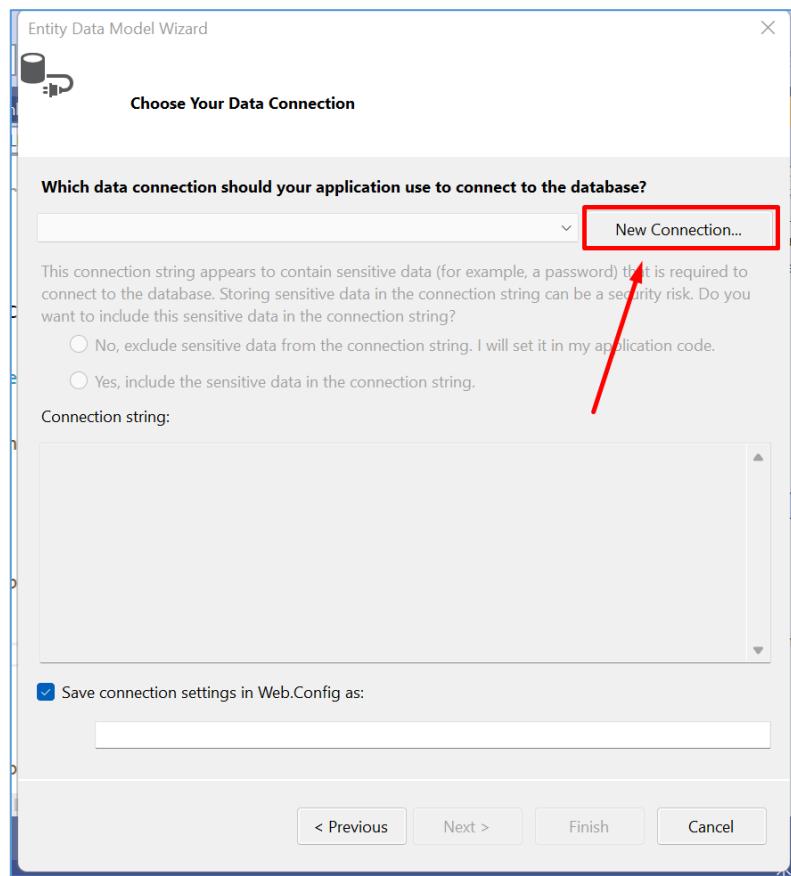
- Chọn class:



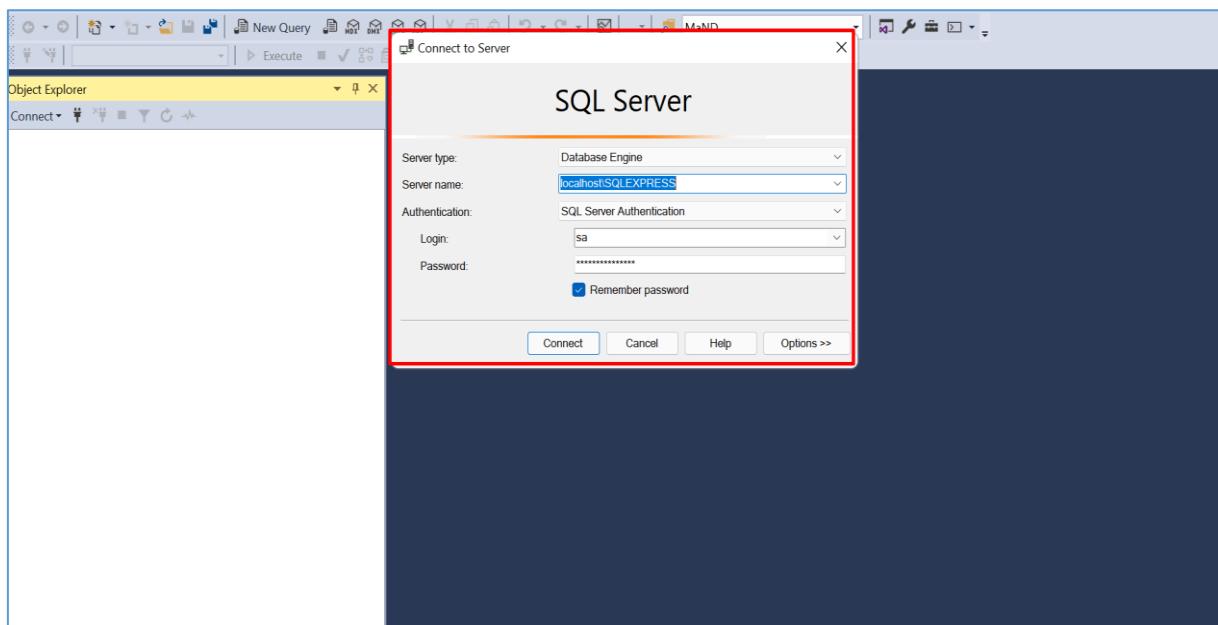
- Chọn Data -> ADO.NET Entity Data Model



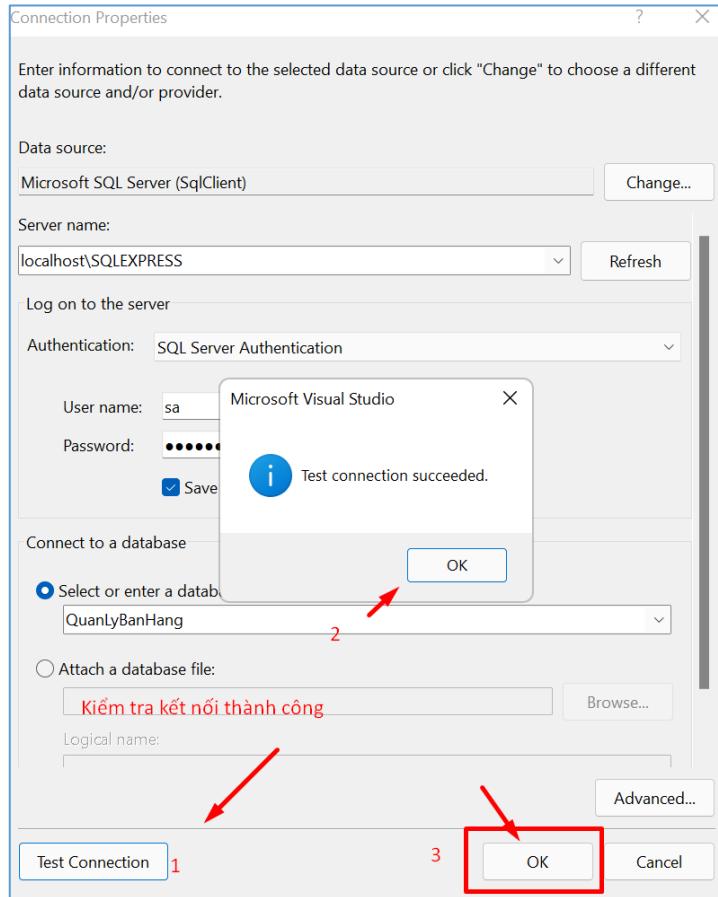
- Tạo mới kết nối (Đối với lần đầu tiên). New Connection. Từ những lần sau, chỉ cần lựa chọn chuỗi kết nối đã có (connection string).



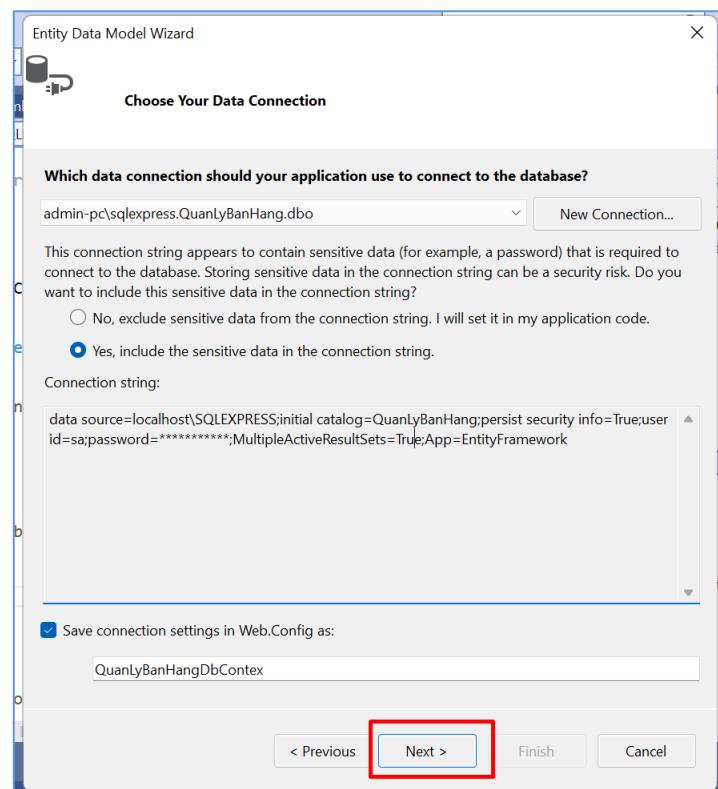
- Sử dụng các thông tin của sql server đã cài đặt:



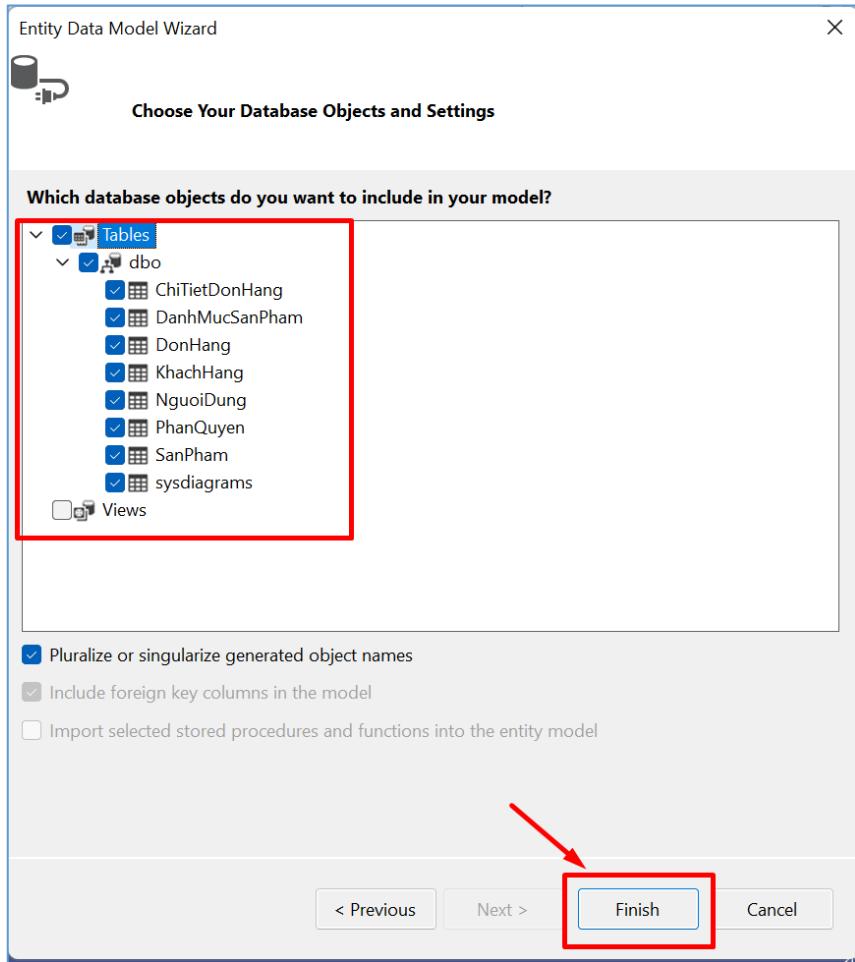
- Điền vào Connection Properties:



- Tiếp theo:



- Tiếp theo:



- Bây giờ dự án đã có các file .cs trong thư mục Models để định nghĩa, kết nối tới Database:

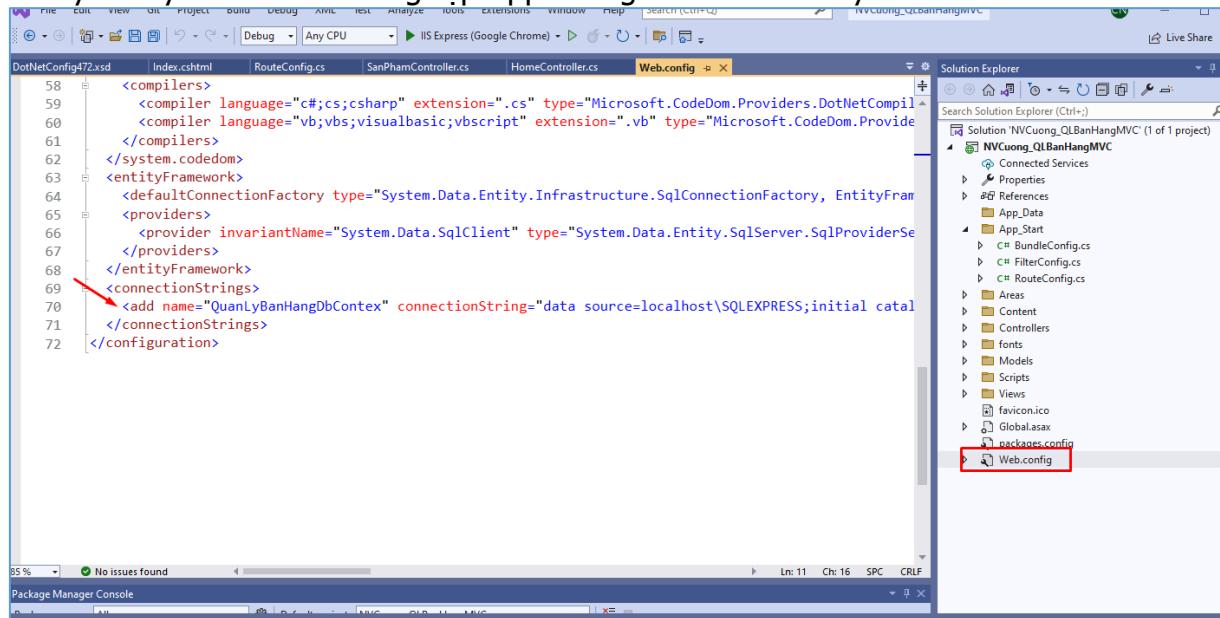
```
QuanLyBanHangModel.cs
1 using System;
2 using System.ComponentModel.DataAnnotations.Schema;
3 using System.Data.Entity;
4 using System.Linq;
5
6 namespace NVCuong_QLBanHangMVC.Models
7 {
8     public partial class QuanLyBanHangModel : DbContext
9     {
10         public QuanLyBanHangModel()
11             : base("name=QuanLyBanHangDbContext")
12         {
13         }
14
15         public virtual DbSet<ChiTietDonHang> ChiTietDonHangs { get; set; }
16         public virtual DbSet<DanhMucSanPham> DanhMucSanPhams { get; set; }
17         public virtual DbSet<DonHang> DonHangs { get; set; }
18         public virtual DbSet<KhachHang> KhachHangs { get; set; }
19         public virtual DbSet<NguoiDung> NguoiDungs { get; set; }
20         public virtual DbSet<PhanQuyen> PhanQuyens { get; set; }
21     }
22 }
```

Solution Explorer

- Properties
- References
- App_Data
- App_Start
- Areas
- Content
- Controllers
- fonts
- Models
 - ChiTietDonHang.cs
 - DanhMucSanPham.cs
 - DonHang.cs
 - KhachHang.cs
 - NguoiDung.cs
 - PhanQuyen.cs
 - QuanLyBanHangModel.cs
 - SanPham.cs
 - sysdiagram.cs
- Scripts
- Views
- favicon.ico
- Global.asax
- packages.config

- Các lưu ý khi thiết lập connection.

Chú ý: Thay đổi kết nối trong tệp App.config đối với các máy tính khác.



Mô tả: data source: Tên máy tính cần thay (Server name); catalog: Tên CSDL (database) cần kết nối.

```

<connectionStrings>
  <add name="QuanLyBanHangDbContex" connectionString="data
source=localhost\SQLEXPRESS;initial catalog=QuanLyBanHang;persist security
info=True;user
id=sa;password=Sa@12345;MultipleActiveResultSets=True;App=EntityFramework"
providerName="System.Data.SqlClient" />
</connectionStrings>

```

1.4. Xây dựng các trang quản lý dữ liệu

- Tạo các trang index (view), create, update, delete, details của các bảng:

- + Danh mục sản phẩm

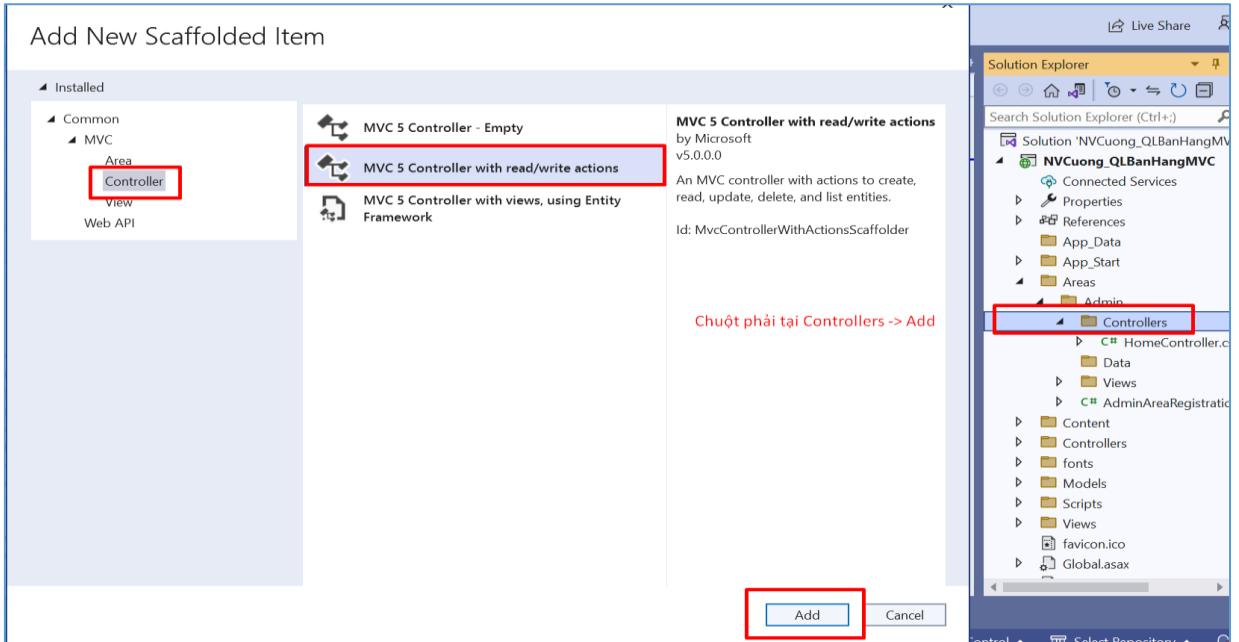
- + Sản phẩm

- + Phân quyền

- + Người dùng

Hướng dẫn tạo index (view), create, update, delete, details cho bảng Danh mục sản phẩm:

- Tạo controller DanhMucSanPham:



- Nhập tên Controller: **DanhMucSanPhamController**



- Nhận được file **DanhMucSanPhamController.cs** định nghĩa sẵn các tên hàm:

DanhMucSanPhamController.cs

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Mvc;
6
7  namespace NVCuong_QLBanHangMVC.Areas.Admin.Controllers
8  {
9      public class DanhMucSanPhamController : Controller
10     {
11         // GET: Admin/DanhMucSanPham
12         public ActionResult Index()
13         {
14             return View();
15         }
16
17         // GET: Admin/DanhMucSanPham/Details/5
18         public ActionResult Details(int id)
19         {
20             return View();
21         }
22
23         // GET: Admin/DanhMucSanPham/Create
24         public ActionResult Create()
25         {
26             return View();
27         }
28     }
29 }

```

Solution Explorer

- NVCuong_QLBanHangMVC
 - Connected Services
 - Properties
 - References
 - App_Data
 - App_Start
 - Areas
 - Admin
 - Controllers
 - C# DanhMucSanPhamController.cs
 - HomeController.cs
 - Data
 - Views
 - DanhMucSanPham
 - Default
 - Home
 - Shared
 - _ViewStart.cshtml
 - web.config
 - C# AdminAreaRegistration.cs
 - Content
 - Controllers

- Thực hiện Build lại dự án

- Tiến hành Add View cho hàm: Index

DanhMucSanPhamController.cs

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;    Chuột phải tại Index
5  using System.Web.Mvc; Chọn Add View
6
7  namespace NVCuong_QLBanHangMVC.Areas.Admin.Controllers
8  {
9      public class DanhMucSanPhamController : Controller
10     {
11         // GET: Admin/DanhMucSanPham
12         public ActionResult Index()    Index()
13         {
14             return View();
15         }
16
17         // GET: Admin/DanhMucSanPham/Details/5
18         public ActionResult Details(int id)
19         {
20             return View();
21         }
22
23         // GET: Admin/DanhMucSanPham/Create
24         public ActionResult Create()
25         {
26             return View();
27         }
28     }
29 }

```

Add New Scaffolded Item

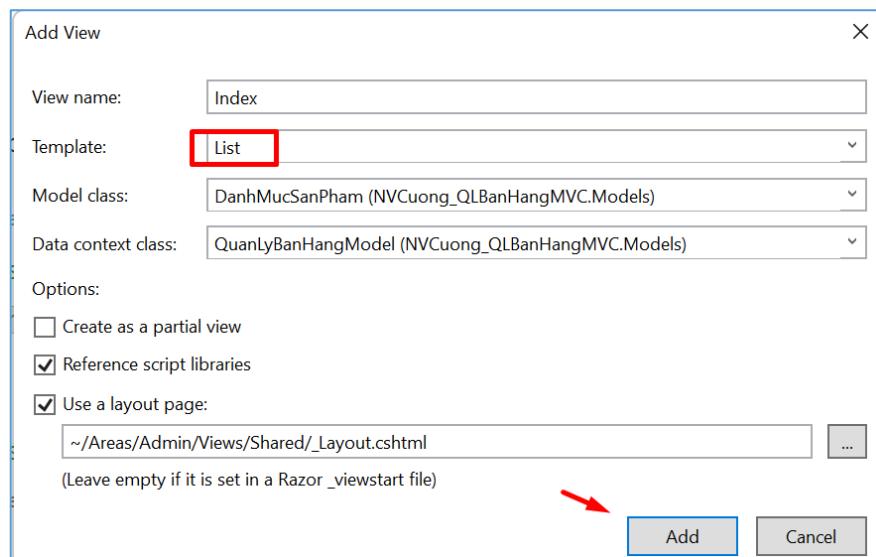
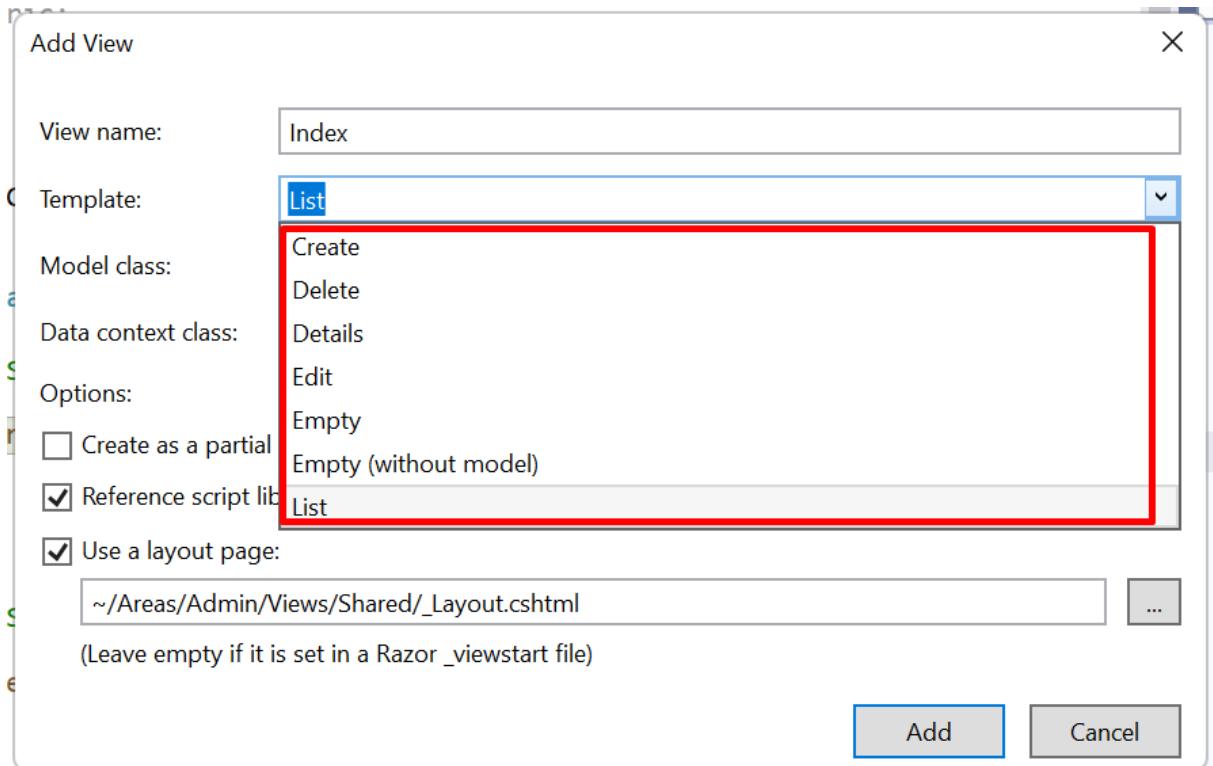
MVC 5 View by Microsoft v5.0.0.0

An MVC view with or without a strongly-typed Model

Id: MvcViewScaffolder

Add Cancel

- Đổi với Hàm index chọn Template là List.
- Đổi với các hàm khác, chọn Template tương ứng.



- Model Class chọn Tên Class mà chúng ta đang thực hiện, Chọn DanhMucSanPham.
- Data context class chọn QuanLyBanHangModel.
- Bấm Add Thành công ta có View được tạo:

```

dex.cshtml  DanhMucSanPhamController.cs  Web.config  QuanLyBanHangModel.cs  Index.cshtml
1 @model IEnumerable<NVCuong_QLBanHangMVC.Models.DanhMucSanPham>
2
3     ViewBag.Title = "Index";
4     Layout = "~/Areas/Admin/Views/Shared/_Layout.cshtml";
5
6
7     <h2>Index</h2>
8
9
10    <p>
11        @Html.ActionLink("Create New", "Create")
12    </p>
13    <table class="table">
14        <tr>
15            <th>
16                @Html.DisplayNameFor(model => model.TenDanhMuc)
17            </th>
18            <th>
19                @Html.DisplayNameFor(model => model.Mota)
20            </th>
21            <th></th>
22        </tr>
23
24       @foreach (var item in Model) {
25            <tr>
26                <td>
27                    @Html.DisplayFor(modelItem => item.TenDanhMuc)
28                </td>

```

- Bây giờ sẽ tiến hành viết Hàm hiển thị Danh mục sản phẩm:

```

cshtml  DanhMucSanPhamController.cs*  Web.config  QuanLyBanHangModel.cs  Index.cshtml  HomeController.cs
Process: [27536] iisexpress.exe  Lifecycle Events  Thread:  Stack Frame:
C:\Users\HUYEN\source\repos\NVCuong_QLBanHangMVC\NVCuong_QLBanHangMVC\Areas\Admin\Controllers\Index.cshtml
7
8     namespace NVCuong_QLBanHangMVC.Areas.Admin.Controllers
9     {
10         0 references
11         public class DanhMucSanPhamController : Controller
12         {
13             QuanLyBanHangModel db = new QuanLyBanHangModel();
14             // GET: Admin/DanhMucSanPham
15
16             public ActionResult Index()
17             {
18                 var listDanhMuc = db.DanhMucSanPhams.ToList();
19                 return View(listDanhMuc);
20             }
21             // GET: Admin/DanhMucSanPham/Details/5
22             0 references

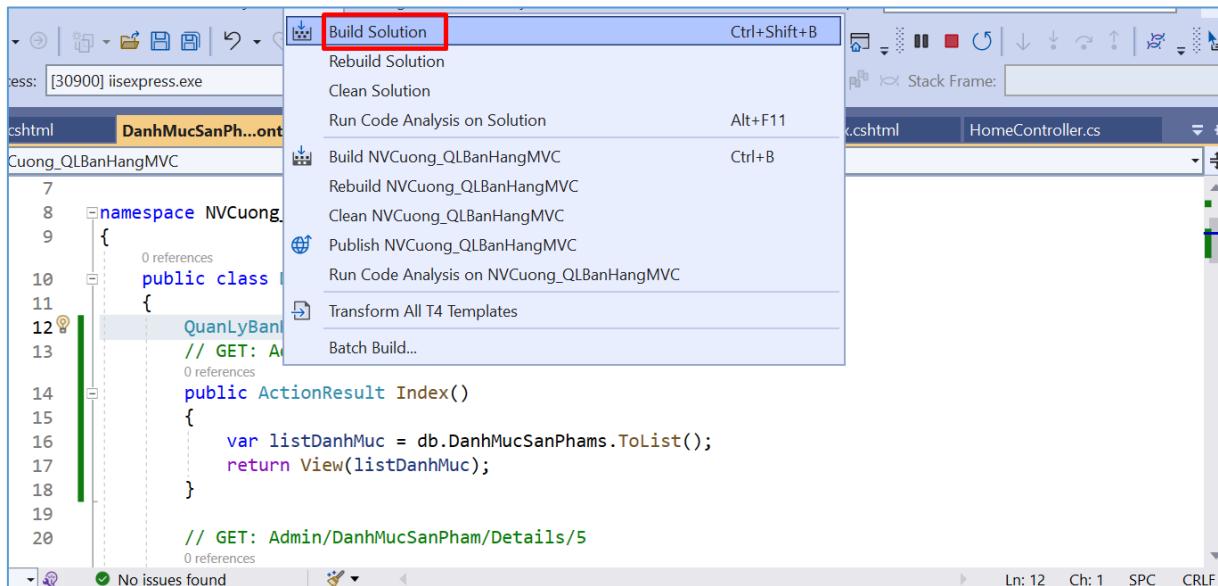
```

```

public class DanhMucSanPhamController : Controller
{
    QuanLyBanHangModel db = new QuanLyBanHangModel();
    // GET: Admin/DanhMucSanPham
    0 references
    public ActionResult Index()
    {
        var listDanhMuc = db.DanhMucSanPhams.ToList();
        return View(listDanhMuc);
    }
}

```

- Build lại dự án:



- Build xong khởi chạy dự án:

<https://localhost:44380/Admin/DanhMucSanPham>

44380 -> là port của website. Các máy tính có thể sẽ khác nhau.

- Chúng ta sẽ hiển thị được danh sách các Danh mục sản phẩm:

A screenshot of a web browser displaying the 'Index' view of an ASP.NET application. The URL in the address bar is 'localhost:44380/Admin/DanhMucSanPham'. The page shows a table with two columns: 'TenDanhMuc' (Category Name) and 'MoTa' (Description). There are eight rows in the table, each representing a category: Xiaomi, Iphone, Nokia, Samsung, LG, Oppo, and Sony. To the right of each row are three links: 'Edit', 'Details', and 'Delete'. At the bottom of the page, there is a copyright notice: '© 2023 - My ASP.NET Application'.

TenDanhMuc	MoTa
Xiaomi	Edit Details Delete
Iphone	Edit Details Delete
Nokia	Edit Details Delete
Samsung	Edit Details Delete
LG	Edit Details Delete
Oppo	Edit Details Delete
Sony	Edit Details Delete

- Hiện tại nếu bấm vào các Button Create New, Edit, Details, Delete sẽ xuất hiện lỗi.
- Tiếp theo chúng ta đi xây dựng View cho các hàm và sẽ viết các lệnh để xử lý cho các hàm Create, Edit, Details, Delete

- Đầu tiên, tiến hành thêm View. Sinh viên thực hiện thêm View tương tự như thêm View cho Hàm Index, chọn Template phù hợp.
- Sau đó, viết code các hàm xử lý:

+ `public ActionResult Details`

```
// GET: Admin/DanhMucSanPham/Details/5
0 references
public ActionResult Details(int id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    DanhMucSanPham danhmucsanpham = db.DanhMucSanPhams.Find(id);
    if (danhmucsanpham == null)
    {
        return HttpNotFound();
    }
    return View(danhmucsanpham);
}
```

+ `public ActionResult Create`

```
public ActionResult Create()
{
    return View();
}

// POST: Admin/DanhMucSanPham/Create
[HttpPost]
0 references
public ActionResult Create(DanhMucSanPham dataForm)
{
    try
    {
        // TODO: Add insert logic here
        db.DanhMucSanPhams.Add(dataForm);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    catch
    {
        return View();
    }
}
```

+ public ActionResult Edit

```
// GET: Admin/DanhMucSanPham/Edit/5
0 references
public ActionResult Edit(int id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    var dataEdit = db.DanhMucSanPhams.Find(id);
    if (dataEdit == null)
    {
        return HttpNotFound();
    }
    return View(dataEdit);
}
```

```
// POST: Admin/DanhMucSanPham/Edit/5
[HttpPost]
0 references
public ActionResult Edit(int id, DanhMucSanPham dataForm)
{
    try
    {
        var dataEdit = db.DanhMucSanPhams.Find(id);
        dataEdit.TenDanhMuc = dataForm.TenDanhMuc;
        dataEdit.MoTa = dataForm.MoTa;
        db.SaveChanges();
        // TODO: Add update logic here
        return RedirectToAction("Index");
    }
    catch
    {
        return View();
    }
}
```

+ public ActionResult Delete

```
// GET: Admin/DanhMucSanPham/Delete/5
public ActionResult Delete(int id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    var dataDelete = db.DanhMucSanPhams.Find(id);
    if (dataDelete == null)
    {
        return HttpNotFound();
    }
    return View(dataDelete);
}

// POST: Admin/DanhMucSanPham/Delete/5
[HttpPost]
public ActionResult Delete(int id, FormCollection collection)
{
    try
    {
        // TODO: Add delete logic here
        var dataDelete = db.DanhMucSanPhams.Find(id);
        db.DanhMucSanPhams.Remove(dataDelete);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    catch
    {
        return View();
    }
}
```

- Build lại dự án.

- Truy cập địa chỉ để kiểm tra các chức năng

<https://localhost:44380/Admin/DanhMucSanPham>

44380 -> là port của website. Các máy tính có thể sẽ khác nhau.

- Kết quả hiển thị:

Application name	
Index	
Create New	
TenDanhMuc	MoTa
Xiaomi	Edit Details Delete
Iphone	Edit Details Delete
Nokia	Edit Details Delete
Samsung	Edit Details Delete
LG	Edit Details Delete
Oppo	Edit Details Delete
Sony	Edit Details Delete

© 2023 - My ASP.NET Application

HƯỚNG DẪN THỰC HÀNH BUỔI SỐ 2

I. Mục tiêu

- Hoàn thiện được các menu quản lý dữ liệu
- Nắm được mục đích của chức năng tìm kiếm dữ liệu, xây dựng được chức năng tìm kiếm dữ liệu.
- Nắm được mục đích, ý nghĩa của chức năng phân trang
- Sử dụng thư viện Pagelist.Mvc thực hiện xây dựng chức năng phân trang
- Kết hợp được chức năng phân trang và tìm kiếm dữ liệu

II. Nội dung

- Xây dựng chức năng nâng cao cho trang quản trị:
 - + Hoàn thiện được các menu quản lý dữ liệu
 - Quản lý sản phẩm
 - Quản lý phân quyền
 - Quản lý người dùng
 - + Thêm tìm kiếm, phân trang cho các trang danh sách dữ liệu (Index)
 - + Kết hợp tìm kiếm và phân trang

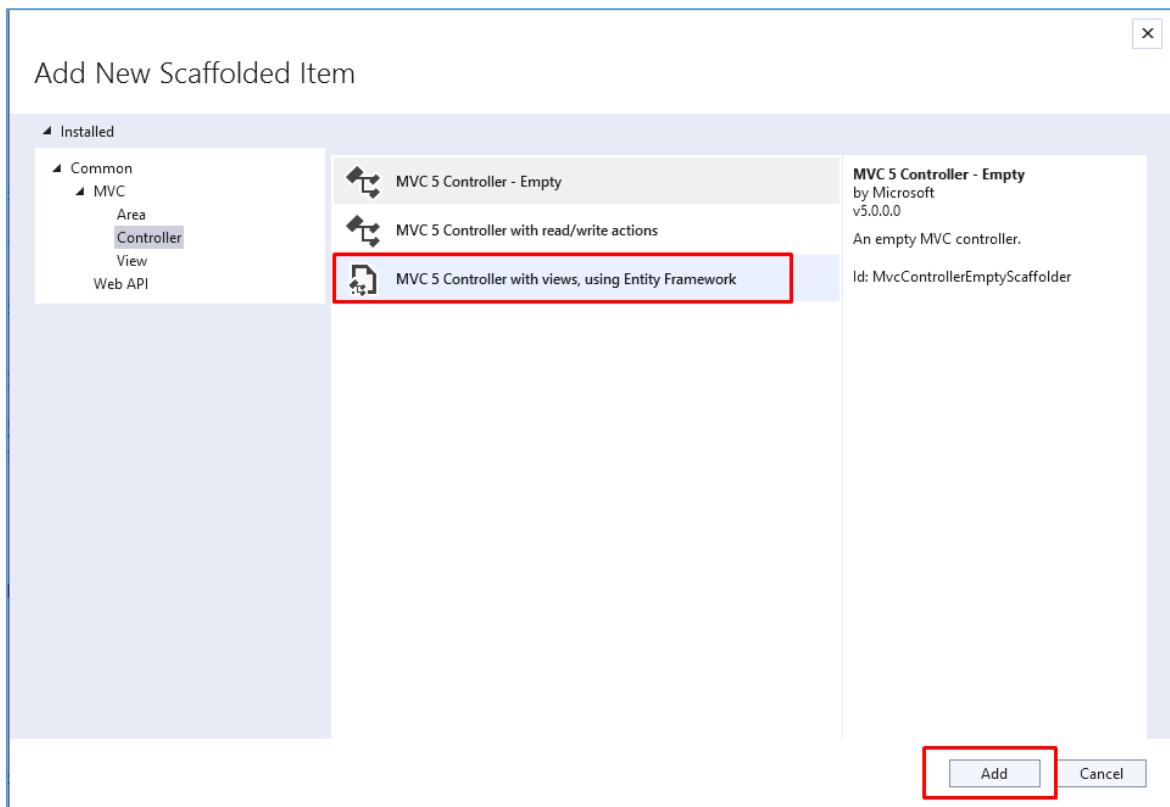
2. Hướng dẫn thực hành: Xây dựng chức năng nâng cao cho trang quản trị

2.1. Hoàn thiện các menu quản lý dữ liệu

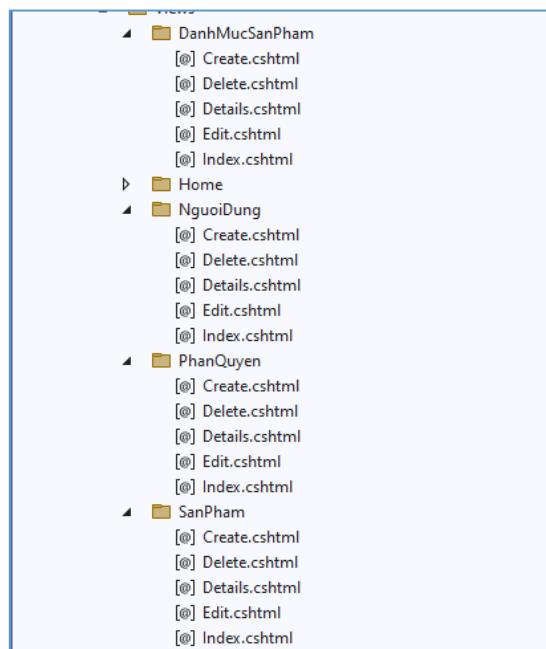
- Sinh viên tiếp tục, thực hiện xây dựng các chức năng Thêm, sửa, xoá, chi tiết đối với các menu:

- + Quản lý sản phẩm
- + Quản lý phân quyền
- + Quản lý người dùng

Gợi ý: Sinh viên có thể sử dụng **MVC 5 Controller with views, using Entity Framework** để tạo ra các phương thức Index, Create, Details, Edit, Detail và Views được xây dựng sẵn.



- Sau khi tạo xong các chức năng cho các menu quản lý tiến hành viết hoá các Button, các label , text ở các View Index, Create, Details, Edit, Detail:



```

<h2>Index</h2>



<p> @Html.ActionLink("Create New", "Create") </p>



| <th> @Html.DisplayNameFor(model => model.TenDanhMuc) </th>                                                                                                                                                                                                                                                                                                                                        | <th> @Html.DisplayNameFor(model => model.MoTa) </th> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| @foreach (var item in Model) {         <td> @Html.DisplayFor(modelItem => item.TenDanhMuc) </td>         <td> @Html.DisplayFor(modelItem => item.MoTa) </td>         <td> @Html.ActionLink("Edit", "edit", new { id=item.MaDanhMuc })   @Html.ActionLink("Details", "Details", new { id=item.MaDanhMuc })   @Html.ActionLink("Delete", "Delete", new { id=item.MaDanhMuc })     </td>     </tr> } |                                                      |


```

- Mặc định tên cột của bảng dữ liệu tại file index.cshtml sẽ hiển thị theo tên trường trong SQL đã tạo. Chúng ta cần thay đổi để tên cột hiển thị đúng:

+ Ban đầu

TenDanhMuc	MoTa	
Xiaomi		Edit Details Delete
Iphone		Edit Details Delete
Nokia		Edit Details Delete
Samsung		Edit Details Delete
LG		Edit Details Delete
Oppo		Edit Details Delete
Sony		Edit Details Delete

© 2023 - My ASP.NET Application

+ Tiến hành chỉnh sửa ở các file trong Models:

Tại file **Models/DanhMucSanPham.cs**

Thêm đoạn mã: [Display(Name = "<Tên cần thay đổi>")] trước đoạn mã khai báo tên cột.

```

0 references
public DanhMucSanPham()
{
    SanPhams = new HashSet<SanPham>();
}

[Key]
3 references
public int MaDanhMuc { get; set; }

[Required]
[StringLength(50)]
[Display(Name = "Tên danh mục")]
2 references
public string TenDanhMuc { get; set; }

[StringLength(200)]
[Display(Name = "Mô tả")]
public string MoTa { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
2 references
public virtual ICollection<SanPham> SanPhams { get; set; }

```

```

[Key]
4 references
public int MaDanhMuc { get; set; }

[Required(ErrorMessage = "Tên danh mục là bắt buộc nhập")]
[StringLength(50)]
[Display(Name = "Tên danh mục")]
4 references
public string TenDanhMuc { get; set; }

[StringLength(200)]
[Display(Name = "Mô tả")]
3 references
public string MoTa { get; set; }

```

+ Chính sửa tương tự đối với các class khác

- Tại view index.cshtml chúng ta có thể thêm các thuộc tính css cho bảng danh sách dữ liệu:

+ Ví dụ thêm viền cho bảng dữ liệu:

```

<h2>Index</h2>

<p> @Html.ActionLink("Create New", "Create") </p>


| @Html.DisplayNameFor(model => model.DanhMucSanPham.TenDanhMuc) |
|----------------------------------------------------------------|
| @Html.DisplayNameFor(model => model.TenSP)                     |
| @Html.DisplayNameFor(model => model.GiaTien)                   |
| @Html.DisplayNameFor(model => model.MoTa)                      |
| @Html.DisplayNameFor(model => model.LaSanPhamMoi)              |
| @Html.DisplayNameFor(model => model.HinhAnh)                   |
| @Html.DisplayNameFor(model => model.SoLuong)                   |


```

- Tiến hành chỉnh sửa `~/Areas/Admin/Views/Shared/_Layout.cshtml` của Trang quản lý để hiển thị các menu chức năng lên Header.

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ViewBag.Title - My ASP.NET Application</title>
    <Styles.Render("~/Content/css")>
    <Scripts.Render("~/bundles/modernizr")>
</head>
<body>
    <div class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <@Html.ActionLink("Application name", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })>
            </div>
            <div class="navbar-collapse collapse">
                <ul class="nav navbar-nav">
                    <li>@Html.ActionLink("Danh mục", "Index", "DanhMucSanPham")</li>
                    <li>@Html.ActionLink("Sản phẩm", "Index", "SanPham")</li>
                    <li>@Html.ActionLink("Phân Quyền", "Index", "PhanQuyen")</li>
                    <li>@Html.ActionLink("Người Dùng", "Index", "NguoiDung")</li>
                </ul>
            </div>
        </div>
    </div>
    <div class="container body-content">
        <@RenderBody()>
        <br />
    </div>
    <@Scripts.Render("~/bundles/jquery")>
</body>
</html>

```

- Thêm mới các thẻ:

```

<div class="navbar-collapse collapse">
    <ul class="nav navbar-nav">
        <li>@Html.ActionLink("Danh mục", "Index", "DanhMucSanPham")</li>
        <li>@Html.ActionLink("Sản phẩm", "Index", "SanPham")</li>
        <li>@Html.ActionLink("Phân Quyền", "Index", "PhanQuyen")</li>
        <li>@Html.ActionLink("Người Dùng", "Index", "NguoiDung")</li>
    </ul>
</div>

```

- Build lại dự án và tiến hành chạy dự án chúng ta được trang quản trị dự án đơn giản tại đường link:

<https://localhost:44380/Admin/DanhMucSanPham>

44380 -> là port của website. Các máy tính có thể sẽ khác nhau.

- Trang quản trị website quản lý bán hàng:

Danh mục sản phẩm

Thêm mới

Tên danh mục	Mô tả
Xiaomi	Sửa Chi tiết Xoá
Iphone	Sửa Chi tiết Xoá
Nokia	Sửa Chi tiết Xoá
Samsung	Sửa Chi tiết Xoá
LG	Sửa Chi tiết Xoá
Oppo	Sửa Chi tiết Xoá
Sony	Sửa Chi tiết Xoá

© 2023 - My ASP.NET Application

2.2. Xây dựng chức năng tìm kiếm

- Đầu tiên, tại phương thức Index của DanhMucSanPhamController chỉnh sửa code thành như sau:

```
public ActionResult Index(string searchString)
{
    // Lấy toàn bộ danh sách
    var danhMucSanPhams = from danhmuc in db.DanhMucSanPhams
                          select danhmuc;
    // kiểm tra chuỗi searchString từ form tìm kiếm gửi lên có rỗng/null hay không
    if (!string.IsNullOrEmpty(searchString))
    {
        //lọc tên danh mục theo searchString
        danhMucSanPhams = danhMucSanPhams.Where(s => s.TenDanhMuc.Contains(searchString));
    }
    //trả về kết quả
    return View(danhMucSanPhams);
}
```

- Tiếp theo tại file **Areas/Admin/Views/DanhMucSanPham/Index.cshtml**

+ Thêm mới đoạn code sau để hiển thị input và button tìm kiếm:

```
<h2>Danh mục sản phẩm</h2>
<p> @Html.ActionLink("Thêm mới", "Create") </p>

@using (Html.BeginForm("Index", "DanhMucSanPham", FormMethod.Get)) // -- phần thêm mới Form tìm kiếm
{
    <p> Tên danh mục: @Html.TextBox("searchString") <input type="submit" value="Tìm kiếm" />
}
```

- + Giao diện sau khi thêm đoạn code hiển thị form tìm kiếm:

Danh mục sản phẩm

Thêm mới

Tên danh mục	Mô tả
Xiaomi	Sửa Chi tiết Xoá
Iphone	Sửa Chi tiết Xoá
Nokia	Sửa Chi tiết Xoá
Samsung	Sửa Chi tiết Xoá
LG	Sửa Chi tiết Xoá
Oppo	Sửa Chi tiết Xoá
Sony	Sửa Chi tiết Xoá

© 2023 - My ASP.NET Application

- + Thực hiện nhập vào tên danh mục cần tìm kiếm và bấm tìm kiếm:
- + Nhập “SAM” bấm tìm kiếm trên thanh url sẽ có đường dẫn là:

<https://localhost:44312/Admin/DanhMucSanPham?SearchString=SAM>

Có nghĩa là từ View đang truyền tham số **SearchString**=SAM xuống phương thức Index của DanhMucSanPhamController để xử lý:

The screenshot shows a browser window with the URL `localhost:44312/Admin/DanhMucSanPham?SearchString=SAM`. The search term 'SAM' is highlighted in the search bar. Below the search bar, a table displays products matching the search term. One row for 'Samsung' is visible, with options to edit ('Sửa'), view details ('Chi tiết'), or delete ('Xoá').

Tên danh mục	Mô tả
Samsung	Sửa Chi tiết Xoá

© 2023 - My ASP.NET Application

- + Phương thức Index nhận về giá trị SAM từ View truyền xuống và lọc dữ liệu tên danh mục có chứa từ khóa “SAM”:

```

public ActionResult Index(string searchString)
{
    // Lấy toàn bộ danh sách
    var danhMucSanPhams = from danhmuc in db.DanhMucSanPhams
                           select danhmuc;
    // kiểm tra chuỗi searchString từ form tìm kiếm gửi lên có rỗng/null hay không
    if (!String.IsNullOrEmpty(searchString))
    {
        //lọc tên danh mục theo searchString
        danhMucSanPhams = danhMucSanPhams.Where(s => s.TenDanhMuc.Contains(searchString));
    }
    //trả về kết quả
    return View(danhMucSanPhams);
}

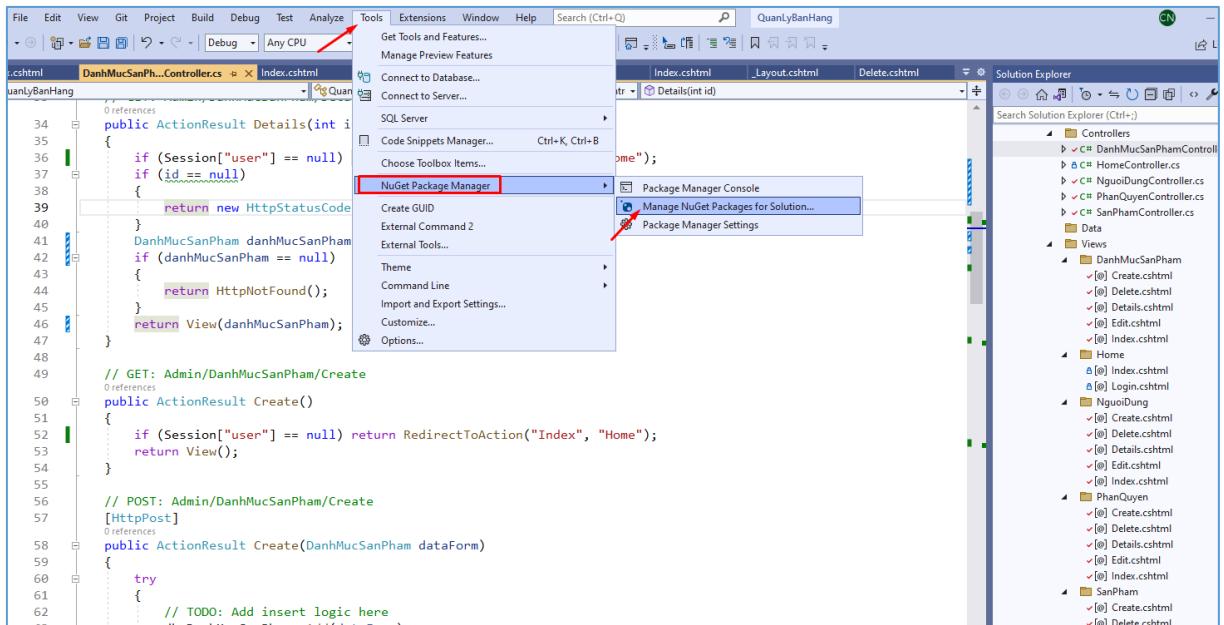
```

+ Ta thấy rằng Phương thức Contains vẫn đưa ra dữ liệu tên danh mục **"samsung"**. Như vậy, ta thấy rằng phương thức Contains không phân biệt chữ cái hoa thường và trả về kết quả của tên danh mục có chứa từ khoá cần tìm kiếm.

- Thực hiện xây dựng chức năng tìm kiếm đối với các Menu quản lý: Người dùng, Sản phẩm, Phân quyền.
- Tuỳ theo các Menu để chúng ta tìm kiếm theo trường dữ liệu cần thiết, có thể phải thiết kế tìm kiếm kết hợp nhiều trường dữ liệu tại một Menu.

2.2. Xây dựng kết hợp tìm kiếm + phân trang

- Tải về Package hỗ trợ phân trang sử dụng NuGet
- + Tools -> NuGet Package Manager -> Manager NuGet packages For Solution



- Sau đó chọn vào Browe, hiển thị:

NuGet - Solution Index.cshtml DanhMucSanPh...Controller.cs Index.cshtml Create.cshtml Index.cshtml Index.cshtml Index.cshtml

Browse Installed ▲ Updates 6 Consolidate

Search (Ctrl+L) 🔍 ⌂ Include prerelease

Manage Packages for Solution
Package source: nuget.org

.NET	Packages	Version
Microsoft.Extensions.DependencyInjection	by Microsoft, 1.92B downloads	7.0.0
Microsoft.Extensions.Logging	by Microsoft, 1.8B downloads	7.0.0
System.Text.Json	by Microsoft, 888M downloads	7.0.2
Microsoft.Bcl.AsyncInterfaces	by Microsoft, 777M downloads	7.0.0
Serilog	by Serilog Contributors, 682M downloads	2.12.0
System.Drawing.Common	by Microsoft, 677M downloads	7.0.0
Microsoft.Win32.SystemEvents	by Microsoft, 627M downloads	7.0.0

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.
 Do not show this again

- Tìm kiếm Packages **PagedList.Mvc**

Browse Installed ▲ Updates 6 Consolidate

PagedList.Mvc 1 🔍 ⌂ Include prerelease

Manage Packages for Solution
Package source: nuget.org

Packages	Version	
PagedList.Mvc	by Troy Goode, 6.96M downloads	4.5.0
PagedList	by Troy Goode, 8.45M downloads	1.17.0
X.PagedList.Mvc.Core	by Copyright Troy Goode, Andrew Gubskiy © 2023, 1.58M downloads	8.4.5
X.PagedList.Mvc	by Copyright Troy Goode, Ernado © 2020, 850K downloads	8.0.7
PagedList.Core.Mvc	by Troy Goode, Hieu Le, 377K downloads	3.0.0
PagedList.NetCore	by Thiago Barradas, 80.5K downloads	1.0.6
X.PagedList.Mvc.Bootstrap4	by Money Yu, 69.3K downloads	8.1.0

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.
 Do not show this again

PagedList.Mvc nuget.org

Versions - 0

Project	Installed
QuanLyBanHang	<input checked="" type="checkbox"/>

Installed: not installed Uninstall

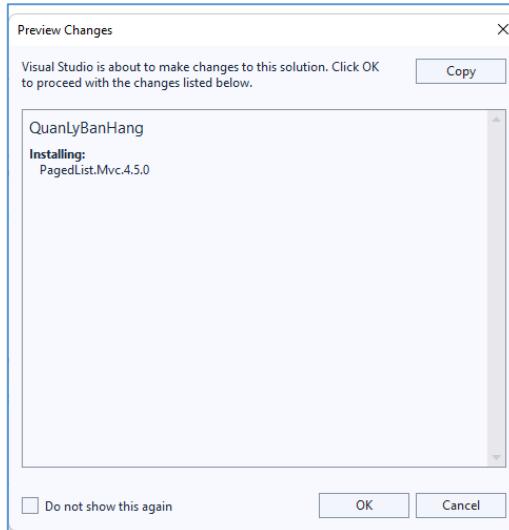
Version: Latest stable 4.5.0 **Install** 2

Options

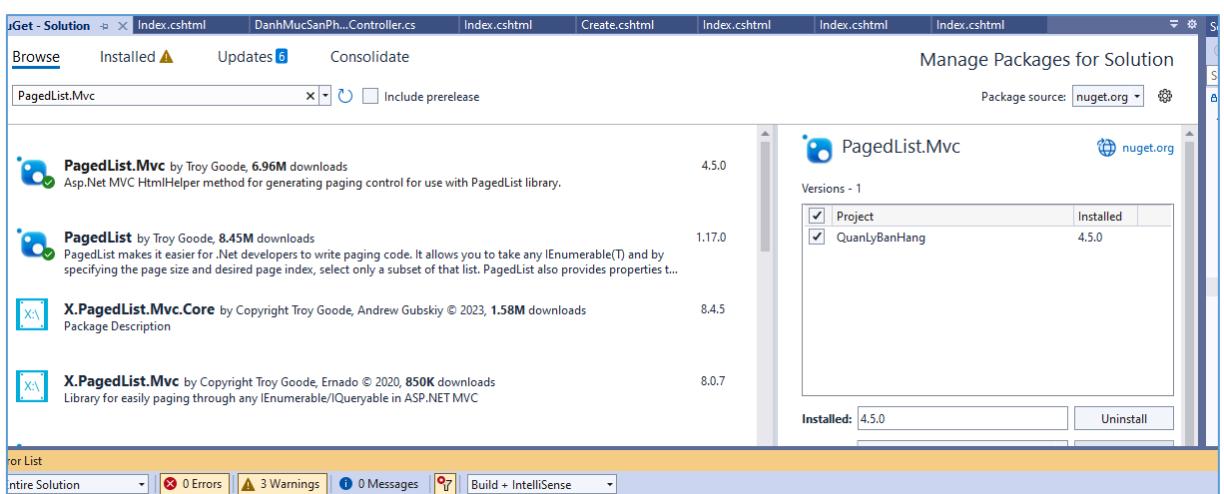
Description
Asp.Net MVC HtmlHelper method for generating paging control for use with PagedList library.

Version: 4.5.0
Author(s): Troy Goode
License: [View License](#)
Date published: Tuesday, October 1, 2013 (10/1/2013)
Project URL: <http://github.com/TroyGoode/PagedList>

- Chọn OK



- Cài đặt thành công.



- Thực hiện phân trang cho Menu Danh mục sản phẩm:

- + Sửa phương thức Index, tập tin *Controller/DanhMucSanPham.cs*
- + Mở tập tin *Controller/DanhMucSanPham.cs*, nhúng thêm thư viện : **using PagedList;** ở đầu trang, sau đó ở phương thức Index sửa như sau.

```

using PagedList; ←
namespace QuanLyBanHang.Areas.Admin.Controllers
{
    public class DanhMucSanPhamController : Controller
    {
        QuanLyBanHangModel db = new QuanLyBanHangModel();
        // GET: Admin/DanhMucSanPham
        public ActionResult Index(int? page, string searchString)
        {
            // Lấy toàn bộ danh sách
            var danhMucSanPhams = from danhmuc in db.DanhMucSanPhams
                                  select danhmuc;
            // kiểm tra chuỗi searchString từ form tìm kiếm gửi lên có rỗng/null hay không
            if (!String.IsNullOrEmpty(searchString))
            {
                //lọc tên danh mục theo searchString
                danhMucSanPhams = danhMucSanPhams.Where(s => s.TenDanhMuc.Contains(searchString));
            }
            // Sắp xếp theo MaDanhMuc Tăng dần
            danhMucSanPhams = danhMucSanPhams.OrderBy(x => x.MaDanhMuc);
            // Phân trang
            if (page == null) page = 1;
            int pageSize = 5;
            int pageNumber = (page ?? 1);
            //trả về kết quả sử dụng phân trang
            return View(danhMucSanPhams.ToPagedList(pageNumber, pageSize));
        }
    }
}

```

+ Tiến hành thay đổi trên View Index:

+ Thay đổi trên đầu trang sử dụng Package:

```

@model PagedList.IPagedList<QuanLyBanHang.Models.DanhMucSanPham>
@using PagedList.Mvc;

```

```

@model PagedList.IPagedList<QuanLyBanHang.Models.DanhMucSanPham>
@using PagedList.Mvc;

 @{
    ViewBag.Title = "Index";
    Layout = "~/Areas/Admin/Views/Shared/_Layout.cshtml";
}



## Danh mục sản phẩm



Thêm mới Create


@using (Html.BeginForm("Index", "DanhMucSanPham", FormMethod.Get)) // -- phần thêm mới Form tìm kiếm
{
    <p> Tên danh mục: @Html.TextBox("searchString") <input type="submit" value="Tìm kiếm" />
}

```

+ Thay đổi phía cuối trang – sau thẻ Table:

```

<!-- Thêm phân trang -->
<br />
Trang @(Model.PageCount < Model.PageNumber ? 0 : Model.PageNumber) /
@Model.PageCount

@Html.PagedListPager(Model, page => Url.Action("Index",
    new { page }))
<!-- Kết thúc -->

```

```

        @Html.ActionLink("Sửa", "Edit", new { id = item.MaDanhMuc }) |
        @Html.ActionLink("Chi tiết", "Details", new { id = item.MaDanhMuc }) |
        @Html.ActionLink("Xoá", "Delete", new { id = item.MaDanhMuc })
    </td>
</tr>
}

</table>

<br />
Trang @(Model.PageCount < Model.PageNumber ? 0 : Model.PageNumber) / @Model.PageCount
@Html.PagedListPager(Model, page => Url.Action("Index",
new { page }))


```

+ Thay đổi phần hiển thị label:

```


| @Html.DisplayNameFor(model=>model.FirstOrDefault().TenDanhMuc) | @Html.DisplayNameFor(model => model.FirstOrDefault().MoTa) |  |
|----------------------------------------------------------------|------------------------------------------------------------|--|
|----------------------------------------------------------------|------------------------------------------------------------|--|


```

Hoặc thay đổi như sau:

```

@using (Html.BeginForm("Index", "DanhMucSanPham", FormMethod.Get)) // -- phần thêm mới Form tìm kiếm
{
    <p>
        Tên danh mục: @Html.TextBox("SearchString") <input type="submit" value="Tìm kiếm" />
    </p>
}

<table class="table table-bordered">
    <tr>
        <th>
            Tên danh mục
        </th>
        <th>
            Mô tả
        </th>
    </tr>

```

- Build lại trang và khởi chạy dự án
- Ta được kết quả phân trang như sau:

The screenshot shows a web application interface for managing product categories. At the top, there is a navigation bar with links: Trang quản trị, Danh mục, Sản phẩm, Phân Quyền, and Người Dùng. Below the navigation bar, the title "Danh mục sản phẩm" is displayed. A search bar labeled "Thêm mới" contains a text input field and a "Tim kiếm" button. Below the search bar is a table listing categories. The table has three columns: "Tên danh mục", "Mô tả", and actions. The categories listed are Xiaomi, Iphone, Nokia, Samsung, and LG. Each row includes a "Sửa | Chi tiết | Xoá" link. Below the table, a red box highlights a pagination control. The control shows "Trang 1 / 2" and includes a set of buttons: "1" (highlighted in blue), "2", and "»". At the bottom of the page, there is a copyright notice: "© 2023 - My ASP.NET Application".

- + Chúng ta đang khai báo pageSize trong phương thức Index = 5 nên mỗi trang sẽ hiển thị 5 bản ghi dữ liệu.
- + Thay đổi tại code để hiển thị bản ghi như mong muốn hoặc xây dựng để có thể chọn động pageSize tại View.

```
// GET: Admin/DanhMucSanPham
0 references
public ActionResult Index(int? page, string searchString)
{
    // Lấy toàn bộ danh sách
    var danhMucSanPhams = from danhmuc in db.DanhMucSanPhams
                          select danhmuc;
    // kiểm tra chuỗi searchString từ form tìm kiếm gửi lên có rỗng/null hay không
    if (!String.IsNullOrEmpty(searchString))
    {
        //lọc tên danh mục theo searchString
        danhMucSanPhams = danhMucSanPhams.Where(s => s.TenDanhMuc.Contains(searchString));
    }
    // Sắp xếp theo MaDanhMuc Tăng dần
    danhMucSanPhams = danhMucSanPhams.OrderBy(x => x.MaDanhMuc);
    // Phân trang
    if (page == null) page = 1;
    int pageSize = 5; // highlighted
    int pageNumber = (page ?? 1);
    //trả về kết quả sử dụng phân trang
    return View(danhMucSanPhams.ToPagedList(pageNumber, pageSize));
}
```

- Thực hiện xây dựng phân trang đối với các Menu quản lý: Người dùng, Sản phẩm, Phân quyền.

HƯỚNG DẪN THỰC HÀNH BUỔI SỐ 3

I. Mục tiêu

- Biết cách Validate các form khi nhập dữ liệu
- Tích hợp được CKeditor, Ckfinder vào dự án.
- Thêm mới được dữ liệu sản phẩm sử dụng Ckfinder để upload hình ảnh sản phẩm
- Biết cách sử dụng @Html.Raw() để hiển thị nội dung html.

II. Nội dung

- Validate form khi nhập dữ liệu
- Tích hợp CKeditor, Ckfinder vào dự án. (Mục đích Upload ảnh sản phẩm và trình soạn thảo văn bản)
- Thêm mới dữ liệu sản phẩm, có kèm hình ảnh sản phẩm, phần mô tả sử dụng ckeditor.
- Sử dụng @Html.Raw() để hiển thị nội dung html.

3. Hướng dẫn thực hành: Xây dựng chức năng nâng cao cho trang quản trị (tiếp)

3.1. Validate form khi nhập dữ liệu

- Khi nhập dữ liệu như: email, số điện thoại, mật khẩu... ở các form chúng ta cần đưa ra cảnh báo nếu không nhập dữ liệu, dữ liệu sai, không đúng định dạng...

The screenshot shows a web application interface. At the top, there is a navigation bar with links: Trang quản trị, Danh mục, Sản phẩm, Phân Quyền, and Người Dùng. Below the navigation bar, the main content area has a title 'Thêm mới'. There are two input fields: 'Tên danh mục' and 'Mô tả'. The 'Tên danh mục' field has a red border and a validation error message: 'The Tên danh mục field is required.' A red arrow points to this message. Below the input fields is a 'Thêm' (Add) button, which is also highlighted with a red border. At the bottom left of the form is a 'Quay lại' (Back) link. At the very bottom of the page, there is a copyright notice: '© 2023 - My ASP.NET Application'.

- Chúng ta sẽ thực hiện cấu hình các class trong thư mục **Models**
- + VD: Đối với class DanhMucSanPham: Trường Tên danh mục chúng ta cần thông báo là bắt buộc nhập.

```

[Key]
4 references
public int MaDanhMuc { get; set; }

[Required(ErrorMessage = "Tên danh mục là bắt buộc nhập")]
[StringLength(50)]
[Display(Name = "Tên danh mục")]
5 references
public string TenDanhMuc { get; set; }

[StringLength(200)]
[Display(Name = "Mô tả")]
4 references
public string MoTa { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:Collect:
public virtual ICollection<SanPham> SanPhams { get; set; }
}

```

DanhMucSanPham.cs

- Home
- NguoiDung
- PhanQuyen
- SanPham
- Shared
- web.config
- Content
- Controllers
- fonts
- Models
- DanhMucSanPham.cs**
- ChiTietDonHang.cs
- DonHang.cs
- KhachHang.cs
- NguoiDung.cs
- PhanQuyen.cs
- QuanLyBanHangModel.cs
- SanPham.cs
- sysdiagram.cs
- Scripts
- Utile

+ Build lại dự án, khởi chạy dự án để xem kết quả:

Trang quản trị Danh mục Sản phẩm Phân Quyền Người Dùng

Thêm mới

Tên danh mục

Tên danh mục là bắt buộc nhập

Mô tả

Thêm

Quay lại

© 2023 - My ASP.NET Application

+ Đối với những trường là Email, Mật khẩu, Số điện thoại chúng ta cần đưa ra những quy định riêng. Ví dụ: Email cần đúng định dạng, số điện thoại 10 ký tự, mật khẩu ít nhất là 8 ký tự, nhiều nhất là 20 ký tự...

```

[StringLength(50)]
[Display(Name = "Email")]
[Required(ErrorMessage = "Email bắt buộc nhập")]
[DataType(DataType.EmailAddress, ErrorMessage = "E-mail sai định dạng")]
0 references
public string Email { get; set; }

[StringLength(10)]
[Display(Name = "Số điện thoại")]
[Required(ErrorMessage = "Số điện thoại bắt buộc nhập")]
1 reference
public string SoDienThoai { get; set; }

[StringLength(50)]
[Required(ErrorMessage = "Tên đăng nhập bắt buộc nhập")]
[Display(Name = "Tên đăng nhập")]
1 reference
public string TenDangNhap { get; set; }

[Required(ErrorMessage = "Mật khẩu bắt buộc nhập")]
[StringLength(maximumLength: 20, MinimumLength = 8, ErrorMessage = "Độ dài mật khẩu từ 8-20 kí tự")]
[Display(Name = "Mật khẩu")]
1 reference
public string MatKhau { get; set; }

```

- Kết quả sau khi cấu hình validate:

Trang quản trị Danh mục Sản phẩm Phân Quyền Người Dùng

Thêm mới người dùng

Họ tên

Họ tên bắt buộc nhập

Địa chỉ

Email

! Please include an '@' in the email address. 'cuongyd196@gmail.com' is missing an '@'.

Số điện thoại bắt buộc nhập

Tên đăng nhập

Tên đăng nhập bắt buộc nhập

Mật khẩu

Mật khẩu bắt buộc nhập

Ảnh đại diện

Quyền Admin

Thêm mới

Quay lại

- Thực hiện Validate cho các trường dữ liệu cần thiết của các class.

3.2. Tích hợp CKeditor và Ckfinder vào dự án

- Mục đích: Để chúng ta sử dụng được trình soạn thảo thông minh cho mô tả sản phẩm và Upload hình ảnh sản phẩm.

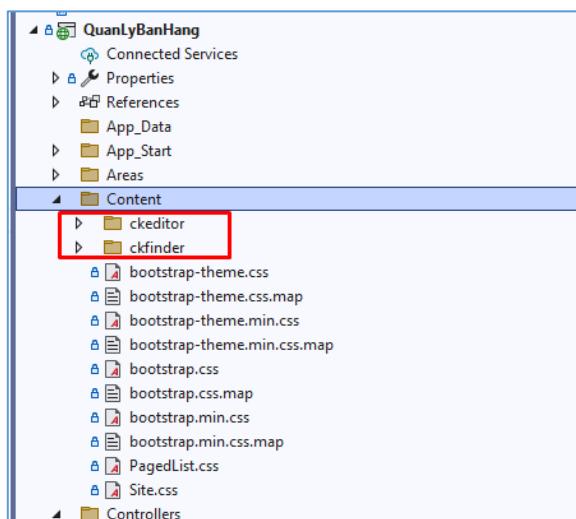
- Tải về gói cài đặt:

Link tải:

https://bit.ly/LabLapTrinhWeb_FictHdu

+ Thư mục download chọn **CkeditorAndCkfider.zip** và tải xuống.

+ Giải nén và cho bên trong thư mục **Content**:



+ Mặc định các config của ckeditor và ckfinder đã config đặt trong Content của dự án. Nếu đặt ckeditor và ckfinder ở thư mục khác cần chỉnh sửa lại các config của ckeditor và ckfinder tại file: ***ckeditor/config.js***

- + Tiếp theo sao chép toàn bộ file trong thư mục **ckfinder/bin/Debug** vào thư mục **bin** ở project:

Local Disk (D) > FICT > Lab_QuanLyBanHangMVC > QuanLyBanHang			
Name	Date modified	Type	Size
App_Start	28/2/2023 12:23 PM	File folder	
Areas	28/2/2023 12:23 PM	File folder	
bin	8/3/2023 7:12 AM	File folder	
Content	28/2/2023 12:23 PM	File folder	
Controllers	8/3/2023 7:12 AM	File folder	
fonts	28/2/2023 12:23 PM	File folder	
Models	8/3/2023 7:12 AM	File folder	
obj	28/2/2023 12:23 PM	File folder	
Properties	28/2/2023 12:23 PM	File folder	
Scripts	28/2/2023 12:23 PM	File folder	
Uploads	28/2/2023 12:23 PM	File folder	
Views	28/2/2023 12:23 PM	File folder	

- Tiến hành chỉnh sửa View thêm mới và View Edit của Sản phẩm"

Areas/Admin/Views/SanPham/Create.cshtml

Areas/Admin/Views/SanPham/Edit.cshtml

- + Tại đầu trang, thêm các thẻ script:

```
<script src="~/Content/ckeditor/ckeditor.js"></script>
<script src="~/Content/ckfinder/ckfinder.js"></script>
<script src="~/Scripts/jquery-3.4.1.slim.js"></script>
```

```

1 @model QuanLyBanHang.Models.SanPham
2 <script src="~/Content/ckeditor/ckeditor.js"></script>
3 <script src="~/Content/ckfinder/ckfinder.js"></script>
4 <script src="~/Scripts/jquery-3.4.1.slim.js"></script>
5 @{
6     ViewBag.Title = "Create";
7     Layout = "~/Areas/Admin/Views/Shared/_Layout.cshtml";
8 }
9
10 <h2>Thêm mới</h2>
11
12 @using (Html.BeginForm())
13 {
14     @Html.AntiForgeryToken()
15
16     <div class="form-horizontal">
17
18         <hr />
19         @Html.ValidationSummary(true, "", new { @class = "text-danger" })
20         <div class="form-group">
21             @Html.LabelFor(model => model.TenSP, htmlAttributes: new { @class = "control-label col-md-2" })
22             <div class="col-md-10">
23                 @Html.EditorFor(model => model.TenSP, new { htmlAttributes = new { @class = "form-control" } })
24                 @Html.ValidationMessageFor(model => model.TenSP, "", new { @class = "text-danger" })
25             </div>
26         </div>
27
28         <div class="form-group">
29             @Html.LabelFor(model => model.GiaTien, htmlAttributes: new { @class = "control-label col-md-2" })
30             <div class="col-md-10">
31                 @Html.EditorFor(model => model.GiaTien, new { htmlAttributes = new { @class = "form-control" } })
32                 @Html.ValidationMessageFor(model => model.GiaTien, "", new { @class = "text-danger" })
33             </div>
34         </div>
35
36     </div>
37
38     <div class="form-group">
39         <input type="submit" value="Thêm mới" class="btn btn-primary" />
40     </div>
41 }
42
43 </div>
44
45 No issues found

```

+ Tại cuối trang, thêm đoạn Script:

```
<script language="javascript" type="text/javascript">
$(function () {
    $("#chonDuongDanAnh").click(function () {
        var ckfiner = new CKFinder();
        ckfiner.selectActionFunction = function (fileUrl) {
            $("#HinhAnh").val(fileUrl);
        };
        ckfiner.popup();
    })
});
var ckeditor;
function createEditor(languageCode, id) {
    var editor = CKEDITOR.replace(id, { language: languageCode });
}
$(function () { createEditor('vi', 'Mo') });
</script>
```

```
78     :         <input type="submit" value="Thêm" class="btn btn-default" />
79     :     </div>
80     : </div>
81     : };
82 };
83
84 <div>
85     :     @Html.ActionLink("Quay lại", "Index")
86 </div>
87 <script language="javascript" type="text/javascript">
88 $(function () {
89     $("#chonDuongDanAnh").click(function () {
90         var ckfiner = new CKFinder();
91         ckfiner.selectActionFunction = function (fileUrl) {
92             $("#HinhAnh").val(fileUrl);
93         };
94         ckfiner.popup();
95     })
96 });
97 var ckeditor;
98 function createEditor(languageCode, id) {
99     var editor = CKEDITOR.replace(id, { language: languageCode });
100 }
101 $(function () { createEditor('vi', 'Mo') });
102 </script>
```

+ Tại input MoTa chỉnh sửa thành:

```
<div class="form-group">
    @Html.LabelFor(model => model.MoTa, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.TextArea("MoTa", new { @class = "ckeditor", @id = "MoTa" })
        @Html.ValidationMessageFor(model => model.MoTa, "", new { @class = "text-danger" })
    </div>
</div>
```

+ Tại input Hình Ảnh, thêm button upload ảnh:

```
<div class="form-group">
    @Html.LabelFor(model => model.HinhAnh, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.HinhAnh, new { htmlAttributes = new { @class = "form-control" } })
        <button id="chonDuongDanAnh" type="button" class="btn btn-info"><i class="fas fa-image font-size-20"></i>Upload</button>
        @Html.ValidationMessageFor(model => model.HinhAnh, "", new { @class = "text-danger" })
    </div>
</div>
```

- + Tại /Areas/Admin/Controllers/SanPhamController.cs
- + Phương thức Create và Edit chúng ta cần thêm “ValidateInput(false)” để có thể lưu dữ liệu được của Mô tả khi soạn thảo bằng CkEditor.

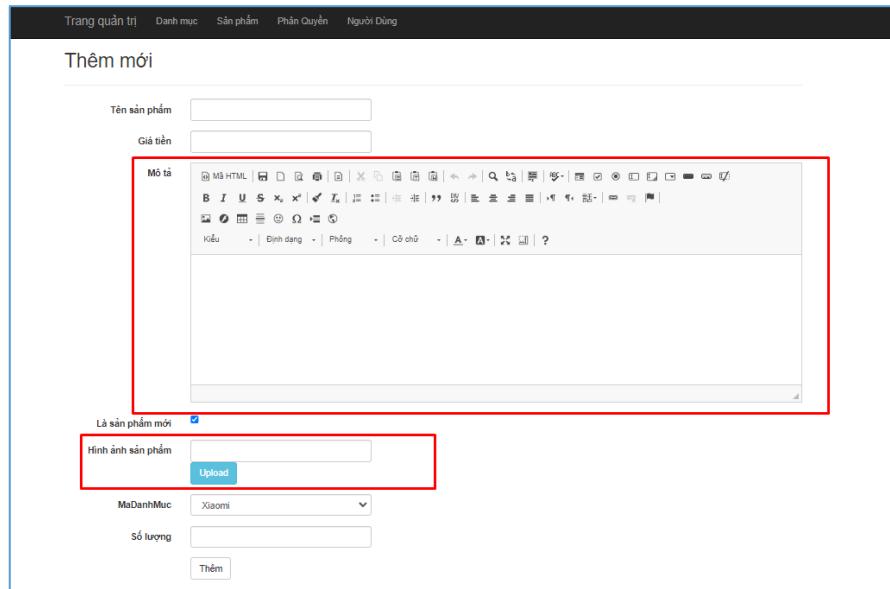
```

public ActionResult Create()...
// POST: Admin/SanPham/Create
// To protect from overposting attacks, enable the specific properties you want to bind more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost, ValidateInput(false)]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "MaSanPham,TenSP,GiaTien,MoTa,LaSanPhamMoi")]
    ...
// GET: Admin/SanPham/Edit/5
public ActionResult Edit(int? id)...
// POST: Admin/SanPham/Edit/5 ...
[HttpPost, ValidateInput(false)]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "MaSanPham,TenSP,GiaTien,MoTa,LaSanPhamMoi")]
{
    if (ModelState.IsValid)
    {
        db.Entry(sanPham).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
}

```

The screenshot shows the Visual Studio interface. On the left is the code editor with the SanPhamController.cs file open. On the right is the Solution Explorer showing the project structure for 'QuanLyBanHang'. Two red arrows point to the [ValidateAntiForgeryToken] attribute on the Create and Edit methods, highlighting the addition of this attribute to handle overposting attacks.

- Build lại dự án và chạy dự án, kết quả:



3.3. Thực hiện thêm mới các sản phẩm có kèm hình ảnh sản phẩm và mô tả.

Thực hiện thêm mới các sản phẩm có kèm hình ảnh sản phẩm và mô tả.

- Tại trang Index sản phẩm, bấm vào Thêm mới.

Trang quản trị													
Danh sách													
Thêm mới													
Tên sản phẩm:	<input type="text"/>	<input type="button" value="Tim kiếm"/>											
<table border="1"> <thead> <tr> <th>Tên danh mục</th><th>Tên sản phẩm</th><th>Giá tiền</th><th>Mô tả</th><th>Là sản phẩm mới</th><th>Hình ảnh sản phẩm</th><th>Số lượng</th></tr> </thead> </table>							Tên danh mục	Tên sản phẩm	Giá tiền	Mô tả	Là sản phẩm mới	Hình ảnh sản phẩm	Số lượng
Tên danh mục	Tên sản phẩm	Giá tiền	Mô tả	Là sản phẩm mới	Hình ảnh sản phẩm	Số lượng							
Trang 0 / 0													
© 2023 - My ASP.NET Application													

- Hiển thị trang thêm mới sản phẩm:

<https://localhost:44312/Admin/SanPham/Create>

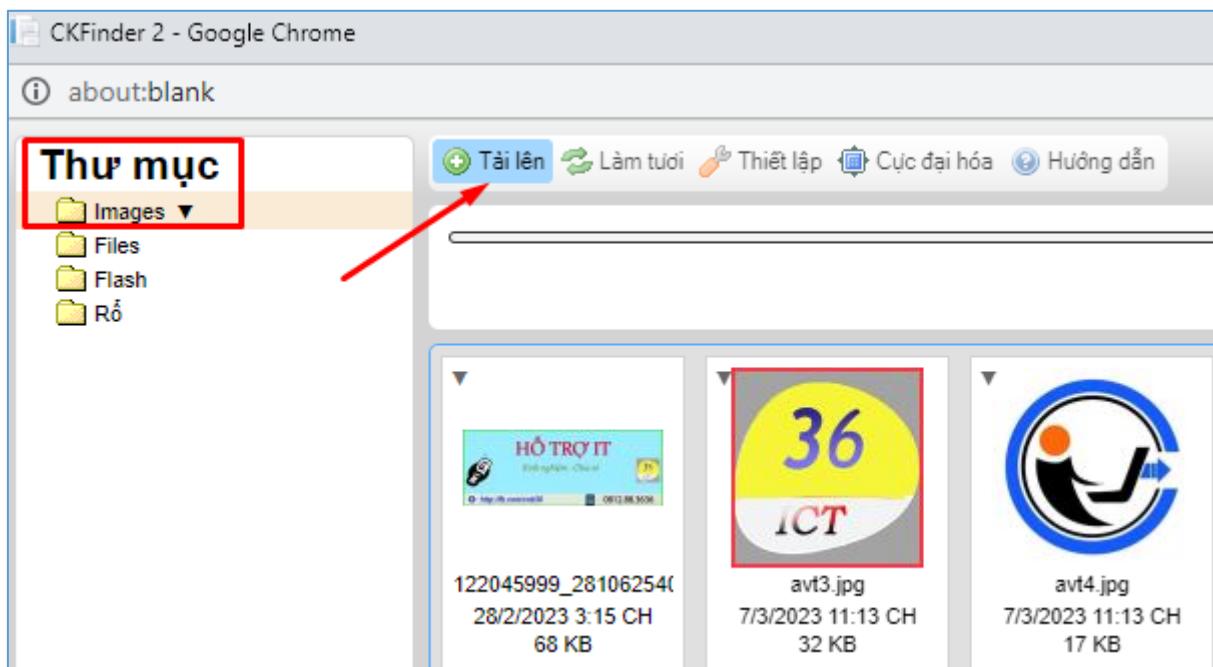
Trang quản trị						
Thêm mới						
Tên sản phẩm	<input type="text"/>					
Giá tiền	<input type="text"/>					
Mô tả	<div style="border: 1px solid #ccc; padding: 5px;"> Mã HTML Font Color Size Align Justify Link Image Table Image Code Find Replace Print Help </div> <div style="border: 1px solid #ccc; height: 100px; margin-top: 5px;"></div>					
Là sản phẩm mới	<input checked="" type="checkbox"/>					
Hình ảnh sản phẩm	<input type="file"/> Upload					
MaDanhMuc	<input type="text" value="Xiaomi"/>					
Số lượng	<input type="text"/>					
	<input type="button" value="Thêm"/>					
Quay lại						

- Thực hiện thêm mới sản phẩm có đính kèm hình ảnh và nhập dữ liệu thông tin mô tả sản phẩm:

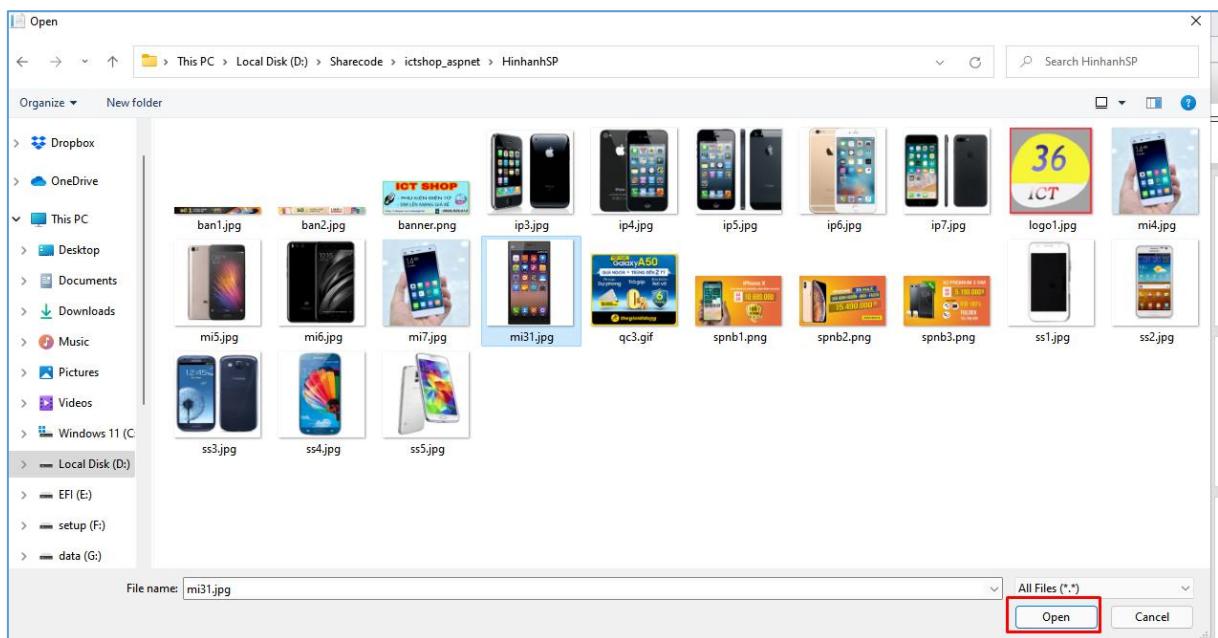
+ Để thêm mới hình ảnh chọn Upload:

<input checked="" type="checkbox"/> Là sản phẩm mới	<input type="checkbox"/>
Hình ảnh sản phẩm	<input type="file"/>
	<input type="button" value="Upload"/>
MaDanhMuc	<input type="text" value="Xiaomi"/>
Số lượng	<input type="text"/>
<input type="button" value="Thêm"/>	

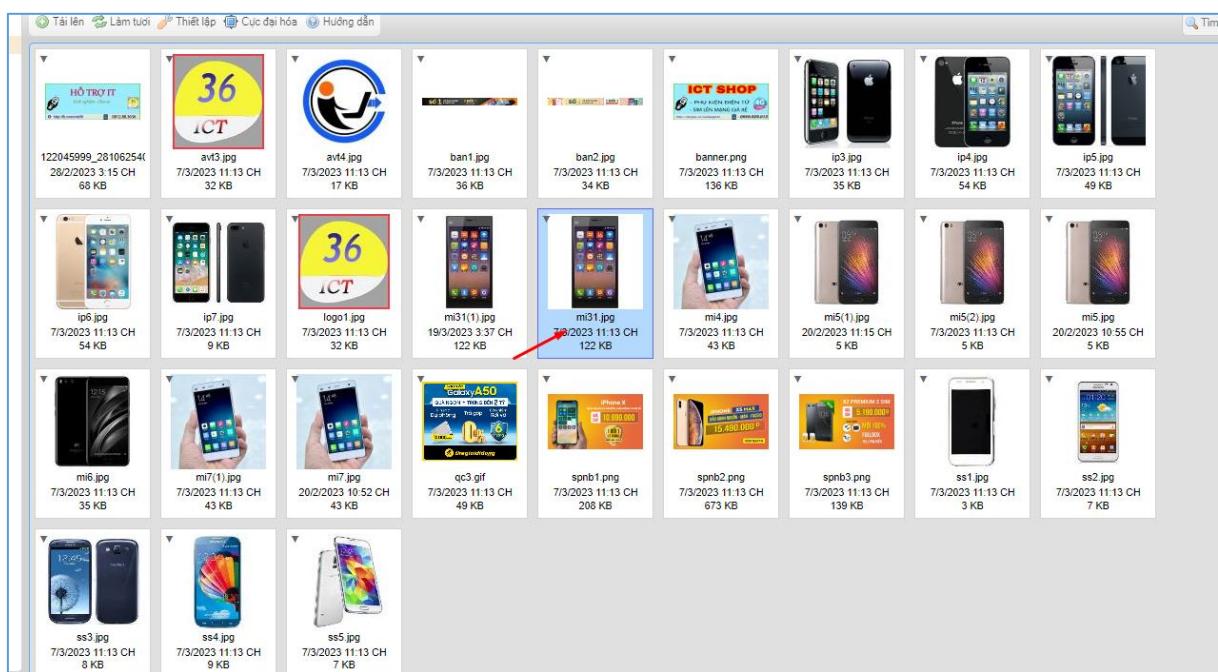
- Tại Thư mục -> Images
- Chọn tải lên:



- Chọn upload hình ảnh từ máy tính và chọn open



- Hiển thị danh sách các hình ảnh đã tải lên:



- Kích đúp chuột vào hình ảnh đã chọn

Thêm mới

Tên sản phẩm: Xiaomi MI3

Giá tiền: 1000000

Mô tả:

NHẬP MÔ TẢ

Xiaomi MI3

- Ram 4 gb
- Bộ nhớ 32gb

Là sản phẩm mới:

Hình ảnh sản phẩm: ĐƯỜNG DẪN ẢNH CỦA SẢN PHẨM

Upload

MaDanhMuc: Xiaomi

Số lượng: 10 The Số lượng field is required.

Thêm

- Bấm thêm mới để thêm mới sản phẩm.

- Thêm mới thành công:

Trang quản trị								
Danh sách								
Thêm mới								
Tên sản phẩm: <input type="text"/> <input type="button" value="Tim kiếm"/>								
Tên danh mục	Tên sản phẩm	Giá tiền	Mô tả	Là sản phẩm mới	Hình ảnh sản phẩm	Số lượng		
Xiaomi	Xiaomi MI3	1,000,000 VND	<p>Xiaomi MI3</p> <p>- Ram 4gb</p> <p>- Bộ nhớ 32gb</p>	<input checked="" type="checkbox" value="True"/>	/Uploads/images/mi31.jpg	10	Sửa	Chi tiết

Trang 1 / 1

1

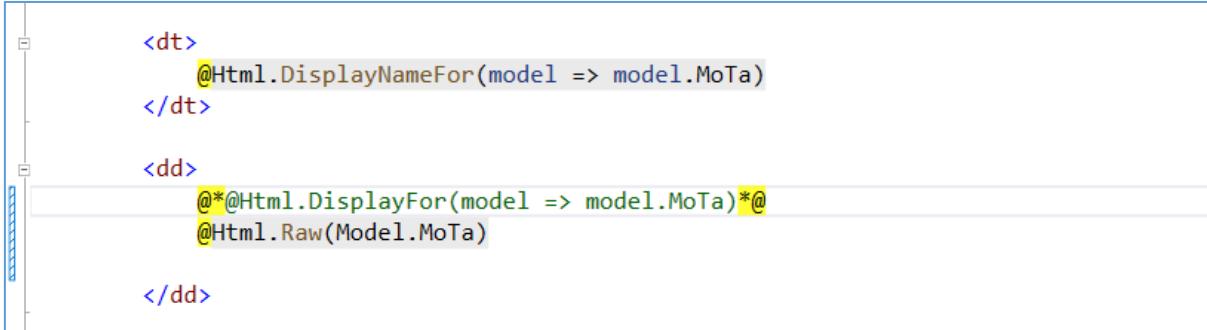
- Bấm vào Chi tiết để xem chi tiết sản phẩm đã thêm mới:

Chi tiết dữ liệu

Tên danh mục	Xiaomi
Tên sản phẩm	Xiaomi MI3
Giá tiền	1,000,000 VND
Mô tả	<p>Xiaomi MI3</p> <p>- Ram 4gb</p> <p>- Bộ nhớ 32gb</p>
Là sản phẩm mới	True
Hình ảnh sản phẩm	/Uploads/Images/mi31.jpg
Số lượng	10

[Sửa](#) | [Quay lại](#)

- Phần mô tả đang hiển thị định dạng các thẻ html, chính vì vậy chúng ta cần sử dụng: **@Html.Raw()** để hiển thị nội dung mô tả.
- Tại view Details của Sản phẩm, chỉnh sửa thành:



- Chú ý các trang hiển thị mô tả của sản phẩm, chúng ta đều cần sử dụng : **@Html.Raw()** để hiển thị nội dung.
- Code hiển thị trang Index sản phẩm:

```

@foreach (var item in Model)
{
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.DanhMucSanPham.TenDanhMuc)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.TenSP)
        </td>
        <td>
            @String.Format("{0:0,0}", item.GiaTien) VNĐ
        </td>
        <td>
            @Html.Raw(item.MoTa)
        </td>
        <td style="text-align: center">
            @Html.DisplayFor(modelItem => item.LaSanPhamMoi)
        </td>

        <td style="text-align: center">
            
            @* @Html.DisplayFor(modelItem => item.HinhAnh) *@
        </td>

        <td style="text-align: center">
            @Html.DisplayFor(modelItem => item.SoLuong)
        </td>
        <td>
            @Html.ActionLink("Sửa", "Edit", new { id = item.MaSanPham }) |
            @Html.ActionLink("Chi tiết", "Details", new { id = item.MaSanPham }) |
            @Html.ActionLink("Xoá", "Delete", new { id = item.MaSanPham })
        </td>
    </tr>
}

```

- Kết quả:

Trang quản trị Danh mục Sản phẩm Phân Quyền Người Dùng

Danh sách

Thêm mới

Tên sản phẩm:

Tên danh mục	Tên sản phẩm	Giá tiền	Mô tả	Là sản phẩm mới	Hình ảnh sản phẩm	Số lượng	
Xiaomi	Xiaomi MI3	1,000,000 VNĐ	Xiaomi MI3 - Ram 4 gb - Bộ nhớ 32gb	True <input type="button" value="▼"/>		10	Sửa Chi tiết Xoá

Trang 1 / 1

1

- Sinh viên tiến hành thêm mới các sản phẩm khác.

- Có thể lấy các hình ảnh sản phẩm tại:

Link tải:

https://bit.ly/LabLapTrinhWeb_FictHdu

+ Thư mục **Download/images**

HƯỚNG DẪN THỰC HÀNH BUỔI SỐ 4

I. Mục tiêu

- Hiểu được khái niệm về session và cách sử dụng session
- Sử dụng session xây dựng được trang đăng nhập cho website quản trị
- Xây dựng được chức năng đăng xuất tài khoản hiện tại.
- Mã hoá được mật khẩu người dùng khi lưu vào Database
- Hoàn thiện được website quản trị.

II. Nội dung

- Xây dựng trang đăng nhập cho Website quản trị sử dụng session
- Xây dựng chức năng đăng xuất tài khoản hiện tại
- Mã hoá mật khẩu người dùng khi lưu vào Database
- Hoàn thành trang quản trị sản phẩm (Chỉnh sửa các label hiển thị, css để các button, văn bản, website bắt mắt, thân thiện với người dùng.)

4. Hướng dẫn thực hành: Hoàn thiện trang quản trị

4.1. Xây dựng trang đăng nhập cho website quản trị sử dụng session

- Tại của *Areas/Admin/Controllers/NguoiDungController.cs* tiến hành thêm mới hàm Login.

```
// GET Admin/NguoiDung/Login
public ActionResult Login()
{
    return View();
}

// POST: Admin/NguoiDung/Login
[HttpPost]
public ActionResult Login(FormCollection userForm)
{
    return View();
}
```

- Add view cho phương thức login:

+Form bao gồm: input nhập tên tài khoản, input nhập mật khẩu và button Đăng nhập.

```

@model QuanLyBanHang.Models.NguoiDung

@{
    ViewBag.Title = "Login";
    Layout = "~/Areas/Admin/Views/Shared/_Layout.cshtml";
}
</br>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h3><center>ĐĂNG NHẬP TRANG QUẢN TRỊ</center></h3>
        <hr />
        <div class="form-group">
            @Html.Label("Tên đăng nhập", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.TenDangNhap, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.TenDangNhap, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.Label("Mật khẩu", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.MatKhau, new { htmlAttributes = new { type = "Password", @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.MatKhau, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Đăng nhập" class="btn btn-default" />
            </div>
        </div>
        <h3 class="text-danger">@ViewBag.Fail</h3>
    </div>
}

```

- Chú ý: input nhập mật khẩu cần thêm thuộc tính **type="Password"** để ẩn mật khẩu khi nhập
- Thêm phương thức HttpPost login (Nhận giá trị từ view và xử lý đăng nhập):
 - + Sử dụng Session để lưu lại giá trị của người dùng đăng nhập:

```

// POST: Admin/NguoiDung/Login
[HttpPost]
public ActionResult Login(FormCollection userForm)
{
    string tendangnhapform = userForm["TenDangNhap"].ToString();
    string matkhauForm = userForm["MatKhau"].ToString();
    var nguoiDung = db.NguoiDungs.SingleOrDefault(x =>
x.TenDangNhap.Equals(tendangnhapform) &&
x.MatKhau.Equals(matkhauForm));
    if (nguoidung != null)
    {
        Session["user"] = nguoiDung;
        return RedirectToAction("Index", "Home");
    }
    else
    {
        ViewBag.Fail = "Tài khoản hoặc mật khẩu không chính xác.";
        return View("Login");
    }
}

```

Build dự án và tiến hành khởi chạy dự án:

- Trang đăng nhập:

The screenshot shows a login form titled "ĐĂNG NHẬP TRANG QUẢN TRỊ". It has two input fields: "Tên đăng nhập" and "Mật khẩu", and a "Đăng nhập" button. Below the form is a copyright notice: "© 2023 - My ASP.NET Application".

- Thực hiện thêm mới 1 người dùng **admin** trong Menu Quản trị Người dùng:

The screenshot shows a list of users. There is one entry for "Admin" with the details: Họ tên: Admin, Địa chỉ: Thanh Hoá, Email: admin@gmail.com, Số điện thoại: 0123445566, Tên đăng nhập: admin, Mật khẩu: 12345678, and Hình đại diện: /hinhanh. A "Sửa | Chi tiết | Xoá" link is also present.

Tên Quyền	Họ tên	Địa chỉ	Email	Số điện thoại	Tên đăng nhập	Mật khẩu	Ảnh đại diện	
Admin	Admin	Thanh Hoá	admin@gmail.com	0123445566	admin	12345678	/hinhanh	Sửa Chi tiết Xoá

- Tiến hàng đăng nhập hệ thống bằng người dùng admin:

Nếu đăng nhập tài khoản không đúng với tài khoản trong bảng NguoiDung, sử dụng **ViewBag** hiển thị thông báo Tài khoản hoặc mật khẩu không chính xác:

The screenshot shows the login page again. Below the form, there is a red error message: "Tài khoản hoặc mật khẩu không chính xác." (Account or password is incorrect).

- Đăng nhập thành công: `return RedirectToAction("Index", "Home");`

Trang quản trị website quản lý bán hàng

© 2023 - My ASP.NET Application

- Để hiển thị: "Xin chào user đang đăng nhập". Cần chỉnh sửa trong `_Layout.cshtml` của Admin như sau:

```
<body>
    @if (var user = Session["user"] as QuanLyBanHang.Models.NguoiDung; )
        <div class="navbar navbar-inverse navbar-fixed-top">
            <div class="container">
                <div class="navbar-header">
                    <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
                        <span class="icon-bar"></span>
                        <span class="icon-bar"></span>
                        <span class="icon-bar"></span>
                    </button>
                    @Html.ActionLink("Trang quản trị", "Index", "Home", new { area = "Admin" }, new { })
                </div>
                <div class="navbar-collapse collapse">

                    <ul class="nav navbar-nav">
                        @if (user != null)
                        {
                            <li>@Html.ActionLink("Danh mục", "Index", "DanhMucSanPham")</li>
                            <li>@Html.ActionLink("Sản phẩm", "Index", "SanPham")</li>
                            <li>@Html.ActionLink("Phân Quyền", "Index", "PhanQuyen")</li>
                            <li>@Html.ActionLink("Người Dùng", "Index", "NguoiDung")</li>
                            <li> <a style="color:ghostwhite" href="#">Xin chào : @user.HoTen </a>
                            <li> @Html.ActionLink("Đăng xuất", "Logout", "NguoiDung") </li>
                        }
                    </ul>
                </div>
            </div>
        </div>
    
```

- Kết quả:

Xin chào : Admin

Đăng xuất

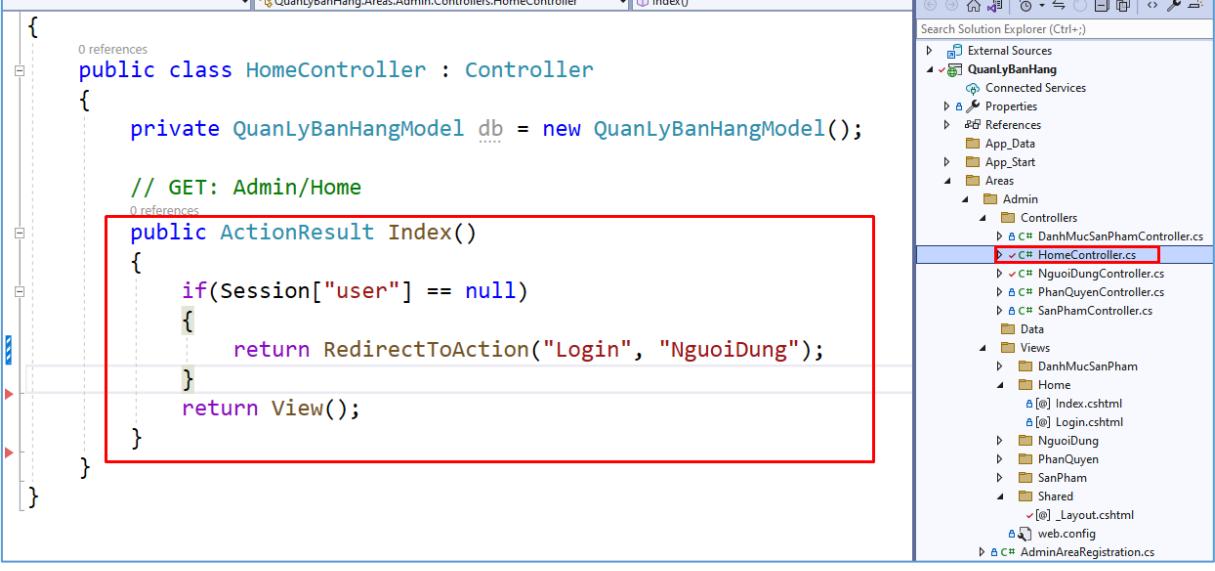
Trang quản trị website quản lý bán hàng

© 2023 - My ASP.NET Application

4.2. Xây dựng chức năng đăng xuất tài khoản hiện tại

- Tại *Areas/Admin/Controllers/HomeController.cs*:

+ Chính sửa hàm Index thành như sau:



```
{  
    public class HomeController : Controller  
    {  
        private QuanLyBanHangModel db = new QuanLyBanHangModel();  
  
        // GET: Admin/Home  
        public ActionResult Index()  
        {  
            if(Session["user"] == null)  
            {  
                return RedirectToAction("Login", "NguoiDung");  
            }  
            return View();  
        }  
    }  
}
```

The Solution Explorer on the right shows the project structure for 'QuanLyBanHang' with 'Admin' and 'Home' areas.

- Xây dựng phương thức Logout (Đăng xuất tài khoản)

+ Khi bấm vào Đăng xuất, sẽ thiết lập lại session user là null. Và chuyển hướng về trang chủ quản trị

+ Tại : *Areas/Admin/Controllers/NguoiDungController.cs* thêm mới:

```
public ActionResult Logout()  
{  
    Session["user"] = null;  
    return RedirectToAction("Index", "Home");  
}
```

4.3. Mã hóa mật khẩu khi lưu vào database

- Tại dự án thêm mới thư mục Common

- Tạo class MaHoaMD5.cs trong thư mục Common:

```

using static System.Net.Mime.MediaTypeNames;

namespace QuanLyBanHang.Common
{
    public class MaHoaMD5
    {
        //create a string MD5
        public static string GetMD5(string str)
        {
            MD5 md5 = new MD5CryptoServiceProvider();
            //compute hash from the bytes of text
            md5.ComputeHash(ASCIIEncoding.ASCII.GetBytes(str));
            //get hash result after compute it
            byte[] result = md5.Hash;
            StringBuilder strBuilder = new StringBuilder();
            for (int i = 0; i < result.Length; i++)
            {
                //change it into 2 hexadecimal digits
                //for each byte
                strBuilder.Append(result[i].ToString("x2"));
            }
            return strBuilder.ToString();
        }
    }
}

```

- *MaHoaMD5.cs*:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Web;
using static System.Net.Mime.MediaTypeNames;

namespace QuanLyBanHang.Common
{
    public class MaHoaMD5
    {
        //create a string MD5
        public static string GetMD5(string str)
        {
            MD5 md5 = new MD5CryptoServiceProvider();
            //compute hash from the bytes of text
            md5.ComputeHash(ASCIIEncoding.ASCII.GetBytes(str));
            //get hash result after compute it
            byte[] result = md5.Hash;
            StringBuilder strBuilder = new StringBuilder();
            for (int i = 0; i < result.Length; i++)
            {
                //change it into 2 hexadecimal digits
                //for each byte
                strBuilder.Append(result[i].ToString("x2"));
            }
            return strBuilder.ToString();
        }
    }
}

```

- Tại *Areas/Admin/Controllers/NguoiDungController.cs* cần thêm:

```
using PagedList;
using QuanLyBanHang.Common;
using QuanLyBanHang.Models;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web.Mvc;
namespace QuanLyBanHang.Areas.Admin.Controllers
{
```

- Tại *Areas/Admin/Controllers/NguoiDungController.cs*

+ Chính sửa code hàm Create và Edit:

```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "MaNguoiDung,HoTen,DiaChi,Email,SoDienThoai,TenDangNhap,MaQuyen")]
{
    if (ModelState.IsValid)
    {
        nguoiDung.MatKhau = (MaHoaMD5.GetMD5(nguoIDung.MatKhau));
        db.Configuration.ValidateOnSaveEnabled = false;
        db.NguoiDungs.Add(nguoIDung);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    ViewBag.MaQuyen = new SelectList(db.PhanQuyens, "MaQuyen", "TenQuyen", nguoIDung.MaQuyen);
    return View(nguoIDung);
}

// GET: Admin/NguoiDung/Edit/5
public ActionResult Edit(int? id){...}

// POST: Admin/NguoiDung/Edit/5
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "MaNguoiDung,HoTen,DiaChi,Email,SoDienThoai,TenDangNhap,MaQuyen")]
{
    if (ModelState.IsValid)
    {
        nguoiDung.MatKhau = (MaHoaMD5.GetMD5(nguoIDung.MatKhau));
        db.Configuration.ValidateOnSaveEnabled = false;
        db.Entry(nguoIDung).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
}
```

- Truy cập vào Menu Quản lý người dùng: truy cập vào chức năng Sửa:

Danh sách

Thêm mới

Tên người dùng: Tìm kiếm

Tên Quyền	Họ tên	Địa chỉ	Email	Số điện thoại	Tên đăng nhập	Mật khẩu	Ảnh đại diện	
Admin	Admin	Thanh Hoá	admin@gmail.com	0123445566	admin	12345678	/hinhanh	Sửa Chi tiết Xoá

Trang 1 / 1

1

- Thực hiện bấm Lưu:

Trang quản trị Danh mục Sản phẩm Phân Quyền Người Dùng Xin chào : Admin Đăng xuất

Chỉnh sửa dữ liệu

Họ tên	<input type="text" value="Admin"/>
Địa chỉ	<input type="text" value="Thanh Hoá"/>
Email	<input type="text" value="admin@gmail.com"/>
Số điện thoại	<input type="text" value="0123445566"/>
Tên đăng nhập	<input type="text" value="admin"/>
Mật khẩu	<input type="text" value="12345678"/>
Ảnh đại diện	<input type="text" value="/hinhanh"/>
Quyền	<input type="text" value="Admin"/>

Quay lại

- Mật khẩu đã được Mã hoá:

Danh sách

Thêm mới

Tên người dùng: Tìm kiếm

Tên Quyền	Họ tên	Địa chỉ	Email	Số điện thoại	Tên đăng nhập	Mật khẩu
Admin	Admin	Thanh Hoá	admin@gmail.com	0123445566	admin	<input type="text" value="25d55ad283aa400af464c76d713c07ad"/>

Trang 1 / 1

1

- Tiếp theo, cần thực hiện chỉnh sửa phương thức Login thành:

```
// POST: Admin/NguoiDung/Login
[HttpPost]
public ActionResult Login(FormCollection userForm)
{
    string tendangnhapform = userForm["TenDangNhap"].ToString();
    string matkhauForm = userForm["MatKhau"].ToString();
    var matkhauMD5 = MaHoaMD5.GetMD5(matkhauForm);
    var nguoiDung = db.NguoiDungs.SingleOrDefault(x =>
x.TenDangNhap.Equals(tendangnhapform) && x.MatKhau.Equals(matkhauMD5));
    if (nguoIDung != null)
    {
        Session["user"] = nguoiDung;
        return RedirectToAction("Index", "Home");
    }
    else
    {
        ViewBag.Fail = "Tài khoản hoặc mật khẩu không chính xác.";
        return View("Login");
    }
}
```

- Như vậy chúng ta đã thực hiện thành công Mã hoá mật khẩu khi lưu vào Database.

4.4. Hoàn thiện website quản lý

- Sinh viên hoàn thiện các chức năng của website quản trị.
- Sinh viên sử dụng có thể sử dụng bootstrap, các thư viện UI khác hoặc thay thế _Layout mặc định bằng Template Admin HMTL, CSS khác để website quản trị bắt mắt, thân thiện với người dùng.

HƯỚNG DẪN THỰC HÀNH BUỔI SỐ 5

I. Mục tiêu

- Xây dựng được website dành cho khách hàng
- Sử dụng được PartialView để hiển thị sản phẩm mới
- Hiển thị được danh sách sản phẩm kèm hình ảnh
- Phân trang được danh sách sản phẩm
- Xây dựng được trang chi tiết sản phẩm

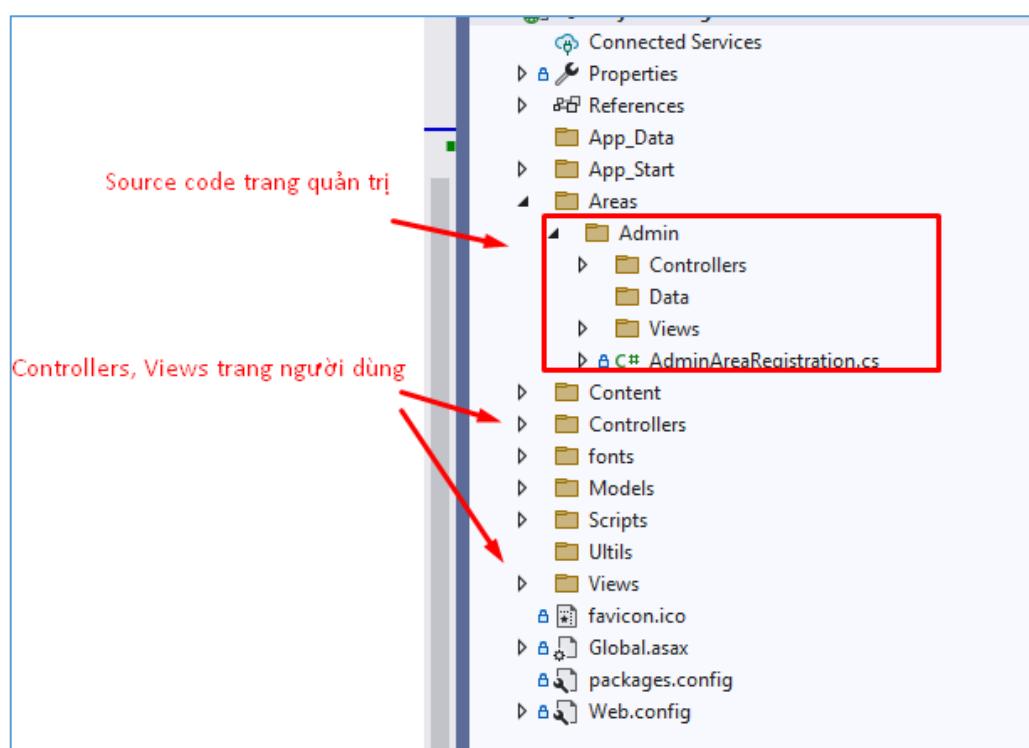
II. Nội dung

- Xây dựng website dành cho khách hàng
- Hiển thị sản phẩm mới sử dụng PartialView
- Hiển thị danh sách sản phẩm
- Phân trang danh sách sản phẩm
- Xây dựng trang chi tiết sản phẩm

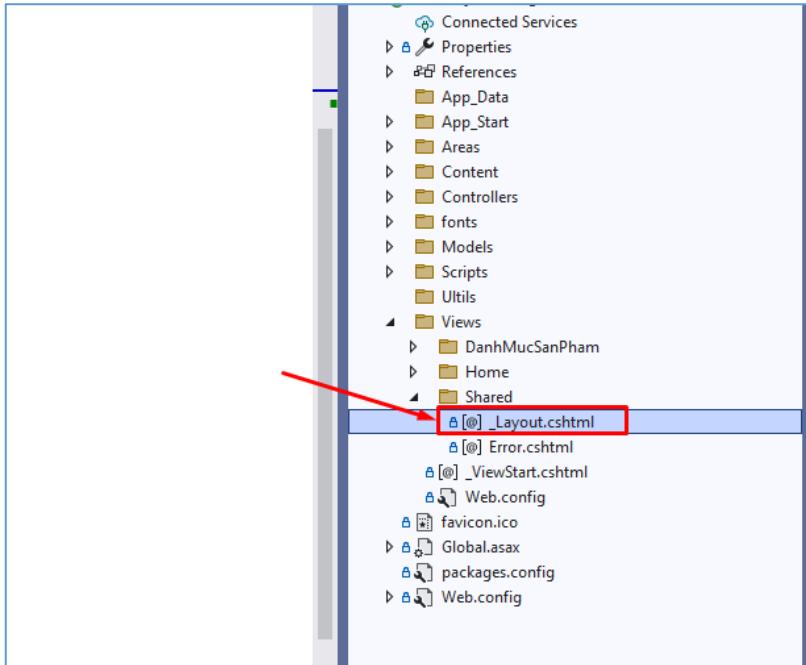
5. Xây dựng trang dành cho người dùng

5.1. Xây dựng trang dành cho người dùng

- Cấu trúc source code:



- Tại _layout.cshtml của người dùng:



- Chính sửa label các menu thành:

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>@ViewBag.Title - My ASP.NET Application</title>
        @Styles.Render("~/Content/css")
        @Scripts.Render("~/bundles/modernizr")
    </head>
    <body>

        <div class="navbar navbar-inverse navbar-fixed-top">
            <div class="container">
                <div class="navbar-header">
                    <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse" title="more options">
                        <span class="icon-bar"></span>
                        <span class="icon-bar"></span>
                        <span class="icon-bar"></span>
                    </button>
                    @Html.ActionLink("ICT SHOP", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })
                </div>
                <div class="navbar-collapse collapse">
                    <ul class="nav navbar-nav">
                        <li>@Html.ActionLink("Trang chủ", "Index", "Home")</li>
                        <li>@Html.ActionLink("Giới thiệu", "About", "Home")</li>
                        <li>@Html.ActionLink("Liên hệ", "Contact", "Home")</li>
                    </ul>
                </div>
            </div>
        <div class="container body-content">
            @RenderBody()
            <hr />
            <footer>
                <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
            </footer>
        </div>

        @Scripts.Render("~/bundles/jquery")
        @Scripts.Render("~/bundles/bootstrap")
        @RenderSection("scripts", required: false)
    </body>

```

5.2. Sử dụng PartialView để hiển thị sản phẩm

- Tại Controllers/HomeController.cs . Code như sau:

```
using PagedList;
using QuanLyBanHang.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;

namespace QuanLyBanHang.Controllers
{
    public class HomeController : Controller
    {
        QuanLyBanHangModel db = new QuanLyBanHangModel();
        // GET:
        public ActionResult Index()
        {
            return View();
        }
        public ActionResult SanPhamMoiPartial()
        {
            var spMoi = db.SanPhams.Where(n => n.LaSanPhamMoi == true).Take(3).OrderBy(x => x.MaSanPham).ToList();
            return PartialView(spMoi);
        }

        public ActionResult SanPhamTrangChuPartial()
        {
            var spTrangChu = db.SanPhams.Take(6).OrderBy(x => x.MaSanPham).ToList();
            return PartialView(spTrangChu);
        }

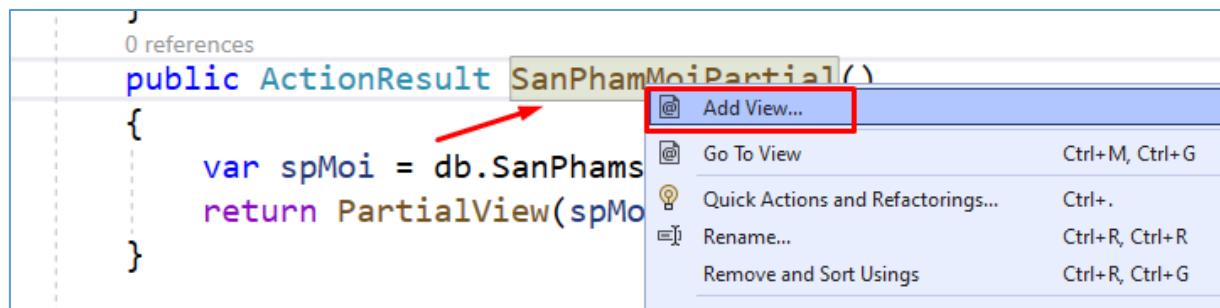
        public ActionResult About()
        {
            ViewBag.Message = "Your application description page.";

            return View();
        }

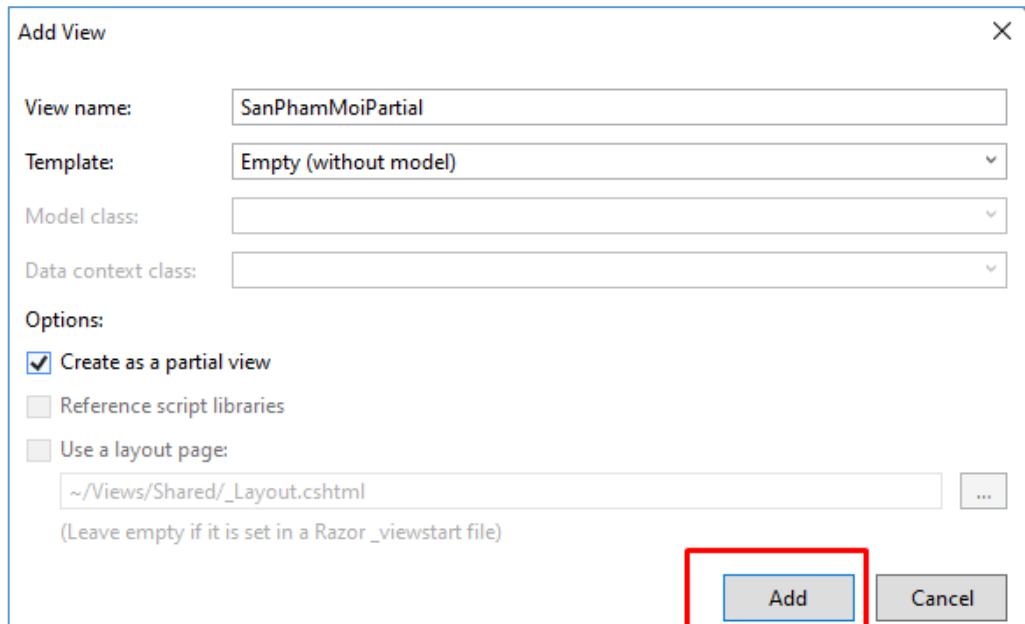
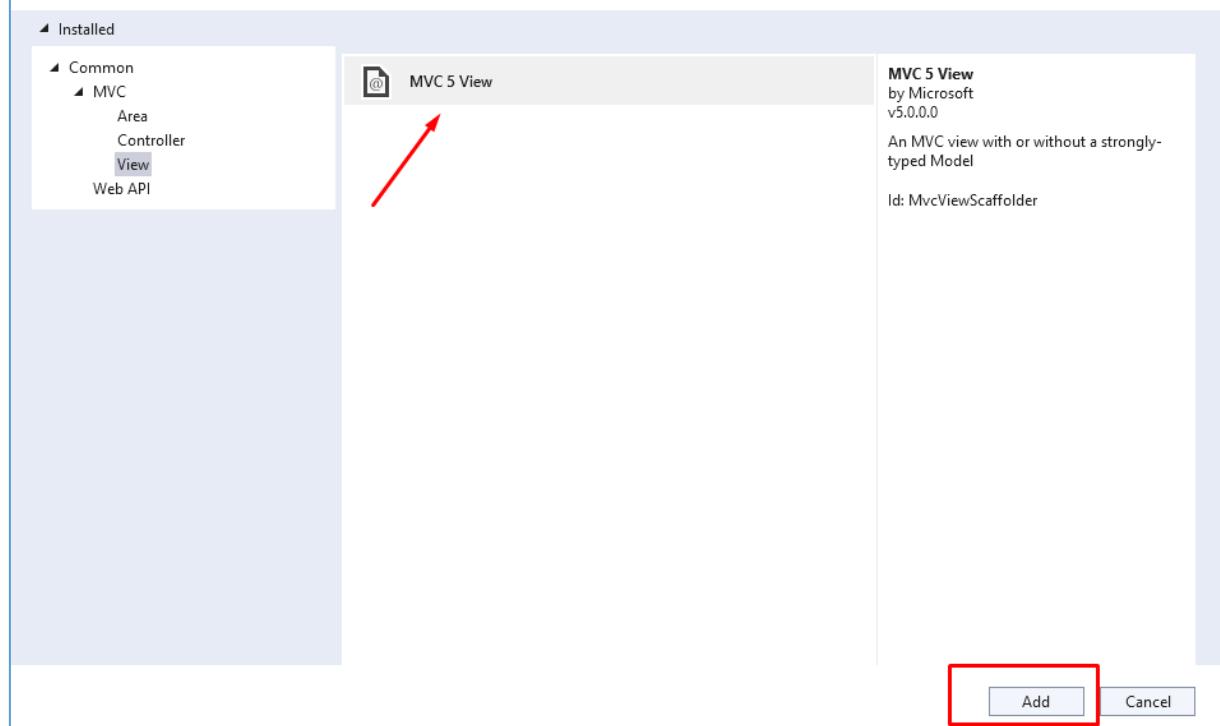
        public ActionResult Contact()
        {
            ViewBag.Message = "Your contact page.";

            return View();
        }
    }
}
```

- Add View cho SanPhamMoiPartial:



Add New Scaffolded Item



- Chính sửa View SanPhamMoiPartial.cshtml:

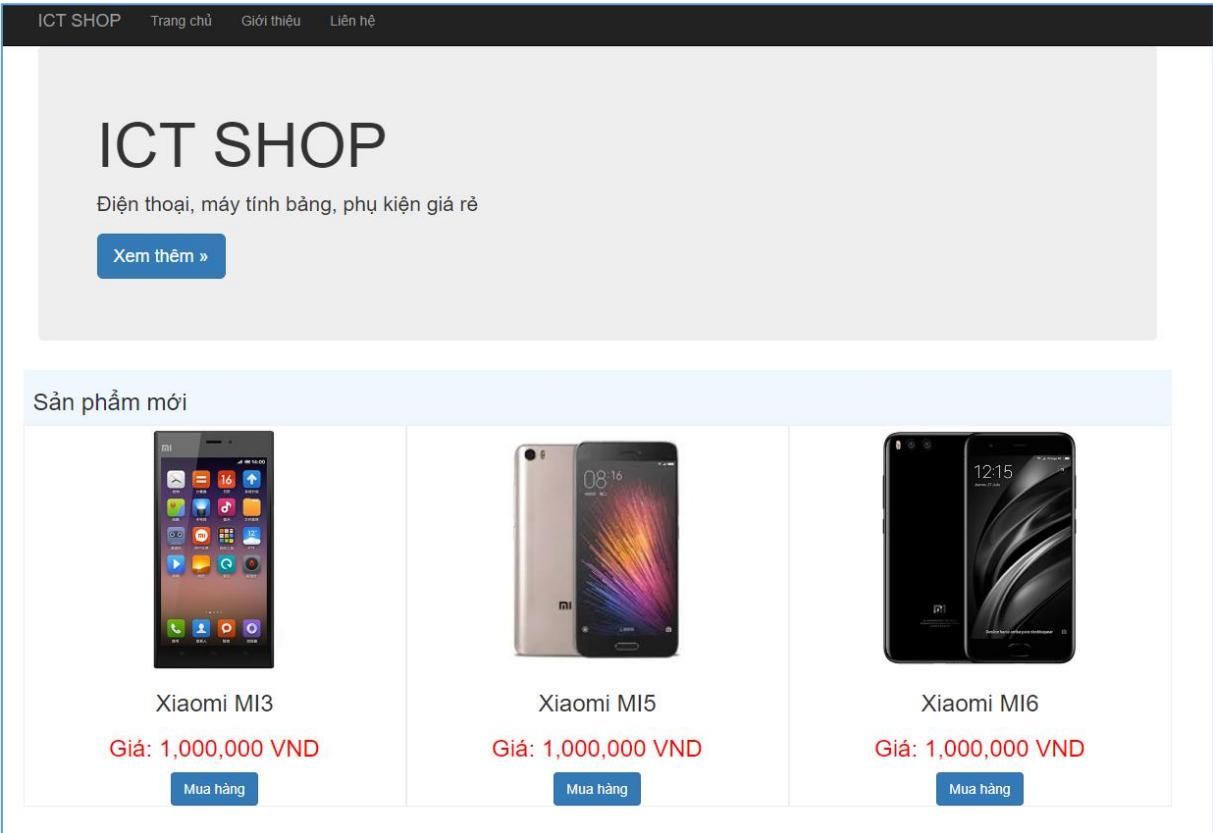
```
<div class="row" style="background-color:aliceblue">
    <h3 style="margin-left:10px"> Sản phẩm mới </h3>
</div>
<div class="row">
    @foreach (var item in Model)
    {
        <div class="col-md-4 col-sm-6" style="border-width: 0.5px; border-style:groove; margin-bottom:15px">
            <center>
                <a href="@Url.Action("ChiTietSanPham", "SanPham", new { @id=item.MaSanPham })">
                    
                </a>
                <h3>@item.TenSP</h3>
                <h3 style="color:red">Giá: @String.Format("{0:0,0}", @item.GiaTien) VND</h3>
                @using (Html.BeginForm("ThemGioHang", "GioHang", new { @iMaSanPham = item.MaSanPham, @strURL = Request.Url.ToString() }))
                {
                    <button type="submit" class="btn btn-primary">Mua hàng</button>
                }
            </center>
        </div>
    }
</div>
```

- Chính sửa: Views/Home/Index.cshtml

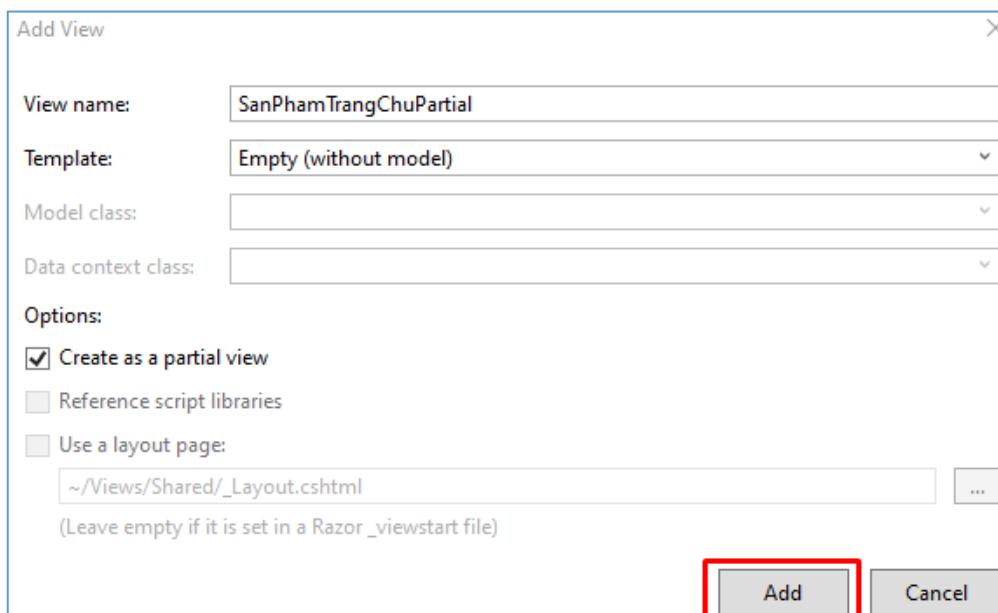
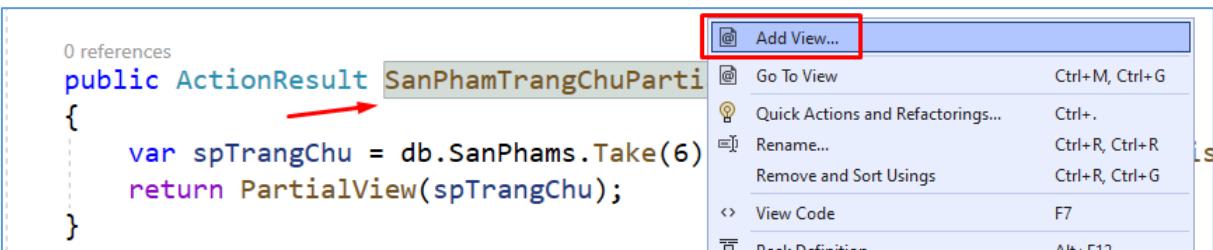
```
@{
    ViewBag.Title = "Home Page";
}
<div class="jumbotron">
    <h1>ICT SHOP</h1>
    <p class="lead"> Điện thoại, máy tính bảng, phụ kiện giá rẻ</p>
    <p><a href="#" class="btn btn-primary btn-lg">Xem thêm &raquo;</a></p>
</div>
<div class="sanphammoi">
    @Html.Action("SanPhamMoiPartial", "Home")
</div>
```

- Build lại Solution

- Chạy dự án:



- Tương tự: Add View cho SanPhamTrangChuPartial:



- Chính sửa View SanPhamTrangChuPartial.cshtml:

```
<div class="row" style="background-color:aliceblue">
    <h3 style="margin-left:10px"> Danh sách sản phẩm </h3>
</div>
<div class="row">
    @foreach (var item in Model)
    {
        <div class="col-md-4 col-sm-6" style="border-width: 0.5px; border-style:groove; margin-bottom:15px">
            <center>
                <a href="@Url.Action("ChiTietSanPham", "SanPham", new { id=item.MaSanPham })">
                    
                </a>
                <h3>@item.TenSP</h3>
                <h3 style="color:red">Giá: @String.Format("{0:0,0}", @item.GiaTien) VND</h3>
                @using (Html.BeginForm("ThemGioHang", "GioHang", new { @iMaSanPham = item.MaSanPham, @strURL = Request.Url.ToString() }))
                {
                    <button type="submit" class="btn btn-primary">Mua hàng</button>
                }
            </center>
        </div>
    }
    <br>
    <center>
        @Html.ActionLink("Xem tất cả...", "Index", "SanPham")
    </center>

```

- Chính sửa: Views/Home/Index.cshtml

```
@{
    ViewBag.Title = "Home Page";
}
<div class="jumbotron">
    <h1>ICT SHOP</h1>
    <p class="lead">Điện thoại, máy tính bảng, phụ kiện giá rẻ</p>
    <p><a href="#" class="btn btn-primary btn-lg">Xem thêm &raquo;</a></p>
</div>
<div class="sanphammoi">
    @Html.Action("SanPhamMoiPartial", "Home")
</div>
<div class="danhsachsanpham">
    @Html.Action("SanPhamTrangChuPartial", "Home")
</div>
```

- Build lại Solution

- Chạy dự án:

- Trang chủ hiển thị:

ICT SHOP Trang chủ Giới thiệu Liên hệ

ICT SHOP

Điện thoại, máy tính bảng, phu kiện giá rẻ

Xem thêm »

Sản phẩm mới

		
Xiaomi MI3	Xiaomi MI5	Xiaomi MI6
Giá: 1,000,000 VND	Giá: 1,000,000 VND	Giá: 1,000,000 VND
Mua hàng	Mua hàng	Mua hàng

Danh sách sản phẩm

		
Xiaomi MI3	Xiaomi MI5	Xiaomi MI6
Giá: 1,000,000 VND	Giá: 1,000,000 VND	Giá: 1,000,000 VND
Mua hàng	Mua hàng	Mua hàng
		
Iphone 3	Iphone 4	Iphone 5
Giá: 1,000,000 VND	Giá: 2,000,000 VND	Giá: 3,000,000 VND
Mua hàng	Mua hàng	Mua hàng

Xem tất cả...

© 2023 - My ASP.NET Application

5.3. Xây dựng trang chi tiết sản phẩm

- Thêm mới SanPhamController:

The screenshot shows the Visual Studio interface. On the left is the code editor with the content of the SanPhamController.cs file. On the right is the Solution Explorer window, which displays the project structure. A red box highlights the 'SanPhamController.cs' file in the 'Controllers' folder under the 'Admin' directory.

```

namespace QuanLyBanHang.Controllers
{
    public class SanPhamController : Controller
    {
        QuanLyBanHangModel db = new QuanLyBanHangModel();
        // GET: Admin/SanPham
        public ActionResult Index(int? page)
        {
            if (page == null) page = 1;
            var sanPhams = db.SanPhams.OrderBy(x => x.MaSanPham);
            int pageSize = 6;
            int pageNumber = (page ?? 1);
            return View(sanPhams.ToPagedList(pageNumber, pageSize));
        }

        // GET: Admin/SanPham/ChiTietSanPham/5
        public ActionResult ChiTietSanPham(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            SanPham sanPham = db.SanPhams.Find(id);
            if (sanPham == null)
            {
                return HttpNotFound();
            }
            return View(sanPham);
        }
    }
}

using QuanLyBanHang.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using PagedList;
namespace QuanLyBanHang.Controllers
{
    public class SanPhamController : Controller
    {
        QuanLyBanHangModel db = new QuanLyBanHangModel();
        // GET: Admin/SanPham
        public ActionResult Index(int? page)
        {
            if (page == null) page = 1;
            var sanPhams = db.SanPhams.OrderBy(x => x.MaSanPham);
            int pageSize = 6;
            int pageNumber = (page ?? 1);
            return View(sanPhams.ToPagedList(pageNumber, pageSize));
        }
        // GET: Admin/SanPham/ChiTietSanPham/5
        public ActionResult ChiTietSanPham(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            SanPham sanPham = db.SanPhams.Find(id);
            if (sanPham == null)
            {
                return HttpNotFound();
            }
            return View(sanPham);
        }
    }
}

```

- Add View cho ChiTietSanPham (View thông thường – không phải Partialview). Mẫu:

```
@model QuanLyBanHang.Models.SanPham
@{
    ViewBag.Title = "ChiTietSanPham";
    Layout = "~/Views/Shared/_Layout.cshtml";
}


<h2>Chi tiết sản phẩm</h2>



<h4>
        <div class="row">
            <div class="col-md-6">
                <dl>
                    <dt>
                        @Html.DisplayNameFor(model => model.TenSP)
                    </dt>
                    <dd>
                        @Html.DisplayFor(model => model.TenSP)
                    </dd>
                    <dt>
                        @Html.DisplayNameFor(model => model.HinhAnh)
                    </dt>
                    <dd>
                        <img src=@Url.Content("~/" + @Model.HinhAnh.ToString())
class="img-responsive" alt="" width="250" height="250" />
                    </dd>
                    <dt>
                        @Html.DisplayNameFor(model => model.DanhMucSanPham.TenDanhMuc)
                    </dt>
                    <dd>
                        @Html.DisplayFor(model => model.DanhMucSanPham.TenDanhMuc)
                    </dd>
                    <dt>
                        @Html.DisplayNameFor(model => model.MoTa)
                    </dt>
                    <dd>
                        @Html.Raw(@Model.MoTa)
                    </dd>
                    <dt>
                        @Html.DisplayNameFor(model => model.SoLuong)
                    </dt>
                    <dd>
                        @Html.DisplayFor(model => model.SoLuong)
                    </dd>
                </dl>
            </div>
            <div class="col-md-6">
                <dl>
                    <dt>
                        @Html.DisplayNameFor(model => model.GiaTien)
                    </dt>
                    <dd>
                        <h2> @Html.DisplayFor(model => model.GiaTien) VND</h2>
                    </dd>
                </dl>
                @using (Html.BeginForm("ThemGioHang", "GioHang", new {
                    @iMaSanPham = Model.MaSanPham, @strURL = Request.Url.ToString() }))
                {
                    <button type="submit" class="btn btn-primary">Mua hàng</button>
                }
            </div>
        </div>
    </h4>


```

- Build lại dự án và khởi chạy dự án.
- Click vào hình ảnh sản phẩm:

ICT SHOP Trang chủ Giới thiệu Liên hệ

Chi tiết sản phẩm

Tên sản phẩm	Giá tiền
Xiaomi MI5	1,000,000 VND
Hình ảnh sản phẩm	Mua hàng
	
Tên danh mục	
Xiaomi	
Mô tả	
Xiaomi MI5	
- Ram 4 gb	
- Bộ nhớ 32gb	
Số lượng	
10	

© 2023 - My ASP.NET Application

- Sinh viên có thể code HTML và CSS trang hiển thị danh sách sản phẩm, chi tiết sản phẩm theo thiết kế của cá nhân.

HƯỚNG DẪN THỰC HÀNH BUỔI SỐ 6

I. Mục tiêu

- Xây dựng được class giỏ hàng, các phương thức xử lý giỏ hàng
- Xây dựng được trang giỏ hàng
- Xây dựng được chức năng thêm mới sản phẩm vào giỏ hàng
- Xây dựng được chức năng chỉnh sửa giỏ hàng, đặt hàng

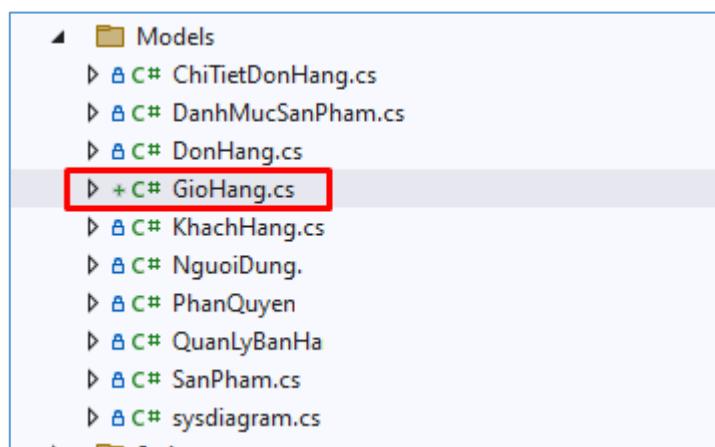
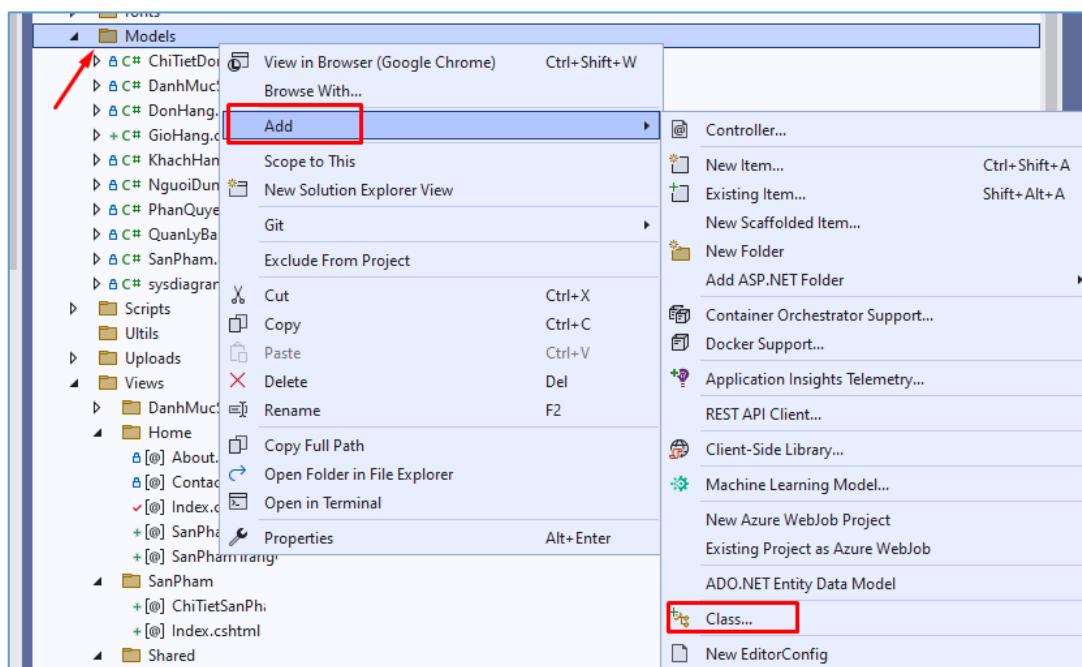
II. Nội dung

- Xây dựng class giỏ hàng, các phương thức xử lý giỏ hàng
- Xây dựng trang giỏ hàng
- Xây dựng chức năng thêm mới sản phẩm vào giỏ hàng
- Xây dựng chức năng chỉnh sửa giỏ hàng, đặt hàng

6. Xây dựng chức năng giỏ hàng

6.1. Xây dựng class giỏ hàng, các phương thức xử lý giỏ hàng

- Tạo class GioHang.cs trong thư mục Models

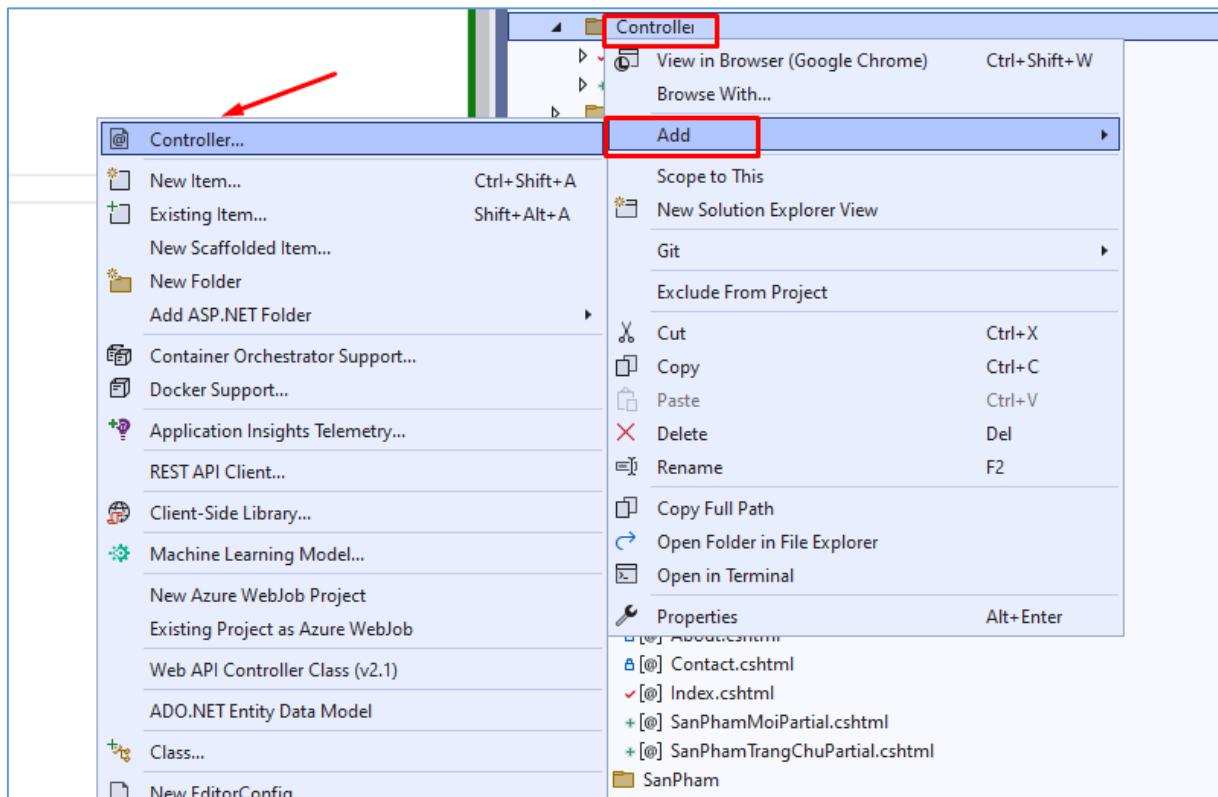


- class GioHang.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace QuanLyBanHang.Models
{
    public class GioHang
    {
        QuanLyBanHangModel db = new QuanLyBanHangModel();
        public int iMaSanPham { get; set; }
        public string sTenSP { get; set; }
        public string sHinhAnh { get; set; }
        public double dGiaTien { get; set; }
        public int iSoLuong { get; set; }
        public double ThanhTien
        {
            get { return iSoLuong * dGiaTien; }
        }
        //Hàm tạo cho giỏ hàng
        public GioHang(int MaSanPham)
        {
            iMaSanPham = MaSanPham;
            SanPham sp = db.SanPhams.Single(n => n.MaSanPham == iMaSanPham);
            sTenSP = sp.TenSP;
            sHinhAnh = sp.HinhAnh;
            dGiaTien = double.Parse(sp.GiaTien.ToString());
            iSoLuong = 1;
        }
    }
}
```

- Thêm controller GioHangController.cs



- GioHangController.cs. Xây dựng các phương thức xử lý giỏ hàng theo code sau:

```
using QuanLyBanHang.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
namespace QuanLyBanHang.Controllers
{
    public class GioHangController : Controller
    {
        QuanLyBanHangModel db = new QuanLyBanHangModel();
        //Lấy giỏ hàng
        public List<GioHang> LayGioHang()
        {
            return db.GioHangs.ToList();
        }
        //Thêm giỏ hàng
        public ActionResult ThemGioHang(int iMaSanPham, string strURL)
        {
            GioHang gh = new GioHang();
            gh.iMaSanPham = iMaSanPham;
            gh.strURL = strURL;
            db.GioHangs.Add(gh);
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        //Cập nhật giỏ hàng
        public ActionResult CapNhatGioHang(int iMaSanPham, FormCollection f)
        {
            GioHang gh = db.GioHangs.Find(iMaSanPham);
            if (f["txtTenSP"] != null)
                gh.strTenSP = f["txtTenSP"];
            if (f["txtGia"] != null)
                gh.dGia = Convert.ToDouble(f["txtGia"]);
            if (f["txtSoLuong"] != null)
                gh.iSoLuong = Convert.ToInt32(f["txtSoLuong"]);
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        //Xóa giỏ hàng
        public ActionResult XoaGioHang(int iMaSanPham)
        {
            GioHang gh = db.GioHangs.Find(iMaSanPham);
            db.GioHangs.Remove(gh);
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        //Xây dựng trang giỏ hàng
        public ActionResult GioHang()
        {
            return View(db.GioHangs);
        }
        // Tính tổng số lượng và tổng tiền ...
        private int TongSoLuong()
        {
            return db.GioHangs.Sum(giоhang => гiоhang.iSoLuong);
        }
        private double TongTien()
        {
            return db.GioHangs.Sum(giоhang => гiоhang.dGia * гiоhang.iSoLuong);
        }
        //tạo partial giỏ hàng
        public ActionResult GioHangPartial()
        {
            return PartialView();
        }
        //Xây dựng 1 view cho người dùng chỉnh sửa giỏ hàng
        public ActionResult SuaGioHang()
        {
            return View(db.GioHangs);
        }
    }
}
```

- GioHangController.cs:

```
using QuanLyBanHang.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace QuanLyBanHang.Controllers
{
    public class GioHangController : Controller
    {
        QuanLyBanHangModel db = new QuanLyBanHangModel();

        //Lấy giỏ hàng
        public List<GioHang> LayGioHang()
        {
            List<GioHang> lstGioHang = Session["GioHang"] as List<GioHang>;
            if (lstGioHang == null)
            {
                lstGioHang = new List<GioHang>();
                Session["GioHang"] = lstGioHang;
            }
            return lstGioHang;
        }
```

```

//Thêm giỏ hàng
public ActionResult ThemGioHang(int iMaSanPham, string strURL)
{
    return RedirectToAction("GioHang");
}
//Cập nhật giỏ hàng
public ActionResult CapNhatGioHang(int iMaSanPham, FormCollection f)
{
    return RedirectToAction("GioHang");
}
//Xóa giỏ hàng
public ActionResult XoaGioHang(int iMaSanPham)
{
    return RedirectToAction("GioHang");
}
//Xây dựng trang giỏ hàng
public ActionResult GioHang()
{
    if (Session["GioHang"] == null)
    {
        return RedirectToAction("Index", "Home");
    }
    List<GioHang> lstGioHang = LayGioHang();
    return View(lstGioHang);
}
//Tính tổng số lượng và tổng tiền
//Tính tổng số lượng
private int TongSoLuong()
{
    return 0;
}
//Tính tổng thành tiền
private double TongTien()
{
    return 0;
}

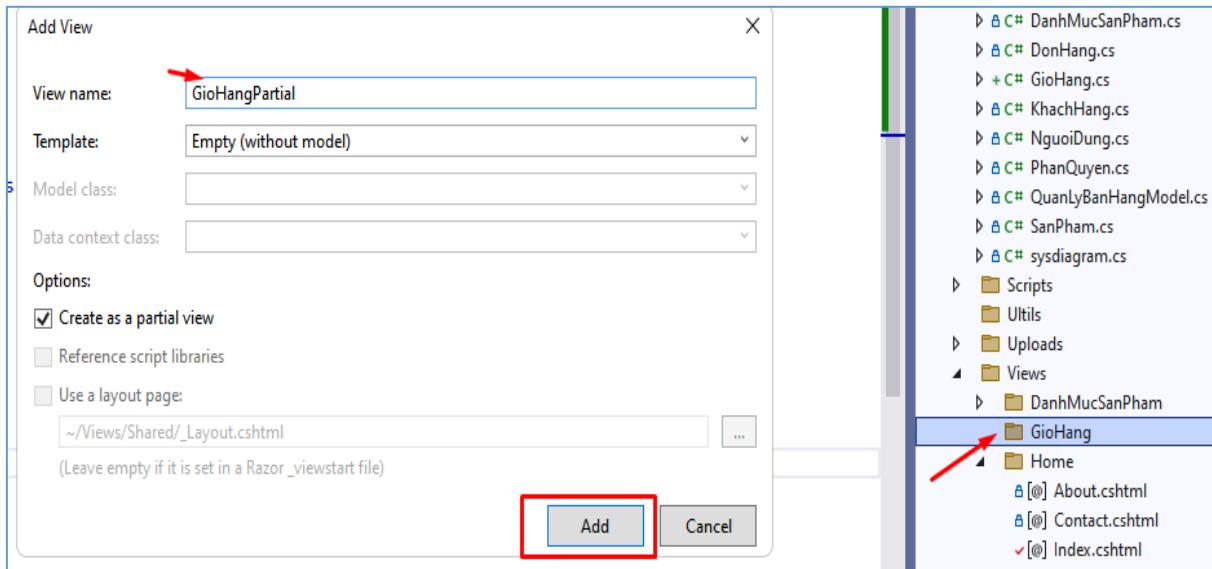
//tạo partial giỏ hàng
public ActionResult GioHangPartial()
{
    if (TongSoLuong() == 0)
    {
        return PartialView();
    }
    ViewBag.TongSoLuong = TongSoLuong();
    ViewBag.TongTien = TongTien();
    return PartialView();
}
//Xây dựng 1 view cho người dùng chỉnh sửa giỏ hàng
public ActionResult SuaGioHang()
{
    return View();
}

}

```

6.2. Xây dựng trang giỏ hàng partial, giỏ hàng

- Thêm mới PartialView: GioHangPartial:



- GioHangPartial.cshtml

```
<style type="text/css">
    .giohang {
        color: white;
    }
</style>
<a href="@Url.Action("GioHang", "GioHang")" class="giohang">
    Giỏ hàng <i class="fa fa-shopping-cart">
    </i> (@ViewBag.TongSoLuong)
</a>
```

- Tại _layout.cshtml: Thêm mới đoạn code sau tại thẻ <head> để sử dụng font-awesome:

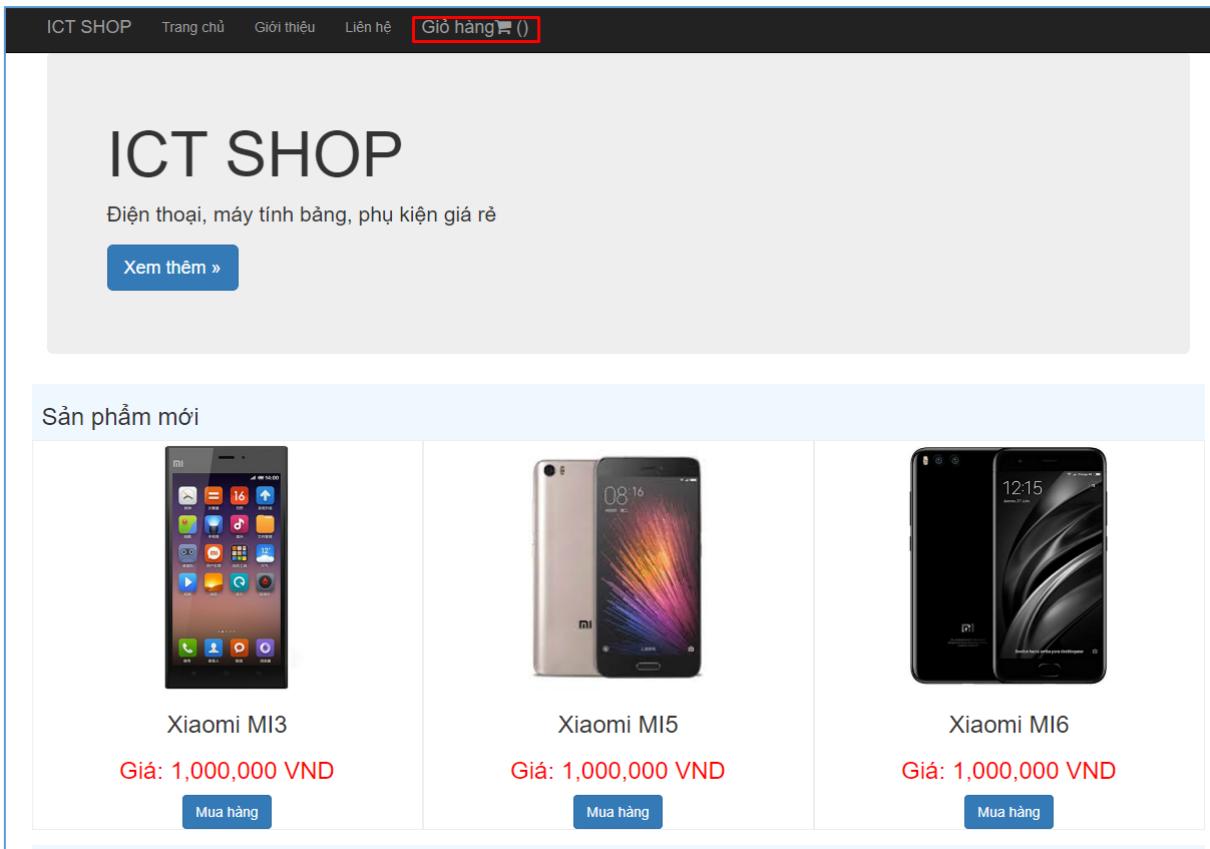
```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

```
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - My ASP.NET Application</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
</head>
```

- Tại _layout.cshtml: Thêm hiển thị giỏ hàng tại header:

```
<ul class="nav navbar-nav">
    <li>@Html.ActionLink("Trang chủ", "Index", "Home")</li>
    <li>@Html.ActionLink("Giới thiệu", "About", "Home")</li>
    <li>@Html.ActionLink("Liên hệ", "Contact", "Home")</li>
    <li>
        @Html.Action("GioHangPartial", "GioHang", new { @class = "giohang" })
    </li>
</ul>
```

- Tiến hành build lại solution và chạy website:



- Code phương thức ThemGioHang

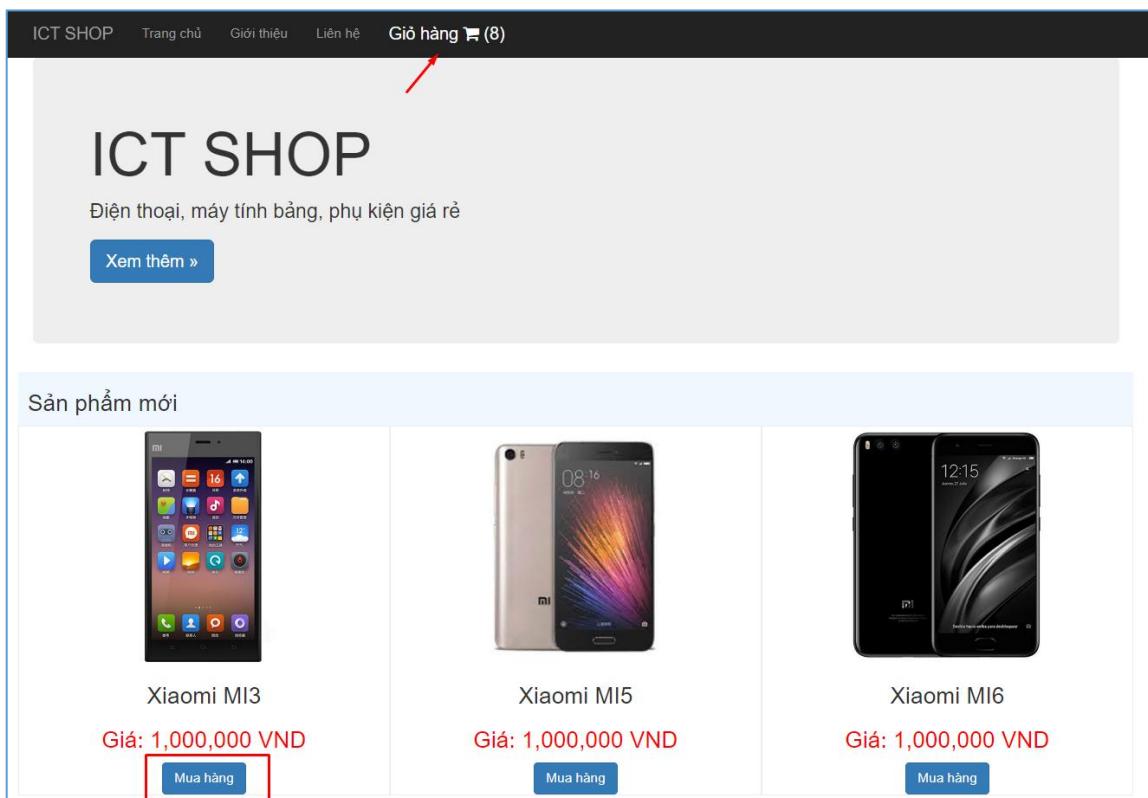
```
//Thêm giỏ hàng
public ActionResult ThemGioHang(int iMaSanPham, string strURL)
{
    SanPham sp = db.SanPhams.SingleOrDefault(n => n.MaSanPham == iMaSanPham);
    if (sp == null)
    {
        Response.StatusCode = 404;
        return null;
    }
    //Lấy ra session giỏ hàng
    List<GioHang> lstGioHang = LayGioHang();
    //Kiểm tra sản phẩm trong session[giohang] đã tồn tại?
    GioHang sanpham = lstGioHang.Find(n => n.iMaSanPham == iMaSanPham);
    if (sanpham == null)
    {
        sanpham = new GioHang(iMaSanPham);
        //Add sản phẩm mới thêm vào list
        lstGioHang.Add(sanpham);
        return Redirect(strURL);
    }
    else
    {
        sanpham.iSoLuong++;
        return Redirect(strURL);
    }
}
```

```

//Tính tổng số lượng
private int TongSoLuong()
{
    int iTongSoLuong = 0;
    List<GioHang> lstGioHang = Session["GioHang"] as List<GioHang>;
    if (lstGioHang != null)
    {
        iTongSoLuong = lstGioHang.Sum(n => n.iSoLuong);
    }
    return iTongSoLuong;
}

```

- Tiến hành build lại solution và chạy website:
- Bấm mua hàng: Giỏ hàng hiển thị số lượng sản phẩm của giỏ hàng:



- Xây dựng View cho trang giỏ hàng:

//Xây dựng trang giỏ hàng

```

public ActionResult GioHang()
{
    if (Session["GioHang"] == null)
    {
        return RedirectToAction("Index");
    }
    List<GioHang> lstGioHang = Layout.GetGioHang();
    return View(lstGioHang);
}

//Tính tổng số lượng và tổng tiền
//Tính tổng số lượng
private int TongSoLuong()
{
    int iTongSoLuong = 0;
    List<GioHang> lstGioHang = Session["GioHang"] as List<GioHang>;
    if (lstGioHang != null)

```

Add View

View name: GioHang

Template: Empty (without model)

Model class:

Data context class:

Options:

Create as a partial view

Reference script libraries

Use a layout page:
~/Views/Shared/_Layout.cshtml
(Leave empty if it is set in a Razor _viewstart file)

Add Cancel

```

@using QuanLyBanHang.Models;
@model List<GioHang>
 @{
    ViewBag.Title = "GIO HÀNG <i class='fa fa-shopping-cart'></i></h2>";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2 class=""> GIỎ HÀNG <i class="fa fa-shopping-cart"></i></h2>
<br />
<table class="table-hover table-bordered divGioHang">
    <tr style="">
        <th> Mã SP</th>
        <th> Tên SP</th>
        <th> Hình ảnh</th>
        <th> Đơn giá</th>
        <th> Số lượng</th>
        <th> Thành tiền</th>
    </tr>
    @foreach (var item in Model)
    {
        <tr class="font-italic">
            <td> @item.iMaSanPham</td>
            <td> @item.sTenSP</td>
            <td> </td>
            <td> @String.Format("{0:0,0}", item.dGiaTien) VNĐ</td>
            <td> @item.iSoLuong</td>
            <td> @String.Format("{0:0,0}", item.ThanhTien) VNĐ</td>
        </tr>
    }
    <tr>
        <td colspan="6" style="text-align:right">
            @Html.ActionLink("Chỉnh sửa giỏ hàng", "SuaGioHang")
        </td>
        <td colspan="6" style="text-align:right;color:blue">
            @using (Html.BeginForm("ThanhToanDonHang", "GioHang", new { @class = "" }))
            {
                <input type="submit" value="Đặt hàng" class="btn-danger" />
            }
        </td>
    </tr>
</table>

/*CSS giỏ hàng*/
<style type="text/css">
    .divGioHang {
        border: 2px solid #EEEEEE;
        margin-left: 10%;
        font-size: large
    }

    .divGioHang td {
        padding: 10px;
        border: 2px solid #EEEEEE;
        text-align: center
    }

    .divGioHang th {
        text-align: center
    }
</style>

```

- Build solution – Chạy website:

- Tiến hành bấm mua hàng để thêm mới sản phẩm vào giỏ hàng. Sau đây click vào giỏ hàng. Hiển thị chi tiết giỏ hàng:

Mã SP	Tên SP	Hình ảnh	Đơn giá	Số lượng	Thành tiền
14	Xiaomi MI5		1,000,000 VNĐ	4	4,000,000 VNĐ
15	Xiaomi MI6		1,000,000 VNĐ	1	1,000,000 VNĐ
17	Iphone 4		2,000,000 VNĐ	1	2,000,000 VNĐ
13	Xiaomi MI3		1,000,000 VNĐ	2	2,000,000 VNĐ

© 2023 - My ASP.NET Application

6.3. Xây dựng chức năng chỉnh sửa giỏ hàng và đặt hàng

- Sinh viên thực hiện xây dựng chức năng Chính sửa giỏ hàng và Đặt hàng.

Phần nâng cao và hướng tìm hiểu thêm cho sinh viên

- Phần website quản trị:

+ Tìm hiểu thay thế _layout.cshtml mặc định bằng một template Admin HTML, CSS miễn phí trên internet để có giao diện đẹp, tiết kiệm thời gian code.

+ Tìm hiểu thêm về bảo mật cho ứng dụng Asp.net MVC

+ Xây dựng trang quản lý đơn hàng, xây dựng dashboard hiển thị thống kê đơn hàng, tiền hàng, số lượng khách hàng...

- Phần website người dùng:

+ Tìm hiểu thay thế _layout.cshtml mặc định bằng một template HTML, CSS miễn phí trên internet để có giao diện đẹp, tiết kiệm thời gian code.

+ Xây dựng chức năng tìm kiếm, bộ lọc sản phẩm

+ Hoàn thành chức năng giỏ hàng, đơn hàng, trang thanh toán cho người dùng.

+ Xây dựng trang đăng ký, đăng nhập, đổi mật khẩu cho người dùng...

---HẾT---