**Phần 1: Giới thiệu Căn bản MVC và tập làm project nho nhỏ**

Chào các bạn SV đại học mở. Sau đây mình giới thiệu 1 chút kiến thức về MVC kết hợp với LinQToSQL hoặc Entity FrameWork 4.0. Tại năng hạn hẹp, kiến thức lơ tơ mơ, các bạn xem và góp ý, trong lúc viết bài này có chút bối rối, có điều chi sơ xót mong các bạn bỏ qua.

1. **Kiến trúc MVC là gì ? ^^**

MVC viết tắt của các chữ cái đầu của Models, Views, Controllers. MVC chia giao diện UI thành 3 phần tương ứng: đầu vào của controller là các điều khiển thông qua HTTP request, model chứa các miền logic, view là những thứ được sinh ra trả về cho trình duyệt.

**Model**: nhiệm vụ cung cấp dữ liệu cho cơ sở dữ liệu và lưu dữ liệu vào các kho chứa dữ liệu. Tất cả các nghiệp vụ logic được thực thi ở Model. Dữ liệu vào từ người dùng sẽ thông qua View được kiểm tra ở Model trước khi lưu vào cơ sở dữ liệu.Việc truy xuất, xác nhận, và lưu dữ liệu là một phần của Model.

**View**: hiển thị các thông tin cho người dùng của ứng dụng và được giao nhiệm vụ cho việc nhận các dữ liệu vào từ người dùng, gửi đi các yêu cầu đến bộ điều khiển, sau đó là nhận lại các phản hồi từ bộ điều khiển và hiển kết quả cho người dùng. Các trang HTML, các thư viện thể và các file nguồn là một phần của thành phần View.

**Controller**: tầng trung gian giữa Model và View. Controller được giao nhiệm vụ nhận các yêu cầu từ phía máy khách. Một yêu cầu được nhận từ máy khách được thực hiện bởi một chức năng logic thích hợp từ thành phần Model và sau đó sinh ra các kết quả cho người dùng và được thành phần View hiển thị. Action, ActionForm và struts-config.xml là các phần của Controller



1. *Mô hình MVC cơ bản*

Lợi ích của việc dùng phương pháp MVC là sự phân đoạn rõ ràng giữa models, views, controllers bên trong ứng dụng. Cấu trúc sạch sẽ giúp cho việc kiểm tra lỗi ứng dụng trở nên dễ dàng hơn.

**Một vài đặc tính trong ASP.NET MVC :**

* Có khả năng mở rộng, mọi thứ trong MVC được thiết kế cho phép dễ dàng thay thế/ tùy biến( ví dụ: có thể lựa chọn sử dụng engine view riêng, routing policy, parameter serialization, v.v…).
* Bao gồm một ánh xạ URL mạnh mẽ cho phép xây dựng ứng dụng với những URL sạch, các URL không cần cs phần mở rộng ( vídụ: có thể ánh xạ địa chỉ /Products/Edit/4 để thực hiện hành động “Edit” của lớp điềukhiển Product Controllers hoặc ánh xạ địa chỉ /Blog/Some Topic để thực hiện hành động “Display Topic” của lớp điều khiển Blog Engine Controller ) .
* Hỗ trợ những file ASP.NET như .ASPX .ASCX và .Master, đánh dấu các tập tin này như một “view template” ( có thể dễ dàng dùng các tính năng của ASP.NET như lồng các trang Master, <%= %> snippets, mô tả server controls, template, data-binding, localization, v.v… ). Tuy nhiên sẽ không còn postback và interactive back server và thay vào đó là interactive end-user tới một Controller class ( không còn viewstate, page lifecycle ) .
* Hỗ trợ đầy đủ các tính năng bảo mật của ASP.NET như forms/ windows authenticate, URL authorization, membership/ roles, output và data caching, session/profile state, configuration system, provider architecture v.v…

**Sự khác biệt với WebForm**

* WebForm sử dụng ViewState để quản lý, các trang ASP.NET đều có lifecycle, postback và dùng các web controls, các events để thực hiện các hành động cho UI khi có sự tương tác với người dùng nên hầu hết ASP.NET WebForm xử lý chậm.
* MVC Framework chia ra thành 3 phần: Models, Views, Controllers. Mọi tương tác của người dùng với Views sẽ được thực hiện hành động trong Controllers, không còn post back, không còn life cycle không còn events.
* Việc kiểm tra ( test ), gỡ lỗi ( debug ) với ASP.NET đều phải chạy tất cả các tiến trình của ASP.NET và mọi sự thay đổi ID của bất kỳ controls nào cũng ảnh hưởng đến ứng dụng. Đối với ASP.NET MVC Framework thì việc có thể sử dụng các unit test có thể sử dụng các unit test có thể thẩm định rất dễ dàng các Controller thực hiện như thế nào.

|  |  |  |
| --- | --- | --- |
| **Tính năng** | **ASP.NET 2.0** | **ASP.NET MVC** |
| Kiến trúc chương trình | Kiến trúc mô hình WebForm 🡪Business 🡪 Database | Kiến trúc sử dụng việc phân chia chương trình thành Controllers, Models, Views |
| Cú pháp chương trình | Sử dụng cú pháp của web form, tất các sự kiện và controls do server quản lý | Các sự kiện được điều khiển bởi controllers, các controls không do server do server quảnlý |
| Truy cập dữ liệu | Sử dụng hầu hết các công nghệ truy cập dữ liệu trong ứng dụng | Phần lớn dùng LINQ to SQL class để tạo mô hình truy cập đối tượng |
| Debug | Debug chương trình phải thực hiện tất cả bao gồm các lớp truy cập dữ liệu, sự hiển thị, điều khiển các controls | Debug có thể sử dụng các unit test kiểm tra các phương thức trong controller |
| Tốc độ phân tải | Tốc độ phân tải chậm khi trong trang có quá nhiều các controls vì ViewState quá lớn | Phân tải nhanh hơn do không phải quản lý ViewState để quản lý các control trongtrang |
| Tương tác với javascript | Tương tác với java script khó khăn vì các controls được điều khiển bởi server | Tương tácvới javascript dễ dàng vì các đối tượng không do server quản lý điều khiển không khó |
| URL Address | Cấu trúc địa chỉ URL có dạng <filename>.aspx?&<các tham số> | Cấu trúc địa chỉ rành mạch theo dạng Controllers/Action/Id |

#### 

1. **Thực hành MVC:**

#### Môi trường thực hiện :

#### Visual 2010 (nếu Visual 2008 thì các bạn phải nâng lên SP1, cài gói MVC1 vào nhé )

#### Sql Server 2008 ( server 2005 cũng được )

#### Thực hiện:

#### Mô tả bài toán: chúng ta sẽ thục hiện project MVC có chức năng Xem, them, xóa, sửa sản phẩm, loại sản phẩm ( bài toán kinh điển )

#### Tạo project như sau:

#### 

#### Nhấn Ok 🡪 trong hộp thoại Create Unit Test Project chọn No, do not create ….

#### Khi đó chúng ta sẽ có 1 project như sau:

#### 

#### Chúng ta chạy thử nhé , Clrt + F5 nhé (chạy ko debug) sẽ xuất hiện giao diện Web MVC mặc định như sau:

#### 

#### F5 mà ko lên được vậy, thì khỏi làm tiếp nhé ( giởn thôi )

#### Xây dựng Database :

#### Cách 1: tạo trên visual lun ( demo thôi, khuyến cáo ko dung cách này )

#### Từ App\_Data click phải chuột chọn Add 🡪 New Item 🡪 SQL Server Database đặt tên Database.mdf

#### 

#### Tạo tiếp 2 bảng giống như trong hình:

#### Bảng *Sản phẩm:*

#### 

#### Bản Loại sản phẩm:

#### 

#### Xây dựng model:

#### Phần model này, mọi thao tác trên web, tương tác tới dử liệu bên dưới, mình để ở phần Model nhé. Vi dụ như : viết LinQ dể lấy dử liệu, hoặc them, xóa sửa gì đó phía dưới có lien quan tới dữ liệu.

#### Giờ chúng ta tạo file LinQToSQL ( LinQtoSQL là gì các bạn từ google nhé ), Mọi thao tác dữ liệu, mình sẽ thong quan file này, nó giao tiếp với database đấy.

#### Trong Solution Explorer click phải chuột vảo Models 🡪 Add 🡪 New Item 🡪LINQ to SQL Classes

#### 

#### Từ Server Explorer 🡪 Kéo thả 2 tables LoaiSanPham, SanPham vào file DataClasses.dbml

#### 

#### Trong file DataClasses.dbml click phải chuột chọn Add 🡪 Association để tạo liên kết giữa 2 table

#### 

#### Kết quả như hình bên dưới:

#### 

#### Giờ chúng ta tạo 1 class tên SanPhamControll ở phần model, để thực hiện các thao tác với dử liệu:

#### 

#### Trong class này sẽ cung cấp các phương thức sau:

#### Lấy sản phẩm

#### Tạo sản phẩm

#### Cập nhật sản phẩm

#### Xóa sản phẩm.

#### Lấy sản phẩm, loại sản phẩm lên để hiển thị

#### 

#### Chú ý:

from p in data.SanPhams where p.LoaiSanPham == idloaisanpham select p ( viết bằng LinQ )

* Select \* from sanpham where loaisanpham = Id ( lệnh T-SQL )

#### Như chúng ta thấy, LinQ trên visual sẽ giúp chúng ta sinh tạo ra 1 cái T-sql xuống database để lấy dữ liệu lên. ( các bạn tìm hiểu thêm về LinQ nhé )

#### Phương thức tạo sản phẩm:

#### 

#### Phương thức cập nhật sản phẩm, loại sản phẩm

#### 

#### Phương thức xóa sản phẩm, loại sản phẩm

#### 

#### Chú ý:

#### Mỗi bảng trong database , LinQTosql sẽ giúp chúng ta Map chúng thành 1 đối tượng cùng tên, các trường trong bảng đó sẽ thành 1 thuộc tính của đối tượng.

#### Xây dựng controllers

#### Ở phần controllers này, các bạn sẽ thực hiện các thao tác nghiệp vụ ở đây. ( Bên mô hình 3 layer thì BLL nó phụ trách việc này ).Chúng ta sẽ sử dụng các phương thức o dưới phần Model cung cấp, để thao tác.

#### Tạo một Controller mới :

#### Trong folder Controllers 🡪 phải chuột chọn Add 🡪New Item 🡪MVC Controller Class

#### 

**Ở đây mình giới thiệu các bạn 1 cái rất quan trong trong phần controllers:** ActionResult. Mỗi ActionResult có thể trả về 1 view ( coi phần lý thuyết view la gì nhé ), hoặc có thể là nó thực hiện 1 thao tác xử lý gì đó, do view yêu cầu ( ví dụ, nhấn nút Add trên view, nó sẻ gọi xuống ActionResult xử lý việc tạo đó).

#### Trong class SanphamController này chúng ta có những phương thức như sau:

#### Tạo Action Mặc định, và action lấy thông tin loại sản phẩm

#### 

#### Tạo loại sản phẩm

#### 

#### Các bạn chú ý mình đã sử dụng Phương thức CreateLoaiSanPham đã tạo ở phần Model.

#### Sửa và xóa loại sản phẩm

#### 

#### Tương tự , chúng ta tạo them các Action xử lý cho Sản phẩm

#### 

#### 

#### 

#### 

#### Tạo View ( Giao diện):

Trong controller HomeController.cs, cả hai phương thức Index() và About() đều trả về một view. Một view chứa các thẻ HTML và được trả về cho browser. Trong mô hình MVC thì một view tương ứng với một trang .aspx trong WebForm.

View phải được tạo đúng vị trí đường dẫn. Ví dụ với controller HomeController.cs thì hành động HomeController.Index() trả về một view nằm ở vị trí đường dẫn \Views\Home\Index và HomeController.About() sẽ trả về một view nằm ở vị trí đường dẫn \Views\Home\About

Tạo view cho controller SanPhamController.cs, trong SanPhamController.cs có 4 phương thức public là Index(), DanhMucLoaiSanPham(), DanhSachSanPham() và ChiTietSanPham() nên trong Views phải tạo 4 view tương ứng là Index.aspx, DanhMucLoaiSanPham.aspx, DanhSachSanPham.aspx, ChiTietSanPham.aspx.Có 2 cách tạo view:

* Từ folder Views click phải chuột chọn Add  New Folder nhập tên folder mới tạo là SanPham. Từ folder SanPham click phải chuột chọn Add  View…
* Trong class SanPhamController các bạn chọn Action muốn trả về view, các bạn nhấn chuột phải ngay tên action đó 🡪 add view…

#### Trong phần này chúng ta có 7 view nhé:

#### Tạo loại sản phẩm (CreateLoaiSP)

#### Tạo sản phẩm (Create)

#### Danh sách hiển thị loại sản phẩm (DSLoaiSanPham)

#### Danh sách sản phẩm hiện thi theo loại sản phẩm (DSSanPham)

#### Hiển thi thông tin chi tiết sản phẩm (Details)

#### Sửa thông tin sản phẩm (Edit)

#### Sửa thông tin loại sản phẩm (EditLoaiSP)

#### Chuôt phải bên class SanPhamController , chọn tên action như mình đã liệt kê, bấm add view. Sau đó tiến hành code.

#### View Tạo sản phẩm *(Phần này không được lười code, copy là ko dc nha, tự gỏ lại đi )*

<%@ Page Title="" Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage<MvcApplication.Models.SanPham>" %>

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

Create

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

<h2>Create</h2>

<% using (Html.BeginForm()) {%>

<%: Html.ValidationSummary(true) %>

<fieldset>

<legend>Fields</legend>

<div class="editor-label">

<%: Html.LabelFor(model => model.Id) %>

</div>

<div class="editor-field">

<%: Html.TextBoxFor(model => model.Id) %>

<%: Html.ValidationMessageFor(model => model.Id) %>

</div>

<div class="editor-label">

<%: Html.LabelFor(model => model.TenSanPham) %>

</div>

<div class="editor-field">

<%: Html.TextBoxFor(model => model.TenSanPham) %>

<%: Html.ValidationMessageFor(model => model.TenSanPham) %>

</div>

<div class="editor-label">

<%: Html.LabelFor(model => model.DonGia) %>

</div>

<div class="editor-field">

<%: Html.TextBoxFor(model => model.DonGia) %>

<%: Html.ValidationMessageFor(model => model.DonGia) %>

</div>

<div class="editor-label">

<%: Html.LabelFor(model => model.SoLuong) %>

</div>

<div class="editor-field">

<%: Html.TextBoxFor(model => model.SoLuong) %>

<%: Html.ValidationMessageFor(model => model.SoLuong) %>

</div>

<div class="editor-label">

<%: Html.LabelFor(model => model.LoaiSanPham) %>

</div>

<div class="editor-field">

<%: Html.TextBoxFor(model => model.LoaiSanPham) %>

<%: Html.ValidationMessageFor(model => model.LoaiSanPham) %>

</div>

<p>

<input type="submit" value="Create" />

</p>

</fieldset>

<% } %>

<div>

<%: Html.ActionLink("Back to List", "Index") %>

</div>

</asp:Content>

* Tạo view tạo loại sản phẩm

<%@ Page Title="" Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage<MvcApplication.Models.LoaiSanPham>" %>

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

CreateLoaiSP

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

<h2>CreateLoaiSP</h2>

<% using (Html.BeginForm()) {%>

<%: Html.ValidationSummary(true) %>

<fieldset>

<legend>Fields</legend>

<div class="editor-label">

<%: Html.LabelFor(model => model.Id) %>

</div>

<div class="editor-field">

<%: Html.TextBoxFor(model => model.Id) %>

<%: Html.ValidationMessageFor(model => model.Id) %>

</div>

<div class="editor-label">

<%: Html.LabelFor(model => model.TenLoaiSanPham) %>

</div>

<div class="editor-field">

<%: Html.TextBoxFor(model => model.TenLoaiSanPham) %>

<%: Html.ValidationMessageFor(model => model.TenLoaiSanPham) %>

</div>

<p>

<input type="submit" value="Create" />

</p>

</fieldset>

<% } %>

<div><%: Html.ActionLink("Back to List", "Index") %></div></asp:Content

* Tạo view hiển thị loại sản phẩm

<%@ Page Title="" Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage<IEnumerable<MvcApplication.Models.LoaiSanPham>>" %>

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

DSLoaiSanPham

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

<h2>DSLoaiSanPham</h2>

<table>

<tr>

<th></th>

<th>

Id

</th>

<th>

TenLoaiSanPham

</th>

</tr>

<% foreach (var item in Model) { %>

<tr>

<td>

<%: Html.ActionLink("Edit", "EditLoaiSP", new { id = item.Id })%> |

<%: Html.ActionLink("Details", "Details", new { id=item.Id })%> |

<%: Html.ActionLink("Delete", "DeleteLoaiSP", new { id = item.Id })%> |

<%: Html.ActionLink("Xem Sản Phẩm", "DSSanPham", new { id = item.Id })%>

</td>

<td>

<%: item.Id %>

</td>

<td>

<%: item.TenLoaiSanPham %>

</td>

</tr>

<% } %> </table><p><%: Html.ActionLink("Create New", "CreateLoaiSP")%> </p>

</asp:Content>

* Tạo view hiển thị danh sách sản phẩm

<%@ Page Title="" Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage<IEnumerable<MvcApplication.Models.SanPham>>" %>

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

DSSanPham

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

<h2>DSSanPham</h2>

<table>

<tr>

<th></th>

<th>

Id

</th>

<th>

TenSanPham

</th>

<th>

DonGia

</th>

<th>

SoLuong

</th>

<th>

LoaiSanPham

</th>

</tr>

<% foreach (var item in Model) { %>

<tr>

<td>

<%: Html.ActionLink("Edit", "Edit", new { id=item.Id }) %> |

<%: Html.ActionLink("Details", "Details", new { id=item.Id })%> |

<%: Html.ActionLink("Delete", "Delete", new { id=item.Id })%>

</td>

<td>

<%: item.Id %>

</td>

<td>

<%: item.TenSanPham %>

</td>

<td>

<%: String.Format("{0:F}", item.DonGia) %>

</td>

<td>

<%: item.SoLuong %>

</td>

<td>

<%: item.LoaiSanPham %>

</td>

</tr>

<% } %>

</table>

<p>

<%: Html.ActionLink("Create New", "Create") %>

</p>

</asp:Content>

* Tạo view Chinh sưa loại sản phẩm

<%@ Page Title="" Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage<MvcApplication.Models.LoaiSanPham>" %>

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

EditLoaiSP

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

<h2>EditLoaiSP</h2>

<% using (Html.BeginForm()) {%>

<%: Html.ValidationSummary(true) %>

<fieldset>

<legend>Fields</legend>

<div class="editor-label">

<%: Html.LabelFor(model => model.Id) %>

</div>

<div class="editor-field">

<%: Html.TextBoxFor(model => model.Id) %>

<%: Html.ValidationMessageFor(model => model.Id) %>

</div>

<div class="editor-label">

<%: Html.LabelFor(model => model.TenLoaiSanPham) %>

</div>

<div class="editor-field">

<%: Html.TextBoxFor(model => model.TenLoaiSanPham) %>

<%: Html.ValidationMessageFor(model => model.TenLoaiSanPham) %>

</div>

<p>

<input type="submit" value="Save" />

</p>

</fieldset>

<% } %>

<div>

<%: Html.ActionLink("Back to List", "DSLoaiSanPham")%></div></asp:Content>

* Tạo view chỉnh sửa sản phẩm

<%@ Page Title="" Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage<MvcApplication.Models.SanPham>" %>

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

Edit

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

<h2>Edit</h2>

<% using (Html.BeginForm()) {%>

<%: Html.ValidationSummary(true) %>

<fieldset>

<legend>Fields</legend>

<div class="editor-label">

<%: Html.LabelFor(model => model.Id) %>

</div>

<div class="editor-field">

<%: Html.TextBoxFor(model => model.Id) %>

<%: Html.ValidationMessageFor(model => model.Id) %>

</div>

<div class="editor-label">

<%: Html.LabelFor(model => model.TenSanPham) %>

</div>

<div class="editor-field">

<%: Html.TextBoxFor(model => model.TenSanPham) %>

<%: Html.ValidationMessageFor(model => model.TenSanPham) %>

</div>

<div class="editor-label">

<%: Html.LabelFor(model => model.DonGia) %>

</div>

<div class="editor-field">

<%: Html.TextBoxFor(model => model.DonGia, String.Format("{0:F}", Model.DonGia)) %>

<%: Html.ValidationMessageFor(model => model.DonGia) %>

</div>

<div class="editor-label">

<%: Html.LabelFor(model => model.SoLuong) %>

</div>

<div class="editor-field">

<%: Html.TextBoxFor(model => model.SoLuong) %>

<%: Html.ValidationMessageFor(model => model.SoLuong) %>

</div>

<div class="editor-label">

<%: Html.LabelFor(model => model.LoaiSanPham) %>

</div>

<div class="editor-field">

<%: Html.TextBoxFor(model => model.LoaiSanPham) %>

<%: Html.ValidationMessageFor(model => model.LoaiSanPham) %>

</div>

<p>

<input type="submit" value="Save" />

</p>

</fieldset>

<% } %>

<div>

<%: Html.ActionLink("Back to List", "DSLoaiSanPham")%>

</div>

</asp:Content>

* Tạo view chi tiết sản phẩm

<%@ Page Title="" Language="C#" MasterPageFile="~/Views/Shared/Site.Master" Inherits="System.Web.Mvc.ViewPage<MvcApplication.Models.SanPham>" %>

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

Details

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

<h2>Details</h2>

<fieldset>

<legend>Fields</legend>

<div class="display-label">Id</div>

<div class="display-field"><%: Model.Id %></div>

<div class="display-label">TenSanPham</div>

<div class="display-field"><%: Model.TenSanPham %></div>

<div class="display-label">DonGia</div>

<div class="display-field"><%: String.Format("{0:F}", Model.DonGia) %></div>

<div class="display-label">SoLuong</div>

<div class="display-field"><%: Model.SoLuong %></div>

<div class="display-label">LoaiSanPham</div>

<div class="display-field"><%: Model.LoaiSanPham %></div>

</fieldset>

<p>

<%: Html.ActionLink("Edit", "Edit", new { id=Model.Id }) %> |

<%: Html.ActionLink("Back to List", "Index") %>

</p>

</asp:Content>

#### Tạo Site.Master

<%@ Master Language="C#" Inherits="System.Web.Mvc.ViewMasterPage" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

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

<head runat="server">

<title><asp:ContentPlaceHolder ID="TitleContent" runat="server" /></title>

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

</head>

<body>

<div class="page">

<div id="header">

<div id="title">

<h1>My MVC Application</h1>

</div>

<div id="logindisplay">

<% Html.RenderPartial("LogOnUserControl"); %>

</div>

<div id="menucontainer">

<ul id="menu">

<li><%: Html.ActionLink("Home", "Index", "Home")%></li>

<li><%: Html.ActionLink("SanPham", "DSLoaiSanPham", "SanPham")%></li>

<li><%: Html.ActionLink("Tìm Kiếm", "Index", "TimKiem")%></li>

<li><%: Html.ActionLink("About", "About", "Home")%></li>

</ul>

</div>

</div>

<div id="main">

<asp:ContentPlaceHolder ID="MainContent" runat="server" />

<div id="footer">

</div>

</div>

</div>

</body>

</html>

#### Kết quả thực hiện được đây:

#### Giao diện view chính

#### 

#### Giao diện view danh sách loại sản phẩm

#### 

#### Giạo diện view Danh sách sản phẩm

#### 

#### Giao diện view tạo loại sản phẩm

#### 

#### Giao diện view tạo moi sản phẩm

#### 

#### Giao diện view chi tiết thong tin sản phẩm

#### 

#### Giao diện view Sửa thong tin san phẩm

#### 

#### Đây là MVC kết hợp với LinQToSQL, mơ mức thuần .Net, nên giao diện chưa mượt mà.

#### Nếu các bạn kết hợp Jquery, Ajax vào thì nó sẽ rất đáng sợ như hình dưới đây:

#### C:\Users\Home\Desktop\capnhathangmuc.png

#### C:\Users\Home\Desktop\kh2.png

#### 

#### 

#### Kết bài: cảm ơn các bạn đã quan tâm. Có thac mắc gì các bạn cứ hỏi nhé, có dịp rảnh,minh sẻ trình bài tiếp phần kết hợp với jquery, ajax,…. Linh tinh vài cái nửa.