**動態網頁程式設計期末專案**

**中興大學訂餐系統**

**第17組**

**4108056018 資工四 王旻玄**

**4108056031 資工四 盧悅盛**

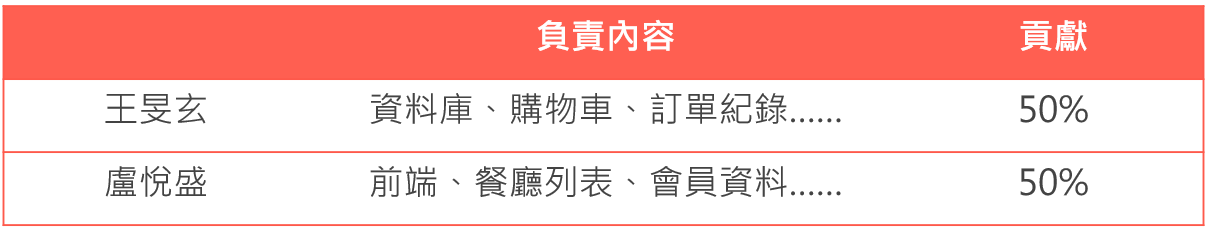
目錄

1. 動機
2. 分工
3. 介紹
4. 程式碼說明
5. 心得
6. 動機

每當中午下課時間，學校餐廳就會一次湧入許多人潮，店家也會一次性收到多筆訂單，導致店家過於忙碌，來不及消化訂單，學生也必須等待非常久的時間，因此我們想建立一個學校餐廳的訂餐系統，讓學生能夠提前訂餐，方便店家分配與管理時間，也節省學生的等待時間。我們透過圖形化的介面與資料庫連接產生訂單，商家也有屬於自己的帳號來管理訂單。

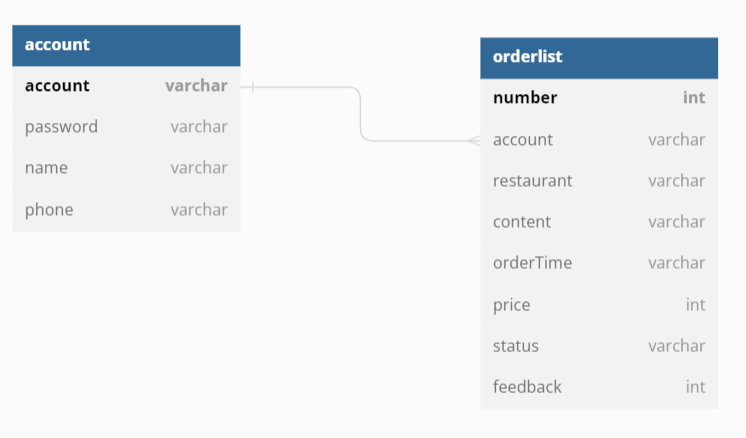
1. 分工

以下是我們的工作分工內容，基本上我們並沒有將工作分那麼開，有那裡需要支援我們就會去支援對方，互相寫Function給對方使用。



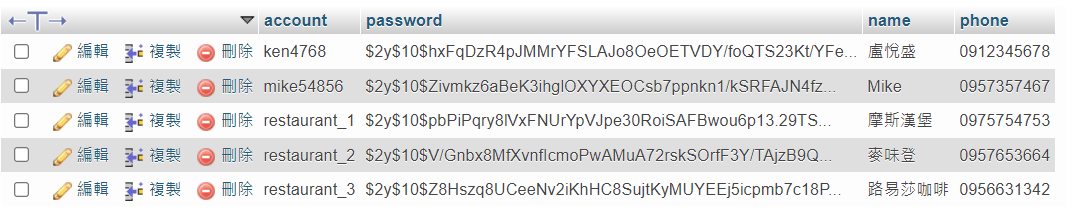
1. 介紹

以下是我們的relational schema，我們分成account與orderlist兩個Table，account用來儲存帳號的資訊，其中account是primary key，不可重複，包含帳號、密碼(加密後)、名字、手機。

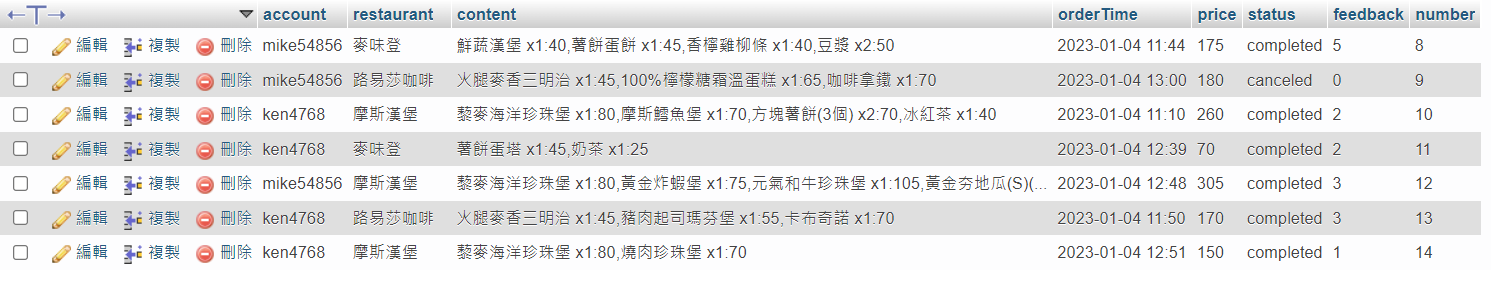
****

而orderlist是用來儲存訂單，number是auto increment的primary key，代表訂單的流水號，每筆訂單都是獨特的，而account會參考account table中的account，restaurant代表餐廳的名字，content代表訂購的內容，orderTime是預定取餐的時間，格式為YYYY-MM-DD mm:ss，而price代表訂單的總金額，status是訂單的狀態，有preparing、canceled、completed三種狀態，feedback則是訂餐人對於該筆訂單的評價，為1到5分。

content中的編碼方式為以下，[商品名稱] x[商品數量]:[商品金額]，中間會用逗號來將不同的商品隔開，下圖為資料庫中的資料。

****

account Table

****

orderlist Table

網頁的登入狀態我們可以區分為以下三種：

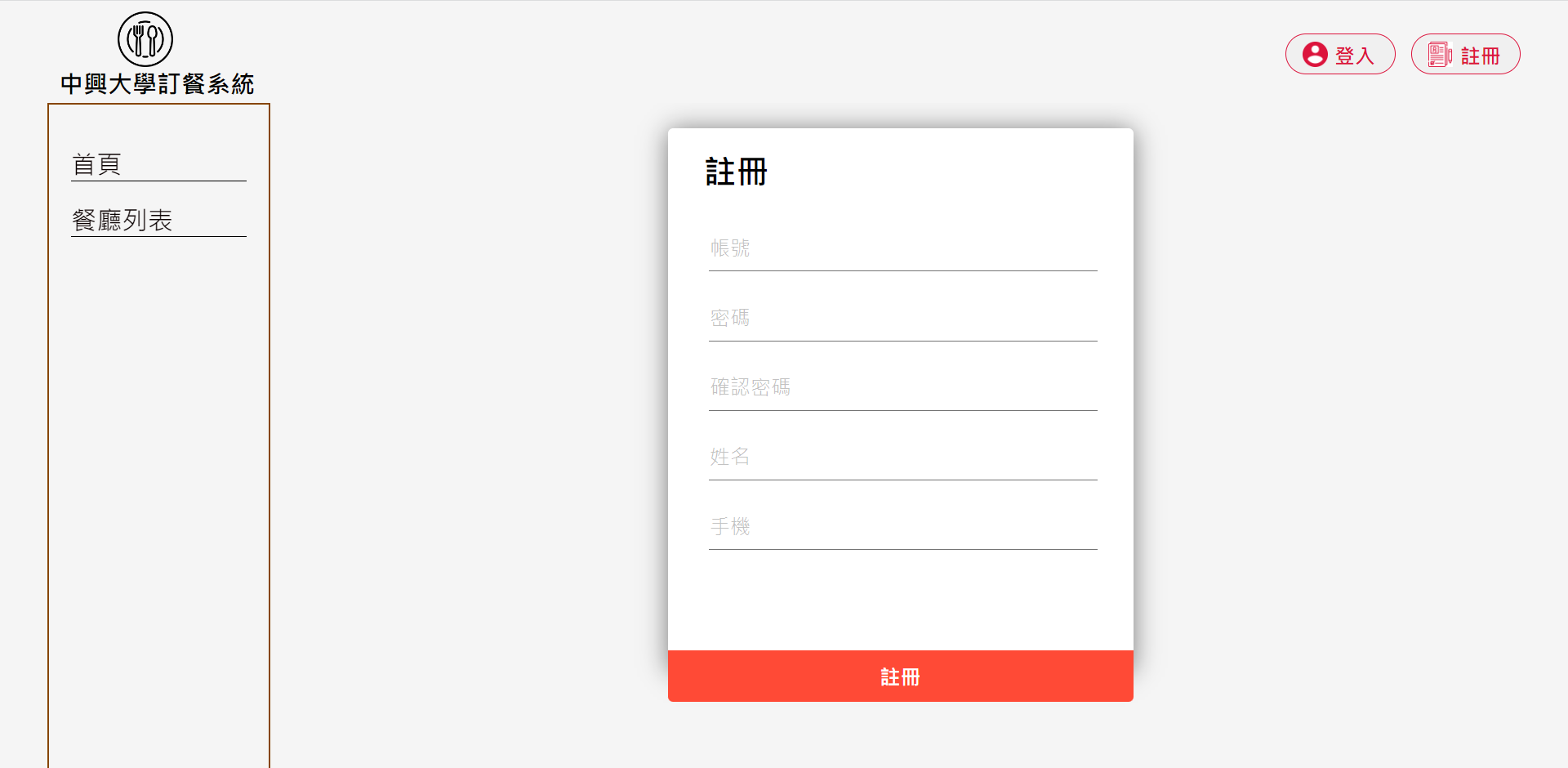
* 未登入
* 一般使用者登入
* 商家登入

而各個狀態能夠造訪的網頁都有所不同：

* 未登入：首頁、登入、註冊、餐廳列表
* 一般使用者登入：首頁、餐廳列表、會員資料、購物車、訂單紀錄
* 商家登入：首頁、訂單管理

如果使用者硬是透過輸入網址的方式去造訪無法進入的頁面，那麼程式會進行自動判斷，並重新導回首頁，此外程式也會去顯示獲刪除特定元素，例如，在未登入狀態下，頁面的右上角應該要顯示登入與註冊，而在登入狀態下，頁面的右上角應該要顯示登出，這些都會在載入頁面的時候根據登入狀態進行修改。

以下是註冊的頁面，註冊時會去資料庫檢查帳號是否已經被使用過，密碼與確認密碼必須相同，姓名不能為空，而最後的手機必須要是10碼的數字。



以下是登入的頁面，按下登入後會將帳號與密碼傳送到後端，再與資料庫進行比對，資料庫中的密碼是加密過的，可以避免伺服器被攻擊後外洩的密碼是密文。

****

接下來我們來看一下首頁，下方是未登入狀態下的首頁，右上方有登入與註冊，而左邊只有首頁與餐廳列表。

****

下方是登入一般會員狀態下的首頁，右上方有登出，而左邊多了會員資料購物車與訂單紀錄，左下角還會記錄購物車目前的總金額。

****

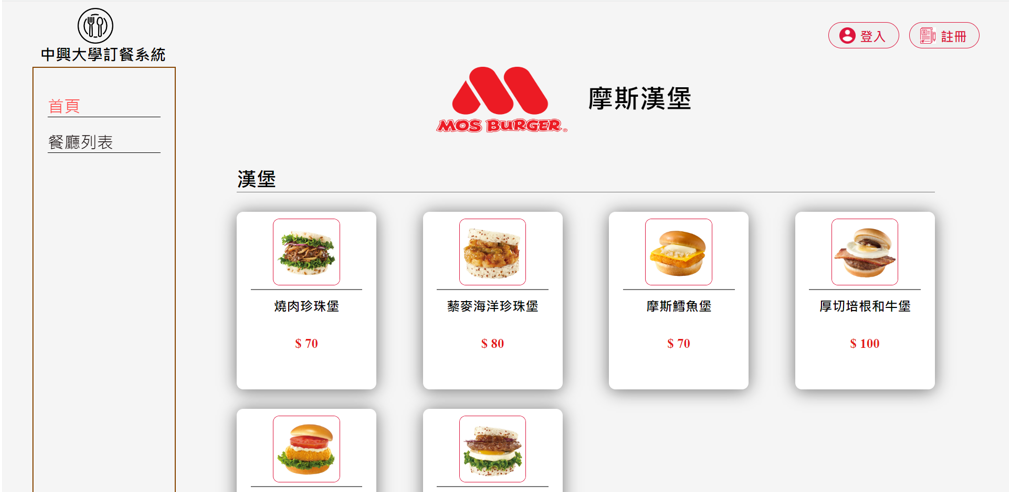
如果登入的是商家帳號，右上角會變成登出，而左邊只會有首頁與病單管理。

****

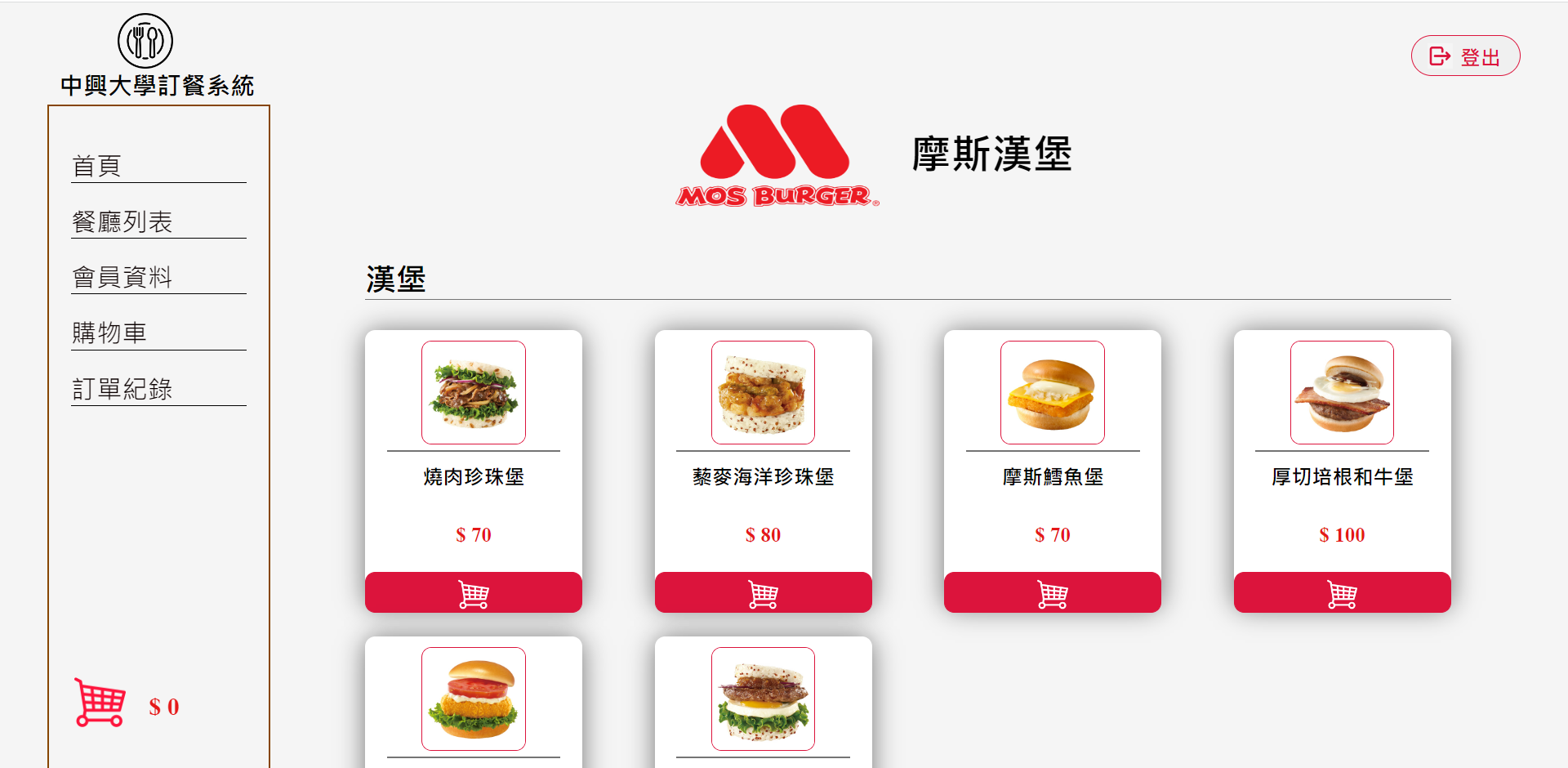
在餐廳列表中，我們可以看到全部的餐廳以及他們的評價，評價滿分為5分，右邊會顯示有幾個評價數量。

****

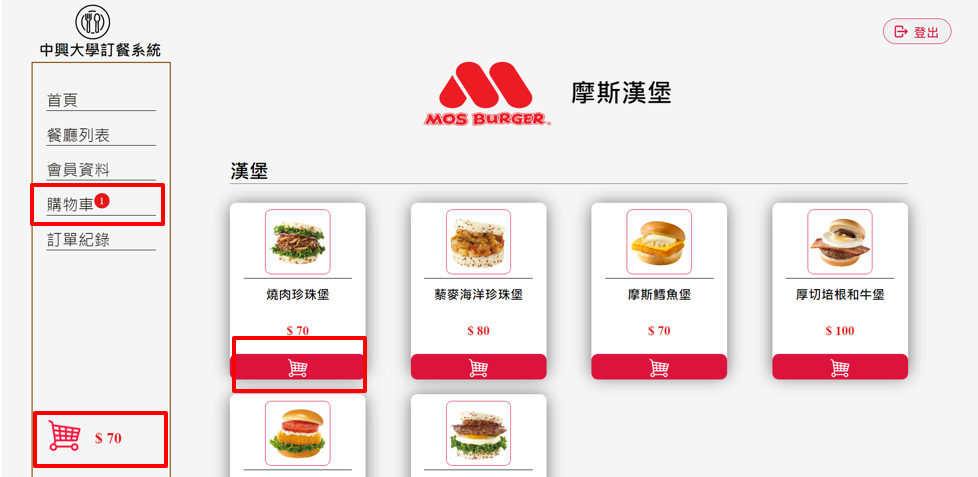
如果我們再進一步點選餐廳，就能夠看到餐廳所販賣的食物，但如果是未登入狀態，就無法進行訂購。

****

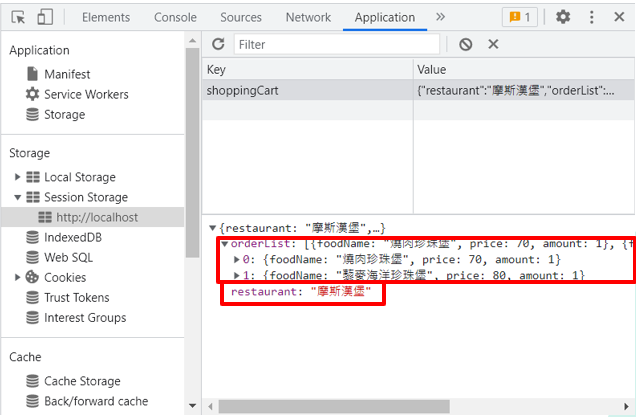
如果是以登入狀態下去點選餐廳，就可以將物品放到購物車中，如下圖。

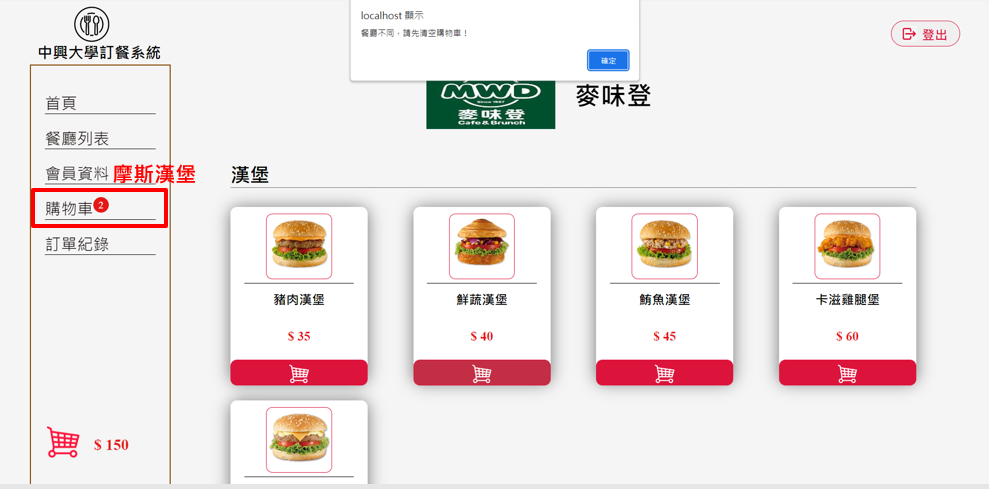
****

假設我們點選燒肉珍珠堡，那麼左方的購物車會多一個數字1，代表目前有一個品項，而左下方的購物車金額變成70，代表總金額是70元。

****

購物車中的資料，我們是存在session storage中，key稱作shoppingCart，有兩個資料，restaurant代表目前選定的餐廳名稱，而orderList則是選購的物品，每個物品有3個資料，分別是品項名稱，價格與數量，我們會讀取這些資料來更新購物車中的內容，以幾網頁左下角的購物車總金額。

****

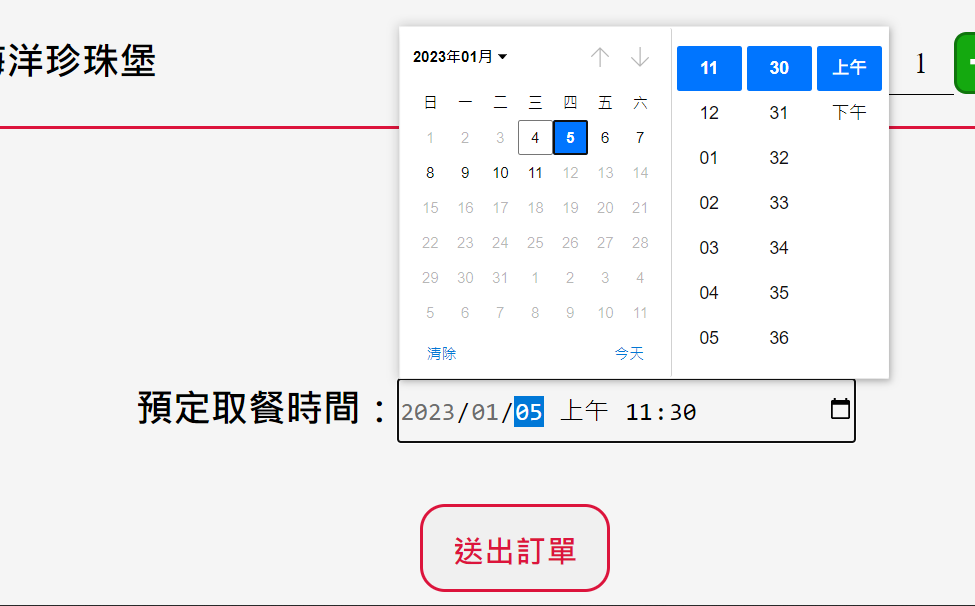
****

我們假設購物車中的物品是摩斯漢堡的，這時候如果我們再點選麥味登的物品，那麼就會跳出警告視窗，會要求先清空購物車。

選擇完餐店後我們可以到購物車中做品項以及數量的確認，若想要換其他餐廳，也必須在這裡將購物車清空，最後我們可以設定取餐時間把訂單送出。

****

取餐時間必須要是現在的時間之後，不能設定成過去的時間，再來我們規定取餐時間只能訂在7天內，而商家的營業時間我們設定為上午11點到下午8點，非營業時段也不能夠點餐。

****

當我們把訂單送出去後就能夠去訂單紀錄查看訂單的狀況，目前的狀態還是preparing，代表店家還在準備中。

****

我們用商家帳號去管理這筆訂單，比起一般使用者，商家看這筆訂單時多了訂購帳號、訂購人名字、訂購人電話等資訊，當訂購人取餐並完成付款後，商家就能按下完成訂單，若商家發現該訂單有問題，則可以取消訂單。

****

最後訂購人可以回到訂單紀錄這邊去確認訂單，並給予評價。

****

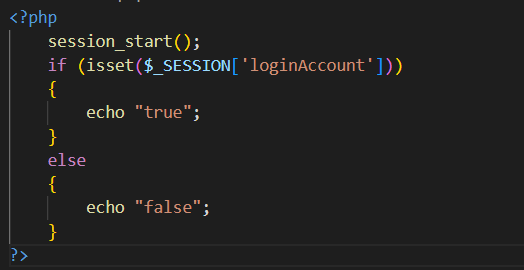
我們也提供會員修改資料的功能，灰色的字代表的是原本的資料。

****

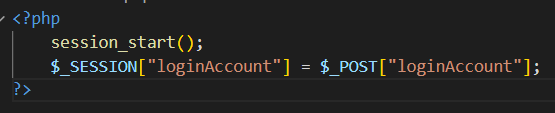
1. 程式碼說明

由於程式實在是太多了，有上千行，因此會挑幾個區塊進行說明。

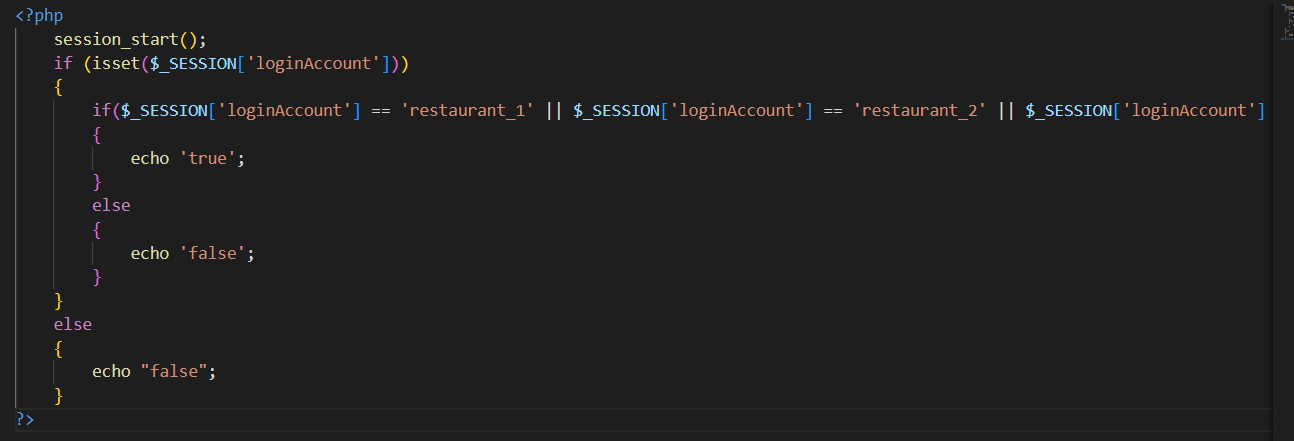
以下是後端中的登入檢查，會看loginAccount是否被設置。



以下是後端中的登入設定，去設定loginAccount。



以下是後端中檢查登入的帳號是否為商家帳號。



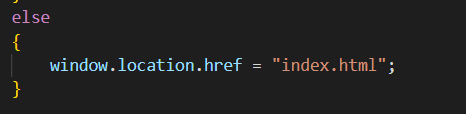
我們之前有提到，每個頁面會根據登入狀態的不同而呈現不同的東西，而每個頁面都有一個類似於下方的檢查程式碼，如果登入的話，就會把登入與註冊的按鈕刪掉，並且查看購物車是否為空，如果有東西，那麼購物車旁邊就要提示目前有幾個品項，並且算出總金額，更新右下角的金額。



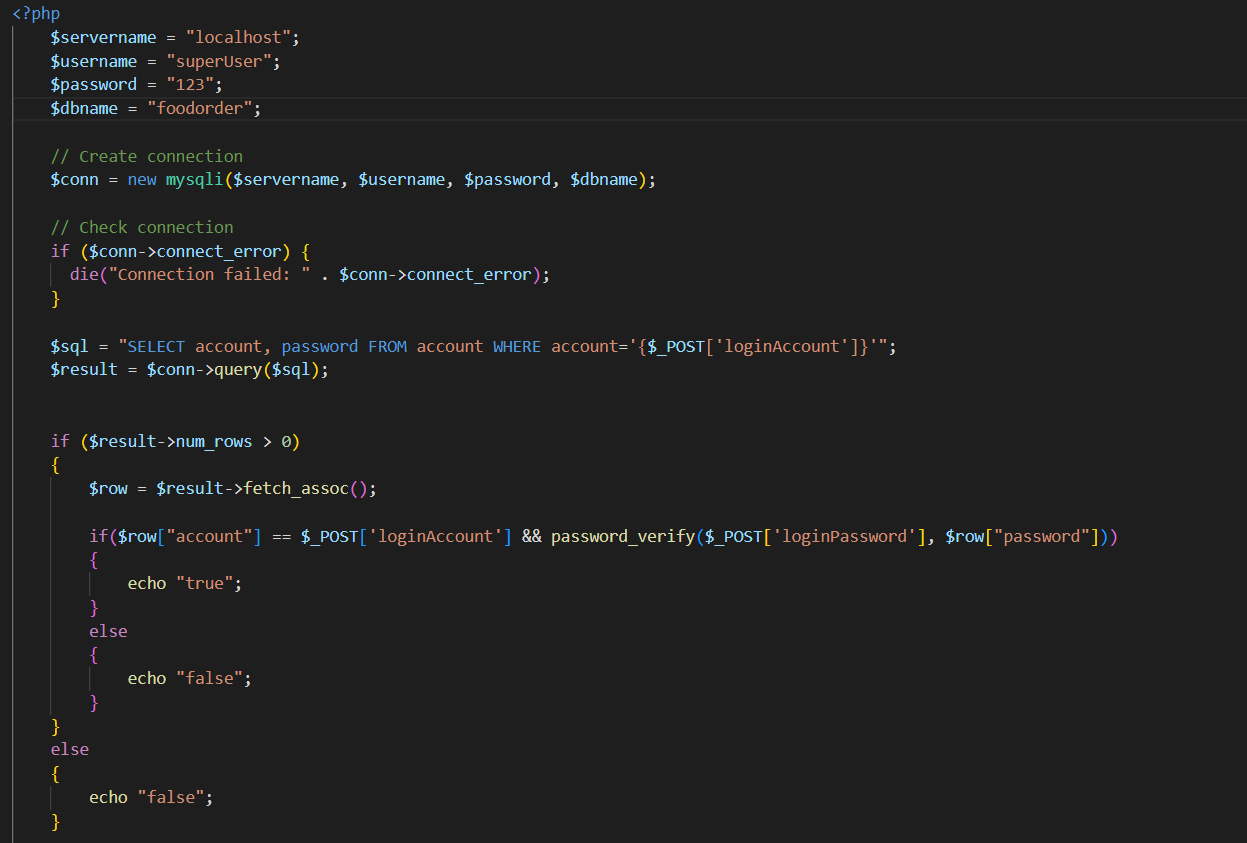
如果沒有登入的話就會刪除掉一些元素，例如登出按鈕、品項數量提示、購物車金額提示等等。



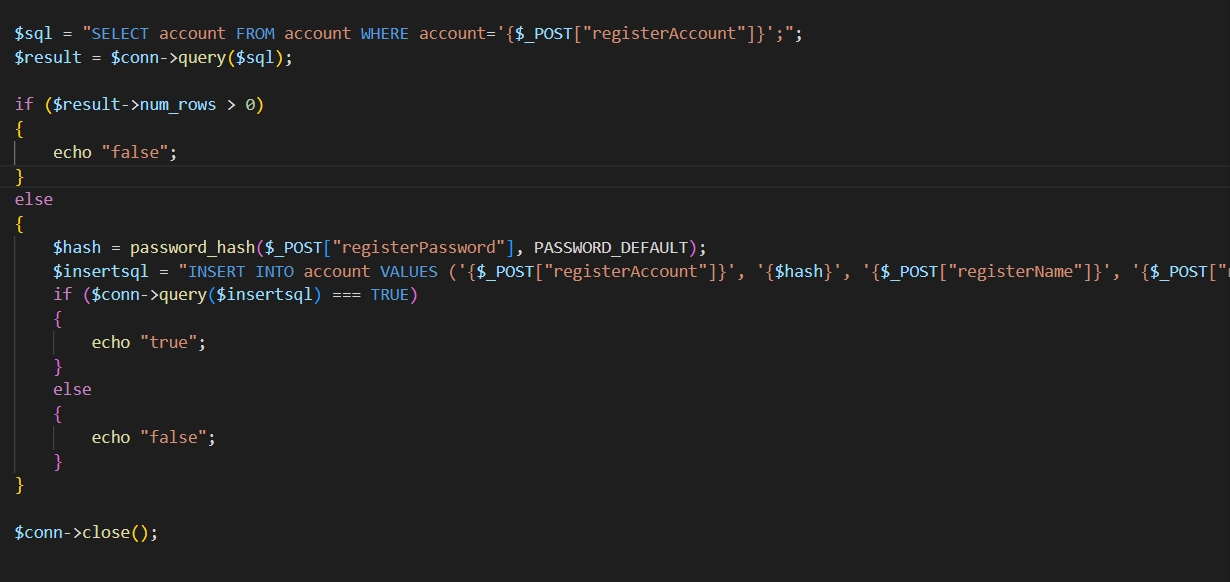
如果進到一些不該進到的頁面，則會被強制重新導向首頁。



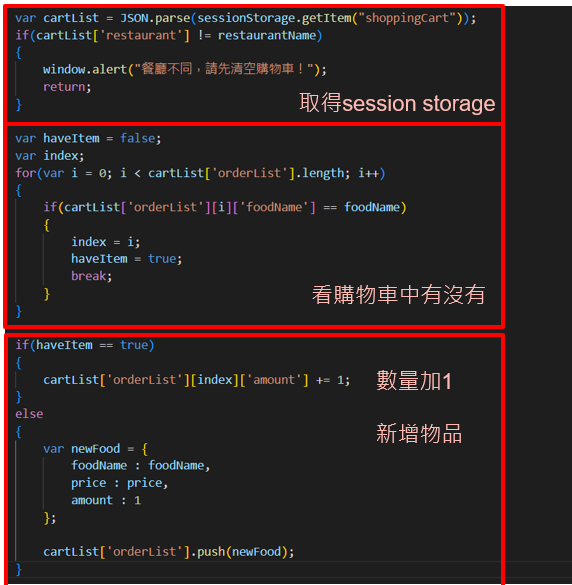
以下是登入在購端的檢查，會根據傳上來的登入帳號去資料庫中獲取密碼，並進行比對。



以下是註冊的後端程式，會先查看資料庫中是否有該帳號，如果沒有的話就會將這組帳號密碼Insert到資料庫中。



接下來我們看一下購物車方面的程式，下方的程式是將物品加入倒購物車中，一開始會先取得session storage中的資料，之後會進行一個iteration去找到物品在orderList中的位置，如果orderList中已經有該物品，那麼就將其數量加一，反之則新增一個物件進去。



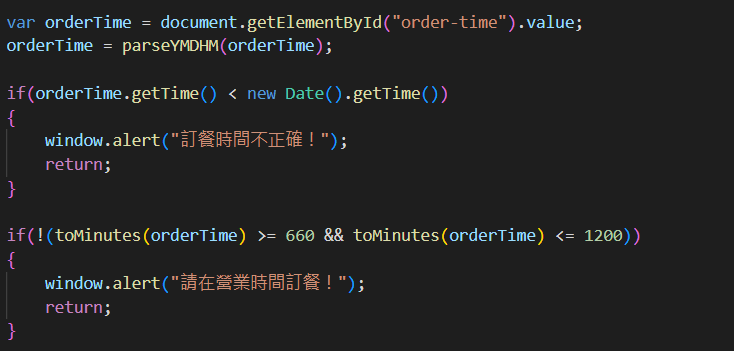
再來會將資訊設定回session storage中，並且根據目前session storage中的資料顯示總價。



下方是編輯購物車的程式，將以增加一個商品數量來舉例，一開始會先取得顯示商品數量的物件，再來一樣會從orderList中去尋找商品的索引，之後讓數量加一，存回session storage中，並讓顯示的金額更新。如果是減少一個商品的程式，則會檢查其商品數量是否已經到1了，是的話就直接return不減少。



下方是檢查訂餐時間的程式，時間必須在現在之後，而且訂購的時間必須要在上午11點到晚上8點。



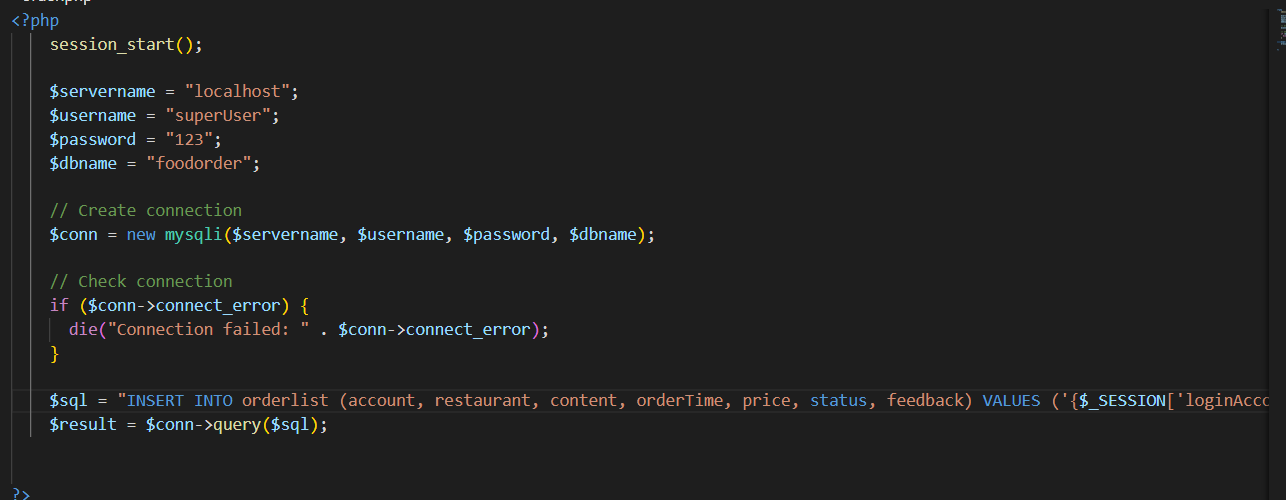
當訂餐時間正確，接下來會把預定的時間編成字串，之後算出總金額，最後會將購物車中的商品編成以下的格式[品項] x[數量]:[金額]中間用逗號隔開，之後我們在訂單紀錄這邊會再根據逗號進行split。



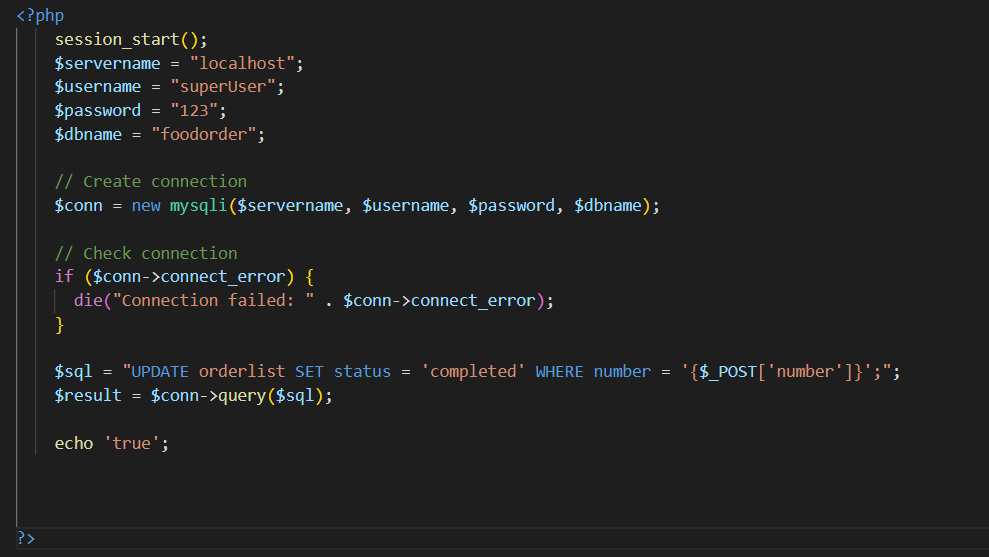
訂單的編成指定格式後，就會呼叫order.php去進行下訂，傳入訂購餐廳、訂購內容、預定取餐時間、總金額。



Server會再根據送上來的資料Insert到orderlist這個Table中，然後訂單的狀態變成preeparing。



商家如果按下完成訂單，則會去更新這個訂單狀態，變成completed，取消訂單的話則是變成canceled。



下方的程式是透過sql來取得特定商家的評分，之後編成JSON的格式回傳。



下方的程式是用在餐廳列表的評分中，會將餐廳的名稱傳到後端，之後再算出平均的分數，如果沒有評分，則將評分設為0。



1. 心得

以前我們從沒有接觸過網頁相關的課程，這次在這堂課中學到了html、css 、javascript 、php，等前端的網頁美化與後端的資料溝通技術。雖然每次作業都需要花非常多的時間，但寫了這幾次作業真的幫助很大，會根據想要的功能去進行搜索與實驗，到最後變成自己的知識與經驗。

雖然網頁中還有許多框架跟套件沒有接觸到，還有最重要的安全性問題我們目前也沒有考慮很多，但我們已有能力可以從0寫出一個網頁，雖然花得時間有點久，但也順利完成期末專案的開發，建構出我們想要的功能。