**HƯỚNG DẪN LẬP TRÌNH WEB VỚI ASP.NET MVC**

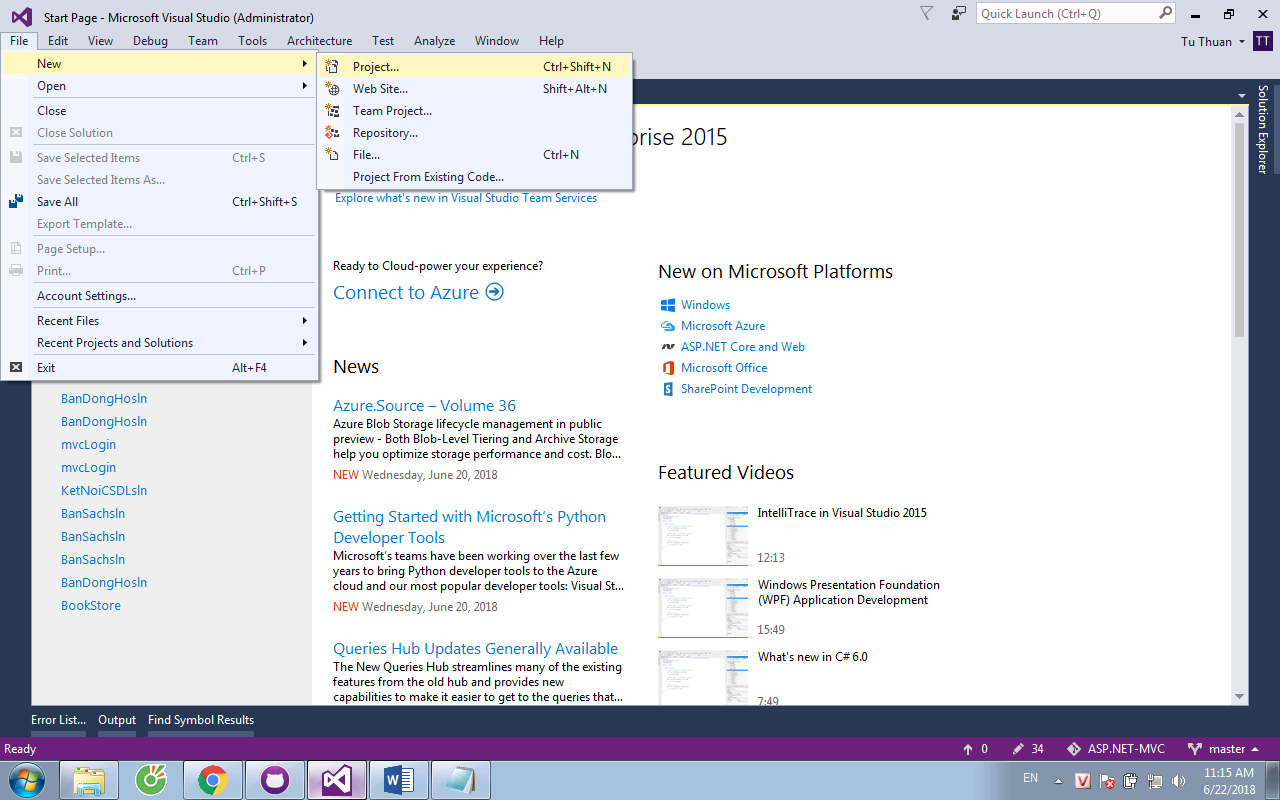
**( Phần 1 - Cơ bản về ASP.Net MVC)**

1. **Các thao tác cơ bản với Visual Studio**

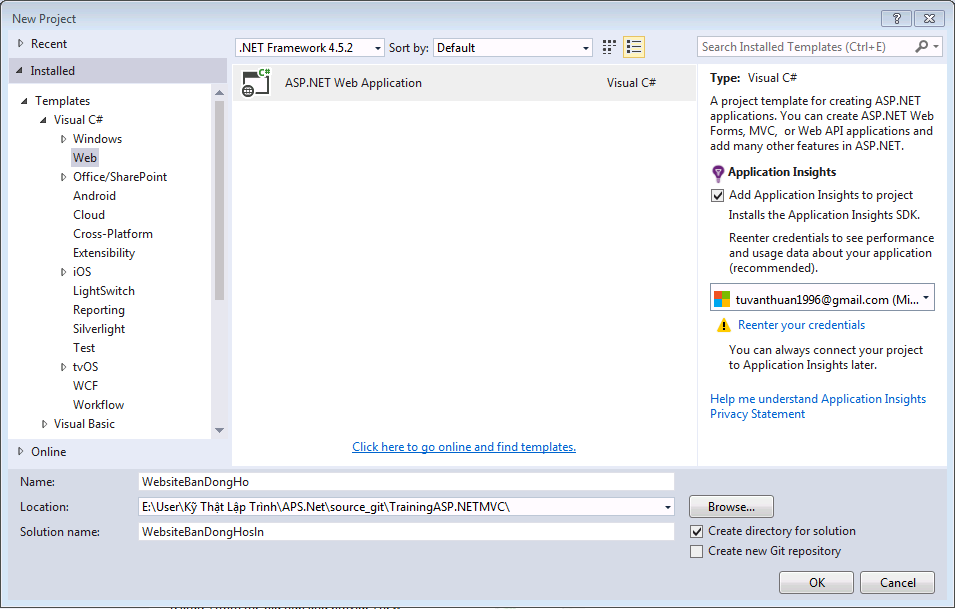
Phần này nêu ra các cách thức thao tác cơ bản với Visual Studio (VS) nếu bạn đã là một master thì có thể bỏ qua phần này.

**1.1. Tạo dự án mới**

Tại menu File chọn *New, Project…*



Màn hình tạo project hiện ra bạn nhập thông tin như sau:



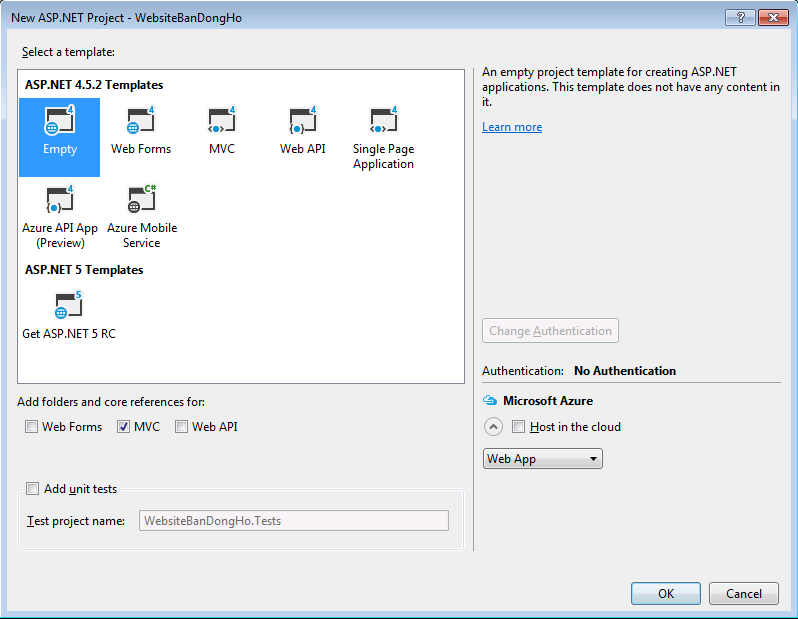
Name: WebsiteBanDongHo – đây là tên dự án.

Location: Địa chỉ local lưu solution trên máy bạn.

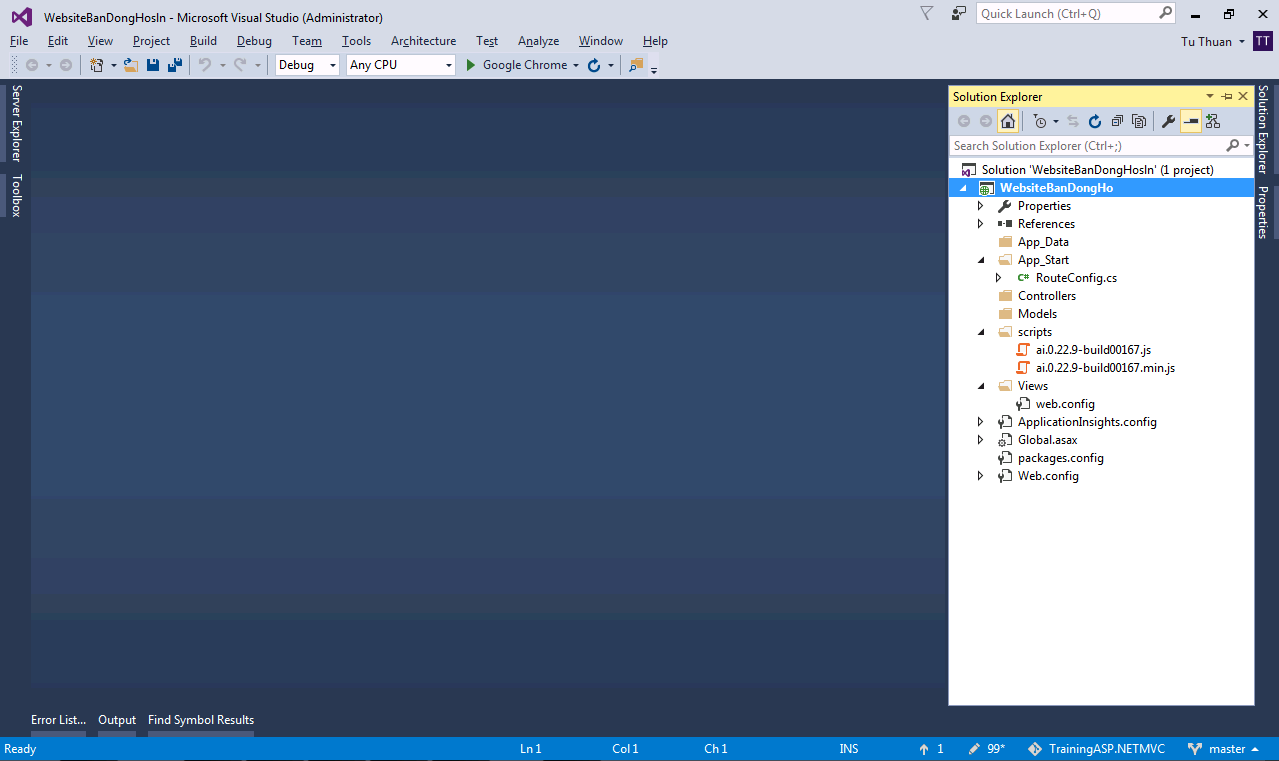
Solution name: Tên của solution.

Chú ý: Ở đây mình tạo dự án web ASP.Net như các bạn thấy.

Nhấn Ok và chọn tiếp Empty, chọn MVC trong mục Add folders and core reference for:



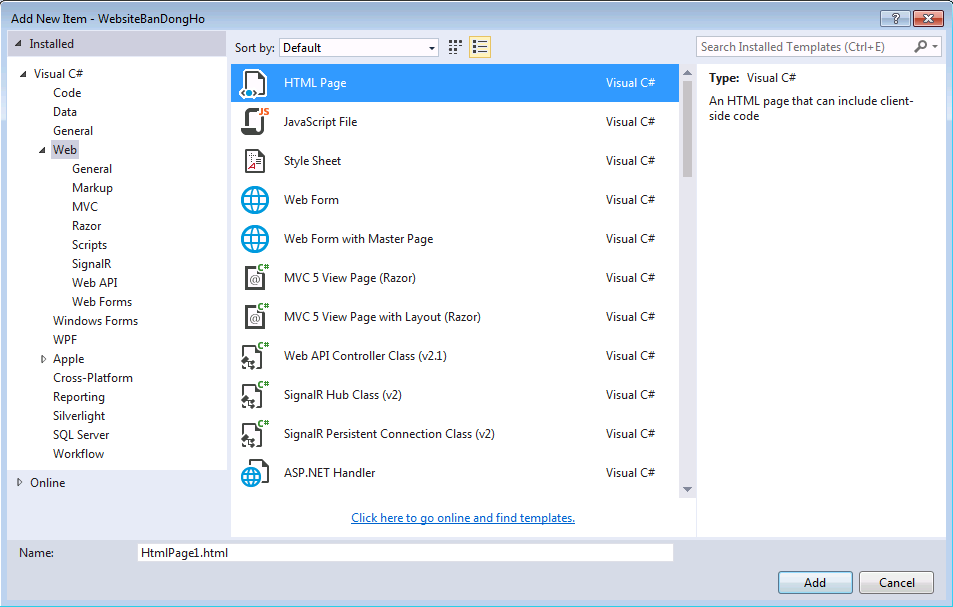
Cuối cùng Visual tạo cho bạn một dự án như sau:



**1.2. Tạo item**

Trong phần này sẽ hướng dẫn các bạn tạo một item (File, model, vv…). Chúng tôi sẽ chỉ hướng dẫn cho các bạn các bước tạo ra một item như thế nào mà thôi. Sau này các bạn cần phải vận dụng các bước này tạo item trong dự án.

Để tạo một Item các bạn chọn vào thư mục cần tạo, chuột phải chọn *Add, New Item,…* Visual cung cấp rất nhiều định dạng Item cho bạn trong dự án lần này chúng ta tập trung vào các mẫu Item sau.



Trong mục Web:

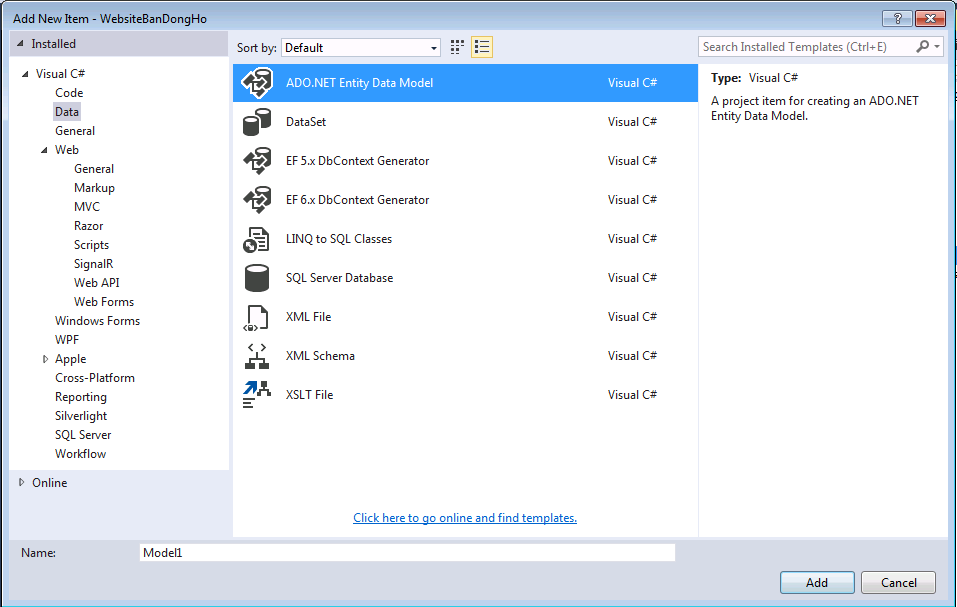
+ HTML Page: tạo ra một file với đôi .html

+ Style Sheet: Tạo ra một file với đuôi .css

+ JavaScript File: Tạo ra một file với đuôi .js

Trong mục Data:

+ ADO.NET Entity Data Model: Tạo ra một model chứa các class tương tác với Cơ sở dữ liệu.



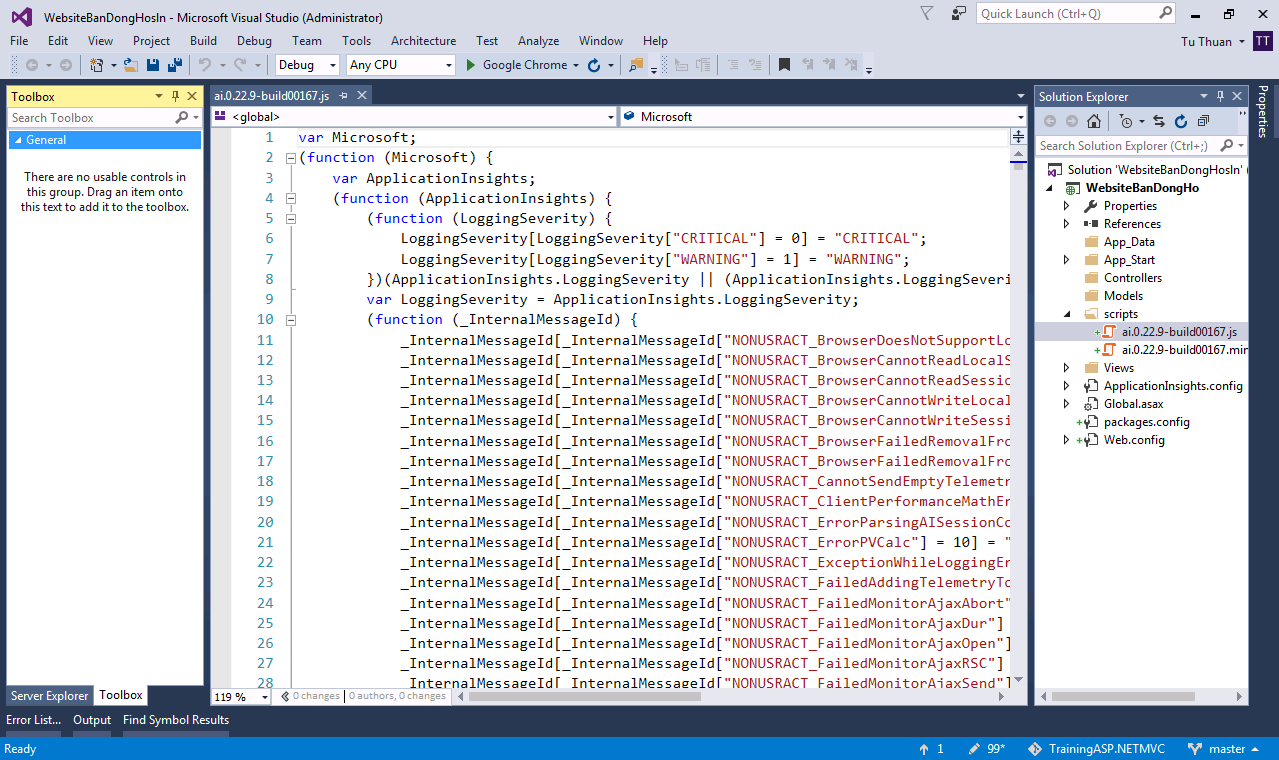
Sau khi tạo file ta tiến hành đặt tên và chọn Add.

**1.3. Tạo thư mục (namespace)**

Trong VS khi tạo mới một thư mục tức là bạn đang tạo ra một namespace mới với tên là tên thư mục. Namespace là một vùng mà tại các Class, funtion hay Object được nhóm lại chủ yếu vì chúng có liên quan với nhau hoặc chúng có cùng chức năng.

Việc tạo mới namespace trong VS khá đơn giản: Chọn vào nơi cần tạo thư mục, chuột phải, *Add, New Folder...* Tự động trong cây thư mục sẽ phát sinh một folder cho bạn.

**1.4. Layout của Visual Studio.**



Trong VS màn hình làm việc được bố cục như sau. Sidebar bên trái là Toolbox hoặc Server Explorer. Sidebar bên phải là Solution Explorer và Property. Vùng ở giửa là khu vực viết mã. Ngoài ra bên dưới các các cửa sổ cần thiết như Error list – Nơi hiển thị các lỗi, Output – nơi hiện các Log, vv…

Để chạy dự án sau khi code xong bạn có thể F5 (Debug) hoặc Ctr+F5 (Run).

Với ứng dụng web bạn có thể chọn trình duyệt mặt định để chạy bằng cách. Chọn vào combobox sau:  sau đó chọn trình duyệt mà bạn muốn chạy mặc định.

* Giờ thì mọi thứ đã sẳn sàn và chúng tôi ( nhóm viết bài lab này ) tin chắc các bạn đã hiểu được các tính năng cơ bản của VS. Trong quá trình thực hiện dự án chúng tôi sẽ tiếp tục giới thiệu đến các bạn các chức năng khác của VS. Phần này được nêu ra để các bạn không phải bở ngở nếu như tôi nói rằng: “ Tạo mới một file css với tên custom.css”. Ok bắt đầu với ASP.NET thôi!

1. **Giới thiệu ASP.Net MVC**

**2.1. Vài điều cần biết về lập trình web**

Có vài điều cần biết để thực hiện dự án này mà các bạn chưa bao giờ lập trình web cần biết.

a. Giao diện web

* HTML ( Ngôn ngữ đánh dấu siêu văn bản - HyperText Markup Language) được sử dụng để viết nội dung một trang web.
* CSS (Cascading Style Sheets) ngôn ngữ dùng để định dạng các phần tử (thẻ HTML)
* JavaScript: Một ngôn ngữ lập trình trong ứng dụng web nó đóng vai trò tạo ra các hiệu ứng và xữ lý ở phía Client.

Nếu như bạn chưa biết về 3 thành phần này thì bạn cần phải nhanh chóng học ngay. Nó là nền tảng cơ bản của lập trình web.

b. HTTP Get, HTTP Post

* HTTP Get: Phương thức gửi dữ liệu đến server mà thông tin nằm ngay trong yêu cầu trang (URL). Ví dụ: *MyWeb/Login?id=abc&pass=12345*

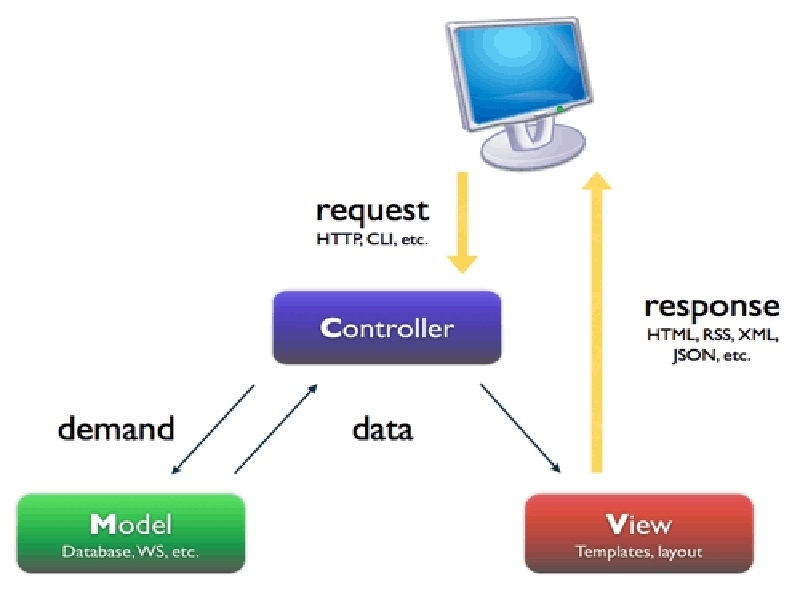
Phần màu xanh là các tham số ( dữ liệu gửi đến server).

* HTTP Post: Phương thức gửi dữ liệu về server mà thông tin được gửi ngầm không hiển thị trên URL. Ví dụ: Khi các bạn nhấn button đăng nhập thì URL giữ nguyên không có tham số phía sau.

c. Session

Khi bạn truy cập vào một trang web và đăng nhập vào đó. Bạn tắt trình duyệt di một thời gian ngắn (vài phút) và sau đó bạn quay lại với trang web đó, thật bất ngờ nó vẫn lưu lại tài khoản và mật khẩu của bạn. Đó là cơ chế session. Nó lưu một thông tin nào đó trong một khoản thời gian. Thông tin này được lưu ở phía server. Session thường được dùng cho việc lưu thông tin đăng nhập.

**2.2. Mô hình MVC**



MVC (Model - View - Controller) đây là ba thành phần của một mô hình lập trình. Một ứng dụng phát triển trên mô hình MVC chia làm 3 phần:

* Model: Chứa các kho lưu trữ và xữ lý nghiệp vụ. Trong ứng dụng này nó kết nối CSDL và thực hiện thao tác lấy dữ liệu lên, xữ lý dữ liệu, vv…
* View: Chứa các thành phần giao diện mà người dùng tương tác.
* Controller: Nhận lấy dữ liệu từ người dùng. Gọi Model xữ lý dữ liệu đó sau đó tìm View phù hợp gửi kèm dữ liệu vào đó.

**2.3. Chèn mã C# vào trang HTML**

ASP.Net trong phiên bản hiện tại hổ trợ mã razor (trang .cshtml). Mã này bao gồm mã html và C#. Để chèn một dòng code C# ta sử dụng @< mã > ngay trong trang cshtml. Nếu muốn chèn nhiều dòng bạn có thể dùng @{ <đoạn mã>}. Để hiểu rõ hơn bạn có thể xem tiếp phần View ( mục 5 ).

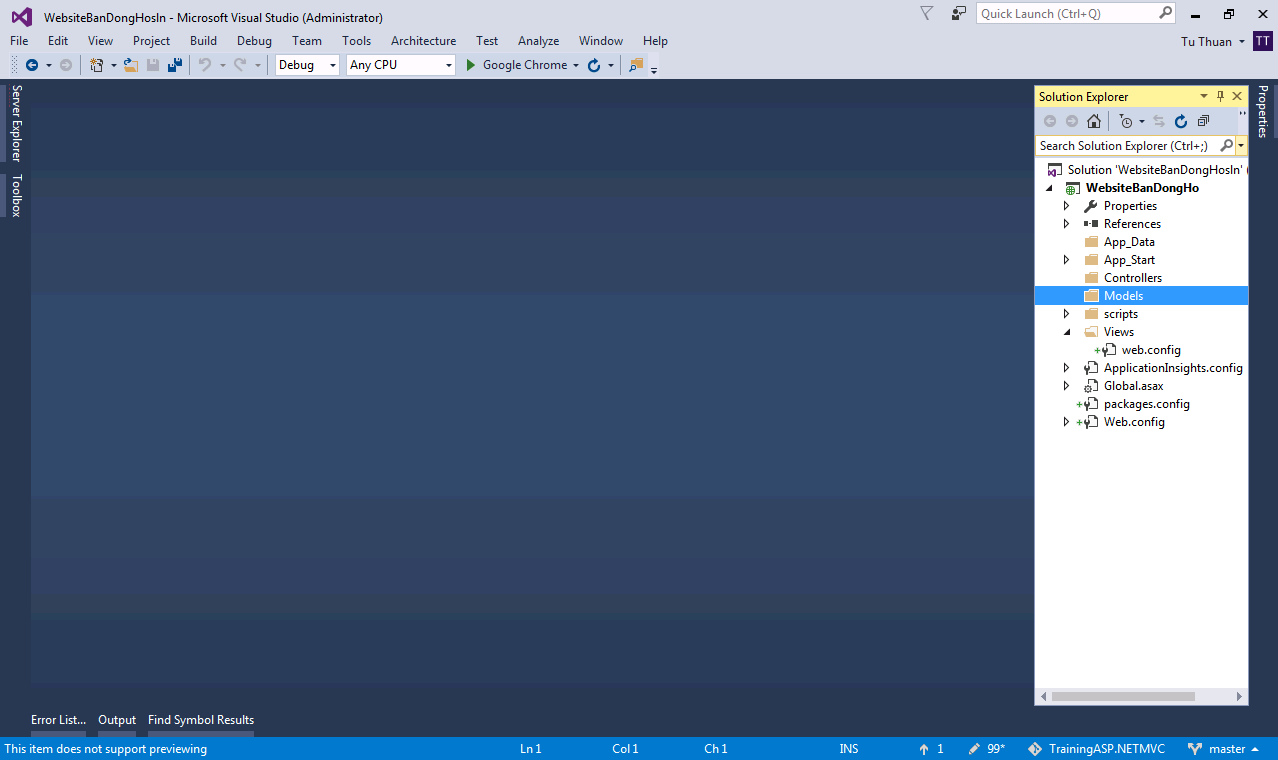
**2.4. Cơ bản về URL trong ASP.Net**

Để đi đến một trang web bạn cần gửi một request – nghe thì thật là phức tạp nhưng cứ hiểu nó là URL mà bản ghỏ để đi đến một trang web ví dụ như [*http://www.google.com*](http://www.google.com). Trong ASP.Net cho phép bạn cấu hình một URL để người dùng truy cập. Mặc định là {Controller}/{Action}. Bạn có thể tham khảo nó trong file RouteConfig.cs trong thư mục App\_Start. Trong các ví dụ sau mình sẽ hướng dẫn các bạn cấu hình URL này.

1. **Model**

Như đã nói từ trước Model là thành phần chứa các kho dữ liệu, kết nối với CSDL và chứa các xữ lý nghiệp vụ.

Thư mục Models được tạo ra khi bạn tạo dự án MVC.



Trong dự án này Model đóng vai trò kết nối với CSDL. Chuyển dữ liệu từ CSDL ra View. Và chứa các dịch vụ xữ lý nghiệp vụ.

Thực hiện thêm 4 thư mục sau vào Models: Context, ViewModel, Service, Implement.

* Context : Chứa model ADO Entity – các class kết nối với CSDL, các class được ánh xạ từ các bảng dữ liệu của SQL Server.
* ViewModel: Một trang view không phải chỉ chứa thông tin của một bảng trong SQL mà có thể chứa nhiều thông tin khác. ViewModel chứa các lớp mô tả thông tin có trên một View.
* Service: Chứa các Interface, mỗi interface cung cấp một số dịch vụ.
* Implement: Chứa các Class kế thừa các Interface trong service.

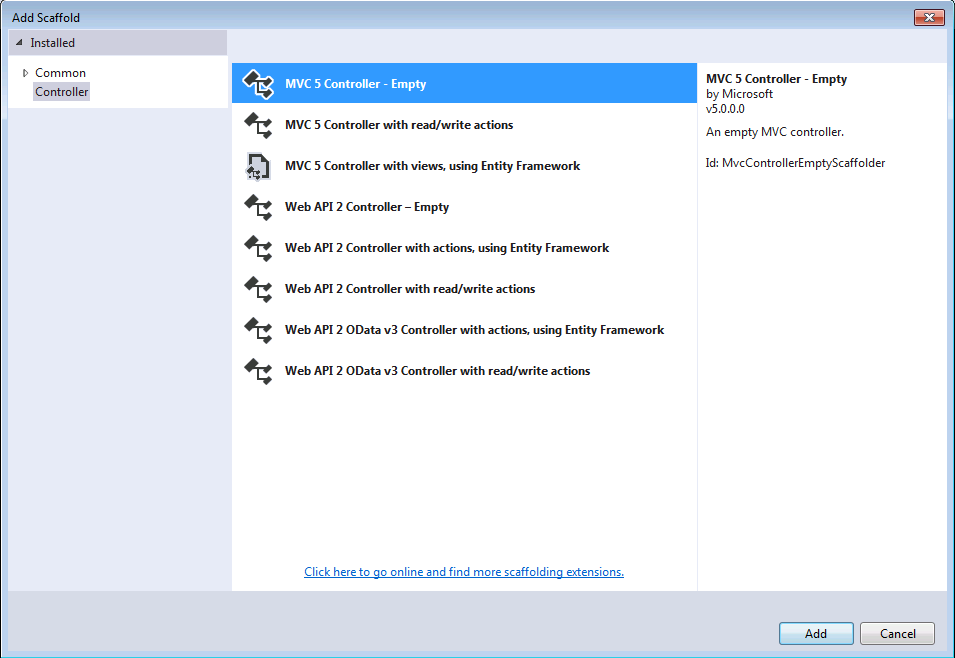
Chú ý: Nếu như bạn chưa biết gì về Entity framework hoặc chưa biết sử dụng ADO.NET Entity thì nagy lập tức tìm hiểu trước khi đọc qua các phần sau nha.

1. **Controller**

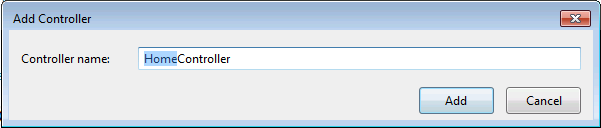
Thư mục này chứa các controller – các class kế thừa class Controller. Mỗi controller chứa các Action xử lý một hành động nào đó. Khi người dùng gửi một yêu cầu thì router sẽ dựa vào URL mà người dụng gửi tìm controller phù hợp sau đó tìm đến Action thích hợp trong controller đó và gọi nó xữ lý.

Trong bài hôm nay bạn sẽ cần phải tạo một controller có tên là Home, controller này xữ lý các yêu cầu cho trang chủ của website. Cách tạo một controller trong VS 2015 như sau:

Chuột phải thư mục Controller và chọn Add, Controller…



Chọn Empty controller và nhấn Add.



Đặt lại tên là Home với hậu tố là Controller – điều này là mặc định. Nhấn Add và VS sẽ tạo cho bạn một Controller.

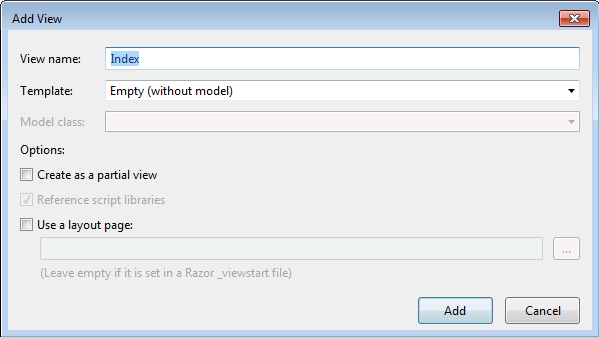
Chú ý: Sau này nếu chúng tôi nói: “ Tạo ra một controller ” thì tức là các bước tương tự như vầy.

1. **View**

View là thành phần hiển thị nội dung ra màn hình cho người dùng. View trong ASP.NET là một file có đuôi .cshtml. View được viết bằng các mã HTML và C# điều này đã được đề cập phía trên. Một Action của controller có thể trả về một View nên một View **thường** liên quan đến một Action. Trong ASP.Net MVC để thêm một View cho một Action cách đơn giản nhất là cho phép Action đó trả về một View. Giờ tiến hành thêm Action Index cho HomeController như sau.

|  |
| --- |
| C# |
| public class HomeController : Controller  {  // GET: Home  public ActionResult Index()  {  return View();  }  } |

Tiến hành chuột phải vào phương thức View() trong return và chọn add view. Điều chỉnh các thông số như màn hình phía dưới.

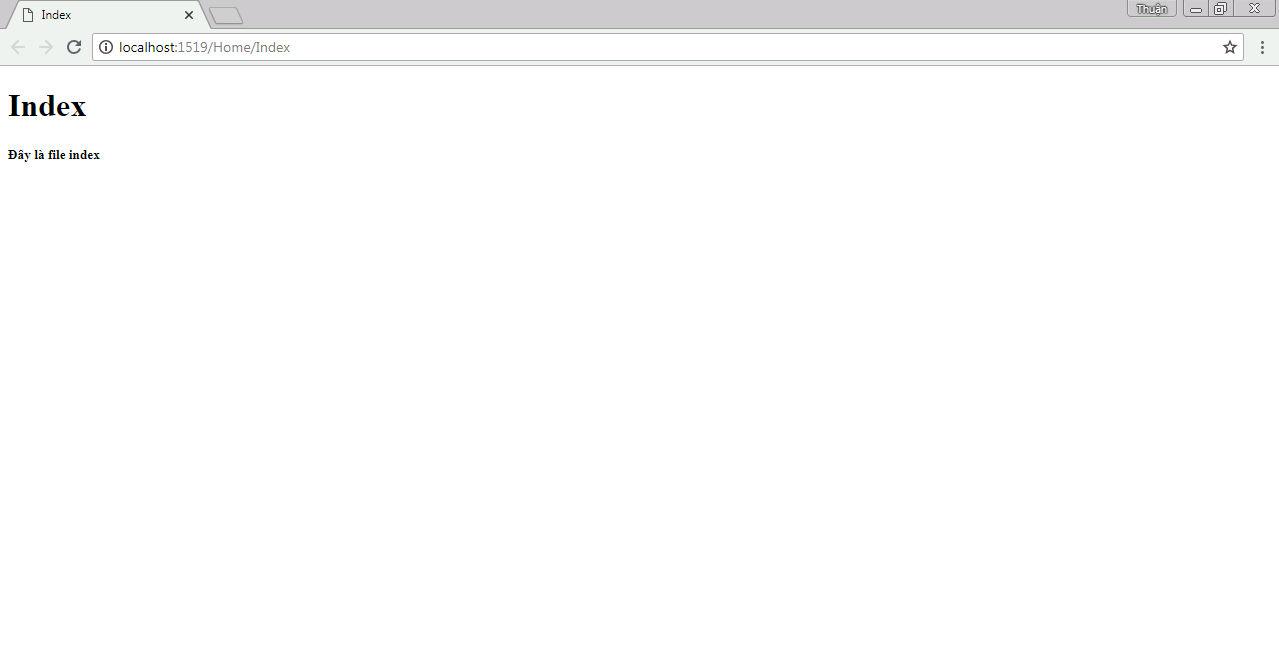


Nhấn Add và ta lập tức thấy một thư mục Home trong thư mục Views. File Index.cshtml nằm trong thư mục Home này. Điều này giúp bản quản lý các controller và các view liên quan một các dễ dàng hơn.

Bây giờ mở file Index.cshtml vừa tạo lên và sửa đổi nó như sau.

|  |
| --- |
| html |
| @{  Layout = null;  }  <!DOCTYPE html>  <html>  <head>  <meta name="viewport" content="width=device-width" />  <title>Index</title>  </head>  <body>  <div>  <h1>Index</h1>  <h5>Đây là file index</h5>  </div>  </body>  </html> |

Ok! Bây giờ tiến hành chạy thử ứng dụng bằng cách nhấn F5.



Chú ý: bạn cần chú ý URL và phần nội dung hiển thị trên màn hình.

1. **Kết quả**

Trong bài này ta tìm hiểu được ASP.NET cơ bản cách tạo ra các Controller, Action và các View. Trong bài tiếp theo chúng ta tiến hành đi sâu vào từng thành phần Model, Controller và View.