list1.sql : 存放客戶購物車的內容(僅為臨時，顧客結完帳則清空)

包含pId(商品編號)、pName(商品名稱)、pNum(購買的商品數量)、 pPrice(商品單價)

List2.sql : 存放管理端商品的內容

包含id(商品編號)、name(商品名稱)、number(商品庫存數量)、 price(商品單價)、orders(訂單數量)

conndb.php : 連接資料庫

UI1.html : 一開始的頁面，選擇客戶端或管理端

客戶端:

client.php : 客戶端的頁面，包含印出商品清單，以及可讓客戶將商品加入購物車或修改，以及列出目前購物車的內容

addcart.php : 操作客戶端加入購物車資料庫(list1)的部分，購物車頁面顯示還是在client.php (若超出庫存數量或購物車有的重複品項，則跳回client頁面重新購買)

updatecart.php : 操作顧客更新購物車資料庫(list1)的部分

deletecart.php : 操作顧客刪除購物車資料庫(list1)品項的部分

pay.php : 顧客確認購物車進行結帳，此時會列出應付金額及購物車內容，但若想回去修改購物車也可再回前頁選擇修改，或是確認結帳

end.php : 顧客結完帳後結束購物的頁面，此時會將這筆訂單加入到管理端資料庫(list2),並將客戶端資料庫(list1)清空,以便新的顧客再來購買

管理端:

manage.php :管理端頁面，會印出商品清單和訂單，可操作更新清單，庫存和出貨，若加入新產品為已經有的品項或庫存不足以出貨，則跳回manage.php重新操作(會造成庫存不足是因為管理端一直不出貨而累積太多訂單，並非單筆顧客下單數量超過庫存)

newproduct.php : 管理端加入新品項的部分，僅是操作資料庫(list2)的變更，實際頁面顯示仍在manage.php

updatelist.php : 管理端更新商品清單的部分，包含更改名稱、價格、庫存，也是操作資料庫(list2)的變更

addproduct.php : 管理端入庫商品數量的部分，操作資料庫(list2)的變更，會將現有庫存加上剛剛輸入的進貨數，即為新的庫存總數

order.php : 管理端將顧客訂單出貨的操作，會扣掉資料庫(list2)的訂單數和現有庫存