

TRƯỜNG CAO ĐẲNG NGHỀ CÔNG NGHIỆP HÀ NỘI

**Chủ biên: Trần Thị Ngân
Đồng tác giả: Bùi Quang Ngọc**



**GIÁO TRÌNH
THIẾT KẾ WEB**



Hà Nội – 2012

LỜI NÓI ĐẦU

Mạng Internet được coi như một tổ hợp các Website, người sử dụng có thể đi từ trang này sang trang khác, có thể tương tác với các trang Web để hiển thị những nội dung khác nhau tùy thuộc theo yêu cầu.

Hiện nay, hầu hết các tổ chức, công ty đều sử dụng Website để giới thiệu và quảng bá sản phẩm, dịch vụ của mình. Việc thiết kế Website là rất quan trọng, nó thể hiện thương hiệu và đẳng cấp của công ty và tổ chức. Vì thế, thiết kế Website đã trở thành công việc và là nghề được nhiều bạn trẻ ưa thích.

Giáo trình môn học “ **Thiết kế Web** ” được biên soạn dựa theo đề cương chi tiết môn học **Thiết kế Web** thuộc chương trình đào tạo nghề **Quản trị mạng** của trường Cao đẳng nghề Công nghiệp Hà nội, ban hành năm 2011. Giáo trình này cung cấp cho sinh viên ngành Công nghệ Thông tin những kiến thức căn bản về thiết kế web như : Các khái niệm cơ bản của thiết kế Web, quy trình thiết kế Web, các thẻ HTML cơ bản, các công cụ và phần mềm để thiết kế được giao diện Web, lập trình với ngôn ngữ ASP.NET kết hợp với cơ sở dữ liệu để tạo ra các trang Web động,....

Tuy đã có nhiều cố gắng trong quá trình biên soạn nhưng vẫn còn nhiều thiếu sót, chúng tôi mong nhận được các ý kiến đóng góp của các thầy cô, đồng nghiệp và các bạn sinh viên để chúng tôi có thể hoàn thiện giáo trình này hơn.

Hà Nội, ngày 5 tháng 12 năm 2012.

Tham gia biên soạn giáo trình

1. Chủ biên: Trần Thị Ngân
2. Đồng tác giả: Bùi Quang Ngọc

Tuyên bố bản quyền

Tài liệu này là loại giáo trình nội bộ dùng trong nhà trường với mục đích làm tài liệu giảng dạy cho giáo viên và học sinh, sinh viên nên các nguồn thông tin có thể được tham khảo.

Tài liệu phải do trường Cao đẳng nghề Công nghiệp Hà Nội in ấn và phát hành.

Việc sử dụng tài liệu này với mục đích thương mại hoặc khác với mục đích trên đều bị nghiêm cấm và bị coi là vi phạm bản quyền.

Trường Cao đẳng nghề Công nghiệp Hà Nội xin chân thành cảm ơn các thông tin giúp cho nhà trường bảo vệ bản quyền của mình.

Địa chỉ liên hệ:

Trường Cao đẳng nghề Công nghiệp Hà Nội.

131 – Thái Thịnh – Đống Đa – Hà Nội

Điện thoại: (84-4) 38532033

Fax: (84-4) 38533523

Website: www.hnivc.edu.vn

MỤC LỤC

	Trang
LỜI NÓI ĐẦU	1
TUYÊN BỐ BẢN QUYỀN	2
MỤC LỤC.....	3
BÀI 1 : TÔNG QUAN VỀ THIẾT KẾ WEB.....	14
1. Lịch sử WWW.....	14
2. Các khái niệm cơ bản	17
2.1. Web Server, Web Browse	17
2.2. Website, Web page, Static page, Dynamic page.....	17
2.3. Web Hosting, Upload web, Download web.....	20
2.4. Địa chỉ URL, địa chỉ URL tuyệt đối và địa chỉ URL tương đối..	21
2.4.1. Địa chỉ URL.....	21
2.4.2. URL tuyệt đối và URL tương đối.....	23
3. Thiết kế Web.	23
3.1. Khái niệm thiết kế Web.	23
3.2. Quy trình thiết kế web.....	24
3.2.1. Quy trình.	24
3.2.2. Xuất bản web.	25
3.3. Các lưu ý khi thiết kế web	26
3.3.1. Từ ngữ.....	26
3.3.2. Màu sắc.....	27
3.3.3. Hình ảnh.....	27
3.3.4. Bộ cục đơn giản, hợp lý.....	28
3.3.5. Thời gian tải.....	29
4. Nhập môn ngôn ngữ HTML.....	29

A. LÝ THUYẾT.....	29
4.1. Soạn thảo trang mã nguồn HTML.....	29
4.2. Các thẻ HTML(Tags).	31
4.3. Cấu trúc tổng quát tệp tin HTML	32
4.4. Thẻ BODY.....	33
4.5. Văn bản trên trang web.....	34
4.5.1. Vấn đề ngắt dòng.	34
4.5.2. Định dạng kiểu dáng – Style.....	36
4.5.3. Thay đổi Font, Size	36
4.5.6. Đường kẻ ngang và ký tự đặc biệt.	38
4.5.7. Tạo văn bản đè mục	39
4.5.8. Tạo danh sách cho các đoạn văn bản.....	40
4.5.9. Tạo hiệu ứng chuyển động chữ.....	44
4.6. Thiết lập liên kết cho trang web.	45
4.6.1. Khái quát.....	45
4.6.2. Liên kết cục bộ (Local Link).	45
4.6.3. Liên kết từ xa (Remote Link)	47
4.6.4. Dùng hình ảnh làm nhãn liên kết.....	47
4.6.5. Bản đồ ảnh dùng thẻ AREA.	48
4.6.6. Màu của liên kết.....	48
4.7. Multimedia trên trang web	48
4.7.1. Chèn hình ảnh	48
4.7.2. Liên kết đến tệp tin âm thanh.....	50
4.7.3. Chèn nhạc nền.....	51
4.7.4. Chèn điều khiển nhạc Plugin	51

4.7.3. Chèn video	52
4.8. Bảng biểu.....	52
4.8.1. Định nghĩa bảng – thẻ Table.....	53
4.8.2 Định nghĩa hàng-thẻ TR :	53
4.8.3. Thẻ định nghĩa ô- thẻ TD :.....	53
4.8.4. Thẻ định nghĩa tiêu đề cột - thẻ TH.....	53
B. THỰC HÀNH.....	55
1. Nội dung thực hành	55
2. Bảng vật tư thiết bị cần thiết.....	57
3. Quy trình thực hiện.....	57
4. Tổ chức thực hiện.	59
5. Kiểm tra đánh giá	59
CÂU HỎI VÀ BÀI TẬP CỦA BÀI 1	60
A. Trắc nghiệm.....	60
B. Bài tập thực hành.....	62
BÀI 2 : THIẾT KẾ WEB TĨNH VỚI PHẦN MỀM DREAMWEAVER..	69
A. LÝ THUYẾT	69
1. Tổng quan.....	69
1.1. Giới thiệu phần mềm Dreamweaver.....	69
1.2. Giao diện và các thanh công cụ cơ bản.	70
1.2.1. Khởi động Dreamweaver 8.....	70
1.2.2. Màn hình làm việc.	70
1.2.3. Các thanh công cụ.....	70
1.2.4. Mở một số bảng điều khiển quản lý các đối tượng	72
1.3. Làm việc với Website.....	72

1. 3.1. Khái quát.....	72
1.3.2. Tạo mới Site:.....	72
1.3.3. Quản lý site	74
2. Trang và văn bản trên trang.....	76
2.1. Các thao tác quản lý trang web.....	76
2.1.1. Tạo 1 trang web mới.	76
2.1.2. Mở một tập tin HTML đang tồn tại.	77
2.1.3. Đóng 1 trang tệp tin HTML.....	77
2.1.4. Lưu một trang tài liệu.	77
2.1.5. Hiển thị trang trên trình duyệt:	78
2.2. Thiết lập thuộc tính cho trang web	78
2.2.1. Thuộc tính chung cho trang: Lớp Appearance	78
2.2.2. Các thuộc tính chung cho liên kết : Lớp Link	79
2.2.3. Tạo mẫu văn bản để mục : Lớp Headings	80
2.2.4. Tiêu đề trang: Chọn mục Title-Encoding	80
2.3. Nhập văn bản	80
2.4. Định dạng văn bản.....	81
2.5. Tạo danh sách cho văn bản.....	82
2.5.1. Tạo danh sách mới	82
2.5.2. Chính sửa các thuộc tính của danh sách:	82
2.6. Chèn ký tự đặc biệt.....	83
3. Multimedia trên trang Web	83
3.1. Hình ảnh.....	83
3.1.1. Chèn hình ảnh	83
3.1.2. Định dạng hình ảnh.....	83

3.1.3. Giữ toạ độ vị trí cho ảnh	84
3.2. Tập tin Media (Audio, Video)	85
3.2.1. Khái quát về các dạng tập tin Multi Media:	85
3.2.2. Liên kết đến tập tin Media	85
3.2.3. Chèn nhạc nền.....	86
3.2.4. Nhúng tệp tin media vào trang web.....	86
4. Liên kết.....	89
4.1. Tạo liên kết đến trang tài liệu khác	89
4.2. Tạo liên kết đến một phần của trang tài liệu.	90
4.3. Liên kết Email	90
4.4. Bản đồ ảnh (hotspot).....	91
4.5. Tạo Rollover.....	92
4.6. Tạo liên kết Navigation Bar.....	93
4.6.1. Chèn thanh điều hướng:.....	93
4.6.2. Hiệu chỉnh thanh điều hướng:.....	94
4.7. Bổ sung nút Flash.	95
4.7.1. Tạo nút Flash.....	95
4.7.2. Chính sửa các đối tượng nút Flash:	96
5. Các yếu tố động trên trang	96
5.1. Chèn tệp tin flash.....	96
5.2. Chèn Java applet.....	97
5.3. Chèn Java Scrpit	97
5.3.1. Giới thiệu về Javascript:	97
5.3.2. Nhúng javascript vào trang web:	97
5.4. Chuỗi ký tự có hiệu ứng Flash.....	99

6. Bảng (Table).....	100
6.1. Sử dụng bảng ở chế độ Standard:	101
6.1.1. Chèn bảng	101
6.1.2. Nhập dữ liệu vào bảng	101
6.1.3. Định dạng bảng	103
6.1.4. Định dạng ô.....	103
6.1.5. Chèn thêm ô, hàng và cột vào bảng.....	104
6.2. Sử dụng bảng ở chế độ Layout View:	105
7. Form	105
7.1. Chèn Form.	106
7.2. Thêm các đối tượng lên Form	106
7.2.1. Text field.....	106
7.2.2. Textarea.....	107
7.2.3. Radio Group:.....	108
7.2.3. Checkbox:	109
7.2.4. List / Menu:	109
7.2.5. Jump menu:	110
7.2.6. File Field.	111
7.2.7. Submit và Reset.	111
7.2.8. Kiểm tra dữ liệu nhập trên Form.	112
8. Frame & Template.....	113
8.1. Frame.	113
8.1.1. Tạo frame.....	113
8.1.2. Thiết lập các thuộc tính cho Frame.....	114
8.1.3. Lưu các Frame và Frameset.....	115

8.1.4. Liên kết trong frameset	115
8.2. Template	116
8.2.1. Tạo Template.	116
8.2.2. Thiết kế tệp tin Template	117
8.2.3. Áp dụng Template cho một tài liệu.	118
8.2.4. Sửa, xoá, đổi tên tập tin Template	118
B. THỰC HÀNH.....	119
1. Nội dung thực hành	119
2. Bảng vật tư thiết bị cần thiết.....	121
3. Quy trình thực hiện.....	121
4. Tổ chức thực hiện	123
5. Kiểm tra đánh giá	123
CÂU HỎI VÀ BÀI TẬP BÀI 2	124
A. Câu hỏi trắc nghiệm.....	124
B. Bài tập	127
BÀI 3 : XÂY DỰNG WEB ĐỘNG.....	133
A. LÝ THUYẾT	133
1. Tổng quan về ASP.NET và ADO.NET.....	133
1.1. Tổng quan về ASP.NET	133
1.1.1. Giới thiệu về ASP.NET	133
1.1.2. Lịch sử phát triển	134
1.1.3. Những ưu điểm của ASP.NET	136
1.2. Tổng quan về ADO.NET.....	137
1.2.1. Giới thiệu	137
1.2.2. Data Provider	138

1.2.3. Các đối tượng của ADO.NET.....	139
1.3.Cài đặt và cấu hình IIS	141
1.3.1. Giới thiệu về IIS.....	141
1.3.2. Cài đặt IIS trên Windows 7.....	142
1.4. Tạo ứng dụng ASP.NET đầu tiên.....	143
2. WEB SERVER CONTROL.....	146
2.1. HTML Control.....	146
2.2. ASP.NET Server Control	148
2.2.1 ASP.NET Page.....	148
2.2.2. Các điệu khiển cơ bản.....	150
2.2.3. Label.....	151
2.2.4. Hyperlink	151
2.2.5. Textbox	152
2.2.6. Image.....	152
2.2.7. Button, ImageButton, LinkButton.....	152
2.2.8. Listbox và DropDownList	153
2.2.9. Checkbox và Radiobutton.....	156
2.2.10. CheckBoxList và RadioButtonList	157
3. Các đối tượng trong ASP.NET.....	158
3.1. Đối tượng Request	158
3.2. Response	159
3.3. Session	159
3.4. Cookie.....	162
4. Các đối tượng ADO.NET	165
4.1. Connection.....	165

4.2. Command.....	169
4.3. DataReader	172
4.4. DataAdapter.....	173
4.5. Đối tượng Dataset và DataTable	174
5. Xây dựng ứng dụng mẫu Website thương mại điện tử.	189
5.1. Thiết kế ứng dụng.....	189
5.2. Thiết kế cơ sở dữ liệu	192
5.2.1. Thiết kế bảng Department.....	192
5.2.2. Thiết kế bảng Category.....	192
5.2.3. Thiết kế bảng Product.....	193
5.2.4. Tạo bảng ProductCategory	195
5.2.5. Thiết kế bảng Orders.....	195
5.2.6. Thiết kế bảng OrderDetails.....	196
5.2.7. Thiết kế bảng ShoppingCart	196
5.3. Viết chương trình.....	197
5.3.1. File web.config	197
5.3.2. Xây dựng lớp GiftShopConfiguration	198
5.3.3. Xây dựng lớp truy cập cơ sở dữ liệu GenericDataAccess.cs	200
5.3.4. Xây dựng lớp CatalogAccess.cs	202
5.3.5. Tạo trang DepartmentList.ascx.....	211
5.3.6. Thiết kế trang hiển thị danh sách sản phẩm.....	216
5.3.7. Thiết kế menu hiển thị danh mục sản phẩm	226
5.3.8. Thiết kế trang tìm kiếm sản phẩm	231
5.4. Thiết kế giỏ hàng	244
5. 4.1. Thực hiện tầng dữ liệu	244

5.4.2. Tầng xử lý dữ liệu.....	247
5.4.3. Tầng xử lý giao diện người dùng.....	254
B. THỰC HÀNH.....	270
1. Nội dung thực hành	270
2. Bảng vật tư thiết bị cần thiết.....	270
3. Quy trình thực hiện công việc	270
4. Tổ chức thực hiện	271
5. Kiểm tra đánh giá	271
BÀI TẬP BÀI 3.....	272
TÀI LIỆU THAM KHẢO.....	277

MỤC TIÊU.

- Định hướng được cách thiết kế Website.
- Thiết kế được giao diện Website.
- Có khả năng sử dụng các thẻ HTML.
- Biết cách tổ chức thông tin trên trang chủ.
- Sử dụng thành thạo các công cụ thiết kế Web.
- Xây dựng được các ứng dụng Multimedia.
- Cài đặt, cấu hình được dịch vụ IIS.
- Lập trình cơ bản website, kết hợp với cơ sở dữ liệu để tạo ra các trang Web động.

NỘI DUNG CỦA MÔ ĐUN

Số TT	Tên các bài trong mô đun	Thời gian			
		Tổng số	Lý thuyết	Thực hành	Kiểm tra
1	Tổng quan về thiết kế web	12	6	5	1
2	Thiết kế web tĩnh với Dreamweaver.	48	16	30	2
3	Thiết kế web động	60	20	38	2
	Cộng	120	42	73	5

Phương pháp đánh giá

- Được đánh giá qua bài viết, kiểm tra vấn đáp hoặc trắc nghiệm, tự luận: nắm rõ các bước quy trình xây dựng Web, viết đúng cú pháp câu lệnh khi lập trình Web động.
- Dựa trên năng lực thực hành thiết kế được một ứng dụng Web tĩnh, xây dựng được một ứng dụng Web thông qua các ngôn ngữ lập trình, sử dụng được các đối tượng lập trình, liên kết được các Cơ sở dữ liệu.

BÀI 1 : TỔNG QUAN VỀ THIẾT KẾ WEB

Mục tiêu :

- Hiểu được lịch sử của WWW.
- Hiểu được các khái niệm cơ bản về 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.

1. Lịch sử WWW.

Để hiểu được lịch sử của WWW điều quan trọng là phân biệt được sự khác nhau giữa WWW và Internet. Internet là mạng của các mạng, kết nối hàng triệu máy tính trên toàn cầu lại với nhau, hình thành một hệ thống mà trong đó bất cứ máy tính nào cũng có thể liên lạc với các máy tính khác khi chúng đang cùng được kết nối vào Internet. Còn WWW (viết tắt từ World Wide Web - Web rộng khắp thế giới - gọi tắt là Web) là dịch vụ trao đổi, tìm kiếm và truy cập thông tin trên mạng theo giao thức HTTP (Hyper Text Transfer Protocol). WWW được sử dụng thông qua trình duyệt Web. Như vậy WWW chỉ là dịch vụ phổ biến nhất của Internet. Ngoài WWW, Internet còn có các dịch vụ khác như Email (thư điện tử), FTP (File Transfer Protocol - truyền tải tệp tin), CHAT (tán gẫu),...

Cha đẻ của WWW chính là viện sĩ Viện Hàn lâm Anh Tim Berners-Lee. Vào năm 1980, Tim Berners Lee khi đang làm kỹ sư phần mềm cho CERN (phòng thí nghiệm vật lý hạt của châu Âu tại Geneva), thấy các đồng nghiệp phải vất vả trong việc tra cứu tài liệu, Ông bắt đầu viết phần mềm Enquire (viết tắt của Enquire Within Upon Everything - tạm dịch là tra khảo mọi thứ). Với phần mềm này, Tim Berners-Lee tạo ra một loại sổ tay “siêu văn bản”: Các từ trong tài liệu được nối kết với các file khác trong máy tính của ông và ông có thể lần theo nối kết này bằng số (lúc này chưa tồn tại con chuột để nhập) và tự động truy ra tài liệu liên đới. Phần mềm này hoạt động hoàn hảo. Vấn đề đặt ra là ông muốn thực hiện điều tương tự trên các máy tính khác trên mạng. Năm 1989, cùng với Robert Cailliau (người Bỉ), Tim Berners Lee đã đề xuất dự án thiết kế hệ thống thông tin toàn cầu World Wide Web (WWW) dựa trên ý tưởng siêu văn bản để cho phép mọi người dễ dàng chia sẻ công việc của họ qua mạng máy tính. Mặc

dù ít được mọi người quan tâm đến, ông vẫn tiếp tục phát triển ba thành phần chính cho các trang web là HTTP, HTML và trình duyệt web. Mới đầu, Tim Berners Lee phát triển hệ thống siêu văn bản để theo dõi hàng trăm dự án, phần mềm, máy tính đang sử dụng của khoa Vật lý năng lượng cao của CERN. Dùng máy tính NeXT, ông làm nên một trình duyệt thô sơ vào mùa thu năm 1990 tên là World Wide Web (sau đổi tên là Nexus). Ngày 06 tháng 8 năm 1991, ông và Cailliau tạo ra được nội dung trang web đầu tiên trên thế giới là danh bạ điện thoại của các nhân viên trong CERN. Một bản copy của trang web vẫn còn tồn tại.

Web đã bùng nổ vào năm 1993 khi một sinh viên tốt nghiệp trường Đại học Illinois tên là Marc Andreessen phát triển thành công một phần mềm giúp hiển thị các trang Web dưới dạng đồ họa. Phần mềm này được gọi là Mosaic, một “trình duyệt” cho phép người sử dụng có thể hiển thị hình ảnh cũng như văn bản trên các trang web. Từ đó, web và Internet cùng hợp nhất nhau để phát triển với tốc độ chóng mặt. Chúng ta có thể điểm một số sự kiện tiêu biểu như sau:

- Ngày 30-4-1993: CERN công bố World Wide Web miễn phí cho tất cả mọi người. Tim Berners Lee đã thuyết phục được Cern “cho không” công nghệ web và các mã chương trình để mọi người đều có thể sử dụng và cùng tham gia phát triển. Đây là một trong những nguyên nhân chủ yếu giúp cho web phát triển nhanh chóng và mạnh mẽ.
- Tháng 5-1993: Viện công nghệ Massachusetts lần đầu tiên đưa một tờ báo lên web.
- Tháng 6-1993: Ngôn ngữ HTML (Hypertext Mark Language) dùng trong lập trình web được công bố.
- Tháng 11-1993: Webcam đầu tiên được kết nối.
- Tháng 2-1994: Tiền thân của Yahoo được đưa lên internet.
- Tháng 4-1994: BBC mở website đầu tiên cho chương trình Ti vi: The Net
- Ngày 13-10-1994: Bill Clinton đưa Nhà Trắng lên web
- Tháng 2-1995 : Radio HK chính thức trở thành một đài phát thanh trực tuyến “fulltime”.
- Ngày 1-7-1995: Hiệu sách trực tuyến Amazon khai trương.
- Ngày 9-8-1995: “Bùng nổ tên miền .com”. Hàng loạt công ty trực tuyến chính thức mở cửa website của mình.

- Ngày 24-8-1995: Microsoft Internet Explorer (IE) được phát hành và là một phần trong Windows 95.
 - Ngày 4-9-1995: EBay, website đấu giá trực tuyến đầu tiên được thành lập.
 - Ngày 15-12-1995: Alta Vista, công cụ tìm kiếm đa ngôn ngữ đầu tiên ra mắt người sử dụng Internet.
 - Ngày 4-7-1996: Hotmail khai trương.
 - Ngày 17-12-1997: Jorn Barger đưa ra khái niệm weblog, sau này rút gọn thành blog.
 - Tháng 9-1998: Gã khổng lồ Google mở cửa văn phòng đầu tiên của mình.
 - Ngày 16-8-1999: Everquest trở thành game nhập vai trực tuyến nhiều người chơi (MMORPG) đầu tiên.
 - Ngày 19-8-1999: MySpace khai trương. Đây là một trong những website đầu tiên về lưu trữ trực tuyến.
 - Tháng 11-1999: Boo.com trở thành site đầu tiên kinh doanh thời trang.
 - Ngày 2-7-2000: Hacker đánh sập 8 website lớn trong đó có Yahoo, CNN và Amazon.
 - Ngày 15-1-2001: Jimmy Wales sáng lập Bách khoa toàn thư trực tuyến Wikipedia.
 - Ngày 22-11-2001: Giáo hoàng John Paul II gửi bức thư điện tử đầu tiên từ chiếc laptop trong phòng làm việc của ông.
 - Ngày 11-12-2002: FBI bắt đầu đăng tin truy nã tội phạm trực tuyến.
 - Ngày 28-4-2003: Apple đưa dịch vụ tải nhạc iTunes vào hoạt động.
 - Tháng 7-2004: Tim Berners Lee được phong tước hiệp sĩ.
 - Ngày 4-2-2004: Zuckerberg thành mạng xã hội Facebook.
 - Ngày 9-11-2004: Mozilla Firefox ra mắt công chúng
 - Tháng 2-2005: Website chia sẻ video youtube.com ra đời
-

Dù rằng trước đây, Web toàn cầu chỉ có thể hỗ trợ cho dạng văn bản và các đồ họa cơ bản thì ngày nay, Web đã có khả năng hỗ trợ cho một số lượng đáng kinh ngạc các loại phương tiện khác nhau. Nhiều trang Web hiện nay chứa đựng cả các hình động, âm thanh, phim ảnh và các ứng dụng vận hành như các phần mềm vi tính thông thường. Thông thường, các trình

duyệt Web như Netscape Navigator và Microsoft Internet Explorer có hỗ trợ cho cả các nội dung này, nhưng trong một số trường hợp, bạn sẽ phải tải về máy phần mềm đặc biệt gọi là plug-in để giúp trình duyệt nhận dạng được chúng.

Ngày nay, với nhiều tỷ trang Web và ngày càng sinh sôi nảy nở, Web đã trở thành một vũ trụ thông tin có thể truy xuất qua mạng, một hiện thân của tri thức nhân loại, và một kho tri thức của loài người.

2. Các khái niệm cơ bản

2.1. Web Server, Web Browse

Hệ thống web là một phần của Internet bao gồm nhiều máy Web Server (máy chủ web) trên khắp thế giới. Các Web Server này lưu trữ các trang web, cho phép mà người dùng ở bất kỳ đâu cũng có thể truy cập.

Web Server cũng chỉ là một máy tính được nối vào Internet. Nhưng nó được kết nối Internet liên tục (24/24 giờ) và cài đặt các hệ điều hành, các phần mềm chuyên dụng (Apache, Internet Information Services) đáp ứng các nhu cầu truy cập web từ phía người sử dụng. Web Server thường là máy tính có cấu hình cao, khả năng hoạt động ổn định, tốc độ đường truyền cao để có thể đáp ứng nhiều yêu cầu truy cập đồng thời từ nhiều người sử dụng.

Web Browser còn gọi là trình duyệt web, được sử dụng bởi người dùng trên máy khách. Nhiệm vụ của trình duyệt web là khai thác các dịch vụ web. Người dùng yêu cầu một trang tài liệu bằng cách gõ một URL(Uniform Resource Location: Địa chỉ truy cập của trang Web) vào khung Address của trình duyệt. Trình duyệt gửi thông tin này tới Web Server. Web Server sử dụng thông tin trong URL để xác định và cho hiển thị trang Web theo yêu cầu của Web Browser. Nếu là các trang web động, Web Server sẽ xử lý các thông tin và trả kết quả về cho người dùng. Các trình duyệt phổ biến hiện nay là (Internet Explorer), Netscape, Opera, Google Chrome, Firefox,...

2.2. Website, Web page, Static page, Dynamic page.

Ngày nay người ta dùng máy tính như một công cụ rất hữu ích để truy nhập Internet, chủ yếu là tìm kiếm thông tin. Vấn đề đặt ra là làm thế nào để mọi người có thể dễ dàng sử dụng máy tính như một công cụ phục vụ đắc lực cho việc tra cứu tìm kiếm thông tin trên mạng Internet. Việc này trở

nên dễ dàng hơn bởi ý tưởng siêu văn bản (Hypertext). Siêu văn bản là các văn bản “thông minh” giúp người xem có thể tìm các tài liệu liên quan. Trong thế giới siêu văn bản, nội dung thông tin không còn bị bó hẹp trong một trang văn bản mà nó còn được mở rộng bằng cách tham khảo, liên kết tới nhiều tài liệu khác (hình ảnh, âm thanh, hoạt họa...). Người xem có thể dễ dàng đi từ tài liệu này sang tài liệu khác thông qua các mối liên kết. Họ có thể du lịch trong xa lộ thông tin phong phú trong khi vẫn ngồi tại nhà. Họ không phải sang tận Louvre ở Paris mà vẫn có thể chiêm ngưỡng được các kiệt tác hội họa. Chính nó đã góp phần tạo ra bước phát triển bùng nổ của Internet trong những năm gần đây.

Để phổ biến thông tin trên toàn cầu, cần một ngôn ngữ phổ biến và dễ hiểu mà toàn bộ các máy tính có thể hiểu được. Ngôn ngữ phổ biến dùng bởi WWW là HTML (Hyper Text Markup Language - ngôn ngữ đánh dấu siêu văn bản). Các trang Web đầy sinh động mà bạn thấy trên World Wide Web là các trang siêu văn bản được viết bằng ngôn ngữ đánh dấu siêu văn bản hay HTML. HTML cho phép bạn tạo ra các trang web phối hợp hài hòa văn bản thông thường với hình ảnh, âm thanh, video và các mối liên kết đến các trang siêu văn bản khác...

Tên gọi ngôn ngữ đánh dấu siêu văn bản phản ánh đúng thực chất của công cụ này:

- **Đánh dấu (Markup):** HTML là ngôn ngữ của các thẻ đánh dấu – Tag. Các thẻ này xác định cách thức trình bày đoạn văn bản tương ứng trên màn hình.

- **Ngôn ngữ (Language):** HTML là một ngôn ngữ tương tự như các ngôn ngữ lập trình, tuy nhiên đơn giản hơn. Nó có cú pháp chặt chẽ để viết các lệnh thực hiện việc trình diễn văn bản. Các từ khoá có ý nghĩa xác định được cộng đồng Internet thừa nhận và sử dụng. Ví dụ `b = bold`, `ul = unordered list`, ...

- **Văn bản (Text):** HTML đầu tiên và trước hết là để trình bày văn bản và dựa trên nền tảng là một văn bản. Các thành phần khác như hình ảnh, âm thanh, hoạt hình.. đều phải "cốm neo" vào một đoạn văn bản nào đó.

- **Siêu văn bản (Hyper):** HTML cho phép liên kết nhiều trang văn bản rải rác khắp nơi trên Internet. Nó có tác dụng che giấu sự phức tạp của Internet đối với người sử dụng. Người dùng Internet có thể đọc văn bản

mà không cần biết đến văn bản đó nằm ở đâu, hệ thống được xây dựng phức tạp như thế nào. HTML thực sự đã vượt ra ngoài khuôn khổ khái niệm văn bản cổ điển.

Trang mã nguồn HTML là một tệp văn bản bình thường gồm các kí tự ASCII, có thể được tạo ra bằng bất cứ trình soạn thảo thông thường nào. Theo quy ước, tất cả các tệp mã nguồn của trang siêu văn bản phải có đuôi là “.html” hoặc “.htm”. Khi trình duyệt Web đọc trang mã nguồn HTML, nó sẽ dịch các thẻ lệnh và hiển thị lên màn hình máy tính thì ta thường gọi là trang Web (Web Page). Vậy trang Web là cái thể hiện của trang mã nguồn qua việc xử lý của trình duyệt. Như sau này ta sẽ thấy, các trình duyệt khác nhau có thể hiển thị cùng một trang mã nguồn không hoàn toàn giống nhau. Tạo trang Web tức là tạo ra trang mã nguồn HTML đúng quy định để trình duyệt web hiểu được và hiển thị đúng.

Để tạo được trang web ta có thể sử dụng những công cụ (phần mềm) thô sơ nhất như Notepad của Windows hoặc những công cụ cao cấp dành riêng cho việc thiết kế web như Dreamweaver, FrontPage, Flash, ASP, PHP...).

Web tĩnh (Static Page) : Là trang web đã được chuẩn bị sẵn cho các yêu cầu nhất định và nội dung của nó ít thay đổi. Web tĩnh có tính chất sau:

- Dễ phát triển.
- Tương tác yếu.
- Sử dụng HTML, do đó người làm Web tĩnh thường dùng các công cụ trực quan để tạo ra trang Web.
- Để cập nhật nội dung, cần phải được thực hiện bởi người có kiến thức về HTML và không sửa chữa trực tiếp trên trình duyệt Web.

Web động (Dynamic Page): Là trang web có các tính chất sau:

- Khó phát triển hơn Web tĩnh.
- Tương tác mạnh.
- Thiết kế bằng việc kết hợp HTML và mã lệnh. Thông thường trang web động sử dụng các ngôn ngữ lập trình như ASP, ASP.NET, PHP, JSP... để kết nối với các cơ sở dữ liệu (database) như Microsoft-Access, SQL Server, MySQL,... Mã lệnh được thực thi trên Server và trả về kết quả HTML cho người sử dụng thông qua các trình duyệt Web.
- Có thể cập nhật dữ liệu thường xuyên, dễ dàng thông qua trình

duyệt web. Người sử dụng chỉ cần biết sử dụng vi tính văn phòng và biết truy cập Internet là có thể thêm / sửa / xoá các thông tin được cho phép.

Ví dụ: Các trang mua/bán trực tuyến, khách hàng vào web để xem giới thiệu hàng hoá, khi đã ưng ý và muốn mua thì khách hàng chỉ việc chọn mặt hàng đó và điền một vài thông tin do người quản trị web quy định. Khi đã Kích nút đồng ý mua thì hệ thống sẽ ghi nhận dữ liệu về khách hàng và tự động tính toán tiền cho đơn hàng của khách hàng hàng,

Website gồm một tập các trang web thuộc một chủ thể (cá nhân hoặc tổ chức), có nội dung thống nhất, được nối với nhau bằng những siêu liên kết (hyperlink). Các Website thường có trang đầu tiên là trang chủ (home page) liệt kê các nội dung chính của Website và chứa các liên kết đến với các trang khác ở trong hay ngoài Website.

2.3. Web Hosting, Upload web, Download web.

Web Hosting (còn gọi tắt là Hosting) là không gian lưu trữ website trên máy chủ Web của nhà cung cấp dịch vụ Internet (Internet Service Provider – ISP) hoặc các đơn vị chuyên dụng. Lý do bạn phải thuê Web Hosting để chứa website là những máy tính đó luôn có một địa chỉ IP cố định khi kết nối vào Internet. Nếu bạn truy cập vào Internet thông qua các ISP thì địa chỉ IP trên máy bạn luôn bị thay đổi, do đó dữ liệu trên máy của bạn không thể truy cập được từ những máy khác trên Internet.

Như vậy, có thể hình dung như sau: Nếu xem website là ngôi nhà, là trụ sở doanh nghiệp, trên Internet, vậy thì hosting chính là mảnh đất, là mặt bằng để xây dựng trụ sở doanh nghiệp trên Internet và tên miền sẽ là biển hiệu, địa chỉ để mọi người nhớ đến và tìm kiếm.

Tùy theo nhà cung cấp dịch vụ Hosting mà chất lượng cũng như phí dịch vụ có các mức khác nhau. Khi đăng ký một Hosting chúng ta cần chú ý đến các yếu tố cơ bản sau :

- Tốc độ truy cập Internet tại máy chủ Hosting đó.
- Dung lượng Hosting: Kích thước để lưu cơ sở trữ dữ liệu của website (hình ảnh, thông tin ...). Đơn vị đo dung lượng thường là Mb hoặc Gb.
 - Các tiện ích và dịch vụ hỗ trợ như : các hệ quản trị cơ sở dữ liệu, ngôn ngữ lập trình,...
 - Băng thông hay dung lượng đường truyền: Tổng số dữ liệu tối đa

mà bạn được phép trao đổi qua lại giữa website và người sử dụng trong một đơn vị thời gian (đơn vị đo thông thường là Mb/tháng).

Upload web là việc gửi dữ liệu từ máy tính của người sử dụng lên một Web hosting còn **Download web** là việc bạn tải dữ liệu từ Web hosting về máy tính của người sử dụng.

2.4. Địa chỉ URL, địa chỉ URL tuyệt đối và địa chỉ URL tương đối.

2.4.1. Địa chỉ URL.

Mọi tài nguyên sẵn dùng trên WWW – tài liệu HTML, ảnh, video clip, chương trình,... đều có một địa chỉ URL (Uniform Resource Locator - tạm dịch là định vị tài nguyên về mặt địa chỉ). Về cơ bản, URL là con trỏ chỉ tới vị trí của một đối tượng. URL mang lại khả năng siêu liên kết cho các trang mạng. Địa chỉ URL của một đối tượng thường được gõ vào vùng Address của các trình duyệt web...

Một URL đầy đủ sẽ tuân theo cú pháp sau:

Scheme://host.domain:port/path/filename

Một URL gồm có nhiều phần được liệt kê dưới đây:

- Giao thức (Scheme): định nghĩa loại giao thức. Ví dụ là http, ftp, news, mailto,... nhưng hầu hết là http. Do đó nếu không gõ, sẽ mặc định là http.

- Tên miền (Domain name) : Địa chỉ website.
- Cổng (Port): Định nghĩa mã số cổng của webhost. Mã số cổng thường bị bỏ qua.Cổng mặc định cho http là 80.
- Đường dẫn (Path) : tuyệt đối trên máy phục vụ của tài nguyên (ví dụ: thumuc/trang) định nghĩa đường dẫn trên WebServer. Nếu đường dẫn bị bỏ qua, nguồn tài liệu phải được lưu trên thư mục chính (root) của website.

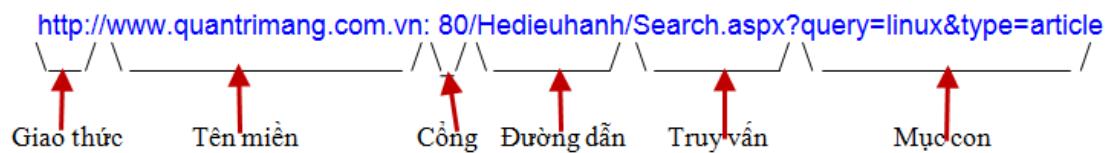
- + Các truy vấn (có thể không cần): chứa từ khoá tìm kiếm tài liệu.

- + Chỉ định mục con (có thể không cần): tên file của tài liệu. Tên file mặc định có thể là default.asp, hoặc index.php hoặc tên khác tùy vào bạn cài đặt máy chủ như thế nào.

- Dấu // chỉ ra rằng tiếp theo sẽ là tên miền và dấu /cho thấy rằng tiếp theo sẽ là tên thư mục hoặc tập tin.

Ví dụ: một địa chỉ web đơn giản như : hnivc.edu.vn sẽ truy cập tới trang chủ của website trường Cao đẳng nghề Công nghiệp Hà Nội hoặc một

địa chỉ đầy đủ của trang web về hệ điều hành Linux, là kết quả tìm kiếm trên website quantrimang.com như sau:



Hình 1.1 : Cấu trúc địa chỉ URL

❖ **Tên miền (Domain Name).**

Tên miền là sự nhận dạng vị trí của một máy tính trên mạng Internet nói cách khác tên miền là tên của các mạng lưới, tên của các máy chủ trên mạng Internet. Mục đích của tên miền là cho người sử dụng kết nối Internet tìm một địa chỉ Website hay gửi đến một địa chỉ mail với tên thân thiện dễ nhớ.

Tên miền là địa chỉ duy nhất của website trên mạng internet. Tên miền bao gồm nhiều phần cấu tạo nên, cách nhau bởi dấu chấm (.). Ví dụ home.vnn.vn là tên miền máy chủ web của VDC. Thành phần thứ nhất "home" là tên của máy chủ , thành phần thứ hai "vnn" thường gọi là tên miền tên miền thứ cấp (tên miền cấp 2) và thành phần cuối cùng "vn" là tên miền mức cao nhất (tên miền cấp 1).

Tên miền mức cao nhất là tên miền bạn đăng ký trực tiếp với các nhà cung cấp tên miền, nó bao gồm các mã quốc gia của các nước tham gia Internet được quy định bằng hai chữ cái theo tiêu chuẩn ISO -3166 như Việt nam là VN, Anh quốc là UK v.v.. và 7 lĩnh vực dùng chung (World Wide Generic Domains) như : COM(Commercial - Thương mại), EDU (Education - Giáo dục), NET (Network - mạng), INT (International Organisations : Các tổ chức quốc tế), GOV (Government - Nhà nước), MIL (Military : Quân sự) và ORG (Other orgnizations - Các tổ chức khác).

Ví dụ: www.yahoo.com, www.hnivc.edu.com được coi là các tên miền cấp 1. Các tên miền cấp 1 thể hiện sự chuyên nghiệp và uy tín trong hoạt động, kinh doanh trên Internet của các tổ chức.

Tên miền cấp 2 là tên miền phải phụ thuộc vào một tên miền cấp 1. Để đăng ký các tên miền kiểu này, thông thường bạn phải liên hệ trực tiếp với người quản lý tên miền cấp cao nhất.

Ví dụ: khi bạn đăng ký với một hosting miễn phí như Byethost.com, bạn sẽ nhận được tên miền cấp 2 có dạng www.tênbandedt.Byethost.com để upload website của mình lên để chia sẻ với bạn bè.

2.4.2. URL tuyệt đối và URL tương đối.

Địa chỉ tuyệt đối (Absolute link) chỉ ra đường dẫn đầy đủ của tài liệu. Đó là đường dẫn phải bắt đầu từ gốc của nguồn lưu trữ. Đường dẫn tuyệt đối tạo ra mối quan hệ chặt chẽ giữa trang web đặt liên kết và đích của liên kết. Còn ngược lại, địa chỉ tương đối (Relative link) chỉ mõi ra quan hệ tương quan giữa trang đặt liên kết với đích của liên kết.

Ví dụ: <http://vietnamnet.vn/Tinkinhte/quocte/tin07.html>

Các URL tương đối có thể gồm các thành phần đường dẫn tương đối như “..” (ký hiệu để chỉ thư mục cấp trên thư mục chứa tập tin chính) và có thể bao gồm các dấu hiệu đoạn.

Ví dụ : ta có trang web đang xét là intro.html có địa chỉ URL tuyệt đối như sau: “<http://www.acme.com/support/intro.html>”.

URL tương đối trong để tham chiếu đến một trang khác là :

` Suppliers `

sẽ mở rộng thành URL đầy đủ

`“http://www.acme.com/support/suppliers.html”`

trong khi URL tương đối trong việc đánh dấu cho một ảnh dưới đây

``

sẽ mở rộng thành URL đầy đủ

`“http://www.acme.com/icons/logo.gif”`

Chúng ta thường dùng địa chỉ tuyệt đối khi liên kết đến một tài liệu thuộc ngoài phạm vi của website và dùng địa chỉ tương đối để liên kết đến một tài liệu trong cùng website. Đường dẫn tương đối có ưu điểm hơn đường dẫn tuyệt đối ở chỗ : sẽ giúp cho trang web bớt nặng nề, load nhanh hơn và đặc biệt tránh bị lỗi đường dẫn khi đưa site lên mạng hoặc copy đến các vị trí khác.

3. Thiết kế Web.

3.1. Khái niệm thiết kế Web.

Thiết kế Web hay thiết kế Website là hoạt động trải rộng từ khâu thu thập ý tưởng, yêu cầu của khách hàng tới việc dựng đề án, bản thảo, tiến hành thiết kế giao diện đồ họa, tổ chức thực hiện thiết lập, xây dựng cơ sở

dữ liệu, lập trình các tính năng cho trang web và cuối cùng, đưa trang web vào chạy thử nghiệm, chỉnh sửa và xuất bản web để hoạt động chính thức. Giai đoạn tiếp sau khi trang web đã đi vào hoạt động là giai đoạn bảo hành, nâng cấp và cập nhật dữ liệu cho toàn bộ hệ thống web.

3.2. Quy trình thiết kế web.

3.2.1. Quy trình.

Trước khi thiết kế một Website, chúng ta cần thiết phải lập một bản kế hoạch rõ ràng, chi tiết để có thể lường trước được độ phức tạp, khó khăn của Website cũng như hạn chế việc phải thay đổi lại cấu trúc, thành phần Website sau này và nên tuân thủ một cách tuần tự qui trình sau :

Lập kế hoạch và xác định chiến lược phát triển → Thiết kế và lập trình Website → Kiểm tra → Lưu trữ và xuất bản web → Quảng bá và phát triển nội dung → Theo dõi, đánh giá và bảo trì website.

- Lập kế hoạch và xác định chiến lược phát triển : Phân tích yêu cầu, xác định mục tiêu, phạm vi nội dung của website, công nghệ hỗ trợ, tính năng tương tác, nhu cầu quảng bá,...

- Thiết kế và lập trình Website : Xây dựng cấu trúc nội dung của Website, các vấn đề cần đề cập trong từng nội dung. Chuẩn bị hình ảnh, video, nhạc,... để minh họa cho phần lời và phải lựa chọn thật cẩn thận để chúng phải chứa đựng thông tin, đẹp và ấn tượng. Thiết kế các giao diện, các cơ sở dữ liệu và lập trình kết hợp với cơ sở dữ liệu để tạo ra các trang web tĩnh có giao diện đẹp, nội dung phong phú và các trang web động có tính tương tác cao với người xem.

- Kiểm tra: Việc kiểm tra kỹ lưỡng trước khi thực hiện việc upload bao gồm : Kiểm tra nội dung, kiểm tra chính tả, kiểm tra tốc độ, kiểm tra các liên kết, thử các lỗi bảo mật, kiểm tra trên nhiều trình duyệt, với mục đích để loại bỏ những thành phần rườm rà, những lỗi sẽ bị mắc phải nhằm hoàn thiện sản phẩm trước khi công bố rộng rãi.

- Xuất bản: Đăng ký tên miền và thuê dịch vụ hosting sau đó upload website lên mạng. Dựa vào hiện trạng và mục tiêu phát triển website để lựa chọn ISP thích hợp.

- Quảng bá và phát triển nội dung : Đăng ký Website vào các máy tìm kiếm (search engine) trong nước và thế giới. Nâng tầm phát triển Website bằng cách tự động hóa dần các chức năng của Website.

- Theo dõi, đánh giá và bảo trì site: Dựa vào các thông tin truy cập thu thập được từ ISP mà ta có thể quyết định tăng cường nội dung cho những phần thường được truy cập và giảm bớt những phần ít được truy cập. Đồng thời còn phải quan tâm đến việc cập nhật nội dung và nâng cao tính thẩm mỹ của giao diện.

Trong các bước ở những quy trình trên, quan trọng nhất là giai đoạn Thiết kế và lập trình Website và giai đoạn Xuất bản Web. Việc thiết kế các trang Web tĩnh sẽ được giới thiệu chi tiết ở bài 2, các trang web động ở bài 3. Sau đây chúng ta tìm hiểu kỹ hơn về quá trình xuất bản một Website.

3.2.2. Xuất bản web.

Giống như một cửa hàng đặt trong một trung tâm thương mại, Website của bạn cũng cần nằm trên một máy chủ Web (WebServer) để tất cả mọi người đều truy cập được. Thực chất của việc xuất bản Website là: đăng ký cho website một tên miền, đăng ký với một nhà cung cấp dịch vụ Web (ISP) để có một Host và sau đó upload thư mục chứa website lên đó. Một số ISP cũng cung cấp các tên miền cấp hai của họ cho người đăng ký Host hệ thống của họ. Việc đăng ký tên miền, đăng ký Host có thể miễn phí hoặc có phí và mức phí tuỳ theo mức độ yêu cầu cung cấp các dịch vụ.

Có 3 cách để tiến hành Upload Website lên máy chủ Web Hostting.

- Upload thông qua trang web của nhà cung cấp dịch vụ (thông qua Browse).

- Upload thông qua các chương trình truyền file.
- Upload ngay trên môi trường thiết kế web.

❖ **Upload thông qua trang web của nhà cung cấp dịch vụ.**

Cách upload này thường chỉ được áp dụng với những host miễn phí. Upload theo cách này rất bất tiện do ta chỉ có thể tải (upload) lên Server lần lượt một hoặc một vài file và còn phải mất công tạo cấu trúc thư mục của site sao cho giống với máy cục bộ. Nó chỉ thích hợp cho những Website cá nhân.

❖ **Upload thông qua các chương trình truyền file.**

Một dạng upload khác là thông qua các chương trình truyền file sử dụng giao thức FTP. Chương trình thường bao gồm hai cửa sổ: một cửa sổ biểu diễn thư mục trên máy còn một cửa sổ biểu diễn thư mục trên Server. Ta chỉ cần tiến hành chọn các file cần tải hoặc cả Website sau đó chương trình sẽ chịu trách nhiệm tải chúng lên. Ngoài ra ta còn có thể thay đổi

thuộc tính của các file cũng như cập nhật các file một cách dễ dàng. Upload web theo cách này được thực hiện rất nhanh Tuy nhiên cách này thường sử dụng cho những Host phải trả tiền.

Chương trình truyền file phổ biến nhất hiện nay là CuteFTP, FlashFX,... Để có thể upload bằng phương pháp này bắt buộc Server phải cung cấp dịch vụ truyền file FTP.

❖ ***Upload ngay trên môi trường thiết kế web:***

Các phổ biến nhất để upload Website là dùng ngay tính năng upload Web trên các phần mềm thiết kế web. Với phương pháp này ta có thể đưa cả Website lên Server theo đúng cấu trúc như ở trên máy cục bộ. Quá trình cập nhật cũng thường được tiến hành tự động, chương trình sẽ tự dò xem file nào được cập nhật, file nào mới được thêm vào rồi chỉ tiến hành cập nhật hoặc thêm vào các file đó trên Server thôi. Phương pháp này cũng đòi hỏi server phải cung cấp dịch vụ truyền file FTP.

3.3. Các lưu ý khi thiết kế web

Một website muốn thu hút được người xem đồng thời giữ được sự quan tâm của những người này phải hấp dẫn và dễ sử dụng. Điều này nghe có vẻ đơn giản nhưng thực tế, chúng ta rất hay gặp những trang web thiết kế xấu, màu sắc loè loẹt và nội dung nghèo nàn.

Nguyên tắc tạo ra một Website hay rất đơn giản. Nếu mục đích của Website là để truyền đạt thông tin, thì yếu tố cần được xem trọng nhất là sự định hướng và cách trình bày nội dung sao cho thật khéo léo. Sau đây là những lưu ý khi thiết kế Web.

3.3.1. Từ ngữ.

Khi gõ nội dung trang web, bạn phải sử dụng font chữ chuẩn. Đối với các văn bản tiếng việt phải dùng bộ gõ Unicode với các font chữ truyền thống. Những trình duyệt khác nhau sẽ hỗ trợ những font chữ khác nhau. Do đó, những font chữ bất bình thường mà bạn cho là đẹp trên web site của mình lại có thể trở thành font chữ phức tạp không thể đọc được trên màn hình của người sử dụng. Có một số font chữ (được gọi là font chữ “dịch thực”) được tạo ra để thích hợp với mọi trình duyệt trong đó tiêu biểu nhất là Times New Roman, Arial, Helvetica và Verdana.

Các nội dung văn bản trên web phải viết đúng ngữ pháp và chính tả đồng thời phải sử dụng những từ ngữ dễ hiểu phù hợp với đối tượng người xem.

3.3.2. Màu sắc.

Hãy sử dụng màu sắc một cách có chừng mực trong khi thiết kế web. Quá nhiều màu sẽ làm cho Website trông rộn rạc và quá loè loẹt. Chỉ nên phối hợp 2 hoặc 3 màu là đủ để tạo ấn tượng. Nên sử dụng màu nền và màu font chữ có sự tương phản cao. Không nên sử dụng những gam màu chói để tạo nền hay tạo font chữ vì những màu này sẽ làm người đọc có cảm giác mắt khó chịu. Không nên sử dụng những hình nền có quá nhiều màu sắc, có quá nhiều chi tiết. Thường là những nền màu quá lòe loẹt sẽ làm cho chữ khó đọc hơn, ví dụ như nền màu xanh sáng kết hợp với chữ vàng hoặc nền đen với chữ xanh sáng đều gần như là không đọc được.

Nên phối màu trang trí trang web một cách hài hoà theo một “Tone” màu nhất định phù hợp với nội dung và hình ảnh của Website.

3.3.3. Hình ảnh.

Hình ảnh là thành phần quan trọng góp phần trang trí trang web thêm hấp dẫn và linh động. Tuy nhiên, nếu được sử dụng quá nhiều, chúng sẽ làm lu mờ nội dung và còn khiến tốc độ truy cập chậm hơn. Do đó hãy sử dụng hình ảnh như một phương pháp bổ sung và tăng giá trị cho nội dung chứ không nên dùng chúng để thay thế thông tin. Vì vậy cần cân nhắc khi cần chèn hình ảnh theo các nguyên tắc sau:

- Màu sắc phải phù hợp với Tone màu chung của trang Web.
- Đừng lạm dụng quá nhiều hình ảnh trên cùng 1 trang.
- Không sử dụng ảnh có Kích thước quá lớn(nếu cần nên tạo liên kết phóng to).
 - Độ phân giải ảnh thông thường để 72dpi .
 - Chọn định dạng ảnh phù hợp theo dạng thức của từng loại ảnh:
 - + Jpeg – Jpg : Ảnh tĩnh, hỗ trợ 16 triệu màu, phù hợp với các ảnh
 - + Gif : Ảnh tĩnh hoặc động hỗ trợ 256 màu.
 - + Swf (Macromedi Flash): Ảnh động vector , hỗ trợ 16 triệu màu tuy nhiên để xem máy tính phải cài Flash Plug in.
 - Ảnh phải lưu trữ trong một thư mục của Website, khi upload trang web, nhớ upload cả ảnh.
 - Tên của các file nên sử dụng chữ thường, không nên sử dụng chữ hoa. Ví dụ: myphoto.jpg chứ không nên là Myphoto.jpg.
 - Không nên dùng các ảnh động đặt liên tiếp nhau vì sẽ làm cho trang web của bạn bị rối lên bởi các ảnh động đó.

Đối với các ảnh quan trọng như ảnh Banner, trước hết bạn nên sử dụng các ứng dụng xử lý ảnh như photoshop, image composer,... để cắt ghép,... tạo được một hình ảnh tổng thể theo yêu cầu. Nếu hình ảnh này có Kích thước nên bạn chia hình đó thành nhiều hình nhỏ và xuất chúng thành các file ảnh khác nhau.Sau đó khi thiết kế Web, bạn sẽ tạo bảng có số ô bằng số file ảnh bạn đã chia và ứng với mỗi ô trong bảng bạn hãy chèn file hình ảnh tương ứng vào ô đó. Khi đó bạn sẽ được trang Web có hình ảnh được sắp đặt giống file ảnh lớn ban đầu nhưng thời gian tải ảnh đã giảm đi đáng kể.

3.3.4. Bố cục đơn giản, hợp lý.

Website của bạn cần có cấu trúc càng rõ ràng, càng dễ hiểu càng tốt. Điều quan trọng ở đây là phải làm sao để người xem thấy được ngay các thông tin mà họ hi vọng có thể thu được từ website của bạn. Nếu website của bạn có quá nhiều thông tin, bạn có thể làm cho trang chủ đơn giản bằng cách thiết kế bảng nội dung chính của website sau đó tạo các liên kết đến các trang thông tin khác đồng thời sử dụng những từ và đoạn ngắn gọn dễ hiểu để thu hút người đọc.

Banner để thể hiện chủ đề chính của Website ví dụ như tên, logo, slogan, địa chỉ, hình đại diện,.. của tổ chức nên đặt ở đầu mỗi trang Web. Các mục liên kết đặt ở phía trái hoặc trên đầu (dưới baner) mỗi trang web. Có thể đặt thêm các mục liên kết ở phía cuối trang nếu trang có nội dung quá dài. Các mục liên kết này, bạn nên sử dụng những từ ngắn gọn, dễ hiểu, thiết kế thật nổi bật để thu hút người xem. Bạn nên thống nhất kiểu thiết kế của các trang web trong website.

Ở mỗi trang web, bạn nên tạo liên kết đến các trang web khác hay tạo liên kết về trang chủ hoặc tạo liên kết đến các phần khác nhau trong nó nếu trang web có nội dung quá dài. Bạn có thể làm cho trang web của mình sinh động hơn, thu hút hơn bằng cách sử dụng những biểu tượng biến hình, những mục liên kết bằng ảnh, hiệu ứng đặc biệt.

Nếu bạn có những nội dung quan trọng trong trang web, hãy làm nó dễ đọc bằng cách chia thành những đoạn quan trọng rồi bôi đậm những câu quan trọng và dùng ngại tô màu văn bản đó.

Khi thiết kế bạn cũng cần để nhiều “khoảng trắng” trong website. Những khoảng trắng này sẽ tạo điều kiện cho mắt người sử dụng được nghỉ ngơi trong giây lát và để họ có thể tập trung hoàn toàn vào những phần riêng biệt khác trong trang.

3.3.5. Thời gian tải.

Nhân tố thời gian là vô cùng quan trọng vì mọi người thường không kiên nhẫn khi vào mạng, đặc biệt là khi không phải ai cũng sử dụng đường truyền Internet tốc độ cao. Để người xem phải nhấn chuột và bỏ đi chỉ vì trang web của bạn tải về quá chậm. Hãy tăng tốc độ truyền của các trang web lên bằng cách:

- Không sử dụng ảnh “GIFS” và Javascripts để tạo các logo và ký tự chạy ngang màn hình hay những gì tương tự. Điều này không chỉ làm tăng thời gian tải về mà còn làm người xem sao lãng nội dung cần trình bày trên trang web của bạn.
- Sử dụng đồ họa trang trí là tốt nhưng đừng lạm dụng. Hãy thu nhỏ kích cỡ, giảm chế độ phân giải của hình ảnh. Nếu bạn rất cần một hình ảnh lớn thì có thể cung cấp cho người xem một hình ảnh nhỏ với tùy chọn xem tiếp một hình ảnh lớn hơn.
- Nên quy định cụ thể Kích cỡ file đồ họa trong mã HTML vì trình duyệt web sẽ rút ngắn cỡ trang nếu cần thiết và nơi hình ảnh sẽ xuất hiện sẽ để một khoảng trống cho file đồ họa tải về.
- Mỗi trình duyệt có tối đa bốn kết nối(socket), mỗi socket sẽ cho phép chuyển một file về máy tính của bạn. Vì thế, nếu bạn có 4 socket thì bạn có thể tải về cùng lúc 4 file. Nếu bạn có 6 file ảnh trong một trang chủ và một file HTML thì tất cả là 7 file cần tải về. Trình duyệt sẽ tải 4 file về trước, sau khi tải xong một file, socket sẽ tải file tiếp theo...Quá trình tải về có thể kéo dài nếu có quá nhiều file đặc biệt khi những file này lớn. Vì thế bạn nên giảm số file trong trang web của bạn, nên cố gắng có dưới 5 file một trang.

4. Nhập môn ngôn ngữ HTML

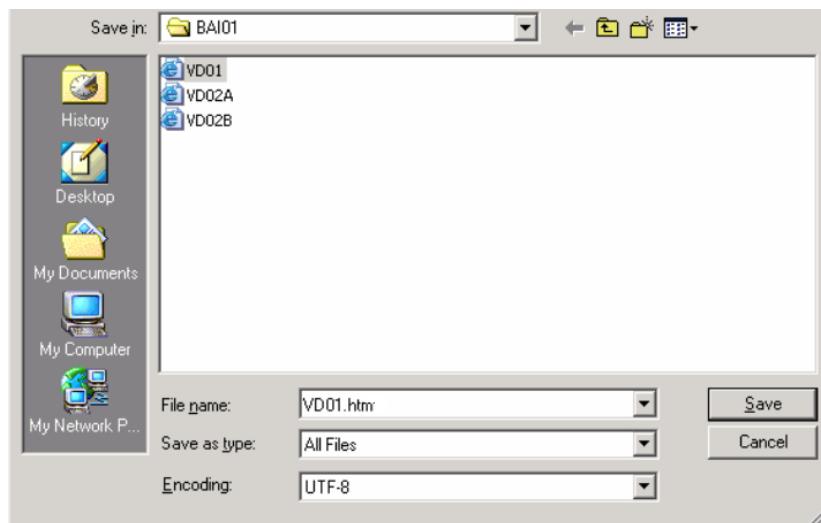
A. LÝ THUYẾT.

4.1. Soạn thảo trang mã nguồn HTML.

Trang mã nguồn HTML là một tệp văn bản bình thường gồm các kí tự ASCII, có thể được tạo ra bằng bất cứ trình soạn thảo thông thường nào. Nói soạn thảo siêu văn bản tức là tạo ra trang mã nguồn HTML đúng quy định để độ duyệt hiểu được và hiển thị đúng. Thông thường chúng ta thường viết mã nguồn HTML trên chương trình soạn thảo văn bản không

định dạng như NotePad. Các bước tiến hành tạo trang mã nguồn HTML như sau :

- Thiết lập bảng mã Unicode cho trình gõ VietKey, Unikey.
- Khởi động Note Pad : **Start/Program/Accessories/NotePad**
- Chọn **Font Unicode** trong trình soạn thảo (**Format / Font** → Arial hoặc **Times New Roman**).
- Gõ mã lệnh HTML.
- Lưu tệp tin:
 - + Tại cửa sổ soạn thảo: **File / Save As**.



Hình 1.2 : lưu tệp tin HTML.

- + Gõ tên tệp tin trong khung **File name** với phần mở rộng .html hoặc .htm (Ví dụ: VD01.htm).
- + Chọn kiểu file trong khung **Save as type** là **AllFiles**.
- + Chọn mã **Encoding** là **UTF-8** (nếu nội dung có là tiếng việt có dấu).
- + Nhấn nút **Save**.
- Mở tệp tin đã soạn thảo: Tại cửa sổ soạn thảo: **File / Open** → Chọn tệp tin cần mở và kích nút **Open**.
- Xem trang Web sau khi soạn thảo:
 - + Cách 1: Mở thư mục đã lưu trữ sau đó Kích đúp chuột vào tệp tin .html.
 - + Cách 2: Khởi động trình duyệt Web → **File / Open** → Chọn **Browser** để tìm tệp tin cần mở sau đó Kích Ok
- Làm tươi nội dung : Khi đang hiển thị một trang web trên trình

duyệt có thể xem và điều chỉnh nội dung mã nguồn từ trình soạn thảo. Khi đã lưu lại nội dung điều chỉnh, từ trình duyệt ta có thể xem lại kết quả theo nội dung mới bằng cách kích vào biểu tượng **Refresh** hoặc nhấn phím F5.

- Xem mã nguồn: Từ trình duyệt có thể xem mã nguồn trang Web bằng cách kích vào **Menu View / Source**.

Chú ý:

Hiện nay có nhiều công cụ soạn thảo siêu văn bản mạnh như Microsoft FrontPage, DreamWeaver.. với giao diện trực quan và tự động sinh mã HTML, cho phép soạn thảo siêu văn bản như soạn thảo thông thường. Tuy nhiên, việc tìm hiểu cú pháp của HTML, nắm vững ý nghĩa của các thẻ khác nhau vẫn rất cần thiết để có thể tạo ra các trang Web động, tương tác với người sử dụng, tức là các ứng dụng Web sau này.

4.2. Các thẻ HTML(Tags).

Tập tin HTML là 1 văn bản chứa các thẻ (Tag) để tạo các thành phần của tài liệu. Thẻ là thành phần cơ bản nhất trong HTML. Thẻ báo cho trình duyệt biết phải thể hiện cái gì hoặc phải làm gì với đối tượng chịu tác động của thẻ.

Mỗi thẻ gồm một từ khoá tên thẻ bao bọc bởi hai dấu "`<`" và "`>`". Hầu hết các lệnh thẻ hiện bằng một cặp hai thẻ: thẻ mở (`<thẻ>`) và thẻ đóng (`</thẻ>`). Dấu gạch chéo ("`/`") ký hiệu thẻ đóng. Lệnh sẽ tác động vào đối tượng nằm giữa hai thẻ.

Cấu trúc tổng quát của một thẻ :

`<thẻ [thuộc tính]> thành phần chịu tác động </thẻ>`

Ví dụ : ` Chào các bạn ! ` . Khi duyệt web, nội dung ‘Chào các bạn’ sẽ được in đậm.

Một số thẻ không có cặp, chúng được gọi là các thẻ rỗng hay thẻ đơn. Chỉ có thẻ mở `<thẻ>` mà thôi.

Nhiều thẻ có kèm các thuộc tính (attribute), cung cấp thêm các tham số chi tiết hơn cho việc thực hiện lệnh. Các thuộc tính được chia làm hai loại: thuộc tính bắt buộc và thuộc tính không bắt buộc hay tùy chọn. Một thuộc tính là bắt buộc nếu như phải có nó thì thẻ lệnh mới thực hiện được.

Ví dụ: Để chèn một hình ảnh vào trang tài liệu ta dùng thẻ ``. Tuy nhiên, cần chỉ rõ cái ảnh nào sẽ được dán vào đây. Điều này được thiết

lập bằng thuộc tính SRC ="địa chỉ của tệp ảnh". Thuộc tính SRC là bắt buộc phải có đối với thẻ .

Một số quy tắc chung khi soạn thảo siêu văn bản bằng HTML:

- Nhiều dấu cách liền nhau cũng chỉ có tác dụng như một dấu cách. Bạn phải sử dụng thẻ để thể hiện nhiều dấu giãn cách liền nhau.
- Gõ Enter để xuống dòng được xem như một dấu cách, để xuống hàng thì chúng ta phải sử dụng thẻ tương ứng.
- Có thể viết tên thẻ không phân biệt chữ in thường và in hoa.
- Có thể chèn các dòng bình luận, chú thích... vào trang mã nguồn bằng cách đặt giữa cặp dấu chú thích <!--đoạn chú thích -->. Trình duyệt sẽ bỏ qua không xét đến phần mã nằm giữa cặp dấu đó.
- HTML cho phép các thẻ lồng nhau, thẻ nào mở sau phải đóng trước.

HTML cho viết các thẻ trên cùng 1 dòng hoặc nhiều dòng. Chúng ta nên viết có cấu trúc.

4.3. Cấu trúc tổng quát tệp tin HTML.

- Mọi tài liệu HTML đều có khung cấu trúc như sau

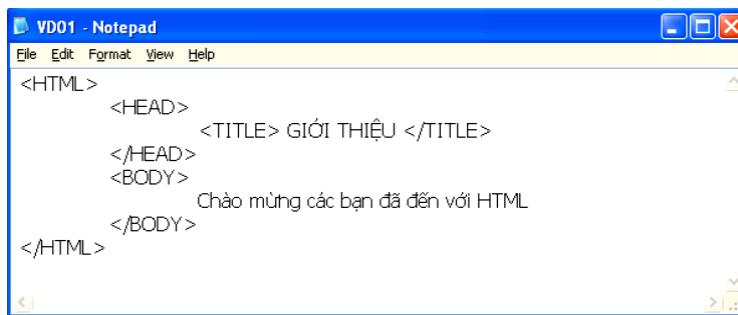
```
<HTML>
  <HEAD>
    <TITLE>
      Nội dung tiêu đề
    </TITLE>
  </HEAD>
  <BODY>
    Phần dữ liệu của trang web
  </BODY>
</HTML>
```

Hình 1.3: Cấu trúc trang HTML

- <HTML> .. </HTML>: Là từ khoá để tất cả các trình duyệt web đều hiểu đó là bắt đầu và kết thúc tệp tin HTML.
- <HEAD> .. </HEAD>: Phần đầu trang web. Được dùng để chỉ ra tiêu đề và một số thông tin khác được dùng cho máy tìm kiếm như:
 - + <Base>..</Base>: Khai báo các địa chỉ cho file HTML.
 - + <Meta> .. </Meta>: Các thông tin về trang web.

- + <Title>..</Title>: xâu ký tự nằm giữa cặp thẻ này sẽ hiển thị tại thanh tiêu đề của trình duyệt và Nếu bỏ trống thì trình duyệt sẽ cho hiện tên tệp thay vào đó.
- Toàn bộ nội dung của tài liệu nằm giữa hai thẻ xác định thân của trang <BODY>... </BODY>. Các dòng văn bản, hình ảnh, âm thanh, video, các mồi liên kết... tạo nên trang Web đều phải nằm ở đây.

Ví dụ:



```

VDO1 - Notepad
File Edit Format View Help
<HTML>
  <HEAD>
    <TITLE> GIỚI THIỆU </TITLE>
  </HEAD>
  <BODY>
    Chào mừng các bạn đã đến với HTML
  </BODY>
</HTML>

```

Hình 1.4 : Soạn thảo mã lệnh HTML

- Kết quả hiển thị.



Hình 1.5: Kết quả hiển thị trang VD01.htm

Để soạn thảo tài liệu HTML đúng cú pháp, cần nắm vững cấu trúc của từng thành phần HTML. Không giống như khi lập trình, nếu bạn mắc lỗi cú pháp HTML sẽ không có một thông báo lỗi nào báo cho biết mà trình duyệt sẽ hiểu làm và trình bày trang Web không như bạn muốn mà thôi.

4.4. Thẻ BODY.

Thẻ BODY là thẻ quan trọng nhất của trang HTML. Các nội dung trang Web sẽ được đặt giữa thẻ chứa <BODY>. Vì thế các thuộc tính của thẻ body sẽ chứa các định dạng chung tác động đến toàn bộ trang web. Ví dụ như màu nền, hình nền, màu siêu liên kết, đặt lề... Các thuộc tính chính của thẻ BODY như sau:

- BgColor: màu nền của trang web.
- Background: ảnh nền của trang web.

- LeftMargin: lề trái của trang web.
- RightMargin: lề phải của trang web.
- TopMargin: lề trên của trang web.
- BottomMargin: lề dưới của trang web.
- Text: xác định màu chữ của văn bản trong trang web.
- Link, Vlink, Alink: xác định màu sắc cho liên kết chưa, đã và đang được kích hoạt.
- Scroll: có hay không có thanh cuộn(Yes/No).

Ví dụ 1: Giả sử cần bổ sung thuộc tính ảnh nền vào 1 trang trang web lưu trong thư mục Htmls, tập tin ảnh lưu trong thư mục Images (cùng cấp với thư mục Htmls). Thuộc tính thẻ BODY như sau:

<BODY Background = "../Images/Bgr06.JPG" >

Để xác lập màu có thể dùng Tên màu hay trị số thập lục phân tương ứng. Sau đây là bảng các mã màu cơ bản.

Tên màu		Trị		Tên màu		Trị	
Black	Đen	#000000		Olive	Nâu vàng	#808000	
Red	Đỏ	#FF0000		Teal	Nâu sẫm	#008080	
Blue	Xanh	#0000FF		Maroon	Nâu	#800000	
Navy	X.Đậm	#000800		Gray	Xám	#808080	
Lime	V.Chanh	#FFFF00		Fuchsia	Hồng	#FF00FF	
White	Trắng	#FFFFFF		Green	Lá	#008000	
Purple	Đ.Tía	#800080		Silver	Bạc	#C0C0C0	
Yellow	Vàng	#FFFF00		Aqua	Ngọc	#00FFFF	

Hình 1.6: Bảng mã màu.

Ví dụ: Xác định chữ màu xanh Navy, màu liên kết là màu đỏ cho toàn bộ trang web ta sử dụng các thuộc tính thẻ BODY như sau:

<BODY Text = “#000800” Link = “#FF0000” >

4.5. Văn bản trên trang web.

4.5.1. Văn đề ngắt dòng.

Tạo tập tin VD02.htm với nội dung như sau:

<HTML>

<HEAD>

<TITLE> > BAI THO CO HAI MO </TITLE>

</HEAD>

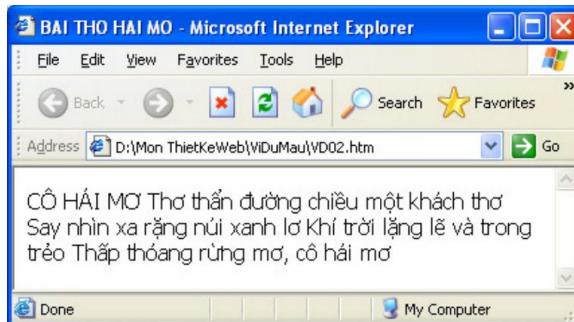
<BODY>

CÔ HÁI MƠ

Thơ thần đường chiều một khách thơ.
Say nhìn xa rặng núi xanh lơ.
Khí trời lặng lẽ và trong trẻo.
Thấp thoáng rừng mơ, cô hái mơ.

</BODY>
</HTML>

Khi duyệt web chúng ta có kết quả như sau:



Hình 1.7 : Ví dụ về không ngắt dòng.

Như vậy các câu thơ không xuống dòng như soạn thảo mà chỉ tự mà
ngắt dòng tùy theo kích thước cửa sổ trình duyệt.

- Để ngắt dòng tạo đoạn mới, tạo dòng trống: Đặt văn bản trong cặp thẻ <P>...</P>.
- Để ngắt dòng không tạo đoạn mới: Chèn thẻ
 tại vị trí muốn
ngắt dòng.
- Để giữ các phần tử trên cùng 1 dòng: Đặt văn bản trong cặp thẻ
<Nobr>...</Nobr>.
- Khi đặt chuỗi ký tự " " vào giữa hai từ thì chúng không bao
giờ bị ngắt rời xuống dòng. Chuỗi ký tự này còn được dùng để tạo các
khoảng trắng cần thiết giữa hai từ hoặc một ô trắng của một bảng.
- Nội dung văn bản trong cặp thẻ <PRE> ..</PRE> sẽ được giữ
nguyên vị trí như khi soạn thảo khi hiển thị trên trình duyệt.

Thẻ <P> có tham số **Align** = "Center / left / right / justify": căn giữa,
trái, phải, căn đều đoạn văn bản. Thông thường người ta áp dụng thẻ <P>
với thẻ , <i>, ,...để tạo và căn chỉnh đoạn văn bản.

Ví dụ : Soạn thảo tệp tin VD03.htm với nội dung như sau
<HEAD>

<TITLE> > BAI THO CO HAI MO </TITLE>
</HEAD>

```

<BODY>
    CÔ HÁI MƠ
    <P align = "left" >
        Thơ thần đường chiều một khách thơ <br>
        Say nhìn xa rặng núi xanh lơ <br>
        Khí trời lặng lẽ và trong tréo <br>
        Thấp thoáng rừng mơ, cô hái mơ <br>
    </P>
</BODY>
</HTML>

```

Kết quả hiển thị trên trình duyệt sẽ như sau:



Hình 1.8: Kết quả sau khi đã ngắt dòng.

Văn bản sẽ bị che khuất khi chiều rộng cửa sổ trình duyệt không đủ và sẽ hiện thanh cuộn ngang.

4.5.2. Định dạng kiểu dáng – Style

- Nội dung văn bản : → Nội dung văn bản được in đậm.
- <I> Nội dung văn bản </I> : → Nội dung văn bản được in nghiêng.
- <U> Nội dung văn bản </U> : → Nội dung văn bản được gạch dưới.
- <S> Nội dung văn bản </S> : → Nội dung văn bản được gạch ngang.

4.5.3. Thay đổi Font, Size.

Thẻ được dùng để thay đổi một số thuộc tính hiển thị cho đoạn văn bản. Một số thuộc tính thông dụng của thẻ:

- face: tên font chữ được chọn cho đoạn văn bản.
- color: màu chữ.
- size: Kích cỡ của chữ.

Ví dụ: Soạn thảo tệp tin VD04.htm với nội dung sau:

```

<HEAD>
    <TITLE> > BAI THO CO HAI MO </TITLE>
</HEAD>
<BODY>
    <P><Font size=4><B> CÔ HÁI MƠ </B></Font></P>
    <I>
        Thơ thần đường chiêu một khách thơ <br>
        Say nhìn xa rặng núi xanh lơ <br> <I>
        <B><I>
            Khí trời lặng lẽ và trong trèo <br>
            Thấp thoáng rừng mơ, cô hái mơ <br> </B></I>
    </I>
</BODY>
</HTML>

```

Kết quả hiển thị sẽ là:



Hình 1.9: Kết quả sau khi định dạng chữ.

Ngoài ra HTML còn sử dụng một số thẻ khác dùng để định dạng font chữ sau:

Thẻ	Ý nghĩa
	Chữ thấp (A ₂)
	Chữ lũy thừa (A ²)
<BIG> </BIG>	Phóng lớn chữ
<SMALL> </SMALL>	Thu nhỏ chữ

Ví dụ : Soạn thảo tệp tin VD05.htm có nội dung như sau:

```

<HTML>
<HEAD>

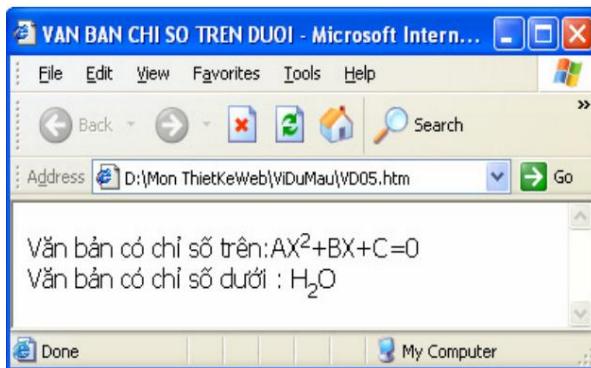
```

```

<TITLE> VAN BAN CHI SO TREN DUOI </TITLE>
</HEAD>
<BODY>
    Văn bản có chỉ số trên: AX2+BX+C=0<BR>
    Văn bản có chỉ số dưới : H2O
</BODY>
</HEAD>
</HTML>

```

Kết quả như sau:



Hình 1.10: Văn bản có chỉ số

4.5.6. Đường kẻ ngang và ký tự đặc biệt.

❖ Đường kẻ ngang

Đường kẻ ngang chủ yếu được dùng để phân định một trang Web thành những phần có tính logic và dễ nhìn. Có không nhiều khả năng đồ họa trong ngôn ngữ HTML, bởi vậy đường kẻ ngang là một công cụ rất có ích. Thẻ rỗng `<HR>` dùng để tạo đường kẻ ngang, nó thường dùng các thuộc tính sau:

- Width: chiều dài (tính theo pixel hoặc % bề rộng cửa sổ trình duyệt).
- Color: màu đường kẻ.
- Size: Độ dày đường kẻ.
- NoShade: Không có bóng.

Ví dụ: `<HR Size=3 Width=50% Align="Right">`

❖ Ký tự đặc biệt

- Có một số ký tự đã được dùng làm thẻ đánh dấu hoặc một số ký tự đặc biệt không gõ được trong trình soạn thảo. Để hiển thị các ký tự này HTML ta sử dụng quy tắc viết các ký tự đặc biệt như sau :

Ký tự	Mã tên	Mã số	Ký tự	Mã tên	Mã số
<	<	<	Khoảng trắng	&nnbsp;	
>	>	>	©	©	©
“	"	"	™	™	™
®	®	®			

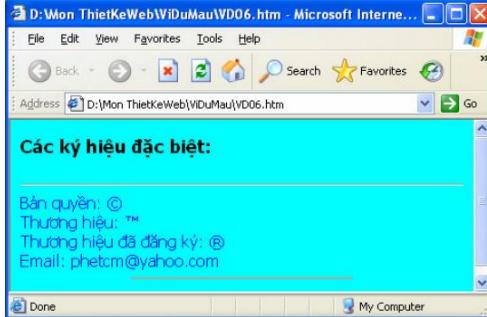
Ví dụ : Soạn thảo trang VD06.htm như sau:

```

<HTML>
    <BODY Bgcolor = "Aqua">
        <P Align=Center><H3>Các ký hiệu đặc biệt:</H3><P>
        <HR><Font Color=Blue>
            Bản quyền: ©<BR>
            Thương hiệu: ™<BR>
            Thương hiệu đã đăng ký: ®<BR>
            Email: phetcm@yahoo.com <BR></Font>
            <HR SIZE=3 WIDTH=50% NOSHADE>
        </BODY>
    </HTML>

```

Kết quả hiển thị trên trình duyệt là:



Hình 1.11: Sử dụng ký tự đặc biệt

4.5.7. Tạo văn bản đề mục

Đề mục là dòng văn bản định dạng khác các dòng văn bản trong nội dung dùng để tạo đề mục phần, chương. Mỗi thẻ có chức năng phân đoạn, cấp 1 lớn nhất, cấp 6 nhỏ nhất, có thể kết hợp thuộc tính canh lề.

<H1> Nội dung văn bản đề mục cấp 1</H1>

<H2> Nội dung văn bản đề mục cấp 2</H2>

.....

<H6> Nội dung văn bản đề mục cấp 6</H6>

Ví dụ: <H3 Align=Center> Đề mục cấp, 3 canh lè giữa</H3>

4.5.8. Tạo danh sách cho các đoạn văn bản

❖ **Danh sách không đánh số thứ tự.**

Danh sách không đánh số thứ tự là: Tạo chỉ mục, hoa thị (Bullets) đầu mỗi đoạn. Mỗi mục sẽ thụt vào đầu dòng.

Việc tạo danh sách không đánh số thứ tự như sau:

```
<UL>
    <LI> Mục 1 của danh sách
    <LI> Mục 2 của danh sách
    . . .
    <LI> Mục n của danh sách
</UL>
```

Thẻ `` và thẻ `` có thuộc tính TYPE với các trị thuộc tính sau:

- Disc: Dấu tròn đen (mặc định).
- square: Hình vuông đen đặc.
- Circle: Hình tròn rỗng.

Ví dụ: Tạo tệp tin VD08.htm với nội dung như sau:

```
<HTML>
    <BODY>
        <H3> CHƯƠNG TRÌNH ĐÀO TẠO THIẾT KẾ WEB</H3>
        <UL TYPE="circle">
            <LI> Ngôn ngữ HTML &amp; FrontPage
            <LI> Dreamweaver
            <LI TYPE="square"> Quản trị Domain, Hosting
            <LI TYPE="square"> Ngôn ngữ kịch bản JavaScript;
            <LI TYPE="disc"> Ngôn ngữ ASP &amp; SQLServer
        </UL>
    </BODY>
</HTML>
```

Kết quả hiển thị trang web trên trình duyệt như sau:



Hình 1.12: Danh sách không đánh số thứ tự.

❖ **Danh sách đánh số thứ tự.**

Danh sách có đánh số TT là: Loại danh sách có đánh thứ tự theo số hay theo ký tự (Numbering) đầu mỗi đoạn.

Tạo danh sách có đánh số thứ tự như sau:

```
<OL>
    <LI> Mục 1 của danh sách
    <LI> Mục 2 của danh sách
    .....
    <LI> Mục n của danh sách
</OL>
```

Thẻ và thẻ có thuộc tính TYPE với các trị thuộc tính sau:

Trị	Kiểu hiển thị
1	1,2,3 ...
a	a,b,c ...
A	A,B,C...
i	i,ii,iii ...
I	I,II,III ...

Hình 1.13: các trị thuộc tính của danh sách có thứ tự.

Ngoài ra hai thẻ trên còn có thuộc tính START = n để chỉnh giá trị khởi đầu của danh sách.

Ví dụ: Soạn thảo trang VD11.htm theo nội dung sau:

```
<HTML>
<BODY >
    <H3> CHƯƠNG TRÌNH ĐÀO TẠO THIẾT KẾ WEB </H3>
    <OL Type="1">
        <LI> Ngôn ngữ HTML &amp; FrontPage
        <LI> DreamWeaver
```

```

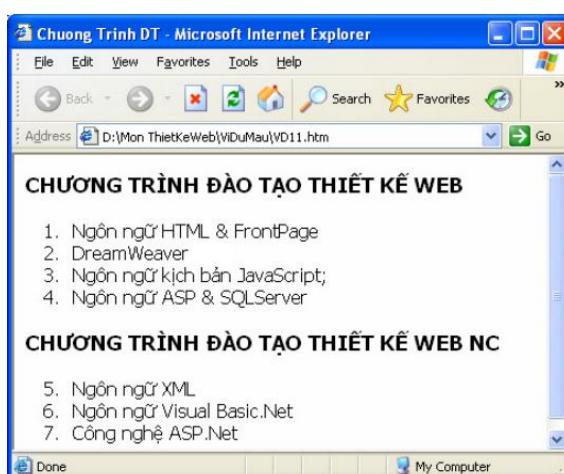
<LI> Ngôn ngữ kịch bản JavaScript;
<LI> Ngôn ngữ ASP & SQLServer
</OL>

<H3> CHƯƠNG TRÌNH ĐÀO TẠO THIẾT KẾ WEB NC</H3>
<OL Type="1" START="5">
<LI> Ngôn ngữ XML
<LI> Ngôn ngữ Visual Basic.Net
<LI> Công nghệ ASP.Net
</OL>

</BODY>
</HTML>

```

Kết quả trang web trên trình duyệt như sau:



Hình 1.14: Danh sách có thứ tự.

❖ **Danh sách định nghĩa.**

Dùng để lập danh sách thuật ngữ. Mỗi mục của danh sách gồm 2 phần: Khai báo thuật ngữ và định nghĩa thuật ngữ.

Tạo danh sách định nghĩa như sau:

```

<DL>
  <DT> Khai báo thuật ngữ
  <DD> Định nghĩa thuật ngữ
  .....
</DT>

```

```

</DL>

```

Ví dụ : Tạo tập tin VD12.htm như sau:

```

<HTML>
  <BODY Text="Blue">

```

```

<B> CHƯƠNG TRÌNH ĐÀO TẠO </B>
<DL>
    <DT><Font Color="Red">TIN HỌC VĂN PHÒNG</FONT>
        <DD> Tin học căn bản & WindowsXP
        <DD> Soạn Thảo văn bản MicroSoft Word
        <DD> Xử lý bảng tính MicroSoft Excel
    </DT>
    <DT><FONT Color="Red"> TIN HỌC QUẢN LÝ </FONT>
        <DD> Lập trình CSDL MicroSoft Access 2003
        <DD> Thiết kế trình chiếu MicroSoft PowerPoint
    </DT>
</DL>
</BODY>
</HTML>

```

Kết quả trang web trên khi duyệt như sau:



Hình 1.15 : Danh sách tự định nghĩa.

❖ **Danh sách lồng nhau.**

Kết hợp các thẻ ****, ****, **** để lập danh sách lồng nhau

Ví dụ: Tạo tập tin VD13.htm như sau:

```

<HTML>
<BODY >
    <H3>CHƯƠNG TRÌNH ĐÀO TẠO NGÀNH WEBSITE</H3>
    <OL TYPE="I">
        <LI><B> THIẾT KẾ WEBSITE </B>
        <OL TYPE = “1”>
            <LI> Ngôn ngữ HTML & FornPage
            <LI> Ngôn ngữ kịch bản JavaScript
    </OL>
</BODY>
</HTML>

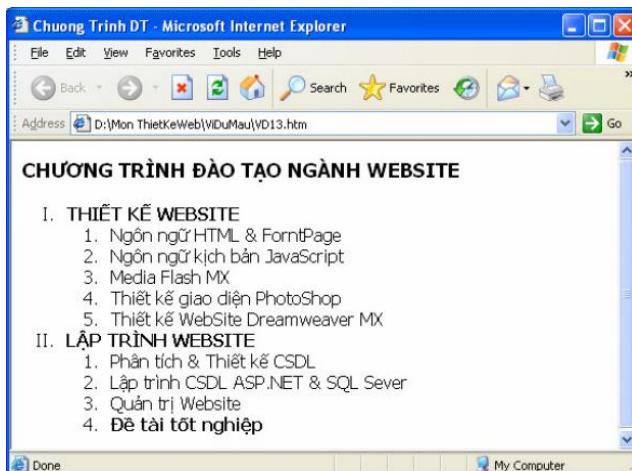
```

```

<LI> Media Flash MX
      <LI> Thiết kế giao diện Photoshop
      <LI> Thiết kế WebSite Dreamweaver
</OL>
<LI><B> LẬP TRÌNH WEBSITE </B>
<OL TYPE = "1">
    <LI> Phân tích & Thiết kế CSDL
    <LI> Lập trình CSDL ASP.NET & SQL Sever
    <LI> Quản trị Website
    <LI><B>Đề tài tốt nghiệp </B>
</OL>
</OL>
</BODY>
</HTML>

```

Kết quả duyệt web như sau:



Hình 1.16: Danh sách lồng nhau.

4.5.9. Tạo hiệu ứng chuyển động chữ.

```

<Marquee Behavior = "Hiệu ứng" Direction = "Hướng" Loop = n1
    ScrollAmount = n2 ScrollDelay = n3 BgColor = "Trí màu">
    Chuỗi ký tự cần chuyển động.
</Marqueee>

```

Trong đó:

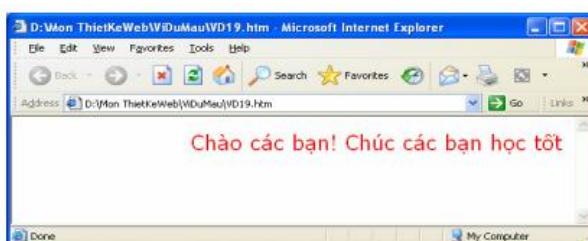
- Hiệu ứng:
 - + Scroll : Bắt đầu từ 1 cạnh biến mất ở cạnh kia.
 - + Slide : Bắt đầu từ 1 cạnh dừng lại ở cạnh kia.

- + Alternate : Bắt đầu từ 1 cạnh chuyển ngược lại ở cạnh kia.
- Hướng: Hướng bắt đầu gồm: Left, Right, Up, Down
- n1: Số lần lặp.
- n2: Khoảng cách(Pixel) giữa mỗi lần lặp.
- n3: Thời gian chờ giữa mỗi lần lặp.

Ví dụ: Soạn thảo tệp tin VD19.htm như sau:

```
<HTML>
<BODY>
<Marquee Behavior="alternate" Direction="right">
<Font color="red" Size=5> Chào các bạn! Chúc các bạn học
tốt</Font>
</Marquee>
<BODY>
</HTML>
```

Kết quả khi duyệt trang web này:



Hình 1.17 : Hiệu ứng chuyển động của chữ.

4.6. Thiết lập liên kết cho trang web.

4.6.1. Khái quát.

Các siêu liên kết (HyperLink) gọi tắt là liên kết là cốt lõi của tập tin HTML cho phép truy cập một tài liệu khác hay một phần tài liệu khác nằm trong tài liệu đó. Tài liệu liên kết có thể là các tập tin : Hình ảnh, âm thanh, Multimedia, HTML . . . cùng máy tính hay ở máy tính khác. Khi kích vào liên kết sự truy cập tại địa chỉ truy cập sẽ thực hiện tự động.

Để tạo một mối liên kết thì việc đầu tiên cần làm là thiết lập cái neo (Anchor). Cặp thẻ tạo neo là <A>...(Anchor). Thẻ này có nhiều thuộc tính bắt buộc phải xác định rõ, ta sẽ tìm hiểu kỹ hơn ở phần sau.

4.6.2. Liên kết cục bộ (Local Link).

- Liên kết đến các tài liệu trên cùng máy .

** Nhãn liên kết **

Ví dụ: Giới thiệu

- Đến 1 vị trí trên cùng trang Web (liên kết bookmark):

 Nhãn liên kết

Khai báo đích đến:

 . . .

(Tên đích ở bất cứ vị trí nào trên cùng tập tin).

Ví dụ : Tạo tệp tin VD13.htm như sau:

<HTML>

<BODY Background="Images/bg.jpg" Bgproperties =Fixed>

 <H3 Align="center"> CHƯƠNG TRÌNH ĐÀO
 TẠO </H3>

 1. Kỹ thuật viên tin học

 2.Chuyên ngành Website <HR>

 Kỹ thuật vin tin học

 Học phần I

 Học phần II

 Học phần III

 <OL Type="I">

 Tin Học Căn Bản

 Tổng quan cấu trúc máy tính

 Hệ điều hành Windows & Internet

 Tin Học Văn Phòng

 Microsoft Word

 MicroSoft Excel

 Tin Học Quản Lý

 Microsoft Access

```

<LI> MicroSoft PowerPoint
</OL></A>
</OL></P>
<P align=Right><A href="#Dau">Đầu trang </A></P>
</BODY>
</HTML>

```

Kết quả trình duyệt web.



Hình 1.18 : Tạo liên kết cục bộ.

4.6.3. Liên kết từ xa (Remote Link)

- Là liên kết đến tài liệu lưu trữ trên máy khác.

 Nhãn liên kết

Ví dụ:

```

<A HREF = "http://www.vietnamnet.vn "> Website tin tức Việt Nam
</A>

```

- Liên kết đến 1 địa chỉ Email

 Nhãn liên kết

Ví dụ:

```

<A HREF ="mailto:admin@yahoo.com"> Gửi Mail cho admin </A>
Sẽ mở chương trình mail mặc định để soạn và gửi thư.

```

4.6.4. Dùng hình ảnh làm nhãn liên kết.

Ví dụ :

```

<A HREF = "http://www.vietnamnet.vn"> <Img Src =

```

"../Images/vietnamnet.jpg" >

4.6.5. Bản đồ ảnh dùng thẻ AREA.

Thử hình dung ta có một bản đồ địa lý hành chính. Khi nhấn chuột vào vùng diện tích của một tỉnh thì một trang thông tin tương ứng với tỉnh đó sẽ hiện ra. Đó chính là bản đồ ảnh (Image Map).

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

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

```
<MAP NAME = "Map01">
    <AREA HREF = "chapter1.html" SHAPE = "rect" coords= "6, 22,
                                                167, 169">
    <AREA HREF = "chapter2.html" SHAPE = "circle" coords = "259,
                                                106, 36">
    <AREA HREF = "chapter3.html" SHAPE = "polygon" coords = "123,
                                                196, 56, 211, 278, 201">
</MAP>
<IMG SRC="image.gif" WIDTH="400" HEIGHT="600"
USEMAP="#Map01">
```

Đoạn mã trên sẽ tạo 3 đầu mối liên kết là ba vùng hình chữ nhật (rect), hình tròn (circle) và hình tam giác (polygon) trên nền ảnh image.gif. Khi nhấn chuột vào phạm vi 3 vùng này thì các tệp tương ứng chapter1.html, chapter2.html và chapter3.html sẽ được nạp về để hiển thị.

4.6.6. Màu của liên kết.

Sau khi đặt liên kết, đoạn văn bản dùng làm điểm đặt liên kết thường được đổi màu và có gạch chân. Để chỉnh lại màu của liên kết bạn cần thay đổi các thuộc tính trong thẻ BODY. Ví dụ, bạn muốn liên kết chưa kích hoạt màu xanh, di chuột lên thì chuyển sang màu trắng, sau khi kích thì màu xám thì bạn phải khai báo:

```
<BODY link="blue" alink="white" vlink="gray">
```

4.7. Multimedia trên trang web

4.7.1. Chèn hình ảnh

Mỗi ảnh được lưu giữ trong máy tính thành một tệp riêng biệt. Tệp ảnh có nhiều định dạng khác nhau, thể hiện qua phần mở rộng (đuôi) của tên tệp: *.bmp, *.gif, *.jpeg, *.jpg,... Tuy nhiên các trình duyệt chỉ có thể “hiểu” được các tệp ảnh dưới dạng GIF, JPEG hay PNG . Hình ảnh dạng

GIF được sử dụng rộng rãi nhất trên Internet, sau đó là đến các ảnh dạng JPEG.

Khi sử dụng hình ảnh cần cân nhắc: số lượng ảnh/1trang, kích thước và độ phân giải, nhằm cải thiện tốc độ truy cập. Phải lưu hình ảnh ở vị trí nào đó trong thư mục chính của Website.

Để thêm hình ảnh vào trang ta dùng thẻ rỗng với một số thuộc tính:

- **Src** = "Địa chỉ URL của tệp tin" : được gán bằng địa chỉ URL tuyệt đối hoặc tương đối của file ảnh cần chèn. Src là thuộc tính này là bắt buộc của thẻ . Các thuộc tính còn lại là tùy chọn.

- **Alt** = "Câu chú thích" : Thuộc tính này cho phép ta chèn một đoạn chữ thay thế vào chỗ có hình và hiện lên xâu chú thích khi đưa trỏ chuột vào ảnh. Thuộc tính này dùng trong trường hợp trình duyệt không hiển thị được hình ảnh thì sẽ hiện dòng văn bản thay thế. Làm như vậy để những người sử dụng trình duyệt không có khả năng đọc ảnh hay đã tắt chức năng đọc ảnh để tăng tốc độ có thể biết được đó là hình ảnh gì và họ có thể chọn xem sau nếu có thời gian.

- **Height/ Width** = n: Thuộc tính dùng để xác định chiều rộng và chiều cao của ảnh. Giá trị này có thể tính theo phần trăm (%) hoặc pixel. Nếu đặt chiều rộng và chiều cao không chuẩn sẽ làm hình trong ảnh bị co giãn méo đi. Để khắc phục nhược điểm này, ta nên xử lý ảnh để kích thước phù hợp với yêu cầu.

- **Border** = n : độ dày đường viền của ảnh.
- **Align** = "hướng" : căn hàng văn bản bao quanh ảnh. Hướng bao gồm:

- + Top: Phần đầu thẳng hàng với dòng đầu văn bản.
- + Middle: Phần giữa thẳng hàng với dòng đầu văn bản.
- + Bottom: Phần cuối thẳng hàng với dòng đầu văn bản.
- + Left: Biên trái ảnh căn thẳng lề trái trang, văn bản bao quanh bên phải ảnh.
- + Right: Biên phải ảnh căn thẳng lề phải trang, văn bản bao quanh bên trái ảnh.

Nếu không khai báo mặc định là thuộc tính Bottom

- **Vspace** và **Hspace**: Khi sử dụng thuộc tính Align, ảnh được chèn vào dòng văn bản với các chữ dính sát liền. Thuộc tính VSPACE và HSPACE dùng để tạo khoảng trống nhỏ viền xung quanh ảnh (tính theo đơn vị pixel).
 - + VSPACE = n : Thêm khoảng trống theo chiều dọc.
 - + HSPACE = n : Thêm khoảng trống theo chiều ngang.

Ví dụ : Soạn thảo trang VD14.htm theo nội dung sau:

<HTML>

<BODY >

<H3 Align = Center> Máy tính "bó tay" với tài sản của Bill Gates

</H3>

<P Align=Justify>

Bill Gates, người sáng lập Công ty phần mềm Microsoft và là người giàu nhất thế giới, cho biết cơ quan thuế của Mỹ phải lưu trữ các dữ liệu tài chính của ông vào một máy tính đặc biệt, bởi tài sản của ông quá nhiều. Phát biểu tại một hội nghị do Microsoft tổ chức ở Lisbon, thủ đô Bồ Đào Nha, ông nói rằng một máy tính thông thường không thể xử lý đúng các số liệu tài chính của ông ...

</P>

</BODY>

</HTML>

Kết quả khi duyệt web là:



Hình 1.19: Chèn hình ảnh vào trang HTML.

4.7.2. Liên kết đến tệp tin âm thanh.

Các dạng tập tin âm thanh thông dụng trên Internet: .wav, .mp3, .wmv, .wma, .mpeg, .Ra, .Rm . . . hay các tập tin Flash : .swf

Liên kết đến *tệp âm thanh*. Để chèn một đoạn âm thanh vào tài liệu HTML ta cũng theo đúng quy tắc tạo mỗi liên kết thông thường. Trong thẻ neo, tại địa điểm URL bây giờ là địa chỉ của tệp âm thanh.

** Nhấn liên kết **
Target=_blank: Mở cửa sổ riêng

Ví dụ : Chia xa

Khi nhấn chuột vào mỗi liên kết này, tệp âm thanh sẽ được phát.

4.7.3. Chèn nhạc nền.

Nhạc nền là nhạc sẽ phát khi ta mở trang web trong trình duyệt web. Để chèn nhạc nền, chúng ta dùng thẻ rỗng Bgsound với các thuộc tính sau:

- **src** : đường dẫn tới file nhạc.
- **loop = n/infinite**.

Nhạc nền được đặt ở phần đầu của trang web (thẻ head).

Ví dụ : <Head> <bgsound src= "media/nhac1.mid" loop = 2> </head>

4.7.4. Chèn điều khiển nhạc Plugin.

Thẻ rỗng <embed> cùng với các thuộc tính sau dùng để chèn điều khiển nhạc vào trang web:

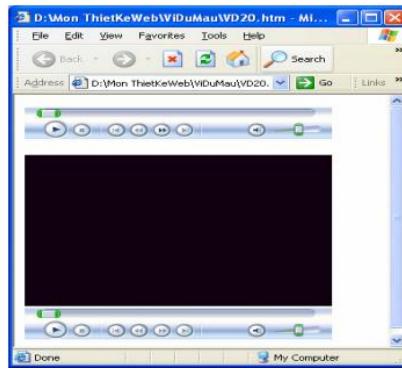
- **Src** = đường dẫn tới file nhạc.
- **Width/height**: chiều rộng và chiều cao của điều khiển nhạc.
- **Autostart** = "true/false": Có /không tự động chạy file nhạc khi mở file nhạc khi mở trang web.
- **Loop = "true / false"**: Lặp lại bản nhạc khi kết thúc hay không.

Đặt thẻ <EMBED> ở đâu thì khung nhìn của giao diện điều khiển việc phát tệp âm thanh xuất hiện ở đó.

Ví dụ: Tạo trang VD20.htm như sau:

```
<HTML>
  <BODY>
    <Embed SRC = "music/tiengdanbau.mp3" AutoStart = true
           width = 300 Height=250>
  </BODY>
</HTML>
```

Kết quả duyệt web như sau:



Hình 1.20: Chèn điều khiển nhạc.

Người ta có thể dùng thẻ <EMBED> để chèn video vào trang web bằng cách thay địa chỉ trang nhạc ở trên bằng địa chỉ file video.

4.7.3. Chèn video.

Thẻ với các thuộc tính sau sẽ dùng để chèn file video vào trang Web:

- **Dynsrc:** địa chỉ của file video.
- **Start = "fileopen/mouseover":** chỉ định file video sẽ được chơi khi tài liệu được mở hay khi trỏ chuột vào nó.
- **Loop= n/infinite:** chỉ định số lần chơi. Nếu infinite thì file video sẽ được chơi vô hạn lần.

Chú ý điều khiển này chỉ hỗ trợ các file video có định dạng: *.avi, *.ASF, *.ram, *.ra.

4.8. Bảng biểu.

Bảng là công cụ tuyệt vời để giúp chúng ta định vị các phần nội dung trên trang web. Bạn có thể đưa vào bảng các văn bản, hình ảnh, danh sách thậm chí chèn một bảng mới.

Bảng rất giống với danh sách, ta phải dùng thẻ chứa <TABLE> để chứa một nhóm các thẻ khác nhau định nghĩa cho mỗi hàng riêng rẽ (<TR>) và mỗi ô riêng rẽ cho từng hàng (<TD>).

Các thẻ để chèn bảng như sau:

<TABLE>

<CAPTION> tiêu đề bảng</ CAPTION>

<TR><TH> tiêu đề cột 1 </TH>....<TH> tiêu đề cột n</TH></TR>

<TR><TD> nội dung ô 1</TD>....<TD> nội dung ô n</TD></TR>

<TR><TD> nội dung ô 1</TD>....<TD> nội dung ô n</TD></TR>

.....

</TABLE>

4.8.1. Định nghĩa bảng – thẻ Table.

Thẻ Table dùng để định nghĩa bảng, nó có các thuộc tính chính sau:

- Border = n: xác lập độ rộng khung của bảng.
- Bordercolor: xác lập màu cho khung.
- Width: chiều rộng của bảng.
- Cellpadding: Khoảng cách từ mép ô đến nội dung của bảng.
- Cellspacing: Khoảng cách giữa các ô.
- bgcolor: xác lập màu nền cho bảng.
- background: xác lập hình nền cho bảng.
- Align : "Left/Right/Center/Justify " căn lề cho bảng.

4.8.2 Định nghĩa hàng-thẻ TR :

Thẻ TR để định nghĩa mỗi hàng trong bảng, số hàng trong bảng được thể hiện bằng số thẻ TR. Thẻ này có thể dùng thuộc tính sau:

- Align = "Left/Right/Center/Justify"
- Valign= "Top/Middle/Bottom"

4.8.3. Thẻ định nghĩa ô- thẻ TD :

Thẻ TD định nghĩa số ô trên một hàng, nó phụ thuộc vào số thẻ TD trên dòng. Thẻ TD có các thuộc tính chính sau:

- Align = "Left/Right/Center/Justify"
- Valign= "Top/Middle/Bottom"
- Bgcolor: màu nền cho ô.
- Background: Xác lập hình nền cho ô.
- Colspan=n: chỉ định ô sẽ kéo dài trong bao nhiêu cột.
- Rowspan=n: chỉ định ô sẽ kéo dài trong bao nhiêu hàng.

4.8.4. Thẻ định nghĩa tiêu đề cột - thẻ TH.

Thẻ TH dùng để định nghĩa tiêu đề cột, nó chỉ khác thẻ TD ở một điểm duy nhất đó là các chữ sẽ hiển thị đậm hơn. Ngoài ra có thể dùng thẻ Div để định dạng cho một nhóm các ô (giống thẻ <p>).

Ví dụ : Tạo tập tin VD17.htm với nội dung như sau:

<HTML>

<BODY>

<TABLE Border=1 BorderColor= "Purple" CellSpacing=0 Width=650>

<Caption><H2> KẾT QUẢ HỌC TẬP</Caption>

<TR Valign=Middle Align=Center BgColor="Lime">

```

<TH RowSpan=2 Width=150>Họ Võ Tên<BR>Học Sinh</TH>
<TH ColSpan=2 Width=150> Năm Sinh</TH>
<TH RowSpan=2 Width=70>Điểm<BR>TB</TH>
<TH RowSpan=2 Width=130> Xếp Loại</TH>
</TR>
<TR Align=Center BgColor="Lime">
<TH Width=75> Nam </TH>
<TH Width=75> Nữ </TH>
</TR>
<TR Align=Center>
<TD Align="Left"> Lê Thanh Xuân </TD>
<TD> &nbsp; </TD>
<TD> 1950 </TD>
<TD> 8.5 </TD>
<TD> Giỏi</TD>
</TR>
.....
<TR Align=Center>
<TD Align="Left"> Lưu Thé Mạc</TD>
<TD> 1985 </TD>
<TD> &nbsp; </TD>
<TD> 4.5 </TD>
<TD> Kém </TD>
</TR>
<TR Align="Center" BgColor="Lime">
<TD ColSpan=3> <B>Tổng số học sinh đạt:</B></TD>
<TD ColSpan=2> <B>4 Học sinh<B></TD>
</TR>
</TABLE></BODY>
</HTML>

```

Kết quả duyệt web như sau:

KẾT QUẢ HỌC TẬP				
Họ và tên Học sinh	Năm Sinh		Điểm TB	Xếp loại
	Nam	Nữ		
Lê Thanh Xuân		1950	8.5	Giỏi
Phan Thị Hạnh	1985		6.5	Khá
Trần Hoài Thu		1986	9.5	Xuất Sắc
Trương Lưu Đông	1984		6.0	Trung Bình
Lưu Thế Mạc	1985		4.5	Kém
Tổng số học sinh đạt:			4 Học sinh	

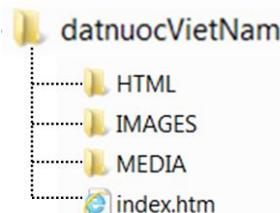
Hình 1.21 : Chèn bảng

B. THỰC HÀNH.

1. Nội dung thực hành

Sử dụng các ảnh và file tư liệu trong thư mục C:/DATA/TKWEB
Hãy tạo website “datnuocVietNam” theo yêu cầu sau:

- Cấu trúc website.



Hình 1.22 : Cấu trúc website

- Trang chủ (index.html):
 - + Giao diện.
 - + Sáu liên kết chính đặt ở góc trên bên trái, có màu nâu đỏ, khi di chuột vào có màu xanh và liên kết đến các trang con tương ứng trong thư mục HTML.



Hình 1.23 : Trang index.htm

- + Dòng chữ “ Thiết kế bởi... ” lặp liên tục từ trái sang phải, chữ trắng nghiêng, đậm cỡ 4.
- + Kích vào bản đồ ở giữa trang sẽ liên kết đến ảnh bản đồ lớn chi tiết các thành phố.
- + Nhúng nhạc nền cho trang web.
- Trang lichsu.html:
 - + Giao diện.
 - + Có danh mục ở đầu trang và đặt liên kết đến từng phần tương ứng trong trang.



Hình 1.24 : Trang lịch sử

2. Bảng vật tư thiết bị cần thiết.

TT	Thiết bị - Vật tư	Thông số kỹ thuật	Số lượng
	Hệ thống mạng và đường truyền Internet	LAN và ADSL	
1	Bộ máy tính	Cài đặt hệ điều hành Windows, phần mềm NotePad	1 bộ/1 học sinh
2	Bộ đĩa lưu các file ảnh, file dữ liệu phục vụ môn học	ảnh JPG	

3. Quy trình thực hiện.

TT	Công việc	Thao tác	Yêu cầu
1	Chuẩn bị	<ul style="list-style-type: none"> - Kiểm tra các hoạt động của máy tính. - Các ảnh mẫu. 	Hoạt động bình thường. Đầy đủ.
2	Thiết kế site	<ul style="list-style-type: none"> - Tạo website - Copy ảnh vào thư mục IMAGES. - Tạo các trang web trắng bằng NotePad và lưu theo quy cách vào thư mục HTML. 	Đúng yêu cầu
3	Thiết kế trang index.htm	<ul style="list-style-type: none"> - Chọn font tiếng việt Unicode của Vietkey. - Mở tệp tin index.htm bằng NotePad. - Đặt màu nền, nhạc nền trong thẻ body - Chèn bảng 1 có hai hàng, ba cột, nền trắng, kích thước 90%, có 2 hàng và 3 cột - Hàng 1 của bảng 1, nối 3 ô, chèn vào ảnh banner. - Ô đầu tiên của hàng 2 bảng 1, 	<ul style="list-style-type: none"> - Giao diện theo mẫu. - Bộ cục hợp lý. - Sử dụng chính xác các thẻ - Màu sắc hài hoà. - Nội dung chính xác.

		<p>chỉnh độ rộng 20%, màu ô vàng nhạt . Chèn vào ô này bảng 2. Bảng 2 có kích thước 100%, có 1 cột và 9 hàng. 6 hàng đầu tạo các liên kết bằng text đến các trang web tương ứng. 3 hàng cuối chèn 3 ảnh trang trí.</p> <ul style="list-style-type: none"> - Ô giữa của hàng 2 bảng 1, chỉnh độ rộng 60%. Chèn vào ô này bảng 3 kích thước 100%, có 3 hàng 1 cột. Ô trên cùng bảng 3 để màu vàng đậm. Ô giữa bảng 3 gõ nội dung theo yêu cầu và chèn ảnh bản đồ Việt Nam (nhỏ) sau đó đặt liên kết đến tệp tin ảnh bản đồ Việt Nam lớn. Ô dưới cùng bảng 3 để màu giống ô trên cùng và chèn chữ chạy - Ô cuối của hàng 2 bảng 1, chỉnh độ rộng 20%, màu ô vàng nhạt. Chèn vào ô này bảng 4. Bảng 4 có kích thước 100%, có 1 cột và 6 hàng. Chèn xen kẽ nội dung và ảnh minh họa vào 6 hàng này. - Lưu lại nội dung. 	
4	Thiết trang lichsu.htm ké	<ul style="list-style-type: none"> - Save as trang index thành trang lichsu.htm và lưu vào thư mục HTML. - Sửa lại đường dẫn file ảnh. - Sửa lại các đường dẫn của liên kết. <ul style="list-style-type: none"> - Xoá nội dung ô giữa bảng 3, nhập nội dung cho trang lịch sử. - Căn chỉnh định dạng. - Tạo liên kết bookmark để chuyển giữa các phần nội dung. 	<ul style="list-style-type: none"> - Giao diện theo mẫu. - Bộ cục hợp lý. - Sử dụng chính xác các thẻ - Màu sắc hài hoà. - Nội dung chính xác.

		<ul style="list-style-type: none"> - Lưu lại nội dung. 	
5	Kiểm tra	<ul style="list-style-type: none"> - Duyệt thử web - Chính sửa cho đúng yêu cầu. 	<ul style="list-style-type: none"> - Các liên kết hoạt động tốt. - Hiển thị văn bản chính xác.

4. Tổ chức thực hiện.

- Giáo viên thao tác mẫu.
- Học sinh thực hiện trên máy dưới sự hướng dẫn của giáo viên.

5. Kiểm tra đánh giá

- Website đúng cấu trúc: 1 điểm.
- Trang index.htm đúng mẫu, màu sắc hài hòa, liên kết tốt và nổi bật: 5 điểm.
- Trang lichsu.htm đúng mẫu, liên kết tốt giữa các phần: 4 điểm

CÂU HỎI VÀ BÀI TẬP CỦA BÀI 1

Trắc nghiệm

1. *Phần mở rộng của tập tin HTML là ?*
 - a. Htm.
 - b. Html.
 - c. Cả 2 câu trên đều đúng.
2. *Tùy HTML là từ viết tắt của từ nào?*
 - a. Hyperlinks and Text Markup Language.
 - b. Home Tool Markup Language.
 - c. Hyper Text Markup Language.
 - d. Tất cả đều sai.
3. *Đâu là tag tạo ra tiêu đề web kích cỡ lớn nhất.*
 - a. <heading>
 - b. <h1>
 - c. <h6>
 - d. <head>
4. *Đâu là tag tạo ra tiêu đề web kích cỡ nhỏ nhất.*
 - a. <heading>
 - b. <h1>
 - c. <h6>
 - d. <head>
5. *Đâu là tag để xuống dòng trong web?*
 - a. <lb>
 - b.

 - c. <break>
6. *Đâu là tag để để giữ nội dung cố định trên dòng trong trang web?*
 - a. <wrap>
 - b.

 - c. <nobreak>
 - d. <nobr>
7. *Đâu là tag tạo ra màu nền của web?*
 - a. <body color="yellow">
 - b. <body bgcolor="yellow">
 - c. <background>yellow</background>

8. Đâu là tag tạo ra chữ in đậm

- a.
- b. <bold>
- c. <bld>
- d. <bb>

9. Đâu là tag tạo ra chữ in nghiêng

- a. <ii>
- b. <italics>
- c. <i>

10. Đâu là tag tạo ra gạch chân chữ

- a. <underline>
- b. <u>
- c. <i>
- d. Tất cả đều sai

11. Đâu là tag tạo ra liên kết (link) trong web

- a. W3Schools.com
- b. <a>http://www.w3schools.com
- c. W3Schools
- d. W3Schools.com

12. Đâu là tag tạo ra liên kết đến Email?

- a.
- b.
- c. <mail>xxx@yyy</mail>
- d. <mail href="xxx@yyy">

13. Làm sao để khi kích chuột vào link thì tạo ra cửa sổ mới?

- a.
- b.
- c.

14. Đâu là những tag dành cho việc tạo bảng?

- a. <table><tr><td>
- b. <thead><body><tr>
- c. <table><head><tfoot>
- d. <table><tr><tt>

15. Đâu là tag căn lề trái cho nội dung 1 ô trong bảng

- a. <tdleft>

- b. <td valign="left">
- c. <td align="left">
- d. <td leftalign>

16. Đâu là tag tạo ra 1 danh sách đứng đầu bằng số?

- a.
- b. <list>
- c.
- d. <dl>

17. Đâu là tag tạo ra 1 danh sách đứng đầu bởi dấu chấm?

- a. <list>
- b.
- c.
- d. <dl>

18. Tag nào dùng để chèn 1 hình vào web?

- a. <image src="image.gif">
- b. image.gif
- c.
- d.

19. Tag nào dùng để tạo hình nền cho web?

- a. <body background="background.gif">
- b. <background img="background.gif">
- c.

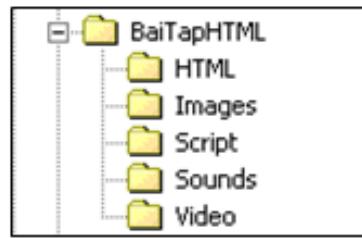
20. Để những liên kết chưa được kích hoạt có màu nâu thì sử dụng cú pháp nào sau đây:

- a. điểm đặt liên kết
- b. điểm đặt liên kết
- c. <body link="brown"> điểm đặt liên kết </body>
- d. <body Vlink="brown">.

Bài tập thực hành

Yêu cầu chung

- Tạo cây thư mục có cấu trúc như sau, lưu giữ các tập tin HTM vào thư mục HTMLS.



- Nội dung văn bản sử dụng bảng mã Unicode.
- Mỗi thư mục sẽ lưu 1 loại tập tin, lưu giữ bài tập để sử dụng về sau.

BÀI TẬP 1

Sử dụng trình soạn thảo NotePad soạn thảo nội dung sau lưu vào thư mục **HTML**.

- Đặt tên là: **Wellcome.htm**.
- Save as type: All Files.
- Encoding: UTF-8.

<HTML>

<HEAD>

<TITLE> Wellcome to HTML </TITLE>

</HEAD>

<BODY>

Chào mừng bạn đã đến với ngôn ngữ thiết kế web HTML

(Hypertext Makup Language)

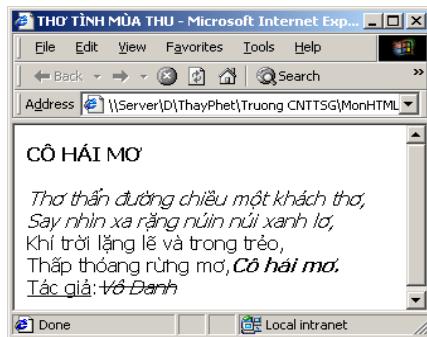
</BODY>

</HTML>

- Mở xem kết quả trang HTML
- Xem Source Code trang web từ trình duyệt.
- Mở tập tin HTML từ trình soạn thảo và lưu lại với tên khác là: Wellcom2.HTM
- Điều chỉnh nội dung hiển thị trong trang là: Chúc các bạn học tốt ngôn ngữ HTML.

BÀI TẬP 2

Sử dụng ngôn ngữ HTM soạn thảo trang Web có nội dung và định dạng theo mẫu:



Yêu cầu:

- Có nội dung thanh tiêu đề, định dạng đậm, nghiêng, gạch chân, gạch ngang chữ.
- Có phân cách các đoạn, xuống dòng cho mỗi câu thơ, có câu ghi chú
- Có định nội dung bài thơ không bị ngắt dòng khi độ rộng cửa sổ trình duyệt không đủ.
- Lưu tập tin tên Cohaimo.htm

BÀI TẬP 3

Sử dụng ngôn ngữ HTM soạn thảo trang Web có nội dung và định dạng theo mẫu:

Văn bản có chỉ số Trên/Dưới

$$AX^2+BX+C=0$$

Hôm nay thời tiết: 36⁰C

$$H_2O + CaCO_3$$

Yêu cầu:

- Dòng 1 cỡ chữ 4 in đậm
- Lưu tập tin tên: ChisoTrenduo.htm

BÀI TẬP 4.

Sử dụng ngôn ngữ HTM soạn thảo trang Web có nội dung và định dạng theo mẫu:

Các ký hiệu đặc biệt:

Bản quyền: ©
Thương hiệu: ™
Thương hiệu đã đăng ký: ®
Email: phetcm@yahoo.com

Yêu cầu:

- Các ký hiệu sử dụng mã tên hay mã code
- 2 đường kẽ ngang không bóng, kích thước 100% và 50% cửa sổ
- Có màu đỏ cho các ký tự đặc biệt.

- Màu nền xanh nhạt. tất cả văn bản không bị ngắt dòng.
- Lưu tập tin tên: Kyhieudacbiet.htm

BÀI TẬP 5

Sử dụng ngôn ngữ HTM soạn thảo trang Web có dòng văn bản trên trang có hiệu ứng chuyển động:

- Lắp liên tục từ trái sang phải, chữ đỏ, cỡ 4.
- Loại hiệu ứng: Đến viền trang hiệu ứng chuyển động ngược lại (Alternate).
- Lưu tập tin tên: Hieuungvb.htm

BÀI TẬP 6

Sử dụng ngôn ngữ HTM soạn thảo trang Web có nội dung và định dạng theo mẫu:

CHƯƠNG TRÌNH ĐÀO TẠO THIẾT KẾ WEB
<ul style="list-style-type: none"> • Ngôn ngữ HTML & FrontPage. • Xử lý ảnh & Thiết kế giao diện PhotoShopCS. • DreamWeaver MX. • Khai thác và Quản trị Domain,Hostting. • Ngôn ngữ kịch bản JavaScript.
Mọi thông tin chi tiết xin liên hệ: Văn phòng giáo vụ. Tel:08.8554258, Giaovu@netnam.vn

- Dòng đầu cỡ tiêu đề H3, màu đỏ
- Các dòng nội dung dạng danh sách không đánh số thứ tự. Màu xanh.
- Dòng cuối có sử dụng văn bản dạng chú thích, có màu khác .
- Lưu tập tin tên: ChuongTrinhDTWeb1.htm

BÀI TẬP 7

Sử dụng ngôn ngữ HTM soạn thảo trang Web có nội dung và định dạng theo mẫu:

CHƯƠNG TRÌNH ĐÀO TẠO NGÀNH WEBSITE
I. THIẾT KẾ WEBSITE <ol style="list-style-type: none"> 1. Ngôn ngữ HTML & FrontPage 2. Ngôn ngữ kịch bản JavaScript 3. Thiết kế ảnh động Media Flash MX 4. Thiết kế giao diện PhotoShop CS 5. Thiết kế WebSite Dreamweaver MX II. LẬP TRÌNH WEBSITE <ol style="list-style-type: none"> 6. Phân tích & Thiết kế CSDL 7. Lập trình CSDL ASP.NET & SQL Sever 8. Quản trị Website 9. Đề tài tốt nghiệp
Mọi thông tin xin liên hệ: Webmaster@yahoo.com

- Thiết kế dạng danh sách có đánh số thứ tự lồng nhau. Đường kẻ ngang không bóng 30% cửa sổ.
- Dòng cuối dạng văn bản chú thích.
- Lưu tập tin tên: ChuongTrinhDTWeb.htm.

BÀI TẬP 8:

Sử dụng ngôn ngữ HTM soạn thảo trang Web có nội dung và định dạng theo mẫu:

CHƯƠNG TRÌNH ĐÀO TẠO
TIN HỌC VĂN PHÒNG
Tin học căn bản & WindowsXP
Soạn Thảo văn bản MicroSoft Word 2003
Xử lý bảng tính MicroSoft Excel 2003
TIN HỌC QUẢN LÝ
Lập trình CSDL MicroSoft Access 2003
Thiết kế trình chiếu MicroSoft PowerPoint 2003

- Thiết kế dạng danh sách định nghĩa. Màu chữ tuỳ ý. Có ảnh nền tuỳ ý lưu trong thư mục Images.
- Lưu tập tin tên: ChuongTrinhDTTHVP.htm

BÀI TẬP 9.

Sử dụng ngôn ngữ HTM soạn thảo trang Web có nội dung và định dạng theo mẫu:

Máy tính "bó tay" với tài sản của Bill Gates
Bill Gates, người sáng lập Công ty phần mềm Microsoft và là người giàu nhất thế giới, cho biết cơ quan thuế của Mỹ phải lưu trữ các dữ liệu tài chính của ông vào một máy tính đặc biệt, bởi tài sản của ông quá nhiều. Phát biểu tại một hội nghị do Microsoft tổ chức ở Lisbon, thủ đô Bill Gates (trái) và Tổng thống Bồ Đào Nha rằng một máy tính thông thường không thể xử lý đúng các số liệu tài chính của ông

- Ảnh cạnh lề phải văn bản cạnh đều bao quanh lề trái.
- Kích thước ngang ảnh 130.
- Trên ảnh có câu chú thích “Bill Gates (trái) và Tổng thống Bồ Đào Nha”.
- Có ảnh nền trang mờ bất động (Ảnh được lưu trong thư mục Images của WebSite).
- Lưu tập tin tên: TaisanBill.htm.

BÀI TẬP 10

Sử dụng ngôn ngữ HTM soạn thảo trang Web có nhúng 1 tập tin nhạc (Audio, Video, Flash) theo các yêu cầu:



- Tự động phát khi mở trang web.
- Lặp lại liên tục .
- Control điều khiển canh giữa.
- Lưu tập tin tên: Nhacnen.htm.

BÀI TẬP 11.

Sử dụng ngôn ngữ HTM soạn thảo trang Web có nội dung và định dạng theo mẫu:

1.Kỹ thuật viên tin học									
2.Chuyên ngành Website									
Kỹ thuật viên tin học									
Học phần I									
Học phần II									
Học phần III									
<table border="0"> <tr> <td>I. Tin Học Căn Bản</td> </tr> <tr> <td> 1. Tổng quan cấu trúc máy tính</td> </tr> <tr> <td> 2. Hệ điều hành Windows & Internet</td> </tr> <tr> <td>II. Tin Học Văn Phòng</td> </tr> <tr> <td> 1. Microsoft Word</td> </tr> <tr> <td> 2. MicroSoft Excel</td> </tr> <tr> <td>III. Tin Học Quản Lý</td> </tr> <tr> <td> 1. Microsoft Access</td> </tr> <tr> <td> 2. MicroSoft PowerPoint</td> </tr> </table>	I. Tin Học Căn Bản	1. Tổng quan cấu trúc máy tính	2. Hệ điều hành Windows & Internet	II. Tin Học Văn Phòng	1. Microsoft Word	2. MicroSoft Excel	III. Tin Học Quản Lý	1. Microsoft Access	2. MicroSoft PowerPoint
I. Tin Học Căn Bản									
1. Tổng quan cấu trúc máy tính									
2. Hệ điều hành Windows & Internet									
II. Tin Học Văn Phòng									
1. Microsoft Word									
2. MicroSoft Excel									
III. Tin Học Quản Lý									
1. Microsoft Access									
2. MicroSoft PowerPoint									
Đầu trang									

Yêu cầu:

- Tại mục **2. Chuyên ngành Website**, tạo liên kết thực hiện mở đến trang web **chuongTrinhDTWeb.htm** (Cùng cửa sổ).
- Học phần I, II, III thực hiện tạo liên kết đến từng học phần tương ứng trong cùng trang hiện tại. Cuối trang có liên kết “Đầu trang” để về đầu trang(Thu nhỏ cửa sổ khi xem để kiểm tra các liên kết nội bộ trang).
- Lưu tập tin tên: ChuongTrinhDaoTao.htm

BÀI TẬP 12

Sử dụng ngôn ngữ HTM soạn thảo trang Web có nội dung và định dạng theo mẫu:

KẾT QUẢ HỌC TẬP				
Họ và tên Học Sinh	Năm Sinh		Điểm TB	Xếp Lộai
	Nam	Nữ		
Lê Thanh Xuân		1950	8.5	Giỏi
Phan Thế Hạ	1985		6.5	Khá
Trần Hòai Thu		1986	9.5	Xuất Sắc
Trường Lưu Đông	1984		6.0	Trung Bình
Lưu Thế Mạc	1985		4.5	Kém
Tổng số học sinh đạt:		4 Học sinh		

Yêu cầu:

- Màu nền dòng tiêu đề và dòng cuối, màu chữ tùy ý
- Độ rộng Table = 600, khoảng cách giữa các ô =0 , độ dày đường viền 1, màu viền tùy ý.
- Canh lề và trộn ô đúng theo mẫu.
- Lưu tập tin tên: KetQuaHocTap.htm.

BÀI 2 : THIẾT KẾ WEB TĨNH VỚI PHẦN MỀM DREAMWEAVER

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.

A. LÝ THUYẾT

1. Tổng quan.

1.1. Giới thiệu phần mềm Dreamweaver.

Dreamweaver là một chương trình thiết kế chuyên nghiệp để biên tập các trang web (hay biên tập mã HTML) thực hiện các công việc thiết kế, quản lý và upload Website. Dreamweaver cung cấp các công cụ phác thảo trang web cao cấp, hỗ trợ các tính năng Dynamic HTML mà không cần viết các dòng lệnh giúp các bạn không biết lập trình web cũng có thể thiết kế các trang web động một cách dễ dàng, trực quan.

Các tính năng của DW có thể khái quát như sau:

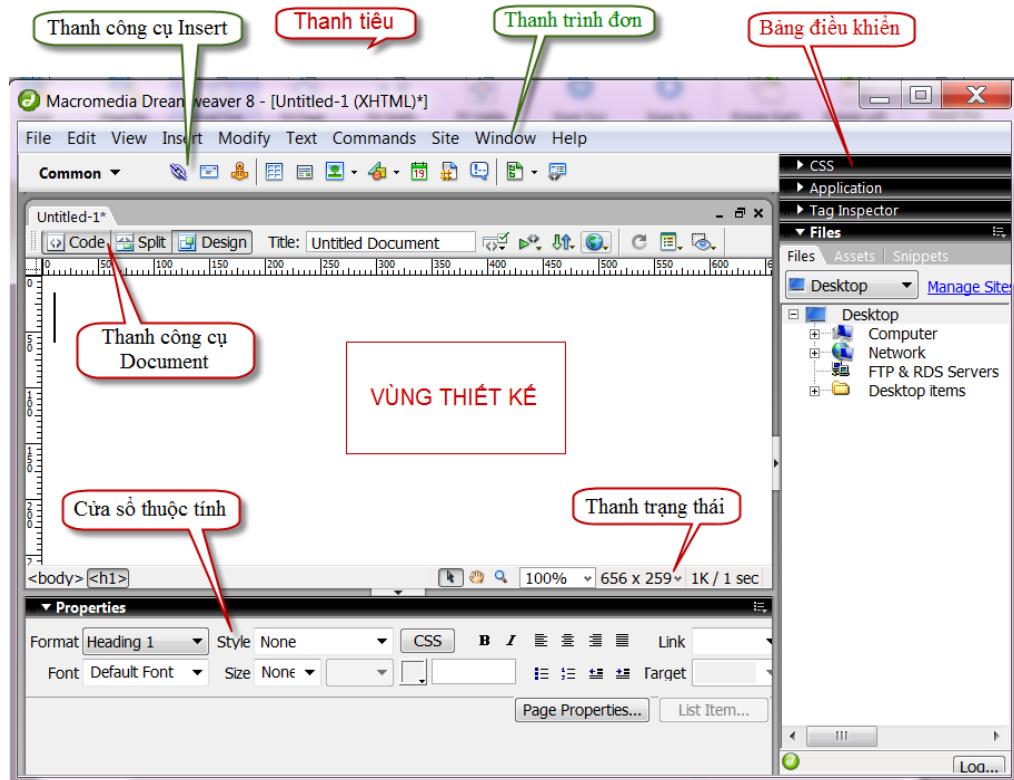
- Là một chương trình dùng để tạo ra và quản lý các trang web. Phần cốt lõi của nó là HTML.
- Là một công cụ mạnh, dễ dùng, một công cụ trực quan: Có thể bổ sung Javascript, biểu mẫu, bảng biểu, Flash, Java Applet và nhiều loại đối tượng khác mà không cần phải viết 1 dòng mã nào.
- Cung cấp chế độ làm việc Design WYSIWYG (những gì thấy là những gì có được) và chế độ làm việc mã –Code. Có thể thực hiện việc xây dựng website mà không cần quan tâm đến mã HTM hoặc có thể chỉnh sửa trực tiếp một cách linh hoạt trên mã HTML.
- Có thể mở các trang web được tạo ra từ những phần mềm thiết kế khác để thay đổi nội dung.
- Hỗ trợ HTML styles và Cascading Style Sheet(CSS) giúp bạn định dạng trang web nhằm tăng tính hấp dẫn khi duyệt các trang web này.
- Sử dụng công nghệ Roundtrip HTML để nhập các tài liệu HTML mà không cần định lại mã chương trình và bạn có thể thiết lập để Dreamweaver xoá hết và định lại khuôn thức HTML mà bạn muốn.

1.2. Giao diện và các thanh công cụ cơ bản.

1.2.1. Khởi động Dreamweaver 8.

**Start → Programs→ MacroMedia → Macromedia Dreamweaver 8
→ Create New →HTML.**

1.2.2. Màn hình làm việc.



Hình 2.1. Màn hình làm việc.

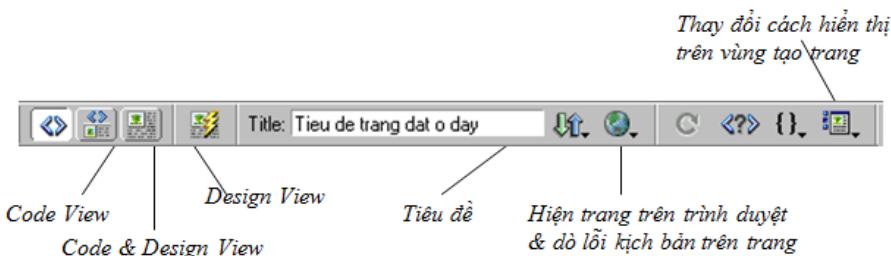
- Thanh tiêu đề : Chứa tên tập tin đang thiết kế.
- Thanh trình đơn: Chứa các tập lệnh thực hiện các chức năng
- Thanh công cụ Insert: Chứa nhóm các công cụ tạo và định dạng các đối tượng.
 - Thanh công cụ Document: Các hình thức trình bày trang đang thiết kế.
 - Thanh trạng thái: Báo khu vực trên trang đang chọn làm việc và kích thước trang.
 - Cửa sổ thuộc tính (Properties): Các thuộc tính của đối tượng đang được chọn.
 - Bảng điều khiển: Trình bày các bảng điều khiển

1.2.3. Các thanh công cụ.

Dreamweaver cung cấp các thanh công cụ giúp cho việc thực hiện các thao tác trên trang một cách nhanh chóng.

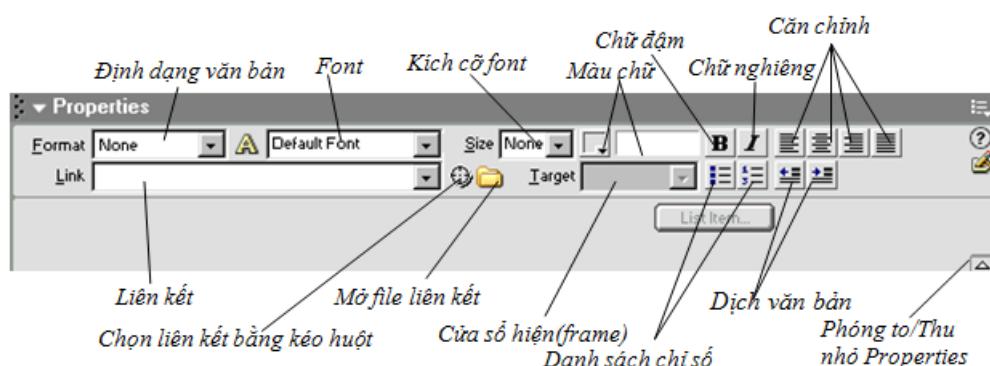
❖ *Thanh công cụ Document*

- Mở thanh công cụ **Document** : View → Toolbars → Document.



Hình 2.2 : Thanh công cụ Document.

- **Chế độ Code View:** hiển thị trang ở chế độ mã lệnh. Chế độ này thích hợp cho trường hợp viết mã kịch bản hoặc sửa đổi trực tiếp trên mã HTML
- **Chế độ Design View:** hiển thị trang ở chế độ thiết kế trực quan. Ở chế độ này trang được thiết kế như thế nào thì khi hiển thị trên trình duyệt nó cũng giống như vậy (trừ một số thành phần không có khả năng hiển thị). Chế độ này thích hợp cho việc thiết kế trang.
- **Chế độ Code and Design View:** hiển thị cả 2 cửa sổ mã lệnh và thiết kế của trang.
- **Preview/Debug in Browser:** Hiển thị trang web trên trình duyệt (phím tắt: F12 hiển thị trình duyệt mặc định, Ctrl-F12 hiển thị trình duyệt phụ).
- ❖ *Cửa sổ thuộc tính Properties:*
- Đây là cửa sổ theo dõi các thuộc tính của các đối tượng được chọn.
- Mở cửa sổ thuộc tính : Window → Properties (Ctrl + F3)

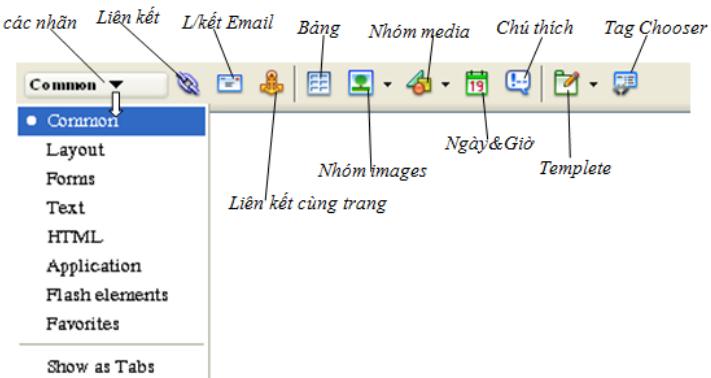


Hình 2.3 : Cửa sổ thuộc tính Properties.

❖ *Thanh Insert:*

Dùng để thêm phần lớn các thành phần của Dreamweaver vào trang Web. Các thành phần này được chia thành các nhóm đặt trong thẻ.

- Mở thanh công cụ Insert : Window → Inssert (Ctrl + F3)



Hình 2.4. Thanh công cụ Insert.

1.2.4. Mở một số bảng điều khiển quản lý các đối tượng

- Quản lý các CSS: Window / CSS Styles (Shift+F11)
- Quản lý các Layer: Window / Layers (F2)
- Quản lý các site: Window / File(F8)
- Quản lý các Template, Libary: Window / Assets (F11)
- Quản lý các Behaviors: Windows / Behaviors (Shift+F3)
- Quản lý các Frames: Windows / Frames (Shift+F2)

1.3. Làm việc với Website.

1. 3.1. Khái quát.

Site là vùng lưu trữ cục bộ trên máy tính chứa các tập tin của 1 Website bao gồm các thư mục con chứa các tập tin âm thanh, hình ảnh . Việc tổ chức Site cẩn thận ngay từ lúc đầu sẽ tiết kiệm thời gian về sau. Dreamweaver ngoài việc cho phép tạo ra các trang Web độc lập thì nó còn có khả năng rất mạnh là quản lý Site chứa các trang Web đó.

Việc quản lý các trang theo Site rất cần thiết bởi ngoài việc mang tính tập trung, các liên kết trên trang sẽ có được đường dẫn tương đối, nhờ đó chúng sẽ không bị đứt gãy khi đưa lên mạng (upload). Sau khi đã đưa lên mạng nhờ việc quản lý Site ta có thể cập nhật Site một cách dễ dàng. Ngoài ra khi quản lý theo Site ta có thể thực hiện những hiệu ứng ảnh hưởng đến nhiều trang một lúc.

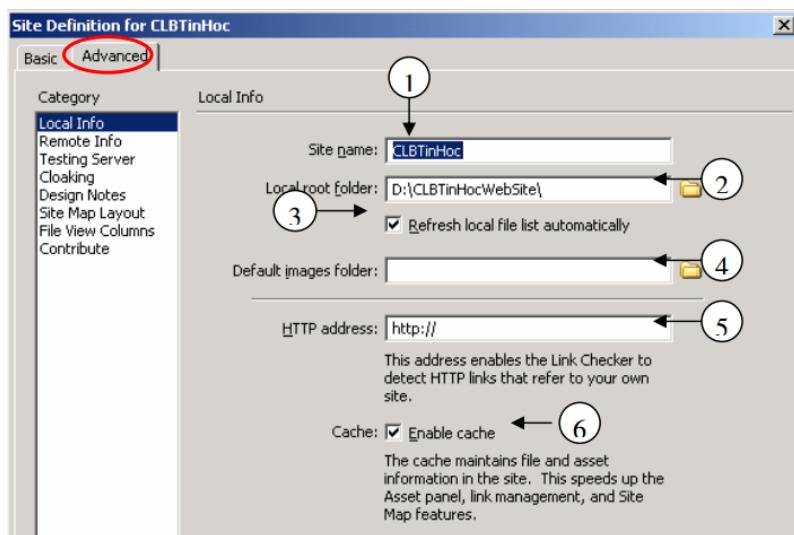
1.3.2. Tạo mới Site:

Trước khi xây dựng một Website ta cần phải tạo một mục Site tương ứng. Các bước thực hiện việc định nghĩa Site như sau:

- Chọn Site → Manage sites → New → Site.

Hộp thoại **Site Definition for Unnamed Site** hiển thị, có hai lớp **Basic** và **Advanced**. Lớp **Advanced** trình bày tổng quát, còn lớp **Basic** trình bày dạng từng bước (sẽ không có sự khác biệt nhau giữa hai lớp này.)

- Chọn lớp **Advanced**.



Hình 2.5: Tạo mới site

- Xác định các mục trong hộp thoại:

- + [1] **Site name**: nhập tên cho site, tên của site sẽ được xuất hiện trên cửa sổ site và trên menu Site / Open Site.
- + [2] **Local Root Folder**: xác định thư mục trên ổ đĩa cục bộ, nơi lưu trữ các tập tin, mẫu tin và các hạng mục của thư viện có trong Site. Kích lên biểu tượng thư mục để duyệt và chọn thư mục hoặc nhập đường dẫn và tên thư mục vào trong khung nhập.
- + [3] **Refresh Local File List Automaticaly**: Chỉ định danh sách liệt kê tập tin cục bộ có được tự động tinh chỉnh mỗi khi bạn sao chép các tập tin vào Site cục bộ hay không.
- + [4] **Default Images Folder**: chỉ định thư mục lưu trữ tập tin ảnh trong Site, bạn kích vào biểu tượng thư mục và duyệt chọn đường dẫn đến thư mục tập tin ảnh trên site hoặc nhập vào đường dẫn trong khung nhập.
- + [5] **Cache**: chỉ định Cache cục bộ có được tạo hay không để cải thiện tốc độ liên kết và quản lý tài nguyên của site. Điều này là tùy thuộc vào bạn (nếu bạn không chọn, Dreamweaver sẽ hỏi bạn lại việc này trước khi nó tạo

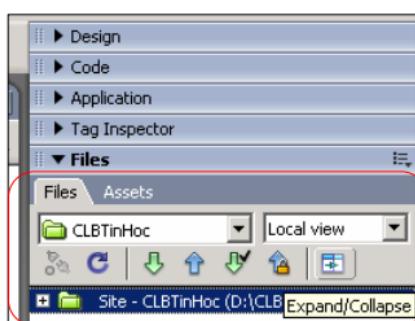
site). Bạn nên chọn mục này, tuy nhiên nếu máy bạn không đủ bộ nhớ RAM bạn có thể bỏ qua tuỳ chọn này.

- Sau khi đã chọn xác định xong các mục trong hộp thoại, bạn kích OK để tạo một site.

1.3.3. Quản lý site

❖ *Bảng điều khiển file*

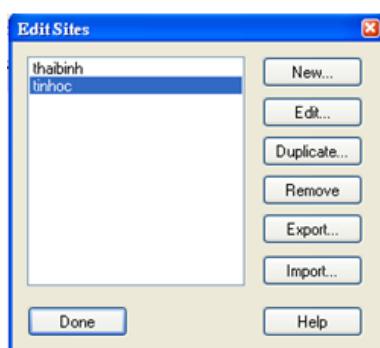
Sử dụng cửa sổ Site cho các thao tác quản lý tập tin trong Site như : Tạo trang HTML, xem, mở và di chuyển; Tạo các thư mục và xoá các thư mục. Bạn cũng có thể sử dụng cửa sổ Site để chuyển các tập tin giữa Site cục bộ và Site ở xa và bạn có thể phác thảo sơ đồ điều hướng cho Site của bạn bằng cách sử Site map (bản đồ site). Để mở cửa sổ Site, chọn menu **Window / Files** (hoặc F8).



Hình 2.6: *Bảng điều khiển file*.

❖ *Cửa sổ quản lý các site.*

Do phải làm việc với nhiều site nên Dreamweaver cũng cung cấp các chức năng để quản lý chúng. Để mở cửa sổ quản lý site, vào menu **Site** chọn tiếp **Manage Sites**.



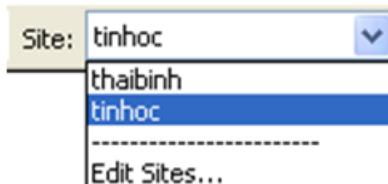
Hình 2.7: *Quản lý các site*

- + **New:** tạo một mục site mới.
- + **Edit:** thay đổi thông tin của một mục site đã có.
- + **Duplicate:** tạo bản copy của một mục site.
- + **Remove:** xoá bỏ một mục site.

- + **Export:** lưu trữ các thông tin của mục site ra file.
- + **Import:** nhập site được định nghĩa trong file vào danh sách site.

❖ *Làm việc với một site*

Khi trên cùng một máy có nhiều người cùng sử dụng Dreamweaver để chọn site làm việc khi có nhiều site được xây dựng đồng thời, thì khi mở Dreamweaver ta phải chọn đúng site cần làm việc từ danh sách site trong bảng điều khiển site ở cửa sổ site.



Hình 2. 8: chọn site làm việc

❖ *Các thao tác quản lý Local Site*

Các thao tác quản lý thư mục và tập tin tại cửa sổ **Local Site** được ánh xạ trực tiếp đến cây thư mục trong đĩa cứng. Khi mới tạo Site, mặc định Site là rỗng. Bạn cần phải tạo thêm các thư mục, file khác cho nó.

❖ *Tạo các thư mục trong site.*

- Kích phải chuột vào tên Site (hoặc chọn thư mục trong site nếu ta muốn tạo thư mục con của nó).
- Chọn **New folder**.
- Đặt tên thư mục.

❖ *Tạo trang Web trong Site.*

- Kích phải chuột vào tên Site (hoặc chọn thư mục trong site nếu ta muốn tạo trang con của nó).
- Chọn **New file**.
- Đặt tên trang web.

❖ *Xoá tệp tin hoặc thư mục.*

- Kích phải chuột vào tệp tin hoặc thư mục cần xoá.
- Chọn **Edit**
- Chọn **Delete**.

❖ *Đổi tên tệp tin hoặc thư mục.*

- Kích phải chuột vào tệp tin hoặc thư mục cần đổi tên.
- Chọn **Edit**
- Chọn **Rename**.
- Gõ tên mới.

❖ **Nhân đôi tệp tin hoặc thư mục.**

- Kích phải chuột vào tệp tin hoặc thư mục cần nhân đôi.
- Chọn **Edit**
- Chọn **Duplicate**.
- Gõ tên mới.

❖ **Sao chép / di chuyển tệp tin hoặc thư mục.**

- Kích phải chuột vào tệp tin hoặc thư mục cần copy hoặc di chuyển.
- Chọn **Edit**
- Chọn **Copy** hoặc **Cut**.
- Mở nơi lưu trữ mới.
- Dán (**Paste**).

❖ **Đặt trang chủ.**

Trang chủ là trang khởi đầu của Site để từ đó có thể chuyển tới các trang khác trong Site (thường được nạp đầu tiên khi trình duyệt truy cập vào Site). Dreamweaver không tự biết trang nào là trang chủ, vì thế khi xây dựng ta phải tự chỉ ra. Chỉ sau khi đặt trang chủ ta mới xem được cấu trúc của Site ở chế độ **MapView**.

Để đặc một trang làm trang chủ ta thực hiện như sau:

- Kích phải chuột lên trang Web cần đặt làm trang chủ từ cửa sổ Site.
- Chọn **Set As Home Page**.

2. Trang và văn bản trên trang

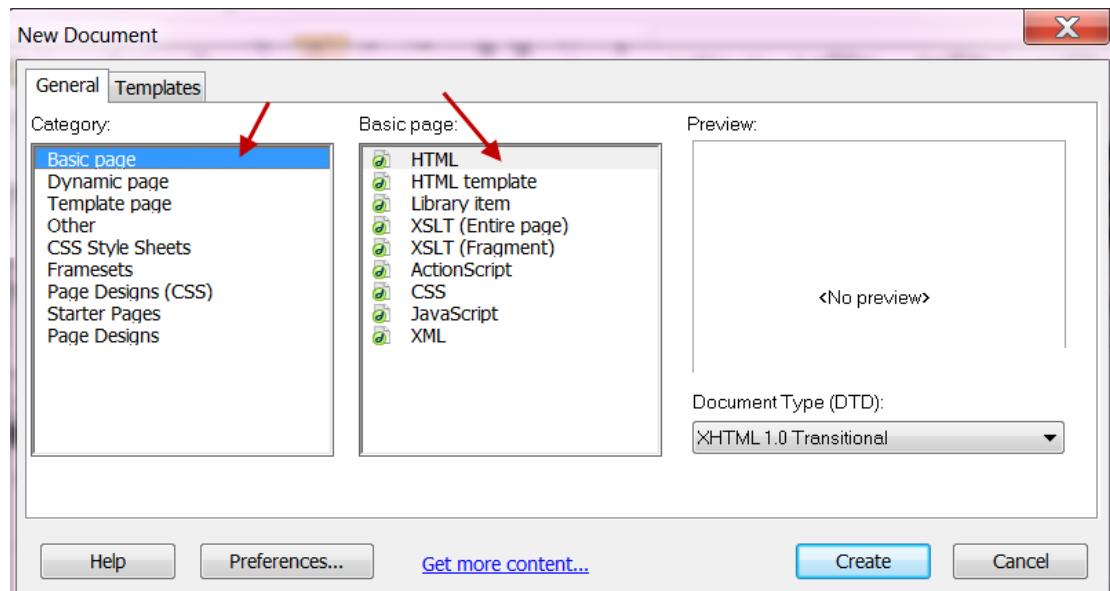
2.1. Các thao tác quản lý trang web.

Khi bạn đã tạo một Site cục bộ, bạn có thể tạo các tài liệu để đưa vào Site. Khi bạn tạo và làm việc với các tài liệu, Dreamweaver sẽ tự động phát sinh mã nguồn HTML và JavaScript cơ bản của chúng.

Ta có thể tạo mới, đổi tên, xoá các trang web trong cửa sổ quản lý Site như trên hoặc thực hiện trên hệ thống menu như sau:

2.1.1. Tạo 1 trang web mới.

- Chọn File → New (Ctrl-N). Hộp thoại Hộp thoại New Document hiển thị



Hình 2.9: Cửa sổ tạo mới trang web.

- Chọn lớp General.
- Trong Category: chọn Basic Page.
- Trong Basic Page: chọn HTML
- Nhấn nút Create.

Trang tài liệu mới hoàn toàn trống không, nó chứa các thẻ **html**, **head** và **body** để khởi tạo.

2.1.2. Mở một tập tin HTML đang tồn tại.

- Chọn menu **File / Open**, chọn file cần mở và nhấn **Open** để mở.
- Nếu tập tin được tạo bằng Word hoặc XML thì bạn chọn menu lệnh **File / Import / Word HTML ...**
- Bạn không thể trực tiếp chèn tập tin Word (*.doc) vào Dreamweaver. Nếu bạn muốn chèn các nội dung của tập tin Word, bạn khởi động Word và lưu tập tin thành dạng HTML trước khi chèn tập tin HTML kết quả vào Dreamweaver.

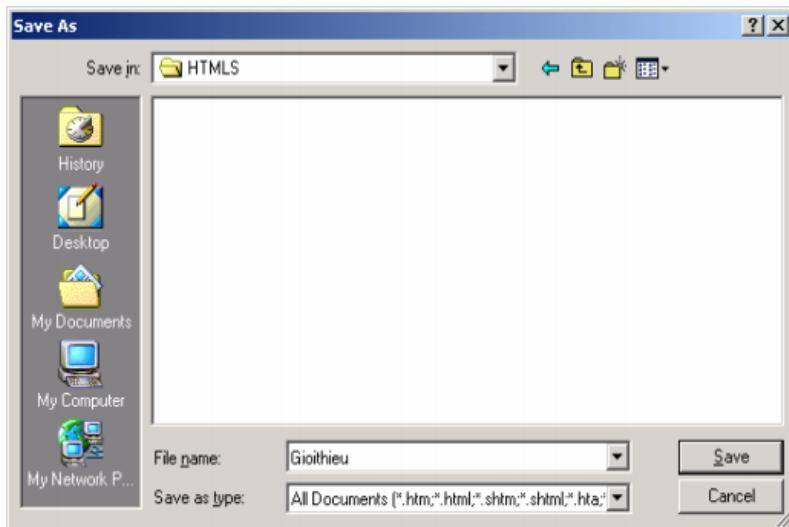
2.1.3. Đóng 1 trang tập tin HTML.

- Đóng tài liệu hiện thời chọn **File / Close**
- Nếu như trường hợp mở đồng thời nhiều tài liệu thì có thể đóng tất cả các tài liệu đang mở bằng cách sau:
 - + Phải chuột lên nhãn của tài liệu
 - + Chọn **Close all**.

2.1.4. Lưu một trang tài liệu.

- Chọn menu lệnh **File / Save**.
- Cửa sổ Save hiển thị (hình 2.10).

- Nhập tên cho tập tin và chọn thư mục chứa tập tin cần lưu.
- Kích nút **Save** để lưu tập tin.



Hình 2.10: Cửa sổ lưu tệp HTML

Nếu muốn lưu tập tin với một tên khác, bạn chọn menu lệnh **File → Save as** sau đó gõ tên tệp tin và chọn nơi lưu trữ mới sau đó kích **Save**

2.1.5. Hiển thị trang trên trình duyệt:

Dreamweaver chỉ làm nhiệm vụ thiết kế trang và tổ chức site. Nếu muốn thấy được hình ảnh và hoạt động thực sự của trang Web ta cần phải mở trang Web trên trình duyệt.

Để hiển thị trang Web trên trình duyệt chính (trình duyệt mặc định) nhấn F12, còn để hiển thị trang trên trình duyệt phụ (nếu có) nhấn Ctrl-F12.

Có thể thay đổi trình duyệt hiển thị trong mục Preview in Browser trong cửa sổ Preferences (Edit → Preferences hoặc Ctrl-U).

2.2. Thiết lập thuộc tính cho trang web.

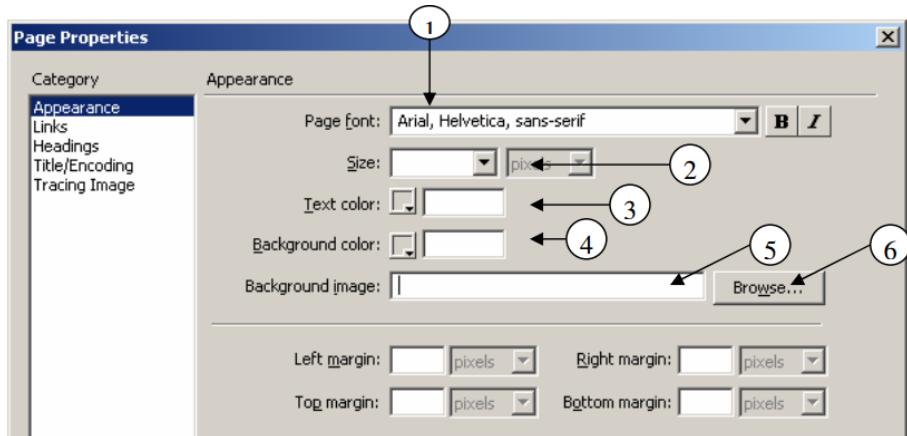
Tiêu đề trang, ảnh nền, màu nền, màu văn bản, liên kết và các lề là các thuộc tính cơ bản của mọi tài liệu HTML.

Để thiết lập các thuộc tính cho trang Web, thực hiện như sau:

- Cách 1: Vào menu **Modify / Page Properties**
- Cách 2: Kích chuột phải trên khoảng trắng của trang đang thiết kế, bạn chọn **Page Properties** trên menu ngữ cảnh hiển thị.

Hộp thoại **Page Properties** hiển thị (*xem hình 2.11*)

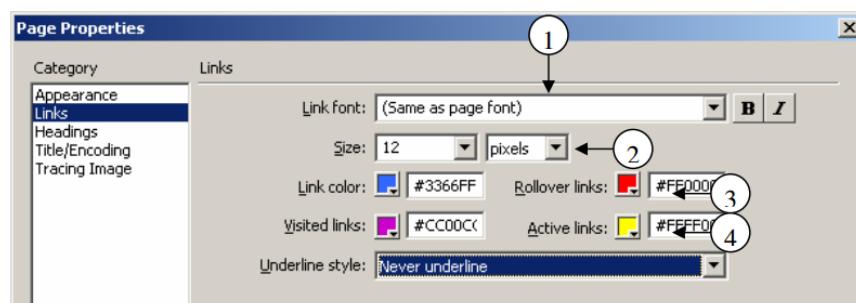
2.2.1. Thuộc tính chung cho trang: Lớp Appearance



Hình 2.11 : Hộp thoại đặt thuộc tính trang web

- (1)-**Page Font:** Font chữ mặc định cho trang.
- (2)-**Size:** Cỡ chữ mặc định cho trang.
- (3)-**Text color:** Màu chữ mặc định cho trang.
- (4)-**Background color:** Màu nền trang.
- (5)-**BackGround Image:** Ảnh nền cho trang (Ảnh phải nằm trong thư mục của Site).
- (6)-**Left/Right/Top/Bottom Margin:** Lê trang.

2.2.2. Các thuộc tính chung cho liên kết : Lớp Link.

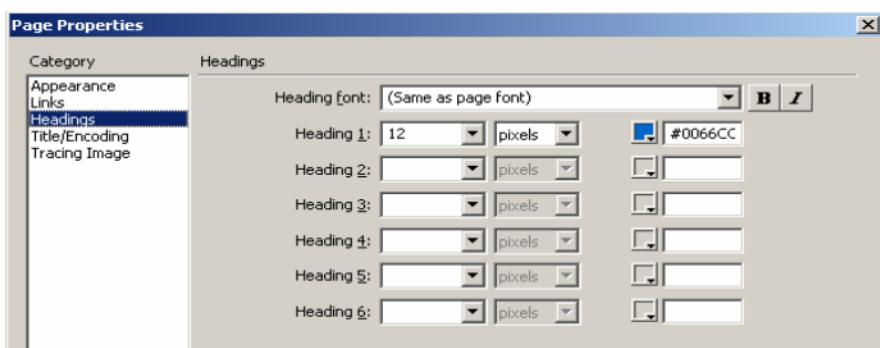


Hình 2.12 : Thiết lập thuộc tính liên kết.

- (1) **Link Font:** Font chữ cho văn bản liên kết.
- (2) **Size:** Cỡ chữ cho văn bản liên kết.
- (3) **Link color:** Màu chữ cho văn bản liên kết.
 - **Visited links:** Màu chữ cho văn bản liên kết khi đã viếng thăm.
 - **Rollover links :** Màu chữ cho văn bản liên kết khi rê chuột qua.
 - **Active links :** Màu chữ cho văn bản liên kết khi kích chuột.
- (4) **Underline style:** Kiểu gạch chân cho văn bản liên kết.
 - **Never . . .:** Không bao giờ có gạch chân.
 - **Always . . .:** Luôn có gạch chân.
 - **Show . . .:** Hiện thị đường kẽ ngang khi rê chuột vào.
 - **Hide. . . .:** Ẩn gạch chân khi rê chuột vào.

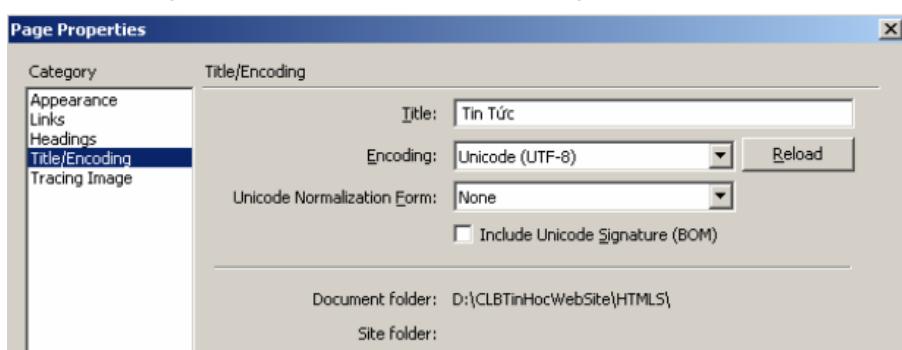
2.2.3. Tạo mẫu văn bản đề mục : Lớp Headings

Tạo các mẫu văn bản đề mục (Heading1, . . .) để dùng khi định dạng.



Hình 2.13: Thiết lập Headings.

2.2.4. Tiêu đề trang: Chọn mục Title-Encoding



Hình 2.14 : Thiết lập tiêu đề trang.

- **Title:** Nhập tiêu đề cho trang vào mục.
- **Encoding:** Chọn **UTF-8** (nếu trên trang có sử dụng văn bản Unicode)
- ❖ **Đặt tiếng việt cho toàn site**

Muốn đặt tiếng Việt cho tất cả các trang trên site được tạo ra sau đó ta thực hiện như sau:

- Chọn **Edit → Preferences (Ctrl-U)**.
- Chọn mục **New Document**.
- Ở mục **Default Encoding** chọn **Unicode(UTF-8)**.
- Chọn **OK**.

2.3. Nhập văn bản

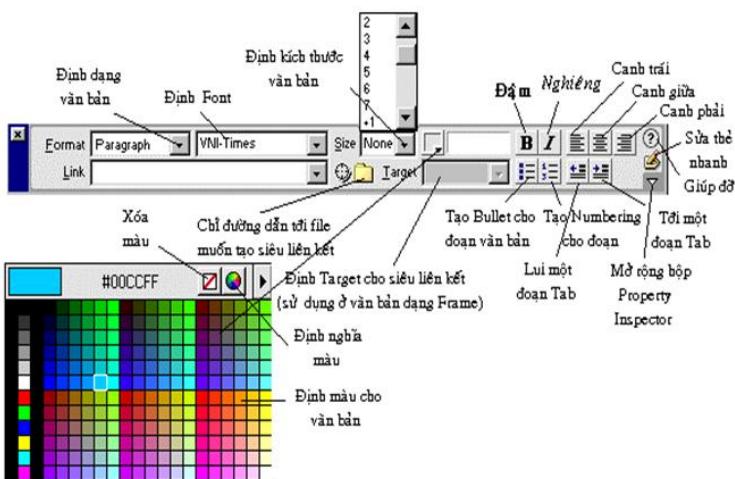
Việc nhập và định dạng văn bản trong Dreamweaver 8 hoàn toàn giống như việc sử dụng một trình soạn thảo văn bản thông thường. Bạn nhập văn bản trực tiếp trong cửa sổ Document. Có thể sao chép văn bản từ một ứng dụng khác bằng cách chọn lệnh Edit / Paste để chép văn bản vào vị trí điểm chèn. Dreamweaver 8 sẽ không quan tâm đến định dạng của văn bản được sao chép, nhưng nó sẽ nhận dạng các dấu ngắt dòng.

Lưu ý:

- Đối với những trang web phức tạp hoặc những trang do nhiều người cùng xây dựng thì đôi khi cần có ghi chú. Các ghi chú này sẽ chỉ xem được khi ta thiết kế còn khi hiển thị trên trang web chúng sẽ không xuất hiện. Để thêm ghi chú ta chọn biểu tượng  trên nhãn **Common** của thanh Insert. Nhập nội dung ghi chú vào ô Comment. Trên trang sẽ xuất hiện biểu tượng  để báo rằng có ghi chú tại đó.

2.4. Định dạng văn bản

- Chọn menu **Window/Properties** để mở hộp thoại **Property Inspector**



Hình 2.15 : Cửa sổ định dạng văn bản.

- Chọn văn bản. Nếu không có văn bản được chọn thì các thay đổi được áp dụng cho văn bản được gõ vào sau đó.
- Thiết lập các tùy chọn sau:
 - + **Để thay đổi Font:** Chọn loại font từ **Properties** (hoặc từ menu **Text/Font**).
 - + **Để thay đổi kiểu Font :** Kích các kiểu chữ Bold hoặc Italic, trên **Properties** (hoặc chọn kiểu Font trên menu **Text/Style**).
 - + **Để thay đổi cỡ chữ:** Chọn một kích cỡ từ **Properties** (hoặc từ menu **Text / Size**) các kích cỡ từ 1 đến 7 (mặc định là 3).
 - + **Để đổi màu văn bản:** Kích chuột vào ô **Text Color** trên **Properties** (hoặc chọn Menu **Text/Color**) để mở bảng màu sau đó chọn 1 màu trong bảng.
 - + **Để canh lề cho đoạn văn bản:** chọn một dạng canh lề trên **Properties** (hoặc chọn menu **Text/Align**).

+ **Thay đổi tổ hợp Font:** để thiết lập danh sách các tổ hợp font được hiển thị:

- Sử dụng lệnh **Edit Font List** trong khung font của cửa sổ **Properties** (hoặc trong **Text / Font**).
- Chọn một font trong danh sách **Available** (các font được phép).
- Nhấn chuyển sang danh sách **Chosen Fonts**.

2.5. Tạo danh sách cho văn bản.

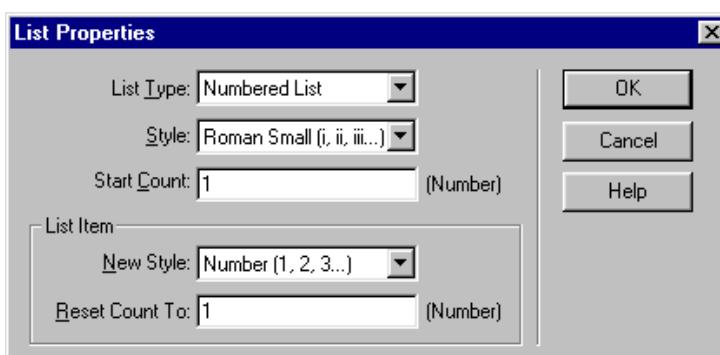
2.5.1. Tạo danh sách mới

- Chọn văn bản để tạo danh sách.
- Kích nút **Unordered List** hay **Ordered List** trên **Properties Inspector**, hay chọn **Text / List** và chọn một kiểu của danh sách.
- Đặt điểm chèn ở vị trí muốn tạo danh sách(hoặc 1 nhóm các dòng).
- Kích nút **Unordered List** hay **Ordered List** trên cửa sổ **Properties** hoặc vào **Menu Text /List** và chọn kiểu của danh sách.
- Ấn Enter để chuyển sang.

2.5.2. Chính sửa các thuộc tính của danh sách:

Các thuộc tính danh sách xác định hình thức hiển thị của toàn bộ danh sách. Để xem hay hiệu chỉnh các thuộc tính của danh sách:

- Chọn một hạng mục trong danh sách và chọn **Text /List /Properties** hoặc kích nút **List Item** trên **Properties Inspector**.
- Trong hộp thoại **List Properties** (*hình 2.16*) có các thuộc tính sau:



Hình 2.16: hộp thoại List Properties.

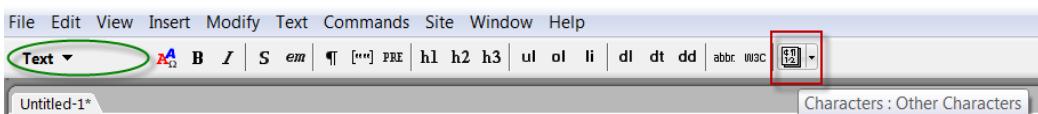
- + **List Type:** kiểu đánh dấu của danh sách
- + **Style:** kiểu của ký tự dùng đánh dấu danh sách
- + **Start Count:** đặt biến bắt đầu đánh dấu cho hạng mục đầu tiên trong danh sách.
- + **New Style:** cho phép bạn chọn một kiểu ký tự mới đánh dấu bắt

đầu từ hạng mục được chọn

- + **Reset Count To:** đặt giá trị bắt đầu đánh dấu từ hạng mục được chọn.

2.6. Chèn ký tự đặc biệt.

- Đặt điểm chèn tại vị trí muốn chèn ký hiệu đặc biệt.
- Chọn **Insert/HTML/Special Charcters** hoặc kích nút **Character** trong lớp **Text** của thanh công cụ Insert.



Hình 2.17: chèn ký tự đặc biệt

- Chọn ký tự đặc biệt muốn chèn và kích OK.

3. Multimedia trên trang Web

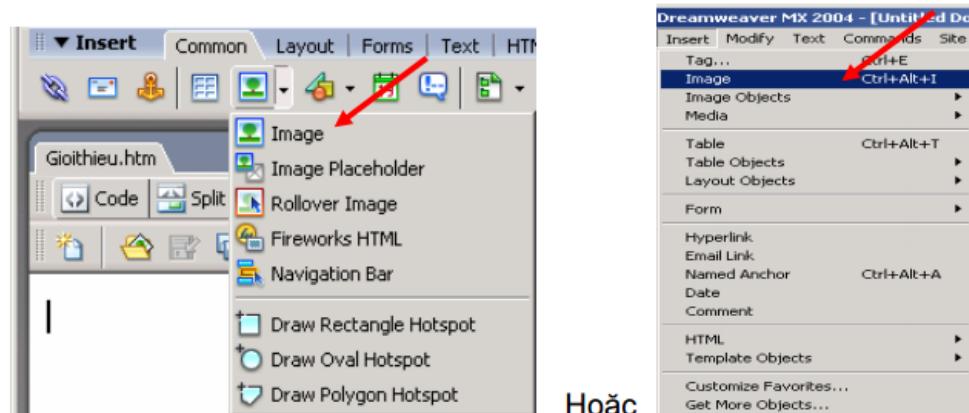
3.1. Hình ảnh.

3.1.1. Chèn hình ảnh.

Web sử dụng 3 kiểu file ảnh chuẩn là *.JPG, *.GIF, *.PNG. Sở dĩ những loại file này được sử dụng bởi chúng là các dạng ảnh nén nên có kích cỡ nhỏ. Loại file ảnh *.GIF còn có thể định dạng để trở thành ảnh động, ảnh trong suốt. Khi chèn ảnh chúng ta cần một số điểm lưu ý (trình bày ở bài 1, mục 3.3.3).

Để thêm ảnh vào trang thực hiện các bước sau:

- Đặt con trỏ tại vị trí cần chèn rồi chọn biểu tượng  trên thanh công cụ **Common** hoặc trình đơn **Insert/ Image**.

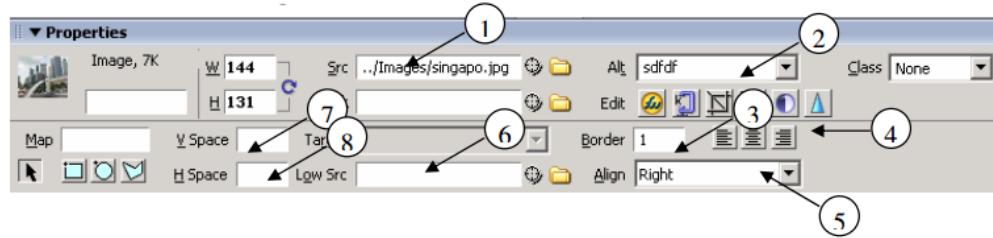


Hình 2.18 : Chèn ảnh

- Chọn file ảnh thích hợp và kích OK.

3.1.2. Định dạng hình ảnh

Sau khi chèn ảnh vào trang Web, bạn có thể kích chuột chọn ảnh để xuất hiện hộp thuộc tính **Properties Inspector** để thay đổi các thuộc tính định dạng của ảnh như sau:



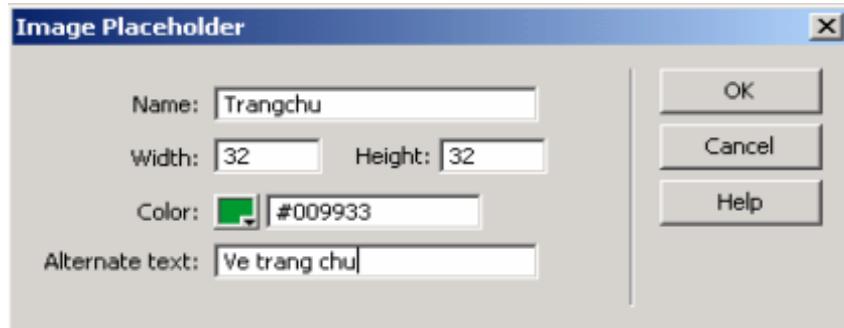
Hình 2.19 : Định dạng hình ảnh.

- [1] **Src**: Đường dẫn tập tin ảnh.
- [2] **Alt**: Câu chú thích khi rê chuột vào ảnh.
- [3] **Border**: Độ dày Khung viền quanh ảnh.
- [4] Các hình thức canh lề ảnh so với trang.
- [5] **Align**: Các hình thức canh lề ảnh so với văn bản trên trang.
- [6] **Low Src**: Đường dẫn tập tin ảnh chất lượng thấp(tải trước ảnh thật).
- [7] **Vspace** : Khoảng trống (Pixel) bên trái/phải ảnh.
- [8] **Hspace** : Khoảng trống (Pixel) phía trên/dưới ảnh.

3.1.3. Giữ toạ độ vị trí cho ảnh.

Giúp cho người thiết kế định vị trước vị trí của các ảnh sẽ chèn bổ sung sau này:

- Đặt điểm chèn ở vị trí muốn định vị.
- Kích biểu tượng **Image** → **Image Placeholder** trên thanh công cụ **Common**. (Hoặc trình đơn **Insert/ Image Object/ Images Placeholder**), cửa sổ **Images Placeholder** xuất hiện.



Hình 2.20 : Hộp thoại Image Placeholder.

- + **Name**: Đặt tên.
- + **Width/Height**: Độ rộng/ Độ cao.
- + **Color**: Màu nền.

- + **AlternateText:** Câu chú thích.
- Bổ sung hình ảnh và định dạng tại vị trí đã giữ chỗ:
 - + Chọn vị trí đã giữ chỗ.
 - + Tại cửa sổ thuộc tính **Properties** thực hiện bổ sung ảnh và khai báo các thuộc tính định dạng.

3.2. Tập tin Media (Audio, Video).

3.2.1. Khái quát về các dạng tập tin Multi Media:

- Plugin là thành phần được cài đặt thêm nhằm hỗ trợ chức năng cho trình duyệt, cung cấp nhiều cách khác nhau để xem nội dung media theo nhiều dạng khác nhau. Khi gặp các thành phần plugin trình duyệt sẽ gọi các chương trình thích hợp ra xử lý (nếu có).
- Dạng .midi (hay .mid) được cung cấp bởi nhiều trình duyệt và không yêu cầu Plug-in.
- Dạng .wav có chất lượng âm thanh tốt, được cung cấp bởi nhiều trình duyệt, không yêu cầu Plug-in. Tuy nhiên kích thước tập tin lớn.
- Dạng .aif tương tự dạng wav, cho chất lượng âm thanh tốt, có thể chạy hầu hết các trình duyệt mà không yêu cầu Plug-in, tuy nhiên kích thước tập tin lớn.
- Dạng .mp3,.avi, wma,wmv . . . là dạng nén làm tập tin âm thanh nhỏ hơn nhiều. Chất lượng âm thanh tốt. Công nghệ mới cho phép "phân luồng" tập tin nên không phải đợi toàn bộ tập tin được tải trước khi nghe nó. Để chạy các tập tin này, người sử dụng phải cài đặt trình ứng dụng như Windows Media.
- Dạng .ra, .ram, .rpm,. . có độ nén rất cao, kích thước tập tin nhỏ hơn. Công nghệ phân luồng tập tin. Chất lượng âm thanh kém hơn các tập tin MP3 Phải cài đặt trình ứng dụng như RealPlayer.

3.2.2. Liên kết đến tập tin Media

Liên kết đến tập tin Media là cách đơn giản và hiệu quả để thêm âm thanh cho trang Web.

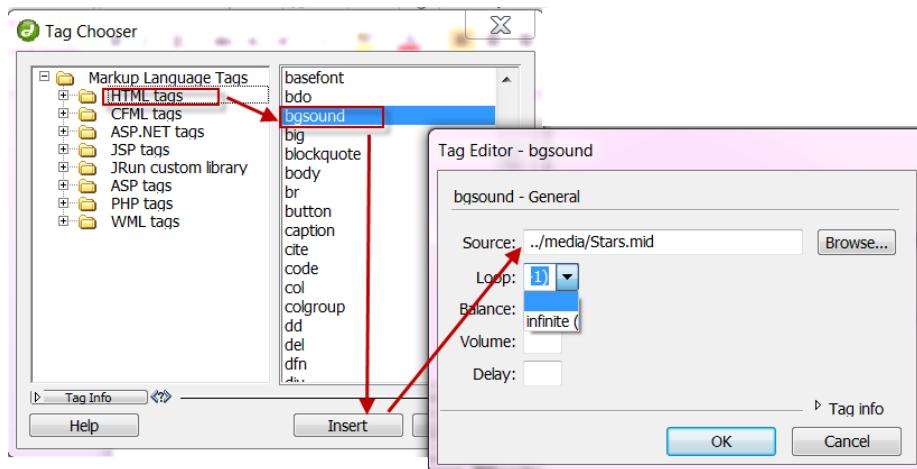
- Chọn văn bản hay hình ảnh muốn sử dụng để liên kết với tập tin Media.
- Trên thanh **properties**, ở trường **Link**, kích vào biểu tượng thư mục để tìm tập tin âm thanh muốn liên kết sau đó kích **Ok** (hoặc gõ vào đường dẫn và tên tập tin).

3.2.3. Chèn nhạc nền.

Tag **Chooser** là công cụ giúp thêm các thẻ từ các thư viện thẻ. Một số thẻ được thiết kế của sổ nhập các thuộc tính giúp tránh sai sót khi thao tác trực tiếp với thẻ trong chế độ **Code View**. Ta có thể thêm thẻ ngay cả khi con trỏ đang ở chế độ **Design View**. Muốn thêm thẻ tại đâu chỉ cần đặt con trỏ tại đó và chọn biểu tượng  trên nhãn **Common** của thanh **Insert**.

Với file nhạc (nên chọn nhạc MDI) nếu muốn đặt nhạc nền thì có thể sử dụng thẻ <BGSOUND>. Các thao tác thực hiện như sau:

- Từ thanh Insert chọn **Tag Chooser** cửa sổ Tag Chooser hiển thị (*hình 2.21*)
- Chọn thẻ **BgSound** trong phần **HTML Tags**
- Chọn nút **Insert**.
- Ở cửa sổ **Tag Editor – Bgsound** nhập các thuộc tính:



Hình 2.21: Chèn nhạc nền

+ **Source**: đường dẫn của file âm thanh (cần phải copy về thư mục của site).

+ **Loop**: số lần phát file nhạc. Nếu gán bằng -1 thì file được phát đi phát lại.

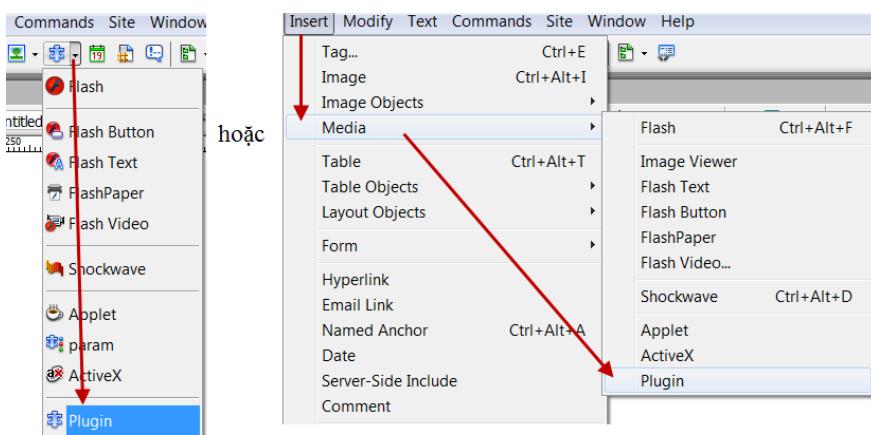
3.2.4. Nhúng tệp tin media vào trang web.

Nhúng Media kết hợp với trình chạy âm thanh trực tiếp vào trang. Tệp tin Media nhúng nên nằm cùng thư mục với trang đang thiết kế.

❖ Chèn nội dung Plug-in:

- Trong chế độ Design, đặt điểm chèn ở vị trí bạn muốn chèn nội dung.

- Kích vào nút **Plug-in**  trong phần mục media trên thanh công cụ **Common** hay chọn lệnh **Insert / Media / Plugin**.



Hình 2.22 : Nhúng tệp tin media.

- Hộp thoại **Select File** hiển thị, kích chọn tìm tập tin Media rồi nhấn **OK** để chèn.
- Sau khi chèn nội dung cho Plug-in, bạn sử dụng **Property inspector** để đặt các tham số cho nội dung đó.



Hình 2.23 : Thuộc tính của Plugin

- + W và H: chỉ định độ rộng và chiều cao xác định vị trí cho đối tượng.
- + **Src:** chỉ định đường dẫn đến tập tin dữ liệu nguồn.
- + **Plg Url:** chỉ định URL của thuộc tính pluginspace. Nhập URL hoàn chỉnh của site nơi những người sử dụng có thể tải các plug-in. Nếu người sử dụng xem trang mà không có Plug-in thì trình duyệt sẽ cố gắng tải nó từ URL này.
- + **Align:** xác định cách căn chỉnh cho đối tượng.
- + **V Space** và **H Space:** chỉ định số pixel cho khoảng trắng ở trên, dưới và cả 2 bên của Plug-in.
- + **Border:** chỉ định độ dày đường viền quanh Plug-in.
- + **Parameters:** mở hộp thoại để nhập các tham số bổ sung truyền

vào **Plug-in Netscape Navigator**. Nhiều Plug-in tương ứng với các tham số đặc biệt. Ví dụ: Plug-in Flash chứa các tham số về bgcolor, align, và scale.

❖ **Chạy các Plug-in trong cửa sổ tài liệu:**

Bạn có thể xem trước các movie và các ảnh động dựa vào các Navigator Plug-in, đó là các đối tượng sử dụng tag embed - trực tiếp trên cửa sổ tài liệu. Bạn cũng có thể chạy tất cả các đối tượng Plug-in cùng một lúc để xem trang sẽ hiển thị như thế nào, hoặc bạn có thể chạy mỗi đối tượng để bạn đảm bảo bạn đã nhúng các đối tượng media chính xác.

Để chạy các movie, các Plug-in thích hợp phải được cài đặt trong máy tính của bạn. Khi Dreamweaver khởi động, nó tự động tìm tất cả các Plug-in đã được cài đặt, đầu tiên là tìm trong thư mục **Configuration / Plug-in**, sau đó là tìm trong thư mục Plug-in của tất cả các trình duyệt đã cài đặt.

- Chọn một đối tượng media mà bạn đã chèn, chọn menu lệnh **View / Plugins / Play** hay kích nút **Play** trên hộp **Property inspector**. Bạn cũng có thể chọn lệnh **View / Plugins / Play All** để chạy tất cả các đối tượng media trên trang được chọn dựa vào các Plug-in.

❖ **Để ngừng chạy nội dung Plug-in:**

Chọn đối tượng media và chọn **View / Plugins / Stop** hoặc kích vào nút **Stop** trên **Property inspector**. Bạn cũng có thể chọn **View / Plugins / Stop All** để ngừng tất cả nội dung plug-in đang chạy.

❖ **Hiệu chỉnh các Plug-in Navigator**

Nếu một vài nội dung Plug-in không chạy, bạn thử làm như sau:

- Kiểm tra để chắc chắn Plug-in tương ứng đã được cài đặt vào máy tính của bạn, và nội dung là tương thích với phiên bản Plug-in bạn có.
- Mở tập tin ..\Macromedia\Dreamweaver\Configuration\Plugins\UnsupportedPlugins.txt trên trình chỉnh sửa văn bản và tìm xem có plug-in nào được liệt kê không. Tập tin này theo dõi các Plug-in có vấn đề trong Dreamweaver và vì vậy mà không được cung cấp.

- Kiểm tra máy bạn có đủ bộ nhớ hay không. Một vài Plug-in yêu cầu 2 đến 5 MB bộ nhớ bổ sung để chạy.

Có các loại tập tin âm thanh khác nhau, các khuôn dạng và các phương pháp thêm âm thanh cho trang Web khác nhau.

4. Liên kết.

Thực chất việc bổ sung các liên kết rất dễ dàng. Một liên kết là một đáp ứng báo cho trình duyệt rằng "khi người dùng nhấp vào đây, hãy mở trang Web này hoặc đi đến phần này của trang". Đây là thành phần cơ bản nhất và có ý nghĩa nhất của trang web giúp tạo ra sự liên kết giữa các trang tài liệu, các thành phần khác để tạo thành website. Nhiều thành phần trên trang web có khả năng đặt liên kết.

4.1. Tạo liên kết đến trang tài liệu khác

- Chọn dòng văn bản hoặc hình ảnh cần tạo liên kết.
- Kích chuột phải và chọn **Make Link** ở menu ngữ cảnh hoặc vào trình đơn **Modify / Make Link** hoặc khai báo tại trường **Link** trên cửa sổ thuộc tính **Properties**.
 - Hộp thoại **Select file** hiển thị, kích chọn file đích liên kết hay nhập một địa chỉ Website đích liên kết vào khung URL.

Riêng đối với cách sử dụng cửa sổ thuộc tính Properties có 3 cách để chỉ ra file liên kết:

- Gõ trực tiếp đường dẫn file liên kết vào hộp nhập.
- Nếu trên danh sách file của cửa sổ Site đã có file đó thì có thể dùng nút  giữ trái chuột và kéo trỏ chuột vào file đó.
- Chọn nút  để chỉ ra file.



Hình 2.24 : Thuộc tính liên kết.

Sau khi đặt liên kết, cửa sổ **Properties** sẽ cho phép chọn thuộc tính **Target**. Đây là thuộc tính chỉ ra đích sẽ mở nằm ở đâu và thường được áp dụng với Website thiết kế theo kiểu Frame .

- + **_self** (hoặc để trắng): hiện trang liên kết trên cùng trang hiện hành.
- + **_blank**: hiện trang liên kết trên một cửa sổ trình duyệt mới.
- + **_parent**: hiện trang liên kết trên vùng cha của trang đặt liên kết.
- + **_top**: hiện trang liên kết trên toàn vùng hiển thị trang web của cửa sổ trình duyệt.

4.2. Tạo liên kết đến một phần của trang tài liệu.

Một trang web với nội dung quá dài thường gây khó khăn cho người đọc hay nói cụ thể hơn là người đọc sẽ mất thời gian để tìm được thông tin mong muốn. Để giải quyết vấn đề này ta có thể tạo ra các đề mục liên kết ở đâu đó trên trang giúp người đọc nhanh chóng chuyển đến vị trí chứa thông tin đó trên cùng trang. Kiểu liên kết như vậy được gọi là liên kết trên cùng một trang (hay là anchor - mỏ neo).

Để thêm liên kết mỏ neo còn phải thực hiện 2 thao tác sau:

- Tạo các điểm dừng (điểm đặt mỏ neo).
- Đặt đường dẫn liên kết đến điểm dừng.

Các điểm dừng được tạo như sau:

- Đặt con trỏ tại vị trí cần đặc điểm dừng.
- Chọn biểu tượng  trên nhãn **Common** của thanh công cụ **Insert**.
- Đặt tên cho điểm dừng.
- Chọn **OK**

Một hình  được đặt trên trang cho biết tại vị trí đó có điểm dừng.

Sau khi tạo điểm dừng ta tạo ra các liên kết. Các liên kết trên cùng một trang được tạo ra giống với liên kết bình thường chỉ khác là thay đường dẫn trang liên kết bằng tên điểm dừng. Và trước tên điểm dừng có thêm dấu #. Ví dụ: #cuoitrang.

4.3. Liên kết Email

Tạo một liên kết thư điện tử hơi khác so với một liên kết thông thường. Với liên kết thư điện tử, khi người dùng nhập vào liên kết này, trình quản lý thư điện tử mặc định của người sử dụng sẽ được mở ra. việc còn lại chỉ là đặt tiêu đề và soạn nội dung cho thư. Với cách thực hiện này người sử dụng sẽ không phải nhớ địa chỉ email cũng như tránh được trường hợp gõ sai.

- Bạn đặt vị trí con trỏ chuột tại nơi muốn đặt dòng liên kết thư điện tử.
- Kích chọn vào biểu tượng  hay vào trình đơn **Insert / Email Link**
- Hộp thoại **Email Link** hiển thị



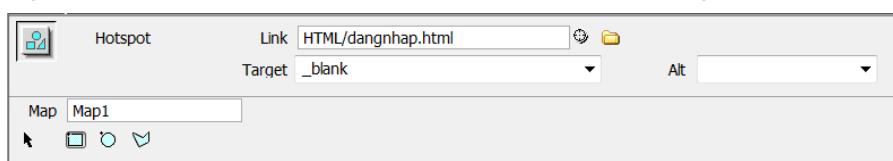
Hình 2.25 : Liên kết Email.

- Bạn nhập địa chỉ thư điện tử vào cả hai trường **Text** và **E-mail**. Sau đó bạn kích **OK**, Dreamweaver 8 sẽ tạo cho bạn một địa chỉ mail liên kết với đúng tên địa chỉ bạn nhập vào.

4.4. Bản đồ ảnh (hotspot).

Một hình ảnh với nhiều liên kết với nó sẽ tạo ra một bản đồ ảnh. Các liên kết được gắn với các vùng khác nhau của cùng một hình ảnh bằng cách sử dụng các điểm nóng (hotspot). Việc nhấp vào các điểm nóng sẽ đưa người sử dụng đến các liên kết tương ứng.

- Chọn một ảnh cần tạo Hotspot ở trang đang thiết kế.
- Trên thanh công cụ **Insert**, bạn chọn lớp **Common**. Lúc này thanh **Properties** trên trang thiết kế sẽ hiển thị công cụ Hotspot.
- + **Rectangular Hotspot**: tạo vùng nóng hình chữ nhật.
- + **Oval Hotspot**: tạo vùng nóng hình tròn hoặc Elip.
- + **Polygon Hotspot**: tạo vùng nóng theo cách vẽ tự do.
- Bạn chọn một trong các công cụ vẽ này sao cho tương ứng với hình của bạn và dùng chuột rê, vẽ lên bức ảnh. Bạn có thể vẽ tạo ra nhiều vùng điểm nóng trên bức ảnh. Sau đó bạn khai báo các thông tin sau:



Hình 2.26: Liên kết bản đồ ảnh.

- + **Map**: Tên bản đồ ảnh.
- + **Link**: Địa chỉ của trang cần liên kết tới.
- + **Target**: Hình thức hiển thị
- + **Alt**: Câu chú thích khi rê vào liên kết.

Chú ý:

Trong trường Map, bạn nhập tên cho bản đồ ảnh. Điều đặc biệt quan trọng là bạn phải đặt tên cho các bản đồ ảnh với các tên duy nhất khi mà bạn có nhiều bản đồ ảnh trên một trang web. Tên của bản đồ ảnh là **bắc**

buộc phải có, nếu bạn không nhập vào tên này, bạn sẽ không thể sử dụng được các công cụ hotspot khi trang có hai bản đồ ảnh cùng tên mặc định.

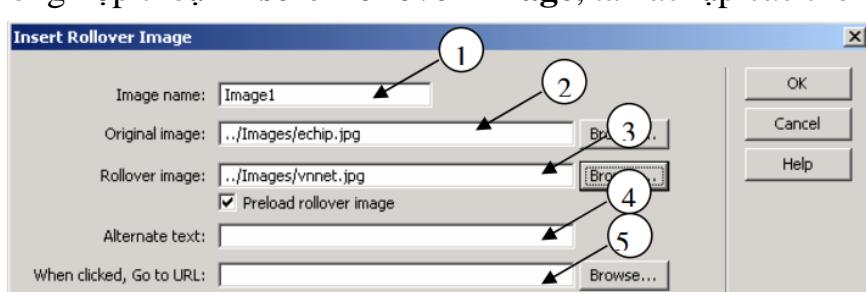
4.5. Tạo Rollover.

Rollover (ảnh cuộn) là một behavior làm cho hình ảnh bị thay đổi bởi một hình ảnh khác khi con trỏ chuột di chuyển lên phía trên nó. Thông thường người sử dụng sẽ nhận biết được rằng một hình ảnh thay đổi chính là một liên kết đến một trang web khác.

Để tạo cho một ảnh có hiệu ứng Rollover ta thực hiện như sau :

- Đặt vị trí con trỏ tại nơi cần tạo Rollover. Sau đó kích chọn biểu tượng  hay vào trình đơn Insert/ Interactive Images/ Rollover.

- Trong hộp thoại **Insert Rollover Image**, ta xác lập các thông số sau:



Hình 2.27 : Chèn liên kết Rollover.

+ **Image Name:** nhập tên cho nút bạn cần tạo sao cho tương thích và dễ nhớ.

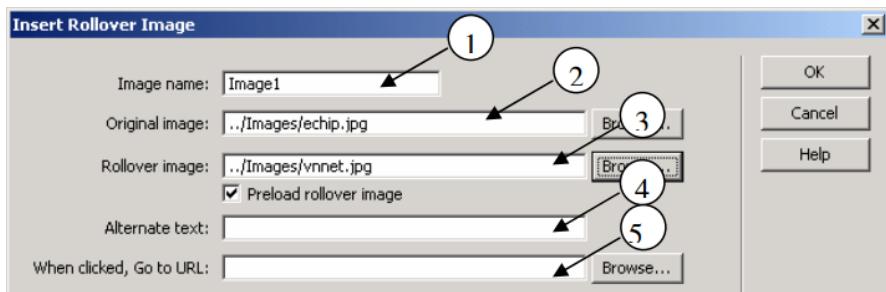
+ **Original Image:** Chọn ảnh gốc (ảnh đầu tiên hiển thị khi người sử dụng mở trang web).

+ **Rollover Image:** Chọn ảnh bạn muốn hiển thị khi người sử dụng di chuyển chuột đến ảnh Original Image.

+ **When Clicked, Go To URL:** chọn trang cần liên kết khi người sử dụng kích vào ảnh.

+ **Alternate text :** Câu chú thích khi rê chuột vào ảnh trang.

Tùy chọn **Preload Rollover Image** sẽ được chọn mặc định, bạn nên giữ nguyên mặc định này. Hình ảnh Rollover sẽ được tải xuống cùng với các hình ảnh còn lại trên trang web và sẽ có hiệu lực ngay khi người dùng di chuyển chuột lên phía trên biểu tượng. Nếu bạn bỏ tùy chọn này, hình ảnh sẽ được tải xuống trước khi nó xuất hiện khi người dùng di chuyển chuột lên trên biểu tượng.



Hình 2.28: *Rollover Image*

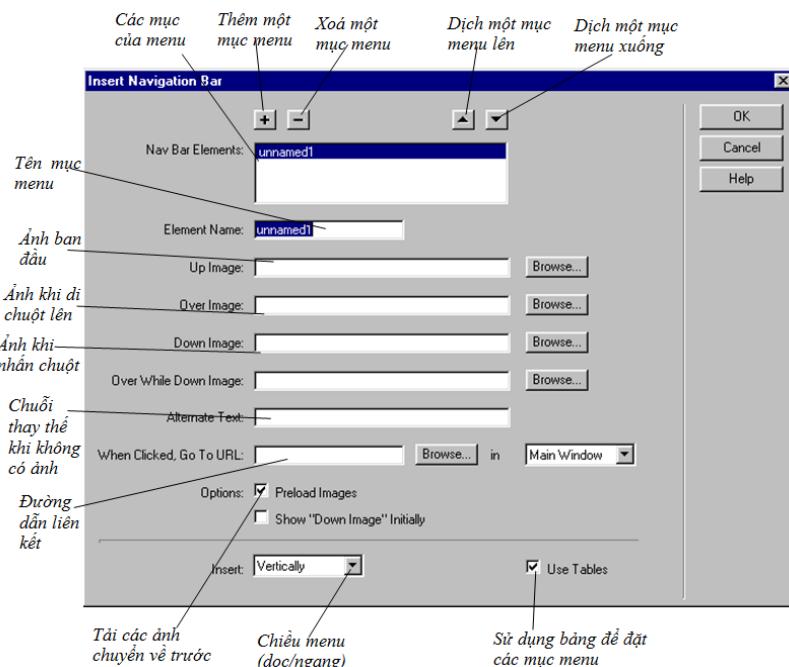
4.6. Tạo liên kết Navigation Bar.

Navigation Bar bao gồm một ảnh (hoặc một tập ảnh) hiển thị những thay đổi theo thao tác của người sử dụng. Nó giúp di chuyển dễ dàng giữa các Site và các tập tin trong Site. Nó thường dùng để tạo ra menu (dọc hoặc ngang) cho trang web. Về bản chất các mục của menu là độc lập và gần giống với Rollover. Trước khi tiến hành tạo Navigation Bar cần phải có đủ các ảnh tạo nên menu (gồm cả các ảnh chuyển) để hiển thị các trạng thái của mỗi phần tử trong thanh Navigation. Một phần tử trên thanh điều hướng gồm có 4 trạng thái:

- **Up:** ảnh xuất hiện khi người sử dụng chưa kích hoặc đã tác động vào nó.
- **Over:** ảnh xuất hiện khi trỏ chuột di chuyển lên nó.
- **Down:** ảnh xuất hiện khi kích chuột.
- **Over While Down:** ảnh xuất hiện khi trỏ chuột di chuyển ra ngoài sau khi kích chọn.

4.6.1. Chèn thanh điều hướng:

- Chọn menu **Insert / Image object / Navigation Bar**, hoặc kích vào biểu tượng . Hộp thoại **Insert Navigation Bar** hiển thị (hình 2.29)
- Nhập tên cho phần tử của thanh điều hướng trong trường **Element Name**. Mỗi phần tử tương ứng với một nút qua cách đặt 4 ảnh trạng thái. Tên phần tử xuất hiện trong danh sách **Nav Bar Elements**. Sử dụng các nút mũi tên để sắp xếp các thành phần trong thanh điều hướng.
- Chọn ảnh cho các trạng thái (bạn phải chọn ảnh cho trạng thái Up; các trạng thái khác thì tùy ý).



Hình 2.29 : Navigation.

- Trong **When Clicked, Go To URL** xác định vị trí mà ở đó tập tin được liên kết sẽ được mở.

- Xác định các tùy chọn nạp ảnh:

- + **Preload Image:** tải các ảnh khi trang được truy cập.
- + **Show Down Image Initially:** để hiển thị phần tử được chọn trong trạng thái Down thay vì trạng thái Up mặc định của nó.

- Chọn hướng trình bày cho thanh Navigation trong mục Insert.

- + Để hiển thị theo chiều dọc hay chiều ngang chọn trong **menu Pop-up**.

- + Để chèn các phần tử trên thanh điều hướng như bảng, kích chọn hộp kiểm **Use Tables**.

- Kích nút dấu (+) để bổ sung một thành phần khác vào thanh điều hướng, sau đó lặp lại từ bước 2 đến bước 5.

- Khi bạn định nghĩa xong tất cả các phần tử trong thanh điều hướng, kích chọn OK.

4.6.2. Hiệu chỉnh thanh điều hướng:

Khi đã tạo một thanh điều hướng, bạn có thể bổ sung hoặc xoá các ảnh ở các trạng thái.

- Chọn **Modify / Navigation Bar**.
- Trên danh sách **Nav Bar Elements**, chọn phần tử mà bạn muốn chỉnh sửa.

- Thao tác hiệu chỉnh xong, kích chọn OK.

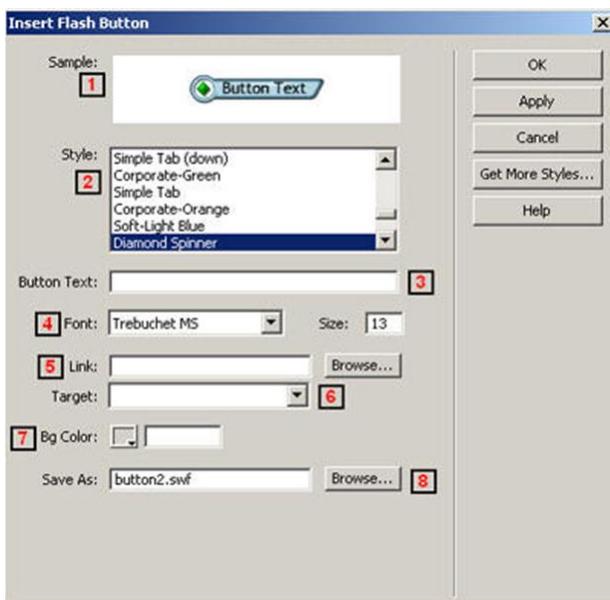
4.7. Bổ sung nút Flash.

Đổi tượng nút Flash cho phép thay đổi tùy ý và chèn một tập các nút Flash đã được định nghĩa trước.

4.7.1. Tạo nút Flash.

- Bạn đặt điểm chèn trong cửa sổ tài liệu, kích nút  trên trên đối tượng **panel Objects** hay từ trình đơn **Insert / Interactive Image / Flash Button**.

- Hộp thoại **Insert Flash Button** hiển thị:



Hình 2.30 : Chèn Flash Button.

[1]: **Sample:** hiển thị mẫu nút cho bạn xem trước khi bạn chọn kiểu nút từ danh sách Style.

[2]: **Style:** hộp hiển thị danh sách các kiểu nút Dreamweaver hỗ trợ cho bạn.

[3]: **Button Text:** nơi nhập nội dung cho button.

[4]: **Font, Size:** chỉ định kiểu chữ và cỡ chữ cho button.

[5]: **Link:** nhập liên kết tương đối hoặc tuyệt đối cho nút. Nếu bạn sử dụng mối liên kết tương đối, thì phải đảm bảo là tập tin SWF được lưu trên cùng thư mục với tập tin HTML.

[6]: **Target:** xác định tên frame đích hoặc cửa sổ đích cho mỗi liên kết của nút Flash sẽ mở.

[7]: **Bg Color:** đặt màu nền cho movie Flash.

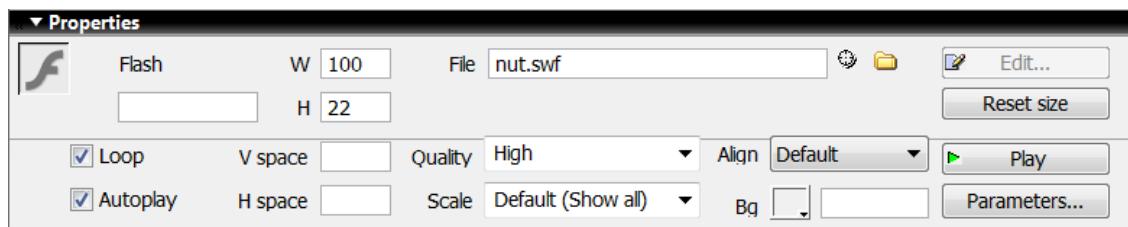
[8]: **Save As:** nhập tên tập tin để lưu tập tin SWF mới. Nếu tập tin

chứa mối liên kết tương đối, bạn phải lưu tập tin vào cùng thư mục chứa tài liệu HTML hiện thời.

4.7.2. Chính sửa các đối tượng nút Flash:

- Chọn đối tượng nút Flash.
- Trên **Properties inspector**, hiển thị các thuộc tính của nút Flash.

Bạn có thể tạo một vài thay đổi trên **Properties inspector**. Các thay đổi này chỉ chỉnh sửa các thuộc tính HTML như độ rộng, chiều cao và màu nền.



Hình 2.31 : *Chỉnh sửa nút Flash.*

- Để thay đổi nội dung, mở hộp thoại Insert Flash Button bằng một trong các phương pháp sau:
 - + Kích đúp chuột vào đối tượng nút **Flash**
 - + Kích chuột vào nút **Edit** trên **Properties inspector**.
 - + Kích phải chuột lên đối tượng và chọn **Edit** từ trình đơn Pop-up.
- Trên hộp thoại **Insert Flash Button**, bạn tạo những chỉnh sửa cho đối tượng.

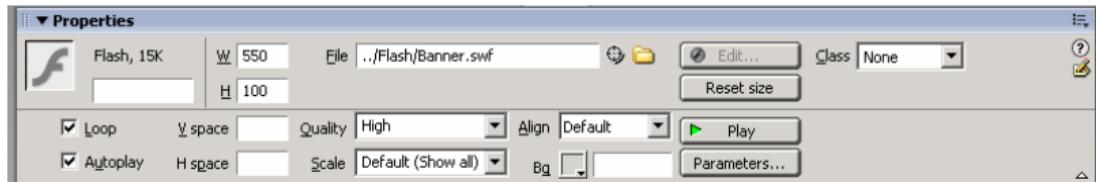
5. Các yếu tố động trên trang.

5.1. Chèn tệp tin flash.

Flash là các đối tượng tạo hiệu ứng chuyển động trên trang web do chương trình Macromedia Flash tạo ra. Nó thường được thể hiện ở dạng banner quảng cáo và hoạt hình. Các file flash có đuôi là swf.

- Đặt điểm chèn ở vị trí bạn muốn chèn movie.
- Kích nút **Insert Flash** trên thanh **panel Objects** hay chọn trình đơn **Insert / Media / Flash**.
- Hộp thoại **Select File** hiển thị: Trong hộp thoại **Select File**, bạn chọn tập tin Flash (.swf). Kích **OK** để chấp nhận.
- Trên cửa sổ **Properties**, xác định độ rộng, chiều cao và canh lề cho

file Flash hoặc có thể xem thử nó chạy bằng cách chọn nút  trên cửa sổ **Properties**.



Hình 2.33 : Chèn tệp tin Flash.

5.2. Chèn Java applet

Java là một ngôn ngữ lập trình cho phép phát triển các ứng dụng thuộc hạng nhẹ có thể được nhúng vào các trang Web. Sau khi bạn tạo một Java applet, bạn có thể chèn applet vào tài liệu HTML sử dụng Dreamweaver sử dụng tag applet để đánh dấu điểm tham khảo tập tin applet.

- Đặt con trỏ chuột tại vị trí muốn chèn applet.
- Kích nút **Applet**  trên **panel Object** hay từ trình đơn **Insert / Media / Applet**.

- Hộp thoại **Select File** hiển thị, bạn chọn tập tin chứa Java applet.

5.3. Chèn Java Script

5.3.1. Giới thiệu về Javascript:

- Với HTML sẽ cho ta biết cách tạo ra trang Web - tuy nhiên chỉ mới ở mức biểu diễn thông tin chứ chưa phải là các trang Web động có khả năng đáp ứng các sự kiện từ phía người dùng.

- JavaScript là ngôn ngữ dưới dạng kịch bản (script) có thể nhúng trực tiếp vào các file HTML. Nó không được biên dịch mà được trình duyệt diễn dịch, trình duyệt đọc JavaScript dưới dạng mã nguồn. Chính vì vậy ta có thể dễ dàng học JavaScript trên các trang Web có sử dụng JavaScript.

5.3.2. Nhúng javascript vào trang web:

Để nhúng javascript ta có 4 cách:

- Nhúng trực tiếp bằng các câu lệnh và các hàm trong cặp thẻ <SCRIPT>...</SCRIPT>.
- Sử dụng các file JavaScript
- Sử dụng một biến thức JavaScript làm giá trị của một thuộc tính HTML
- Sử dụng thẻ sự kiện (event handlers) trong một thẻ HTML nào đó.

Trong đó, sử dụng cặp thẻ <SCRIPT>...</SCRIPT> và nhúng một file nguồn JavaScript là được sử dụng nhiều hơn cả.

❖ **Nhúng trực tiếp.**

Javascript thường đường nhúng vào tài liệu HTML bằng cách chèn các mã lệnh javascript bên trong cặp thẻ <SCRIPT>...</SCRIPT> và trong một tài liệu HTML có thể có nhiều cặp thẻ như vậy.

Câu lệnh javascript có thể đặt bất cứ đâu trong tài liệu HTML, tuy nhiên nếu đặt trong phần <Head>, nó sẽ được tải và sẵn sàng trước khi phần còn lại của văn bản được tải. Sử dụng cú pháp sau:

```
<Script>
// Chèn các mã Javascript vào đây
</Script>
```

Một số trình duyệt phiên bản cũ không hiểu được mã javascript chính vì vậy người ta thường đặt mã lệnh javascript giữa ký hiệu <!-- và //--> để đánh lừa chúng. Các trình duyệt cũ coi đó là chú thích và sẽ bỏ qua.

Ví dụ: Hiển thị trên trang Web lịch “Hôm nay là ngày tháng năm”

Thẻ Javascript được đặt ở vị trí thích hợp trong phần thân của trang html như sau.

```
<head>
<title> Ví dụ chèn JavaScript </title>
</head>
<body>
<script language="JavaScript">
    var Thu=new Array(" Chủ nhật, "," Thứ hai, "," Thứ ba, "," Thứ tư, ","
    Thứ năm, "," Thứ sáu, "," Thứ bảy, ");
    var
thang=newArray("01","02","03","04","05","06","07","08","09","10","11",
"12");
    var today=new Date();
    var Year=today.getYear();
    document.write('Hôm nay là' + Thu[today.getDay()] + ' Ngày ' +
today.getDate() + ' Tháng ' + Thang[today.getMonth()] + ' Năm '+
Year)
</script>
</body>
```

</html>

Kết quả duyệt Web là :



Hình 2.34 : JavaScript ngày tháng đầy đủ

Có thể sưu tầm các mã JavaScript từ Website <http://www.javascriptbank.com.vn>, www.echip.com. hoặc trong thư mục C:/DATA/TKWEB.

❖ *Sử dụng file JavaScript.*

Cách thứ hai là ta liên kết file văn bản chứa mã JavaScript với tài liệu HTML. Thuộc tính SRC của thẻ <SCRIPT> được dùng để chỉ ra tên file Javascript như sau:

```
<script language= "Javascript " src= "filename.js">  
</script>
```

File liên kết phải có đuôi .js và chỉ chứa các lệnh và định nghĩa hàm javascript .

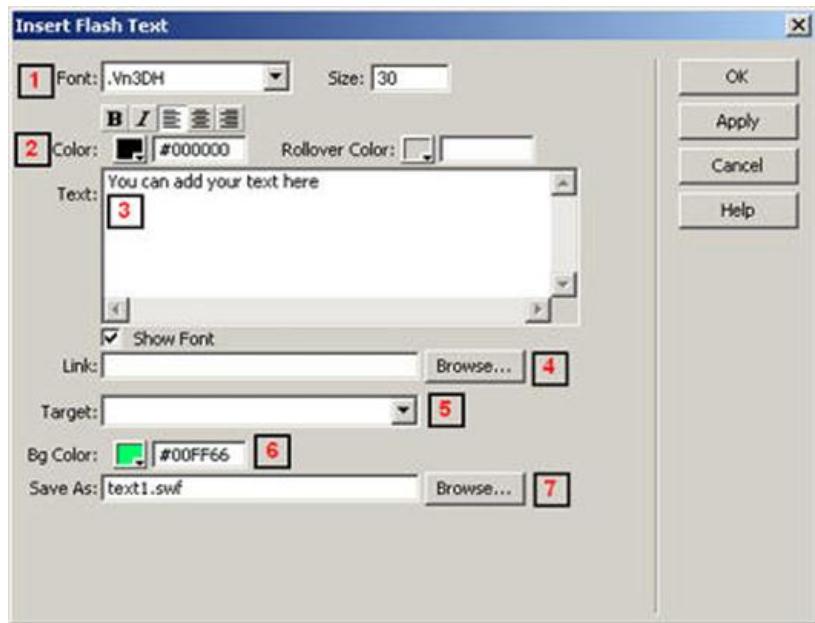
Ví dụ :

```
<html>  
<head>  
  <script src= "../media/test.js">  
  </script>  
</head>  
<body>  
Sử dụng file js liên kết  
</body>  
</html>
```

5.4. Chuỗi ký tự có hiệu ứng Flash.

Đối tượng này cho phép bạn tạo và chèn movie Flash chỉ có text.

- Trong cửa sổ tài liệu, chọn nút  **Insert Flash Text** từ panel Objects, hoặc chọn trình đơn **Insert / media / Flash Text**.
- Cửa sổ **Insert Flash Text** hiển thị:



Hình 2.32 : Chuỗi ký tự liên kết có hiệu ứng flash.

[1] **Font, Size:** chọn kiểu chữ và cỡ chữ cho đoạn Flash Text.

[2] **Color, Rollover Color:** đặt màu cho chữ và đặt màu cho chữ khi di chuyển chuột lên đoạn text.

[3] **Text:** nhập nội dung bạn muốn vào trường text, để thấy kiểu font xuất hiện trên trường text, bạn chọn **Show Font**.

[4] **Link:** nếu bạn muốn kết hợp một mối liên kết với đối tượng văn bản Flash, nhập vào mối liên kết tuyệt đối hoặc tương đối.

[5] **Target:** nếu bạn đã nhập mối liên kết, thì có thể chỉ định frame hoặc cửa sổ đích cho mối liên kết đó.

[6] **Bg Color:** đặt màu nền cho văn bản.

[7] **Save As:** nhập tên tập tin để lưu tập tin SWF mới. Nếu tập tin chưa có mối liên kết tương đối, bạn phải lưu tập tin vào cùng thư mục chứa tài liệu HTML hiện thời.

6. Bảng (Table).

Bảng là thành phần rất quan trọng trong thiết kế giao diện trang web, là một công cụ rất tiện lợi để sắp đặt các ảnh và các dữ liệu trên một trang HTML.

Bảng bao gồm 3 thành phần cơ bản: hàng (Row), cột (Column) và ô (Cell: ô). Nhờ sử dụng bảng ta có thể bố trí các thành phần trên trang tại một vị trí chính xác bằng cách thay đổi vị trí các ô đặt thành phần.

Ta có thể làm việc với bảng ở một trong hai chế độ:

- Standard (mặc định).

- Layout.

Các bảng có thể đặt lồng nhau để tạo những giao diện phức tạp. Tuy nhiên khi thiết kế cố gắng càng sử dụng ít bảng lồng nhau càng tốt. Điều này giúp ta dễ làm việc ở chế độ Layout

6.1. Sử dụng bảng ở chế độ Standard:

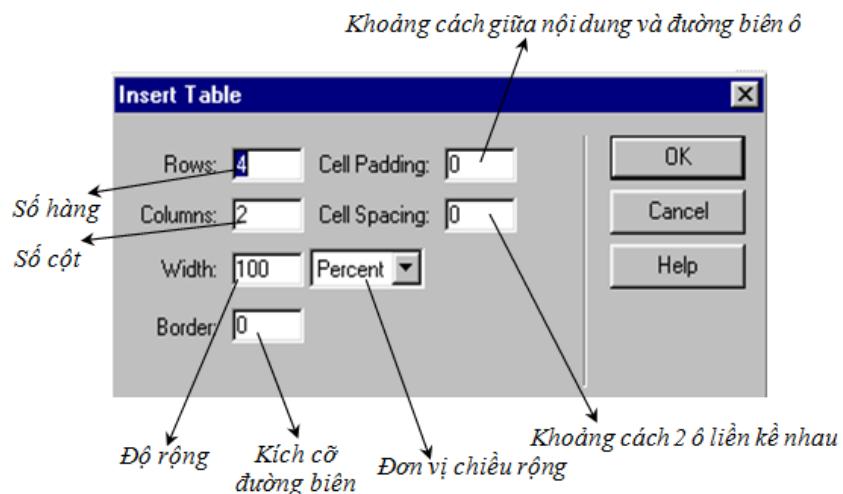
6.1.1. Chèn bảng

- Chuyển sang chế độ **Standard** cách chọn ô **Standard** trên nhãn **Layout** của thanh công cụ Insert .



Hình 2.33 : Chuyển chế độ Standard.

- Chọn biểu tượng **Insert Table** trên thanh công cụ **Common** hoặc **Layout** (hoặc chọn menu **Insert / Table**) để tạo một bảng mới.
- Hộp thoại **Insert Table** hiển thị, nhập các giá trị thích hợp vào trong hộp thoại rồi nhấn **OK**.



Hình 2.34 : Chèn bảng.

Để bảng không hiển thị khi mở trên trình duyệt thì ta cần đặt thuộc tính Border = 0. Đặt CellPadding và Ô CellSpacing đều = 0 để không có khoảng cách nào giữa các ô. Kích cỡ của bảng có thể đặt theo điểm ảnh (pixel) hoặc theo phần trăm (%). Nếu đặt theo điểm ảnh thì kích cỡ của bảng luôn là cố định dù ở bất kỳ độ phân giải nào cũng như ở bất kỳ kích thước nào của cửa sổ trình duyệt. Còn nếu kích cỡ bảng được đặt theo % thì bảng sẽ tự co giãn theo độ phân giải cũng như kích cỡ của trình duyệt.

6.1.2. Nhập dữ liệu vào bảng

- ❖ Thêm nội dung vào các ô trong bảng

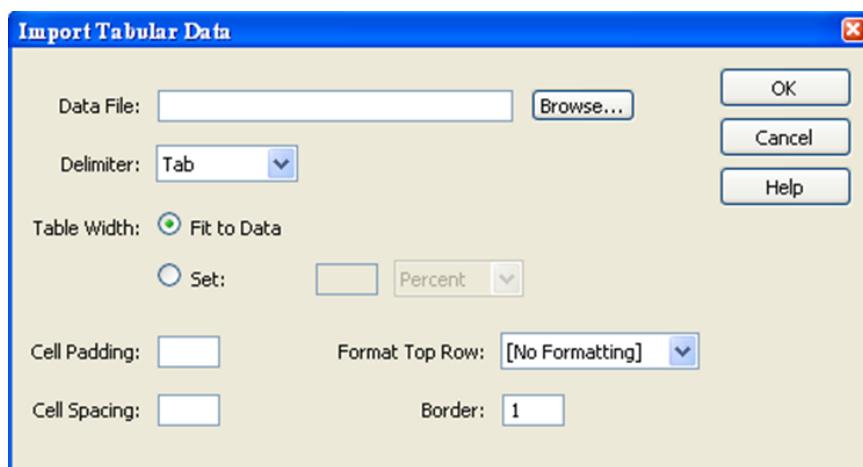
Có thể chèn văn bản, hình ảnh hoặc thậm chí một bảng khác vào các ô trong bảng.

- **Chèn văn bản vào ô:** Đặt điểm chèn vào trong ô để gõ văn bản hay có thể dán văn bản từ một tài liệu khác.
- **Chèn hình ảnh vào ô:** Kích vào trong ô muốn chèn hình ảnh, sau đó chèn hình ảnh như hướng dẫn phần 6.1.1.
- **Chèn bảng vào ô:** Kích vào trong ô muốn chèn bảng và tiến hành chèn bảng như hướng dẫn phần 3.1.1.

❖ Nhập dữ liệu cho bảng từ một file ngoài:

Dữ liệu được tạo từ một trình ứng dụng khác (ví dụ: Excel) và được lưu dưới định dạng có ranh giới (như được phân cách bởi Tab, dấu phẩy, dấu hai chấm, dấu chấm phẩy . . .) có thể import vào trong Dreamweaver như sau :

- Chọn menu **Insert / Table Object / Import Tabular Data**
- Hộp thoại **Import Tabular Data** hiển thị, khai báo các thông số:



Hình 2.35 : Nhập dữ liệu cho bảng từ file ngoài

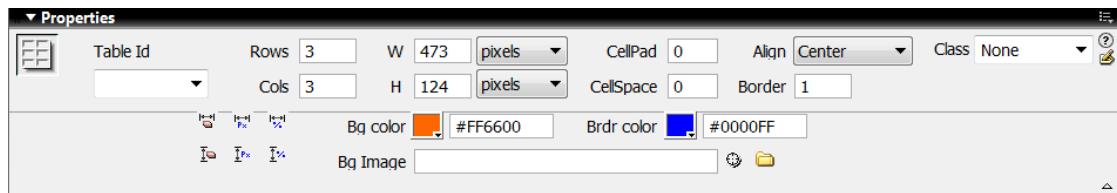
- + **Data File:** chọn tập tin dữ liệu cần chèn thành bảng vào tài liệu, kích nút **Browse** để duyệt tập tin.
- + **Delimter:** chọn kiểu định dạng của dữ liệu phân cách được nhập vào.
- + **Table width:** Kích chọn để độ rộng của cột vừa với độ dài của chuỗi văn bản dài nhất.
- + **Set:** cho phép thiết lập độ rộng của cột tính theo phần trăm hoặc pixel.
- + **Cell Padding:** chỉ định số pixel giữa nội dung trong ô và đường biên của ô.

- + **Cell Spacing:** chỉ định số pixel giữa các ô trong table.
- + **Format Top Row:** chọn định dạng cho hàng trên cùng.
- + **Border:** chỉ định số pixel cho độ rộng đường viền của bảng.

6.1.3. Định dạng bảng

Có thể thay đổi kích thước, khoảng cách giữa các cell . . . của table như sau:

- Thay đổi kích cỡ hàng, cột, ô : có thể thay đổi bằng cách giữ trái chuột tại các đường biên và kéo hoặc nhập kích thước thay đổi vào Width(W) và Height(H) trong cửa sổ **Properties Inspector**.



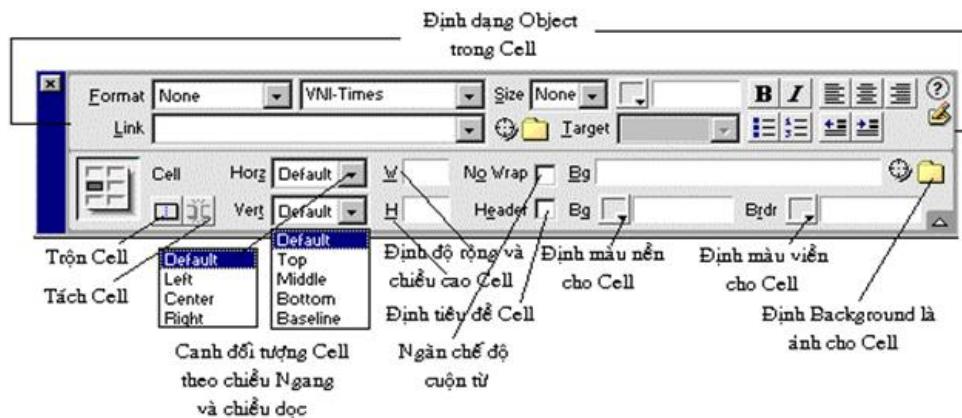
Hình 2.36 : Thuộc tính của bảng

- Căn chỉnh bảng: Chọn bảng được căn trái, phải hoặc giữa ở ô Align.
- Đường kẻ bảng : Điều chỉnh kích cỡ ở ô **Border**, màu khung ở ô **Brd color**.

6.1.4. Định dạng ô

Có thể canh chỉnh nội dung, thiết lập chiều cao, rộng, màu nền . . . cho ô(cell) bằng sử dụng **Properties Inspector**.

- Kích vào biên của ô hoặc nhấn giữ phím Ctrl và kích vào bên trong ô để chọn ô.
- Các thuộc tính của ô sẽ xuất hiện trên **Properties Inspector**, bạn có thể chọn và thay đổi các thuộc tính:



Hình 2.37 : Định dạng ô.

- + Thay đổi chiều rộng của ô: nhập số pixel vào ô **W**(Width) hay chọn **Autostretch**.
 - + Thay đổi chiều cao của ô: nhập số pixel vào ô **H**(Height.).
 - + Chọn màu nền/hình nền cho ô: Kích vào ô **Bg** và chọn một màu trong bảng chọn màu. Chọn hình nền cho ô ta kích biểu tượng để tìm file hình.
 - + Chọn màu cho đường kẻ ô: Kích ô **Brdr** và chọn một màu trong bảng chọn màu.
 - + Canh chỉnh nội dung của ô: Kích chọn trong menu **Horz** và **Vert**.
 - + Để nối (trộn) và tách ô : Chọn ô hoặc nhóm ô sau đó kích vào biểu tượng **Merges** hoặc **Splits**.
 - + Nếu bật tùy chọn **No Wrap**, ô sẽ tự động nới rộng để chứa nội dung mà không tự động ngắt dòng.
 - + Di chuyển các ô: đặt trỏ chuột lên biên của các ô, kích và rê để định lại vị trí của ô.
 - + Đánh dấu các ô: Giữ trái chuột tại ô đầu tiên (hoặc cuối cùng) của vùng cần đánh dấu và kéo.

6.1.5. Chèn thêm ô, hàng và cột vào bảng

- Mở lớp **Layout** của thanh công cụ **Insert**.



Hình 2.38 : Chèn hàng, cột.

- Chọn vị trí chèn bằng cách chọn ô , hàng hoặc cột .
- Nhấn chuột vào kiểu ô, hàng hoặc cột muốn chèn thêm trên thanh **Layout**.

6.2. Sử dụng bảng ở chế độ Layout View:

Chế độ **Layout View** giúp cho việc thao tác với bảng giống như thao tác vẽ. Ở chế độ **Standard View** thay đổi vị trí các ô thường kéo theo sự thay đổi vị trí hàng và cột. Còn trong chế độ **Layout View** việc thao tác với ô trỏ nên dễ dàng và độc lập hơn. Bên trong **Table** ta có thể thêm các ô và di chuyển chúng đến bất kì vị trí nào trong **Table** đó. Giống như bảng bình thường ta cũng có thể đặt các bảng lồng với nhau.

Để vẽ bảng hay ô cần chọn mục tương ứng trên nhãn **Layout** sau đó giữ trái chuột trên vùng thiết kế và kéo rê một vùng chữ nhật như mong muốn rồi thả chuột ra.

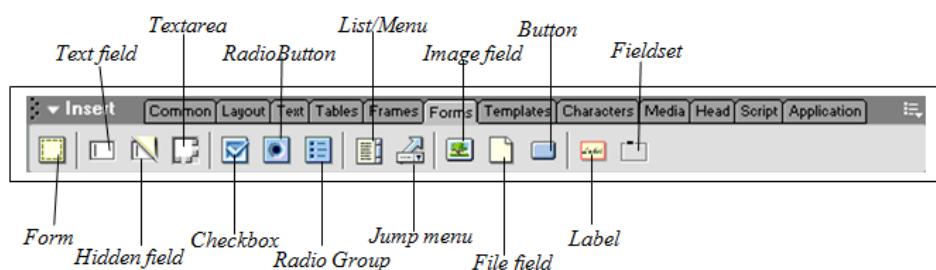
Chú ý ô chỉ được phép vẽ bên trong bảng do vậy phải vẽ bảng trước khi vẽ ô. Sau khi tạo xong các ô ta có thể đưa vào đó các thành phần như văn bản, hình ảnh... như với ô của bảng ở chế độ Standard View. Bảng đặt trong một bảng khác thì đường biên của nó sẽ không vượt quá được đường biên của bảng chứa nó. Do vậy nếu muốn kéo dãn đường biên của bảng trong thì phải kéo dãn đường biên của bảng ngoài trước.

Trong quá trình thiết kế ta có thể thay đổi vị trí, kích thước của ô bằng chuột hoặc bằng **Properties Inspector** với các thuộc tính giống như đối với chế độ **Standard View**.

7. Form

Form(khung nhập hoặc biểu mẫu) là một kỹ thuật cho phép chủ trang Web giao tiếp với người truy cập. Trên trang Web người dùng có thể được yêu cầu trả lời một câu hỏi, cho 1 ý kiến, chọn một mục trong danh sách định trước . . . chủ trang Web sẽ tiếp nhận và xử lý thông tin ấy. Bạn có thể dùng Dreamweaver để thiết kế các Form sau đó áp dụng các Behavior hay các đoạn mã JavaScript vào để xử lý dữ liệu trong form.

Các đối tượng form được đặt trên Tab Forms của thanh công cụ Insert như sau:



Hình 2.39: Form và các đối tượng trên form

Các đối tượng đều có thuộc tính tên là thuộc tính cần thiết để chương trình xử lý trên máy chủ có thể phân biệt thông tin của các thành phần khác nhau.

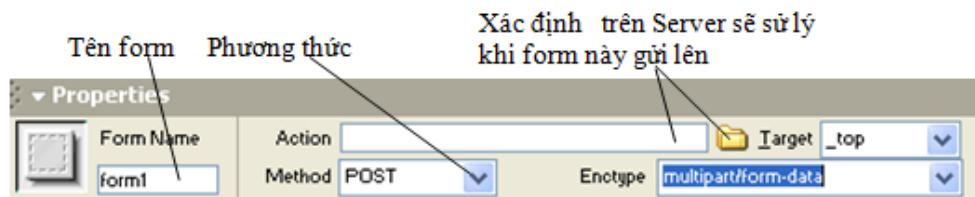
7.1. Chèn Form.

Là đối tượng dùng để chứa các đối tượng khác. Thông tin sẽ được gửi lên máy chủ theo từng form, mỗi thời điểm chỉ có 1 form được gửi. Các thao tác chèn một form mới như sau:

- Đặt trỏ tại nơi cần chèn Form.
- Chọn công cụ **Forms** trên thanh công cụ hoặc từ menu

Insert/form/form.

- Đường khung viền màu đỏ đại diện giới hạn của form, mọi đối tượng khác trên form phải nằm trong khung viền này.
- Thuộc tính của form bao gồm:



Hình 2.40 : Thuộc tính của form

- + **Form Name**: Tên của Form
- + **Action**: Đường dẫn chứa trang xử lý form hoặc có thể gửi thông qua Mail (MailTo: WebmasterPhetitc@gmail.com).
- + **Method**: Phương thức truyền dạng Post hay Get.

7.2. Thêm các đối tượng lên Form.

Để thêm đối tượng lên form chỉ cần đặt con trỏ lên form và thực hiện như một vùng bình thường trên trang web. Để có thể căn chỉnh các thành phần đặt trong form thẳng hàng hay theo vị trí mong muốn ta nên sử dụng bảng (thực hiện chèn bảng vào form cũng giống như thực hiện với trang Web).

7.2.1. Text field.

Text Field (ô nhập liệu một dòng) dùng để nhập các dòng thông tin ngắn gọn nằm trong cùng một dòng.

- Định vị con trỏ trong đường viền của form.
- Chọn menu **Insert / Form/ Text Field** hoặc kích biểu tượng
- Chọn **Text Field** vừa tạo, chỉ định các thuộc tính như sau:

- + **Text Field:** Định tên cho ô nhập liệu.
- + **Char Width:** Số ký tự cho chiều dài ô.
- + **Max Chars:** Số ký tự nhiều nhất có thể nhập.
- + **SingleLine:** Ô nhập liệu 1 dòng.
- + **MultiLine:** Ô nhập liệu nhiều dòng.
 - + **Password:** Ô nhập loại mật mã (Ký tự nhập được hiển thị thành *).

The image shows a login interface with a blue header bar. The title 'ĐĂNG NHẬP DIỄN ĐÀN' is centered at the top. Below it, there are two text input fields. The first field is labeled 'Tên đăng nhập:' and contains the text 'Trần Thị Ngân'. The second field is labeled 'Mật khẩu:' and contains five asterisks ('*****'). At the bottom of the form are two buttons: a grey 'Đăng nhập' button on the left and a grey 'Phục hồi' button on the right.

Hình 2.41 : Minh họa Text Field.

7.2.2. Textarea.

Textarea (ô nhập liệu nhiều dòng) dùng để nhập các thông tin, dữ liệu dài nhiều dòng, ví dụ như các ghi chú, ý kiến đóng góp,...

- Đặt con trỏ trong Form.
- Chọn menu **Insert / Form/ TextArea** hoặc chọn hình trên thanh công cụ **Forms**.
- Chọn **TextArea** đã tạo, chỉ định các thuộc tính trong cửa sổ **Properties** như sau:

- + **TextField:** Đặt tên cho TextArea.
- + **CharWidth:** Số ký tự cho chiều dài ô.
- + **NumLines:** Số dòng cần hiển thị.
- + **Wrap:** Văn bản tự xuống dòng.
- + **Init Value:** Giá trị mặc định ban đầu.
- + **Type:** Chọn kiểu:
 - **Single Line:** Dạng TextField 1 dòng.
 - **MultiLine:** Dạng TextArea nhiều dòng.
 - **Password:** Dạng mật mã.



Hình 2.42: Minh họa TextArea

7.2.3. Radio Group:

❖ Nút chọn lựa Radio Button.

Trong nhóm các nút chọn, bắt buộc chọn 1 và chỉ 1 mà thôi. Các Radio cùng nhóm phải cùng tên chỉ khác nhau về giá trị.

- Định vị con trỏ trong **form**.
- Chọn **Insert/ Form/ Radio Button** hoặc kích biểu tượng trên thanh công cụ **forms**.
- Chọn **Radio Button** đã tạo, chỉ định các thuộc tính từ cửa sổ **Properties**
 - + **Radio Button:** Tên nút (Các Radio Button cùng nhóm đặt cùng tên).
 - + **Checked value:** Giá trị của nút khi được chọn.
 - + **Initial State:**
 - **Checked:** Mặc định được chọn.
 - **UnChecked:** Mặc định không được chọn.

❖ Radio button Group.

Nếu không muốn tạo từng **Radio Button** riêng biệt đặt cùng tên để thành **Radion Button Group** bạn có thể chèn cả nhóm **Radio Button** như sau:

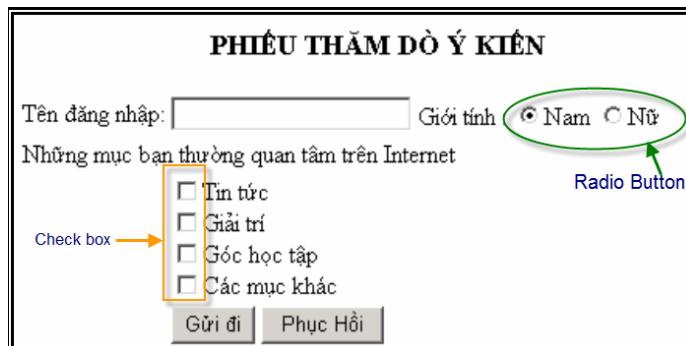
- Đặt con trỏ trong **Form**.
- Chọn menu **Insert / Form/ Radio Button Group** hoặc nhấn vào biểu tượng trên thanh công cụ **forms**.
 - Hộp thoại Radio Group xuất hiện, thiết lập các thuộc tính sau:
 - + Nhấn nút để thêm nút, nhấn dấu xoá bỏ nút.
 - + **Label:** Khai báo nhãn nút.
 - + **Value:** Giá trị khi nút được chọn gửi về cho Webserver.

- + **Layout Using:** Cách hiển thị.
 - **LineBreak:** Mỗi nút xuống dòng.
 - **Table:** Các nút nằm trong bảng.

7.2.3. Checkbox:

Hộp kiểm tra **Checkbox** tương tự như **Radio Button** nhưng cho phép tùy chọn: 1, 1 số, tất cả hoặc không.

- Đặt con trỏ trong **Form**.
- Chọn menu **Insert / Form/ CheckBox** hoặc nhấn vào biểu tượng  trên thanh công cụ **Forms**.
- Chọn **CheckBox** đã tạo, chỉ định các thuộc tính trong cửa sổ **Properties** như sau:
 - + **CheckBox:** Tên nhóm các Checkbox.
 - + **Checked Value:** Giá trị trả về cho từng CheckBox.
 - + **Initial State:** Trạng thái mặc định ban đầu.
 - **Checked:** Nút được chọn.
 - **UnChecked:** Nút chưa được chọn.



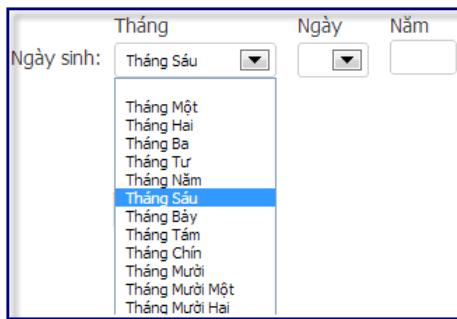
Hình 2.43 : Minh họa Radio Button và Check Box.

7.2.4. List / Menu:

List / Menu (hộp liệt kê) dùng để chứa danh sách các nội dung theo từng dòng. Khi cần xem danh sách nhấn nút mũi tên xuống bên phải hộp chọn để sổ danh sách xuống. Nếu đặt theo kiểu là menu thì chỉ được phép chọn 1 mục trong danh sách, còn nếu đặt kiểu là list thì có thể chọn nhiều mục.

- Đặt con trỏ trong Form
- Chọn menu **Insert / Form/ List Menu** hoặc nhấn vào biểu tượng  trên thanh công cụ **Forms**.
- Chọn hộp liệt kê mới được chèn, chỉ định các thuộc tính trong cửa sổ **Properties** như sau:

- **List/Menu:** Đặt tên cho hộp liệt kê
- **Type:** Chọn kiểu
- **Menu:** Dạng hộp chọn.
- **List:** Dạng danh sách chọn (khung chọn).
 - Chọn nút **List Value** xuất hiện hộp thoại **ListValue**.
 - Nhấn nút **[+]** để thêm mục chọn, nhấn nút **[-]** để xoá mục chọn, nhấn nút **[▲]** và **[▼]** để di chuyển lên xuống giữa các mục chọn.
- **ItemLabel:** Nhập nhãn cho từng mục chọn.
- **Value:** Chỉ định giá trị trả về cho mỗi mục chọn.
- Nhấn **Ok** để hoàn thành.



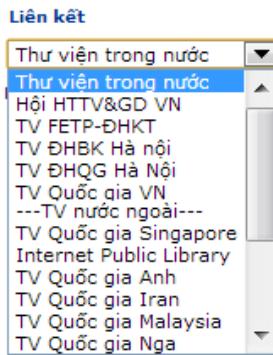
Hình 2.44 : Minh họa List/Menu

7.2.5. Jump menu:

Liên kết trong Menu/List của Form được gọi là Jump menu. Đó là cách xếp gọn danh sách liên kết thành hộp chọn, người sử dụng khi cần thì nhấn vào nút sổ danh sách xuống để chọn liên kết đó. Mặc định loại liên kết này sẽ nhảy nhanh để trang liên kết. Ngoài ra, bạn có thể tạo thêm nút Go để sau khi chọn liên kết rồi, kích nút Go thì sẽ chuyển đến trang liên kết.

- Định vị con trỏ trong Form.
- Chọn menu **Insert/Form/Jump Menu** hoặc nhấn vào biểu tượng  trên thanh công cụ Forms.
- Hộp thoại JumpMenu xuất hiện hãy khai báo các thuộc tính sau:
 - + Nhấn nút **[+]** hay **[-]** để thêm hay bớt một Menu Item, nhấn nút **[▲]** và **[▼]** để di chuyển lên xuống giữa các Menu Item.
 - + **MenuItem:** Danh sách các Menu Item được tạo.
 - + **Text:** Nhập nhãn Menu Item.
 - + **When Selected Goto URL:** Địa chỉ tập tin liên kết đến hay Mailto.

- + **Open URL in:** Target của khung hay cửa sổ trang hiển thị.
- + **Select First Item After URL Change:** Chọn Item đầu sau khi liên kết.
- + **Insert Go Button After Menu:** Chèn thêm nút Go.
- + Lặp lại quá trình tạo cho các Menu Item khác.



Hình 2.45 : Minh họa Jump Menu.

7.2.6. File Field.

Ngoài các thông tin ngắn gọn có thể nhập hoặc chọn trực tiếp, nếu khách truy cập muốn gửi nội dung văn bản nhiều dòng hơn hoặc 1 tập tin văn bản thì bạn cần tạo 1 File Field. Đây là thành phần trên Form giúp gửi kèm tập tin.

- Đặt con trỏ trong Form.
- Chọn menu **Insert / Form/ File Field** hoặc nhấn vào biểu tượng trên thanh công cụ Forms.
- Chọn File Field đã tạo, chỉ định các thuộc tính trong cửa sổ Properties như sau:

- + **File Field:** Đặt tên cho File Field.
- + **Char Width:** Số ký tự cho chiều dài ô.
- + **Max Chars:** Số ký tự nhiều nhất có thể nhập.

Khi khách truy cập thì nhấn vào nút Browse rồi tìm chọn tập tin cần gửi (hoặc nhập đường dẫn tập tin cần gửi)

7.2.7. Submit và Reset.

Trong Form cần thiết phải có 2 nút đặc biệt là Submit & Reset. Nút Submit để gửi thông tin về về người dùng cho chủ nhân Website. Nút Reset để xoá dữ liệu trên form về trạng thái mặc định ban đầu.

- Đặt con trỏ trong Form
- Chọn menu **Insert / Form/ Button** hoặc nhấn vào biểu tượng trên thanh công cụ Forms.

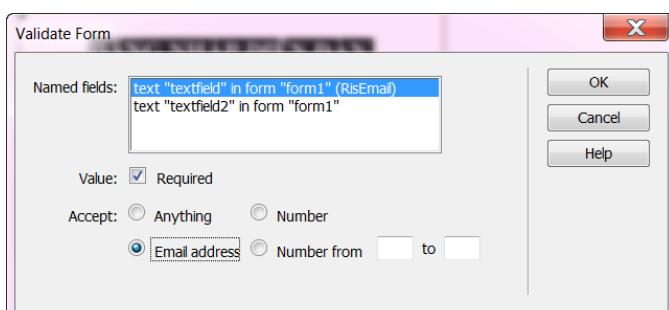
- Chọn Button đã tạo chỉ định các thuộc tính trong cửa sổ Properties như sau:
 - + **Button Name:** Đặt tên cho Button.
 - + **Label:** Nhãn hiển thị trên nút.
 - + **Action :**
 - **Submit Form:** Đóng gói dữ liệu và gửi đi
 - **Reset Form:** Xoá dữ liệu trả về trang thái mặc định ban đầu.
 - **None :** Nút bình thường cho các hành động khác như thông báo, kiểm tra.

7.2.8. Kiểm tra dữ liệu nhập trên Form.

TextField là ô nhập liệu cho người truy cập. Tuy nhiên có trường hợp họ không nhập hoặc nhập không đúng kiểu giá trị. Bạn có thể sử dụng Behavior(đáp ứng) để kiểm tra các giá trị nhập.

Ví dụ : Bạn cần thiết kế form với yêu cầu: bắt buộc phải nhập liệu cho ô Họ và tên, ô Email phải đúng là địa chỉ Mail (có ký hiệu @ . .), ô năm sinh phải nhập từ 1900 đến 2012, ô ghi thông tin có thể bỏ qua,... Để thực hiện việc kiểm tra này ta thực hiện như sau :

- Chọn Form đã tạo
- Mở cửa sổ Behavior bằng cách vào menu **Windows/ Behavior** hoặc nhấn tổ hợp phím **Shift +F4**.
- Nhấn dấu và chọn **Validate Form** .
- Hộp thoại **Validate Form** hiển thị:



Hình 2.46 : Kiểm tra giá trị nhập trên form.

- + **Name Field:** Danh sách các TextFields trên form, muốn đặt điều kiện kiểm tra TextField nào thì chọn TextField đó.
- + **Value required:** Yêu cầu TextField đó phải được nhập liệu.
- + **AnyThing:** Dữ liệu loại nào cũng được.
- + **Email Address:** Dữ liệu nhập phải là dạng địa chỉ Mail

- + **Number**: Dữ liệu nhập phải là dạng số.
- + **Number From .. To...**: Dữ liệu số trong khoảng xác định.
- Lần lượt thực hiện điều kiện cho các Field theo yêu cầu.

8. Frame & Template.

8.1. Frame.

Frame được dùng để bố trí nhiều trang web hiển thị trên cùng một cửa sổ trình duyệt. Frame bao gồm hai thành phần chính : Một frameset và các frame cá nhân:

Ví dụ: Nếu một trang web chứa 3 frame (chia trình duyệt thành 3 phần) thì sẽ có 4 trang HTML riêng biệt: 1 tệp tin frameset và 3 tệp tin nội dung. Khi thiết kế một trang sử dụng các frameset, phải lưu tất cả các tập tin trên để có thể xem được trên trình duyệt.

Việc tạo giao diện kiểu frame rất đơn giản tuy nhiên chỉ thích hợp cho những Site đơn giản vì liên kết với các frame rất phức tạp, đòi hỏi bạn phải hiểu rõ mối quan hệ giữa các frame và các frameset.

8.1.1. Tạo frame.

Trước khi tạo một frameset hay làm việc với các frame, bạn hãy bật chế độ hiển thị đường viền của frame bằng cách : Chọn menu **View/ Visual Aids / Frame Borders**.

- Đặt điểm chèn trong cửa sổ tài liệu.
- Kích biểu tượng  trong tab **Layout** của thanh công cụ hoặc chọn menu **Insert /HTML/ Frame**.
- Muốn sử dụng kiểu frameset nào thì chọn từ biểu tượng tương ứng. Có thể đặt lồng các frameset với nhau gọi là frameset kết tố.

❖ Sử dụng Panel Frame

Panel Frames cung cấp một hình ảnh trực quan của các frame có trong tài liệu.

- Mở Fanel Frames : Chọn **Window / Frames** hoặc nhấn tổ hợp phím **Shift + F12**. Cửa sổ Panel Frames sẽ hiển thị.
- Có thể kích chọn một frame hay frameset trong panel Frames bằng cách:
 - + Để chọn một frame: Kích chọn frame đó trong panel Frames .
 - + Để chọn một frameset: Kích đường viền xung quanh các frame trong Panel Frames.

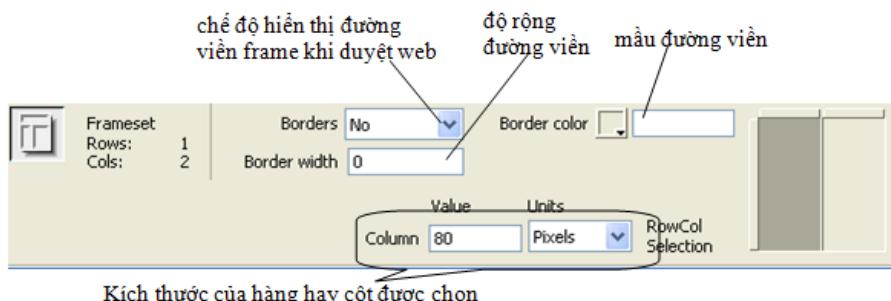
- + Kích chọn một frame hay frameset trong cửa sổ tài liệu.
- Khi chọn một frame hay frameset, đường viền chọn xuất hiện trong cửa sổ tài liệu và trong panel Frames.

8.1.2. Thiết lập các thuộc tính cho Frame

Các frame và frameset đều hiển thị các thuộc tính trên Properties inspector. Các thuộc tính của frame xác định tên, tệp tin nguồn, lề, thanh cuộn và đường viền của frame riêng biệt bên trong một frameset. Các thuộc tính frameset điều khiển các kích thước và màu và độ rộng của đường viền giữa các frame.

❖ *Thiết lập các thuộc tính frameset:*

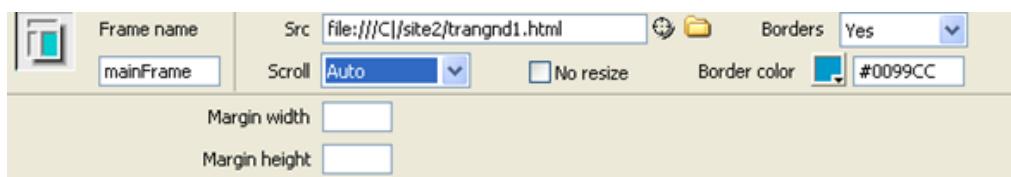
- Chọn frameset trong cửa sổ tài liệu hay trong Panel Frames.
- Hiệu chỉnh các thuộc tính trong cửa sổ **Properties**:



Hình 2.47: Thuộc tính Frameset.

❖ *Thiết lập các thuộc tính frame:*

- Chọn một frame trong Panel Frames, cửa sổ Properties sẽ hiển thị như sau:



Hình 2.48: Thuộc tính các Frame.

+ **Frame Name:** nhập tên cho frame. Frame Name xác định tên của frame hiện có để sử dụng cho mục đích siêu liên kết (hyperlink target) và tham chiếu ngôn ngữ (scripting references).

+ **Src:** xác định tài liệu nguồn cho frame. Mặc định khi mới tạo frameset thì trang frame ta chọn sẽ tạo ra một trang web trắng với tên mặc định Untitled-... để ta có thể nhập trực tiếp nội dung sau đó sẽ lưu lại hoặc ta có thể sử dụng các trang đã tạo ra từ trước bằng cách kéo các trang này từ cửa sổ Site vào frame tương ứng.

- + **Scroll Bar**: xác định có xuất hiện thanh cuộn hay không khi không đủ không gian để hiển thị nội dung trong frame.
- + **No Resize**: giới hạn kích thước của frame và ngăn người sử dụng rê các đường viền frame.
- + **Borders**: điều khiển sự hiển thị của đường viền frame. Các giá trị có thể chọn: Yes, No, Default. Chọn một tùy chọn đường viền sẽ chồng lên đường viền đã được thiết đặt cho frameset.
- + **Border Color**: thiết đặt màu viền cho frame.
- + **Margin Width**: chọn độ rộng của lề trái và lề phải.
- + **Margin Height**: chọn độ rộng của lề trên và lề dưới.

8.1.3. Lưu các Frame và Frameset.

Các tập tin của frameset và các frame thành phần của nó phải được lưu trước khi bạn xem trên trình duyệt. Có thể lưu riêng biệt một trang frame hay một trang frameset, hay lưu tất cả các tập tin frame và trang frameset đang được mở.

❖ Để lưu một tập tin trong frameset:

- Chọn frameset hoặc chọn một frame trong Panel Frames
- Chọn **File/ Save / Frameset** hoặc **File / Save Frame** hay **Save Frame As** (hoặc chọn biểu tượng tương ứng trên thanh công cụ).
- Chọn đường dẫn thích hợp và đặt tên cho tệp tin.
- Kích OK.

❖ Để lưu tất cả tập tin trong một frameset

- Chọn **File / Save All Frames**.
- Quan sát trên Panel Frame xem đang lưu frame nào để chọn đường dẫn và đặt tên thích hợp cho tệp tin.

8.1.4. Liên kết trong frameset

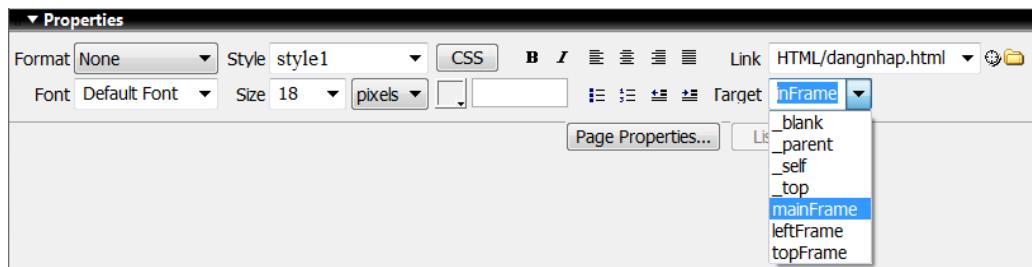
Sử dụng các liên kết trong frame, bạn phải chọn một target cho liên kết. Target là frame mà nội dung liên kết sẽ được mở ở đó.

Ví dụ: nếu menu định hướng ở frame bên trái và bạn muốn tài liệu được liên kết xuất hiện trong frame chính(mainframe) ở bên phải, bạn phải sử dụng các target cho tất cả các mối liên kết ở menu định hướng. Target đó là tên của mainframe. Khi người sử dụng kích một liên kết định hướng, nội dung sẽ mở trong mainframe.

❖ Để tạo đích cho một frame:

- Chọn văn bản hay đối tượng cần tạo liên kết.

- Ở trường **Link** trên **Properties inspector** chọn tài liệu được liên kết.
- Trong menu **Target**, chọn vị trí mà ở đó mối liên kết sẽ được mở. Nếu bạn đã đặt tên cho các frame thì tên các frame sẽ xuất hiện trong menu này. Chọn một frame được đặt tên để mở tài liệu được liên kết trong frame đó.



Hình 2.49 : *Dịch của một frame.*

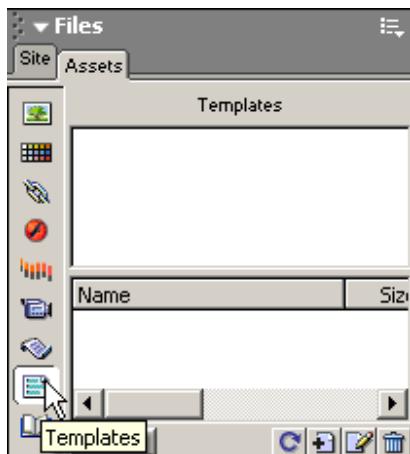
- + **_blank**: mở tài liệu được liên kết trong một cửa sổ trình duyệt mới.
- + **_parent**: mở tài liệu được liên kết trong frameset.
- + **_self**: mở tài liệu được liên kết ở trong frame hiện thời, thay thế nội dung trong frame đó.
- + **_top**: mở mối liên kết ở trong frameset ở bên ngoài tài liệu hiện hành, thay thế tất cả các frame.

8.2. Template.

Một Template là một tài liệu mẫu có thể sử dụng để tạo ra nhiều trang có cùng bố cục tương tự nhau. Khi tạo một Template, có thể chỉ ra những vùng không thể thay đổi và những vùng nào có thể thay đổi được trong các trang web được tạo dựa trên Template đó. Có thể chỉnh sửa một Template thậm chí sau khi dùng nó để tạo các tài liệu, khi đó hàng trăm trang web đã áp dụng mẫu sẽ được tự động thay đổi theo. Khi hiệu chỉnh một Template, vùng bị khoá trong các tài liệu sử dụng Template này đều được cập nhật để phù hợp với các thay đổi cho Template.

8.2.1. Tạo Template.

- Chọn **Window/Assets**, **Panel Assets** xuất hiện, chọn icon **Template**.
- Kích biểu tượng dấu cộng ở góc dưới phải của Panel Assets.



Hình 2.50 : Tạo template.

- Trong khi Template vẫn được chọn, bạn hãy gõ tên cho Template mới. Dreamweaver lưu các Template với tên có phần mở rộng là **.dwt** trong thư mục Templates. Bạn không thể di chuyển các thư mục Template ra khỏi thư mục gốc của site, vì như vậy nó sẽ bị lỗi đường dẫn trong các Template.

Bạn có thể tạo một Template từ một tài liệu HTML đã có, sau đó hiệu chỉnh nó để phù hợp với các nhu cầu của bạn như sau:

- Khi tài liệu đã được mở, chọn **File / Save as Template**.
- Trong hộp thoại **Save as template**, chọn một site trong menu, sau đó nhập tên cho Template vào trường **Save as**.

8.2.2. Thiết kế tệp tin Template

Trong Template những vùng có thể biên tập (editable regions) là những vùng có thể thay đổi trong các trang web sử dụng template, những vùng bị khoá (locked regions) là những phần không thay đổi trên mỗi trang sử dụng Template.

Tất cả các vùng trong một Template tạo mới đều được khoá. Vì vậy để tạo một Template có thể sử dụng được, bạn phải tạo các vùng có thể biên tập.

- Thêm các đối tượng cần thiết vào trong Template như đối với trang Web bình thường.
- Trong tệp tin Template, chọn vùng mà bạn muốn chuyển thành vùng có thể biên tập.
- Thực hiện một trong các cách sau:
 - + Nhấn tổ hợp phím **Ctrl + Alt + V**.
 - + Kích chuột phải lên vùng chọn trong trang, chọn **Templates** trên

menu ngữ cảnh sau đó chọn tiếp **New Editable Region**.

+ Chọn menu lệnh **Editable Region** trên nhóm biểu tượng **Templates** của thanh công cụ Insert.

- Hộp thoại **New Editable Region** hiển thị, bạn nhập tên cho vùng trong mục **Name**. Vùng có thể biên tập trong Template được viền quanh một đường màu sáng.

8.2.3. Áp dụng Template cho một tài liệu.

- Mở một trang Web trắng.
- Chọn một Template trong **Panel Assets**.
- Kích nút **Apply** 

Chú ý:

➤ Lúc bạn áp dụng một Template vào một tài liệu đang có, nội dung trong Template được bổ sung vào tài liệu. Các tài liệu được tạo dựa trên Template sẽ kế thừa các thuộc tính của trang Template trừ tiêu đề trang.

➤ Nếu bạn áp dụng một Template cho một trang web và sau đó muốn thay đổi các thuộc tính trang web, bạn phải hiệu chỉnh các thuộc tính trang của Template rồi cập nhật các trang đã sử dụng Template.

8.2.4. Sửa, xoá, đổi tên tập tin Template.

❖ **Hiệu chỉnh một tập tin template:**

- Trong **Panel Assets**, chọn biểu tượng Template.
- Chọn một Template cần hiệu chỉnh và kích vào biểu tượng **Edit** .
- Thực hiện hiệu chỉnh Template trong cửa sổ tài liệu, tạo các vùng có thể biên tập.
- Lưu Template đã hiệu chỉnh bằng cách chọn **File / Save**.

❖ **Đổi tên một template trong Panel Assets:**

- Kích vào tên của một Template trong **Panel Assets** để chọn.
- Kích chuột phải lên tên của Template, một trình đơn hiển thị bạn chọn **Rename**.

- Nhập tên mới cho Template và nhấn Enter.

❖ **Xoá một template:**

- Chọn một Template trong cửa sổ **Panel Assets**.
- Kích nút Delete  để xoá.

Tập tin template được xoá khỏi site, các tài liệu sử dụng Template không tách khỏi Template, chúng giữ lại cấu trúc các vùng có biên tập trên

Template đã xoá. Để tách tài liệu ra khỏi Template, chọn menu **Modify / Templates / Detach from Template**.

B. THỰC HÀNH.

1. Nội dung thực hành

Sử dụng các ảnh trong thư mục C:/DATA/IMAGES để thiết kế các trang web theo yêu cầu sau:

- Trang Template mẫu : mau.dwt



Hình 2.51: Trang mau.dwt

Trong đó:

- + Sử dụng kỹ thuật Navigation Bar để tạo Menu ngang bên trên và menu dọc bên trái.
- + Sử dụng JavaScript để trình diễn các ảnh mờ dần trên phần bên trái của Banner.
- + Sử dụng JavaScript để chèn lịch ngày tháng.
- Áp dụng mẫu trên vào các trang web gioithieu.htm và dangky.htm.

- Trong phần nội dung của trang dangky.htm tạo form nhập liệu như hình 2.52.

ĐĂNG KÝ THÔNG TIN CÁ NHÂN

Chi Tiết Đăng Ký
(Bắt buộc đối với những phần có dấu *)

Tên sử dụng *	<input type="text"/>	(được thể hiện trên diễn đàn)
Mật khẩu *	<input type="password"/>	
Nhập lại mật khẩu *	<input type="password"/>	
Giới tính *	<input checked="" type="radio"/> Nam <input type="radio"/> Nữ	
Email	<input type="text"/>	(cần thiết khi bạn quên mật khẩu)

Thông Tin Chi Tiết (không bắt buộc)

Họ	<input type="text"/>
Tên	<input type="text"/>
Ngày sinh	Ngày <input type="text" value="01"/> Tháng <input type="text" value="01"/> Năm <input type="text" value="1930"/>
Sở thích	<input type="text"/>
Nghề nghiệp	<input type="text" value="Học sinh"/>
Địa chỉ nhà riêng	<input type="text"/>
Địa chỉ cơ quan	<input type="text"/>
Tỉnh/Thành phố	<input type="text" value="TP Hồ Chí Minh"/>
Số điện thoại liên lạc	<input type="text"/>

Các qui định khi gia nhập trở thành thành viên của diễn đàn

Các thành viên phải tuân thủ các qui định của diễn đàn:

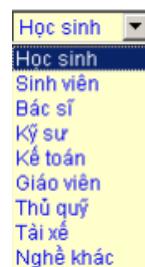
1. Các thành viên phải post mỗi tuần 1 bài viết.
2. Thời hạn trả lời sẽ là 1 tuần.
3. Diễn đàn sẽ tự động hủy những bài không có người trả lời.
4. Quản trị diễn đàn sẽ hủy tài khoản của thành viên sau 1 tuần kể từ ngày post bài viết cuối cùng.

Đồng ý các qui định của diễn đàn

Hình 2.52 : mẫu Form đăng ký.

Yêu cầu:

- + ComboBox **Tỉnh/Thành phố** hiển thị các tỉnh/tỉnh/city/country trong cả nước.
 - + ComboBox **Ngày** hiển thị ngày 01 đến 31.
 - + ComboBox **Tháng** hiển thị từ tháng 01 đến 12.
 - + ComboBox **Năm** từ năm 1930 đến 1940.
 - + ComboBox **Nghề nghiệp** hiển thị các giá trị sau:



2. Bảng vật tư thiết bị cần thiết.

TT	Thiết bị - Vật tư	Thông số kỹ thuật	Số lượng
1	Hệ thống mạng và đường truyền Internet.	LAN và ADSL	
2	Bộ máy tính.	Cài đặt hệ điều hành Windows, phần mềm Macromedia Dreamweaver 8.0	1 bộ/1 học sinh
3	Bộ đĩa lưu các file ảnh, file dữ liệu phục vụ môn học.	ảnh JPG	

3. Quy trình thực hiện.

TT	Công việc	Thao tác	Yêu cầu
1	Chuẩn bị	<ul style="list-style-type: none"> - Kiểm tra các hoạt động của máy tính. - Phần mềm Dreamweaver. - Các ảnh mẫu và Javascripts. 	MT Hoạt động bình thường. Đã cài đặt đầy đủ.
2	Thiết kế Site	<ul style="list-style-type: none"> - Tạo site. - Tạo mới các thư mục. - Copy ảnh vào thư mục IMAGES. - Tạo mới các trang web. 	Đúng cấu trúc.
3	Thiết kế file mau. dwt	<ul style="list-style-type: none"> - Tạo mới Template. - Lưu với tên là mau.dwt. - Mở mau.dwt ở chế độ design. - Chèn bảng 1(5 hàng 1cột). - Đặt màu nền toàn bảng màu đỏ đậm. - Hàng 1 bảng 1 chèn bảng 2 (cỡ 99%, màu trắng, một hàng 2 hai cột) - Cột(ô) trái bảng 2 chèn ảnh. 	- Giao diện theo mẫu. - màu sắc hài hòa. - Liên kết tốt. - Xác định chính xác vùng thay

		<ul style="list-style-type: none"> - Cột(ô) phải bảng 2 chèn JavaScript ảnh mờ dần, chỉnh thông số ảnh phù hợp với kích thước bảng. - Hàng 2 bảng 1 chèn bảng 3(cỡ 100%, 1 hàng 3 cột) - 2 Ô đầu bên trái cùng bảng 3 chèn 2 liên kết đến trang đăng ký và đăng nhập - Ô bên phải của bảng 3 tạo menu ngang bằng Navigation Bar. - Hàng 3 bảng 1 chèn bảng 4 (cỡ 99% màu nâu hồng nhạt, 1 hàng 2 cột) - Ô đầu của bảng 4, chèn bảng 5 (màu đỏ đậm, cỡ 80%, căn giữa có 1 cột, 4 hàng). - Hàng 1 bảng 5 gõ SẢN PHẨM màu trắng. - Hàng 2 bảng 5, tạo menu dọc bằng Navigation Bar - Hàng 3 bảng 5 để trống - Hàng 4 bảng 5 chèn Javascript lịch ngày tháng. - Ô phải của bảng 4 chèn bảng 6 (màu trắng, cỡ 100%, 1 cột , 2 hàng). - Hàng đầu của bảng 6, chia thành 3 ô và chèn ảnh theo mẫu. - Hàng 2 của bảng 6, đặt làm vùng thay đổi của Template. - Lưu Template. 	đổi.
4	Áp dụng mau.dwt	<ul style="list-style-type: none"> - Mở trang gioithieu.htm và trang dangky.htm. - Apply mau.dwt cho 2 trang web này. 	Áp dụng được
4	Thiết kế	<ul style="list-style-type: none"> - Chèn bảng. 	

	trang dangky.htm	<ul style="list-style-type: none"> - Chèn form. - Chèn các label. - Chèn các textbox. - Chèn các radio button. - Chèn các combobox. - Chèn Textarea. - Chèn checkbox. - Chèn nút lệnh. - Kiểm tra. 	Thiết kế form theo mẫu
5	Kiểm tra	<ul style="list-style-type: none"> - Duyệt thử web. - Chính sửa cho đúng yêu cầu. 	Các liên kết hoạt động tốt

4. Tổ chức thực hiện

Mỗi học sinh một máy, thực hành dưới sự hướng dẫn của giáo viên.

5. Kiểm tra đánh giá

- Tạo được site và trang mau.dwt : 1 điểm
- Thiết kế được trang mau.dwt đúng yêu cầu, mỹ thuật, cân đối : 5 điểm
- Thiết kế được trang dangky.htm đúng yêu cầu, mỹ thuật: 4 điểm.

CÂU HỎI VÀ BÀI TẬP BÀI 2

A. Câu hỏi trắc nghiệm

1. *Tính năng nào trong Dreamweaver cho phép hiển thị mã nguồn của tài liệu.*
 - a. Code view.
 - b. Design view.
 - c. Combined view.
 - d. Standard view.
 - e. Layout view.
2. _____ dùng để chỉ định các thuộc tính cho các đối tượng như văn bản, đoạn văn, hình ảnh.
 - a. Property Inspector.
 - b. Insert panel.
 - c. Objects panel.
 - d. Code inspector.
3. *Phần mở rộng của flash movie là:*
 - a. Swt.
 - b. Gif.
 - c. Dwt.
 - d. Swf.
4. *Thao tác chèn table*
 - a. Control + Alt + T.
 - b. Insert | Table.
 - c. Chọn icon Insert Table trên thanh Object.
 - d. Tất cả.
5. *Các thao tác nào dùng để huỷ bỏ những thao tác đã thực hiện trong tài liệu hiện hành.*
 - a. Ctrl + Z .
 - b. Ctrl + Y.
 - c. Undo.
 - d. Edit/Repeating entries.
6. *Các thành phần cơ bản của Table*
 - a. Rows, Columns, Cells.
 - b. Rows, Columns, Width.

- c. Rows, Columns, Border.
 - d. Rows, Columns, Cell spacing, Cell padding.
7. *Chọn phát biểu sai.*
- a. Chỉ có trộn các ô lại với nhau với điều kiện, các ô cần trộn phải được chọn và phần diện tích được chọn tạo thành một hình chữ nhật.
 - b. Khi chọn một ô, chỉ có thể chọn split cell
 - c. Một lần sử dụng thao tác split cell chỉ có thể chia ô được chọn thành các dòng hoặc cột.
 - d. Có thể chọn cùng lúc hai thao tác merge cell và split cell
8. *Trong 1 rollover image, nếu hai hình không cùng kích thước, khi hiển thị:*
- a. Hình rollover tự động có kích thước bằng kích thước hình original.
 - b. Hình rollover có kích thước riêng, hình original có kích thước riêng.
 - c. Hình original tự động có kích thước bằng kích thước hình rollover.
 - d. Tất cả đều sai.
9. *Không thể tạo và ứng dụng các template khi không có website*
- a. Đúng.
 - b. Sai.
10. *Đối với Layer, chọn giá trị không phải của thuộc tính Vis*
- a. Inherit.
 - b. Visible.
 - c. Hidden.
 - d. Show.
11. *Dreamweaver sử dụng giao thức gì để tải một site cục bộ lên web server?*
- a. HTTP.
 - b. FTP.
 - c. SMTP.
 - d. TCP/IP.
12. *Các templates trong Dreamweaver được lưu với phần mở rộng là gì?*
- a. Css.
 - b. Dwt.
 - c. Html.
 - d. Swf.

13. Sau khi tạo một table, thực hiện thao tác nào sau đây để tạo thêm một hàng mới vào table.

- a. Đưa con trỏ đến ô cuối cùng. Sau đó nhấn phím tab.
- b. Ctrl + M.
- c. Modify/ Table /Insert row.
- d. Insert/Table.

14. Thuộc tính Alt của hình ảnh xác định.

- a. Dòng chữ xuất hiện ở vị trí của hình ảnh khi trình duyệt không hiển thị hình.
- b. Dòng chữ xuất hiện bên trên hình ảnh khi để con trỏ lên trên hình 1 vài giây.
- c. Dòng chữ xuất hiện bên trên hình ảnh.
- d. Cả a và b.

15. Cellspacing:số pixel giữa các ô với nhau. Cellpadding:số pixel giữa nội dung ô và đường viền ô.

- a. Đúng.
- b. Sai.

16. Chọn phát biểu đúng.

- a. Chọn nút Clear Row Heights và Clear Column Widths để xoá tất cả các thuộc tính chiều rộng, chiều cao của tất cả các ô trong table.
- b. Hình nền của table có cơ chế strech như hình nền của desktop.
- c. Khi đã chọn hình nền của table, không thể chọn hình nền cho các ô trong table đó.
- d. Độ dày lớn nhất của border table là 10.

17. Chọn phát biểu sai

- a. Jump Menu không cho phép liên kết đến địa chỉ email.
- b. Jump Menu là một đối tượng của form.
- c. Jump Menu thực chất là đối tượng List/Menu có gắn kèm một số hàm của Javascript.
- d. Khi xoá form bao quanh Jump Menu, đối tượng này vẫn hoạt động bình thường.

18. Mặc định nội dung tập tin được ứng dụng bởi template bị khoá

- a. Đúng.
- b. Sai.

19. Xác định phát biểu sai.

- a. Có thể thêm, sửa, xoá các editable region.
 - b. Có thể đánh dấu nhiều ô của table như một vùng editable region.
 - c. Không phân biệt giữa 1 ô table như một vùng chỉnh sửa và nội dung của một ô table như một vùng chỉnh sửa.
 - d. Không thể thêm editable region trong tập tin htm.
20. Chọn phát biểu sai.
- a. Có thể chuyển từ layer sang table và ngược lại.
 - b. Không thể chuyển từ layer sang table và ngược lại đối với các tập tin Template và những tập tin được ứng dụng bởi template.
 - c. Để chuyển từ layer sang table, layer đó phải có ít nhất là màu nền.
 - Tương tự cho table.
 - d. Chỉ có ở chế độ Layout mới có thể chuyển từ table sang layer và ngược lại.

B. Bài tập

BÀI 1: WEBSITE CÔNG TY MAY NHÀ BÈ.

1. Tạo Website CtyMayNhaBe có các thư mục con HTMLS, Images (chứa hình & nội dung do giáo viên cung cấp).
2. Sử dụng Layout Mode thiết kế trang Template **Trangmau.dwt**

Yêu cầu:

- Một Layout Table với các Layout cell có kích thước phù hợp để thiết kế phần Banner, các nút liên kết có sử dụng ảnh thay thế khi rê chuột vào.
- Một Layout Table còn lại định nghĩa là vùng cho soạn thảo (không khoá).



3. Sử dụng trang Tempate đã tạo thực hiện tạo các trang sau (tất cả lưu trong thư mục HTMLS): Gioithieu.htm, Nhamay.htm, Sanpham.htm, Lienhe.htm.

CÔNG TY MAY NHÀ BÈ
Nhabe Garment Company

Công ty may NHÀ BÈ được thành lập trên vốn tự có

Trụ sở chính của Công ty may NHÀ BÈ

CÔNG TY MAY NHÀ BÈ là một doanh nghiệp nhà nước, thành viên của TỔNG CÔNG TY MÁY VIỆT NAM thuộc Bộ công nghiệp, với tên giao dịch quốc tế là NHABECO.

NHABECO có quá trình lịch sử từ nhiều năm nay. Nơi đây trước 1975 có tên gọi là khu chế xuất SÀI GÒN, được khởi công xây dựng năm 1972, tuy nhiên cho tới đầu năm 1975 mới hình thành cơ bản 2 xưởng may là LEGDINE và JEAN SYMI.

Trang gioithieu.htm

CÔNG TY MAY NHÀ BÈ
Nhabe Garment Company

Shop 1 Quốc lộ 1A Shop 2 An Thạnh Đông

Veston - Là sản phẩm cao cấp, được các thương gia trên thế giới ưa chuộng. Sản phẩm của Công ty may NHÀ BÈ chất lượng và giá cả hợp lý.

Hiện nay, có 90 đại lý trong cả nước, đến năm 2005 tất cả các Tỉnh, Thành trong toàn quốc sẽ có các đại lý giới thiệu và bán sản phẩm của Công ty may NHÀ BÈ.

Trang nhamay.htm

CÔNG TY MAY NHÀ BÈ
Nhabe Garment Company

Trang Sanpham.htm.

Trang Lienhe.htm.

4. Bổ sung liên kết đến các trang trong trang Template và cập nhật lại tất cả các trang.

BÀI 2 : WEBSITE CÔNG TY THUỶ SẢN NHA TRANG

1. Tạo bộ Web tên: CtythuySanNhaTrang có các thư mục con HTMLS, Images(chứa hình & nội dung do giáo viên cung cấp),
2. Sử dụng Layout Mode thiết kế trang Template **Trangmau.dwt**

Yêu cầu:

- Một Layout Table với các Layout cell có kích thước phù hợp để thiết kế phần Banner, các nút liên kết có sử dụng ảnh thay thế khi rê chuột vào. Có chuỗi chữ chạy(Well. . .), có phần Footer.
- Một Layout Table còn lại định nghĩa là vùng cho soạn thảo (không khoá).



3. Sử dụng trang Tempate đã tạo thực hiện tạo các trang sau (tất cả lưu trong thư mục HTMLS): About.htm, Product.htm, Contact.htm.

NHATRANG - FISCO

www.nhatrang-fisco.com

Wellcom to NhaTrang-Fisco.com

INTRODUCTION

NHA TRANG FISHERIES JOINT STOCK COMPANY - NHATRANG FISCO was established in 17 th, Sep 1999. With high determination to construct our company to become the factory specializing in producing refined seafoods; we have built a modern factory with large scale for processing high quality products for exporting directly to market of Japan , European, The United States, Taiwan , Hongkong , Thailand, Korea.

Contact freeze 700 kgs/2h
blast freezer 200 kgs/h
contact freezer h 700 kgs/3h

THE CAPACITY OF PROCESSING LINES

1. Personnel:

- 50 young engineers being very curious, active, quickly adapting themselves to the production of high quality product.
- 10 experienced specialists, good professional skill
- 400 worker have been well-trained in accordance with the hard requirements for industrial

Trang About.htm

NHATRANG - FISCO

www.nhatrang-fisco.com

Wellcom to NhaTrang-Fisco.com

Can you click to choose Products lines NhaTrang-Fisco

(c)Designed 2002 by Art - Design Studio -SAIGON SOFTWARE PARK.

Trang Contact.htm

NHATRANG - FISCO

www.nhatrang-fisco.com

Wellcom to NhaTrang-Fisco.com

To Fill the form your infomations :

Full name:
Company:
E.Mail:
Phone:
Requirement:

(c)Designed 2002 by Art - Design Studio -SAIGON SOFTWARE PARK

Trang Product.htm

4. Bổ sung liên kết đến các trang trong trang Template và cập nhật lại tất cả các trang.

BÀI 3: Website Du Lịch Việt Nam.

1. Tạo Website tên: VinhHaLong có các thư mục con HTMLS, thư mục con IMAGES (chứa hình, Flash & nội dung do giáo viên cung cấp).
2. Sử dụng Layout Mode thiết kế trang Template **Trangmau.dwt**



Yêu cầu:

- Một Layout Table với các Layout cell có kích thước phù hợp để thiết kế phần Banner, các nút liên kết có sử dụng ảnh thay thế.
 - Một Layout Table (bên trái), định nghĩa là vùng cho soạn thảo
3. Sử dụng trang Tempate đã tạo thực hiện tạo các trang sau (tất cả lưu trong thư mục HTML): Gioithieu.htm, Dacsan.htm, Khudulich.htm, Danhlam.htm.
 4. Bổ sung liên kết đến các trang trong trang Template và cập nhật lại tất cả các trang.



Trang Gioithieu.htm

VỊNH HẠ LONG
Vẻ đẹp thiên nhiên huyền bí

VỊNH HẠ LONG - TRANG CHỦ | GIỚI THIỆU | TIN TỨC | DU LỊCH HẠ LONG | ĀM THỰC HẠ LONG | GIỚI THIỆU WEB HAY | LIÊN HỆ

Đặc sản Hạ Long

Món mực hấp ối



Các món ăn chế biến từ mực là món ngon, mát và chứa đầy chất dinh dưỡng của đạm. Thuộc họ nhà bạch tuộc, ruốc biển có tính giòn, ngọt, khi chế biến thành món ăn rất có sức hấp dẫn. Mực hấp vị ối nổi bật bởi hương ối đặc trưng cùng vị biển hòa quyện. [...]

Tôm hùm Hạ Long

Tôm hùm

Tôm hùm thì có ở rất nhiều nơi nhưng tới Quảng Ninh mà thường thức trong lúc lênh đênh trên Vịnh Hạ Long vào những đêm hè thì chắc chắn không có nơi nào có được. Tôm hùm có nhiều loại trọng lượng khác nhau, tuỳ theo số lượng khách mà nhà hàng có thể [...]

Tìm kiếm thông tin...

THÔNG TIN DU LỊCH HẠ LONG

- » DANH LAM THẮNG CẢNH HẠ LONG
- » KHU DU LỊCH, GIẢI TRÍ HẠ LONG
- » KHÁCH SẠN - NHÀ NGHỈ HẠ LONG
- » NHÀ HÀNG HẠ LONG
- » ĐẶC SẢN HẠ LONG

DANH BẠ WEBSITE

Trang Dacsan.htm

VỊNH HẠ LONG
Vẻ đẹp thiên nhiên huyền bí

VỊNH HẠ LONG - TRANG CHỦ | GIỚI THIỆU | TIN TỨC | DU LỊCH HẠ LONG | ĀM THỰC HẠ LONG | GIỚI THIỆU WEB HAY | LIÊN HỆ

Khu du lịch - Vui chơi giải trí

Bãi tắm Ti Tốp



Cách Bãi Cháy chừng 14 km về phía đông. Bãi tắm này có hình vầng trăng ôm trọn lấy chân đảo, bãi cát tuy nhô nhưng rất thoáng đãng và yên tĩnh, bốn mùa nước sạch và trong xanh, đặc biệt phong cảnh trời mây, non nước ở đây tuyệt đẹp.

Sở dĩ có tên là [...]

Bãi tắm Quan Lan



Bãi nằm ở đảo Quan Lan trong vịnh Bái Tử Long, giữa hai xã Minh Châu và Quan Lan thuộc huyện Vân Đồn, cách thành phố Hạ Long 55 km. Đây là bãi biển đẹp, còn giữ nguyên được vẻ hoang sơ, môi trường sinh thái chưa bị ô nhiễm. Nước biển xanh ngắt, sóng [...]

Tìm kiếm thông tin...

THÔNG TIN DU LỊCH HẠ LONG

- » DANH LAM THẮNG CẢNH HẠ LONG
- » KHU DU LỊCH, GIẢI TRÍ HẠ LONG
- » KHÁCH SẠN - NHÀ NGHỈ HẠ LONG
- » NHÀ HÀNG HẠ LONG
- » ĐẶC SẢN HẠ LONG

DANH BẠ WEBSITE

Trang Khudulich.htm.

Vẻ đẹp thiên nhiên huyền bí

VỊNH HẠ LONG - TRANG CHỦ | GIỚI THIỆU | TIN TỨC | DU LỊCH HẠ LONG | ĀM THỰC HẠ LONG | GIỚI THIỆU WEB HAY | LIÊN HỆ

Danh lam thắng cảnh HẠ LONG - Cảnh đẹp vịnh HẠ LONG - Địa điểm du lịch HẠ LONG

Chùa Long Tiên



Vị trí: Chùa tọa lạc tại phố Long Tiên, dưới chân núi Bài Thơ, Tp. Hạ Long, tỉnh Quảng Ninh.

Đặc điểm: Đây là ngôi chùa lớn nhất, là di tích lịch sử danh thắng nổi tiếng ở Tp. Hạ Long.

Chùa Long Tiên được xây dựng vào năm 1941, chùa có phong c [...]

Cụm di tích núi Bài Thơ



Vị trí: Núi Bài Thơ ở trung tâm Tp. Hạ Long, tỉnh Quảng Ninh.

Đặc điểm: Đây là một di tích có giá trị văn hóa lịch sử quan trọng. Nơi đây còn lưu giữ nhiều bài thơ khắc trên đá của một số vị vua đi kinh lý, cảm hứng trước vẻ đẹp thần tiên của Vịnh [...]

Tìm kiếm thông tin...

THÔNG TIN DU LỊCH HẠ LONG

- » DANH LAM THẮNG CẢNH HẠ LONG
- » KHU DU LỊCH, GIẢI TRÍ HẠ LONG
- » KHÁCH SẠN - NHÀ NGHỈ HẠ LONG
- » NHÀ HÀNG HẠ LONG
- » ĐẶC SẢN HẠ LONG

DANH BẠ WEBSITE

Trang danhlam.htm

BÀI 3 : XÂY DỰNG WEB ĐỘNG

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.*
- *Xây dựng được một ứng dụng Website cơ bản.*

A. LÝ THUYẾT

1. Tổng quan về ASP.NET và ADO.NET.

1.1. Tổng quan về ASP.NET

1.1.1. Giới thiệu về ASP.NET

ASP.NET là Active Server Pages .NET (.NET ở đây là .NET framework). Nói đơn giản, ngắn và gọn thì ASP.NET là một công nghệ có tính cách mạng dùng để phát triển các ứng dụng về mạng hiện nay cũng như trong tương lai. Bạn lưu ý ở chỗ ASP.NET là một phương pháp tổ chức hay khung tổ chức (framework) để thiết lập các ứng dụng hết sức hùng mạnh cho mạng dựa trên CLR (Common Language Runtime) chứ không phải là một ngôn ngữ lập trình. Ngôn ngữ lập trình được sử dụng trong giáo trình này là C#.NET (C Sharp .NET) và C#.NET chỉ là một trong 25 ngôn ngữ .NET hiện nay được dùng để phát triển các trang ASP.NET.

Tuy mang họ tên gần giống như ASP cổ điển nhưng ASP.NET không phải là ASP. ASP.NET được phác thảo (re-design) lại từ số không, nó được thay đổi tận gốc rễ và phát triển (developer) phù hợp với yêu cầu hiện nay cũng như vạch một hướng đi vững chắc cho tương lai công nghệ thông tin.

Lưu ý: Mặc dù ASP.NET và ASP khác biệt nhau nhưng chúng có thể hoạt động hài hòa với nhau trong WebServer của bạn. Do đó, khi bạn cài ASP.NET, bạn không cần lặp trình lại các ứng dụng hiện có dưới dạng ASP của bạn.

SỰ THAY ĐỔI CƠ BẢN

ASP đã và đang thi hành sứ mạng được giao cho nó để phát triển mạng một cách tốt đẹp như vậy thì tại sao ta cần phải đổi mới hoàn toàn?

Lý do đơn giản là ASP không còn đáp ứng đủ nhu cầu hiện nay trong lãnh vực phát triển mạng của công nghệ Tin Học.

Trong khi đó, ASP.NET là một cơ cấu trong các cơ cấu của hệ điều hành Windows dưới dạng nền hay khung .NET (.NET framework), như vậy ASP.NET không những có thể dùng các object của các ứng dụng cũ mà còn có thể sử dụng tất cả mọi tài nguyên mà Windows có một cách dễ dàng.

Sự thay đổi của ASP.NET:

- Tập tin của ASP.NET (ASP.NET file) có đuôi mở rộng là .ASPX, còn tập tin của ASP là .ASP.
- Tập tin của ASP.NET (ASP.NET file) được phân tích ngữ pháp bởi XSPISAPI.DLL, còn tập tin của ASP được phân tích bởi ASP.DLL.
- ASP.NET là kiểu mẫu lập trình phát động bằng sự kiện (event), còn các trang ASP được thi hành theo thứ tự tuần tự từ trên xuống dưới.
- ASP.NET sử dụng trình biên dịch (compiled code) nên rất nhanh, còn ASP dùng trình thông dịch (interpreted code) do đó hiệu suất và tốc độ chậm hơn nhiều.
- ASP.NET hỗ trợ gần 25 ngôn ngữ lập trình mới với .NET và chạy trong môi trường biên dịch (compiled environment), còn ASP chỉ chấp nhận VBScript và JavaScript nên ASP chỉ là một scripted language trong môi trường thông dịch. Không những vậy, ASP.NET còn kết hợp nhuần nhuyễn với XML (Extensible Markup Language) để chuyển vận các thông tin qua mạng.
- ASP.NET hỗ trợ tất cả các trình duyệt và quan trọng hơn nữa là hỗ trợ các thiết bị di động (mobile devices). Chính các thiết bị di động, mà mỗi ngày càng phổ biến, đã khiến việc dùng ASP trong việc phát triển mạng nhằm vươn tới thị trường mới trở nên vô cùng khó khăn.

1.1.2. Lịch sử phát triển.

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

trợ nhiều ngôn ngữ: VBScript, JavaScript. Chính những ưu điểm đó, ASP đã được yêu thích trong một thời gian dài.

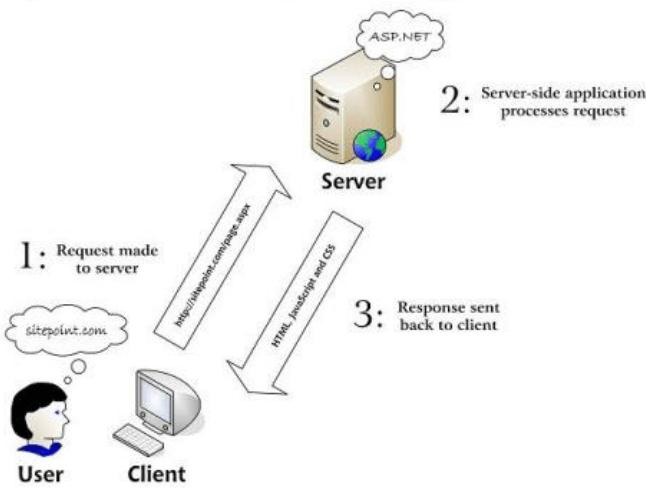
Tuy nhiên, ASP vẫn còn tồn đọng một số khó khăn như Code ASP và HTML lẫn lộn, điều này làm cho quá trình viết code khó khăn, thể hiện và trình bày code không trong sáng, hạn chế khả năng sử dụng lại code. Bên cạnh đó, khi triển khai cài đặt, do không được biên dịch trước nên dễ bị mất source code.Thêm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

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

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

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

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



Hình 3.1. Người sử dụng tương tác với một ứng dụng Web

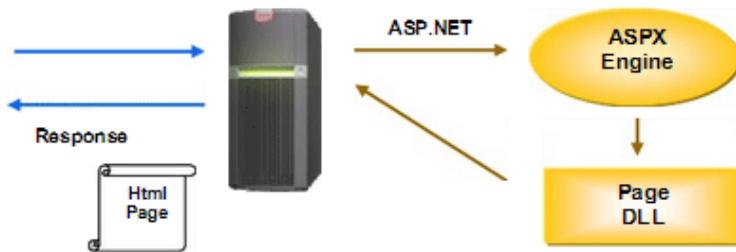
ASP.NET là một kỹ thuật dành cho việc phát triển các ứng dụng web. Một ứng dụng web đơn giản chỉ các trang web động. Các ứng dụng thường được lưu trữ thông tin trong *database* và cho phép khách truy cập có thể truy xuất và thay đổi thông tin. Nhiều kỹ thuật và ngôn ngữ lập trình khác cũng đã được phát triển để tạo ra các ứng dụng web như *PHP, JSP, Ruby on Rails, CGI* và *ColdFusion*. Tuy nhiên thay vì trói buộc bạn vào một ngôn ngữ và một công nghệ nhất định, ASP.NET cho phép bạn viết ứng dụng web bằng các loại ngôn ngữ lập trình quen thuộc khác nhau.

ASP.NET sử dụng *.NET Framework*, *.NET Framework* là sự tổng hợp tất cả các kỹ thuật cần thiết cho việc xây dựng một ứng dụng nền *desktop*, ứng dụng *web*, *web services*... thành một gói duy nhất nhằm tạo ra cho chúng khả năng giao tiếp với hơn 25 ngôn ngữ lập trình.

Thậm chí với những sự lý giải kỹ càng như vậy, bạn vẫn ngạc nhiên tự hỏi điều gì làm nên một ASP.NET tốt như vậy. Sự thật là có rất nhiều kỹ thuật *server-side* với điểm mạnh và điểm yếu riêng nhưng ASP.NET có những tính năng gần như là duy nhất.

1.1.3. Những ưu điểm của ASP.NET

- ASP.net cho phép bạn lựa chọn một trong các ngôn ngữ lập trình mà bạn yêu thích: Visual Basic .NET, C#, J# ...
- Trang ASP.NET được biên dịch trước. Thay vì phải đọc và thông dịch mỗi khi trang web được yêu cầu, ASP.NET biên dịch những trang web động thành những tập tin DLL mà server có thể thi hành nhanh chóng và hiệu quả. Yếu tố này là một bước nhảy vọt đáng kể so với kỹ thuật thông dịch của ASP.



Hình 3.2. Quá trình biên dịch của một trang ASP.NET

- ASP.NET hỗ trợ phong phú bộ thư viện mạnh mẽ và đa dạng của .NET Framework, làm việc với XML, Web service, truy cập cơ sở dữ liệu thông qua ADO.NET ...
- ASP.NET và ASP có thể cùng hoạt động trong một ứng dụng
- ASP.NET sử dụng phong cách lập trình mới: Code behind, tách code riêng, giao diện riêng do đó làm cho việc quản lý và bảo trì trở nên dễ dàng.
- Kiến trúc lập trình giống ứng dụng trên Windows.
- Hỗ trợ quản lý trạng thái của các Control.
- Tự động phát sinh mã HTML cho các Server control tương ứng với từng loại Browse.
- Hỗ trợ nhiều cơ chế Cache.
- Triển khai khi cài đặt.
 - + Không cần lock, không cần đăng ký DLL.
 - + Cho phép nhiều hình thức đăng ký ứng dụng.
- Hỗ trợ quản lý ở mức toàn cục
 - + Global.asax có nhiều sự kiện hơn.
 - + Quản lý Session trên nhiều server, không cần Cookies.

1.2. Tổng quan về ADO.NET.

Phần này sẽ giới thiệu các khái niệm chính của ADO.NET và các đối tượng mà bạn sẽ học trong các bài tiếp theo. Ngoài việc hiểu rõ hơn về khái niệm ADO.NET, bạn sẽ được biết thêm về các khái niệm cơ bản khác như data provider và các đối tượng connection, command, DataReader, DataSet, DataAdapter.

1.2.1. Giới thiệu.

ADO.NET là một bộ các thư viện hướng đối cho phép bạn tương tác với dữ liệu nguồn. Thông thường thì dữ liệu nguồn là một cơ sở dữ liệu (database), nhưng nó cũng có thể là file text, exel hoặc XML. Theo những

mục tiêu của hướng dẫn này, chúng ta sẽ chỉ xem xét tới cách ADO.NET làm việc với database.

Như bạn có thể biết rằng, có rất nhiều loại database hiện nay như Microsoft SQL Server, Microsoft Access, Oracle, Borland Interbase, và IBM DB2,... Để làm rõ hơn phạm vi của loạt bài này, tất cả ví dụ sẽ sử dụng SQL Server.

1.2.2. Data Provider

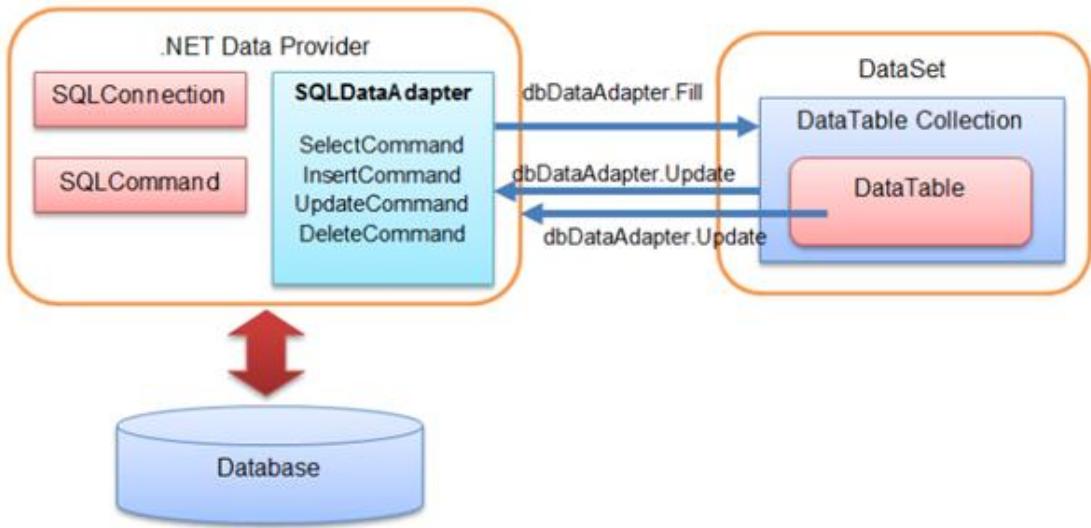
Chúng ta biết rằng ADO.NET cho phép tương tác với các loại dữ liệu và kiểu database. Mỗi loại dữ liệu cần một cách thức khác nhau để có thể truy xuất. Các loại dữ liệu cũ sử dụng giao thức ODBC, các loại dữ liệu mới hơn sử dụng giao thức OLEDB. Vì vậy cần có một thư viện thống nhất để làm việc với chúng, đây chính là lý do mà ADO.NET được tạo ra.

ADO.NET cung cấp một cách thức chung để tương tác với nguồn dữ liệu, nhưng với mỗi loại dữ liệu bạn phải sử dụng một thư viện khác nhau. Các thư viện này được gọi là Data Provider và thường được đặt tên theo giao thức hoặc loại dữ liệu mà chúng cho phép bạn truy xuất. Bảng sau liệt kê các data provider phổ biến, tiền tố (prefix) API mà chúng sử dụng và kiểu dữ liệu mà bạn có thể làm việc.

Tên Provider	API prefix	Mô tả
ODBC Data Provider	Odbc	Sử dụng cho các hệ cơ sở dữ liệu cũ
OleDb Data Provider	OleDb	Nguồn dữ liệu sử dụng giao diện OLEDB, ví dụ: Access or Excel.
Oracle Data Provider	Oracle	Cho cơ sở dữ liệu Oracle
SQL Data Provider	Sql	Cho cơ sở dữ liệu Microsoft SQL Server.
Borland Data Provider	Bdp	Cho nhiều loại cơ sở dữ liệu như SQL Server, IBM DB2, and Oracle.

Một ví dụ có thể giúp bạn hiểu nghĩa của API prefix. Một trong những đối tượng ADO.NET bạn sẽ học là connection, chúng cho phép thiết lập một kết nối đến nguồn dữ liệu. Nếu dùng OLEDB Data Provider để kết nối, sẽ cần dùng đến đối tượng connection tên là OleDbConnection. Tương tự, đối tượng connection sẽ dùng prefix là Odbc hoặc Sql cho đối tượng OdbcConnection hoặc SqlConnection để kết nối với nguồn dữ liệu tương

ứng. Như đã nói lúc đầu, ta sẽ học cách làm việc với SQL Server nên các đối tượng API có prefix là Sql.



Hình 3.3. Kiến trúc truy cập dữ liệu ADO.NET

1.2.3. Các đối tượng của ADO.NET

ADO.NET bao gồm nhiều đối tượng bạn có thể dùng với dữ liệu. Phần này giới thiệu một vài đối tượng chính bạn sẽ sử dụng. Theo tiến độ trong các bài viết sau, bạn sẽ khám phá thêm nhiều đối tượng ADO.NET và cách chúng được sử dụng trong mỗi phần riêng. Các đối tượng dưới đây bắt buộc bạn phải biết để làm việc với dữ liệu khi dùng ADO.NET.

❖ **SqlConnection**

Để tương tác với database, bạn phải có một kết nối tới nó. Kết nối giúp xác định database server, database name, user name, password, và các tham số cần thiết để kết nối tới database. Một đối tượng connection được dùng bởi đối tượng command vì thế chúng sẽ biết database nào để thực thi lệnh.

❖ **SqlCommand**

Quá trình tương tác với database cần phải biết hành động nào bạn muốn xảy ra. Điều này được thực hiện bởi đối tượng command. Bạn dùng đối tượng command để gửi một câu lệnh SQL tới database. Một đối tượng command dùng một đối tượng connection để xác định database nào sẽ được truy xuất. Bạn có thể dùng một đối tượng command riêng lẻ để thực thi lệnh trực tiếp, hoặc để gắn một tham chiếu của đối tượng command cho một SqlDataAdapter – đối tượng giữ các command sẽ làm việc trên một nhóm dữ liệu như sẽ đề cập tới trong phần dưới.

❖ **SqlDataReader**

Nhiều thao tác dữ liệu đòi hỏi bạn chỉ lấy một luồng dữ liệu để đọc. Đối tượng data reader cho phép bạn lấy được kết quả của một câu lệnh SELECT từ một đối tượng command. Để tăng hiệu suất, dữ liệu trả về từ một data reader là một luồng dữ liệu fast forward-only. Có nghĩa là bạn chỉ có thể lấy dữ liệu từ luồng theo một thứ tự nhất định. Mặc dù điều này có lợi về mặt tốc độ, nhưng nếu bạn cần phải thao tác dữ liệu, thì một DataSet sẽ là một đối tượng tốt hơn để làm việc.

❖ **DataSet**

Đối tượng DataSet là một thể hiện của dữ liệu trong bộ nhớ. Chúng chứa nhiều đối tượng DataTable, bên trong DataTable lại có nhiều column và row, giống như các database table thông thường. Bạn thậm chí có thể định nghĩa dữ liệu giữa các table để tạo các quan hệ parent-child. DataSet được thiết kế đặc biệt để giúp quản lý dữ liệu trong bộ nhớ và để hỗ trợ các thao tác không cần kết nối (disconnected) trên dữ liệu. DataSet là một đối tượng được dùng bởi tất cả Data Provider, đó là lý do tại sao nó không có một Data Provider prefix trong tên gọi.

❖ **SqlDataAdapter**

Đôi lúc dữ liệu mà bạn làm việc là read-only và bạn ít khi cần thay đổi dữ liệu nguồn. Vài trường hợp cần lưu trữ tạm dữ liệu trong bộ nhớ để hạn chế truy xuất đến database. Data adapter làm điều này dễ dàng bằng cách giúp bạn quản lý dữ liệu trong chế độ ngắt kết nối. Data adapter sẽ đổ vào DataSet khi đọc dữ liệu và thực hiện thay đổi dữ liệu một lượt vào database.

Data adapter chứa một tham chiếu đến đối tượng connection và mở/đóng kết nối tự động khi đọc và ghi dữ liệu vào database. Hơn nữa, data adapter chứa đối tượng command cho những thao tác SELECT, INSERT, UPDATE và DELETE trên dữ liệu. Bạn sẽ có một data adapter được định nghĩa cho mỗi table trong một DataSet và nó sẽ quản lý các giao tiếp với database cho bạn. Tất cả những gì bạn cần làm là chỉ cho data adapter khi nào nạp hoặc ghi vào database.

❖ **Tổng kết**

ADO.NET là một kỹ thuật .NET để thao tác với nguồn dữ liệu. Bạn có một vài Data Provider, cho phép bạn giao tiếp với các nguồn dữ liệu khác nhau, dựa trên giao thức mà chúng dùng hoặc kiểu database. Không

cần quan tâm đến điều này, với mỗi Data Provider được sử dụng, bạn sẽ dùng các đối tượng tương tự nhau để thao tác với dữ liệu. Đối tượng SqlConnection cho phép bạn quản lý một kết nối đến nguồn dữ liệu. SqlCommand cho phép bạn gửi lệnh đến dữ liệu. Để đọc dữ liệu nhanh theo cơ chế forward-only, sử dụng SqlDataReader. Nếu bạn muốn làm việc với dữ liệu đã ngắt kết nối, dùng một DataSet và hiện thực việc đọc và ghi đến dữ liệu nguồn bằng một SqlDataAdapter.

1.3.Cài đặt và cấu hình IIS.

1.3.1. Giới thiệu về IIS.

Microsoft Internet Information Services (các dịch vụ cung cấp thông tin Internet) là các dịch vụ dành cho máy chủ chạy trên nền Hệ điều hành Window nhằm cung cấp và phân tán các thông tin lên mạng, nó bao gồm nhiều dịch vụ khác nhau như Web Server, FTP Server,... Nó có thể được sử dụng để xuất bản nội dung của các trang Web lên Internet hoặc Intranet bằng việc sử dụng “Phương thức chuyển giao siêu văn bản“ - Hypertext Transport Protocol (HTTP).

Như vậy, sau khi bạn thiết kế xong các trang Web của mình, nếu bạn muốn đưa chúng lên mạng để mọi người có thể truy cập và xem chúng thì bạn phải nhờ đến một Web Server, ở đây là IIS. Nếu không thì trang Web của bạn chỉ có thể được xem trên chính máy của bạn hoặc thông qua việc chia sẻ tệp (file sharing) như các tệp bất kỳ trong mạng nội bộ mà thôi.

Nhiệm vụ của IIS là tiếp nhận yêu cầu của máy trạm và đáp ứng lại yêu cầu đó bằng cách gửi về máy trạm những thông tin mà máy trạm yêu cầu. Bạn có thể sử dụng IIS để:

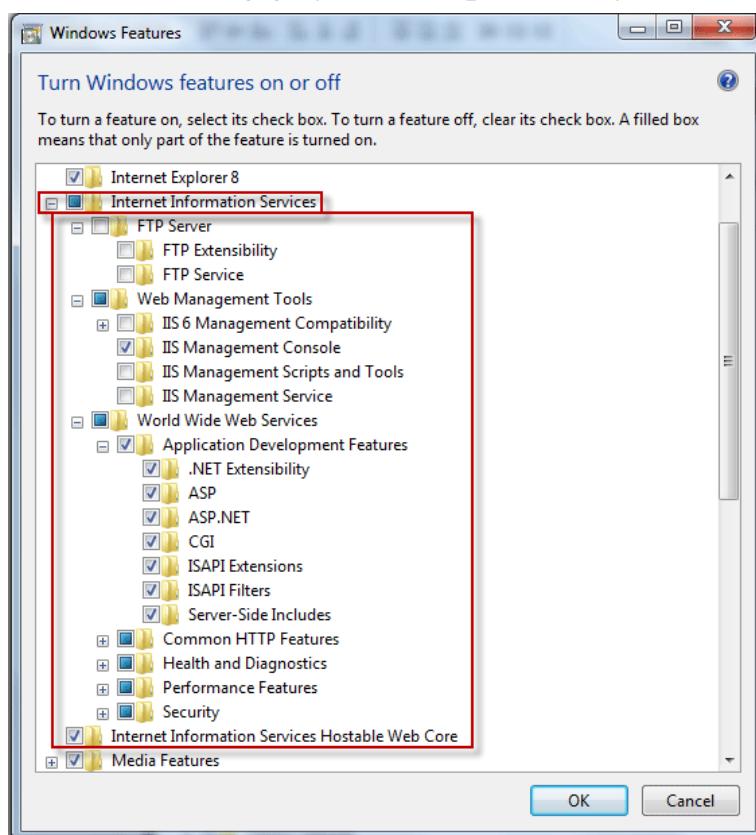
- Xuất bản một Website của bạn trên Internet
- Tạo các giao dịch thương mại điện tử trên Internet (hiện các catalog và nhận được các đơn đặt hàng từ người mua hàng)
- Chia sẻ file dữ liệu thông qua giao thức FTP.
- Cho phép người ở xa có thể truy xuất database của bạn (gọi là Database remote access).

IIS sử dụng các giao thức mạng phổ biến là HTTP (Hyper Text Transfer Protocol) và FPT (File Transfer Protocol) và một số giao thức khác như SMTP, POP3,... để tiếp nhận yêu cầu và truyền tải thông tin trên mạng với các định dạng khác nhau.

Một trong những dịch vụ phổ biến nhất của IIS mà chúng ta quan tâm là dịch vụ WWW (World Wide Web), nói tắt là dịch vụ Web. Dịch vụ Web sử dụng giao thức HTTP để tiếp nhận yêu cầu (Requests) của trình duyệt Web (Web browser) dưới dạng một địa chỉ URL (Uniform Resource Locator) của một trang Web và IIS phản hồi lại các yêu cầu bằng cách gửi về cho Web browser nội dung của trang Web tương ứng.

1.3.2. Cài đặt IIS trên Windows 7

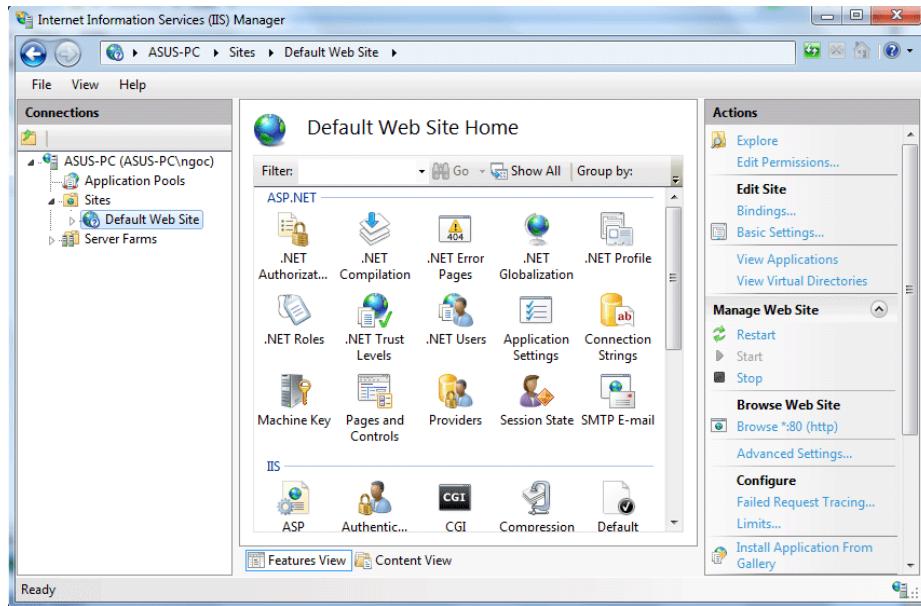
- Kích chọn **Control Panel/All Control Panel Items/Programs and Features** → Chọn **Turn Windows features on or off**
 - Hộp thoại **Window Features hiển thị** (hình 3.4).
 - Chọn **Internet Information Services**.
 - Chọn **World Wide Web Services**.
 - Chọn hết các mục bên trong **Application Development Features** để hỗ trợ chạy các ứng dụng ASP/ASP.NET
 - Kích **OK** và chờ trong giây lát để cập nhật thay đổi.



Hình 3.4. Cài đặt IIS trên Windows 7

Khởi động IIS như sau : Kích chọn **Control Panel/All Control Panel Items/Administrative Tools**.

Lưu ý: Nếu như không thể khởi động được IIS không được vì lỗi port 80 thì bạn có thể đổi port lại. Vì đôi khi như máy đã thiết lập PHP Server ở port 80, cần đổi lại port thành 8080.



Hình 3.5. Giao diện quản lý Web trên IIS

Kiểm tra trên trình duyệt bằng cách gõ vào thanh địa chỉ <http://localhost>. Nếu thành công sẽ hiển thị như hình sau:

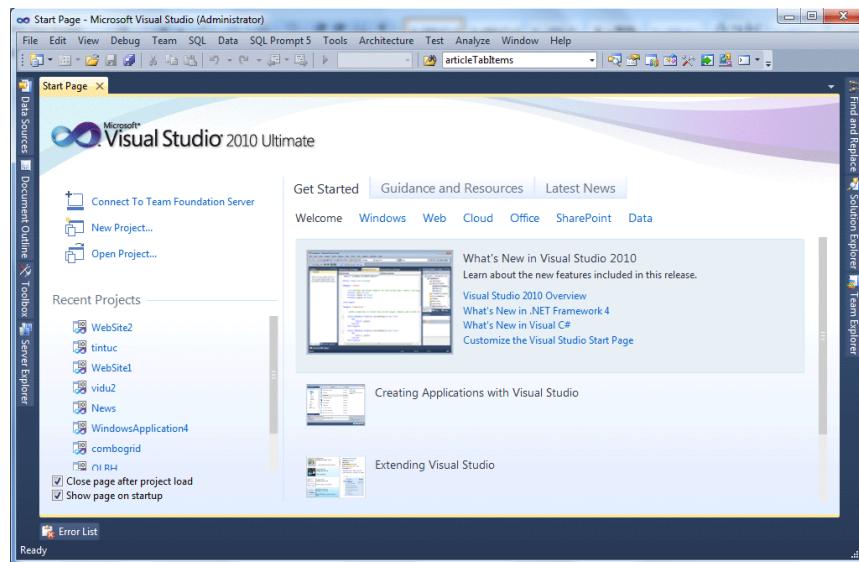


Hình 3.6. Hiển thị kết quả khi cài đặt thành công

1.4. Tạo ứng dụng ASP.NET đầu tiên

Chúng ta bắt đầu bằng việc làm quen với môi trường phát triển ứng dụng (IDE) của Visual Studio 2010. Hình 3.7 là màn hình khởi đầu của Visual Studio 2010 hình hiển thị chính giữa đang hiển thị trang Start Page, Recent Projects.

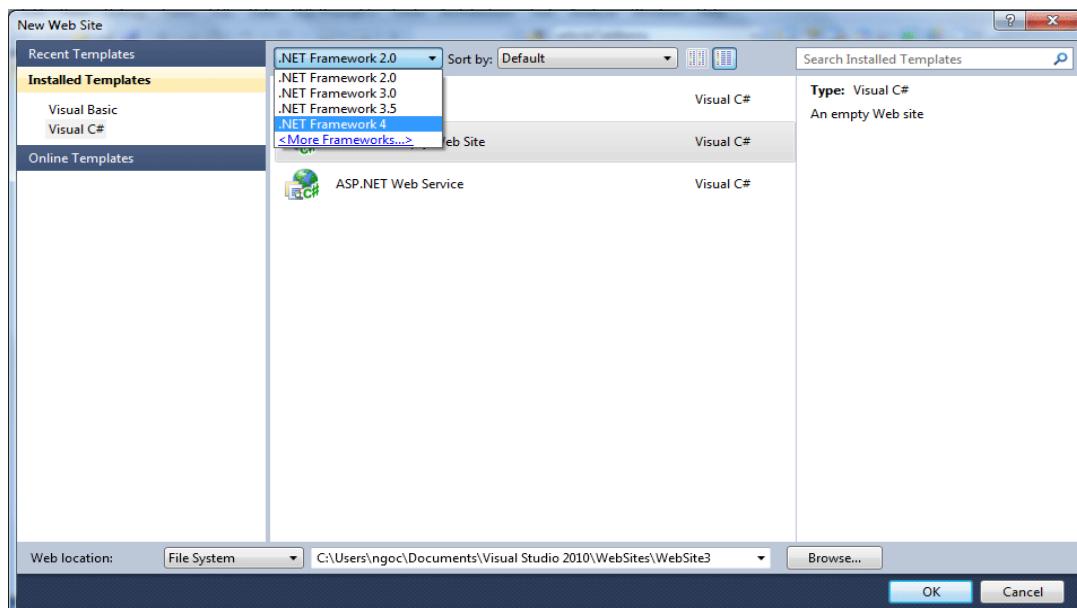
Recent Projects liệt kê các project mà chúng ta đã làm việc trong thời gian gần đây. Trên mục này, chúng ta cũng có thể tạo mới một project bằng cách nhấp vào nút **New Project** hoặc mở một Project đã có bằng cách nhấp vào nút **Open Project**.



Hình 3.7. Cửa sổ giao diện MS Visual Studio 2010

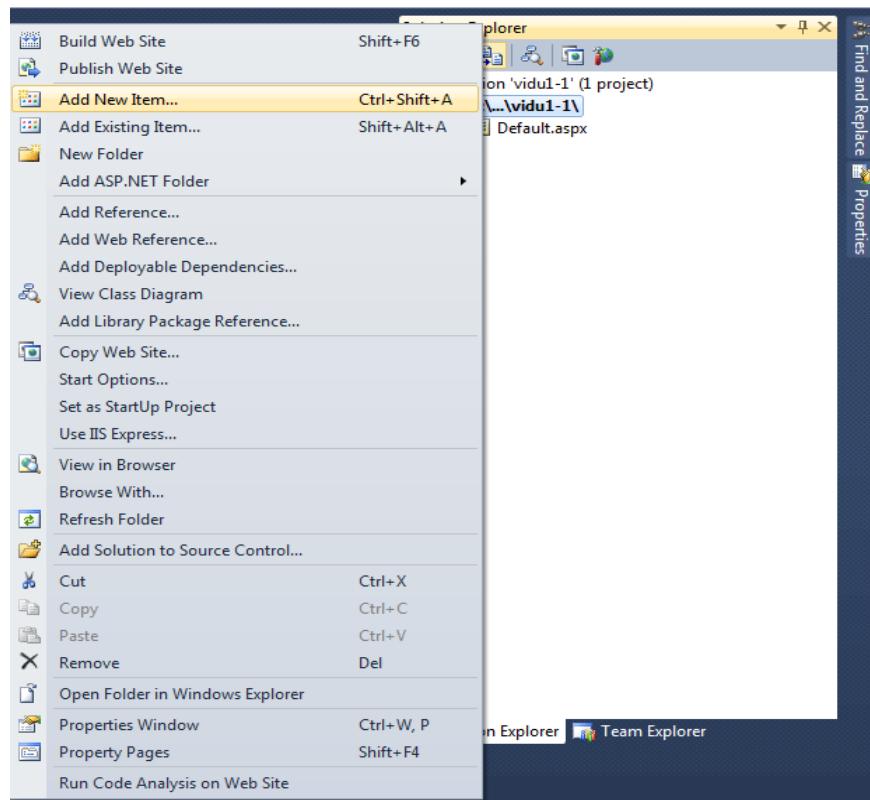
Chúng ta có thể tạo ứng dụng ASP.Net sử dụng Visual C# Project theo các bước sau:

- Bước 1. Chọn từ thực đơn File / New/ WebSite. Xuất hiện hộp thoại tạo mới Project.
 - + Chọn loại *Language* là **Visual C#**
 - + **Chọn ASP.Net Web Site** từ vùng **Templates**.
 - + Ứng dụng mới được tạo mặc định có tên là **WebSiteXX** (XX là số thứ tự tự động). Chúng ta có thể thay đổi tên của Project tại mục **Location**. Trong ví dụ này, chúng ta thay đổi tên Project **WebSite1** thành **Vidu1-1**.



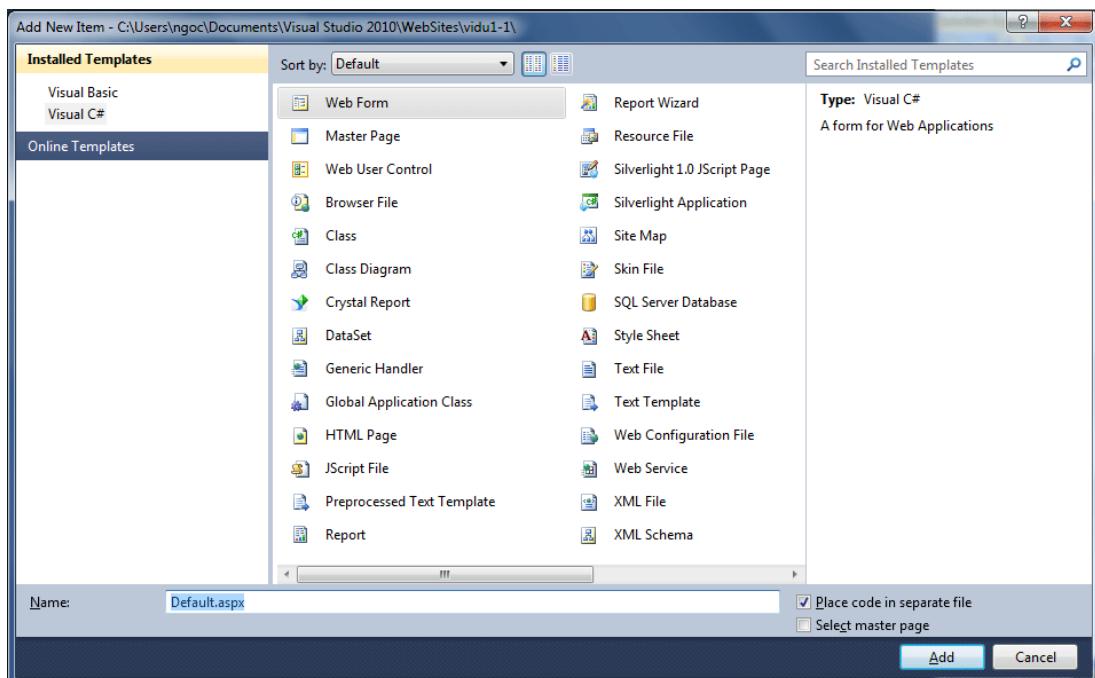
Hình 3.8. Màn hình tạo mới Website

- Bước 2: Tạo một trang Web mới bằng cách kích chuột phải trên thanh Solution Explorer / Tên Project Chọn **Add New Item**.



Hình 3.9. Tạo một trang mới

Sau đó màn hình hiển thị một cửa sổ mới, ta chọn **Web form** và gõ tên trang Web ở phần **Name**.



Hình 3.10. Giao diện Tạo một tệp mới

- Bước 3: Thiết kế giao diện thực thi và ứng dụng ;

Trên hộp công cụ (Toolbox), nếu chưa có hộp công cụ chọn **View/ToolBox**, mở thẻ **Standard** (chứa các Web Server Control) kích lần

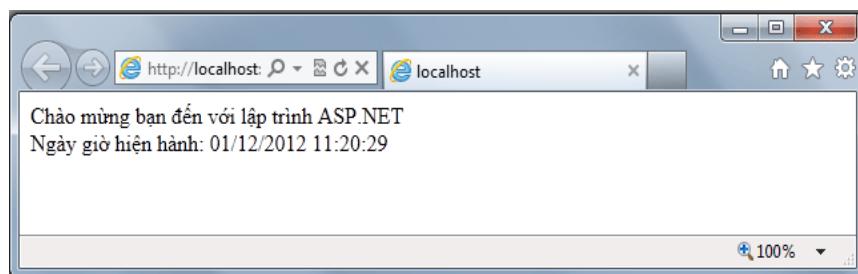
lượt vào hai điều khiển (Control) nhãn (Label) và dán vào trang Default.aspx.

Để viết lệnh cho trang Default.aspx các bạn vào menu View | Code hay nhấn phím F7, màn hình viết lệnh xuất hiện và viết lệnh cho sự kiện Page_Load.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Web;
4  using System.Web.UI;
5  using System.Web.UI.WebControls;
6
7  public partial class _Default : System.Web.UI.Page
8  {
9      protected void Page_Load(object sender, EventArgs e)
10     {
11         Label1.Text = "Chào mừng bạn đến với lập trình ASP.NET";
12         Label2.Text = "Ngày giờ hiện hành: " +
13             DateTime.Now.ToString("dd/MM/YYYY hh:mm:ss");
14     }
15 }
```

Hình 3.11: Màn hình soạn thảo mã lệnh

Nhấn F5 hoặc Ctrl + F5 để thi hành ứng dụng. Ứng dụng sẽ được biên dịch (compiler) để kiểm tra lỗi và hiện nội dung ra trình duyệt mặc định trên máy.



Hình 3.12. Màn hình hiển thị kết quả trang Default.aspx

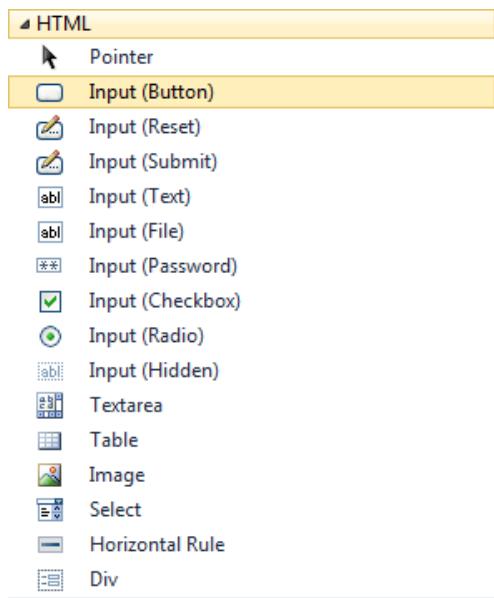
2. WEB SERVER CONTROL

ASP.NET cung cấp hai loại điều khiển trình chủ là: **HTML Server Control** và **ASP.NET Server Control**.

2.1. HTML Control

Điều khiển HTML trong trang ASP.NET mặc định là những chuỗi văn bản bình thường. Để có thể được sử dụng lập trình phía Server, ta cần gán thuộc tính **runat="server"** cho các điều khiển HTML đó. Những điều khiển HTML có thuộc tính runat="server" được gọi là HTML Server Control.

Lưu ý: Tất cả các điều khiển HTML Server Control đều phải nằm trong thẻ form với thuộc tính runat="server".



Hình 3.13. Các điều khiển HTML trên thanh công cụ

Để chuyển điều khiển HTML thành HTML Server, ta cần thêm vào thuộc tính runat="server" như sau:

```
<input type="..." id="..." runat="server" />
```

Ví dụ: Các điều khiển HTML: Textbox, Button.

```
<input type="text" id="text1" runat="server" />
```

```
<input type="button" id="button1" value="button" runat="server" />
```

Ví dụ : Làm việc với HTML Control tạo một file có tên vidu2-1.aspx như sau:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="vidu2-1.aspx.cs" Inherits="vidu2_1" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
```

```

Nhập Tên bạn:<br />
<input type="text" runat="server" id="txtTen" />
<input type="submit" runat="server" id="Submit1" value="Submit"
onserver_Click="submit" />
<br />
<p runat="server" id="p1"></p>
</div>
</form>
</body>
</html>

```

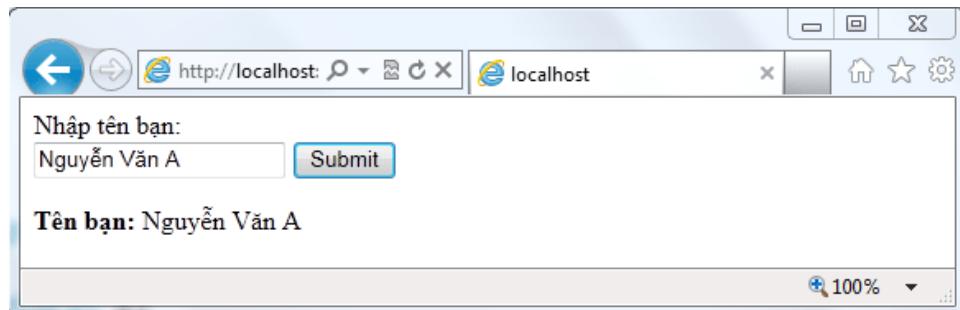
Code phía server như sau:

```

protected void submit(object sender, EventArgs e)
{
    p1.InnerHtml = "<b>Tên bạn:</b> " + txtTen.Value;
}

```

Giao diện của chúng như sau:



Hình 3.14. Kết quả khi thực thi ví dụ 2.1

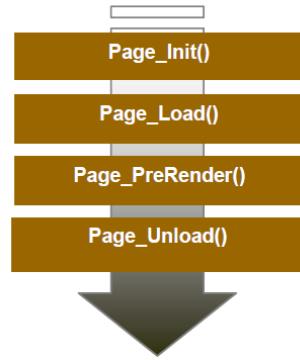
2.2. ASP.NET Server Control

ASP.NET Server Control là những điều khiển trình chủ có nhiều thuộc tính và tính năng mạnh mẽ hơn **HTML Server Control**.

ASP.NET Server Control tự động phát sinh các thẻ HTML theo từng loại trình duyệt.

2.2.1 ASP.NET Page

- a. Các sự kiện trong ASP.NET page



Hình 3.15: Sự kiện trong trang ASP.NET

Chuỗi sự kiện của đối tượng Page.

- **Init:** Sự kiện Page_Init xảy ra đầu tiên khi trang Web được yêu cầu
- **Load:** sự kiện này là nơi sẽ đặt phần lớn các xử lý, giá trị khởi động ban đầu cho trang Web. Sự kiện này luôn xảy ra mỗi khi trang Web được yêu cầu.
 - **PreRender:** Sự kiện này xảy ra khi trang Web chuẩn bị được trả về cho Client.
 - **Unload:** Sự kiện này đối lập với sự kiện **Init**, là sự kiện xảy ra sau cùng, xảy ra sau tất cả những sự kiện khác.

b. Các thuộc tính trong ASP.NET page.

- **IsPostBack:** đây là thuộc tính kiểu Boolean, giá trị của thuộc tính này cho biết trạng thái của trang web khi được load. Nếu lần load đầu tiên, giá trị của thuộc tính này được tính bằng false. Thuộc tính này thường được sử dụng trong sự kiện Page_Load để kiểm tra trạng thái của trang web.
- **SmartNavigation:** Đây là thuộc tính kiểu Boolean, trong trường hợp nội dung của trang web vượt quá kích thước của màn hình và bạn đang đọc ở phần giữa của trang web, khi được Reload lại, màn hình sẽ hiển thị phần đầu của trang web. Nếu giá trị của thuộc tính này là true, trình duyệt sẽ vẫn giữ nguyên vị trí mà bạn đang đọc sau khi Reload.

So sánh giữa ASP.NET Server Control và HTML Control

Tính năng	ASP.NET Server Control	HTML Control
Server Event	Kích hoạt được một số sự kiện cụ thể trên server.	Chỉ có thể kích hoạt các sự kiện mức trang trên server (post back)
Quản lý trạng	Dữ liệu nhập vào Control	Dữ liệu không được

thái	được lưu giữ lại sau mỗi Request.	lưu trữ lại, phải tự lưu và điền vào sử dụng script.
Tương thích	Tự động nhận diện trình duyệt và tạo hiển thị cho phù hợp.	Không tự động nhận diện trình duyệt.
Các thuộc tính	.Net Framework cung cấp một tập các thuộc tính cho mỗi Control, cho phép thay đổi phần hiển thị và hành vi thông qua mã lệnh.	Chỉ có các thuộc tính chuẩn của HTML

2.2.2. Các điều khiển cơ bản.

Bảng liệt kê các thuộc tính chung của ASP.NET Server Control.

Thuộc tính	Kiểu	Ý nghĩa
(ID)	String	Quy định tên của điều khiển, tên của điều khiển là duy nhất.
AccessKey	String	Quy định ký tự để di chuyển nhanh đến điều khiển – ký tự xử lý phím nóng.
BackColor	AttributeCollection	Tập hợp các điều khiển của HTML.
BackColor	Color	Quy định màu của điều khiển.
BorderColor	Color	Quy định màu nền đường viền của điều khiển
BorderStyle	BorderStyle	Quy định kiểu đường viền của điều khiển.
BorderWidth	Unit	Quy định độ rộng của đường viền.
CssClass	String	Quy định hình thức hiển thị của điều khiển thông qua CSS
Enabled	Boolean	Quy định điều khiển có được hiển thị hay không. Giá trị mặc định của điều khiển này là true

		– có được hiển thị.
Font	FontInfo	Quy định font hiển thị cho điều khiển.
ForeColor	Color	Quy định màu chữ hiển thị cho điều khiển.
Height	Unit	Quy định chiều cao của điều khiển.
Width	Unit	Quy định độ rộng của điều khiển.
Tooltip	String	Dòng chữ sẽ hiển thị khi di chuột vào điều khiển.

2.2.3. Label.

Label thường được dùng để hiển thị và trình bày nội dung trên trang Web. Nội dung được hiển thị trong Label được xác định thông qua thuộc tính Text có thể nhận và hiển thị nội dung với các thẻ HTML.

Ví dụ: lblThontin.Text = “Trường Cao đẳng Nghề công nghiệp Hà Nội”;

2.2.4. Hyperlink

Điều khiển này được dùng để tạo ra các liên kết siêu văn bản. Các thuộc tính của nó như sau:

- **ImageUrl:** Qui định hình hiển thị trên điều khiển.
- **Text:** Chuỗi văn bản sẽ được hiển thị trên điều khiển. Trong trường hợp cả 2 thuộc tính ImageURL và Text được thiết lập, thuộc tính ImageURL được ưu tiên, thuộc tính Text sẽ được hiển thị như Tooltip.
- **NavigateUrl:** Đường dẫn cần liên kết đến.
- **Target:** Xác định cửa sổ sẽ hiển thị cho mỗi liên kết
 - + **_blank:** Hiển thị trang liên kết ở một cửa sổ mới.
 - + **_self:** Hiển thị trang liên kết tại chính cửa sổ chứa liên kết đó.
 - + **_parent:** Hiển thị trang liên kết ở frame cha.

Ví dụ: Tạo một thẻ Hyperlink và đặt tên là hplCDN, ta viết đoạn mã trong code behide như sau:

```
hplCDN.Text="Trường Cao đẳng Nghề công nghiệp Hà Nội";
hplCDN.NavigateUrl="www.hnivc.edu.vn";
```

```
hlpCDN.ImageURL="Images/imageCDN.gif";
hlpCDN.Target = “_blank”;
```

2.2.5. Textbox

Textbox là điều khiển được dùng để nhập và hiển thị dữ liệu. Textbox thường được sử dụng nhiều trên các ứng dụng Windows Form. Các thuộc tính của nó như sau :

- **Text:** Nội dung chứa trong Textbox.
- **TextMode:** Quy định chức năng của textbox có các chức năng sau:
 - + **SingleLine:** Hiển thị và nhập liệu một dòng văn bản.
 - + **MultiLine:** Hiển thị và nhập liệu nhiều dòng văn bản.
 - + **Password:** Hiển thị dấu * thay cho các ký tự trong textbox.
- **Rows:** Trong trường hợp thuộc tính TextMode=MultiLine, thuộc tính rows sẽ quy định số dòng văn bản được hiển thị.
- **MaxLength:** Quy định số ký tự tối đa được nhập vào textbox.
- **Wrap:** Quy định việc hiển thị của văn bản có được phép tự động xuống dòng khi kích thước ngang của của điều khiển không đủ để hiển thị dòng nội dung văn bản. Giá trị mặc định của thuộc tính này là True - tự động xuống dòng.
- **AutoPostBack:** Thuộc tính này quy định điều khiển có được phép tự động PostBack về Server khi nội dung trong Textbox bị thay đổi hay không. Giá trị mặc định của thuộc tính này là False - không tự động Postback.

2.2.6. Image

Điều khiển này dùng để hiển thị hình ảnh trên trang web. Các thuộc tính của điều khiển này như sau:

- **ImageUrl:** Đường dẫn đến tập tin hình ảnh cần hiển thị.
- **AlternateText:** Chuỗi văn bản sẽ hiển thị khi tập tin được thiết lập trong thuộc tính ImageURL không tồn tại.
- **ImageAlign:** Vị trí hiển thị giữa hình và nội dung văn bản (NotSet, Left, Middle, TextTop, Right).

2.2.7. Button, ImageButton, LinkButton.

Các điều khiển Button, ImageButton, LinkButton mặc định đều là các nút Submit Button, mỗi khi được nhấn vào sẽ PostBack về Server. Khi chúng ta thiết lập giá trị thuộc tính CommandName cho các điều khiển này, chúng ta gọi tên chung cho các điều khiển này là Command Button.

Các thuộc tính chung của Button, ImageButton, LinkButton:

Thuộc tính	Ý nghĩa
Text	Chuỗi văn bản hiển thị trên điều khiển.
CommandName	Tên lệnh. Được sử dụng trong sự kiện Command.
CommandArgument	Thông tin bổ sung cho sự kiện Command.
CausesValidation	Trang web mặc định kiểm tra tính hợp lệ dữ liệu mỗi khi được PostBack. Các điều khiển Button, ImageButton, LinkButton luôn PostBack về Server mỗi khi được nhấn luôn kiểm tra tính hợp lệ dữ liệu trên trang web. Muốn trang Web bỏ qua việc kiểm tra dữ liệu khi được nhấn, gán trị cho thuộc tính này = False. Giá trị mặc định của thuộc tính này là True.

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

2.2.8. Listbox và DropDownList

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

❖ Các thuộc tính

- **AutoPostBack:** Thuộc tính này qui định điều khiển có được phép tự động PostBack về Server khi chỉ số của mục chọn bị thay đổi. Giá trị mặc định của thuộc tính này là False - không tự động Postback.
- **Items:** Đây là tập hợp chứa các mục chọn của điều khiển. Ta có thể thêm vào mục chọn vào thời điểm thiết kế thông qua cửa sổ ListItem Collection Editor, hoặc thông qua lệnh.
- **Rows:** Qui định chiều cao của ListBox theo số dòng hiển thị.
- **SelectionMode:** Thuộc tính này xác định cách thức chọn các mục trong ListBox. SelectionMode chỉ được phép thay đổi trong quá trình thiết kế, vào lúc thực thi chương trình, thuộc tính này chỉ đọc.
 - + **Single:** Chỉ được chọn một mục có trong danh sách (mặc định).
 - + **Multiple:** Cho phép chọn nhiều lựa chọn.

❖ *Xử lý mục chọn*

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

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

❖ *Tìm hiểu về tập hợp Items*

- **Add:** Thêm mục mới vào cuối danh sách, sử dụng phương thức Items.Add

Cú pháp : Items.Add(<string>)

Items.Add(<ListItem>)

- **Insert:** Thêm một mục mới vào danh sách tại một vị trí nào đó, sử dụng phương thức Items.Insert

Cú pháp: Items.Insert(<index>,<string>)

Items.Insert(<index>,<ListItem>)

- **Count:** Trả về số mục (Item) có trong danh sách

Cú pháp: Items.Count

- **Remove:** Xóa đối tượng Item ra khỏi danh sách.

Cú pháp: Items.Remove(<ListItem>)

Items.Remove(<string>)

- **Contains:** Kiểm tra xem một Item đã có trong tập hợp Items hay chưa, nếu có phương thức này sẽ trả về giá trị true, ngược lại trả về giá trị false

Cú pháp: Items.Contains(<ListItem>)

Trong trường hợp các đối tượng Item là kiểu string (chuỗi) ta truyền vào một string để xóa. Nếu có nhiều giá trị giống nhau trong danh sách, chỉ có mục chọn đầu tiên bị xóa. Trong trường hợp các đối tượng Item là đối tượng, ta truyền vào một biến tham chiếu đến đối tượng cần xóa.

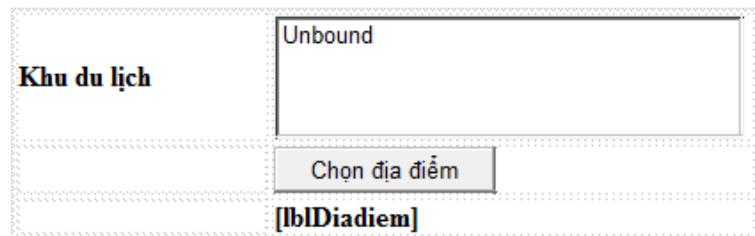
- **RemoveAt:** Xóa một Item tại vị trí index Item cần xóa

Cú pháp: Items.RemoveAt(<index>)

- **Clear:** Phương thức clear của tập hợp Item được dùng để xóa tất cả những Item có trong danh sách.

Cú pháp: Items.Clear()

Ví dụ: Điều khiển danh sách LstKhudl, SelectionMode=Multiple, Rows=4 có thiết kế như sau:



Hình 3.16: Cửa sổ Design

Xử lý sự kiện:

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

```
{
```

```
    if (!IsPostBack)
```

```
{
```

```
        LstKhudl.Items.Add("Suối tiên");
```

```
        LstKhudl.Items.Add("Nha trang");
```

```
        LstKhudl.Items.Add("Thác bạc");
```

```
        LstKhudl.Items.Add("Đầm sen");
```

```
}
```

```
}
```

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

```
{
```

```
    lblDiadiem.Text = "";
```

```
    for (int i = 0; i < LstKhudl.Items.Count; i++)
```

```
{
```

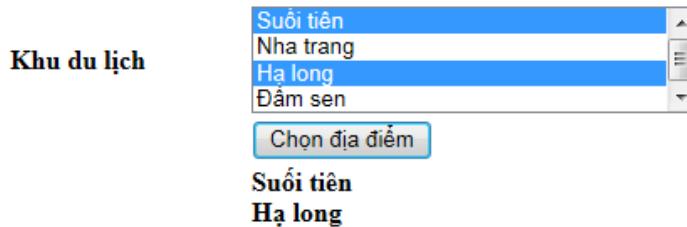
```
    if(LstKhudl.Items[i].Selected)
```

```
        lblDiadiem.Text += LstKhudl.Items[i].Text+"<br />";
```

```
}
```

```
}
```

Khi thi hành trang:



Hình 3.17.: Kết quả khi thi hành trang

2.2.9. Checkbox và Radiobutton

❖ Các thuộc tính:

- **Checked:** Cho biết trạng thái của mục chọn có được chọn hay không.
- **TextAlign:** Quy định vị trí hiển thị của điều khiển so với chuỗi văn bản.
- **AutoPostBack:** Thuộc tính này quy định điều khiển có được phép Postback về server khi các mục chọn của điều khiển bị thay đổi. Giá trị thay đổi của thuộc tính này là false – không tự động Postback.
- **GroupName (Radiobutton):** Tên nhóm - Thuộc tính này được sử dụng để nhóm các điều khiển RadioButton thành một nhóm.

Ví dụ:

Ngoại ngữ	<input checked="" type="checkbox"/> Tiếng anh <input type="checkbox"/> Tiếng trung <input type="checkbox"/> Tiếng nhật
Giới tính	<input checked="" type="radio"/> Nam <input type="radio"/> Nữ
Thu nhập	<input type="radio"/> dưới 3 triệu <input checked="" type="radio"/> 3 đến 5 triệu <input type="radio"/> trên 5 triệu

Hình 3.18.: Mẫu hình thiết kế

Trong ví dụ trên có hai nhóm Radiobutton: giới tính và thu nhập.

Danh sách các điều khiển:

Điều khiển	Loại	Thuộc tính	Giá trị
ckbAnh	Checkbox	Checked	True
ckbTrung	Checkbox		
ckbNhat	Checkbox		
rdbNam	RadioButton	Checked	True
		GroupName	Gioitinh
rdbNu	RadioButton		
		GroupName	Gioitinh
rdbThunhap1	RadioButton		
		GroupName	Thunhap
rdbThunhap2	RadioButton	Checked	True

		GroupName	Thunhap
rdlThunhap3	RadioButton		
		GroupName	Thunhap

2.2.10. CheckBoxList và RadioButtonList

Hai điều khiển này sử dụng để tạo ra một nhóm các Checkbox, Radio button. Do đây là biểu thức danh sách nên nó cũng có các thuộc tính Items chứa tập hợp các mục chọn như Listbox, DropDownList. Các thao tác trên tập hợp Items, xử lý mục chọn cũng tương tự như Listbox, DropDownList.

❖ Các thuộc tính:

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

Ví dụ: Viết chương trình như thiết kế dưới đây

Hình 3.19: Cửa sổ Design

Xử lý sự kiện:

```
protected void rdblThunhap_SelectedIndexChanged(object sender,
    EventArgs e)
{
    lblthunhap.Text = "Thu nhập của bạn: " +
    rdblThunhap.SelectedItem.Text;
}
```

Kết quả khi duyệt

Hình 3.20: Kết quả thực thi trang

3. Các đối tượng trong ASP.NET

Đối tượng là một nhóm các hàm và biến. Một số đối tượng đã được xây dựng sẵn và có thể sử dụng ngay mà không cần khởi tạo: Request, Response, Session, Application, Server. Một số đối tượng cần khởi tạo nếu muốn sử dụng Dictionary, Connection, Recordset...

3.1. Đối tượng Request

Request và Response là 2 đối tượng được dùng nhiều nhất trong lập trình ASP, dùng trao đổi dữ liệu giữa trình duyệt và server. Request cho phép lấy về các thông tin từ client. Khi browser gửi một yêu cầu trang web lên server ta gọi là 1 request

Chúng ta thường sử dụng các lệnh request sau:

❖ **Request.QueryString**

Cho phép server lấy về các giá trị được gửi từ người dùng qua URL hoặc form (method GET).

Ví dụ ở trang home.aspx chúng ta đặt một dòng liên kết sang trang gioithieu.aspx với thẻ sau:

```
<a href="/gioithieu.aspx?tacgia=Tran Van A">Nhấn vào đây để  
sang trang giới thiệu</a>
```

Biến "tacgia" có giá trị là "Tran Van A" được người dùng gửi tới server kèm theo URL. Người dùng có thể gõ thăng địa chỉ :

http://localhost/alias/gioithieu.aspx?tacgia=Tran Van A

trên thanh Address của trình duyệt. Server muốn nhận lại giá trị này thì dùng **request.QueryString** ở trang gioithieu.aspx

String str

```
str=Request.QueryString("tacgia") //lúc này tacgia có giá trị là  
"Tran Van A"
```

```
Response.Write("Tác giả của trang home.asp là: " + str);
```

Tương tự như vậy nếu người dùng gửi giá trị Tran Van A thông qua một biến trong form và chọn method GET

```
<form method="get" action ="gioithieu.asp">  
<input type="text" name="tacgia" value="Tran Van A" />  
<input type="submit" name="submit" value="Nhấn vào đây để sang  
trang giới thiệu" />  
</form>
```

❖ Request.Form

Cho phép server lấy về các giá trị được gửi từ người dùng qua form (method POST). Chẳng hạn file form.aspx:

```
<form method="POST" action ="xulyform.aspx">
<input type="text" name="User" />
<input type="submit" name="submit" value="Nhan vao day de sang
trang
gioi thieu" />
</form>
```

File xulyform.aspx làm nhiệm vụ xử lý thông tin từ Form này sẽ dùng câu lệnh request.form để nhận lại thông tin người dùng đã gõ vào:

```
string user;
user=Request.Form["User"];
Response.Write ("Tên người dùng là: "+user);
```

3.2. Response

Đối tượng Response dùng để gửi các đáp ứng của server cho client. Chúng ta thường dùng một số lệnh Response sau:

❖ Response.Write : Đưa thông tin ra màn hình trang web.

Ví dụ : Để đưa câu chào Hello ra màn hình ta dùng lệnh:

```
Response.Write ("Hello");
```

Để hiển thị thời gian trên server ra màn hình:

```
Response.Write (DateTime.Now);
```

Ở đây DateTime.Now là hàm lấy ngày giờ hệ thống trên server.

❖ Response.Redirect : Chuyển xử lý sang một trang Asp.Net khác.

Ví dụ trang xulyform.aspx sau khi kiểm tra form đăng nhập thấy người dùng không có quyền vào website thì nó sẽ chuyển cho file Error.aspx(file này hiển thị một thông báo lỗi user không có quyền truy cập).

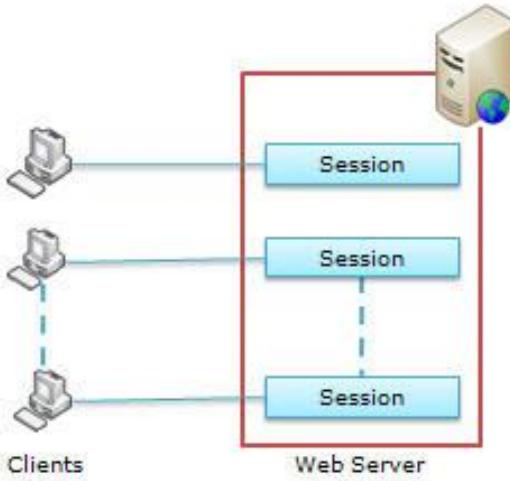
```
Response.Redirect("error.aspx" )
```

❖ Response.End : Ngừng xử lý các Script.

Dùng lệnh này khi muốn dừng xử lý ở một vị trí nào đó và bỏ qua các mã lệnh ASP.NET ở phía sau. Đây là cách rất hay dùng trong một số tình huống, chẳng hạn như debug lỗi.

3.3. Session.

Đối tượng Session là biến được tạo ra ở mỗi phiên làm việc của một người dùng, vậy là mỗi người đăng nhập vào sẽ có một session riêng và phiên làm việc của họ cũng có biến session riêng và được hủy khi phiên làm việc đó kết thúc và họ logout.



Hình 3.21 : Session trong ASP.Net

Session là một trong những biến quan trọng nhất trong ứng dụng web, nó là riêng biệt với các đối tượng người dùng tuy nhiên biến này được tạo ra sẽ sử dụng ở tất cả mọi trang trong website và lưu vết của người dùng đó đã làm những gì trên website mình.

Ví dụ: Web bán hàng khi khách hàng chọn hàng vào giỏ ở trang chủ, và sau đó khách tham quan các trang sản phẩm khác vẫn add được hàng vào giỏ hàng đó số lượng tăng lên... và giỏ đó hủy khi bấm hủy giỏ hay thanh toán.

Cũng như ở Cookies, Session cũng có thời gian nếu quá thời gian đó session sẽ tự hủy đó người ta gọi là thuộc tính time out.

Như đã nói Session là biến hoạt động trên Server nên tồn bộ nhớ, khi không có tác động từ client để duy trì sự tồn tại của biến Session thì Session vẫn tồn tại nên rất tồn cùng như server do đó tính tự hủy khi hết time sẽ hoạt động nhờ vào một phương thức người ta gọi đó là phương thức Abandon.

Sử dụng biến toàn cục Session trong ASP.net

- Cách khai báo Session : `Session["tên biến"] = giá trị;`
- Lấy giá trị của biến Session: `<biến> = Session["tên biến"];`

Ví dụ: bạn tạo trang đăng nhập đơn giản sau khi submit các thông tin đăng nhập sẽ gửi các thông tin đó qua trang khác và hiển thị lên.

Giả sử bạn tạo một trang đăng nhập: dangnhap.aspx với như sau:

```
Tên: <asp:TextBox ID="txtTen" runat="server"></asp:TextBox>
Pass: <asp:TextBox ID="txtPass" runat="server" TextMode="Password">
      </asp:TextBox>
<asp:Button ID="Submit" runat="server" onClick="Submit_Click"
Text="Gửi" />
```

Trong nút gửi bạn viết đoạn code sau:

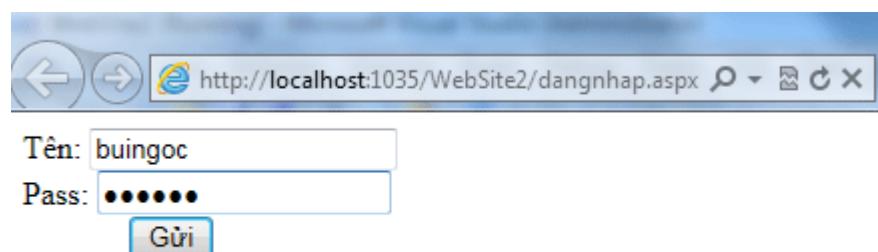
```
protected void Submit_Click(object sender, EventArgs e)
{
    Session["ten"] = txtTen.Text;
    Session["pass"] = txtPass.Text;
    Response.Redirect("Nhan.aspx");
}
```

Tiếp theo ta khai báo hai biến Session một là lấy tên, hai là lấy Pass và tiến hành gửi qua trang Nhận: Nhan.aspx. Ở trang Nhan.aspx bạn lấy thông tin như sau:

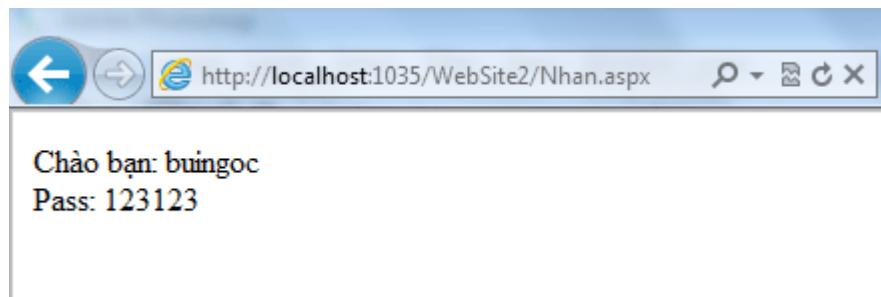
```
protected void Page_Load(object sender, EventArgs e)
```

```
{  
    string ten = Session["ten"].ToString();  
    string pass = Session["pass"].ToString();  
    Response.Write("Chào bạn: " + ten + "<br />Pass: " + pass);  
}
```

Sau đó bạn chạy trang Đăng nhập lên nhập thông tin và gửi sẽ nhận được thông tin ở trang nhận như sau:



Hình 3.22: Trang dangnhap.aspx

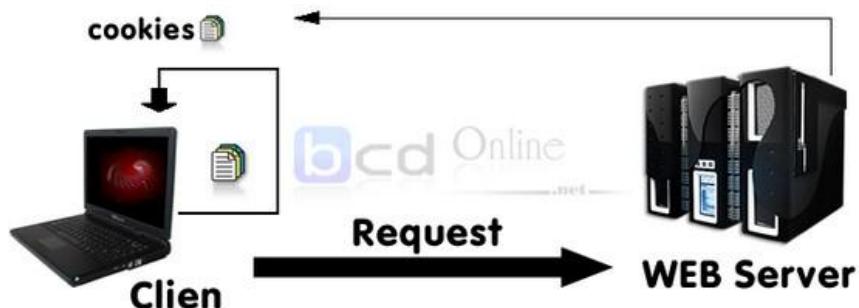


Hình 3.23: Trang Nhan.aspx

3.4. Cookie.

Bạn đã từng sử dụng diễn đàn, đăng ký thành viên và đăng nhập. Sau đó, mỗi lần sau vào trang diễn đàn đó tự nhiên bạn không cần đăng nhập nữa mà vẫn xuất ra thông báo chào mình. Đó chính là một ứng dụng hay của đối tượng cookies.

Đây là một tiện ích ở máy client, nó được hiểu như một tập tin nhỏ với các thông tin bạn đã khai báo lúc đầu, khi bạn lướt web có yêu cầu gửi đến Server thì mặc nhiên các thông tin đó cũng gửi lên server đỡ mất công đăng nhập nữa. Đối tượng cookies này tồn tại trên trình duyệt ở một thời gian nhất định tùy theo cài đặt, sau thời gian đó thì bạn phải khởi tạo lại biến đó mới sử dụng được nữa.



Hình 3.24: Đối tượng cookies trong ASP.Net

❖ Làm việc với cookies trong ASP.Net

Thêm cookies theo cú pháp sau:

Response.Cookies.Add(<HttpCookie>);

Ví dụ:

```

HttpCookie ck = new HttpCookie("TenDangNhap");
ck.Value = txtTenDangNhap.Text;
ck.Expires = DateTime.Now.AddDays(15);
Response.Cookies.Add(ck);

```

Cách khai báo trên có ý nghĩa là: tạo ra 1 Cookies có tên là **TenDangNhap** lấy giá trị từ textbox **txtTenDangNhap** lưu trữ trên máy người dùng 15 ngày kể từ ngày tạo ra.

Lấy giá trị từ Cookie

```
HttpCookie ck = Request.Cookies("TenDangNhap");  
string s = ck.Value;
```

❖ Đối tượng Application

Application đại diện cho toàn bộ ứng dụng, bao gồm tất cả các trang web trong website. Để lưu trữ những thông tin có tác dụng trong toàn ứng dụng, tức là có giá trị trong tất cả các trang asp.net và tất cả các phiên, người ta dùng đối tượng Application.

Điểm khác của biến application so với biến session là session chỉ có tác dụng đối với mỗi phiên, còn biến application có tác dụng với mọi phiên.

Ví dụ: Để đếm xem có bao nhiêu người truy cập vào Website, chúng ta có thể dùng một biến Application. Mỗi khi một người dùng mới truy cập vào website ta tăng biến này lên 1 đơn vị để chỉ rằng đã có thêm 1 người truy cập.

```
Application["online"] = Application["online"] + 1;
```

Trang home.aspx muốn hiển thị số người truy cập chỉ cần in giá trị của biến này:

```
Response.Write ("Số người đã truy cập vào website  
là: "+application["online"]);
```

Với 2 phiên khác nhau thì giá trị Application["online"] là như nhau. Khi A và B khi truy cập vào trang home.aspx đều thấy: "Số người đã truy cập vào website là 3" (trong trường hợp application["online"] =3).

Việc khởi tạo và kết thúc 1 biến application có thể viết trong các hàm sự kiện Application_Start và Application_End được định nghĩa trong file global.asax.

❖ Khóa Application

Do biến application có thể được dùng chung bởi nhiều phiên nên sẽ có trường hợp xảy ra xung đột khi có 2 phiên cùng thay đổi giá trị một biến application.

Để ngăn chặn điều này chúng ta có thể dùng phương thức Application.Lock để khóa biến application trước khi thay đổi nó. Sau khi

sử dụng xong biến này có thể giải phóng khóa bằng phương thức Application.Unlock.

❖ File Global.asax

File này là file tùy chọn chứa các khai báo đối tượng, biến có phạm vi toàn ứng dụng. Mã lệnh viết dưới dạng Script. Mỗi ứng dụng chỉ được phép có nhiều nhất 1 file Global.asax, nằm ở thư mục gốc của ứng dụng. Người ta thường dùng global.asax trong trường hợp muốn có những xử lý khi một session bắt đầu hay kết thúc, một application bắt đầu hay kết thúc, thông qua các hàm sự kiện:

- **Application_Start**: hàm sự kiện này xảy ra khi ứng dụng asp bắt đầu hoạt động, tức là khi người dùng đầu tiên truy cập tới trang web đầu tiên khi ứng dụng hoạt động.
- **Session_Start**: hàm sự kiện này xảy ra mỗi khi có một người dùng mới truy cập vào ứng dụng (bắt đầu 1 session).
- **Session_End**: hàm sự kiện này xảy ra mỗi khi 1 người dùng kết thúc session của họ.
- **Application_End**: hàm sự kiện này xảy ra khi ứng dụng dừng.

```
void Application_Start(object sender, EventArgs e)
{
    // Đoạn mã khởi động ứng dụng
}

void Application_End(object sender, EventArgs e)
{
    // Đoạn mã ngừng ứng dụng
}

void Application_Error(object sender, EventArgs e)
{
    // Mã chạy khi xảy ra một lỗi chưa được xử lý
}

void Session_Start(object sender, EventArgs e)
{
    // Mã chạy khi một Session mới được bắt đầu
}

void Session_End(object sender, EventArgs e)
{
```

```
// Mã chạy khi một Session kết thúc.
```

```
}
```

Lưu ý: Sự kiện Session_End thực hiện khi chế độ sessionstate được thiết lập để trong file Web.config. Nếu chế độ Session được thiết lập để ở chế độ StateServer hoặc SQLServer, sự kiện này không được thực thi.

4. Các đối tượng ADO.NET

Tùy theo loại Connection mà phương thức sẽ trả về một OleDbTransaction hay SqlTransaction. Giao tác (Transaction) là quá trình cập nhật tạm thời các sửa đổi trên nguồn dữ liệu và chỉ cập nhật thực sự khi phương thức Commit của giao tác được gọi hoặc phục hồi lại những gì đã thực hiện trên dữ liệu qua phương thức Rollback.

Những đối tượng trong ADO.NET đều bắt nguồn từ tên miền:

System.Data

System.Data.OleDb;

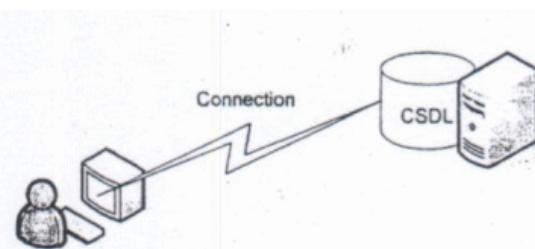
System.Data.SqlClient;

Tên miền System.Data.OleDb chứa các lớp được thiết kế để làm việc với bất kỳ nguồn dữ liệu nào, ngược lại System.Data.SqlClient gồm các lớp được thiết kế tối ưu cho CSDL của SQL Server 2000 trở về sau.

Để sử dụng chức năng của các đối tượng này, chúng ta phải tham chiếu đến tên miền System.Data.

4.1. Connection.

Vai trò của Connection trong ADO.NET là tạo kết nối giữa ứng dụng và nguồn dữ liệu.



Hình 3.25: Vai trò của Connection

Trước khi sử dụng đối tượng Connection, ta cần khai báo các không gian tên gọi là Data Provider. Data Provider được sử dụng để kết nối đến một database, thực thi lệnh và lấy kết quả.

❖ Data Provider.

Các Data Provider có sẵn trong ADO.NET là :

- System.Data.OleDb.
- System.Data.SqlClient.

Ứng với mỗi tên miền, chúng ta có một Connection tương ứng:

- System.Data.OleDb.OleDbConnection.
- System.Data.SqlClient.SqlConnection.

Chúng ta có thể sử dụng System.Data.OleDb.OleDbConnection cho mọi nguồn dữ liệu. Nhưng để tận dụng các đặc điểm của những Data Provider khác nhau, chúng ta có thể chỉ rõ Data Provider muốn sử dụng. Ngoài SQL Server Provider, ADO.NET cũng hỗ trợ một số Data Provider khác như :

- System.Data.OracleClient qua tập tin.
- System.Data.OracleClient.dll cho Oracle.
- Microsoft.Data.Odbc qua tập tin Microsoft.Data.Odbc.dll cho ODBC.

- Microsoft.Data.SqlXml qua tập tin Microsoft.Data.SqlXml.dll cho Xml trên SQL Server.

❖ **Connection String.**

Trước khi thực hiện kết nối cần khai báo các thông tin cần thiết cho Connection thông qua thuộc tính ConnectionString. Cách khai báo thay đổi tùy thuộc Data Provider. Với OLEDB Provider có thể gồm các thành phần :

- Provider : Khai báo Data Provider
- Data Source : Nguồn dữ liệu
- User ID : Tên người dùng
- Password : Mật khẩu

Ví dụ : Tạo Connection đến CSDL Access

```
string strConn;
strConn = "Provider=" +
"Microsoft.Jet.OleDb.4.0;" +
"DataSource=" + <đường dẫn>;
OleDbConnection conn = new OleDbConnection(strConn);
```

Tạo Connection đến CSDL MS SQL Server

```
string strConn;
strConn = "DataSource=Tên server;" +
"Initial Catalog = Tên database;" +
```

```

"user id=Tên truy cập;" +
"password=mật khẩu";
SqlConnection conn = new SqlConnection(strConn);

```

Khi sử dụng SqlConnection, cách khai báo khác hơn với các thành phần chính như sau :

- **Data Source** : Tên máy nơi cài đặt SQL Server có nguồn dữ liệu muốn kết nối.
- **Initial Catalog** : Tên database muốn kết nối.
- **Integrated Security** : True(SSPI)/False chỉ định dùng cơ chế bảo mật của Windows Login (True) hay cơ chế bảo mật của SQL Server (False).
- **User ID** : Tên người dùng (phải khai báo khi Integrated Security = False).
- **Password** : Mật khẩu (phải khai báo khi Integrated Security = False).

Việc sử dụng cơ chế bảo mật nào tùy thuộc SQL Server quy định.

Ví dụ :

```

string strConnection = "Data Source=<Địa chỉ TCP/IP hay tên máy>" +
    + "Initial Catalog=<Tên database>; user id=<tên người dùng>;" +
    + "pwd=<Mật khẩu>;";
SqlConnection cnn = new SqlConnection(strConnection);

```

Hoặc

```

string strConnection = "Data Source=<Địa chỉ TCP/IP hay tên máy>" +
    + "Initial Catalog=<Tên database>;" +
    + "Integrated Security=true;;";
SqlConnection cnn = new SqlConnection(strConnection);

```

❖ Các thuộc tính khác của Connection :

- **Database** : (Chỉ đọc) Tương ứng với Initial Catalog (SQL) hay tên database muốn làm việc (OLEDB).
- **DataSource** : (Chỉ đọc) Tương ứng với Data Source (OLEDB và SQL).
- **Provider** : (Chỉ đọc) Tương ứng với Provider (OLEDB).
- **State** : Tình trạng kết nối của Connection với các giá trị.
 - + *Broken* : Kết nối với nguồn dữ liệu đã bị ngắt. Tình trạng này

chỉ xảy ra sau khi đã kết nối.

- + *Closed* : Kết nối đã đóng.
- + *Connecting* : Đang kết nối với nguồn dữ liệu.
- + *Executing* : Kết nối đang thực hiện một lệnh.
- + *Fetching* : Kết nối đang truy xuất dữ liệu.
- + *Open* : Kết nối đang mở.

❖ Các phương thức trên Connection.

- **BeginTransaction**: Bắt đầu một giao tác dữ liệu và trả về một đối tượng OleDbTransaction hoặc SqlTransaction tương ứng.

- **ChangeDatabase**: Thay đổi database làm việc.

```
cnn.ChangeDatabase("tên database");
```

- **Close** : Dóng kết nối với nguồn dữ liệu. Sử dụng phương thức này để đóng Connection đang mở.

- **CreateCommand**: Tạo và trả về một Command dựa vào Connection hiện hành.

```
cnn.CreateCommand();
```

- **Dispose** : Xóa mọi tài nguyên liên quan đến Connection trên vùng nhớ.

- **Open** : Thực hiện kết nối với các thông tin đã khai báo trong ConnectionString.

❖ Transaction

Tùy theo loại Connection mà phương thức sẽ trả về một OleDbTransaction hay SqlTransaction. Giao tác (Transaction) là quá trình cập nhật tạm thời các sửa đổi trên nguồn dữ liệu và chỉ cập nhật thực sự khi phương thức Commit của giao tác được gọi hoặc phục hồi lại những gì đã thực hiện trên dữ liệu qua phương thức Rollback.

Cách tạo Transaction

Transaction được tạo ra bằng cách gọi phương thức BeginTransaction của connection.

Với OleDbConnection

```
OleDbTransaction <transaction> = <connection>.BeginTransaction();  
OleDbTransaction <transaction> =  
<connection>.BeginTransaction(<IsolationLevel>);
```

Với SqlConnection, ngoài hai cách như trên còn có hai cách sau:

```

SqlTransaction <connection> = <connection>.BeginTransaction(<tên
giao tác>);

SqlTransaction <connection> = <connection>.BeginTransaction(<tên
giao tác>, <IsolationLevel>);

```

<IsolationLevel> : định tính chất của Transaction, xem thuộc tính.

Các thuộc tính:

- **Connection :** Trả về Connection kết hợp với giao tác.
- **IsolationLevel :** Cho biết giá trị của IsolationLevel (mức độ cô lập).

.Net data provider sử dụng giá trị này để thực hiện giao tác. IsolationLevel có hiệu lực cho đến khi được thay đổi giá trị và có thể thay đổi tại bất kỳ thời điểm nào. Giá trị mới sẽ được dùng trong thời gian thực hiện giao tác. Nếu việc thay đổi xảy ra trong khi giao tác đang diễn biến, giá trị có hiệu lực cho các lệnh còn lại trong giao tác. IsolationLevel có các trị sau : Chaos, ReadCommitted, ReadUnCommitted, RepeatableRead, Serializable và Unspecified.

Các phương thức :

- **Begin :** Bắt đầu một giao tác lồng (nested transaction).
- **Commit :** Cập nhật giao tác đang xảy ra.
- **Rollback :** Phục hồi những thay đổi giao tác đã thực hiện.

Sau đây là một ví dụ tạo Connection kết nối SQL Server 2008 với cơ sở dữ liệu ShopGifts bằng lệnh :

```

string strConnection = "Data Source=ASUS-PC;" +
    + "Initial Catalog=ShopGifts;" +
    + "user id=sa; pwd=123123;" +
    SqlConnection cnn = new SqlConnection(strConnection);

```

4.2. Command

Sau khi tạo kết nối với nguồn dữ liệu, mọi thao tác trên nguồn dữ liệu đó đều được thực hiện thông qua Command. Tùy theo loại Connection, đối tượng Command thuộc tên miền như sau :

System.Data.OleDbCommand

System.Data.SqlClient

❖ Tạo Command:

Chúng ta có thể tạo đối tượng Command qua các cách sau thông qua một đối tượng Connection :

```
SqlCommand <biến command> = new SqlCommand();
<biến command>.Connection = <biến connection>;
<biến command>.CommandText = <lệnh SQL>;
```

Hoặc

```
SqlCommand <biến command> = new SqlCommand(<Lệnh SQL>);
<biến command>.Connection = <biến connection>;
```

Hoặc

```
SqlCommand <biến command> = new SqlCommand(<Lệnh
SQL>,<biến connection>);
```

Hoặc

```
SqlCommand <biến command> = new SqlCommand(<biến
connection>);
<biến command>.CommandText = <lệnh SQL>;
```

❖ Các thuộc tính của Command :

- **CommandText** : lệnh SQL hoặc tên stored procedure muốn thực hiện trên nguồn dữ liệu (đọc ghi).
- **CommandType** : Giá trị cho biết nội dung CommandText là gì với các giá trị sau:
 - + *Text* : (mặc định) một câu lệnh SQL.
 - + *StoredProcedure* : tên một thủ tục nội.
- **Connection** : Đối tượng Connection sử dụng cho Command.
- **Parameters** : Tập hợp các tham số dùng trong Command.
- **Transaction** : Giao tác mà trên đó Command đang thực hiện.

Parameter

Lệnh SQL trong CommandText có thể sử dụng các dấu hỏi (?) thay thế cho trị chưa xác định và khi thực hiện sẽ dùng đối tượng Parameter để truyền giá trị vào chỗ các dấu hỏi. Tùy theo Command, Parameter sẽ khai báo từ lớp SqlParameter hay SqlParameter. Cú pháp khai báo :

```
SqlParameter <biến parameter> = new SqlParameter();
SqlParameter <biến parameter> = new SqlParameter(<tên>);
SqlParameter <biến parameter> = new SqlParameter(<tên>, <giá trị>);
```

Các thuộc tính của Parameter :

- **Direction:** Giá trị cho biết loại tham số với các trị sau : (đọc ghi).
 - + *Input* (*mặc định*) : loại tham số đầu vào.
 - + *InputOutput* : loại tham số vào và ra.
 - + *Output* : loại tham số đầu ra.
 - + *ReturnValue* : loại tham số nhận giá trị về của một thủ tục nội, một hàm, hay một hàm do người dùng định nghĩa.
- **OleDbType, SqlDbType :** Kiểu dữ liệu OleDb hoặc SqlDbType của tham số (đọc ghi).
- **ParameterName :** Tên tham số (đọc ghi).
- **Value :** Giá trị của tham số (đọc ghi).

Để đưa một tham số cho Command, chúng ta có thể sử dụng một trong các cách sau :

Dùng phương thức CreateParameter của Command và Add của tập hợp Parameters

Ví dụ : Khi sử dụng OleDbCommand

```
OleDbParameter tham_so = new OleDbParameter();
tham_so.Value = 1;
cmm.Parameters.Add(tham_so);
```

Khi sử dụng SqlCommand

```
SqlParameter tham_so = new SqlParameter();
Tham_so.ParameterName= "@<Tên trường>"
tham_so.Value = 1;
<biến command>.Parameters.Add(tham_so);
```

Ví dụ 4-1. Các thao tác với cơ sở dữ liệu

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
public partial class vidu4_1 : System.Web.UI.Page
{
```

```

protected void Page_Load(object sender, EventArgs e)
{
    string strconn = "Data Source=ASUS-PC; Initial Catalog=Tintuc;
user id=sa;password=123123;";
    //Đối tượng kết nối cơ sở dữ liệu
    SqlConnection sqlconn = new SqlConnection(strconn);
    //Command điều khiển truy vấn sql
    SqlCommand sqlcom = sqlconn.CreateCommand();
    sqlcom.CommandText = "select Tieude from tblBaiviet where
ID=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);
}
}

```

Từ ví dụ trên ta có thể đưa ra các bước cơ bản thực hiện với database như sau:

- *Bước 1: Tạo 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.*

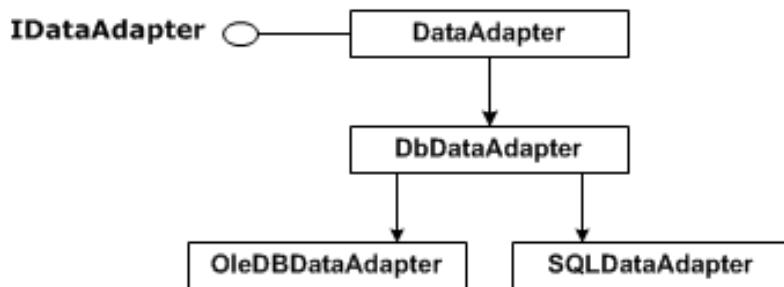
4.3. DataReader

DataReader hỗ trợ về mặt tốc độ khi truy cập dữ liệu, tuy nhiên nó sử dụng cursor ở phía Server và kết nối với Server trong suốt quá trình đọc dữ liệu. Việc này khác với cơ chế sử dụng Disconnect data của ADO.NET và giảm khả năng mở rộng của ứng dụng. Vì vậy, khi đã có một đối tượng DataReader, cần phải tiến hành việc đọc dữ liệu càng nhanh càng tốt.

Sử dụng Connection kết nối với Command và DataReader là một cách được coi là "low level" của ADO.NET để thao tác với dữ liệu. ADO.NET cung cấp DataReader là một kết hợp của Connection với các Command giúp kết nối DataSet với database trên Server.

4.4. DataAdapter

DataAdapter được ADO.NET cung cấp như một công cụ giúp kết nối dữ liệu của một table trong DataSet với database trên Server.



Hình 3.26. DataAdapter

Hình trên mô tả mô hình phân cấp các class của DataAdapter, cả SqlDataAdapter và OleDbDataAdapter đều thừa kế từ class DbDataAdapter, DbDataAdapter lại thừa kế từ abstract class DataAdapter, đến lượt abstract class này lại implement IDataAdapter interface để hỗ trợ hai phương thức Fill và Update.

Để lấy dữ liệu cho một table trong DataSet, sử dụng phương thức Fill(), để cập nhật dữ liệu cho DataSet, sử dụng phương thức Update().

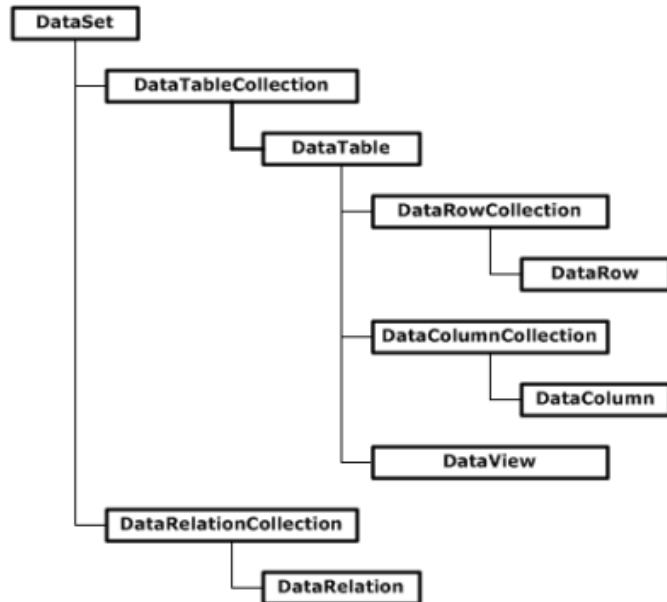
Để đọc dữ liệu, DataAdapter dùng SELECT query thông qua thuộc tính SelectCommand. Để cập nhật dữ liệu, DataAdapter dùng các câu truy vấn UPDATE, INSERT, DELETE thông qua các thuộc tính UpdateCommand, InsertCommand và DeleteCommand.

Ngoài Fill và Update, DataAdapter còn kế thừa thuộc tính TableMappings giúp cung cấp cho người dùng các tên table và tên column dễ đọc hơn các tên thực được sử dụng trong Database.

Trong các đối tượng Command của DataAdapter, chỉ có đối tượng SelectCommand là bắt buộc, các đối tượng còn lại đều có thể tự động được phát sinh. Các bước thường thấy khi sử dụng DataAdapter như sau:

- Tạo một đối tượng DataAdapter.
- Tạo Select query cho đối tượng SelectCommand.
- TạoConnectionString cho đối tượng Connection của DataAdapter.
- (Có hoặc không) Tạo các đối tượng Command cập nhật dữ liệu.

- Gọi phương thức Fill() để lấy dữ liệu cho một DataSet.
- Gọi phương thức Update() của DataAdapter để cập nhật dữ liệu từ DataSet.



Hình 3.27. Đối tượng thuộc Managed Provider Components

4.5. Đối tượng Dataset và DataTable.

Là thành phần chính của kiến trúc không kết nối cơ sở dữ liệu, được dùng để nắm giữ dữ liệu của mọi cơ sở dữ liệu và cho phép thay đổi dữ liệu bên trong đối tượng này để sau đó cập nhật trở lại cơ sở dữ liệu nguồn bằng phương thức Update của đối tượng DataAdapter.

Khởi tạo

```
DataSet dataset = new DataSet();
DataSet dataset = new DataSet("Mydataset");
```

Thuộc tính Tables, dataset được dùng để chứa danh sách các đối tượng DataTable.

Ví dụ 5-2: Sử dụng dataset để lấy dữ liệu.

Trong giao diện của trang web ta đưa control Gridview vào trong trang. Và sau đó viết đoạn code sau trong trang vidu5-2.aspx.cs

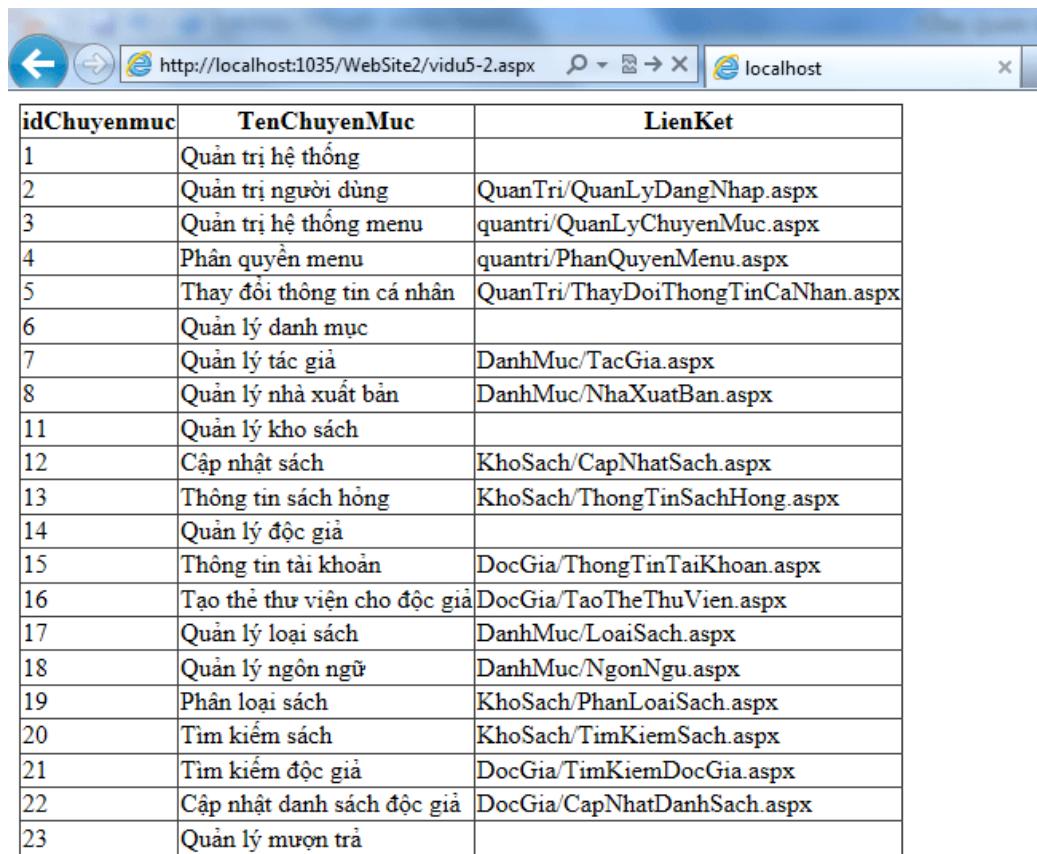
```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
```

```

using System.Data.SqlClient;
public partial class vidu4_2 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        string strconn = "Data Source=ASUS-PC; Initial
Catalog=QuanLyThuVien; user id=sa;password=123123;";
        //Đối tượng kết nối cơ sở dữ liệu
        SqlConnection sqlconn = new SqlConnection(strconn);
        //Command điều khiển truy vấn sql
        SqlCommand sqlcom = sqlconn.CreateCommand();
        sqlcom.CommandText = "select idChuyenmuc, TenChuyenMuc,
LienKet from TB_Chuyenmuc";
        //Mở kết nối dữ liệu
        sqlconn.Open();
        //Thực thi câu lệnh SQL
        sqlcom.ExecuteNonQuery();
        SqlDataAdapter da = new SqlDataAdapter(sqlcom);
        DataSet ds = new DataSet("ChuyenMuc");
        da.Fill(ds);
        GridView1.DataSource = ds;
        GridView1.DataBind();
        //Đóng kết nối
        sqlconn.Close();
    }
}

```

Khi chạy trang vidu 5-2 ta được kết quả sau:



The screenshot shows a Microsoft Internet Explorer browser window with the URL <http://localhost:1035/WebSite2/vidu5-2.aspx>. The page displays a grid view of data from a database table. The table has three columns: idChuyenmuc, TenChuyenMuc, and LienKet. The data is as follows:

idChuyenmuc	TenChuyenMuc	LienKet
1	Quản trị hệ thống	
2	Quản trị người dùng	QuanTri/QuanLyDangNhap.aspx
3	Quản trị hệ thống menu	quantri/QuanLyChuyenMuc.aspx
4	Phân quyền menu	quantri/PhanQuyenMenu.aspx
5	Thay đổi thông tin cá nhân	QuanTri/ThayDoiThongTinCaNhan.aspx
6	Quản lý danh mục	
7	Quản lý tác giả	DanhMuc/TacGia.aspx
8	Quản lý nhà xuất bản	DanhMuc/NhaXuatBan.aspx
11	Quản lý kho sách	
12	Cập nhật sách	KhoSach/CapNhatSach.aspx
13	Thông tin sách hỏng	KhoSach/ThongTinSachHong.aspx
14	Quản lý độc giả	
15	Thông tin tài khoản	DocGia/ThongTinTaiKhoan.aspx
16	Tạo thẻ thư viện cho độc giả	DocGia/TaoTheThuVien.aspx
17	Quản lý loại sách	DanhMuc/LoaiSach.aspx
18	Quản lý ngôn ngữ	DanhMuc/NgonNgu.aspx
19	Phân loại sách	KhoSach/PhanLoaiSach.aspx
20	Tìm kiếm sách	KhoSach/TimKiemSach.aspx
21	Tìm kiếm độc giả	DocGia/TimKiemDocGia.aspx
22	Cập nhật danh sách độc giả	DocGia/CapNhatDanhSach.aspx
23	Quản lý mượn trả	

Hình 3.28. Kết quả khi thực thi trang vidu5-2.aspx

Ví dụ 5-3: Sử dụng DataTable để lấy dữ liệu.

Trong giao diện của trang web ta đưa control Gridview vào trong trang. Và sau đó viết đoạn code sau trong trang vidu5-3.aspx.cs

using System;

using System.Collections.Generic;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

public partial class vidu4_3 : System.Web.UI.Page

{

 protected void Page_Load(object sender, EventArgs e)

 {

 string strconn = "Data Source=ASUS-PC; Initial Catalog=QuanLyThuVien; user id=sa;password=123123;";

 //Đối tượng kết nối cơ sở dữ liệu

 SqlConnection sqlconn = new SqlConnection(strconn);

```

//Command điều khiển truy vấn sql
SqlCommand sqlcom = sqlconn.CreateCommand();
sqlcom.CommandText = "select idChuyenmuc, TenChuyenMuc,
LienKet from TB_Chuyenmuc";
//Mở kết nối dữ liệu
sqlconn.Open();
//Thực thi câu lệnh SQL
sqlcom.ExecuteNonQuery();
SqlDataAdapter da = new SqlDataAdapter(sqlcom);
DataTable dt = new DataTable();
da.Fill(dt);
GridView1.DataSource = dt;
GridView1.DataBind();
//Đóng kết nối
sqlconn.Close();
}
}

```

Khi chạy ví dụ 5-3 ta được kết quả tương tự như ví dụ 5-2.

Ví dụ 5-4: Thực hiện đọc nội dung của bảng [**Product**] trong cơ sở dữ liệu **AdventureWorks** và đưa vào một DropDownList. Khi chọn một giá trị trong DropDownList, nội dung mẫu tin lựa chọn sẽ được hiển thị trên trang.

Trong ví dụ này ta thực hiện việc khai báo cấu hình ConnectionString trong **Web.config** : không cần phải viết code ở các trang mã lệnh.

```

<configuration>
  <appSettings/>
  <connectionStrings>
    <add name="AdventureWorks" connectionString="Server=ASUS-
PC; Database=AdventureWorks; user id=sa; pwd=123123;" />
  </connectionStrings>
</configuration>

```

Nội dung thiết kế trang vidu5-4.aspx

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="vidu5-4.aspx.cs" Inherits="vidu5_4" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0

```

```

Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Product Name</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:label id="Label1" runat="server" Width="120px"
Height="20px">Select
                Author:</asp:label>&nbsp;
                <asp:dropdownlist id="lstProduct" runat="server" Width="256px"
Height="22px"
                    AutoPostBack="True"
                    onselectedindexchanged="lstProduct_SelectedIndexChanged">
                    </asp:dropdownlist>
                    <br /><br /><br />
                    <asp:label id="lblResults" runat="server" Width="384px"
Height="168px">
                    </asp:label>
                </div>
            </form>
        </body>
    </html>

```

Phân code trong trang code behide vidu5-4.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Configuration;
using System.Data.SqlClient;
using System.Text;

```

```

public partial class vidu5_4 : System.Web.UI.Page
{
    // Lấy chuỗi kết nối từ trang Web.config
    private string connectionString =
        ConfigurationManager.ConnectionStrings["AdventureWorks"].ConnectionString;
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!this.IsPostBack)
        {
            FillProductList();
        }
    }
    private void FillProductList()
    {
        lstProduct.Items.Clear();
        string selectSQL = "SELECT Top 1000 [ProductID],[Name]
FROM [Production].[Product]";
        // Định nghĩa đối tượng ADO.NET.
        SqlConnection con = new SqlConnection(connectionString);
        SqlCommand cmd = new SqlCommand(selectSQL, con);
        SqlDataReader reader;
        // Mở và đọc thông tin dữ liệu.
        try
        {
            con.Open();
            reader = cmd.ExecuteReader();
            // Đọc tất cả dữ liệu trong đối tượng reader
            // Đưa dữ liệu vào DropDownList, thông qua Value và Text.
            while (reader.Read())
            {
                ListItem newItem = new ListItem();
                newItem.Text = reader["Name"].ToString();
                newItem.Value = reader["ProductID"].ToString();
            }
        }
    }
}

```

```

        lstProduct.Items.Add(newItem);
    }
    reader.Close();
}
catch (Exception err)
{
    lblResults.Text = "Lỗi không thể liệt kê danh sách. ";
    lblResults.Text += err.Message;
}
finally
{
    con.Close();
}
}

protected void lstProduct_SelectedIndexChanged(object sender,
EventArgs e)
{
    string selectSQL;
    selectSQL = "SELECT * FROM [Production].[Product] ";
    selectSQL += "WHERE [ProductID] = " +
lstProduct.SelectedItem.Value + "";
    // Định nghĩa đối tượng ADO.NET.
    SqlConnection con = new SqlConnection(connectionString);
    SqlCommand cmd = new SqlCommand(selectSQL, con);
    SqlDataReader reader;
    // Mở và đọc thông tin dữ liệu.
    try
    {
        con.Open();
        reader = cmd.ExecuteReader();
        reader.Read();
        StringBuilder sb = new StringBuilder();
        sb.Append("<b>");
        sb.Append("ProductName: ");

```

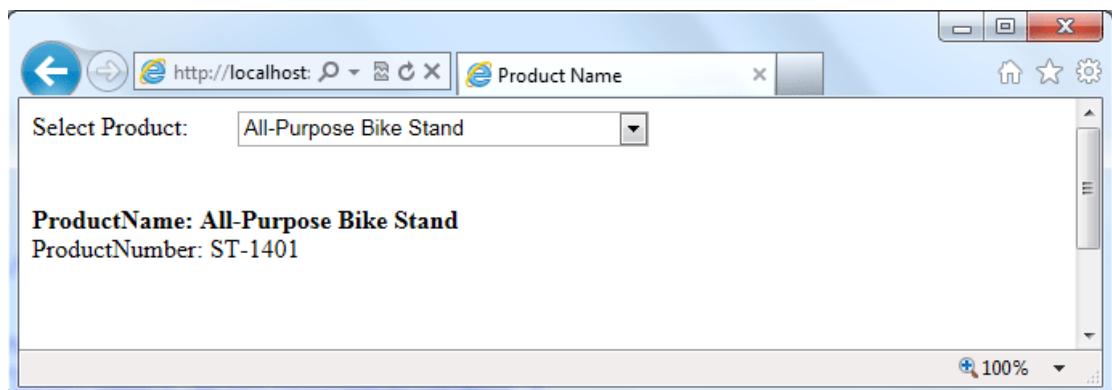
```

        sb.Append(reader["Name"]);

        sb.Append("</b><br />");
        sb.Append("ProductNumber: ");
        sb.Append(reader["ProductNumber"]);
        lblResults.Text = sb.ToString();
        reader.Close();
    }
catch (Exception err)
{
    lblResults.Text = "Không thể lấy dữ liệu. ";
    lblResults.Text += err.Message;
}
finally
{
    con.Close();
}
}
}

```

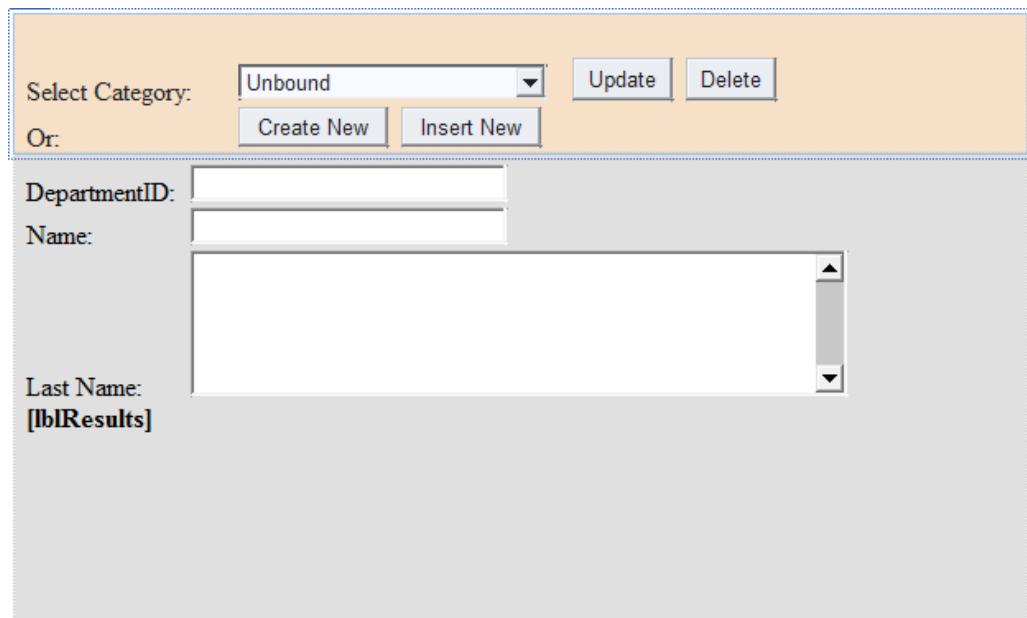
Kết quả thực thi như sau:



Hình 3.29. Kết quả khi thực thi trang vidu5-4.aspx

Ví dụ 5-5: Thực hiện việc cập nhật, thêm sửa và xóa dữ liệu, ta có cơ sở dữ liệu Shopgifts và bảng Category(CategoryID, DepartmentID, Name, Description).

Tạo trang vidu5-5.aspx với thiết kế giao diện như sau:



Hình 3.30: Giao diện thiết kế vidu5-5

Phần mã lệnh trong trang vidu5-5.aspx.cs như sau:

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Configuration;
using System.Data.SqlClient;
public partial class vidu5_5 : System.Web.UI.Page
{
    Private string connectionString =
ConfigurationManager.ConnectionStrings["Shopgifts"].ConnectionString;
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!this.IsPostBack)
        {
            FillCategoryList();
        }
    }
    private void FillCategoryList()
    {
        lstCategory.Items.Clear();
```

```

string selectSQL = "SELECT CategoryID, Name FROM Category";
// Định nghĩa đối tượng ADO.NET.
SqlConnection con = new SqlConnection(connectionString);
SqlCommand cmd = new SqlCommand(selectSQL, con);
SqlDataReader reader;
// Mở và đọc thông tin dữ liệu.
try
{
    con.Open();
    reader = cmd.ExecuteReader();
    // Đọc tất cả dữ liệu trong đối tượng reader
    // Dưa dữ liệu vào DropDownList, thông qua Value và Text.
    while (reader.Read())
    {
        ListItem newItem = new ListItem();
        newItem.Text = reader["Name"].ToString();
        newItem.Value = reader["CategoryID"].ToString();
        lstCategory.Items.Add(newItem);
    }
    reader.Close();
}
catch (Exception err)
{
    lblResults.Text = "Lỗi không thể liệt kê danh sách. ";
    lblResults.Text += err.Message;
}
finally
{
    con.Close();
}
}

protected void lstCategory_SelectedIndexChanged(object sender,
EventArgs e)
{

```

```

string selectSQL;
selectSQL = "SELECT * FROM Category ";
selectSQL += "WHERE CategoryID=" + 
lstCategory.SelectedItem.Value + "";
// Định nghĩa đối tượng ADO.NET.
SqlConnection con = new SqlConnection(connectionString);
SqlCommand cmd = new SqlCommand(selectSQL, con);
SqlDataReader reader;
// Mở và đọc thông tin dữ liệu.
try
{
    con.Open();
    reader = cmd.ExecuteReader();
    reader.Read();
    //Đưa dữ liệu vào Controls
    txtDepartmentID.Text = reader["DepartmentID"].ToString();
    txtName.Text = reader["Name"].ToString();
    txtDes.Text = reader["Description"].ToString();
    reader.Close();
    lblResults.Text = "";
}
catch (Exception err)
{
    lblResults.Text = "Không thể lấy dữ liệu. ";
    lblResults.Text += err.Message;
}
finally
{
    con.Close();
}
}

protected void cmdUpdate_Click(object sender, EventArgs e)
{
    // Định nghĩa đối tượng ADO.NET.
}

```

```

string updateSQL;
updateSQL = "UPDATE Category SET ";
updateSQL += "DepartmentID=@DepartmentID, ";
updateSQL += "Name=@Name, Description=@Description ";
updateSQL += "WHERE CategoryID=@CategoryID";
SqlConnection con = new SqlConnection(connectionString);
SqlCommand cmd = new SqlCommand(updateSQL, con);
// Add the parameters.
cmd.Parameters.AddWithValue("@DepartmentID",
txtDepartmentID.Text);
cmd.Parameters.AddWithValue("@Name", txtName.Text);
cmd.Parameters.AddWithValue("@Description", txtDes.Text);
cmd.Parameters.AddWithValue("@CategoryID",
lstCategory.SelectedItem.Value);
//Mở dữ liệu và thực hiện việc cập nhật dữ liệu
int updated = 0;
try
{
    con.Open();
    updated = cmd.ExecuteNonQuery();
    lblResults.Text = updated.ToString() + " bản ghi đã được cập
nhật.";
}
catch (Exception err)
{
    lblResults.Text = "Cập nhật lỗi. ";
    lblResults.Text += err.Message;
}
finally
{
    con.Close();
}
//Làm mới dữ liệu khi cập nhật thành công
if (updated > 0)

```

```

    {
        FillCategoryList();
    }
}

protected void cmdDelete_Click(object sender, EventArgs e)
{
    // Định nghĩa đối tượng ADO.NET.
    string deleteSQL;
    deleteSQL = "DELETE FROM Category ";
    deleteSQL += "WHERE CategoryID=@CategoryID";
    SqlConnection con = new SqlConnection(connectionString);
    SqlCommand cmd = new SqlCommand(deleteSQL, con);
    cmd.Parameters.AddWithValue("@CategoryID ",
    lstCategory.SelectedItem.Value);
    // Mở dữ liệu và thực thi xóa bản ghi.
    int deleted = 0;
    try
    {
        con.Open();
        deleted = cmd.ExecuteNonQuery();
        lblResults.Text = "Xóa bản ghi thành công.";
    }
    catch (Exception err)
    {
        lblResults.Text = "Lỗi không thể xóa. ";
        lblResults.Text += err.Message;
    }
    finally
    {
        con.Close();
    }
    // Nếu xóa dữ liệu thành công thì làm mới danh sách Category
    if (deleted > 0)
    {

```

```

        FillCategoryList();
    }
}

protected void cmdNew_Click(object sender, EventArgs e)
{
    txtDepartmentID.Text = "";
    txtDes.Text = "";
    txtName.Text = "";
    txtDepartmentID.Focus();
    lblResults.Text = "Click Insert New để thêm mới mẫu tin.";
}

protected void cmdInsert_Click(object sender, EventArgs e)
{
    if (txtDepartmentID.Text == "" || txtName.Text == "")
    {
        lblResults.Text = "Bạn không thể để CategoryID, DepartmentID, Name rỗng";
        return;
    }
    //Định nghĩa đối tượng ADO.NET
    string insertSQL;
    insertSQL = "INSERT INTO Category(";
    insertSQL += "DepartmentID, Name, ";
    insertSQL += "Description) ";
    insertSQL += "VALUES (";
    insertSQL += "@DepartmentID, @Name, ";
    insertSQL += "@Description)";
    SqlConnection con = new SqlConnection(connectionString);
    SqlCommand cmd = new SqlCommand(insertSQL, con);
    // Add the parameters.
    cmd.Parameters.Add(new SqlParameter("@DepartmentID",
        txtDepartmentID.Text));
    cmd.Parameters.Add(new SqlParameter("@Name",
        txtName.Text));
}

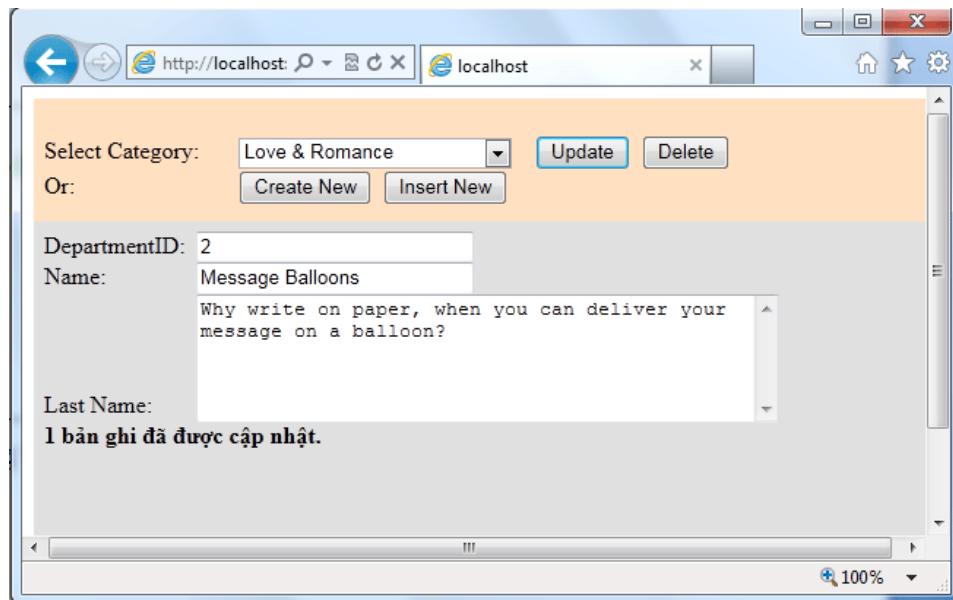
```

```

cmd.Parameters.Add(new SqlParameter("@Description",
txtDes.Text));
// Mở dữ liệu để thực hiện việc cập nhật dữ liệu mới.
int added = 0;
try
{
    con.Open();
    added = cmd.ExecuteNonQuery();
    lblResults.Text = added.ToString() + " bản ghi đã được thêm
vào.";
}
catch (Exception err)
{
    lblResults.Text = "Không thể thêm bản ghi... ";
    lblResults.Text += err.Message;
}
finally
{
    con.Close();
}
//Nếu thêm mới thành công thì làm mới danh sách Category
if (added > 0)
{
    FillCategoryList();
}
}
}

```

Kết quả duyệt trang này như sau:



Hình 3.31: Kết quả khi thực thi trang vidu5-5.aspx

Giải thích hoạt động của chương trình: Kích nút **Update**, **Delete** để cập nhật, xóa bản ghi hiện tại. Kích nút **Create New** để xóa trắng các Control Textbox và kích nút **Insert New** để thêm mới một bản ghi.

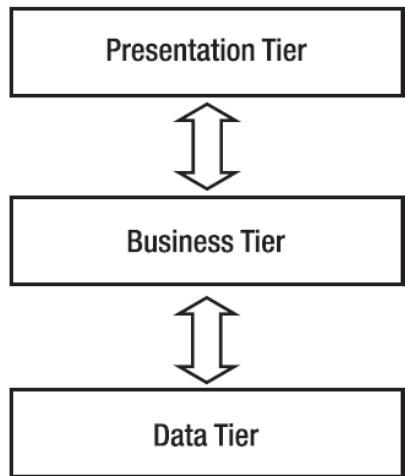
5. Xây dựng ứng dụng mẫu Website thương mại điện tử.

Trong phần này chúng ta sẽ làm việc với một Website thương mại điện tử (trang web bán hoa). Chúng ta sẽ nói về việc sử dụng công nghệ, các công cụ và làm thế nào để sử dụng được chúng.

5.1. Thiết kế ứng dụng

Trong bài toán này ta sẽ thiết kế ứng dụng theo mô hình 3 lớp.

Trước đây, đối với các phần mềm có sử dụng đến dữ liệu, người lập trình thường tích hợp việc giao tiếp với người sử dụng, xử lý rồi ghi xuống dữ liệu trên cùng một form (đây là mô hình một lớp). Nhưng trong kiến trúc 3 lớp (mô hình 3 lớp), phải có việc phân biệt rạch ròi giữa các lớp này. Mô hình 3 lớp có thể được mô tả như sau :



Hình 3.32: Mô hình 3 lớp

- **PresentationTier** (lớp giao diện): Đây là lớp thứ nhất, lớp này chỉ thuận xử lý việc giao tiếp với người sử dụng, nhập xuất, ... mà không thực hiện việc tính toán, kiểm tra, xử lý, hay các thao tác liên quan đến cơ sở dữ liệu.

- **Business Tier** (lớp xử lý) : Lớp thứ hai chuyên thực hiện các xử lý , kiểm tra các ràng buộc, các qui tắc ứng xử của phần mềm , các chức năng cốt yếu, ... Việc thực hiện này độc lập với cách thiết kế cũng như cài đặt giao diện. Thông tin cho lớp này thực hiện các xử lý của mình được lấy từ lớp giao diện.

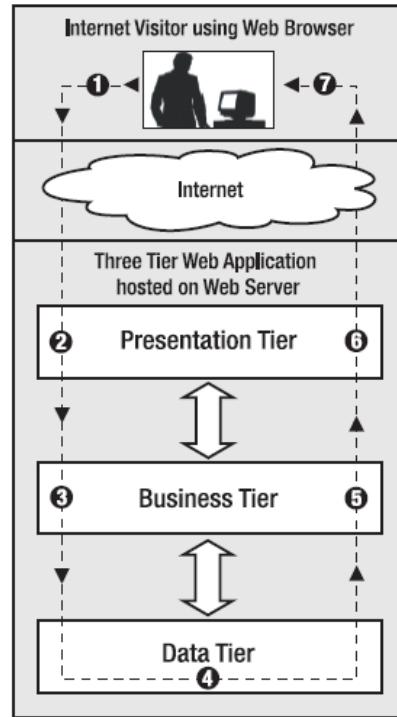
- **Data Tier** (lớp dữ liệu) : Lớp thứ ba này chuyên thực hiện các công việc liên quan đến dữ liệu. Dữ liệu có thể lấy từ cơ sở dữ liệu (Access, SQL Server ...) hoặc tập tin (text, binary, XML ...). Đối với cơ sở dữ liệu, lớp này thực hiện kết nối trực tiếp với cơ sở dữ liệu và thực hiện tất cả các thao tác liên quan đến cơ sở dữ liệu mà phần mềm cần thiết. Đối với tập tin, lớp này thực hiện việc đọc, ghi tập tin theo yêu cầu của phần mềm. Việc thực hiện này do lớp xử lý gọi.

Rõ ràng, với mô hình này, các công việc của từng lớp là độc lập với nhau. Việc thay đổi ở một lớp không làm thay đổi các lớp còn lại, thuận tiện hơn cho quá trình phát triển và bảo trì phần mềm.

- **Một kịch bản đơn giản cho ứng dụng thương mại điện tử:**

Nó đơn giản được hiểu là dữ liệu được thông qua và chuyển đổi giữa các lớp nếu bạn có một cái nhìn gần hơn là một ví dụ đơn giản. Để làm ví dụ liên quan đến dự án chúng ta sẽ phân tích một tình huống sẽ xảy ra trong dự án thương mại điện tử. Kịch bản này là điển hình cho mô hình 3 lớp.

Cũng giống như hầu hết các trang thương mại điện tử, trang web mà chúng ta làm sẽ có một giỏ mua hàng. Khách hàng truy cập sẽ thêm một sản phẩm vào giỏ mua hàng bằng cách nhấn vào nút thêm giỏ hàng (Add to Cart). Hình dưới đây sẽ cho ta thấy làm thế nào dòng chảy thông tin thông qua các ứng dụng khi ta nhấn nút đó.



Hình 3.33: Khách truy cập thao tác với một ứng dụng ba lớp

❖ ASP.NET và Kiến trúc ba lớp.

Cơ hoạt động của Web forms, Web User Controls, và Master Pages hình thành lên ứng dụng ba lớp. Chúng chính là một phần được tạo ra mã HTML mà người dùng duyệt Web bởi trình duyệt.

Giao diện người dùng được lưu trữ bởi các tập tin gọi là code behind của Web forms, Web User Control và Master Pages. Lưu ý rằng mặc dù bạn không cần phải sử dụng các tập tin code behind với các tập tin ASP.NET (có thể sử dụng để trộn lẫn mã HTML với mã lệnh giống như làm việc với ASP), ở đây chúng ta sẽ sử dụng mô hình code behind cho mô ứng dụng này.

❖ Sử dụng Ngôn ngữ C# và VB.NET

C# và VB.NET là hai ngôn ngữ có thể được sử dụng để viết code behind cho Web forms. Trong ứng dụng này chúng ta sẽ sử dụng ngôn ngữ C# để viết, và sẽ sử dụng ASP.NET phiên bản 2.0 trở lên để viết.

❖ Sử dụng SQL server 2005 trở lên

Trong giáo trình này chúng ta sẽ sử dụng SQL Server 2008.

5.2. Thiết kế cơ sở dữ liệu:

Thiết kế cơ sở dữ liệu bán quà tặng Shopgifts.

5.2.1. Thiết kế bảng Department:

Bảng Department dùng để lưu trữ gian hàng.

Tên trường	Kiểu dữ liệu	Các thuộc tính khác
DepartmentID	Int	Primary Key and Identity column
Name	Nvarchar(50)	Không được phép Null
Description	Nvarchar(1000)	Cho phép Null

The screenshot shows the 'ASUS-PC.ShopFlo... dbo.Department' table in SSMS. The 'Column Name' column lists 'DepartmentID', 'Name', and 'Description'. The 'Data Type' column shows 'int' for DepartmentID, 'nvarchar(50)' for Name, and 'nvarchar(1000)' for Description. The 'Allow Nulls' column shows checkboxes for Name and Description, while DepartmentID has its checkbox unchecked. Below the table, the 'Column Properties' window is open for the 'Description' column. Under 'Identity Specification', the '(Is Identity)' checkbox is checked, and the 'Identity Increment' and 'Identity Seed' fields are both set to 1. Other properties like 'Deterministic', 'DTS-published', and 'Full-text Specification' are also visible.

Hình 3.34: Bảng Department

5.2.2. Thiết kế bảng Category.

Bảng Category sử dụng để lưu trữ danh mục hàng, một gian hàng có nhiều danh mục hàng.

Tên trường	Kiểu dữ liệu	Thuộc tính khác
CategoryID	Int	Primary Key and Identity column
DepartmentID	Int	Không cho phép Null
Name	Nvarchar(50)	Không cho phép Null

Description

Nvarchar(1000)

Cho phép Null

The screenshot shows the 'ASUS-PC.ShopFlowers - dbo.Category*' table structure. The 'Description' column is set to nvarchar(1000) and allows null values. The 'Column Properties' dialog for the 'Description' column is displayed, showing detailed settings like 'Is Identity' being selected.

Hình 3.35: Bảng Category

5.2.3. Thiết kế bảng Product

Bảng Product sử dụng để lưu trữ thông tin sản phẩm.

Tên trường	Kiểu dữ liệu	Thuộc tính khác
ProductID	Int	Primary Key and Identity column
Name	Nvarchar(50)	Không cho phép Null
Description	Nvarchar(5000)	Không cho phép Null
Price	Money	Không cho phép Null
Image1FileName	Varchar(50)	Cho phép null
Image2FileName	Varchar(50)	Cho phép null
OnCatalogPromotion	Bit	Không cho phép null và có giá trị mặc định bằng 0
OnDepartmentPromotion	Bit	Không cho phép null và có giá trị mặc định bằng 0

The screenshot shows the 'Product' table structure in the 'ASUS-PC.ShopFlowers - dbo' database. The table has eight columns:

Column Name	Data Type	Allow Nulls
ProductID	int	<input type="checkbox"/>
Name	nvarchar(50)	<input type="checkbox"/>
Description	varchar(5000)	<input type="checkbox"/>
Price	money	<input type="checkbox"/>
Image1FileName	varchar(50)	<input checked="" type="checkbox"/>
Image2FileName	varchar(50)	<input checked="" type="checkbox"/>
OnCatalogPromotion	bit	<input type="checkbox"/>
OnDepartmentPromotion	bit	<input type="checkbox"/>

The 'ProductID' column is defined as an identity column with the following properties:

- Full-text Specification: No
- Has Non-SQL Server St. No
- Identity Specification: Yes
 - (Is Identity): Yes
 - Identity Increment: 1
 - Identity Seed: 1
 - Indexable: Yes
 - Is Columnset: No
 - Is Snarce: No

Hình 3.36: Bảng Product

Mô tả trường dữ liệu của bảng Product

- ProductID xác định duy nhất một sản phẩm, là khóa chính của bảng.
- Name lưu trữ tên của sản phẩm.
- Description lưu trữ thông tin mô tả của sản phẩm.
- Price lưu trữ giá sản phẩm.
- Image1FileName lưu trữ tên file ảnh thứ nhất của sản phẩm (được hiển thị trên danh sách sản phẩm).
- Image2FileName lưu trữ tên file ảnh thứ hai của sản phẩm, nó được hiển thị trên trang chi tiết sản phẩm.
- OnCatalogPromotion là một giá trị bit (có thể được thiết lập để 0 và 1) xác định xem sản phẩm có được đặt lên trang chủ không. Trang chủ của trang Web sẽ liệt kê các sản phẩm có bit thiết lập bằng 1. Không cho phép null và giá trị mặc định bằng 0.
- OnDepartmentPromotion là một giá trị bit xác định liệu sản phẩm có được giới thiệu trên các trang bộ phận không. Khi truy cập vào một gian hàng, trang web được chỉ định hiển thị những sản phẩm đặc trưng của bộ phận đó. Nếu một sản phẩm thuộc nhiều hơn một gian hàng, nó sẽ được liệt

kê như là các đặc trưng của các gian hàng đó. Trường này không chấp nhận Null và giá trị mặc định bằng 0.

5.2.4. Tạo bảng ProductCategory.

Bảng này sử dụng để liên kết bảng Product với bảng Category (thực hiện mối quan hệ nhiều-nhiều với sản phẩm và danh mục sản phẩm).

Column Name	Data Type	Allow Nulls
ProductID	int	<input type="checkbox"/>
CategoryID	int	<input type="checkbox"/>

Hình 3.37: Bảng ProductCategory

5.2.5. Thiết kế bảng Orders.

Bảng này dùng để lưu trữ thông tin của đơn đặt hàng.

Tên cột	Kiểu dữ liệu	Allow Nulls	Thuộc tính khác
OrderID	Int	No	Khóa chính, tự động tăng
DateCreated	Smalldatetime	No	Mặc định: GetDate()
DateShipped	Smalldatetime	Yes	
Verified	Bit	No	Default value or Binding: 0
Completed	Bit	No	Default value or Binding: 0
Cancel	Bit	No	Default value or Binding: 0
Comment	Nvarchar(1000)	Yes	
CustomerName	Nvarchar(50)	Yes	
CustomerEmail	Varchar(50)	Yes	
ShippingAddress	Nvarchar(500)	Yes	

Column Name	Data Type	Allow Nulls
OrderID	int	<input type="checkbox"/>
DateCreated	smalldatetime	<input type="checkbox"/>
DateShipped	smalldatetime	<input checked="" type="checkbox"/>
Verified	bit	<input type="checkbox"/>
Completed	bit	<input type="checkbox"/>
Canceled	bit	<input type="checkbox"/>
Comments	nvarchar(1000)	<input checked="" type="checkbox"/>
CustomerName	nvarchar(50)	<input checked="" type="checkbox"/>
CustomerEmail	varchar(50)	<input checked="" type="checkbox"/>
ShippingAddress	nvarchar(500)	<input checked="" type="checkbox"/>

Column Properties

Has Non-SQL Server Sub No
 Identity Specification Yes
 (Is Identity) Yes
 Identity Increment 1
 Identity Seed 1
 Indexable Yes
 Is Columnset No
 To Column
Identity Specification

Hình 3.38: Bảng Orders

5.2.6. Thiết kế bảng OrderDetails.

Bảng này dùng để lưu trữ thông tin chi tiết về đơn hàng.

Tên cột	Kiểu dữ liệu	Allow Nulls	Thuộc tính khác
OrderID	Int	No	Khóa chính
ProductID	Int	No	Khóa chính
ProductName	Nvarchar(50)	No	
Quantity	Int	No	
UnitCost	Money	No	
SubTotal		No	Computed Column Specification Formula: Quantity * UnitCost Is Persisted: No

Column Name	Data Type	Allow Nulls
OrderID	int	<input type="checkbox"/>
ProductID	int	<input type="checkbox"/>
ProductName	nvarchar(50)	<input type="checkbox"/>
Quantity	int	<input type="checkbox"/>
UnitCost	money	<input type="checkbox"/>
Subtotal		<input checked="" type="checkbox"/>

Column Properties

Table Designer

- Collation: <database default>
- Computed Column Specification (Formula): $([Quantity]*[UnitCost])$
- Is Persisted: No
- Condensed Data Type
- Description
- Deterministic: Yes
- DTS-published: No

Computed Column Specification

Hình 3.39: Bảng OrderDetail

5.2.7. Thiết kế bảng ShoppingCart

Bảng này dùng để lưu trữ thông tin về giỏ hàng

Column Name	Data Type	Allow Nulls
CartID	char(36)	<input type="checkbox"/>
ProductID	int	<input type="checkbox"/>
Quantity	int	<input type="checkbox"/>
DateAdded	smalldatetime	<input type="checkbox"/>

Column Properties

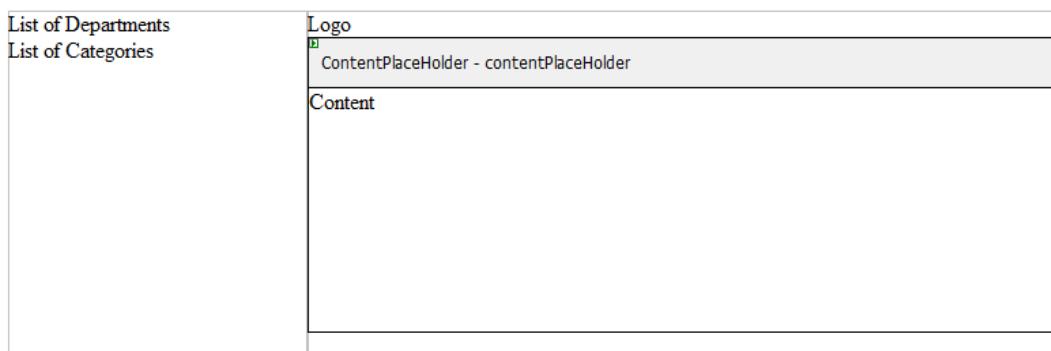
(General) <table border="1"> <tr> <td>(Name)</td> <td>CartID</td> </tr> <tr> <td>Allow Nulls</td> <td>No</td> </tr> <tr> <td>Data Type</td> <td>char</td> </tr> </table>			(Name)	CartID	Allow Nulls	No	Data Type	char
(Name)	CartID							
Allow Nulls	No							
Data Type	char							
(Advanced)								

Hình 3.40: Bảng ShoppingCart

5.3. Viết chương trình

Thiết kế trang chủ website có bố cục như sau:

- Phần bên trái là danh sách của gian hàng (Department) và danh mục hàng (Category).
- Phần bên phải: Ở phía trên là logo của website, phía dưới là hiển thị nội dung, sản phẩm của website.



Hình 3.41: Bố cục trang chủ

5.3.1. File web.config

Trong file web.config ta thêm vào đoạn mã sau:

```
<appSettings>
    <add key="EnableErrorLogEmail" value="true" />
    <add key="ProductsPerPage" value="6"/>
    <add key="ProductDescriptionLength" value="60"/>
    <add key="SiteName" value="GiftShop"/>
</appSettings>
<connectionStrings>
```

```
<add name="ShopGiftConnection" connectionString="Server=Asus-  
PC;Integrated Security=True;Database=ShopGifts"  
providerName="System.Data.SqlClient"/>  
</connectionStrings>
```

Để xây dựng được trang này đầu tiên chúng ta cần xây dựng các lớp dùng để truy cập dữ liệu.

5.3.2. Xây dựng lớp GiftShopConfiguration

Lớp này dùng để thiết lập cấu hình cho trang web. Kích chuột phải vào Tên dự án trong Solution Explorer và chọn Add New Item từ menu, chọn mẫu class và đặt tên cho nó là GiftShopConfiguration.cs. Sau đó kích Add. Bạn sẽ thấy hiển thị hộp thoại hỏi bạn “about adding the class into the App_Code folder”, chọn Yes.

```
using System;  
using System.Configuration;  
/// <summary>  
/// Kho lưu trữ thiết lập cấu hình Shopgift  
/// </summary>  
public static class ShopgiftConfiguration  
{  
    // Số ngày hết hạn mua hàng  
    private readonly static int cartPersistDays;  
    // Chuỗi kết nối  
    private readonly static string dbConnectionString;  
    // data provider name  
    private readonly static string dbProviderName;  
    // số sản phẩm trong một trang  
    private readonly static int productsPerPage;  
    // độ dài đoạn mô tả trong danh sách  
    private readonly static int productDescriptionLength;  
    // tên site của bạn  
    private readonly static string siteName;  
    // khởi tạo các thuộc tính constructor  
    static ShopgiftConfiguration()  
    {
```

```

    cartPersistDays =
    Int32.Parse(ConfigurationManager.AppSettings["CartPersistDays"]);
    dbConnectionString =
    ConfigurationManager.ConnectionStrings["ShopgiftConnection"].Conn
    ectionString;
    dbProviderName =
    ConfigurationManager.ConnectionStrings["ShopgiftConnection"].Provi
    derName;
    productsPerPage =
    Int32.Parse(ConfigurationManager.AppSettings["ProductsPerPage"]);
    productDescriptionLength =
    Int32.Parse(ConfigurationManager.AppSettings["ProductDescriptionLe
    ngth"]);
    siteName = ConfigurationManager.AppSettings["SiteName"];
}

// trả về ngày hết hạn mua hàng
public static int CartPersistDays
{
    get
    {
        return cartPersistDays;
    }
}

// Trả về chuỗi kết nối đến cơ sở dữ liệu
public static string DbConnectionString
{
    get
    {
        return dbConnectionString;
    }
}

// trả về tên dịch vụ
public static string DbProviderName
{
}

```

```

get
{
    return dbProviderName;
}
}

// Trả về số sản phẩm được hiển thị trên một trang
public static int ProductsPerPage
{
    get
    {
        return productsPerPage;
    }
}

// Trả về tên của Website
public static string SiteName
{
    get
    {
        return siteName;
    }
}
}

```

5.3.3. Xây dựng lớp truy cập cơ sở dữ liệu GenericDataAccess.cs

```

using System;
using System.Data;
using System.Data.Common;
using System.Configuration;
/// <summary>
/// Lớp có chức năng truy cập dữ liệu từ tầng Business
/// </summary>
public static class GenericDataAccess
{
    // static constructor

```

```
static GenericDataAccess()
{
    // TODO: Add constructor logic here
}

// Thực thi lệnh command trả về kết quả là đối tượngDataTable
public static DataTable ExecuteSelectCommand(DbCommand
command)
{
    // Datatable dùng để trả về
    DataTable table;
    // thực thi lệnh command nếu kết nối ngược lại thì đóng kết nối
    try
    {
        // Mở kết nối
        command.Connection.Open();
        // thực thi lệnh và lưu lại vào một DataTable
        DbDataReader reader = command.ExecuteReader();
        table = new DataTable();
        table.Load(reader);
        // đóng reader
        reader.Close();
    }
    catch (Exception ex)
    {
    }
    finally
    {
        // Đóng kết nối
        command.Connection.Close();
    }
    return table;
}

// Tạo một đối tượng DbCommand
public static DbCommand CreateCommand()
```

```

{
    // database provider name
    string dataProviderName = ShopgiftConfiguration.DbProviderName;
    // Chuỗi kết nối dữ liệu
    string connectionString =
        ShopgiftConfiguration.DbConnectionString;
    // tạo một data provider factory
    DbProviderFactory factory =
        DbProviderFactories.GetFactory(dataProviderName);
    // kết nối dữ liệu
    DbConnection conn = factory.CreateConnection();
    // đặt chuỗi kết nối
    conn.ConnectionString = connectionString;
    // tạo lệnh thực thi
    DbCommand comm = conn.CreateCommand();
    // đặt thực thi stored procedure
    comm.CommandType = CommandType.StoredProcedure;
    // trả về đối tượng command
    return comm;
}
}

```

5.3.4. Xây dựng lớp CatalogAccess.cs

Lớp này dùng để thao tác với dữ liệu

```

using System;
using System.Data;
using System.Data.Common;

/// <summary>
/// chi tiết gian hàng
/// </summary>
public struct DepartmentDetails
{
    public string Name;
}

```

```
    public string Description;
}
/// <summary>
/// Danh mục chi tiết
/// </summary>
public struct CategoryDetails
{
    public int DepartmentId;
    public string Name;
    public string Description;
}
/// <summary>
/// chi tiết sản phẩm
/// </summary>
public struct ProductDetails
{
    public string Name;
    public string Description;
    public decimal Price;
    public string Image1FileName;
    public string Image2FileName;
    public bool OnDepartmentPromotion;
    public bool OnCatalogPromotion;
}
/// <summary>
/// Các thành phần của lớp business tier
/// </summary>
public static class CatalogAccess
{
    static CatalogAccess()
    {
        //
        // TODO: Add constructor logic here
        //
    }
}
```

```

}

// Lấy danh sách các gian hàng
public static DataTable GetDepartments()
{
    DbCommand comm = GenericDataAccess.CreateCommand();
    // thực thi store procedure
    comm.CommandText = "GetDepartments";
    // thực thi store và trả về kết quả
    return GenericDataAccess.ExecuteSelectCommand(comm);
}

// Chi tiết gian hàng
public static DepartmentDetails GetDepartmentDetails(string
departmentId)
{
    // tạo DbCommand
    DbCommand comm = GenericDataAccess.CreateCommand();
    // thực thi thủ tục
    comm.CommandText = "GetDepartmentDetails";
    // Tạo một tham số
    DbParameter param = comm.CreateParameter();
    param.ParameterName = "@DepartmentID";
    param.Value = departmentId;
    param.DbType = DbType.Int32;
    comm.Parameters.Add(param);
    // Thực thi thủ tục
    DataTable table =
        GenericDataAccess.ExecuteSelectCommand(comm);
    // Lấy dữ liệu đưa vào đối tượng DepartmentDetails
    DepartmentDetails details = new DepartmentDetails();
    if (table.Rows.Count > 0)
    {
        details.Name = table.Rows[0]["Name"].ToString();
        details.Description = table.Rows[0]["Description"].ToString();
    }
}

```

```

    }

    // trả về chi tiết gian hàng
    return details;
}

// Chi tiết danh mục hàng
public static CategoryDetails GetCategoryDetails(string categoryId)
{
    // Cấu hình đối tượng DbCommand
    DbCommand comm = GenericDataAccess.CreateCommand();

    // Thực thi thủ tục
    comm.CommandText = "GetCategoryDetails";
    // Tạo một tham số
    DbParameter param = comm.CreateParameter();
    param.ParameterName = "@CategoryID";
    param.Value = categoryId;
    param.DbType = DbType.Int32;
    comm.Parameters.Add(param);

    // Thực thi thủ tục
    DataTable table =
        GenericDataAccess.ExecuteSelectCommand(comm);
    // lấy dữ liệu đưa vào đối tượng CategoryDetails
    CategoryDetails details = new CategoryDetails();
    if (table.Rows.Count > 0)
    {
        details.DepartmentId =
            Int32.Parse(table.Rows[0]["DepartmentID"].ToString());
        details.Name = table.Rows[0]["Name"].ToString();
        details.Description = table.Rows[0]["Description"].ToString();
    }

    // Trả về chi tiết danh mục hàng
    return details;
}

// Get product details
public static ProductDetails GetProductDetails(string productId)

```

```

{
    // Cấu hình đối tượng DbCommand
    DbCommand comm = GenericDataAccess.CreateCommand();
    // Thực thi thủ tục
    comm.CommandText = "GetProductDetails";
    // Tạo một tham số
    DbParameter param = comm.CreateParameter();
    param.ParameterName = "@ProductID";
    param.Value = productId;
    param.DbType = DbType.Int32;
    comm.Parameters.Add(param);
    // Thực thi thủ tục
    DataTable table =
        GenericDataAccess.ExecuteSelectCommand(comm);
    // đưa dữ liệu vào đối tượng ProductDetails
    ProductDetails details = new ProductDetails();
    if (table.Rows.Count > 0)
    {
        // Nhận hàng đầu tiên
        DataRow dr = table.Rows[0];

        details.Name = dr["Name"].ToString();
        details.Description = dr["Description"].ToString();
        details.Price = Decimal.Parse(dr["Price"].ToString());
        details.Image1FileName = dr["Image1FileName"].ToString();
        details.Image2FileName = dr["Image2FileName"].ToString();
        details.OnDepartmentPromotion =
            bool.Parse(dr["OnDepartmentPromotion"].ToString());
        details.OnCatalogPromotion =
            bool.Parse(dr["OnCatalogPromotion"].ToString());
    }
    return details;
}
// Danh sách danh mục hàng trong gian hàng

```

```

public static DataTable GetCategoriesInDepartment(string
departmentId)
{
    // Cấu hình đối tượng DbCommand
    DbCommand comm = GenericDataAccess.CreateCommand();
    // Thực thi thủ tục
    comm.CommandText = "GetCategoriesInDepartment";
    // Tạo một tham số
    DbParameter param = comm.CreateParameter();
    param.ParameterName = "@DepartmentID";
    param.Value = departmentId;
    param.DbType = DbType.Int32;
    comm.Parameters.Add(param);
    // Thực thi thủ tục
    return GenericDataAccess.ExecuteSelectCommand(comm);
}

public static DataTable GetProductsOnCatalogPromotion(string
pageNumber, out int howManyPages)
{
    // Cấu hình đối tượng DbCommand
    DbCommand comm = GenericDataAccess.CreateCommand();
    // Thực thi thủ tục
    comm.CommandText = "GetProductsOnCatalogPromotion";
    // Tạo một tham số
    DbParameter param = comm.CreateParameter();
    param.ParameterName = "@DescriptionLength";
    param.Value = GiftShopConfiguration.ProductDescriptionLength;
    param.DbType = DbType.Int32;
    comm.Parameters.Add(param);
    // Tạo một tham số
    param = comm.CreateParameter();
    param.ParameterName = "@PageNumber";
    param.Value = pageNumber;
    param.DbType = DbType.Int32;
}

```

```

comm.Parameters.Add(param);
// Tạo một tham số
param = comm.CreateParameter();
param.ParameterName = "@ProductsPerPage";
param.Value = GiftShopConfiguration.ProductsPerPage;
param.DbType = DbType.Int32;
comm.Parameters.Add(param);
// Tạo một tham số
param = comm.CreateParameter();
param.ParameterName = "@HowManyProducts";
param.Direction = ParameterDirection.Output;
param.DbType = DbType.Int32;
comm.Parameters.Add(param);
// Thực thi thủ tục and save the results in a DataTable
DataTable table =
GenericDataAccess.ExecuteSelectCommand(comm);
// Tổng số trang
int howManyProducts =
Int32.Parse(comm.Parameters["@HowManyProducts"].Value.ToString());
howManyPages = (int)Math.Ceiling((double)howManyProducts /
(double)GiftShopConfiguration.ProductsPerPage);
// trả về trang sản phẩm
return table;
}
public static DataTable GetProductsOnDepartmentPromotion(string
departmentId, string pageNumber, out int howManyPages)
{
// Cấu hình đối tượng DbCommand
DbCommand comm = GenericDataAccess.CreateCommand();
// Thực thi thủ tục
comm.CommandText = "GetProductsOnDepartmentPromotion";
// Tạo một tham số
DbParameter param = comm.CreateParameter();

```

```

param.ParameterName = "@DepartmentID";
param.Value = departmentId;
param.DbType = DbType.Int32;
comm.Parameters.Add(param);
// Tạo một tham số
param = comm.CreateParameter();
param.ParameterName = "@DescriptionLength";
param.Value = GiftShopConfiguration.ProductDescriptionLength;
param.DbType = DbType.Int32;
comm.Parameters.Add(param);
// Tạo một tham số
param = comm.CreateParameter();
param.ParameterName = "@PageNumber";
param.Value = pageNumber;
param.DbType = DbType.Int32;
comm.Parameters.Add(param);
// Tạo một tham số
param = comm.CreateParameter();
param.ParameterName = "@ProductsPerPage";
param.Value = GiftShopConfiguration.ProductsPerPage;
param.DbType = DbType.Int32;
comm.Parameters.Add(param);
// Tạo một tham số
param = comm.CreateParameter();
param.ParameterName = "@HowManyProducts";
param.Direction = ParameterDirection.Output;
param.DbType = DbType.Int32;
comm.Parameters.Add(param);
// Thực thi thủ tục and save the results in a DataTable
DataTable table =
GenericDataAccess.ExecuteSelectCommand(comm);
// tổng số trang
int howManyProducts =
Int32.Parse(comm.Parameters["@HowManyProducts"].Value.ToString())

```

```

));
howManyPages = (int)Math.Ceiling((double)howManyProducts /
    (double)GiftShopConfiguration.ProductsPerPage);
return table;
}

public static DataTable GetProductsInCategory(string categoryId,
string pageNumber, out int howManyPages)
{
    // Cấu hình đối tượng DbCommand
    DbCommand comm = GenericDataAccess.CreateCommand();
    // Thực thi thủ tục
    comm.CommandText = "GetProductsInCategory";
    // Tạo một tham số
    DbParameter param = comm.CreateParameter();
    param.ParameterName = "@CategoryID";
    param.Value = categoryId;
    param.DbType = DbType.Int32;
    comm.Parameters.Add(param);
    // Tạo một tham số
    param = comm.CreateParameter();
    param.ParameterName = "@DescriptionLength";
    param.Value = GiftShopConfiguration.ProductDescriptionLength;
    param.DbType = DbType.Int32;
    comm.Parameters.Add(param);
    // Tạo một tham số
    param = comm.CreateParameter();
    param.ParameterName = "@PageNumber";
    param.Value = pageNumber;
    param.DbType = DbType.Int32;
    comm.Parameters.Add(param);
    // Tạo một tham số
    param = comm.CreateParameter();
    param.ParameterName = "@ProductsPerPage";
}

```

```

param.Value = GiftShopConfiguration.ProductsPerPage;
param.DbType = DbType.Int32;
comm.Parameters.Add(param);
// Tạo một tham số
param = comm.CreateParameter();
param.ParameterName = "@HowManyProducts";
param.Direction = ParameterDirection.Output;
param.DbType = DbType.Int32;
comm.Parameters.Add(param);
// Thực thi thủ tục and save the results in a DataTable
DataTable table =
GenericDataAccess.ExecuteSelectCommand(comm);
int howManyProducts =
Int32.Parse(comm.Parameters["@HowManyProducts"].Value.ToString());
howManyPages = (int)Math.Ceiling((double)howManyProducts /
(double)GiftShopConfiguration.ProductsPerPage);
return table;
}
}

```

5.3.5. Tạo trang DepartmentList.ascx

Tạo một thủ tục dùng để hiện thị danh sách Department trong SQL Server

```

--Liệt kê danh sách gian hàng
CREATE PROCEDURE GetDepartments AS
SELECT DepartmentID, Name, Description
FROM Department

```

- Đầu tiên tạo một thư mục UserControl. Kích chuột phải lên thư mục UserControl và chọn Add New Item. Chọn Web User Control và đặt tên là DepartmentList.ascx.
- Trong cửa sổ Design, Ở thanh toolbox mở Data tab, và kích đúp vào Datalist. Nó sẽ thêm một Control DataList vào DepartmentList.ascx.
- Sử dụng bảng thuộc tính của, và thay đổi một số thuộc tính sau:

Tên thuộc tính	Giá trị
(ID)	list
Width	200px
CssClass	DepartmentListContent
HeaderStyle-CssClass	DepartmentListHead

- Mở cửa sổ Design trang DepartmentList, phải chuột vào Datalist, và Chọn **Edit Template => Header and Footer Templates.**
 - Chọn một phần Header Template.
 - Chuột phải vào DataList và chọn **Edit Template => Item Template.**
 - Thêm vào một Hyperlink Control từ tab Standard toolbox vào Item Template.
 - Đặt lại thuộc tính Text của Control HyperLink là rỗng
 - Chuyển sang Source code và chỉnh sửa lại code như sau:

```
<asp:DataList ID="list" runat="server" Width="200px"
CssClass="DepartmentListContent">
<ItemTemplate>
    &nbsp;&raqo;
    <asp:HyperLink
        ID="HyperLink1"
        Runat="server"
        NavigateUrl='<%# "../Catalog.aspx?DepartmentID=" +
        Eval("DepartmentID")%>'
        Text='<%# Eval("Name") %>' 
        ToolTip='<%# Eval("Description") %>'

    CssClass='<%# Eval("DepartmentID").ToString() ==
    Request.QueryString["DepartmentID"] ?
    "DepartmentSelected" : "DepartmentUnselected" %>'>
    </asp:HyperLink>
    &nbsp;&laquo;
</ItemTemplate>
<HeaderTemplate>
```

```

Choose a Department
</HeaderTemplate>
<HeaderStyle CssClass="DepartmentListHead" />
</asp:DataList>

```

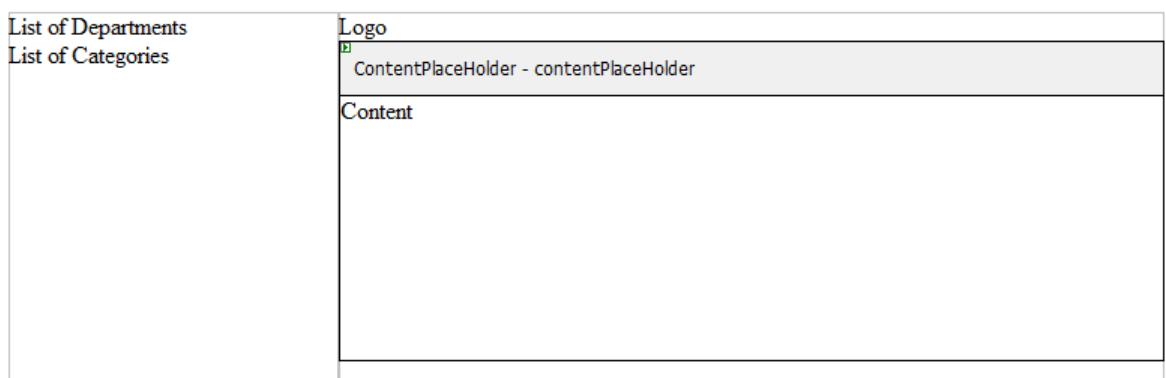
- Mở file code-behind của User control (DepartmentList.ascx.cs) và thêm vào đoạn mã trong sự kiện Page_Load.

```

// Tải toàn bộ danh sách Department vào DataList
protected void Page_Load(object sender, EventArgs e)
{
    // CatalogAccess.GetDepartments trả về đối tượng được chứa trong
    // bảng
    // dữ liệu sẽ được đọc vào ItemTemplate của DataList
    list.DataSource = CatalogAccess.GetDepartments();
    // ràng buộc dữ liệu với Datalist
    list.DataBind();
}

```

- Tạo một Master Page bằng cách phải chuột vào tên dự án trong Solution Explorer/Add New Item và chọn Master Page, đặt tên là GiftShop.Master. Tiếp theo ta thiết kế giao diện cho trang Master Page như hình sau:



Hình 3.42: Cửa sổ Design trang Master page

- Mở tệp GiftShop.master ở cửa sổ Design. Kéo thả trang DepartmentList.ascx gần từ “List of Departments” và Xoá từ “List of Departments” ở cột đó đi.



Hình 3.43: Thêm danh sách Departments vào trang Master page

- Tạo một Web form với tên Default.aspx và trong khi tạo ta chọn “Select master page”. Khi đó nó sẽ nhận layout của trang master page.
- Viết Styles cho trang Web
 - + Kích chuột phải trong thư mục gốc Solution Explorer và chọn Add Folder/ Theme Folder đặt tên cho thư mục mới là GiftShopDefault.
 - + Kích chuột phải vào GiftShopDefault trong Solution Explorer và chọn Add New Item. Từ cửa sổ Templates, chọn Style Sheet và đặt tên là GiftShop.css, sau đó kích Add.
 - + Mở GiftShop.css bằng cách kích đúp chuột vào tệp, xoá nội dung ở trong thêm vào đoạn mã styles như dưới đây:

```
.DepartmentListHead
{
    border-right: #01a647 1px solid;
    border-top: #01a647 1px solid;
    border-left: #01a647 1px solid;
    border-bottom: #01a647 1px solid;
    background-color: #30b86e;
    font-family: Verdana, Arial;
    font-weight: bold;
    font-size: 10pt;
    color: #f5f5dc;
```

```
padding-left: 3px;  
text-align: center;  
}  
.DepartmentListContent  
{  
border-right: #01a647 1px solid;  
border-top: #01a647 1px solid;  
border-left: #01a647 1px solid;  
border-bottom: #01a647 1px solid;  
background-color: #9fe1bb;  
text-align: center;  
}  
a.DepartmentUnselected  
{  
font-family: Verdana, Arial;  
font-weight: bold;  
font-size: 9pt;  
color: #5f9ea0;  
line-height: 25px;  
padding-left: 5px;  
text-decoration: none;  
}  
a.DepartmentUnselected:hover  
{  
padding-left: 5px;  
color: #2e8b57;  
}  
a.DepartmentSelected  
{  
font-family: Verdana, Arial;  
font-weight: bold;  
font-size: 9pt;  
color: #556b2f;  
line-height: 25px;
```

```

padding-left: 5px;
text-decoration: none;
}

```

- Cuối cùng mở file Web.config và thêm vào theme phần mặc định của style vào như sau:

```

<system.web>
    <pages theme="GiftShopDefault"/>
        <!--
        ....

```

- Nhấn F5 để chạy ta được kết quả như sau:



Hình 3.44: Thực thi GiftShop Project

5.3.6. Thiết kế trang hiển thị danh sách sản phẩm

Tạo một trang ProductsList.ascx

Trước khi viết cho trang ProductList.ascx ta cần tạo các thủ tục cho SQL Server sau:

- Thủ tục GetProductsOnDepartmentPromotion liệt kê danh sách các sản phẩm đặc biệt trong gian hàng.

```

CREATE PROCEDURE GetProductsOnDepartmentPromotion
(@DepartmentID INT,
@DescriptionLength INT,
@PageNumber INT,
@ProductsPerPage INT,
@HowManyProducts INT OUTPUT)
AS
-- declare a new TABLE variable

```

```

DECLARE @Products TABLE
(RowNumber INT,
 ProductID INT,
 Name NVARCHAR(50),
 Description NVARCHAR(5000),
 Price MONEY,
 Image1FileName VARCHAR(50),
 Image2FileName VARCHAR(50),
 OnDepartmentPromotion BIT,
 OnCatalogPromotion BIT)
-- populate the table variable with the complete list of products
INSERT INTO @Products
SELECT ROW_NUMBER() OVER (ORDER BY ProductID) AS Row,
       ProductID, Name, SUBSTRING>Description, 1,
       @DescriptionLength + '...' AS Description,
       Price, Image1FileName, Image2FileName,
       OnDepartmentPromotion, OnCatalogPromotion
FROM
(SELECT DISTINCT Product.ProductID, Product.Name,
       SUBSTRING(Product.Description, 1, @DescriptionLength) + '...' AS Description,
       Price, Image1FileName, Image2FileName, OnDepartmentPromotion,
       OnCatalogPromotion
FROM Product INNER JOIN ProductCategory
          ON Product.ProductID = ProductCategory.ProductID
          INNER JOIN Category
          ON ProductCategory.CategoryID = Category.CategoryID
WHERE Product.OnDepartmentPromotion = 1
      AND Category.DepartmentID = @DepartmentID
) AS ProductOnDepPr
-- return the total number of products using an OUTPUT variable
SELECT @HowManyProducts = COUNT(ProductID) FROM
@Products
-- extract the requested page of products

```

```

SELECT ProductID, Name, Description, Price, Image1FileName,
       Image2FileName, OnDepartmentPromotion, OnCatalogPromotion
FROM @Products
WHERE RowNumber > (@PageNumber - 1) * @ProductsPerPage
AND RowNumber <= @PageNumber * @ProductsPerPage

```

- Tạo thủ tục GetProductsInCategory hiển thị danh sách sản phẩm theo danh mục sản phẩm

```

CREATE PROCEDURE GetProductsInCategory
(@CategoryID INT,
@DescriptionLength INT,
@PageNumber INT,
@ProductsPerPage INT,
@HowManyProducts INT OUTPUT)
AS
-- Tạo một bảng
DECLARE @Products TABLE
(RowNumber INT,
ProductID INT,
Name NVARCHAR(50),
Description NVARCHAR(5000),
Price MONEY,
Image1FileName VARCHAR(50),
Image2FileName VARCHAR(50),
OnDepartmentPromotion BIT,
OnCatalogPromotion BIT)
-- bảng chứa danh sách sản phẩm được lọc
INSERT INTO @Products
SELECT ROW_NUMBER() OVER (ORDER BY Product.ProductID),
       Product.ProductID, Name,
       SUBSTRING>Description, 1, @DescriptionLength) + '...' AS
Description,
       Price, Image1FileName, Image2FileName,
       OnDepartmentPromotion, OnCatalogPromotion

```

```

FROM Product INNER JOIN ProductCategory
    ON Product.ProductID = ProductCategory.ProductID
WHERE ProductCategory.CategoryID = @CategoryID
-- đưa ra tổng số sản phẩm có
SELECT @HowManyProducts = COUNT(ProductID) FROM
@Products
-- Phân trang sản phẩm
SELECT ProductID, Name, Description, Price, Image1FileName,
Image2FileName, OnDepartmentPromotion, OnCatalogPromotion
FROM @Products
WHERE RowNumber > (@PageNumber - 1) * @ProductsPerPage
AND RowNumber <= @PageNumber * @ProductsPerPage

```

- Tạo thủ tục GetProductDetails hiển thị chi tiết của một sản phẩm

```

CREATE PROCEDURE GetProductDetails
(@ProductID INT)
AS
SELECT      Name,      Description,      Price,      Image1FileName,
Image2FileName, OnDepartmentPromotion, OnCatalogPromotion
FROM Product
WHERE ProductID = @ProductID
RETURN

```

- Thêm một Web User Control mới với tên ProductsList vào thư mục UserControls.
- Trong khung Source, thêm vào đoạn mã sau, dùng để hiển thị phân trang danh sách sản phẩm.

```

<asp:Label ID="pagingLabel" runat="server" CssClass="PagingText"
Visible="false" />
 &nbsp;&nbsp;
<asp:HyperLink ID="previousLink" runat="server"
CssClass="PagingText" Visible="false">Previous</asp:HyperLink>
 &nbsp;&nbsp;
<asp:HyperLink ID="nextLink" runat="server" CssClass="PagingText">

```

```
Visible="false">Next</asp:HyperLink>
```

- Mở khung Design, kéo thả Control Datalist từ cửa sổ toolbox vào trang.
- Đổi tên ID của Datalist thành list, và đặt thuộc tính RepeatColumns thành 2 (Hiển thị 2 cột trên một dòng) và RepeatDirection là Horizontal
- Vào khung Source và chỉnh sửa mã của Datalist như sau:

```
<asp:DataList ID="list" Runat="server" RepeatColumns="2">  
  <ItemTemplate>  
    <table cellPadding="0" align="left">  
      <tr height="105">  
        <td align="center" width="110">  
          <a href='Product.aspx?ProductID=<%# Eval("ProductID")%>'>  
            <img width="100" src='ProductImages/<%#  
Eval("Image1FileName") %>' border="0"/>  
          </a>  
        </td>  
        <td Align="top" width="250">  
          <a class="ProductName" href='Product.aspx?ProductID=<%#  
Eval("ProductID")%>'>  
            <%# Eval("Name") %>  
          </a>  
          <br/>  
          <span class="ProductDescription">  
            <%# Eval("Description") %>  
            <br/><br/>  
            Price:  
          </span>  
          <span class="ProductPrice">  
            <%# Eval("Price", "{0:c}") %>  
          </span>  
        </td>  
      </tr>  
    </table>
```

```
</ItemTemplate>  
</asp:DataList>
```

- Thêm vào đoạn styles vào GiftShop.css

```
a.ProductName  
{  
    color: Red;  
    font-family: 'Trebuchet MS';  
    text-decoration: none;  
    font-weight: bold;  
    font-size: 12px;  
}  
a.ProductName:hover  
{  
    text-decoration: underline;  
}  
.ProductDescription  
{  
    color: Black;  
    font-family: Verdana, Helvetica, sans-serif;  
    font-size: 11px;  
}  
.ProductPrice  
{  
    color: Black;  
    font-family: Verdana, Helvetica, sans-serif;  
    font-weight: bold;  
    font-size: 11px;  
}  
.PagingText  
{  
    font-family: Verdana, Helvetica, sans-serif;  
    font-size: 11px;  
    color: Black;  
}
```

```
.ProductTitle  
{  
    color: Blue;  
    font-family: Verdana, Helvetica, sans-serif;  
    text-decoration: none;  
    font-size: 24px;  
    font-weight: bold;  
    line-height: 15px;  
}
```

- Chính sửa class ProductsList trong ProductsList.ascx.cs giống như sau:

```
using System;  
using System.Data;  
using System.Configuration;  
using System.Collections;  
using System.Web;  
using System.Web.Security;  
using System.Web.UI;  
using System.Web.UI.WebControls;  
using System.Web.UI.WebControls.WebParts;  
using System.Web.UI.HtmlControls;  
using System.Collections.Specialized;  
public partial class ProductsList : System.Web.UI.UserControl  
{  
    protected void Page_Load(object sender, EventArgs e)  
    {  
        PopulateControls();  
    }  
    private void PopulateControls()  
    {  
        // Nhận DepartmentID từ query string  
        string departmentId = Request.QueryString["DepartmentID"];  
        // Nhận CategoryID từ query string
```

```

string categoryId = Request.QueryString["CategoryID"];
// nhận Page từ query string
string page = Request.QueryString["Page"];
if (page == null) page = "1";
// Sản phẩm có bao nhiêu trang?
int howManyPages = 1;
// Nếu trình duyệt hiển thị là ...
if (categoryId != null)
{
    // Danh sách sản phẩm trong một danh mục
    list.DataSource =
        CatalogAccess.GetProductsInCategory(categoryId, page, out
        howManyPages);
    list.DataBind();
}
else if (departmentId != null)
{
    // danh sách sản phẩm tiêu biểu trong gian hàng
    list.DataSource =
        CatalogAccess.GetProductsOnDepartmentPromotion(departmentId,
        page, out howManyPages);
    list.DataBind();
}
else
{
    // Danh sách sản phẩm đặc biệt trong danh mục
    list.DataSource =
        CatalogAccess.GetProductsOnCatalogPromotion(page, out
        howManyPages);
    list.DataBind();
}
// hiển thị trang
if (howManyPages > 1)
{

```

```

// Trang hiện tại
int currentPage = Int32.Parse(page);
// làm ẩn điều khiển
pagingLabel.Visible = true;
previousLink.Visible = true;
nextLink.Visible = true;
// đặt chuỗi hiển thị trang
pagingLabel.Text = "Trang " + page + " của " +
howManyPages.ToString();
// tạo một liên kết trang trước
if (currentPage == 1)
    previousLink.Enabled = false;
else
{
    NameValueCollection query = Request.QueryString;
    string paramName, newQueryString = "?";
    for (int i = 0; i < query.Count; i++)
        if (query.AllKeys[i] != null)
            if ((paramName = query.AllKeys[i].ToString()).ToUpper() != "PAGE")
                newQueryString += paramName + "=" + query[i] + "&";
    previousLink.NavigateUrl = Request.Url.AbsolutePath +
newQueryString + "Page=" + (currentPage - 1).ToString();
}
// Liên kết trang tiếp theo
if (currentPage == howManyPages)
    nextLink.Enabled = false;
else
{
    NameValueCollection query = Request.QueryString;
    string paramName, newQueryString = "?";
    for (int i = 0; i < query.Count; i++)
        if (query.AllKeys[i] != null)
            if ((paramName = query.AllKeys[i].ToString()).ToUpper() != "PAGE")
                newQueryString += paramName + "=" + query[i] + "&";
    nextLink.NavigateUrl = Request.Url.AbsolutePath +
newQueryString + "Page=" + (currentPage + 1).ToString();
}

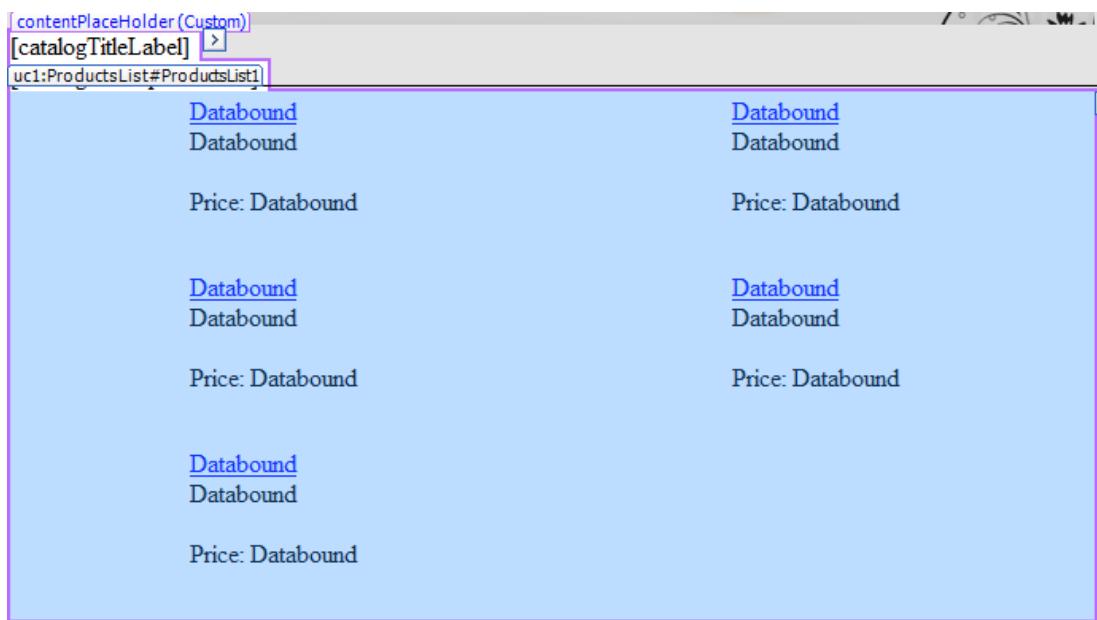
```

```

    "PAGE")
        newQueryString += paramName + "=" + query[i] + "&";
        nextLink.NavigateUrl = Request.Url.AbsolutePath +
        newQueryString + "Page=" + (currentPage + 1).ToString();
    }
}
}
}
}

```

- Tạo trang Catalog.aspx trong dự án web. Mở trang Catalog.aspx trong Design. Kéo thả ProductsList.ascx từ Solution Explorer vào.



Hình 3.45: Thêm danh sách ProductList vào trang Catalog.aspx

- Đôi với trang Default.aspx chúng ta làm tương tự.
- Nhấn F5 để thực thi dự án, được kết quả sau:



Hình 3.46: Kết quả hiển thị trang

5.3.7. Thiết kế menu hiển thị danh mục sản phẩm

Tạo một Web User Control hiển thị danh sách danh mục. Trước khi tạo ta viết một thủ tục SQL liệt kê danh sách các danh mục trong gian hàng.

```

CREATE PROCEDURE GetCategoriesInDepartment
(@DepartmentID INT)
AS
SELECT CategoryID, Name, Description
FROM Category
WHERE DepartmentID = @DepartmentID
  
```

- Tạo một Web User Control trong thư mục UserControls. Trong Solution Explorer, phải chuột vào thư mục UserControls, chọn Add New Item, chọn mẫu Web User Control và đặt tên là CategoriesList.ascx đánh dấu vào mục **Place code in separate file** Sau đó kích Add.

- Kéo thả Control DataList từ toolbox vào.
- Thiết đặt thuộc tính cho đối tượng DataList.

Tên thuộc tính	Giá trị
(ID)	List
Width	200px
CssClass	CategoryListContent
HeaderStyle-CssClass	CategoryListHead

- Ở khung Design, Chuột phải vào DataList và chọn Edit Template/Header And Footer Templates, chọn một mục trong Header Template.
- Chuột phải vào DataList và chọn Edit Template/Item templates. Thêm một Control Hyperlink từ tab Standard của thanh toolbox tới Item Template. Đặt thuộc tính Text của HyperLink là rỗng.
- Trong khung Source, mã được sinh tự động bởi Visual Studio như sau:

```
<asp:DataList ID="list" runat="server"
CssClass="CategoryListContent" Width="200px">
<ItemTemplate>
    &nbsp;&raquo;
    <asp:HyperLink
        ID="HyperLink1"
        Runat="server">
    </asp:HyperLink>
    &nbsp;&laquo;
</ItemTemplate>
<HeaderTemplate>
    Choose a Category
</HeaderTemplate>
<HeaderStyle CssClass="CategoryListHead" />
</asp:DataList>
```

- Chính sửa mã phần <ItemTemplate> như dưới đây:

```
<asp:DataList ID="list" runat="server"
CssClass="CategoryListContent" Width="200px">
<ItemTemplate>
    &nbsp;&raquo;
    <asp:HyperLink
        ID="HyperLink1"
```

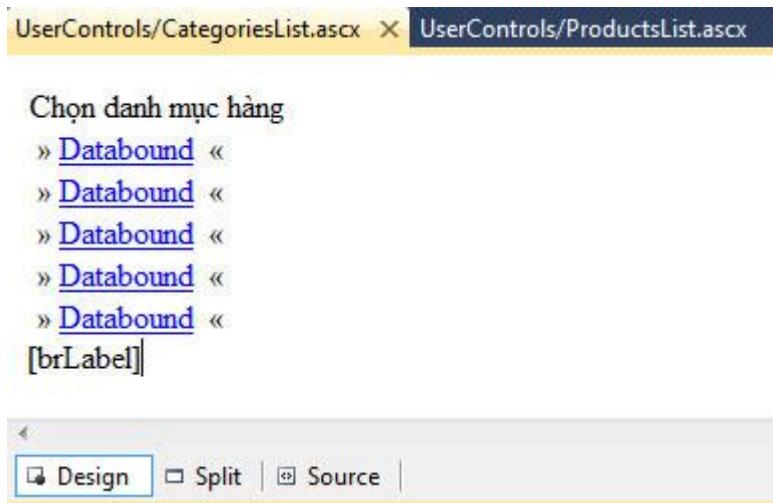
```

Runat="server"
NavigateUrl='<%# "../Catalog.aspx?DepartmentID=" +
Request.QueryString["DepartmentID"] + "&CategoryID=" +
Eval("CategoryID") %>' 
Text='<%# Eval("Name") %>' 
ToolTip='<%# Eval("Description") %>' 
CssClass='<%# Eval("CategoryID").ToString() ==
Request.QueryString["CategoryID"] ? "CategorySelected" :
"CategoryUnselected" %>'>>
</asp:HyperLink>
 &nbsp;&laquo;
</ItemTemplate>
<HeaderTemplate>
    Chọn danh mục hàng
</HeaderTemplate>
<HeaderStyle CssClass="CategoryListHead" />
</asp:DataList>

```

- Thêm ở phía sau DataList một control Label với thuộc tính text rỗng, khi liệt kê danh sách danh mục thì đặt thuộc tính text của Label là một thẻ
 cho việc trình bày được đẹp hơn.

- Vào khung Design ta được một giao diện như sau:



Hình 3.47: Cửa sổ Design trang CategoriesList.ascx

- Thêm vào file GiftShop.css đoạn styles sau:

```

.CategoryListHead
{

```

```
border-right: #ea6d00 1px solid;
border-top: #ea6d00 1px solid;
border-left: #ea6d00 1px solid;
border-bottom: #ea6d00 1px solid;
background-color: #ef8d0e;
font-family: Verdana, Arial;
font-weight: bold;
font-size: 10pt;
color: #f5f5dc;
text-align: center;
}

.CategoryListContent
{
border-right: #ea6d00 1px solid;
border-top: #ea6d00 1px solid;
border-left: #ea6d00 1px solid;
border-bottom: #ea6d00 1px solid;
background-color: #f8c78c;
text-align: center;
}

.a.CategoryUnselected
{
font-family: Verdana, Arial;
font-weight: bold;
font-size: 9pt;
color: #cd853f;
line-height: 25px;
padding-right: 5px;
padding-left: 5px;
text-decoration: none
}

.a.CategoryUnselected:hover
{
color: #d2691e;
```

```

padding-right: 5px;
padding-left: 5px
}
a.CategorySelected
{
font-family: Verdana, Arial;
font-weight: bold;
font-size: 9pt;
color: #a0522d;
line-height: 25px;
padding-right: 5px;
padding-left: 5px;
text-decoration: none
}

```

- Bây giờ mở code be-hind của file User Control (CategoriesList.ascx) và sửa sự kiện Page_Load như dưới đây:

```

// Lấy ID của gian hàng được chọn
string departmentId = Request.QueryString["DepartmentID"];
// Tiếp tục chỉ kiểm tra xem DepartmentID có tồn tại trong query
string
if (departmentId != null)
{
    // Trả về danh sách danh mục trong gian hàng đưa vào đối tượng
    DataTable (lấy từ thủ tục GetCategoriesInDepartment)
    // Dữ liệu của danh mục được hiển thị bởi DataList
    list.DataSource =
CatalogAccess.GetCategoriesInDepartment(departmentId);
    // Ràng buộc dữ liệu với control
    list.DataBind();
    // Đưa vào một thẻ xuống dòng cho control Label
    brLabel.Text = "<br />";
}

```

- Mở file GiftShop.Master trong khung Design. Kéo thả

CategoriesList.ascx từ Solution Explorer đưa nó vào gần đoạn văn bản “List of Categories”. Xoá dòng văn bản đó đi chỉ để lại Control này.

- Thực thi Dự án, chọn một gian hàng (Department) và chọn một danh mục hàng. Hình dưới đây thể hiện điều đó:



Hình 3.48: Danh sách sản phẩm của danh mục sản phẩm được chọn
5.3.8. Thiết kế trang tìm kiếm sản phẩm.

Trong phần này ta sẽ thiết kế cho phép người dùng có thể tìm kiếm theo từng từ hoặc nhiều từ. chúng ta thiết kế khung tìm kiếm giống hình ảnh sau:

Tìm kiếm sản phẩm	
<input type="text"/>	<input type="button" value="Tìm"/>
<input type="checkbox"/> Tìm tất cả các từ	

Hình 3.49: Hộp tìm kiếm

Để việc tìm kiếm được tốt ta cần viết một thủ tục (Store Procedure) cho việc tìm kiếm sản phẩm. Trước khi viết thủ tục ta viết một hàm tìm kiếm một từ trong một cụm từ.

- Trong SQL server 2008, mở cơ sở dữ liệu Shopgifts/

Programmability/ Functions/ phải chuột vào Scalar-valued Functions chọn New Scalar-valued Functions. Viết vào đoạn mã sau:

```
CREATE FUNCTION dbo.WordCount
(@Word VARCHAR(15),
@Phrase VARCHAR(1000))
RETURNS SMALLINT
AS
BEGIN
/* Nếu một từ (@Word) hoặc cụm từ (@Phrase) rỗng thì hàm trả về 0*/
IF @Word IS NULL OR @Phrase IS NULL RETURN 0
/* Khai báo BiggerWord là một từ dài hơn Word*/
DECLARE @BiggerWord VARCHAR(21)
SELECT @BiggerWord = @Word + 'x'
/* Thay thế Word thành BiggerWord trong cụm từ Phrase */
DECLARE @BiggerPhrase VARCHAR(2000)
SELECT @BiggerPhrase = REPLACE (@Phrase, @Word,
@BiggerWord)
RETURN LEN(@BiggerPhrase) - LEN(@Phrase)
END
```

- Bây giờ chúng ta viết một thủ tục (stored procedure) SearchCatalog. Thủ tục này sẽ sử dụng hàm WordCount để tính toán giá trị tìm kiêm. Sử dụng các bước đã biết , thêm thủ tục này vào nơi chứa cơ sở dữ liệu ShopGifts.

```
CREATE PROCEDURE SearchCatalog
(@DescriptionLength INT,
@PageNumber TINYINT,
@ProductsPerPage TINYINT,
@HowManyResults SMALLINT OUTPUT,
@AllWords BIT,
@Word1 NVARCHAR(15) = NULL,
@Word2 NVARCHAR(15) = NULL,
@Word3 NVARCHAR(15) = NULL,
@Word4 NVARCHAR(15) = NULL,
@Word5 NVARCHAR(15) = NULL)
```

```

AS
/* Tạo một bảng mới sẽ chứa kết quả tìm kiếm */
DECLARE @Products TABLE
(RowNumber SMALLINT IDENTITY (1,1) NOT NULL,
 ProductID INT,
 Name NVARCHAR(50),
 Description NVARCHAR(1000),
 Price MONEY,
 Image1FileName VARCHAR(50),
 Image2FileName VARCHAR(50),
 Rank INT)
/* Tìm từng từ */
IF @AllWords = 0
    INSERT INTO @Products
    SELECT ProductID, Name,
        SUBSTRING>Description, 1, @DescriptionLength) + '...' AS
Description,
        Price, Image1FileName, Image2FileName,
        3 * dbo.WordCount(@Word1, Name) +
dbo.WordCount(@Word1, Description) +
        3 * dbo.WordCount(@Word2, Name) +
dbo.WordCount(@Word2, Description) +
        3 * dbo.WordCount(@Word3, Name) +
dbo.WordCount(@Word3, Description) +
        3 * dbo.WordCount(@Word4, Name) +
dbo.WordCount(@Word4, Description) +
        3 * dbo.WordCount(@Word5, Name) +
dbo.WordCount(@Word5, Description)
        AS Rank
    FROM Product
    ORDER BY Rank DESC
/* Tìm toàn bộ từ */
IF @AllWords = 1
    INSERT INTO @Products

```

```

SELECT ProductID, Name,
       SUBSTRING(Description, 1, @DescriptionLength) + '...' AS
Description,
       Price, Image1FileName, Image2FileName,
       (3 * dbo.WordCount(@Word1, Name) +
dbo.WordCount(@Word1, Description)) *
CASE
    WHEN @Word2 IS NULL THEN 1
    ELSE 3 * dbo.WordCount(@Word2, Name) +
dbo.WordCount(@Word2, Description)
END *
CASE
    WHEN @Word3 IS NULL THEN 1
    ELSE 3 * dbo.WordCount(@Word3, Name) +
dbo.WordCount(@Word3, Description)
END *
CASE
    WHEN @Word4 IS NULL THEN 1
    ELSE 3 * dbo.WordCount(@Word4, Name) +
dbo.WordCount(@Word4, Description)
END *
CASE
    WHEN @Word5 IS NULL THEN 1
    ELSE 3 * dbo.WordCount(@Word5, Name) +
dbo.WordCount(@Word5, Description)
END
AS Rank
FROM Product
ORDER BY Rank DESC
/* Ghi lại số sản phẩm tìm được */
SELECT @HowManyResults = COUNT(*)
FROM @Products
WHERE Rank > 0
/* Lấy những sản phẩm tìm được, phân trang hiển thị trên trang web */

```

```

SELECT ProductID, Name, Description, Price, Image1FileName,
Image2FileName, Rank
FROM @Products
WHERE Rank > 0
    AND RowNumber BETWEEN (@PageNumber-1) *
@ProductsPerPage + 1
        AND @PageNumber * @ProductsPerPage
ORDER BY Rank DESC

```

Đây là một thủ tục khá phức tạp, trước hết chúng ta cần biết các tham số của nó:

- + `@DescriptionLength` là chiều dài tối đa của mô tả sản phẩm.
- + `@PageNumber` quy định cụ thể trang kết quả tìm kiếm.
- + `@ProductsPerPage` xác định có bao nhiêu sản phẩm được trả về.
- + `@HowManyResults` là một tham số đầu ra, mà ta sẽ thiết lập tổng số kết quả tìm kiếm được, để tính toán số lượng các trang tìm kiếm.
- + `@AllWord` là một tham số đầu vào là bit, xác định xem bạn tìm toàn bộ từ hay bất kỳ từ nào được nhập.
- + `@Word1` tới `@Word5` là những từ được tìm kiếm. Tất cả đều có giá trị mặc định là Null.
- Tiếp theo chúng ta cần thêm vào class CatalogAccess một phương thức **Search** được sử dụng làm nhiệm vụ tìm kiếm sản phẩm:

```

// tìm kiếm sản phẩm
public static DataTable Search(string searchString, string allWords,
string pageNumber, out int howManyPages)
{
    // Tạo đối tượng command
    DbCommand comm = GenericDataAccess.CreateCommand();
    // Sử dụng thủ tục SearchCatalog để làm việc
    comm.CommandText = "SearchCatalog";
    // Tạo một tham số mới
    DbParameter param = comm.CreateParameter();
    param.ParameterName = "@DescriptionLength";
    param.Value = ShopgiftConfiguration.ProductDescriptionLength;
    param.DbType = DbType.Int32;
}

```

```

comm.Parameters.Add(param);
// Tạo một tham số mới
param = comm.CreateParameter();
param.ParameterName = "@AllWords";
param.Value = allWords.ToUpper() == "TRUE" ? "True" : "False";
param.DbType = DbType.Boolean;
comm.Parameters.Add(param);
// Tạo một tham số mới
param = comm.CreateParameter();
param.ParameterName = "@PageNumber";
param.Value = pageNumber;
param.DbType = DbType.Int32;
comm.Parameters.Add(param);
// Tạo một tham số mới
param = comm.CreateParameter();
param.ParameterName = "@ProductsPerPage";
param.Value = ShopgiftConfiguration.ProductsPerPage;
param.DbType = DbType.Int32;
comm.Parameters.Add(param);
// Tạo một tham số mới
param = comm.CreateParameter();
param.ParameterName = "@HowManyResults";
param.Direction = ParameterDirection.Output;
param.DbType = DbType.Int32;
comm.Parameters.Add(param);
// định nghĩa số từ lớn nhất
int howManyWords = 5;
// biến đổi chuỗi tìm kiếm vào một mảng từ đặc biệt
char[] wordSeparators = new char[] { ',', ';', '.', '!', '?', '-', '' };
string[] words = searchString.Split(wordSeparators,
StringSplitOptions.RemoveEmptyEntries);
int index = 1;
// thêm vào những từ và đưa vào tham số trong Procedure
for (int i = 0; i <= words.GetUpperBound(0) && index <=

```

```

howManyWords; i++)
    // từ ngắn
    if (words[i].Length > 2)
    {
        // tạo một @Word parameters
        param = comm.CreateParameter();
        param.ParameterName = "@Word" + index.ToString();
        param.Value = words[i];
        param.DbType = DbType.String;
        comm.Parameters.Add(param);
        index++;
    }
    // Thực thi và trả về kết quả là một DataTable
    DataTable table =
GenericDataAccess.ExecuteSelectCommand(comm);
    int howManyProducts =
Int32.Parse(comm.Parameters["@HowManyResults"].Value.ToString())
;
    howManyPages = (int)Math.Ceiling((double)howManyProducts /
        (double)ShopgiftConfiguration.ProductsPerPage);
    return table;
}

```

- **Tạo hộp tìm kiếm**

- + Đầu tiên, viết thêm một lớp **Utilities** (tạo một tệp class có tên Utilities.cs trong thư mục App_Code) lớp này dùng để xử lý các chức năng mở rộng của website.
- + Thêm các namespace vào trong lớp Utilities.cs

```

using System;
using System.Net.Mail;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;

```

- + Thêm phương thức sau vào lớp Utilities

```

public static void TieButton(Page page, Control TextBoxToTie, Control
ButtonToTie)
{
    // tạo jscript
    string jsString = "";
    // Kiểm tra loại nút và nhận yêu cầu từ jscript
    if (ButtonToTie is LinkButton)
    {
        jsString = "if ((event.which && event.which == 13) ||
(event.keyCode && event.keyCode == 13)) {" +
            page.ClientScript.GetPostBackEventReference(ButtonToTie,
            "").Replace(":", "$") + ";return false;} else return true;";
    }
    else if (ButtonToTie is ImageButton)
    {
        jsString = "if ((event.which && event.which == 13) ||
(event.keyCode && event.keyCode == 13)) {" +
            page.ClientScript.GetPostBackEventReference(ButtonToTie,
            "").Replace(":", "$") + ";return false;} else return true;";
    }
    else
    {
        jsString = "if ((event.which && event.which == 13) ||
(event.keyCode && event.keyCode == 13)) {document."
            + "forms[0].elements['" + ButtonToTie.UniqueID.Replace(":", "_") + "'].Click();return false;} else return true; ";
    }
    // đính kèm thuộc tính onkeydown – Phục vụ cho HtmlControl hoặc
    WebControl
    if (TextBoxToTie is HtmlControl)
    {
        ((HtmlControl)TextBoxToTie).Attributes.Add("onkeydown",
jsString);
    }
}

```

```

else if (TextBoxToTie is WebControl)
{
    ((WebControl)TextBoxToTie).Attributes.Add("onkeydown",
jsString);
}
}

```

+ Đầu tiên tạo một Web User Control với tên SearchBox.ascx trong thư mục UserControls.

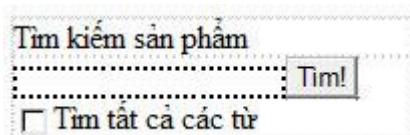
+ Trong cửa sổ Design ta thêm vào đoạn mã HTML sau:

```

<table border="0" cellpadding="0" cellspacing="0" width="200px">
<tr>
<td class="SearchBoxHead">
    Tìm kiếm sản phẩm
</td>
</tr>
<tr>
<td class="SearchBoxContent">
    <asp:TextBox ID="searchTextBox" Runat="server" Width="128px"
    CssClass="SearchBox" BorderStyle="Dotted" MaxLength="100"
    Height="16px" />
    <asp:Button ID="goButton" Runat="server" CssClass="SearchBox"
    Text="Tìm!" Width="36px" Height="21px" OnClick="goButton_Click"
    /><br />
    <asp:CheckBox ID="allWordsCheckBox" CssClass="SearchBox"
    Runat="server" Text="Tìm tất cả các từ" />
</td>
</tr>
</table>

```

+ Chuyển sang cửa sổ Design ta được giao diện giống hình sau:



Hình 3.50: SearchBox.ascx trong cửa sổ Design

+ Thêm đoạn mã styles vào GiftShop.css:

```
.SearchBoxHead
{
    border-right: #0468a4 1px solid;
    border-top: #0468a4 1px solid;
    border-bottom: #0468a4 1px solid;
    border-left: #0468a4 1px solid;
    background-color: #0583b5;
    font-family: Verdana, Arial;
    font-weight: bold;
    font-size: 10pt;
    color: #f5f5dc;
    text-align: center;
}

.SearchBoxContent
{
    border-right: #0468a4 1px solid;
    border-top: #0468a4 1px solid;
    border-left: #0468a4 1px solid;
    border-bottom: #0468a4 1px solid;
    background-color: #8bc8dd;
    font-family: Arial, Verdana;
    font-size: 9pt;
    color: darkblue;
    padding-top: 5px;
    padding-left: 12px;
    padding-bottom: 5px;
}

.SearchBox
{
    font-family: Verdana, Helvetica, sans-serif;
    font-size: 9pt;
    margin-bottom: 5px;
}
```

- + Mở code-behind của tệp SearchBox.ascx.cs, thêm đoạn mã sau vào sự kiện Page_Load:

```

if (!IsPostBack)
{
    // Hộp văn bản tìm kiếm nút nhấn Go
    Utilities.TieButton(this.Page, searchTextBox, goButton);
    // tải giá trị trong hộp văn bản tìm kiếm
    string allWords = Request.QueryString["AllWords"];
    string searchString = Request.QueryString["Search"];
    if (allWords != null)
        allWordsCheckBox.Checked = (allWords.ToUpper() == "TRUE");
    if (searchString != null)
        searchTextBox.Text = searchString;
}

```

- + Chuyển sang cửa sổ Design, kích đúp chuột nút Go nó sẽ tự động sinh ra sự kiện Click, và ta thêm đoạn mã như sau:

```

// Thực hiện việc tìm kiếm sản phẩm
protected void goButton_Click(object sender, EventArgs e)
{
    ExecuteSearch();
}

// phương thức chuyển kết quả đến trang tìm kiếm
private void ExecuteSearch()
{
    if (searchTextBox.Text.Trim() != "")
        Response.Redirect(Request.ApplicationPath +
            "/Search.aspx?Search=" + searchTextBox.Text +
            "&AllWords=" + allWordsCheckBox.Checked.ToString());
}

```

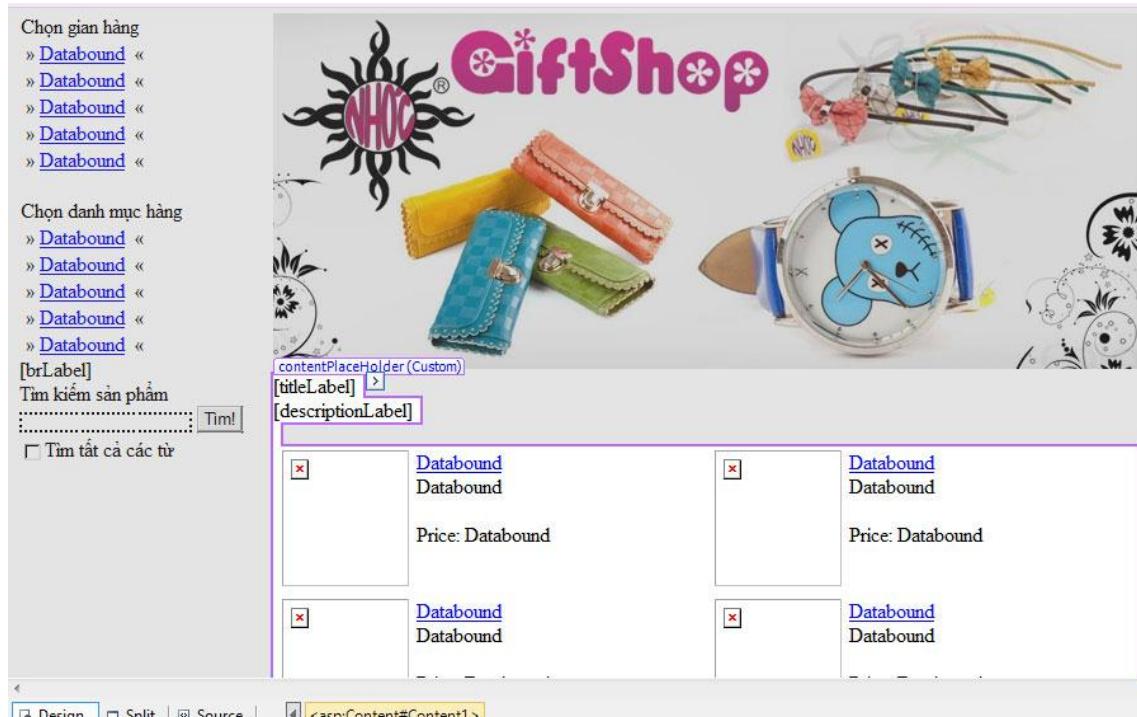
- + Trong giao diện của trang GiftShop.master, thực hiện việc kéo thả SearchBox.ascx trong Solution Explorer dưa vào phía dưới điều khiển CategoriesList. (Xem hình sau)



Hình 3.51. Thêm hộp tìm kiếm vào trong GiftShop.master

- Tạo trang hiển thị kết quả Search.aspx

- + Tạo một Web Form mới trong thư mục gốc GiftShop. Phải chuột vào GiftShop trong Solution Explorer và chọn Add New Item. Hộp thoại Add New xuất hiện, đặt tên cho file là Search.aspx, sau đó kích Add.
- + Trong hộp thoại được mở, bạn chọn GiftShop.master Master Page và Kích add.
- + Trong cửa sổ Design của tệp Search.aspx, thêm vào hai điều khiển Label (đặt tên lần lượt cho hai control này là: titleLabel và descriptionLabel), kéo thả ProductsList.ascx từ Solution Explorer vào vùng Content, hiển thị trong hình dưới đây:



Hình 3.52: Cửa sổ Design trong Search.aspx

- + Thiết đặt thuộc tính CssClass cho hai control Label lần lượt là CatalogTitle và CatalogDescription. Chuyển sang cửa sổ Design ta được đoạn mã HTML giống dưới đây:

```
<%@ Page Language="C#" MasterPageFile="~/GiftShop.master"
AutoEventWireup="true" CodeFile="Search.aspx.cs" Inherits="Search"
Title="Untitled Page" %>
<%@ Register Src="UserControls/ProductsList.ascx"
TagName="ProductsList" TagPrefix="uc1" %>
<asp:Content ID="Content1"
ContentPlaceHolderID="contentPlaceHolder" Runat="Server">
<asp:Label ID="titleLabel" runat="server"
CssClass="CatalogTitle"></asp:Label><br />
<asp:Label ID="descriptionLabel" runat="server"
CssClass="CatalogDescription"></asp:Label><br /><br />
<uc1:productslist id="ProductsList1"
runat="server"></uc1:productslist>
</asp:Content>
```

- + Tiếp theo vào code-behind ta thêm vào sự kiện Page_Load đoạn mã giống như sau:

```
public partial class Search : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        // Đưa nội dung vào Table (Trong trang ProductsList.ascx)
        string searchString = Request.QueryString["Search"];
        titleLabel.Text = "Product Search";
        descriptionLabel.Text = "You searched for <font color=/red/>" +
        searchString + "</font>.";
        // Đặt tiêu đề của trang
        this.Title = BalloonShopConfiguration.SiteName +
                    " : Product Search : " + searchString;
    }
}
```

- + Nhấn F5 để chạy dự án ta được kết quả như hình dưới đây:



Hình 3.53. Tìm kiếm từ “Hoa hồng”

5.4. Thiết kế giỏ hàng

5.4.1. Thực hiện tầng dữ liệu

Phần này bạn cần thêm các thủ tục sau đây vào cơ sở dữ liệu ShopGifts:

- ShoppingCartAddItem thêm một sản phẩm vào giỏ hàng.
- ShoppingCartRemoveItem xoá một sản phẩm ở giỏ hàng.
- ShoppingCartUpdateItem cập nhật giỏ hàng.
- ShoppingCartGetItems lấy danh sách các sản phẩm trong giỏ hàng.
- ShoppingCartGetTotalAmount trả về tổng giá những sản phẩm trong giỏ hàng.
- **Thủ tục ShoppingCartAddItem**

Shopping CartAddItem được gọi khi khách hàng kích vào nút “Add to Cart” chọn một sản phẩm. Nếu khách hàng chọn một sản phẩm đã tồn tại trong giỏ hàng, thì số lượng sẽ tăng lên một; nếu sản phẩm chưa tồn tại thì một sản phẩm mới sẽ được thêm vào.

```

CREATE PROCEDURE ShoppingCartAddItem
(@CartID char(36),
 @ProductID INT)
AS
IF EXISTS

```

```

(SELECT CartID
FROM ShoppingCart
WHERE ProductID = @ProductID AND CartID = @CartID)

UPDATE ShoppingCart
SET Quantity = Quantity + 1
WHERE ProductID = @ProductID AND CartID = @CartID

ELSE
IF EXISTS (SELECT Name FROM Product WHERE
ProductID=@ProductID)
    INSERT INTO ShoppingCart (CartID, ProductID, Quantity,
DateAdded)
    VALUES (@CartID, @ProductID, 1, GETDATE())

```

- **Thủ tục ShoppingCartRemoveItem.**

Đây là thủ tục loại bỏ một sản phẩm khỏi giỏ hàng. Điều này xảy ra khi khách nhấp chuột vào nút Remove đối với một sản phẩm có trong giỏ hàng. Thêm thủ tục **ShoppingCartRemoveItem** vào cơ sở dữ liệu ShopGifts:

```

CREATE PROCEDURE ShoppingCartRemoveItem
(@CartID char(36),
@ProductID INT)
AS
DELETE FROM ShoppingCart
WHERE CartID = @CartID and ProductID = @ProductID

```

- **Thủ tục ShoppingCartUpdateItem**

ShoppingCartUpdateItem được sử dụng khi bạn muốn cập nhật số lượng của một sản phẩm hiện có trong giỏ hàng, thủ tục này sẽ nhận ba giá trị là ba tham số @CartID, @ProductID, và @Quantity.

Nếu @Quantity bằng 0 hoặc nhỏ hơn, ShoppingCartUpdateItem sẽ gọi ShoppingCartRemoveItem để loại bỏ các sản phẩm từ giỏ hàng. Nếu không, nó cập nhật số lượng và cập nhật thời gian thêm sản phẩm (DateAdded).

```

CREATE Procedure ShoppingCartUpdateItem
(@CartID char(36),
 @ProductID INT,
 @Quantity INT)
As
IF @Quantity <= 0
    EXEC ShoppingCartRemoveItem @CartID, @ProductID
ELSE
    UPDATE ShoppingCart
    SET Quantity = @Quantity, DateAdded = GETDATE()
    WHERE ProductID = @ProductID AND CartID = @CartID

```

- **Thủ tục ShoppingCartGetItems.**

Thủ tục này sẽ lấy về các trường ID, Name, Price, Quantity, và SubTotal cho mỗi sản phẩm trong giỏ hàng. Bởi vì bảng ShoppingCart chỉ lưu trữ các ProductID cho mỗi sản phẩm nó lưu trữ bạn cần nối bảng ShoppingCart với bảng Product để lấy thông tin bạn cần.

```

CREATE PROCEDURE ShoppingCartGetItems
(@CartID char(36))
AS
SELECT Product.ProductID, Product.Name, Product.Price,
ShoppingCart.Quantity,
        Product.Price * ShoppingCart.Quantity AS Subtotal
FROM ShoppingCart INNER JOIN Product
ON ShoppingCart.ProductID = Product.ProductID
WHERE ShoppingCart.CartID = @CartID

```

- **Thủ tục ShoppingCartGetTotalAmount.**

ShoppingCartGetTotalAmount trả về tổng giá trị của các sản phẩm trong giỏ hàng. Nó sẽ được gọi khi hiển thị tổng số tiền cho giỏ hàng.

```

CREATE PROCEDURE ShoppingCartGetTotalAmount
(@CartID char(36))
AS
SELECT ISNULL(SUM(Product.Price * ShoppingCart.Quantity), 0)
FROM ShoppingCart INNER JOIN Product

```

```
ON ShoppingCart.ProductID = Product.ProductID  
WHERE ShoppingCart.CartID = @CartID
```

5.4.2. Tàng xử lý dữ liệu.

Bạn sẽ viết các phương thức xử lý dữ liệu cho việc xử lý giỏ hàng trong một lớp riêng biệt có tên ShoppingCartAccess.

Lớp ShoppingCartAccess hỗ trợ các chức năng cần thiết cho việc xử lý một giỏ hàng và sẽ gọi các thủ tục mà bạn đã viết trước đó.

Chuẩn bị cho việc viết lớp ShoppingCartAccess.

- Lưu trữ số ngày hết hạn của cookie trong tập tin Web.config. Mở tập tin Web.config và thêm vào các thiết lập sau:

```
<appSettings>  
  <add key="CartPersistDays" value="10" />  
  ...
```

- Thay đổi trong lớp BalloonShopConfiguration. Bằng cách thêm một thuộc tính cartPersistDays(số ngày hết hạn mua hàng) trong BalloonShopConfiguration.cs.

```
// Lưu trữ số ngày hết hạn của giỏ mua hàng  
private readonly static int cartPersistDays;
```

- Thêm vào một dòng mã để lấy số ngày trong tập tin Web.config

```
static BalloonShopConfiguration()  
{  
    cartPersistDays =  
    Int32.Parse(ConfigurationManager.AppSettings["CartPersistDays"]);  
    ...
```

- Thêm vào một thuộc tính public trả về giá trị cấu hình

```
// Trả về số ngày hết hạn của giỏ mua hàng  
public static int CartPersistDays  
{  
    get  
    {  
        return cartPersistDays;  
    }
```

```
    }  
}
```

- Bây giờ bắt đầu viết mã thực vào, phải chuột vào thư mục App_Code trong Solution Explorer và chọn Add New Item.
 - Chọn class từ Templates window và đặt tên là ShoppingCartAccess.cs.
 - Khi tập tin được bật ra cần thêm không gian tên System.Data.Common và viết vào thuộc tính shoppingCartId như dưới đây:

```
// Trả về ID giỏ mua hàng cho người dùng hiện tại  
private static string shoppingCartId  
{  
    get  
    {  
        HttpContext context = HttpContext.Current;  
        // lấy ID từ đối tượng session người  
        string cartId = "";  
        object cartIdSession = context.Session["BalloonShop_CartID"];  
        if (cartIdSession != null)  
            cartId = cartIdSession.ToString();  
        // Nếu ID đã tồn tại...  
        if (cartId != "")  
            // trả về giá trị cho nó  
            return cartId;  
        else  
            // Nếu ID chưa tồn tại...  
            {  
                // Kiểm tra xem ID giỏ hàng đã tồn tại là một cookie hay không  
                if (context.Request.Cookies["BalloonShop_CartID"] != null)  
                {  
                    // nếu giỏ hàng đã tồn tại như là một cookie, sử dụng cookie làm  
                    giá trị
```

```

cartId = context.Request.Cookies["BalloonShop_CartID"].Value;
    // ghi lại id vào session
    context.Session["BalloonShop_CartID"] = cartId;
    // trả lại id
    return cartId;
}
else
    // if the cart ID doesn't exist in the cookie as well, generate a new
ID
{
    // generate a new GUID
    cartId = Guid.NewGuid().ToString();
    // create the cookie object and set its value
    HttpCookie cookie = new HttpCookie("BalloonShop_CartID",
    cartId.ToString());
    // set the cookie's expiration date
    int howManyDays = BalloonShopConfiguration.CartPersistDays;
    DateTime currentDate = DateTime.Now;
    TimeSpan timeSpan = new TimeSpan(howManyDays, 0, 0, 0);
    DateTime expirationDate = currentDate.Add(timeSpan);
    cookie.Expires = expirationDate;
    // set the cookie on client's browser
    context.Response.Cookies.Add(cookie);
    // save the ID to the Session as well
    context.Session["BalloonShop_CartID"] = cartId;
    // return the CartID
    return cartId.ToString();
}
}
}
}

```

❖ Thực hiện các chức năng của giỏ hàng.

Bạn có năm phương thức trong lớp xử lý dữ liệu, ứng với năm thủ tục mà bạn đã viết trước đó. Thêm các phương thức vào class

ShoppingCartAccess.

+ **Phương thức AddItem.**

AddItem gọi thủ tục ShoppingCartItem. Nếu sản phẩm đã tồn tại trong các giỏ hàng, số lượng của nó được tăng thêm một. Nếu không, sản phẩm được bổ sung thêm một.

```
// thêm một sản phẩm mới vào giỏ hàng
public static bool AddItem(string productId)
{
    DbCommand comm = GenericDataAccess.CreateCommand();
    // lấy tên thủ tục
    comm.CommandText = "ShoppingCartAddItem";
    // tạo một tham số mới
    DbParameter param = comm.CreateParameter();
    param.ParameterName = "@CartID";
    param.Value = shoppingCartId;
    param.DbType = DbType.String;
    param.Size = 36;
    comm.Parameters.Add(param);
    // tạo một tham số mới
    param = comm.CreateParameter();
    param.ParameterName = "@ProductID";
    param.Value = productId;
    param.DbType = DbType.Int32;
    comm.Parameters.Add(param);
    // trả về true nếu hoàn thành, false nếu lỗi
    try
    {
        // thực thi thủ tục và trả về true nếu hoàn thành, hoặc false trong
        // trường hợp
        return (GenericDataAccess.ExecuteNonQuery(comm) != -1);
    }
    catch
    {
        // lỗi
    }
}
```

```
    return false;  
}  
}
```

+ Phương thức UpdateItem.

UpdateItem gọi thủ tục ShoppingCartUpdateItem để thay đổi một sản phẩm đã tồn tại trong giỏ hàng.

```
// Cập nhật số lượng sản phẩm trong giỏ hàng  
public static bool UpdateItem(string productId, int quantity)  
{  
    DbCommand comm = GenericDataAccess.CreateCommand();  
    // lấy tên thủ tục  
    comm.CommandText = "ShoppingCartUpdateItem";  
    // tạo một tham số mới  
    DbParameter param = comm.CreateParameter();  
    param.ParameterName = "@CartID";  
    param.Value = shoppingCartId;  
    param.DbType = DbType.String;  
    param.Size = 36;  
    comm.Parameters.Add(param);  
    // tạo một tham số mới  
    param = comm.CreateParameter();  
    param.ParameterName = "@ProductID";  
    param.Value = productId;  
    param.DbType = DbType.Int32;  
    comm.Parameters.Add(param);  
    // tạo một tham số mới  
    param = comm.CreateParameter();  
    param.ParameterName = "@Quantity";  
    param.Value = quantity;  
    param.DbType = DbType.Int32;  
    comm.Parameters.Add(param);  
    // trả về true nếu hoàn thành, false nếu lỗi  
    try  
    {
```

```

    // thực thi thủ tục và trả về true nếu hoàn thành, hoặc false trong
    // trường hợp
    return (GenericDataAccess.ExecuteNonQuery(comm) != -1);
}
catch
{
    // lỗi
    return false;
}
}

```

+ Phương thức RemoveItem

RemoveItem sẽ loại bỏ một sản phẩm khỏi giỏ hàng.

```

// Loại bỏ một sản phẩm khỏi giỏ hàng
public static bool RemoveItem(string productId)
{
    DbCommand comm = GenericDataAccess.CreateCommand();
    // lấy tên thủ tục
    comm.CommandText = "ShoppingCartRemoveItem";
    // tạo một tham số mới
    DbParameter param = comm.CreateParameter();
    param.ParameterName = "@CartID";
    param.Value = shoppingCartId;
    param.DbType = DbType.String;
    param.Size = 36;
    comm.Parameters.Add(param);
    // tạo một tham số mới
    param = comm.CreateParameter();
    param.ParameterName = "@ProductID";
    param.Value = productId;
    param.DbType = DbType.Int32;
    comm.Parameters.Add(param);
    // trả về true nếu hoàn thành, false nếu lỗi
    try
    {

```

```

    // thực thi thủ tục và trả về true nếu hoàn thành, hoặc false trong
    // trường hợp
    return (GenericDataAccess.ExecuteNonQuery(comm) != -1);
}
catch
{
    // lỗi
    return false;
}
}

```

+ Phương thức GetItems

GetItems lấy ra toàn bộ các sản phẩm có trong giỏ hàng, phương thức này được gọi khi khách hàng muốn xem giỏ hàng của mình.

```

// Lấy toàn bộ sản phẩm có trong giỏ hàng
public static DataTable GetItems()
{
    DbCommand comm = GenericDataAccess.CreateCommand();
    // lấy tên thủ tục
    comm.CommandText = "ShoppingCartGetItems";
    // tạo một tham số mới
    DbParameter param = comm.CreateParameter();
    param.ParameterName = "@CartID";
    param.Value = shoppingCartId;
    param.DbType = DbType.String;
    param.Size = 36;
    comm.Parameters.Add(param);
    // trả về kết quả là table
    DataTable table =
        GenericDataAccess.ExecuteSelectCommand(comm);
    return table;
}

```

+ Phương thức GetTotalAmount.

GetTotalAmout sẽ tính toán tổng giá trị của một giỏ hàng, đúng như

với tên gọi của nó.

```
// Lấy tất cả sản phẩm có trong giỏ hàng
public static decimal GetTotalAmount()
{
    DbCommand comm = GenericDataAccess.CreateCommand();
    // lấy tên thủ tục
    comm.CommandText = "ShoppingCartGetTotalAmount";
    // tạo một tham số mới
    DbParameter param = comm.CreateParameter();
    param.ParameterName = "@CartID";
    param.Value = shoppingCartId;
    param.DbType = DbType.String;
    param.Size = 36;
    comm.Parameters.Add(param);
    // trả về kết quả là một table
    return Decimal.Parse(GenericDataAccess.ExecuteScalar(comm));
}
```

5.4.3. Tầng xử lý giao diện người dùng

Các chức năng đã được viết, bây giờ bạn có thể xây dựng giao diện người dùng cho các chức năng giỏ mua hàng bao gồm các bước chủ yếu sau đây:

- Tạo nút bấm thêm sản phẩm vào giỏ hàng (Add to Cart).
- Hiển thị thông tin tóm tắt của giỏ hàng.
- Tạo trang giỏ mua hàng.
- Cho phép người dùng cập nhật giỏ hàng.
- Thực hiện chức năng tiếp tục mua hàng “Continue Shopping”.

❖ **Tạo nút bấm Add to Cart.**

- Thêm đoạn styles sau vào GiftShop.css, bạn sẽ sử dụng nó cho nút bấm Add to Cart.

```
.SmallButtonText
{
    color: Black;
    font-family: Verdana, Helvetica, sans-serif;
```

```
    font-size: 10px;  
}
```

- Thêm một nút nhấn vào ItemTemplate của DataList trong trang ProductList.ascx, ngay dưới giá sản phẩm. Nhưng nhớ rằng bạn cần sử dụng Edit ItemTemplate tính trong cửa sổ Design. Trong bất kỳ trường hợp nào, hãy chắc chắn rằng bạn có nút này trong ItemTemplate.

```
...
```

```
<span class="ProductDescription">  
    <%# Eval("Description") %>  
    <br/><br/>  
    Price:  
    </span>  
    <span class="ProductPrice">  
        <%# Eval("Price", "{0:c}") %>  
    </span>  
    <br />  
    <asp:Button ID="addToCartButton" runat="server" Text="Add to  
    Cart" CommandArgument='<%# Eval("ProductID") %>'  
    CssClass="SmallButtonText"/>  
...
```

- Chuyển sang cửa sổ Design của ProductList.ascx, chọn điều khiển DataList sử dụng cửa sổ Properties để thêm xử lý sự kiện ItemCommand, mã nguồn hoàn thiện được tạo ra sau đây:

```
// khi thêm vào giỏ hàng nút được nhấp  
protected void list_ItemCommand(object source,  
DataListCommandEvent e)  
{  
    // Đổi số của nút nhấn chứa productId  
    string productId = e.CommandArgument.ToString();  
    // Thêm một sản phẩm vào giỏ hàng  
    ShoppingCartAccess.AddItem(productId);  
}
```

-

- Thêm các chứng năng tương tự đến Product.aspx, mở Product.aspx và thêm vào nút ở dưới cùng của trang (lưu ý nút này không cần có đối số vì nút này không phải là một phần của DataList, vì vậy bạn không cần phải có chức năng này).

```
<br />
<asp:Button ID="addToCartButton" runat="server" Text="Add to
Cart" CssClass="SmallButtonText" />
```

- Trong cửa sổ Design kích đúp chuột vào nút nhấn, sẽ tự động tạo ra sự kiện trong Visual Studio. Hoàn thành chức năng này bằng cách thêm vào đoạn mã sau:

```
// thêm sản phẩm vào giỏ hàng
protected void addToCartButton_Click(object sender, EventArgs e)
{
    // lấy ProductID từ query string
    string productId = Request.QueryString["ProductID"];
    // thêm sản phẩm vào giỏ hàng
    ShoppingCartAccess.AddItem(productId);
}
```

❖ **Hiển thị thông tin tóm tắt giỏ hàng.**

Bản tóm tắt của giỏ hàng được thực hiện như là một Web User Control tên là CartSummary.ascx. Bạn sẽ sử dụng điều khiển này trong trang GiftShop.master, do đó nó xuất hiện ở tất cả các trang, tuy nhiên bạn cần viết một đoạn mã kiểm soát để chắc chắn rằng nó không còn xuất hiện trong các trang giỏ hàng. Bởi vì bạn không muốn hiển thị cả chi tiết giỏ hàng với phần tóm tắt đó trên cùng một trang.

- Thêm đoạn mã styles sau vào trong GiftShop.css

```
.CartSummary
{
    border-right: #0468a4 2px solid;
    border-top: #0468a4 2px solid;
    border-left: #0468a4 2px solid;
```

```

border-bottom: #0468a4 2px solid;
background-color: snow;
font-family: Arial;
font-size: 9pt;
color: Navy;
padding-top: 3px;
padding-left: 2px;
padding-bottom: 5px;
}
a.CartLink
{
color: Black;
font-family: Arial;
text-decoration: none;
font-size: 12px;
}
a.CartLink:hover
{
color: Red;
}

```

- Thêm vào một Web User Control trong thư mục UserControls với tên CartSummary.ascx. Chọn Place code in separate file trong hộp checkbox bên cạnh.

- Thêm vào CartSummary.ascx đoạn mã sau:

```

<table border="0" cellpadding="0" cellspacing="1" width="200">
<tr>
<td class="CartSummary">
<b><asp:Label ID="cartSummaryLabel" runat="server" /></b>
<asp:HyperLink ID="viewCartLink" runat="server"
NavigateUrl=".~/ShoppingCart.aspx"
CssClass="CartLink" Text="(view details)" />
<asp:DataList ID="list" runat="server">
<ItemTemplate>
<%# Eval("Quantity") %> x <%# Eval("Name") %>

```

```

    </ItemTemplate>
</asp:DataList>

Total:
<span class="ProductPrice">
    <asp:Label ID="totalAmountLabel" runat="server" />
</span>
</td>
</tr>
</table>

```

- Mở tập tin code-behind (CartSummary.ascx.cs) của điều khiển, và thêm hàm Page_PreRender cùng với hàm PopulateControls để trợ giúp cho nó.

```

// Điện nội dung tóm tắt giỏ hàng vào sự kiện PreRender
protected void Page_PreRender(object sender, EventArgs e)
{
    PopulateControls();
}

// Điện dữ liệu vào control
private void PopulateControls()
{
    // đưa sản phẩm vào giỏ hàng
    DataTable dt = ShoppingCartAccess.GetItems();
    // Nếu giỏ hàng rỗng...
    if (dt.Rows.Count == 0)
    {
        cartSummaryLabel.Text = "Your shopping cart is empty.";
        totalAmountLabel.Text = String.Format("{0:c}", 0);
        viewCartLink.Visible = false;
        list.Visible = false;
    }
    else
        // nếu giỏ hàng không rỗng...
}

```

```

{
    // danh sách nội dung giỏ hàng
    list.Visible = true;
    list.DataSource = dt;
    list.DataBind();
    // cài đặt điều khiển
    cartSummaryLabel.Text = "Cart summary ";
    viewCartLink.Visible = true;
    // hiển thị tổng giá giỏ hàng
    decimal amount = ShoppingCartAccess.GetTotalAmount();
    totalAmountLabel.Text = String.Format("{0:c}", amount);
}
}

```

- Bởi vì bạn se gộp giỏ hàng vào trang Master page, thông thường nó sẽ hiển thị trong mỗi trang của website của bạn. Nếu bạn không muốn phần tóm tắt giỏ hàng của bạn khi khách hàng xem giỏ hàng của mình, thêm đoạn mã sau và lớp CartSummary trong CartSummary.ascx.cs

```

// Không hiển thị phần nội dung tóm tắt trong trang Shopping cart
protected void Page_Init(object sender, EventArgs e)
{
    // lấy trang hiện thời
    string page = Request.AppRelativeCurrentExecutionFilePath;
    // nếu đang trong giỏ hàng, không hiển thị nội dung tóm tắt
    if (String.Compare(page, "~/ShoppingCart.aspx", true) == 0)
        this.Visible = false;
    else
        this.Visible = true;
}

```

- Chuyển sang GiftShop.master cửa sổ Design , kéo thả CartSummary.ascx từ Solution Explorer vào GiftShop.master, hiển thị như hình dưới.



Hình 3.54: Thêm điều khiển CartSummary.ascx vào Master page

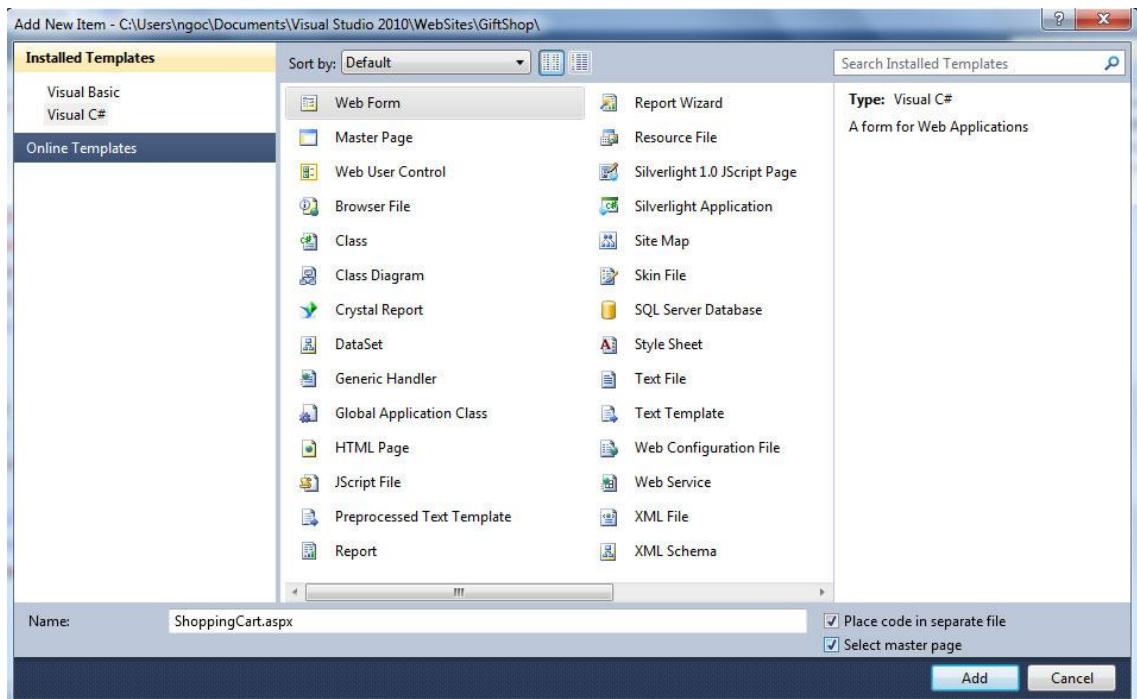
❖ Hiển thị giỏ hàng.

- Trước khi làm việc với giỏ hàng ta thêm đoạn mã styles sau vào

GiftShop.css

```
.ShoppingCartTitle
{
    color: Red;
    font-family: Verdana, Helvetica, sans-serif;
    font-size: 16px;
}
```

- Chuột phải vào dự án trong Solution Explorer và kích Add New Item.
 - Chọn Web Form Template, đặt tên ShoppingCart.aspx. Kích vào hộp chọn Place code in separate file and Select master page như hình dưới và sau đó kích Add.



Hình 3.55: Thêm ShoppingCart.aspx

- Chọn GiftShop.master trong hộp thoại hiện ra và kích OK.
- Bạn hãy thiết kế trang ShoppingCart giống hình dưới đây, và đặt thuộc các thuộc tính có trong trang như bảng dưới.

Tên sản phẩm	Giá	Số lượng	Thành tiền	Xóa
Databound	Databound	Datas	Databound	Xóa
Databound	Databound	Datas	Databound	Xóa
Databound	Databound	Datas	Databound	Xóa
Databound	Databound	Datas	Databound	Xóa
Databound	Databound	Datas	Databound	Xóa

Hình 3.56: ShoppingCart.aspx trong cửa sổ Design

Thuộc tính các điều khiển trong ShoppingCart.aspx

Loại điều khiển	Thuộc tính ID	Thuộc tính Text	Thuộc tính CssClass
Label	titleLabel	Giò hàng của bạn	ShoppingCartTitle
Label	statusLabel	(đèn trống)	AdminPageText
GridView	grid		
Label	totalAmountLabel		ProductPrice
Button	updateButton	Cập nhật số lượng	ButtonText
Button	ContinueShoppingButton	Tiếp tục mua hàng	ButtonText

- Bây giờ bạn định dạng điều khiển GridView. Thiết lập thuộc tính AutoGenerateColumns là false, DataKeyNames là ProductID, độ rộng 100%, và BorderWidth là 0px.
- Trong cửa sổ Design , chọn GridView, kích vào Smart Link, và chọn Add New Column và thêm vào các cột theo danh sách dưới đây:

Column Type	Header Text	Data field	Thuộc tính khác
BoundField	Tên sản phẩm	Name	Read Only
BoundField	Giá	Price	Read Only
TemplateField	Số lượng		
BoundField	Thành tiền	Subtotal	Read Only
			Command Name: Delete Text: Xóa Button Type: Button

- Kích GridView Smart Link và chọn Edit Columns, trong bảng Selected fields, chọn Price và đặt thuộc tính DataFormatString giá trị {0:c}.
- Tương tự như vậy đặt thuộc tính DataFormatString giá trị {0:c} cho trường Subtotal.
- Trường Quantity là một Template Field vì vậy bạn phải điền nó bằng tay, chuyển sang cửa sổ Source thêm Textbox sau đây tới Item Template:

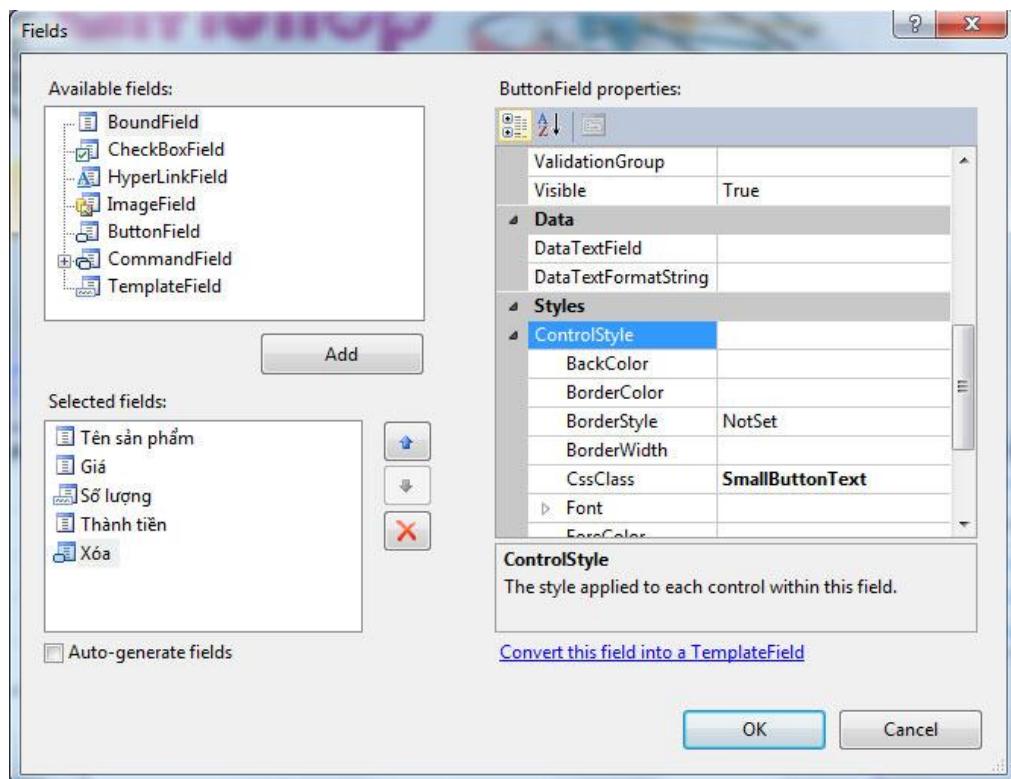
```
<asp:TemplateField HeaderText="Số lượng">
    <ItemTemplate>
        <asp:TextBox ID="editQuantity" runat="server"
        CssClass="GridEditingRow" Width="24px" MaxLength="2"
        Text='<%#Eval("Quantity")%>' />
```

```

    </ItemTemplate>
</asp:TemplateField>

```

- Sử dụng style SmallButtonText cho nút nhấn Delete, chuyển sang cửa sổ Design kích vào Grid Smart Link, chọn Edit Columns. Trong hộp thoại mở ra, chọn Delete field kích chuột thuộc tính ControlStyle, và đặt CssClass là SmallButtonText, như hình dưới đây:



Hình 3.57: Chọn style cho nút nhấn Xóa

- Kích chuột OK để đóng hộp thoại. Xác minh rằng trong phần nội dung của nó phải giống hình 3.58.

Tên sản phẩm	Giá	Số lượng	Thành tiền	Xóa
Databound	Databound	Databound	Databound	Xóa
Databound	Databound	Databound	Databound	Xóa
Databound	Databound	Databound	Databound	Xóa
Databound	Databound	Databound	Databound	Xóa
Databound	Databound	Databound	Databound	Xóa

Hình 3.58. Phần nội dung của ShoppingCart.aspx trong cửa sổ Design

- Mở tập tin code-behind (ShoppingCart.aspx.cs), viết những đoạn mã giống dưới đây:

```
public partial class ShoppingCart : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        // Tải trang ban đầu
        if (!IsPostBack)
            PopulateControls();
    }
    // đỗ dữ liệu vào điều khiển giỏ hàng
    private void PopulateControls()
    {
        // đặt tiêu đề cho trang
        this.Title = BalloonShopConfiguration.SiteName + " : Shopping
Cart";
        // lấy những sản phẩm có trong giỏ hàng
        DataTable dt = ShoppingCartAccess.GetItems();
        // nếu giỏ hàng là rỗng...
        if (dt.Rows.Count == 0)
        {
            titleLabel.Text = "Giỏ hàng trống!";
            grid.Visible = false;
            updateButton.Enabled = false;
            totalAmountLabel.Text = String.Format("{0:c}", 0);
        }
        else
        // nếu giỏ hàng không rỗng...
        {
            // liệt kê danh sách vào nội dung giỏ hàng
            grid.DataSource = dt;
            grid.DataBind();
            // cài đặt điều khiển
            titleLabel.Text = "Những sản phẩm trong giỏ hàng của bạn:";
            grid.Visible = true;
            updateButton.Enabled = true;
        }
    }
}
```

```

    // hiển thị tổng giá
    decimal amount = ShoppingCartAccess.GetTotalAmount();
    totalAmountLabel.Text = String.Format("{0:c}", amount);
}
}
}

```

❖ **Chỉnh sửa số lượng sản phẩm**

- Mở ShoppingCart.aspx trong cửa sổ Design, chọn GridView, trong cửa Properties trong phần Action kích đúp vào RowDeleting cho việc tạo sự kiện tự động:

- Viết vào đoạn mã giống dưới đây:

```

// xoá sản phẩm khỏi giỏ hàng
protected void grid_RowDeleting(object sender,
GridViewDeleteEventArgs e)
{
    // Chỉ số hàng bị xóa
    int rowIndex = e.RowIndex;
    // ID của sản phẩm bị xóa
    string productId = grid.DataKeys[rowIndex].Value.ToString();
    // Xoá sản phẩm khỏi giỏ hàng
    bool success = ShoppingCartAccess.RemoveItem(productId);
    // Hiển thị trạng thái
    statusLabel.Text = success ? "<br />Xóa sản phẩm thành công!<br />" :
        "<br />Có lỗi trong việc xoá sản phẩm khỏi giỏ hàng!<br />";
    PopulateControls();
}

```

- Trong ShoppingCart.aspx kích đúp vào nút Update Quantities, và viết vào đoạn mã như dưới đây:

```

// Cập nhật số lượng sản phẩm vào giỏ hàng
protected void updateButton_Click(object sender, EventArgs e)
{

```

```

// Số hàng trong GridView
int rowsCount = grid.Rows.Count;
// lưu trữ một hàng của GridView
GridViewRow gridViewRow;
// Đưa vào một textbox nhập số lượng trong GridView
TextBox quantityTextBox;
// Khai báo biến lưu trữ ID sản phẩm và số lượng
string productId;
int quantity;
// Khi cập nhật thành công?
bool success = true;
// duyệt qua các hàng của GridView
for (int i = 0; i < rowsCount; i++)
{
    // Lấy một hàng
    gridViewRow = grid.Rows[i];
    // ID của sản phẩm bị xoá
    productId = grid.DataKeys[i].Value.ToString();
    // Lấy Textbox số lượng trong hàng
    quantityTextBox = (TextBox)gridViewRow.FindControl("editQuantity");
    // Lấy số lượng và bảo vệ các giá trị không có thật
    if (Int32.TryParse(quantityTextBox.Text, out quantity))
    {
        // Cập nhật số lượng sản phẩm
        success = success && ShoppingCartAccess.UpdateItem(productId,
quantity);
    }
    else
    {
        // Không thành công
        success = false;
    }
    // thông báo trạng thái
    statusLabel.Text = success ? "Thành công" : "Không thành công";
}

```

```

    "<br />Cập nhật giỏ hàng thành công!<br />" ;
    "<br />Lỗi cập nhật! Làm ơn kiểm tra lại!<br />";
}
PopulateControls();
}

```

❖ Chức năng “Continue Shopping” (Tiếp tục mua hàng)

- Chính sửa ShoppingCart.aspx trong cửa sổ Design kích đúp vào nút Continue Shopping tự động tạo ra phương thức continueShoppingButton_Click, sửa nó giống dưới đây:

```

// Chuyển hướng đến các trang danh mục trước đó
protected void continueShoppingButton_Click(object sender,
EventArgs e)
{
    // Chuyển hướng đến trang danh mục cuối cùng truy cập
    // hoặc đến trang chính của danh mục
    object page;
    if ((page = Session["LastVisitedCatalogPage"]) != null)
        Response.Redirect(page.ToString());
    else
        Response.Redirect(Request.ApplicationPath);
}

```

- Mở GiftShop.master.cs và sửa đổi nó để có thể lưu vị trí trang hiện tại và phiên làm việc (Session) của khách truy cập.

```

public partial class GiftShop : System.Web.UI.MasterPage
{
    // Các trang web được coi là “danh mục trang” mà người truy cập
    // có thể tiếp tục mua hàng
    private static string[] catalogPages = { "~/Default.aspx",
    "~/Catalog.aspx", "~/Search.aspx" };
    // Thực thi bất kỳ trang nào khi trang được tải
    protected void Page_Load(object sender, EventArgs e)
    {

```

```

// Khi trang tải lần đầu
if (!IsPostBack)
{
    /* Lưu truy cập vào trang danh mục mới nhất vào session
       Để hỗ trợ chức năng tiếp tục mua hàng “Continue Shopping” */
    // lấy những trang hiện được nạp
    string currentLocation =
Request.AppRelativeCurrentExecutionFilePath;
    // Nếu trang là một nơi mà khác muốn tiếp tục mua hàng
    // ghi nó lại vào session của người truy cập
    for (int i = 0; i < catalogPages.GetLength(0); i++)
        if (String.Compare(catalogPages[i], currentLocation, true) == 0)
    {
        // ghi lại vị trí hiện thời
        Session["LastVisitedCatalogPage"] = Request.Url.ToString();
        // ngừng vòng lặp
        break;
    }
}
}
}

```

- Mở trang Product.aspx và thêm một nút nhấn Continue Shopping cạnh nút Add to Cart, với những thuộc tính sau:

Tên thuộc tính	Giá trị thuộc tính
ID	ContinueShoppingButton
CssClass	SmallButtonText
Text	Tiếp tục mua hàng

- Trong cửa sổ Design, kích đúp vào để có sự kiện Click của nó được tạo ra và hoàn thành đoạn mã cho nút “Tiếp tục mua hàng”:

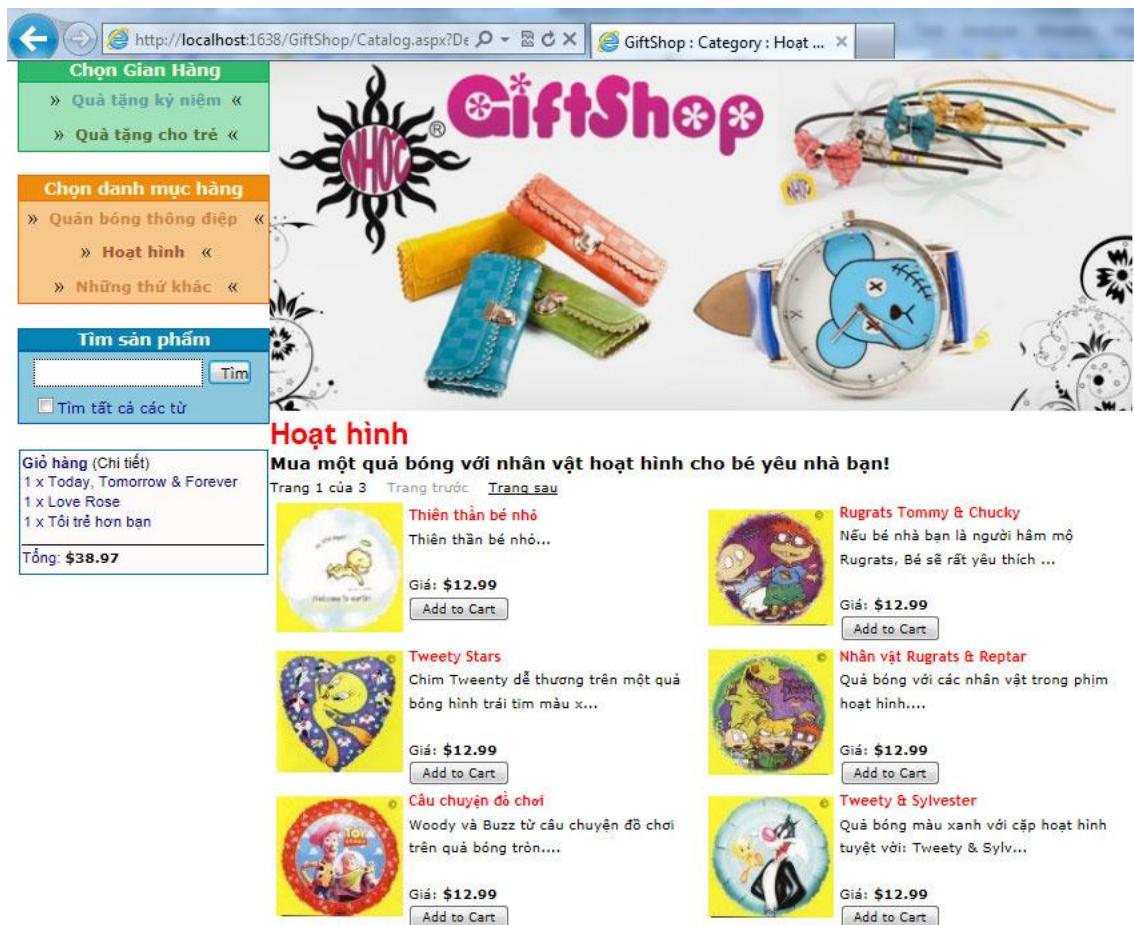
```

// Chuyển hướng đến các trang danh mục đã truy cập trước đó
protected void continueShoppingButton_Click(object sender,
EventArgs e)

```

```
{
    // chuyển hướng đến trang danh mục cuối cùng được truy cập
    object page;
    if ((page = Session["LastVisitedCatalogPage"]) != null)
        Response.Redirect(page.ToString());
    else
        Response.Redirect(Request.ApplicationPath);
}
```

- Thực thi dự án và đảm bảo rằng không xảy ra lỗi nào.



Hình 3.59. Hiển thị nội dung tóm tắt của giỏ hàng



Hình 3.60: Trang ShoppingCart.aspx khi có sản phẩm

B. THỰC HÀNH

1. Nội dung thực hành

Thực hành lại các ví dụ và ứng dụng đã làm ở bài học

2. Bảng vật tư thiết bị cần thiết

TT	Thiết bị - Vật tư	Thông số kỹ thuật	Số lượng
1	Một bộ máy tính	<ul style="list-style-type: none"> - CPU: duo core 2.0 Ghz trở lên - Ram: 1 Gb trở lên - HDD: 40 GB 	1 bộ / học sinh
2	Phần mềm	<ul style="list-style-type: none"> - Hệ điều hành Windows 7 - Microsoft SQL Server 2008 - Microsoft Visual Studio 2010 	

3. Quy trình thực hiện công việc

TT	Công việc	Thao tác	Yêu cầu
1	Chuẩn bị	<ul style="list-style-type: none"> - Kiểm tra các hoạt động của máy tính. - Phần mềm Visual Studio 2010 - Phần mềm MS SQL Server 2008 	MT Hoạt động bình thường. Đã cài đặt đầy đủ.

2	Xác định yêu cầu.	<ul style="list-style-type: none"> - Xác định nội dung, mục đích của website. - Xác định đối tượng chủ yếu sử dụng website 	<ul style="list-style-type: none"> - Đầy đủ và chính xác.
3	Tạo cơ sở dữ liệu.	<ul style="list-style-type: none"> - Thiết kế các bảng. - Ràng buộc giữa các bảng với Nhau. - Thiết kế các thủ tục cần thiết xử lý dữ liệu. - Thiết kế các hàm cần dùng nếu có. 	Đầy đủ, chính xác.
4	Xây dựng các lớp xử lý dữ liệu.	<ul style="list-style-type: none"> - Lớp cấu hình. - Lớp xử lý dữ liệu. 	Đúng
5	Xây dựng lớp hiển thị.	<ul style="list-style-type: none"> - Thiết kế khung trang trong tệp tin Master page. - Xây dựng các Web User Control. - Xây dựng các trang theo yêu cầu của người sử dụng. 	Đúng , trang web phải hoạt động tốt.
6	Kiểm tra.	<ul style="list-style-type: none"> - Chạy Website. - Chính sửa lại các trang theo yêu cầu. 	<ul style="list-style-type: none"> - Hoạt động tốt. - Không lỗi.

4. Tổ chức thực hiện.

Mỗi học sinh một máy, thực hành dưới sự hướng dẫn của giáo viên.

5. Kiểm tra đánh giá.

- Tạo và liên kết được với cơ sở dữ liệu: 2 điểm
- Các chức năng trên các form giao diện hoạt động tốt : 6 điểm.
- Các trang web đẹp, bố cục hợp lý, màu sắc hài hoà : 2 điểm.

BÀI TẬP BÀI 3.

Bài 1: Thực hành lại tất cả các thí dụ về HTML Server Controls và Web Server Controls đã được học. Có thể viết các bài thực hành có nội dung tương đương các thí dụ để nắm vững bài học.

Bài 2: Sử dụng các điều khiển để thực hiện các bài tập sau:

Tạo trang Default.aspx để thực hiện các nội dung sau:

- 1) Nhập x, in ra trị tuyệt đối của x.
- 2) Tính $T = 1+2+3+\dots+n$ với n nguyên và $n > 0$.
- 3) In ra các số có 3 chữ số sao cho $abc = a^3 + b^3 + c^3$
- 4) Tính $S = 1+2/2^2+3/3^2+\dots+n/n^n$, với n nguyên > 0 .

Hãy tạo các đường link đến từng trang cho từng bài tập ở trên.

Bài 3: Thực hành lại các ví dụ trong bài học, có thể tùy chỉnh theo ý của người làm.

Bài 4: Đọc và giải thích nội dung của phần mã lệnh trang bai4.aspx.

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="bai4.aspx.cs" Inherits="bai4" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Session</title>
</head>
<body runat="server" id="BodyTag">
    <form id="form1" runat="server">
        <div>
            <asp:DropDownList runat="server" id="Chonmau"
                autopostback="true"
                onselectedindexchanged="Chonmau_IndexChanged">
                <asp:ListItem value="White" selected="True">Chọn màu
                ...</asp:ListItem>
                <asp:ListItem value="Red">Red</asp:ListItem>
                <asp:ListItem value="Green">Green</asp:ListItem>
                <asp:ListItem value="Blue">Blue</asp:ListItem>
            </asp:DropDownList>
        </div>
    </form>
</body>
</html>
```

```

</asp:DropDownList>
</div>
</form>
</body>
</html>

```

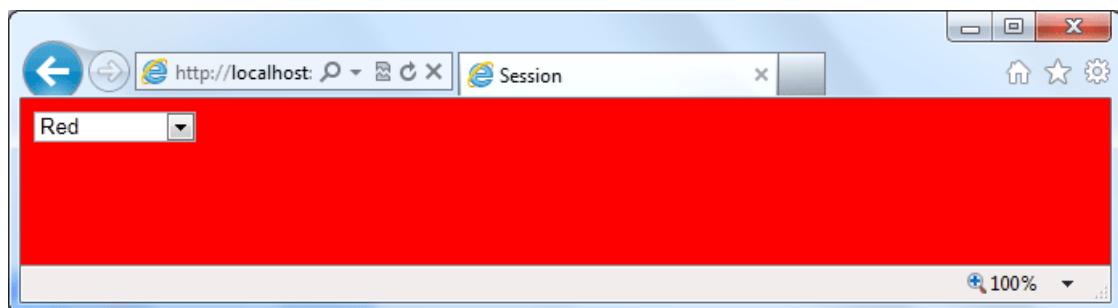
Phần code behind

```

public partial class bai4 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["BackgroundColor"] != null)
        {
            Chonmau.SelectedValue =
                Session["BackgroundColor"].ToString();
            BodyTag.Style["background-color"] = Chonmau.SelectedValue;
        }
    }
    protected void Chonmau_IndexChanged(object sender, EventArgs e)
    {
        BodyTag.Style["background-color"] = Chonmau.SelectedValue;
        Session["BackgroundColor"] = Chonmau.SelectedValue;
    }
}

```

Thực thi trang, khi chọn màu Red từ DropDownList, kết quả thực thi trang:



Hình 3.61: Kết quả thực thi trang

- Kết thúc ứng dụng và chạy lại ứng dụng nà trong cùng một cửa sổ trình duyệt đang mở, trang hiển thị ra sao? Giải thích.
- Nếu đóng cửa sổ trình duyệt và thực thi lại ứng dụng , trang hiển thị ra sao? Giải thích.

Bài 5: Tạo một ứng dụng ASP.NET về đối tượng Application và tập tin Global.asax .

-Tạo trang Global.asax.

```
<%@ Application Language="C#" %>
<script runat="server">
void Application_Start(object sender, EventArgs e)
{
    int SiteHitCounter=500;
    int CurrentUsers=100;
    Application["SiteHitCounter"] = SiteHitCounter;
    Application["CurrentUsers"] = CurrentUsers;
}
void Application_End(object sender, EventArgs e)
{
    HitCountClass cls = new HitCountClass();
}
void Application_Error(object sender, EventArgs e)
{
    // Code that runs when an unhandled error occurs
}
void Session_Start(object sender, EventArgs e)
{
    Application["SiteHitCounter"] =
        (int)Application["SiteHitCounter"] + 1;
    Application["CurrentUsers"] =
        (int)Application["CurrentUsers"] + 1;
}
void Session_End(object sender, EventArgs e)
{
    Application["CurrentUsers"] =
        (int)Application["CurrentUsers"] - 1;
    // Code that runs when a session ends.
    // Note: The Session_End event is raised only when the sessionstate
    mode
```

```

// is set to InProc in the Web.config file. If session mode is set to
StateServer
// or SQLServer, the event is not raised.
}
</script>

```

Tạo trang Counter.aspx

```

<%@      Page      Language="C#"      AutoEventWireup="true"
CodeFile="Counter.aspx.cs"
Inherits="ApplicationPage" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>Counter</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:Label ID="Label1" runat="server"></asp: Label><br />
<br />
<asp:Label ID="Label2" runat="server"></asp: Label>&nbsp;</div>
</form>
</body>
</html>

```

Viết đoạn mã sau cho trang Counter.aspx.cs

```

public partial class Counter : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
    Label.Text = "Hit Counter: " +
    Application["SiteHitCounter"].ToString();
    Label.Text = "Current users: " +

```

```
Application["CurrentUsers"].ToString();  
}  
}
```

Khi thực thi trang Counter ta được kết quả sau:

Hit Counter: 501

Current users: 101

Hãy giải thích tại sao lại có kết quả trên?

Bài 6: Hãy viết chương trình thêm sửa xóa cho các bảng (Department, Category, Product) và quản lý hóa đơn để hoàn thành ứng dụng mẫu đã thiết kế ở bài 3, phần 5.

TÀI LIỆU THAM KHẢO

Tiếng Việt.

- [1] Macromedia Dreamweaver 8 - Nguyễn Tường Sinh và Lê Minh Hoàng - NXB Lao động xã hội, năm 2008.
- [2] Kỹ thuật xây dựng ASP.NET – Nguyễn Văn Lân – NXB Lao động xã hội, 2008.
- [3] Học và thực hành thiết kế web chuyên nghiệp với Macromedia Dreamweaver – Nguyễn Mạnh Hải – NXB Văn hóa thông tin, 2007.
- [4] Giáo trình HTML và thiết kế Web - Khoa Công nghệ thông tin, Trường ĐH Công Nghệ, ĐHQGHN.
- [5] Bài giảng thiết kế web - Dương Thành Phát trường Cao đẳng Công nghệ thông tin Hồ Chí Minh.
- [6] Giáo trình thiết kế Web và làm hoạt hình - Phạm Quang Huy - NXB Giao thông vận tải, năm 2003.
- [7] Giáo trình tự học thiết kế Web động -Hoàng Hải - NXB Lao động Xã Hội , năm 2007.

Tiếng Anh

- [8] HTML, DHTML and JavaScript - Aptech WorldWide's book..
- [9] Beginning ASP.NET 2.0 in C# 2005-Matthew MacDonald - 2006
- [10] Pro ASP.NET 2.0 in C# 2005 - Matthew MacDonald và Mario Szpuszta -2005
- [11] Beginning SQL Server 2005 for Developers - Robin Dewson -2006
- [12] Professional ASP.NET 2.0 - Wiley Publishing xuất bản năm 2006
- [13] Web Designer For Teens - Thomson Course Technology PTR - 2005
- [14] Microsoft® ASP.NET 4Step by Step tác giả George Shepherd - 2010