

# 逢甲大學學生報告 ePaper

## 資料庫專題實作-二手書交易平台

### Database System Seminar Project - Used book trading platform

作者：施冠彰, 丁泓哲, 戎宥杰, 林永吉, 晁文勤

系級：資訊二丙

學號：D0745925, D0745867, D0783026, D0783056, D0746129

開課老師：林明言

課程名稱：資料庫系統

開課系所：資訊工程系

開課學年：108 學年度 第 2 學期



## 中文摘要

### (1) 目的

大學生每年都得為新的課堂添購課本，但新課本價格昂貴，甚至課程結束後，難有機會再次使用它。因此，我們希望建立一個平台，讓第一次上新課程的同學，可以透過我們的平台，用較低的價格買到二手教科書之餘，也可以讓已經上完課的同學，透過我們的平台，將課本轉售出去。如此一來，可以紓解大學生買課本的經濟壓力之外，還可以達到環保愛地球的功效。

### (2) 過程及方法

本次專題首先是用畫 ER-MODEL 設計資料庫的分類和內容，接著是將資料正規化，避免設計不周全的錯誤，在規劃完善之後，即開始在電腦主機上做實作，以及匯入必要之資料進資料庫，主要著重使用 SQL、XAMPP 軟體、HTML、CSS、JavaScript 與 PHP 等程式語言撰寫，最後將完成的成果架上網站。

### (3) 結果：

我們打造出了二手書交易平台，已經可以實現書本交易、上架書本、管理上架書本、管理交易紀錄與系統後台管理平台的功能。雖然還存在一些問題導致無法立刻上線，但我們會繼續努力，將該平台的功能完善後上線經營。

**關鍵字：**PHP、程式語言、資料庫系統、電商平台開發、

網頁系統開發

# **Abstract**

## **Purpose**

**Every year, College students have to buy textbooks for a new courses, but the new textbooks are too expensive, and even in the end of courses, it is difficult to have the opportunity to use them again.**

**Therefore, we hope to build a platform so that students who are taking a new course for the first time can buy second-hand textbooks at a lower price through our platform, and students who have already finished the courses can also use our platform to resell their textbooks.**

**In this way, it not only relieve the financial pressure of college students to buy textbooks, but can also achieve the effect of environmental protection and love for the earth.**

## **Process and Approach**

**First, this project is using ER-MODEL to design the classification and content of the database, and then to normalize data to avoid the error of design incompleteness. After finishing the design, we start to work on computer host and put necessary data into database. We primarily focus on using programming language like SQL, XAMPP, HTML, JavaScript, CSS, and so on. Finally, we put the completed result on the website.**

## **Result**


**We have already built a second-hand transaction platform that can accomplish functions of book transaction, book shelf management, transaction record management and system background management platform. Although there are still some problems which hinder the platform from working on the Internet immediately, we will continue to work hard to improve the functions of the platform and make it work online.**

## **Keyword**

**PHP programming language, database system, e-commerce platform development, web system development**

## 目 次

主題	4
目的	4
系統使用者	4
使用者需求功能與輸入資料、輸出資料	4
ERMODEL	7
Relation Schema	8
資料表	8
SQL 建立 TABLE 指令	11
系統功能	15
組員心得	41



1. 主題：二手書媒合交易平台

2. 目的：讓學生可以以更便宜的價格買到教科書，同時舊書可以有一個拍賣的平台，不僅能方便買賣雙方，節省時間成本，更能夠清楚知道商品狀態。此外，也可達到舊書再利用、環保的目的。當作 C2C 平台，直接媒合買賣雙方。

3. 系統使用者

- 大學生
- 二手書店

4. 使用者需求功能與輸入資料、輸出資料（顯示資料）

● 買家

功能	輸入資料	輸出資料
註冊	輸入會員註冊之基本資料（名字/性別/暱稱/生日/學生證驗證/偏好面交地點/E-mail/帳號/密碼/學校/學生證照片/學號/系所）	顯示獲得會員使用資格
更新/新增會員資料	輸入會員註冊之基本資料（名字/性別/暱稱/生日/學生證驗證/偏好面交地點/E-mail/帳號/密碼/學校/學生證照片/學號/系所）	儲存資料至資料庫

登入	輸入帳號、密碼	成功登入或是登入失敗
找回密碼(忘記密碼)	輸入信箱、更改後的新密碼	寄發驗證信到用戶信箱，確認身分，成功更改密碼
查詢書籍	欲查詢的商品基本資料 (書名、ISBN、標籤分類)	顯示符合規定之書籍
查詢訂單交易紀錄	可用訂單編號進行搜尋	顯示交易紀錄列表

## ● 賣家

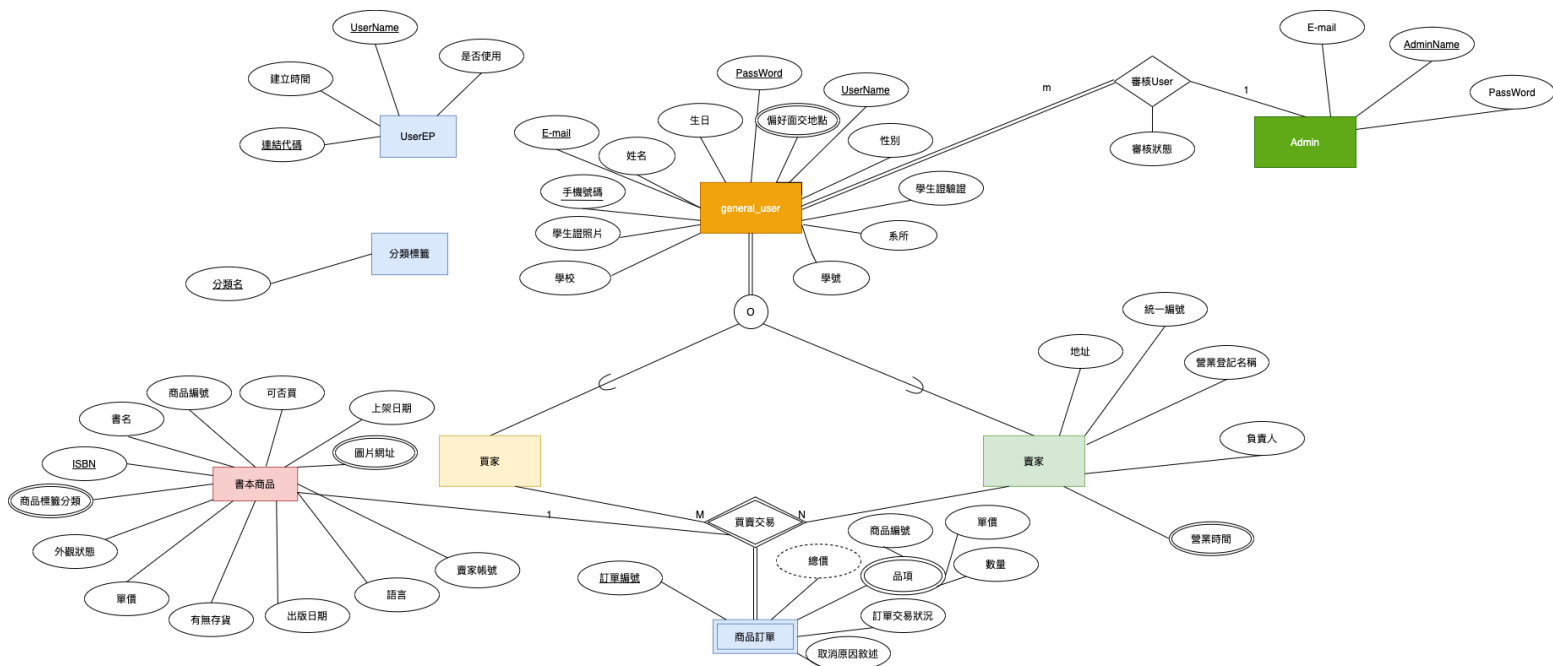
功能	輸入資料	輸出資料
更新/新增會員資料	輸入會員註冊之基本資料 ( 名字/性別/暱稱/生日/學生證驗證/偏好面交地點/E-mail/帳號/密碼/學校/學生證照片/學號/系所)  如果是店家增加輸入( 地址/統一編號/營業登記名稱/負責人)	儲存資料至資料庫
新增/修改商品資料	欲更動或新增的商品基本資料 ( 書名/ISBN/商品標籤分類/外觀狀態/	發布於平台刊登商品介面

	單價/存貨量/出版日期/ 語言/圖片 )	
查詢訂單交易記錄	可用訂單編號進行搜尋	比對結果後顯示於畫面上
登入	輸入帳號、密碼	成功登入或是登入失敗
找回密码(忘記密碼)	輸入信箱、更改後的新密碼	寄發驗證信到用戶信箱，確認身分，成功更改密碼

## ● 管理員

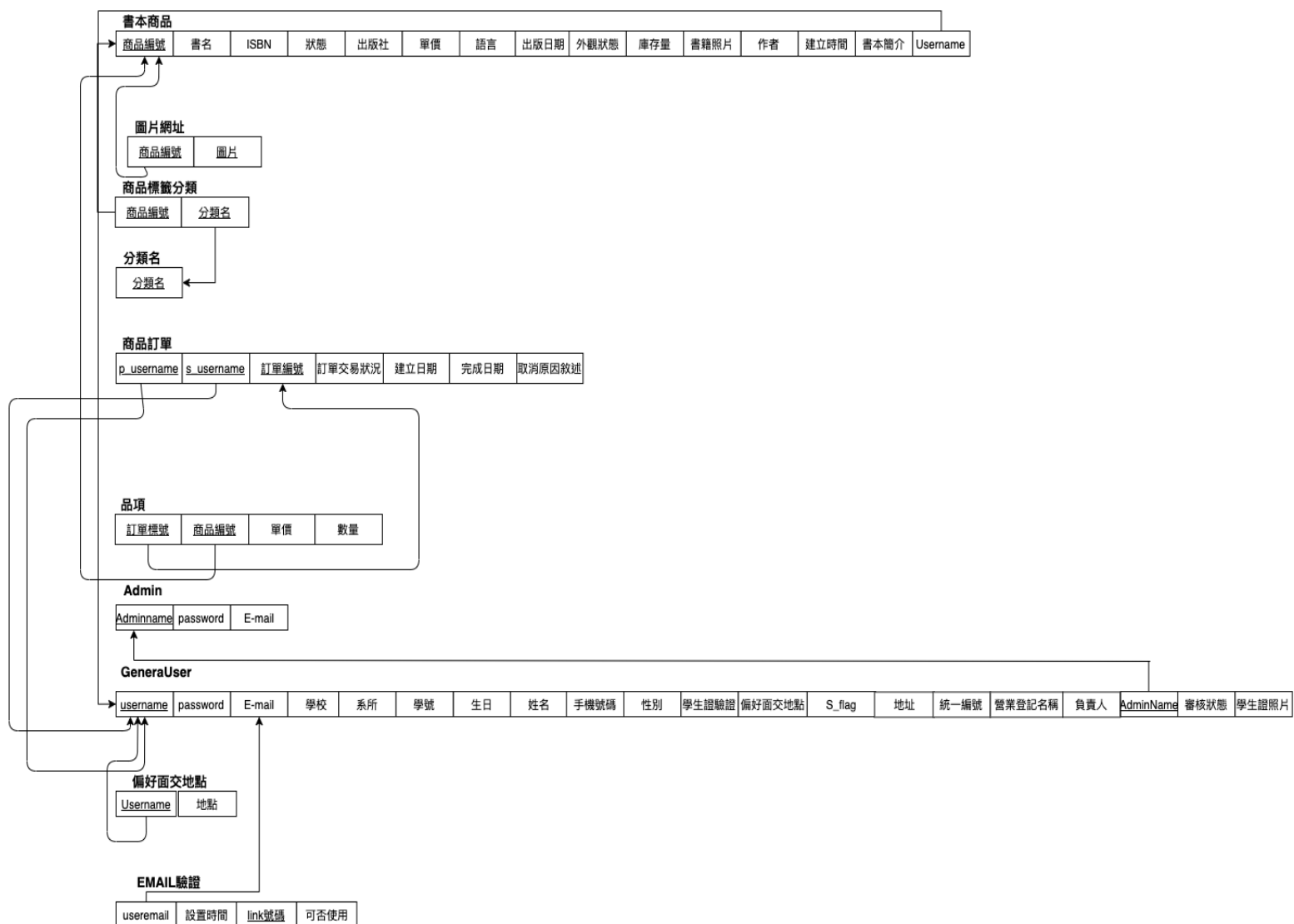
功能	輸入資料	輸出資料
新增/修改/刪除學校	學校名稱	成功新增/修改/刪除
新增/修改/刪除標籤	標籤名稱	成功新增/修改/刪除
商品管理	商品名稱/編號/賣家帳號/上架時間	顯示所有符合條件之商品
訂單管理	訂單編號/買家帳號/賣家帳號/建立日期/結案日期	顯示所有符合條件之訂單
登入	輸入管理員帳號、密碼	成功登入後台
查看用戶資料	使用者名稱/姓名 (篩選條件:尚未身份驗證/尚未 email 認證/帳號狀態條件)	顯上使用者的 Username/姓名/帳號狀態/身份驗證狀況
更改帳號/email 驗證/身份驗證的狀態	使用者名稱/姓名 (篩選條件:尚未身份驗證/尚未 email 認證/帳號狀態條件)	成功更改用戶帳號/email 驗證/身份驗證的狀態更改
顯示用戶所有詳細資料	無	顯示所有資料庫有寫的用戶資訊

### 5. EER MODEL





## 6. Relation Schema



## 7. 資料表

商品訂單 TUPLE:7

欄位名稱	中文名稱	資料型態	長度	NULL	備註
p_username	賣家使用者名稱	Varchar	128	No	FK
s_username	買家使用者名稱	Varchar	128	No	FK
Order_no	訂單編號	int		No	PK
order_status	訂單交易狀況	Varchar	20	No	

build_date	建立日期	date		No	
complete_date	完成日期	Date		Yes	
Cancel_reason	取消原因 敘述	Varchar	256	Yes	

品項 TUPLE:4

欄位名稱	中文名稱	資料型態	長度	NULL	備註
order_no	訂單編號	int		No	FK, PK
product_no	商品編號	int		No	FK, PK
Price	單價	Int		No	
amount	數量	Int		No	

Admin TUPLE:3

欄位名稱	中文名稱	資料型態	長度	NULL	備註
Admin_name	Adminname	Varchar	128	No	PK
Password	Password	Varchar	128	No	
Email	Email	Varchar	128		

書本商品 TUPLE:15

欄位名稱	中文名稱	資料型態	長度	NULL	備註
Product_no	商品編號	Int		No	PK
Book_name	書名	Varchar	50	No	
ISBN	標準書號	Varchar	20	Yes	
Available	狀態	Boolean	1	No	
Publisher	出版社	Varchar	20	No	
Price	單價	Int		No	
B_language	語言	Varchar	20	No	
Publish_date	出版日期	Date		No	
Exterior	外觀狀態	Varchar	256	Yes	
Stock	庫存量	Int		No	
Photo	書籍照片	Text		No	
Author	作者	Varchar	20	No	
Set_time	建立時間	Timestamp		No	
Username	使用者名稱	Varchar	128	No	fk
Introduct	書本簡介	Varchar	256	Yes	

圖片網址 TUPLE:2

欄位名稱	中文名稱	資料型態	長度	NULL	備註
Product_no	商品編號	Int		No	FK, PK
Image	圖片	Varchar	100	No	PK

商品標籤分類 TUPLE:2

欄位名稱	中文名稱	資料型態	長度	NULL	備註
Product_no	商品編號	Int		No	FK
Class_name	分類名	Varchar	25	No	FK, PK

分類名 TUPLE:1

欄位名稱	中文名稱	資料型態	長度	NULL	備註
Class_name	分類名	Varchar	25	No	FK

偏好面交地點 TUPLE:2

欄位名稱	中文名稱	資料型態	長度	NULL	備註
username	使用者名稱	varchar	128	NO	PK
prefer_loc	偏好面交地點	varchar	128	YES	FK

General User TUPLE:20

欄位名稱	中文名稱	資料型態	長度	NULL	備註
username	使用者名稱	varchar	128	NO	PK
password	密碼	varchar	128	NO	
bdate	生日	date		NO	
name	姓名	varchar	4	NO	
email	信箱	varchar	128	NO	
university	學校	varchar	128	NO	
major	系所	varchar	128	NO	
phonenumner	手機	varchar	10	NO	
gender	性別	char	1	NO	
stdId	學號	char	8	NO	
std_authen	學生證驗證	tinyint	1	YES	
prefer_loc	偏好面交地點	varchar	128	YES	
s_flag	賣家旗標	tinyint	1	YES	
address	地址	varchar	128	YES	
uni_code	統一編號	varchar	128	YES	

信箱驗證 TUPLE:4

欄位名稱	中文名稱	資料型態	長度	NULL	備註
usermail	使用者信箱	varchar	128	NO	FK
set_time	設置時間	timestamp		NO	
link_no	link 號碼	varchar	128	NO	PK
used	效力	boolean	1	NO	

## 八、SQL 建立 TABLE 指令

```

create table book_product (
  product_no int primary key,
  book_name varchar(50) not null,
  ISBN varchar(20) not null,
  publisher varchar (20) not null,
  avialiable boolean not null,
  price int not null,
  b_language varchar(20),
  publish_date date not null,
  exterior varchar (256) ,
  author varchar(20) not null,
  stock int not null,
  username varchar(128) not null,
  introduct varchar(256),
  set_time timestamp username varchar(128) not null);
create table photo(
  product_no varchar(20) not null,
  image text not null,
  primary key(product_no,photo)
);

```

```

create table p_order(
  p_username varchar(128) not null,
  s_username varchar(128) not null,
  order_no int primary key,
  order_status varchar(20) not null,
  build_date date not null,
  complete_date date ,

```

```
cancel_reason varchar(256)
);
```

```
create table items (
  product_no int ,
  order_no int ,
  price int not null,
  amount int not null,
  primary key(product_no,order_no)
);
```

```
CREATE TABLE Classify_tag (
  product_no int not null,
  class_name varchar(25) not null,
  PRIMARY KEY (class_name)
);
```

```
CREATE TABLE tag (
  Class_name varchar(25) primary key
);
```

```
CREATE TABLE Admin (
  Admin_name varchar(128) not null,
  password varchar(128),
  Email varchar(128) ,
  PRIMARY KEY (Admin_name)
);
```

```
CREATE TABLE GeneralUser (
  username varchar(128) not null,
  password varchar(128) not null,
  bdate date not null,
  name varchar(4) not null,
  phonenumber int(10) not null,
  gender char(1) not null,
  stdId char(8) not null,
  university varchar(128) not null,
  email varchar(128) not null,
```

```
major          varchar(128) not null,
std_authen     boolean,
prefer_loc     varchar(128),
s_flag        boolean,
address        varchar(128),
uni_code       varchar(128),
register_name  varchar(128),
man_in_charge  varchar(4),
Admin_name     varchar(4),
certi_state    boolean,
stdid_img      text not null,
PRIMARY KEY (username)
);
CREATE TABLE Prefer_loc (
  prefer_loc   varchar(128),
  s_username   varchar(128),
  PRIMARY KEY (prefer_loc)
);
CREATE TABLE UserEP (
  useremail varchar(128),
  set_time  timestamp,
  link_no   varchar(128),
  used Boolean not null,
  PRIMARY KEY(link_no)
);
ALTER TABLE book_product
ADD FOREIGN KEY (username) REFERENCES
GeneralUser(username);
ALTER TABLE photo
ADD FOREIGN KEY (product_no) REFERENCES
book_product(product_no);
ALTER TABLE Classify_tag
ADD FOREIGN KEY (product_no) REFERENCES book_product
```

```
(product_no);  
ADD FOREIGN KEY (class_name) REFERENCES tag (Class_name);
```

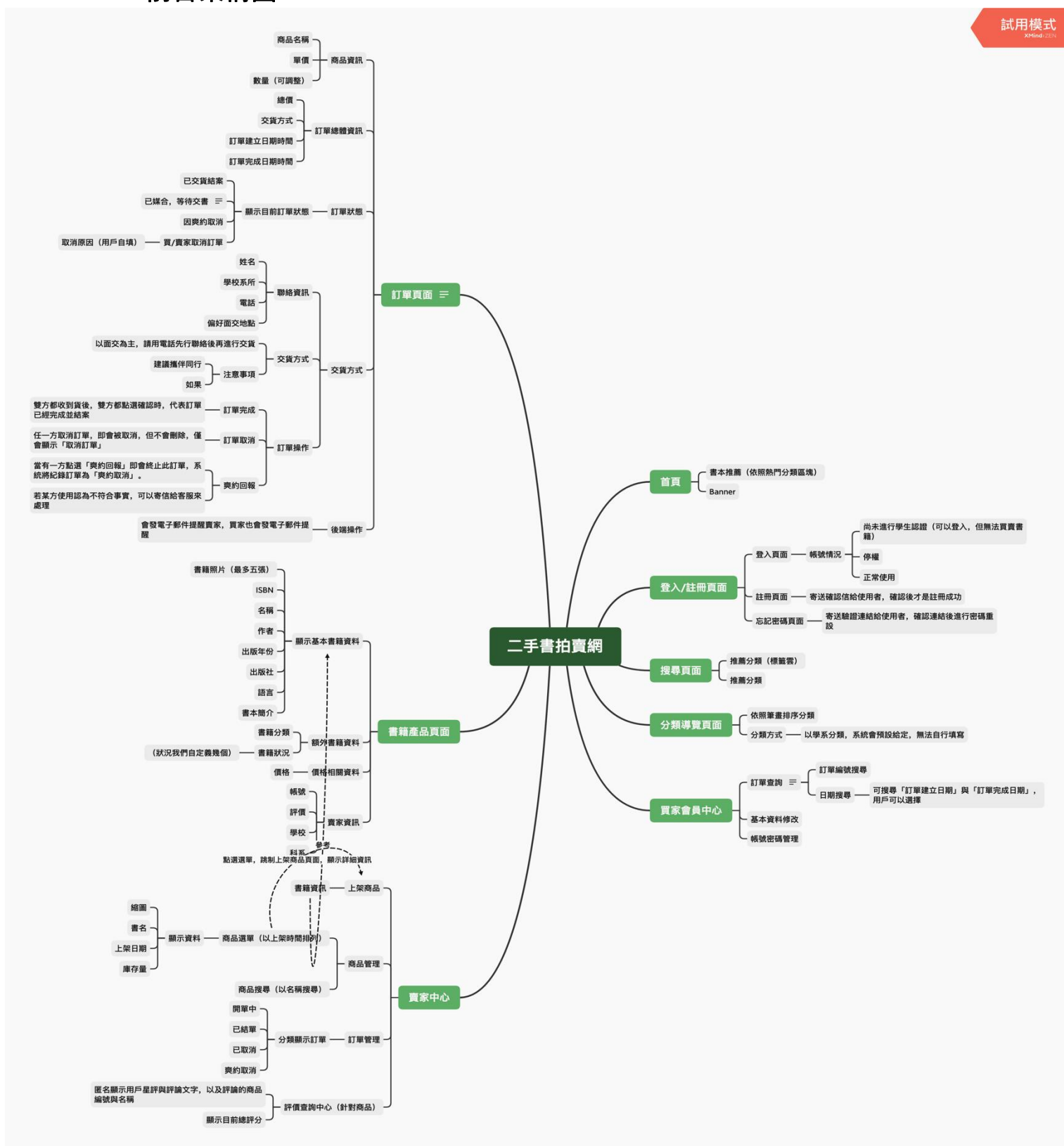
```
ALTER TABLE p_order  
ADD FOREIGN KEY (p_username) REFERENCES GeneralUser  
(username),  
ADD FOREIGN KEY (s_username) REFERENCES GeneralUser  
(username);
```

```
ALTER TABLE items  
ADD FOREIGN KEY (order_no) REFERENCES p_order (order_no),  
ADD FOREIGN KEY (product_no) REFERENCES book_product  
(product_no);
```



# 九、系統功能

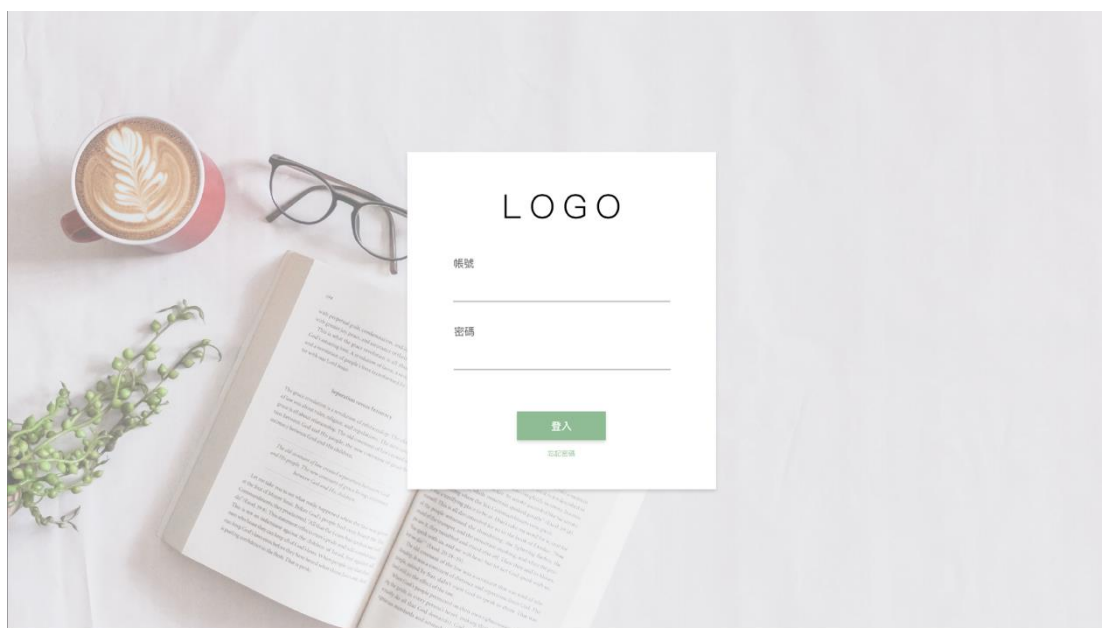
## 前台架構圖





## 一、登入頁面

用 POST 的方法傳送給資料庫，並比對資料庫內有無此帳號和對應密碼



```
2 require __DIR__ . '/../vendor/autoload.php';
3
4 // 建立 Session
5 session_start();
6
7 // 獲取資料庫連線 DAO
8 $db = Database::get();
9
10 // 取得POST登入資料
11 $username = $_POST['username'];
12 $password = $_POST['password'];
13
14 // 查詢資料庫
15
16 // TODO: 針對使用者POST的資料做安全性過濾，再傳入 SQL 裡 query
17
18 //有安全疑慮的寫法 $result = $db->execute("SELECT * FROM user WHERE username = " . "'" . $username . "'" . " ");
19 $result = $db->execute("SELECT * FROM GeneralUser WHERE username = ? AND password = ?;", array($username, $password));
20 if ($db->getRowCount()) {
21     if (!strcmp(sha1($password), sha1($result[0]['password']))) {
22         echo "歡迎登入, ". $username;
23         // username 註冊到 session 變數 ?>
24         <script type="text/javascript">window.location.href="homepage.php"</script>; //重新導向
25         <?php
26         $_SESSION['username'] = $result[0]['username'];
27     }
28 } else {
29     echo '<script type="text/javascript">alert("登入失敗!請確認帳號或密碼");</script>'; ?>
30 <script type="text/javascript">window.location.href="login.php"</script>; //重新導向
31 <?php
```

利用 SQL 語法和資料庫取出相對應的 username，如果有比對密碼，如果密碼不對的話，跳出警告，而如果帳號不在資料庫內，則跳出查無帳號的警告

如果成功登入，存取帳號 SESSION，並跳回首頁。

## 二、忘記密碼



先是輸入電子信箱，放在 URL 上面利用 POST 用法傳送到確認信箱的.php 檔

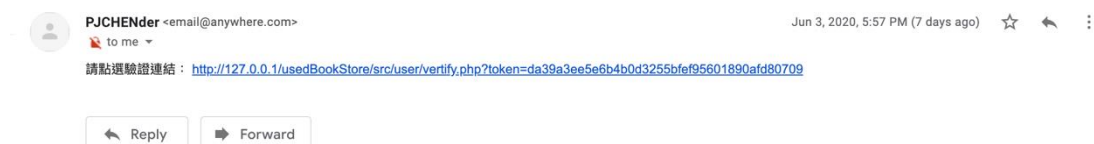
```
$usermail = stripslashes(trim($_POST['email']));

$result = $db->execute("SELECT * FROM USERNAME WHERE email = ?", array($usermail));
if ($db->getRowCount() {
    $token = $num->getrand_id();
    $token = sha1($token);
    $db->execute("INSERT INTO UserEP(usermail, link_no, used) VALUE(?, ?, ?);", array($usermail, $token,0));
    echo $token;
    $link = "http://127.0.0.1/usedBookStore/src/user/verify.php?". "token=$token";
    mb_internal_encoding("utf-8");
    $to= $usermail;
    $subject=mb_encode_mimeheader("你不得不承認這是一個忘記密碼的信", "utf-8");
    $message= "請點選驗證連結: ".$link ;
    $headers="MIME-Version: 1.0\r\n";
    $headers="Content-type: text/html; charset=utf-8\r\n";
    $headers.="From:".mb_encode_mimeheader("PJCHENder", "utf-8")."<email@anywhere.com>\r\n";

    if (mail($to, $subject, $message, $headers)) {
        echo '<script type="text/javascript">alert("驗證連結已寄送到Email!");</script>'; ?>
        <script type="text/javascript">window.location.href="login.php"</script>; //重新導向
        <?php
        //echo "<br />hekkko" . $db->getRowCount();
    } else {
        echo "失敗";
    }
} else {
    echo '<script type="text/javascript">alert("查無此使用者Email!請重新嘗試");</script>'; ?>
    <script type="text/javascript">window.location.href="forget.php"</script>; //重新導向
    <?php
}
?>
```

將 POST 過來得到的 Email 和資料庫比對，如果有在我們的使用者資料中。就生成一個 token 而 token 是用長度為十的亂數其中包括英文字母與數字。再將亂數用 sha1 雜湊成一個 token。生成之後將獨有的 token 與信箱，存入專門管理信箱驗證的 Table:UserEP，接著將生成的 token 串接在驗證頁面之 URL 後頭。

如果查無 Email，則通知失敗，引導回申請忘記密碼頁面。



接著透過 mail 函式傳送連結到資料庫內對應 token 的信箱。

```
1 <?php
2     require __DIR__ . '/../vendor/autoload.php';
3     session_start();
4     // 獲取資料庫連線 DAO
5     $db = Database::get();
6     $token = $_GET["token"];
7     $result = $db->execute("SELECT * FROM UserEP WHERE link_no = ?;", array($token));
8     if ($db->getRowCount()) {
9         if (!$result[0]["used"]) {
10            $_SESSION['token'] = $token;
11            echo "驗證成功<br>";
12            echo "請重設密碼";
13        } else {
14            echo "驗證失敗";
15            echo '<script type="text/javascript">alert("帳號已經重設過!");</script>'; ?>
16        <script type="text/javascript">
17            window.location.href = "login.php"
18        </script>; //重新導向
19    <?php
20    }
21    } else {
22        echo "驗證失敗";
23        echo '<script type="text/javascript">alert("無法取的修改權限!");</script>'; ?>
24    <script type="text/javascript">
25        window.location.href = "login.php"
```

Token 認證頁面從 URL 上取下 token，並從存放 token 的 table UserEP 尋找相同的 token，並且透過 token 的 boolean 值檢查是否使用過，如有使用過跳出驗證失敗訊息，導回首頁。

如果 token 未被使用過，並存在於資料庫內，則驗證成功，進入密碼重新設定之頁面。



接著在重新設定密碼頁面輸入密碼與重複輸入相同密碼，經過前端的 JavaScript 驗證之後，POST 到驗證的.php 檔中執行。

```
$db = Database::get();
$password = $_POST['pass'];
$check = $_POST['checkpass'];

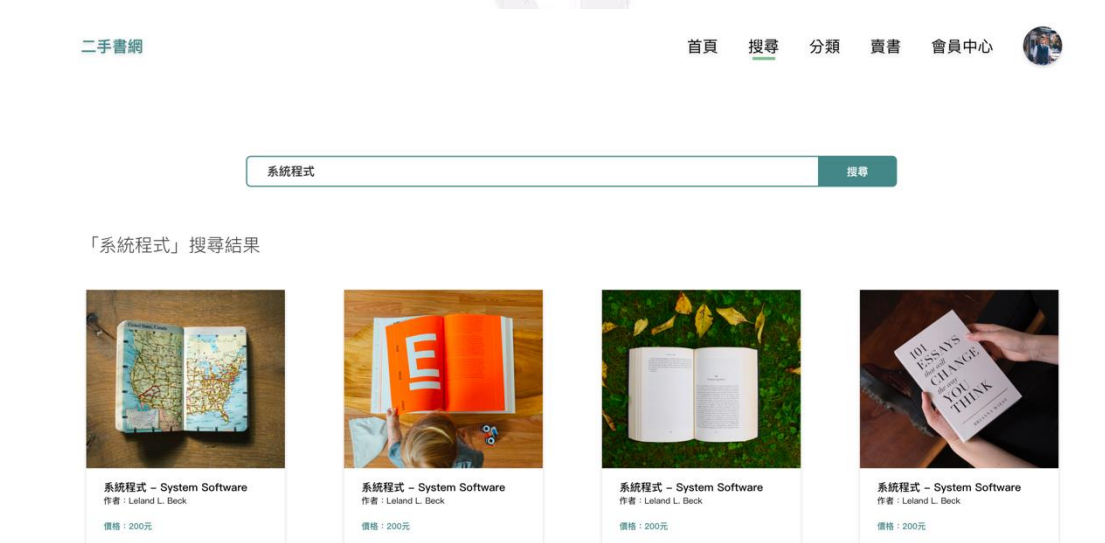
echo $_SESSION["token"]; //TODO token 驗證
if (!strcmp($password, $check)) { // 比較密碼與確認密碼有無被取用
    $result = $db->execute("UPDATE USERNAME SET password = ?
    WHERE email = (SELECT usermail FROM UserEP WHERE link_no = ?)";, array($password, $_SESSION["token"])); //修改密碼
```

如果密碼驗證通過，到存放使用者帳號密碼的 table，修改 (UPDATE) 對應之信箱的使用者密碼。

```
WHERE email = (SELECT usermail FROM UserEP WHERE link_no = ?)";, array($password, $_SESSION["token"])); //
if ($db->getRowCount()) { //如果
    $result = $db->execute("UPDATE UserEP SET used = ? WHERE link_no = ?";, array(1, $_SESSION["token"]));
    if ($db->getRowCount()) {
        echo '<script type="text/javascript">alert("更新成功!請至首頁登入");</script>'; ?>
        <script type="text/javascript">window.location.href="login.php"</script>; //重新導向至首頁
    }
}</php
```

如果順利更新成功，則將 token table 的 token 是否使用的資料由 0 新成 1，標示為使用過，往後如果在使用這組 token 將不能通過驗證

### 三、首頁



```
<?php
require __DIR__ . '/../vendor/autoload.php';
$db = Database::get();
$result = $db->execute("SELECT * FROM book_product ORDER BY RAND() LIMIT 4";, array()); // 隨機取四本
?>
```

隨機推薦，隨機取出資料庫所有的書籍商品，一次取出四筆。這裡有使用標籤的功能，將書的類別分類成標籤，取出特定分類的書籍顯示在頁面上

### 商學熱門

看更多



**企業倫理-商業的道德規範**  
作者：余坤東  
價格: 69



**專題製作最佳秘笈 (商業與管理群適用)**  
作者：林淑芬  
價格: 139



**商業自動化概論2/e**  
作者：傅新彬  
價格: 129



**商業設計藝術**  
作者：新球強  
價格: 19

```
<?php
$result = $db->execute("SELECT * FROM book_product AS b,Classify_tag AS c
WHERE b.product_no = c.product_no AND class_name='國貿系' ORDER BY RAND() LIMIT 4", array());
?>
```

### 資工熱門

看更多



**資料庫系統設計理論**  
作者：李紹繪  
價格: 289



**電信級IP RAN實現**  
作者：陳運清  
價格: 209



**電腦犯罪**  
作者：歐榮禮  
價格: 259



**網際網路應用實務 (第七版)**  
作者：全華研究室  
價格: 149

```
<?php
$result = $db->execute("SELECT * FROM book_product AS b,Classify_tag AS c
WHERE b.product_no = c.product_no AND class_name='資工系' ORDER BY RAND() LIMIT 4", array());
?>
```

上面的 class\_name 國貿系、資工系即是標籤名稱，一次取出四筆該標籤分類資料。

## 四、搜尋頁面

熱門關鍵字

# 二手書，可以救地球，還有學分

輸入關鍵字搜尋，將 keyword 放到 URL 上使用 GET 方法，將關鍵字取下

```
127.0.0.1/usedBookStore/src/user/search.php?keyword=資料
```

接著丟入資料庫搜尋名稱相關的書籍，全數取出並加以分页。  
 計算書籍總數然後以一頁要顯示幾筆，來計算總頁面。進而知道該顯示幾個頁面的頁碼。

```

1 <?php
2 require __DIR__ . '/../vendor/autoload.php';
3 $db = Database::get();
4
5 $keyword=empty($_GET['keyword'])?$_GET['keyword'];
6 if (!empty($keyword)) {
7     $where='where book_name like "%'.$keyword.'"';
8     $link="&keyword=".$keyword; //keyword為空 送回去
9 } else {
10    $where = 'where book_name like "%'. 'nope' . '%"';
11 }
12 echo $keyword;
13 $page_size = 5;
14 $count = $db->execute("SELECT count(*) as C FROM book_product ".$where); // 紀錄搜尋總數
15 $amount = $count[0]['C'];
16 if ($amount == 0) {
17     echo "請輸入關鍵字搜尋";
18 } else {
19     $page_cnt = ceil($amount/$page_size); // 紀錄頁面有幾個
20     $page_num = empty($_GET['page'])?1:$_GET['page'];
21
22     if ($page_cnt==0) {
23         $page_cnt = 1;
24     } elseif ($page_cnt>$page_size) {
25         $page_cnt = $page_size;
26     }
27     // TODO 拼接搜尋SQL 語句
28
29     $limit="limit ".$($page_num-1)*$page_size."
30 //echo $limit;
31 echo "<br>";
32 $sql = "SELECT * FROM book_product ".$where
33 //echo $sql;
34 $result = $db->execute($sql);
  
```

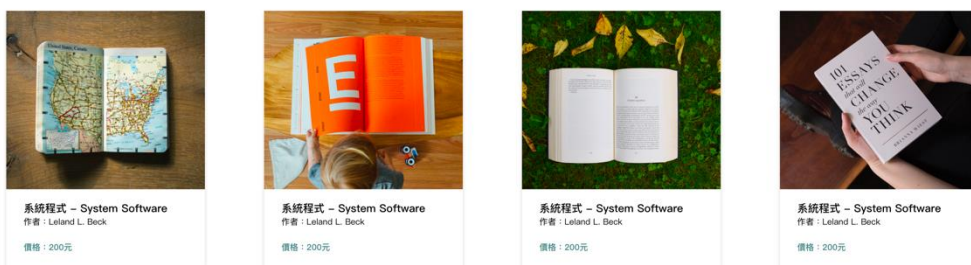
```

35 echo "<a href='page1'".$link.">首頁</a>";
36 if ($page_num>1) {
37     echo "<a href='page'.".($page_num-1).$link.">上一頁</a>";
38 }
39 if ($page_cnt>$page_num) {
40     echo "<a href='page'.".($page_num+1).$link.">下一頁</a>";
41 }
42 if ($page_num==1) {
43     echo "<a href='page'.".($page_cnt-1).$link.">最後一頁</a>";
44 }
45 //echo $result[0]['book_name']; 測試資料輸出
46 echo "<br/>";
47 echo "當前頁", $page_num, "頁 共", $page_cnt, "頁本 有", (($page_num!=$page_cnt)|($amount%$page_size==0)?$page_size:($amount%$page_size)), "筆";
48
49 if ($page_num>6) {
50     // 如果目前的當前頁碼>7的，應該7-5=2開始，最大顯示到7+4=11;
51     for ($i=($page_num-5);$i<=($page_num+4);$i++) {
52         echo "<a class='page' href='page'.". $i ."'>".$i."</a>";
53     }
54 } else {
55     // 當前頁碼小於6，判斷頁總數是否大於10，如果10，最多顯示10個頁碼，否則等於$page_cnt 個頁碼
56     if ($page_cnt>=10) {
57         $size=$page_cnt;
58     } else {
59         $size=10;
60     }
61     for ($i=1;$i<=$size;$i++) {
  
```

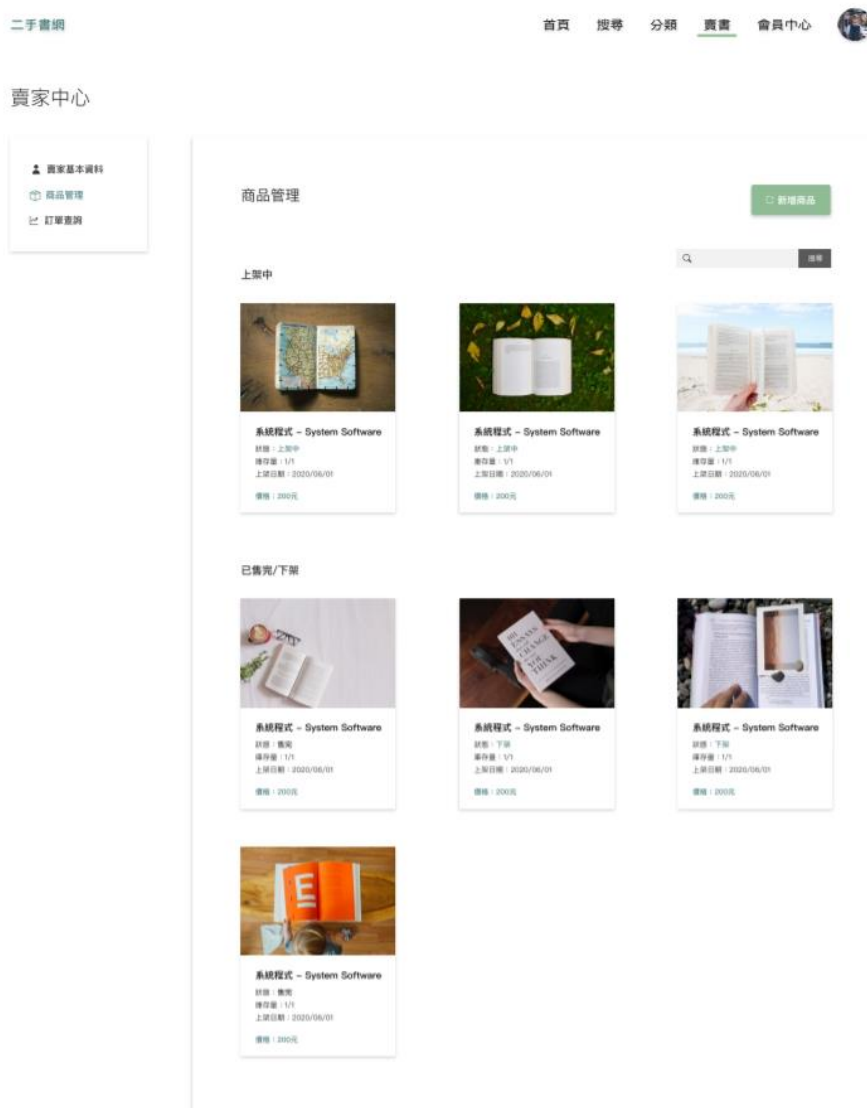


上圖為搜尋成果

「系統程式」搜尋結果



## 五、會員中心 - 商品管理



此頁面紀錄賣家上架中的商品以及已售完或已經下架的商品資料，賣家也可在此頁面點擊各個商品的 Card 來新增/刪除商品資料。

### 新增商品

```
INSERT INTO book_product  
VALUES (.....) ← 商品資料  
SELECT username  
FROM GeneralUser  
WHERE username = username
```

### 刪除商品

```
DELETE FROM book_product  
WHERE product_no = "xxxxxx"
```

## 六、商品詳細頁面

The screenshot shows a product detail page for a book titled "系統程式 - System Software" by Leland L. Beck. The page features a large image of the book cover, which is orange with a white 'E' logo. To the right of the image, the book's details are listed: author (Leland L. Beck), ISBN (978-996-290-064-5), publication year (2017), publisher (書林出版社), language (中文), and category (資訊系). Below this, there are sections for "書籍狀況" (Book Condition) and "銷售狀況" (Sales Status). The price is listed as NTS 300, and there is a green "立即購買" (Buy Now) button. A "書本簡介" (Book Introduction) section provides a brief overview of the book's content, focusing on Windows API programming. At the bottom of the page, there is a dark purple footer with navigation links for "認識我們", "網站地圖", "聯絡我們", and "Logo".

此頁面可以查看特定商品包含的所有資料，如欲購買商品可以點擊立即購買按鈕，即會切換到訂單頁面，如遇商品描述與實體不相符則可檢舉賣家。

檢舉賣家

```
SELECT product_no, username
```

```
FROM book_product
```

```
WHERE product_no="xxxxxx"AND username="xxxxx"
```

立即購買



```
SELECT product_no  
FROM book_product  
WHERE product_no="xxxxxx"
```

## 七、註冊頁面

LOGO  
註冊帳號

E-mail  
輸入email...  
點選進入選擇清單

密碼  
至少8個字，並且包含一個英文及一個數  
點選進入選擇清單

姓名  
輸入姓名...  
點選進入選擇清單

手機號碼  
輸入手機號碼...  
點選進入選擇清單

性別  
點選進入選擇清單

生日  
點選進入選擇清單

學校  
點選進入選擇清單

科系  
點選進入選擇清單

學號  
輸入學號...  
點選上傳檔案(上傳正面)  
點選上傳檔案(上傳反面)

上傳學生證(認證身份, 查看個資隱私政策)

立即註冊

```
register-api.php X  
eric12345566 > src > user > register-api.php > ...  
1 <?php  
2 require __DIR__ . '/../vendor/autoload.php';  
3  
4 // 建立 Session  
5 session_start();  
6  
7 // 獲取資料庫連線 DAO  
8 $db = Database::get();  
9  
10 // 取得POST註冊資料  
11 $username = $_POST['username'];  
12 $email = $_POST['email'];  
13 $password = $_POST['password'];  
14 $name = $_POST['name'];  
15 $phonenumner = $_POST['phonenumner'];  
16 $gender = $_POST['gender'];  
17 $bdate = $_POST['bdate'];  
18 $university = $_POST['university'];  
19 $major = $_POST['major'];  
20 $stdId = $_POST['stdId'];  
21  
22 //判斷是否填寫
```

建立 session 以及資料庫的連線，並利用 POST 回傳資料給資料庫。



## 八、賣家基本資料



```
<?php
require __DIR__ . '/../vendor/autoload.php';
$db = Database::get();
$result = $db->execute("SELECT address,uni_code,man_in_charge FROM generaluser
WHERE username = ? ";, array($username));
```

透過 SELECT 語句挑選出賣家基本資料。

## 九、基本資料修改頁面

二手書網 首頁 搜尋 分類 賣書 會員中心

會員中心

- 個人基本資料
- 帳號密碼修改
- 訂單查詢

姓名：施冠彰 手機號碼：09123456789 性別：男  
生日：1999/10/01 Email：eric12345656@gmail.com

學生證已通過驗證

學校：逢甲大學 偏好寄交地點：人言大樓  
學號：D0745925 是否公費學校： 是  否  
系所：資訊工程學系 申請修改學籍資料：點此申請

儲存修改 取消

認識我們 網站地圖 聯絡我們 Logo  
感謝團隊 首頁 錯誤回報  
關於此專案 搜尋 admin@2handbook.com

```
<?php
require __DIR__ . '/../vendor/autoload.php';
$db = Database::get();
$result = $db->execute("UPDATE generaluser SET name=?,phonenumber=?,gender=?,bdate=?,email=?
WHERE username = $username ;", array($name,$phonenumber,$gender,$bdate,$email));
?>
```

使用 UPDATE，先藉由 WHERE 條件句找出欲修改基本資料的使用者，接者透過 SET 將修改後的基本資料回傳到對應的表格中。

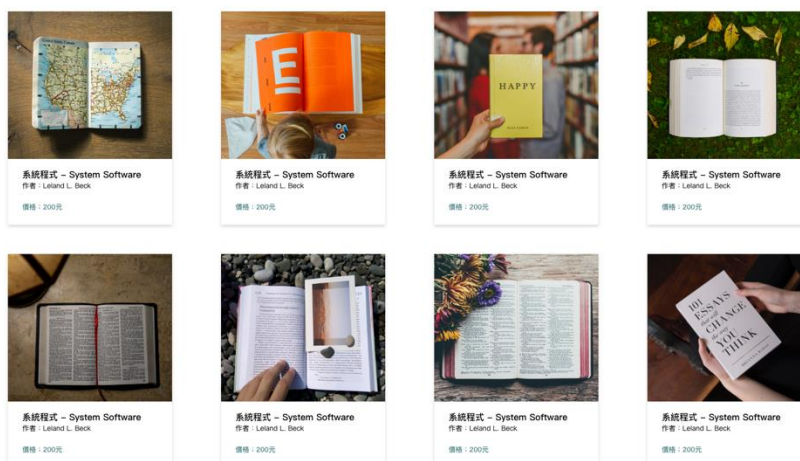
## 十、分類導覽頁面：

透過分類標籤，使用者可以選擇自己要的科系、學院，讓頁面顯示此分類的書籍產品。



### 電機系

看更多



```
1 <?php
2 require 'vendor/autoload.php';
3
4 $tag = $_POST['tag'];
5 $db = Database::get();
6
7 $result = $db->execute("SELECT * FROM book_product WHERE EXISTS (SELECT * FROM Classify_tag WHERE class_name = ?)", array($tag));
8 if ($db->getRowCount() > 0) {
9     for ($i=0; $i < $db->getRowCount(); $i++) {
10         $imgResult = $db->execute("SELECT * FROM photo WHERE product_no = ?", array($result[0]['product_no']));
11         echo "
12         <div class='card'>
13             <div class='card-img'>
14                 <img src='data:image/png;base64, $result[0]['book_name'] |
15             </div>
16             <div class='card-title'>
17                 $result[0]['book_name']
18             </div>
```

先一個從 Classify\_tag 資料庫 SELECT 使用者選擇的分類標籤，再透過另外的 WHERE 去選出有這個標籤的書本詳細資料。並且用此結果 query 出來的 product\_id 再去 query 出相此書本產品的相片 base64 字串。並且將它顯示在我們已經模組設計化的「卡片」上，藉此來顯示我們的書籍資料。

## 十一、賣家中心 - 訂單查詢

可以查詢此賣家的所有賣出交易訂單，並且可在此進入詳細訂單頁面。

二手書網 首頁 搜尋 分類 賣書 會員中心

賣家中心

- 賣家基本資料
- 商品管理
- 訂單查詢

### 訂單資訊

訂單編號	總價	訂單交易狀況	訂單操作
00000001	\$520	已購合、尚未確認	查看詳細
00000001	\$520	已購合、尚未確認	查看詳細
00000001	\$520	已購合、尚未確認	查看詳細
00000001	\$520	已購合、尚未確認	查看詳細
00000001	\$520	已購合、尚未確認	查看詳細
00000001	\$520	已購合、尚未確認	查看詳細
00000001	\$520	已購合、尚未確認	查看詳細
00000001	\$520	已購合、尚未確認	查看詳細
00000001	\$520	已購合、尚未確認	查看詳細
00000001	\$520	已購合、尚未確認	查看詳細
00000001	\$520	已購合、尚未確認	查看詳細

◀ 1 2 3 4 ▶

認識我們  
認識團隊  
關於此專案  
加入我們

網站地圖  
首頁  
搜尋  
分類  
賣書  
會員中心

聯絡我們  
錯誤回報  
admin@2handbook.com  
其他服務  
admin@2handbook.com

Logo  
L O G O

```
13 *
14 * @license MIT - http://datatables.net/license\_mit
15 */
16
17 /* *****
18 * Easy set variables
19 */
20
21 // DB table to use
22 $table = 'p_order';
23
24 // Table's primary key
25 $primaryKey = 'order_no';
26
27 // Array of database columns which should be read and sent back to DataTables.
28 // The 'db' parameter represents the column name in the database, while the 'dt'
29 // parameter represents the DataTables column identifier. In this case simple
30 // indexes
31 $columns = array(
32     array( 'db' => 'order_no', 'dt' => 0 ),
33     array( 'db' => 'price', 'dt' => 1 ),
34     array( 'db' => 'order_status', 'dt' => 1 ),
35 );
36
37 // SQL server connection information
38 $sql_details = array(
39     'user' => MySQL:USERNAME,
40     'pass' => MySQL:PASSWORD,
41     'db' => MySQL:DATABASE,
42     'host' => MySQL:ADDRESS
43 );
44
45
46 /* *****
47 * If you just want to use the basic configuration for DataTables with PHP
48 * server-side, there is no need to edit below this line.
49 */
50
51 require('ssp.class.php');
52
53 echo json_encode(
54     SSP::simple($_GET, $sql_details, $table, $primaryKey, $columns)
55 );
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js">
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js">
<script type="text/javascript" charset="utf8" src="https://code.jquery.com/jquery-3.4.1.min.js">
<script type="text/javascript">
    $(document).ready(function() {
        var table = $('#myTable').DataTable({
            "processing": true,
            "serverSide": true,
            "ajax": "./API/porder-table.php"
        });
    });
```

使用 DataTables 套件，可以快速建立 Table，並且包含搜尋、排序等功能。並且建立 API，可讓前端網頁使用 Ajax 的方式 get 資料與達成搜尋功能。

## 十二、賣家中心 - 商品管理 - 新增書本產品

可在此頁面新增書本產品，並且上傳書本產品照片，以及輸入書籍詳細資料（包含書名、ISBN、作者、書本簡介、書籍狀況……等資料），其中，書籍狀況的設計，是希望使用者在選購二手書時，可以在購買前就知曉所有書籍的外觀狀態，而減少買到自己無法接受的产品。

The screenshot shows the '新增商品' (Add New Product) page in the '賣家中心' (Seller Center). The page includes a navigation bar with '二手書網', '首頁', '搜尋', '分類', '賣書', and '會員中心'. A sidebar on the left contains '賣家基本資料', '商品管理', and '訂單查詢'. The main content area is titled '新增商品' and features a '提交上傳' (Submit Upload) button. Below this, there are three image upload slots labeled 'pic1.jpg', 'pic2.jpg', and 'pic3.jpg'. The form fields include: '書名' (Book Name), '作者' (Author), 'ISBN', '出版年份' (Publication Year), '出版商' (Publisher), '語言' (Language), '書籍分類' (Book Category), '庫存量' (Inventory), '書籍狀況' (Book Condition) with a text input field, '書籍簡介' (Book Introduction) with a text input field, and '單價' (Unit Price). At the bottom, there are '儲存修改' (Save Changes) and '取消' (Cancel) buttons.



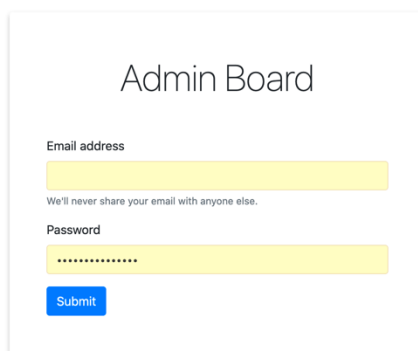


```
1 <?php
2 require 'vendor/autoload.php';
3
4 $book_name = $_POST['book_name'];
5 $ISBN = $_POST['ISBN'];
6 $publisher = $_POST['publisher'];
7 $exterior = $_POST['exterior'];
8 $author = $_POST['author'];
9 $price = $_POST['price'];
10 $db = Database::get();
11
12 $result = $db->execute("INSERT INTO book_product(book_name, ISBN, publisher, exterior, author, price) VALUES (?, ?, ?, ?, ?, ?)"
13 , array($book_name, $ISBN, $publisher, $exterior, $author, $price));
14 if ($db->getRowCount() > 0) {
15     header("Location: ../product-new.php");
16 }
17
```

先透過前端的 form action 將資料使用 POST 送到後端伺服器接收資料的 PHP API。在 PHP 程式中，我們透過擷取 POST 資料，將他放進變數後。在使用 INSERT 指令將資料新增至資料庫中，如果成功則跳回商品列表頁面。

### 十三、 管理員後台 – 登入頁面

此為登入管理員後台的介面，已經針對非法與惡意字元進行過濾。

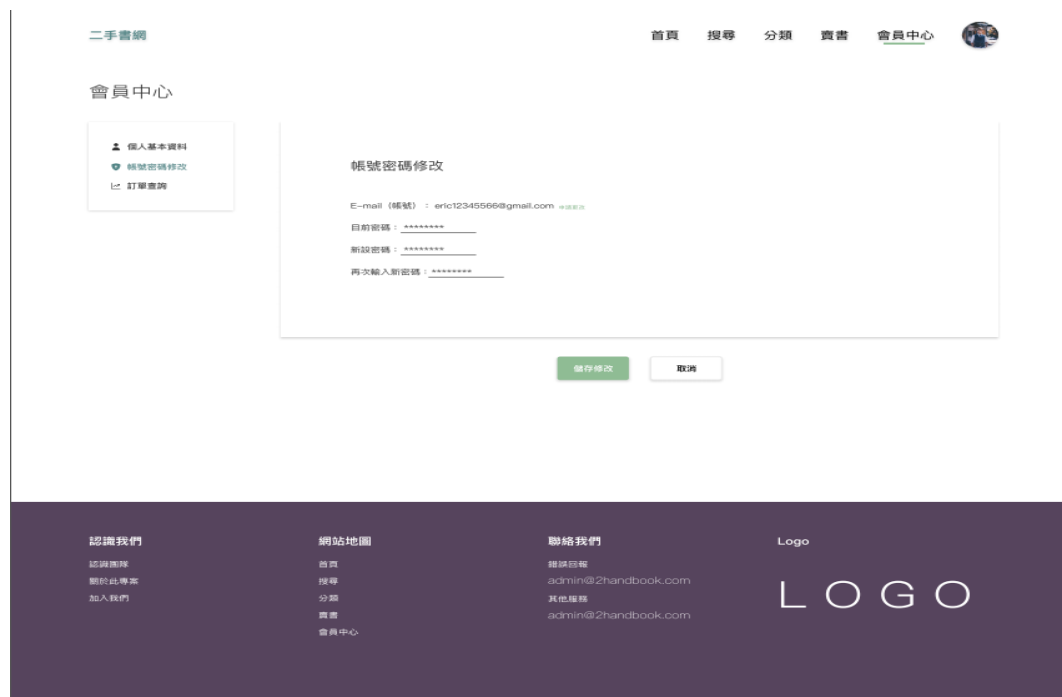


```
$db = Database::get();
$result = $db->execute("SELECT * from Admin Where Admin_name = ?", array($username));
if ($db->getRowCount() && password_verify($password, $result[0]['password'])) {
    $_SESSION['adminName'] = $username;
    header("Location: http://" . Server::serverUrl . Server::prefixUrl . "/backboard/");
} else {
    header("Location: http://" . Server::serverUrl . Server::prefixUrl . "/backboard/");
}
```

使用 SELECT 去獨立的 Admin 資料表 query 帳號資料。值得注意的是，再此會先透過使用者名稱 query 出所有用戶資料（其中包含 hash 過的密碼），並且再透過 password\_verify 的 function 去驗證用戶輸入的密碼與 hash 過得密碼是否一致。如果是，則會在 session 新增 adminName 新增 username 資料，否則會跳

到 login-error.php 頁面，並且顯示錯誤訊息。

### 十三、會員中心



```
<?php
require __DIR__ . '/../vendor/autoload.php';
$db = Database::get();
$result = $db->execute("UPDATE generaluser SET username=?,password=?,
WHERE username = $username ;", array($newusername,$newpassword));
```

使用 UPDATE，先藉由 WHERE 條件句找出欲修改帳密的使用者，接使用 UPDATE，先藉由 WHERE 條件句找出欲修改帳密的使用者，接者透過 SET 將修改後的帳密回傳到對應的表格中。者透過 SET 將修改後的帳密回傳到對應的表格中。

## 十四、交易訂單

### 交易訂單

訂單編號：000000001

書本名稱	單價	數量
系統程式 - System Software	NT\$ 300元	1

總價：NT\$ 300

訂單建立日期：2020/05/24  
 訂單結案日期：尚未結案

#### 聯絡資訊

**買家資訊**

張三  
電話：0933447507  
偏好匯交地點：逢甲大學 買電腦

**賣家資訊**

李四  
電話：0912345678  
偏好匯交地點：逢甲大學 商學院

#### 交書方式

請先與對方用電話約定時間與地點，並在依照約定好的時間地點，與對方實體交書。  
 若已成功完成交書，請回到此頁面點選「訂單完成」，雙方都按下訂單完成後，訂單才會結案。

**注意事項**

- 為了您的安全，建議攜帶同伴前往交書，書很重要的，你需有人幫你 <3
- 請依照約定時間地點赴約，爽約行為若被回報，將會留下紀錄，嚴重者停用帳號。

#### 訂單狀態

🕒 已媒合，等待交書

#### 訂單操作

訂單完成
訂單取消
爽約回報

顯示交易的內容，當訂單完成則會確實減掉賣出的數量。若按取消則會還原原本的商品數量。爽約回報給買家點擊後會有視窗跳出給予填寫

```

1  <?php
2  require 'vendor/autoload.php';
3
4  $db = Database::get();
5  $getId = $_GET['orderid'];
6  $result = $db->execute("SELECT * FROM p_order WHERE order_no = ?", array($getId));
7  if ($db->getRowCount() > 0) {
8      $item = $db->execute("SELECT * FROM items WHERE order_no = ?", array($getId));
9      for ($i=0; $i<$db->getRowCount(); $i++) {
10         $book = $db->execute("SELECT * FROM book_product WHERE product_no = ?", array($item[$i]['product_no']));
11         echo "<td>
12             {$book[0]['book_name']}
13         </td>
14         <td>
15             {$item[$i]['price']}
16         </td>
17         <td>
18             {$item[$i]['amount']}
19         </td>
20     ";
21 }
    
```

```

$puser = $db->execute("SELECT * FROM GeneralUser WHERE username = ?", array($result[0]['p_username']));
$suser = $db->execute("SELECT * FROM GeneralUser WHERE username = ?", array($result[0]['s_username']));
    
```

## 十五、訂單資訊

會員中心

- 個人基本資料
- 帳號密碼修改
- 訂單查詢

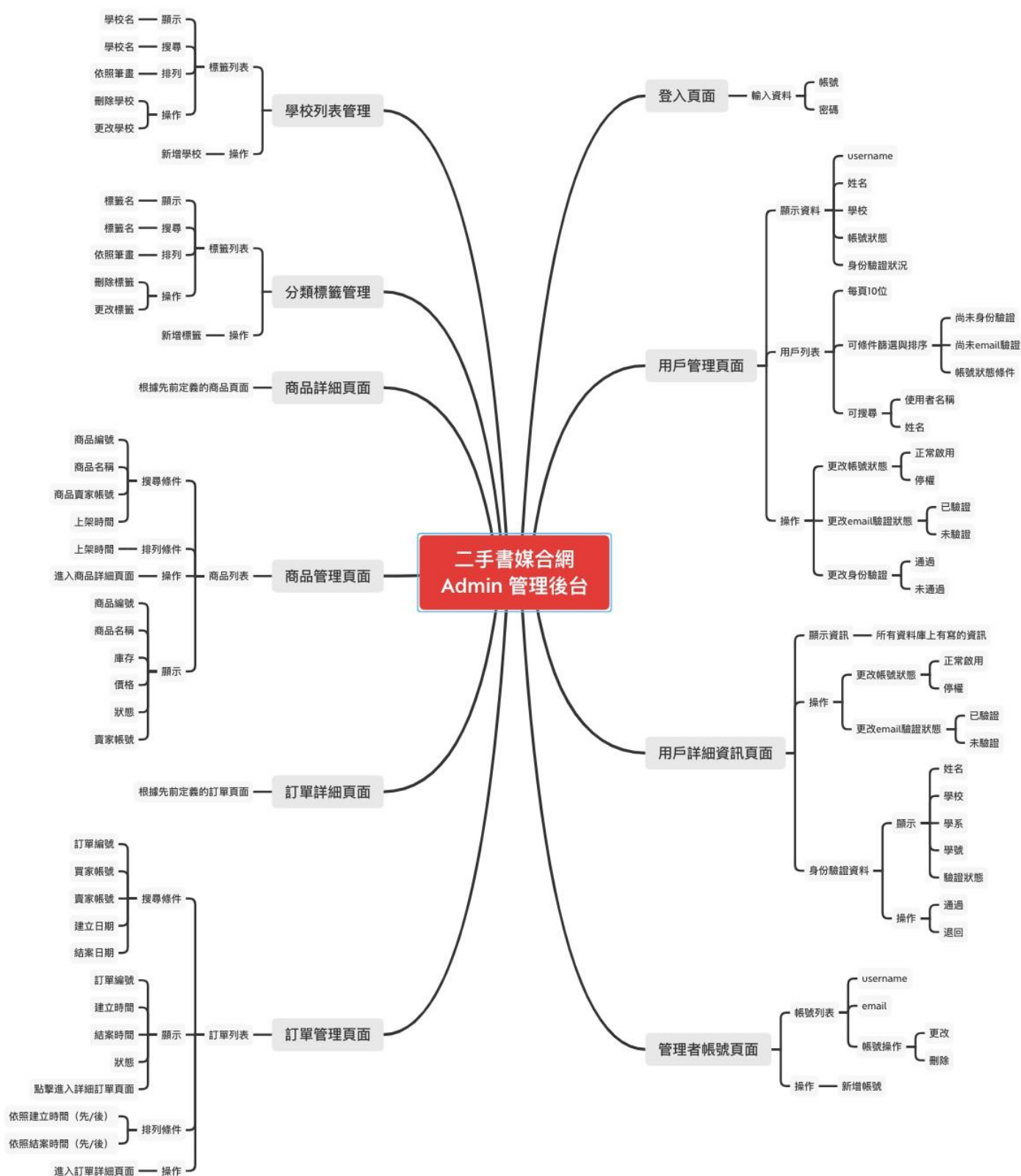
訂單資訊			
訂單編號	總價	訂單交易狀況	訂單操作
00000001	\$520	已媒合, 尚未確認	<a href="#">查看詳細</a>
00000001	\$520	已媒合, 尚未確認	<a href="#">查看詳細</a>
00000001	\$520	已媒合, 尚未確認	<a href="#">查看詳細</a>
00000001	\$520	已媒合, 尚未確認	<a href="#">查看詳細</a>
00000001	\$520	已媒合, 尚未確認	<a href="#">查看詳細</a>
00000001	\$520	已媒合, 尚未確認	<a href="#">查看詳細</a>
00000001	\$520	已媒合, 尚未確認	<a href="#">查看詳細</a>
00000001	\$520	已媒合, 尚未確認	<a href="#">查看詳細</a>
00000001	\$520	已媒合, 尚未確認	<a href="#">查看詳細</a>
00000001	\$520	已媒合, 尚未確認	<a href="#">查看詳細</a>
00000001	\$520	已媒合, 尚未確認	<a href="#">查看詳細</a>
00000001	\$520	已媒合, 尚未確認	<a href="#">查看詳細</a>

比對資料庫內有無此帳, 利用 SQL 語法從資料庫取出相對應的 username, 藉由 WHERE 條件句找出欲符合帳號的使用者 取出對應的 username 資料庫所有的訂單

```
13 *
14 * @license MIT - http://datatables.net/license_mit
15 */
16
17 /*****
18 * Easy set variables
19 */
20
21 // DB table to use
22 $table = 'p_order';
23
24 // Table's primary key
25 $primaryKey = 'order_no';
26
27 // Array of database columns which should be read and sent back to DataTables.
28 // The 'db' parameter represents the column name in the database, while the 'dt'
29 // parameter represents the DataTables column identifier. In this case simple
30 // indexes
31 $columns = array(
32     array( 'db' => 'order_no', 'dt' => 0 ),
33     array( 'db' => 'price', 'dt' => 1 ),
34     array( 'db' => 'order_status', 'dt' => 1 ),
35 );
36
37 // SQL server connection information
38 $sql_details = array(
39     'user' => MySQL::USERNAME,
40     'pass' => MySQL::PASSWORD,
41     'db' => MySQL::DATABASE,
42     'host' => MySQL::ADDRESS
43 );
44
45
46 /*****
47 * If you just want to use the basic configuration for DataTables with PHP
48 * server-side, there is no need to edit below this line.
49 */
50
51 require('ssp.class.php');
52
53 echo json_encode(
54     SSP::simple($_GET, $sql_details, $table, $primaryKey, $columns)
55 );
```

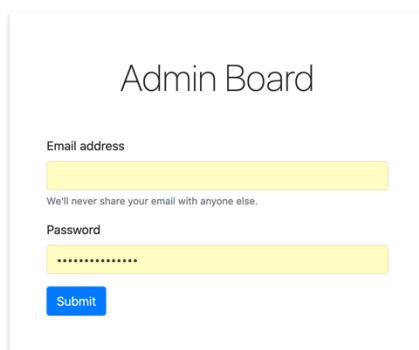
```
src="https://ajax.googleapis.com/ajax/libs/jquery/3.
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/d
src="https://stackpath.bootstrapcdn.com/bootstrap/4.
type="text/javascript" charset="utf8" src="https://c
type="text/javascript">
ment).ready(function() {
table = $('#myTable').DataTable({
rocessing": true,
erverSide": true,|
jax": "./API/porder-table.php"
```

## 後台架構圖



### 十六、管理員後台 - 登入頁面

此為登入管理員後台的介面，已經針對非法與惡意字元進行過濾。



The image shows a login form titled "Admin Board". It contains two input fields: "Email address" and "Password". Below the "Email address" field is a small text note: "We'll never share your email with anyone else." Below the "Password" field is a "Submit" button.

```
$db = Database::get();  
$result = $db->execute("SELECT * from Admin Where Admin_name = ?", array($username));  
if ($db->getRowCount() && password_verify($password, $result[0]['password'])) {  
    $_SESSION['adminName'] = $username;  
    header("Location: http://" . Server::serverUrl . Server::prefixUrl . "/backboard/  
} else {  
    header("Location: http://" . Server::serverUrl . Server::prefixUrl . "/backboard/  
}
```

使用 SELECT 去獨立的 Admin 資料表 query 帳號資料。值得注意的是，再此會先透過使用者名稱 query 出所有用戶資料（其中包含 hash 過的密碼），並且再透過 password\_verify 的 function 去驗證用戶輸入的密碼與 hash 過得密碼是否一致。如果是，則會在 session 新增 adminName 新增 username 資料，否則會跳



The image shows the "Admin Board" login form with an error message. The error message is displayed in a yellow box: "登入失敗，請重新登入！". Below the error message is a "回到登入頁面" button.

到 login-error.php 頁面，並且顯示錯誤訊息。

## 十七、管理後台 - 管理者帳號頁面

針對後台帳號管理員進行列表與新增、更改等操作。並且列表的部分，支援搜尋、排序、分頁功能。

Admin Board

你好, Eric 登出

儀表板  
用戶管理  
訂單管理  
商品管理  
分類標籤管理  
學校列表管理  
管理者帳號

### 管理者帳號

+ 新增管理者帳號

Show 10 entries Search:

Admin Name	Email
admin	admin@anyway.com
dsaasd	eric12345566@gmail.com
eric	eric@anyway.com
kom	eric12345566@gmail.com

Showing 1 to 4 of 4 entries Previous 1 Next

Admin Board

### 新增管理者帳號

管理者帳號

Email address

密碼

Close 送出

kom eric12345566@gmail.com

Showing 1 to 4 of 4 entries Previous 1 Next

```

1 <?php
2 require __DIR__ . '/../vendor/autoload.php';
3 /*
4  * DataTables example server-side processing script.
5  *
6  * Please note that this script is intentionally extremely simple to show how
7  * server-side processing can be implemented, and probably shouldn't be used as
8  * the basis for a large complex system. It is suitable for simple use cases as
9  * for learning.
10 |
11 * See http://datatables.net/usage/server-side for full details on the server-
12 * side processing requirements of DataTables.
13 *
14 * @license MIT - http://datatables.net/license\_mit
15 */
16
17 /* *****
18  * Easy set variables
19  */
20
21 // DB table to use
22 $table = 'Admin';
23
24 // Table's primary key
25 $primaryKey = 'Admin_name';
26
27 // Array of database columns which should be read and sent back to DataTables.
28 // The 'db' parameter represents the column name in the database, while the 'dt'
29 // parameter represents the DataTables column identifier. In this case simple
30 // indexes
31 $columns = array(
32     array( 'db' => 'Admin_name', 'dt' => 0 ),
33     array( 'db' => 'Email', 'dt' => 1 ),
34 );
35
36 // SQL server connection information
37 $sql_details = array(
38     'user' => MySQL::USERNAME,
39     'pass' => MySQL::PASSWORD,
40     'db' => MySQL::DATABASE,
41     'host' => MySQL::ADDRESS
42 );
43
44 /* *****
45  * If you just want to use the basic configuration for DataTables with PHP
46  * server-side, there is no need to edit below this line.
47  */
48
49 require('ssp.class.php');
50
51 echo json_encode(
52     SSP::simple($_GET, $sql_details, $table, $primaryKey, $columns)
53 );
54
55

```

```

186 <!-- Optional JavaScript -->
187 <!-- jQuery first, then Popper.js, then Bootstrap JS -->
188 <!-- <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integ
189 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.mi
190 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper
191 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootst
192 <script type="text/javascript" charset="utf8" src="https://cdn.datatables
193 <script type="text/javascript">
194     $(document).ready(function() {
195         var table = $('#myTable').DataTable({
196             "processing": true,
197             "serverSide": true,
198             "ajax": "./API/admin-table.php"
199         });

```

針對 Admin 帳號列表的部分，使用 DataTables 套件來實現，他有自定義規範與資料庫 query 資料的 API，實現了資料顯示、搜尋、排序與分頁等功能。在前端部分，使用 ajax 的方式來取資料，因此可以在不重整全部頁面得狀況下，進行分頁、搜尋與排序等動作的頁面資料更新。以此來省下伺服器與使用者瀏覽器的資源。



```

1 <?php
2 session_start();
3 require __DIR__ . '/../vendor/autoload.php';
4 if (!isset($_SESSION['adminName'])) {
5     header("Location: http://" . Server::serverUrl . Server::prefixUrl . "/backboard/login.php");
6 }
7
8
9 $adminName = $_POST['adminName'];
10 $email = $_POST['email'];
11 $password = $_POST['password'];
12 $hashPwd;
13 $isError = false;
14 $error_msg = "000"; // 1: adminName error, 2: email error, 3: password error
15
16 if (!preg_match('/^(?!.*[\x21-\x7e])(?=.{4,10})(?=[\W])(?=[a-zA-Z])(?=[\d]).*$/i', $password)) {
17     $error_msg[2] = '1';
18     $isError = true;
19 }
20
21 if (!preg_match('/^[A-Za-z0-9_]+[A-Za-z0-9_]*@[1]{1}[A-Za-z0-9_]+[A-Za-z0-9_]*[.]{1}[A-Z-a-z]{2,5}$/i', $email)) {
22     $error_msg[1] = '1';
23     $isError = true;
24 }
25
26 $adb = Database::get();
27 if ($isError) {
28     header("Location: ../admin.php?errorMsg=" . $error_msg);
29 } else {
30     $hashPwd = password_hash($password, PASSWORD_DEFAULT);
31     $result = $adb->execute("INSERT INTO Admin (Admin_name, password, Email) VALUES (?, ?, ?);", array($adminName, $hashPwd, $email));
32     if ($adb->getRowCount() > 0) {
33         header("Location: ../admin.php");
34     }
35 }
36

```

針對註冊頁面的部分，我們接受來自前端 POST 的資料後，會先對 email 與密碼進行過濾與規則的檢查（尤其密碼，為了安全性，我們有一定的包含字元與長度的要求）。並且我們在寫入密碼時，為了預防資料庫不幸被攻破，資料外洩的狀況下，不小心把密碼暴露出來，因此，我們針對密碼進行了雜湊加密，因此即使外洩也可以保證密碼難以外流。並且，我們使用 INSERT 指令，將資料寫入 ADMIN 資料表中。



## 十、組員心得

### 戎宥杰 心得:

繼上學期的軟體工程導論做的團體專題報告之後，這是我上大學後第二次實作較有規劃性的團隊合作專題。有了前一次與其他組員工同合作的經驗，這次我們在分配工作與規劃流程上有了較完整的規劃與構想。在專題實作的過程中，我慢慢地體會、學習到一個軟體工程團隊該如何運作、分工來盡可能的在一定的時限內將大家的結晶組合起來。或許畢了業以後，這種團隊合作模式極有可能就會直接套用在我們將來的工作上。不管是 尋找組員、規劃行事曆、工作分配、選擇最合適的程式語言實作專題、摸索新的程式語言、撰寫企劃書、構想主題、畫出 ER-model 與 EER-model、繪製 UI、畫出 relation schema、複習 HTML、將功能輸入輸出建成 TABLE、建立後端資料庫 SQL、將前後端接在一起、使用 github 維護我們的專案、一再修改所有資料、學習時間管理、頂住期末壓力，只有當每個人各司其職，發揮自己的專長，盡可能將自己負責的部分寫到最好，才有可能讓別人看見我們團隊的價值。

### 丁泓哲 心得:

在上這堂資料庫的課之前，有先做過軟體工程的專題，那時候就有稍微接觸到資料庫的東西，但沒想到資料庫是有那麼多學問的，尤其是在實作專題時，才深深感受到，前面課程鋪成的重要性。要有前導的那些資料庫課程，規劃起資料庫和應用程式才有比較有條理和效率。

這是我第一次撰寫後端程式，本來是想說要使用 node.js 來時做，但後期發現我們的時間其實不太足夠，加上 P H P 入門感覺對初學者比較容易就使用 PHP，本來認為這是一個真的非常困難的東西。但在完成功能的過程中，其實自然就會找到新的解決方法，新的技巧。所以我覺得資料庫專題對我來說是獲益良多的，其中經歷了撰寫 SQL 語法的思考邏輯、PHP 的 bug，能夠一一解決，也是增長了許多的經驗。

最後想感謝我組別的隊友，和他們已經合作一段時間了，也都會彼此扶持，遇到問題一起解決，能夠找到一起做專題那麼有默契的隊友真的難能可貴，雖然到 DEMO 之前都還是每天都忙翻天。但收穫真的是滿滿。

### 施冠彰 心得:

這是一個有趣挑戰！怎麼說呢？事實上，這次的資料庫專題，我們可以說是身處在身兼多份專題、考試、比賽（而且繳交時間都卡在一起）的狀況下，利用自己僅剩的零碎時間，好不容易把我們一直很想做的二手書媒合交易平台給弄出來，當然，以現在的情況，我們也真的快弄出來了，真的可喜可賀。

事實上，有好幾個時間點，我們在極度缺乏時間的狀況，認為很高的機率會寫不出來。坦白說在我寫這篇心得時，還有許多東西我們還在修理與趕工。但驅使我們前進的力量，絕對不只是希望得到高分，而是我們對於這個系統的熱愛、可望看到他可以問世的那種心情，驅使我們可以不斷前進，就算工作到三更半夜，也要想辦法把那個 UI 設計到最好，也要想盡辦法把那些複雜的功能與頁面，用 PHP、用 HTML 把它接起來。我們期望他的成果是可以真正在網路上運行的，並且可以在有朝一日，真的幫我們把家裡的二手書賣掉。而我認為，我們離夢想的實現，已經不遠了。

這次，我負責的部分是 UI/UX 的設計、系統架構的規劃，以及指揮這次的開發進度，並且擔任全端工程師的角色，除了設計整個管理員後台（Admin Board）的前後端程式，也會在其他前後端的組員需要幫忙時，適時給予協助。坦白說，我很感謝我的組員，可以在這次的開發中，即使日常課業事務繁忙的狀況下，也願意全力支持我們團隊，把安排的事務按時完成。同時也要告訴我的組員，各位辛苦了，我們這次，又創造了一個大學生涯中美好的回憶。

#### **晁文勤 心得：**

做過這次的專題之後，才明白設計一個資料庫真的不是一件輕鬆的事情，整個資料庫的構想，從基本的建立 ER、EER model，到 relation schema，以及建立 MySQL 的 table 和欄位，都是經過詳細的討論，當然過程中也會遇到要修改的地方，所以還要回去修改資料庫的內容，而且也要學習沒學過的程式語言 PHP，遇到問題就要一直 google 或問同學，所以真的是很累人。不過也因為這次的專題，我覺得我對資料庫越來越熟悉，如何 query、如何 insert、如何 select，每寫一次就更清楚 SQL 的語法。

#### **林永吉 心得：**

資料庫系統專題，是我認為本課程的精華所在，因為平時上課只是聽完懂得概念而已，例如 relation schema，但是對於應用還是不夠瞭解，但為了建立專題的資料庫，我們從 ER 到 EER 再至 relation schema，過程中我們不斷思考各種繪製方式審視架構圖，以得到更完善的架構。然而在將圖表轉為程式碼時，我們使用老師上課所傳授的 SQL 語法配合 PHP 撰寫專題後端，藉此更熟悉 SQL 語法，不僅如此，為了有一個漂亮的頁面，我重新複習了 html、css、bootstrap、javascript 等撰寫網站的程式語言，單單一週的時間，我就寫了比我上學期修的 web 設計課程所寫的網頁還多，資料庫系統真的是上一門學兩門的好課程。

## 參考文獻

- 克夫勒（2006）。MySQL 徹底研究。新北市：博碩。
- 陳俊宏（2000）。PHP 4 網站實作，深度研究篇。臺北市：旗標。
- 杜克特（2012）。HTML&CSS：網站設計建置優化之道。臺北市：碁峰。
- 丁月光（2009）。PHP + MySQL 動態網站開發。臺北市：佳魁資訊。
- 唐漢明（2014）。MySQL 完全攻略：資料庫開發與效能調校。新北市：博碩

