

BÀI 3: XÂY DỰNG WEB ĐỘNG

Mã bài: MĐQTM23-03

Giới thiệu:

Bài học này giới thiệu một trong các cách lập trình tạo web động (web có tương tác với cơ sở dữ liệu động) sử dụng đối tượng ASP.Net, ADO.Net trên môi trường Visual Studio.Net.

Mục tiêu:

- Cài đặt, cấu hình được IIS và ngôn ngữ lập trình Web.
- Sử dụng thành thạo môi trường phát triển và lập trình web động.
- Sử dụng tốt các đối tượng ASP.Net, ADO.Net, tiện ích của ngôn ngữ lập trình.
- Có khả năng kết nối với cơ sở dữ liệu.
- Thực hiện tốt công tác an toàn lao động và vệ sinh phân xưởng.

Nội dung chính:

1. Tổng quan về ASP.Net và ADO.Net:

Mục tiêu: Giới thiệu tổng quan về công nghệ phát triển web động bằng Asp.Net và ADO.Net, giới thiệu về nền tảng .Net Framework, cách cài đặt, cấu hình máy chủ web ảo IIS. Giới thiệu môi trường lập trình web – bộ Visual Studio .Net.

1.1. Tổng quan về ngôn ngữ ASP.Net:

1.1.1. Giới thiệu ASP.Net (Active Server Page. Net):

a) Các mô hình code trong trang ASP.NET

Một trang Web bao gồm hai phần Markup Code và Programming code. Khi duyệt một trang ta thấy phân layout của nó trên trình duyệt. Phần layout được thiết kế bởi các thẻ markup. Programming code được viết cho sự điều khiển sự tương tác của người sử dụng và một số sự kiện.

- Markup file chỉ dẫn cho trình biên dịch cách biên dịch và xử lý trang ASP.NET.

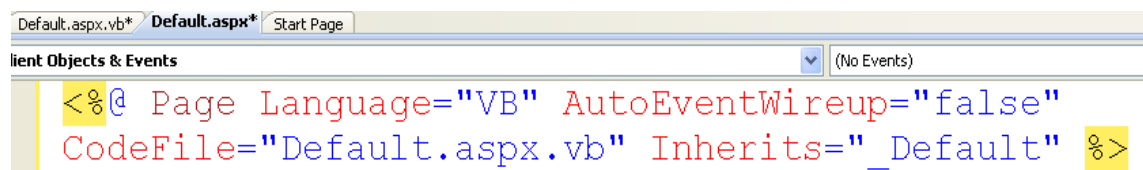
- Programming code bao gồm các kỹ thuật xử lý sự kiện khác nhau.

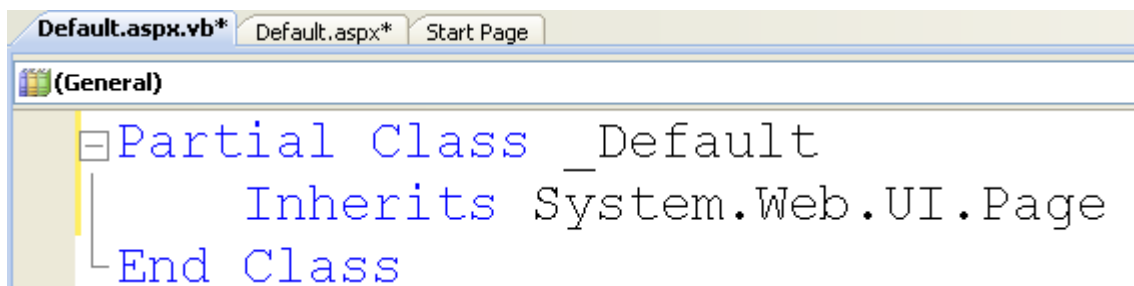
Markup code và Programming code có thể ở trên cùng một trang hoặc trên các trang khác nhau. Dựa vào đó người ta chia Web Page ra làm hai dạng:

- Single File Page: chứa cả Markup code và Programming code trên cùng một trang.

- Code Behind Page: IDE sẽ tạo ra hai file Markup (có phần mở rộng là .aspx chứa markup code) và Class (phần mở rộng là .aspx.cs, .aspx.vb,.. chứa logic lập trình. Markup file chứa tham chiếu đến Code Behind file nên 2 file này được liên kết với nhau.

File Code Behind tự động tạo Partial Class. Từ khóa Partial chỉ định lớp này không chứa đầy đủ các thành phần mà nó được kế thừa từ lớp Page.





Lớp Page: dẫn xuất từ lớp Page thuộc namespace "System.Web.UI", được kế thừa các thuộc tính, các phương thức và các sự kiện của lớp này.

<i>Thuộc tính</i>	<i>Mô tả</i>
ID	Chỉ định hoặc truy xuất định danh đối tượng của lớp Page.
Title	Chỉ định hoặc truy xuất tiêu đề của trang
Server	Truy xuất đến thể hiện của lớp Server
Session	Truy xuất đến thể hiện của lớp Session đối với Session hiện tại.
Controls	Truy xuất đến thể hiện của lớp ControlCollection cho tất cả các điều khiển Server Controls.
ErrorPage	Chỉ định hoặc truy xuất đến trang Error mà trang này được điều hướng đến khi có lỗi xảy ra.

<i>Phương thức</i>	<i>Mô tả</i>
HasControls	Kiểm tra xem có tồn tại các controls trên trang hay không.
LoadControl	Load một thể hiện của lớp Control
GetValidators	Trả về tập các đối tượng Validator liên quan đến nhóm Validation chỉ định.
MapPath	Trả về đường dẫn ảo ánh xạ đến.

Trong đó cấu trúc cơ bản của file .aspx cho cả hai dạng bao gồm các chỉ dẫn trang, script code và user interface code.

b) Các chỉ dẫn (Directives)

Các chỉ dẫn là các câu lệnh mô tả ứng dụng ASP.NET được biên dịch như thế nào. Các chỉ dẫn được bao trong cặp <%%> và thuộc file .aspx. Các chỉ dẫn tốt nhất nên khai báo ở vị trí đầu tiên trên trang. Sau đây là một số chỉ dẫn:

<i>Chỉ dẫn</i>	<i>Mô tả</i>
@Page	Định nghĩa các thuộc tính cho Web Page
@Import	Import một tên miền cho trang
@Assembly	Liên kết một Assembly với trang hoặc với một điều khiển
@Master	Định nghĩa các thuộc tính của Master.
@Reference	Liên kết một trang, một điều khiển tới trang hiện tại
@Register	Khai báo một điều khiển của người dùng.

- **Chỉ dẫn @Page:** Page là chỉ dẫn các thuộc tính của trang. Mỗi thuộc tính

được gán một giá trị, theo đó mà trang ASP.NET sẽ được biên dịch.

Thuộc tính	Diễn giải	Ví dụ
Language	Là thuộc tính chỉ định ngôn ngữ cho quá trình biên dịch code	<%@ Page Language="VB"%>
CodeFile	Là thuộc tính chỉ định đường dẫn file Code Behind.	<%@ Page Language="C#" Page CodeFile= "Sample.aspx.cs"
Inherits	Là thuộc tính chỉ định lớp Code Behind và thường đi cùng với thuộc tính CodeFile	
ErrorPage	Là thuộc tính chỉ định URL khi có lỗi xảy ra trên trang.	<%@ Page
Title	Là thuộc tính chỉ định tiêu đề cho trang	Language="C#" Page ErrorPage= "ErrorPage.aspx"
AutoEventWireUp	Là thuộc tính cho phép tự động buộc các sự kiện tương ứng đến file .aspx hoặc file Code Behind, nó nhận giá trị Boolean, mặc định là true	Title="Ví dụ" AutoEventWireUp ="True" %>

Tổng quát: Ta có cách khai báo chỉ định @Page:

<%@ Page <attribute>="<value>" %>

- **Chỉ dẫn @Import:** khai báo thêm các không gian tên miền vào Page. Khi khai báo chỉ dẫn @Import, tất cả các Class của không gian tên sẽ sẵn sàng trong khi lập trình.

Để khai báo chỉ dẫn @Import, bạn có thể sử dụng cú pháp sau:

<%@ import Namespace="System.Threading" %>

<%@ import Namespace="System.IO" %>

Mỗi không gian tên bạn sử dụng một khai báo chỉ dẫn @Import. Nếu có nhiều không gian tên cần thêm vào Page bạn sẽ sử dụng nhiều khai báo @Import.

Tuy nhiên, bạn chỉ cần khai báo @Import với các không gian tên mà .NET Framework tùy chọn. Điều này có nghĩa là những không gian tên sau đây tự động thêm vào trang.

System System.Collection.Specialized System.IO

System.Text.RegularExpression

System.Collections System.Configuration System.Text System.Web

System.Web.Security System.Web.UI

System.Web.UI.WebControls

System.Web.SessionState

System.Web.UI.HtmlControls

c) Điều khiển các sự kiện trong Web pages:

Mỗi một Web page có một vòng đời, mỗi bước trong vòng đời xuất hiện một sự kiện. Các sự kiện được xử lý tự động bởi các Event Handler tương ứng. Sau đây là một số sự kiện tự động điều khiển trong Webpages: Init -> Load -> PreRender->Unload.

- Init: Là sự kiện đầu tiên trong vòng đời của trang web. Thường dùng khởi tạo tất cả các điều khiển trong trang web. Init cũng còn được dùng đối với các biến cần được khai báo và khởi tạo trước khi xử lý Web page.

- Load: Là sự kiện xảy ra sau sự kiện Init. Thực hiện bất cứ khi nào Web Page được Request. Khi một người dùng Request một trang Web thì các biến và các điều khiển trong trang được khởi tạo và sau đó trang đó được load lên.

- PreRender: Là sự kiện được thực hiện trước khi trang được submit (đệ trình) cho người sử dụng.

- UnLoad: Là sự kiện xảy ra sau khi trang được submit cho người dùng.

Chú ý: Trong tiến trình xử lý các sự kiện của trang liên quan đến một khái niệm đó làPostBack.

- PostBack là thông tin được submit từ Browse đến Server. Khi người dùng Submit thông tin bằng cách click vào một nút lệnh hoặc chọn một item trên lưới thì Browse sẽ chuyển thông tin này đến server.

Khi người dùng tương tác với Web Page bằng cách phát sinh một sự kiện thì trang đó được gửi quay trở lại Server. Server xử lý code hiện tại trong các event handlers và sau đó trang này được tạo lại. Như vậy trang này lại trải qua một vòng đời mới với các sự kiện Init() và Load ()

- Thuộc tính IsPostBack dùng để kiểm tra Web Page được Request lần đầu tiên hay là kết quả của một PostBack. Nếu trang đó được Request lần đầu tiên thì thuộc tính IsPostBack nhận giá trị false. Nếu trang đó được submit quay trở lại từ Server thì thuộc tính IsPostBack nhận giá trị True

1.1.2. Giới thiệu .Net Framework:

.NET Framework là cơ sở hạ tầng cung cấp cho người dùng cách thức sử dụng đa ngôn ngữ lập trình để truy cập thông tin, file, hoặc các chương trình của họ ở mọi lúc mọi nơi trên mọi cấu hình phần cứng và thiết bị.

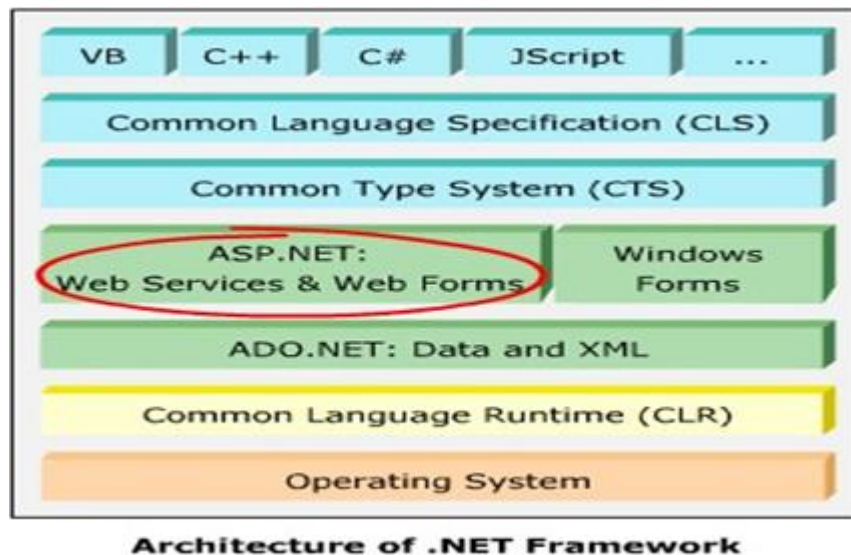
Tâm điểm của .NET Framework là CLR (Common Language Runtime) và tập phân cấp các bộ thư viện hợp nhất và ASP.NET. CLR quản lý sự thực thi của đoạn mã .NET và cung cấp các dịch vụ tạo quá trình phát triển chương trình ứng dụng dễ dàng hơn. Các trình biên dịch và các công cụ làm cho chức năng của thư viện thực thi runtime trở nên phong phú và hiệu quả hơn.

Ngoài ra, dịch vụ Web trong .NET Framework cho phép ta phát triển ứng dụng Internet hay Intranet trong hiện tại lẫn tương lai bằng bất cứ ngôn ngữ lập trình và truy cập đến hệ thống bất kỳ.

Ta có thể tham khảo tất cả những thành phần cấu thành trong .NET Framework như hình dưới đây, mức trên cùng là trình biên dịch Visual Basic hoặc các trình biên dịch của các ngôn ngữ khác trong bộ Visual Studio .NET.

Có thể sử dụng Visual Studio.NET kết hợp với môi trường phát triển (Integrated Development Environment - IDE) để lập trình ASP.NET. Visual Basic, C⁺⁺, C # .v.v...

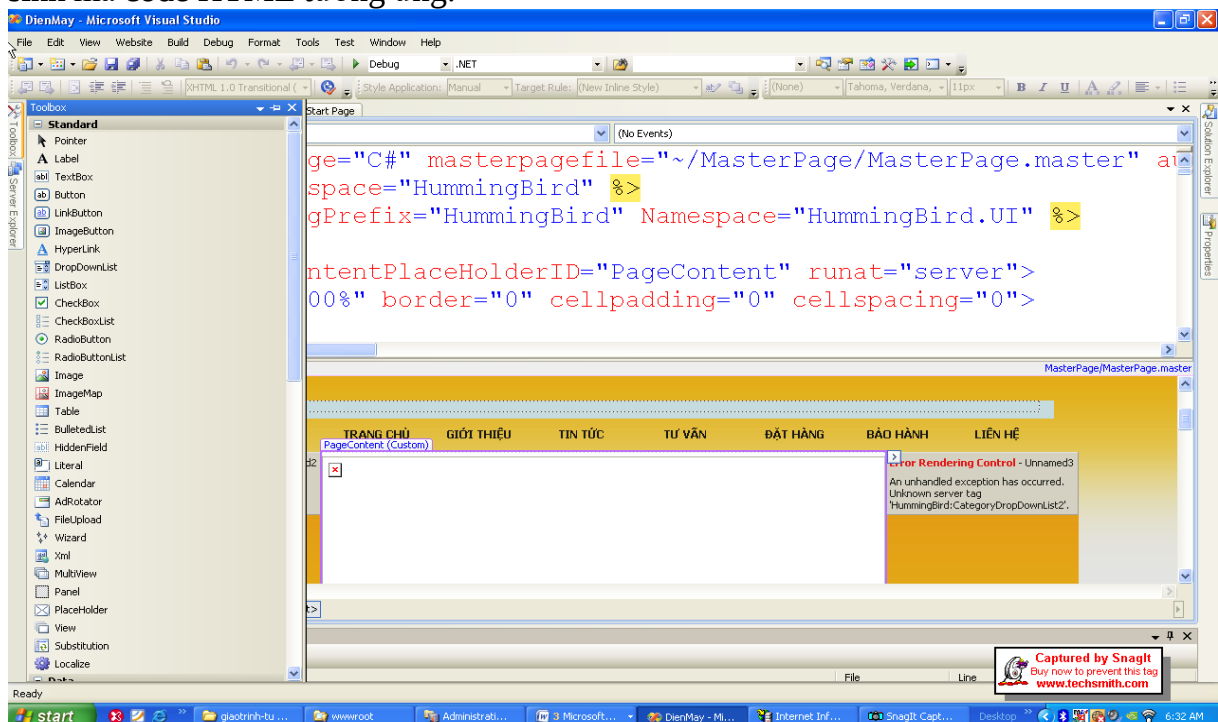
.NET Framework còn kết hợp mô hình lập trình đơn giản, dễ sử dụng với các giao thức mở và biến đổi được của Internet.



1.1.2. Giới thiệu môi trường lập trình

Ta có thể sử dụng Microsoft Visual Studio 20xx IDE (Integrated Development Environment) tạo các ứng dụng web ASP.NET. IDE bao gồm các tools cho việc phát triển, gỡ rối và triển khai ứng dụng.

Web Page Designing: Visual Studio 20xx IDE cung cấp bộ thiết kế (Design Editor) theo hiệu ứng “What you see is What you get”, do đó rất đơn giản để tạo một web page layout. Cung cấp khả năng kéo thả các thành phần vào Web Form và sinh mã code HTML tương ứng.



a) Cấu hình ứng dụng với IIS:

Internet Information Services (IIS) là một Web Server được phát triển bởi Microsoft để tạo host cho các websites trên một server đơn dùng để quảng bá và quản trị website trên Internet hoặc Intranet.

Làm việc IIS với các ứng dụng ASP.NET: Các ứng dụng ASP.NET được lưu trữ trên một thư mục ảo của IIS. Các bước truyền thông giữa ASP.NET và IIS như sau:

1. Browse gửi Request đến IIS

2. IIS nhận request, tìm kiếm file tương ứng và gửi file này đến ASP.NET Script Engine.

3. Script Engine thực hiện các Server - Side Script, sinh HTML page và gửi cho IIS.

4. IIS nhận và gửi cho Browse

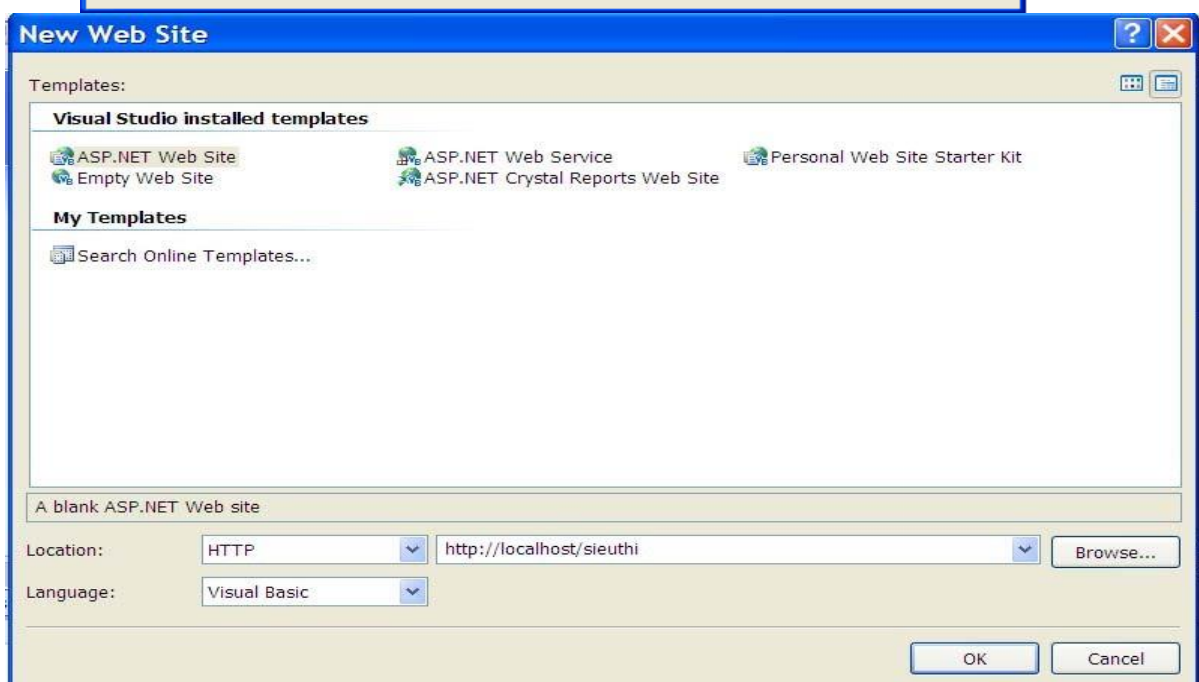
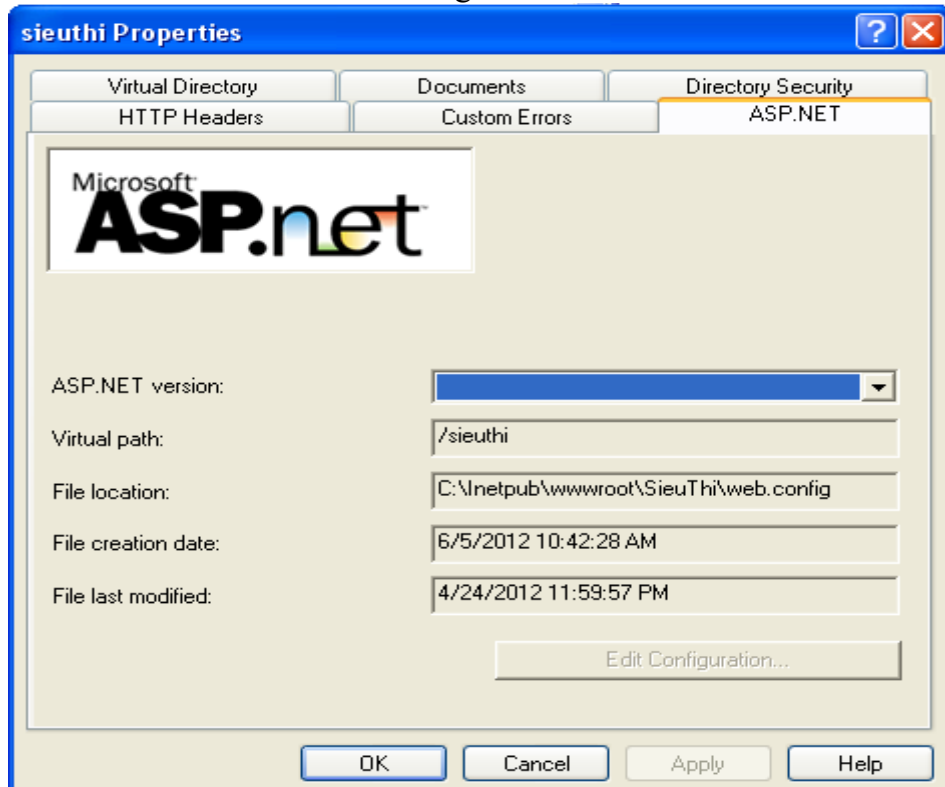
Cấu hình IIS

- Mở IIS.

- Right click lên Default Web Site chọn: New\Virtual Directory và làm theo các chỉ dẫn.

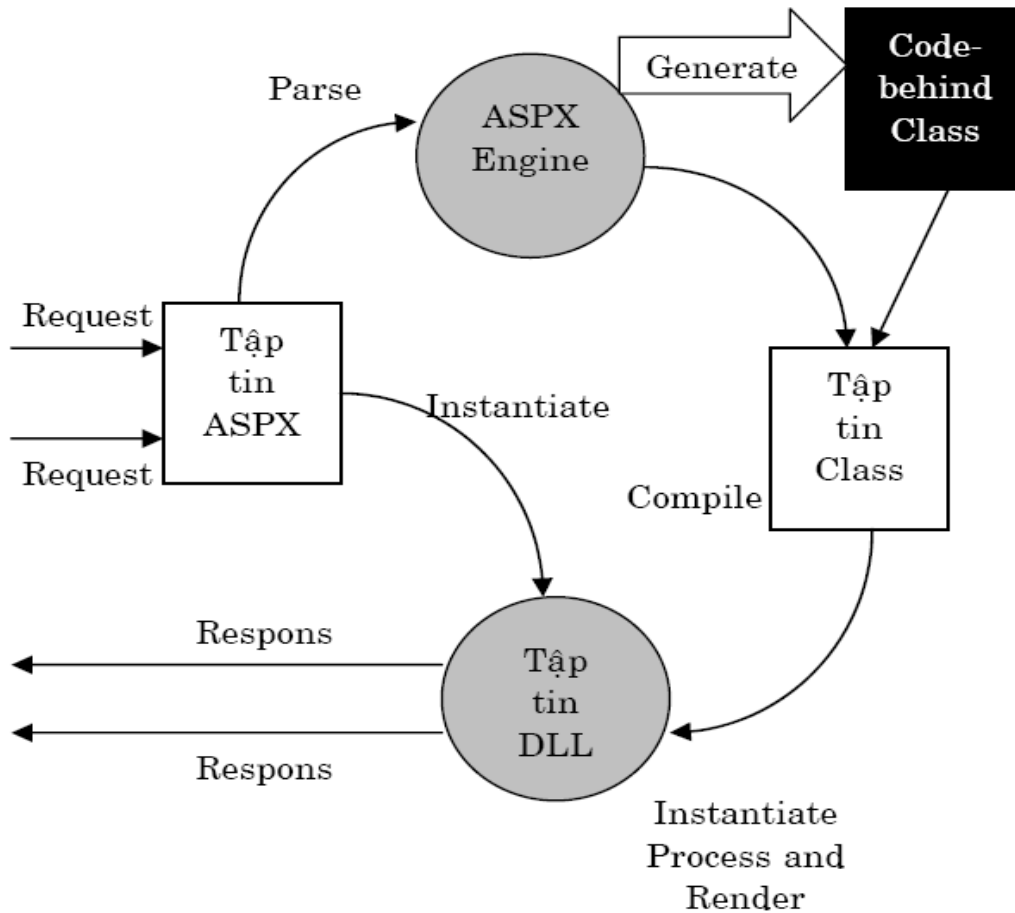
- Chỉ định IIS Web Project: Chỉ định trong Location là HTTP

Xem các hình minh họa sau dùng để cấu hình cho thư mục web có tên Sieuthi.



b) Biên dịch trang .Aspx:

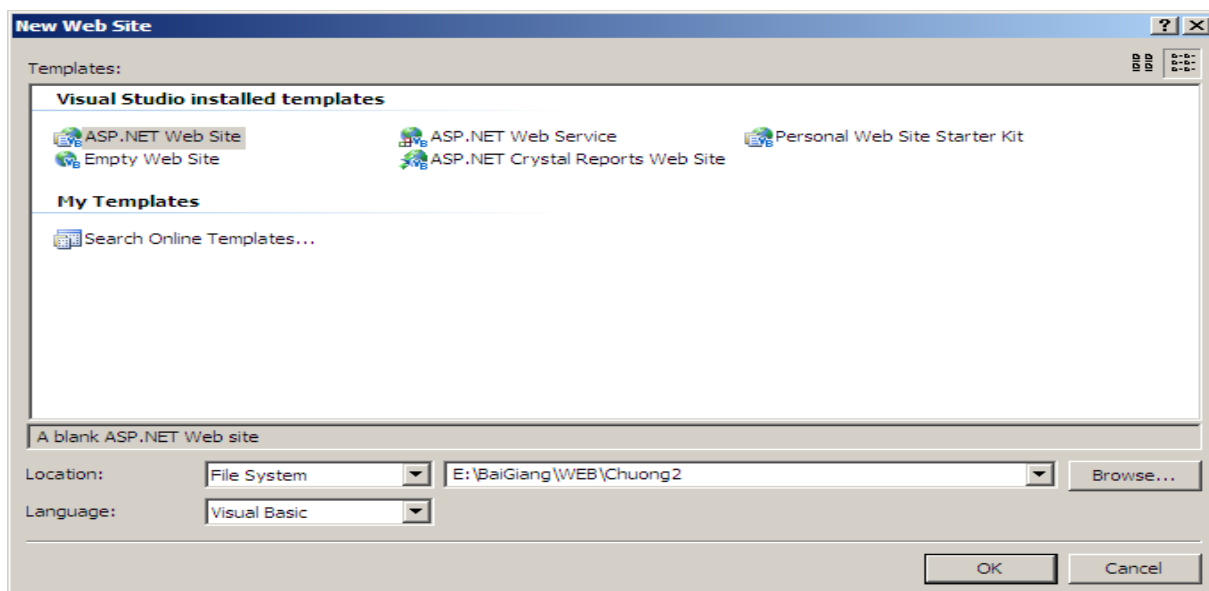
Các trang ASP.NET có đuôi là *.aspx. Khi người sử dụng lần đầu tiên triệu gọi trang ASPX, thì IIS triệu gọi trình biên dịch dịch trang ASPX (trang Code- behind) thành tập tin Class. Tiếp theo, tập tin Class này được biên dịch thành tập tin DLL. Cuối cùng, trang DLL thực thi và trả về kết quả cho người sử dụng.



Trong trường hợp người sử dụng triệu gọi lại trang ASPX, thì tập tin DLL sẽ được gọi và thực thi để trả kết quả về cho người sử dụng. Trang ASPX sẽ chỉ biên dịch lại tập tin DLL khi chúng tìm thấy cấu trúc của nó thay đổi hoặc chúng không tìm thấy tập tin DLL tương ứng.

c) Tạo ứng dụng ban đầu:

1. Khởi động Visual studio, File\ New Web site. Ta thấy hộp thoại New Web Site như hình sau:
2. Trong hộp thoại New Web Site, chọn ASP.NET Web Site và ngôn ngữ Visual Basic (hoặc C#,...).
3. Click File System trong hộp Location, click Browse, và sau đó chỉ định đường dẫn đến thư mục lưu trữ.
4. Click OK để hoàn thành các thay đổi.



Visual Studio nạp Visual Web Developer và tạo một Web page. Mỗi một Web page chứa hai phần:

- Một Web Forms page, chứa HTML và các điều khiển để tạo giao diện người sử dụng.
- file code-behind, chứa các module code, chứa program code trong Web Forms page.

Tương ứng với nó là hai file được mặc định là: file Default.aspx chứa giao diện người sử dụng và file Default.aspx.vb chứa phần code cho Web page này.

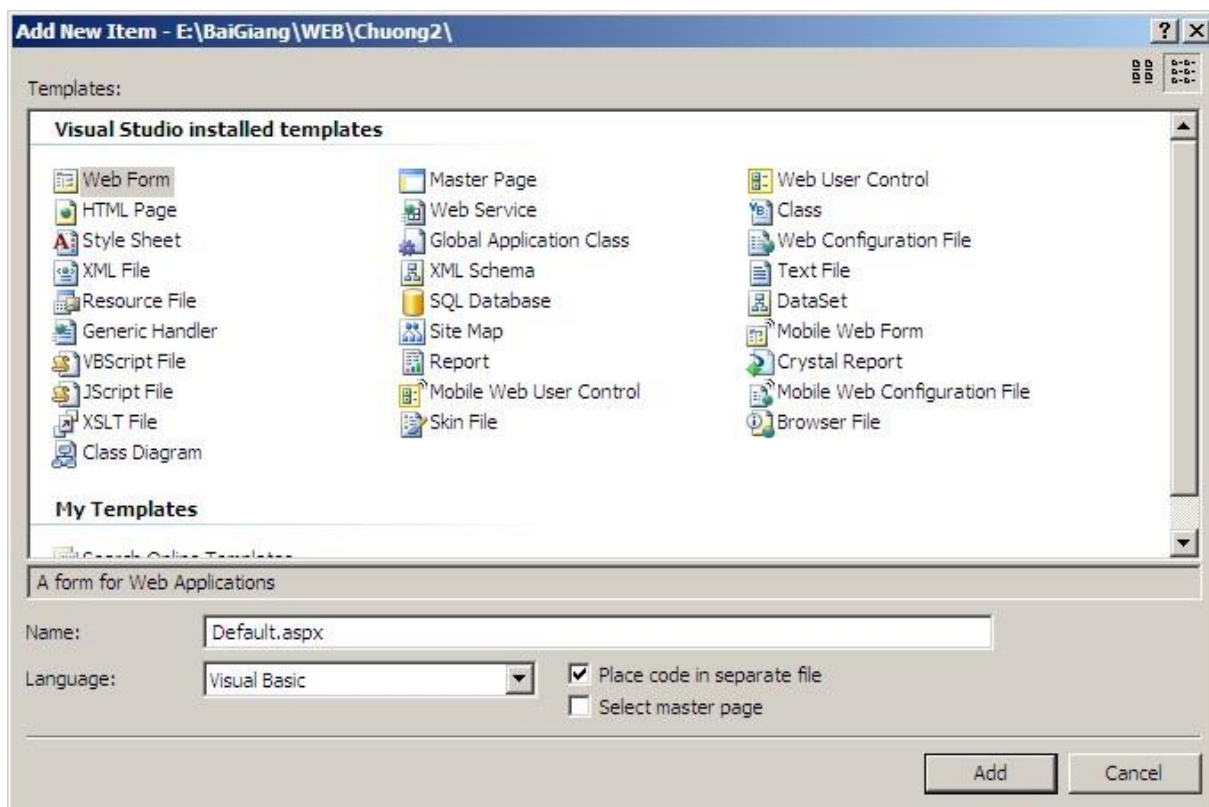
Ngoài web pages, web sites có thể chứa các modules (.vb files, .cs files), HTML pages (.htm files), configuration information (Web.config file), global Web application information (Global.asax file), và các thành phần khác. Ta có thể sử dụng Web Page Designer và Solution Explorer để chuyển đổi giữa các thành phần này một cách nhanh chóng và hữu hiệu.

Trong cửa sổ Web Page Designer, HTML source code cho Web page được hiển thị trên tab Source. Tab Design dùng để hiển thị Web page như khi trên Web browser. Khi tab Design được chọn thì một trang trắng xuất hiện đó là kết quả của phần code khởi tạo. Ta có thể thiết kế thêm các điều khiển và điều chỉnh các đối tượng trên trang. Ta sử dụng trang *.aspx.vb hoặc *.aspx.cs để viết mã code cho trang.

d) Thêm web page:

Để thêm một trang vào web site ta thực hiện các bước sau:

1. Click vào menu Website chọn Add New Item (Hoặc trong cửa sổ Solution Explorer, right click, và chọn Add New Item). Xuất hiện cửa sổ Add New Item như hình sau:



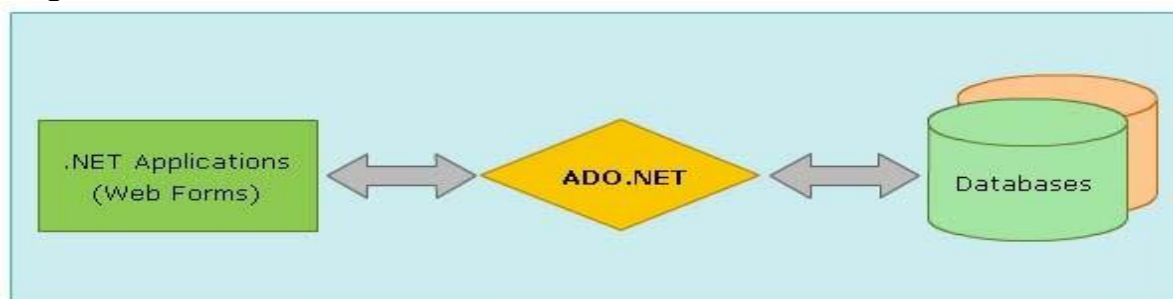
Trong cửa sổ này ta thực hiện các lựa chọn

- Mục Templates: Chọn Web form
- Mục Name: Đặt tên cho trang
- Mục Language: Chọn ngôn ngữ lập trình.
- Chọn: Place code in separate file mặc định là tách trang này ra làm 2 file như đã mô tả trên. Trong trường hợp không tách ta có thể bỏ lựa chọn này.

Khi bỏ lựa chọn này thì HTML và các điều khiển để tạo giao diện người sử dụng cùng với code nằm trên cùng một file.

1.2 Mô hình ADO.Net

Khi làm việc với .NET để phát triển các ứng dụng web thì sự nỗ lực của bạn để điều khiển dữ liệu sẽ là nhỏ nhất. Bởi vì, ASP.NET cung cấp các đối tượng ActiveX Data Objects cho .NET (ADO.NET), một tập hợp các công nghệ truy xuất dữ liệu là một phần của .NET Framework, giúp cho việc điều khiển và làm việc với dữ liệu trở lên dễ dàng hơn.



ADO.NET được thiết kế để cung cấp kiến trúc rời rạc (disconnected architecture). Có nghĩa là các ứng dụng connect tới database truy cập dữ liệu và lưu trữ trong memory. Sau đó disconnect với database và sử dụng bộ nhớ sao chép (memory copy) của dữ liệu đó. Nếu như database cần phải được update với các thay đổi trên memory copy, một kết nối mới (connection) được hình thành và database được update. Bộ nhớ chính lưu trữ dữ liệu là DataSet, nơi mà chứa các bộ nhớ lưu trữ dữ liệu khác,

như là: Các đối tượng DataTable; ta có thể lọc và sắp xếp dữ liệu trên DataSet bằng việc sử dụng các đối tượng DataView,...

Chúng ta sẽ tìm hiểu về ADO.net sâu hơn ở phần sau.

2. Các đối tượng ASP.Net:

Mục tiêu: Sử dụng tốt các đối tượng ASP.Net hỗ trợ lập trình.

2.1. ASP.Net Web Server Controls:

2.1.1. Giới thiệu các điều khiển web

a) Server Controls và HTML Controls

Server Controls: có nhiều khả năng hơn HTML controls và chức năng của nó trong một số cách giống như Windows Forms controls. Nhiều điều khiển server controls có cùng tên như các điều khiển Windows Forms controls và cùng cấp nhiều thuộc tính, phương thức và sự kiện giống nhau, thêm vào đó là các điều khiển như FileUpload, LoginView, và RequiredFieldValidator.

HTML Controls: là một tập hợp các điều khiển giao diện người dùng trước đây mà được hỗ trợ bởi hầu hết các trình duyệt web và phù hợp với chuẩn HTML phát triển cho việc quản lý các phần tử giao diện người dùng trên các trang web điển hình. Chúng bao gồm: Button, Text Field, Checkbox và các điều khiển cơ sở hữu ích cho việc quản lý thông tin trên một trang Web mà có thể được trình bày hoàn toàn bằng mã code HTML.

b) ASP.NET Web Server Controls

Khi tạo trang ASP.NET Web pages, ta có thể sử dụng các kiểu controls:

- **Web server controls** Controls có nhiều tính năng cài sẵn hơn HTML server controls. Web server controls không chỉ bao gồm các điều khiển như buttons and text boxes, mà còn các điều khiển cho mục đích đặc biệt như calendar, menus và tree view control. Web server controls có nhiều tính năng trừu tượng hơn HTML server controls.

- **HTML server controls** Các phần tử HTML có thêm tính năng tương tác trên trình chủ (server) vì vậy ta có thể lập trình chúng. HTML server controls trưng bày một đối tượng mà ánh xạ rất gần tới các phần tử HTML mà chúng hoàn trả.

- **Validation controls** Controls kết hợp logic cho phép ta kiểm soát những gì mà người dùng chèn vào các điều khiển Input như điều khiển TextBox. Validation controls cho phép ta kiểm tra quy định của một trường, kiểm tra dựa vào một giá trị đặc biệt, hoặc một mẫu ký tự, kiểm tra xem một giá trị nào đó có nằm trong một phạm vi hay không, .v.v...

- **User controls** Controls mà được tạo như ASP.NET Web pages. Ta có thể nhúng ASP.NET user controls trên các trang ASP.NET Web pages khác. Đây là cách đơn giản để tạo toolbars và các phần tử được sử dụng lại khác.

- Và một số nhóm điều khiển khác như: Data, Navigation, login,...

2.1.2. Web Server controls

Các điều khiển Web Server cung cấp giao diện đồ họa (Graphical User Interface). Các điều khiển này được sử dụng để cung cấp giao diện và chức năng cho web form.

Sau đây là danh sách các điều khiển Web server hay sử dụng:

<i>Tên điều khiển</i>	<i>Mô tả</i>	<i>Một số thuộc tính, phương thức và sự kiện</i>
-----------------------	--------------	--

Lable	Cho phép hiển thị nhãn trên Web Form	Text Property
TextBox	Điều khiển này thường được sử dụng để nhận input từ người sử dụng	Properties: AccessKey, AutoPostBack, MaxLength, Text, TextMode Event: TextChanged
Button	Điều khiển Button chủ yếu được sử dụng để Submit thông tin lên Server	Properties: AccessKey, Text, Enable Event: Click, Command
Image	Điều khiển Image cho phép hiển thị ảnh trên Web page	Properties: ImageUrl, AlternateText, ImageAlign, ToolTip
ImageButton	Điều khiển ImageButton cho phép tạo Button dạng ảnh trên Web page	Properties: ImageUrl, AlternateText, ImageAlign, OnClientClick Event: Click, Command
LinkButton	Điều khiển LinkButton cho phép hiển thị trên Web page một nút lệnh dưới dạng HyperLink	Properties: PostBackURL, Event: Click, Command
Pannel	Điều khiển Panel được sử dụng giống như một container chứa các điều khiển Web Server khác.	Properties: Controls, GroupingText,...
CheckBox	Điều khiển cho phép chọn hay không chọn.	Properties: AutoPostBack, Checked, Text. Event: CheckedChanged
RadioButton	Điều khiển RadioButton được sử dụng để chọn một trong nhiều lựa chọn khác nhau.	Properties: AutoPostBack, Checked, Text, GroupName Event: CheckedChanged
CheckBoxList	Điều khiển CheckBoxList được xét như một nhóm các điều khiển CheckBox có liên quan với nhau.	Properties: AutoPostBack, Items, Text, SelectedIndex, SelectedValue Method: ClearSelection
RadioButtonList	Tương tự như một nhóm các điều khiển RadioButton	Properties: AutoPostBack, Items, Text, SelectedIndex, SelectedValue Method: ClearSelection

ListBox	Điều khiển ListBox được sử dụng cho phép lựa chọn nhiều phần tử trong một danh sách cho trước.	Properties: AutoPostBack, Items, SelectedIndex, SelectedValue, SelectionMode Method: ClearSelection(), GetSelectedIndices() Event: SelectedIndexChanged
DropDownList	Điều khiển DropDownList là điều khiển cho phép chọn một phần tử trong danh sách các phần tử thả xuống.	Properties: AutoPostBack, Items Method: ClearSelection() Event: SelectedIndexChanged
Calendar	Điều khiển Calendar là điều khiển cho phép hiển thị lịch trên Browse	Properties: DayHeaderStyle, DayNameFormat, SelectedDate, SelectionMode.
AdRotator	là điều khiển đặc biệt của ASP.NET cho phép hiển thị các banner quảng cáo.	
Menu	Là điều khiển dùng để điều hướng các trang trong WebSite	Properties: Items, Orientation, PathSeparator, StaticDisplayLevels Method: FindItem() Event: MenuItemClick
TreeView	Điều khiển TreeView được xem như một menu có thể mở rộng và thu hẹp giống như thanh trái của cửa sổ Explorer.	Properties: Nodes, ShowCheckBoxes, SelectedValue, NodeStyle, ShowLines, SelectedNode Method: FindNode() Event: SelectedNodeChanged, TreeNodeCollapsed, TreeNodeExpanded, TreeNodeCheckChanged
ImageMap	ImageMap cho phép tạo các điểm nóng ảnh có khả năng clickable.	Properties: HotSpotMode, HotSpot, ImageURL, AlternateText. Event: Click
FileUpload	cho phép định vị và Upload các file	Properties: HasFile, FileName, PostedFile, ... Method: SaveAs()
View	Điều khiển này thường được sử dụng giống như một container chứa các điều khiển khác.	

MultiView	Một điều khiển MultiView chứa một hoặc nhiều View	Properties: ActiveViewIndex, Views. Method: SetActiveView Event: ActiveViewChanged
-----------	---	--

Sử dụng control AdRotator: Quảng cáo là một trong những cách tạo ra những lợi nhuận cho nhà sở hữu website. AdRotator cho phép chọn ngẫu nhiên các ảnh đặt lên trang web quảng cáo, để làm được điều này trước tiên ta phải tạo danh sách các file ảnh cần đưa lên trang web quảng cáo. Danh sách này được đặt trong một file XML (vào menu Website\Add New Item\ và chọn Template là XML file – ads.XML) theo cấu trúc sau:

Ví dụ. Sử dụng control AdRotator:

a) file ads.XML

```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements >
  <Ad>
    <ImageUrl>~/ads/yahooCall.gif</ImageUrl>
    <NavigateUrl>Vidu2_5.aspx</NavigateUrl>
    <AlternateText> Click here to visit our sponsor.
  </AlternateText>
    <Impressions>1</Impressions>
    <Keyword> A </Keyword>
  </Ad>
  <Ad>
    <ImageUrl>~/ads/yahooCall.png</ImageUrl>
    <NavigateUrl>//www.gotdotnet.com</NavigateUrl>
    <AlternateText> ASP.NET tutorials and more.</AlternateText>
    <Impressions>5</Impressions>
    <Keyword> B </Keyword>
  </Ad>
</Advertisements>
```

Trong đó các thẻ được sử dụng cho trong bảng sau.

Bảng các thẻ của Advertisements

Tag	Ý nghĩa
<Ad>	Bắt đầu thẻ Ad
<ImageUrl>	Địa chỉ của Ad để hiển thị (địa chỉ file ảnh)
<NavigateUrl>	Địa chỉ điều hướng nếu người sử dụng click vào Ad
<AlternateText>	Text để hiển thị như là ToolTip nếu người sử dụng di chuột trên Ad. Và nếu địa chỉ ImageUrl không thể hiển thị thì dòng text này sẽ hiển thị thay thế.
<Keyword>	Một tên category sử dụng cho việc lọc các Ads khi hiển thị.
<Impressions>	Một số thể hiện khả năng một Ad sẽ được hiển thị. Các Ads với số càng cao thì khả năng hiển thị càng nhiều.

b) file vidu2_2.aspx

```
<%@ Page Language="VB" %>
```

```

<html xmlns="http://www.w3.org/1999/xhtml" >
  <head runat="server">
    <title> Hello My web</title>
  </head>
  <body>
    <form id="form1" runat="server">
      <div>
        <asp:AdRotator ID="AdRotator1" runat="server"
          AdvertisementFile="~/ads.xml" />
      </div>
    </form>
  </body>
</html>

```

2.1.3. Web HTML Server Control:

Theo mặc định các thẻ HTML được sử dụng để tạo các điều khiển HTML tương ứng trên các ứng dụng ASP.NET và được xét như là plain text, không thể code bên server-side.

HTML Server Controls cung cấp để truy xuất vào các thuộc tính và tính chất của các thẻ HTML.

HTML Server Controls trong ASP.NET là tập các điều khiển tương ứng các thẻ HTML, chúng được khai báo trên Web page bằng cách thêm thuộc tính runat="Server".

Sau đây là một số class hay sử dụng của nhóm điều khiển HTML Server:

<i>Class</i>	<i>Mô tả</i>
HtmlButton	Cho phép lập trình sử dụng thẻ HTML <button> trên Server
HtmlForm	Cho phép lập trình sử dụng thẻ HTML <form> trên Server
HtmlInputButton	Cho phép lập trình sử dụng thẻ HTML <Input type=button > <Input type= submit >, <Input type=reset > trên Server
HtmlInputText	Cho phép lập trình sử dụng thẻ HTML <Input type=text > <Input type= password > trên Server
HtmlTable	Cho phép lập trình sử dụng thẻ HTML <table> trên Server
HtmlSelect	Cho phép lập trình sử dụng thẻ HTML <select> trên Server

Một câu hỏi đặt ra là ta nên sử dụng HTML Controls hay ASP.NET Server Controls? Cả hai loại đều sinh ra mã ánh xạ vào các phần tử HTML như nhau. Câu hỏi này liên quan đến các phần tử đơn trên một trang, như thẻ text box, the button, tables, .v..v.. Và được các nhà phát triển kiến trúc ASP.NET của Microsoft trả lời rằng: Mục đích là sử dụng chuyển, sử dụng loại điều khiển nào tùy thuộc vào mục đích, tình huống sử dụng, và thói quen của người sử dụng.

Server controls và HTML controls đưa ra các chức năng chồng chéo lên nhau. Tổng quát, làm việc với Server controls thường dễ dàng hơn. Bảng sau đưa ra danh sách các Server controls và HTML controls với các task lập trình.

Task	Server controls	HTML controls
------	-----------------	---------------

Hiển thị text	Label, TextBox, Literal	Label, Text Field, Text Area, Password Field
Hiển thị tables	Table, DataGrid	Table
Chọn từ danh sách	DropDownList, ListBox, DataList, Repeater	List Box, Dropdown
Thực thi lệnh	Button, LinkButton, ImageButton	Button, Reset Button, Submit Button
Set values	CheckBox, CheckBoxList, RadioButton, RadioButtonList	Checkbox, Radio Button
Hiển thị images	Image, ImageButton	Image
Navigation	Hyperlink	none (use <a> tags in text)
Group controls	Panel, Placeholder	Flow Layout, Grid Layout
Làm việc với dates	Work with dates	none
Hiển thị ads (Quảng cáo)	AdRotator	none
Hiển thị horizontal rules	Literal	Horizontal Rule
Lấy filenames từ client	None	File Field
Store data on page	(provided by state management)	Input Hidden
Validate data	RequiredFieldValidator, CompareValidator, RangeValidator, RegularExpressionValidator, CustomValidator, ValidationSummary	none (use page- level scripts)

2.1.4. Validation Controls:

Validation Controls là các điều khiển đảm bảo sự xác nhận tính hợp lệ của dữ liệu được nhập vào.

Một số điều khiển kiểm tra tính hợp lệ

- RequiredFieldValidator: RequiredFieldValidator là điều khiển đảm bảo người dùng nhập dữ liệu trên các điều khiển Input. Nếu người dùng không nhập dữ liệu trên điều khiển Input mà liên kết với điều khiển RequiredFieldValidator thì tiến trình của trang sẽ bị dừng lại cho đến khi dữ liệu được nhập vào.

- CompareValidator là điều khiển dùng để so sánh giá trị của một điều khiển Input với một điều khiển Input khác hoặc với một giá trị hằng. Ta có thể sử dụng thuộc tính Type quy định kiểu dữ liệu của các giá trị nhập vào để so sánh như: String (mặc định), Integer, Double, Date và Currency.

- RangeValidator là điều khiển được ASP.NET cung cấp để kiểm tra giá trị người dùng nhập vào có nằm trong phạm vi được chỉ định hay không? Điều khiển này kiểm tra tính hợp lệ của các giá trị dạng numeric, character và date.

- RegularExpressionValidator là điều khiển dùng để kiểm tra tính hợp lệ của giá trị trên điều khiển Input theo một khuôn mẫu của biểu thức. Ví dụ, expression có thể là pattern của phone number, mail Address,...

- CustomValidator là điều khiển dùng để kiểm tra tính hợp lệ của giá trị nhập trên điều khiển Input có hợp lệ theo một logic đã chỉ định hay không. Ta có thể dùng điều khiển này để kiểm tra số chẵn, lẻ, nguyên tố, ...

- ValidationSummary là điều khiển dùng để xem lại các thông báo lỗi xuất hiện trên các điều khiển Validation khác. Điều khiển ValidationSummary tổng hợp và hiển thị tất cả các thông báo lỗi trên một trang.

2.2. Các đối tượng trong ASP.NET

2.2.1. Đối tượng Request

Đối tượng Request dùng để nhận thông tin yêu cầu được gửi từ Client Side đến Server Side. Nó được cài đặt trong lớp *HttpRequest* thuộc tên miền System.Web. Khi sử dụng đối tượng Request ta có thể sử dụng các thành phần (collection) và thuộc tính của nó cho trong bảng sau:

<i>Thuộc tính/ Phương thức</i>	<i>Mô tả</i>
FilePath	Trả về đường dẫn ảo của yêu cầu, thuộc tính này tương đương với SCRIPT_NAME trong ASP.
Files	Trả về HttpRequestCollection của tập nhiều tập tin được tải lên Server (sử dụng cho dạng multi- part/forms).
Form	Trả về một tập dữ liệu của nội dung từ Form(NameValueCollection). Tham khảo Request Collection trong phần kế tiếp.
Params	Thuộc tính này sử dụng để lấy giá trị của phương thức trong Form, QueryString, ServerVariable hay Cookie
Path	Đường dẫn ảo của yêu cầu, tương đương với PATH_INFO trong ASP.
PathInfo	Đường dẫn ảo của yêu cầu, tương đương với PATH_INFO trong ASP.
PhysicalApplicat ionPath	Đường dẫn vật lý của thư mục gốc, tương đương với APPL_PHYSICAL_PATH
PhysicalPath	Đường dẫn vật lý của yêu cầu, tương đương với PATH_TRANSLATED trong ASP.
QueryString	Trả về một tập dữ liệu của nội dung từ QueryString (NameValueCollection). Cách truy cập tập dữ liệu này khác với truy cập tập dữ liệu từ Form của ASP.
TotalBytes	Dung lượng của Stream trong luồng dữ liệu.

Url	Đối tượng Url chứa đựng chi tiết của yêu cầu. Đối tượng Url (từ không gian tên System) bao gồm các thông tin chi tiết như Port, DNS,...
UserHostAddress	Địa chỉ IP của người sử dụng, tương đương với REMOTE_ADDR trong ASP.
UserHostName	Tên DNS của người sử dụng, tương đương với REMOTE_NAME trong ASP.
MapPath()	Chuyển đổi đường dẫn ảo thành đường dẫn vật lý.
SaveAS()	Lưu yêu cầu HTTP vào đĩa.

QueryString: QueryString được định nghĩa là một chuỗi nằm sau dấu ? trong chuỗi URL (Uniform Resource Locator) trên phần Address của trình duyệt khi triệu gọi đến một trang Web.

Ví dụ: Ta gọi triệu địa chỉ sau: <http://192.168.10.12/Chapter-03/ex5.asp?al=A&page=cust> Trong đó, QueryString bao gồm các cặp tham số và giá trị sau: al=A&page=cust

Các tham số được khai báo cách nhau bằng ký tự & và giá trị của mỗi tham số được khai báo sau dấu = của mỗi tham số.

- Nếu tham số không có dấu bằng theo sau, giá trị của tham số đó sẽ được trả về khi sử dụng Request.QueryString là null.

- Trong trường hợp có dấu bằng nhưng không tồn tại giá trị thì kết quả trả về khi sử dụng Request.QueryString là rỗng.

Đối tượng Request cho phép truy cập đến nội dung của Form hay QueryString, trong đó bao gồm các phần tử cùng tên như trong các trình bày chúng ta sử dụng thẻ checkbox, radio, v.v... cùng tên và khác giá trị.

2.2.2. Đối tượng Response:

Đối tượng Respose dùng để gửi thông tin ngược trở lại Client Side từ Server Side. Nó được cài đặt trong lớp *HttpResponse* thuộc tên miền System.Web. Các thuộc tính và phương thức hay sử dụng đối với đối tượng này cho trong bảng sau:

Thuộc tính/ Phương thức	Mô tả
Expires	Thuộc tính này chỉ định thời gian trước khi trang đó được lưu trong bộ nhớ Cache của trình duyệt. Người sử dụng có thể trở lại các trang trước đó khi khoảng thời gian này chưa hết.
ContentType	Chỉ định nội dung phức tạp. Thuộc tính cho phép nhận và thiết lập kiểu Multipurpose Internet Mail Extension (MIME) của Response (mặc định là "text/html").
Cookies	Thuộc tính cho phép nhận tập hợp các cookies được truyền tới client trong Response Public ReadOnly Property Cookies() As System.Web.HttpCookieCollection

Redirect()	Chuyển hướng đến địa chỉ file trong cùng ứng dụng hay URL khác trong lúc thi hành.
Write()	Ghi thông tin từ các kiểu dữ liệu như Char, Object, String, Array ra trang Web.
Flush()	Phương thức này gửi Buffer của Response đến client.

2.2.3. Đối tượng Server:

Đối tượng Server thể hiện các phương thức tiện ích đa dạng để chuyển đổi điều khiển giữa các trang, giải mã HTML text, nhận thông tin lỗi, ... Đối tượng Server thuộc lớp `HttpServerUtility` thuộc không gian miền `System.Web`

- Lớp `HttpServerUtility` cung cấp các phương thức cho các tác vụ như tiến trình xử lý các yêu cầu, mã hóa và giải mã các chuỗi URL, truy cập các thông tin lỗi,...

Lớp `HttpServerUtility` cung cấp hai thuộc tính:

+ `MachineName`: Trả về tên Server mà ứng dụng đang chạy.

`Public ReadOnly Property MachineName() As String`

+ `ScriptTimeout`: Lấy hoặc thiết lập thời gian (tính bằng giây) mà một Request timeout đã xác định.

`Public Property ScriptTimeout() As Integer`

Ví dụ:

`Response.Write("Time Out:" + Server.ScriptTimeout.ToString)`

Một số phương thức của lớp `HttpServerUtility` thường được sử dụng đó là:

+ `Execute`: Là phương thức xử lý trang Web được chỉ định trên ngữ cảnh (context) của current Request.

`Public Sub Execute(ByVal path As String)`

path: Là URL path cần được thực thi.

+ `HtmlEncode`: Là phương thức mã hóa chuỗi ký tự để hiển thị trên trình duyệt. Có hai dạng hàm:

`Public Function HtmlEncode(ByVal s As String) As String`

Dùng để mã hoá chuỗi *s* và trả về chuỗi đã mã hóa.

Ví dụ:

`str = "Xin chào </br>"`

`strE = Server.HtmlEncode(str) Response.Write("Bản gốc:" + str)`

`Response.Write("Bản mã:" + strE)`

`Public Sub HtmlEncode(ByVal s As String, ByVal output As System.IO.TextWriter)>> Dùng để mã chuỗi s và trả về chuỗi đã mã hóa cho output.`

Ví dụ:

`Dim out As New System.IO.StreamWriter("E:\Test.txt") Server.HtmlEncode("Xin chào <Message>", out) Server.HtmlDecode("Xin chào <Message>", out) out.Flush()`

+ `MapPath`: Là một hàm trả về đường dẫn vật lý.

`Public Shared Function MapPath(ByVal virtualPath As String) As String`

Ví dụ: `Response.Write(Server.MapPath("\Testing"))`

+ `UrlEncode`: Là phương thức mã hóa chuỗi URL:

`Public Function UrlEncode(ByVal s As String) As String`

`Public Sub UrlEncode(ByVal s As String, ByVal output As System.IO.TextWriter)`

2.2.4. Đối tượng Application

Đối tượng Application được sử dụng để tham chiếu đến thể hiện của lớp HttpSessionState. Application State (trạng thái ứng dụng) lưu trữ toàn bộ thông tin sử dụng qua lại giữa nhiều phiên giao dịch (Session) và các yêu cầu (request). Đối tượng Application nắm hầu hết các thông tin được sử dụng cho nhiều trang của ứng dụng.

Lớp HttpSessionState thuộc không gian miền System.Web. Lần đầu tiên Client yêu cầu tài nguyên URL từ thư mục ảo của ứng dụng ASP.NET thì một thể hiện của lớp HttpSessionState được tạo.

a) Các thuộc tính của đối tượng Application: Đối tượng Application sẽ thể hiện các thuộc tính của lớp HttpSessionState. Một vài các thuộc tính quan trọng của lớp này là:

- AllKeys: Truy xuất các Access Keys từ tập các trạng thái ứng dụng. Đoạn code sau minh họa cách sử dụng thuộc tính AllKeys.

```
Dim strKeys(Application.Count) As String strKeys = Application.AllKeys
```

- Contents: truy xuất tham chiếu đến đối tượng HttpSessionState.

```
Application.Contents.RemoveAll()
```

- Count: đếm và truy xuất số các đối tượng.

```
intObj=Application.Count
```

- Item: Truy xuất đến một đối tượng cụ thể trong tập hợp HttpSessionState.

```
Application.Item("User")= "Huongct"
```

```
Response.Write(Application("User")& "<br>")
```

```
Response.Write(Application(0)& "<br>")
```

Chú ý: Cách truy xuất Application.Item("User") tương đương với Application("User") và tương đương với Application(0) trong đó 0 là chỉ số Index của đối tượng đầu tiên trong tập hợp.

b) Phương thức của lớp "HttpSessionState: Sau đây là một số phương thức quan trọng của lớp HttpSessionState:

- Add: Là phương thức thêm một đối tượng mới vào tập trạng thái ứng dụng.

```
Public Sub Add(ByVal name As String, ByVal value As Object)
```

Đoạn code sau minh họa cách sử dụng phương thức Add

```
Application.Add("UserName", "Huongct")
```

- Clear: Là phương thức xóa tất cả đối tượng trong tập trạng thái ứng dụng. Cú pháp của phương thức Clear là:

```
Public Sub Clear()
```

Đoạn code sau minh họa cách sử dụng phương thức:

```
Application.Clear()
```

- Remove: Là phương thức xóa một đối tượng chỉ định trong tập trạng thái ứng dụng. Cú pháp của phương thức là:

```
Public Sub Remove(ByVal name As String)
```

Đoạn code sau minh họa cách sử dụng phương thức:

```
Application.Remove("UserName");
```

- RemoveAll Method: Là phương thức xóa tất cả đối tượng trong tập trạng thái ứng dụng. Cú pháp của phương thức là:

```
Public Sub RemoveAll()
```

Đoạn code sau minh họa cách sử dụng phương thức:

```
Application.RemoveAll();
```

- RemoveAt Method: Là phương thức xóa một đối tượng chỉ định trong tập trạng thái ứng dụng. Cú pháp của phương thức là:

Public Sub RemoveAt(ByVal index As Integer)

Đoạn code sau minh họa cách sử dụng phương thức:

Application.RemoveAt(0);

c) Phương thức “Lock” và “UnLock”

Phương thức Lock():

Khi nhiều người dùng cùng truy cập vào một ứng dụng ASP.NET, ứng dụng không thể cho phép họ đồng thời truy cập thay đổi cùng một dữ liệu. Để làm được điều này, ta cần khóa các biến dữ liệu. Phương thức Lock(), thuộc lớp HttpSessionState, ngăn cản người dùng khác thay đổi các biến lưu trữ trong đối tượng Application. Cú pháp của phương thức Lock():

Public Sub Lock()

Phương thức UnLock():

Phương thức UnLock(), thuộc lớp HttpSessionState, được sử dụng để mở khóa cho các biến lưu trữ trong đối tượng Application. Các biến khi đã được mở khóa thì các người dùng khác có thể thay đổi chúng.

Phương thức Lock() được mở tương ứng bởi phương thức UnLock(). Trong trường hợp không gọi phương thức UnLock(), .NET Framework tự động xóa Lock() khi Request được hoàn thành, hoặc Request đến thời gian TimeOut, hoặc khi xuất hiện ngoại lệ (Request fail).

Cú pháp của phương thức UnLock():

Public Sub UnLock()

Ví dụ: Sử dụng các phương thức trên.

Application.Lock()

Application("VisitorsCount")= Application("VisitorsCount")+1

Application.UnLock()

2.2.5. Đối tượng Session

Khi có nhu cầu truyền giá trị từ trang này sang trang khác trong một phiên làm việc, ta sử dụng đối tượng Session. Bằng các sử dụng phương thức và thuộc tính của đối tượng này. Ta có thể khởi tạo, gán giá trị, truy cập và hủy đối tượng này trong một phiên làm việc nhằm quản lý người sử dụng khi họ truy cập vào Web Site. Web Server sẽ tự động tạo ra đối tượng Session khi chúng chưa tạo ra, đối tượng này sẽ có chu trình sống cho đến khi một trong các điều kiện sau xảy ra:

- Người sử dụng đóng trình duyệt, tức là đóng cửa sổ cuối cùng của cùng một trình duyệt.

- Khi thời gian sống (expiry) của chúng hết hạn.

- Chúng ta khai báo hủy bỏ đối tượng Session bằng các phương thức của chúng.

Đối tượng Session được cài đặt trong lớp HttpSessionState thuộc không gian tên System.Web.SessionState.

a) Session Variables (Các biến Session)

Các biến Session được sử dụng để lưu trữ thông tin về phiên làm việc của một người dùng. Các thông tin này có thể sử dụng cho tất cả các trang trong ứng dụng. Thông thường, thông tin được lưu trữ trong các biến Session là UserName, Password,... Các biến Session được xóa ngay khi phiên làm việc với Website của người dùng kết thúc.

Session Variables được lưu trữ trong lớp SessionStateItemCollection thuộc không gian miền System.Web.SessionState. Các biến này được thể hiện thông qua thuộc tính System.Web.HttpContext.Session.