UBND TỈNH LÂM ĐỒNG
TRƯỜNG CAO ĐẲNG ĐÀ LẠT

GIÁO TRÌNH

MÔ ĐUN: XÂY DỰNG WEBSITE BẰNG CÔNG CỤ

NGÀNH/NGHỀ: CÔNG NGHỆ THÔNG TIN (ƯDPM)

TRÌNH ĐỘ: TRUNG CẤP

(Ban hành kèm theo Quyết định số: /QĐ-CĐNĐL ngày …tháng…năm… của Hiệu trưởng Trường Cao đẳng Đà Lạt)

LƯU HÀNH NỘI BỘ

Lâm Đồng, năm 2017

TUYÊN BỐ BẢN QUYỀN

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

LỜI GIỚI THIỆU

Trong thời đại Internet hiện nay và tương lai, Website là một phần rất quan trọng đối với các doanh nghiệp trong việc quảng bá và giao dịch trong các hoạt động sản xuất, kinh doanh,…

Để thiết kế và phát triển một website đáp ứng được các mục đích trên thường người xây dựng phải lập trình (coding) bằng ngôn ngữ PHP, ASP hoặc ASP.NET. Tuy nhiên việc lập trình (coding) thường gây khó khăn cho những người không biết hoặc không có kinh nghiệm về lập trình trong việc xây dựng một website động cho doanh nghiệp, cũng như sẽ khó khăn với cả những người.

Từ thực tiễn đó, giáo trình này nhằm hỗ trợ học sinh, sinh viên những kỹ thuật thiết kế & phát triển website động mà không cần phải có nhiều kiến thức về lập trình vẫn dễ dàng xây dựng & phát triển các website động nhanh chóng, có tính mỹ thuật cao và nội dung sinh động, phong phú như: Website Âm nhạc, Điện ảnh, Tin tức, diễn đàn, Blog, Gallery, Bán hàng - Thương mại điện tử. v.v . . .

Đà Lạt, ngày 07 tháng 07 năm 2017

Tham gia biên soạn

Chủ biên: Ths. Phạm Đình Nam

MỤC LỤC

TRANG

LỜI GIỚI THIỆU 2
BÀI 1: MÔI TRƯỜNG LẬP TRÌNH ASP.NET 7
1. Sử dụng MS Visual Studio .NET 7
2. Tạo và quản lý ứng dụng web 10
2.1 Tập tin Global.asax 10
2.2 Tập tin Web.config 10
3. Các đối tượng điều khiển của ASP.NET 11
3.1 Điều khiển hiển thị thông tin (Label) 11
3.2. Điều khiển Literal 12
3.3. Điều khiển cho phép người dùng nhập liệu (TextBox) 12
3.4. Sử dụng điều khiển CheckBox 13
3.5. Điều khiển RadioButton 14
3.6. Điều khiển Button 15
3.7. Điều khiển LinkButton 15
3.8. Điều khiển ImageButton 15
3.9. Điều khiển hiển thị ảnh 18
3.9.1. Điều khiển Image 19
3.9.2. Điều khiển ImageMap 19
3.10. Điều khiển Panel 20
3.11. Điều khiển HyperLink 22
BÀI 2: CƠ BẢN VỀ ASP.NET 23
1. Kiến trúc .NET 23
2. Mối quan hệ giữa ASP.NET và kiến trúc .NET 24
3. Các lớp .NET Framework 25
4. Tạo ứng dụng ASP.NET 25
5. Cấu trúc tổng quát của một chương trình ASP.NET 27
6. Biến và Hằng 29
6.1. Biến 29
6.2. Hằng 30
7. Dữ liệu kiểu trị và kiểu qui chiếu 30
7.1. Kiểu giá trị được định nghĩa trước (Predefined Value Types) 31
7.1.1. Các kiểu Integer 31
7.1.2. Kiểu dữ liệu số dấu chấm di động (Floating Point Types) 32
7.1.3. Kiểu dữ liệu số thập phân (Decimal Type) 32

7.1.4. Kiểu Boolean 32
7.1.5. Kiểu Character Type 32
7.2. Kiểu tham khảo tiền định nghĩa 33
8. Các câu lệnh điều khiển 34
8.1. Câu lệnh lựa chọn if 34
8.2. Câu lệnh lựa chọn Case 35
8.3. Cấu trúc lặp for 40
8.4. Cấu trúc lặp while 41
8.5. Cấu trúc lặp do 42
8.6. Câu lệnh try…catch 43
9. Mảng 45
9.1. Giới thiệu về mảng trong C# 45
9.2. Các kiểu mảng trong C# 46
10. Các phép toán 47
11. An toàn kiểu 56
12. Kiểu liệt kê 57
13. Các không gian tên 58
14. Sử dụng các chú thích 59
15. Từ định danh và từ khoá 59
BÀI 3: NHỮNG TÍNH NĂNG TIÊN TIẾN TRONG ASP.NET 62
1. Lỗi và xử lí biệt lệ 62
2. Các sự kiện 63
3. Thuộc tính 65
4. Quản lý bộ nhớ bên dưới 68
4.1. Giá trị các kiểu dữ liệu 68
4.2. Các kiểu dữ liệu tham chiếu 71
BÀI 4: TRUY CẬP VÀ XỬ LÝ CƠ SỞ DỮ LIỆU VỚI .NET 74
1. Tổng quan về ADO.NET 74
2. Các công cụ kết nối dữ liệu 75
2.1. Đối tượng Connection 75
3. Hiển thị dữ liệu 80
4. Thao tác với dữ liệu 89
BÀI 5: LÀM VIỆC VỚI TẬP TIN 95
1. Quản lý hệ thống tập tin 95
2. Di chuyển, Sao chép, Huỷ File 98
3. Đọc và viết vào File 99
4. Quản lý tập tin upload 100

BÀI 6: LẬP BÁO CÁO VỚI CRYSTAL REPORT 103
1. Giới thiệu công cụ Crystal Report 103
1.1. Tải và cài đặt Crystal Report 103
1.2 Tạo một Crystal Report 106
2. Gắn kết dữ liệu vào báo cáo 110
3. Xây dựng và triển khai báo cáo 110
BÀI 7: MỘT SỐ CÔNG CỤ XÂY DỰNG WEBSITE 112
1. Quy trình tổng quát để xây dựng website 112
2. Các công cụ xây dựng website 114
3. Chọn công cụ và phiên bản 116
4. Cài đặt công cụ lên máy chủ web cục bộ 116
5. Tùy chỉnh các thông số cơ bản 121
BÀI 8: CÁC MÔ-ĐUN BỔ SUNG CHỨC NĂNG CHO WEBSITE 124
1. Danh mục chức năng của website 124
2. Mô-đun bổ sung chức năng 128
3. Tích hợp các mô-đun bổ sung chức năng vào website 130
BÀI 9: QUẢN LÝ NỘI DUNG CỦA WEBSITE 146
1. Làm việc với hệ quản trị của website 146
1.1. Thay đổi logo của website bạn 146
1.2. Quản lý người sử dụng website của bạn 148
1.3. Tải tập tin lên website của bạn 152
2. Quản lý các mô-đun thông tin 154
2.1. Sử dụng module Text/HTML 154
2.2. Sử dụng module thông báo 155
3. Tạo giao diện mới cho website 157
3.1. Thay đổi skin website 157
3.2. Thay đổi container 161
BÀI 10: XUẤT BẢN WEBSITE 164
1. Chọn và đăng ký tên miền cho website 164
2. Chọn và đăng ký máy chủ 166
- Thuê host hay là mua host? 166
- Chọn host Việt Nam hay host nước ngoài ? 166
3. Chọn công cụ xuất bản 168
4. Xuất bản Website 168

GIÁO TRÌNH MÔ ĐUN

Tên mô đun: Xây dựng website bằng công cụ

Mã mô đun: MĐ19

Vị trí, tính chất, ý nghĩa và vai trò của mô đun:

- Vị trí: Mô đun này được học sau các mô đun chuyên môn nghề: Thiết kế và quản trị website, An toàn bảo mật thông tin, Đồ họa ứng dụng 1 và 2.

- Tính chất: Là mô đun chuyên môn bắt buộc, yêu cầu sinh viên phải có kiến thức, kỹ năng thiết kế đồ họa xử lý hình ảnh cũng như biên tập nội dung HTML.

- Ý nghĩa và vai trò của môn học/mô đun: Tài liệu này được thiết kế theo từng mô đun/ môn học thuộc hệ thống mô đun/môn học của một chương trình, để đào tạo hoàn chỉnh nghề Công nghệ thông tin (ứng dụng phần mềm) ở trình độ Trung cấp và Cao đẳng, đồng thời được dùng làm Giáo trình cho học viên trong các khoá đào tạo ngắn hạn hoặc cho các công nhân kỹ thuật, các nhà quản lý và người sử dụng nhân lực tham khảo.

Mục tiêu của mô đun:

Về kiến thức:

- Xác định được nhu cầu quảng bá, kinh doanh của công ty đối với website;

- Trình bày được tiến trình thiết kế và xuất bản một website;

- Trình bày được kiến trúc hạ tầng .NET.

Về kỹ năng:

- Vận dụng được cú pháp của ngôn ngữ lập trình ASP.NET.

- Sử dụng được phương pháp lập trình hướng đối tượng trong ngôn ngữ lập trình ASP.NET.

- Khai thác được các tính năng tiên tiến trong ASP.NET.

- Tạo được các ứng dụng web kết nối với cơ sở dữ liệu.

- Tạo được báo cáo.

- Sử dụng một số công cụ, dịch vụ để xây dựng website;

- Xác định được các nhu cầu, chuẩn bị đầy đủ các vật liệu, nội dung sẽ xuất bản;

- Thiết kế được các trang thân thiện và logic;

- Lựa chọn tên miền website và công cụ xuất bản phù hợp với quy mô bài toán;

Về năng lực tự chủ và trách nhiệm:

- Có khả năng tự nghiên cứu, tự học, tham khảo tài liệu liên quan đến môn học để vận dụng vào hoạt động hoc tập.

- Vận dụng được các kiến thức tự nghiên cứu, học tập và kiến thức, kỹ năng đã được học để hoàn thiện các kỹ năng liên quan đến môn học một cách khoa học, đúng quy định.

Nội dung của mô đun:

BÀI 1: MÔI TRƯỜNG LẬP TRÌNH ASP.NET

Mã Bài: MĐ19_01

Giới thiệu:

Trong nhiều năm qua, ASP đã được cho rằng đó thực sự là một lựa chọn hàng đầu cho web developers trong việc xây dựng những web sites trên nền máy chủ web Windows bởi nó vừa linh hoạt mà lại đầy sức mạnh. Đầu năm 2002, Microsoft đã cho ra đời một công nghệ mới đó chính là ASP.NET. Đây thực sự là một bước nhảy vượt bậc của ASP cả về phương diện tinh tế lẫn hiệu quả cho các developers. Nó tiếp tục cung cấp khả năng linh động về mặt hỗ trợ ngôn ngữ, nhưng hơn hẳn về mặt lĩnh vực ngôn ngữ script vốn đã trở nên hoàn thiện và trở thành ngôn ngữ cơ bản của các developers. Việc phát triển trong ASP.NET không chỉ yêu cầu hiểu biết về HTML và thiết kế web mà còn khả năng nắm bắt những khái niệm của lập trình và phát triển hướng đối tượng.

Lần lượt, chúng ta sẽ tìm hiểu về ASP.NET, về cách cài đăt các phần mềm, tạo web server… cũng như cách thức thiết kế một trang ASP.NET.

Mục tiêu:

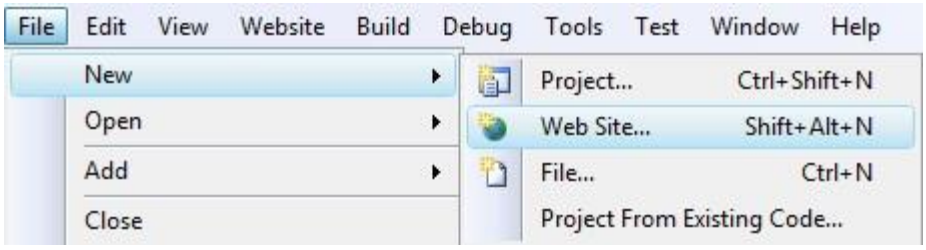
- Sử dụng được công cụ phát triển ứng dụng web MS Visual Studio.NET.
- Sử dụng được tất cả các công cụ trong ToolBox .
- Tổ chức và quản lý được ứng dụng web.

Nội dung chính:

1. Sử dụng MS Visual Studio .NET

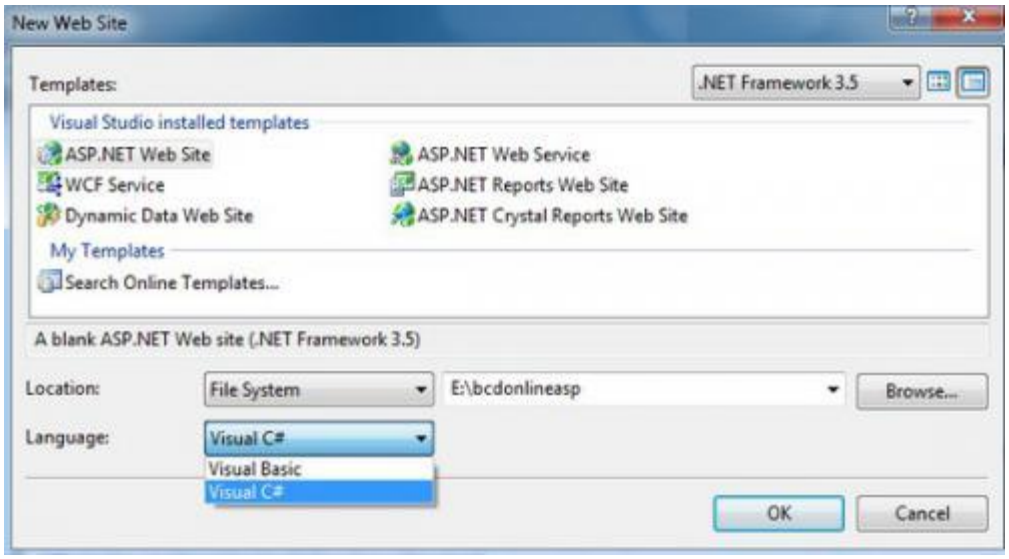
Visual Studio là một công cụ cực kỳ mạnh của hãm phần mềm nổi tiếng Microsoft hiện nay có nhiều phiên bản sử dụng: vs2005, vs2008, vs2010…., với công cụ này bạn có thể phát triển được các ứng dụng phần mềm, web với nhiều ngôn ngữ lập trình khác nhau đặt biệt với các ứng dụng winform. webform C# hay visual basic được hỗ trợ rất tốt.

Bây giờ chúng ta sẽ tạo dự án webform băng Visual Studio 2008. Bạn khởi động Visual Studio lên click vào menu file new một website mới.



Tạo mới website asp.net bằng visual studio

Lúc này một cửa sổ mới hiện ra bạn chọn vào ASP.Net Web Site phía dưới chọn vào ngôn ngữ C# và nơi chứa ứng dụng website của bạn.



New Website Asp.net bằng Visual Studio

Vậy là chúng ta tạo được một website ASP.net bằng Visual Studio, bây giờ chúng ta sẽ tìm hiểu các công cụ và làm quên với giao diện của phần mềm này.

Những thanh công cụ mà ta hay dùng là: Toolbox, Solution Explorer, Properties, công cụ Error list.

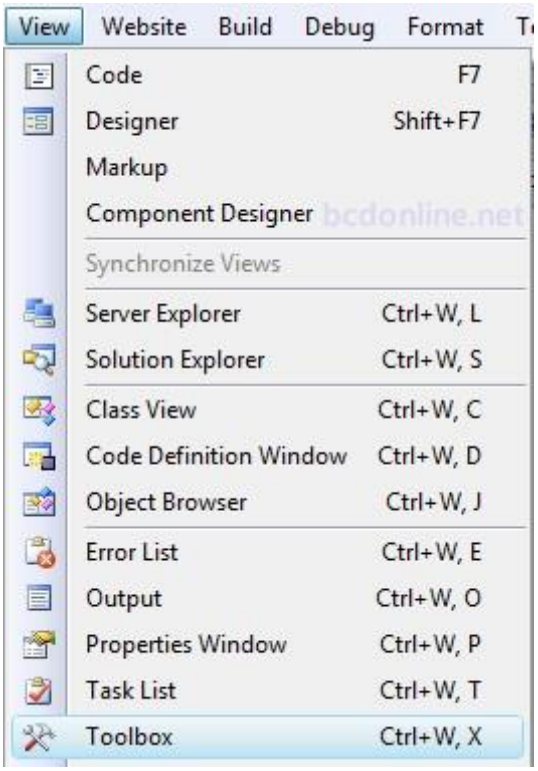
Toolbox: Chứa những công cụ cho ta sử dụng trong quá trình làm việc ví dụ như: kéo textbox, button….

Solution Explorer: Phần quản lý các file khi ta sử dụng.

Properties: Thanh thuộc tính của một đối tượng, có thể đổi tên đối tượng, gán các hành động cho nó…

Error list: Là phần hiển thị danh sách lỗi khi chúng ta build các code của mình.

Nếu mất các thành phần trên bạn có thể vào menu view của Visual Studio để mở nó lên và sử dụng.

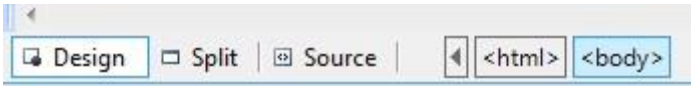


Ngoài ra còn rất nhiều các công cụ khác.

Để bắt lỗi câu lệnh thì bạn chỉ cần Ctrl + Shift + B là có thể thấy nó hiện lên Error list, còn các lỗi logic bạn sẽ nắm bắt được nó sẽ diễn biến thế nào để sửa cho hợp lý.

Ví dụ bạn tạo một textbox và button như hình dưới đây. Bạn kéo từ toolbox qua bên giao diện bên phải ở trang default.aspx.

Visual Studio .NET cho phép bạn thiết kế ở các chế độ: design tức là giao diện, source là code và split là vừa design vừa code.



Khi double click vào button vừa tạo ra, sẽ cho bạn đế file .cs ví dụ mình khai báo biến string là a gán giá trị của texbox.

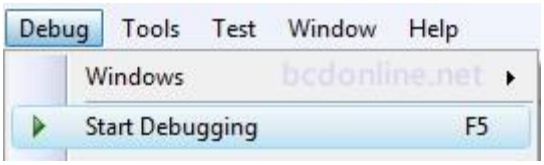
string a = txt1.Text;

Bạn click vào đầu dong theo hình dưới đây.

```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void btSubmit_Click(object sender, EventArgs e)
    {
        string a = txt1.Text;
    }
}
```

Để chặn debug khi đang debug thì sẽ chạy từ dòng mình đánh dấu, sau đó vào menu chọn debug / start debugging.



Tiếp đến bạn nhận giá trị nào đó trong text box và bấm nút button bạn sẽ được dẫn đến ngay dòng mình đánh dấu lúc nảy, bạn dùng phím f10, f11 để di chuyển và cho chuột đế giá trị của biến để xem giá trị hiện tại của nó.

```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void btSubmit_Click(object sender, EventArgs e)
    {
        string a = txt1.Text;
    }
}
```
a | "bcdonline.net"

Đó là cách mà chúng ta sẽ lần ra lối từ các giá trị mà nó có qua mỗi lúc, mỗi giai đoạn.

2. Tạo và quản lý ứng dụng web

2.1 Tập tin Global.asax

Một project chỉ có một tập tin Global.asax chứa các đoạn code xử lý trong các sự kiện cấp Session và Aplication. Tập tin này dùng để khai báo và khởi tạo cho biến Application, Session trong các sự kiện của hai đối tượng này.

2.2 Tập tin Web.config

- Trong một project được phép có nhiều tập tin Web.config
- Tập tin web.config có dạng cấu trúc XML
- Chứa các thông tin cấu hình của ứng dụng Web

Để bổ sung các thông tin tùy chọn riêng cho ứng dụng chúng ta khai báo khóa trong thẻ <appSettings> như sau:

```
<appSettings>
        <add key="tên khoá"  value="giá trị khóa" />
</appSettings>
```

Để Truy xuất thông tin:

```
System.Configuration.ConfigurationManager.AppSettings["Khóa"];
```

3. Các đối tượng điều khiển của ASP.NET

3.1 Điều khiển hiển thị thông tin (Label)

Các thuộc tính của label	
Tên thuộc tinh	Ảnh hưởng
BackColor	Cho phép thay đổi màu nền của Label
BorderColor	Cho phép thiết lập đường viền của Label
BorderStyle	Hiển thị đường viền của Label theo các dạng Notset, None, Dotted, Dashed, Solid, Double, Groove, Ridge, Insert và OutSet.
cssClass	Cho phép đưa vào một lớp css
Font	Thiết lập thuộc tính của Font
ForeColor	Thiết đặt màu chữ nội dung của Label
Style	Cho phép gán các thuộc tính cho Label
ToolTip	Hiển thị nội dung khi di chuột vào điều khiển Label
ID	Tên của điều khiển

Tuy điều khiển Label chứa đựng rất nhiều thuộc tính để kết xuất hiển thị nhưng với kinh nghiệm của mình trong quá trình thiết kế Web tôi khuyên bạn lên sử dụng thuộc tính cssClass để sử dụng một lớp css.

Gán thuộc tính trực tiếp:

```
<asp:Label ID="Label1" BackColor="ActiveBorder"
BorderColor="ActiveCaption" ForeColor="Blue" Font-Size="12pt"
runat="server" Text="Trung tâm công nghệ cao iTechPro" />
```

Sử dụng cssClass

```
<asp:Label ID="Label2" CssClass="label" runat="server"
Text="Trung tâm công nghệ cao iTechPro" />
```

Kết quả

Gán thuộc tính trực tiếp

Trung tâm công nghệ cao iTechPro

Sử dụng cssClass

Trung tâm công nghệ cao iTechPro

3.2. Điều khiển Literal

Giống với điều khiển Label bạn có thể dùng Literal để trình bày Text hoặc nội dung Html. Literal hỗ trợ một thuộc tính mà Label không hỗ trợ đó là: thuộc tính Mode gồm 3 giá trị là: Pass through, Endcode, transform.

```
<asp:Literal ID="lit1" runat="server" Text="<hr/>"
Mode="PassThrough" /> <asp:Literal ID="lit2" runat="server"
Text="<hr/>" Mode="Encode" /> <asp:Literal ID="lit3"
runat="server" Text="<hr/>" Mode="Transform" />
```

Hiển thị:

<hr/>

3.3. Điều khiển cho phép người dùng nhập liệu (TextBox)

Các thuộc tính của Textbox	
Thuốc tính	
Textmode	Kiểu hiển thị của Textbox gồm 3 giá trị: SingleLine- hiển thị 1 trường nhập liệu trên 1 dòng, MultiLine- hiển thị 1 trường nhập liệu nhiều dòng, Password- hiển thị 1 trườngnhập mà Text sẽ được thay thế bằng các ký tự đặc biệt
AccessKey	Cho pheps bạn chỉ định phím để di chuyển tới

	controlTextBox
AutoComplete Type	Cho phép bạn kết hợp với một lớp autoComplete với điều khiển TextBox.
AutoPostBack	Cho phép gửi dữ liệu lên server khi nội dung của TextBox thay đổi.
Columms	Cho phép chỉ định số cột được hiển thị trong TextBox
Enabled	Cho phép hay không cho phép nhập liệu trên TextBox
MaxLength	Cho phép quy đinh độ dài của dữ liệu mà một người sửdụng có thể nhập trên TextBox
ReadOnly	Cho phép chỉ đưa dữ liệu ra TextBox chứ không nhập dữliệu vào.
Rows	Cho phép chỉ định số dòng hiển thị trong TextBox
TabIndex	Cho phép chỉ định thứ tự Tab của TextBox
Wrap	Cho phép chỉ định có word-wraps không khi thiết lập thuộc tính TextMode là Multiline

Điều khiển TextBox hỗ trợ phương thức và sự kiện sau:

- Focus: cho phép thiết lập form khởi tạo ưu tiên tới TextBox

- TextChanged: Xảy ra trên Server khi nội dung TextBox thay đổi. để sự kiên này xảy ra bạn cần thiết đặt thuộc tính AutoPostback là true.

3.4. Sử dụng điều khiển CheckBox

Các thuộc tính của Checkbox	
AccessKey	Enables you to specify a key that navigates to the TextBox contro
AutoPostBack	nablesyou to post the form containing the CheckBox back to theserver automatically when the CheckBox is checked or unchecked
Checked	Cho phép bạn gán hoặc thiết đặt trạng thái chọn hay không chọn của CheckBox
Enabled	Cho phép hoặc không cho phép

TabIndex	Enables you to specify the tab order of the check box.
Text	Enablesyou to provide a label for the check box.
TextAlign	Enables you to align the label for the check box. Possible values areLeft and Right.

CheckBox hỗ trợ phương thức va sự kiện

- Focus: Enables you to set the initial form focus to the check box.
- CheckedChanged: Raisedontheserverwhenthecheckboxischeckedorunchecked.

3.5. Điều khiển RadioButton

Điều khiển RadioButton luôn được sử dụng trong một nhóm và trong nhóm đó chỉ một RadioButton được chọn

Các thuộc tính của điều khiển RadioButton	
Thuộc tính	
AccessKey	Enablesyou to specify a key that navigates to the RadioButton control
AutoPostBack	Enables you to post the form containing the RadioButton back tothe server automatically when the radio button is checked or unchecked
Checked	Enables you to get or set whether the RadioButton control is checked.
Enabled	Enables you to disable the RadioButton
GroupName	Enables you to group RadioButton controls
TabIndex	Enables you to specify the tab order of the RadioButton control.
Text	Enables you to label the RadioButton control.
TextAlign	Enablesyou to align the RadioButton label. Possible values are Leftand Right.

RadioButton hỗ trợ các phương thức và sự kiện

- Focus: Enables youto set the initial form focus to the RadionButton control.

- CheckedChanged: Raised on the server when the unchecked.

3.6. Điều khiển Button

Các thuộc tính của điều khiển Button	
AccessKey	Cho phép chỉ định phím di chuyển tới điều khiển Button.
CommandArgument	Cho phép bạn chỉ rõ đối số được truyền tới lệnh thực hiện.
CommandName	Chỉ định tên một lệnh được truyền tới trong Command Event.
Enabled	Cho phép vô hiệu hoá điều khiển Button
OnClientClick	Cho phép chỉ định đến một hàm phía client khi nhấn vào Button.
PostBackUrl	Cho phép trả dữ liệu lên một trang khác.
TabIndex	Chỉ định thứ tự tab của Button.
Text	Nội dung text hiển thị trên điều khiển Button.
UseSubmitBehavior	Cho phép sử dụng javascript để trả dữ liệu lên một form.

Các phương thức và sự kiện

- Focus: Cho phép thiết lập khi khơi tạo Form ưu tiên điều khiển TextBox.
- Click: Xảy ra khi điều khiển Button được nhấn.
- Command: Xảy ra khi điều khiển Button được nhấn. CommandName và CommandArgument được truyền qua sự kiện.

3.7. Điều khiển LinkButton

Các phương thức và thuộc tính giống với điều khiển Button nhưng cách hiển thị của nó dưới dạng Text giống như thẻ <a> của HTML và có thể áp dụng thuộc tính css của thể <a> cho đối tượng LinkButton.

3.8. Điều khiển ImageButton

Các thuộc tính của điều khiển ImageButton	
Thuộc tính	
ImageU	Chỉ đến đường dẫn của ảnh

rl	
ImageAlign	Cho phép căn chỉnh ảnh trong ImageButton các giá trị của nó có thể là: AbsBottom, AbsMiddle, Baseline, Bottom,Left, Middle, NotSet, Right, TextTop, and Top.

Sử dụng Client Scripts với điều khiển Button

Cả ba điều khiển Button trên đều có thuộc tính OnClientClick, bạn có thể sử dụng thuộc tính này để thực hiện mã phía Client mà bạn cần khi điều khiển Button được nhấn.

```
<%@ Page Language="C#"%> <!DOCTYPE html PUBLIC "-//W3C//DTD
XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<script runat="server"> protected void btnDelete_Click(object
sender, EventArgs e) { lblResult.Text = "All pages deleted!"; }
</script> <html> <head runat="server"> <title>iTechPro</title>
</head> <body> <form id="form1" runat="server"> <div>
<asp:Button ID="Button1" OnClick="Button1_Click"
OnClientClick="return confirm('Bạn có chắc chắn?');"
runat="server" Text="Button" /><br /> <asp:Label ID="Label1"
runat="server" Text="iTechPro" /> </div> </form> </body> </html>
```

Giải thích ví dụ: trên đoạn mã trên trong form gồm 2 điều khiển Button và Label. Trong điều khiển Button ta gán vào 2 sự kiện: OnClientClick phía Clien, và OnClick phía Server

OnClientClick sẽ thực hiện khi người dùng phía Client nhấn vào Button, trong mã Client trình bày một hàm confirm trong javascript với mục đính hỏi người dùng xác nhận việc thực hiện nào đó “Bạn có chắc chắn?” nếu người dùng chọn OK thì Sự kiện thứ 2 OnClick sẽ được thực hiện còn nếu chọn Cancel thì sự kiện phía Server không được thực hiện.

Mặc định khi bạn nhấn vào Button nó sẽ thực hiện công việc ngay trên trang và trang của chúng ta sẽ Load lại một lần nhưng bạn có thể sử dụng thuộc tính PostBackUrl để chuyển sang một trang khác.

Trang ButtonSearch.aspx

```
<%@ Page Language="C#" %> <!DOCTYPE html PUBLIC "-//W3C//DTD
XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html> <head runat="server"> <title>iTechPro</title> </head>
<body> <form id="form1" runat="server"> <div> <asp:Label
ID="lblSearch" runat="server" Text="Search:" /> <asp:TextBox
ID="txtSearch" runat="server" /> <asp:Button ID="btnSearch"
Text="GO!" runat="server" PostBackUrl="ButtonSearchResult.aspx"
/> </div> </form> </body> </html>
```

Trong thuộc tính PostBackUrl của điều khiển btnSearch sẽ chuyển sang trang ButtonSearchResult

```
<script runat="server"> void Page_Load(object sender, EventArgs
e) { if (PreviousPage != null) { TextBox txtSearch =
(TextBox)PreviousPage.FindControl("txtSearch"); lblSearch.Text =
txtSearch.Text; } } </script> <html
xmlns="http://www.w3.org/1999/xhtml"> <head runat="server">
<title>iTechPro</title> </head> <body> <form id="form1"
runat="server"> <div> <asp:Label ID="lblSearch" runat="server"
/> </div> </form> </body> </html> <%@ Page Language="C#" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Trong đoạn mã trên ta để ý phương thức FindControl của PreviousPage nó sẽ tìm đến Điều khiển trên trang ButtonSearch.aspx mà ta cung cấp ID của nó.

Chỉ định một Button mặc định

Ví dụ trên trang của bạn có nhiều điều khiển Button, khi nhập dữ liệu bạn lại quen nhập xong dữ liệu và nhấn nút Enter trên bàn phím, bạn có thể để mặc định nút cập nhật dữ liệu làm mặc định khi nhấn phím Emter. Như ví dụ dưới đây.

```
<%@ Page Language="C#" %> <script runat="server"> void
btnXacnhan_Click(object sender,EventArgs e) { lblThongbao.Text =
txtHoten.Text; } </script> <!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>iTechPro</title> </head> <body> <form id="form1"
runat="server" defaultbutton="btnXacnhan"> <div> <asp:Label
ID="lblHoten" runat="server" Text="Nhập họ tên" /> <asp:TextBox
ID="txtHoten" runat="server" /> <hr /> <asp:Button
ID="btnXacnhan" OnClick="btnXacnhan_Click" runat="server"
```

```
Text="Xác nhận" /> <asp:Button ID="btnBoqua" runat="server"
Text="Bỏ qua" /> <hr /> <asp:Label ID="lblThongbao"
runat="server" Text="" /> </div> </form> </body> </html>
```

Điều khiển Command Event

Sự khác nhau giữa Command Event và Command Click là trong Command Event bạn có thể cung cấp Command Name và Command argument

Trong các ví dụ trước tôi đã lấy ví dụ về sự kiện Command Click nên bây giờ tôi sẽ lấy một ví dụ về Command Event để bạn so sánh.

Ví dụ trên trang bạn có 3 điều khiển Button như ví dụ dưới đây:

```
<%@ Page Language="C#"%> <script runat="server"> void
hcubiuChon(object sender, CommandEventArgs e) { if
(e.CommandName == "language") { switch
(e.CommandArgument.ToString()) { case "C#": lblComandEvent.Text
= "CShap"; break; case "VBNET": lblComandEvent.Text = "VB.NET";
break; case "JAVA": lblComandEvent.Text = "Java"; break; } }
</script> <!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>iTechPro</title> </head> <body>
<form id="form1" runat="server"> <div> Bạn chọn ngôn ngữ:
<asp:Label ID="lblComandEvent" runat="server"
Text=""></asp:Label> <br /> <asp:Button ID="btna"
OnCommand="hcubiuChon" CommandName="language"
CommandArgument="C#" runat="server" Text="C#" /> <asp:Button
ID="btnb" OnCommand="hcubiuChon" CommandName="language"
CommandArgument="VBNET" runat="server" Text="VB.NET" />
<asp:Button ID="btnc" OnCommand="hcubiuChon"
CommandName="language" CommandArgument="JAVA" runat="server"
Text="Java" /> </div> </form> </body> </html>
```

Cả ba điều khiển Button trên đều chứa Tên lệnh và đối số lệnh và tuỳ thuộc vào đối số lệnh khác nhau để ta có thể thực hiện một công việc tương ứng.

3.9. Điều khiển hiển thị ảnh

ASPNET bao gồm 2 điều khiển hiển thị ảnh. điều khiển Image và ImageMap.

3.9.1. Điều khiển Image

Điều khiển này dùng để hiển thị ảnh giống với thẻ <Img> trong HTML Các thuộc tính quan tâm:

Các thuộc tính của điều khiển Image	
Thuộc tính	
AlternateText	Nội dung thay thế khi lỗi đường dẫn của ảnh
DescriptionUrl	Cho phép bạn cung cấp một đường dẫn đến trang miêu tả chi tiết nội dung của ảnh
ImageAlign	Cho phép căn chỉnh ảnh lên quan tới các thành phần HTML khác trong trang và nó có thể là các giá trị sau:AbsBottom, AbsMiddle, Baseline, Bottom, Left, Middle, NotSet, Right, TextTop, and Top.
ImageUrl	Đường dẫn của ảnh trên điều khiển

```
<%@ Page Language="C#" %> <script runat="server"> void
Page_Load(object sender, EventArgs e) { Random rnd = new
Random(); switch (rnd.Next(3)) { case 0: Image1.ImageUrl =
"Images/images1.jpg"; Image1.AlternateText = "Picture 1"; break;
case 1: Image1.ImageUrl = "Images/images2.jpg";
Image1.AlternateText = "Picture 2"; break; case 2:
Image1.ImageUrl = "Images/images3.jpg"; Image1.AlternateText =
"Picture 3"; break; } } </script> <!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>iTechPro</title> </head> <body> <form id="form1"
runat="server"> <div> <asp:Image ID="Image1" runat="server" />
</div> </form> </body> </html>
```

3.9.2. Điều khiển ImageMap

Cho phép bạn tạo bản đồ ảnh trên 1 ảnh hiển thị trên trang web mà người dùng phía Client có thể chọn vào một vùng của ảnh để thực hiện một công việc nào đó.

```
<%@ Page Language="C#"%> <script runat="server"> void
ImageMap1_Click(object sender, ImageMapEventArgs e) { switch
```

```
(e.PostBackValue) { case "top": lblResult.Text = "Day la phan
dau"; break; case "middle": lblResult.Text = "day la phan giua";
break; case "under": lblResult.Text = "day la phan cuoi"; break;
} } </script> <!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>iTechPro</title> </head> <body>
<form id="form1" runat="server"> <div> <asp:ImageMap
ID="ImageMap1" HotSpotMode="PostBack" OnClick="ImageMap1_Click"
ImageUrl="~/Images/jiwoo18034.jpg" runat="server">
<asp:RectangleHotSpo t PostBackValue="top" Left="0" Top="0"
Right="300" Bottom="150" AlternateText="Top" />
<asp:RectangleHotSpot PostBackValue="middle" Left="0" Top="150"
Right="300" Bottom="300" AlternateText="middle" />
<asp:RectangleHotSpot PostBackValue="under" Left="0" Top="300"
Right="300" Bottom="450" AlternateText="Under" />
</asp:ImageMap> <asp:Label ID="lblResult" runat="server" />
</div> </form> </body> </html>
```

Các thuộc tính của ImageMap giống với Image và thêm vào hai thuộc tính đó là:

- HotSpots: cho phép bạn điền tập hợp thông tin của HotSpots được chứa đựng trong điều khiển ImageMap.

- HotSpotMode:Cho phép bạn chỉ định Enables you to specifythe behavior of the image map when you click a region. Possible values are Inactive, Navigate, NotSet, and PostBack.

3.10. Điều khiển Panel

Khi bạn cần đưa các điều khiển trang vào một nhóm để giải thích nghĩa cho nhóm đó hoặc có thể là để ẩn hoặc hiện nhóm điều khi nhấn 1 sự kiện nào đó trên trang của bạn, bạn có thể dùng điều khiển panel.

Một số thuộc tính của điều khiển panel mà bạn cần lưu ý là:

- DefaultButton: Cho phép bạn định nghĩa một button mặc định trong panel mà button mặc định này sẽ được thực hiện khi bạn nhấn phím Enter

- Direction: Cho phép bạn gán hoặc thiết đặt hướng hiển thị nội dung được đưa ra trong panel, có thể là các giá trị:NotSet, LeftToRight, and RightToLeft.

- GroupingText: Cho phép bạn trình bày Panel như 1 Fieldset với một chú giải riêng biệt

- HorizontalAlign: Cho phép bạn chỉ ra hướng ngang thể hiện nội dung của panel và nó có thẻ là các giá trị: Center, Justify, Left, NotSet, and Right.

- ScrollBars: Cho phép bạn hiển thị scrollbars khi bạn cố định chiều cao hoặc chiều rộng của panel và nội dung trong panel vượt quá độ rộng hoặc độ cao đó, nó có thể là các giá trị: Auto, Both, Horizontal, None, and Vertical.

```
<%@ Page Language="C#" %> <script runat="server"> void
Page_Load(object sender, EventArgs e) { for (int i = 1; i < 100;
i++) { buletnghenghiep.Items.Add("Nghề "+i.ToString()); } } void
hcubiuSothich(object sender, EventArgs e) { if
(chkhtsothich.Checked == true) panelsothich.Visible = true; else
panelsothich.Visible = false; } void hcubiuNghenghiep(object
sender, EventArgs e) { if (chkhtnghenghiep.Checked == true)
panelnghenghiep.Visible = true; else panelnghenghiep.Visible =
false; } </script> <!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>iTechPro</title> </head> <body>
<form id="form1" runat="server"> <div> <asp:Panel ID="panelpage"
runat="server" GroupingText="iTechPro"> <table> <tr> <td>
<asp:Panel ScrollBars="auto" ID="panelsothich" Width="200px"
Height="200px" runat="server" GroupingText="Sở thích">
<asp:CheckBox ID="CheckBox1" Text="Bóng đá" runat="server" /><br
/> <asp:CheckBox ID="CheckBox2" Text="Xem phim" runat="server"
/><br /> <asp:CheckBox ID="CheckBox3" Text="Kinh doanh"
runat="server" /> </asp:Panel> </td> <td> <asp:Panel
ID="panelnghenghiep" Width="200px" Height="200px"
ScrollBars="Vertical" runat="server" GroupingText="Nghề nghiệp">
<asp:BulletedList ID="buletnghenghiep" runat="server">
</asp:BulletedList> </asp:Panel> </td> </tr> <tr> <td>
<asp:CheckBox Checked="true" ID="chkhtsothich"
AutoPostBack="true" OnCheckedChanged="hcubiuSothich" Text="Hiển
thị sở thích" runat="server" /><br /> <asp:CheckBox
Checked="true" ID="chkhtnghenghiep" AutoPostBack="true"
```

```
OnCheckedChanged="hcubiuNghenghiep" Text="Hiển thị nghề nghiệp"
runat="server" /><br /> </td> </tr> </table> </asp:Panel> </div>
</form> </body> </html>
```

3.11. Điều khiển HyperLink

Điều khiển HyperLink cho phép tạo 1 link tới trang web khác, không giống với LinkButton, HyperLink không đệ trình dữ liệu lên server.

BÀI 2: CƠ BẢN VỀ ASP.NET

Mã Bài: MĐ19_02

Giới thiệu:

Để tạo một ứng dụng web bằng ASP.net cơ bản bạn cần có được những hiểu biết căn bản về ASP.net: kiến trúc của DotNet Framework cùng các phần mềm cần thiết khi sử dụng để tạo ra một ứng dụng web với ASP.net, cùng các ngôn ngữ lập trình được sử dụng trong ASP.net.

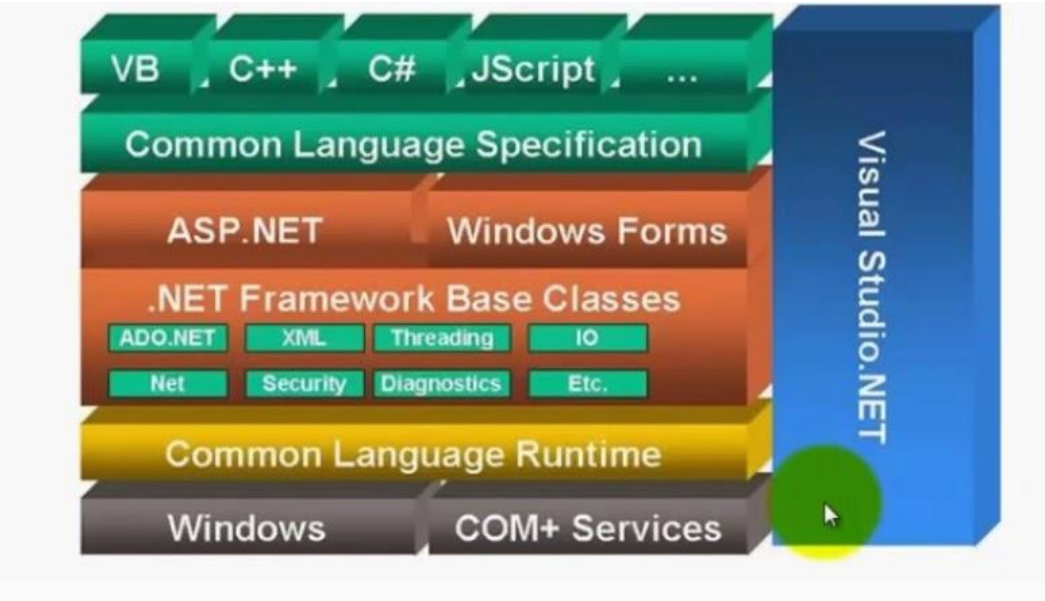
Mục tiêu:

- Trình bày được kiến trúc của .NET.
- Trình bày được quan hệ giữa ASP.NET và kiến trúc .NET
- Trình bày được các lớp trong .NET Framework.
- Sử dụng được ngôn ngữ lập trình ASP.NET.
- Quản lý được chương trình bởi không gian tên.

Nội dung chính:

1. Kiến trúc .NET

Đầu tiên chúng ta sẽ tìm hiểu về kiến trúc của DotNet FrameWork một thành phần không thể thiếu trong lập trình web với ASP.net



Hình 2.1: Cấu trúc của DotNet FrameWork

Cấu trúc của DotNet FrameWork: Gồm 5 phần chính.

Phần trên cùng Common Language specification : phần thể hiện các ngôn ngữ được hỗ trợ trong FrameWord như VB; C#; C++; Jscript…

Phần thứ 2 bao gồm: Windows Forms được sử dụng để xây dựng các ứng trên Destop; ASP.net là phần được dùng để xây dựng các ứng dụng web đây là phần mà chúng ta sẽ cùng nhau tìm hiểu sâu hơn.

Phần thứ 3 là: .Net FrameWork Base class bao gồm các lớp thư viện cơ bản trong DotNet: ADO.Net; XML; Threading IO; .Net; Seurity; Dlagnostlcs; Etc.

Phần 4 gồm: Các dịch vụ chạy DotNet Common language và Runtime được sử dụng để chạy chương trình lập trình sau khi được trình biên dịch chuyển đổi

Phần cuối là các dịch vụ services và hệ điều hành nền tảng Windows.

Để có thể tạo lên một trang web với ASP.net bạn cần có các kỹ năng cơ bản về lập trình web như sau:

+ Kỹ năng lập C#, SQL Server, ngôn ngữ truy vấn T – SQL.

+ Hiểu về cơ chế hoạt động của WebSite và cơ sở dữ liệu.

+ Hiểu về kiến trúc và cơ chế của .Net FrameWork

Yêu cầu đối với máy được sử dụng để tạo một ứng dụng web với ASP.net như sau:

- Cài đặt .Net Framework 4.0 hoặc bản 4.5
- Cài Visual Stdio 2012 hoặc 2010.
- Cài SQL Server 2008.
- Internet Information Services.

Ngoài ra bạn cũng nên có thêm các kiến thức về PhotoShop và HTML để tạo ra các giao diện cho trang web.

2. Mối quan hệ giữa ASP.NET và kiến trúc .NET

ASP.NET là một phần của .NET FrameWork. Để xây dựng trang ASP.NET bạn cần thêm vào các đặc tính của .netframework. NetFrameWork chứa đựng hai phần FrameWork Class Library và Commom Language Runtime.

C# là một ngôn ngữ lập trình hướng đối tượng rất mới. Cấu trúc và lập luận của C# có đầy đủ các đặc tính của một ngôn ngữ lập trình hướng đối tượng trước đó (C++, Java). C# được thiết kế dùng cho nền .NET framework, một công nghệ mới và đầy triển vọng trong việc phát triển các ứng dụng hệ thống và mạng internet. Bên cạnh đó, C# còn là một ngôn ngữ lập trình hoàn toàn độc lập, điều đó có nghĩa là mã của C# được chạy trên .NET nhưng có những đặc tính của C# mà .NET không hỗ trợ (quá tải toán tử) hay là những đặc tính của .NET mà C# không hỗ trợ.

3. Các lớp .NET Framework

Có lẽ một trong những lợi ích lớn nhất của viết mã có quản, ít nhất là đối với một nhà phát triển, đó là bạn có thể sử dụng thư viện lớp cơ sở của .NET.

Thư viện lớp cơ sở của .NET là một tập hợp lớn các lớp mã có quản được viết bởi Microsoft, những lớp này cho phép bạn thao tác rất nhiều các tác vụ sẵn có trong Windows. Bạn có thể tạo các lớp của mình từ các lớp có sẵn trong thư viện lớp cơ sở của .NET dựa trên cơ chế thừa kế đơn.

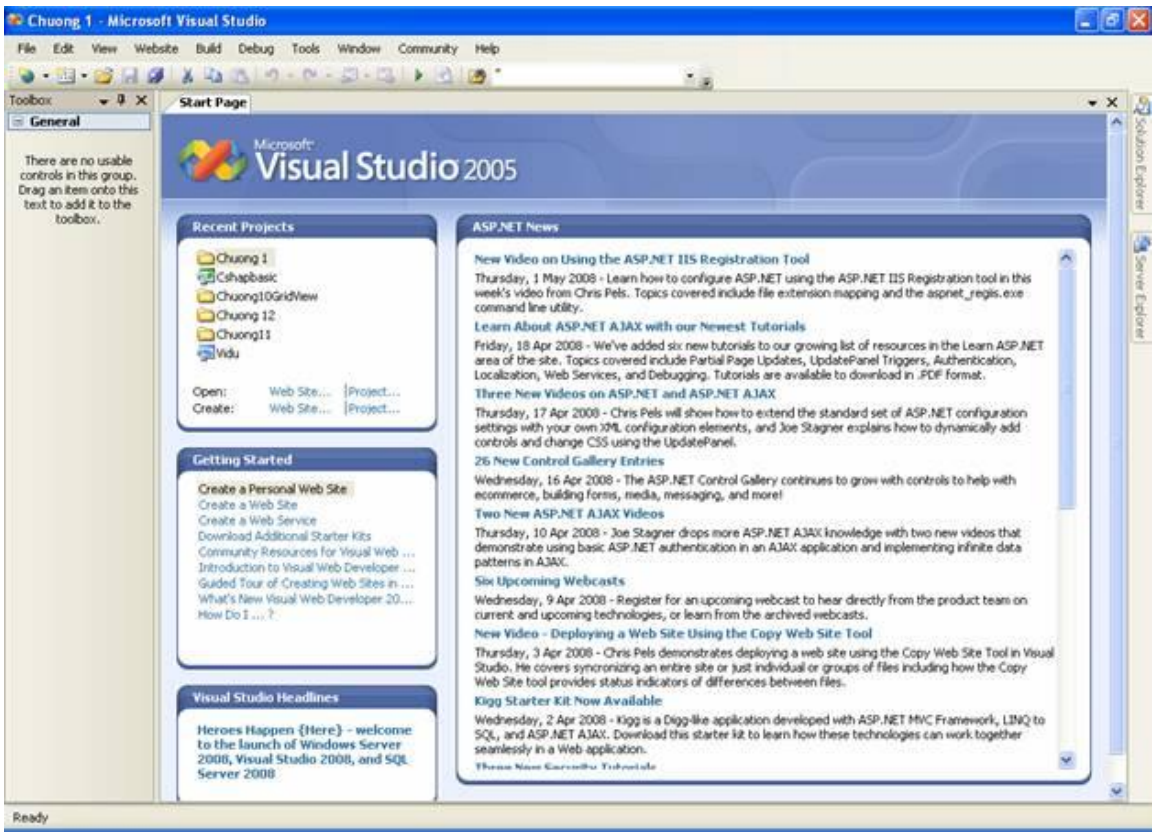
Thư viện lớp cơ sở của .NET rất trực quan và rất dễ sử dụng. Ví dụ, để tạo một tiến trình mới, bạn đơn giản gọi phương thức Start() của lớp Thread. Để disable một TextBox, bạn đặt thuộc tính Enabled của đối tượng TextBox là false. Thư viện này được thiết kế để dễ xài như là Visual Basic và Java. Tất nhiên là nó dễ sử dụng hơn các lớp của C++: các vỏ bọc ngoài các hàm API thô như GetDIBits(), RegisterWndClassEx(), và IsEqualIID().

Mặt khác, những nhà phát triển C++ luôn dễ dàng truy cập đến các API, ngược lại những nhà phát triển Visual Basic và Java đã bị giới hạn trong những thao tác hệ thống cơ bản mà ngôn ngữ đã từng ngôn ngữ đã cung cấp sẵn. Cái mới của thư viện lớp cơ sở .NET là kết hợp tính đơn giản của các thư viện Visual Basic và Java với hầu hết các đặc tính trong các hàm Windows API. Có nhiều đặc tính của Windows không sẵn có trong các lớp của thư viện .NET, trong trường hợp đó bạn cần phải gọi các hàm API, những đặc tính này thường là các đặc tính lạ, ít sử dụng. Những đặc tính thông dụng đều đã được hỗ trợ đầy đủ trong thư viện lớp của .NET. Và nếu bạn muốn gọi một hàm API, .NET gọi là "platform-invoke", cơ chế này luôn bảo đảm tính đúng đắn của kiểu dữ liệu, vì vậy thao tác này không khó hơn việc gọi trực tiếp từ mã C++, nó được hỗ trợ cho cả C#, C++, và VB.NET.

Một cách tình cờ, hầu hết các thư viện lớp cơ sở của .NET được viết bằng C#!

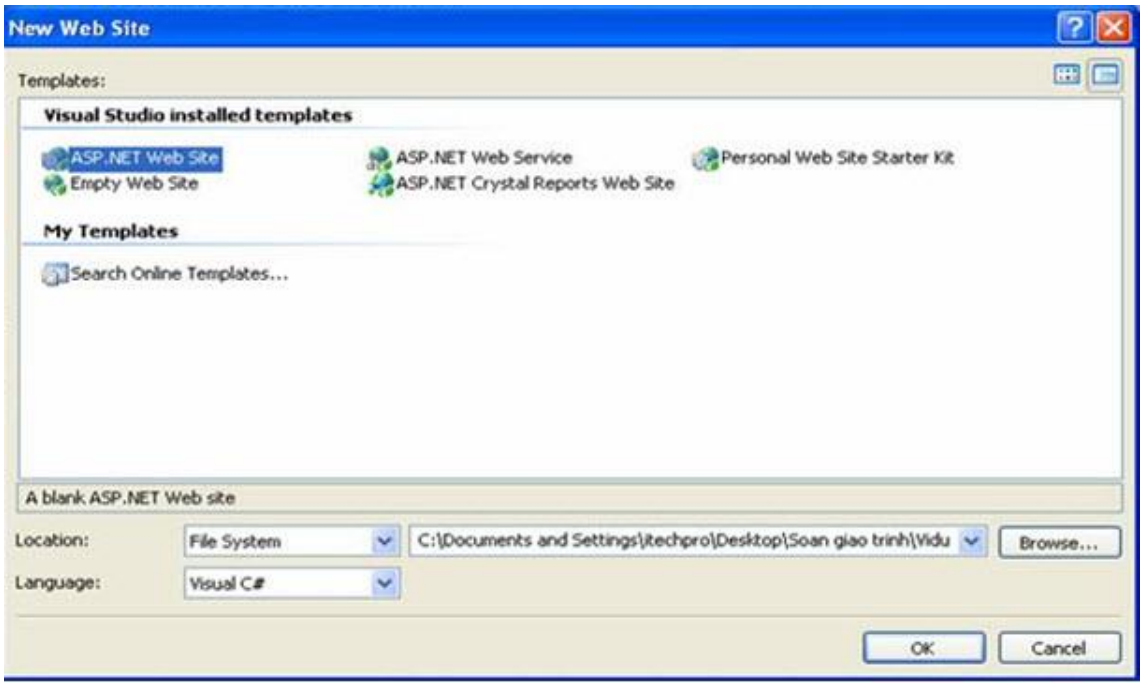
4. Tạo ứng dụng ASP.NET

Để tạo một Wesite mới bạn khởi động VS. giao diện của nó sẽ hiện ra như sau:

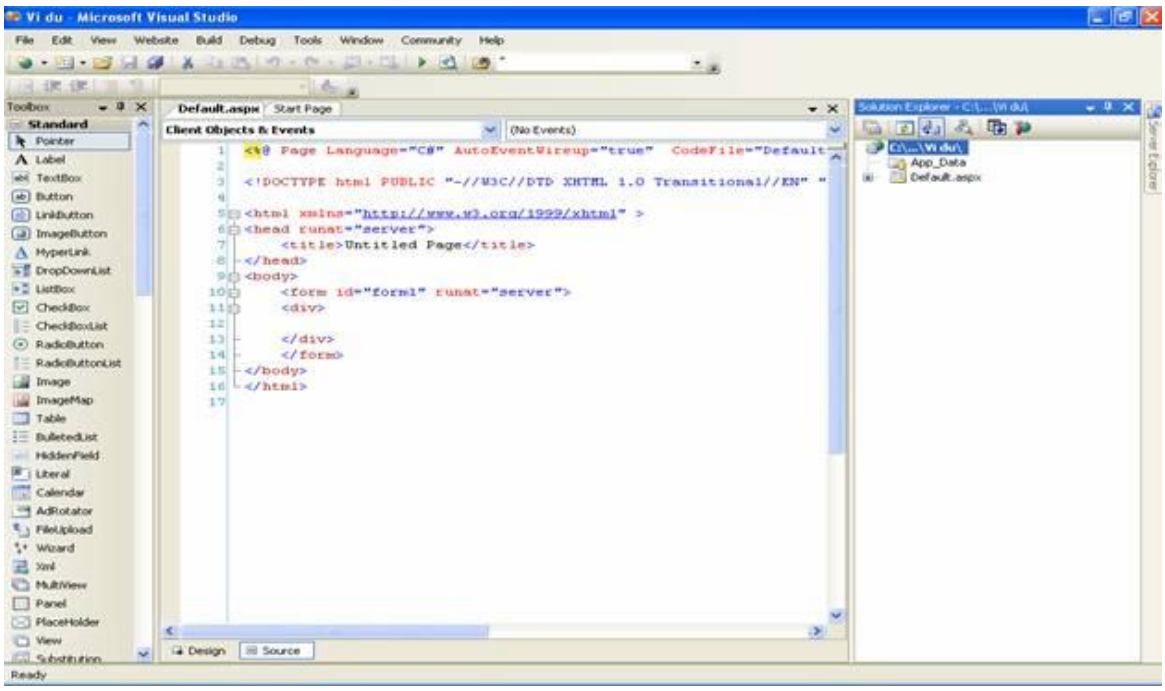


Trong Box Recent Project bạn chọn "Web site…" ở dòng Create

Hộp thoại New Website hiện ra bạn chọn ASP.NET WebSite rồi chọn thư mục bạn để Website và ngôn ngữ kịch bản để bản viết Web và nhấn OK



VS sẽ tạo ra một website với tên của Website là tên bạn vừa đặt và mặc định sẽ có một trang Default.aspx như hình sau:



Trang default.aspx

5. Cấu trúc tổng quát của một chương trình ASP.NET

Framework chứa đựng hàng nghìn lớp mà bạn có thể sử dụng trong ứng dụng của mình. Ví dụ một vài lớp của .Net Framework

-Lớp File: cho phép bạn tạo file, sửa, xoá hay kiểm tra sự tồn tại của file trên đĩa cứng…

-Lớp Graphics: Cho phép bạn làm việc với nhiều kiểu của ảnh, bạn cũng có thể tạo ra các ảnh từ các phương thức trên lớp này.

-Lớp SmtpClient: Cho phép bạn gửi thư.

Hiểu về Namespaces: hơn 13 nghìn lớp trong Netframework. Đây là một con số rất lớn, Microsoft đã chia các lớp cùng xử lý về một vấn đề gì đó vào các không gian tên chung hay namespaces.

Một Namespace đơn giản là một danh mục, ví dụ tất cả các lớp thao tác với File và thư mục chúng ta đưa vào một namespaces chung gọi là System.IO, hay tất cả các lớp làm việc với SqlServer có thể đưa vào namespace System.Data.SqlClient.

Các namespaces chung nhất trong net:

. System

. System.Collections

. System.Collections.Specialized

. System.Configuration

. System.Text

. System.Text.RegularExpressions

. System.Web

. System.Web.Caching

. System.Web.SessionState

. System.Web.Security

. System.Web.Profile

. System.Web.UI

. System.Web.UI.WebControls

. System.Web.UI.WebControls.WebParts

Các điều khiển asp.net là phần quan trọng nhất trong ASP.NET Framework. một Control ASP.NET là một lớp mà thực thi trên server và đưa ra nội dung trên trình duyệt. ASP.NET có hơn 70 control mà bạn có thể sử dụng trong xây dựng ứng dụng web của bạn và cơ bản nó chia ra các nhóm control sau:

Standard control: bao gồm các điều khiển đưa ra các thành phần chuẩn của form như: Label, Button, TextBox…

Validator Control: là các control cho phép bản kiểm tra tính hợp lệ của các control cho phép nhập giá trị trên form.

Rich Control: là những điều khiển như FileUpload, Calendar…

Data Control là các điều khiển cho phép thao tác với dữ liệu

Navigation Control: là những điều khiển giúp bạn dễ dàng di chuyển giữa các trang trong website.

Login control: Là các điều khiển về bảo mật của ứng dụng cho phép bạn đưa ra các form đăng nhập, thay đổi mật khẩu…

HTML Control: cho phép bạn chuyển các điều khiển của HTML thành các điều khiển có thể làm việc trên server.

Hiểu về trang asp.net:

Thay vì sử dụng <script runat=”sever”/> </script> ngay trên trang asp.net. người ta đưa ra thêm một trang gọi là Code-behind chứa các mã lệnh thực hiện trên trang asp.net.

Ví dụ như phần đầu của chương khi ta tạo ra một website thì mặc định sẽ tạo ra một lớp Default.aspx và nó sẽ kèm theo một trang Default.aspx.cs.

Trang 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
> <head runat="server"> <title>Default</title> </head> <body>
<form id="form1" runat="server"> <div> </div> </form> </body>
</html>
```

Trang Defautl.aspx.cs

```
using System; using System.Data; using System.Configuration;
using System.Collections; using System.Web; using
System.Web.Security; using System.Web.UI; using
System.Web.UI.WebControls; using
System.Web.UI.WebControls.WebParts; using
System.Web.UI.HtmlControls; public partial class _Default :
System.Web.UI.Page { protected void Page_Load(object sender,
EventArgs e) { } }
```

6. Biến và Hằng

6.1. Biến

Biến là một đại lọợng dùng để chứa dữ liệu tạm thời trong quá trình tính toán. Tất cả các biến được sử dụng trong chương trình đều phải được khai báo ngay từ đầu, biến được chia thành 3 loại bao gồm: Biến đầu vào, biến đầu ra và biến trung gian.

Biến có thể được khai báo tại 2 nơi gồm:

+ Bên trong phần định nghĩa lớp của Form.
+ Bên trong một phương thức.

```
public partial class frmWelcome : Form
{
// Nơi khai báo biến;
private void bntDisplay_Click(object sender, EventArgs e)
{
// Nơi khai báo biến;
}
}
```

Phạm vi của một biến: phụ thuộc vào vị trí khai báo của biến, nếu biến được khai báo trong phần định nghĩa lớp của Form thì là biến toàn cục có tác dụng trong toàn bộ các đoạn mã lệnh của form, nếu biến được khai báo bên trong một phương thức thì là biến cục bộ chỉ có tác dụng trong phương thức chứa nó.

Biến giống như những chiếc hộp trong bộ nhớ có khả năng lưu giữ giá trị, có nhiều kiểu giá trị khác nhau mà C# có thể xử lý như: kiểu số nguyên, kiểu số thực, kiểu ký tự… Khi khai báo một biến ta phải chỉ ra kiểu giá trị mà nó sẽ lưu trữ.

Cú pháp khai báo biến như sau:

Kiểu dữ liệu *Tênbiến* **[**=*Giá trị***];**

Tênbiến: là một chuỗi các ký tự do người lập trình tự đặt bao gồm các chữ cái, chữ số và dấu gạch dọới. Tên biến phải bắt đầu bằng một chữ cái, không được chứa dấu cách, C# phân biệt chữ hoa chữ thường.

6.2. Hằng

Hằng là đại lọợng dùng để chứa những dữ liệu có giá trị không đổi trong suốt quá trình tính toán. Sử dụng hằng làm chương trình sáng sủa dễ đọc nhờ tên gọi gợi nhớ thay vì các con số.

Hằng được khai báo theo cú pháp sau:

const *Kiểu dữ liệu* **Tênhằng =** *Giá trị;*

7. Dữ liệu kiểu trị và kiểu qui chiếu

C# là một ngôn ngữ được kiểm soát chặt chẻ về mặt kiểu dữ liệu, ngoài ra C# còn chia các kiểu dữ liệu thành hai loại khác nhau: kiểu trị (value type) và kiểu qui chiếu (reference type). Nghĩa là trên một chương trình C# dữ liệu được lưu trữ một hoặc hai nơi tuỳ theo đặc thù của kiểu dữ liệu.

Chỗ thứ nhất là stack một vùng ký ức dành lưu trữ dữ liệu chiều dài cố định, chẳng hạn int chiếm dụng 4 bytes . Mỗi chương trình khi đang thi hành đều được cấp phát riêng một stack riêng biệt mà các chương trình khác không được mó tới. Khi một hàm được gọi hàm thi hành thì tất cả các biến cục bộ của hàm được ấn vào stack và khi hàm hoàn thành công tác thì những biến cục bộ của hàm đều bị tống ra. Đây là cách thu hồi khi hàm hết hoạt động.

Chỗ thứ hai là heap, một vùng ký ức dùng lưu trữ dữ liệu có bề dày thay đổi và khá đồ sộ, string chẳng hạn, hoặc dữ liệu có một cuộc sống dài hơn phương thức của một đối tượng chẳng hạn, Thí dụ khi phương thức thể hiện (instantiate) một đối tượng , đối tượng đuợc lưu trữ trên heap, và nó không bị tống ra khi hàm hoàn thành giống như stack, mà ở nguyên tại chỗ và có thể trao cho các phương thức khác thông qua một qui chiếu. Trên C# heap này được gọi là managed heap, khôn

lanh vì heap này có một bộ phận gọi là garbage collector (GC,dịch vụ hốt rác) chuyên lo thu hồi ký ức lâu ngày không dùng đến (nghĩa là không quy chiếu đến).

C# cũng hỗ trợ kiểu con trỏ (pointer type) giống như C++ nhưng ít khi dùng đến và chỉ dùng khi làm việc với đoạn mã unmanaged. Đoạn mã unmanaged là đoạn mã được tạo ra ngoài sàn diễn .NET, chẳng hạn những đối tượng COM.

7.1. Kiểu giá trị được định nghĩa trước (Predefined Value Types)

Kiểu dữ liệu bẩm sinh (The built-in value types) trình bày ban đầu như integer và floating-point numbers, character, và Boolean types.

7.1.1. Các kiểu Integer

C# hỗ trợ 8 kiểu dữ liệu số nguyên sau:

Name	CTS Type	Description	Range (min:max)
sbyte	System.SByte	8-bit signed integer	-128:127 (-2^7:2^7-1)
short	System.Int16	16-bit signed integer	-32,768:32,767 (-2^{15}:2^{15}-1)
int	System.Int32	32-bit signed integer	-2,147,483,648:2,147,483,647 (-2^{31}:2^{31}-1)
long	System.Int64	64-bit signed integer	-9,223,372,036,854,775,808: 9,223,372,036,854,775,807 (-2^{63}:2^{63}-1)
byte	System.Byte	8-bit signed integer	0:255 (0:2^8-1)
ushort	System.UInt16	16-bit signed integer	0:65,535 (0:2^{16}-1)
uint	System.UInt32	32-bit signed integer	0:4,294,967,295 (0:2^{32}-1)
ulong	System.UInt64	64-bit signed integer	0:18,446,744,073,709,551,615 (0:2^{64}-1)

Ví dụ:

```
long x = 0x12ab;// ghi theo hexa
uint ui = 1234U;
long l = 1234L;
ulong ul = 1234UL;
```

7.1.2. Kiểu dữ liệu số dấu chấm di động (Floating Point Types)

Name	CTS Type	Description	Significant Figures	Range (approximate)
Float	System.Single	32-bit single-precision floating-point	7	±1.5 × 10-45 to ±3.4 × 1038
Double	System.Double	64-bit double-precision floating-point	15/16	±5.0 × 10-324 to ±1.7 × 10308

Ví dụ:

float f = 12.3F;

7.1.3. Kiểu dữ liệu số thập phân (Decimal Type)

Name	CTS Type	Description	Significant Figures	Range (approximate)
decimal	System.Decimal	128-bit high precision decimal notation	28	±1.0 × 10-28 to ±7.9 × 1028

Ví dụ:

decimal d = 12.30M ; //có thể viết decimal d = 12.30m;

7.1.4. Kiểu Boolean

Name	CTS Type	Value
Bool	System.Boolean	true or false

7.1.5. Kiểu Character Type

Name	CTS Type	Value

char	System.Char	Represents a single 16-bit (Unicode) character

7.2. Kiểu tham khảo tiền định nghĩa

C# hỗ trợ hai kiểu dữ liệu được định nghĩa trước:

Name	CTS Type	Description
object	System.Object	The root type, from which all other types in the CTS derive (including value types)
string	System.String	Unicode character string

Các ký tự escape thông dụng:

Escape Sequence	Character
\'	Single quote
\"	Double quote
\\	Backslash
\0	Null
\a	Alert
\b	Backspace
\f	Form feed
\n	Newline
\r	Carriage return
\t	Tab character
\v	Vertical tab

Các kiểu chuỗi :

Đối tượng kiểu string thường chứa một chuỗi ký tự.Khi khai báo một biến chuỗi sử dụng từ khoá string giống như sau:

string myString;

Thường thì phải khởi gán một biến chuỗi sử dụng đến một kiểu string:

```
string myString = "Xin chao" ;
 string str1 = "Hello ";
string str2 = "World";
string str3 = str1 + str2; //
```

Ví dụ:

```
using System;
class StringExample
{
   public static int Main()
   {
      string s1 = "a string";
      string s2 = s1;
      Console.WriteLine("s1 is " + s1);
      Console.WriteLine("s2 is " + s2);
      s1 = "another string";
      Console.WriteLine("s1 is now " + s1);
      Console.WriteLine("s2 is now " + s2);
      return 0;
   }
}
```

Kết quả như sau:

```
s1 is a string
s2 is a string
s1 is now another string
s2 is now a string
```

Chuỗi được khai báo là một đường dẫn:

```
string filepath = "C:\\ProCSharp\\First.cs";
```

8. Các câu lệnh điều khiển

8.1. Câu lệnh lựa chọn if

Dạng 1:

```
if (Điềukiện)
{
Khối lệnh;
```

```
}
```

Hoạt động: Nếu <Điều kiện> nhận giá trị đúng thì <Khối lệnh> được thực hiện.

Dạng 2:

```
if (Điều kiện)
{

}
else
{

}
Khối lệnh 1;




Khối lệnh 2;
```

Hoạt động: Nếu <Điều kiện> nhận giá trị đúng thì <Khối lệnh 1> được thực hiện, <Khối lệnh

2> bị bỏ qua. Ngợc lại nếu <Điều kiện> nhận giá trị sai thì <Khối lệnh 2> được thực hiện,

<Khối lệnh 1> bị bỏ qua.

8.2. Câu lệnh lựa chọn Case

```
switch (Biểu thức kiểm tra)
{
case <Biểu thức 1>: Khối lệnh 1; break;
case <Biểu thức 2>:
Khối lệnh 2;
break;
……..
default:
Khối lệnh n+1;
break;
}
```

Hoạt động: máy so sánh giá trị của <Biểu thức kiểm tra> với giá trị của các <Biểu thức i>.

+ Nếu <Biểu thức kiểm tra> có giá trị thỏa mãn <Biểu thức i> thì <Khối lệnh i> được thực hiện, sau đó máy sẽ thoát ngay ra khỏi câu lệnh switch.

+ Nếu <Biểu thức kiểm tra> không thỏa mãn <Biểu thức i> nào thì <Khối lệnh n+1> được thực hiện.

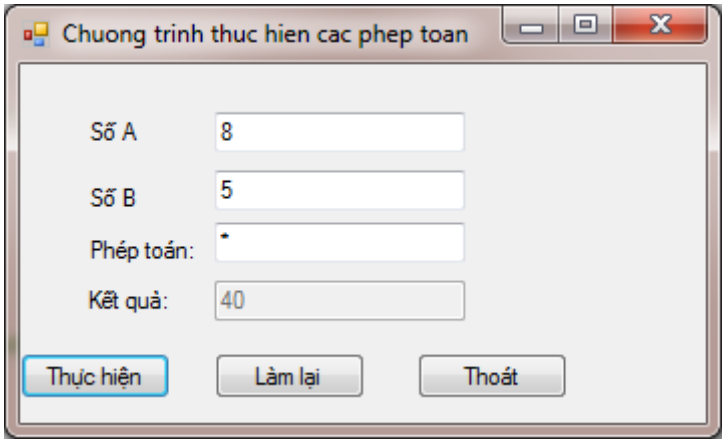
Chú ý: nếu các biểu thức khác nhau cùng thực hiện chung một khối lệnh thì ta có thể viết gộp như sau:

```
switch (Biểu thức kiểm tra)
{
case <Biểu thức 1>:
case <Biểu thức 2>: Khối lệnh;
break;
……..
default:
Khối lệnh n+1;
break;
}
```

Bài tập 1.

Xây dựng chương trình thực hiện các phép toán theo giao diện sau (các phép toán bao gồm:

+, - , *, /, %).



Hình 8.1:Giao diện bài tập 1

Yêu cầu: + Nút thực hiện có tác dụng thực hiện phép toán đối với số A và số B, kết quả lưu

vào ô kết quả.

+ Kết quả chỉ được tính khi người dùng nhập đủ giá trị cho số A, B và phép toán.

+ Phép toán chia phải kiểm tra trọờng hợp mẫu =0.

+ Ô kết quả không được phép chỉnh sửa dữ liệu.

Tạo dự án mới và thiết lập các thuộc tính của các điều khiển như sau:

Điều	Thuộc tính	Giá trị
Form1	Name	frmPheptoan
	FormBorderSty	Fixed3D
	Icon	Chọn file ảnh có đuôi .ico bất kỳ
	Text	Chuong trinh thuc hien cac phep
Lable1	Text	Số A
Lable2	Text	Số B
Lable3	Text	Phép toán:
Lable4	Text	Kết quả:
TextBox1	Name	txtSoA
TextBox2	Name	txtSoB
TextBox3	Name	txtPheptoan
TextBox4	Name	txtKetqua
Button1	Name	btnThuchien
	Text	&Thuchien
Button2	Name	btnLamlai
	Text	&Làm lại
Button3	Name	btnThoat
	Text	T&hoát

Mã lệnh:

```
uing System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.DrawingSystem.Linq;
using System.Text;
```

```
using System.Windows.Forms;

namespace WindowsFormsApplication3
{
public partial class frmPheptoan : Form
{

public frmPheptoan()
{
InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
txtKetqua.Enabled = false;
btnLamlai.Enabled = false;
}

private void btnThuchien_Click(object sender, EventArgs e)
{
int a, b, kq;
if (txtSoA.Text == "")
{
MessageBox.Show("Bạn phải nhập A", "Thông báo",
MessageBoxButtons.OK, MessageBoxIcon.Information);
txtSoA.Focus();
return;
}
if (txtSoB.Text == "")
{
MessageBox.Show("Bạn phải nhập B", "Thông báo",
MessageBoxButtons.OK, MessageBoxIcon.Information);
txtSoB.Focus();
return;
}
```

```
a = Convert.ToInt16(txtSoA.Text); b =
Convert.ToInt16(txtSoB.Text); switch (txtPheptoan.Text)
{
case "+":
kq=a+b;
txtKetqua.Text = kq.ToString();
break;
case "-":
kq=a-b;
txtKetqua.Text = kq.ToString();
break;
case "*":
kq = a * b;
txtKetqua.Text = kq.ToString();
break;
case "/":
if (txtSoB.Text == "0")
{
MessageBox.Show("Giá trị B phải khác 0!", "Thông Báo",
MessageBoxButtons.OK, MessageBoxIcon.Error);
txtSoB.Text = "";
txtSoB.Focus();
return;
}
kq = a / b;
txtKetqua.Text = kq.ToString();
break;
case "%":
kq = a % b;
txtKetqua.Text = kq.ToString();
break;
default:
MessageBox.Show("Bạn phải nhập lại phép toán", "Thông Báo",
MessageBoxButtons.OK, MessageBoxIcon.Error);
txtPheptoan.Text = ""; txtPheptoan.Focus(); return;
```

```
}
btnThuchien.Enabled = false;
btnLamlai.Enabled = true;
txtSoA.ReadOnly =true ;
txtSoB.ReadOnly = true ;
txtPheptoan.ReadOnly = true;
}

private void btnLamlai_Click(object sender, EventArgs e)
{
txtSoA.Text = "";
txtSoA.ReadOnly = false;
txtSoB.Text ="";
txtSoB.ReadOnly = false;
txtPheptoan.Text = "";
txtPheptoan.ReadOnly = false;
txtKetqua.Text = "";
btnThuchien.Enabled = true;
txtSoA.Focus();
}

private void btnThoat_Click(object sender, EventArgs e)
{
if (MessageBox.Show("Bạn có muốn thoát không?", "Thông báo",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
System.Windows.Forms.DialogResult.Yes)
Application.Exit();
}
}
}
```

8.3. Cấu trúc lặp for

Cho phép thực hiện lặp đi lặp lại một đoạn chương trình nhiều lần, với số lần lặp xác định.

```
for (biểu thức khởi tạo; biểu thức điều kiện; biểu thức cập
nhật)
{
Khối lệnh;
[break;]
}
```

Hoạt động: Đầu tiên máy thực hiện biểu thức khởi tạo để khởi tạo giá trị của biến điều khiển, sau đó máy kiểm tra giá trị của biểu thức điều kiện, nếu biểu thức này đúng thì <Khối lệnh> được thực hiện và cập nhật giá trị của biến điều khiển thông qua biểu thức cập nhật. Sau đó quay lại kiểm tra giá trị của biểu thức điều kiện, cứ lặp lại như vậy cho đến khi biểu thức điều kiện nhận giá trị sai thì dừng lại.

Chú ý: để thoát ngay ra khỏi vòng lặp for ta có thể dùng lệnh break.

Ví dụ: Dùng vòng lặp for để khởi tạo các giá trị ngẫu nhiên trong khoảng (0, 100) cho mảng một chiều gồm 10 phần tử.

Mở một dự án mới rồi gõ đoạn mã sau vào cửa sổ code.

```
private void Form1_Load(object sender, EventArgs e)
{
int[] m = new int[10];
Random rnd = new Random();
for(int i = 0; i<10; i++)
{
m[i] = rnd.Next(0,100); MessageBox.Show(m[i].ToString());
}
}
```

Bấm F5 để thực hiện chương trình.

8.4. Cấu trúc lặp while

Cho phép thực hiện lặp đi lặp lại một đoạn chương trình nhiều lần, với số lần lặp không được xác định trước.

```
while (Biểu thức điều kiện)
{
Khối lệnh;
[break;]
```

```
}
```

Hoạt động: Đầu tiên máy kiểm tra giá trị của <Biểu thức điều kiện>, nếu biểu thức này nhận giá trị đúng thì <Khối lệnh> được thực hiện. Sau đó lại quay lại kiểm tra giá trị của <Biểu thức điều kiện>, cứ lặp lại như vậy cho đến khi <Biểu thức điều kiện> nhận giá trị sai thì dừng lại.

Chú ý: + Vì <Biểu thức điều kiện> được kiểm tra trước, nên <Khối lệnh> có thể không được thực hiện lần nào nếu ngay từ đầu <Biểu thức điều kiện> đã nhận giá trị sai và trước khi thực hiện khối lệnh phải khởi gán giá trị cho <Biểu thức điều kiện>.

+ Trong <Khối lệnh> phải có ít nhất một lệnh làm thay đổi giá trị của <Biểu thức điều kiện> để đến một lúc nào đó <Biểu thức điều kiện> nhận giá trị sai, nhằm dừng vòng lặp lại, nếu không nó sẽ lặp mãi không dừng.

Ví dụ: Nhập số thực a, tìm số tự nhiên n nhỏ nhất sao cho tổng: T = 1+1/2+…+1/n >= a

Mở một dự án mới rồi gõ đoạn mã sau vào cửa sổ code.

```
private void Form1_Load(object sender, EventArgs e)
{
double a= 2;
double T= 0;
int n = 0;
while (T<a)
{
n = n + 1;
T = T + 1.0/n;
}
MessageBox.Show("Gia tri n thoa man = " + n.ToString ());
}
```

Bấm F5 thực hiện chương trình, có thể thay đổi giá trị của a để có những kết quả khác nhau.

8.5. Cấu trúc lặp do

Cho phép thực hiện lặp đi lặp lại một đoạn chương trình nhiều lần, với số lần lặp không được xác định trước.

```
do
{
Khối lệnh; [break;]
}
while (Biểu thức điều kiện);
```

Hoạt động: Đầu tiên máy thực hiện <Khối lệnh>, sau đó kiểm tra giá trị của <Biểu thức điều kiện>, nếu biểu thức này nhận giá trị đúng thì tiếp tục thực hiện <Khối lệnh>, cứ lặp lại như vậy cho đến khi <Biểu thức điều kiện> nhận giá trị sai thì dừng lại.

Chú ý: + Vì <Biểu thức điều kiện> được kiểm tra sau, nên <Khối lệnh> luôn được thực hiện ít nhất 1 lần.

+ Trong <Khối lệnh> phải có ít nhất một lệnh làm thay đổi giá trị của <Biểu thức điều kiện> nhằm dừng vòng lặp lại.

Ví dụ: Tính tổng T = 1 + 2 + … + 10

Mở một dự án mới rồi gõ đoạn mã sau vào cửa sổ code.

```
private void Form1_Load(object sender, EventArgs e)
{
int T = 0; int i = 1; do
{
T = T + i;
i = i + 1;
}
while (i <= 10);
MessageBox.Show("T = " + T.ToString ());
}
```

8.6. Câu lệnh try…catch

Được dùng trong các câu lệnh bẫy lỗi của chương trình, cho phép bắt một số lỗi trong quá trình thực thi ứng dụng, ví dụ: biến các ký tự không phải dạng số thành số, thực hiện phép chia cho 0, sử dụng biến null…

Cú pháp bẫy lỗi trong C# được thể hiện như sau:

```
try
{
// mã cho việc thực thi bình thường
}
```

```
catch (System.Exception)
{
// xử lí lỗi
}
finally
{
// dọn dẹp
}
```

Cú pháp trên gồm 3 khối:

+ Khối *try* chứa đựng đoạn mã cần phải thực thi trong chương trình, nhưng đoạn mã này có thể gặp phải một vài trạng thái lỗi.

+ Khối *catch* chứa đựng đoạn mã giải quyết những những lỗi xẩy ra trong *try*, tham số của *catch* là các lớp bắt lỗi. C# có rất nhiều lớp bắt lỗi, trong đó *System.Exception* là lớp ở mức cao nhất có thể bắt được mọi loại lỗi xẩy ra trong *try*.

+ Khối *finally* chứa đựng đoạn mã dọn dẹp tài nguyên hoặc bất kì hành động nào bạn muốn thực hiện sau khối *try* hay *catch*, khối này có thể có hoặc không.

Hoạt động: Đầu tiên chương trình thực thi các câu lệnh trong khối *try*, nếu không xuất hiện lỗi thì các câu lệnh được thực hiện bình thường sau đó sẽ nhảy đến thực hiện các câu lệnh trong khối *finally*, tuy nhiên nếu xuất hiện lỗi trong khối *try* thì chương trình sẽ tự động nhảy ngay tới thực thi các câu lệnh trong khối *catch* mà không đột ngột dừng chương trình và sau đó cũng thực hiện các câu lệnh trong khối *finally*.

Chú ý: C# không cho phép đặt lệnh `return` bên trong khối *finally*.

Ví dụ: Bẫy lỗi đoạn chương trình tính tổng 2 số nguyên a và b trong trọờng hợp không nhập dữ liệu dạng số.

```
private void bntTong_Click(object sender, EventArgs e)
{
try
{
txtTong.Text = Convert.ToString(Convert.ToInt32(txtSoA.Text)
+ Convert.ToInt32(txtSoB.Text));
}
catch (System.Exception ex)
{
```

```
}
```

9. Mảng

9.1. Giới thiệu về mảng trong C#

– Trong chương trình đôi khi chúng ta cần thực hiện lưu trữ một loạt các thông tin có cùng kiểu dữ liệu để làm điều này thường chúng ta sẽ phải khai báo rất nhiều biến tương ứng với mỗi một thông tin như vậy. Việc quản lý số lượng lớn các biến gây ra rất nhiều khó khăn. Để giải quyết các khó khăn này thì C# cũng như các ngôn ngữ lập trình khác cung cấp một kiểu cấu trúc dữ liệu cho phép quản lý các biến có cùng tên, cùng kiểu dữ liệu. Đó là mảng.

– Mảng là một tập hợp hữu hạn các phần tử có cùng kiểu dữ liệu. Có thể coi mảng là một cách quản lý danh sách các biến có cùng kiểu dữ liệu.

– Để khai báo mảng trong C# chúng ta sử dụng cú pháp:

[Kiểu dữ liệu] [ ] … [ ] [tên mảng];

Trong đó:

[Kiểu dữ liệu]: là bất kì một kiểu dữ liệu nào được hỗ trợ trong C#.

[Tên mảng]: là một định danh để phân biệt các mảng với các biến, các đối tượng khác.

Số lượng các cặp dấu [ ] : chỉ ra chiều của mảng.

Nếu có 1 dấu [ ] thì mảng là mảng 1 chiều [].

Nếu có 2 dấu [ ] thì mảng là mảng 2 chiều [][].

Nếu có 3 dấu [ ] thì mảng là mảng 3 chiều [][][].

…….

Nếu có n dấu [ ] thì mảng là mảng n chiều. [][]….[].

Ví dụ: Khai báo mảng 1 chiều lưu trữ phần tử nguyên có tên là x. int[] x;

– Khởi tạo mảng: là quá trình khai báo số lượng các phần tử của mảng và gán giá trị cho các phần tử đó. Để khởi tạo mảng trong C# chúng ta có các cách như sau:

Khai báo và chỉ ra số phần tử của mảng.

[Kiểu dữ liệu] [ ] [tên mảng] = new [Kiểu dữ liệu] [số lượng phần tử];

Ví dụ: khai báo một mảng số nguyên lưu trữ 20 phần tử.

```
int[] x = new int[20];
```

Khai báo sau đó mới khởi tạo số phần tử của mảng.

[Kiểu dữ liệu] [ ] [tên mảng];

[tên mảng] = new [kiểu dữ liệu] [số phần tử];

Ví dụ: khai báo một mảng số nguyên lưu trữ 20 phần tử.

```
int[] x;
x = new int[20];
```

Khai báo, chỉ ra số lượng các phần tử mảng và gán các giá trị ban đầu cho các phần tử mảng.

```
int[] x = new int[5] { 5, -7, 8, 1, 9 };
```

Khai báo, ko chỉ ra số lượng các phần tử mảng và gán giá trị cho các phần tử của mảng.

– Truy cập các phần tử mảng: một mảng là một danh sách các phần tử có cùng kiểu dữ liệu, các phần tử đó được đánh số thứ tự bắt đầu từ 0 đến n-1 (Trong đó n là số phần tử của mảng).

– Như vậy để truy cập đến 1 phần tử của mảng thì chúng ta sử dụng một số nguyên để chỉ ra số thứ tự của phần tử đó trong mảng, phần tử nguyên này được gọi là chỉ số (index).

– Cú pháp tổng quát để truy cập đến phần tử thứ i của mảng là: [tên mảng] [i-1];

Ví dụ : gán giá trị -9 cho phần tử thứ 3 của mảng : x[2] = -9;

– Duyệt qua các phần tử mảng. Kể từ khi mảng là một tập hợp các hữu hạn phần tử do đó để duyệt qua các phần tử mảng thường chúng ta sử dụng vòng lặp for.

Ví dụ: in danh sách các phần tử mảng x ở trên.

```
Console.WriteLine(“Danh sach cac phan tu trong mang:”);
for (int i = 0; i < 5; i++)
Console.Write(x[i] + “t”);
```

9.2. Các kiểu mảng trong C#

Có 3 kiểu mảng trong C#:

– Mảng 1 chiều (single-dimension array): mảng 1 chiều là một mảng các phần tử. Mảng 1 chiều được khai báo chỉ bao gồm 1 cặp dấu [ ].

– Mảng nhiều chiều (multi-dimension array): mảng nhiều chiều là mảng có số chiều từ 2 trở lên (có từ 2 cặp [ ] trở lên).

– Mảng nhọn (jagged array): mảng nhọn là một mảng của các mảng, trong đó kích thước của mỗi một mảng có thể khác nhau. Mảng nhọn cho phép chúng ta sử dụng bộ nhớ một cách tiết kiệm hơn.

10. Các phép toán

Toán tử gán

Đến lúc này toán tử gán khá quen thuộc với chúng ta, hầu hết các chương trình minh họa từ đầu sách đều đã sử dụng phép gán. Toán tử gán hay phép gán làm cho toán hạng bên trái thay đổi giá trị bằng với giá trị của toán hạng bên phải. Toán tử gán là toán tử hai ngôi.

Đây là toán tử đơn giản nhất thông dụng nhất và cũng dễ sử dụng nhất.

Toán tử toán học

Ngôn ngữ C# cung cấp năm toán tử toán học, bao gồm bốn toán tử đầu các phép toán cơ bản. Toán tử cuối cùng là toán tử chia nguyên lấy phần dư. Chúng ta sẽ tìm hiểu chi tiết các phép toán này trong phần tiếp sau.

Các phép toán số học cơ bản (+,-,*,/)

Các phép toán này không thể thiếu trong bất cứ ngôn ngữ lập trình nào, C# cũng không ngoại lệ, các phép toán số học đơn giản nhưng rất cần thiết bao gồm: phép cộng (+), phép trừ (-), phép nhân (*), phép chia (/) nguyên và không nguyên.

Khi chia hai số nguyên, thì C# sẽ bỏ phần phân số, hay bỏ phần dư, tức là nếu ta chia 8/3 thì sẽ được kết quả là 2 và sẽ bỏ phần dư là 2, do vậy để lấy được phần dư này thì C# cung cấp thêm toán tử lấy dư sẽ được trình bày trong phần kế tiếp.

Tuy nhiên, khi chia cho số thực có kiểu như float, double, hay decimal thì kết quả chia được trả về là một số thực.

Phép toán chia lấy dư

Để tìm phần dư của phép chia nguyên, chúng ta sử dụng toán tử chia lấy dư (%). VíThật sự phép toán chia lấy dư rất hữu dụng cho người lập trình . Khi chúng ta thực hiện một phép chia dư n cho một số khác, nếu số này là bội số của n thì kết quả của phép chia dư là 0.

20 % 5 = 0 vì 20 là một bội số của 5.

Điều này cho phép chúng ta ứng dụng trong vòng lặp, khi muốn thực hiện một công việc nào đó cách khoảng n lần, ta chỉ cần kiểm tra phép chia dư n, nếu kết quả bằng 0 thì thực hiện công việc. Cách sử dụng này đã áp dụng trong ví dụ minh họa sử dụng vòng lặp for bên trên. Ví dụ sau minh họa sử dụng các phép toán chia trên các số nguyên, thực...

Phép chia và phép chia lấy dư.

```
using System;
    class Tester
```

```
{
public static void Main()
{
int i1, i2;
float f1, f2;
double d1, d2;
decimal dec1, dec2;
i1 = 17;
i2 = 4;
f1 = 17f; f2 = 4f; d1 = 17; d2 = 4;
dec1 = 17;
dec2 = 4;
Console.WriteLine("Integer: \t{0}", i1/i2);
Console.WriteLine("Float: \t{0}", f1/f2);
Console.WriteLine("Double: \t{0}", d1/d2);
Console.WriteLine("Decimal: \t{0}", dec1/dec2);
Console.WriteLine("\nModulus: : \t{0}", i1%i2);
}
}
```

Kết quả:

Integer: 4

float: 4.25 double: 4.25 decimal: 4.25

Modulus: 1

Toán tử tăng và giảm

Khi sử dụng các biến số ta thường có thao tác là cộng một giá trị vào biến, trừ đi một giá trị từ biến đó, hay thực hiện các tính toán thay đổi giá trị của biến sau đó gán giá trị mới vừa tính toán cho chính biến đó.

Tính toán và gán trở lại

Giả sử chúng ta có một biến tên Luong lưu giá trị lương của một người, biến Luong này có giá trị hiện thời là 1.500.000, sau đó để tăng thêm 200.000 ta có thể viết như sau:

Trong câu lệnh trên phép cộng được thực hiện trước, khi đó kết quả của vế phải là 1.700.000 và kết quả này sẽ được gán lại cho biến Luong, cuối cùng Luong có giá trị là 1.700.000. Chúng ta có thể thực hiện việc thay đổi giá trị rồi gán lại cho biến với bất kỳ phép toán số học nào:

Luong = Luong * 2;

Luong = Luong – 100.000;

...

Do việc tăng hay giảm giá trị của một biến rất thường xảy ra trong khi tính toán nên C# cung cấp các phép toán tự gán (self- assignment). Bảng sau liệt kê các phép toán tự gán.

Mô tả các phép toán tự gán

Toán tử	Ý nghĩa
+=	Cộng thêm giá trị toán hạng bên phảivào giá trị toán hạng bên trái
-=	Toán hạng bên trái được trừ bớt đi một lượng bằng giá trị của toán hạng bênphải
*=	Toán hạng bên trái được nhân với một lượng bằng giá trị của toán hạng bênphải.
/=	Toán hạng bên trái được chia với một lượng bằng giá trị của toán hạng bênphải.
%=	Toán hạng bên trái được chia lấy dư với một lượng bằng giá trị của toán hạngbên phải.

Dựa trên các phép toán tự gán trong bảng ta có thể thay thế các lệnh tăng giảm lương như sau:

Luong += 200.000; Luong *= 2;

Luong -= 100.000;

Kết quả của lệnh thứ nhất là giá trị của Luong sẽ tăng thêm 200.000, lệnh thứ hai sẽ làm cho giá trị Luong nhân đôi tức là tăng gấp 2 lần, và lệnh cuối cùng sẽ trừ bớt 100.000 của Luong.

Do việc tăng hay giảm 1 rất phổ biến trong lập trình nên C# cung cấp hai toán tử đặc biệt là tăng một (++) hay giảm một (--).

Khi đó muốn tăng đi một giá trị của biến đếm trong vòng lặp ta có thể viết như sau:

bienDem++;

Toán tử tăng giảm tiền tố và tăng giảm hậu tố

Giả sử muốn kết hợp các phép toán như gia tăng giá trị của một biến và gán giá trị của biến cho biến thứ hai, ta viết như sau:

var1 = var2++;

Câu hỏi được đặt ra là gán giá trị trước khi cộng hay gán giá trị sau khi đã cộng. Hay nói cách khác giá trị ban đầu của biến var2 là 10, sau khi thực hiện ta muốn giá trị của var1 là 10, var2 là 11, hay var1 là 11, var2 cũng 11?

Để giải quyết yêu cầu trên C# cung cấp thứ tự thực hiện phép toán tăng/giảm với phép toán gán, thứ tự này được gọi là tiền tố (prefix) hay hậu tố (postfix). Do đó ta có thể viết:

var1 = var2++; // Hậu tố

Khi lệnh này được thực hiện thì phép gán sẽ được thực hiện trước tiên, sau đó mới đến phép toán tăng. Kết quả là var1 = 10 và var2 = 11. Còn đối với trường hợp tiền tố:

var1 = ++var2;

Khi đó phép tăng sẽ được thực hiện trước tức là giá trị của biến var2 sẽ là 11 và cuối cùng phép gán được thực hiện. Kết quả cả hai biến var1 và var2 điều có giá trị là 11.

Để hiểu rõ hơn về hai phép toán này chúng ta sẽ xem ví dụ minh họa sau

Minh hoạ sử dụng toán tử tăng trước và tăng sau khi gán.

```
using System;
    class Tester
    {
    static int Main()
    {
    int valueOne = 10;
    int valueTwo;
    valueTwo = valueOne++;
    Console.WriteLine("Thuc hien tang sau: {0}, {1}", valueOne, valueTwo);
    valueOne = 20;
    valueTwo = ++valueOne;
```

```
    Console.WriteLine("Thuc hien tang truoc: {0}, {1}", valueOne, valueTwo);
    return 0;
    }
    }
```

Kết quả:

Thuc hien tang sau: 11, 10

Thuc hien tang truoc: 21, 21

Toán tử quan hệ

Những toán tử quan hệ được dùng để so sánh giữa hai giá trị, và sau đó trả về kết quả là một giá trị logic kiểu bool (true hay false). Ví dụ toán tử so sánh lớn hơn (>) trả về giá trị là true nếu giá trị bên trái của toán tử lớn hơn giá trị bên phải của toán tử. Do vậy 5 > 2 trả về một giá trị là true, trong khi 2 > 5 trả về giá trị false.

Các toán tử quan hệ trong ngôn ngữ C# được trình bày ở bảng 3.4 bên dưới. Các toán tử trong bảng được minh họa với hai biến là value1 và value2, trong đó value1 có giá trị là 100 và value2 có giá trị là 50.

Các toán tử so sánh (giả sử value1 = 100, và value2 = 50)

Tên toán tử	Kí hiệu	Biểu thức so sánh	Kết quả so sánh
So sánh bằng	==	value1 == 100value1 == 50	truefalse
Không bằng	!=	value2 != 100value2 != 90	falsetrue
Lớn hơn	>	value1 > value2value2 > value1	truefalse
Lớn hơn hay bằng	>=	value2 >= 50	true
Nhỏ hơn	<	value1 < value2value2 < value1	falsetrue
Nhỏ hơn hay bằng	<=	value1 <= value2	false

Như trong bảng trên ta lưu ý toán tử so sánh bằng (==), toán tử này được ký hiệu bởi hai dấu bằng (=) liền nhau và cùng trên một hàng , không có bất kỳ khoảng trống nào xuất hiện giữa chúng. Trình biên dịch C# xem hai dấu này như một toán tử.

Toán tử logic

Trong câu lệnh if mà chúng ta đã tìm hiểu trong phần trước, thì khi điều kiện là true thì biểu thức bên trong if mới được thực hiện. Đôi khi chúng ta muốn kết hợp nhiều điều kiện với nhau như: bắt buộc cả hai hay nhiều điều kiện phải đúng hoặc chỉ cần một trong các điều kiện đúng là đủ hoặc không có điều kiện nào đúng...C# cung cấp một tập hợp các toán tử logic để phục vụ cho người lập trình.

Bảng dưới liệt kệ ba phép toán logic, bảng này cũng sử dụng hai biến minh họa là x, và y trong đó x có giá trị là 5 và y có giá trị là 7.

Các toán tử logic (giả sử x = 5, y = 7)

Tên toán tử	Ký hiệu	Biểu thức logic	Giá trị	Logic
and	&&	(x == 3) && (y ==7)	false	Cả hai điều kiệnphải đúng
or	||	(x == 3) || (y == 7)	true	Chỉ cần một điềukiện đúng
not	!	! (x == 3)	true	Biểu thức trongngoặc phải sai.

Toán tử and sẽ kiểm tra cả hai điều kiện. Trong bảng trên có minh họa biểu thức logic sử dụng toán tử and:

(x == 3) && (y == 7)

Toàn bộ biểu thức được xác định là sai vì có điều kiện (x == 3) là sai.

Với toán tử or, thì một hay cả hai điều kiện đúng thì đúng, biểu thức sẽ có giá trị là sai khi cả hai điều kiện sai. Do vậy ta xem biểu thức minh họa toán tử or:

(x == 3) || (y == 7)

Biểu thức này được xác định giá trị là đúng do có một điều kiện đúng là (y == 7) là đúng.

Đối với toán tử not, biểu thức sẽ có giá trị đúng khi điều kiện trong ngoặc là sai, và ngược lại, do đó biểu thức:

!(x == 3)

có giá trị là đúng vì điều kiện trong ngoặc tức là (x == 3) là sai.

Như chúng ta đã biết đối với phép toán logic and thì chỉ cần một điều kiện trong biểu thức sai là toàn bộ biểu thức là sai, do vậy thật là dư thừa khi kiểm tra các điều kiện còn lại một khi có một điều kiện đã sai. Giả sử ta có đoạn chương trình sau:

```
int x = 8;
if ((x == 5) && (y == 10))
```

Khi đó biểu thức if sẽ đúng khi cả hai biểu thức con là (x == 5) và (y == 10) đúng. Tuy nhiên khi xét biểu thức thứ nhất do giá trị x là 8 nên biểu thức (x == 5) là sai. Khi đó không cần thiết để xác định giá trị của biểu thức còn lại, tức là với bất kỳ giá trị nào của biểu thức (y== 10) thì toàn bộ biểu thức điều kiện if vẫn sai.

Tương tự với biểu thức logic or, khi xác định được một biểu thức con đúng thì không cần phải xác định các biểu thức con còn lại, vì toán tử logic or chỉ cần một điều kiện đúng là đủ:

```
int x =8;
if ( (x == 8) || (y == 10))
```

Khi kiểm tra biểu thức (x == 8) có giá trị là đúng, thì không cần phải xác định giá trị của biểu thức (y == 10) nữa.

Ngôn ngữ lập trình C# sử dụng logic như chúng ta đã thảo luận bên trên để loại bỏ các tính toán so sánh dư thừa và cũng không logic nữa!

Độ ưu tiên toán tử

Trình biên dịch phải xác định thứ tự thực hiện các toán tử trong trường hợp một biểu thức có nhiều phép toán, giả sử, có biểu thức sau:

```
var1 = 5+7*3;
```

Biểu thức trên có ba phép toán để thực hiện bao gồm (=, +,*). Ta thử xét các phép toán theo thứ tự từ trái sang phải, đầu tiên là gán giá trị 5 cho biến var1, sau đó cộng 7 vào 5 là 12 cuối cùng là nhân với 3, kết quả trả về là 36, điều này thật sự có vấn đề, không đúng với mục đích yêu cầu của chúng ta. Do vậy việc xây dựng một trình tự xử lý các toán tử là hết sức cần thiết. Các luật về độ ưu tiên xử lý sẽ bảo trình biên dịch biết được toán tử nào được thực hiện trước trong biểu thức.Tương tự như trong phép toán đại số thì phép nhân có độ ưu tiên thực hiện trước phép toán cộng, do vậy 5+7*3 cho kết quả là 26 đúng hơn kết quả 36. Và cả

hai phép toán cộng và phép toán nhân điều có độ ưu tiên cao hơn phép gán. Như vậy trình biên dịch sẽ thực hiện các phép toán rồi sau đó thực hiện phép gán ở bước cuối cùng. Kết quả đúng của

câu lệnh trên là biến var1 sẽ nhận giá trị là 26.

Trong ngôn ngữ C#, dấu ngoặc được sử dụng để thay đổi thứ tự xử lý, điều này cũng giống trong tính toán đại số. Khi đó muốn kết quả 36 cho biến var1 có thể viết:

var1 = (5+7) * 3;

Biểu thức trong ngoặc sẽ được xử lý trước và sau khi có kết quả là 12 thì phép nhân được thực hiện.

Bảng dưới liệt kê thứ tự độ ưu tiên các phép toán trong C#.

Thứ tự ưu tiên các toán tử

STT	Loại toán tử	Toán tử	Thứ tự
1	Phép toán cơ bản	(x) x.y f(x) a[x] x++ x—new typeofsizeof checked unchecked	Trái
2		+ - ! ~ ++x –x (T)x	Trái
3	Phép nhân	* / %	Trái
4	Phép cộng	+ -	Trái
5	Dịch bit	<< >>	Trái
6	Quan hệ	< > <= >= is	Trái
7	So sánh bằng	== !=	P

			hải
8	Phép toán logicAND	&	Trái
9	Phép toán logicXOR	^	Trái
10	Phép toán logic OR	|	Trái
11	Điều kiện AND	&&	Trái
12	Điều kiện OR	||	Trái
13	Điều kiện	?:	Phải
14	Phép gán	= *= /= %= += -= <<= >>= &=^= |=	Phải

Các phép toán được liệt kê cùng loại sẽ có thứ tự theo mục thứ thự của bảng: thứ tự trái tức là độ ưu tiên của các phép toán từ bên trái sang, thứ tự phải thì các phép toán có độ ưu tiên từ bên phải qua trái. Các toán tử khác loại thì có độ ưu tiên từ trên xuống dưới, do vậy các toán tử loại cơ bản sẽ có độ ưu tiên cao nhất và phép toán gán sẽ có độ ưu tiên thấp nhất trong các toán tử.

Toán tử ba ngôi

Hầu hết các toán tử đòi hỏi có một toán hạng như toán tử (++, --) hay hai toán hạng như (+,-,*,/,...). Tuy nhiên, C# còn cung cấp thêm một toán tử có ba toán hạng (?:). Toán tử này có cú pháp sử dụng như sau:

<Biểu thức điều kiện > ? <Biểu thức thứ 1> : <Biểu thức thứ 2>

Toán tử này sẽ xác định giá trị của một biểu thức điều kiện, và biểu thức điều kiện này phải trả về một giá trị kiểu bool. Khi điều kiện đúng thì <biểu thức thứ 1> sẽ được thực hiện, còn ngược lại điều kiện sai thì <biểu thức thứ 2> sẽ được thực

hiện. Có thể diễn giải theo ngôn ngữ tự nhiên thì toán tử này có ý nghĩa : "Nếu điều kiện đúng thì làm công việc thứ nhất, còn ngược lại điều kiện sai thì làm công việc thứ hai". Cách sử dụng toán tử ba ngôi này được minh họa trong ví dụ sau.

Sử dụng toán tử bao ngôi.

```
using System;
    class Tester
    {

    public static int Main()
    {
    int value1;
    int value2;
    int maxValue; value1 = 10; value2 = 20;
    maxValue = value1 > value2 ? value1 : value2;
    Console.WriteLine("Gia tri thu nhat {0}, gia tri thu hai {1}, gia tri lon
nhat  {2}",value1,value2, maxValue);
    return 0;
    }
    }
```

Kết quả:

Gia tri thu nhat 10, gia tri thu hai 20, gia tri lon nhat 20

Trong ví dụ minh họa trên toán tử ba ngôi được sử dụng để kiểm tra xem giá trị của value1 có lớn hơn giá trị của value2, nếu đúng thì trả về giá trị của value1, tức là gán giá trị value1 cho biến maxValue, còn ngược lại thì gán giá trị value2 cho biến maxValue.

11. An toàn kiểu

Khi chúng ta chuyển đổi một số kiểu dữ liệu sau:

- int to short – Có thể mất dữ liệu
- int to uint –Có thể mất dữ liệu
- uint to int –Có thể mất dữ liệu
- float to int – Sẽ mất hết tất cả đằng sau dấu chấm thập phân
- Any numeric type to char – Dữ liệu sẽ mất
- decimal to any numeric type – Chỉ phần thập phân bên trong cấu trúc khác so với kiểu integers và floating-point numbers

Ví dụ:

```
long val = 30000;
int i = (int)val;   // Giá trị hợp lệ. The maximum int is
2147483647
long val = 3000000000;
int i = (int)val;         // Giá trị không hợp lệ. The maximum
int is 2147483647
```

Có thể gây ra một số lỗi như trên, để khắc phục vấn đề trên chúng ta sử dụng quá trình chuyển đổi từ một kiểu giá trị sang kiểu tham khảo (Boxing) và ngược lại (Unboxing). Điều này rất hữu dụng trong C#.

ví dụ sau đây sử dụng Boxing:

```
using System;
class BoxingDemo
{
    public static void Main()
        {
            int i = 100;
            object o = i ;
                Console.WriteLine(" The object value = {0}" ,
i);
        }
}
```

Trên đây giá trị i kiểu int được chuyển thành một o kiểu object.

Đoạn code sau sử dụng Unboxing , o thuộc kiểu đối tượng object được chuyển thành j kiểu int.

```
int i = 20;
object o = i;     // Box the int
int j = (int)o;   // Unbox it back into an int
ví dụ sau dùng Boxing và Unboxing
long a = 333333423;
object b = (object)a;
int c = (int)b;
```

12. Kiểu liệt kê

Một Enumeration (liệt kê) là một tập hợp các hằng số nguyên được đặt tên. Một kiểu enum được khai báo bởi sử dụng từ khóa enum trong C#.

Các kiểu liệt kê trong C# là kiểu dữ liệu giá trị. Nói cách khác, kiểu liệt kê chứa các giá trị của nó và không thể kế thừa hoặc không thể truyền tính kế thừa.

Cú pháp chung để khai báo một Enumeration trong C# là:

```
enum <tên_enum>
{
   danh_sách_enum
};
```

Ví dụ sau minh họa cách sử dụng của biến enum trong C#:

```
using System;
namespace VietJackCsharp
{
    public class TestCsharp
    {
        enum Days { Sun, Mon, Tue, Wed, Thu, Fri, Sat };
        public static void Main(string[] args)
        {
            Console.WriteLine("Enum trong C#");
            Console.WriteLine("-----------------------\n");

            int dau_tuan = (int)Days.Mon;
            int cuoi_tuan = (int)Days.Fri;
            Console.WriteLine("Thu hai: {0}", dau_tuan);
            Console.WriteLine("Thu sau: {0}", cuoi_tuan);
            Console.ReadKey();

        }
    }
}
```

13. Các không gian tên

Namespace là cách mà .NET dùng để chống lại sự xung đột tên giữa các lớp. Chẳng hạn như trường hợp bạn có một lớp mô tả khách hàng gọi là lớp Customer, và sau đó một người khác cũng có một lớp giống như vậy.

Một namespace không chỉ là một nhóm các kiểu dữ liệu, mà nó làm cho tên của tất cả các kiểu dữ liệu trong cùng một không gian tên sẽ có tiếp đầu ngữ là tên của namespace đó. Nó cũng cho phép một không gian tên nằm trong một không

gian tên khác. Ví dụ, hầu hết các hỗ trợ chung của các thư viện lớp cơ sở .NET đều nằm trong một không gian tên gọi là System. Lớp cơ sở Array nằm trong không gian tên này có tên đầy đủ là System.Array.

.NET yêu cầu tất cả các kiểu đều phải được định nghĩa trong một không gian tên, ví dụ bạn có thể đặt lớp Customer của bạn trong một không gian tên gọi là YourCompanyName. Lớp này sẽ có tên đầy đủ là YourCompanyName.Customer.

Nếu một namespace không được khai báo rõ ràng, các kiểu sẽ được đặt vào một namespace toàn cục không tên.

Microsoft khuyên rằng các hỗ trợ của bạn nên đặt vào một namespace ít nhất là 2 cấp, cấp một là tên của công ty của bạn, cấp hai là tên của công nghệ hoặc là phần mềm của gói sản phẩm đó, chẳng hạn như YourCompanyName.SalesServices.Customer. Làm như vậy trong hầu hết các trường hợp đảm bảo rằng, các lớp trong ứng dụng của bạn không xung đột tên với các lớp của các tổ chức khác.

14. Sử dụng các chú thích

Có hai loại "chú thích" (comment) sử dụng trong .NET framework: loại chú thích dòng và chú thích XML.

- Loại thông thường nhất là các loại "chú thích dòng" (Bất cứ khi nào bạn sử dụng // với C# hoặc VB.NET để đánh dấu một dòng là chú thích.) Loại chú thích này thường được sử dụng để giải thích những gì mà dòng lệnh hiện hành hoặc dòng mã kế tiếp đang thực hiện.

- Chú thích XML ghi chép tài liệu cho một lớp hoặc phương thức bằng cách sử dụng một phần XML. Ngoài ra, nó còn tạo tài liệu API cho ứng dụng.

Sau đây là một ví dụ về các chú thích XML được áp dụng và một phương thức C#. Vì nó xuất hiện theo thẻ mặc định, nên có thể nó là thẻ quen nhất trong các loại thẻ. Nó nên được sử dụng để ghi chép mục đích chung nhất của tất cả các phương thức public, thuộc tính và trường của một kiểu.

15. Từ định danh và từ khoá

Trong phần cuối của chương cơ bản C# chúng ta khảo sát các quy tắc để đặt tên cho các biến, lớp, các phương thức...

Từ định danh là tên chúng ta đặt cho biến, để định nghĩa kiểu sử dụng như các lớp, cấu trúc, và các thành phần của kiểu này. C# có một số quy tắc để định rõ các từ định danh như sau:

Chúng phải bắt đầu bằng ký tự không bị gạch dưới

Chúng ta không được sử dụng từ khoá làm từ định danh

Trong C# có sẵn một số từ khoá (keyword).

abstract	do	implicit	params	switch
as	double	in	private	this
base	else	int	protected	throw
bool	enum	interface	public	true
break	event	internal	readonly	try
byte	explicit	is	ref	typeof
case	extern	lock	return	uint
catch	false	long	sbyte	ulong
char	finally	namespace	sealed	unchecked
checked	fixed	new	short	unsafe
class	float	null	sizeof	ushort
const	for	object	stackalloc	using
continue	foreach	operator	static	virtual
decimal	goto	out	string	volatile

default	if	override	struct	void
delegate				while

BÀI 3: NHỮNG TÍNH NĂNG TIÊN TIẾN TRONG ASP.NET

Mã Bài: MĐ19_03

Giới thiệu:

CSS (Cascading Style Sheets) là một ngôn ngữ dùng để định dạng cho các phần tử HTML (Ví dụ: chỉnh kích cỡ chữ, chỉnh font chữ, màu chữ, màu nền, hình nền, đường viền,....) dựa trên các cặp thuộc tính:giá trị thuộc tính.

Với việc sử dụng CSS, ta có thể định dạng ra các phần tử HTML thật đặc biệt và chuyên nghiệp.

Mục tiêu:

- Bắt được các lỗi do người sử dụng gây ra.
- Quản lý được bộ nhớ.

Nội dung chính:

1. Lỗi và xử lí biệt lệ

Một Exception (ngoại lệ) là một vấn đề xuất hiện trong khi thực thi một chương trình. Một Exception trong C# là một phản hồi về một tình huống ngoại lệ mà xuất hiện trong khi một chương trình đang chạy, ví dụ như chia cho số 0.

Exception cung cấp một cách để truyền điều khiển từ một phần của một chương trình tới phần khác. Exception Handling (Xử lý ngoại lệ) trong C# được xây dựng dựa trên 4 từ khóa là: **try**, **catch**, **finally**, và **throw**.

try: Một khối try nhận diện một khối code mà ở đó các exception cụ thể được kích hoạt. Nó được theo sau bởi một hoặc nhiều khối catch.

catch: Một chương trình bắt một Exception với một Exception Handler tại vị trí trong một chương trình nơi bạn muốn xử lý vấn đề đó. Từ khóa **catch** trong C# chỉ dẫn việc bắt một exception.

finally: Một khối finally được sử dụng để thực thi một tập hợp lệnh đã cho, dù có hay không một exception đươc ném hoặc không được ném. Ví dụ, nếu bạn mở một file, nó phải được đóng, nếu không sẽ có một exception được tạo ra.

throw: Một chương trình ném một exception khi có một vấn đề xuất hiện. Điều này được thực hiện bởi sử dụng từ khóa **throw** trong C#.

Cú pháp

Giả sử một khối tạo một Exeption, một phương thức bắt một exception bởi sử dụng kết hợp các từ khóa try và catch. Một khối try/catch được đặt xung quanh code mà có thể tạo một exception. Code bên trong một khối try/catch được xem như là code được bảo vệ, và cú pháp để sử dụng try/catch trong C# như sau:

```
try
{
   // các lệnh có thể gây ra ngoại lệ (exception)
}
catch( tên_ngoại_lệ e1 )
{
   // phần code để xử lý lỗi
}
catch( tên_ngoại_lệ e2 )
{
   // phần code để xử lý lỗi
}
catch( tên_ngoại_lệ eN )
{
   // phần code để xử lý lỗi
}
finally
{
   // các lệnh được thực thi
}
```

Bạn có thể liệt kê nhiều lệnh catch để bắt các kiểu exception khác nhau trong trường hợp khối try của bạn xuất hiện nhiều hơn một exception trong các tình huống khác nhau.

2. Các sự kiện

Sự kiện (Event) là các hành động của người dùng, ví dụ như nhấn phím, click, di chuyển chuột, … Các Application cần phản hồi các sự kiện này khi chúng xuất hiện. Ví dụ, các ngắt (interrupt). Các sự kiện (Event) được sử dụng để giao tiếp bên trong tiến trình.

Sử dụng Delegate với Event trong C#

Các Event được khai báo và được tạo trong một lớp và được liên kết với Event Handler bởi sử dụng các Delegate bên trong cùng lớp đó hoặc một số lớp khác. Lớp mà chứa Event được sử dụng để công bố event đó. Điều này được gọi là lớp Publisher. Một số lớp khác mà chấp nhận Event này được gọi là lớp Subscriber. Các Event trong C# sử dụng mô hình Publisher-Subscriber.

Một **Publisher** trong C# là một đối tượng mà chứa định nghĩa của event và delegate đó. Mối liên hệ event-delegate cũng được định nghĩa trong đối tượng này. Một đối tượng lớp Publisher triệu hồi Event và nó được thông báo tới các đối tượng khác.

Một **Subscriber** trong C# là một đối tượng mà chấp nhận event và cung cấp một Event Handler. Delegate trong lớp Publisher triệu hồi phương thức (Event Handler) của lớp Subscriber.

Khai báo Event trong C#

Để khai báo một Event bên trong một lớp, đầu tiên một kiểu delegate cho Event đó phải được khai báo. Ví dụ:

```
public delegate void BoilerLogHandler(string status);
```

Tiếp theo, chính Event đó được khai báo, bởi sử dụng từ khóa event trong C#:

```
//định nghĩa event dựa vào delegate ở trên
public event BoilerLogHandler BoilerEventLog;
```

Code trên định nghĩa một delegate với tên là BoilerLogHandler và một Event với tên là BoilerEventLog, mà triệu hồi delegate đó khi nó được tạo ra.

Ví dụ 1

Tạo hai lớp có tên lần lượt là EventTest, TestCsharp như sau:

Lớp EventTest:

```
using System;
namespace VietJackCsharp
{
    class EventTest
    {
        private int value;
        public delegate void NumManipulationHandler();
        public event NumManipulationHandler ChangeNum;
        protected virtual void OnNumChanged()
        {
            if (ChangeNum != null)
            {
                ChangeNum();
            }
            else
```

```
            {
                Console.WriteLine("Kich hoat su kien!");
            }
        }

        public EventTest(int n)
        {
            SetValue(n);
        }
        public void SetValue(int n)
        {
            if (value != n)
            {
                value = n;
                OnNumChanged();
            }
        }
    }
}
```

3. Thuộc tính

Thuộc tính - Property là các thành viên được đặt tên của các lớp, cấu trúc, và Interface. Các biến thành viên hoặc các phương thức trong một lớp hoặc cấu trúc được gọi là các **Field**. Thuộc tính là một sự kế thừa của các Field và được truy cập bởi sử dụng cùng cú pháp. Chúng sử dụng **accessor** thông qua các giá trị của các Private Field có thể được đọc, được viết và được thao tác.

Thuộc tính (Property) không đặt tên các vị trí lưu giữ. Thay vào đó, chúng có accessors mà đọc, ghi hoặc tính toán các giá trị của chúng.

Ví dụ, chúng ta có một lớp với tên Student, với các Private Field cho age, name, và code. Chúng ta không thể trực tiếp truy cập các Field này từ bên ngoài phạm vi lớp đó, nhưng chúng ta có thể có các thuộc tính để truy cập các Private Field này.

Accessor trong C#

Trong C#, **accessor** là một thuộc tính chứa các lệnh có thể thực thi, mà giúp đỡ trong việc *lấy* (đọc hoặc tính toán) hoặc *thiết lập* (ghi) thuộc tính. Các khai báo accessor có thể thu được một get accessor, một set accessor, hoặc cả hai. Ví dụ:

```
// khai báo một thuộc tính Code có kiểu dữ liệu string:
public string Code
{
   get
   {
      return code;
   }
   set
   {
      code = value;
   }
}

// khai báo một thuộc tính Name có kiểu dữ liệu String:
public string Name
{
   get
   {
      return name;
   }
   set
   {
      name = value;
   }
}

// khai báo một thuộc tính Age có kiểu dữ liệu int:
public int Age
{
   get
   {
      return age;
   }
   set
   {
```

```
        age = value;
    }
}
```

Ví dụ dưới đây minh họa cách sử dụng của các thuộc tính trong C#: tạo 2 lớp có tên lần lượt là Student, TestCsharp như sau:

Lớp Student:

```
using System;
namespace VietJackCsharp
{
    class Student
    {
        private string code = "N/A";
        private string name = "unknown";
        private int age = 0;

        // khai bao thuoc tinh Code co kieu string:
        public string Code
        {
            get
            {
                return code;
            }
            set
            {
                code = value;
            }
        }

        // khai bao thuoc tinh Name co kieu string:
        public string Name
        {
            get
            {
                return name;
            }
```

```
            set
            {
                name = value;
            }
        }

        // khai bao thuoc tinh Age co kieu int:
        public int Age
        {
            get
            {
                return age;
            }
            set
            {
                age = value;
            }
        }
        public override string ToString()
        {
            return "MSSV = " + Code + ", Ho Ten = " + Name + ", Tuoi = " + Age;
        }
    }
}
```

4. Quản lý bộ nhớ bên dưới

Một trong những ưu điểm của C# là ta không cần quan tâm về việc quản lí bộ nhớ bên dưới vì điều này đã được bộ gom rác (garbage collector) của C# làm rồi . Mặc dù vậy nếu ta muốn viết các đoạn mã tốt, có hiệu suất cao , ta cần tìm hiểu về cách quản lí bộ nhớ bên dưới. Bài viết này sẽ trình bày về điều gì xảy ra trong bộ nhớ khi ta dùng các biến. Để chúng ta thực sự hiểu được cơ chế quản lý bộ nhớ bên dưới của C#.

4.1. Giá trị các kiểu dữ liệu

– Chúng ta sẽ tìm hiểu những gì xảy ra khi ta tạo ra một biến của kiểu giá trị. Kiểm tra điều gì xảy ra trong bộ nhớ khi thực hiện các dòng lệnh :

```
{
```

```
        int nRacingCars = 10;
        double engineSize = 3000.0;
       // do calculations;
    }
```

– Window dùng hệ thống địa chỉ ảo (*virtual addressing*) ánh xạ từ địa chỉ bộ nhớ mà chương trình ta thấy đến vị trí thực sự trong bộ nhớ vật lý hoặc trên đĩa được quản lí phía sau windows. Kết quả là mỗi ứng dụng trên nền xử lí 32-bit thấy được 4GB bộ nhớ , không cần biết bộ nhớ vật lí thực sự có kích thước bao nhiêu (nền xử lí 64bit thì bộ nhớ này lớn hơn) 4GB bộ nhớ này được gọi là không gian địa chỉ ảo (*virtual address space*) hay bộ nhớ ảo (*virtual memory*) .để đơn giản ta gọi nó là bộ nhớ mỗi vùng nhớ từ 4GB này được đánh số từ 0 . Nếu ta muốn chỉ định 1 giá trị lưu trữ trên 1 phần cụ thể trong bộ nhớ , ta cần cung cấp số đại diện cho vùng nhớ này.trong ngôn ngữ cấp cao , như là C# ,VB,C++ ,Java ..1 trong những thứ mà trình biên dịch làm là chuyển đổi tên đọc được (ví dụ tên biến) thành địa chỉ vùng nhớ mà bộ xử lí hiểu. 4GB bộ nhớ này thực sự chứa tất cả các phần của chương trình bao gồm mã thực thi và nội dung của biến được dùng khi chương trình chạy. bất kì DLLs được gọi sẽ nằm trong cùng không gian địa chỉ này, mỗi mục của mã hoặc dữ liệu sẽ có vùng định nghĩa riêng đâu đó trong bộ nhớ là 1 vùng gọi là ***stack***(ngăn xếp) . stack là nơi giá trị kiểu dử liệu được lưu.khi ta gọi phương thức ***stack***cũng được dùng để sao chép các thông số được truyền.để hiểu cách ***stack*** làm việc.ta cần lưu ý về phạm vi của biến trong C#. ví dụ :

```
{
       int a;
       // do something
       {
          int b;
          // do something else
       }
}
```

– Đầu tiên a được khai báo, thì bên khối bên trong , b được khai báo. Sau đó khối bên trong bị ngắt và b ra ngoài phạm vi, a ra ngoài phạm vi , vì thế đời sống của b nằm trong đời sống của a .

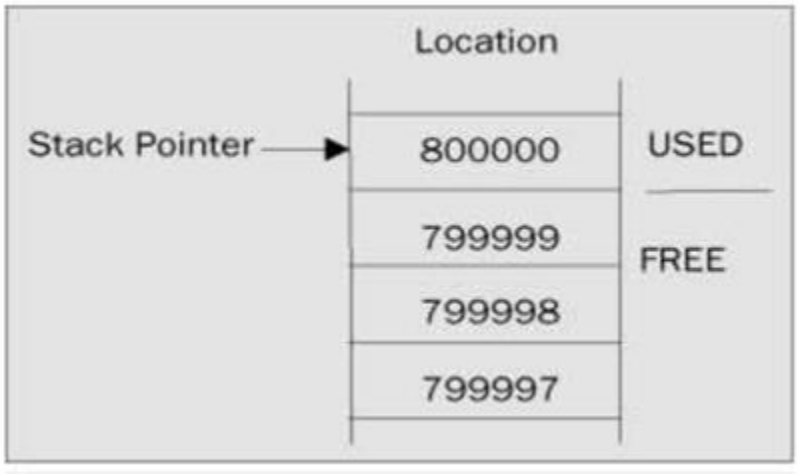
– Chúng ta sẽ xem điều gì xảy ra khi ta khai báo biến nRacingCars và engineSize từ ví dụ đầu đâu đó trong chương trình có một thứ gọi là **con trỏ stack (stack pointer)**. Đây đơn giản là 1 biến (và 1 địa chỉ trong bộ nhớ) mà chỉ

cho ta địa chỉ của vùng trống kế tiếp trong stack.khi chương trình chạy,con trỏ stack sẽ trỏ đến phần cuối của khối bộ nhớ mà được giữ trong stack.

– Và đây là một đoạn tiếng anh nói về các đặc tính của Stack:

"The stack actually fills downwards, from high memory addresses to low addresses. As data is put on the stack, the stack pointer will be adjusted accordingly, so it always points to just past the next free location. Now, we don't know exactly where in the address space the stack is – we don't need to know for C# development – but let's say for the sake of argument, that immediately before the above code that allocates the variables is executed, the stack pointer contains the value 800000 or, in hexadecimal, 0xC3500). We are taking this memory location just for the sake of argument, to keep the numbers simple. In fact, later on when we start running code that uses pointers, we will see that the stack actually starts round about memory location 1243328 (0x12F8C8). However, we can explain the principles of how the stack works just as well using any address, so we may as well pick a simple one. We will also mostly use decimal for addresses, again for simplicity, although it is more common to write memory addresses in hexadecimal format."

– Tình huống được minh hoạ trong biểu đồ, trong đó ,chữ in đậm chỉ định nội dung của vị trí bộ nhớ,chữ thường chỉ định địa chỉ hoặc 1 mô tả của vị trí



– Ở điểm này ,biến *nRacingCars* nằm trong phạm vi và giá trị 10 được đặt trong nó.nếu giá trị 10 được đặt ở vị trí 799996-799999, 4 byte bên dưới vị trí được trỏ đến bởi con trỏ ***stack*** -mà là 4 byte đầu tiên trống trên ***stack***.

– Dòng kế khai báo biến *engineSze*, 1 double và khởi tạo n1o với giá trị 3000.0. một số double chiếm 8 byte, vì thế giá trị 3000.0 sẽ được đặt ở vị trí 799988-799995 torng stack.con trỏ stack sẽ được tăng theo mỗi lần 8 để trỏ đến vùng trống dầu tiên trên stack .

4.2. Các kiểu dữ liệu tham chiếu

– Ta có thể muốn dùng 1 vài phương thức để cấp phát vùng nhớ để lưu trữ dữ liệu ,và giữ cho dữ liệu còn giá trị đến sau khi phương thức kết thúc.điều này có thể được làm với một toán tử mới – khi là trường hợp trong tất cả các kiểu tham chiếu.

– Nếu ta đã làm việc với các đọan mã quảnl ý bộ nhớ cấp thấp torng quá khứ,ta sẽ quen với stack và heap được dùng trong các chương trình trước .NET. heap quản lý (*managed heap*) không giống như heap mà các đoạn mã trước .NET như C++ dùng. nó làm việc dưới sự điều khiển của bộ thu gom rác. và có 1 số lợi ích so với hea truyền thống.

heap quản lý (ngắn gọn là heap) chỉ là một vùng nhớ khác torng vùng 4GB. Để xem làm thế nào heap làm việc và cách bộ nhớ được cấp phát cho các kiểu dữ liệu tham chiếu, ta xem đoạn mã sau:

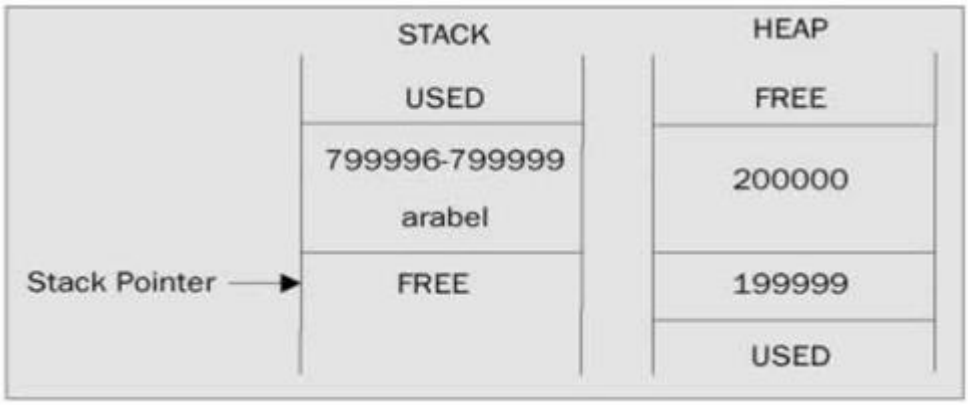
```
void DoWork()
{
      Customer arabel;
      arabel = new Customer();
      Customer mrJones = new Nevermore60Customer();
}
```

trong đoạn mã này ta giả sử gồm 2 lớp Customer và Nevermore60Customer.những lớp này được lấy từ ví dụ Mortimer Phones. Ta khai báo 1 tham chiếu gọi là arabel được cấp phát trong stack nhưng nên nhớ rằng đây là 1 tham chiếu , không phải là 1 thể hiện Customer. không gian mà tham chiếu arabel chiếm là 4 byte.tacần 4 byte để có thể lưu 1 số nguyên giá trị từ 0 đến 4GB sau đó ta có dòng :

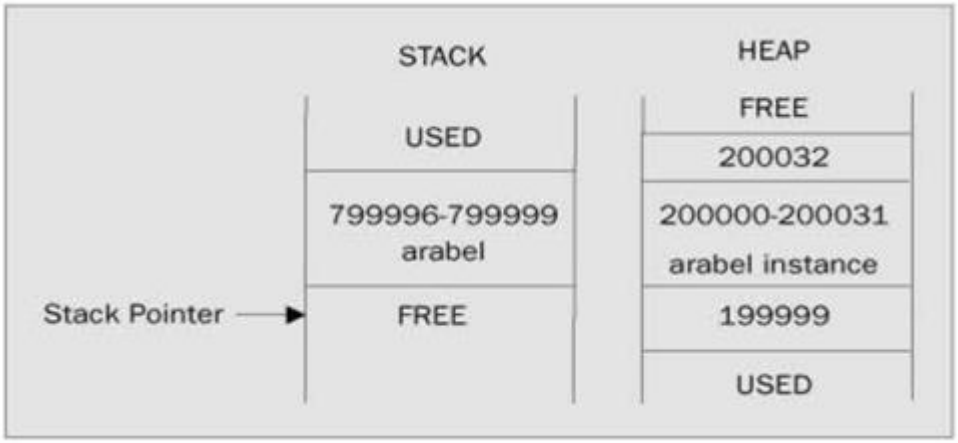
```
arabel = new Customer();
```

– Dòng này trong mã sẽ thực thi một số điều. Đầu tiên nó cấp phát vùng nhớ trong heap để lưu trữ 1 thể hiện Customer. sau đó nó đặt biến arabel để lưu địa chỉ củavùng nhớ được cấp phát.đồng thời nó cũng gọi hàm dựng Customer() để khởi tạo 1 thể hiện lớp. Thể hiện của customer không đặt trong stack- mà sẽ đặt trong heap, ta không biết chính xác 1 thể hiện Customer chiếm bao nhiêu byte, ta xem nó là 32 byte. 32 byte này chứa các trường thể hiện của Customer cùng vài thông tin mà .NET dùng để xác định danh tính và quản lí các thể hiện lớp của nó , bao gồm vtable.

.NET runtime tìm torng heap khối 32 byte còn trống, giả sử nằm ở địachỉ 200000,và tham chiếu arabel nằm ở vĩ trí 799996-799999 trên stack.



không như stack ,bộ nhớ trong heap được cấp phát theo chiều từ dưới lên , vì thế không gian trống được tìm thấy phía trên không gian đã dùng. Sau khi cấp phát đối tượng ,bộ nhớ trông như sau:



dòng kế tiếp thực hiện tươn gtự,ngoại trừ không gian trên stack ch otham chiếu mrJones cần được cấp phát vào cùng lúc cấp phát mrJones trên heap:

```
Customer mrJones = new Nevermore60Customer();
```

4 byte được cấp phát trên stack cho tham chiếu mrJones. lưu ở 799992-799995. trong khi thể hiện mrJones sẽ được cấp phát từ vị trí 200032 đi lên trên heap.

– .NET runtime sẽ cần duy trì thông tin về trạng thái của heap, thông tin này cũng cần được cập nhật khi dữ liệu mới được thêm vào heap.để minh họa điều này, ta hãy xem điều gì xày ra khi ta thoát phương thứ phương thức và tham chiếu arabel và mrJones nằm ngoài phạm vi theo cách làm việc bình thường thì con trỏ stack sẽ được tăng để những biến này không còn tồn tại nữa.tuy nhiên các biến này chỉ lưu địa chỉ, không phải 1 thể hiện lớp dữ liệu của nó vẫn nằm trong heap. Ta có thể thiết lập các biến tham chiếu khác nhau để trỏ đến cùng 1 đối tượng- nghĩa là những đối tượng đó sẽ có giá trị sau khi tham chiếu arabel và

mrJones nằm ngoài phạm vi, và sự khác biệt quan trọng giữa stack và heap: đối tượng được cấp phát liên tiếp trên heap, các thời gian sống không lồng nhau.

– Khi ta giải thích cách hoạt động trên heap, ta nhấn mạnh rằng chỉ stack mới có khả năng lồng thời gian sống của các biến.Vậy khi thời gian sống của các tham chiếu nằm ngoài phạm vi thì heap làm việc như thế nào trên các biến này câu trả lời là bô thu gom rác sẽ làm điều này khi bộ gom rác chạy. Nó sẽ gỡ bỏ tất cả những đối tượng từ heap mà không còn tham chiếu nữa. Ngay sau khi nó làm điều này, heap sẽ có các đối tượng rải rác trên nó, nằm lẫn với các khoảng trống như hình sau :

– Bộ gom rác không để heap trong tình trạng này, ngay khi nó giải phóng tất cả các đối tượng có thể,nó sẽ di chuyển tất cả chúng trở về cuối của heap để thành 1 khối liên tục lại. Tất nhiên khi những đối tượng này được di chuyển tất cả các tham chiếu của nó đều được cập nhật lại.

BÀI 4: TRUY CẬP VÀ XỬ LÝ CƠ SỞ DỮ LIỆU VỚI .NET

Mã Bài: MĐ19_04

Giới thiệu:

Khi phát triển các ứng dụng trên nền web thì công việc chủ yếu phải giải quyết là xử lý các nghiệp vụ, trong đó phần lớn là xử lý Cơ sở dữ liệu. Trong môi trường phát triển Microsoft .NET tất cả các ứng dụng webform hay winform đều thống nhất sử dụng chung một bộ thư viện để truy xuất và thao tác Cơ sở dữ liệu gọi là ADO.NET (Active Data Object).

ADO.NET là một tập các lớp nằm trong bộ thư viện lớp cơ sở của .NET Framework, cho phép các ứng dụng windows (như C#, VB.NET) hay ứng dụng web (như ASP.NET) thao tác dễ dàng với các nguồn dữ liệu.

Mục tiêu:

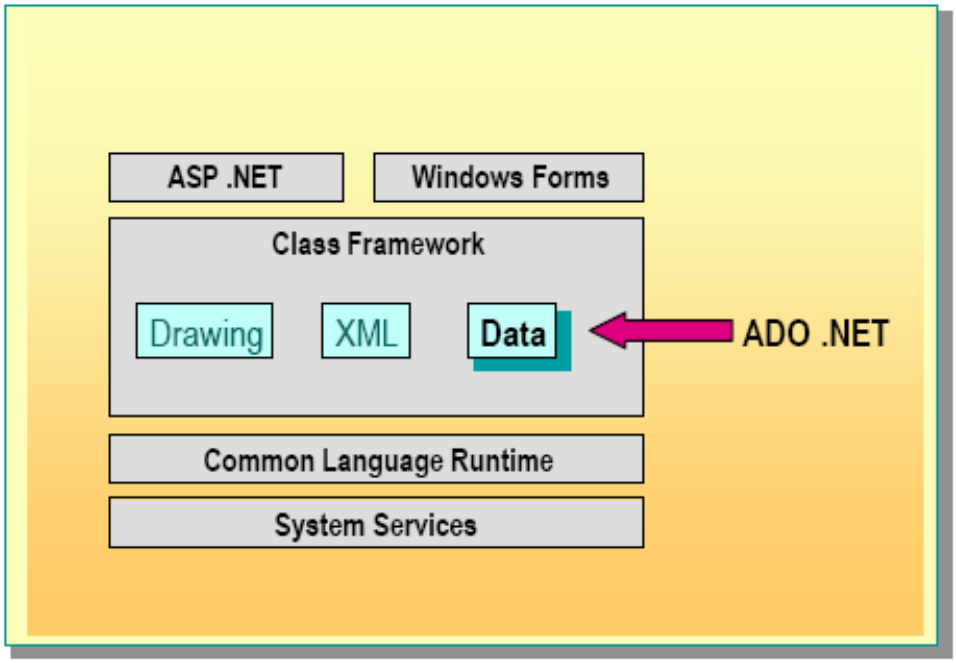
- Sử dụng các đối tượng ADO để thực hiện các thao tác tìm kiếm, thêm, sửa,
- xóa các bản ghi trong cơ sở dữ liệu

Nội dung chính:

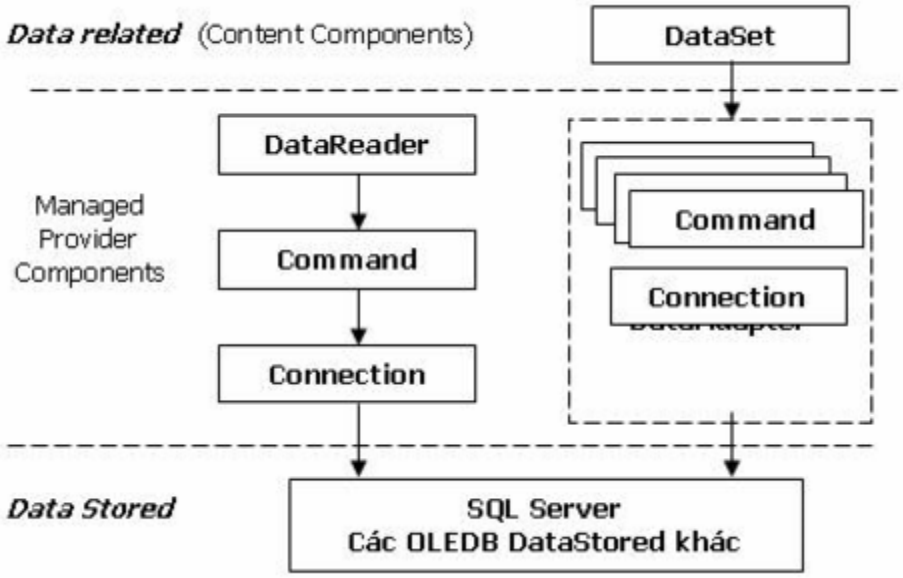
1. Tổng quan về ADO.NET

ADO.NET bao gồm 2 Provider (2 bộ thư viện thường dùng) để thao tác với các CSDL là: OLE DB Provider (nằm trong System.Data.OLEDB) dùng để truy xuất đến bất kỳ CSDL nào có hỗ trợ OLEDB; SQL Provider (nằm trong System.Data.SQLClient) chuyên dùng để truy xuất đến CSDL SQL Server (Không qua OLE DB nên nhanh hơn). Hiện nay, các hãng thứ ba còn cung cấp các Provider khác như : MySQL, Oracle… provider để cho phép ứng dụng .NET truy xuất đến các cơ sở dữ liệu không phải của Microsoft khác.

Vị trí của ADO.NET trong kiến trúc của .NET Framework



Từ kiến trúc ta thấy rằng: ADO.NET là một thành phần nội tại (Instrict) của .NET framework, do vậy nó có thể được sử dụng trong tất cả các ngôn ngữ hỗ trợ .NET như C#, VB.NET… mà không có sự khác biệt nào (Tức là các chức năng cũng như cách sử dụng hoàn toàn giống nhau).



2. Các công cụ kết nối dữ liệu

2.1. Đối tượng Connection

Kết nối cơ sở dữ liệu SQLServer

Bạn cần nhập khẩu lớp SqlClient

using System.Data.SqlClient;

Khai báo và khởi tạo:

```
SqlConnection sqlcon;
string driver="server=localhost; UID=sa; PWD=;
database=name_database";
      sqlcon=new Sqlconnection();
sqlcon.ConnectionString=driver;
Driver là chuỗi kết nối đến cơ sở dữ liệu trong trường hợp này
mình kết nối với sqlserver 2000/2005
Kết nối với cơ sở dữ liệu Access
Bạn cận nhập khẩu lớp OleDb
                using System.Data.OleDb;
        OleDbConnection oleconn;
        string driver = "Provider=Microsoft.jet.OLEDB.4.0; Data
Source=duongdan_tendata";
        oleconn = new OleDbConnection();
        oleconn.ConnectionString = driver;
```

Thuộc tính:

ConnectString: chứa đựng chuỗi kết nối tới cơ sở dữ liệu

Database: Chứa đựng tên cơ sở dữ liệu trong chuỗi kết nối ConnectString ở trên và bạn có thể thay đổi cơ sở dữ liệu trong lúc thực thi bằng phương thức ChangeDataBase:

Sqlconn.ChangeDatabase("name_database_thaydoi");

Server: tên máy chủ bạn trỏ tới

Connect Timeout: số thời gian(tính bằng giây) chờ kết nối dữ liệu mặc đình là 15giây, nếu trong khoảng thời gian này mà vẫn chưa kết nối xong một lỗi Connect Timeout được đưa ra.

State: trả về trạng thái của đối tượng SqlConnection: bạn có thể kiểm tra trạng thái của State như sau

Response.Write(sqlconn.State)

Các phương thức của đối tượng Connection

Open: cho phép mở dữ liệu với các thuộc tính đã khai báo trong ConnectString

Close: Đóng cơ sở dữ liệu đang mở

CreateCommand: phương thức cho phép gán hay trả về một đối tượng Command ứng với đối tượng Connection, như ví dụ HelloWorld

```
        SqlConnection sqlconn = new SqlConnection(driver);
```

```
        SqlCommand sqlcom = sqlconn.CreateCommand();
        sqlcom.CommandText = "select sTitle from tblHello where
pkHelloID=1";
```

BeginTransaction: Phương thức này khai báo bắt đầu một chuyển tác, để kết thúc chuyển tác bạn dùng Table Commit

Rollback: trong trương hợp có lỗi trong quá trình thực thi bạn có thể sử dụng phương thức Rollback để huỷ bỏ các chuyển tác đã thực hiện.

Dispose: dùng để huỷ bỏ hay giải phóng đối tượng Connection đang sử dụng

Đối tượng SqlCommand

Khai báo và khởi tạo đối tượng

Cách 1:

SqlCommand sqlcom;

sqlcom=new SqlCommand(ssql,sqlconn)

cách 2:

SqlCommand sqlcom = new SqlCommand();

sqlcom.Connection = sqlconn;

sqlcom.CommandType = CommandType.Text;

sqlcom.CommandText = "select sTitle from tblHello where pkHelloID=1";

Phương thức

ExcuteReader: dùng để thực thi đọc cơ sở dữ liệu từ bảng cơ sở dữ liệu

ExcuteNoneQuery: Dùng để thực thi các phát biểu T-Sql như: Insert, Update, Delete, Create,…

ExcuteScalar: trả về từ phát biết SQL dạng Select chỉ có một cột một hàng.

Đối tượng SqlDataReader

Đối tượng này được net cung cấp để đọc dữ liệu từ bảng cơ sở dữ liệu, nó là đối tượng chỉ phục vụ thao tác đọc dữ liệu(Read only). Trong khi truy xuất dữ liệu nó sẽ giữ kết nối liên tục với database(hướng kết nối)

Khai báo và khởi tạo đối tượng

SqlDataReader sqlreader;

sqlreader = sqlcom.ExecuteReader();

Đối tượng DataAdapter

OleDataAdapter được xem như bộ đọc dữ liệu từ cơ sở dliệu nguồn và điền chúng vào đối tượng DataSet hay DataTable

Khai báo, khởi tạo và giải phóng đối tượng.

```
string ssql;
```

Khai báo đối tượng

```
Dim sqlcom As SqlCommand
Dim sqlconn As SqlConnection
Dim sqladapter As SqlDataAdapter
sqlconn.Open();
```

cách 1.

```
sqladapter = New OleDbDataAdapter(ssql, sqlconn)
sqlcom = New SqlCommand(ssql, sqlconn)
```

cách 2.

```
sqladapter = new SqlDataAdapter(sqlcom);
```

Giải phóng đối tượng

```
sqladapter.Dispose();
```

Thuộc tính:

Các thuộc tính bao gồm SelectCommand, InsertCommand, UpdateCommand, DeleteCommand: thực hiện các thao tác select, insert, update, delete dữ liệu

Phương thức:

Fill: Phương thức thực thi câu lệnh select trong sql rồi điền kết quả cho DataSet hoặc Datatable.

Update: gọi lệnh cập nhật các thay đổi vào dữ liệu lên các dữ liệu nguồn

Điền dữ liệu từ Adapter vào DataSet

Dataset là một thùng chứa dữ liệu không kết nối

```
public static DataSet Filldataset(string ssql)
    {
        DataSet dataset = new DataSet();
        opendata();
        try
        {
```

```
            sqladapter = new SqlDataAdapter(ssql, sqlconn);
            sqladapter.Fill(dataset);
            sqladapter.Dispose();
        }
        catch (Exception exp)
        {
            closedata();
            System.Web.HttpContext.Current.Response.Write(exp.ToString());
        }
        closedata();
        return dataset;
    }
```

Điền dữ liệu vào DataTable

```
  public static DataTable FillDatatable(string ssql)
    {
        opendata();
        DataTable datatable = new DataTable();
        try
        {
            sqladapter = new SqlDataAdapter(ssql, sqlconn);
            sqladapter.Fill(datatable);
            sqladapter.Dispose();
        }
        finally
        {
            closedata();
        }
        closedata();
        return datatable;
    }
```

3. Hiển thị dữ liệu

Khi bạn có nhu cầu hiển thị dữ liệu dưới dạng bảng gồm nhiều dòng dữ liệu thì GridView chính là lựa chọn phù hợp nhất. Control này cho phép hiển thị nhiều dòng dữ liệu, hỗ trợ phân trang và cho phép định dạng theo ý muốn nội dung trong từng cột.

Trước hết bạn phải hiểu rõ khi nào thì sử dụng GridView. Đó là khi bạn muốn dữ liệu của mình (thuộc dạng danh sách hoặc tập hợp) hiển thị nhiều dòng trong một bảng với tiêu đề bên trên. Ví dụ như trang web sau sử dụng GridView để hiển thị danh sách khách hàng

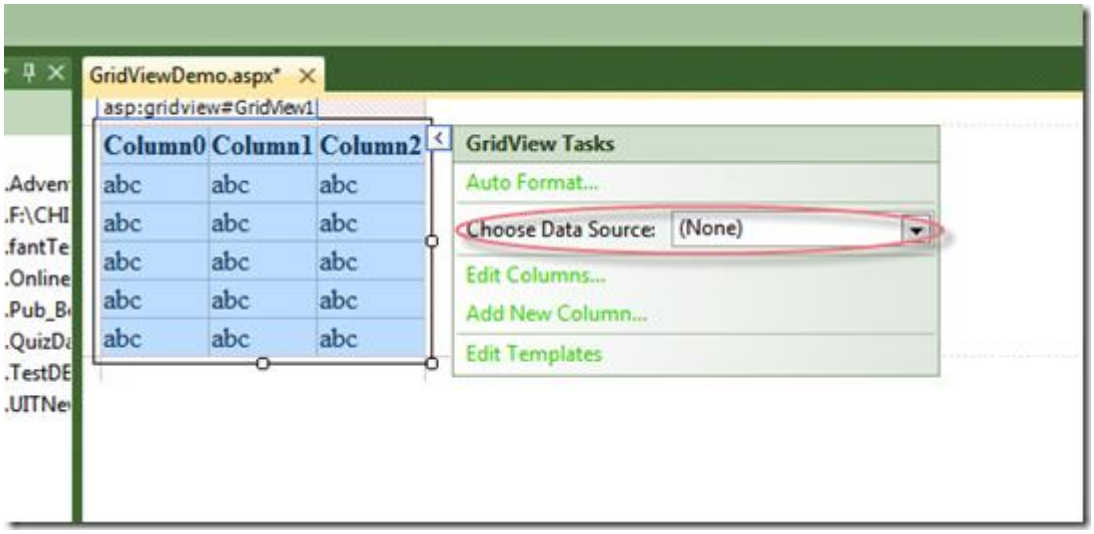
Customer View By Country

Customer: Venezuela Details

Customer	Contact	Title	Address	City	Phone
GROSELLA-Restaurante	Manuel Pereira	Owner	5ª Ave. Los Palos Grandes	Caracas	(2) 283-2951
HILARION-Abastos	Carlos Hernández	Sales Representative	Carrera 22 con Ave. Carlos Soublette #8-35	San Cristóbal	(5) 555-1340
LILA-Supermercado	Carlos González	Accounting Manager	Carrera 52 con Ave. Bolívar #65-98 Llano Largo	Barquisimeto	(9) 331-6954
LINO-Delicateses	Felipe Izquierdo	Owner	Ave. 5 de Mayo Porlamar	I. de Margarita	(8) 34-56-12

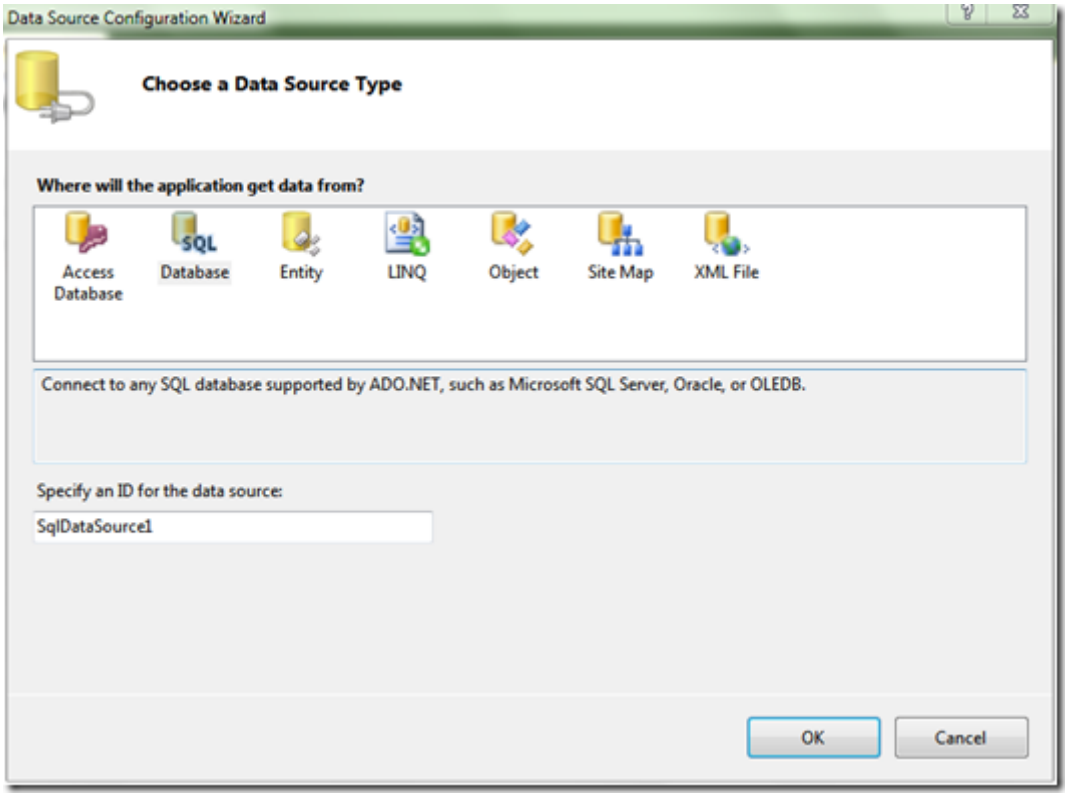
Để minh họa việc sử dụng GridView, chúng ta sẽ cùng tạo một ứng dụng ASP.NET đơn giản. Cơ sở dữ liệu được sử dụng sẽ là AdventureWorks. Nếu bạn chưa có CSDL này thì có thể tải về tại địa chỉ sau: http://msftdbprodsamples.codeplex.com/

Bây giờ các bạn hãy tạo một trang web ASPX trống hoàn toàn, sau đó kéo thả control GridView trong Toolbox vào giao diện của trang web

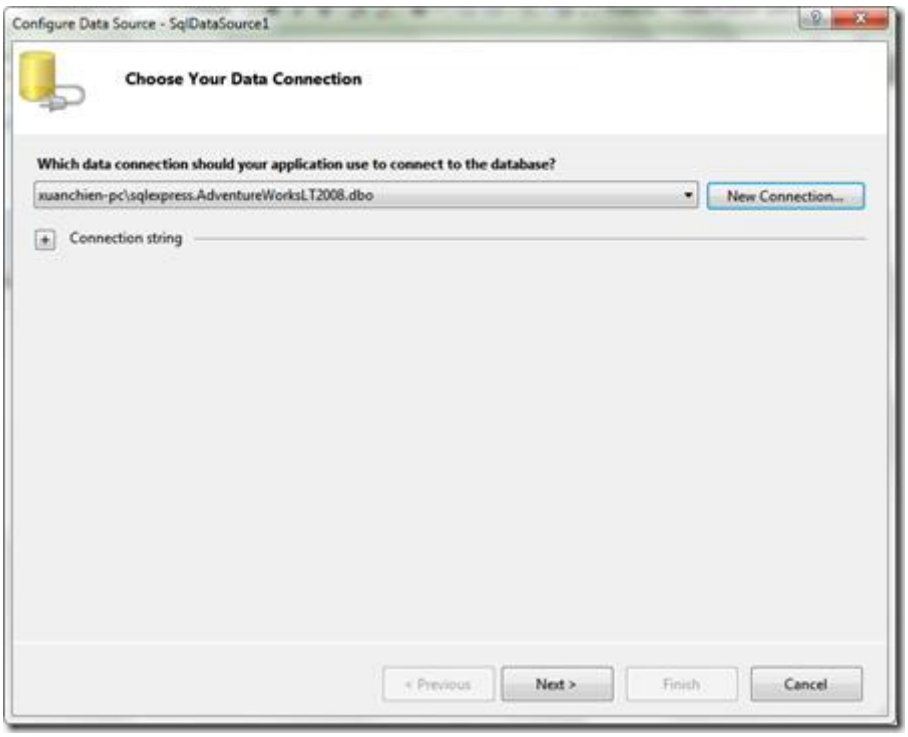


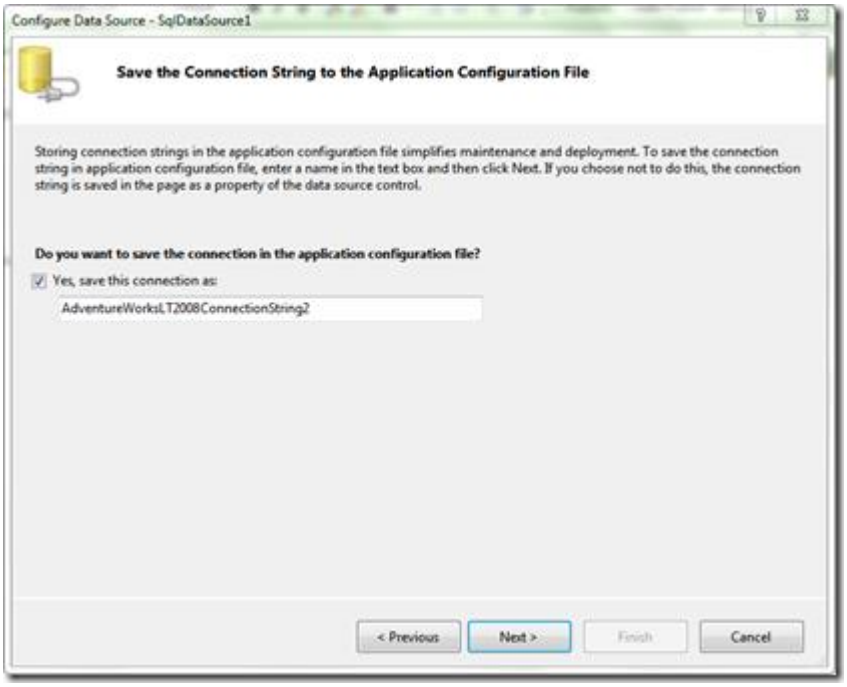
Nhấn vào mũi tên nhỏ góc trên bên phải của GridView các bạn sẽ có khung nhỏ cho phép chọn nguồn dữ liệu của GridView. Vì hiện tại chúng ta chưa tạo bất

cứ các Control để kết nối dữ liệu (như ObjectDataSource, SqlDataSource..) cho nên trong danh sách "Choose Data Source" các bạn chọn "New Data Source"

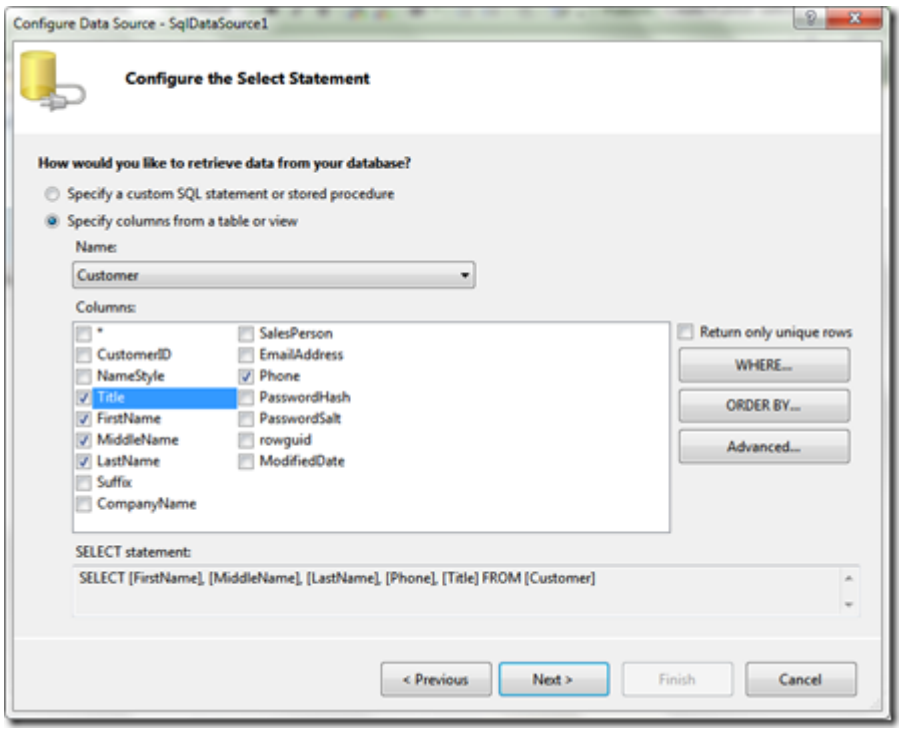


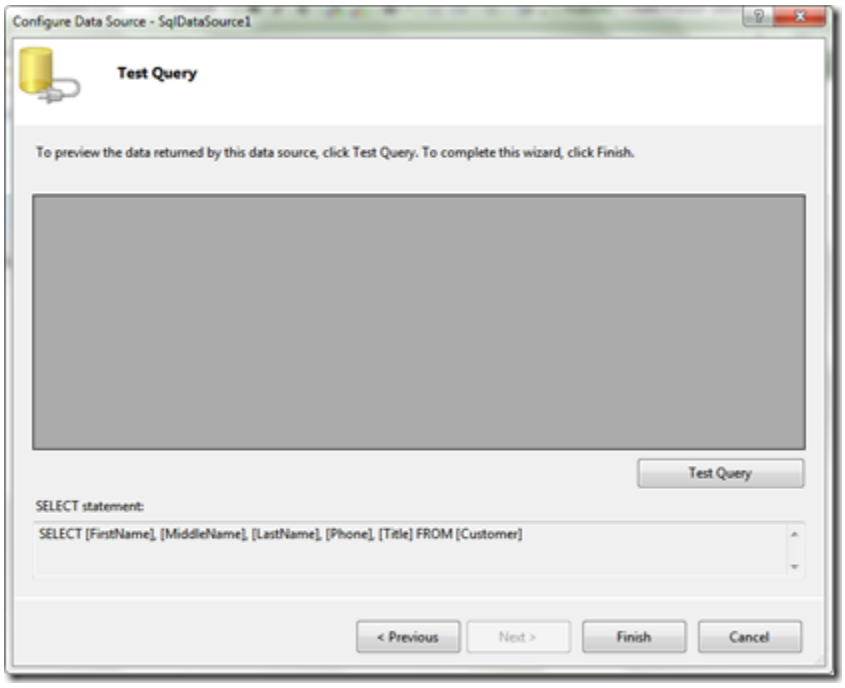
Ở đây, để đơn giản minh họa cho nhân vật chính là "GridView" nên tạm thời chúng ta không cần phải làm phức tạp thêm các nhân vật phụ, các bạn chọn SqlDataSource, sau đó nhấn OK. Hộp thoại tiếp theo yêu cầu bạn chỉ định kết nối đến cơ sở dữ liệu. Như đã nói ở đầu thì chúng ta sẽ sử dụng Cơ sở dữ liệu AdventureWorks. Nếu trong danh sách kết nối mà vẫn chưa có kết nối đến CSDL này thì các bạn phải tạo mới bằng cách nhấn vào nút New Connection (chi tiết của các thao tác này có thể xem thêm tại bài viết giới thiệu về ADO.NET và sau đó là bài về SqlDataSource).



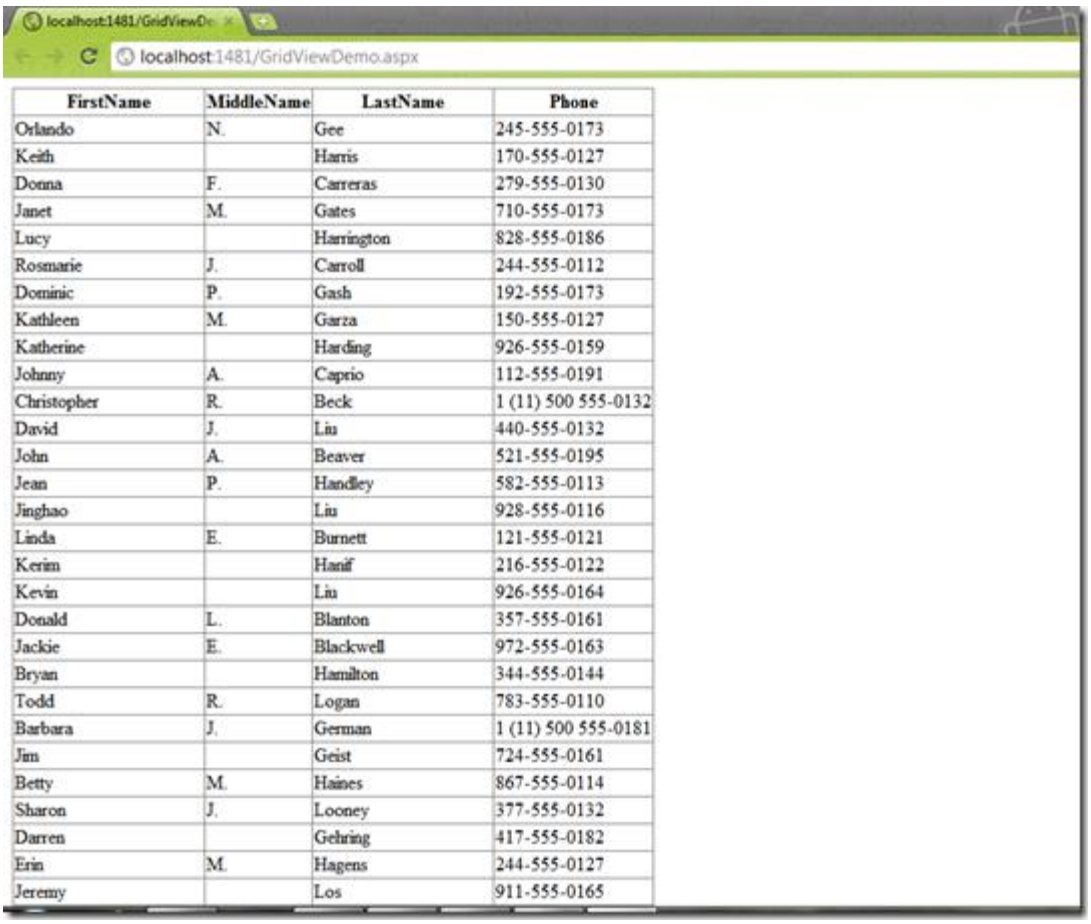


Ở phần chọn bảng và các cột cần hiển thị thì chúng ta sẽ chọn bảng Customer và chỉ hiển thị một số cột trong bảng này.





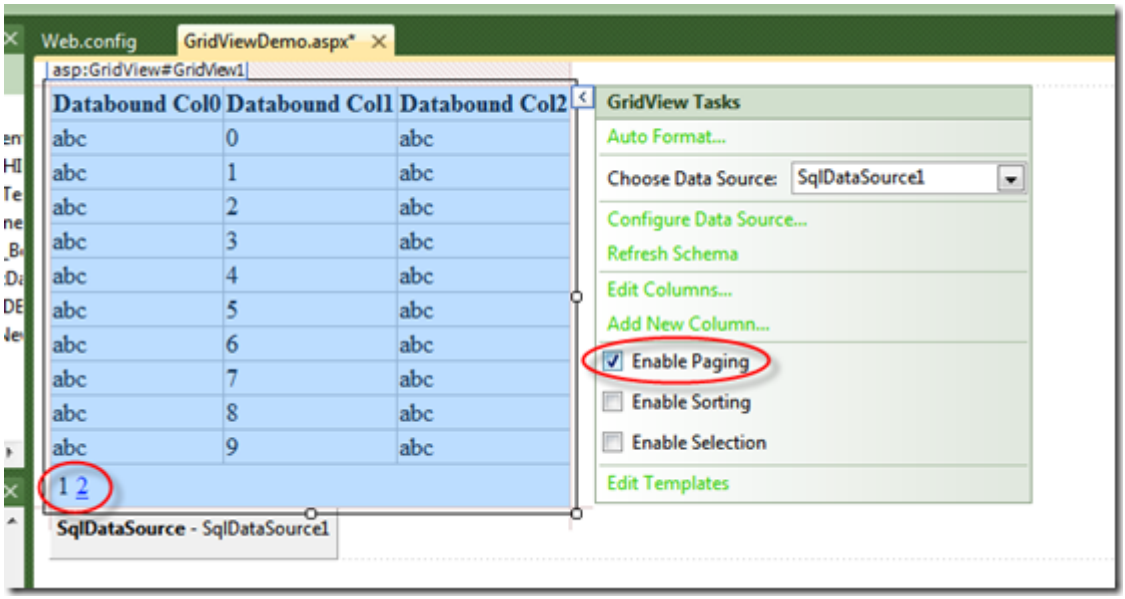
Chạy thử trang web vừa tạo, bạn sẽ có được một bảng dữ liệu gồm tất cả dữ liệu trong database hiện có của bảng Customer.



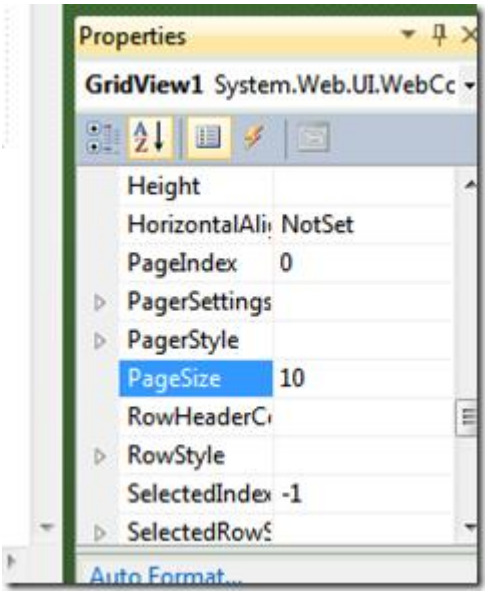
FirstName	MiddleName	LastName	Phone
Orlando	N.	Gee	245-555-0173
Keith		Harris	170-555-0127
Donna	F.	Carreras	279-555-0130
Janet	M.	Gates	710-555-0173
Lucy		Harrington	828-555-0186
Rosmarie	J.	Carroll	244-555-0112
Dominic	P.	Gash	192-555-0173
Kathleen	M.	Garza	150-555-0127
Katherine		Harding	926-555-0159
Johnny	A.	Caprio	112-555-0191
Christopher	R.	Beck	1 (11) 500 555-0132
David	J.	Liu	440-555-0132
John	A.	Beaver	521-555-0195
Jean	P.	Handley	582-555-0113
Jinghao		Liu	928-555-0116
Linda	E.	Burnett	121-555-0121
Kerim		Hanif	216-555-0122
Kevin		Liu	926-555-0164
Donald	L.	Blanton	357-555-0161
Jackie	E.	Blackwell	972-555-0163
Bryan		Hamilton	344-555-0144
Todd	R.	Logan	783-555-0110
Barbara	J.	German	1 (11) 500 555-0181
Jim		Geist	724-555-0161
Betty	M.	Haines	867-555-0114
Sharon	J.	Looney	377-555-0132
Darren		Gehring	417-555-0182
Erin	M.	Hagens	244-555-0127
Jeremy		Los	911-555-0165

Phân trang

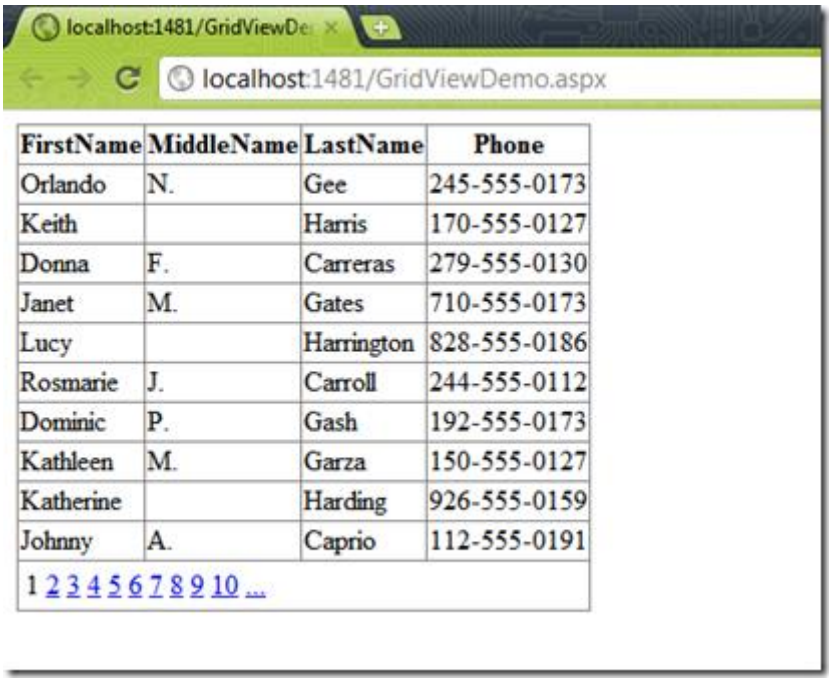
GridView hỗ trợ sẵn cho các bạn phân trang dữ liệu được hiển thị nhằm tránh tình trạng số dòng hiển thị quá lớn. Để bật tính năng phân trang của GridView, các bạn nhấp mũi tên góc bên phải của GridView và chọn Enable Paging. Lúc này theo mặc định, GridView sẽ hiển thị tối đa 10 dòng cùng lúc và phần footer của GridView sẽ có thêm các liên kết đến những trang tiếp theo của dữ liệu.



Nếu như bạn muốn số lượng dòng hiển thị ở mỗi trang nhiều hơn hoặc ít hơn thì bạn có thể chỉnh thuộc tính PageSize trong khung Properties của GridView



Thử chạy lại trang web này, bạn sẽ thấy danh sách hiển thị đã được phân chia thành nhiều trang khác nhau với số trang được hiển thị thành những liên kết ở phía dưới cùng.

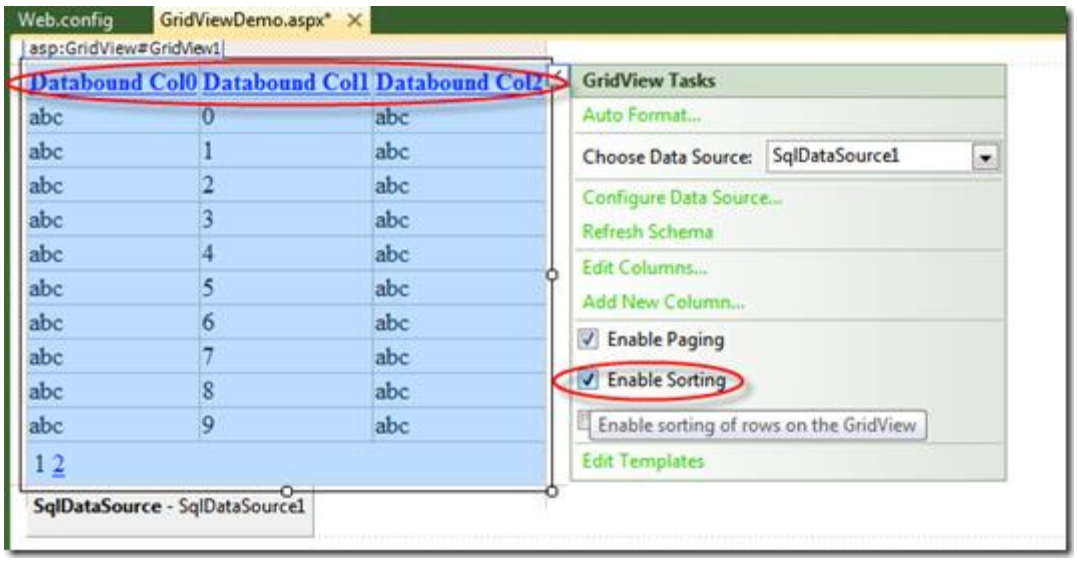


FirstName	MiddleName	LastName	Phone
Orlando	N.	Gee	245-555-0173
Keith		Harris	170-555-0127
Donna	F.	Carreras	279-555-0130
Janet	M.	Gates	710-555-0173
Lucy		Harrington	828-555-0186
Rosmarie	J.	Carroll	244-555-0112
Dominic	P.	Gash	192-555-0173
Kathleen	M.	Garza	150-555-0127
Katherine		Harding	926-555-0159
Johnny	A.	Caprio	112-555-0191

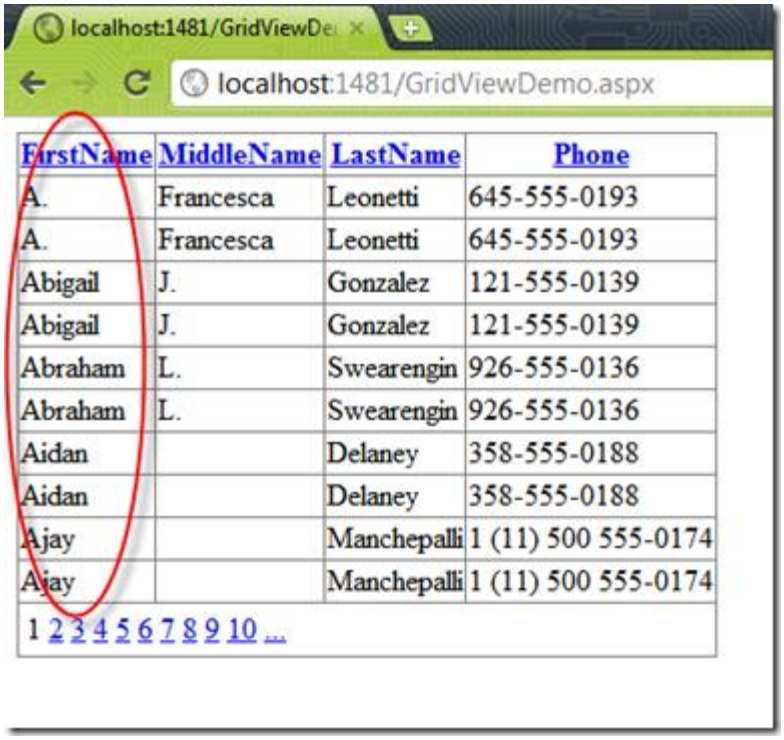
1 2 3 4 5 6 7 8 9 10 ...

Sắp xếp

Tính năng này cho phép bạn sắp xếp theo thứ tự các cột. Khi tính năng này được bật, tiêu đề của mỗi cột sẽ có dạng Hyperlink, nếu như bạn nhấn vào tiêu đề của cột nào thì cột đó sẽ được sắp xếp. Bạn cũng bật tính năng này bằng cách chọn Enable Sorting trong khung cửa sổ nhỏ khi nhấn vào mũi tên ở góc trên bên phải của GridView



Kết quả xuất ra là một bảng cho phép bạn sắp xếp nội dung của từng cột. Nếu nhấn lần thứ nhất thì nội dung được sắp xếp tăng dần, nhấn lần thứ hai thì sắp xếp giảm dần

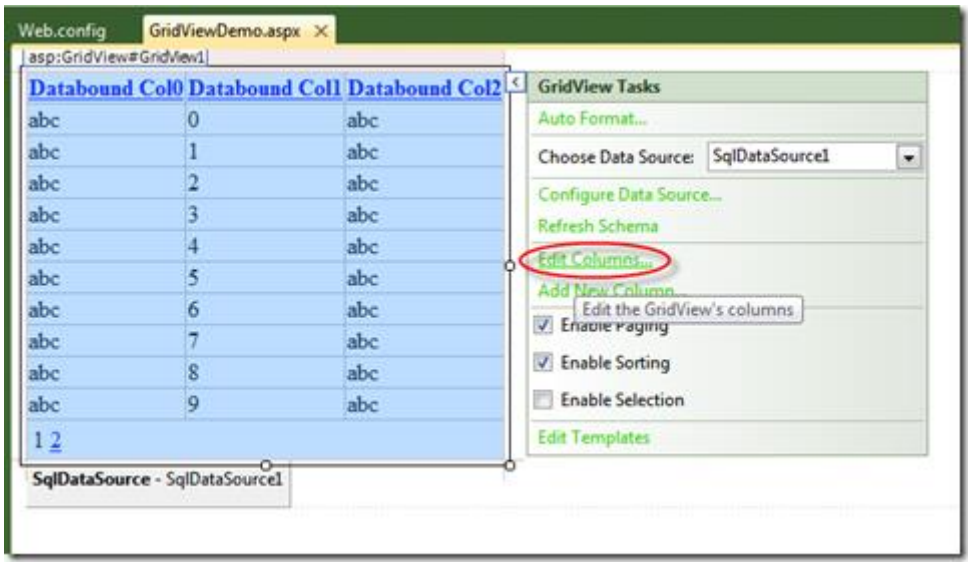


Tạo cột tự định nghĩa

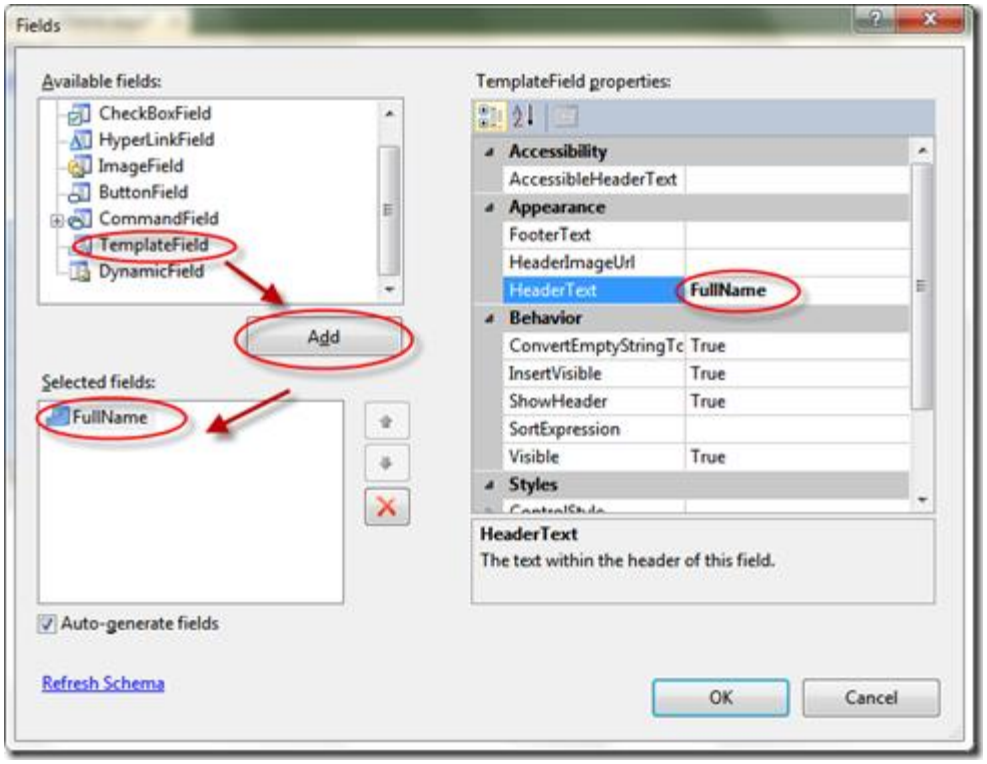
Khi dữ liệu được lấy từ database được đưa lên GridView thì mặc định GridView sẽ sử dụng chủ yếu là các control Label cho việc hiển thị nội dung (một số trường hợp khác thì GridView tự suy ra control phù hợp, ví dụ như các trường boolean trong CSDL sẽ được hiển thị thành các checkbox…). Vậy giả sử bạn muốn nội dung trong một cột nào đó không phải hiển thị theo cách thông thường mà phải hiển thị nội dung theo cách bạn định nghĩa cho nó thông qua mã HTML.

Ví dụ, giả sử trong trường hợp này chúng ta muốn tạo ra thêm một cột là "Full Name", cột này là kết hợp nội dung của hai cột First Name và Last Name. Các bước thực hiện là như sau

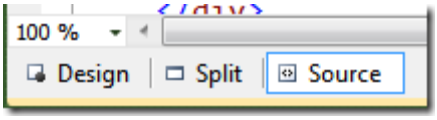
– Chọn chức năng Edit Columns trong hộp thoại nhỏ của GridView



– Trong hộp thoại mới hiện ra, chọn TemplateField và nhấn Add để thêm một cột do chúng ta tự định nghĩa nội dung hiển thị, chọn Header của cột TemplateField này là "Full Name", và sau đó nhấn OK



– Bây giờ là phần quan trọng nhất, bạn phải viết mã ASPX để hiển thị nội dung cho cột này. Các bạn chuyển sang chế độ chỉnh sửa mã

ASPX 

. Trong chế độ này, các bạn sẽ thấy là GridView của chúng ta được định nghĩa qua thẻ <asp:GridView>, trong thẻ này sẽ có thẻ <Columns> chứa định nghĩa cho tất cả các cột được hiển thị. Lưu ý là vì ở đây chúng ta đã chọn thuộc tính tự động tạo cột (AutoGenerateColumns = true) dựa vào dữ liệu trong database nên các cột này không cần phải định nghĩa trong thẻ <Columns>. Thẻ <Columns> của chúng ta lúc này chỉ chứa duy nhất định nghĩa cho cột FullName mà thôi. Tuy nhiên, như các bạn thấy là thẻ này vẫn chưa có mã ASPX báo nó biết là hiển thị những gì và hiển thị như thế nào

```
<asp:GridView ID="GridView1" runat="server" DataSourceID="SqlDataSource1"
    AllowPaging="True" AllowSorting="True">
    <Columns>
        <asp:TemplateField HeaderText="FullName"></asp:TemplateField>
    </Columns>
</asp:GridView>
```

Để hiển thị FirstName và LastName trong cột này thì chúng ta sẽ sử dụng cú pháp sau đây

```
<asp:TemplateField HeaderText="FullName">
    <ItemTemplate>
        <%# Eval("FirstName") %>,
        <%# Eval("LastName") %>
    </ItemTemplate>
</asp:TemplateField>
```

Thẻ <ItemTemplate> cho biết rằng nội dung bên trong thẻ này sẽ được hiển thị khi GridView ở chế bộ hiển thị bình thường với dữ liệu ở mỗi dòng (GridView còn có các chế độ như Update, Insert). Cú pháp Eval được sử dụng để lấy dữ liệu của thuộc tính được chỉ định, trong trường hợp này, chúng ta sẽ lấy ra giá trị của thuộc tính FirstName và LastName của mỗi dòng được hiển thị trên GridView. Các bạn có thể xem thêm về Data Binding tại đây: Data Binding trong ASP.NET

Chạy trang web vừa được chỉnh sửa, bạn sẽ có thêm được cột FullName với giá trị là sự kết hợp giữa cột FirstName và cột LastName

FullName	FirstName	MiddleName	LastName	Phone
Orlando, Gee	Orlando	N.	Gee	245-555-0173
Keith, Harris	Keith		Harris	170-555-0127
Donna, Carreras	Donna	F.	Carreras	279-555-0130
Janet, Gates	Janet	M.	Gates	710-555-0173
Lucy, Harrington	Lucy		Harrington	828-555-0186
Rosmarie, Carroll	Rosmarie	J.	Carroll	244-555-0112
Dominic, Gash	Dominic	P.	Gash	192-555-0173
Kathleen, Garza	Kathleen	M.	Garza	150-555-0127
Katherine, Harding	Katherine		Harding	926-555-0159
Johnny, Caprio	Johnny	A.	Caprio	112-555-0191
1 2 3 4 5 6 7 8 9 10 ...				

Có thể nói TemplateField cho bạn khả năng tùy biến cực lớn về cách dữ liệu sẽ hiển thị như thế nào trong mỗi ô của bảng. Bạn có thể thay vì hiển thị dạng Text thì hiển thị dưới dạng ảnh (ví dụ hiển thị Avatar của người dùng), âm thanh, ListBox… Điều đó phụ thuộc vào nhu cầu của bạn.

4. Thao tác với dữ liệu

Việc đầu tiên là bạn phải tạo ra một cơ sở dữ liệu để chúng ta thao tác trên nó. Bạn mở SQL Server lên và tạo một query mới gõ vào đoạn code sau:

```
create database bcdonlinesv
```

```
go
use bcdonlinesv
go
create table sinhvien
(
	masv int identity(1,1) primary key,
	tensv varchar(50),
	ngaysinh datetime,
	phai bit,
	diachi varchar(50),
	dienthoai varchar(15),
	email varchar(50)
```

```
)
```

Ở đây là các thuộc tính của bảng sinh viêc, trong đó masv là kiểu int tự tăng với bước nhảy là 1.

Chuẩn bị giao diện

Sau khi đã tạo được CSDL bạn cần tạo nên một giao diện để test thử bài tập từ đó rút ra được bài học. Hiển nhiên là bạn tạo giao diện phải tương ứng với các thuộc tính trong bảng trên.

Bạn sử dụng các control đã được hỗ trợ trong Toolbox của Visual Studio và bao nút nhấn: Insert, Update và Delete dưới đây là giao diện mình làm.

Giao diện demo thêm xóa sửa trong ASP.Net

Ở đây mình chỉ demo cách mà bạn thêm, xóa, sửa được vào cơ sở dữ liệu nên giao diện đề nghị ở đây rất đơn giản đủ để demo tính năng.

Insert – Theme trong ASP.Net

Công việc này sẽ làm các dữ liệu bạn nhập, chọn ngoài giao diện sẽ đi vào trong CSDL của bcdonlinesv mà ta tạo lúc đầu.

Bạn Click dounle và nut Insert visual sẽ tự phát sinh cho bạn một hành động đó là Button Click trong đó bạn gõ vào đoạn code sau:

```
protected void btInsert_Click(object sender, EventArgs e)
```

```
    {
        SqlConnection con = new
SqlConnection("server=.;database=bcdonlinesv;Integrated
Security=true;");
        int phai = 0;
        if (RadioNam.Checked == true)
            phai = 1;
        SqlCommand cmd = new SqlCommand("insert into
sinhvien(tensv,ngaysinh,phai,diachi,dienthoai,email) values('" +
txtTen.Text + "'," + NgaySinh.SelectedDate.ToShortDateString() +
",'" + phai + "','" + txtDiaChi.Text + "','" + txtDienThoai.Text
+ "','" + txtEmail.Text + "')", con);
        con.Open(); // mo ket noi
        cmd.ExecuteNonQuery(); // thuc thi
        con.Close();
    }
```

Quên trước đó bạn phải khai báo thư viện để sử dụng được SqlConnection trong đây:

```
using System.Data.SqlClient;
```

Ghi chú:

SqlConnection: là lớp dùng để kế nối cơ sở dữ liệu SQL server, bên trong nó là chuỗi Connection.

SqlCommand: Thực thi một truy vấn nào đó bên trong nó, ở đây là INSERT.

Do mình thiết kế lúc đầu MaSV tăng tự động nên chúng ta không nhập.

Sau đó bạn chạy lên và thêm thử, ở hình demo phía trên và mở lại CSDL để xem.

BANCAMDEN.bcdonl...v - dbo.sinhvien

masv	tensv	ngaysinh	phai	diachi	dienthoai	email
7	bcdonline	1900-01-01 00:...	True	DongThap	0123456789	bcdonline.net@...
NULL	NULL	NULL	NULL	NULL	NULL	NULL

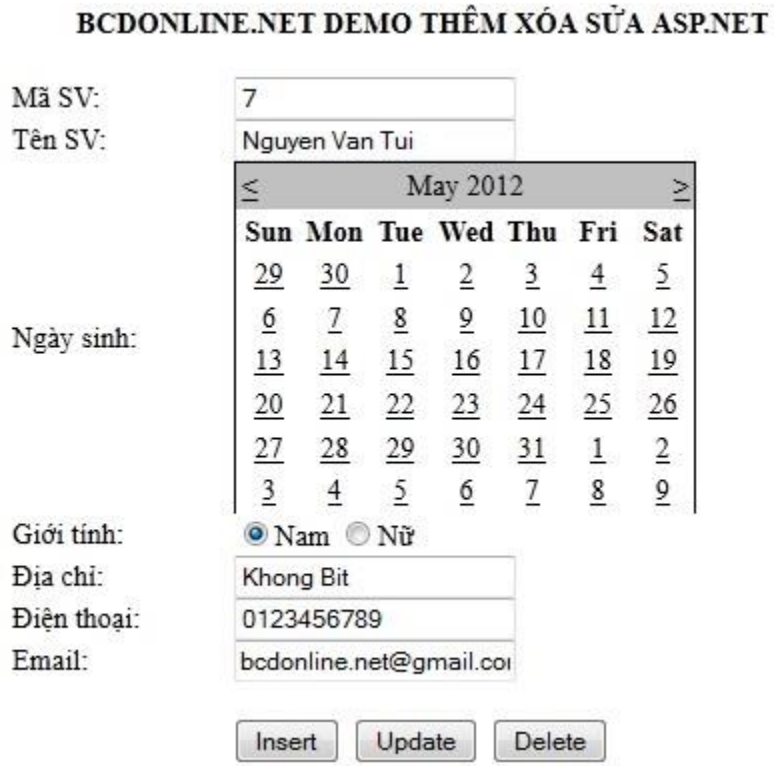
Cơ sở dữ liệu sau khi thêm một sinh viên

Sửa – Update trong ASP.Net

Cơ chế này là sửa thông tin các bảng dữ liệu nhờ vào khóa chính của bộ. Bạn click double vào nut update như đã là ở Insert, sau đó bạn gõ vào các dòng code sau nơi mà sự kiện click tại nút update (Bạn vừa click double).

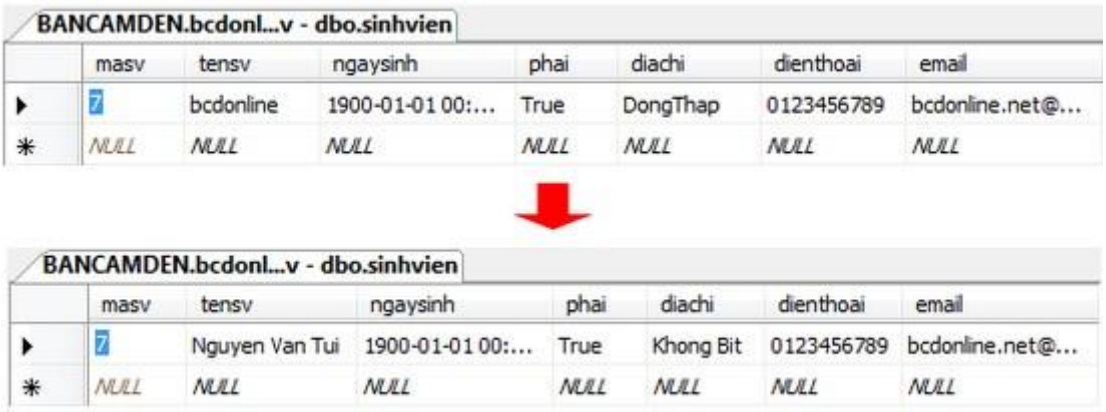
```
        protected void btUpdate_Click(object sender, EventArgs
e)
        {
            SqlConnection          con          =          new
SqlConnection("server=.;database=bcdonlinesv;Integrated
security=true;");
            int phai = 0;
            if (RadioNam.Checked == true)
                phai = 1;
            SqlCommand cmd = new SqlCommand("update sinhvien
set    tensv='"    +    txtTen.Text    +    "',ngaysinh="    +
NgaySinh.SelectedDate.ToShortDateString() + ",phai='" + phai
+   "',diachi='"   +   txtDiaChi.Text   +   "',dienthoai='"   +
txtDienThoai.Text + "',email='" + txtEmail.Text + "'  where
masv='" + txtMa.Text + "'", con);
            con.Open(); // mo ket noi
            cmd.ExecuteNonQuery(); // thuc thi
            con.Close();
        }
```

Bạn dễ dàng thấy là chỉ khác biệt ở truy vấn trong SqlCommand, cái này bạn phải học SQL. Sau đó lưu lại và chạy test bằng trình duyệt, bạn nhập bất kỳ các thông tin gì và bấm nút Update.



Chức năng Update - sửa trong ASP.Net

Để xem kết quả mình làm có đúng hay không bạn vào hệ quản cơ sở dữ liệu SQL Server để kiểm tra. Bạn sẽ dễ dàng thấy được sự khác biệt của hai bộ trước và sau khi update.



Thay đổi sau khi update sinh vien

Delete – Xóa trong ASP.Net

Đây là công việc xóa tất cả các dòng dữ liệu thuộc một bộ nào đó, mình chỉ demo xóa thôi, không tính quan hệ dữ liệu ở đây. Công việc cũng như trên bạn click double vào nút Delete và code đoạn mã sau:

```
protected void btDelete_Click(object sender, EventArgs e)
{
```

```
        SqlConnection con = new
SqlConnection("server=.;database=bcdonlinesv;Integrated
security=true;");
        SqlCommand cmd = new SqlCommand("delete from sinhvien
where masv='" + txtMa.Text + "'", con);
        con.Open(); // mo ket noi
        cmd.ExecuteNonQuery(); // thuc thi
        con.Close();
    }
```

BÀI 5: LÀM VIỆC VỚI TẬP TIN

Mã Bài: MĐ19_05

Giới thiệu:

Upload tài liệu là một trong những tính năng không thể thiếu trong mọi lĩnh vực hoạt động trực tuyến nhất là với các trang web thương mại điện tử và tin tức.

Việc tương tác vào nội dung tập tin upload dạng Text có thể thông qua các lớp **StreamReader** và **StreamWriter** trong C#. Hai lớp này được sử dụng để đọc và ghi dữ liệu tới text file. Những lớp này kế thừa từ lớp abstract cơ sở là Stream, mà hỗ trợ việc đọc và ghi các byte vào trong File Stream.

Mục tiêu:

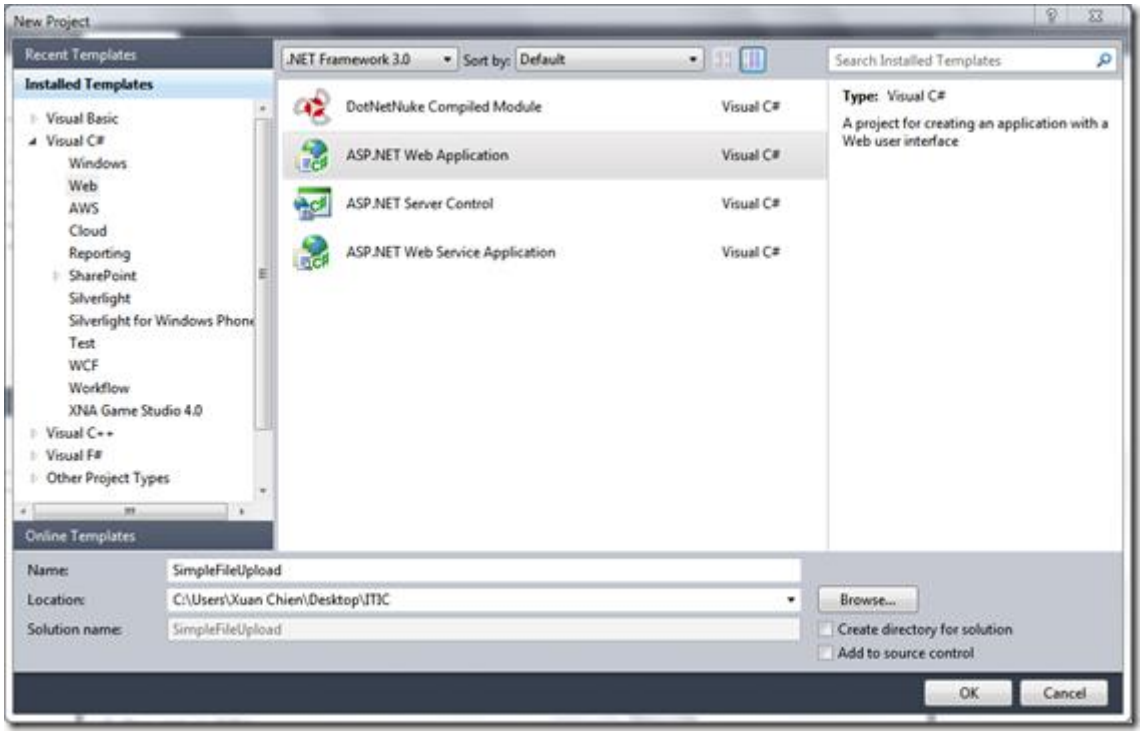
- Viết được các ứng dụng web quản lý tập tin, thư mục trên Server và tập tin upload.

Nội dung chính:

1. Quản lý hệ thống tập tin

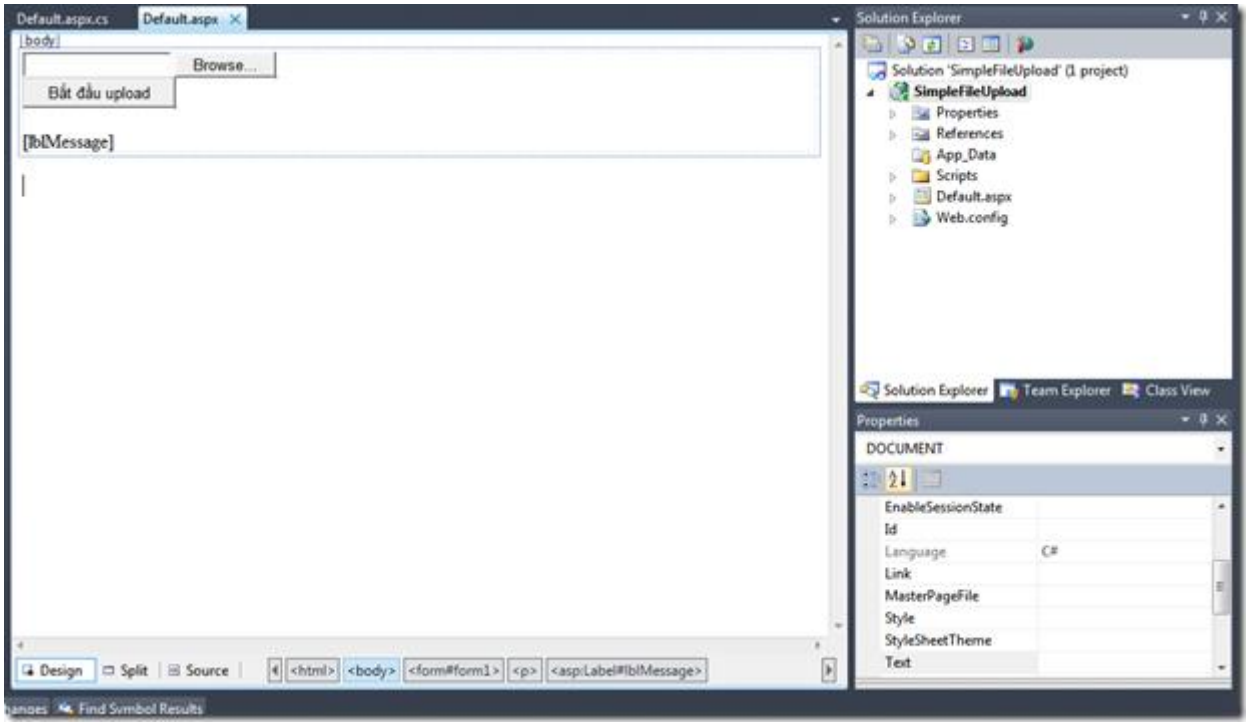
Chúng ta sẽ sử dụng một controls có sẵn trong ASP.NETdùng để upload một tập tin lên Server. Chúng ta còn thực hiện thêm việc kiểm tra loại tập tin…

Trước hết, chúng ta tạo một project ASP.NET Web Application mới tên là SimpleFileUpload



Trong phần thiết kế của trang Default.aspx, chúng ta sẽ kéo control FileUpload trong ToolBox vào trang. Đặt trên cho control này là ***fuPicture***. Sau đó kéo thêm

một Button vào trang và đặt tên cho button này là ***btnUpload.*** Để biết được quá trình Upload đã hoàn thành hay chưa, chúng ta sẽ dùng thêm một Label đặt trên trang web để thông báo, đặt tên Label này là ***lblMessage***



Việc tiếp theo là viết phương thức xử lý sự kiện người dùng click chuột vào nút "Bắt đầu upload". Các bạn nhấn double click vào button này để bắt đầu viết xử lý. Nội dung phương thức xử lý sự kiện là như sau:

```
protected void btnUpload_Click(object sender, EventArgs e)
{
    if (fuPicture.HasFile)
    {
        fuPicture.SaveAs(Server.MapPath("~") +
fuPicture.FileName);
        lblMessage.Text = "Upload thành công";
    }
}
```

Control FileUpload cung cấp cho chúng ta nhiều thuộc tính để biết được thông tin về tập tin đã được chọn. Một số thuộc tính quan trọng là:

HasFile: nếu đã có tập tin được chọn

PostedFile: đây là thuộc tính lưu trữ các thông tin khác về tập tin như kích thước tập tin, loại nội dung tập tin…

FileName: tên của tập tin được chọn

Ngoài ra, Control FileUpload còn có phương thức SaveAs cho phép chúng ta lưu trữ tập tin được chọn. Các bạn lưu ý rằng trong đoạn code trên thì chúng ta có sử dụng phương thức Server.MapPath("~") cộng với tên tập tin. Lý do là vì phương thức SaveAs yêu cầu chúng ta phải cung cấp đường dẫn đầy đủ của Server tính từ địa chỉ gốc, do đó Server.MapPath("~") chính là dùng để lấy địa chỉ gốc của ứng dụng Web chúng ta đang chạy.

Bây giờ nhấn F5 để xem chạy trang web. Các bạn chọn một tập tin sau đó nhấn "Bắt đầu upload". Mở thư mục chứa trang web thì chúng ta sẽ thấy tập tin vừa được chọn xuất hiện trong đó.

Bây giờ, chúng ta sẽ bổ sung thêm tính năng hạn chế chỉ cho phép một số loại tập tin được phép upload lên bằng cách kiểm tra phần mở rộng của tập tin đó. Nội dung đoạn mã kiểm tra phần mở rộng của tập tin như sau, giả sử rằng chúng ta đang muốn chỉ cho phép người dùng upload các tập tin đuôi .png

```
private bool CheckFileType(string FileName)
{
    string ext = Path.GetExtension(FileName);
    if (ext.Equals(".png"))
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

Chúng ta sẽ thực hiện tích hợp đoạn mã kiểm tra với đoạn mã upload như sau:

```
protected void btnUpload_Click(object sender, EventArgs e)
{
    if (fuPicture.HasFile)
    {
        if (CheckFileType(fuPicture.FileName))
        {
            fuPicture.SaveAs(Server.MapPath("~") +
fuPicture.FileName);
            lblMessage.Text = "Upload thành công";
```

```
        }
        else
        {
            lblMessage.Text = "Loại file không hợp lệ";
        }
    }
}
```

Nhấn F5 để biên dịch lại ứng dụng Web và thử chọn một tập tin bất kỳ, dễ dàng thấy được rằng bạn chỉ có thể upload được các tập tin có đuôi .png, còn các loại tập tin khác sẽ không thể upload được.

2. Di chuyển, Sao chép, Huỷ File

Dựa trên các đối tượng FileInfo và DriveInfo đã tạo ra ở trên mà chúng ta có thể gọi các phương thức để thực hiện việc Sao chép, di chuyển và xóa tập tin/thư mục. Các phương thức quan trọng của các lớp FileInfo là:

CopyTo: Sao chép tập tin đến đường dẫn được chỉ định và trả về đối tượng FileInfo của tập tin vừa được sao chép. Ví dụ sao chép tập tin đến ổ C: fileInfo.CopyTo(@"C:\" + fileInfo.Name); (các bạn lưu ý kí tự @ trước chuỗi để khỏi phải gõ ký tự \ hai lần.

Delete: Xóa tập tin vĩnh viễn nếu nó thực sự tồn tại. Lưu ý rằng phương thức này sẽ xóa trực tiếp tập tin vĩnh viễn chứ không phải là di chuyển tập tin đó đến thùng rác. Phiên bản hiện tại của C# không có cung cấp phương thức nào để chúng ta di chuyển tập tin đến thùng rác mà chúng ta cần phải tham chiếu đến thư viện của Visual Basic. Xem thêm tại: Delete File to Recycle Bin in C#

MoveTo: Di chuyển tập tin này đến vị trí mới. Phương thức này cũng được dùng để đổi tên một tập tin mà không thay đổi đường dẫn của nó. Ví dụ di chuyển tập tin đến ổ C: fileInfo.MoveTo(@"C:\" + fileInfo.Name);

Replace: Thay thế nội dung của tập tin nào đó được chỉ định bằng nội dung của tập tin chứa trong đối tượng FileInfo

Đối với thư mục thì chúng ta có các phương thức sau đây nằm trong lớp DirectoryInfo:

Create: Tạo ra một thư mục mới từ đối tượng DirectoryInfo

CreateSubdirectory: Tạo ra thư mục con chứa trong đường dẫn được chỉ định

Delete: Xóa thư mục

MoveTo: Di chuyển thư mục (kể cả nội dung bên trong) đến đường dẫn mới nằm trên cùng một ổ đĩa. Phương thức MoveTo cũng được dùng để đổi tên thư mục.

Ví dụ:

```
//Sao chép tập tin
File.Copy(@"C:\abc.txt", @"F:\Temp\test.txt");
//Xóa tập tin
File.Delete(@"C:\abc.txt");
//Di chuyển tập tin
File.Move(@"C:\abc.txt", @"F:\Temp\test.txt");

//trả về mảng chuỗi các tập tin trong thư mục MyFolder của ổ
//đĩa C
Directory.GetFiles(@"C:\MyFolder\");
```

3. Đọc và viết vào File

Trong ASP.NET chúng ta có thể dễ dàng đọc ghi một file text giống như Notepad. Để làm được như vậy chúng ta chỉ cần sử dụng thư viện System.IO

```
using System.IO;
Code ghi, đọc file
 protected void Button1_Click(object sender, EventArgs e)
    {
        string path = Server.MapPath("TextFile.txt");
        StreamWriter SW = new StreamWriter(path);
        SW.Flush();
        string str = TextBox1.Text;
        SW.WriteLine(str);
        SW.Close();
    }
 protected void Button2_Click(object sender, EventArgs e)
    {
        string path = Server.MapPath("TextFile.txt");
        StreamReader SR = new StreamReader(path);
        TextBox1.Text = SR.ReadLine();
        SR.Close();
    }
```

4. Quản lý tập tin upload

Khi muốn upload một tập tin lên Server chúng ta sử dụng lệnh sau:

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Upload : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void btnUpload_Click(object sender, EventArgs e)
    {
        if (Page.IsValid && FileUpload1.HasFile)
        {
            string fileName = "images/" + FileUpload1.FileName;
            string filePath = MapPath(fileName);
            FileUpload1.SaveAs(filePath);
            Image1.ImageUrl = fileName;
        }
    }
}
```

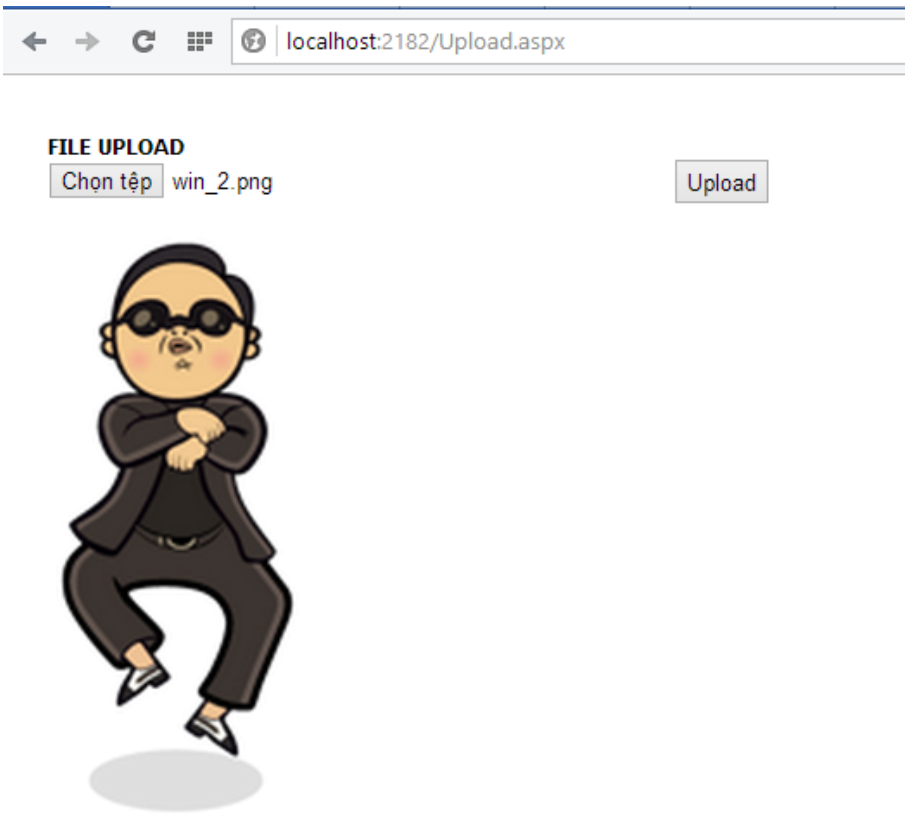
Trong đó:

+ FileUpload1.HasFile: Kiểm tra xem có tệp tin được tải lên không ?

+ string filePath = MapPath(fileName): Dòng lệnh này lấy đường dẫn tuyệt đối của tệp tin sẽ lưu trên website và lưu thông tin đó vào biến filePath.

+ FileUpload1.SaveAs(filePath): Dòng lệnh dùng để lưu tệp tin lại theo đường dẫn filePath (đã có từ câu lệnh trên).

Hoàn tất trang upload của bạn sẽ có giao diện giống hình sau:



Tuy nhiên, để tránh việc ghi đè các tệp tin bị trùng tên, bạn có thể sửa dòng code:

```
        string fileName = "images/" + FileUpload1.FileName;
```

trong phương thức btnUpload_Click như sau:

```
string fileName = "images/"
+DateTime.Now.ToString("ddMMyyyy_hhmmss_tt_")
+FileUpload1.FileName;
```

Lúc này tên tệp tin của bạn sẽ được chèn thêm thời gian upload nên tránh được trường hợp trùng tên file.

+ Nếu muốn hạn chế loại tệp tin được upload. Bạn có thể kiểm tra dựa trên phần mở rộng của tên tệp tin theo đoạn code dưới đây:

```
bool CheckFileType(string fileName)
    {

        string ext = Path.GetExtension(fileName);
        switch (ext.ToLower())
        {
            case ".gif":
                return true;
```

```
            case ".png":
                return true;
            case ".jpg":
                return true;
            case ".jpeg":
                return true;
            default:
                return false;
        }
    }

protected void btnUpload_Click(object sender, EventArgs e)
    {
        if (Page.IsValid && FileUpload1.HasFile &&
CheckFileType(FileUpload1.FileName))
        {
            string fileName = "images/"
+DateTime.Now.ToString("ddMMyyyy_hhmmss_tt_")
+FileUpload1.FileName;
            string filePath = MapPath(fileName);
            FileUpload1.SaveAs(filePath);
            Image1.ImageUrl = fileName;
        }
 }
```

BÀI 6: LẬP BÁO CÁO VỚI CRYSTAL REPORT

Mã Bài: MĐ19_06

Giới thiệu:

Crystal Report có thể tạo trong cả hai ngôn ngữ là VB.NET hoặc C#.NÊT và nó có thể là một Window Application hoặc ASP.NET. Thiết kế báo cáo với Crystal Report là một kiểu chương trình độc lập. Những bước để làm một Crystal thì giống nhau cho cả hai kiểu chương trình ứng dụng này.

Mục tiêu:

- Xuất được dữ liệu từ cơ sở dữ liệu Access hoặc SQL Server ra công cụ Crystal

Report để phục vụ cho báo cáo.

Nội dung chính:

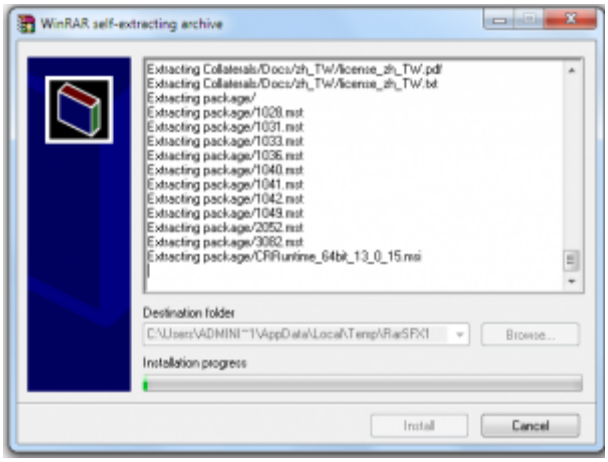
1. Giới thiệu công cụ Crystal Report

1.1. Tải và cài đặt Crystal Report

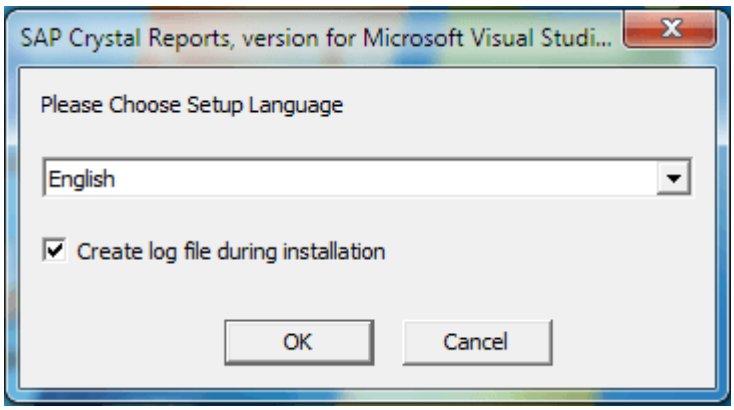
Đầu tiên, các bạn có thể download phiên bản mới nhất của của Crystal Report để cài đặt.

Tiếp theo bạn cài đặt phiên bản Crystal Report Service Pack hỗ trợ cho các phiên bản của Visual Studio từ 2010 trở về sau.

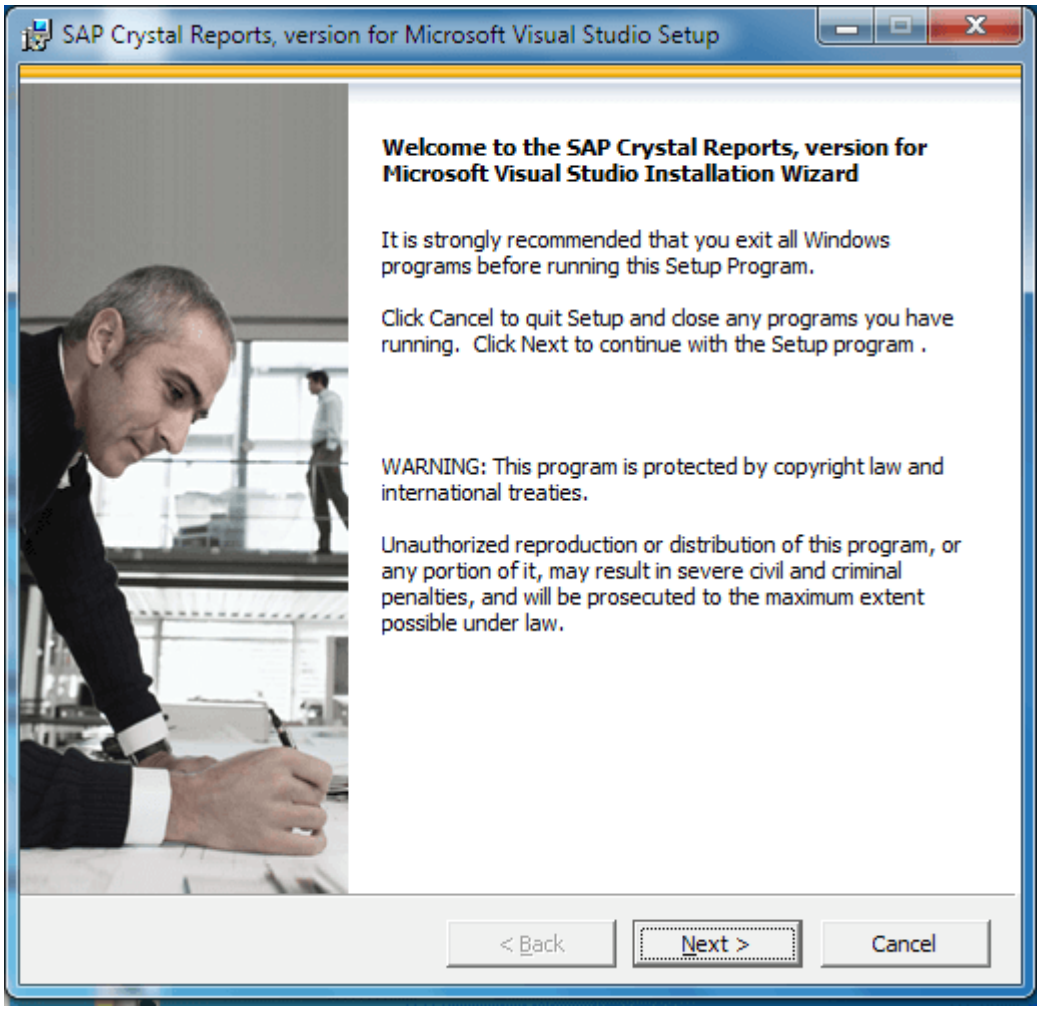
Đăng nhập vào máy với quyền quản trị (Administrator). Tải tập tin cài đặt theo địa chỉ trên. Xong chạy tập tin vừa tải về này (nhấp đúp chuột để chạy)



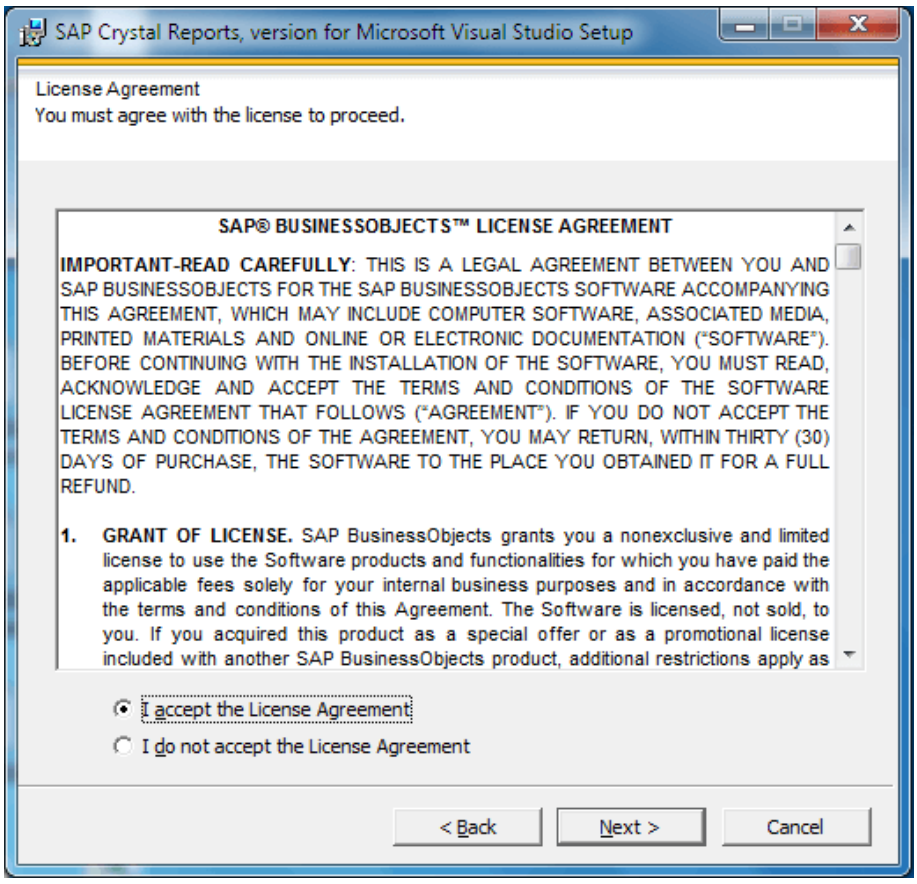
Hộp thoại chọn ngôn ngữ xuất hiện, chọn ngôn ngữ là English, xong chọn *OK* để đồng ý.



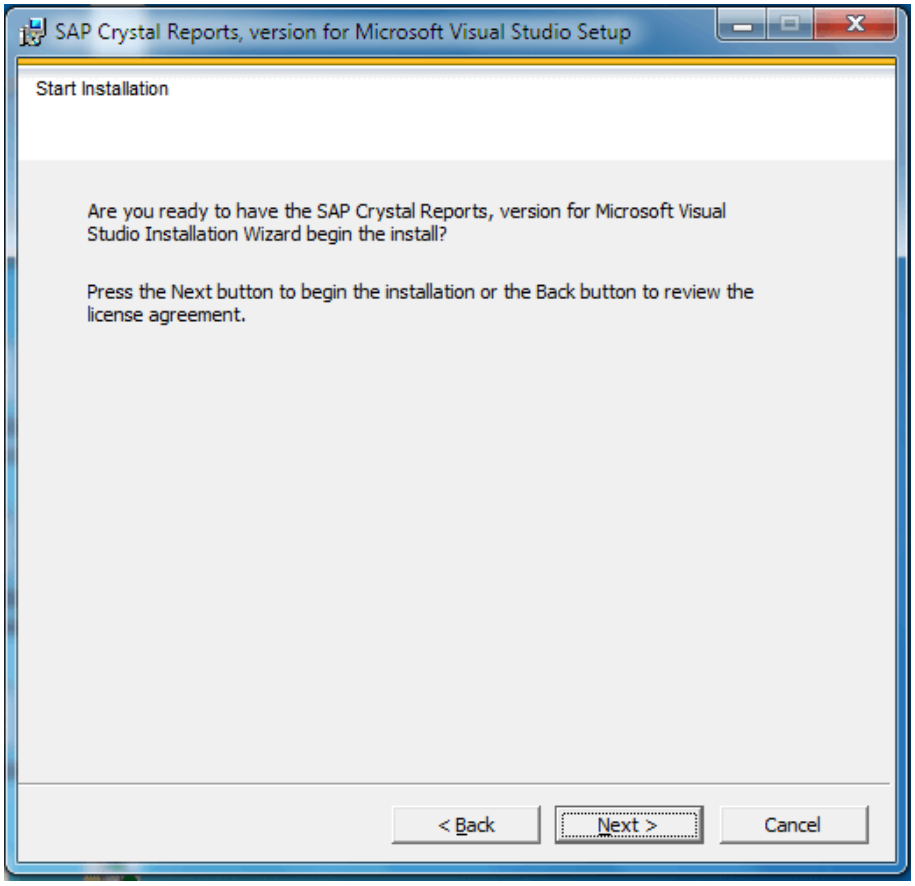
Màn hình chào của Crystal Report xuất hiện, bạn chọn *Next* để tiếp tục.



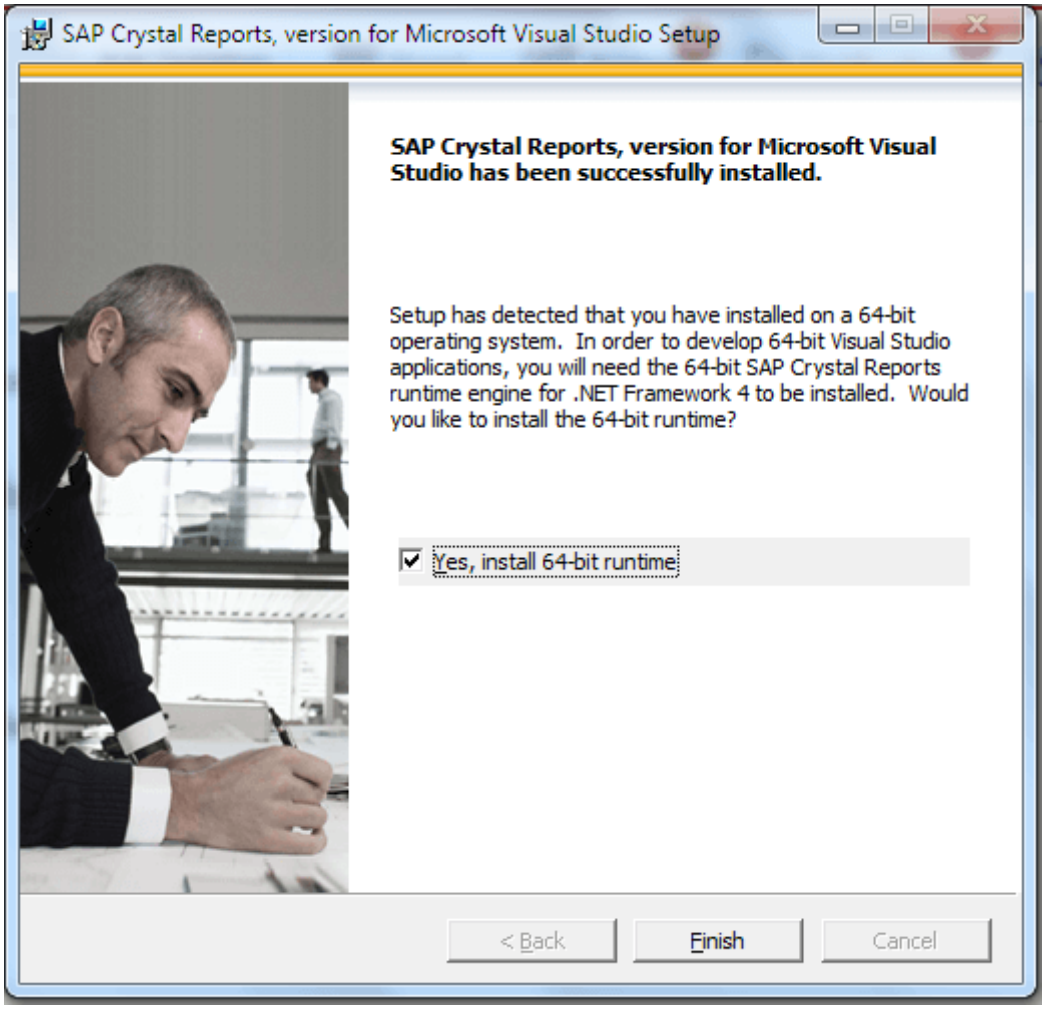
Màn hình bản quyền của Crystal Report xuất hiện, bạn chọn *I accept the License Agreement* để tiếp tục.



Màn hình bắt đầu cài đặt xuất hiện, chọn *Next* để tiếp tục.



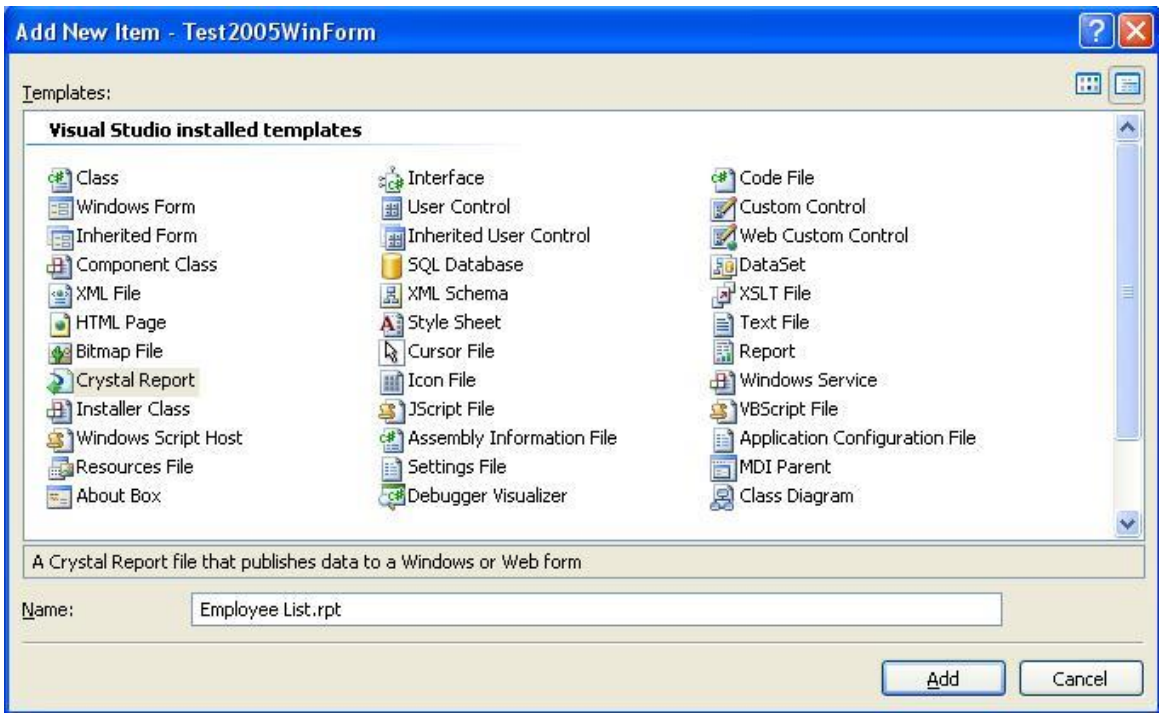
Quá trình cài đặt bắt đầu, thời gian cài đặt phụ thuộc vào cấu hình máy, thường mất khoảng 3 – 6 phút. Sau khi cài đặt hoàn thành, màn hình bên dưới sẽ xuất hiện. Nếu bạn muốn cài đặt phần Runtime của Crystal Report, chọn *Yes, install 64-bit runtime* hoặc 32-bit tùy theo máy. Xong chọn *Finish*.



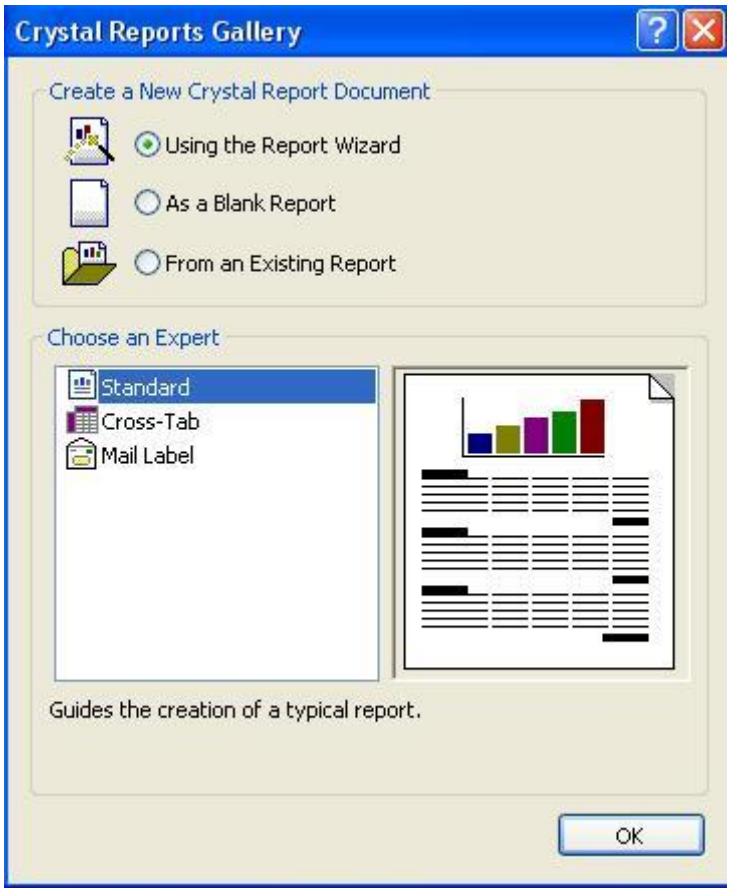
Như vậy, quá trình cài Crystal Report hoàn tất.

1.2 Tạo một Crystal Report

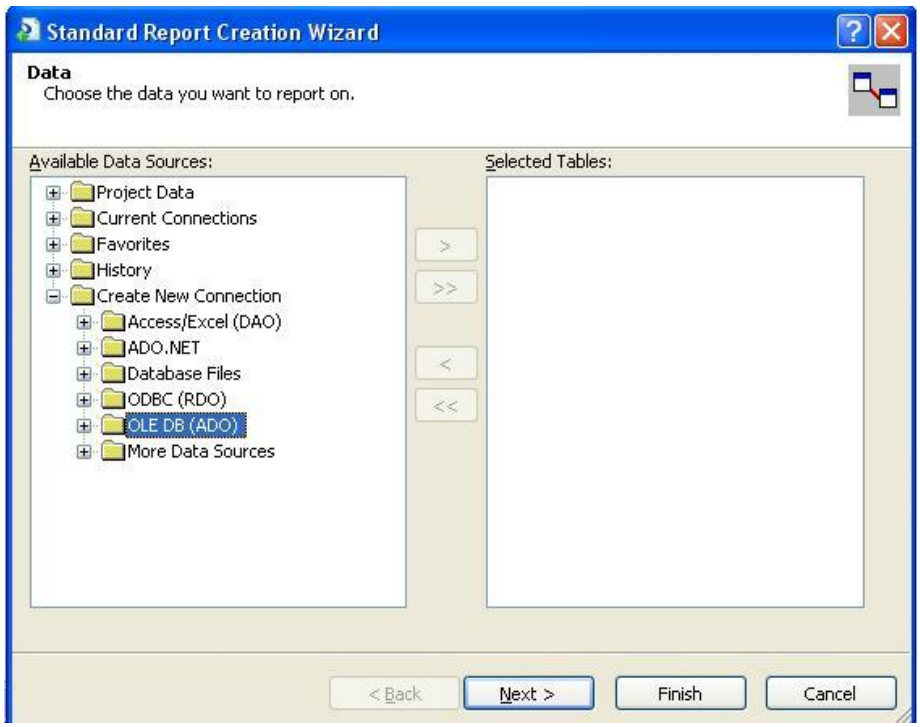
Trong Project được mở bạn chọn Project > Add New Item . Điều này sẽ làm hiện ra những dánh sách các kiểu mẫu chương trình có thể thấy. Bạn chọn kiểu trong này là Crystal Report . Nhập tên cho nó là Employee List . Hình dưới sẽ cho bạn thấy hộp hội thoại này cho một Window Application của bạn . Giờ bạn hãy click vào create để tạo một report .



Khi đó hộp thoại Crystal Report Gallery sẽ xuất hiện ... Bạn hãy chấp nhận hai giá trị mặc định của nó là Using The Report Wizard và Standard và nhấn OK (Hình dưới)

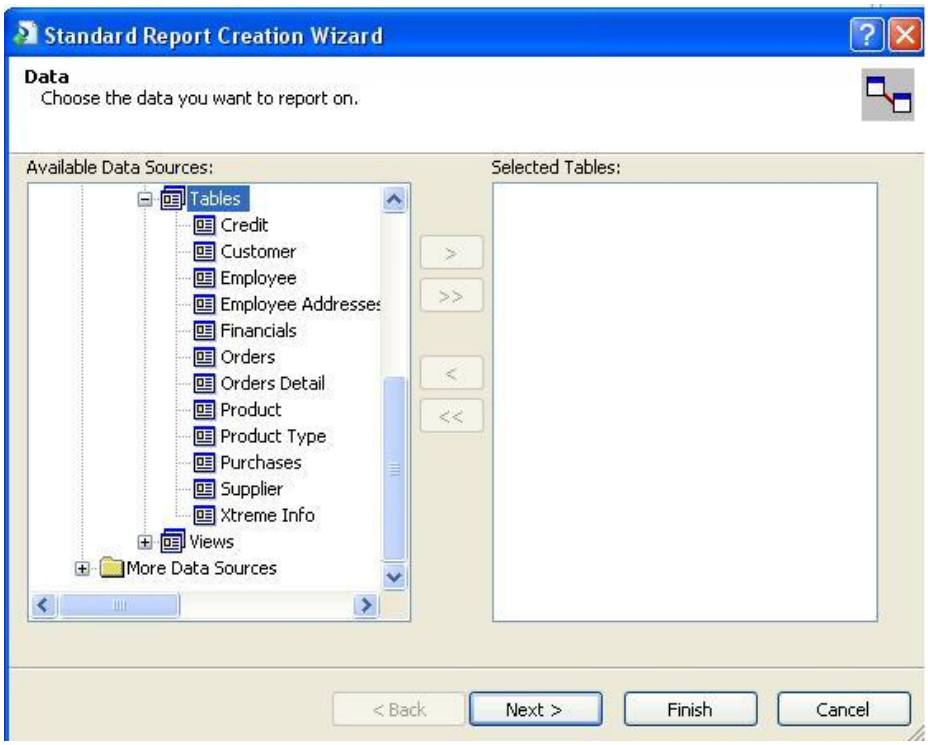


Bây giờ điều tiếp theo là ta phải chọn Cơ sỡ dữ liệu Xtreme bằng việc nhấn chuột vào tuỳ chọn OLE DB(ADO) (hình dưới) .

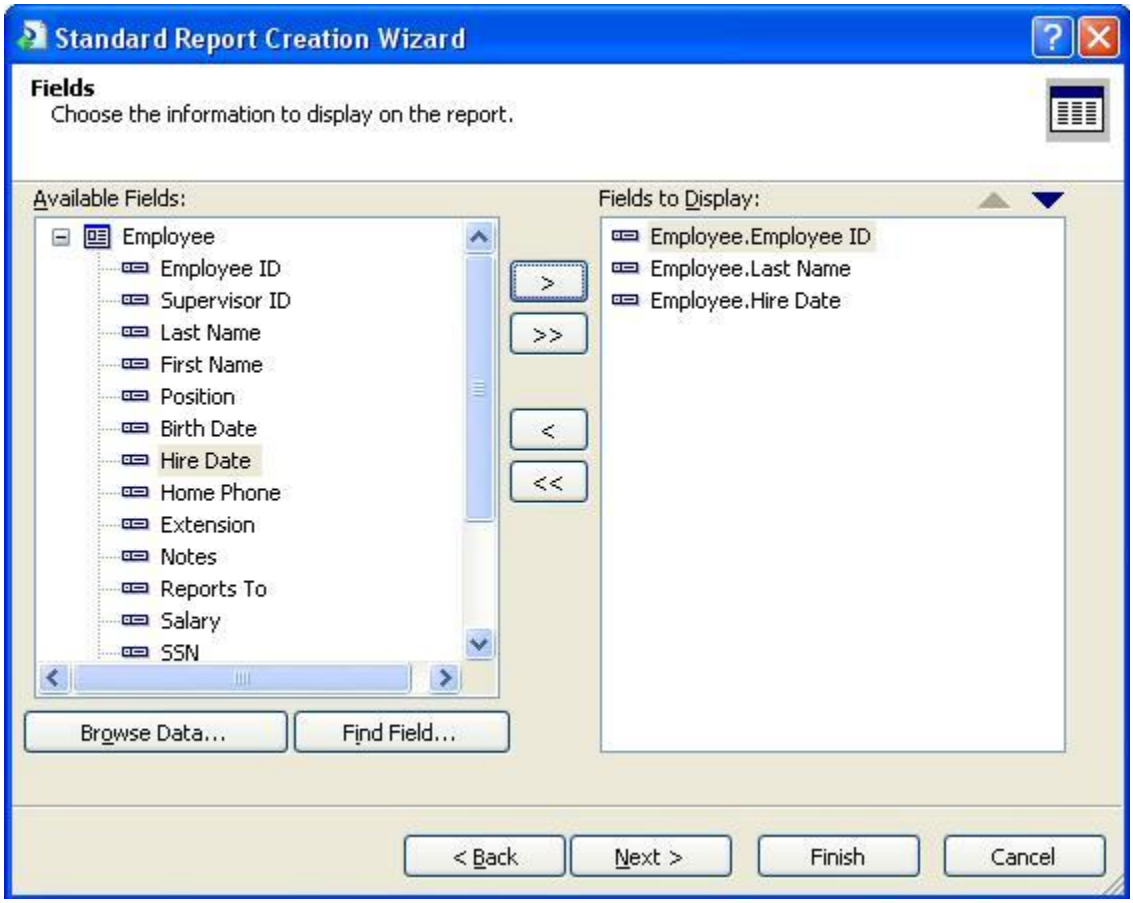


Điều này sẽ làm xuất hiện một hộp thoại mới và bạn chọn trong này là Microsoft Jet 4.0 OLE DB Provider . Trong hộp thoại tiếp theo bạn chọn đường dẫn đến căn cứ dữ liệu Xtreme (database name phải là tên đường dẫn đầy đủ trong máy bạn ví dụ D:\Microsoft Visual Studio .NET 2005\Crystal Reports\Samples\en\Databases) . Sau đó bạn Click vào Finish .

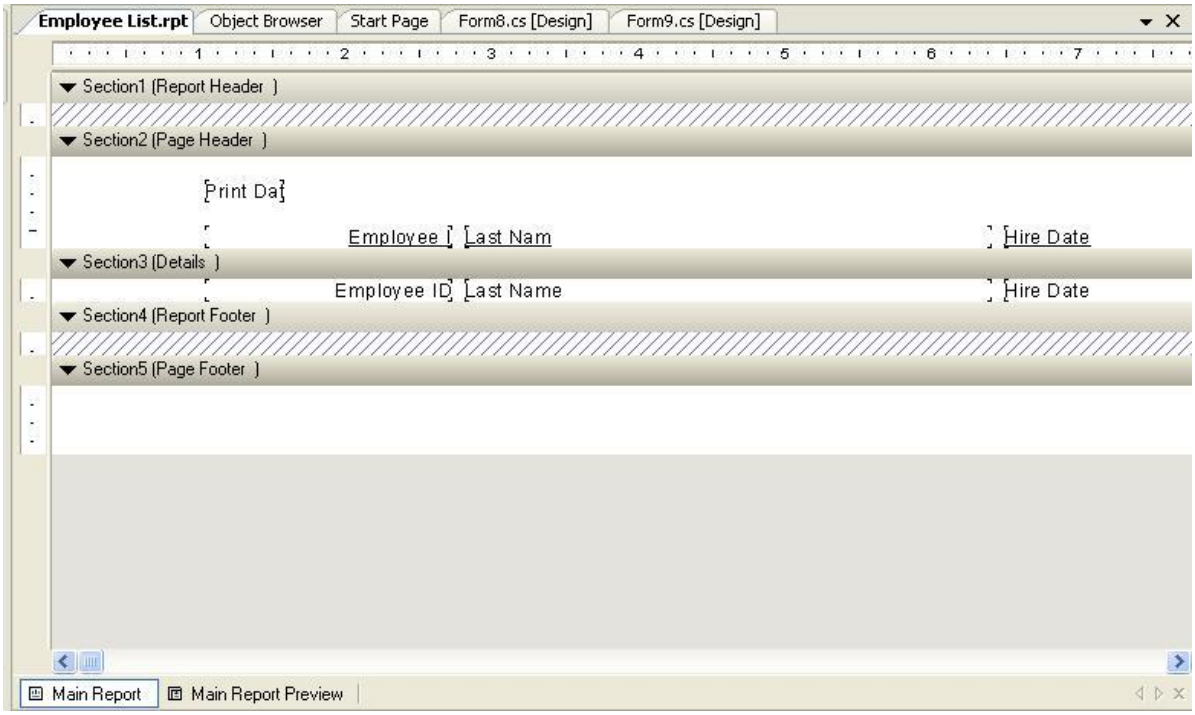
Giờ bạn sẽ chở lại tab Data . Bạn hãy Click vào mắt Tables để bung nó ra và double vào bảng Employee (hoặc bạn có thể kéo và thả nó) để có thể di chuyển nó qua cửa sổ mới tên là Tables In Report . Click vào Next



Để hiển thị những vùng cần có trên báo cáo , bạn có thể chọn chúng tại tab Fields . Giờ bạn hãy double vào những vùng bạn chọn để chèn chúng vào cửa sổ bên phải : Trong ví dụ này Tôi sẽ chọn các vùng Employee ID , Last Name , Hire Data . Bạn sẽ thấy nó như hình dưới .



Đến đây bạn có thể tiếp tục theo tác với báo cáo của bạn như nhóm lại và chọn dữ liệu để in . Nhưng đây chỉ là một ví dụ đơn giản , bạn hạy bỏ qua những tab còn lại và cứ thế nhấn Next để đến tab Style rồi nhấn Finish . Giờ bạn sẽ thấy một báo cáo của bạn sẽ thấy như hình dưới . Nếu báo cáo của bạn sử dụng trong một chương trình ứng dụng thực tế thì bước tiếp theo của bạn là bạn phải sửa đổi Form để có thể xem và in báo cáo này .



2. Gắn kết dữ liệu vào báo cáo

Các bước thực hiện gắn kết dữ liệu từ DataSet (data) vào CrystalReport1:

```
//tạo một instance của report đã tạo trong Designer
CrystalReport1 crysRpt = new CrystalReport1();
//thiết lập datasoruce chỉ tới chính table đã được sử dụng
//trong khi thiết kế report
crysRpt.SetDataSource(data.Tables[0]);
```

3. Xây dựng và triển khai báo cáo

Trong môi trường web, để báo cáo in ra được đẹp và đúng mẫu chúng ta có thể export từ Crystal report ra nhiều định dạng như PDF, Word, Excel, … có thể làm việc này bằng cách lập trình.

Ví dụ đoạn code chương trình sau sẽ xuất dữ liệu từ Crystal Report ra Excel:

```
using CrystalDecisions.CrystalReports.Engine;
......
private void button1_Click(object sender, EventArgs e)
        {    cryRpt = new ReportDocument();
             cryRpt.Load(PUT CRYSTAL REPORT PATH
HERE\\CrystalReport1.rpt");
             crystalReportViewer1.ReportSource = cryRpt;
             crystalReportViewer1.Refresh();
        }
```

```
      private void button2_Click(object sender, EventArgs e)
        {
            try
            {
                ExportOptions CrExportOptions ;
                DiskFileDestinationOptions
CrDiskFileDestinationOptions = new DiskFileDestinationOptions();
                ExcelFormatOptions CrFormatTypeOptions = new
ExcelFormatOptions();
                CrDiskFileDestinationOptions.DiskFileName =
"c:\\csharp.net-informations.xls";
                CrExportOptions = cryRpt.ExportOptions;
                CrExportOptions.ExportDestinationType =
ExportDestinationType.DiskFile;
                CrExportOptions.ExportFormatType =
ExportFormatType.Excel;
                CrExportOptions.DestinationOptions =
CrDiskFileDestinationOptions;
                CrExportOptions.FormatOptions =
CrFormatTypeOptions;
                cryRpt.Export();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.ToString());
            }
        }
```

Việc quy định kiểu tập tin cần xuất phụ thuộc vào tham số ExportFormatType (Excel, Word, PDF, …), từ dó chúng ta có giải pháp mở tập tin kết quả ra để in để đảm bảo được định dạng như thiết kế ban đầu trong Crystal Report.

BÀI 7: MỘT SỐ CÔNG CỤ XÂY DỰNG WEBSITE

Mã Bài: MĐ19_07

Giới thiệu:

Để việc phát triển website được thuận lợi với nền tảng .NET đã được học, trong bài này chúng tôi sẽ trọng tâm giới thiệu công cụ DotNetNuke.

Mục tiêu:

- Xác định được chức năng của một số công cụ xây dựng website như WordPress, Vbulletin, DotNetNuke ...;

- Phân tích xác định được chức năng của website để lựa chọn công cụ phù hợp;

- Xác định được công cụ xây dựng website cần dùng dựa trên chức năng của website cần xây dựng;

- Cài đặt được công cụ (ứng dụng web) trên máy chủ web cục bộ;

- Tùy chỉnh được các thông số cơ bản của công cụ cho phù hợp với thông tin của website cần xây dựng;

- Thận trọng, tỉ mỉ trong quá trình làm việc;

Nội dung chính:

1. Quy trình tổng quát để xây dựng website

Bước 1: Khách hàng chuẩn bị nội dung - hình ảnh

Khi có ý định thiết kế 1 website, Quý khách hàng nên chuẩn bị sẵn nội dung và hình ảnh mà mình muốn đưa lên website, như vậy chúng ta sẽ có cơ sở thiết kế được tốt hơn, đẹp hơn, nhanh chóng hơn. Tránh trường hợp trang web khi đưa lên internet mà chưa có nội dung.

Bước 2: Đăng ký tên miền (Domain name) và Thuê máy chủ (Webhosting)

* Đăng ký một tên miền là việc đầu tiên cần làm, tên miền riêng khẳng định vị trí, giúp khách hàng dễ tìm đến website của bạn, vừa bảo vệ thương hiệu của doanh nghiệp bạn trên Interrnet.

* Webhosting là nơi không gian trên máy chủ có cài dịch vụ Internet như ftp, www,... nơi đó bạn có thể chứa nội dung trang web hay dữ liệu trên không gian đó.

* Khách hàng lựa chọn tên miền cho website (Ví dụ: www.ten-cong-ty.com)

* Lựa chọn các gói dịch vụ Web Hosting phù hợp với Website.

Bước 3: Lấy yêu cầu thiết kế website

Ghi nhận các thông tin yêu cầu của khách hàng, tập trung vào các nội dung:

* Ghi nhận các yêu cầu về tính năng của website

* Ghi nhận các yêu cầu mỹ thuật đối với website

* Ghi nhận các yêu cầu về tên miền và máy chủ lưu trữ website

* Tư vấn thêm các tính năng dựa vào yêu cầu thực tế của khách hàng

* Tổng hợp bản yêu cầu xây dựng website và chi tiết kế hoạch thực hiện gửi khách hàng

* Tiếp nhận tư liệu, thông tin, hình ảnh phục vụ cho việc thiết kế website

* Ký hợp đồng thiết kế website

Bước 4: Thiết kế website

*** Thiết lập Cấu trúc giao diện thiết kế - Giá cả - Thời gian thực hiện**

Tư vấn cho khách hàng các giải pháp tốt nhất để xây dựng cấu trúc giao diện và các chức năng chính của website. Đây là bước khởi đầu hết sức quan trọng nhằm có một cấu trúc nội dung rõ ràng, mạch lạc, tránh lủng củng trong quá trình thiết kế Website. Tạo cho người xem một cái nhìn thân thiện, dễ sử dụng và truy cập nhanh.

*** Thiết kế giao diện mẫu**

Thiết kế một số giao diện mẫu để khách hàng chọn lựa và bổ sung ý tưởng thiết kế. Bước này rất quan trọng, vì mỗi người đều có cách nhìn về mặt thẩm mỹ khác nhau nên rất cần sự hổ trợ ý kiến đóng góp từ khách hàng nhằm làm hài lòng quý khách và làm cho Website trở nên sinh động hơn.

*** Nghiệm thu giao diện**

Để tránh trường hợp khó khăn về mặt thời gian và công sức khi thiết kế Website và lập trình, khách hàng quyết định chọn giao diện mà khách hàng cảm thấy hài lòng nhất và hai bên sẽ tiến hành ký nghiệm thu giao diện này. Đây là giao diện chính thức sử dụng cho toàn bộ Website và khách hàng không được tự ý thay đổi nữa. Hoặc nếu vì một lý do thỏa đáng nào đó, hai bên sẽ thỏa thuận lại bảng hợp đồng thiết kế cho phù hợp với yêu cầu hiện tại của khách hàng.

*** Tiến hành thiết kế Website và lập trình**

Sử dụng giao diện đã được nghiệm thu để thiết kế và tiến hành lập trình. Lập trình đúng nội dung hợp đồng soạn thảo, sẽ hạn chế thay đổi khi đã lập trình xong. Nếu khách hàng có nhu cầu sửa đổi bổ sung về mặt nội dung lập trình.

*** Hoàn thiện website**

Hoàn thiện các tính năng chi tiết của website, cập nhật nội dung cơ bản. Khách hàng kiểm tra toàn bộ tính năng của website. Website sẽ được chạy thử nghiệm

trên Internet, nếu Quý khách hàng không có yêu cầu sửa chữa, bổ sung hay thắc mắc gì, hai bên sẽ tiến hành nghiệm thu hợp đồng.

Tuỳ theo quy mô và nhu cầu của từng cá nhân và doanh nghiệp, các website sẽ có các chức năng, và độ phức tạp khác nhau.

Bước 5: Hướng dẫn sử dụng và quản lý website

Hướng dẫn nhân viên của Quý khách cách thức sử dụng Website dễ dàng và thuận tiện nhất chỉ với kiến thức tin học văn phòng (kèm theo tài liệu Hướng dẫn sử dụng). Sau khi bàn giao, Quý khách toàn quyền sử dụng và quản lý Website.

Bước 6: Bảo trì website

Website sau khi được xây dựng, cần thường xuyên cập nhật thông tin để đảm bảo độ tươi mới của nó. Qua đó, khách hàng có thể thấy được tình hình phát triển của công.

Bước 7: Quảng bá website

Để website hoạt động có hiệu quả nhất, ngoài việc in địa chỉ website trên danh thiếp của công ty, giới thiệu với bạn bè và người thân, bạn còn có thể quảng bá trên các phương tiện thông tin đại chúng, các mạng thương mại, báo chí và các trang web điện tử. Đưa website lên các công cụ tìm kiếm Google, Yahoo, MSN,...

2. Các công cụ xây dựng website

Hiện nay có rất nhiều công cụ xây dựng website mã nguồn mở như WordPress, Vbulletin, DotNetNuke, ...

WordPress là một phần mềm nguồn mở (Open Source Software) được viết bằng ngôn ngữ lập trình website PHP (Hypertext Preprocessor) và sử dụng hệ quản trị cơ sở dữ liệu MySQL. WordPress được ra mắt lần đầu tiên vào ngày 27/5/2003 bởi tác giả Matt Mullenweg và Mike Little. Hiện nay WordPress được sở hữu và phát triển bởi công ty Automattic có trụ sở tại San Francisco, California thuộc hợp chủng quốc Hoa Kỳ.

WordPress là một mã nguồn mở bằng ngôn ngữ PHP để hỗ trợ tạo blog cá nhân, và nó được rất nhiều người sử dụng ủng hộ về tính dễ sử dụng, nhiều tính năng hữu ích. Qua thời gian, số lượng người sử dụng tăng lên, các cộng tác viên là những lập trình viên cũng tham gia đông đảo để phát triển mã nguồn WordPress có thêm những tính năng tuyệt vời. Và cho đến thời điểm viết bài này là 2015, WordPress đã được xem như là một hệ quản trị nội dung (CMS – Content Management System) vượt trội để hỗ trợ người dùng tạo ra nhiều thể loại website khác nhau như blog, website tin tức/tạp chí, giới thiệu doanh nghiệp, bán hàng – thương mại điện tử, thậm chí với các loại website có độ phức tạp cao như đặt

phòng khách sạn, thuê xe, đăng dự án bất động sản,…vâng…vâng…Hầu như mọi hình thức website với quy mô nhỏ và vừa đều có thể triển khai trên nền tảng WordPress.

vBulletin là một "gói mã nguồn" (source-code) cài đặt forum, tôi dùng từ "gói" thay cho từ "phần mềm" vì thực chất là chúng ta không cài đặt lên máy tính của mình 1 chương trình, mà "gói" này cho phép chúng ta "cài đặt" lên hosting-server 1 forum. vBulletin được viết bằng PHP và sử dụng cơ sở dữ liệu chủ yếu là MySQL.

– vBulletin không phải là một sản phẩm free, được cung cấp bởi hãng Jelsoft Enterprises Limited; người ta phải trả tiền mua bản quyền để download "gói" cài đặt… Tuy nhiên, chũng ta đều đã quen xài lậu, nếu bạn định phát triển diễn dàn lớn mạnh, lâu dài… bạn nên MUA bản quyền.

– Vì tình trạng xài lậu khá nhiều, nên nhiều nhà cung cấp dịch vụ hosting-server (nhà cung cấp máy chủ lưu trang web/diễn đàn) từ chối không chấp nhận vBulletin. Các bạn nên chú ý điều này, nhất là các dịch vụ host FREE.

DotNetNuke là một hệ thống quản lý nội dung mã nguồn mở viết bằng ngôn ngữ lập trình VB.NET trên nền tảng ASP.NET. Tuy nhiên, các nhà phát triển đã bắt đầu chuyển DotNetNuke core trên nền C#. Đây là một hệ thống mở, tùy biến dựa trên skin và module. DotNetNuke có thể được sử dụng để tạo các trang web cộng đồng một cách dễ dàng và nhanh chóng.

DotNetNuke được phát triển dựa trên cổng điện tử IBuySpy được Microsoft giới thiệu như là một ứng dụng mẫu dựa trên nền tảng .NET Framework. DotNetNuke đã được phát triển qua nhiều phiên bản và hiện nay đang ở phiên bản 9.x Phiên bản hiện tại đòi hỏi.NET Framework 4.x trong khi các phiên bản trước đó chạy trên nền.NET Framework 1.1.

Một cài đặt DotNetNuke có thể phục vụ nhiều trang web, mỗi trang có thể có giao diện và cộng đồng người sử dụng riêng biệt. Phiên bản hiện tại cũng cho phép chức năng đa ngôn ngữ.

Tên DotNetNuke đã được đặt để phản ánh nền tảng .NET mà nó sử dụng đồng thời theo trào lưu của một số cổng điện tử khá thành công trước đó như PHP-Nuke và PostNuke. Tên này cũng đã được công ty Perpetual Motion Interactive Systems Inc. Đăng ký bảo hộ vào ngày 15 tháng 11 năm 2004 (TMA 625.364). Shaun Walker, chủ tịch của công ty này, chính là người tạo ra và duy trì DotNetNuke.

DotNetNuke có một nhân cơ bản có thể mở rộng với các module để bổ sung các tính năng mới; giao diện của hệ thống có thể tùy biến nhờ vào kỹ thuật skin.

3. Chọn công cụ và phiên bản

Với phạm vi của giáo trình này, chúng tôi chọn công cụ DotNetNuke làm công cụ giảng dạy và tham khảo cho học sinh, sinh viên.

Để nắm được nguyên tắc cơ bản để có nền tảng sử dụng các phiên bản cao hơn, bài này sẽ giới thiệu cách sử dụng mã nguồn mở DotNetNuke 7.x để xây dựng và phát triển website.

4. Cài đặt công cụ lên máy chủ web cục bộ

B1. Cài Internet Information Services (IIS)

B2. Download và cài đặt .NET Framework

B3. Cài đặt Microsoft SQL Server 2008 trở lên

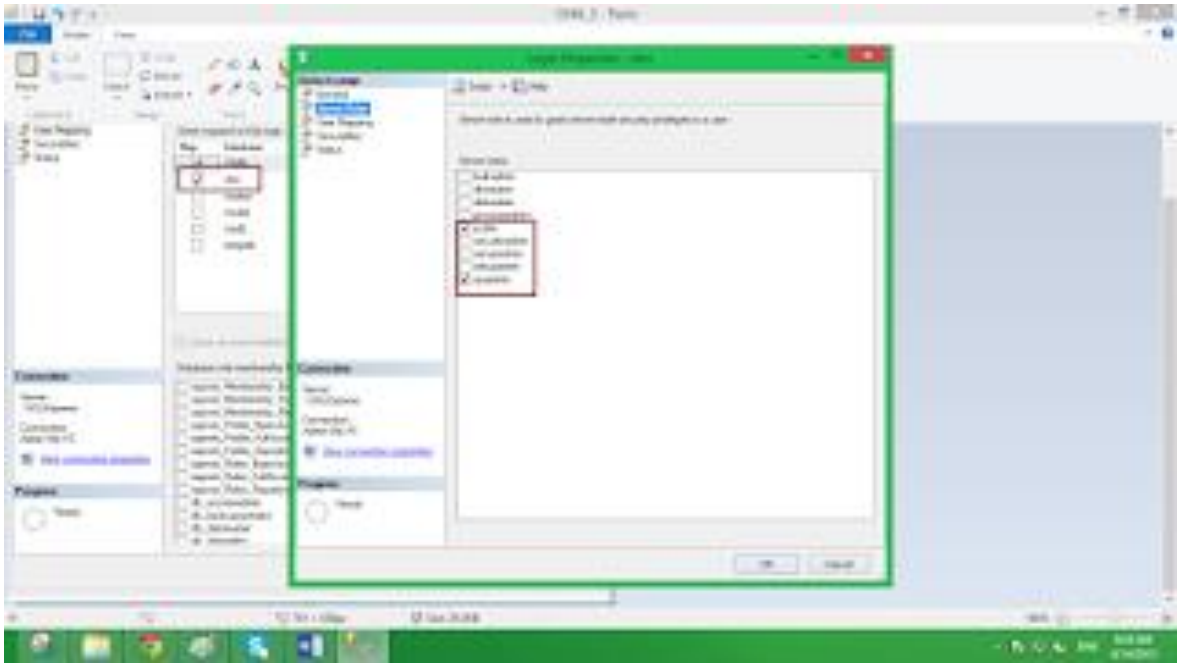
B4. Cài DotNetNuke

Sau khi tiến hành cài đặt IIS và SQLSever lên máy tính, bước tiếp theo là cài đặt DNN.

Bước 1: tạo cơ sở dữ liệu và tài khoản người dùng.

Sử dụng SQLSever tạo 1 cơ sở dữ liệu có tên là DNN và 1 người dùng có tên đăng nhập là: dnn và mật khẩu là: dnn. Tên cở sở dữ liệu, tên đăng nhập và mật khẩu có thể tùy chọn.

Cấp quyền cho tài khoản:



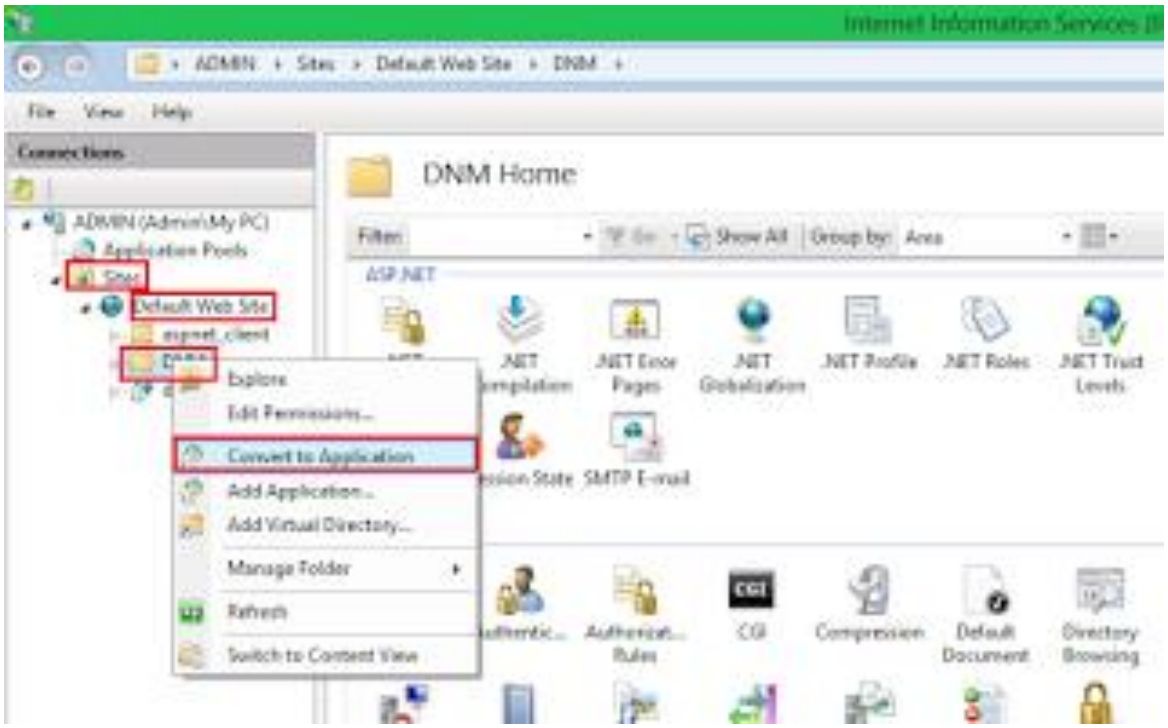
Bước 2: Tải mã nguồn

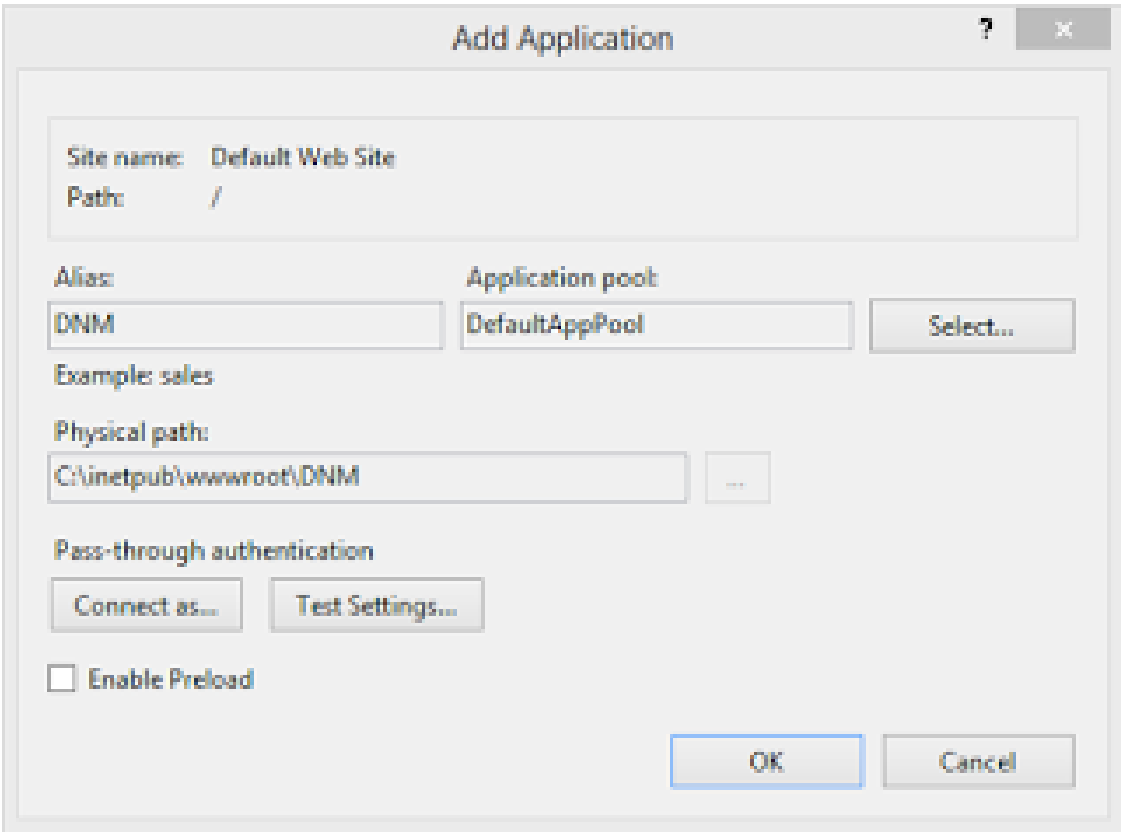
Địa chỉ tải xuống mã nguồn là:

http://dotnetnuke.codeplex.com/releases/view/105676, ở địa chỉ này có đầy đủ các phiên bản và các loại mã nguồn của DNN. Chọn phiên bản thích hợp sau đó tải về.

Bước 3: Giải nén, đổi tên thư mục và copy thư mục mã nguồn vào thư mục: "C:\inetpub\wwwroot"

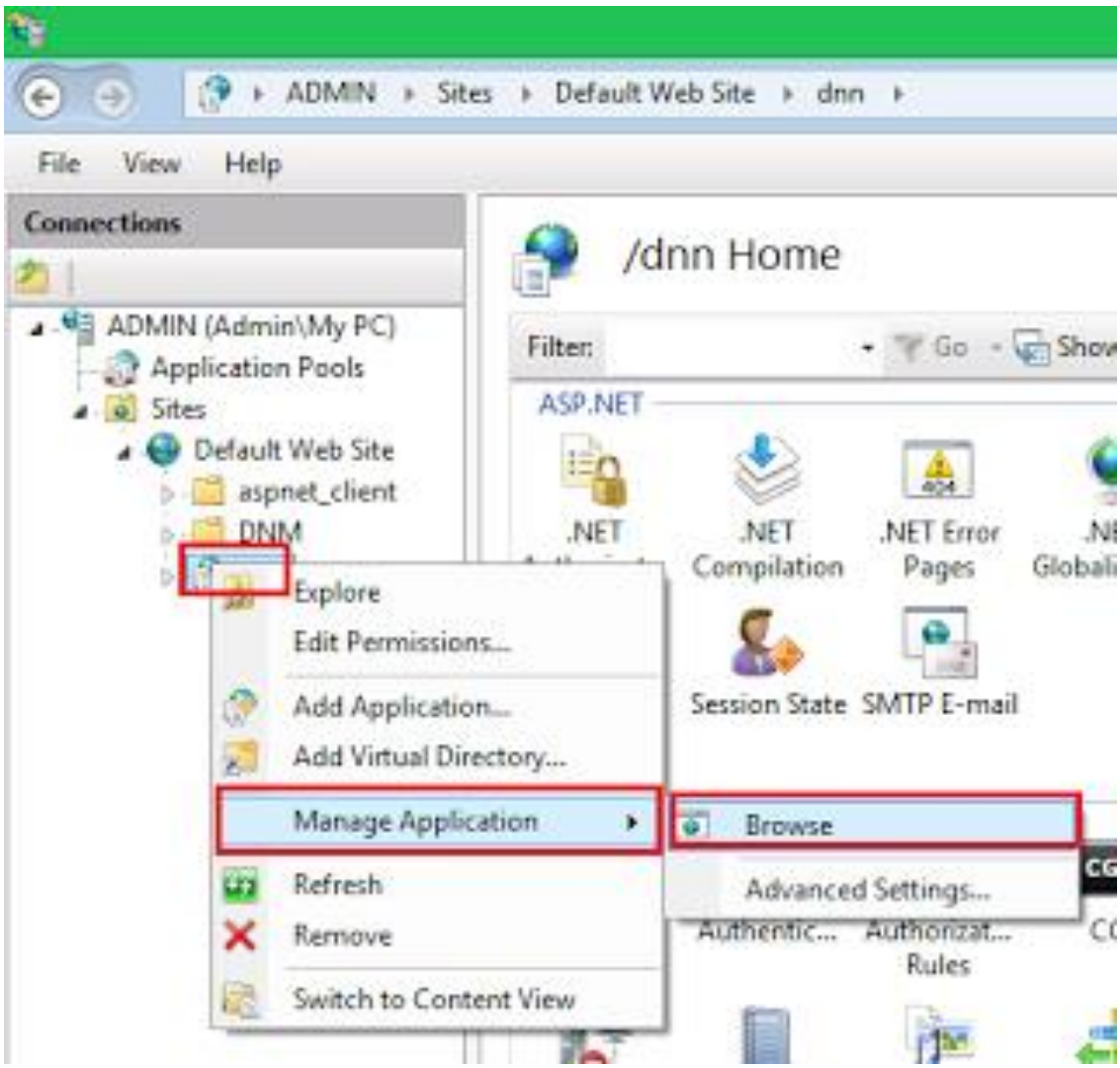
Bước 4: Bật IIS lên: IIS->Sites->Default web sites: sẽ thấy thư mục chúng ta vừa copy vào "C:\inetpub\wwwroot" ấn chuột phải lên thư mục.





Sau khi nhấn "OK" biểu tưởng của thư mục sẽ thay đổi.

Tiếp tục ấn chuột phải vào thư mục-> Managen Application-> Browse


ADMIN ▸ Sites ▸ Default Web Site ▸ dnn ▸
File View Help
Connections
ADMIN (Admin\My PC)
Application Pools
Sites
Default Web Site
aspnet_client
DNN
/dnn Home
Filter:
ASP.NET
.NET
.NET Compilation
.NET Error Pages
Session State
SMTP E-mail
Explore
Edit Permissions...
Add Application...
Add Virtual Directory...
Manage Application
Refresh
Remove
Switch to Content View
Browse
Advanced Settings...
Authentic...
Authorizat... Rules

Kết quả:

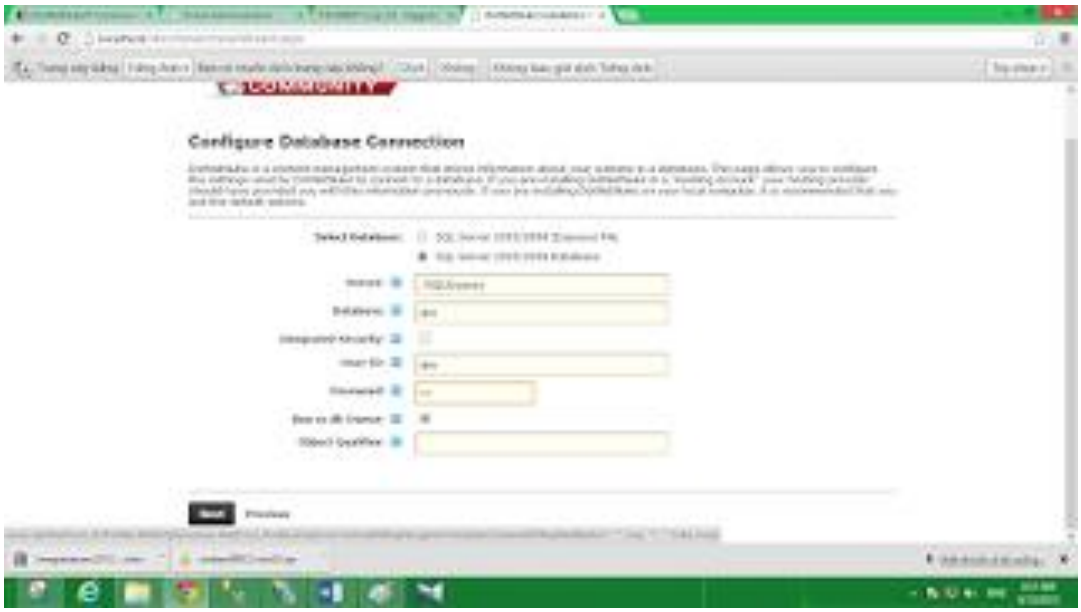
DotNetNuke Installation Wizard - Version 06.02.07

Tiếp theo chọn ngôn ngữ và ấn “Next”.

DNN kiểm tra các quyền được cấp cho thư mục chứa mã nguồn, nếu 1 trong các quyền chưa được cấp thì tiến hành cấp quyền cho DNN.

Sau đó ấn “Next”

Bỏ dấu tích trong ô “Intrgrated Security”:



Nhập tên cơ sở dữ liệu, tên đăng nhập và mật khẩu, các thông tin này đã được tạo từ các bước trên.

Sau đó ấn “Next”: quá trình cài đặt bắt đầu

Chờ quá trình xong sau đó ấn “Next”:

Nhập các thông tin tài khoản người quản trị.

5. Tùy chỉnh các thông số cơ bản

Tùy chọn modules cho hệ thống.

Tùy chọn ngôn ngữ cho hệ thống.

Tùy chọn Skins và containers

Sau khi tiến hành các bước, thu được kết quả như sau:

BÀI 8: CÁC MÔ-ĐUN BỔ SUNG CHỨC NĂNG CHO WEBSITE

Mã Bài: MĐ19_08

Giới thiệu:

Để thành thạo với việc sử dụng công cụ DotNetNuke để thiết kế và phát triển website, trong bài này chúng tôi giới thiệu chi tiết một số mô đun cơ bản và cách cài các mô đun này vào website DotNetNuke.

Mục tiêu:

- Xác định được danh sách chức năng cung cấp sẵn của công cụ xây dựng website đang dùng;
- Phân tích xác định được danh sách chức năng của website;
- Tìm kiếm và xác định được các mô-đun bổ sung chức năng cho công cụ;
- Cài đặt tích hợp được các mô-đun bổ sung chức năng vào công cụ;
- Thận trọng, tỉ mỉ trong quá trình làm việc;

Nội dung chính:

1. Danh mục chức năng của website

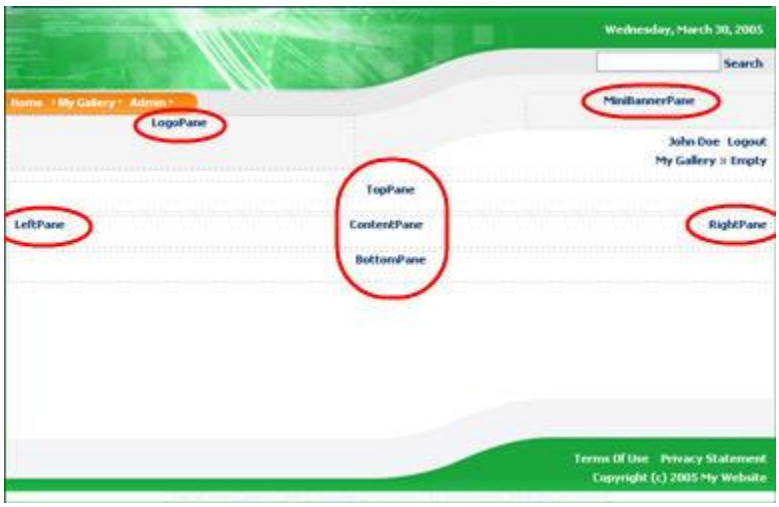
Website được tạo từ 1 hoặc nhiều trang. Nhưng những trang đó không hiển thị thông tin để người truy cập vào xem. Đó chính là vai trò của Modules.

Modules cung cấp những một vài thông tin hoặc chức năng của website mà đó là cách kết hợp duy nhất để làm trang web đặc biệt. Modules có thể đơn giản là một đoạn văn bản hoặc hình ảnh, hoặc có thể phức tạp như để tính phí bảo hiểm hoặc thiết thực là nói chuyện trực tuyến.

Sự điều hướng: bạn đặt Module ở đâu?

Các Module nằm trong các ô cửa ở trên trang. Các ô cửa đó chính là các phần được xác định trước ở trên trang là cái giúp chúng duy trì khi xuất hiện. Mỗi trang Skin có ít nhất là một cửa sổ. Một Skin có thể có vài cửa sổ để dành cho những mục đích đặc biệt.(thí dụ: logo hoặc banner nằm ở trên đầu của một website).

Modules



Nội dung: Module có những dạng nào?

Dot Net Nuke® được định dạng với những con số chuẩn của **Module.** Có nhiều loại module dùng được với nhiều loại nguồn khác nhau (cả miễn phí và buôn bán). Danh sách ngắn gọn ở dưới mô tả các đinh dạng của module.

Tài khoản đăng nhập cho phép người sử dụng đăng nhập vào portal của bạn.

Announcements của nhà sản xuất là danh sách của các đoạn văn bản thông báo bao gồm tiêu đề và hướng dẫn ngắn gọn.

Banner cung cấp cho website những quảng cáo lặt vặt. Bạn có thể chọn số lượng banner hiển thị cũng như là dạng banner.

Blog weblog cho quyền người truy cập sử dụng blog.

Contacts hiển thị nội dung thông tin cho một nhóm người như là một nhóm đề án.

Discussion danh sách một nhóm thông điệp được gửi lên cho một chủ đề cụ thể.

Document hiển thị danh sách các tài liệu, bao gồm các đường liên kết được duyệt qua hay tải về.

Events hiển thị danh sách các sự kiện được đưa lên, bao gồm thời gian và vị trí. Danh sách các sự kiện này có thể được thiết lập để thay phiên nhau hoàn trả trong định dạng lịch biểu.

FAQs cho phép bạn quản lý danh sách các câu hỏi thường xuyên và những câu trả lời của họ.

Feedback cho phép những người truy cập gởi những thông điệp đến quản trị viên của website. Nếu như người truy cấp đã đăng nhập vào, tên bạn sẽ tự động

Forum là nơi một nhóm người tạo ra và quản lý thảo luận và đáp lạ của các chủ đề. Điều tiết người sử dụng, các avatar, biểu tượng mặt cười và những email thông báo đều được ủng hộ.

Gallery hiển thị các album đồ họa và các tập tin đa phương tiện.

IFrame là nét đặc biệt của phần duyệt internet Explorer nó cho phép hiển thị nội dung từ trang web khác trong module của bạn.

Image hiển thỉ những image chỉ định. Đơn giản là module sẽ đề cập tới những image có liên quan hoặc URL tuyệt đối, để cho những file image không cần phải trú ngự trong trang web của bạn.

Links hiển thị danh sách của các đường siêu liên kết đến bất cứ trang nào, image hoặc là một file nào trên website của bạn, hoặc đến một tập tin, image hoặc là một trang của website khác.

New Feed (RSS**)** hiển thị những thông tin mới được cung cấp trên trang của bạn. Nó cung cấp cho người truy cập cập nhật được những thông tin mới nhất, những thông tin của đề tài trên bất cứ lãnh vực nào của chủ đề.

Search Input có thể hiển thị trên bất cứ trang nào.

Tìm kiếm kết quả có thể hiển thị trên bất cứ trang nào.

Text/HTML cho phép bạn thiết kế nội dung của một đoạn văn bản, hoặc thêm html. Bao gồm hình ành của tất cả image bạn tải lên.

Tải khỏan của người sử dụng module cho phép người đăng ký sử dụng thêm, sửa chữa và cập nhật thêm những thông tin chi tiết vào User Account. Các dịch vụ của hội viên sẽ được quản lý tại đây.

User Defined Table cho phép bạn tạo bảng dữ liệu tùy thích cho việc quản lý danh sách các thông tin. Những dữ liệu được sắp xếp theo thứ tự A-Z hoặc Z-A khi click vào tiêu đề từng cột.

XML/XSL hiển thị kết quả của sự chuyển đổi XML/XSL. Các tập tin XML và XSL được nhận dạng bởi đường dẫn UNC trong đặc tính xmlsrc và xslsrc của module.

Sự bảo mật: Ai là người có thể nhìn thấy và thay đổi Module?

Thiết lập sự bảo mật cho **Module** cũng giống như của trang. Bạn phải chắc rằng là bạn không muốn bất kì ai nhìn thấy những thông tin module hiển thị trên website của bạn. Nếu website của bạn cho phép người truy cập đăng ký, bạn sẽ cho ngươi sử dụng những sự cho phép đặc biệt. Bạn có thể cho hoặc vô hiêu hóa người sử dụng truy cập vào những module chỉ định mà chỉ có bạn mới có thê thấy hết toàn bộ trang.

Module có sự lựa chọn đặc biệt là đến **Inherit View permissions from Page**. Đơn giản là đưa cho module cùng sự cho phép như là những trang đã chỉ định. Nếu trang cho phép thay đổi thì module cũng cho phép thay đổi chúng.

Hình thức: module nhìn như thế nào?

Module hiển thị nội dung theo chức năng. Vì thế nội dung của **Contacts** module khác với nội dung của module **Feedback**. Để giúp cho người truy cập có thể phân biệt được nội dung và những chức năng thông thường(như phần thiết lập và giúp đỡ), module có containers (hoặc trình bao bọc) là cái mà giúp chúng xuất hiện nhất quán với nhau.

Có vài mẫu **Skin**, người thiết kế web có thể cung cấp một hoặc nhiều chức năng của **Containe**r. Bất cứ container nào cũng có thể sử dụng với bất cứ Skin nào, dù họ thường xuyên thiết kế để nó phù hợp với skin.

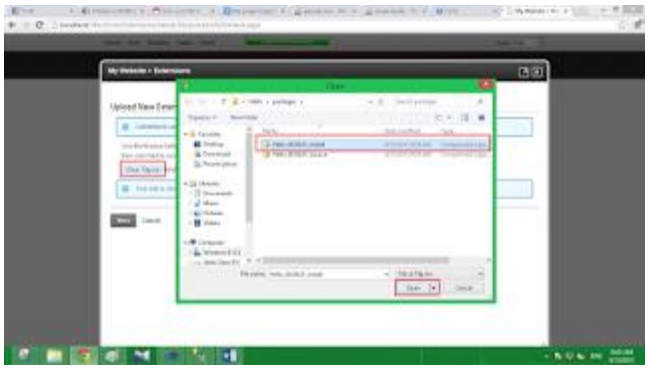
2. Mô-đun bổ sung chức năng

Để cài các mô đun vào website để sử dụng, trước tiên phải đăng nhập vào trang web với toàn khoản quản trị, sau đó:

Vào link: host->Extension:

Chọn Manage->Intall extension wizad:

Tiếp theo ấn chọn " Chọn tệp tin" sau đó duyệt tới 2 file zip đã được build từ các bước phía trên.



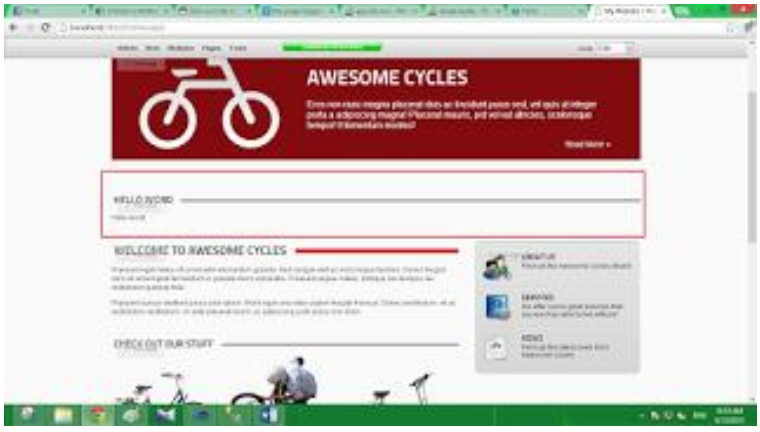
Tiếp theo ấn Next, quá trình cài modules được băt đầu:

Như vậy là module đã được cài đặt vào trang web.

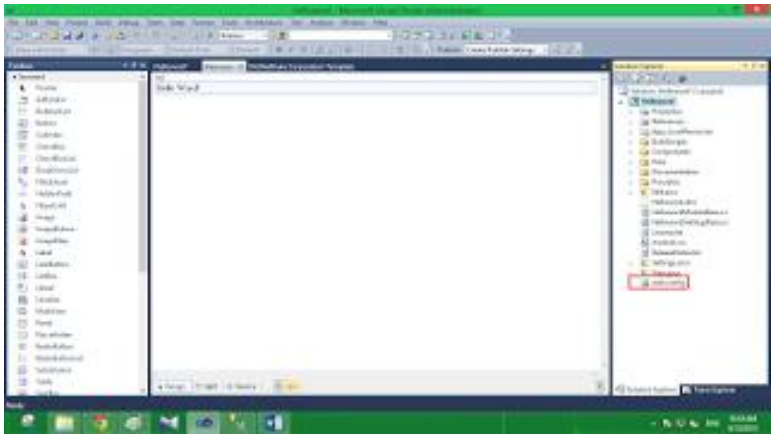
Tiếp theo là quá trình sử dụng module hay là gọi module ra:

Vào trang mà muốn sử dụng module, tiến hành các bước sau và chọn modules muốn sử dụng và ấn Add Module

Kết quả sẽ được như sau:

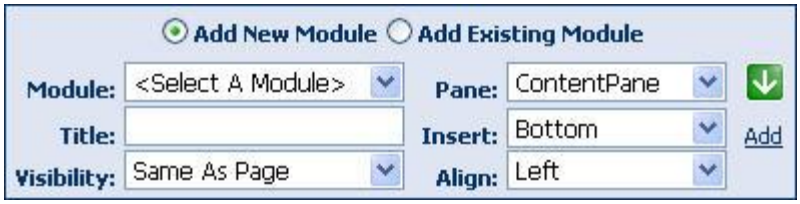


Chú ý nếu trong module tự tạo file web.config thì phải xóa file web.config nếu không sẽ gây lỗi cho module.



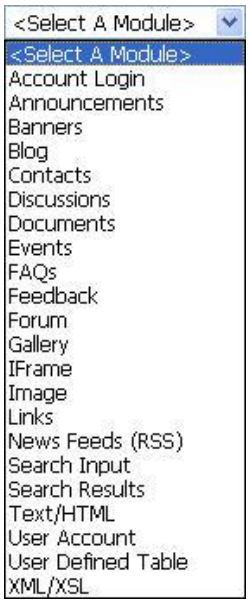
3. Tích hợp các mô-đun bổ sung chức năng vào website

*** Thêm mô đun đã cài đặt vào trang:**



Cách thêm module vào trang này chỉ đòi hỏi 2 bước.

1) Chọn Module từ hộp danh sách rơi xuống.



2) Click vào nút Add trên control Panel.

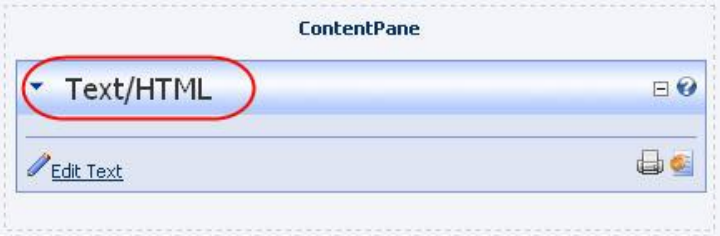
Module sẽ được thêm vào với những thiết lập mặc định. (được hiển thị trong control panel)

Các bước thực hiện: với tất cả các lựa chọn

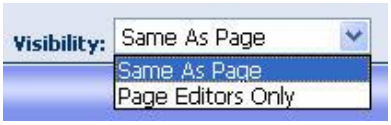
Như một sự lựa chọn, bạn có thể thay đổi các thiết lập khác trong control panel trước khi thêm trang. Mội một thiết lập đều có thể được lựa chọn.

1) Chọn Module từ hộp danh sách rơi xuống.

2) Đặt con trỏ chuột vào vùng tiêu đề và đánh tên của Module vào. Tên này sẽ xuất hiện trong phần tiêu đề của module container khi bạn thêm trang(như minh họa). Nếu vùng đó ở phía bên trái, thì đó là tên của module (thí dụ ” Text/HTML” sẽ đổi).



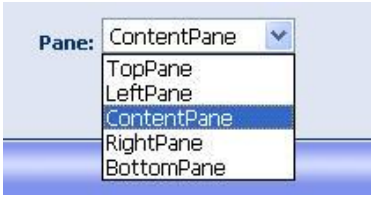
3) Chọn Visibility của trang từ hộp danh sách. Ở đây chỉ có 2 sự lựa chọn.



Same AS Page có nghĩa là Module sẽ hiển thị ở bất cứ website nào thì người truy cập cũng có thể thấy trang đó. Nếu trang đó đã hiển thị rộng rãi thì module của bạn cũng sẽ hiển thị ngay lập tức.

Nếu module mới đòi hỏi cấu hình mà bạn chọn phải là Page Editors Only. Giữ cho module không hiển thị để không cho phép bất cứ người truy cập nào có thể vào đó sửa đổi. Tình trạng hiển thị có thể bị thay đổi trong Module settings khi nó đã sẵn sàng.

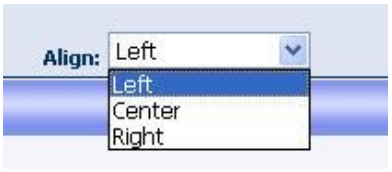
4) Chọn ô cửa của Skin là nơi mà bạn sẽ thêm Module vào. Nội dung của hộp danh sách rơi xuống này có thể khác với Skin khác. Nội dung của ô cửa được chỉ định bởi những người thiết kế skin.



Theo minh họa dưới đây thì nó sẽ thể hiện Skin với những ô cửa khác nhau ở những đủ mọi vị trí và kích cỡ khác nhau. Mộ vài ô cửa sẽ có kích cỡ phù hợp và nó để dành cho những mục đích cụ thể.(như là hiển thị của "minibanner" hoặc logo).

5) Chọn nơi mà muốn chèn module vào (ở trên đầu hay là ở giữa). module hiển thị trong cột của ô cửa. Nếu module đã nằm trong ô cửa rồi, thì module mới sẽ được thêm vào ở trên đầu hoặc là ở giữa (như lý thuyết). Nếu cần thiết Module có thể được di chuyển sau khi thêm vào trang.

6) Chọn cách mà nội dung sắp xếp trong module (trái, giữa, phải). Một vài container không cho phép sắp xếp.



7) Chọn Add New Module (mặc định) hoặc Add Existing Module.

Nếu bạn chọn thêm vào module hiện hành, trang sẽ refresh, module và Title của vùng sẽ được thay thế trong Control Panel với việc người chọn chỉ định trang và module hiện hành. Đây là chức năng sao chép từ module hiện hành. Khi mà nội dung được cập nhật nguyên bản hay sao chep, thì tất cả sự thay đổi được sao chép lại.

8) Click vào nút Add trên Control Panel.

*** Di chuyển module**

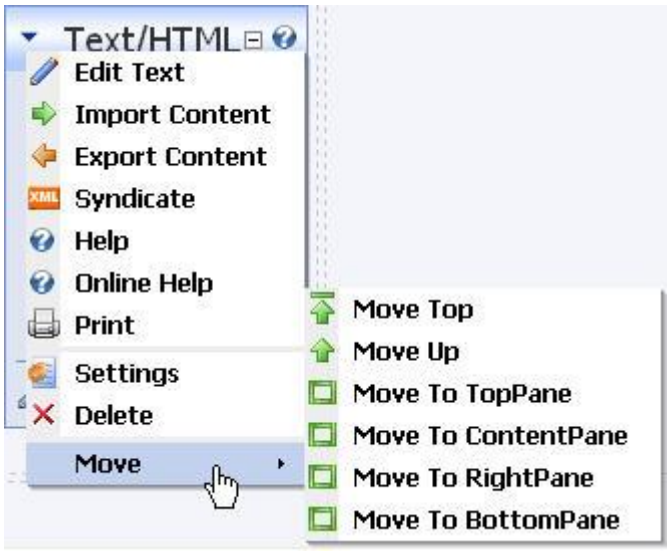
Bạn có thể di chuyển module từ vị trí này sang vị trí khác ở trên trang. Có thể sẽ thay đổi thứ tự của module trong ô cửa hoặc di chuyển module từ ô cửa này sang ô cửa khác. Bạn còn có thể di chuyển module từ trang này sang trang khác.

Có 2 cách khác nhau để di chuyển module. Bạn có thể dùng lệnh trong Module Actions Menu hoặc là bạn dùng Drag&Drop.

Các bước thực hiện việc di chuyển module trong Module Actions Menu

Hầu như trong bất cứ module Container nào cũng sẵn có Module Actions Menu. Chúng ta có thể sử dụng lệnh để thay đổi vị trí của hai module Text/HTML trên trang.

1) Đưa con chuột vào Module Action Menu và cuộn xuống mục menu Move. Lệnh Move to luôn luôn hiện diện và có thể khác với skin khác. Chỉ có thể dùng Move to cho mỗi ô cửa của Skin.



Lệnh di chuyển phụ thuộc vào sự thay đổi vị trí của module. Từ ví dụ đặt làvào giữa cột, không có sự lựa chọn di chuyển xuống (chỉ có di chuyển lên).

2) Chọn mục Menu để nhận biết nơi muốn di chuyển Module đến đó.

Các bước để thực hiện việc di chuyển bằng Drag& Drop

Hầu như trong bất cứ module nào cũng sẵn có Drag&Drop nơi mà tiêu đề được hiển thị. Chúng ta sử dụng Drag&Drop để thay đổi vị trí của 2 module Text/HTML trên trang.

1) Nhấp chuột trái và giữ chuột cho hiện ra ở dưới Title của module. Bạn sẽ chú ý đến module được lộ ra xung quanh nó có đường viền ngoài. Đường viền ngoài chỉ rằng module sẵn sàng được di chuyển sang vị trí mới.

2) Giữ nút chuột trái, di chuyển Module sang ô cửa khác. Khi bạn vượt qua ô cửa khác thì nó sẽ hiện ở vị trí nổi bật nhất, ra dấu là module có thể thụt xuống ở vị trí mới.

3) Bỏ chuột ra khi ô cửa sáng lên. Ô cửa sẽ refresh lại và Module sẽ ở vị trí mới.

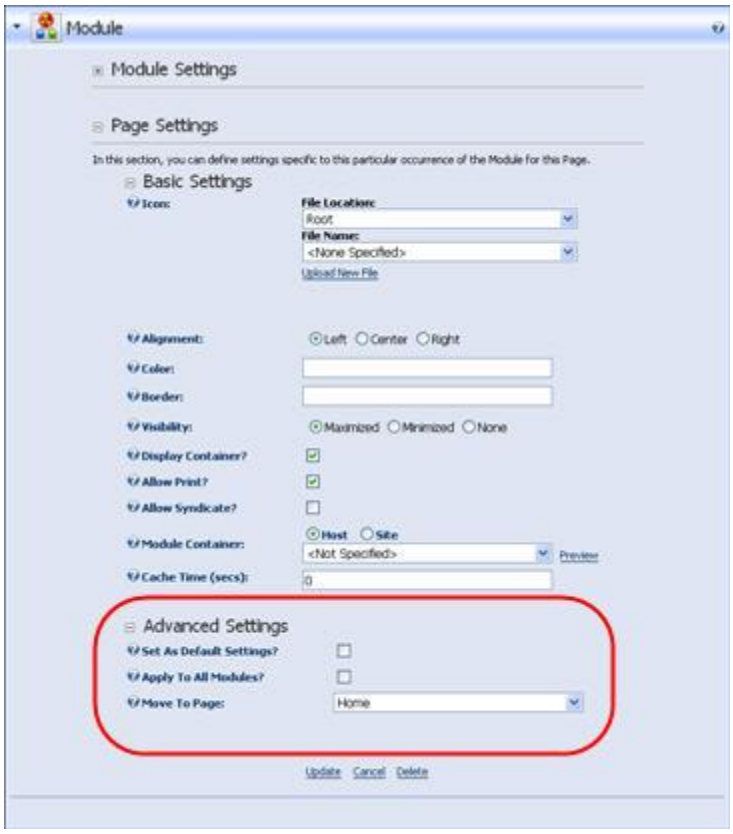
Các bước để thực hiện việc di chuyển sang trang khác

Di chuyển Module từ trang này sang trang khác thì phải làm xong từ các trang Module Settings.

1) Trỏ con chuột vào Module Actions Menu và di chuyển vào mục thiết lập. Như là một sự lựa chọn, nhấp vào Container có hiển thị biểu tượng cho Module Settings.

OR

2) Trên trang Module Settings, di chuyển xuống phần Advanced Settings.



3) Chọn trang từ hộp danh sách rơi xuống của Move to page để di chuyển Module.

4) Click vào update ở giữa trang Module Settings để di chuyển module sang trang mới (và để lưu lại các thay đổi này).

*** Xóa module**

Khi một Module không còn cần thiết nữa thì bạn có thể xóa chúng.

Các bước thực hiện trong Module Actions Menu

Hầu như bất cứ module Container nào cũng sẵn có Module Action Menu. Chúng ta sẽ dùng lệnh để xóa module.

1) Trỏ con chuột xuống Module Actions Menu và di chuyển xuống mục Delete và click chuột trái.

2) Click vào OK để xác nhận sự xóa module.

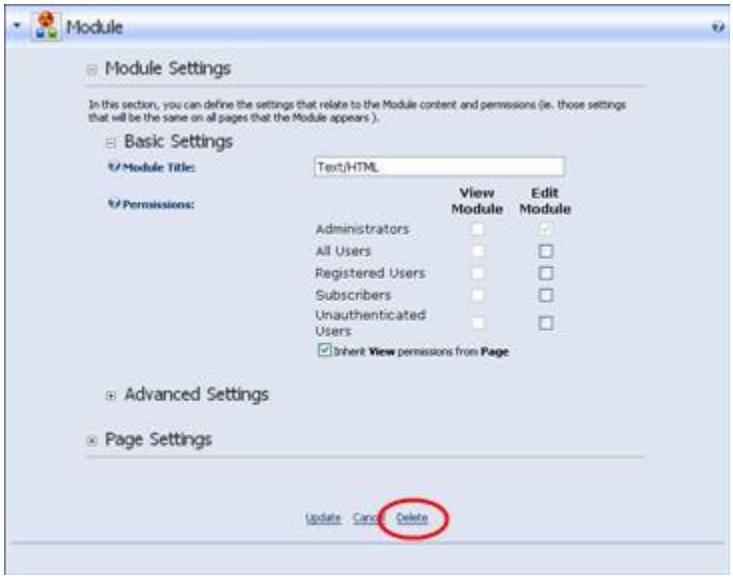
Các bước thực hiện trong module setting page

Bạn có thể xóa Module trong trang Module Settings.

1) Trỏ con chuột xuống Module Actions Menu và di chuyển xuống mục thiết lập. Như là một sự lựa chọn, các Container hiển thị biểu tượng cho Module Settings rồi click vào đó.

OR

2) Di chuyển vào giữa trang Module Settings và nhấp vào nút Delete.



3) Click vào OK để xác định là xóa module.

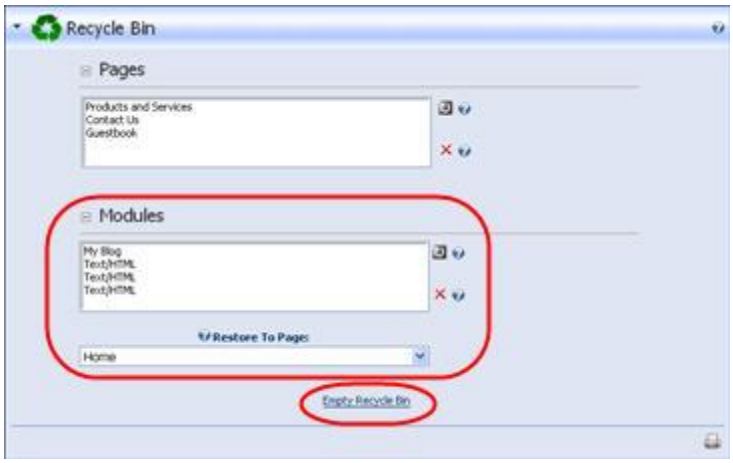
*** Phục hồi module bị xóa**

Thùng rác là nơi mà các Module (và Trang) sẽ đến đó khi bạn xóa chúng. Chúng ta có thể phục hồi từ thùng rác cho đến khi thùng rác trống. Phục hồi lại các module bị xóa cũng tương tự như phục hồi trang bị xóa.

Di chuyển chuột xuống menu Admin và chọn Recycle Bin.

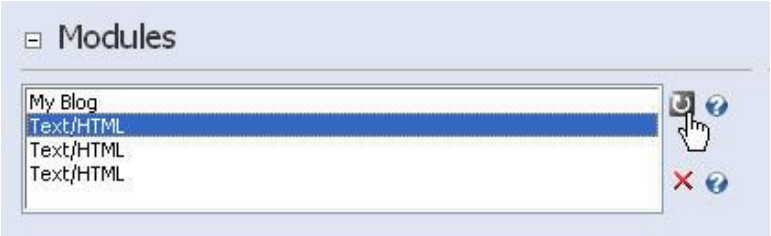
Các bước phục hồi module

1) Chọn Module mà bạn muốn Restore. Nội dung của danh sách đó là tất cả những module mà bạn đã xóa từ lúc trước.



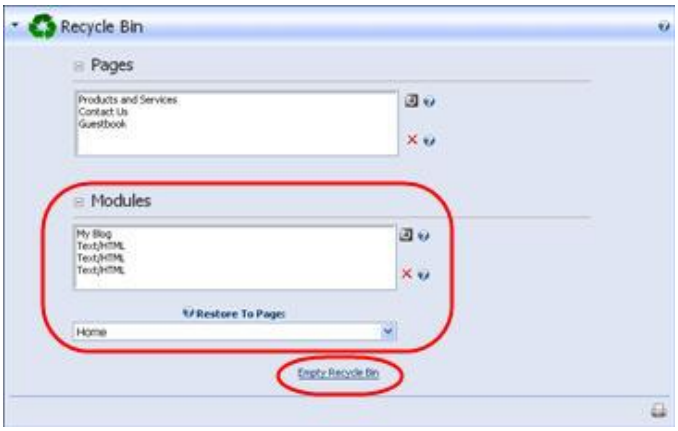
2) Chọn Page từ hộp danh sách rơi xuống nơi mà bạn muốn Module sẽ được hoàn lại. Module thêm vào sẽ được chỉ định vào nội dung của ô cửa mặc định.

3) Click vào nút Restore. Module của bạn sẽ được di chuyển từ danh sách các module của thừng rác(và nó sẽ ở tại nơi mà Trang đã chị định).

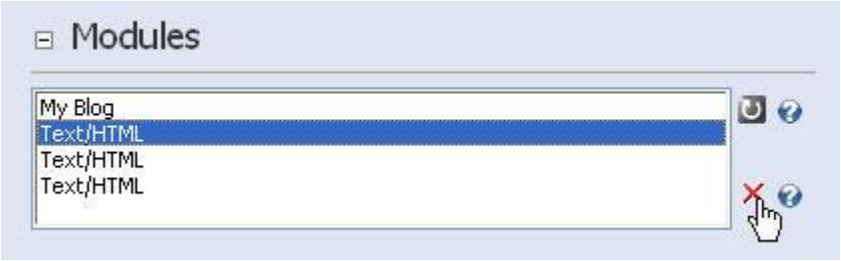


Các bước xóa trang (vĩnh viễn)

1) Chọn Module mà bạn muốn xóa đi vĩnh viễn. Nội dung của danh sách đó là những module mà bạn đã xóa từ lúc trước.



2) Click vào nút Delete trên phần Module.



3) Click vào OK để xác định xóa. Module của bạn sẽ di chuyển từ danh sách các module của thùng rác và bi xóa đi vĩnh viễn.

Các bước để làm trống thùng rác

Nếu thùng rác của bạn có quá nhiều rác, bạn muốn thùng rác trống để tiện sử dụng sau này. Làm trống thùng rác là bạn xóa đi vĩnh viễn Trang và Module ở trong đó.

1) Click vào Empty Recycle Bin nằm ở giữa trang.

2) Click vào OK để xác định là xóa hết tất cả các Trang và Module trong thùng rác . Chúng sẽ được di chuyển từ danh sách các trang và module trong thùng rác và bị xóa đi vĩnh viễn.

Sau đó thùng rác sẽ xuất hiện trống rỗng.

*** Phân quyền sử dụng module**

Đôi khi bạn không muốn ẩn các trang bằng cách phân quyền sử dụng chúng. Đôi khi bạn muốn trang đó hiển thị và được bảo vệ bởi một vài Module chứa nó. Đó thật sự là môt điều hữu dụng làm cho người truy cập website bạn rằng họ có thể đánh giá thêm thông tin nếu họ có đăng ký và sau đó họ đăng nhập. Nếu người truy cập website không đánh giá được các module riêng, thì đơn giản là họ sẽ không nhìn thấy được trang đó.

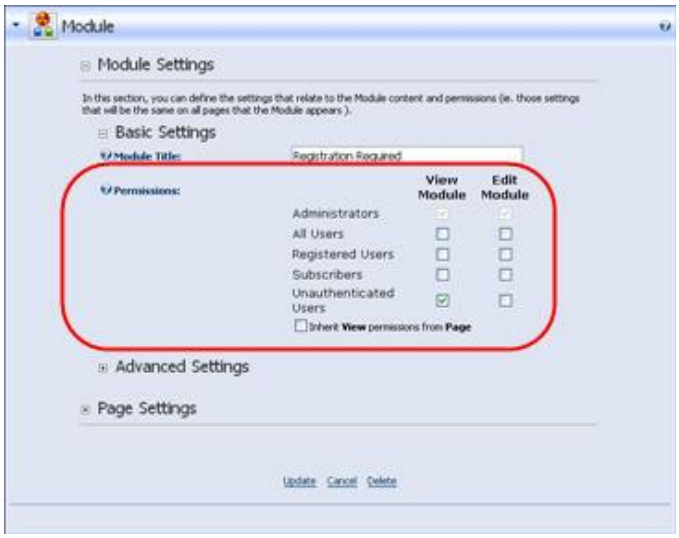
Trang này danh cho những người truy cập ẩn danh:

Và nó sẽ như thế này sau khi họ đăng nhập vào… chú ý rằng Module đầu tiên dường như "không xuất hiện" và một vài module mới có "xuất hiện". Đó là do việc thực hiện Module Permission.

Các bước thực hiện việc thiết lập Module Permissions

Chúng ta vừa sử dụng trang hình ảnh ở ví dụ trên. Các quản trị viên (đó là bạn) thấy được những điều hiển thị trên tất cả Module (vì thế bạn có thể thấy và thay đổi các thiết lập của bạn).

1) Click biểu tượng thiết lập Module hoặc chọn thiết lập từ menu Module Action để thấy thông điệp " registration required"(chúng ta làm module viewable chỉ dành cho những người không đăng nhập). Nó nằm ở cửa sổ đầu trang.



Check vào ☑ chấp nhận cho người sử dụng không xác định thấy module. Không check ☐ bất kì nhóm nào khác (chú ý quản trị viên luôn luôn được View Module permission).

Click vào Update ở giữa trang để lưu lại những sự thay đổi của bạn.

2) Bây giờ nhấp vào biểu tượng thiết lập Module hoặc chọn thiết lập từ menu Module Actions cho mỗi một module (chúng ta làm Module viewable này là chỉ dành cho những người sử dụng có đăng nhập vào). Bạn cần làm điều này cho mội một module trên trang của bạn nếu bạn chỉ muốn cho những người sử dụng có đăng ký dùng.

Check vào ☑Registered Users cho phép View Module . Không check vào ☐ bất cứ nhóm khác.

Click vào Update ở giữa trang để lưu lại những thay đổi của bạn.

Những thông tin thêm

Additional Information:

Có bốn cách khác nhau để đưa ra đánh giá về Module trên website của bạn. Ai là người có thể thấy được mỗi một module?

(1) Administrators: chỉ có người sử dụng này mới có quyền quản lý website của bạn, có thể thấy được các module (người đó là bạn)! Đây là một thiết lập nên dùng cho module khi mà bạn thiết lập một quá trình nhưng không muốn bất cứ ai có thể nhìn thấy cho đến khi bạn hoàn thành. Bạn có thể dùng chức năng Page Editor Only của Visibility selector khi bạn thêm module mới và giữ riêng chúng cho đến khi bạn đã sẵn sàng

Khi mà một Module chỉ hiển thị cho quản trị viên của website thấy nó sẽ viền đỏ để báo hiệu cho bạn. Bạn sẽ không quên cách này để không có ai có thể nhìn thấy ngoại trừ bạn.

(2) Unauthenticated Users: tất cả những người truy cập đều có thể thấy module ẩn của bạn. "Không xác thực" có nghĩa là những người sử dụng này không đăng nhập và có thể họ cũng không đăng ký như là người sử dụng website của bạn. Đánh giá đó là điều mang tính công chúng nên bạn muốn bất kỳ ai cũng có thể thấy được.

(3) Registered Users: người sử dụng này đã điền vào mẫu đăng ký trên website của bạn và bây giờ họ có tên sử dụng và mật mã và họ đăng nhập vào.

4) Nhóm người sử dụng tùy thích: đó là những người đăng ký sử dụng mà bạn đặt họ trong một nhóm đặc biệt hoặc tạo cho họ môt "vai trò" (thí dụ"gia đình" hoặc "khách hàng").

BÀI 9: QUẢN LÝ NỘI DUNG CỦA WEBSITE

Mã Bài: MĐ19_09

Giới thiệu:

Bài này chúng tôi sẽ trình bày cách sử dụng một số mô đun để đưa thông tin thiết kế website trên DotNetNuke.

Mục tiêu:

- Đưa được các nội dung cần thiết vào website;
- Quản lý được các mô-đun thông tin (mô-đun bổ sung chức năng) trên website;
- Thay đổi được giao diện website theo yêu cầu thiết kế;
- Thận trọng, tỉ mỉ trong quá trình làm việc;

Nội dung chính:

1. Làm việc với hệ quản trị của website

1.1. Thay đổi logo của website bạn

Trước khi bắt đầu, điều quan trọng là chúng ta phải hiểu **Logo** của bạn được hiển thị như là một chức năng **Skin** của website bạn. Nếu skin không hỗ trợ hiển thị cho logo, thì sau khi bạn thay đổi nó cũng sẽ không có hiệu quả! Nhưng hầu hết các mặc định của skin đều đến với DotNetNuke® (và nhiều thứ được tạo bởi cái khác) để thực hiện điều đó.

Đây là một ví dụ của **Skin** với phần **Logo** được làm nổi bật.

Đây là ví dụ của **Skin** không có hiển thị **Logo**.

Các bước thực hiện việc thay đổi logo của bạn

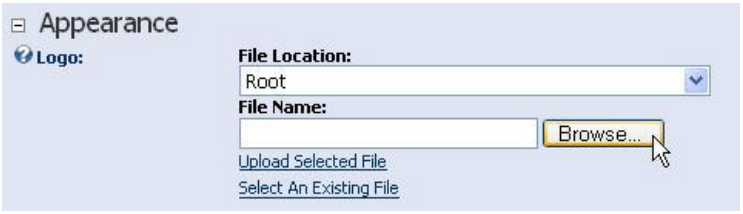
Logo là một thuộc tính của **Site Settings**, chỉ có **quản trị viên mới** có thể tới được nó trên website của bạn.

1. Chọn **Site Settings** từ menu Admin.

2. Trên trang **Site Settings**, di chuyển xuống phần **Appearance** và tìm phần thiết lập **Logo.**

3. Nếu như bạn đã tải tập tin **Logo** của bạn lên, đơn giản là chọn danh mục (vị trí tập tin) có nội dung của logo bạn từ hộp danh sách rơi xuống. Trang sẽ tự động làm mới lại trong khi **File Name** sẽ rơi xuống từ danh sách các tập tin của bạn, sau đó chọn tên tập tin.

Nếu như bạn cần tải lên một file mới, bạn có thể nhấp vào **Upload New File** và sau đó chon file nằm ở trên máy vi tính của bạn để tải lên.



Phải chắc rằng vị trí tập tin chỉ định để tải tập tin lên (**Root** là mặc định), nhấp vào **Browse** ở trên file trong máy vi tính của bạn và sau đó nhấp vào **Upload Sesected File** để di chuyển chúng từ máy vi tính bạn lên website ban.

4. Nhấp vào **Update** ở giữa trang để lưu lại những thay đổi của bạn! Logo mới của bạn bây giờ sẽ thay thế logo cũ.

1.2. Quản lý người sử dụng website của bạn

Ai là **Users** trên website của bạn? Thuật ngữ "User" được áp dụng cho bất kỳ người nào truy cập vào website của ban. Nhưng dó chỉ là hai dạng cơ bản để chỉ những người vào truy cập website của bạn.

Dạng người đầu tiên là người chỉ nhìn lướt xung quanh và không bao giờ đăng ký hoặc ít nhất là chưa bao giờ nhập vào hệ thống, những người đó gọi là **người sử dụng không xác thực**. Người sử dụng không xác thực là ẩn danh.

Dạng người thứ hai là người có đăng nhập vào. Để đăng nhập vào thì họ phải đăng ký từ trước và vì thế họ được đề cập tới như là những **người sử dụng có đăng ký**. Để nhận ra những người sử dụng có đăng ký đó là họ được biết và được bạn cho sự chấp thuận đặc biệt để xem hoặc thay đổi cái gì đó mà người sử dụng khác không thể làm được.

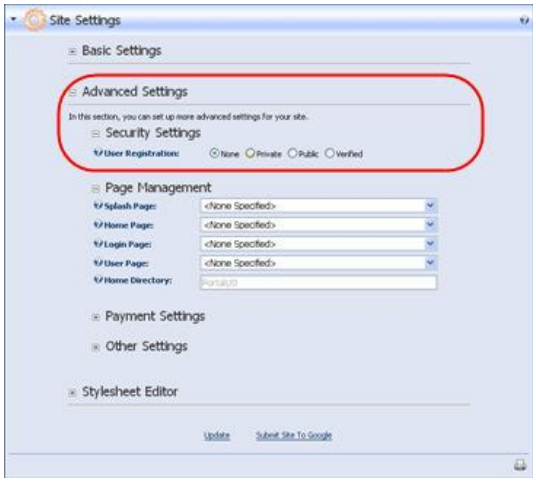
Đó không phải là sự đòi hỏi của website bạn yêu cầu người sử dụng phải đăng ký… đó là mộtlựa chọn mà bạn có thể chọn. Có thể bạn muốn có một website có nội dung dành cho tất cả những người truy cập vào. Nhưng bất kể điều gì thì phải có ít nhất một người sử dụng đăng ký vào website bạn…bạn(**quản trị viên**)!

Các bước thực hiện chọn dạng website đăng ký

Đầu tiên bạn muốn chọn dạng đăng ký mà bạn muốn sử dụng trên website của bạn. Quan trọng là cách lựa chọn, nhưng bạn không nên thay đổi chúng khi làm lần đầu tiên.

1. Chọn **Site Settings** từ menu Admin

2. Trên trang **Site Settings**. Di chuyển xuống phần **Advanced settings** và tìm phần thiết lập **User Registration**.



3. Chọn một trong những hộp option bằng cách nhấp vào nút thích hợp.

None: **Registration** là không có hộp lựa chọn sẵn có chỗ người truy cập. Nút **Login** vẫn hiển thị cho **quản trị viên** có thể đăng nhập vào; tuy nhiên Nút registration vẫn được giấu đi. Phần thiết lập thích hợp cho chỗ đó nó không bao gồm bất cứ nội dung .

Private: Những người truy cập cung cấp cho website đánh giá. Đánh giá của tất cả người sử dụng được giới hạn đến những người sử dụng ẩn danh cho đến khi bạn dứt khoát cho phép họ. Phần thiết lập thích hợp cho chỗ đó yêu cầu phải có sự chấp thuận của những đề nghị **Registration** (thí dụ website gia đình cá nhân chỉ mời những người bạn và những người họ hàng vào mà thôi).

Public: **Registration** tự động (và ngay lập tức) cho phép mà không cần có sự xác nhận hợp lệ nào. Phần thiết lập thích hợp cho chỗ đó đơn giản là muốn giữ cách đối xử mà không cần đòi hỏi bao gồm cả những thông tin hợp lệ.

Verified: **Registration** tạo ra mã xác minh bao gồm cả emai "welcome" gửi đến địa chỉ mà người sử dụng cung cấp. Có sự công nhận như vậy khi mà người sử dụng cung cấp mã xác minh khi họ đăng nhập lần đầu tiên. Quá trình này phải được bảo đảm rằng tất cả những **người sử dụng có đăng ký** đã cung cấp đúng địa chỉ email.

4. Nhấp vào **Update** ở giữa trang để lưu lại những thay đổi của bạn! Sơ đồ đăng ký của bạn bây giờ đã có hiệu lực.

Các bước để quản lý người sử dụng

Một khi bạn đã có người sử dụng, thì bạn có thể **thêm, cập nhật**, hoặc **xóa** họ.

1. Chọn **User Accounts** từ menu Admin.

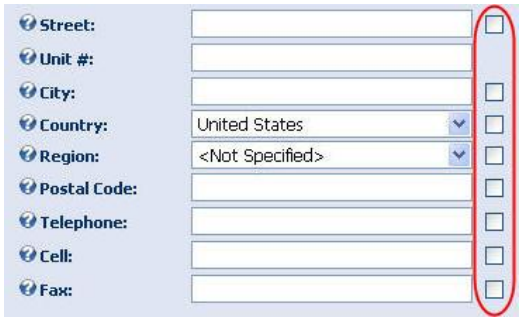
Từ trang **User Account** bạn có thể tìm thấy người sử dụng đang dùng bằng cách tra cứu **Username** hoặc **địa chỉ email** hoặc bạn có thể lập danh sách người sử dụng theo alpha. Danh sách này nhanh chóng đưa cho bạn thông tin người sử dụng bao gồm cả những người có hoặc không cho phép.

2. Nhấp vào biểu tượng cây bút chì kế bên **Username** để mở trang **Edit User Account** của người sử dụng.

Từ trang này có một số chức năng chọn lựa quản lý mà ban có thể áp dụng.

Cho phép người sử dụng bằng cách check vào **authorized** ☑ (hoặc UN-authorize bằng cách không check vào hộp☐. Gọi lại nếu bạn thiết lập sơ đồ đăng ký cá nhân, các bước cho phép được làm bằng tay đòi hỏi hoặc không khi người sử dụng bạn đăng nhập vào!

Require (hoặc có thể lựa chọn) ở vùng thông tin cho người **User Registration**. Nếu các hộp đều được check ☑ thì vùng này sẽ đòi hỏi khi người sử dụng đăng ký trên website bạn. Nếu các hộp đều không check ☐ thì vùng này sẽ được coi như là có thể lựa chọn.



Xác lập lại mật mã cho người sử dụng khi họ nhập vào bằng mật mã mới và xác định lại chúng bằng cách nhập lại lần nữa.

3. Nhấp vào **Update** ở giữa trang để lưu lại những thay đổi của bạn!

1.3. Tải tập tin lên website của bạn

Website của bạn được làm duy nhất bởi nội dung để người truy cập vào có thể tìm chúng ở đó. Bạn có thêm những dạng nội dung khác nhau thông qua các loại module (thí dụ những thông báo, đoạn văn bản, thông tin bao hàm, v..v.). Nhưng tại một vài điểm bạn sẽ muốn tải những tập tin của bạn lên đó như hình ảnh, banner, form yêu cầu, chứng từ kinh doanh,v..v.

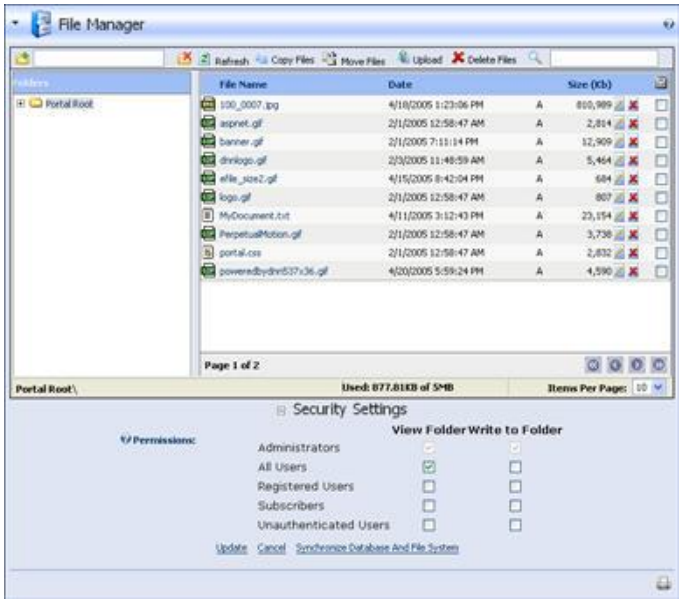
Bất cứ lúc nào bạn cũng cần phải chỉ rõ tập tin(như là khi bạn **thay đổi logo**) bạn sẽ nhìn thấy phần điều khiển nhỏ cho bạn tải tập tin lên. Đó là phần điều khiển bằng tay, nhưng nó đòi hỏi những công cụ mạnh để quản lý tập tin và thư mục gọi là **tập tin quản lý.**

Các bước thực hiện tải tập tin lên với phần File Manager

File manager cung cấp toàn bộ để bạn quản lý các tập tin và thư mục. Bạn có thể dùng nó để tải một tập tin mới lên website của bạn.

1. Chọn **File Manager** từ menu Admin.

Nó sẽ mở ra trang **File Manager**.

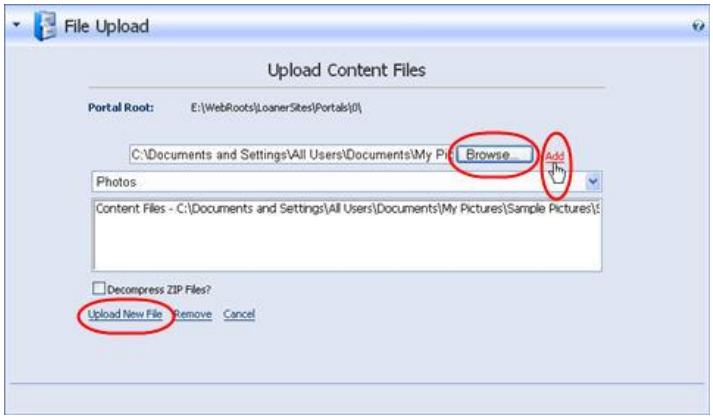


2. Tạo một **Folder** để tải tập tin của bạn lên. Các Folder sẽ giúp bạn tổ chức lại khi bạn có quá nhiều tập tin.

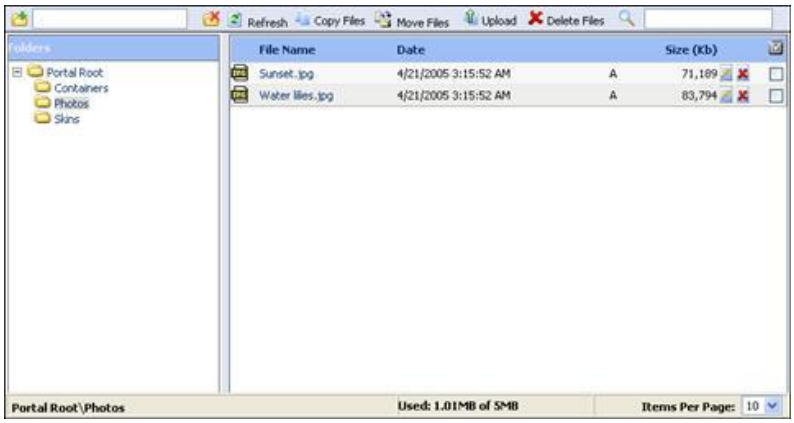
Nhấp vào **Folder** hiện tại là nơi mà bạn muốn thư mục mới của bạn được tạo (thí dụ "**Portal Root**"). Đưa con trỏ vào mục nhập của Folder mới và nhấp vào nút **Add Folder** .

3. Chọn **Folder** mà bạn muốn tại tập tin lên và sau đó nhấp vào nút **Upload File** . Nó sẽ dẫn bạn đến trang khác mà nó sẽ chỉ rõ cho bạn một hoặc nhiều tập tin muốn tải lên.

Bạn có thể chú ý thư mục của bạn chọn trước như là mục tiêu để bạn tải lên. Bạn có thể thay đổi thư mục mục tiêu bằng cách chọn cái khác từ hộp danh sách rơi xuống.



4. Nhấp vào nút **Browse** và chọn tập tin cần tải lên ở trong máy vi tính của bạn. Sau đó nhấp vào nút **Add** và lưu lại chúng trong danh sách (nên nhớ bạn có thể tải lên nhiều file). Khi bạn đã chọn tất cả các tập tin, nhấp vào **Upload New File** đề di chuyển tập tin lên website của bạn.



2. Quản lý các mô-đun thông tin

2.1. Sử dụng module Text/HTML

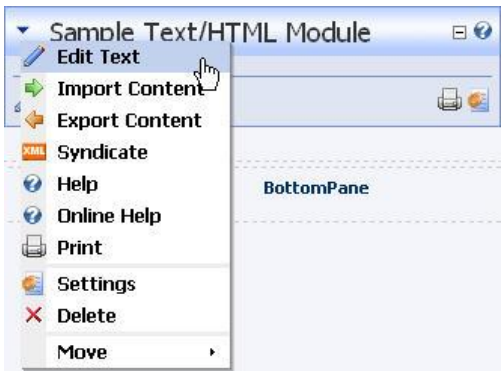
Module **Text/HTML** là "Swiss Army Knife" của module. Nó cung cấp một trình soạn thảo **WYSIWYG** để tạo ra một nội dung của chính bạn bằng HTML hoặc là văn bản thuần túy. Giao diện module tương tự như một trình biên tập document phổ biến và rất dễ làm.

Dùng những gì mà bạn đã học ở trong phần **thêm module vào trang** để thêm một **Text/HTML** mới vào trang của bạn.

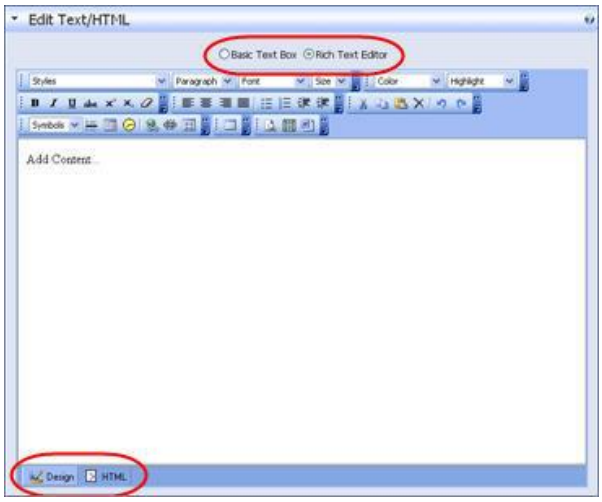
Các bước để soạn thảo văn bản(đơn giản)

Module ban đầu bạn tạo ra không có nội dung. Nó chỉ hiển thị một **Container** trống, với **Title** và **Action Buttons**.

1. Nhấp vào nút **Edit Text** (hoặc chọn **Edit text** từ **Module Actions Menu**).



Trên trang **Edit Text/HTML** bạn sẽ thấy rằng bạn (mặc định) đang sử dụng "**Rich Text Editor**" trong mẫu thiết kế. Nó sẽ cung cấp cho bạn một ý kiến tốt nhất để khi mà bạn hoàn thành thì nội dung sẽ như thế này.(WYSIWYG = " What you see is what you get").



2.2. Sử dụng module thông báo

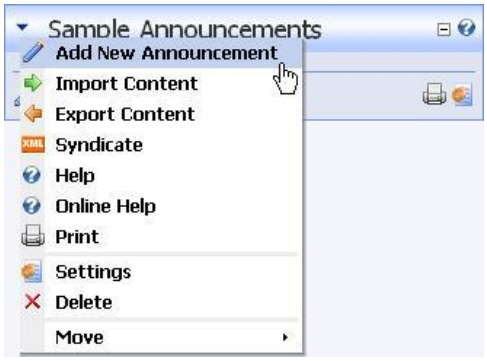
Module thông báo đưa ra một danh sách các đoạn thông báo đơn giản bao gồm tiêu đề và những hướng dẫn ngắn gọn. Nó rất thuận tiện để những người truy cập website nhận biết được các mục quan tâm khi có liên kết những thông tin thêm vào.

Dùng những gì bạn đã học trong **thêm module vào trang** để thêm **module thông báo** vào trang của bạn.

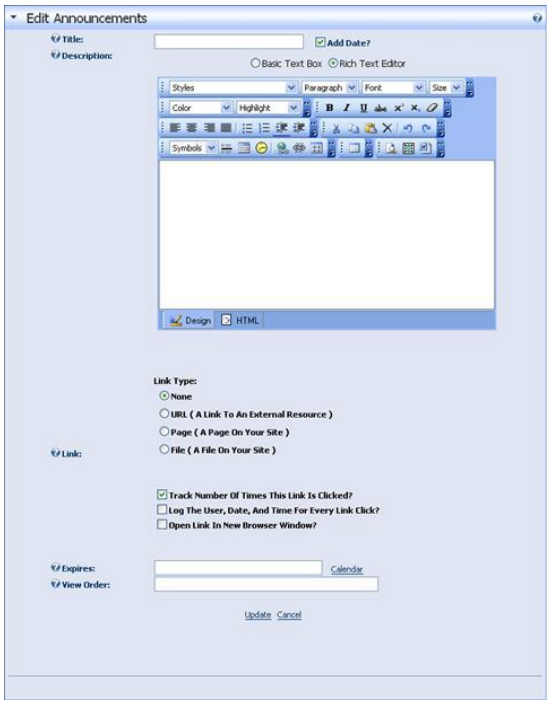
Các bước để thêm thông báo mới

Module của bạn được tạo ra lúc ban đầu thì không có nội dung. Nó chỉ hiển thị một **Container** trống, với **Title** và **Action Buttons**.

1. Nhấp vào **Add New Announcement** (hoặc chọn **Add New Announcement** từ **Module Actions Menu**).

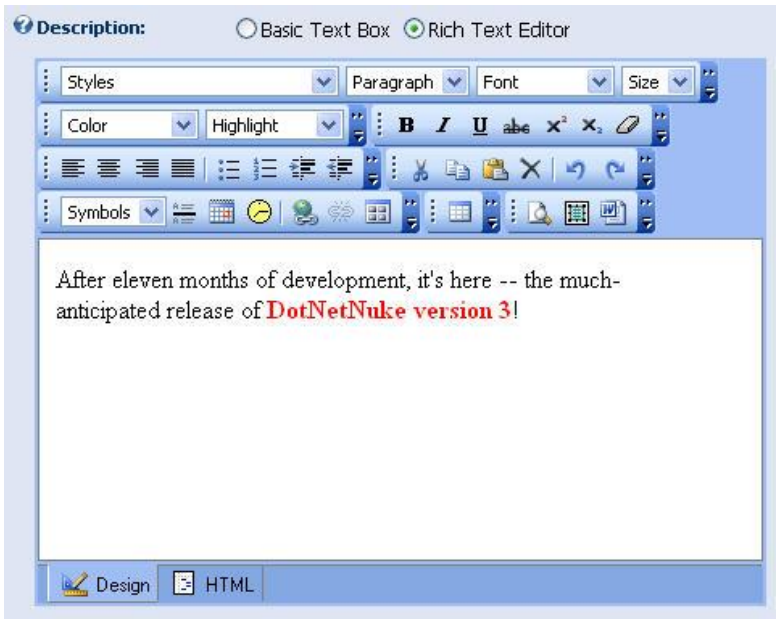


Trên trang **Edit Announcements** bạn sẽ thấy rằng có một vài vùng bạn cần phải điền vào trình văn bản **WYSIWYG** như lúc sử dụng cho module **Text/HTML**.



2. Đặt con trỏ chuột vào vùng **Title** và nhập vào đó một câu lệnh ngắn gọn để tóm tắt thông báo của bạn. Tùy chọn, đánh vào hộp **Add Date** để nối dữ liệu vào ngày hiện hành tiêu đề thông báo của bạn.

3. Đặt con trỏ chuột vào phần mô tả và dùng những tính năng của trình văn bản **WYSIWYG** để tạo ra phần nội dung cho **thông báo** của bạn.



4. Chọn dạng liên kết cho thông báo của bạn và hoàn thành những thông tin thích hợp. Bạn có thể chỉ rõ đường liên kết cho website, trên trang của website bạn hoặc đến một tập tin trên website (như là hinh ảnh hoặc tài liệu PDF).

5. Chọn đường liên kết từ những hộp tùy chọn có sẵn.

Nếu như chọn **Tracking**, bạn sẽ có thể kiểm tra đường liên kết này được click vào bao nhiêu lần. Nếu bạn chọn **Logging**, bạn có thể đoán được ai đã click vào đường liên kết này (Nếu người sử dụng đăng nhập vào hệ thống). Và nếu như bạn chọn **New Window**, đường link của bạn sẽ làm cho một cửa sổ trình duyệt được mở ra cho đường link đó.

6. Chỉ rõ ngày mà thông báo của bạn ngưng không hiển thị nữa. Bạn có thể nhấp vào **Calendar** để bật lên lịch biểu là nơi mà bạn có thể chọn cái ngày đó.

7. Bao gồm **View Order** để sắp xếp lại thông báo của bạn. Theo mặc định, chúng sẽ được sắp xếp như lúc mà bạn đã tạo ra chúng.

8. Khi bạn đã làm xong, phải bảo đảm rằng bạn đã nhấp vào **Update** ở giữa trang để lưu lại những thay đổi của bạn. Bạn phải update như thế nào để người truy cập website có thể nhìn thấy.

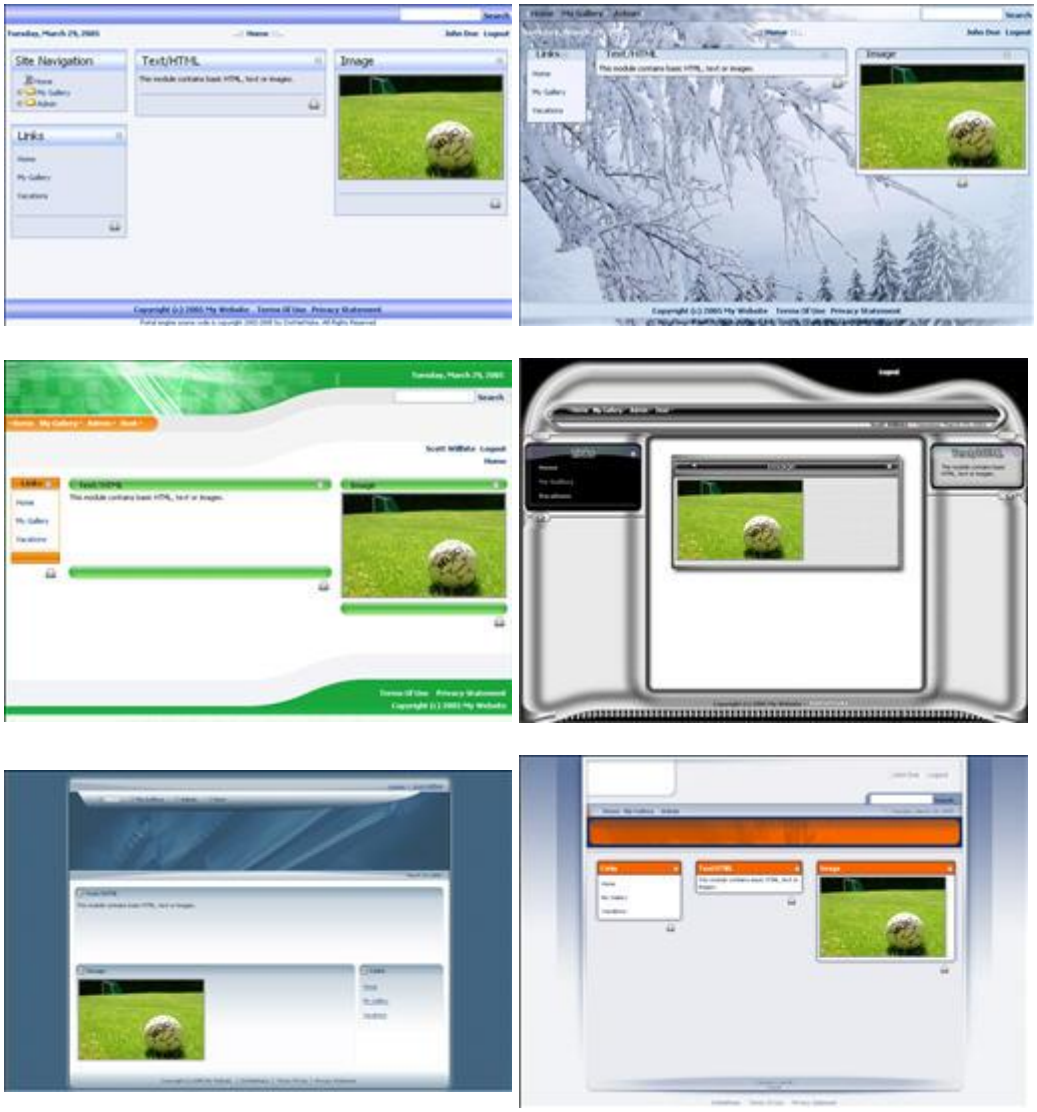
3. Tạo giao diện mới cho website

3.1. Thay đổi skin website

Skin là phương thức dùng để mô tả cách hiển thị và cách bố trí các trang trên website của bạn. Và bạn có thể thay đổi chúng với một vài nhấp chuột đơn giản!

Một **Skin** có thể rất đơn giản hoặc phức tạp. Nó có thể cung cấp **Menu** nằm dọc hay nằm ngang, cung cấp những dạng khác nhau của **Container** (như skin cho **Module**), cung cấp các cách bố trí khác nhau, v..v.. Đó là tùy theo sự thu thập skin của người thiết kế.

Ở đây có ví dụ về vài **Skin** bao gồm cả **DotNetNuke**. Nhiều skin có thể dùng nhiều nguồn khác nhau (cả miễn phí lẫn bán). Skin của website có thể được thiết kế tùy thích.



Các bước thay đổi mặc định của skin

Skin mặc định của website bạn được quản lý ở trên trang chỉ bị ảnh hưởng bởi **quản trị viên**. Thiết lập này cung cấp cho bạn toàn bộ website và tất cả các trang mà bạn tạo ra.

1. Chọn **Skins** từ menu Admin

Trên trang **Skins,** bạn thấy được một vùng và chọn cả hai **Skin** và **Container** cho website của bạn.

2. Chọn trong tập sưu tập **Skin** để xem từ hộp danh sách rơi xuống ở đầu trang.

Check vào hộp **Host** và **Site** ☑ để cho biết là bạn muốn thấy skin options được cung cấp bởi máy chủ (cho mỗi người) và chỉ phụ thuộc vào site của bạn. Những chọn lựa đó sẽ xuất hiện trong hộp danh sách rơi xuống. Khi bạn chọn một, thì tang sẽ làm mới lại để hiển thị bộ sưu tập của **Skins** và **Container.**

3. Ở giữa trang bạn có thể nhận dạng skin bạn chon được áp dụng cho public **Portal** của bạn hoặc chỉ ở phần **Admin** của website.

Check ☑ hộp **Portal** và **Admin** để cho biết là bạn muốn áp dụng skin mà ban chọn cho cả 2 dạng trang (đây là mặc định).

4. Chọn **Skin** mà bạn thích như là một mặc định cho trang web của bạn và nhấp **Apply.**

Nếu **Skin** bao gồm hình ảnh, nó tạo ra hình nhỏ cái mà bạn có thể nhấp vào để thấy hình lớn hơn. Bạn có thê nhấp vào **Preview** cái mà sẽ bật lên cửa sổ hiện ra trang chủ của website với Skin vừa mới áp dụng (bạn phải chắc là đã đóng cửa sổ khi bạn đã xem trước xong).

Website của bạn bây giờ đã thay đổi **Skin** mặc định (như minh họa bên dưới). Bạn cũng có thể thiết đặc mặc định **Container** khiến cho skin của bạn đẹp hơn (chú ý container xanh vẫn là mặc định như bên dưới).

3.2. Thay đổi container

Các bước để thay đổi container mặc định

Quản lý **Container** trên trang web của bạn cũng giống như trên **Skin**. Tương tự như skin, các thiết lập của bạn được áp dụng trên toàn bộ website và tới tất cả các **Module** mới mà bạn tạo. Nó có thể định rõ các container khác nhau trên các module cá nhân (nếu bạn muốn).

Chúng ta sẽ lập lại bước 1,2 &3 như ở trên và sau đó chỉ mặc định của **Container**.

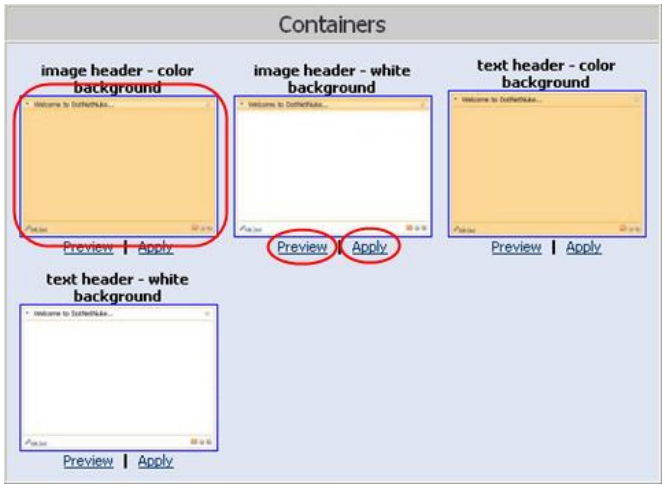
1. Chọn **Skins** từ menu Admin.

2. Chọn bộ sưu tập **Skin** hoặc **Container** muốn xem từ hộp danh sách rơi xuống ở đầu trang. Thường thì bộ sưu tập skin sẽ bao gồm các mẫu mà bạn muốn chọn.

3. Ở giữa trang bạn được chỉ rõ là **Container** mà bạn chọn nên áp dụng vào puclic **Portal** của bạn hoặc chỉ là phần **Admin** trên website của bạn.

4. Chọn **Container** là mặc định cho trang web của bạn và nhấp vào **Apply**.

Nếu **Container** có hình ảnh, nó được tạo từ hình nhỏ là cái mà có thể nhấp vào để cho ra hình lớn. Bạn có thể nhấp vào **Preview** là cái nó sẽ bật lên cửa sổ hiển ra trang chủ của website với skin đã được áp dụng (phải chắc là bạn đã đóng cửa sổ window khi bạn đã xem trước xong).



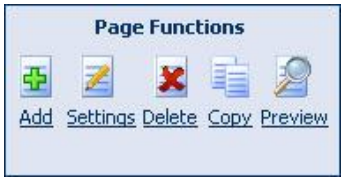
Website của bạn bây giờ sẽ thay đổi **Container** mặc định (như minh họa ở bên dưới). Bạn đã thay đổi xong cả **Skin** và container, bề ngoài trang web của bạn đã làm xong!

Các bước thay đổi skin/container cho trang đơn

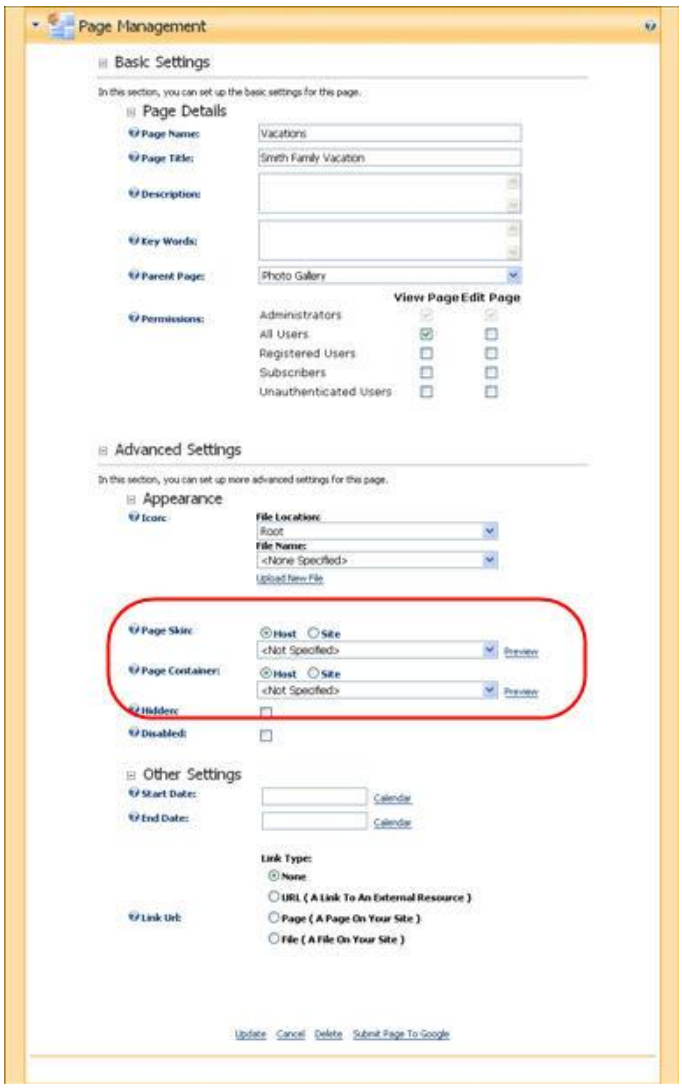
Có thể chỉ rõ sự khác nhau của **Skin** và **Container** mặc định cho một trang cá nhân của website bạn. Bạn phải chú ý răng nếu bạn thay đổi skin hoặc container tại trang đó, thì những thiết lập sẽ không có tác động khi bạn thay đổi mặc định.

Khi bạn thay đổi mặc định của **Container** cho một trang đơn, container được áp dụng bởi mặc định của bất cứ **Module** nào được thêm vào ở trang đó.

1. Thông qua trang mà bạn muốn thay đổi **Skin** hoặc mặc định của **Container.**
2. Chọn **Setting** từ **Page Fuctions** của **Control Panel.**



3. Trên trang **Page Management,** mở rộng ra là nhóm các thiết lập nâng cao khi nhấp vào biểu tượng ⊞. Cuốn tới phần thiết lập **Page Skin** và **Page Container** ở phần **Appearance.**



4. Nút **Host** và **Site** cho biết là bạn muốn nhìn thấy skin options được kiểm bởi host (cho mọi người) hay là phụ thuộc vào site của bạn. Khi bạn chọn một, trang sẽ làm mới lại để hiển thị bộ sưu tập của **Skins** và **Containers.**

Chọn **Skin** hoặc **Container** mặc định mà bạn thích rồi áp dụng vào trang.

Nút **Preview** làm việc như là nó ở trên trang **Skin**, cung cấp của sổ bật lên với trang chủ của website bạn với sự lựa chọn áp dụng.

5. Khi bạn đã làm xong những bước trên, di chuyển vào giữa trang rồi nhấp vào **Update** để lưu lại những thay đổi của bạn!

BÀI 10: XUẤT BẢN WEBSITE

Mã Bài: MĐ19_10

Giới thiệu:

Để website hoạt động được trên Internet, bước cuối cùng là chúng ta phải triển khai website trên máy chủ web. Trong bài này chúng tôi sẽ trình bày một số giải pháp đơn giản nhất.

Mục tiêu:

- Xác định được một số công cụ xuất bản Website;
- Xác định được một số dịch vụ và cách đăng ký tên miền;
- Xuất bản được website lên thư mục cục bộ, kiểm tra tổng thể các chức năng của site;
- Lựa chọn được máy chủ, tên miền phù hợp;
- Xuất bản thành công site lên internet;
- Rèn luyện thái độ làm việc nghiêm túc, tinh thần trách nhiệm cao.

Nội dung chính:

1. Chọn và đăng ký tên miền cho website

Giá trị của tên miền thể hiện ở chỗ bạn sẽ nhớ ngay ra tên miền đó khi muốn tìm kiếm một website. Ví dụ, nếu bạn muốn truy cập vào trang web của một Công ty XYZ bạn có các cách phổ biến như gõ địa chỉ www.xyz.com hoặc www.xyz.com.vn. Do tên miền .com và .com.vn rất quen thuộc với chúng ta

Tuy nhiên, ngoài 2 tên miền phổ biến trên, còn có rất nhiều tên miền khác. Khi sử dụng chúng, có thể bạn sẽ gặp phải những vấn đề mà tôi kể dưới đây:

- Nếu bạn sử dụng tên miền khác, một lượng khách hàng tiềm năng có thể vị mất đi vì họ sẽ có khuynh hướng rất tự nhiên và theo thói quen là truy cập vào trang web theo 2 tên miền phổ biến là .com và .com.vn.
- Nếu công ty bạn chưa sử dụng tên miền này thì rất có thể nó sẽ bị mua bởi một kẻ có ý đồ xấu, muốn lập ra một trang web, sau đó tung tin làm công ty bạn bị thiệt hại về mặt uy tín. Cuối cùng, bạn phải bó tiền ra mua lại tên miền với giá cao.
- Bạn chưa sử dụng tên miền đó nên đối thủ cạnh tranh đã tranh thủ thời cơ mua nó và tạo ra trang web để đưa tin không đúng làm công ty bạn gặp khó khăn.
- Một cách để xây dựng uy tín công ty là lập trang web có tên miền theo tên công ty của bạn. Nếu bạn không làm được điều này, uy tín của công ty sẽ bị giảm sút.

Cách đăng ký tên miền cho một website:

Trước hết, nên chọn đuôi tên miền ".com", ".net" hay ".org"?

Xin trả lời như sau: nếu bạn sử dụng tên miền đó vào mục đích kinh doanh thì TLD của bạn phải là ".com", không có ngoại lệ. Tất cả mọi người đều nhớ đến ".com" trước tất cả các loại "DOT" khác. Có người còn cho là mọi tên miền trên thế giới đều có phần đuôi là ".com". Nếu bạn đang sử dụng một tên miền ".net" và cố gắng quảng cáo thương hiệu của mình, chẳng hạn nếu chúng tôi sử dụng tên miền xyz.vn và cố gắng làm cho mọi người nhớ đến chúng tôi qua cụm từ "xyz". Chúng tôi xin cam đoan rằng có nhiều khách hàng thiện chí sẽ gõ vào trình duyệt của họ là xyz.com và nhấn Enter. Điều gì sẽ xảy ra?. Chúng tôi đang làm giàu cho người có tên miền xyz.com mà không phải là chúng tôi, xyz.vn. Còn một điều nữa, nếu người truy cập quên không gõ vào trình duyệt của họ phần đuôi (TLD) thì bất cứ mọi trình duyệt nào hiện nay điều mặc định thêm vào phần đuôi ".com".

Hơn 80% khả năng sẽ không còn tên miền đó vì có rất nhiều công ty có tên trùng nhau. Hơn nữa, có nhiều đối thử sừng sỏ đang tìm cách đăng ký tên miền của bạn. Để làm gì?. Để hạn chế cạnh tranh khi họ khai thác khách hàng trên Internet. Nếu không còn tên thương hiệu của bạn thì cũng không sao. Hãy nghĩ tới tên sản phẩm, thêm bớt một số từ ghép… Nhưng cách nhanh nhất là hãy liên hệ với chúng tôi để có được một sự tư vấn phù hợp.

Mua lại tên miền để sử dụng là phương án đầu tiên cần phải nghĩ đến, nếu tên miền mà bạn muốn mua đang thuộc quyền sở hữu của người khác.

2. Chọn và đăng ký máy chủ

* Chọn dịch vụ hosting

- Thuê host hay là mua host?

Cái này chỉ là vấn đề nhỏ thôi nhưng việc biết nó cũng ít nhiều gì cũng có ích.

Nếu nói về host, thì bạn có thể gọi là mua host hoặc thuê host gì cũng được, nhưng chính xác nhất thì là thuê host. Sở dĩ chúng ta gọi là thuê bởi vì bạn phải trả tiền sử dụng theo hàng tháng vì chi phí cho các thứ như mạng, cơ sở hạ tầng của datacenter, tủ rack đựng máy chủ, máy chủ vật lý đều là của họ.

- Chọn host Việt Nam hay host nước ngoài ?

Host Việt Nam là như thế nào?

Hình 5.3: Datacenter của Viettel IDC Sóng Thần – Bình Dương

Host Việt Nam không phải hoàn toàn là nhà cung cấp đó tại Việt Nam, mà một yếu tố nữa đó là các máy chủ được đặt tại một (hoặc nhiều) datacenter tại Việt Nam.

Theo mình biết thì Việt Nam ta đang có khoảng 5 datacenter trên cả nước. Trong đó 1 datacenter tại Bình Dương và 4 datacenter tại Hà Nội, cụ thể:

IDC Viettel Datacenter (Bình Dương)

FPT Datacenter (Hà Nội)

GDS Hà Nội – Thăng Long

Hanel – CSF Datacenter (Hà Nội)

Telehouse International Vietnam (Hà Nội)

Ưu điểm của Host Việt Nam

- Tốc độ truy cập trong nước rất nhanh vì sử dụng chung một hạ tầng, băng thông nội bộ và khoảng cách địa lý từ người dùng đến datacenter thấp.
- Dễ mua, dễ thanh toán.
- Dễ yêu cầu hỗ trợ.
- Không bất đồng ngôn ngữ.

Nhược điểm của Host Việt Nam

- Giá hơi cao do mặt bằng băng thông cao so với Châu Âu hay Mỹ.
- Gây cản trở khi truy cập từ nước ngoài nếu có đứt cáp.
- Một số dịch vụ Hosting hỗ trợ sau bán hàng không được tốt.

Host Nước Ngoài là như thế nào?

Bên trong một Datacenter tại Dallas, Texas. Mỹ

Ngắn gọn là các nhà cung cấp host này có trụ sở và máy chủ không đặt tại Việt Nam. Hiện nay một số nơi đặt datacenter thông dụng nhất trên thế giới cho các dịch vụ hosting là:

Dallas, Texas, Mỹ.

Amsterdam, Hà Lan.

Michigan, Mỹ.

Singapore

London, Anh.

Stockholm, Thụy Điển.

Berlin, Đức.

Đó chỉ là nơi đặt server, còn trụ sở datacenter thì hiện nay rất nhiều nhà cung cấp đều có một datacenter riêng của họ chứ ít khi thuê của ai.

Ưu điểm của host nước ngoài

- Tác phong phục vụ chuyên nghiệp.
- Nhiều chương trình khuyến mãi do độ cạnh tranh cao.
- Tốc độ truyền tải đi toàn cầu rất tốt. Ở Việt Nam nếu ping sang host nước ngoài thì chỉ khoảng từ 250 đến 300ms. Nhưng ở nước ngoài ping về VN thì ít nhất là 450ms.

Nhược điểm của host nước ngoài

- Bất đồng ngôn ngữ.
- Khó khăn khi mua và thanh toán. Khi mua họ cần phải chứng thực qua CMND hoặc gọi điện. Và bạn chỉ có thể thanh toán qua Credit Card (Visa/Mastercard) hoặc PayPal.
- Số lượng quá nhiều khó chọn.
- Nhiều chỗ không chào đón người Việt cho lắm.

3. Chọn công cụ xuất bản

Với mã nguồn DotNetNuke, để thuận lợi cho việc cài đặt và triển khai chúng tôi khuyến cáo nên sử dụng bản Install. Bản này nhỏ gọn đồng thời có thể giúp người lập trình phát triển thêm mô đun.

4. Xuất bản Website

Chúng ta nên cài đặt website và thiết kế giao diện trên máy cục bộ, sau đó để xuất bản website phát triển bằng DotNetNuke bạn cần backup cơ sở dữ liệu, đồng thời copy thư mục web đã cài đặt gửi cho nhà cung cấp hosting.

Thường để thuận lợi cho việc gửi file cho nhà cung cấp hosting, chúng ta nên sử dụng phần mềm FTP Client để upload toàn bộ dữ liệu (file .bak) và source của website, sau đó nhờ bộ phận kỹ thuật của hosting cài đặt theo ý muốn của chúng ta.

TÀI LIỆU THAM KHẢO

[1] VN-Guide (2004), *Thiết kế trang web với HTML*, NXB Thống kê Hà Nội;

[2] VN Guide (1995), *Internet toàn tập,* NXB Khoa học kỹ thuật;

[3] SCC Technology (2004), *Thiết kế website với Macromedia Dreamweaver*;

[4] Nguyễn Trường Sinh (2006), *Thiết kế website với FrontPage*, NXB Thống kê;

[5] Khuất Thùy Phương (2007), *Lập Trình Ứng Dụng Web Với PHP*, NXB Thống kê;

[6] Maricopa Center for Learning and Instruction (1999), *Writing HTML Tutorial*;

[7] Mediaspace Club (2004), *Macromedia DreamweaverMX*.

[8] http://www.php.net/manual/

[9] Các tài liệu, bài tập PHP trên internet;

[10] Tài liệu hướng dẫn sử dụng ứng dụng web làm công cụ xây dựng website.

[12] Phạm Hữu Khang (chủ biên) (2005), *Lập trình ASP.NET 2.0*, NXB Lao động - Xã hội.

[13] Dương Quang Thiện (biên soạn) (2005), *Lập trình Web dùng ASP.NET Và C#*, NXB Tổng Hợp TP. Hồ Chí Minh.

[14] Các tài liệu tham khảo khác trên mạng.