



CHƯƠNG 7: WEB APPLICATION

Phan Trọng Tiến

BM Công nghệ phần mềm

Khoa Công nghệ thông tin, VNUA

Email: phantien84@gmail.com

Website: <http://timoday.edu.vn>



Nội dung chính

- I. Giới thiệu ASP.Net
- II. Tạo Web Forms
- III. Các Control Server của ASP.Net
- IV. Các Control Validation



I. Giới thiệu ASP.Net

- ❑ ASP.Net không giống phiên bản ASP. ASP.Net có các đặc điểm mới như :
 - ❑ Hỗ trợ nhiều ngôn ngữ lập trình
 - ❑ Có các điều khiển mới
 - ❑ Hỗ trợ XML
 - ❑ Nhiều khả năng bảo mật: chứng thực người sử dụng
 - ❑ Hiệu năng thực thi code cao
- ❑ Các phiên bản
 - ❑ ASP 1.0 năm 1996
 - ❑ Rồi phiên bản ASP 2.0 và 3.0
 - ❑ ASP.Net cung cấp một cách tiếp cận khác để phát triển ứng dụng Web.



Giới thiệu ASP.Net

- ❑ ASP.Net là một cuộc cách mạng trong phát triển ứng dụng Web
- ❑ ASP.Net dựa trên cơ sở .Net Framework
- ❑ .Net Framework chạy trên CLR.
- ❑ CLR có các ưu điểm:
 - ❑ Tự động quản lý bộ nhớ
 - ❑ Hỗ trợ nhiều ngôn ngữ
 - ❑ Bảo đảm việc chứng thực người sử dụng
 - ❑ Dễ cấu hình
 - ❑ Dễ triển khai



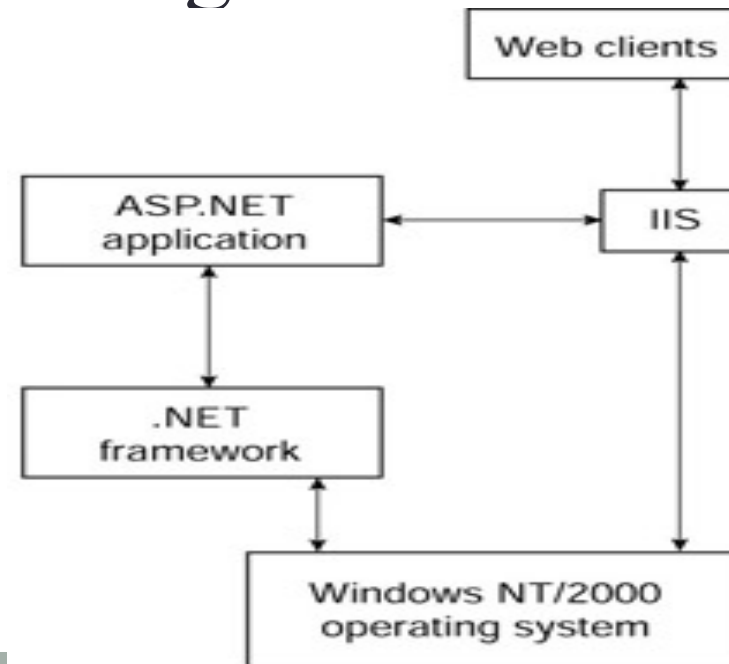
Các lợi ích mà ASP.Net cung cấp

- ❑ Hỗ trợ nhiều ngôn ngữ lập trình
- ❑ Phát triển qua nhiều ngôn ngữ
- ❑ Phần giao diện và phần code tách riêng biệt.
- ❑ Chứng thực người sử dụng
- ❑ Kiến trúc xử lý mới trên server
- ❑ Cải tiến chức năng gỡ rối và lần vết
- ❑ Cấu hình ứng dụng với nhiều điều khiển
- ❑ Dễ triển khai
- ❑ Cải tiến các tính năng caching như caching mức trang, caching mức đoạn, caching API. Dùng các tính năng caching trong ASP.Net sẽ tăng tốc độ và hiệu năng cho trang Web của bạn.



Các mô hình lập trình ASP.Net

- ❑ .Net Framework Software development Kit (SDK) được sử dụng để phát triển các ứng dụng Web. Nó chạy trên nền .Net Framework SDK cùng với IIS.





Hai mô hình lập trình ASP.Net

Web Forms

- Cho phép bạn tạo các trang Web động.
- Cũng có thể dùng các control để tạo các UI components

Web Services

- Mô hình lập trình này cho phép bạn thi hành một số các chức năng trên server.
- Web Services đóng vai trò quan trọng trong tích hợp các ứng dụng trên các nền khác nhau vì nó không giới hạn công nghệ.
- Web Services giúp bạn thay đổi dữ liệu trên client-server hoặc kiến trúc server-server.
- Web Services dùng các chuẩn như HTTP và XML để trao đổi dữ liệu



Các yêu cầu nền ASP.Net

- ❑ ASP.Net là một phần của .Net Framework SDK, được download <http://msdn.microsoft.com/downloads>
- ❑ Cần cài IE 5.5 trở lên
- ❑ Các hệ điều hành hỗ trợ .Net
 - ❑ Windows XP Professional
 - ❑ Windows 2000
 - ❑ Windows NT 4.0 with Service Pack 6a
 - ❑ Windows 98



II. Tạo Web Forms

- ❑ ASP.Net cho phép bạn tạo các trang web động nhanh hơn.
- ❑ Các đặc điểm Web Form:
 - ❑ Dùng .Net Framework chạy trên Web Server để tạo các trang web động.
 - ❑ Dùng các đặc điểm của CLR như sự an toàn và có sự kế thừa.
 - ❑ Thiết kế và lập trình sử dụng Tool Rapid Application Development(RAD) của VS.Net.
 - ❑ Không phụ thuộc vào client
 - ❑ Tương thích với bất kỳ trình duyệt web và thiết bị mobile



Các thành phần Web Form

❑ Giao diện người sử dụng

- ❑ Diễn tả nội dung tới người sử dụng. Nó bao gồm một file gồm code HTML hoặc code XML và các controls Server.
- ❑ Được lưu trữ trong file với đuôi mở rộng là .aspx

❑ Logic lập trình(code)

- ❑ Làm việc tương tác với người sử dụng với trang web form.
- ❑ Bất kỳ ngôn ngữ lập trình .Net (Vb.Net,C#...) dùng để viết code logic cho trang Web.
- ❑ Hai mô hình viết code: *code-inline* và *code-behind*
 - ❑ *Code-inline*: code được nhúng trực tiếp vào trang ASP.Net
 - ❑ *Code-behind*: code nằm ở một file riêng, và trang ASP.Net tham chiếu tới

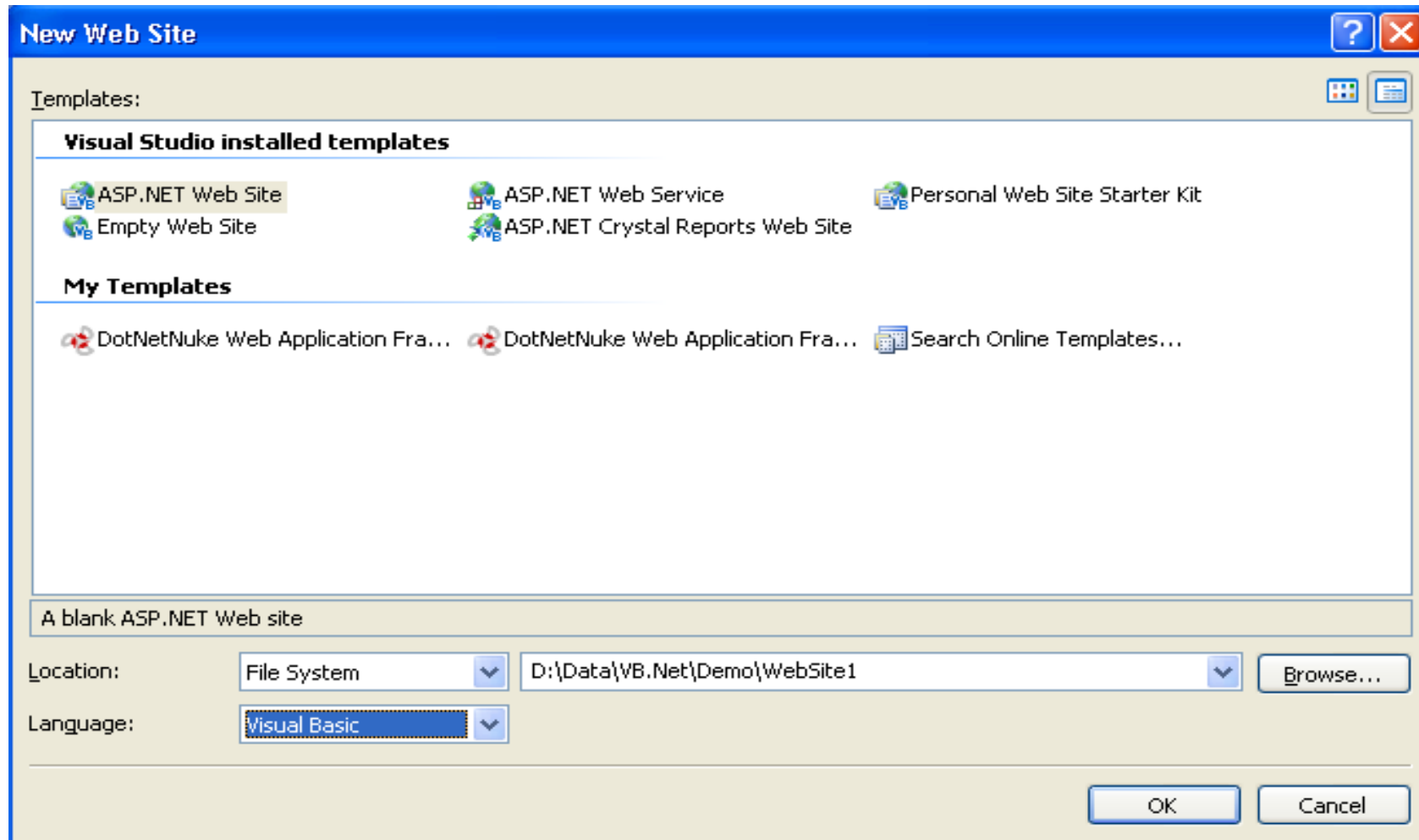


Thiết kế Web Forms

- ❑ B1: Start VS.Net vào *File\New\Web Site*
... để mở hộp thoại *New Project*
- ❑ B2: Chọn *Template* là *ASP.NET Web Site*
- ❑ B3: Chọn nơi đặt Website ở *Location*
- ❑ B4: Chọn ngôn ngữ cho trang ASP.Net ở
Language
- ❑ B5: Kích OK để hoàn thành việc

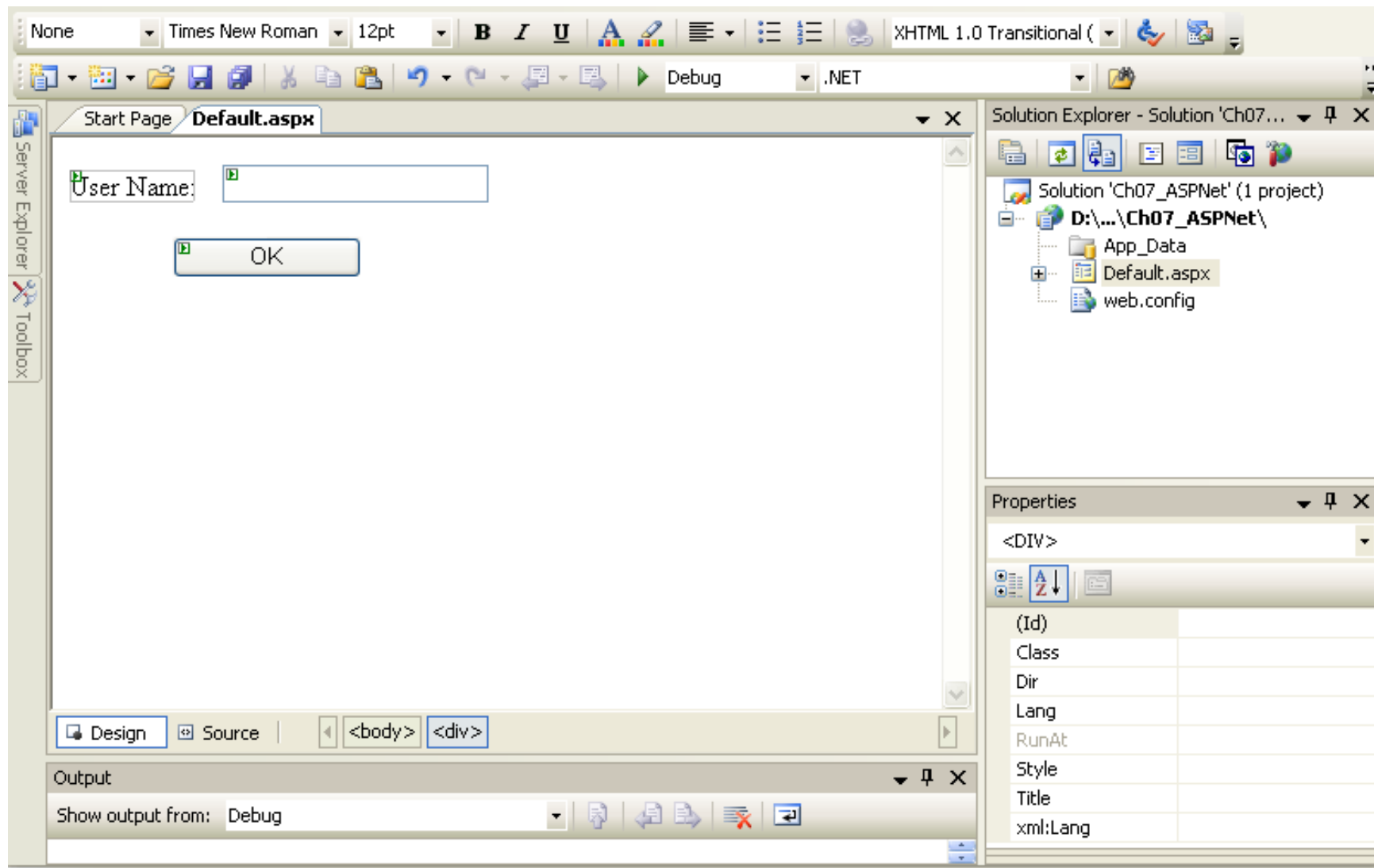


Thiết kế Web Forms





Cửa sổ VS.Net





ASP.Net tự tạo các file

Tên file	Mục đích
Web.Config	Thông tin cấu hình ứng dụng
Default.aspx	Các điều khiển sự kiện mức ứng dụng
App_Data	Thư mục chứa dữ liệu



Thiết kế Web Forms

- ❑ Trang .aspx cho phép bạn đặt các control ở Toolbox trực tiếp lên form bằng cách kéo thả.
- ❑ Mặc định trang hiển thị ở chế độ lưới (grid layout) cho phép bạn đặt control trên Form đúng vị trí. Bạn có thể chuyển sang chế độ Flow layout thì sẽ cho phép bạn đánh trực tiếp lên form.



Thiết kế Web Forms

- ❑ Cách thay đổi: Trên form thiết kế, ấn phím F4, cửa sổ Property được chọn. Bạn tìm đến thuộc tính Layout để thay đổi giữa hai chế độ.
- ❑ Khi thiết kế trong chế độ Design, code HTML phát sinh tự động. Bạn có thể xem code HTML bằng cách ấn nút **Source** dưới cuối của trang.
- ❑ Bạn cũng có thể viết code trên cùng trang HTML



Thiết kế Web Forms

- ❑ Một file code – behind cũng tồn tại với trang này và không hiển thị trong cửa sổ *Solution Explorer*. Để hiển thị click vào icon *Show All File* trong cửa sổ *Solution Explorer* hoặc ấn F7 để xem code của trang đó.
- ❑ Bạn cần tìm hiểu các đoạn code tự động phát sinh khi thiết kế form có những gì trước khi bạn sửa nó:



Thiết kế Web Forms

❑ Trong chế độ hiển thị HTML

```
<%@ Page Language="vb" AutoEventWireup="false" CodeFile="default.aspx.vb"  
Inherits="DemoWeb.WebForm1"%>
```

- ❑ `@Page` chỉ dẫn các thuộc tính của Form và để các thuộc tính này có tác động lên trang của bạn.
- ❑ Thuộc tính `Language`: ngôn ngữ .Net hỗ trợ trên trang của bạn.
- ❑ Thuộc tính `AutoEventWireup`: là giá trị Boolean chỉ định các sự kiện của trang có tự động phát sinh sự kiện hay không, mặc định là `False`.
- ❑ Thuộc tính `CodeFile`: chỉ định file code-behind
- ❑ Thuộc tính `Inherit`: chỉ định tên trang mà class code-behind kế thừa



Thiết kế Web Forms

- ❑ Trong thẻ `<head>` của code HTML bạn cũng có thể viết code cho trang của bạn

```
<Script runat="server" language="vb">
```

```
'Code statements
```

```
</Script>
```

- ❑ Các controls hoặc text được add trong thẻ `<body>`, được nằm trong khối `<% %>`



Code-behind

```
Public Class WebForm1
    Inherits System.Web.UI.Page
    #Region "Web Form Designer Generated Code "
        'This call is required by the Web Form Designer.
        <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
            End Sub
        'NOTE: The following placeholder declaration is required by the Web Form Designer...
        Private designerPlaceholderDeclaration As System.Object
        Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Init...
    #End Region
    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'Put user code to initialize the page here
        Dim UserName As String
        UserName = Request.QueryString("Name")
        Response.Write("<Center>" + "<B>Welcome</B>" + "<B>" + UserName + "</B>" + "</Center>")
        Response.Write("<Center>" + "<B>This is your first Web application!!!</B>" + "</Center>")
    End Sub
End Class
```



Giải thích

❑ Public Class *WebForm1*

Inherits System.Web.UI.Page

❑ Class *WebForm1* kế thừa từ lớp *Page*

❑ Lớp *Page* nằm trong namespace *System.Web.UI*

❑ Hai phương thức *InitializeComponent* và *Page_Init*

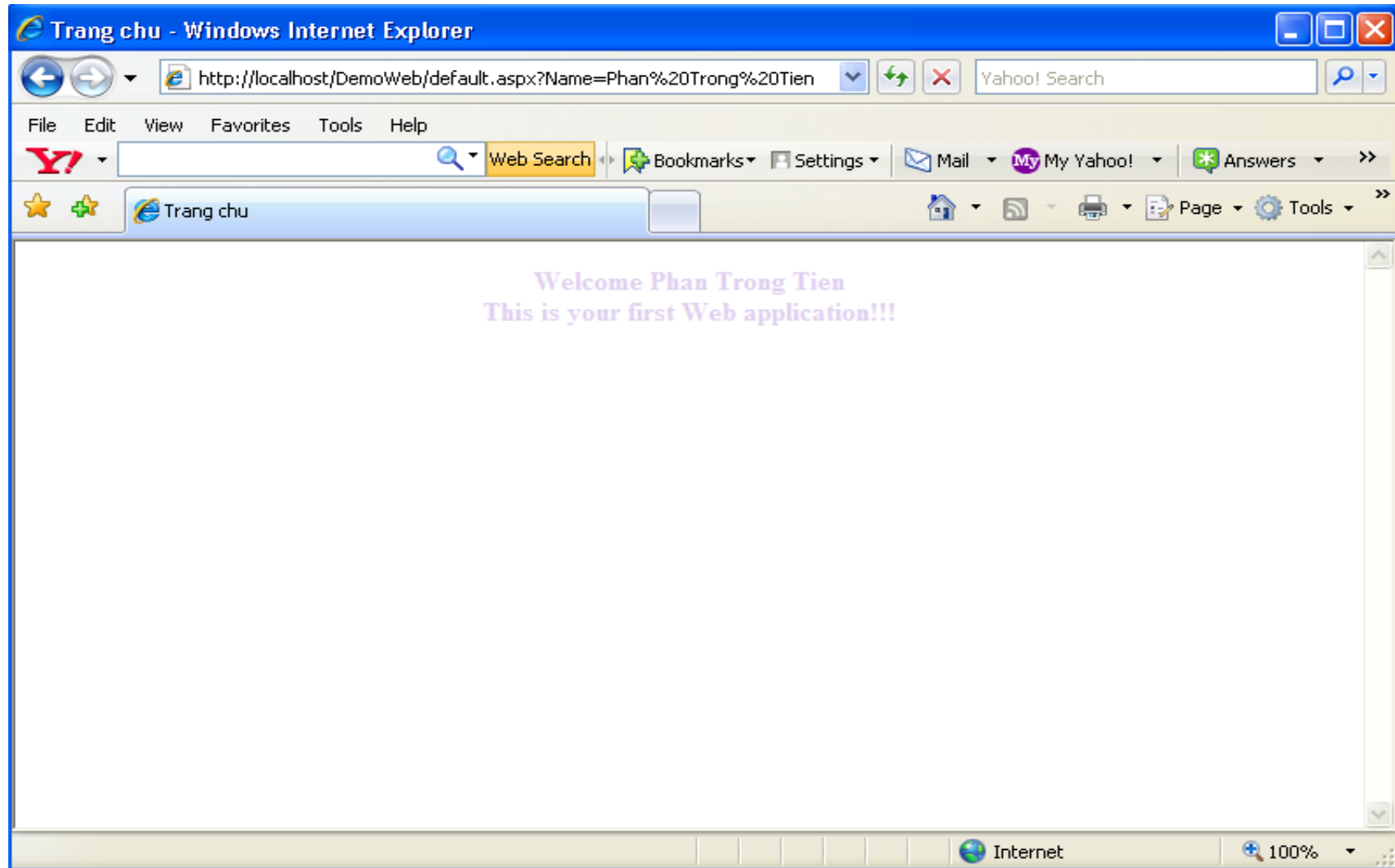
❑ *InitializeComponent*: bao gồm code khởi tạo cho trang như các control.

❑ *Page_Init* là sự kiện trang cho sự kiện *Init* của trang.

❑ Phương thức *Page_Load* dùng để điều khiển sự kiện *Load* của trang. Bạn cũng có thể thêm các sự kiện khác trên trang.



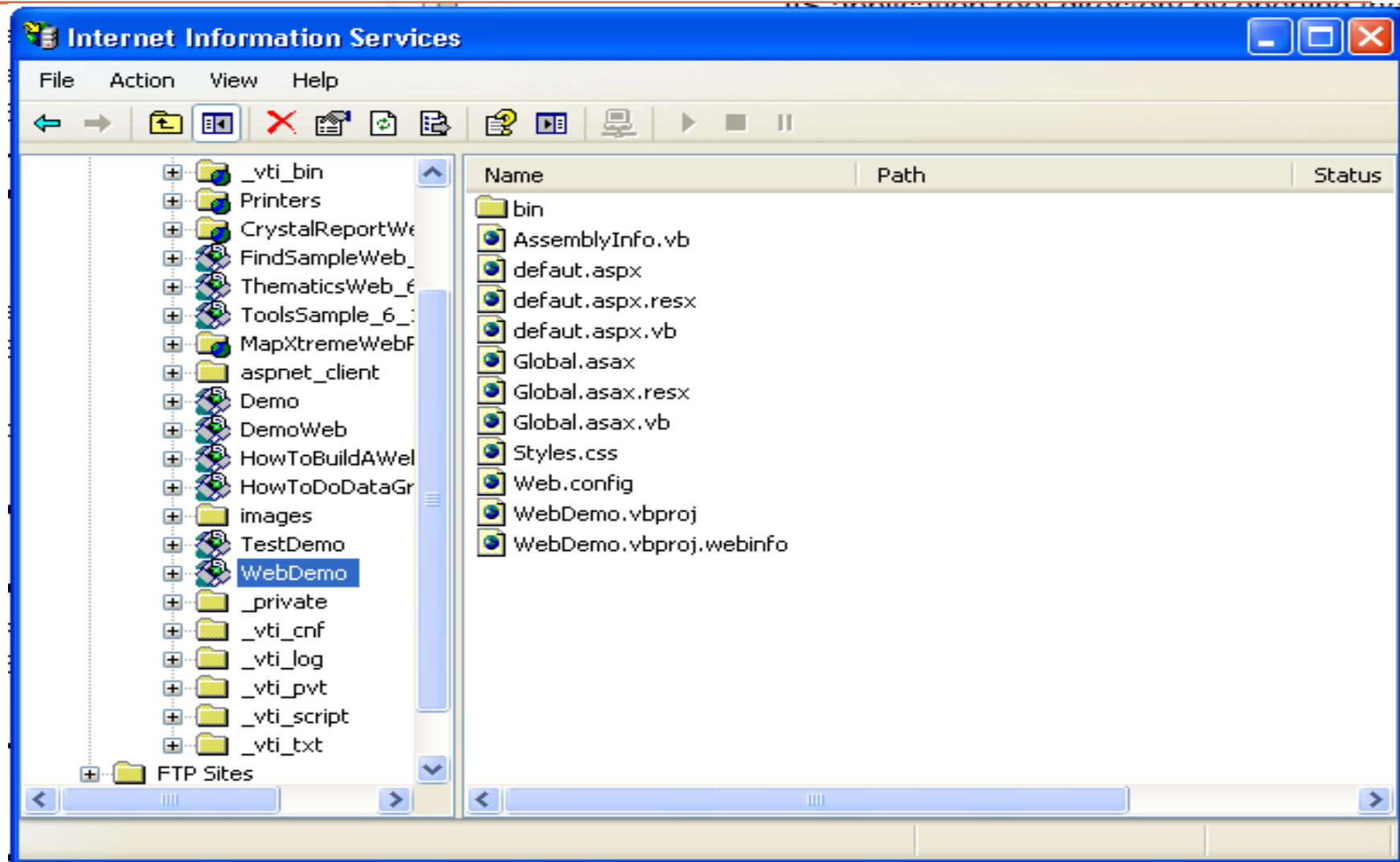
Chạy ứng dụng Ấn Ctrl + F5





Thư mục gốc ứng dụng IIS

- ❑ VS.Net khi tạo ứng dụng ASP.Net bạn cần chỉ định tên project và vị trí đặt ứng dụng như: **http://localhost** hoặc **http://<name of the computer>**.
- ❑ Vị trí đặt ứng dụng là tên máy tính và tham chiếu đến đường dẫn thư mục
- ❑ Khi phát triển ứng dụng ASP.Net thư mục gốc của ứng dụng được tạo trong Default Web Site của IIS
- ❑ Thư mục ứng dụng nằm tại:
<drive name>:\inetpub\wwwroot\<name of the project>.





III. Các Control Server của ASP.Net

- ❑ ASP.Net cung cấp cho bạn nhiều control tạo các trang web động và tương tác với người sử dụng.
- ❑ Các control chấp nhận mô hình lập trình phía server mà người sử dụng ở phía client tương tác với control server để phát sinh các sự kiện sẽ xử lý phía server.
- ❑ Trang này được biên dịch -> đối tượng gọi là **Page**
- ❑ Khi trang được yêu cầu các control server được biên dịch và thực thi trên server.



Server Controls khác gì HTML Controls thông thường?

- ❑ HTML Controls không có bất kỳ tương tác nào với server sau khi chúng hiển thị trên trang.
- ❑ Còn Server Controls cho phép truy cập các phương thức, các thuộc tính, các sự kiện tại phía Server.



Các kiểu Server Control

- ❑ .Net Framework hỗ trợ HTML server controls và Web server controls.
- ❑ HTML server controls: là các thẻ HTML bạn có thể sử dụng code phía server. Nằm trong namespace `System.Web.UI.HtmlControls` được lấy từ lớp cơ sở `HtmlControl`
- ❑ Web server controls: nằm trong namespace `System.Web.UI.WebControls`. Các control này cũng gọi là Web Controls.



Web control cũng bao gồm:

- ❑ *List controls: là các control tạo các danh sách(list). Ví dụ có thể sử dụng ListBox và DropDownList*
- ❑ *Validation controls: là các control kiểm tra và validate giá trị được nhập vào các control khác trên trang. Ví dụ RequiredFieldValidator và CustomValidator*
- ❑ *Rich controls: là các control đặc biệt được sử dụng tạo đầu ra cho các công việc đặc biệt. Ví dụ: Calendar và AdRotator*
- ❑ *User controls: bạn có thể tạo các control như các trang Web form và nhúng các control vào trang web khác.*



Bảng HTML server control và thẻ tương ứng

HTML server control	Thẻ HTML
<i>HtmlForm</i>	<code><form></code>
<i>HtmlInputText</i>	<code><input type = "text"></code> Và <code><input type = "password"></code>
<i>HtmlInputButton</i>	<code><input type="button"></code>
<i>HtmlInputCheckBox</i>	<code><input type = "check"></code>
<i>HtmlInputRadioButton</i>	<code><input type = "radio"></code>
<i>HtmlInputImage</i>	<code><input type = "image"></code>



Bảng HTML server control và thẻ tương ứng(tiếp)

HTML server control	Thẻ HTML
<i>HtmlAnchor</i>	<a>
<i>HtmlButton</i>	<button>
<i>HtmlTable</i>	<table>
<i>HtmlTableRow</i>	<tr>
<i>HtmlTableCell</i>	<td>



Sự khác nhau giữa HTML server controls và Web controls?

- ❑ Khi tạo ứng dụng web, bạn phân vân nên chọn HTML server controls hay Web controls?
- ❑ Phụ thuộc vào yêu cầu của bạn và chức năng của mỗi control. Bạn phải hiểu chức năng của cả hai loại server control.
- ❑ So sánh hai loại control:
 - ❑ HTML server controls và Web controls



Sự khác nhau giữa HTML server controls và Web controls?

- ❑ Ánh xạ tới thẻ HTML (Mapping to HTML tags):
 - ❑ HTML server controls ánh xạ trực tiếp tới thẻ HTML, nó được chuyển đổi thành server control bằng việc dùng thuộc tính *runat = "server"*.
 - ❑ Web control không ánh xạ trực tiếp tới thẻ HTML. Do đó bạn phải gọi (include) control từ bên thứ ba.
- ❑ Mô hình hướng đối tượng (Object Model):
 - ❑ HTML server control thiết lập các thuộc tính dùng cặp chuỗi tên/giá trị không định kiểu mạnh.
 - ❑ Web control thiết lập theo chuẩn property.
- ❑ Trình duyệt đích (Target browser):
 - ❑ HTML server control không thay đổi, phụ thuộc vào trình duyệt đích -> cần đảm bảo control trả về đúng với trình duyệt.
 - ❑ Web control trả về kết quả đầu ra, tự động điều chỉnh kết quả dựa vào trình duyệt đích -> chắc chắn control trả về đúng với trình duyệt



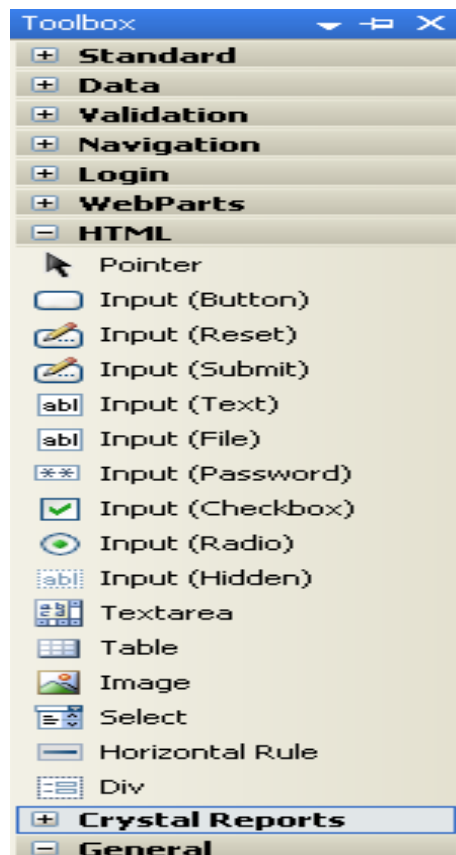
Thêm các web controls lên Form

- ❑ Thêm các server control lúc thiết kế hoặc lúc chạy.
- ❑ Thêm lúc thiết kế có thể dùng Toolbox hoặc ở chế độ HTML của trang .aspx
- ❑ Có thể thêm lúc chạy bằng việc dùng thẻ `<Script>` trong trang .aspx hoặc file code behind.



Dùng Toolbox

- ❑ Toolbox phân loại các control thành các nhóm thuận tiện cho việc truy cập.





Dùng ở chế độ hiển thị HTML

- ❑ Có thể thêm server control bằng cách chỉ định code ASP.Net trực tiếp trong chế độ hiển thị HTML của file .aspx.

```
<asp:TextBox id = "MyTextBox" runat = "server" Text =  
"Greetings"></asp:TextBox>
```



Dùng trong code behind

- ❑ ASP.Net cho phép bạn thêm các server control lúc chạy. Bạn có thể tạo một *instance* của lớp **Control** thừa kế từ lớp cơ sở **WebControl**
- ❑ Giả sử bạn muốn tạo một textbox lúc chạy:

Dim txt As New TextBox

txt.ID = "txtTest"

txt.Text = "ABC"

Panel1.Controls.Add(txt)

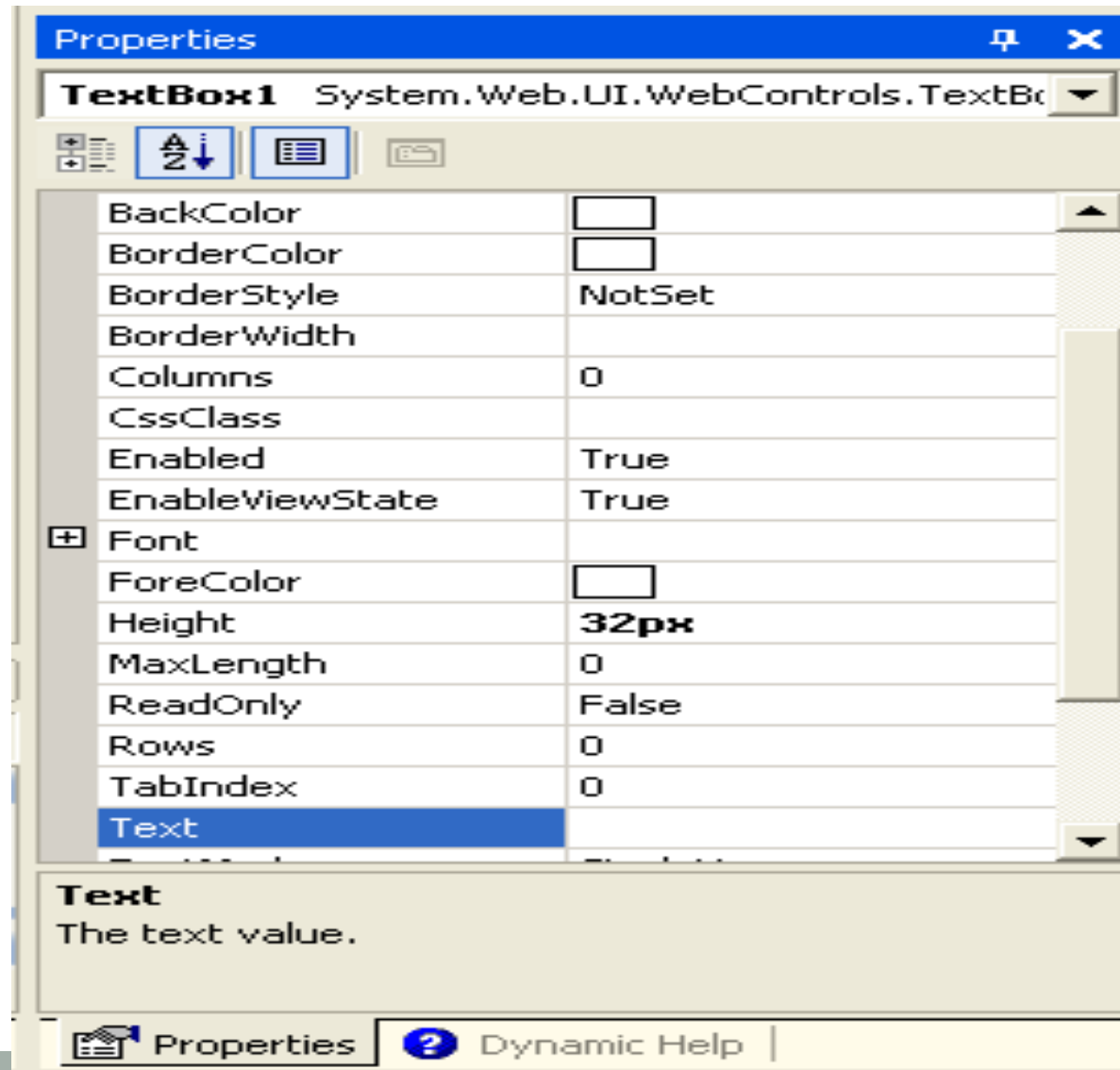


Thiết lập thuộc tính WebControl

- ❑ Các server control có các thuộc tính thông dụng kế thừa từ lớp cơ sở *WebControl*.
- ❑ Có thể thiết lập thuộc tính lúc thiết kế hoặc lúc chạy.
- ❑ Thiết lập thuộc tính của control lúc thiết kế bạn dùng cửa sổ **Properties**
- ❑ Hiện thị cửa sổ Property của control, kích chuột phải trên control rồi chọn Properties từ context menu hoặc ấn **F4**



Cửa sổ Properties của Textbox





Thiết lập thuộc tính WebControl

- ❑ Cũng có thể thiết lập các thuộc tính của WebControl trực tiếp trong chế độ hiển thị HTML.

```
<asp:TextBox Id = "Text_Box" runat = "server" Enabled =  
False></asp:TextBox>
```

- ❑ Đôi khi cần thiết lập thuộc tính lúc chạy.
Cú pháp sử dụng trong lập trình:

```
ControlID.PropertyName = Value
```

Ví dụ: `txtTest.Enabled = True`



Các sự kiện điều khiển trong WebControl

- ❑ Mỗi control có các sự kiện riêng có thể điều khiển trong code behind của trang .aspx
- ❑ Ví dụ: nút button có sự kiện click

```
Private Sub btnAccept_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles AcceptButton.Click
    'Code statements
End Sub
```
- ❑ Để mở file code behind có thể dùng cửa sổ Solution Explorer hoặc ấn F7



Các sự kiện điều khiển trong WebControl

❑ Cũng có thể tạo các sự kiện server control trong file .aspx

1. `<asp:Button Id = "btnAccept" runat = "server" OnClick = "btnAccept_Click"></asp:Button>`
2. Viết sự kiện trong trang Aspx

```
<script language="vb" runat="server">
```

```
Sub btnAccept_Click(sender as Object, e as  
    EventArgs)
```

```
    'Code comes here
```

```
End Sub
```

```
</script>
```



Các thuộc tính, phương thức, sự kiện hay sử dụng trong WebControl

Property/Method/Event	Mô tả
<i>AccessKey</i> property	Get hoặc Set shortcut để truy cập tới control
<i>BackColor</i> property	Get hoặc Set màu nền
<i>BorderColor</i> property	Get hoặc Set Border màu border
<i>BorderStyle</i> property	Get hoặc Set Border style
<i>Controls</i> property	Trả về một đối tượng của lớp <i>ControlCollection</i>
<i>DataBind</i> method	Gắn data từ nguồn dữ liệu tới server control



Các thuộc tính, phương thức, sự kiện hay sử dụng trong webcontrols

Property/Method/Event	Mô tả
<i>DataBinding</i> event	Phát sinh khi control chuẩn bị tương tác với DataSource
<i>Enabled</i> property	Chấp nhận gt Bool chỉ định khi nào control được enabled
<i>EnableViewState</i> property	Chấp nhận gt Bool chỉ định control có duy trì trạng thái khi hiển thị
<i>Font</i> property	Get hoặc Set Font
<i>ForeColor</i> property	Get hoặc Set màu chữ
<i>Height</i> property	Get hoặc Set chiều cao control



Các thuộc tính, phương thức, sự kiện hay sử dụng trong webcontrols

Property/Method/Event	Mô tả
<i>ID</i> property	Get hoặc Set nhận biết control
<i>TabIndex</i> property	Get hoặc Set TabIndex
<i>ToolTip</i> property	Get hoặc Set ToolTip
<i>Visible</i> property	Chỉ định control hiển thị hoặc không hiển thị trên UI
<i>Width</i> property	Get hoặc Set độ rộng



Các control validation

- ❑ Dữ liệu người sử dụng phải được validation trước khi insert vào CSDL
- ❑ ASP.Net, việc validate dùng các control validation -> bạn không cần phải viết code validation.
- ❑ Ví dụ: bạn muốn validation dữ liệu nhập vào textbox bạn có thể add control validation và kết hợp với control bạn cần validate.

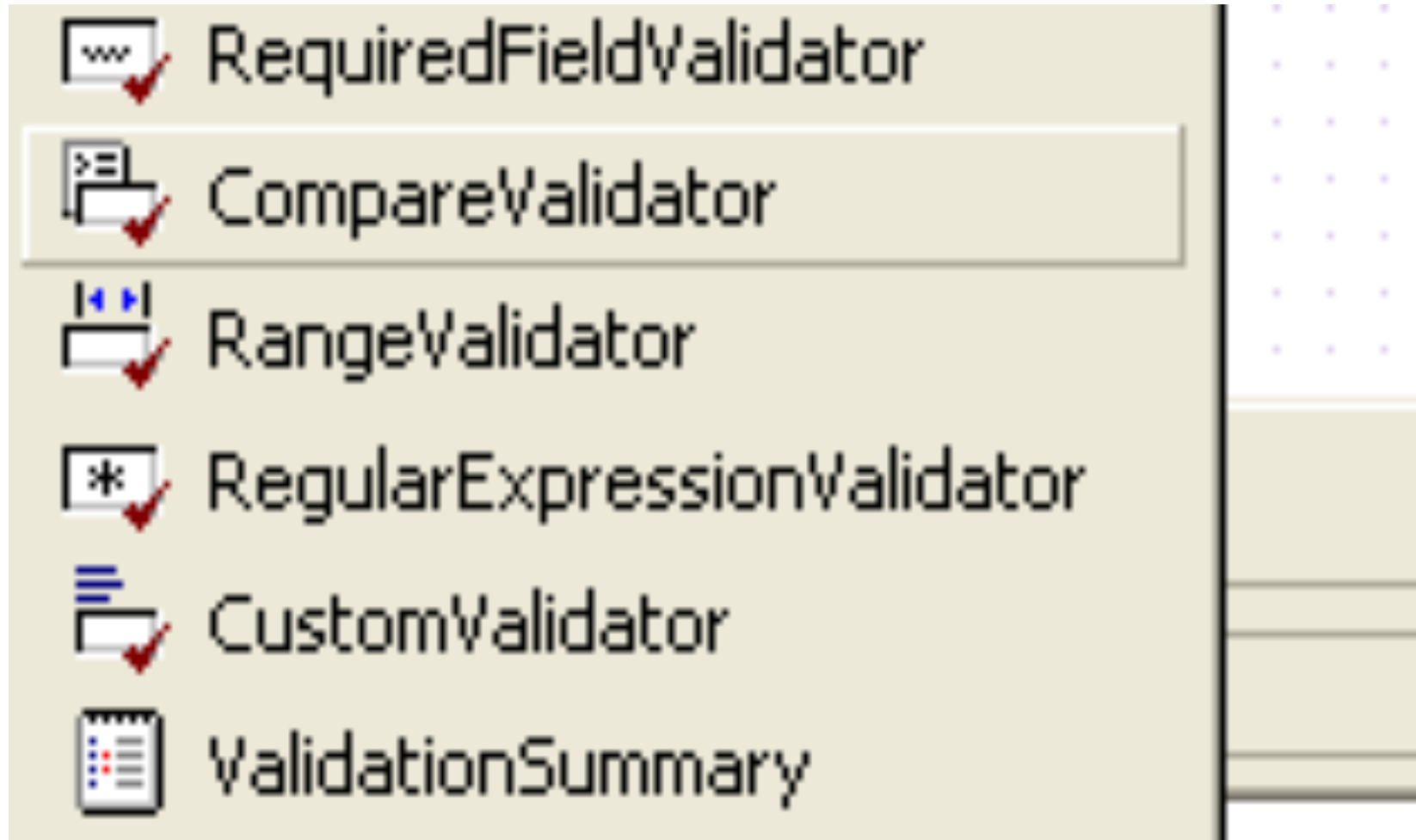


Các control validate gì?

- ❑ ASP.Net cung cấp 6 control validate, tất cả đều kế thừa từ lớp cơ sở *BaseValidator*
- ❑ *RequiredFieldValidator*: kiểm tra control là rỗng hay không.
- ❑ *CompareValidator*: so sánh giá trị các control có khớp nhau hay không.
- ❑ *RangeValidator*: kiểm tra giá trị control là trong khoảng text hay khoảng của số.
- ❑ *RegularExpressionValidator*: kiểm tra giá trị control khớp biểu thức biểu thức chính qui.
- ❑ *CustomValidator*: thi hành validate do người dùng tự định nghĩa.
- ❑ *ValidationSummary*: hiển thị tất cả các lỗi trong nhóm



Các control validation





Cách làm để validation: 4 bước

- ❑ Add control validation. Kết hợp nó với control cần validation. Đặt thuộc tính *ControlToValidate* là control này.
- ❑ Gắn các control validations và các control cần validate. Các control validation gắn vào control cần validation tương ứng giá trị nhập vào control này.
- ❑ Các control validation hợp lệ khi các giá trị thỏa mãn yêu cầu control.
- ❑ Khi kết thúc validation, giá trị trả về qua thuộc tính *IsValid* bằng **True** hoặc **False**. Sau khi validation thành công, trang gửi dữ liệu lên server.



Cách dùng Validate control

- ❑ Tham khảo ví dụ dùng validate control
- ❑ Cách dùng các control validation sẽ kiểm tra tại server. Một cách khác bạn có thể validation tại client bằng việc dùng script.
- ❑ Validation tại client sẽ giảm thời gian quay vòng và cải thiện hiệu năng. Các lỗi sẽ hiển thị ngay khi người sử dụng dữ liệu không hợp lệ.



Thiết kế DemoWeb

Control	Text	ID
Button	Accept	btnAccept
Label	Message	lblMessage
Text box	Confirm Password	txtConfirm
Text box	Customer ID	txtCusID
Text box	Number Of CDs	txtNumCDs
Text box	Password	txtPassword
Text box	Telephone Number	txtTelNumber
Text box	Username	txtUserName



RequiredFieldValidator

- ❑ Dùng kiểm tra giá trị của control có hợp lệ hay không. Bảng thuộc tính

Property	Value
ID	rfvUserName
ControlToValidate	txtUserName
ErrorMessage	Nhập vào Username
Display	Dynamic



Code

```
Private Sub btnAccept_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
btnAccept.Click
```

```
    If Page.IsValid = True Then
```

```
        'Specify a message to be displayed on the label
```

```
        lblMessage.Text = "Welcome " + txtUserName.Text
```

```
        'Make the label visible
```

```
        lblMessage.Visible = True
```

```
    End If
```

```
End Sub
```



CompareValidator

- ❑ Dùng để so sánh giá trị nhập bởi control này với một control khác hoặc một giá trị có sẵn.
- ❑ Các toán tử: *Equal*, *NotEqual*, *GreaterThan* và *LessThan* cho phép so sánh các kiểu: *String*, *Integer*, *Double*, *Date* và *Currency*



Bảng thuộc tính CompareValidator

Thuộc tính	Mô tả
ControlToValidate	
ControlToCompare	
Display	
ErrorMessage	
Operator	
Type	
ValueToCompare	



Code HTML

```
<asp:CompareValidator id="cpvPassword" style="Z-  
INDEX: 116; LEFT: 392px; POSITION: absolute; TOP:  
64px"  
        runat="server" ErrorMessage="Please  
retype the password" ControlToValidate="txtConfirm"  
        Display="Dynamic"  
        ControlToCompare="txtPassword">  
</asp:CompareValidator>
```



RangeValidator

- ❑ Dùng để kiểm tra giá trị nằm trong một khoảng xác định. Bạn có thể thiết lập giá trị max và min cho cho control. Cũng có thể giá trị hăng từ các control khác. Các thuộc tính thông dụng *ControlToValidate*, *ErrorMessage* và *Display*.
- ❑ Các thuộc tính khác: *MaximumValue*, *MinimumValue*, *Type*



RegularExpressionValidator

- ❑ Dùng để validate giá trị nhập vào control tuân theo một chuẩn nào đó ví dụ như số điện thoại, địa chỉ email, Zip codes ...
- ❑ Các thuộc tính thường sử dụng:
ControlToValidate, ErrorMessage, Display, ValidationExpression.
- ❑ Thuộc tính *ValidationExpression* được dùng nhập vào mẫu định dạng biểu thức cần kiểm tra xem có đúng định dạng không.



Ví dụ `RegularExpressionValidator`

- ❑ Thuộc tính: `ValidationExpression = [0-9]{3}-[0-9]{3}\s[0-9]{4}`
 - ❑ `[0-9]` nhận tất cả các số 0-9
 - ❑ `{3}` yêu cầu 3 số được nhập cho phần đầu tiên
 - ❑ `-` là dấu bắt nhập
 - ❑ `\s` chỉ định một khoảng trắng



CustomValidator

- ❑ Cho phép bạn validate một control theo kiểu người dùng tự định nghĩa.
- ❑ Đây là một control cho phép bạn validation ở phía client hoặc server.
- ❑ Thuộc tính *ClientValidationFunction* dùng để thiết lập hàm/thủ thực script.
- ❑ Phía Server, control cung cấp sự kiện *ServerValidate*, bạn phải viết code ở đây.



Validation Summary

- ❑ Dùng control này để hiển thị tất cả các lỗi trên trang.
- ❑ Bạn có thể hiển thị lỗi trong 1 list, dạng chỉ mục hoặc định dạng một đoạn văn.
- ❑ Cung cấp tùy chọn để hiển thị các lỗi trong một dòng hoặc dạng pop-up.



Các thuộc tính CustomerValidator

Thuộc tính	Mô tả
DisplayMode	List, BulletList, SingleParagraph
HeaderText	Text hiển thị ở title validation
ShowSummary	Chỉ định khi validation hiển thị ở dạng dòng. Mặc định là <i>True</i> .
ShowSummaryBox	Chỉ định validation hiển thị ở dạng Pop-up. Mặc định là <i>False</i> .



Dùng nhiều control validation

- ❑ Bạn có thể kết hợp nhiều control validation cho nhiều điều kiện.
- ❑ Ví dụ: Bạn cần dùng control validation để kiểm tra độ dài của Password xem người dùng có nhập đúng độ dài hay không, và bạn cũng muốn kiểm tra Password vừa chữ lại vừa có số thì bạn có thể kết hợp cả hai control validation: **RangeValidator, RegularExpressionValidator**



Tổng kết

- ❑ Chương này bao gồm cơ sở ASP.Net và các yêu cầu platform cho ứng dụng ASP.Net.
- ❑ Giới thiệu chi tiết về kiến trúc ASP.Net và giới thiệu các ứng dụng Web.
- ❑ Các kiểu server control và validation control mà bạn có thể sử dụng trên trang
- ❑ ASP.Net cung cấp cấu trúc framework cho bạn hoàn thành các ứng dụng Web