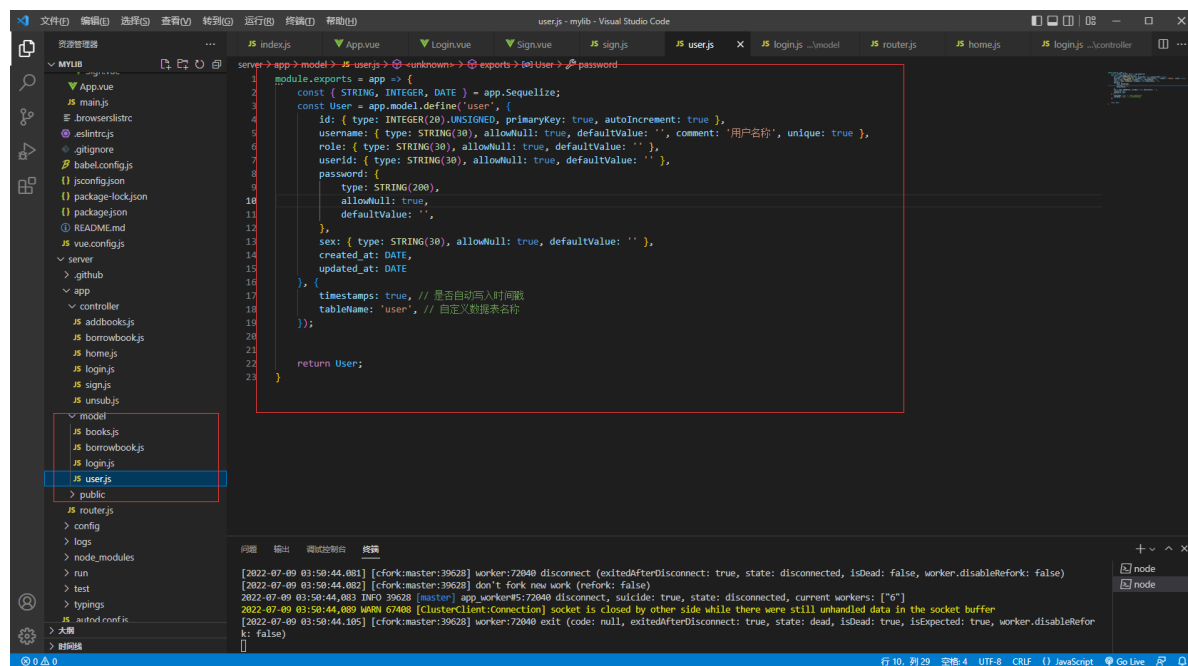


# 《通信软件开发与应用》课程结业报告

本人做的是一个基于前后端交互的一个图书管理系统，前端的页面方面我使用的是Vue2框架进行搭建，配合使用前端热门组件Element-ui来进行一定的样式美化，并配合node环境下的Egg.js框架搭建一个简单的后台，同时使用axios来实现前后端的数据交互，然后通过后台调用MYSQL数据库，实现简单的增删改查功能。本系统拥有两种不同的用户，一种是普通的用户，可以注册使用该系统的查看书籍、借阅书籍、归还书籍、修改密码等，还有一种是管理人员，可以导入书籍、增加用户、修改用户、查看借阅书籍的信息等。

## 1、登录注册

首先进入本系统可以看到的是一个登录页面，唯一ID是标识用户的，当登录时，唯一ID不存在会直接提示用户不存在，这样一步直接通过axios向后台发起post请求，将登录人员数据在后台调用数据库进行循环比对，然后在.then()里面进行页面跳转，由于后台这里我们没有设置jwt验证，所有前端发送的请求都是不带token的，登录和注册安全存在问题安全存在问题。同时为了调用数据库，我们还要做一个数据的映射模型。



Visual Studio Code editor showing the code for the login.js controller. The code defines a login class extending Controller, with methods for login, create, and destroy. The login method checks the user's role and password. The create method creates a new user. The destroy method destroys a user.

```
class login extends Controller {
  async index() {
    let id = this.ctx.request.query.id;
    let loginList = await this.app.model.User.findAll();
    this.ctx.body = loginList;
  }

  async create() {
    let login = this.ctx.request.body.login;
    let loginList = await this.app.model.User.findAll();
    let a = '没有';
    loginList.forEach(item => {
      if (login.role == '用户') {
        if (login.username == item.username && login.password == item.password && login.role == item.role && login.sex == item.sex && login.us...
          a = '用户';
        }
      } else if (login.role == '后台工作人员') {
        if (login.username == item.username && login.password == item.password && login.role == item.role && login.sex == item.sex && login.us...
          a = '后台工作人员';
        }
      }
    });
    this.ctx.body = a;
  }

  async destroy() {
    let id = this.ctx.params.id;
    await this.app.model.User.destroy({
      where: {
        id: id
      }
    });
  }
}
```

### 登陆

用户名

唯一ID

密码

性别

角色

68.43.83:8080/Login

192.168.43.83:8080 显示  
查无此人, 请重新输入

用户名

唯一ID

密码

性别

角色

普通用户不存在时，可以注册成为该系统的普通用户。

## 注册

用户名

6

唯一ID

6

密码

123

性别

男

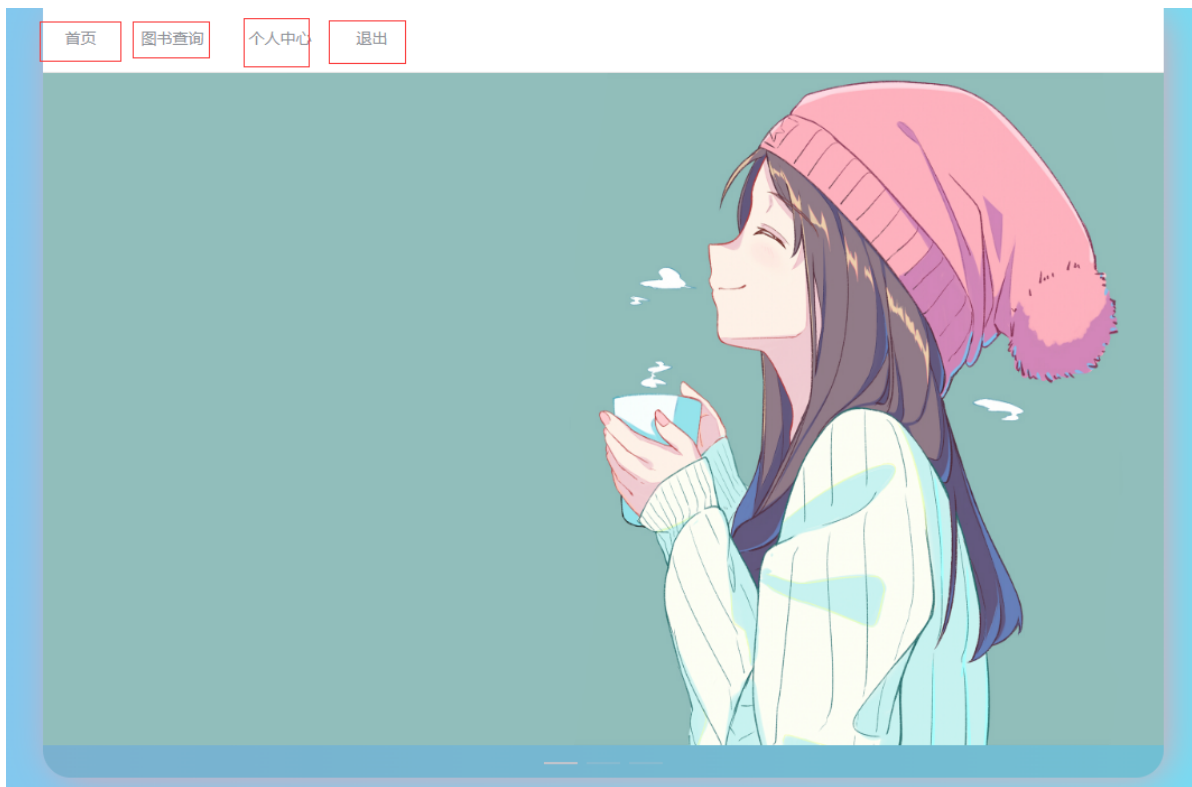
注册

返回

(只读)								限制行 第	
<input type="checkbox"/>	id	username	role	userid	password	sex	created_at	updated_at	
<input type="checkbox"/>	1	1	用户	1	12	男	2022-07-07 13:56:53	2022-07-07 14:04:31	
<input type="checkbox"/>	3	2	后台工作人员	2	2	男	2022-07-07 18:49:28	2022-07-07 18:49:28	
<input type="checkbox"/>	5	3	用户	3	3	男	2022-07-08 23:04:35	2022-07-08 23:04:35	
<input type="checkbox"/>	6	4	后台工作人员	4	4	女	2022-07-08 23:06:25	2022-07-08 23:06:25	
<input type="checkbox"/>	7	5	用户	5	5	男	2022-07-08 23:07:40	2022-07-08 23:07:40	
<input type="checkbox"/>	8	6	用户	6	123	男	2022-07-09 03:07:17	2022-07-09 03:07:17	

## 2、普通用户

然后我们就可以登录该系统进行使用了，进来之后就可以看到图书馆的首页，这是一个轮播图，然后头部导航条，导航条的内容我写在个人的页面中，然后，首页轮播图、图书查询、个人中心这些功能页面我都是写在组件上面的，父组件要使用子组件就需要先注册组件，并且前端的页面跳转还需要设置前端路由，但是这里我没有设置路由守卫，所有直接通过不同页面的链接就可以直接访问。



```
demo > src > views > pubuser.vue > {} script > default > methods > exitdocument

1 <template>
2   <div class="app">
3     <!-- 导航条 -->
4     <el-menu class="el-menu-demo" mode="horizontal">
5       <el-menu-item index="1" @click="doshow1">首页</el-menu-item>
6       <el-menu-item index="2" @click="doshow">图书查询</el-menu-item>
7       <el-menu-item index="3" @click="doshow2">个人中心</el-menu-item>
8       <el-menu-item index="4" @click="exitdocument">退出</el-menu-item>
9     </el-menu>
10
11     <!-- 轮播图 -->
12     <BannerFiguer v-if="show1"></BannerFiguer>
13
14     <!-- 图书列表 -->
15     <SelectBook v-if="show"></SelectBook>
16
17     <!-- 个人界面 -->
18     <PersonalView v-if="show2"></PersonalView>
19
20   </div>
21 </template>
22 <script>
23   //import axios from 'axios';
24   import SelectBook from '../components/SelectBook.vue';
25   import BannerFiguer from '../components/BannerFiguer.vue';
26   import PersonalView from '../components/PersonalView.vue';
27   export default {
28     components: {
29       SelectBook: SelectBook,
30       BannerFiguer: BannerFiguer,
31       PersonalView: PersonalView,
32     },
33     data() {
34       return {
```

图书查询功能，我们可以实现按照不同类别进行查询，如图书的名字、作者、类别等，这一步我直接调用书籍表中的书籍，进行循环遍历即可。

首页

图书查询

个人中心

退出

请选择

请输入内容

Q

图书名字

作者

类别

		作者	类别	状态	操作
		1	1	以借阅	<a href="#">借阅</a> <a href="#">退订</a>
2	2	2	2	未借出	<a href="#">借阅</a> <a href="#">退订</a>
3	3	3	3	以借阅	<a href="#">借阅</a> <a href="#">退订</a>
4	4	4	4	未借出	<a href="#">借阅</a> <a href="#">退订</a>
5	5	5	5	未借出	<a href="#">借阅</a> <a href="#">退订</a>
5	2	3	4	未借出	<a href="#">借阅</a> <a href="#">退订</a>
3	1	3	2	未借出	<a href="#">借阅</a> <a href="#">退订</a>
2	3	3	2	未借出	<a href="#">借阅</a> <a href="#">退订</a>
6	3	3	1	未借出	<a href="#">借阅</a> <a href="#">退订</a>

首页

图书查询

个人中心

退出

图书名字

▼

2

Q

图书名字	位置	作者	类别	状态	操作
2	2	2	2	未借出	<a href="#">借阅</a> <a href="#">退订</a>
2	3	3	2	未借出	<a href="#">借阅</a> <a href="#">退订</a>

本系统还可以进行图书的订阅和退订，当已经订阅时会显示以订阅，为订阅时会显示为订阅。这一步就是我将订阅的书籍存入另外一个表中，然后修改原书籍表中书籍状态，同样，退订的时候，将退订书籍从订阅表中删除，同时修改原书籍表中的书籍状态。

图书名字	位置	作者	类别	状态	操作
1	1	1	1	未借出	<a href="#">借阅</a> <a href="#">退订</a>
2	2	2	2	未借出	<a href="#">借阅</a> <a href="#">退订</a>

<input type="checkbox"/>	id	name	userid	position	author	type	state	avatar_url	created_at	updated_at
<input type="checkbox"/>	22	1	1	1	1	1	以借阅		2022-07-08 22:56:21	2022-07-08 22:56:21
<input type="checkbox"/>	23	3	1	3	3	3	以借阅		2022-07-08 22:56:27	2022-07-08 22:56:27

个人中心这里可以看到我的登录人员信息，这里我使用了本地存储，在登录的时候将用户名和ID进行本地存储，但是我认为这样做是不合理的，因为，本地存储不安全，同时，如我在同一台浏览器上进行第二个用户的登录，就会覆盖掉我前面一个人员的信息。

```

if (res.data == '用户') {
  localStorage.setItem('username', this.sub.username);
  localStorage.setItem('sex', this.sub.sex);
  localStorage.setItem('role', this.sub.role);
  localStorage.setItem('userid', this.sub.userid);
  this.$router.push('/PubUser');
}

```

[首页](#)
[图书查询](#)
[个人中心](#)
[退出](#)

个人中心

[修改密码](#)
[显示个人信息](#)
[隐藏个人信息](#)

🔍 用户名	****
🔍 唯一ID	****
角色	****
性别	****

个人中心

[修改密码](#)
[显示个人信息](#)
[隐藏个人信息](#)

🔍 用户名	6
🔍 唯一ID	6
角色	用户
性别	男

退出界面直接就退出到登录页面，这里我在退出时设置了清空本地存储。

```

},
exitdocument() {
  localStorage.clear();
  this.$router.push('/');
},

```

### 3、系统管理员

本系统用于系统管理员，系统管理员登录也是如上的登录页面，然后我们进入到主页，我们可以看到本系统有用图书管理，借阅书籍查询，人员管理等。

图书管理以借阅的图书人员管理中心退出

\*\*\*增加图书\*\*\*

图书名字	位置	作者	类别	状态	操作
1	1	1	1	未借出	删除
2	2	2	2	以借阅	删除
3	3	3	3	以借阅	删除
4	4	4	4	未借出	删除
5	5	5	5	未借出	删除
5	2	3	4	未借出	删除
3	1	3	2	未借出	删除
2	3	3	2	未借出	删除
6	3	3	1	未借出	删除

这里我们可以增加书籍，包括书籍的名称、书籍的位置、书籍的作者、借阅状态。

提示

图书名字

7

位置

7

作者

7

类别

7

立即创建

取消

7	7	7	7	未借出	删除
---	---	---	---	-----	----

<input type="checkbox"/>	id	name	position	author	type	state	avatar_url	created_at	updated_at
<input type="checkbox"/>	2	1	1	1	1	未借出		2022-07-07 18:40:21	2022-07-09 03:25:52
<input type="checkbox"/>	3	2	2	2	2	以借阅		2022-07-07 18:40:32	2022-07-09 03:25:41
<input type="checkbox"/>	4	3	3	3	3	以借阅		2022-07-07 18:40:43	2022-07-08 22:56:27
<input type="checkbox"/>	5	4	4	4	4	未借出		2022-07-07 18:40:53	2022-07-07 19:07:37
<input type="checkbox"/>	6	5	5	5	5	未借出		2022-07-07 18:41:05	2022-07-08 22:27:41
<input type="checkbox"/>	7	5	2	3	4	未借出		2022-07-09 03:16:42	2022-07-09 03:16:42
<input type="checkbox"/>	8	3	1	3	2	未借出		2022-07-09 03:17:04	2022-07-09 03:17:04
<input type="checkbox"/>	9	2	3	3	2	未借出		2022-07-09 03:17:12	2022-07-09 03:26:00
<input type="checkbox"/>	10	6	3	3	1	未借出		2022-07-09 03:17:26	2022-07-09 03:17:26
<input type="checkbox"/>	11	7	7	7	7	未借出		2022-07-09 03:39:11	2022-07-09 03:39:11

以借阅的图书我可以看到借阅者的ID，以及图书的详情等。

图书管理

以借阅的图书

人员管理中心

退出

图书名字	位置	作者	类别	状态	唯一ID
1	1	1	1	以借阅	1
3	3	3	3	以借阅	1

人员管理中心这里可以新增加用户，也可以增加管理员，当然也可以实现对用户的删除

\*\*\*增加人员\*\*\*

提示

名字

7

角色

用户

唯一ID

7

密码

7

性别

女

立即创建

取消

7	用户	7	7	女	删除
---	----	---	---	---	----



<input type="checkbox"/>	id	username	role	userid	password	sex	created_at	updated_at
<input type="checkbox"/>	1	1	用户	1	12	男	2022-07-07 13:56:53	2022-07-07 14:04:31
<input type="checkbox"/>	3	2	后台工作人员	2	2	男	2022-07-07 18:49:28	2022-07-07 18:49:28
<input type="checkbox"/>	5	3	用户	3	3	男	2022-07-08 23:04:35	2022-07-08 23:04:35
<input type="checkbox"/>	6	4	后台工作人员	4	4	女	2022-07-08 23:06:25	2022-07-08 23:06:25
<input type="checkbox"/>	7	5	用户	5	5	男	2022-07-08 23:07:40	2022-07-08 23:07:40
<input type="checkbox"/>	8	6	用户	6	123	男	2022-07-09 03:07:17	2022-07-09 03:07:17
<input type="checkbox"/>	9	7	用户	7	7	女	2022-07-09 03:45:25	2022-07-09 03:45:25

## 4、总结

本次是我第一次编写前后端结合的项目，在本次开发中，我遇到了许多的问题，比如，我并不太熟悉后台开发，我的后台开发是在网上学的一个速成的教学，所以我遇到了许多的问题，比如说，后台调用数据库，然后要写一个数据映射模型，那一步我跟着Egg.js的官方教程走，但是我感觉那个教程有点省略，大量JSON数据导入数据库时，那个实例代码写得比较模糊，最后还是试探出来的，然后就是前后端的交互，第一个接触后端，对应许多接口的写法还不是很熟悉。前端方面，我Vue也算是初学，我看了Vue那个的官方文档，按照那个教程来，Element-ui方面，我也是仅仅会简单复制官方文档提供的组件，虽然本项目比较简陋，而且BUG也比较多，但终究是我第一次写，我相信通过不断地学习，我也能够更加深入了解这些技术。