

MỤC LỤC

Chương 1: GIỚI THIỆU TỔNG QUAN VỀ ASP.NET	7
I. Giới thiệu về Dotnet Framework 3.5	7
1.1. Dot Net Framework là gì?	7
1.2. Một số điểm mới trong .net 3.5	7
1.3. Một số thư viện tiêu biểu trong .NET Framework	8
II. Sự phát triển của việc xây dựng ứng dụng Web	10
ASP.NET	10
III. 7 tính năng quan trọng ASP.NET	10
3.1. ASP.NET được kết hợp .NET Framework.....	10
3.2. ASP.NET được biên dịch, không thông dịch.....	10
3.3. ASP.NET hỗ trợ đa ngôn ngữ	11
3.4. ASP.NET thực thi ở giai đoạn Common Language Runtime	11
3.5. ASP.NET hướng đối tượng	11
3.6. ASP.NET đa thiết bị và đa trình duyệt.....	11
3.7. ASP.NET dễ dàng triển khai và cấu hình.....	12
IV.	ASP.NET 2.0 12
Chương 2: WEB FORM.....	14
I. HTML Forms	14
II. Mô hình sự kiện trong ASP.NET	15
III.Các giai đoạn xử lý của một Web Form.....	15
3.1. Khởi tạo Page framework – Framework Init.....	16
3.2. Khởi tạo mã người dùng - User code	16
3.3. Kiểm tra – Validation	16
3.4. Quản lý sự kiện - Event handling	16
3.5. Liên kết dữ liệu tự động - data binding	16
3.6. Dọn dẹp - Cleanup.....	17
3.7. Ví dụ về PageFlow	17
IV.Đối tượng trang Page	19
4.1. Đối tượng Session, Application và Cache.....	19
4.2. Request.....	19
4.2. Response.....	19
4.3. Đối tượng Server	20
4.4. Đối tượng User	20
Bài tập chương 2	21
Lab 2.1. Sử dụng HTML Control trong ASP.NET	21

Lab 2.2: Tạo Form Đăng Ký	23
Chương 3: SERVER CONTROLS	24
Bài 1: HTML Server Controls.....	24
I. Giới thiệu	24
II. Lược đồ lớp Server Control	25
III.HTML server controls.....	26
3.1. Lược đồ tổng quát của HTML Server Controls.....	26
3.2. Quản lý sự kiện phía Server	27
Bài 2: WEB CONTROLS	30
I. Giới thiệu về Web Controls	30
II. Lớp cơ sở WebControl.....	30
III.Danh sách các web controls	31
IV. Các List Controls	32
4.1. Một số List controls.....	32
4.2. Các thuộc tính của lớp ListControl	33
4.3. Ví dụ:.....	33
V. Mô hình sự kiện Web Control.....	35
Bài 3: ĐIỀU KHIỂN KIỂM TRA NHẬP LIỆU	38
I. Điều khiển kiểm tra nhập liệu:	38
II. Các loại điều khiển kiểm tra nhập liệu.....	38
III.Lớp cơ sở Validator	39
IV.RegularExpression Validator	41
V. CustomValidator Control	42
Bài 4: Rich Controls	43
I. Rich controls là gì?	43
II. Điều khiển tạo tập ảnh quảng cáo AdRotator	43
III. Điều khiển tạo lịch Calendar	44
Bài tập Chương 3	46
Lab 3.1: Sử dụng ASP.net Web Control cơ bản.....	46
Lab 3.2: Sử dụng các Web Control cơ bản	50
Lab 3.3: Sử dụng Validation Controls.....	53
Chương 4: ĐIỀU KHIỂN CHUYÊN HƯỚNG DỮ LIỆU.....	56
I. Multiview và Wizard Controls.....	56
1.1. MultiView Control	56
1.2. Wizard Control.....	58
II. Site Map	60
2.1. Tạo một file Web.sitemap file	60

2.2. Ví dụ:.....	60
III.Điều khiển TreeView	61
3.1. Tạo một TreeView đơn giản.....	61
3.2. Định dạng Tree View	63
3.3. Liên kết TreeView với SiteMap	66
IV. Điều khiển MenuView	67
4.1. Tạo một Menu đơn giản	67
4.2. Định dạng MenuView	68
4.3. Ví dụ Menu	68
Chương 5: ADO.NET	70
Bài 1: TỔNG QUAN VỀ ADO.NET	70
I. Giới thiệu	70
II. Kiến trúc của ADO.NET.....	71
2.1. ADO.NET Data Providers.....	71
2.2. Các lớp ADO.net cơ bản	71
III.Lớp Connection.....	72
3.1. Chuỗi kết nối	72
3.2. Kiểm tra kết nối:.....	73
Bài 2: Đối tượng Command và DataReader.....	75
I. Đối tượng Command.....	75
1.1. Các kiểu câu lệnh Command.....	75
1.2. Các phương thức của đối tượng Command.....	76
II. Đối tượng DataReader.....	76
III. Mô tả cơ sở dữ liệu.....	77
IV. Một số ví dụ về Lớp Command và Data Reader.....	77
4.1. Áp dụng Phương thức ExcuteNonQuery().....	77
4.2. Phương thức ExecuteScalar().....	79
4.2. Phương thức ExecuteNonQuery().....	80
4.3. Làm việc với thủ tục Stored Procedure	81
Bài 3: Lớp DataSet và Data Adapter.....	84
I. Lớp DataSet là gì?.....	84
II. Lớp Data Adapter.....	84
III. Ví dụ:	85
Bài 4: Data Binding.....	87
I. Giới thiệu	87
II. Cơ bản về Data Binding	87
2.1. Single Value Binding	87

2.2. Liên kết dữ liệu có lặp (Repeated Value Binding)	88
2.3. Liên kết dữ liệu với DataReader	90
2.4. Chu kỳ sống của trang web có sử dụng Data Binding.....	91
BÀI 5: ĐIỀU KHIỂN NGUỒN DỮ LIỆU	93
I. Các loại điều khiển nguồn dữ liệu.....	93
II. Điều khiển SqlDataSource:	93
2.1. Truy xuất các mẫu tin:.....	94
2.2. Câu lệnh có tham số	94
2.3. Cập nhật dữ liệu	96
2.4. Xóa mẫu tin	96
2.5. Thêm mẫu tin.....	96
2.6. Ví dụ:.....	97
III.Một số hạn chế của SqlDataSource.....	98
Bài 6: ASP.NET VỚI XML	99
I. Vai trò của XML trong .net.....	99
1.1. Tập tin cấu hình.....	99
1.2. Truy xuất dữ liệu trong ADO.NET	99
1.3. Một số ví dụ về XML.....	99
II. Cơ bản về XML.....	100
2.1. Một vài quy tắc khi tạo file XML	100
2.2. Thuộc tính:	100
2.3. Các lớp XML:	101
III. Ví dụ XML với c#.....	101
3.1. Đọc XML và đưa vào DataSet	101
3.2. Đọc XML với XMLDataSource.....	102
Bài tập chương 5:	104
HƯỚNG DẪN	106
Chương 6: ĐIỀU KHIỂN TRÌNH BÀY DỮ LIỆU	118
I. Giới thiệu tổng quan.....	118
II. Điều khiển Gridview	118
2.1. Định dạng cột	119
2.2. Định dạng hiển thị Gridview	121
2.3. Sự kiện trong GridViews.....	121
2.4. Sử dụng GridView Template:	123
III.DetailView	125
IV. FormView	125
BÀI TẬP CHƯƠNG 6	129

Chương 7: QUẢN LÝ TRẠNG THÁI.....	135
I. Các biến ViewState, Cookies và Sessions	135
1.1. View State	135
1.2. Cookies.....	136
1.3. Sessions:.....	137
II. Application State.....	138
III.Tập tin ứng dụng global.asax	138
IV. Tập tin Web.config	139
4.1. Cấu trúc tập tin web.config.	139
4.2. Tập tin cấu hình ứng dụng Web.config:.....	139
4.3. Các cấu hình mặc định	140
4.4. SessionState.....	141
4.5. Làm việc với tập tin web.config.....	141
Bài 2: USER CONTROL & MASTER PAGE	142
I. User controls	142
II. Themes	143
III.Master Page.....	145
Bài tập chương 7	147
Lab 7.1. TẠO LAYOUT TRANG TIN TỨC	147
Lab 7.2: Tạo trang tóm tắt tin tức.....	151

DANH MỤC HÌNH ẢNH

Hình 1: Mô hình .NET Framework	6
Hình 2: Mô hình .NET Framework	7
Hình 3: Sơ đồ thư viện lớp của .NET	8
Hình 4: Mô hình CLR trong .NET	10
Hình 5: Form kết quả của ví dụ HTML Form	13
Hình 6: Chu trình sống của một trang web	15
Hình 7: Form Kết quả của PageLayout	17
Hình 8: Lược đồ lớp Server Controls	24
Hình 9: Các loại HTML Server Controls	25
Hình 10: Form kết quả Sự kiện trong HTML server controls	28
Hình 11: Form kết quả Sự kiện trong HTML server controls	28
Hình 12: Lược đồ Web Controls	29
Hình 13: Form Kết quả với Web Controls	31
Hình 14: Form Kết quả với List Controls	34
Hình 15: Mô hình sự kiện Auto Postback	35
Hình 16: Form Kết quả với sự kiện trong asp.net	36
Hình 17: Form Kết quả với Điều khiển Kiểm tra Nhập liệu	40
Hình 18: Màn hình thiết kế với các View	57
Hình 19: Màn hình Kết quả thiết kế điều khiển Wizard	59
Hình 20: Lược đồ liên kết trong SiteMap	60
Hình 21: Màn hình thiết kế TreeView ở chế độ Design	62
Hình 22: Màn hình Kết quả Form với TreeView	63
Hình 23: Các thuộc tính định dạng các nút trong TreeView	64
Hình 24: Các thuộc tính định dạng TreeView	65
Hình 25: Các thuộc tính định dạng TreeView (2)	65
Hình 26: Form kết quả TreeView (3)	67
Hình 27: Form kết quả Menu	68
Hình 28: Kiến trúc ADO.net	71
Hình 29: Kết quả Form Kiểm tra kết nối	74
Hình 30: Form Kết quả Truy vấn dữ liệu với câu lệnh SELECT	79
Hình 31: Form Kết quả thực hiện phương thức ExcecuteScalar	80
Hình 32: Form Kết quả thực hiện phương thức ExcecuteNonQuery()	81
Hình 33: Form Kết quả ví dụ trang 82	83
Hình 34: Các lớp của tập dữ liệu DataSet	84
Hình 35: Mô hình Lớp DataAdapter tương tác với nguồn dữ liệu	85
Hình 36: Form Kết quả ví dụ trang 85	86
Hình 37: Form Kết quả ví dụ trang 89	90
Hình 38: Form Kết quả ví dụ trang 91	91
Hình 39: Form Kết quả ví dụ trang 95	95
Hình 40: Form Kết quả ví dụ trang 102	102
Hình 41: Form Kết quả ví dụ trang 119	119
Hình 42: Form Kết quả ví dụ trang 123	123
Hình 43: Form Kết quả ví dụ trang 124	124
Hình 44: Form Kết quả ví dụ trang 126	127
Hình 45: Form Kết quả ví dụ trang 126	128
Hình 46: Form Kết quả ví dụ trang 126	128
Hình 47: Các biến quản lý trạng thái trong ASP.net	135
Hình 48: Yêu cầu một trang trong ASP.net.....	135
Hình 49: Form kết quả trang 140	140
Hình 50: Mô hình Theme và Skin	144
Hình 51: Form kết quả áp dụng Theme	144

Chương 1: GIỚI THIỆU TỔNG QUAN VỀ ASP.NET

Mục tiêu học tập: Sau khi học xong bài này, người học có thể:

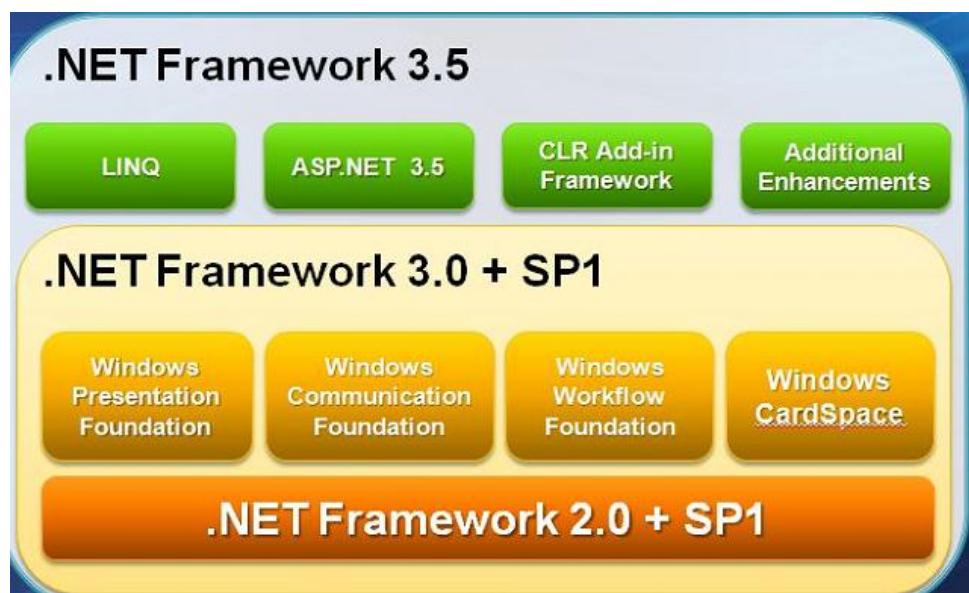
- Hiểu được Dotnet Framework 3.5.
- Trình bày được các giai đoạn của phát triển một ứng dụng web
- Trình bày một số điểm chính trong ASP.net 3.5

I. Giới thiệu về Dotnet Framework 3.5

1.1. Dot Net Framework là gì?

Microsoft .Net Framework là một môi trường lập trình được phát triển bởi Microsoft. Nó được sử dụng để tạo, triển khai và thực thi các ứng dụng và dịch vụ sử dụng công nghệ .Net như ứng dụng windows và các dịch vụ web.

Microsoft .NET Framework là một phần mềm có thể chạy trên các hệ điều hành windows. Nó bao gồm thư viện các lớp được viết sẵn để đơn giản hóa việc viết code của lập trình viên. Đồng thời nó cũng có hỗ trợ máy ảo để quản lý việc thực thi của chương trình đối với framework.



Hình 1: Mô hình .NET Framework

1.2. Một số điểm mới trong .net 3.5.

.NET Compact Framework version 3.5 mở rộng hỗ trợ cho ứng dụng mobiles với công nghệ Windows Communication Foundation (WCF). Đặc biệt có tính năng mới như LINQ, các công cụ và ứng dụng được cập nhật mới. Cụ thể là một số đặc tính mới trong C# 3.0 và trình biên dịch VB.NET 9.0, Visual C#, Visual C++ 2008, Visual Basic Language Three Microsoft.

1.2.1. Truy vấn tích hợp ngôn ngữ (Language-Integrated Query LINQ)

LINQ là một tính năng mới trong Visual Studio 2008 và .NET Framework 3.5. LINQ bổ sung đặc tính truy vấn dữ liệu cơ bản đối với C# và VB.NET cùng với trình biên dịch và các hỗ trợ thông minh. Công nghệ này có thể được mở rộng để hỗ trợ bất kỳ kho dữ liệu nào. .NET Framework 3.5 bao gồm LINQ để truy vấn vào thư viện .NET Framework, cơ sở dữ liệu SQL server, tập dữ liệu ADO.net và tài liệu XML.

1.2.2. Thêm hỗ trợ cây biểu thức và các phương thức lambda

Các cây biểu thức mới được trình bày trong .NET Framework 3.5 và cung cấp một cách trình bày mã lệnh ở mức ngôn ngữ dưới hình thức dữ liệu. Cây biểu thức được dùng rất nhiều trong câu truy vấn LINQ nhằm vào khai thác các nguồn dữ liệu như cơ sở dữ liệu SQL. Các câu truy vấn được trình bày dưới dạng cây biểu thức và sự trình bày này làm thuận lợi các nhà cung cấp câu truy vấn để kiểm tra và biên dịch ngôn ngữ truy vấn theo miền. Biểu thức Lambda là cách tốt nhất để viết delegates dưới dạng cô đọng nhất.

1.2.3. Hỗ trợ phân trang trong ADO.NET

Cung cấp chức năng phân trang cho các điều khiển data-bound để thực thi giao tiếp IPageableItemContainer như điều khiển ListView. Điểm thuận lợi tốt nhất là lập trình viên là không phải tốn nhiều thời gian để viết chức năng phân trang. Chức năng page.DataPager cung cấp lập trình viên cách phù hợp để phân trang với các điều khiển có hỗ trợ phân trang.

1.2.4. Hỗ trợ pipeline HTTP và cung cấp syndication.

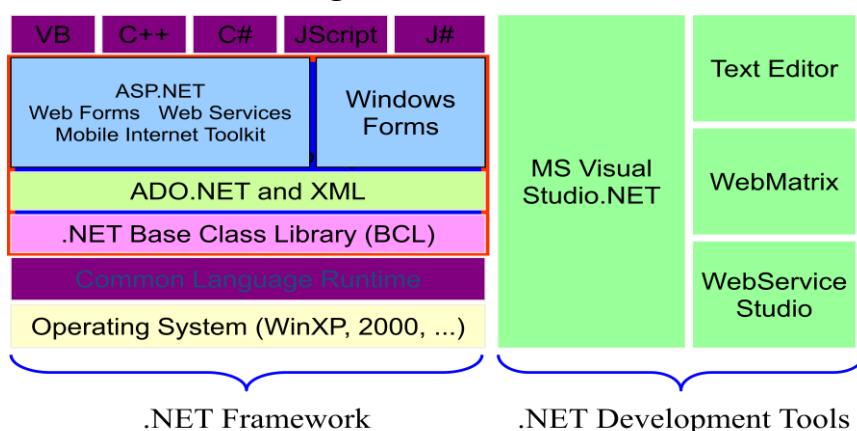
HTTP pipelining là một kỹ thuật mà khi có nhiều yêu cầu HTTP được viết để mà không cần phải đợi đáp ứng phù hợp. Một web feed là một định dạng được dùng để cung cấp người dùng. Nhà phân bố nội dung, vì vậy cho phép người dùng để đóng góp vào nó. Tạo một tập các web feeds là một kỹ thuật giải quyết trường hợp khi có nhiều yêu cầu HTTP xảy ra đồng thời.

1.2.5. ASP.NET AJAX được thêm vào thư viện (namespace) System.CodeDom.

Trong ASP.NET 2.0, để sử dụng AJAX, chúng ta phải download phần mở rộng và cài đặt nó. Tuy nhiên, trong ASP.NET 3.5, ASP.NET AJAX được tích hợp trực tiếp vào .net Framework. Thư viện Microsoft AJAX hỗ trợ các cài đặt hướng người dùng và hướng đối tượng. Bằng cách sử dụng lớp thư viện trong ECMAScript (JavaScript), bạn có thể bổ sung và chức năng giao tiếp người dùng UI mà không đến máy chủ. Các nhà phát triển Visual Web Developer thêm vào các hỗ trợ IntelliSense support dành cho JavaScript và hỗ trợ thư viện Microsoft AJAX.

1.3. Một số thư viện tiêu biểu trong .NET Framework.

1.3.1 Mô hình Framework trong .net

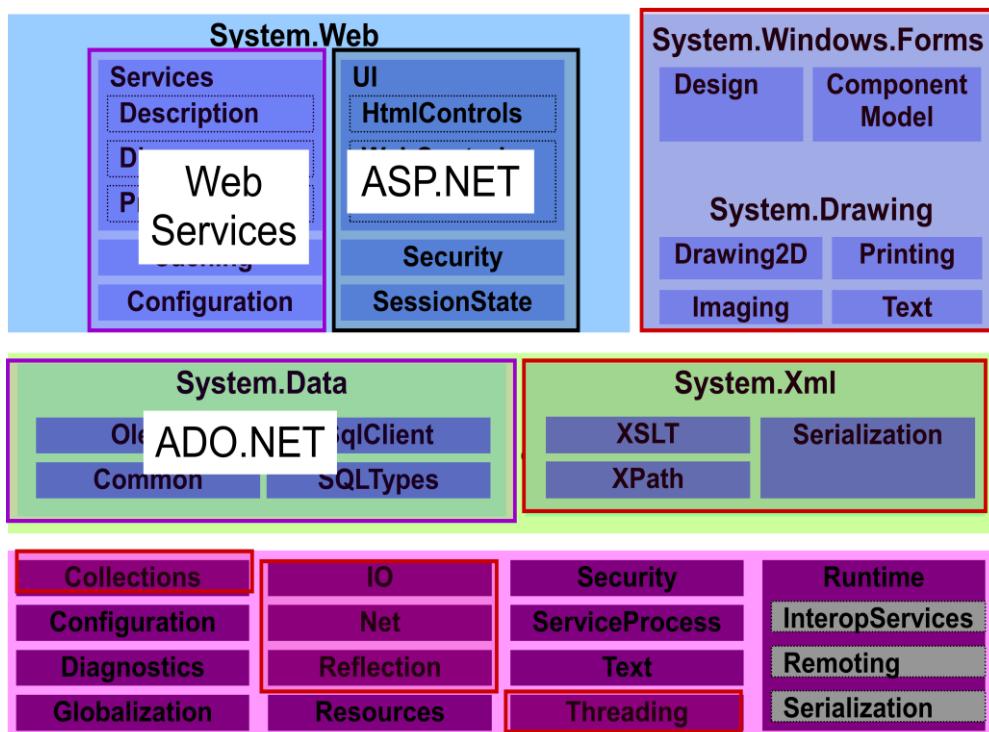


Hình 2: Mô hình .NET Framework

- Xây dựng các khối dưới hình thức các lớp và các components.
 - Khối lệnh cho Input/output
 - Khối lệnh lập trình giao diện đồ họa GUI
 - Khối lệnh cho lập trình mạng Networking
 - Các tập hợp và danh sách

- Cung cấp các lệnh phát triển phần mềm đơn giản và hiệu quả.
- .NET cung cấp một thư viện rất đầy đủ với hơn 100 thư viện (namespaces) và hơn 200 kiểu dữ liệu cho các lĩnh vực khác nhau.
- Hỗ trợ các công nghệ quan trọng như: XML, cryptography, reflection và multi-threading.

1.3.2. Thư viện lớp .Net



Hình 3: Sơ đồ thuỷ viện lớp của .NET

- Namespace **System.Configuration** cung cấp các lớp và giao tiếp cho phép thiết lập các cấu hình .NET Framework.
- Namespace **System.Diagnostics** cung cấp các lớp cho phép giao tiếp với các tiến trình hệ thống, ghi nhớ sự kiện và các bộ đếm thời gian thực hiện.
- Namespace **System.Globalization** chứa các lớp định nghĩa thông tin liên quan đến lĩnh vực văn hóa như ngôn ngữ, quốc gia/ khu vực.
- Namespace **System.Resources** cung cấp các lớp và giao tiếp mà nó hỗ trợ cho các nhà phát triển để tạo, lưu trữ và quản lý các nguồn văn hóa khác nhau.
- Namespace **System.Runtime.InteropServices** cung cấp một nhóm thành viên hỗ trợ tích hợp COM.
- Namespace **System.Runtime.Remoting** cung cấp các lớp và giao tiếp cho phép các nhà phát triển để tạo và cấu hình các ứng dụng phân tán.
- Namespace **System.Runtime.Serialization** chứa các lớp được dùng cho việc đồng bộ và tách động bộ các đối tượng.
- Namespace **System.Security** cung cấp các kiến trúc nền của hệ thống đảm bảo an toàn cho CLR.
- Namespace **System.ServiceProcess** cung cấp các lớp cho phép bạn cài đặt và điều khiển các ứng dụng dịch vụ trên windows.

II. Sự phát triển của việc xây dựng ứng dụng Web

Trong các giai đoạn đầu của phát triển ứng dụng web. Các ứng dụng nhỏ rời rạc được thực thi bởi các lệnh gọi phía server: Cài đặt đầu tiên là CGI (**Command Gateway Interface**). Vấn đề chính của mô hình phát triển này là chúng tồn rất nhiều tài nguyên server bởi vì mỗi một yêu cầu cần một thể hiện ứng dụng tách rời. Kết quả là, các ứng dụng theo phương pháp này không thể phát triển những ứng dụng lớn hơn.

Kịch bản (Scripts) được biên dịch bởi các tài nguyên phía server ASP (**Active Server Pages**) và các cài đặt trực tiếp của ColdFusion rơi vào loại này. Để sử dụng những loại mã script này, nhà phát triển web cần tạo file chứa các thẻ HTML và các đoạn mã kịch bản (script). Các file script được phân tích ngay lúc thực thi, mà nhờ chuyển đổi HTML và thực thi các mã script được nhúng vào. Quy trình này thì ít hiệu quả hơn thực thi ở giai đoạn biên dịch.

ASP.NET

Trong ASP.net vẫn giữ lại các ý tưởng phát triển một ứng dụng web trước đây. Mỗi một ứng dụng web sẽ bao gồm một số trang web. Lập trình viên web vẫn có thể trả về thẻ HTML và sử dụng JavaScript, tạo các components để đóng gói ứng dụng. Tuy nhiên, ASP.net có một vài điểm khác biệt với nền phát triển ứng dụng web trước đây như sau:

- ASP.NET: Hoàn toàn áp dụng theo mô hình **lập trình hướng đối tượng bao gồm hướng sự kiện**, kiến trúc dựa trên control hỗ trợ đóng gói ứng dụng và sử dụng lại code.
- ASP.NET: Cung cấp khả năng viết code dưới bất kỳ ngôn ngữ .Net nào (*bao gồm Visual Basic, C#, J# và nhiều ngôn ngữ khác có third-party compilers*).
- ASP.NET: Khả năng thực thi cao. Các trang ASP.net và các components được **biên dịch theo yêu cầu** thay vì được thông dịch mỗi khi chúng được dùng. ASP.net bao gồm cả mô hình truy xuất dữ liệu tốt nhất và caching dữ liệu linh hoạt để làm tăng hiệu quả thực thi.

III. 7 tính năng quan trọng ASP.NET

3.1. ASP.NET được kết hợp .NET Framework

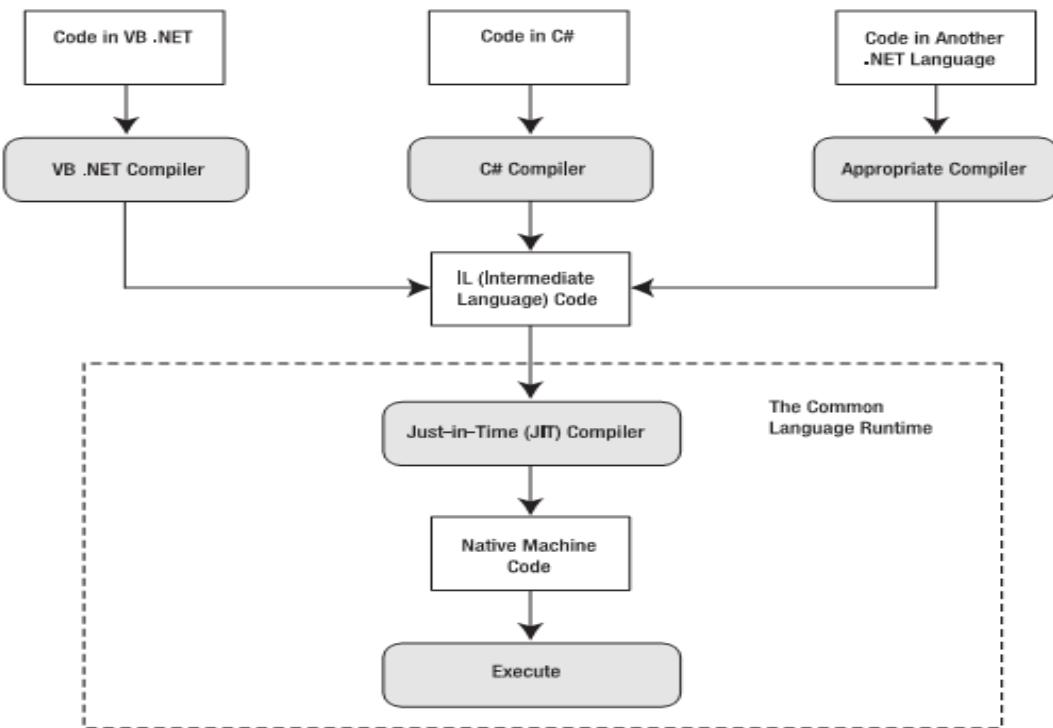
.Net Framework cung cấp một tập hợp rất lớn các hàm chức năng được xây dựng sẵn dưới dạng các lớp. Số lượng các lớp này hiện tại có hơn 10.000 kiểu khác nhau (bao gồm các lớp, cấu trúc, giao tiếp, và các thành phần lập trình chính khác). Tập hợp các lớp được nhóm thành các vùng làm việc Namespace. Các vùng làm việc khác nhau cung cấp chức năng khác nhau.

3.2. ASP.NET được biên dịch, không thông dịch

Các ứng dụng ASP.net đều được biên dịch để thực thi các mã lệnh C# hoặc Visual Basic. Các ứng dụng này sẽ được thực thi qua 2 giai đoạn IL (Intermediate Language) và JIT (Just In Time) như sau:

- Giai đoạn thứ nhất IL: đây là giai đoạn biên dịch các mã lệnh C# thành ngôn ngữ trung gian gọi là Microsoft Intermediate Language (MSIL). Giai đoạn này được thực thi tự động khi trang web được gọi lần đầu tiên.

- Giai đoạn thứ 2 JIT: Giai đoạn này xảy ra ngay trước khi trang web được thực thi. Các mã lệnh IL ở giai đoạn đầu sẽ được chuyển sang mã máy.



Hình 4: Mô hình CLR trong .NET

3.3. ASP.NET hỗ trợ đa ngôn ngữ

Để có thể phát triển một ứng dụng web trên nền dotNet Framework, ta có thể sử dụng một trong các ngôn ngữ như sau: C#, J#, Visual Basic. Cho dù lập trình viên lựa chọn ngôn ngữ nào đi nữa để phát triển ứng dụng, tất cả các mã lệnh đều được biên dịch sang mã IL. Lý do là các mã ASP.net đều được lưu trữ và thực thi trên server.

3.4. ASP.NET thực thi ở giai đoạn Common Language Runtime

Đây một đặc điểm giúp ứng dụng ASP.net thực hiện nhanh các yêu cầu từ người dùng bởi vì ASP.net không cần phải biên dịch lại trang web mỗi khi có yêu cầu từ trình khách.

3.5. ASP.NET hướng đối tượng

ASP.net cung cấp mô hình đối tượng tương đối yếu. Nó cung cấp một tập hợp các đối tượng với chi tiết thô về HTTP và HTML. Mặt khác, ASP.net thì thuần hướng đối tượng. Lập trình viên hoàn toàn có thể truy xuất vào tất cả các lớp của .net Framework và khai thác toàn bộ các ý tưởng trong môi trường OOP như: cài đặt giao tiếp, mở rộng các lớp hiện có với kế thừa, và các hàm chức năng khác.

3.6. ASP.NET đa thiết bị và đa trình duyệt

Một trong các thử thách lớn nhất của các nhà phát triển web là sự đa dạng của trình duyệt web. Các trình duyệt, ấn bản và cấu hình khách nhau hỗ trợ HTML khác nhau. Vì vậy, họ cần phải chọn nội dung trả về trình duyệt dưới dạng HTML 3.2, HTML 4.0 hoặc XHTML 1.0 hoặc ngay cả WML (Wireless Markup Language) cho ứng dụng di động. Tương tự như vậy, các lập trình viên web cũng gặp khó khăn khi viết các lệnh javascript để kiểm tra nhập liệu hay tạo trang web động.

ASP.net giải quyết các khó khăn trên bằng cách tạo ra các server controls mà có thể trả về các thẻ HTML tương thích. Ví dụ: các điều khiển kiểm tra nhập liệu (Validation controls) sử dụng JavaScripts và DHTML để tăng cường ảnh hưởng nếu client hỗ trợ nó. Điều này giúp

cho các điều khiển Validation hiển thị các thông báo lỗi ở ngay phía trình duyệt mà không đòi hỏi trình client gửi thông tin về phía server như trước đây.

3.7. ASP.NET dễ dàng triển khai và cấu hình

Các lớp chính yếu trong ASP.net Framework thì giống nhau ở các bản khác nhau. Vì vậy, việc triển khai một ứng dụng web khá đơn giản. Đối với ứng dụng web bình thường, cơ bản chỉ cần sao chép toàn bộ từ mục web vào web server miễn là máy chủ web có .NET Framework. Tương tự như vậy, việc phân bổ các components vào ứng dụng cũng rất dễ dàng bằng cách chép các components cùng với các trang web khi thực hiện triển khai ứng dụng. Bởi vì tất cả các thông tin về component được lưu trữ trực tiếp trong các file metadata, ta không cần phải cấu hình ở Registry.

Việc cấu hình trang web trong ASP.net được thực hiện chủ yếu trong tập tin web.config. Tập tin web.config có định dạng XML. Ta có thể thay đổi cấu hình website bằng cách đơn giản là thay đổi nội dung trong tập tin web.config.

IV. ASP.NET 2.0

ASP.net 2.0 kế thừa và phát triển các chức năng đã có trong ASP.net 1.0 và ASP.net 1.1. Các điểm chính yếu của ASP.net 2.0.

- **Cung cấp thêm nhiều controls:** ASP.net 2.0 giới thiệu thêm hơn 40 controls mới. Một trong những số đó là TreeView và Menu thực hiện trên JavaScript.
- **Master pages:** Lập trình viên có thể sử dụng chức năng này để tạo trang web với cấu trúc chung như: header, footer, và các navigations controls.
- **Themes:** Cho phép các nhà phát triển web thiết đặt các thuộc tính định dạng cho các web controls. Ta có thể áp dụng các định dạng này với tất cả các web controls ở tất cả các trang khác nhau. Chức năng này giúp cho website có một giao diện thân thiện người dùng.
- **Điều khiển nguồn dữ liệu:** Mô hình nguồn dữ liệu cho phép ta định nghĩa trang web giao tiếp với các nguồn dữ liệu một cách trực quan thông qua các đánh dấu hơn là phải viết các lệnh truy xuất dữ liệu.
- **Web parts:** Một kiểu ứng dụng web phổ biến là portal để tập trung các thông tin khác nhau bằng cách sử dụng các phần tách rời trên trang web.
- **Profiles:** Chức năng này cho phép bạn lưu trữ thông tin chi tiết của người dùng vào cơ sở dữ liệu. Thay vì, ASP.net quan tâm đến công việc khó khăn để truy xuất thông tin dữ liệu khi cần thiết và lưu trữ dữ liệu khi thay đổi.

Câu hỏi:

1. Dotnet Framework 3.5 là gì?
2. Trình bày được các giai đoạn của phát triển ứng dụng web?
3. Hãy nêu một số điểm chính trong ASP.net 3.5?

Chương 2: WEB FORM

Mục tiêu học tập:

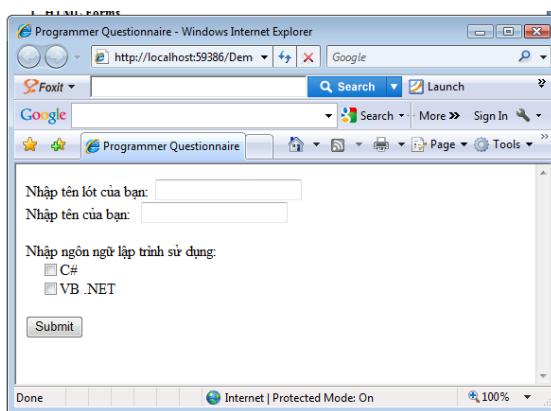
- Trình bày các thành phần cơ bản trong web form.
- Trình bày mô hình sự kiện trong ASP.net
- Mô tả các giai đoạn thực hiện của một web form
- Vận dụng một số controls trong ASP.net để tạo form nhập liệu
- Thực hiện chuyển tiếp thông tin giữa các trang web.

I. HTML Forms

HTML forms là công cụ rất quen thuộc với người lập trình web. Để người dùng có thể gửi thông tin về server, thẻ form là một thẻ chính yếu để thực hiện công việc này. Ví dụ: đoạn mã lệnh sau đây dùng để tạo một form gồm có 2 ô nhập liệu text, 2 ô check box và 1 nút lệnh submit. Khi người dùng nhấn chọn nút lệnh submit, toàn bộ thông tin được nhập vào textbox, và checkbox sẽ chuyển về server. Kết tiếp trang web page.aspx là trang web có nhiệm vụ xử lý các thông tin được nhận ở server.

```
<form method="post" action="page.aspx">
<div>
    Nhập vào họ và tên lót: <input type="text" name="FirstName" />
    Nhập vào tên: <input type="text" name="LastName" />
    Bạn có khả năng lập trình:
        <input type="checkbox" name="CS" />C#
        <input type="checkbox" name="VB" />VB .NET
        <input type="submit" value="Gửi thông tin" id="OK" />
</div>
</form>
```

Kết quả của đoạn mã lệnh trên là:



Hình 5: Form kết quả của ví dụ HTML Form

Khi người dùng click chọn nút lệnh submit, trình duyệt sẽ chọn giá trị hiện tại của mỗi control và chuyển nó cùng với chuỗi lệnh như sau:

FirstName=Matthew&LastName=MacDonald&CS=on&VB=on

Trong ASP, ASP.net cho phép ta truy xuất các giá trị từ form thông qua đọc tập dữ liệu được gửi Request.Form. Ta có thể sử dụng đoạn mã lệnh như sau:

```

        string firstName = Request.Form["FirstName"];
    Hoặc ta cũng có thể sử dụng thuộc tính Text của textbox để lập trình.
        string firstName = txtFirstName.Text;

```

II. Mô hình sự kiện trong ASP.NET

ASP.net cung cấp một thay đổi hoàn toàn mới với mô hình hướng sự kiện. Ở mô hình này, ta chỉ cần thêm một control vào webform và xác định sự kiện nào được gắn với nó mỗi khi có một thay đổi trên control giống như lập trình ứng dụng trên windows.

Một số bước chính trong mô hình hướng sự kiện:

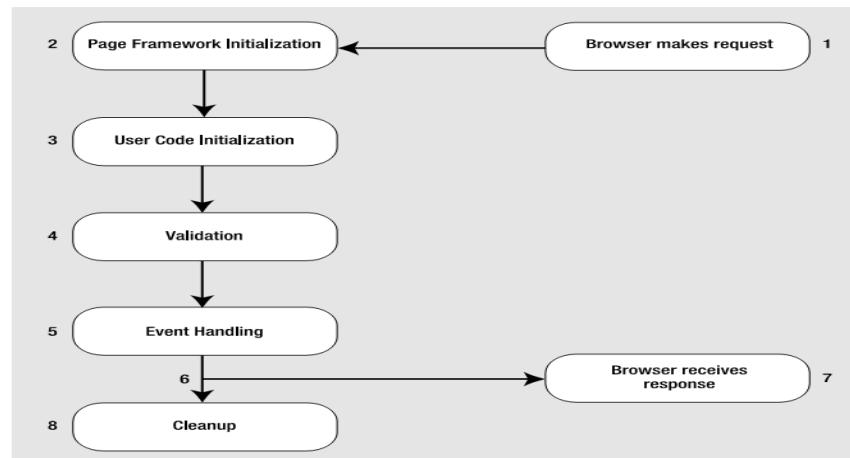
1. Khi trang web được gọi lần đầu tiên. ASP.net tạo đối tượng trang và đối tượng controls, khởi tạo mã lệnh thực thi và trả về thẻ HTML về phía trình duyệt. Đối tượng trang sau đó cũng được giải phóng khỏi vùng nhớ của server.
2. Ở trình khách, người dùng có thể kích hoạt một vài postback, như click vào nút lệnh, hoặc thay đổi chọn lựa trong listbox nào đó. Tất cả dữ liệu trên form sẽ được chuyển về server.
3. ASP.net tiếp tục tạo lại đối tượng trang một lần nữa
4. Kế tiếp, ASP.net kiểm tra các thao tác được gọi bởi postback và gọi sự kiện tương ứng. Ví dụ như cập nhật dữ liệu vào cơ sở dữ liệu hoặc đọc dữ liệu từ tập tin và sau đó chỉnh sửa lại các đối tượng để hiển thị thông tin.
5. Trang web sau đó được chỉnh sửa và trả về thẻ HTML. Đối tượng trang được giải phóng khỏi bộ nhớ. Nếu có một postback nào đó xảy ra, ASP.net sẽ thực hiện lại các bước từ 2-4.

Automatic Postback

Các web controls trong ASP.net cung cấp đặc tính postback tự động. Với đặc tính này, các điều khiển nhập liệu có thể kích hoạt các sự kiện như sau, và các mã lệnh server có thể đáp ứng ngay lập tức. Ví dụ, khi ta kích hoạt một postback bằng cách click vào checkbox, thay đổi một mục chọn trong danh sách, hoặc thay đổi ô text trong textbox và sau đó di chuyển đến các thành phần thông tin khác.

III. Các giai đoạn xử lý của một Web Form

Liên quan đến việc xử lý thông tin trên web server. Thông tin trên web form xảy ra dưới dạng giai đoạn. Các giai đoạn chính của xử lý một trang ASP.NET có thể được mô tả như hình sau:



Hình 6: Chu trình sống của một trang web

3.1. Khởi tạo Page framework – Framework Init

Trong giai đoạn này ASP.NET tạo trang web. Nó phát sinh ra tất cả controls được định nghĩa trước đó trong trang web có phần mở rộng .aspx. Thêm vào đó, nếu trang web không được yêu cầu lần đầu tiên (nó được postback), ASP.NET phát sinh các thông tin view state và áp dụng nó vào tất cả các controls.

Vào lúc này, sự kiện Page.Init được kích hoạt. Tuy nhiên, sự kiện này ít được quản lý bởi trang, bởi vì nó vẫn còn quá sớm để thực hiện khởi tạo trang. Đó là bởi vì các đối tượng controls có thể không được khởi tạo và thông tin về view state chưa được nạp.

3.2. Khởi tạo mã người dùng - User code

Trong giai đoạn này, sự kiện PageLoad được kích hoạt. Hầu hết các trang web quản lý sự kiện này để thực hiện các khởi tạo cần thiết. Sự kiện PageLoad luôn luôn kích hoạt bất chấp trang web có được yêu cầu lần đầu tiên hoặc trang được yêu cầu như một phần của postback.

Để xác định trạng thái hiện thời của trang, bạn có thể kiểm tra thuộc tính IsPostBack của trang sẽ là false trong lần đầu tiên trang web được gọi. Ví dụ

```
if (!IsPostBack)  
{  
    FirstName.Text = "Nhập tên của bạn vào đây";  
}
```

3.3. Kiểm tra – Validation

ASP.net có các điều khiển kiểm tra nhập liệu để kiểm tra các dữ liệu do người dùng nhập vào và hiển thị thông báo lỗi.

3.4. Quản lý sự kiện - Event handling

Những sự kiện này kích hoạt ngay lập tức đối với web controls nếu AutoPostBack là true. Ngược lại, chúng kích hoạt lần kế tiếp khi trang web được post back.

- **Sự kiện đáp ứng ngay lập tức:** Click vào nút submit, click vào một vài button khác, click vào khu vực hình ảnh, hoặc click vào liên kết trong một web control,
- **Change events:** Thay đổi chọn lựa trong một control hoặc một ô text hay một textbox, AutoPostBack = true được kích hoạt ngay lập tức, AutoPostBack = false: được kích hoạt lần kế tiếp

Ví dụ: Ta có một trang web gồm có một textbox và một nút lệnh. Sau đó ta nhập liệu vào textbox trên và click nút lệnh được cho. Vào lúc này, ASP.net sẽ gọi thực thi một vài sự kiện theo thứ tự như sau:

- Page.Init
- Page.Load
- TextBox.TextChanged
- Button
- Page.PreRender
- Page.Unload

Ví dụ về Page Flow ở mục 3.7 sẽ nói rõ hơn về phần nội dung này.

3.5. Liên kết dữ liệu tự động - data binding

Trong các chương tiếp theo, ta sẽ nghiên cứu thêm các chức năng liên kết dữ liệu rất hữu ích trong ASP.net. Bất kỳ thay đổi nào trên nguồn dữ liệu (ví dụ: insert, delete, or

updates) được thực hiện trước khi sự kiện control Page.PreRender được kích hoạt. Sau đó, khi sự kiện Page.PreRender kích hoạt, nguồn dữ liệu thực hiện các truy xuất dữ liệu và thêm dữ liệu được truy xuất vào các điều khiển được gắn vào.

3.6. Dọn dẹp - Cleanup

Đây là giai đoạn cuối cùng của một trang web được xử lý ở phía trình chủ. Sau khi trang web được trả về với thẻ HTML, server sẽ thực hiện quá trình dọn dẹp bằng cách gọi Page.Unload. Vào lúc này, đối tượng trang vẫn còn nhưng trang HTML không thể thay đổi được nữa.

3.7. Ví dụ về PageFlow

Ví dụ về PageFlow sẽ minh họa một cách chi tiết các bước thực hiện ở trang web từ khi được gọi lần đầu tiên đến khi kết thúc trang web.

Trang PageFlow.aspx

```
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label id="lblInfo" runat="server" EnableViewState="false">
            </asp:Label>
            <asp:Button id="Button1" runat="server" OnClick="Button1_Click"
Text="Click here!">
            </asp:Button>
        </div>
    </form>
</body>
```

Trang PageFlow.aspx.cs

```
private void Page_Load(object sender, System.EventArgs e)
{
    lblInfo.Text += "Gọi lần 2";
    lblInfo.Text += "Page.Load được nạp.<br />";

    if (Page.IsPostBack)
    {
        lblInfo.Text +=
            "<b>Sau khi kích hoạt một sự kiện.</b><br />";
    }
}

private void Page_Init(object sender, System.EventArgs e)
{
    lblInfo.Text += "Gọi lần 1";
    lblInfo.Text += "Page.Init được nạp.<br />";

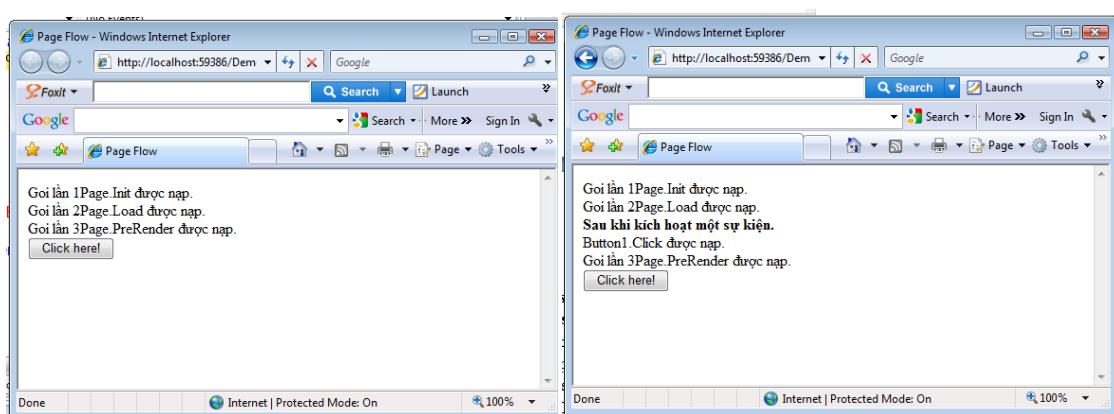
}

private void Page_PreRender(object sender, System.EventArgs e)
{
    lblInfo.Text += "Gọi lần 3";
    lblInfo.Text += "Page.PreRender được nạp.<br />";
}

private void Page_Unload(object sender, System.EventArgs e)
{
    lblInfo.Text += "Page.Unload được nạp.<br />";
}

protected void Button1_Click(object sender, System.EventArgs e)
{
    lblInfo.Text += "Button1.Click được nạp.<br />";

}
```



Hình 7: Form Kết quả của PageLayout

IV. Đối tượng trang Page

4.1. Đối tượng Session, Application và Cache

Session là một thực thể của lớp System.Web.SessionState.HttpSessionState. Lưu trữ bất kỳ kiểu dữ liệu của người dùng cần để duy trì các yêu cầu giữa các trang. Đối tượng Session cung cấp truy xuất kiểu từ điển để đặt các cặp name/value thể hiện dữ liệu của người dùng trong session trên. Trạng thái của session được dùng để duy trì các thông tin như user's name, user's ID, a shopping cart, hoặc những thành phần khác bị bỏ khi người dùng không còn truy xuất website nữa.

Application: Lưu trữ đối tượng toàn cục được truy xuất bởi bất kỳ client nào và nó cũng có dữ liệu name/value. Application là một đối tượng thuộc lớp System.Web.HttpApplicationState. Ví dụ phổ biến là bộ đếm toàn cục để theo dõi bao nhiêu lần hệ điều hành thực hiện bởi tất cả các clients.

Cache: Là một thực thể của lớp System.Web.Caching.Cache. Nó cũng lưu trữ các thông tin toàn cục dưới dạng cặp name/value. Một ưu điểm của Cache là cung cấp cơ chế lưu trữ tốt hơn nhiều bởi vì ASP.NET có thể loại bỏ các đối tượng nếu bộ nhớ máy chủ khan hiếm.

4.2. Request

Đối tượng Request là một thể hiện của lớp System.Web.HttpRequest. Đối tượng này thể hiện giá trị và thuộc tính của các yêu cầu HTTP được tạo ra khi trang được load. Nó lưu trữ tất cả các tham số URL và tất cả các thông tin được gửi bởi trình khách. Nhiều này thể hiện giá trị và thuộc tính của HTTP request khi trang web được load. Nó chứa địa chỉ URL và tất cả thông tin được chuyển từ client.

Thuộc tính	Ý nghĩa
Browser	Chứa thông tin mô tả các đặc điểm của các browser khác nhau.
Form	Tập hợp các biến form được chuyển đến trang. Trong tất cả các trường hợp, bạn sẽ phải truy xuất thông tin này từ thuộc tính control thay vì sử dụng collection
QueryString	Cung cấp các tham số được truyền cùng với chuỗi querystring.
FilePath	Cho biết đường dẫn vật lý của tập tin
CurrentExecutionFilePath	

4.2. Response

Đối tượng Response là một thực thể của lớp System.Web.HttpResponse. Response dùng để gửi phản hồi của server đối với yêu cầu của client. Trong ASP cổ điển, đối tượng Response là công cụ chủ yếu để gửi file HTML về trình duyệt máy khách.

Ví dụ:

Response.Redirect("newpage.aspx"); *Chuyển đến một file trong cùng thư mục*

Response.Redirect("http://www.prosetech.com"); *Chuyển đến một website khác.*

Thuộc tính	Ý nghĩa
Cache	Cho phép cấu hình output caching
Cookies	Tập hợp các cookies được gửi theo response. Ta cũng có thể sử dụng thuộc tính này để thêm vào cookies
Redirect()	Phương thức này cho phép người dùng chuyển đến trang web khác trong cùng ứng dụng hoặc website khác

4.3. Đối tượng Server

Đối tượng Server là một thực thể của lớp System.Web.HttpServerUtility. Nó cung cấp rất nhiều phương thức và thuộc tính.

Phương thức	Ý nghĩa
MapPath()	Cho biết đường dẫn tập tin vật lý tương ứng với đường dẫn ảo trên webserver
Transfer()	Chuyển thực thi đến trang web khác trong cùng ứng dụng. Phương thức này tương tự như phương thức Response.Redirect() nhưng nó thực hiện nhanh hơn. Phương thức Transfer() không thể được dùng để chuyển sang các webserver hoặc trang web không ở dạng asp.net

Ví dụ:

Server.Transfer("newpage.aspx"): Chuyển đến tập tin trong cùng thư mục web

Server.Transfer("http://www.tvu.edu.vn"): Xảy ra lỗi vì không thể chuyển sang website khác.

```
String physicalPath=Server.MapPath("info.txt");
StreamReader reader= new StreamReader(physicalPath);
reader.Close();
```

4.4. Đối tượng User

Đối tượng User thể hiện thông tin về người dùng gửi yêu cầu đến web server, và nó cho phép ta kiểm tra vai trò của người dùng. Đối tượng User cài đặt giao tiếp System.Security.Principal.IPrincipal. Lớp này phụ thuộc vào kiểu nhận dạng ta đang dùng. Ví dụ, ta có thể nhận dạng người dùng dựa trên thông tin tài khoản Windows sử dụng IIS thông qua nhận dạng dựa trên cookie cùng với trang login.

4.5. Đối tượng Trace

Đối tượng Trace là một thể hiện của lớp System.Web.TraceContext. Nó cho phép ta ghi thông tin đến nhật ký ở mức trang. Nhật ký này có thông tin chi tiết để debug các đối tượng Trace.

Bài tập chương 2

Lab 2.1. Sử dụng HTML Control trong ASP.NET

1. Mục tiêu

- Tích hợp HTML Control vào trang ASP.NET
- Tạo trình xử lý sự kiện cho page và các server controls trên ASP.NET page.

2. Yêu cầu

- Trước khi thực hiện bài thực hành này, sinh viên phải biết cách tạo trang HTML.
- Sinh viên phải sử dụng được CSS

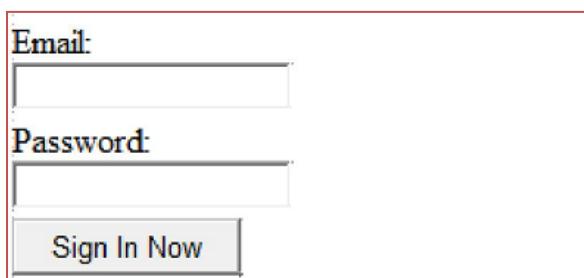
3. Kịch bản

Người sử dụng nhập vào **địa chỉ email** và **password** để sign in trong hệ thống. Thông thường là trang Login.aspx. Trong bài thực hành này sinh viên sẽ tạo trang ASP.NET đầu tiên với HTML server control rồi sau đó viết trình xử lý sự kiện của button. Những thông tin user gửi tới server sẽ được trả về client thông qua thành phần HTML span.

4. Tạo trang ASP.NET

Trong phần này sinh viên sẽ tạo ASP.NET page đơn giản chỉ có duy nhất 1 form, hai **HTML textbox**, **1 HTML button** và **1 thành phần HTML **.

1. Sử dụng **Visual Studio .NET** tạo một **Web project** có tên là Lab1
 - a. Trong menu **File**, chọn **New**, chọn **Project**
 - b. Trong hộp thoại **New Project**, chọn **ASP.NET Web Application**, thiết lập tên là **Lab1**, xác nhận đường dẫn lưu trữ các file trong project ở Location, sau đó xác nhận OK để hoàn tất việc khởi tạo project.
 - e. Sử dụng các HTML control để tạo form như hình minh họa sau, lưu ý tag **** được thêm vào sau button **Sign In Now**.



Hình 3: Giao diện minh họa của Web Form Default.aspx. Bảng mô tả các controls trên web form:

Control	Attribute
Email	type =”text” id = “txtEmail”
Password	type =”password” id = “txtPassword”
Button	type =”submit” value =”Sign In Now”
Span	id = “spnInfo”

2. Thêm sự kiện cho button “Sign In Now”

- a. Thiết lập thuộc tính **onServerClick** của button là **cmdLogin_Click**
- b. Trong file **.cs** của web form cài đặt hàm **cmdLogin_Click** như sau:
- c. Sử dụng thẻ span để hiện thị thông tin đăng nhập gồm Email và password do người dùng nhập vào

Lab 2.2: Tạo Form Đăng Ký

Bài tập 2: Tạo một Web Form đăng ký thông tin tuyển dụng cơ bản

- Trong phần bài tập này sinh viên tạo một ASP.NET Web Form có tên **JobRegister.aspx**. Form chứa các server control dạng **Web Control** hoặc **HTML Server Control**. Form cho phép user nhập các thông tin cá nhân các kỹ năng chuyên môn.

- Giao diện của Form được minh họa như hình phía bên dưới:
- Sau khi hoàn tất việc đăng ký thông tin này sẽ được hiển thị đầy đủ trong một trang ASP.NET khác là **RegisteringComplete.aspx**.

ĐĂNG KÝ THÔNG TIN VIỆC LÀM

Thông tin cá nhân

* Họ & tên :

* Ngày tháng năm sinh :

Giới tính :

Tình trạng hôn nhân :

Thông tin liên lạc

* Địa chỉ :

* Tỉnh/Thành phố :

Số điện thoại :

Số điện thoại di động :

* Email :

Trình độ học vấn

* Trình độ học vấn :

* Thông tin về học vấn :

Ngoại ngữ :

Kỹ năng :

Kinh nghiệm làm việc

Số năm kinh nghiệm :

Kinh nghiệm làm việc :

Mong muốn của bản thân

Việc làm mong muốn :

* Mức lương thỏa thuận : VNĐ

Chương 3: SERVER CONTROLS

Bài 1: HTML Server Controls

Mục tiêu học tập:

- Trình bày các thành phần cơ bản của điều khiển HTML ở trình chủ trong ASP.net.
- Tạo form nhập liệu với HTML server Controls
- Thực hiện chuyển tiếp thông tin giữa các trang web.
- Cài đặt các sự kiện với HTML Server Controls

I. Giới thiệu

ASP.net cung cấp rất nhiều server controls khác nhau được phân chia thành nhiều nhóm. Trong chương này chỉ đề cập đến một số nhóm chính cần thiết trong quá trình xây dựng một ứng dụng web.

- **HTML server controls:** Là các lớp đóng gói các thẻ HTML chuẩn. Ngoại trừ đặc điểm trên, các khai báo cho HTML server control giống như trong HTML. Hai ví dụ bao gồm HtmlAnchor (đối với thẻ <a>) và HtmlSelect (đối với thẻ <select>). Tuy nhiên ta vẫn có thể chuyển thẻ HTML vào trong server control. Nếu không có lớp tương ứng, ASP.net sẽ sử dụng lớp HtmlGenericControl. Để chuyển các thẻ HTML bình thường sang server control, ta chỉ đơn giản thêm thuộc tính runat="server" vào các thẻ.

- **Web controls:** Các lớp này cung cấp tập các thuộc tính và phương thức giúp các lập trình viên dễ khai báo và truy xuất. Ví dụ: HyperLink, Listbox, và Button là một trong những ví dụ của web controls

- **Rich controls:** Các controls này có thẻ tự động phát sinh ra một số lượng lớn thẻ HTML và ngay cả các thẻ Javascript để tạo giao diện. Một số rich controls tiêu biểu như: Calendar, Adrotator và TreeView controls.

- **Validation controls:** Tập các controls cho phép bạn dễ dàng kiểm tra điều khiển nhập liệu có thỏa quy định ràng buộc dữ liệu hay không. Ví dụ: Để kiểm tra ô nhập liệu nào đó bắt buộc phải nhập liệu. Nếu không thỏa điều kiện này, người dùng có thể ngăn chặn các xử lý tiếp theo bằng cách tạo ra các thông báo lỗi tại ô nhập liệu tương ứng.

- **Data Controls:** Các điều khiển dạng này bao gồm các tables và các danh sách tương đối phức tạp để hiển thị tập hợp lớn các dữ liệu. Các hỗ trợ trong data controls rất mạnh như template, chỉnh sửa, sắp xếp và phân trang. Nó cũng hỗ trợ kết buộc dữ liệu với các nguồn dữ liệu khác nhau mà không cần sử dụng các câu lệnh lập trình phức tạp.

- **Navigation controls:** Được thiết kế để hiển thị site maps và cho phép người dùng di chuyển từ trang này sang trang khác.

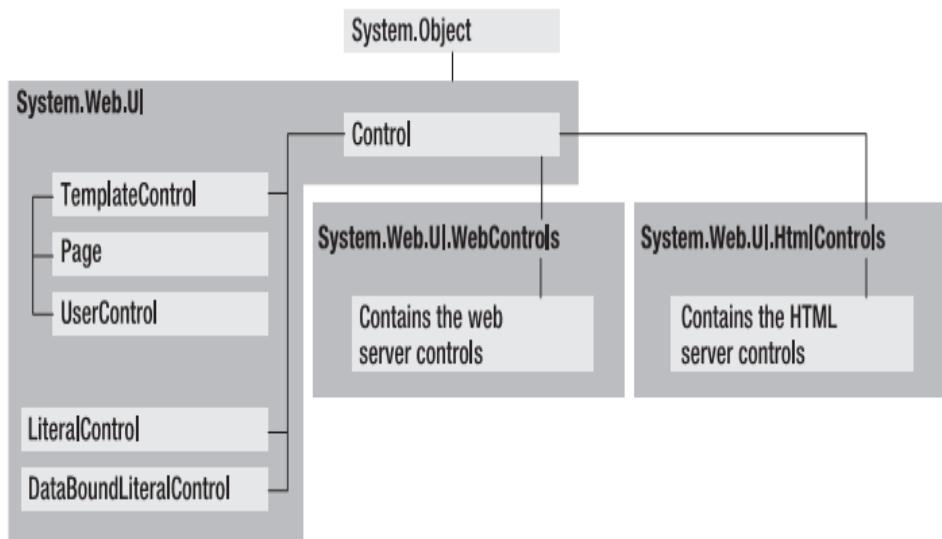
- **Login controls:** Điều khiển này hỗ trợ nhận diện forms bao gồm thông tin như cơ sở dữ liệu và trạng thái người dùng.

- **Web parts:** Xây dựng các components và cấu hình các webportals

- **ASP.NET Ajax controls:** Các điều khiển này cho phép người dùng sử dụng kỹ thuật Ajax trong các trang web để viết các mã lệnh phía người dùng.

- **ASP.net mobile controls:** Tập hợp các web controls được cấu hình để hỗ trợ các clients như PDAs, smart phones bằng cách trả về các trang với mã HTML 3.2 hoặc WML1.1. Bạn có thể tìm hiểu tại <http://www.asp.net/mobile>.

II. Lược đồ lớp Server Control



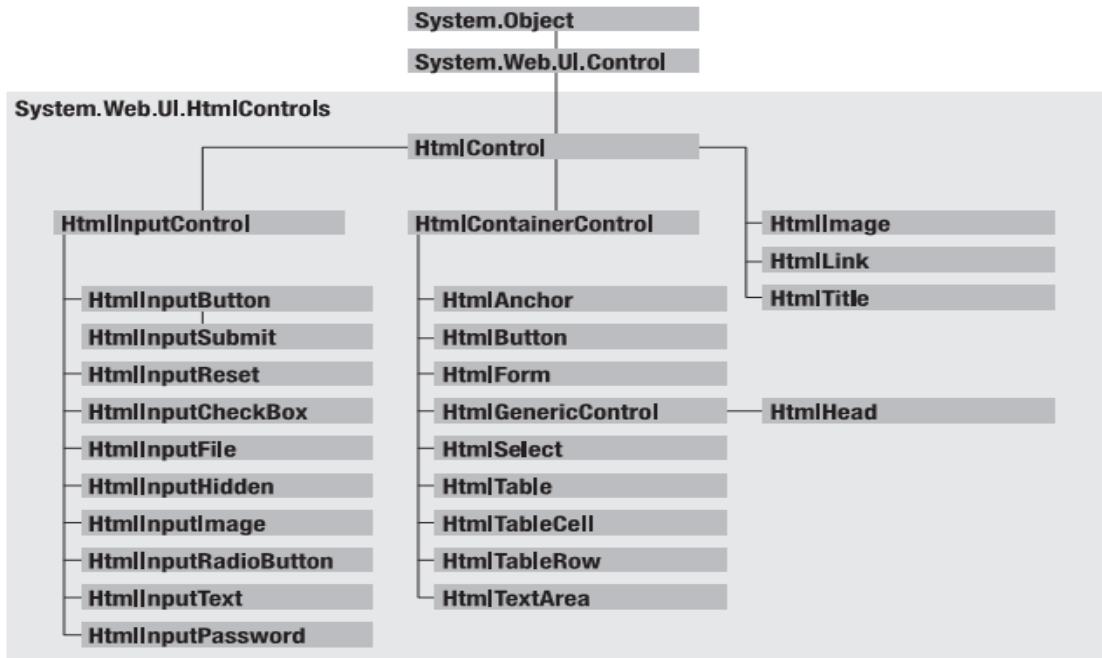
Hình 8: Lược đồ lớp Server Controls

Thuộc tính	Ý nghĩa
ClientID	Cho biết nhận dạng của control với tên được tạo ra bởi ASP.net ngay khi trang được khởi tạo
Controls	Cho biết tập hợp các controls con cháu. Sử dụng Page.Controls để lấy tập hợp các controls của trang.
ID	Cho biết hoặc gán tên của control. Trong thực tế, tên này được sử dụng để truy xuất các controls trong các đoạn mã phía trình chủ hoặc ở code behind class
Visible	Mặc định là true. Nếu là false: không xuất hiện ở phía trình khách
Page	Cho biết trang web có chứa controls

Các thuộc tính chính của lớp Control

III. HTML server controls

3.1. Lược đồ tổng quát của HTML Server Controls



Hình 9: Các loại HTML Server Controls

Bảng sau đây liệt kê tất cả các HTML server controls, các thuộc tính cụ thể và các sự kiện có thể có. Khai báo HTML server controls rất đơn giản bằng cách thêm vào thuộc tính runat="server". Đây là thuộc tính cho phép ASP.NET xử lý và biên dịch thành các đối tượng tương ứng với các lớp trong .NET. Vì vậy, HTML server control phù hợp với các ứng dụng chuyển từ HTML hoặc ASP sang ASP.net.

Tên thẻ	Lớp .net	Thuộc tính
	HtmlAnchor	HRef, Target, Title, Name, ServerClick event
<button runat="server">	HtmlButton	CausesValidation, ValidationGroup, ServerClick event
<form runat="server">	HtmlForm	Enctype, Method, Target, DefaultButton, DefaultFocus
	HtmlImage	Align, Alt, Border, Height, Src, Width
<input type="button" runat="server">	HtmlInputButton	Type, Value, CausesValidation, ValidationGroup, ServerClick event
<input type="reset" runat="server">	HtmlInputReset	Type, Value
<input type="submit" runat="server">	HtmlInputSubmit	Type, Value, CausesValidation, ValidationGroup, ServerClick event
<input type="checkbox" runat="server">	HtmlInputCheckBox	Checked, Type, Value, ServerClick event
<input type="file" runat="server">	HtmlInputFile	Accept, MaxLength, PostedFile, Size, Type, Value
<input type="hidden" runat="server">	HtmlInputHidden	Type, Value, ServerChange event

Tên thẻ	Lớp .net	Thuộc tính
<input type="image" runat="server">	HtmlInputImage	Align, Alt, Border, Src, Type, Value, CausesValidation, ValidationGroup, ServerClick event
<input type="radio" runat="server">	HtmlInputRadioButton	Checked, Type, Value, ServerChange event
<input type="text" runat="server">	HtmlInputText	MaxLength, Type, Value, ServerChange event
<select runat="server">	HtmlSelect	Multiple, SelectedIndex, Size, Value, DataSource, DataTextField, DataValueField, Items (collection), ServerChange event
<table runat="server">, <td runat="server">	HtmlTable	Align, BgColor, Border, BorderColor, CellPadding, CellSpacing, Height, NoWrap, Width, Rows (collection)
<th runat="server">	HtmlTableCell	Align, BgColor, Border, BorderColor, ColSpan, Height, NoWrap, RowSpan, VAlign, Width
<tr runat="server">	HtmlTableRow	Align, BgColor, Border, BorderColor, Height, VAlign, Cells (collection)
<textarea runat="server">	HtmlTextArea	Cols, Rows, Value, ServerChange event

3.2. Quản lý sự kiện phía Server

HTML server controls cung cấp một vài mô hình sự kiện như: ServerClick và ServerChange. Sự kiện ServerClick event được xử lý ở trình chủ. Sự kiện này thường xảy ra với các nút lệnh buttons và nó cho phép thực thi hành động ngay lập tức.

Sự kiện ServerChange xảy ra khi có một thay đổi nào đó ở ô textbox hoặc các điều khiển chọn lựa. Sự kiện này không xảy ra cho đến khi trang được gửi trở lại (ví dụ, sau khi người dùng click vào nút lệnh submit). Lúc này, sự kiện ServerChange xảy ra đối với tất cả các điều khiển được theo sau bởi ServerClick tương

Bảng sau đây cho biết controls nào có thể cài đặt sự kiện ServerClick hoặc ServerChange

Sự kiện	Controls có thể áp dụng
ServerClick	HtmlAnchor, HtmlButton, HtmlInputButton, HtmlInputSubmit, HtmlInputReset, HtmlInputImage
ServerChange	HtmlInputText, HtmlInputCheckBox, HtmlInputRadioButton, HtmlInputHidden, HtmlSelect, HtmlTextArea

3.3. Ví dụ:

Nội dung của tập tin ServerEvent.aspx như sau:

```
<body>
    <form id="form1" runat="server" >
        <div>
            <select runat="server" id="List1" size="5" multiple="true"
                    OnServerChange="List1_ServerChange">
                <option>Lập trình C</option>
                <option>Lập trình Java</option>
            </select>
        <br/>
        Khác: <input type="text" runat="server" ID="Textbox1" size="10"
                    OnServerChange="Ctrl_ServerChange"/>
        <input type="checkbox" runat="server" ID="Checkbox1"
                    OnServerChange="Ctrl_ServerChange" /> Chuyên nghiệp
        <input type="submit" runat="server" ID="Submit1" value="Submit Query"
                    OnServerClick="Submit1_ServerClick" />
    </div>
</form>
</body>
```

Nội dung của tập tin ServerEvent.aspx.cs như sau:

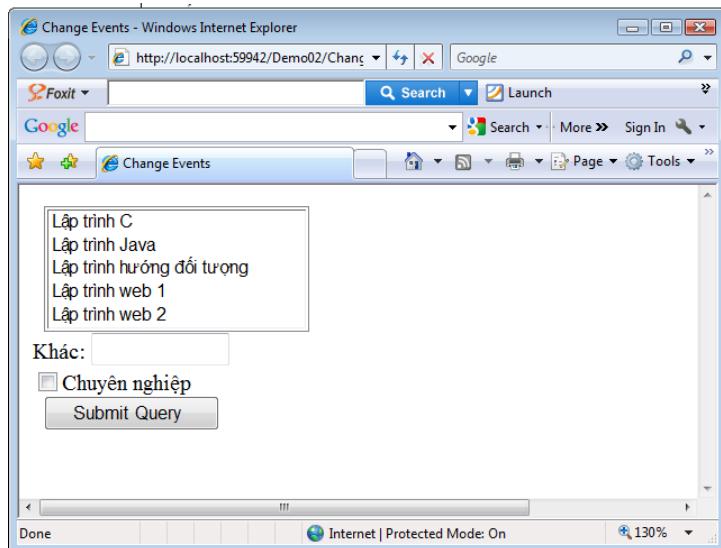
```
protected void Page_Load(object sender, System.EventArgs e)
{
    //Thêm vào List1 3 mục chọn khi trang web được gọi lần đầu tiên
    if (!Page.IsPostBack)
    {
        List1.Items.Add("Lập trình hướng đối tượng");
        List1.Items.Add("Lập trình web 1");
        List1.Items.Add("Lập trình web 2");
    }
}

protected void Ctrl_ServerChange(object sender, System.EventArgs e)
{
    Response.Write("<li>Sự kiện ServerChange được kích hoạt tại" +
        ((Control)sender).ID + "</li>");
}

protected void List1_ServerChange(object sender, EventArgs e)
{
    Response.Write("<li>Sự kiện ServerChange được kích hoạt tại List1. " + "Các mục
được chọn là:</li><br/>");
    foreach (ListItem li in List1.Items)
    {
        if (li.Selected)
            Response.Write("&nbsp;&nbsp;- " + li.Value + "<br/>");
    }
}

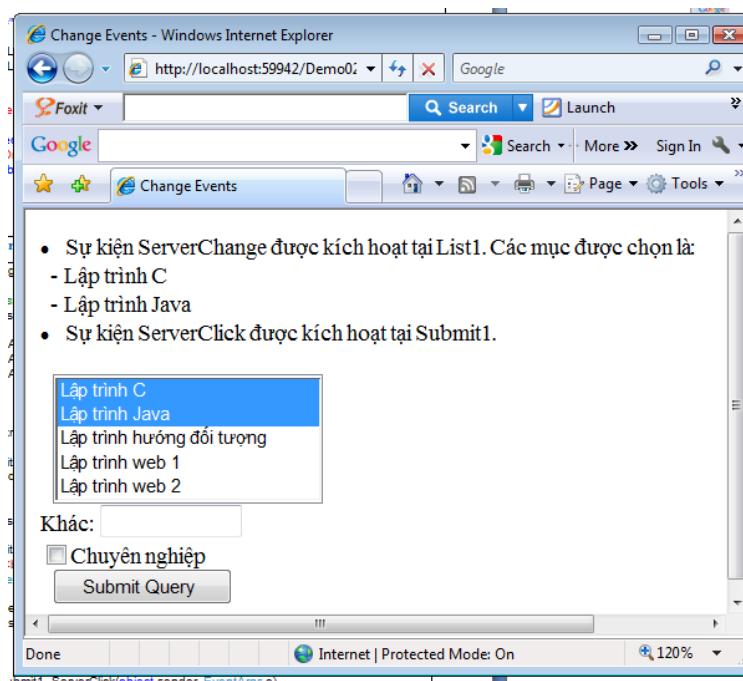
protected void Submit1_ServerClick(object sender, EventArgs e)
{
    Response.Write("<li>Sự kiện ServerClick được kích hoạt tại Submit1.</li>");
}
```

Kết quả của đoạn mã lệnh trên như hình bên dưới



Hình 10: Form kết quả Sự kiện trong HTML server controls

Sau khi người dùng chọn 2 mục chọn “Lập trình C” và “Lập trình Java”. Kết quả sẽ được như hình phía dưới.



Hình 11: Form kết quả Sự kiện trong HTML server controls

Câu hỏi củng cố:

1. Trình bày các thành phần cơ bản của điều khiển HTML ở trình chủ trong ASP.net.
2. Phân biệt sự khác nhau giữa HTML thông thường và HTML Server control trong ASP.net
3. Trình bày các cách thực hiện chuyển tiếp thông tin giữa các trang web.
4. Nêu các dạng sự kiện và các controls tương ứng trong HTML Server Controls
5. Thực hành bài tập Lab3.1 ở trang 43

Bài 2: WEB CONTROLS

Mục tiêu học tập:

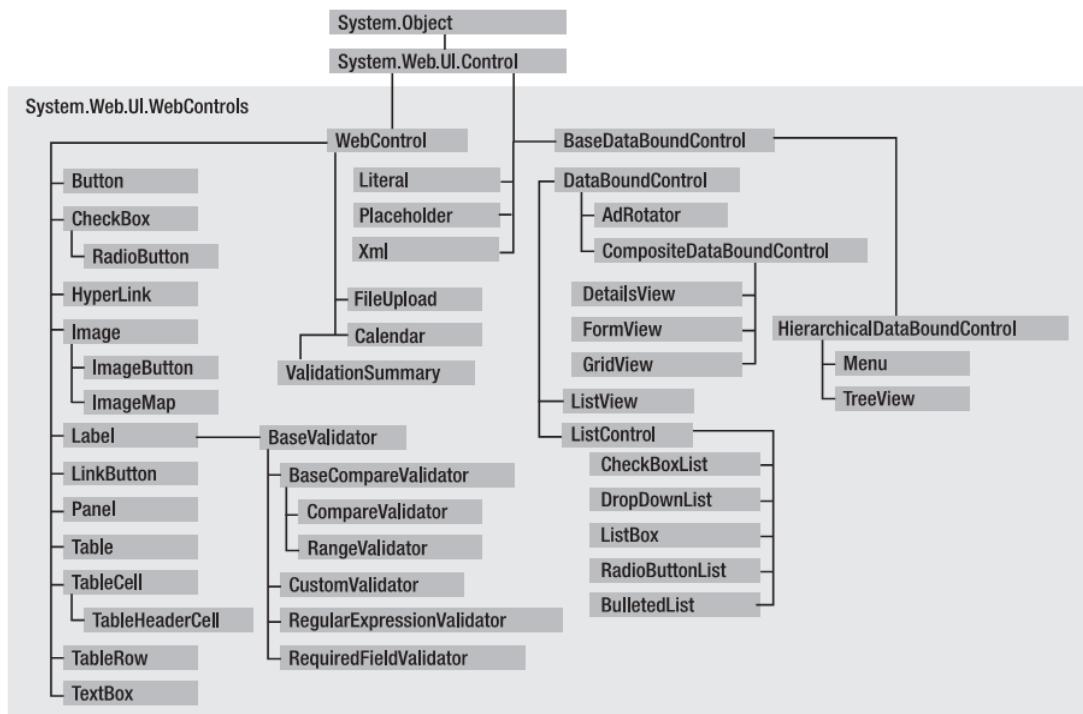
- Trình bày các thành phần cơ bản của điều khiển Web controls trong ASP.net.
- Tạo form nhập liệu với Web Controls
- Thực hiện chuyển tiếp thông tin giữa các trang web.
- Cài đặt các sự kiện với Web Controls

I. Giới thiệu về Web Controls

Được định nghĩa trong thư viện **System.Web.UI.WebControls** và được dẫn xuất từ lớp cơ sở **WebControl**. Cung cấp mô hình tương đối khá quan và trùu tượng hơn các HTML server controls. Web controls cung cấp thêm các tính năng nổi bật như postback tự động.

Các controls được mở rộng có thể tạo ra một số thẻ HTML và đoạn Javascript. Ví dụ: danh sách các check box, radio buttons, calendars, grids,...

Lược đồ lớp của các web Controls tổng quát



Hình 12: Lược đồ Web Controls

II. Lớp cơ sở WebControl

Tất cả các controls đều thừa kế từ lớp **WebControl**. **WebControl** được thừa kế từ lớp **Control**. Kết quả là nhiều thuộc tính và phương thức như **Controls**, **Visible**, và **FindControl()** thì giống như HTML server Controls. Tuy nhiên lớp **WebControl** có bổ sung thêm một số thuộc tính giống như bảng sau:

Thuộc tính	Ý nghĩa
AccessKey	Trả về hoặc đặt các phím tắt cho phép người dùng chuyển đến control nhanh hơn. Ví dụ khi gán AccessKey = A ta có thể chuyển đến control bằng cách nhấn Alt+A.

Thuộc tính	Ý nghĩa
BackColor	Trả về tập màu nền
BorderColor	Trả về màu đường kẻ
BorderStyle	Các giá trị có thể có là Dashed, Dotted, Double, Groove, Ridge, Inset, Outset, Solid, and None.
BorderWidth	Chiều rộng của đường
CssClass	Trả về CSS style để liên kết với control. CSS style có thể được định nghĩa trong style> section ở đầu trang hoặc tập tin CSS được tham chiếu bởi trang.
Enabled	Trả về hoặc gán trạng thái của control. Nếu Enabled=false, control được trả về là màu xám và không thể sử dụng.
Font	Trả về một đối tượng với tất cả thông tin kiểu của font được dùng đối với text control.
ForeColor	Trả về hoặc gán màu chữ của control. Ví dụ màu chữ của text box.
Height	Trả về hoặc gán chiều cao của controls.
TabIndex	Một con số cho phép bạn đặt thứ tự của control. Control có TabIndex=0 sẽ được gán vào khi trang được load lần đầu tiên.

III. Danh sách các web controls

Khai báo thẻ ASP.net	Thẻ HTML được trả về	Thuộc tính
<asp:Button>	<input type="submit"/> hoặc <input type="button"/>	Text, CausesValidation, PostBackUrl, ValidationGroup, Click event
<asp:CheckBox>	<input type="checkbox"/>	AutoPostBack, Checked, Text, TextAlign, CheckedChanged event
<asp:FileUpload>	<input type="file">	FileBytes, FileContent, FileName, HasFile, PostedFile, SaveAs()
<asp:HiddenField>	<input type="hidden">	Value
<asp:HyperLink>	<a>...	ImageUrl, NavigateUrl, Target, Text
<asp:Image>		AlternateText, ImageAlign, ImageUrl
<asp:ImageButton>	<input type="image"/>	CausesValidation, ValidationGroup, Click event
<asp:ImageMap>	<map>	HotSpotMode, HotSpots (collection), AlternateText, ImageAlign, ImageUrl
<asp:Label>	...	Text, AssociatedControlID
<asp:LinkButton>	<a>	Text, CausesValidation, ValidationGroup, Click event
<asp:Panel>	<div>...</div>	BackImageUrl, DefaultButton, GroupingText, HorizontalAlign, Scrollbars, Wrap
<asp:RadioButton>	<input type="radio"/>	AutoPostBack, Checked, GroupName, Text, TextAlign, CheckedChanged event
<asp:Table>	<table>...</table>	BackImageUrl, CellPadding, CellSpacing, GridLines, HorizontalAlign, Rows (collection)

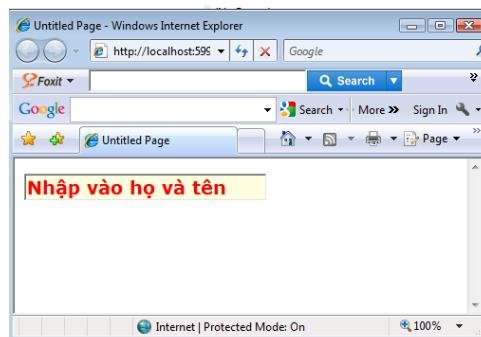
Khai báo thẻ ASP.net	Thẻ HTML được trả về	Thuộc tính
<asp:TableCell>	<td>...</td>	ColumnSpan, HorizontalAlign, RowSpan, Text, VerticalAlign, Wrap
<asp:TableRow>	<tr>...</tr>	Cells (collection), HorizontalAlign, VerticalAlign
<asp:TextBox>	<input type="text"/> or <textarea>...</textarea>	AutoPostBack, Columns, MaxLength, ReadOnly, Rows, Text, TextMode, Wrap, TextChanged event

Ví dụ:

- Trong ASP.net ta có khai báo một textbox như sau:

```
<asp:TextBox runat="server" ID="TextBox1" Text="Nhập vào họ tên" ForeColor="red"
BackColor="lightyellow" Width="250px" Font-Name="Verdana" Font-Bold="True" Font-
Size="20" />
```

- Sau khi trình chủ ASP.net biên dịch và chuyển về client với đoạn mã HTML như sau
- ```
<input name="TextBox1" type="text" value="Nhập vào họ và tên" id="TextBox1"
style="color:Red;background-color:LightYellow;font-family:Verdana;font-size:14pt;font-
weight:bold;width:250px;" />
```



Hình 13: Form Kết quả với Web Controls

## IV. Các List Controls

### 4.1. Một số List controls

- List controls trong Web controls có thể tạo ra các listbox, drop-down list và những điều khiển có thể lặp lại khác và cũng có thể liên kết vào dữ liệu (như cơ sở dữ liệu hoặc tập hợp giá trị khác như Collections).
- Hầu hết tất cả các list control cho phép người dùng chọn một hoặc nhiều mục chọn items.
- BulletedList thể hiện danh sách được đánh dấu hoặc bulleted cố định.
- Một số List controls được ASP.net cung cấp như sau:

| Thẻ                | Mô tả                                                                                                                                                                   |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <asp:DropDownList> | Một danh sách trong tập hợp các đối tượng <asp:ListItem>. Trong HTML, nó được trả về bởi thẻ <Select> với thuộc tính size="1".                                          |
| <asp:ListBox>      | Một danh sách trong tập hợp đối tượng <asp:ListItem>. Trong HTML, nó được trả về bởi thẻ <select> với thuộc tính size="x" với x là số lượng các mục chọn được hiển thị. |

| Thẻ                   | Mô tả                                                                                                                                                              |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <asp:CheckBoxList>    | Các mục chọn của nó được trả về như là các check boxes, cùng với một bảng với một hoặc nhiều cột.                                                                  |
| <asp:RadioButtonList> | Giống như <asp:CheckBoxList> nhưng các mục chọn được trả về là các radio buttons.                                                                                  |
| <asp:BulletedList>    | Một danh sách liệt kê dạng số hoặc dạng bullets. Trong HTML nó được trả về sử dụng thẻ <ul> or <ol> tags. Người ta thường dùng nó để tạo danh sách các hyperlinks. |

#### 4.2. Các thuộc tính của lớp ListControl

Lớp ListControl được thừa kế từ lớp **System.Web.UI.WebControls.ListControl**. Lớp này hiển thị tất cả các thuộc tính và phương thức giống như các web controls khác. Ta có thể đưa dữ liệu vào các list controls tự động hoặc bằng cách viết code.

| Thuộc tính                               | Mô tả                                                                                                                                                                                                                                                   |
|------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DataMember</b>                        | Được dùng cùng với data binding khi nguồn dữ liệu chứa nhiều hơn một table ( <i>khi nguồn dữ liệu là DataSet</i> ). <b>DataMember</b> chỉ định Table nào mà ta muốn sử dụng.                                                                            |
| <b>DataTextField</b>                     | Được dùng cùng với data binding để chỉ định thuộc tính hoặc trường nào trong data source nên được dùng đối với mục chọn hiển thị của list item.                                                                                                         |
| <b>DataValueField</b>                    | Được dùng cùng với data binding để chỉ định thuộc tính hoặc trường nào được dùng trong nguồn dữ liệu (data source) nên được dùng cho thuộc tính của mỗi mục chọn trong danh sách ( <i>Không được hiển thị nhưng có thể được đọc bởi chương trình</i> ). |
| <b>DataTextFormatString</b>              | Gán chuỗi được định dạng hoặc trả về nội dung văn bản của mục chọn ( <i>phải phù hợp với thuộc DataTextField</i> ).                                                                                                                                     |
| <b>RepeatLayout:</b>                     | Chỉ định checkboxes hoặc radio buttons nào được trả về trong table ( <i>tùy chọn mặc định</i> ) hoặc inline. Các giá trị là Table và Flow.                                                                                                              |
| <b>RepeatDirection:</b>                  | Chỉ định hướng của danh sách controls được trả về theo hướng <b>horizontally</b> hoặc <b>vertically</b> .                                                                                                                                               |
| <b>RepeatColumns</b>                     | Đặt số cột trong trường hợp RepeatLayout được đặt là Table.                                                                                                                                                                                             |
| <b>CellPadding</b><br><b>CellSpacing</b> | Nếu RepeatLayout=Table, sau đó những thuộc tính này cấu hình để chỉ định khoảng cách và canh lề văn bản của các ô trong table.                                                                                                                          |
| <b> TextAlign</b>                        |                                                                                                                                                                                                                                                         |

#### 4.3. Ví dụ:

Nội dung của tập tin SelectableListControl.aspx như sau

```

<form id="form1" runat="server">
 <asp:ListBox runat="server" ID="Listbox1" SelectionMode="Multiple" Rows="5">
 <asp:ListItem Selected="true">Mục chọn 1</asp:ListItem>
 <asp:ListItem>Mục chọn 2 </asp:ListItem>
 </asp:ListBox>

 <asp:DropDownList runat="server" ID="DropdownList1">

```

```

<asp:ListItem Selected="true">Mục chọn 1</asp:ListItem>
<asp:ListItem>Mục chọn 2</asp:ListItem>
</asp:DropDownList>

<asp:CheckBoxList runat="server" ID="CheckboxList1" RepeatColumns="3" >
 <asp:ListItem Selected="true">Mục chọn 1</asp:ListItem>
 <asp:ListItem>Mục chọn 2</asp:ListItem>
</asp:CheckBoxList>

<asp:RadioButtonList runat="server" ID="RadiobuttonList1"
RepeatDirection="Horizontal" RepeatColumns="3">
 <asp:ListItem Selected="true">Mục chọn 1</asp:ListItem>
 <asp:ListItem>Mục chọn 2</asp:ListItem>
</asp:RadioButtonList>

<asp:Button ID="Button1" runat="server" Text="Submit" OnClick="Button1_Click"/>
</form>

```

Nội dung của tập tin SelectableListControl.aspx.

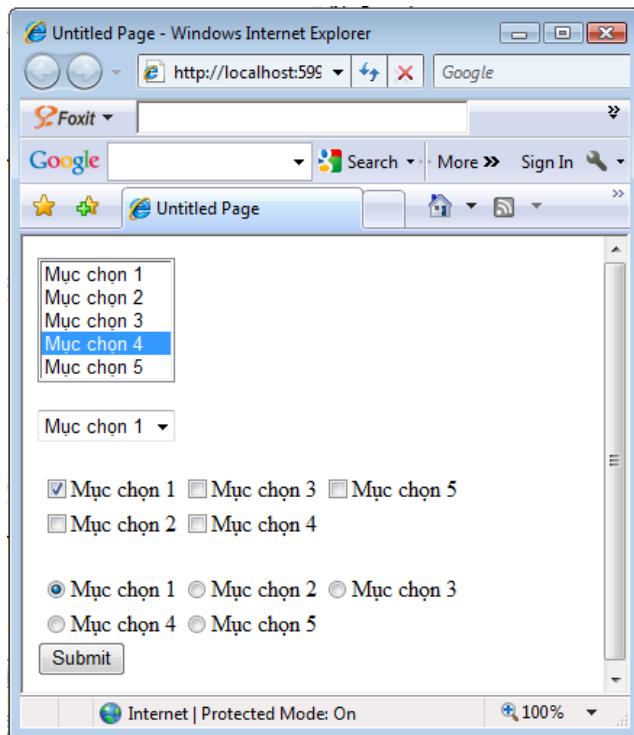
Hàm PageLoad ở code-behind để bổ sung thêm các mục chọn từ 3 đến 5 đối với mỗi thành phần của List box ở trên.

```

protected void Page_Load(object sender, EventArgs e)
{
 if (!Page.IsPostBack)
 {
 for (int i = 3; i <= 5; i++)
 {
 Listbox1.Items.Add("Mục chọn " + i.ToString());
 DropDownList1.Items.Add("Mục chọn " + i.ToString());
 CheckboxList1.Items.Add("Mục chọn " + i.ToString());
 RadiobuttonList1.Items.Add("Mục chọn " + i.ToString());
 }
 }
}

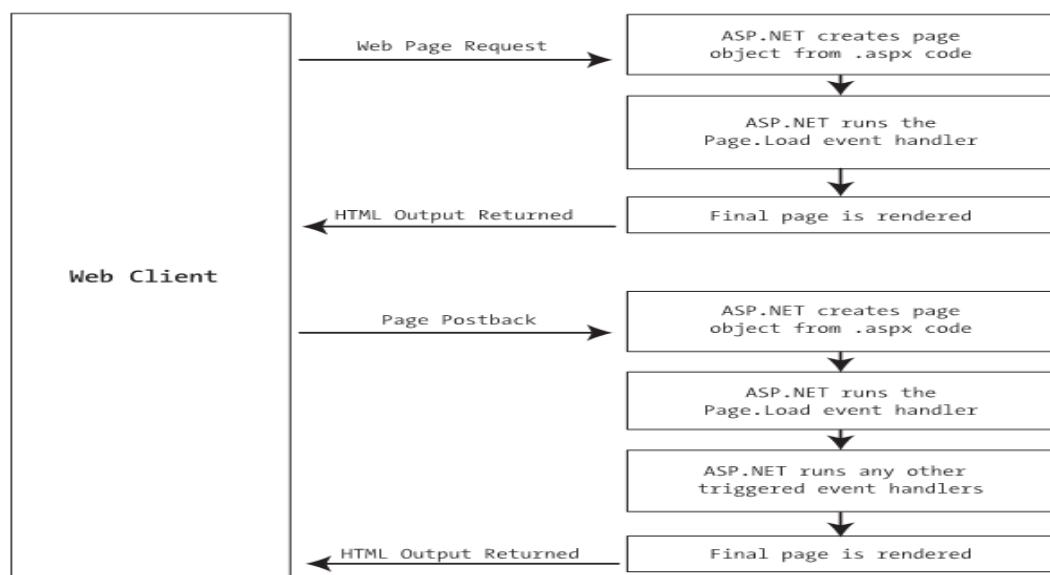
```

Kết quả của đoạn mã lệnh được cho như sau:

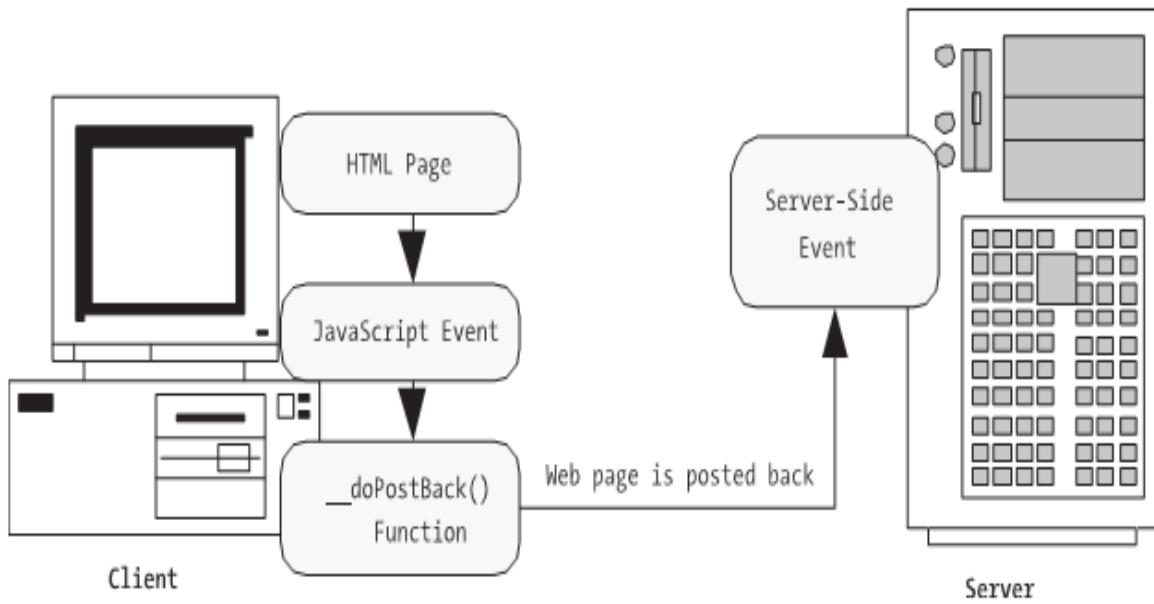


**Hình 14: Form Kết quả với List Controls**

## V. Mô hình sự kiện Web Control



| Sự kiện              | Web controls được áp dụng                                              | Posts Back |
|----------------------|------------------------------------------------------------------------|------------|
| Click                | Button, ImageButton                                                    | True       |
| TextChanged          | TextBoxx (kích hoạt chỉ sau khi người dùng di chuyển đến control khác) | False      |
| CheckedChanged       | CheckBox, RadioButton                                                  | False      |
| SelectedIndexChanged | DropDownList, ListBox, CheckBoxList, RadioButtonList                   | False      |



**Hình 15: Mô hình sự kiện Auto Postback**

Khi người dùng click chọn nút lệnh button1 ở ví dụ 5.3 có nghĩa là người dùng đã kích hoạt một sự kiện click trên nút lệnh. Đoạn code sau đây được viết ở code-behind sẽ cho biết các mục chọn mà người dùng chọn từ trang web ở ví dụ 5.3.

```

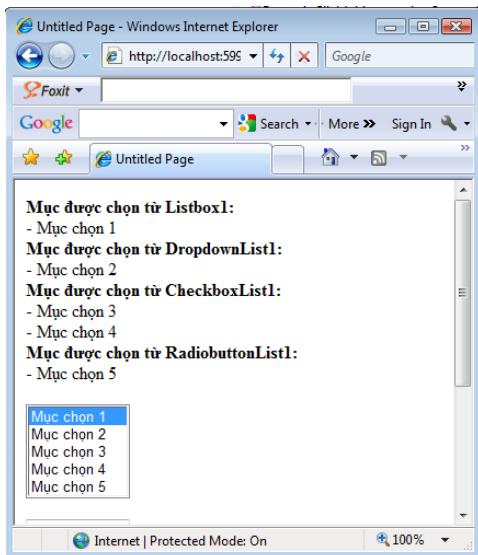
protected void Button1_Click(object sender, System.EventArgs e)
{
 Response.Write("Mục được chọn từ Listbox1:
");
 foreach (ListItem li in Listbox1.Items)
 {
 if (li.Selected)
 Response.Write("- " + li.Text + "
");
 }

 Response.Write("Mục được chọn từ DropDownList1:
");
 Response.Write("- " + DropDownList1.SelectedItem.Text + "
");

 Response.Write(" Mục được chọn từ CheckboxList1:
");
 foreach (ListItem li in CheckboxList1.Items)
 {
 if (li.Selected) Response.Write("- " + li.Text + "
");
 }

 Response.Write("Mục được chọn từ RadioButtonList1:
");
 Response.Write("- " + RadioButtonList1.SelectedItem.Text + "
");
}

```



*Hình 16: Form Kết quả với sự kiện trong asp.net*

### Câu hỏi củng cố:

1. Trình bày các thuộc tính cơ bản của lớp cơ sở WebControl trong ASP.net.
2. Nêu các thuộc tính cơ bản của lớp List Controls
3. Hãy nêu ý nghĩa của thuộc tính AutoPostBack
4. Nêu các dạng sự kiện và các controls tương ứng trong các Web Controls
5. Thực hành bài tập Lab3.2 ở trang 47 bằng cách viết code
6. Thực hành bài tập Lab3.2 ở trang 47 bằng cách sử dụng công cụ Wizard.

# Bài 3: ĐIỀU KHIỂN KIỂM TRA NHẬP LIỆU

## Mục tiêu học tập:

- Trình bày các loại Điều khiển kiểm tra nhập liệu trong ASP.net
- Sử dụng được lớp cơ sở Validator để tạo các ràng buộc về nhập liệu cho thành phần trên form.
- Áp dụng các quy tắc biểu diễn biểu thức nhập liệu để tạo các ràng buộc theo yêu cầu.
- Cài đặt các kiểm tra nhập liệu ở trình chủ hoặc trình khách

## I. Điều khiển kiểm tra nhập liệu:

Một trong những việc dùng phổ biến của trang web là tập hợp dữ liệu. Thông thường, trang web sẽ yêu cầu người dùng nhập vào một số thông tin để lưu trữ vào cơ sở dữ liệu. Trong các trường hợp như vậy, dữ liệu cần phải được kiểm tra để đảm bảo rằng bạn không lưu trữ thông tin vô ích, sai hoặc không thỏa điều kiện người dùng đặt ra.

Kiểm tra dữ liệu nhập vào của người dùng nên thực hiện ở phía trình khách để đảm bảo người dùng không cung cấp các thông tin sai. Tuy nhiên bắt buộc kiểm tra ở phía trình khách được thực hiện, dữ liệu trên form phải được kiểm tra tại server. Nếu không, trang web có thể bị tấn công thông qua các đoạn code phía trình khách.

Viết validation code bằng tay là một việc rất mất thời gian, đặc biệt đối với các mô hình lập trình phía trình khách (*cụ thể là các JavaScript*). Lập trình phía server-side (*trong trường hợp này ASP.NET*) không giống như vậy. Các nhà phát triển ở Microsoft nhận biết rõ về điều này, họ bổ sung thêm các điều khiển kiểm tra nhập liệu (validation controls). Những controls này có thể khai báo trên các web form và gắn với các input control khác. Một khi được gắn vào các thành phần nhập liệu, các validation controls thực hiện kiểm tra tự động ở *client-side* và *server-side validation*. Nếu các control tương ứng là rỗng, hoặc không theo một quy luật cụ thể, việc kiểm tra sẽ ngăn ngừa tình trạng được posted back trở lại server nhiều lần.

## II. Các loại điều khiển kiểm tra nhập liệu

ASP.net bao gồm 6 điều khiển kiểm tra nhập liệu. Các controls này thực hiện một phần của việc kiểm tra nhập liệu và giúp bạn tiết kiệm những công việc phải kiểm tra việc nhập liệu nhiều lần.

| Lớp                              | Diễn giải                                                                                                                             |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <asp:RequiredFieldValidator>     | Kiểm tra control tương ứng phải phù hợp khi nhấn nút lệnh submit trên form                                                            |
| <asp:RangeValidator>             | Kiểm tra giá trị được gắn với controls liên kết được chỉ định ở một mức nào đó. Giá trị có thể là một khoảng số, ngày hoặc chuỗi.     |
| <asp:CompareValidator>           | Kiểm tra giá trị của control liên kết thỏa mãn sự so sánh cụ thể (less than, greater than,...) với giá trị hằng hoặc điều khiển khác. |
| <asp:RegularExpressionValidator> | Kiểm tra giá trị của control phải phù hợp với regular expression nào đó.                                                              |
| <asp:CustomValidator>            | Cho phép ta chỉ định cụ thể kiểm tra phía trình khách client-side JavaScript.                                                         |
| <asp:ValidationSummary>          | Thể hiện tóm tắt các lỗi đối với mỗi kiểm tra trên trang (hoặc một pop-up message box).                                               |

**Lưu ý:** Ta có thể sử dụng nhiều kiểm tra nhập liệu trên cùng một điều khiển webcontrol. Ví dụ, ta có thể dùng một validator để kiểm tra một ô nhập liệu nào đó là rỗng hay không để đảm bảo chắc chắn là nó có dữ liệu trước khi kiểm tra một ràng buộc nào khác. Nếu ta sử dụng RangeValidator, CompareValidator hoặc RegularExpressionValidator, sẽ không thể phát hiện lỗi trừ khi ta có sử dụng kiểm tra RequiredFieldValidator.

Ta không thể kiểm tra nhập liệu vào RadioButton hoặc Checkbox controls. Các controls có thể sử dụng để validate là Textbox, HtmlSelect, ListBox, DropDownList, RadioButtonList. Khi sử dụng kiểm tra nhập liệu trên các list controls, thuộc tính Value của đối tượng ListItem được chọn.

### III. Lớp cơ sở Validator

Bảng sau đây liệt kê một số thuộc tính quan trọng khi thực hiện thiết kế các ràng buộc về nhập liệu trên form.

| Thuộc tính                | Mô tả                                                                                                                                                                                                                                       |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ControlToValidate</b>  | Chỉ định tên input control để kiểm tra.                                                                                                                                                                                                     |
| <b>Display</b>            | Chỉ định lỗi phát sinh như thế nào. Nếu <b>Display= Static</b> , khoảng trắng cần để hiển thị lỗi được tính toán và được thêm vào khoảng cách đầu tiên. Nếu <b>Display=Dynamic</b> , ASP.net sẽ tự động thay đổi để hiện thị thông báo lỗi. |
| <b>EnableClientScript</b> | Cho biết <b>các kiểm tra ở phía trình khách</b> có xảy ra hay không. Mặc định là true.                                                                                                                                                      |
| <b>Enabled:</b>           | Cho phép người dùng có thể enable hoặc disable validator. Khi control là disabled, nó sẽ không kiểm tra gì cả.                                                                                                                              |
| <b>ErrorMessage:</b>      | Nếu có lỗi xảy ra. Chuỗi thông báo lỗi sẽ được thể hiện bởi Validation-Summary control                                                                                                                                                      |

#### Một số ví dụ

```
<asp:TextBox runat="server" ID="Name" />
<asp:RequiredFieldValidator ID="rfvName" runat="server"
 ControlToValidate="Name"
 ErrorMessage="Bạn cần nhập tên"
 Display="dynamic">
</asp:RequiredFieldValidator>
```

**Ý nghĩa:** Kiểm tra xem người dùng đã nhập tên vào textbox Name hay chưa? Nếu chưa sẽ xuất hiện thông báo lỗi: "Bạn cần nhập tên"

```
<asp:TextBox runat="server" ID="DayOff"/>
<asp:RangeValidator runat="server" Display="dynamic"
 ID="rvDayOff"
 ControlToValidate="DayOff" Type="Date"
 ErrorMessage="Ngày nghỉ phải trong khoảng"
 MinimumValue="08/05/2008"
 MaximumValue="08/20/2008">
</asp:RangeValidator>
```

**Ý nghĩa:** Kiểm tra xem người dùng đã nhập ngày nghỉ vào textbox tên Dayoff có nằm trong khoảng từ 5/8/2008 đến ngày 20/8/2008 hay không?

```
<asp:TextBox runat="server" ID="Age" >
<asp:CompareValidator ID="cvAge" runat="server"
 Display="dynamic" ControlToValidate="Age"
```

**Ý nghĩa:** Kiểm tra xem người dùng đã nhập tuổi vào textbox tên Age thỏa điều kiện lớn hơn hoặc bằng

```

 ValueToCompare="18" Type="Integer"
 Operator="GreaterThanOrEqual"
 ErrorMessage="Bạn phải lớn hơn 18 tuổi">
 </asp:CompareValidator>

```

"GreaterThanOrEqual" hay không?. Nếu không thỏa thì hiển thị thông báo lỗi "Bạn phải lớn hơn 18 tuổi".

```

<asp:TextBox runat="server" TextMode="Password"
ID="Password" />
<asp:TextBox runat="server" TextMode="Password"
ID="Password2" BorderStyle="Solid" BorderColor="Blue" />
<asp:CompareValidator ID="CompareValidator2"
runat="server"
 ControlToValidate="Password2"
 ControlToCompare="Password"
 ErrorMessage="Password không khớp!"
 Type="String" Display="dynamic">
</asp:CompareValidator>

```

**Ý nghĩa:** Kiểm tra xem người dùng đã nhập mật khẩu vào Password2 có khớp với mật khẩu đã nhập vào textbox Password hay không?. Nếu không thỏa thì hiển thị thông báo lỗi "Password không khớp!".

```

<asp:RegularExpressionValidator
ID="RegularExpressionValidator1" runat="server"
 ControlToValidate="Email"
 ValidationExpression=".*@.{2,}\.{2,}"
 ErrorMessage="E-mail không hợp lệ"
 Display="dynamic">
</asp:RegularExpressionValidator>

```

**Ý nghĩa:** Kiểm tra xem người dùng đã nhập địa chỉ email vào textbox Email có đúng định dạng .\*@.{2,}\.{2,} hay không?. Nếu không thỏa thì hiển thị thông báo lỗi "Email không hợp lệ!".

```

<asp:ValidationSummary ID="errSum" runat="server"
DisplayMode="List" />

```

**Ý nghĩa:** Tóm tắt các thông báo lỗi từ các ô nhập liệu trên trang và hiển thị dưới dạng danh sách

Kết quả của đoạn mã kiểm tra một số thông tin được nhập vào được hiển thị như màn hình bên dưới

| Điều kiện                                           | Giá trị nhập | Thông báo lỗi               |
|-----------------------------------------------------|--------------|-----------------------------|
| Nhập họ và tên:                                     |              | Bạn cần nhập tên            |
| Nhập số ngày nghỉ<br>(Từ 08/05/2008 đến 20/08/2008) | 45           | Ngày nghỉ phải trong khoảng |
| Nhập tuổi (>18):                                    | 15           | Bạn phải lớn hơn 18 tuổi    |
| Nhập mật khẩu:                                      | ***          |                             |
| Nhập lại mật khẩu                                   | ****         | Password không khớp!        |
| Nhập địa chỉ Email:                                 | nguyen       | E-mail không hợp lệ         |

Tóm tắt thông báo lỗi

- Bạn cần nhập tên
- Ngày nghỉ phải trong khoảng
- Bạn phải lớn hơn 18 tuổi
- Password không khớp!

**Hình 17: Form Kết quả với Điều khiển Kiểm tra Nhập liệu**

#### IV. Regular Expression Validator

Bảng sau đây mô tả một số ký tự đại diện được dùng trong mẫu nhập liệu có thể sử dụng trong thiết kế ràng buộc cho địa chỉ email hoặc mật khẩu đăng nhập.

| Ký tự | Mô tả                                                                                                  |
|-------|--------------------------------------------------------------------------------------------------------|
| [^ ]  | Matches 1 ký tự không nằm trong dãy cho trước. Ví dụ [^A-B] matches bất kỳ ký tự nào ngoại trừ A và B. |
| .     | Bất kỳ ký tự nào ngoại trừ newline. Ví dụ, .here matches where và there.                               |
| \s    | Bất kỳ ký tự khoảng trắng nào (như tab hoặc space).                                                    |
| \S    | Bất kỳ ký tự không phải khoảng trắng nào                                                               |
| \d    | Bất kỳ ký tự số nào.                                                                                   |
| \D    | Bất kỳ ký tự nào không là số                                                                           |
| \w    | Bất kỳ “word” (ký tự, số, hoặc dấu gạch dưới)                                                          |
| \W    | Bất kỳ ký tự nào không là “word” (ký tự, số, hoặc gạch dưới).                                          |
| *     | 0 hoặc nhiều hơn 1 ký tự đứng trước Ví dụ 7*8 => 7778 hoặc 8.                                          |
| +     | 1 hoặc nhiều hơn 1 ký tự đứng trước. Ví dụ 7+8 => 7778, not 8.                                         |
| ( )   | Nhóm ký tự được xem như 1 thành phần. Ví dụ (78)+ => 78 và 787878.                                     |
| {m,n} | Ký tự đứng trước có thể xuất hiện từ m đến n lần. Ví dụ, A{1,3} => A, AA, or AAA.                      |
|       | Ví dụ: 8 6 => 8 hoặc 6.                                                                                |
| [ ]   | Matches 1 ký tự trong một dãy ký tự phù hợp. Ví dụ [A-C] matches A, B, hoặc C.                         |

Ví dụ một số mẫu kiểm tra nhập liệu thường sử dụng

| Nội dung kiểm tra               | Mẫu kiểm tra      | Mô tả                                                                                                                   |
|---------------------------------|-------------------|-------------------------------------------------------------------------------------------------------------------------|
| Địa chỉ email                   | \S+@\S+\.\S+      | Định nghĩa một địa chỉ email cần có ký tự @ và ký tự dấu chấm(.)                                                        |
| Mật khẩu                        | \w+               | Mật khẩu bắt buộc phải là một chuỗi các từ (ký tự, khoảng cách, hoặc dấu gạch dưới)                                     |
| Mật khẩu với chiều dài xác định | \w{4, 10}         | Mật khẩu phải có ít nhất từ 4 đến 10 ký tự                                                                              |
| Mật khẩu phức tạp hơn           | [a-zA-Z]\w*\d+\w* | Mật khẩu phải bắt đầu bằng 1 ký tự, theo sau là 0 hoặc nhiều ký tự. Một hoặc nhiều chữ số, sau đó là 0 hoặc nhiều ký tự |
| Kiểm tra mật khẩu               | \S{4,10}          | Mật khẩu phải có 4 đến 10 ký tự nhưng có thể có các ký tự đặt biệt như (*. &,...)                                       |

## V. CustomValidator Control

CustomValidator cho phép ta kiểm tra dữ liệu cả hai phía client side và server side. Ta có thể gắn kết những với điều khiển để mà validation được thực thi một cách tự động. Nếu validation fails, thuộc tính Page.IsValid được đặt là false tương tự như các điều khiển kiểm tra nhập liệu khác.

Cả hai kiểm tra nhập liệu ở client và server được thực hiện tương tự nhau. Một phương thức với 2 tham số  $\langle Tên\ hàm \rangle$  ( $\langle Tên\ control \rangle$ ,  $\langle Đối\ tượng\ kiểm\ tra \rangle$ ). Đối tượng kiểm tra cung cấp thuộc tính Value chứa giá trị được gắn kết với điều khiển nhập liệu. Ví dụ, để kiểm tra số nhập vào textbox có thể chia hết cho 5 hay không

```
<script type="text/javascript">
 function EmpIDClientValidate(ctl, args)
 {
 args.IsValid=(args.Value%5 == 0);
 }
</script>

<asp:TextBox runat="server" ID="EmpID" />
<asp:CustomValidator runat="server" ControlToValidate="EmpID"
 ClientValidationFunction="EmpIDClientValidate"
 ErrorMessage="ID must be a multiple of 5" Display="dynamic">*
</asp:CustomValidator>
```

Khi thông tin của form được chuyển về server, ASP.net sẽ tự động gọi sự kiện CustomValidator.ServerValidate. Ta có thể thực hiện các kiểm tra tương tự bằng mã lệnh C#. Đoạn mã lệnh sau đây dùng để kiểm tra nhập liệu ở trình chủ.

```
protected void EmpIDServerValidate(object sender, ServerValidateEventArgs args)
{
 try
 {
 args.IsValid = (int.Parse(args.Value)%5 == 0); }
 catch
 {
 args.IsValid = false; // Khi có một lỗi xảy ra }
}
```

### Câu hỏi/bài tập củng cố:

1. Hãy nêu tên các điều khiển kiểm tra nhập liệu trong ASP.net
2. Cho biết các thuộc tính của lớp cơ sở Validator.
3. Liệt kê các ký tự đặc biệt được dùng để tạo mẫu kiểm tra nhập liệu.
4. Cho biết sự khác nhau giữa loại điều khiển nhập liệu CustomValidator và các loại điều khiển nhập liệu khác.
5. Thực hành bài tập Lab 3.3 ở trang 50 bằng cách viết code
6. Thực hành bài tập Lab 3.3 ở trang 50 bằng công cụ Wizard.

## Bài 4: Rich Controls

### Mục tiêu học tập:

- Tạo file chứa nội dung quảng cáo xml
- Liên kết tập tin quảng cáo với điều khiển AdRotator
- Tạo được một quyển lịch với điều khiển Calendar.
- Cài đặt sự kiện và định dạng quyển lịch Calendar.

### I. Rich controls là gì?

Rich controls là các webcontrols có thể tạo các thành phần giao diện người dùng khá phức tạp. Mặc dù không có một định nghĩa cụ thể nào đối với thuật ngữ rich controls, chúng ta có thể hiểu rich controls là các web controls cung cấp một mô hình đối tượng khác biệt với các trình bày HTML. Một rich control có thể được khai báo dưới dạng các thẻ, và sau đó sẽ trả về một loạt các thẻ HTML và cũng có thể có các mã lệnh JavaScript.

ASP.NET bao gồm rất nhiều các rich controls khác nhau được trình bày ở các phần khác nhau trong tài liệu này. Cụ thể là các điều khiển dựa trên dữ liệu, các điều khiển chuyển hướng dữ liệu. Một số rich controls tiêu biểu được trình bày theo sau.

- **AdRotator:** Điều khiển này là một bảng quảng cáo để hiển thị một trong tập các ảnh được đặt định kỳ trong file XML
- **Calendar:** Control này cho phép ta tạo một lịch biểu có thể hiển thị tháng và ngày. Ta cũng có thể chọn ngày nào đó.
- **MultiView, View, Wizard:** Ta có thể nghĩ các controls này như là các bảng cho phép ta chuyển đổi các nhóm controls trong một trang. Control Wizard có thể bao gồm chức năng chuyển hướng dữ liệu.
- **Xml:** controls này đọc tập tin XML và XSLT, sau đó hiển thị dữ liệu dưới dạng HTML ở trình duyệt.

Trong nội dung bài học này, chúng ta chỉ đi vào 2 nội dung chính là AdRotator và Calendar. Các điều khiển chuyển hướng dữ liệu như MultiView, View, Wizard sẽ được trình bày trong một chương khác. Tương tự với XML sẽ được trình bày trong phần ADO.net

### II. Điều khiển tạo tập ảnh quảng cáo AdRotator

AdRotator chọn ngẫu nhiên một hình ảnh nào đó để làm tiêu đề cho quảng cáo trong danh sách các hình ảnh chỉ ra trong tập tin XML. Trước khi tạo điều khiển này, ta cần tạo tập tin XML.

#### Ví dụ:

- Tạo tập tin **Ads.xml** với nội dung như sau:

**<Advertisements>**

```
<Ad>
 <ImageUrl>BannerTravel1.jpg</ImageUrl>
 <NavigateUrl> http://www.dulichvietnam.vn </NavigateUrl>
 <AlternateText> Web site du lịch Việt Nam </AlternateText>
 <Impressions> 20 </Impressions>
 <Keyword> books </Keyword>
```

```

</Ad>
<Ad>
 <ImageUrl> BannerTravel2.jpg </ImageUrl>
 <NavigateUrl> http://www.dulichvietnam.vn </NavigateUrl>
 <AlternateText> Web site du lịch Việt Nam </AlternateText>
 <Impressions> 20 </Impressions>
 <Keyword> Java </Keyword>
</Ad>
</Advertisements>

```

*Bảng sau đây mô tả các thẻ của tập tin Advertisement*

Thành phần	Mô tả
ImageUrl	Ảnh được hiển thị. Có thể là một liên kết tương đối ( <i>một tập tin trong thực mục hiện tại</i> ) hoặc một địa chỉ URL.
NavigateUrl	Liên kết được theo sau nếu người dùng click vào banner
AlternateText	Đoạn text sẽ được hiện thị nếu hình ảnh không được hiển thị. Đoạn text này cũng được dùng như tooltip trong một vài browser
Impressions	Một số cho biết quảng cáo đặt bao nhiêu lần. Ví dụ: một banner với giá trị 10 sẽ được xuất hiện 2 lần so với banner có giá trị 5
Keyword	Một từ khóa xác định một nhóm các quảng cáo. Có thể sử dụng để lọc dữ liệu.

- Liên kết điều khiển AdRotator với tập tin Ads.xml như sau:  
`<asp:AdRotator runat="server" AdvertisementFile="Ads.xml" Target="_blank" />`

### III. Điều khiển tạo lịch Calendar

Điều khiển Calendar tạo một quyển lịch khá đẹp với nhiều tính năng thêm vào. Một số tính năng nổi bật của điều khiển này như sau:

- Nó có thể hiển thị các ngày trong tháng.
- Người dùng có thể thay đổi tháng xem, chọn ngày hoặc một số ngày nào đó.
- Người dùng có thể định dạng màu nền, font chữ, tiêu đề, định dạng ngày.
- Lập trình viên có thể cài đặt sự kiện SelectionChanged.
- Sử dụng sự kiện DayRender để thay đổi định dạng của ô trong quyển lịch.
- Có thể làm nổi bật các ngày quan trọng trong quyển lịch.

#### Tạo mới một Calendar

```

<asp:Calendar runat="server" ID="Calendar1"
 ForeColor="red" BackColor="lightyellow" />

```

- Màu nền và màu chữ thay đổi sang Green và Yellow khi ngày là ngày cuối tuần

#### Cài đặt sự kiện DayRender

```
protected void Calendar1_DayRender(object sender,
```

```
 DayRenderEventArgs e)
```

```
{
```

```
 if (e.Day.IsWeekend)
```

```

 {
 e.Cell.BackColor = System.Drawing.Color.Green;
 e.Cell.ForeColor = System.Drawing.Color.Yellow;
 e.Day.IsSelectable = false;
 }
}

```

### Một số thuộc tính định dạng của Calendar

Thuộc tính	Ý nghĩa
DayHeaderStyle	Định dạng khu vực Calendar hiển thị ngày trong tuần
DayStyle	Định dạng mặc định ngày trong tháng hiện hành
NextPrevStyle	Định dạng navigation controls trong phần tiêu đề để di chuyển từ tháng này sang tháng khác.
OtherMonthDayStyle	Kiểu ngày không được hiển thị trong tháng hiện tại. Các ngày này được dùng để fill calendar grid.
SelectedDayStyle	Định dạng ngày được chọn trong calendar.
SelectorStyle	Định dạng controls cho tuần hoặc tháng được chọn.
TitleStyle	Định dạng khu vực tiêu đề.
TodayDayStyle	Định dạng ngày được gán là ngày hôm nay.
WeekendDayStyle	Định dạng ngày cuối tuần

### Câu hỏi/Bài tập củng cố:

- Tạo được một quyền lịch với điều khiển Calendar như hình bên.
- Yêu cầu:
  - Khi mở trang web hiển thị ngày giờ hiện tại
  - Tạo ghi nhớ cho các ngày 1/1, 1/5, 1/6 và 20/11
  - Khi ngày hiện tại giống như ngày đã ghi nhớ thì có thông báo nhắc nhở.



# Bài tập Chương 3

## Lab 3.1: Sử dụng ASP.net Web Control cơ bản

### Mục tiêu:

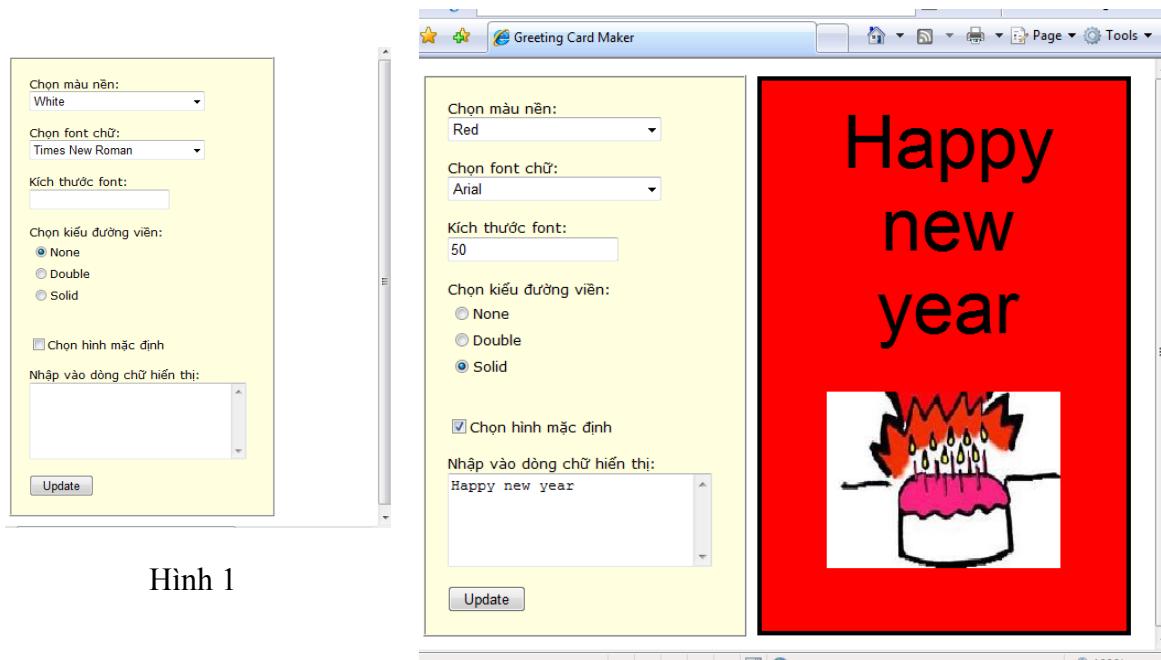
- Sinh viên có thể sử dụng web controls trong ASP.net
- Sinh viên có thể sử dụng các thuộc tính của các đối tượng trong Web controls
- Cài đặt được các sự kiện đơn giản trong ASP.net

### Kiến thức cần thiết:

- Sinh viên đã học WebControls.
- Sinh viên có kiến thức về mô hình quản lý sự kiện trong ASP.net

### Nội dung:

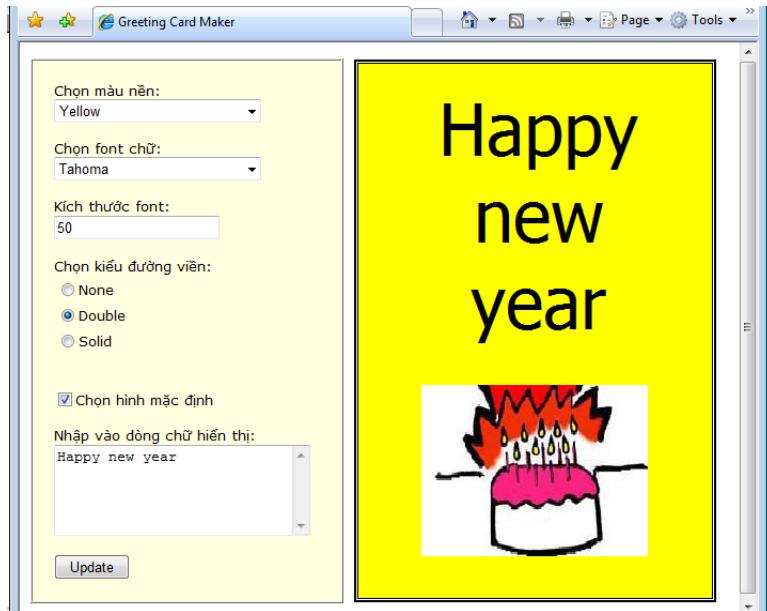
1. Sinh viên thiết kế trang web có nội dung như **hình 1**:  
Lần đầu tiên khi trang web được gọi sẽ hiển thị các thông tin cần thiết cho người dùng chọn lựa các thông tin cần thiết.
2. Khi người dùng xác định các thông tin, trang web sẽ hiển thị như **hình 2** bên dưới:



Hình 1

Hình 2

3. Người dùng thay đổi chọn lựa thì trang web sẽ hiển thị như hình 3 phía dưới



Hướng dẫn:

Hình 3

Bước 1: Tạo trang GreetingCards.aspx có các thành phần như sau:

Tên	Kiểu	Thuộc tính	Diễn giải
IstMaunen	dropdownlist	runat="server"	Danh sách màu nền
IstTenfont	dropdownlist	runat="server"	Danh sách tên font
txtFontSize	textbox		Kích thước font
IstBorder	radiobuttonlist		Chọn màu nền
chkPicture	checkbox		
txtGreeting	textbox	TextMode="MultiLine" runat="server"	Dòng chữ hiển thị
cmdUpdate	button	Text="Update" onclick="cmdUpdate_Click"	Nút lệnh Update
lblGreeting	Label		
imgDefault	Image	Visible="False"	
pnlCard	panel	Height="507px" Width="339px"	Khung hình

Bước 2: Phần xử lý CodeBehind của GreetingCards.aspx

- Code xử lý trong hàm PageLoad

```
protected void Page_Load(object sender, System.EventArgs e)
{
 if (this.IsPostBack == false)
 {
 // Thêm các mục chọn.
 IstMaunen.Items.Add("White");
 IstMaunen.Items.Add("Red");
 IstMaunen.Items.Add("Green");
 IstMaunen.Items.Add("Blue");
 }
}
```

```

IstMaunen.Items.Add("Yellow");

// Thêm mục chọn tên font
IstTenfont.Items.Add("Times New Roman");
IstTenfont.Items.Add("Arial");
IstTenfont.Items.Add("Verdana");
IstTenfont.Items.Add("Tahoma");

// Thêm mục chọn kiểu đường viền
ListItem item = new ListItem();
// Mục chọn có tên kiểu đường là None.
item.Text = BorderStyle.None.ToString();
item.Value = ((int)BorderStyle.None).ToString();

// Thêm mục chọn
IstBorder.Items.Add(item);
item = new ListItem();
item.Text = BorderStyle.Double.ToString();
item.Value = ((int)BorderStyle.Double).ToString();
IstBorder.Items.Add(item);

item = new ListItem();
item.Text = BorderStyle.Solid.ToString();
item.Value = ((int)BorderStyle.Solid).ToString();
IstBorder.Items.Add(item);

// Mặc định không có đường viền cho khung hình.
IstBorder.SelectedIndex = 0;
// Đặt hình nền.
imgDefault.ImageUrl = "defaultpic.png";
}

}

```

- Code xử lý cho nút lệnh Command
 

```

protected void cmdUpdate_Click(object sender, System.EventArgs e)
{
 // Cập nhật màu nền.
pnlCard.BackColor = Color.FromName(IstMaunen.SelectedItem.Text);

 // Cập nhật font chữ
lblGreeting.Font.Name = IstFontName.SelectedItem.Text;

 try
 {
 if (Int32.Parse(txtFontSize.Text) > 0)

```

```

 {
 lblGreeting.Font.Size =
 FontUnit.Point(Int32.Parse(txtFontSize.Text));
 }
}

// Cập nhật kiểu đường viền
pnlCard.BorderStyle =
 (BorderStyle)Int32.Parse(lstBorder.SelectedItem.Value);
// Cập nhật hình.
if (chkPicture.Checked == true)
{
 imgDefault.Visible = true;
}
else
{
 imgDefault.Visible = false;
}

//Nội dung câu chào
lblGreeting.Text = txtGreeting.Text;
}

```

-----Hết---

## Lab 3.2: Sử dụng các Web Control cơ bản

### 1. Mục tiêu

- Sử dụng các thành phần web server control cơ bản
- Sử dụng lớp StreamReader và StreamWriter để đọc ghi file dạng text
- Các thao tác xử lý trên chuỗi, chia chuỗi, thay thế chuỗi.

### 2. Yêu cầu

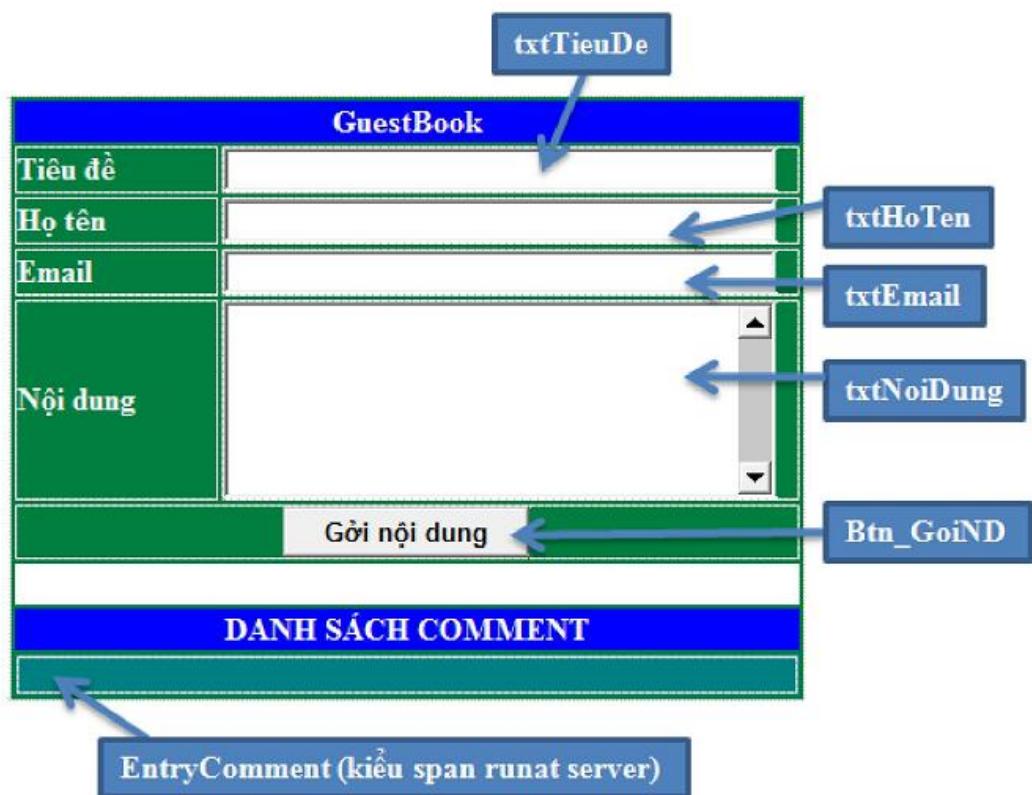
- Đã làm quen với các server control cơ bản,
- Đã học qua các lớp xử lý nhật xuất file của C#.

### 3. Nội dung

- Các trang web thường có phần GuestBook, cho phép user có thể lưu lại các comment cho chủ nhân của trang web. Trong bài này chúng ta sẽ tạo một form minh họa dạng GuestBook, cho user nhập vào các thông tin:
  - Tiêu đề
  - Họ tên
  - Địa chỉ email
  - Nội dung comment
- Các thông tin này sẽ được lưu lại trên server (qua file text)
- Những entry của user trước đó post lên cũng được hiển thị trong trang này.
- Hình 1 minh họa form **GuestBook**

### 4. Hướng dẫn:

#### a. Thiết kế giao diện:



**b. Chi tiết của các control như sau:**

Tên control	Kiểu	Thuộc tính	Điễn giải
txtTieuDe	TextBox	TextMode = SingleLine	Nhập tiêu đề comment
txtHoTen	TextBox	TextMode = SingleLine	Nhập họ tên
txtEmail	TextBox	TextMode = SingleLine	Địa chỉ email
txtNoiDung	TextBox	TextMode = MultiLine Rows = 6	Nhập nội dung comment
Btn_GoiND	Button		Thực thi hành động post comment mới lên server. Trình xử lý sự kiện này sẽ thêm một entry mới vào file.
EntryComment	Span	Runat = server	Chứa toàn bộ các entry comment được lấy từ trong file trên server

**c. Phần xử lý trong code behind**

Bao gồm có 2 phần chính: Gửi nội dung comment và hiển thị danh sách các comment hiện có trong tập tin ở server.

**Hàm xử lý sự kiện cho nút Gửi nội dung:**

```
protected void Btn_GuiND_Click(Object sender, EventArgs e)
{
 //Tạo file theo đường dẫn của website
 string sfile = "D://BaiTH_Web2//Lab2//data.txt";
 using (StreamWriter writer = new StreamWriter(sfile, true))
 {
 writer.WriteLine(txtTieude.Text);
 writer.WriteLine(txtHoTen.Text);
 writer.WriteLine(txtEmail.Text);
 writer.WriteLine(txtNoidung.Text);
 writer.WriteLine("#END");
 writer.Close(); //Đóng file sau khi ghi
 }
 EntryComment.InnerHtml = "Ban da gui thanh cong!";
}
```

**Hàm hiển thị nội dung hiện có trong tập tin ở server**

```
protected void ReadComment()
{
 string stempt = " ";
 string sfile = "D://BaiTH_Web2//Lab2//data.txt";
 using (StreamReader reader = new StreamReader(sfile))
 {
```

```

string snoidung = reader.ReadToEnd(); //đọc nội dung
string[] delimiter = { "#END" };
string[] sArr = snoidung.Split(delimiter,
 StringSplitOptions.RemoveEmptyEntries);
//Lọc ra từng Entry và đưa vào mảng chuỗi
foreach (string s in sArr)
{
 stempt = stempt + s + "</br>";
}

EntryComment.InnerHtml = "Noi dung " + stempt;
}

```

**Trong phần xử lý đọc file ta làm theo các bước sau**

- Đọc toàn bộ nội dung file comment
- Chia các entry vào từng chuỗi
- Cuối cùng định dạng nội dung comment vào trong tag <td> và đưa vào thuộc tính
- InnerHTML của span có tên EntryComment.

#### **d. Phần mở rộng:**

Sinh viên bổ sung thêm các thành phần sau:

- Ngày tháng năm viết comment và hiển thị thông tin này lên cùng với entry comment.
  - Thử không sử dụng tag span EntryComment mà dùng Web control Table để hiển thị các comment trên. Tìm hiểu các lớp **Table**, **TableRow** và **TableCell** trong phần tự xây dựng các Web control bằng cách sử dụng các lớp trong ASP.net.

---Hết---

## Lab 3.3: Sử dụng Validation Controls

### 1. Mục tiêu

- Sử dụng các **web control** trong trang ASP.NET như TextBox, Button, DropDownList
- Sử dụng input **validation controls** trong trang ASP.NET
  - RequiredFieldValidator
  - CompareValidator
  - RegularExpressionValidator
  - ValidationSummary
- Hiển thị các thông báo lỗi khi người dùng nhập dữ liệu không hợp lệ

### 2. Yêu cầu

- Đã làm quen với các server control cơ bản
- Biết xử lý các sự kiện của ASP.NET web control.

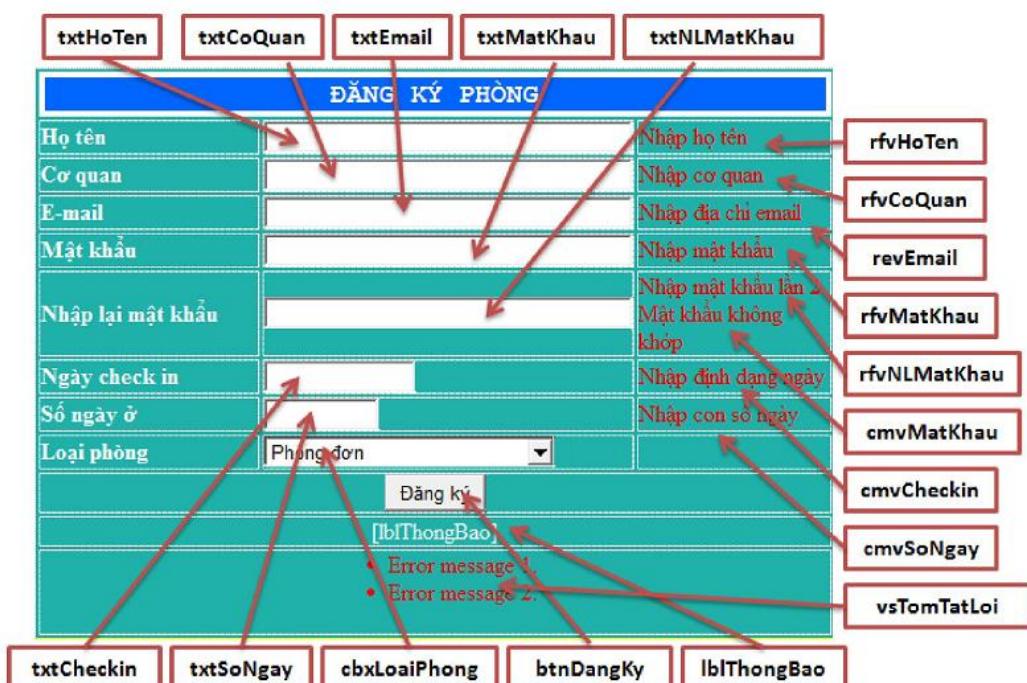
### 3. Nội dung

- Xây dựng trang web đăng ký phòng cho một hội thảo khoa học. Các nhà khoa học phải đăng ký online trước, tạo cho mình 1 account để sau khi đăng ký xong thì họ có thể đăng nhập vào trang web của hội nghị để đăng ký phòng, và xem các thông tin khác, cũng như các dịch vụ mà trong suốt quá trình lưu trú mà khách đã dùng. Chúng ta chỉ xây dựng trang **Register\_room.aspx**, yêu cầu user phải nhập các thông tin sau: *Họ tên, Đơn vị công tác; địa chỉ email (làm username đăng nhập), mật khẩu để đăng nhập, ngày check in và số ngày ở, chọn loại phòng.*

- Trang web này phải có đầy đủ chức năng kiểm tra dữ liệu khi user nhập vào. Sau khi tất cả dữ liệu hợp lệ thì thông báo việc đăng ký thành công!

### 4. Hướng dẫn:

#### a. Thiết kế giao diện:



#### b. Chi tiết của các control như sau:

Tên control	Kiểu	Thuộc tính	Điễn giải
txtHoTen	TextBox		
txtCoQuan	TextBox		
txtEmail	TextBox		
txtMatKhau	TextBox	TextMode = Password	
txtNLMatKhau	TextBox	TextMode = Password	
txtCheckin	TextBox		
txtSoNgay	TextBox		
cbxLoaiPhong	DropDownList	Items	{"Phòng đơn", "Phòng đôi", "Phòng VIP đơn", "Phòng VIP đôi"}
btnDangKy	Button		
lblThongBao	Label		
rfvHoTen	RequiredFieldValidator	ControlToValidate ErrorMessage	txtHoTen “Nhập họ tên”
rfvCoQuan	RequiredFieldValidator	ControlToValidate ErrorMessage	txtCoQuan “Nhập cơ quan”
revEmail	RegularExpressionValidator	ControlToValidate ErrorMessage ValidationExpression	txtEmail “Nhập địa chỉ email” Internet email address
rfvMatKhau	RequiredFieldValidator	ControlToValidate ErrorMessage	txtMatKhau “Nhập mật khẩu”
rfvNLMatKhau	RequiredFieldValidator	ControlToValidate ErrorMessage	txtNLMatKhau “Nhập mật khẩu lần 2”
cmvMatKhau	CompareValidator	ControlToValidate ErrorMessage ControlToCompare	txtNLMatKhau “Mật khẩu không khớp” txtMatkhau
cmvCheckin	CompareValidator	ControlToValidate ErrorMessage Type Operator	txtCheckin “Nhập ngày checkin” Date DataTypeCheck
cmvSoNgay	CompareValidator	ControlToValidate ErrorMessage Type Operator	txtSoNgay “Nhập con số ngày” Integer DataTypeCheck

Tên control	Kiểu	Thuộc tính	Điễn giải
vsTomTatLoi	ValidationSummary		

### c. Phần xử lý trong code behind

Phần xử lý sự kiện của nút lệnh btnDangky

- Khi người dùng nhập đầy đủ thông tin và thỏa các điều kiện trong các thành phần của web form. Sau đó, người dùng kích hoạt sự kiện bằng cách click vào nút lệnh “đăng ký” thì nhãn lbThongbao được gán với nội dung là: “**Bạn đã đăng ký thành công!!**”.
- Hiển thị thông tin chi tiết của việc đăng ký sau khi người dùng submit thông tin thành công, các thông tin này hiển thị trên lblThongBao. (*Bổ sung thêm phần xử lý của sự kiện click btnDangKy*).

# Chương 4: ĐIỀU KHIỂN CHUYỂN HƯỚNG DỮ LIỆU

## Mục tiêu học tập:

- Sử dụng được Multiview, wizards để tạo trang web
- Sử dụng được TreeView để tạo trang web
- Sử dụng Menu View để tạo trang web
- Sử dụng Sitemap để tạo bản đồ cho website

### I. Multiview và Wizard Controls

#### 1.1. MultiView Control

Hầu hết các website cần tách các công việc thành nhiều trang khác nhau. Ví dụ, nếu ta cần thêm một mẫu tin vào shopping cart, ta cần phải chuyển từ trang này sang trang khác. Multiview là một công cụ rất hiệu quả để thực hiện công việc đó.

Trong tình huống khác, ta cần phải nhúng mã vào một vài trang nhỏ hơn bên trong một trang chính. Ví dụ, ta cần tạo nhiều views cho cùng một loại dữ liệu và cho phép người dùng chuyển từ view này sang view khác mà không cần rời trang web.

Để tạo mới một Multiview với 3 view view1, view2 và view3 như sau:

```
<asp:MultiView ID="MultiView1" runat="server">
 <asp:View ID="View1" runat="server">...</asp:View>
 <asp:View ID="View2" runat="server">...</asp:View>
 <asp:View ID="View3" runat="server">...</asp:View>
</asp:MultiView>
```

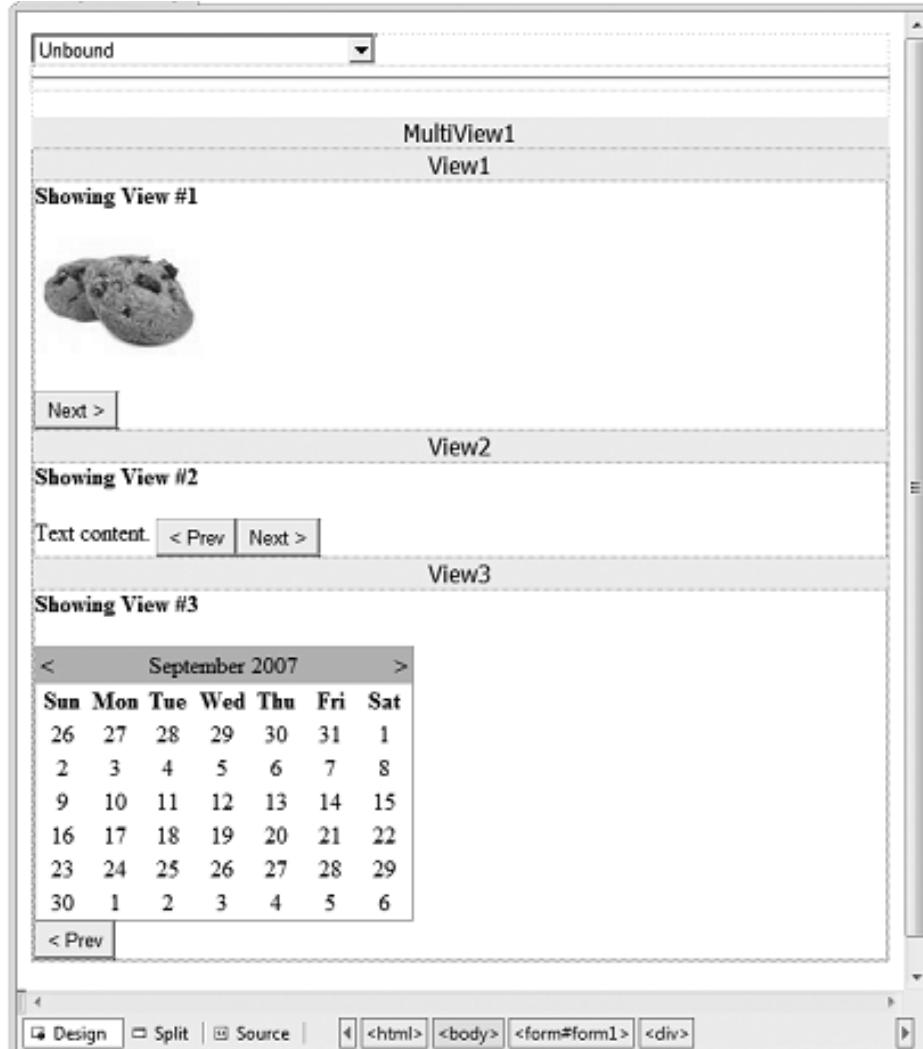
Bên trong thẻ `<asp:View>`, ta thêm thẻ HTML và web controls đối với các view tương ứng.

```
<asp:MultiView ID="MultiView1" runat="server" ActiveViewIndex="0">
 <asp:View ID="View1" runat="server">
 Showing View #1

 <asp:Image ID="Image1" runat="server" ImageUrl="~/cookies.jpg" />
 </asp:View>
 <asp:View ID="View2" runat="server">
 Showing View #2

 Text content.
 </asp:View>
 <asp:View ID="View3" runat="server">
 Showing View #3

 <asp:Calendar ID="Calendar1" runat="server"></asp:Calendar>
 </asp:View>
</asp:MultiView>
```



**Hình 18: Màn hình thiết kế với các View**

```

protected void Page_Load(object sender, EventArgs e)
{
 if (!Page.IsPostBack)
 {
 DropDownList1.DataSource = MultiView1.Views;
 DropDownList1.DataTextField = "ID";
 DropDownList1.DataBind();
 }
}

// Đoạn code sau đây sẽ đặt view hiện tại tùy thuộc vào chỉ mục trong list
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
 MultiView1.ActiveViewIndex = DropDownList1.SelectedIndex;
}

```

Thuộc tính MultiView.ActiveIndex xác định tên view được hiển thị. Mặc định Activeindex có giá trị là -1 có nghĩa là không có view nào thể hiện cả.

## 1.2. Wizard Control

Các điều khiển Wizard controls cũng tạo ra các MultiView controls. Nó cho phép hiển thị một vài Views cùng một lúc, nhưng nó có thể thêm vào các navigation controls khác như nút lệnh, sidebar, links, styles hoặc templates

Thông thường, wizards thể hiện một công việc riêng rẽ theo trình tự nào đó, người dùng chuyển tuần tự thông qua từng công việc. Bên cạnh đó nó cũng cho phép người dùng di chuyển một cách ngẫu nhiên giữa các công việc hoặc có thể bỏ một số việc không cần thiết.

Ví dụ sau đây là các bước thực hiện đăng ký thông tin của khách hàng vào hệ thống. Việc đăng ký được chia thành 4 công việc nhỏ hơn. Mỗi việc được thực hiện ở một view riêng rẽ.

The image displays three sequential screens of a wizard control, each with a header, navigation links, and specific content:

- Header Template - Thông tin cá nhân**  
Left sidebar: [Thông tin cá nhân](#), [Thông tin công ty](#), [Phần mềm](#), [Hoàn tất](#).  
Content: **Thông tin cá nhân**, dropdown "Ngôn ngữ lập trình yêu thích": C#, [Next](#).
- Header Template - Thông tin công ty**  
Left sidebar: [Thông tin cá nhân](#), [Thông tin công ty](#), [Phần mềm](#), [Hoàn tất](#).  
Content: **Thông tin về công ty:**, "Số lượng nhân viên: 34", "Số lượng địa điểm: 2", [Previous](#), [Next](#).
- Header Template - Phần mềm**  
Left sidebar: [Thông tin cá nhân](#), [Thông tin công ty](#), [Phần mềm](#), [Hoàn tất](#).  
Content: **Thông tin phần mềm**, "Yêu cầu đăng ký:",  Visual Studio,  Office,  Windows 2003 Server,  SQL Server 2005,  BizTalk 2004, [Previous](#), [Finish](#).

Bạn đã chọn:  
Ngôn ngữ lập trình yêu thích: C#  
Tổng số nhân viên: 34  
Tổng số địa điểm: 2  
Yêu cầu đăng ký: Visual Studio Windows 2003 Server

**Cảm ơn đã điền đầy đủ thông tin.  
Sản phẩm sẽ được giao đến bạn nhanh chóng.**

**Hình 19: Màn hình Kết quả thiết kế điều khiển Wizard**

```
<asp:Wizard ID="Wizard1" runat="server" Width="467px"
 BackColor="#EFF3FB" BorderColor="#B5C7DE" BorderWidth="1px">
 <WizardSteps>
 <asp:WizardStep ID="WizardStep1" runat="server" Title="Personal">
 <h3> Thông tin cá nhân </h3>
 Ngôn ngữ lập trình yêu thích:
 <asp:DropDownList ID="lstLanguage" runat="server" >
 <asp:ListItem> C# </asp:ListItem>
 <asp:ListItem> VB </asp:ListItem>
 <asp:ListItem> J# </asp:ListItem>
 <asp:ListItem> Java </asp:ListItem>
 <asp:ListItem> C++ </asp:ListItem>
 <asp:ListItem> C </asp:ListItem>
 </asp:DropDownList>

 </asp:WizardStep>
 <asp:WizardStep ID="WizardStep2" runat="server" Title="Company">
 <h3> Company Profile </h3>
 Số nhân viên: <asp:TextBox ID="txtEmpCount" runat="server" />
 Số địa điểm: <asp:TextBox ID="txtLocCount" runat="server" />
 </asp:WizardStep>
 <asp:WizardStep ID="WizardStep3" runat="server" Title="Software" >
 <h3> Thông tin phần mềm </h3> Licenses Required:
 <asp:CheckBoxList ID="lstTools" runat="server" >
 <asp:ListItem> Visual Studio 2008 </asp:ListItem>
 <asp:ListItem> Office 2007 </asp:ListItem>
 <asp:ListItem> Windows Server 2008 </asp:ListItem>
 <asp:ListItem> SQL Server 2008 </asp:ListItem>
 </asp:CheckBoxList>
 </asp:WizardStep>
 </WizardSteps>
</asp:Wizard>
```

```

<asp:WizardStep ID="Complete" runat="server" Title="Complete"
 StepType="Complete" >

 Cảm ơn bạn đã điền đầy đủ thông tin.

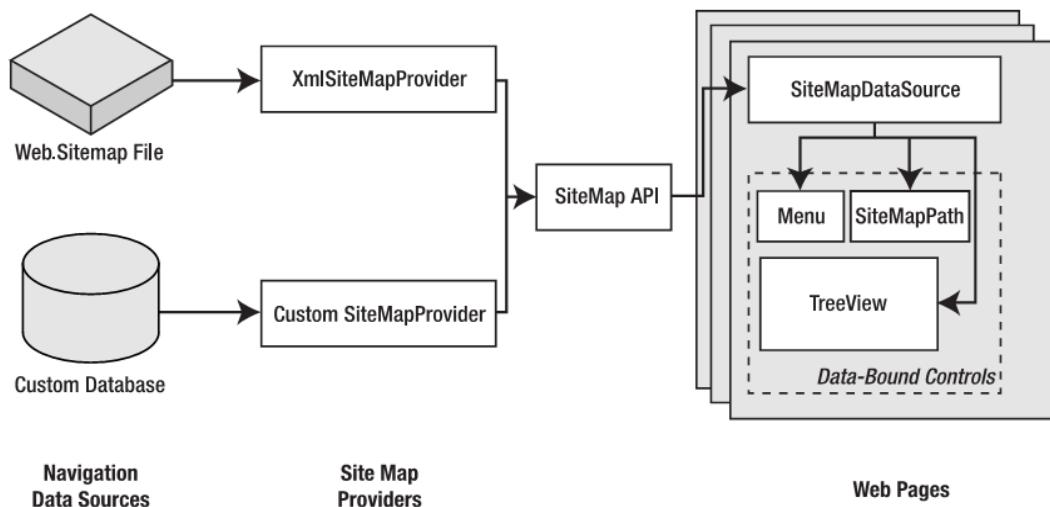
 Sản phẩm sẽ được giao nhanh chóng.

</asp:WizardStep>
</WizardSteps>
</Wizard>

```

## II. Site Map

Nếu website của ta có nhiều trang web, ta cần có một cách nào đó để sắp xếp các trang để người dùng có thể di chuyển từ trang này sang trang khác một cách dễ dàng.



Hình 20: Lược đồ liên kết trong SiteMap

### 2.1. Tạo một file Web.sitemap file

- Để thêm sitemap ta thực hiện Website *Add New Item (hoặc Project Add New Item trong web project)*, chọn Site Map click Add.
- Xác định cấu trúc của website sử dụng thành phần `<siteMap>` và `<siteMapNode>`. **Liên kết SiteMap với điều khiển TreeView hoặc MenuView.**

### 2.2. Ví dụ: Tạo tập tin Sitemap với các thành phần như sau

```

<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0">
 <siteMapNode title="Home" description="Home" url="~/default.aspx">
 <siteMapNode title="Products" description="Our products" url="~/Products.aspx">
 <siteMapNode title="Hardware" description="Hardware choices" url="~/Hardware.aspx" />
 <siteMapNode title="Software" description="Software choices" url="~/Software.aspx" />
 </siteMapNode>
</siteMap>

```

```

</siteMapNode>
<siteMapNode title="Services" description="Services we offer" url="~/Services.aspx">
 <siteMapNode title="Training" description="Training classes" url="~/Training.aspx" />
 <siteMapNode title="Consulting" description="Consulting services"
 url="~/Consulting.aspx" />
 <siteMapNode title="Support" description="Support plans" url="~/Support.aspx" />
</siteMapNode>
</siteMapNode>
</siteMap>

```

### III. Điều khiển TreeView

#### 3.1.Tạo một TreeView đơn giản

Tree View là một điều khiển hướng rất được ưa chuộng so với các điều khiển hướng khác. Tree view có thể phát sinh mã HTML để trả về phía trình khách. Tree View cũng cung cấp nhiều mẫu định dạng cho phép lập trình viên chọn loại phù hợp với ứng dụng của mình.

```

<asp:TreeView ID="TreeView1" runat="server" >
 <Nodes>
 <asp:TreeNode Text="Products" >
 <asp:TreeNode Text="Hardware" />
 </asp:TreeNode>
 <asp:TreeNode Text="Services" />
 </Nodes>
</asp:TreeView>

```

Ta có thể lập trình để thêm một node vào treeview khi trang web được load:

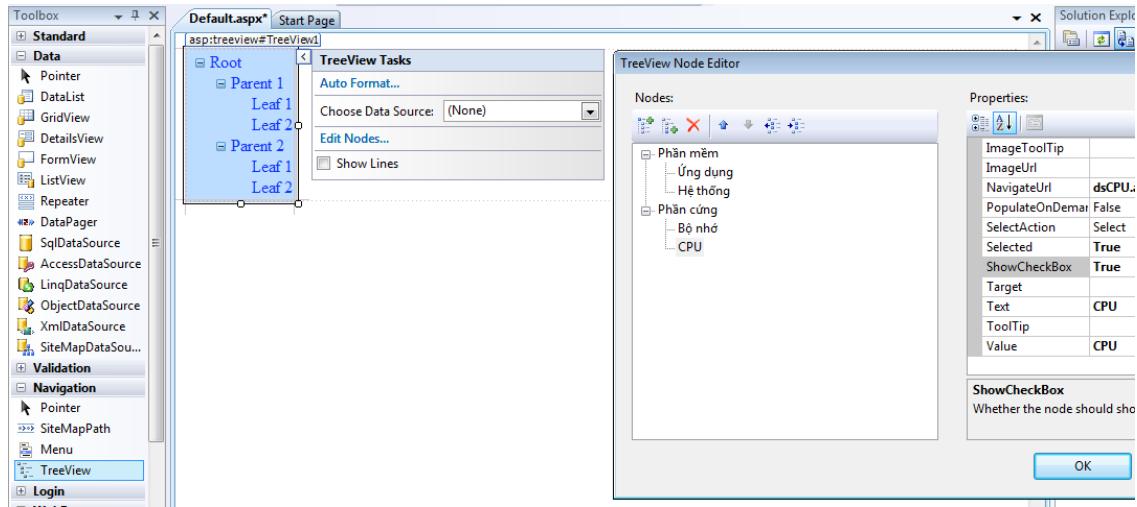
```

TreeNode newNode = new TreeNode("Software");
// Thêm vào nút con đầu tiên Add as a child of the first root node
// (the Products node in the previous example).
TreeView1.Nodes[0].ChildNodes.Add(newNode);

```

Thuộc tính	Mô tả
Text	Chuỗi văn bản hiển thị
ToolTip	Chuỗi văn bản khi ta click chuột vào mục chọn
Value	Lưu trữ giá trị không được hiển thị với dữ liệu thêm vào node
NavigateUrl	Địa chỉ trang web sẽ chuyển đến khi người dùng click chuột vào mục chọn
Target	Nếu thuộc tính NavigateUrl được đặt, nó sẽ mở trang web ở cửa sổ được chỉ định. Nếu target không được đặt, trang web mới được mở trong cửa sổ hiện thời
ImageUrl	Hình ảnh được hiển thị kế nút của cây

## Các thuộc tính của mỗi nút trên cây



Hình 21: Màn hình thiết kế TreeView ở chế độ Design

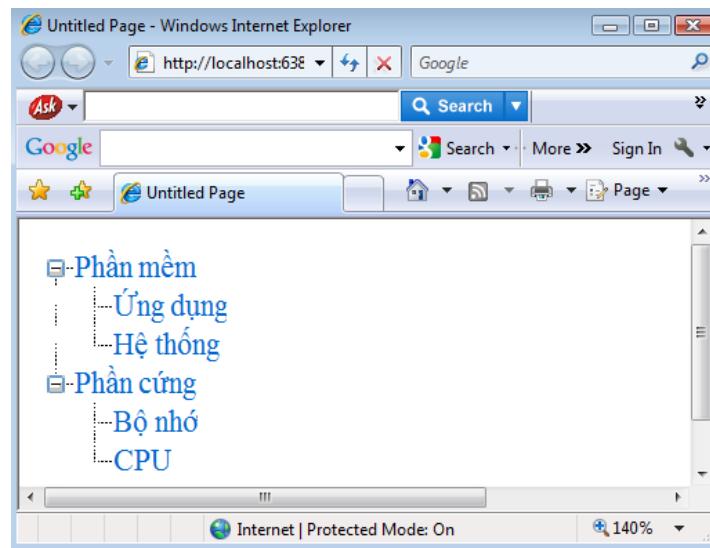
### Các bước thực hiện tạo một TreeView bằng công cụ Wizard.

- Chọn TreeView từ nhóm Navigation của hộp thoại Toolbox.
- Vẽ một vùng trong khu vực soạn thảo ➔ chọn Edit Nodes để mở hộp cửa sổ soạn thảo TreeView
- Trong lúc tạo các mục cho Treeview, ta cần phải xác định giá trị cho một số thuộc tính theo bảng đã cho ở trên.

Đoạn mã HTML tự phát sinh như sau:

```
<asp:TreeView ID="TreeView1" runat="server" ShowLines="True">
 <Nodes>
 <asp:TreeNode Text="Phần mềm" Value="Phần mềm">
 <asp:TreeNode NavigateUrl="Ungdung.aspx" Text="Ứng dụng" Value="ud">
 </asp:TreeNode>
 <asp:TreeNode NavigateUrl="Hethong.aspx" Text="Hệ thống" Value="ht">
 </asp:TreeNode>
 </asp:TreeNode>
 <asp:TreeNode Text="Phần cứng" Value="Phần cứng">
 <asp:TreeNode NavigateUrl="Bonho.aspx" Text="Bộ nhớ" Value="bn">
 </asp:TreeNode>
 <asp:TreeNode NavigateUrl="CPU.aspx" Text="CPU" Value="4">
 </asp:TreeNode>
 </asp:TreeNode>
 </Nodes>
</asp:TreeView>
```

Màn hình kết quả của thiết kế được tạo ở trên



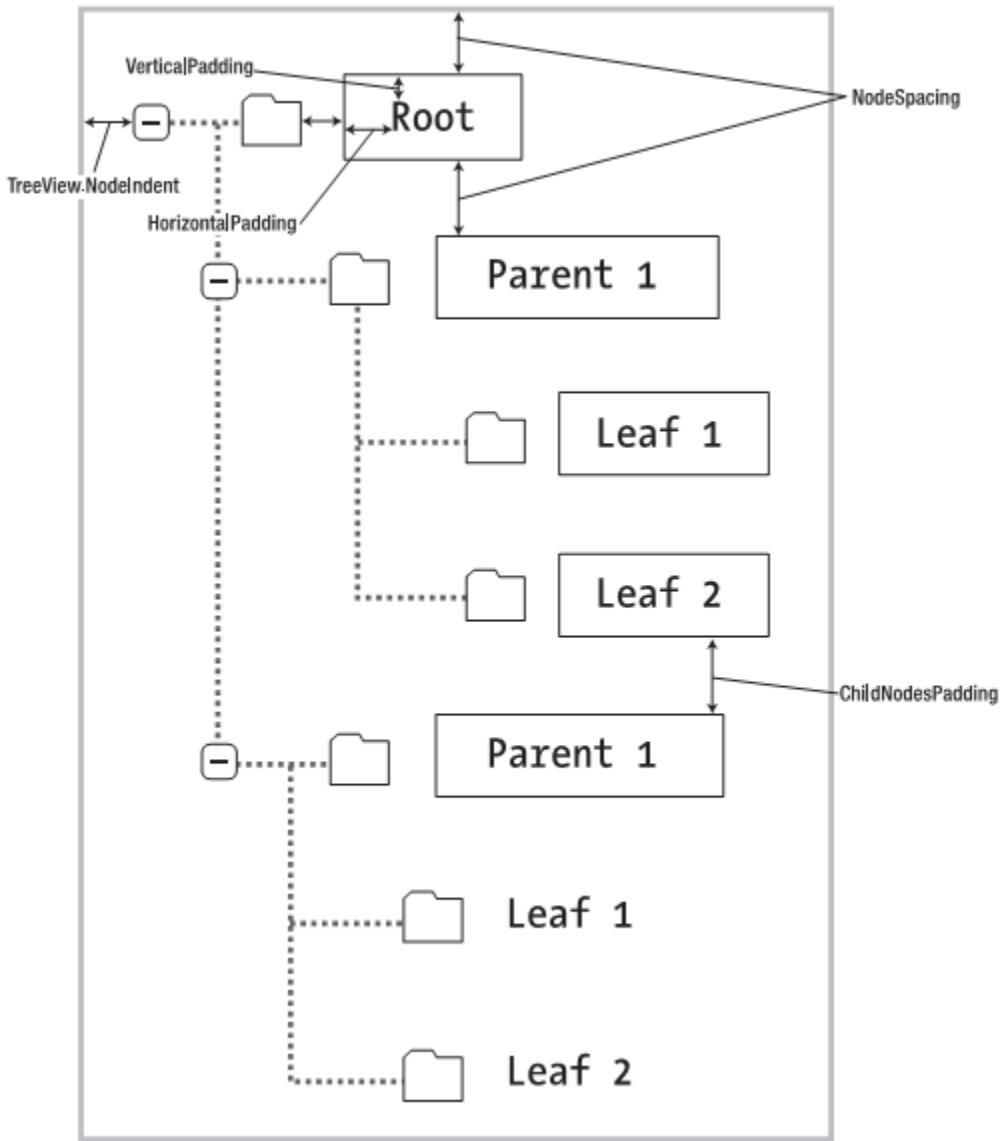
**Hình 22: Màn hình Kết quả Form với TreeView**

### 3.2. Định dạng Tree View

#### a. Định dạng các nút của TreeView

Thuộc tính	Mô tả
ImageUrl	Địa chỉ của hình ảnh được hiển thị gần mỗi nút
NodeSpacing	Khoảng cách giữa nút hiện thời và các nút lân cận
Vertical Padding	Khoảng cách giữa nút trên và nút dưới
Horizontal Padding	Khoảng cách giữa bên phải và bên trái của nút và đường biên xung quanh text
ChildNodesPadding	Khoảng cách giữa nút cuối cùng và nút cha

**Các thuộc tính định dạng cho TreeView**

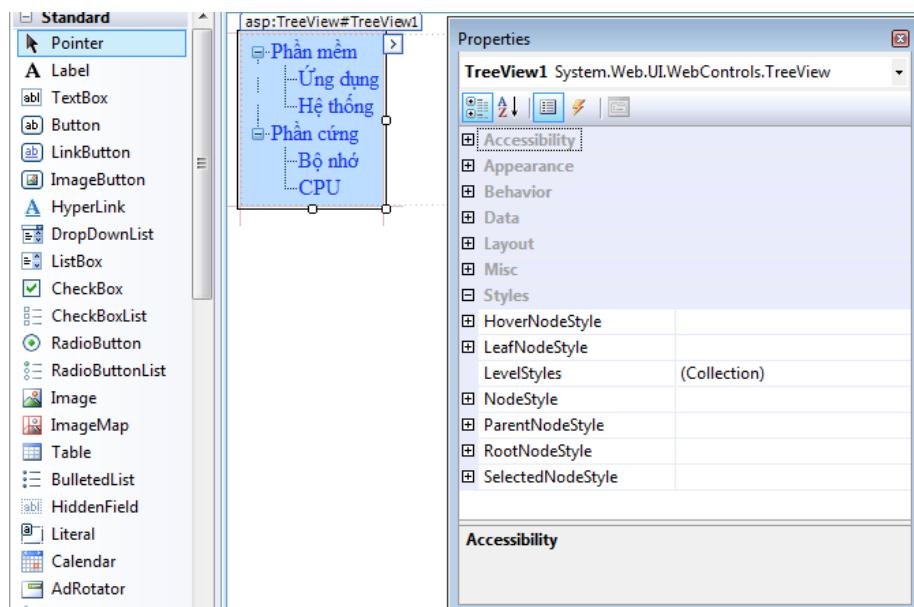


*Hình 23: Các thuộc tính định dạng các nút trong TreeView*

b. Các thuộc tính định dạng cho TreeView

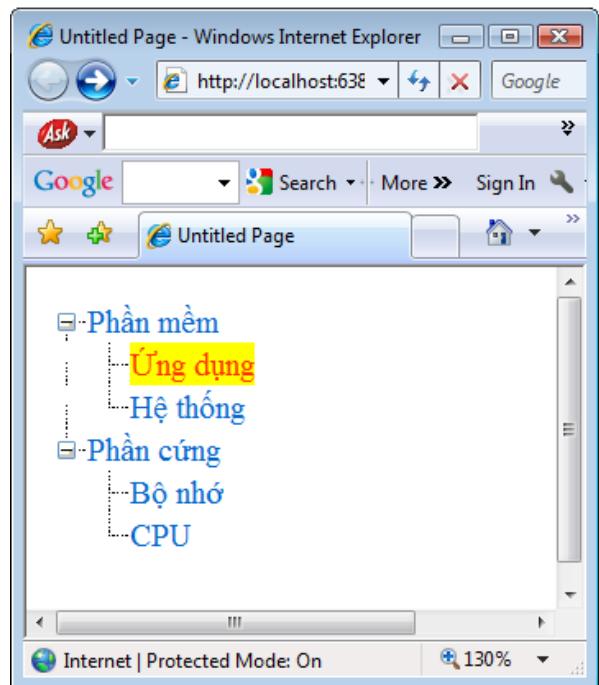
Thuộc tính	Mô tả
NodeStyle	Áp dụng định dạng đối với tất cả các nút
RootNodeStyle	Áp dụng định dạng cho các nút ở mức đầu tiên root
ParentNodeStyle	Áp dụng định dạng cho tất cả các nút có chứa nút khác, ngoại trừ nút gốc
LeafNodeStyle	Áp dụng định dạng cho tất cả các nút không chứa nút con và không chứa nút gốc
SelectedNodeStyle	Áp dụng định dạng cho các nút được chọn hiện thời
HoverNodeStyle	Áp dụng định dạng khi người dùng đưa trỏ chuột đến nút chọn

**Các thuộc tính định dạng cho TreeView**



*Hình 24: Các thuộc tính định dạng TreeView*

- c. Ví dụ: Định dạng TreeView ở trên với định dạng HoverNodeStyle với màu chữ là màu đỏ và màu nền là màu vàng khi người dùng thực hiện đưa chuột vào mục chọn trên cây TreeView.



*Hình 25: Các thuộc tính định dạng TreeView (2)*

Đoạn mã phát sinh như sau:

```

<asp:TreeView ID="TreeView1" runat="server" ShowLines="True">
 <HoverNodeStyle BackColor="Yellow" ForeColor="Red" />
 <SelectedNodeStyle BackColor="Green" />
 <Nodes>
 <asp:TreeNode Text="Phần mềm" Value="pm">
 <asp:TreeNode NavigateUrl="Ungdung.aspx" Text="Ứng dụng" Value="ud">
 </asp:TreeNode>
 <asp:TreeNode NavigateUrl="Hethong.aspx" Text="Hệ thống" Value="ht">
 </asp:TreeNode>
 </asp:TreeNode>
 <asp:TreeNode Text="Phần cứng" Value="pc">
 <asp:TreeNode NavigateUrl="Bonho.aspx" Text="Bộ nhớ" Value="bn">
 </asp:TreeNode>
 <asp:TreeNode NavigateUrl="CPU.aspx" SelectAction="Expand"
 Text="CPU" Value="cpu"></asp:TreeNode>
 </asp:TreeNode>
 </Nodes>
 </asp:TreeView>

```

### 3.3. Liên kết TreeView với SiteMap

- Tạo tập tin SiteMap với nội dung như sau:

```

<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
 <siteMapNode title="Trang chủ">
 <siteMapNode url="" title="Phần mềm" description="">
 <siteMapNode url="Ungdung.aspx" title="Ứng dụng" description="" />
 <siteMapNode url="Hethong.aspx" title="Hệ thống" description="" />
 </siteMapNode>

 <siteMapNode url="" title="Phần cứng" description="">
 <siteMapNode url="Bonho.aspx" title="Bộ nhớ" description="" />
 <siteMapNode url="CPU.aspx" title="CPU" description="" />
 </siteMapNode>
 </siteMapNode>
</siteMap>

```

- Liên kết SiteMap với Treeview

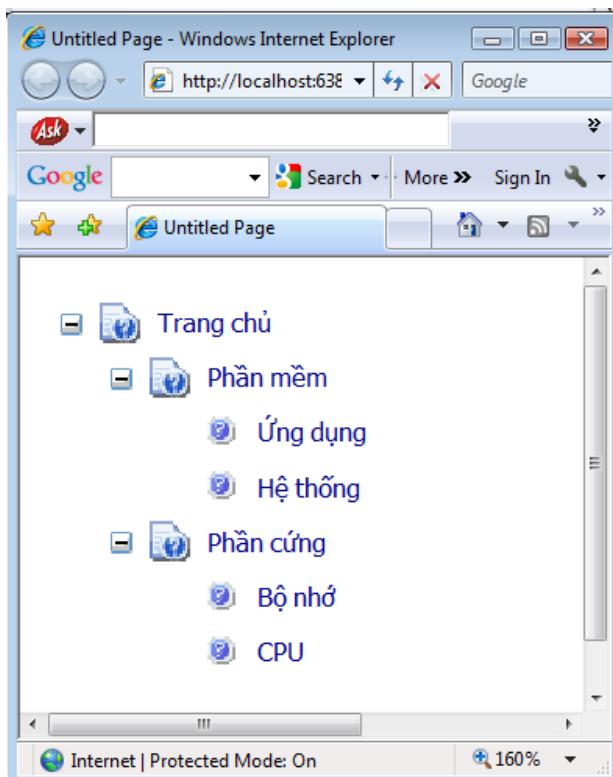
```

<asp:TreeView ID="TreeView1" runat="server" DataSourceID="sourceSiteMap"
 ShowLines="True"> </asp:TreeView>
<asp:SiteMapDataSource ID="sourceSiteMap" runat="server" />

```

Sinh viên có thể sử dụng công cụ Wizard của Visual Studio để thực hiện khai báo nguồn dữ liệu site map với tên là sourceSiteMap và công điều khiển TreeView với tên là TreeView1. Sau đó sử dụng tính năng Autoformat để có được hình ảnh TreeView như hình sau. Đây là công việc khá đơn giản.

- Màn hình kết quả như sau:



**Hình 26: Form kết quả TreeView (3)**

## IV. Điều khiển MenuView

### 4.1. Tạo một Menu đơn giản

Điều khiển Menu là một cách thức trình bày khác để hiển thị dữ liệu dạng cấu trúc. Giống như TreeView, ta có thể liên kết Menu với nguồn dữ liệu hoặc có thể tạo trực tiếp từng Mục chọn (MenuItem).

Lớp MenuItem trong điều khiển MenuView không giàu tính năng như TreeView. Ví dụ MenuItem không hỗ trợ chức năng Checkbox để lập trình. Tuy nhiên MenuView vẫn có một số thuộc tính phổ biến như: hình ảnh, định dạng khi chọn mục chọn và xác định liên kết cho mục chọn.

Thuộc tính	Mô tả
Text	Đoạn Text hiển thị trong mục chọn MenuItem
ToolTip	Đoạn text hiển thị khi người dùng đưa chuột vào mục chọn
Value	Là giá trị không hiển thị bên ngoài, Value chứa thông tin về mục chọn ( <i>thích hợp cho lập trình</i> )
NavigateUrl	Nếu mục chọn được chọn, trang web sẽ được chuyển đến một trang web khác được chỉ định
Target	Nếu ta gán giá trị cho NavigateUrl, ta cần xác định trang web chuyển đến sẽ hiển thị ở đâu trong target. Có thể là frame nào đó, có thể là ở một cửa sổ trình duyệt mới
Selectable	Nếu là false, mục chọn sẽ không được chọn

## Một số thuộc tính của mục chọn MenuItem

### 4.2. Định dạng MenuView

Giống như TreeView, MenuView hỗ trợ định dạng mục chọn ở nhiều mức khác nhau. Điều khiển MenuView cho phép ta chọn lựa giữa hai loại mục chọn static và dynamic. Mục chọn Static cho phép người dùng định dạng

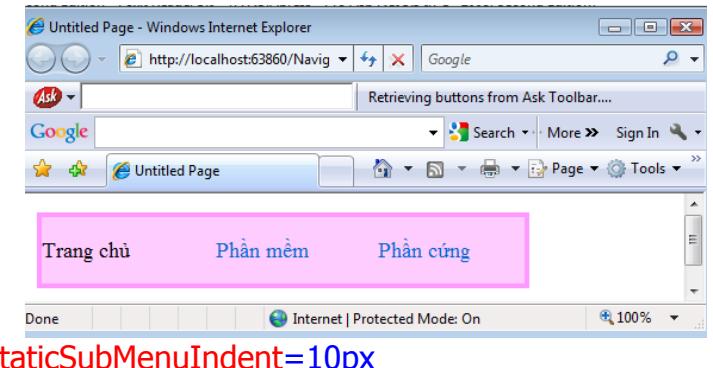
Loại Static	Loại Dynamic	Mô tả
StaticMenuItemStyle	DynamicMenuItemStyle	Nếu StaticMenuItemStyle được gán. Menu sẽ được hiển thị trên trang. Ngược lại Menu sẽ hiển thị dưới dạng Popup
StaticMenuItemStyle	DynamicMenuItemStyle	Gán thuộc tính định dạng cho mỗi mục chọn MenuItem trong menu
StaticSelectedStyle	DynamicSelectedStyle	Gán thuộc tính định dạng cho các mục được chọn (sau khi người dùng click vào mục nào đó)
StaticHoverStyle	Dynamic HoverStyle	Gán thuộc tính định dạng cho các mục chọn khi người dùng đưa mũi tên chuột vào mục chọn

### 4.3. Ví dụ Menu

- + Nguồn dữ liệu để hiển thị menu là sourceSiteMap giống như phần Treeview
- + Một số thuộc tính được chọn như sau:

Orientation=Horizontal

StaticDisplayLevels=2



StaticSubMenuIndent=10px



+ Nguồn dữ liệu hiển thị cho menu dọc là tập tin SiteMap như ví dụ trên

+ Một số thuộc tính được chọn là:

StaticDisplayLevels=2

StaticSubMenuIndent=5px

Orientation = Vertical

Hình 27: Form kết quả Menu

## Câu hỏi/Bài tập củng cố

1. Nêu các thuộc tính của TreeNode
2. Nêu các thuộc tính thông dụng để định dạng TreeView
3. Sử dụng điều khiển Wizards để tạo form nhập liệu giống như trang 56,57
4. Tạo TreeView giống như trang 63 và trang 65
5. Tạo menu dọc và ngang giống như trang 66 bằng cách viết code và sử dụng công cụ wizard

## Chương 5: ADO.NET

### Bài 1: TỔNG QUAN VỀ ADO.NET

#### Mục tiêu học tập

- Trình bày kiến trúc về ADO.net
- Sử dụng đối tượng Connection để kết nối với cơ sở dữ liệu (CSDL)
- Kiểm tra Kết nối với CSDL
- Trình bày thông tin được truy xuất trên trang web.

#### I. Giới thiệu

Hầu hết các ứng dụng bao gồm ứng dụng windows và ứng dụng web đều là các ứng dụng theo hướng dữ liệu. Các công việc chính thường là truy xuất, hiển thị và chỉnh sửa dữ liệu. Truy xuất và xử lý dữ liệu thường được xem là công việc đơn giản, tuy nhiên thời gian sau này các việc này có vẻ khá phức tạp. Cụ thể là, các lập trình viên đã phải chuyển từ truy xuất dữ liệu đơn giản trên các cơ sở dữ liệu cục bộ đến các hệ thống phân tán trên các cơ sở dữ liệu trung tâm nào đó. Tương tự như vậy, các công nghệ truy xuất dữ liệu cũng phát triển rất nhanh. Ví dụ Microsoft đã phát triển các công nghệ truy xuất dữ như ODBC, DAO, RDO, RDS, và ADO.

.NET Framework bao gồm công nghệ truy xuất dữ liệu riêng gọi là ADO.NET. ADO.NET bao gồm các lớp cho phép các ứng dụng .NET truy xuất với các nguồn dữ liệu (thông thường là dữ cơ sở dữ liệu quan hệ), thực thi các câu lệnh, và quản lý dữ liệu không kết nối.

Chương này sẽ mô tả kiến trúc ADO.NET và nhà cung cấp dữ liệu ADO.NET. Bạn sẽ học các chức năng cơ bản về ASP.net như mở kết nối, thực thi câu lệnh SQL, stored procedure, và truy vấn kết quả của câu truy vấn.

#### Truy xuất dữ liệu không sử dụng ADO.net

Trong ASP.NET, có vài cách để truy xuất thông tin khỏi cơ sở dữ liệu trực tiếp mà không sử dụng các lớp ADO.Net. Tùy thuộc vào yêu cầu công việc của bạn, mà có thể sử dụng một trong các cách sau đây:

- **Điều khiển SqlDataSource:** Điều khiển SqlDataSource cho phép bạn định nghĩa các câu truy vấn rõ ràng. Bạn có thể kết nối SqlDataSource với các điều khiển rich controls như GridView và cung cấp các trang web của bạn khả năng để chỉnh sửa, cập nhật dữ liệu mà không viết các câu lệnh ADO.NET. Tuy nhiên, SqlDataSource có một vài điểm hạn chế bởi vì nó đặt việc xử lý dữ liệu vào trong phần thẻ của trang web. Các nhà phát triển web thích sử ObjectDataSource, tương tự như chucuc nang ket noi dữ liệu trong data binding nhưng SqlDataSource sử dụng các gói dữ liệu. Khi bạn sử dụng ObjectDataSource, bạn có thể tạo các thành phần dữ liệu và viết các mã lệnh ADO.net phía sau.

- **LINQ to SQL:** Với LINQ to SQL, bạn cần viết các câu truy vấn sử dụng mã lệnh C# code (or the LinqDataSource control) và cơ sở dữ liệu được phát sinh tự động. LINQ to SQL hỗ trợ cập nhật, phát sinh các câu lệnh SQL. Giống như các điều khiển SQLDataSource, LINQ to SQL không cho phép lập trình viên thực thi các câu lệnh trên cơ sở dữ liệu mà không khớp với câu truy vấn trực tiếp và lệnh cập nhật (các lệnh tạo các bảng). Tuy nhiên, không giống như điều khiển SqlDataSource, LINQ to SQL chỉ làm việc được SQL Server và hoàn toàn độc lập với ADO.

- **Profiles:** Đặc điểm của profiles cho phép bạn lưu trữ khối dữ liệu trong cơ sở dữ liệu mà không sử dụng mã lệnh ADO.NET. Tuy nhiên đặc tính khói này thì cực kì giới hạn. ví dụ, nó không cho phép bạn điều khiển trên cấu trúc vật lý của bảng dữ liệu. Kết quả là, các nhà phát triển ứng dụng thường tạo các nhà cung cấp profile riêng với các lệnh ADO.net.

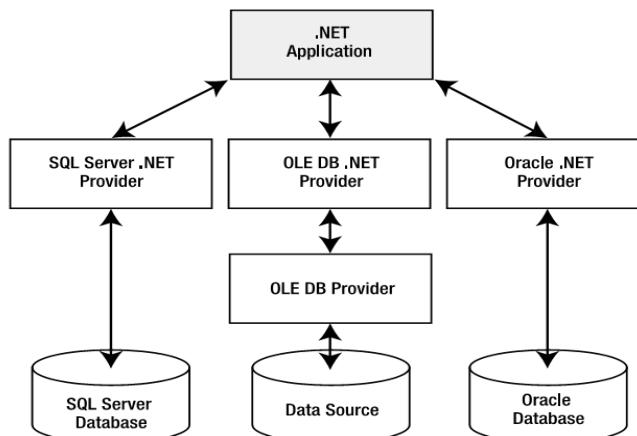
## II. Kiến trúc của ADO.NET

ADO.net sử dụng kiến trúc đa tầng để giải quyết các vấn đề về đối tượng kết nối (**Connection**), đối tượng câu lệnh (**Command**) và đối tượng tập dữ liệu (**DataSet**).

### 2.1. ADO.NET Data Providers

Nhà cung cấp dữ liệu là tập hợp các lớp ADO.net cho phép thực hiện truy xuất đến cơ sở dữ liệu nào đó, thực hiện các câu lệnh SQL và truy vấn dữ liệu. Nhà cung cấp dữ liệu là một cầu nối giữa chương trình ứng dụng và nguồn dữ liệu. Các lớp tạo thành một Data Provider bao gồm:

- **Connection:** Sử dụng đối tượng này để thiết lập một kết nối với nguồn dữ liệu (data source).
- **Command:** Sử dụng đối tượng này để thực hiện câu lệnh SQL và thủ tục stored procedures.
- **DataReader:** Truy xuất dữ liệu trả về từ kết quả câu truy vấn.
- **DataAdapter:** Thực hiện 2 nhiệm vụ: Sử dụng **DataAdapter** để đổ dữ liệu vào **DataSet** với thông tin được rút ra từ **Data source**. Ta cũng có thể sử dụng DataAdapter để thay đổi **data Source** phù hợp với những chỉnh sửa được thực hiện trong **DataSet**.



*Hình 28: Kiến trúc ADO.net*

- **SQL Server provider:** Truy xuất đến CSDL SQL Server (version 7.0 hoặc sau)
- **OLE DB provider:** Truy xuất đến bất cứ nguồn dữ liệu nào có OLE DB driver. Bao gồm SQL Server databases trước version 7.0.
- **Oracle provider:** Cung cấp truy xuất đến cơ sở dữ liệu Oracle database (version 8i hoặc các bản sau này).
- **ODBC provider:** Cung cấp truy xuất đến bất cứ dữ liệu nguồn nào có ODBC driver.

### 2.2.Các lớp ADO.net cơ bản

ADO.NET có 2 kiểu đối tượng chính: đối tượng dựa trên kết nối (*connection-based*) và đối tượng dựa trên nội dung (*content-based*).

- Các đối tượng liên quan đến kết nối: Các đối tượng này bao gồm: Connection, Command, DataReader, và DataAdapter. Những đối tượng này được tìm trong namespace System.Data.SqlClient đối với SQL Server provider.

- Các đối tượng liên quan đến nội dung: Các đối tượng này bao gồm: DataSet, DataColumn, DataRow, DataRelation, và nhiều đối tượng khác tại namespace System.Data.

Các lớp trong ADO.net được nhóm lại thành một vài namespaces. Mỗi nhà cung cấp dữ liệu có namespace riêng và các lớp tổng quát như DataSet được lưu trữ trong namespaces System.Data.

Tên Namespace	Mô tả
<b>System.Data</b>	Chứa các lớp để làm việc liên quan đến columns, relations, tables, datasets, rows, views, and constraints.
<b>System.Data.Common</b>	Định nghĩa chức năng chính của ADO.NET. Data providers thừa kế từ những lớp này (như DbConnection, DbCommand) để tạo những version khác nhau.
<b>System.Data.OleDb</b>	Chứa các lớp để kết nối với OLE DB provider, bao gồm OleDbCommand, OleDbConnection, OleDbDataReader, và OleDbDataAdapter. Các lớp này hỗ trợ hầu hết nhà cung cấp OLE DB.
<b>System.Data.SqlClient</b>	Có các lớp để kết nối với CSDL Microsoft SQL Server như SqlCommand, SqlConnection, SqlDataReader, và SqlDataAdapter.
<b>System.Data.OracleClient</b>	Chứa các lớp để kết nối với CSDL Oracle(version 8.1.7 or later), bao gồm OracleCommand, OracleConnection, OracleDataReader và OracleDataAdapter.
<b>System.Data.Odbc:</b>	Chứa các lớp để kết nối với hầu hết ODBC drivers. Những lớp này bao gồm OdbcCommand, OdbcConnection, OdbcDataReader, và OdbcDataAdapter.
<b>System.Data.SqlTypes</b>	Chứa các cấu trúc phù hợp với kiểu dữ liệu trong SQL Server. Những lớp này thì không cần thiết nhưng nó cung cấp một giải pháp chuẩn đối với kiểu dữ liệu .NET,

### III. Lớp Connection

#### 3.1. Chuỗi kết nối

Khi tạo một đối tượng kết nối Connection, cần phải cung cấp một chuỗi kết nối connectionstring. Một chuỗi kết nối **ConnectionString** bao gồm các cặp name/value (*tên/giá trị*) được phân cách bởi dấu “ ; ”. Thứ tự của các cặp thì không quan trọng và không phân biệt chữ hoa hay chữ thường. Connectionstring chỉ định các thông tin cơ bản cần thiết để thực hiện kết nối với CSDL.

Mặc dù mỗi nhà cung cấp dữ liệu yêu cầu các ConnectionString khác nhau. Tuy nhiên sau đây là các thông tin bắt buộc phải có:

- **Data Source:** Tên máy chủ chứa cơ sở dữ liệu
- **Initial Catalog:** Tên cơ sở dữ liệu
- **Integrated Security:** Cách cơ sở dữ liệu nhận dạng

### a. Kết nối dữ liệu với CSDL MS SQL Server

**Ví dụ:** Một chuỗi kết nối có tên connectionString để kết nối với cơ sở dữ liệu NorthWind trong MS SQL server như sau:

```
string connectionString = "Data Source=localhost; Initial Catalog=Northwind;
Integrated Security=SSPI";
```

Tên máy chủ chứa cơ sở dữ liệu: là Localhost; Tên cơ sở dữ liệu là Northwind; và nhận dạng là tài khoản người dùng windows để login vào cơ sở dữ liệu.

Nếu integrated security không được hỗ trợ, kết nối phải chỉ định rõ cả hai người dùng và mật khẩu truy cập. Đối với cơ sở dữ liệu SQL mới được thiết lập, tài khoản sa (*system administrator*) thường được sử dụng.

### b. Kết nối dữ liệu với CSDL Oracle

**Ví dụ:**

```
string connectionString = "Data Source=localhost; Initial Catalog=Northwind;
user id=sa; password=openesame";
```

Để kết nối vào nhà cung cấp dữ liệu OLE DB. Chuỗi kết nối cũng tương tự, tuy nhiên cần phải thêm vào chuỗi xác nhận driver OLE DB. Ví dụ sau cho thấy chuỗi kết nối với cơ sở dữ liệu Oracle thông qua nhà cung cấp MSDAORA OLE.

```
string connectionString = "Data Source=localhost; Initial Catalog=Sales; user id=sa;
password=da#ta_li#nk_43;Provider=MSDAORA";
```

### c. Kết nối với cơ sở dữ liệu Access:

```
string connectionString = "Provider=Microsoft.Jet.OLEDB.4.0; @"Data
Source=C:\DataSources\Northwind.mdb";
```

### d. Khai báo chuỗi kết nối trong tập tin web.config

Thay vì viết chuỗi kết nối trong code-behind rất phức tạp và mất nhiều thời gian. Ta có thể khai báo chuỗi kết nối trong tập tin web.config như sau:

```
<configuration>
 <connectionStrings>
 <add name="Northwind" connectionString=" Data Source=localhost;
 Initial Catalog=Northwind; Integrated Security=SSPI"/>
 </connectionStrings>
</configuration>
```

Khi đó để sử dụng chuỗi kết nối trên, ta sẽ sử dụng tập hợp WebConfigurationManager.ConnectionStrings thông qua câu lệnh như sau:

```
string connectionString =
 WebConfigurationManager.ConnectionStrings["Northwind"].ConnectionString;
```

#### 3.2. Kiểm tra kết nối:

Sau khi có chuỗi kết nối và khai báo đối tượng kết nối ta có thể sử dụng phương thức open() và close() để kiểm tra kết nối với cơ sở dữ liệu:

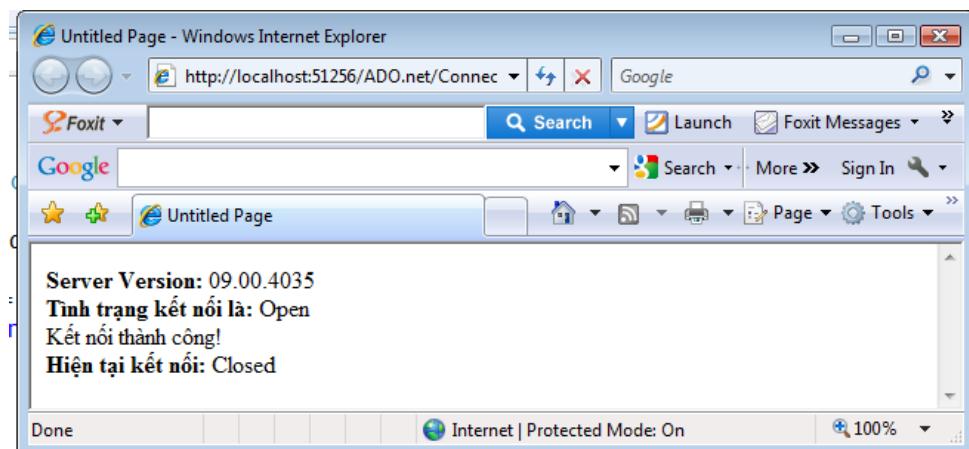
```
protected void Page_Load(object sender, EventArgs e)
{
```

```

 string connectionStr = "Data Source=localhost; Initial Catalog=QLNV; Integrated Security=SSPI";
 SqlConnection con = new SqlConnection(connectionStr);
 //Mở kết nối với CSDL
 try{
 con.Open();
 lblTB.Text += "Server Version: " + con.ServerVersion;
 lblTB.Text += "
Tình trạng kết nối là: " + con.State.ToString();
 lblTB.Text += "
Kết nối thành công!";
 }
 //Hiển thị thông báo lỗi nếu có lỗi phát sinh
 catch(SqlException ex){
 lblTB.Text += " Kết nối thất bại! " + ex.ToString();
 }
 //Thực hiện đóng kết nối nếu không còn làm việc với CSRD nữa
 finally{
 con.Close();
 lblTB.Text += "
Hiện tại kết nối: " + con.State.ToString();
 }
 }
}

```

Kết quả sau khi kết nối với cơ sở dữ liệu và đóng kết nối như sau:



**Hình 29: Kết quả Form Kiểm tra kết nối**

### Câu hỏi/Bài tập củng cố

1. Trình bày kiến trúc về ADO.net
2. Hãy nêu các thuộc tính cần thiết để tạo chuỗi kết nối
3. Nêu tên thư viện có chứa lớp Connection
4. Nêu tên các phương thức thông dụng của lớp Connection.
5. Tạo form kết nối với cơ sở dữ liệu giống như trang 71 và 72

## Bài 2: Đối tượng Command và DataReader

### Mục tiêu học tập

- Trình bày ứng dụng đối tượng Command và DataReader
- Tạo được đối tượng Command
- Trình bày các thuộc tính của đối tượng Command.
- Sử dụng các phương thức của câu lệnh Command trong thực thi câu truy vấn SQL

### I. Đối tượng Command

Lớp Command cho phép ta thực hiện bất cứ kiểu câu lệnh SQL nào. Mặc dù ta có thể dùng một lớp Command để thực hiện các khai báo dữ liệu (*như tạo và chỉnh sửa cơ sở dữ liệu, bảng hoặc chỉ mục*). Thông thường lập trình viên thường thực hiện các thao tác với cơ sở dữ liệu như: truy xuất và cập nhật các mẫu tin trong một table.

Lớp Command cài đặt giao tiếp IDBCommand. Giao tiếp này định nghĩa một vài thuộc tính chính và phương thức để thực hiện câu lệnh sau khi mở kết nối.

#### 1.1. Các kiểu câu lệnh Command

Trước khi sử dụng command, ta cần chọn kiểu câu lệnh, gán giá trị cho chuỗi câu lệnh và gắn câu lệnh vào một chuỗi kết nối nào đó. Để thực hiện điều này ta cần xác định thuộc tính CommandType, CommandText và Connection. Hoặc ta có thể truyền các tham số trên thông qua hàm khởi tạo của đối tượng Command.

Giá trị	Mô tả
CommandType.Text	Câu lệnh sẽ thực thi câu lệnh SQL trực tiếp. Câu lệnh SQL được cung cấp trong thuộc tính CommandText. Đây là giá trị mặc định
CommandType.StoredProcedure	Câu lệnh sẽ thực thi một thủ tục stored procedure trong nguồn dữ liệu. Thuộc tính CommandText cung cấp tên của thủ tục.
CommandType.TableDirect	Câu lệnh sẽ truy vấn toàn bộ mẫu tin trong . CommandText là tên của table mà câu lệnh sẽ truy xuất các mẫu tin.

**Ví dụ:** Để tạo một đối tượng Command để thực hiện một câu truy vấn “SELECT \* FROM Employees”

#### Cách 1: Sử dụng hàm khởi tạo không tham số

```
SqlCommand cmd = new SqlCommand();
cmd.Connection = con;
cmd.CommandType = CommandType.Text;
cmd.CommandText = "SELECT * FROM Employees";
```

#### Cách 2: Truyền tham số trong hàm khởi tạo đối tượng SqlCommand

```
SqlCommand cmd = new SqlCommand(
```

"SELECT \* FROM Employees", con);

### Cách 3: Đè sử dụng Store procedure

```
SqlCommand cmd = new SqlCommand("GetEmployees", con);
cmd.CommandType = CommandType.StoredProcedure;
```

## 1.2.Các phương thức của đối tượng Command

Tên phương thức	Mô tả
ExecuteNonQuery()	<ul style="list-style-type: none"><li>Thực thi câu lệnh non-SELECT, như là <b>insert, delete, or update</b> mẫu tin. Giá trị trả về cho biết số dòng bị ảnh hưởng bởi câu lệnh.</li><li>Thực thi câu lệnh <b>create, alter, or delete</b> các đối tượng cơ sở dữ liệu (<i>như tables, indexes, constraints, ...</i>).</li></ul>
ExecuteScalar()	<ul style="list-style-type: none"><li>Thực hiện câu truy vấn SELECT và trả về giá trị của field đầu tiên của dòng đầu tiên trong tập dữ liệu được trả về bởi câu lệnh.</li><li>Thường được dùng để thực thi các câu lệnh tổng hợp <b>SELECT</b> có sử dụng các hàm như <b>COUNT()</b> hoặc <b>SUM()</b>.</li></ul>
ExecuteReader()	<ul style="list-style-type: none"><li>Thực hiện câu truy vấn SELECT và trả về đối tượng DataReader</li></ul>

## II. Đối tượng DataReader

Lớp DataReader cho phép bạn đọc dữ liệu trả về bởi câu lệnh SELECT dưới dạng chỉ đọc (read-only). Sử dụng DataReader là cách đơn giản nhất để lấy dữ liệu từ cơ sở dữ liệu. Tuy nhiên nó hạn chế trong việc lưu trữ và xử lý dữ liệu kiểu không kết nối như DataSet. Một số phương thức của đối tượng DataReader được mô tả như bảng sau:

Phương thức	Mô tả
GetValue()	<ul style="list-style-type: none"><li>Trả về giá trị được lưu trữ trong trường dữ liệu với chỉ mục được chỉ định, trong phạm vi dòng được chọn.</li><li>Nếu truy xuất đến trường được chỉ định bởi index hoặc index không phù hợp, một ngoại lệ <b>IndexOutOfRangeException</b> sẽ được trả về.</li></ul>
GetValues()	<ul style="list-style-type: none"><li>Lưu trữ các giá trị của dòng hiện thời vào một mảng.</li><li>Số lượng trường được lưu phụ thuộc vào kích cỡ của mảng mà bạn truyền vào phương thức này.</li><li>Ta có thể sử dụng thuộc tính <b>DataReader.FieldCount</b> để xác định số lượng trường trong một dòng.</li></ul>
Read()	<ul style="list-style-type: none"><li>Chuyển con trỏ đến dòng kế tiếp trong luồng dữ liệu.</li><li>Phương thức này được gọi trước khi đọc dòng dữ liệu đầu tiên.</li><li>Trả về true nếu có một dòng nữa để đọc, hoặc false nếu nó đọc dòng cuối cùng.</li></ul>
GetInt32(), GetChar(), GetDateTime(), GetXxx():	<ul style="list-style-type: none"><li><b>GetInt32()</b>, <b>GetChar()</b>, <b>GetDateTime()</b>, <b>GetXxx()</b>: Những phương thức này trả về giá trị của trường với chỉ mục được chỉ định ở dòng hiện tại.</li><li>Lưu ý nếu bạn cố gắng gán giá trị trả về đến biến sai kiểu, một ngoại lệ <b>InvalidCastException</b> sẽ được trả về.</li><li>Các giá trị này không hỗ trợ kiểu NULL, nếu trường dữ liệu có chứa giá trị NULL, ta cần phải kiểm tra trước khi gọi các phương thức này.</li></ul>

NextResult()	- Nếu câu lệnh được sinh ra bởi DataReader được trả về có nhiều hơn một rowset, phương thức này chuyển con trỏ đến rowset kế tiếp.
Close()	- Đóng reader.

### III. Mô tả cơ sở dữ liệu

Cho cơ sở dữ liệu nhân viên **gồm 2 tables Nhanvien và Phongban** với cấu trúc như sau:

Column Name	Data Type
MaNhanvien	int
TenNhanvien	nvarchar(50)
Ngaysinh	datetime
Diachi	nchar(50)
Dienthoai	nchar(10)
Gioitinh	nchar(5)
Maphong	int
Hinhanh	image

Column Name	Data Type
Maphong	int
Tenphong	nvarchar(50)
Chucnang	nvarchar(50)
Trphong	int

Table Phongban

Table Nhanvien

Dữ liệu của Table Nhân viên

MaN...	TenNhanvien	Ngaysinh	Diachi	Dienthoai	Gioitinh	Maphong	Hinhanh
1	Nguyen Thi	1/1/1980 12:00:...	Cảng Long, Trà Vinh ...	0919800657	N	1	NULL
2	Võ Thị Thu Duyên	1/1/1974 12:00:...	Cảng Long, Trà Vinh ...	0123456789	Nữ	2	NULL
3	Lê Thành Nguyễn	1/1/2012 12:00:...	Cầu Ngang, Trà Vinh ...	0987987987	Nam	2	NULL
5	Nguyễn Ngọc Tha...	6/13/1983 12:00:...	Vũng Liêm, Vĩnh Long ...	0981112221	Nữ	2	NULL
6	Trần Thị Xuân Thúy	10/9/1983 12:00:...	Măng Thịt, Vĩnh Long ...	0981112222	Nữ	1	NULL
7	Cao Anh Thư	7/19/1986 12:00:...	Cầu Ngang, Trà Vinh ...	0981112223	Nữ	3	NULL
8	Nguyễn Song Toàn	1/1/1986 12:00:...	Trà Cú, Trà Vinh ...	0981112224	Nam	2	NULL
9	Võ Thị Thu Trang	2/2/1980 12:00:...	Trà Cú, Trà Vinh ...	0981112225	Nữ	1	NULL
10	Phạm Thị Cẩm Hồng	3/2/1976 12:00:...	Tam Bình, Vĩnh Long ...	0981112226	Nữ	3	NULL
12	Nguyễn Duy Kiệt	6/6/1982 12:00:...	Mỏ Cày, Bến Tre ...	0981112228	Nam	2	NULL
13	Trang Thị Minh	4/4/1983 12:00:...	Châu Thành, Trà Vinh ...	0981112229	Nữ	3	NULL
15	Huỳnh Thị Ngọc Quý	3/5/2011 12:00:...	Vũng Liêm, Vĩnh Long ...	0981112230	Nữ	1	NULL
16	Nguyễn Thanh Sang	5/3/2011 12:00:...	Châu Thành - Trà Vinh ...	0981112231	Nam	1	NULL
18	Oanh Thone	5/3/2011 12:00:...	Châu Thành - Trà Vinh ...	0917145587	Nữ	1	NULL
20	Vo Cong Duc	5/3/2011 12:00:...	Mang Thích - Vĩnh Long...	0917145587	Nam	3	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Dữ liệu của Table Phongban

Maphong	Tenphong	Chucnang	Trphong
1	Quản lý Nhân sự	Quản lý nhân sự	1
2	Thiết bị	Quản lý thiết bị	1
3	Hành chính	Quản trị hành chính	2
NULL	NULL	NULL	NULL

### IV. Một số ví dụ về Lớp Command và Data Reader

#### 4.1. Áp dụng Phương thức ExecuteNonQuery()

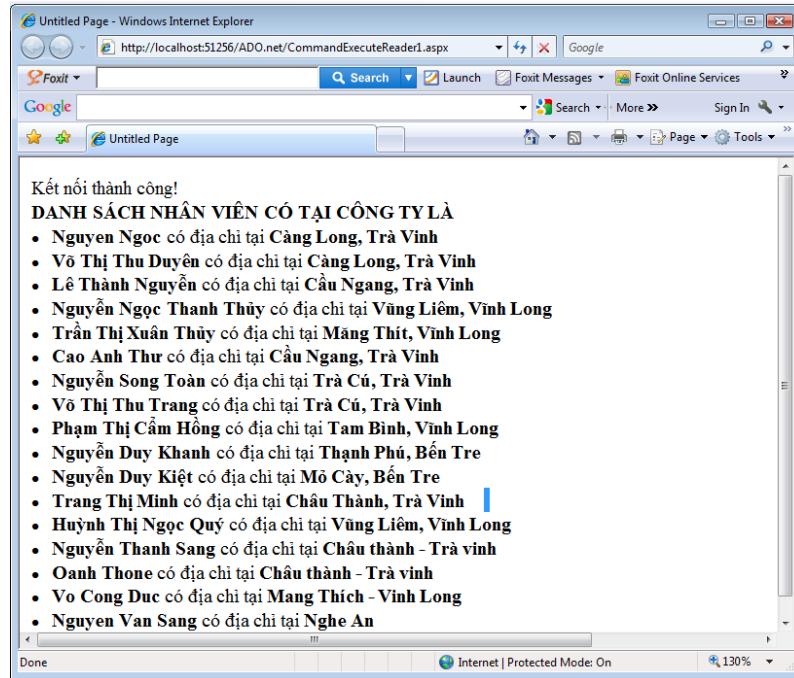
Đoạn mã lệnh sau đây được dùng để kết nối với cơ sở dữ liệu có tên là QLNV. Sau khi kết nối với CSDL, chương trình sẽ đọc dữ liệu trong table Nhanvien bằng cách sử dụng phương thức ExecuteNonQuery(). Bước kế tiếp là gọi thực thi phương thức hienthi để hiển thị kết quả trả về trong đối tượng SqlDataReader.

```

protected void Page_Load(object sender, EventArgs e)
{
 string connectionStr = "Data Source=localhost; Initial Catalog=QLNV; Integrated
Security=SSPI";
 SqlConnection con = new SqlConnection(connectionStr);
 SqlCommand cmd = new SqlCommand("Select * from nhanvien");
 cmd.CommandType = CommandType.Text;
 cmd.Connection = con;
 try
 {
 con.Open();//Mở kết nối với CSDL
 SqlDataReader reader = cmd.ExecuteReader();
 lbtB.Text = "Kết nối thành công!
";
 lbtB.Text += "DANH SÁCH NHÂN VIÊN CÓ TẠI CÔNG TY LÀ
</BR>";
 Hienthi(reader);
 }
 catch (SqlException ex)
 {
 lbtB.Text = " Kết nối thất bại! " + ex.ToString();
 }
 finally
 {
 con.Close();
 }
}
//Hiển thị dữ liệu kết quả dưới dạng Danh sách
protected void Hienthi(SqlDataReader reader)
{
 StringBuilder htmlStr = new StringBuilder("");
 while (reader.Read())
 {
 htmlStr.Append("");
 htmlStr.Append("" + reader["tenhanvien"] + "");
 htmlStr.Append(" có địa chỉ tại " + reader["Diachi"] + "
");
 }
 ltrketqua.Text = htmlStr.ToString();
}

```

**Kết quả của đoạn mã lệnh trên**



**Hình 30: Form Kết quả Truy vấn dữ liệu với câu lệnh SELECT**

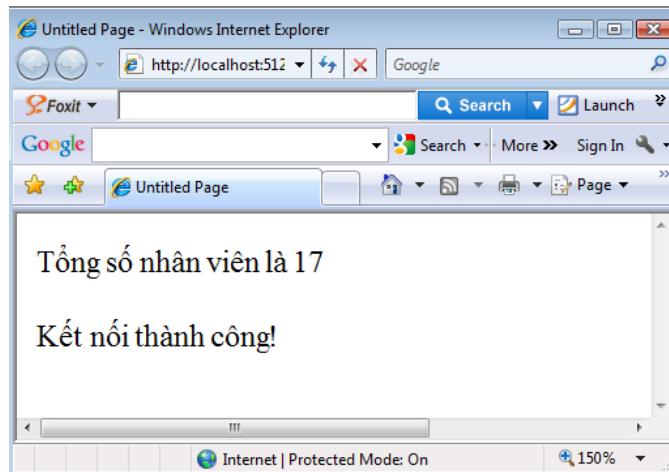
#### 4.2. Phương thức ExecuteScalar()

Đoạn mã lệnh sau dùng để kết nối với cơ sở dữ liệu có tên là QLNV. Sau khi kết nối với nguồn dữ liệu, chương trình sử dụng phương thức **ExecuteScalar()** để thực hiện câu lệnh SQL **SELECT count(\*) from Nhanvien**; đếm số lượng nhân viên hiện có trong bảng dữ liệu Nhanvien. Sau đó hiển thị dữ liệu kết quả trên trang web.

```
protected void Page_Load(object sender, EventArgs e)
{
 string connectionStr = "Data Source=localhost; Initial Catalog=QLNV;
Integrated Security=SSPI";
 SqlConnection con2 = new SqlConnection(connectionStr);
 SqlCommand cmd2 = new SqlCommand("SELECT count(*) from Nhanvien",
con2);

 try
 {
 con2.Open();
 lblTB.Text = "Kết nối thành công!";
 int sonv = (int)cmd2.ExecuteScalar();
 Response.Write("Tổng số nhân viên là: " + sonv.ToString());
 }
 catch (SqlException ex)
 {
 lblTB.Text = " Kết nối thất bại! " + ex.ToString();
 }
 finally
 {
 con2.Close();
 }
}
```

Kết quả của đoạn mã lệnh trên như sau:



**Hình 31: Form Kết quả thực hiện phương thức ExcecuteScalar**

#### 4.2. Phương thức ExecuteNonQuery()

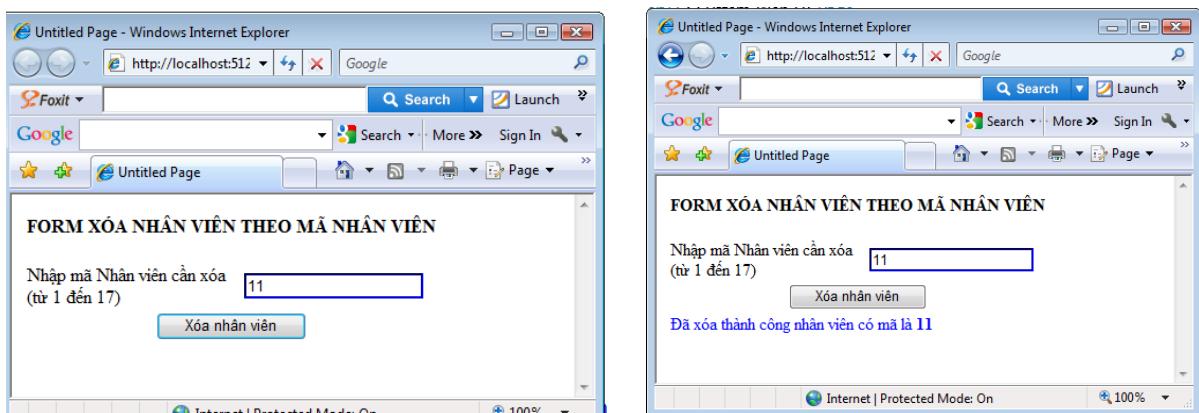
Đoạn mã lệnh sau dùng để kết nối với cơ sở dữ liệu có tên là QLNV. Sau khi kết nối với nguồn dữ liệu, chương trình sử dụng phương thức **ExecuteNonQuery()** để thực hiện câu lệnh SQL: **DELETE FROM Nhanvien WHERE MaNhanvien = <>Mã nhân viên>>**; để xóa nhân viên có mã là **<>Mã nhân viên>>** trong bảng dữ liệu Nhanvien.

Đoạn mã lệnh để thiết kế Form xóa nhân viên, sinh viên tự tìm hiểu.

```
protected void btnClick_Click(object sender, EventArgs e)
{
 string connectionStr = "Data Source=localhost; Initial Catalog=QLNV;
 integrated Security=SSPI";
 SqlConnection con = new SqlConnection(connectionStr);
 string sql = "DELETE FROM Nhanvien WHERE MaNhanvien ="
 + txtManv.Text;
 SqlCommand cmd = new SqlCommand(sql, con);
 cmd.CommandType = CommandType.Text;
 try
 {
 con.Open();
 cmd.ExecuteNonQuery();
 lblTB.Text += "Đã xóa thành công nhân viên có mã là " +
 txtManv.Text.ToString() + "";
 }
 catch (SqlException ex)
 {
 lblTB.Text += "Không có nhân viên với mã là: " +
 txtManv.Text.ToString();
 lblTB.Text += sql.ToString();
 }
}
```

Kết quả của xóa nhân viên có mã là 11 như sau:

**Hình 31: Form Kết quả thực hiện phương thức ExcecuteScalar**



**Hình 32: Form Kết quả thực hiện phương thức ExecuteNonQuery()**

### 4.3. Làm việc với thủ tục Stored Procedure

Các câu lệnh với tham số là bước ngắn nhất để tạo stored procedures. Thủ tục stored procedure là tập hợp một hoặc nhiều câu lệnh SQL được lưu trữ ở cơ sở dữ liệu. Stored procedures tương tự như các hàm được đóng gói, có thể nhận dữ liệu và trả kết quả thông qua cá tham số. Trong xây dựng các ứng dụng web. Sử dụng các thủ tục có nhiều điểm thuận lợi như:

Dễ bảo trì hệ thống: Ví dụ, bạn có thể tối ưu hóa các câu lệnh trong thủ tục mà không cần biên dịch ứng dụng liên quan. Bạn cũng có thể chuẩn hóa dữ liệu truy xuất tại chỗ. Các thủ tục cũng giúp ta xây dựng ứng dụng an toàn hơn. Ví dụ, ta có thể cho phép người dùng sử dụng tài khoản windows để thực thi các lệnh trong ASP.NET nhưng hạn chế truy xuất đến một số tables quan trọng. Thủ tục cũng làm tăng hiệu quả thực hiện chương trình. Ta có thể thực hiện một loạt các thao tác với cơ sở dữ liệu trong phạm vi một kết nối. Đặc biệt, thủ tục rất có ích trong việc truy xuất dữ liệu ở.

Các bước chính để gọi thực thi một thủ tục:

**Bước1: Tạo đối tượng SqlCommand và gán thuộc tính CommandText là InsertEmployee.**

- SqlCommand cmd = new SqlCommand("InsertEmployee", con);
- InsertEmployee là tên thủ tục ở HQTCSQL

**B2: Gán giá trị cho thuộc tính CommandType**

cmd.CommandType = CommandType.StoredProcedure

**B3: Gán giá trị cho các tham số**

- cmd.Parameters.Add(new SqlParameter(
  - "@TitleOfCourtesy", SqlDbType.NVarChar, 25));
- cmd.Parameters["@TitleOfCourtesy"].Value = title;

Ví dụ: Đoạn mã lệnh sau đây thực hiện gọi thủ tục xem thông tin mã nhân viên do người dùng nhập vào từ bàn phím.

a. Tạo thủ tục getNhanvien.

```
CREATE PROCEDURE getNhanvien
@MaNV int,
AS BEGIN
 SELECT * FROM Nhanvien WHERE Manhanvien = @MaNV
```

**END**

b. Đoạn mã lệnh gọi thủ tục getNhanvien và hiển thị dữ liệu kết quả dưới dạng danh sách.

```
protected void btnClick_Click(object sender, EventArgs e)
{
 string conStr = "Data Source=localhost; Initial Catalog=QLNV; Integrated
Security=SSPI";
 SqlConnection con = new SqlConnection(conStr);
 //Tạo đối tượng cmd gọi thực thi thủ tục getNhanvien
 SqlCommand cmd = new SqlCommand("getNhanvien", con);

 cmd.CommandType = CommandType.StoredProcedure;
 //Khai báo tham số tên là @MaNV kiểu Int
 cmd.Parameters.Add(new SqlParameter("@MaNV", SqlDbType.Int));
 //Gán giá trị cho tham số @MaNV
 cmd.Parameters["@MaNV"].Value = txtManv.Text.ToString();

 try
 {
 con.Open();
 SqlDataReader reader = cmd.ExecuteReader();
 Hienthi(reader);
 }
 catch (SqlException ex)
 {
 lblTB.Text = " Ket noi that bai!" + ex.ToString();
 }
 finally
 {
 con.Close();
 }
}
protected void Hienthi(SqlDataReader reader)
{
 StringBuilder str = new StringBuilder("");
 str.Append("THÔNG TIN NHÂN VIÊN");
 while (reader.Read())
 {
 str.Append("Mã Nhân viên: " + reader["Manhanvien"]);
 str.Append("Tên nhân viên: " + reader["Tennhanvien"]);
 str.Append("Ngày sinh: " + reader["Ngaysinh"]);
 str.Append("Địa chỉ: " + reader["Diachi"]);
 str.Append("Điện thoại: " + reader["Dienthoai"]);
 str.Append("Mã phòng: " + reader["Maphong"]);

 }
 lblTB.Text = str.ToString();
}
```

The screenshot shows a Windows Internet Explorer window with the title "Untitled Page - Windows Internet Explorer". The address bar displays "http://localhost:51256/ADO.net/StoreP". The page content is titled "FORM XEM THÔNG TIN NHÂN VIÊN THEO MÃ NHÂN VIÊN". It contains a text input field labeled "Nhập mã Nhân viên cần xem (từ 1 đến 17)" with the value "12" entered. Below this is a blue "Xem thông tin" button. To the right, under the heading "THÔNG TIN NHÂN VIÊN", there is a bulleted list of employee details:

- Mã Nhân viên: 12
- Tên nhân viên: Nguyễn Duy Kiệt
- Ngày sinh: 6/6/1982 12:00:00 AM
- Địa chỉ: Mỏ Cày, Bến Tre
- Điện thoại: 0981112228
- Mã phòng: 2

The status bar at the bottom of the browser window shows "Internet | Protected Mode: On" and "120%".

*Hình 33: Form Kết quả ví dụ trang 82*

### Câu hỏi/Bài tập củng cố

1. Trình bày ứng dụng đối tượng Command và DataReader.
2. Hãy nêu một số thuộc tính và phương thức thường sử dụng của đối tượng Command.
3. Hãy nêu một số thuộc tính và phương thức thường sử dụng của đối tượng Reader.
4. Tạo lại các form ở trang 77, 79, 81 bằng cách viết code
5. Thực hành bài Quản lý bài hát trang số 102

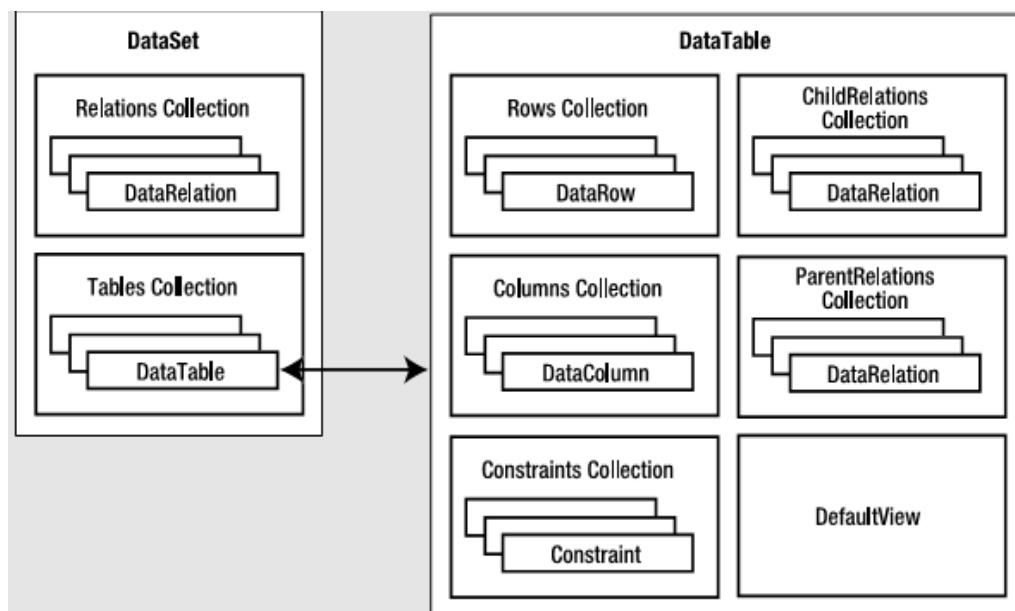
# Bài 3: Lớp DataSet và Data Adapter

## Mục tiêu học tập

- Trình bày kiến trúc của lớp DataSet
- Sử dụng lớp Data Adapter để đổ dữ liệu vào DataSet

### I. Lớp DataSet là gì?

Trung tâm của truy xuất dữ liệu kiểu không kết nối. Mỗi DataSet chứa 2 thành phần quan trọng: Một tập Tables (*được thể hiện thông qua thuộc tính Tables*) và một tập Relations dùng để liên kết các tables lại với nhau.



Hình 34: Các lớp của tập dữ liệu DataSet

### Một số lợi ích khi sử dụng DataSet so với DataReader.

Khi ứng dụng được kết nối với cơ sở dữ liệu, nó sẽ thực hiện tạo một bản sao của dữ liệu vào DataSet. Vì vậy, nếu ta thay đổi thông tin trong DataSet, thông tin ở dưới CSDL không hề bị ảnh hưởng. Một số trường hợp sau đây ta nên sử dụng DataSet.

- Khi ta cần truy xuất dữ liệu một cách ngẫu nhiên trong tập dữ liệu. DataSet có thể giúp ta truy xuất các mẫu tin sau hoặc trước mẫu tin nào đó. Điều này thì ta không thể thực hiện được với DataReader.
- Khi ta cần truy xuất dữ liệu từ nhiều tables khác nhau
- Có thể sử dụng tốt khi muốn thực hiện chức năng DataBinding
- Khi ta cần thao tác dữ liệu với XML.

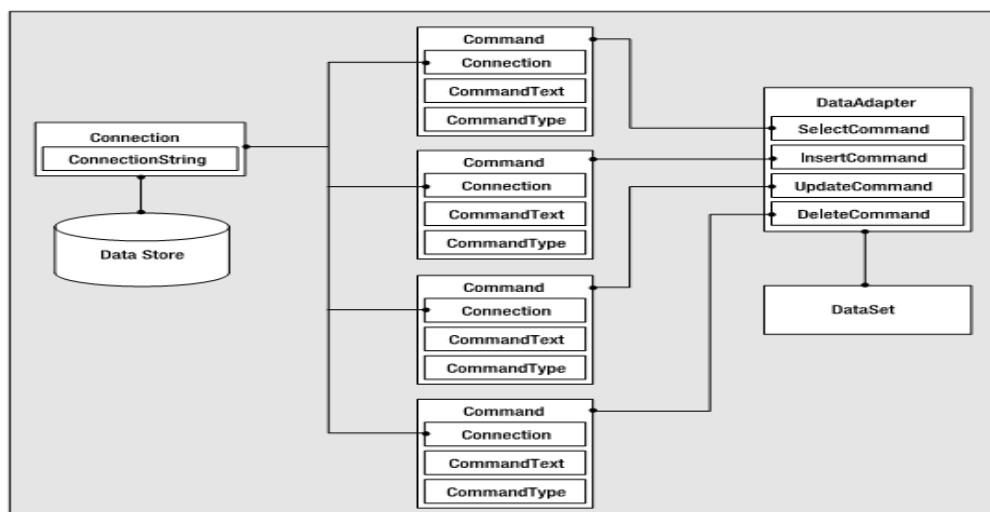
### II. Lớp Data Adapter

- Lớp DataAdapter như là cầu nối giữa DataTable trong DataSet và nguồn dữ liệu. Nó chứa các câu lệnh có sẵn để truy xuất và cập nhật nguồn dữ liệu.
  - Để DataAdapter có thể thực hiện các chức năng như edit, delete, và insert mẫu tin, cần xác định đối tượng Command cụ thể gồm các thuộc tính UpdateCommand, DeleteCommand, InsertCommand của DataAdapter.

- Sử dụng DataAdapter để fill một DataSet, bạn cần phải gán giá trị cho thuộc tính SelectCommand.

Các phương thức của lớp DataAdapter

Phương thức	Mô tả
Fill()	Thêm một DataTable vào một DataSet bằng cách thực thi câu truy vấn trong câu lệnh SelectCommand. Nếu câu truy vấn trả về kết quả là nhiều tập dữ liệu, phương thức này sẽ thêm nhiều đối tượng DataTable cùng một lúc.
FillSchema()	Thêm một DataTable vào DataSet bằng cách thực thi câu truy vấn trong câu lệnh SelectCommand và truy xuất lược đồ thông tin. Phương thức này không thể thêm bất kỳ dữ liệu nào vào DataTable. Thay vào đó, nó chỉ thực hiện cấu hình lại DataTable với thông tin chi tiết như tên cột, kiểu dữ liệu, khóa chính, và các ràng buộc dữ liệu
Update()	Kiểm tra tất cả các thay đổi trong DataTable và thực hiện tất cả các thay đổi đến nguồn dữ liệu bằng cách thực thi các câu lệnh phù hợp trong InsertCommand, UpdateCommand, và DeleteCommand.



**Hình 35: Mô hình Lớp DataAdapter tương tác với nguồn dữ liệu**

### III. Ví dụ:

Sử dụng DataAdapter để làm cầu nối với nguồn dữ liệu QLNV. Sau khi thực hiện câu lệnh truy vấn `select * from nhanvien` và `Select * from phongban`, tập DataSet sẽ có 2 DataTable là NV và PB tương ứng. Cuối cùng, hiển thị kết quả dưới dạng Literal như kết quả ở hình sau:

```

protected void Page_Load(object sender, EventArgs e)
{
 string connStr= "Data Source=localhost; Initial Catalog=QLNV;
 Integrated Security=SSPI";
 SqlConnection conn = new SqlConnection(connStr);
 string sql = "select * from nhanvien";
 SqlDataAdapter da = new SqlDataAdapter(sql, conn);
 DataSet ds = new DataSet();
 da.Fill(ds, "NV");

```

```

ltr_nhanvien.Text = hienthi(ds.Tables["NV"]).ToString();
da.SelectCommand.CommandText = "Select * from phongban";
da.Fill(ds, "PB");

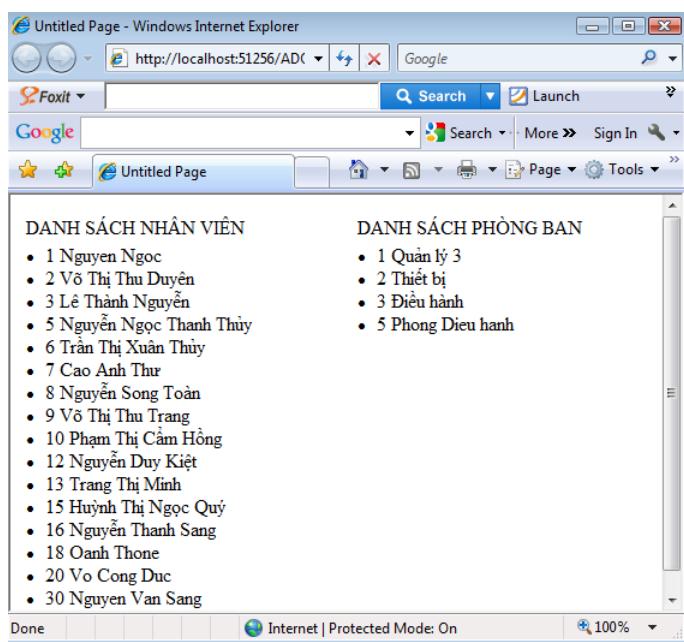
ltr_Phongban.Text = hienthi(ds.Tables["PB"]).ToString();

}

//Hien thi duoi dang danh sach
StringBuilder hienthi(DataTable dt) {
 StringBuilder str = new StringBuilder();
 foreach (DataRow dr in dt.Rows)
 str.Append(" " + dr[0] + " " + dr[1]);
 return str;
}

```

Kết quả của đoạn mã lệnh trên như sau:



*Hình 36: Form Kết quả ví dụ trang 85*

### Câu hỏi/Bài tập củng cố

1. Trình bày kiến trúc của lớp DataSet
2. Nêu một số thuộc tính và phương thức của lớp.
3. Thực hành bài tập Quản lý Album trong trang 102

# Bài 4: Data Binding

## Mục tiêu học tập:

- Sử dụng DataBinding để liên kết dữ liệu tập hợp với web controls
- Liên kết nguồn dữ liệu trong DataReader với web controls

## I. Giới thiệu

Hầu hết các ứng dụng web phải làm việc với cơ sở dữ liệu trong cơ sở dữ liệu quan hệ hoặc các tập tin XML. Việc truy xuất dữ liệu chỉ là một phần việc trong việc xây dựng ứng dụng web. Một ứng dụng web hiện đại cần phải thể hiện tính tiện ích, linh hoạt và hấp dẫn để hiển thị dữ liệu trên trang web. ASP.net bao gồm các mô hình có đầy đủ chức năng của Data-binding. Data Binding cho phép lập trình viên thực hiện kết nối đối tượng dữ liệu được truy xuất với một hoặc vài web control, sau đó sẽ tự động hiển thị dữ liệu. Điều này có nghĩa là các lập trình viên không cần phải viết các đoạn code phức tạp để duyệt qua các dòng dữ liệu, đọc dữ liệu từ nhiều cột khác nhau và thao tác với từng controls riêng biệt.

Để mọi việc được dễ dàng hơn, ta có thể sử dụng các điều khiển nguồn dữ liệu. Một điều khiển nguồn dữ liệu cho phép bạn định nghĩa liên kết rõ ràng giữa trang web của bạn và nguồn dữ liệu. Điều khiển nguồn dữ liệu là một công cụ tốt để thực hiện kết buộc dữ liệu. Sau khi cấu hình nguồn dữ liệu xong, ta có thể sử dụng nó trong lúc thiết kế. Thật vậy, bằng cách sử dụng điều khiển nguồn dữ liệu, bạn có thể tạo ra các trang web phức tạp cho phép bạn truy vấn và cập nhật dữ liệu mà không cần viết từng dòng lệnh.

## II. Cơ bản về Data Binding

Data binding là chức năng cho phép bạn thực hiện kết nối dữ liệu giữa nguồn dữ liệu với web control và có thể tự động hiển thị dữ liệu của bạn. Đặc điểm chính của Data Binding là tính rõ ràng, không mang tính lập trình. Điều này có nghĩa là Data binding được định nghĩa bên ngoài code của bạn cùng với các điều khiển trong trang .aspx.

### 2.1. Single Value Binding

Trong Asp.net, hầu hết các điều khiển web (bao gồm TextBox, LinkButton, Image,...) đều hỗ trợ single-value binding. Với Single-value binding, ta có thể kết buộc một thuộc tính của điều khiển với nguồn dữ liệu, nhưng điều khiển có thể hiển thị chỉ một giá trị đơn. Thuộc tính mà bạn có thể kết buộc không cần hiển thị trực tiếp trên trang. Ví dụ, bạn có thể liên kết thuộc tính Text trong thẻ Hyperlink. Ta có thể sử dụng thuộc tính Text, hoặc có thể sử dụng thuộc tính NavigateUrl để xác định liên kết của trang web.

**Cú pháp:** <%# Biểu thức kết buộc %>

**Ví dụ:** <%# EmployeeName %> để liên kết với trường thông tin EmployeeName

```
<html xmlns="http://www.w3.org/1999/xhtml">
<body>
<form method="post" runat="server">
 <asp:Image ID="image1" runat="server" ImageUrl='<%# FilePath %>' />

 <asp:Label ID="label1" runat="server" Text='<%# FilePath %>' />


```

```

<asp:TextBox ID="textBox1" runat="server" Text='<%# GetFilePath() %>' />

<asp:HyperLink ID="hyperLink1" runat="server"
 NavigateUrl='<%# LogoPath.Value %>' Font-Bold="True" Text="Show logo" />

<input type="hidden" ID="LogoPath" runat="server" value="apress.gif">
<%# FilePath %>

protected string GetFilePath()
{
 return "apress.gif";
}

protected string FilePath
{
 get { return "apress.gif"; }
}

protected void Page_Load(object sender, System.EventArgs e)
{
 this.DataBind();
}

```

## 2.2. Liên kết dữ liệu có lặp (Repeated Value Binding)

Nhiều web controls có hỗ trợ Repeated value binding có nghĩa là chúng sẽ trả về một tập các mục chọn. Các Repeated Value control thường tạo danh sách hoặc các lưới (Listbox hoặc Gridview). Nếu một điều khiển có hỗ trợ dạng này phải có thuộc tính DataSource để chấp nhận đối tượng dữ liệu. Khi gán thuộc tính DataSource, ta đã thực hiện một liên kết từ server control đến đối tượng dữ liệu cần hiển thị. Tuy nhiên nó không tự động thực hiện, mà ta phải gọi phương thức DataBind(). Phương thức này sẽ duyệt qua toàn bộ nguồn dữ liệu, truy xuất dữ liệu và trả về dữ liệu dưới dạng trang web.

Repeated-value binding cho phép bạn kết nối toàn bộ danh sách thông tin đến một control. Danh sách thông tin được trình bày bởi đối tượng dữ liệu có thể đóng gói trong một tập đối tượng có thể là tập hợp các custom objects (vd: trong một ArrayList hoặc hashtable) hoặc một tập các mẫu tin (vd, DataReader hoặc DataSet). ASP.NET bao gồm một vài list controls có hỗ trợ repeated-value binding:

Cho phép kết buộc toàn bộ danh sách thông tin đến một trong các controls

- Tất cả các control trả về có sử dụng thẻ <select> bao gồm HtmlSelect, ListBox, và DropDownList
- CheckBoxList, RadioButtonList trả về các mục chọn con ở check box hoặc radio button tách rời.
- BulletedList control, tạo danh sách các bullet hoặc number. Tất cả các control này hiển thị trường giá trị riêng của thuộc tính từ mỗi mục chọn dữ liệu.

- Khi thực hiện liên kết dữ liệu với các controls này, ta sẽ sử dụng các thuộc tính được liệt kê sau:

Thuộc tính	Mô tả
<b>DataSource</b>	Đối tượng dữ liệu chứa tập các mục chọn dữ liệu cần hiển thị. Đối tượng dữ liệu này phải cài đặt một trong những giao tiếp mà ASP.NET data binding hỗ trợ, thường là ICollection.
<b>DataSourceID</b>	Liên kết list control với data source control bằng cách đặt giá trị cho thuộc tính này. Data source control sẽ tạo ra đối tượng dữ liệu yêu cầu tự động. Ta có thể dùng thuộc tính <b>DataSource</b> hoặc <b>DataSourceID</b> , nhưng không phải cả hai.
<b>DataTextField</b>	Một list control có thể hiển thị duy nhất một giá trị từ danh sách mục chọn. DataTextField chỉ định field hoặc property của mục chọn dữ liệu chứa giá trị hiển thị trong trang
<b>DataTextFormatString:</b>	Định dạng dữ liệu cần hiển thị trong <b>DataTextValue</b> : Ví dụ: ta có thể chỉ định một số phải được định dạng dưới dạng kiểu tiền tệ currency.
<b>DataValueField</b>	Giá trị từ mục chọn dữ liệu không hiển thị trong trang. Được lưu trữ trong thuộc tính <b>value</b> của thẻ HTML Thường dùng để lưu trữ các trường <b>ID</b> hoặc <b>khóa chính</b> Có thể sử dụng nó để truy xuất nhiều dữ liệu khi người dùng chọn một mục chọn cụ thể nào đó.

### Ví dụ:

Tạo một số các list controls bao gồm: listbox, dropdownlist, checkboxlist và radiobuttonlist ở trang ListBinding.aspx. Ở code behinde ListBiding.aspx.cs, thực hiện tạo nguồn dữ liệu để liên kết với các list controls được nêu ở trên.

Đoạn script ở trang ListBinding.aspx như sau:

```

<form id="form1" runat="server">

Sử dụng listbox để hiển thị
 <asp:ListBox id="MyListBox" runat="server" Width="169px" ></asp:ListBox>

Sử dụng HTML control DropDownList để hiển thị

<select id="MyHTMLSelect" size="1" runat="server" />

Sử dụng Web control DropDownList để hiển thị

<asp:DropDownList id="MyDropDownListBox" runat="server"
 Width="200" ></asp:DropDownList>

Sử dụng CheckBoxList để hiển thị
 <asp:CheckBoxList id="MyCheckBoxList" runat="server"
 RepeatColumns=3 RepeatDirection=Horizontal ></asp:CheckBoxList>

Sử dụng RadioButtonList để hiển thị
 <asp:RadioButtonList id="MyRadioButtonList" runat="server"
 RepeatColumns=3 RepeatDirection=Horizontal></asp:RadioButtonList>
</form>

```

Đoạn mã lệnh ở trang ListBinding.aspx.cs

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

```

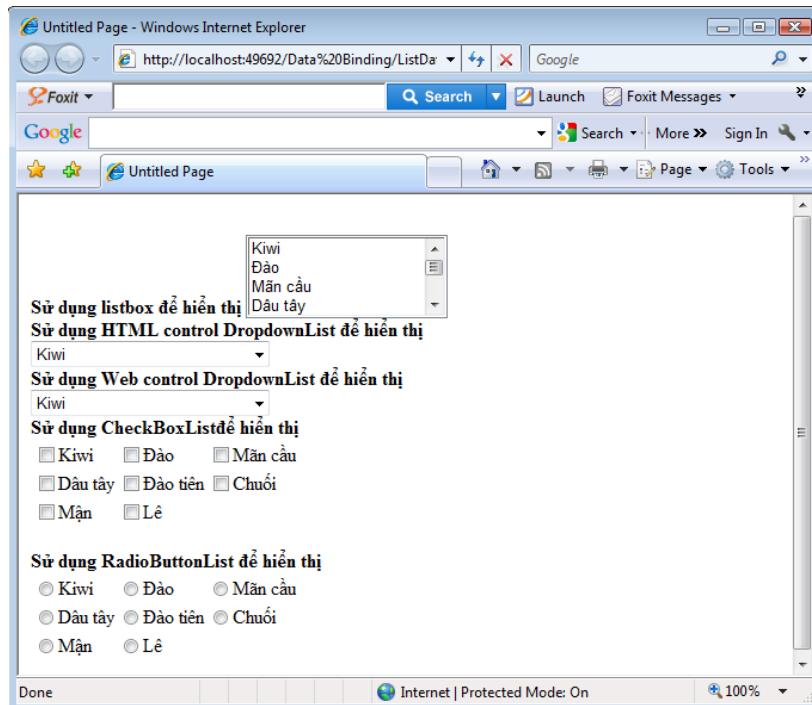
{
 // Tạo và fill danh sách
 List<string> fruit = new List<string>();
 fruit.Add("Kiwi");
 fruit.Add("Đào");
 fruit.Add("Mân cầu");
 fruit.Add("Dâu tây");
 fruit.Add("Đào tiên");
 fruit.Add("Chuối");
 fruit.Add("Mận");
 fruit.Add("Lê");

 // Định nghĩa một kết nối với list controls.
 MyListBox.DataSource = fruit;
 MyDropDownListBox.DataSource = fruit;
 MyHTMLSelect.DataSource = fruit;
 MyCheckBoxList.DataSource = fruit;
 MyRadioButtonList.DataSource = fruit;

 // Thực hiện liên kết
 this.DataBind();
}

```

Kết quả của đoạn chương trình trên được minh họa như hình.



*Hình 37: Form Kết quả ví dụ trang 89*

### 2.3. Liên kết dữ liệu với DataReader

- Liệt kê tên của các nhân viên trong table Nhanvien đưa vào listbox như hình bên
- Cho phép người dùng click chọn một vài nhân viên. Sau đó click vào nút lệnh “Chọn nhân viên” sẽ hiện thị tên của các nhân viên được chọn ở phía dưới.
- Khai báo trong tập tin Binding to DataReader.aspx như sau. Nguồn dữ liệu cho listbox lb\_dsNhanvien được viết trong code \_behind.

```
<form id="form1" runat="server">
```

```

<asp:ListBox ID="lb_dsNhanvien" runat="server" SelectionMode=Multiple
Rows=10 ForeColor=Blue DataTextField="Tennhanvien"
DataValueField="Manhanvien">
</asp:ListBox>
<asp:Button ID="btnChon" runat="server" Text=" Chọn nhân viên "
onclick="btnChon_Click" />
Danh sách nhân viên được chọn là:
<asp:Label ID="lblTB" runat=server></asp:Label>
</form>

```

- Cài đặt trong code behind và hàm xử lý sự kiện khi click nút lệnh Chọn nhân viên như sau:

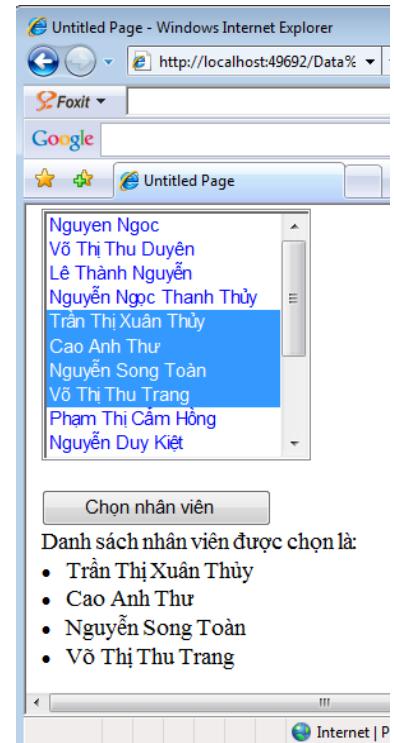
```

protected void Page_Load(object sender, EventArgs e)
{
 if (!Page.IsPostBack)
 {
 string conStr = "Data source=localhost; Initial Catalog=QLNV;
Integrated Security=SSPI";
 SqlConnection con = new SqlConnection(conStr);
 SqlCommand cmd = new SqlCommand("Select * from Nhanvien", con);
 try
 {
 con.Open();
 SqlDataReader reader = cmd.ExecuteReader();
 ddl_dsNhanvien.DataSource = reader;
 ddl_dsNhanvien.DataBind();
 }
 finally
 {
 con.Close();
 }
 }
}

protected void btnChon_Click(object sender,
EventArgs e)
{
 foreach (ListItem li in ddl_dsNhanvien.Items)
 if (li.Selected)
 lblTB.Text += "" + li.Text.ToString();
}

```

- Kết quả của các cài đặt trên được hiển thị như hình theo sau:



**Hình 38: Form Kết quả ví dụ trang 91**

## 2.4. Chu kỳ sống của trang web có sử dụng Data Binding

- Đối tượng trang được tạo (dựa trên file .aspx).
- Chu kỳ sống của trang bắt đầu, các sự kiện Page.Init và Page.Load được kích hoạt

- Tất cả các sự kiện control khác được kích hoạt.
- Điều khiển nguồn dữ liệu có thể thực hiện bất kỳ cập nhật dữ liệu nào. Nếu một dòng dữ liệu đang được cập nhật, sự kiện Updating và Updated được kích hoạt. Nếu một dòng đang được insert, các sự kiện Inserting và Inserted kích hoạt. Nếu một dòng đang được deleted, các sự kiện Deleting và Deleted kích hoạt.
- Sự kiện Page.PreRender được kích hoạt.
- Data source controls thực hiện bất kỳ một truy vấn và insert dữ liệu đến controls liên quan. Sự kiện Selecting và Selected được kích hoạt.
- Đối tượng trang được trả về và giải phóng tài nguyên ở server.

### Câu hỏi/Bài tập củng cố:

1. Nếu các thuộc tính chính của lớp List Control để sử dụng trong Liên kết dữ liệu (Binding)
2. Thực hiện tạo lại Form trang 88
3. Thực hiện tạo lại Form trang 90
4. Thực hành bài tập Quản lý Album ở trang 102

# BÀI 5: ĐIỀU KHIỂN NGUỒN DỮ LIỆU

## Mục tiêu của bài học:

- Tạo điều khiển nguồn dữ liệu SqlDataSource
- Liên kết nguồn dữ liệu với các web controls
- Thực hiện các thao tác với CSDL trên nguồn dữ liệu

### I. Các loại điều khiển nguồn dữ liệu

Điều khiển nguồn dữ liệu bao gồm bất cứ các điều khiển nào có cài đặt giao tiếp **IDataSource**. .NET Framework bao gồm các data source controls sau đây:

- **SqlDataSource:** Nguồn dữ liệu này cho phép bạn kết nối đến bất kỳ nguồn dữ liệu nào có nhà cung cấp dữ liệu ADO.NET. Nó bao gồm SQL Server, Oracle, và OLE DB hoặc nguồn dữ liệu. Khi sử dụng nguồn dữ liệu, ta không cần phải viết mã lệnh truy xuất dữ liệu.
- **ObjectDataSource:** Nguồn dữ liệu này cho phép ta kết nối đến các lớp truy xuất dữ liệu. Đây là mô hình được ưa chuộng đối với các ứng dụng web lớn.
- **AccessDataSource:** Nguồn dữ liệu này cho phép ta đọc và ghi dữ liệu lên các tập tin cơ sở dữ liệu Access (.mdb).
- **XmlDataSource:** Nguồn dữ liệu này cho phép ta kết nối với tập tin XML.
- **SiteMapDataSource:** Nguồn dữ liệu này cho phép ta kết nối với tập tin Web.sitemap để mô tả cấu trúc di chuyển của website.

Trong phạm vi của tài liệu này, ta chỉ trình bày nội dung chi tiết về điều khiển nguồn dữ liệu SqlDataSource. Các điều khiển nguồn dữ liệu còn lại, sinh viên có thể nghiên cứu thêm trong các tài liệu tham khảo được giới thiệu ở phần sau:

### II. Điều khiển SqlDataSource:

Điều khiển nguồn dữ liệu được viết trong tập tin .aspx của trang web giống như các điều khiển bình thường khác.

Ví dụ:

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server" ... />
```

SqlDataSource trình bày một kết nối với cơ sở dữ liệu có sử dụng nhà cung cấp dữ liệu ADO.net. Tuy nhiên, SqlDataSource cần một cách chung để tạo các Connection, Command và đối tượng DataReader. SqlDataSource cần cung cấp một Data Provider để truy xuất vào nguồn dữ liệu. Trong .Net chúng ta có 4 nhà cung cấp như sau:

- System.Data.SqlClient
- System.Data.OracleClient
- System.Data.OleDb
- System.Data.Odbc

Xác định chuỗi kết nối trong tập tin Web.Config

```
<configuration>
 <connectionStrings>
 <add name="qlnvstr"
 connectionString="Data Source=localhost;Initial Catalog=QLNV;">
```

```

Integrated Security=SSPI"/>
</connectionStrings>
...
</configuration>
<asp:SqlDataSource ConnectionString="<%$ ConnectionStrings:qlnvstr %>" ... />

```

## 2.1. Truy xuất các mẫu tin:

Ta có thể sử dụng một điều khiển SqlDataSource để truy xuất một câu truy vấn riêng lẻ. Tuy nhiên, ta có thể thêm các câu lệnh để thực hiện xóa, thêm và cập nhật một mẫu tin.

Câu lệnh SqlDataSource được cung cấp thông qua 4 thuộc tính: SelectCommand, InsertCommand, UpdateCommand, và DeleteCommand. Các thuộc tính này nhận giá trị là một câu lệnh SQL. Trong trường hợp này các thuộc tính liên quan như SelectCommandType, InsertCommandType, UpdateCommandType, và DeleteCommandType có giá trị là Text. Nếu ứng dụng chỉ cần thực hiện truy xuất mẫu tin thôi thì ta chỉ cần dùng thuộc tính SelectCommand.

Ví dụ: Để tạo một nguồn dữ liệu SqlDataSource để định nghĩa câu lệnh SELECT để truy xuất các mẫu tin từ table Employees:

```

<asp:SqlDataSource ID="sourceNhanvien" runat="server"
 ProviderName="System.Data.SqlClient"
 ConnectionString="<%$ ConnectionStrings:qlnvstr %>"
 SelectCommand=
 "SELECT Manhanvien, TenNhanvien, Diachi FROM Nhanvien"/>

```

## 2.2. Câu lệnh có tham số

### Các kiểu tham số có thể sử dụng

Giá trị của tham số không cần thiết phải lấy từ các điều khiển khác. Ta có thể kết nối một tham số với bất kỳ kiểu dữ liệu tham số nào trong bảng sau đây:

Nguồn	Điều khiển	Mô tả
Control	<asp:ControlParameter>	Giá trị được lấy từ các điều khiển khác trong trang web.
Querystring	<asp:QueryStringParameter>	Giá trị từ chuỗi truy vấn
Session	<asp:SessionParameter>	Giá trị được lưu trữ trong phiên làm việc(session) của người dùng.
Cookie	<asp:CookieParameter>	Giá trị được lấy từ cookies được gắn vào yêu cầu hiện tại.
Profile	<asp:ProfileParameter>	Giá trị được lấy từ thông tin hồ sơ hiện tại của người dùng.

Trong ví dụ ở phần 2.1 ở trên, ta có thể xem tất cả các nhân viên trong công ty. Nếu số nhân viên trong công ty quá nhiều thì việc lọc ra rất mất thời gian. Để truy xuất một tập con của dữ liệu, chẳng hạn như ta cần xem danh sách các nhân viên trong một phòng ban nào đó. Ta có thể sử dụng tham số trong câu truy vấn như sau:

Ví dụ:

- Khai báo nguồn dữ liệu để cho biết tất cả các phòng ban hiện có trong công ty

```

<asp:SqlDataSource ID="sds_Phongban" runat="server"
 ConnectionString="<%$ConnectionString:QLNV %>"
 ProviderName="System.Data.SqlClient"
 SelectCommand="Select * from Phongban">
</asp:SqlDataSource>

```

- Sử dụng drop-down list có tên lstPhong để hiển thị dữ liệu từ nguồn dữ liệu sourcePhongban

```

<asp:DropDownList ID="ddl_dsPhong" runat="server"
 DataSourceID="sourcePhongban"
 DataTextField="Tenphong"
 DataValueField="Maphong"
 AutoPostBack="true">
</asp:DropDownList>

```

- Điều khiển dropdownlist có thuộc tính AutopostBack = true để trang web được tự động chuyển về server mỗi khi người sử dụng thay đổi chọn lựa. Cho phép trang web của ta thực hiện cập nhật thông tin danh sách nhân viên tương ứng với kết quả chọn lựa phòng ban từ dropdownlist.

- Khi một phòng ban được chọn, nguồn dữ liệu thứ hai sẽ truy xuất tất cả nhân viên thuộc phòng ban.

```

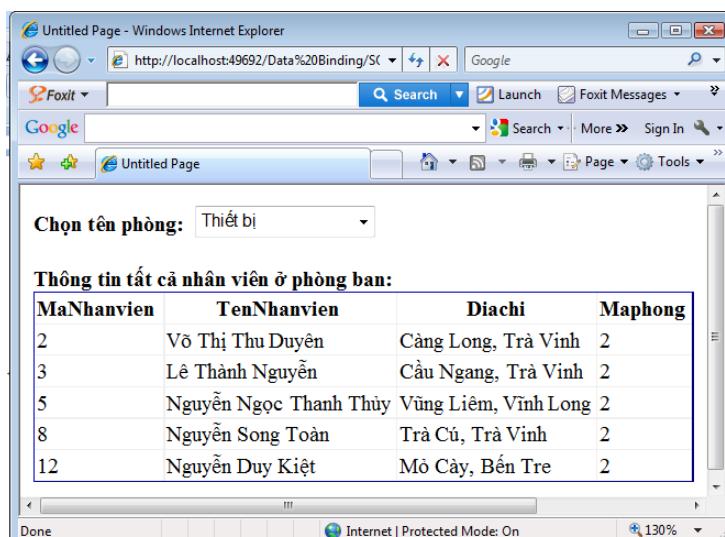
<asp:SqlDataSource ID="sds_Nhanvien" runat="server"
 ConnectionString="<%$ConnectionString:QLNV %>"
 SelectCommand="Select MaNhanvien, TenNhanvien, Diachi, Maphong from
 Nhanvien where Maphong=@Maphong"
 ProviderName="System.Data.SqlClient">

 <SelectParameters>
 <asp:ControlParameter ControlID="ddl_dsPhong"
 Name="Maphong"
 PropertyName="SelectedValue"/>
 </SelectParameters>

</asp:SqlDataSource>

```

- Kết quả của đoạn chương trình trên được hiển thị như sau:



**Hình 39: Form Kết quả ví dụ trang 95**

### 2.3. Cập nhật dữ liệu

Bên cạnh việc chọn dữ liệu chỉ là một phần việc trong SqlDataSource, SqlDataSource có thể áp dụng để cập nhật dữ liệu. Tuy nhiên, không phải tất cả các điều khiển hỗ trợ cập nhật. Ví dụ, Listbox không cho phép người dùng cập nhật giá trị, xóa một mục chọn nào đó hoặc thêm một mục chọn mới. Tuy nhiên các rich data controls như GridView, DetailsView, và FormView cho phép người dùng thực hiện các chức năng cập nhật dữ liệu trên.

Xác định các câu lệnh đối với thao tác cập nhật dữ liệu cần thực hiện như thêm (InsertCommand), xóa (DeleteCommand), và cập nhật (UpdateCommand). Việc định nghĩa các câu lệnh trên giống như ta đã thực hiện trong phần Chọn mẫu tin.

Ví dụ:

```
<asp:SqlDataSource ID="sds_Nhanvien" runat="server"
 ProviderName="System.Data.SqlClient"
 ConnectionString="<%$ConnectionString:QLNV %>"
 SelectCommand="Select * from Nhanvien"
 UpdateCommand="Update Nhanvien
 Set Diachi=@Diachi,
 Tennhanvien=@Tennhanvien
 where Manhanvien=@MaNhanvien">
</asp:SqlDataSource>
```

Đoạn mã lệnh trên được dùng để tạo nguồn dữ liệu cho phép cập nhật 2 thông tin Tennhanvien và Diachi của nhân viên.

Trong ví dụ này, tham số tên không cần chọn lựa. Nếu trường dữ liệu trong cơ sở dữ liệu có tên là **Tennhanvien** thì tham số mặc định sẽ là **@Tennhanvien**. Đây là một hỗ trợ thuận lợi trong việc chọn lựa tên cho tham số.

### 2.4. Xóa mẫu tin

Xóa mẫu tin tương tự như cập nhật mẫu tin. Ta cần phải tạo câu lệnh DeleteCommand để xóa mẫu tin cần xóa. Ví dụ sau dùng để xóa thông tin của nhân viên với mã nhân viên được chỉ định qua tham số @Manhanvien

```
<asp:SqlDataSource ID="sourceEmployees" runat="server"
 ProviderName="System.Data.SqlClient"
 ConnectionString="<%$ConnectionString:Northwind %>"
 SelectCommand=
 "SELECT * FROM Nhanvien"
 DeleteCommand="DELETE Nhanvien WHERE Manhanvien=@Manhanvien ">
</asp:SqlDataSource>
```

- Việc xóa một mẫu tin nào đó có khả năng ảnh hưởng đến các mẫu tin khác trong cơ sở dữ liệu. Vì vậy, có thể sẽ phát sinh lỗi khi thực hiện xóa một mẫu tin nào đó.
- GridView có cho phép thực hiện chức năng xóa dữ liệu

### 2.5. Thêm mẫu tin

Điều khiển **GridView** hỗ trợ chỉnh sửa và xóa mẫu tin, nhưng nó không hỗ trợ thêm mẫu tin. Để thêm mẫu tin, ta cần phải sử dụng **DetailsView** và **FormView**. Về nguyên lý cơ bản thì giống nhau có nghĩa là ta cần phải khai báo câu lệnh InsertCommand để thực hiện thêm mẫu tin vào cơ sở dữ liệu:

```
<asp:SqlDataSource ID="sourceNhanvien" runat="server"
 ProviderName="System.Data.SqlClient"
```

```

ConnectionString="<%$ ConnectionStrings:QLNV %>"
SelectCommand=
 "SELECT Manhanvien, Tennhanvien, Diachi, Dienthoai FROM Nhanvien"
InsertCommand="INSERT INTO Nhanvien (Manhanvien, Tennhanvien, Diachi,
Dienthoai) VALUES (@Manhanvien,@ Tennhanvien,@ Diachi,@ Dienthoai)">
</asp:SqlDataSource>
```

Với chức năng thêm dữ liệu, ta không cần quan tâm đến lỗi conflict bởi vì không có ảnh hưởng nào với mẫu tin được thêm vào. Đoạn script sau sử dụng DetailsView, để thêm mẫu tin vào table trong nguồn dữ liệu sourceNhanvien, AutoGenerateInsertButton=true để bổ sung thêm nút lệnh insert vào trang web.

```

<asp:DetailsView ID="DetailsView1" runat="server">
 DataSourceID = "sourceNhanvien" AutoGenerateInsertButton="true">
</asp:DetailsView>
```

## 2.6. Ví dụ:

Đoạn mã lệnh sau đây được dùng để cập nhật thông tin của nhân viên trong table Nhanvien. Ta cũng có thể xóa một nhân viên nào đó với điều kiện mẫu tin được xóa không vi phạm ràng buộc toàn vẹn được đặt ra. Trong đoạn code này có sử dụng lớp <asp:Gridview> được trình bày ở phần sau:

```

<asp:SqlDataSource ID="sds_Nhanvien" runat="server">
 ProviderName="System.Data.SqlClient"
 ConnectionString="<%$ConnectionStrings:QLNV %>"
 SelectCommand="Select * from Nhanvien"
 UpdateCommand="Update Nhanvien
 Set Manhanvien=@Manhanvien,
 Tennhanvien=@Tennhanvien
 where Manhanvien=@MaNhanvien"
 DeleteCommand="Delete FFrom Nhanvien where
 Manhanvien=@Manhanvien">
</asp:SqlDataSource>

<asp:GridView ID="gv_dsNhanvien" DataSourceID="sds_Nhanvien" runat=server
DataKeyNames="Manhanvien"
 AutoGenerateColumns="false" AllowPaging=true PageSize=5 >
<Columns>
 <asp:BoundField HeaderText="Mã Nhân viên" DataField="MaNhanvien" />
 <asp:BoundField HeaderText="Tên Nhân Viên" DataField="TenNhanvien" />
 <asp:CommandField ShowEditButton=true HeaderText="Cập nhật" />
 <asp:CommandField ShowDeleteButton=true HeaderText="Xóa" />
 <asp:TemplateField HeaderText="Phòng ban">
 <ItemTemplate>
 Địa chỉ:<%#Eval("Diachi") %>
 Điện thoại:<%#Eval("Dienthoai") %>
 </ItemTemplate>
 </asp:TemplateField>
</Columns>
</asp:GridView>
```

### **III. Một số hạn chế của SqlDataSource**

- **Truy xuất dữ liệu được nhúng trực tiếp vào trang:** Để tạo SqlDataSource control, ta cần phải viết câu lệnh SQL phức tạp trong chính trang web của mình. Không thể chỉnh sửa câu truy vấn mà không làm thay đổi trang web. Hạn chế này không thể chấp nhận được bởi vì việc chỉnh sửa câu truy vấn rất thường xảy ra theo yêu cầu.
- **Bảo trì các ứng dụng lớn:** Mỗi trang truy xuất cơ sở dữ liệu cần **SqlDataSource** riêng. Trường hợp khi ta có nhiều trang sử dụng cùng một câu truy vấn.
- **Thiếu linh hoạt:** Mỗi công việc cần truy xuất cần một SqlDataSource riêng. Nếu bạn muốn cung cấp người dùng nhiều cách để xem hoặc truy vấn dữ liệu, ta có thể bị chìm ngập trong các đối tượng nguồn dữ liệu. Đây là một công việc vô cùng phức tạp đối với các ứng dụng lớn.

#### **Câu hỏi/Bài tập củng cố:**

1. Nêu các thuộc tính cần thiết để tạo nguồn dữ liệu SqlDataSource
2. Nêu các câu lệnh trong điều khiển SqlDataSource
3. Nêu tên các loại tham số trong điều khiển SqlDataSource
4. Thực hành tạo Form trang 93 bằng công cụ Wizard và viết code.

# Bài 6: ASP.NET VỚI XML

## Mục tiêu học tập:

- Trình bày vài trò của XML trong ASP.net
- Nêu một số quy tắc để tạo file XML
- Sử dụng XML làm nguồn dữ liệu cho một số Web control

### I. Vai trò của XML trong .net

Nơi hữu ích của XML không phải trực tiếp trong ứng dụng web mà là nền hỗ trợ ứng dụng web. Microsoft sử dụng XML trong ASP.net. Vì vậy để hiểu rõ được ứng dụng của XML trong ASP.net, chúng ta cần phải có kiến thức về XML.

#### 1.1. Tập tin cấu hình

ASP.NET lưu trữ các cài đặt trong định dạng XML có thể đọc được từ người dùng như: machine.config và web.config. XML cung cấp các cú pháp để lưu trữ dữ liệu được chuẩn hóa và phù hợp bằng cách sử dụng các thẻ. Nếu ta có một ít kiến thức về XML, ta sẽ dễ dàng hiểu được các tập tin cấu hình trong ASP.net được tổ chức ra sao.

#### 1.2. Truy xuất dữ liệu trong ADO.NET

Đối tượng ADO.NET DataSet có thể trình bày bất kỳ dữ liệu nào như tài liệu XML mà không cần bước chuyển đổi nào. Điều này có một vài kết quả thú vị như sau. Ví dụ, nó cho phép bạn dễ dàng lưu thông tin bạn truy xuất trong cơ sở dữ liệu XML để sử dụng sau này. Đặc điểm này rất hữu ích đối với các ứng dụng người dùng không thể kết nối với mạng.

#### 1.3. Một số ví dụ về XML

```
<?xml version="1.0"?>
<SuperProProductList>
 <Product>
 <ID>1</ID>
 <Name>Chair</Name>
 <Price>49.33</Price>
 <Available>True</Available>
 <Status>3</Status>
 </Product>
 <Product>
 <ID>2</ID>
 <Name>Car</Name>
 <Price>43399.55</Price>
 <Available>True</Available>
 <Status>3</Status>
 </Product>
 <Product>
 <ID>3</ID>
 <Name>Fresh Fruit Basket</Name>
 <Price>49.99</Price>
```

```
<Available>False</Available>
<Status>4</Status>
</Product>
</SuperProProductList>
```

### Giải thích:

Định dạng này tương đối dễ hiểu. Mỗi một sản phẩm product được đóng với thẻ <Product>, mỗi một mẩu thông tin có tên tương ứng. Các thành phần được lưu trữ theo các cấp để thể hiện mối quan hệ. Khi cần thiết, XML có thể cung cấp cấu trúc thành phần, các lập trình viên có thể định nghĩa các thành phần mà bạn cần sử dụng. Đó là lý do tại sao XML được mô tả như là meta language. Nó là ngôn ngữ được dùng để tạo ngôn ngữ cho riêng bạn. Trong tập tin ví dụ SuperProProductList trên, ngôn ngữ XML language định nghĩa các thành phần như <Product>, <ID>, <Name>.

## II. Cơ bản về XML.

### 2.1. Một vài quy tắc khi tạo file XML

Các thành phần XML được soạn trong thẻ bắt đầu <Name> và thẻ kết thúc </Name>. Nếu bạn thêm vào thẻ bắt đầu, bạn cần phải thêm vào thẻ kết thúc tương ứng. Cách viết này tương tự như cấu trúc trong điều khiển ASP.NET.

Khoảng cách giữa cách thành phần bị loại bỏ. Có nghĩa là bạn có thể sử dụng tabs một cách tùy ý. Ta cũng có thể sử dụng các ký tự trong nội dung của một thành phần. Bạn không thể thêm cần thêm vào ký tự đặc biệt như <> và dấu & mà phải sử dụng một số ký tự tương đương như: &lt; and &gt; (<>) và &amp; (&). Ký tự thay thế này sẽ được tự động chuyển đổi trong các lớp .NET phù hợp

Tất cả các thành phần phải được đặt bên trong thành phần gốc (root). Trong ví dụ ở trên thành phần SuperProProductList là thành phần gốc. Khi thành phần gốc (root) được kết thúc nghĩa là tài liệu XML được đóng lại.

Mỗi thành phần phải được đóng lại hoàn toàn. Nói cách khác bạn cần mở thành phần con, bạn cần phải đóng nó trước khi bạn đóng thành phần cha. Ví dụ: <Product><ID></ID></Product> thì đúng, nhưng <Product><ID></Product></ID> thì sai.

Các tài liệu XML phải được bắt đầu với khai báo XML như <?xml version="1.0"?>. Chức năng là để thông báo tài liệu chứa XML và chỉ định mã lệnh đặc biệt. Tuy nhiên, các parser XML làm việc tương đối tốt nếu phần nội dung này bị bỏ qua.

### 2.2. Thuộc tính:

Thuộc tính dùng để thêm một số thông tin vào thành phần. Thay vì thêm một số thẻ con, ta có thể sử dụng thuộc tính tại thẻ. Trong cộng đồng XML, thường không có sự đồng thuận trong việc sử dụng thuộc tính hay thành phần con khi tạo tài liệu XML. Trong ví dụ SuperProProductList sau đây sử dụng ID và Name thay vì thành phần con ID và Name.

```
<?xml version="1.0" encoding="utf-8" ?>
<SuperProProductList xmlns="yourURI" >
 <Product ID="1" Name="Chair">
 <Price>49.33</Price>
 </Product>
 <Product ID="2" Name="Car">
 <Price>43398.55</Price>
```

```

</Product>
<Product ID="3" Name="Fresh Fruit Basket">
 <Price>49.99</Price>
</Product>
</SuperProProductList>

```

Sử dụng thuộc tính trong XML thì khá nghiêm ngặt hơn trong HTML. Trong XML, các thuộc tính phải luôn luôn có giá trị, và những giá trị này phải sử dụng dấu """. Ví dụ: <Product Name="Chair" /> thì đúng, nhưng <Product Name=Chair /> hoặc <Product Name /> thì không đúng. Tuy nhiên, bạn cần phải linh hoạt trong sử dụng dấu "". Tuy nhiên cũng có những trường hợp ta có thể sử dụng dấu nháy đơn thay vì dấu nháy đôi.

### 2.3. Các lớp XML:

.NET cung cấp một tập các lớp để thao tác với XML trong một vài namespaces bắt đầu với System.Xml. Một trong những đặc điểm dễ nhầm lẫn của sử dụng XML với .net là kết hợp các lớp cần dùng. Nhiều trường hợp cung cấp các chức năng tương tự trong cách khá khác nhau.

Phần lớn các ví dụ mà bạn thấy sử dụng các kiểu được định nghĩa trong gói System.Xml namespace. Các lớp này cho phép bạn đọc và viết các file XML, thao tác dữ liệu XML trong bộ nhớ và kiểm tra tính hợp lệ của tài liệu XML. Một vài chức năng của XML được trình bày trong chương này như sau:

- Đọc và viết XML một cách trực tiếp, đơn giản như khi bạn đọc và viết các files sử dụng XmlTextWriter and XmlTextReader.
- Thao tác với XML như một tập hợp các đối tượng trong bộ nhớ như XmlDocument và XmlNode.
- Kết buộc nguồn dữ liệu XmlDataSource để trình bày thông tin XML.

## III. Ví dụ XML với c#

### 3.1. Đọc XML và đưa vào DataSet

#### Tập tin Data.XML

```

<?xml version="1.0" encoding="utf-8" ?>
<SuperProProductList xmlns="yourURI" >
 <Product ID="1" Name="Chair">
 <Price>49.33</Price>
 </Product>
 <Product ID="2" Name="Car">
 <Price>43398.55</Price>
 </Product>
 <Product ID="3" Name="Fresh Fruit Basket">
 <Price>49.99</Price>
 </Product>
</SuperProProductList>

```

#### Tập tin Data.xsd

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema
 targetNamespace="yourURI"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified" >

```

```

<xs:element name="SuperProProductList">
 <xs:complexType>
 <xs:sequence maxOccurs="unbounded">
 <xs:element name="Product">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="Price" minOccurs="1" type="xs:double" />
 </xs:sequence>
 <xs:attribute name="ID" use="required" type="xs:int" />
 <xs:attribute name="Name" use="required" type="xs:string" />
 </xs:complexType>
 </xs:element>
 </xs:sequence>
 </xs:complexType>
</xs:element>
</xs:schema>

```

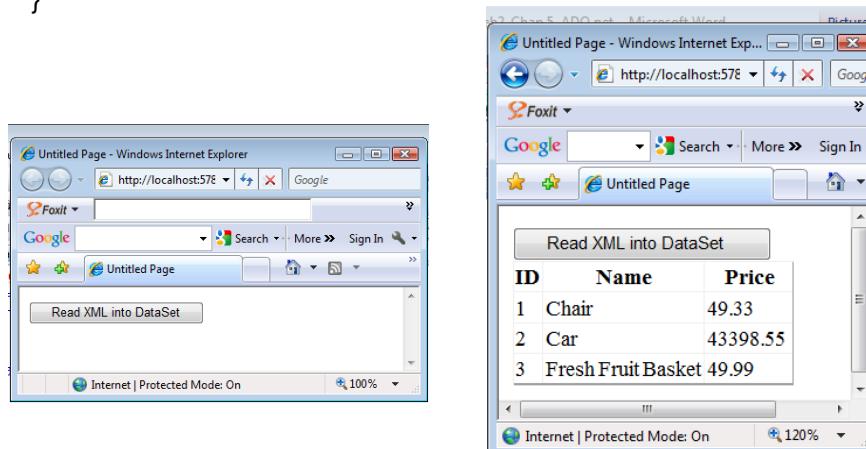
#### Đoạn mã trong chương trình:

```

protected void cmdXmlToDataSet_Click(object sender, EventArgs e)
{
 XmlDataDocument dataDoc = new XmlDataDocument();
 dataDoc.DataSet.ReadXmlSchema("Data.xsd");
 dataDoc.Load("Data.xml");

 gridData.DataSource = dataDoc.DataSet;
 gridData.DataBind();
}

```



Hình 40: Form Kết quả ví dụ trang 102

### 3.2. Đọc XML với XMLDataSource

- Khai báo nguồn dữ liệu XMLDataSource

```

<asp:XmlDataSource ID="XmlDataSource1" runat="server" DataFile "~/Video.xml">
</asp:XmlDataSource>

```

- Liên kết nguồn dữ liệu với GridView

```
<asp:GridView ID="GridView1" runat="server" AllowPaging="True"
 AutoGenerateColumns="False" DataSourceID="XmlDataSource2">
<Columns>
 <asp:BoundField DataField="ID" HeaderText="ID" SortExpression="ID" />
 <asp:BoundField DataField="Name" HeaderText="Name" SortExpression="Name" />
</Columns>
</asp:GridView>
```

### Câu hỏi củng cố:

1. Trình bày vài trò của XML trong ASP.net
2. Nêu một số quy tắc để tạo file XML
3. Nêu các bước để sử dụng XML làm nguồn dữ liệu cho một số Web control như GridView

# Bài tập chương 5: QUẢN LÝ ALBUM BÀI HÁT

## 1. Mục tiêu:

- Sử dụng các câu lệnh Insert, Select để truy vấn và thao tác trên CSDL SQL Server.
- Sử dụng Parameter để truyền tham số cho câu lệnh SQL.
- Binding dữ liệu từ CSDL vào các controls như DropDownList, GridView.
- Sử dụng Session để lưu các thông tin và truy xuất biến session ở các trang web khác nhau.
- Sử dụng Response.Redirect để chuyển người dùng sang web page khác.

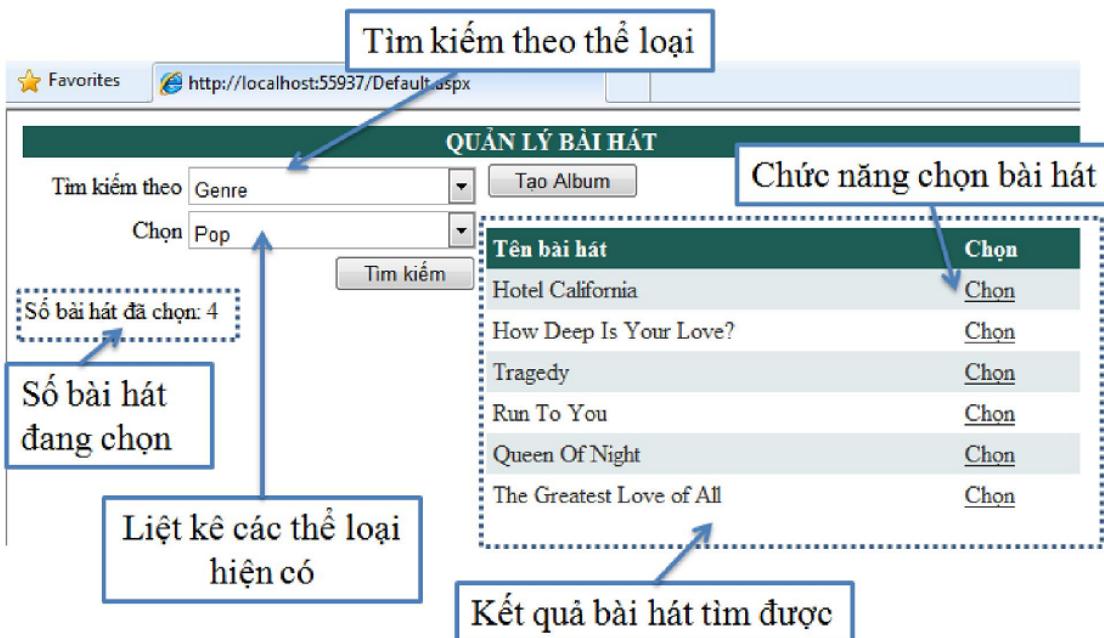
## 2. Yêu cầu:

- Tạo một ứng dụng web cơ bản cho phép quản lý album bài hát:
  - o Cho phép user xem các bài hát theo các mục như: **tên tác giả, thể loại, album**
  - o User có thể tạo mới một album từ các bài hát đã có trong danh sách bài hát
- Ứng dụng gồm 2 trang chính như sau:
  - o **SongManaging.aspx**: trang này cho phép user chọn xem các bài hát theo tên tác giả, thể loại và các album có sẵn. Giao diện của trang web như sau:

The screenshot shows a web application window titled "QUẢN LÝ BÀI HÁT". At the top left, there is a dropdown menu labeled "Tìm kiếm theo" with options "Genre", "Chọn", and a dropdown menu currently set to "Pop". To the right of this is a "Tạo Album" button. Below this is a search input field with the placeholder "Tìm kiếm" and a "Tim kiem" button. The main area displays a table with columns "Tên bài hát" and "Chọn". The table contains the following data:

Tên bài hát	Chọn
Hotel California	<a href="#">Chọn</a>
How Deep Is Your Love?	<a href="#">Chọn</a>
Tragedy	<a href="#">Chọn</a>
Run To You	<a href="#">Chọn</a>
Queen Of Night	<a href="#">Chọn</a>
The Greatest Love of All	<a href="#">Chọn</a>

Hình 1: Minh họa giao diện trang SongManaging.aspx



Hình 2: Thao tác tìm kiếm và chọn bài hát để đưa vào album

**Mô tả:** Mục “**Tìm kiếm theo**”, tạo sẵn 3 mục chọn là: **Artist**, **Genre** và **Album** trong dropdownlist. Mỗi khi chọn một tiêu chuẩn nào đó thì mục “Chọn” sẽ hiện giá trị có thể có trong cơ sở dữ liệu tương ứng với mục **Tìm kiếm theo**. Ví dụ nếu mục tìm kiếm theo chuẩn chọn Artist thì trong mục chọn này sẽ truy vấn trong bảng Artist lấy ra tất cả các nghệ sĩ và đưa vào DropDownList Chọn.

Sau khi chọn xong, thì kích vào button tìm kiếm, kết quả sẽ liệt kê ra tất cả bài hát theo tiêu chuẩn tìm kiếm trên trong GridView. GridView này chỉ hiển thị tên bài hát và một liên kết có chức năng chọn. User có thể chọn các bài hát vừa ý để tạo ra một album khác. Mỗi lần chọn một bài hát, thông tin bài hát này được lưu lại (*sử dụng mã bài hát*) trong session. User có thể chọn tiêu chuẩn tìm kiếm khác để xem và chọn các bài khác. Khi hoàn tất quá trình chọn bài hát user kích vào button “**Tạo Album**” để chuyển sang phần tạo album mới từ các bài hát đã chọn (*gọi trang Album.aspx*).

- **Trang Album.aspx:** Cho phép user tạo album mới với tên gọi và danh sách các bài hát đã chọn trong trang SongManaging.aspx.
- **Mô tả:** trang này lấy danh sách các bài hát đã chọn trong bước trước ở trang SongManaging.aspx, tại trang này user có thể hiệu chỉnh lại danh sách bài hát bằng cách xóa các bài hát trước khi thực hiện thao tác cuối cùng tạo album.



Hình 3: Trang thêm album

## HƯỚNG DẪN

**Bước 1:** Tạo CSDL SongAlbum có các bảng như sau:

- Albums:** chứa danh sách các album
- Songs:** chứa danh sách các bài hát
- AlbumSong:** chứa danh sách các bài hát thuộc album
- Artist:** danh sách các nghệ sĩ
- e

Column Name	Data Type	Allow Nulls
AlbumID	int	<input type="checkbox"/>
AlbumName	nvarchar(100)	<input checked="" type="checkbox"/>
CreateDate	smalldatetime	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Bảng Album

Column Name	Data Type	Allow Nulls
AlbumID	int	<input type="checkbox"/>
SongID	int	<input type="checkbox"/>
		<input type="checkbox"/>

Bảng AlbumSong

	Column Name	Data Type	Allow Nulls
PK	ArtistID	int	<input type="checkbox"/>
PK	GenreID	tinyint	<input type="checkbox"/>
	GenreName	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Bảng Genres

	Column Name	Data Type	Allow Nulls
PK	SongID	int	<input type="checkbox"/>
	SongName	nvarchar(100)	<input checked="" type="checkbox"/>
	GenreID	tinyint	<input checked="" type="checkbox"/>
	ArtistID	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Bảng Song

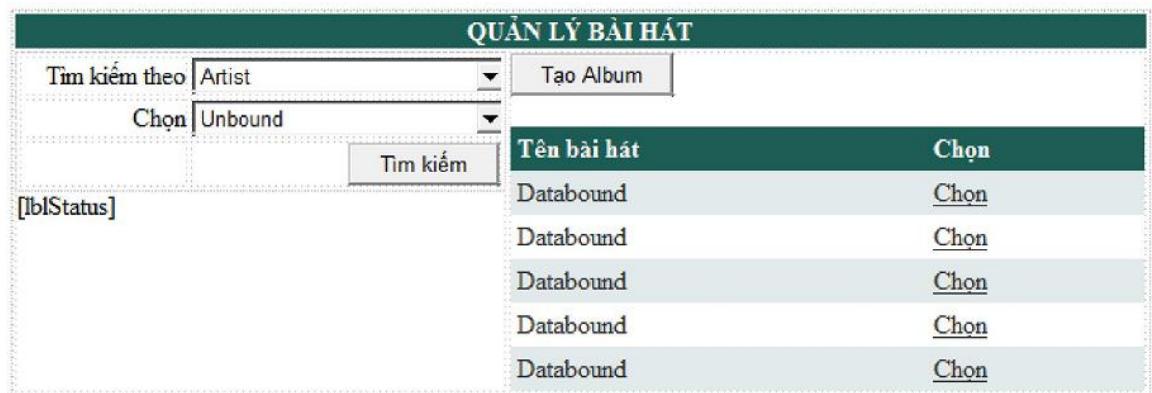
Một số dữ liệu cho các bảng:

Artist	
ID	ArtistName
1	Boyzzone
2	Backstreet Boys
3	Westlife
4	Metallica
5	Aerosmith
6	Taylor Swift
8	Louis Armstrong
9	Kenny G

Genres	
GenreID	GenreName
1	Pop
2	Rock
3	Country
4	Jazz
5	Hiphop

SongID	SongName	GenreID	ArtisID
1	Happy New Year	1	1
2	Dancing Queen	1	1
3	No Matter What	1	2
4	I'll Never Break your heart	1	3
5	As long as you love me	1	3
6	Soledad	1	4
7	Seasons in the sun	1	4
8	Too hard to say good bye	1	4
9	All out of love	1	4
10	The unforgiven II	2	5
11	Nothing else matters	2	5
12	My Apocalypse	2	5
13	One	2	5
14	I Don't want to miss a thing	2	6
15	Dream on	2	6
16	Lay it down	2	6
17	Kings and Queens	2	6
18	Back to December	3	7
19	White Horse	3	7
20	Tell me why	3	7
21	Beautiful Eyes	3	7
22	What a Wonderful World	4	8
23	A Kiss to Build a Dream On	4	8
24	Fall Again	4	9
25	Forever in Love	4	9
26	Songbird	4	9

**Bước 2:** Tạo trang web **SongManaging.aspx** có layout như sau:



Tên	Kiểu	Thuộc tính	Diễn giải
ddlSearchBy	DropDownList	Items = { Artist, Genre, Album }	Combobox cho phép chọn tiêu chuẩn tìm kiếm
ddlSearchOption	DropDownList		Chứa các giá trị có thể chọn, để tìm kiếm, giá trị này phụ thuộc vào giá trị của ddlSearchBy bên trên
btnSearch	Button		Kích vào để tìm bài hát
lblStatus	Label		Hiển thị thông tin số bài hát đã chọn
gvSongs	GridView		Hiển thị kết quả các bài hát tìm được
btnCreateAlbum	Button		Gọi trang Album để tạo album mới từ các bài hát đã chọn

**Bước 3:** Tạo chuỗi kết nối CSDL dữ liệu trong file cấu hình web.config. Bổ sung vào tag <configuration> một tag <appSettings> như sau:

```
<appSettings>
 <add key="ConnectionString" value="<chuỗi kết nối>"></add>
</appSettings>
```

Để lấy chuỗi kết nối trong web.config trong code behind như sau

String chuoiketnoi =

```
ConfigurationManager.AppSettings["ConnectionString"].ToString();
```

**Bước 4:** Tạo hàm FillSearchOption trong code behind của trang SongManaging.aspx: hàm này sẽ lấy thông tin user chọn trong combobox ddlSearchBy và truy xuất vào cơ sở dữ liệu để lấy dữ liệu tương ứng.

```
private void FillSearchOption()
```

```
{
```

```
 string scnn = ConfigurationManager.AppSettings["ConnectionString"].ToString();
```

```

SqlConnection conn = new SqlConnection(scnn);
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
SqlDataAdapter da = new SqlDataAdapter(cmd);
// tạo datatable để lấy dữ liệu từ database
DataTable dt = new DataTable();
// kiểm tra xem SearchBy được chọn
switch (ddlSearchBy.SelectedIndex)
{
 case 0: // Nếu người dùng chọn xem Artist
 cmd.CommandText = "SELECT * FROM artists";
 break;
 case 1: // Nếu người dùng chọn xem Genres
 cmd.CommandText = "SELECT * FROM genres";
 break;
 case 2: // Nếu người dùng chọn xem Albums
 cmd.CommandText = "SELECT * FROM albums";
 break;
}
try
{
 da.Fill(dt); // đưa dữ liệu vào datatable
}
catch (SqlException ex)
{
 Response.Write(ex.Message);
}
// hiển thị kết quả từ datatable ra combobox ddlSearchOption
ddlSearchOption.DataSource = dt.DefaultView;
ddlSearchOption.DataTextField = dt.Columns[1].ColumnName;
ddlSearchOption.DataValueField = dt.Columns[0].ColumnName;
ddlSearchOption.DataBind();
} //FillSearchOption

```

**Bước 5:** Khởi tạo giá trị khi trang web được gọi lần đầu cho dropdownlist trong hàm Page\_Load.

```

protected void Page_Load(object sender, EventArgs e)
{
 if (!IsPostBack)

```

```

 { // mặc định chọn mục chọn đầu tiên
 ddlSearchBy.SelectedIndex = 0;
 FillSearchOption(); // gọi hàm FillSearchOption ở Bước 4
 }
}

```

**Bước 6:** Tạo hàm FillGridView: hàm này được gọi khi user kích vào nút lệnh “**Tìm kiếm**”, hàm có chức năng lấy các thông tin được chọn từ người dùng trong 2 Dropdownlist là *ddlSearchBy* và *ddlSearchOption* để truy vấn cơ sở dữ liệu tìm ra các bài hát thỏa các tiêu chí được chọn, sau đó kết quả hiển thị trong GridView.

```

private void FillGridView()
{
 string scnn = ConfigurationManager.AppSettings["ConnectionString"].ToString();
 SqlConnection cnn = new SqlConnection(scnn);
 SqlCommand cmd = new SqlCommand();
 cmd.Connection = cnn;
 SqlDataAdapter da = new SqlDataAdapter(cmd);
 // tạo datatable để lấy dữ liệu
 DataTable dt = new DataTable();
 // lấy giá trị được chọn trong SearchOption
 string value = ddlSearchOption.SelectedValue;
 // kiểm tra xem search theo tiêu chuẩn nào
 switch (ddlSearchBy.SelectedIndex)
{
 case 0: // lấy bài hát theo id của nghệ sĩ
 cmd.CommandText = "SELECT * FROM Songs WHERE
 ArtistID = "+ value;
 break;
 case 1: // lấy bài hát theo id của thể loại
 cmd.CommandText = "SELECT * FROM Songs WHERE
 GenreID = "+value;
 break;
 case 2: // lấy các bài hát thuộc album id
 cmd.CommandText = "SELECT * FROM Songs WHERE
 SongID in (SELECT SongID FROM AlbumSong
 WHERE AlbumID = " + value+ ")";
 break;
}
try
{
 da.Fill(dt); // Đưa kết quả tìm kiếm vào dataTable dt
}

```

```

 }
 catch (SqlException ex)
 {
 Response.Write(ex.Message);
 }
 // hiển thị kết quả danh sách bài hát trong datagrid
 gvSongs.DataSource = dt.DefaultView;
 gvSongs.DataBind();
 }
}

```

**Bước 7.** Khai báo trình xử lý sự kiện click của button btnSearch

```

protected void btnSearch_Click(object sender, EventArgs e)
{
 FillGridView(); // thực hiện tìm kiếm và hiển thị ds bài hát
}

```

**Bước 8. Thiết kế lại GridView để hiển thị danh sách tên bài hát.**

Trong gridview ta cần hiển thị 2 cột “tên bài hát”, và cột còn lại là cột chức năng “chọn”.

```

<asp:GridView ID="gvSongs" runat="server" AutoGenerateColumns="False"
 DataKeyNames="SongID"
 OnSelectedIndexChanged="gvSongs_SelectedIndexChanged">
 <Columns>
 <asp:BoundField DataField="SongName" HeaderText="Tên bài hát" />
 <asp:ButtonField ButtonType=Link Text="Chọn"
 CommandName="Select" />
 </Columns>
</asp:GridView>

```

- Thuộc tính DataKeyNames cho biết tên cột dữ liệu làm khóa trong GridView. Mỗi một dòng tương ứng trong Gridview sẽ có giá trị chính là mã tương ứng trong tập mã bài hát SongID

- Mỗi khi người dùng chọn bài hát để đưa vào Album, mã bài hát SongID đó sẽ được lưu trữ vào trong một biến Session có tên là SelectedSongs theo cú pháp sau Session[“SelectedSongs”], là dạng ArrayList.

- Khai báo trình xử lý sự kiện **gvSongs\_SelectedIndexChanged** cho GridView, sự kiện này được gọi khi một nút lệnh chọn bài hát được click.

Nội dung của hàm xử lý sự kiện **gvSongs\_SelectedIndexChanged** cho GridView

```

protected void gvSongs_SelectedIndexChanged(object sender, EventArgs e) {
 int index = gvSongs.SelectedIndex;
}

```

```

int songID = (int)gvSongs.SelectedDataKey.Values["SongID"];
//Kiểm tra mã bài hát được chọn && Session đã có rồi
if (Session["selectedSongs"] != null)
{
 ArrayList selectedSongs = (ArrayList)Session["selectedSongs"];
 if (!selectedSongs.Contains(songID))
 selectedSongs.Add(songID);
}
else
{
 ArrayList selectedSongs = new ArrayList();
 selectedSongs.Add(songID);
 Session["selectedSongs"] = selectedSongs;
}
lblStatus.Text = "Số bài hát được chọn " +
 ((ArrayList)Session["selectedSongs"]).Count;
}

```

7. Khai báo xử lý khi user kích vào button “**Tạo Album**”, button này cho phép chuyên user sang trang Album.aspx để thêm album

```
Response.Redirect("album.aspx");
```

8. Tạo trang **Album.aspx** có layout như sau:

QUẢN LÝ ALBUM			
Tên Album			Tạo Album
<b>Phải nhập tên Album</b>			
Tên bài hát	Tên tác giả	Thể loại	Xóa
Databound	Databound	Databound	X
Databound	Databound	Databound	X
Databound	Databound	Databound	X
Databound	Databound	Databound	X
Databound	Databound	Databound	X
[lblStatus]			

Chi tiết các controls trên trang Album như sau:

Tên	Kiểu	Thuộc tính	Điễn giải
txtAlbumName	TextBox		Nhập tên của album mới
btnCreateAlbum	Button		Chức năng tạo album
rfvAlbumName	RequiredFieldValidator	ControlToValidate = txtAlbumName ErrorMessage="Phải nhập tên Album"	Xác nhận xem user có nhập tên album hay không, nếu không nhập thì không tạo mới
gvSongs	GridView	DataKeyField = SongID	Hiển thị các bài hát đã chọn trong trang SongManaging.aspx
lblStatus	Label		Hiển thi thông tin: lỗi hoặc hoàn tất...

#### 9. Khai báo GridView gvSongs như sau:

```
<asp:GridView ID="gvAlbum" DataKeyNames="SongID" runat="server"
 AutoGenerateColumns=false
 onSelectedIndexChanged="gvAlbum_SelectedIndexChanged">
 <Columns>
 <asp:BoundField DataField="SongName" HeaderText="Tên bài hát" />
 <asp:BoundField DataField="TenArtist" HeaderText="Tên tác giả" />
 <asp:BoundField DataField="GenreName" HeaderText="Thể loại" />
 <asp:CommandField ShowSelectButton=true HeaderText="Xóa" />
 <asp:ButtonField ButtonType="Link" HeaderText="Xóa" Text="X"
 CommandName="select" />
 </Columns>
</asp:GridView>
```

10. Tạo hàm FillGridView để lấy các bài hát đã chọn và hiển thị lên GridView

**protected void Page\_Load(object sender, EventArgs e)**

{

    if (!IsPostBack)

        FillGridView();

}

**private void FillDataGrid()** {

    if(Session["selectedSongs"]==null){

        lbsStatus.Text = "<a href=\"default.aspx\">Chưa chọn bài hát</a>";

        btnCreateAlbum.Enabled=false;

```

 return;
 }

 //Kiểm tra số mục chọn được chọn
 ArrayList selectedSongs = (ArrayList) Session["selectedSongs"];
 if(selectedSongs.Count<1){
 lbsStatus.Text=Chưa chọn bài hát ";
 btnCreateAlbum.Enabled=false;
 return;
 }

 string songIDs="";
 foreach(int id in selectedSongs)
 songIDs +=id+", ";
 songIDs=songIDs.Remove(songIDs.Length-2, 2);
 string constr
 WebConfigurationManager.ConnectionStrings["QlBaihat"].ConnectionString;
 SqlConnection con=new SqlConnection(constr);
 SqlCommand cmd = new SqlCommand();
 cmd.Connection=con;
 cmd.CommandText ="Select * from song s, Genres g, Artist a where s.artistid =
a.artistID and s.genreid = g.genreid and songid in ("+songIDs+ ")";
 SqlDataAdapter da=new SqlDataAdapter(cmd);
 DataTable dt=new DataTable();
 da.Fill(dt);
 gvAlbum.DataSource = dt.DefaultView;
 gvAlbum.DataBind();
}

```

11. Cài đặt chức năng khi người dùng chọn xóa một bài hát khỏi danh sách

```

protected void gvAlbum_IndexChanged(object sender, EventArgs e) {
 int songID =(int) gvAlbum.SelectedDataKey.Values["SongID"];
 ArrayList selectedSongs = (ArrayList)Session["selectedSongs"];
 if (selectedSongs.Contains(songID)) {
 selectedSongs.Remove(songID);
 Session["selectedSongs"] = selectedSongs;
 FillGridView();
 }
}

```

12. Khai báo trình xử lý sự kiện click của button: “Tạo album”

```

protected void btnCreateAlbum_Click(object sender, EventArgs e)
{

```

```

if (!IsValid) return; // nếu chưa validation xong
// lấy tên của album
string AlbumName = txtAlbumName.Text.Trim();
// lưu AlbumID
int AlbumID=0;

// thực hiện thao tác thêm album mới
string scnn = ConfigurationManager.AppSettings["ConnectionString"];
SqlConnection cnn = new SqlConnection(scnn);
SqlCommand cmd = new SqlCommand();
cmd.Connection = cnn;
cmd.CommandText = "Insert into Albums
values(@albumname,@createDate);
select MAX(AlbumID) from albums";
// tạo tên album
cmd.Parameters.Add("@albumname", SqlDbType.VarChar);
cmd.Parameters["@albumname"].Value = AlbumName;
// ngày tạo album
cmd.Parameters.Add("@createDate", SqlDbType.SmallDateTime);
cmd.Parameters["@createDate"].Value = DateTime.Now.ToString();
try
{
 cnn.Open();
 AlbumID = Convert.ToInt32(cmd.ExecuteScalar());// thực thi
 if (AlbumID > 0) // thêm thành công
 lblStatus.Text = "Thêm album thành công!";
}
catch (SqlException ex)
{
 Response.Write(ex.Message);
}
finally
{
 cnn.Close();
}

// thêm thông tin vào bảng AlbumSong
cmd.CommandText = "INSERT INTO AlbumSong
values(@AlbumID,@SongID)";

```

```

cmd.Parameters.Add("@AlbumID", SqlDbType.Int);
cmd.Parameters["@AlbumID"].Value = AlbumID;
cmd.Parameters.Add("@SongID", SqlDbType.Int);
ArrayList SelectedSongs = (ArrayList)Session["SelectedSongs"];
try
{
 cnn.Open();
 foreach (int id in SelectedSongs)
 {
 cmd.Parameters["@SongID"].Value = id;
 cmd.ExecuteNonQuery();
 }
}
catch (SqlException ex)
{
 Response.Write(ex.Message);
}
finally { cnn.Close(); }
// xóa tất cả các bài đã chọn
SelectedSongs.Clear();
Session["SelectedSongs"] = SelectedSongs;
// cập nhật lại giao diện
gvSongs.DataSource = null;
gvSongs.DataBind();
FillGridView();
}

```

---Hết---

# Chương 6: ĐIỀU KHIỂN TRÌNH BÀY DỮ LIỆU

## Mục tiêu học tập:

- Tạo trang web với GridView
- Tạo trang web với ListView
- Tạo trang web với DetailView

### I. Giới thiệu tổng quan

**GridView:** GridView là một all-in-one grid superpower trong ASP.net. Nó hiển thị tất cả các mẫu tin trong các dòng của table, có thể sử dụng nhiều cột, templates, hoặc kết hợp cả hai. GridView được đưa vào ASP.NET 2.0 và không thay đổi ở ASP.NET 3.5.

**ListView:** ListView một điều khiển dựa trên mẫu không có nhiều tính năng như GridView nhưng có thể hiển thị dữ liệu mà không cần thẻ <table>. ListView chỉ có trong ASP.NET 3.5.

**DetailsView:** DetailsView trong ASP.NET's hiển thị dữ liệu của từng mẫu tin. DetailsView sử dụng định dạng bảng và hỗ trợ mẫu định dạng sẵn (templates). DetailsView được giới thiệu ở ASP.NET 2.0 và không thay đổi trong ASP.NET 3.5.

### II. Điều khiển Gridview

Gridview là một control rất linh hoạt cho phép hiển thị dữ liệu dưới dạng table với nhiều hàng và nhiều cột. Nó bao gồm một tập chức năng bao gồm chọn, phân trang, sắp xếp và chỉnh sửa. Đặc điểm nổi bật của Gridview hơn DataGrid là hỗ trợ tự động phát sinh mã lệnh theo ngữ cảnh. Mỗi một mẫu tin trong nguồn dữ liệu có thể là một dòng trong lưới. Mỗi trường trong mẫu tin là một cột trong lưới. Một số tính năng được cung cấp bởi Gridview như sau:

- Cung cấp chức năng ready-made bao gồm: phân trang, sắp xếp, chọn và chỉnh sửa
- Có thể hiển thị một mẫu tin ở một thời điểm
- Cho phép gán thuộc tính AutoGenerateColumns = true
- Định nghĩa thẻ <Columns>: Được định nghĩa và thêm các cột trước các cột được tự động tạo ra. Đây là cách tốt nhất là nên sử dụng Gridview

Kiểu cột	Mô tả
BoundField	Hiển thị dữ liệu text từ một trường trong nguồn dữ liệu.
ButtonField	Hiển thị nút lệnh đối với mỗi mục chọn trong danh sách.
CheckBoxField	Hiển thị checkbox cho mỗi mục chọn trong danh sách. Được dùng đối với các trường có kiểu dữ liệu là true/false ( <i>trong SQL Server, các trường này có kiểu dữ liệu bit</i> ).
CommandField	Cho phép hiển thị nút lệnh Select và Edit.
HyperLinkField	Hiển thị nội dung ( <i>một trường trong nguồn dữ liệu hoặc dữ liệu tĩnh</i> ).
ImageField	Hiển thị dữ liệu hình ảnh.
TemplateField	Xác định nhiều trường, custom controls và thẻ HTML bình thường sử dụng custom template.

Kiểu cột cơ bản là BoundField, có thể liên kết một cột với một đối tượng dữ liệu. Ví dụ, đây dùng để định nghĩa cột hiển thị dữ liệu của trường TenNhanvien trong table Nhanvien.

```
<asp:BoundField DataField="TenNhanvien" HeaderText="Tên nhân viên" />
```

Sử dụng định nghĩa từng cột là một cải tiến tốt hơn so với việc chọn tự động phát sinh cột.

Ví dụ: Tạo một GridView để hiển thị dữ liệu cho 4 cột kết quả dựa trên câu truy vấn SELECT \* FROM Nhanvien. Đoạn mã lệnh sau đây được dùng để thực hiện yêu cầu trên.

```
<asp:GridView ID="gridEmployees" runat="server" DataSourceID="sourceNhanvien"
AutoGenerateColumns="False">
 <Columns>
 <asp:BoundField DataField="Manhanvien" HeaderText="Mã nhân viên" />
 <asp:BoundField DataField="TenNhanvien" HeaderText="Tên nhân viên" />
 <asp:BoundField DataField="Diachi" HeaderText="Địa chỉ" />
 <asp:BoundField DataField="Dienthoai" HeaderText="Điện thoại" />
 </Columns>
</asp:GridView>
<asp:SqlDataSource ID="sourceNhanvien" runat="server"
 ConnectionString="<%$ ConnectionStrings:QLNV %>"
 ProviderName="System.Data.SqlClient"
 SelectCommand="SELECT * FROM Nhanvien">
</asp:SqlDataSource>
```

Kết quả của đoạn mã lệnh trên như hình sau, trong đó có một số cột không được hiển thị trong bảng kết quả bởi vì trong Gridview không có khai báo.

Mã nhân viên	Tên nhân viên	Địa chỉ	Điện thoại
1	Nguyen Thi	Càng Long, Trà Vinh	0919800657
2	Võ Thị Thu Duyên	Càng Long, Trà Vinh	0123456789
3	Lê Thành Nguyễn	Cầu Ngang, Trà Vinh	0987987987
5	Nguyễn Ngọc Thanh Thúy	Vũng Liêm, Vĩnh Long	0981112221
6	Trần Thị Xuân Thúy	Măng Thít, Vĩnh Long	0981112222
7	Cao Anh Thư	Cầu Ngang, Trà Vinh	0981112223
8	Nguyễn Song Toàn	Trà Cú, Trà Vinh	0981112224
9	Võ Thị Thu Trang	Trà Cú, Trà Vinh	0981112225
10	Phạm Thị Cẩm Hồng	Tam Bình, Vĩnh Long	0981112226
12	Nguyễn Duy Kiệt	Mỏ Cày, Bến Tre	0981112228
13	Trang Thị Minh	Châu Thành, Trà Vinh	0981112229
15	Huỳnh Thị Ngọc Quý	Vũng Liêm, Vĩnh Long	0981112230
16	Nguyễn Thanh Sang	Châu thành - Trà vinh	0981112231
18	Oanh Thone	Châu thành - Trà vinh	0917145587
20	Vo Cong Duc	Mang Thích - Vĩnh Long	0917145587

**Hình 41: Form Kết quả ví dụ trang 119**

**Lưu ý:** Định nghĩa từng cột sẽ thực hiện nhanh hơn so với tự động phát sinh cột. Bởi vì các cột được sinh ra tự động buộc GridWiew phải hiển thị dữ liệu ngay khi thực hiện.

## 2.1. Định dạng cột

Một số thuộc tính trong cột

Thuộc tính	Mô tả
DataField	Khai báo tên trường hoặc thuộc tính (đối với đối tượng) của mẫu dữ liệu cần hiển thị trong cột
DataFormatString	Định dạng dữ liệu bên trong cột. Thông thường đối với trường dữ liệu kiểu ngày hoặc kiểu số
HeaderText, FooterText, và HeaderImageUrl	<ul style="list-style-type: none"> <li>- HeaderText: Cho biết chuỗi hiển thị ở dòng đầu tiên của GridView</li> <li>- FooterText: cho biết chuỗi hiển thị ở dòng cuối cùng của GridView.</li> <li>- HeaderImageURL: Cho biết địa chỉ của hình ảnh hiển thị ở cột.</li> </ul>
ReadOnly	Là true nếu giá trị trong cột không thay đổi ở chế độ edit. Các cột là khóa chính thông thường ở chế độ Read-only.
NullDisplayText	Định nghĩa đoạn text được hiển thị thay cho giá trị rỗng. Mặc định là rỗng.

#### **Định dạng dữ liệu bên trong một cột**

- Mỗi cột BoundField cung cấp thuộc tính DataFormatString để thay đổi hiển thị bằng cách sử dụng định dạng string. Format
    - Một chuỗi định dạng như {0:C}. Trong đó 0 hiển thị cho giá trị định dạng, chữ sau xác định kiểu định dạng. C nghĩa là định dạng theo kiểu tiền tệ.
    - Ví dụ định dạng một số kiểu phô biến như sau:

```
<asp:BoundField DataField="UnitPrice" HeaderText="Price"
 DataFormatString="{0:C}" />
+ Định dạng cột BirthDate theo dạng tháng/ngày/năm
<asp:BoundField DataField="BirthDate" HeaderText="Birth Date"
 DataFormatString="{0:MM/dd/yy}" />
```

## Các kiểu định dạng số

Kiểu	Chuỗi	Ví dụ
Tiền tệ	{0:C}	\$1,234.50
Số khoa học	{0:E}	1.234.50E+004
Phần trăm	{0:P}	45.6%
Số thực	{0:F?{}}	Phụ thuộc vào vị trí của số chữ số thập phân {0:F3} sẽ là 123.400. {0:F0} sẽ là 123

## Các kiểu định dạng ngày giờ

Kiểu	Chuỗi	Ví dụ
Short Date	{0:d}	M/d/yyyy (ví dụ: 10/30/2008)
Long Date Short Time	{0:f}	dddd, MMMM dd, yyyy HH:mm aa (for Ví dụ Monday, January 30, 2008 10:00 AM)

Long Date	{0:D}	dddd, MMMM dd, yyyy (for example: Monday, January 30, 2008)
Long Date Long Time	{0:F?}	dddd, MMMM dd, yyyy HH:mm:ss aa (Ví dụ: Monday, January 30, 2008 10:00:23 AM)

## 2.2. Định dạng hiển thị Gridview

Định dạng Gridview bao gồm một vài chức năng sau:

Kiểu	Mô tả
HeaderStyle	Thay đổi định dạng dòng đầu tiên của Gridview
RowStyle	Định dạng các dòng còn lại
AlternatingRowStyle	Thêm vào RowStyle
SelectedRowStyle	Định dạng cho các ô được chọn
EditRowStyle	Định dạng ô đang ở chế độ cập nhật
EmptyDataRowStyle	Định dạng dòng không có dữ liệu
FooterStyle	Định dạng dòng cuối cùng của gridview
PagerStyle	Cấu hình định dạng dòng với liên kết đến trang

### Chọn một dòng trong GridView

Chọn một dòng trong GridView bằng cách click chuột vào một nút lệnh nào đó để làm nổi bật lên dòng được chọn hoặc làm kích hoạt một sự kiện nào đó.

Điều khiển GridView có hỗ trợ sẵn cho chức năng chọn. Ta chỉ cần thêm vào một cột với kiểu là CommandField và gán thuộc tính ShowSelectButton là true. Cột với kiểu CommandField có thể trả về một hyperlink, button, hoặc một image. Ta gán thuộc tính ButtonType với giá trị là hyperlink, button, image. Sau đó cung cấp chuỗi hiển thị cho thuộc tính SelectText, mặc định là từ Select. Nếu là hình ảnh thì gán chuỗi chỉ đường dẫn vào thuộc tính SelectImageUrl.

#### Ví dụ:

Tạo nút lệnh với nhãn Select để chọn một dòng nào đó trong Gridview

```
<asp:CommandField ShowSelectButton="True"ButtonType="Button"
SelectText="Select" />
```

Tạo nút lệnh Select với nhãn là một hình để thực hiện chọn một dòng nào đó

```
<asp:CommandField ShowSelectButton="True"ButtonType="Image"
SelectImageUrl="select.gif" />
```

### 2.3. Sự kiện trong GridViews

Khi người dùng click chọn nút lệnh Select, một loạt các sự kiện khác sẽ xảy ra. Đầu tiên, sự kiện **GridView.SelectedIndexChanged** sẽ được gọi. Sau đó là thuộc tính **GridView.SelectedIndex** được xác định tương ứng với dòng được chọn. Cuối cùng, sự kiện **GridView.SelectedIndexChanged** sẽ được gọi để thực thi.

Ta có thể thực hiện viết lệnh cho sự kiện **GridView.SelectedIndexChanged** để cho biết nhân viên nào được chọn.

**Ví dụ:** Đoạn mã trong tập tin GridviewSelectRow.aspx dùng để tạo một nguồn dữ liệu có tên là sourceNhanvien, một GridView có tên là gvNhanvien và một nhãn tên là lblKetqua để hiện thị thông tin của nhân viên được chọn.

```

<asp:GridView ID="gvNhanvien" runat="server" DataKeyNames="Manhanvien"
 OnSelectedIndexChanged="gvNhanvien_SelectedIndexChanged"
 DataSourceID="sourceNhanvien" AutoGenerateColumns="False">
 <SelectedRowStyle BackColor=Yellow />
 <Columns>
 <asp:BoundField DataField="Manhanvien" HeaderText="Mã nhân viên" />
 <asp:BoundField DataField="Tennhanvien" HeaderText="Tên nhân viên" />
 <asp:BoundField DataField="Diachi" HeaderText="Địa chỉ" />
 <asp:BoundField DataField="Dienthoai" HeaderText="Điện thoại" />
 <asp:CommandField ShowSelectButton=trueButtonType=Link
 SelectText="Chọn" HeaderText="Chọn"/>
 </Columns>
</asp:GridView>
<asp:SqlDataSource ID="sourceNhanvien" runat="server"
 ConnectionString="<%$ ConnectionStrings:QLNV %>"
 ProviderName="System.Data.SqlClient"
 SelectCommand="SELECT Manhanvien, Tennhanvien, Diachi, Dienthoai
 FROM Nhanvien">
</asp:SqlDataSource>

<asp:Label ID="lblketqua" runat=server></asp:Label>
 - Đoạn mã cài đặt sự kiện khi chọn một nhân viên nào đó
 gvNhanvien_SelectedIndexChanged như sau:
 protected void gvNhanvien_SelectedIndexChanged(object sender, EventArgs e)
 {
 int index = gvNhanvien.SelectedIndex;
 int ID = (int)gvNhanvien.SelectedDataKey.Values["Manhanvien"];
 string hoten = gvNhanvien.SelectedRow.Cells[1].Text;
 string diachi = gvNhanvien.SelectedRow.Cells[2].Text;
 lblketqua.Text = " Bạn đã chọn: ";
 lblketqua.Text += "
Họ tên: " + hoten;
 lblketqua.Text += "
Địa chỉ: " + diachi;

 }
 - Màn hình kết quả

```

Mã nhân viên	Tên nhân viên	Địa chỉ	Điện thoại	Chọn
1	Nguyen Thi	Càng Long, Trà Vinh	0919800657	<a href="#">Chọn</a>
2	Võ Thị Thu Duyên	Càng Long, Trà Vinh	0123456789	<a href="#">Chọn</a>
3	Lê Thành Nguyễn	Cầu Ngang, Trà Vinh	0987987987	<a href="#">Chọn</a>
5	Nguyễn Ngọc Thanh Thúy	Vũng Liêm, Vĩnh Long	0981112221	<a href="#">Chọn</a>
6	Trần Thị Xuân Thúy	Mảng Thịt, Vĩnh Long	0981112222	<a href="#">Chọn</a>
7	Cao Anh Thư	Cầu Ngang, Trà Vinh	0981112223	<a href="#">Chọn</a>
8	Nguyễn Song Toàn	Trà Cú, Trà Vinh	0981112224	<a href="#">Chọn</a>
9	Võ Thị Thu Trang	Trà Cú, Trà Vinh	0981112225	<a href="#">Chọn</a>
10	Phạm Thị Cẩm Hồng	Tam Bình, Vĩnh Long	0981112226	<a href="#">Chọn</a>
12	Nguyễn Duy Kiệt	Mỏ Cày, Bến Tre	0981112228	<a href="#">Chọn</a>
13	Trang Thị Minh	Châu Thành, Trà Vinh	0981112229	<a href="#">Chọn</a>
15	Huỳnh Thị Ngọc Quý	Vũng Liêm, Vĩnh Long	0981112230	<a href="#">Chọn</a>
16	Nguyễn Thanh Sang	Châu thành - Trà vinh	0981112231	<a href="#">Chọn</a>
18	Oanh Thone	Châu thành - Trà vinh	0917145587	<a href="#">Chọn</a>
20	Vo Cong Duc	Mang Thich - Vĩnh Long	0917145587	<a href="#">Chọn</a>

Bạn đã chọn:  
Họ tên: Lê Thành Nguyễn  
Địa chỉ: Cầu Ngang, Trà Vinh

**Hình 42: Form Kết quả ví dụ trang 123**

#### 2.4. Sử dụng GridView Template:

Cho đến bây giờ, ta đã xem xét qua các ví dụ được sử dụng trong Gridview để hiển thị dữ liệu trong từng cột riêng biệt. Nếu ta mong muốn đặt nhiều giá trị trong cùng một ô, hoặc cần phải chỉnh sửa nội dung của một ô bằng cách bổ sung thêm vào các thẻ HTML hay server controls. Ta có thể sử dụng chức năng của TemplateField

**Ví dụ 1:** Để tạo cột Nhân viên gồm 3 thông tin như tên nhân viên, giới tính và điện thoại như sau:

```
<asp:TemplateField HeaderText="Nhân viên">
 <ItemTemplate>
 <%# Eval("Tennhanvien") %>
 <%# Eval("Gioitinh") %>
 <%# Eval("Dienthoai") %>
 </ItemTemplate>
</asp:TemplateField>
```

Gridview nạp dữ liệu từ nguồn dữ liệu và duyệt qua tập các mẫu tin. Nó xử lý ItemTemplate đối với một mục chọn, evaluate biểu thức kết buộc dữ liệu, và thêm vào các thẻ HTML vào table.

**Hàm Eval():** Một phương thức tính thuộc lớp **System.Web.UI.DataBinder**. **Eval()** thực hiện truy xuất dữ liệu có liên kết với mẫu tin hiện tại.

**Ví dụ:**

```
<%# Eval("BirthDate", "{0:MM/dd/yy}") %>
```

Hiển thị dữ liệu cột BirthDate với định dạng là MM/dd/yy

### Ví dụ:

```
<asp:GridView ID="gvNhanvien" runat="server" DataKeyNames="Manhanvien"
 DataSourceID="sourceNhanvien" AutoGenerateColumns="False">
 <SelectedRowStyle BackColor=Yellow />
 <Columns>
 <asp:BoundField DataField="Manhanvien" HeaderText="Mã nhân viên" />
 <asp:TemplateField HeaderText="Nhân viên">
 <ItemTemplate>

Tên nhân viên: <%# Eval("Tennhanvien") %>

Địa chỉ: <%#Eval("Diachi") %>

Điện thoại: <%#Eval("Dienthoai") %>
 </ItemTemplate>
 </asp:TemplateField>
 <asp:CommandField ShowSelectButton=true ButtonType=Link
 SelectText="Chọn" HeaderText="Chọn"/>
 </Columns>
</asp:GridView>
<asp:SqlDataSource ID="sourceNhanvien" runat="server"
 ConnectionString="<%$ ConnectionStrings:QLNV %>"
 ProviderName="System.Data.SqlClient"
 SelectCommand= "SELECT Manhanvien, Tennhanvien, Diachi, Dienthoai FROM
Nhanvien">
</asp:SqlDataSource>
```



Hình 43: Form Kết quả ví dụ trang 124

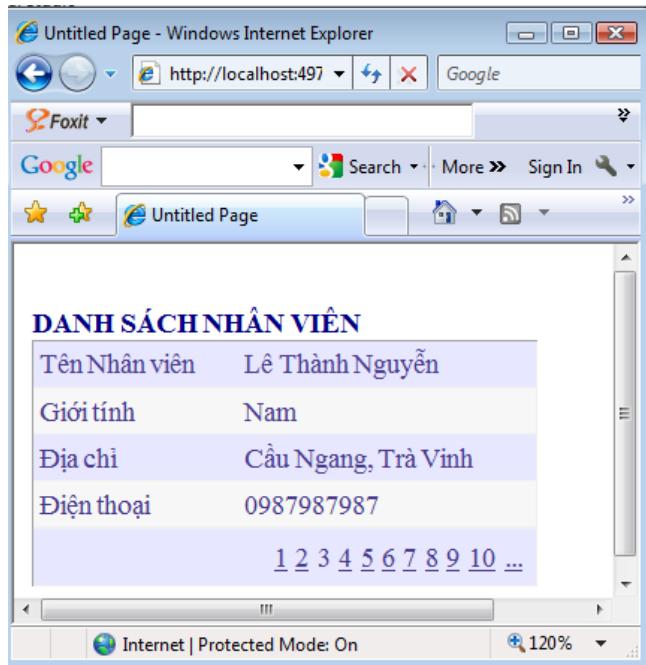
### III. DetailView

- DetailsView được thiết kế để hiển thị mẫu tin tại một thời điểm. Mỗi một thông tin trong mẫu tin được trình bày trong 1 dòng của table. DetailsView cũng có thể liên kết với tập dữ liệu các mẫu tin. Tuy nhiên nó chỉ hiển thị mẫu tin đầu tiên trong nhóm, vì vậy ta cần phải dùng chức năng của điều khiển phân trang
- Khi AllowPaging = true. Ta có thể cấu hình phân trang bằng cách sử dụng một số thuộc tính như: PagingStyle PagingSettings giống như trong GridView.
- Tương tự như GridView, DetailView có thể cho phép lập trình viên thực hiện tự động tạo cột cho ứng dụng. Tuy nhiên, ta có thể dùng thẻ <Fields> thay vì <Columns> như trong GridView.
- Ví dụ sau đây tạo một Detail View để hiển thị danh sách nhân viên với thuộc tính AllowPaging = true. Tóm tắt đoạn code được sử dụng như sau:

```
<asp:SqlDataSource ID="sourceNhanvien" runat="server"
 ConnectionString="<%$ ConnectionStrings:QLNV %>"
 ProviderName="System.Data.SqlClient"
 SelectCommand= "SELECT * FROM Nhanvien" >
</asp:SqlDataSource>

<asp:DetailsView ID="DetailsView1" runat="server" AutoGenerateRows=False
 DataSourceID="sourceNhanvien" AllowPaging="True" >
 <Fields>
 <asp:BoundField DataField="Tennhanvien" HeaderText="Tên Nhân viên" />
 <asp:BoundField DataField="Gioitinh" HeaderText="Giới tính" />
 <asp:BoundField DataField="Diachi" HeaderText="Địa chỉ" />
 <asp:BoundField DataField="Dienthoai" HeaderText="Điện thoại" />
 </Fields>
</asp:DetailsView>
```

- Kết quả của yêu cầu trên được hiển thị như hình bên dưới



Hình 35: Form Kết quả ví dụ trang 125

### IV. FormView

- FormView cung cấp một điều khiển cho phép hiển thị và chỉnh sửa từng mẫu tin riêng rẽ.

- Mô hình định dạng trong FormView gần giống với TemplateField trong GridView. Có nghĩa là bạn có thể thực hiện các định dạng sau đây.

Kiểu định dạng	Mô tả
ItemTemplate	Định dạng mẫu tin hiển thị
EditItemTemplate	Định dạng mẫu tin dành cho chỉnh sửa
InsertItemTemplate	Định dạng mẫu tin dành cho insert
FooterTemplate	Định dạng tiêu đề cuối trang
HeaderTemplate	Định dạng tiêu đề đầu trang
EmptyDataTemplate	Định dạng các trường không có dữ liệu
PagerTemplate	Định dạng phân trang

- Ví dụ sau đây để tạo một form hiển thị thông tin của nhân viên, chỉnh sửa, và thêm một mẫu tin mới vào table Nhanvien theo phòng ban được chọn.

Khai báo nguồn dữ liệu cho FormView bao gồm các câu lệnh SelectCommand, InsertCommand, DeleteCommand và UpdateCommand.

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
 ConnectionString="<%$ ConnectionStrings:QLNVConnectionString %>"
 SelectCommand="SELECT * FROM [Nhanvien] WHERE ([Maphong] = @Maphong)"
 DeleteCommand="DELETE FROM [Nhanvien] WHERE [MaNhanvien] =
@MaNhanvien"
 InsertCommand="INSERT INTO [Nhanvien] ([MaNhanvien], [TenNhanvien],
[Ngaysinh], [Diachi], [Dienthoai], [Gioitinh], [Maphong], [Hinhanh]) VALUES (@MaNhanvien,
@TenNhanvien, @Ngaysinh, @Diachi, @Dienthoai, @Gioitinh, @Maphong, @Hinhanh)"
 UpdateCommand="UPDATE [Nhanvien] SET [TenNhanvien] = @TenNhanvien,
[Ngaysinh] = @Ngaysinh, [Diachi] = @Diachi, [Dienthoai] = @Dienthoai, [Gioitinh] =
@Gioitinh, [Maphong] = @Maphong, [Hinhanh] = @Hinhanh WHERE [MaNhanvien] =
@MaNhanvien">
 <SelectParameters>
 <asp:ControlParameter ControlID="DropDownList1" Name="Maphong"
 PropertyName="SelectedValue" Type="Int32" />
 </SelectParameters>
 <DeleteParameters>
 <asp:Parameter Name="MaNhanvien" Type="Int32" />
 </DeleteParameters>
 <UpdateParameters>
 <asp:Parameter Name="TenNhanvien" Type="String" />
 <asp:Parameter DbType="Date" Name="Ngaysinh" />
 <asp:Parameter Name="Diachi" Type="String" />
 <asp:Parameter Name="Dienthoai" Type="String" />
 <asp:Parameter Name="Gioitinh" Type="String" />
 <asp:Parameter Name="Maphong" Type="Int32" />
 <asp:Parameter Name="Hinhanh" Type="Object" />
 <asp:Parameter Name="MaNhanvien" Type="Int32" />
 </UpdateParameters>
 <InsertParameters>
 <asp:Parameter Name="MaNhanvien" Type="Int32" />
 <asp:Parameter Name="TenNhanvien" Type="String" />
 <asp:Parameter DbType="Date" Name="Ngaysinh" />
 <asp:Parameter Name="Diachi" Type="String" />

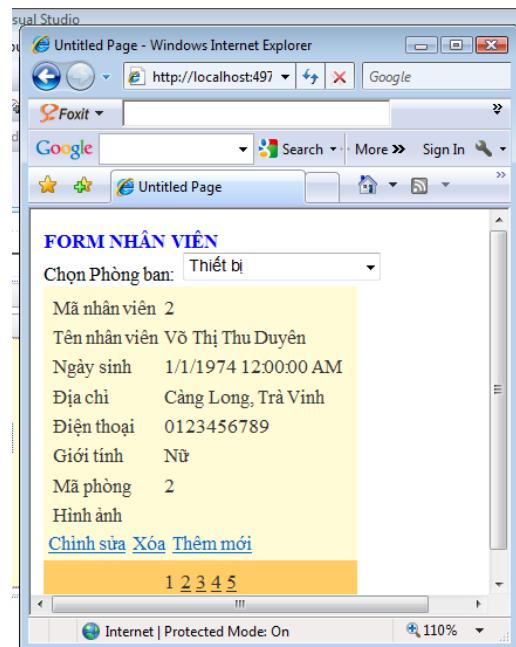
```

```

<asp:Parameter Name="Dienthoai" Type="String" />
<asp:Parameter Name="Gioitinh" Type="String" />
<asp:Parameter Name="Maphong" Type="Int32" />
<asp:Parameter Name="Hinhanh" Type="Object" />
</InsertParameters>
</asp:SqlDataSource>

```

Màn hình thêm mới nhân viên:



*Hình 44: Form Kết quả ví dụ trang 126*

- Màn hình chỉnh sửa thông tin của nhân viên

**FORM NHÂN VIÊN**

Chọn Phòng ban: Quản lý 3

Mã nhân viên: 1  
Tên nhân viên: Nguyen Thi  
Ngày sinh: 1/1/1980 12:00:00 AM  
Địa chỉ: Cảng Long, Trà Vinh  
Điện thoại: 0919800657  
Giới tính: N  
Mã phòng: 1  
Hình ảnh:

[Update](#) [Cancel](#)

1 2 3 4 5 6

*Hình 45: Form Kết quả ví dụ trang 126*

- Màn hình thêm mới nhân viên

**FORM NHÂN VIÊN**

Chọn Phòng ban: Quản lý 3

Mã nhân viên:  
Tên nhân viên:  
Ngày sinh:  
Địa chỉ:  
Điện thoại:  
Giới tính:  
Mã phòng:  
Hình ảnh:

[Thêm mới](#) [Cancel](#)

*Hình 46: Form Kết quả ví dụ trang 128*

# BÀI TẬP CHƯƠNG 6

## 1. Mô tả cơ sở dữ liệu

Cho một cơ sở dữ liệu Quản lý việc mượn sách của độc giả gồm 3 bảng dữ liệu như sau:

- DMSach để lưu trữ danh mục các quyền sách được lưu trữ trong thư viện
- Docgia để lưu trữ thông tin về các độc giả hiện có trong thư viện
- Muon để lưu trữ các thông tin về các giao dịch mượn trả sách của độc giả

Cấu trúc của các bảng dữ liệu được trình bày như trong hình phía dưới.

The screenshot shows three tables from a database named DIEMHANH\_PC\SQL...en - dbo. The first table, DMSach, has columns MaSach (int, primary key), Tuasach (nchar(200)), NhaXB (nchar(40)), NamXB (int), Tacgia (nchar(40)), Ngongnu (nchar(40)), and Soluong (int). The second table, Docgia, has columns MaDG (int, primary key), TenDG (nchar(30)), Gioitinh (nchar(10)), Ngsinh (datetime), Diachi (nchar(200)), and NgayDK (datetime). The third table, Muon, has columns MaDG (int, primary key), MaSach (int), NgayMuon (datetime), NgayTra (datetime), Tienphat (int), and Ghichu (ntext).

Column Name	Data Type	Allow Nulls
MaSach	int	<input type="checkbox"/>
Tuasach	nchar(200)	<input checked="" type="checkbox"/>
NhaXB	nchar(40)	<input checked="" type="checkbox"/>
NamXB	int	<input checked="" type="checkbox"/>
Tacgia	nchar(40)	<input checked="" type="checkbox"/>
Ngongnu	nchar(40)	<input checked="" type="checkbox"/>
Soluong	int	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
MaDG	int	<input type="checkbox"/>
TenDG	nchar(30)	<input checked="" type="checkbox"/>
Gioitinh	nchar(10)	<input checked="" type="checkbox"/>
Ngsinh	datetime	<input checked="" type="checkbox"/>
Diachi	nchar(200)	<input checked="" type="checkbox"/>
NgayDK	datetime	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
MaDG	int	<input type="checkbox"/>
MaSach	int	<input type="checkbox"/>
NgayMuon	datetime	<input type="checkbox"/>
NgayTra	datetime	<input checked="" type="checkbox"/>
Tienphat	int	<input checked="" type="checkbox"/>
Ghichu	ntext	<input checked="" type="checkbox"/>

## 2. Dữ liệu trong các tables trên như sau:

### 2.1. Danh mục sách

The screenshot shows the DMSach table with 10 rows of data. The columns are MaSach, Tuasach, NhaXB, NamXB, Tacgia, Ngongnu, and Soluong. The data includes various book titles, publishers, and publication years.

MaSach	Tuasach	NhaXB	NamXB	Tacgia	Ngongnu	Soluong
1	365 chuyện kể h...	Kim Đồng	...	2005	Vân Lam	...
2	Ông già Khổt -ta...	Kim Đồng	...	2006	Vân Lam	...
3	Lời nguyễn	Giáo dục	...	2007	Nam Cao	...
4	Những câu chuy...	Minh Khai	...	2008	Vân Lam	...
5	Những nhân vật...	Văn hóa, văn nghệ...	2009	Vân Lam	...	10
6	Bộ sách dành ch...	Văn hóa, văn nghệ...	2005	Ann Major	...	10
7	Thảm họa tinh	Văn hóa, văn nghệ...	2006	Nam Cao	...	5
8	Linh hồn và quý ...	Hội nhà văn	...	2007	Vân Lam	...
9	Làm chủ giọng n...	Hội nhà văn	...	2008	Ann Major	...
10	100 ý tưởng Kin...	Kinh doanh	...	2009	Ann Major	...

### 2.2. Danh mục độc giả

MaDG	TenDG	Gioitinh	Ngsinh	Diachi	NgayDK
1	Huỳnh Thị Trúc L...	Nữ	1/1/1985 12:00:...	Châu Thành, Trà Vinh ...	1/1/2011 12:00:00 AM
2	Tăng Định Luật ...	Nam	1/1/1986 12:00:...	Càng Long, Trà Vinh ...	1/12/2010 12:00:00 AM
3	Ngô Đình Bảo Lộ...	Nam	1/1/1987 12:00:...	Châu Thành, Trà Vinh ...	1/10/2010 12:00:00 AM
4	Nguyễn Ngọc Th...	Nam	1/1/1988 12:00:...	Càng Long, Trà Vinh ...	1/9/2010 12:00:00 AM
5	Lê Minh Chuyên ...	Nam	1/1/1982 12:00:...	Vũng Liêm, Vĩnh Long ...	1/8/2010 12:00:00 AM
6	Phan Thị Kim Xu...	Nữ	1/1/1981 12:00:...	Vũng Liêm, Vĩnh Long ...	1/7/2010 12:00:00 AM
7	Trần Thiện Khoa...	Nữ	1/1/1982 12:00:...	Càng Long, Trà Vinh ...	1/6/2011 12:00:00 AM
8	Thạch Thị Dung ...	Nữ	1/1/1983 12:00:...	Châu Thành, Trà Vinh ...	1/5/2011 12:00:00 AM
9	Tăng Chí Nghĩa ...	Nam	1/1/1984 12:00:...	Càng Long, Trà Vinh ...	1/4/2011 12:00:00 AM
10	Huỳnh Ngọc Yến...	Nữ	1/1/1989 12:00:...	Châu Thành, Trà Vinh ...	1/3/2011 12:00:00 AM
11	Trần Minh Khang...	Nam	1/1/1990 12:00:...	Vũng Liêm, Vĩnh Long ...	1/2/2011 12:00:00 AM
12	Thạch Ngọc Giàu...	Nam	1/1/1991 12:00:...	Châu Thành, Trà Vinh ...	1/1/2011 12:00:00 AM
13	Nguyễn Trí Cảnh...	Nam	1/1/1992 12:00:...	Càng Long, Trà Vinh ...	1/11/2011 12:00:00 AM
14	Lâm Trường Gia...	Nam	1/1/1993 12:00:...	Châu Thành, Trà Vinh ...	11/11/2011 12:00:00 AM
15	Nguyễn Hiếu ...	Nữ	1/1/1994 12:00:...	Vũng Liêm, Vĩnh Long ...	11/20/2011 12:00:00 AM

### 2.3. Table Muon

MaDG	MaSach	NgayMuon	Ngaytra	Tienphat	Ghichu
1	1	6/30/2011 12:0...	NULL	0	
2	1	6/30/2011 12:0...	NULL	0	
3	1	6/30/2011 12:0...	NULL	0	
5	1	6/30/2011 12:0...	NULL	0	
2	3	7/11/2011 12:0...	NULL	0	
2	4	7/11/2011 12:0...	NULL	0	
1	5	6/30/2011 12:0...	NULL	0	
1	6	6/30/2011 12:0...	NULL	0	
1	7	7/10/2011 12:0...	NULL	0	
1	8	7/10/2011 12:0...	NULL	0	
2	8	7/11/2011 12:0...	NULL	0	
2	9	7/11/2011 12:0...	NULL	0	
2	10	7/11/2011 12:0...	NULL	0	

### 3. Sử dụng các lớp trong ADO.net ở chương 5

- a. Tạo trang web có tên: **Thongtinsach.aspx** cho phép người dùng xem các thông tin về sách đang được đọc giả mượn:

## XEM THÔNG TIN SÁCH

Chọn tựa sách: **365 chuyện kể hằng đêm**

Thông tin chi tiết:

Tựa sách	365 chuyện kể hằng đêm
Năm Xuất bản	2005
Nhà Xuất bản	Kim Đồng

Độc giả đang mượn:

Mã độc giả	Mã sách	Ngày mượn	Ngày trả	Tiền phạt	Ghi chú
1	1	6/30/2011 12:00:00 AM		0	
2	1	6/30/2011 12:00:00 AM		0	
3	1	6/30/2011 12:00:00 AM		0	
5	1	6/30/2011 12:00:00 AM		0	

### Yêu cầu chi tiết

+ Dropdownlist **Chọn tựa sách** có ID là **ddlTuasach** và các mục chọn của dropdownlist này được load từ table **DMSach** trong CSDL ngay khi trang web được gọi lần đầu tiên.

+ Cài đặt sự kiện cho dropdownlist **ddlTuasach** sao cho khi người dùng chọn một tựa sách nào đó thì chương trình thực hiện tìm kiếm trong table DMSach thông tin chi tiết (**Tuasach, namxb, nhaxb**) của quyển sách được chọn và hiển thị các thông tin này trong DetailsView có tên là dvSach. Đồng thời, chương trình cũng thực hiện tìm kiếm trong CSDL thông tin về tình trạng mượn sách của quyển sách được chọn (**Masach, Madg, Ngaymuon, Ngaytra, Tienphat, Ghichu**) và các thông tin này được hiển thị trong GridView có tên là gvMuon như trên hình ảnh minh họa trên.

Ví dụ trên hình minh họa kết quả tìm kiếm khi người dùng chọn tựa sách có tên là **365 truyện kể hằng đêm**

#### 4. Sử dụng các lớp trong ADO.net

- Tạo trang web có tên: **Thongdoctgia.aspx** cho phép người dùng xem các thông tin về việc mượn sách của độc giả:

## THÔNG TIN ĐỘC GIẢ MUỢN SÁCH

Xem thông tin mượn sách của độc giả: **Huỳnh Thị Trúc Linh**

Mã độc giả	Mã sách	Ngày mượn	Ngày trả	Tiền phạt	Ghi chú
1	1	6/30/2011 12:00:00 AM		0	
1	5	6/30/2011 12:00:00 AM		0	
1	6	6/30/2011 12:00:00 AM		0	
1	7	7/10/2011 12:00:00 AM		0	
1	8	7/10/2011 12:00:00 AM		0	

- Yêu cầu chi tiết:

+ Trong đó Dropdownlist **Xem thông tin mượn sách của độc giả** có ID là **ddlDocgia** và các mục chọn của dropdownlist này được load từ table **Docgia** trong CSDL ngay khi trang web được gọi lần đầu tiên.

+ Cài đặt thủ tục có tên là **sp\_Docgiamuonsach** có tham số là **Madocgia** để thực hiện tìm kiếm các thông tin về việc mượn sách gồm (**MaDG,MaSach, Ngaymuon, Ngaytra, Tienphat, Ghichu**) của độc giả tương ứng.

+ Cài đặt sự kiện cho dropdownlist **ddlDocgia** sao cho khi người dùng chọn tên của một độc giả nào đó thì chương trình thực hiện tìm kiếm trong table **Muon** thông tin chi tiết (**MaDG,MaSach, Ngaymuon, Ngaytra, Tienphat, Ghichu**) về việc mượn sách của độc giả được chọn và hiển thị các thông tin trên trong GridView có tên là **gvMuon** như trên hình ảnh minh họa

+ *Ví dụ trên hình minh họa kết quả tìm kiếm khi người dùng chọn độc giả có tên là Huỳnh Thị Trúc Linh.*

## 5. Sử dụng SQL DataSource

- Tạo trang web có tên là **Thongtinmuonsach.aspx** để xem thông tin độc giả mượn sách. Trong đó các thành phần được mô tả chi tiết như sau:

Mã độc giả	Mã sách	Ngày mượn	Ngày trả	Tiền phạt	Ghi chú	Xóa	Chỉnh sửa
1	1	6/30/2011 12:00:00 AM	7/11/1894 12:00:00 AM	0		<a href="#">Xóa</a>	<a href="#">Chỉnh sửa</a>
1	5	6/30/2011 12:00:00 AM	7/11/1894 12:00:00 AM	0		<a href="#">Xóa</a>	<a href="#">Chỉnh sửa</a>
1	6	6/30/2011 12:00:00 AM	7/11/1894 12:00:00 AM	0		<a href="#">Xóa</a>	<a href="#">Chỉnh sửa</a>
1	7	7/10/2011 12:00:00 AM	7/11/1894 12:00:00 AM	0		<a href="#">Xóa</a>	<a href="#">Chỉnh sửa</a>
1	8	7/10/2011 12:00:00 AM	7/11/1894 12:00:00 AM	0		<a href="#">Xóa</a>	<a href="#">Chỉnh sửa</a>

### - Yêu cầu chi tiết:

+ Tạo một **SqlDataSource** có tên là **sourceDocgia** để liệt kê thông tin **mã độc giả và tên độc giả** hiện có trong cơ sở dữ liệu.

+ Tạo một **Dropdownlist** có tên là **ddlDocgia** để hiển thị dữ liệu từ nguồn dữ liệu **sourceDocgia** ở câu a trên.

+ Tạo một **SqlDataSource** có tên là **sourceMuon**; **sourceMuon** là một nguồn dữ liệu có sử dụng tham số được chọn từ **ddlDocgia** ở câu b; để liệt kê các thông tin mượn sách của độc giả do người dùng chọn ở **ddlDocgia**. Ví dụ: Khi người dùng chọn độc giả có tên là Huỳnh Thị Trúc Linh thì dữ liệu kết quả của **sourceMuon** là danh sách các mẩu tin mượn sách của độc giả trên.

+ Tạo một **GridView** có tên là **gvMuon** để hiển thị dữ liệu từ nguồn dữ liệu **sourceMuon** ở câu c trên. Thêm và kiểm tra các cột mới dạng **ButtonField** để cho phép người dùng thực hiện **xóa, cập nhật** thông tin mượn sách như màn hình thiết kế ở trên.

## 6. Tạo FormView

Tạo một Form view dùng để thực hiện chức năng xem thông tin độc giả, chỉnh sửa thông tin độc giả và thêm độc giả mới vào danh sách độc giả. Các yêu cầu chi tiết như sau:

FORM QUẢN LÝ ĐỘC GIẢ

Chọn tên độc giả: Ngô Đình Bảo Lộc

Thông tin chi tiết	Mã độc giả: 3
	Tên độc giả: Ngô Đình Bảo Lộc
	Giới tính: Nam
	Ngày sinh: 1/1/1987 12:00:00 AM
	Địa chỉ: Châu Thành, Trà Vinh
	Ngày đăng ký: 1/10/2010 12:00:00 AM

Edit Delete New

- Dropdownlist Chọn tên độc giả hiển thị danh sách các độc giả hiện có trong table DMDocgia. Cài đặt chức năng AutoPostBack = true.
- Khu vực formView sẽ hiển thị tất cả thông tin liên quan của độc giả được chọn ở Dropdownlist Chọn tên độc giả ở trên.
- Khi người dùng chọn Edit để chỉnh sửa thông tin. Màn hình sẽ chuyển sang chế độ chỉnh sửa như sau

FORM QUẢN LÝ ĐỘC GIẢ

Chọn tên độc giả: Ngô Đình Bảo Lộc

Thông tin chi tiết	Mã độc giả	3
	Tên độc giả	Ngô Đình Bảo Lộc
	Giới tính	Nam
	Ngày sinh	1/1/1987 12:00:00 AM
	Địa chỉ	Châu Thành, Trà Vinh
	Ngày đăng ký	1/10/2010 12:00:00 AM

Update Cancel

- Màn hình khi người dùng chọn mục chọn New như sau

Untitled Page - Windows Internet Explorer

http://localhost:49775/DataC... Google

Foxit Search Launch

Google Search More > Sign In

Untitled Page Page Tools

**FORM QUẢN LÝ ĐỘC GIÀ**

Chọn tên độc giả:

Thông tin chi tiết

Mã độc giả	<input type="text"/>
Tên độc giả	<input type="text"/>
Giới tính	<input type="text"/>
Ngày sinh	<input type="text"/>
Địa chỉ	<input type="text"/>
Ngày đăng ký	<input type="text"/>

[Insert](#) [Cancel](#)

Done Internet | Protected Mode: On 110%

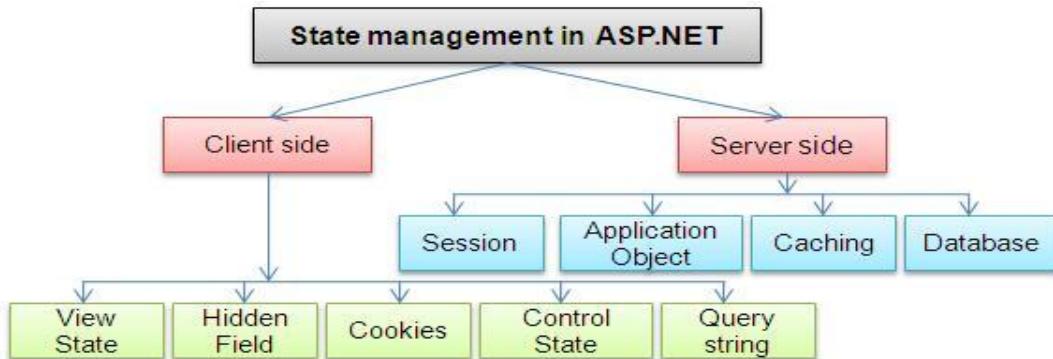
The screenshot shows a Microsoft Internet Explorer window with the title bar "Untitled Page - Windows Internet Explorer". The address bar displays the URL "http://localhost:49775/DataC...". The toolbar includes standard buttons for back, forward, search, and launch. A search bar for "Google" is present. The main content area contains a form titled "FORM QUẢN LÝ ĐỘC GIÀ". It starts with a dropdown menu labeled "Chọn tên độc giả:" with the value "Huỳnh Thị Trúc Linh". Below this is a section titled "Thông tin chi tiết" containing six input fields for "Mã độc giả", "Tên độc giả", "Giới tính", "Ngày sinh", "Địa chỉ", and "Ngày đăng ký". At the bottom of the form are two buttons: "Insert" and "Cancel". The status bar at the bottom of the browser window shows "Internet | Protected Mode: On" and a zoom level of "110%".

# Chương 7: QUẢN LÝ TRẠNG THÁI

## Mục tiêu học tập:

- Sử dụng được một số biến trạng thái ở client.
- Sử dụng được một số biến trạng thái ở server.

### I. Các biến ViewState, Cookies và Sessions



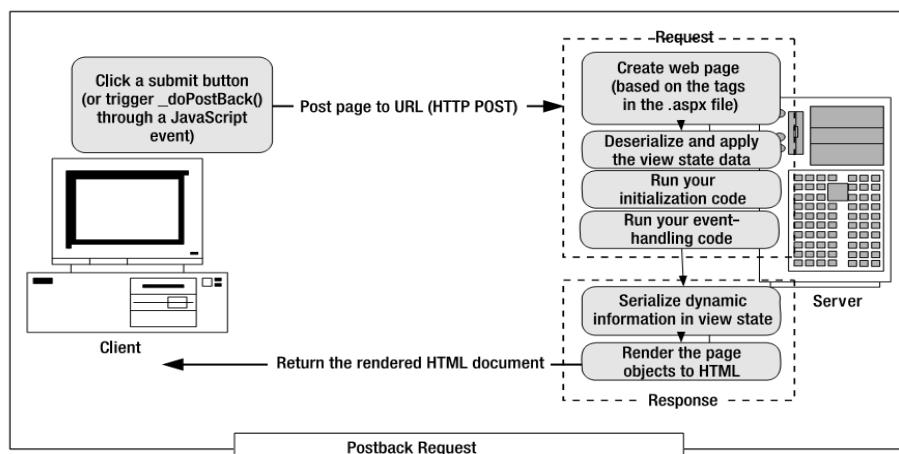
Hình 47: Các biến quản lý trạng thái trong ASP.net

Một số công cụ để hỗ trợ quản lý trạng thái tiêu biểu như sau:

- View State
- Cookies
- Sessions
- Applications

#### 1.1. View State

View state được sử dụng trong các web controls của ASP.net. View state cho phép các web controls giữ lại các thuộc tính của nó giữa các postback. Viewstate được xem như trường ẩn cho trang web được trả về sau cùng



Hình 48: Yêu cầu một trang trong ASP.net

- a. Tạo mới một viewstate
  - ViewState["Counter"] = 1

- Ý nghĩa: Tạo mới một viewstate có tên là Counter
- b. Đọc một viewstate
  - Int count=ViewState[“Counter”]
- c. Khi nào nên sử dụng viewstate
  - Biến dữ liệu có kích thước nhỏ
  - Nội dung dữ liệu thay đổi
  - Không nên sử dụng dữ liệu có tính bảo mật trong viewstate

## 1.2. Cookies

Cookies là các file nhỏ được lưu ở đĩa cứng của máy khách hoặc bộ nhớ của trình duyệt. Được sử dụng để lưu trữ thông tin mà sau này có thể sử dụng lại.

### Thuận lợi:

- Người dùng không nhận biết thông tin được lưu trữ.
- Có thể được dùng bởi bất kỳ trang web nào của ứng dụng và được truy xuất giữa các lần viếng trang web khác nhau.

### Bất lợi:

- Giới hạn bởi chuỗi thông tin gửi vào cookies
- Dễ dàng truy xuất và đọc nếu người dùng thấy và mở file tương ứng
- Người dùng có thể vô hiệu hóa cookies ở trình duyệt.

### Có hai loại cookies:

- Session cookies: Session cookie chỉ tồn tại trong bộ nhớ.
- Persistent cookies: Persistent cookie có thể tồn tại hàng tháng hoặc hàng năm. Nó được lưu trữ vĩnh viễn trong máy tính của người dùng

using System.Net;

// Tạo đối tượng Cookies

```
HttpCookie cookie = new HttpCookie("Preferences");
```

// Gán giá trị cho cookie.

```
cookie["LanguagePref"] = "English";
```

// Tạo thêm cookies khác

```
cookie["Country"] = "US";
```

// đưa cookies xuống trình duyệt

```
Response.Cookies.Add(cookie);
```

```
<form id="form1" runat="server">
 <asp:Label id="lblCookieValue" Text="Cookie Value:">
 AssociatedControlID="txtCookieValue" Runat="server" />
 <asp:TextBox id="txtCookieValue" Runat="server" />
 <asp:Button id="btnAdd" Text="Add Value" OnClick="btnAdd_Click" Runat="server" />
</form>
```

```

protected void btnAdd_Click(object sender, EventArgs e)
{
 Response.Cookies["message"].Value = txtCookieValue.Text;
}

```

Tạo một Persistence Cookies

```

void Page_Load()
{
 int counter = 0;
 if (Request.Cookies["counter"] != null)
 counter = Int32.Parse(Request.Cookies["counter"].Value);
 counter++;
 Response.Cookies["counter"].Value = counter.ToString();
 Response.Cookies["counter"].Expires = DateTime.Now.AddYears(2);
 //Cookies tên counter sẽ tồn tại ở máy khách 2 năm
 lblCounter.Text = counter.ToString();
}

<form id="form1" runat="server">
<div>
 <asp:Label id="lblCounter" Runat="server" />
</div>
</form

You have visited this page
times!


```

### 1.3. Sessions:

- Session là một thực thể của lớp System.Web.SessionState.HttpSessionState.
- Biến session được lưu trữ ở máy chủ server. Nó có thể có bất kỳ kiểu dữ liệu nào để lưu giữ trạng thái được yêu cầu giữa các trang web ở server.
- Ứng dụng cần yêu cầu bảo mật mà không thể lưu trữ thông tin về client trong các view state hoặc trong cookies.
- Đối tượng Session cung cấp truy xuất kiểu từ điển để set các cặp name/value thể hiện dữ liệu của người dùng trong session trên
- Trạng thái của session được dùng để duy trì các thông tin như user's name, user's ID, a shopping cart, hoặc những thành phần khác bị bỏ khi người dùng không còn truy cập website nữa

#### - Ghi Session

Cú pháp:

Session[“Tên session”] = “Giá trị session”

Ví dụ:

Session["message"] = "Hello World!";

Tạo một session tên là message và có giá trị là Hello World!

#### - Đọc dữ liệu từ session

lblMessage.Text = Session["message"].ToString();

Đọc biến session message và hiển thị vào nhãn lblMessage

## **II. Application State**

- Lưu trữ đối tượng toàn cục và có thể được truy xuất bởi bất kỳ client nào. Trạng thái Application dựa trên lớp HttpSessionState
- Hỗ trợ các loại đối tượng tương tự như với Session State, nó cũng có dữ liệu name/value.
- Biết trạng thái Application có thể chấp nhận nhiều kiểu dữ liệu khác nhau. Ứng dụng phổ biến là để tạo bộ đếm toàn cục để theo dõi số lần thực hiện các yêu cầu bởi tất cả các clients.
- Ví dụ, ta có thể tạo một sự kiện global.asax để theo dõi báo nhiêu sessions được tạo hoặc bao nhiêu requests được nhận vào ứng dụng. Hoặc bạn có thể sử dụng các similar logic trong trang Page.Load để theo dõi số lần trang được yêu cầu bởi các clients khác nhau.

```
protected void Page_Load(Object sender, EventArgs e)
{
 int count = 0;
 if (Application["HitCounterForOrderPage"] != null)
 count = (int)Application["HitCounterForOrderPage"];
 count++;
 Application["HitCounterForOrderPage"] = count;
 lblCounter.Text = count.ToString();
}
```

Nếu có 2 client truy xuất biến counter đồng thời, ta cần phải dùng phương thức Lock() và UnLock để quản lý. Điều này cho phép chỉ một client truy xuất vào tập Application vào một thời điểm như sau:

```
protected void Page_Load(Object sender, EventArgs e)
{
 // Acquire exclusive access.
 Application.Lock();
 int count = 0;
 if (Application["HitCounterForOrderPage"] != null)
 count = (int)Application["HitCounterForOrderPage"];
 count++;
 Application["HitCounterForOrderPage"] = count;
 // Release exclusive access.
 Application.UnLock();
 lblCounter.Text = count.ToString();
}
```

## **III. Tập tin ứng dụng global.asax**

Tập tin global.asax cho phép ta viết các thao tác sự kiện mà tác động sự kiện global. Người dùng không cần truy xuất tập tin global.asax trực tiếp. Thay vào đó, tập tin global.asax thực thi đoạn mã tự động để đáp ứng các sự kiện ứng dụng nào đó. Tập tin asax cung cấp một dịch vụ tương tự như tập tin global.asa trong các ứng dụng ASP cổ điển.

Ta có thể lập trình tập tin global.asaxx tương tự như web form. Có một sự khác biệt là tập tin global.asax không chứa thẻ HTML hoặc thẻ ASP.net. Thay vào đó, nó chứa các phương thức với tên cụ thể, rõ ràng. Ví dụ, tập tin global.asax tác động đến sự kiện `HttpApplication.EndRequest` xảy ra trước khi trang được gửi đến người dùng như sau:

```
<%@ Application Language="C#" %>
<script language="C#" runat="server">
 protected void Application_OnEndRequest()
 {
 Response.Write("<hr />This page was served at " +
 DateTime.Now.ToString());
 }
</script>
```

Một số sự kiện không kích hoạt với mỗi yêu cầu:

- **Application\_Start()**: Phương thức này được gọi khi ứng dụng đầu tiên được khởi tạo và ứng dụng được tạo. Sự kiện này thì rất hữu ích để cung cấp mã lệnh người dùng. Ví dụ, ngay tại thời điểm ta load và cache dữ liệu không thay đổi trong suốt thời gian sống của ứng dụng như là navigation trees.
- **Session\_Start()**: phương thức được gọi mỗi khi có một phiên giao dịch mới bắt đầu. Thường được dùng để khởi tạo thông tin người dùng.
- **Application\_Error()**: Phương thức này được gọi mỗi khi có một ngoại lệ xảy ra trong ứng dụng.
- **Session\_End()**: Phương thức này được gọi mỗi khi phiên giao dịch kết thúc. Một phiên giao dịch kết thúc khi mã lệnh người dùng
- **Application\_End()**: Phương thức này được gọi bất cứ khi nào phiên giao dịch dừng. Một phiên làm việc dừng khi,
- **Application\_Disposed()**: Phương thức này được gọi vài lần sau khi ứng dụng được đóng.

#### IV. Tập tin Web.config

##### 4.1. Cấu trúc tập tin web.config.

- Web.config là một tập tin văn bản được sử dụng để lưu trữ thông tin cấu hình của một ứng dụng, được tự động tạo ra khi chúng ta tạo mới ứng dụng web. Tập tin web.config được viết theo định dạng XML.

- Web.config được tạo kể thừa các giá trị từ tập tin Windows\Microsoft.NET\Framework\[Framework Version]\CONFIG\machine.config

##### 4.2. Tập tin cấu hình ứng dụng Web.config:

```
<configuration>
 <appSettings>
 <add key="Ole_Con" value="Provider=SQLOLEDB; Data Source=dlthien;
Initial Catalog=QlBanSach; User ID=Qlbs_app; Password=Qlbs_app"/>
 </appSettings>
```

```

<system.web>
 <compilation defaultLanguage="vb" debug="true"/>
 <customErrors mode="RemoteOnly" />
 <authentication mode="Windows" />
 <authorization>
 <allow users="*" /> <!-- Allow all users -->
 </authorization>
</system.web>
</configuration>

```

#### **4.3. Các cấu hình mặc định**

**<compilation defaultLanguage="vb" debug="true"/>**

defaultLanguage: qui định ngôn ngữ mặc định của ứng dụng.

debug: Bật/tắt chế độ debug của ứng dụng

**<customErrors mode="RemoteOnly"/>**

Đây là một cấu hình khá cần thiết cho ứng dụng Web. Hiệu chỉnh cấu hình này cho phép chúng ta quản lý việc xử lý lỗi khi có lỗi phát sinh trong ứng dụng.

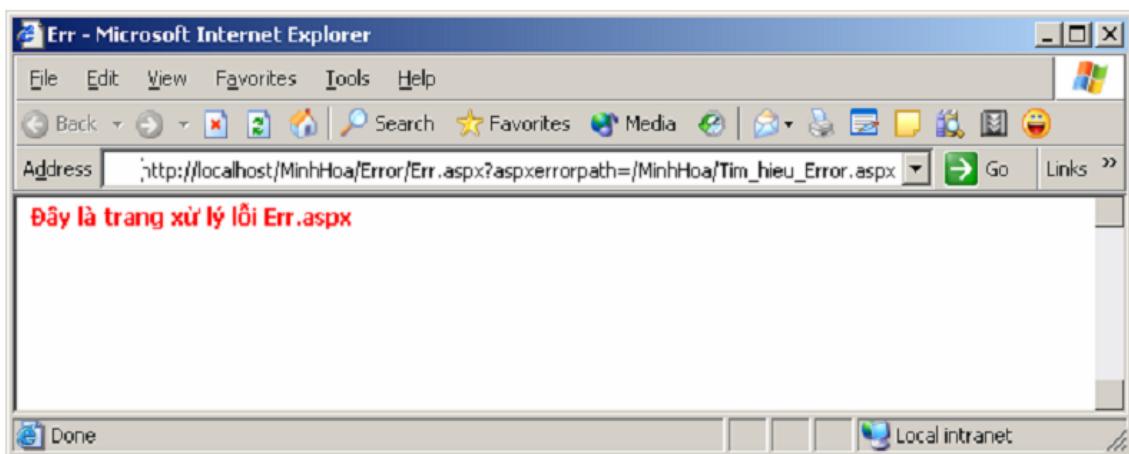
Thuộc tính mode có các giá trị: RemoteOnly, On và Off.

- RemoteOnly: Cho phép người dùng thấy thông báo lỗi của hệ thống hoặc trang thông báo lỗi được chỉ định qua defaultRedirect (nếu có). Thông báo lỗi gồm: Mã lỗi và mô tả lỗi tương ứng

**<customErrors mode="RemoteOnly" defaultRedirect="Error/Err.aspx"/>**

On: Tùy theo giá trị của defaultRedirect mà có các trường hợp tương ứng:

- Có qui định trang thông báo lỗi qua defaultRedirect: Hiển thị trang thông báo lỗi.



**Hình 49: Form kết quả trang 140**

Không có thuộc tính defaultRedirect: Hiển thị trang báo lỗi nhưng không có hiển thị mã lỗi và mô tả lỗi.

#### 4.4. SessionState

```
<sessionState mode="InProc" stateConnectionString="tcpip=127.0.0.1:42424"
 sqlConnectionString="data source=127.0.0.1; Trusted_Connection=yes"
 cookieless="false" timeout="20" />
```

- **mode:** Thuộc tính này có 3 giá trị: InProc, sqlserver (lưu trong database), và stateserver (lưu trong bộ nhớ)

- **stateConnectionString:** Cấu hình địa chỉ và cổng (port) của máy để lưu trữ thông tin của Session trong vùng nhớ (nếu chức năng này được chọn).

- **sqlConnectionString:** Cấu hình kết nối đến SQL Server được dùng để lưu thông tin Session (nếu chức năng này được chọn).

- **cookieless:** Nếu giá trị của thuộc tính này = True, thông tin cookie sẽ được lưu trữ trong URL, ngược lại, nếu = False, thông tin cookies sẽ được lưu trữ tại client (nếu client có hỗ trợ)

- **timeout:** Khoảng thời gian (tính bằng phút) mà đối tượng Session được duy trì. Sau khoảng thời gian này, đối tượng Session sẽ bị huỷ. Giá trị mặc định của thuộc tính này là 20.

#### 4.5. Làm việc với tập tin web.config

Tập tin web.config có hỗ trợ tag `<appSettings>` với 2 thuộc tính là key và value cho phép chúng ta thêm vào các biến dùng để cấu hình ứng dụng.

Lưu ý: Các tên tag trong tập tin cấu hình web.config có phân biệt chữ hoa, chữ thường.

```
<appSettings>
 <add key="Tên biến" value="Giá trị" />
</appSettings>
```

#### Ví dụ:

Tạo biến cấu hình Ole\_Con dùng để lưu trữ thông tin của chuỗi kết nối đến cơ sở dữ liệu SQL Server:

```
<configuration>
 <appSettings>
 <add key="Ole_Con" value="Provider=SQLOLEDB; Data Source=dlthien;
 Initial Catalog=QLBanSach; User ID=Qlbs_app;
 Password=Qlbs_app"/>
 </appSettings>
</configuration>
```

Lấy giá trị đã thiết lập trong tập tin web.config

```
gChuoi_ket_noi = ConfigurationSettings.AppSettings("Ole_Con")
```

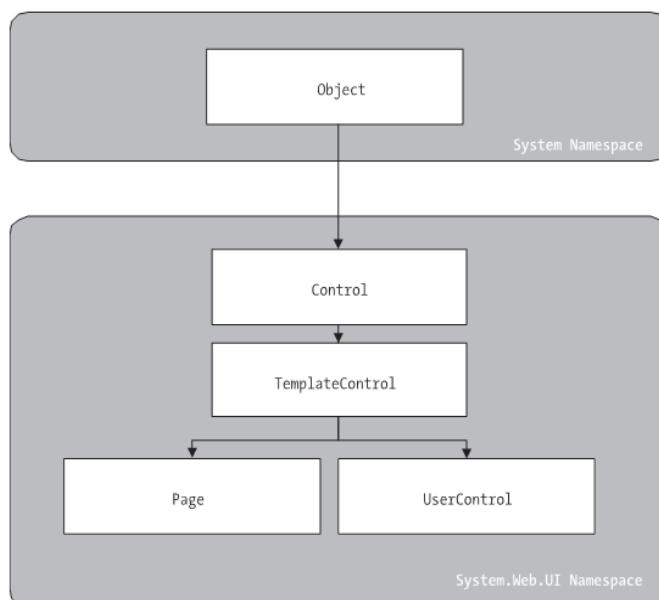
## Bài 2: USER CONTROL & MASTER PAGE

### Mục tiêu học tập:

- Sử dụng User control để tạo giao diện cho trang web
- Sử dụng Master page để tạo giao diện cho website

### I. User controls

- **User controls:** Một user control là một phần nhỏ trên trang web có thể thêm vào các thẻ HTML và các web server controls. User controls là cách tốt nhất để tạo các phần nhỏ được sử dụng nhiều lần trong các trang web.
  - Có thể sử dụng lại user control ở nhiều trang trong cùng một ứng dụng web.
  - Có thể thêm các properties, events và methods.
  - Tập tin User controls có phần mở rộng ascx.



Các tập tin User control (.ascx) tương tự như tập tin của Web Form trong ASP.NET. Giống như web forms, user controls được bao gồm các phần giao diện nhỏ với thẻ điều khiển (tập tin .ascx và có thể sử dụng các đoạn mã nội tuyến bên trong tập tin cs ở code behind).

User controls có thể chứa bất kỳ thành phần nào mà một trang web bình thường có thể có như thẻ HTML, các điều khiển webs, các sự kiện và các đối tượng ASP.net như Application, Session, Request, và Response.

#### Một số điểm khác giữa User Controls và các Web forms

- User controls bắt đầu với tiền chỉ thị Control thay vì tiền chỉ thị Page.
- User controls sử dụng phần mở rộng .ascx thay vì .aspx, và tập tin code behind được thừa kế từ lớp System.Web.UI.UserControl. Do lớp UserControl và Page có thể thừa kế từ lớp TemplateControl nên cả hai lớp này có nhiều phương thức và sự kiện chung.
- User controls không được gọi trực tiếp từ trình duyệt của máy khách. Thay vào đó, user controls được nhúng trực tiếp vào bên trong một trang web khác.

#### Tạo User control

**Website   Add New Item   chọn Web User Control template.**

```
<%@ Control Language="C#" AutoEventWireup="true"
 CodeFile="Header.ascx.cs" Inherits="Header" %>
<table width="100%" border="0" style="background-color: Blue">
 <tr>
 <td style="...">
 User Control Test Page
 </td>
 </tr>
 <tr>
 <td align="right" style="...">
 An Apress Creation © 2008
 </td>
 </tr>
</table>
```

Để kiểm tra thực hiện của user control, bạn cần gắn user control vào trong một web form bằng cách đặt user control vào tiên chỉ thị **Register** như sau:

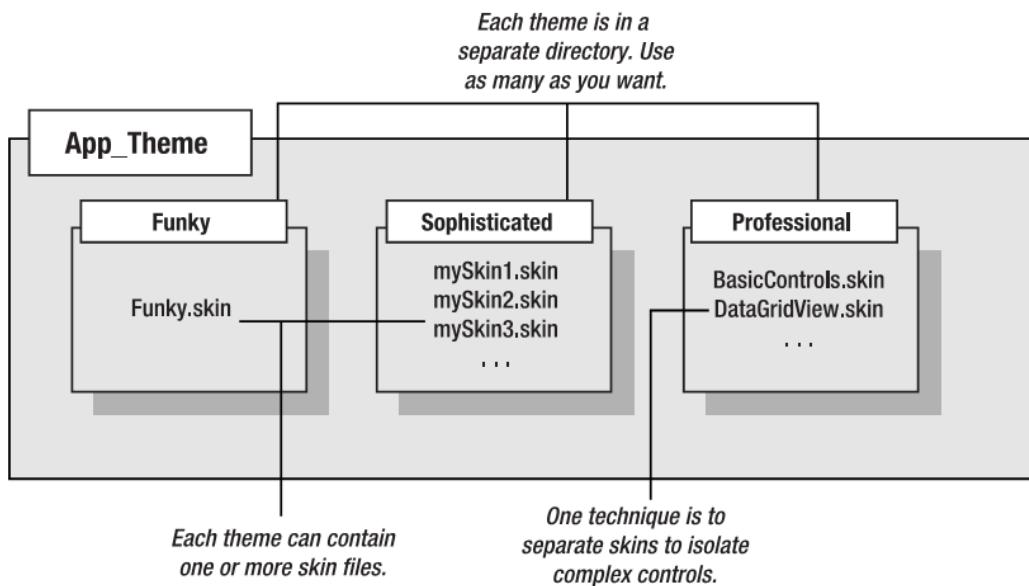
```
<%@ Register TagPrefix="apress" TagName="Header" Src="Header.ascx" %>

<%@ Page Language="C#" AutoEventWireup="true"
 CodeFile="HeaderTest.aspx.cs"
 Inherits="HeaderTest" %>
<%@ Register TagPrefix="apress" TagName="Header" Src="Header.ascx" %>
<html xmlns="http://www.w3.org/1999/xhtml">
 <head>
 <title>HeaderHost</title>
 </head>
 <body>
 <form id="Form1" method="post" runat="server">
 <apress:Header id="Header1" runat="server"></apress:Header>
 </form>
 </body>
</html>
```

## **II. Themes**

Giống CSS, themes cho phép bạn định nghĩa một loạt thuộc tính style để áp dụng vào trong các controls ở nhiều trang web. Tuy nhiên, không giống như CSS themes không được cài đặt bởi browser:

- Themes dựa trên control, không dựa trên thẻ Themes được áp dụng trên server
  - Themes có thẻ được áp dụng thông qua các file cấu hình
- Themes không cập nhật giống như CSS

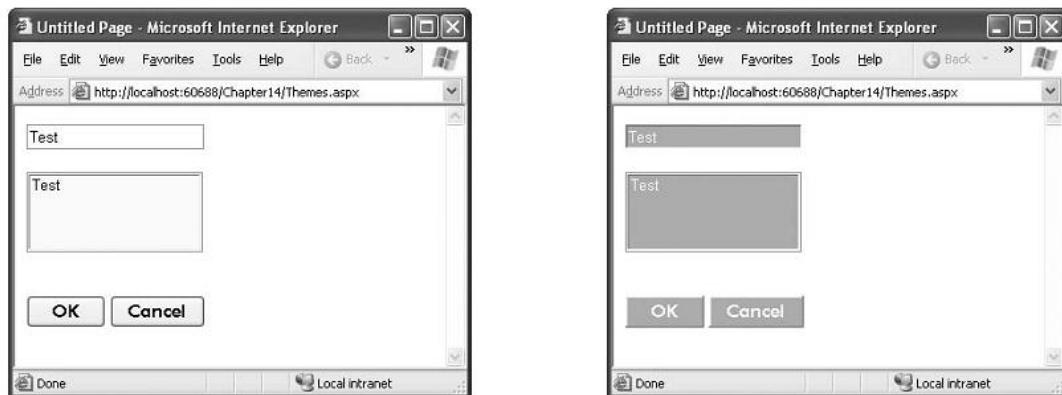


**Hình 50: Mô hình Theme và Skin**

### Áp dụng một Theme đơn giản

Để thêm một theme vào project, bạn chọn Website → Add New Item (or Project) → Add New Item, chọn Skin File. Visual Studio sẽ thông báo bạn biết thư mục để đặt file skin là App\_Themes.

```
<asp:ListBox runat="server" ForeColor="White" BackColor="Orange"/>
<asp:TextBox runat="server" ForeColor="White" BackColor="Orange"/>
<asp:Button runat="server" ForeColor="White" BackColor="Orange"/>
```



**Hình 51: Form kết quả áp dụng Theme**

### b. Tạo nhiều skins trên cùng một control

```
<asp:ListBox runat="server" ForeColor="White" BackColor="Orange" />
<asp:TextBox runat="server" ForeColor="White" BackColor="Orange" />
<asp:Button runat="server" ForeColor="White" BackColor="Orange" />
<asp:TextBox runat="server" ForeColor="White" BackColor="DarkOrange"
Font-Bold="True" SkinID="Dramatic" />
<asp:Button runat="server" ForeColor="White" BackColor="DarkOrange"
Font-Bold="True" SkinID="Dramatic" />
<asp:Button ID="Button1" runat="server" ... SkinID="Dramatic" />
```

### III. Master Page

- Chức năng để định nghĩa một phần của trang web tách rời và sử dụng lại nó trên nhiều trang khác nhau.
- Chức năng để tạo locked-in layout mà định nghĩa một khu vực có thể soạn thảo được. Các trang sử dụng template kiểu này thường ràng buộc với việc thêm vào hoặc chỉnh sửa nội dung trong khu vực được cho phép.
- Chức năng để cho phép một vài tùy biến về các thành phần được dùng trong mỗi trang.
- Chức năng để kết nối một trang với trang template rõ ràng (*Không có code*) hoặc kết nối đến một trang động vào lúc thực thi.
- Chức năng để thiết kế một trang có sử dụng các template với công cụ như Visual Studio.

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="SiteTemplate.master.cs" Inherits="SiteTemplate" %>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server"> <title> Untitled Page </title> </head>
<body>
<form id="form1" runat="server">
<div style="...">

 My Site

</div>
<asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
</asp:ContentPlaceHolder>
</form>
</body>
</html>
```

Vì vậy để tạo một trang nội dung đầy đủ có dùng trang master page là SiteTemplate, bạn chỉ cần thêm nội dung vào **ContentPlaceHolderID** là ContentPlaceHolder1. Ví dụ:

```
<%@ Page Language="C#" MasterPageFile("~/SiteTemplate.master"
 AutoEventWireup="true" CodeFile="SimpleContentPage.aspx.cs"
 Inherits="SimpleContentPage" Title="Content Page" %>
<asp:Content ID="Content1"
 ContentPlaceHolderID="ContentPlaceHolder1"
 runat="Server">
 Nội dung văn bản
</asp:Content>
```

### Câu hỏi/Bài tập củng cố:

1. Nêu sự khác biệt giữa user control và các web form bình thường khác.
2. Liên kết web form với user controls
3. Làm bài tập Thực hành Lab 7.1 và Lab 7.2 ở trang 145 bằng viết code
4. Làm bài tập Thực hành Lab 7.1 và Lab 7.2 ở trang 145 bằng công cụ wizard.

# Bài tập chương 7

## Lab 7.1. TẠO LAYOUT TRANG TIN TỨC

### 1. Mục tiêu

- Sử dụng Master page để tạo phần layout của trang web tin tức
- Sử dụng control AdRotator, để hiển thị thông tin quảng cáo
- Tạo các trang nội dung (content page) sử dụng master page đã thiết kế.

### 2. Yêu cầu:

Thiết kế trang web tin tức du lịch đơn giản có bố cục được mô tả như sau:

- Phần header: chứa banner của trang web
- Phần left: chứa menu chọn các mục tin
- Phần right: chứa các phần quảng cáo
- Phần footer: chứa thông tin bản quyền, thông tin trang web...

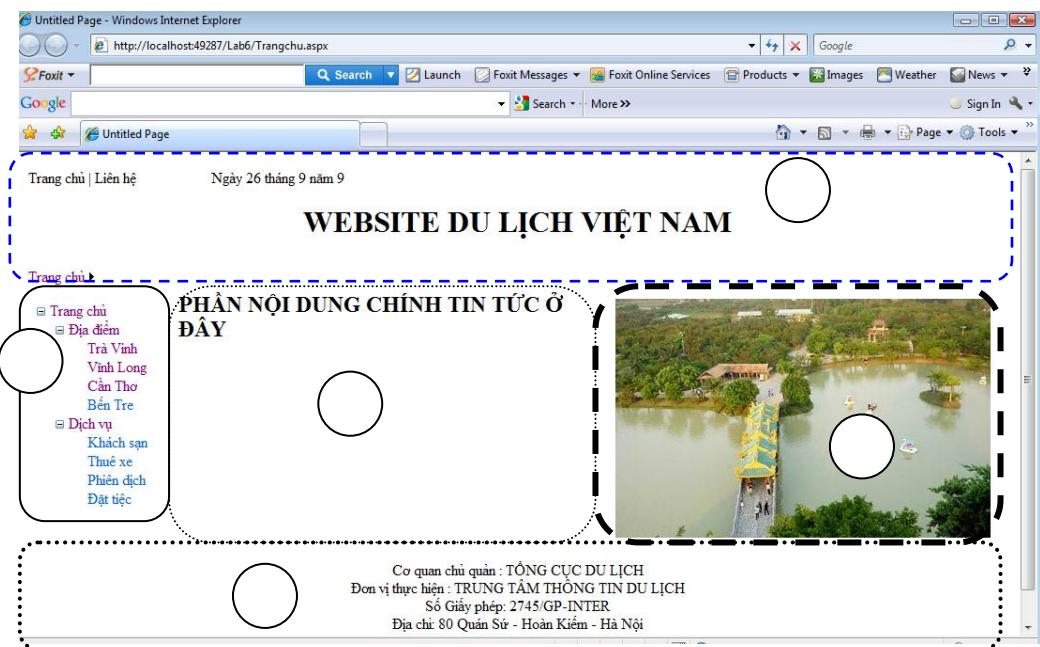


Các thành phần trên giao diện được mô tả (hình 2) trên

- **Phần 1:** chính là header, chứa một banner, bên dưới 2 hyperlink và thông tin ngày tháng năm hiện hành.
- **Phần 2:** chứa các mục nội dung tin tức
- **Phần 3:** chứa nội dung chính tin tức sẽ hiển thị ở đây
- **Phần 4:** chứa quảng cáo
- **Phần 5:** footer, thông tin bản quyền, thông tin của trang web...

### 3. Hướng dẫn:

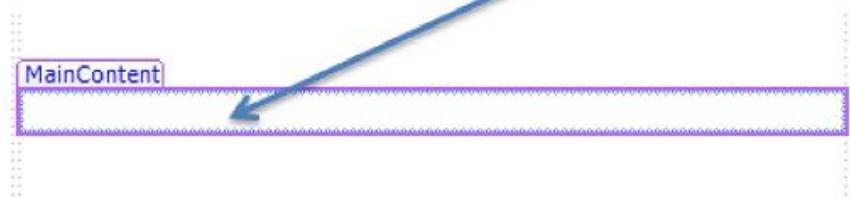
- **Bước 1:** Sinh viên tạo ứng dụng web có tên WebsiteDulich.
- **Bước 2:** Tạo master page cho ứng dụng trên
  - o Kích chuột phải vào project trong solution explorer, chọn Add -> New Item → Chọn Master Page.
- Thiết kế trang master theo mẫu



- Đặt một ContentPlaceHolder có ID là MainContent ở phần giữa: phần này dùng cho các trang content sẽ hiển thị nội dung riêng của nó.

```
<td>
 <asp:ContentPlaceHolder ID="MainContent" runat="server">

 </asp:ContentPlaceHolder>
</td>
```



- Thêm một Label vào bên phải của header, đặt tên là lblDate, label này sẽ hiển thị thông tin ngày tháng năm. Phần code được viết trong hàm Page\_Load của trang MasterPage

```
public partial class MasterPage : System.Web.UI.MasterPage
{
 protected void Page_Load(object sender, EventArgs e)
 {
 // hiển thị ngày tháng năm trên top
 DateTime date = DateTime.Now;
 string strDate = string.Format("Ngày {0} tháng {1} năm
{2}", date.Day, date.Month, date.Year);
 lblDate.Text = strDate;
 }
}
```

}

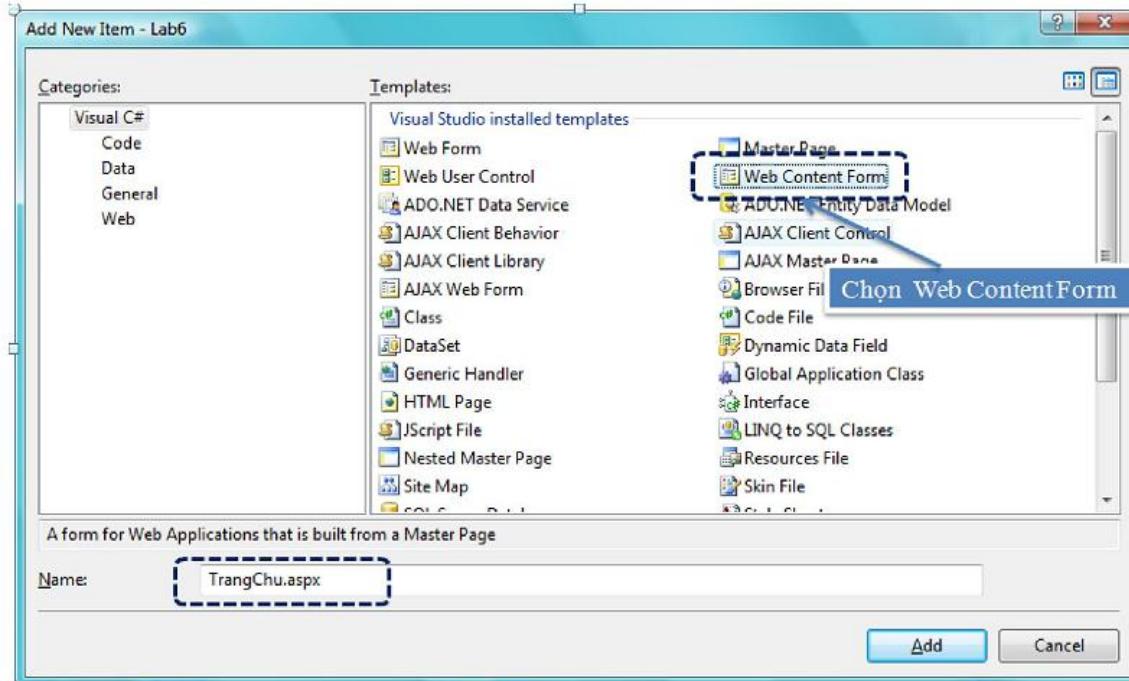
- **Bước 3:** Thêm một AdRotator control vào cột thứ 3 table con. AdRotator là control chứa các mẫu hình quảng cáo, mỗi khi trang refresh thì Adrotator tự động chuyển sang hình quảng cáo khác. Các mẫu quảng cáo này được khai báo trong một file XML, và thuộc tính AdvertisementFile sẽ link đến file XML này.

```
<Advertisements>
<Ad>
 <ImageUrl> Đường dẫn đến tập tin ảnh </ImageUrl>
 <NavigateUrl> Liên kết đến Website khi bấm chuột vào </NavigateUrl>
 <AlternateText> Dòng chữ hiện ra khi rê chuột vào </AlternateText>
 <Keyword> Từ khoá dùng để lọc ảnh </Keyword>
 <Impressions> Tần suất hiển thị của ảnh </Impressions>
</Ad>
</Advertisements>
```

- Ta tạo 2 mẫu quảng cáo và lưu trong file QuangCao.XML

```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
<Ad>
 <ImageUrl>images\MCC.jpg</ImageUrl>
 <NavigateUrl>http://www.mcc.dulich.vn</NavigateUrl>
 <AlternateText> Du lịch Mù Cang Chải </AlternateText>
 <Keyword> Vùng núi Tây Bắc </Keyword>
 <Impressions>80 </Impressions>
</Ad>
<Ad>
 <ImageUrl>images\LDS.jpg</ImageUrl>
 <NavigateUrl>http://www.quangnam.dulich.vn</NavigateUrl>
 <AlternateText> Du lịch Quảng Nam </AlternateText>
 <Keyword> Halong, Halong Bay, du lịch </Keyword>
 <Impressions>80 </Impressions>
</Ad>
</Advertisements>
```

4. **Bước 4:** Tạo trang Content có tên TrangChu.aspx có khai báo sử dụng master page trên



Chọn master page cho trang nội dung

Phần code của trang TrangChu.aspx như sau: phần nội dung chưa có, trong lab 7 tiếp theo ta sẽ làm phần nội dung đầy đủ.

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.Master"
 AutoEventWireup="true" CodeBehind="TrangChu.aspx.cs" Inherits="Lab6.tintuc"%>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server"></asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">
 <h2>PHẦN NỘI DUNG CHÍNH TIN TỨC Ở ĐÂY</h2>
</asp:Content>
```

## Lab 7.2: Tạo trang tóm tắt tin tức

### 1. Mục tiêu

- Sử dụng **Master page** để tạo phần layout của trang web tin tức
- Sử dụng **SQLDataSource** để kết nối với bảng dữ liệu tin tức trong SQL Server
- Sử dụng **DataList** để hiển thị tóm tắt các tin tức.

### 2. Nội dung

- Tiếp tục phát triển bài Lab 8.1 trong buổi trước, trong bài lab 8.2 chỉ xây dựng Layout của trang tin tức, cụ thể là phần master page, còn các trang nội dung từng chuyên mục thì chưa có nội dung.
  - Trong bài Lab 8.2. này chúng ta sẽ xây dựng trang **TinDulich.aspx**. Trang này chứa toàn bộ các tin tức tóm tắt của chuyên mục **Địa điểm**, trang này được gọi từ menu trái, mục **Địa điểm**.
  - Giao diện cuối cùng của trang **TinDulich.aspx** sẽ hiện như sau.



Hình 1: Giao diện của ứng dụng ở chuyên mục Dulich

### 3. Hướng dẫn

1. Mở lại project của bài **Thuchanh8.1** (đã làm xong phần master page của trang)
2. Tạo mới một cơ sở dữ liệu đơn giản chứa tin tức có tên là **WebsiteDulich** với một table có tên **BanTin** có cấu trúc như sau:

Column Name	Data Type	Allow Nulls
MaBantin	int	<input type="checkbox"/>
Tieude	nvarchar(255)	<input checked="" type="checkbox"/>
NDTomtat	text	<input checked="" type="checkbox"/>
Ngaydang	datetime	<input checked="" type="checkbox"/>
Hinhanh	nchar(50)	<input checked="" type="checkbox"/>
Theloi	nchar(2)	<input checked="" type="checkbox"/>

Tên trường	Kiểu	Điễn giải
MaBanTin	int, tự động tăng	Mã bản tin
TieuDe	nvarchar(255)	Tiêu đề của bạn tin
NDTomTat	nvarchar(255)	Nội dung tóm tắt của bản tin
NoiDung	ntext	Nội dung chính của bản tin
NgayDang	datetime	Ngày đăng tin
HinhAnh	nvarchar(50)	Chứa tên file minh họa trong bản tin
TheLoai	nchar(2)	

- Hình sau đây là tập một số mẫu tin minh họa cho bài thực hành. Trong thực tế, sinh viên có thể bổ sung vào thêm một số mẫu tin khác phù hợp theo các yêu cầu khác nhau của ứng dụng

	MaBantin	Tieude	NDTomtat	Ngaydang	Hinhanh	Theloi
	1	Du lịch Hà Nội - Quảng Ninh: Liên kết để phát triển	Hà N?i và Qu?ng Ninh là hai d?a phuon...	1/9/2011 12...	upload/1.jpg ...	1
	2	An Giang phát triển du lịch sinh thái nông nghiệp và s...	Nhi?m v? chính trong chu?ng trình phâ...	2/9/2011 12...	upload/2.jpg ...	2
	3	Tổ chức Cuộc thi trình diễn pháo hoa quốc tế Đà Nẵn...	UBND thành ph? Đà N?ng v?a th?ng n...	3/9/2011 12...	upload/dn.jpg ...	3
▶	4	L?n lung tr?i ng?m "mùa thu vàng" Mù Cang Chải	Khi nh?ng con mua r?ng dã thua th?t d...	4/9/2011 12...	upload/vc.jpg ...	4
	5	Xây dựng Khu bảo tồn biển Hòn Cau, Bình Thuận	Vu`a qua, t?i xã Vinh Tân (Tuy Phong)...	5/9/2011 12...	upload/bt.jpg ...	5
	6	Tây Ninh: Long Điền Sơn - điểm du lịch đặc đáo của v...	Long Đi?n Sơn t?a l?c t?i ?p Ninh Trung...	6/9/2011 12...	upload/tn.jpg ...	3
*	NULL	NULL	NULL	NULL	NULL	NULL

3. Tạo trang tóm tắt tin: Dulich.aspx khai báo sử dụng master page.

4. Tạo DataSource cho trang web:

```
<asp:SqlDataSource ID="sdsSourceBantin" runat="server"
 ConnectionString="<%$ConnectionString="TintucDulichcnstr %>
 SelectCommand="SELECT * FROM [Bantin]">>
</asp:SqlDataSource>
```

2. Tạo một DataList có sử dụng kết nối với Nguồn dữ liệu như sau:

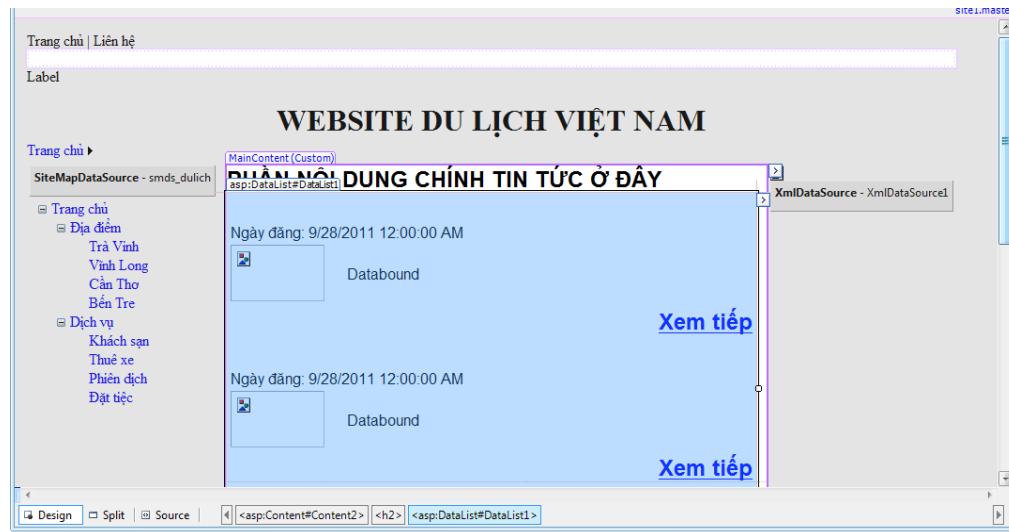
```
<asp:DataList ID="DataList1" runat="server" DataSourceID="SqlDataSource2"
 onselectedindexchanged="DataList1_SelectedIndexChanged" Width="564px">
 <ItemTemplate>
 <table align="left" style="width: 100%;">
 <tr>
 <td colspan="2" class="style6" >
 Ngày đăng: <asp:Label ID="Label1" runat="server"
 Text='<%# Eval("NgayDang") %>'></asp:Label>
```

```

 </td>
 </tr>
 <tr>
 <td>
 <asp:Image ID="Image1" Width="100px" Height="60px"
 runat="server" ImageUrl='<%# Eval("Hinhanh") %>' />
 </td>
 <td>
 <asp:HyperLink ID="HyperLink1" runat="server"
 Text='<%# Eval("NDTomtat") %>'></asp:HyperLink>
 </td>
 </tr>
 <tr>
 <td colspan=2 align=right> <asp:HyperLink ID="Xemtiep" runat="server"
NavigateUrl="~/Trangchu.aspx">Xem tiếp</asp:HyperLink></td>
 </tr>
</table>
</ItemTemplate>
</asp:DataList>

```

Màn hình thiết kế có dạng như sau:



3. Chạy thử nghiệm trang web và xem kết quả.

## TÀI LIỆU THAM KHẢO

### ❖ TÀI LIỆU THAM KHẢO ĐỀ BIÊN SOẠN NỘI DUNG MÔN HỌC:

- [1]. **Pro Asp Net 3.5 In C# 2008, Second Edition** , Tác giả: Matthew MacDonald và Mario Szpuszta, NXB: 2008
- [2.] **Beginning Asp.Net 3.5 In C# 2008, Second Edition** , Tác giả: Matthew MacDonald và Mario Szpuszta, NXB: 2008
- [3]. **OReilly-Programming C#, 2Ed**, Tác giả: Jesse Liberty ; NXB: 2002
- [4]. **Silde bài giảng Lập trình Ứng dụng Web với ASP.net;** Trung tâm Tin học Đại học Khoa học tự nhiên; NXB: 2009
- [5]. **Bài tập thực hành ASP.net;** Trung tâm Tin học Trường Đại học Khoa học tự nhiên; NXB: 2009

### ❖ TÀI LIỆU THAM KHẢO ĐỀ NGHỊ CHO HỌC VIÊN:

- [1]. Learning ASP.net Địa chỉ: [www.asp.net](http://www.asp.net)
- [2]. ASP.net Tutorial: Địa chỉ: [www.w3schools.com/ASPNET](http://www.w3schools.com/ASPNET)
- [3]. Java2S: Địa chỉ: <http://www.java2s.com/>