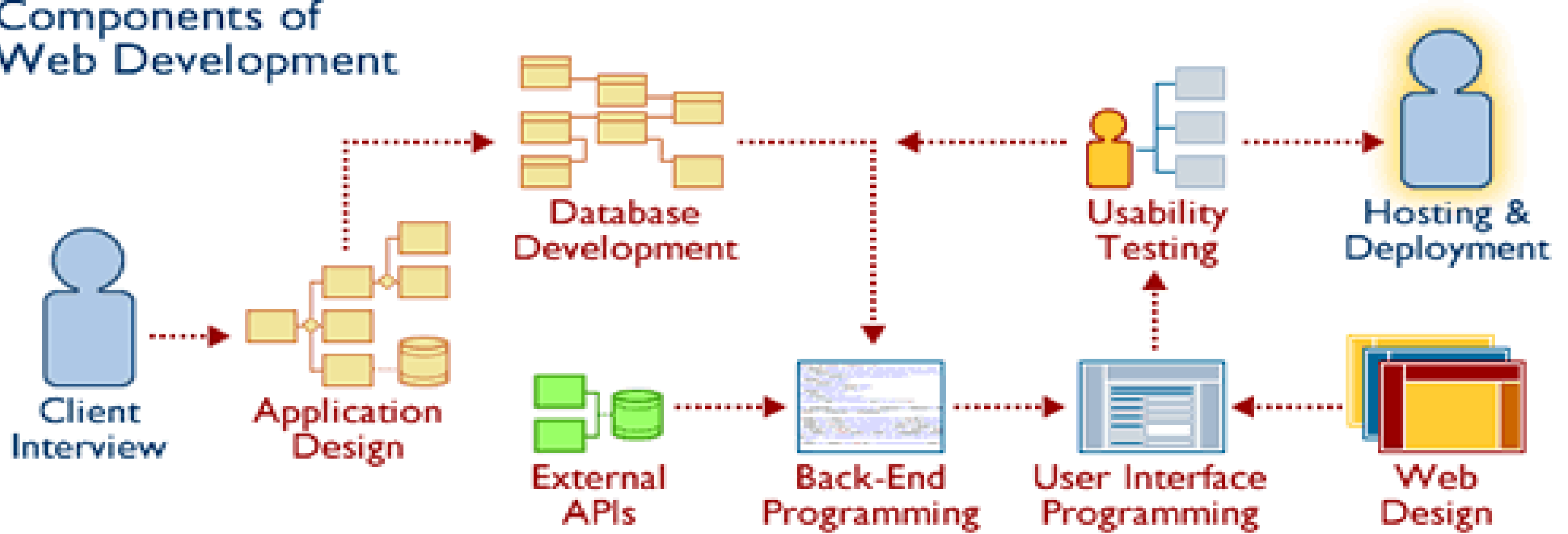


Components of Web Development



LẬP TRÌNH WEB – ASP.NET

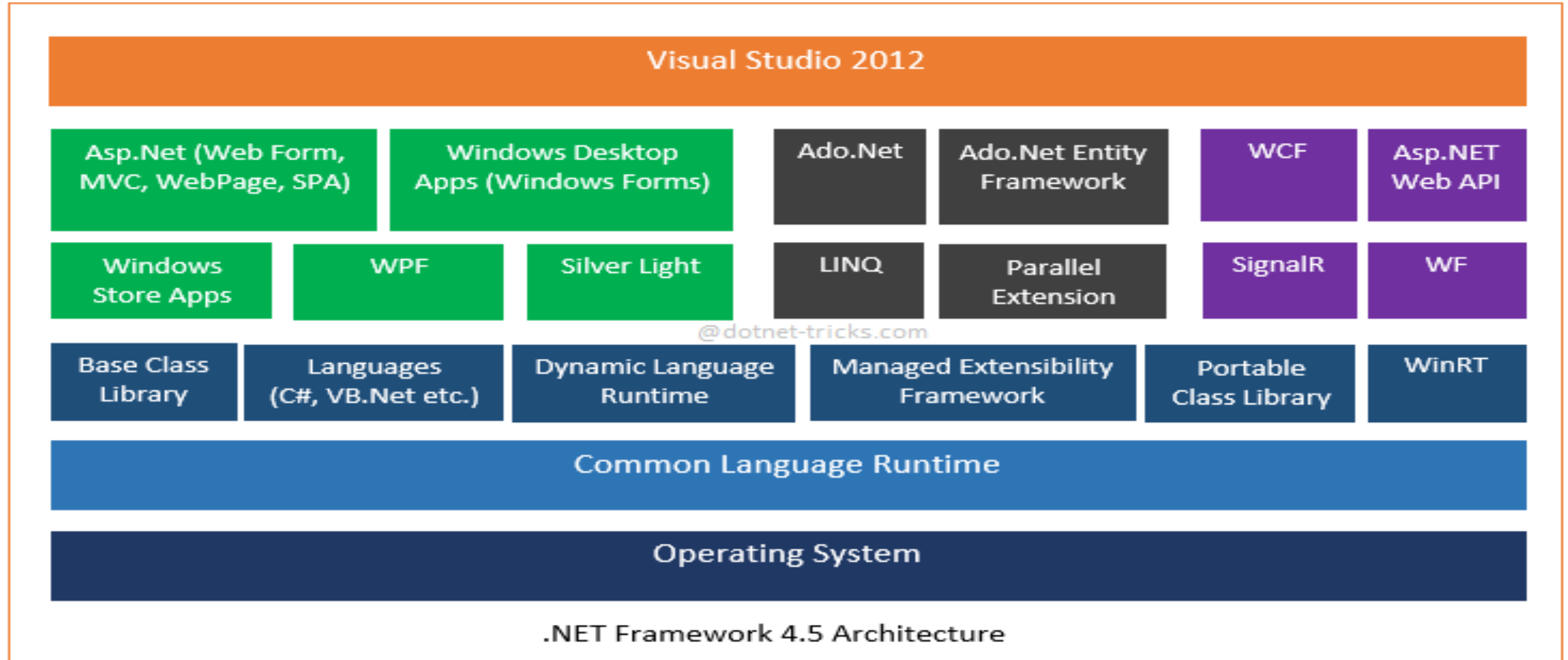
TRẦN QUANG DIỆU

NỘI DUNG

1. Giới thiệu.
2. Cấu trúc website bằng ASP.Net
3. Cơ bản về lập trình C# trong ASP.Net
4. Các điều khiển chuẩn (Standard Control).
5. Các đối tượng nâng cao trong ASP.Net

.NET FRAMEWORK

Common Language Runtime (CLR)
.Net Framework Class Library (Microsoft ASP)



CÁC NGÔN NGỮ KHẢ DỤNG VỚI .NET FRAMEWORK

.NET programming languages	
APL	Mondrian
C#	Oberon
COBOL	Oz
Component Pascal	Pascal
Curriculum	Perl
Eiffel	Python
Forth	RPG
Fortran	Scheme
Haskell	Smalltalk
Java	Standard ML
JScript	Visual Basic
Mercury	Visual C++

Các ngôn ngữ khả dụng với .Net Framework

GIỚI THIỆU ASP.NET

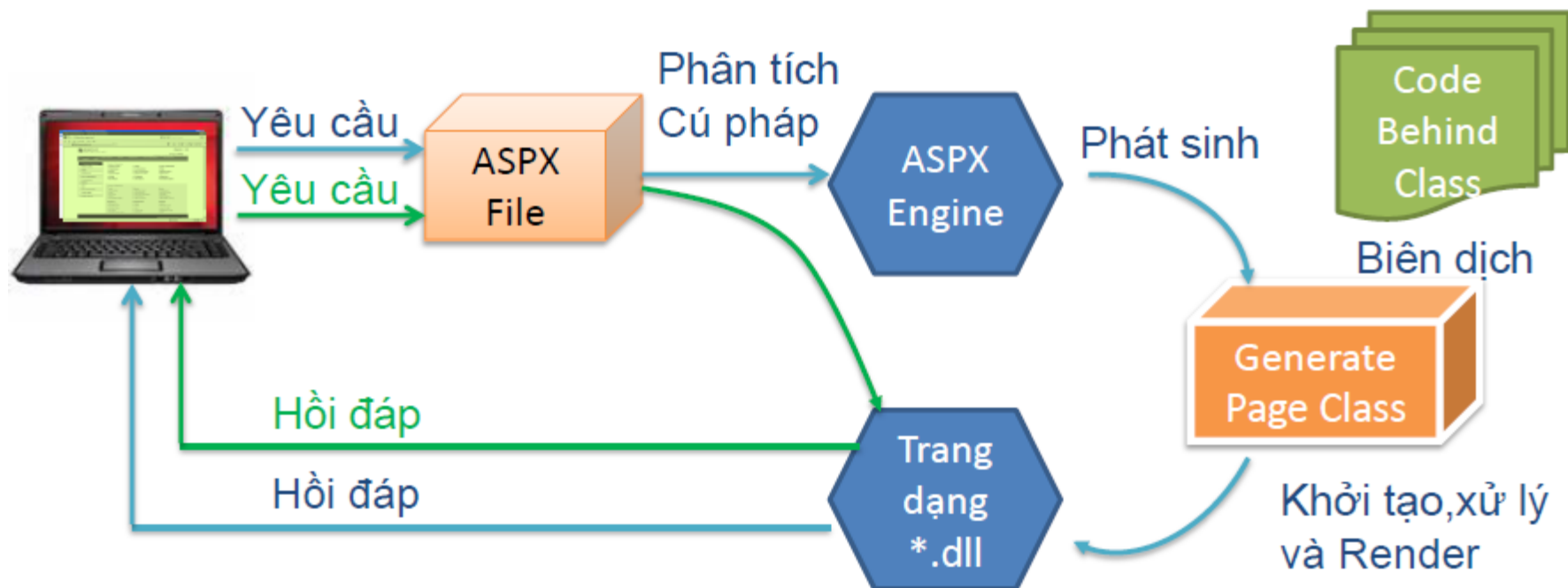
a. ASP.Net là gì?

- ASP.Net là kỹ thuật lập trình và phát triển ứng dụng Web ở phía Server (**Server-side**) trên môi trường Visual Studio.Net, dùng kết hợp các ngôn ngữ như VB.Net hoặc **C#** với HTML, Javascript, CSS (khác với kỹ thuật lập trình ở phía Client (**Client-side**) chỉ dùng các ngôn ngữ như HTML, Javascript và CSS..)

Yêu cầu: người học phải được trang bị kiến thức cơ bản về lập trình bằng ngôn ngữ **C#**.

CÁC ĐẶC ĐIỂM CỦA TRANG ASP.NET

- Một trang ASPX để tự động phát sinh mã HTML hiển thị trên Browser → trang Asp.Net đó cần được biên dịch trước thành **tập tin DLL** mà Server có thể thi hành.
- Trong cùng một ứng dụng, lập trình viên có thể sử dụng nhiều ngôn ngữ khác nhau để hoàn thành ứng dụng.



CẤU TRÚC WEBSITE

Một website ASP.Net được tạo ra sẽ cung cấp sẵn một số tập tin và thư mục dùng để phục vụ cho việc tổ chức lưu trữ một ứng dụng, gồm có:

- **Default page:** tự động thi hành khi người dùng nhập tên miền.
- **Web config:** chứa các khai báo mặc định của Web server.
- **Application folders:** dùng để lưu trữ các tập tin có phần mở rộng tương ứng với ý nghĩa của thư mục lưu trữ.

CẤU TRÚC WEBSITE

The image shows a screenshot of the Visual Studio IDE with several red annotations. The main window displays the 'Default.aspx' file in Design view. The Solution Explorer on the right shows the project structure: 'Solution 'Helloworld' (1 project)' containing 'D:\...\Helloworld\' with subfolders 'App_Data' and files 'Default.aspx', 'Default.aspx.cs', and 'web.config'. The Properties window at the bottom right shows the 'DOCUMENT' properties, with 'ALink' selected and its value 'Color of all active links in the document' visible. The Error List at the bottom left shows '0 Errors', '0 Warnings', and '0 Messages'. The status bar at the bottom left says 'Ready'.

Cấu trúc website ASP.Net

Cửa sổ viết code và design

Cửa sổ chứa các đối tượng và control

Cửa sổ thuộc tính

TÌM HIỂU MỘT TRANG ASP.NET

Một trang ASP.Net có phần đuôi mở rộng ***.aspx** và thông thường kèm theo một lớp phục vụ ẩn đằng sau (***code behind***) có phần đuôi mở rộng là ***.aspx.cs**

Để viết code (C#,VB.Net,..) xây dựng một trang web asp.net ta có thể thực hiện một trong các cách sau:

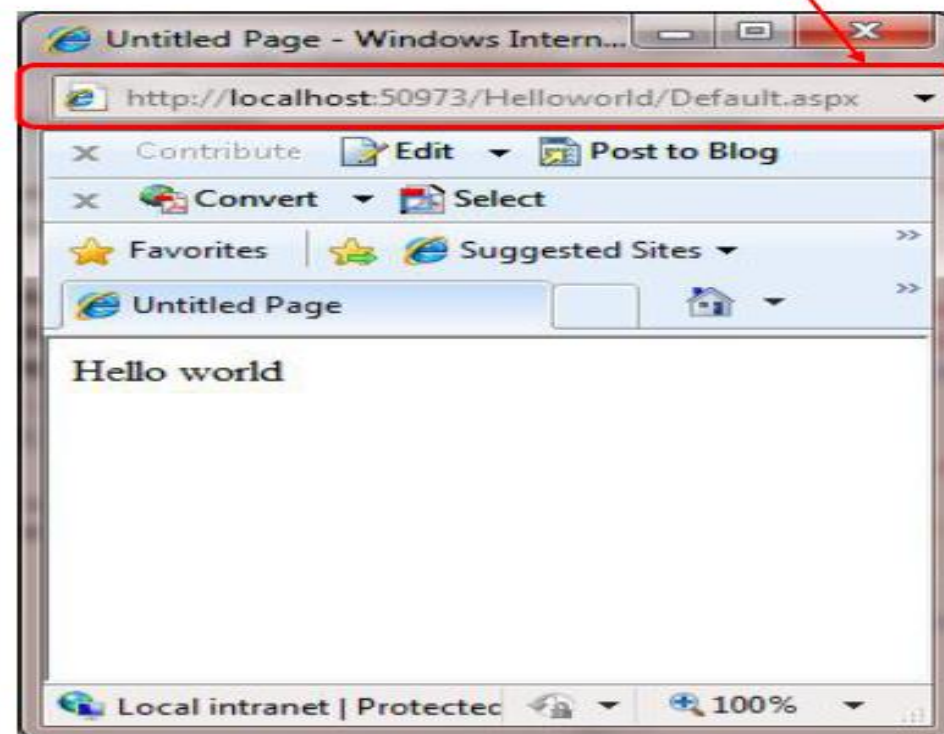
- **Viết code trực tiếp trong trang *.aspx:**
 - **<% %>**: khai báo biến hoặc viết các hàm, lớp trong cặp thẻ này.
 - **<%= %>**: lấy giá trị của biến hay của 1 hàm nào đó,
 - **<%# %>**: lấy giá trị các đối tượng ràng buộc dữ liệu.
- **Viết code trong trang code-behind *.aspx.cs (thường dùng)**

VÍ DỤ

Đoạn code bằng C# viết trực tiếp trong trang aspx

Địa chỉ thực hiện trang aspx

```
Default.aspx
Client Objects & Events (No Events)
<%@ Page Language="C#" AutoEventWireup="true" Code
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transi
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Untitled Page</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<% string abc = "Hello world";
Response.Write(abc);%>
</div>
</form>
</body>
</html>
```



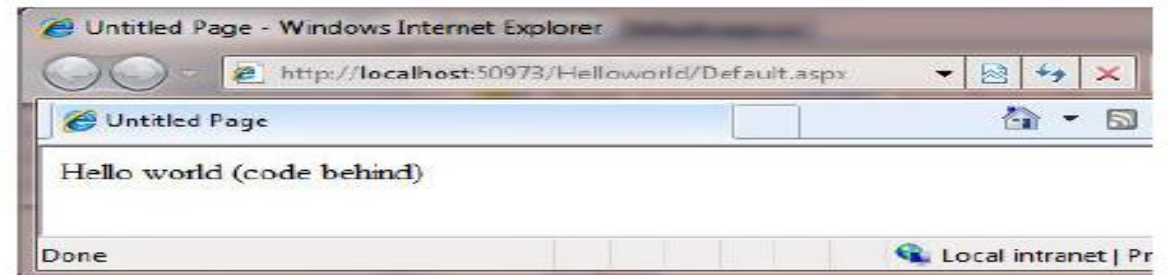
VÍ DỤ

```
Default.aspx
Client Objects & Events (No
<%@ Page Language="C#" AutoEventWireup="true"
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Untitled Page</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<%
Response.Write (abc) ;
%>
</div>
</form>
</body>
</html>
```

```
Default.aspx.cs
_Default Page_Load(object s
using System;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class _Default : System.Web.UI.Page
{
protected string abc;
protected void Page_Load(object sender, EventArgs e)
{
abc = "Hello world (code behind)";
}
}
```

KẾT QUẢ VẪN
KHÔNG ĐỔI



KIỂU DỮ LIỆU CƠ BẢN

KIỂU C#	KIỂU .NET	SỐ BYTE	MÔ TẢ
byte	Byte	1	số nguyên không dấu từ 0 đến 255
char	Char	2	Kiểu ký tự Unicode
bool	Boolean	1	Kiểu luận lý true/false
sbyte	Sbyte	1	Số nguyên có dấu, từ -128 đến 127
short	Int16	2	Số nguyên có dấu từ -32768 đến 32767
ushort			Số nguyên không dấu từ 0 đến 65.535
int	Int32	4	Số nguyên có dấu -2.147.483.647 đến 2.147.483.647
uint			Số nguyên không dấu 0 đến 4.294.967.295
float	Single	4	kiểu dấu chấm động, giá trị xấp xỉ từ $3,4E-38$ đến $3,4E+38$, với 7 chữ số có nghĩa.
double	Double	8	Kiểu dấu chấm động có độ chính xác gấp đôi, giá trị xấp xỉ từ $1,7E-308$ đến $1,7E+308$, với 15,16 chữ số có nghĩa
decimal	Decimal	8	Có độ chính xác đến 28 con số và giá trị thập phân, được dùng trong tính toán tài chính, kiểu này đòi hỏi phải có hậu tố m hoặc M kèm theo sau.

CÂU LỆNH XỬ LÝ

Khai báo biến – hằng: (tương tự như C/C++)

Các cấu trúc điều khiển: (tương tự như C/C++)

Cấu trúc điều kiện – rẽ nhánh:

if .. else

switch..case

Cấu trúc lặp:

for

while

do..while

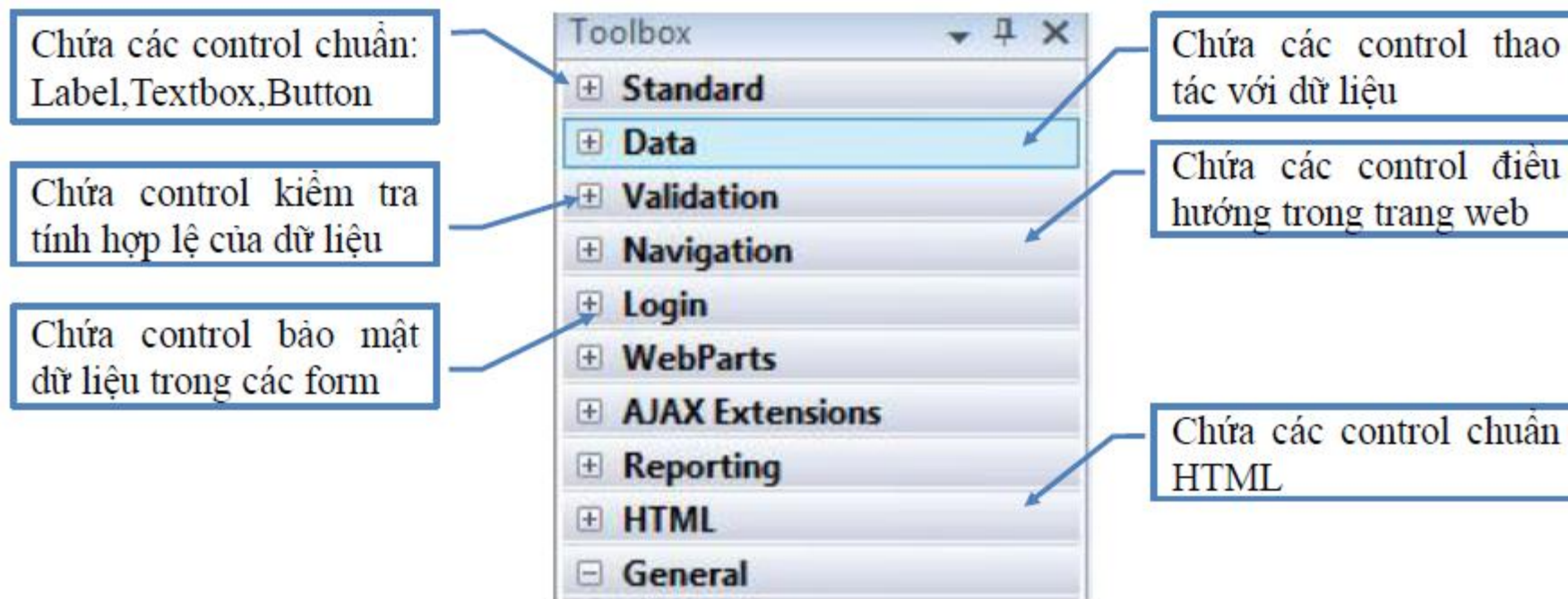
Hàm: (tương tự như C/C++)

Lưu ý: do ngôn ngữ C# là ngôn ngữ thuần hướng đối tượng nên cần lưu ý đến quyền truy cập của các thành viên.

CÁC ĐIỀU KHIỂN TRONG ASP.NET

Các điều khiển ASP.Net (**control**) là phần quan trọng nhất trong ASP.Net Framework (một control ASP.NET là **một lớp** thực thi trên server và đưa ra nội dung trên trình duyệt).

ASP.NET có hơn 70 control hỗ trợ xây dựng ứng dụng web, được chia ra các nhóm control sau:

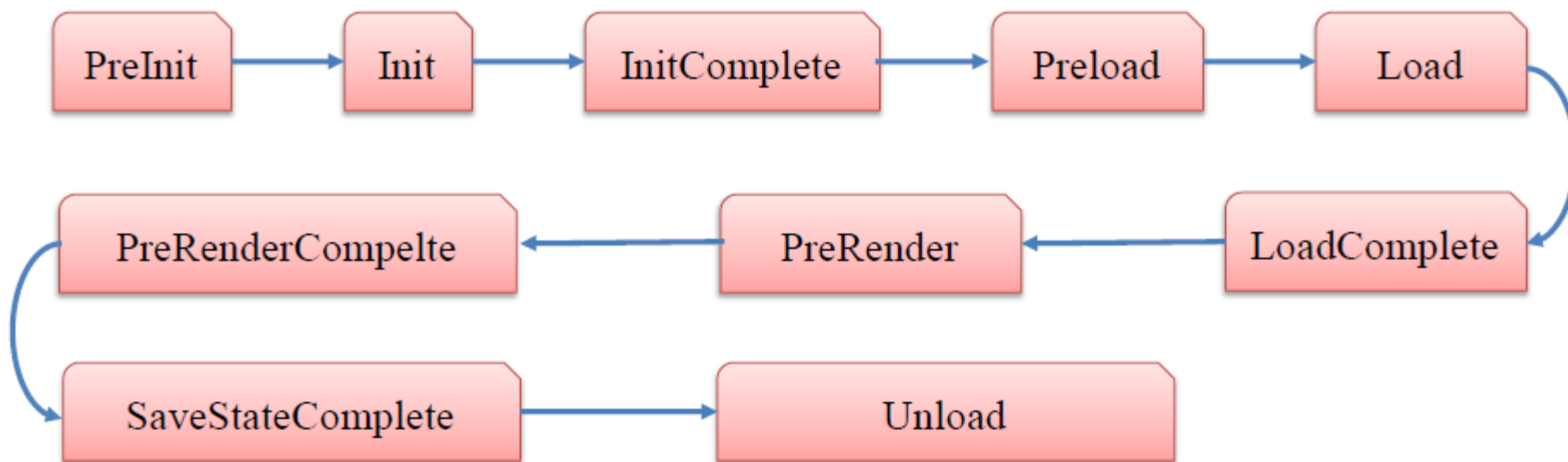


ĐIỀU KHIỂN SỰ KIỆN TRONG ASP.NET

Phần lớn các điều khiển của ASP.Net hỗ trợ 1 hoặc nhiều sự kiện.

Ví dụ: điều khiển ASP.NET Button hỗ trợ sự kiện **Click**, khi người sử dụng nhấn chuột vào Button một sự kiện sẽ được đưa ra và công việc này sẽ được xử lý trên server.

Một trang ASP.Net khi thực thi sẽ gồm có các sự kiện sau xảy ra:



VÍ DỤ

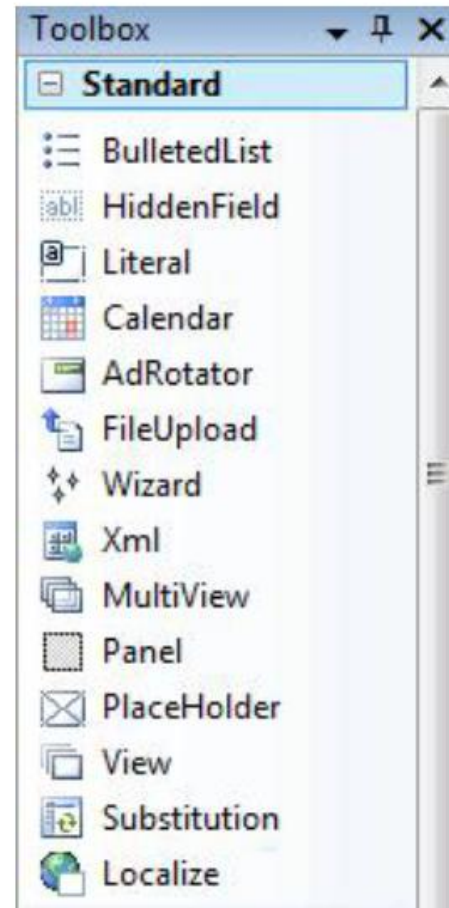
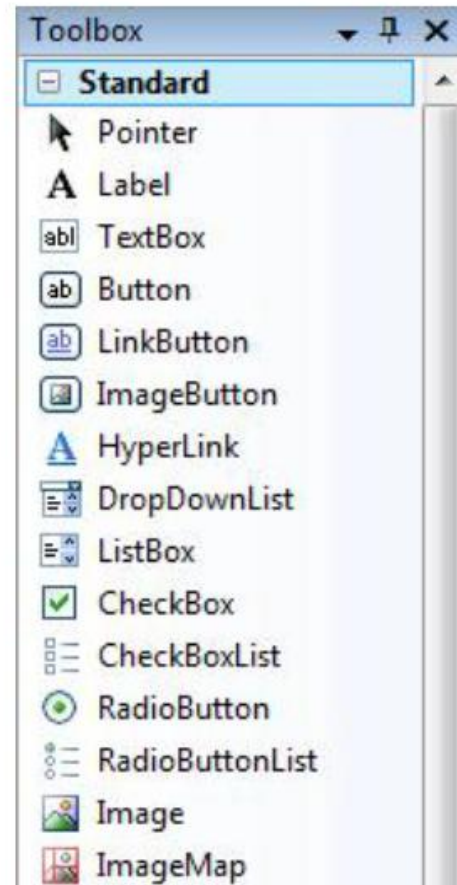
```
Default.aspx.cs | Default.aspx | Start Page
_Default
Page_PreInit(object sender, EventArgs e)
using System;
using System.Configuration;
using System.Data;
public partial class _Default : System.Web.UI.Page
{
    protected void Page_PreInit(object sender, EventArgs e)
    {
    }
    protected void Page_Init(object sender, EventArgs e)
    {
    }
    protected void Page_PreLoad(object sender, EventArgs e)
    {
    }
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Page_PreRender(object sender, EventArgs e)
    {
    }
    protected void Page_Unload(object sender, EventArgs e)
    {
    }
}
```

Sự kiện **Page_PreInit()** thường dùng kiểm tra trang có PostBack hay không?, tạo Theme động, Master động...

Sự kiện **Page_Init()** thường dùng để đọc hoặc đặt giá trị thuộc tính cho sự kiện.

Nếu có **PostBack** thì phát sinh sự kiện của điều khiển: như click của Button...

CÁC ĐIỀU KHIỂN CHUẨN



CONTROL: LABEL - TEXTBOX - BUTTON

Bước 1: Tạo project

Bước 2: Thiết kế form như hình

- ID Label: lContent
- ID Textbox: txtContent
- ID Button: bChange

Bước 3: Viết code xử lý sự kiện click cho nút Change như sau:

- Double click vào nút Change

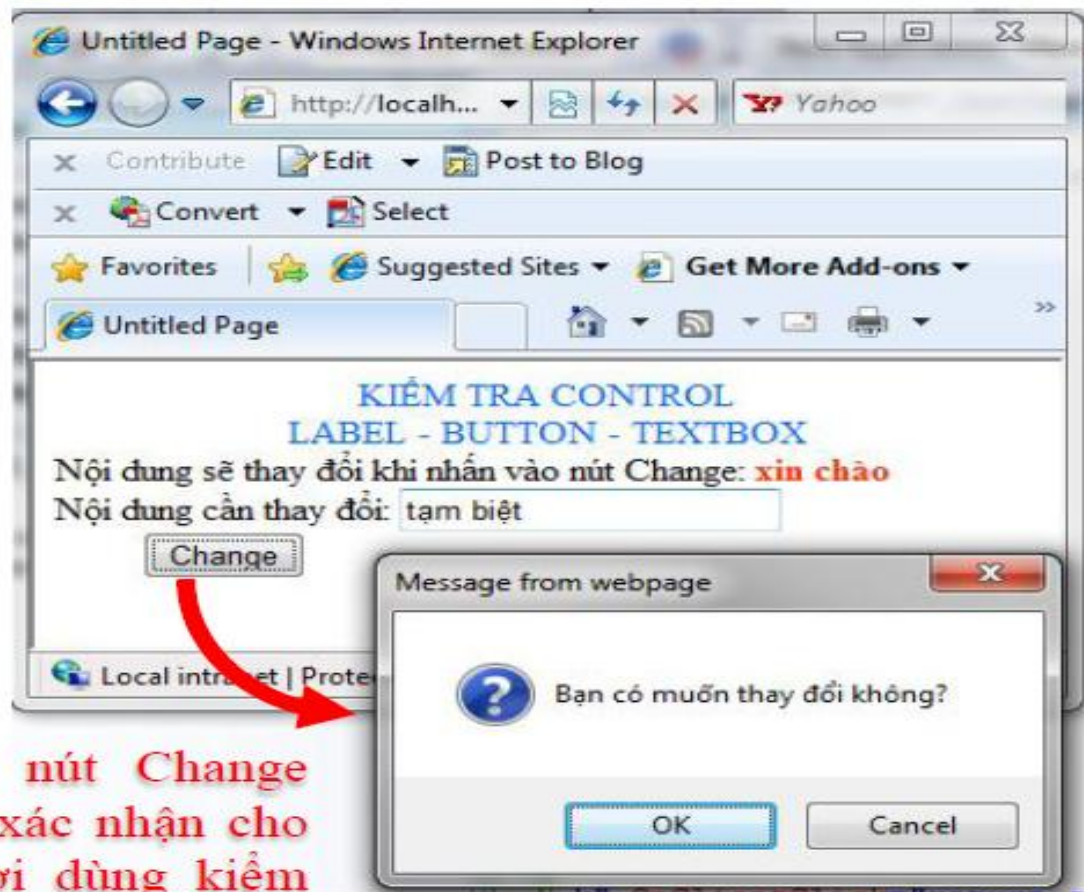
The screenshot shows a web browser window with the following content:

Untitled Page - Windows Internet Explorer
http://localh...
Yahoo
Untitled Page
KIỂM TRA CONTROL
LABEL - BUTTON - TEXTBOX
Nội dung sẽ thay đổi khi nhấn vào nút Change: **XIN CHÀO**
Nội dung cần thay đổi: XIN CHÀO
Change

A red arrow points from the 'Change' button to the code editor in Visual Studio. The code editor shows the following code:

```
Default.aspx.cs Default.aspx Start Page
_Default
using System;
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void bChange_Click(object sender, EventArgs e)
    {
        lContent.Text = txtContent.Text;
    }
}
```

BUTTON KẾT HỢP ASP.NET VÀ JAVASCRIPT

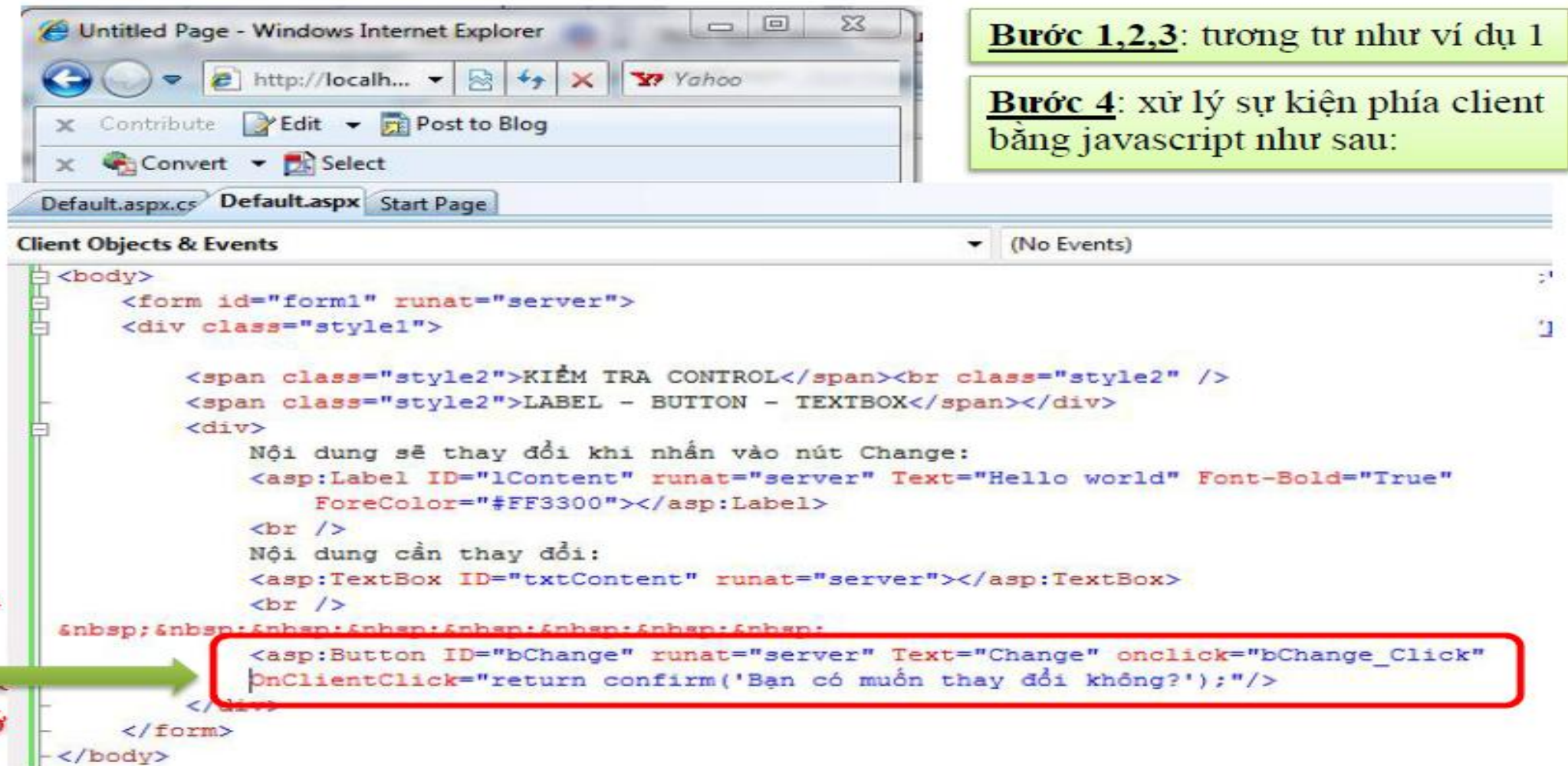


Bước 1,2,3: tương tự như ví dụ 1

Bước 4: xử lý sự kiện phía client bằng javascript như sau:

Khi nhấn nút Change hợp thoại xác nhận cho phép người dùng kiểm tra lại trước khi thay đổi

BUTTON KẾT HỢP ASP.NET VÀ JAVASCRIPT



The screenshot shows a web browser window with the address bar at `http://localhost...`. The page content includes a form with a button labeled "Change". The button's `onclick` event is set to `bChange_Click`, which calls a JavaScript function `confirm('Bạn có muốn thay đổi không?')`. The Client Objects & Events pane shows the code for the button.

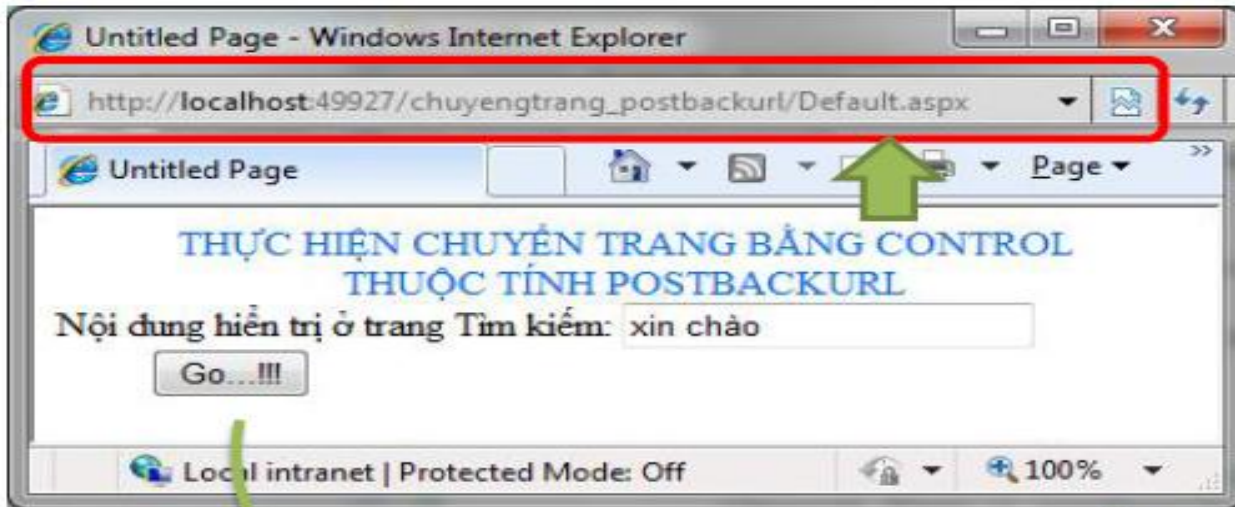
Bước 1,2,3: tương tự như ví dụ 1

Bước 4: xử lý sự kiện phía client bằng javascript như sau:

```
<asp:Button ID="bChange" runat="server" Text="Change" onclick="bChange_Click"
  OnClientClick="return confirm('Bạn có muốn thay đổi không?');"/>
```

Khi nhấn
hộp thoại
phép
tra lại trước

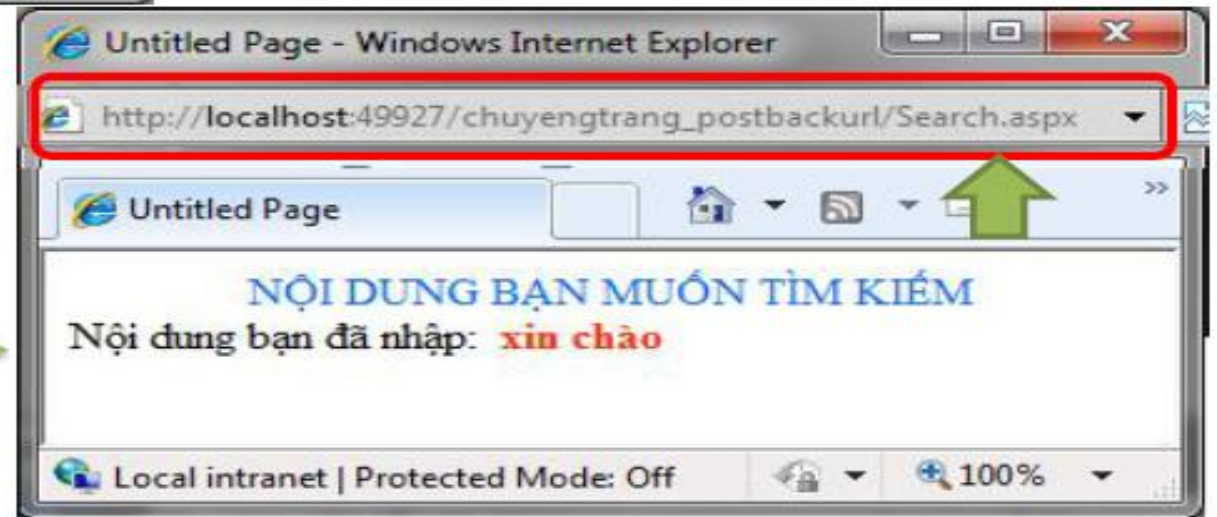
THỰC HIỆN CHUYỂN TRANG



Bước 1,2: tương tự như ví dụ 1, tạo project, cùng các control có ID như sau:

- Trang default.aspx:
 - ID TextBox: txtSearch
 - ID Button: bGo
- Trang search.aspx:
 - ID Label: lSearch

Bước 3: viết code cho nút Go trên trang default.aspx.



THỰC HIỆN CHUYỂN TRANG

The image shows a Visual Studio IDE with three main windows:

- Browser Preview:** Displays a page titled "THỰC HIỆN CHUYỂN TRANG BẰNG CONTROL THUỘC TÍNH POSTBACKURL". It contains a search form with a text box and a "Go...!!!" button. A red box highlights the URL in the address bar.
- Client Objects & Events:** Shows the HTML markup for the search form. A red arrow points from the "Go...!!!" button in the browser to the corresponding ASP.NET Button control in the code:

```
<asp:Button ID="bGo" runat="server" Text="Go...!!!" PostBackUrl="Search.aspx"/>
```
- search.aspx.cs:** Shows the C# code for the search page. A red arrow points from the "Bước 4" text box to the Page_Load method, which is highlighted with a green box:

```
protected void Page_Load(object sender, EventArgs e)  
{  
    if (PreviousPage != null)  
    {  
        TextBox txtSearch=(TextBox)PreviousPage.FindControl("txtSearch");  
        lSearch.Text = txtSearch.Text;  
    }  
}
```

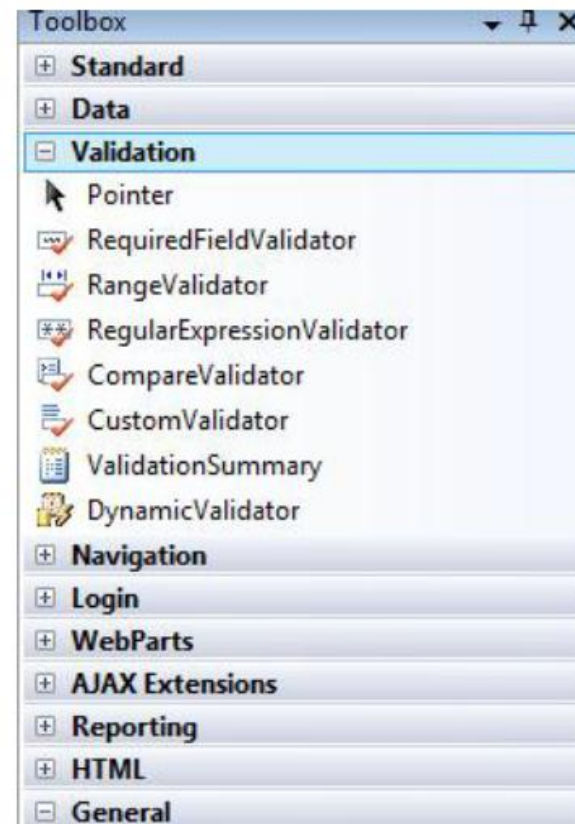
BƯỚC 3: viết code cho nút Go trên trang default.aspx.

BƯỚC 4: viết code cho trang search.aspx.

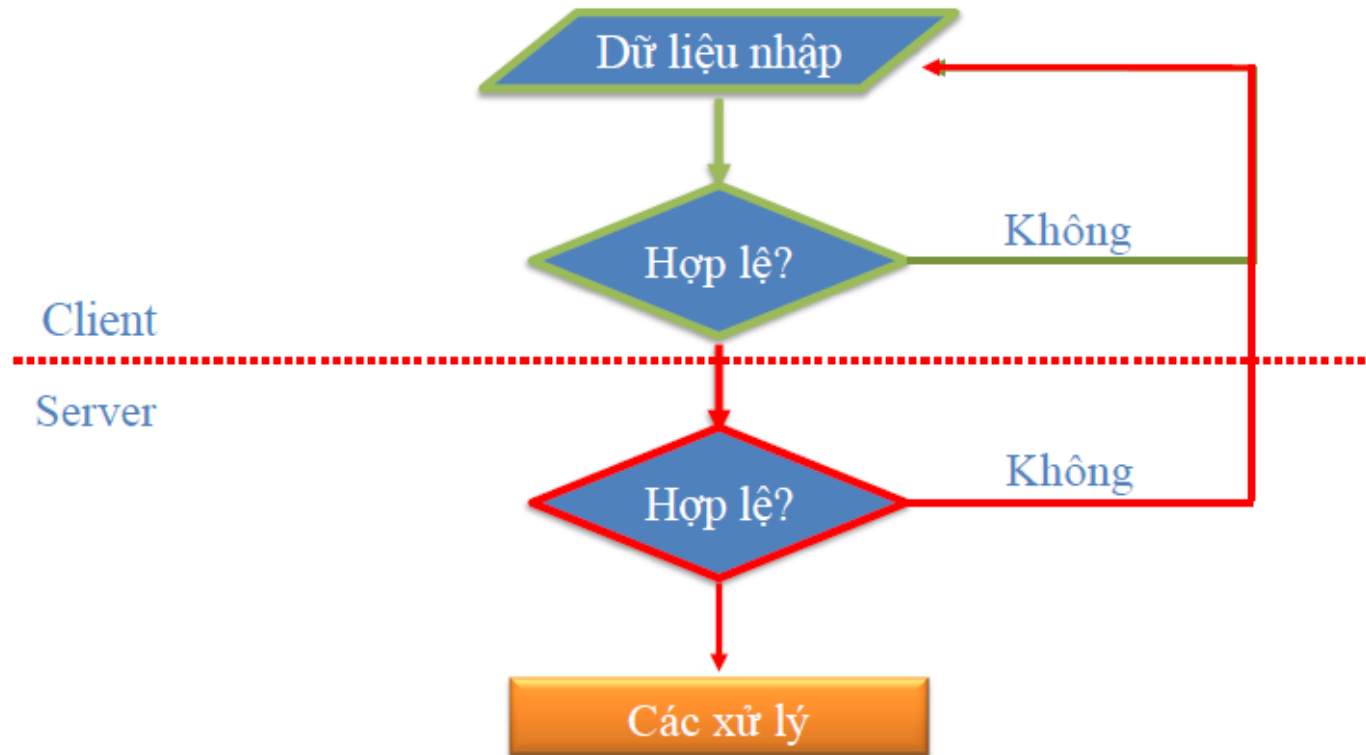
ĐIỀU KHIỂN KIỂM TRA DỮ LIỆU (VALIDATION)

Dùng để kiểm tra tính hợp lệ của dữ liệu.

Tự động phát sinh code kiểm tra dữ liệu ở client-side tùy thuộc vào web browser có hỗ trợ thực thi script ở client không.



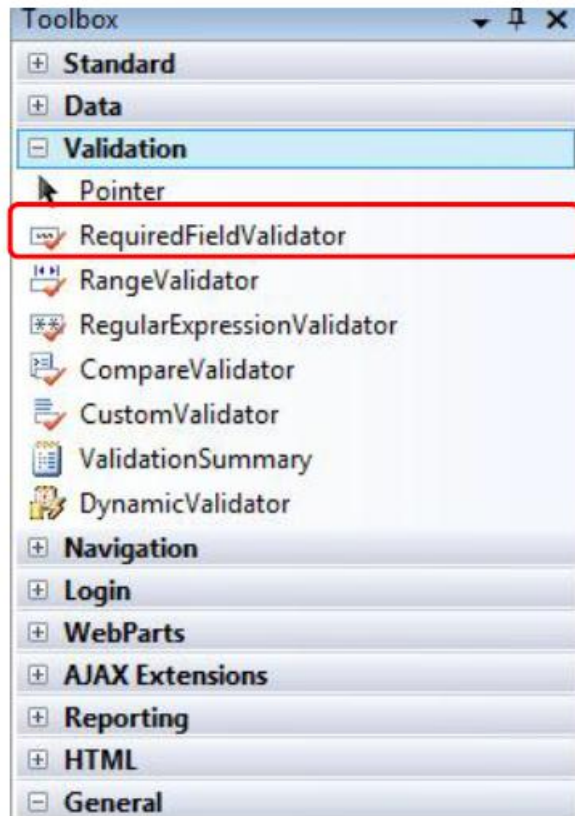
QUY TRÌNH KIỂM TRA DỮ LIỆU



MỘT SỐ THUỘC TÍNH CƠ BẢN CỦA VALIDATION

- `ControlToValidate`: tên điều khiển cần kiểm tra.
- `Text`: chuỗi thông báo xuất hiện khi có lỗi.
- `ErrorMessage`: chuỗi thông báo xuất hiện trong điều khiển `Validation Summary`.
- `EnableClientScript`: cho phép thực hiện kiểm tra ở phía Client hay không (`True/False`).
- `SetFocusError(True/False)`: đặt con trỏ vào điều khiển khi dữ liệu không hợp lệ.
- `ValidationGroup`: tên nhóm. Nhóm các điều khiển có cùng giá trị `ValidationGroup`.

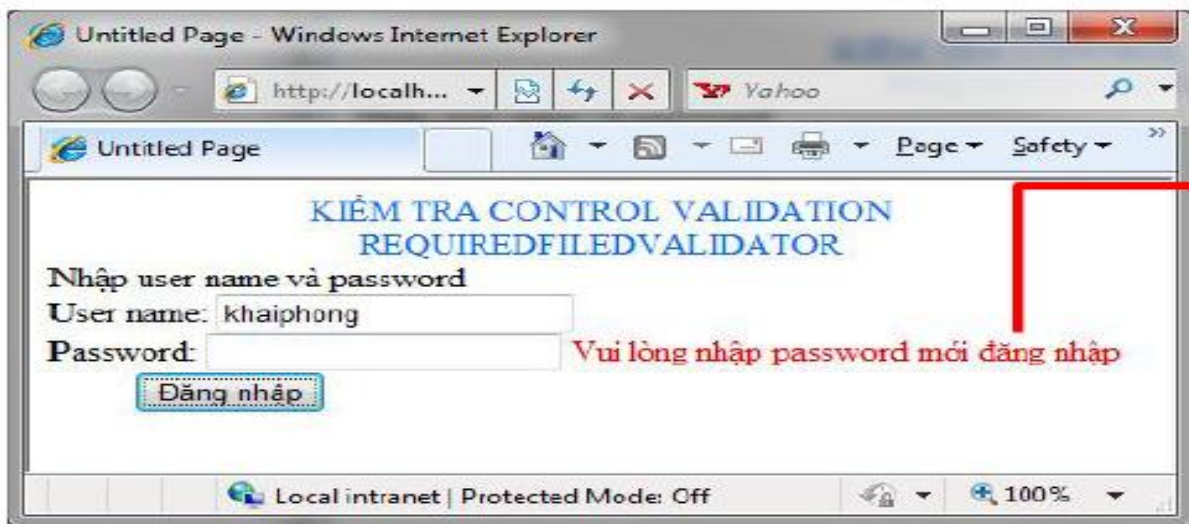
REQUIREDFIELDVALIDATOR



Kiểm tra giá trị trong điều khiển phải được nhập

- Thuộc tính InitialValue:
- Giá trị khởi động
- Giá trị nhập vào phải khác với giá trị của thuộc tính này

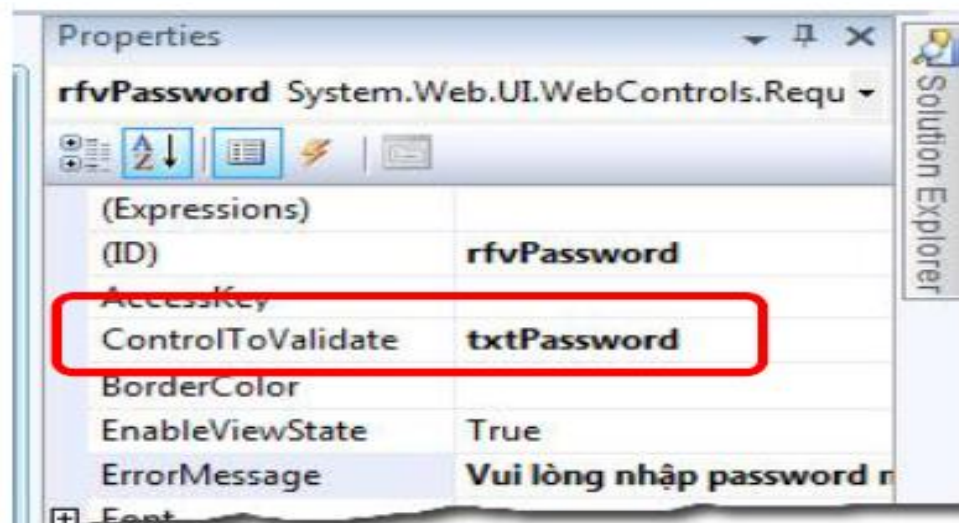
VÍ DỤ



Bước 1: tạo project như hình bên

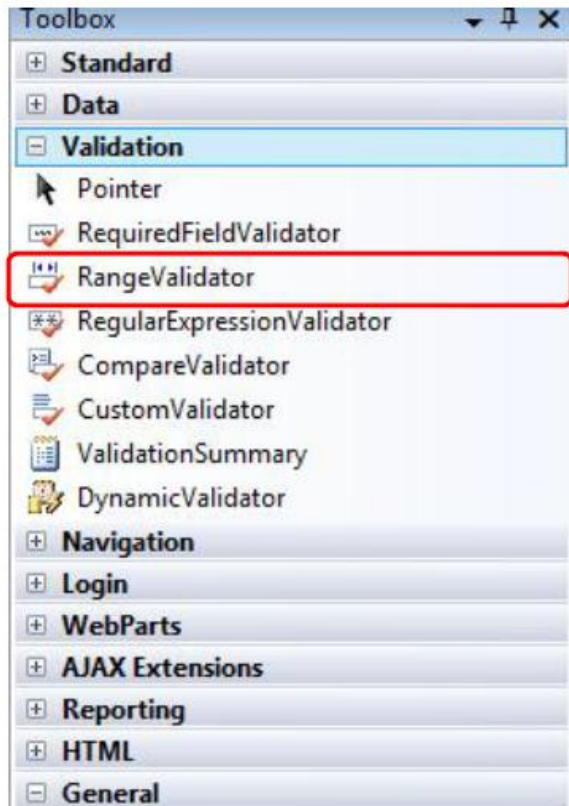
- ID userName: txtUser
- ID Password: txtPassword
- ID nút Đăng nhập: bLogin
- ID RequiredFieldValidator: rfvPassword

Bước 2: thiết lập thuộc tính cho rfvPassword



Gán control
muốn kiểm tra

RANGEVALIDATOR

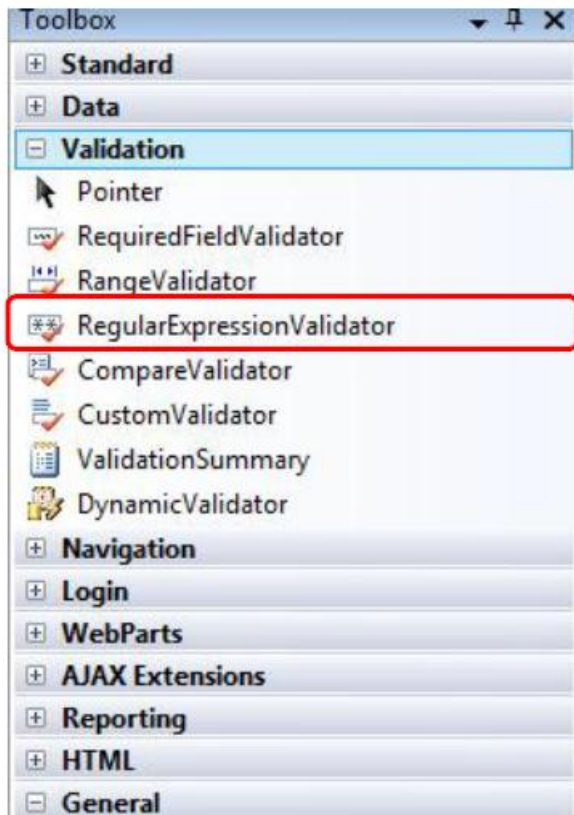


Kiểm tra giá trị của điều khiển nằm trong đoạn [min-max]

Thuộc tính:

- MinimumValue: giá trị nhỏ nhất
- MaximumValue: giá trị lớn nhất
- Type: xác định kiểu dữ liệu kiểm tra (String/Integer/Double/Date/Currency)

REGULAREXPRESSIONVALIDATOR

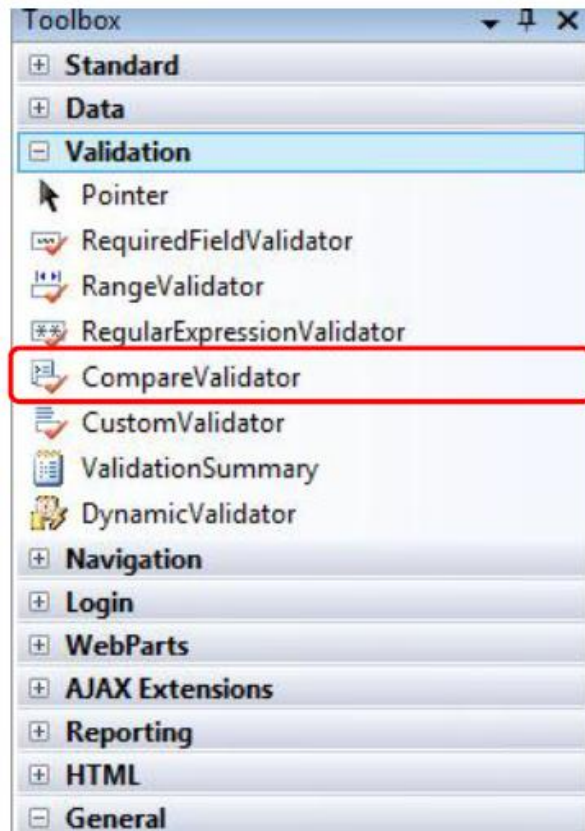


Kiểm tra giá trị của điều khiển phải theo mẫu qui định

Thuộc tính:

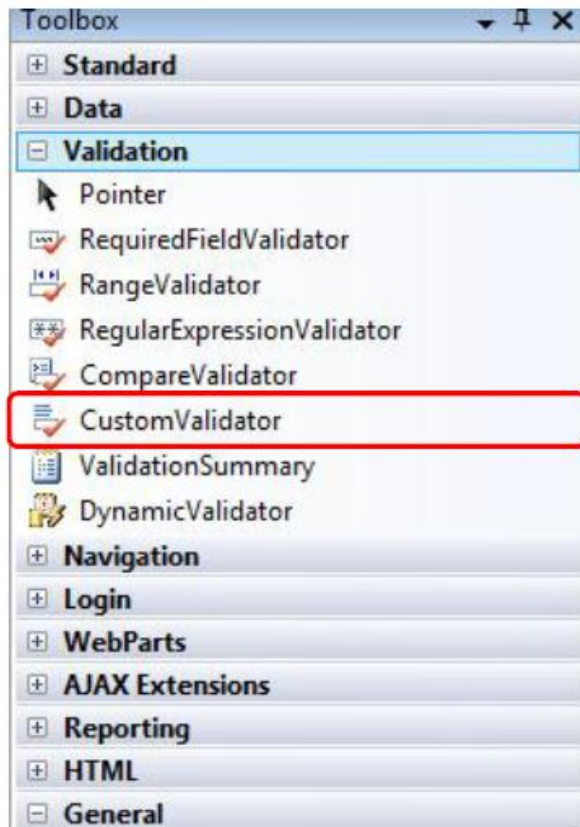
- ValidationExpression: qui định mẫu kiểm tra dữ liệu dựa vào các ***ký hiệu qui định***.

COMPAREVALIDATOR



- So sánh giá trị của một điều khiển với giá trị của một điều khiển khác.
- So sánh giá trị của một điều khiển với một giá trị được xác định trước.
- Trong trường hợp không nhập dữ liệu, điều khiển sẽ không thực hiện kiểm tra
- Thuộc tính:
 - ControlToCompare (ưu tiên)
 - Operator
 - Type
 - ValueToCompare

CUSTOMVALIDATOR



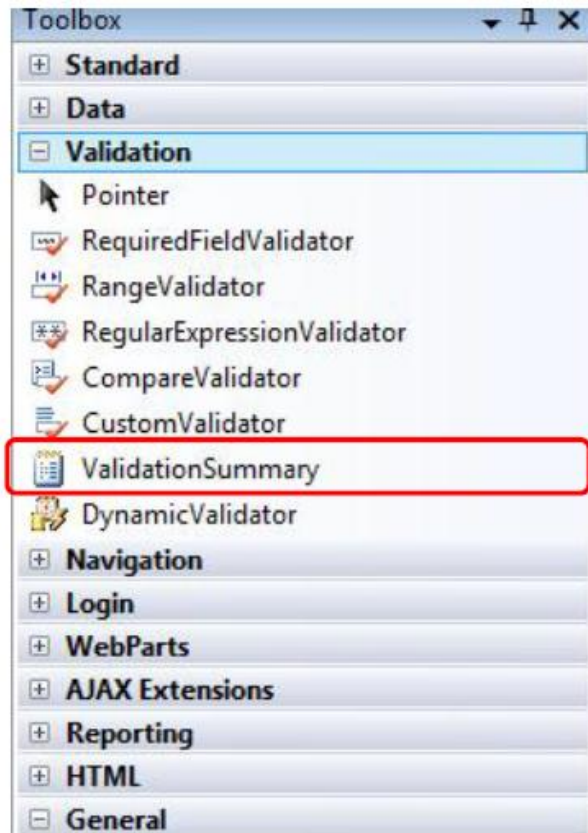
Cho phép người dùng tự viết hàm xử lý kiểm tra lỗi.

Thuộc tính:

- `ValidateEmptyText(True/False)`: có kiểm tra khi giá trị nhập là rỗng.
- `ClientValidationFunction`: tên hàm kiểm tra ở mức client.

Sự kiện `ServerValidate`: dùng để xử lý kiểm tra dữ liệu ở mức Server.

VALIDATIONSUMMARY

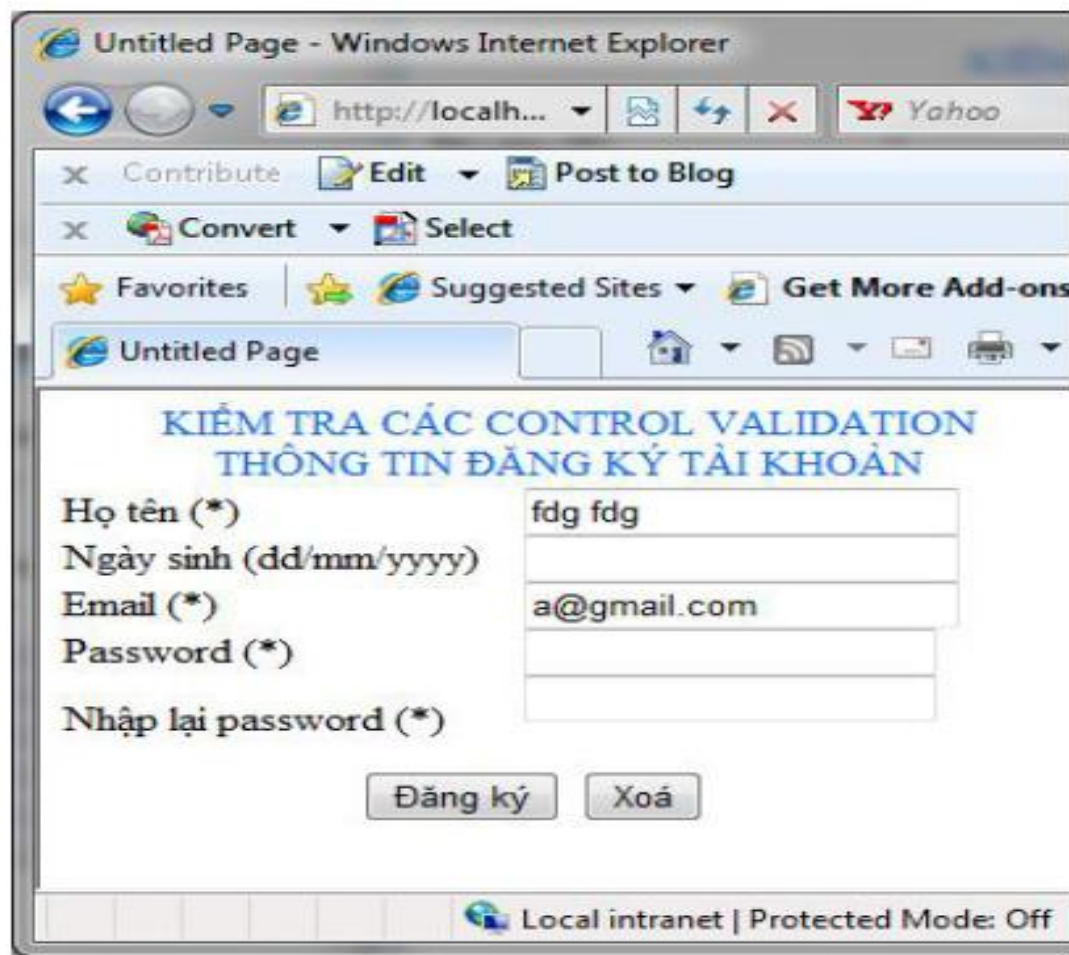


Dùng để hiển thị bảng lỗi - tất cả các lỗi hiện có trên trang Web (thuộc tính **ErrorMessage**).

Thuộc tính:

- HeaderText: chuỗi văn bản của dòng tiêu đề.
- ShowMessageBox (True/False): có thể hiện hộp thoại thông báo lỗi trên Browser.

VÍ DỤ



The screenshot shows a Windows Internet Explorer browser window displaying a registration form. The browser's address bar shows a local intranet URL. The form is titled "KIỂM TRA CÁC CONTROL VALIDATION THÔNG TIN ĐĂNG KÝ TÀI KHOẢN" and contains several input fields with validation icons (a red 'x') next to them. The fields are: "Họ tên (*)" with the value "fdg fdg", "Ngày sinh (dd/mm/yyyy)", "Email (*)" with the value "a@gmail.com", "Password (*)", and "Nhập lại password (*)". At the bottom of the form are two buttons: "Đăng ký" and "Xoá".

Bước 1: tạo project như hình bên

- ID userName: txtName
- ID Birthday: txtBirthday
- ID Email: txtEmail
- ID Password: txtPassword
- ID Confirm PWD: txtConfirmPWD
- ID Đăng ký: bRegister
- ID Xoá: bReset

Bước 2: thêm các control Validation vào trang ứng

ĐIỀU KHIỂN KIỂM TRA DỮ LIỆU (VALIDATION)

Standard Control	Tên control Validation	Thuộc tính	
		Name	Giá trị
txtName	RequiredFieldValidator	ID ControlToValidate ErrorMessage	rfvName txtName Nhập họ tên đầy đủ
txtBirthday	CompareValidator	ID ControlToValidate ErrorMessage Operator Type	cvBirthday txtBirthday Nhập đúng định dạng ngày tháng DataTypeCheck Date
txtEmail	RegularExpressionValidator	ID ControlToValidate ErrorMessage ValidationExpression	revEmail txtEmail Nhập đúng địa chỉ dạng Email Internet email address

CÁC ĐỐI TƯỢNG QUẢN LÝ ỨNG DỤNG WEB

- Đối tượng Request và Response
- Đối tượng Server
- Đối tượng Cookies
- Đối tượng Application và Session

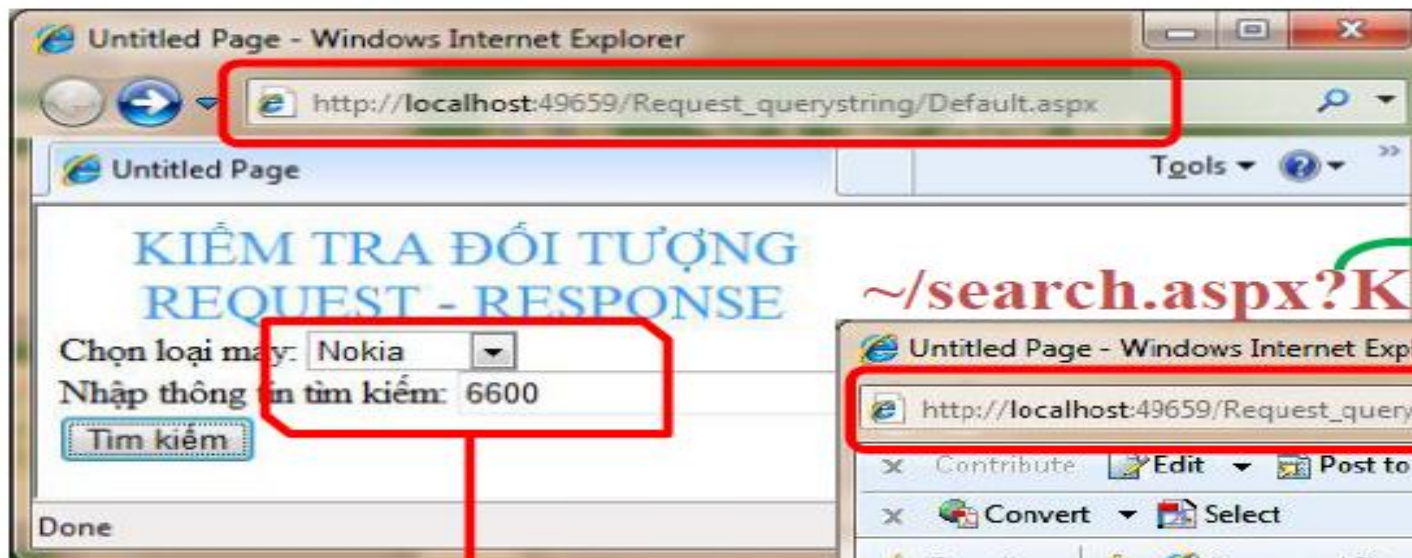
ĐỐI TƯỢNG REQUEST VÀ RESPONSE

Mô tả	REQUEST	RESPONSE
Ý nghĩa	Dùng để nhận giá trị từ Client gửi về cho Web server thông qua địa chỉ URL.	Được sử dụng để giao tiếp và điều phối thông tin giữa Web Server và Client (gửi kết quả đến Client).
Thuộc tính trong đối tượng	<code>Request.QueryString["<Tên_tham_số>"];</code> Ít sử dụng Ví dụ: địa chỉ URL <code>http: // <host> [: <port>] [<path> [? <QueryString>]]</code>	Ít sử dụng
Phương thức	Ít sử dụng	<code>Response.Write("chuỗi") ;</code> <code>Response.Redirect("URL") ;</code>

In 1 chuỗi ra màn hình phía Client

Yêu cầu Client truy cập đến URL khác

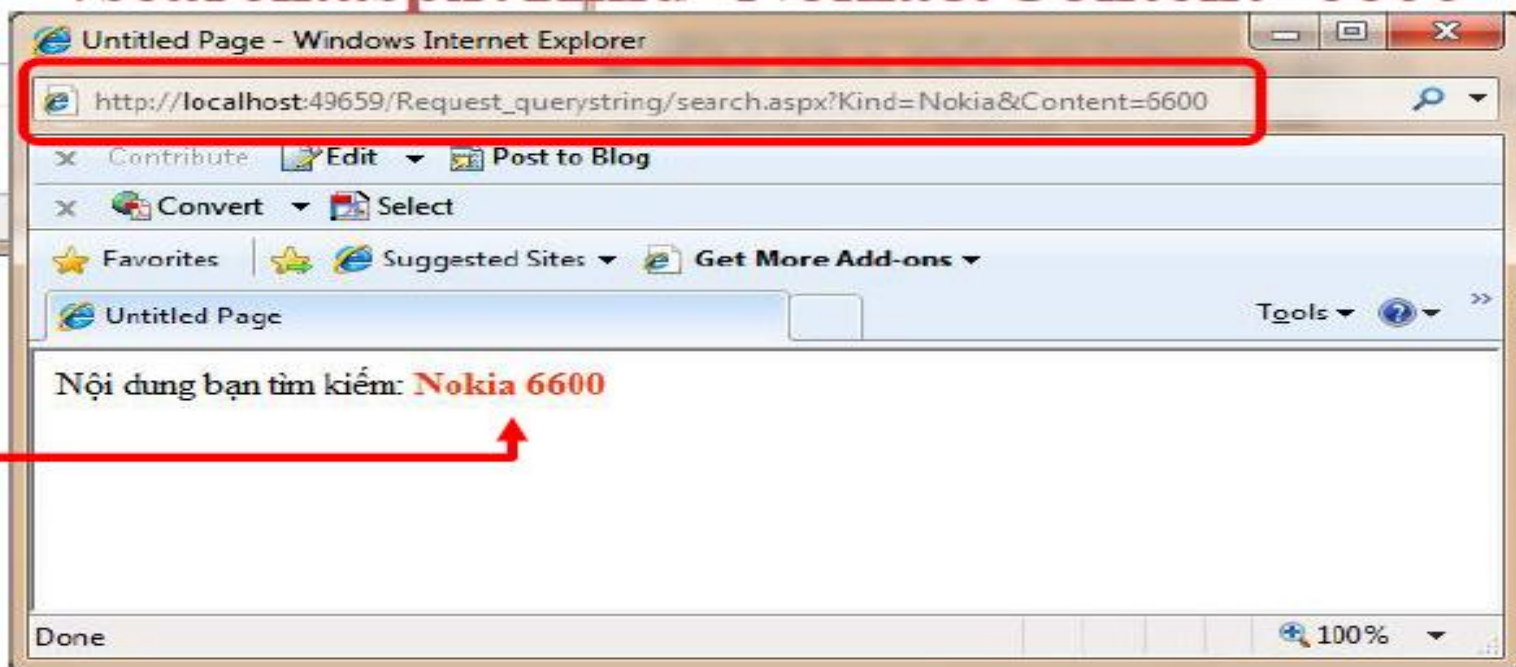
VÍ DỤ



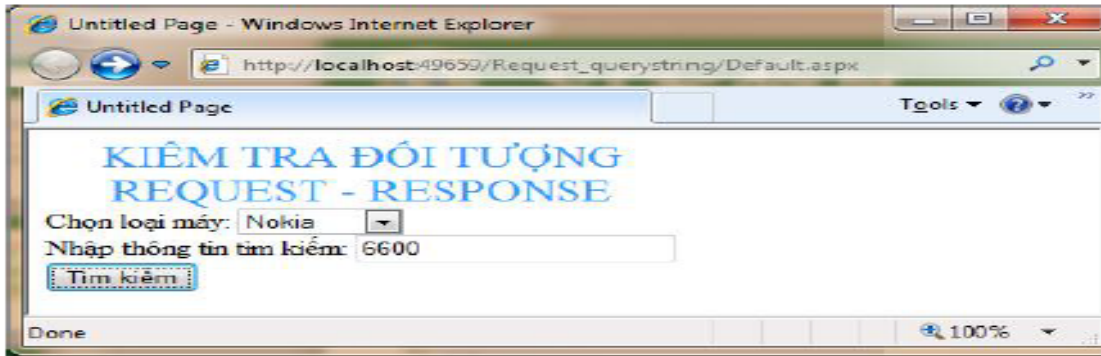
Thông tin được
truyền từ trang
default.aspx
→ search.spax

Chuỗi Query String

`~/search.aspx?Kind=Nokia&Content=6600`



ĐỒI TƯỢNG NÂNG CAO



❖ Các bước xây dựng trang default.aspx

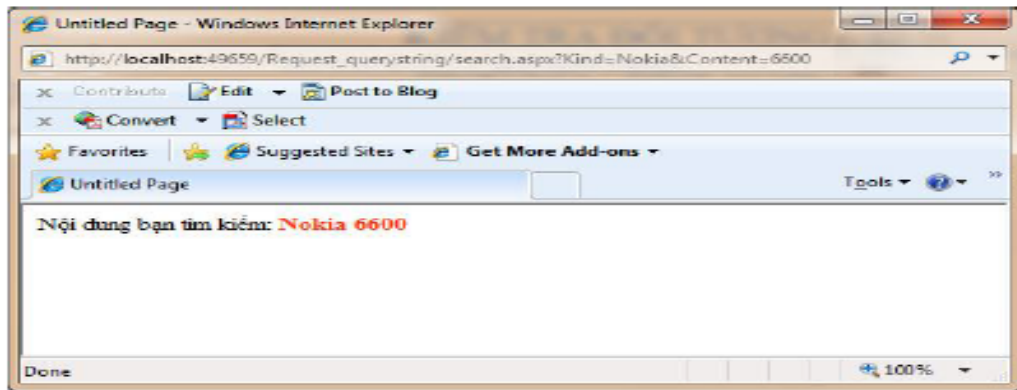
Bước 1: tạo project như hình bên

- ID Dropdownlist: cbKind
- ID Textbox: txtSearch
- ID Button: bSearch

Bước 2: coding cho trang default.aspx.cs:
double click vào nút bSearch

```
search.aspx.cs Default.aspx.cs Default.aspx search.aspx Start Page
_Default
using System;
using System.Configuration;
using System.Data;
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void bSearch_Click(object sender, EventArgs e)
    {
        Response.Redirect("search.aspx?Kind="+cbKind.Text+"&Content="+txtSearch.Text);
    }
}
```

ĐỒI TƯỢNG NÂNG CAO



❖ Các bước xây dựng trang search.aspx

Bước 1: tạo project như hình bên
- ID Label: lSearch

Bước 2: coding cho trang search.aspx.cs: xử lý ở sự kiện Page_Load

```
search.aspx.cs* Default.aspx.cs Default.aspx search.aspx Start Page
_Default
Page_Load(object sender, EventArgs e)
using ...
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        lSearch.Text = Request.QueryString["Kind"]+" "+Request.QueryString["Content"];
    }
}
```

ĐỐI TƯỢNG SERVER

Chức năng: được sử dụng để cung cấp thông tin của Web Server cho ứng dụng.

Phương thức thường dùng:

Server.Transfer("URL"): ngừng thi hành trang hiện hành, gửi yêu cầu mới đến trang khác.

Server.MapPath([đối số]): trả về đường dẫn vật lý tương ứng với đường dẫn ảo trên Web Server.

```
Server.Transfer("~/Thongtin/Loithuonggap.aspx")
```

```
Server.MapPath(""): trả về đường dẫn đến thư mục của trang hiện hành.
```

Lưu ý: sử dụng ký hiệu “~” để chỉ đến đường dẫn tương đối của một đối tượng và dấu “..” để chỉ đến đường dẫn tuyệt đối

ĐỐI TƯỢNG COOKIES

- Chức năng: dùng để lưu trữ thông tin của người dùng tại máy Client.
- Sử dụng đối tượng:

➤ Tạo đối tượng Cookies:

```
HttpCookie cookName = new HttpCookie("cookName ");  
cookName.Value = "Kiem-tra-Cookie";  
cookName.Expires = DateTime.Today.AddDays(3) // hết hạn
```

➤ Lưu Cookies vào máy Client:

```
Response.Cookies.Add(<HttpCookie Object>);
```

➤ Lấy giá trị của Cookies:

```
Request.Cookies["Tên Cookie"].Value;
```

ĐỐI TƯỢNG SESSION

Chức năng: lưu trữ thông tin trong một phiên làm việc cụ thể khi người dùng kết nối đến Web server lần đầu tiên (lưu trữ thông tin trao đổi giữa các trang aspx).

Phương thức:

Session.Timeout(phút) thời gian duy trì Session, nếu người dùng không tương tác với Web Server (default=20').

Session.Abandon([đổi số]): hủy Session và giải phóng tài nguyên cho Web Server.

Sử dụng đối tượng:

- Tạo biến Session: `Session["<tên biến>"] = <giá trị>;`
- Lấy giá trị của biến Session: `<Biến>=Session["<tên biến>"];`

ĐỐI TƯỢNG APPLICATION

Chức năng:

Đối tượng/biến toàn cục, quản lý toàn bộ ứng dụng Web (chỉ bị **đóng/hủy** khi **tắt** Web Server).

Thông tin được lưu trữ trong đối tượng Application được “**hiểu**” ở tất cả các trang aspx trong suốt thời gian “**sống**” của ứng dụng.

Sử dụng đối tượng:

- Tạo biến Application: `Application.Lock();`
`Application[“<tên biến>”] = <giá trị>;`
`Application.Unlock();`
- Lấy giá trị của biến Application: `<Biến>=Application[“<tên biến>”];`