


Trường ĐH CNTP TP.HCM Khoa: CNTT Bộ môn: Khoa học máy tính LẬP TRÌNH WEB	BÀI 1 MÔ HÌNH MVC (MVC, Routeconfig, LayoutTemplate)	
---	---	---

A. MỤC TIÊU:

- Trình bày được cấu trúc của mô hình MVC
- ASP.Net MVC tạo được ApplicationWeb theo mô hình này.
- Thiết kế View – Control hiển thị được các thẻ html (...).
- LayoutTemplate, RenderBody(), ActionLink

B. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

1.1. Kiến thức cần nhớ

- Mô hình MVC: là một mô hình phát triển ứng dụng có kiến trúc tốt, có khả năng kiểm thử tốt và dễ dàng vận hành, bảo trì.
 - Model : chứa dữ liệu, luân chuyển dữ liệu, Model là các lớp chức năng thể hiện cho việc xử lý dữ liệu và các vấn đề liên quan đến nghiệp vụ và các quy tắc ràng buộc dữ liệu. Đồng thời, nó còn có nhiệm vụ lưu trữ thông tin, trạng thái của các đối tượng vào CSDL.
 - View: hiển thị dữ liệu, tương tác người dùng, View là thành phần chịu trách nhiệm hiển thị các giao diện dưới dạng html theo cơ chế phát sinh động của ứng dụng. Các thông tin cần hiển thị sẽ được lấy từ Models thông qua các Controllers.
 - Controller: Controller nhận yêu cầu, xử lý yêu cầu, nếu cần truy xuất dữ liệu thì controller sẽ gửi yêu cầu đến tầng model để truy xuất dữ liệu trả về cho View
- So sánh ASP.Net MVC và ASP.Net WebForm
- Tạo project mới, giới thiệu cấu trúc ASP.Net MVC Project (App_Data, App_Start, Controllers, Models, Views,...)

Controllers	Chứa các file xử lý đóng vai trò điều khiển
Models	Chứa các class
Views	Chứa giao diện (Code html, css)
App_Data	Lưu trữ các tập tin dữ liệu, tập tin XML hoặc Database
App_Start	Chứa các tập tin liên quan đến việc cấu hình cho các tính năng : routers , filters,...

Content	Chứa tập tin CSS, hình ảnh liên quan đến giao diện
Scripts	Chứa các thư viện JavaScript
Packages.config	Chứa tất cả các thư viện sd trong ứng dụng
Web.config	Cấu hình cho website
Global.asax	...

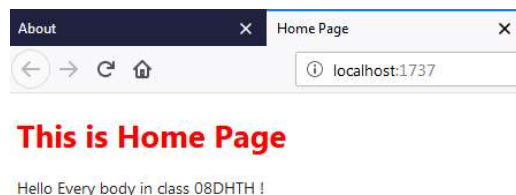
- Layout Template (Master page) : cách tạo và sử dụng
- Html.ActionLink: Tạo kết nối với một action trong một Controller bất kỳ và thực thi action đó

1.2. Giới thiệu bài tập mẫu

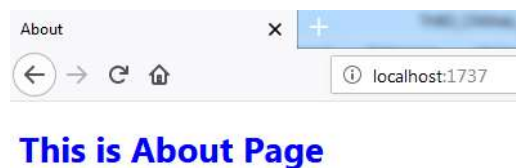
Bài 1. Tạo project website sử dụng mô hình MVC đầu tiên có tên Bai1_HelloWord

Yêu cầu:

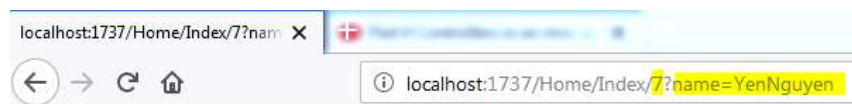
- Tạo 01 controller Home (HomeController.cs) gồm 02 phương thức cho phép hiển thị 02 view tương ứng là View Home.cshtml và About.cshtml
- Thiết kế View Home như sau



- Thiết kế View About như sau



- Điều hướng trang bằng RouteConfig.cs sao cho view hiển thị đầu tiên là view home trong controller Home
- Sau đó thực hiện điều hướng trang sang view About trong Controller Home.
- Tạo một phương thức Index trong HomeController thực hiện trả về chuỗi lấy từ hai tham số id và name như sau:



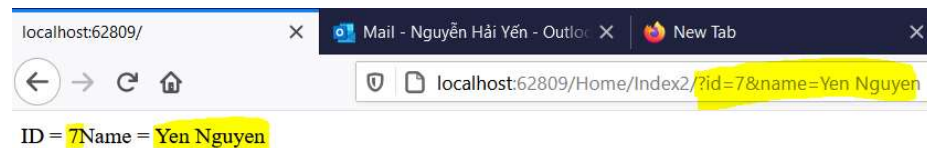
ID = 7 Name = YenNguyen

Yêu cầu truyền tham số theo 2 cách

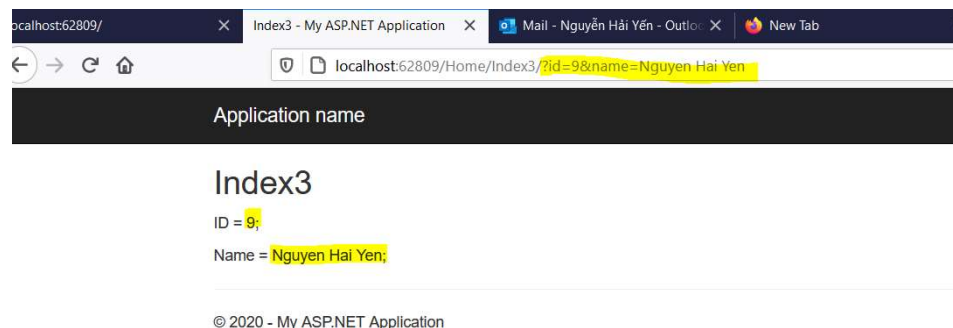
(1) Truyền trong file RouteConfig.cs

(2) Truyền trực tiếp trên URL

- Tạo một phương thức **Index2()** trong HomeController thực hiện trả về chuỗi lấy từ hai tham số id và name như sau:

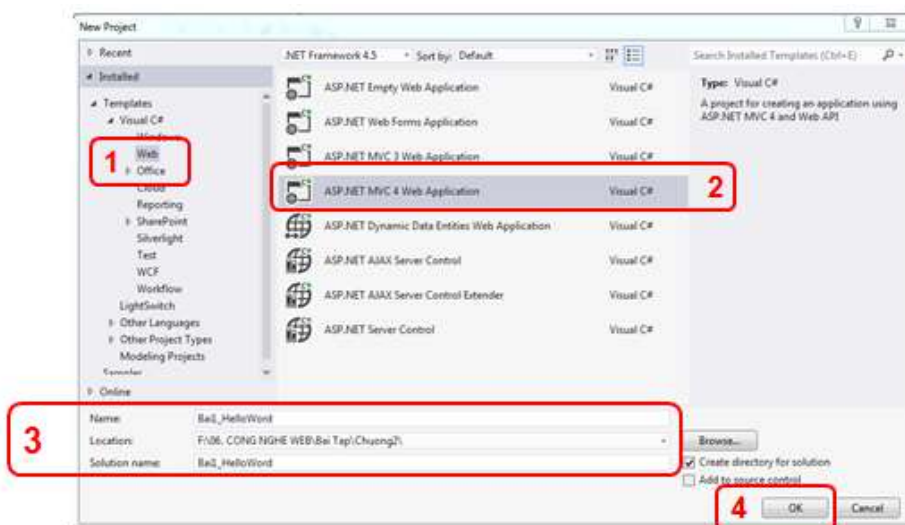


- Tạo một phương thức **Index3()** trong HomeController thực hiện trả về View Index3.cshtml; thực hiện truyền tham 2 số id và name lấy từ URL, sau đó xuất giá trị 2 tham số này trên view Index3.cshtml như sau (Dùng ViewData và Dùng ViewBag)

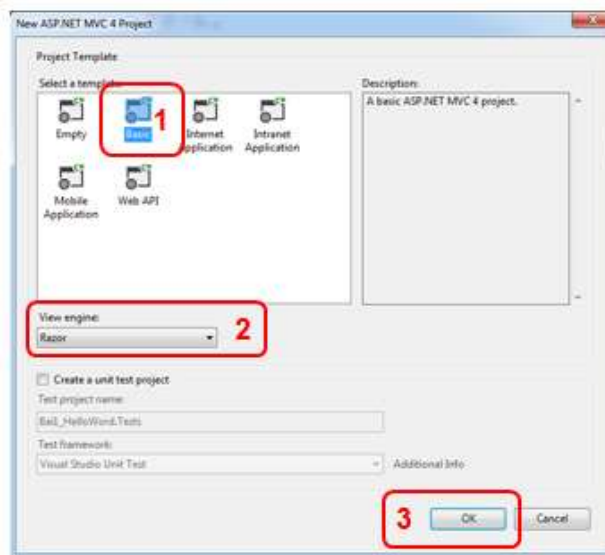


Hướng dẫn:

B1: Tạo Project mới File → New → Project..



B2: Chọn Template



B3: Tạo HomeController.cs

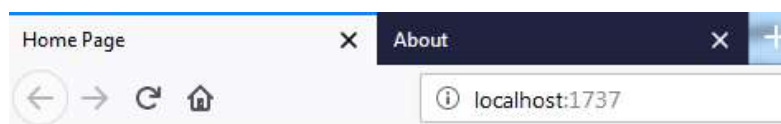
B4: Tạo View Home.cshtml; Tạo View About.cshtml

B5: Điều hướng trang bằng RouteCongig.cs

```
namespace Bai1_HelloWord
{
    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 = "About", id = UrlParameter.Optional }
            );
        }
    }
}
```

Kết quả:



This is Home Page

Hello Every body in class 08DHTH !

Bài 2: Thực hiện trao đổi dữ liệu và tương tác 3 tầng M – V - C

Trong Folder Models tạo một lớp KhoaModel

```

3 references
public class KhoaModels
{
    1 reference
    public String TenKhoa { get; set; }
    1 reference
    public int NamThanhLap { get; set; }
    2 references
    public String Message { get; set; }

    1 reference
    public KhoaModels()
    {
        TenKhoa = "Khoa Công nghệ Thông tin";
        NamThanhLap = 2003;
        Message = "Chào mừng các bạn đến với FIT-HUFI";
    }
}

```

Tạo KhoaController

```

0 references
public class KhoaController : Controller
{
    // GET: Khoa
    0 references
    public ActionResult Index()
    {
        Models.KhoaModels khoaCNTT = new Models.KhoaModels();
        khoaCNTT.Message = "FIT-HUFI: HỌC TẬP - NĂNG ĐỘNG - SÁNG TẠO";
        return View(khoaCNTT);
    }
}

```

Tạo View Index.cshtml như sau:

```

.cs      Index.cshtml  TestController.cs  HomeController.cs
@{
    Layout = null;
    ViewBag.Title = "Index";
}

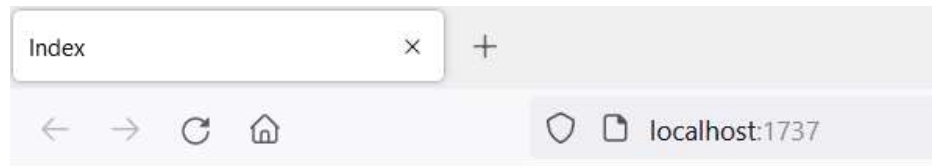
@model Bai1_HelloWord.Models.KhoaModels
<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
</head>
<body>
    <div>
        <h2>@Model.Message</h2>
        Thông tin khoa
        Tên Khoa: @Model.TenKhoa <br />
        Năm thành lập: @Model.NamThanhLap <br />
    </div>
</body>
</html>

```

Để truyền dữ liệu từ tầng Models sang tầng Controller

Gọi thực thi phương thức Index trong KhoaController và xem kết quả



FIT-HUFI: HỌC TẬP - NĂNG ĐỘNG - SÁNG TẠO

Thông tin khoa:
Tên Khoa: Khoa Công Nghệ Thông Tin
Năm thành lập: 2003

2. Bài tập tại lớp

Bài 3. Thực hiện trao đổi dữ liệu và tương tác 3 tầng M – V - C

Tạo project website sử dụng mô hình MVC, đặt tên là Bai2_DangNhap

Yêu cầu:

- Tạo controller Đăng nhập (DNController.cs) gồm 02 phương thức DangNhap và DangKy trả về 02 view tương ứng là DangNhap.cshtml và DangKy.cshtml
- Thiết kế View Đăng Nhập như sau

THÔNG TIN ĐĂNG NHẬP	
Họ và Tên	<input type="text" value="Nguyễn Hải Yến"/>
Email	<input type="text" value="yennh@cntp.edu.vn"/>
Tên Đăng Nhập	<input type="text"/>
Password	<input type="password"/>
<input checked="" type="checkbox"/> Remember Password	
<input type="button" value="ĐĂNG NHẬP"/>	

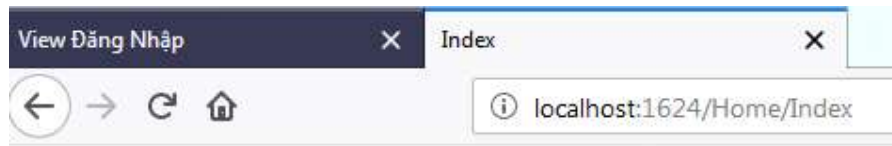
- Thiết kế View Đăng ký như sau

ĐĂNG KÝ TÀI KHOẢN	
Full Name	<input type="text" value="Nguyễn Hải Yến"/>
Email	<input type="text" value="yennh@cntp.edu.vn"/>
Phone Number	<input type="text" value="0908xxxxxx"/>
User Name	<input type="text"/>
Password	<input type="password"/>
Retype Password	<input type="password"/>
<input type="button" value="SUBMIT"/>	

- Thực hiện chạy thử View Đăng ký trong Controller DN bằng cách gõ URL vào thanh Address.

- Trong View Đăng nhập thực hiện kiểm tra
 - + UserName = “yennh”
 - + Password = “123456”

Nếu thỏa chuyển hướng sang View Index của Controller Home. Trong View Index xuất ra câu “Welcome UserName”



Welcome Nguyễn Hải Yến !

Hướng dẫn: Kiểm tra View Đăng Nhập

B1: Tạo 1 Model mới có tên User (Add class)

```
public class User
{
    public string login { get; set; }
    public string UserName { get; set; }
    public string Password { get; set; }
}
```

B2: Tạo HomeController thực hiện kiểm tra nếu tên đăng nhập = null thì chuyển hướng sang view DangNhap của Controller DN

```
public class HomeController : Controller
{
    //
    // GET: /Home/

    public ActionResult Index()
    {
        if (Session["UserName"] == null)
        {
            return RedirectToAction("DangNhap", "DN");
        }
        return View();
    }
}
```

B3: Tạo DNController có:

- 2 phương thức DangNhap và DangKy tương ứng trả về 2 View DangNhap.cshtml và DangKy.cshtml
- 01 Phương thức kiểm tra Đăng nhập với name và password


```
[HttpPost]
public ActionResult DangNhap( string name, string password)
{
    if ("yennh".Equals(name) && "123456".Equals(password))
    {
        Session["user"] = new User() { login = name, UserName = "Nguyễn Hải Yến" };
    }
    return RedirectToAction("Index", "Home");
}
```

B4: Trong Action DangKy thực hiện kiểm tra :

- Chiều dài User Name >= 5 ký tự.
- Chiều dài Password >=6 ký tự
- ReTypePassword = Password

Nếu thỏa chuyển hướng sang View DangNhap của DNController.cs

B5: Trong View index.cshtml của Controller Home

```
<body>
    @{
        var user = Session["user"] as User;
    }
    <h2>Welcome @user.UserName </h2>
</body>
```

B6: Xây dựng phương thức DangXuat trong DNController.cs cho phép đăng xuất ra khỏi tài khoản hiện tại. Sau khi đăng xuất cho trở về View Index của Home

Hướng dẫn :

Session.Clear();

Hoặc Session["user"] = null;

Tạo liên kết từ View Index cho phép thực hiện đăng xuất

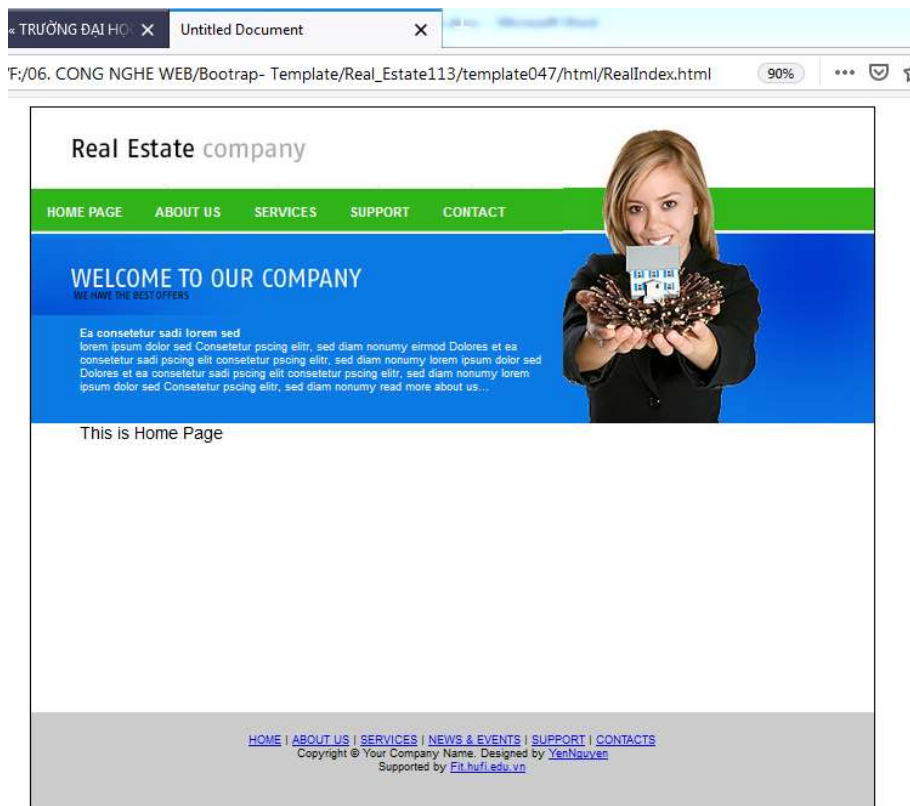


Bài 4:

Tạo Project mới có tên **RealEstate**.

Yêu cầu:

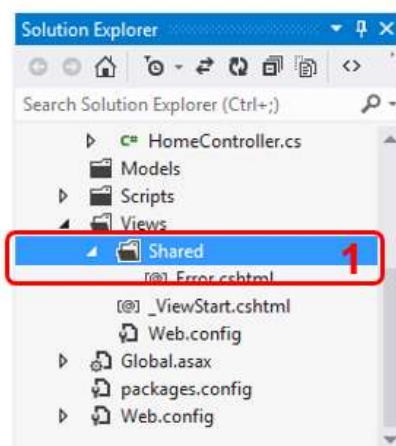
- Tạo layout dùng chung cho các trang như sau:



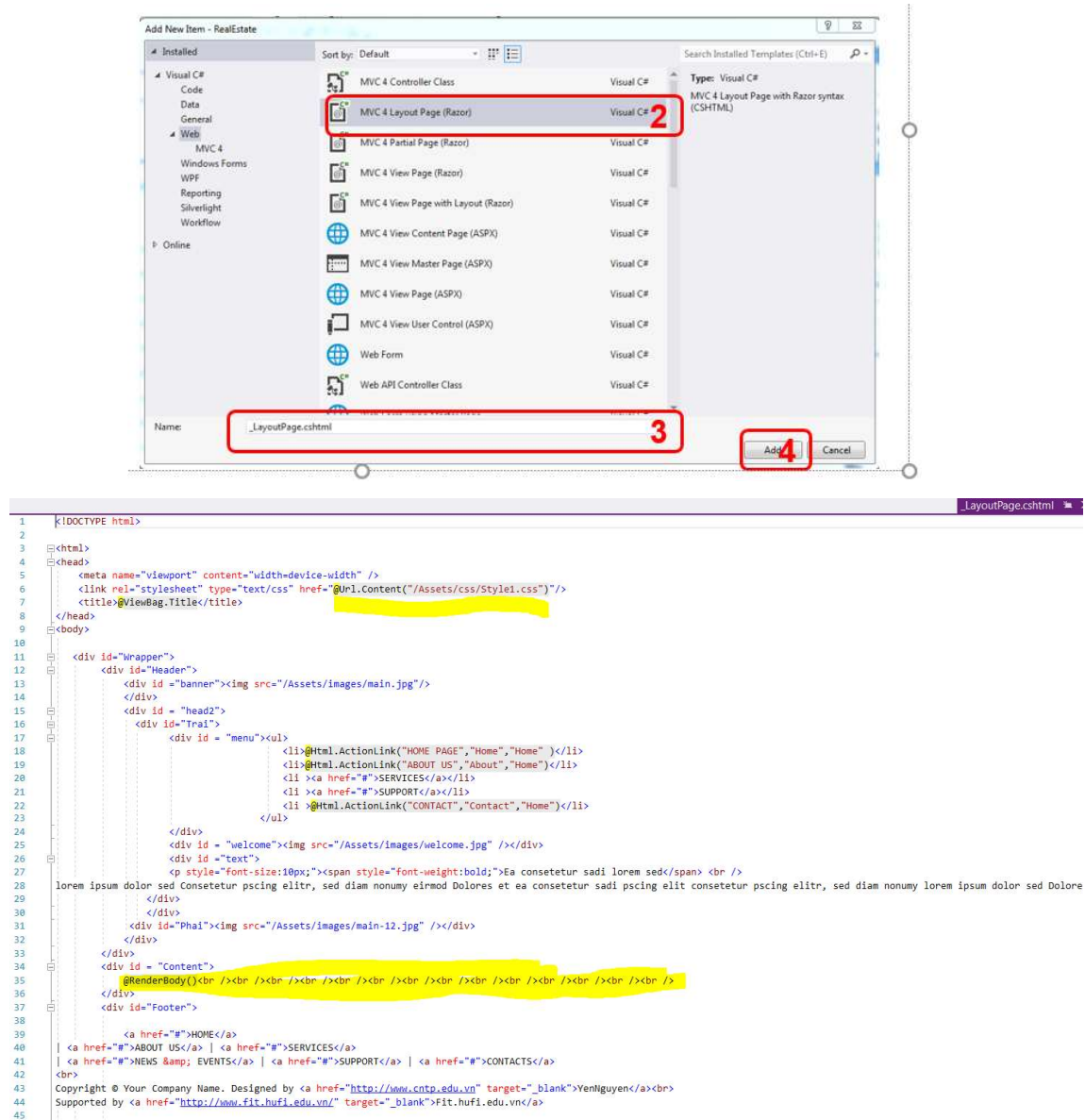
- Tạo Controller home trong đó có các phương thức HomePage, AboutUs, Services, Support, Contact trả về các View tương ứng.
- Tạo các view HomePage.cshtml, ... Contact.cshtml sử dụng layout dùng chung vừa tạo trong yc 1.
- Sử dụng `html.ActionLink` thực hiện chuyển giữa các View khi người dùng chọn menu tương ứng.

Hướng dẫn:

Bước 1: Rclick → Shared → Add → New Item →



Bước 2: Chọn MVC4 LayoutPage

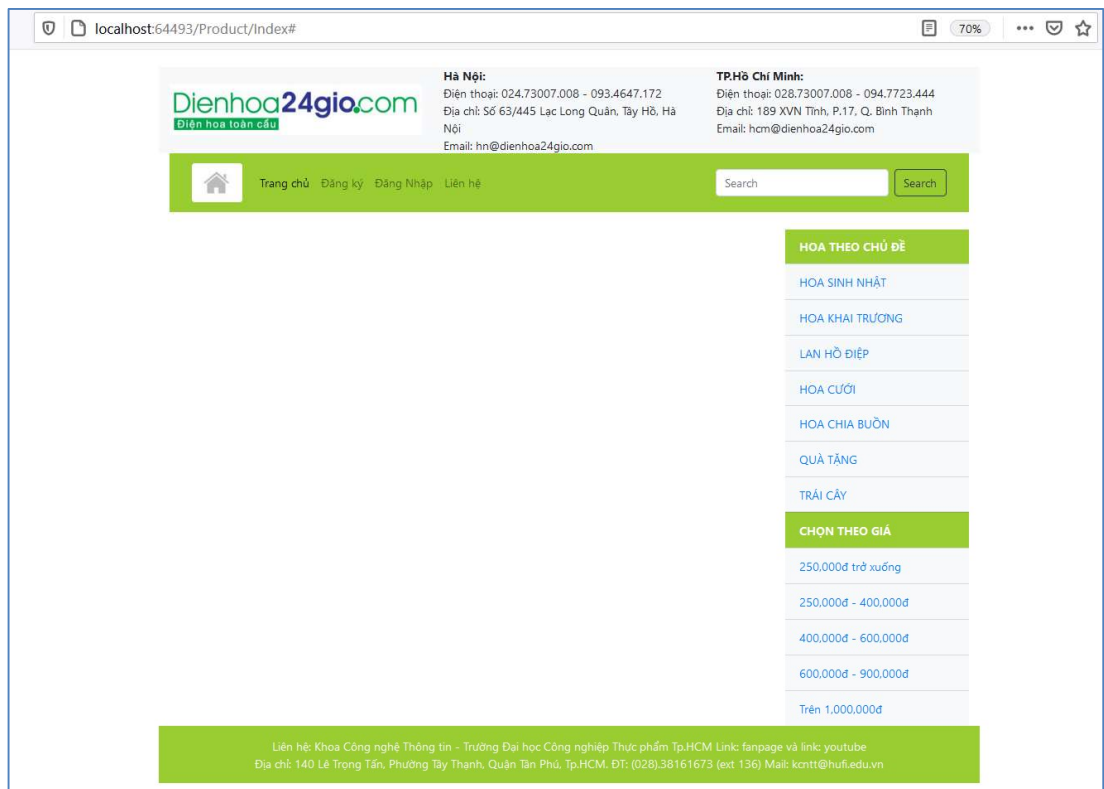


Bài 5: Xây dựng ứng dụng web theo mô hình MVC

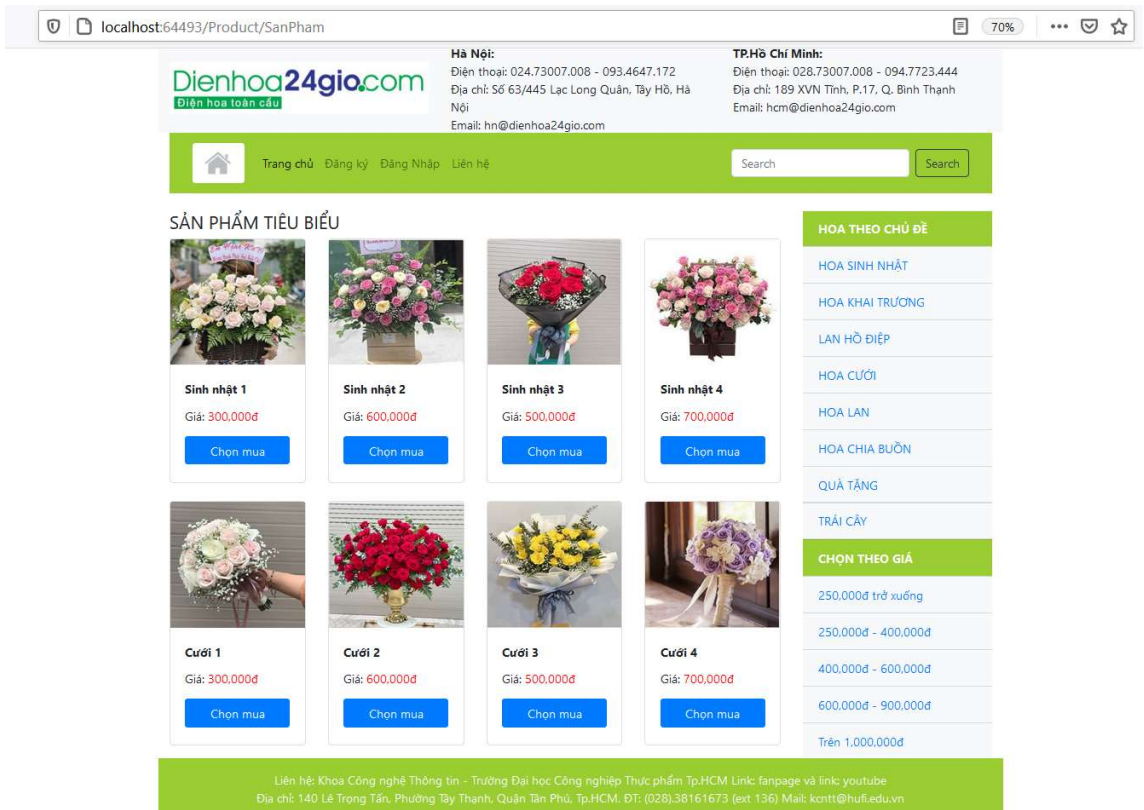
Yêu cầu :

- Tạo giao diện chung khi thiết kế khung trang
- Sử dụng các control trong html, CSS, Bootstrap để thiết kế
- Sử dụng mô hình MVC để xây dựng ứng dụng

(1) Tạo giao diện _LayoutPage.cshtml như sau:



(2) Tạo trang chủ kế thừa từ _LayoutPage như sau:



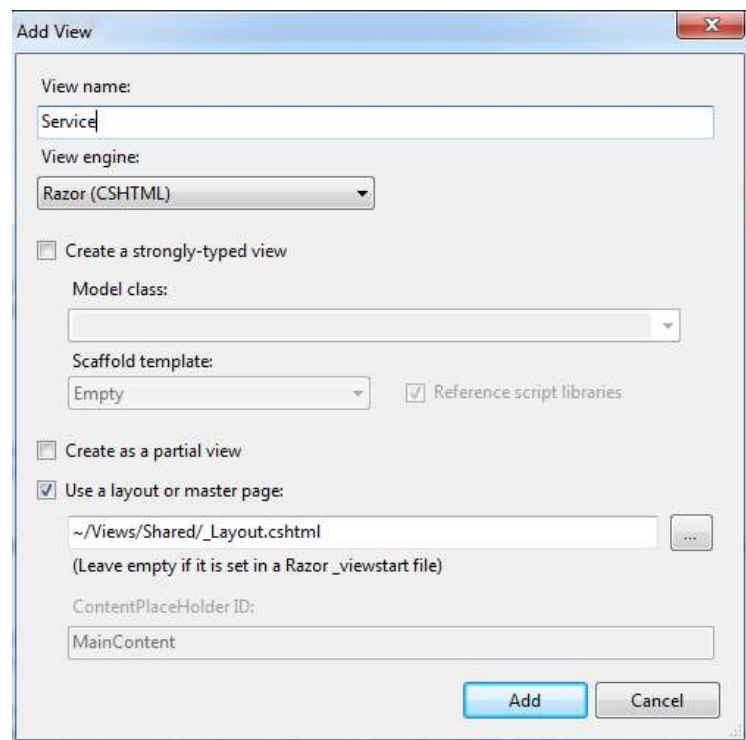
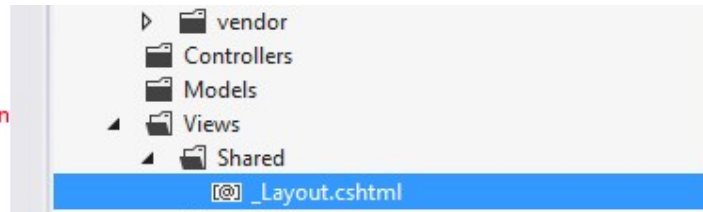
(3) Tạo trang đăng ký

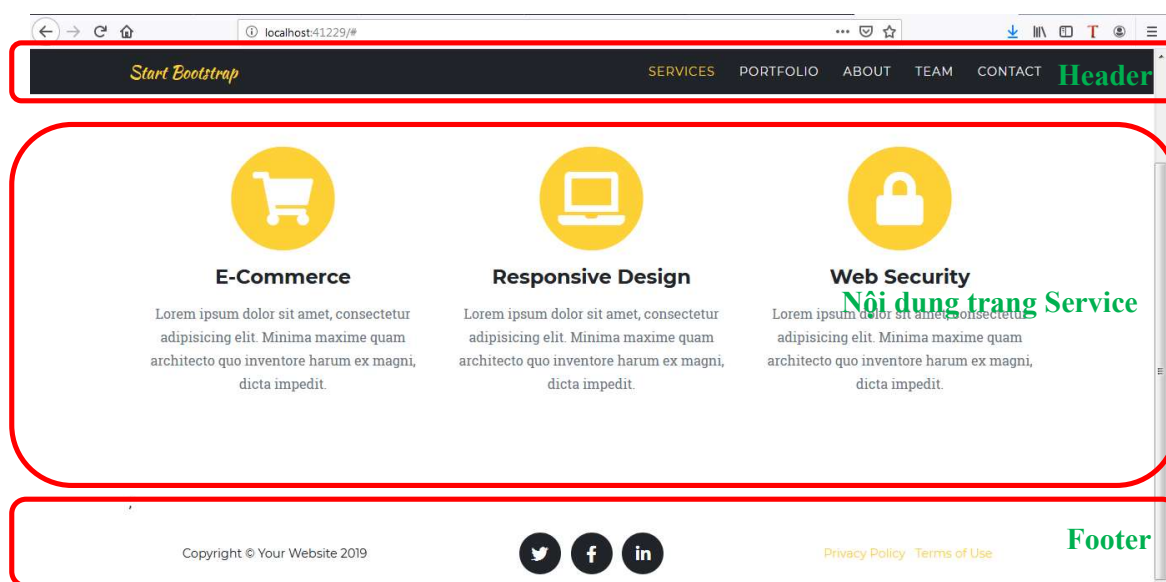
Bài 6: Tạo Project mới có tên Bai5_Layout dùng chung cho các trang (Master page)

Hướng dẫn:

- Vào link sau tải 01 giao diện mẫu: <https://startbootstrap.com/themes/agency/>
- Tạo Folder Shared trong Views để lưu các giao diện mẫu (Master page);

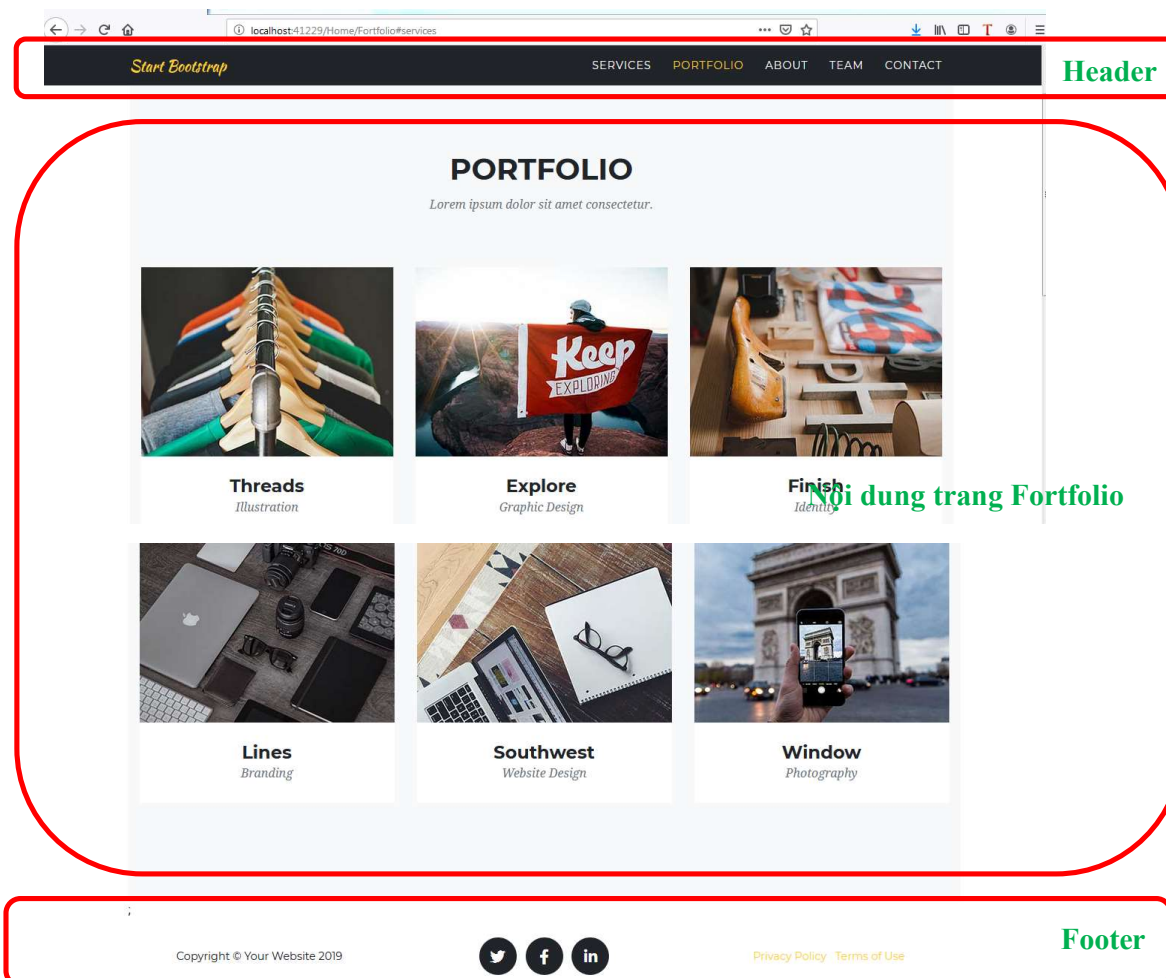
Add mới một giao diện mẫu





3. Bài tập về nhà

Bài 7. Tạo View FORTFOLIO sử dụng Layout Template (_Layout.cshtml đã tạo trong bài 5)



Bài 8:

Tạo View About sử dụng Layout Template (_Layout.cshtml đã tạo trong bài 5)

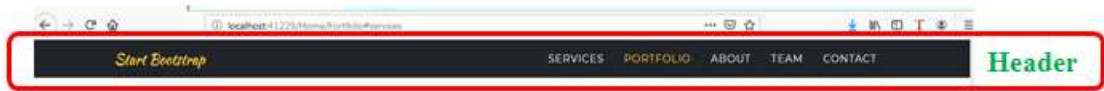
Bài 9:

Tạo View Team sử dụng Layout Template (_Layout.cshtml đã tạo trong bài 5)

Bài 10:

Tạo View Contact sử dụng Layout Template (_Layout.cshtml đã tạo trong bài 5)

Y/C điều chỉnh file _Layout.cshtml phần Navigate



Có sử dụng

```
<li class="nav-item", "nav-link js-scroll-trigger">
  @Html.ActionLink("Services", "Service", "Home")
</li>
```

Để tạo liên kết giữa các View.

Bài 11:

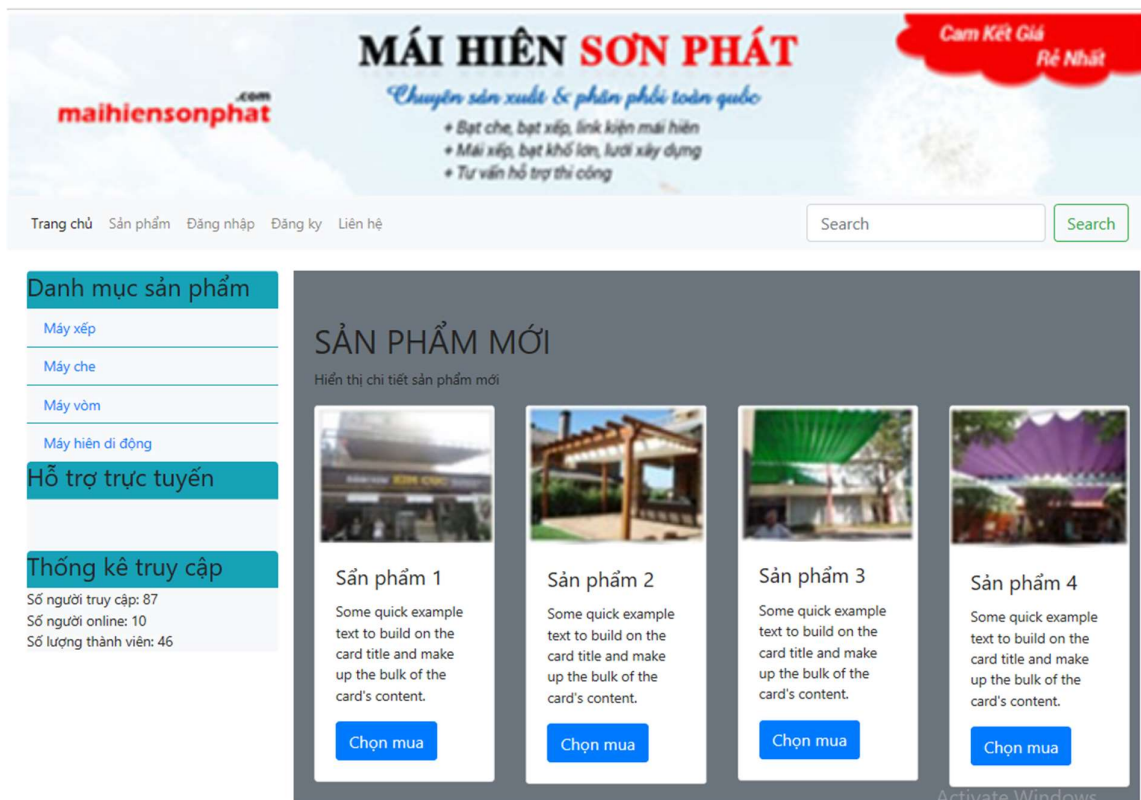
Xây dựng ứng dụng Web theo mô hình MVC, trong ứng dụng có các chức năng sau:

- Trang chủ (hình 1)
- Trang đăng ký (hình 2)
- Trang đăng nhập (hình 3)
- Trang liên hệ (hình 4)

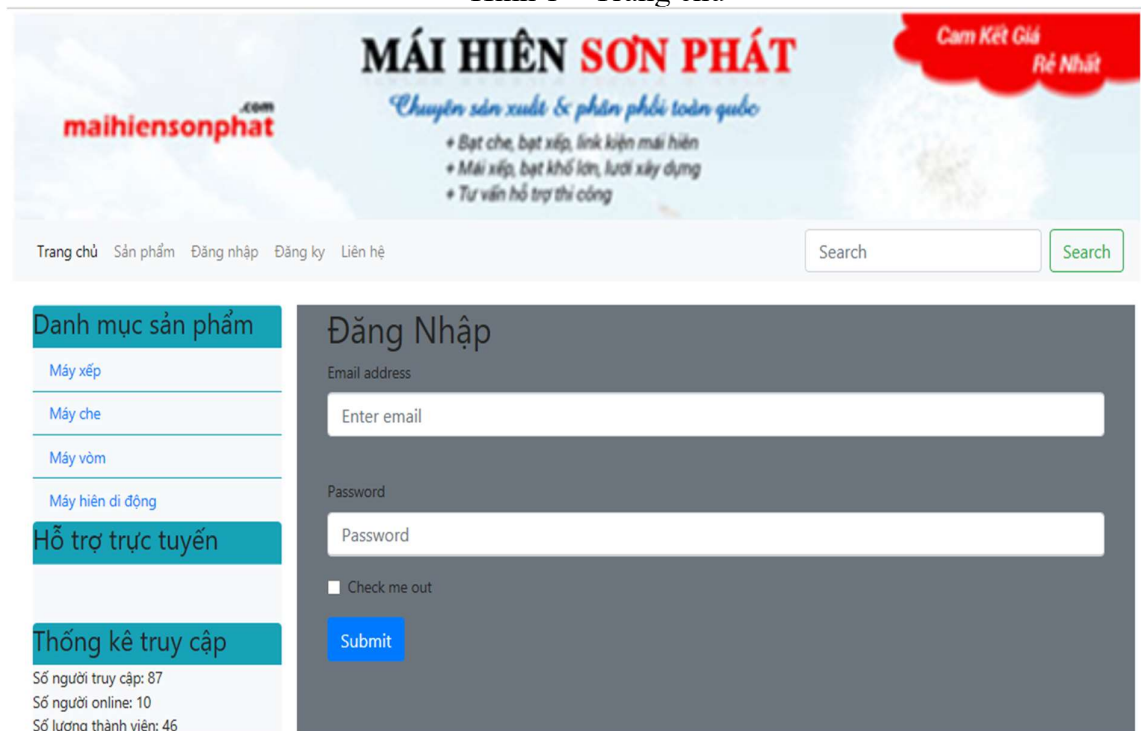
Yêu cầu:

- Sử dụng các control html, css, thư viện bootstrap để thiết kế
- Sử dụng mô hình MVC để xây dựng ứng dụng

Sử dụng giao diện chung (layout) khi thiết kế phần khung trang



Hình 1 – Trang chủ



Hình 2 – Trang đăng nhập

