座號：19 姓名：柯昱廷

階段性作業七

後台管理-登入、資料CRUD實做練習

* 承前階段性作業六所完成的Model，完成下列要求：
  1. Layout頁面：
     1. 建立名為\_Layout\_Manager的Layout頁面給後台管理者。
     2. 管理者登入後功能選單包含「客戶管理」、「商品管理」、「登出」。
     3. 選單列須以不同顏色區分角色。
  2. 管理者登入：
     1. 建立後台管理者登入功能，其帳號與密碼為Employees資料表中的Account及Password。
     2. 登入失敗顯示提示詞，成功則進入後台管理頁面。
     3. 登入成功後頁面須有提示字顯示目前是管理員角色。
     4. 在選單按下登出鈕後則返回最初登入畫面。
  3. 會員（Members）資料管理：
     1. 檢視客戶清單頁（Master Page）—列出會員資料，含「MemberID」、「MemberName」、「MemberBirthday」、「CreatedDate」欄位。
     2. 在檢視顧客清單建立分頁功能，每15筆資料為一頁。
     3. 檢視客戶詳細資料功能—點選指定客戶資料，可查閱客戶之詳細資料（所有欄位）。
     4. 新增會員資料功能—可建立新會員資料，須檢查會員編號不可重複，按下按鈕可寫回資料庫更新資料。
     5. 修改會員資料功能—點選指定會員資料，可編輯會員資料（MemberID、Account、Password、MemberPhotoFile、CreatedDate不可修改），按下按鈕可寫回資料庫更新資料。
  4. 畫面編排樣式不限，但安排與流程需合乎一般使用習慣與原則。

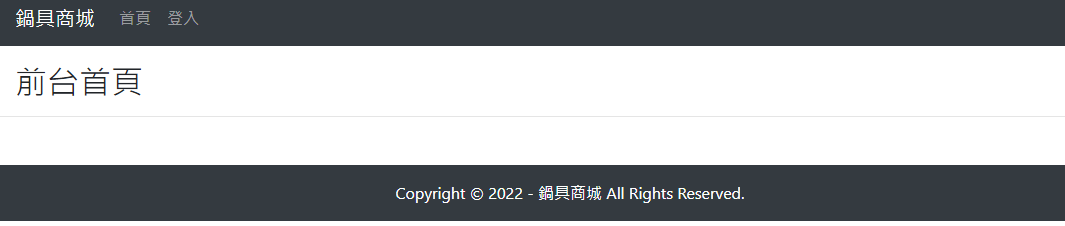
**繳交作業時，請以FTP上傳程式原始檔與本文件檔各一份，原始程式碼請將整個方案壓縮成一個檔，壓縮檔案名稱為「MCSDD○○」，其中○○為自己的座號，並放置於「完成結果檔」資料夾內。**

* ※以上各項均須完成始得通過※

執行畫面：

Layout選單(至少包含登入前與登入後)

登入前



登入後



管理者登入功能(至少包含成功與失敗)

登入成功

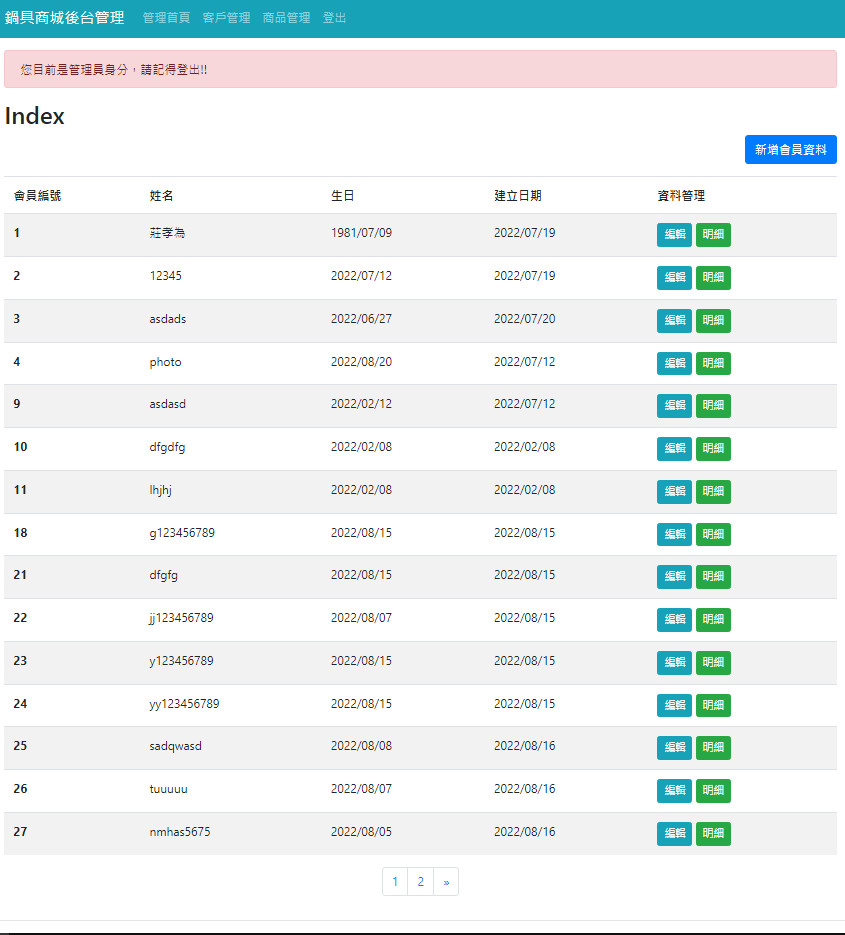


登入失敗

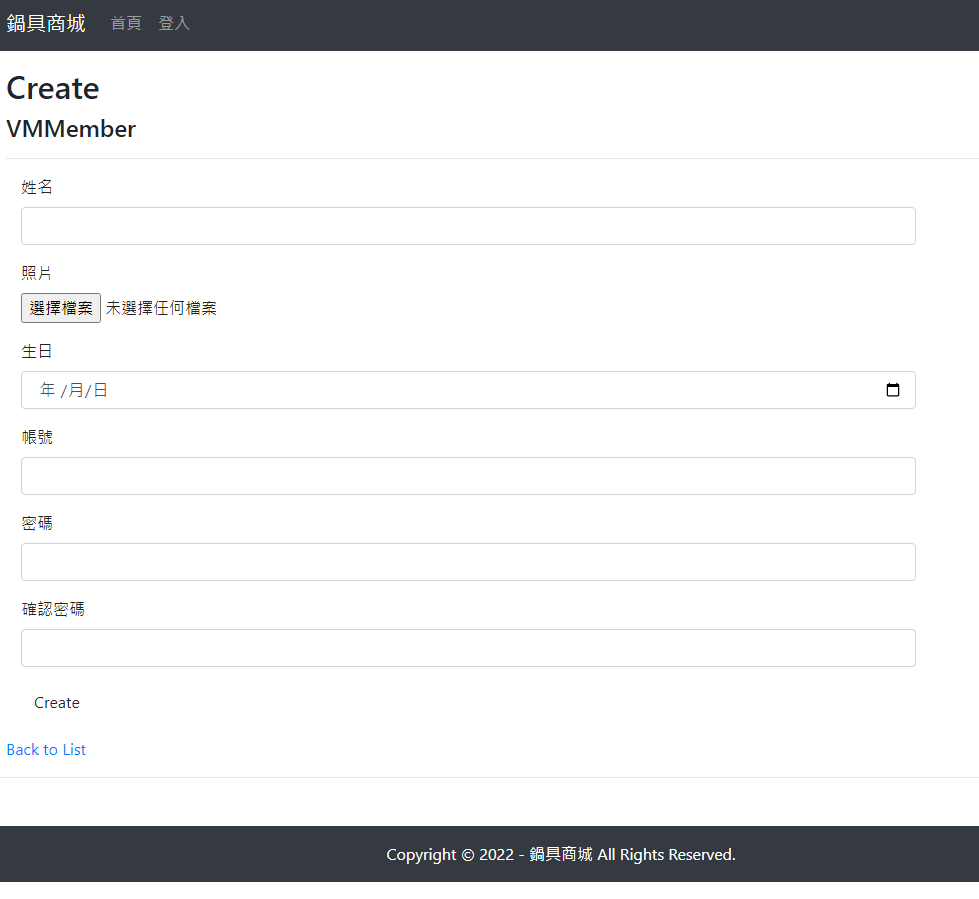


會員資料管理功能(至少包含清單檢視、新增、編輯、檢視詳細資料、分頁五個畫面)

Index



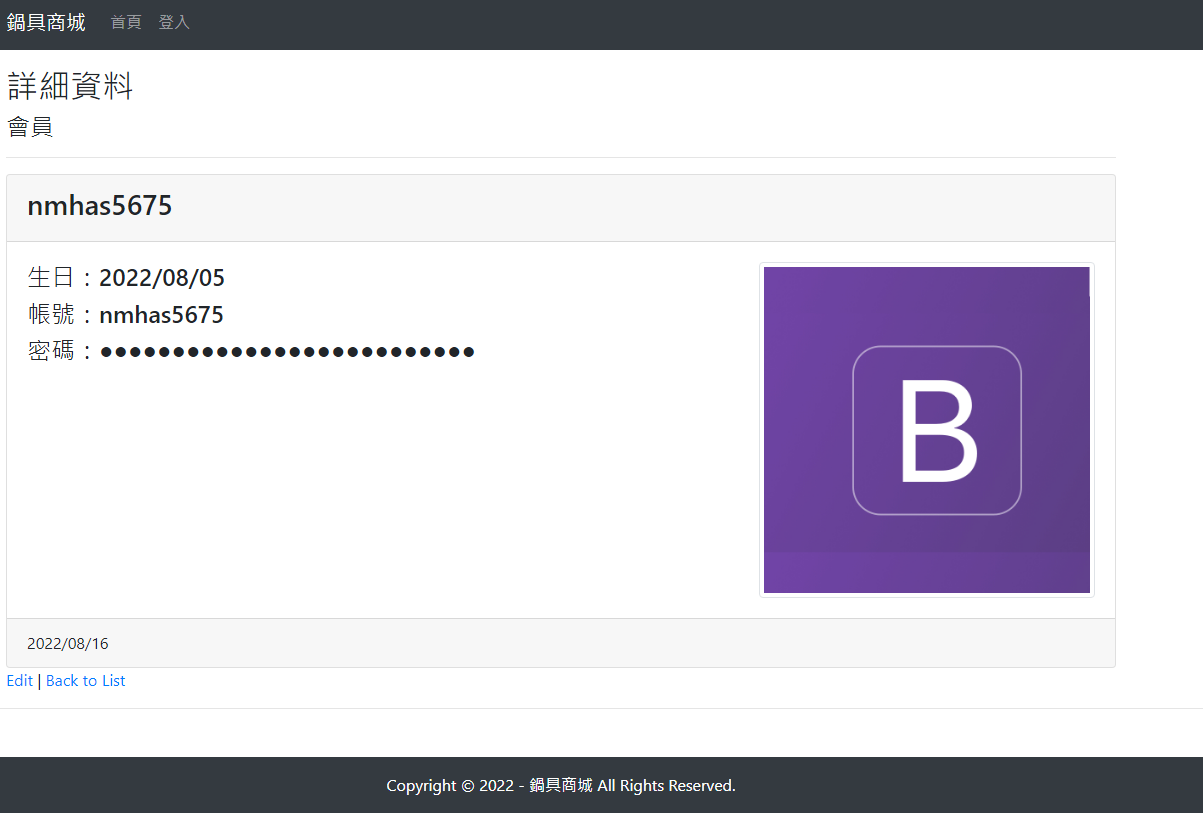
Create



Edit



Details



分頁



程式碼(所有的程式碼均須貼上)：

Controller (請標示Controller名稱)

### EmployeesController

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

using HW7Project.Models;

namespace HW7Project.Controllers

{

public class EmployeesController : Controller

{

private HW7ProjectContext db = new HW7ProjectContext();

// GET: Employees

public ActionResult Index()

{

return View(db.Employees.ToList());

}

// GET: Employees/Details/5

public ActionResult Details(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Employees employees = db.Employees.Find(id);

if (employees == null)

{

return HttpNotFound();

}

return View(employees);

}

// GET: Employees/Create

public ActionResult Create()

{

return View();

}

// POST: Employees/Create

// 若要避免過量張貼攻擊，請啟用您要繫結的特定屬性。

// 如需詳細資料，請參閱 https://go.microsoft.com/fwlink/?LinkId=317598。

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "EmployeeID,EmployeeName,CreatedDate,Account,Password")] Employees employees)

{

if (ModelState.IsValid)

{

db.Employees.Add(employees);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(employees);

}

// GET: Employees/Edit/5

public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Employees employees = db.Employees.Find(id);

if (employees == null)

{

return HttpNotFound();

}

return View(employees);

}

// POST: Employees/Edit/5

// 若要避免過量張貼攻擊，請啟用您要繫結的特定屬性。

// 如需詳細資料，請參閱 https://go.microsoft.com/fwlink/?LinkId=317598。

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit([Bind(Include = "EmployeeID,EmployeeName,CreatedDate,Account,Password")] Employees employees)

{

if (ModelState.IsValid)

{

db.Entry(employees).State = EntityState.Modified;

db.SaveChanges();

return RedirectToAction("Index");

}

return View(employees);

}

// GET: Employees/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Employees employees = db.Employees.Find(id);

if (employees == null)

{

return HttpNotFound();

}

return View(employees);

}

// POST: Employees/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Employees employees = db.Employees.Find(id);

db.Employees.Remove(employees);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}

}

#### HomeController

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace HW7Project.Controllers

{

public class HomeController : Controller

{

// GET: Home

public ActionResult Index()

{

return View();

}

}

}

#### HomeManagerController

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using HW7Project.Models;

namespace HW7Project.Controllers

{

public class HomeManagerController : Controller

{

HW7ProjectContext db = new HW7ProjectContext();

// GET: HomeManager

//\*\*\*

[LoginCheck]

public ActionResult Index()

{

//if (Session["user"] == null)

//{ return RedirectToAction("Index", "Home"); }

//return View();

//這樣子寫需要每一個Action都要檢查

//比較好的方法是:在Controller建立LoginCheck的類別

return View();

}

public ActionResult Login()

{

return View();

}

[HttpPost]

public ActionResult Login(VMLogin vMLogin)

{

//select \* from Employee where account and password=@password

var user = db.Employees.Where(m => m.Account == vMLogin.Account && m.Password == vMLogin.Password).FirstOrDefault();

if (user == null)

{

ViewBag.ErrMsg = "帳號或密碼有錯!!";

return View(vMLogin);

}

Session["user"] = user;

return RedirectToAction("Index");

}

[LoginCheck]

public ActionResult Logout()

{

Session["user"] = null;

//登出後回到前台首頁(HOME)

return RedirectToAction("Index","Home");

}

}

}

#### LoginCheck

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace HW7Project.Controllers

{

public class LoginCheck:ActionFilterAttribute

{

void LoginState(HttpContext context) //傳入HttpContext context這樣才能用Session

{

if (context.Session["user"] == null)

{

context.Response.Redirect("/Home/Index");

}

}

public override void OnActionExecuting(ActionExecutingContext filterContext)

{

HttpContext context = HttpContext.Current;

LoginState(context);

}

//再ACTION被執行時就使用FILTER

}

}

#### LogReporter

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Web;

using System.Web.Mvc;

using System.Web.Routing;

using HW7Project.Models;

namespace HW7Project.Controllers

{

public class LogReporter:ActionFilterAttribute

{

HttpContext context;

void LogValues(RouteData routeData, HttpContext context)

{

var logTime = DateTime.Now;

var controllerName = routeData.Values["controller"];

var actionName = routeData.Values["action"];

var parameter = routeData.Values["id"] == null ? "N/A" : routeData.Values["id"];

var user = context.Session["user"] == null ? "Guest" : ((Employees)context.Session["user"]).EmployeeID.ToString();

StreamWriter sw = new StreamWriter(context.Server.MapPath("\\ValueLog.csv"), true, Encoding.Default); //路徑,是不是append,編碼方法

sw.WriteLine(logTime + "," + controllerName + "," + actionName + "," + parameter + ", " + user);

sw.Close();

}

void RequestLog(HttpContext context)

{

var ip = context.Request.ServerVariables["REMOTE\_ADDR"];

var host = context.Request.ServerVariables["REMOTE\_HOST"];

var browser = context.Request.ServerVariables["HTTP\_USER\_AGENT"];

var requestType = context.Request.RequestType;

var userHostAddress = context.Request.UserHostAddress;

var userHostName = context.Request.UserHostName;

var httpMethod = context.Request.HttpMethod;

var userAgent = context.Request.UserAgent;

var requestTime = DateTime.Now;

StreamWriter sw = new StreamWriter(context.Server.MapPath("\\RequestLog.txt"), true, Encoding.Default);

sw.WriteLine(requestTime + ", " + ip + ", " + host + ", " + browser + ", " + requestType + ", "

+ userHostAddress + ", " + userHostName + ", " + httpMethod + ", " + userAgent);

sw.Close();

}

public override void OnActionExecuting(ActionExecutingContext filterContext)

{

context = HttpContext.Current; // HttpContext.Current取得HttpContext

LogValues(filterContext.RouteData, context);

}

public override void OnResultExecuted(ResultExecutedContext filterContext)

{

context = HttpContext.Current;

RequestLog(context);

}

}

}

#### MembersController

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data;

using System.Data.Entity;

using System.Data.SqlClient;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

using HW7Project.Models;

using PagedList; //做分頁的套件

namespace HW7Project.Controllers

{

//整個Controller都Check

//[LoginCheck]

public class MembersController : Controller

{

private HW7ProjectContext db = new HW7ProjectContext();

// GET: Members

public ActionResult Index(int page=1)

{

//分頁功能

var members = db.Members.ToList();

int pagesize = 15;

var pagedList = members.ToPagedList(page, pagesize); //ToPagedList(目前在第幾頁，每次顯示幾筆)

return View(pagedList);

}

public ActionResult IndexModal()

{

return View(db.Members.ToList());

}

// GET: Members/Details/5

public ActionResult Details(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Members members = db.Members.Find(id);

if (members == null)

{

return HttpNotFound();

}

//\*\*\*若Detail使用Modal，就能回傳PartialView部分檢視(沒有Layout)

return View(members);

}

// GET: Members/Create

public ActionResult Create()

{

return View();

}

// POST: Members/Create

// 若要避免過量張貼攻擊，請啟用您要繫結的特定屬性。

// 如需詳細資料，請參閱 https://go.microsoft.com/fwlink/?LinkId=317598。

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create([Bind(Include = "MemberID,MemberName,MemberPhotoFile,MemberBirdthday,CreatedDate,Account,Password")] Members members, HttpPostedFileBase photo)

{

if (photo != null)

{

if(photo.ContentLength>0)

{

string extensionName = System.IO.Path.GetExtension(photo.FileName); //抓副檔名

if(extensionName == ".jpg"|| extensionName == ".png")

{

photo.SaveAs(Server.MapPath("~/MemberPhotos/" + members.Account + extensionName));

members.MemberPhotoFile = members.Account + extensionName;

}

}

}

//var account = db.Members.Where(m => m.Account == members.Account).FirstOrDefault();

//if (account != null)

//{

// ViewBag.Error = "已經有相同帳號";

// return View();

//}

if (ModelState.IsValid)

{

db.Members.Add(members);

db.SaveChanges();

return RedirectToAction("Index");

}

return View(); //\*\*\*為甚麼沒有return View(members)，表單不會被清空

}

// GET: Members/Edit/5

public ActionResult Edit(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Members members = db.Members.Find(id);

if (members == null)

{

return HttpNotFound();

}

return View(members); //需要帶值至View，所以View頁面不能用VM

}

// POST: Members/Edit/5

// 若要避免過量張貼攻擊，請啟用您要繫結的特定屬性。

// 如需詳細資料，請參閱 https://go.microsoft.com/fwlink/?LinkId=317598。

[HttpPost]

[ValidateAntiForgeryToken]

//Bind正向表列

public ActionResult Edit(Members members)

{

//利用ADO.NET的寫法

string sql = "update members set MemberName=@MemberName,MemberBirdthday=@MemberBirdthday where MemberID=@MemberID";

SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["HW7ProjectConnection"].ConnectionString); //裡面的參數是Web.config的連線資訊

SqlCommand cmd = new SqlCommand(sql, conn);

cmd.Parameters.AddWithValue("@MemberName", members.MemberName);

cmd.Parameters.AddWithValue("@MemberBirdthday", members.MemberBirdthday);

cmd.Parameters.AddWithValue("@MemberID", members.MemberID);

conn.Open();

try

{

cmd.ExecuteNonQuery();

return RedirectToAction("Index");

}

catch (Exception ex)

{

ViewBag.Error = ex.Message;

}

conn.Close();

return View(members);

//if (ModelState.IsValid)

//{

// db.Entry(members).State = EntityState.Modified;

// db.SaveChanges();

// return RedirectToAction("Index");

//}

}

// GET: Members/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Members members = db.Members.Find(id);

if (members == null)

{

return HttpNotFound();

}

return View(members);

}

// POST: Members/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Members members = db.Members.Find(id);

db.Members.Remove(members);

db.SaveChanges();

return RedirectToAction("Index");

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

}

}

View (請標示View名稱)

#### Home/index

@{

ViewBag.Title = "Index";

}

<h2>前台首頁</h2>

#### HomeManager/Index

@{

ViewBag.Title = "Index";

Layout = "~/Views/Shared/\_Layout\_Manager.cshtml";

}

<h2>後台首頁</h2>

#### HomeManager/Login

@model HW7Project.Models.VMLogin

@{

ViewBag.Title = "Login";

}

<h2>Login</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>VMLogin</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.Account, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Account, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Account, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.Password, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Password, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Password, "", 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" />

</div>

</div>

<div class="col-md">

<p class="text-danger">@ViewBag.ErrMsg</p>

</div>

</div>

}

<div>

@Html.ActionLink("Back to List", "Index")

</div>

@\*這裡有個區塊我把它取名為Scripts\*@

@section scripts{

<script src="~/Scripts/jquery.validate.min.js"></script>

<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

}

#### Members/Create

@model HW7Project.Models.VMMember

@{

ViewBag.Title = "Create";

var CreatedDate = DateTime.Today.ToShortDateString(); //ToShortDateString只抓日期，不抓時間

}

<h2>Create</h2>

@using (Html.BeginForm("Create", "Members", FormMethod.Post, new { enctype = "multipart/form-data" })) //表單必須含有enctype = "multipart/form-data"才能傳檔案

{

@Html.AntiForgeryToken()

<input id="CreatedDate" type="hidden" name="CreatedDate" value="@CreatedDate" />

<div class="form-horizontal">

<h4>VMMember</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.MemberName, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.MemberName, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.MemberName, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.MemberPhotoFile, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@\*@Html.EditorFor(model => model.MemberPhotoFile, new { type="file",@class="form-control",name="photo"})\*@

<input type="file" name="photo" id="photo" class="from-control" />

@Html.ValidationMessageFor(model => model.MemberPhotoFile, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.MemberBirdthday, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.MemberBirdthday, new { htmlAttributes = new { @class = "form-control"} })

@Html.ValidationMessageFor(model => model.MemberBirdthday, "", new { @class = "text-danger" })

</div>

</div>

@\*<div class="form-group">

@Html.LabelFor(model => model.CreatedDate, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.CreatedDate, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.CreatedDate, "", new { @class = "text-danger" })

</div>

</div>\*@

<div class="form-group">

@Html.LabelFor(model => model.Account, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Account, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Account, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.Password, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Password, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Password, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.ConfirmPassword, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.ConfirmPassword, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.ConfirmPassword, "", 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" />

</div>

</div>

</div>

}

<div>

@Html.ActionLink("Back to List", "Index")

</div>

<script src="~/Scripts/jquery-3.4.1.min.js"></script>

<script src="~/Scripts/jquery.validate.min.js"></script>

<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

#### Members/Details

@model HW7Project.Models.Members

@{

ViewBag.Title = "Details";

}

<h2>詳細資料</h2>

<div>

<h4>會員</h4>

<hr />

<div class="card">

<div class="card-header">

<h3>@Html.DisplayFor(model => model.MemberName)</h3>

</div>

<div class="card-body">

<div class="row">

<div class="col-md-8">

<h4>生日：@Html.DisplayFor(model => model.MemberBirdthday)</h4>

<h4> 帳號：@Html.DisplayFor(model => model.Account)</h4>

<h4>密碼：●●●●●●●●●●●●●●●●●●●●●●●●●●</h4>

</div>

<div class="col-md-4">

@if (Model.MemberPhotoFile == null)

{

<img src="~/MemberPhotos/defaultPhoto.jpg" class="img-thumbnail" />

}

else

{

//<img src="~/MemberPhoto/@Html.DisplayFor(model => model.MemberPhotoFile)" class="img-thumbnail" />

<img src="~/MemberPhotos/@Model.MemberPhotoFile" class="img-thumbnail" />

}

</div>

</div>

</div>

<div class="card-footer">

@Html.DisplayFor(model => model.CreatedDate)

</div>

</div>

</div>

<p>

@Html.ActionLink("Edit", "Edit", new { id = Model.MemberID }) |

@Html.ActionLink("Back to List", "Index")

</p>

#### Members/Edit

@model HW7Project.Models.Members

@{

ViewBag.Title = "Edit";

Layout = "~/Views/Shared/\_Layout\_Manager.cshtml";

}

<h2>Edit</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>Members</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

@Html.HiddenFor(model => model.MemberID)

<div class="form-group">

@Html.LabelFor(model => model.MemberName, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.MemberName, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.MemberName, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.MemberBirdthday, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.MemberBirdthday, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.MemberBirdthday, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<input type="submit" value="Save" class="btn btn-default" />

<span class+"test-danger">@ViewBag.Error</span>

</div>

</div>

</div>

}

<div>

@Html.ActionLink("Back to List", "Index")

</div>

<script src="~/Scripts/jquery-3.4.1.min.js"></script>

<script src="~/Scripts/jquery.validate.min.js"></script>

<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

#### Members/Index

@using PagedList;

@using PagedList.Mvc;

@\*@model IEnumerable<HW7Project.Models.Members>\*@

@model IPagedList<HW7Project.Models.Members>

@{

ViewBag.Title = "Index";

Layout = "~/Views/Shared/\_Layout\_Manager.cshtml";

}

<h2>Index</h2>

<p class="text-right">

@Html.ActionLink("新增會員資料", "Create", null, new { @class = "btn btn-primary" })

</p>

<table class="table table-striped table-hover">

<thead>

<tr>

<th>

@\*@Html.DisplayNameFor(model => model.MemberID)\*@

@Html.DisplayNameFor(model => model.FirstOrDefault().MemberID)

</th>

<th>

@Html.DisplayNameFor(model => model.FirstOrDefault().MemberName)

</th>

<th>

@Html.DisplayNameFor(model => model.FirstOrDefault().MemberBirdthday)

</th>

<th>

@Html.DisplayNameFor(model => model.FirstOrDefault().CreatedDate)

</th>

<th>

資料管理

</th>

</tr>

</thead>

@foreach (var item in Model)

{

<tr>

<th>

@Html.DisplayFor(modelItem => item.MemberID)

</th>

<td>

@Html.DisplayFor(modelItem => item.MemberName)

</td>

<td>

@Html.DisplayFor(modelItem => item.MemberBirdthday)

</td>

<td>

@Html.DisplayFor(modelItem => item.CreatedDate)

</td>

<td>

@Html.ActionLink("編輯", "Edit", new { id = item.MemberID }, new { @class = "btn btn-info btn-sm" })

@Html.ActionLink("明細", "Details", new { id = item.MemberID }, new { @class = "btn btn-success btn-sm" })

</td>

</tr>

}

</table>

@Html.PagedListPager(Model, page => Url.Action("Index", new { page }))

@section scripts{

<script>

$('.pagination-container').addClass("d-flex justify-content-center");

$('.pagination-container>ul>li').removeClass().addClass("page-item");

$('.pagination-container a').addClass("page-link");

</script>

}

#### Shared/\_Layout

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>@ViewBag.Title - 鍋具商城</title>

<link href="~/Content/bootstrap.min.css" rel="stylesheet" type="text/css" />

<script src="~/Scripts/modernizr-2.8.3.js"></script>

@\*<link href="~/Content/Site.css" rel="stylesheet" type="text/css" />\*@

</head>

<body>

<div class="navbar navbar-expand-lg navbar-dark bg-dark mb-3">

<div class="container">

@Html.ActionLink("鍋具商城", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#MynavbarMenu">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="MynavbarMenu">

<div class="navbar-nav">

@Html.ActionLink("首頁", "Index", "Home", new { area = "" }, new { @class = "nav-link" })

@Html.ActionLink("登入", "Login", "HomeManager", new { area = "" }, new { @class = "nav-link" })

</div>

</div>

</div>

</div>

<section class="container">

@RenderBody()

</section>

<hr />

<footer class="bg-dark mt-5">

<p class="text-white text-center p-3">Copyright &copy; @DateTime.Now.Year - 鍋具商城 All Rights Reserved.</p>

</footer>

<script src="~/Scripts/jquery-3.4.1.min.js"></script>

<script src="~/Scripts/bootstrap.min.js"></script>

@RenderSection("scripts", false)

</body>

</html>

#### Shared/\_Layout\_Manager

<!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>

@\*<link href="~/Content/Site.css" rel="stylesheet" type="text/css" />\*@

<link href="~/Content/bootstrap.min.css" rel="stylesheet" type="text/css" />

<script src="~/Scripts/modernizr-2.8.3.js"></script>

</head>

<body>

<div class="navbar navbar-expand-lg navbar-dark bg-info mb-3">

<div class="container">

@Html.ActionLink("鍋具商城後台管理", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#MynavbarMenu">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="MynavbarMenu">

<div class="navbar-nav">

@Html.ActionLink("管理首頁", "Index", "HomeManager", new { area = "" }, new { @class = "nav-link" })

@Html.ActionLink("客戶管理", "Index", "Members", new { area = "" }, new { @class = "nav-link" })

@Html.ActionLink("商品管理", "Index", "Products", new { area = "" }, new { @class = "nav-link" })

@Html.ActionLink("登出", "Logout", "HomeManager", new { area = "" }, new { @class = "nav-link" })

</div>

</div>

</div>

</div>

<section class="container">

<div class="alert alert-danger">您目前是管理員身分，請記得登出!!</div>

@RenderBody()

</section>

<hr />

<footer class="bg-dark mt-5">

<p class="text-white text-center p-3">Copyright &copy; @DateTime.Now.Year - 鍋具商城 All Rights Reserved.</p>

</footer>

<script src="~/Scripts/jquery-3.4.1.min.js"></script>

<script src="~/Scripts/bootstrap.min.js"></script>

@RenderSection("scripts", false)

</body>

</html>

#### Employees/Create

@model HW7Project.Models.Employees

@{

ViewBag.Title = "Create";

}

<h2>Create</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>Employees</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.EmployeeName, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.EmployeeName, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.EmployeeName, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.CreatedDate, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.CreatedDate, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.CreatedDate, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.Account, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Account, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Account, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.Password, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Password, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Password, "", 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" />

</div>

</div>

</div>

}

<div>

@Html.ActionLink("Back to List", "Index")

</div>

<script src="~/Scripts/jquery-3.4.1.min.js"></script>

<script src="~/Scripts/jquery.validate.min.js"></script>

<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

#### Employees/Delete

@model HW7Project.Models.Employees

@{

ViewBag.Title = "Delete";

}

<h2>Delete</h2>

<h3>Are you sure you want to delete this?</h3>

<div>

<h4>Employees</h4>

<hr />

<dl class="dl-horizontal">

<dt>

@Html.DisplayNameFor(model => model.EmployeeName)

</dt>

<dd>

@Html.DisplayFor(model => model.EmployeeName)

</dd>

<dt>

@Html.DisplayNameFor(model => model.CreatedDate)

</dt>

<dd>

@Html.DisplayFor(model => model.CreatedDate)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Account)

</dt>

<dd>

@Html.DisplayFor(model => model.Account)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Password)

</dt>

<dd>

@Html.DisplayFor(model => model.Password)

</dd>

</dl>

@using (Html.BeginForm()) {

@Html.AntiForgeryToken()

<div class="form-actions no-color">

<input type="submit" value="Delete" class="btn btn-default" /> |

@Html.ActionLink("Back to List", "Index")

</div>

}

</div>

#### Employees/Details

@model HW7Project.Models.Employees

@{

ViewBag.Title = "Details";

}

<h2>Details</h2>

<div>

<h4>Employees</h4>

<hr />

<dl class="dl-horizontal">

<dt>

@Html.DisplayNameFor(model => model.EmployeeName)

</dt>

<dd>

@Html.DisplayFor(model => model.EmployeeName)

</dd>

<dt>

@Html.DisplayNameFor(model => model.CreatedDate)

</dt>

<dd>

@Html.DisplayFor(model => model.CreatedDate)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Account)

</dt>

<dd>

@Html.DisplayFor(model => model.Account)

</dd>

<dt>

@Html.DisplayNameFor(model => model.Password)

</dt>

<dd>

@Html.DisplayFor(model => model.Password)

</dd>

</dl>

</div>

<p>

@Html.ActionLink("Edit", "Edit", new { id = Model.EmployeeID }) |

@Html.ActionLink("Back to List", "Index")

</p>

#### Employees/Edit

@model HW7Project.Models.Employees

@{

ViewBag.Title = "Edit";

}

<h2>Edit</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>Employees</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

@Html.HiddenFor(model => model.EmployeeID)

<div class="form-group">

@Html.LabelFor(model => model.EmployeeName, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.EmployeeName, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.EmployeeName, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.CreatedDate, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.CreatedDate, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.CreatedDate, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.Account, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Account, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Account, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.Password, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.Password, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Password, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<input type="submit" value="Save" class="btn btn-default" />

</div>

</div>

</div>

}

<div>

@Html.ActionLink("Back to List", "Index")

</div>

<script src="~/Scripts/jquery-3.4.1.min.js"></script>

<script src="~/Scripts/jquery.validate.min.js"></script>

<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

#### Employees/Index

@model IEnumerable<HW7Project.Models.Employees>

@{

ViewBag.Title = "Index";

}

<h2>Index</h2>

<p>

@Html.ActionLink("Create New", "Create")

</p>

<table class="table">

<tr>

<th>

@Html.DisplayNameFor(model => model.EmployeeName)

</th>

<th>

@Html.DisplayNameFor(model => model.CreatedDate)

</th>

<th>

@Html.DisplayNameFor(model => model.Account)

</th>

<th>

@Html.DisplayNameFor(model => model.Password)

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.EmployeeName)

</td>

<td>

@Html.DisplayFor(modelItem => item.CreatedDate)

</td>

<td>

@Html.DisplayFor(modelItem => item.Account)

</td>

<td>

@Html.DisplayFor(modelItem => item.Password)

</td>

<td>

@Html.ActionLink("Edit", "Edit", new { id=item.EmployeeID }) |

@Html.ActionLink("Details", "Details", new { id=item.EmployeeID }) |

@Html.ActionLink("Delete", "Delete", new { id=item.EmployeeID })

</td>

</tr>

}

</table>

其他(若有才需貼，並請標示檔名)

#### App\_Start/FilterConfig

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using HW7Project.Controllers;

namespace HW7Project.App\_Start

{

//Global>>>WebApiConfig>>>RouteConfig

//要在Global註冊FilterConfig

public class FilterConfig

{

public static void RegisterGlobalFilters(GlobalFilterCollection filters)

{

filters.Add(new LogReporter());

}

}

}

#### App\_Start/RouteConfig

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Routing;

namespace HW7Project

{

public class RouteConfig

{

public static void RegisterRoutes(RouteCollection routes)

{

routes.IgnoreRoute("{resource}.axd/{\*pathInfo}");

routes.MapRoute(

name: "Default",

url: "{controller}/{action}/{id}",

defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }

);

}

}

}

#### App\_Start/WebApiConfig

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web.Http;

namespace HW7Project

{

public static class WebApiConfig

{

public static void Register(HttpConfiguration config)

{

// Web API 設定和服務

// Web API 路由

config.MapHttpAttributeRoutes();

config.Routes.MapHttpRoute(

name: "DefaultApi",

routeTemplate: "api/{controller}/{id}",

defaults: new { id = RouteParameter.Optional }

);

}

}

}

#### Global

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Routing;

using System.Web.Security;

using System.Web.SessionState;

using System.Web.Http;

using HW7Project.Models;

using System.Data.Entity;

using HW7Project.App\_Start;

namespace HW7Project

{

public class Global : HttpApplication

{

void Application\_Start(object sender, EventArgs e)

{

//新增這一段會初始化資料庫

//Database.SetInitializer<HW7ProjectContext>(new DBInitializer());

// 應用程式啟動時執行的程式碼

AreaRegistration.RegisterAllAreas();

GlobalConfiguration.Configure(WebApiConfig.Register);

RouteConfig.RegisterRoutes(RouteTable.Routes);

FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);

}

}

}