ỦY BAN NHÂN DÂN TỈNH LÂM ĐỒNG

**TRƯỜNG CAO ĐẲNG ĐÀ LẠT**

**GIÁO TRÌNH**

**MÔN HỌC/ MÔ ĐUN: QUẢN TRỊ VÀ**

**THIẾT KẾ WEBSITE THƯƠNG MẠI**

**NGÀNH/ NGHỀ: THƯƠNG MẠI ĐIỆN TỬ**

**TRÌNH ĐỘ: CAO ĐẲNG**

**(LƯU HÀNH NỘI BỘ)**

**Lâm Đồng, năm 2019**

***Lâm Đồng, năm 2018***

**TUYÊN BỐ BẢN QUYỀN**

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

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

Giáo trình được lưu hành nội bộ Trường Cao đẳng Đà Lạt.

**LỜI GIỚI THIỆU**

*Vài nét về xuất xứ giáo trình:*

Giáo trình này được viết theo căn cứ Thông tư số 03/2017/TT-BLĐTBXH ngày 01 tháng 03 năm 2017 của Bộ Lao động – Thương binh và Xã hội về việc Quy định về quy trình xây dựng, thẩm định và ban hành chương trình; tổ chức biên soạn, lựa chọn, thẩm định giáo trình đào tạo trình độ trung cấp, trình độ cao đẳng.

*Quá trình biên soạn:*

Giáo trình này được biên soạn có sự tham gia tích cực của các giáo viên có kinh nghiệm, cùng với những ý kiến đóng góp quý báu của các chuyên gia về lĩnh vực công nghệ thông tin.

*Mối quan hệ của tài liệu với chương trình, mô đun/môn học:*

Căn cứ vào chương trình đào tạo nghề Thiết kế trang web, giáo trình giúp cung cấp cho người học những kiến thức cũng như kỹ năng thiết kế website thương mại điện tử.

Để học được mô đun này người học cần có kiến thức cơ bản về tin học và toán học.

*Cấu trúc chung của giáo trình này bao gồm 6 chương:*

Chương 1. LÝ THUYẾT THƯƠNG MẠI ĐIỆN TỬ

Chương 2. NGÔN NGỮ ĐỊNH DẠNG SIÊU VĂN BẢN HTML

CHƯƠNG 3. ASP.NET (ACTIVE SERVER PAGE .NET)

Chương 4. CÁC ĐIỀU KHIỂN LIÊN KẾT DỮ LIỆU

Chương 5. QUẢN LÝ TRẠNG THÁI

Chương 6. BẢO MẬT ỨNG DỤNG WEB

*Lời cảm ơn*

Giáo trình được biên soạn trên cơ sở các văn bản quy định của Nhà nước và tham khảo nhiều tài liệu liên quan có giá trị. Song chắc hẳn quá trình biên soạn không tránh khỏi những thiếu sót nhất định. Ban biên soạn mong muốn và thực sự cảm ơn những ý kiến nhận xét, đánh giá của các chuyên gia, các thầy cô đóng góp cho việc chỉnh sửa để giáo trình ngày một hoàn thiện hơn.

*Lâm Đồng, ngày 10 tháng 12 năm 2019*

Tham gia biên soạn

1. Phạm Đình Nam

2. Ngô Thiên Hoàng

3. Nguyễn Quỳnh Nguyên

4. Phan Ngọc Bảo

**NỘI DUNG**

Chương 1. LÝ THUYẾT THƯƠNG MẠI ĐIỆN TỬ................................................................................6

1.1.THƯƠNG MẠI ĐIỆN TỬ LÀ GÌ?............................................................................................ 7

1.1.1. Khái niệm thương mại điện tử.........................................................................................................7

1.1.2.Các đặc trưng của thương mại điện tử.............................................................................................7

1.1.3.Cơ sở để phát triển thương mại điện tử...........................................................................................8

1.2.LỢI ÍCH CỦA THƯƠNG MẠI ĐIỆN TỬ............................................................................... 8

1.2.1.Thu thập được nhiều thông tin.........................................................................................................8

1.2.2.Giảm chi phí sản xuất........................................................................................................................9

1.2.3.Giảm chi phí bán hàng, tiếp thị và giao dịch.....................................................................................9

1.2.4.Xây dựng quan hệ với đối tác...........................................................................................................9

1.2.5.Tạo điều kiện sớm tiếp cận kinh tế trí thức......................................................................................9

1.3.CÁC LOẠI HÌNH GIAO DỊCH THƯƠNG MẠI ĐIỆN TỬ..................................................... 10

1.4.CÁC HÌNH THỨC HOẠT ĐỘNG CHỦ YẾU CỦA TMĐT..........................................................11

1.4.1.Thư điện tử.....................................................................................................................................11

1.4.2.Thanh toán điện tử.........................................................................................................................11

1.4.3.Trao đổi dữ liệu điện tử..................................................................................................................12

1.4.4.Truyền nội dung số hóa..................................................................................................................13

1.4.5.Mua bán hàng hóa hữu hình...........................................................................................................14

Chương 2. NGÔN NGỮ ĐỊNH DẠNG SIÊU VĂN BẢN HTML........................................................ 16

2.1.MÔ HÌNH WEB CLIENT/SERVER................................................................................................17

2.2.NGÔN NGỮ ĐỊNH DẠNG SIÊU VĂN BẢN HTML..................................................................... 18

2.2.1.Cấu trúc cơ bản của trang HTML....................................................................................................18

2.2.2.Cú pháp chi tiết của tag HTML.......................................................................................................19

2.3.ĐỊNH DẠNG CƠ BẢN.................................................................................................................... 20

2.3.1.Các thao tác khối............................................................................................................................20

2.3.2.Định dạng ký tự..............................................................................................................................21

2.3.3.Định dạng đoạn..............................................................................................................................21

2.3.4.Chèn đường kẻ ngang.....................................................................................................................23

2.3.5.Chèn hình ảnh.................................................................................................................................23

2.3.6.Tìm và thay đổi dữ liệu...................................................................................................................24

2.4.LIÊN KẾT (HYPERLINK)..............................................................................................................25

2.4.1.Liên kết đến một trang web khác...................................................................................................25

2.4.2.Cách tạo một email-link..................................................................................................................26

2.4.3.Liên kết ImageMap.........................................................................................................................26

2.5.TẠO FRAMESET, TABLE............................................................................................................ 27

2.5.1.Frameset.........................................................................................................................................27

2.5.2.Table...............................................................................................................................................31

2.6.ĐỊNH DẠNG CẢI TIẾN VỚI CSS................................................................................................. 35

2.6.1.Style Sheets....................................................................................................................................35

2.6.2.Định dạng Inline..............................................................................................................................36

2.6.3.Định dạng nhúng.............................................................................................................................36

2.6.4.Định dạng bên ngoài.......................................................................................................................37

2.7.TẠO FORM.......................................................................................................................................38

2.7.1.Tạo form.........................................................................................................................................38

2.7.2.Các đối tượng trên form.................................................................................................................39

CHƯƠNG 3. ASP.NET (ACTIVE SERVER PAGE .NET)...................................................................41

3.1.TỔNG QUAN VỀ LẬP TRÌNH ỨNG DỤNG WEB...................................................................... 42

3.1.1.Mô hình ứng dụng 2 lớp.................................................................................................................42

3.1.2.Mô hình ứng dụng 3 lớp.................................................................................................................43

3.2.NGÔN NGỮ C#...................................................................................................................................44

3.2.1.Kiểu dữ liệu.....................................................................................................................................44

3.2.2.Chuyển đổi các kiểu dữ liệu............................................................................................................46

3.3.GIỚI THIỆU VỀ ASP.NET?.......................................................................................................... 46

3.3.1.Tìm hiểu về .Net Platform...............................................................................................................46

3.3.2.Tìm hiểu về .Net Framework...........................................................................................................47

3.2.3.Tìm hiểu về ASP.Net.......................................................................................................................51

3.3.NHỮNG ƯU ĐIỂM CỦA ASP.NET.................................................................................................52

3.4.QUÁ TRÌNH XỬ LÍ TẬP TIN ASPX...............................................................................................53

3.5.CÁCH THỨC HOẠT ĐỘNG CỦA ASP.NET TRÊN IIS............................................................ 53

3.5.1.Internet Information Services (IIS)..................................................................................................53

3.5.2.Cài đặt Web Server.........................................................................................................................54

3.5.3.Kiểm tra kết quả cài đặt..................................................................................................................55

3.6.TẠO TRANG WEB ASP.NET ĐẦU TIÊN..................................................................................... 56

3.6.1.Tạo trang web ASP.NET đầu tiên....................................................................................................56

3.6.2.Phân loại tập tin trong ASP.NET......................................................................................................59

3.7.LÀM VIỆC VỚI CÁC ASP.NET SERVER CONTROL.................................................................. 60

3.7.1.HTML Controls...............................................................................................................................60

3.7.2.Web Server Controls.......................................................................................................................62

3.7.3.Validation Controls.........................................................................................................................69

3.7.4.User Controls..................................................................................................................................78

Chương 4. CÁC ĐIỀU KHIỂN LIÊN KẾT DỮ LIỆU............................................................................80

4.1. ĐIỀU KHIỂN GRIDVIEW..............................................................................................................81

4.1.1.Load dữ liệu lên GridView...............................................................................................................81

4.1.2.Các thao tác định dạng lưới............................................................................................................84

4.1.3.Xử lí sắp xếp....................................................................................................................................89

4.1.4.Tùy biến các cột (đọc thêm)............................................................................................................89

* 1. 5.Cập nhật dữ liệu trực tiếp trên lưới (tham khảo)...........................................................................92
  2. ĐIỀU KHIỂN DATALIST.........................................................................................................................94

4.2.1.Sử dụng DataList hiển thị dữ liệu....................................................................................................94

4.2.2.Cập nhật dữ liệu với DataList (tham khảo).....................................................................................96

Chương 5. QUẢN LÝ TRẠNG THÁI.................................................................................................. 98

5.1.ĐỐI TƯỢNG REQUEST, RESPONSE........................................................................................ 100

5.1.1.Đối tượng Response.....................................................................................................................100

5.1.2.Đối tượng Request........................................................................................................................101

5.2.ĐỐI TƯỢNG VIEWSTATE.......................................................................................................... 102

5.3.ĐỐI TƯỢNG SESSION, APPLICATION................................................................................. 103

5.3.1.Đối tượng Application...................................................................................................................103

5.3.2.Đối tượng Session.........................................................................................................................104

5.4.ĐỐI TƯỢNG SERVER................................................................................................................. 105

5.5.ĐỐI TƯỢNG COOKIES............................................................................................................ 105

5.5.1.Giới thiệu......................................................................................................................................105

5.5.2.Làm việc với Cookies...................................................................................................................106

Chương 6. BẢO MẬT ỨNG DỤNG WEB.......................................................................................... 108

6.1.GIỚI THIỆU SƠ LƯỢC CÁC KỸ THUẬT TẤN CÔNG ỨNG DỤNG WEB............................ 109

6.1.1.Kiểm soát truy cập Web................................................................................................................109

6.1.2.Chiếm hữu phiên làm việc............................................................................................................109

6.1.3.Lợi dụng các thiếu sót trong việc kiểm tra giá trị nhập hợp lệ......................................................109

6.2.CÁC KỸ THUẬT TẤN CÔNG VÀ BẢO MẬT ỨNG DỤNG WEB.......................................... 110

6.2.1.Thao tác trên tham số truyền.......................................................................................................110

6.2.2.Thao tác trên Biến ẩn Form..........................................................................................................111

6.2.3.Thao tác trên Cookie.....................................................................................................................112

6.3.BẢO MẬT CƠ SỞ DỮ LIỆU (đọc thêm)....................................................................................... 113

6.3.1.Các kỹ thuật tấn công cơ sở dữ liệu...............................................................................................113

6.3.2.Cách phòng chống.........................................................................................................................116

# Chương 1

**LÝ THUYẾT THƯƠNG MẠI ĐIỆN TỬ**

## MỤC TIÊU

1. Về kiến thức
   * Khái niệm về thương mại điện tử
   * Lợi ích của thương mại điện tử
   * Các hình thức thương mại điện tử qua mạng
2. Về nội dung

− Các loại hình giao dịch thương mại điện tử

− Các hình thức hoạt động chủ yếu của thương mại điện tử

## HÌNH THỨC VÀ PHƯƠNG PHÁP GIẢNG DẠY

* Trình chiếu Powerpoint, kết hợp dùng bảng
* Đặt vấn đề, trao đổi

## TÀI LIỆU THAM KHẢO

1. Nguyễn Đăng Hậu (2004), Kiến thức thương mại điện tử, Viện Đào tạo Công nghệ và Quản lý quốc tế - Khoa Công nghệ thông tin.
2. Hà Thành – Trí Việt (2009), Làm quen với Internet

**1.1.THƯƠNG MẠI ĐIỆN TỬ LÀ GÌ?**

### 1.1.1. Khái niệm thương mại điện tử

Thương mại điện tử là hình thức mua bán hàng hóa và dịch vụ thông qua mạng máy tính toàn cầu.

Phạm vi của Thương mại điện tử rất rộng, bao quát hầu hết các lĩnh vực hoạt động kinh tế, việc mua bán hàng hóa và dịch vụ chỉ là một trong hàng ngàn lĩnh vực áp dụng của Thương mại điện tử. Theo nghĩa hẹp thương mại điện tử chỉ gồm các hoạt động thương mại được tiến hành trên mạng máy tính mở như Internet. Trên thực tế, chính các hoạt động thương mại thông qua mạng Internet đã làm phát sinh thuật ngữ Thương mại điện tử.

Thương mại điện tử gồm các hoạt động mua bán hàng hóa và dịch vụ qua phương tiện điện tử, giao nhận các nội dung kỹ thuật số trên mạng, chuyển tiền điện tử, mua bán cổ phiếu điện tử, vận đơn điện tử, đấu giá thương mại, hợp tác thiết kế, tài nguyên mạng, mua sắm công cộng, tiếp thị trực tuyến với người tiêu dùng và các dịch vụ sau bán hàng.

Thương mại điện tử được thực hiện đối với cả thương mại hàng hóa (ví dụ như hàng tiêu dùng, các thiết bị y tế chuyên dụng) và thương mại dịch vụ (ví dụ như dịch vụ cung cấp thông tin, dịch vụ pháp lý, tài chính); các hoạt động truyền thống (như chăm sóc sức khỏe, giáo dục) và các hoạt động mới (ví dụ như siêu thị ảo). Thương mại điện tử đang trở thành một cuộc cách mạng làm thay đổi cách thức mua sắm của con người.

### 1.1.2.Các đặc trưng của thương mại điện tử

So với các hoạt động thương mại truyền thống, thương mại điện tử có một số điểm khác biệt cơ bản sau:

* Các bên tiến hành giao dịch trong thương mại điện tử không tiếp xúc trực tiếp với nhau và không đòi hỏi phải biết nhau từ trước.
* Các giao dịch thương mại truyền thống được thực hiện với sự tồn tại của khái niệm biến giới quốc gia, còn thương mại điện tử được thực hiện trong một thị trường không có biên giới (thị trường thống nhất toàn cầu). Thương mại điện tử trực tiếp tác động đến môi trường cạnh tranh toàn cầu.
* Trong mọi hoạt động giao dịch thương mại điện tử đều có sự tham gia của ít nhất ba chủ thể, trong đó có một bên không thể thiếu được đó là người cung cấp dịch vụ mạng, các cơ quan chứng thực.
* Đối với thương mại truyền thống thì mạng lưới thông tin chỉ là phương tiện để trao đổi dữ liệu, còn đối với thương mại điện tử thì mạng lưới thông tin chính là thị trường.

### 1.1.3.Cơ sở để phát triển thương mại điện tử

Để phát triển TMĐT cần phải có hội đủ một số cơ sở:

* Hạ tầng kỹ thuật Internet phải đủ nhanh, mạnh đảm bảo truyền tải các nội dung thông tin bao gồm âm thanh, hình ảnh trung thực và sống động. Một hạ tầng internet mạnh cho phép cung cấp các dịch vụ như xem phim, xem TV, nghe nhạc, v.v.. trực tiếp. Chi phí kết nối internet phải rẻ để đảm bảo số người dùng internet phải lớn.
* Hạ tầng pháp lý: phải có luật về thương mại điện tử công nhận tính pháp lý của các chứng từ điện tử, các hợp đồng điện tử ký qua mạng; phải có luật bảo vệ quyền sở hữu trí tuệ, bảo vệ sự riêng tư, bảo vệ người tiêu dùng,v.v.. để điều chỉnh các giao dịch qua mạng.
* Phải có cơ sở thanh toán điện tử an toàn, bảo mật. Thanh toán điện tử qua thẻ, qua tiền điện tử, thanh toán qua EDI. Các ngân hàng phải triển khai hệ thống thanh toán điện tử rộng khắp.
* Phải có hệ thống an toàn bảo mật cho các giao dịch, chống xâm nhập trái phép, chống virus,…
* Phải có nhân lực am hiểu kinh doanh, công nghệ thông tin, thương mại điện tử để triển khai tiếp thị, quảng cáo, xúc tiến, bán hàng và thanh toán qua mạng.

## 1.2.LỢI ÍCH CỦA THƯƠNG MẠI ĐIỆN TỬ

### 1.2.1.Thu thập được nhiều thông tin

Thương mại điện tử giúp người tham gia thu được nhiều thông tin về thị trường, đối tác, giảm chi phí tiếp thị và giao dịch, rút ngắn thời gian sản xuất, tạo dựng và củng cố quan hệ bạn hàng. Các doanh nghiệp nắm được thông tin phong phú về kinh tế thị trường, nhờ đó có thể xây dựng được chiến lược sản xuất và kinh doanh thích hợp với xu thế phát triển của thị trường trong nước, khu vực và quốc tế. Điều này đặc biệt có ý nghĩa đối với các doanh nghiệp vừa và nhỏ, hiện nay đang được nhiều nước quan tâm, coi là một trong những động lực phát triển kinh tế.

### 1.2.2.Giảm chi phí sản xuất

Thương mại điện tử giúp giảm chi phí sản xuất, trước hết là chi phí văn phòng. Các văn phòng không giấy tờ chiếm diện tích nhỏ hơn rất nhiều, chi phí tìm kiếm chuyển giao tài liệu giảm nhiều lần (trong đó khâu in ấn hầu như được bỏ hẳn). Điều quan trọng hơn, với góc độ chiến lược, các nhân viên có năng lực được giải phóng khỏi nhiều công đoạn sự vụ có thể tập trung vào nghiên cứu phát triển, sẽ đưa đến những lợi ích to lớn lâu dài.

### 1.2.3.Giảm chi phí bán hàng, tiếp thị và giao dịch

Thương mại điện tử giúp giảm thấp chi phí bán hàng và chi phí tiếp thị. Bằng phương tiện Internet/Web, một nhân viên bán hàng có thể giao dịch được với rất nhiều khách hàng, catalogue điện tử trên Web phong phú hơn nhiều và thường xuyên cập nhật so với catalogue in ấn chỉ có khuôn khổ giới hạn.

Thương mại điện tử qua Internet/Web giúp người tiêu thụ và các doanh nghiệp giảm đáng kể thời gian và chi phí giao dịch (giao dịch được hiểu là từ quá trình quảng cáo, tiếp xúc ban đầu, giao dịch đặt hàng, giao dịch thanh toán). Thời gian giao dịch qua Internet chỉ bằng 7% giao dịch qua Fax, và bằng khoảng 0.5 phần nghìn thời gian giao dịch qua bưu điện chuyển phát nhanh, chi phí thanh toán điện tử qua Internet chỉ bằng từ 10% đến 20% chi phí thanh toán theo lối thông thường.

Tổng hợp tất cả các lợi ích trên, chu trình sản xuất được rút ngắn, nhờ đó sản phẩm mới xuất hiện nhanh và hoàn thiện hơn.

### 1.2.4.Xây dựng quan hệ với đối tác

Thương mại điện tử tạo điều kiện cho việc thiết lập và củng cố mối quan hệ giữa các thành viên tham gia vào quá trình thương mại: thông qua mạng (Internet/Web) các thành viên tham gia (người tiêu thụ, doanh nghiệp, các cơ quan Chính phủ…) có thể giao tiếp trực tiếp và liên tục với nhau, có cảm giác như không có khoảng cách về địa lý và thời gian nữa; nhờ đó sự hợp tác và sự quản lý đều được tiến hành nhanh chóng một cách liên tục: các bạn hàng mới, các cơ hội kinh doanh mới được phát hiện nhanh chóng trên phạm vi toàn quốc, toàn khu vực, toàn thế giới, và có nhiều cơ hội để lựa chọn hơn.

### 1.2.5.Tạo điều kiện sớm tiếp cận kinh tế trí thức

Trước hết, thương mại điện tử sẽ kích thích sự phát triển của ngành công nghệ thông tin tạo cơ sở cho phát triển kinh tế tri thức. Lợi ích này có một ý nghĩa lớn đối với các nước đang phát triển: nếu không nhanh chóng tiếp cận nền kinh tế tri thức thì sau khoảng một thập kỷ nữa, các nước đang phát triển có thể bị bỏ rơi hoàn toàn. Khía cạnh lợi ích này mang tính chiến lược công nghệ và tính chính sách phát triển cần cho các nước công nghiệp hóa.

## 1.3.CÁC LOẠI HÌNH GIAO DỊCH THƯƠNG MẠI ĐIỆN TỬ

Trong thương mại điện tử có ba chủ thể tham gia: Doanh nghiệp (B) giữ vai trò động lực phát triển thương mại điện tử, người tiêu dùng (C) giữ vai trò quyết định sự thành công của thương mại điện tử và chính phủ (G) giữ vai trò định hướng, điều tiếp và quản lý. Từ các mối quan hệ của các chủ thể trên, ta có ba loại giao dịch thương mại điện tử: B-B, B-C, B-G, C-G, C-C…trong đó B-B và B-C là hai loại hình giao dịch thương mại điện tử quan trọng nhất.

**Business to Business (B-B):** Mô hình thương mại điện tử giữa các doanh nghiệp

Thương mại điện tử B-B là việc thực hiện các giao dịch giữa các doanh nghiệp với nhau trên mạng. Ta thường gọi là giao dịch B-B. Các bên tham gia giao dịch B-B gồm: người trung gian trực tuyến, người mua và người bán. Các loại giao dịch B-B bao gồm: mua ngay theo yêu cầu khi giá cả thích hợp và mua theo hợp đồng dài hạn, dựa trên đàm phán cá nhân giữa người mua và người bán.

Các loại giao dịch B-B cơ bản:

* Bên bán (một bên bán nhiều bên mua) là mô hình dựa trên công nghệ web trong đó một công ty bán cho nhiều công ty mua. Có ba phương pháp bán trực tiếp trong mô hình này: Bán từ catalogue điện tử, Bán qua quá trình đấu giá, Bán theo hợp đồng cung ứng dài hạn đã thỏa thuận trước. Công ty bán có thể là nhà sản xuất hoặc nhà trung gian thông thường là nhà phân phối hay đại lý.
* Bên mua - một bên mua nhiều bên bán
* Sàn giao dịch – nhiều bên bán - nhiều bên mua
* Thương mại điện tử phối hợp – các đối tác phối hợp với nhau ngay trong quá trình thiết kế chế tạo sản phẩm

**Business to Consumer (B-C):** Mô hình thương mại điện tử giữa doanh nghiệp và người tiêu dùng

Đây là mô hình bán lẻ trực tiếp đến người tiêu dùng. Trong thương mại điện tử, bán lẻ điện tử có thể từ nhà sản xuất, hoặc từ một cửa hàng thông qua kênh phân phối. Hàng hóa bán lẻ trên mạng thường là hàng hóa, máy tính, đồ điện tử, dụng cụ thể thao, đồ dùng văn phòng, sách và âm nhạc, đồ chơi, sức khỏe và mỹ phẩm, giải trí, v.v..

Hai loại giao dịch trên là giao dịch cơ bản của thương mại điện tử. Ngoài ra, trong thương mại điện tử người ta còn sử dụng các loại giao dịch: Government to Business (G-B) là mô hình thương mại điện tử giữa doanh nghiệp với cơ quan chính phủ , Government to Citizens (G-C) là mô hình TMĐT giữa các cơ quan chính phủ và công dân hay còn gọi là chính phủ điện tử, Consumer to Consumer (C-C) là mô hình TMĐT giữa các người tiêu dùng và mobile commerce là TMĐT qua điện thoại di động.

## 1.4.CÁC HÌNH THỨC HOẠT ĐỘNG CHỦ YẾU CỦA TMĐT

### 1.4.1.Thư điện tử

Các doanh nghiệp, các cơ quan Nhà nước,.. sử dụng thư điện tử để gửi thư cho nhau một cách “trực tuyến” thông qua mạng, gọi là thư điện tử (e\_mail). Thông tin trong thư điện tử không phải tuân theo một cấu trúc định trước nào.

### 1.4.2.Thanh toán điện tử

Thanh toán điện tử (electronic payment) là việc thanh toán tiền thông qua thông điệp điện tử (electronic message). Ví dụ: trả lương bằng cách chuyển tiền trực tiếp vào tài khoản, trả tiền mua hàng bằng thẻ mua hàng, thẻ tín dụng,v.v.. thực chất đều là dạng thanh toán điện tử. Ngày nay, với sự phát triển của TMĐT, thanh toán điện tử đã mở rộng sang các lĩnh vực mới đó là:

* Trao đổi dữ liệu điện tử tài chính (Financial Electronic Data Interchange, gọi tắt là FEDI) chuyên phục vụ cho việc thanh toán điện tử giữa các công ty giao dịch với nhau bằng điện tử.
* Tiền lẻ điện tử (Internet Cash) là tiền mặt được mua từ một nơi phát hành (ngân hàng hoặc một tổ chức tín dụng nào đó), sau đó được chuyển đổi tự do sang các đồng tiền khác thông qua Internet, áp dụng trong cả phạm vi một nước cũng như giữa các quốc gia; tất cả đều được thực hiện bằng kỹ thuật số hóa, vì thế tiền mặt này còn có tên gọi là “tiền mặt số hóa” (digital cash. Tiền lẻ điện tử đang trên đà phát triển nhanh, nó có ưu điểm nổi bật sau:

+ Dùng để thanh toán những món hàng giá trị nhỏ, thậm chí ngay cả tiền mua báo (vì phí giao dịch mua hàng và chuyển tiền rất thấp);

+ Có thể tiến hàng giữa hai con người hoặc hai công ty bất kỳ, các thanh toán là vô danh;

+ Tiền mặt nhận được đảm bảo là tiền thật, tránh được tiền giả

* Ví điện tử (electronic purse); là nơi để tiền mặt Internet, chủ yếu là thẻ thông minh (smart card), còn gọi là thẻ giữ tiền (stored value card), tiền được trả cho bất kỳ ai đọc được thẻ đó; kỹ thuật của túi tiền điện tử tương tự như kỹ thuật áp dụng cho “tiền lẻ điện tử”. Thẻ thông minh, nhìn bề ngoài như thẻ tín dụng, nhưng ở mặt sau của thẻ, có một chíp máy tính điện tử có một bộ nhớ để lưu trữ tiền số hóa, tiền ấy chỉ được “chi trả” khi sử dụng hoặc thư yêu cầu (như xác nhận thanh toán hóa đơn) được xác thực là “đúng”.

− Giao dịch điện tử của ngân hàng (digital banking). Hệ thống thanh toán điện tử của ngân hàng là một hệ thống lớn gồm nhiều hệ thống nhỏ:

* + Thanh toán giữa ngân hàng với khách hàng qua điện thoại, tại các điểm bán lẻ, các kiôt, giao dịch cá nhân tại các gia đình, giao dịch tại trụ sở khách hàng, giao dịch qua Internet, chuyển tiền điện tử, thẻ tín dụng, thông tin hỏi đáp…,
  + Thanh toán giữa ngân hàng với các đại lý thanh toán (nhà hàng, siêu thị…,)
  + Thanh toán nội bộ một hệ thống ngân hàng
  + Thanh toán liên ngân hàng

### 1.4.3.Trao đổi dữ liệu điện tử

Trao đổi dữ liệu điện tử (Electronic Data Interchange - EDI) là việc trao đổi các dữ liệu dưới dạng “có cấu trúc”, từ máy tính điện tử này sang máy tính điện tử khác, giữa các công ty hoặc đơn vị đã thỏa thuận buôn bán với nhau.

Theo Ủy ban liên hợp quốc về luật thương mại quốc tế (UNCITRAL), “Trao đổi dữ liệu điện tử (EDI) là việc chuyển giao thông tin từ máy tính điện tử này sang máy tính điện tử khác bằng phương tiện điện tử, có sử dụng một tiêu chuẩn đã được thỏa thuận để cấu trúc thông tin”.

EDI ngày càng được sử dụng rộng rãi trên toàn cầu, chủ yếu phục vụ cho việc mua và phân phối hàng (gửi đơn hàng, các xác nhận, các tài liệu gửi hàng, hóa đơn,v.v..), người ta cũng dùng cho các mục đích khác, như thanh toán tiền khám bệnh, trao đổi các kết quả xét nghiệm v.v..

Trước khi có Internet đã có EDI, khi đó người ta dùng “mạng giá trị gia tăng” (Value Added Network, viết tắt là VAN) để liên kết các đối tác EDI với nhau; cốt lõi của VAN là một hệ thống thư điện tử cho phép các máy tính điện tử liên lạc được với nhau, và hoạt động như một phương tiện lưu trữ và tìm kiếm; khi nối vào VAN, một doanh nghiệp có thể liên lạc với nhiều máy tính điện tử nằm ở nhiều thành phố trên khắp thế giới.

Ngày nay EDI chủ yếu được thực hiện thông qua mạng Internet. Để phục vụ cho buôn bán giữa các doanh nghiệp thuận lợi hơn với chi phí truyền thông không quá tốn kém người ta đã xây dựng một kiểu mạng mới gọi là “mạng riêng ảo” (virtual private network), là mạng riêng dạng intranet của một doanh nghiệp nhưng được thiết lập dựa trên chuẩn trang Web và truyền thông qua mạng Internet.

Công việc trao đổi EDI trong TMĐT thường gồm các nội dung sau: o Giao dịch kết nối o Đặt hàng o Giao dịch gửi hàng o Thanh toán

Vấn đề này đang được tiếp tục nghiên cứu và xử lý, đặc biệt là buôn bán giữa các nước có quan điểm chính sách, và luật pháp thương mại khác nhau, đòi hỏi phải có một pháp lý chung trên nền tảng thống nhất quan điểm về tự do hóa thương mại và tự do hóa việc sử dụng mạng Internet, chỉ như vậy mới bảo đảm được tính khả thi, tính an toàn, và tính có hiệu quả của việc trao đổi dữ liệu điện tử (EDI).

### 1.4.4.Truyền nội dung số hóa

Nội dung của hàng hóa số, giá trị của nó không phải trong vật mang tin mà nằm trong bản thân nội dung của nó. Hàng hóa số có thể được giao qua mạng. Ví dụ hàng hóa số là: Tin tức, nhạc, phim, các chương trình phát thanh, truyền hình, các chương trình phần mềm, các ý kiến tư vấn, vé máy bay, vé xem phim, xem kịch, hợp đồng bảo hiểm v.v..

Trước đây, hàng hóa số được trao đổi dưới dạng hiện vật bằng cách đưa vào đĩa, vào băng, in thành sách báo, thành văn bản, đóng gói bao bì chuyển đến tay người sử dụng, hoặc đến điểm phân phối để người sử dụng mua và nhận trực tiếp. Ngày nay, hàng hóa số được số hoa và gửi theo mạng, gọi là “giao gửi số hóa”.

Các tờ báo, các tư liệu công ty, các ca-ta-lô sản phẩm lần lượt đưa lên Web, người ta gọi là “xuất bản điện tử” (electronic publishing hoặc Web publishing), khoảng 2700 tờ báo đã được đưa lên Web gọi là “sách điện tử”, các chương trình phát thanh, truyền hình, giáo dục, ca nhạc, kể chuyện v.v..cũng được số hóa, truyền qua Internet, người sử dụng tải xuống (download); và sử dụng thông qua màn hình và thiết bị âm thanh của máy tính điện tử.

### 1.4.5.Mua bán hàng hóa hữu hình

Đến nay danh sách các hàng hóa bán lẻ qua mạng đã mở rộng, từ hoa tới quần áo, ôtô và xuất hiện một loạt hoạt động gọi là “mua hàng điện tử” hay “mua hàng trên mạng”; ở một số nước, Internet đã bắt đầu trở thành công cụ để cạnh tranh bán lẻ hàng hóa hữu hình. Tận dụng tính năng đa phương tiện của môi trường Web, người bán xây dựng trên mạng các cửa hàng ảo, gọi là ảo bởi vì, cửa hàng có thật nhưng ta nhưng ta chỉ xem toàn bộ quang cảnh cửa hàng và các hàng hóa chứa trong đó trên từng trang màn hình một.

Để có thể mua – bán hàng, khách hàng tìm trang Web của cửa hàng, xem hàng hóa hiển thị trên màn hình, xác nhận mua và trả tiển bằng thanh toán điện tử. Lúc đầu (giai đoạn một), việc mua bán như vậy còn ở dạng sơ khai: người mua chọn hàng rồi đặt hàng thông qua mẫu đơn (form) cũng đặt ngay trên Web. Nhưng có trường hợp muốn lựa chọn giữa nhiều loại hàng ở các trang Web khác nhau (của cùng một cửa hàng) thì hàng hóa miêu tả nằm ở một trang, đơn đặt hàng lại nằm ở trang khác, gây ra nhiều phiền toái. Để khắc phục, giai đoạn hai, xuất hiện loại phần mềm mới, cùng với hàng hóa của cửa hàng trên màn hình đã có thêm phần “xe mua hàng” (shopping cart, shopping trolley), giỏ mua hàng (shopping basket, shopping bag) giống như giỏ mua hàng hay xe mua hàng thật mà người mua thường dùng khi vào cửa hàng siêu thị. Xe và giỏ mua hàng này đi theo người mua suốt quá trình chuyển từ trang Web này đến trang Web khác để chọn hàng, khi tìm được hàng vừa ý, người mua ấn phím “Hãy bỏ vào giỏ” ( Put in into shopping bag); các xe hay giỏ mua hàng này có nhiệm vụ tự động tính tiền (kể cả thuế, cước vận chuyển) để thanh toán với khách mua. Vì hàng hóa là hữu hình, nên tất yếu sau đó cửa hàng phải dùng tới các phương tiện gửi hàng theo kiểu truyền thống để đưa hàng đến tay người tiêu dùng.

## CÂU HỎI

1. Website là gì?
2. Vì sao ngày càng có nhiều ứng dụng Website ra đời phục vụ nhu cầu mua bán và trao đổi qua mạng?
3. Những vấn đề gì cần phải cân nhắc khi mua bán qua mạng?
4. Các hình thức kinh doanh trên mạng?

# Chương 2

**NGÔN NGỮ ĐỊNH DẠNG SIÊU VĂN BẢN HTML**

## MỤC TIÊU

1. Về nhận thức
   * Cung cấp những kiến thức cơ bản về HTML và CSS
   * Các thẻ HTML cơ bản và nâng cao
2. Về kỹ năng

Vận dụng những kiến thức đã học vào thực tế để làm việc với ứng dụng Web.

* + Sử dụng các thẻ HTML từ cơ bản tới nâng cao
  + Định dạng trang HTML

## HÌNH THỨC VÀ PHƯƠNG PHÁP GIẢNG DẠY

* Trình chiếu Powerpoint
* Phương pháp dạy học theo dự án
* Thực nghiệm kết hợp với máy tính
* Bài tập về nhà

## TÀI LIỆU THAM KHẢO

1. Nhóm tác giả Elicom (2000), Ngôn ngữ HTML 4 cho World Wide Web, NXB Hà Nội (Sách có tại thư viện ĐHDT)
2. <http://vi.wikipedia.org/wiki/HTML>
3. [http://www.w3schools.com/ **html** /default.asp](http://www.w3schools.com/html/default.asp)

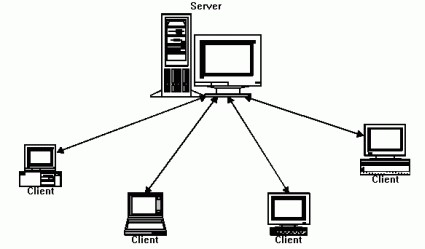
## 2.1.MÔ HÌNH WEB CLIENT/SERVER

Thuật ngữ Server được dùng cho những chương trình thi hành như một dịch vụ trên toàn mạng. Các chương trình Server này chấp nhận tất cả các yêu cầu hợp lệ đến từ mọi nơi trên mạng, sau đó nó thi hành dịch vụ và trả kết quả về máy yêu cầu. Một chương trình được coi là Client khi nó gửi các yêu cầu tới máy có chương trình Server và chờ đợi câu trả lời từ Server. Chương trình Server và Client nói chuyện với nhau bằng các thông điệp (messages) thông qua một cổng truyền thông liên tác IPC (Interprocess Communication).

Để một chương trình Server và một chương trình Client có thể giao tiếp được với nhau thì giữa chúng phải có một chuẩn để nói chuyện, chuẩn này được gọi là giao thức. Nếu một chương trình Client nào đó muốn yêu cầu lấy thông tin từ Server thì nó phải tuân theo giao thức mà Server đó đưa ra. Bản thân chúng ta khi cần xây dựng một mô hình Client/Server cụ thể thì ta cũng có thể tự tạo ra một giao thức riêng nhưng thường chúng ta chỉ làm được điều này ở tầng ứng dụng của mạng. Với sự phát triển mạng như hiện này thì có rất nhiều giao thức chuẩn trên mạng ra đời nhằm đáp ứng nhu cầu phát triển này. Các giao thức chuẩn (ở tầng mạng và vận chuyển) được sử dụng rộng rãi nhất hiện nay như: giao thức TCP/IP, UDP/IP, giao thức SNA của IBM, OSI, ISDN, X.25 hoặc giao thức LAN-toLAN NetBIOS.

**Một máy tính chứa chương trình Server được coi là một máy chủ hay máy phục vụ (Server) và máy chứa chương trình Client được coi là máy khách (Client). Mô hình mạng trên đó có các máy chủ và máy khách giao tiếp với nhau theo 1 hoặc nhiều giao thức được gọi là mô hình Client/Server.** Thực tế thì mô hình Client/Server là sự mở rộng tự nhiên và tiện lợi cho việc truyền thông liên tiến trình trên các máy tính cá nhân. Mô hình này cho phép xây dựng các chương trình Client/Server một cách dễ dàng và sử dụng chúng để liên tác với nhau để đạt hiệu quả hơn.

Mô hình Client/Server như sau: Client/Server là mô hình tổng quát nhất, trên thực tế thì một Server có thể được nối tới nhiều Server khác nhằm làm việc hiệu quả và nhanh hơn. Khi nhận được 1 yêu cầu từ Client, Server này có thể gửi tiếp yêu cầu vừa nhận được cho Server khác ví dụ như Database Server vì bản thân nó không thể xử lý yêu cầu này được. Máy Server có thể thi hành các nhiệm vụ đơn giản hoặc phức tạp. Ví dụ như một máy chủ trả lời thời gian hiện tại trong ngày, khi một máy Client yêu cầu lấy thông tin về thời gian nó sẽ phải gửi một yêu cầu theo một tiêu chuẩn do Server định ra, nếu yêu cầu được chấp nhận thì máy Server sẽ trả về thông tin mà Client yêu cầu.



*Hình 2.1 Mô hình máy khách, máy chủ*

## 2.2.NGÔN NGỮ ĐỊNH DẠNG SIÊU VĂN BẢN HTML

### 2.2.1.Cấu trúc cơ bản của trang HTML

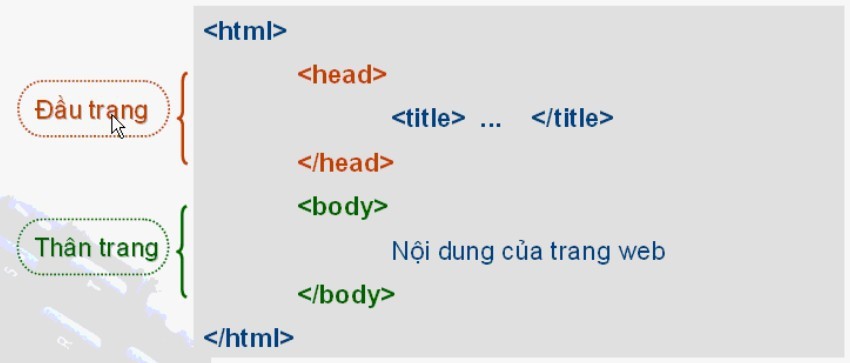
HTML (Hyper Text Markup Language) là một ngôn ngữ định dạng siêu văn bản được thiết kế để tạo nên các trang web, nghĩa là các mẫu thông tin được trình bày trên World Wide Web.

Ngôn ngữ HTML tạo trang web tĩnh dưới dạng hồ sơ với tên đuôi là .htm hay .html. Ngôn ngữ HTML là tập hợp của nhiều thẻ lệnh dưới dạng

<Thẻ\_Lệnh>Thông\_Tin</Thẻ\_Lệnh>

Trang mạng có thể soạn thảo bằng các Trình chuyên về thảo trang mạng như FrontPage của Microsoft hoặc Dreamweaver của Adobe.

Mọi trang mạng HTML đều có cấu trúc sau:



Ví dụ:

Trang trống:

<html>

<head>

<title>Trang Mang Trong</title>

</head>

<body>

</body>

</html>

Trang chào mừng:

<html>

<head>

<title>Trang Chao Mung</title>

</head>

<body>

Xin Chào mừng bạn đến với trang mạng của chúng tôi

</body>

</html>

### 2.2.2.Cú pháp chi tiết của tag HTML

Cú pháp chung của thẻ lệnh HTML



Thẻ lệnh không có **nội dung** gọi là thẻ rỗng (empty tag) và có cấu trúc như sau:



Một số điểm cần lưu lý khi soạn thảo siêu văn bản bằng HTML:

* Dấu cách chỉ có tác dụng duy nhất một lần cách, nếu muốn thể hiện nhiều dấu giãn cách liền nhau phải sử dụng thẻ
* Xuống dòng văn bản phải sử dụng thẻ tương ứng
* Tên thẻ không phân biệt chữ in hoa và chữ in thường
* Dòng chú thích trong mã nguồn được đặt giữa cặp dấu chú thích <!-- và -->. Trình duyệt sẽ không quan tâm đến phần nằm giữa cặp dấu đó
* Quy tắc viết các ký tự đặc biệt trong HTML: tên qui định nằm giữa dấu “&” và dấu “;” &tên\_qui\_định;

Năm ký tự đặc biệt hay dùng

**Ky ́tự**

**Caćh viết**

<

&lt;

>

&gt;

“

&quot;

Ky ́tự trắng

&nbsp;

&

&amp;

## 2.3.ĐỊNH DẠNG CƠ BẢN

### 2.3.1.Các thao tác khối

Khối trích dẫn thường được dùng trong các trường hợp sử dụng các trích dẫn mở rộng. Toàn khối trích dẫn sẽ được căn lề thụt vào cả ở hai phía và tạo thành một khối riêng so với các đoạn khác của trang văn bản HTML.

Khối trích dẫn được mở đầu bằng thẻ **<BLOCKQUOTE>** và kết thúc bằng thẻ

**</BLOCKQUOTE>**.

Ví dụ:

<blockquote>

Đây là ví dụ về khối trích dẫn. Toàn khối trích dẫn sẽ được

|  |
| --- |
| căn lề thụt vào cả ở hai phía và tạo thành một khối riêng so với các đoạn khác của trang văn bản HTML.  </blockquote> |

Kết quả như sau:

Đây là ví dụ về khối trích dẫn. Toàn khối trích dẫn sẽ được căn lề thụt vào cả ở hai phía và tạo thành một khối riêng so với các đoạn khác của trang văn bản HTML.

### 2.3.2.Định dạng ký tự

Các thẻ kiểu chữ trong ngôn ngữ HTML cho phép bạn điều khiển được vẻ bề ngoài của bản thân các ký tự – chẳng hạn chúng ta có thể định dạng một từ là **đậm** hoặc *nghiêng*. Các định dạng này có thể áp dụng từ phạm vi một đoạn hay toàn bộ văn bản cho tới từng ký tự đơn lẻ.

### Mã nguồn Kết quả

<b>đậm</b> **đậm**

<u>gạch dưới</u> gạch dưới

<i>in nghiêng</i> *nghiêng*

<big>chữ lớn hơn bình thường</big> chữ lớn hơn bình thường

<small>chữ nhỏ hơn bình thường</small> chữ nhỏ hơn bình thường

<sup>chỉ số trên</sup> chỉ số trên

<sub>chỉ số dưới></sub> chỉ số dưới

<basefont>kích thước font chữ</basefont> Dùng để qui định font chữ của cả trang web

<font>kiểu chữ hiển thị</font> Qui định tên font, màu sắc, kích cỡ font <center>chữ được canh giữa</center> chữ được canh giữa

**2.3.3.Định dạng đoạn** **Các thẻ xử lí đoạn**

### a. Thẻ phân chia đoạn

Trong HTML, các đoạn tài liệu, văn bản, hình ảnh,…được phân chia bằng cặp thẻ <P>văn bản</P>

Thẻ <P> có một số thuộc tính sau:

− Align: Thuộc tính này sẽ chứa 1 trong 3 giá trị:

* Center: đoạn tài liệu sẽ được canh chỉnh vào giữa o Left: đoạn tài liệu được canh chỉnh theo lề trái o Right: đoạn tài liệu được canh chỉnh theo lề phải o Justify: đoạn tài liệu được canh chỉnh theo hai bên Ví dụ:

<HTML>

<BODY>

<p align ="justify"> Canh chỉnh theo hai bên</p>

</BODY>

</HTML>

− Style: Thuộc tính này sẽ qui định khoảng cách lề của đoạn. Trong thuộc tính này lại có các thuộc tính con, tuy nhiên các thuộc tính con này được đặt cách nhau bằng dấu chấm phẩy:

* Margin-left: x (x là số nguyên chỉ định chiều rộng của lề trái) o Margin-right: y (y là số nguyên chỉ định chiều rộng của lề phải) o Margin-top: z (z là số nguyên chỉ định chiều rộng của lề trên) o Margin-bottom: t (t là số nguyên chỉ định chiều rộng của lề dưới) o Line-height: u% (u là khoảng cách giữa các dòng tính theo đơn vị 100%) Ví dụ:

<HTML>

<BODY>

<p align ="center" style ="margin-left: 10; margin-right: 5; margin-top: 6; margin-bottom: 6"> Đoạn này xác định đoạn văn bản căn giữa, có lề trái = 10, lề phải bằng 5, lề trên = 6, lề dưới = 6</p> <p align ="left" > Xác định đoạn văn bản căn trái, các lề đặt theo mặc định của trình duyệt</p>

<p align ="left" style="line-height: 150%" > Đoạn này xác định đoạn văn bản căn trái, các lề đặt theo mặc định của trình duyệt, khoảng cách giữa các dòng 150%. </p>

</BODY>

</HTML>

### b. Thẻ xuống dòng

Trong HTML, các ký tự xuống dòng không được sử dụng. Để ngắt một dòng nào đó, ta dùng thẻ <BR>. Đây là một thẻ đơn.

<HTML>

<HEAD>

<TITLE></TITLE>

</HEAD>

<BODY>

<p align="left" style="margin-left : 30">

Mỗi năm hoa đào nở<BR>Lại thấy ông đồ già <BR>

Bày mực tàu giấy đỏ<BR>Bên phố đông người qua<BR></p>

</BODY>

</HTML>

#### 2.3.4.Chèn đường kẻ ngang

Tạo một đường kẻ ngang trong trang HTML bằng thẻ <hr>. <hr> là một thẻ tự đóng, điều đó có nghĩa là nó không cần thẻ đóng như những thẻ HTML khác. Thẻ <hr> sẽ tạo ra một đường kẻ ngang trên trình duyệt và khoảng cách giữa đường kẻ và nội dung thì phụ thuộc vào các trình duyệt khác nhau.

Tuy nhiên bạn hoàn toàn có thể thay đổi cách hiển thị của thẻ <hr> theo ý mình trên một số trình duyệt. Đối với một số trình duyệt mới (IE6, IE7, Firefox, Opera, Mozilla...) thì cách định dạng của bạn hoàn toàn có thể tương thích.

Trong cách định dạng thẻ <hr> chúng ta cũng cần phải chú ý một chút về cách mà cách trình duyệt xử lí đối với đối tượng thẻ <hr>. Với Internet Explore (IE) sẽ sử dụng thuộc tính color để hiển thị màu của đường kẻ và thuộc tính **background** sẽ không có tác dụng. Tuy nhiên Mozilla(Netscape) và Opera thì lại dùng thuộc tính **background** để hiển thị màu của đường kẻ. Chúng ta cũng có thể sử dụng thuộc tính **border** để hiển thị định dạng, nhưng đó không phải là một cách mà chúng ta muốn.

<hr width = 80% color= red background= red size= 1 height=

1px>

Để thêm khoảng cách giữa đường kẻ và nội dung, chúng ta sẽ dùng thuộc tính margin để qui định khoảng cách này.

<hr width = 80% color= red background= red size= 2 height= 1px margin= 10px 0 20px>

Với thuộc tính **margin** như trên chúng ta sẽ có khoảng cách giữa đường kẻ với phía trên là 10px, phía dưới là 20px và khoảng cách hai bên bằng 0.

#### 2.3.5.Chèn hình ảnh

Nếu bạn muốn Web site của mình nổi trội hơn, bạn phải theo xu hướng đồ hoạ với các hình ảnh được chọn kỹ lưỡng. Làm thế nào để chèn hình vào trong khi các tập tin HTML chỉ có văn bản? Nhờ bổ sung thẻ <IMG> vào tài liệu, nó sẽ ra lệnh Chèn một hình vào đây. Thẻ này xác định tên của tập tin đồ hoạ để trình duyệt có thể mở tập tin đó và hiển thị các hình:

<IMG SRC=filename>

SRC là viết tắt của source (nguồn) và tên tập tin là tên và đường dẫn của tập tin đồ hoạ mà bạn muốn hiển thị (dùng dạng GIF hoặc JPG).

Giá trị của thuộc tính SRC được gán bằng URL là một địa chỉ trên máy tính địa phương hay trên Internet. Nếu tệp ảnh nằm ngay trên máy tính địa phương thì cần ghi rõ tên tệp ảnh. Nếu tệp ảnh lấy từ một nơi khác trên Internet thì phải ghi đầy đủ địa chỉ URL.

Giả sử bạn có một hình tên là logo.gif. Để đưa nó vào trang Web, bạn dùng dòng sau đây.

<IMG SRC= “logo.gif”>

Cấu trúc đầy đủ của thẻ <IMG>:

<IMG ALIGN=align-type ALT=text SRC=url BORDER=n HEIGTH=n

WIDTH=n HSPACE=n VSPACE=n>

Trong đó:

− ALIGN=align-type: lề cho ảnh hay cho văn bản bao quanh ảnh. Giá trị có thể là TOP, MIDDLE, BOTTOM, LEFT, RIGHT.

− ALT=text: văn bản hiển thị hay thay thế cho hình ảnh khi chức năng "Show Picture" trong trình duyệt bị tắt.

− SRC=url: chỉ định ảnh

− BORDER=n: đường viền ảnh. Nếu ảnh được dùng cho hyperlink thì đường viền có màu trùng với màu hyperlink. Nếu ảnh không dùng cho hyperlink thì đường viền không hiển thị.

− HEIGTH=n, WIDTH=n: chỉ độ cao và độ rộng của ảnh. Ðơn vị bằng pixel.

− HSPACE=n, VSPACE=n: chỉ định khoảng cách từ ảnh đến văn bản quanh nó theo chiều ngang và dọc.

#### 2.3.6.Tìm và thay đổi dữ liệu

Đây là chức năng hỗ trợ người dùng tìm kiếm hoặc thay đổi dữ liệu nhanh chóng bằng cách sử dụng các công cụ sẵn có trong bộ soạn thảo FrontPage. Để tìm kiếm trong một tập tin ta phải tiến hành ở tập tin để chỉnh sửa trong FrontPage.

* Vào Replace trong menu Edit, hoặc nhấn Ctrl-H.
* Nhập các từ để tìm vào ô tìm kiếm và các từ để thay thế trong hộp thay thế.
* Kiểm tra lại để chắc chắc chắn rằng trang cần thay đổi đang được chọn, và nhấn vào Replace. Hãy nhấp tiếp vào Replace cho đến FrontPage đã thay thế tất cả các trường hợp trên trang.

**Để tìm kiếm trên toàn bộ một trang web:** Mở FrontPage và mở một trang web được xác định trước .

* Trong danh sách thư mục, đánh dấu các file bạn muốn tìm kiếm.
* Vào Replace trong menu Edit, hoặc nhấn Ctrl-H.
* Nhập các từ để tìm vào ô tìm kiếm và các từ để thay thế trong hộp thay thế.
* Kiểm tra lại để chắc chắn rằng các trang được chọn nếu bạn chỉ muốn tìm kiếm một số trang trong web của bạn, hoặc "Tất cả các trang" nếu bạn muốn tìm kiếm tất cả các trang. Sau đó nhấp vào "Thay thế Tất cả".
* FrontPage sẽ cảnh báo bạn rằng bạn sẽ không thể hủy bước này hoạt động. Nhấp vào "Yes".
* FrontPage sẽ hiển thị cho bạn tất cả những nơi mà chuỗi tìm kiếm của bạn đã được tìm thấy.
* Nhấp vào "Cancel" khi bạn đang thực hiện.

**Lời khuyên:**

1. Để tránh kết hợp vào các thay thế dữ liệu không đúng như mong đợi ta cần tìm kiếm những từ đầy đủ nghĩa.
2. Bạn có thể nhấp đúp chuột vào các trang kết quả để xem những gì đã được thay đổi, và thực hiện chỉnh sửa.

## 2.4.LIÊN KẾT (HYPERLINK)

### 2.4.1.Liên kết đến một trang web khác

Đây là một ví dụ thực tế mà chúng ta thường hay gặp, bất cứ lúc nào lướt web, chúng ta nhấp vào đâu đó, và thế là được chuyển hướng sang trang khác. Để làm được việc này, ta bọc đối tượng ( đối tượng có thể là chữ, hình ảnh) với tag <a> và dĩ nhiên câu hỏi đặt ra là làm sao định hướng được đi tới đâu, câu trả lời:

Trong tag <a> ta cần có thêm thuộc tính **href** (hypertext reference). Giá trị của thuộc tính này là đường dẫn tới trang mà ta muốn nó đến.

Ví dụ: <a href = http://www.google.com.vn>Tìm kiếm</a>

HREF (Hypertext Reference – tham chiếu siêu văn bản) được dùng để liên kết đến:

Một tài liệu khác (external link) hay một phần khác nằm trong chính tài liệu đang đọc

(internal link)

Giao thức để tham chiếu HREF là HTTP. Nếu là liên kết nội tại – internal link thì không cần phải có phần giao thức.

### 2.4.2.Cách tạo một email-link

Chúng ta cũng thấy trên nhiều trang web có những link mà chỉ càn gõ vào nó là chương trình e-mail tự động mở ra cho chúng ta viết thư tới một địa chỉ đã được định sẵn, nhiều khi tiêu đề (subject) của e-mail cũng đã có sẵn. Click [vào đây] hoặc vào biểu tượng bên để gửi e-mail.

Ví dụ: Giáo viên: <a href = mailto:vuthimaiphuong@duytan.edu.vn>Vũ Thị Mai Phương </a> Email: vuthimaiphuong@duytan.edu.vn

Thay vì địa chỉ một trang web, bạn chỉ cần đưa mailto: và địa chỉ e-mail, muốn có subject và text sẵn, chỉ cần thêm ?subject=tiêu đề&body=text

### 2.4.3.Liên kết ImageMap

Với Image Map, “đầu mối liên kết” là một vùng trên một hình ảnh có hình dạng tùy ý nào đó.

Các công cụ soạn thảo trực quan cho phép dễ dàng tạo Image Map.

Một ví dụ về mã nguồn HTML của thành phần Image Map như sau:

<MAP NAME=”Map01”>

<AREA HREF=”chapter1.html” SHAPE=”rect” coords=”6, 22, 167, 169”>

<AREA HREF=”chapter2.html” SHAPE=”circle” coords=”259, 106, 36”>

<AREA HREF=”chapter3.html” SHAPE=”polygon” coords=”123, 196, 56, 211,

278, 201”>

</MAP>

<IMG BORDER=”0” SRC=”image.gif” WIDTH=”400” HEIGHT=”600”

USEMAP=”#Map01”>

Đoạn mã trên sẽ tạo 3 đầu mối liên kết là ba vùng hình chữ nhật (rect), hình tròn

(circle) và hình tam giác (polygon) trên nền ảnh image.gif. Khi nhấn chuột vào phạm vi 3 vùng này thì các tệp tương ứng chapter1.html, chapter2.html và chapter3.html sẽ được nạp về để hiển thị.

## 2.5.TẠO FRAMESET, TABLE

### 2.5.1.Frameset

#### *2.5.1.1. Frameset - Cols*

Dùng Frames, bạn có thể chia một window ra làm nhiều phần riêng biệt, không phụ thuộc vào nhau và có thể chứa những nội dung khác nhau. Từ Frame này có thể thay đổi nội dung của Frame kia. Như vậy website của bạn sẽ được bố cục rõ ràng hơn, tiện lợi hơn và người xem có thể tìm được cái mình tìm nhanh chóng hơn. Ai hay sử dụng Windows Explorer sẽ thấy được tác dụng tương tự của Frames.

Khi chia window, bạn sẽ có ít nhất 2 Frames. Những Frames này được gọi là một Frame-Set. Một trang sử dụng Frames có cấu trúc cơ bản như sau:

<html>

<frameset>

</frameset>

</html>

Ðể chia một window ra làm hai phần theo chiều dọc, phần bên trái chiếm 30%, phần bên phải 70% chiều rộng của window ta làm như sau:

<html>

<frameset cols="30%,70%">

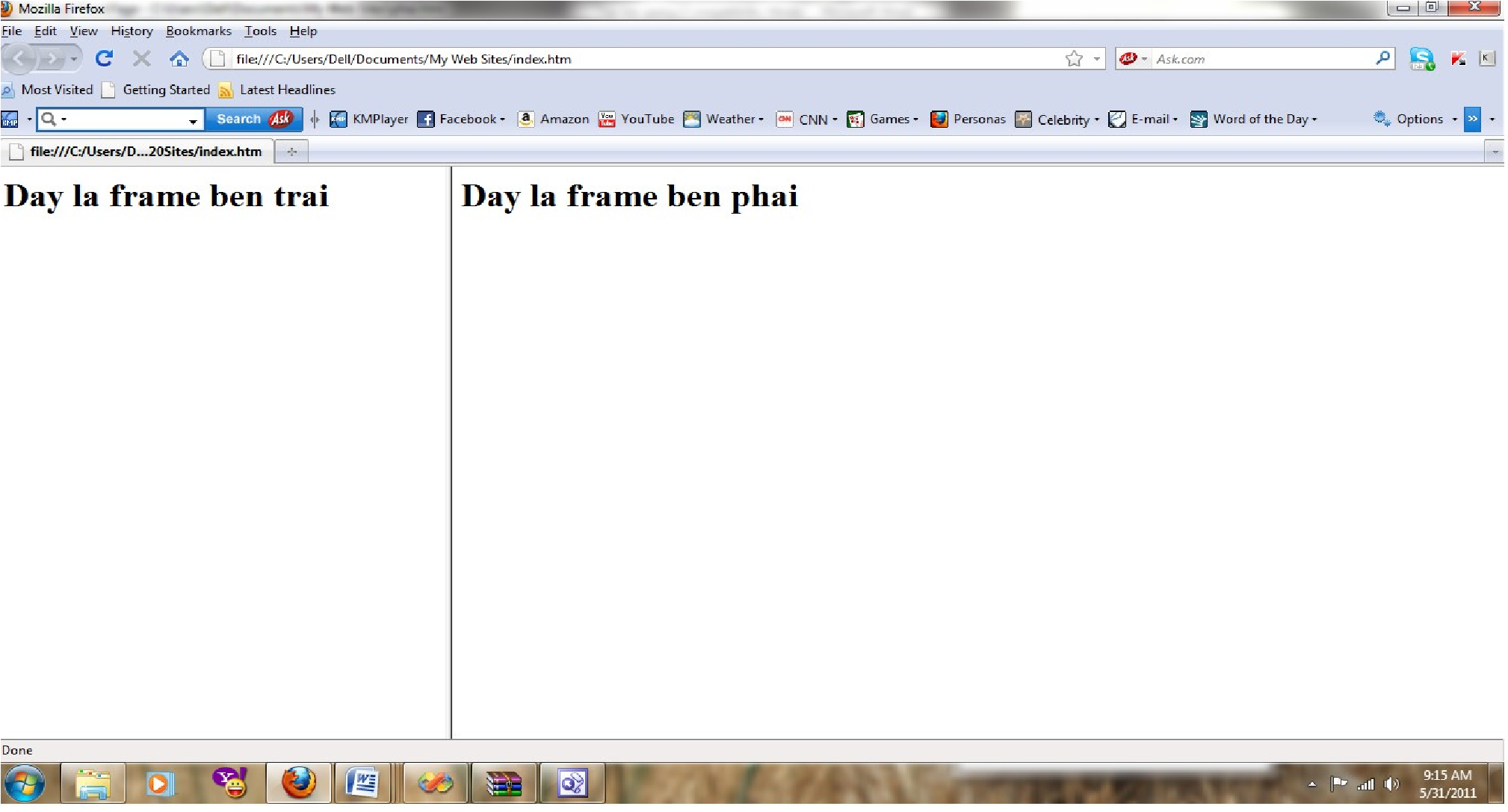
<frame name ="ben trai" src="index.htm">

<frame name ="ben phai" src="noidung.htm">

</frameset>

</html>

Kết quả như hình II.2:



Hình 2.2. Frameset – Columns

#### *2.5.1.2. Frameset – Rows*

Tương tự như vậy ta chia theo chiều ngang, nhưng lần này chia làm 3 phần (20%, 20% và \* là phần còn lại của màn hình):

<html>

<frameset rows="20%,20%,\*">

<frame name ="tren" src="tren.htm">

<frame name ="duoi" src="duoi.htm">

<frame name ="con\_lai" src="conlai.htm">

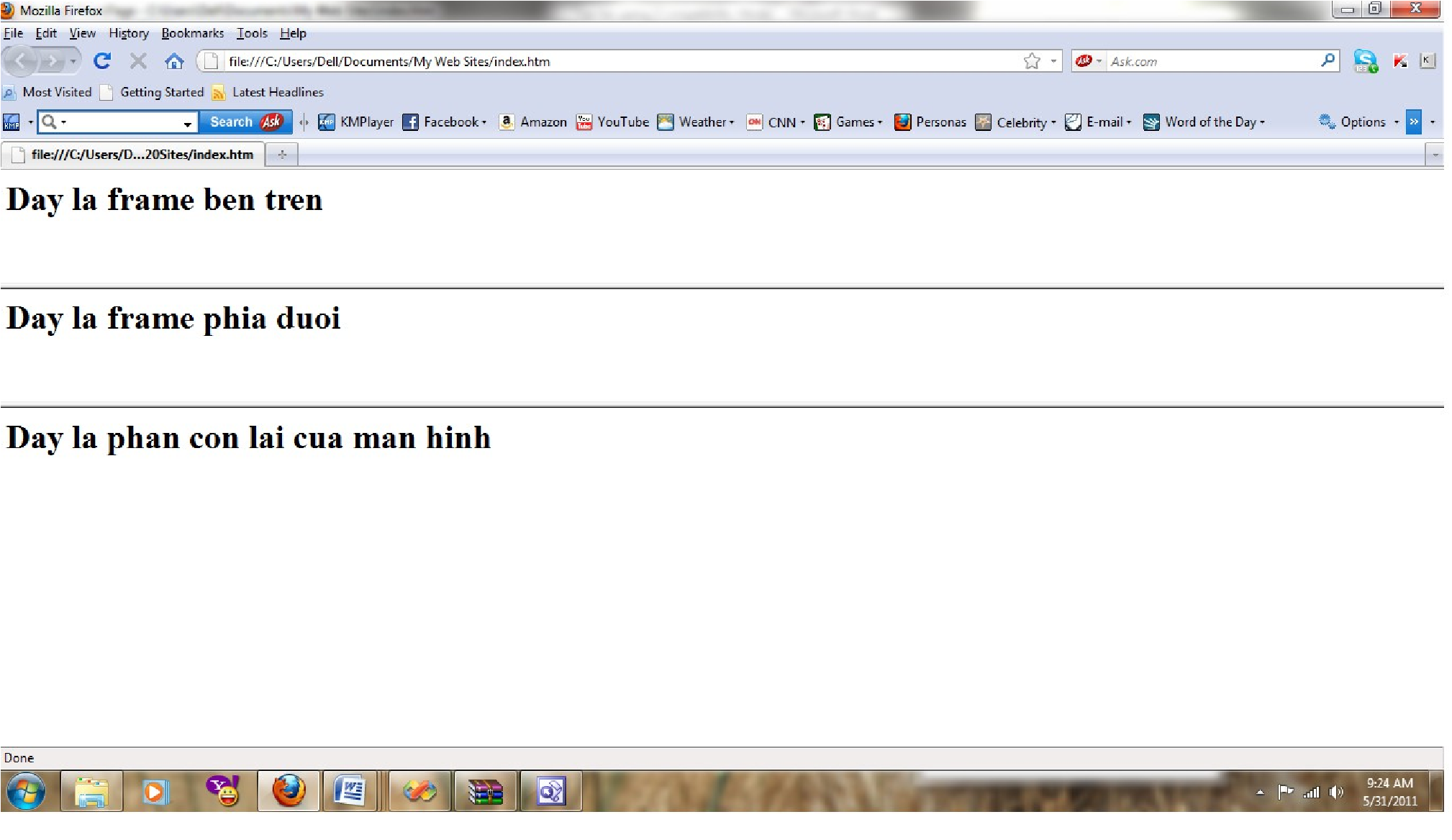
</frameset>

</html>

Trong đó:

* **Cols**: (columns) chia window theo chiều dọc.
* **Rows**: cho phép chia theo chiều ngang. 20%, 20% là chiều cao của hai Frames đầu. Dấu \* cho biết chiều cao của Frame thứ 3 là phần còn lại của window.
* **<frame name ="ben trai" src="trai.htm">** tên và nội dung của frame bên trái. Bạn cứ tưởng tượng, bạn tạo được một khung ảnh có hai ngăn (tương tự như một frameset), mỗi ngăn chứa một bức ảnh (đây là một trang web). Frame bên trái tên là "trai" và chứa trang "trai.htm". Tương tự như vậy với "ben phai", "phai.htm", "tren",

"tren.htm", "duoi", "duoi.htm", "con\_lai", "conlai.htm" Kết quả như hình II.3:



Hình 2.3. Frameset – Rows

#### *2.5.1.3.Frameset – Navigation Menu*

Như vậy ta đã có được những bức ảnh trong khung. Nhưng mục đích chính của Frames không phải là làm thế nào để có nhiều trang trong một window mà là làm thế nào để dùng link của Frame này thay đổi nội dung của Frame kia. Bây giờ ta hãy thử dùng trang "trai.htm" làm navigation để thay đổi nội dung Frame "phai". Khi đó ta phải dùng thêm một dòng code nữa trong trang "trai.htm": <target="ben phai">

#### *2.5.1.4.Framset in frameset*

Và một ví dụ nữa sẽ cho thấy khả năng của Frames. Trong ví dụ này, đầu tiên ta chia window ra làm 2 phần sau đó lại chia phần thứ 2 ra làm đôi. Code của nó như sau:

<html>

<frameset rows="20%,70%">

<frame src="tren.htm" name="ben tren">

<noframe>

<frameset cols="20%,80%">

<frame src="trai.htm" name="ben trai">

<frame src="phai.htm" name="ben phai">

</frameset>

<body>

This page uses frames but your browser doesn't support them.

Please download a higher version and visit me again. Thanks anyway!

</body>

</html>

Bạn thấy đấy, ta đã chia window ra làm 4 Frames nhưng chỉ thấy có 3. Ðó là tác dụng của tag <noframe> Hiện nay số lượng browser không biết Frames hầu như không còn nữa, nhưng để chắc chắn, ta hãy viết thêm một phần <body> và chú thích rằng browser của người đọc không biết frames, nếu không người đó sẽ chẳng đọc được một chữ gì khi đến thăm homepage của bạn.

Ngoài ra còn một số đặc tính khác của Frames như:

* **scrolling="no"** (nếu bạn viết như vậy trong tag <frameset>, người đọc sẽ không thể scroll để đọc tiếp nếu như trang html có cỡ to hơn Frame chứa nó).
* **frameborder="0"** (frame không có khung)
* **framespacing="0"** (tương tự như trong table)
* **noresize** (không cho phép người đọc thay đổi kích cỡ của window) Kết quả như hình II.4:



Hình 2.4. Frameset in frameset

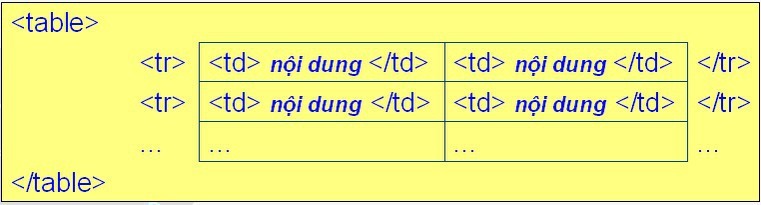
### 2.5.2.Table

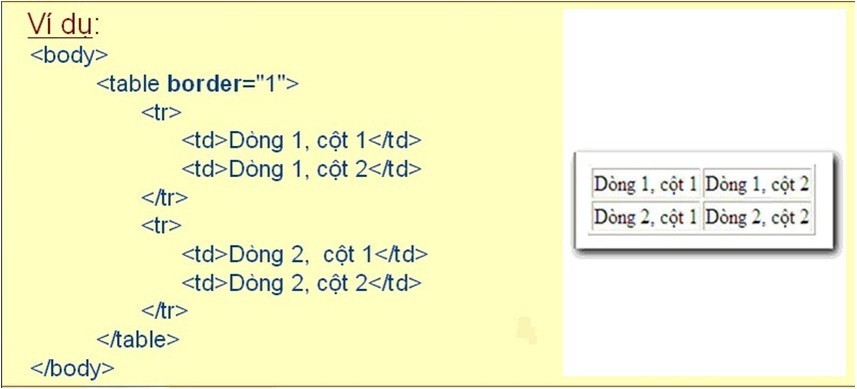
Table - một yếu tố rất quan trong trong các Website đẹp. Table cho phép chúng ta có một sự chính xác đến từng pixel trong bố cục của Homepage. Với Table ta có thể trang trí và bố cục trang web của mình như một tờ báo với nhiều cột khác nhau. Table làm cho trang web của chúng ta có một layout sinh động hơn.

### *a) Tạo bảng biểu*

Cặp TAG <table></table> tạo ra một Table. Như chúng ta biết, một Table được tạo ta từ nhiều dòng và mỗi dòng lại có thể chứa nhiều ô. Mỗi cặp TAG <tr></tr> tạo ra một dòng, trong dòng ấy ta có thể sử dụng nhiều cặp TAG <td></td> để có nhiều ô.

Khung cấu trúc:

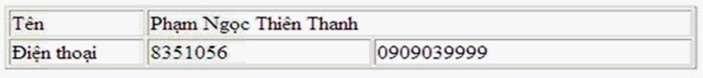




Thuộc tính **border = <giá trị>** dùng định nghĩa một table có đường viền và khai báo độ đậm của đường viền. Tuy nhiên không làm ảnh hưởng tới độ rộng của table.

### *b) Trộn dòng, trộn cột*

* Trộn cột



Trong table trên, ta thấy các ô có chiều rộng khác nhau, chính xác hơn là ô thứ hai của dòng thứ 1, rộng bằng 2 ô bên dưới. Ðiều đó được thực hiện bởi thuộc tính ***colspan***

(xem HTML Code)

<

body

>

<

table border=”1” width=”600”

>

<

tr

>

<

td width=”200”>Tên</td

>

<

td

***colspan***

=

”2” width=”400”>Phạm Ngọc Thiên Thanh</td

>

<

/tr

>

<

tr

>

<

td>Điện thoại</td

>

<

td>8351056</td

>

<

td>0909039999</td

>

<

/tr

>

<

/table

>

<

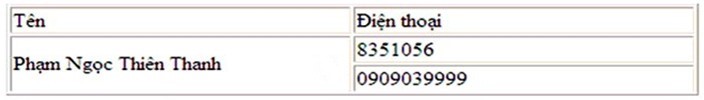
/body

>

Thuộc tính **width = <giá trị>** dùng để đặt độ rộng của table. Giá trị này có thể được gán bằng con số hoặc có thể khai báo bằng %. Nếu khai báo theo % thì độ rộng của table được tính theo % kích thước của cửa sổ trình duyệt.

Lưu ý: mặc định, nếu như dòng thứ nhất ta đã khai báo độ rộng cho các cột và dòng tiếp theo chúng ta không thay đổi độ rộng thì dòng tiếp theo cũng có độ rộng bằng với dòng đầu tiên.

* Trộn dòng



Trong table này, chúng ta lại thấy các dòng có độ rộng khác nhau. Cụ thể, dòng thứ hai, ô thứ nhất có độ rộng bằng hai dòng. Đó là tác dụng của thuộc tính ***rowspan***. (xem

HTML Code)

body

>

<

>

<

table border=”1”

<

tr

>

<

>

td>Tên</td

>

td>Điện thoại</td

<

<

/tr

>

<

tr

>

<

td

***rowspan***

”2”>Phạm Ngọc Thiên Thanh</td

=

>

<

td>8351056</td

>

>

/tr

<

<

>

tr

td>0909039999</td

<

>

<

/tr

>

<

>

/table

>

<

/body

***c)***

***Định dạng table***

Table - một yếu tố rất quan trong

trong các Website đẹp. Table cho

phép bạn có một sự chính xác đến

từng pixel trong bố cục của

Homepage. Với Table bạn có thể

trang trí và bố cục trang web của

mình như một tờ báo với nhiều cột

Table - một yếu tố rất quan trong

trong các Website đẹp. Table cho

phép bạn có một sự chính xác đến

từng pixel trong bố cục của

Homepage. Với Table bạn có thể

trang trí và bố cục trang web của

mình như một tờ báo với nhiều cột

Table - một yếu tố rất quan trong

trong các Website đẹp. Table cho

phép bạn có một sự chính xác đến

từng pixel trong bố cục của

Homepage. Với Table bạn có thể

trang trí và bố cục trang web của

khác nhau. khác nhau. mình như một tờ báo với nhiều cột khác nhau.

Với Table bạn có thể trang trí và bố Với Table bạn có thể trang trí và bố Với Table bạn có thể trang trí và bố cục trang web của mình như một tờ cục trang web của mình như một tờ cục trang web của mình như một tờ báo với nhiều cột khác nhau. Table báo với nhiều cột khác nhau. Table báo với nhiều cột khác nhau. Table làm cho trang web của bạn có một làm cho trang web của bạn có một làm cho trang web của bạn có một layout sinh động hơn. layout sinh động hơn. layout sinh động hơn.

HTML Code của Table trên được viết như sau:

|  |
| --- |
| <table border="1" bordercolor="red" width="80%" align="center" cellpadding="10" cellspacing="5" bgcolor="gray"> <tr>  <td width="33%" bgcolor="#C0C0C0" align="left">Dòng 1 - ô 1</td>  <td width="33%" bgcolor="#C0C0C0" align="left">Dòng 1 - ô 2</td> <td width="34%" bgcolor="#C0C0C0" align="left">Dòng 1 - ô 3</td>  </tr>  <tr>  <td width="33%" bgcolor="#C0C0C0" align="left">Dòng 2 - ô 1</td>  <td width="33%" bgcolor="#C0C0C0" align="left">Dòng 2 - ô 2</td> <td width="34%" bgcolor="#feedd1" align="left">Dòng 2 - ô 3</td>  </tr>  </table> |

* **<table width="100%">** cho biết chiều rộng của table này là 100%, con số này không có một giá trị cố định, nó sẽ thay đổi theo mức rộng của màn ảnh của người đến xem, như vậy người xem không phải scroll ngang. Nhưng khi bạn dùng một giá trị cố định, chẳng hạn: width="800", Table này sẽ luôn luôn rộng chừng ấy pixel, bất kể màn ảnh là 800 hay 1024 pixels.
* **<td width="33%">** cho biết, ô đó chiếm 33% chiều rộng của dòng.
* **<table align="center">** hướng bảng vào trung tâm của trang web, <td align= "left"> định hướng toàn bộ nội dung của một ô. align có thể mang các giá trị sau: "left" (gía trị mặc định - không cần viết cũng được), "center" (trung tâm), "right" (phải)
* **bgcolor** có thể sử dụng cho trang web (<body>), toàn bộ table (<table>) hoặc từng ô (<td>)
* **border="1" bordercolor="red"** có nghĩa: bảng có khung với độ dày 1, màu đỏ
* Bạn có thể định khoảng cách giữa nội dung và khung trong một ô bằng **cellpadding, cellpadding="10"** có nghĩa là text cách khung 10 pixels. Tương tự như vậy với khoảng cách giữa các ô trong bảng **(cellpadding): cellspacing="5"** có nghĩa là các ô của table cách nhau 5 pixels

## 2.6.ĐỊNH DẠNG CẢI TIẾN VỚI CSS

### 2.6.1.Style Sheets

Tương tự trong Word, bạn có thể dùng Style Sheets (CSS) trong HTML để định dạng cho trang web. Thay vì phải viết đi viết lại một số TAGs, bạn có thể viết sẵn trong một đoạn code ở đầu trang hoặc trong một .css file rồi import vào trang web.

Ví dụ:

|  |
| --- |
| <html>  <style type="text/css"> .button{ background-color: #7083c7; color: #ffffff; border-left: 1px solid #9dbcff; border-right: 1px solid navy; border-top: 1px solid #9dbcff; border-bottom: 1px solid navy } .text{ font-family: arial, verdana; font-color: red; font-size: 12px; } body { SCROLLBAR-ARROW-COLOR: #ffffff;  SCROLLBAR-BASE-COLOR: #7083c7  }  </style>  <body>  <form>  <input type="button" value="send" class="button">  </form>  <div class="text">This is a test</div> |

</body>

</html>

Trong ví dụ trên, ta thấy dòng chữ "This is a test" không nằm trong cặp TAG <font> và </font> mà nằm giữa <div> và <div>. TAG <div> mang thuộc tính class="text" và có tác dụng như: <font face="arial, verdana" color="red" size="2">. Cái lợi ở đây là khi muốn thay đổi giao diện cả trang, bạn không cần sửa từng đoạn code mà chỉ cần thay đổi thuộc tính trong CSS.

### 2.6.2.Định dạng Inline

Định dạng inline đánh mất nhiều ưu điểm của style sheets bởi vì nó trộn lẫn giữa nội dung và hình thức trình bày. Cho nên người ta ít sử dụng phương pháp này.

Để sử dụng địng dạng inline, ta sử dụng thuộc tính định dạng trong thẻ có liên quan. Thuộc tính định dạng có thể chứa bất kỳ thuộc tính CSS nào. Ví dụ sau chỉ ra cách thức thay đổi màu sắc và canh lề trái một đoạn:

<p style="color:sienna;margin-left:20px">This is a paragraph.</p>

### 2.6.3.Định dạng nhúng

Ta thấy rằng nếu ta làm việc với một trang web có nhiều thông tin như các trang web về tin tức, giới thiệu sản phẩm…thì số lần lặp lại của các tiêu đề, nội dung là rất lớn. Nếu ta xây dựng các định dạng kiểu dáng theo kiểu **Inline** như trên thì rất tốn thời gian và khả năng thay đổi giao diện về sau là rất khó khăn. Có một cách xây dựng định dạng kiểu dáng khác đó là ta xây dựng các định dạng thế vào các đối tượng có sẵn.

Chúng ta xem ví dụ sau đây:

|  |
| --- |
| <html>  <head>  <title>An Embedded Style </title>  <style type="text/css">  H1 { font-style:italic;background:yellow;color:red }  </style>  </head>  <body>  <h1>  In Line Style  </h1>  <p> |
| An inline style is included with the thẻ it affects.  </p>  <h1>  An Embedded Style  </h1>  <p> Embedded style sheet is a part of HTML document.  </p>  </body>  </html> |

Ta thấy rằng việc xây dựng đối tượng H1 với các tham số kiểu dáng màu nền…được sử dụng lại trong toàn trang, vì vậy khi bạn muốn thay đổi giao diện bạn chỉ cần vào thay đổi thông tin của H1 trên cùng mà không cần quan tâm đến các đối tượng sử dụng nó.

### 2.6.4.Định dạng bên ngoài

Định dạng bên ngoài là lý tưởng khi định dạng này được áp dụng cho nhiều trang. Với một định dạng ngoài, bạn muốn thay đổi giao diện của toàn bộ website chỉ cần thay đổi một file. Mỗi trang sử dụng link liên kết đến file định dạng bằng thẻ <link>. Thẻ <link> nằm trong phần head của trang:

<head>

<link rel="stylesheet" type="text/css" href="mystyle.css" />

</head>

Ở phần phía trước của mã lệnh, REL = stylesheet chỉ định trang này sẽ sử dụng một style sheet, HREF = “mystyle.css” chỉ định trang web này sẽ tham chiếu đến file mystyle.css, TYPE = “text/css” xác nhận rằng nó tham chiếu đến file .css nhằm xác định cụ thể đối tượng liên kết.

File định dạng ngoài có thể được viết bằng bất kỳ trình soạn thảo nào. File này không nên chứa bất kỳ thẻ HTML nào. Nếu không file định dạng của bạn sẽ không được lưu với đuôi file .CSS. Một ví dụ về file định dạng được hiển thị dưới đây:

hr {color:sienna;} p {margin-left:20px;}

body {background-image:url("images/back40.gif");}

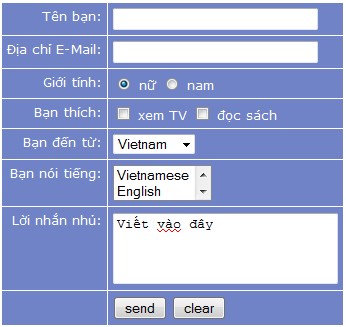
Không nên để khoảng trắng giữa giá trị của thuộc tính và đơn vị tính!

“margin-left: 20px” (thay vì “margin-left:20px”) sẽ làm việc được trong IE, nhưng không làm việc được trong Firefox hoặc Opera.

## 2.7.TẠO FORM

### 2.7.1.Tạo form

Khi vào một trang contact, bạn thường thấy một form như sau:



**Hình 2.5. Form lấy dữ liệu đầu vào**

Dùng form này, bạn có thể ghi vào sổ lưu bút, gửi ý kiến tới Web Server xử lí…Để có được form như trên, bạn viết code HTML như sau:

<form method="post" action="mailto:webmaster@yahoo.com">

Tên bạn: <input type="text" size="30" name="ten">

E-Mail: <input type="text" size="30" name="e-mail">

Giới tính:<input type="radio" value="f" name="sex" checked> nữ <input type ="radio" value="m" name="sex"> nam

Bạn thích:<input type="checkbox" value="Xem TV" name="TV"> xem TV <input type="checkbox" value="Doc sach" name="Book"> đọc sách

Bạn đến từ: <select name="land"><option value="VN">Vietnam <option value="USA"> USA <option value="De"> Germany</select>

Bạn nói tiếng: <select name="language" size="2"><option value="Viet">Viet

<option value="Engl"> Anh <option value="Deutsch"> Duc</select>

Lời nhắn nhủ: <textarea cols="10" rows="3" name="tin">Viết vào đây</textarea>

<input type="submit" value="send">

<input type="reset" value="clear">

</form>

Căp̣ thẻ để taọ Form la ̀<FORM...>...</FORM>.

Mọi thaǹ h phâǹ của form như sẽ triǹ h baỳ dưới đây đều phaỉ năm̀ trong phaṃ vi giới haṇ bởi cặp the ̉ naỳ .

Thuộc tiń h METHOD chỉ ra phương thức trao đổi dữ liêụ giưã trình duyệt Web và maý chủ Web. Có 2 phương thưć la ̀POST, GET. Đối với form phương thức thươǹ g la ̀POST.

Thuộc tính ACTION để thiết lâp̣ điạ chỉ URL của chương trình sẽ nhâṇ và xử lý dữ liêụ gửi từ form.

**2.7.2.Các đối tượng trên form**

### a. TextBox

TextBox là nơi để nhập dữ liệu kiểu xâu ký tự. TextBox được hiển thị như sau:



Thẻ được dùng để tạo ra TextBox là:

<INPUT TYPE="TEXT" NAME="*Tên*" SIZE="n">

Trong đó TEXT là từ khoá ưń g với kiểu là Text box, *Tên* là tên của hộp văn ban̉ naỳ còn n là một số nguyên chỉ ra chiêù dài cuả trường. *Tên* phaỉ duy nhât́ trong trang, không được trùng nhau. **b. Password**

Password là nơi để nhập mật khẩu kiểu xâu ký tự. Password được hiển thị như sau:



Thẻ để tạo ra nó là:

<INPUT TYPE="PASSWORD" SIZE="n">

Trong đó PASSWORD la ̀ từ khoa ́ để tạo hôp̣ mâṭ khâủ , n la ̀ số nguyên chỉ ra chiều dài cuả hộp.

Hộp mâṭ khâủ khác hôp̣ văn ban̉ ở chỗ khi gõ các kí tự từ baǹ phím thì nó không hiên̉ thị kí tự tương ưń g trong khung ma ̀ thay băǹ g các dâú sao \*.

*Lưu ý:* Việc hiển thị kí tự dấu \* để thay thế chỉ có tać duṇ g che giấu với người nhiǹ tại maý cục bộ. Password vâñ được gửi vào maṇ g dưới dạng rõ, không mã hoá, có thể bị xem trôṃ.

Nhiêṃ vu ̣ ma ̃ la ̀ của giao thưć maṇ g. **c. Text Window**

Text Window dùng để hiển thị nhiều dòng văn bản. Text Window có dạng như sau:



Thẻ để tạo ra Text Window là:

<TEXTAREA NAME= “TenVung" ROWS=m COLS=n> Van ban hien thi</TEXTAREA>

Trong đó, *Tên vùng* la ̀ tên cuả vùng văn ban̉ , m va ̀ n la ̀ các số nguyên, chỉ ra chiều cao và chiều rộng của vuǹ g văn ban̉ . Nêú baṇ để trôń g phâǹ "văn ban̉ hiên̉ thị " thì sẽ không có văn ban̉ naò xuât́ hiện trong cưả sổ.

#### d. RadioButton

RadioButton là một nhóm nút tròn, chỉ cho phép bạn chọn một trong nhiều nút.

RadioButton có dạng như sau:



Thẻ để tạo ra RadioButton là:

<input type="radio" value="f" name="sex" checked> n&#7919;

<input type="radio" value="m" name="sex"> nam

## CÂU HỎI

1. HTML là gì?
2. Làm thế nào để tạo một trang web đầu tiên bằng HTML?
3. Làm thế nào để đưa trang web của bạn lên mạng Internet cho người khác xem?
4. Làm thế nào để tạo một link từ văn bản A đến văn bản B?
5. Làm thế nào để thay đổi màu sắc, hình dạng của link?
6. CSS là gì?
7. Làm thế nào để tạo một hiệu ứng CSS trong trang HTML?
8. Làm thế nào để viết được tiếng Việt trong văn bản HTML?

# CHƯƠNG 3

**ASP.NET (ACTIVE SERVER PAGE .NET)**

## MỤC TIÊU

1. Về kiến thức
   * Tìm hiểu môi trường phát triển ứng dụng web với Visual Studio.Net 2005
   * Làm quen với các đối tượng ASP.NET Server Control
2. Về nội dung

Vận dụng các kiến thức đã học để làm việc với các ứng dụng web:

* + Sử dụng được các đối tượng ASP.NET Server Control trong bộ Visual Studio 2005.
  + Tạo và tổ chức một ứng dụng web sử dụng các điều khiển Web Server Controls

## HÌNH THỨC VÀ PHƯƠNG PHÁP GIẢNG DẠY

* Trình chiếu power point
* Phương pháp dạy học theo dự án
* Thực nghiệm kết hợp với máy tính
* Bài tập về nhà

## TÀI LIỆU THAM KHẢO

1. Nguyễn Phương Lan, Lê Hữu Đạt (2001), ASP.NET: Giáo trình tin học lý thuyết và bài tập, NXB Giáo dục Tp.HCM.
2. Dương Quang Thiện (2005), .NET Toàn Tập - Lập Trình Web Dùng ASP.NET Và

C# - Tập 3

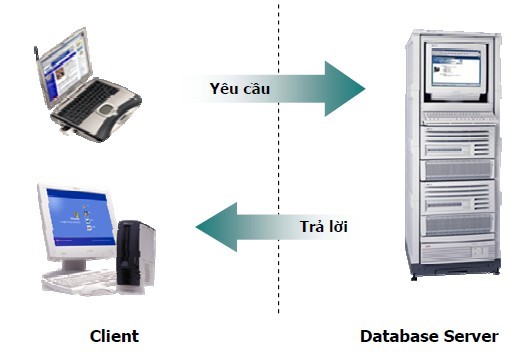
1. Dương Quang Thiện (2005), .NET Toàn Tập - Lập Trình Web Dùng ASP.NET Và

C# - Tập 4

1. [www.microsoft.com/net/](http://www.microsoft.com/net/)

## 3.1.TỔNG QUAN VỀ LẬP TRÌNH ỨNG DỤNG WEB

### 3.1.1.Mô hình ứng dụng 2 lớp



**Hình 3.1.Mô hình ứng dụng 2 lớp**

Đây là một dạng mô hình đơn giản, khá phổ biến của một ứng dụng phân tán. Trong mô hình này, việc xử lý dữ liệu được thực hiện trên Database Server, việc nhận và hiển thị dữ liệu được thực hiện ở Client.

### a. Ưu điểm

* Dữ liệu tập trung  đảm bảo dữ liệu được nhất quán.
* Dữ liệu được chia sẻ cho nhiều người dùng.

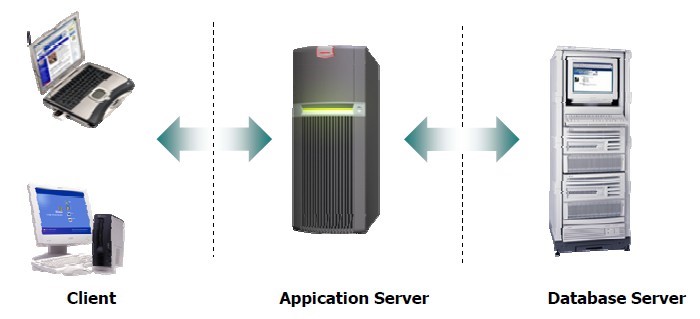
### b. Khuyết điểm

* Các xử lí tra cứu và cập nhật dữ liệu được thực hiện ở Database Server, việc nhận kết quả và hiển thị phải được thực hiện ở Client  khó khăn trong vấn đề bảo trì và nâng cấp
* Khối lượng dữ liệu truyền trên mạng lớn  chiếm dụng đường truyền, thêm gánh nặng cho Database Server

#### 3.1.2.Mô hình ứng dụng 3 lớp

Mô hình 2 lớp phần nào đáp ứng được các yêu cầu khắt khe của một ứng dụng phân tán, tuy nhiên, khi khối lượng dữ liệu lớn, ứng dụng đòi hỏi nhiều xử lý phức tạp, số người dùng tăng, mô hình 2 lớp không thể đáp ứng được.

Mô hình 3 lớp sử dụng thêm Application Server giữ nhiệm vụ tương tác giữa Client và Database server, giảm bớt các xử lý trên Database server, tập trung các xử lý nhận và hiển thị dữ liệu tại Application server.



**Hình 3.2.Mô hình ứng dụng 3 lớp**

### a. Ưu điểm

* Hỗ trợ nhiều người dùng
* Giảm bớt xử lí cho Client  Không yêu cầu máy tính ở Client có cấu hình mạnh
* Xử lí **nhận** và **hiển thị dữ liệu** tập trung tại Application Server  Dễ quản lý, bảo trì và nâng cấp
* Xử lí truy cập dữ liệu tập trung tại Database Server

**b. Khuyết điểm**

Phải sử dụng thêm một Application Server  Tăng chi phí

## 3.2.NGÔN NGỮ C#

Ngôn ngữ C# là ngôn ngữ được dẫn xuất từ C và C++, nhưng nó được tạo từ nền tảng phát triển hơn. Microsoft bắt đầu với công việc trong C và C++ và thêm vào những đặc tính mới để làm cho ngôn ngữ này dễ sử dụng hơn. Nhiều trong số những đặc tính này khá giống với những đặc tính có trong ngôn ngữ Java. Không dừng lại ở đó, Microsoft đưa ra những mục đích khi xây dựng ngôn ngữ này. Những mục đích này được tóm tắt như sau:

* C# là ngôn ngữ đơn giản
* C# là ngôn ngữ hiện đại
* C# là ngôn ngữ hướng đối tượng
* C# là ngôn ngữ mạnh mẽ và mềm dẻo
* C# là ngôn ngữ có ít từ khóa
* C# là ngôn ngữ hướng module
* C# sẽ trở nên phổ biến

Như chúng ta đã biết, C# là ngôn ngữ lập trình hướng đối tượng rất mạnh và công việc của người lập trình là kế thừa để tạo và khai thác các đối tượng. Do vậy, để nắm vững và phát triển tốt, người lập trình cần phải đi từ bước đầu tiên là tìm hiểu những thành phần cơ bản và cốt lõi nhất của ngôn ngữ.

### 3.2.1.Kiểu dữ liệu

C# là ngôn ngữ lập trình mạnh về kiểu dữ liệu, một ngôn ngữ mạnh về kiểu dữ liệu là phải khai báo kiểu của mỗi đối tượng khi tạo và trình biên dịch sẽ giúp cho người lập trình không bị lỗi khi chỉ cho phép một loại kiểu dữ liệu có thể được gán cho các kiểu dữ liệu khác. Kiểu dữ liệu của một đối tượng là một tín hiệu để trình biên dịch nhận biết kích thước của một đối tượng (kiểu *int* có kích thước là 4 byte) và khả năng của nó (một đối tượng *Button* có thể vẽ, phản ứng khi nhấn,..).

Tương tự như C++ hay Java, C# chia thành hai tập hợp kiểu dữ liệu chính: kiểu xây dựng sẵn mà ngôn ngữ cung cấp cho người lập trình, và kiểu do người dùng tự định nghĩa.

### Kiểu dữ liệu xây dựng sẵn

Ngôn ngữ C# đưa ra các kiểu dữ liệu xây dựng sẵn rất hữu dụng, phù hợp với một ngôn ngữ lập trình hiện đại, mỗi kiểu dữ liệu được ánh xạ đến một kiểu dữ liệu được hỗ trợ bởi hệ thống xác nhận ngôn ngữ chung (common language specification: CLS) trong MS.NET. Việc ánh xạ các kiểu dữ liệu nguyên thủy của C# đến các kiểu dữ liệu của .NET sẽ đảm bảo các đối tượng được tạo ra trong C# có thể được sử dụng đồng thời với các đối tượng được tạo ra trong bất kỳ ngôn ngữ khác được biên dịch bởi .NET.

Mỗi kiểu dữ liệu có một sự xác nhận và kích thước không thay đổi, không giống như C++, kiểu *int* trong C# luôn có kích thước là 4 byte bởi vì nó được ánh xạ bởi kiểu *Int32* trong .NET.

### Kiểu Số Kiểu Mô tả C# byte .NET

Byte 1 Byte Số nguyên dương không dấu từ 0-255

Char 2 Char Ký tự Unicode

Bool 1 Boolean Giá trị logic true/false

Sbyte 1 Sbyte Số nguyên có dấu từ -128 đến 127

Short 2 Int16 Số nguyên có dấu giá trị từ -32768 đến 32767

Int 4 Int32 Số nguyên có dấu

Float 4 Single Kiểu dấu chấm động, với 7 chữ số có nghĩa

Double Double Kiểu dấu chấm động có độ chính xác gấp đôi với 15, 16 chữ 8

số có nghĩa

Decimal 8 Decimal Có độ chính xác đến 28 con số

Long Int64 Kiểu số nguyên có dấu có giá trị trong khoảng: 8 -9.223.370.036.854.775.808 đến

9.223.370.036.854.775.807

***Ghi chú***: Kiểu giá trị logic chỉ có thể nhận được giá trị là true/false mà thôi. Một giá trị nguyên không thể gán vào một biến kiểu logic trong C# và không có bất cứ chuyển đổi ngầm định nào. Điều này khác với C/C++, cho phép biến kiểu logic được gán giá trị nguyên, khi đó giá trị nguyên 0 là *false* và các giá trị còn lại là *true*.

Ký tự Ý nghĩa

\’ Dấu nháy đơn

\” Dấu nháy kép

\\ Dấu chéo

\0 Ký tự null

\a Alert

\b Backspace

\f Sang trang form feed

\n Dòng mới

\r Đầu dòng

\t Tab ngang

\v Tab dọc

#### 3.2.2.Chuyển đổi các kiểu dữ liệu

Những đối tượng của một kiểu dữ liệu này có thể được chuyển sang những đối tượng của một kiểu dữ liệu khác thông qua cơ chế chuyển đổi tường minh hay ngầm định. Chuyển đổi ngầm định được thực hiện một cách tự động, trình biên dịch sẽ thực hiện công việc này. Còn chuyển đổi tường minh diễn ra khi chúng ta gán ép một giá trị cho kiểu dữ liệu khác.

Việc chuyển đổi giá trị ngầm định được thực hiện một cách tự động và đảm bảo là không mất thông tin. Ví dụ, chúng ta có thể gán một số kiểu *short* (2 byte) vào một số kiểu *int* (4 byte) một cách ngầm định. Sau khi gán hoàn toàn không mất dữ liệu vì bất cứ giá trị nào của *short* cũng thuộc về *int*.

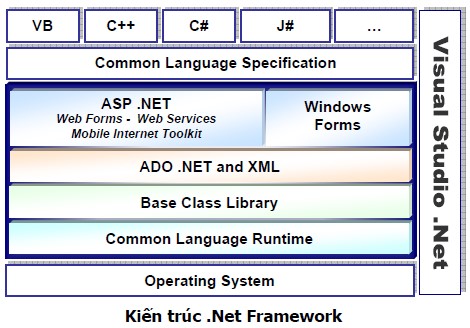
**3.3.GIỚI THIỆU VỀ ASP.NET?**

#### 3.3.1.Tìm hiểu về .Net Platform

**.Net Platform** bao gồm .Net Framework và những công cụ được dùng để xây dựng, phát triển ứng dụng và dịch vụ ASP.Net

Những sản phẩm công nghệ .Net của Microsoft bao gồm: MSN.Net, Office.Net, Visual Studio.Net và Windows Server 2003 được biết đến với tên gọi Windows .Net Server. Visual Studio .Net là bộ phần mềm được dùng để xây dựng và phát triển các ứng dụng bao gồm các ngôn ngữ lập trình: C++.Net, Visual Basic.Net, ASP.Net, C# và J#. Tất cả các ngôn ngữ này được xây dựng dựa trên nền .Net Framework, vì thế, bạn sẽ nhận thấy cú pháp cơ bản của những ngôn ngữ này tương tự nhau.

#### 3.3.2.Tìm hiểu về .Net Framework



### 1) Hệ điều hành

Cung cấp các chức năng xây dựng ứng dụng.

Với vai trò quản lý việc xây dựng và thi hành ứng dụng, .NET Framework cung cấp các lớp đối tượng (Class) để bạn có thể gọi thi hành các chức năng mà đối tượng đó cung cấp. Tuy nhiên, lời kêu gọi của bạn có được "hưởng ứng" hay không còn tùy thuộc vào khả năng của hệ điều hành đang chạy ứng dụng của bạn.

Các chức năng đơn giản như hiển thị một hộp thông báo (Messagebox) sẽ được .NET

Framework sử dụng các hàm API của Windows. Chức năng phức tạp hơn như sử dụng các COMponent sẽ yêu cầu Windows phải cài đặt Microsoft Transaction Server (MTS) hay các chức năng trên Web cần Windows phải cài đặt Internet Information Server (IIS).

Như vậy, bạn cần biết rằng lựa chọn một hệ điều hành để cài đặt và sử dụng .NET framework cũng không kém phần quan trọng. Cài đặt .NET Framework trên các hệ điều hành Windows 2000, 2000 Server, XP, XP.NET, 2003 Server sẽ đơn giản và tiện dụng hơn trong khi lập trình.

### 2) Common Language Runtime

Là thành phần "kết nối" giữa các phần khác trong .NET Framework với hệ điều hành.

Common Language Runtime (CLR) giữ vai trò quản lý việc thi hành các ứng dụng viết bằng .NET trên Windows. CLR sẽ thông dịch các lời gọi từ chương trình cho Windows thi hành, đảm bảo ứng dụng không chiếm dụng và sử dụng tràn lan tài nguyên của hệ thống. Nó cũng không cho phép các lệnh "nguy hiểm" được thi hành. Các chức năng này được thực thi bởi các thành phần bên trong CLR như Class loader, Just In Time compiler, Garbage collector,

Exception handler, COM marshaller, Security engine,…

Trong các phiên bản hệ điều hành Windows mới như XP.NET và Windows 2003, CLR được gắn kèm với hệ điều hành. Điều này đảm bảo ứng dụng viết ra trên máy tính của chúng ta sẽ chạy trên máy tính khác mà không cần cài đặt, các bước thực hiện chỉ đơn giản là một lệnh xcopy của DOS!

### 3) Bộ thư viện các lớp đối tượng

Nếu phải giải nghĩa từ "Framework" trong thuật ngữ .NET Framework thì đây là lúc thích hợp nhất. Framework chính là một tập hợp hay thư viện các lớp đối tượng hỗ trợ người lập trình khi xây dựng ứng dụng. Có thể một số người trong chúng ta đã nghe qua về MFC và JFC. Microsoft Foundation Class là bộ thư viện mà lập trình viên Visual C++ sử dụng trong khi Java Foundation Class là bộ thư viện dành cho các lập trình viên Java. Giờ đây, có thể coi .NET Framework là bộ thư viện dành cho các lập trình viên .NET.

### Base class library – thư viện các lớp cơ sở

Đây là thư viện các lớp cơ bản nhất, được dùng trong khi lập trình hay bản thân những người xây dựng .NET Framework cũng phải dùng nó để xây dựng các lớp cao hơn.

Ví dụ các lớp trong thư viện này là String, Integer, Exception,…

### ADO.NET và XML

Bộ thư viện này gồm các lớp dùng để xử lý dữ liệu. ADO.NET thay thế ADO để trong việc thao tác với các dữ liệu thông thường. Các lớp đối tượng XML được cung cấp để bạn xử lý các dữ liệu theo định dạng mới: XML. Các ví dụ cho bộ thư viện này là

SqlDataAdapter, SqlCommand, DataSet, XMLReader, XMLWriter,…

### ASP.NET

Bộ thư viện các lớp đối tượng dùng trong việc xây dựng các ứng dụng Web.

ASP.NET không phải là phiên bản mới của ASP 3.0. Ứng dụng web xây dựng bằng

ASP.NET tận dụng được toàn bộ khả năng của .NET Framework. Bên cạnh đó là một "phong cách" lập trình mới mà Microsoft đặt cho nó một tên gọi rất kêu: code behind. Đây là cách mà lập trình viên xây dựng các ứng dụng Windows based thường sử dụng – giao diện và lệnh được tách riêng. Tuy nhiên, nếu bạn đã từng quen với việc lập trình ứng dụng web, đây đúng là một sự "đổi đời" vì bạn đã được giải phóng khỏi mớ lệnh HTML lộn xộn tới hoa cả mắt.

Sự xuất hiện của ASP.NET làm cân xứng giữa quá trình xây dựng ứng dụng trên Windows và Web. ASP.NET cung cấp một bộ các Server Control để lập trình viên bắt sự kiện và xử lý dữ liệu của ứng dụng như đang làm việc với ứng dụng Windows. Nó cũng cho phép chúng ta chuyển một ứng dụng trước đây viết chỉ để chạy trên Windows thành một ứng dụng Web khá dễ dàng. Ví dụ cho các lớp trong thư viện này là WebControl, HTMLControl,

…

### Web services

Web services có thể hiểu khá sát nghĩa là các dịch vụ được cung cấp qua Web (hay Internet). Dịch vụ được coi là Web service không nhằm vào người dùng mà nhằm vào người xây dựng phần mềm. Web service có thể dùng để cung cấp các dữ liệu hay một chức năng tính toán.

Ví dụ, công ty du lịch của bạn đang sử dụng một hệ thống phần mềm để ghi nhận thông tin về khách du lịch đăng ký đi các tour. Để thực hiện việc đặt phòng khách sạn tại địa điểm du lịch, công ty cần biết thông tin về phòng trống tại các khách sạn. Khách sạn có thể cung cấp một Web service để cho biết thông tin về các phòng trống tại một thời điểm. Dựa vào đó, phần mềm của bạn sẽ biết rằng liệu có đủ chỗ để đặt phòng cho khách du lịch không? Nếu đủ, phần mềm lại có thể dùng một Web service khác cung cấp chức năng đặt phòng để thuê khách sạn. Điểm lợi của Web service ở đây là bạn không cần một người làm việc liên lạc với khách sạn để hỏi thông tin phòng, sau đó, với đủ các thông tin về nhiều loại phòng người đó sẽ xác định loại phòng nào cần đặt, số lượng đặt bao nhiêu, đủ hay không đủ rồi lại liên lạc lại với khách sạn để đặt phòng. Đừng quên là khách sạn lúc này cũng cần có người để làm việc với nhân viên của bạn và chưa chắc họ có thể liên lạc thành công.

Web service được cung cấp dựa vào ASP.NET và sự hỗ trợ từ phía hệ điều hành của Internet Information Server.

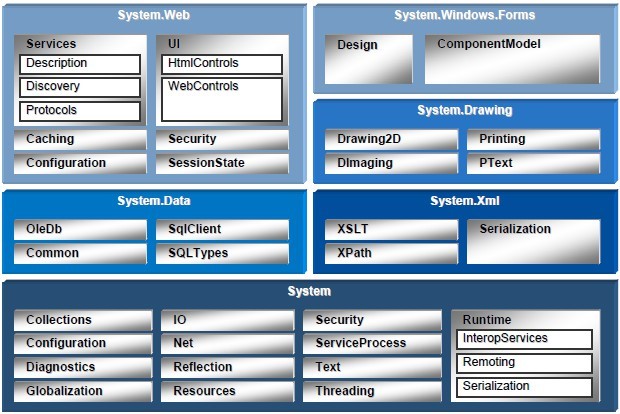
### Window form

Bộ thư viện về Window form gồm các lớp đối tượng dành cho việc xây dựng các ứng dụng Windows based. Việc xây dựng ứng dụng loại này vẫn được hỗ trợ tốt từ trước tới nay bởi các công cụ và ngôn ngữ lập trình của Microsoft. Giờ đây, ứng dụng chỉ chạy trên Windows sẽ có thể làm việc với ứng dụng Web dựa vào Web service. Ví dụ về các lớp trong thư viện này là: Form, UserControl,…

### 4) Phân nhóm các lớp đối tượng theo loại

Một khái niệm không được thể hiện trong hình vẽ trên nhưng cần đề cập đến là Namespace. Đây là tên gọi một nhóm các lớp đối tượng phục vụ cho một mục đích nào đó. Chẳng hạn, các lớp đối tượng xử lý dữ liệu sẽ đặt trong một namespace tên là Data. Các lớp đối tượng dành cho việc vẽ hay hiển thị chữ đặt trong namespace tên là Drawing.

Một namespace có thể là con của một namespace lớn hơn. Namespace lớn nhất trong .NET Framework là System.



**Hình 3.4. Hệ thống tên miền (Namespace)**

Lợi điểm của namespace là phân nhóm các lớp đối tượng, giúp người dùng dễ nhận biết và sử dụng. Ngoài ra, namespace tránh việc các lớp đối tượng có tên trùng với nhau không sử dụng được. .NET Framework cho phép chúng ta tạo ra các lớp đối tượng và các namespace của riêng mình. Với hơn 5000 tên có sẵn, việc đặt trùng tên lớp của mình với một lớp đối tượng đã có là điều khó tránh khỏi. Namespace cho phép việc này xảy ra bằng cách sử dụng một tên đầy đủ để nói đến một lớp đối tượng. Ví dụ, nếu muốn dùng lớp WebControls, chúng ta có thể dùng tên tắt của nó là WebControls hay tên đầy đủ là System.Web.UI.WebControls.

Đặc điểm của bộ thư viện các đối tượng .NET Framework là sự trải rộng để hỗ trợ tất cả các ngôn ngữ lập trình .NET như chúng ta thấy ở hình vẽ trên. Điều này sẽ giúp những người mới bắt đầu ít bận tâm hơn trong việc lựa chọn ngôn ngữ lập trình cho mình vì tất cả các ngôn ngữ đều mạnh ngang nhau. Cũng bằng cách sử dụng các lớp đối tượng để xây dựng ứng dụng, .NET Framework buộc người lập trình phải sử dụng kỹ thuật lập trình hướng đối tượng.

#### 3.2.3.Tìm hiểu về ASP.Net

Từ khoảng cuối thập niên 90, ASP (Active Server Page) đã được nhiều lập trình viên lựa chọn để xây dựng và phát triển ứng dụng web động trên máy chủ sử dụng hệ điều hành Windows. ASP đã thể hiện được những ưu điểm của mình với mô hình lập trình thủ tục đơn giản, sử dụng hiệu quả các đối tượng COM: ADO (ActiveX Data Object) - xử lý dữ liệu, FSO (File System Object) - làm việc với hệ thống tập tin…, đồng thời, ASP cũng hỗ trợ nhiều ngôn ngữ: VBScript, JavaScript. Chính những ưu điểm đó, ASP đã được yêu thích trong một thời gian dài.

Tuy nhiên, ASP vẫn còn tồn đọng một số khó khăn như Code ASP và HTML lẫn lộn, điều này làm cho quá trình viết code khó khăn, thể hiện và trình bày code không trong sáng, hạn chế khả năng sử dụng lại code. Bên cạnh đó, khi triển khai cài đặt, do không được biên dịch trước nên dễ bị mất source code. Thêm vào đó, ASP không có hỗ trợ cache, không được biên dịch trước nên phần nào hạn chế về mặt tốc độ thực hiện. Quá trình xử lý Postback khó khăn, …

Đầu năm 2002, Microsoft giới thiệu một kỹ thuật lập trình Web khá mới mẻ với tên gọi ban đầu là ASP+, tên chính thức sau này là ASP.Net. Với ASP.Net, không những không cần đòi hỏi bạn phải biết các tag HTML, thiết kế web, mà nó còn hỗ trợ mạnh lập trình hướng đối tượng trong quá trình xây dựng và phát triển ứng dụng Web.

ASP.Net là kỹ thuật lập trình và phát triển ứng dụng web ở phía Server (Server-side) dựa trên nền tảng của Microsoft .Net Framework.

Hầu hết, những người mới đến với lập trình web đều bắt đầu tìm hiểu những kỹ thuật ở phía Client(Client-side) như: HTML, Java Script, CSS (Cascading Style Sheets). Khi Web browser yêu cầu một trang web (trang web sử dụng kỹ thuật client-side), Web server tìm trang web mà Client yêu cầu, sau đó gởi về cho Client. Client nhận kết quả trả về từ Server và hiển thị lên màn hình.

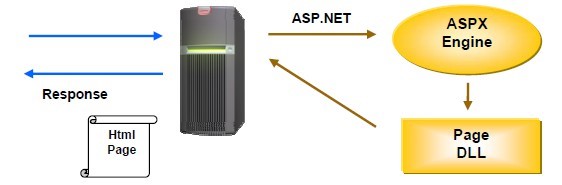
ASP.Net sử dụng kỹ thuật lập trình ở phía server thì hoàn toàn khác, mã lệnh ở phía server (ví dụ: mã lệnh trong trang ASP) sẽ được biên dịch và thi hành tại Web Server. Sau khi được Server đọc, biên dịch và thi hành, kết quả tự động được chuyển sang HTML/JavaScript/CSS và trả về cho Client. Tất cả các xử lý lệnh ASP.Net đều được thực hiện tại Server và do đó, gọi là kỹ thuật lập trình ở phía server.

## 3.3.NHỮNG ƯU ĐIỂM CỦA ASP.NET

* ASP.Net cho phép bạn lựa chọn một trong các ngôn ngữ lập trình mà bạn yêu thích:

VisualBasic.Net, J#, C#,…

* Trang ASP.Net được biên dịch trước. Thay vì phải đọc và thông dịch mỗi khi trang web được yêu cầu, ASP.Net biên dịch những trang web động thành những tập tin DLL mà Server có thể thi hành nhanh chóng và hiệu quả. Yếu tố này là một bước nhảy vọt đáng kể so với kỹ thuật thông dịch của ASP.

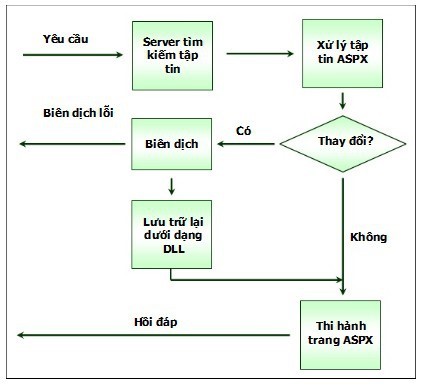


**Hình 3.5. Cách thức thi hành trang ASPX**

* ASP.Net hỗ trợ mạnh mẽ bộ thư viện phong phú và đa dạng của .Net Framework, làm việc với XML, Web Service, truy cập cơ sở dữ liệu qua ADO.Net, … ASPX và ASP có thể cùng hoạt động trong 1 ứng dụng.
* ASP.Net sử dụng phong cách lập trình mới: Code behide. Tách code riêng, giao diện riêng  Dễ đọc, dễ quản lý và bảo trì.
* Kiến trúc lập trình giống ứng dụng trên Windows.
* Hỗ trợ quản lý trạng thái của các control
* Tự động phát sinh mã HTML cho các Server control tương ứng với từng loại Browser Hỗ trợ nhiều cơ chế cache.
* Triển khai cài đặt
  + Không cần lock, không cần đăng ký DLL
  + Cho phép nhiều hình thức cấu hình ứng dụng Hỗ trợ quản lý ứng dụng ở mức toàn cục
  + Global.aspx có nhiều sự kiện hơn
  + Quản lý session trên nhiều Server, không cần Cookies

## 3.4.QUÁ TRÌNH XỬ LÍ TẬP TIN ASPX

Khi Web Server nhận được yêu cầu từ phía Client, nó sẽ tìm kiếm tập tin được yêu cầu thông qua chuỗi URL được gửi về, sau đó tiến hành xử lí theo sơ đồ sau:

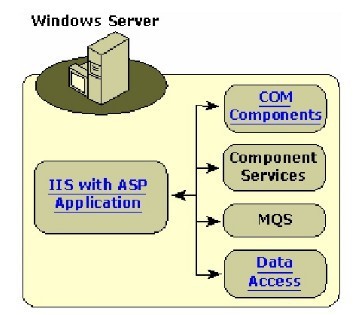


**Hình 3.6. Quá trình xử lí tập tin ASPX**

## 3.5.CÁCH THỨC HOẠT ĐỘNG CỦA ASP.NET TRÊN IIS

### 3.5.1.Internet Information Services (IIS)

IIS có thể được sử dụng như một Web server, kết hợp với ASP để xây dựng ứng dụng Web tận dụng các điểm mạnh của Server-side Script, COM component, … theo mô hình Client/Server.

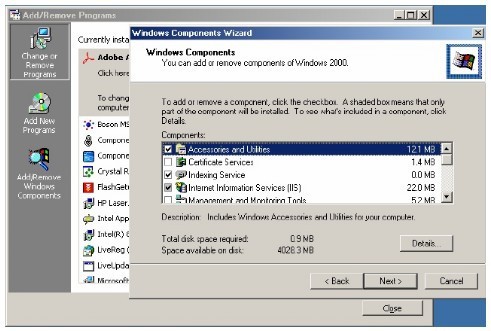


### 3.5.2.Cài đặt Web Server

Cài đặt Web Server trên Windows 2000/ Windows XP Professional.

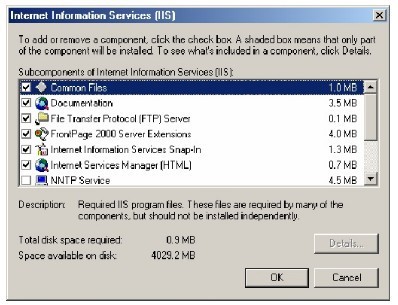
Bước 1: Chọn Control Panel | Add/Remove programs.

Bước 2: Chọn Add/Remove Windows Components



Bước 3: Đánh dấu vào mục Internet Information Services (IIS).

Bước 4: Chọn nút Details để chọn các mục chi tiết.



Bước 5: Chọn các mục cần cài đặt, trong đó bạn nhớ chọn:

* FrontPage 2000 Server Extensions.
* Internet Information Services Snap-In.
* Internet Services Manager (HTML).

Bước 6: Click OK để hệ thống tự cài đặt.

### 3.5.3.Kiểm tra kết quả cài đặt

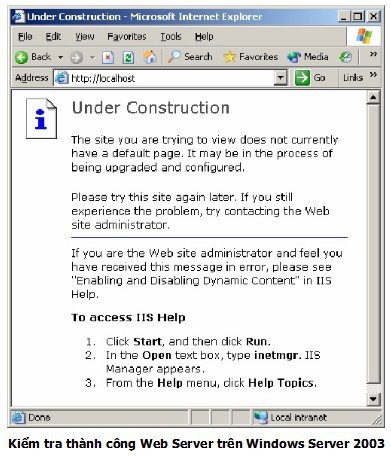
Khi đã cài đặt IIS, bạn có thể kiểm tra xem Web Server đã làm việc hay chưa. Mở Web Browser (internet explorer) và gõ: //localhost trong phần địa chỉ. Một khi Web server đã được cài đặt, một trang web mẫu sẽ được hiển thị.

* Localhost là địa chỉ máy cục bộ bạn đang làm việc. Nếu máy của bạn đang kết nối vào mạng LAN và có một địa chỉ IP, bạn có thể dùng địa chỉ này thay cho localhost.
* Khi gõ //localhost, bạn sẽ thấy trên thanh địa chỉ tự động đổi thành: [http://localhost.](http://localhost/)

HTTP là giao thức mặc định được dùng trên Internet. Vì HTTP là giao thức thuộc bộ TCP/IP nên bạn cần có địa chỉ IP để các máy tính khác trong mạng có thể truy cập được đến địa chỉ trang web của bạn.

* Sau khi cài đặt Web Server, mặc định trên ổ đĩa C:\ sẽ có sẵn thư mục C:\inetpub\wwwroot. Đây là thư mục Web server mặc định ánh xạ vào //localhost, do đó các trang web đặt trong wwwroot có thể được truy cập bởi các máy tính khác.

Để kiểm tra, hãy tạo một trang web và đặt vào C:\inetpub\wwwroot. Trên IE, gõ địa chỉ //localhost/<tên file HTML>.



## 3.6.TẠO TRANG WEB ASP.NET ĐẦU TIÊN

### 3.6.1.Tạo trang web ASP.NET đầu tiên

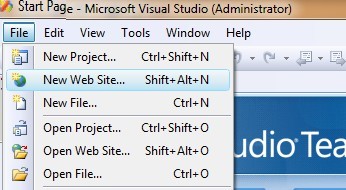
Ta sử dụng Microsoft Visual Studio 2005 để soạn các trang ASP.NET và qua đó sử dụng giao diện Graphic để việc phát triển web được dễ dàng và linh động.

Mục đích: Hiển thị hàng chữ “Welcome to ASP.NET” trong trang ASP.NET đầu tiên.

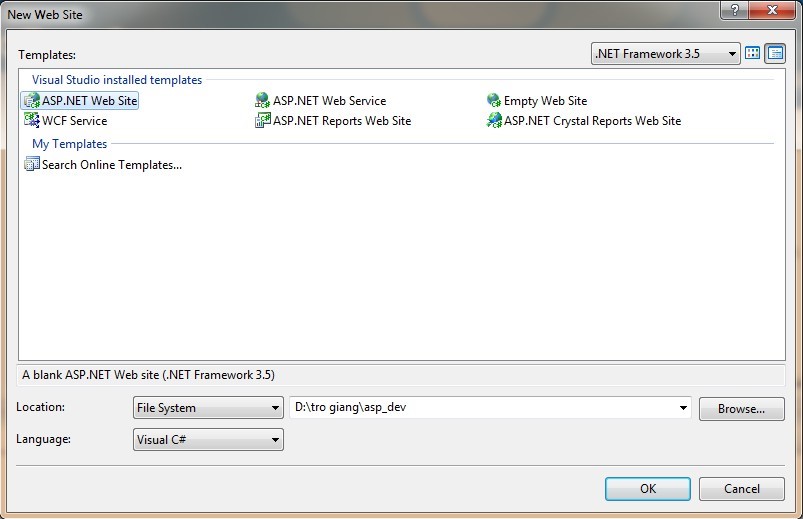
Các bước thứ tự như sau:

Bước 1: Khởi động Microsoft Visual Studio 2008

Bước 2: Trên thanh menu chọn File  New Website

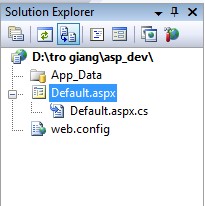


Bước 3: Chọn ASP.NET Web Site, nên lưu ý kiểm tra hàng chữ trong hộp Location để xem dự án được tạo ra đúng chỗ ta mong muốn hay chưa?



Bước 4: Nhấn OK

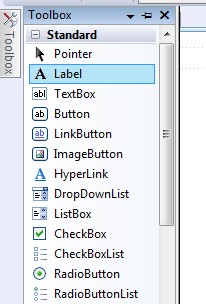
Sau khi tạo xong, trên phần Solution Explorer (thường ở góc trên bên phải màn hình) sẽ hiển thị như sau:



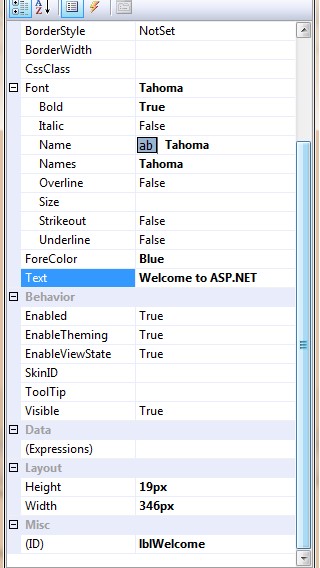
Trong đó:

* App\_Data: Thư mục lưu nội dung của các lớp dùng chung sẽ viết sau này
* Default.aspx, Default.aspx.cs: đây là một trang mặc định, ta có thể xoá bỏ và thêm mới một trang theo ý của ta
* Web.config: tập tin cấu hình

Bước 5: Mở hộp công cụ (ToolBox) và nhấp đôi chuột vào công cụ Label, một cách thông dụng khác là ta có thể kéo thả (drag and drop) công cụ Label vào trong phần Design.



Bước 6: Bố trí các đặc tính của công cụ Label đó như sau:



Bước 7: Cuối cùng ta dùng trình duyệt (IE hoặc Firefox) để hiển thị trang web vừa tạo xong.

### 3.6.2.Phân loại tập tin trong ASP.NET

Thông thường một một trang web ASP.NET gồm hai file chính, file có đuôi mở rộng aspx và file có đuôi là aspx.cs. Nhiệm vụ chính của file có đuôi aspx là tạo giao diện người dùng còn file có đuôi aspx.cs là tập hợp các mã lệnh (C#, VB, C++) xử lý các sự kiện trên form. Sau đây là một ví dụ về nhiệm vụ của từng trang như sau:

Trang tạo giao diện người dùng .aspx:

<%@ Page Language="C#"

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title>Welcome</title>

</head>

<body>

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

<p align = center >

<asp:Label ID="lblWelcome" runat="server" Font-Bold="True" Font-

Names="Tahoma"

ForeColor="Blue" Height="19px" Text="Welcome to ASP.NET"

Width="346px"></asp:Label>

</p>

</form>

</body>

</html>

<%@ Page Language="C#" được gọi là Page Directives: phần này cung cấp cho ASP.NET những thông tin đặc biệt để ASP.NET dùng C# làm ngôn ngữ lập trình mặc định cho trang web.

Phần còn lại gọi là Code Declaration Block giống như mã ở Client Side nhưng khác một chút là có kèm theo **runat = “server”** chỉ thị cho ASP.NET biết thi hành trang này ở Server side.

Trang xử lí các sự kiện trên form (.aspx.cs):

public partial class \_Default : System.Web.UI.Page

{

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

{

lblWelcome.Text = "Welcome to ASP.NET Class";

}

}

Phần mã này tạo ra một thủ tục có tên là Page\_Load mặc định cho các trang ASP.NET giống như Form\_Load trong việc lập trình dùng ngôn ngữ Visual Basic với sự kiện Load, khi trang này được đưa ra trình bày, ASP.NET sẽ gắn hàng chữ “Welcome to ASP.NET class” vào hộp có nhãn hiệu lblWelcome.

### Tiến trình xử lý

Bây giờ, ta sẽ tìm hiểu về tiến trình xử lí trang web như thế nào khi có khách viếng thăm trang ASP.NET của chúng ta.

Khi có yêu cầu tham khảo trang web của ta, trước hết ASP.NET sẽ biên dịch mã ở phần Code Declaration Block trong trang ASP.NET. Nếu để ý một chút ta sẽ thấy có một sự trì hoãn mặc dù không đáng kể trước khi Browser hiển thị trang web đó, nhưng nếu cũng chính trang web đó được viếng thăm lần thứ nhì hoặc những lần sau đó thì sự trì hoãn ta nhận thấy trước kia sẽ biến mất, lý do là vì trang đó đã được biên dịch rồi, ASP.NET chỉ việc dùng mà thôi chứ không cần phải biên dịch lại lần nữa, do đó hiệu suất sẽ gia tăng.

Sau khi trang web được biên dịch, ASP.NET bắt đầu tiến trình xử lí tất cả mã nguồn, kể cả các sự kiện ví như ta gõ vài chữ trong hộp TextBox hoặc hoặc click chuột vào một nút nào đó thì ASP.NET sẽ nghiên cứu sự kiện đó để quyết định cách phản ứng và thi hành để đáp ứng lại sự kiện theo kế hoạch lập trình đã qui định trước.

Tiếp theo, ASP.NET sẽ biến đổi tất cả các Server Controls trong trang web ra thành những yếu tố HTML tương đương ví dụ như biến đổi <asp:Label> ra thành HTML <span></span> để hiển thị hàng chữ “Welcome to ASP.NET Class” như ví dụ ở trên.

Cuối cùng, ASP.NET sẽ gửi kết quả dưới dạng HTML tới Client Browser, Client

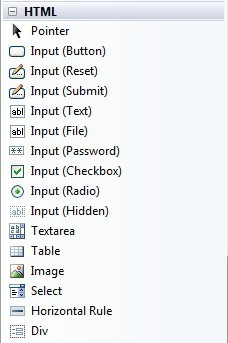
Browser chỉ nhận được trang web dưới dạng phù hợp với tiêu chuẩn của HTML, trong đó ASP.NET dấu đi các mã nguồn hoặc các Control thuộc phạm vi Server.

Như vậy, ta có thể dùng bất cứ một web browser nào cũng có thể hiển thị các trang ASP.NET.

## 3.7.LÀM VIỆC VỚI CÁC ASP.NET SERVER CONTROL

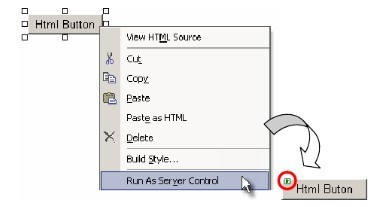
### 3.7.1.HTML Controls

Điều khiển HTML (tag HTML) trong trang ASP.NET có thể xem như những chuỗi văn bản bình thường. Để có thể được sử dụng lập trình ở phía Server, ta gán thuộc tính **runat = “Server”** cho các điều khiển HTML đó. Những điều khiển HTML (tag HTML) có thuộc tính runat = “Server” được gọi là HTML Server Controls.

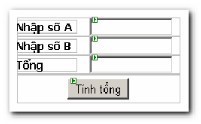


**Hình 3.8. Các điều khiển HTML trên thanh công cụ**

Trong MS Visual Studio 2005, để chuyển các điều khiển HTML thành điều khiển HTML Server, ta chọn Run As Server Control từ menu ngữ cảnh.



### Hình 3.9. Chuyển điều khiển HTML thành điều khiển HTML Server. Ví dụ: Các điều khiển HTML: Label, Textbox, Button



Bạn có thể tham khảo thêm phần xử lí sự kiện trên bằng ngôn ngữ C# và VisualBasic.

Ngôn ngữ C#:

Private void btn\_Tong\_ServerClick(object sender, EventArgs e)

{

int tong =Convert.ToInt32(txtA.Text) + Convert.ToInt32(txtB.Text);

txtTong.Text = tong.ToString();

}

Ngôn ngữ Visual Basic:

Private Sub butTong\_

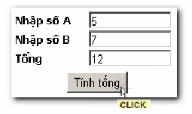
**ServerClick**

(

…) …

txtTong.Value = Val(txtA.Value) + Val(txtB.Value)

End Sub



Khi thi hành ứng dụng.

#### 3.7.2.Web Server Controls

Dưới đây là các lý do bạn nên sử dụng ASP.Net Web Control:

− Đơn giản, tương tự như các điều khiển trên Windows Form.

− Đồng nhất: Các điều khiển Web server có các thuộc tính giống nhau  dễ tìm hiểu và sử dụng.

− Hiệu quả: Các điều khiển Web Server tự động phát sinh ra các tag HTML theo từng loại Browser.

### Bảng liệt kê các thuộc tính chung của các Web Control Thuộc tính Kiểu Ý nghĩa

(ID) Chuỗi Qui định tên của điều khiển. Tên của điều khiển là duy

nhất.

AccessKey String Qui định ký tự để di chuyển nhanh đến điều khiển - ký tự xử lý phím nóng.

Attributes AttributeCollectio Tập hợp các thuộc tính của điều khiển HTML. n

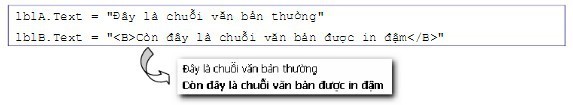
BackColor Color Qui định màu nền của điều khiển.

BorderColor Color Qui định màu đường viền của điều khiển.

#### *3.7.2.1.Label*

Label thường được sử dụng để hiển thị và trình bày nội dung trên trang web. Nội dung được hiển thị trong label được xác định thông qua thuộc tính Text. Thuộc tính text có thể nhận và hiển thị nội dung với các tag HTML.

Ví dụ:



### Cách tạo

Mở hộp công cụ (ToolBox) và nhấp đôi chuột vào công cụ Label, hoặc kéo thả (drag and drop) công cụ Label vào trong phần Design.

Cú pháp hiển thị ở chế độ Source:

<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>

**Chức năng**

Hiển thị dữ liệu

### Thuộc tính Thuộc tính Sử dụng

CssClass Định dạng hiển thị của label thông qua một phương thức css

Enabled True = hiển thị, false = vô hiệu hóa label

Text Thiết lập hay trả về giá trị hiển thị trên label

ToolTip Hiển thị phần chú thích của label khi đưa con trỏ đến label khi đang thực hiện chương trình

Visible True = hiển thị, false = ẩn

#### *3.7.2.2.HyperLink*

Điều khiển này được sử dụng để tạo ra các liên kết siêu văn bản. **Các thuộc tính**

* ImageUrl: qui định hình hiển thị trên điều khiển
* Text: chuỗi văn bản sẽ được hiển thị trên điều khiển. Trong trường hợp cả hai thuộc tính ImageUrl và Text đều được thiết lập, thuộc tính ImageUrl sẽ được ưu tiên, thuộc tính Text sẽ được hiển thị như Tooltip.
* NavigateUrl: đường dẫn cần liên kết đến.
* Target: xác định cửa sổ sẽ hiển thị cho mối liên kết.
  + \_blank: hiển thị trang liên kết ở một cửa sổ mới. o \_self: hiển thị trang liên kết tại chính cửa sổ chứa liên kết đó.
  + \_parent: hiển thị trang liên kết ở frame cha.

Ví dụ:

hplASP\_net.Text = “Trang chủ ASP.Net”

hplASP\_net.ImageUrl = “Hinh\Asp\_net.jpg”

hplASP\_net.NavigateUrl = “

[http://www.asp.ne](http://www.asp.net/)

[t](http://www.asp.net/)

[”](http://www.asp.net/)

hplASP\_net.Target = “\_blank”



Kết quả hiển thị trên trang web

#### *3.7.2.3.TextBox*

**Cách tạo TextBox**:

Mở hộp công cụ (ToolBox) và nhấp đôi chuột vào công cụ TextBox, một cách thông dụng khác là ta có thể kéo thả (drag and drop) công cụ TextBox vào trong phần Design.

Cú pháp hiển thị ở chế độ Source:

<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>

**Chức năng**: dùng để chứa dữ liệu nhập vào bởi người dùng và được lập trình trên server để thay đổi dữ liệu của nó thông qua các control khác hay chính nó (thuộc tính AutoPosBack=true với sự kiện TextChange).

**Các thuộc tính:**

### Thuộc tính Sử dụng

Text Lấy/đặt dữ liệu cho TextBox

TextMode SingleLine: Hiển thị và nhập liệu 1 dòng văn bản.

MultiLine: Hiển thị và nhập liệu nhiều dòng văn bản.

Password: Hiển thị dấu \* thay cho các ký tự có trong TextBox

Enable Enable/Disable TextBox

Visible Show/hide TextBox

ReadOnly Khi được thiết lập là **true**, ngăn không cho người dùng thay đổi dữ liệu trong TextBox

AutoPostBack Khi được thiết lập là **true**, mỗi khi người dùng thay đổi dữ liệu trong TexBox sẽ kích hoạt sự kiện Post-Back về Server. Giá trị

mặc định là false = không tự động Post-Back

MaxLength Qui định số ký tự tối đa được nhập vào cho Textbox

Rows Số dòng của TextBox, chỉ có chế độ TextMode = “MultiLine”

thì số dòng mới có thể khác 1

TabIndex Thứ tự ưu tiên được trỏ đến khi nhấn phím Tab

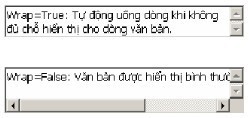
Wrap Thuộc tính này qui định việc hiển thị văn bản có được phép tự

động xuống dòng khi kích thước ngang của điều khiển không đủ

để hiển thị dòng nội dung văn bản. Giá trị mặc định của thuộc

tính này là true = tự động xuống dòng.

Ví dụ:



**Các sự kiện trên TextBox:**

TextChanged: Sự kiện khi có sự thay đổi dữ liệu trên TextBox

#### *3.7.2.4.Image* Cách tạo

Mở hộp công cụ (ToolBox) và nhấp đôi chuột vào công cụ Image, hoặc kéo thả (drag and drop) công cụ Image vào trong phần Design.

Cú pháp hiển thị ở chế độ Source:

<asp:Image ID="Image1" runat="server" ImageUrl="~/App\_Data/Koala.jpg" /> **Chức năng**

Control này dùng để hiển thị hình ảnh. Control này chỉ chứa một ảnh và ảnh này chỉ có thể được thay đổi thông qua các control khác. Vì bản thân control này không có sự kiện

Click để thay đổi ảnh, nếu muốn sử dụng ảnh có sự kiện Click thì dùng control ImageButton

### Thuộc tính

ImageURL: đường dẫn đến file ảnh cần hiển thị

AlternateText: chuỗi văn bản sẽ hiển thị khi tập tin được thiết lập trong thuộc tính ImageUrl không tồn tại.

ImageAlign: vị trí hiển thị giữa hình ảnh và nội dung văn bản o NotSet:



* Left:



* TextTop:



* Right:



#### *3.7.2.5.Button, ImageButton, LinkButton*

Các điều khiển Button, ImageButton, LinkButton mặc định đều là các nút Submit Button, mỗi khi được nhấn vào sẽ PostBack về Server.

Các thuộc tính chung của Button, ImageButton và LinkButton

### Thuộc tính Sử dụng

Text Chuỗi văn bản hiển thị trên điều khiển CommandName Tên lệnh. Được sử dụng trong sự kiện Command.

CommandArgument Thông tin bổ sung cho sự kiện Command.

CausesValidation Trang web mặc định kiểm tra tính hợp lệ dữ liệu mỗi khi được PostBack. Các điều khiển Button, ImageButton, LinkButton luôn PostBack về server mỗi khi được nhấn luôn kiểm tra tính hợp lệ của dữ liệu trên trang web.

Ngoài những thuộc tính trên, điều khiển ImageButton còn có các thuộc tính ImageUrl, ImageAlign và AlternateText như điều khiển Image.



#### *3.7.2.6.ListBox và DropdownList*

ListBox và DropdownList là điều khiển hiển thị danh sách lựa chọn mà người dùng có thể chọn một hoặc nhiều (chỉ dành cho ListBox). Các mục lựa chọn có thể được thêm vào danh sách thông qua lệnh hoặc ở cửa sổ thuộc tính (Property Windows). **Các thuộc tính**

* AutoPostBack: thuộc tính này qui định điều khiển có được phép tự động PostBack về

Server khi chỉ số các mục chọn bị thay đổi. Giá trị mặc định của thuộc tính này là False – không tự động PostBack.

* Items: đây là tập hợp chứa các mục chọn của điều khiển. Ta có thể thêm vào mục chọn vào thời điểm thiết kế thông qua cửa sổ ListItem Collection Editor, hoặc thông qua lệnh.
* Rows: qui định chiều cao của ListBox theo số dòng hiển thị.
* SelectionMode: thuộc tính này xác định cách thức chọn các mục trong trong ListBox. SelectionMode chỉ được phép thay đổi trong quá trình thiết kế, vào lúc thực thi chương trình, thuộc tính này chỉ đọc.
  + Single: chỉ được chọn một mục có trong danh sách (mặc định).
  + Multiple: cho phép chọn nhiều lựa chọn.

### Xử lí mục chọn

Các thuộc tính sau sẽ giúp xác định chỉ số, giá trị của mục đang được chọn. Trong trường hợp điều khiển cho phép chọn nhiều, ta duyệt qua các item trong tập hợp Items, sử dụng thuộc tính Selected của đối tượng Item để kiểm tra xem mục đó có được chọn hay không.

* SelectedIndex: cho biết chỉ số của mục được chọn. Trong trường hợp chọn nhiều mục, SelectedIndex trả về chỉ số mục chọn đầu tiên.
* SelectedItem: cho biết mục được chọn. Trong trường hợp chọn nhiều mục, SelectedItem sẽ trả về mục chọn đầu tiên.
* SelectedValue: cho biết giá trị của mục được chọn. Trong trường hợp chọn nhiều mục, SelectedValue sẽ trả về giá trị mục chọn đầu tiên.

#### *3.7.2.7.CheckBox, RadioButton*

**Các thuộc tính** o Checked: cho biết trạng thái của mục chọn – có được chọn hay không

* TextAlign: qui định vị trí hiển thị của điều khiển so với chuỗi văn bản

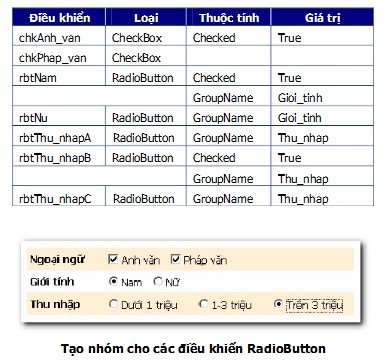


* AutoPostBack: thuộc tính này qui định điều khiển có được phép tự động PostBack về Server khi các mục chọn của điều khiển bị thay đổi. Giá trị mặc định của thuộc tính này là False – không tự động PostBack.
* GroupName (RadioButton): Tên nhóm. Thuộc tính này được sử dụng để nhóm các điều khiển RadioButton thành một nhóm.

**Ví dụ**:



Danh sách các điều khiển



#### *3.7.2.8.CheckboxList, RadioButtonList*

Hai điều khiển này được dùng để tạo ra một nhóm các CheckBox/RadioButton. Do đây là điều khiển danh sách nên nó cũng có thuộc tính Items chứa tập hợp các mục chọn như ListBox/DropDownList. Các thao tác trên tập hợp Items, xử lí mục chọn cũng tương tự như ListBox/DropDownList.

**Các thuộc tính**:

* RepeatColumns: qui định số cột hiển thị.
* RepeatDirection: qui định hình thức hiển thị.
* Vertical: theo chiều dọc
* Horizontal: theo chiều ngang
* AutoPostBack: thuộc tính này qui định điều khiển có được phép tự động PostBack về server khi các mục chọn của điều khiển bị thay đổi. Giá trị mặc định của thuộc tính này là false = không tự động PostBack.

### Ví dụ



Bạn có thể (tham khảo) xử lí sự kiện bằng C# như sau:

Private void rblThu\_nhap\_SelectedIndexChanged (object sender, EventArg e)

{

lblThu\_nhap.Text = “Bạn chọn thu nhập:” +

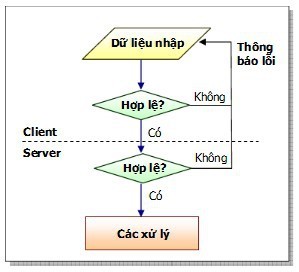
rblThu\_nhap.SelectedItem.Text;

}



#### 3.7.3.Validation Controls

Validation control kiểm tra tính đúng đắn của dữ liệu do client nhập vào trước khi được gửi về cho server. Mặc dù việc kiểm tra xảy ra ở client nhưng nó chỉ thực hiện khi có một sự kiện post-back xảy ra. Bạn có thể thêm một button để kiểm tra kết quả.



**Hình 3.11. Sơ đồ xử lí kiểm tra dữ liệu nhập tại Client và Server**

Như các bạn đã biết, mỗi khi PostBack về Server, trang Web luôn kiểm tra tính hợp lệ dữ liệu (nếu có yêu cầu khi thiết kế). Nếu dữ liệu không hợp lệ (bỏ trống, vi phạm miền giá trị, mật khẩu nhập lại không đúng, …), trang web sẽ không thể PostBack về Server.

### Các thuộc tính chung của các điều khiển ValidationControl Thuộc tính Ý nghĩa

ControlToValidate Tên điều khiển cần kiểm tra. Đây là thuộc tính mà các bạn phải xác định khi sử dụng Validation Control.

Text Chuỗi thông báo xuất hiện khi có lỗi.

ErrorMessage Chuỗi thông báo xuất hiện trong điều khiển Validation Summary. Giá trị này sẽ được hiển thị tại vị trí của điều khiển nếu chúng ta không gán giá trị cho thuộc tính Text.

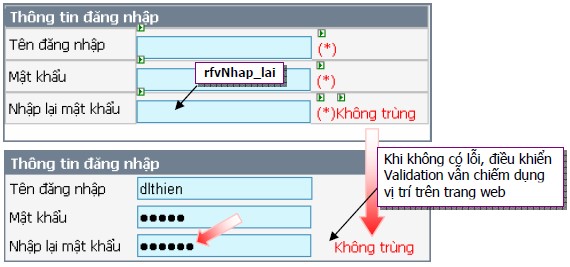
Display Qui định hình thức hiển thị:

* None: Không hiển thị thông báo lỗi (vẫn có kiểm tra dữ liệu)
* Static: Trong trường hợp không có vi phạm dữ liệu, điều khiển không có hiển thị nhưng vẫn chiếm vị trí như trong lúc thiết kế.
* Dynamic: Trong trường hợp không có vi phạm dữ liệu, điều khiển không chiếm dụng vị trí trên màn hình.

EnableClientScript Có cho phép thực hiện kiểm tra ở phía Client hay không. Giá trị mặc định là True - có kiểm tra.

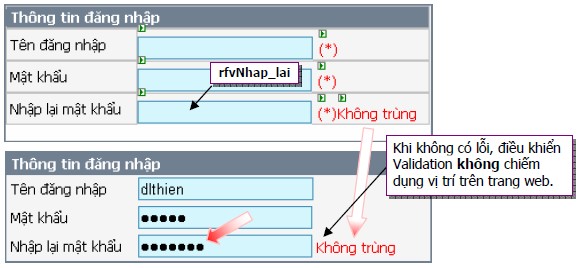
**Ví dụ:** Minh họa thuộc tính Display: Tại ô nhập lại mật khẩu, ta có hai điều khiển kiểm tra dữ liệu: Một điều khiển kiểm tra không được phép rỗng (rfvNhap\_lai), một điều khiển kiểm tra xem nhập lại mật khẩu có giống với mật khẩu đã nhập ở trên hay không .

rfvNhap\_lai.Display = Static



**Lựa chọn hình thức hiển thị lỗi**

rfvNhap\_lai.Display = Dynamic



**Lựa chọn hình thức hiển thị lỗi**

#### *3.7.3.1.RequiredFieldValidator*

Điều khiển này được dùng để kiểm tra giá trị trong điều khiển phải được nhập  Sử dụng điều khiển này để kiểm tra ràng buộc dữ liệu khác rỗng (bắt buộc nhập). **Thuộc tính**

InitialValue: Giá trị khởi động. Giá trị bạn nhập vào phải khác với giá trị của thuộc tính này. Giá trị mặc định của thuộc tính này là chuỗi rỗng.

#### *3.7.3.2.CompareValidator*

Điều khiển này được dùng để so sánh giá trị của một điều khiển với giá trị của một điều khiển khác hoặc một giá trị được xác định trước.

Thông qua thuộc tính Operator, chúng ta có thể thực hiện các phép so sánh như: =, <>, >, >=, <, <= hoặc dùng để kiểm tra kiểu dữ liệu (DataTypeCheck).

 Sử dụng điều khiển này để kiểm tra ràng buộc miền giá trị, kiểu dữ liệu, liên thuộc tính.

**Lưu ý:** Trong trường hợp không nhập dữ liệu, điều khiển sẽ không thực hiện kiểm tra vi phạm.

### Các thuộc tính

* ControlToCompare: Tên điều khiển cần so sánh giá trị. Nếu bạn chọn giá trị của thuộc tính Operator = DataTypeCheck thì không cần phải xác định giá trị cho thuộc tính này.
* Operator: Qui định phép so sánh, kiểm tra kiểu dữ liệu:
  + Equal: = (đây là giá trị mặc định) o GreaterThan: > o GreaterThanEqual: >= o LessThan: < o LessThanEqual: <= o Not Equal: <>
  + DataTypeCheck: kiểm tra kiểu dữ liệu
* Type: Qui định kiểu dữ liệu để kiểm tra hoặc so sánh.
  + String o Integer o Double o Date o Currency
* ValueToCompare: Giá trị cần so sánh. Trong trường hợp bạn xác định giá trị của cả 2 thuộc tính ControlToCompare và ValueToCompare thì giá trị của điều khiển được qui định bởi thuộc tính ControlToCompare được ưu tiên dùng để kiểm tra.

#### *3.7.3.3.RangeValidator*

Điều khiển này được dùng để kiểm tra giá trị trong điều khiển phải nằm trong đoạn [giá trị nhỏ nhất-giá trị lớn nhất]  Sử dụng điều khiển này để kiểm tra ràng buộc miền giá trị của dữ liệu.

**Lưu ý:** Trong trường hợp không nhập dữ liệu, điều khiển sẽ không thực hiện kiểm tra vi phạm.

### Các thuộc tính

* MinimumValue: giá trị nhỏ nhất
* MaximumValue: giá trị lớn nhất
* Type: Xác định kiểu để kiểm tra dữ liệu. Ta có thể thực hiện kiểm tra trên các kiểu dữ liệu sau: o Integer o Double o Date o Currency

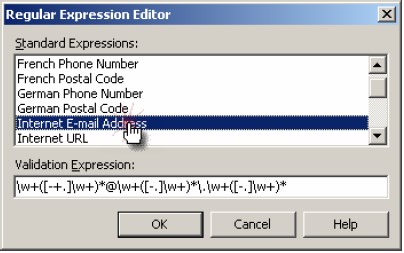
#### *3.7.3.4.RegularExpressionValidator*

Điều khiển này được dùng để kiểm tra giá trị của điều khiển phải theo mẫu được qui định trước: địa chỉ email, số điện thoại, mã vùng, số chứng minh thư, …

**Lưu ý:** Trong trường hợp không nhập dữ liệu, điều khiển sẽ không thực hiện kiểm tra vi phạm.

### Thuộc tính

ValidationExpression: Qui định mẫu kiểm tra dữ liệu



**Hộp thoại RegularExpression**

Bảng mô tả các ký hiệu thường sử dụng trong ValidationExpression



#### *3.7.3.5.CustomValidator ( đọc thêm)*

Đây là hình thức xác nhận dữ liệu nhập vào theo cách của người lập trình, bạn có thể tạo ra công thức kiểm tra và các điều kiện hiển thị như các đối tượng xác nhận trên. Xét ví dụ sau:

1.<asp:Label ID="Label1" runat="server" Text="Họ tên: "> </asp:Label>

2.<asp:TextBox ID="txtHoVaTen" runat="server"></asp:TextBox>

3.<asp:CustomValidator ID="CustomValidator1" runat="server"

ErrorMessage="Chưa nhập họ tên" ClientValidationFunction="xuLy"

ControlToValidate="txtHoVaTen"

ValidateEmptyText="True"></asp:CustomValidator><br />

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

<script> function xuLy(){ if(document.getElementById('<%=txtHoVaTen.ClientID

%>').value.length > 50){

alert('Bạn nhập tên dài quá 50 ký tự'); return false

}

}

</script>

Ví dụ trên ta thực hiện kiểm tra một đối tượng nhập họ tên người dùng, và kiểm tra dữ liệu nhập vào có dài quá 50 ký tự hay không.

Các bước thực hiện vẫn như các đối tượng trên bạn nên chú ý một số thuộc tính mà nó cần như:

* ControlToValidate: Xác định đối tượng cần kiểm tra. o ValidateEmptyText: Khi nhập dữ liệu được trống thì có kiểm tra hay không.
* ClientValidationFunction: Đây là thuộc tính quan trọng nhất, nó cho phép bạn kiểm tra thông tin nhập bằng các lệnh script, bạn nhập tên hàm mà mình viết cho đối tượng này ở đây, trong ví dụ trên hàm của tôi là xử lý và nó thực hiện kiểm tra số ký tự nhập vào phải nhỏ hơn 50. Khi thực hiện thuộc tính này của đối tượng bạn phải biết dùng script.

**Sự kiện** o ServerValidate: Đặt các xử lí kiểm tra dữ liệu trong sự kiện này. Việc kiểm tra này được thực hiện ở Server.

Ví dụ: Xử lí kiểm tra dữ liệu nhập tại điều khiển txtSoA có phải là số chẵn hay không.

protected void CustomValidator\_ServerValidate(object source,

ServerValidateEventArgs args)

{

int tam = Convert.ToInt32(TextBox1.Text); if (tam % 2 == 0) args.IsValid = true;

else args.IsValid = false; }

#### *3.7.3.6.ValidationSummary*

Điều khiển này được dùng để hiển thị ra bảng lỗi - tất cả các lỗi hiện có trên trang

Web. Nếu điều khiển nào có dữ liệu không hợp lệ, chuỗi thông báo lỗi - giá trị thuộc tính

ErrorMessage của Validation Control sẽ được hiển thị. Nếu giá trị của thuộc tính ErrorMessage không được xác định, thông báo lỗi đó sẽ không được xuất hiện trong bảng lỗi.

### Các thuộc tính

* HeaderText: Dòng tiêu đề của thông báo lỗi
* ShowMessageBox: Qui định bảng thông báo lỗi có được phép hiển thị như cửa sổ MessageBox hay không. Giá trị mặc định của thuộc tính này là False - không hiển thị.



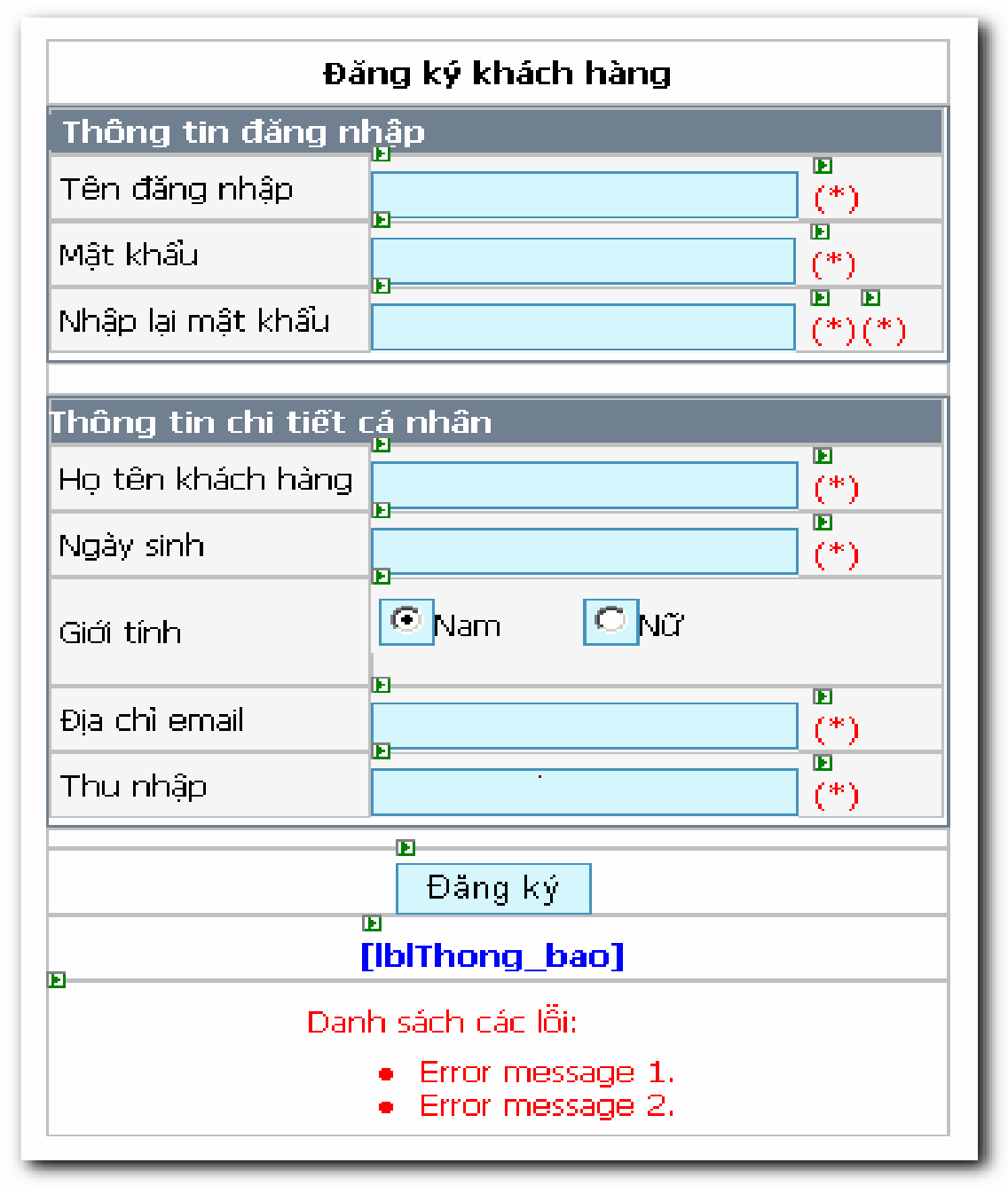
* ShowSummary: Qui định bảng thông báo lỗi có được phép hiển thị hay không.

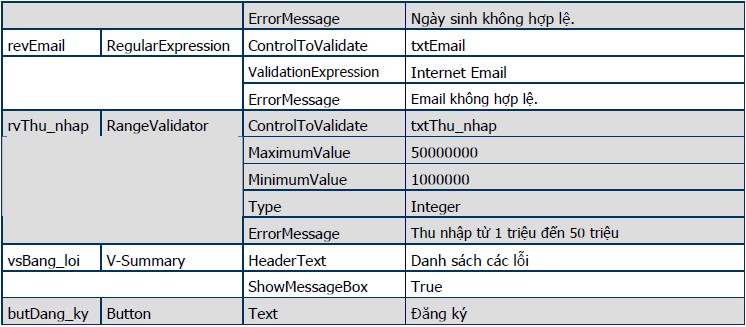
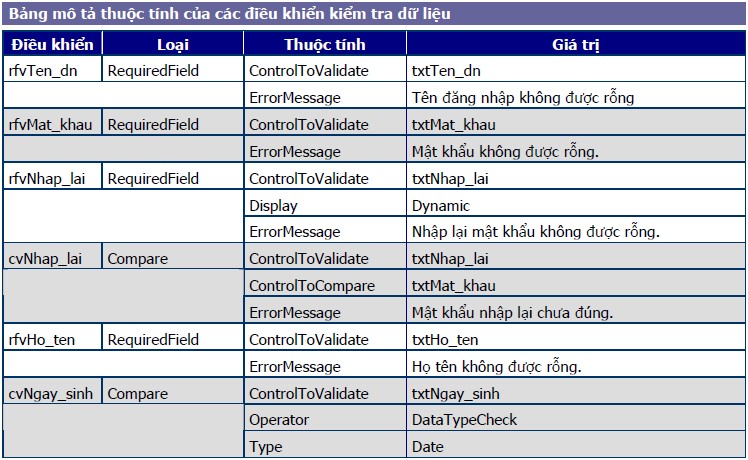
Giá trị mặc định của thuộc tính này là True - được phép hiển thị.



**Ví dụ:** Sử dụng các điều khiển Validation Controls

Trong ví dụ dưới đây, chúng ta thực hiện kiểm tra dữ liệu nhập trên các điều khiển có trong hồ sơ đăng ký khách hàng.





Thuộc tính Text của các điều khiển: **(\*)** Xử lý sự kiện:

protected

void

bt\_Dang\_ky\_Click(

object

sender,

EventArgs

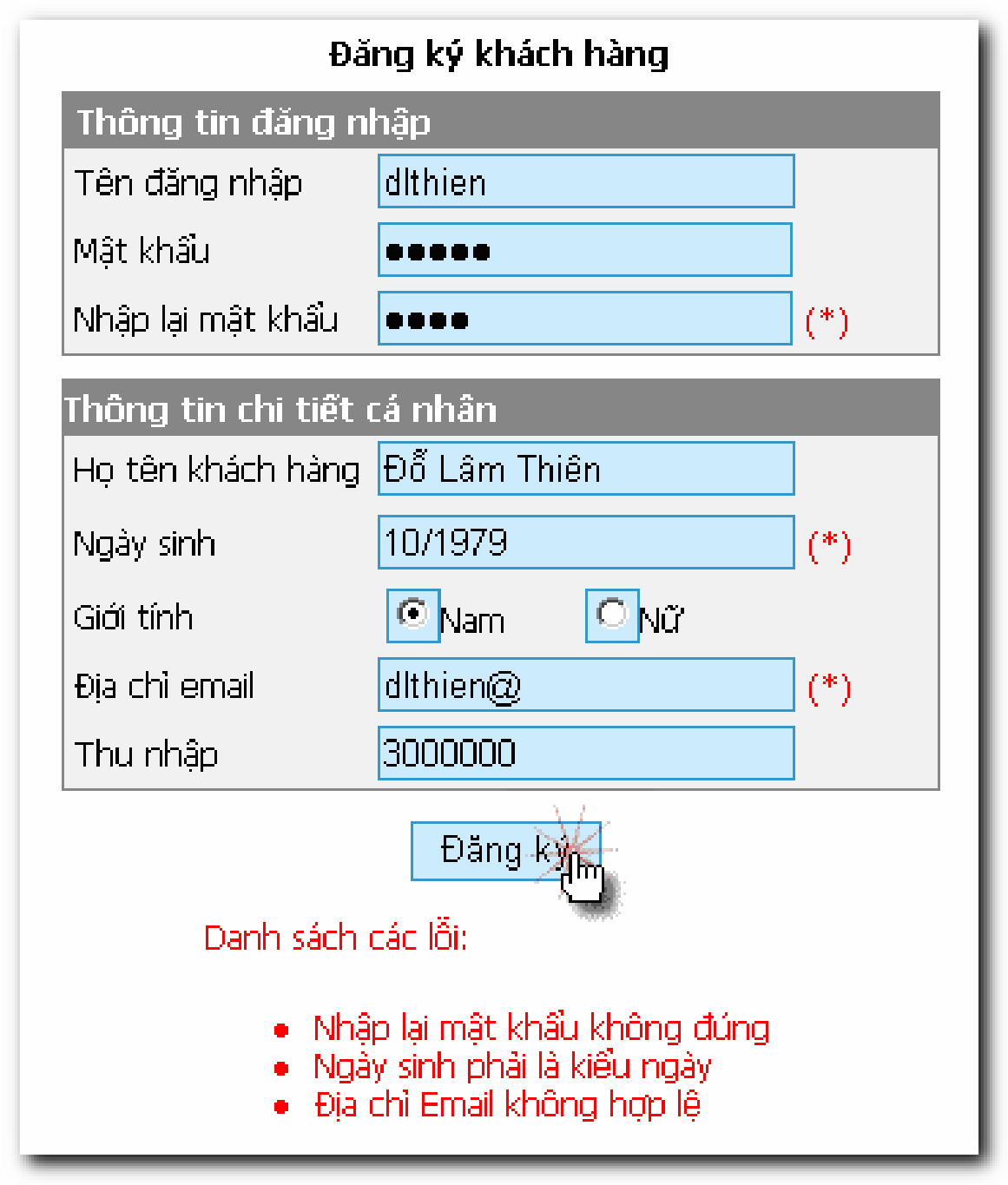
e)

{

lblThong\_bao.Text = “Đăng ký thành công”;

}

Các thông báo lỗi xuất hiện trên màn hình nhập liệu khi dữ liệu nhập không hợp lệ.



Các thông báo lỗi xuất hiện qua hộp thoại khi dữ liệu nhập không hợp lệ:



#### 3.7.4.User Controls

User controls là một trong các phương thức của Asp.Net nhằm gia tăng việc sử dụng lại mã nguồn. Một Web User Control trong Asp.Net cũng giống như một trang web Asp.Net hoàn chỉnh (.aspx file), bao gồm cả giao diện và mã nguồn. Tạo một user control cũng giống như tạo một trang Asp.Net.

User Controls khác so với một trang web Asp.Net ở những điểm sau:

* Đuôi file mở rộng của user control là .ascx, thay cho .aspx trong trang web

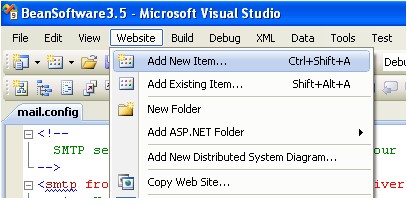
Asp.Net

* User control không thể thực thi như một file độc lập, bạn phải gắn user control vào một trang Asp.Net thì user control mới có ý nghĩa.
* Trong user control không có các thẻ <html>, <body> hay <form>.

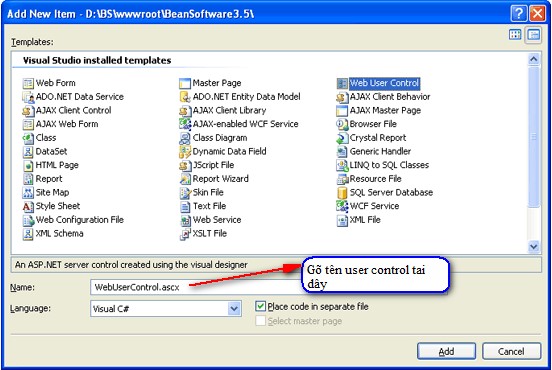
Web user control được sử dụng khi bạn muốn lặp lại mã nguồn nào đó ở nhiều trang khác nhau. Ngoại các đoạn mã HTML, web user control còn bao gồm cả web controls. Bạn có thể dễ dàng kéo thả các item ở trong toolbox vào user control. Cũng giống như các điều khiển khác, user control cũng có thuộc tính, phương thức và sự kiện để làm cho chúng trở nên hữu dụng.

### Tạo một User Control

Tất cả các user controls được hỗ trợ bởi lớp thư viện **System.Web.UI.UserControl**. Từ menu Visual Studio, chọn Website  Add New Item,… như hình bên dưới.



Hộp thoại mới xuất hiện. Để tạo ra một user control mới, chọn web user control template, như hình tiếp theo. Gõ tên cho control và Click Add.



Sau khi thiết kế xong user control (thiết kế giống như thiết kế trang web asp.net thường), ta kéo thả user control vừa tạo vào trang Asp.Net.

## CÂU HỎI

1. ASP.NET hỗ trợ các loại control nào? Khi nào ta nên dùng HTML control, khi nào nên dùng Server control?
2. Giả sử có một yêu cầu tạo một form đăng nhập vào trang web. Theo bạn phải dùng ít nhất những control nào để đáp ứng yêu cầu?
   * Thiết kế cơ bản các chức năng trên form
   * Xác nhận dữ liệu nhập
3. Thuộc tính AutoPostBack dùng để làm gì? Một đối tượng NonPostBack sẽ thực hiện sự kiện như thế nào khi không thiết lập AutoPostBack?

# Chương 4

**CÁC ĐIỀU KHIỂN LIÊN KẾT DỮ LIỆU**

## MỤC TIÊU

1. Về kiến thức

Cung cấp cho sinh viên những kiến thức:

* + Các điều khiển liên kết dữ liệu: GridView, DataList
  + Các phương pháp hiển thị và cập nhật dữ liệu sử dụng các điều khiển liên kết dữ liệu.

1. Về nội dung

Vận dụng các kiến thức đã học vào thực tế để làm việc:

* + Tạo các kết nối dữ liệu.
  + Truy cập và xử lí dữ liệu
  + Xây dựng ứng dụng ASP.Net sử dụng các điều khiển GridView, DataList.

## HÌNH THỨC VÀ PHƯƠNG PHÁP GIẢNG DẠY

* Trình chiếu power point
* Đặt vấn đề, giảng giải vấn đề
* Thực nghiệm kết hợp với máy tính
* Bài tập về nhà

## TÀI LIỆU THAM KHẢO

1. Phạm Hữu Khang (2007), C# 2005: tập 5: Lập trình ASP.NET 2.0:Quyển 3: Điều khiển kết nối cơ sở dữ liệu, NXB Lao động – Xã hội TPHCM.
2. <http://www.scribd.com/doc/53368069/Tai-Lieu-Giang-Day-ASP-net-Lelongt1k4>
3. <http://chienuit.wordpress.com/2011/06/11/gridview-trong-asp-net/>

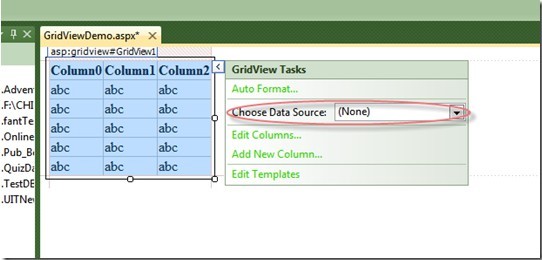
## 4.1. ĐIỀU KHIỂN GRIDVIEW

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

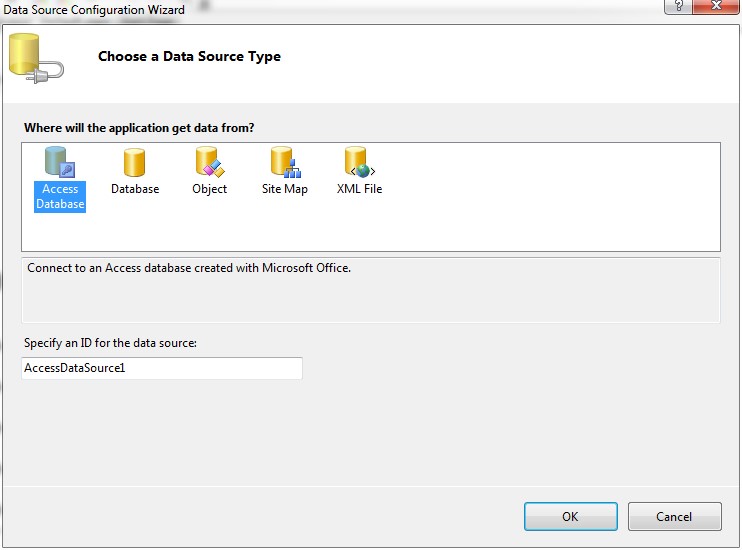
Trước hết bạn phải hiểu rõ khi nào thì sử dụng GridView. Đó là khi bạn muốn dữ liệu của mình (thuộc dạng danh sách hoặc tập hợp) hiển thị nhiều dòng trong một bảng với tiêu đề bên trên. Ví dụ, sử dụng GridView để hiển thị danh sách khách hàng.

### 4.1.1.Load dữ liệu lên GridView

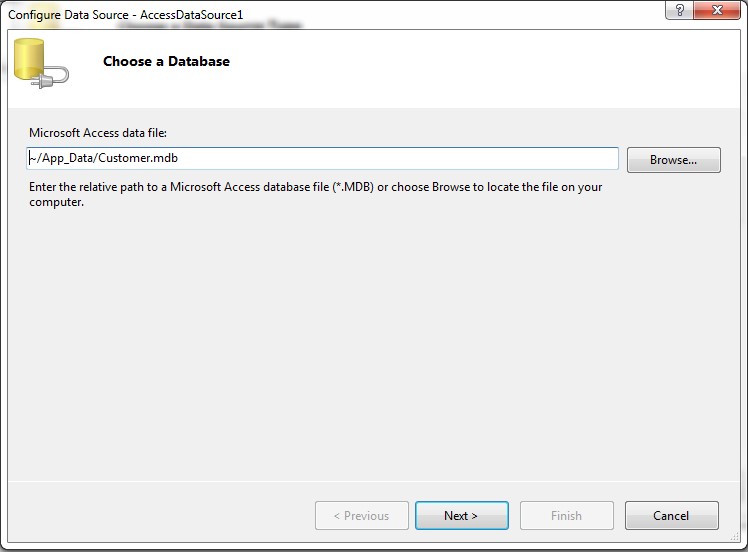
Kéo thả control Gridview trong toolbox vào giao diện của trang web



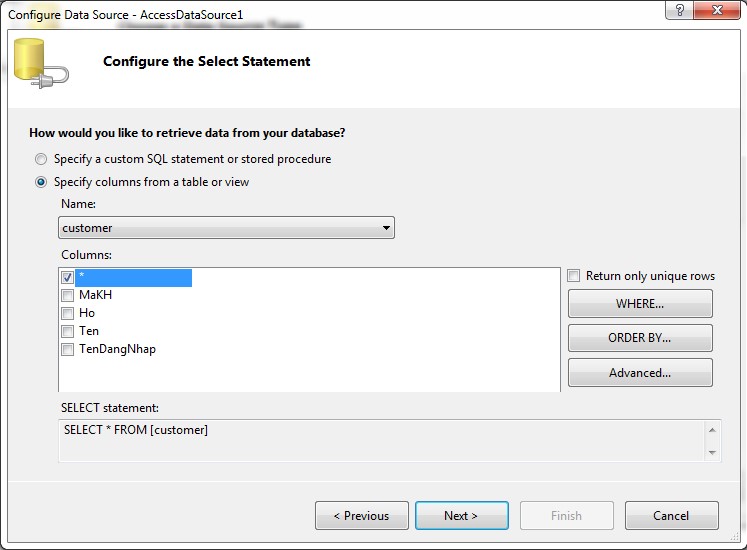
Nhấn vào mũi tên nhỏ góc trên bên phải của GridView các bạn sẽ có khung nhỏ cho phép chọn nguồn dữ liệu của GridView. Vì hiện tại chúng ta chưa tạo bất cứ các Control để kết nối dữ liệu (như ObjectDataSource, SqlDataSource..) cho nên trong danh sách “Choose Data Source” các bạn chọn “New Data Source”.

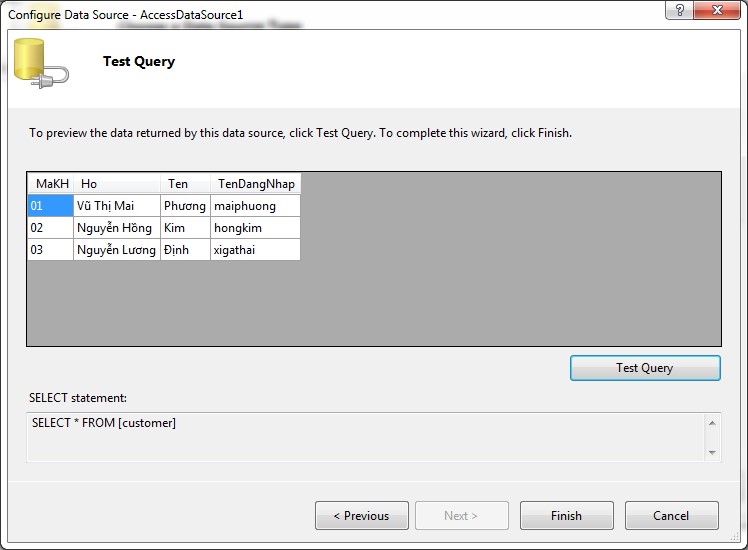


Ở đây, để đơn giản minh họa cho nhân vật chính là “GridView” nên tạm thời chúng ta không cần phải làm phức tạp thêm các nhân vật phụ, các bạn chọn AccessDatabase, sau đó nhấn OK. Hộp thoại tiếp theo yêu cầu bạn chỉ định kết nối đến cơ sở dữ liệu.



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





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



### 4.1.2.Các thao tác định dạng lưới

Để thực hiện các thao tác định dạng, chúng ta chọn chức năng Properties từ menu ngữ cảnh.

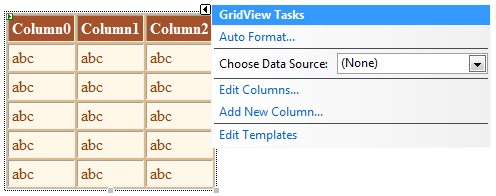
### Định dạng chung

ShowHeader: Qui định dòng tiêu đề trên có được phép hiển thị hay không. (mặc định là có hiển thị dòng tiêu đề)

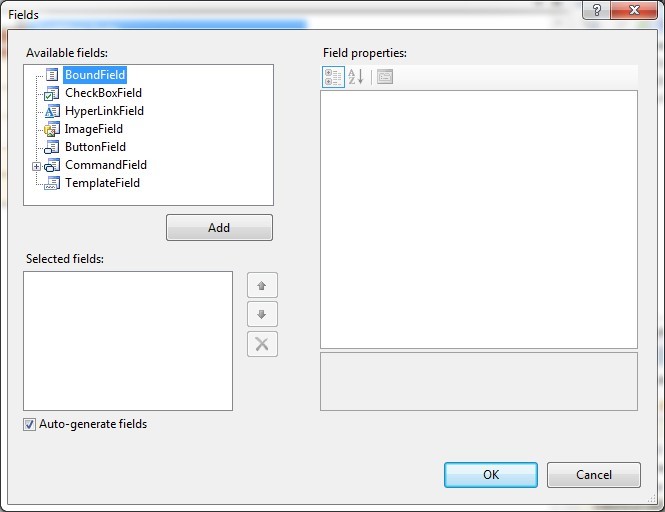
* ShowFooter: Qui định dòng tiêu đề dưới có được phép hiển thị hay không. (mặc định là không hiển thị dòng tiêu đề dưới)
* AllowSorting: Có cho phép sắp xếp dữ liệu hay không. (mặc định là không cho phép sắp xếp)

### Quản lý thông tin các cột

Muốn qui định các cột cần hiển thị, chọn EditColumns trong GridView Tasks



Cửa sổ hiển thị như sau:



* Auto – generate fields: Khi chọn chức năng này, GridView sẽ tự động phát sinh đầy đủ các cột có trong nguồn dữ liệu. Nếu chúng ta muốn qui định các cột cần hiển thị, chúng ta không chọn chức năng này.
* BoundField: cột có liên kết với nguồn dữ liệu

ButtonField: Cột dạng nút lệnh đã được thiết kế sẵn. Điều khiển GridView cung cấp cho chúng ta 3 loại cột dạng này:

− Select: Nút lệnh chọn dòng dữ liệu

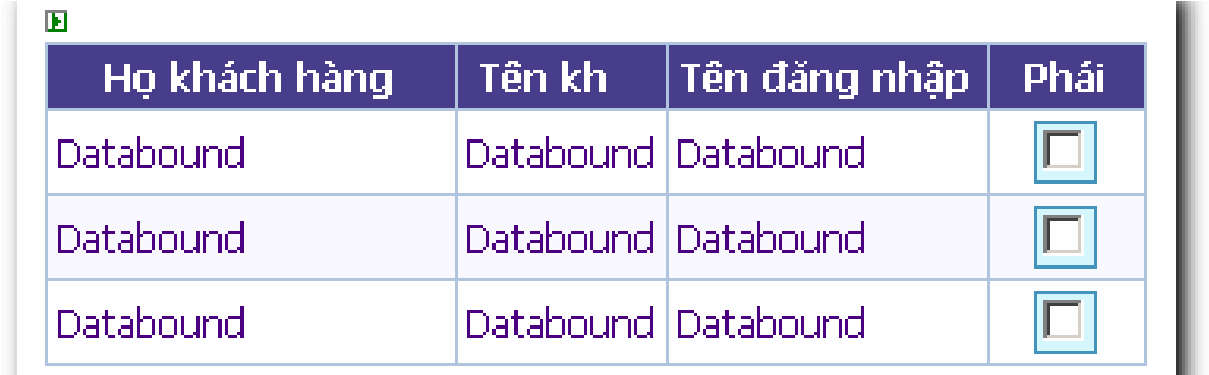
− Edit, Cancel, Update: Các nút lệnh hỗ trợ chức năng cập nhật dữ liệu trực tiếp trên lưới.

− Delete: Nút lệnh xóa dòng dữ liệu

Chúng ta sẽ có dịp tìm hiểu kỹ hơn về các nút lệnh này trong phần Cập nhật dữ liệu trực tiếp trên lưới.

* HyperLinkField: Cột có liên kết dữ liệu dạng liên kết
* TemplateField: Cột do người dùng tự thiết kế. Đây là loại cột có khả năng làm việc khá linh hoạt.

Ví dụ bạn cần hiển thị danh sách khách hàng. Tại cột Phái, bạn không muốn hiển thị Nam/Nữ, thay vào đó, bạn muốn hiển thị điều khiển checkbox thay thế, nếu checkbox được chọn - thể hiện phái Nam và ngược lại. Trong tình huống này, TemplateColumn là sự chọn lựa tốt dành cho bạn.



Qui định thông tin chi tiết cho các cột:

* HeaderText, FooterText: Thông tin tiêu đề trên/dưới của cột
* HeaderImageUrl: Hình hiển thị trên tiêu đề cột (thay thế thông tin tiêu đề cột - Header Text).
* SortExpression: Biểu thức sắp xếp của cột
* Visible: Qui định cột có được hiển thị hay không
* DataField: Qui định tên field hay tên thuộc tính của đối tượng dữ liệu cần hiển thị.
* DataFormatString: Biểu thức định dạng dữ liệu.

Mẫu định dạng: **{0:<chuỗi định dạng>}.** Ví dụ:

+ Định dạng số: {0:000.00}, {0:0.##}

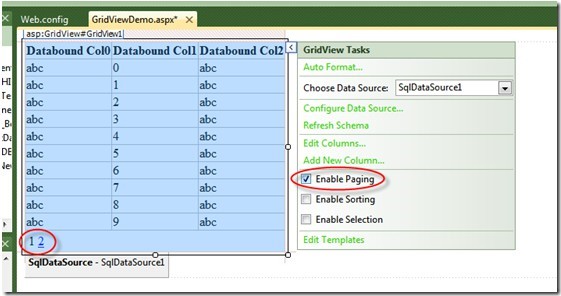
+ Định dạng ngày giờ: {0:dd/MM/yyyy}, {0:hh/mm/ss tt}

ReadOnly: Chọn giá trị này để cột chỉ được phép đọc, không cho phép cập nhật dữ liệu.

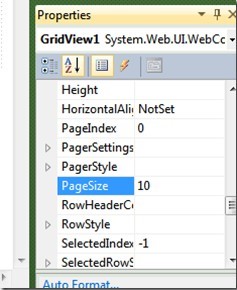
* Convert this Field into a TemplateField: Chuyển cột hiện hành thành cột dạng Template Field.

### Quản lý phân trang

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



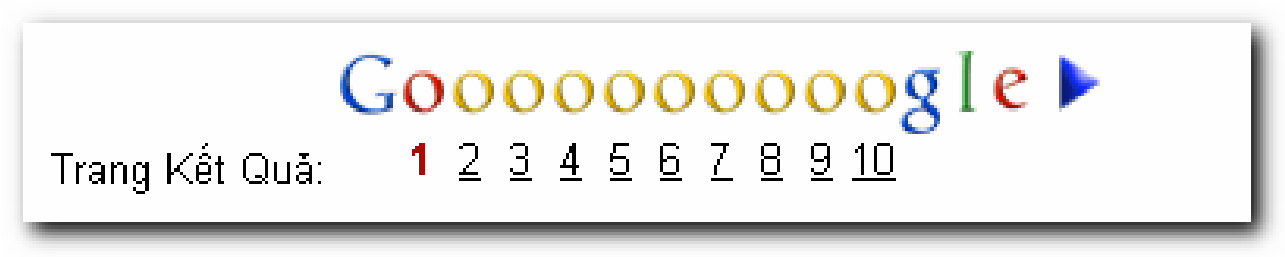
Nếu như bạn muốn số lượng dòng hiển thị ở mỗi trang nhiều hơn hoặc ít hơn thì bạn có thể chỉnh thuộc tính PageSize trong khung Properties của GridView



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



* AllowPagging: Có cho phép phân trang hay không.
* PageSize: Qui định số dòng của mỗi trang.
* Position: Qui định vị trí hiển thị của bộ nút di chuyển. Ở phía trên thanh tiêu đề, ở phía dưới hay cả hai.
* Mode: Qui định hình thức hiển thị của bộ nút di chuyển. Hiển thị dạng số trang hay là các chuỗi ký tự đại diện (Next page/Previous page button text). Trong trường hợp hiển thị dạng số, Numeric buttons qui định số nút lệnh được hiển thị tối đa.



**Google hiển thị kết quả được phân trang theo dạng số**

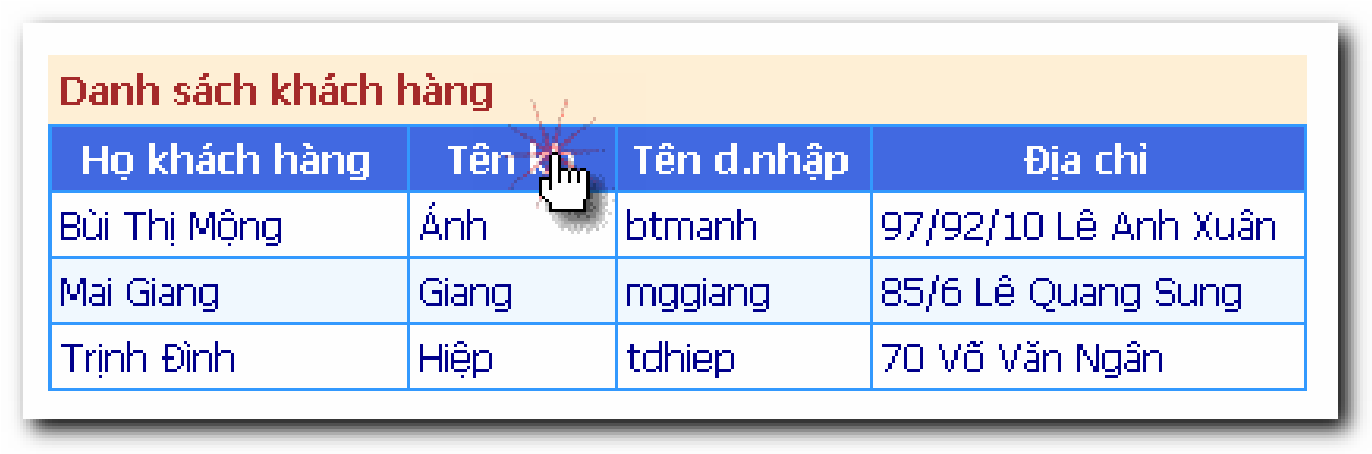
#### 4.1.3.Xử lí sắp xếp

Sắp xếp dữ liệu trên lưới là một công việc rất cần thiết đối với người sử dụng. Hãy thử tưởng tượng xem trong trường hợp chúng ta có khá nhiều dữ liệu hiển thị trên màn hình (giả sử là danh sách nhân viên chẳng hạn), thật khó để chọn ra các nhân viên có thâm niên làm việc lâu nhất hay các nhân viên có số giờ tham gia đề án nhiều nhất…. Với chức năng sắp xếp trên lưới sẽ giúp cho người dùng dễ dàng chọn ra các nhân viên thỏa mãn các yêu cầu trên.

Để thực hiện được thao tác sắp xếp dữ liệu trên lưới, chúng ta cần phải thực hiện các công việc sau:

Giá trị thuộc tính Allow sorting = True

Nhập giá trị cho thuộc tính Sort expression của các cột cần sắp xếp.



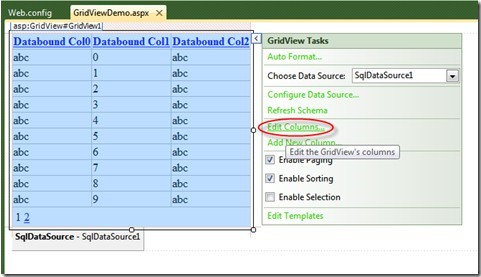
**Sắp xếp khách hàng tăng dần theo tên**

**4.1.4.Tùy biến các cột (**đọc thêm**)**

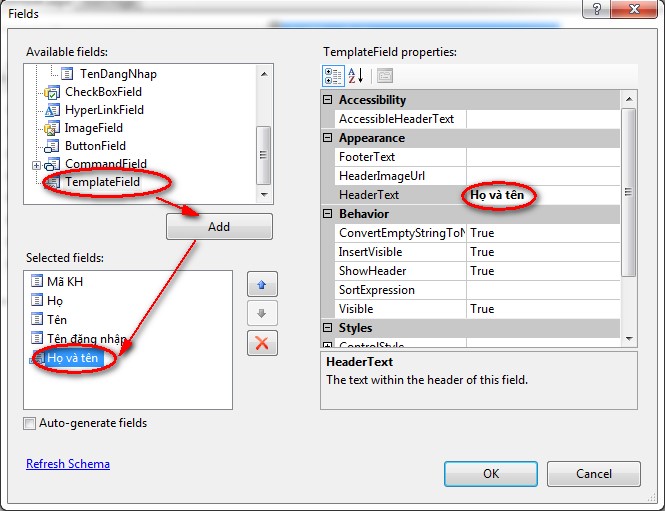
### Tạo cột tự định nghĩa

Khi dữ liệu được lấy từ database được đưa lên GridView thì mặc định GridView sẽ sử dụng chủ yếu là các control Label cho việc hiển thị nội dung (một số trường hợp khác thì GridView tự suy ra control phù hợp, ví dụ như các trường boolean trong CSDL sẽ được hiển thị thành các checkbox…). Vậy giả sử bạn muốn nội dung trong một cột nào đó không phải hiển thị theo cách thông thường mà phải hiển thị nội dung theo cách bạn định nghĩa cho nó thông qua mã HTML. Ví dụ, giả sử trong trường hợp này chúng ta muốn tạo ra thêm một cột là “Họ và tên”, cột này là kết hợp nội dung của hai cột Họ và Tên. Các bước thực hiện như sau:

Chọn chức năng Edit Columns trong hộp thoại nhỏ của GridView



Trong hộp thoại mới hiện ra, chọn TemplateField và nhấn Add để thêm một cột do chúng ta tự định nghĩa nội dung hiển thị, chọn Header của cột TemplateField này là “Họ và tên”, và sau đó nhấn OK.



Bây giờ là phần quan trọng nhất, bạn phải viết mã ASPX để hiển thị nội dung cho cột này. Các bạn chuyển sang chế độ chỉnh sửa mã ASPX. Trong chế độ này, các bạn sẽ thấy là GridView của chúng ta được định nghĩa qua thẻ <asp:GridView>, trong thẻ này sẽ có thẻ

<Columns> chứa định nghĩa cho tất cả các cột được hiển thị. Cột Họ và tên được định nghĩa trong thẻ <Columns>. Tuy nhiên, như các bạn thấy là thẻ này vẫn chưa có mã ASPX báo nó biết là hiển thị những gì và hiển thị như thế nào.

<asp:GridView ID="GridView1" runat="server" AllowPaging="True"

AutoGenerateColumns="False" DataKeyNames="MaKH" DataSourceID="AccessDataSource1" Height="170px" Style="z-index: 100; left: 15px; position: absolute; top: 18px" Width="458px"> <Columns>

<asp:BoundField DataField="MaKH" HeaderText="Mã KH" ReadOnly="True"

SortExpression="MaKH" />

<asp:BoundField DataField="Ho" HeaderText="Họ" SortExpression="Ho" />

<asp:BoundField DataField="Ten" HeaderText="Tên" SortExpression="Ten" />

<asp:BoundField DataField="TenDangNhap" HeaderText="Tên đăng nhập"

SortExpression="TenDangNhap" />

<asp:TemplateField HeaderText="Họ và tên"></asp:TemplateField>

</Columns>

</asp:GridView>

Để hiển thị Họ và Tên trong cột này thì chúng ta sẽ sử dụng cú pháp sau đây:

<asp:TemplateField HeaderText="Họ và tên">

<ItemTemplate>

<%# Eval("Ho") %>

<%# Eval("Ten") %>

</ItemTemplate>

</asp:TemplateField>

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

Chạy trang web vừa được chỉnh sửa, bạn sẽ có thêm được cột Họ tên với giá trị là sự kết hợp giữa cột Họ và cột Tên.



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

#### 4.1.5.Cập nhật dữ liệu trực tiếp trên lưới (tham khảo)

Cập nhật dữ liệu trực tiếp trên lưới trong ASP.Net được hỗ trợ khá tốt về giao diện. Công việc còn lại của chúng ta là thiết kế các nút lệnh như: Chọn, Sửa/Ghi - Không, Hủy, … và viết các lệnh cập nhật dữ liệu.

Trong cửa sổ thuộc tính của lưới, chúng ta tạo bộ nút lệnh cần thiết hỗ trợ cho việc cập nhật dữ liệu. Ở ví dụ này, chúng tôi tạo bộ nút (Select - chọn), (Edit, Update, Cancel - Sửa,

Ghi, Không) và (Delete - Hủy)

Đối với các nút lệnh trên, các bạn cần chú ý đến giá trị của thuộc tính CommandName. Ứng với mỗi nút lệnh có giá trị CommandName khác nhau, nhờ đó, ta viết lệnh xử lý với chức năng tương ứng được chọn.

− Select: CommandName = "Select"

− Edit, Update/Cancel: CommandName = "Edit", "Update"/"Cancel"

− Delete: CommandName = "Delete"

### Tìm hiểu về thuộc tính CommandName

Cũng cần bàn thêm một chút ở đây về thuộc tính Command Name. Như các bạn cũng biết, các nút lệnh ở trên (Chọn, Thêm - Sửa / Ghi / Không - Hủy) là do VS.Net hỗ trợ, giá trị thuộc tính CommandName của các nút lệnh trên là những giá trị mặc định được qui định sẵn.

Ứng với mỗi CommandName mặc định, sẽ có các sự kiện để ta thực hiện các xử lý tương ứng:

− Command name="Edit": Sự kiện **EditCommand**

− Command name="Update": Sự kiện **UpdateCommand**

− Command name="Cancel": Sự kiện **CancelCommand**

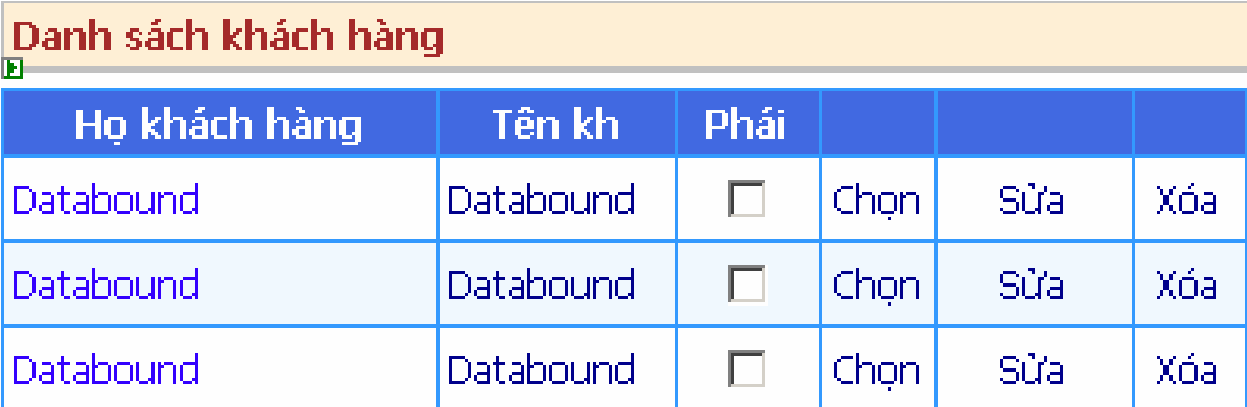
− Command name="Delete": Sự kiện **DeleteCommand**

Chắc hẳn các bạn sẽ thắc mắc tại sao không có sự kiện SelectCommand? Bốn sự kiện được liệt kê trên là 4 sự kiện dành riêng, tương ứng với giá trị của các Command name mặc định là Edit, Update, Cancel, Delete. Đối với những CommandName có giá trị khác, chúng ta sẽ sử dụng sự kiện dành chung cho tất cả các nút lệnh có thuộc tính CommandName (Button, Linkbutton, ImageButton) được đặt trên lưới - sự kiện **ItemCommand**.

Tại sao vậy? Vì khi ta đặt các nút lệnh vào lưới (sử dụng cột Template column), chúng (các nút lệnh) không còn sự kiện **Click** nữa, thay vào đó, tất cả các nút lệnh khi được nhấn sẽ gây ra sự kiện **ItemCommand.** Dựa vào giá trị **e.CommandName** (tham số trong sự kiện) để chúng ta xác định nút lệnh nào đã được nhấn.

Cũng cần lưu ý thêm ở đây là bất kỳ nút lệnh nào khi được nhấn đều gây ra sự kiện

ItemCommand. Do đó, đối với các nút lệnh có giá trị thuộc tính CommandName là Edit, Update, Cancel, Delete khi được nhấn vẫn gây ra sự kiện ItemCommand trước khi gây ra các sự kiện dành riêng cho chúng.



**Giao diện lưới sau khi thêm bộ nút lệnh**

## 4.2. ĐIỀU KHIỂN DATALIST

Như điều khiển DataGrid, điều khiển DataList được sử dụng để hiển thị dữ liệu. Tuy nhiên đối với DataList, chúng ta phải tự thiết kế hình thức hiển thị dữ liệu (giống như Templete Column của DataGrid).



### 4.2.1.Sử dụng DataList hiển thị dữ liệu

Một số thuộc tính cần lưu ý của DataList:

* RepeatDirection: qui định hướng hiển thị dữ liệu o Horizontal: hiển thị dữ liệu theo chiều ngang.



o Vertical (mặc định): hiển thị dữ liệu theo chiều đứng



* RepeatColumn: qui định số cột hiển thị của DataList

Thiết kế hình thức hiển thị cho DataList cũng tương tự như thiết kế cho cột Template Column của GridView.

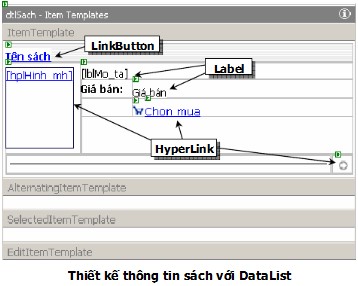
Chọn Edit Template | Item Template từ menu ngữ cảnh để thực hiện thiết kế hình thức hiển thị cho DataList.



*Trong quá trình thực hành, để công việc thiết kế được dễ dàng, các bạn thực hiện thiết kế ở bên ngoài điều khiển DataList. Sử dụng Table để định vị trí hiển thị của các điều khiển.*

*Sau khi hoàn tất công việc thiết kế, chúng ta kéo kết quả đã thiết kế vào vị trí cần hiển thị trong DataList.*

Ví dụ: Hiển thị thông tin sách với DataList

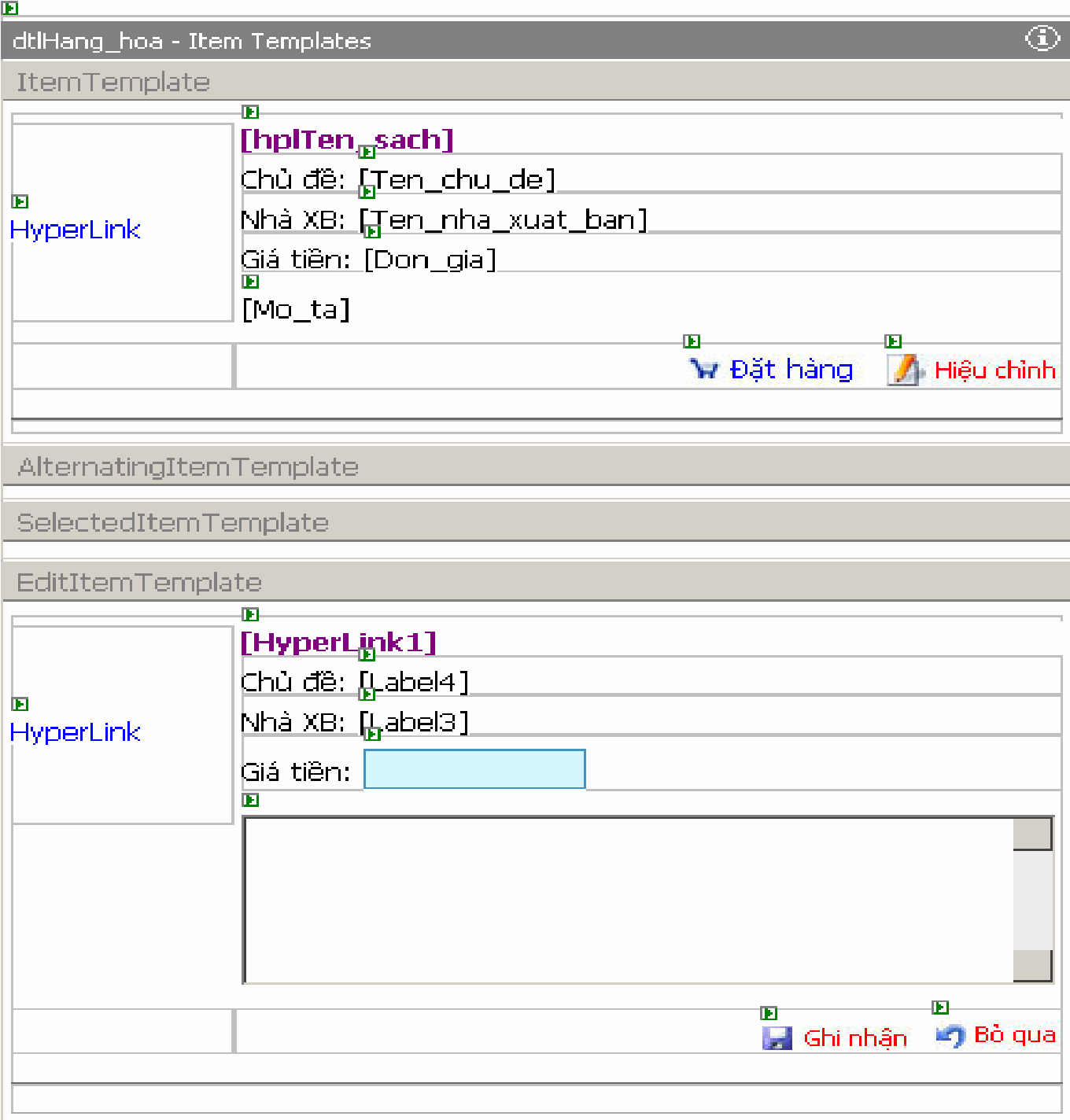


Xử lí hiển thị dữ liệu cho DataList được viết trong sự kiện **ItemDataBound**. Xử lí nhấn các Button đặt trong DataList được viết trong sự kiện **ItemCommand**.

### 4.2.2.Cập nhật dữ liệu với DataList (*tham khảo*)

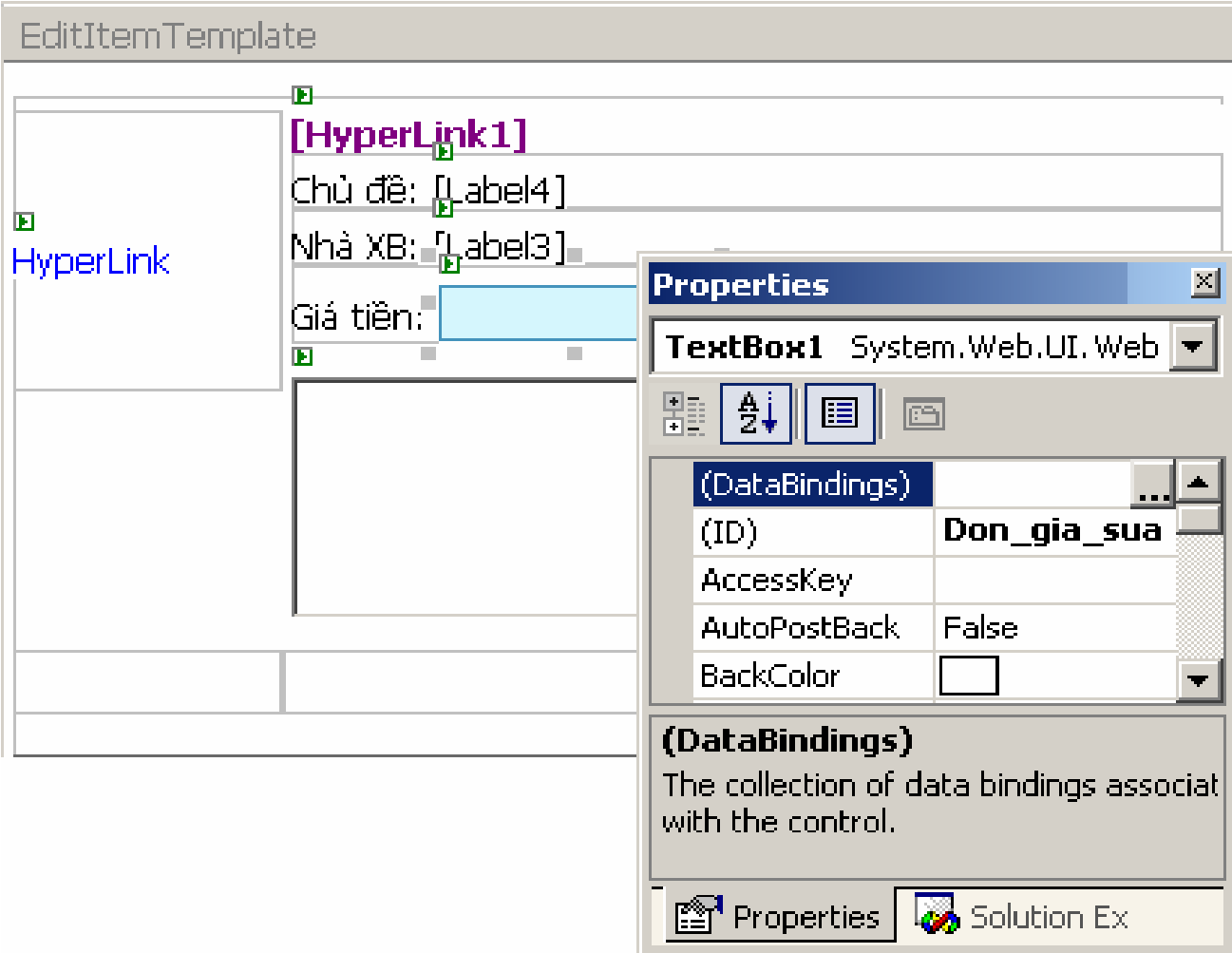
Ngoài việc hiển thị dữ liệu, DataList cũng hỗ trợ các thao tác cập nhật dữ liệu. Để thực hiện chức năng cập nhật dữ liệu với DataList, chúng ta cần phải thiết kế thêm vùng

EditItemTemplate cho DataList. (xem hình)

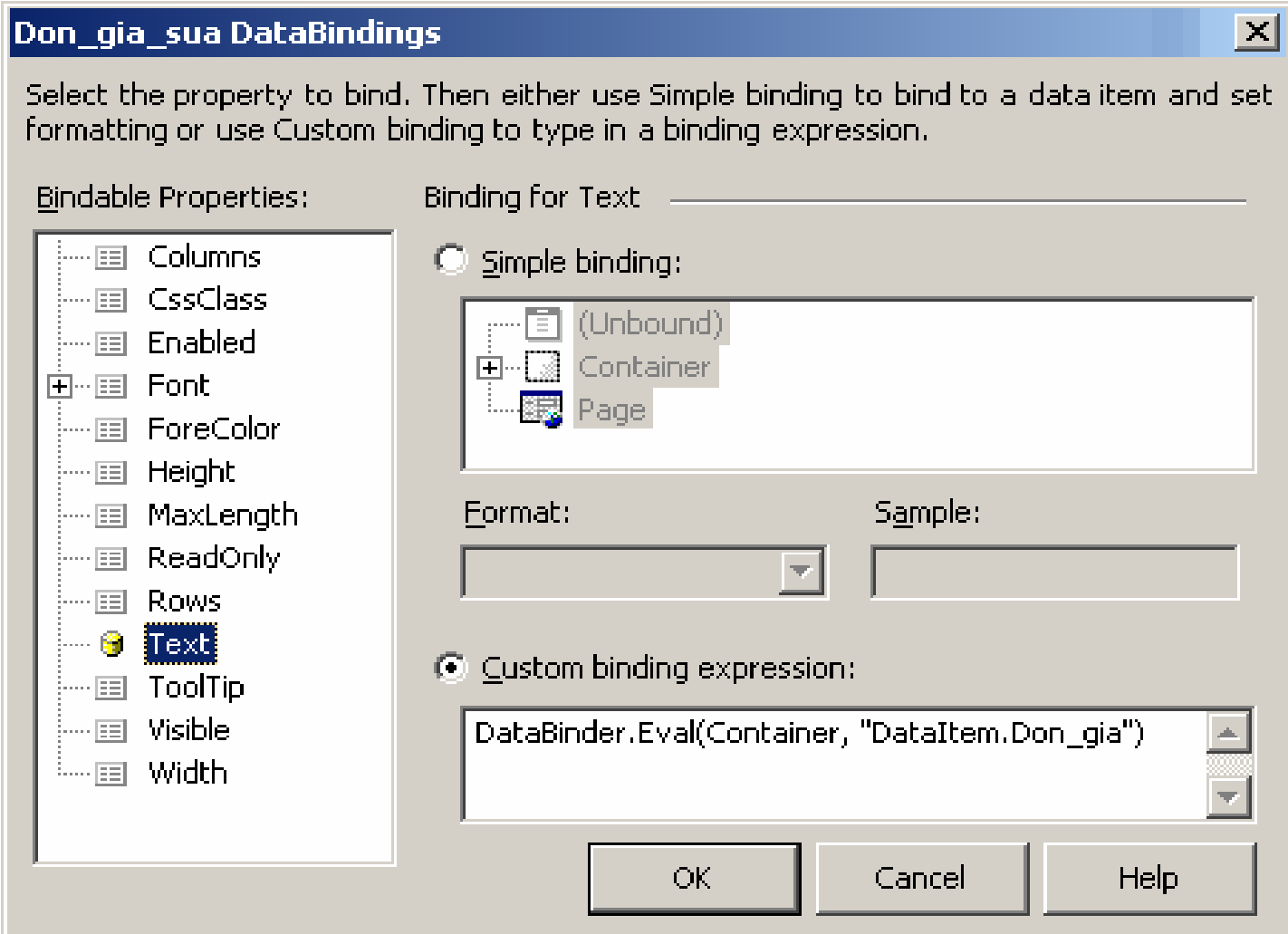


Thiết kế cả 2 vùng ItemTemplate và EditItemTemplate. Thực hiện các thao tác liên kết dữ liệu cho các điều khiển trong vùng EditItemTemplate thông qua cửa sổ thuộc tính tương tự như trong ItemTemplate.

Chú ý: Chúng ta hoàn toàn có thể thực hiện việc liên kết dữ liệu trong sự kiện ItemDataBound.



**Chọn chức năng DataBiding cho ô đơn giá**



**Liên kết dữ liệu với cột đơn giá**

Xử lý các sự kiện EditCommand, CancelCommand, UpdateCommand để thực hiện/bỏ qua việc thay đổi dữ liệu.

## CÂU HỎI

1. ASP.Net có thể kết nối với các loại cơ sở dữ liệu nào?
2. Nêu những điểm mạnh và những điểm chưa tốt của việc kết nối và xử lí dữ liệu qua giao diện?
3. Trong tất cả các đối tượng liên kết dữ liệu của ASP.Net, đối tượng nào có thể phân trang, sắp xếp dữ liệu?

# Chương 5

**QUẢN LÝ TRẠNG THÁI**

## MỤC TIÊU

1. Về kiến thức

Cung cấp cho sinh viên những kiến thức:

* + Các đối tượng quản lý trạng thái của Asp.Net
  + Nguyên tắc sử dụng các đối tượng này
  + Một số phương thức của đối tượng

1. Về nội dung

Vận dụng các kiến thức đã học vào thực tế để làm việc với ứng dụng web:

* + Khai thác thông tin của Client – Máy duyệt web.
  + Khai thác thông tin của Server
  + Sử dụng và quản lý dữ liệu dùng chung

## HÌNH THỨC VÀ PHƯƠNG PHÁP GIẢNG DẠY

* Trình chiếu power point
* Đặt vấn đề, giảng giải vấn đề
* Thực nghiệm kết hợp với máy tính
* Bài tập về nhà

## TÀI LIỆU THAM KHẢO

1. .NET Toàn Tập - Lập Trình Web Dùng ASP.NET Và C# - Tập 4 – Dương Quang

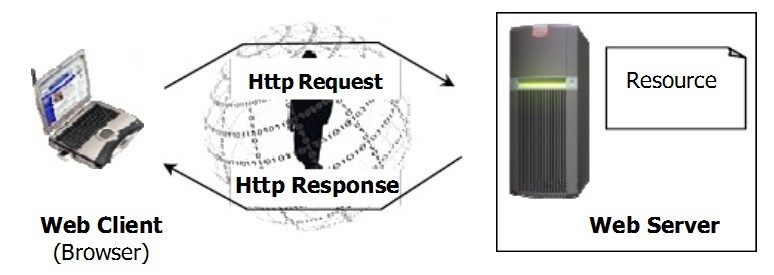
Thiện – 2005 – chương 1, 4

1. .NET Toàn Tập - Lập Trình Web Dùng ASP.NET Và C# - Tập 5 – Dương Quang Thiện – 2005 – chương 3, 4
2. [http://codeproject.com](http://codeproject.com/)
3. [http://dotnetspider.com](http://dotnetspider.com/)

Sự khác nhau quan trọng nhất giữa lập trình WinForm và WebForm là quản lý trạng thái, làm sao để lưu giữ thông tin qua chu kỳ sống của ứng dụng. Thông tin có thể là tên người dùng, nếu website thương mại điện tử là giỏ hàng… Lập trình WinForm ít quan tâm đến quản lý trạng thái, trong khi đó lập trình Web cần phải chú trọng vì ứng dụng có rất nhiều người dùng cùng lúc, sử dụng trên cùng một máy chủ chạy ứng dụng. Do đó, trạng thái đóng vai trò quan trọng để tạo nên hiệu quả ứng dụng.

Quá trình lưu giữ thông tin đòi hỏi phải an toàn, kiểm soát các trạng thái làm việc, lưu trữ… để ứng dụng được bảo mật hơn. Chương này giúp chúng ta tìm hiểu trạng thái và quản lý trạng thái trong ứng dụng, khi nói đến trạng thái các thuật ngữ như Session, Cookies thường xuất hiện

## 5.1.ĐỐI TƯỢNG REQUEST, RESPONSE



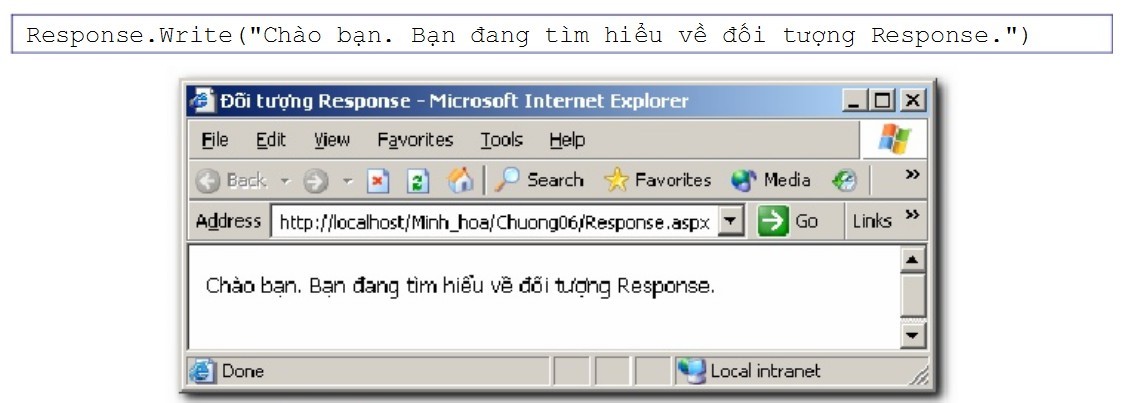
Hình 5.1. Quá trình Request – Response của HTTP

### 5.1.1.Đối tượng Response

Đối tượng Response được sử dụng để giao tiếp với Client, nó quản lý và điều phối thông tin từ Web Server đến các trình duyệt của người dùng.

#### *1.1.1.1 .Phương thức Write*

Phương thức Write của đối tượng Response được dùng để in ra một chuỗi trên trang Web. Phương thức này là một trong những phương thức chủ lực trong các ứng dụng web sử dụng ASP 3.0 khi cần gửi kết quả từ Server về cho Client.



Trong ASP.Net chúng ta có thể thực hiện như sau:

lblChao.Text = “Chào bạn. Bạn đang tìm hiểu về đối tượng Response.”

Qua ví dụ trên, chắc có lẽ bạn cũng nhận ra rằng, khi sử dụng phương thức Response.Write, chúng ta không thể qui định vị trí hiển thị của chuỗi trên trong trang Web.

Thay vào đó, với ASP.Net, thông qua các Server Control, chúng ta có thể thực hiện chức năng tương tự nhưng linh hoạt hơn bằng cách đặt điều khiển tại vị trí cần hiển thị.

#### *1.1.1.2 .Phương thức Redirect*

Phương thức Redirect gởi thông điệp yêu cầu Web Browser truy cập đến một địa chỉ khác. Ví dụ:

‘

Nếu đăng nhập thành công

If <Kiểm tra đăng nhập> Then

Response.Redirect (“Chao.aspx”);

Else

Response.Redirect (“Dangnhap.aspx”);

End If

### 5.1.2.Đối tượng Request

Đối tượng Request được sử dụng để nhận thông tin từ trình duyệt của người dùng gởi về cho Web Server.

### Thuộc tính QueryString

HTTP được xác định qua URLs (Uniform Resource Locators), với cấu trúc chuỗi có định dạng như sau:

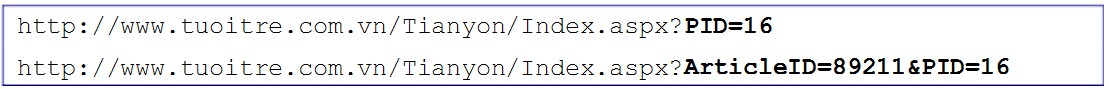
http:// <host> [:post][path]**[? <QueryString>]**]

Phần cuối của chuỗi URL là QueryString – còn được gọi là chuỗi tham số, có cấu trúc như sau:

[?Tham\_so\_1 = Gia\_tri\_1[&Tham\_so\_2 = Gia\_tri\_2[&….]]]

Trong trường hợp có nhiều tham số, các cặp [<Tham\_so> = <Gia\_tri>] phân cách nhau bằng dấu &.

**Ví dụ** :



Thuộc tính QueryString của đối tượng Request cho phép chúng ta nhận các giá trị truyền qua chuỗi tham số này.



**Ví dụ**: Một người dùng gởi thông điệp đến Web Server yêu cầu trang:

“Request.aspx?Chuc\_nang=Hieu\_chinh&ID=123”. Để lấy giá trị hai tham số trong chuỗi QueryString, chúng ta thực hiện như sau:

String sChuc\_nang;

Int Id;

sChuc\_nang = Request.QueryString(“Chuc\_nang”);

Id = Request.QueryString(“ID”);

Trong trường hợp tên tham số không tồn tại trong chuỗi QueryString, thuộc tính Request.QueryString() sẽ trả về giá trị nothing.

## 5.2.ĐỐI TƯỢNG VIEWSTATE

Đối tượng ViewState được cung cấp để lưu lại những thông tin của trang web sau khi web server gởi kết quả về cho Client. Mặc định, các trang web khi được tạo sẽ cho phép sử dụng đối tượng ViewState thông qua thuộc tính EnableViewState (của trang web) = True.

Gán giá trị cho ViewState:

ViewState["Tên trạng thái"] = <giá trị>

Ví dụ, gán giá trị cho Counter có giá trị là 7 cho ViewState:

**This.ViewState[“Counter”] = 7;**

Nhận giá trị từ đối tượng ViewState:

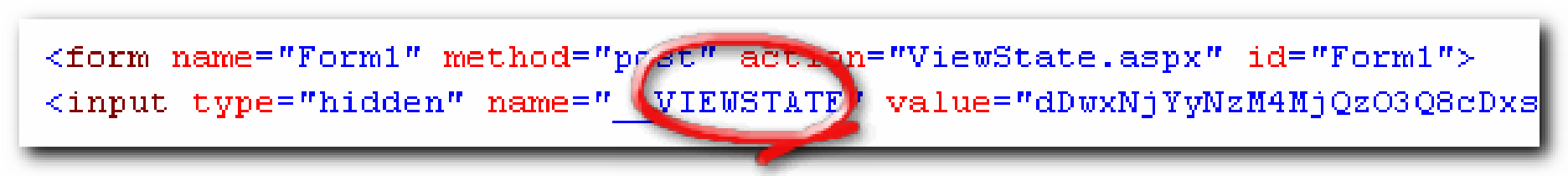
<biến> = ViewState["Tên trạng thái"]

**int counter;**

**counter = (int)this.ViewState[“Counter”];**

Tùy thuộc vào cách lưu trữ đối tượng mà ép kiểu dữ liệu cho phù hợp.

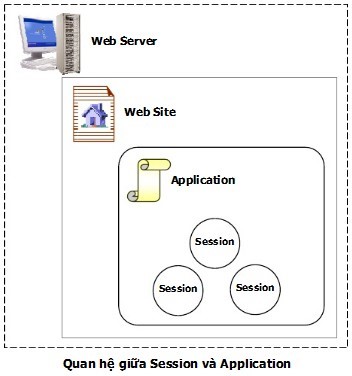
Về bản chất, các giá trị trong đối tượng ViewState được lưu trong một điều khiển hidden và các giá trị này đã được mã hóa. Đối tượng ViewState giúp chúng ta giảm bớt công sức trong việc lưu trữ và truy xuất các thông tin mà không phải sử dụng nhiều điều khiển hidden.



**Chọn chức năng View | Source từ browser**

## 5.3.ĐỐI TƯỢNG SESSION, APPLICATION

Application và Session là hai đối tượng khá quan trọng trong ứng dụng web, giúp các trang aspx có thể liên kết và trao đổi dữ liệu cho nhau.



### 5.3.1.Đối tượng Application

Đối tượng Application được sử dụng để quản lý tất cả các thông tin của một ứng dụng web. Thông tin được lưu trữ trong đối tượng Application có thể được xử lí trong bất kì trang aspx nào trong suốt chu kỳ sống của ứng dụng.

### Sử dụng biến Application

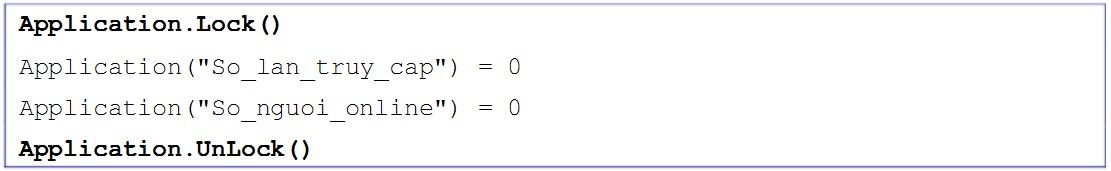
Tạo biến Application



Lấy giá trị từ biến Application



**Ví dụ**:



**Chú ý**:

Do tại một thời điểm có thể có cùng lúc nhiều người cùng truy cập và thay đổi giá trị của các thông tin được lưu trữ trong đối tượng Application, chúng ta nên sử dụng bộ lệnh Lock và Unlock ngay trước và sau khi cập nhật giá trị của biến Application.

Biến Application có thể được sử dụng ở bất kì trang nào và được duy trì trong suốt chu kỳ sống của ứng dụng.

#### 5.3.2.Đối tượng Session

Đối tượng Session được dùng để lưu trữ thông tin người dùng trong ứng dụng. Thông tin được lưu trữ trong Session là của một người dùng trong một phiên làm việc cụ thể. Web Server sẽ tự động tạo một đối tượng Session cho mỗi người dùng mới kết nối vào ứng dụng và tự động hủy chúng nếu người dùng không còn làm việc với ứng dụng nữa.

Tuy nhiên, không giống như đối tượng Application, đối tượng Session không thể chia sẻ thông tin giữa những lần làm việc của người dùng, nó chỉ có thể cung cấp, trao đổi thông tin cho các trang trong lần làm việc tương ứng.

Trong ứng dụng web, đối tượng Session giữ vai trò khá quan trọng. Do sử dụng giao thức HTTP, một giao thức phi trạng thái, Web Server hoàn toàn không ghi nhớ những gì giữa những lần yêu cầu của Client. Đối tượng Session tỏ ra khá hữu hiệu trong việc thực hiện “lưu vết và quản lý thông tin của người dùng”. **Thuộc tính và phương thức *a) Thuộc tính Timeout***

Qui định khoảng thời gian (tính bằng phút) mà Web Server duy trì đối tượng Session nếu người dùng không gửi yêu cầu nào về lại Server. Giá trị mặc định của thuộc tính này là 20.

Nếu không có yêu cầu nào kể từ lần yêu cầu sau cùng một khoảng thời gian là <Timeout> phút, đối tượng Session mà Web Server cấp cho lần làm việc đó sẽ tự động được giải phóng. Những yêu cầu sau đó được Web Server coi như một người dùng mới, và đương nhiên sẽ được cấp một đối tượng Session mới. ***b) Phương thức Abandon***

Như các bạn đã biết, trong khoảng thời gian <Timeout> phút kể từ lần yêu cầu sau cùng của Client, đối tượng Session vẫn được duy trì dù không có sự tương tác nào của Client. Điều này đồng nghĩa với việc Web Server phải sử dụng một vùng nhớ để duy trì đối tượng Session trong một khoảng thời gian tương ứng.

Phương thức Abandon của đối tượng Session sẽ giải phóng vùng nhớ được dùng để duy trì đối tượng Session trên Web Server ngay khi được gọi thực hiện. Những yêu cầu sau đó được Web Server coi như một người dùng mới.

## 5.4.ĐỐI TƯỢNG SERVER

Đối tượng Server được sử dụng để cung cấp thông tin của Server cho ứng dụng.

* Thuộc tính MachineName: thuộc tính này được dùng để lấy tên của Web Server
* Phương thức Mappath: được dùng để lấy đường dẫn vật lý hoặc đường dẫn ảo đến một thư mục trên Server.
* Phương thức Transfer(<đường dẫn tới trang cần yêu cầu>): ngừng thi hành trang hiện hành, gởi yêu cầu mới đến trang được gọi thực hiện.

## 5.5.ĐỐI TƯỢNG COOKIES

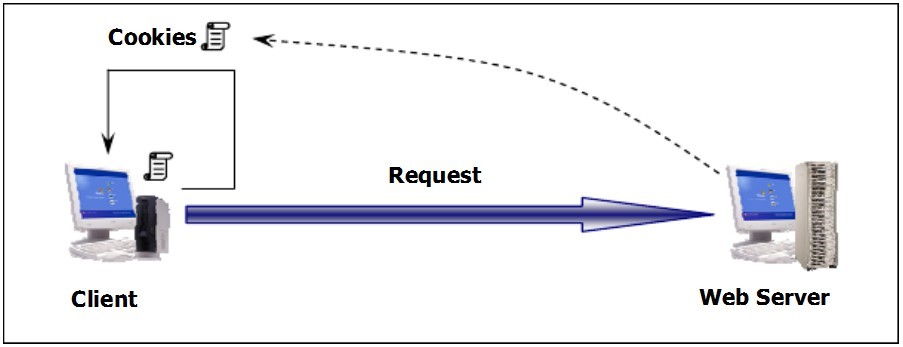
### 5.5.1.Giới thiệu

Có lẽ bạn cũng đã từng đăng ký làm thành viên của một trang web hay một forum nào đó, và chắc cũng không ít lần ngạc nhiên khi bạn vừa yêu cầu đến một trang web hay forum mà bạn đã đăng ký trước đó, trang web nhận ngay ra bạn chính là thành viên của họ và gởi ngay lời chào đến bạn.

Làm sao mà Web Server nhận ra được? Không đâu xa cả, những thông tin đó được lưu ngay tại chính máy tính của bạn. Những thông tin được Web Server lưu tại máy Client được gọi là Cookies.

Không giống như đối tượng Session, đối tượng Cookies cũng được dùng để lưu trữ thông tin của người dùng, tuy nhiên thông tin này được lưu ngay tại máy gởi yêu cầu đến Web Server.

Có thể xem một Cookies như một tập tin (với kích thước khá nhỏ) được Web Server lưu tại máy của người dùng. Mỗi lần có yêu cầu đến Web Server, những thông tin của Cookies cũng được gửi theo về Web Server.



### 5.5.2.Làm việc với Cookies

Để sử dụng Cookies, bạn cần khai báo không gian tên: **using System.Net;** Để thiết lập Cookies, bạn cần khởi tạo đối tượng HttpCookie.

*//Tạo đối tượng cookie*

HttpCookie cookie = new HttpCookie(“cookies”);

*//Thiết lập giá trị*

cookie.Value = “ASP .NET - Security”; **Thêm Cookies**



*//thiết lập thời gian sống của cookies 1 năm*

Cookie.Expires = Datetime.Now.AddYears(1);

**Ví dụ**:

HttpCookie

cookTen\_dn =

new

HttpCookie

(

"Ten\_dang\_nhap"

)

;

cookTen\_dn.Value = txtTen\_dang\_nhap.Text;

cookTen\_dn.Expires = Date.Today.AddDays(1);

Response.Cookies.Add(cookTen\_dn);

Trong ví dụ trên, chúng ta đã tạo ra Cookies có tên là Ten\_dang\_nhap lưu trữ tên đăng nhập của người dùng. Thông tin này sẽ được lưu trữ trên Cookies một ngày kể từ ngày hiện hành trên Web Server.

### Lấy giá trị từ Cookies

HttpCookie <biến Cookie>;

<biến Cookie> = Request.Cookies(<Tên Cookies>)

<biến Cookie>.Value ' Giá trị của Cookies

Trước khi lấy giá trị của Cookies, bạn cần kiểm tra:

*//Kiểm tra tồn tại của cookie*

if (cookie != null) {

}

Để xóa giá trị của Cookies, bạn sử dụng đoạn mã sau:

HttpCookie cookie = new HttpCookie(“author”);

Cookie.Expires = Datetime.Now.AddDays(-1);

Response.Cookies.Add(cookie);

Trong trường hợp Cookies chưa được lưu hoặc đã hết thời hạn duy trì tại Client, giá trị nhận được là Nothing.

## CÂU HỎI

1. Đối tượng nào dùng để truy vấn thông tin dữ liệu từ máy duyệt web? Cho biết những thông tin nào sẽ lấy được. Ví dụ một số thông tin không thể đọc.
2. Giả sử có một bài toán yêu cầu đếm số lượng người truy cập vào trang web. Bạn sẽ dùng đối tượng nào? Viết phương thức đếm?
3. Cookie là gì? Chúng ta thường sử dụng Cookie vào những mục đích cụ thể nào? Thời gian sống của Cookie có thể được thay đổi bởi người dùng hay không?
4. Session là gì? Hãy nên những trường hợp bạn không nên dùng Session để lưu trữ dữ liệu?
5. Làm thế nào để thiết lập thời gian sống của Session? Trường hợp cụ thể nào nên thiết lập thời gian sống của Session?

# Chương 6

**BẢO MẬT ỨNG DỤNG WEB**

## MỤC TIÊU

1. Về kiến thức

Cung cấp cho sinh viên những kiến thức:

* + Bảo mật một ứng dụng web
  + Các hình thức bảo mật

1. Về nội dung

Vận dụng các kiến thức đã học vào thực tế để làm việc với ứng dụng web:

* + Chọn một hình thức bảo mật phù hợp với ứng dụng web.
  + Cấu hình ứng dụng web để có tính bảo mật cao
  + Khai thác thành công những đặc điểm tiên tiến của Asp.Net hỗ trợ phát triển ứng dụng web bảo mật.

## HÌNH THỨC VÀ PHƯƠNG PHÁP GIẢNG DẠY

* Trình chiếu power point
* Đặt vấn đề, trao đổi

## TÀI LIỆU THAM KHẢO

1. Giáo trình C#, Trần Nam Tiến, 2007
2. [http://codeproject.com](http://codeproject.com/)
3. [http://dotnetspider.com](http://dotnetspider.com/)

## 6.1.GIỚI THIỆU SƠ LƯỢC CÁC KỸ THUẬT TẤN CÔNG ỨNG DỤNG WEB

**6.1.1.Kiểm soát truy cập Web**

### Thâm nhập hệ thống qua cửa sau (Back-door)

Trong quá trình thiết kế ứng dụng, những người phát triển ứng dụng có thể cài một “cửa sau” (back door) để sau này có thể thâm nhập vào hệ thống một cách dễ dàng.

**6.1.2.Chiếm hữu phiên làm việc**

### Ấn định phiên làm việc

Là kỹ thuật tấn công cho phép Hacker mạo danh người dùng hợp lệ bằng cách gửi một

SessionID hợp lệ đến người dùng, sau khi người dùng đăng nhập vào hệ thống thành công, Hacker sẽ dùng lại SessionID đó và nghiễm nhiên trở thành người dùng hợp lệ

### Đánh cắp phiên làm việc

Là kỹ thuật tấn công cho phép Hacker mạo danh người dùng hợp lệ sau khi nạn nhân đã đăng nhập vào hệ thống bằng cách giải mã SessionID của họ được lưu trữ trong Cookie hay tham số URL, biến ẩn của Form.

#### 6.1.3.Lợi dụng các thiếu sót trong việc kiểm tra giá trị nhập hợp lệ

Hacker lợi dụng những ô nhập dữ liệu để gởi đi những đoạn mã bất kỳ khiến cho khiến cho hệ thống phải thực thi đoạn lệnh đó hay bị phá vỡ hoàn toàn.

### Kiểm tra tính đúng đắn của dữ liệu bằng ngôn ngữ phía trình duyệt

Do ngôn ngữ phía trình duyệt (JavaScript, VB Script) được thực thi trên trình duyệt nên hacker có thể sửa đổi mã nguồn để có thể vô hiệu hóa sự kiểm tra.

### Tràn bộ đệm

Một khối lượng dữ liệu được gửi cho ứng dụng vượt quá lượng dữ liệu được cấp phát khiến cho ứng dụng không thực thi được câu lệnh dự định kế tiếp mà thay vào đó phải thực thi một đoạn mã bất kỳ do hacker đưa vào hệ thống

### Chèn mã lệnh thực thi trên trình duyệt nạn nhân

Đây là kỹ thuật tấn công chủ yếu nhằm vào thông tin trên máy tính của người dùng hơn là vào hệ thống máy chủ. Bằng cách thêm một đoạn mã bất kỳ (thường được lập trình bằng ngôn ngữ JavaScript, VBScript), Hacker có thể thực hiện việc đánh cắp thông tin quan trọng như Cookie để từ đó trở thành người dùng hợp lệ của ứng dụng dựa trên những thông tin đánh cắp này.

### Chèn câu truy vấn SQL

Trong lập trình với cơ sở dữ liệu, người lập trình đã sai sót trong vấn đề kiểm tra giá trị nhập vào để từ đó hacker lợi dụng thêm vào những câu truy vấn hay những giá trị không hợp lệ để dễ dàng đăng nhập vào hệ thống.

## 6.2.CÁC KỸ THUẬT TẤN CÔNG VÀ BẢO MẬT ỨNG DỤNG WEB

### 6.2.1.Thao tác trên tham số truyền

Thao tác trên tham số truyền là kỹ thuật thay đổi thông tin quan trọng trên Cookie, URL, hay biến ẩn của form. Các kỹ thuật tấn công đều cần dùng đến các tham số này để hoàn thiện các bước tấn công của hacker. Có thể nói tham số truyền là đầu mối cho mọi hoạt động của hacker trong quá trình tấn công ứng dụng. ***a) Thao tác trên URL***

Ví dụ có một trang web ứng dụng cho phép thành viên được thay đổi mật khẩu

http://www.nganhang.com/example?user=thang&newpass=123

Với:

+ username là tên người cần thay đổi mật khẩu

+ newpass là mật khẩu mới cho username

Tuy nhiên bằng cách thay đổi tham số như sau:

http://www.nganhang.com/example?user=admin&newpass=111111

Hacker đã có thể thay đổi mật khẩu của admin bằng một mật khẩu mới bất kỳ, trong ví dụ này là ‘111111’.

### *b) Biện pháp khắc phục*

Để chống lại kiểu tấn công bằng thay đổi nội dung của một chuỗi URL, ứng dụng có thể áp dụng biện pháp sau:

* Sử dụng cơ chế bảng băm (hash table): Sau khi người dùng chứng thực thành công với một username, ứng dụng sẽ sinh ra một khóa tương ứng. Khóa này sẽ được lưu trên server cùng với biến username trong đối tượng bảng băm. Mỗi khi người dùng kết nối đến ứng dụng, khóa và username này sẽ được gửi đi và được so sánh với khóa và username trong bảng băm. Nếu tương ứng với bản ghi trong dữ liệu thì hợp lệ. Ngược lại server biết rằng người dùng đã thay đổi URL.
* Ngoài ra, với những thông tin có giá trị, cần mã hóa thông tin này trước khi cho hiển thị trên trình duyệt để tránh bị hacker sửa đổi. **6.2.2.Thao tác trên Biến ẩn Form**

### *a) Biến ẩn Form*

Biến ẩn Form không hiển thị trên màn hình trình duyệt nhưng người dùng có thể tìm thấy nội dung của nó trong “View | Page Source” trên menu trình duyệt, vì thế đây là một điểm yếu để hacker lợi dụng bằng cách thay đổi nội dung trang web và gửi đến trình chủ.

Ví dụ form gốc có nội dung như sau:

<form action = “http://www.tancong.com/cuahang.pl” method=”POST”>

…

<

input type=“hidden” name =“giaca” value=“99.99”

>

…

<

/form

>

Nếu không có sự thay đổi nào thì yêu cầu đến trình chủ có nội dung:

POST /cuahang.pl HTTP/1.0

…

Giaca=99.99

Nhưng hacker gán một giá trị khác cho trường “giaca”:

<form action=“http://www.tancong.com/cuahang.pl” method=“POST”>

…

<input type=“hidden” name=“giaca” value=“0.99”>

…

<

/form

>

Thì yêu cầu sẽ thay đổi:

POST /cuahang.pl HTTP/1.0

…

Giaca=0.99

### *b) Biện pháp khắc phục*

* Chỉ nên sử dụng biến ẩn của form để hiển thị dữ liệu trên trình duyệt, không được sử dụng giá trị của biến ẩn để thao tác trong xử lí ứng dụng.
* Ghép tên và giá trị của biến ẩn thành một chuỗi đơn. Sử dụng thuật toán mã hóa MD5 hoặc một kiểu hash một chiều khác để tổng hợp chuỗi đó và lưu nó vào một biến ẩn (hidden field) gọi là “chuỗi mẫu”. Khi giá trị trong form được gửi đi, các thao tác trong form được thực hiện lại với cùng một khóa mà ta định trước. Sau đó đem so sánh với “chuỗi mẫu”, nếu chúng không khớp nhau thì chứng tỏ giá trị trong biến ẩn đã bị thay đổi.
* Dùng một sessionID để tham chiếu đến thông tin được lưu trữ trên cơ sở dữ liệu.

#### 6.2.3.Thao tác trên Cookie

Vì cookie là thành phần lưu trữ thông tin bảo mật nhất nên Cookie thường được dùng để lưu giữ trạng thái cho giao thức HTTP hơn là biến ẩn form và biến URL. Nó còn được dùng để lưu trữ thông tin của người dùng khi sử dụng ứng dụng và những dữ liệu khác của session. Tất cả các loại cookie như persistent hay non-persistent, secure hay insecure đều có thể bị thay đổi bởi người dùng và được gởi về cho trình chủ. Do đó, hacker có thể thay đổi nội dung cookie để phá hoại ứng dụng.

**Một số biện pháp khắc phục**:

* Sử dụng đối tượng session lưu trữ thông tin quan trọng trên trình chủ. Khi ứng dụng cần kiểm tra thông tin của một người dùng, ứng dụng sẽ dùng sessionID của người dùng để chỉ đến thông tin của người dùng đó trong cache hay cơ sở dữ liệu.
* Xây dựng một cơ chế kiểm tra nội dung của cookie để tìm ra những giá trị không hợp lệ từ đó biết được cookie đó là giả. Ví dụ, nếu biến cờ “*người quản trị*” được thiết lập là đúng trong cookie, nhưng giá trị nhưng giá trị của số thứ tự người dùng trong cookie lại không giống như giá trị số thứ tự của “*người quản trị*” được lưu trữ trên server.
* Phương pháp cuối cùng là mã hóa cookie. Có một số phương pháp mã hóa như symmetric (dùng một khóa duy nhất cho cả mã hóa và giải mã) hay asymmetric (mã hóa dùng 2 khóa riêng biệt, một khóa dùng chung cho mã hóa và một khóa riêng để giải mã)

## 6.3.BẢO MẬT CƠ SỞ DỮ LIỆU (đọc thêm)

**6.3.1.Các kỹ thuật tấn công cơ sở dữ liệu**

### *a) Sửa đổi nội dung câu truy vấn*

Sửa đổi nội dung câu truy vấn (SQL Injection) là cách lợi dụng những lỗ hổng trong trong quá trình lập trình Web về phần truy xuất cơ sở dữ liệu. Đây không chỉ là khuyết điểm của riêng SQL Server mà nó còn là vấn đề chung cho các cơ sở dữ liệu khác như MS Access hay Oracle.

Khi Hacker gửi những dữ liệu (thông qua webform), ứng dụng web sẽ thực hiện và trả về cho trình duyệt kết quả câu truy vấn hay những thông báo lỗi có liên quan đến cơ sở dữ liệu. Và nhờ những thông tin này mà hacker biết được nội dung cơ sở dữ liệu và có thể điều khiển toàn bộ hệ thống ứng dụng.

Dưới đây là kỹ thuật SQL Injection đơn giản nhất để vượt qua các form đăng nhập:

Giả sử ứng dụng Web có đoạn mã sau:

“’AND PASSWORD= ‘”&Password& “’”

SQLQuery= “SELECT Username FROM User WHERE Username= ‘”&strUsername&

flag=GetQueryResult(SQLQuery)

If flag= “” then check=FALSE

Else check=TRUE

End if

Đoạn mã trên kiểm tra chuỗi nhập Username và Password. Nếu tồn tại trong bảng User thì kiểm tra là đúng và ngược lại thì biến kiểm tra gán bằng false.

Giá trị nhập vào là:

Username: ’OR ‘’=’

Password: ’or ‘’=’

Câu lệnh SQL lúc này như sau:

SQLQuery= “SELECT Username FROM User WHERE Username= ‘’ OR ‘’= ‘’ AND

PASSWORD= ‘’ OR ‘’= ‘’

Câu lệnh so sánh trên luôn luôn đúng (vì ‘’ luôn bằng ‘’). Do đó, câu điều kiện trong mệnh đề WHERE luôn luôn đúng. Giá trị tên người sử dụng đầu tiên trong bảng sẽ được chọn.

Kết hợp với các ký tự đặc biệt của SQL:

* Ký tự “;”: đánh dấu kết thúc một câu truy vấn
* Ký tự “--”: ẩn chuỗi ký tự phía sau nó trên cùng một dòng Hoặc giá trị nhập vào là:

Username: ’; drop table User-Password:

Câu lệnh SQL lúc này như sau:

SQLQuery= “SELECT Username FROM User WHERE Username= ‘’;drop table User-AND PASSWORD= ‘”&Password& “’”

Với câu lệnh trên thì bảng User sẽ bị xóa hoàn toàn.

Hoặc giá trị nhập vào là:

Username: admin’-Password:

Câu lệnh SQL như sau:

SQLQuery= “SELECT Username FROM User WHERE Username= ‘admin’-- AND

PASSWORD= ‘”&Password& “’”

Câu lệnh trên cho phép đăng nhập vào hệ thống với quyền admin mà không đòi hỏi password.

### *b) Tấn công dựa vào câu lệnh HAVING và câu lệnh kết hợp UNION*

HAVING sử dụng cùng chung với mệnh đề GROUP BY là phương pháp hữu hiệu để nhận thông tin bảng dữ liệu, trường dữ liệu.

Lệnh SELECT được dùng để lấy thông tin từ cơ sở dữ liệu. Thông thường vị trí có thể chèn thêm vào mệnh đề SELECT là sau WHERE. Để có thể trả về nhiều dòng thông tin trong bảng, thay đổi điều kiện trong mệnh đề WHERE bằng cách chèn thêm UNION SELECT.

Sau đây là những ví dụ được thực hiện khi không biết nội dung cơ sở dữ liệu dựa vào

HAVING, GROUP BY, UNION:

Đầu tiên, để biết tên bảng và tên trường mà câu truy vấn sử dụng, sử dụng câu điều kiện “having” như ví dụ sau:

Giá trị nhập vào:

Username: ’having 1=1--

Lỗi phát sinh:

Column ‘User.Username’is invalid in the select list because it is not contained in an aggregate function and there is no GROUP BY clause.

Nhờ vào lỗi phát sinh này mà biết được bảng sử dụng trong câu truy vấn là User và trong bảng tồn tại một trường tên là Username.

Sau đó sử dụng GROUP BY

Username: ‘group by User.Username having 1=1--

Lỗi phát sinh:

Column ‘User.Password’is invalid in the select list because it is not contained in an aggregate function and there is no GROUP BY clause.

Như vậy, Password là một trường của bảng User và được sử dụng trong câu truy vấn. Tiếp tục dùng GROUP BY cho đến khi biết được tất cả các trường trong bảng User tham gia vào câu truy vấn. Khi không còn báo lỗi cú pháp GROUP BY nữa thì chuyển qua công đoạn kiểm tra kiểu của từng trường trong bảng. Lúc này UNION được sử dụng.

Username: ’union select sum(Username) from User

Lệnh SUM() là lệnh tính tổng cho đối số bên trong dấu ngoặc. Đối số phải là kiểu số.

Nếu đối số không là kiểu số thì phát sinh lỗi như sau:

The sum or average aggregate operation **cannot take a varchar data type** as an argument.

Với thông báo lỗi trên thì Username chắc chắn phải là kiểu varchar. Với phương pháp trên dễ dàng xác định được từng trường trong bảng. Sau khi đã nhận đầy đủ thông tin trên thì hacker dễ dàng tự thêm thông tin vào bảng User.

Username: ’; insert into User(Username,Password) values(‘admin’, ‘’)--

Hacker thêm nội dung như ví dụ trên  bây giờ trở thành người quản trị mạng mà không cần mật khẩu để chứng thực.

### *c) Tấn công dựa vào câu lệnh INSERT*

Từ khóa INSERT dùng để đưa thông tin vào cơ sở dữ liệu. Thông thường câu lệnh INSERT được dùng trong các trường hợp như: thông tin đăng ký người sử dụng.

Kỹ thuật “;”, “--” được dùng như đã từng dùng với câu lệnh SELECT, phải đảm bảo đúng số lượng và kiểu giá trị được nhập vào nhằm tránh lỗi về cú pháp (nếu không xác định được kiểu dữ liệu có thể nhập tất cả là số).

### *d) Tấn công dựa vào STORED PROCEDURE*

Stored Procedure được sử dụng trong lập trình Web với mục đích giảm sự phức tạp trong ứng dụng và tránh sự tấn công trong kỹ thuật SQL Injection. Tuy nhiên hacker vẫn có thể lợi dụng những Store Procedure để tấn công vào hệ thống.

Ví dụ, Stored Procedure sp\_login gồm hai tham số là username và password.

Nếu nhập:

Username: nhimmap

Password: ‘;shutdown--

Lệnh gội Stored Procedure như sau:

Exec sp\_login ‘nhimmap’,‘’;shutdown--

Lệnh shutdown thực hiện dừng SQL Server ngay lập tức.

#### 6.3.2.Cách phòng chống

* Trong hầu hết trình duyệt, những ký tự nên được mã hóa trên địa chỉ URL trước khi được sử dụng
* Việc tấn công theo SQL Injection dựa vào những câu thông báo lỗi do đó việc phòng chống hay nhất vẫn là không cho hiển thị những thông điệp lỗi cho người dùng bằng cách những lỗi thông báo bằng một trang do người phát triển thiết kế mỗi khi lỗi xảy ra trên ứng dụng.
* Kiểm tra kỹ giá trị nhập vào của người dùng, thay thế những ký tự như ‘; v.v..
* Hãy loại bỏ những ký tự meta như “ ‘, /, \, ; “ và các ký tự extend như NULL, CR, LF, … trong các string nhận được từ:
  + Dữ liệu nhập do người dùng đệ trình o Các tham số từ URL
  + Các giá trị từ Cookie
* Đối với các giá trị numeric, hãy chuyển nó sang integer trước khi thực hiện câu truy vấn SQL, hoặc dùng ISNUMERIC để chắc chắn nó là một số integer.
* Dùng thuật toán để mã hóa dữ liệu.
* Viết câu lệnh SQL an toàn o Hạn chế viết SELECT \* FROM: Nếu bảng dữ liệu có bao nhiêu trường thì câu truy vấn ghi bấy nhiêu. Sử dụng \* sẽ nhanh chóng nhưng ở mức độ nào đó người tấn công có thể tận dụng lỗi phát sinh khi sử dụng \*.
  + Khi thiết kế dữ liệu trên bất kỳ hệ quản trị CSDL nào cũng nên tạo trường duy nhất (khóa chính) cho bảng. o Sử dụng thủ tục (Store Procedure) sẽ chạy nhanh hơn và an toàn hơn

## CÂU HỎI

1. Hãy nêu các kỹ thuật tấn công ứng dụng web?
2. Trình bày các hình thức bảo mật và phân tích ưu, nhược điểm của các hình thức bảo mật đó?
3. Trình bày các kỹ thuật tấn công cơ sở dữ liệu và cách phòng chống?
4. Hãy xây dựng một trang đăng nhập vào một ứng dụng web và cấu hình bảo mật cho ứng dụng này?

# DANH MỤC HÌNH

Trang

Hình 2.1.Mô hình máy khách, máy chủ…………………………………………………….17

Hình 2.2.Frameset – Columns…………………………………………………………........27

Hình 2.3.Frameset – Rows……………………………………………………………….....28

Hình 2.4.Frameset in frameset………………………………………………………………30

Hình 2.5.Form lấy dữ liệu đầu vào………………………………………………………….37

Hình 3.1.Mô hình ứng dụng 2 lớp ……………………………………………………….....42 Hình 3.2.Mô hình ứng dụng 3 lớp ………………………………………………………….43

Hình 3.3.Kiến trúc .Net Frame work………………………………………………………..44

Hình 3.4.Hệ thống tên miền ………………………………………………………………...47

Hình 3.5.Cách thức thi hành trang ASPX ………………………………………………….49

Hình 3.6.Quá trình xử lí tập tin ASPX ……………………………………………………..50

Hình 3.7.Kiểm tra thành công Web Server ………………………………………………....53

Hình 3.8.Các điều khiển HTML trên thanh công cụ ……………………………………….58

Hình 3.9.Chuyển điều khiển HTML thành điều khiển HTML Server …………………….59

Hình 3.10.Tạo nhóm cho các điều khiển RadioButton …………………………………….66

Hình 3.11.Sơ đồ xử lí kiểm tra dữ liệu nhập tại Client và Server …………………………67

Hình 5.1.Quá trình Request – Response của HTTP ……………………………………….98

Hình 5.2.Quan hệ giữa Session và Application …………………………………………...101