

MỤC LỤC

BÀI 1. TỔNG QUAN VỀ WWW – NGÔN NGỮ HTML.....	1
1. Mục tiêu:.....	1
2. Nội dung bài học.....	1
2.1 Lịch sử WWW	1
2.2 Nhập môn ngôn ngữ HTML	6
2.3 Trang và văn bản trên trang	13
2.4 Bảng biểu(Table) và trang khung(Frame).....	16
2.5 Multimedia trên trang Web.....	18
Câu hỏi ôn tập.....	21
BÀI 2. THIẾT KẾ WEB TĨNH	22
1. Mục tiêu:.....	22
2. Nội dung bài học.....	22
2.1 Tổng quan	22
2.2 Đối tượng Form	22
2.3 Các đối tượng của Form	23
2.4 Các yếu tố động trên trang	25
2.5 Liên kết	25
2.8 Hoàn thiện trang web tĩnh.....	29
Câu hỏi ôn tập.....	34
BÀI 3. XÂY DỰNG WEB ĐỘNG.....	35
1. Mục tiêu:.....	35
2. Nội dung bài học.....	35
2.1 Tổng quan về ASP.Net và ADO.Net	35
2.2 Các đối tượng ASP.Net.....	43
2.3 Các đối tượng ADO.Net	46
2.4 Điều khiển kiểm tra hợp lệ.....	53
2.5 Điều khiển hiển thị dữ liệu	54
2.8 Điều khiển kết nối dữ liệu.....	55
2.9 Thiết kế giao diện phía người dùng web.....	64
2.10 Thiết kế giao diện quản trị Web.....	66
2.11 Hoàn thiện Website động.....	67
Câu hỏi ôn tập.....	72
Tài liệu cần tham khảo:.....	73

BÀI 1. TỔNG QUAN VỀ WWW – NGÔN NGỮ HTML

Giới thiệu:

World Wide Web (www), gọi tắt là web, là một không gian thông tin toàn cầu mà mọi người có thể truy nhập (gửi và nhận thông tin) qua các máy tính nối với mạng Internet. Các tài liệu trên web được lưu trữ trong một hệ thống siêu văn bản (hypertext) đặt tại các máy Webserver nối mạng Internet.

HTML (tiếng Anh, viết tắt cho HyperText Markup Language, hay là "Ngôn ngữ Đánh dấu Siêu văn bản") là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẫu thông tin được trình bày trên World Wide Web. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho World Wide Web.

1. Mục tiêu:

- Hiểu được lịch sử của WWW;
- Hiểu được cấu trúc của một trang HTML;
- Có khả năng thiết kế được giao diện;
- Biết cách tổ chức được thông tin trong trang chủ;
- Thực hiện các thao tác an toàn với máy tính.

2. Nội dung bài học

2.1 Lịch sử WWW

Các khái niệm cơ bản

- Webpage: là nói đến một trang Web thường trình bày về một vấn đề nào đó như: phổ biến kiến thức, trao đổi, bàn luận, thông báo, quảng cáo...
- Website: là tập hợp nhiều trang Web mà các trang này có thể liên kết với nhau.
- Web Browser: là trình duyệt Web dùng để hiển thị xem các thông tin của trang Web yêu cầu.
- Web Server: là một máy chủ cung cấp các dịch vụ về Web trên môi trường Internet.
- Domain: là tên miền của một trang Web. Có rất nhiều tên miền thông thường thì tên miền .com dành cho các công ty xí nghiệp, tên miền .net dành cho các nhà cung cấp dịch vụ, thông tin, tên miền .org dành cho các cơ quan hành chính, các trang có nội dung thuộc về chính trị ...
- WWW: là dịch vụ cung cấp, truy tìm, trao đổi Internet. Được viết tắt từ ba chữ đầu của World Wide Web.
- Network: là mạng máy tính là hệ truyền thông và trao đổi dữ liệu được xây dựng bằng sự ghép nối vật lý hai hoặc nhiều máy tính.
- Administrator: là quản trị viên, người có quyền hạn cao nhất trên máy Server.
- Web Master: là chủ một Website người có quyền hạn cao nhất trên Website đó.
- LAN: là mạng cục bộ. Được viết tắt từ ba chữ đầu của Local Area Network.
- HTML !??: Được viết tắt từ bốn chữ đầu của Hyper Text Markup Language là một ngôn ngữ đánh dấu siêu văn bản. Ngôn ngữ này sử dụng các Tag để đánh dấu các thành phần hiển thị trong trang Web. Thông thường một trang Web được viết bằng ngôn ngữ HTML có cấu trúc tương tự như sau:

+ Web Tĩnh: Trang web tĩnh là trang web mà người dùng không có khả năng tương tác với nó. Trang web tĩnh không yêu cầu phải lập trình gì cả, nói một cách khác, xem

một trang web tĩnh như là xem một tờ báo (in trên giấy) nhưng có thêm hình ảnh và âm thanh...

- + Web Động: là trang web ở trên đó người dùng (người duyệt web) có thể tương tác, thay đổi các thông số, cách thông tin được hiển thị v.v. tùy ý. Cần lưu ý là các hình ảnh nhảy nhót "động đậy" (kiểu animated GIF) không hẳn là biểu hiện của một trang web động.

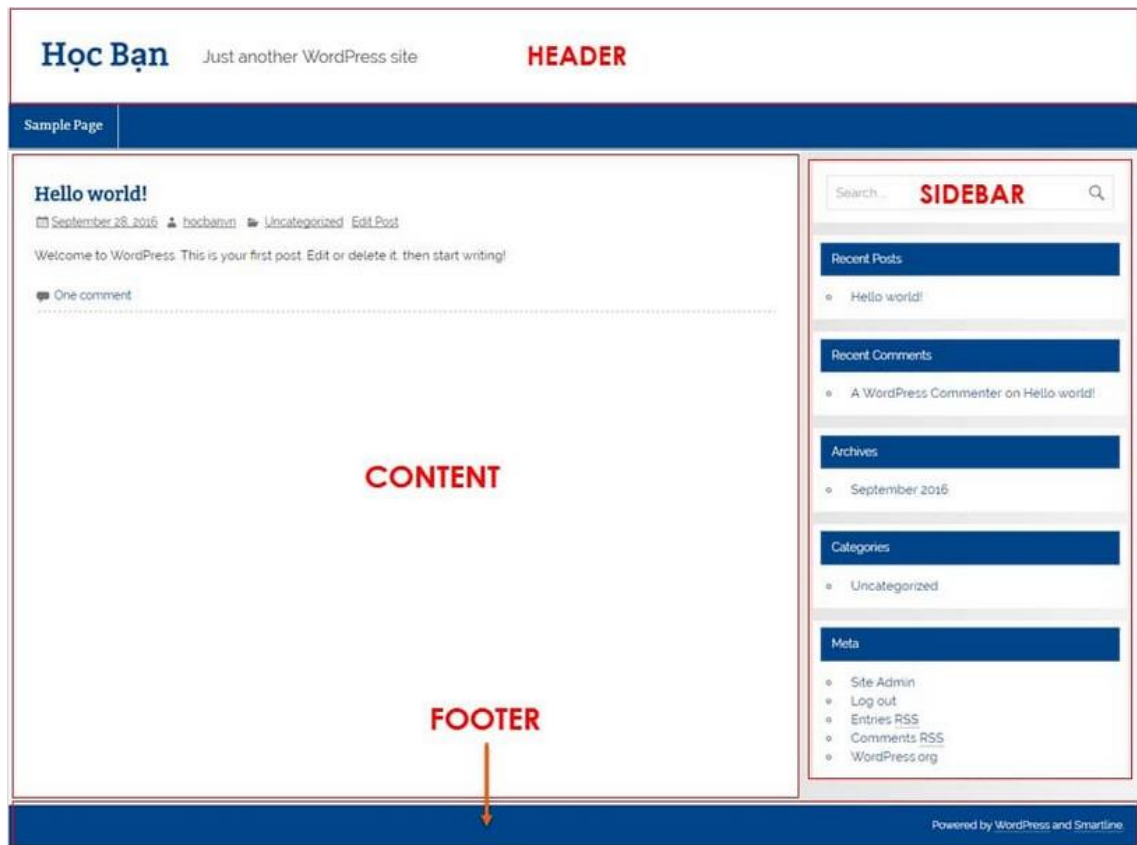
Thiết kế giao diện

- Thiết kế Giao diện dành cho Người sử dụng là một công việc trong quy trình phát triển một Web site. Tại Việt Nam, nhiều doanh nghiệp vừa và nhỏ không quan tâm chất lượng phần mềm, dịch vụ sau bán hàng mà thường quan tâm đến giao diện Web có đẹp hay không. Tuy nhiên đây lại là hạn chế của hầu hết các doanh nghiệp thiết kế Web. Trong phạm vi cho phép, giáo trình này xin trình bày các bước xây dựng Giao diện Web cho người sử dụng, nhằm giúp học sinh để đạt hiệu quả thiết kế giao diện cao tuân theo quy trình bao gồm các bước sau:

Quy trình thiết kế giao diện

- **Bước 1:** Xác định yêu cầu của khách hàng.
 - + Mục tiêu của giai đoạn này là xác định chính xác yêu cầu hoặc tư vấn cho khách hàng. Yêu cầu là một điều kiện hoặc khả năng mà hệ thống phải tuân theo hoặc có.
 - + Yêu cầu phải đạt những tiêu chí sau:
 - + Yêu cầu phải bao quát giao diện, chức năng, cấu trúc nội dung, đối tượng xem Web site.
 - + Trao đổi thông tin dựa trên các yêu cầu đã xác định trước khi tiếp cận khách hàng. Bạn phải nghiên cứu về yêu cầu chung của khách hàng trước khi tiếp cận.
 - + Xây dựng bảng câu hỏi logic để chuyển đổi sang phân tích yêu cầu nghiệp vụ, yêu cầu hệ thống đơn giản, dễ dàng.
 - + Đặt độ ưu tiên, lọc và theo dõi các yêu cầu.
 - + Đánh giá khách quan các chức năng và hiệu năng.
- **Bước 2:** Phác thảo ý tưởng.
 - + Mục tiêu của bước này là định hình bố cục của trang Web
 - + Banner không quá 1/3 màn hình thực của người sử dụng (màn hình thực là màn hình của trình duyệt có thể xem được trang Web, đã bỏ đi các thanh toolbar của trình duyệt Web).
 - + Sitebar không lớn quá 25% chiều rộng trang Web.
 - + Vùng template (theo chuẩn IWP)
 - + Vùng hiệu chỉnh.
 - + Vùng template là vùng không hiệu chỉnh hoặc hiệu chỉnh rất ít xuyên suốt các trang Web của Web site.
 - + Vùng hiệu chỉnh là vùng có thay đổi nội dung trong hầu hết các trang Web của Web site.

Ví dụ:



Hình 1.1 Bố cục của một trang web

- **Bước 3:** Đánh giá mẫu phác thảo.
- **Bước 4:** Thiết kế đồ họa bản đơn sắc.
- **Bước 5:** Phối màu cho giao diện Web.
- **Bước 6:** Xây dựng tài liệu về chuẩn CSS, clientsite script, ảnh, folder cho trang Web.
- **Bước 7:** Sử dụng các ngôn ngữ đánh dấu, lập trình để thiết kế giao diện.
- **Bước 8:** Kiểm tra giao diện trên các trình duyệt.
- **Bước 9:** Chuyển mã nguồn tới bộ phận phát triển Web.

Giới thiệu một số website mẫu.




[TRANG CHỦ](#)
[GIỚI THIỆU](#)
[TUYỂN SINH & GTVL](#)
[PHÒNG BAN](#)
[KHOA - TRUNG TÂM](#)
[ĐOÀN THỂ](#)
[LỊCH CÔNG TÁC](#)
[SINH VIÊN](#)

[THƯ VIỆN](#)

ỦY BAN NHÂN DÂN HUYỆN CỦ CHI
TRƯỜNG TRUNG CẤP NGHỀ CỦ CHI

THÔNG BÁO TUYỂN SINH

A. TRÌNH ĐỘ TRUNG CẤP - XÉT TUYỂN CHO CÁC ĐỐI TƯỢNG CÓ BẰNG TỐT NGHIỆP THCS (LỚP 9) TRỞ LÊN, GỒM CÁC NGHỀ:

1. ĐIỆN CÔNG NGHIỆP
2. ĐIỆN TỬ CÔNG NGHIỆP
3. CÔNG NGHỆ Ô TÔ
4. CẮT GỌT KIM LOẠI (CƠ KHÍ)
5. KỸ SỬA CHỮA - LẮP RÁP MÁY TÍNH
6. KỸ MÁY LẠNH & ĐIỀU HÒA KHÔNG KHÍ
7. MÁY THỜI TRANG
8. THỦ Y

- TUYỂN SINH VÀO THÁNG 03 HÀNG NĂM; KHAI GIẢNG VÀO THÁNG 03 HÀNG NĂM
- THỜI GIAN ĐÀO TẠO:
TỐT NGHIỆP THCS: 3 NĂM (HỌC NGHỀ VÀ VĂN HÓA); 2 NĂM (KHÔNG HỌC VĂN HÓA)
TỐT NGHIỆP THPT: 2 NĂM (KHÔNG HỌC VĂN HÓA)
- QUYỀN LỢI:
HỌC SINH TỐT NGHIỆP THCS, ĐIỆN CHÍNH SÁCH, MỘT NGHỀ ĐƯỢC MIỄN 100% HỌC PHÍ HỌC SINH KHÁ, GIỎI, XUẤT SẮC ĐƯỢC CẤP HỌC BỔNG KHUYẾN KHÍCH HỌC NGHỀ SẴN KHI TỐT NGHIỆP ĐƯỢC GỬI THIẾU VIỆC LÀM TẠI CÁC CTY, XÍ NGHIỆP, CÁC KHU CÔNG NGHIỆP TRONG VÀ NGOÀI HUYỆN HOẶC HỢP TÁC LAO ĐỘNG TẠI NHẬT BẢN, HÀN QUỐC...

B. TRÌNH ĐỘ SƠ CẤP:

- KHAI GIẢNG THUỶNG TUYỂN HÀNG THÁNG CHO MỖI ĐỐI TƯỢNG, THỜI GIAN HỌC TỪ 3 ĐẾN 6 THÁNG.
- TỔ CHỨC HỌC LUẬT ĐẠO THƯỜNG BƯỞNG ĐỘ VÀ THÍ SẮT BẠCH GỖ LƯU HANG 31 HÀNG THÁNG.
- TỔ CHỨC HỌC VÀ THÍ CHỨNG CHỈ CÔNG NGHỆ THÔNG TIN CƠ BẢN, NÂNG CAO HÀNG THÁNG.

C. LIÊN KẾT ĐÀO TẠO: LIÊN KẾT VỚI TRƯỜNG ĐẠI HỌC MỞ TP.HCM ĐÀO TẠO TRÌNH ĐỘ ĐẠI HỌC CÁC NGÀNH:

- KINH TẾ - LUẬT - KẾ TOÁN
- QUẢN TRỊ KINH DOANH - XÂY DỰNG

ĐỐI TƯỢNG XÉT TUYỂN: TỐT NGHIỆP THPT HOẶC TƯƠNG ĐƯƠNG, HOẶC TỐT NGHIỆP TRUNG CẤP.
THỜI GIAN ĐÀO TẠO: 4 NĂM, HỌC VÀO CÁC NGÀY THỨ BẢY, CHỦ NHẬT

Mọi chi tiết xin liên hệ: **PHÒNG GHI DANH. ĐT : (028) 38 920 312**
Địa chỉ : **Đường Nguyễn Đại Năng, Khu phố 1 , Thị trấn Củ Chi, Huyện Củ Chi, TP. HCM**

TIN TỨC TRƯỜNG

- ✓ Thông báo lần 4 Về việc kéo dài thời gian cho HSSV tạm nghỉ học để phòng, chống dịch bệnh Covid-19



Thông báo

- ✓ Thông báo lần 4 Về việc kéo dài thời gian cho HSSV tạm nghỉ học để...
- ✓ Thông báo lần 3 Về việc cho HSSV nghỉ học đến ngày 29/02/2020 để...
- ✓ Thông báo Về việc cho HSSV nghỉ học đến hết ngày 16/02/2020 để...
- ✓ Thông báo Về việc cho HSSV nghỉ học để phòng, chống dịch bệnh viêm...
- ✓ Thông báo thời hạn thu học phí lớp Kinh tế luật 2019-2020
- ✓ Thông báo V/v treo cờ Tổ quốc và nghi Lễ Quốc Khánh (02/9) năm 2019

Xem nhiều hơn

TUYỂN SINH - ĐÀO TẠO - GIỚI THIỆU VIỆC LÀM

- ✓ Lịch thi học kỳ 1 năm học 2019-2020 (lần 1, đợt 2)



Thông báo

- ✓ Lịch thi học kỳ 1 năm học 2019-2020 (lần 1, đợt 2)
- ✓ LỊCH THI HỌC KỲ II - KHOẢ 11, 12 (LẦN 2)
- ✓ KẾ HOẠCH THI TỐT NGHIỆP NGHỀ KHOẢ 10 TRÌNH ĐỘ TRUNG CẤP
- ✓ Danh sách dự thi Chứng chỉ ứng dụng công nghệ thông tin ngày...
- ✓ Thư ngỏ tuyển dụng của Công ty TNHH Phương Yển
- ✓ Công ty Phước Sơn tuyển dụng

Xem nhiều hơn

HOẠT ĐỘNG SINH VIÊN



Thông báo

ĐĂNG KÍ XÉT TUYỂN ONLINE

TIN TỨC MỚI

- ✓ Thông báo lần 4 Về việc kéo dài thời gian cho...
- ✓ Thông báo lần 3 Về việc cho HSSV nghỉ học đến...
- ✓ Thông báo Về việc cho HSSV nghỉ học đến hết...
- ✓ Thông báo Về việc cho HSSV nghỉ học để phòng,...
- ✓ Thông báo thời hạn thu học phí lớp Kinh tế luật...
- ✓ Thông báo V/v treo cờ Tổ quốc và nghi Lễ Quốc...

THỜI KHÓA BIỂU

- ✓ Máy thời trang
- ✓ Kỹ thuật sửa chữa & lắp ráp máy tính

Hình 1.2 Giao diện Website trường Trung Cấp Nghề Củ Chi



Hình 1.3 Giao diện website công ty TNHH Phú Nghĩa



Hình 1.4 Giao diện website công ty TNHH Mỹ Á

2.2 Nhập môn ngôn ngữ HTML

2.2.1. Cấu trúc chung của một trang web.

```
<html>
  <head>
    Trình bày phần mở đầu của trang web
  </head>
  <body>
    Trình bày phần thân của trang web
  </body>
</html>
```

2.2.2. Các thẻ định cấu trúc tài liệu.

2.2.2.1 HTML

Cặp thẻ này được sử dụng để xác nhận một tài liệu là tài liệu HTML, tức là nó có sử dụng các thẻ HTML để trình bày. Toàn bộ nội dung của tài liệu được đặt giữa cặp thẻ này.

Cú pháp:

```
<HTML>
... Toàn bộ nội của tài liệu được đặt ở đây
</HTML>
```

Trình duyệt sẽ xem các tài liệu không sử dụng thẻ <HTML> như những tệp văn bản bình thường.

2.2.2.2 HEAD

Thẻ HEAD được dùng để xác định phần mở đầu cho tài liệu.

Cú Pháp

```
<HEAD>
... Phần mở đầu (HEADER) của tài liệu được đặt ở đây
</HEAD>
```

2.2.2.3 TITLE

Cặp thẻ này chỉ có thể sử dụng trong phần mở đầu của tài liệu, tức là nó phải nằm trong thẻ phạm vi giới hạn bởi cặp thẻ <HEAD>.

Cú pháp:

```
<TITLE>Tiêu đề của tài liệu</TITLE>
```

2.2.2.4 BODY

Thẻ này được sử dụng để xác định phần nội dung chính của tài liệu – phần thân (body) của tài liệu. Trong phần thân có thể chứa các thông tin định dạng nhất định để đặt ảnh nền cho tài liệu, màu nền, màu văn bản siêu liên kết, đặt lề cho trang tài liệu... Những thông tin này được đặt ở phần tham số của thẻ.

Cú pháp:

```
<BODY>
.... phần nội dung của tài liệu được đặt ở đây
</BODY>
```

Trên đây là cú pháp cơ bản của thẻ **BODY**, tuy nhiên bắt đầu từ HTML 3.2 thì có nhiều thuộc tính được sử dụng trong thẻ **BODY**. Sau đây là các thuộc tính chính:

BACKGROUND=	Đặt một ảnh nào đó làm ảnh nền (background) cho văn bản. Giá trị của tham số này (phần sau dấu bằng) là URL của file ảnh. Nếu kích thước ảnh nhỏ hơn cửa sổ trình duyệt thì toàn bộ màn hình của sổ trình duyệt sẽ được lát kín bằng nhiều ảnh.
BGCOLOR=	Đặt màu nền cho trang khi hiển thị. Nếu cả hai tham số BACKGROUND và BGCOLOR cùng có giá trị thì trình duyệt sẽ hiển thị màu nền trước, sau đó mới tải ảnh lên phía trên.
TEXT=	Xác định màu chữ của văn bản, kể cả các đề mục.
ALINK=,VLINK=, LINK=	Xác định màu sắc cho các siêu liên kết trong văn bản. Tương ứng, alink (<i>active link</i>) là liên kết đang được kích hoạt - tức là khi đã được kích chuột lên; vlink (<i>visited link</i>) chỉ liên kết đã từng được kích hoạt;

2.2.3. Các thẻ định dạng khối

2.2.3.1 Thẻ P

Thẻ **<P>** được sử dụng để định dạng một đoạn văn bản.

Cú pháp:

<P> Nội dung đoạn văn bản **</P>**

2.2.3.2 Các thẻ định dạng đề mục H1/H2/H3/H4/H5/H6

HTML hỗ trợ 6 mức đề mục. Chú ý rằng đề mục chỉ là các chỉ dẫn định dạng về mặt logic, tức là mỗi trình duyệt sẽ thể hiện đề mục dưới một khuôn dạng thích hợp. Có thể ở trình duyệt này là font chữ 14 point nhưng sang trình duyệt khác là font chữ 20 point. Đề mục cấp 1 là cao nhất và giảm dần đến cấp 6. Thông thường văn bản ở đề mục cấp 5 hay cấp 6 thường có kích thước nhỏ hơn văn bản thông thường.

- Dưới đây là các thẻ dùng để định dạng văn bản ở dạng đề mục:

<H1> ... </H1>	Định dạng đề mục cấp 1
<H2> ... </H2>	Định dạng đề mục cấp 2
<H3> ... </H3>	Định dạng đề mục cấp 3
<H4> ... </H4>	Định dạng đề mục cấp 4
<H5> ... </H5>	Định dạng đề mục cấp 5
<H6> ... </H6>	Định dạng đề mục cấp 6

2.2.3.3 Thẻ xuống dòng BR

Thẻ này không có thẻ kết thúc tương ứng (**</BR>**), nó có tác dụng chuyển sang dòng mới. Lưu ý, nội dung văn bản trong tài liệu HTML sẽ được trình duyệt Web thể hiện liên tục, các khoảng trắng liền nhau, các ký tự tab, ký tự xuống dòng đều được coi như một khoảng trắng. Để xuống dòng trong tài liệu, bạn phải sử dụng thẻ **
**

2.2.3.4 Thẻ PRE

Để giới hạn đoạn văn bản đã được định dạng sẵn bạn có thể sử dụng thẻ **<PRE>**. Văn bản ở giữa hai thẻ này sẽ được thể hiện giống hệt như khi chúng được đánh vào, ví dụ dấu xuống dòng trong đoạn văn bản giới hạn bởi thẻ **<PRE>** sẽ có ý nghĩa chuyển sang dòng mới (trình duyệt sẽ không coi chúng như dấu cách)

Cú pháp:

<PRE>Văn bản đã được định dạng</PRE>

2.2.4. Các thẻ định dạng danh sách

2.2.4.1 Danh sách thông thường

Cú pháp:

```
<UL>
    <LI> Mục thứ nhất
    <LI> Mục thứ hai
</UL>
```

- Có 4 kiểu danh sách:
 - + Danh sách không sắp xếp (hay không đánh số)
 - + Danh sách có sắp xếp (hay có đánh số) , mỗi mục trong danh sách được sắp xếp thứ tự.
 - + Danh sách thực đơn <MENU>
 - + Danh sách phân cấp <DIR>
- Với nhiều trình duyệt, danh sách phân cấp và danh sách thực đơn giống danh sách không đánh số, có thể dùng lẫn với nhau. Với thẻ OL ta có cú pháp sau:

+ Trong đó:

```
<OL TYPE=1/a/A/i/I>
    <LI>Mục thu nhất
    <LI>Mục thu hai
    <LI>Mục thu ba
</OL>
```

- + TYPE
 - =1 Các mục được sắp xếp theo thứ tự 1, 2, 3...
 - =a Các mục được sắp xếp theo thứ tự a, b, c...
 - =A Các mục được sắp xếp theo thứ tự A, B, C...
 - =i Các mục được sắp xếp theo thứ tự i, ii, iii...
 - =I Các mục được sắp xếp theo thứ tự I, II, III...

Ngoài ra còn thuộc tính **START**= xác định giá trị khởi đầu cho danh sách.

Thẻ < LI > có thuộc tính **TYPE**= xác định ký hiệu đầu dòng (bullet) đứng trước mỗi mục trong danh sách. Thuộc tính này có thể nhận các giá trị : **disc** (chấm tròn đậm); **circle** (vòng tròn); **square** (hình vuông).

2.2.2.5. Các thẻ định dạng ký tự.

Các thẻ định dạng in ký tự

- Sau đây là các thẻ được sử dụng để quy định các thuộc tính như in nghiêng, in đậm, gạch chân... cho các ký tự, văn bản khi được thể hiện trên trình duyệt.

 	In chữ đậm
<I> ... </I> ... 	In chữ nghiêng
<U> ... </U>	In chữ gạch chân

<DFN>	Đánh dấu đoạn văn bản giữa hai thẻ này là định nghĩa của một từ. Chúng thường được in nghiêng hoặc thể hiện qua một kiểu đặc biệt nào đó.
<S> ... </S> <STRIKE> ... </STRIKE>	In chữ bị gạch ngang.
<BIG> ... </BIG>	In chữ lớn hơn bình thường bằng cách tăng kích thước font hiện thời lên một. Việc sử dụng các thẻ <BIG> lồng nhau tạo ra hiệu ứng chữ tăng dần. Tuy nhiên đối với mỗi trình duyệt có giới hạn về kích thước đối với mỗi font chữ, vượt quá giới hạn này, các thẻ <BIG> sẽ không có ý nghĩa.
<SMALL> ... </SMALL>	In chữ nhỏ hơn bình thường bằng cách giảm kích thước font hiện thời đi một. Việc sử dụng các thẻ <SMALL> lồng nhau tạo ra hiệu ứng chữ giảm dần. Tuy nhiên đối với mỗi trình duyệt có giới hạn về kích thước đối với mỗi font chữ, vượt quá giới hạn này, các thẻ <SMALL> sẽ không có ý nghĩa.
^{...}	Định dạng chỉ số trên (SuperScript)
_{...}	Định dạng chỉ số dưới (SubScript)
<BASEFONT>	Định nghĩa kích thước font chữ được sử dụng cho đến hết văn bản. Thẻ này chỉ có một tham số size= xác định cỡ chữ. Thẻ <BASEFONT> không có thẻ kết thúc.
 ... 	Chọn kiểu chữ hiển thị. Trong thẻ này có thể đặt hai tham số size= hoặc color= xác định cỡ chữ và màu sắc đoạn văn bản nằm giữa hai thẻ. Kích thước có thể là tuyệt đối (nhận giá trị từ 1 đến 7) hoặc tương đối (+2,-4...) so với font chữ hiện tại.

Căn lề văn bản trong trang Web

- Trong trình bày trang Web của mình các bạn luôn phải chú ý đến việc căn lề các văn bản để trang Web có được một bố cục đẹp. Một số các thẻ định dạng như **P**, **Hn**, **IMG...** đều có tham số **ALIGN** cho phép bạn căn lề các văn bản nằm trong phạm vi giới hạn bởi của các thẻ đó.

- Các giá trị cho tham số **ALIGN**:

LEFT	Căn lề trái
CENTER	Căn giữa trang
RIGHT	Căn lề phải

- Ngoài ra, ta có thể sử dụng thẻ **CENTER** để căn giữa trang một khối văn bản.

+ **Cú pháp:**

<CENTER>Văn bản sẽ được căn giữa trang</CENTER>

Các ký tự đặc biệt

- Ký tự & được sử dụng để chỉ chuỗi ký tự đi sau được xem là một thực thể duy nhất.

Ký tự; được sử dụng để tách các ký tự trong một từ.

Ký tự	Mã ASCII	Tên chuỗi
<	<	<
>	>	>
&	&	&

Sử dụng màu sắc trong thiết kế các trang Web

- Một màu được tổng hợp từ ba thành phần màu chính, đó là: Đỏ (Red), Xanh lá cây (Green), Xanh nước biển (Blue). Trong HTML một giá trị màu là một số nguyên dạng hexa (hệ đếm cơ số 16) có định dạng như sau: **#RRGGBB**

Trong đó:

- + **RR** - là giá trị màu Đỏ.
- + **GG** - là giá trị màu Xanh lá cây.
- + **BB** - là giá trị màu Xanh nước biển.

Màu sắc có thể được xác định qua thuộc tính bgcolor= hay color=. Sau dấu bằng có thể là giá trị RGB hay tên tiếng Anh của màu. Với tên tiếng Anh, ta chỉ có thể chỉ ra 16 màu trong khi với giá trị RGB ta có thể chỉ tới 256 màu.

Sau đây là một số giá trị màu cơ bản:

Màu sắc	Giá trị	Tên tiếng Anh
Đỏ	#FF0000	RED
Đỏ sẫm	#8B0000	DARKRED
Xanh lá cây	#00FF00	GREEN
Xanh nhạt	#90EE90	LIGHTGREEN
Xanh nước biển	#0000FF	BLUE
Vàng	#FFFF00	YELLOW
Vàng nhạt	#FFFFE0	LIGHTYELLOW
Trắng	#FFFFFF	WHITE
Đen	#000000	BLACK
Xám	#808080	GRAY
Nâu	#A52A2A	BROWN
Tím	#FF00FF	MAGENTA
Tím nhạt	#EE82EE	VIOLET
Hồng	#FFC0CB	PINK
Da cam	#FFA500	ORANGE
Màu đồng phục hải quân	#000080	NAVY
	#4169E1	ROYALBLUE

Cú pháp:

```
<BODY
  LINK           = color
  ALINK          = color
  VLINK          = color
  BACKGROUND     = url
  BGCOLOR        = color
  TEXT           = color
  TOPMARGIN      = pixels
  RIGHTMARGIN    = pixels
  LEFTMARGIN     = pixels
>
    .... phần nội dung của tài liệu được đặt ở đây
</BODY>
```

Sau đây là ý nghĩa các tham số của thẻ BODY:

Các tham số	ý nghĩa
LINK	Chỉ định màu của văn bản siêu liên kết
ALINK	Chỉ định màu của văn bản siêu liên kết đang đang chọn
VLINK	Chỉ định màu của văn bản siêu liên kết đã từng mở
BACKGROUND	Chỉ định địa chỉ của ảnh dùng làm nền
BGCOLOR	Chỉ định màu nền
TEXT	Chỉ định màu của văn bản trong tài liệu
SCROLL	YES/NO - Xác định có hay không thanh cuộn
TOPMARGIN	Lề trên
RIGHTMARGIN	Lề phải
LEFTMARGIN	Lề trái

2.2.5 Thiết kế web với CSS

2.2.5.1 Định nghĩa

CSS (cascading style sheet) cho phép định dạng (font chữ, kích thước, màu sắc) của các đối tượng sử dụng css đó. ưu điểm của CSS là tách riêng phần định dạng ra riêng khỏi phần nội dung do đó rất tiện khi muốn thay đổi giao diện của một website.

2.2.5.2 Cách sử dụng

Khai báo CSS trong file HTML

```
<head>
  <title><title>
    <style type="text/css">
      nội dung các css
    </style>
</head>
<body>
  Nội dung của web và sử dụng các định dạng của CSS
</body>
```

Sử dụng CSS được định nghĩa từ một file riêng:

```
<head>
    <title><title>
    <link rel="stylesheet" type="text/css" href="style.css" />
</head>
```

Hoặc

```
<head>
    <title><title>
    <style type="text/css"> @import url(Path To stylesheet.css) </style>
</head>
```

2.2.5.3 Đặc điểm trang web khi sử dụng CSS

Tiện lợi của việc định nghĩa một file css sau đó gọi vào trong trang HTML thay vì định nghĩa các CSS trực tiếp trong HTML.

- Code CSS nằm riêng trong một file nên dễ quản lý khủng bị rối.
- Nếu có nhiều file cùng sử dụng một file css thì chỉ cần thay đổi file css thì phân định dạng của tất cả các file sử dụng nó cũng được cập nhật (tiện lợi khi update giao diện cho website).
- Đỡ tốn băng thông đường truyền (bandwidth) hơn (do file được tải (load) một lần và được dùng lại cho các lần sau).

2.2.5.4 Cú pháp CSS

Tên_css { thuộc_tính: giá_trị_của_thuộc_tính; }

Ví dụ:

```
body {
    background: #eeeeee;
    font-family: Verdana, Arial, serif;
}
```

2.2.5.5 Một số tích chất của CSS

Tính kế thừa:

Như ví dụ trên thì toàn bộ các tag HTML có tên body + những tag HTML nằm trong body đều được định dạng theo body, nghĩa là background: #eeeeee và font-family: Verdana, Arial, serif; mà body là thẻ lớn nhất chứa nội dung của website cho nên tất cả các tag khác đều sử dụng các định dạng của body.

Trong trường hợp muốn sử dụng một định dạng khác trong một đối tượng nhỏ hơn body giả sử: p thì chỉ việc định nghĩa thêm đối tượng đó p{font-family: Tahoma, serif;}. Lúc này tất cả nội dung trong thẻ HTML đều có font là Tahoma chứ không phải Verdana của body.

Tính kết hợp:

Có thể định nghĩa nhiều css cùng một thuộc tính.

- Ví dụ: **h1, h2, h3, h4, h5, h6 {color: #009900; font-family: Georgia, sans-serif;}**

Như vậy, định nghĩa cùng một thuộc tính cho tất cả các tag h1, h2, h3, h4, h5, h6 thay vì phải định nghĩa:

```
h1{color: #009900; font-family: Georgia, sans-serif; }
h2 {...}
```



```
h3 {...}
```

```
...
```

```
h6 { color: #009900; font-family: Georgia, sans-serif; }
```

2.2.5.6 CSS Classes

Class: cho phép định dạng style của các đối tượng(table,td,div,span...), có thể sử dụng lặp đi lặp lại nhiều lần trong cùng một file HTML

Ví dụ:

```
<style>
.php { color: red; }
</style>
<span class="php">SPAN</span>
<p class="php">P</p>
<div class="php">DIV</div>
```

Kết quả:

SPAN

P

DIV

2.2.5.7 IDs

Cũng giống như **class** nhưng **ID** chỉ được sử dụng một lần, nếu sử dụng nhiều lần một id thì vẫn được tuy nhiên nó không đúng chuẩn của w3c và sẽ gặp rắc rối khi dùng ID trong javascript.

Ví dụ:

```
<style>
#php {color: red;}
</style>
<div id="php">ABC</div>
```

Kết quả: ABC (Chữ màu đỏ)

Chú ý: sử dụng dấu # cho IDs và dấu . cho class

2.3 Trang và văn bản trên trang

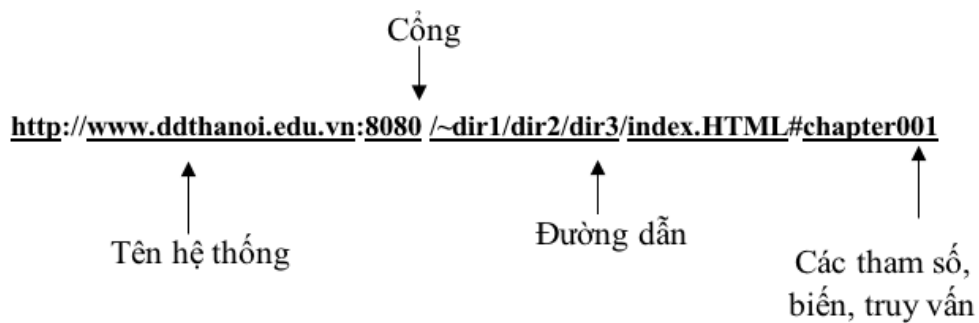
2.3.1 Chọn kiểu cho văn bản

Cú pháp:

```
<FONT
FACE      = font-name
COLOR     = color
SIZE      = n >
...
</FONT>
```

2.3.2 Khái niệm văn bản siêu liên kết

Văn bản siêu liên kết hay còn gọi là siêu văn bản là một từ, một cụm từ hay một câu trên trang Web được dùng để liên kết tới một trang Web khác. Siêu văn bản là môi trường trong đó chứa các liên kết (link) của các thông tin. Do WWW cấu thành từ nhiều hệ thống khác nhau, cần phải có một quy tắc đặt tên thống nhất cho tất cả các văn bản trên Web. Quy tắc đặt tên đó là URL (Universal Resource Locator).



Hình 1.5 Thành phần của URL

Các thành phần của URL được minh họa ở hình trên.

Dịch vụ: Là thành phần bắt buộc của URL. Nó xác định cách thức trình duyệt của máy khách liên lạc với máy phục vụ như thế nào để nhận dữ liệu. Có nhiều dịch vụ như **http**, **wais**, **ftp**, **gopher**, **telnet**.

Tên hệ thống: Là thành phần bắt buộc của URL. Có thể là tên miền đầy đủ của máy phục vụ hoặc chỉ là một phần tên đầy đủ – trường hợp này xảy ra khi văn bản được yêu cầu vẫn nằm trên miền của bạn. Tuy nhiên nên sử dụng đường dẫn đầy đủ.

Cổng: Không là thành phần bắt buộc của URL. Cổng là địa chỉ socket của mạng dành cho một giao thức cụ thể. Giao thức http ngầm định nối với cổng 8080.

Đường dẫn thư mục: Là thành phần bắt buộc của URL. Phải chỉ ra đường dẫn tới file yêu cầu khi kết nối với bất kỳ hệ thống nào. Có thể đường dẫn trong URL khác với đường dẫn thực sự trong hệ thống máy phục vụ. Tuy nhiên có thể rút gọn đường dẫn bằng cách đặt biệt danh (alias). Các thư mục trong đường dẫn cách nhau bởi dấu gạch chéo (/).

Tên file: Không là thành phần bắt buộc của URL. Thông thường máy phục vụ được cấu hình sao cho nếu không chỉ ra tên file thì sẽ trả về file ngầm định trên thư mục được yêu cầu. File này thường có tên là index.html, index.htm, default.html hay default.htm. Nếu cũng không có các file này thì thường kết quả trả về là danh sách liệt kê các file hay thư mục con trong thư mục được yêu cầu

Các tham số: Không là thành phần bắt buộc của URL. Nếu URL là yêu cầu tìm kiếm trên một cơ sở dữ liệu thì truy vấn sẽ gắn vào URL, đó chính là đoạn mã đăng sau dấu chấm hỏi (?). URL cũng có thể trả lại thông tin được thu thập từ form. Trong trường hợp dấu thăng (#) xuất hiện đoạn mã đăng sau là tên của một vị trí (location) trong file được chỉ ra.

Để tạo ra một siêu văn bản chúng ta sử dụng thẻ **<A>**.

Cú pháp:

```
<A   HREF      = url
      NAME      = name
      TABINDEX  = n
      TITLE     = title
      TARGET    = _blank / _self >
... siêu văn bản
</A>
```

Ý nghĩa các tham số:

HREF	Địa chỉ của trang Web được liên kết, là một URL nào đó.
NAME	Đặt tên cho vị trí đặt thẻ.
TABLEINDEX	Thứ tự di chuyển khi ấn phím Tab
TITLE	Văn bản hiển thị khi di chuột trên siêu liên kết.
TARGET	Mở trang Web được liên trong một cửa sổ mới (<i>_blank</i>) hoặc trong cửa sổ hiện tại (<i>_self</i>), trong một frame (tên frame).

2.3.3 Địa chỉ tương đối

URL được trình bày ở trên là URL tuyệt đối. Ngoài ra còn có URL tương đối hay còn gọi là URL không đầy đủ. Địa chỉ tương đối sử dụng sự khác biệt tương đối giữa văn bản hiện thời và văn bản cần tham chiếu tới. Các thành phần trong URL được ngăn cách bằng ký tự ngăn cách (ký tự gạch chéo /). Để tạo ra URL tương đối, đầu tiên phải sử dụng ký tự ngăn cách. URL đầy đủ hiện tại sẽ được sử dụng để tạo nên URL đầy đủ mới. Nguyên tắc là các thành phần bên trái dấu ngăn cách của URL hiện tại được giữ nguyên, các thành phần bên phải được thay thế bằng thành phần URL tương đối. Chú ý rằng trình duyệt không gửi URL tương đối, nó bổ sung vào URL cơ sở đã xác định trước thành phần URL tương đối xác định sau thuộc tính href=. Ký tự đầu tiên sau dấu bằng sẽ xác định các thành phần nào của URL hiện tại sẽ tham gia để tạo nên URL mới.

Ví dụ, nếu URL đầy đủ là: <http://it-department.vnuh.edu.vn/HTML/index.htm> thì:

- **Dấu hai chấm (:)** chỉ dịch vụ giữ nguyên nhưng thay đổi phần còn lại. Ví dụ <http://www.fpt.com/> sẽ tải trang chủ của máy phục vụ **www.fpt.com** với cùng dịch vụ http.
- **Dấu gạch chéo (/)** chỉ dịch vụ và máy phục vụ giữ nguyên nhưng toàn bộ đường dẫn thay đổi. Ví dụ </Javascript/index.htm> sẽ tải file **index.htm** của thư mục **Javascript** trên máy phục vụ **www.it-department.vnuh.edu.vn**.
- **Không có dấu phân cách** chỉ có tên file là thay đổi. Ví dụ **index1.htm** sẽ tải file **index1.htm** ở trong thư mục **HTML** của máy phục vụ **www.it-department.vnuh.edu.vn**.
- **Dấu thăng (#)**: chỉ dịch vụ, máy phục vụ, đường dẫn và cả tên file giữ nguyên, chỉ thay đổi vị trí trong file.
- Do đường dẫn được xem là đơn vị độc lập nên có thể sử dụng phương pháp đường dẫn tương đối như trong UNIX hay MS-DOS (tức là . chỉ thư mục hiện tại còn .. chỉ thư mục cha của thư mục hiện tại).
- URL cơ sở có thể được xác định bằng thẻ <BASE>.

2.3.4 Kết nối mailto

Nếu đặt thuộc tính href= của thẻ <a> giá trị mailto:address@domain thì khi kích hoạt kết nối sẽ kích hoạt chức năng thư điện tử của trình duyệt.

```
<ADDRESS>
Trang WEB này được
<A href="mailto:webmaster@vnuh.edu.vn">
WEBMASTER
<\A> bảo trì
<\ADDRESS>
```

2.3.5 Vẽ một đường thẳng nằm ngang

Cú pháp:

```
<HR  
    ALIGN      = LEFT / CENTER / RIGHT  
    COLOR      = color  
    NOSHADE  
    SIZE       = n  
    WIDTH      = width  
>
```

Ý nghĩa các tham số:

ALIGN	Căn lề (căn trái, căn phải, căn giữa)
COLOR	Đặt màu cho đường thẳng
NOSHADE	Không có bóng
SIZE	Độ dày của đường thẳng
WIDTH	Chiều dài (tính theo pixel hoặc % của bề rộng cửa sổ trình duyệt).

Thẻ này giống như thẻ **BR**, nó cũng không có thẻ kết thúc tương ứng.

2.4 Bảng biểu(Table) và trang khung(Frame)

2.4.1 Các thẻ định dạng bảng biểu (table)

Sau đây là các thẻ tạo bảng chính:

<TABLE> ... </TABLE>	Định nghĩa một bảng
<TR> ... </TR>	Định nghĩa một hàng trong bảng
<TD> ... </TD>	Định nghĩa một ô trong hàng
<TH> ... </TH>	Định nghĩa ô chứa tiêu đề của cột
<CAPTION> ... </CAPTION>	Tiêu đề của bảng

Cú pháp:

```
<TABLE  
    ALIGN      = LEFT / CENTER / RIGHT  
    BORDER      = n  
    BORDERCOLOR = color  
    BORDERCOLORDARK = color  
    BORDERCOLORLIGHT = color  
    BACKGROUND = url  
    BGCOLOR     = color  
    CELSPACING  = spacing  
    CELLPADDING = pading  
>  
<CAPTION>Tiêu đề của bảng biểu</CAPTION>  
... Định nghĩa các dòng  
<TR  
    ALIGN = LEFT/CENTER/RIGHT  
    VALIGN = TOP/MIDDLE/BOTTOM>
```

... Định nghĩa các ô trong dòng	
<TD	
ALIGN	= LEFT / CENTER / RIGHT
VALIGN	= TOP / MIDDLE / BOTTOM
BORDERCOLOR	= color
BORDERCOLORDARK	= color
BORDERCOLORLIGHT	= color
BACKBROUND	= url
BGCOLOR	= color
COLSPAN	= n
ROWSPAN	= n
>	
... Nội dung của ô	
</TD>	
...	
</TR>	
...	
</TABLE>	

Ý nghĩa các tham số:

ALIGN / VALIGN	Căn lề cho bảng và nội dung trong mỗi ô.
BORDER	Kích thước đường kẻ chia ô trong bảng, được đo theo pixel. Giá trị 0 có nghĩa là không xác định lề, giữa các ô trong bảng chỉ có một khoảng trắng nhỏ để phân biệt. Nếu chỉ để border thì ngầm định border=1. Với những bảng có cấu trúc phức tạp, nên đặt lề để người xem có thể phân biệt rõ các dòng và cột.
BORDERCOLOR	Màu đường kẻ
BORDERCOLORDARK BORDERCOLORLIGHT	Màu phía tối và phía sáng cho đường kẻ nổi.
BACKGROUND	Địa chỉ tới tệp ảnh dùng làm nền cho bảng
BGCOLOR	Màu nền
CELLSPACING	Khoảng cách giữa các ô trong bảng
CELLPADDING	Khoảng cách giữa nội dung và đường kẻ trong mỗi ô của bảng.
COLSPAN	Chỉ định ô sẽ kéo dài trong bao nhiêu cột
ROWSPAN	Chỉ định ô sẽ kéo dài trong bao nhiêu hàng

2.4.2 các thể trình bày mẫu biểu (Form)

2.4.2.1 HTML Forms

Các HTML Form có thể có các hộp văn bản, hộp danh sách lựa chọn, nút bấm, nút chọn...

2.4.2.2 Tạo Form

Để tạo ra một form trong tài liệu HTML, ta sử dụng thẻ FORM với cú pháp như sau:

Cú pháp:

```
<FORM
    ACTION    = url
    METHOD    = GET | POST
    NAME      = name
    TARGET    = frame_name | _blank | _self >
    <!-- Các phần tử điều khiển của form được đặt ở đây -->
    <INPUT ...>
    <INPUT ...>
</FORM>
```

Trong đó

ACTION	Địa chỉ sẽ gửi dữ liệu tới khi form được submit (có thể là địa chỉ tới một chương trình CGI, một trang ASP...).
METHOD	Phương thức gửi dữ liệu.
NAME	Tên của form.
TARGET	Chỉ định cửa sổ sẽ hiển thị kết quả sau khi gửi dữ liệu từ form đến server.

Đặt các đối tượng điều khiển (như hộp văn bản, ô kiểm tra, nút bấm...) vào trang Web

Cú pháp thẻ INPUT:

```
<INPUT
    ALIGN = LEFT | CENTER | RIGHT
    TYPE  = BUTTON | CHECKBOX | FILE | IMAGE | PASSWORD | RADIO
          | RESET | SUBMIT | TEXT
    VALUE = value >
```

2.4.2.3 Tạo một danh sách lựa chọn

Cú pháp:

```
<SELECT NAME="tên danh sách" SIZE="chiều cao">
    <OPTION VALUE=value1 SELECTED> Tên mục chọn thứ nhất
    <OPTION VALUE=value2> Tên mục chọn thứ hai
    <!-- Danh sách các mục chọn -->
</SELECT>
```

2.4.2.4 Tạo hộp soạn thảo văn bản

Cú pháp:

```
<TEXTAREA COLS=số cột ROWS=số hàng NAME=tên>
    Văn bản ban đầu
</TEXTAREA>
```

2.5 Multimedia trên trang Web

2.5.1 Các thẻ chèn âm thanh hình ảnh

2.5.1.1 Giới thiệu

Liên kết với file đa phương tiện cũng tương tự như liên kết bình thường. Tuy vậy phải đặt tên đúng cho file đa phương tiện. Phần mở rộng của file phải cho biết kiểu của file.

Kiểu	Phân mở rộng	Mô tả
Image/GIF	.gif	Viết tắt của Graphics Interchange Format. Khuôn dạng này xuất hiện khi mọi người có nhu cầu trao đổi ảnh trên nhiều hệ thống khác nhau. Nó được sử dụng trên tất cả các hệ thống hỗ trợ giao diện đồ họa. Định dạng GIF là định dạng chuẩn cho mọi trình duyệt WEB. Nhược điểm của nó là chỉ thể hiện được 256 màu. Mở rộng của chuẩn này là GIF89, được thêm nhiều chức năng cho các ứng dụng đặc biệt như làm ảnh nền trong suốt - tức là ảnh có thể nổi bằng cách làm màu nền giống với màu nền của trình duyệt.
Image/JPEG	.jpeg	Viết tắt của Joint Photographic Expert Group. Là khuôn dạng ảnh khác nhưng có thêm khả năng nén. Ưu điểm nổi bật của khuôn dạng này là lưu trữ được hàng triệu màu và độ nén cao nên kích thước file ảnh nhỏ hơn và thời gian download nhanh hơn. Nó là cơ sở cho khuôn dạng MPEG. Tất cả các trình duyệt đều có khả năng xem ảnh JPEG.
Image/TIFF	.tiff	Viết tắt của Tagged Image File Format. Được Microsoft thiết kế để quét ảnh từ máy quét cũng như tạo các ấn phẩm.
Text/HTML	.HTML .htm	
PostScript	.eps, .ps	Được tạo ra để hiển thị và in các văn bản có chất lượng cao.
Adobe Acrobat	.pdf	Viết tắt của Portable Document Format. Acrobat cũng sử dụng các siêu liên kết ngay trong văn bản cũng giống như HTML . Từ phiên bản 2.0, các sản phẩm của Acrobat cho phép liên kết giữa nhiều văn bản. Ưu điểm lớn nhất của nó là khả năng WYSISYG.
Video/MPEG	.mpeg	Viết tắt của Motion Picture Expert Group, là định dạng dành cho các loại phim (video). Đây là khuôn dạng thông dụng nhất dành cho phim trên WEB.
Video/AVI	.avi	Là khuôn dạng phim do Microsoft đưa ra.
Video/QuickTime	.mov	Do Apple Computer đưa ra, chuẩn video này được cho là có nhiều ưu điểm hơn MPEG và AVI. Mặc dù đã được tích hợp vào nhiều trình duyệt nhưng vẫn chưa phổ biến bằng hai loại định dạng trên.
Sound/AU	.au	
Sound/MIDI	.mid	Là khuôn dạng dành cho âm nhạc điện tử hết sức thông dụng được nhiều trình duyệt trên các hệ thống khác nhau hỗ trợ. File Midi được tổng hợp số hoá trực tiếp từ máy tính.
Sound/RealAudio	.ram	Định dạng audio theo dòng. Một bất tiện khi sử dụng các định dạng khác là file âm thanh thường có kích thước lớn - do vậy thời gian tải xuống lâu, Trái lại

Kiểu	Phần mở rộng	Mô tả
		audio dòng bắt đầu chơi ngay khi tải được một phần file trong khi vẫn tải về các phần khác. Mặc dù file theo định dạng này không nhỏ hơn so với các định dạng khác song chính khả năng dòng đã khiến định dạng này phù hợp với khả năng chơi ngay lập tức.
VRML	.vrmI	Viết tắt của Virtual Reality Modeling Language. Các file theo định dạng này cũng giống như HTML . Tuy nhiên do trình duyệt có thể hiển thị được cửa sổ 3 chiều nên người xem có thể cảm nhận được cảm giác ba chiều.

2.5.1.2 Đưa âm thanh vào một tài liệu HTML

Cú pháp:

```
<BGSOUND
    SRC = url
    LOOP= n
>
```

Thẻ này không có thẻ kết thúc tương ứng (</BGSOUND>). Để chơi lặp lại vô hạn cần chỉ định **LOOP** = -1 hoặc **LOOP** = *INFINITE*. Thẻ **BGSOUND** phải được đặt trong phần mở đầu (tức là nằm trong cặp thẻ **HEAD**).

2.5.1.3 Chèn một hình ảnh, một đoạn video vào tài liệu HTML

Để chèn một file ảnh (.jpg, .gif, .bmp) hoặc video (.mpg, .avi) vào tài liệu HTML, bạn có thể sử dụng thẻ **IMG**.

Cú pháp:

```
<IMG
    ALIGN      = TOP/MIDDLE/BOTTOM
    ALT        = text
    BORDER     = n
    SRC        = url
    WIDTH      = width
    HEIGHT     = height
    HSPACE     = vspace
    VSPACE     = hspace
    TITLE      = title
    DYN SRC     = url
    START      = ILEOPEN/MOUSEOVER
    LOOP       = n
>
```

Trong đó:

ALIGN = TOP/ MIDDLE/ BOTTOM/ LEFT/ RIGHT	Căn hàng văn bản bao quanh ảnh
ALT = text	Chỉ định văn bản sẽ được hiển thị nếu chức năng show picture của browser bị tắt đi hay hiển thị thay

	thể cho ảnh trên những trình duyệt không có khả năng hiển thị đồ họa. Văn bản này còn được gọi là nhãn của ảnh. Đối với trình duyệt có khả năng hỗ trợ đồ họa, dòng văn bản này sẽ hiện lên khi di chuột qua ảnh hay được hiển thị trong vùng của ảnh nếu ảnh chưa được tải về hết. Chú ý phải đặt văn bản trong hai dấu nháy kép nếu trong văn bản chứa dấu cách hay các ký tự đặc biệt - trong trường hợp ngược lại có thể bỏ dấu nháy kép.
BORDER = n	Đặt kích thước đường viền được vẽ quanh ảnh (tính theo pixel).
SRC = url	Địa chỉ của file ảnh cần chèn vào tài liệu.
WIDTH/HEIGHT	Chỉ định kích thước của ảnh được hiển thị.
HSPACE/VSPACE	Chỉ định khoảng trống xung quanh hình ảnh (tính theo pixel) theo bốn phía trên, dưới, trái, phải.
TITLE = title	Văn bản sẽ hiển thị khi con chuột trỏ trên ảnh
DYNSRC = url	Địa chỉ của file video.
START = FILEOPEN/MOUSEOVER	Chỉ định file video sẽ được chơi khi tài liệu được mở hay khi trỏ con chuột vào nó. Có thể kết hợp cả hai giá trị này nhưng phải phân cách chúng bởi dấu phẩy.
LOOP = n/INFINITE	Chỉ định số lần chơi. Nếu LOOP = INFINITE thì file video sẽ được chơi vô hạn lần.

Câu hỏi ôn tập

1. Hãy nêu lịch sử của WWW?
2. Cấu trúc của một trang HTML như thế nào?
3. Thiết kế giao diện một trang web mà bạn thích.
4. Hãy nêu cách tổ chức thông tin trong trang web.

BÀI 2. THIẾT KẾ WEB TĨNH

Giới thiệu:

Website tĩnh là gì: là trang web sử dụng hoàn toàn ngôn ngữ HTML, sau khi tải trang HTML từ máy chủ xuống, trình duyệt sẽ biên dịch mã và hiển thị nội dung trang web, người dùng hầu như không thể tương tác với trang web.

- Nội dung website ít khi cập nhật và ít nên bạn muốn tiết kiệm chi phí
- Website bạn nhỏ và bạn thuê luôn người chuyên về web để quản trị

+ Ưu điểm của website tĩnh: Tốc độ truy cập nhanh bởi nó chỉ là những file HTML Chi phí đầu tư thấp bởi bạn không phải trả tiền nhiều cho Coder, về giao diện Designer có thể thiết kế theo kiểu mới lạ.

+ Nhược điểm của website tĩnh: Khó quản lý nội dung, khó nâng cấp bảo trì...

1. Mục tiêu:

- Hiểu được các chế độ hiển thị một trang Web;
- Có khả năng đưa một File vào Web;
- Có khả năng tạo được các bảng biểu và các Frame;
- Tạo được ứng dụng bảng liên kết trang Web;
- Xây dựng được các ứng dụng Multimedia;
- Sử dụng tốt các công cụ hỗ trợ thiết kế Web;
- Thực hiện các thao tác an toàn với máy tính.

2. Nội dung bài học

2.1 Tổng quan

Website tĩnh là gì?

Là trang web sử dụng hoàn toàn ngôn ngữ HTML, sau khi tải trang HTML từ máy chủ xuống, trình duyệt sẽ biên dịch mã và hiển thị nội dung trang web, người dùng hầu như không thể tương tác với trang web.

Khi nào nên sử dụng web tĩnh?

- Nội dung website ít khi cập nhật và ít nên bạn muốn tiết kiệm chi phí
- Website bạn nhỏ và bạn thuê luôn người chuyên về web để quản trị
- Nếu bạn là doanh nghiệp muốn tự mình làm website thì bạn có thể học các kiến thức căn bản và tự làm một Web tĩnh cho mình

Ưu điểm của website tĩnh:

- Tốc độ truy cập nhanh bởi nó chỉ là những file HTML
- Chi phí đầu tư thấp bởi bạn không phải trả tiền nhiều cho Coder
- Về giao diện Designer có thể thiết kế theo kiểu mới lạ thân thiện với bộ máy tìm kiếm bởi bạn có thể đặt tên file tùy ý

Nhược điểm của website tĩnh:

- Khó quản lý nội dung.
- Khó nâng cấp bảo trì...

2.2 Đối tượng Form

2.2.1. Khái niệm về Form

Form đơn giản là các biểu mẫu nhằm giúp người dùng tương tác với hệ thống, mà cụ thể là người dùng có thể gửi các thông tin cho hệ thống hay người quản trị hệ thống thông qua Form

2.2.2. Các ứng dụng của Form

Form là một thành phần không thể thiếu đối với các website, nó có thể dùng để tạo nên rất nhiều biểu mẫu phục vụ cho việc tương tác giữa người dùng với hệ thống như:

- Form đăng nhập hệ thống
- Form đăng ký
- Form liên hệ...

2.3 Các đối tượng của Form

Các phần tử trong FORM

Các phần tử trong FORM là nơi mà người dùng nhập dữ liệu, lựa chọn dữ liệu, các phần tử dữ liệu hay dùng trong form có thể kể đến là: `<input>` `<textarea>` `<select>` `<button>` `<datalist>` `<label>` `<fieldset>` `<datalist>`

Thường thì mỗi phần tử của FORM để người dùng nhập dữ liệu thì phần tử đó cần có thuộc tính name là tên loại dữ liệu cần nhập.

Thẻ `<input>`

Thẻ input form, thẻ này có thể hiện thị giao diện nhập liệu nhiều cách tùy vào thuộc tính type. Trong form người ta cũng thường dùng thẻ label để cho biết tiêu đề của một thẻ input

Sau đây là một số **type** hay dùng:

`<input type="text">` Tạo ra một hộp để nhập dữ liệu.

Ví dụ:

```
<form>
  Họ:<br>
  <input type="text" name="firstname"><br>
  Tên:<br>
  <input type="text" name="lastname">
</form>
```

Họ:

Tên:

`<input type="password">` Tạo ra một hộp để nhập mật khẩu.

Ví dụ:

```
<form>
  Tên:<br>
  <input type="text" name="username"><br>
  Mật khẩu:<br>
  <input type="password" name="psw">
</form>
```

Tên:

Mật khẩu:

<input type="submit"> Tạo ra một nút bấm và khi bấm vào dữ liệu trong FORM chứa nó sẽ được gửi đi.

Ví dụ:

```
<form action="url">
```

Nhập số:


```
<input type="text" name="number" value="2017">
```

```
<input type="submit" value="Submit">
```

```
</form>
```

Nhập số:

<input type="reset"> Tạo ra một nút bấm và khi bấm vào dữ liệu trong FORM phục hồi về mặc định.

<input type="radio"> Tạo ra các lựa chọn để người dùng chọn một trong các tùy chọn đó.

Ví dụ:

```
<form>
```

```
<input type="radio" name="gioitinh" value="Nam" checked> Nam<br>
```

```
<input type="radio" name="gioitinh" value="Nữ"> Nữ<br>
```

```
<input type="radio" name="gioitinh" value="Khác"> Khác
```

```
</form>
```

☐ Nam

☐ Nữ

☒ Khác

<input type="checkbox"> Tạo ra các lựa chọn để người dùng chọn một hoặc nhiều trong các tùy chọn đó.

Ví dụ:

```
<form>
```

```
<input type="checkbox" name="love1" value="LovePHP"> Tôi thích PHP<br>
```

```
<input type="checkbox" name="love1" value="LoveHTML"> Tôi thích HTML
```

```
</form>
```

☐ Tôi thích PHP

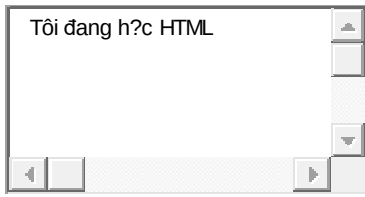
☒ Tôi thích HTML

Thẻ <textarea>

Thẻ textare tạo ra một hộp nhập dữ liệu dạng text có nhiều dòng. Có thuộc tính col độ rộng và row số dòng.

Ví dụ:

```
<textarea name="info" rows="5" cols="30">
Tôi đang học HTML
</textarea>
```



2.4 Các yếu tố động trên trang

Một số thẻ định dạng được sử dụng cho Form

Cặp thẻ <fieldset></fieldset>

- Tạo ra khung bao lấy các đối tượng trong Form

```
<html>
  <head>
    <title>Học html</title>
  </head>
  <body><fieldset>
    <form name="lienhe" action="lienhe.php" method="post"></form>
  </fieldset>
</body>
</html>
```

Cặp thẻ <legend></legend>

- Cặp thẻ <legend></legend> nằm bên trong cặp thẻ <fieldset></fieldset> và dùng để tạo tiêu đề cho Form và nằm lưng chừng ở phần khung bao
- Chú ý: Chúng ta hoàn toàn sử dụng được các thẻ HTML định dạng văn bản để định dạng cho các phần tử nằm bên trong cặp thẻ <legend></legend>

```
<html>
  <head>
    <title>Học html</title>
  </head>
  <body>
    <fieldset>
      <legend> font face="arial" color="red" size="-1"<b>Liên Hệ</b></font></legend>
      <form name="lienhe" action="lienhe.php" method="post">
        <select name="luachon"></select></form>
    </fieldset>
  </body>
</html>
```

2.5 Liên kết

Bạn có thể thấy liên kết xuất hiện trên hầu hết tất cả các website. Liên kết cho phép người dùng chuyển từ trang này tới một trang khác. Các liên kết trong HTML được gọi là những siêu liên kết, siêu liên kết là một đoạn chữ hoặc một bức ảnh mà bạn có thể nhấn vào để chuyển tới một trang khác.

Liên kết trong HTML

Trong HTML, các liên kết được tạo bởi thẻ <a>: tên liên kết

```
<!DOCTYPE html>
<html>
  <body><p><a href="https://webvn.com">Tự Học HTML</a></p></body>
</html>
```

Thuộc tính href chứa địa chỉ của liên kết (https://webvn.com)

Tên liên kết sẽ là phần sẽ được hiển thị (Tự Học HTML).

Khi người dùng nhấn vào tên liên kết họ sẽ được chuyển sang một trang mới.

Tên liên kết không chỉ bao gồm chữ. Nó có thể là một ảnh hoặc bất kỳ thành phần HTML nào khác.

Liên kết nội bộ

Liên kết ở ví dụ bên trên sử dụng đường dẫn tuyệt đối (Địa chỉ đầy đủ).

Một liên kết nội bộ (liên kết tới trang khác trên cùng một website) là một địa chỉ tương đối (không có phần http://tenwebsite...).

```
<!DOCTYPE html>
<html>
<body>
  <p><a href="/hoc-html/">Học HTML</a> là liên kết tới một trang trên cùng
website.</p>
  <p><a href="http://vi.wikipedia.org/">Wikipedia</a> là liên kết tới một website
bên ngoài.</p>
</body>
</html>
```

Liên kết trong HTML – Màu sắc và biểu tượng

Khi bạn di chuột vào một liên kết, sẽ có hai điều thường xảy ra:

- Chuột sẽ chuyển từ dạng mũi tên sang hình một bàn tay nhỏ
- Màu của liên kết sẽ thay đổi
- Mặc định, trên các trình duyệt liên kết sẽ được hiển thị như sau:
 - + Một liên kết chưa được nhấn vào sẽ được gạch chân và có màu xanh
 - + Một liên kết đã được nhấn vào sẽ được gạch chân và có màu tím
 - + Khi một liên kết được nhấn sẽ được gạch chân và có màu đỏ
- Bạn có thể thay đổi kiểu mặc định bằng cách sử dụng CSS:

```
<!DOCTYPE html>
<html>
<head>
<style>
a:link {
  color:#000000;
  background-color:transparent;
  text-decoration:none;}

```

```

a:visited {
color:#000000;
background-color:transparent;
text-decoration:none;}
a:hover {
color:#ff0000;
background-color:transparent;
text-decoration:underline;}
a:active {
color:#ff0000;
background-color:transparent;
text-decoration:underline;}
</style>
</head>
<body>
<p>Bạn có thể thay đổi màu mặc định của liên kết</p>
<a href="/css-trong-html/" target="_blank">CSS trong HTML</a>
</body>
</html>

```

Liên kết trong HTML – Thuộc tính target

Thuộc tính target quy định vị trí mở của trang được liên kết.

Trong ví dụ dưới đây liên kết sẽ được mở trong một cửa sổ hoặc một tab mới:

```

<!DOCTYPE html>
<html>
<body>
<a href="https://webvn.com/html/hoc-html/" target="_blank">Tự học
HTML</a>
<p>Nếu bạn thiết lập thuộc tính target là "_blank", liên kết sẽ được mở trong một
cửa sổ hoặc tab mới.</p>
</body>
</html>

```

Target	Miêu tả
_blank	Mở liên kết trong một cửa sổ hoặc một tab mới
_self	Mở liên kết tại chính cửa sổ hiện tại (Mặc định)
_parent	Mở liên kết tại frame ngoài
_top	Mở liên kết tại cửa sổ hiện tại (Thoát ra khỏi frame nếu đang ở trong frame).
Tên frame	Mở liên kết theo tên của frame

Nếu trang của bạn đang nằm ở trong một frame, bạn có thể dùng `target="_top"` để thoát khỏi frame:

```
<!DOCTYPE html>
```



```
<html><body>
  <p>Trang của bạn đang nằm trong frame? <a href="https://webvn.com/html/hoc-
html/" target="_top">Nhấn vào đây!</a></p>
</body>
</html>
```

Liên kết trong HTML – Sử dụng ảnh

- Cách phổ biến nhất sử dụng ảnh trong liên kết là:

```
<!DOCTYPE html>
<html>
<body>
  <p>Ảnh có chứa liên kết, bạn có thể nhấn để chuyển sang trang mới.</p>
  <a href="/html/hoc-html/">
    
  </a>
</body>
</html>
```

+ “border:0” được thêm vào để tránh trình duyệt Internet Explorer 9 (và các phiên bản trước đó) hiển thị viền xung quanh ảnh.

Liên kết trong HTML – Thuộc tính id

Thuộc tính id có thể được sử dụng để tạo một chỉ mục trong trang HTML.

Chỉ mục không hiển thị trên trình duyệt.

Ví dụ:

- Thêm thuộc tính id vào bất kỳ thẻ <a> nào:
 - + Khu vực được đánh dấu
- Sau đó tạo một liên kết tới chỉ mục (Khu vực được đánh dấu):
 - + Tới khu vực được đánh dấu
- Hoặc tạo một liên kết tới chỉ mục từ một trang khác:
 - + Tới khu vực được đánh dấu

```
<!DOCTYPE html>
<html>
<body>
  <p><a href="#block6">Xem Block 6.</a></p>
  <h2>Khu vực 1</h2>
  <p>Nội dung block</p>
  <h2>Khu vực 2</h2>
  <p>Nội dung block</p>
  <h2>Khu vực 3</h2>
  <p>Nội dung block</p>
  <h2>Khu vực 4</h2>
```

```

<p>Nội dung block</p>
<h2>Khu vực 5</h2>
<p>Nội dung block</p>
<h2><a id="block6">Khu vực 6</a></h2>
<p>Nội dung block</p>
<h2>Khu vực 7</h2>
<p>Nội dung block</p>
<h2>Khu vực 8</h2>
<p>Nội dung block</p>
<h2>Khu vực 9</h2>
<p>Nội dung block</p>
<h2>Khu vực 10</h2>
<p>Nội dung block</p>
<h2>Khu vực 11</h2>
<p>Nội dung block</p>
<h2>Khu vực 12</h2>
<p>Nội dung block</p>
<h2>Khu vực 13</h2>
<p>Nội dung block</p>
<h2>Khu vực 14</h2>
<p>Nội dung block</p>
</body>
</html>

```

2.8 Hoàn thiện trang web tĩnh

Làm thế nào để tạo trang web hoàn diện

- **Bước 1:** Lên ý tưởng cho website
- **Bước 2:** Đăng ký tên miền
- **Bước 3:** Mua Web Hosting
- **Bước 4:** Chọn một nền tảng và tiến hành tạo trang web
- **Bước 5:** Lên kế hoạch phát triển website

Như vậy, làm thế nào để tạo trang web dường như là câu hỏi đầu tiên mà bất kỳ webmaster mới nào đều đã tự hỏi. Cách thức và nền tảng cho trang web sẽ quyết định cho sự phát triển của bất kỳ dự án online nào. Vì vậy, bạn cần tạo một kế hoạch thật chắc chắn và tìm hiểu kỹ các lựa chọn trước khi bắt đầu.

Trong giáo trình này, chúng tôi sẽ hướng dẫn cách tạo trang web thông qua: **Content management systems (CMS), E-commerce scripts, Website builder**

Mỗi phương thức tạo trang web có điểm mạnh và điểm yếu riêng. Đó là lý do vì sao chúng tôi sẽ nhấn mạnh các khía cạnh quan trọng trong việc tạo trang web để giúp bạn đưa ra lựa chọn dễ dàng.

Vậy hãy bắt đầu ngay thôi, sơ lược chúng ta có 5 bước cho một trang web từ giấy trắng đến vận hành một cách chuyên nghiệp như sau:

- **Bước 1:** Lên ý tưởng cho website

- Bước 2: Đăng ký tên miền
- Bước 3: Mua hosting
- Bước 4: Chọn nền tảng và tạo website
- Bước 5: Lên kế hoạch phát triển website

Bước 1: Lên ý tưởng cho website

- Từng bước 1, bạn cần quyết định trước tiên website của bạn sẽ nói về điều gì. Có rất nhiều ý tưởng website nên bạn nên cân nhắc kỹ. Với việc lựa chọn đúng định hướng cho website, bạn sẽ còn có thể kiếm tiền từ nó nữa sau khi website đã vững mạnh.

- Hãy lựa chọn trước mục đích và tính năng chính của trang web bạn muốn làm là gì. Đây là một số loại website phổ biến nhất hiện nay:

- **Website bán hàng online.** Các cửa hàng online hiện là loại website được truy cập nhiều nhất bây giờ. Với sự phát triển của drop shipping, tất cả mọi người đều có cơ hội tạo ra một việc kinh doanh tay trái. Bạn có thể bán các sản phẩm công nghệ, các sản phẩm vật lý hay kỹ thuật số đến bất kỳ nơi đâu trên toàn cầu (sẽ đơn giản hơn rất nhiều so với một cửa hàng vật lý ngoài đời)

- **Blogs.** Rất nhiều webmaster thành công đã bắt đầu sự nghiệp của mình bằng một truy vấn tìm kiếm đơn giản: cách tạo blog. Một blog có thể viết về rất nhiều chủ đề, blogging có thể phục vụ cho việc kinh doanh, hoặc phục vụ cho sở thích của bạn (du lịch, ẩm thực, âm nhạc, sách).

- **Website giáo dục.** Bạn có kỹ năng đặc biệt đặc biệt phải không? Tại sao không tạo riêng cho mình khóa học online hoặc là website dạy chỉ cho thành viên. Nếu bạn là đam mê một chủ đề gì đó, việc tạo một trang web về giáo dục cũng giúp bạn trở thành chuyên gia trong lĩnh vực đó.

- **Website tin tức.** Trong thời đại “loạn lạc” của thế giới mạng, việc khởi tạo một trang web hoàn toàn mới rất đơn giản. Mọi người luôn tìm kiếm thông tin qua nhiều nguồn khác nhau. Nếu bạn có thể đưa tin, đăng các câu chuyện hấp dẫn, thì việc tạo trang web về tin tức sẽ giúp tin bài của bạn lan truyền nhanh hơn.

- **Cộng đồng online.** Số lượng lớn của cộng đồng online là tín hiệu tốt cho bạn nếu bạn thích tạo các site phục vụ cho cộng đồng. Vì xây dựng cộng đồng tốn rất nhiều thời gian, nên bạn bắt đầu càng sớm chừng nào càng tốt chừng ấy.

- **Trang web danh mục sản phẩm (portfolios).** Thể hiện kỹ năng của bạn bằng cách đưa ra những dự án bạn đã làm lên website, để tăng tính chuyên nghiệp và tạo ra kênh giao tiếp hữu hiệu với khách hàng tiềm năng của bạn. Hơn nữa, một trang portfolios online sẽ tạo ra lợi thế đáng kể đối với các cuộc phỏng vấn, các đợt đấu thầu, các kế hoạch kinh doanh.

- Sau khi bạn đã xác định được ý tưởng chính trong đầu, việc chọn website hosting và tên miền cho website sẽ dễ dàng hơn.

Bước 2: Đăng ký tên miền

Trước khi tạo website, bạn cần có một cái tên miền hấp dẫn trước đã. Tên miền là tên website của bạn. Cho nên hãy chọn lựa thật kỹ trước khi mua vì đó là tên mà mọi người phải nhớ để gõ vào trình duyệt. Hãy tìm tên nào hấp dẫn và hay như Google.com hay Facebook.com vậy đó. Trang web của bạn có một khởi đầu thuận lợi hay không là nhờ nó.

Để đăng ký tên miền, bạn cần thực hiện:

- Lên ý tưởng cho tên website.
- Tìm thử trên trang kiểm tra tên miền xem có khả dụng không.
- Chạy tìm kiếm tên miền
- Chọn một hoặc vài tên miền tốt nhất
- Xem lại đơn hàng và hoàn tất đăng ký

Để chọn được một tên miền bạn hài lòng chắc là hơi khó, nhưng bạn chỉ cần cân nhắc các mẹo sau:

- Ngắn: Sẽ không ai đủ kiên nhẫn gõ `Googlebutwithabiggerdomainname.com`.
- + Gắn liền với thương hiệu: Tên miền của bạn nên giúp visitor hình dung công việc kinh doanh của bạn là gì.
- + Chọn đúng miền mở rộng: sau dấu chấm (.) sẽ là .com, .net, .org. Bạn nên cân nhắc chọn .com, tuy nhiên nếu cần một tên miền độc nhất, chọn .io, .tech, .xyz cũng có thể là lựa chọn không tồi.
- + Bạn thử phát âm nó xem cảm giác như thế nào? Tên miền nên dễ phát âm, để có thể truyền miệng dễ dàng giữa các khách hàng của bạn phải không?
- + Tránh sử dụng số và dấu gạch ngang. Việc thêm các ký tự này sẽ tăng rủi ro tên miền của bạn bị gõ sai
- + Kiểm tra tên miền ngay

Bước 3: Mua Web Hosting

Cơ bản, Hosting là nơi bạn đặt toàn bộ files của website vào trong đó, để trình duyệt của khách truy cập có thể tải dữ liệu khi cần. Domain, mặt khác là một địa chỉ độc nhất bạn chọn, liên quan đến thương hiệu của bạn để khách có thể gõ domain name đó lên trình duyệt. Tóm lại, ít nhất bạn cần cả web hosting và domain name cho một trang web. Có nhiều loại web hosting, tùy vào kích thước dự án mà bạn nên cân nhắc loại web hosting nào để phù hợp.

Tại Hostinger, có 3 loại sản phẩm web hosting phù hợp với mọi loại website từ nhỏ đến lớn. Hãy xem qua các lựa chọn cơ bản này nhé:

Shared Hosting

Ý tưởng của shared hosting là host website trên cùng một server có nhiều người dùng để chia sẻ tài nguyên

Phù hợp cho các trang web cỡ nhỏ và vừa, shared hosting là loại server chứa nhiều người dùng chung. Nó giống như phương tiện công cộng như là xe bus và tàu hỏa.

Tất cả website sẽ dùng chung nguồn tài nguyên từ một server vật lý. Cách thiết lập độc đáo này giúp giảm giá thành xuống rất nhiều nhưng có giới hạn nhất định. Chủ yếu do để phù hợp cho số đông nên một người dùng trong server không được dùng quá mức tài nguyên của server mà có thể gây ảnh hưởng tới người dùng khác.

Shared hosting có nhiều công cụ để bạn tạo website độc đáo, như là:

1-click cài đặt. Dễ dàng cài đặt ngay tất cả các CMSs nổi tiếng bao gồm WordPress, Joomla!, Drupal, ...

Công cụ chia sẻ truy cập. Khi làm việc với các dự án cần nhiều người dùng, gửi thông tin đăng nhập của bạn thì không nên, cho nên chúng tôi có tính năng quan trọng này để bạn

chia sẻ và gán quyền truy cập dịch vụ cho những người khác để làm việc cùng một dự án, mà không phải gửi thông tin đăng nhập của riêng bạn!

FTP/File managers. Dễ dàng chỉnh sửa files, thư mục, code bằng file manager nền web. Hoặc sử dụng công cụ ưa thích của mọi người là FileZilla để kết nối tới web.

Backup hằng tuần. Tự động backup files và database trong control panel. Lưu tới 3 bản website!

Quản lý Cache. Kích hoạt caching sẽ giúp bạn bỏ qua việc xử lý PHP bằng cách tạo ra các trang web tĩnh. Website nhanh hơn, khách truy cập sẽ vui hơn, kết quả tốt hơn!

Đăng ký tên miền miễn phí. Nếu bạn chọn gói Cao Cấp hay Doanh Nghiệp sẽ tặng riêng cho bạn một tên miền miễn phí năm đầu.

Control panel mạnh mẽ. Nhờ vào việc thiết kế control panel (trình điều khiển riêng), còn có thể nhanh chóng tích hợp các công cụ mà khách hàng cần để quản lý hiệu quả hơn.

Loại website nào cần dùng Shared Hosting?

Thường thì, shared hosting sẽ phù hợp cho ai mới khởi đầu làm web hoặc các dự án online. Nó cũng phù hợp cho các site có lượng truy cập chưa lớn, giải pháp này là rất hoàn hảo để bắt đầu tạo trang web.

Cloud Hosting

Một giải pháp mới nhất cloud hosting kết hợp tính đơn giản của shared hosting nhưng vẫn đạt được hiệu năng cực mạnh của server ảo riêng.

Cloud hosting giống một cỗ máy 2 trong 1, sự đơn giản của shared hosting và tài nguyên riêng. Bằng công nghệ điện toán đám mây, giải pháp đạt được thời gian uptime cao nhất bằng cách lưu bản copy của website bạn trên nhiều server vật lý hoạt động đồng thời cùng với nhau. Trong trường hợp một server bị lỗi, những server khác trong cùng cluster sẽ cùng đảm nhiệm nhiệm vụ của server đó và giữ website của bạn vẫn chạy bình thường.

Bên cạnh **đặc quyền của shared hosting**, bạn còn nhận được:

Một địa chỉ IP riêng. IP được gán vào server của bạn sẽ độc nhất và chỉ dành riêng cho bạn.

Tài nguyên riêng. So với shared hosting, tài nguyên bạn nhận được khi xài gói cloud hosting sẽ dành riêng cho bạn mà không phải ai khác.

Một chứng chỉ SSL trọn đời. Hãy giữ cho dữ liệu của bạn an toàn tuyệt đối! Vì HTTPS đã thành tiêu chuẩn trên thế giới web, khi bạn xài gói cloud hosting sẽ đảm bảo bạn có được tiêu chuẩn này miễn phí cho bạn.

Tích hợp CloudFlare. Thêm một lớp bảo mật chống DDoS, WAF, monitoring tools, và những tính năng cao cấp khác.

Loại website nào cần đến cloud hosting?

Việc kinh doanh phụ thuộc nhiều vào thời gian uptime của website sẽ cần loại web hosting này nhất. Nếu bạn đang gặp trường hợp có lượng truy cập tăng cao (một dấu hiệu tốt), vậy cloud hosting là lựa chọn an toàn và hiệu quả nhất.

Bên cạnh đó, một dự án lớn như cửa hàng thương mại điện tử, mạng xã hội, hoặc các loại website cần nhiều tài nguyên, website có lượng khách truy cập lớn đều phù hợp nhất với cloud hosting.

Nếu bạn đang gặp phải vấn đề về hiệu năng mà không muốn lo lắng thiết lập hạ tầng, thì cloud hosting dĩ nhiên là lựa chọn chính xác nhất cho bạn.

VPS Hosting

Viết tắt của chữ **virtual private server (máy chủ riêng ảo)**, VPS là một sản phẩm web hosting có thể tự do điều khiển hơn. Nó đòi hỏi kỹ năng quản trị server và biết cách sử dụng terminal.

Máy chủ riêng ảo sẽ rất mạnh mẽ và linh hoạt nếu bạn có khả năng điều khiển nó bằng kiến thức lập trình của bạn

Nói ngắn gọn, khi làm website bằng VPS hosting, bạn sẽ có khả năng tùy chỉnh server cao nhất theo ý mình để tạo ra giải pháp phù hợp. Bạn có thể tùy chỉnh từ hệ điều hành, đến bộ phần mềm trong control panel, mọi chuyện đều có thể thực hiện được nhờ vào quyền truy cập root.

Địa chỉ IP riêng: Mỗi VPS nhận được một địa chỉ IP tĩnh.

1-click cài đặt OS: Sử dụng đúng hệ điều hành bạn hay dùng bằng các lựa chọn Linux distribution phổ biến có sẵn trong control panel.

Server snapshots: Files và cấu hình server của bạn sẽ được tự động backup/restore khi cần thiết.

Tài nguyên được cô lập: Every drop of power in your hands. No other user on the machine will be able to tap into your resources.

Tự do cài đặt môi trường của riêng bạn: Với quyền root trong tay, bạn có thể tùy chỉnh phần mềm theo ý riêng để đưa ra giải pháp đặc thù cho website bằng cách tùy chỉnh cấu hình ngay phía server

Loại website nào cần VPS hosting?

With shared/cloud hosting, server đã được cấu hình sẵn và có nhiều thông số không thể thay đổi (như software stack).

Tuy nhiên, VPS cho bạn quyền root access, để bạn có thể thực hiện bất kỳ tùy chỉnh nào để biến nó thành nền tảng phù hợp cho riêng bạn. Nếu bạn cần một giải pháp riêng, đặc thù, thì tốt nhất là bạn nên dùng VPS để tạo trang web.

Bước 4: Chọn một nền tảng và tiến hành tạo trang web

Giờ là lúc khó khăn đây. Cả quá trình tạo trang web sẽ phụ thuộc vào bước này – chọn nền tảng. Mục tiêu của bạn cần làm gì sẽ đóng vai trò quyết định bạn nên chọn nền tảng nào.

- Các lựa chọn bạn cần cân nhắc là:

+ **Sử dụng CMS:** Hãy cứ hình dung hệ thống quản trị nội dung (CMS) là một công cụ cao cấp để giúp bạn làm website, thiết kế trang web, và tạo nội dung cho nó. Có 3 thương hiệu CMS nổi tiếng nhất thế giới là WordPress, Drupal và Joomla! Cả 3 CMS này đều miễn phí và là mã nguồn mở, nhưng WordPress chiếm ưu thế hơn hẳn vì nó là CMS mạnh mẽ và phổ biến nhất. Cả 3 đều sử dụng PHP làm nền, và có templates/theme để bạn tự thiết kế giao diện trang web được, mà không đòi hỏi kiến thức lập trình. Cộng đồng hỗ trợ cũng lớn, mặc dù Joomla! và Drupal không thể so sánh được với WordPress về mức độ tiện dụng, sự đơn giản và sự linh hoạt. Vì vậy WordPress luôn là sự lựa chọn hàng đầu cho người mới sử dụng, và trong bài này sẽ dùng CMS WordPress làm ví dụ

để hướng dẫn bạn tạo trang web.

+ **E-commerce Scripts:** Như tên gọi của nó, nếu bạn muốn tạo một cửa hàng online, vậy bạn cần sử dụng các script thương mại điện tử, nó sẽ giúp bạn làm website bạn muốn. Một số script nổi tiếng có thể kể đến là *PrestaShop*, *Magento*, *WooCommerce*.

+ **Sử dụng Website Builder:** Vấn đề tự động hóa trong việc làm trang web đạt đến tầm cao mới khi website builders bắt đầu tồn tại. Đầu tiên, các lựa chọn tùy biến và tính năng còn giới hạn, thiết kế cũng bị đóng khung trong các lựa chọn, tuy nhiên, sau vài năm website builder đã tìm được chỗ đứng vững chắc trong hệ thống website hiện đại. Mặc dù so với CMS, website builder không thể tùy biến nhiều được, nhưng bù lại nó rất nhanh, hiệu quả, và dường như là công cụ hoàn hảo cho những ai vừa mới bắt đầu học tạo trang web, nhưng lại không muốn mất quá nhiều thời gian để tìm hiểu. Đặc biệt website builder nổi tiếng với khả năng kéo thả có thể giúp bạn chỉ mất 5 phút để hoàn thành trang web của mình.

+ **Viết code từ con số 0:** Lựa chọn thứ 3 là nếu bạn đã biết viết code, bạn cũng có thể bắt đầu xây dựng từ con số 0. Mặc dù nó sẽ tốn nhiều thời gian và công sức hơn, nhưng bạn có thể triển khai thêm tính năng quan trọng mà bạn muốn.

Câu hỏi ôn tập

1. Hãy nêu các chế độ hiển thị một trang Web?
2. Có thể đưa một File vào Web không?
3. Tạo các bảng biểu và các Frame;
4. Tạo ứng dụng bảng liên kết trang Web;
5. Xây dựng ứng dụng Multimedia;
6. Sử dụng các công cụ nào để hỗ trợ thiết kế Web;

BÀI 3. XÂY DỰNG WEB ĐỘNG

Giới thiệu:

Website động là gì: là một tập hợp các dữ liệu số hóa được tổ chức thành cơ sở dữ liệu, các dữ liệu số hóa được gọi ra trình diễn trên các trang web dưới dạng văn bản, âm thanh, hình ảnh. nó có thêm các phần xử lý thông tin và truy xuất dữ liệu còn website tĩnh thì không. Khác với web tĩnh, web động luôn luôn có thông tin mới do các thông tin này được cập nhật bởi phần mềm quản trị web do các công ty thiết kế website cung cấp. Các thông tin mới này được lưu vào cơ sở dữ liệu của website và đưa ra sử dụng dựa theo yêu cầu của người dùng.

Trang web động được các chuyên gia lập trình, sử dụng các ngôn ngữ lập trình tạo ra mã nguồn dựa theo yêu cầu của trang web.

1. Mục tiêu:

- Cài đặt, cấu hình được IIS và ngôn ngữ lập trình Web;
- Sử dụng tốt công cụ ADO.Net;
- Sử dụng tốt các đối tượng, tiện ích của ngôn ngữ lập trình;
- Có khả năng kết nối với cơ sở dữ liệu;
- Hoàn thiện Website động
- Thiết kế giao diện phía người dùng và phía quản trị Web
- Thực hiện các thao tác an toàn với máy tính.

2. Nội dung bài học

2.1 Tổng quan về ASP.Net và ADO.Net

2.1.1. Tổng quan về lập trình ứng dụng Web ASP.Net

Quá trình Request - Response



Hình 3.1 Quá trình Request - Response

Web Server – Web Browser – HTTP

- Web Server nhận và điều phối các yêu cầu từ Web Browser và gửi kết quả
- Web Browser thể hiện dữ liệu, tập hợp dữ liệu của người dùng và gửi đến Web Server
- HTTP dùng để trao đổi thông tin giữa Web Browser với Web Server



Hình 3.2 Tổng quan về ASP.Net

- Mô hình ứng dụng

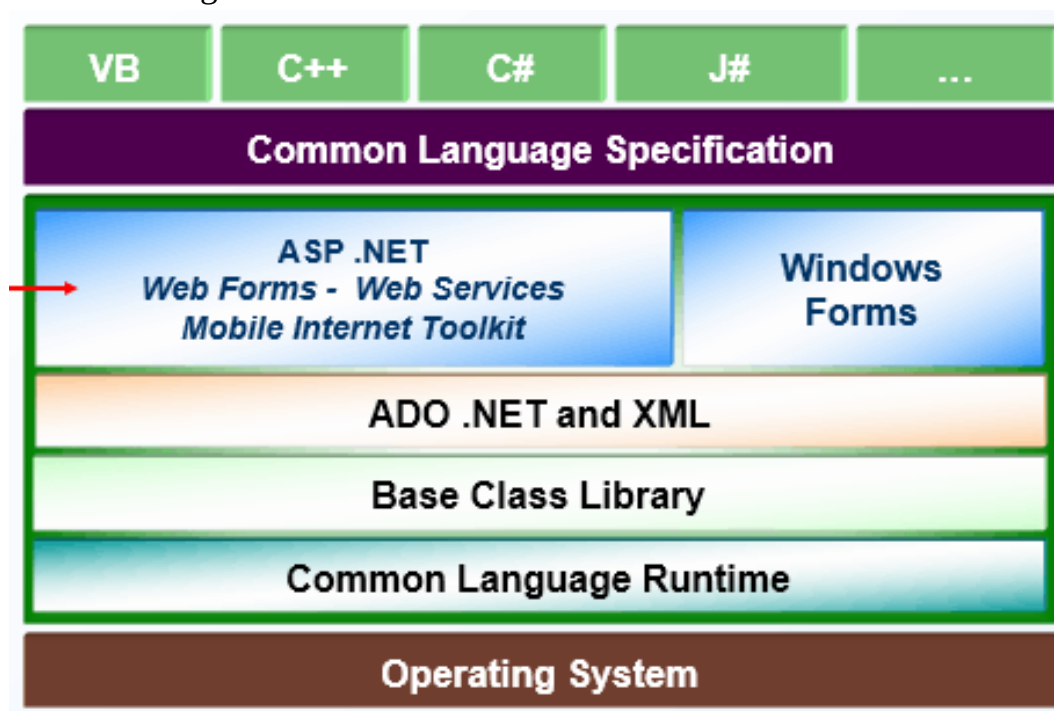


Hình 3.2 Mô hình ứng dụng

2.1.1.2. Giới thiệu về ASP.NET

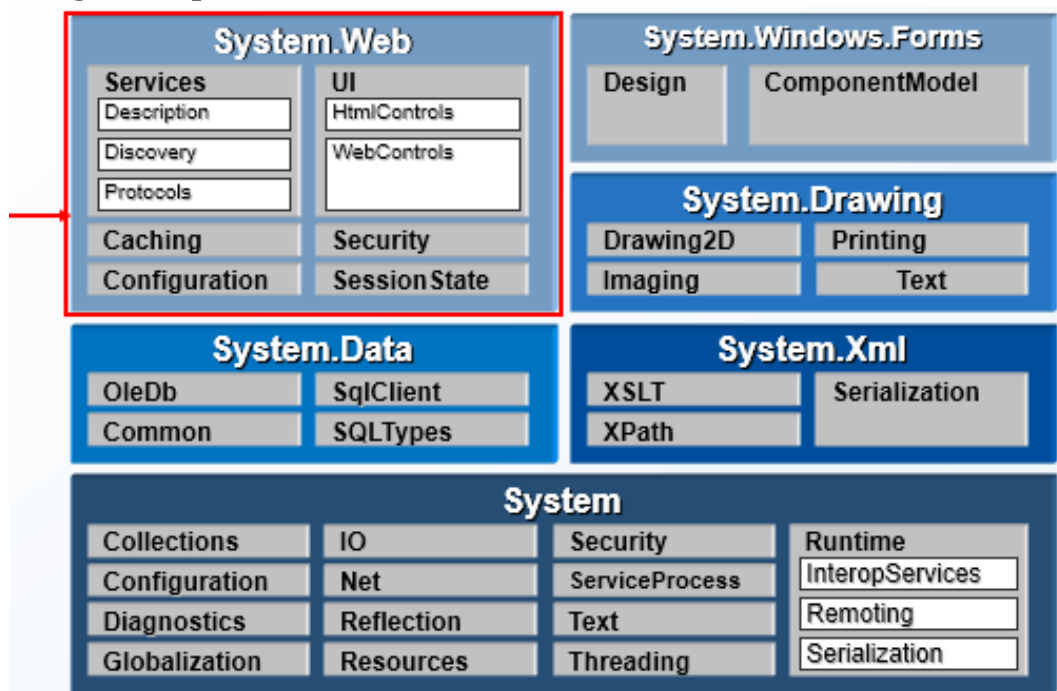
Tìm hiểu về ASP.Net

- ASP.NET trong .Net Framework



Hình 3.3 ASP.NET trong .Net Framework

Hệ thống namespace



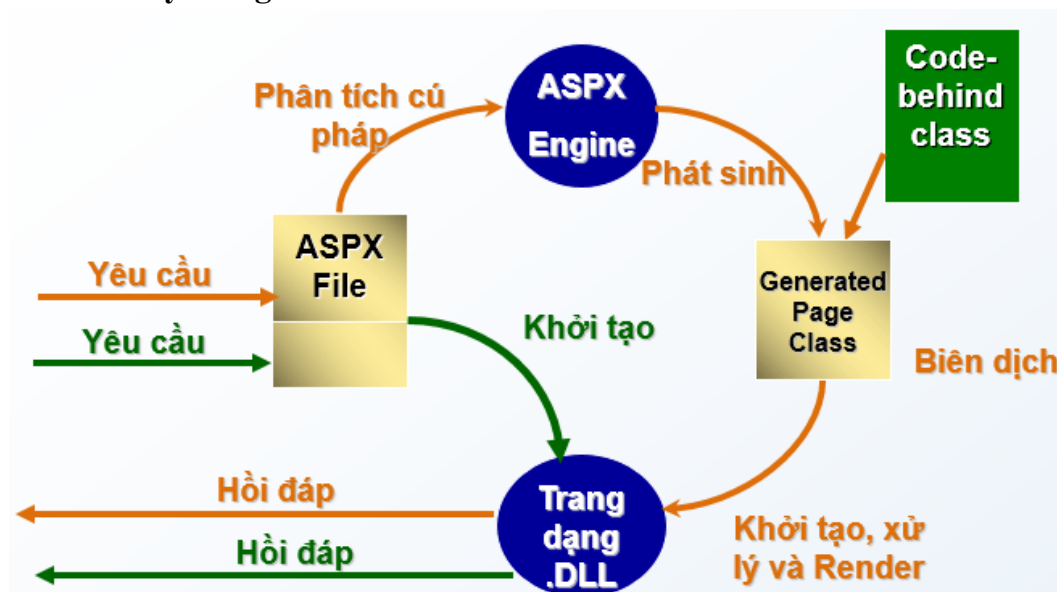
Hình 3.4 Hệ thống namespace

ASP.Net là kỹ thuật lập trình và phát triển ứng dụng web ở phía Server trên môi trường Visual Studio .Net

Một số các đặc điểm của ASP.NET

- Trang Asp.Net được biên dịch trước, thành tập tin DLL mà Server có thể thi hành
- Tự động phát sinh mã HTML cho các Server control tương ứng với từng loại Browser
- Hỗ trợ các ngôn ngữ khác nhau trong cùng một ứng dụng ...

Quá trình xử lý trang ASPX



Hình 3.5 Quá trình xử lý trang ASPX

2.1.1.3. Tạo ứng dụng web đầu tiên

- Khởi động MS Visual Studio .Net
- Tạo mới một ứng dụng web

- Tạo một ứng dụng nhỏ với một câu chào
- Các kiểu Web site (Web Site Types):
 - + **File System**: không cần cài đặt IIS, VS 2010 sẽ cung cấp web server riêng khi ta thi hành ứng dụng
 - + **Local HTTP**: phải có IIS và khai báo URL đã được ánh xạ
 - + **FTP**: sử dụng giao thức truyền file để quản lý các tập tin trong web site
- Lưu trang web với font Unicode, trang web có phần mở rộng: aspx (file giao diện) và aspx.cs (file code-behind)
- Xác định trang khởi động
- Thi hành ứng dụng
- Làm quen với các thành phần giao diện
- Minh họa trên môi trường VS.Net 2010

2.1.1.4. Tổ chức lưu trữ ứng dụng web

Tổng quát

- ASP.Net cung cấp một số tên tập tin và thư mục dùng để phục vụ việc tổ chức lưu trữ một ứng dụng, trong đó:
 - + **Default page**: tự động thi hành khi người dùng nhập tên miền
 - + **Application folders**: dùng để lưu trữ các tập tin có phần mở rộng tương ứng với ý nghĩa của thư mục lưu trữ
 - + **Giới thiệu một số thư mục thường dùng**
 - + **App_Data**: lưu trữ tập tin dữ liệu như .mdb, .mdf, .xml, ...
 - + **App_Code**: lưu trữ các tập tin chứa source code, có phần mở rộng .vb, .cs, ...
 - + **App_Themes**: lưu trữ các tập tin dùng để tạo giao diện chung theo chủ đề, thường có phần mở rộng .skin, .css hoặc các file hình ảnh
 - + **Thao tác tạo các thư mục**: Trên thực đơn: chọn Website => Add ASP.NET Folder => chọn thư mục cần tạo

2.1.1.5. Làm việc với tập tin cấu hình

Tập tin cấu hình – Configuration files

Dùng để khai báo các thông số mặc định cho ứng dụng, có dạng XML

Tập tin cấu hình đầu tiên là Machine.config, được đặt trong thư mục Configuration của máy

Machine.config cấu hình cho tất cả các ứng dụng .NET như Windows, Console, Web Application, ...

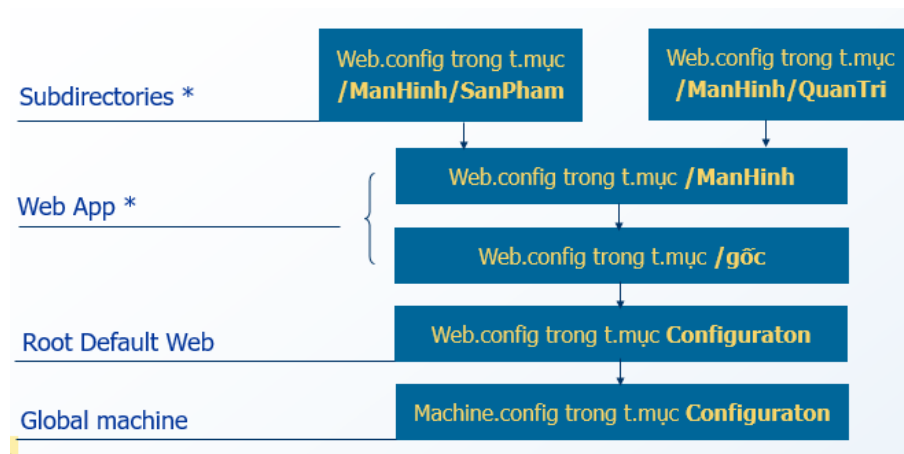
Web.config gốc chứa các khai báo mặc định của Web Server, nằm trong cùng thư mục với Machine.config

Web.config có thể ghi đè lên một số khai báo trong Machine.config

Trong mỗi ứng dụng và các thư mục con của ứng dụng Web có thể có tập tin Web.config riêng

Xử lý tập tin cấu hình

- Khi ứng dụng Web được thi hành, sẽ cache các khai báo theo trình tự như sau:



Hình 3.6 Xử lý tập tin cấu hình

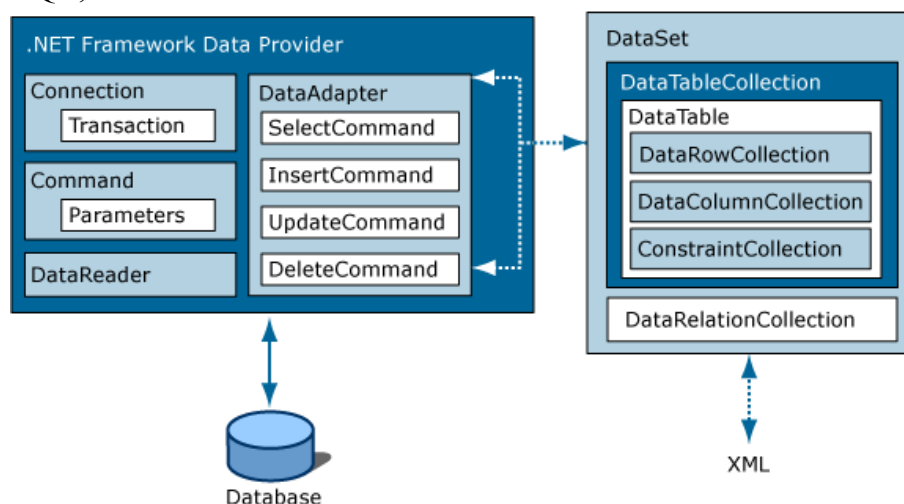
- Ghi chú: * là tùy chọn
- Trình tự xử lý:
 - + Các khai báo trong Machine.config được truy xuất trước
 - + Các khai báo trong Web.config được đưa vào caches, viết đè lên các khai báo trước đó của Machine.config
 - + Nếu có tập tin Web.config trong thư mục gốc của Web site, tập tin này sẽ được đọc vào cache, viết đè lên các khai báo trước đó
 - + Nếu có tập tin Web.config trong thư mục con của ứng dụng Web, tập tin này sẽ được đọc vào cache, viết đè lên các khai báo trước đó. Thông thường tập tin này chỉ cấu hình cho các tập tin và thư mục con của thư mục con này. (Chi tiết sẽ được minh họa trong các bài 6 và 7)

Cập nhật tập tin cấu hình

- Mở tập tin và cập nhật trực tiếp
- Hoặc dùng Web Site Administrator Tool do VS 2010 cung cấp, thao tác: Trên thực đơn chọn: Website => ASP.NET Configuration

2.2.1 Tổng quan về ADO.NET

ADO.NET là một thư viện trong .Net framework của Microsoft viết sẵn để hỗ trợ người dùng xử lý trên các hệ quản trị cơ sở dữ liệu quan hệ như Microsoft SQL Server, Oracle, My SQL, Microsoft Accesss.....



Hình 3.7 Tổng quan về ADO.NET

- ADO.NET gồm 2 thành phần chính ADO.NET :
 - + Các .NET Framework data providers cung cấp đối tượng hỗ trợ cho việc mở kết nối tới các cơ sở dữ liệu, các đối tượng hỗ trợ thực thi các câu lệnh cơ bản thực thi trên dữ liệu của SQL như select, insert, update và delete.
 - + DataSet và các lớp liên quan tới nó như DataRow, DataTable, DataView dùng cho việc lưu trữ dữ liệu của cơ sở dữ liệu sau khi thực thi các câu lệnh của .NET Framework data providers và các dữ liệu này của DataSet thông thường được lưu trữ trên RAM của bộ nhớ.
 - + Các thành phần và các lớp của ADO.NET chủ yếu được viết trong không gian tên (namespace) **System.Data** của .Net framework, trong đó DataSet nằm trong System.Data, và .NET Framework data providers các thường nằm trong các **không gian tên (namespace) con** của **System.Data** theo loại hệ quản trị dữ liệu mà chúng ta phát triển.

2.2.1.1 .NET Framework data providers

Khi phát triển ứng dụng sử dụng ADO.NET để kết nối tới cơ sở dữ liệu, các bạn có thể sử dụng .NET Framework data providers được viết sẵn trong .Net framework do Microsoft phát triển hoặc sử dụng các **data providers** do cộng đồng hay do chính các đơn vị phát triển hệ quản trị cơ sở dữ liệu viết ra.

Trong .Net framework, Microsoft đã cung cấp sẵn cho các bạn các .NET Framework data providers như sau:

Data provider	Namsepace	Mô tả
.NET Framework Data Provider dành cho Microsoft SQL Server	System.Data.SqlClient	Cung cấp các kết nối và thao tác tới các phiên bản Microsoft SQL Server từ 7.0 trở lên
.NET Framework Data Provider dành cho các OLE DB	System.Data.OleDb	Cung cấp các kết nối và thao tác tới các hệ quản trị dữ liệu dựa trên OLE DB.
.NET Framework Data Provider dành cho ODBC	System.Data.Odbc	Cung cấp các kết nối và thao tác tới các hệ quản trị dữ liệu dựa trên ODBC.
.NET Framework Data Provider dành cho Oracle	System.Data.OracleClient	Cung cấp các kết nối và thao tác tới các phiên bản Oracle client software từ 8.1.7 trở lên.
EntityClient Provider	System.Data.EntityClient	Cung cấp các kết nối và thao tác dành cho các ứng dụng Entity Data Model (EDM)

Chú ý: Đối với các hệ quản trị cơ sở dữ liệu quan hệ khác như Oracle, My SQL,

PostgreSQL, ...các bạn có thể sử dụng các Data Provider do chính đơn vị cung cấp hệ quản trị dữ liệu hoặc công đồng phát triển để có được sử tối ưu và phù hợp hơn so với các bản chuẩn mà Microsoft cung cấp. Ví dụ như Data Provider dành cho Oracle là System.Data.OracleClient cũng được chính bản thân Microsoft khuyến cáo không nên sử dụng cho các ứng dụng thực tế, các bạn nên sử dụng Oracle Data Provider for .NET do chính Oracle phát triển.

Trong mỗi NET Framework data providers đều có nhiều lớp với các thao tác nghiệp vụ xử lý khác nhau, nhưng cơ bản chúng cần phải cung cấp lớp cơ bản như sau:

Lớp	Miêu tả
Connection	Cung cấp các đối tượng dùng để thiết lập một kết nối tới cơ sở dữ liệu dựa trên một địa chỉ cơ sở dữ liệu cụ thể. Lớp Connection kế thừa từ lớp trừu tượng DbConnection. Đầu vào của mỗi Connection là một chuỗi kết nối, chứa thông tin tới địa chỉ của cơ sở dữ liệu nằm trong không gian tên System.Data.Common.
Command	Cung cấp đối tượng hỗ trợ thực thi các câu lệnh liên quan tới việc xử lý và trích xuất dữ liệu từ một cơ sở dữ liệu. Lớp Command tồn tại trong một phiên kết nối tới đối tượng Connection. Lớp Command kế thừa từ lớp trừu tượng DbCommand nằm trong không gian tên System.Data.Common. Đầu vào của lớp Command là một kết nối tới cơ sở dữ liệu (Connection) và các tham số (Parameter)
DataReader	Cung cấp đối tượng tạo ra một luồng hỗ trợ chỉ đọc forward-only, read-only dữ liệu từ một cơ sở dữ liệu dưới dạng thực thi câu lệnh Select của SQL. Lớp DataReader kế thừa từ lớp trừu tượng DbDataReader nằm trong không gian tên System.Data.Common. Đầu vào là một đối tượng Command.
DataAdapter	Cung cấp đối tượng hỗ trợ nhận các dữ liệu trả về của các câu lệnh Insert, Update, Delete của SQL, và nhận dữ liệu trả về dưới dạng DataSet đối với các câu lệnh Select của SQL. Lớp DataAdapter kế thừa từ lớp trừu tượng DbDataAdapter nằm trong không gian tên System.Data.Common. Đầu vào là một đối tượng Command.

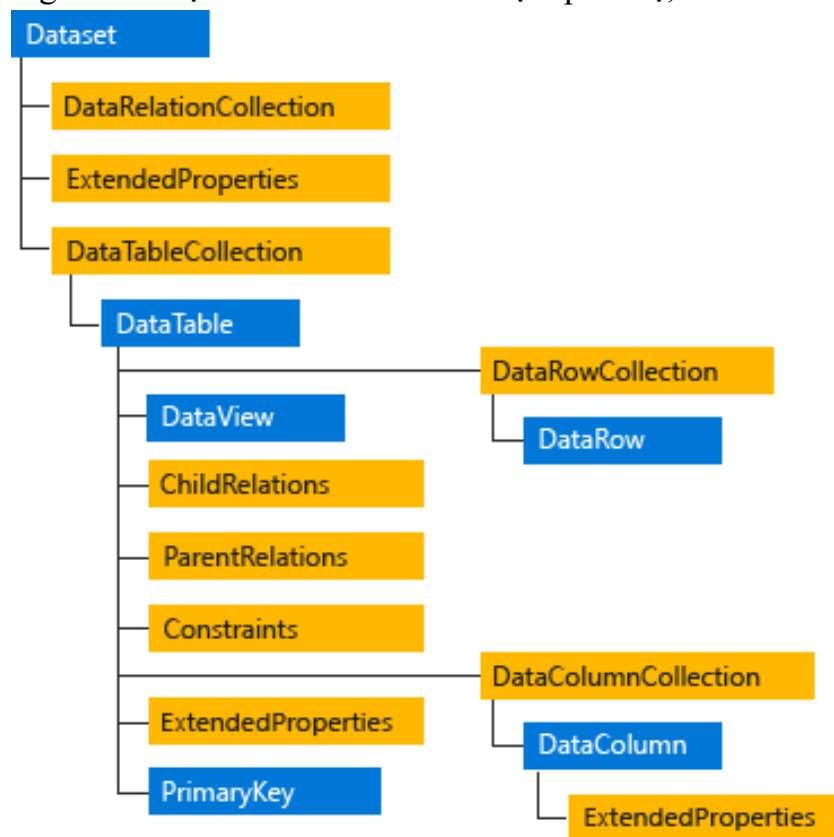
Các ví dụ sử dụng .NET Framework data providers

- Sử dụng System.Data.SqlClient trong ADO.NET để kết nối tới cơ sở dữ liệu Microsoft SQL Server
- Sử dụng System.Data.OracleClient trong ADO.NET để kết nối tới cơ sở dữ liệu Oracle
- Sử dụng Oracle Data Provider for .NET của Oracle để kết nối tới cơ sở dữ liệu Oracle

2.2.1.2 DataSet

DataSet là một mô hình tổ chức dữ liệu có cấu trúc tương đối giống mô hình của cơ sở dữ liệu quan hệ, dùng để lưu trữ dữ liệu trên bộ nhớ máy tính độc lập với tất cả các nguồn dữ liệu. DataSet gồm 1 tập của các DataTable, mà mỗi DataTable có mô hình như

một bảng dữ liệu trong hệ cơ sở dữ liệu quan hệ, điều này giúp cho DataSet có thể lưu trữ được nhiều nguồn dữ liệu khác nhau như dữ liệu quan hệ, XML



Hình 3.8 mô hình tổ chức dữ liệu

Dưới đây là các lớp cơ bản trong ADO.NET mà chúng ta thường sử dụng trong không gian tên System.Data.

STT	Tên lớp	Mô tả
1	DataSet	Cung cấp một đối tượng lưu trữ dữ liệu trong bộ nhớ. DataSet là một tập hợp nhiều DataTable và DataRelation
2	DataTable	Cung cấp một đối tượng bảng trong bộ nhớ (table of in-memory). Một bảng gồm nhiều DataRow và DataColumn
3	DataRelation	Cung cấp một đối tượng miêu tả mối quan hệ cha con (parent/child relationship) giữa 2 DataTable
4	DataViews	Cung cấp một đối tượng miêu tả các khung nhìn khác nhau của dữ liệu DataTable đã lưu trong bộ nhớ. Có thể hiểu DataTable giống như câu lệnh select lấy tất cả cột và dòng dữ liệu của bảng. Còn DataViews là hình thức select có lựa chọn cột cần lấy và mệnh đề where để tùy chỉnh số lượng dòng cần hiển thị, và tùy vào yêu cầu nghiệp vụ của ứng dụng mà số lượng cột và số dòng lấy ra phù hợp với giao diện của ứng dụng

STT	Tên lớp	Mô tả
5	DataColumn	Cung cấp đối tượng minh họa một giản đồ một cột của đối tượng DataTable. Dùng DataColumn để định nghĩa kiến trúc của DataTable
6	DataColumnCollection	Cung cấp đối tượng minh họa một tập các cột DataColumn trong đối tượng DataTable
7	DataRow	Cung cấp đối tượng minh họa một dòng dữ liệu của đối tượng DataTable. Dùng DataRow để lưu trữ các dòng dữ liệu thực tế của DataTable
8	DataRowCollection	Cung cấp đối tượng minh họa một tập các cột DataRow trong đối tượng DataTable
9	DataTableReader	Cung cấp một đối tượng hỗ trợ việc đọc một hoặc nhiều DataTable theo nguyên tắc chỉ đọc, chỉ chuyển tiếp (read-only, forward-only)

Các ví dụ sử dụng DataSet và các thành phần trong ADO.NET

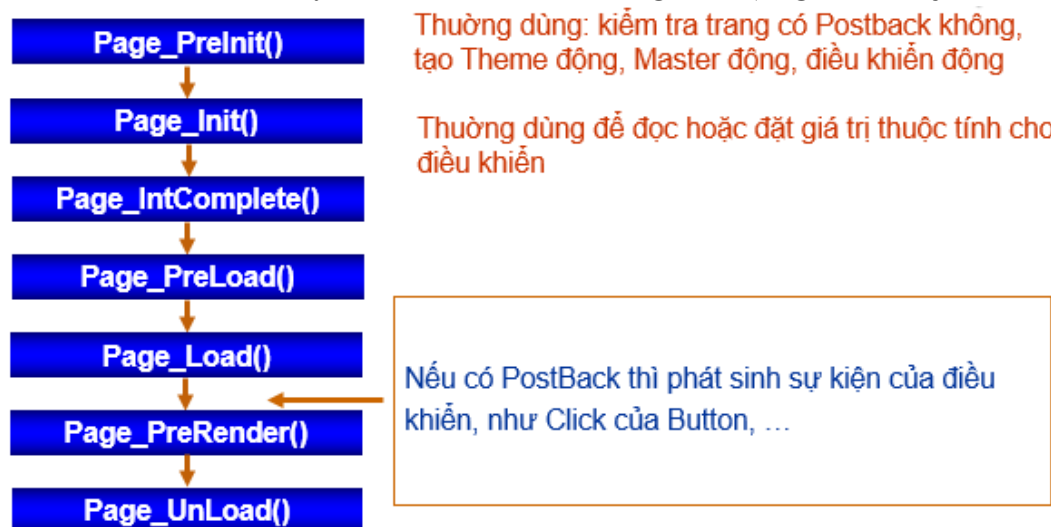
- Tạo một DataTable trong bộ nhớ bằng code (Programmatically)
- Tạo một DataTable trong bộ nhớ bằng cách sử dụng System.Data.SqlClient trong Microsoft SQL
- Tạo một DataSet trong bộ nhớ bằng code (Programmatically)
- Tạo một DataSet trong bộ nhớ bằng cách sử dụng System.Data.SqlClient trong Microsoft SQL

2.2 Các đối tượng ASP.Net

2.2.1 Đối tượng Asp.Net Page

Sự kiện

- Trình tự các sự kiện xảy ra khi render một trang web (Page Life-Cycle Events)



Hình 3.9 đối tượng Page

- Thuộc tính
 - + IsPostBack (True/False): trả về trạng thái của trang web, cho biết được load lần

đầu tiên hay không

- + MaintainScrollPositionOnPostBack (True/False): có giữ nguyên vị trí mà người dùng đang đọc sau khi PostBack không

2.2.2. HTML Server Controls

HTML Controls

- Được tạo ra từ tag HTML tĩnh
- Thường được sử dụng lập trình ở phía client
- Thuộc tên miền System.Web.UI.HtmlControls

HTML Server Controls

- Sử dụng HTML Controls trong quá trình lập trình ở phía server, ta thực hiện như sau:

- + Bổ sung thuộc tính runat="server" vào tag điều khiển
- + Hoặc chọn chức năng Run As Server Control từ thực đơn ngữ cảnh
- + Điều khiển có thuộc tính runat="server" gọi là HTML Server Controls

2.2.3 Web Server Controls

Web Server Controls là các đối tượng của .Net Framework

Được chuyển đổi sang dạng HTML tĩnh lúc thực thi

Thường được sử dụng lập trình ở phía server Thuộc tên miền System.Web.UI.WebControls

HTML Source lúc thiết kế có dạng: <asp:textbox attributes runat="server" />

- Label
 - + Dùng để hiển thị thông tin
 - + Thuộc tính Text có thể nhận và hiển thị nội dung với các tag HTML.
- HyperLink
 - + Dùng để tạo ra các liên kết siêu văn bản
 - + ImageURL: hình ảnh hiển thị trên điều khiển (ưu tiên)
 - + Text: chuỗi văn bản hiển thị trên điều khiển.
 - + NavigateUrl: đường dẫn cần liên kết đến
 - + Target(_blank, _self, _parent): dạng cửa sổ hiển thị trang kết quả
- TextBox
 - + TextBox Dùng để nhập và hiển thị dữ liệu
 - + Text: nội dung chứa trong
 - + TextBox TextMode (SingleLine, MultiLine, Password): định dạng hiển thị của Textbox.
 - + Wrap: có được phép tự động xuống dòng hay không
- Image
 - + Dùng để hiển thị hình ảnh
 - + ImageURL: đường dẫn đến tập tin hình ảnh cần hiển thị.
 - + AlternateText: chuỗi văn bản sẽ hiển thị khi đường dẫn của thuộc tính ImageURL không tồn tại.
- Button, ImageButton, LinkButton
 - + Sẽ PostBack về server khi được chọn

- + CausesValidation(True/False): có kiểm tra tính hợp lệ trên trang web không
- + OnClientClick: ngôn ngữ Client-side script sẽ được thi hành ở mức client
- CheckBox, RadioButton
 - + CheckBox và RadioButton Checked: cho biết trạng thái của điều khiển có được chọn hay không
 - + TextAlign: qui định vị trí hiển thị của điều khiển so với chuỗi văn bản
 - + GroupName: nhóm các điều khiển thành một nhóm, dành cho RadioButton

2.2.4. Các điều khiển khác

Panel và Placeholder

- Được sử dụng để chứa các điều khiển khác
- Điều khiển Panel cho phép đặt những điều khiển khác vào bên trong lúc thiết kế còn Placeholder thì không
 - Xử lý thêm điều khiển khác vào lúc thực thi: `TextBox txtSo_A = new TextBox();`
`<Điều khiển Panel | Placeholder>.Controls.Add(txtSo_A);`

Table

- Table Dùng để thao tác, thể hiện dữ liệu dưới dạng bảng
- Bao gồm tập hợp các dòng - TableRow (thông qua thuộc tính Rows), mỗi dòng được tạo thành từ tập hợp các ô - TableCell (thông qua thuộc tính Cells)
- Mỗi ô trong Table có thể chứa các điều khiển khác

Calendar

- Dùng để thể hiện lịch trong tháng
- Sự kiện:
 - + SelectionChanged: khi thay đổi ngày khác
 - + VisibleMonthChanged: khi thay đổi tháng khác
 - + DayRender: khi các ngày được tạo ra

FileUpload

- Dùng để upload tập tin về server
- Thuộc tính:
 - + HashFile (True/False): điều khiển có khai báo tập tin chưa
 - + FileName: trả về tên tập tin được upload
 - + Phương thức: SaveAs: upload tập tin lên Server
- View, MultiView
 - + MultiView và View Là control container, dùng để chứa các điều khiển khác
 - + MultiView gồm có nhiều View
 - + Thuộc tính: ActiveViewIndex: chỉ số của view hiện hành
 - + Phương thức: SetActiveView: thay đổi view hiện hành

2.2.5 Đối tượng ViewState

Dùng để lưu lại những thông tin của trang web sau khi Web Server gửi kết quả về cho client => Quản lý tình trạng ở mức client

Thuộc tính EnableViewState của trang web cho phép sử dụng đối tượng này hay không (True/False)

Sử dụng:

Gán giá trị cho ViewState: ViewState["Tên trạng thái"] = <Giá trị>;

Nhận giá trị: <Biến> = ViewState["Tên trạng thái"];

Lưu ý: "Tên trạng thái" có phân biệt chữ HOA/thường

2.3 Các đối tượng ADO.Net

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

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

Mục tiêu chính của ADO.NET là:

Cung cấp các lớp để thao tác CSDL trong cả hai môi trường là phi kết nối (Disconnected data) và kết nối (Connected data).

Tích hợp chặt chẽ với XML (Extensible Markup Language)

Tương tác với nhiều nguồn dữ liệu thông qua mô tả dữ liệu chung.

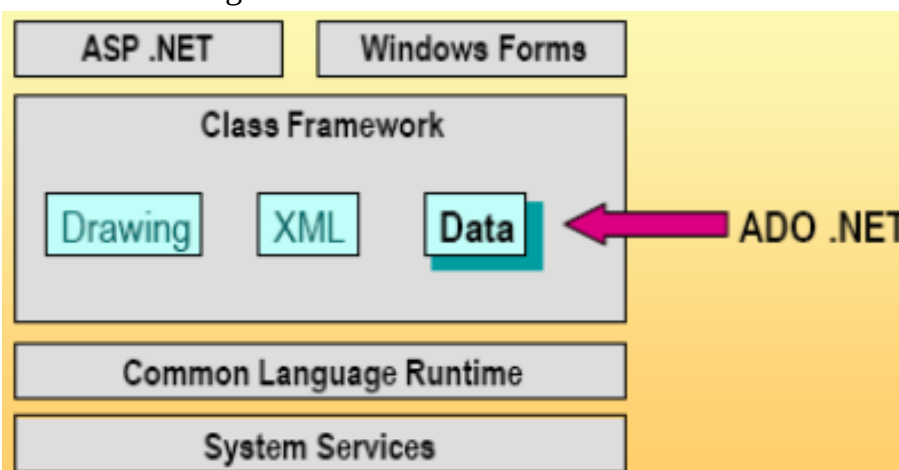
Tối ưu truy cập nguồn dữ liệu (OLE DB & SQL server).

Làm việc trên môi trường Internet.

Các lớp của ADO.NET được đặt trong Namespace là System.Data/ System.Data.oledb

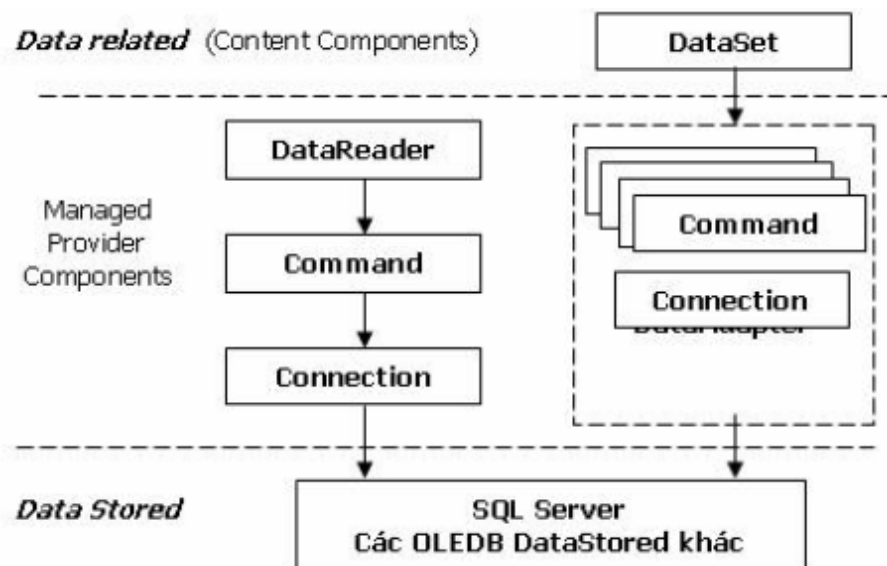
ADO.NET bao gồm 2 Provider (2 bộ thư viện thường dùng) để thao tác với các CSDL là: OLE DB Provider (nằm trong System.Data.OLEDB) dùng để truy xuất đến **bất kỳ CSDL nào có hỗ trợ OLEDB**; SQL Provider (nằm trong System.Data.SqlClient) chuyên dùng để truy xuất đến CSDL SQL Server (Không qua OLE DB nên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

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



Hình 3.10 Vị trí của ADO.NET trong kiến trúc của .net Framework

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



Hình 3.11 biểu diễn kiến trúc ADO.NET

Kiến trúc ADO.NET có thể chia làm 2 phần chính:

Managed Provider Component: bao gồm các đối tượng như DataAdapter, DataReader,... giữ nhiệm vụ làm việc trực tiếp với dữ liệu như database, file,...

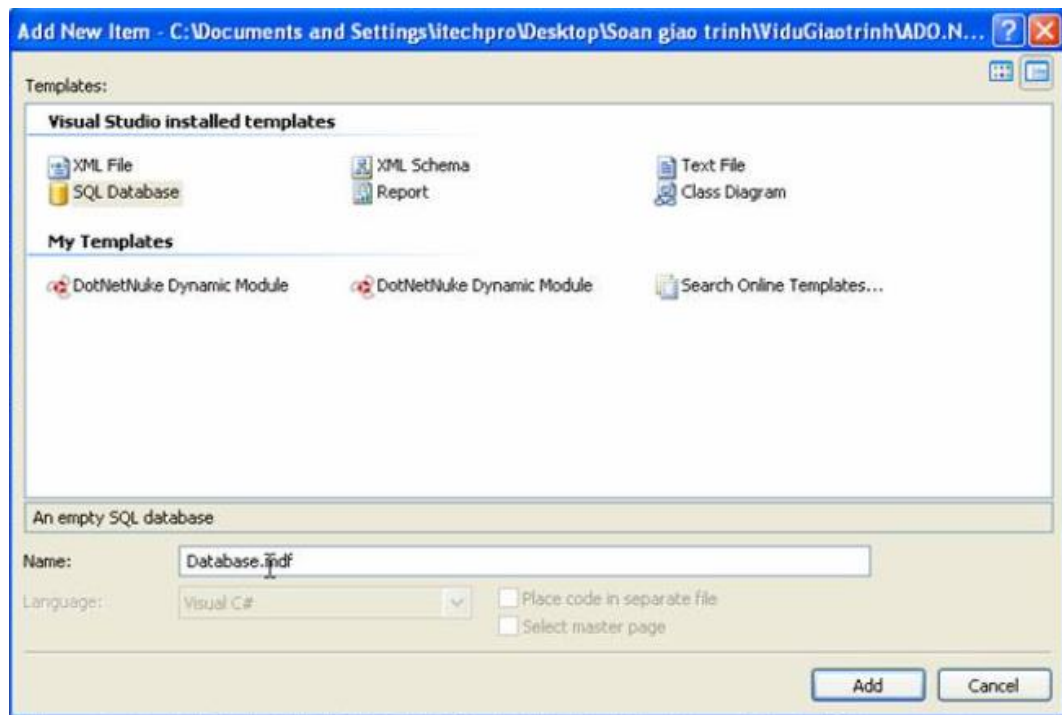
- Content Component: bao gồm các đối tượng như DataSet, DataTable,... đại diện cho dữ liệu thực sự cần làm việc. DataReader là đối tượng mới, giúp truy cập dữ liệu nhanh chóng nhưng forward-only và read-only giống như ADO RecordSet sử dụng Server cursor, OpenForwardOnly và LockReadOnly.

- DataSet cũng là một đối tượng mới, không chỉ là dữ liệu, DataSet có thể coi là một bản sao gọn nhẹ của CSDL trong bộ nhớ với nhiều bảng và các mối quan hệ. DataAdapter là đối tượng kết nối giữa DataSet và CSDL, nó bao gồm 2 đối tượng Connection và Command để cung cấp dữ liệu cho DataSet cũng như cập nhật dữ liệu từ DataSet xuống CSDL.

Trước khi đi vào học cụ thể các đối tượng của ADO.NET chúng ta cùng xem qua một ví dụ HelloWorld với ADO.NET qua đó bạn sẽ thấy được công việc cần thực hiện khi thao tác với database (ở ví dụ này mình dùng với SQLExpress).

Để làm ví dụ này bạn thực hiện theo các bước sau:

- **Bước 1:** Nhấn chuột phải vào thư mục App_Data chọn new Item.
 - + Cửa sổ Add New Item hiện ra bạn chọn SqlDatabase như hình 3.12 sau



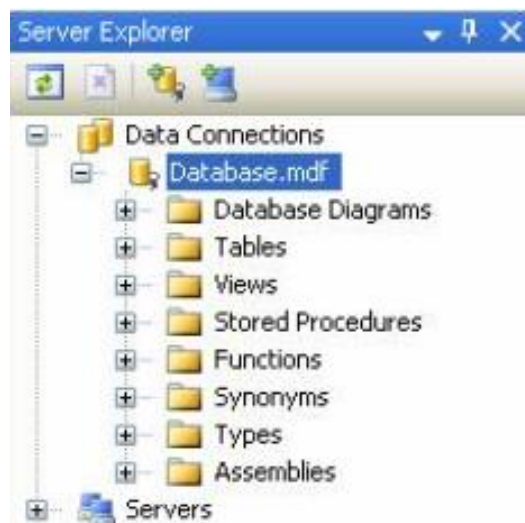
Hình 3.12

- **Bước 2.** Nhập tên Database vào hộp Name sau đó nhấn Add.
- + Trong Solution Explorer sẽ thêm vào Database trong thư mục App_Data.



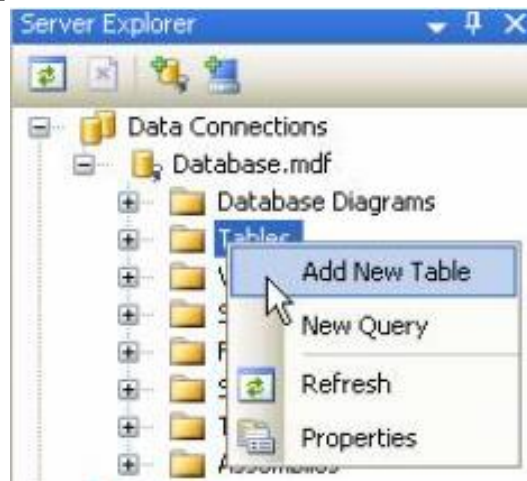
Hình 3.13

- **Bước 3.** tạo bảng dữ liệu cho Database.mdf
- + Bạn click đúp chuột vào Datatabase.mdf -> Server Explorer hiện ra như sau:



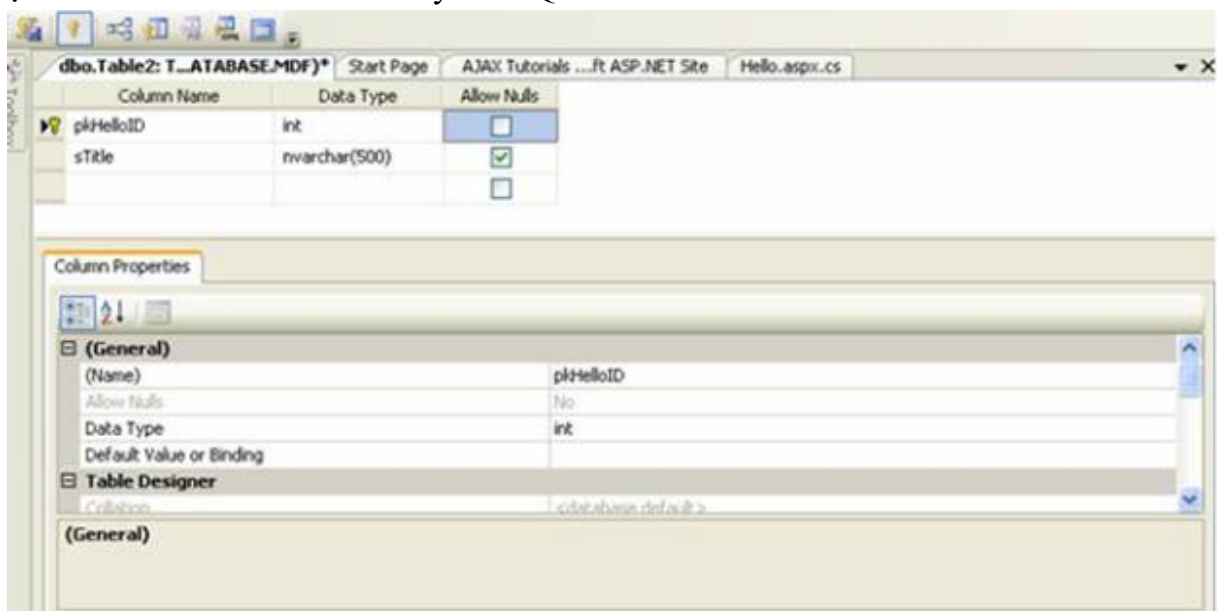
Hình 3.14

- + Bạn nhấn chuột phải vào Tables và chọn Add New Table



Hình 3.15

- Trong màn hình của VS sẽ hiện ra như hình 3.16 và bạn thao tác tạo các trường dữ liệu như thao tác với Access hay MSSQL 2000/2005



Hình 3.16

Viết code cho Hello.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        //chuỗi kết nối đến nguồn dữ liệu
        string driver = "Data
Source=(local)\\SQLEXPRESS;AttachDbFilename=|DataDirectory|Database.mdf;Integra
ted Security=True;User Instance=True";
        //đối tượng kết nối tới cơ sở dữ liệu
```

```

SqlConnection sqlconn = new SqlConnection(driver);
//Command điều khiển truy vấn sql
SqlCommand sqlcom = sqlconn.CreateCommand();
sqlcom.CommandText = "select sTitle from tblHello where pkHelloID=1";
//mở kết nối dữ liệu
sqlconn.Open();
//lấy về chuỗi giá trị trong cơ sở dữ liệu
string result = (string)sqlcom.ExecuteScalar();
//đóng kết nối
sqlconn.Close();
//in giá trị ra màn hình
Response.Write(result);
}
}

```

Các bước Cơ bản thực hiện với database

- Bước 1: Tại kết nối
- Bước 2: Tạo lệnh điều khiển truy vấn SQL
- Bước 3: Mở kết nối dữ liệu
- Bước 4: thực thi lệnh
- Bước 5: đóng kết nối
- Bước 6: in kết quả

Đối tượng Connection

Kết nối cơ sở dữ liệu SQLServer

Bạn cần nhập khẩu lớp SqlClient

using System.Data.SqlClient;

Khai báo và khởi tạo:

SqlConnection sqlcon;

string driver="server=localhost; UID=sa; PWD=; database=name_database";

sqlcon=new SqlConnection();

sqlcon.ConnectionString=driver;

Driver là chuỗi kết nối đến cơ sở dữ liệu trong trường hợp này mình kết nối với sqlserver 2000/2005

Kết nối với cơ sở dữ liệu Access

Bạn cần nhập khẩu lớp OleDb

using System.Data.OleDb;

OleDbConnection oleconn;

string driver = "Provider=Microsoft.jet.OLEDB.4.0; Data Source=duongdan_tendata";

oleconn = new OleDbConnection();

oleconn.ConnectionString = driver;

Thuộc tính:

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

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

```
Sqlconn.ChangeDatabase("name_database_thaydoi");
```

Server: tên máy chủ bạn trở tới

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

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

```
Response.Write(sqlconn.State)
```

2.3.1 Phương thức của đối tượng Connection

Các phương thức của đối tượng Connection

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

Close: Đóng cơ sở dữ liệu đang mở

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

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

```
SqlCommand sqlcom = sqlconn.CreateCommand();
```

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

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

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

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

Đối tượng SqlCommand

Khai báo và khởi tạo đối tượng

Cách 1:

```
SqlCommand sqlcom;
```

```
sqlcom=new SqlCommand(ssql,sqlconn)
```

cách 2:

```
SqlCommand sqlcom = new SqlCommand();
```

```
sqlcom.Connection = sqlconn;
```

```
sqlcom.CommandType = CommandType.Text;
```

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

Phương thức

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

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

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

Đối tượng SqlDataReader

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

Khai báo và khởi tạo đối tượng

```
SqlDataReader sqlreader;  
sqlreader = sqlcom.ExecuteReader();
```

Đối tượng DataAdapter

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

Khai báo, khởi tạo và giải phóng đối tượng.

```
string ssl;
```

Khai báo đối tượng

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

cách 1.

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

cách 2.

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

Giải phóng đối tượng

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

Thuộc tính:

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

Phương thức:

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

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

Điền dữ liệu từ Adapter vào DataSet

Dataset là một thùng chứa dữ liệu không kết nối

```
public static DataSet Filldataset(string ssl)  
{  
    DataSet dataset = new DataSet();  
    opendata();  
    try  
    {  
        sqladapter = new SqlDataAdapter(ssl, sqlconn);  
        sqladapter.Fill(dataset);  
        sqladapter.Dispose();  
    }  
}
```

```

        catch (Exception exp)
        {
            closedata();
            System.Web.HttpContext.Current.Response.Write(exp.ToString());
        }
        closedata();
        return dataset;
    }

```

Điền dữ liệu vào DataTable

```

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

```

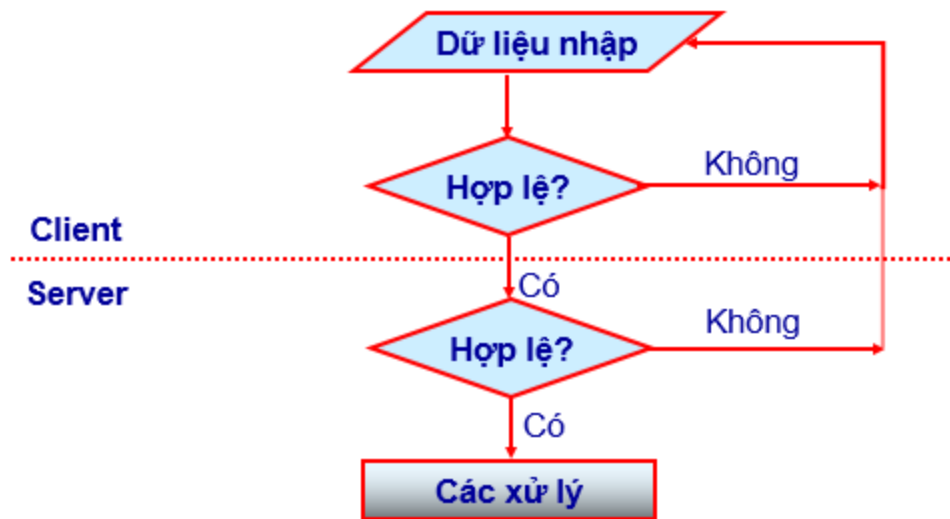
2.4 Điều khiển kiểm tra hợp lệ

Giới thiệu chung

Dùng để kiểm tra tính hợp lệ của dữ liệu

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

Qui trình kiểm tra:



Hình 3.17 Quy trình kiểm tra

Một số thuộc tính cơ bản

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

2.5 Điều khiển hiển thị dữ liệu

2.5.1. RequireFieldValidator

- Kiểm tra giá trị trong điều khiển phải được nhập
- Thuộc tính `InitialValue`:
- Giá trị khởi động
- Giá trị nhập vào phải khác với giá trị của thuộc tính này

2.5.2. CompareValidator

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

2.5.3. RangeValidator

- Kiểm tra giá trị của điều khiển nằm trong đoạn [min-max]
- Thuộc tính:
 - + `MinimumValue`: giá trị nhỏ nhất
 - + `MaximumValue`: giá trị lớn nhất
 - + `Type`: xác định kiểu dữ liệu để kiểm tra (String/Integer/Double/Date/Currency)

2.5.4. RegularExpressionValidator

- Kiểm tra giá trị của điều khiển phải theo mẫu qui định
- Thuộc tính:
 - + ValidationExpression: qui định mẫu kiểm tra dữ liệu dựa vào các ký hiệu qui định

2.5.5. CustomValidator

- Cho phép người dùng tự viết hàm xử lý kiểm tra lỗi
- Thuộc tính:
 - + ValidateEmptyText(True/False): có kiểm tra khi giá trị nhập là rỗng
 - + ClientValidationFunction: tên hàm kiểm tra ở mức client
 - + Sự kiện ServerValidate: dùng để xử lý kiểm tra dữ liệu ở mức Server

2.5.6. ValidationSummary

- Dùng để hiển thị bảng lỗi - tất cả các lỗi hiện có trên trang Web (thuộc tính ErrorMessage)
- Thuộc tính:
 - + HeaderText: chuỗi văn bản của dòng tiêu đề
 - + ShowMessageBox(True/False): có thể hiện hộp thoại thông báo lỗi trên Browser

2.8 Điều khiển kết nối dữ liệu

2.8.1 Tổng quan

- Phân loại
 - + Điều khiển Danh sách - List Controls
 - + Điều khiển liên kết dữ liệu – Data-Bound controls
 - + Điều khiển nguồn dữ liệu – Data-Source controls
- Các điều khiển này có khả năng liên kết với một nguồn dữ liệu
- Nguồn dữ liệu dùng để liên kết: ArrayList, SortedList, HashTable, DataSet, DataTable, DataView, DataReader, ...
- Thao tác liên kết dữ liệu của điều khiển List và Data-Bound: 2 cách
 - + Lúc thực thi: thường sử dụng thuộc tính DataSource của điều khiển
 - + Lúc thiết kế: sử dụng các điều khiển Data source (AccessDataSource, SqlDataSource, ...) kết hợp với thuộc tính DataSourceID

2.8.2 Các điều khiển Danh sách

ListBox và DropDownList

- Dùng để hiển thị danh sách
- Items: chứa tập hợp các mục chọn, có thể thêm vào lúc thiết kế (Design) hoặc lúc thực thi (Run-time)
- Rows: số dòng hiển thị của ListBox
- SelectionMode: cách thức chọn các mục trong ListBox (Single/Multi)
- DataTextField: tên field có giá trị thể hiện trên điều khiển
- DataValueField: tên field có giá trị lưu trữ
- AutoPostBack: có tự độngPostBack về Server khi mục chọn của điều khiển bị thay đổi (True/False)
- Một số thuộc tính lúc thực thi:

- + SelectedIndex: chỉ số của mục được chọn
- + SelectedItem: mục được chọn (kiểu ListItem)
- + SelectedValue: giá trị của mục được chọn
- + Sự kiện SelectedIndexChanged: xảy ra khi thay đổi dòng được chọn

Một số thao tác:

- + Thêm mới một mục vào cuối danh sách Items.Add(<String>) hoặc Items.Add(<ListItem>)
- + Thêm mới một mục tại một vị trí xác định Items.Insert(<Index>,<ListItem>) hoặc Items.Insert(<Index>,<String>)
- + Xóa một mục ra khỏi danh sách Items.Remove(<ListItem>) hoặc Items.Remove(<String>)
- + Xóa tất cả các mục trong danh sách Items.clear
- + Kiểm tra một mục đã có trong danh sách chưa Items.Contains(<ListItem>) As Boolean hoặc Items.IndexOf(<ListItem>) As Integer
- + Lấy giá trị tổng số mục có trong danh sách Items.Count
- CheckBoxList và RadioButtonList
 - + RepeatColumns: qui định số cột hiển thị
 - + RepeatDirection: qui định hình thức hiển thị (Vertical/Horizontal)
 - + Các thao tác thêm, xóa các mục chọn: giống ListBox
- BulletedList
 - + BulletStyle: style dùng làm Bullet
 - + BulletImageUrl: URL của tập tin hình ảnh, phải có BulletStyle=CustomImage
 - + DisplayMode: hình thức thể hiện (Text, HyperLink, LinkButton)
 - + Các thao tác thêm, xóa các mục chọn: giống ListBox
- Thao tác liên kết dữ liệu
 - 1- ArrayList <nguồn> = new ArrayList();//Khởi tạo nguồn
 - <nguồn>.Add(Object); //Thêm dữ liệu
 - 2- <đ.k>.DataSource=<nguồn>; //Khai báo nguồn dữ liệu
 - 3- <đ.k>.DataTextField="Field"; //Khai báo field thể hiện
 - 4- <đ.k>.DataValueField="Field"; //Khai báo field lưu trữ
 - 5- <đ.k>.DataBind(); //Liên kết

//Dòng 3, 4 có thể bỏ qua tùy vào nguồn dữ liệu

2.8.3 Điều khiển GridView

Giới thiệu

- Dùng hiển thị dữ liệu dạng dòng-cột (Tabular)
- Thể hiện trên trình duyệt dưới dạng HTML Table
- Đặc điểm:
 - + Hỗ trợ cơ chế sắp xếp dữ liệu
 - + Hỗ trợ cơ chế phân trang dữ liệu
 - + Có thể cập nhật dữ liệu trực tiếp trên lưới

DANH SÁCH SÁCH			
Mã	Tên sách	Đơn giá	Ngày cập nhật
1	Giáo trình Tin học cơ bản	26,000	17/07/2004
2	Giáo trình Tin học văn phòng	12,000	05/06/2004
3	Lập Trình Cơ Sở Dữ Liệu Với Visual Basic 2005 Và ADO.NET 2.0	11,500	12/07/2004

Hình 3.18 Điều khiển GridView

Liên kết dữ liệu và định dạng

- Thao tác liên kết dữ liệu lúc thực thi:
 - + <GridView>.DataSource = <Nguồn dữ liệu>;
 - + <GridView>.DataBiding();
- Thuộc tính định dạng:
 - + HtmlEncoded(True/False): cho phép mã hóa dưới dạng HTML không
 - + DataFormatString: định dạng cho giá trị của field liên kết

Các xử lý sắp xếp trên GridView

- Khai báo t.t AllowSorting = True
- Khai báo tên cột cần sắp xếp trong ô Sort expression
- Sử dụng sự kiện Sorting để xử lý:
 - + Liên kết dữ liệu và sắp xếp theo cột cần sắp xếp
 - + Giá trị e.SortExpression cho biết thông tin của cột được chọn
- Các xử lý phân trang trên GridView
 - + Khai báo t.t AllowPaging = True
 - + Khai báo t.t PageSize = <Số dòng hiển thị trên một trang>
 - + Sử dụng sự kiện PageIndexChanged để xử lý: <GridView>.PageIndex = e.NewPageIndex ; //Liên kết dữ liệu
 - + Giá trị e.NewPageIndex cho biết chỉ số trang được chọn
 - + Thuộc tính PageIndex: chỉ số trang hiện hành trên lưới

DANH SÁCH KHÁCH HÀNG			
Họ khách hàng	Tên đăng nhập	Địa chỉ	Ngày sinh
Phạm Văn Khoa	pvkhoa	Trần Huy Liệu	08/07/1962
Nguyễn Tiến Luân	ntlun	Quận 6	15/07/1974
Đặng Quốc Hòa	dqhoa	Sư Vạn Hạnh	24/07/1965
Ngô Ngọc Ngân	nnngan	Khu chung cư	15/08/1946
Đỗ Quỳnh Hương	dqhuong	Cổng Quỳnh	24/03/1982
1 2 3			

Hình 3.19 xử lý sắp xếp trên GridView

Các xử lý tùy biến cột trên Gridview

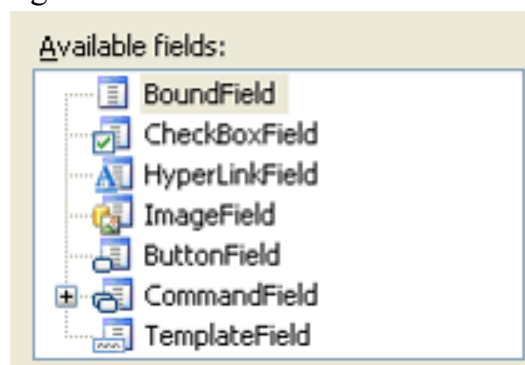
- Bước 1: khi thiết kế, cột thể hiện có loại là TemplateField
- Bước 2: dành cho xử lý đơn giản

- + Chèn một điều khiển vào cột TemplateField, liên kết DL với điều khiển này
- + Khai báo biểu thức trong ô Custom binding => Code Expression
- + Ví dụ: hiển thị cột Giới tính là “Nam” hay “Nữ” ((bool) Eval("Gioi_tinh")==false?"Nữ":"Nam")
- Bước 2: dành cho xử lý phức tạp, viết code trong sự kiện RowDataBound
 - + RowDataBound: xảy ra khi lưới được liên kết dữ liệu (DataBinding). Ứng với mỗi dòng, xảy ra sự kiện này
 - + e.Row.RowIndex: chỉ số của dòng hiện hành (Header /Footer= -1)
 - + e.Row.RowType: loại của mỗi dòng (Dòng chẵn, lẻ, ...)
- Bước 2: dành cho xử lý phức tạp, viết code trong sự kiện RowDataBound
 - + e.Row.FindControl(“Tên điều khiển”): tìm kiếm điều khiển trên dòng hiện hành
 - + DataBinder.Eval(e.Row.DataItem,“Tên cột”): truy xuất dữ liệu của cột trên dòng hiện hành

DANH SÁCH KHÁCH HÀNG				
Họ tên	Ngày sinh	Giới tính		Địa chỉ
Phạm Văn Khoa	08/07/1962	Nam		Trần Huy Liệu
Nguyễn Tiến Luân	15/07/1974	Nam		Quận 6
Đặng Quốc Hòa	24/07/1965	Nam		Sư Vạn Hạnh
Ngô Ngọc Ngân	15/08/1946	Nam		Khu chung cư
Đỗ Quỳnh Hương	24/03/1982	Nữ		Cổng Quỳnh
Trần Thị Thu Trang	05/08/1962	Nữ		Nơ Trang Long
Nguyễn Thiên Thanh	04/04/1979	Nữ		Hai Bà Trưng
1 2				

Hình 3.20 xử lý sắp xếp trên GridView

Giới thiệu 3 loại cột trong GridView:



Hình 3.21 Ba loại cột trong GridView

ImageField

- DataImageUrlField: chứa Image URL liên kết với field
- DataImageUrlFormatString: định dạng cho giá trị liên kết với thuộc tính DataImageUrlField

Ví dụ:

- + DataImageUrlField: Hình_minh_hoa
- + DataImageUrlFormatString: ~/Hinh_du_lieu/{0}.jpg

HyperLinkField

- DataNavigateUrlFields: tên field liên kết với thuộc tính NavigateUrl của hyperlink
- DataNavigateUrlFormatString: định dạng cho giá trị liên kết với thuộc tính DataNavigateUrlFields

- DataTextField: field có nội dung được hiển thị

ButtonField

- ButtonType: kiểu nút lệnh (Button/Link/Image)
- CommandName: tên kết hợp với nút lệnh
- DataTextField: field có nội dung được hiển thị
- DataTextFormatString: định dạng chuỗi thể hiện trên nút
- Sự kiện RowCommand: xảy ra khi nhấn nút lệnh

CẬP NHẬT THÔNG TIN NHÀ XUẤT BẢN					
Tên nhà xuất bản	Địa chỉ	Điện thoại			
NXB Thống kê	Đồng Nai	19001511	Chọn	X	Sửa
Kim đồng	Tp.HCM	19001570	Chọn	X	Sửa
Đại học quốc gia	Tp.HCM	0908419981	Chọn	X	Sửa
Văn hóa nghệ thuật	Đà Nẵng	0903118833	Chọn	X	Sửa
Văn hóa	Bình Dương	0913336677			Ghi Không
Lao động - Xã hội	Tp.HCM	0989888888	Chọn	X	Sửa
Khoa Học & Kỹ Thuật	Hà Nội	8201552	Chọn	X	Sửa
NXB Tài Chính	Huế		Chọn	X	Sửa
NXB Phụ Nữ	Tp.HCM	8351056	Chọn	X	Sửa

Hình 3.22 Ba loại cột trong GridView

Cập nhật dữ liệu:

Yêu cầu đầu tiên: khai báo các field dùng làm khóa cho lưới
 <GridView>.DataKeyNames = new string[] { "tên field1", ... };

- Giai đoạn 1: thiết kế, chọn loại cột thể hiện là CommandField. Thuộc tính CommandName của các nút sau có giá trị mặc định theo chức năng tương ứng:

- + Nút Delete => CommandName = "Delete"
- + Nút Edit => CommandName = "Edit"
- + Nút Update => CommandName = "Update"
- + Nút Cancel, Select

- Giai đoạn 2: xử lý. Sử dụng các sự kiện sau:

- + RowDeleting: xảy ra khi nhấn nút Delete, trước khi GridView xóa dòng
- + RowEditing: xảy ra khi nhấn nút Edit, trước khi GridView sang chế độ edit

- + RowUpadating: xảy ra khi nhấn nút Update, trước khi GridView cập nhật
 - + RowCanceling: xảy ra khi nhấn nút Cancel, trước khi GridView thoát chế độ edit
 - + SelectedIndexChanging: khi nhấn nút Select
 - Giai đoạn 2: xử lý. Sử dụng các thuộc tính thường dùng sau:
 - + <GridView>.EditIndex: chỉ số dòng được sửa ở trạng thái edit
 - + e.NewEditIndex: chỉ số của dòng đang sửa (sự kiện RowEditing)
 - + e.RowIndex: chỉ số của dòng hiện hành e.Rows[n1].Cells[n2].Controls[n3]:
- nhận điều khiển
- + <GridView>.DataKeys[n]: truy xuất giá trị khóa tại dòng có chỉ số là n
 - Giai đoạn 2: xử lý. Có các chức năng sau:
 - + Chọn dòng
 - + Sửa: chuyển GridView sang trạng thái edit
 - + Không: khôi phục dữ liệu ban đầu và trả về trạng thái ReadOnly
 - + Ghi: cập nhật dữ liệu vào CSDL và trả về trạng thái ReadOnly

2.8.4 Các điều khiển DataSource

Là các điều khiển dùng để chứa nguồn dữ liệu được rút trích từ các loại CSDL (Access, Sql Server, ...)

- Dùng để liên kết với các điều khiển Data-Bound và khi liên kết thì chúng sẽ hỗ trợ các tính năng sau:
 - + Hỗ trợ phân trang
 - + Hỗ trợ sắp xếp tăng/giảm tự động
 - + Cập nhật dữ liệu
 - + Tất cả các xử lý trên gần như không cần viết code

Các điều khiển

- AccessDataSource: kết nối đến tập tin CSDL Access
- SqlDataSource : kết nối đến CSDL SQL Server, Oracle, ...
- XmlDataSource : kết nối đến tập tin XML
- ObjectDataSource : kết nối đến một object (class, ...) hoặc Typed-DataSet đặt trong thư mục Bin | App_Code
- SitemapDataSource : liên kết đến tập tin .sitemap

Các thuộc tính cơ bản

- DataFile: tập tin CSDL Access (đường dẫn đầy đủ)
- ConnectionString: chuỗi liên kết với các CSDL(SQL Server, Oracle, ...)
- SelectQuery: truy vấn chọn lựa
- InsertQuery: truy vấn thêm mẫu tin
- UpdateQuery: truy vấn cập nhật
- DeleteQuery: truy vấn xóa

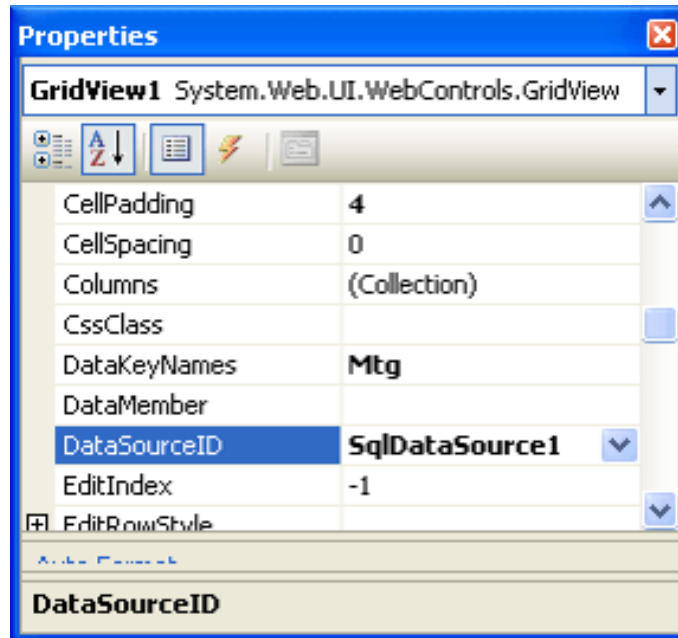
Điều khiển SqlDataSource

- Thao tác cấu hình:
- Chọn điều khiển, nhấn phím phải => configure Data Source
- Khai báo Connection Chọn Data Source
- Chọn Microsoft SQL Server

- Khai báo Server name và CSDL
- Cấu hình câu lệnh Select và phát sinh các câu lệnh cập nhật
- Kiểm tra và kết thúc

Sử dụng SqlDataSource với GridView

- Khai báo thuộc tính DataSourceID của GridView



Hình 3.23 Sử dụng SqlDataSource với GridView

Sử dụng SqlDataSource với GridView Lọc dữ liệu

- FilterExpression: biểu thức lọc, chỉ áp dụng khi thuộc tính DataSourceMode là DataSet
- FilterParameters: các tham số tham gia trong thuộc tính
- FilterExpression SelectCommandType (Text/StoredProcedure): kiểu tạo truy vấn chọn lựa (dùng câu lệnh SQL hay thủ tục nội)

Sử dụng SqlDataSource với GridView Cập nhật dữ liệu

- Hủy: viết câu lệnh truy vấn xóa có tham số tại thuộc tính DeleteQuery. Có thể bổ sung phần kiểm tra trước khi hủy trong sự kiện RowDeleting
- Cập nhật: viết câu lệnh truy vấn cập nhật có tham số tại thuộc tính UpdateQuery
 - + Có thể bổ sung phần kiểm tra dữ liệu trước khi cập nhật trong sự kiện RowUpdating.
 - + Sử dụng thuộc tính e.NewValues[“tên field”] để kiểm tra

```
protected void GridView4_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    //Kiểm tra Tên chủ đề khác ""
    if (string.IsNullOrEmpty((string)e.NewValues["Ten_chu_de"]))
        e.Cancel = true;
}
```

2.8.5 Điều khiển DataList

Dùng để hiển thị dữ liệu theo nhiều dạng

Thẻ hiện trên trình duyệt dưới dạng HTML Table

Đặc điểm:

- Không hỗ trợ cơ chế sắp xếp dữ liệu, phân trang

- Có thể cập nhật dữ liệu trực tiếp trên danh sách
- Phải tự thiết kế hình thức



Hình 3.24 Điều khiển DataList

Liên kết và định dạng

- + Thao tác liên kết dữ liệu: giống GridView
- + Thao tác thiết kế: giống như cột TemplateField của GridView
- + Định dạng: giống GridView
 - RepeatDirection(Horizontal/Vertical): hướng hiển thị dữ liệu
 - RepeatColumns: số cột hiển thị

Cập nhật dữ liệu datalist

Yêu cầu đầu tiên: khai báo field dùng làm khóa cho danh sách <DataList>.DataKeyField = “tên field”;

- Bước 1: thiết kế, bổ sung các điều khiển trong thành phần EditItemTemplate. Nên khai báo thuộc tính CommandName của các nút chức năng:

- + Nút Edit => CommandName = “Edit”
- + Nút Update => CommandName = “Update”
- + Nút Cancel => CommandName = “Cancel”
- Bước 2: xử lý, có các sự kiện
 - + EditCommand # RowEditing (GridView)
 - + UpdateComand # RowDeleting (GridView)
 - + ItemComand #RowCommand (GridView)
- Bước 2: xử lý, có các thuộc tính thường dùng
 - + <DataList>.EditItemIndex: chỉ số dòng được sửa ở trạng thái edit
 - + e.Item.ItemIndex: chỉ số của dòng hiện hành

- + `e.Item.FindControl("tên điều khiển")`: tìm kiếm điều khiển trên dòng hiện hành
- + `<DataList>.DataKeys[n]`: truy xuất giá trị khóa tại dòng có chỉ số là n

2.8.6 DetailsView và FormView

Giới thiệu FormView

- Dùng hiển thị dữ liệu dạng cột (Columnar)
- Thể hiện trên trình duyệt dưới dạng HTML Table
- Đặc điểm:
 - + Hỗ trợ cơ chế phân trang dữ liệu
 - + Cho phép cập nhật dữ liệu trực tiếp
 - + Cho phép kết hợp với GridView để tạo ra dạng Master-Detail

Thông tin khách hàng	
Họ tên	Nguyễn Thiên Thanh
Ngày sinh	04/04/1979
Phái	☐
Điện thoại	0908320111
Địa chỉ	Hai Bà Trưng
Email	ntthanh@t3h.hcmuns.edu.vn
1 2 3 4 5 6 7 8 9 10 ...	

Hình 3.25 DetailsView và FormView

Liên kết dữ liệu và định dạng

- Thao tác liên kết dữ liệu: giống GridView
- Thao tác định dạng

Các xử lý

- Phân trang: giống GridView
- Cập nhật dữ liệu: Khai báo các thuộc tính phát sinh các nút cập nhật:
 - + `AutoGenerateDeleteButton`
 - + `AutoGenerateEditButton`
 - + `AutoGenerateInsertButton`
 - + Xử lý thì giống GridView, khác nhau về tên sự kiện và một số thuộc tính

Sự kiện trên GridView	Sự kiện trên DetailView
RowCommand	ItemCommand
RowInserting	ItemInserting
RowUpdating	ItemUpdating
RowDeleting	ItemDeleting
	ModeChanging : khi chuyển sang một trạng thái khác(edit, insert, read-only)

Hình 3.26 Liên kết dữ liệu và định dạng

Tạo dạng Master-Detail

- Bước 1: tạo `SqlDataSource1` dùng làm nguồn cho GridView

- Bước 2: tạo GridView dùng làm Master và kết nguồn vào (Nên tạo một nút lệnh xử lý chọn dòng)
- Bước 3: tạo SqlDataSource2 dùng làm nguồn cho DetailsView + FilterParameters: giá trị của tham số là <GridView>.SelectedValue + FilterExpression: biểu thức lọc tương ứng với tham số
- Bước 4: tạo DetailsView dùng làm Detail và kết nguồn vào

Danh sách sách		Chi tiết sách	
Mã sách	Tên sách	Tên sách	Giáo trình Tin học cơ bản
1	Giáo trình Tin học cơ bản	Hình ảnh	
2	Giáo trình Tin học văn phòng		
3	Lập Trình Cơ Sở Dữ Liệu Với Visual Basic 2005 Và ADO.NET 2.0	Mô tả	Nội dung của cuốn: Tin Học Cơ Bản Windows XP gồm có 7 chương: Chương 1: Một số vấn đề cơ bản. Chương 2: Sử dụng nhanh thành công cụ và thanh thực đơn trong My Computer và Windows Explorer. Chương 3: Các thao tác trong windows XP. Chương 4: Các thiết lập trong Windows XP. Chương 5: Bảo trì máy tính. Chương 6: Các phím tắt Chương 7: Hỏi và đáp các thắc mắc. Xin trân trọng giới thiệu cuốn sách cùng bạn
4	Visual Basic 2005 Tập 3, Quyển 2: Lập Trình Web Với Cơ Sở Dữ Liệu		
5	Giáo trình Tin học quản lý 1	Đơn giá	26,000
6	Giáo trình Tin học quản lý 2		
7	Giáo trình SQL Server		
8	Giáo trình Phân tích hệ thống		
9	Giáo trình VB.Net 1		
10	Giáo trình VB.Net 2		
1 2 3 4 5			

Hình 3.27 Tạo dạng Master-Detail

Giới thiệu FormView

- Tương tự DetailsView nhưng phải tự thiết kế dạng hiển thị
- Thể hiện trên trình duyệt dưới dạng HTML Table
- Đặc điểm:
 - + Hỗ trợ cơ chế phân trang dữ liệu
 - + Cho phép cập nhật dữ liệu trực tiếp

Liên kết, định dạng và các xử lý:

- Giống DetailsView

2.9 Thiết kế giao diện phía người dùng web

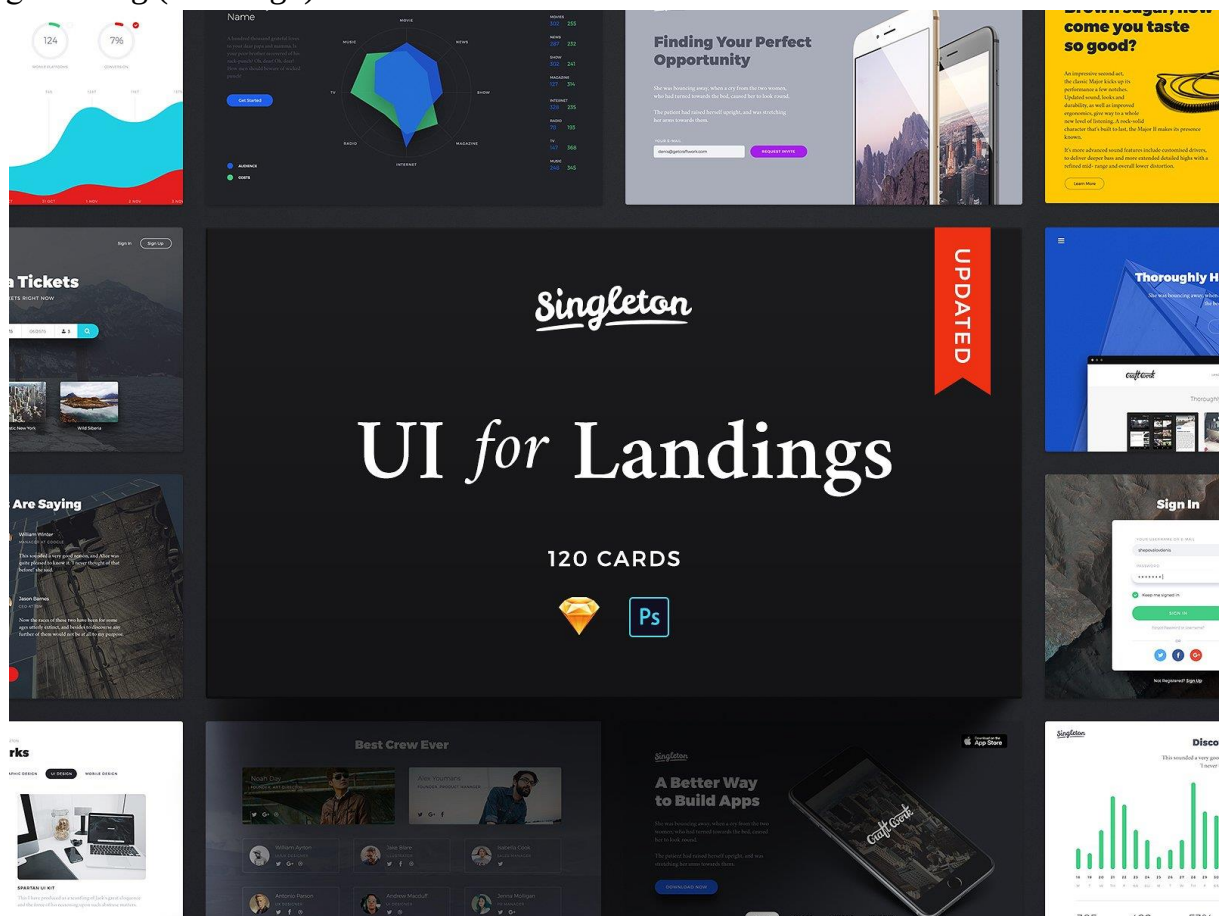
Chúng ta vẫn thường nghe đến việc thiết kế giao diện người dùng (UI). Vậy chúng ta có thật sự hiểu rõ về UI chưa ? Hãy cùng tìm hiểu thông qua giáo trình này nhé.

2.9.1 Thiết kế giao diện người dùng (UI) là gì ?

Giao diện người dùng hay còn gọi là UI là giao diện của người dùng cuối và phần mềm phía sau nó. Ví dụ như trong một chiếc smartphone, màn hình người dùng của các ứng dụng như Google Maps hay là Facebook, thông qua đó người dùng có thể đưa thông tin và nhận phản hồi, đó chính là giao diện người dùng.

Các nhà **thiết kế website**, nhà phát triển ứng dụng và kinh doanh thương mại điện tử dành rất nhiều công sức để tìm cách hiểu được yêu cầu của người dùng: ví dụ như họ muốn điều hướng như thế nào, menu có những gì,... trước khi bắt tay vào thiết kế UI cho ứng dụng của họ. Toàn bộ quá trình thu thập yêu cầu người dùng, đặt những yếu tố khác nhau

của phần mềm và tạo ra một giao diện người dùng hiệu quả được gọi là thiết kế giao diện người dùng (UI design).



Hình 3.28 Thiết kế giao diện người dùng UI

2.9.2. Các thành phần của UI Design

Thiết kế giao diện người dùng UI bao gồm:

Bố cục

- Bố cục sẽ giúp xác định các thành phần sẽ được bố trí trên trang, chúng sẽ được đặt ở đâu và đặt như thế nào. Đây là một yếu tố vô cùng quan trọng. Bố cục nên đơn giản để người dùng có thể dễ dàng tìm được thông tin họ muốn và cái bạn muốn họ tìm thấy trên *thiết kế website* của bạn.

Màu sắc

- Màu sắc cũng ảnh hưởng khá nhiều đến giao diện người dùng. Chúng phải thể hiện được thương hiệu của công ty và gây được ấn tượng với người dùng.

Kiểu chữ

- Nghệ thuật sắp xếp chữ cũng sẽ góp phần vào thành công trong việc thiết kế giao diện người dùng. Cho dù là một dòng chữ nhỏ nó cũng phải tạo nên sự thu hút

Đồ họa

- Thiết kế đồ họa đẹp là cả một nghệ thuật. Đó là về truyền đạt những thông điệp bạn muốn chỉ trong một bức tranh. Một lần nữa, cũng giống như tất cả các yếu tố khác của thiết kế giao diện, thiết kế đồ họa cần phải được quan tâm để mà thu hút đối tượng mục tiêu. Một thiết kế giao diện người dùng tốt là một thiết kế mà trong đó có sự kết hợp giữa những tính năng khác nhau, dòng thông tin tự do và các biểu tượng được đặt ở vị trí phù hợp cho

tầm nhìn tốt hơn.

2.10 Thiết kế giao diện quản trị Web

Để thay đổi nội dung của khung website, thay đổi logo, ảnh, slide, banner quảng cáo, Keyweb cung cấp hệ thống Option, kết hợp giao diện trực quan ngay trong admin, cho phép khách hàng vừa sửa vừa xem.

Bước 1: Mở trình quản lý giao diện

Để chỉnh sửa giao diện, bạn cần truy cập mục “**Giao diện**”(1) ở thanh menu bar bên trái. Trình duyệt sẽ chuyển sang trang mới, hiển thị website trực quan và cột Option bên phải.



Hình 3.29 trình quản lý giao diện

Tại cột bên phải sẽ có các Tab chỉnh sửa, Click vào các Tab này sẽ hiện ra chi tiết các Option cho từng layout, trang chủ, trang liên hệ...

- **Thiết lập chung (2):** Tab này cho phép chỉnh các thông tin cơ bản của mọi website, gồm: logo, favicon, cấu hình chung mã nguồn, cấu hình mã phần mềm chat, mã Pixel facebook...

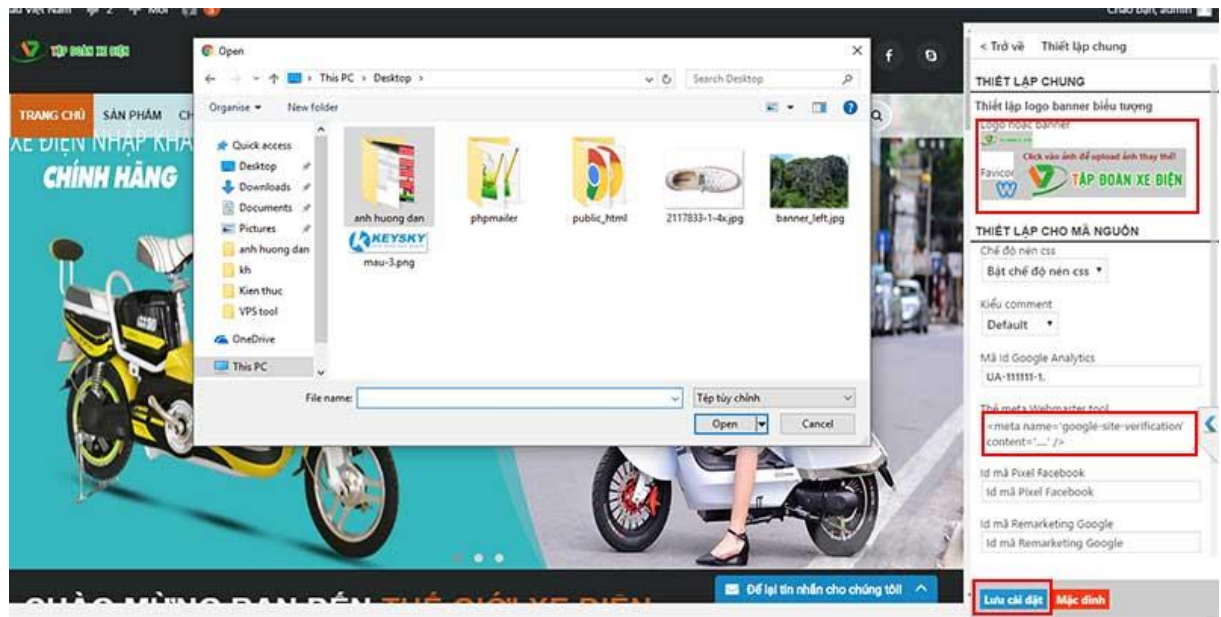
- **Thiết lập layout (3):** Gồm các mục: Chỉnh sửa Header, Chỉnh sửa Footer, Chỉnh sửa Sidebar. Tại đây bạn sẽ chỉnh sửa được nội dung, văn bản, hình ảnh, số điện thoại nếu layout đó có...

- **Thiết lập trang (4):** Gồm thiết lập cho các trang cơ bản, trang chủ, trang liên hệ...

- **Thiết lập slide, banner (5):** Tùy từng website sẽ sử dụng slider, banner, ảnh quảng cáo, chúng tôi tích hợp các Option riêng ra để bạn dễ sử dụng hơn.

Bước 2: Chỉnh sửa ảnh, văn bản

Mở từng Tab để tìm mục chỉnh sửa.



Hình 3.30 trình quản lý giao diện

- Với ảnh, ta click trực tiếp vào ảnh đó để mở trình Upload, sau đó chọn file ảnh Upload.
 - Với text ta chỉnh sửa trực tiếp vào ô nội dung.
- Sau khi hoàn thành bạn nhấn lưu để upload và lưu trữ nội dung vừa sửa

2.11 Hoàn thiện Website động

2.11.1. Master Pages

Master Pages cho phép tạo ra một bố cục nhất quán trên các trang web trong cùng một Web site (tương tự Template)

Nội dung thể hiện trên trình duyệt chính là sự kết hợp giữa trang master (Master page) và trang aspx (Content page)

Master Pages gồm 2 phần chính:

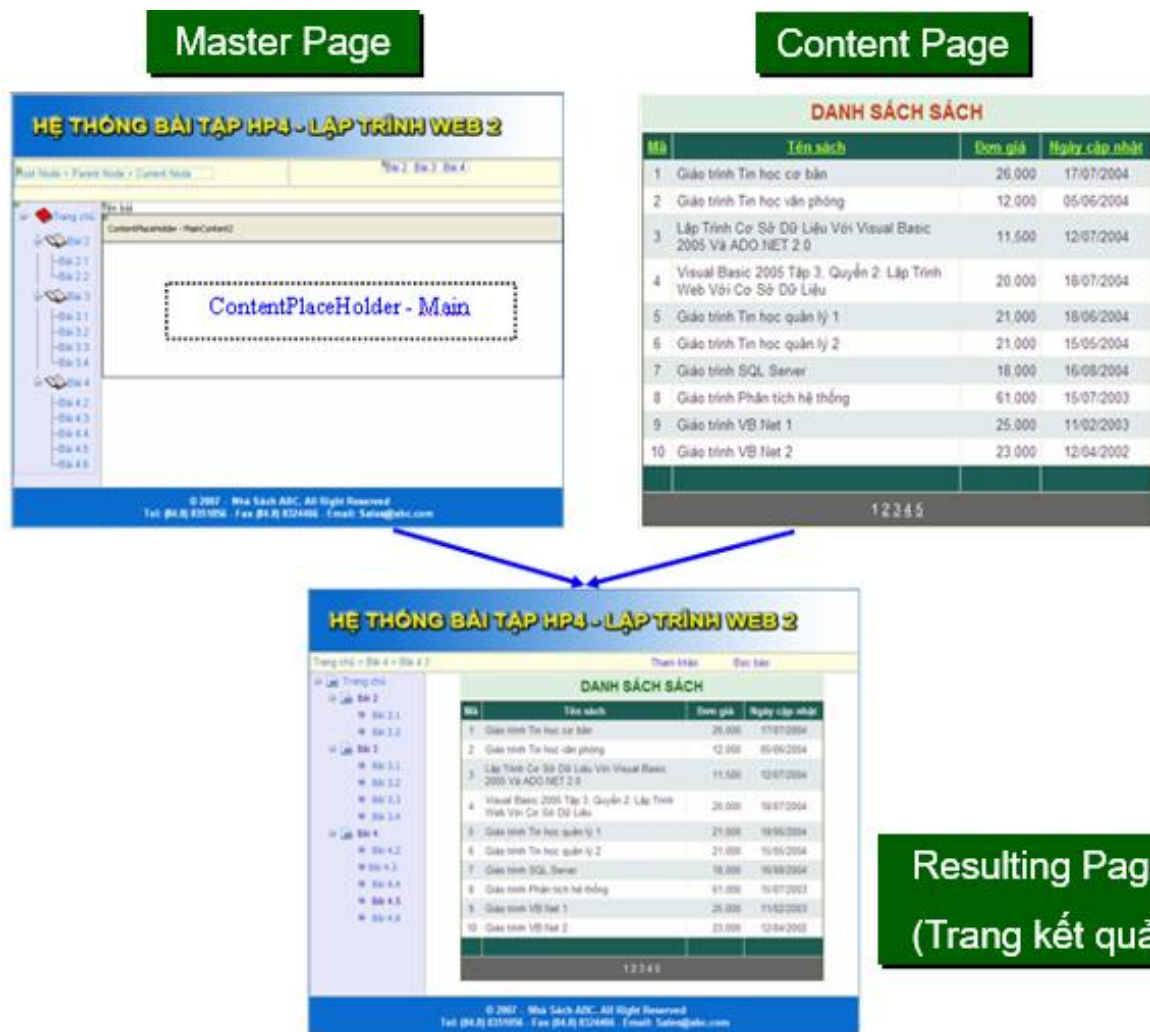
Master page

- Dùng để xây dựng một bố cục chung cho Web site
- Tập tin lưu trữ có phần mở rộng là .master
- Thay thế tag định hướng <%@ Page ...%> bằng tag <%@ Master ...%>
- Thao tác thiết kế: giống trang .aspx, nhưng phải bổ sung ít nhất một điều khiển

ContentPalceHolder

Content page: chính là trang .aspx nhưng có các đặc điểm sau

- Bổ sung thuộc tính MasterPageFile="Tên tập tin master" trong định hướng <%@ Page ... %>
- Không chứa các tag tạo ra cấu trúc trang HTML
- Chứa tag <asp:Content ...> </asp:Content>: số lượng tag này tùy thuộc vào trang master được kết hợp
- Chỉ được phép tạo nội dung trong tag <asp:Content ...>



Hình 3.31 Master Pages

Cách bước tạo Master Pages

- Bước 1: tạo trang Master => chọn icon Master Page
- Bước 2: thiết kế bố cục chung và bổ sung điều khiển ContentPlaceHolder vào trang Master
- Bước 3: tạo trang Content và khai báo trang Master
- Bước 4: thiết kế nội dung trên trang Content
- Tham chiếu thuộc tính và điều khiển của trang Master từ trang Content
- Bước 1: tạo thuộc tính (Public) trong trang Master
- Bước 2: bổ sung định hướng `<%@ MasterType VirtualPath="<tập tin>.master"%>` trong trang Content
 - Bước 3: trong trang Content, tại sự kiện Page_Load ta tham chiếu đến thuộc tính như sau: **Master.<tên thuộc tính>**
 - Sử dụng phương thức FindControl tại sự kiện Page_Load của trang Content
 - Ví dụ: điều khiển Nhan_Master trên trang Master
 - + `Label2.Text = ((Label) Master.FindControl("Nhan_Master")).Text;`
 - **Master động – Dynamic Master**
 - + Thay đổi trang Master lúc thực thi
 - + Tại sự kiện PreInit của trang Content, khai báo như sau: **Page.MasterPageFile**

= "tên tập tin Master"

2.11.2. Themes và Skin

Themes là tập hợp các khai báo thuộc tính về hình thức hiển thị (màu sắc, nét chữ,...) của điều khiển

Dùng để áp dụng chung cho tất cả các trang trong một ứng dụng hoặc tất cả các ứng dụng trên Web server

Themes được hiểu và xử lý ở phía Server

Themes được tạo ra từ các thành phần: **Skins**, **CSS**, **Images** và các nguồn khác nhưng tối thiểu phải có **Skins**

Tất cả các thành phần trên được đặt trong thư mục App_Themes

- Skins: Có 2 loại skin:

+ **Default skin** : không khai báo thuộc tính SkinId, và mặc định ứng dụng sẽ áp dụng skin này cho tất cả các điều khiển có cùng kiểu khai báo

+ **Named skin** : có khai báo thuộc tính SkinId, muốn sử dụng thì phải khai báo tường minh thuộc tính SkinId trên một điều khiển cụ thể

□ Skins:

- Có 2 loại skin:

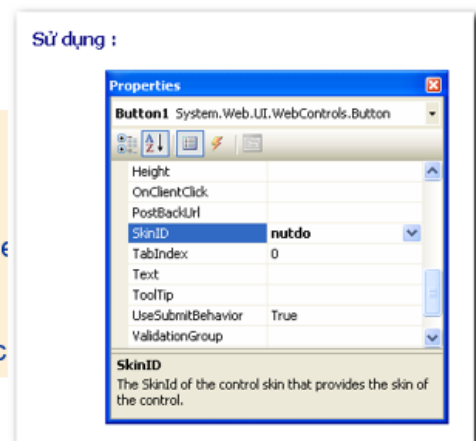
Ví dụ: khai báo trong tập tin .skin

//default skin

<asp:button runat="server" BackColor="lightblue

//named skin

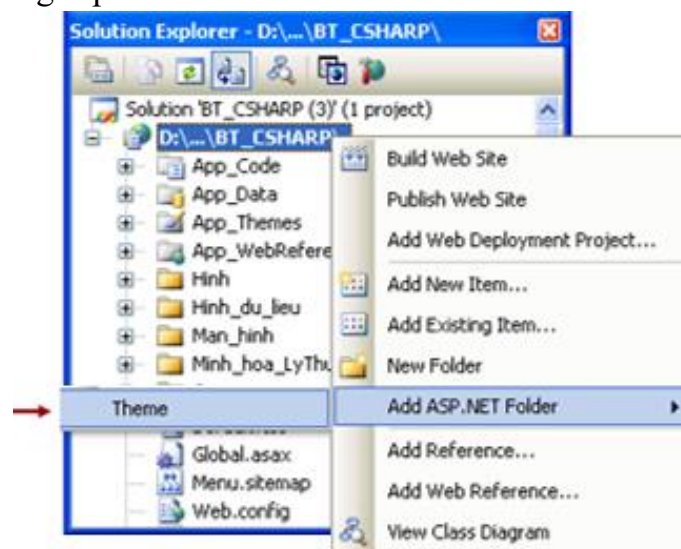
<asp:button SkinId="nutdo" runat="server" Bac



Hình 3.32 Themes và Skin

Tạo Themes

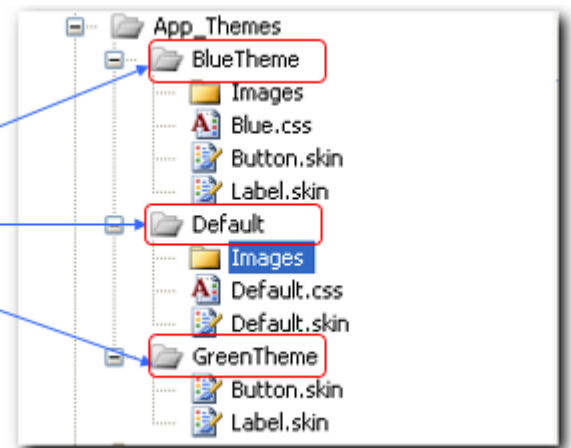
- Tạo themes trong asp.net



Hình 3.33 Themes và Skin

Tổ chức thư mục lưu trữ

Ứng dụng có 3 Themes



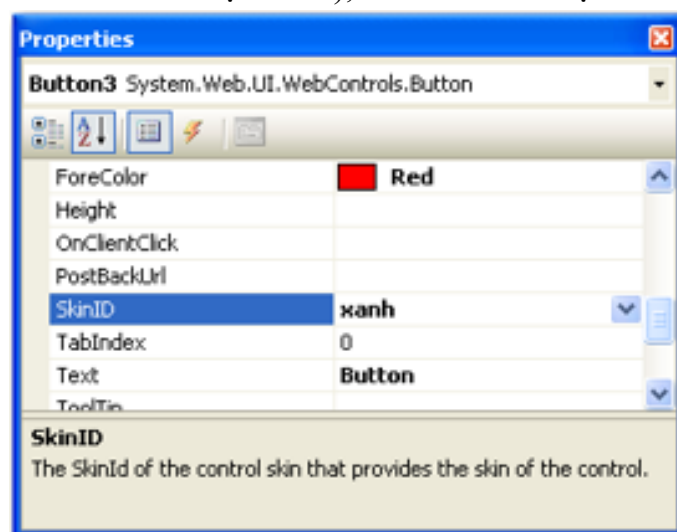
Hình 3.34 chức thư mục lưu trữ

Sử dụng Themes

- **Cấp ứng dụng:** áp dụng cho tất cả các trang. Khai báo trong tập tin Web.config
`<pages theme="MasterCode">`
...
`</pages>`
- **Cấp trang – page Themes:** áp dụng cho một trang cụ thể. Khai báo trong tag định hướng `<%@ Page ...%>` của trang .aspx
`<%@ Page Theme="MasterCode" %>`
- **Cấp trang – page Themes:** áp dụng cho một trang cụ thể. Khai báo trong tag định hướng `<%@ Page ...>` của trang .aspx
`<%@ Page StylesheetTheme="MasterCode" ... %>`

Hủy bỏ Themes:

- Muốn hủy bỏ Themes ở cấp ứng dụng cho một trang cụ thể, ta sử dụng thuộc tính `EnableTheming` trong `<%@ Page EnabledTheming = "false" ... %>`
- Muốn hủy bỏ Themes cho riêng một điều khiển, ta khai báo thuộc tính `EnableTheming` là `false`
- **Khai báo skin cụ thể cho một điều khiển:** Muốn xác định skin cụ thể (không sử dụng skin mặc định của Themes hiện hành), ta khai báo thuộc tính `SkinId`



Hình 3.35 Khai báo skin cụ thể cho một điều khiển

Themes động – Dynamic Themes

- Sử dụng sự kiện Page_PreInit của trang
- Thay đổi Themes

```
protected void Page_PreInit( ... )
{
    Page.Theme = "tên Theme";
}
protected void Page_PreInit( ... )
{
    <điều khiển>.SkinID = "giá trị SkinId";
}
```

2.11.3. Site Navigation Controls

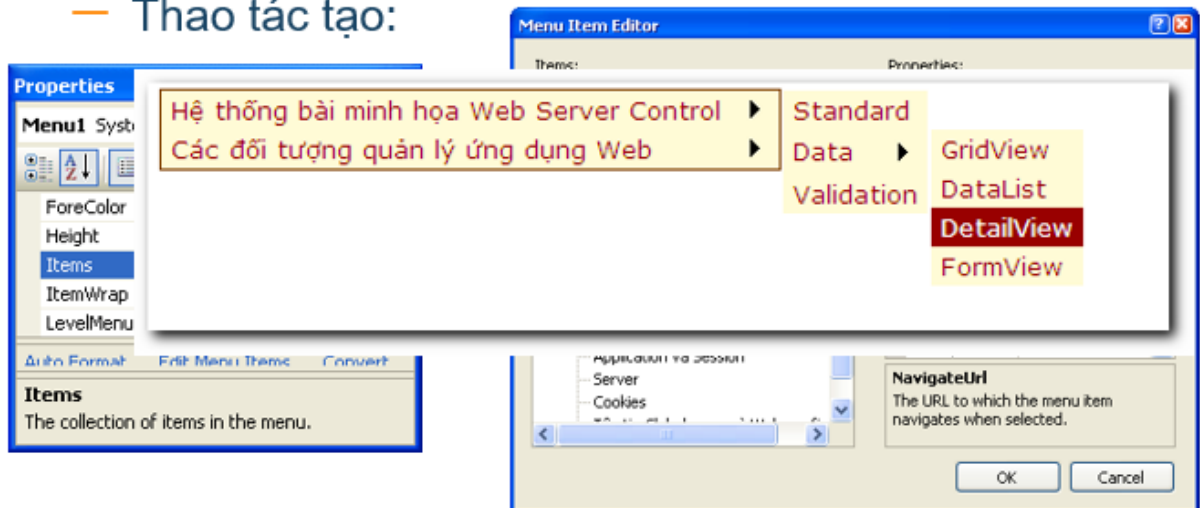
- Dùng để chuyển đến (navigate) một trang web khác và thể hiện cấu trúc logic của Web site (site map)
- Gồm các điều khiển: Menu, TreeView, SiteMapPath kết hợp với SiteMapDataSource
- Các điều khiển này thường được sử dụng trong trang Master

Menu

- + Dùng để thể hiện thực đơn và cho phép người dùng chuyển đến một trang web khác khi chọn một chức năng
- + Thao tác tạo: chọn thuộc tính Items: khai báo các chức năng trên thực đơn

• Menu

— Thao tác tạo:



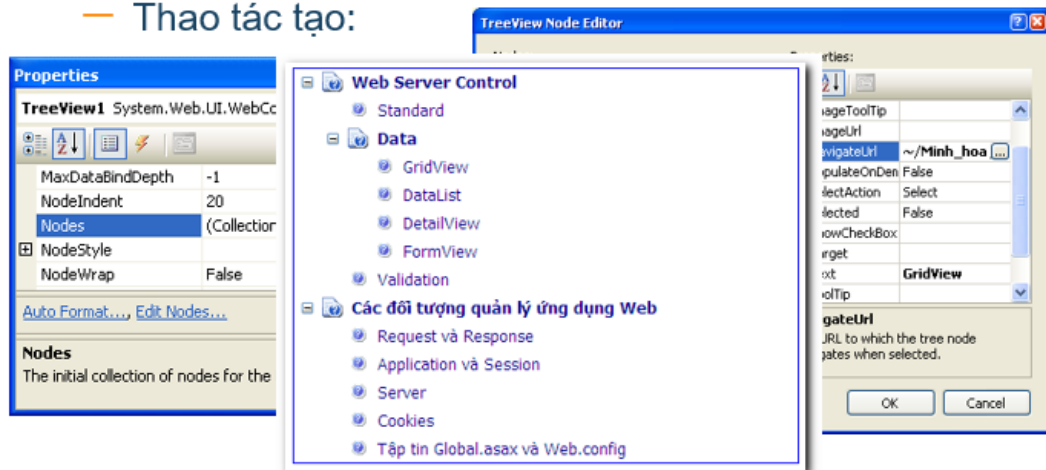
Hình 3.36 Site Navigation Controls

TreeView

- Dùng để thể hiện thực đơn có dạng cây và cho phép người dùng chuyển đến một trang web khác khi chọn một node
- Thao tác tạo: chọn thuộc tính Nodes: khai báo các chức năng trên thực đơn

• TreeView

— Thao tác tạo:



Hình 3.37 Site Navigation Controls

SiteMapPath

- Dùng để thể hiện cấu trúc logic của website và cho phép người dùng chuyển đến một trang web khác
- Nội dung cấu trúc của website mặc định được chứa trong tập tin Web.sitemap
- Nếu nội dung cấu trúc nằm trong tập tin ???sitemap (không phải Web.sitemap), thì phải định nghĩa SiteMapProvider trong tập tin Web.config

SiteMapDataSource

- Dùng để chứa nguồn dữ liệu được lưu trữ trong tập tin .sitemap
 - Nguồn dữ liệu mặc định là tập tin Web.sitemap
 - Thường sử dụng kết hợp với các điều khiển Menu và TreeView
 - Các thuộc tính:
 - + SiteMapProvider: tên SiteMapProvider đã khai báo trong tập tin Web.config.
- Nếu nguồn là Web.sitemap thì không cần khai báo thuộc tính này
- + ShowStartingNode(True/False): bắt đầu bằng siteMapNode gốc hay không
 - + SiteMapProvider: tên SiteMapProvider
 - + StartingNodeUrl: địa chỉ URL cho node bắt đầu
 - Liên kết với Menu và TreeView:
 - + Sử dụng thuộc tính DataSourceID của 2 điều khiển trên, giá trị của thuộc tính là ID của SiteMapDataSource

Câu hỏi ôn tập

1. Các em hãy cài đặt, cấu hình ngôn ngữ lập trình Web;
2. Nêu chi tiết từng công cụ ADO.Net;
3. Sử dụng DataView để kết nối với cơ sở dữ liệu;
4. Hoàn thiện Website động có nội dung sau:
 - Thiết kế giao diện phía người dùng và phía quản trị Web

Tài liệu cần tham khảo:

- [1]. Nguyễn Văn Lâm, Kỹ thuật xây dựng ứng dụng ASP.NET Tập 1,2,3, NXB Lao động và Xã hội 2008.
- [2]. Kỹ sư Phạm Quang Huy, Giáo trình thiết kế Web và làm hoạt hình, NXB Giao thông vận tải, năm 2003.
- [3]. Hoàng Hải, Giáo trình tự học thiết kế Web động, NXB Lao động Xã Hội, Tháng 9 năm 2007.