座號： 19姓名：柯昱廷

階段性作八

客戶下訂單實作練習

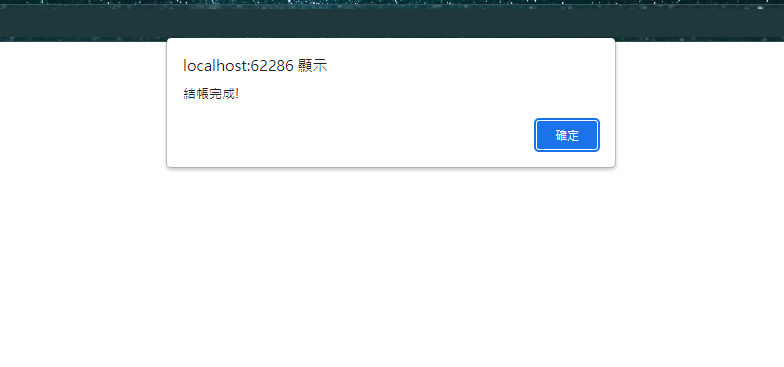
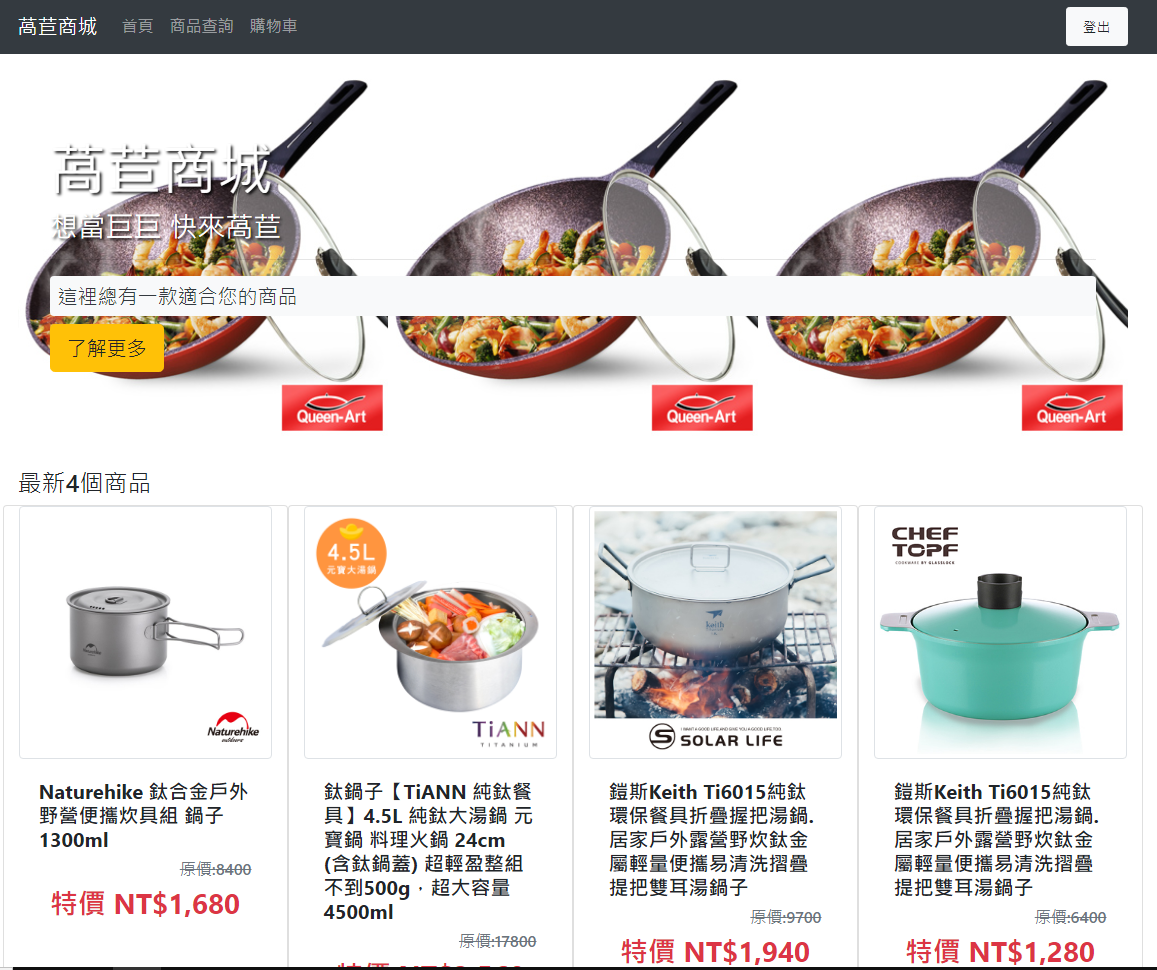
* 承前階段性作業七，完成下列要求：
  1. 客戶登入：
     1. 建立客戶登入功能，其帳號與密碼為Members資料表中的Account與Password。
     2. 登入失敗顯示提示詞，成功則進入預設頁面。
     3. 在選單按下登出鈕後則返回最初登入畫面。
  2. 客戶下訂單功能：
     1. 能由購物車頁面進入下訂單功能。
     2. 客戶必須先登入才能正式下單(以程式流程來控制)。
     3. 建立適當的訂單表單並配置適當的資料。
     4. 使用資料庫作業九的「getOrderID()」的函數取得訂單編號，並將訂單資料正確地存入資料庫中。
     5. 客戶必須先登入才能正式下單。
  3. 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「MyProject○○」，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：

客戶登入功能(至少包含成功與失敗)



下訂單功能(至少包含填表單、下訂成功)

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;

using HW7Project.Models;

namespace HW7Project.Controllers

{

public class HomeController : Controller

{

HW7ProjectContext db=new HW7ProjectContext();

[HandleError(View="Error3")]

public ActionResult ExciptionDemo()

{

int i = 0;

int j = 100 / i;

return View();

}

public ActionResult Test()

{

throw new NotImplementedException();

}

public ActionResult Index()

{

Random r = new Random();

int i=r.Next(0, db.Products.Count());

//select ProductID from Products

var productIDs = db.Products.Select(p => p.ProductID).ToList();

string PID = productIDs[i];

ViewBag.PID = PID;

return View();

}

public ActionResult ProductList()

{

return View();

}

[ChildActionOnly]

public ActionResult \_ProductList(int itemCount=0)

{

List<Products> products;

if (itemCount==0)

products = db.Products.Where(p=>p.Discontinued==false).OrderByDescending(p=>p.CreatedDate).ThenByDescending(p=>p.ProductID).ToList();

else

products = db.Products.Where(p => p.Discontinued == false).OrderByDescending(p => p.CreatedDate).ThenByDescending(p => p.ProductID).Take(itemCount).ToList();

return View(products);

}

[ChildActionOnly]

public ActionResult \_ProductPhotoShow()

{

var products = db.Products.OrderByDescending(p => p.CreatedDate).ThenByDescending(p => p.ProductID).Take(10).ToList();

return View(products);

}

public ActionResult MyCart()

{

return View();

}

public ActionResult Login()

{

return View();

}

[HttpPost]

public ActionResult Login(VMLogin vMLogin)

{

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

string pw = Members.getHashPassword(vMLogin.Password);

var member = db.Members.Where(m => m.Account == vMLogin.Account && m.Password == pw).FirstOrDefault();

if (member == null)

{

ViewBag.ErrMsg = "帳號或密碼有錯!!";

return View(vMLogin);

}

Session["member"] = member;

return RedirectToAction("ProductList");

}

[LoginCheck]

public ActionResult Logout()

{

Session["member"] = null;

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

}

}

}

### 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();

//[LoginCheck]

public ActionResult Index()

{

return View();

}

public ActionResult Login()

{

return View();

}

[HttpPost]

public ActionResult Login(VMLogin vMLogin)

{

//select \* from Employee where account=@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;

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)

{

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

{

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

}

}

public override void OnActionExecuting(ActionExecutingContext filterContext)

{

HttpContext context = HttpContext.Current;

LoginState(context);

}

}

}

### LogReporter

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Routing;

using HW7Project.Models;

using System.IO;

using System.Text;

namespace HW7Project.Controllers

{

public class LogReporter : ActionFilterAttribute

{

HttpContext context;

public bool recordFlag = true;

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);

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)

{

if (recordFlag)

{

context = HttpContext.Current;

LogValues(filterContext.RouteData, context);

}

}

public override void OnResultExecuted(ResultExecutedContext filterContext)

{

if (recordFlag)

{

context = HttpContext.Current;

RequestLog(context);

}

}

}

}

### MemberOrderController

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using HW7Project.Models;

namespace HW7Project.Controllers

{

public class MemberOrderController : Controller

{

private HW7ProjectContext db = new HW7ProjectContext();

[ChildActionOnly]

public ActionResult \_Order()

{

ViewBag.PayTypeID = new SelectList(db.PayTypes, "PayTypeID", "PayTypeName");

ViewBag.ShipID = new SelectList(db.Shippers, "ShipID", "ShipVia");

return View();

}

[HttpPost]

public ActionResult Order(string ShipName, string ShipAddress, int ShipID, int PayTypeID, string data)

{

//如果沒登入,導去登入

if (Session["member"] == null)

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

int memberID = ((Members)Session["member"]).MemberID;

//如果已登入,就寫入資料庫

List<SqlParameter> list = new List<SqlParameter>

{

new SqlParameter("ShipName",ShipName),

new SqlParameter("ShipAddress",ShipAddress),

new SqlParameter("ShipID",ShipID),

new SqlParameter("PayTypeID",PayTypeID),

new SqlParameter("MemberID",memberID),

new SqlParameter("cart",data)

};

SetData sd = new SetData();

sd.executeSqlBySP("addOrders", list);

TempData["cartFlag"] = "OK";

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

}

}

}

### MembersController

using System;

using System.Collections.Generic;

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;

using System.Configuration;

namespace HW7Project.Controllers

{

//[LoginCheck]

public class MembersController : Controller

{

private HW7ProjectContext db = new HW7ProjectContext();

// GET: Members

public ActionResult Index(int page=1)

{

var members = db.Members.ToList();

int pagesize = 15;

var pagedList = members.ToPagedList(page, pagesize);

return View(pagedList);

}

// 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();

}

return PartialView(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,MemberBirthday,CreatedDate,Account,Password")] Members members,HttpPostedFileBase photo)

{

if (photo != null)

{

if(photo.ContentLength>0)

{

string extensionName = System.IO.Path.GetExtension(photo.FileName);

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

{

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

members.MemberPhotoFile = members.Account + extensionName;

}

}

}

if (ModelState.IsValid)

{

db.Members.Add(members);

db.SaveChanges();

return RedirectToAction("Index");

}

return View();

}

[Route(@"M/{name}")]

public ActionResult EditByName(string name)

{

Members members = db.Members.Where(m => m.MemberName == name).FirstOrDefault();

if (members == null)

{

return HttpNotFound();

}

return View("Edit",members);

}

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);

}

// 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]

public ActionResult Edit(Members members)

{

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

SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["HW7ProjectConnection"].ConnectionString);

SqlCommand cmd = new SqlCommand(sql, conn);

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

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

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);

}

}

}

### ProductsController

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 ProductsController : Controller

{

private HW7ProjectContext db = new HW7ProjectContext();

// GET: Products

public ActionResult Index()

{

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

}

[LogReporter(recordFlag = false)]

public FileContentResult GetImage(string id)

{

var photo = db.Products.Find(id);

if (photo == null)

return null;

return File(photo.PhotoFile, photo.ImageMimeType);

}

// GET: Products/Details/5

public ActionResult Details(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Products products = db.Products.Find(id);

//Products products = db.Products.Where(p => p.ProductID == id).FirstOrDefault();

if (products == null)

{

return HttpNotFound();

}

return PartialView(products);

}

// GET: Products/Create

public ActionResult Create()

{

return PartialView();

}

// POST: Products/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(Products products, HttpPostedFileBase photo)

{

if (photo == null)

{

ViewBag.ErrMessage = "請上傳商品照片";

return View(products);

}

if (db.Products.Find(products.ProductID) != null)

{

ViewBag.ErrMessage2 = "商品編號重複";

return View(products);

}

products.ImageMimeType = photo.ContentType;

products.PhotoFile = new byte[photo.ContentLength];

photo.InputStream.Read(products.PhotoFile, 0, photo.ContentLength);

products.CreatedDate = DateTime.Today;

products.Discontinued = false;

ModelState.Remove("PhotoFile");

if (ModelState.IsValid)

{

db.Products.Add(products);

db.SaveChanges();

return RedirectToAction("Index");

}

return RedirectToAction("Index");

}

// GET: Products/Edit/5

public ActionResult Edit(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Products products = db.Products.Find(id);

if (products == null)

{

return HttpNotFound();

}

return View(products);

}

// POST: Products/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(Products products, HttpPostedFileBase photo)

{

if (photo != null)

{

products.ImageMimeType = photo.ContentType;

products.PhotoFile = new byte[photo.ContentLength];

photo.InputStream.Read(products.PhotoFile, 0, photo.ContentLength);

}

ModelState.Remove("PhotoFile");

if (ModelState.IsValid)

{

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

db.SaveChanges();

return RedirectToAction("Index");

}

return RedirectToAction("Index");

}

public ActionResult ProductStatusChange(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Products products = db.Products.Find(id);

products.Discontinued = !products.Discontinued;

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

db.SaveChanges();

return RedirectToAction("Index");

}

// GET: Products/Delete/5

public ActionResult Delete(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Products products = db.Products.Find(id);

if (products == null)

{

return HttpNotFound();

}

return View(products);

}

// POST: Products/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(string id)

{

Products products = db.Products.Find(id);

db.Products.Remove(products);

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)

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.EmployeesID }) |

@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.EmployeesID)

<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.EmployeesID }) |

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

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

</td>

</tr>

}

</table>

Home>Index

@{

ViewBag.Title = "Index";

int c = 4;

}

@section css{

<style>

.jumbotron {

background-image: url('/Products/GetImage/@ViewBag.PID');

background-repeat:round;

}

</style>

}

<div class="jumbotron text-white">

<h1 class="display-4 text-shadow "><strong>萵苣商城</strong></h1>

<h3 class="text-shadow">想當巨巨 快來萵苣</h3>

<hr>

<h5 class="bg-light text-dark p-2 rounded">這裡總有一款適合您的商品</h5>

@Html.ActionLink("了解更多","ProductList", null, new { @class="btn btn-warning btn-lg"})

</div>

<div class="row">

<div class="col-12">

<h4>最新@(c)個商品</h4>

@Html.Action("\_ProductList", new { itemCount = c })

<div class="text-right">@Html.ActionLink("查看更多...", "ProductList")</div>

</div>

</div>

<div class="row">

<div class="col-12">

@Html.Action("\_ProductPhotoShow")

</div>

</div>

@section scripts{

<script src="~/Scripts/MyJS.js"></script>

}

Home>Mycart

@{

ViewBag.Title = "MyCart";

var OderOK = 0;

if (@TempData["cartFlag"] != null) { OderOK = 1; }

}

<h2 class="text-center text-info shadow p-3">我的購物車</h2>

<div id="MyCart">

</div>

<div id="total" class="text-right"></div>

<div class="text-right">

<a class="btn btn-info " id="updateCart" href="#">更新購物車</a>

<a class="btn btn-warning" id="clearCart" href="#">清空購物車</a>

<button class="btn btn-primary checkout" type="button" data-toggle="collapse" data-target="#collapseOrder">

<i class="bi bi-cash-coin"></i> 結帳

</button>

</div>

<div class="collapse rounded bg-light mt-5 p-4" id="collapseOrder">

<h2 class="text-center">訂單資料填寫</h2>

@Html.Action("\_Order", "MemberOrder")

</div>

<div>

@Html.ActionLink("返回商品頁面", "ProductList")

</div>

@section scripts{

<script>

if (localStorage.getItem("cart")) {

$('#data').val(localStorage.getItem("cart"));

}

var cart = [];

ShowCart();

if (@OderOK== 1) {

alert("結帳完成!");

clearCart();

}

//顯示購物車商品

function ShowCart() {

if (localStorage.getItem('cart')) {

cart = JSON.parse(localStorage.getItem('cart'));

let cartList = "";

let total = 0;

for (let p in cart) {

/\*cart[p]["PName"];\*/

let price = Number(cart[p]["Price"]) \* Number(cart[p]["Amount"]);

total += price;

cartList += `

<div class="media border broder-black m-2 p-2">

<img src="/Products/GetImage/${cart[p]["PID"]}" width="200" class="align-self-center mr-3" alt="...">

<div class="media-body ">

<h3 class="mt-0">${cart[p]["PName"]}</h3>

</div>

<div>

<p>單價:${cart[p]["Price"]}</p>

<p>數量:<input min="1" id="${cart[p]["PID"]}" type="number" value="${cart[p]["Amount"]}" style="width:60px" /></p>

<h5 class="text-danger">總價:${price}</h5>

<button class="btn-sm btn-outline-danger btnDelItem" id="${p}">刪除此商品</button>

</div>

</div>

`

}

$('#total').html(`總計：<span class="text-danger">${total}</span>`);

$('#MyCart').html(cartList);

}else {

$('#MyCart').html(`<h2 class='text-center text-danger'>購物車目前無商品</h2>`);

}

}

function clearCart() {

cart = [];

localStorage.removeItem('cart');

$('#MyCart').text("");

$('#total').text("");

ShowCart();

}

//clear購物車

$('#clearCart').click(() => {

if (confirm('是否確定清空購物車?')) {

clearCart();

}

});

//updateCart

$('#updateCart').click(() => {

for (i = 0; i < cart.length; i++) {

cart[i]["Amount"] = $('#' + cart[i]["PID"]).val();

}

localStorage.setItem("cart", JSON.stringify(cart));

ShowCart();

});

//刪除單一商品

$('.btnDelItem').on('click',function (evt) {

if (confirm('確定刪除此商品')) {

cart.splice(evt.target.id, 1);

localStorage.setItem("cart", JSON.stringify(cart));

if (localStorage.getItem("cart") == "[]") {

$('#MyCart').text("");

$('#total').text("");

localStorage.removeItem("cart");

}

ShowCart();

}

})

</script>

}

Home> ProductList

@model IEnumerable<HW7Project.Models.Products>

@{

ViewBag.Title = "ProductList";

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

}

<h2>ProductList</h2>

<div class="row">

@foreach (var item in Model)

{

<div class="card col-3">

<img class="img-thumbnail" src="@Url.Action("GetImage","Products", new { id = item.ProductID })" />

<div class="card-body d-flex flex-column justify-content-between ">

<input class="PID" type="hidden" name="name" value="@item.ProductID" />

<h4><strong class="PName">@Html.DisplayFor(modelItem => item.ProductName)</strong></h4>

<div>

<h5 class="text-secondary text-right " style="text-decoration-line:line-through;">原價 @(item.UnitPrice\*3)</h5>

<h3 class="text-danger " style="text-shadow:2px 1px 3px #f86666;">超值特價<span class="Price" title="@item.UnitPrice">@Html.DisplayFor(modelItem => item.UnitPrice)</span></h3>

</div>

@\*@Html.DisplayFor(modelItem => item.Description)\*@

</div>

<div class="card-footer">

<button class="btn btn-info btn-lg btn-block addCart"><i class="bi bi-cart4"></i>加入購物車</button>

</div>

</div>

}

<span style="font-size:4rem; position:fixed;right:6px;bottom:50px;width:80px;height:80px;">

<span class="text-white bg-danger rounded-circle text-center" style="position:fixed;font-size:1rem;width:20px;height:20px;right:30px;bottom:100px" id="itemAmount">0</span><a href="@Url.Action("MyCart")"><i class="bi bi-cart4"></i></a>

</span>

</div>

@section scripts{

<script>

var cart = [];

if (localStorage.getItem("cart")) {

cart = JSON.parse(localStorage.getItem("cart"));

$('#itemAmount').text(cart.length);

}

console.log(cart.length);

$('.addCart').click(function(){

let product = $(this).closest('.card');

let newPID = product.find(".PID").val();

let foundItem = cart.find((item) => item.PID == newPID);

if (foundItem === undefined) { setCart(product); $('#itemAmount').text(cart.length); } else {

foundItem.Amount += 1;

alert('商品數量已更新');

localStorage.setItem('cart', JSON.stringify(cart));

}

//let pName = product.find(".PName").text();

//let price = product.find(".Price").attr("title");

//let amount = 1;

//if (cart.length > 0) {

// for (let index in cart) {

// if (newPID == cart[index]["PID"]) {

// cart[index].Amount += 1;

// alert('商品數量已更新');

// localStorage.setItem("cart", JSON.stringify(cart));

// break;

// } else if (cart.length - 1 == index) {

// setCart()

// alert('已加入購物車');

// }

// }

//} else {

// setCart(product);

//}

$('#itemAmount').text() = cart.length;

});

function setCart(product) {

let newItem = {

PID: product.find(".PID").val(),

PName: product.find(".PName").text(),

Price: product.find(".Price").attr("title"),

Amount: 1

}

cart.push(newItem);

localStorage.setItem("cart", JSON.stringify(cart));

alert('已加入購物車');

}

</script>

}

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 = "後台管理";

}

<h2 class="mt-3">後台管理</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>登入</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>

<h1 class="text-danger">@ViewBag.ErrMsg</h1>

<div class="form-group">

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

<input type="submit" value="登入" class="btn btn-success" />

<input type="submit" value="清除" class="btn btn-dark" />

</div>

</div>

</div>

}

<div>

@Html.ActionLink("Back to 首頁", "Index", "Home")

</div>

@section scripts{

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

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

}

MemberOrder>\_Order

@model HW7Project.Models.Orders

@{ Layout = "";}

@using (Html.BeginForm("Order", "MemberOrder", FormMethod.Post))

{

@Html.AntiForgeryToken()

<div class="form-horizontal ">

<hr />

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

<div class="form-group">

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

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

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

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

</div>

</div>

<div class="form-group">

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

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

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

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

</div>

</div>

<div class="form-group">

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

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

@Html.DropDownList("ShipID", null, htmlAttributes: new { @class = "form-control" })

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

</div>

</div>

<div class="form-group">

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

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

@Html.DropDownList("PayTypeID", null, htmlAttributes: new { @class = "form-control" })

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

</div>

</div>

<input type="hidden" id="data" name="data" />

<div class="form-group">

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

<input type="submit" value="確定結帳" class="btn btn-info" />

</div>

</div>

</div>

}

Members> Create

@model HW7Project.Models.VMMember

@{

ViewBag.Title = "Create";

var CreateDate = DateTime.Today.ToShortDateString();

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

}

<h2>Create</h2>

@\*<form action="/" method="post" enctype="multipart/form-data"></form>\*@

@using (Html.BeginForm("Create","Members",FormMethod.Post, new { enctype = "multipart/form-data" }))

{

@Html.AntiForgeryToken()

<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 { htmlAttributes = new { @class = "form-control" } })\*@

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

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

@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>

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

@\*<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" } })

@\*<span class="text-danger">@ViewBag.Error</span>\*@

@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> Delete

@model HW7Project.Models.Members

@{

ViewBag.Title = "Delete";

}

<h2>Delete</h2>

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

<div>

<h4>Members</h4>

<hr />

<dl class="dl-horizontal">

<dt>

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

</dt>

<dd>

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

</dd>

<dt>

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

</dt>

<dd>

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

</dd>

<dt>

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

</dt>

<dd>

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

</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>

Members> Details

@model HW7Project.Models.Members

@{

ViewBag.Title = "Details";

}

<div>

<div class="card">

<div class="card-header">

<h2 class="text-center">@Html.DisplayFor(model => model.MemberName)</h2>

</div>

<div class="card-body">

<div class="row">

<div class="col-md-8 d-flex justify-content-center flex-column ">

<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">

@\*@Html.DisplayFor(model => model.MemberPhotoFile)\*@

@\*var photo=(Model.MemberPhotoFile == null?"":"~/MemberPhoto/Koala.jpg")\*@

@if (Model.MemberPhotoFile == null)

{

<img src="~/MemberPhoto/Koala.jpg" class="img-thumbnail shadow" />

}

else

{

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

}

</div>

</div>

</div>

<div class="card-footer">

<div class="text-center">建立日期:@Html.DisplayFor(model => model.CreatedDate)</div>

</div>

</div>

<dl class="dl-horizontal">

</dl>

</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>修改基本資料</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>會員</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.MemberPhotoFile, htmlAttributes: new { @class = "control-label col-md-2" })

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

@\*@Html.EditorFor(model => model.MemberPhotoFile, new { htmlAttributes = new { @class = "form-control", @readonly = "@readonly" } })\*@

<!--@{

var photo = Model.MemberPhotoFile == null ? "Koala.jpg" : Model.MemberPhotoFile;

}

<img src="~/MemberPhoto/@photo" class="img-thumbnail" width="300" />

@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",type="date" } })

@\*@Html.Raw(""+Model.MemberBirdthday.ToString())\*@

@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="儲存" class="btn btn-success" />

</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 IPagedList<HW7Project.Models.Members>

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

@{

ViewBag.Title = "Index";

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

}

<h2>會員資料列表</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>

<td>

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

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

</td>

<td class="text-right">

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

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

@\*@Html.ActionLink("Delete", "Delete", new { id = item.MemberID })\*@

<button title="@item.MemberID" type="button" class="btn btn-success btnDetail" data-toggle="modal" data-target="#exampleModal" >

詳細

</button>

</td>

</tr>

}

</table>

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

<!-- Button trigger modal -->

<!-- Modal -->

<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLabel">詳細資料</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

@\*<button type="button" class="btn btn-primary">Save changes</button>\*@

</div>

</div>

</div>

</div>

@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 ");

$('.pagination-container>ul>li>a:not(["href"])').addClass("active");

/\*$('.pagination-container li > a!["href"] ').addClass("active");\*/

$('.btnDetail').click(function (ev) {

console.log("/Members/Details/" + ev.target.title);

$('#exampleModal .modal-body').load("/Members/Details/" + ev.target.title);

});

</script>

}

Products> Create

@model HW7Project.Models.Products

@{

ViewBag.Title = "Create";

Layout = "";

}

@using (Html.BeginForm("Create", "Products", FormMethod.Post, new { enctype = "multipart/form-data" }))

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

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

<div class="form-group">

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

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

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

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

<span class="field-validation-valid text-danger">@ViewBag.ProductIdErr</span>

</div>

</div>

<div class="form-group">

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

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

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

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

</div>

</div>

<div class="form-group">

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

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

<img width="200" id="newPhoto" />

<input id="photo" name="photo" type="file" class="form-control-file" data-val="true" data-val-required="請上傳商品照片" />

<span class="field-validation-valid text-danger" data-valmsg-for="photo" data-valmsg-replace="true"></span>

<span class="field-validation-valid text-danger">@ViewBag.PhotoUpFileErr</span>

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

</div>

</div>

<div class="form-group">

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

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

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

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

</div>

</div>

<div class="form-group">

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

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

@Html.TextAreaFor(model => model.Description, new { @class = "form-control" })

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

</div>

</div>

<div class="form-group">

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

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

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

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

</div>

</div>

<div class="form-group">

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

<input type="submit" value="確定新增" class="btn btn-info" />

</div>

</div>

</div>

}

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

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

<script>

$('#photo').change(() => {

var oFReader = new FileReader();

var file = document.getElementById('photo').files[0];

oFReader.readAsDataURL(file);

oFReader.onloadend = function (oFRevent) {

var src = oFRevent.target.result;

$('#newPhoto').attr('src', src);

}

});

/\*範例程式

$("#photo").change(() => {

var oFReader = new FileReader();

var file = document.getElementById('photo').files[0];

oFReader.readAsDataURL(file);

oFReader.onloadend = function (oFRevent) {

var src = oFRevent.target.result;

$('#newPhoto').attr('src', src);

}

});\*/

</script>

Products> Delete

@model HW7Project.Models.Products

@{

ViewBag.Title = "Delete";

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

}

<h2>Delete</h2>

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

<div>

<h4>Products</h4>

<hr />

<dl class="dl-horizontal">

<dt>

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

</dt>

<dd>

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

</dd>

<dt>

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

</dt>

<dd>

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

</dd>

<dt>

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

</dt>

<dd>

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

</dd>

<dt>

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

</dt>

<dd>

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

</dd>

<dt>

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

</dt>

<dd>

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

</dd>

<dt>

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

</dt>

<dd>

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

</dd>

<dt>

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

</dt>

<dd>

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

</dd>

<dt>

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

</dt>

<dd>

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

</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>

Products> Details

@model HW7Project.Models.Products

@{

ViewBag.Title = "Details";

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

}

<div class="card">

<div class="card-header bg-danger text-white">

<h4>@Html.DisplayFor(model => model.ProductName)</h4>

</div>

<div class="card-body row">

<div class="col-4">

<img src="@Url.Action("GetImage", new { id = Model.ProductID })" class="img-thumbnail" />

</div>

<div class="col-8 d-flex flex-column justify-content-center align-items-center ">

<div class="border border-secondary p-2 ">

<p>

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

</p>

<p>

@Html.Raw( @Html.DisplayFor(model => model.Description).ToString().Replace("\n","<br />"))

</p>

</div>

<p>

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

@Html.DisplayFor(model => model.UnitPrice, new {@class="text-danger" }) |

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

@Html.DisplayFor(model => model.UnitsInStock) |

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

@{ if (Model.Discontinued)

{<i class="bi bi-bookmark-x text-danger"></i> }

else

{ <i class="bi bi-bookmark-check text-success"></i>}

} |

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

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

</p>

</div>

</div> <hr />

<div class="card-footer text-center justify-content-center">

<p>

@Html.ActionLink("修改", "Edit", new { id = Model.ProductID }, new { @class = "btn-lg btn-info" })

</p>

</div>

</div>

Products> Edit

@model HW7Project.Models.Products

@{

ViewBag.Title = "Edit";

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

}

<h2>修改</h2>

@using (Html.BeginForm("Edit", "Products", FormMethod.Post, new { enctype = "multipart/form-data" }))

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<hr />

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

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

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

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

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

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

<div class="form-group">

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

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

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

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

</div>

</div>

<div class="form-group">

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

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

原有照片：<img src="@Url.Action("GetImage", new {id=Model.ProductID})" width="200" />

新照片預覽：<img width="200" id="newPhoto" />

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

<span class="field-validation-valid text-danger">@ViewBag.ErrMessage</span>

</div>

</div>

<div class="form-group">

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

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

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

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

</div>

</div>

<div class="form-group">

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

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

@\*<textarea id="TextArea1" rows="2" cols="20"></textarea>\*@

@Html.TextAreaFor(model => model.Description, new { @class = "form-control", rows = "5" })

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

</div>

</div>

<div class="form-group">

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

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

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

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

</div>

</div>

</div>

<div class="form-group">

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

<input type="submit" value="確定更新" class="btn btn-warning" />

</div>

</div>

}

<div>

@Html.ActionLink("回商品清單", "Index")

</div>

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

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

<script>

$('#photo').change(() => {

var oFReader = new FileReader();

var file = document.getElementById('photo').files[0];

oFReader.readAsDataURL(file);

oFReader.onloadend = function (oFRevent) {

var src = oFRevent.target.result;

$('#newPhoto').attr('src', src);

}

});

</script>

Products> Index

@model IEnumerable<HW7Project.Models.Products>

@{

ViewBag.Title = "Index";

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

string a = "";

}

<h2>商品管理</h2>

<p>

@\*@Html.ActionLink("新增商品", "Create")\*@

<button type="button" class="btn btn-info" data-toggle="modal" data-target="#exampleModal" id="btnCreate">

新增商品

</button>

</p>

<table class="table">

<tr>

<th>

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

</th>

<th>

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

</th>

@\*<th>

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

</th>\*@

<th>

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

</th>

<th>

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

</th>

<th>

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

</th>

<th>

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

</th>

<th>

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

</th>

<th>

</th>

</tr>

@foreach (var item in Model)

{

<tr>

<td>

<span class="d-inline-block text-truncate" style="max-width: 300px;">

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

</span>

</td>

<td><img src="@Url.Action("GetImage", new { id = item.ProductID })" width="100" /></td>

@\*<td>

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

</td>\*@

@\*<td>

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

</td>\*@

<td>

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

</td>

<td>

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

</td>

<td>

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

</td>

<td>

@{ if (item.Discontinued)

{<i class="bi bi-bookmark-x text-danger"></i>

a = "上";

}

else

{ <i class="bi bi-bookmark-check text-success"></i>

a = "下";

}

}

@\*@Html.DisplayFor(modelItem => item.Discontinued)\*@

</td>

<td>

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

</td>

<td>

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

@\*@Html.ActionLink("Details", "Details", new { id = item.ProductID })\*@

@Html.ActionLink("商品"+a+"架", "ProductStatusChange", new { id=item.ProductID})

<button type="button" class="btn btn-primary myBtn" id="@item.ProductID" data-toggle="modal" data-target="#exampleModal">

明細

</button>

</td>

</tr>

}

</table>

<!-- Button trigger modal -->

<!-- Modal -->

<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog modal-dialog-scrollable modal-xl modal-dialog-centered">

<div class="modal-content">

<div class="modal-header">

<h2 class="modal-title" id="exampleModalLabel"></h2>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

...

</div>

</div>

</div>

</div>

@section scripts{

<script>

$('.myBtn').click(function (evt) {

$('#exampleModal .modal-body').load("/Products/Details?id=" + evt.target.id);

$('#exampleModalLabel').text("商品明細");

});

$('#btnCreate').click(function () {

$('#exampleModal .modal-body').load("/Products/Create")

$('#exampleModalLabel').text("新增商品");

});

</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 - 萵苣商城</title>

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

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

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.9.1/font/bootstrap-icons.css">

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

</head>

<body>

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

<div class="container">

@Html.ActionLink("萵苣商城", "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 d-flex justify-content-between" id="MynavbarMenu">

<div class="navbar-nav">

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

@Html.ActionLink("商品查詢", "ProductList", "Home", new { area = "" }, new { @class = "nav-link" })

</div>

<div>

@if (Session["member"] == null)

{

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

else

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

</div>

</div>

</div>

</div>

<section class="container">

@RenderBody()

</section>

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

<p class="text-white text-center p-3">Copyright &copy; @DateTime.Now.Year - 萵苣商城 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 - 萵苣商城</title>

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

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

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.9.1/font/bootstrap-icons.css">

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

</head>

<body>

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

<h1 class="text-white text-right">管理者</h1>

<div class="container">

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

</div>

</div>

</div>

<h1 class="text-white text-left">管理者</h1>

</div>

<section class="container">

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

@RenderBody()

</section>

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

<div class="d-flex justify-content-center bg-danger justify-content-between">

<h1 class="text-white text-left">管理者</h1>

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

<h1 class="text-white text-left">管理者</h1>

</div>

</footer>

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

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

@RenderSection("scripts", false)

</body>

</html>

其他(若有才需貼，並請標示檔名)

LogReporter.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Routing;

using HW7Project.Models;

using System.IO;

using System.Text;

namespace HW7Project.Controllers

{

public class LogReporter:ActionFilterAttribute

{

HttpContext context;

public bool recordFlag = true;

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["ooxx"] == null?"Guest":context.Session["ooxx"].ToString();

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

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 httpMetod = context.Request.HttpMethod;

var userAgent = context.Request.UserAgent;

var requestTime = DateTime.Now;

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

sw.WriteLine($"{requestTime},{ip},{host},{browser},{requestType},{userHostAddress},{userHostName},{httpMetod},{userAgent}");

sw.Close();

}

public override void OnActionExecuting(ActionExecutingContext filterContext)

{

if (recordFlag) {

context=HttpContext.Current;

LogValues(filterContext.RouteData, context);}

}

public override void OnResultExecuted(ResultExecutedContext filterContext)

{

if (recordFlag) {

context = HttpContext.Current;

RequestLog(context);}

}

}

}