

## Lab 02:

## XÂY DỰNG ỨNG DỤNG WEB THEO MÔ HÌNH MVC VỚI CƠ SỞ DỮ LIỆU LINQ TO SQL

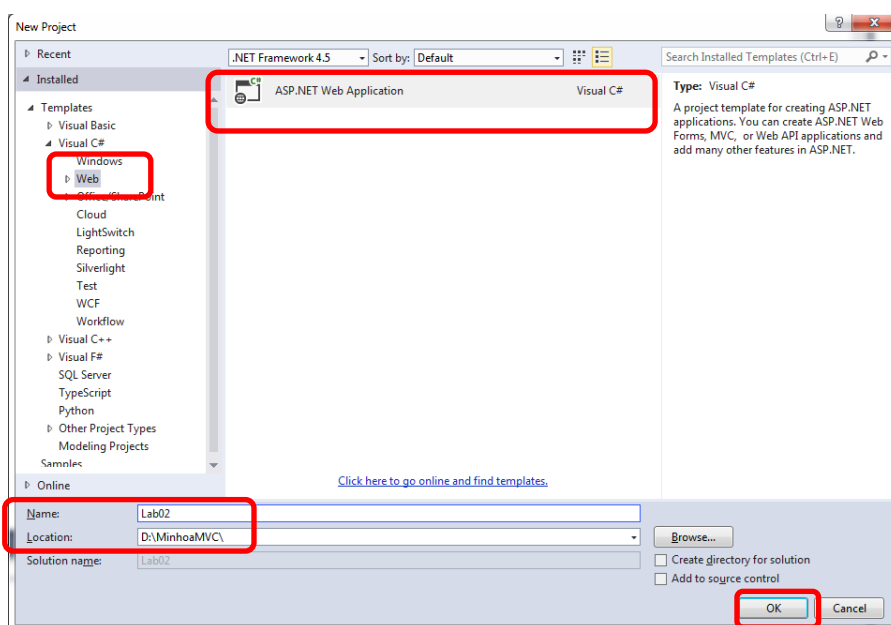
### A. MỤC TIÊU:

- ✓ Xây dựng ứng dụng MVC với ASP.Net và SQL server
- ✓ Ứng dụng minh họa là xây dựng các trang Xem, thêm, xóa, sửa dữ liệu cho các bảng trong ứng dụng “ASP.Net MVC Application News”.
- ✓ Để xây dựng được ứng dụng chúng ta cần các công cụ phần mềm hỗ trợ sau :Visual studio 2013. (Nếu 2010/2012 thì cần cài thêm bộ MVC 5), SQL server 2005 /2008/2012

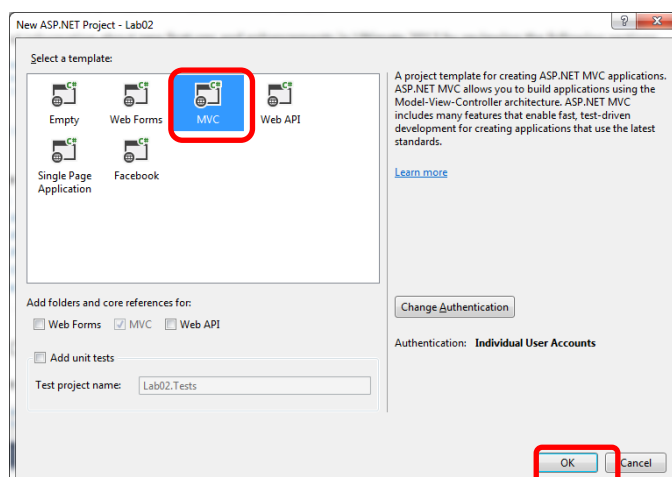
### B. NỘI DUNG:

#### Bài tập 1. Tạo mới project

- ✓ Mở Visual Studio 2013 chọn **File** → **New** → **Project** để tạo 1 ứng dụng mới .
- ✓ Chọn loại Project Visual C# → Web → **ASP.Net web Application**  
→ Tên project là Lab02

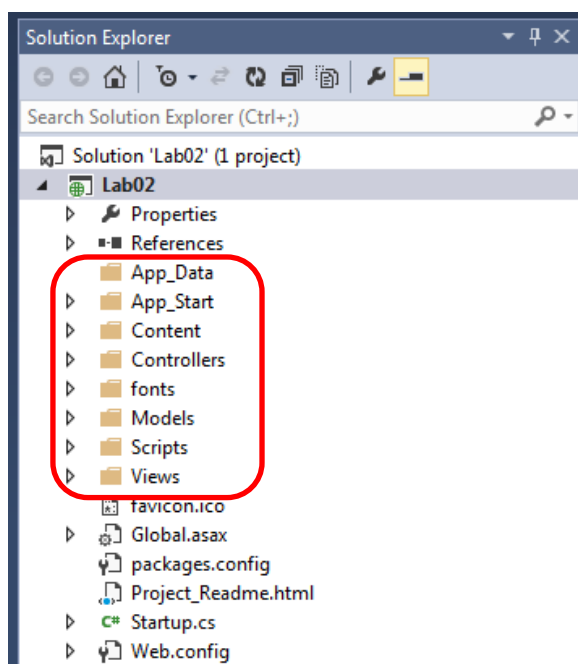


Hình 1: Màn hình tạo mới Project Website ASP.Net



Hình 2: Màn hình chọn loại Project Website là MVC

- ✓ Trong cửa sổ Solution Explorer ta thấy có 8 thư mục :
  - App\_Data: Nên chứa các file Database
  - App\_Start: Nơi chứa các file cấu hình điều hướng
  - Content: Chứa những file css
  - Controllers: Chứa những file điều khiển
  - Fonts: Chứa các fonts cho ứng dụng
  - Models: chứa những file Models ánh xạ dữ liệu .
  - Scripts: Chứa các file JS
  - Views: Chứa những file hiển thị aspx



Hình 3: Màn hình Solution Explorer

## Bài tập 2. Tạo cơ sở dữ liệu :

- ✓ Trong SQLServer tạo CSDL Tintuc gồm 2 bảng và lưu file lệnh vào thư mục Database\CSDLTintuc.sql:
  - Theloaitin(IDLoai int, Tentheloi nvarchar(100))
  - Tintuc(IdTin int, IDLoai int, Tieudetin nvarchar(100), Noidungtin nText )

**Chú ý:** Với cột IDLoai và IdTin chúng ta thiết lập là số nguyên tự động.

```
--Tao Database
Create Database Tintuc
--Tao cac Table
use Tintuc
Create Table Theloaitin
(
    IDLoai    int identity primary key,
    Tentheloi nvarchar(100)
)
Go
Create Table Tintuc
(
    IdTin    int identity Primary key,
    IDLoai   int references Theloaitin(IDLoai),
    Tieudetin nvarchar(100),
    Noidungtin nText
)
```

```

Noidungtin nText
)
Go
--Nhap lieu
Insert into Theloaitin(Tentheloai) values(N'Thể thao')
Insert into Theloaitin(Tentheloai) values(N'Kinh tế')
Insert into Theloaitin(Tentheloai) values(N'Thể giới')

Insert into Tintuc(IDLoai, Tieudetin, Noidungtin) values(2,N'Khủng hoảng
kinh tế trong năm 2012',N'Tình hình khủng hoảng kinh tế năm 2012 được các
chuyên gia kinh tế đánh giá . . .')
Insert into Tintuc(IDLoai, Tieudetin, Noidungtin) values(1,N'Tranh chấp
trên biển đông',N'Trên các diễn đàn quân sự đang nóng dẫn về tình hình biển
đông . . .')

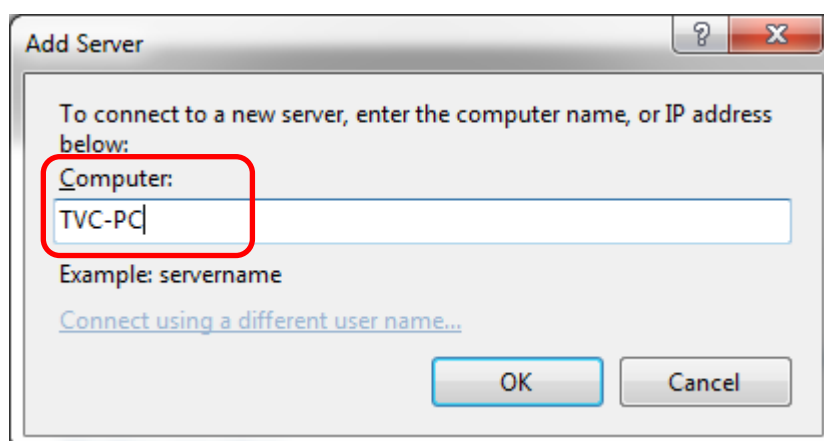
```

IDLoai	Tentheloai	IdTin	IDLoai	Tieudetin	Noidungtin
1	Thể thao	1	2	Khủng hoảng kinh tế trong năm 2012	Tình hình khủng hoảng kinh tế năm 2012 được...
2	Chính trị	2	1	Tranh chấp trên biển đông	ân sự đang nóng dẫn về tình hình biển đông . . .
3	Thể giới				

Hình 4: CSDL Tin tức trong SQLServer

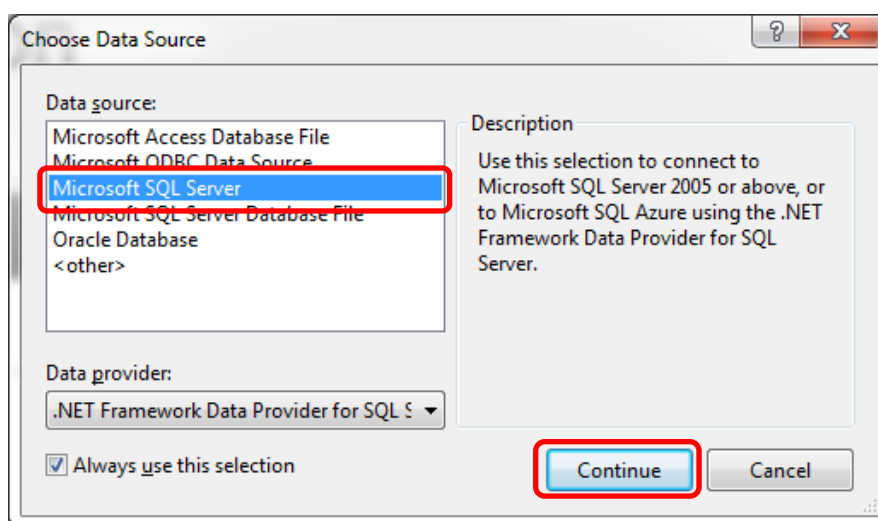
**Bài tập 3: Kết nối CSDL với ứng dụng**

- ✓ Chọn Menu Tools ➔ Connect to Server => Nhập Server Name



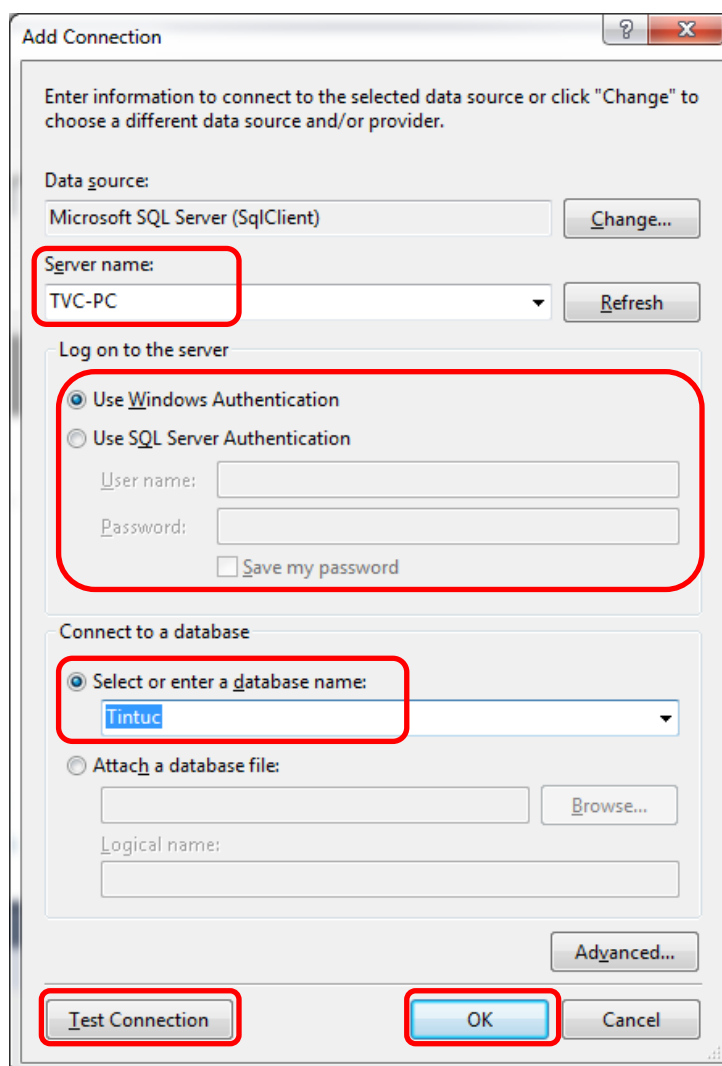
Hình 5: Màn hình Add Server

- ✓ Chọn Menu Tools => Connect to Database => Chọn Microsoft SQLServer ➔ Continue



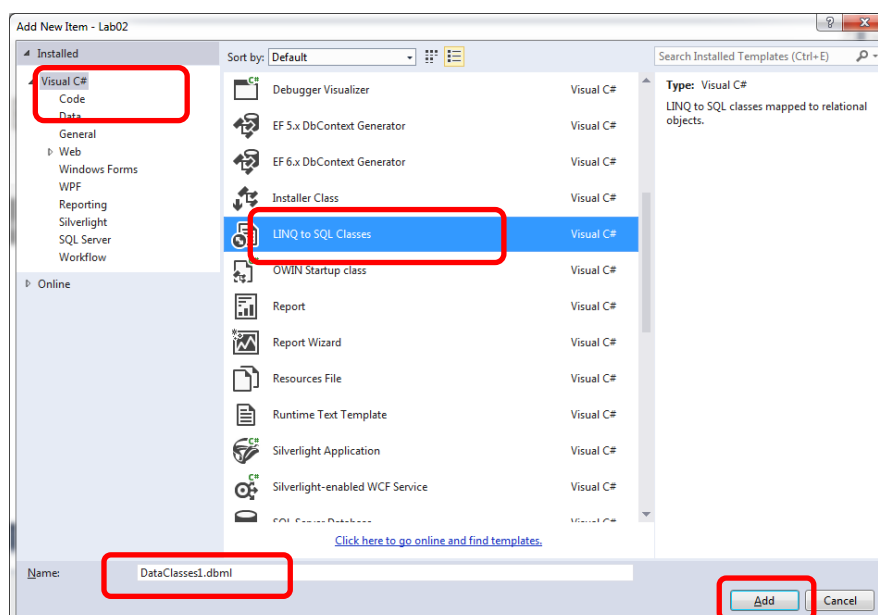
Hình 6: Màn hình Choose Data Source

- ✓ Nhập Servername và chọn Database cần kết nối (Tintuc) (username, password nếu có). Chọn Test Connection nếu cần kiểm tra.



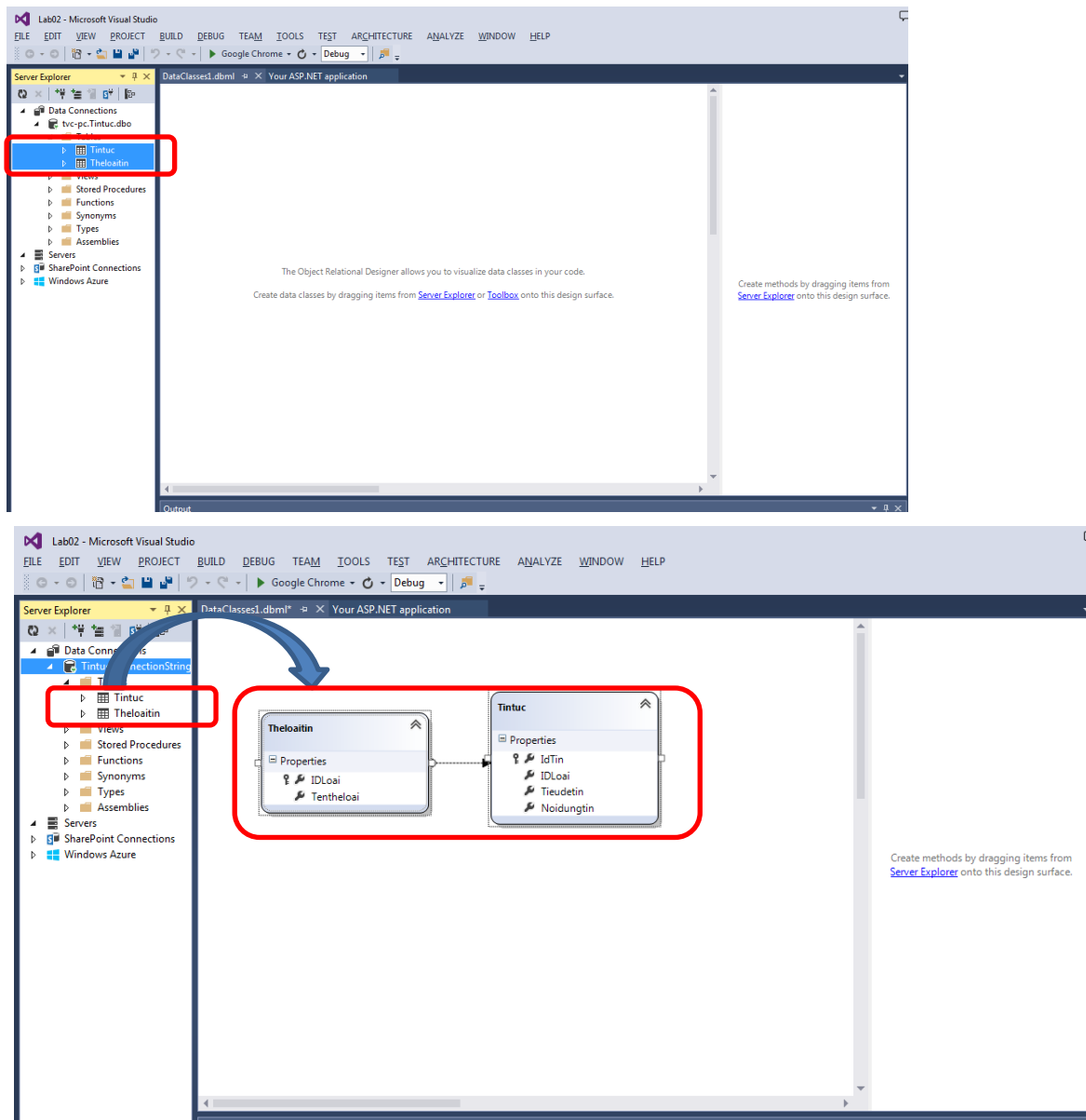
Hình 7: Màn hình Add Connection

- ✓ Click phải chuột vào folder Models => add=> chọn Class



Hình 8: Màn hình Tạo mới Class Model

- ✓ Chọn **LINQ to SQL classes** ( Vì chúng ta truy xuất dữ liệu bằng LINQ) . Đặt tên cho file Class là **DataClasses1.dbml** .
- ✓ Trên thanh Server Explorer kéo thả các bảng của Database sang trang Dataclasses.dbml

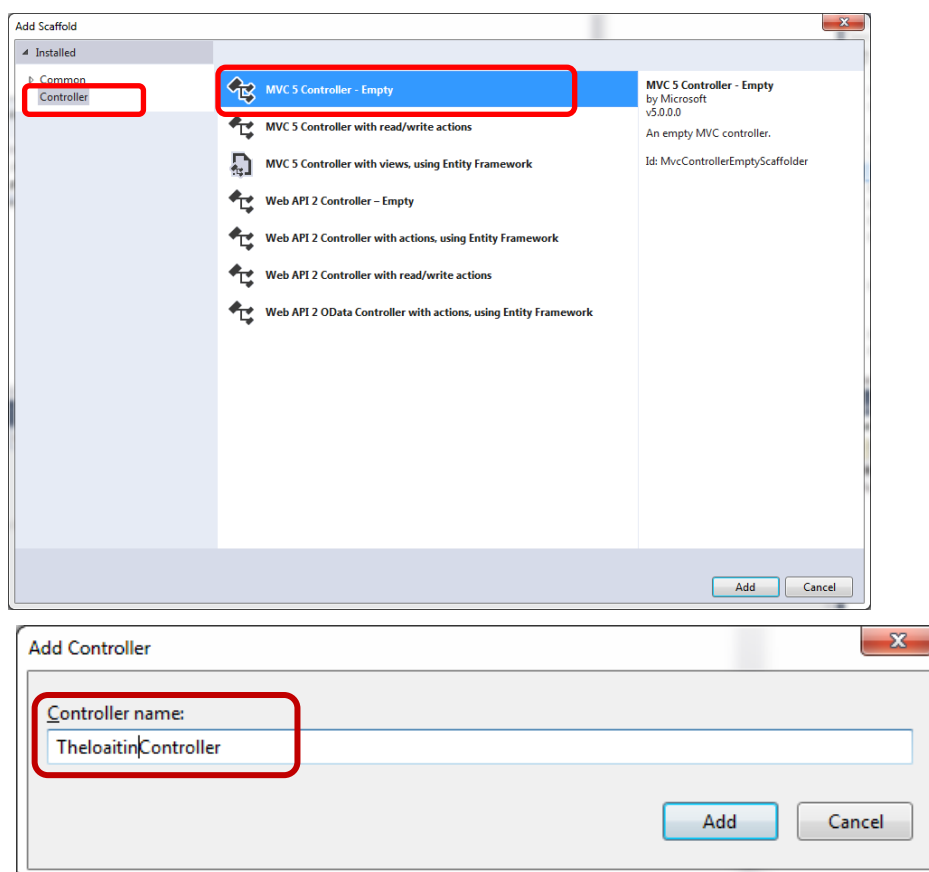


Hình 8: Màn hình Tạo Database cho Dataclasses.dbml

#### Bài tập 4: Xử lý Controller

##### Xử lý bảng Thể loại tin:

- Tại Folder controller click chuột phải chọn Add ➔ Controller



Hình 9: Màn hình tạo mới Controller

- Đặt tên cho Controller (chú ý tên của lớp xử lý phải theo sau bằng từ Controller) đặt tên lớp xử lý bằng Theloaitin là TheloaitinController.cs. (Nếu muốn trình biên dịch tự tạo các hàm Create, Update, Delete thì bạn chọn mục Add action methods bên dưới)

✓ **Bổ sung Code vào file TheloaitinControllers.cs như sau :**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Lab02.Models;

namespace Lab02.Controllers
{
    public class TheloaitinController : Controller
    {
        //
        // GET: /TheLoaiTinTuc/
        DataClasses1DataContext data = new DataClasses1DataContext();
        public ActionResult Index()
        {
            var All_Loaitin = from tt in data.TheLoaitins select tt;
            return View(All_Loaitin);
        }
    }
}
```

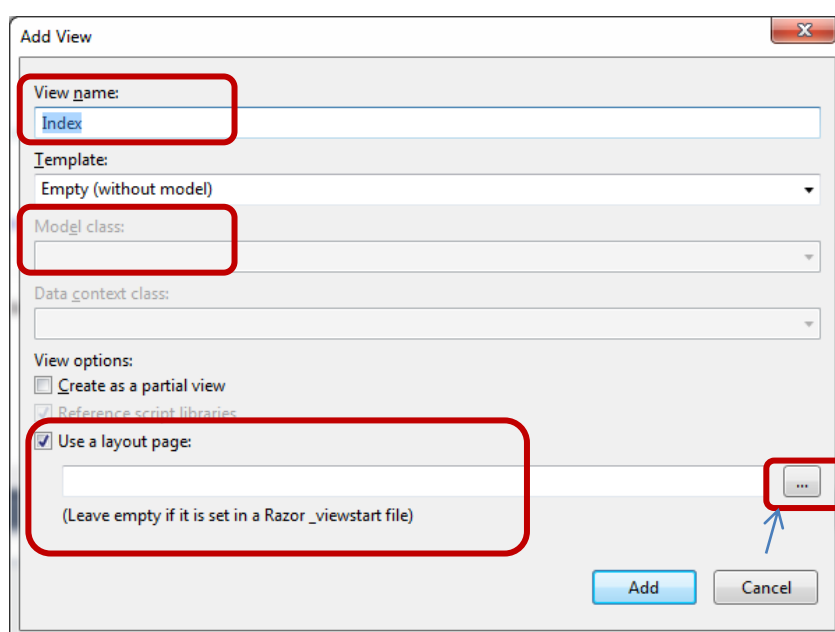
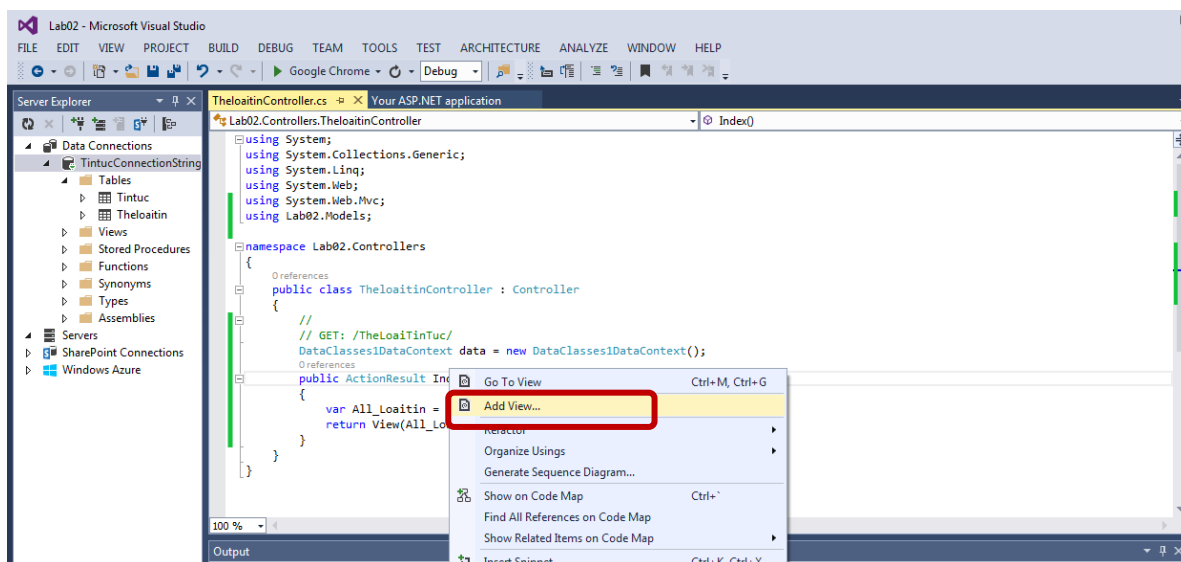
Tên Model

Tên DataTable trong Model

Truy vấn dữ liệu với LinQ

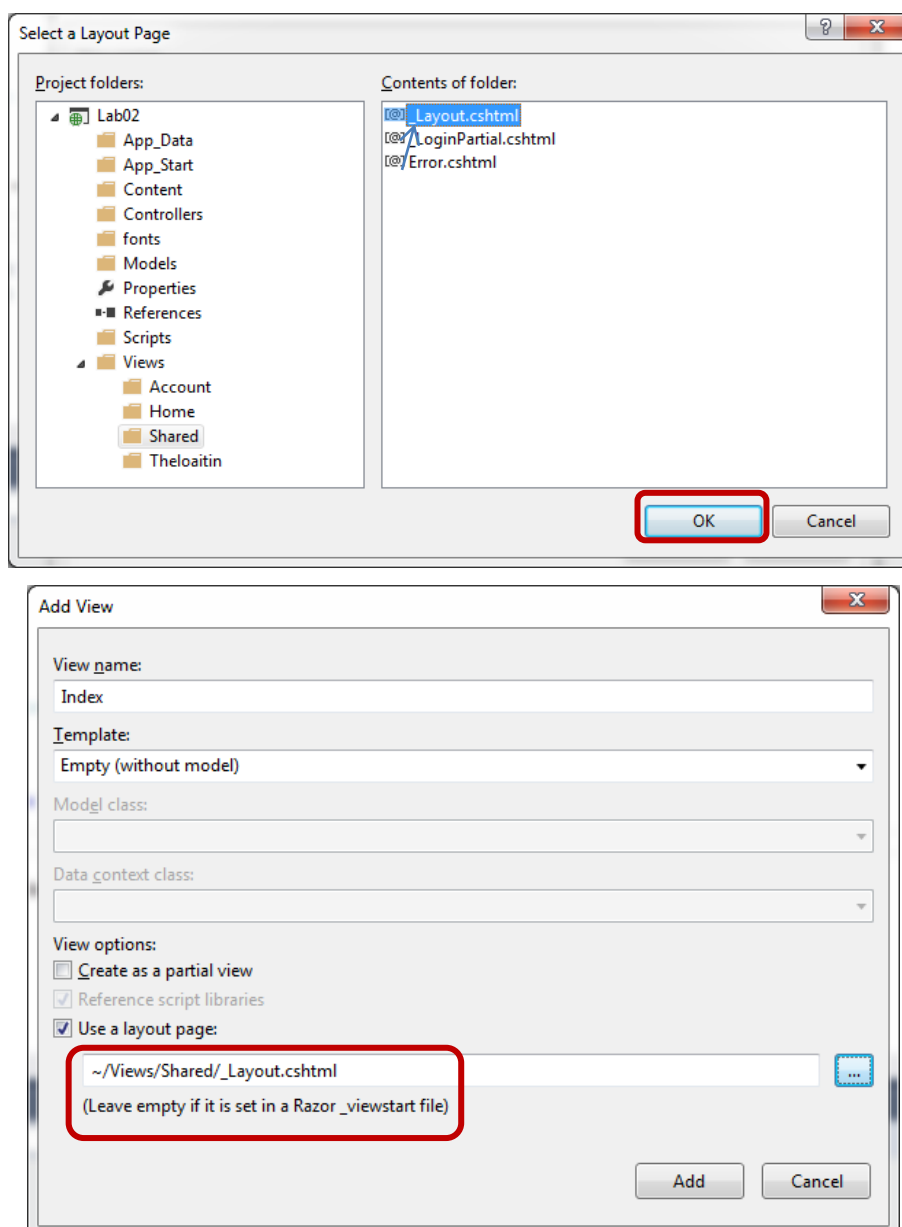
**Bài tập 5: Xử lý View để hiện thị nội dung lên Web.**

- ✓ Ta có file TheloaitinControllers.cs trên thư mục Controlesrs thì tương ứng dưới thư mục View ta cũng phải có thư mục tên Theloaitin để chứa các trang .cshtml cho các hàm thuộc file TheloaitinControllers.cs.
- ✓ Click phải chuột vào View => add => New Folder ➔ **Theloaitin**
- ✓ Trong File TheloaitinControllers.cs ở trên ta có hàm:
  - Index : Hiện thị danh sách
- ✓ Click phải vào tên hàm (Ví dụ hàm Index ) => add view...



Hình 10: Màn hình tạo mới View tương ứng hàm xử lý trong Controller

- ✓ Trong bảng Add View ta điền các thông số
  - Viewname: Tên của trang aspx ( trùng tên với hàm ta đang tạo view)
  - View data class: Lớp dữ liệu , controllers xử lý kết nối với trang
  - Mục create as a partial view để tạo partial tương ứng với user control
  - Use a layout page: Khu vực hiển thị ( chọn là trang Layout)



Hình 11: Màn hình chọn Layout cho View

Khi đó Visual Studio sẽ cho 1 trang aspx trong thư mục View/Theloaitin ta chỉ cần chỉnh sửa theo đúng ý mình:

### Index.cshtml

```
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

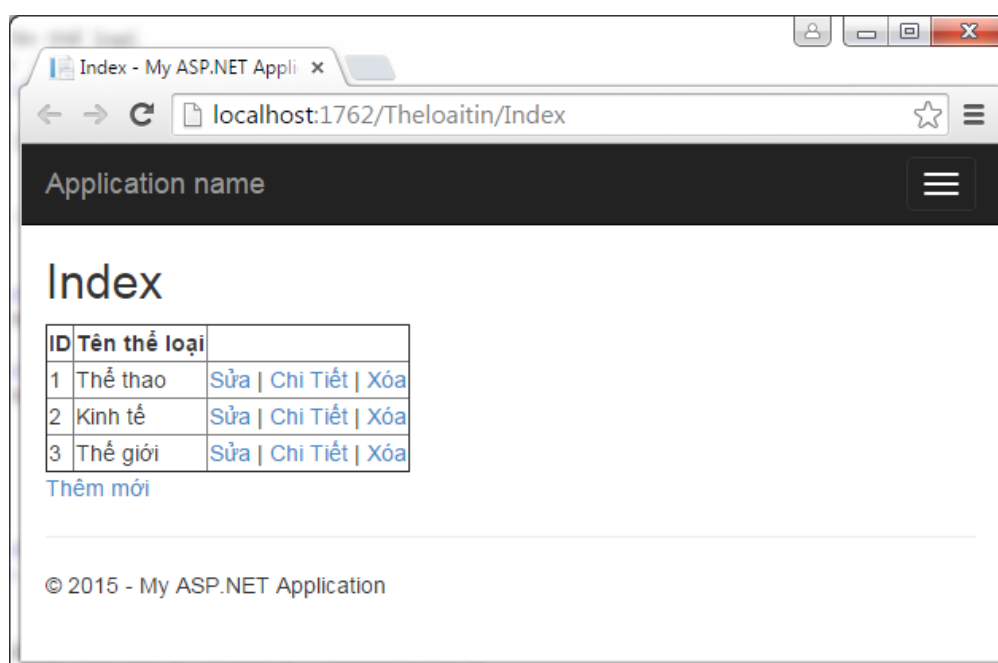
<h2>Index</h2>
<table border="1px">
    <tr>
        <th>
            ID
        </th>
        <th>
            Tên thể loại
        </th>
    </tr>
    @foreach (var item in Model)
```



```

{
    <tr>
        <td>
            <!-- Hiển thị ID của đối tượng với -->
            @item.IDLoai
        </td>
        <td>
            @item.Tentheloai
        </td>
        <td>
            <!-- Thẻ Html.ActionLink dùng để link đến 1 trang khác
            ( Tên Link hiển thị / Thư mục ) than số chuyển ra là ( id = id.Item)-->
            @Html.ActionLink("Sửa", "Edit", new { id = item.IDLoai }) |
            @Html.ActionLink("Chi Tiết", "Details", new { id = item.IDLoai }) |
            @Html.ActionLink("Xóa", "Delete", new { id = item.IDLoai })
        </td>
    </tr>
}
</table>
@Html.ActionLink("Thêm mới", "Create", "TheLoaiTinTuc")

```



Hình 12: Màn hình kết quả trang Index Thể loại tin

**Bài tập 6:** Tương tự chúng ta có các hàm khác trong Controllers như sau:

- Details : Hiển thị chi tiết
- Create : Tạo mới
- Edit : Sửa
- Delete : Xóa

*Ghi chú: Sinh viên nên lần lượt thực hiện từng hàm và tạo từng view (trong bài tập 7)*

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Lab02.Models;

namespace Lab02.Controllers

```

```
{
    public class TheloaitinController : Controller
    {
        //
        // GET: /TheLoaiTinTuc/
        DataClasses1DataContext data = new DataClasses1DataContext();
        public ActionResult Index()
        {
            var All_Loaitin = from tt in data.Theloaitins select tt;
            return View(All_Loaitin);
        }

        //Hàm Details truyền dữ liệu sang trang Details.aspx
        //Với tham số được truyền là IDLoai (trong bảng Theloaitin)
        public ActionResult Details(int id)
        {
            var Details_tin = data.Theloaitins.Where(m => m.IDLoai == id).First();
            return View(Details_tin);
        }

        //Hàm Create (get) tạo khung cho người sử dụng nhập liệu
        public ActionResult Create()
        {
            return View();
        }

        //Hàm Create(Post) xử lý dữ liệu được chuyển về từ trang Create.aspx
        //và trả về kết quả
        [HttpPost]
        public ActionResult Create(FormCollection collection, Theloaitin ltin)
        {
            // Tạo biến CB_Loaitin và gán giá trị của người dùng nhập vào từ
            //form trong trang Create.aspx
            var CB_Loaitin = collection["Tentheloai"];
            //Nếu CB_Loaitin có giá trị == null ( để trống )
            if (string.IsNullOrEmpty(CB_Loaitin))
            {
                ViewData["Loi"] = " Thể loại Tin không được để trống ";
            }
            else
            {
                ltin.Tentheloai = CB_Loaitin;
                data.Theloaitins.InsertOnSubmit(ltin);
                //Thực hiện tạo mới
                data.SubmitChanges();
                return RedirectToAction("Index");
            }
            return this.Create();
        }

        // GET:Hàm Edit(get) truyền thông số của đối tượng sang trang Edit.aspx
        //Với thông số là id.
        public ActionResult Edit(int id)
        {
            var EB_tin = data.Theloaitins.First(m => m.IDLoai == id);
            return View(EB_tin);
        }

        // POST: Hàm Edit(post) thực hiện update dữ liệu từ trang Edit.aspx
        //khi Click Submits
        [HttpPost]
        public ActionResult Edit(int id, FormCollection collection)
        {
            // Tạo một biến Ltin gán với đối tượng có id=id truyền vào
            var Ltin = data.Theloaitins.First(m => m.IDLoai == id);
            var E_Loaitin = collection["Tentheloai"];
            //vì ta sửa đối tượng lên Id của biến Ltin = Id chuyển vào .
            Ltin.IDLoai = id;
            // Nếu người dùng để phần Loại Tin trống báo lỗi

```

```

        if (string.IsNullOrEmpty(E_Loaitin))
        {
            ViewData["Loi"] = "Thẻ loại tin không được để trống ";
        }
        // Ngược lại gán các trường của biến Ltin bằng các giá trị
        // của người dùng nhập vào
        else
        {
            Ltin.Tentheloai = E_Loaitin;
            // Thực hiện updat
            UpdateModel(Ltin);
            data.SubmitChanges();
            return RedirectToAction("Index");
        }
        return this.Edit(id);
    }
    // GET: Hàm Delete ( get ) đưa dữ liệu của đối tượng cần xóa lên trang Delete
    // cho người dùng xem. Tham số truyền vào là id
    public ActionResult Delete(int id)
    {
        var D_tin = data.Theloitins.First(m => m.IDLoai == id);
        return View(D_tin);
    }
    // POST: Hàm Delete ( post ) thực thi lệnh xóa đối tượng khi người dùng
    // click xóa từ trang Delete.aspx . Với tham số Id

    [HttpPost]
    public ActionResult Delete(int id, FormCollection collection)
    {
        // Tạo biến D_Tin gán với đối tượng có ID bằng với ID tham số
        var D_tin = data.Theloitins.Where(m => m.IDLoai == id).First();
        //xóa
        data.Theloitins.DeleteOnSubmit(D_tin);
        data.SubmitChanges();
        return RedirectToAction("Index");
    }
}

```

**Bài tập 7:** Xây dựng các View hiển thị dữ liệu xử lý trong các hàm tương ứng

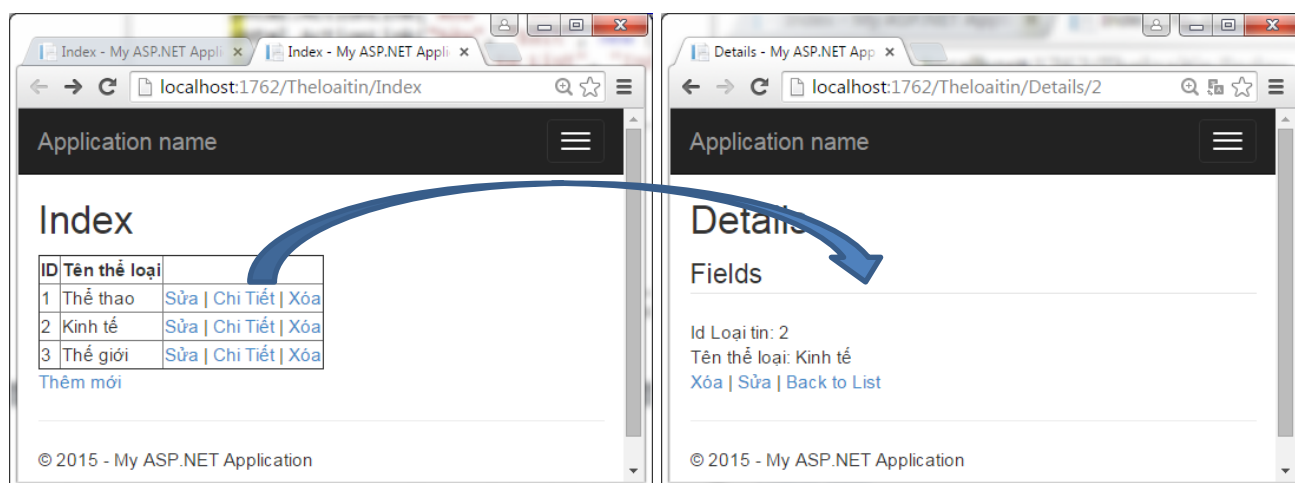
**View: Details.cshtml**

```

@{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Details</h2>
<fieldset>
    <legend>Fields</legend>
    <div class="display-label">Id Loại tin: @Model.IDLoai</div>
    <div class="display-label">Tên thẻ loại: @Model.Tentheloai</div>
</fieldset>
@using (Html.BeginForm()) {
    <p>
        @Html.ActionLink("Xóa ", "Delete", new { id = Model.IDLoai })|
        @Html.ActionLink("Sửa", "Edit", new { id = Model.IDLoai }) |
        @Html.ActionLink("Back to List", "Index")
    </p>
}

```

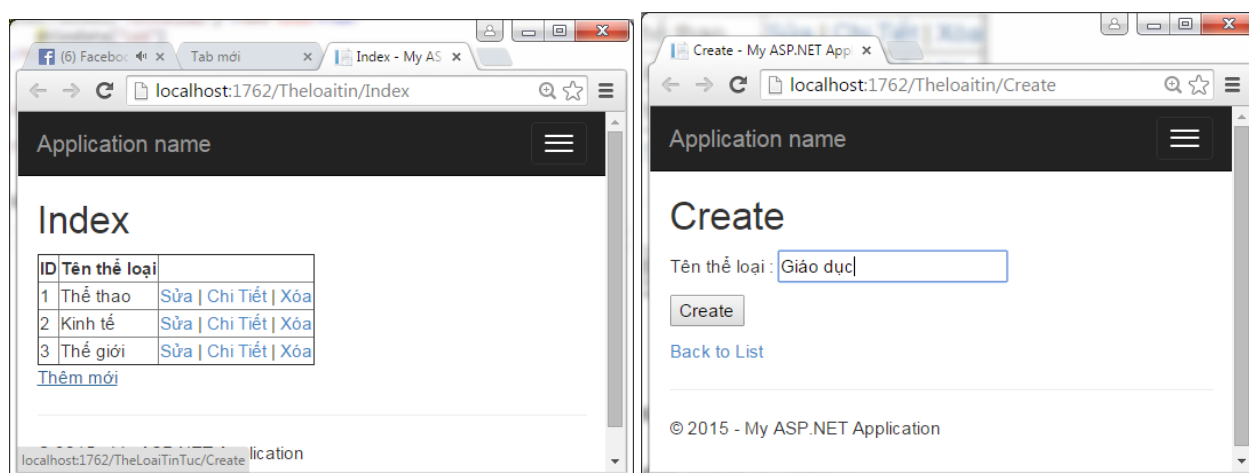


Hình 13: Màn hình kết quả trang Details Thẻ loại tin

**View: Create.cshtml**

```
@{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Create</h2>
@using (Html.BeginForm()) {
    @Html.ValidationSummary(true)
    <fieldset>
        <p>
            Tên thể loại :
            @Html.TextBox("Tentheloai") <font color=red>
                @ViewData["Loi"]
            </font>
        </p>
        <p>
            <input type="submit" value="Create" />
        </p>
    </fieldset>
}
<div>
    @Html.ActionLink("Back to List", "Index")
</div>
```



Hình 14: Màn hình kết quả trang Create (Thêm mới) Thẻ loại tin

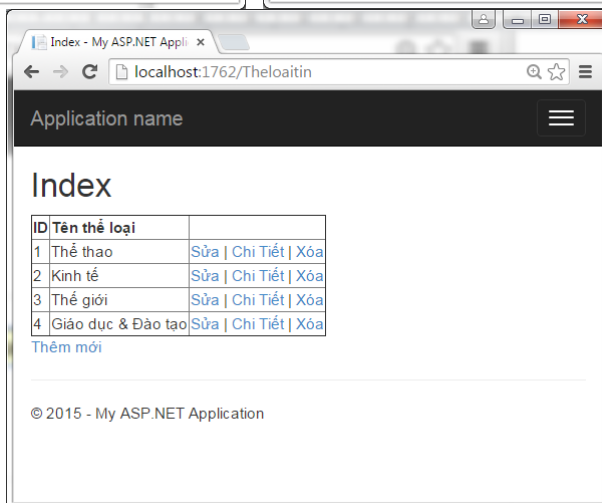
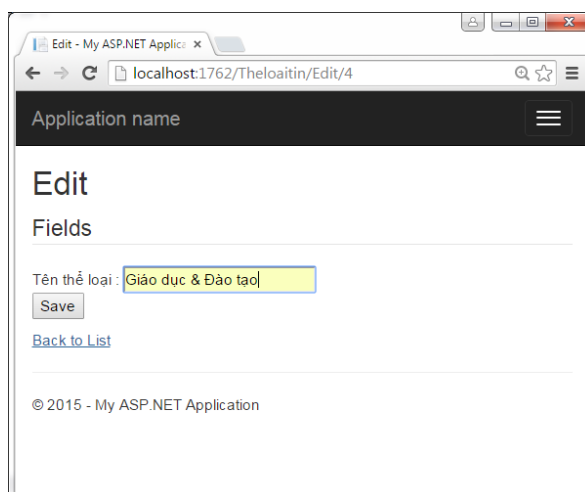
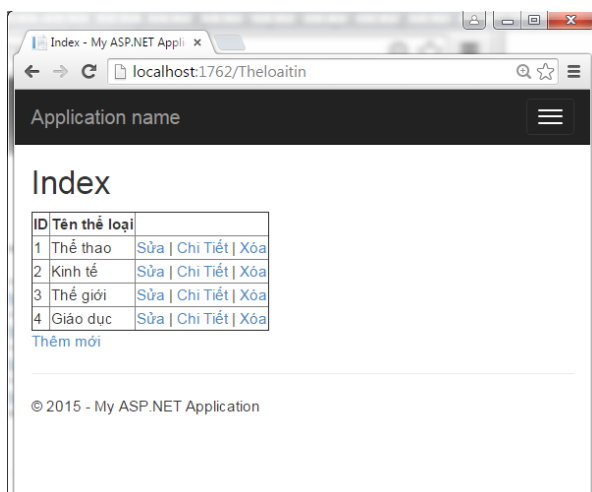
## View: Edit.cshtml

```

@{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Edit</h2>
@using (Html.BeginForm()) {
    @Html.ValidationSummary(true)
    <fieldset>
        <legend>Fields</legend>
        <div class="editor-field">
            Tên thể loại :
            @Html.TextBox("Tentheloai")
            @ViewData["Loi"]
        </div>
        <p>
            <input type="submit" value="Save" />
        </p>
    </fieldset>
}
<div>
    @Html.ActionLink("Back to List", "Index")
</div>

```

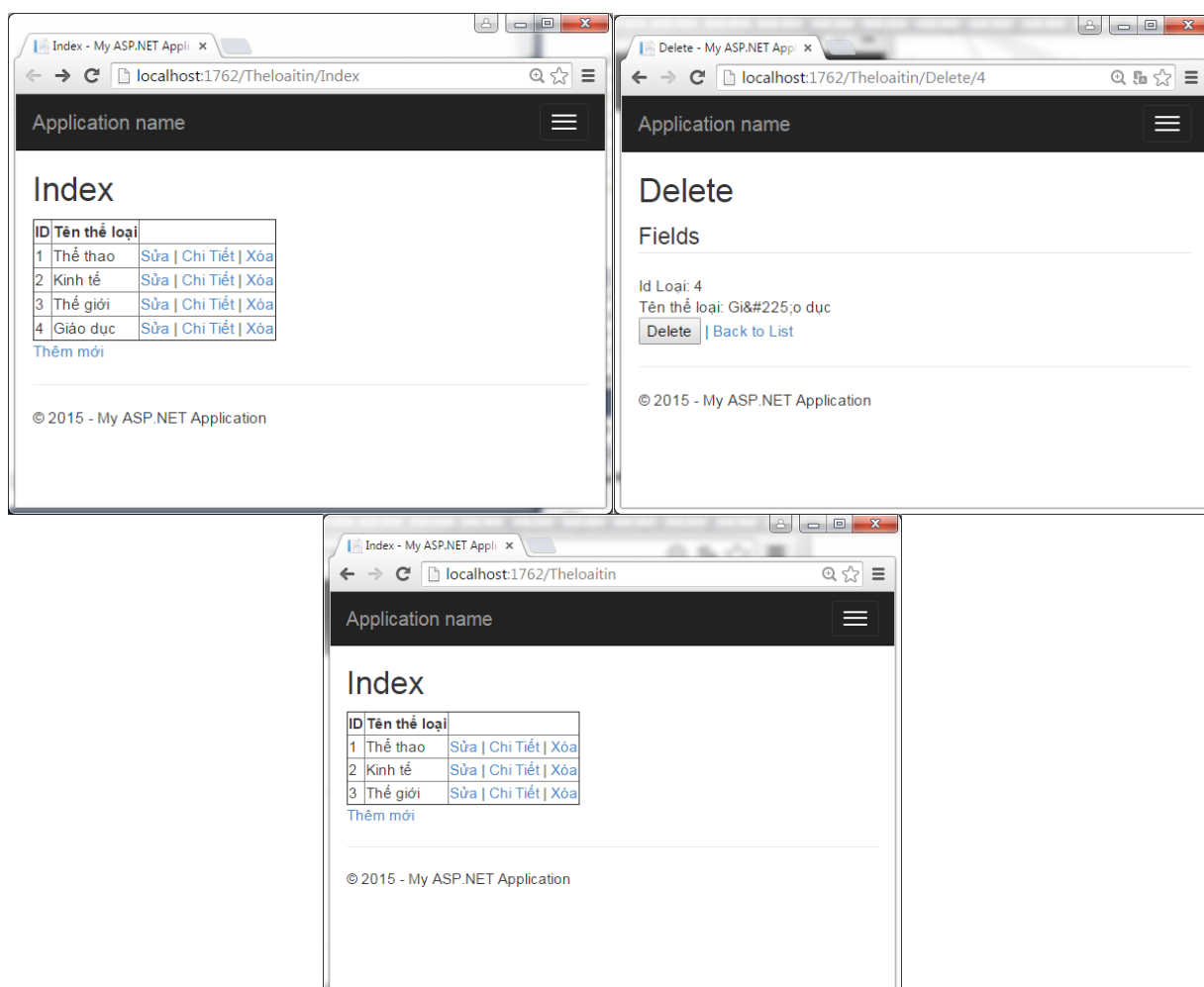


Hình 15: Màn hình kết quả trang Edit (Điều chỉnh) Thể loại tin

## View: Delete.cshtml

```
@{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Delete</h2>
<fieldset>
    <legend>Fields</legend>
    <div class="display-label">Id Loại: @Html.Encode(Model.IDLoai)</div>
    <div class="display-label">Tên thể loại: @Html.Encode(Model.Tentheloai)</div>
</fieldset>
@using (Html.BeginForm()) {
    <p>
        <input type="submit" value="Delete" /> |
        @Html.ActionLink("Back to List", "Index")
    </p>
}
```



Hình 16: Màn hình kết quả trang Delete (Xóa) Thể loại tin

Như vậy là đã xử lý xong phần Thể loại tin. Sau khi đã hoàn thành ta sẽ phải sửa đổi lại file Layout để có đường dẫn tới các trang mà chúng ta vừa tạo. Mở file Layout: **Shared/\_Layout.cshtml**

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - My ASP.NET Application News</title>
    @Styles.Render("~/Content/css")
```

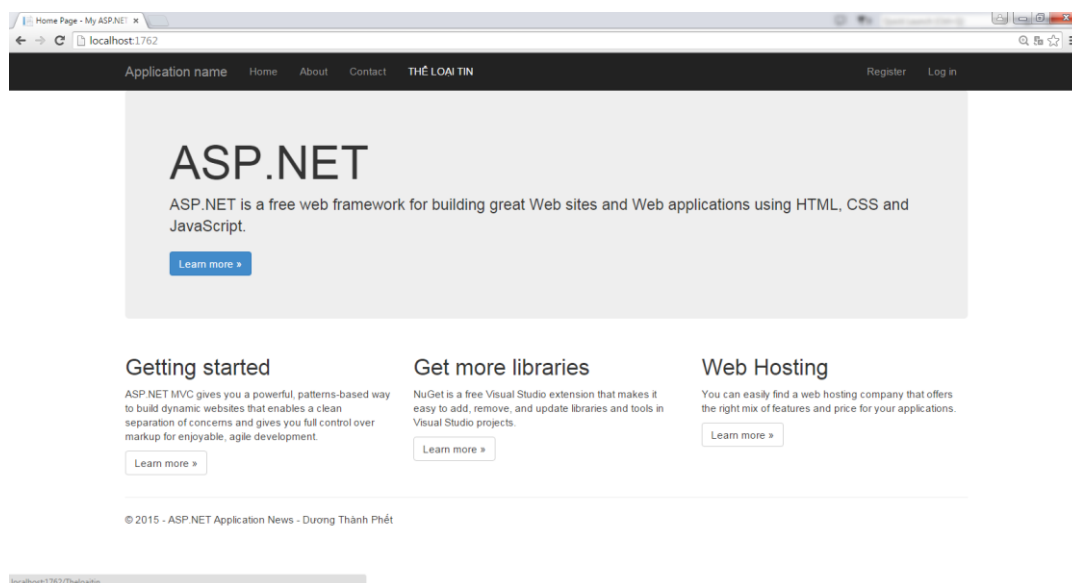
```

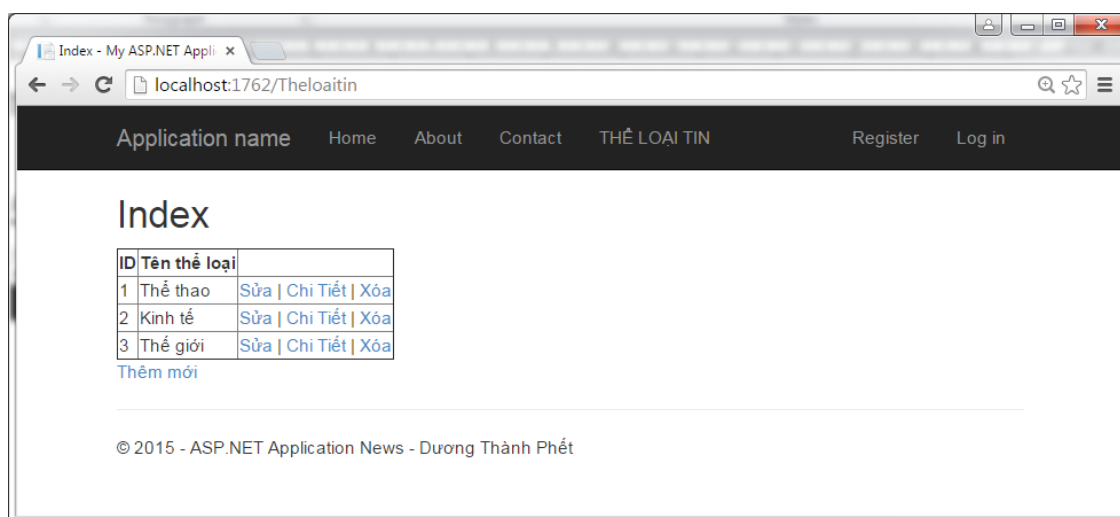
@Scripts.Render("~/bundles/modernizr")
</head>
<body>
    <div class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse"
                    data-target=".navbar-collapse">

                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                @Html.ActionLink("Application name", "Index", "Home", null,
                    new { @class = "navbar-brand" })
            </div>
            <div class="navbar-collapse collapse">
                <ul class="nav navbar-nav">
                    <li>@Html.ActionLink("Home", "Index", "Home")</li>
                    <li>@Html.ActionLink("About", "About", "Home")</li>
                    <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
                    <li>@Html.ActionLink("THỂ LOẠI TIN", "Index", "TheLoaitin")</li>
                </ul>
                @Html.Partial("_LoginPartial")
            </div>
        </div>
    </div>
    <div class="container body-content">
        @RenderBody()
        <hr />
        <footer>
            <p>&copy; @DateTime.Now.Year - ASP.NET Application News
                - Dương Thành Phết</p>
        </footer>
    </div>
    @Scripts.Render("~/bundles/jquery")
    @Scripts.Render("~/bundles/bootstrap")
    @RenderSection("scripts", required: false)
</body>
</html>

```

## Kết quả:





Hình 17: Màn hình kết quả với Layout đã điều chỉnh

**Bài tập 8: Xử lý bảng Tin tức:***Tương tự như xử lý bảng Thể loại tin***✓ Tạo file TintucControllers.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Lab02.Models;
namespace Lab02.Controllers
{
    public class TintucController : Controller
    {
        //
        // GET: /TinTuc/
        //Kết nối CSDL
        DataClasses1DataContext data = new DataClasses1DataContext();
        //Hàm Index truyền dữ liệu sang trang Index.aspx
        // để hiển thị danh sách tin tức
        public ActionResult Index()
        {
            // Khởi tạo biến All_tin và gán với đối tượng trong bảng New
            var All_tin = from tt in data.Tintucs select tt;
            //Truyền dữ liệu sang trang Index.aspx
            return View(All_tin);
        }
        //Hàm Details truyền dữ liệu của một đối tượng sang trang Details.aspx
        // với tham số truyền vào là Id của đối tượng
        public ActionResult Details(int id)
        {
            var D_tin = data.Tintucs.First(m => m.IdTin == id);
            return View(D_tin);
        }
        // GET: Hàm Create(get) Tạo một khung cho người dùng nhập liệu
        public ActionResult Create()
        {
            var L_tin = from lt in data.Theloatins select lt;
            ViewData["Loaitin"] = new SelectList(data.Theloatins, "IdLoai",
                                                    "Tentheloai");
            return View();
        }
        // POST: Hàm Create(post) Nhận dữ liệu từ trang Create
        // và thực hiện việc tạo mới dữ liệu
        [HttpPost]

```



```

public ActionResult Create(FormCollection collection, Tintuc TTin)
{
    // Gán các giá trị người dùng nhập liệu cho các biến
    var N_tin = collection["Tieudetin"];
    var L_tin = int.Parse(collection["Loaitin"]);
    var C_tin = collection["Noidungtin"];
    if (String.IsNullOrEmpty(N_tin))
    {
        ViewData["Loi"] = "Tiêu đề tin không được để trống";
    }
    else if (String.IsNullOrEmpty(C_tin))
    {
        ViewData["Loi1"] = "Bạn hãy nhập nội dung cho tin";
    }
    else
    {
        //Gán giá trị cho đối tượng được tạo mới (TTin)
        TTin.IDLoai = L_tin;
        TTin.Tieudetin = N_tin;
        TTin.Noidungtin = C_tin;
        //Thực hiện tạo mới
        data.Tintucs.InsertOnSubmit(TTin);
        data.SubmitChanges();
        return RedirectToAction("Index");
    }
    return this.Create();
}
// GET: Hàm Edit (get) truyền dữ liệu tới file Edit.aspx
// Với tham số được truyền vào là IDTin
public ActionResult Edit(int id)
{
    var E_tin = data.Tintucs.First(m => m.IdTin == id);
    var L_tin = from lt in data.TheLoaitins select lt;
    ViewData["Loaitin"] = new SelectList(data.TheLoaitins, "IDLoai",
                                         "Tentheloai");

    return View(E_tin);
}
// POST: Hàm Edit(post) nhận dữ liệu mà người dùng
//nhập từ trang Edit.aspx và thực hiện sửa đổi
[HttpPost]
public ActionResult Edit(int id, FormCollection collection)
{
    var N_tin = collection["Tieudetin"];
    var L_tin = int.Parse(collection["Loaitin"]);
    var C_tin = collection["Noidungtin"];
    var Etin = data.Tintucs.First(m => m.IdTin == id);
    if (String.IsNullOrEmpty(N_tin))
    {
        ViewData["Loi"] = "Tiêu đề tin không được để trống";
    }
    else if (String.IsNullOrEmpty(C_tin))
    {
        ViewData["Loi1"] = "Bạn hãy nhập nội dung cho tin";
    }
    else
    {
        Etin.IDLoai = L_tin;
        Etin.Tieudetin = N_tin;
        Etin.Noidungtin = C_tin;
        UpdateModel(Etin);
        data.SubmitChanges();
        return RedirectToAction("Index");
    }
    return this.Edit(id);
}
// GET: Hàm Delete(get) truyền dữ liệu tới trang Delete.aspx

```

```
// với tham số ID của mục cần xóa
public ActionResult Delete(int id)
{
    var D_Tin = data.Tintucs.First(m => m.IdTin == id);
    return View(D_Tin);
}
// POST: Hàm Delete(get) thực hiện lệnh xóa
// với tham số truyền vào là id mục cần xóa
[HttpPost]
public ActionResult Delete(int id, FormCollection collection)
{
    var D_tin = data.Tintucs.Where(m => m.IdTin == id).First();
    data.Tintucs.DeleteOnSubmit(D_tin);
    data.SubmitChanges();
    return RedirectToAction("Index");
}
}
```

### ✓ Xử lý View:

Click phải chuột vào View => add => New Folder ➔ **Tintuc**

Trong File TintucControllers.cs ở trên ta có 5 hàm chính :

- Index : Hiển thị danh sách
- Details : Hiển thị chi tiết
- Create : Tạo mới
- Edit : Sửa
- Delete : Xóa

Tương ứng với 5 hàm này ta sẽ có 5 trang hiển thị aspx trong folder Tintuc

### View: Index.cshtml

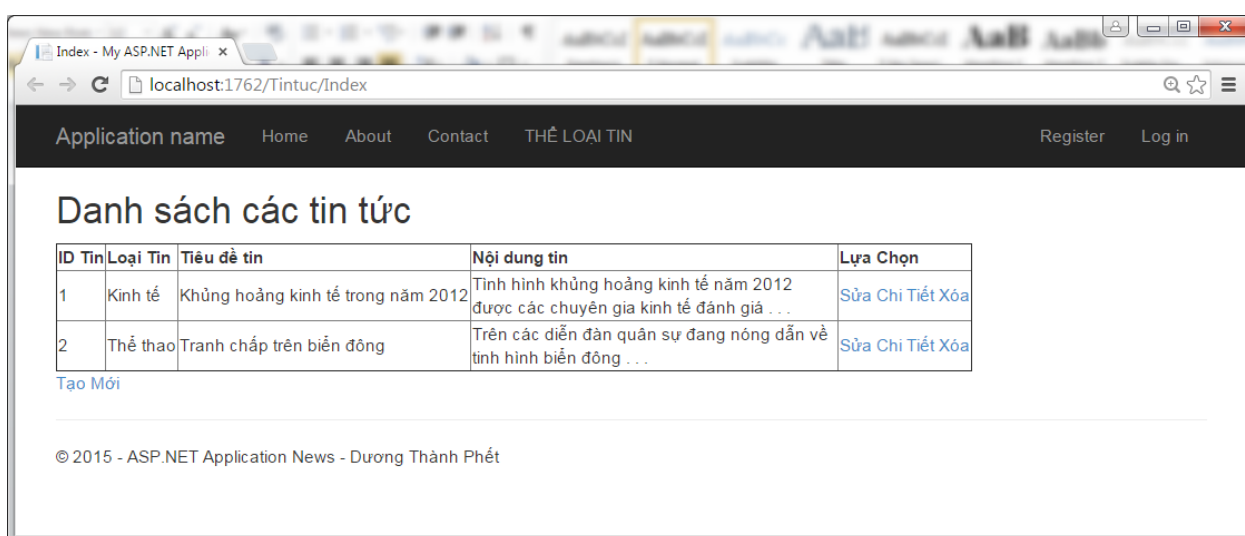
```
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Danh sách các tin tức</h2>
<table border="1">
    <tr>
        <th>
            ID Tin
        </th>
        <th>
            Loại Tin
        </th>
        <th>
            Tiêu đề tin
        </th>
        <th>
            Nội dung tin
        </th>
        <th>
            Lựa Chọn
        </th>
    </tr>
    @foreach (var item in Model) {
        <tr>
            <td>
                @item.IdTin
            </td>
```

```

        <td>
            @item.Theloaitin.Tentheloai
        </td>
        <td>
            @item.Tieudetin
        </td>
        <td width=300>
            @item.Noidungtin
        </td>
        <td>
            @Html.ActionLink("Sửa", "Edit", new { id=item.IdTin })
            @Html.ActionLink("Chi Tiết", "Details", new { id=item.IdTin })
            @Html.ActionLink("Xóa", "Delete", new { id=item.IdTin })
        </td>
    </tr>
}
</table>
<p> @Html.ActionLink("Tạo Mới", "Create")</p>

```



Hình 18: Màn hình kết quả trang Index Tin tức

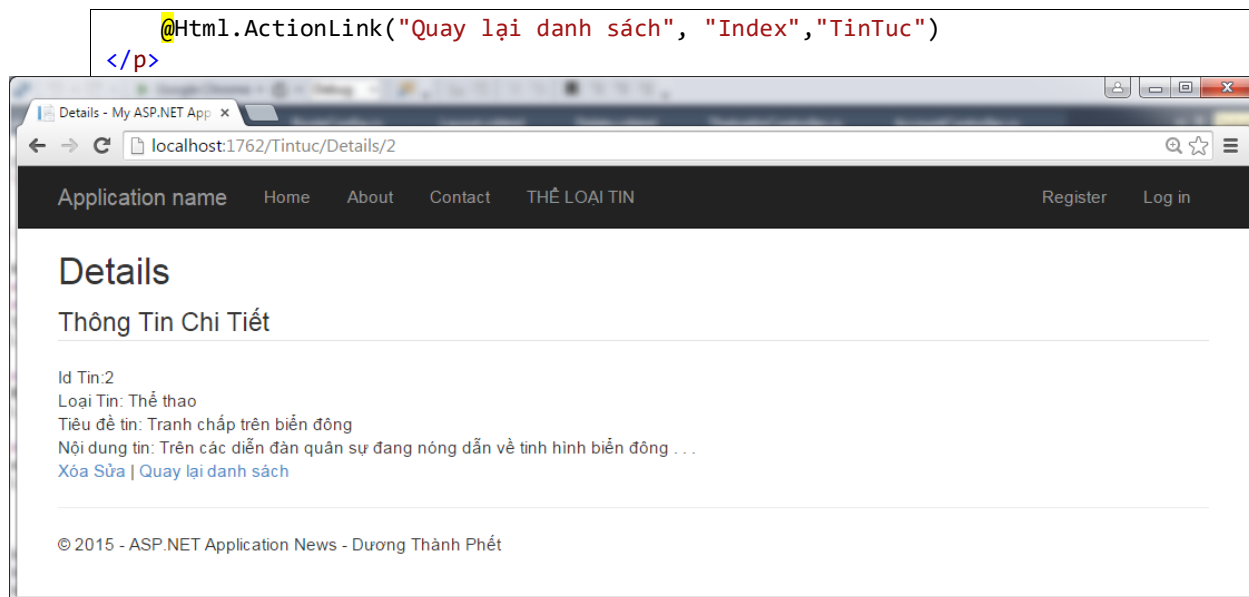
**View: Details.cshtml**

```

@{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Details</h2>
<fieldset>
    <legend>Thông Tin Chi Tiết</legend>
    <div class="display-label">Id Tin:@Model.IdTin</div>
    <div class="display-label">
        Loại Tin:
        @Model.Theloaitin.Tentheloai
    </div>
    <div class="display-label">
        Tiêu đề tin:
        @Model.Tieudetin
    </div>
    <div class="display-label">
        Nội dung tin:
        @Model.Noidungtin
    </div>
</fieldset>
<p>
    @Html.ActionLink("Xóa", "Delete", new { id=Model.IdTin })
    @Html.ActionLink("Sửa", "Edit", new { id=Model.IdTin }) |

```

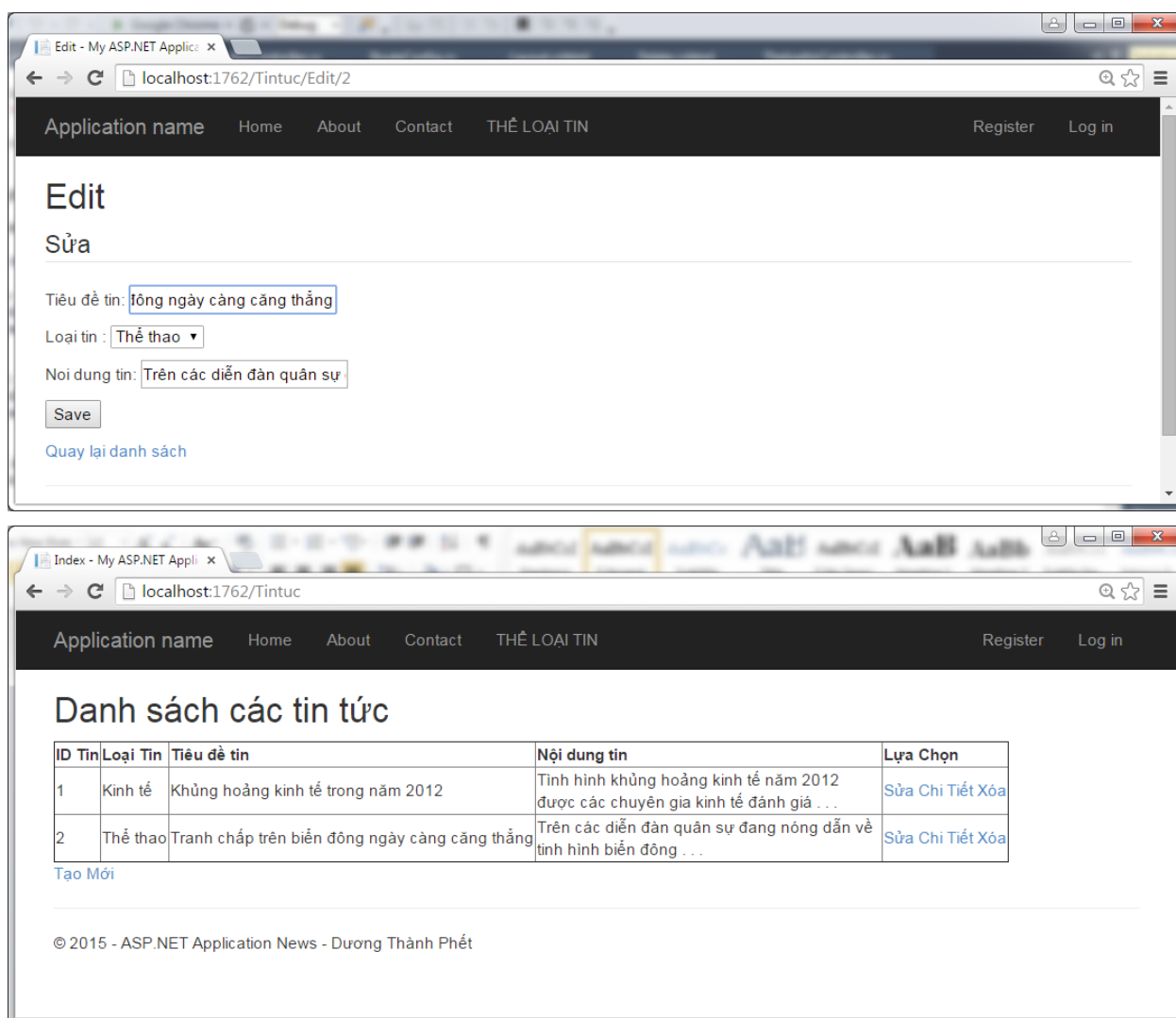


Hình 19: Màn hình kết quả trang Details (Chi tiết Tin tức)

**View: Edit.cshtml**

```
@{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
@model Lab02.Models.Tintuc

<h2>Edit</h2>
@using (Html.BeginForm())
{
    @Html.ValidationSummary(true)
    <fieldset>
        <legend>Sửa</legend>
        <p>
            Tiêu đề tin:
            @Html.TextBoxFor(m => m.Tieudetin)
            @ViewData["Loi"]
        </p>
        <p>
            Loại tin :
            @Html.DropDownList("Loaitin")
            @ViewData["Loi1"]
        </p>
        <p>
            Noi dung tin:
            @Html.TextBoxFor(m => m.Noidungtin)
        </p>
        <p>
            <input type="submit" value=Save />
        </p>
    </fieldset>
}
<div> @Html.ActionLink("Quay lại danh sách", "Index", "TinTuc") </div>
```



Hình 20: Màn hình kết quả trang Edit (Điều chỉnh Tin tức)

**View: Create.cshtml**

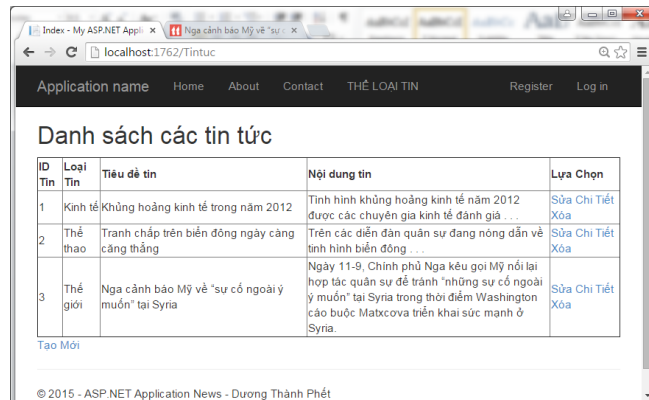
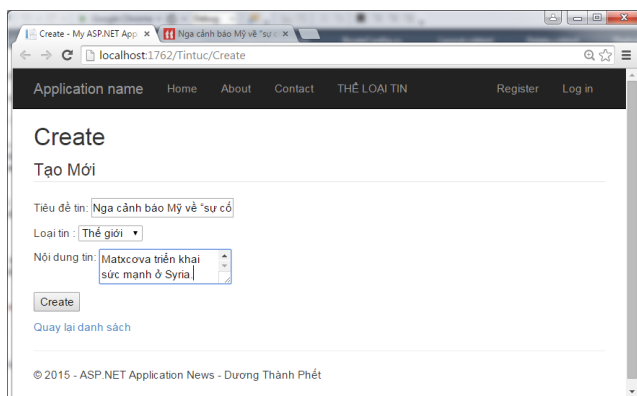
```
@{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Create</h2>
@using (Html.BeginForm())
{
    @Html.ValidationSummary(true)
    <fieldset>
        <legend>Tạo Mới</legend>
        <p>
            Tiêu đề tin:
            @Html.TextBox("Tieudetin")
            @ViewData["Loi"]
        </p>
        <p>
            Loại tin :
            @Html.DropDownList("Loaitin")
        </p>
        <p>
            Nội dung tin:
            @Html.TextArea("Noidungtin")
        </p>
    </fieldset>
}
```

```

        <input type="submit" value="Create" />
    </p>
</fieldset>
}
<div>
    @Html.ActionLink("Quay lại danh sách", "Index")
</div>

```



Hình 21: Màn hình kết quả trang Create (Thêm mới Tin tức)

**View: Delete.cshtml**

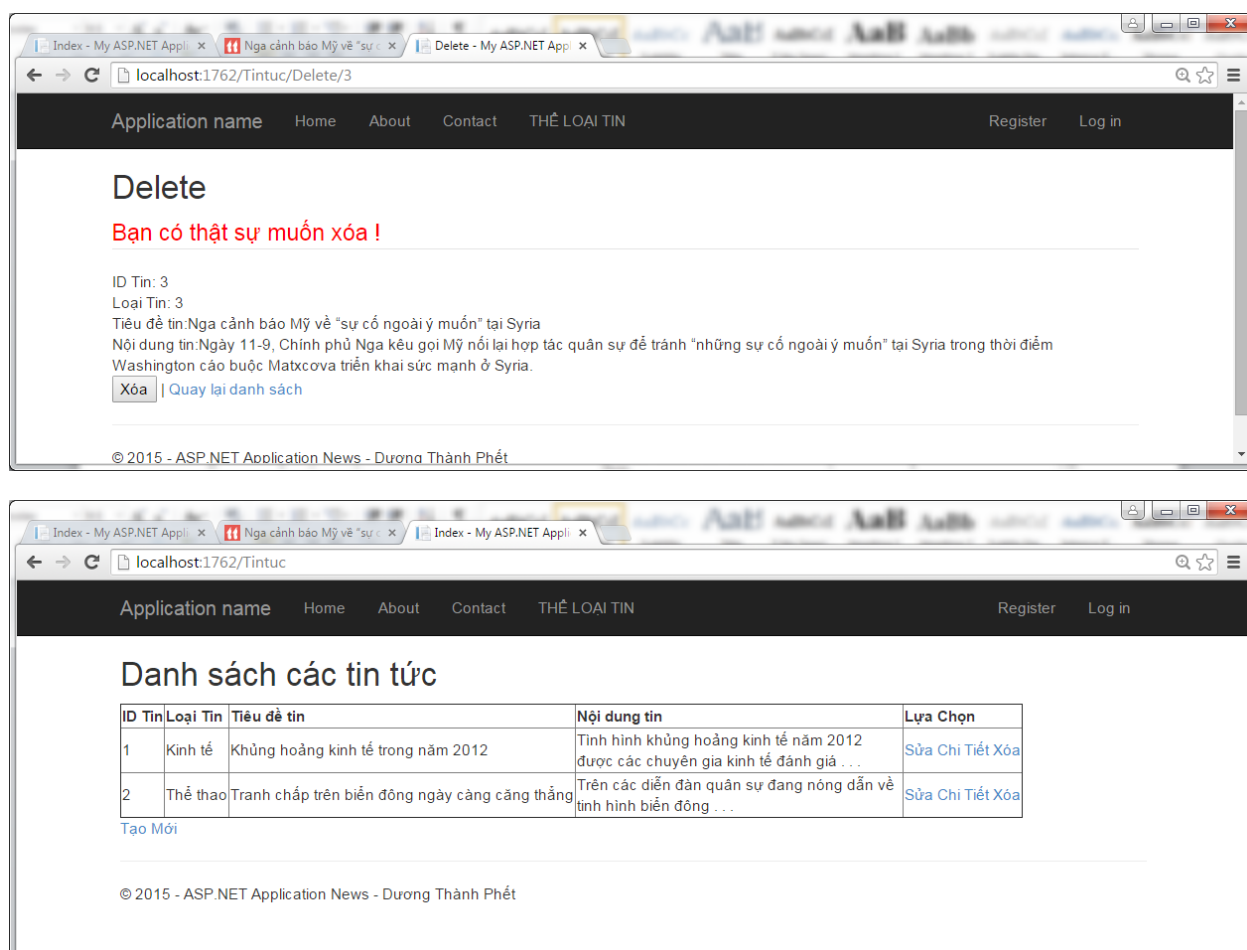
```

@{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Delete</h2>
<fieldset>
    <legend><font color=red>Bạn có thật sự muốn xóa !</font></legend>
    <div class="display-label">ID Tin: @Model.IdTin</div>
    <div class="display-label">Loại Tin: @Model.Theloaitin</div>
    <div class="display-label">Tiêu đề tin:@Model.Tieudetin</div>
    <div class="display-label">Nội dung tin:@Model.Noidungtin</div>
</fieldset>
    @using (Html.BeginForm()) {
    <p>
        <input type="submit" value="Xóa" /> |
        @Html.ActionLink("Quay lại danh sách", "Index")
    </p>
    }

```





Hình 22: Màn hình kết quả trang Delete (Xóa Tin tức)

-----Hết Lab 02-----