

MỤC LỤC

1. ASP.NET là gì?	2
1.1 ASP.NET tích hợp với NET Framework	2
1.2 ASP.NET là đa ngôn ngữ.....	2
1.3 ASP.NET là hướng đối tượng (Object-Oriented)	2
1.4 ASP.NET được biên dịch.....	2
1.5 ASP.NET được lưu trữ trên máy bởi Common Language Runtime.....	3
1.6 ASP.NET dễ dàng triển khai và cấu hình	3
2. Các tính năng mới trong ASP.NET 3.5	3
2.1 ASP.NET AJAX	4
2.2 Các Control mới.....	4
2.3 LINQ	4
2.4 Các Assembly mới	4
2.5 ASP.NET Dynamic Data	4
2.6 ASP.NET MVC (Model View Control)	4
3. Cách tạo trang web trong Visual Studio 2008	5
3.1 Khởi động Visual Studio 2008 hoặc Visual Web Developer 2008 (hình 1).....	5
3.2 Cách tạo dự án mới	5
3.3 Ví dụ 1.....	7
4. Giới thiệu về sự kiện (Event) trong ASP.NET	9
4.1 Một số tệp cơ bản trong ASP.NET	9
4.2 Chu kỳ sống của trang web trong ASP.NET (ASP.NET Page Life Cycle).....	9
4.3 Một số các sự kiện của trang web (Page Event)	9
4.4 Sự kiện ứng dụng (Event Application)	10
5. Tài liệu tham khảo	11

GIỚI THIỆU TỔNG QUAN VỀ ASP.NET 3.5

Bài này giới thiệu tổng quan về công nghệ ASP.NET 3.5 (Active Server Page). ASP.NET là gì? Giới thiệu những tính năng mới của ASP.NET 3.5 so với ASP.NET 2.0. Cách tạo website trong Visual Studio 2008.

1. ASP.NET là gì?

Trước hết, tên đầy đủ của ASP.NET là Active Server Pages .NET (.NET ở đây là .NET framework). Nói đơn giản thì ASP.NET là một công nghệ có tính cách mạng dùng để phát triển các ứng dụng về mạng hiện nay cũng như trong tương lai. Bạn lưu ý ở chỗ ASP.NET là một phương pháp tổ chức hay khung tổ chức (framework) để thiết lập các ứng dụng hết sức hùng mạnh cho mạng dựa trên CLR (Common Language Runtime).

1.1 ASP.NET tích hợp với NET Framework

.NET Framework được chia thành bộ các tác vụ cho từng chức năng gồm các lớp (class), các cấu trúc (structures), các giao diện (interfaces) và các lõi (core) thành phần chương trình. Trước khi sử dụng thành phần nào bạn phải hiểu cơ bản về chức năng, các tổ chức của nó. Mỗi một trong hàng nghìn các tầng lớp được nhóm theo trình tự logic, thứ bậc được gọi là một **namespace**. Mỗi **namespace** cung cấp một tính năng.

1.2 ASP.NET là đa ngôn ngữ

Để xây dựng một ứng dụng web chúng ta không chỉ chọn một ngôn ngữ mà có thể chọn nhiều ngôn ngữ khác. Điều quan trọng là các ngôn ngữ chúng ta chọn mã của nó dịch được ra mã IL. Điều đó có nghĩa là IL là ngôn ngữ của .NET và chỉ có CLR nhận biết được IL.

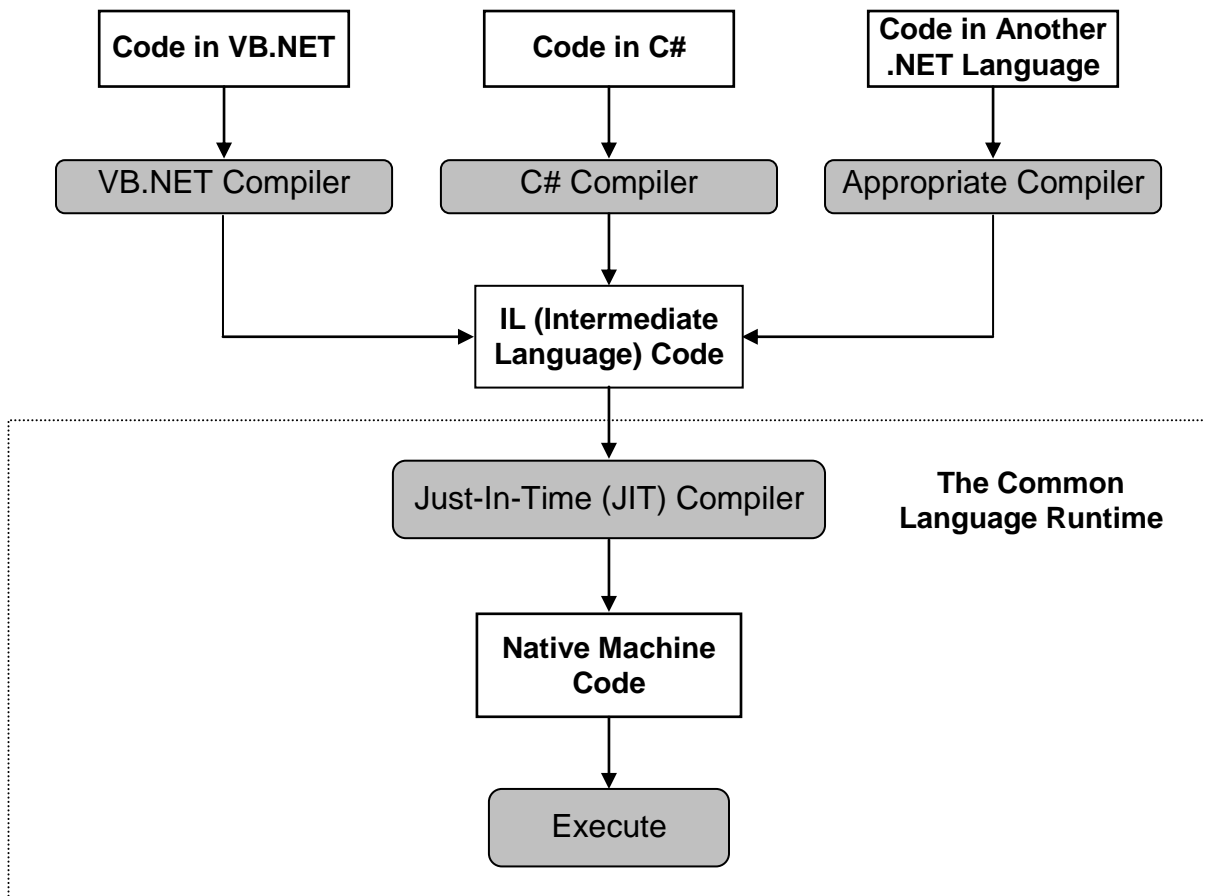
1.3 ASP.NET là hướng đối tượng (Object-Oriented)

ASP.NET là ngôn ngữ lập trình hướng đối tượng sử dụng các ngôn ngữ .NET. Nó cung cấp bộ các đối tượng nhỏ và thực sự là một tầng làm việc trên nguyên lý của HTTP và HTML. Mã nguồn của chúng ta không được truy cập toàn bộ các đối tượng trong .NET Framework nhưng có thể khai thác tất cả các quy ước của một môi trường OOP (Object Oriented Programming). Chúng ta có tạo các lớp, giao diện, kế thừa các lớp... Chúng ta có thể kiểm soát được các đối tượng trong chương trình như hiển thị dữ liệu và các sự kiện của đối tượng.

1.4 ASP.NET được biên dịch

Một ứng dụng ASP.NET luôn luôn được biên dịch, nó không chạy bằng mã của C# hoặc Visual Basic mà không được biên dịch trước. Một ứng dụng ASP.NET thực sự được biên dịch thông qua 2 giai đoạn:

- Giai đoạn đầu tiên những các mã (code) bạn viết (C#, Visual Basic hoặc ngôn ngữ .NET khác) được dịch bởi **Microsoft Intermediate Language (MSIL)**. Giai đoạn dịch này được dịch tự động khi trang web đầu tiên yêu cầu. Chúng ta có thể thực hiện dịch trước. Các tệp được dịch thành mã IL (Intermediate Language Code).
- Giai đoạn tiếp theo được dịch trước khi trang Web được thực thi. Tại giai đoạn này mã IL được dịch thành bản mã máy (Native Machine Code). Giai đoạn này được gọi là **Just-In-Time (JIT)**.



1.5 ASP.NET được lưu trữ trên máy bởi Common Language Runtime

Khía cạnh quan trọng nhất của công cụ ASP.NET là nó chạy trong môi trường thời gian thực (Runtime) của CLR (**Common Language Runtime**). CLR là máy ảo (virtual machine) trong Microsoft .NET, do có ngôn ngữ trung gian IL nên khi phát triển ứng dụng trên .NET, chúng ta không bị phụ thuộc vào thiết bị, có nghĩa là theo Microsoft nói thì ứng dụng .NET có thể chạy trên bất kỳ thiết bị nào có .NET Framework. Tất cả các namespace, các ứng dụng, các lớp trong bộ .NET Framework được gọi tắt là bộ quản lý mã. CLR cũng cung cấp các dịch vụ quan trọng khác như:

- Quản lý bộ nhớ
- Thu nhặt rác
- Quản lý các tuyến
- Xử lý ngoại lệ
- An toàn

1.6 ASP.NET dễ dàng triển khai và cấu hình

Mọi sự cài đặt .NET Framework đều cung cấp các lớp như nhau. Để triển khai ứng dụng ASP.NET chúng ta chỉ cần sao chép các tập tin vào thư mục ảo trên máy chủ (server) và máy chỉ cần có .NET Framework.

Việc cấu hình dễ dàng đơn giản không phụ thuộc vào IIS (Internet Information Services). Cấu hình trong ASP.NET được đặt trong tệp **web.config**. Tệp **web.config** được để cùng với thư mục chứa trang web của chúng ta. Tệp web.config không bao giờ bị khóa, chúng ta có thể truy cập bất kỳ lúc nào, việc sửa tệp này hoàn toàn dễ dàng vì chúng được lưu dưới dạng XML.

2. Các tính năng mới trong ASP.NET 3.5

Microsoft phát hành ASP.NET vào ngày 19 tháng 11 năm 2007 cùng với Visual Studio 2008. Đây là bước tiến hóa từ ASP.NET 2.0 tới ASP.NET 3.5. Nếu chúng ta đang sử dụng ASP.NET 2.0 để sử dụng ASP.NET 3.5 chỉ cần cài đặt thêm các tính năng mới của ASP.NET 3.5.

Trong phần này sẽ giới thiệu một số tính năng mới trong ASP.NET 3.5

2.1 ASP.NET AJAX

Trong ASP.NET 2.0, ASP.NET AJAX được cài đặt thêm vào. Tuy nhiên trong ASP.NET 3.5, ASP.NET AJAX đã được tích hợp vào trong .NET Framework, do đó quá trình xây dựng giao diện người dùng được dễ dàng và trực quan. ASP.NET AJAX Control Extenders đã được tích hợp Toolbox của Visual Studio 2008. Các tính năng của ASP.NET AJAX sẽ được trình bày ở bài sau.

2.2 Các Control mới

Các control **ListView** và **DataPager** có thể dùng mới kiểu dữ liệu mới **LinqDataSource**.

ListView linh hoạt hơn và có chứa toàn bộ tính năng của **GridView**, **Datagrid**, **Repeater** trong ASP.NET 2.0. Nó cung cấp các khả năng và chèn, xóa, sửa, sắp xếp, phân trang. Chúng ta hoàn toàn định dạng được việc dữ liệu hiển thị trên ListView mà không cần phải sử dụng thẻ <table>. Các Template trong ListView rất phong phú và đa dạng. **Datager** cung cấp cho ListView trong việc phân trang.

2.3 LINQ

LINQ (Language Integrated Query) đưa ra khả năng lập trình mới trong .NET. Giải pháp lập trình hợp nhất, đem đến khả năng truy vấn dữ liệu theo cú pháp SQL trực tiếp trong C# hay VB.NET, áp dụng cho tất cả các dạng dữ liệu từ đối tượng đến Cơ sở dữ liệu quan hệ và XML.

ASP.NET LinqDataSource cho phép chúng ta có thể sử dụng LINQ để lọc (filter), order và nhóm (group) dữ liệu trước khi binding vào List Control.

2.4 Các Assembly mới

- System.Core.dll – Các implementation cho LINQ to Objects
- System.Data.Linq.dll - Các implementation cho LINQ to SQL
- System.Xml.Linq.dll - Các implementation cho LINQ to XML
- System.Data.DataSetExtensions.dll - Các the implementation cho LINQ to DataSet
- System.Web.Extensions.dll: Các implementation for ASP.NET AJAX

2.5 ASP.NET Dynamic Data

ASP.NET Dynamic Data cung cấp một Framework cho phép chúng ta nhanh chóng xây dựng một chức năng ứng dụng driver-data, dựa trên LINQ to SQL hay Entity Framework. Nó cũng có thêm nhiều tính linh hoạt cho các DetailsView, FormView, GridView, ListView trong kiểm tra tính hợp lệ của dữ liệu, hoặc chỉnh sửa lại các mẫu để thay đổi cách hiển thị dữ liệu.

2.6 ASP.NET MVC (Model View Control)

Kiến trúc MVC là việc chia tất cả mục của một ứng dụng ra làm ba thành phần (component) khác nhau Model, View và Controller:

Model: Model được giao nhiệm vụ cung cấp dữ liệu cho cơ sở dữ liệu và lưu dữ liệu vào các kho chứa dữ liệu. Tất cả các nghiệp vụ logic được thực thi ở Model. Dữ liệu vào từ người dùng sẽ thông qua View được kiểm tra ở Model trước khi lưu vào cơ sở dữ liệu. Việc truy xuất, xác nhận, và lưu dữ liệu là một phần của Model.

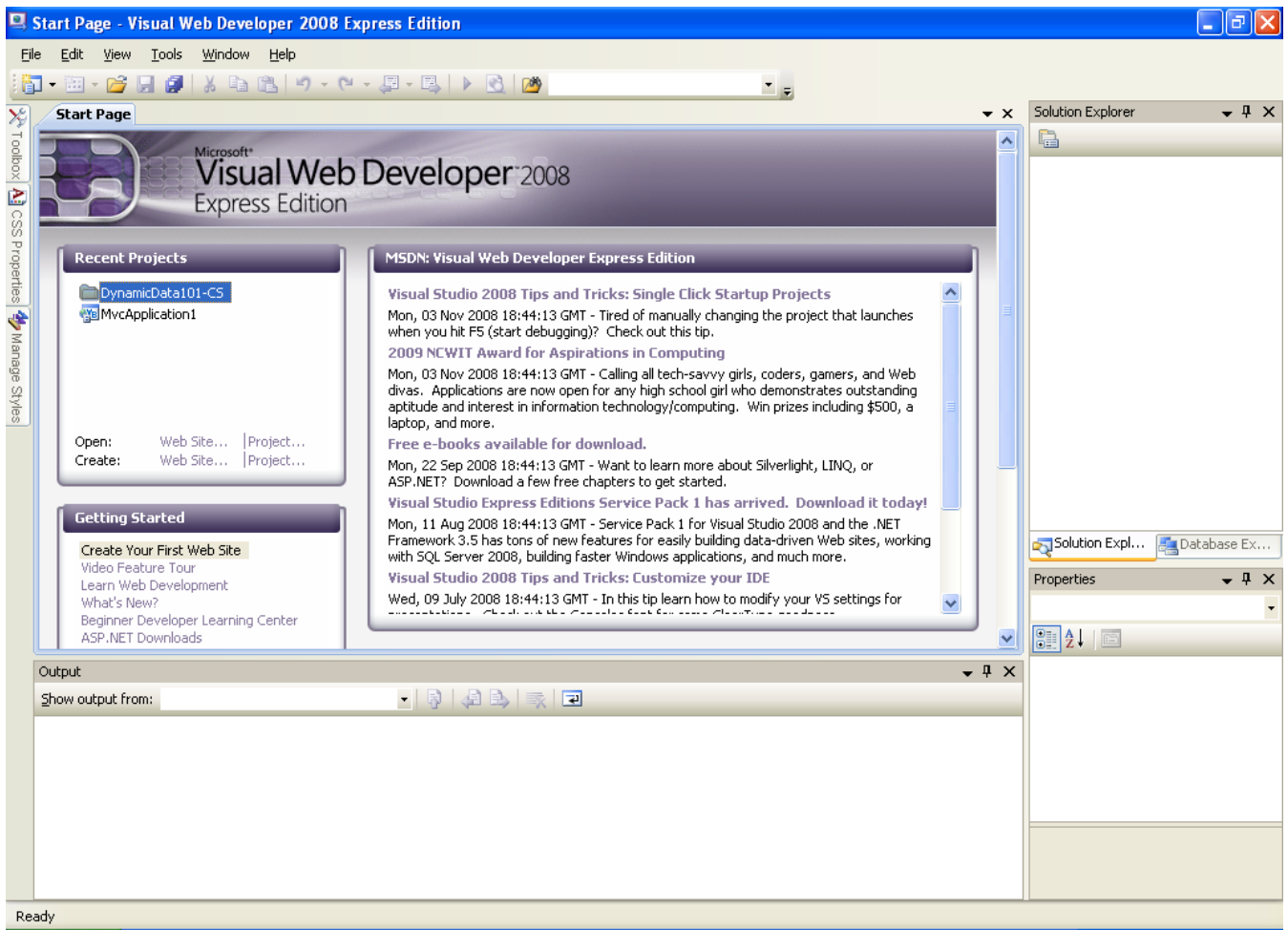
View: View hiển thị các thông tin cho người dùng của ứng dụng và được giao nhiệm vụ cho việc nhận các dữ liệu vào từ người dùng, gửi đi các yêu cầu đến bộ điều khiển, sau đó là nhận lại các phản hồi từ bộ điều khiển và hiển kết quả cho người dùng. Các trang HTML, JSP, các thư viện thể và các file nguồn là một phần của thành phần View.

Controller: Controller là tầng trung gian giữa Model và View. Controller được giao nhiệm vụ nhận các yêu cầu từ phía máy khách. Một yêu cầu được nhận từ máy khách được thực hiện bởi một chức năng logic thích

hợp từ thành phần Model và sau đó sinh ra các kết quả cho người dùng và được thành phần View hiển thị. ActionServlet, Action, ActionForm và struts-config.xml là các phần của Controller.

3. Cách tạo trang web trong Visual Studio 2008

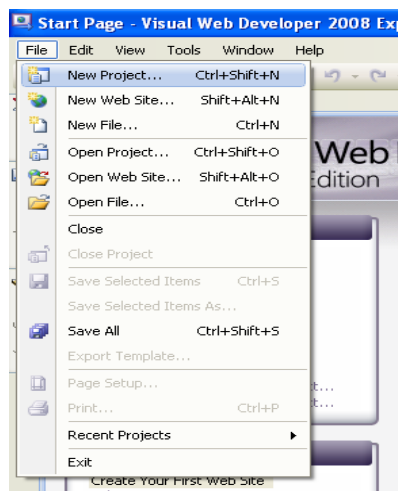
3.1 Khởi động Visual Studio 2008 hoặc Visual Web Developer 2008 (hình 1)



Hình 1: Visual Web Developer 2008

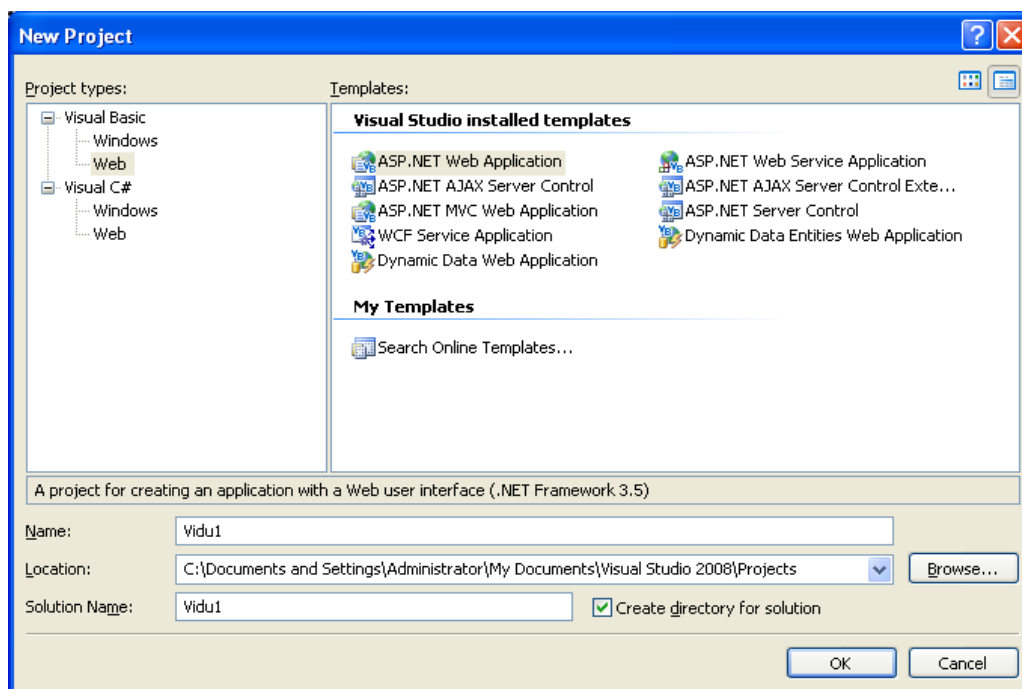
3.2 Cách tạo dự án mới

Chọn **File** → **New Project** (Ctrl+Shift+N) hoặc **New Web Site** (Shift+Alt+N) (Hình 2)



Hình 2: Tạo dự án mới

➤ Nếu chọn **New Project** xuất hiện hộp thoại **New Project** sau (hình 3):



Hình 3

Trong **Project Type**:

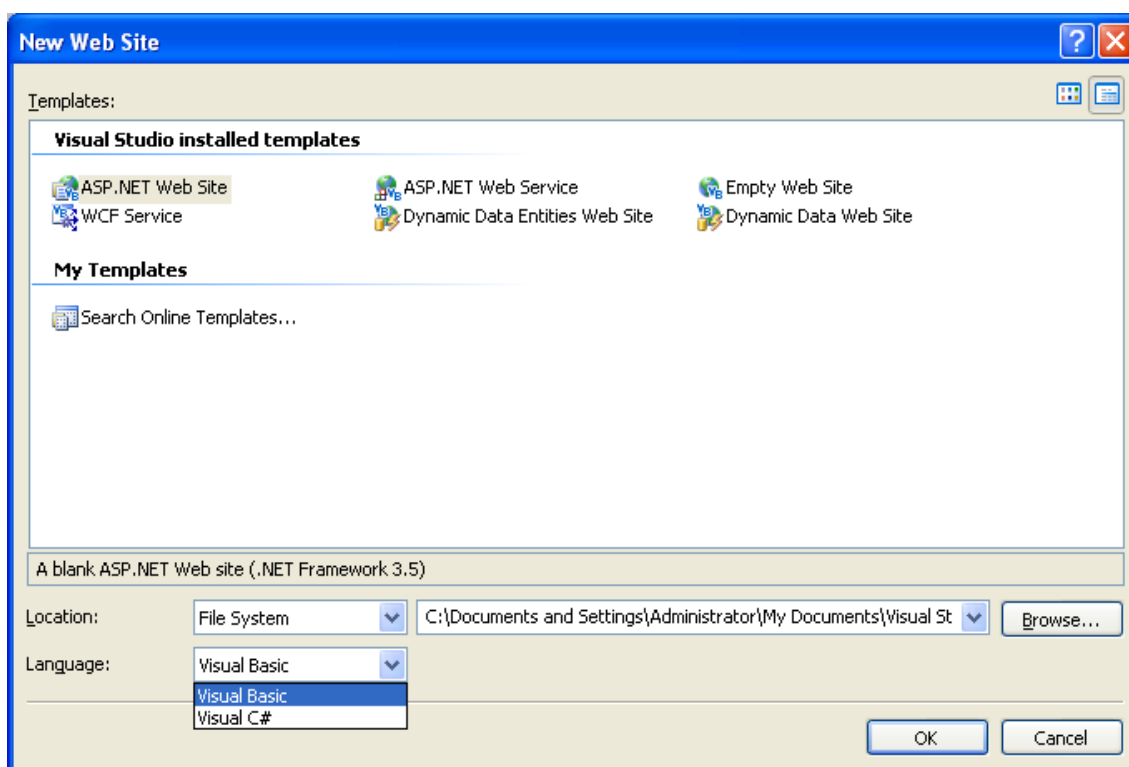
- **Visual Basic** → **Web** nếu chọn Visual Basic là ngôn ngữ viết ứng dụng.
- **Visual C#** → **Web** nếu chọn C# là ngôn ngữ viết ứng dụng.

Trong **Templates** chọn **ASP.NET Web Application**.

Trong **Name** đặt tên cho Project, tên do ta đặt.

Trong **Location** là thư mục chứa Project.

➤ Nếu chọn **New Web Site** xuất hiện hộp thoại **New Project** sau (hình 4):



Hình 4

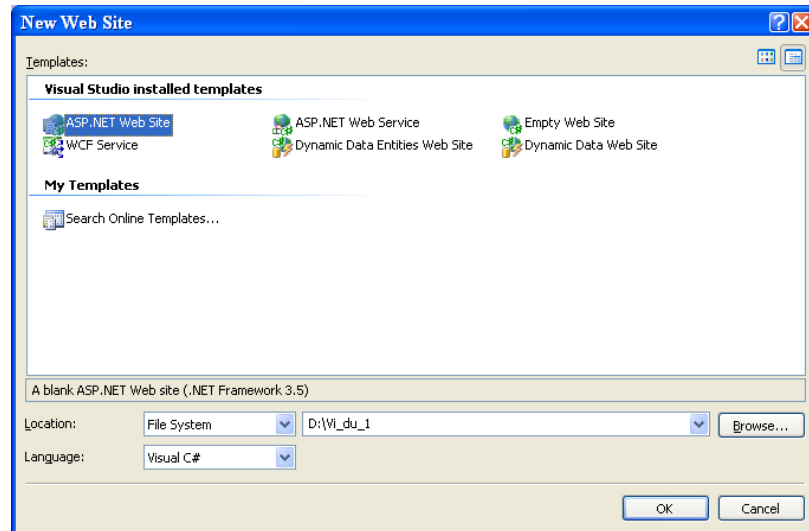
Trong **Language**:

- **Visual Basic** → **Web** nếu chọn Visual Basic là ngôn ngữ viết ứng dụng.
- **Visual C#** → **Web** nếu chọn C# là ngôn ngữ viết ứng dụng.

Trong **Templates** chọn **ASP.NET Web Application**.

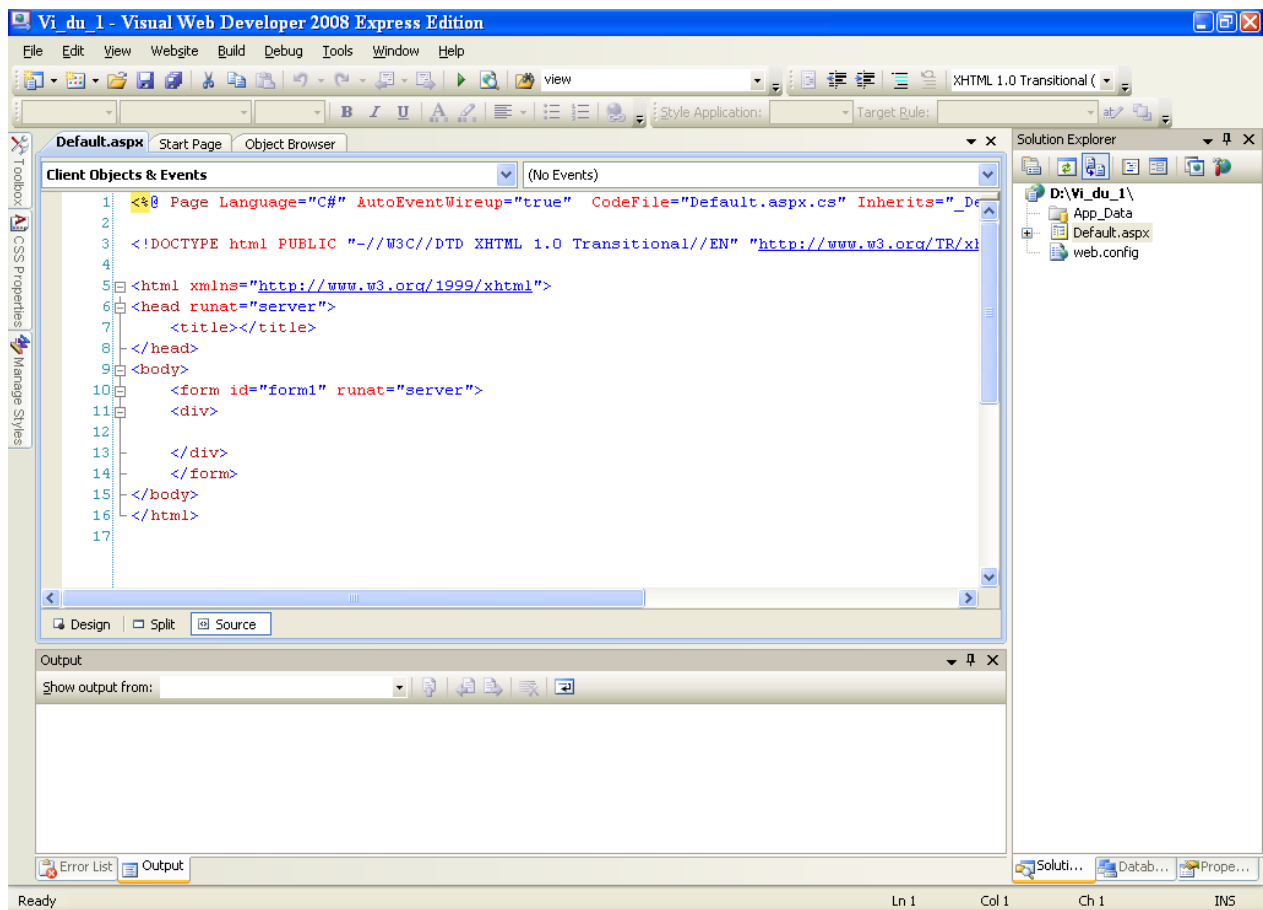
Trong **Location** là chọn System File và Chọn tên và thư mục chứa Project.

3.3 Ví dụ 1



Hình 5

Ví dụ tạo một dự án có tên là “**Vi_du_1**”, đặt tại ổ đĩa D, chọn ngôn ngữ chọn là C# (hình 5) sau đó bấm **OK**. Sau đó chúng ta được file “**Default.aspx**” (hình 6).



Hình 6

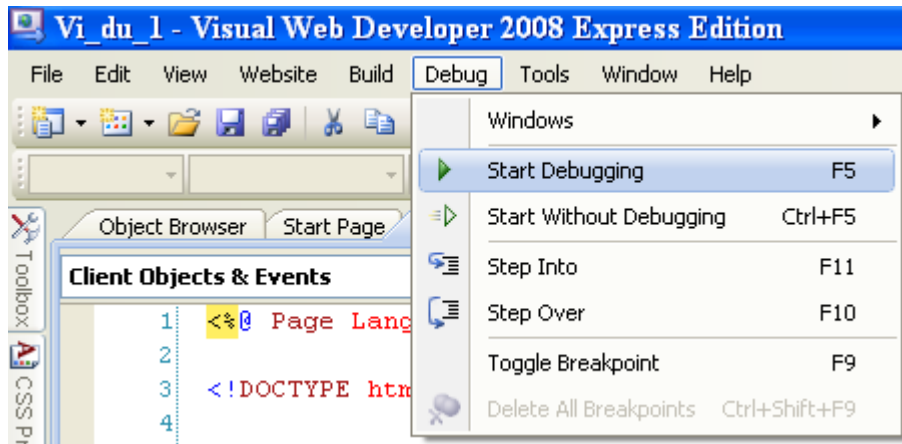
Trong ví dụ đầu tiên này chúng ta soạn một đoạn mã sau trong Default.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

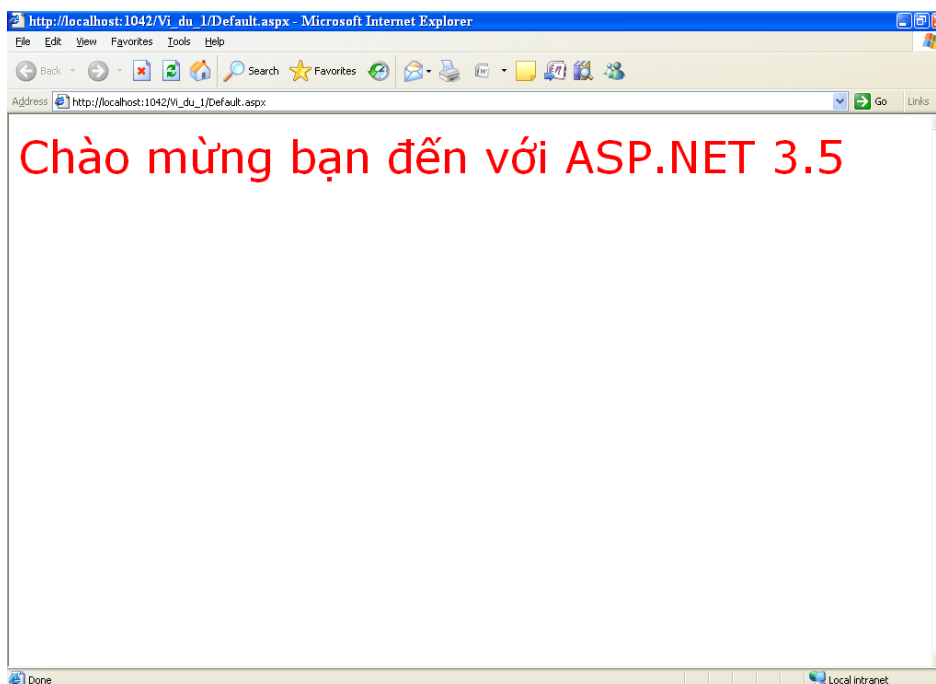
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <font face="verdana" color="red" size="20">Chào mừng bạn đến với ASP.NET
3.5 </font>
    </div>
  </form>
</body>
</html>
```

Để chạy web này dưới dạng Debug chúng ta chọn **Debug** → **Start Debugging** hoặc bấm phím **F5** (hình 7).



Hình 7

Kết quả khi chạy Vi_du_1 (hình 8)



Hình 8

4. Giới thiệu về sự kiện (Event) trong ASP.NET

4.1 Một số tệp cơ bản trong ASP.NET

TỆP	MIÊU TẢ
*.aspx	Tệp này chứa toàn bộ giao diện người dùng, các tùy chọn, các đoạn code ứng dụng. Đây là một trong những trang web đầu tiên được bắt đầu khi người sử dụng duyệt web.
*.cs	Tệp này chứa các mã nguồn của C#
web.config	Tệp tin dạng XML chứa cấu hình cho ứng dụng ASP.NET. Nó bao gồm các cấu hình về an ninh, quản lý trạng thái, quản lý bộ nhớ...
global.asax	Chúng ta sử dụng tập tin này để khai báo các biến toàn cục và phản ứng với các sự kiện toàn cục.

Khi chúng ta xây dựng xong một ứng dụng Web:

- Phần giao diện của ứng dụng nằm trong ở các tệp .aspx và html.
- Tất cả các tệp mã nguồn (C#, VB.NET hoặc ngôn ngữ .NET) sẽ được biên dịch thành tệp .DLL và nằm trong thư mục /Bin.

4.2 Chu kỳ sống của trang web trong ASP.NET (ASP.NET Page Life Cycle)

- Chu kỳ sống được bắt đầu khi trình duyệt yêu cầu một trang web gọi là **Session**.
- Chu kỳ sống vẫn tiếp tục nếu:
 - **Session** đang hoạt động.
 - Người sử dụng tương tác với giao diện web cho đến khi kích hoạt một sự kiện.
 - Dữ liệu của trang (View State) web được gửi về cho Server.
 - Server nhận được View State và trả lại yêu cầu từ View State.
- Chu kỳ sống kết thúc khi:
 - Người dùng kết thúc trình duyệt.
 - Session kết thúc (timeout).

4.3 Một số các sự kiện của trang web (Page Event)

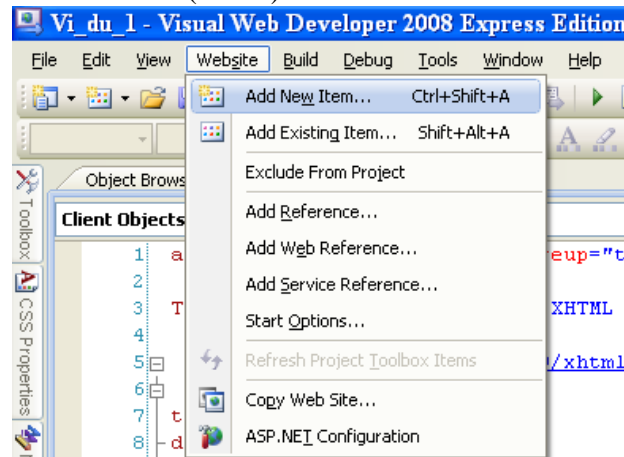
SỰ KIỆN	MIÊU TẢ
Pre_Init	Kiểm tra thuộc tính IsPostBack khi lần đầu tiên trang web đang được sử lý. Tạo mới hoặc tạo mới lại các Control động (Dynamic Control). Thiết lập trang chủ (Master page), thuộc tính Theme. Đọc và thiết lập giá trị cho thuộc tính Profile.
Init	Làm nổi (Raised) sau khi các Control đã được khởi động và các Skin đã được ứng dụng. Chúng ta sử dụng sự kiện này để đọc hoặc khởi tạo các thuộc tính của Control.

Load	Sử dụng sự kiện này để thiết lập thuộc tính cho Control, thiết lập các kết nối với Cơ sở dữ liệu.
InitComplete	Sử dụng sự kiện này để xử lý các tác vụ có yêu cầu khi tất cả các khởi tạo hoàn thành.
Unload	Trang được giải phóng khỏi bộ nhớ
Error	Khi xảy ra lỗi trong trang.
DataBinding	Server Control trên trang được gắn với nguồn dữ liệu

4.4 Sự kiện ứng dụng (Event Application)

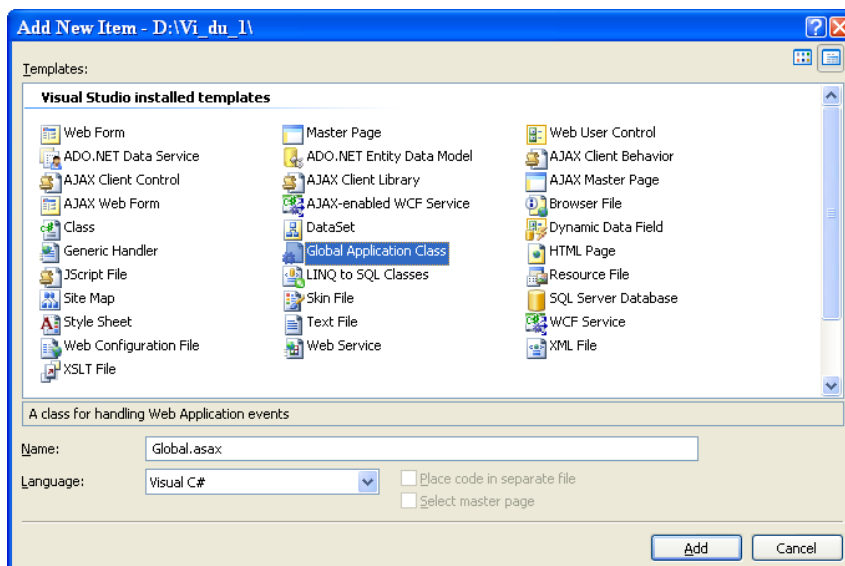
SỰ KIỆN	MIÊU TẢ
Application_Start	Người dùng đầu tiên duyệt trang web.
Application_End	Khi không còn người dùng nào duyệt trang web.
Application_Error	Khi có lỗi xảy ra trong ứng dụng
Session_Start	Khi người dùng duyệt một trang web
Session_End	Khi người dùng đóng trình duyệt hoặc Session kết thúc (time out)

Ví dụ sau đếm số người truy cập và số người online
 Chọn Project → Add New Item (hình 9).



Hình 9

Trong “Add New Item” chọn “Global Application Class” (hình 10).



Hình 10

Trong file **Global.asax** soạn đoạn mã sau:

```
<%@ Application Language="C#" %>
<%@ Import Namespace="System.IO" %>
<script runat="server">

    void Application_Start(object sender, EventArgs e)
    {
        if (!File.Exists("D:\\Dem.txt"))
            File.WriteAllText("D:\\Dem.txt", "0");
        Application["SNTruyCap"] = int.Parse(File.ReadAllText("D:\\Dem.txt"));
    }

    void Application_End(object sender, EventArgs e)
    {
    }

    void Application_Error(object sender, EventArgs e)
    {
    }

    void Session_Start(object sender, EventArgs e)
    {
        if (Application["SNOnline"] == null)
            Application["SNOnline"] = 1;
        else
            Application["SNOnline"] = (int)Application["SNOnline"] + 1;

        Application["SNTruyCap"] = (int)Application["SNTruyCap"] + 1;
        File.WriteAllText("D:\\Dem.txt", Application["SNTruyCap"].ToString());
    }

    void Session_End(object sender, EventArgs e)
    {
        Application["SNOnline"] = (int)Application["SNOnline"] - 1;
    }
}
</script>
```

5. Tài liệu tham khảo

1. Visual Web Developer, URL: <http://msdn.microsoft.com/en-us/library/ms178093.aspx>
2. Microsoft ASP.NET, URL: <http://www.asp.net/>
3. Bill Evjen, Scott Hanselman, Devin Rader, “*Professional ASP.NET 3.5 In C# And VB*”