

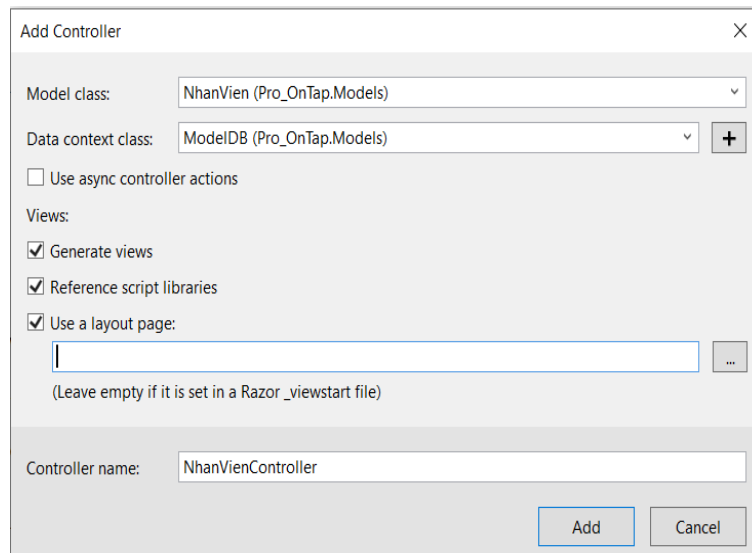
BÀI ÔN TẬP

Chuẩn bị:

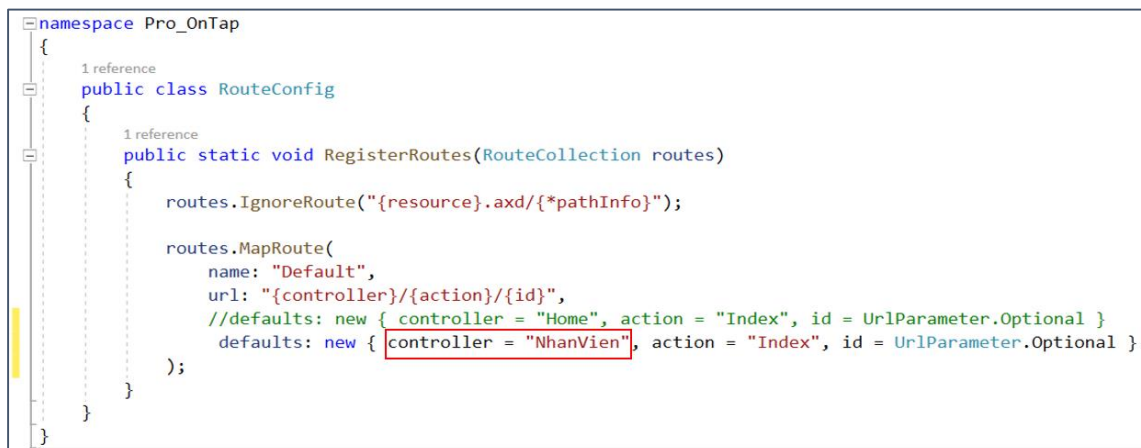
- Sử dụng SQL Server, chạy script đã cho để tạo csdl **QLNV** gồm 3 bảng:
Phong(**Maphong**, Tenphong)
NhanVien(**Manv**, Hoten, Tuoi, Diachi, Luong, Maphong)
tblUser(**id**, username, password, role)

Câu 1: Tạo một Controller mới từ **MVC5 Controller with view, using Entity Framework**, đặt tên là **NhanvienController** (gắn với bảng **NhanVien**)

- Tạo một project đặt tên là **Pro_OnTap**, chọn mẫu **MVC**.
- Cài đặt **EntityFramework** sử dụng **NuGet Package Manager**
- Sử dụng EF code first để kết nối với CSDL **QLNV** (đặt tên ADO.NET Entity Model là **ModelDB**) 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)
- Tạo một controller theo mẫu **MVC 5 Controller with view, using EntityFramework** gắn với model **NhanVien**.



Câu 2: Đặt phương thức ActionResult **Index** đã có ở trên là mặc định khi chạy Project.



Câu 3: Sửa mã ở controller **NhanVienController** và view để thêm nhân viên sử dụng công nghệ **Ajax**, cập nhập một phần trang web: Khi click vào button “Save” sẽ **hiển thị thông báo** thêm thành công hay không ngay ở phía dưới form nhập dữ liệu.

(Thông thường trang Web sẽ được nạp lại toàn bộ mỗi khi request. Công nghệ **ajax** cho phép cập nhật 1 phần trang Web.)

+ Sửa View NhanVien/Create.cshtml

```
@model Pro_OnTap.Models.NhanVien
@{
    ViewBag.Title = "Create";
}
<h2>Create</h2>

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>NhanVien</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.Hoten, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Hoten, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Hoten, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.Tuoi, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Tuoi, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Tuoi, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.Diachi, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Diachi, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Diachi, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.Luong, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Luong, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Luong, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.Maphong, "Maphong", htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownList("Maphong", null, htmlAttributes: new { @class = "form-control" })
                @Html.ValidationMessageFor(model => model.Maphong, "", new { @class = "text-danger" })
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                @*<input type="submit" value="Create" class="btn btn-default" />*@
                <input class="btn btn-primary" name="submitButton" id="btnSave" value="Save" type="button" />
            </div>
        </div>
    </div>
}
<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@*Thêm id msg để thông báo việc thêm dữ liệu có thành công hay không *@
<div id="message" style="color:red"> </div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")*}

<script type="text/javascript">
$(function () {
    $("#btnSave").click(function () {
        var nv = {};
        @*Tạo các thuộc tính và gán giá trị. Tên các thuộc tính này phải trùng với tên các thuộc tính của lớp NhanVien*@
        nv.Hoten = $("#Hoten").val();
        nv.Tuoi = $("#Tuoi").val();
        nv.Diachi = $("#Diachi").val();
        nv.Luong = $("#Luong").val();
        nv.Maphong = $("#Maphong").val();
    });
});
</script>
```

```

$.ajax({
    type: "POST",
    url: '@Url.Action("/Create")',
    data: '{nv: ' + JSON.stringify(nv) + '}',
    dataType: "json",
    contentType: "application/json; charset=utf-8",
    success: function (response) {
        if (response.result == true) {

            $("#message").html('Add data successfully.');
        }
        else {
            $("#message").html(response.error);
        }
    },
    error: function (response) {
        alert("Error in ajax block");
    }
});
return false;
});
});
</script>

```

```

}

```

+ Sủa Controller **NhanvienController**

```

using Pro_OnTap.Models;

namespace Pro_OnTap.Controllers
{
    public class NhanVienController : Controller
    {
        private ModelDB db = new ModelDB();

        // GET: NhanVien
        public ActionResult Index()
        {
            var nhanViens = db.NhanViens.Include(n => n.Phong);
            return View(nhanViens.ToList());
        }

        // GET: NhanVien/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            NhanVien nhanVien = db.NhanViens.Find(id);
            if (nhanVien == null)
            {
                return HttpNotFound();
            }
            return View(nhanVien);
        }

        // GET: NhanVien/Create
        public ActionResult Create()
        {
            ViewBag.Maphong = new SelectList(db.Phongs, "Maphong", "Tenphong");
            return View();
        }

        // POST: NhanVien/Create
        // To protect from overposting attacks, enable the specific properties you want to bind to, for
        // more details see https://go.microsoft.com/fwlink/?LinkId=317598.

        // Code Create gốc
        //[HttpPost]
        //[ValidateAntiForgeryToken]
        //public ActionResult Create([Bind(Include = "Manv,Hoten,Tuoi,Diachi,Luong,Maphong")] NhanVien nhanVien)
        //{
        //    if (ModelState.IsValid)
        //    {

```

```
//      db.NhanViens.Add(nhanVien);
//      db.SaveChanges();
//      return RedirectToAction("Index");
//  }

//  ViewBag.Maphong = new SelectList(db.Phongs, "Maphong", "Tenphong", nhanVien.Maphong);
//  return View(nhanVien);
//}
```

```
//Code Create mới
[HttpPost]
public ActionResult Create(NhanVien nv)
{
    try
    {
        db.NhanViens.Add(nv);
        db.SaveChanges();
        //Trả về xâu Json nếu result = true
        return Json(new { result = true, JsonRequestBehavior.AllowGet });
    }
    catch (Exception ex)
    { //Trả về xâu Json nếu result =false, vì có lỗi ko thêm được
        return Json(new { result = false, error = ex.Message });
    }
}
```

```
// GET: NhanVien/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    NhanVien nhanVien = db.NhanViens.Find(id);
    if (nhanVien == null)
    {
        return HttpNotFound();
    }
    ViewBag.Maphong = new SelectList(db.Phongs, "Maphong", "Tenphong", nhanVien.Maphong);
    return View(nhanVien);
}

// POST: NhanVien/Edit/5
// To protect from overposting attacks, enable the specific properties you want to bind to, for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "Manv,Hoten,Tuoi,Diachi,Luong,Maphong")] NhanVien nhanVien)
{
    if (ModelState.IsValid)
    {
        db.Entry(nhanVien).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.Maphong = new SelectList(db.Phongs, "Maphong", "Tenphong", nhanVien.Maphong);
    return View(nhanVien);
}

// GET: NhanVien/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    NhanVien nhanVien = db.NhanViens.Find(id);
    if (nhanVien == null)
    {
        return HttpNotFound();
    }
    return View(nhanVien);
}

// POST: NhanVien/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    NhanVien nhanVien = db.NhanViens.Find(id);
```

```

        db.NhanViens.Remove(nhanVien);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}
}

```

Chạy chương trình, thêm 1 nhân viên, ta thấy trang Web không tải lại mà chỉ cập nhật DL cho **id message**

Câu 4: Sử dụng **NhanVienController**, viết các action method, **partialview** và sửa file **_Layout.cshtml** sao cho khi chạy project, trên menu chỉ hiển thị các tên phòng lấy từ bảng **Phong** và mục **Index**; trong đó Index link tới xem toàn bộ danh sách nhân viên ở câu 1 và tên phòng sẽ link tới action ở câu 5 phía dưới.

```

using Pro_OnTap.Models;

namespace Pro_OnTap.Controllers
{
    public class NhanVienController : Controller
    {
        private ModelDB db = new ModelDB();

        // GET: NhanVien
        public ActionResult Index()
        {
            var nhanViens = db.NhanViens.Include(n => n.Phong);
            return View(nhanViens.ToList());
        }

        [ChildActionOnly]
        public PartialViewResult CategoryMenu()
        {
            var li = db.Phongs.ToList();
            return PartialView(li);
        }
    }
}

```

...

Chạy chương trình:

Tenphong	Hoten	Tuoi	Diachi	Luong	
Hành chính	Lan Anh	32	Ha noi	3500	Edit Details Delete
Kế toán	Thi Huong	30	Ha noi	5000	Edit Details Delete
Hành chính	Van Ha	30	Hai phong	4000	Edit Details Delete
Tổng hợp	Hong Linh	35	Thai binh	5500	Edit Details Delete
Hành chính	Van Hung	32	Hai phong	5000	Edit Details Delete
Hành chính	Hai Yen	28	Thai binh	3000	Edit Details Delete
Kế toán	My Ha	26	Hai phong	3000	Edit Details Delete
Tổng hợp	Phuong Thuy	24	Hai phong	2000	Edit Details Delete

Câu 5: Tiếp tục sử dụng **NhanVienController**, tạo các action và view để khi click vào một phòng ban trên menu, danh sách nhân viên thuộc về phòng đó hiển thị ra màn hình.

+ Thêm phương thức hiển thị danh sách nhân viên theo phòng ban:

using Pro_OnTap.Models;

namespace Pro_OnTap.Controllers

```
{
    public class NhanVienController : Controller
    {
        private ModelDB db = new ModelDB();

        // GET: NhanVien
        public ActionResult Index()
        {
            var nhanViens = db.NhanViens.Include(n => n.Phong);
            return View(nhanViens.ToList());
        }

        [ChildActionOnly]
        public PartialViewResult CategoryMenu()
        {
            var li = db.Phongs.ToList();
            return PartialView(li);
        }

        [Route("NhanVienTheoPhong/{Maphong}")] // Định tuyến cho phương thức
        public ActionResult HienThiTheoPhong(int Maphong)
        {
            var li = db.NhanViens.Where(e => e.Maphong == Maphong).ToList();
            return View(li);
        }
    }
}
```

...

+ Tạo View hiển thị DS nhân viên theo phòng ban: **HienThiTheoPhong.cshtml**

Câu 6: Tạo controller mới tên là **LoginController**. Tạo các action và view cần thiết cho phép người dùng đăng nhập vào hệ thống. Sử dụng bảng **tblUser** đã cho trong CSDL để kiểm tra người dùng có nhập đúng **username** và **password** hay không.

- Nếu nhập đúng thì hiển thị lời chào **Welcome + username** trên menu và xuất hiện link **Logout** bên cạnh lời chào, đồng thời chuyển đến trang hiển thị toàn bộ danh sách ở câu 1. Nếu đăng nhập sai thì thông báo lỗi.
- Khi người dùng click vào link **Logout** thì đăng xuất khỏi hệ thống, trên menu không còn link **Logout** mà thay vào đó là link **Login**.

+ Tạo Controller **LoginController**

```
using Pro_OnTap.Models;

namespace Pro_OnTap.Controllers
{
    public class LoginController : Controller
    {
        // GET: Login
        private ModelDB db = new ModelDB();
        public ActionResult Index()
        {
            return View();
        }
        public ActionResult Login()
        {
            return View();
        }
        [HttpPost]
        public ActionResult Login(string username, string password)
        {
            var user = db.tblUsers.Where(u => u.username == username &&
                                           u.password == password).FirstOrDefault();
            if (user == null)
            {
                ViewBag.errLogin = "Sai tên đăng nhập hoặc mật khẩu";
                return View("Login");
            }
            else
            {
                Session["username"] = username;
                return RedirectToAction("Index", "NhanVien"); //chạy tới action Index
            }
        }
        public ActionResult Logout()
        {
            Session["username"] = null;
            return RedirectToAction("Index", "NhanVien");
        }
    }
}
```

+ Tạo View **Login.cshtml**

```
@{
    ViewBag.Title = "Login";
}
<h2>Login</h2>
<div class="login">

    <div class="input">
        @using (Html.BeginForm()) //ko cần viết rõ vì cùng tên action method Login
        {
            <p><label>Username:</label> <input type="text" name="username" /></p> <br />
            <p><label>Password:</label><input type="text" name="password" /> </p> <br />
            <p><input class="lg" type="submit" value="Login" /></p>
        }
    </div>
    @ViewBag.errLogin
</div>
```

+ Sura file **_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</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
</head>
<body>
    <div>
        <center>
            @*gọi phương thức hiển thị DS nhân viên*@
            @Html.ActionLink("Index", "Index", "NhanVien")<label>&nbsp;&nbsp;&nbsp;&nbsp;&|&nbsp;&nbsp;&nbsp;&nbsp;&</label>
            @*gọi phương thức hiển thị PartialView*@
            @{Html.RenderAction("CategoryMenu", "NhanVien");}



@if (Session["username"] == null)
                {
                    <a href="/Login/Login">Login</a>
                }
                else
                {
                    <label> Welcome&nbsp;&nbsp;&nbsp;&nbsp;&</label><b>@Session["username"]</b><label>&nbsp;&nbsp;&nbsp;&nbsp;&|&nbsp;&nbsp;&nbsp;&nbsp;&</label>
                    <a href="/Login/Logout">Logout</a>
                }
            }
            </center>
            <hr />
            <div class="container body-content">
                @RenderBody()
                <hr />
                <footer>
                    <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
                </footer>
            </div>
        </div>
        @Scripts.Render("~/bundles/jquery")
        @Scripts.Render("~/bundles/bootstrap")
        @RenderSection("scripts", required: false)
    </body>
</html>


```

Chạy chương trình, Click vào Link **Login** trên menu, Test các trường hợp đúng/sai

[Index](#) | [Hanh chinh](#) | [Ke toan](#) | [Tong hop](#) | [Login](#)

Login

Username:

Password:

Login

Index

[Create New](#)

deptname	name	age	addr	salary	image	
Hanh chinh	Lan Anh	32	Ha noi	3500	a01.jpg	Edit Details Delete
Ke toan	Thi Huong	30	Ha noi	5000	a02.jpg	Edit Details Delete
Hanh chinh	Van Ha	30	Hai phong	4000	a03.jpg	Edit Details Delete
Tong hop	Hong Linh	35	Thai binh	5500	a04.jpg	Edit Details Delete