

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP QUẢNG NINH

GIÁO TRÌNH
THIẾT KẾ WEB
DÙNG CHO TRÌNH ĐỘ ĐẠI HỌC

QUẢNG NINH - 2019

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN VỀ ASP.NET VÀ C#	1
1.1. Sự ra đời của .NET	4
1.2. .NET Framework là gì ?	4
1.3. Giới thiệu ASP.NET	6
1.4. Cài đặt Web Server IIS	7
1.5. Cài đặt Visual Studio 2010 Ultimate (trên Win 7)	12
1.6. Tạo ứng dụng Website đầu tiên	28
1.7. Phân loại tập tin trong ASP.NET	30
1.8. Trang ASP.NET	30
1.9. Điều khiển sự kiện của trang ASP.NET	32
1.10. Cơ bản về C#	32
CÂU HỎI ÔN TẬP CHƯƠNG 1	36
CHƯƠNG 2: ĐIỀU KHIỂN WEB SERVER CONTROL	37
2.1. Lý do sử dụng các điều khiển Web Server	37
2.2. Các điều khiển Standard Web Server	37
2.3. Điều khiển kiểm tra dữ liệu Validation	42
2.4. Điều khiển FileUpload	54
CÂU HỎI ÔN TẬP CHƯƠNG 2	58
CHƯƠNG 3: TẠO VÀ SỬ DỤNG USER CUSTOM CONTROL (UCC)	61
3.1. Giới thiệu User Custom Control	61
3.2. Các bước tạo User Custom Control	61
3.3. Ví dụ minh họa tạo UCC	62
CÂU HỎI ÔN TẬP CHƯƠNG 3	66
CHƯƠNG 4: THIẾT KẾ WEBSITE VỚI MASTERPAGE	67
4.1. MasterPage là gì?	67
4.2. Tạo MasterPage	67
CÂU HỎI ÔN TẬP CHƯƠNG 4	71
CHƯƠNG 5: ASP.NET VÀ CƠ SỞ DỮ LIỆU	72
5.1. Cơ sở dữ liệu SQL Server	72
5.2. Các bước cần thực hiện trước khi thao tác với cơ sở dữ liệu	78
5.3. Công nghệ ADO.NET	79
5.4. Các lớp thao tác với cơ sở dữ liệu SQL Server 2008	85

5.4.1. Lớp Connection	85
5.4.2. Lớp Command	88
5.4.3. Lớp DataReader	95
5.4.4. Lớp DataTable	98
5.4.5. Lớp DataSet	100
5.4.6. Lớp DataAdapter.....	100
5.5. Data Binding.....	103
5.5.1. Giới thiệu Data Binding	103
5.5.2. Dạng gắn kết dữ liệu có sự lặp lại (Repeated Data Binding)	103
5.6. Các điều khiển Data Source.....	105
5.6.1. Giới thiệu về Data Source Controls.....	105
5.6.2. Sử dụng SqlDataSource để chọn (select) dữ liệu và hiển thị thông qua điều khiển GridView	105
5.6.3. Sử dụng SqlDataSource để cập nhật, xóa dữ liệu thông qua điều khiển GridView	108
5.7. Điều khiển GridView.....	112
5.7.1. Tổng quan về GridView	112
5.7.2. Các mẫu hiển thị cho GridView	114
5.7.3. Tạo cột BoundField thủ công.....	115
5.8. Các điều khiển hỗ trợ Templates	117
5.8.1. Giới thiệu tổng quan	117
5.8.2. Ví dụ minh họa thiết kế một Template cho GridView	118
BÀI TẬP CHƯƠNG 5	124
CHƯƠNG 6: XÂY DỰNG ỨNG DỤNG WEBSITE BẰNG ASP.NET	129
6.1. Giao diện phía trang chủ	129
6.2. Giao diện phía quản trị	129
6.3. Xây dựng trang đăng nhập	131

CHƯƠNG 1: TỔNG QUAN VỀ ASP.NET VÀ C#

1.1. Sự ra đời của .NET

Trước đây và cả ngày nay, trong lĩnh vực phát triển phần mềm có rất nhiều (hàng ngàn thậm chí hàng vạn) ngôn ngữ lập trình được sử dụng để phát triển phần mềm (như Delphi, Ada, Cobol, Fortran, Basic, LISP, Prolog, Foxpro, Java, Pascal, C/C++, Visual Basic, VC++, C#...). Mỗi ngôn ngữ đều có những ưu và nhược điểm riêng, chẳng hạn Fortran là lựa chọn số một cho các tính toán khoa học; Prolog là lựa chọn rất tốt để phát triển các phần mềm thông minh (AI, Expert Systems...); Java có lợi thế phát triển các ứng dụng mạng, ứng dụng Mobile và độc lập hệ điều hành (Write One – Run Everywhere); Visual Basic tỏ ra dễ học và dễ phát triển các ứng dụng Winform; C# vượt trội bởi sự kết hợp giữa sức mạnh của C++ và sự dễ dàng của Visual Basic...

Những ưu điểm có tính đặc thù của từng ngôn ngữ là điều đã được khẳng định. Tuy nhiên, điều mà ai cũng thấy rõ là rất khó để có thể tận dụng được sức mạnh của tất cả các ngôn ngữ lập trình trong một dự án phần mềm, chẳng hạn không thể hoặc rất khó khăn để viết một ứng dụng có sử dụng đồng thời cả ngôn ngữ Visual Basic và Java hay Foxpro với Delphi v.v... Nói cách khác, việc “liên thông” giữa các ngôn ngữ là gần như không thể.

Cũng do sự khác biệt giữa các ngôn ngữ lập trình mà việc tiếp cận hay chuyển đổi sang ngôn ngữ lập trình mới sẽ tốn rất nhiều thời gian (tuy rằng về tư tưởng và nguyên lý có tương tự nhau). Vì vậy, khi các dự án sử dụng ngôn ngữ lập trình khác nhau thì chi phí cho chuyển đổi/ học hỏi sẽ là rất lớn, gây lãng phí thời gian không cần thiết và chất lượng phần mềm chắc chắn không cao.

Ngoài ra, cùng với sự phát triển như vũ bão của Internet thì mô hình phát triển ứng dụng cũng rất khác xưa. Các ứng dụng ngày nay không chỉ chạy riêng lẻ (stand-alone) trên máy tính PC mà còn có thể chạy trên môi trường mạng, cung cấp hay truy cập các dịch vụ từ xa (ứng dụng phân tán). Vai trò của phần mềm đã dần chuyển từ chỗ cung cấp các chức năng (Functional) cụ thể sang cung cấp các dịch vụ (Services).

Từ những hạn chế trong quá trình phát triển phần mềm như đã nêu, đòi hỏi phải có một cách tiếp cận sao cho tối ưu nhất, vừa đảm bảo tốn ít chi phí chuyển đổi vừa đảm bảo nhiều người có thể tham gia cùng một dự án mà không nhất thiết phải viết trên cùng một ngôn ngữ lập trình, đồng thời ứng dụng phải hoạt động tốt trong môi trường mạng Internet. Đó chính là lý do để Microsoft cho ra công nghệ phát triển phần mềm mới .NET!

Microsoft .NET là một nền tảng (Platform) phát triển ứng dụng mới và hoàn chỉnh nhất từ trước tới nay. Sự ra đời của Microsoft.NET có tính cách mạng, nó đem đến cho các nhà lập trình một phong cách phát triển phần mềm đột phá, khắc phục hầu hết các hạn chế trước đây của các ngôn ngữ lập trình. Việc sử dụng .NET không chỉ giúp phát triển các ứng dụng đơn lẻ mà còn có thể phát triển các ứng dụng phân tán ở qui mô rất lớn; .NET làm giảm thiểu thời gian phát triển ứng dụng, nâng cao rõ rệt chất lượng sản phẩm phần mềm.

Phiên bản .NET đầu tiên (v 1.0) được Microsoft đưa ra thị trường vào năm 2001.

1.2. .NET Framework là gì ?

.NET Framework của Microsoft là một *nền tảng lập trình* tập hợp các thư viện lập trình có thể được cài thêm hoặc đã có sẵn trong các hệ điều hành Windows. Nó cung cấp những giải pháp thiết yếu cho những yêu cầu thông thường của các chương trình điện toán như lập trình giao diện người dùng, truy cập dữ liệu, kết nối cơ sở dữ liệu, ứng dụng web, các giải thuật số

học và giao tiếp mạng. Ngoài ra, **.NET Framework** quản lý việc thực thi các chương trình được viết dựa trên **.NET Framework** do đó người dùng cần phải cài **.NET Framework** để có thể chạy các chương trình được viết trên nền **.NET**.

Chẳng hạn, để thiết kế một trò chơi đua xe, nếu không có bộ Framework chuyên dụng cho game, người lập trình game phải tự tạo ra: khung xe, bánh xe, người, đường đi, cây, biển báo... rồi mới tính đến chuyện “lắp ghép” chúng lại với nhau để tạo ra không gian cho game; trong khi cũng với dạng trò chơi này, nhưng nếu dùng bộ Framework có sẵn đã được phát triển thì người lập trình viên chỉ cần viết các lệnh để lấy chúng ra từ Framework và ghép chúng lại.

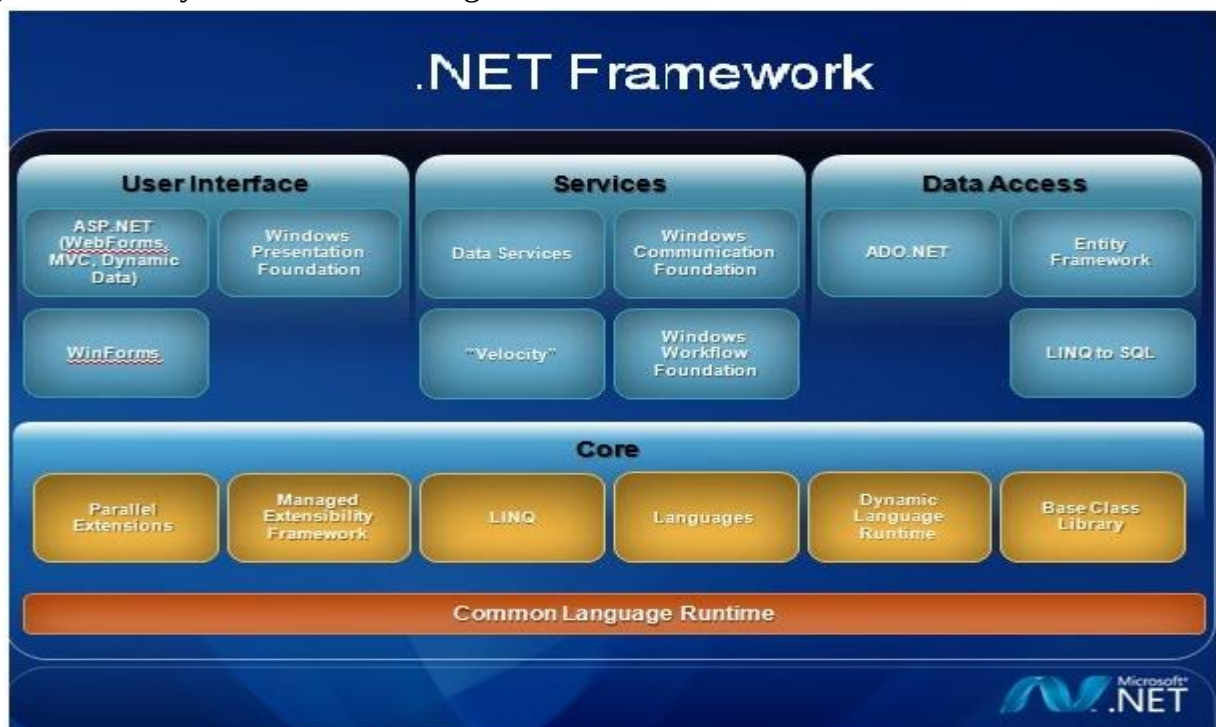
Không phải mọi ngôn ngữ lập trình đều khai thác được Framework, muốn sử dụng các “vật liệu” trong bộ Framework, đòi hỏi người lập trình viên phải dùng các ngôn ngữ lập trình có hỗ trợ công nghệ **.NET** như VB.NET, C#.NET, ASP.NET...

Framework có 2 thành phần chính là:

1. *Common Language Runtime (CLR)*

CLR là thành phần kết nối giữa các phần khác trong **.NET Framework** với hệ điều hành. CLR là chương trình viết bằng **.NET**, không được biên dịch ra mã máy mà nó được dịch ra một ngôn ngữ trung gian Microsoft Intermediate Language (MSIL). Khi chạy chương trình, CLR sẽ dịch MSIL ra mã máy để thực thi các tính năng, đảm bảo ứng dụng không chiếm dụng và sử dụng tràn lan tài nguyên của hệ thống. Nó cũng không cho phép các lệnh nguy hiểm được thi hành. Các chức năng này được thực thi bởi các thành phần bên trong **CLR** như **Class loader**, **Just In Time compiler**, **Garbage collector**, **Exception handler**, **COM marshaller**, **Security engine**, ...

Trong các phiên bản hệ điều hành Windows mới như XP.Net và Windows 2003, CLR được gắn kèm với hệ điều hành. Điều này đảm bảo ứng dụng viết ra trên máy tính của bạn sẽ chạy trên các máy tính khác mà không cần cài đặt.



2. .NET Framework class library

.NET Framework class library cung cấp thư viện lập trình như cho ứng dụng, cơ sở dữ liệu, dịch vụ web...

- Base class library – thư viện các lớp cơ sở

Đây là thư viện các lớp cơ bản nhất, được dùng trong khi lập trình hay bản thân những người xây dựng **.NET Framework** cũng phải dùng nó để xây dựng các lớp cao hơn. Ví dụ các lớp trong thư viện này là **String**, **Integer**, **Exception**, ...

- ADO.NET và XLM

Bộ thư viện này gồm các lớp dùng để xử lý dữ liệu. **ADO.NET** thay thế **ADO** để trong việc thao tác với các dữ liệu thông thường. Các lớp đối tượng **XML** được cung cấp để bạn xử lý các dữ liệu theo định dạng mới : **XML**. Các ví dụ cho bộ thư viện này là **SqlDataAdapter**, **SqlCommand**, **DataSet**, **XMLReader**, **XMLWriter**, ...

- ASP.NET

Ứng dụng Web xây dựng bằng **ASP.NET** tận dụng được toàn bộ khả năng của **.NET Framework**. Bên cạnh đó là một phong cách lập trình mới mà Microsoft đặt cho nó một tên gọi là **code behind**. Đây là cách mà lập trình viên xây dựng các ứng dụng **Windows based** thường sử dụng – giao diện và lệnh được tách riêng. Tuy nhiên, nếu bạn đã từng quen với việc lập trình ứng dụng web, đây là việc mà bạn giải phóng khỏi các lệnh HTML.

Sự xuất hiện của **ASP.NET** làm cân xứng giữa quá trình xây dựng ứng dụng trên Windows và Web. **ASP.Net** cung cấp một bộ các **Server Control** để lập trình viên bắt sự kiện và xử lý dữ liệu của ứng dụng như đang làm việc với ứng dụng của Windows. Nó cũng cho phép bạn chuyển một ứng dụng trước đây viết chỉ chạy trên Windows thành một ứng dụng Web khá dễ dàng. Ví dụ cho các lớp trong thư viện này là **WebControl**, **HTML Control**, ...

- Web services

Web services là các dịch vụ được cung cấp qua Web (hay Internet). Dịch vụ được coi là **Web service** không nhằm vào người dùng mà nhằm vào người xây dựng phần mềm. **Web services** có thể dùng để cung cấp các dữ liệu hay một chức năng tính toán

- Windows form

Bộ thư viện về **Windows form** gồm các lớp đối tượng dành cho việc xây dựng các ứng dụng **Windows based**. Việc xây dựng ứng dụng loại này vẫn được hỗ trợ tốt từ trước đến nay bởi các công cụ và ngôn ngữ lập trình của Microsoft. Giờ đây, ứng dụng chỉ chạy trên Windows sẽ có thể làm việc với ứng dụng Web dựa vào **Web service**. Ví dụ về các lớp trong thư viện này là **Form**, **UserControl**, ...

1.3. Giới thiệu ASP.NET

ASP.NET là công nghệ phát triển các ứng dụng trên nền web, thế hệ kế tiếp của ASP (Active Server Page – Trang web được xử lý bên phía máy chủ). ASP.NET là một thành phần nội tại (có sẵn) của **.NET Framework**. Vì vậy nó tận dụng được sức mạnh của **.NET Framework**. ASP.NET có một số ưu điểm chính:

- Có thể sử dụng để phát triển các ứng dụng web đủ mọi kích cỡ, từ ứng dụng nhỏ nhất cho đến ứng dụng toàn doanh nghiệp (Enterprise).

- Ứng dụng viết bằng ASP.NET dễ dàng tương thích với nhiều loại trình duyệt khác nhau. Nhà phát triển không cần phải quan tâm nhiều đến trình duyệt nào được sử dụng để duyệt website, điều này sẽ được framework tự render ra mã tương ứng.

- Khi sử dụng bộ IDE của Visual Studio, cách thức lập trình sẽ giống hệt như lập trình winform.

- Truy xuất dữ liệu bằng công nghệ ADO.NET có sẵn của .NET Framework.
- Chạy ứng dụng cực nhanh bởi cơ chế biên dịch và Cached.
- Có thể tăng tốc ứng dụng bằng cách Cache các điều khiển, các trang.
- Bảo mật vượt trội.
- Tốn ít dòng lệnh hơn so với ASP/PHP/Perl khi thực hiện cùng một công việc.
- Dễ dàng bảo trì và dễ đọc hơn bởi Code và Giao diện được tách biệt. Điều này cũng giúp cho tính chuyên biệt hóa cao hơn. (Một người chỉ lo code phần xử lý nghiệp vụ, người khác thì chỉ lo code phần giao diện v.v...).
- ASP sử dụng ngôn ngữ lập trình VB.NET hoặc C# hoặc cả hai để phát triển ứng dụng.

*** Môi trường phát triển ứng dụng ASP.NET**

- Môi trường máy tính độc lập (một máy tính đóng vai trò là Client đồng thời là Server). Sinh viên sẽ sử dụng môi trường này để phát triển ứng dụng ASP.NET

- Môi trường mạng LAN.
- Môi trường mạng Internet

*** Phát triển ứng dụng ASP.NET cần cài đặt những thành phần sau:**

- **Bước 1: Cài đặt Web Server IIS (có sẵn trong HĐH Windows)**
- **Bước 2: Cài đặt SQL Server 2008 Express (bản miễn phí)**
- **Bước 3: Cài đặt Visual Studio 2010 (thông thường khi cài đặt Full bộ cài Visual Studio 2010 có kèm theo cài đặt một phần SQL Server 2008 Express, để sử dụng được SQL Server 2008 Express cần tiến hành cài thêm SQL Server 2008 Express vào, không cần phải cài riêng như bước 2).**

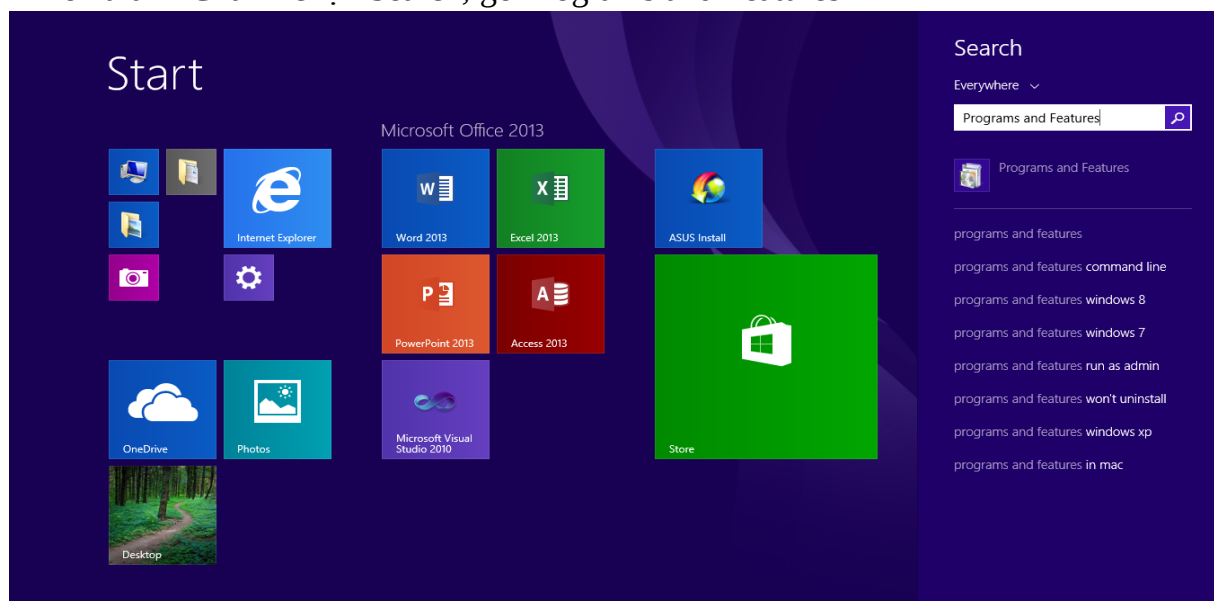
Các phần phía dưới hướng dẫn thực hiện Bước 1,3, vì Bước 3 đã bao gồm Bước 2.

1.4. Cài đặt Web Server IIS

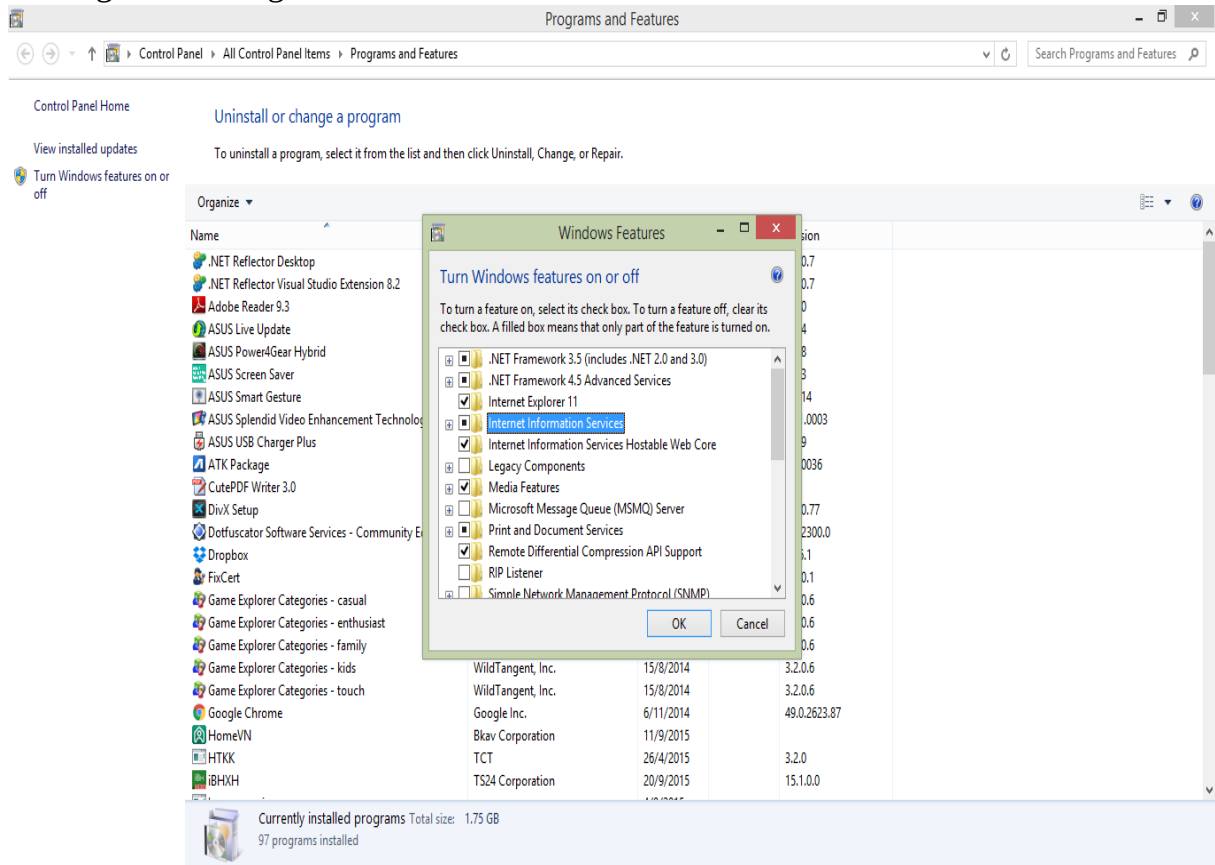
1) Cài đặt IIS

*** Cài đặt Web Server IIS trên Windows 8.1**

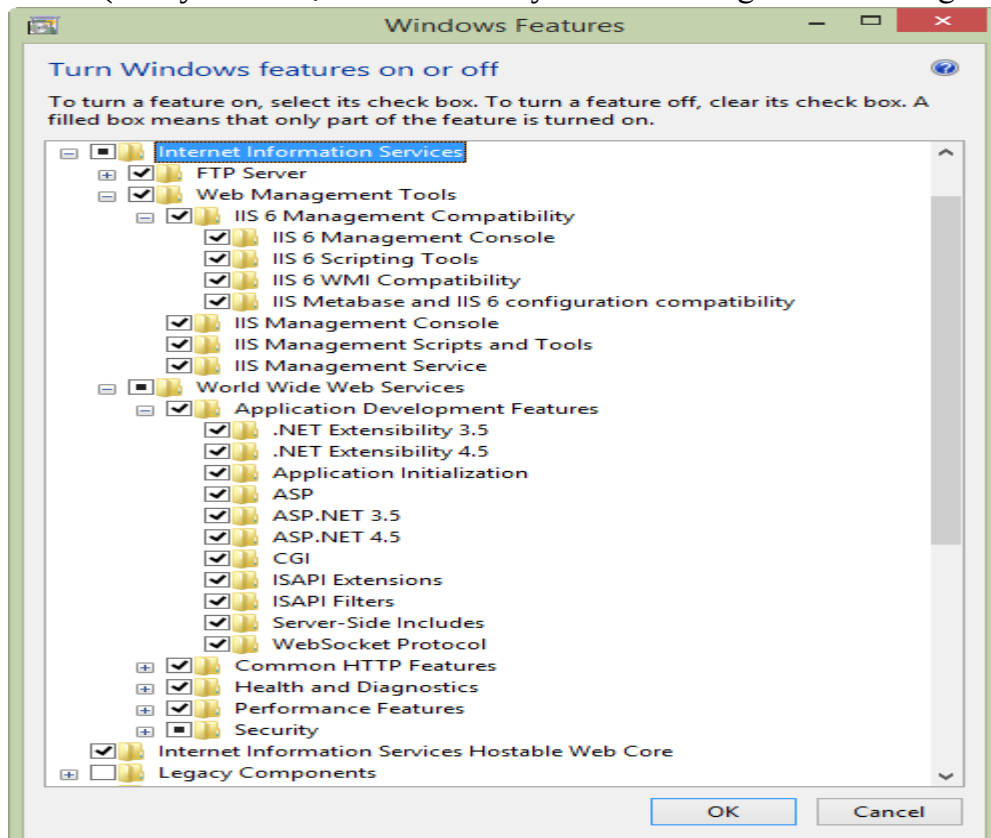
- Trên thanh Charm chọn Search, gõ Programs and Features



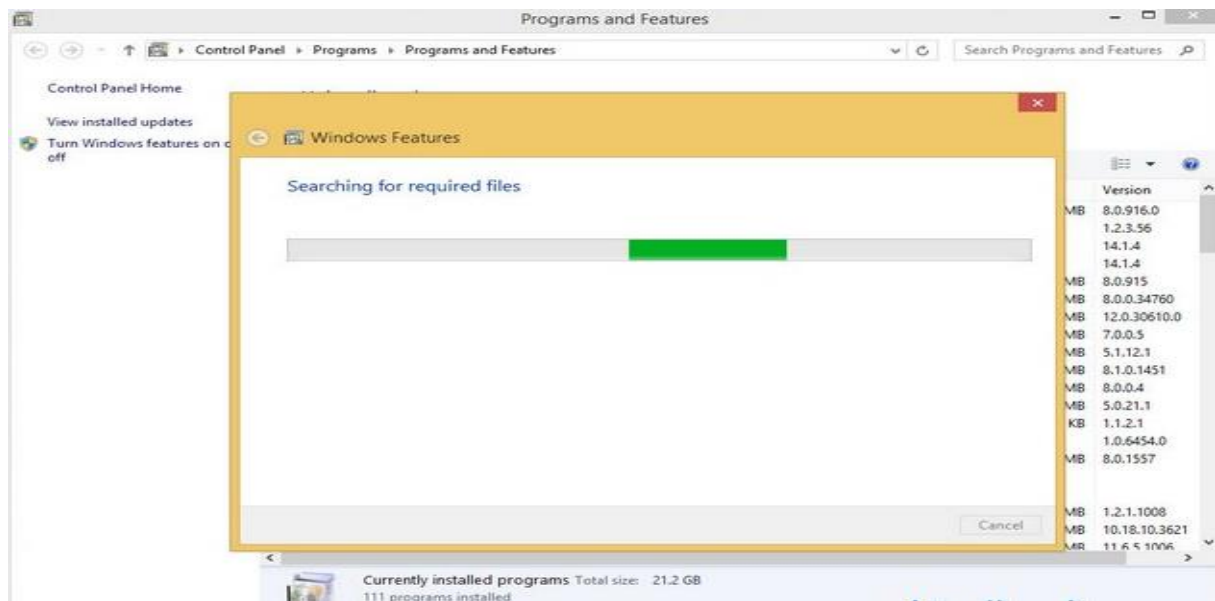
- Trong cửa sổ Programs and Features Click Turn Windows features on or off



- Mở rộng Internet Information Service. Trong đó có 3 phần, bạn chỉ cần quan tâm Web Management Tools & World Wide Web Services. Mở rộng 2 nhánh này và tick chọn vào checkbox. (Chú ý: nên chọn hết để sau này khỏi mất công đôi khi không chạy)

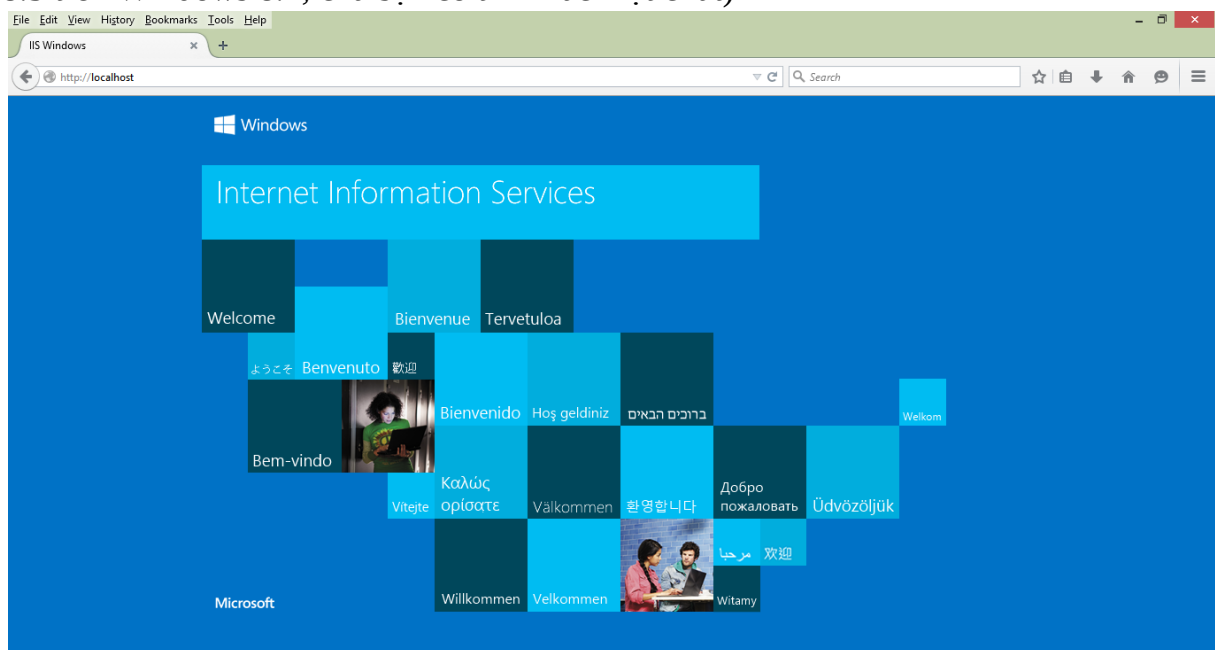


- Nhấn OK và chờ vài phút để hệ thống cập nhật lại. Khởi động lại máy tính của bạn nếu cần.



2) Kiểm tra kết quả cài đặt IIS

- Bạn mở trình duyệt lên gõ: <http://localhost> hoặc <http://127.0.0.1>. Nếu trình duyệt xuất hiện màn hình như dưới đây thì xin chúc mừng, bạn đã cài đặt thành công. (Hình bên dưới là IIS 8.5 trên Windows 8.1, của bạn có thể khác một chút)



- **Localhost** là địa chỉ của máy cục bộ mà bạn đang làm việc. Nếu máy của bạn đang kết nối vào mạng LAN và có một địa chỉ IP, bạn có thể dùng địa chỉ này thay cho localhost.

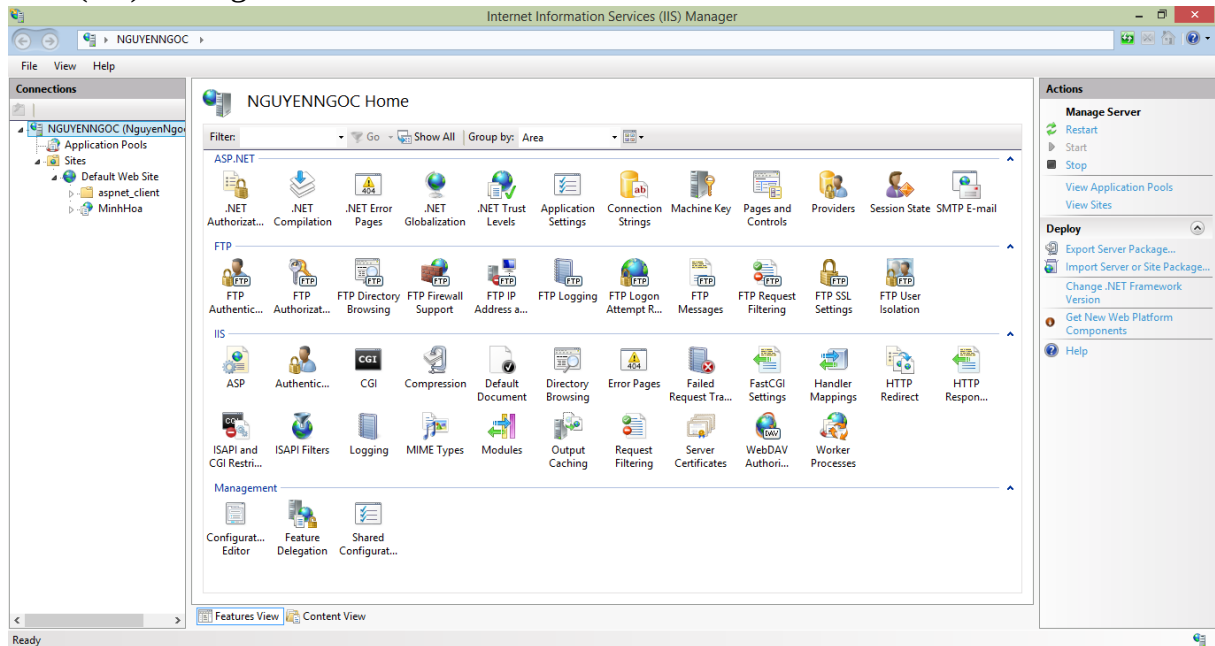
- Khi gõ `//localhost`, bạn sẽ thấy trong thanh địa chỉ tự động đổi thành: `http://localhost`. HTTP là giao thức mặc định được dùng trên Internet. Vì HTTP là một giao thức thuộc bộ TCP/IP, bạn cần có địa chỉ IP để các máy tính khác trong mạng có thể truy cập được đến trang web của bạn.

- Sau khi cài đặt Web Server, mặc định trên ổ đĩa C:\ sẽ có sẵn thư mục C:\inetpub\wwwroot. Đây là thư mục mà Web Server mặc định ánh xạ vào //localhost, do đó, các trang web đặt trong wwwroot có thể được truy cập bởi các máy tính khác.

3) Cấu hình IIS

Cấu hình IIS trên Windows 8.1

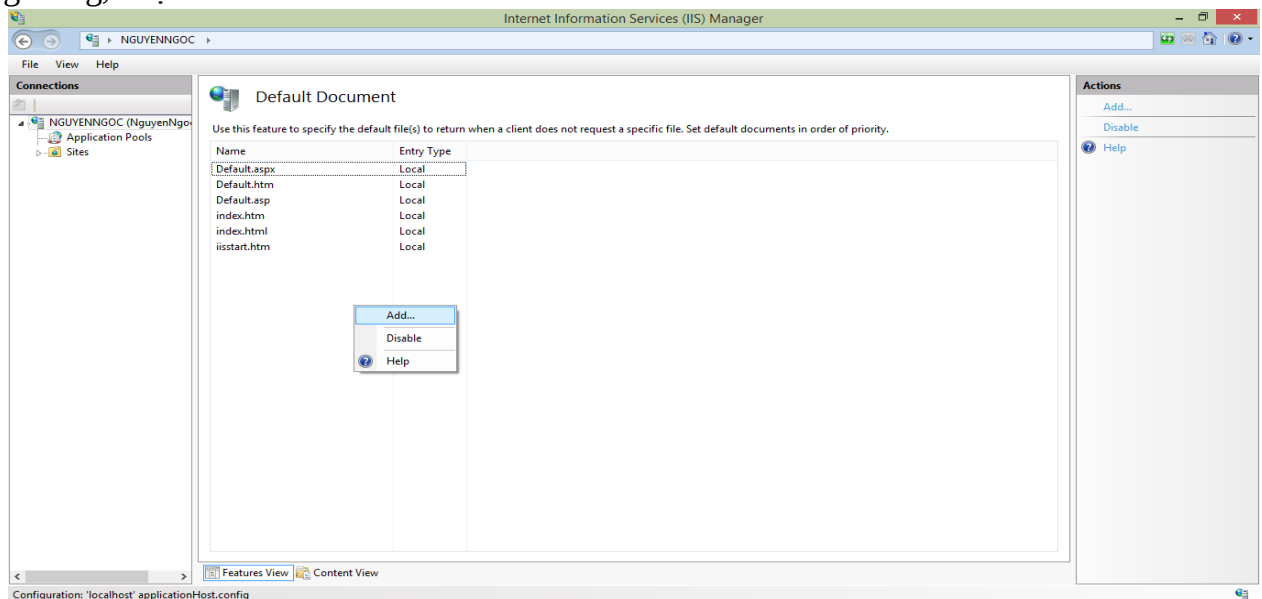
Mở Control Panel, chọn mục Administrative Tools, Click chọn Internet Information Services (IIS) Manager



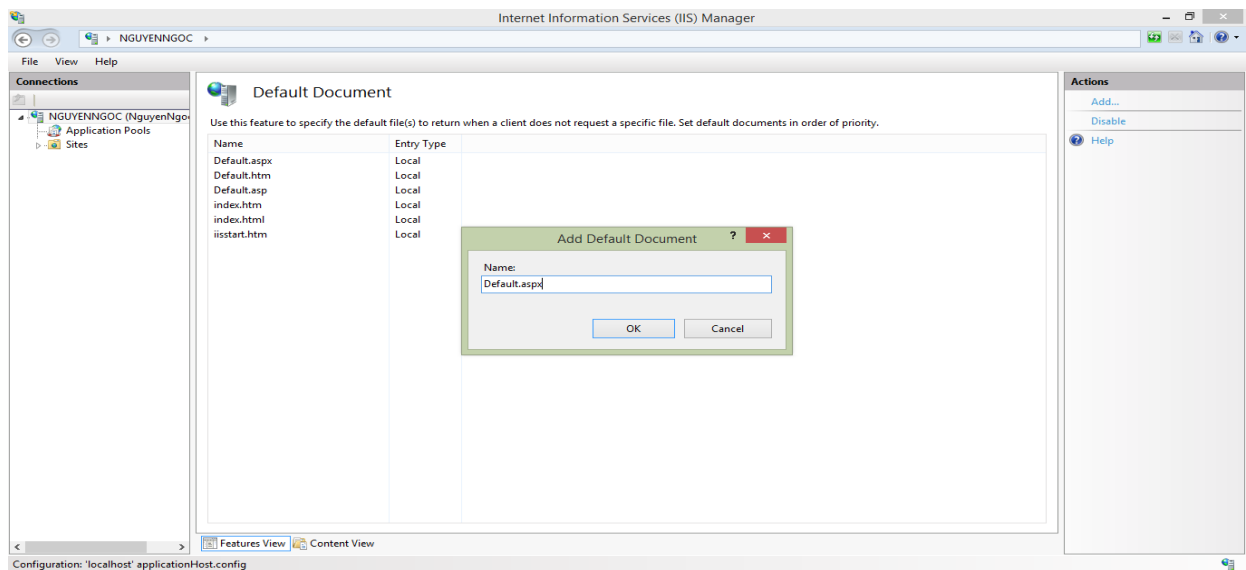
==> Thêm trang khởi chạy cho ứng dụng Web, chẳng hạn đối với ứng dụng ASP.NET thường có trang Default.aspx, muốn khi chạy ứng dụng Web trang này được chạy đầu tiên, tiến hành qua các bước sau:

- Click chọn tên Server hình trên server là NGUYENNGOC

- Click vào mục Default Document, trong cửa sổ Default Document, click chuột phải vào vùng trống, chọn Add...



- Xuất hiện màn hình dưới, gõ vào tên trang sẽ khởi chạy đầu tiên khi chạy ứng dụng Website.



4) Tạo các ứng dụng Web trên IIS

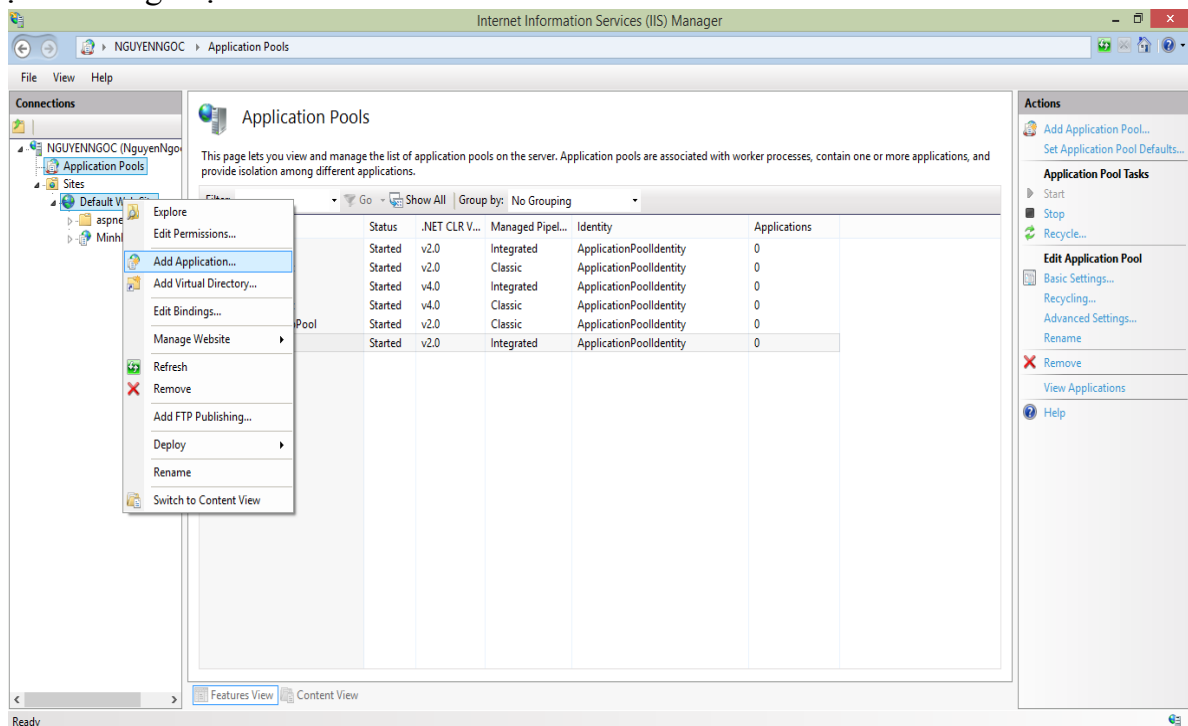
Có 2 cách để chạy một ứng dụng Web ASP.NET trên trình duyệt:

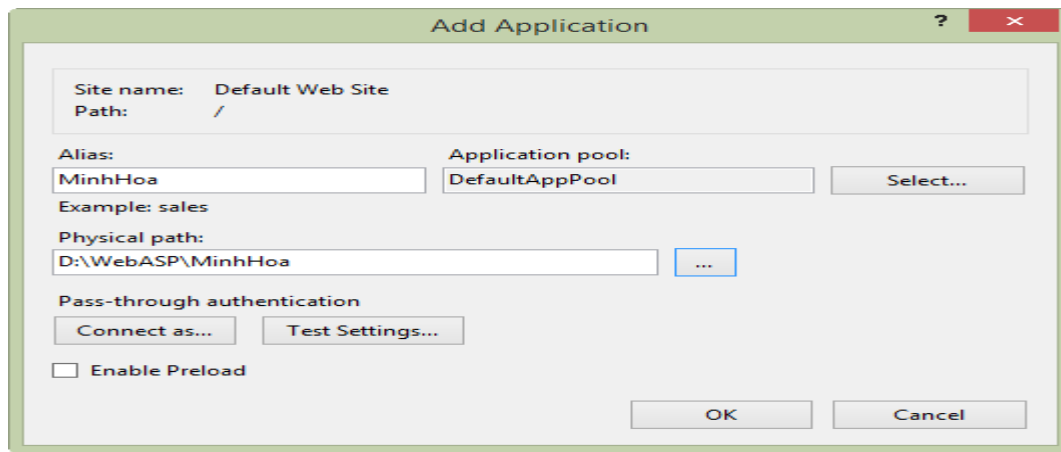
- Copy ứng dụng web vào thư mục wwwroot
- Trong Internet Information Services (IIS) Manager tạo một đường dẫn đến thư mục chứa Web.

=> **Cách tạo một đường dẫn đến thư mục Web ASP.NET**

Ví dụ: ta có sẵn một ứng dụng web có tên là “MinhHoa”, nằm tại D:\WebASP, tiến hành các bước sau để tạo đường dẫn đến ứng dụng web này:

+ Mở Internet Information Services (IIS) Manager, Click chuột phải vào Default Web site sau đó chọn như hình dưới để tạo một ứng dụng trong Default Web Site, sau này trên trình duyệt chỉ cần gõ địa chỉ Localhost/MinhHoa:





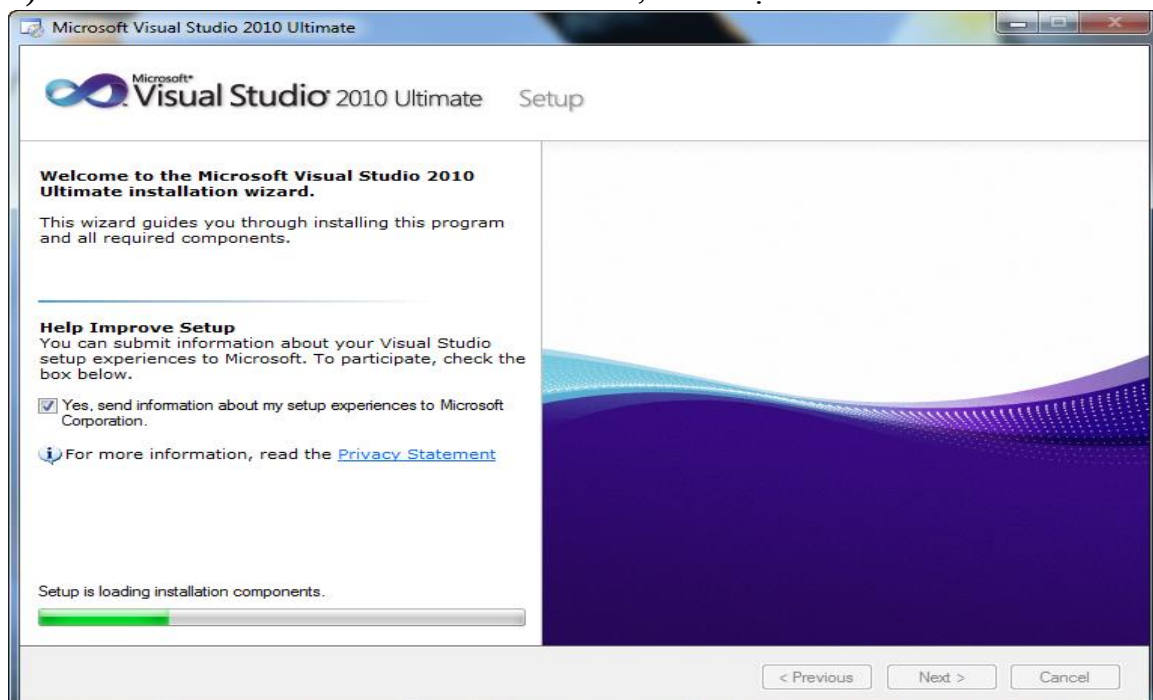
1.5. Cài đặt Visual Studio 2010 Ultimate (trên Win 7)

Cần đĩa cài đặt Visual Studio 2010 Ultimate

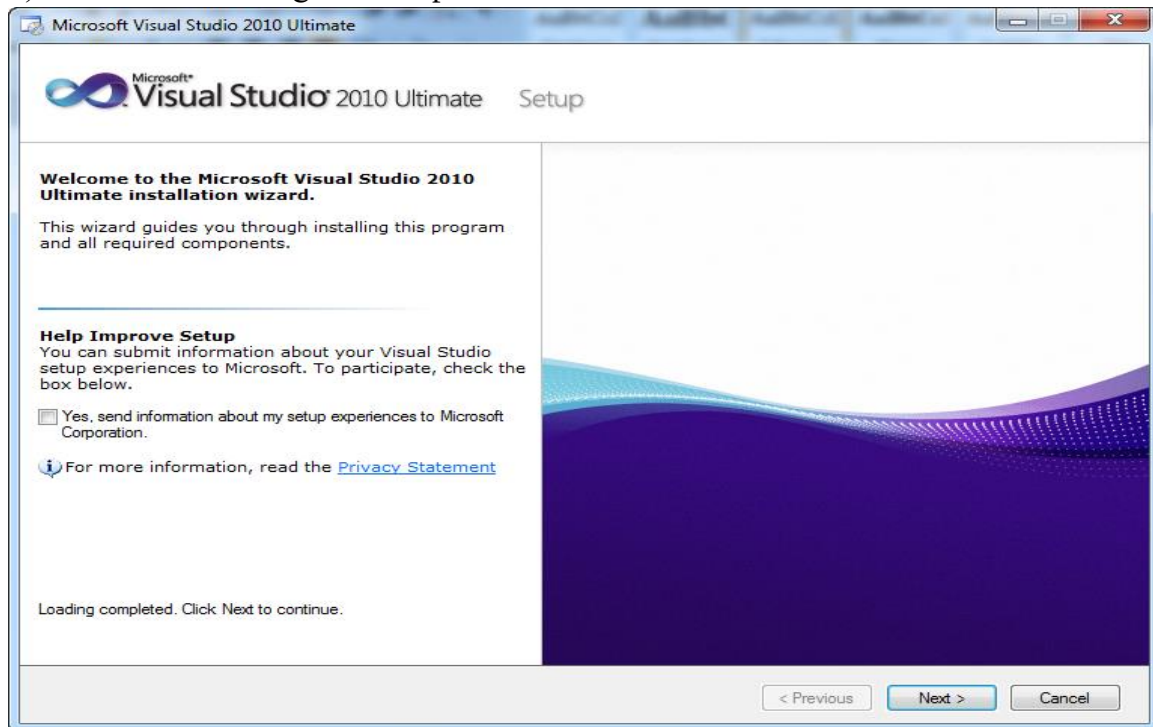
1) Click file cài đặt setup.exe



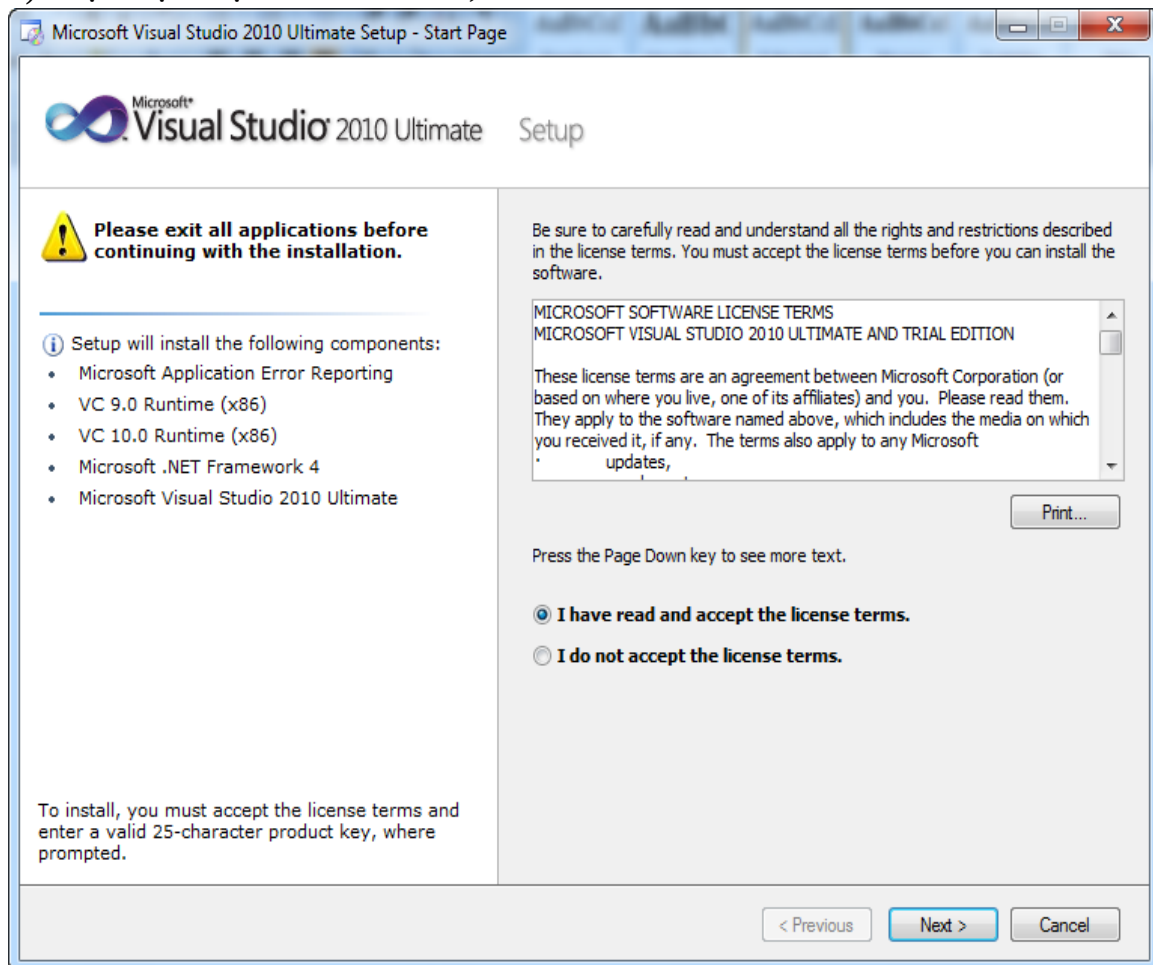
2) Click Install Microsoft Visual Studio 2010, xuất hiện cửa sổ dưới



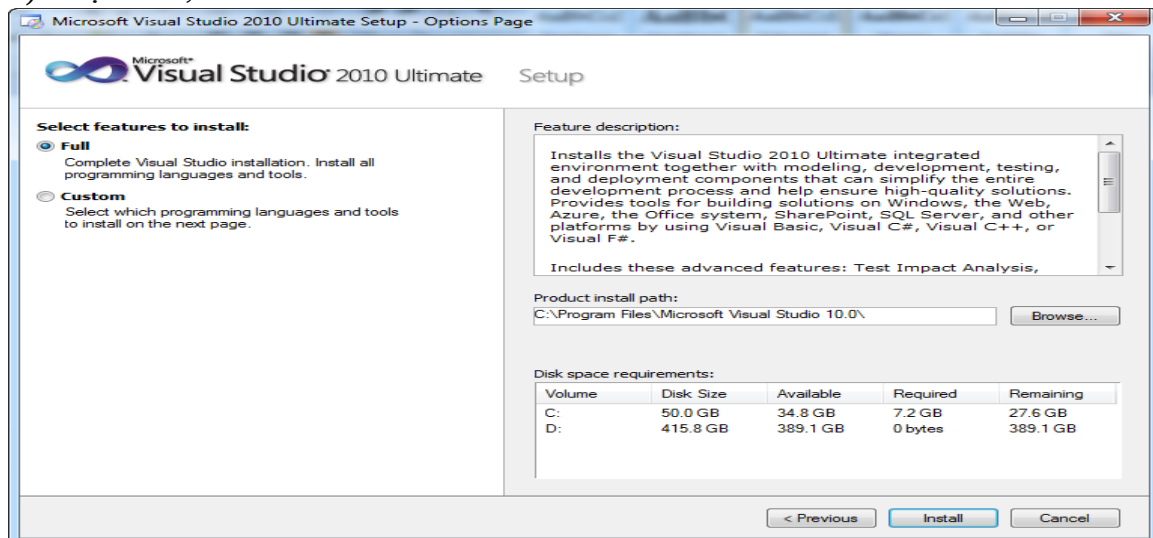
3) Click Next để sang bước tiếp theo



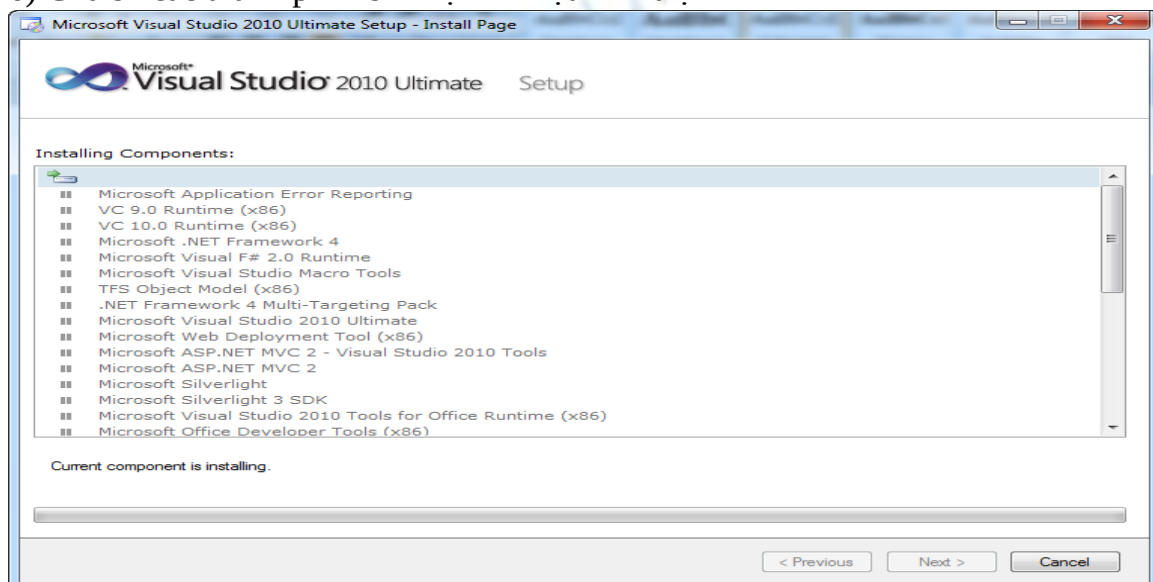
4) Thực hiện chọn như hình dưới, Click Next



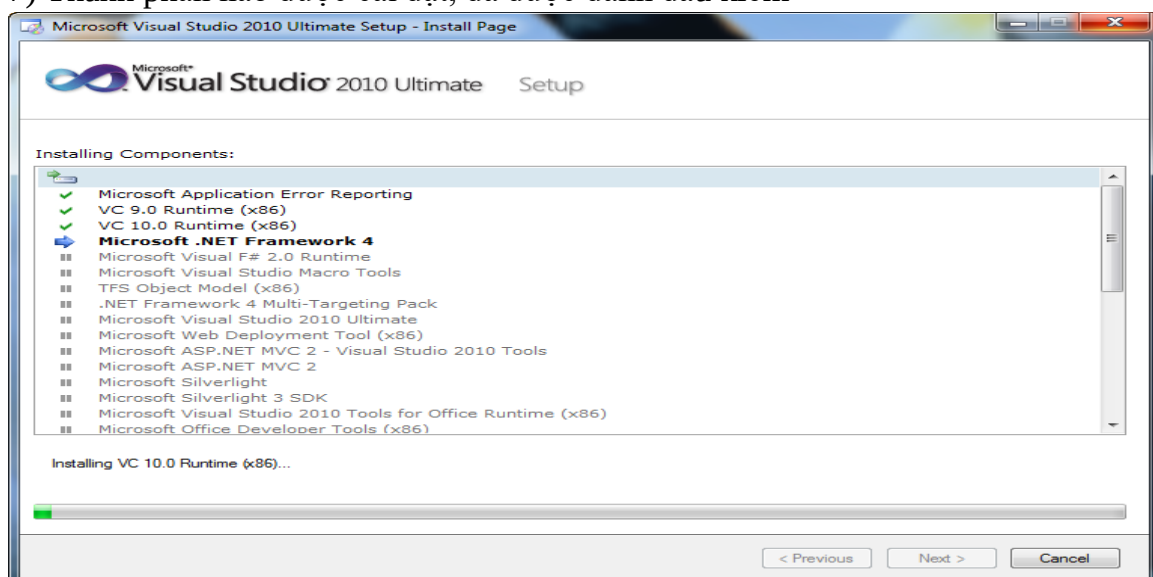
5) Chọn Full, Click Install



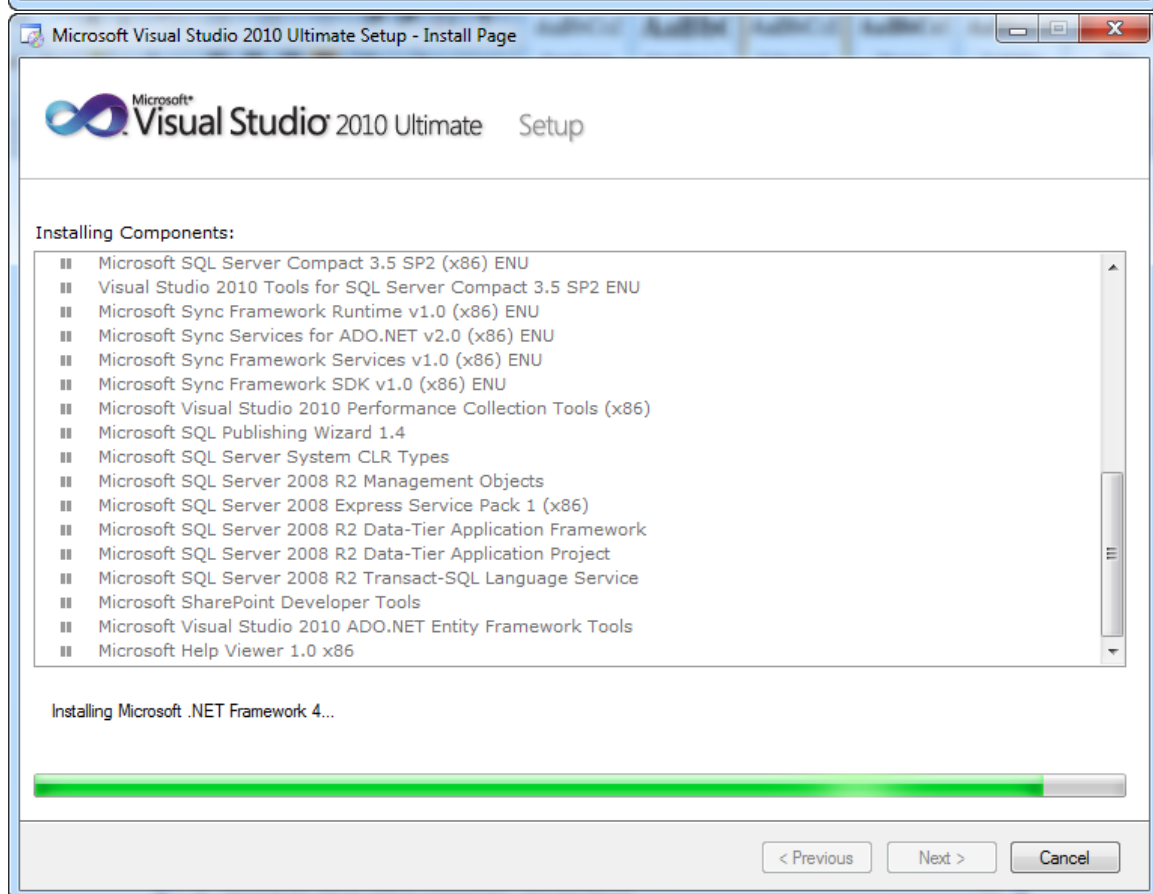
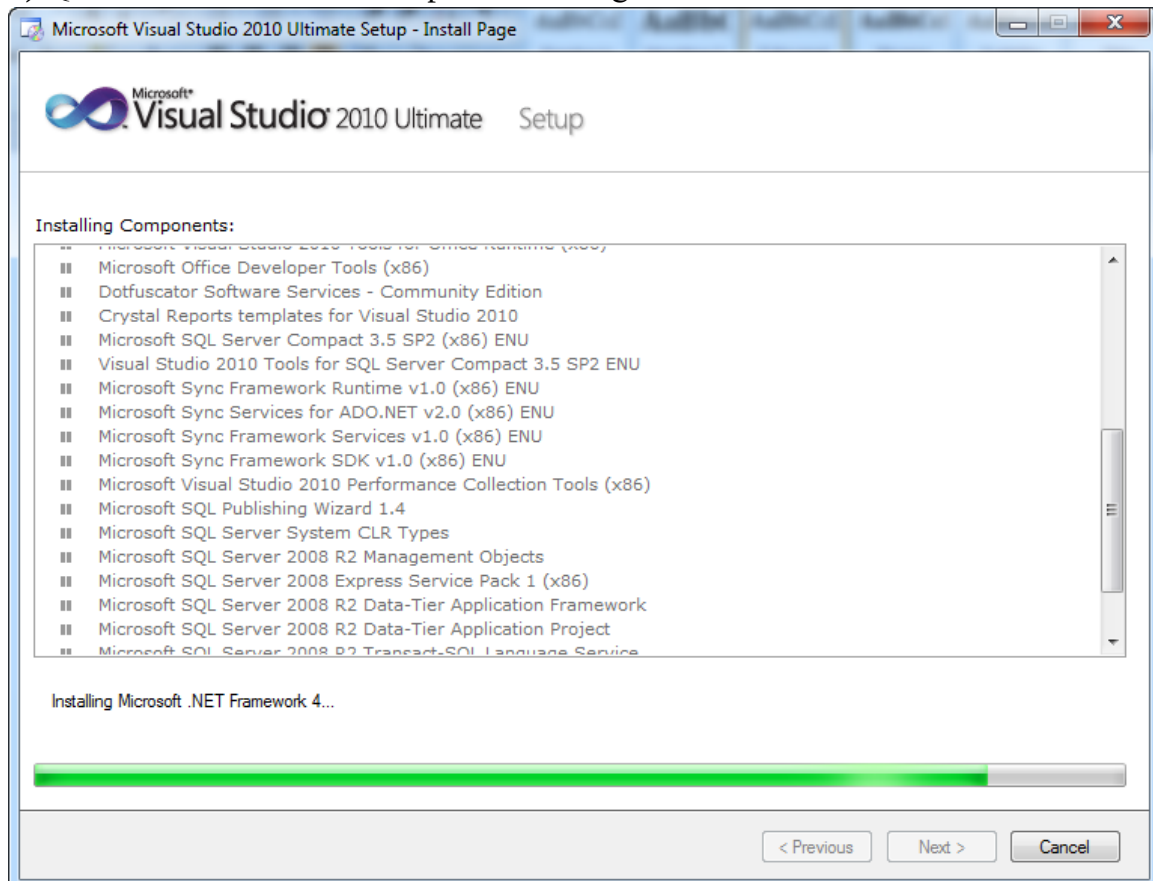
6) Cửa sổ các thành phần sẽ được cài đặt hiển thị như hình dưới



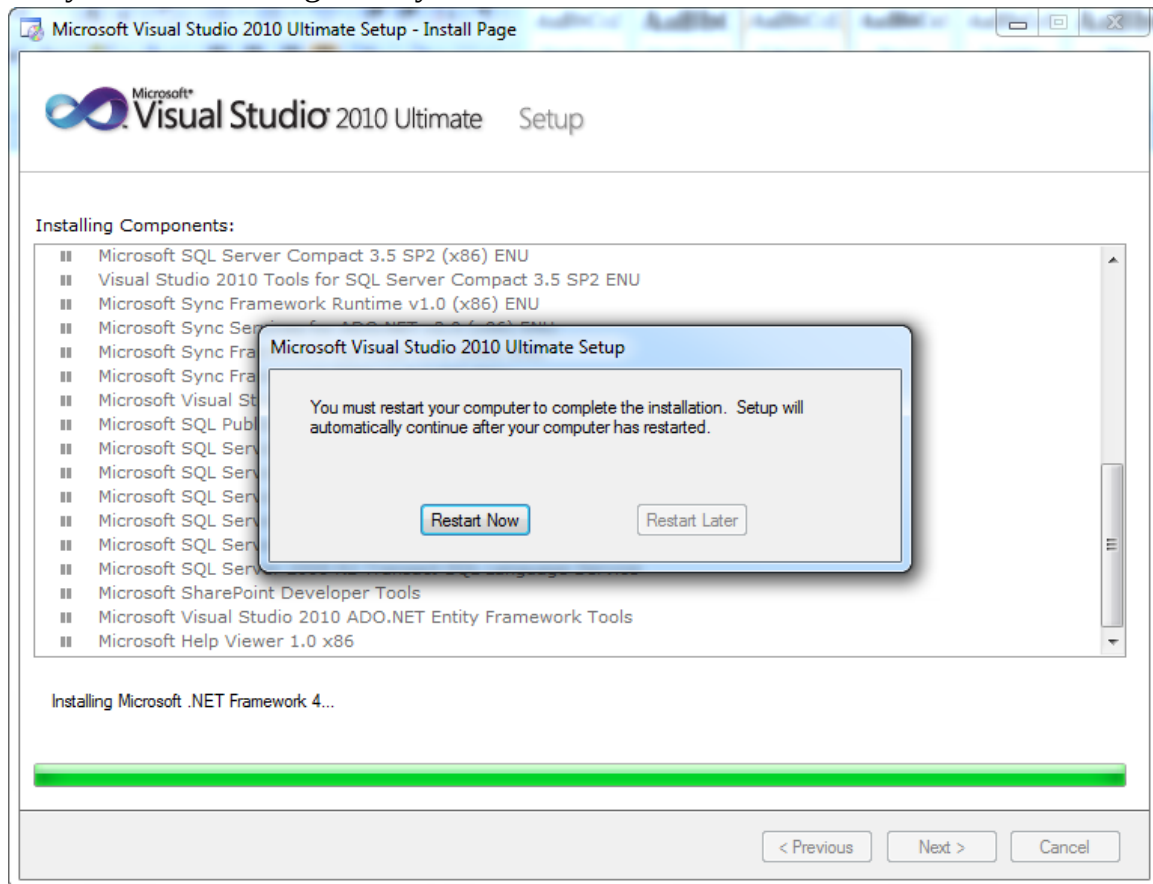
7) Thành phần nào được cài đặt, đã được đánh dấu kiểm



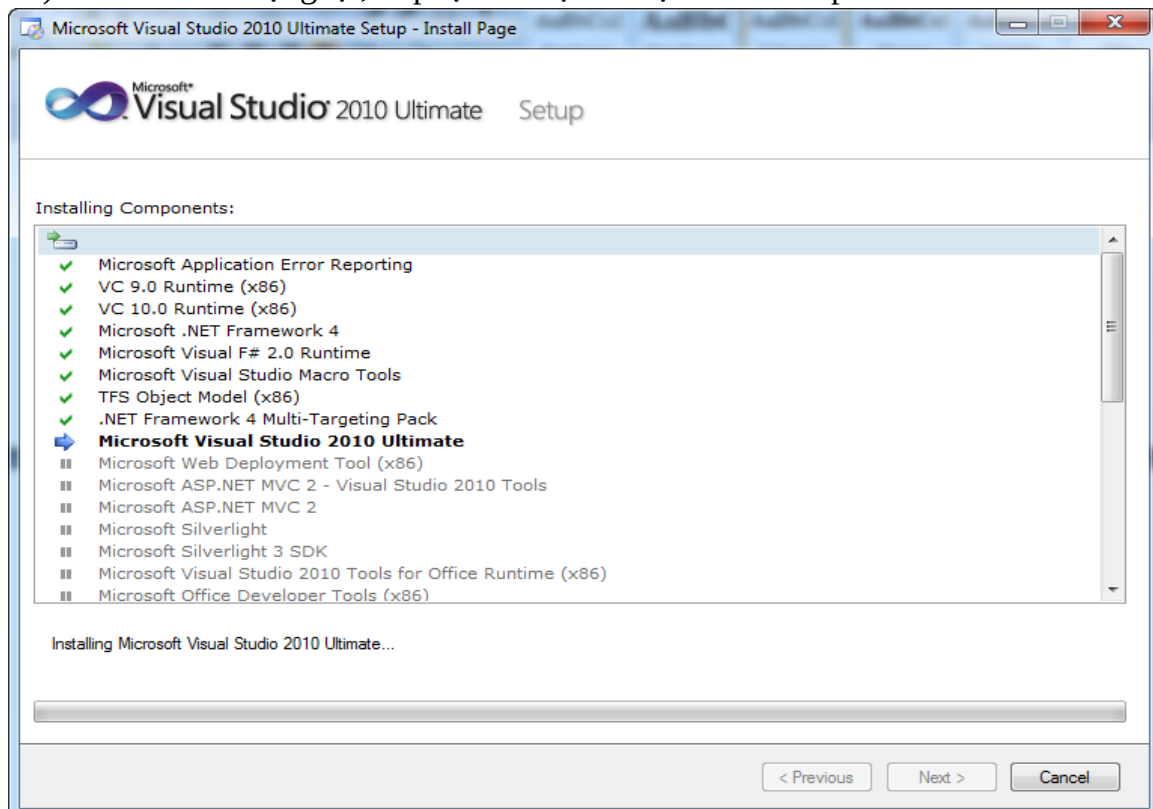
8) Quá trình cài đặt các thành phần vẫn đang diễn ra



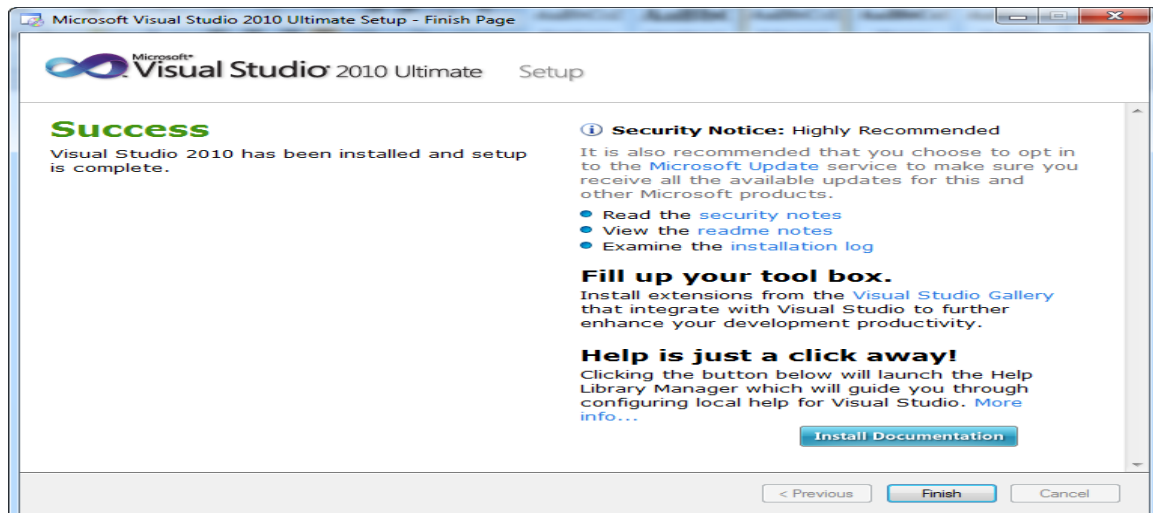
9) Một số thành phần sau khi cài đặt cần khởi động lại máy, khi đó xuất hiện 1 hộp thoại hiển thị yêu cầu khởi động lại máy tính, chọn Restart Now.



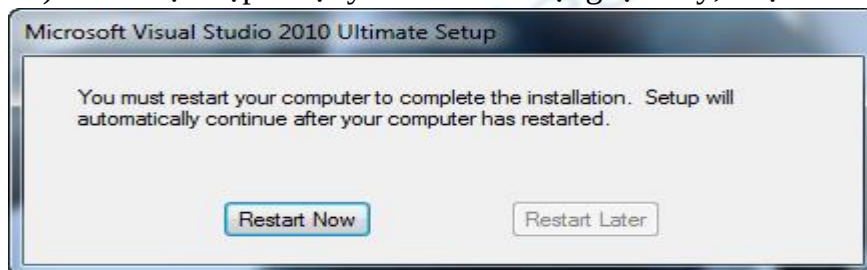
10) Sau khi khởi động lại, tiếp tục cài đặt cài đặt các thành phần khác



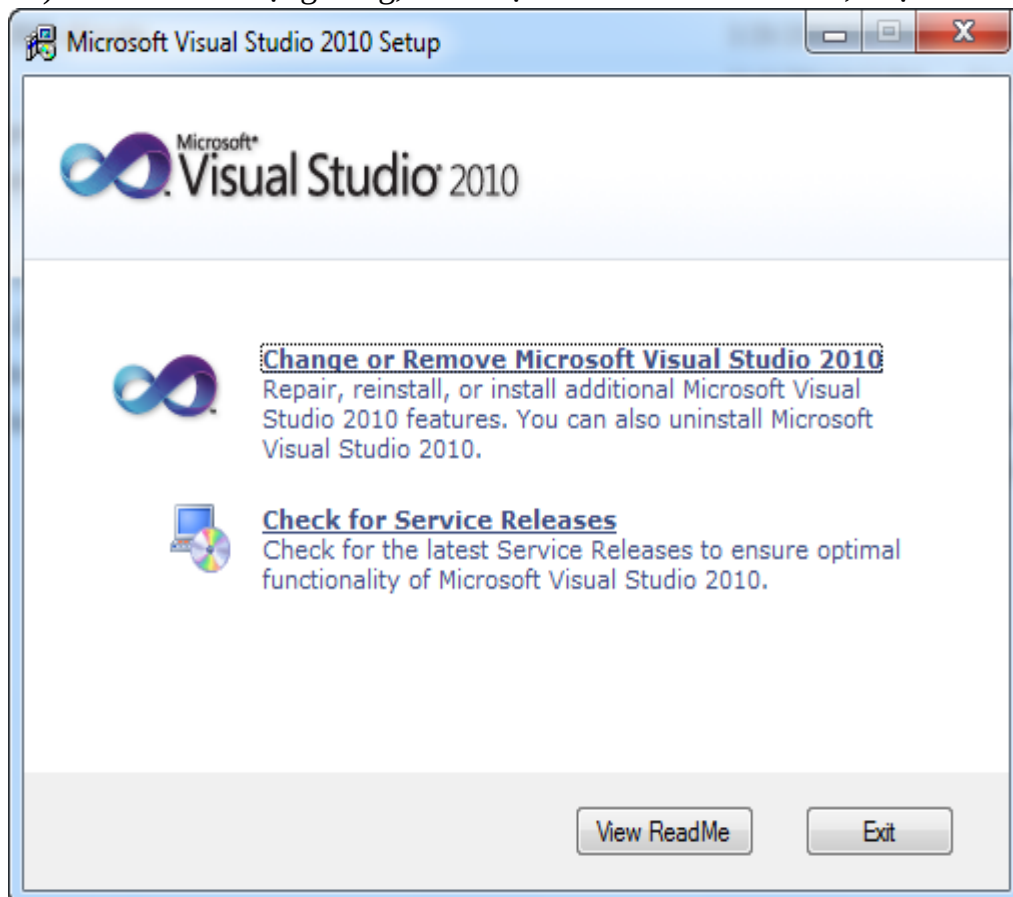
11) Sau khi các thành phần đã được cài đặt hết, hộp thoại hoàn thành cài đặt hiển thị như hình dưới, Click Finish



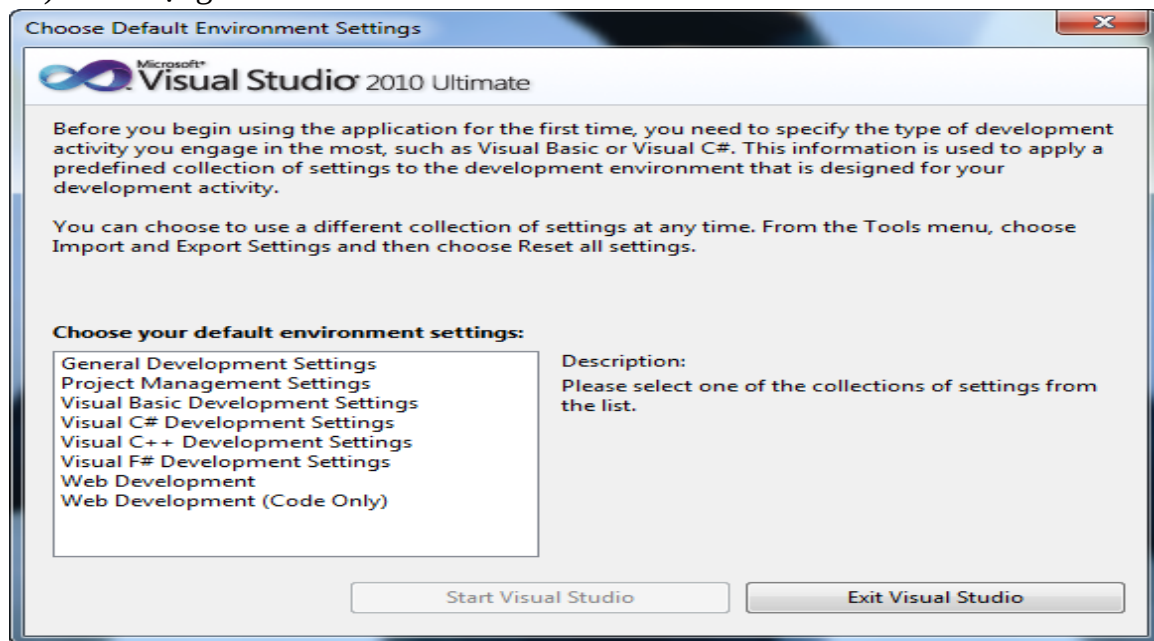
12) Xuất hiện hộp thoại yêu cầu khởi động lại máy, chọn Restart Now.



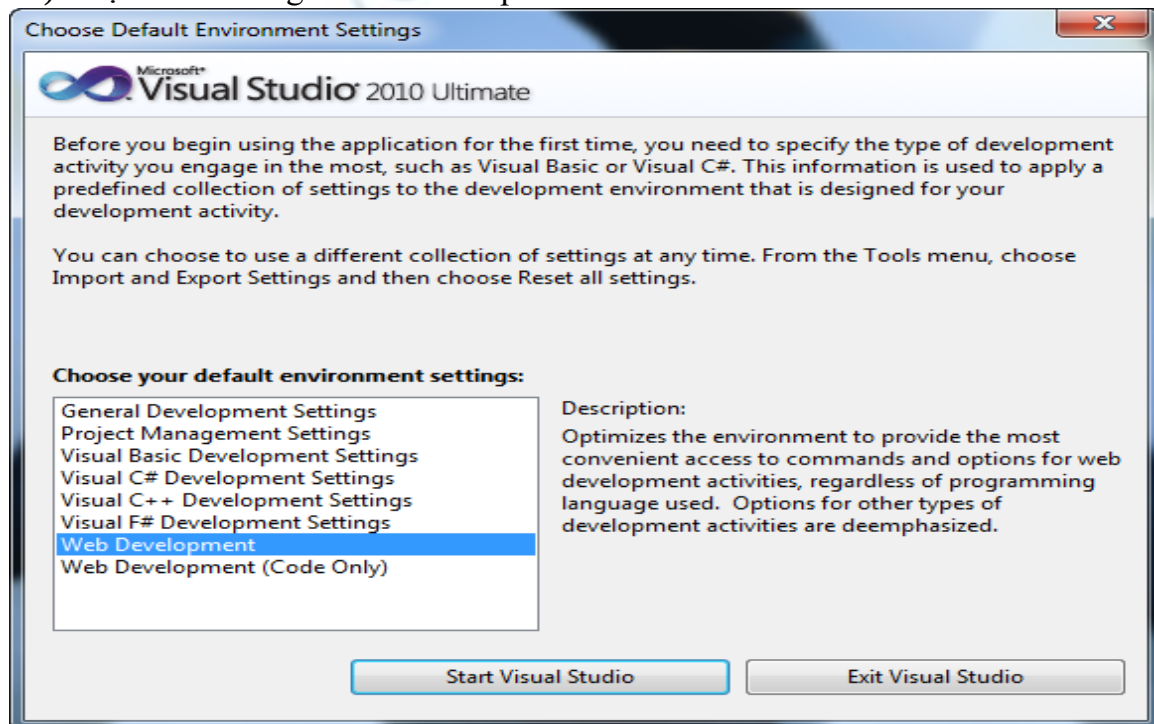
13) Sau khi khởi động xong, xuất hiện cửa sổ như hình dưới, chọn Exit.



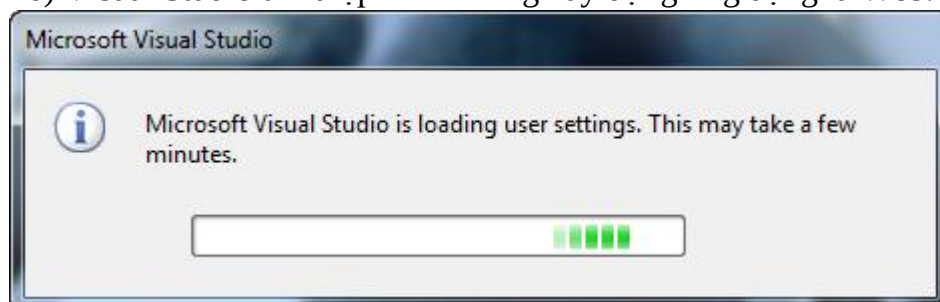
14) Khởi động Visual Studio 2010 Ultimate



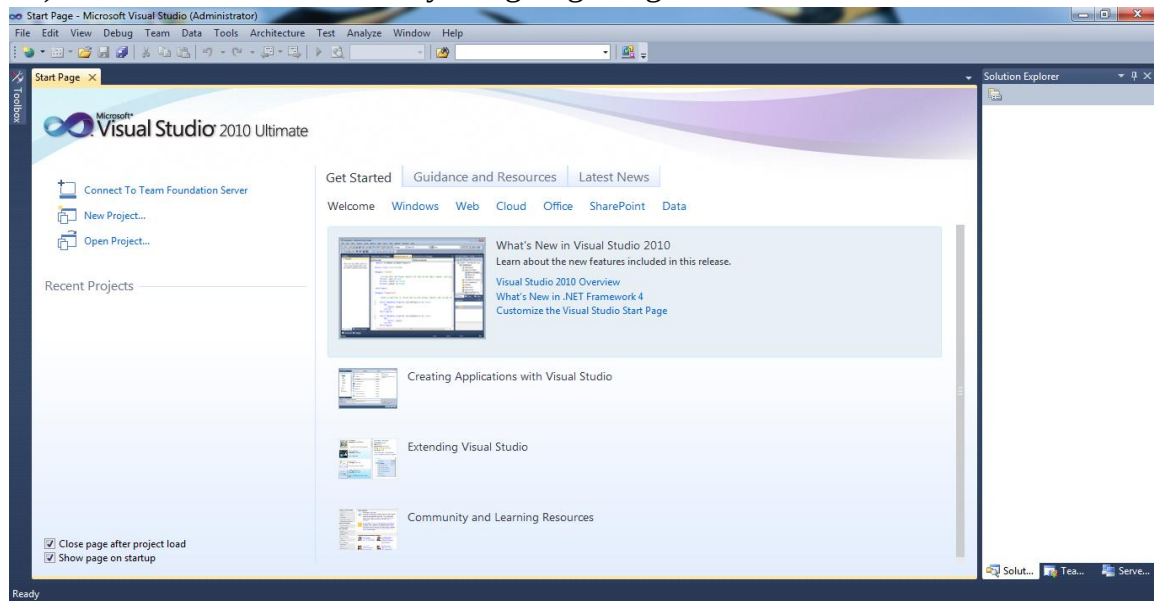
15) Chọn môi trường “Web Development” như hình dưới



16) Visual Studio thiết lập môi trường xây dựng ứng dụng là Web.



17) Màn hình chuẩn bị cho xây dựng ứng dụng Website.

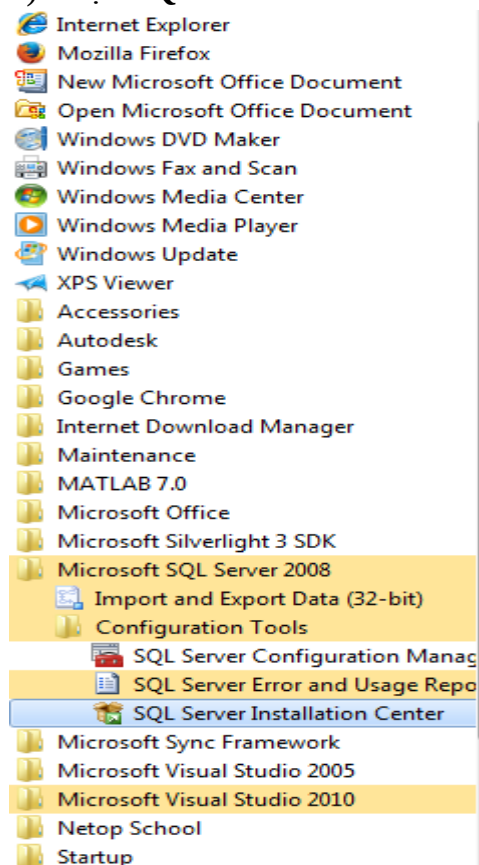


* Cài đặt SQL Server 2008 Express sau khi đã cài Visual Studio 2010 Ultimate (trên Win 7)

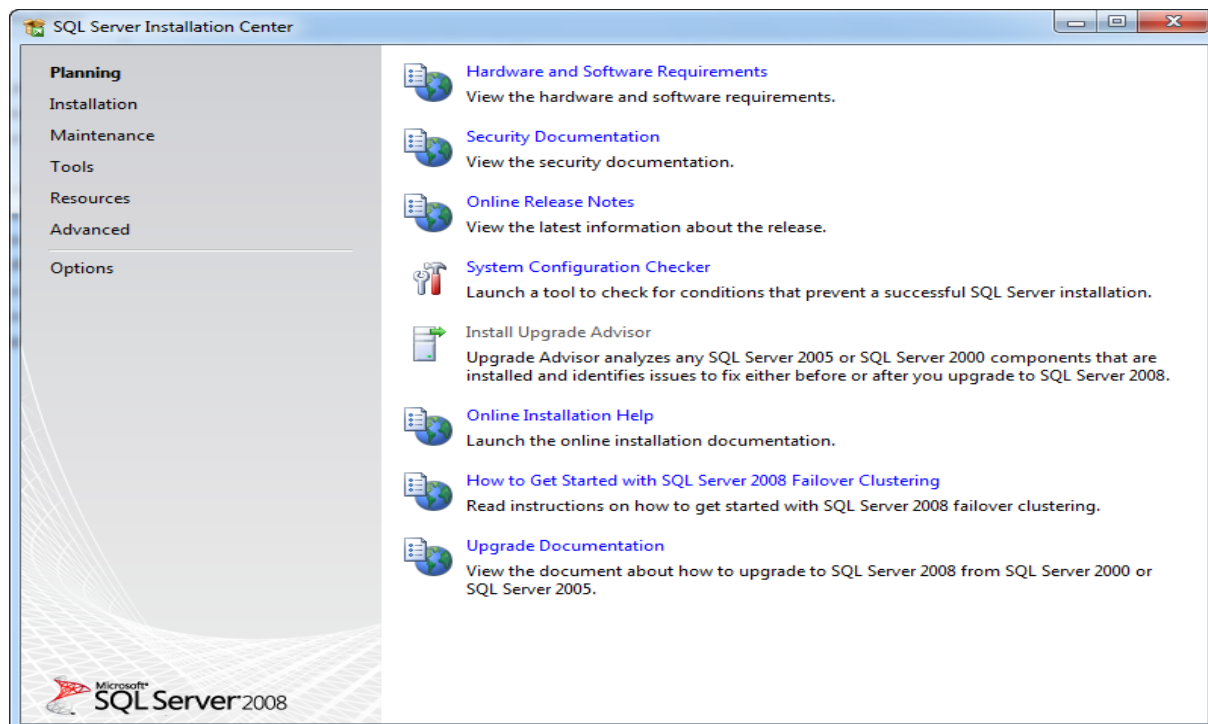
Chuẩn bị file cài đặt SQL Server 2008 Express (SQLEXPRAADV_x86_ENU.exe), rồi Extract ra thư mục cài đặt.

1) Cài Visual Studio 2010 Ultimate đã có sẵn SQL Server 2008, nhưng chưa dùng được, tiến hành cài SQL Server 2008 như sau:

2) Chọn **SQL Server Installation Center** để cài bắt đầu cài đặt như hình dưới:



3) Chọn SQL Server Installation Center như hình trên, xuất hiện màn hình cài đặt như hình dưới



4) Chọn mục Installation, để cài đặt chọn New SQL Server stand-alone installation or add features to an existing installation.

