

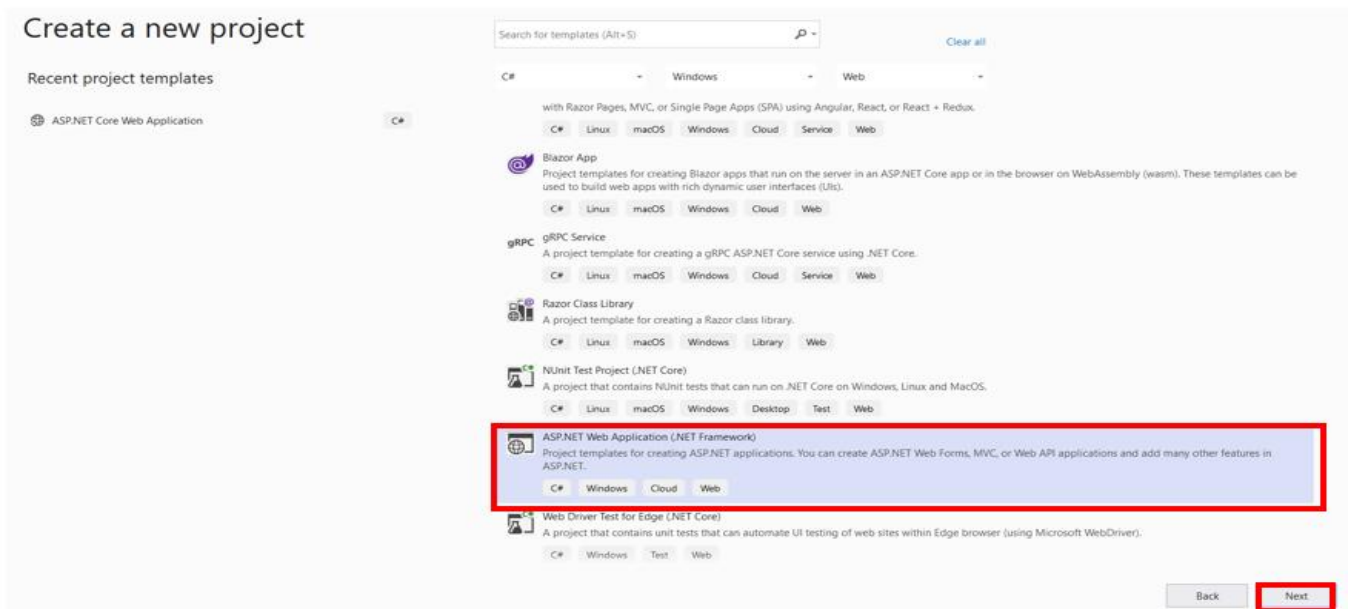
Bài tập 1

Lab 01 – Tạo một project ASP.NET theo mô hình MVC

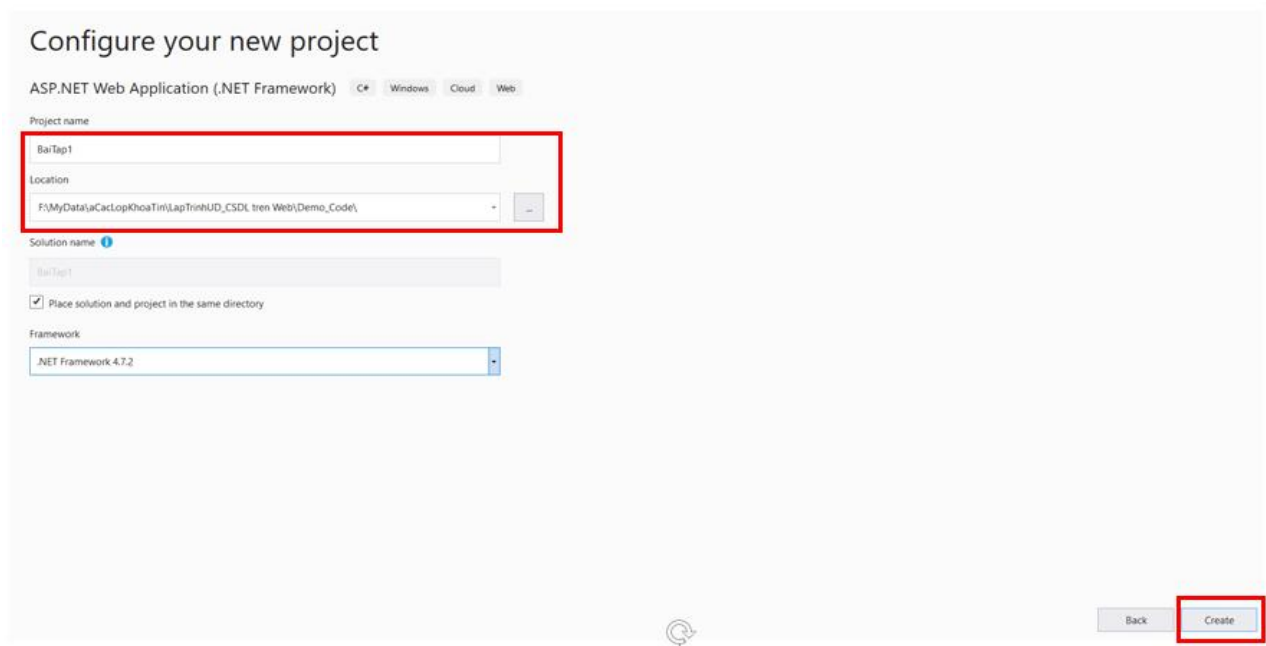
1. Mở Visual Studio. Chọn **Create a new project**



2. Chọn **ASP.NET Web Application (.NET framework)** sau đó kích nút **Next**



3. Đặt tên project là **BaiTap1** và kích vào nút dấu ba chấm ... để chọn vị trí đặt project.



4. Chọn mẫu **Empty**, kích chọn **MVC** phần Add folders & core references và kích nút **Create**

Create a new ASP.NET Web Application

Empty
An empty project template for creating ASP.NET applications. This template does not have any content in it.

Web Forms
A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

MVC
A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Web API
A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

Single Page Application
A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

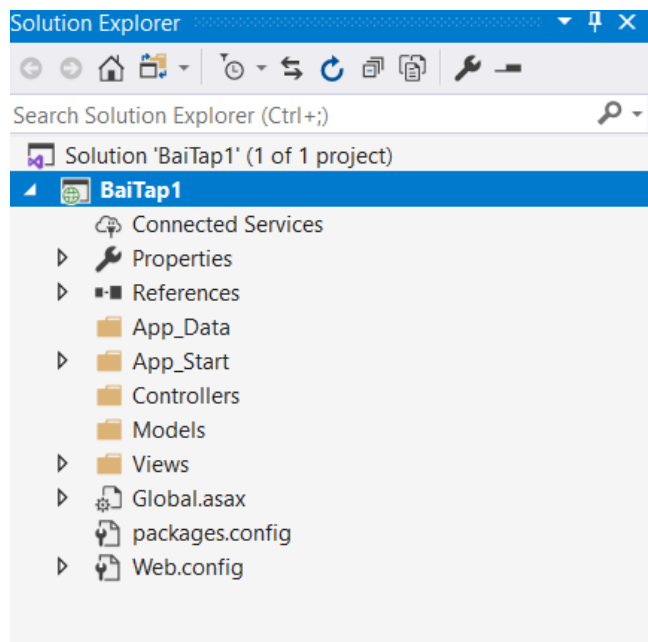
Authentication
No Authentication
[Change](#)

Add folders & core references
☐ Web Forms
☒ **MVC**
☐ Web API

Advanced
☒ Configure for HTTPS
☐ Docker support
(Requires [Docker Desktop](#))
☐ Also create a project for unit tests
[BaiTap1.Tests](#)

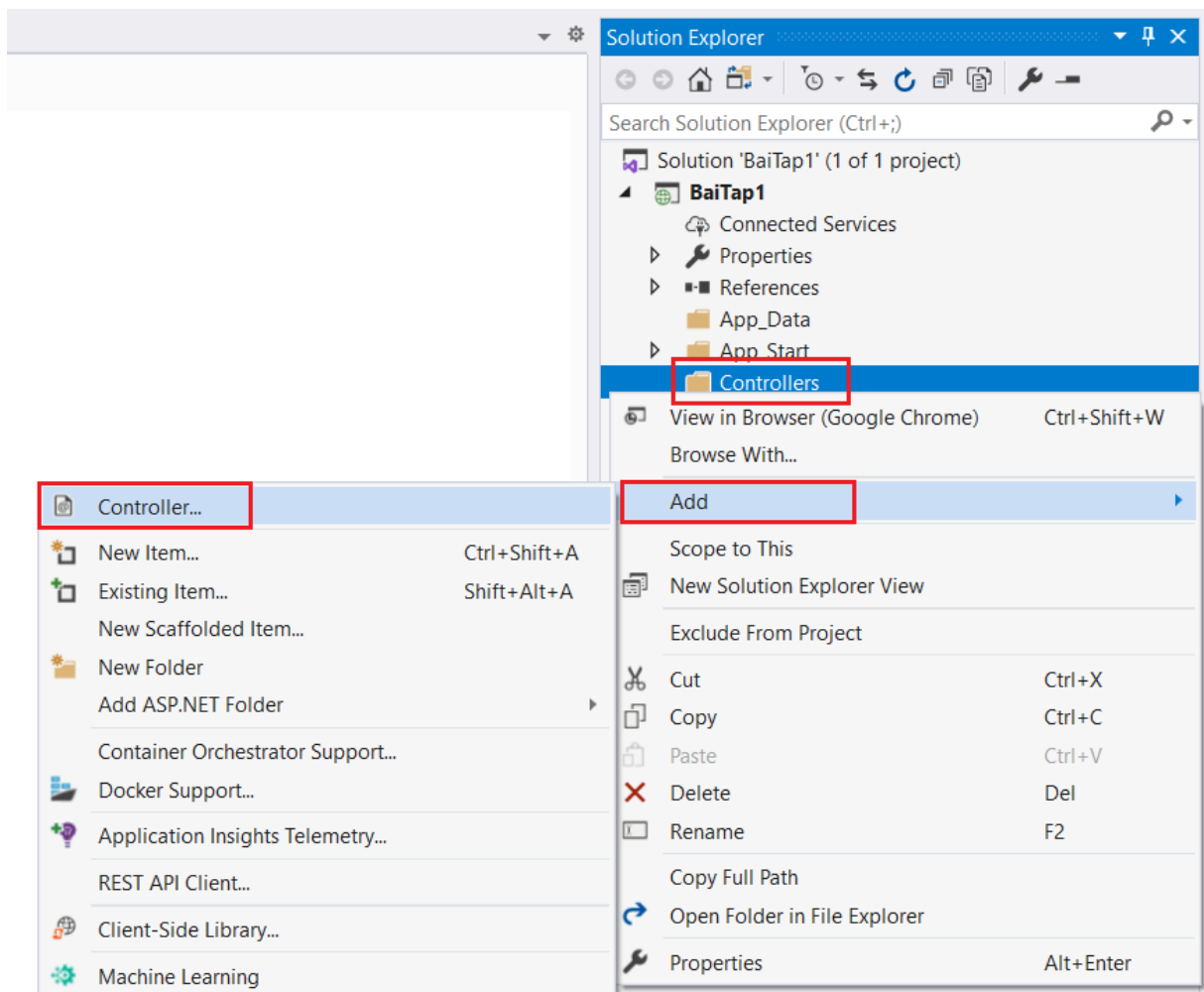
[Back](#) [Create](#)

Các folder trong Project

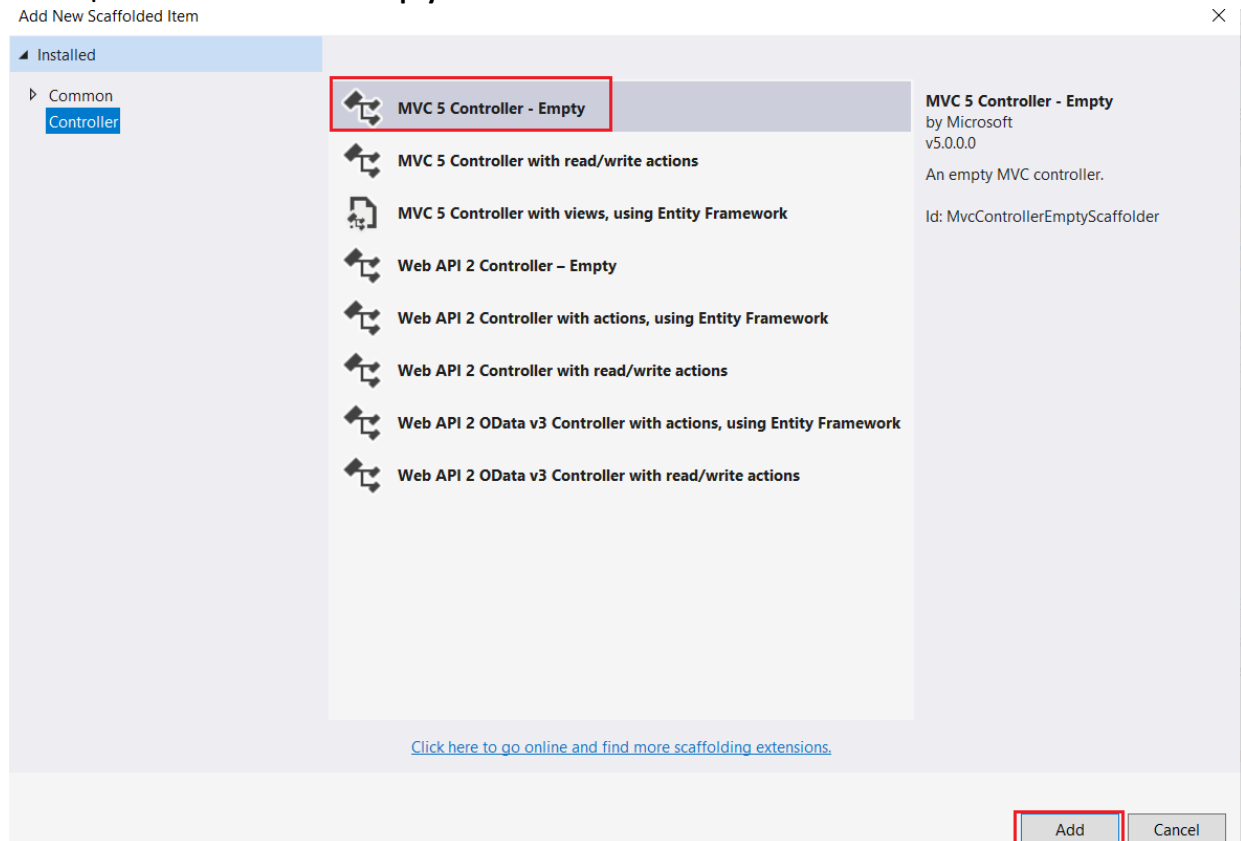


5. Tạo một controller

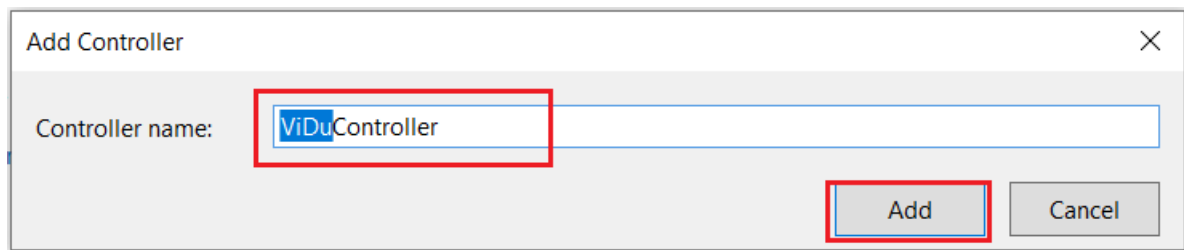
- Kích chuột phải vào folder Controllers, chọn Add, chọn Controller...



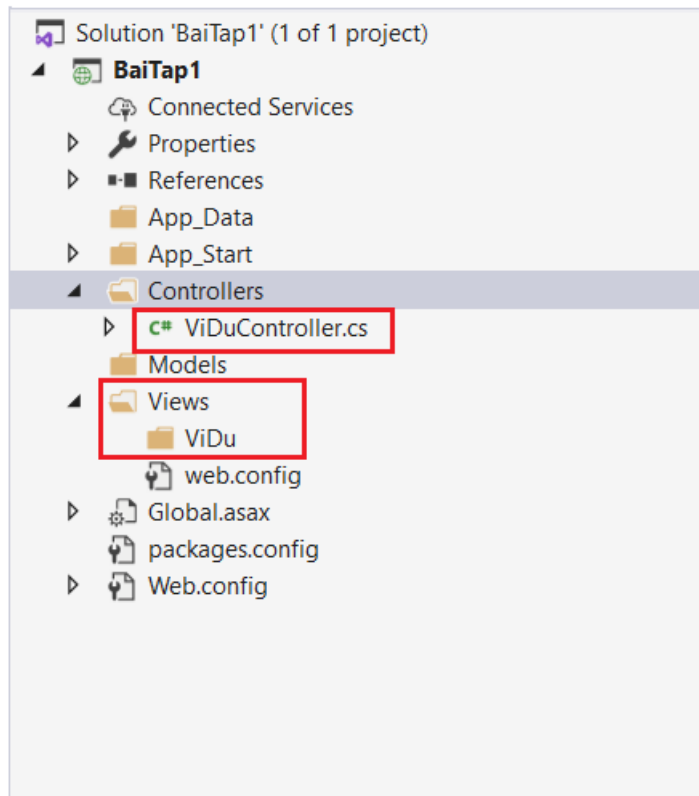
- Chọn **MVC 5 Controller – Empty** và kích nút **Add**



- Đặt tên cho controller là **ViDuController** (lưu ý tên của tất cả các controller đều phải có đuôi là **Controller**) rồi kích vào nút **Add**

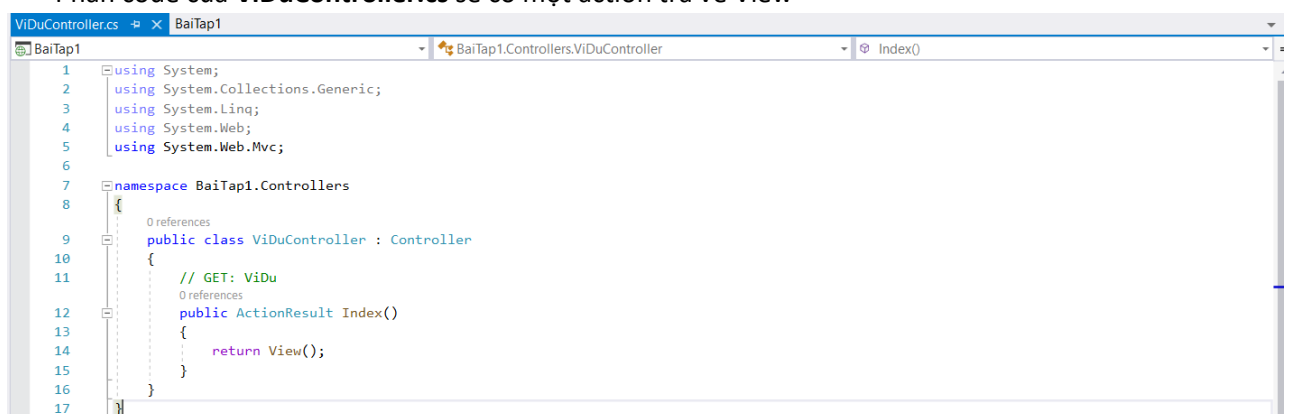


- Khi một controller được tạo thì trong Views sẽ có một folder có tên trùng với tên của controller đó



*****Mỗi controller sẽ có một folder tương ứng trong View**

- Phần code của **ViDuController.cs** sẽ có một action trả về View



6. Cơ chế tìm đường (routing) trong ASP.NET MVC

- Sửa code của **ViDuController.cs** như sau

```
namespace BaiTap1.Controllers
{
    public class ViDuController : Controller
    {
        // GET: ViDu
        public String Index()
        {
            return "Đây là action <b>mặc định</b> của tôi...";
        }
    }
}
```

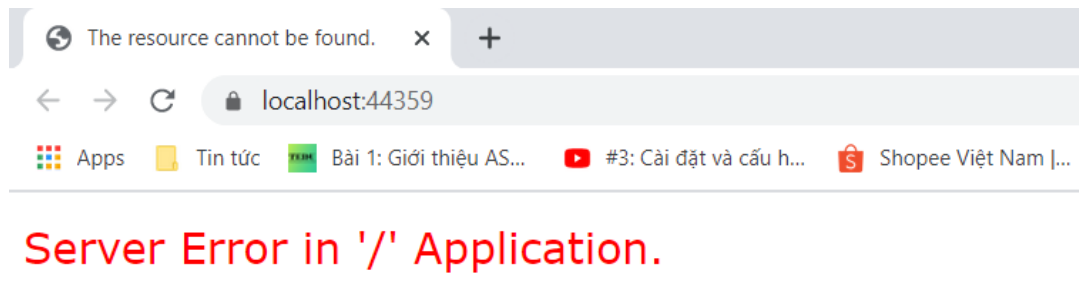
```

    }

    // GET: /ViDu/Welcome/
    public string Welcome()
    {
        return "Đây là action method Welcome ...";
    }
}

```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử sẽ có thông báo lỗi như sau

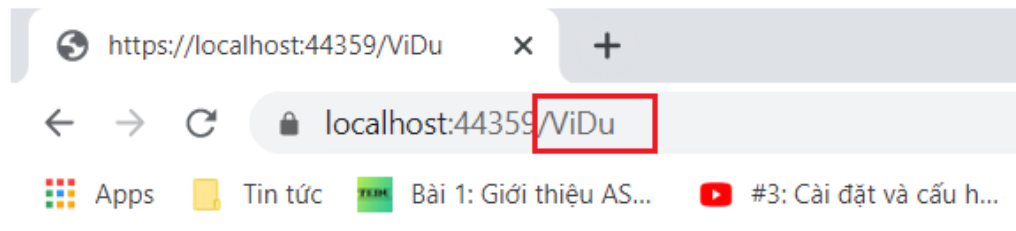


The resource cannot be found.

Description: HTTP 404. The resource you are looking for (or one of its dependencies) could have been removed, hac

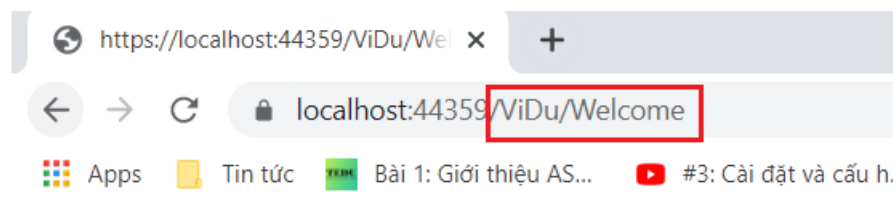
Requested URL: /

- Nhập lại URL thêm **/ViDu** vào sau địa chỉ localhost sẽ được kết quả hiển thị như sau



Đây là action **mặc định** của tôi...

- Nhập lại URL thêm **/ViDu/Welcome** vào sau địa chỉ localhost sẽ được kết quả hiển thị như sau



Đây là action method Welcome ...

*****ASP.NET MVC gọi các lớp controller khác nhau (và các action methods khác nhau bên trong chúng) tùy thuộc vào URL. URL routing logic mặc định được ASP.NET MVC sử dụng theo định dạng là: `/[Controller]/[ActionName]/[Parameters]`**

- Thiết lập định dạng cho routing trong file **App_Start/RouteConfig.cs**

```

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 }
        );
    }
}

```

Khi chạy ứng dụng không cung cấp bất kỳ URL nào, ứng dụng mặc định chọn controller "Home" và action method "Index" được chỉ định trong phần code mặc định ở trên.

Với URL thứ nhất <https://localhost:44359/ViDu> thì action method không có vậy mặc định là action method Index() được gọi.

Với URL thứ 2 <https://localhost:44359/ViDu/Welcome> thì action method Welcome() được gọi

- Sửa code của action method **Welcome()** trong **ViDuController.cs** như sau:

```

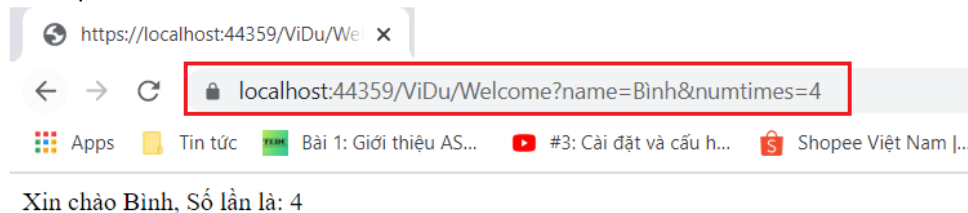
public string Welcome(string name, int numTimes = 1)
{
    return HttpUtility.HtmlEncode("Xin chào " + name + ", Số lần là: " + numTimes);
}

```

Ấn phím F5 (hoặc Ctrl+F5) để chạy thử sau đó nhập địa chỉ URL

<http://localhost:xxxx/ViDu/Welcome?name=Bình&numtimes=4>

được kết quả là



Ở đây không sử dụng phần **Parameters** trong URL của routing, các tham biến **name** và **numTimes** được chuyển qua query strings.

- Sửa tiếp code của action method **Welcome()** trong **ViDuController.cs** như sau:

```

public string Welcome(string name, int ID = 1)
{
    return HttpUtility.HtmlEncode("Xin chào " + name + ", ID: " + ID);
}

```

Ấn phím F5 (hoặc Ctrl+F5) để chạy thử sau đó nhập địa chỉ URL

<http://localhost:xxxx/ViDu/Welcome/3?name=Minh>

được kết quả là



Lần này phần **Parameters** trong URL của routing gắn với biến ID. Action method **Welcome()** có chứa một tham biến (**ID**) gắn với URL được mô tả trong method **RegisterRoutes()**

*****Trong ASP.NET MVC, việc truyền các tham số dưới dạng route data (như đã làm với ID ở trên) thường được dùng hơn là sử dụng Query String.**

- Thêm một route để truyền hai tham biến **name** và **id** qua URL trong file **App_Start/RouteConfig.cs**:

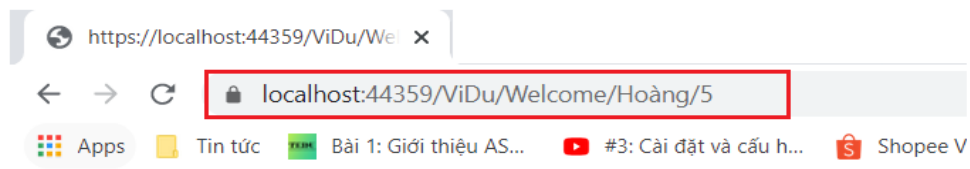
```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

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

    routes.MapRoute(
        name: "ViDu",
        url: "{controller}/{action}/{name}/{id}"
    );
}
```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử sau đó nhập địa chỉ URL <http://localhost:xxxx/ViDu/Welcome/Hoàng/5>

được kết quả là



Xin chào Hoàng, ID: 5

- Sửa route trong file **App_Start/RouteConfig.cs** để controller mặc định là ViDu:

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

    routes.MapRoute(
        name: "Default",
        url: "{controller}/{action}/{id}",
        defaults: new { controller = "ViDu", action = "Index", id =
UrlParameter.Optional }
    );

    routes.MapRoute(
        name: "ViDu",
        url: "{controller}/{action}/{name}/{id}"
    );
}
```

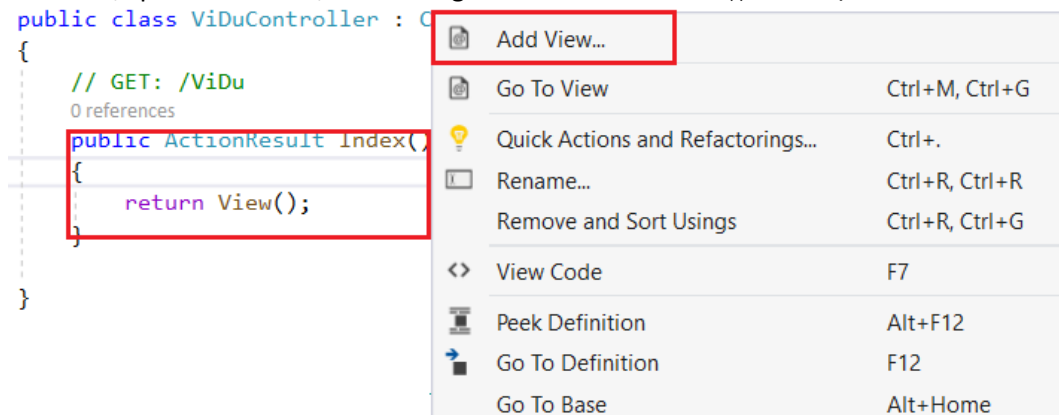
Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.

7. Thêm một View cho ViDuController

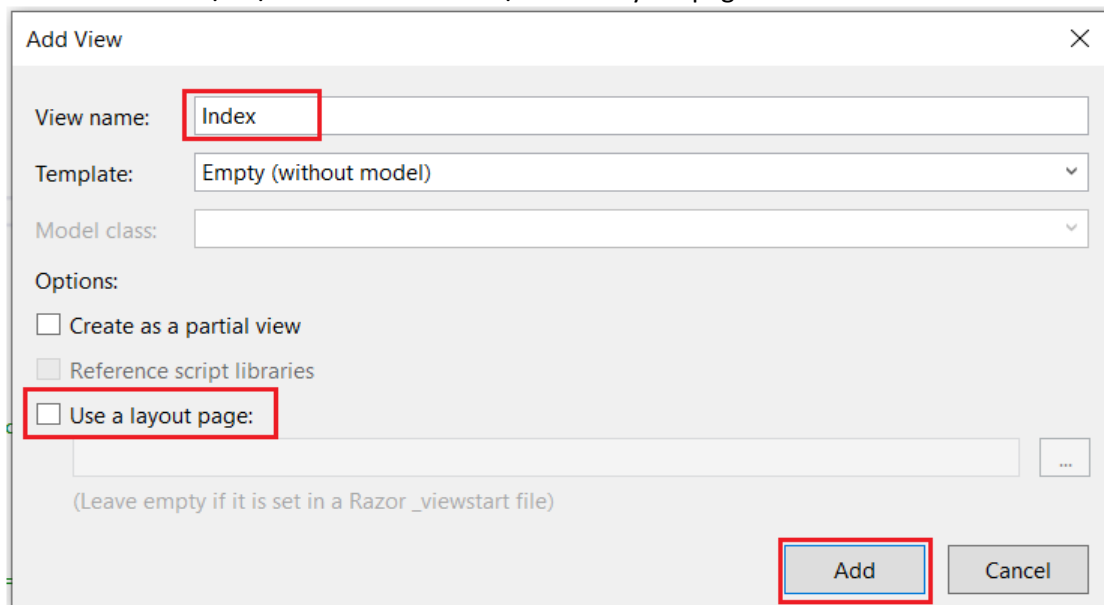
- Sửa code của ViDuController.cs như sau:

```
public class ViDuController : Controller
{
    // GET: /ViDu
    public ActionResult Index()
    {
        return View();
    }
}
```

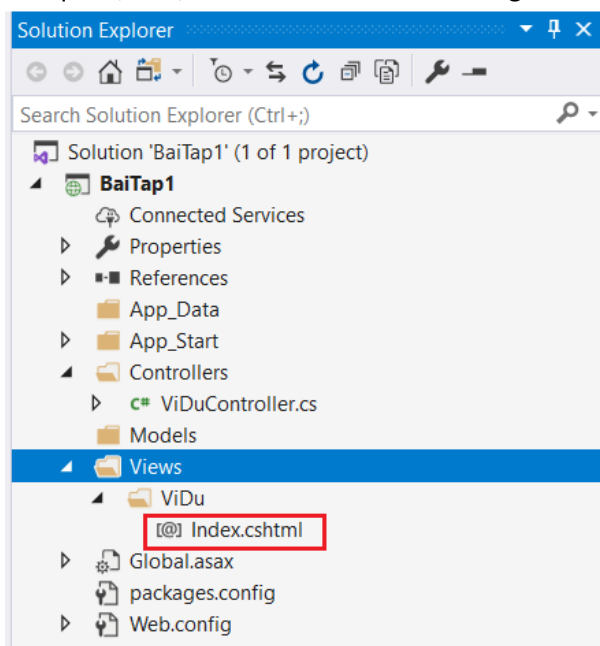
- Kích chuột phải vào khu vực bên trong action method **Index()** và chọn Add View...



- Để View name mặc định là Index và bỏ chọn Use a layout page và kích vào nút Add



Kết quả tạo một view là **Index.cshtml** trong folder **Views/Vidu**



Với code trong trang **Index.cshtml** như sau


```
Index.cshtml
1
2  @{
3      Layout = null;
4  }
5
6  <!DOCTYPE html>
7
8  <html>
9  <head>
10     <meta name="viewport" content="width=device-width" />
11     <title>Index</title>
12 </head>
13 <body>
14     <div>
15     </div>
16 </body>
17 </html>
```

*** Trong ASP.NET MVC, tên của View (file .cshtml) phải trùng với tên của action method tương ứng trong Controller

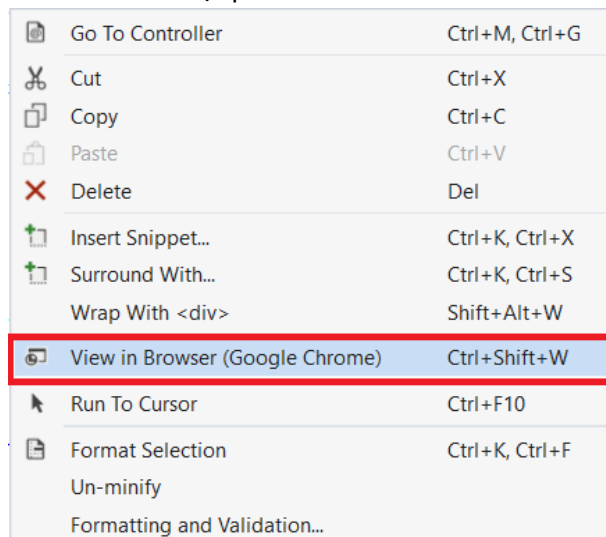
- Sửa code của Index.cshtml như sau:

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
</head>
<body>
    <div>
        <p>Xin chào từ View Template của chúng tôi!</p>
    </div>
</body>
</html>
```

Cách 1: kích chuột phải vào cửa sổ code của view rồi chọn View in Browser để xem view trên trình duyệt.



Cách 2: Ấn phím F5 (hoặc Ctrl+F5) và nhập URL với action method tương ứng

8. Truyền dữ liệu từ Controller đến View sử dụng ViewBag

- Sửa code của ViDuController.cs thêm action method **Welcome()**:

```
public class ViDuController : Controller
```

```

{
    // GET: /ViDu
    public ActionResult Index()
    {
        return View();
    }

    public ActionResult Welcome(string name, int id = 1)
    {
        ViewBag.Message = "Xin chào " + name;
        ViewBag.NumTimes = id;

        return View();
    }
}

```

Dùng **ViewBag** để truyền biến từ Controller sang View. **ViewBag** là một đối tượng động, có thể đặt bất cứ thứ gì vào đó; Đối tượng ViewBag không có thuộc tính xác định nào cho đến khi đặt thứ gì đó vào bên trong nó.

- Thêm một View cho action method `Welcome()`: Kích chuột phải vào khu vực bên trong action method `Welcome()` và chọn Add View...

- Sửa code của **Welcome.cshtml** như sau:

```

@{
    Layout = null;
}

<!DOCTYPE html>

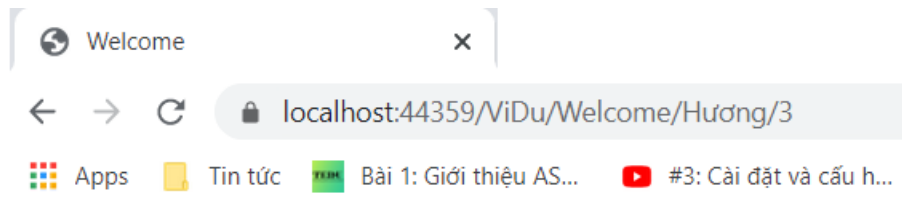
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Welcome</title>
</head>
<body>
    <div>
        <ul>
            @for (int i = 0; i < ViewBag.NumTimes; i++)
            {
                <li>@ViewBag.Message</li>
            }
        </ul>
    </div>
</body>
</html>

```

***** Chú ý: ViewBag.NumTimes và ViewBag.Message đã được gán giá trị trong action method Welcome() của ViDuController.cs.**

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử sau đó nhập địa chỉ URL <http://localhost:xxxx/ViDu/Welcome/Huong/3>

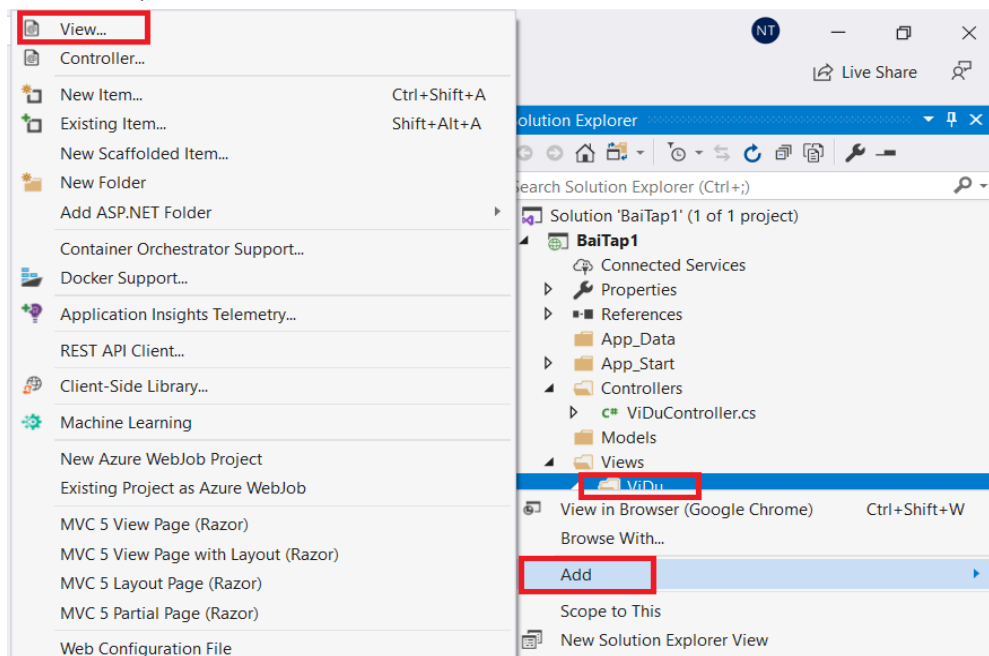
được kết quả là



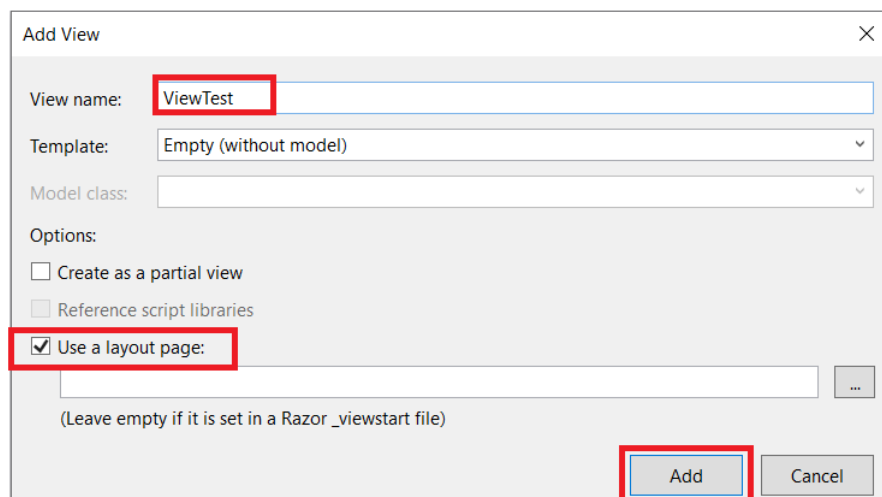
- Xin chào Hương
- Xin chào Hương
- Xin chào Hương

9. Tạo một View với layout

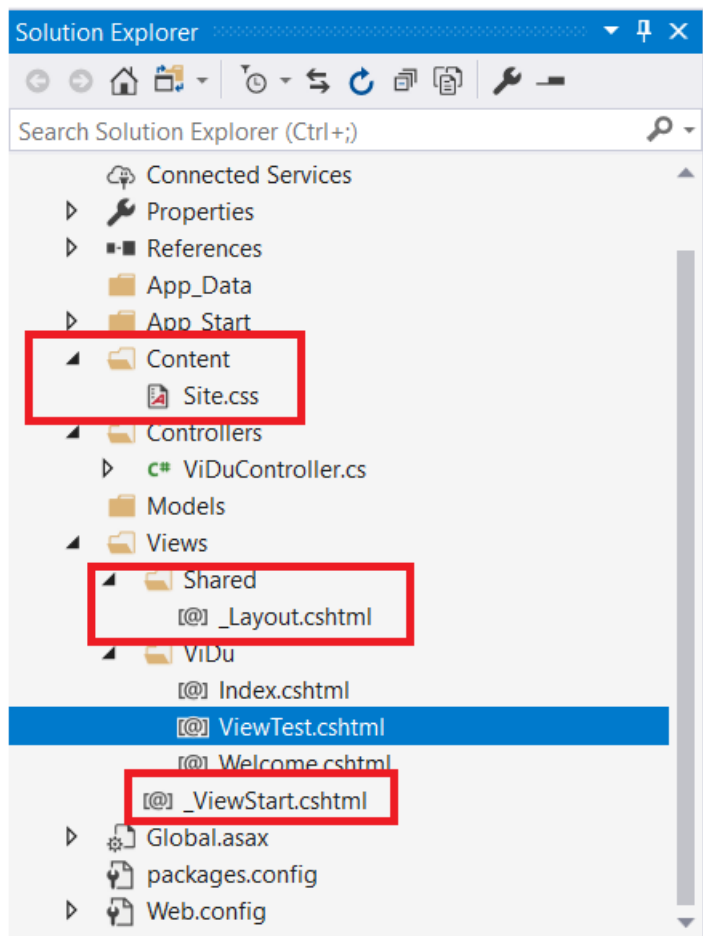
- Kích chuột phải vào folder ViDu và chọn Add sau đó chọn View...



- Nhập tên cho View và chọn Use a layout page sau đó kích nút Add như hình



- Cửa sổ Solution Explorer sau khi View được thêm sẽ có thêm một số folder và file hỗ trợ cho layout của View mới.



- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử

Lab 02 – Tạo một project ASP.NET MVC với Template có sẵn

- Tạo một Project có tên là BaiTap2 và chọn Template MVC

Configure your new project

ASP.NET Web Application (.NET Framework) C# Windows Cloud Web

Project name
BaiTap2

Location
F:\MyData\CaCachKhoaTin\LapTrinhUD_CSDL tren Web\Demo_Code\


Solution name
BaiTap2


☒ Place solution and project in the same directory


Framework
.NET Framework 4.7.2


Back Create


Create a new ASP.NET Web Application

**Empty**
An empty project template for creating ASP.NET applications. This template does not have any content in it.

**Web Forms**
A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

**MVC**
A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

**Web API**
A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

**Single Page Application**
A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

Authentication
No Authentication
[Change](#)

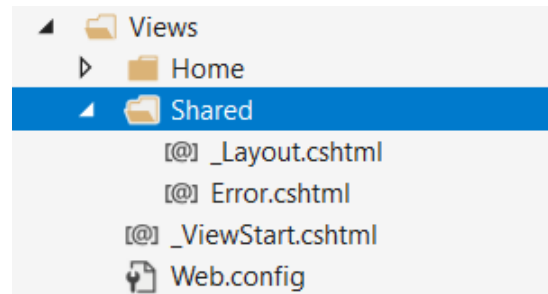
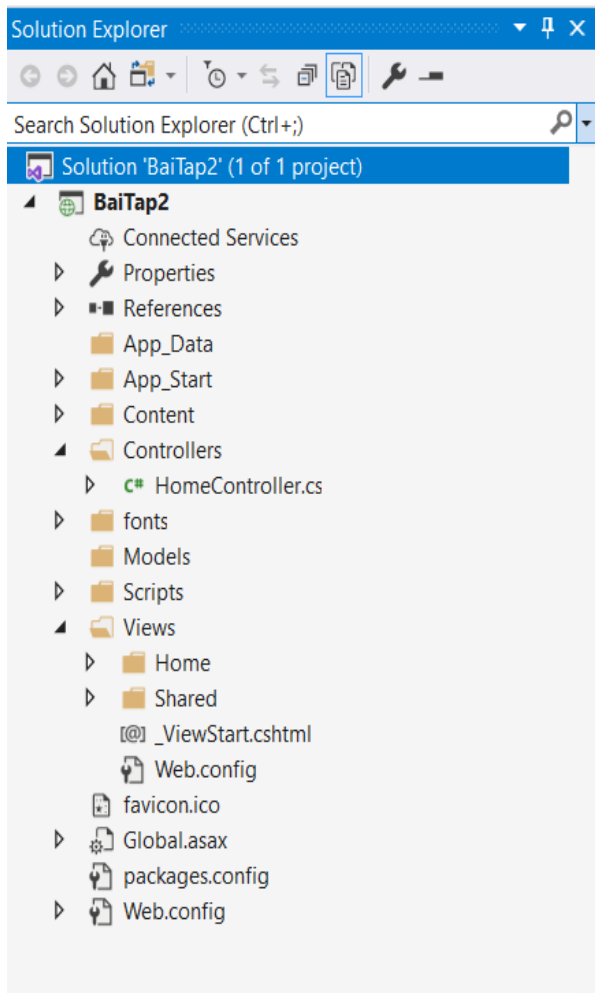
Add folders & core references
☐ Web Forms
☒ MVC
☐ Web API

Advanced
☒ Configure for HTTPS
☐ Docker support
(Requires [Docker Desktop](#))
☐ Also create a project for unit tests

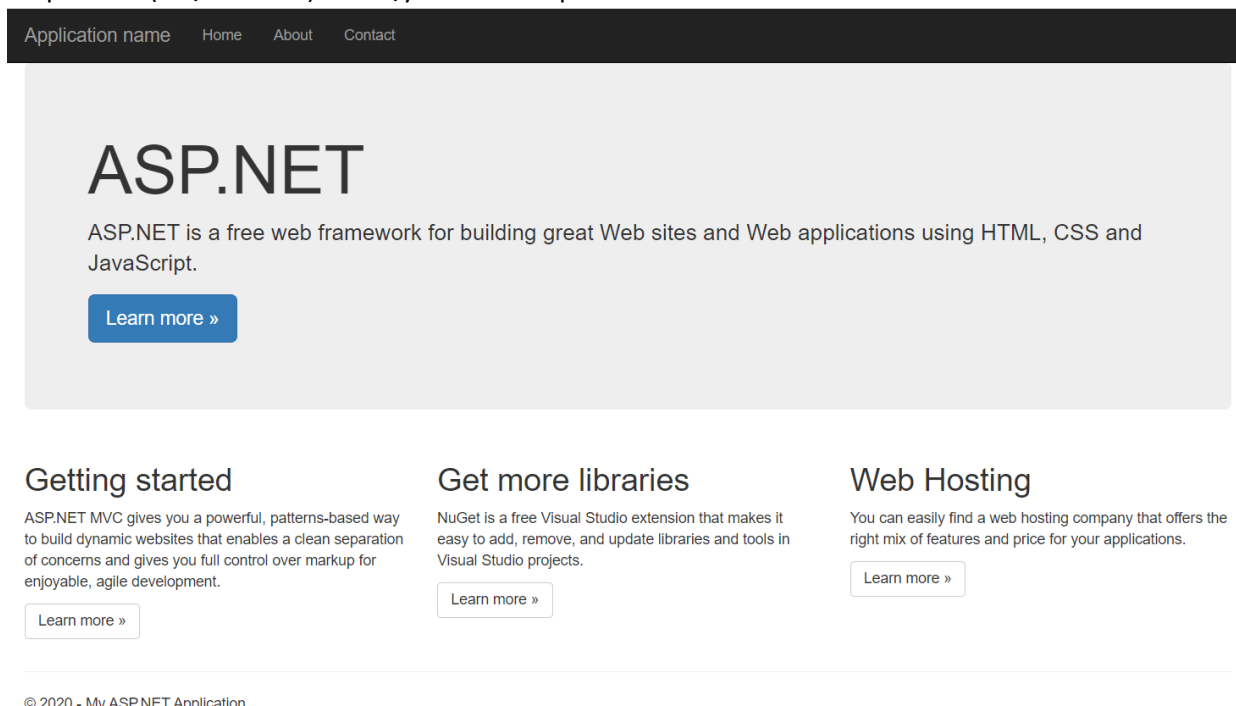
2. Các folder trong Project

***Chú ý: các thành phần cơ bản trong một project ASP.Net MVC

- **Properties:** Chứa các thuộc tính của project.
- **References:** Chứa các thư viện được sử dụng trong Project
- **App_Data:** Thư mục chứa file dữ liệu của Project nếu add cả file dữ liệu vào project
- **App_Start:** Thư mục chứa các file cấu hình khởi động và biên dịch của project. Chú ý đến 2 file
 - **FilterConfig.cs** dùng để khai báo các filter sử dụng trước khi thực hiện 1 hành động nào đó
 - **RouteConfig.cs** định nghĩa các routes trong project.
- **Content:** Thư mục chứa các file .CSS (dùng cho các view)
- **Controllers:** Thư mục chứa các file xxController.cs là các Controller
- **Models:** Thư mục chứa các file .cs là các Model gắn với các bảng trong CSDL.
- **Scripts:** Thư mục chứa các file .JS (dùng cho các view)
- **Views:** Thư mục chứa các view trong các folder, mỗi view là một file HTML với đuôi là .cshtml.
 - **Shared:** Thư mục chứa các file HTML với đuôi là .cshtml dùng chung trong các view.
- **Global.aspx:** File chứa các khai báo chung sử dụng cho toàn bộ project (Biến toàn cục).
- **package.config:** File quản lý các package chúng ta cài vào
- **Web.config:** File quan trọng, định nghĩa các cài đặt hệ thống cho project.

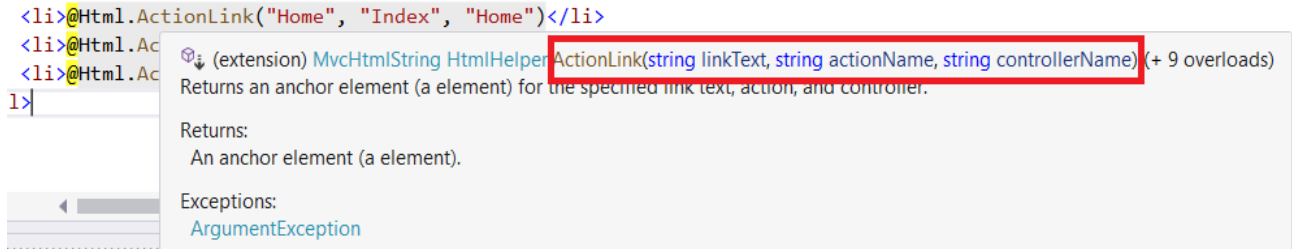


3. Ấn phím F5 (hoặc Ctrl+F5) để chạy thử có kết quả là



Xem code trong Controller và View.

Mở file `_Layout.cshtml` trong View/Shared. Đặt con trỏ vào ActionLink để xem cú pháp của method



@Html.ActionLink là cách tạo kết nối action trong một controller và thực thi action đó. Cú pháp cụ thể là **@Html.ActionLink (Text của link, Tên action, Tên controller)**

@Html.ActionLink("Home", "Index", "Home"): là liên kết hiển thị trên trang web là "Home", liên kết để thực thi action method có tên là Index() ở trong Controller có tên là "Home"

4. Tạo thêm một nút lệnh trên menu là "Liên kết của tôi":

- Thêm một **@Html.ActionLink("Liên kết của tôi", "MyLink", "Home")** vào trong file _Layout.cshtml

```
</div>
<div class="navbar-collapse collapse">
    <ul class="nav navbar-nav">
        <li>@Html.ActionLink("Home", "Index", "Home")</li>
        <li>@Html.ActionLink("About", "About", "Home")</li>
        <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
        <li>@Html.ActionLink("Liên kết của tôi", "MyLink", "Home")</li>
    </ul>
</div>
```

- Trong HomeController.cs thêm một Action method tên là MyLink

```
namespace BaiTap2.Controllers
{
    public class HomeController : Controller
    {
        .....

        public ActionResult MyLink()
        {
            ViewBag.Message = "Liên kết của tôi.";
            return View();
        }
    }
}
```

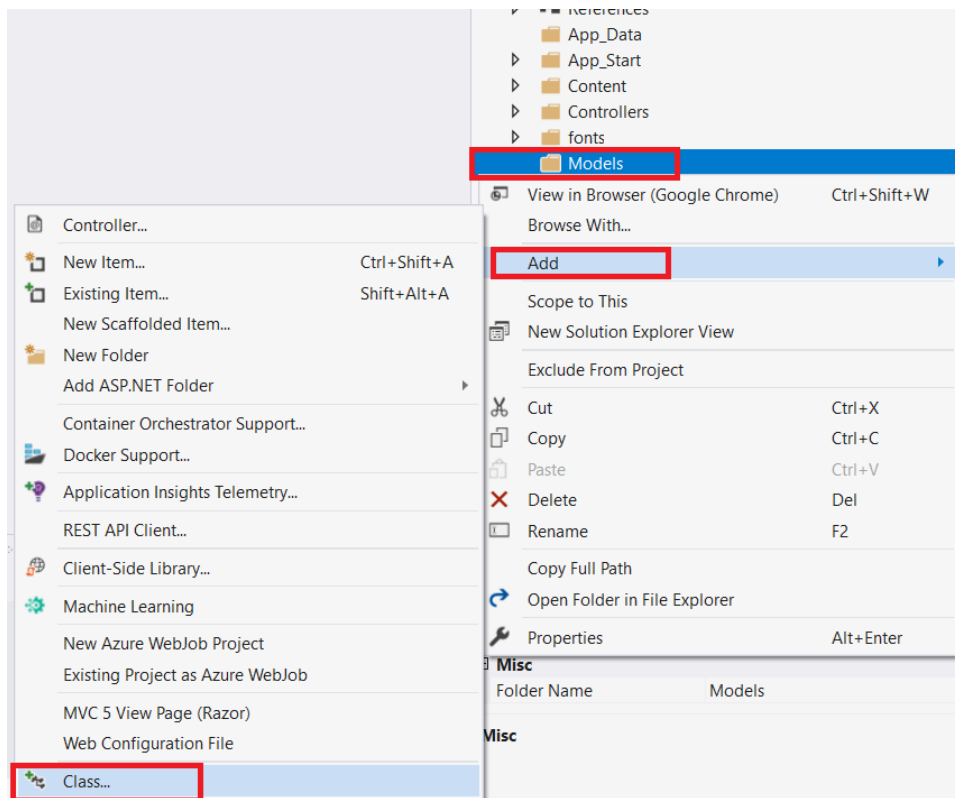
- Tạo một View có tên là MyLink cho action method MyLink() và sửa code của MyLink.cshtml như sau:

```
@{
    ViewBag.Title = "MyLink";
}

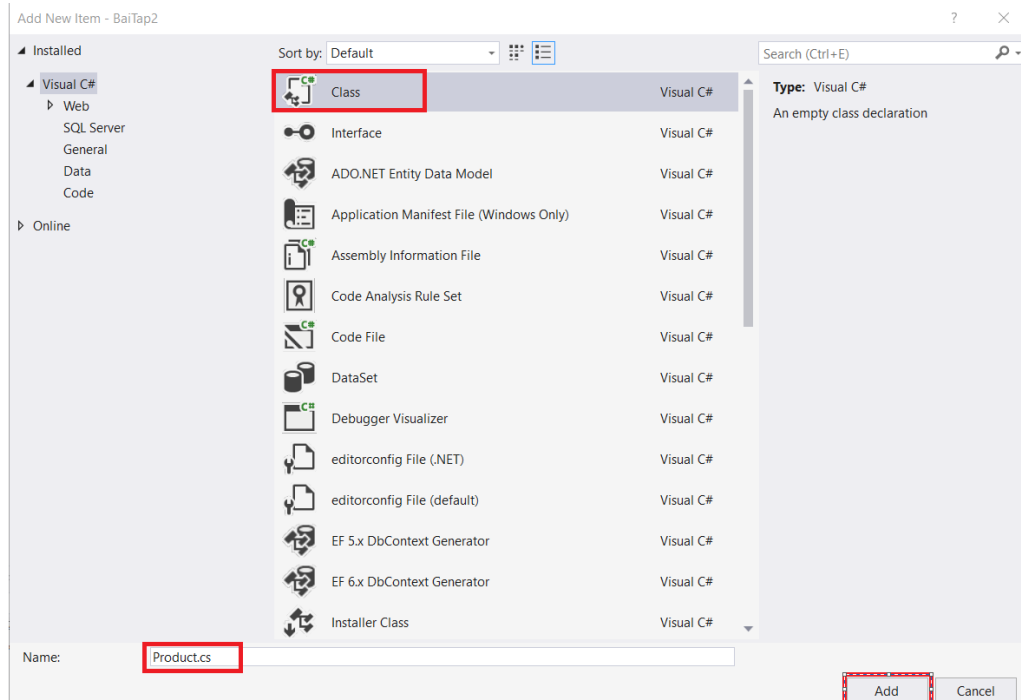
<h2>@ViewBag.Title.</h2>
<h3>@ViewBag.Message</h3>

<p>Đây là trang liên kết của tôi để test ActionLink trong ASP.NET MVC</p>
```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử kiểm tra hoạt động của Link "Liên kết của tôi"
5. Tạo một Model
- Kích chuột phải vào folder Model chọn Add sau đó chọn Class...



- Chọn Class, đặt tên lớp là Product.cs sau đó kích chọn Add

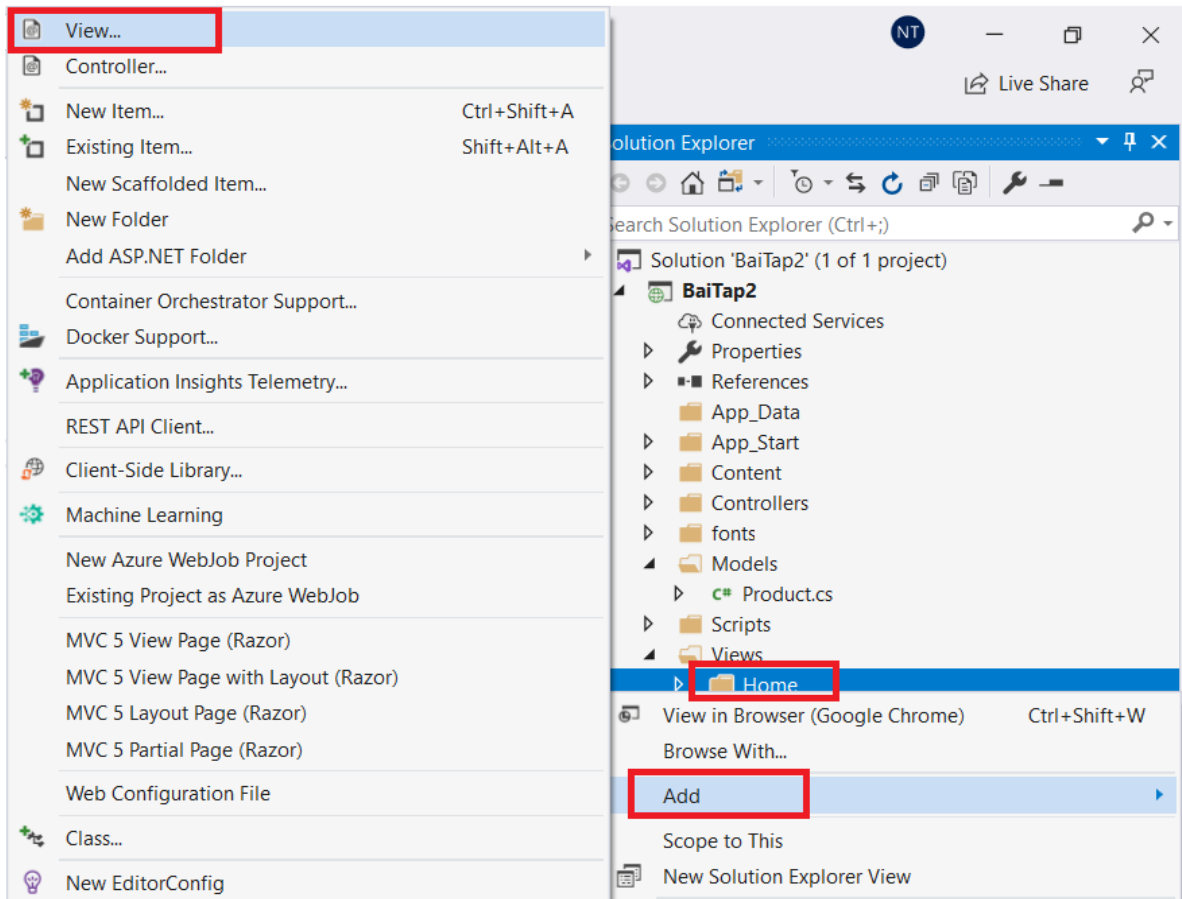


- Nhập code vào file Product.cs như sau

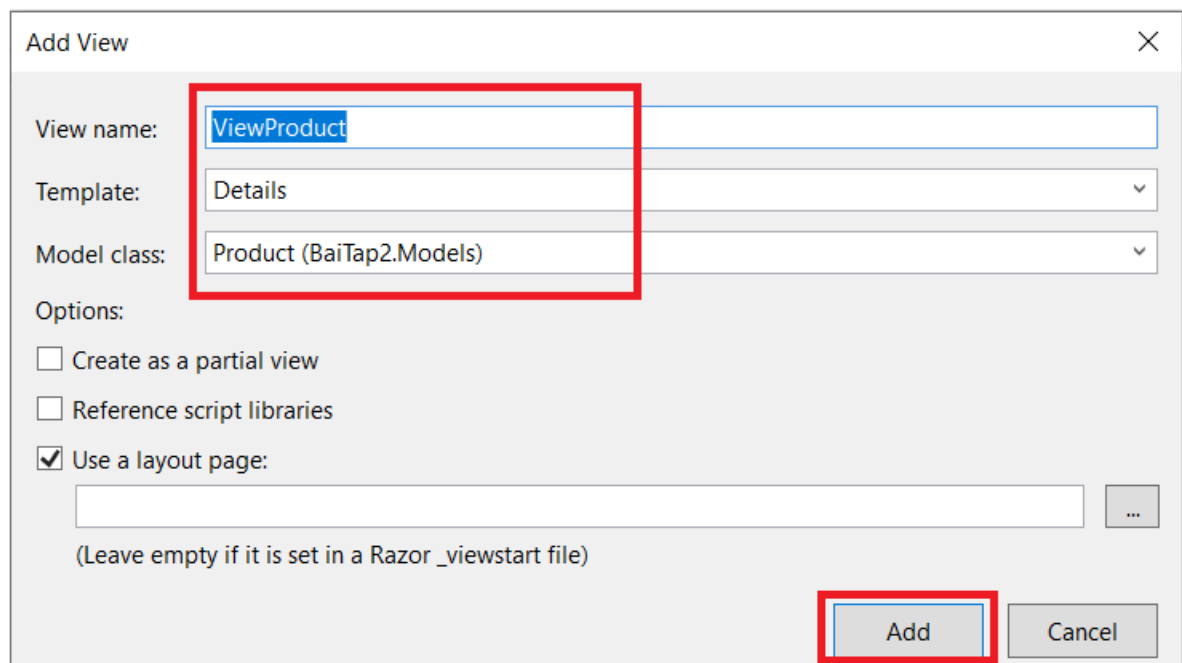
```
public class Product
{
    public int MaSP { get; set; }
    public string TenSP { get; set; }
    public decimal Gia { get; set; }
    public int SoLuong { get; set; }

    public Product()
    {
    }
}
```


- Tạo một view cho model Product: Kích chuột phải vào folder Home sau đó chọn Add rồi chọn View...



- Nhập tên view là ViewProduct, Template là Details và Model class là Product sau đó kích vào nút Add



- Xem code trong cửa sổ của file ViewProduct
- Trong HomeController.cs thêm một Action method cho ViewProduct

```
namespace BaiTap2.Controllers
{
    public class HomeController : Controller
    {
        .....

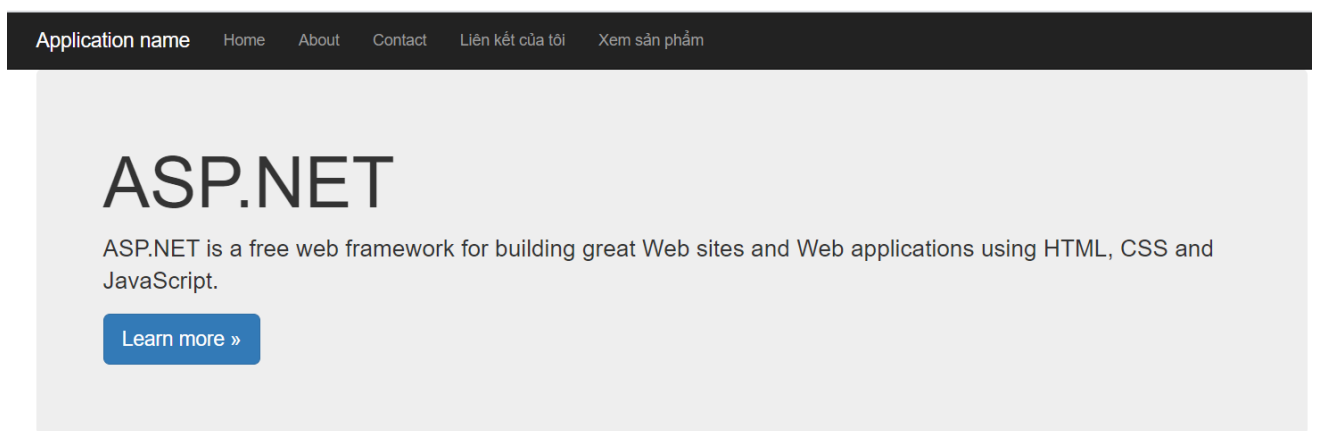
        public ActionResult ViewProduct()
```

```
{
    Product product = new Product();
    product.MaSP = 5;
    product.TenSP = "Lập trình ASP.NET MVC";
    product.Gia = 1250;
    product.SoLuong = 10;

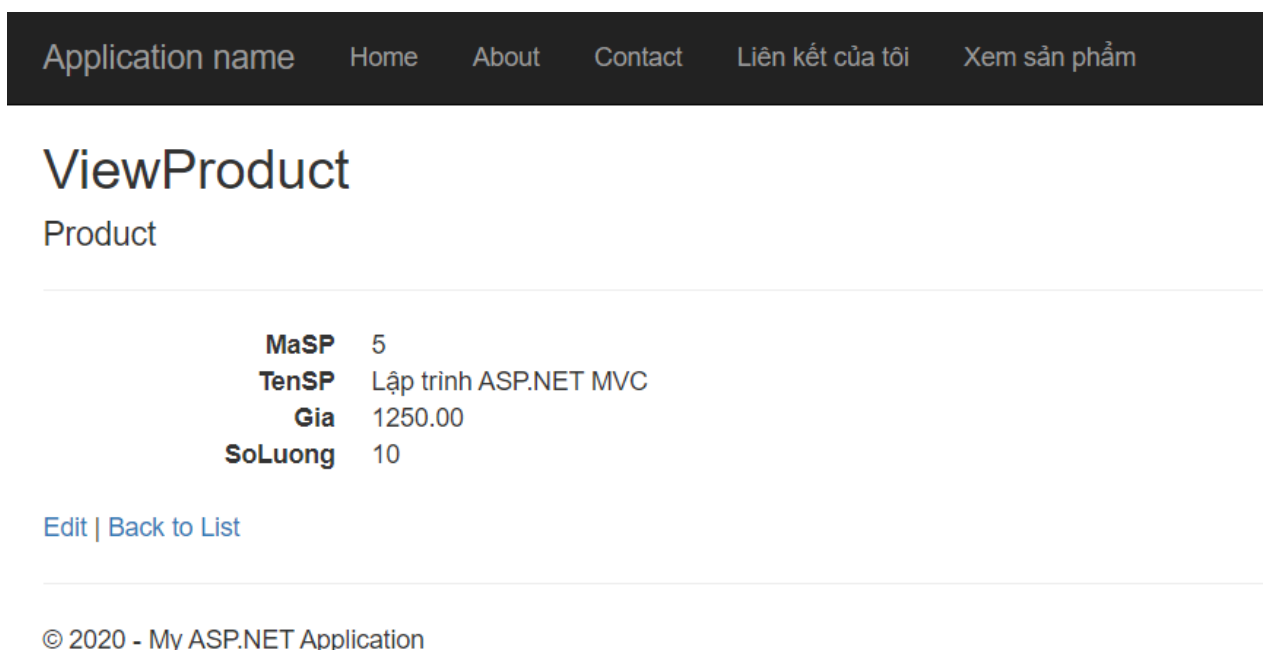
    ViewBag.Message = "Xem sản phẩm.";

    return View(product);
}
```

- Thêm một ActionLink trong file _Layout.cshtml để liên kết với ViewProduct:
`@Html.ActionLink("Xem sản phẩm", "ViewProduct", "Home")`
- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử



Kích vào "Xem sản phẩm"



Lab 03 – Tự làm

Tạo một project ASP.NET MVC có tên là **BaiTap3** sử dụng template MVC.

1. Tạo một Model Student.cs như sau:

```
...
using System.ComponentModel.DataAnnotations;
namespace BaiTap3.Models
{
    public class Student
    {
        [Display(Name = "Mã sinh viên")]
        public int ID { get; set; }
        [Display(Name = "Tên sinh viên")]
        public string Name { get; set; }
        [Display(Name = "Điện thoại")]
        public string Phone { get; set; }
        [Display(Name = "Ngày bắt đầu")]
        public DateTime StartDate { get; set; }

        public Student()
        {
            ID = 12345;
            Name = "Hoàng Văn Hùng";
            Phone = "0918385798";
            StartDate = DateTime.Now;
        }
    }
}
```

2. Tạo một view tên là **ViewStudent** để hiển thị thông tin sinh viên.
3. Tạo một ActionResult **ViewStudent()** trong HomeController và tạo một Student sau đó chuyển qua hiển thị trong view.
4. Khi chạy project trang chủ hiển thị như sau:

Học lập trình MVC Trang chủ Sinh viên

Thông tin sinh viên

© 2020 - Ứng dụng ASP.NET của "Cần thay thế đoạn văn bản này bằng họ tên của bạn"

5. Khi kích vào “Sinh viên” màn hình hiển thị thông tin sinh viên

Học lập trình MVC Trang chủ Sinh viên

Thông tin sinh viên

Tên sinh viên	Hoàng Văn Hùng
Điện thoại	0918385798
Ngày bắt đầu	07/09/2020 3:03:17 pm

[Quay lại](#)

© 2020 - Ứng dụng ASP.NET của "Cần thay thế đoạn văn bản này bằng họ tên của bạn"