

Bài thực hành 11

1. Tạo khai báo biến toàn cục (sử dụng cho những biến dùng lại nhiều lần)
Ví dụ khai báo chuỗi kết nối

Cách làm:

Bước 1: vào file Web.config

Thêm thẻ <connectionStrings> chứa đường dẫn chuỗi kết nối theo cấu trúc

```
<add name="key" connectionString= "chuỗi kết nối"/>
```

Cụ thể:

```
<connectionStrings>
  <add name="connec" connectionString= "chuỗi kết nối"/>
</connectionStrings>
```

Kết nối bằng SQL bên ngoài

```
<connectionStrings>
  <add name="Tên tự đặt" connectionString="Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\DienThoai.mdf;Integrated
Security=True;User Instance=True" providerName="System.Data.SqlClient" />
</connectionStrings>
```



Tên CSDL

Bước 2: sử dụng

Vào các trang aspx có sử dụng kết nối database ta khai báo như sau

```
string stcn = ConfigurationManager.ConnectionStrings["key"].ConnectionString;
```

Lưu ý: khai báo thêm thư viện using System.Configuration; để sử dụng đối tượng ConfigurationManager

⇒ **Các Em thực hiện lại bài tập vừa rồi,**

2. Thực hiện chức năng Quản trị (Thêm, Xóa, Sửa) cho website

Ví dụ quản trị cho Loại hàng

Bước 1: Tạo Webform ThemLoaihang.aspx

Bao gồm 1 Control GridView

Bước 2: Viết class để xử lý một số chức năng chung

Chọn project -> Add -> New item -> chọn class (c#) đặt tên tùy ý (gợi ý ketnoi.cs) và code như sau

(Lưu ý: tất cả các hàm phải để trong class ketnoi)

```
//Thêm 2 thư viện sau để làm việc với CSDL
using System.Data;
using System.Data.SqlClient;
```

```
//khai báo 1 biến toàn cục dùng để kết nối với SQL
SqlConnection con;
```

Viết các hàm xử lý

```
//tạo phương thức lấy kết nối
private void layknoi()
{
    //khởi tạo giá trị cho đối tượng SqlConnection
    con = new SqlConnection(@"chuỗi kết nối");
    con.Open();//mở kết nối
```

```

    }
    //xây dựng hàm đóng kết nối
    private void dongketnoi()
    {
        if (con.State == ConnectionState.Open)//nếu kết nối đang mở
            con.Close();
    }
    //cập nhật dữ liệu
    public int capnhat(string sql)
    {
        int kq = 0;
        try
        {
            layknoi();
            SqlCommand cmd = new SqlCommand(sql, con);
            kq = cmd.ExecuteNonQuery();//thực thi câu len không cần truy vấn
        }
        catch
        {
            kq = 0;
        }
        finally
        {
            dongketnoi();
        }
        return kq;
    }

    //tạo hàm lấy dữ liệu có kiểu dữ liệu là DataTable, giá trị truyền vào là 1 chuỗi sql
    public DataTable laydata(string sql)
    {
        DataTable dt = new DataTable();
        try
        {
            layknoi();//mở kết nối
            SqlDataAdapter adap = new SqlDataAdapter(sql, con);
            adap.Fill(dt);//đổ dữ liệu vào dt
        }
        catch
        {
            dt = null;
        }
        finally
        {
            dongketnoi();
        }
        return dt;//kết quả trả về là 1 DataTable
    }
}

```

Bước 2: Kết nối dữ liệu vào Control GridView

```

ketnoi kn = new ketnoi();//khởi tạo kết nối
protected void Page_Load(object sender, EventArgs e)
{
    if (IsPostBack == false)
    {
        //Gán chuỗi kết nối cho dataSource của Control(GridView1)
        GridView1.DataSource = kn.laydata("SELECT * FROM Loaihang");
        GridView1.DataBind();//load dữ liệu lên đối tượng
    }
}

```

Bước 3: hiện dữ liệu lên giao diện

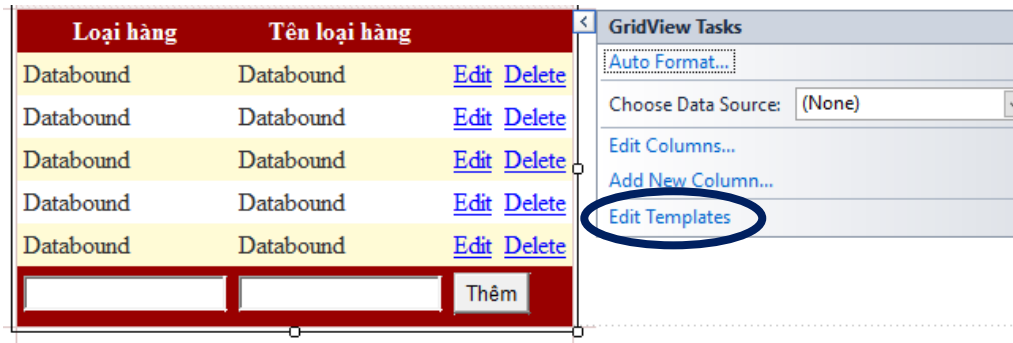
Các em tự làm nhé (cho dữ liệu vào cột Mã hàng, Loại hàng của GridView)

Bước 4: thực hiện chức năng Insert

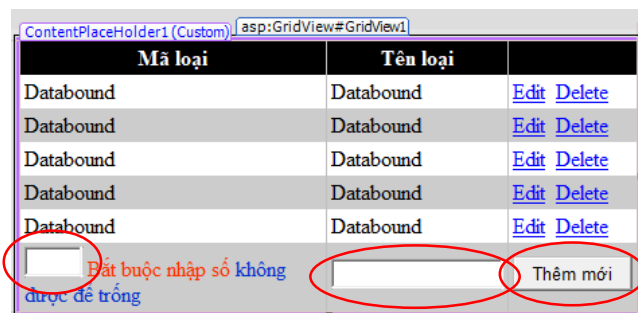
Tạo nút Insert:

Chọn GridView, vào Properties

- Chọn ShowFooter để hiện thêm 1 dòng dưới cùng trong GridView
- Thêm 1 cột để chứa các chức năng sửa, xóa:
Chọn Add New Columns -> Choose a field type chọn : CommandField, click chọn: Delete, Edit/Update -> OK
- Thêm các control để xử lý chức năng Thêm:
Chọn Edit Columns -> chọn các cột để thêm vào template



Thêm các control (Textbox, Button) để xử lý chức năng insert vào sau thẻ `</ItemTemplate>` của các đối tượng:



Lưu ý: nhớ đặt id cho các đối tượng vừa thêm vào

Ví dụ: thêm đối tượng button

- ```
</ItemTemplate>
<FooterTemplate>
 <asp:Button ID="btnThem" runat="server" Text="Thêm mới"
 onclick="btnThem_Click" />
</FooterTemplate>
```
- Nên thêm các ràng buộc (validation)

Xem kết quả

Loại hàng	Tên loại hàng	
1	iphone	<a href="#">Edit</a> <a href="#">Delete</a>
2	samsung	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="text"/>	<input type="text"/>	Thêm

- Xử lý nút Thêm mới:  
Tạo 1 hàm chuyên làm công việc thêm, sửa, xóa

Mở file class (**ketnoi.cs**) thêm hàm sau vào

```
public int xuly(string sql)
{
```

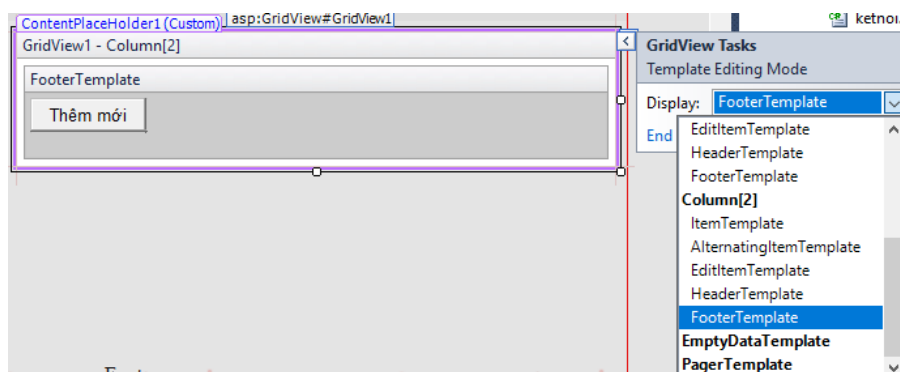
```

int kq = 0;
try
{
 layknoi();
 SqlCommand cmd = new SqlCommand(sql, con);
 kq = cmd.ExecuteNonQuery();//thuc thi cau len khong can truy van
}
catch
{
 kq = 0;
}
finally
{
 dongknoi();
}
return kq;
}

```

Bước 5: Code chức năng cho **nút insert**

Chọn Edit Templates



Kích đúp vào nút Thêm mới để viết code

```

protected void btnThem_Click(object sender, EventArgs e)
{
 TextBox txtMaloi = (TextBox)GridView1.FooterRow.FindControl("txtMaloi");
 TextBox txtTenloai = (TextBox)GridView1.FooterRow.FindControl("txtTenloai");
 string maloi = txtMaloi.Text;
 string tenloai = txtTenloai.Text;
 int kq = kn.xuly("insert into Loaihang values ('" + maloi + "', '" + tenloai +
 "')");
 if (kq > 0)//neu cap nhat duoc thi hien thong bao
 {
 Response.Write("<script>alert('cap nhat thanh công');</script>");
 GridView1.DataSource = kn.laydata("SELECT * FROM Loaihang");
 GridView1.DataBind();
 }
 else
 {
 Response.Write("<script>alert('cap nhat không thành công');</script>");
 }
}

```

- Xử lý cho chức năng Xóa:

Kích chọn GridView -> vào properties -> chọn nút Events, tìm đến thuộc tính RowDeleting nhấp chuột và enter

```

protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
 string maloi = e.Values["MaLoai"].ToString();
 int kq = kn.capnhat("delete from Loai where MaLoai = "+ maloi);
 if (kq > 0)//neu cap nhat duoc thi hien thong bao
 {
 Response.Write("<script>alert('Xóa thành công');</script>");
 GridView1.DataSource = kn.laydata("SELECT Loai.* FROM Loai");
 GridView1.DataBind();
 }
}

```

```

else
{
 Response.Write("<script>alert('Xóa không thành công');</script>");
}
}

```

Tương tự xử lý cho sự kiện sửa

```

protected void GridView1_RowEditing(object sender, GridViewEditEventArgs e)
{
 GridView1.EditIndex = e.NewEditIndex;
 GridView1.DataSource = abc.laydata("SELECT * FROM Loai");
 GridView1.DataBind();
}

////////////////////////////////////

protected void GridView1_RowCancelingEdit(object sender, GridViewCancelEditEventArgs e)
{
 GridView1.EditIndex = -1; //không lấy giá trị cột nào hết
 GridView1.DataSource = kn.laydata("SELECT * FROM Loai");
 GridView1.DataBind();
}

////////////////////////////////////

protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
 string maloai = e.NewValues["MaLoai"].ToString();
 string tenloai = e.NewValues["TenLoai"].ToString();
 int kq = kn.capnhat("update Loai set MaLoai= '" + maloai + "', TenLoai='" +
tenloai + "' where MaLoai='" + maloai + "'");
 if (kq > 0) //neu cap nhat duoc thi hien thong bao
 {
 Response.Write("<script>alert('Cập nhật thành công');</script>");
 GridView1.DataSource = kn.laydata("SELECT * FROM Loai");
 GridView1.EditIndex = -1;
 GridView1.DataBind();
 }
 else
 {
 Response.Write("<script>alert('Cập nhật không thành công');</script>");
 }
}

```