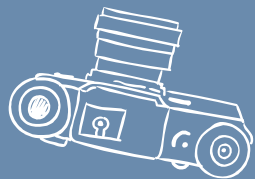
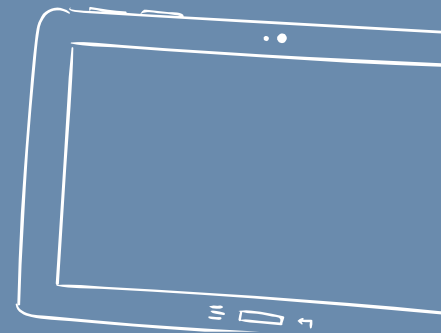


第八組

10527144 資訊四甲 周謐嬰

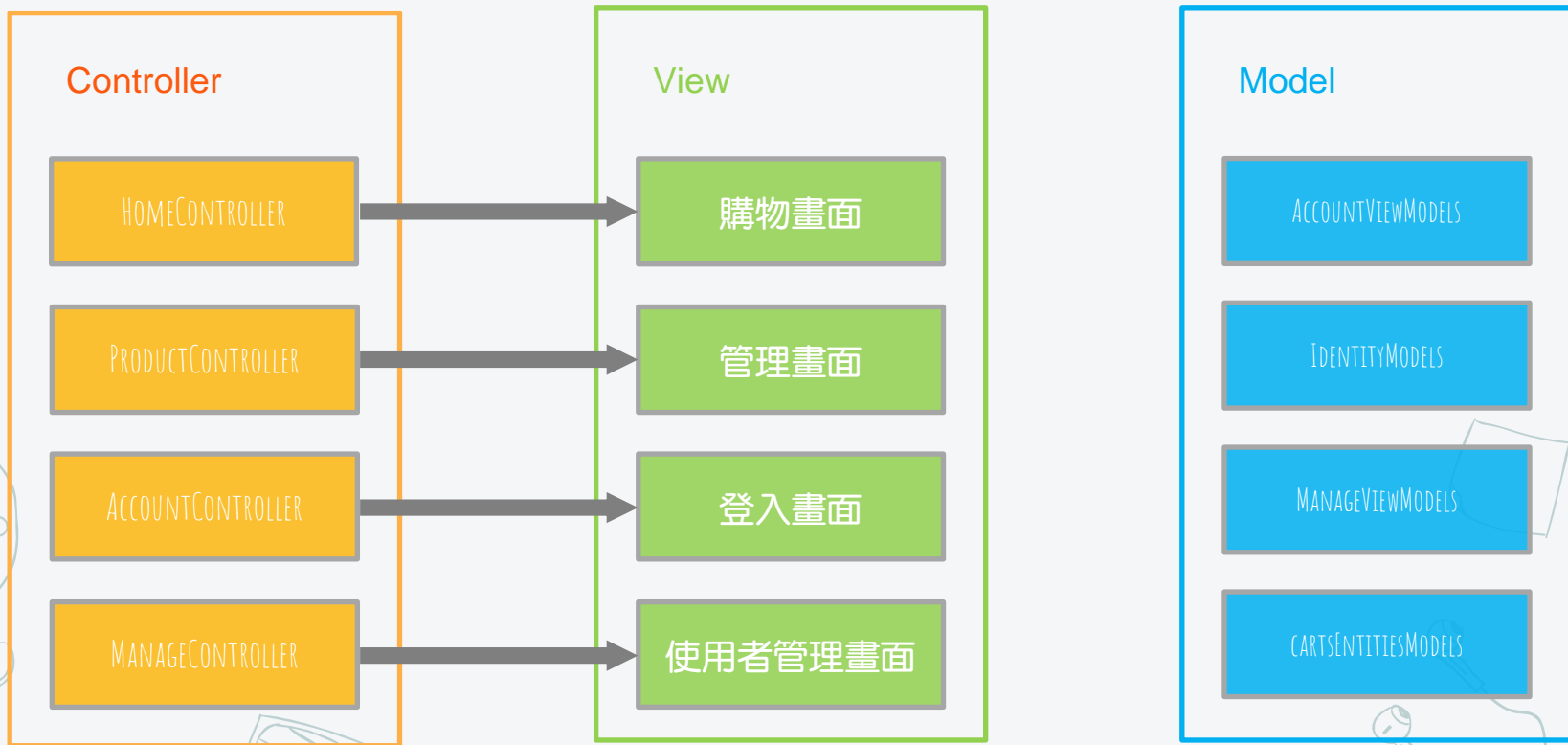
10620115 資訊三乙 王俊儒



1. 技術深度



網頁架構



資料驗證

需要在view的頁面上加上

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

載入函式庫，不然無法正常顯示

購物網站 首頁 購物車 註冊 登入

登入.

使用本機帳戶進行登入。

電子郵件

密碼

☐ 記住我?

登入

以新使用者身分註冊

忘記密碼?

© 2020 - 進階網路資訊系統第八組

購物網站 首頁 新增商品 您好 Admin

新增商品
tProducts

商品名稱

upload

價錢

描述

數量

商品名稱 欄位是必要項。
選擇檔案 永遠選擇任何檔案
欄位 價錢 必須是數字。
描述 欄位是必要項。
請輸入大於1的數字

Create

© 2020 - 進階網路資訊系統第八組

```
3 個參考  
public class RegisterViewModel  
{  
    [Required]  
    [EmailAddress]  
    [Display(Name = "電子郵件")]  
    5 個參考  
    public string Email { get; set; }  
  
    [Required]  
    [StringLength(100, ErrorMessage = "{0} 的長度至少必須為 {2} 個字元。", MinimumLength = 6)]  
    [DataType(DataType.Password)]  
    [Display(Name = "密碼")]  
    3 個參考  
    public string Password { get; set; }  
  
    [DataType(DataType.Password)]  
    [Display(Name = "確認密碼")]  
    [Compare("Password", ErrorMessage = "密碼和確認密碼不相符。")]  
    2 個參考  
    public string ConfirmPassword { get; set; }  
}
```

```
public partial class tProducts  
{  
    [Display(Name = "序號")]  
    14 個參考  
    public int fld { get; set; }  
  
    [Required]  
    [Display(Name = "商品名稱")]  
    9 個參考  
    public string fName { get; set; }  
  
    13 個參考  
    public string fImage { get; set; }  
  
    [Required]  
    [Display(Name = "價錢")]  
    [Range(1, 9999, ErrorMessage = "請輸入大於1的數字")]  
    [RegularExpression(@"^[0-9]*$", ErrorMessage = "請輸入數字")]  
    8 個參考  
    public int fPrice { get; set; }  
  
    [Required]  
    [Display(Name = "描述")]  
    8 個參考  
    public string fDescription { get; set; }  
  
    [Required]  
    [Range(1, 9999, ErrorMessage = "請輸入大於1的數字")]  
    [RegularExpression(@"^[0-9]*$", ErrorMessage = "請輸入數字")]  
    [Display(Name = "數量")]  
    9 個參考  
    public int fQuantity { get; set; }  
}
```

登入流程

透過PasswordSignInAsync回傳登入結果給result，根據result做不同的行為。

shouldLockout: 是否在登入失敗紀錄次數，預設為true

```
public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }

    // 這不會計算為帳戶鎖定的登入失敗
    // 若要啟用密碼失敗來觸發帳戶鎖定，請變更為 shouldLockout: true
    var result = await SignInManager.PasswordSignInAsync(model.Email, model.Password, model.RememberMe, shouldLockout: false);
    switch (result)
    {
        case SignInStatus.Success:
            if (model.Email == "Admin@admin")
                return RedirectToRoute(new { controller = "Product", action = "Index" });
            else
                return RedirectToLocal(returnUrl);
        case SignInStatus.LockedOut:
            return View("Lockout");
        case SignInStatus.RequiresVerification:
            return RedirectToAction("SendCode", new { ReturnUrl = returnUrl, RememberMe = model.RememberMe });
        case SignInStatus.Failure:
        default:
            ModelState.AddModelError("", "登入嘗試失誤。");
            return View(model);
    }
}
```

設定管理員帳號

Admin@admin

登入成功後導向管理頁面→

其餘帳號導向購物頁面

註冊流程

ValidateAntiForgeryToken:防止 CSRF (Cross-Site Request Forgery) 跨站偽造請求的攻擊

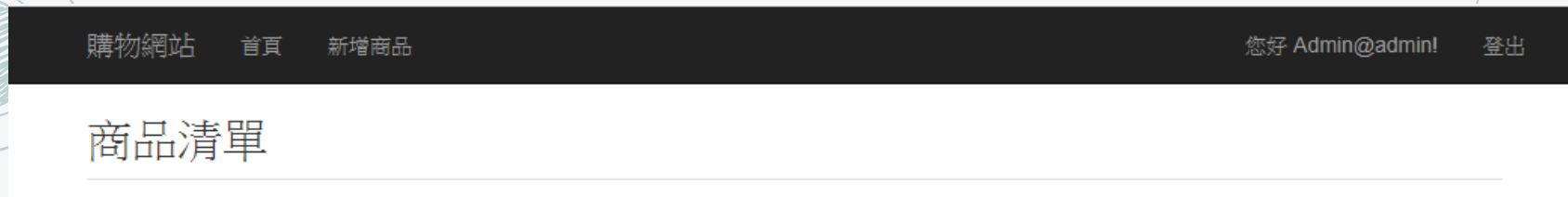
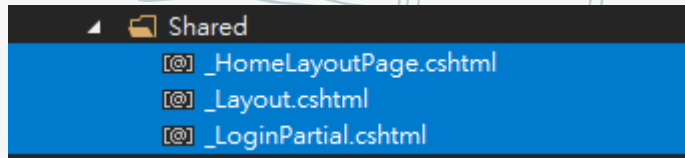
透過CreateAsync新增user跟他的密碼

```
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
0 個參考
public async Task<ActionResult> Register(RegisterViewModel model)
{
    if (ModelState.IsValid)
    {
        var user = new ApplicationUser { UserName = model.Email, Email = model.Email };
        var result = await UserManager.CreateAsync(user, model.Password);
        if (result.Succeeded)
        {
            await SignInManager.SignInAsync(user, isPersistent:false, rememberBrowser:false);
            if (model.Email == "Admin@admin")
            {
                return RedirectToRoute(new { controller = "Product", action = "Index" });
            }
            else
            {
                return RedirectToAction("Homepage", "Home");
            }
        }
        AddErrors(result);
    }

    // 如果執行到這裡，發生某項失敗，則重新顯示表單
    return View(model);
}
```

_layout

根據不同頁面功能顯示不同的_layout



購物流程-資料庫

會使用到的資料表

tProducts: 儲存所有已上架的商品

tCart: 儲存加入購物車的商品

tOrder: 儲存購買人購買之商品及收件資訊

tProducts	
屬性	
fId	
fName	
fImage	
fPrice	
fDescription	
fQuantity	
fUserId	
導覽屬性	

tCart	
屬性	
fId	
fUserId	
fPdName	
fPdImage	
fPdPrice	
fPdDescription	
fBuyQuantity	
導覽屬性	

tOrder	
屬性	
fId	
fUserId	
fPdName	
fPdImage	
fPdPrice	
fPdDescription	
fBuyQuantity	
fUserName	
fUserPhone	
fUserAddress	
fUserPayway	
導覽屬性	

管理頁面-新增商品

新增圖片到/Images資料夾中，檔名為商品名稱。

新增商品

tProducts

商品名稱

upload

2.jpg

價錢

描述

數量

```
if (file != null && file.ContentLength > 0)
{
    string fileName = Request.Form["fName"] + Path.GetExtension(file.FileName);
    product.fImage = fileName;
    var path = Path.Combine(Server.MapPath("~/Images"), fileName);
    file.SaveAs(path);
}
```

管理頁面-編輯商品

如果編輯商品有上傳新的圖片，把原本資料夾的圖片刪掉，新增一個跟商品名稱一樣的圖片到Images

修改商品內容

tProducts

商品名稱

product3

upload

選擇檔案 2.jpg

價錢

300

描述

紙箱的貓

數量

300

Save

返回

```
public ActionResult Edit(int id, string fName, string fImage, int fPrice, string fDescription, int fQuantity, HttpPostedFileBase file)
{
    var product = db.tProducts.Where(m => m.fId == id).FirstOrDefault();
    string path = "";
    if (product.fImage != null)
        path = Path.Combine(Server.MapPath("~/Images"), product.fImage.ToString());

    product.fName = fName;
    product.fPrice = fPrice;
    product.fDescription = fDescription;
    product.fQuantity = fQuantity;

    /*如果有新的檔案上傳，就要把之前的刪掉再新增*/
    if (file != null && file.ContentLength > 0)
    {
        if (System.IO.File.Exists(path))
            System.IO.File.Delete(path);

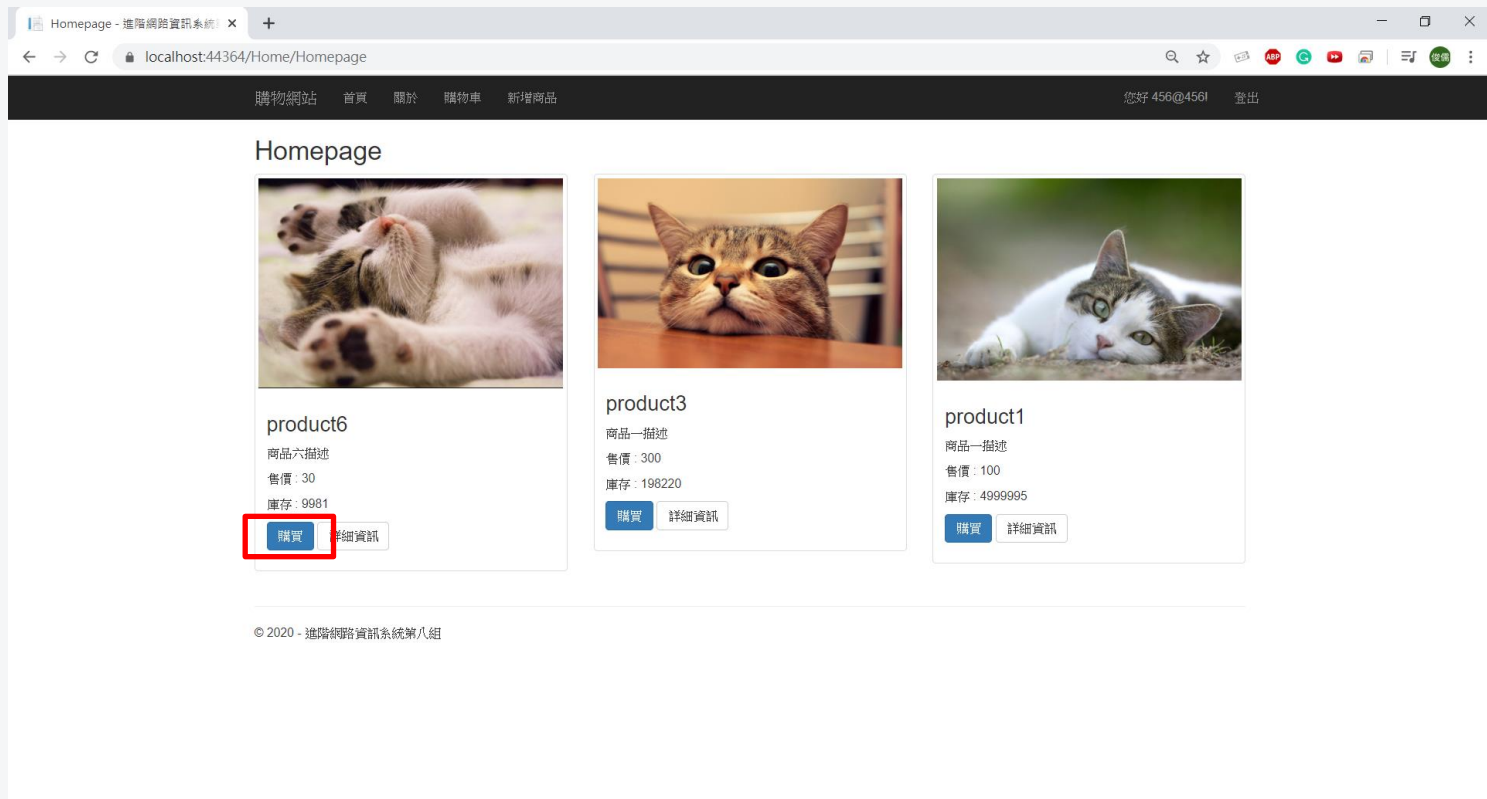
        string fileName = fName + Path.GetExtension(file.FileName);
        product.fImage = fileName;
        path = Path.Combine(Server.MapPath("~/Images"), fileName);
        file.SaveAs(path);
    }

    db.SaveChanges();

    return RedirectToAction("Index");
}
```

購物流程-購買按鈕

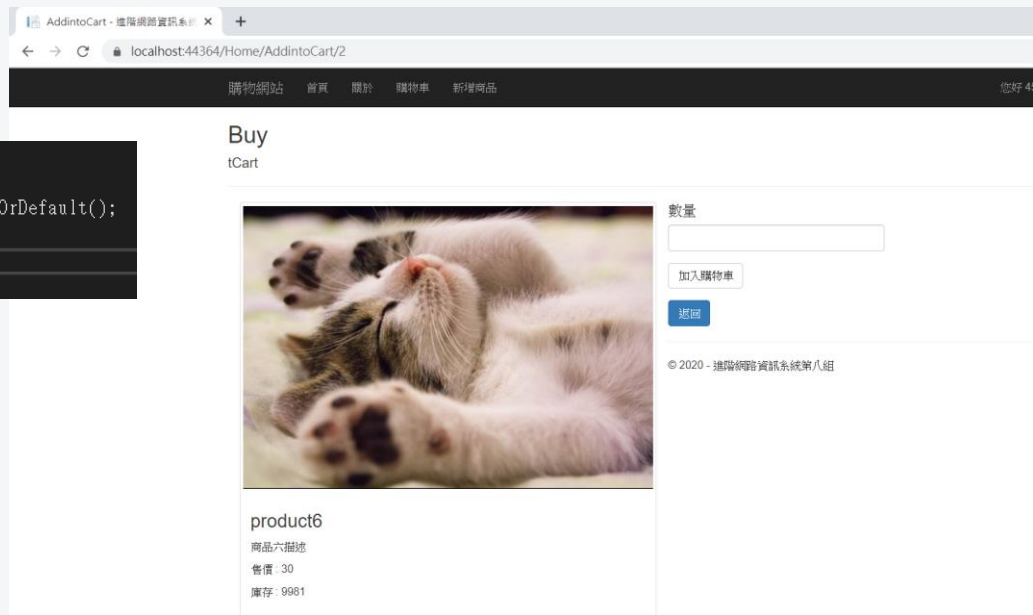
購買按鈕



購物流程-加入購物車

當使用者按下購買按鈕時，會經由Controller，跳到AddintoCart這個Action，這個Action有一個參數”id”，用來接收商品的id，並回傳一個View檢視。

```
153 public ActionResult AddintoCart(int id)
154 {
155     var product = db.tProducts.Where(m => m.fId == id).FirstOrDefault();
156     return View(product);
157 }
158
```



購物流程-加入購物車

這個檢視沒有使用範本，使用資料模型
tProducts用來顯示商品資料→

當要顯示要購買的商品資料時，會使用
Razor 語法↓

```

7      <h2>Buy</h2>
8      <h4>tCart</h4>
9      <hr />
10     @Html.ValidationSummary(true, "", new { @class = "text-danger" })
11     <div class="col-md-6">
12         <div class="thumbnail">
13             
14             <div class="caption">
15                 <h3>@Model.fName</h3>
16                 <p>@Model.fDescription</p>
17                 <p>售價 : @Model.fPrice</p>
18                 <p>庫存 : @Model.fQuantity</p>
19             </div>
20         </div>
21     </div>
22 </div>

```

```

@model finalProject_V1.Models.tProducts
@{
    ViewBag.Title = "AddintoCart";
    Layout = "~/Views/Shared/_HomeLayoutPage.cshtml";
}

```



購物流程-加入購物車

在AddintoCart這個View檢視中，有可以讓購買人填寫購買數量的欄位，底下有加入購物車按鈕，當填寫完欄位按下加入購物車的話，會藉由Controller跳到另一個View檢視並接收商品id和數量。

```
24 @using (Html.BeginForm(Html.BeginForm("AddintoCart", "Home", FormMethod.Post, new { enctype = "multipart/form-data" })))
25 {
26     @Html.AntiForgeryToken()
27     <div class="form-group">
28         <h4>數量</h4>
29         <div>
30             <input type="text" name="fQuantity" class="form-control" required="required" />
31         </div>
32     </div>
33
34     <div class="form-group">
35         <div class="col-md-offset-2">
36             <input type="submit" value="加入購物車" class="btn btn-default" />
37         </div>
38     </div>
39 }
40 }
```

數量

加入購物車

返回

© 2020 - 進階網路資訊系統第八組

數量

加入購

! 請填寫這個欄位。

返回

© 2020 - 進階網路資訊系統第八組


購物流程-填寫收件資訊

當Controller內的AddintoCart接收到商品id及數量時，會立即將您所購買的商品和數量加到tCart這個資料表內，並透過[httpPost]方式重新導向到ShowCarts這個Action。

i64/Home/ShowCarts

購物網站 首頁 關於 購物車 新增商品 您好 456@456 登出

ShowCarts

商品名稱	商品圖片	商品描述	數量	金額
product6		商品六描述	120	3600
			總金額	3600

收件人姓名

收件人電話

收件人地址

付款方式

© 2020 - 遠端網路資訊系統第八組

```

159
160
161
162
163
164
165
166
167
168

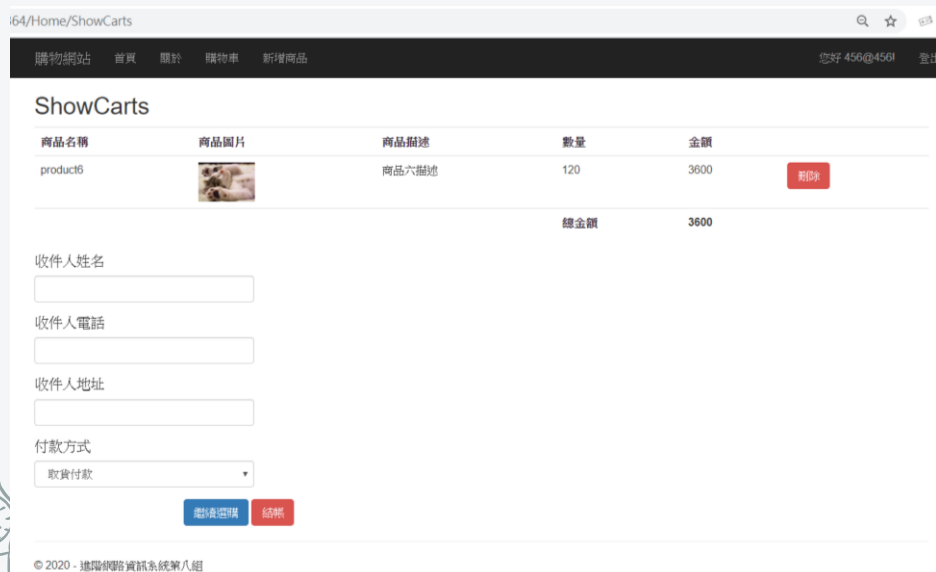
[HttpPost]
0 個參考
public ActionResult AddintoCart(int id, string fQuantity)
{
    tCart cart = new tCart();
    var product = db.tProducts.Where(m => m.fId == id).FirstOrDefault();
    cart.fUserId = User.Identity.GetUserId();
    cart.fPdImage = product.fImage;
    cart.fPdName = product.fName;
    cart.fPdDescription = product.fDescription;
    cart.fPdPrice = product.fPrice * Convert.ToInt32(fQuantity);
    cart.fBuyQuantity = Convert.ToInt32(fQuantity);
    product.fQuantity -= Convert.ToInt32(fQuantity);
    db.tCart.Add(cart);
    db.SaveChanges();

    return RedirectToAction("ShowCarts");
}


```

購物流程-填寫收件資訊

而ShowCarts這個View檢視套用list範本，使用tCart (購物車) 資料模型，下方也有填收件資料的欄位，當填完之後按下結帳，會將使用者填的資料透過[httpost]方式傳到PayMoney這個Action。



The screenshot shows a web browser window with the URL `i64/Home/ShowCarts`. The page has a navigation bar with links: 購物網站, 首頁, 關於, 購物車, 新增商品. The main content area is titled "ShowCarts" and displays a shopping cart table.

商品名稱	商品圖片	商品描述	數量	金額
product6		商品六描述	120	3600
			總金額	3600

Below the table, there is a form for checkout information:

收件人姓名

收件人電話

收件人地址

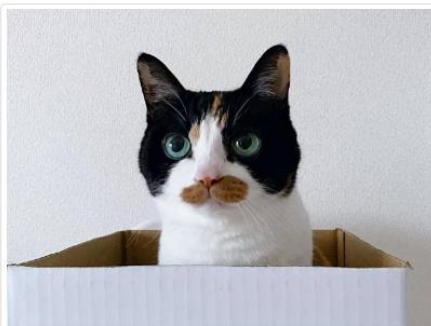
付款方式
取貨付款

Buttons: 刪除全部, 結帳

Footer: © 2020 - 遠端傳信資訊系統第八組

購物流程-購物車刪除

如果不要購物車內的商品，可以按下刪除按鈕，此時會透過DeleteCartItem這個Action接收id這個參數，刪除購物車資料表 (tCart) 內的此項目，而商品庫存也會隨之變動。



product7

商品七描述

售價：700

庫存：100

原本的庫存

購買

詳細資訊



product7

商品七描述

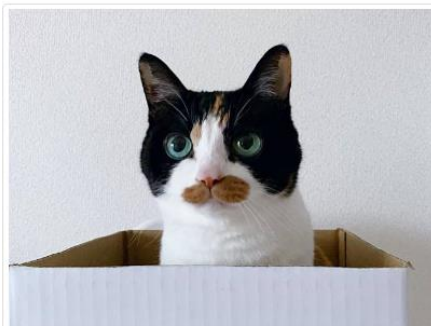
售價：700

庫存：50

按下購買後

購買

詳細資訊



product7

商品七描述

售價：700

庫存：100

從購物車刪除後

購買

詳細資訊

購物流程-完成購物



PayMoney這個Action接收到前端傳送過來的資料後，會立刻把購物車的內容和收件資訊存到tOrder這個資料表內，並重新導向至BuyList這個Action。

```

196 [HttpPost]
197 0 個參考
198 public ActionResult PayMoney(string Name, string Phone, string Address, string Payway )
199 {
200     tOrder order = new tOrder();
201     var carts = db.tCart.ToList();
202     foreach (var item in carts)
203     {
204         if( User.Identity.GetUserId() == item.fUserId)
205         {
206             order.fBuyQuantity = item.fBuyQuantity;
207             order.fPdDescription = item.fPdDescription;
208             order.fPdImage = item.fPdImage;
209             order.fPdName = item.fPdName;
210             order.fPdPrice = item.fPdPrice;
211             order.fUserId = User.Identity.GetUserId();
212             order.fUserName = Name;
213             order.fUserPhone = Phone;
214             order.fUserAddress = Address;
215             order.fUserPayway = Payway;
216             db.tOrder.Add(order);
217             db.tCart.Remove(item);
218             db.SaveChanges();
219         }
220     }
221
222     return RedirectToAction("BuyList");
223 }
    
```

BuyList會顯示使用者所購買的所有東西

fPdName	fPdImage	fPdDescription	fBuyQuantity	fPdPrice
product16		商品六描述	120	3600
		商品六描述	123	123

取貨付款

© 2020 - 遠傳電信資訊系統第八組

2.

網頁展示





Thanks!

