TỰ CODE CÁC TRANG XEM, THÊM, SỬA, XÓA

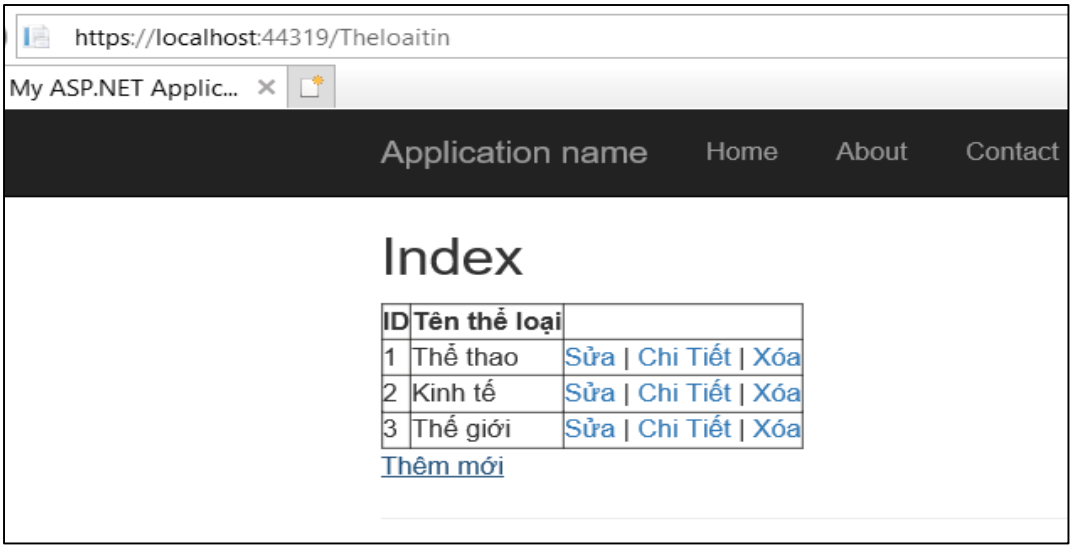
❖Trong SQLServer tạo CSDL **Tintuc** gồm 2 bảng:

+ **Theloaitin**(IDLoai int, Tentheloai 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,
  Tentheloai nvarchar(100)
 )
Go
Create Table Tintuc
 (
  IdTin int identity Primary key,
  IDLoai int references Theloaitin(IDLoai),
  Tieudetin nvarchar(100),
  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'Thế 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 . . .')
```

Tạo ứng dụng Web MVC với các chức năng: xem, thêm, sửa, xóa theo mẫu sau:



- Tạo một project đặt tên là **MvcTinTuc**, chọn mẫu **MVC**.
- Cài đặt EntityFramework sử dụng NuGet Package Manager

- Sử dụng EF code first để kết nối với cơ sở dữ liệu **MvcTinTuc** (đặt tên ADO.NET Entity Model là **QLTintucDB**) chọn tất cả các bảng trong cơ sở dữ liệu.
- Kích **ReBuild Solution** để build lại project (mỗi khi sửa model cần Rebuild lại hệ thống)

❖Xử lý Controller

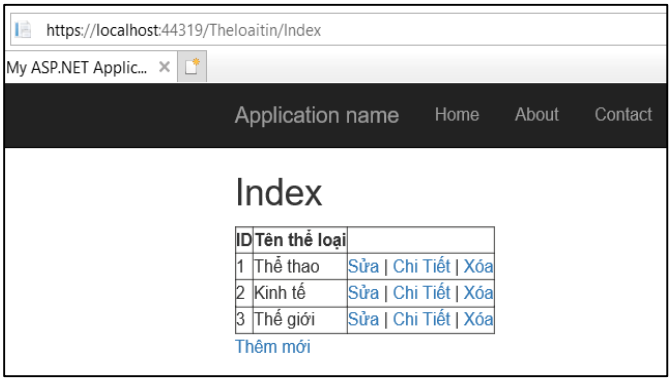
- Tạo mới Controller-Empty: với tên **TheloaitinController**
- Bổ sung Code vào file **TheloaitinControllers.cs** như sau :

```
using McvTinTuc.Models;
namespace McvTinTuc.Controllers
 {
   public class TheloaitinController : Controller
   {
        QLTintucDB db = new QLTintucDB();
        // GET: Theloaitin
        public ActionResult Index()
        {
            var All_Loaitin = from tt in db.Theloaitins select tt;
            return View(All_Loaitin);
        }
   }
 }
```

- Tạo View **Index** tương ứng:

```
@{
    ViewBag.Title = "Index";
}
<h2>Index</h2>
<table border="1px">
    <tr>
        <th>
            ID
        </th>
        <th>
            Tên thể loại
        </th>
        <th></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
                     tham số truyền 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", "Theloaitin")
```

✓ Chạy View Index:



❖**Tương tự chúng ta có các hàm khác trong Controllers như sau:**

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

```
using McvTinTuc.Models;
namespace McvTinTuc.Controllers
{
    0 references
    public class TheloaitinController : Controller
    {
        QLTintucDB db = new QLTintucDB();
        // GET: Theloaitin
        0 references
        public ActionResult Index()
        {
            var All_Loaitin = from tt in db.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)
        0 references
        public ActionResult Details(int id)
        {
            var Details_tin = db.Theloaitins.Where(m => m.IDLoai == id).First();
            return View(Details_tin);
        }
        //Hàm Create (get )tạo khung cho người sử dụng nhập liệu
        1 reference
        public ActionResult Create()
        {
            return View();
        }
        //Hàm Create(Post) xử lý dữ liệu được truyền về từ trang Create.aspx
        //và trả về kết quả
        [HttpPost]
        0 references
        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;
```

```
            db.Theloaitins.Add(ltin);
        //Thực hiện tạo mới
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return this.Create();
}
// GET:Hàm Edit(get) t ruyền thông số của đối tượng sang trang Edit.aspx
//Với thông số là id.
public ActionResult Edit(int id)
{
    var EB_tin = db.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 = db.Theloaitins.First(m => m.IDLoai == id);
    var E_Loaitin = collection["Tentheloai"];
    //vì ta sửa đối tượng lên Id của biến Ltin = Id truyền vào
    Ltin.IDLoai = id;
    // Nếu người dùng để phần Loại Tin trống báo lỗi
    if (string.IsNullOrEmpty(E_Loaitin))
    {
        ViewData["Loi"] = "Thể laoại 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 update
        UpdateModel(Ltin);
        db.SaveChanges();
        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 = db.Theloaitins.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 = db.Theloaitins.Where(m => m.IDLoai == id).First();
         //xóa
         db.Theloaitins.Remove(D_tin);
         db.SaveChanges();
         return RedirectToAction("Index");
      }
  }
}
```

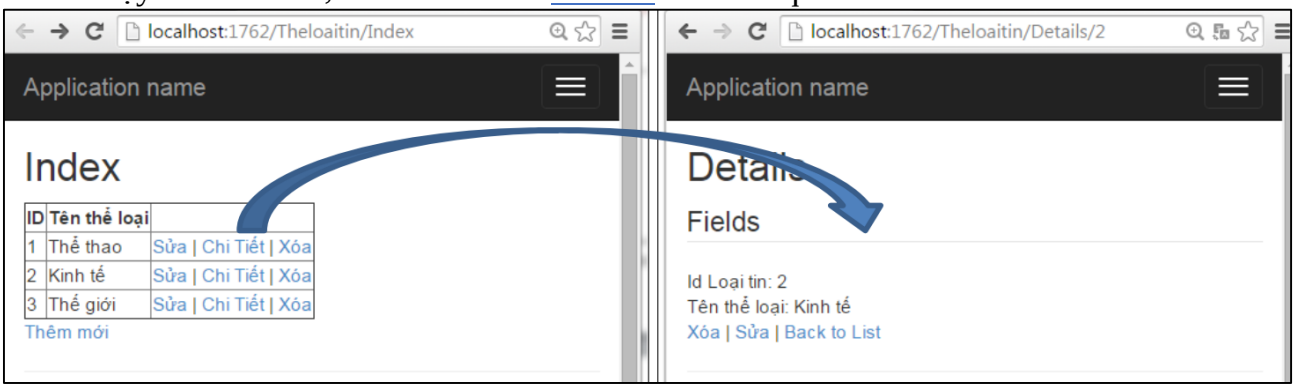
❖Xây dựng các View hiện thị dữ liệu xử lý trong các hàm tương ứng

- **View: Details.cshtml**

```
@{
    ViewBag.Title = "Details";
}

<h2>Details</h2>
<fieldset>
    <legend>Fields</legend>
    <div class="display-label">Id Loại tin: @Model.IDLoai</div>
    <div class="display-label">Tên thể 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>
}
```

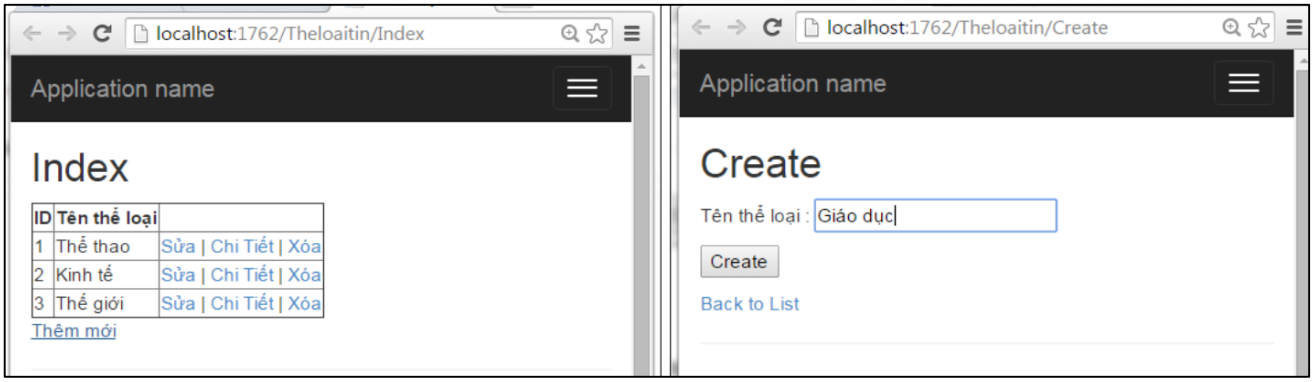
✓ Chạy View Index, click vào Link Chi tiết xem kết quả:



- **View: Create.cshtml**

```
@{
    ViewBag.Title = "Create";
}
<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>
```

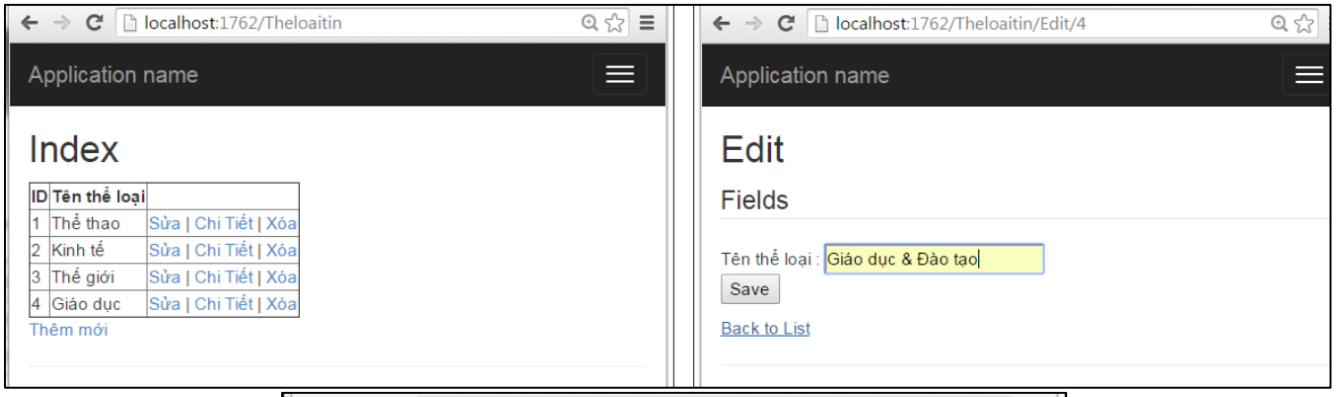
✓ Chạy View Index, click vào Link Thêm xem kết quả:

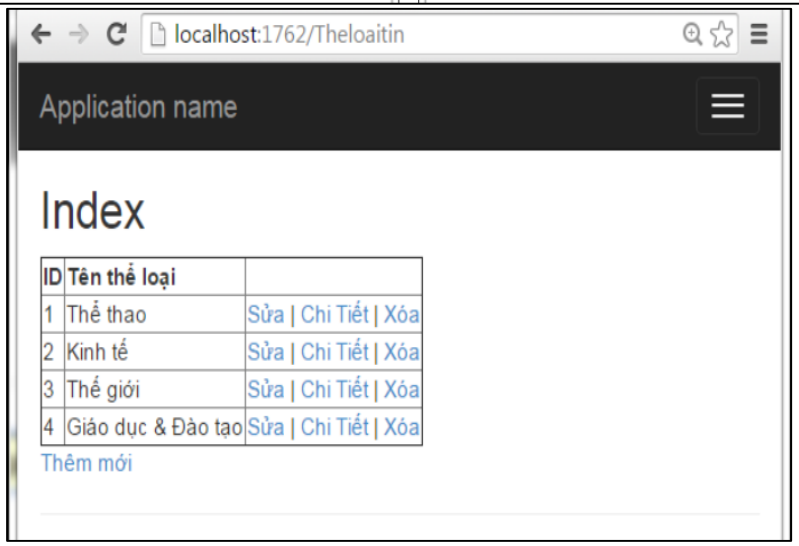


- **View: Edit.cshtml**

```
@{
    ViewBag.Title = "Edit";
 }
<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>
```

Chạy View Index, click vào Link Sửa xem kết quả:

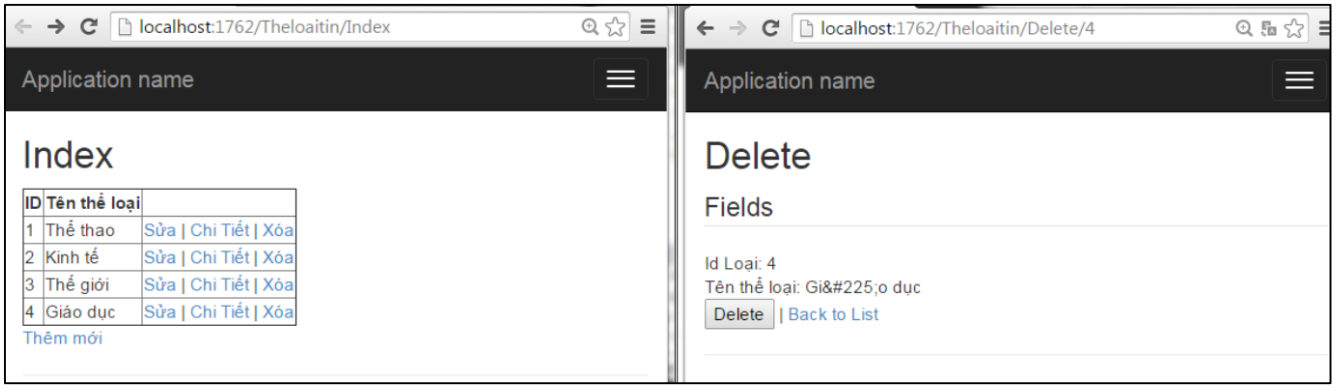




•View: Delete.cshtml

```
@{
    ViewBag.Title = "Delete";
 }
<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>
 }
```

✓ Chạy View Index, click vào Link Xóa xem kết quả:



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**

```
<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>
```

✓ F5 chạy, xem kết quả: