

TỰ CODE CÁC TRANG XEM, THÊM, SỬA, XÓA

- ❖ Trong SQLServer tạo CSDL **QLLuong** gồm 1 bảng:
+ **NhanVien**(Manv, Hoten, Phong, Luong)

DESKTOP-2KB389E\...e1 - dbo.NhanVien			
	Column Name	Data Type	Allow Nulls
Manv		char(10)	<input type="checkbox"/>
Hoten		nvarchar(50)	<input checked="" type="checkbox"/>
Phong		nvarchar(50)	<input checked="" type="checkbox"/>
Luong		float	<input checked="" type="checkbox"/>

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

QUẢN LÝ NHÂN VIÊN

Mã NV/Họ tên	Tên phòng	Lương	
01 Lê Thị Hà	Tài vụ	4000000	Sửa Chi tiết Xóa
02 Trần Minh Đại	Tài vụ	5000000	Sửa Chi tiết Xóa
03 Đỗ Thị Vân	Kỹ thuật	7000000	Sửa Chi tiết Xóa
04 Ngô Văn Hà	Kỹ thuật	6000000	Sửa Chi tiết Xóa

[Thêm mới](#)

Index.cshtml

THÊM NHÂN VIÊN

Mã nhân viên:

Họ tên:

Tên phòng:

Lương:

| [Về danh sách](#)

Them.cshtml

SỬA NHÂN VIÊN

Họ tên:

Tên phòng:

Lương:

| [Về danh sách](#)

Sua.html

XÓA NHÂN VIÊN

Mã nhân viên: 01
Họ tên: Lê Thị Hà
Tên phòng: Tài vụ
Lương: 4000000

| [Về danh sách](#)

Xoa.cshtml

XEM CHI TIẾT

Mã nhân viên: 01
Họ tên: Lê Thị Hà
Tên phòng: Tài vụ
Lương: 4000000

[Sửa](#) | [Xóa](#) | [Về danh sách](#)

ChiTiet.cshtml

- Tạo một project đặt tên là **MvcLuong**, chọn mẫu **MVC**.
- Cài đặt EntityFramework sử dụng NuGet Package Manager
- Sử dụng EF code first để kết nối với cơ sở dữ liệu **QLLuong** (đặt tên ADO.NET Entity Model là **QLLuongDB**) 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 **NhanvienController**
- Bổ sung Code vào file **NhanvienControllers.cs** như sau :

```
using MvcLuong.Models;
namespace MvcLuong.Controllers
{
    public class NhanvienController : Controller
    {
        QLLuongDB db = new QLLuongDB();

        [HttpGet]
        public ActionResult Index()
        {
            var query = db.NhanViens.Select(p => p);
            return View(query);
        }
    }
}
```

- Tạo View **Index** tương ứng:

```
@{
    ViewBag.Title = "Index";
}
<h3>QUẢN LÝ NHÂN VIÊN</h3>
<table border="1" style="font-size:14px">
    <tr>
        <th>Mã NV</th>
        <th>Họ tên</th>
        <th>Tên phòng</th>
        <th>Lương</th>
        <th></th>
    </tr>
    @foreach (var item in Model)
    {
        <tr>
            <td>@item.Manv</td>
            <td>@item.Hoten</td>
            <td>@item.Phong</td>
            <td>@item.Luong</td>
            <td>
                @Html.ActionLink("Sửa", "Sua", new {id = item.Manv.Trim()})|
                @Html.ActionLink("Chi tiết", "ChiTiet", new { id = item.Manv.Trim()})|
                @Html.ActionLink("Xóa", "Xoa", new {id = item.Manv.Trim()})
            </td>
        </tr>
    }
</table>
@Html.ActionLink("Thêm mới", "Them", "Nhanvien")
```

✓ Chạy View Index xem kết quả

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

- ChiTiet : Hiển thị chi tiết
- Them: Tạo mới
- Sua: Sửa
- Xoa: Xóa

```

using MvcLuong.Models;
namespace MvcLuong.Controllers
{
    public class NhanvienController : Controller
    {
        QLLuongDB db = new QLLuongDB();

        [HttpGet]
        public ActionResult Index()
        {
            var query = db.NhanViens.Select(p => p);
            return View(query);
        }

        public ActionResult ChiTiet(string id)
        {
            var query = db.NhanViens.Where(m => m.Manv == id).First();
            return View(query);
        }

        [HttpGet]
        public ActionResult Them()
        {
            return View();
        }

        [HttpPost]
        public ActionResult Them(FormCollection data, NhanVien nv)
        {
            //Gán giá trị người dùng nhập cho các biến
            var ma= data["Manv"];
            var ten = data["Hoten"];
            var phong = data["Phong"];
            var luong =data["Luong"];
            if (String.IsNullOrEmpty(ma))
            { ViewData["Loi1"] = "Mã nhân viên không được để trống!"; }
            else if (String.IsNullOrEmpty(ten))
            { ViewData["Loi2"] = "Họ tên không được để trống!"; }
            else if (String.IsNullOrEmpty(phong))
            { ViewData["Loi3"] = "Phòng không được để trống!"; }
            else if (String.IsNullOrEmpty(luong))
            { ViewData["Loi4"] = "Lương không được để trống!"; }
            else
            {
                //Gán giá trị cho đối tượng được tạo mới (nv)
                nv.Manv =ma;
                nv.Hoten = ten;
                nv.Phong = phong;
                nv.Luong = Convert.ToDouble(luong);
                db.NhanViens.Add(nv);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            return this.Them();
        }

        [HttpGet]
        public ActionResult Xoa(string id)
        {
            var query = db.NhanViens.First(m => m.Manv == id);
            return View(query);
        }

        [HttpPost]
        public ActionResult Xoa(string id,FormCollection Data)
        {
            var query = db.NhanViens.Where(m => m.Manv == id).First();
            db.NhanViens.Remove(query);
            db.SaveChanges();
            return RedirectToAction("Index");
        }
    }
}

```

```

[HttpGet]
public ActionResult Sua(string id)
{
    var query = db.NhanViens.First(m => m.Manv == id);
    return View(query);
}
[HttpPost]
public ActionResult Sua(string id, FormCollection data)
{
    var nv = db.NhanViens.First(m => m.Manv == id);
    //Gán giá trị người dùng nhập sửa cho các biến
    var ten = data["Hoten"];
    var phong = data["Phong"];
    var luong = data["Luong"];
    if (String.IsNullOrEmpty(ten))
    { ViewData["Loi1"] = "Họ tên không được để trống!"; }
    else if (String.IsNullOrEmpty(phong))
    { ViewData["Loi2"] = "Phòng không được để trống!"; }
    else if (String.IsNullOrEmpty(luong))
    { ViewData["Loi3"] = "Lương không được để trống!"; }
    else
    { //Gán giá trị cho đối tượng được tạo mới (nv)
        nv.Hoten = ten;
        nv.Phong = phong;
        nv.Luong = Convert.ToDouble(luong);
        UpdateModel(nv);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return this.Sua(id);
}
}
}
}

```

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

• View: ChiTiet.cshtml

```

@{
    ViewBag.Title = "ChiTiet";
}
<h2>XEM CHI TIẾT</h2>
<fieldset>
    <div>Mã nhân viên: @Model.Manv</div>
    <div>Họ tên: @Model.Hoten</div>
    <div>Tên phòng: @Model.Phong</div>
    <div>Lương: @Model.Luong</div>
</fieldset> <br/>
@using(Html.BeginForm())
{
    <p>
        @Html.ActionLink("Sửa", "Sua", new { id = Model.Manv.Trim() })|
        @Html.ActionLink("Xóa", "Xoa", new { id = Model.Manv.Trim() })|
        @Html.ActionLink("Về danh sách", "Index")
    </p>
}

```

✓ Chạy View Index, click vào Link [Chi tiết](#) xem kết quả.

• View: Them.cshtml

```
@{
    ViewBag.Title = "Them";
}
<h2>THÊM NHÂN VIÊN</h2>
<table>
    @using (Html.BeginForm())
    {
        @Html.ValidationSummary(true)//Bắt lỗi
        <tr style="height:40px">
            <td>Mã nhân viên: </td>
            <td>@Html.TextBox("Manv")<font color="red"> @ViewData["Loi1"] </font></td>
        </tr>
        <tr style="height:40px">
            <td>Họ tên: </td>
            <td>@Html.TextBox("Hoten")<font color="red"> @ViewData["Loi2"] </font></td>
        </tr>
        <tr style="height:40px">
            <td>Tên phòng: </td>
            <td>@Html.TextBox("Phong")<font color="red"> @ViewData["Loi3"] </font></td>
        </tr>
        <tr style="height:40px">
            <td>Lương: </td>
            <td>@Html.TextBox("Luong")<font color="red"> @ViewData["Loi4"] </font></td>
        </tr>
        <tr style="height:40px">
            <td colspan="2" style="text-align:center">
                <input type="submit" value="Thêm" /> |
                @Html.ActionLink("Về danh sách", "Index")
            </td>
        </tr>
    }
</table>
```

✓ Chạy View Index, click vào Link [Thêm](#) xem kết quả.

• View: Sua.cshtml

```
@{
    ViewBag.Title = "Sua";
}
<h2>SỬA NHÂN VIÊN</h2>
<table>
    @using (Html.BeginForm())
    {
        @Html.ValidationSummary(true)//Bắt lỗi
        <tr style="height:40px">
            <td>Họ tên: </td>
            <td>@Html.TextBox("Hoten")<font color="red"> @ViewData["Loi1"] </font></td>
        </tr>
        <tr style="height:40px">
            <td>Tên phòng: </td>
            <td>@Html.TextBox("Phong")<font color="red"> @ViewData["Loi2"] </font></td>
        </tr>
        <tr style="height:40px">
            <td>Lương: </td>
            <td>@Html.TextBox("Luong")<font color="red"> @ViewData["Loi3"] </font></td>
        </tr>
        <tr style="height:40px">
            <td colspan="2" style="text-align:center">
                <input type="submit" value="Lưu" /> |
                @Html.ActionLink("Về danh sách", "Index")
            </td>
        </tr>
    }
</table>
```

- ✓ Chạy View Index, click vào Link [Sửa](#) xem kết quả.

•View: Xoa.cshtml

```
@{
    ViewBag.Title = "Xoa";
}
<h2>XÓA NHÂN VIÊN</h2>
<fieldset>
    <div>Mã nhân viên:@Model.Manv</div>
    <div>Họ tên: @Model.Hoten</div>
    <div>Tên phòng: @Model.Phong</div>
    <div>Lương: @Model.Luong</div>
</fieldset> <br/>
@using (Html.BeginForm())
{
    <p>
        <input type="submit" value="Xóa"/> |
        @Html.ActionLink("Về danh sách","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("Xem nhân viên", "Index", "Nhanvien")</li>
    </ul>
</div>
```

- ✓ F5 chạy, xem kết quả