

CHƯƠNG 3: MASTERPAGE VÀ USER CUSTOM CONTROL

3.1. MasterPage

Trong một hệ thống website thường sẽ có nhiều trang con khác nhau, mỗi trang sẽ phụ trách một mảng nội dung nào đó, có thể kết cấu trang cũng có những điểm khác biệt nhất định nhưng tất cả sẽ được thiết kế theo một khuôn mẫu nào đó, và áp dụng chung cho tất cả các trang. Khuôn mẫu này thường chúng ta sẽ gọi là Theme hay Template, còn trong ASP.Net có khái niệm **MasterPage**. MasterPage là một trang dùng để tạo ra cấu trúc giao diện chung cho các trang web trong hệ thống, những trang thừa kế từ MasterPage sẽ áp dụng lại giao diện mẫu của nó. Trong một website có thể có 1 hoặc nhiều MasterPage.

Trong MasterPage được chia thành 2 phần:

- Phần chung: Đây là giao diện chung áp dụng cho những trang con kế thừa từ nó
- Phần riêng: Đây là nơi chứa nội dung riêng của từng trang.

3.2 User Custom Control

Visual studio cung cấp cho chúng ta rất nhiều các điều khiển để phát triển ứng dụng gọi là điều khiển nội tại (Instrict control). Ngoài ra, nó còn cung cấp cho chúng ta khả năng tự xây dựng các điều khiển tùy biến nếu các điều khiển hiện hành không đáp ứng được yêu cầu. Ví dụ: Nếu ứng dụng của bạn cần chiếc máy tính (Calculator) ở rất nhiều trang thì giải pháp tốt nhất là tạo một điều khiển Calculator riêng thay việc kết hợp các điều khiển truyền thống, khi đó ta có thể sử dụng điều khiển này trong toàn bộ ứng dụng.

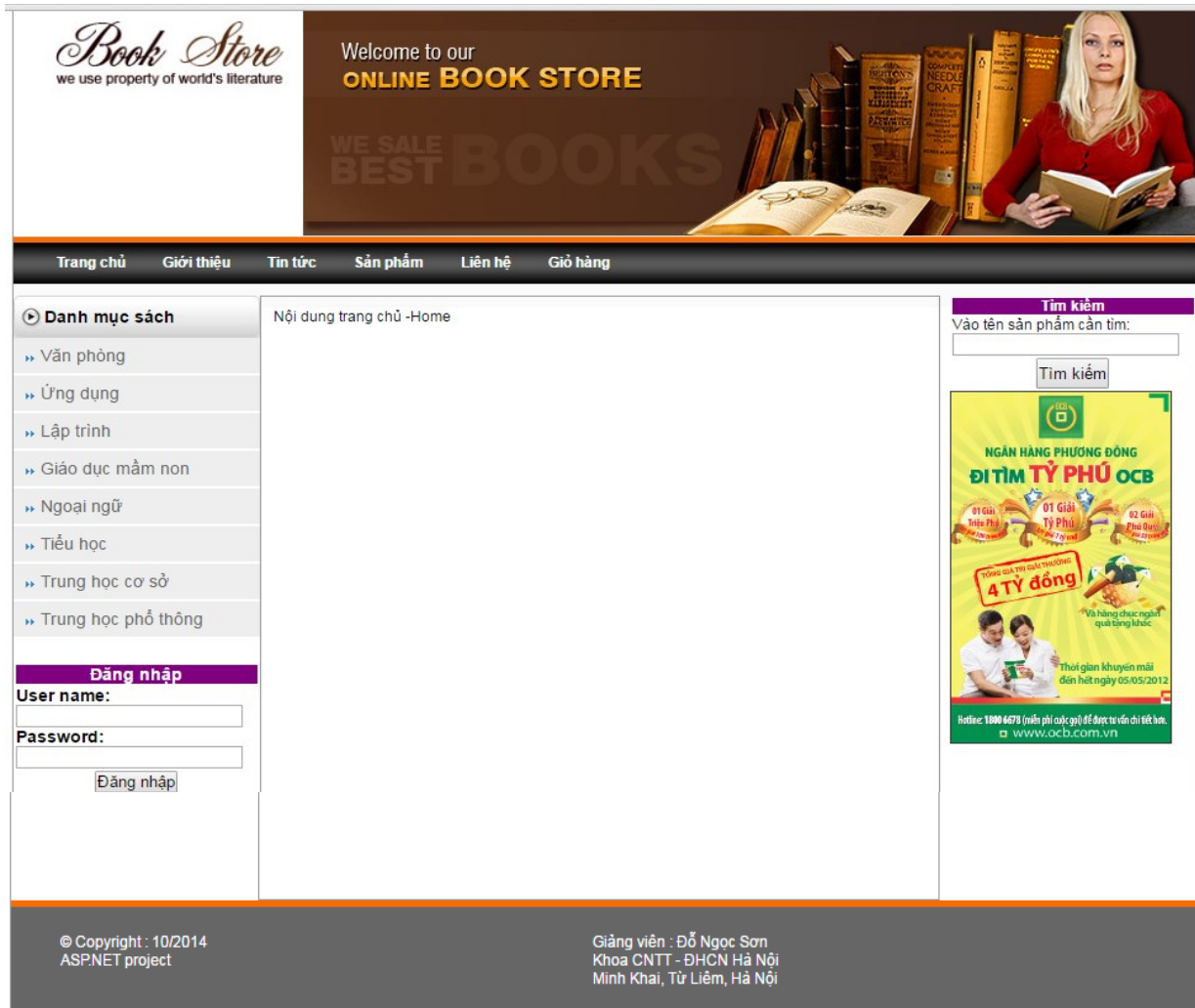
Bài học này sẽ hướng dẫn cách tạo và sử dụng điều khiển do lập trình viên tự xây dựng – hay còn gọi là điều khiển tùy biến (Custom Control). Tiếp theo sẽ minh họa thêm một số ví dụ về tạo điều khiển tùy biến để người đọc hiểu rõ hơn.

Thực chất của User Custom Control (UCC) chính là một "trang con", trong đó có thể chứa **bất kỳ nội dung nào** (trừ các thẻ <HTML> <BODY>, <FORM>, vì một trang chỉ có duy nhất một lần xuất hiện các thẻ này). "Trang con" này sau đó có thể được chèn (Include) vào các trang khác để sử dụng. Khi muốn cập nhật nội dung ở tất cả các trang, ta chỉ việc sửa đổi duy nhất UCC ban đầu. Khả năng này của ASP.NET giúp chúng ta xây dựng ứng dụng nhanh hơn, dễ bảo trì hơn.

Mỗi một UCC được đặt trong một trang có phần mở rộng là *.ascx. File này có đặc điểm là không truy cập trực tiếp từ trình duyệt mà chỉ được chèn vào các trang aspx.

BÀI TẬP

(Tạo trang Master và các trang con kế thừa từ trang Master)



1. Chuẩn bị:

- Tạo dự án WebShop
- Tạo thư mục: Page trong WebShop
- copy thư mục CSS, Images vào WebShop

2. Tạo trang **FrontEnd.Master** trong Page: Click chuột phải vào Page, chọn Add New Item, chọn Master Page, đặt Name: FrontEnd.Master

Source Code trang FrontEnd.Master như sau :

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="FrontEnd.master.cs"
Inherits="Page_FrontEnd" %>
```

```
<%@ Register src="Login.ascx" tagName="Login" tagprefix="uc1" %>
```

```
<%@ Register src="Search.ascx" tagName="Search" tagprefix="uc2" %>
```

```
<!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>Web Shop</title>
```

```
<link href=" ../CSS/Style.css" rel="stylesheet" type="text/css" />
```



```

</div> <%--End Main--%>
<div id="footer">
  <p>
    &copy; Copyright : 10/2018<br />
    ASP.NET project
  </p>
  <p>
    Giảng viên : Đỗ Ngọc Sơn<br />
    Khoa CNTT - ĐHCN Hà Nội<br />
    Minh Khai, Từ Liêm, Hà Nội
  </p>
</div> <%--End Footer--%>
</div> <%-- End page--%>
</form>
</body>
</html>

```

3. Tạo trang Home.aspx (kế thừa từ FrontEnd.Master)

Click chuột phải vào Page/Add New Item/ Web Form

+Name: **Home.aspx**, chọn **Select Master Page** nhấn **Add**; Chọn **FrontEnd.Master**, nhấn **OK**

<asp:Content ID="Content1" ContentPlaceHolderID="main_body" runat="server">

<p> **Nội dung trang chủ - Home** </p>

</asp:Content>

Chạy thử trang **Home.aspx**

4. Tương tự tạo các trang: Instructions.aspx; News.aspx; Product.aspx; Contact.aspx; ShowCart.aspx kế thừa từ **FrontEnd.Master**

Login.ascx

```

<%@ Control Language="C#" AutoEventWireup="true" CodeFile="Login.ascx.cs"
Inherits="Page_Login" %>
<table width="200px">
  <tr>
    <td align="center" style="background-color:Purple; color:White;font-weight:bold">
      Đăng nhập
    </td>
  </tr>
  <tr>
    <td>
      User name: <br />
      <asp:TextBox runat="server" ID="txtUserName" Width="92%"> </asp:TextBox> <br />
      Password: <br />
      <asp:TextBox runat="server" ID="txtPassword" Width="92%"> </asp:TextBox> <br />
    </td>
  </tr>
  <tr>
    <td align="center">
      <asp:Button runat="server" ID="cmdLogin" Text="Đăng nhập" />
    </td>
  </tr>
</table>

```

Search.ascx

```
<%@ Control Language="C#" AutoEventWireup="true" CodeFile="Search.ascx.cs"
Inherits="Page_Search" %>
<table width="200px">
<tr>
<td align="center" style="background-color:Purple; color:White;font-weight:bold">
Tìm kiếm
</td>
</tr>
<tr>
<td>
Vào tên sản phẩm cần tìm: <br />
<asp:TextBox runat="server" ID="txtten" Width="92%"> </asp:TextBox> <br />
</td>
</tr>
<tr>
<td align="center">
<asp:Button runat="server" ID="cmdTim" Text="Tìm kiếm" />
</td>
</tr>
</table>
```

CHƯƠNG 4: CÔNG NGHỆ ADO.NET

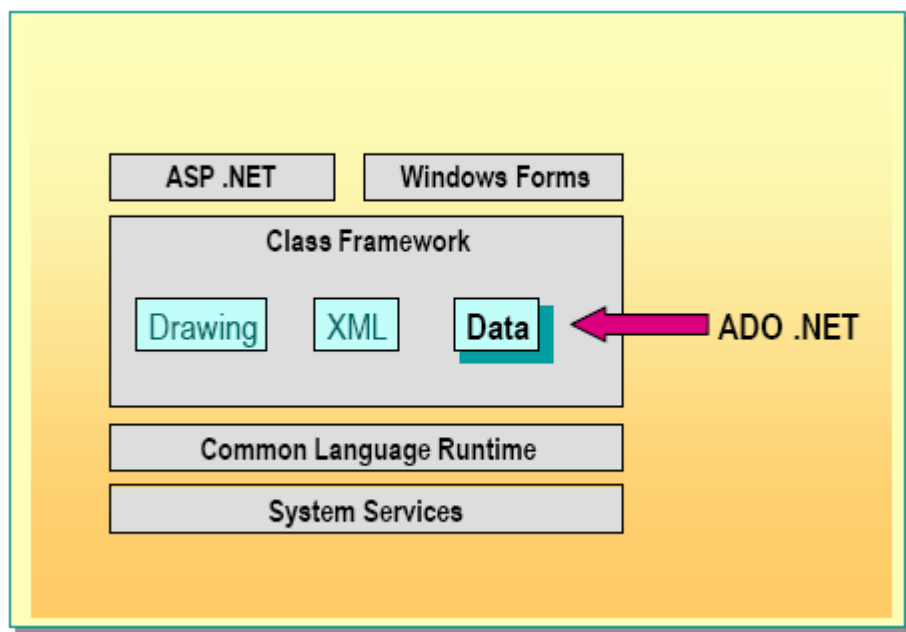
4.1 Giới thiệu chung

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

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

- **Mục tiêu chính của ADO.NET là:**

- Cung cấp các lớp để thao tác CSDL trong cả hai môi trường là phi kết nối (Disconnected data) và kết nối (Connected data).
 - Tích hợp chặt chẽ với XML (Extensible Markup Language)
 - Tương tác với nhiều nguồn dữ liệu thông qua mô tả dữ liệu chung.
 - Tối ưu truy cập nguồn dữ liệu (OLE DB & SQL server).
- Làm việc trên môi trường Internet. Các lớp của ADO.NET được đặt trong Namespace là System.Data/ System.Data.oledb
- ADO.NET bao gồm 2 Provider (2 bộ thư viện thường dùng) để thao tác với các CSDL là: OLE DB Provider (nằm trong System.Data.OLEDB) dùng để truy xuất đến **bất kỳ CSDL nào có hỗ trợ OLEDB**; SQL Provider (nằm trong System.Data.SqlClient) chuyên dùng để truy xuất đến CSDL SQL Server (Không qua OLE DB nên nhanh hơn). Hiện nay, các hãng thứ ba còn cung cấp các Provider khác như : MySQL, Oracle... provider để cho phép ứng dụng .NET truy xuất đến các cơ sở dữ liệu không phải của Microsoft khác.
- Vị trí của ADO.NET trong kiến trúc của .NET Framework



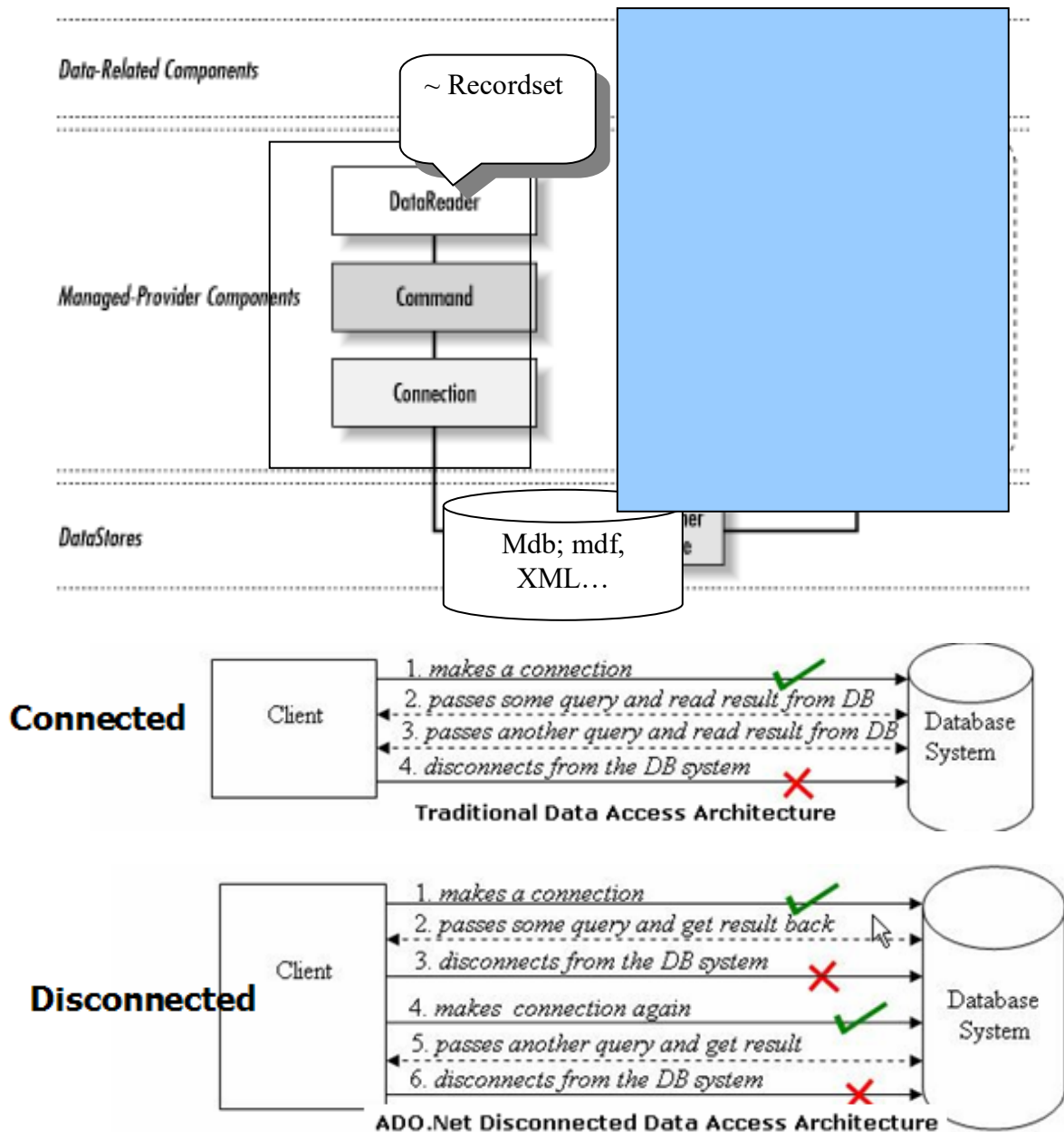
Vị trí của ADO.NET trong kiến trúc của .net Framework

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

4.2 Kiến trúc của ADO.NET

ADO.NET cho phép làm việc ở cả hai chế độ, chế độ Kết nối (Connected) và phi kết nối (Disconnected).

Bộ ba Connection, Command và DataReader: cho phép làm việc ở chế độ Connected; còn DataAdapter, Connection, Command và Dataset làm việc ở chế độ Disconnected. Trong chế độ Connected thì mỗi khi thao tác (như sửa, xóa, thêm) thì đều đòi hỏi ứng dụng phải kết nối và thao tác trực tiếp với cơ sở dữ liệu (CSDL); còn trong chế độ Disconnected thì vẫn có thể thêm, sửa, xóa dữ liệu trên đối tượng cục bộ; không nhất thiết phải kết nối ngay đến CSDL (Xem mô hình ở dưới).



4.3 Các lớp thao tác với CSDL: Connection, Command,....

4.3.1 Lớp Connection

SqlConnection là đối tượng chịu trách nhiệm quản lý kết nối tới nguồn dữ liệu (Data Source). Đối tượng SqlConnection của ADO.NET chỉ nhận một tham số đầu vào là chuỗi kết nối (connection string). Trong chuỗi kết nối, các thông số được cách nhau bằng dấu “;”, chuỗi kết nối này có các thông số sau:

- **Provider:** tên hệ quản trị cơ sở dữ liệu, đối với cơ sở dữ liệu Access cần khai báo là SQLOLEDB; đối với SQL Server thì chuỗi kết nối không cần thuộc tính này.
- **DataSource (hoặc Server):** tên / địa chỉ Database Server cần kết nối tới.

- **Initial catalog (hoặc Database):** tên của cơ sở dữ liệu cần tương tác.
- **Uid:** tài khoản để đăng nhập vào Database Server.
- **Pwd:** mật khẩu để đăng nhập vào Database Server.

Đối với việc kết nối với cơ sở dữ liệu SQL Server thì tham số Provider có thể được bỏ đi và chuỗi kết nối như sau:

Server=(local);Database=CSharpEx;uid=csharp;pwd=cpass;

a) Các thuộc tính của SqlConnection

- **ConnectionString:** thuộc tính thiết lập hoặc lấy chuỗi kết nối.
- **ConnectionTimeout:** thuộc tính thiết lập hoặc lấy thời gian chờ trong khi truy xuất vào cơ sở dữ liệu. Khi truy xuất, chương trình sẽ chờ đúng khoảng thời gian này nếu chờ qua khoảng thời gian này mà vẫn không kết nối được vào cơ sở dữ liệu thì chương trình sẽ báo lỗi.
- **Database:** thuộc tính thiết lập hoặc lấy tên của cơ sở dữ liệu của đối tượng *SqlConnection* hiện thời.
- **DataSource:** thuộc tính thiết lập hoặc lấy tên của *Database Server* của đối tượng *SqlConnection* hiện thời.
- **State:** thuộc tính này lấy trạng thái hiện thời của *SqlConnection* bao gồm: Connecting, Broken, Open, Closed, Executing, Fetching.

b) Các phương thức của SqlConnection

- **BeginTransaction:** phương thức này được sử dụng cho trường hợp xử lý giao tác của ứng dụng. Việc xử lý giao tác rất có lợi trong khi xử lý dữ liệu từ database vì có lúc việc xử lý dữ liệu gặp lỗi và lúc ấy cần thực hiện câu lệnh như Rollback để khôi phục lại trạng thái của dữ liệu trước khi xử lý.
- **Open:** phương thức này thực hiện việc mở một kết nối. Phương thức này cần được thực hiện trong khối *try/catch* để xử lý các ngoại lệ.
- **Close:** phương thức này thực hiện việc đóng kết nối và giải phóng tài nguyên. Phương thức này thường được đặt trong khối *finally* để thực hiện giải phóng tài nguyên khi kết thúc xử lý với cơ sở dữ liệu.

4.3.2 Lớp Command

SqlCommand là đối tượng thực hiện các lệnh tương tác với cơ sở dữ liệu. Đối tượng này được xây dựng để vừa có thể xử lý các stored procedure của SQL Server vừa có thể thực hiện các câu lệnh truy vấn trực tiếp đến cơ sở dữ liệu.

a) Các thuộc tính của SqlCommand

- **CommandType:** Thiết lập hoặc lấy kiểu lệnh tương tác, lệnh tương tác này có thể là stored procedure hoặc là câu lệnh truy vấn.
- **CommandText:** Thiết lập hoặc lấy lệnh thao tác với dữ liệu. Lệnh này có thể là tên của stored procedure đã có sẵn trong cơ sở dữ liệu hoặc là câu lệnh truy vấn tùy thuộc vào thuộc tính CommandType.
- **CommandTimeout:** Thiết lập hoặc lấy thời gian chờ thực hiện lệnh. Sau khoảng thời gian này nếu tương tác cơ sở dữ liệu vẫn chưa xong thì chương trình sẽ báo lỗi.

- **Parameters:** Các tham số truyền vào cho đối tượng *command*. Thuộc tính này được sử dụng hiệu quả khi *CommandType* là *stored procedure*.
- **Connection:** Thiết lập hoặc lấy kết nối đang được đối tượng *SqlCommand* sử dụng.

b) Các phương thức của *SqlCommand*

ExecuteReader: Thực thi câu lệnh *CommandText* của đối tượng *SqlCommand* và trả về kiểu *SqlDataReader*. Phương thức này thường được sử dụng khi nội dung câu lệnh tương tác cơ sở dữ liệu là lệnh *select*.

ExecuteNonQuery: Thực thi câu lệnh *CommandText* của đối tượng *SqlCommand*, đây là dạng câu lệnh cập nhật cơ sở dữ liệu (thêm hoặc xóa hoặc sửa) nên chỉ trả về số dòng bị ảnh hưởng mà không trả về dòng dữ liệu nào.

ExecuteScalar: Thực thi câu truy vấn của đối tượng *Command* và chỉ trả về cột đầu tiên của dòng đầu tiên của kết quả. Các kết quả còn lại bị bỏ qua.

4.3.3 Lớp *DataReader*

SqlDataReader là đối tượng được .Net Framework cung cấp nhằm phục vụ việc truy cập vào cơ sở dữ liệu. Dữ liệu sau khi được truy vấn từ cơ sở dữ liệu sẽ được lưu trữ trong *SqlDataReader* dưới dạng bảng gồm nhiều dòng và nhiều cột giống như trong cơ sở dữ liệu. Dữ liệu trong *SqlDataReader* được truy xuất một cách tuần tự và một chiều. Điều này nghĩa là người lập trình chỉ được đọc từ *SqlDataReader* một cách tuần tự, muốn đọc được dòng thứ *i* thì phải đọc *i-1* dòng trước đó (giống như việc đọc file trên đĩa). *SqlDataReader* không cung cấp cơ chế sắp xếp cũng như cơ chế truy xuất ngẫu nhiên, do đó người lập trình thường sử dụng vòng lặp khi thực hiện đọc dữ liệu từ *SqlDataReader* để hiển thị lên giao diện.

a) Các thuộc tính của *SqlDataReader*

FieldCount: thuộc tính này trả về số trường trong *record* hiện hành.

IsClosed: thuộc tính này trả về trạng thái của *SqlDataReader* là đóng hay mở.

HasRows: thuộc tính này chỉ định *SqlDataReader* có *record* dữ liệu nào hay không.

b) Các phương thức của *SqlDataReader*

Close: phương thức này thực hiện việc đóng *SqlDataReader* và giải phóng tài nguyên.

GetBoolean, GetByte, GetChar, GetDateTime, GetDecimal: lấy các giá trị tại cột đang xét tùy vào kiểu dữ liệu.

GetValue, GetValues: lấy về giá trị hoặc tập giá trị ở dạng “nguyên thủy” (kiểu dữ liệu gốc của Database).

Read: Đọc *record* tiếp theo của *DataReader*.

c) Phương thức *ExecuteReader*

ExecuteReader là phương thức của đối tượng *SqlCommand*, phương thức này thực hiện việc truy vấn cơ sở dữ liệu trong trường hợp nội dung câu truy vấn (*CommandText*) là câu lệnh *select*. Kết quả trả về của phương thức *ExecuteReader* là *SqlDataReader*, đối tượng này tổ chức dữ liệu thành bảng giống như kết quả trả về của câu lệnh truy vấn trong hệ quản trị cơ sở dữ liệu.

4.3.4 *SqlDataAdapter*

SqlDataAdapter là một đối tượng dùng để làm cầu nối giữa cơ sở dữ liệu và *DataSet*. *SqlDataAdapter* chứa một phần dữ liệu của cơ sở dữ liệu và hoạt động theo cơ chế “kết nối”. Với

cơ chế “kết nối”, *SqlDataAdapter* được trang bị một số phương thức để lấy dữ liệu từ cơ sở dữ liệu hoặc điền ngược dữ liệu vào cơ sở dữ liệu khi cần thiết.

a) Các thuộc tính thông dụng của *SqlDataAdapter* bao gồm:

- **SelectCommand:** Thuộc tính này quy định câu lệnh select của *SqlDataAdapter*. Câu lệnh select của thuộc tính này thường được triệu gọi khi *SqlDataAdapter* thực hiện phương thức *Fill()*.
- **InsertCommand, UpdateCommand, DeleteCommand:** Các thuộc tính này lần lượt quy định câu lệnh insert, update, delete của *SqlDataAdapter*. Các câu lệnh này được triệu gọi khi *SqlDataAdapter* thực hiện phương thức *Update()* để cập nhật dữ liệu từ *SqlDataAdapter* vào cơ sở dữ liệu.

b) Các phương thức thông dụng của *SqlDataAdapter* bao gồm:

- **Fill:** Đối số của phương thức này là *DataSet* hoặc *DataTable*. Phương thức này thực hiện việc điền dữ liệu tương ứng với câu lệnh select từ cơ sở dữ liệu vào một *DataSet* hoặc *DataTable*.
- **Update:** Thực hiện việc cập nhật dữ liệu từ *SqlDataAdapter* vào cơ sở dữ liệu.

4.3.5 DataSet

DataSet là một đối tượng ở tầng “không kết nối” trong mô hình ADO.NET. *DataSet* được thiết kế tách biệt với cơ sở dữ liệu và khi vận hành không cần biết đến việc cơ sở dữ liệu thuộc kiểu gì, kết nối ra sao. Nhiệm vụ của *DataSet* là nhận dữ liệu về từ *DataAdapter* và xử lý nó. *DataSet* có thể được xem như một cơ sở dữ liệu trong bộ nhớ gồm tất cả các bảng, dữ liệu, quan hệ và ràng buộc dữ liệu. *DataSet* có nhiều đối tượng cấp thấp hơn đi kèm với nó như : *DataTable* (tương đương với một bảng), cấp thấp hơn của *DataTable* có các đối tượng *DataRow* (tương đương với một dòng), *DataColumn* (tương đương với một cột), *DataRelation* (tương đương với các quan hệ).

DataSet nhận dữ liệu từ *DataAdapter* thông qua phương thức *Fill()* và dữ liệu này được hiển thị lên giao diện thông qua thuộc tính *DataSource* của đối tượng trình bày.

4.3.6 Lớp DataTable

Chức năng: Quản lý dữ liệu dạng bảng 2 chiều (Các hàng và các cột).

Một số phương thức:

- *DataRow NewRow()* : Trả về một đối tượng *DataRow*;
- *Clear()*: Xóa tất cả các dữ liệu trong *DataTable*

Một số thuộc tính:

- *Columns*: Là một tập hợp, quản lý toàn bộ các cột (Thêm, xóa, sửa...) của *DataTable*. *Columns* lại có các phương thức để thêm/xóa cột.
- *Rows*: Là một tập hợp, quản lý toàn bộ các hàng trong *DataTable*. *Rows* cũng có các phương thức để thêm/xóa hàng.

4.4 Các điều khiển liên kết dữ liệu

4.4.1 Điều khiển GridView

GridView là một điều khiển trình diễn dữ liệu quan trọng nhất của ASP.NET. Nó cho phép gắn và hiển thị dữ liệu ở dạng bảng, trong đó mỗi hàng là một bản ghi, mỗi cột ứng với một trường dữ liệu. Ngoài việc hiển thị, *GridView* còn có rất nhiều tính năng khác mà trước đây người ta phải viết rất nhiều dòng code mới có được, ví dụ: Định dạng, phân trang, sắp xếp, sửa đổi, xóa dữ liệu.

GridView có thể gắn kết dữ liệu với các nguồn như DataReader, SqlDataSource, ObjectDataSource hay bất kỳ nguồn nào có cài đặt System.Collections.Enumerable.

4.4.1.1 Các thuộc tính và cột thuộc tính

GridView ngoài việc hiển thị thuần túy các trường của một nguồn dữ liệu, nó còn cho phép hiển thị dưới các hình thức khác (ví dụ dưới dạng nút, dạng HyperLink, dạng checkbox...), các cột khác hỗ trợ cho việc thao tác dữ liệu như Select, Update, Delete hoàn toàn có thể tùy biến trong GridView.

Để chỉnh sửa các cột dạng này, click chọn "smart tag" của GridView và chọn "Edit Field" hoặc chọn thuộc tính Columns của GridView trong cửa sổ thuộc tính.

Loại cột	Mô tả
BoundField	Hiển thị giá trị của một trường thuộc nguồn dữ liệu.
ButtonField	Hiển thị một nút lệnh cho mỗi mục trong GridView. Nút này cho phép bạn có thể tạo ra các nút tùy biến kiểu như Add hay Remove.
CheckBoxField	Hiển thị một checkbox ứng với mỗi mục trong GridView. Cột này thường được dùng để hiển thị các trường kiểu Boolean (Yes/No).
CommandField	Hiển thị các nút lệnh đã được định nghĩa sẵn để thực hiện các thao tác select, edit, hay delete.
HyperLinkField	Hiển thị giá trị của một trường dưới dạng siêu liên kết (hyperlink). Loại cột này cho phép bạn gắn một trường thứ hai vào URL của siêu liên kết.
ImageField	Hiển thị một ảnh ứng với mỗi mục trong GridView.
TemplateField	Hiển thị nội dung tùy biến của người dùng cho mỗi mục dữ liệu trong GridView, theo như mẫu định sẵn. Loại cột này cho phép ta tạo ra các cột tùy biến.

Các thuộc tính:

Thuộc tính	Mô tả
GridLines	Ẩn, hiện các đường viền của GridView.
ShowHeader	Cho phép Hiện/ ẩn phần Header
ShowFooter	Cho phép Hiện/ ẩn phần Footer
PageSize	Get/Set cho biết mỗi trang chứa bao nhiêu dòng.
PageCount	Cho biết số trang của nguồn dữ liệu mà nó gắn kết
PageIndex	Get/Set chỉ số của trang đang được hiển thị
AllowPaging	Có cho phép phân trang không (true = có)
AllowSorting	Có cho phép sắp xếp không (true=có)

AutoGenerateColumns	Có tự động sinh ra các cột ứng với các cột trong nguồn dữ liệu hay không ? Mặc định = true (có)
AutoGenerateDeleteButton	Tự động tạo ra cột Delete (true = tự động)
AutoGenerateUpdateButton	Tự động tạo ra cột Update (true = tự động)
AutoGenerateSelectButton	Tự động tạo ra cột Select (true = tự động)
EditIndex (SelectedIndex)	Đặt hàng nào đó về chế độ edit. EditIndex = 2 → hàng thứ 3 (chỉ số 2) sẽ về chế độ edit. Nếu đặt EditIndex = -1 thì sẽ thoát khỏi chế độ Edit.
SelectedIndex	Trả về chỉ số của dòng đang chọn
Rows	Một tập hợp chứa các hàng của GridView.
Columns	Một tập hợp chứa các cột của GridView.

4.4.1.2 Các style áp dụng cho GridView

GridView rất linh hoạt trong việc trình bày dữ liệu, nó cho phép ta định dạng các phần thông qua style. Ví dụ ta có thể định dạng cho phần Header, footer, các mục dữ liệu, các hàng chẵn-lẻ v.v... Bảng dưới đây sẽ giải thích rõ ý nghĩa một số thuộc tính:

Thuộc tính style	Mô tả
<u>AlternatingRowStyle</u>	Style áp dụng cho các hàng dữ liệu chẵn-lẻ trong GridView. Khi đặt thuộc tính này thì các hàng sẽ được hiển thị với định dạng luân phiên giữa <u>RowStyle</u> và <u>AlternatingRowStyle</u> .
<u>EditRowStyle</u>	Style để hiển thị hàng hiện đang được sửa (Edit).
<u>FooterStyle</u>	Style áp dụng cho phần Footer.
<u>HeaderStyle</u>	Style áp dụng cho phần Header.
<u>PagerStyle</u>	Style áp dụng cho phần phân trang (các trang << 1 2 3 ... >>).
<u>RowStyle</u>	Style áp dụng cho các hàng dữ liệu trong GridView control. Khi <u>AlternatingRowStyle</u> được thiết lập thì sẽ áp dụng luân phiên giữa <u>RowStyle</u> và <u>AlternatingRowStyle</u> .
<u>SelectedRowStyle</u>	Style áp dụng cho hàng đang được chọn (Selected) của GridView .

4.4.1.3 Các sự kiện

GridView có rất nhiều sự kiện quan trọng, các sự kiện này khi kích hoạt sẽ cung cấp cho ta những thông tin hữu ích trong quá trình xử lý. Ví dụ, khi chúng ta click nút Update, nó sẽ kích hoạt sự kiện Updating và trả về cho chúng ta các giá trị mà người dùng vừa sửa....

Dưới đây là bảng tổng hợp một số sự kiện hay dùng nhất:

Tên sự kiện	Mô tả
<u>PageIndexChanged</u>	Xuất hiện khi ta click chọn các nút (<< <u>1</u> <u>2</u> <u>3</u> >>) trong hàng phân trang.
<u>PageIndexChanging</u>	Xuất hiện khi người dùng click chọn các nút (<< <u>1</u> <u>2</u> <u>3</u> >>) trong hàng phân trang nhưng TRƯỚC khi GridView thực hiện việc phân trang. Ta có thể hủy việc phân trang tại sự kiện này.
<u>RowCancelingEdit</u>	Xuất hiện khi nút Cancel được click nhưng trước khi thoát khỏi chế độ Edit.
<u>RowCommand</u>	Xuất hiện khi một nút được click.
<u>RowCreated</u>	Xuất hiện khi một hàng mới được tạo ra. Thường được sử dụng để sửa nội dung của hàng khi nó vừa được tạo ra.
<u>RowDataBound</u>	Xuất hiện khi một hàng dữ liệu được gắn vào GridView. Tại đây ta có thể sửa đổi nội dung của hàng đó.
<u>RowDeleted</u>	Xuất hiện khi nút Delete của một hàng được click, nhưng sau khi GridView đã delete bản ghi từ nguồn.
<u>RowDeleting</u>	Xuất hiện khi nút Delete được click nhưng trước khi GridView xóa bản ghi từ nguồn. Tại đây có thể Cancel việc Delete.
<u>RowEditing</u>	Xuất hiện khi nút Edit được click, nhưng trước khi GridView về chế độ sửa.
<u>RowUpdated</u>	Xuất hiện khi nút Update được click, nhưng sau khi GridView update hàng dữ liệu.
<u>RowUpdating</u>	Xuất hiện khi nút Update được click, nhưng trước khi GridView update hàng dữ liệu.
<u>SelectedIndexChanged</u>	Xuất hiện khi nút Select của hàng được click nhưng sau khi GridView xử lý xong thao tác Select.
<u>SelectedIndexChanging</u>	Xuất hiện khi nút Select của hàng được click nhưng trước khi GridView xử lý xong thao tác Select.
<u>Sorted</u>	Xuất hiện khi Hyperlink (tiêu đề cột) được click, nhưng sau khi GridView thực hiện việc sắp xếp.
<u>Sorting</u>	Xuất hiện khi Hyperlink (tiêu đề cột) được click, nhưng trước khi GridView thực hiện việc sắp xếp. Sự kiện này khi xảy ra, nó sẽ cung cấp cho chúng ta thông tin về tên cột vừa được click. Dựa vào đó ta có thể thực hiện việc sắp xếp một cách dễ dàng.

4.4.1.4 Các phương thức

Tên phương thức	Mô tả
DataBind()	Gắn kết dữ liệu giữa GridView và nguồn dữ liệu (đặt các thuộc tính DataSource, DataTextField hoặc DataSourceID).
DeleteRow(int)	Xóa một dòng trong GridView
UpdateRow(int i, bool Valid)	Cập nhật một dòng trong GridView.
Sort(Biểu thức sx, hướng sx)	Sắp xếp dựa trên biểu thức và hướng.

4.4.1.5 Các tính năng hỗ trợ của GridView

a. Phân trang

Để thực hiện phân trang, cần đặt thuộc tính AllowPaging = True. Khi phân trang, có thể tùy biến hiển thị các trang (hiển thị dạng các số 1 2 3 hay mũi tên << >>) bằng cách đặt các thuộc tính con trong PagerSettings.

The screenshot displays a web application interface on the left and a Visual Studio Properties window on the right. The web application shows a GridView control with a blue header and footer. The header has three columns labeled 'Column0', 'Column1', and 'Column2'. The body contains four rows, each with the text 'abc'. The footer contains the text 'Next' and 'LastPage'. The Properties window on the right is titled 'Properties' and shows the configuration for 'GridView1' (System.Web.UI.WebControls.GridView). The 'PagerSettings' property is expanded, showing the following values: FirstPageImageUrl, FirstPageText (FirstPage), LastPageImageUrl, LastPageText (LastPage), Mode (NextPreviousFirstLast), NextPageImageUrl, NextPageText (Next), PageButtonCount (5), Position (TopAndBottom), PreviousPageImageUrl, PreviousPageText (Previous), and Visible (True). Other properties like ForeColor, GridLines (Vertical), Height, HorizontalAlign (NotSet), PageIndex (0), and PagerStyle (PageSize: 4, RowHeaderColumn) are also visible.

Đáp ứng với thao tác click nút chuyển trang gây ra sự kiện **PageIndexChanging** trên GridView. Bắt buộc trong phương thức ủy thác phải có dòng lệnh `GridView1.PageIndex = e.NewPageIndex;`

```
protected void GridView1_PageIndexChanging(object sender, GridViewPageEventArgs e)
{
    GridView1.PageIndex = e.NewPageIndex;
    .....
}
```

b. Tính năng tự động sắp xếp

Tính năng này cho phép dữ liệu trong GridView sẽ tự động được sắp xếp theo giá trị của cột mà người dùng click. Ở đây ta có thể sắp xếp theo chiều tăng (Ascending) hoặc giảm (Descending). Để bật tính năng này, cần đặt thuộc tính `AllowSorting = true` trong GridView.

Khi người dùng click chuột vào một cột tiêu đề nào đó của GridView thì sự kiện `Sorting` sẽ được kích hoạt, tại đây ta cần phải chỉ rõ cho GridView biết là sắp theo cột nào (`SortExpression`) và theo chiều tăng hay giảm (`SortDirection`).

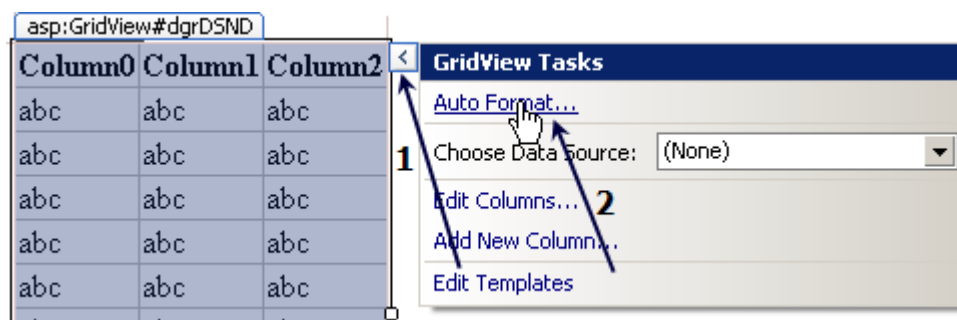
c. Các mẫu hiển thị - Template

ASP.NET cung cấp cho chúng ta sẵn một số Template (mẫu) để hiển thị GridView cũng khá đẹp. Vì vậy, bạn có thể sử dụng ngay các template này khi xây dựng ứng dụng.

Cách thức chọn template cho GridView như sau:

b1: Mở trang ở chế độ Design

b2: Chọn GridView và chọn smart tag, tiếp theo chọn `AutoFormat`



b3: Chọn Format trong danh sách.

Sau khi chọn Template, Asp sẽ tự động tạo ra các thuộc tính (thẻ) tương ứng trong GridView, tại đây bạn có thể tiếp tục tùy biến thêm theo như ý muốn.

4.4.1.6 Tạo các cột tùy biến HyperLink, BoundColumn...

a. Tạo cột BoundField thủ công

Để tạo các cột thủ công, cần đặt thuộc tính `AutoGenerateColumns = "False"`, sau đó soạn thủ công các cột trong cửa sổ `Edit Columns`.

b. Tạo một cột hyperlink

Trong ASP.NET, GridView có khả năng hiển thị (render) các trường có chứa HyperLink thành các thẻ `<a href ...>` trên trình duyệt. Do vậy, về ý tưởng chúng ta cần phải tạo một cột mới có chứa sẵn Hyperlink sau đó "Chèn" trường này vào cột Hyperlink của GridView.

Chú ý: Trong SQL, thông thường trong câu lệnh `SELECT` chúng ta chỉ chọn các trường sẵn có trong bảng CSDL, tuy nhiên hoàn toàn có thể tạo ra một cột mới kiểu như sau:

SELECT Ten + Ho as **HoVaTen** FROM ...

Trong đó: Ten, Ho là 2 trường của bảng, HoVaTen là một cột mới (Do ta tạo ra ngay trong câu lệnh SELECT, còn trong bảng CSDL thì không có trường này)

Hay một ví dụ khác:

SELECT TenHang, NgayXua, SoLuong, DonGia, SoLuong * DonGia As **ThanhTien** ...

Trường **ThanhTien** là một trường mới. Giá trị của nó bằng giá trị SoLuong * DonGia của bản ghi hiện tại.

Để tạo cột hiển thị được HyperLink, GridView cung cấp thẻ : **<asp: HyperLinkField>**

c. Tạo cột Image

Tương tự như cột HyperLink, GridView cũng có một cột chuyên để hiển thị hình ảnh (ImageField) nếu trường dữ liệu gắn với nó chứa đường dẫn đến ảnh nằm trong ứng dụng.

Để tạo cột cho phép hiển thị Image, dùng thẻ **<asp:ImageField DataImageUrlField ../>**

4.4.1.7 Tạo và xử lý các cột Select, Edit, Delete, Update...

a. Thêm cột Select, Edit - Update, Delete

GridView không chỉ hiển thị được các bảng dữ liệu mà còn hỗ trợ rất tốt trong việc chỉnh sửa và xóa dữ liệu. Đặc biệt khi nguồn dữ liệu là SqlDataSource thì việc sửa và xóa hoàn toàn tự động, không cần phải viết bất kỳ dòng code nào. Để bật tính năng này, cần bổ sung thêm thuộc tính vào GridView với giá trị là true cho **AutoGenerateSelectColumn**, **AutoGenerateEditColumn**, **AutoGenerateDeleteColumn**.

b. Cập nhật dữ liệu

Các bước cần tiến hành:

- Tạo một nguồn dữ liệu SqlDataSource
- Thêm thuộc tính UpdateCommand với câu lệnh cập nhật Sql.
- Tạo GridView và đặt thuộc tính **DataKeyNames** = "Tên trường Khóa của bảng CSDL"
- Gắn kết GridView với SqlDataSource bằng cách đặt DataSourceID của GridView = ID của SqlDataSource.

c. Xóa dữ liệu

Việc xóa cũng hoàn toàn tương tự như Update. Tức là ta cũng cần phải thêm thuộc tính DeleteCommand vào trong SqlDataSource.

Ví dụ áp dụng: Thao tác với *GridView*

* Tạo CSDL đặt tên là **QLHS**, có cấu trúc gồm 2 bảng như sau:

LOPHOC			
STT	Tên trường	Kiểu dữ liệu	Ghi chú
1	MaLop	INT	Primary Key
2	TenLop	Nvarchar (40)	
3	DienGiai	Nvarchar (60)	
4	LopChuyen	bit	
5	HoTenGVCN	Nvarchar (40)	

HOCSINH			
STT	Tên trường	Kiểu dữ liệu	Ghi chú
1	MaHocSinh	INT	Primary Key

2	TenHocSinh	Nvarchar (40)	
3	NgaySinh	Date	
4	MaLop	INT	

Nhập một số thông tin vào CSDL

1. Xây dựng trang web cập nhật thông tin bảng LopHoc:

	<u>Mã lớp</u>	<u>Tên lớp</u>	<u>Diễn giải</u>	<u>Lớp chuyên</u>	<u>Giáo viên chủ nhiệm</u>		
Chọn dòng	1	12T1	Lớp chuyên toán 1	<input checked="" type="checkbox"/>	Đinh Thị Thuý	Sửa	Xóa
Chọn dòng	2	12A	Lớp 12 A	<input checked="" type="checkbox"/>	Lê Thu Hà	Sửa	Xóa
Chọn dòng	3	12B	Lớp 12 B	<input type="checkbox"/>	Trần Thị Lan	Sửa	Xóa
1 2							

+ Tạo 1 Web Form, đặt tên là QLLH.aspx

+ Sử dụng **SqlDataSource Control**:

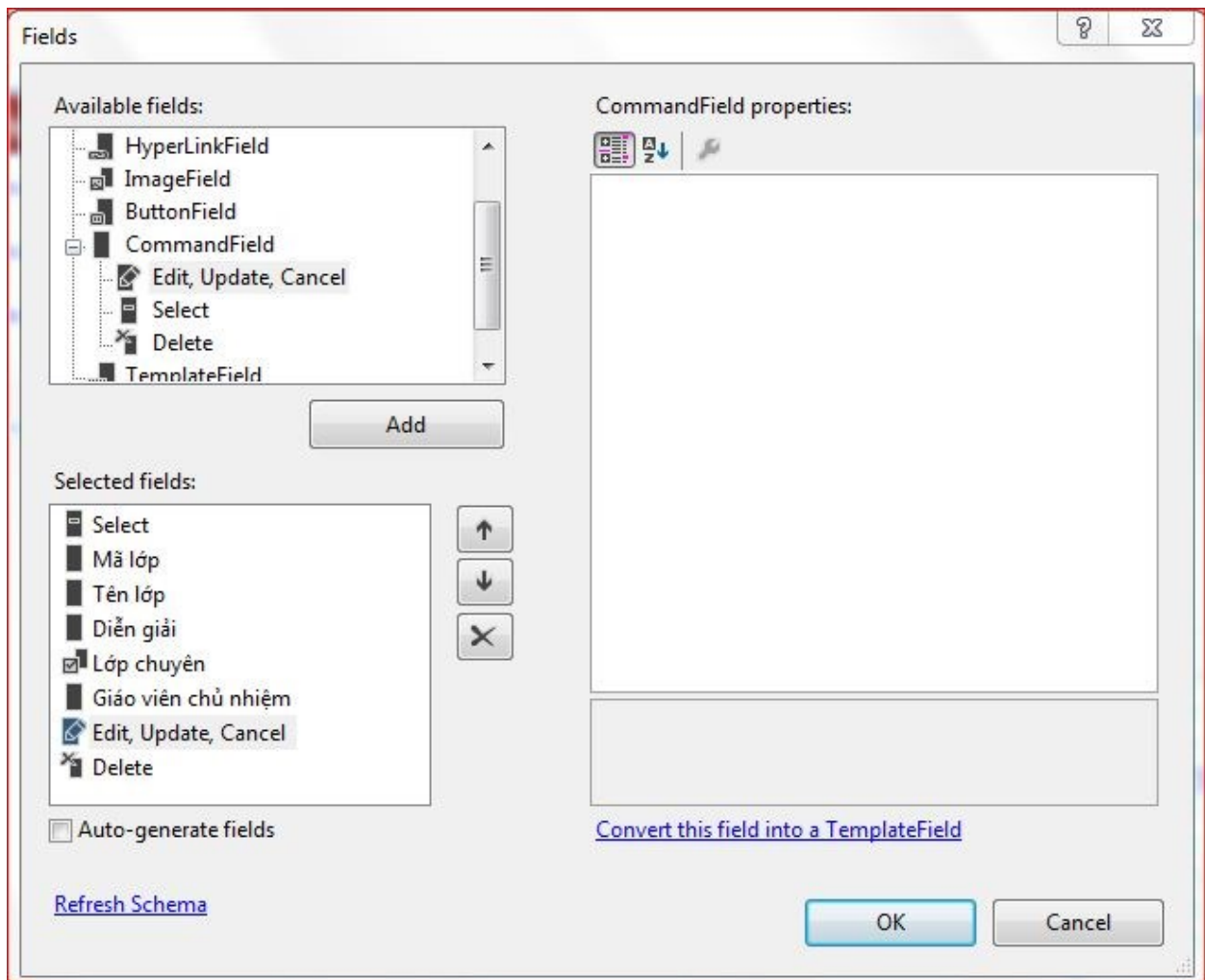
- Kéo thả một **SqlDataSource** vào QLLH.aspx tên là SqlDataSource1
- Kết nối đến tập tin QLHS
- Tạo câu truy vấn Select

- Tạo các câu truy vấn Insert, Update, Delete
 1. Bấm vào nút Advanced
 2. Chọn Generate INSERT, UPDATE, and DELETE statements , OK

+ Sử dụng **GridView Control**:

- Kéo thả một **GridView Control** vào QLLH.aspx, đặt tên là GridView1
- Chọn **Choose data source** : SqlDataSource1
- Thay đổi giao diện : phải chuột, chọn Auto format, chọn Select a scheme : chọn *Autumn*
- Điều chỉnh lại cột cho gridview : vào Property, chọn Columns...

Trong **Selected fields** có sẵn 5 cột, thêm vào 3 cột: **Select**; **Edit**, **Update**, **Cancel** và **Delete** trong **CommandField**. Thay đổi thuộc tính của các cột.



Tên cột	Thuộc tính	Giá trị
MaLop	HeaderText (tiêu đề cột)	Mã lớp
TenLop	HeaderText	Tên lớp
DienGiai	HeaderText	Diễn giải
LopChuyen	HeaderText	Lớp chuyên
HoTenGVCN	HeaderText	Giáo viên chủ nhiệm
Edit, Update, Cancel	EditText	Sửa
	UpdateText	Cập nhật
	CancelText	Bỏ qua
Delete	DeleteText	Xoá
Select	SelectText	Chọn dòng

- Thiết lập phân trang cho **GridView** : vào Properties chọn
 1. Allow paging : true (cho phép phân trang)
 2. Page size : 3 (mỗi trang chỉ hiển thị 3 dòng dữ liệu)
 - Thiết lập sắp xếp cho cột: AllowSorting: True
- =>Chạy và kiểm tra trang web

2. Xây dựng trang web tra cứu thông tin học sinh theo từng lớp khi chọn dòng

	Mã lớp	Tên lớp	Điểm giải	Lớp chuyên	Giáo viên chủ nhiệm		
Chọn dòng	1	12T1	Lớp chuyên toán 1	<input checked="" type="checkbox"/>	Đinh Thị Thuý	Sửa	Xóa
Chọn dòng	2	12A	Lớp 12 A	<input checked="" type="checkbox"/>	Lê Thu Hà	Sửa	Xóa
Chọn dòng	3	12B	Lớp 12 B	<input type="checkbox"/>	Trần Thị Lan	Sửa	Xóa
1 2							

MaHocSinh	TenHocSinh	NgaySinh	MaLop
1	Lê Thị Vân	08/03/1999	1
2	Trần Thị Dung	04/04/1998	1
3	Lê Thị Hạ	07/07/1988	1

Gợi ý:

- + Sử dụng **SqlDataSource2** kết nối với bảng **HocSinh**
- + Sử dụng **GridView2** lấy nguồn dữ liệu là **SqlDataSource2**
- + Sửa lại mã lệnh thiết kế trang của **SqlDataSource2**:

```

DANH SÁCH HỌC SINH: <br />
<asp:SqlDataSource ID="SqlDataSource2" runat="server"
ConnectionString="<%"$ ConnectionStrings:QLHSConnectionString52 %">
SelectCommand="SELECT * FROM [HOCSINH] WHERE MaLop=@MaLop">
<SelectParameters>
<asp:ControlParameter Name="MaLop" ControlID="GridView1"
PropertyName="SelectedDataKey.Value" DefaultValue="1" />
</SelectParameters>
</asp:SqlDataSource>

```

3. Xây dựng trang web tra cứu thông tin học sinh theo từng lớp (chọn lớp trong hộp lựa chọn)

Chọn lớp học: <input type="text" value="12T1"/> <input type="button" value="Tìm"/> <input type="button" value="Bỏ Tìm"/>		
Mã HS	Họ Tên HS	Ngày Sinh
1	Trần Thị Hà	20/11/1989
2	Đỗ Ngân Hà	24/11/1990
3	Nguyễn Thị Mỹ Bình	02/10/1979
1 2		

Gợi ý:

- + Sử dụng **SqlDataSource1** kết nối với bảng **LopHoc**
 - + Tạo **DropDownList1** với 3 thuộc tính:
 - ✓ DataSourceID : SqlDataSource1
 - ✓ DataTextField : TenLop (cột TenLop sẽ được hiển thị)
 - ✓ DataValueField : MaLop (cột MaLop được dùng để lấy giá trị khi 1 phần tử được chọn)
 - + Tạo **GridView1** (không lấy nguồn dữ liệu)
 - Thay đổi giao diện : phải chuột, chọn Auto format, chọn Select a scheme : chọn *Autumn*
 - Điều chỉnh lại cột cho gridview : vào Property, chọn Columns...
- Thêm 3 cột **BoundField**. Thay đổi thuộc tính của các cột.

Tên cột	Thuộc tính	Giá trị
BoundField	HeaderText (tiêu đề cột)	Mã học sinh
	DataField	MaHocSinh
BoundField	HeaderText	Họ Tên HS
	DataField	TenHocSinh
BoundField	HeaderText	Ngày sinh
	DataField	NgaySinh
	Data Format string	{0:dd/MM/yyyy}

(bỏ Auto)

* Viết mã lệnh

- Mức class
<code>string strCon=@"Data Source=.;Initial Catalog=QLHS;Integrated Security=True";</code>
<code>SqlConnection conn=null;</code>
<code>SqlDataAdapter da=null;</code>
<code>DataTable dt = null;</code>
- Page Load
<code>conn = new SqlConnection(strCon); conn.Open();</code>
- Nút Tìm
<code>string strMaLop = DropDownList1.SelectedValue;</code>
<code>string sql = "Select MaHocSinh, TenHocSinh, NgaySinh from hocsinh where malop=" + strMaLop + """;</code>
<code>da = new SqlDataAdapter(sql, conn);</code>
<code>dt = new DataTable();</code>
<code>da.Fill(dt);</code>
<code>GridView1.DataSource = dt;</code>
<code>GridView1.DataBind();</code>

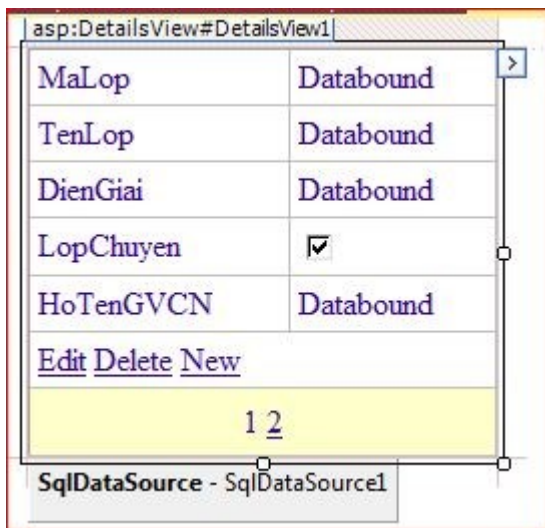
* **Lưu ý:** sinh viên tự làm cách khác: Không sử dụng **SqlDataSource1**. Viết code lấy dữ liệu từ bảng **LopHoc** cho **DropDownList1**

4.4.2 Điều khiển DetailsView

- Dùng để hiển thị dữ liệu dưới dạng cột (Columnar)
- Đặc điểm:
 - ✓ Cho phép phân trang
 - ✓ Cho phép cập nhật dữ liệu trực tiếp (thêm, sửa, xóa)

Ví dụ: (làm kéo thả như gridview không cần chỉnh sửa gì)

+ **Thiết kế giao diện trang DetailsView1.aspx:**



+ Phần mã lệnh thiết kế trang DetailsView1.aspx:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server"> <title></title> </head>
<body>
<form id="form1" runat="server">
<div>
<asp:DetailsView ID="DetailsView1" runat="server" Height="50px" Width="234px"
AllowPaging="True" AutoGenerateRows="False" BackColor="White"
BorderColor="#CC9966" BorderStyle="None" BorderWidth="1px" CellPadding="4"
DataKeyNames="MaLop" DataSourceID="SqlDataSource1">
<EditRowStyle BackColor="#FFCC66" Font-Bold="True" ForeColor="#663399" />
<Fields>
<asp:BoundField DataField="MaLop" HeaderText="MaLop" ReadOnly="True"
SortExpression="MaLop" />
<asp:BoundField DataField="TenLop" HeaderText="TenLop" SortExpression="TenLop" />
<asp:BoundField DataField="DienGiai" HeaderText="DienGiai" SortExpression="DienGiai" />
<asp:CheckBoxField DataField="LopChuyen" HeaderText="LopChuyen"
SortExpression="LopChuyen" />
<asp:BoundField DataField="HoTenGVCN" HeaderText="HoTenGVCN"
SortExpression="HoTenGVCN" />
<asp:CommandField ShowDeleteButton="True" ShowEditButton="True"
ShowInsertButton="True" />
</Fields>
<FooterStyle BackColor="#FFFFCC" ForeColor="#330099" />
<HeaderStyle BackColor="#990000" Font-Bold="True" ForeColor="#FFFFCC" />
<PagerStyle BackColor="#FFFFCC" ForeColor="#330099" HorizontalAlign="Center" />
<RowStyle BackColor="White" ForeColor="#330099" />
</asp:DetailsView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<?%$
ConnectionStrings:QLHSCConnectionString %>"
SelectCommand="SELECT * FROM [LOPHOC]"
DeleteCommand="DELETE FROM [LOPHOC] WHERE [MaLop] = @MaLop"
InsertCommand="INSERT INTO [LOPHOC] ([MaLop], [TenLop], [DienGiai],
[LopChuyen], [HoTenGVCN]) VALUES (@MaLop, @TenLop, @DienGiai, @LopChuyen,
@HoTenGVCN)"
UpdateCommand="UPDATE [LOPHOC] SET [TenLop] = @TenLop, [DienGiai] =
@DienGiai, [LopChuyen] = @LopChuyen, [HoTenGVCN] = @HoTenGVCN WHERE
[MaLop] = @MaLop">

```

```

</asp:SqlDataSource>
</div>
</form>
</body>
</html>

```

+ Kết quả thực hiện:

MaLop	1
TenLop	12T1
DienGiai	Lớp chuyên toán 1
LopChuyen	<input checked="" type="checkbox"/>
HoTenGVCN	Đinh Thị Thuý
Edit Delete New	
1 2 3 4 5 6	

MaLop	1
TenLop	12T1
DienGiai	Lớp chuyên toán 1
LopChuyen	<input checked="" type="checkbox"/>
HoTenGVCN	Đinh Thị Thuý
Update Cancel	
1 2 3 4 5 6	

MaLop	
TenLop	
DienGiai	
LopChuyen	<input type="checkbox"/>
HoTenGVCN	
Insert Cancel	

4.4.3 Điều khiển DataList

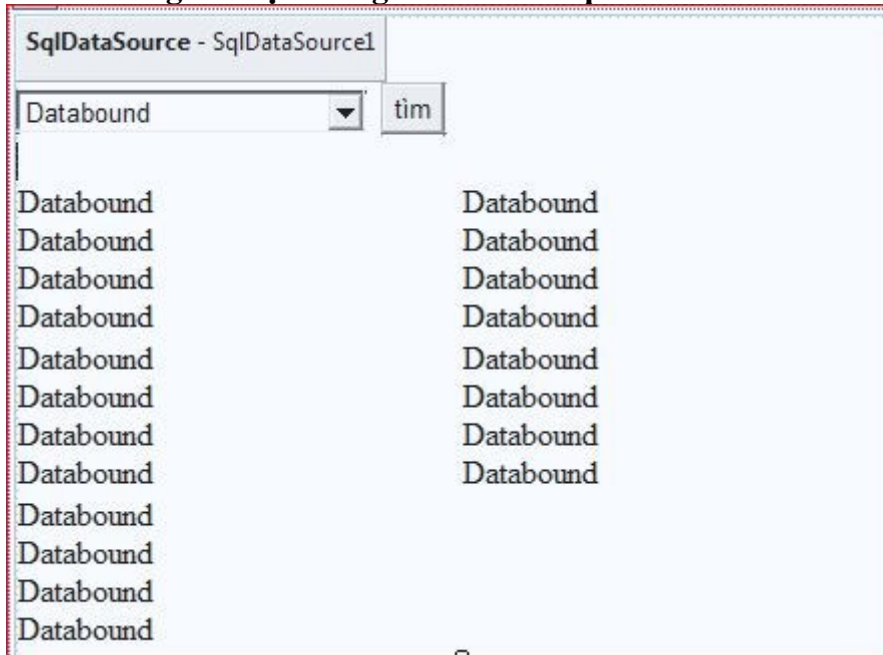
DataList được sử dụng để hiển thị dữ liệu là một mẫu tin tại một thời điểm. Sử dụng thẻ **ItemTemplate** để tự thiết kế ra hình thức hiển thị dữ liệu. Một số thuộc tính hay dùng:

- ✓ **RepeatColumns**: Qui định số cột hiển thị của DataList.

- ✓ **RepeaterDirection**: hướng để hiển thị các ô, có thể giá trị là **Horizontal** (chiều ngang) hoặc **Verical** (chiều đứng – mặc định)

Ví dụ:

+ **Thiết kế giao diện trang DataList1.aspx:**



+ **Phần mã lệnh thiết kế trang DataList1.aspx:**

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server"> <title></title> </head>
<body>
<form id="form1" runat="server">
<div>
<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$
ConnectionStrings:QLHSCConnectionString %>" SelectCommand="SELECT * FROM
[LOPHOC]">
</asp:SqlDataSource>
<asp:DropDownList ID="DropDownList1" runat="server" DataSourceID="SqlDataSource1"
DataTextField="TenLop" DataValueField="MaLop" Height="16px" Width="174px">
</asp:DropDownList> &nbsp;
<asp:Button ID="cmdTim" runat="server" Text="tìm" OnClick="cmdTim_Click" />
<br />
<asp:DataList ID="DataList1" runat="server" RepeatColumns="2"
RepeatDirection="Horizontal" Width="220px">
<ItemTemplate>
<asp:Label ID="Label1" runat="server" Text="<%# Eval("MaHocSinh") %>"
Width="220px"></asp:Label> <br />
<asp:Label ID="Label2" runat="server" Text="<%# Eval("TenHocSinh") %>"
Width="220px"></asp:Label> <br />
<asp:Label ID="Label3" runat="server" Text="<%# Eval("NgaySinh") %>"
Width="220px"></asp:Label> <br />
<asp:Label ID="Label4" runat="server" Text="<%# Eval("MaLop") %>"
Width="220px"></asp:Label>
</ItemTemplate>
</asp:DataList> <br />
</div>
```

```
</form>
</body>
</html>
```

+ Phần mã lệnh thực thi trang DataList1.aspx.cs:

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

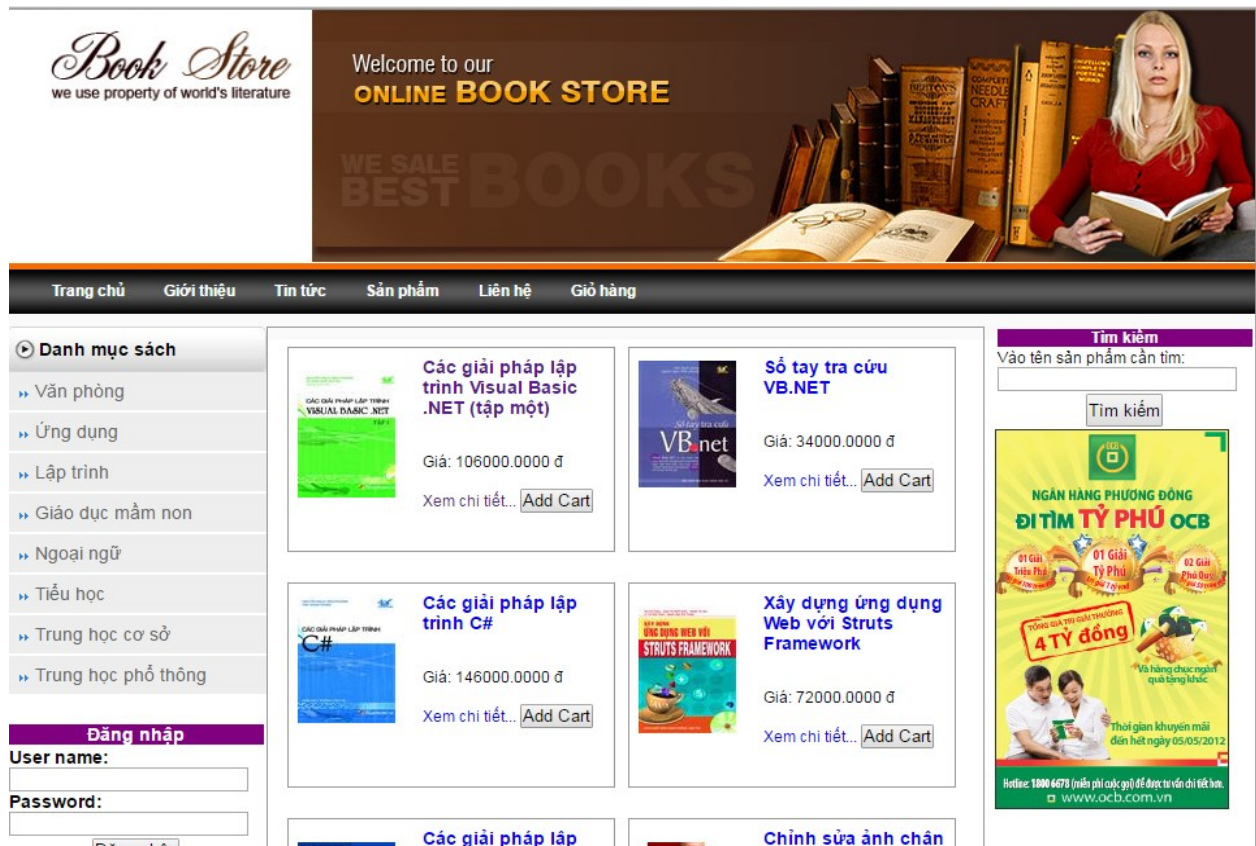
public partial class WebForm1 : System.Web.UI.Page
{
    string strCon = @"Data Source=.;Initial Catalog=QLHS;Integrated Security=True";
    SqlConnection conn = null;
    SqlDataAdapter da = null;
    DataTable dt = null;

    protected void cmdTim_Click(object sender, EventArgs e)
    {
        string ma = DropDownList1.SelectedValue;
        conn = new SqlConnection(strCon); conn.Open();
        string sql = "Select * from HocSinh where MaLop='" + ma + "'";
        da = new SqlDataAdapter(sql, conn);
        dt = new DataTable();
        da.Fill(dt);
        DataList1.DataSource = dt;
        DataList1.DataBind();
    }
}
```

* **Lưu ý:** sinh viên tự làm cách khác: Không sử dụng **SqlDataSource1**. Viết code lấy dữ liệu từ bảng **LopHoc** cho **DropDownList1**

BÀI TẬP

Bài 1: Tạo trang **Product.aspx** hiển thị các sách theo mẫu (Dùng DataList)



DataList → Table

<p>A/HyperLink (Photo)</p>	<p>+ A/HyperLink (Title)</p> <p>+ Literal/Label (Price)</p> <p>+ A/HyperLink (Xem chi tiết...)</p> <p>+ Button (Add Cart)</p>
-----------------------------------	---

*Products.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Page/FrontEnd.master"
AutoEventWireup="true" CodeFile="Product.aspx.cs" Inherits="Page_Product" %>

<asp:Content ID="Content1" ContentPlaceHolderID="main_body" Runat="Server">
<asp:DataList ID="DataList1" runat="server" DataKeyField="Id" RepeatColumns="2"
Width="520px" >
<ItemTemplate>
<table class="myBox">
<tr>
<td style="width: 97px">
<asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl='<%# "DetailsProducts.aspx?Id=" + Eval("Id").ToString() %>'
ImageUrl='<%# Eval("Photo") %>'
ToolTip="Xem chi tiết..." >
</asp:HyperLink>
</td>
<td class="name" style="width: 150px" >
<h3>
<asp:HyperLink ID="HyperLink2" runat="server"
NavigateUrl='<%# "DetailsProducts.aspx?Id=" + Eval("Id").ToString() %>'
Text='<%# Eval("Title") %>'
ToolTip="Xem chi tiết..." >
</asp:HyperLink>
</h3>
<br /> <br />
Giá:
<asp:Literal ID="Literal1" runat="server"
Text='<%# Eval("Price") %>' /> đ
<br /> <br />
<asp:HyperLink ID="hyperLink3" runat="server"
NavigateUrl='<%# "DetailsProducts.aspx?Id=" + Eval("Id").ToString() %>'>
Xem chi tiết...
</asp:HyperLink>
<asp:Button ID="ButCart" runat="server" CommandName="AddToCart"
Text="Add Cart" />
</td>
</tr>
</table> <br />
</ItemTemplate>
</asp:DataList>
</asp:Content>
```

*Products.aspx.cs

```
using System.Data; // them vao
using System.Data.SqlClient; // them vao

public partial class Page_Product : System.Web.UI.Page
{
    string strCon = @ "Data Source=.;Initial Catalog=WebShop;Integrated Security=True";
```

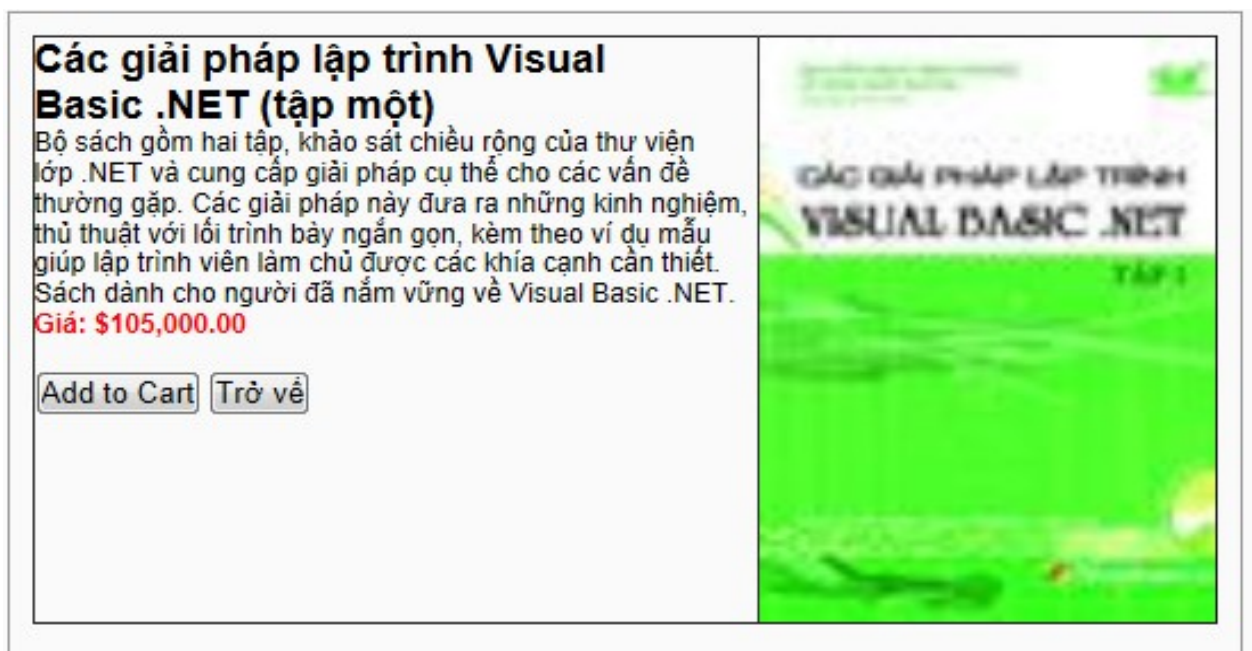


```

SqlConnection conn = null;
SqlDataAdapter da = null;
DataTable dt = null;
protected void Page_Load(object sender, EventArgs e)
{
    conn = new SqlConnection(strCon); conn.Open();
    string sql = "Select top 8 * from product";
    da = new SqlDataAdapter(sql, conn);
    dt = new DataTable();
    da.Fill(dt);
    DataList1.DataSource = dt;
    DataList1.DataBind();
}
}

```

Bài 2: Tạo trang **DetailsProducts.aspx** xem chi tiết sách theo mẫu (Dùng DataList)



Table

	Image/HyperLink (Photo)
+Literal/Label (Title)	
+Literal/Label (Description)	
+Literal/Label (Price)	
+2 Button	

*DetailsProducts.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Page/FrontEnd.master"
AutoEventWireup="true" CodeFile="DetailsProducts.aspx.cs" Inherits="Page_DetailsProducts"
%>
<asp:Content ID="Content1" ContentPlaceHolderID="main_body" Runat="Server">
    <asp:DataList ID="DataList1" runat="server" Width="520px">
        <ItemTemplate>
            <table style="width: 100%;">
                <tr>
                    <td style="width: 320px" valign="top" align="justify" >
                        <h2><asp:Literal ID="TitleLabel" runat="server" Text='<%# Eval("Title")
%>'></asp:Literal>&nbsp;  </h2>
                        <p>
                            <asp:Literal ID="DescriptionLabel" runat="server" Text='<%# Eval("Description")
%>'></asp:Literal>
                        </p>
                        <p style="color:Red; font-weight:bold ">
                            Giá:
                            <asp:Label ID="PriceLabel" runat="server" Text='<%# String.Format("{0:c}",
Eval("Price")) %>' ></asp:Label>
                        </p>
                        <br />
                        <p>
                            <asp:Button ID="btnAddToCart" runat="server" Text = "Add to Cart" />
                            <input type= "button" value="Trở về" onclick="history.go(-1);" /></p>
                        <p>
                            &nbsp;</p>
                    </td>
                    <td style="width: 200px; ">
                        <asp:Image ID="Image1" runat="server" ImageUrl='<%# Eval("Photo") %>'
Width="200px" />
                    </td>
                </tr>
            </table>
            <br />
            <br />
        </ItemTemplate>
    </asp:DataList>
</asp:Content>
```

*DetailsProducts.aspx.cs

```
using System.Data; // them vao
using System.Data.SqlClient; // them vao
public partial class Page_DetailsProducts : System.Web.UI.Page
{
    string strCon = @"Data Source=.;Initial Catalog=WebShop;Integrated Security=True";
    SqlConnection conn = null;
    SqlDataAdapter da = null;
    DataTable dt = null;
    protected void Page_Load(object sender, EventArgs e)
    {
```

```


    int ID = Int32.Parse(Request.QueryString["Id"].ToString());
    conn = new SqlConnection(strCon); conn.Open();
    string sql = "Select * from product where Id='" + ID + "'";
    da = new SqlDataAdapter(sql, conn);
    dt = new DataTable();
    da.Fill(dt);
    DataList1.DataSource = dt;
    DataList1.DataBind();
}
}

```

Bài 3: Tạo trang **News.aspx** hiển thị các tin tức theo mẫu:



Bài 4: Tạo trang **DetailNews.aspx** hiển thị chi tiết tin tức theo mẫu sau:

Nga dọa giáng cho Ukraine một đòn kinh tế	
Tổng thống Nga Putin hôm qua đe dọa thu tiền trước khi bán gas cho Ukraine, hành động có thể khiến kinh tế của nước láng giềng, vốn đang bên bờ phá sản, suy sụp.	
	
<p>Phát biểu trong một cuộc họp với các bộ trưởng hôm qua, Putin cho rằng việc thu tiền trước bán khí đốt say là "phù hợp với các điều khoản của hợp đồng" giữa Nga và Ukraine. Tuy nhiên, ông nói thêm rằng tập đoàn năng lượng Nga Gazprom có thể chưa cần thực hiện các biện pháp mạnh trước khi có các kết quả tham vấn bổ sung giữa hai nước. Nga đã loại bỏ cơ chế giảm giá gas bán cho Ukraine bởi cơ chế này gắn liền với việc Nga thuê quân cảng ở bán đảo Crimea, nay đã được Nga sáp nhập. Ukraine cũng đã cam kết với IMF về việc xóa bỏ trợ giá khí đốt trong nước để đổi lấy khoản vay 14 tỷ USD cứu nền kinh tế. Trước đó, Bộ trưởng Năng lượng Ukraine Yuri Prodan nói nước này sẽ chỉ trả tiền khí đốt tháng ba sau khi hai nước thỏa thuận xong về giá. Kiev không đồng ý mức giá mới cao hơn hẳn mà Gazprom đưa ra. Tổng thống Putin cho biết cho dù Nga không công nhận chính phủ ở Kiev, nước này vẫn tiếp tục viện trợ kinh tế cho nước láng giềng, tuy nhiên việc viện trợ không kéo dài mãi mãi, mà phải tùy thuộc vào các diễn biến ở Ukraine, theo Ria Novosti. Nga đã ngừng nhập khẩu sữa của 6 công ty sữa Ukraine với các lý do kỹ thuật, sau khi Kiev tuyên bố không cho phép nhiều công ty lương thực của Nga bán hàng vào Ukraine. Kiev và Moscow cũng ngừng một số hợp đồng về khí tài quân sự. Ukraine là nhà cung cấp phụ tùng kỹ thuật quân sự lớn của Nga. Ảnh Dương</p>	

Bài 5: Tạo trang **Loaisach.aspx** hiển thị các sách khi chọn loại sách từ menu bên trái theo mẫu:



DataList → Table

A/HyperLink (Title)	
A/HyperLink (Photo)	+ Literal/Label (Price) + A/HyperLink (Xem chi tiết...)

***Loaisach.aspx (kế thừa từ FrontEnd.master)**

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Page/FrontEnd.master"
AutoEventWireup="true" CodeFile="Loaisach.aspx.cs" Inherits="Page_Laptrinh" %>

<asp:Content ID="Content1" ContentPlaceHolderID="main_body" Runat="Server">
<asp:DataList ID="DataList1" runat="server" RepeatColumns="3" Width="520px">
  <ItemTemplate>
    <table class="myBox1">
      <tr>
        <td class="name" style="width: 170px" colspan="2" >
          <h3>
            <asp:HyperLink ID="HyperLink4" runat="server"
              NavigateUrl='<%# "DetailsProducts.aspx?Id=" + Eval("Id").ToString()
%>'
              Text='<%# Eval("Title") %>'
              ToolTip="Xem chi tiết..." >
            </asp:HyperLink>
          </h3>
        </td>
      </tr>
    </table>
  </ItemTemplate>
</asp:DataList>
```

```

        </tr>
        <tr>
            <br /> <br />
            <td style="width: 80px">
                <asp:HyperLink ID="HyperLink1" runat="server"
                    NavigateUrl='<%# "DetailsProducts.aspx?Id=" + Eval("Id").ToString()
%>'
                    ImageUrl='<%# Eval("Photo") %>'
                    ToolTip="Xem chi tiết..." >
                </asp:HyperLink>
            </td>
            <td class="name" style="width: 90px" >
                <h4 style="color:Red ">
                    Giá:
                    <asp:Literal ID="Literal1" runat="server"
                        Text='<%#String.Format("{0:000,0 đ}", Eval("Price")) %>' />
                </h4>
                <br /> <br />

                <asp:HyperLink ID="hyperLink3" runat="server"
                    NavigateUrl='<%# "DetailsProducts.aspx?Id=" + Eval("Id").ToString()
%>' >
                    Xem chi tiết...
                </asp:HyperLink><br />
            </td>
        </tr>
    </table>
    <br />
</ItemTemplate>
</asp:DataList>
</asp:Content>

```

*Loaisach.aspx.cs

```

using System.Data; // them vao
using System.Data.SqlClient; // them vao

public partial class Page_Product : System.Web.UI.Page
{
    string strCon = @"Data Source=.;Initial Catalog=WebShop;Integrated Security=True";
    SqlConnection conn = null;
    SqlDataAdapter da = null;
    DataTable dt = null;
    protected void Page_Load(object sender, EventArgs e)
    {
        int ID = Int32.Parse(Request.QueryString["CategoryId"].ToString());
        conn = new SqlConnection(strCon); conn.Open();
        string sql = "Select * from product where CategoryId='" + ID + "'";
        da = new SqlDataAdapter(sql, conn);
        dt = new DataTable();
        da.Fill(dt);
        DataList1.DataSource = dt;
        DataList1.DataBind();
    }
}

```


}
*Sửa Style.css (.myBox1)

Bài 6: Tạo trang **SearchProduct.aspx** để hiển thị các sách tìm được khi gõ vào *từ khóa tiêu đề sách* trên Form tìm

Book Store
we use property of world's literature

Welcome to our
ONLINE BOOK STORE

WE SALE BEST BOOKS

Trang chủ Giới thiệu Tin tức Sản phẩm Liên hệ Giỏ hàng

Danh mục sách

- » Văn phòng
- » Ứng dụng
- » Lập trình
- » Giáo dục mầm non
- » Ngoại ngữ
- » Tiểu học
- » Trung học cơ sở
- » Trung học phổ thông

Đăng nhập

User name:

Password:

Các giải pháp lập trình Visual Basic .NET (tập một)
Giá: 106000.0000 đ
[Xem chi tiết...](#)

Các giải pháp lập trình ASP.NET 2.0 (tập một)
Giá: 75000.0000 đ
[Xem chi tiết...](#)

Các giải pháp lập trình C#
Giá: 146000.0000 đ
[Xem chi tiết...](#)

Thủ thuật lập trình Visual Basic 6
Giá: 72000.0000 đ
[Xem chi tiết...](#)

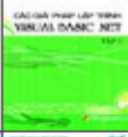

Tìm kiếm
Vào tên sản phẩm cần tìm:

Ngân hàng Phương Đông
ĐI TÌM TỶ PHÚ OCB
01 Giải Triệu Phú
01 Giải Tỷ Phú
02 Giải Phú Quý
Tổng giá trị giải thưởng: 4 TỶ đồng
Thời gian khuyến mãi đến hết ngày 05/05/2012
Hotline: 1800 6670 (miễn phí cuộc gọi) để được tư vấn chi tiết hơn.
www.ocb.com.vn

Hướng dẫn:

Bài 7: Tạo trang thêm dữ liệu vào bảng **Product** theo mẫu:

Tiêu đề sách:	<input type="text"/>
Nội dung tóm tắt:	<input type="text"/>
Giá sách:	<input type="text"/>
Số lượng:	<input type="text"/>
Loại sách:	Văn phòng <input type="button" value="v"/>
Hình ảnh:	<input type="text"/> <input type="button" value="Browse..."/>
<input type="button" value="Thêm"/> <input type="button" value="Nhập lại"/>	

Tiêu đề sách	Giá sách	Hình ảnh	Chi tiết
Các giải pháp lập trình Visual Basic .NET (tập một)	107000.0000		Xem Chi tiết
Các giải pháp lập trình C#	136000.0000		Xem Chi tiết

12345678

*InsertProduct.aspx

+ **Nhập dữ liệu:** Sử dụng các điều khiển: Table, Textbox, DropDownList, FileUpload và Button.

- Tạo SqlDataSource lấy dữ liệu từ **Category**
- Control **DropDownList**. Đặt 3 thuộc tính:
 - DataSourceID: SqlDataSource1
 - DataTextField: Name
 - DataValueField: Id

+ **Hiển thị trên GridView:**

//Tạo **SqlDataSource** lấy nguồn từ bảng **Product**

```
<asp:SqlDataSource ID="SqlDataSource2" runat="server"
    ConnectionString="<%$ ConnectionStrings:WebShopInsert %>"
    SelectCommand="SELECT
    Id,Title,Price,Photo, 'UpdateProduct.aspx' as
    ChiTiet FROM [Product]">
</asp:SqlDataSource>
```

```
<asp:GridView ID="GridView1" runat="server"
    DataSourceID="SqlDataSource2"
    AllowPaging="true" PageSize="4"
    PagerStyle-HorizontalAlign="Center" Width="520px"
    AutoGenerateColumns="False" DataKeyNames="Id" >
    <Columns>
    <asp:BoundField DataField="Title" HeaderText="Tiêu đề sách" />
    <asp:BoundField DataField="Price" HeaderText="Giá sách" />
```

```

        <asp:ImageField DataImageUrlField="Photo" HeaderText="Hình ảnh" ControlStyle-width="80" />
        <asp:HyperLinkField DataNavigateUrlFields="ChiTiet" HeaderText="Chi tiết" Text="Xem Chi tiết" />
    </Columns>
</asp:GridView>    <br />    <br />

```

* InsertProduct.aspx.cs

```



public partial class Page_InsertProduct : System.Web.UI.Page
{
    string strCon = @"Data Source=.;Initial Catalog=WebShop;Integrated Security=True";
    SqlConnection conn = null;
    SqlDataAdapter da = null;
    DataTable dt = null;
    SqlCommand cmd = null;
    protected void Page_Load(object sender, EventArgs e)
    {
        conn = new SqlConnection(strCon); conn.Open(); }
    private void SaveFileUpload()
    {
        string strFilePath;
        if (FileUploadAnh.FileName!="")
        {
            strFilePath = Server.MapPath("../images/products/" + FileUploadAnh.FileName);
            FileUploadAnh.PostedFile.SaveAs(strFilePath);
        }
    }
    protected void ButThem_Click(object sender, EventArgs e)
    {
        string sTieude=txtTieude.Text;
        string sNoidung=txtNoidung.Text;
        string sGia=txtGia.Text;
        string sSoluong=txtSoluong.Text;
        string sLoai=ListLoaisach.SelectedValue;
        string sAnh = FileUploadAnh.FileName;

        string strCmd = "INSERT INTO PRODUCT(Title,Description,Price,Quantity,CategoryId,Photo)
VALUES('"+ sTieude + "','"+ sNoidung + "','"+ sGia + "','"+ sSoluong + "','"+ sLoai + "','"+
"~/images/products/" + sAnh + "')";
        SaveFileUpload();
        cmd = new SqlCommand(strCmd,conn);
        cmd.ExecuteNonQuery();
    } }

```

Bài 8: Tạo trang **UpdateProduct.aspx** cho phép sửa, xóa, lưu dữ liệu trong Gridview theo mẫu:(khi click chọn liên kết **Xem chi tiết** trong bài 7)

*Sử dụng **TemplateField** để tạo cột tùy biến

Thông tin sách	Hình ảnh	Sửa	Xóa
Các giải pháp lập trình Visual Basic .NET (tập một) Bộ sách gồm hai tập, khảo sát chiều rộng của thư viện lớp .NET và cung cấp giải pháp cụ thể cho các vấn đề thường gặp. Các giải pháp này đưa ra những kinh nghiệm, thủ thuật với lối trình bày ngắn gọn, kèm theo ví dụ mẫu giúp lập trình viên làm chủ được các khía cạnh cần thiết. Sách dành cho người đã nắm vững về Visual Basic .NET. Giá: 107000.0000đ		Sửa	Xóa
Tiêu đề: Các giải pháp lập trình C# Nội dung tóm tắt: Quyển sách này khảo sát chiều rộng của thư viện lớp .NET và cung cấp giải pháp cụ thể cho các vấn đề thường gặp. Mỗi giải pháp được trình bày theo dạng "vấn đề/giải pháp" một cách ngắn gọn và kèm theo là các ví Giá sách: 136000.0000		Lưu Hủy	

*UpdateProduct.aspx

```

<asp:Content ID="Content1" ContentPlaceHolderID="main_body"
Runat="Server">
<p>
    <asp:SqlDataSource ID="SqlDataSource1" runat="server"
        ConnectionString="<%$ ConnectionStrings:WebShop_Product
%>"
        SelectCommand="SELECT * FROM [Product]"
        UpdateCommand="Update Product SET
            Title=@Title,Description=@Description,Price=@Price Where
            Id=@Id"
        DeleteCommand="Delete Product Where Id=@Id" >
    </asp:SqlDataSource>
</p>
<p>
    <asp:GridView ID="GridView1" runat="server"
        DataSourceID="SqlDataSource1"
        AutoGenerateColumns="False" Width="100%"
        DataKeyNames="Id" >
    <Columns>
        <asp:TemplateField HeaderText="Thông tin sách">
            <ItemTemplate>
                <h2 style="color:Blue; font-weight:bold"><%# Eval("Title")
%></h2><br/>
                <p align="justify"> <%# Eval("Description") %> </p> <br />
                <h2 style="color:Red; font-weight:bold">Giá: <%#
Eval("Price") %>đ</h2>
            </ItemTemplate>
            <EditItemTemplate>
                <b>Tiêu đề:</b><br />

```

```

        <asp:TextBox runat="server" ID="txtTD" Width="98%"
            TextMode="MultiLine" Rows="3" Text='<%# Bind("Title")
%>' >
    </asp:TextBox>
    <b>Nội dung tóm tắt:</b><br />
    <asp:TextBox runat="server" ID="txtND" Width="98%"
        TextMode="MultiLine" Rows="7" Text='<%# Bind("Description")
%>' >
    </asp:TextBox>
    <b>Giá sách:</b><br />
    <asp:TextBox runat="server" ID="txtG" Width="50%"
        Text='<%# Bind("Price") %>' > </asp:TextBox>
    </EditItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Hình ảnh">
    <ItemTemplate>
        <asp:Image runat="server" Width="150px"
            ImageUrl='<%# Eval("Photo") %>' />
    </ItemTemplate>
</asp:TemplateField>
<asp:CommandField ShowEditButton="True" CancelText="Hủy"
    EditText="Sửa"
        HeaderText="Sửa" UpdateText="Lưu" />
<asp:CommandField ShowDeleteButton="True" HeaderText="Xóa"
    DeleteText="Xóa" />
</Columns>
</asp:GridView>
</p>
<p><a href="InsertProduct.aspx">&lt; Quay lại</a> </p>
</asp:Content>

```

***UpdateProduct.aspx.cs (không cần)**