

# 浙江大学实验报告

课程名称：\_\_\_\_\_数据库系统原理\_\_\_\_\_实验类型：\_\_\_\_\_上机\_\_\_\_\_

实验项目名称：\_\_\_\_\_图书管理系统\_\_\_\_\_

学生姓名：\_\_\_\_\_王子腾\_\_\_\_\_专业：\_\_\_\_\_软件工程\_\_\_\_\_学号：\_\_\_\_\_3180102173\_\_\_\_\_

同组学生姓名：\_\_\_\_\_无\_\_\_\_\_指导老师：\_\_\_\_\_陈岭\_\_\_\_\_

实验地点：\_\_\_\_\_个人电脑\_\_\_\_\_实验日期：\_\_\_\_\_2020\_\_\_\_\_年\_\_\_\_\_4\_\_\_\_\_月\_\_\_\_\_16\_\_\_\_\_日

## 一、实验目的和要求

设计并实现一个精简的图书管理系统，要求具有图书入库、查询、借书、还书、借书证管理等功能。  
通过该图书馆系统的设计与实现，提高学生的系统编程能力，加深对数据库系统原理及应用的理解。

## 二、实验内容和原理

### (1) 基本数据对象

对象名称	包含属性
书	书号, 类别, 书名, 出版社, 年份, 作者, 价格, 总藏书量, 库存
借书证	卡号, 姓名, 单位, 类别 (教师 学生等)
管理员	管理员 ID, 密码, 姓名, 联系方式
借书记录	卡号, 借书证号, 借期, 还期, 经手人 (管理员 ID)

### (2) 基本功能模块

模块名称	功能描述
管理员登陆	输入管理员 ID, 密码; 登入系统 或 返回 ID/密码 错误.
图书入库	1. 单本入库 2. 批量入库 (方便最后测试) 图书信息存放在文件中, 每条图书信息为一行. 一行中的内容如下 ( 书号, 类别, 书名, 出版社, 年份, 作者, 价格, 数量 ) Note: 其中 年份、数量是整数类型; 价格是两位小数类型; 其余为字符串类型 Sample:

	( book_no_1, Computer Science, Computer Architecture, xxx, 2004, xxx, 90.00, 2 )
图书查询	<p>要求可以对书的 类别, 书名, 出版社, 年份(年份区间), 作者, 价格(区间) 进行查询. 每条图书信息包括以下内容:</p> <p>( 书号, 类别, 书名, 出版社, 年份, 作者, 价格, 总藏书量, 库存 )</p> <p>可选要求: 可以按用户指定属性对图书信息进行排序. (默认是书名)</p>
借书	<p>1.输入借书证卡号</p> <p>显示该借书证所有已借书籍 (返回, 格式同查询模块)</p> <p>2.输入书号</p> <p>如果该书还有库存, 则借书成功, 同时库存数减一。</p> <p>否则输出该书无库存, 且输出最近归还的时间。</p>
还书	<p>1.输入借书证卡号</p> <p>显示该借书证所有已借书籍 (返回, 格式同查询模块)</p> <p>2.输入书号</p> <p>如果该书在已借书籍列表内, 则还书成功, 同时库存加一。</p> <p>否则输出出错信息。</p>
借书证管理	增加或删除一个借书证。

除图书查询功能外, 其余功能模块都应该由图书管理员操作。

### (3) 用户界面

可采用图形界面或字符界面。如果采用图形或网页界面, 酌情加分。

## 三、 主要仪器设备

1. 操作系统: windows 10
2. DBMS: MYSQL 8.0.18
3. 运行环境: WampServer 3.1.9

## 四、 操作方法与实验步骤

### (一) 系统总体设计

#### 1. 系统整体架构描述

本系统采用网页形式作为图形化界面, 使用 MVC 模式设计, 服务器端渲染架构。前端 HTML 辅以 PHP 渲染用于在浏览器中展示视图并通过向后端 PHP 发送请求, 控制过程由 JavaScript 以及 PHP 实现, 当后端处理请求遇到错误返回或成功操作后, 向前端构造出返回通道和反馈信号, 并转移回前端界面。

其中, 部分 HTML 和 JS 代码参考了 bootstrap 框架, PHP 与 MySQL 的交互通过 mysqli 的面向对象方式进行。

MVC 架构图：

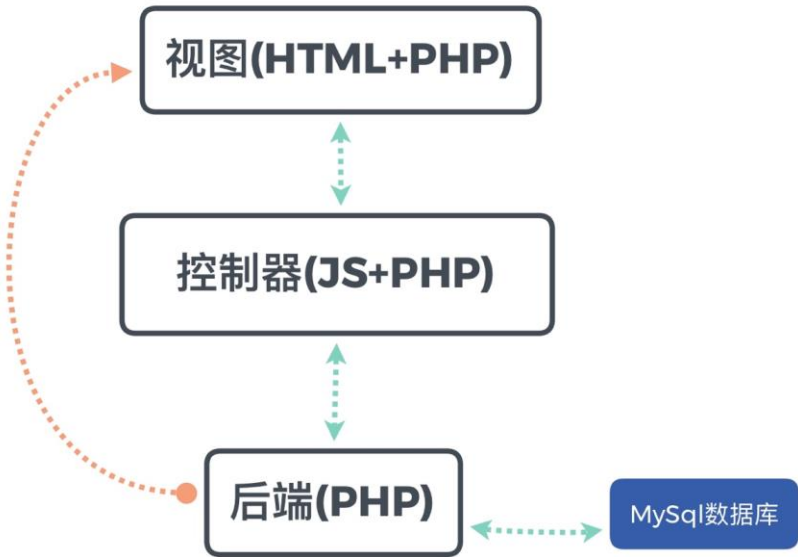


图 1

文件部分架构如下：

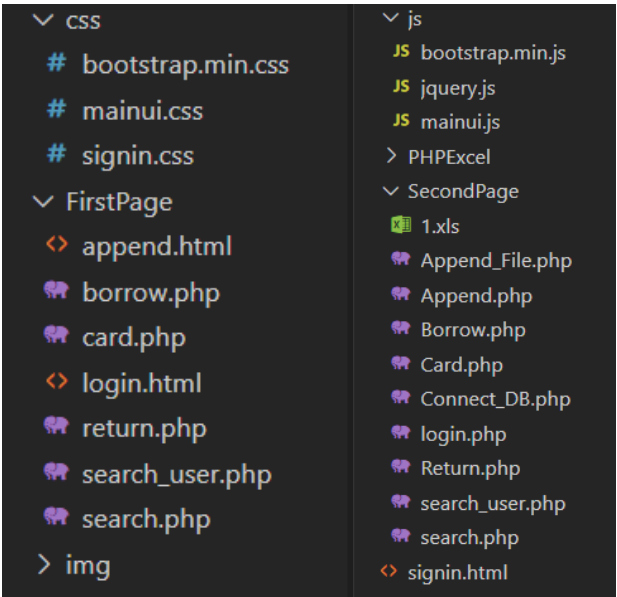


图 2

其中①signin.html 文件为程序的入口文件；②css 目录内为.css 文件，即层叠样式表，用于绘制 html 表现；③FirstPage 目录下存储了前端文件，用以作为视图表现；④img 内存储的是背景图片；⑤js 目录存储了程序控制用的 js 文件；⑥PHPExcel 为引用的库文件，便于后端处理 Excel 文件；⑦SecondPage

目录内为程序的后端文件，用于与 mysql 数据库交互处理前端反馈。

## 2. 数据库表结构设计

建表 book, card, admin, borrow 并定义如下：

```
1.  create table book
2.  (
3.      bno char(8),
4.      category char(10),
5.      title varchar(40),
6.      press varchar(30),
7.      year int,
8.      author varchar(20),
9.      price decimal(7,2),
10.     total int,
11.     stock int,
12.     primary key(bno)
13. );
14.
15. create table card
16. (
17.     cno char(7),
18.     name varchar(10),
19.     department varchar(40),
20.     type char(1),
21.     primary key(cno),
22.     check (type in ('T','S'))
23. );
24.
25. create table admin
26. (
27.     admin_id char(20),
28.     password char(20),
29.     name char(20),
30.     email char(30),
31.     primary key(admin_id)
32. );
33.
34. create table borrow
35. (
36.     cno char(7),
37.     bno char(8),
38.     borrow_date date,
```

```

39.     return_date date,
40.     done_by char(20),
41.     foreign key (bno) references book(bno) on delete cascade,
42.     foreign key (cno) references card(cno) on update cascade
43. );

```

## (二) 系统详细设计

### 实现流程图

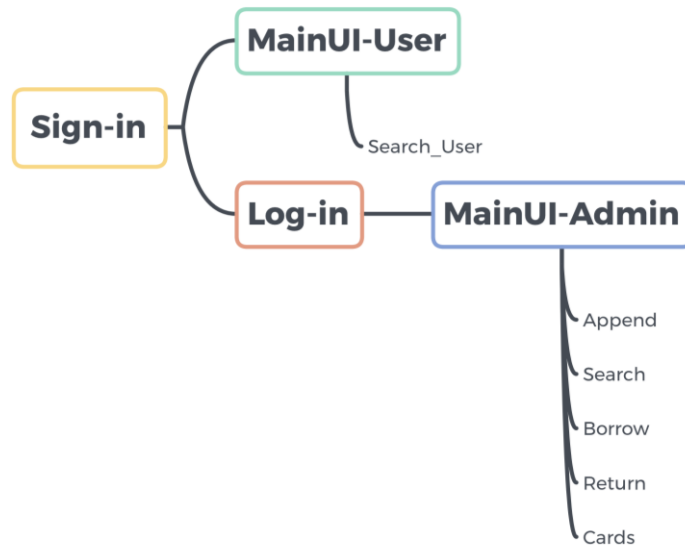


图 3

### 1. PHP 与 MySQL 连接

通过 PHP 的 `mysqli` 功能，用面向对象的方式连接数据库，其中为本门课程建立的数据库端口号为 3306，库名 `test`。

代码实现如下：

```

1. $conn = new mysqli('localhost', 'root', 'wzt75276477', 'test', '3306');
2. if ($conn->connect_error) {
3.     die("连接失败: " . $conn->connect_error);
4. }

```

### 2. 背景与主界面效果渲染

Css 文件夹内的为 `bootstrap.min.css` 为 css 库文件，`mainui.css` 则修改了本程序所需的特效，其中 `body` 部分设定代码如下：

```

1. body {
2.     background-image: url("../img/demo-1-bg.jpg");

```

```

3.  background-position: center 0;
4.  background-repeat: no-repeat;
5.  background-attachment: fixed;
6.  background-size: cover;
7.  -webkit-background-size: cover;
8.  -o-background-size: cover;
9.  -moz-background-size: cover;
10. -ms-background-size: cover;
11.
12. /* margin-bottom: 60px; */
13.
14. padding-top: 56px;
15. }

```

表格部分实现如下：以 search.php 为例：

```

1. <div class="col-md-7 order-md-2 text-center">
2.     <table class="table text-light h5">
3.         <thead>
4.             <tr>
5.                 <th scope="col">bno</th>
6.                 <th scope="col">category</th>
7.                 <th scope="col">title</th>
8.                 <th scope="col">press</th>
9.                 <th scope="col">year</th>
10.                <th scope="col">author</th>
11.                <th scope="col">price</th>
12.                <th scope="col">total</th>
13.                <th scope="col">stock</th>
14.            </tr>
15.        </thead>
16.        <tbody>
17.            <?php
18.                while($row=$result->fetch_object())
19.                {
20.                    echo '
21.                        <tr>
22.                            <td>' . $row->bno . '</td>
23.                            <td>' . $row->category . '</td>
24.                            <td>' . $row->title . '</td>
25.                            <td>' . $row->press . '</td>
26.                            <td>' . $row->year . '</td>
27.                            <td>' . $row->author . '</td>
28.                            <td>' . $row->price . '</td>

```

```

29.         <td>'.$row->total.'</td>
30.         <td>'.$row->stock.'</td>
31.     </tr>';
32.     }
33.     ?>
34. </tbody>
35. </table>
36. </div>

```

### 3. Sign-in 用户界面

只使用前端 HTML 架构，根据用户选择跳转相应界面。

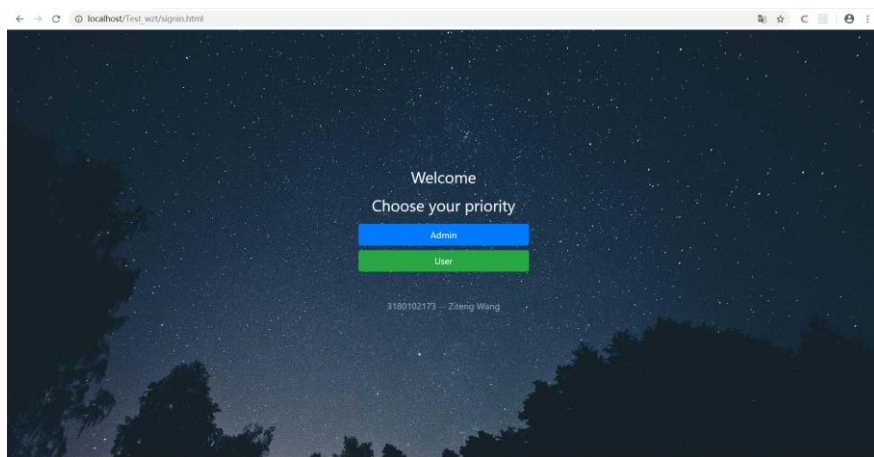


图 4

跳转实现：

```

1. <h1 class="h3 mb-3 text-light font-weight-normal">Welcome</h1>
2. <h1 class="h3 mb-3 text-light font-weight-normal">Choose your priority</h1>
3. <a class="btn btn-primary btn-block" href="FirstPage/login.html">Admin</a>
4. <a class="btn btn-success btn-block" href="FirstPage/search_user.php">User</a>

```

### 4. MainUI-User 用户主界面（仅能搜索）

此处使用 HTML，配合 PHP 与数据库交互渲染表格效果，如图 5 所示。查询功能后端实现与 MainUI-Admin 中实现相同：通过前端将输入通过 Action-Post 方式传递后端 PHP 文件，根据填写值实现差异化 SQL 查询语句，并将结果通过与 HTML 组合形式输出在表格中，如当输入种类为“计算机”，最低价 50 元，作于 2010 年以后，按照库存排序的查询条件时，

结果如图 6。同时用户之前的输入值被封锁在表格内，由于未具有管理员权限，导航栏设置为 disabled 模式，无法跳转到其他功能。

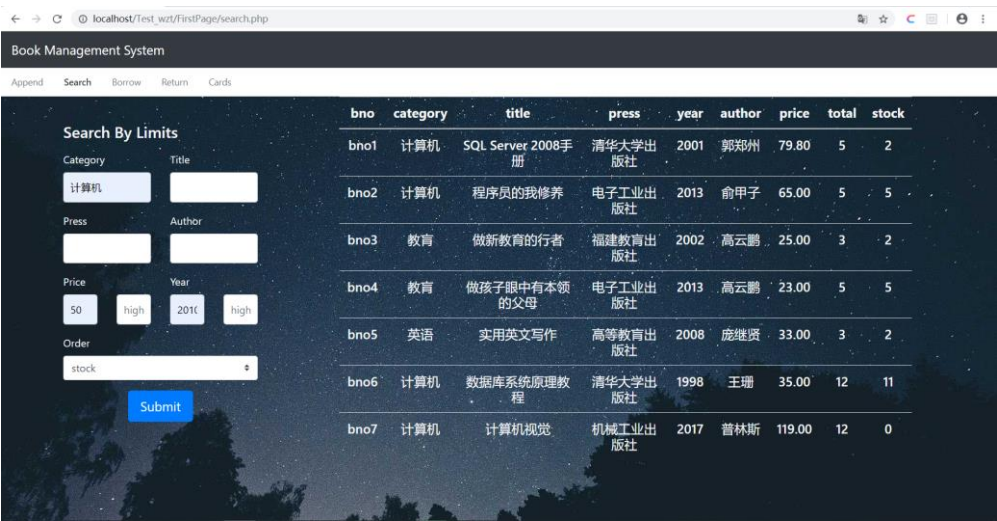


图 5

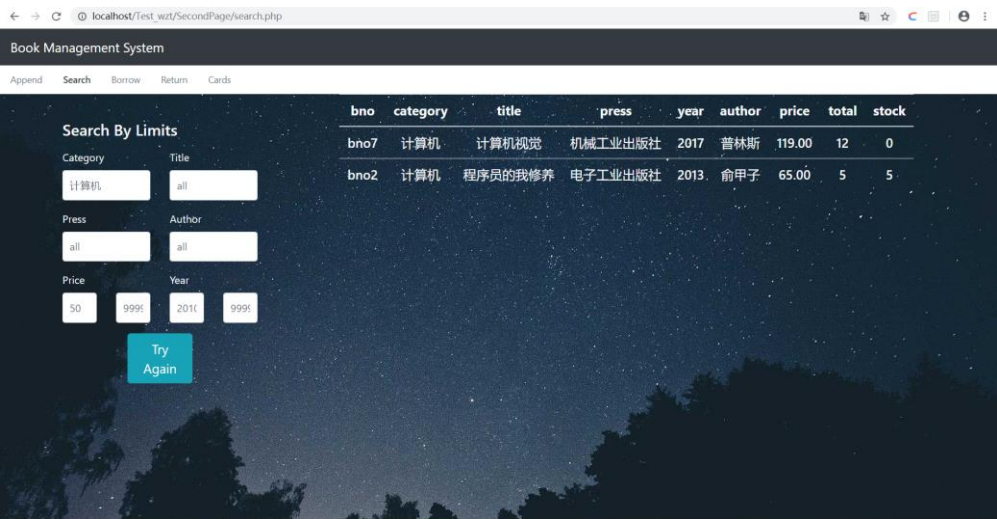


图 6

构造筛选 sql 的语句算法如下：

```
1. $cat = $_POST['cat'];
2. $tit = $_POST['tit'];
3. $pre = $_POST['pre'];
4. $aut = $_POST['aut'];
5. $prl = $_POST['prl'];
6. $prh = $_POST['prh'];
7. $yel = $_POST['yel'];
8. $yeh = $_POST['yeh'];
9. $opt = $_POST['opt'];
10.
```



```
11. $sql = "select * from book where";
12.
13. $isnull = true;
14. if($cat != ""){
15.     $sql = $sql." category='".$cat."' ";
16.     $isnull = false;
17. }
18. if($tit != ""){
19.     if($isnull){
20.         $sql = $sql." title='".$tit."' ";
21.     }
22.     else{
23.         $sql = $sql."and title='".$tit."' ";
24.     }
25.     $isnull = false;
26. }
27. if($pre != ""){
28.     if($isnull){
29.         $sql = $sql." press='".$pre."' ";
30.     }
31.     else{
32.         $sql = $sql."and press='".$pre."' ";
33.     }
34.     $isnull = false;
35. }
36. if($aut != ""){
37.     if($isnull){
38.         $sql = $sql." author='".$aut."' ";
39.     }
40.     else{
41.         $sql = $sql."and author='".$aut."' ";
42.     }
43.     $isnull = false;
44. }
45. if($prl != "" || $prh != ""){
46.     if($prl == "") $prl = "0";
47.     if($prh == "") $prh = "9999";
48.     if($isnull){
49.         $sql = $sql." price between ".$prl." and ".$prh." ";
50.     }
51.     else{
52.         $sql = $sql."and price between ".$prl." and ".$prh." ";
53.     }
54.     $isnull = false;
```

```

55. }
56. if($ye1 != "" || $yeh != ""){
57.     if($ye1 == "") $ye1 = "0";
58.     if($yeh == "") $yeh = "9999";
59.     if($isnull){
60.         $sql = $sql." year between ".$ye1." and ".$yeh." ";
61.     }
62.     else{
63.         $sql = $sql."and year between ".$ye1." and ".$yeh." ";
64.     }
65.     $isnull = false;
66. }
67. if ($isnull){
68. q1 = "select * from book ";
69. }
70. switch($opt)
71. {
72.     case 1:
73.         $sql=$sql."order by bno";break;
74.     case 2:
75.         $sql=$sql."order by category";break;
76.     case 3:
77.         $sql=$sql."order by title";break;
78.     case 4:
79.         $sql=$sql."order by press";break;
80.     case 5:
81.         $sql=$sql."order by author";break;
82.     case 6:
83.         $sql=$sql."order by price";break;
84.     case 7:
85.         $sql=$sql."order by total";break;
86.     case 8:
87.         $sql=$sql."order by stock";break;
88. }
89.
90. $result = $conn->query($sql);

```

通过检测前端发回的信息，构建含有特定搜索条件的 `select` 语句并通过 `query` 函数执行。

## 5. Log-in 管理员登录界面

通过前端 HTML 构建输入条件（图 7），传送至后端 PHP 在 MySQL 中

admin 表内执行查找操作，若查找用户名和密码匹配，则登陆成功并获得主界面地址（图 8），若查找失败，查无此人，则弹出错误信息，并返回登录界面（图 9）。

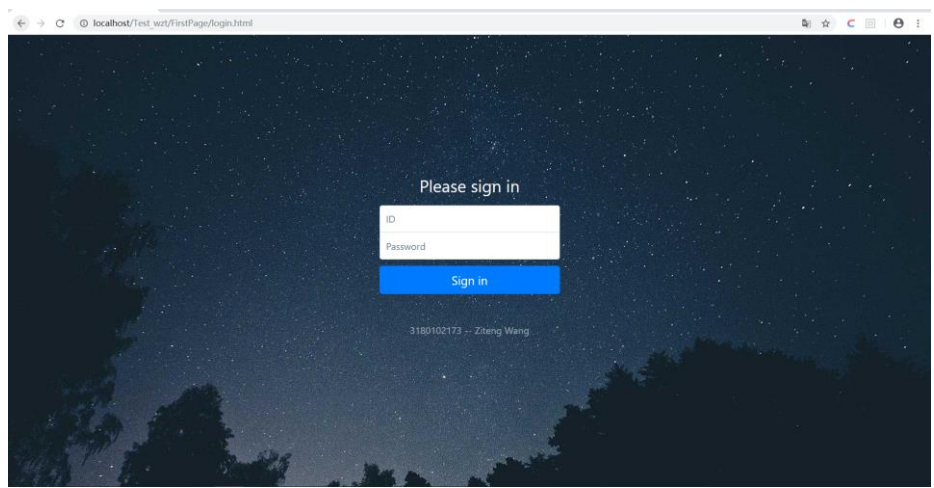


图 7

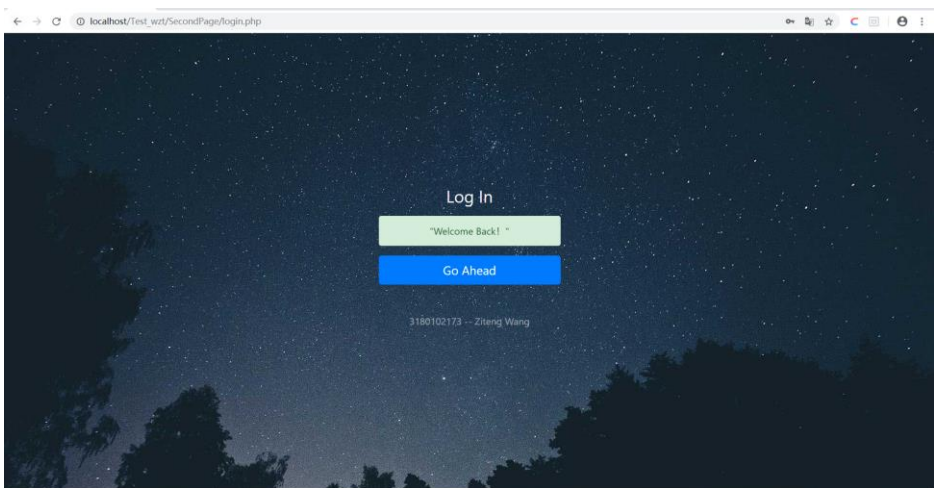


图 8

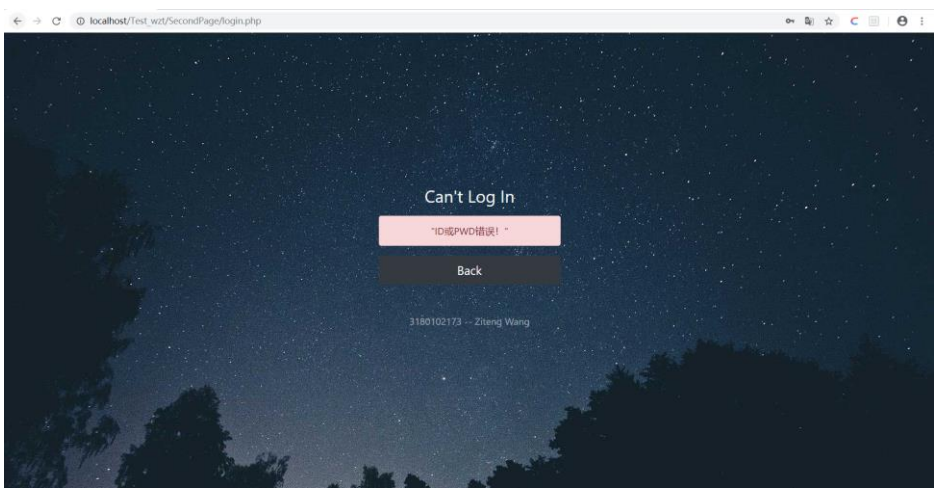


图 9

后端 login.php 文件的 sql 语句构建如下：

```
1. $id = $_POST['id'];
2. $pwd = $_POST['pwd'];
3. $url = "search.php";
4. $_SESSION['id']=$id;
5.
6. $sql = "select * from admin where admin_id='".$id.'" and password = '".$pwd.'" ";
7. $result = $conn->query($sql);
8. $info = mysqli_fetch_array($result);
```

## 6. MainUI-Admin 管理员主界面

本部分为整个系统的核心架构，包含图书入库、查询、借书、还书、借书证管理等功能。

### 1) Append 图书入库

界面如图 10 所示，有两种上传方式，分为“单本导入”和“批量导入”。填写表中信息并点击 **Append** 后，可将单本图书信息通过后端 PHP 文件导入数据库，并反馈处理信息，若处理失败则返回失败原因，成功则返回成功提示（图 11）。当点击 **Upload File** 时即可将 SecondPage 中的 1.xls 文件导入数据库，并返回导入信息（图 12）。

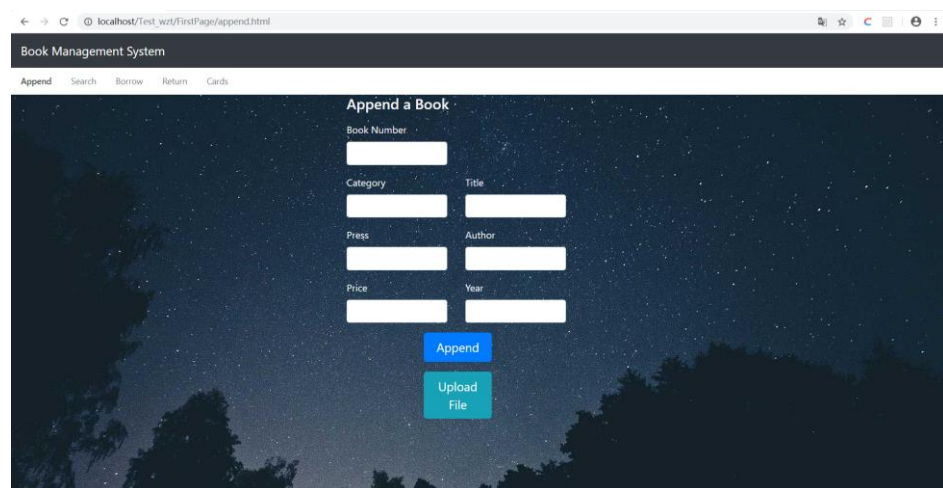


图 10

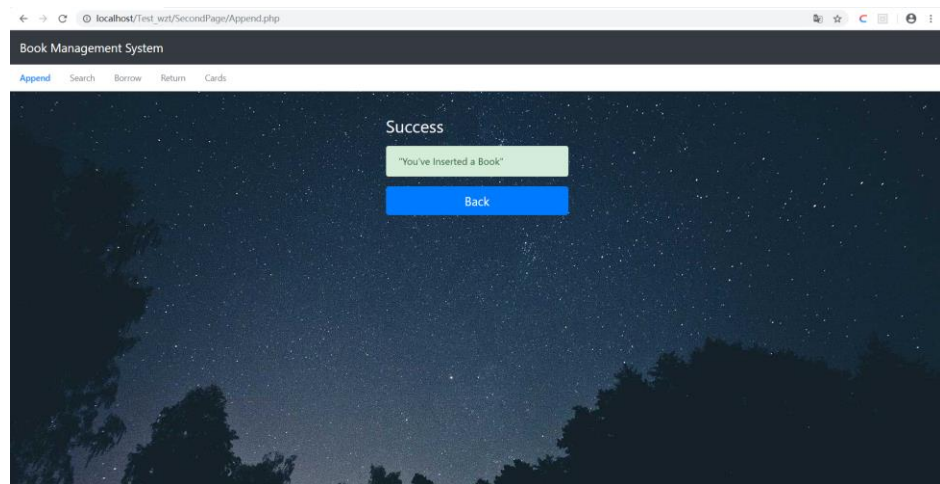


图 11

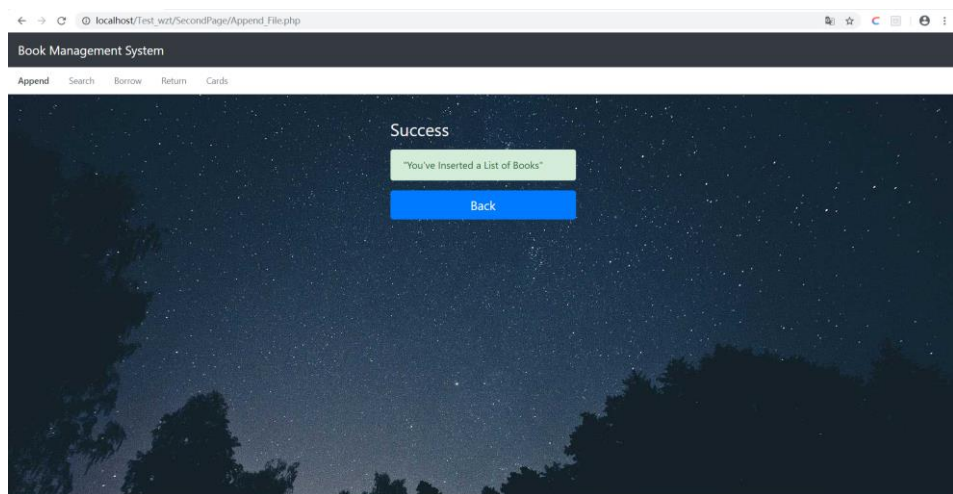


图 12

其中单本加入的 PHP 处理代码如下：

```
1. $sql = "select * from book where bno='".$bno."'";
2. $ERROR = false;
3.
4. $result = $conn->query($sql);
5.
6. if($row=$result->fetch_object())
7. {
8.     $row->total++;
9.     $row->stock++;
10. }
11. else
12. {
13.     $sql = "insert into book values('".$bno."','".$cat."','".$tit."','".$pre
        .",".$yea."','".$aut."','".$pri.",1,1)";
14.     $result = $conn->query($sql);
```

```

15.     if(!$result)
16.     {
17.         $ERROR = true;
18.     }
19. }

```

## 2) Search 图书查找

详细实现部分与第 4 模块中的 `search_user` 相同，在此不再赘述，不同之处在于上方的导航栏连接生效，如图 13 所示。

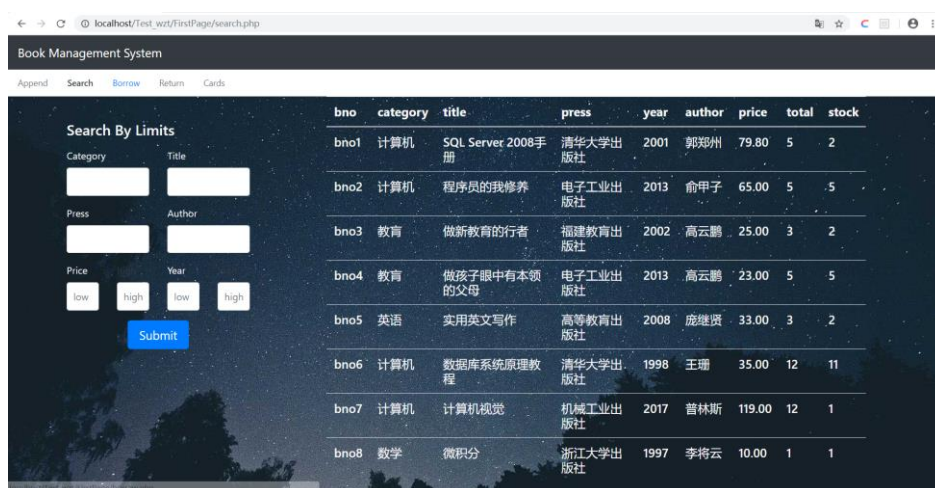


图 13

## 3) Borrow 借书模块

本模块有两个功能，一是查询借书证所有借书记录，输入相应的借书证号就可以显示出所有借书记录（图 14）；二是借书功能，输入借书证号和图书号，点击 `submit` 就可以实现借书请求，实现情况如图 15 所示，根据反馈信息看出是否成功借书。

对于借书错误的情况：①借书证不存在（图 16）②该用户已借过这本书③本书库存为 0（图 17）④SQL 语句执行失败。分别对四种情况进行不同反馈，实验选取 1,3 两种情况。



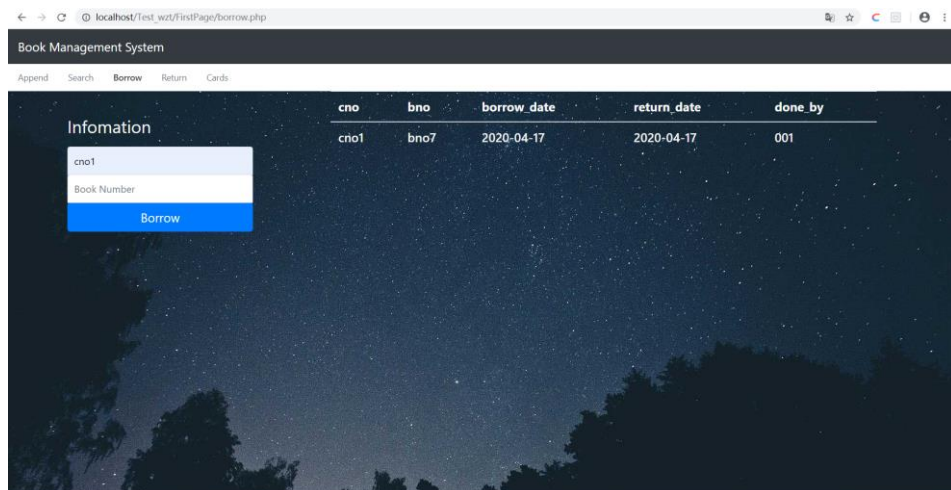


图 14

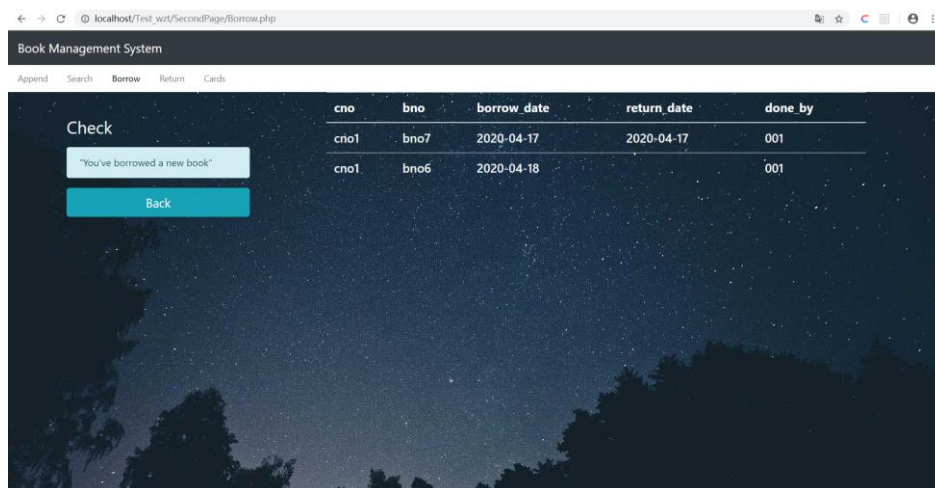


图 15

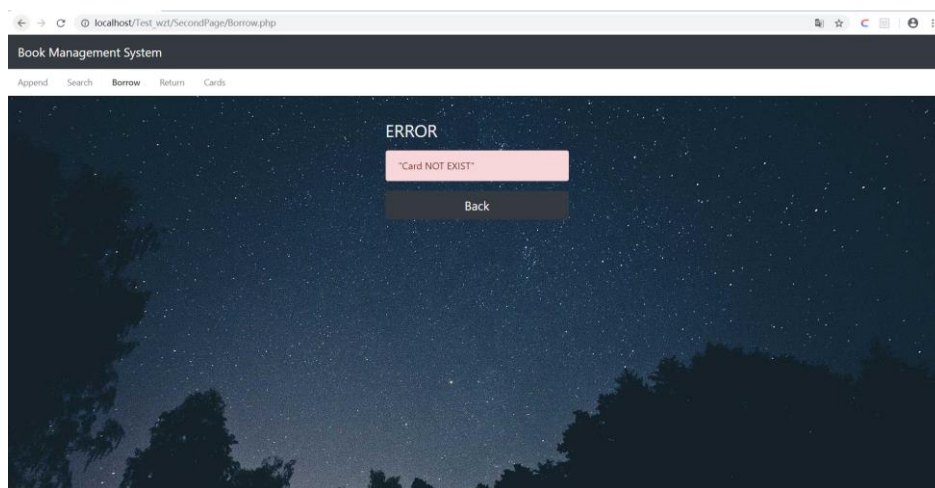


图 16

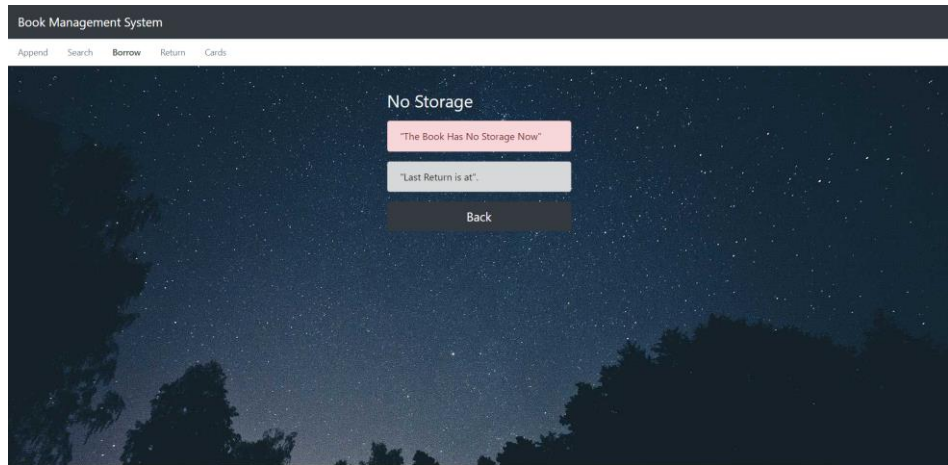


图 17

PHP 核心代码如下：

```
1. $cno = $_POST['cno'];
2. $bno = $_POST['bno'];
3. $showcard = false;
4. $ERROR=0;
5.
6. $sql = "select * from card where cno='".$cno."'";
7. $result=$conn->query($sql);
8. $row=$result->fetch_array();
9. if(!$row)
10. {
11.     $ERROR=1; // 借书证不存在
12. }
13. else if($bno=="") // Show borrowed by cno
14. {
15.     $showcard = true;
16. }
17. else{ // Borrow book
18.     $sql="select * from borrow where cno='".$cno.'" and bno='".$bno.'" and r
        eturn_date is null";
19.     $result=$conn->query($sql);
20.
21.     if($result->fetch_object()){
22.         $ERROR=2; // 该用户已借过这本书了
23.     }
24.     else{
25.         $sql="select * from book where bno='".$bno."'";
26.         $result=$conn->query($sql);
27.         $check=$result->fetch_object();
28.         if($check->stock<=0)
```



```

29.     {
30.         $ERROR=3; // 库存 0
31.     }
32.     else{
33.         $sql="update book set stock = stock-1 where bno='".$bno.'"";
34.         if(!($result=$conn->query($sql)))
35.         {
36.             $ERROR=4; // SQL 查询失败
37.         }
38.         else
39.         { $sql="insert into borrow values('".$cno."','".$bno."','".$date("Y-m-d")."',null,'".$id."'");
40.           if(!($result=$conn->query($sql)))
41.           {
42.               $ERROR=4; // SQL 查询失败
43.           }
44.         }
45.     }
46. }
47.
48. }

```

#### 4) Return 还书模块

还书部分实现与借书部分相似，去掉了对书存量的检查（还书只会让 **stock** 正增加），加入了查询所归还的借书记录是否存在，并对所处理的借书记录补充了 **return\_date** 信息，表示书已归还（图 18）。

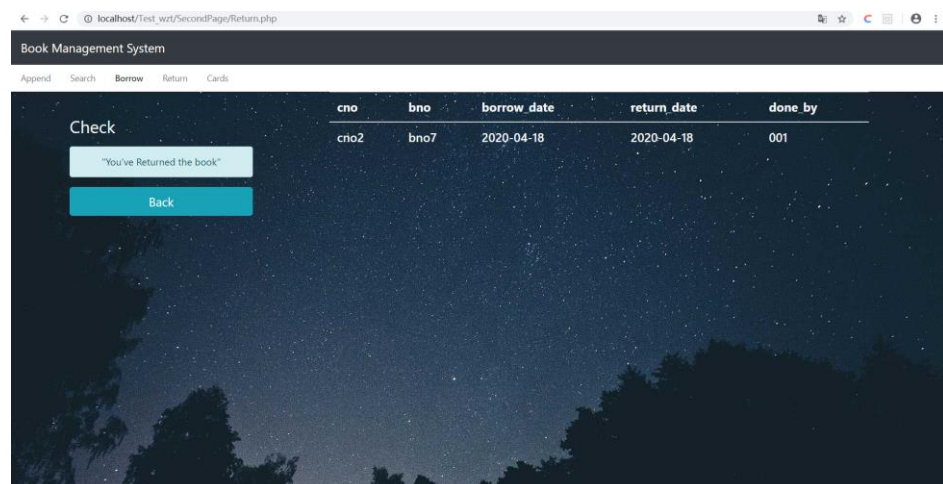


图 18

检查借书记录：

```

1. $sql="select * from borrow where cno='".$cno.'" and bno='".$bno.'" and return_date is null";
2. $result=$conn->query($sql);
3. $row=$result->fetch_array();
4.
5. if(!$row){
6.     $ERROR=2;// 该用户未借过这本书
7. }
8. else{
9.     $sql="update book set stock = stock+1 where bno='".$bno.'"";
10.    if(!($result=$conn->query($sql)))
11.    {
12.        $ERROR=3;// SQL 查询失败
13.    }
14.    else
15.    {
16.        $sql="update borrow set return_date='".$date("Y-m-d")."' where cno='".$cno.'" and bno='".$bno.'"";
17.        if(!($result=$conn->query($sql)))
18.        {
19.            $ERROR=3;// SQL 查询失败
20.        }
21.    }
22. }

```

借/还书流程图:

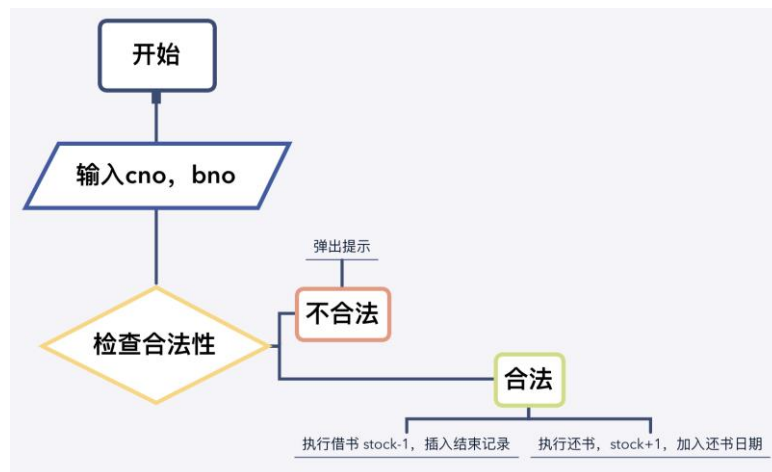


图 19

## 5) Cards 借书证管理

界面如图 20 所示, 通过选择操作 Append 或 Delete 来对借书证进行增加或删除, 为防止管理员操作失误误删, 系统设置需要将所有信息填写准确后才会向后端发出操作信号。

错误信息：①加入已存在的卡②删除不存在的卡③删除有书未还的卡，展示 1,3 错误反馈，如图 21,22。

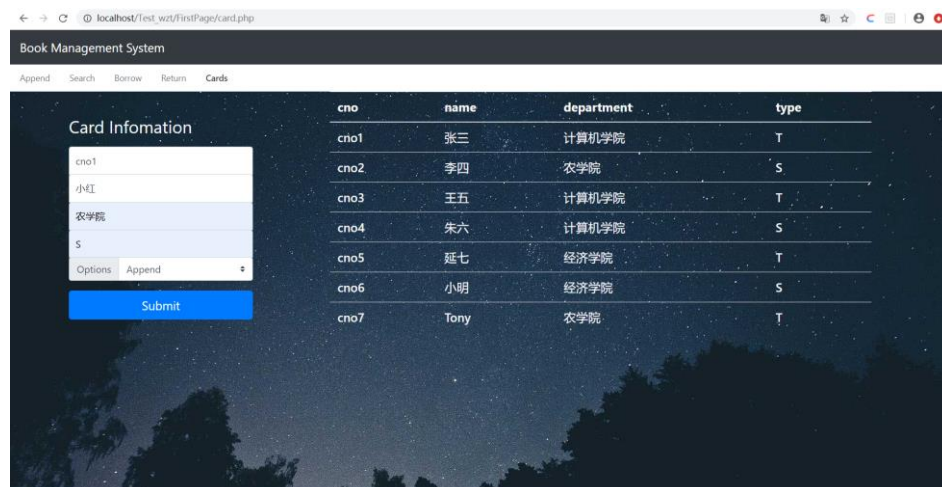


图 20（错误输入示例）

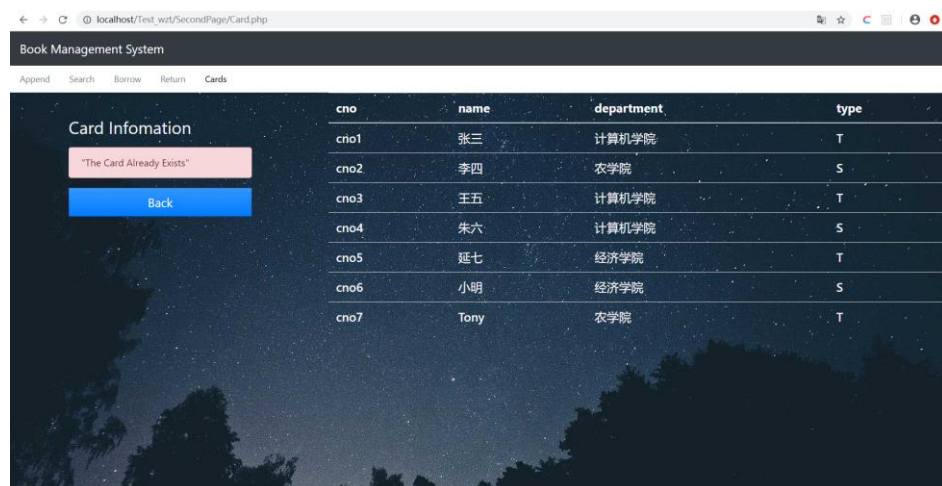


图 21

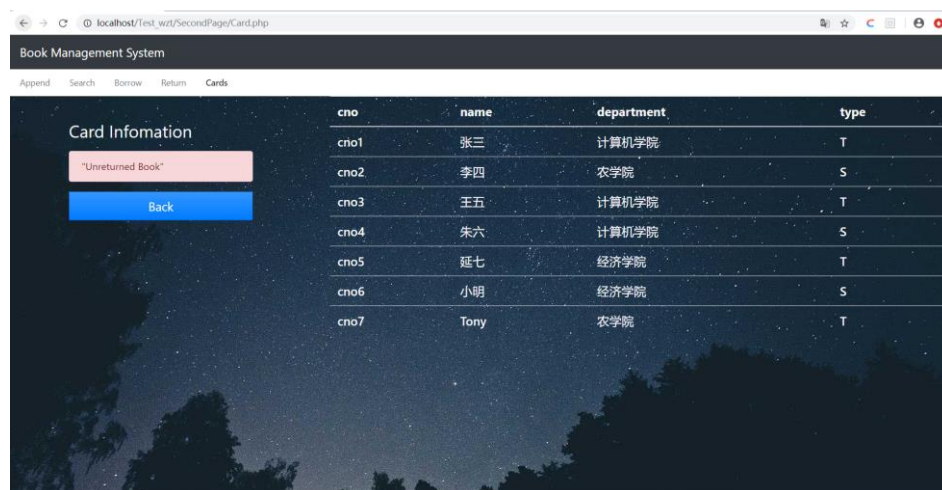


图 22

## 五、 技术栈

编程语言：HTML + JavaScript + JQuery + PHP + CSS

库框架：Bootstrap

## 六、 实验结果与分析

实现了题目要求的所有条件并加入了人性化设计，因为对技术的实现不够熟练，算是边学边试，在运行时遇到过很多次 bug，下面简单说两个错误和最终处理对策：

1. Mysqli 连接 MySQL 数据库失败：因为 WampServer 自带一个低版本的数据库，并且默认端口号也为 3306，与之前安装的冲突，导致连接失败。对此我修改了低版本数据库的 my.ini 文件，将端口号切换到 3307，并在 PHP 的 mysqli 连接函数中加入了“3306”端口号，用以正确连接原数据库；
2. SQL 语句查询失败：这个问题困扰了很久，后来根据错误信息发现是因为 post 方式返回的值在构建 sql 语句时没有自动加上引号，对此只需要在 \$sql 语句连接字符串变量时在两端补全引号就可以了。

## 七、 讨论、心得

本次实验总共耗时四天半，从一开始的迷茫，并开始学习 PHP 和 HTML 编程，到整体架构思路清晰，期间也遇到了很多很多问题，但欣慰的是最终都一一解决了，很开心自己花了时间取得了效果，更重要的是为自己的技术栈里增加了一碗饭，学会了最基本的前后端交互式开发。