

**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**  
**KHOA CÔNG NGHỆ THÔNG TIN**

**CÔNG NGHỆ PHÁT TRIỂN WEBSITE**  
**(LẬP TRÌNH WEBSITE VỚI ASP.NET MVC 5)**

**CHƯƠNG 1:**

**TỔNG QUAN VỀ ASP.NET MVC 5**



**Giảng Viên: ThS. Dương Thành Phết**

**Email: phetcm@gmail.com**

**Website: phetcm@gmail.com -  
www.thayphet.net**

**Tel: 091815867**

## *Kiến thức:*

- Trình bài được các khái niệm, đặc tính về ASP.Net MVC 5 Framework
- So sánh sự khác biệt ASP.Net MVC 5 với Web Form
- Trình bày khái niệm về các đối tượng trong mô hình MVC

## *Kỹ năng:*

- Thực hiện được việc truy cập với LINQ
- Xây dựng được ứng dụng truy xuất dữ liệu với mô hình ASP.Net MVC 5

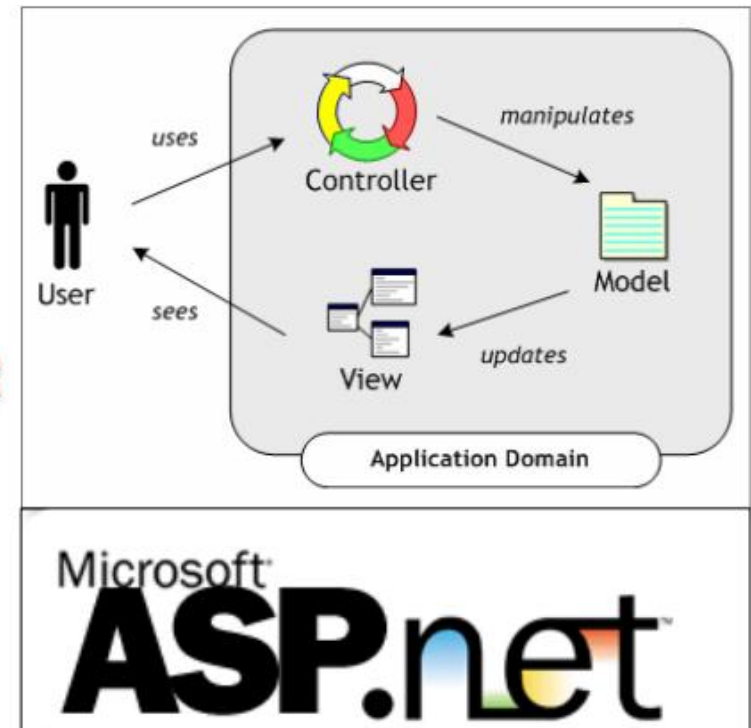
# NỘI DUNG

1. Giới thiệu về ASP.NET MVC 5
2. Xây dựng ứng dụng Web MVC 5
3. Giới thiệu về **C**ontroller
4. Giới thiệu về **V**iew
5. Giới thiệu về **M**odels
6. Giới thiệu về Định tuyến
7. Xây dựng ứng dụng web MVC với CSDL

# 1. GIỚI THIỆU VỀ ASP.NET MVC

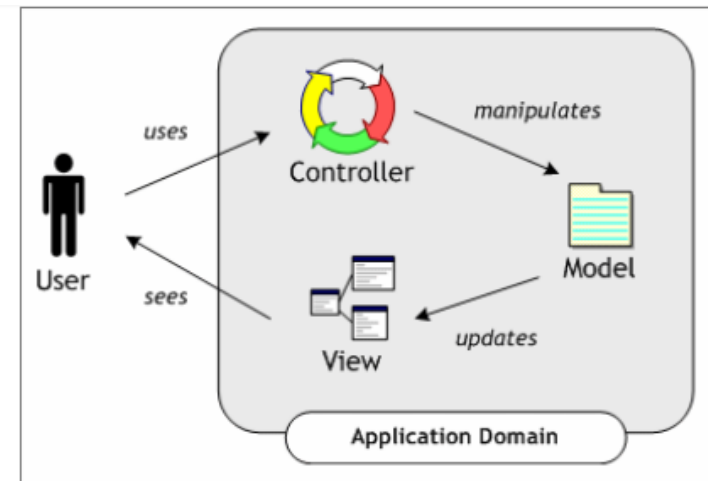
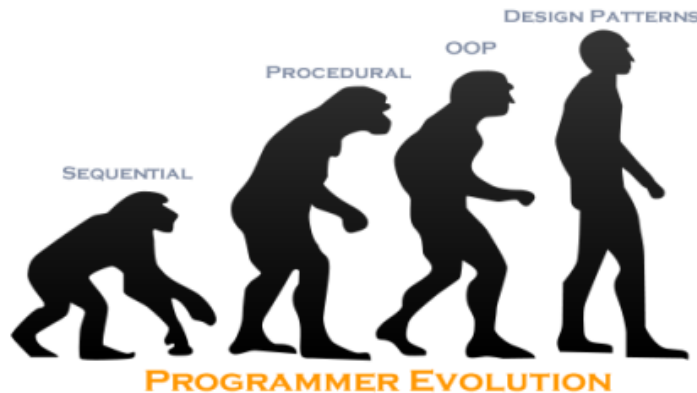
## 1.1. ASP.NET MVC là gì ?

Microsoft  
**ASP.net** MVC =



# 1. GIỚI THIỆU VỀ ASP.NET MVC

## 1.1. ASP.NET MVC là gì ?

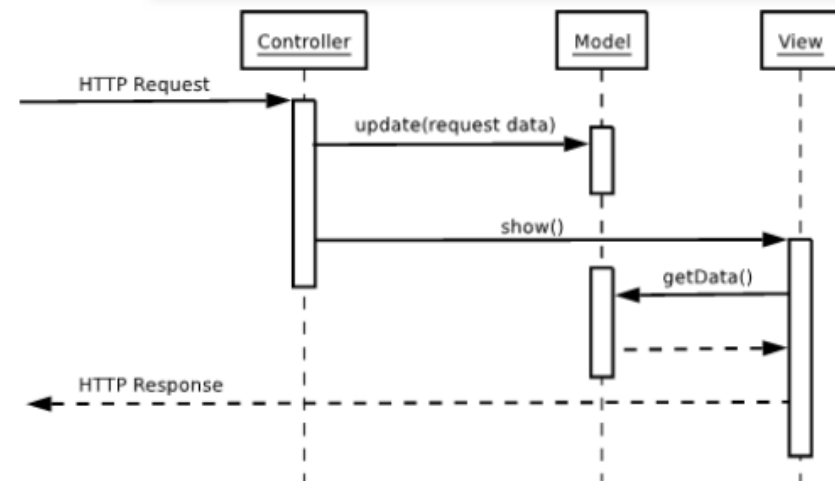


### □ MVC Pattern

- ✗ Model
- ✗ Controller
- ✗ View

### □ MVC Framework

- ✗ Java: Struts, JSF, Spring...
- ✗ PHP: Zend
- ✗ **ASP.NET MVC**



# 1. GIỚI THIỆU VỀ ASP.NET MVC

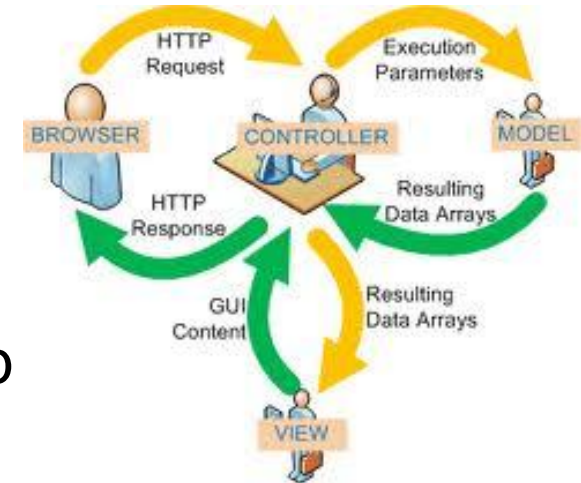
## 1.1. ASP.NET MVC là gì ?

### ✓ Controller

- Nhận yêu cầu từ user
- Xử lý và xây dựng model phù hợp
- Chuyển Model cho View

✓ **View:** Tiếp nhận Model từ Controller để sinh giao diện phù hợp

✓ **Model:** Chứa dữ liệu chia sẻ chung giữa Controller và View



# 1. GIỚI THIỆU VỀ ASP.NET MVC

## Models:

- ✓ Có nhiệm vụ lưu trữ thông tin, trạng thái của các đối tượng, như là 1 lớp được ánh xạ từ 1 bảng trong **CSDL**.
- ✓ Chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất database, như các Class, hàm xử lý..

Ví dụ lớp *Product* được sử dụng để mô tả dữ liệu từ bảng *Products* trong **SQL**, bao gồm *ProductID*, *OrderDate*...

# 1. GIỚI THIỆU VỀ ASP.NET MVC

## Views

- ✓ Chịu trách nhiệm hiển thị các thông tin lên cho người dùng thông qua giao diện.
- ✓ Chứa các đối tượng GUI(Textbox, images... ).
- ✓ Các thông tin cần hiển thị được lấy từ thành phần **Models**.

Ví dụ: Đối tượng *Product* có "*Edit*" view bao gồm các textboxes, các dropdowns và checkboxes để chỉnh sửa các thuộc tính của sản phẩm;



# 1. GIỚI THIỆU VỀ ASP.NET MVC

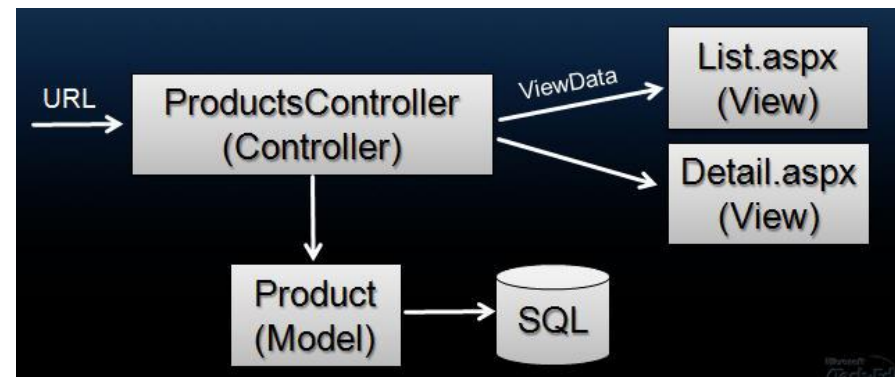
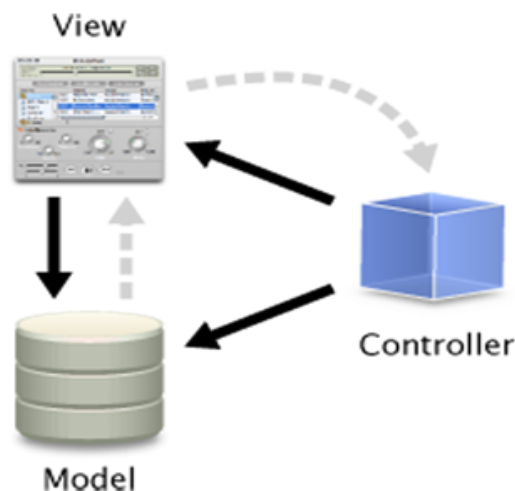
## Controllers

- ✓ Xử lý các tác động về mặt giao diện, các thao tác đối với models, và chọn view để hiển thị ra màn hình.
- ✓ Điều hướng các yêu cầu từ người dùng và gọi phương thức xử lý
- ✓ Trong **MVC**, view chỉ có tác dụng hiển thị giao diện, còn điều khiển vẫn do **Controllers** đảm trách.

# 1. GIỚI THIỆU VỀ ASP.NET MVC

## 1.2. MVC làm việc như thế nào?

- ✓ User tương tác với View, bằng cách click vào button, gửi yêu cầu đi.
- ✓ Controller nhận và điều hướng đến phương thức xử lý ở Model.
- ✓ Model nhận thông tin và thực thi các yêu cầu, View sẽ nhận kết quả từ Model và hiển thị lại cho người dùng.



# 1. GIỚI THIỆU VỀ ASP.NET MVC

## 1.3. Ưu & Nhược điểm của MVC

### Ưu điểm:

- ✓ Thể hiện tính chuyên nghiệp trong lập trình, phân tích thiết kế.
- ✓ Được chia thành các thành phần độc lập nên giúp phát triển ứng dụng nhanh, dễ nâng cấp, bảo trì..
- ✓ Ứng dụng tạo ra chạy ổn định trên Windows
- ✓ Đáp ứng nhiều loại thiết bị truy cập
- ✓ An toàn, Dễ tích hợp

# 1. GIỚI THIỆU VỀ ASP.NET MVC

## 1.3. Ưu & Nhược điểm của MVC

### Khuyết điểm:

- ✓ Đối với dự án nhỏ việc áp dụng mô hình MVC gây cồng kềnh, tốn thời gian trong quá trình phát triển.
- ✓ Tốn thời gian trung chuyển dữ liệu của các thành phần.

# 1. GIỚI THIỆU VỀ ASP.NET MVC

## 1.4. Sự khác biệt với WebForm

| Tính năng              | ASP.NET 2.0   | ASP.NET MVC   |
|------------------------|---|---|
| Kiến trúc chương trình | Kiến trúc mô hình WebForm → Business → Database   | Kiến trúc sử dụng việc phân chia chương trình thành Controllers, Models, Views              |
| Cú pháp chương trình   | Sử dụng cú pháp của webform, tất các sự kiện và controls do server quản lý                                      | Các sự kiện được điều khiển bởi controllers, các controls không do server do server quản lý |
| Truy cập dữ liệu       | Sử dụng hầu hết các công nghệ truy cập dữ liệu trong ứng dụng   | Phần lớn dùng LINQ to SQL class để tạo mô hình truy cập đối tượng                           |
| Debug                  | Debug chương trình phải thực hiện tất cả bao gồm các lớp truy cập dữ liệu, sự hiển thị, điều khiển các controls | Debug có thể sử dụng các unit test kiểm tra các phương thức trong controller                |

# 1. GIỚI THIỆU VỀ ASP.NET MVC

## 1.4. Sự khác biệt với WebForm

| Tính năng                | ASP.NET 2.0   | ASP.NET MVC  |
|--------------------------|---|--|
| Tốc độ phân tải          | Tốc độ phân tải chậm khi trong trang có quá nhiều các controls vì ViewState quá lớn | Phân tải nhanh hơn do không phải quản lý ViewState để quản lý các control trong trang          |
| Tương tác với javascript | Tương tác với javascript khó khăn vì các controls được điều khiển bởi server        | Tương tác với javascript dễ dàng vì các đối tượng không do server quản lý điều khiển không khó |
| URL Address              | Cấu trúc địa chỉ URL có dạng <filename>.aspx?&<các tham số>                         | Cấu trúc địa chỉ rành mạch theo dạng Controllers/Action/Id                                     |

# 1. GIỚI THIỆU VỀ ASP.NET MVC

---

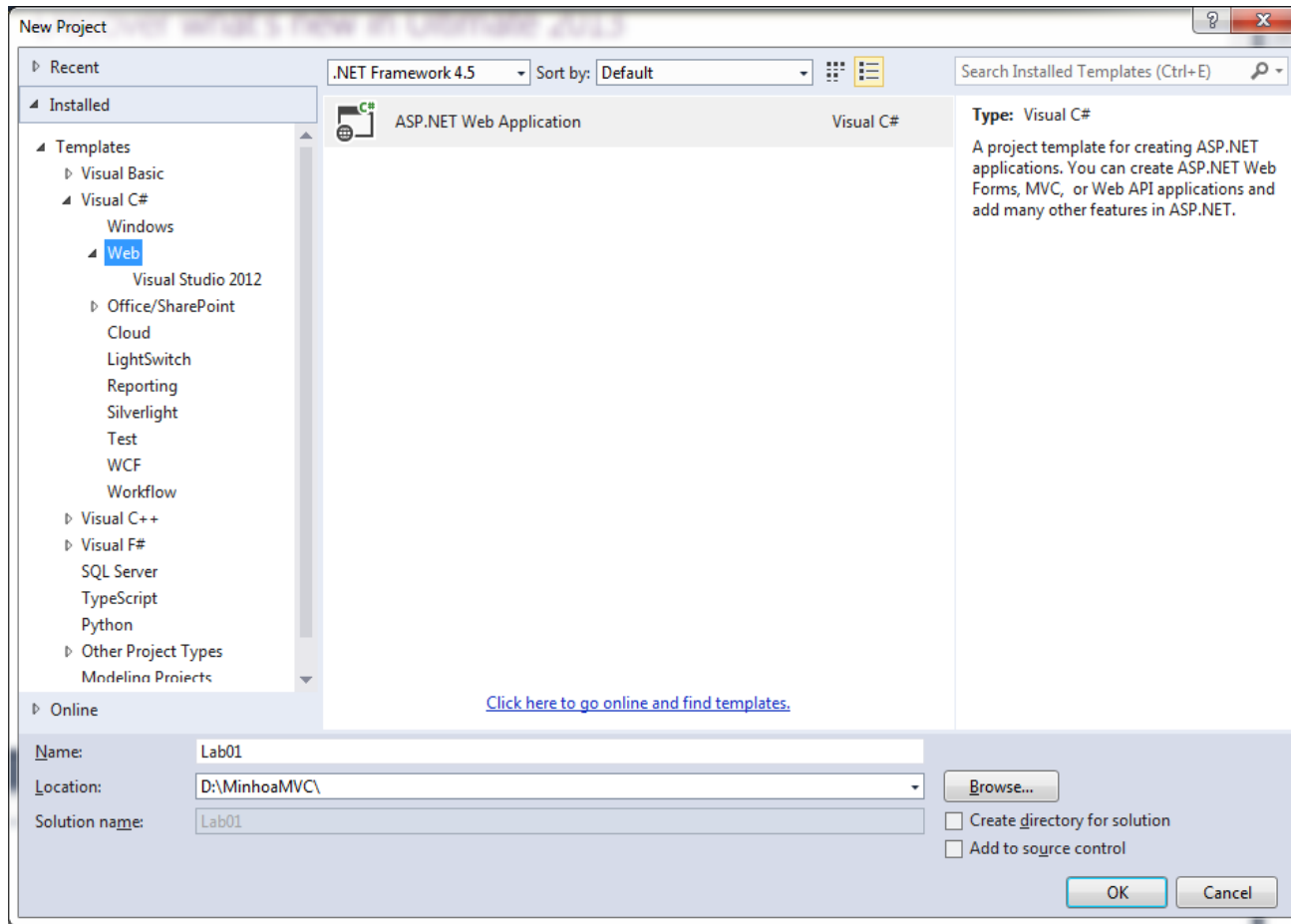
## 1.5. Lợi ích web dựa trên mô hình MVC:

- ✓ Dễ dàng quản lý sự phức tạp của ứng dụng bằng cách chia ứng dụng thành model, view, controller
- ✓ Không sử dụng view state hoặc server-based form. Điều này tốt cho những lập trình viên muốn quản lý hết các khía cạnh của một ứng dụng.
- ✓ Sử dụng mẫu Front Controller, mẫu này giúp quản lý các requests (yêu cầu) chỉ thông qua một Controller. Hỗ trợ tốt hơn cho mô hình phát triển ứng dụng hướng kiểm thử (TDD)
- ✓ Hỗ trợ tốt cho các ứng dụng được xây dựng bởi những đội có nhiều lập trình viên và thiết kế mà vẫn quản lý được tính năng của ứng dụng

## 2. XÂY DỰNG ỨNG DỤNG WEB

### 2.1. Tạo mới Project

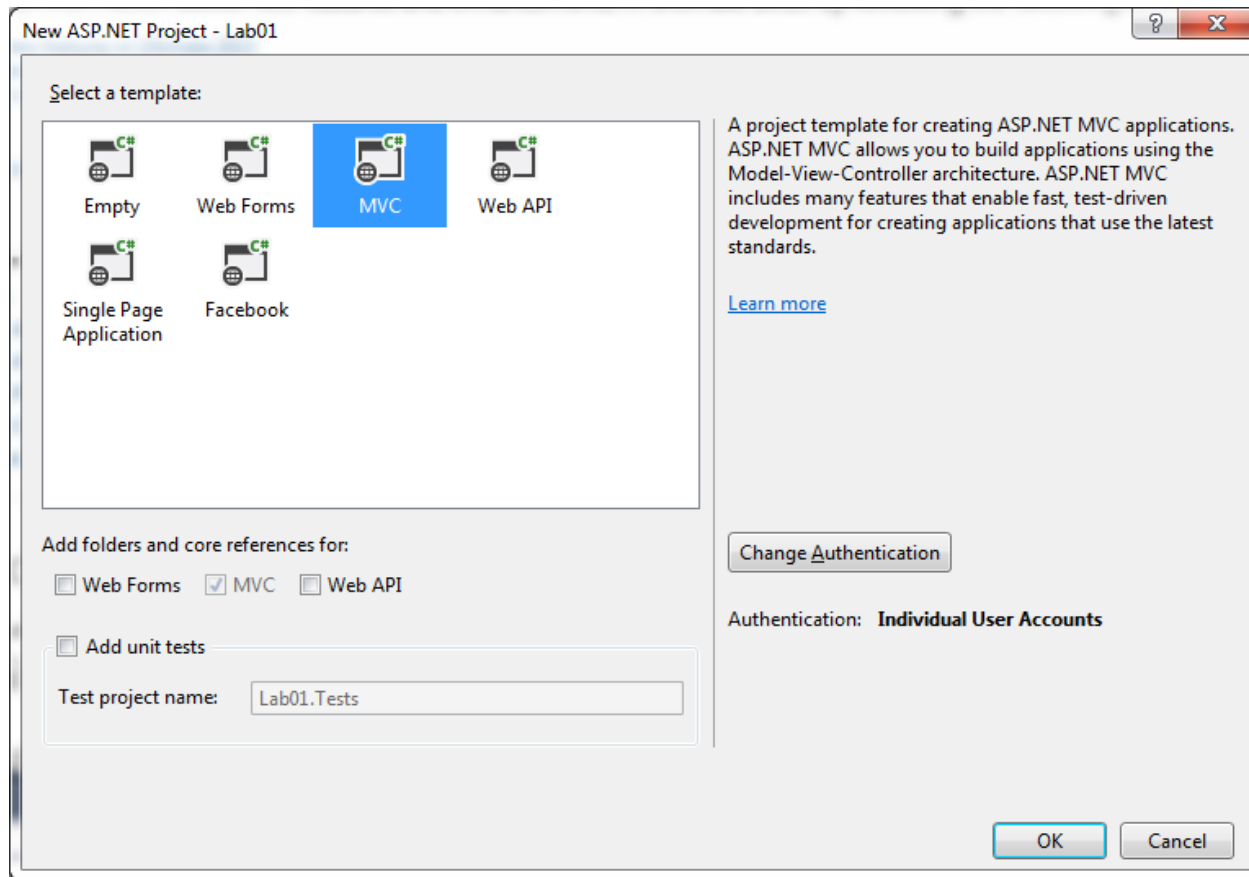
Trong Visual Studio 2013: *File* → *new Project*





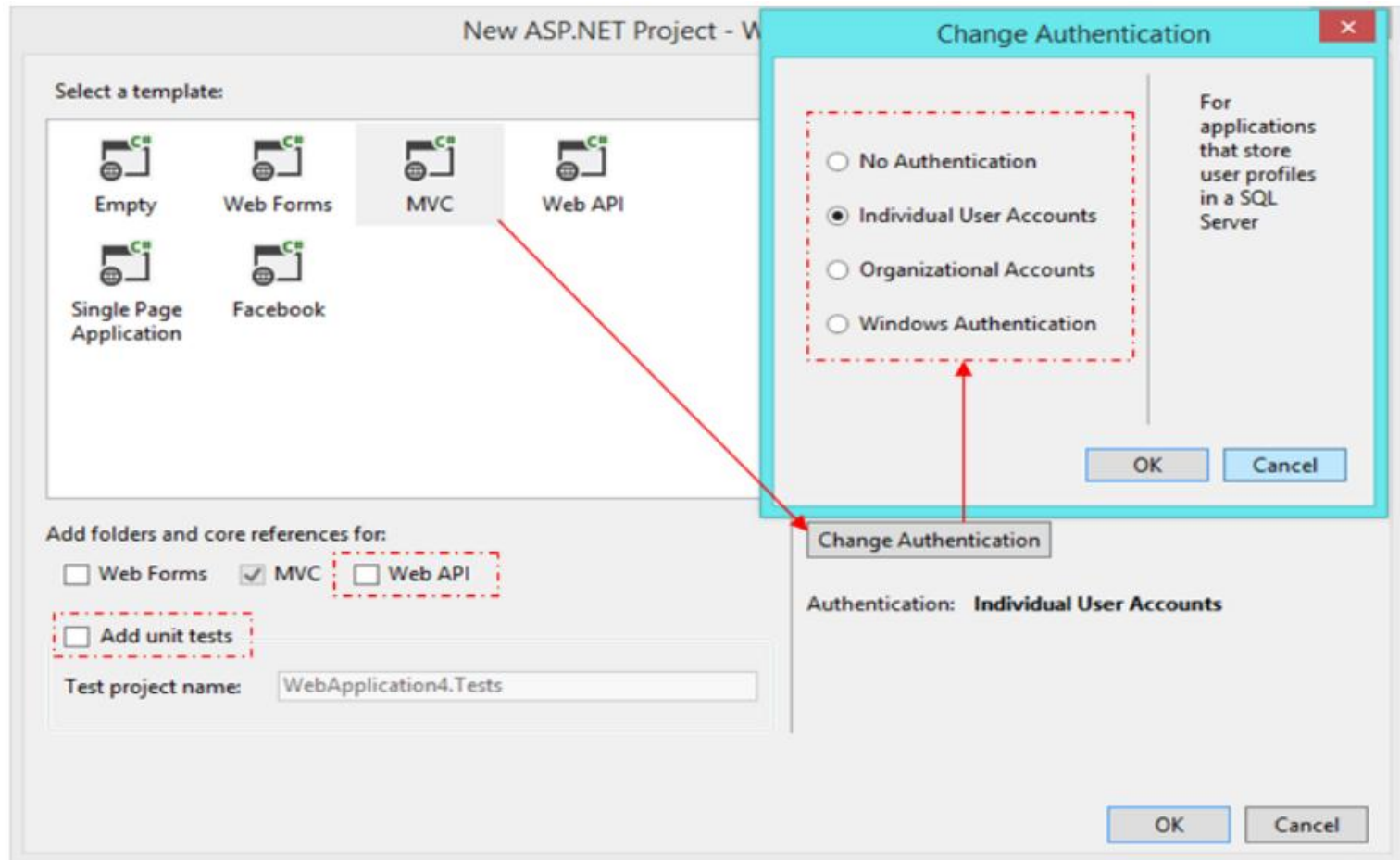
## 2. XÂY DỰNG ỨNG DỤNG WEB

- Chọn mô hình phát triển Web



## 2. XÂY DỰNG ỨNG DỤNG WEB

- Phân biệt các loại dự án MVC



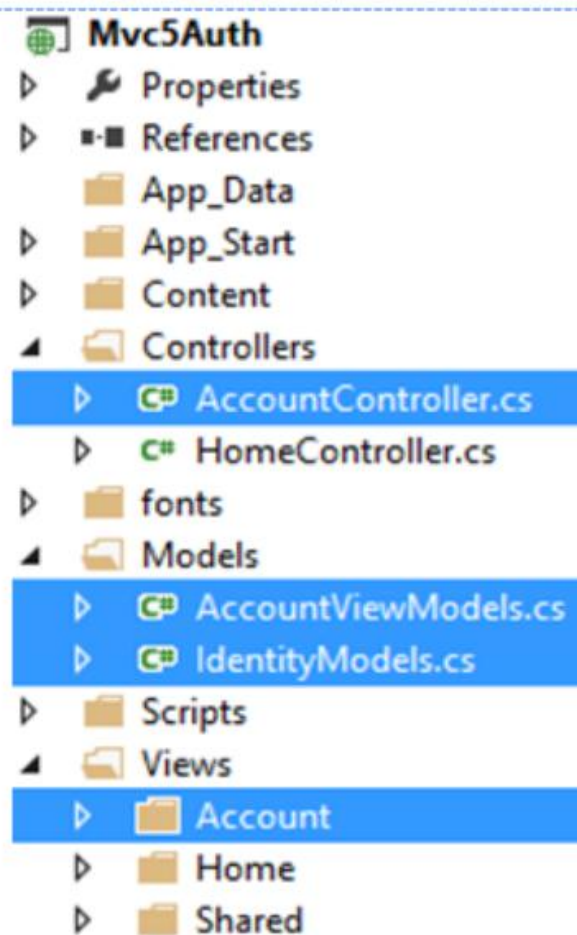
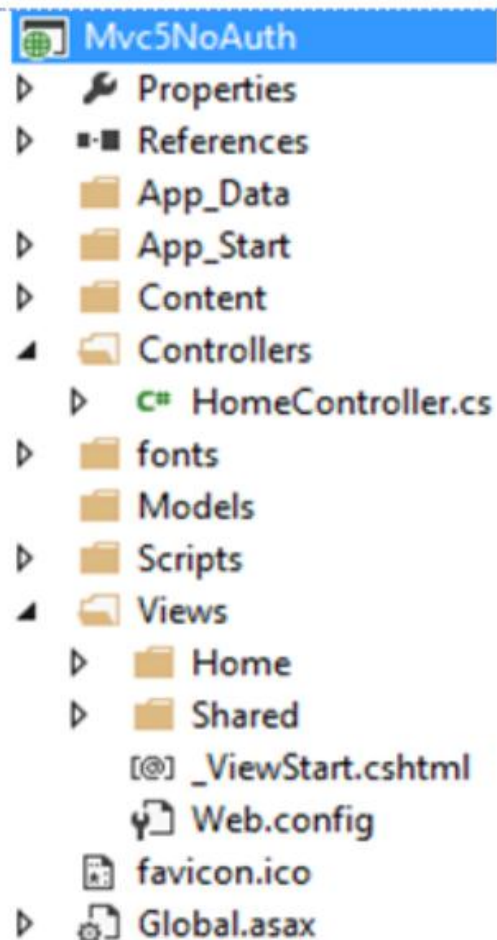
## 2. XÂY DỰNG ỨNG DỤNG WEB

---

- ✓ **Individual User Accounts:** Dự án được tạo ra là loại dự án Internet bao gồm cả phần security
- ✓ **No Authentication:** Dự án được tạo ra là dự án Internet không bao gồm phần security.
- ✓ **Windows Authentication:** Dự án được tạo ra là loại dự án Intranet bao gồm cả phần security nhưng tài khoản được quản lý trên mạng nội bộ.
- ✓ **Organizational Accounts:** Dự án được tạo ra là loại dự án sử dụng tài khoản từ cloud computing.
- ✓ **Ngoài ra:** bạn có thể chọn Web API để bổ sung khả năng hỗ trợ xây dựng thư viện web

## 2. XÂY DỰNG ỨNG DỤNG WEB

### - Cấu trúc của dự án:



## 2. XÂY DỰNG ỨNG DỤNG WEB

---

### - Cấu trúc của dự án:

- ✓ **Controllers:** Thư mục này chứa tất cả controller trong ứng dụng. Các Controller đáp ứng yêu cầu đến từ trình duyệt, quyết định phải làm gì và phản hồi kết quả.
- ✓ **Models:** Các lớp mô hình dữ liệu được tổ chức để nắm giữ và thao tác dữ liệu
- ✓ **Views:** View chứa các mẫu giao diện người dùng của ứng dụng

## 2. XÂY DỰNG ỨNG DỤNG WEB

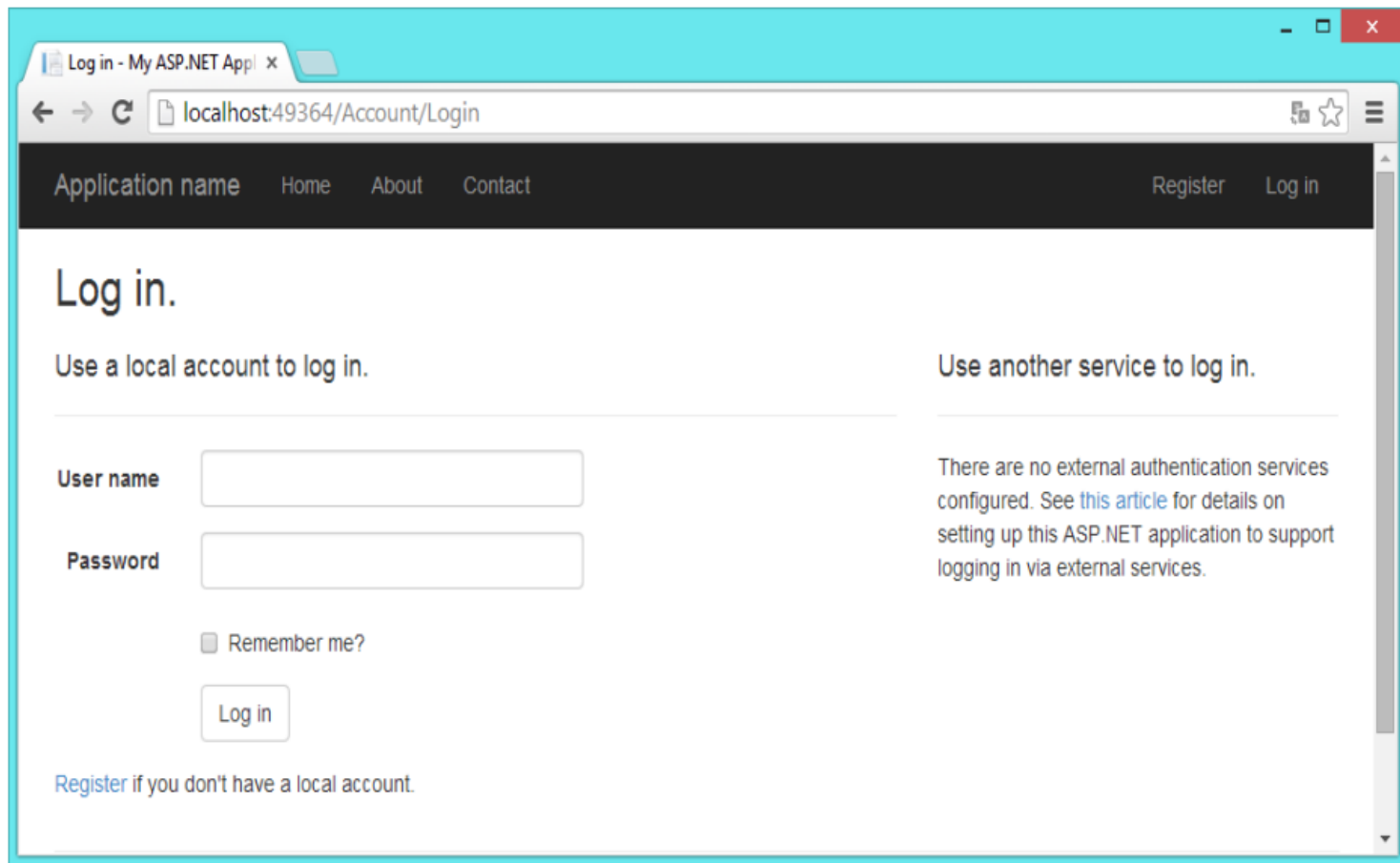
---

### - Cấu trúc của dự án:

- ✓ **App\_Data**: chứa các file dữ liệu cơ sở dữ liệu (nếu có)
- ✓ **App\_Start**: Chứa lớp được chạy một lần khi ứng dụng bắt đầu.
- ✓ **Content**: Chứa tài nguyên tĩnh như hình ảnh, CSS...
- ✓ **Scripts**: Chứa JavaScript cần thiết của ứng dụng
- ✓ **Global.asax**: Tập tin sự kiện chứa các điều khiển sự kiện nhằm kiểm soát vòng đời của Request, Session, Application...
- ✓ **Web.config**: tập tin xml chứa thông tin cấu hình của ứng dụng web

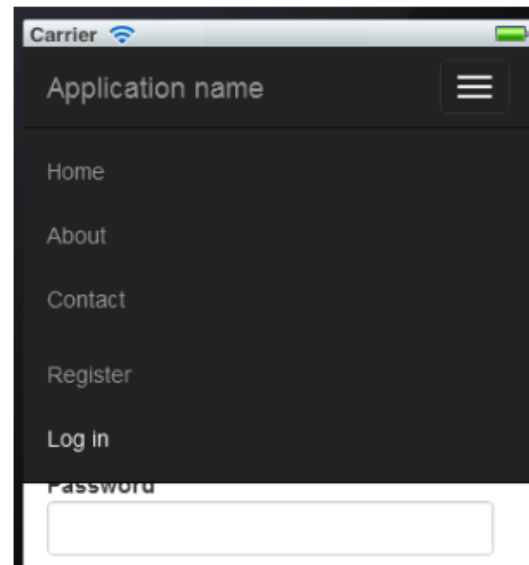
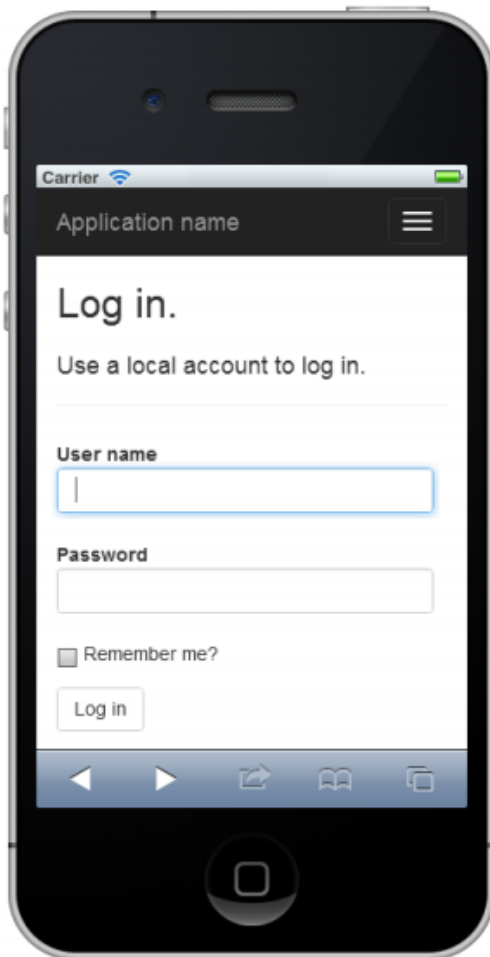
## 2. XÂY DỰNG ỨNG DỤNG WEB

- Nhấn F5 để thực thi dự án



Thực thi trên Desktop

## 2. XÂY DỰNG ỨNG DỤNG WEB

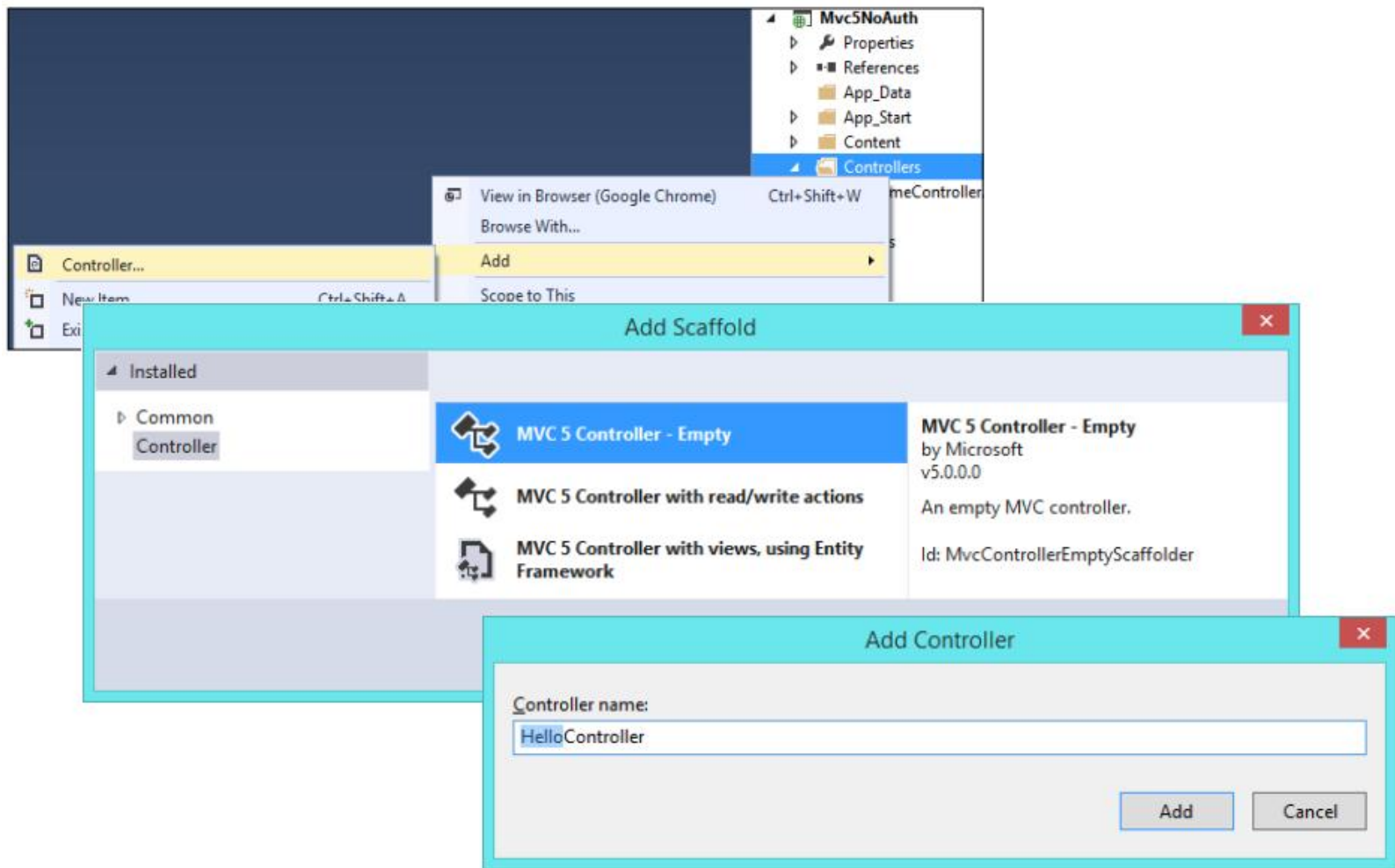


Thực thi thiết bị di động



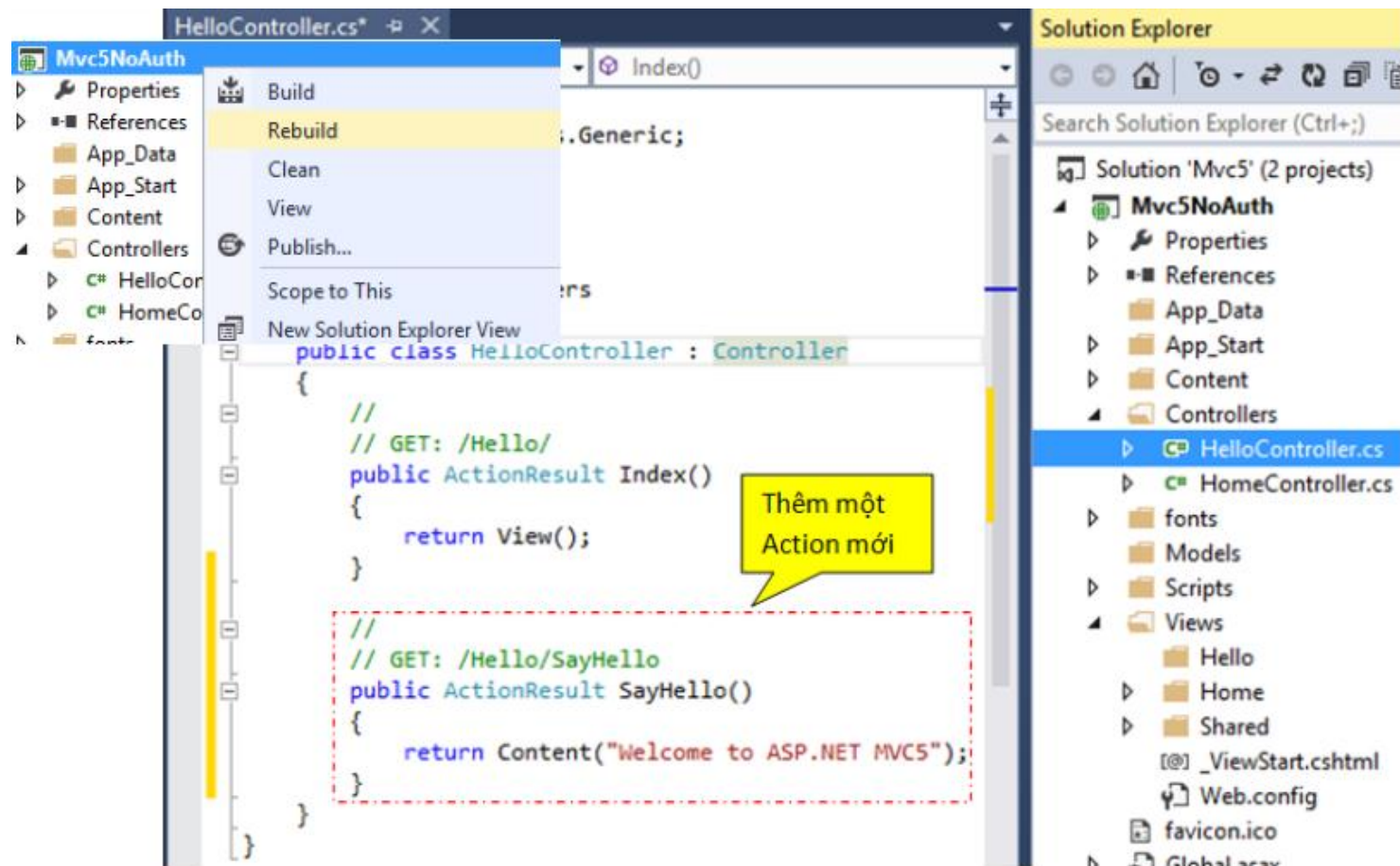
# 3. GIỚI THIỆU VỀ CONTROLLER

## Tạo mới controller



### 3. GIỚI THIỆU VỀ CONTROLLER

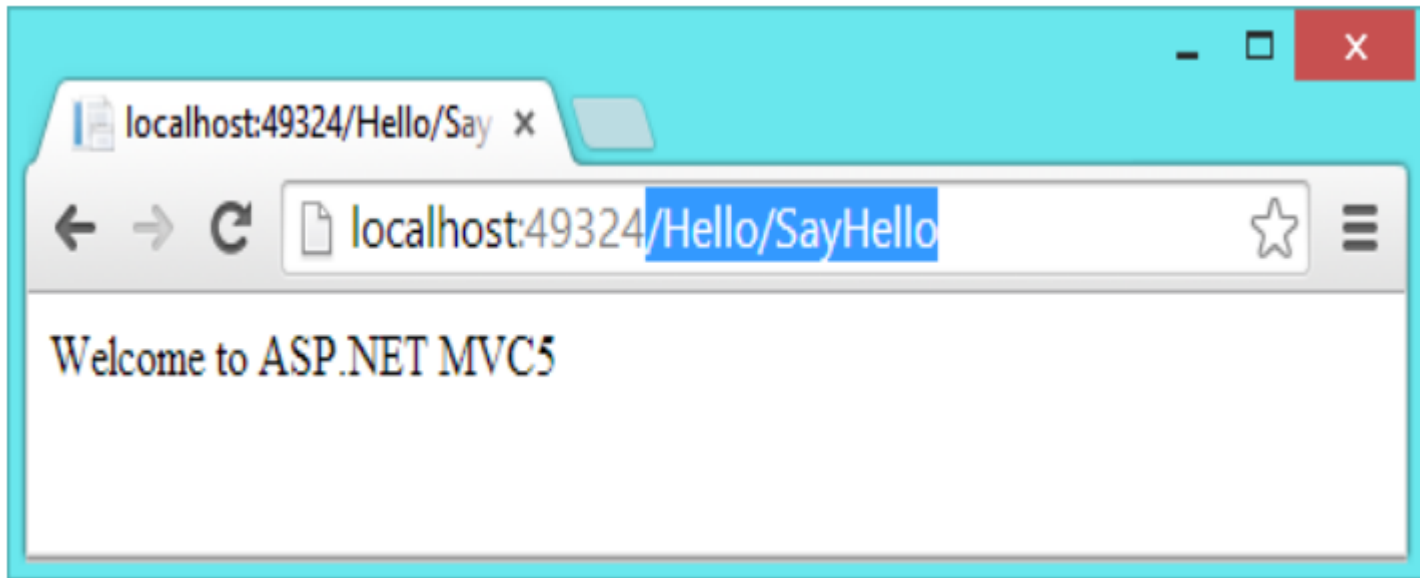
#### Thêm Action cho Controller



### 3. GIỚI THIỆU VỀ CONTROLLER

---

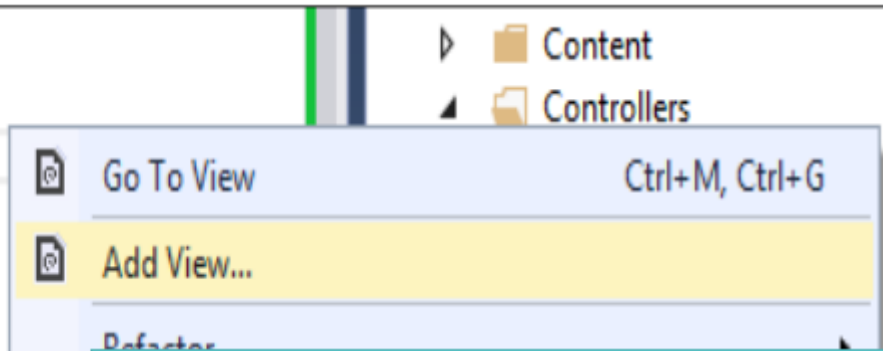
**Thực thi xem kết quả:**



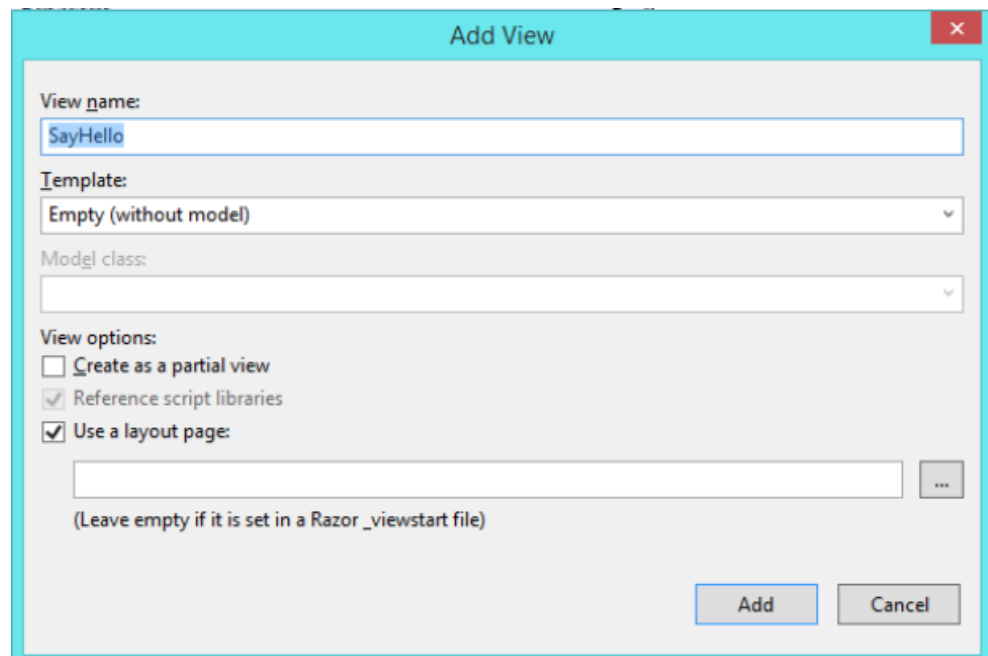
## 4. GIỚI THIỆU VỀ VIEW

### Tạo View cho Action

```
//  
// GET: /Hello/SayHello  
public ActionResult SayHello()  
{  
    return View();  
}
```



Chú ý: **return View()**



View name: SayHello

Template: Empty (without model)

Model class:

View options:

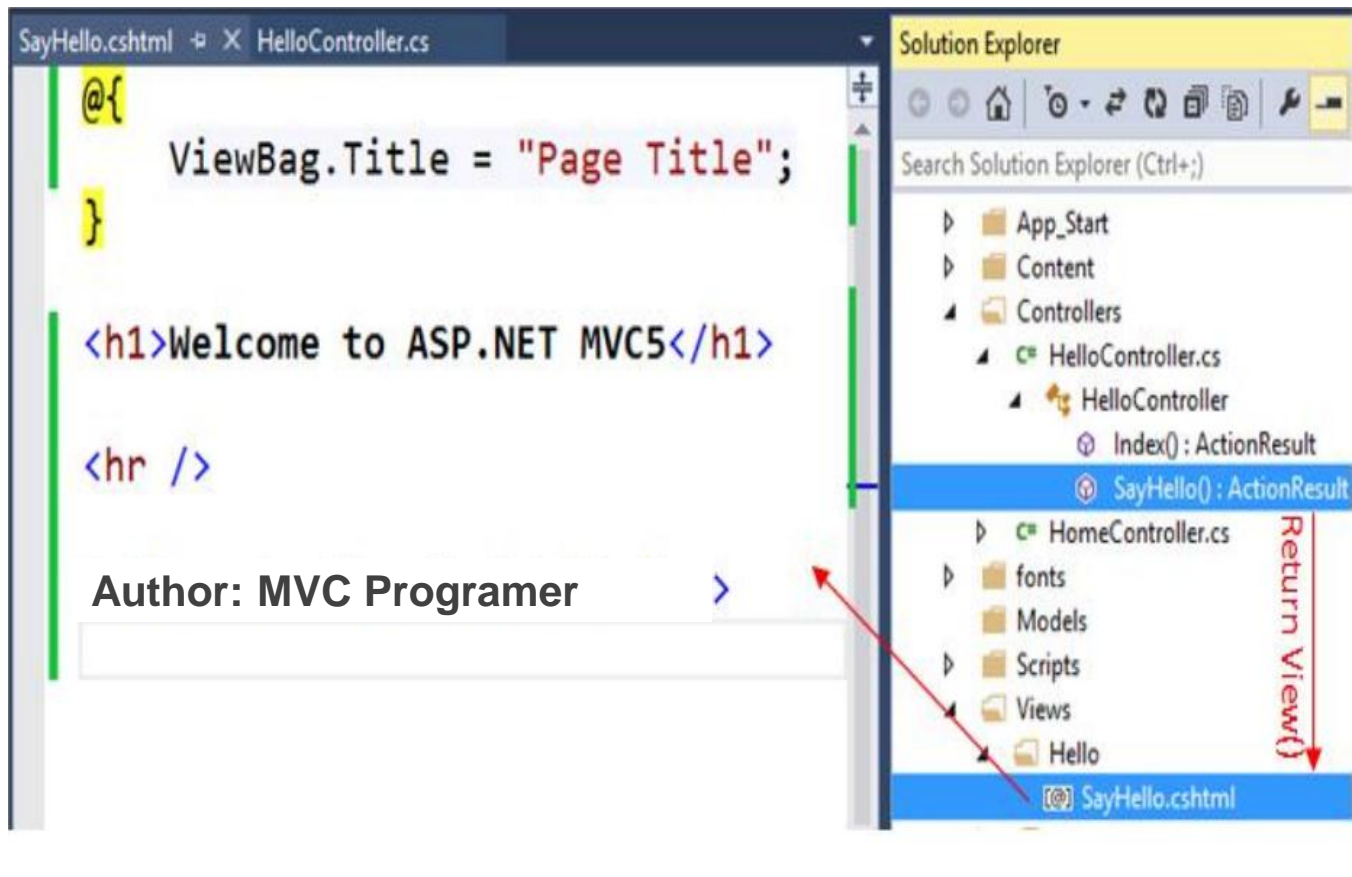
- ☐ Create as a partial view
- ☒ Reference script libraries
- ☒ Use a layout page:

(Leave empty if it is set in a Razor \_viewstart file)

Add Cancel

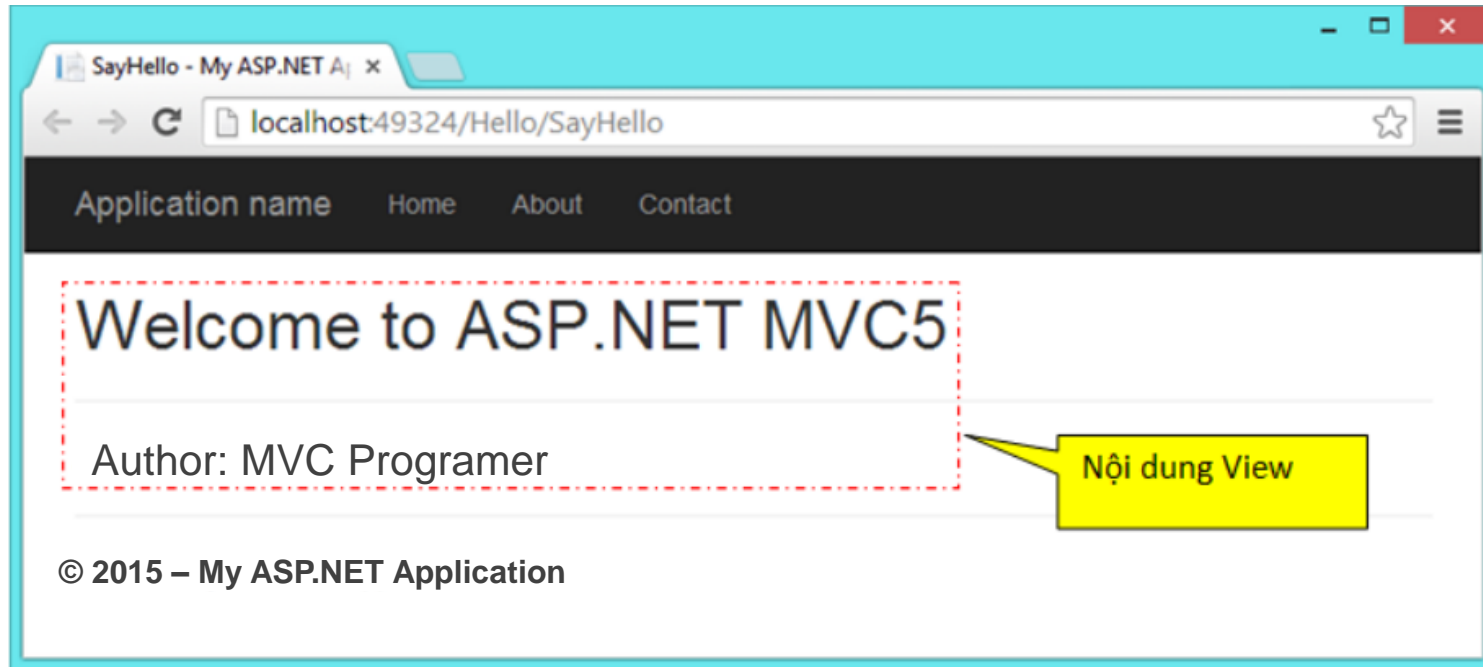
## 4. GIỚI THIỆU VỀ VIEW

### Code Razar cho View



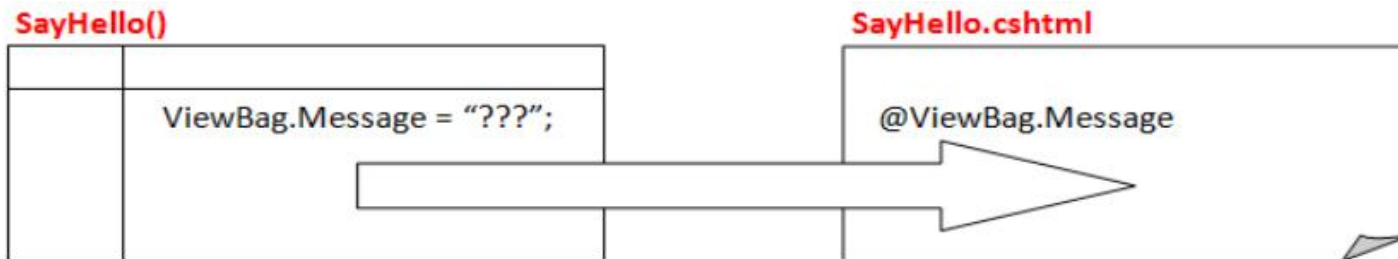
## 4. GIỚI THIỆU VỀ VIEW

### Kết quả thực thi Action



## 4. GIỚI THIỆU VỀ VIEW

### Truyền dữ liệu từ Controller sang View đơn giản



```
public ActionResult SayHello()  
{  
    ViewBag.Message = "Welcome to ASP.NET MVC5";  
    return View();  
}
```

The screenshot shows a code editor with two tabs: **SayHello.cshtml** and **HelloController.cs**. The **SayHello.cshtml** file contains the following code:

```
@{  
    ViewBag.Title = "Page Title";  
}  
  
<h1>@ViewBag.Message</h1>  
  
<hr />  
  
Author: MVC Programmer
```

A red dashed box highlights the `<h1>@ViewBag.Message</h1>` line, and a blue arrow points from the `return View();` line in the controller code to this line in the view.

## 5. GIỚI THIỆU VỀ ĐỊNH TUYẾN URL

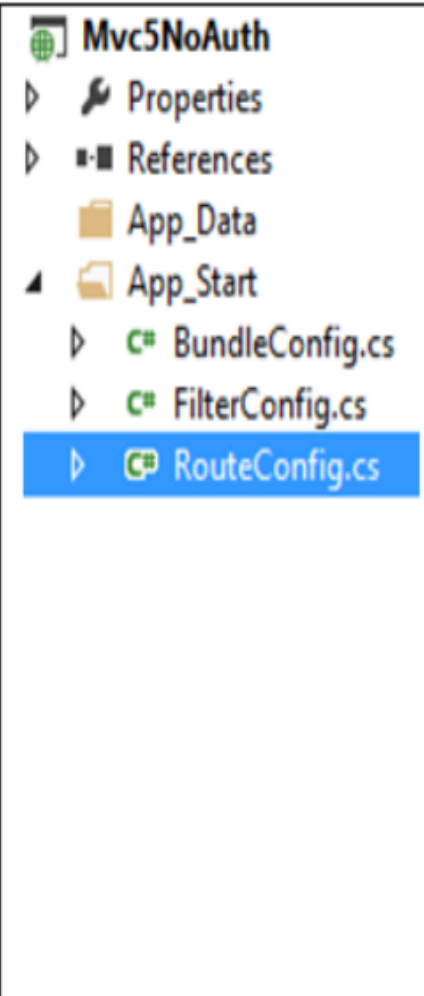
---

### 5.1. Định tuyến là gì?

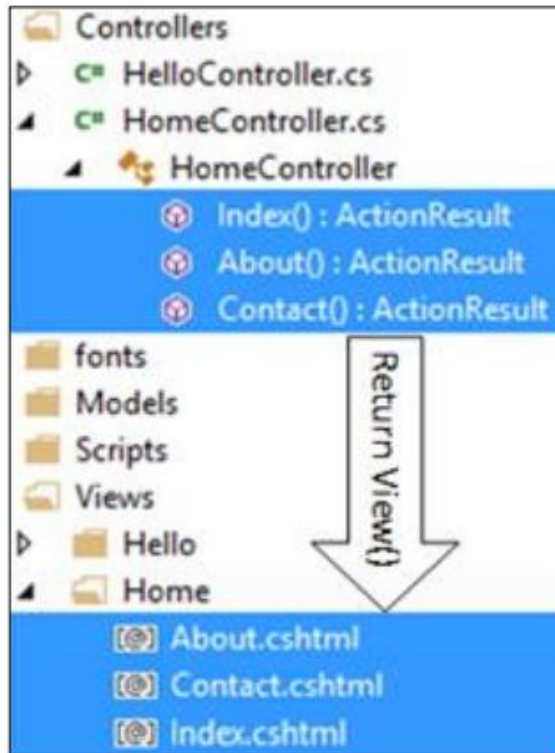
- ✓ Browser yêu cầu một địa chỉ từ controller action được gọi là định tuyến URL(URL routing).
- ✓ Url routing sẽ chỉ định request tới controller action.
- ✓ URL routing sử dụng một bảng định tuyến để điều khiển các request.
- ✓ Bảng định tuyến được tạo khi ứng dụng bắt đầu được chạy lần đầu tiên, được thiết lập trong file *App\_Start\RouteConfig.cs*



## 5. GIỚI THIỆU VỀ ĐỊNH TUYẾN URL

|   |   |
|---|---|
|  <p>Mvc5NoAuth</p> <ul style="list-style-type: none"><li>Properties</li><li>References</li><li>App_Data</li><li>App_Start<ul style="list-style-type: none"><li>BundleConfig.cs</li><li>FilterConfig.cs</li><li>RouteConfig.cs</li></ul></li></ul> | <p>Việc cấu hình này được thực hiện trong file RouteConfig.cs đặt tại thư mục App_Start. Sau đây là nội dung của file này.</p> <pre>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             }         );     } }</pre> |
|---|---|

## 5. GIỚI THIỆU VỀ ĐỊNH TUYẾN URL



Gọi action Index() của HomeController:

- ✓ <http://localhost:49324/Home/Index/2000>
- ✓ <http://localhost:49324/Home/Index>
- ✓ <http://localhost:49324/Home>
- ✓ <http://localhost:49324>

Gọi action About() của HomeController:

- ✓ <http://localhost:49324/Home/About/2000>
- ✓ <http://localhost:49324/Home/About>

Gọi action Index() của HelloController:

- ✓ <http://localhost:49324/Hello/Index/2000>
- ✓ <http://localhost:49324/Hello/Index>
- ✓ <http://localhost:49324/Hello>

## 6. TÌM HIỂU VỀ MODELS

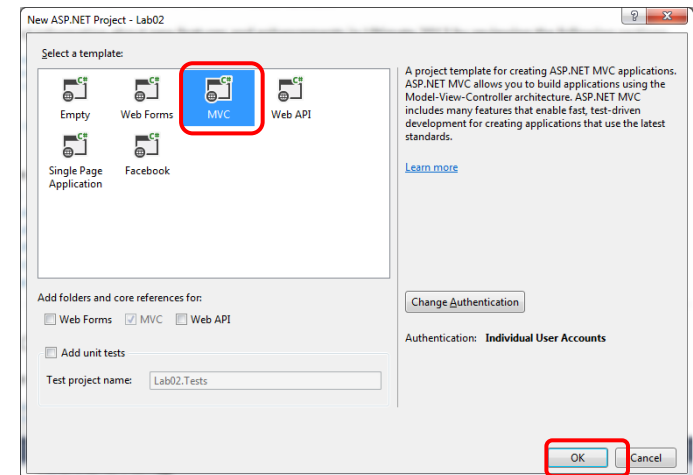
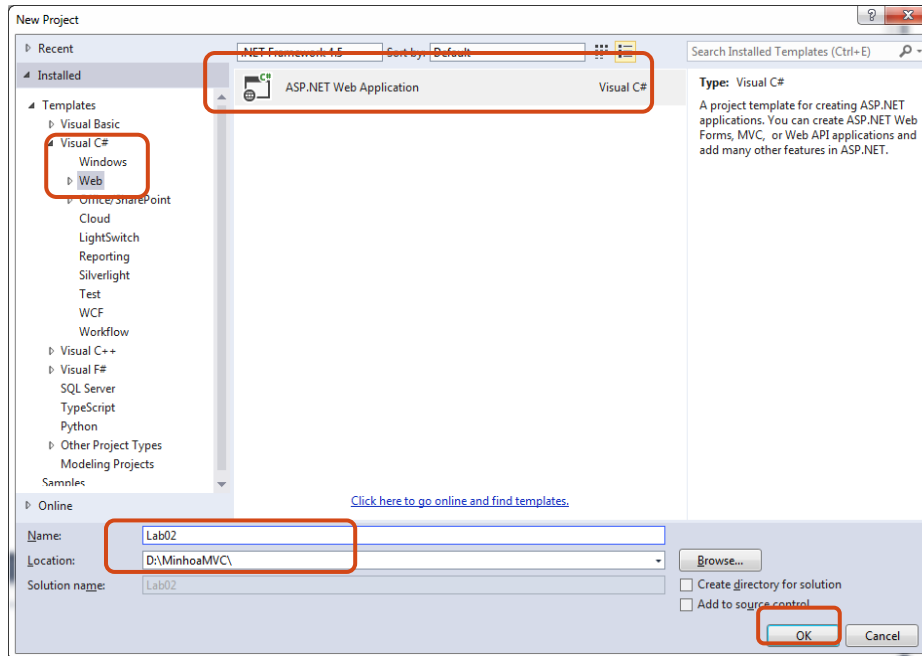
---

- ✓ Một model trong ứng dụng ASP.NET MVC chứa tất cả các nghiệp vụ logic mà không có trong controllers và views.
- ✓ Models chứa tất cả các tầng truy xuất dữ liệu logic và tầng nghiệp vụ logic.
- ✓ Ví dụ, nếu sử dụng LINQ to SQL để truy nhập dữ liệu thì phải tạo LINQ to SQL class ( file định dạng dbml ) trong folder Models.

# 7. XÂY DỰNG ỨNG DỤNG WEB MVC VỚI CSDL

## Bước 1: Tạo mới Project

- ✓ Mở Visual Studio 2013 chọn File → New → Project để tạo 1 ứng dụng mới.
- ✓ Chọn loại Project Visual C# → Web → ASP.Net web Application
- ✓ →Name: <Tên project>



# 7. XÂY DỰNG ỨNG DỤNG WEB MVC VỚI CSDL

## Bước 2: Tạo CSDL

Trong SQLServer tạo CSDL Tintuc gồm 2 bảng và lưu file lệnh vào thư mục Database\CSDLTintuc.sql:

- ✓ Theloaitin(IDLoai int, Tentheloai nvarchar(100) )
- ✓ Tintuc(IdTin int, IDLoai int, Tieudetin nvarchar(100), Noidungtin nText )

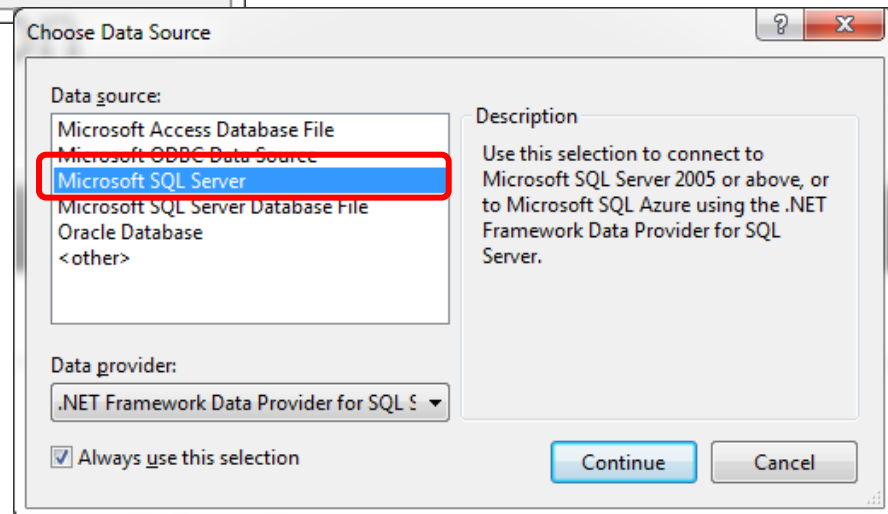
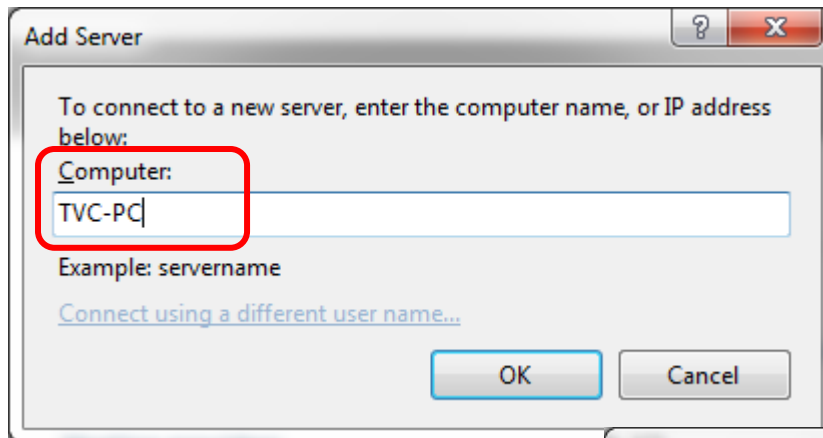
| IDLoai | Tentheloai |
|--------|------------|
| 1      | Thể thao   |
| 2      | Chính trị  |
| 3      | Thế giới   |

| IdTin | IDLoai | Tieudetin                          | Noidungtin                                       |
|-------|--------|------------------------------------|--|
| 1     | 2      | Khủng hoảng kinh tế trong năm 2012 | Tình hình khủng hoảng kinh tế năm 2012 được...   |
| 2     | 1      | Tranh chấp trên biển đông          | lân sự đang nóng dẫn về tình hình biển đông .  . |

# 7. XÂY DỰNG ỨNG DỤNG WEB MVC VỚI CSDL

## Bước 3: Kết nối CSDL với ứng dụng mVC 5

Chọn Menu Tools → Connect to Server => Nhập Server Name



## 7. XÂY DỰNG ỨNG DỤNG WEB MVC VỚI CSDL

Nhập Servername và chọn Database cần kết nối (Tintuc) (username, password nếu có). Chọn Test Connection nếu cần kiểm tra.

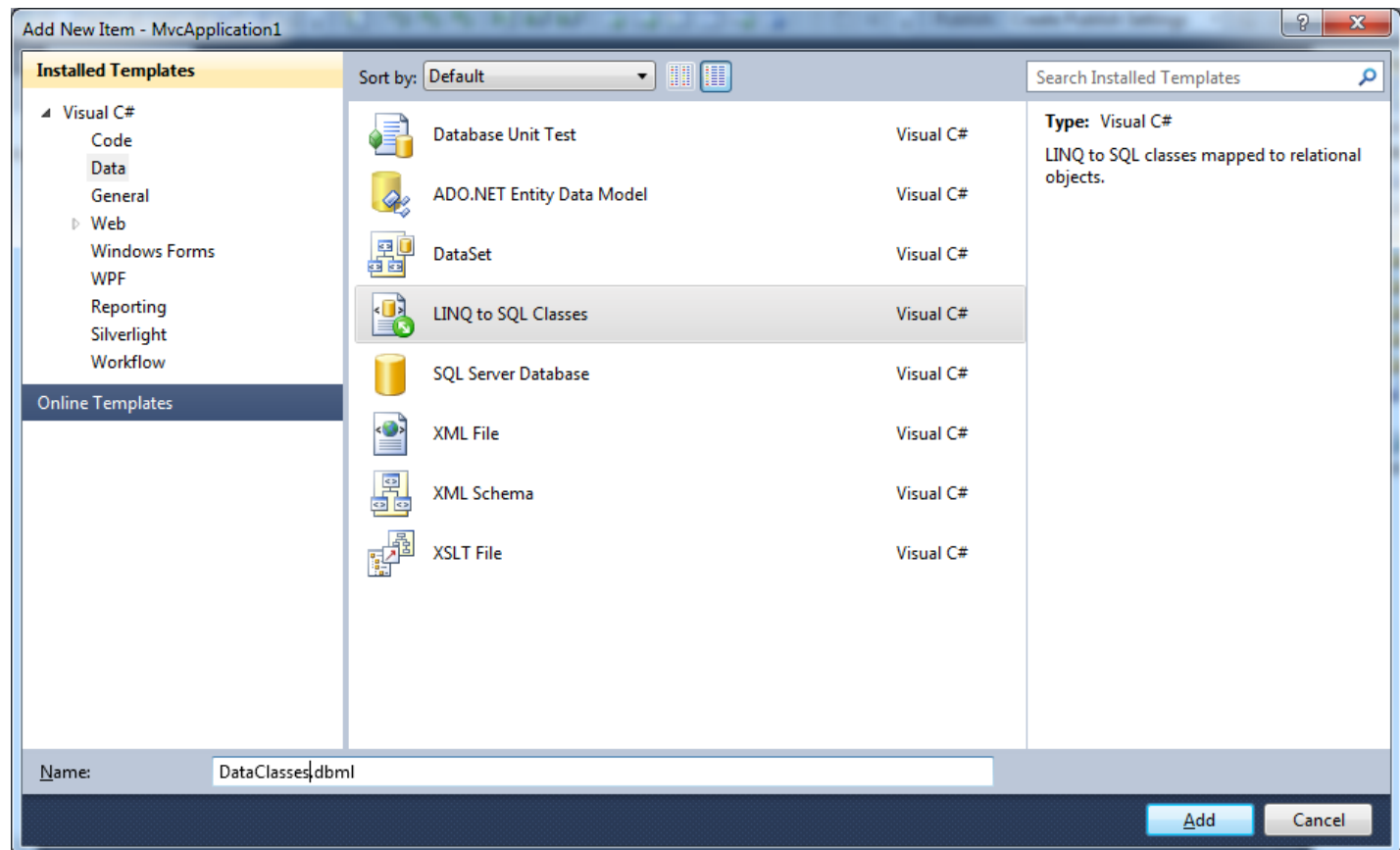
The screenshot shows the 'Add Connection' dialog box with the following configuration:

- Data source:** Microsoft SQL Server (SqlClient)
- Server name:** TVC-PC
- Log on to the server:** Use Windows Authentication (selected)
- Connect to a database:** Select or enter a database name (selected), with 'Tintuc' entered in the dropdown.
- Buttons:** Test Connection, OK, and Cancel are visible at the bottom.

# 7. XÂY DỰNG ỨNG DỤNG WEB MVC VỚI CSDL

## ✓ Sử dụng mô hình LINQ to SQL tạo Models

Trong Solution Explorer click phải → Models → Add → New Item → LINQ to SQL Classes

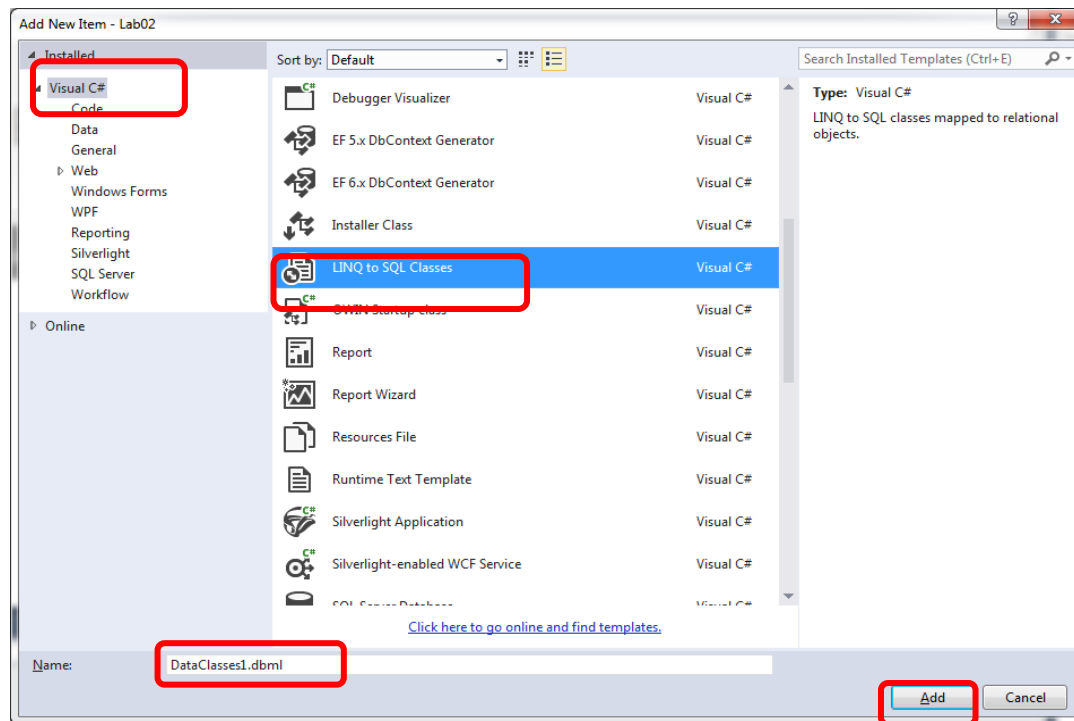




# 7. XÂY DỰNG ỨNG DỤNG WEB MVC VỚI CSDL

## Bước 4: Sử dụng mô hình LINQ to SQL tạo Models

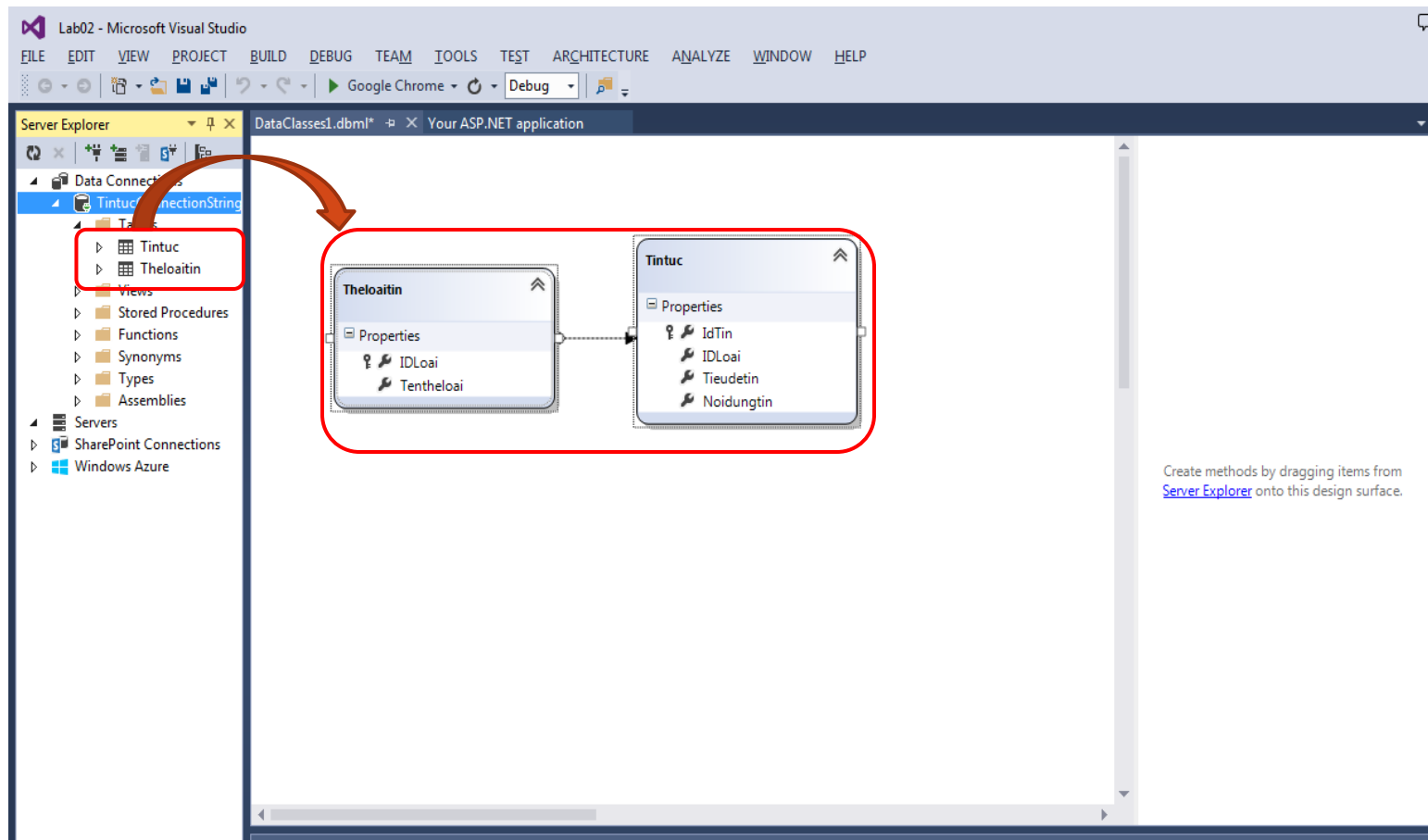
Click phải chuột vào folder Models => add=> chọn Class



Chọn LINQ to SQL classes ( Vì truy xuất dữ liệu bằng LINQ). Đặt tên cho file Class là DataClasses1.dbml .

# 7. XÂY DỰNG ỨNG DỤNG WEB MVC VỚI CSDL

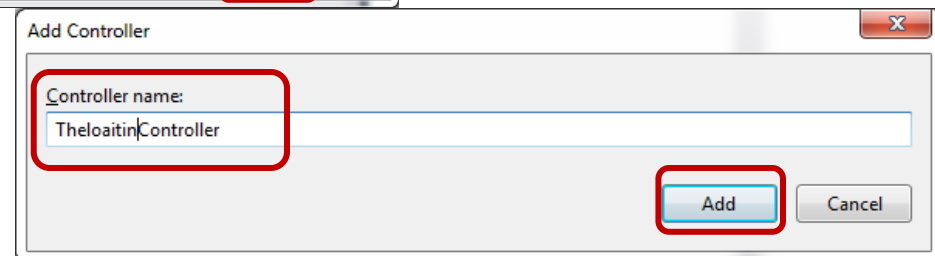
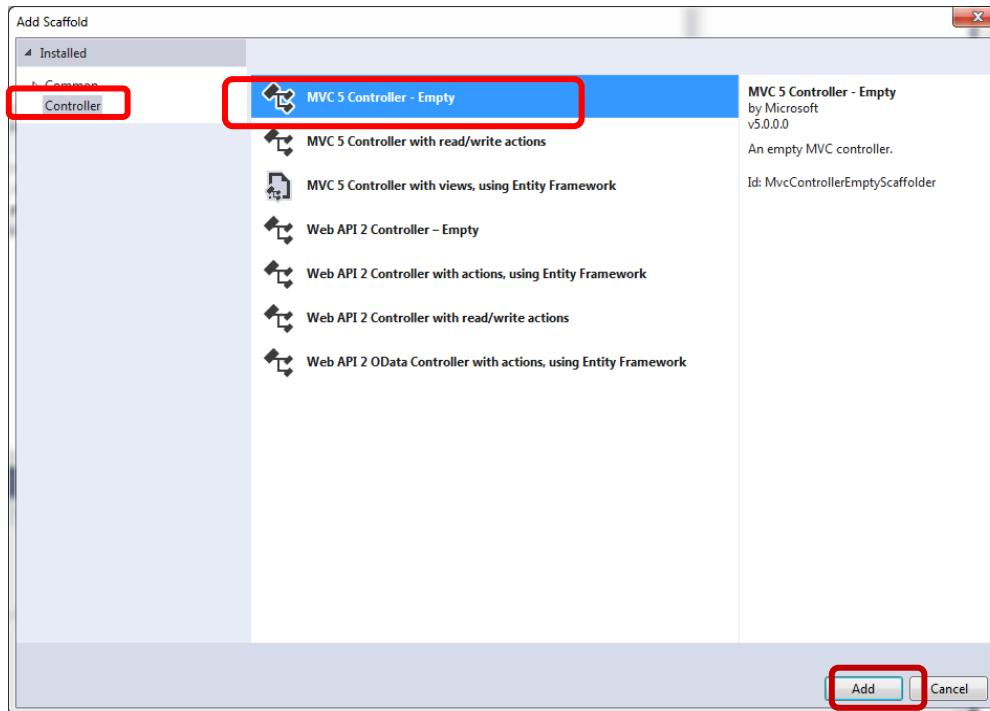
Trên thanh Server Explorer kéo thả các bảng của Database sang trang Dataclasses.dbml



# 7. XÂY DỰNG ỨNG DỤNG WEB MVC VỚI CSDL

## Bước 5: Xử lý Controller

Tại Folder controller click chuột phải chọn Add → Controller



## 7. XÂY DỰNG ỨNG DỤNG WEB MVC VỚI CSDL

Bổ sung Code vào file TheloaitinControllers.cs như sau :

```
using System;|
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Lab02.Models;

namespace Lab02.Controllers
{
    public class TheloaitinController : Controller
    {
        //
        // GET: /TheLoaiTinTuc/
        DataClasses1DataContext data = new DataClasses1DataContext();
        public ActionResult Index()
        {
            var All_Loaitin = from tt in data.TheLoaitins select tt;
            return View(All_Loaitin);
        }
    }
}
```

Tên Model

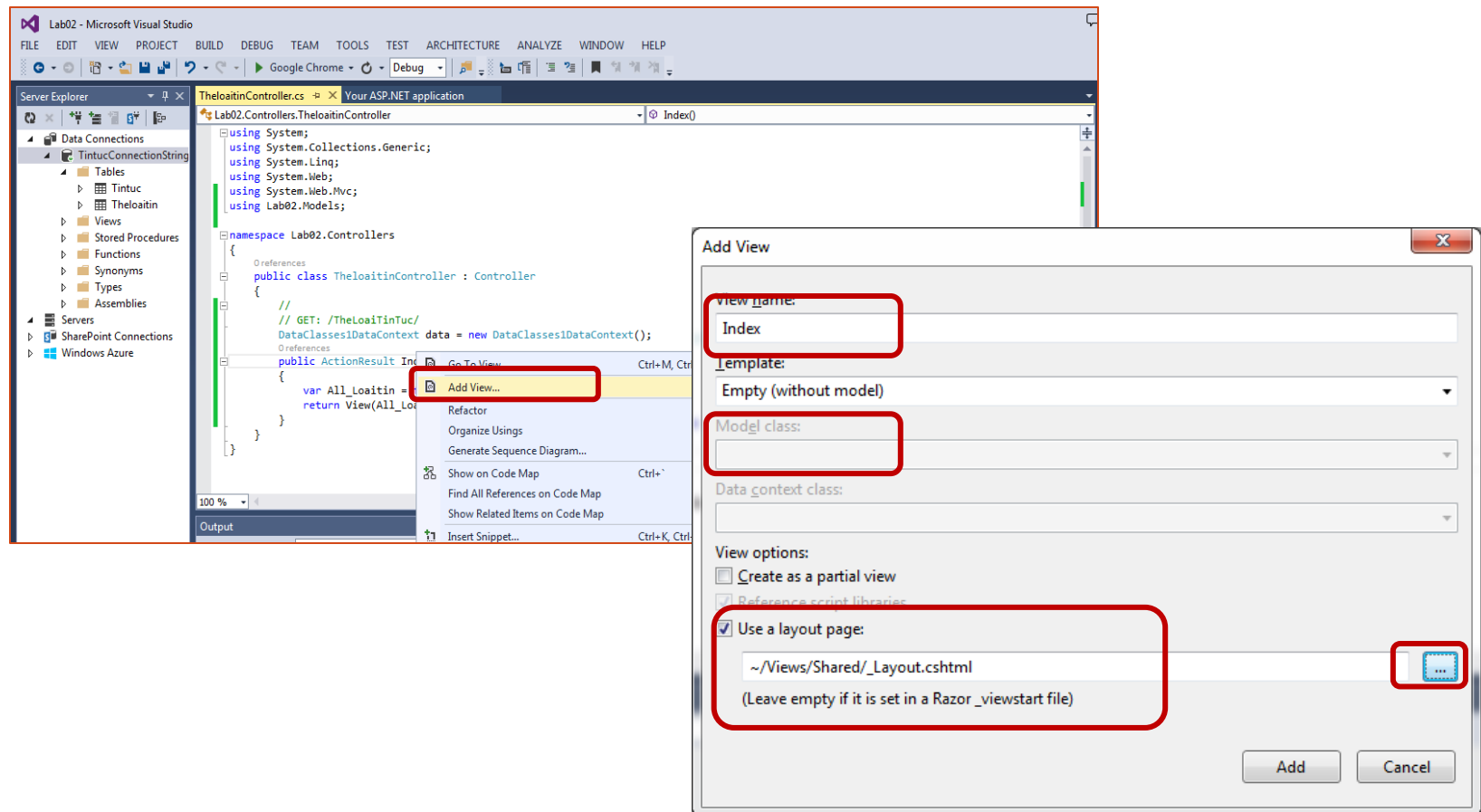
Tên DataTable trong Model

Truy vấn dữ liệu với Linq

# 7. XÂY DỰNG ỨNG DỤNG WEB MVC VỚI CSDL

## Bước 6: Tạo View hiển thị nội dung

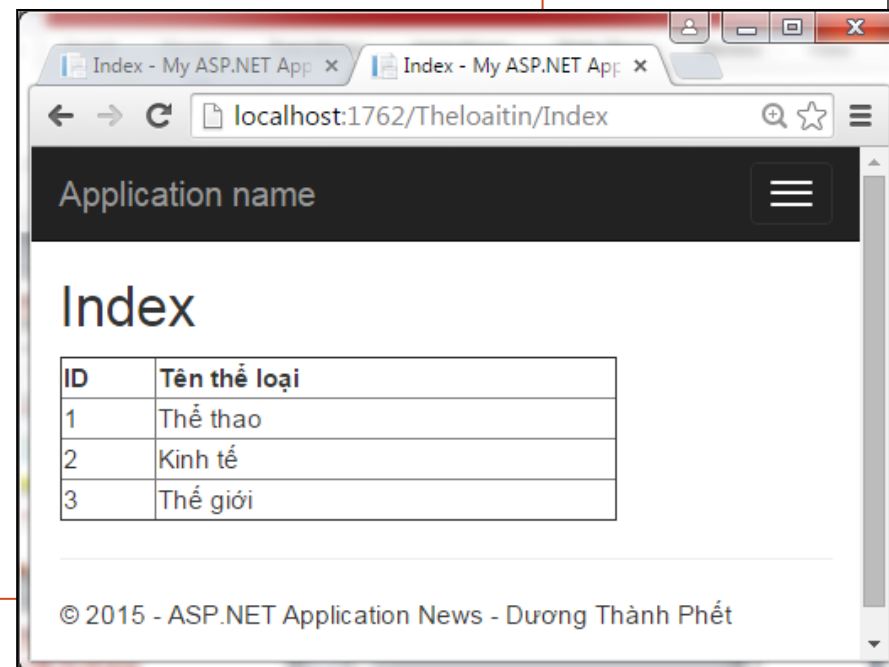
Trong File TheloaitinControllers.cs ở trên ta có hàm: Index()  
Click phải vào tên hàm=> add view...



# 7. XÂY DỰNG ỨNG DỤNG WEB MVC VỚI CSDL

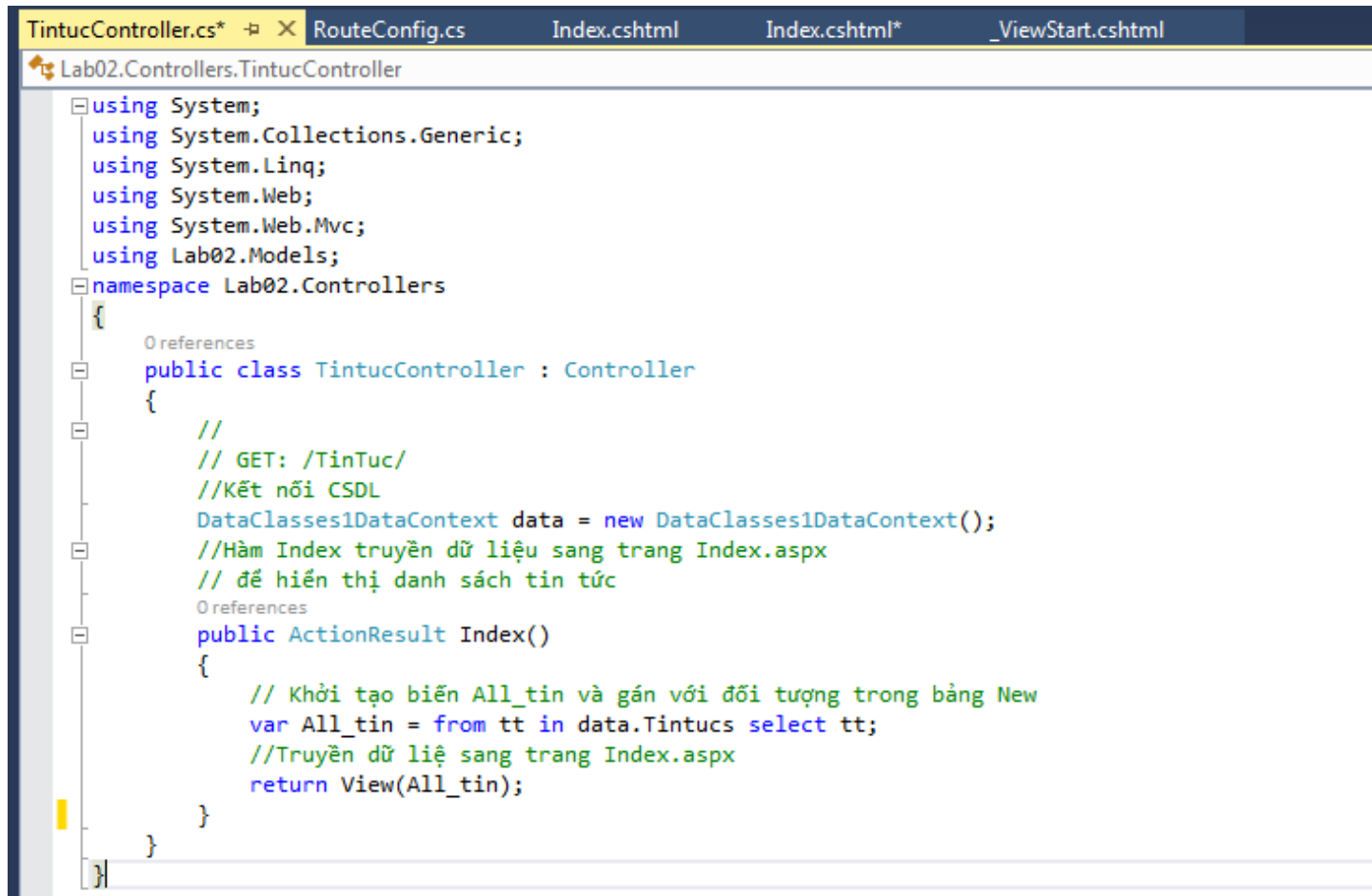
Bổ sung code để hiển thị dữ liệu trong *Index.cshtml* như sau:

```
RouteConfig.cs | Index.cshtml | Index.cshtml | _ViewStart.cshtml
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Index</h2>
<table border="1" width="300">
    <tr>
        <th>
            ID
        </th>
        <th>
            Tên thể loại
        </th>
    </tr>
    @foreach (var item in Model)
    {
        <tr>
            <td>
                <!-- Hiển thị ID của đối tượng với -->
                @item.IDLoai
            </td>
            <td>
                @item.Tentheloai
            </td>
        </tr>
    }
</table>
```



# 7. XÂY DỰNG ỨNG DỤNG WEB MVC VỚI CSDL

*Tương tự cho trang Tin tức: TintucController*



```
TintucController.cs* X RouteConfig.cs Index.cshtml Index.cshtml* _ViewStart.cshtml
Lab02.Controllers.TintucController

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Lab02.Models;

namespace Lab02.Controllers
{
    0 references
    public class TintucController : Controller
    {
        //
        // GET: /TinTuc/
        //Kết nối CSDL
        DataClasses1DataContext data = new DataClasses1DataContext();
        //Hàm Index truyền dữ liệu sang trang Index.aspx
        // để hiển thị danh sách tin tức
        0 references
        public ActionResult Index()
        {
            // Khởi tạo biến All_tin và gán với đối tượng trong bảng New
            var All_tin = from tt in data.Tintucs select tt;
            //Truyền dữ liệu sang trang Index.aspx
            return View(All_tin);
        }
    }
}
```

# 7. XÂY DỰNG ỨNG DỤNG WEB MVC VỚI CSDL

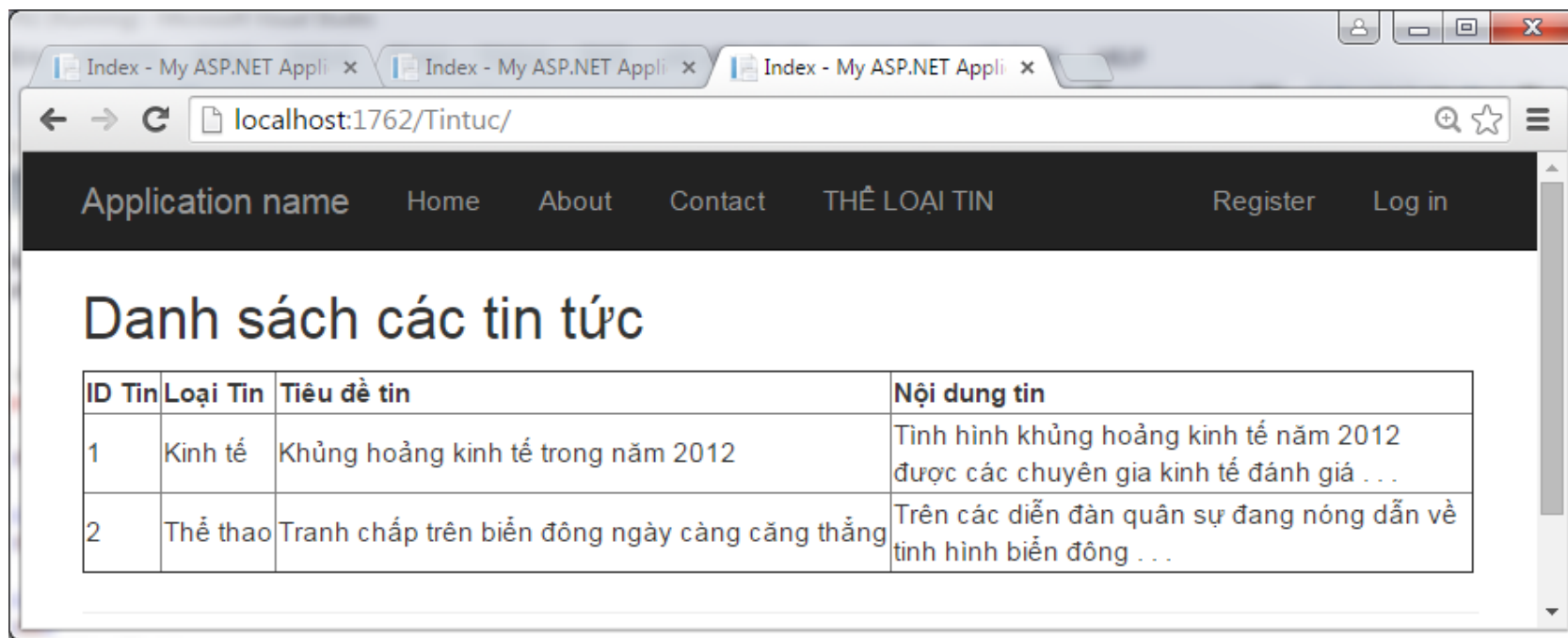
## Tương tự cho trang Tin tức: View Index.cshtml

```
RouteConfig.cs  X Index.cshtml  Index.cshtml*  X _ViewStart.cshtml
D:\Minhhoa\MVC\Lab02\App_Start\RouteConfig.cs
ViewBag.Title = "Index";
Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Danh sách các tin tức</h2>
<table border="1">
  <tr>
    <th> ID Tin</th>
    <th> Loại Tin </th>
    <th> Tiêu đề tin </th>
    <th> Nội dung tin </th>
  </tr>
  @foreach (var item in Model)
  {
    <tr>
      <td> @item.IdTin </td>
      <td> @item.Theloaitin.Tentheloai</td>
      <td> @item.Tieudetin </td>
      <td width=300> @item.Noidungtin </td>
    </tr>
  }
</table>
```



# 7. XÂY DỰNG ỨNG DỤNG WEB MVC VỚI CSDL

*Tương tự cho trang Tin tức: Kết quả View Index.cshtml*



**HẾT**