**数据库系统实验报告**

课程名称： 数据库系统原理 实验项目： 实验五 数据库管理系统设计

学生姓名： 庄毅非 专业： 软件工程 学号： 3200105872

指导老师： 苗晓晔 实验日期： 2022 年 4 月 9 日

1. **实验目的和要求**

掌握使用JDBC或ODBC进行数据库管理系统开发的能力

1. **实验内容和要求**
2. 设计图书管理系统的ER模型
3. 设计对应的实体和联系模式
4. 使用JDBC或ODBC实现一个简单的图书管理系统，能够对图书、借书证和图书借阅行为进行管理
5. **开发环境**
6. OS: MacOS Monterey 12.3.1
7. 数据库管理系统： Mysql
8. 实现框架:
   1. 后端： go/gorm, gin
   2. 前端： nextjs,material-ui
9. **程序用法和实验步骤**

**4.1 确定需要建立的数据表**

**4.1.1: Book表**

图书馆中有许多种类的藏书，每本书本身的属性有书号，类别，书名，出版社，年份，作者，价格， 总藏书量， 库存。图书管理员需要在图书上执行借阅和归还操作。

**4.1.2: Card表**

图书馆使用借书证管理用户的借阅信息，借书证上的信息包括卡号、姓名、单位、类别，图书管理员需要能够添加借书证，以及通过借书证号查询借书证。

**4.1.3 Borrow 表**

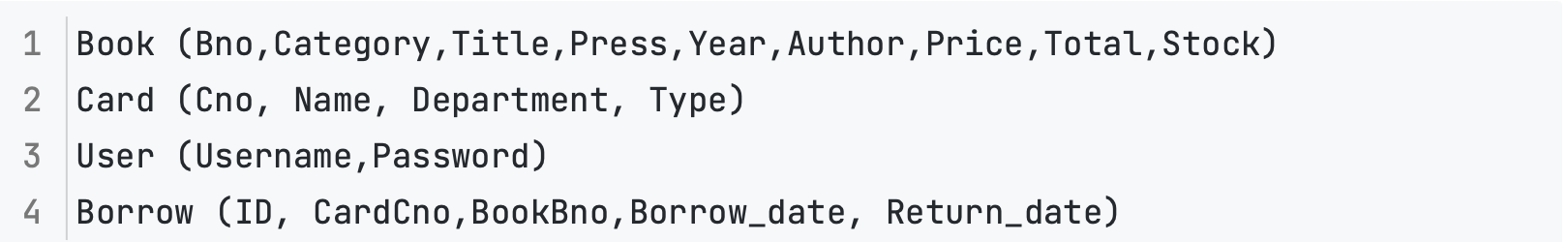
系统使用borrow表管理借阅行为，borrow表的信息包括书号，借书证号，借期和换期。图书管理员需要能够执行归还和借阅操作。

**4.2 建立关系模型**

最终建立的数据概念模型如下

|  |  |
| --- | --- |
| 对象名称 | 属性 |
| 书 | 书编号、类别、标题、出版社、年份、作者、价格、总藏书量、现有库存 |
| 借书证 | 借书证号、持卡者姓名、部门、类别 |
| 管理员 | 用户名、密码 |
| 借阅信息 | 自增编号，借书证号。书编号、借阅时间、归还时间 |

**4.3 在ODBC中建立对应模式**



**4.4 各模块功能设计**

|  |  |
| --- | --- |
| 模块名称 | 功能描述 |
| 管理员登陆 | 输入账号、密码登陆进系统，或者返回登陆失败，并显示失败原因 |
| 管理员注册 | 进行管理员账户的注册 |
| 图书入库 | 1. 单本入库 2. 批量入库 |
| 图书查询 | 要求对书的类别，书名，出版社，年份（区间），作者，价格，价格（区间）进行查询 |
| 借书 | 1. 输入借书证号 显示该借书证所有已经借的书籍 2. 输入书号 如果书籍还有库存，则借阅成功，否则输出书籍没有库存，输出最近一次归还的时间 |
| 还书 | 1. 输入借书证卡号 显示该借书证所有已经借的书籍 2. 输入书号 如果该书籍在已经借阅的书籍列表中，那么还书成功，否则输出出错信息 |
| 借书证管理 | 增加或删除借书证 |

**4.5 搭建数据库**

我选用的ODBC是gorm, 为了建立数据表，我在book\_backend/model文件夹中建立了Book.go, Card.go, User.go, Borrow.go四个表文件，内容如下

Book.go



Card.go



User.go



Borrow.go



**4.6 后端启动**

我选用的是gin网络框架搭建后端，这是一个golang的网络微框架，具有速度快，搭建路由简单等特性。在项目中，我首先通过gorm连接到数据库，之后在gin引擎添加各表功能对应的api，并监听9000端口，前端只需要访问9000端口即可获取所需信息。

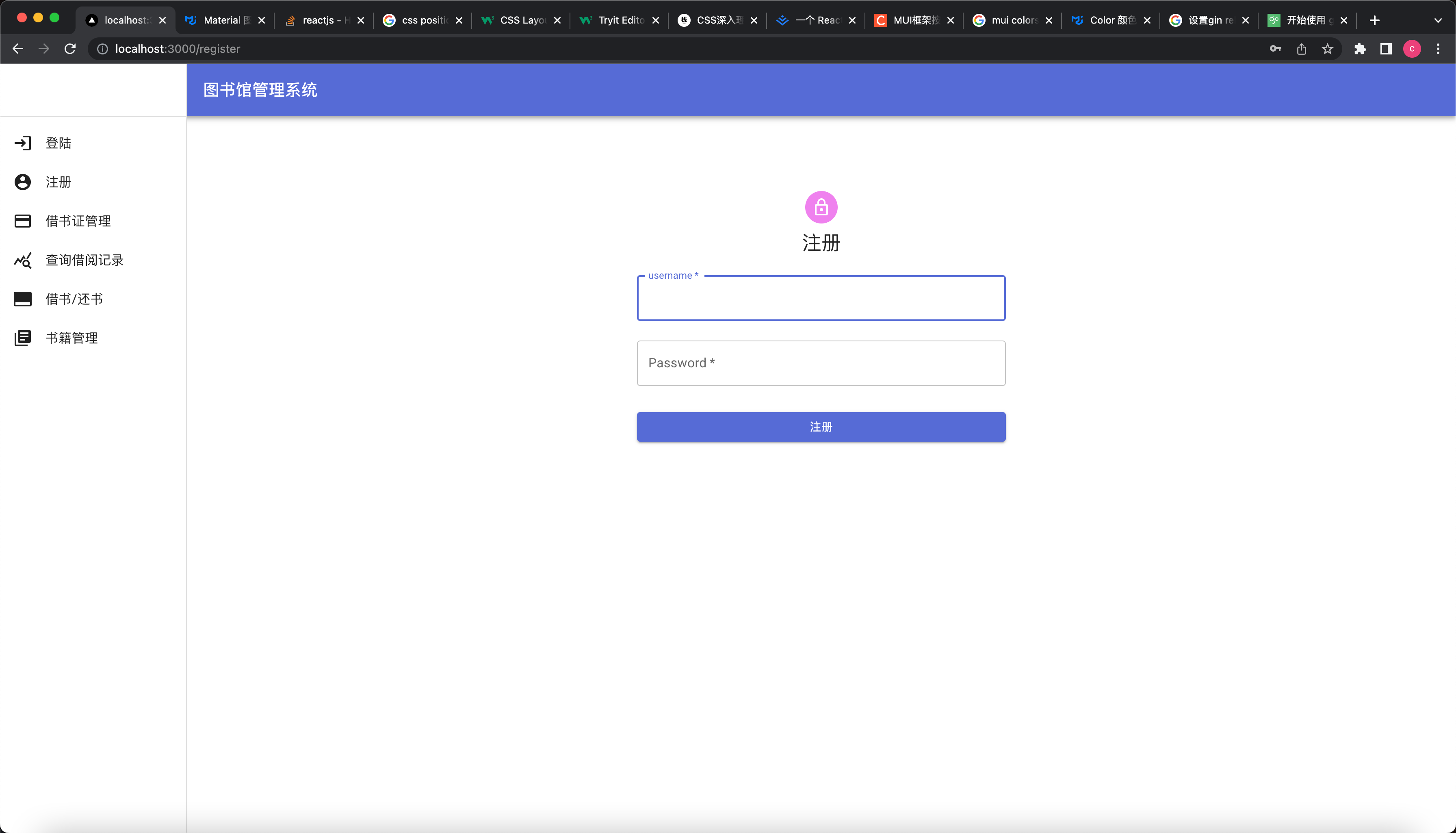
**4.7 前端启动**

我选用的是nextjs进行前端搭建，这是一个基于React和typescript的前端框架，能够智能读取页信息，快速进行网页搭建，具有灵活性高，性能强大的特点。

1. **实验结果和分析**

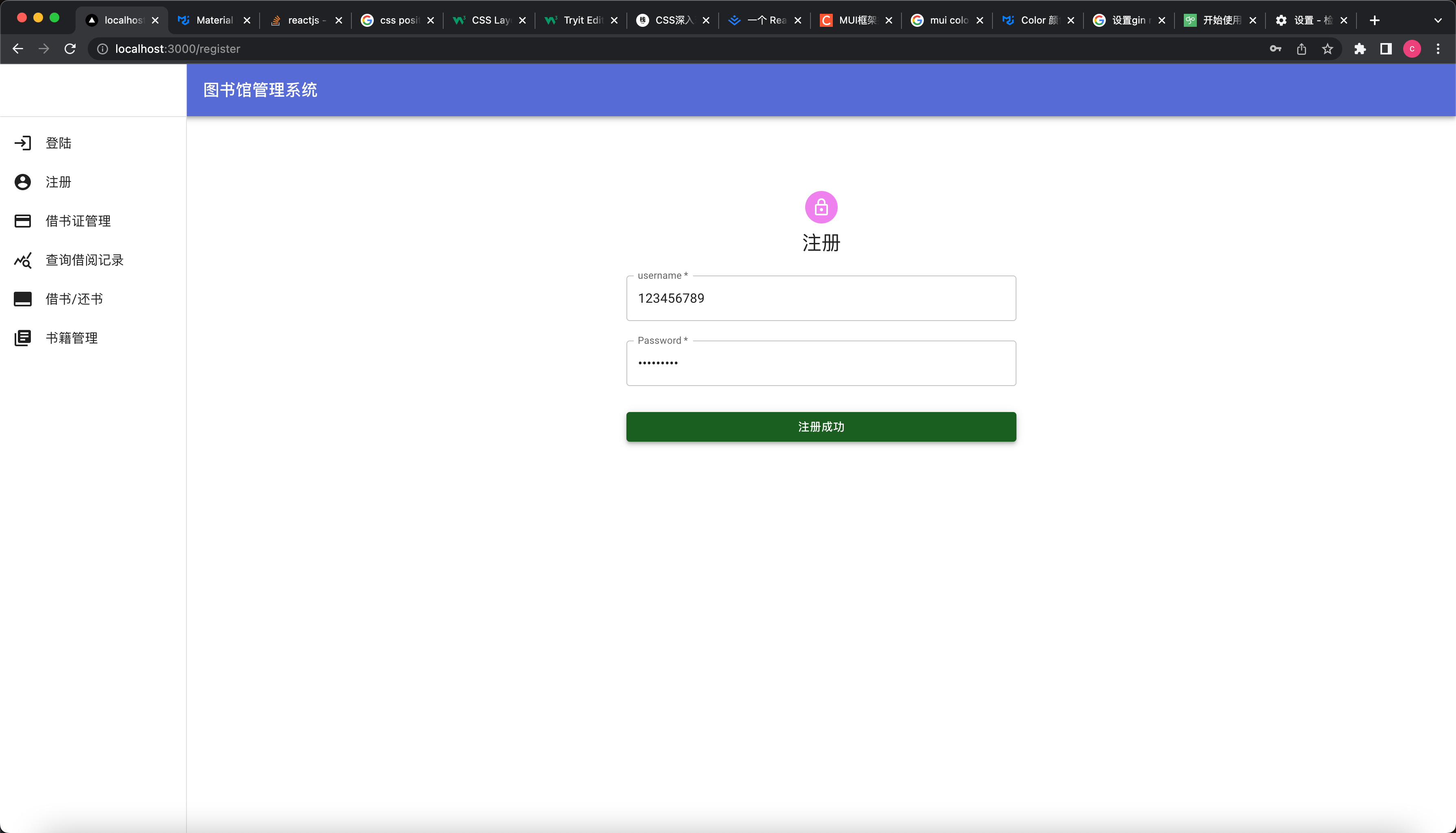
**如果您在测试过程中发现程序不能够正常运行，可能是因为jwt token已经过期，请您重新进行登陆，更新持有的jwt令牌**

**5.1 用户注册**

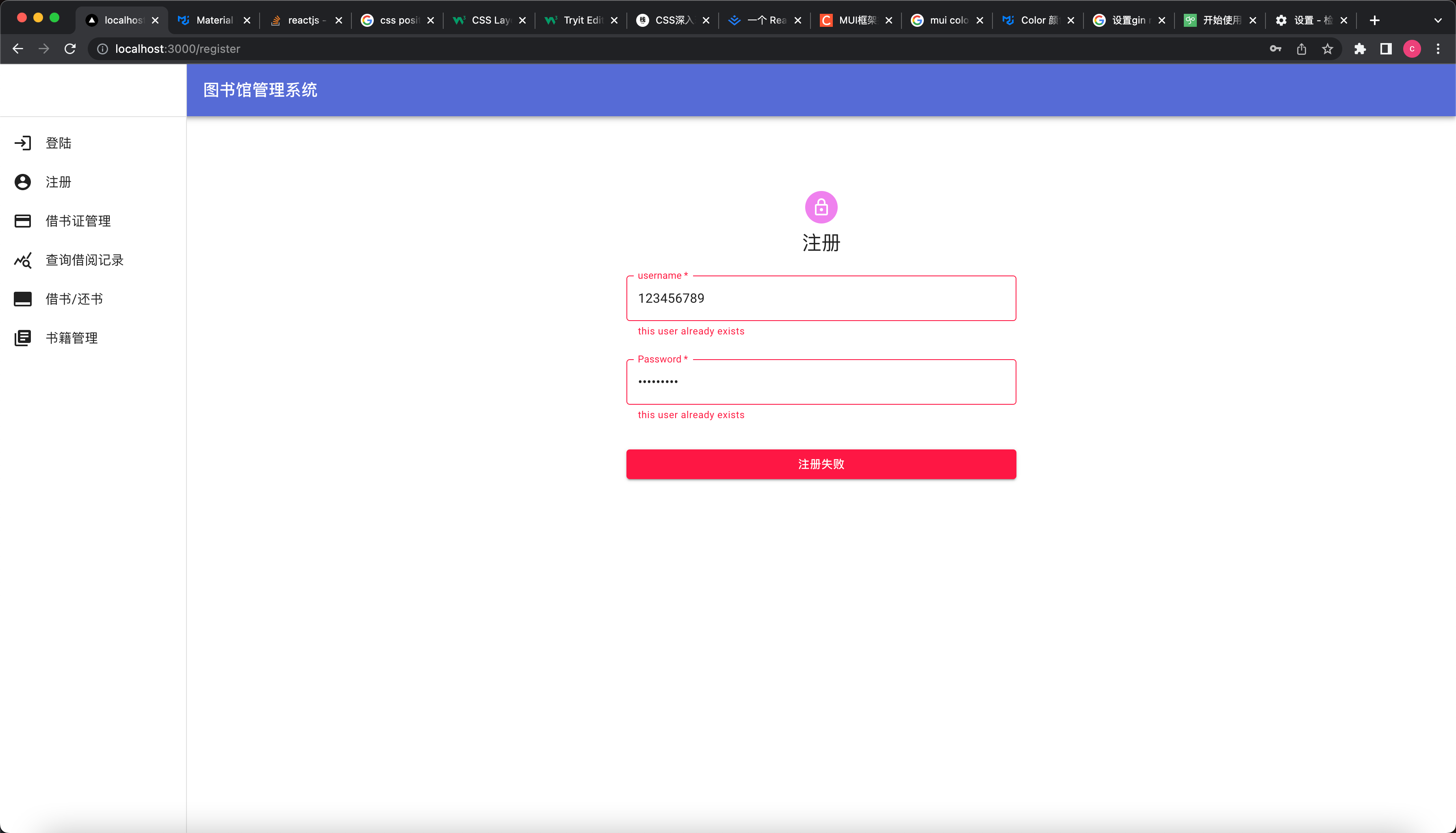


用户在输入账号和密码之后，点击Register按钮就可以进行注册，如果注册成功和失败都会回显信息。

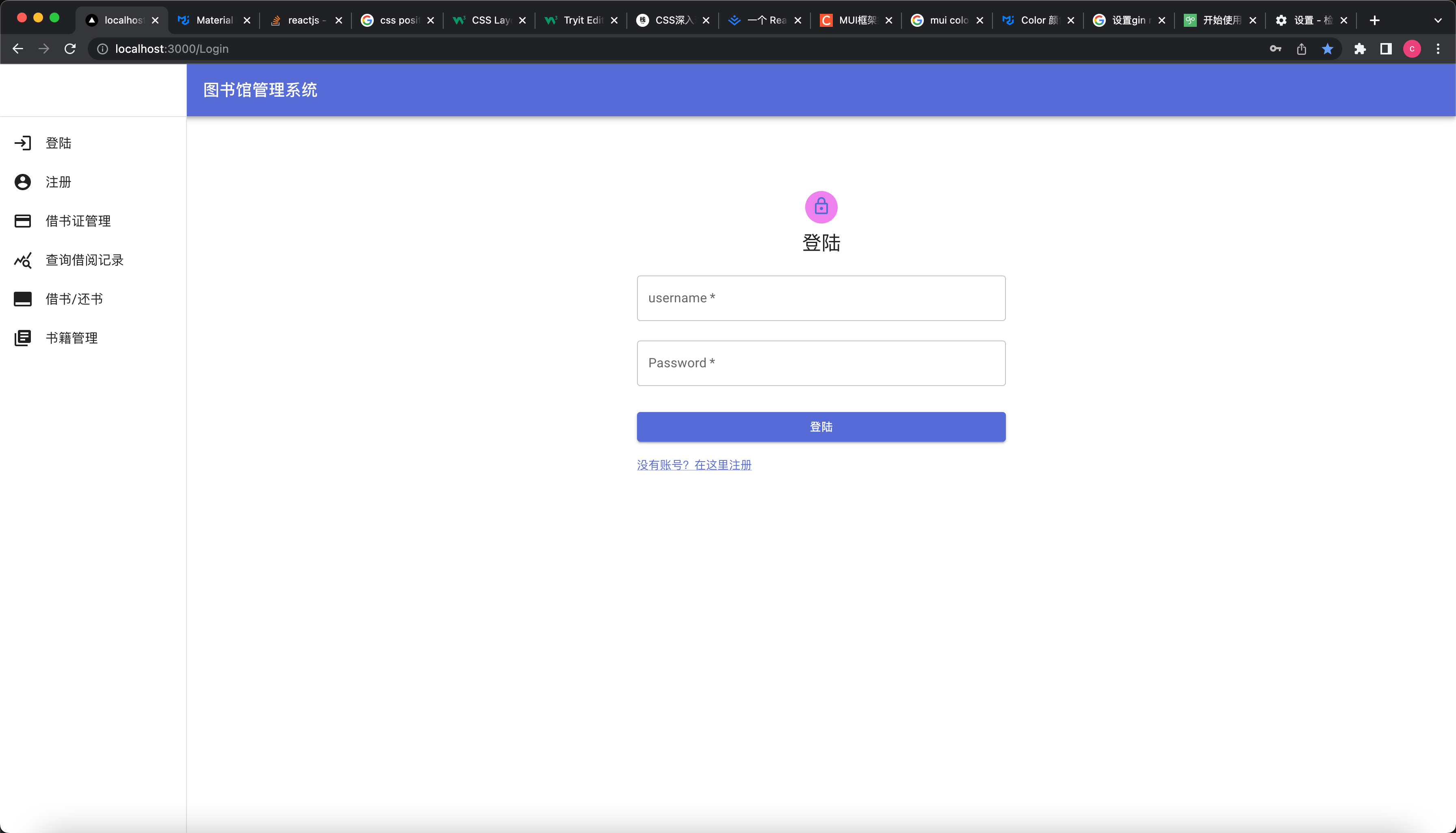
注册成功：



注册失败：



**5.2 用户登录**

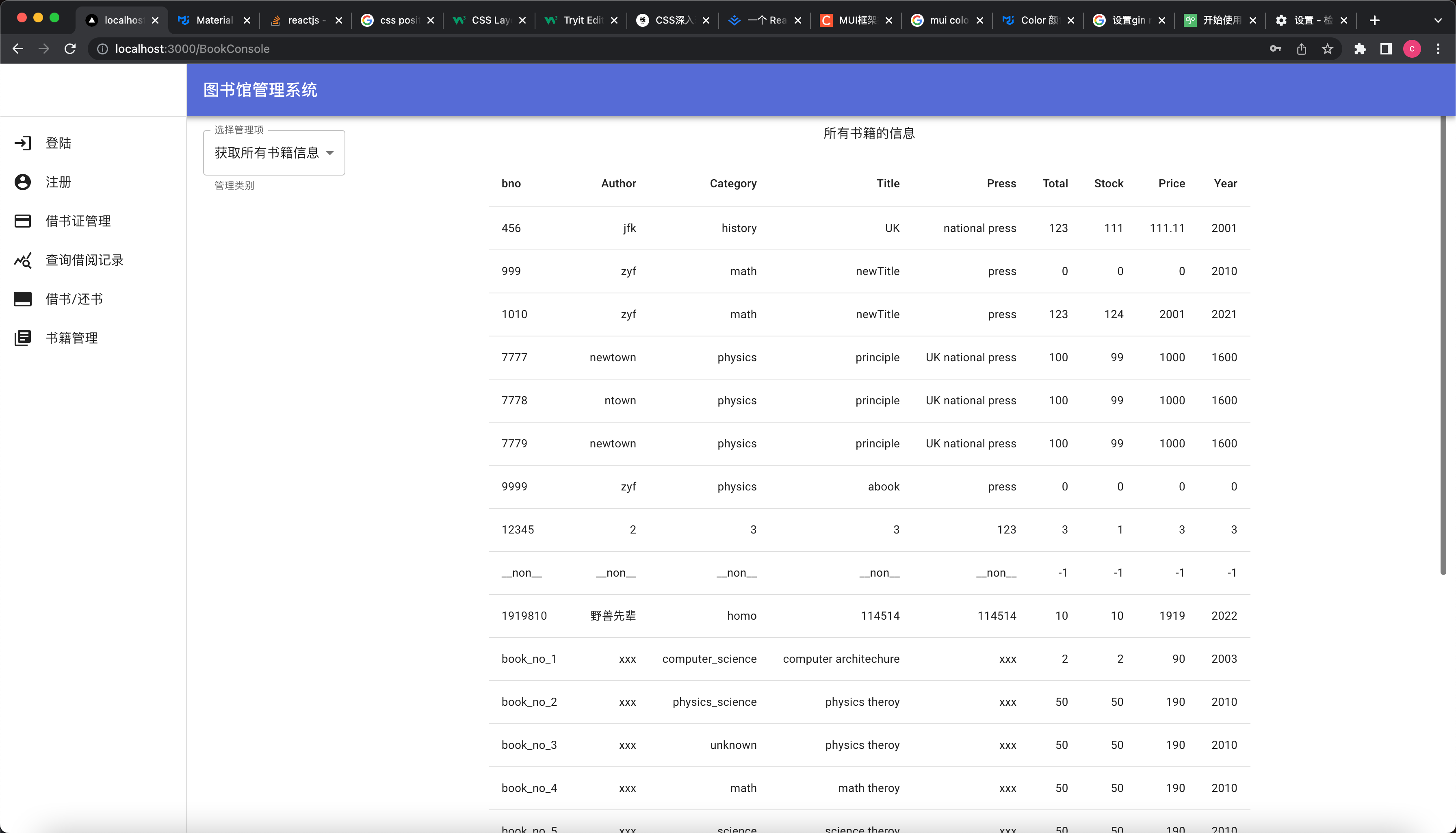


和注册一样，无论成功失败都会回显对应信息，这里限于篇幅不再展示。

值得注意的是，为了确保安全性，在登陆成功之后，后端会给前端返回一个jwt协议生成的token，添加到cookie中，用户只有持有上述token才能进行借书证管理，借阅信息管理等操作，token有效期为10分钟，超时需要重新登陆。

5.3 **图书管理**

**5.3.1 查询所有书籍的信息**

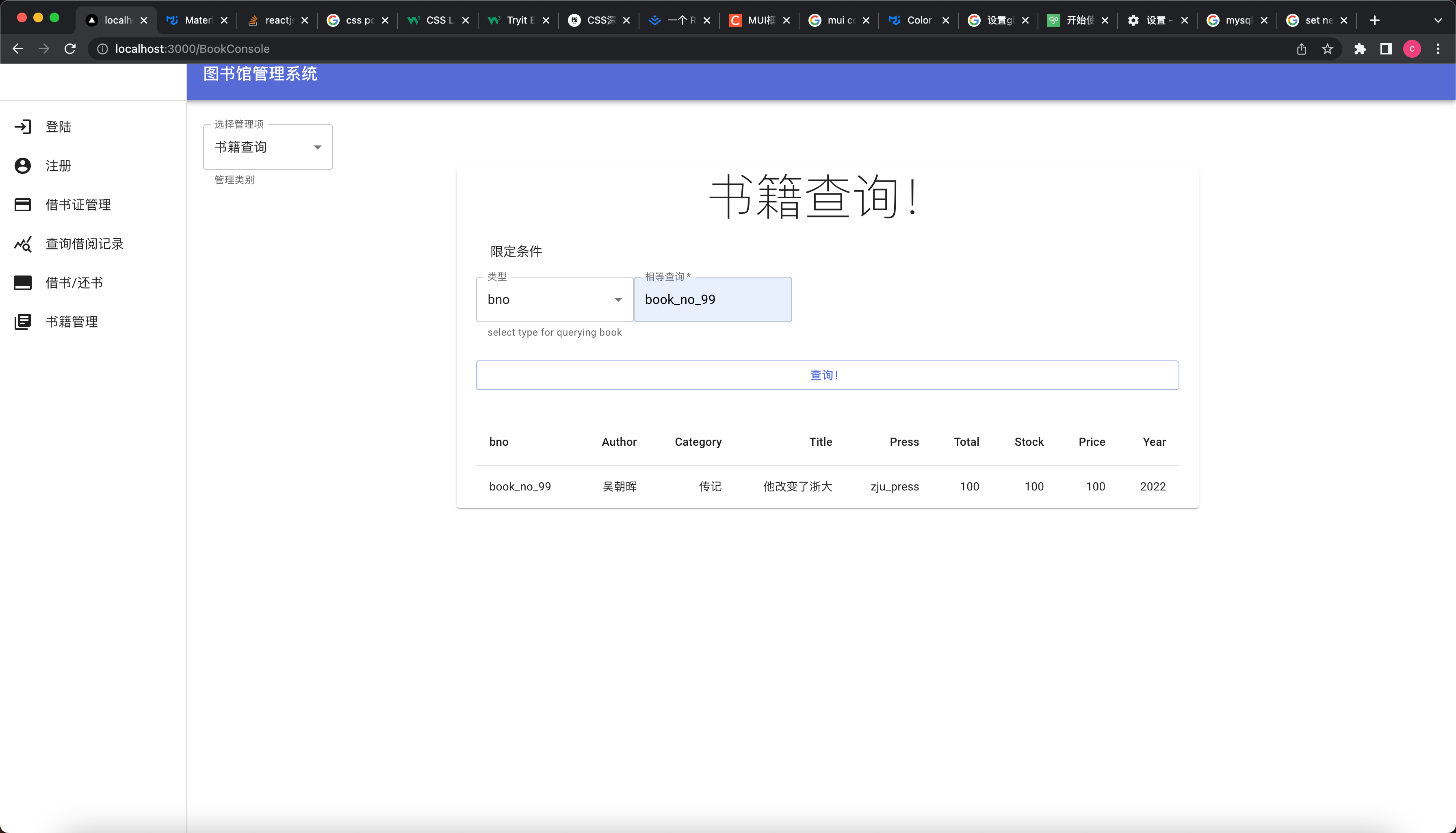


**5.3.2 更新书籍信息或加入新的书籍（图书入库）**

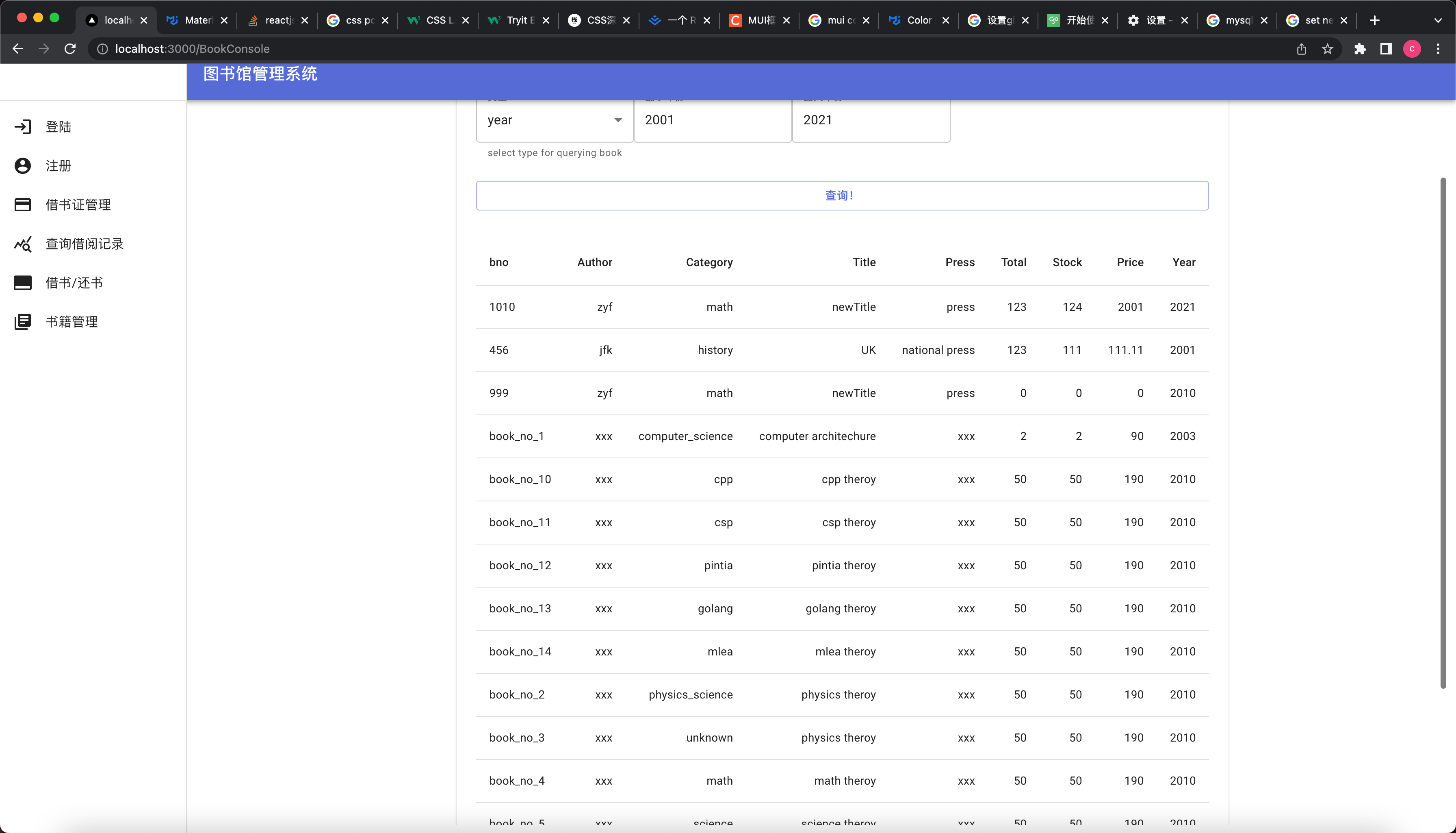
如果书籍是已经存在的，那么将其更新，否则视为插入新的书籍。

**5.3.3 图书查询**

相等查询，这里以上一步插入的数据为例

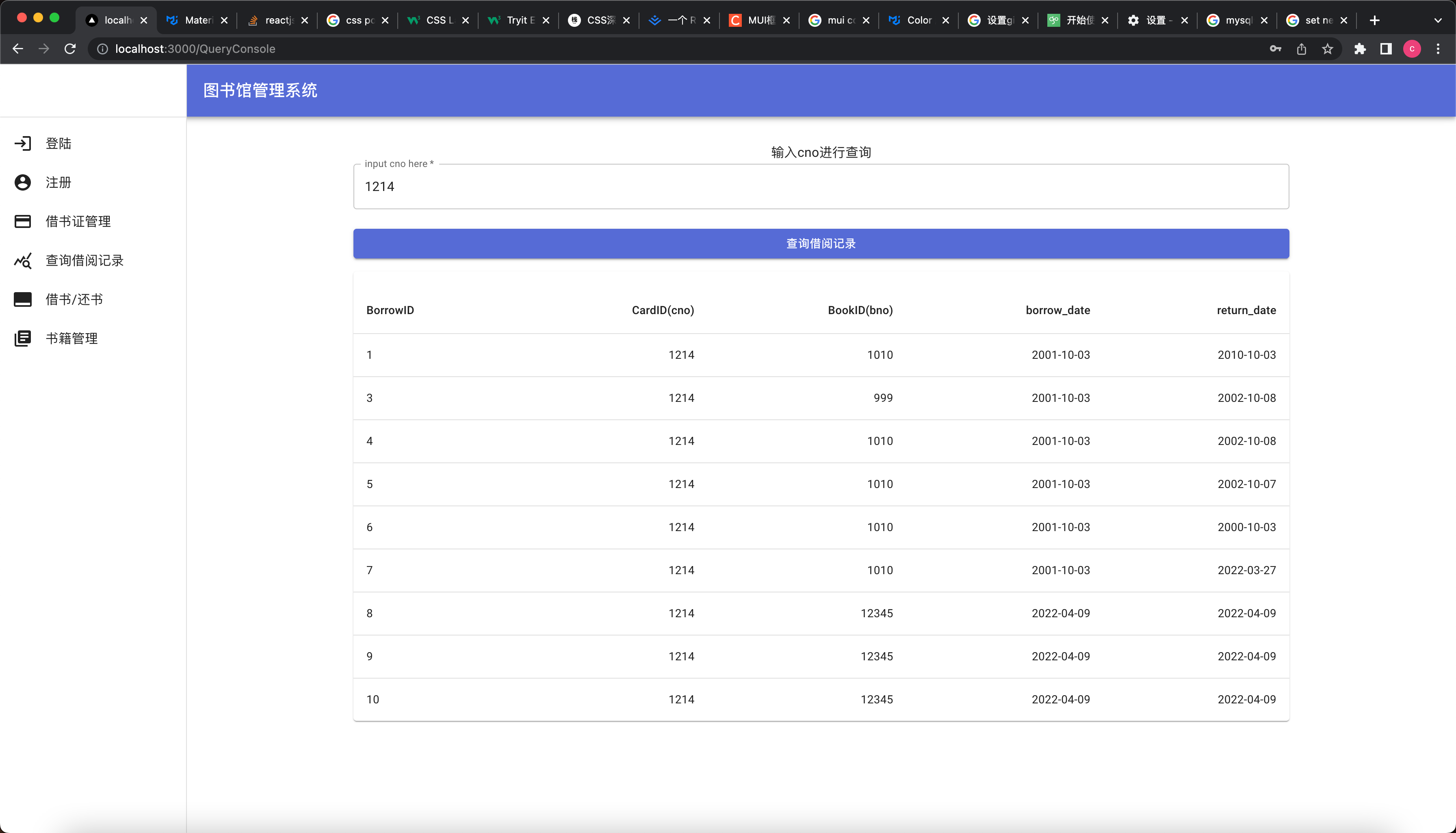


区间查询，查询2001~2021年之间的书籍

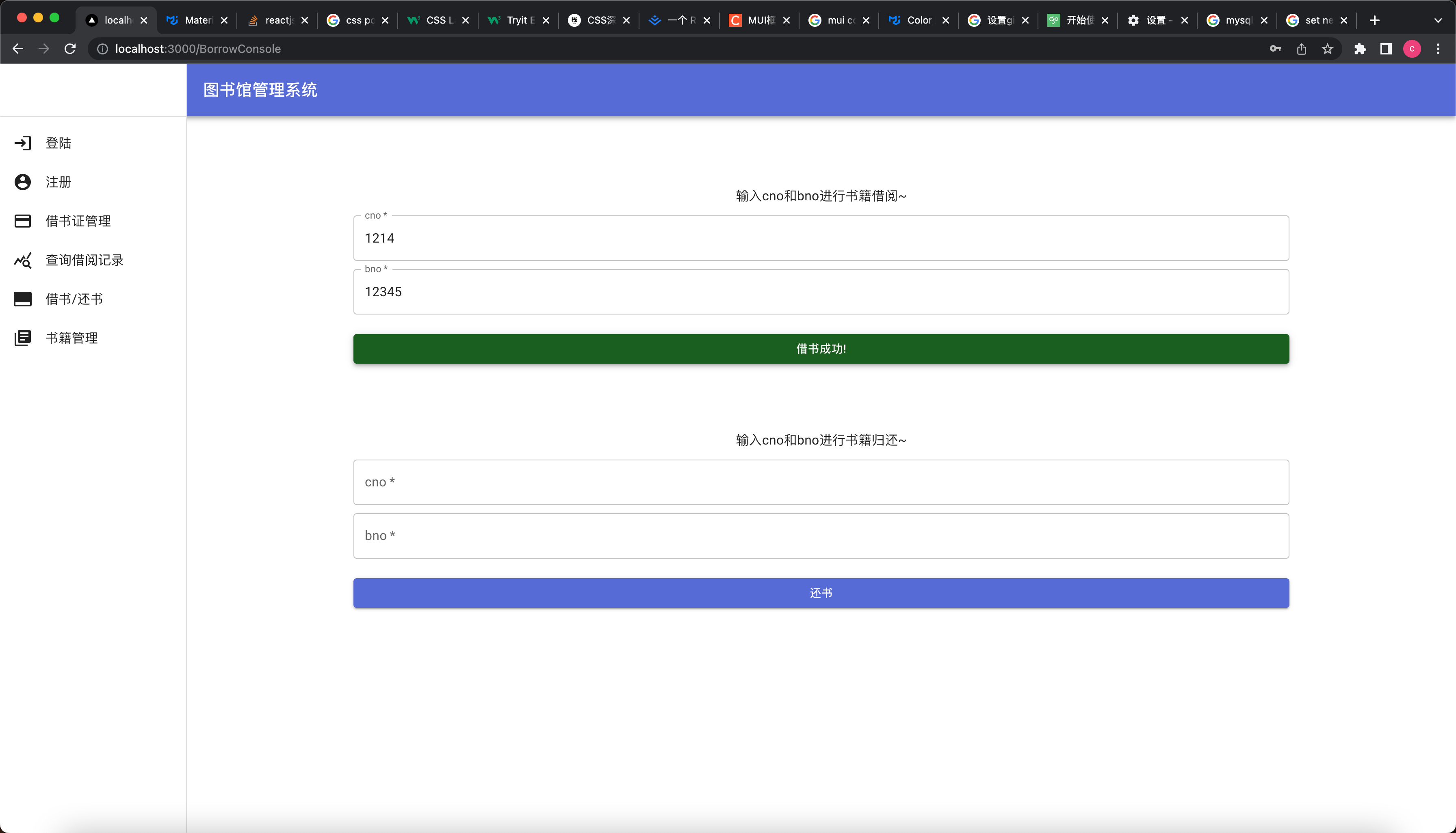


**5.4 图书借阅**

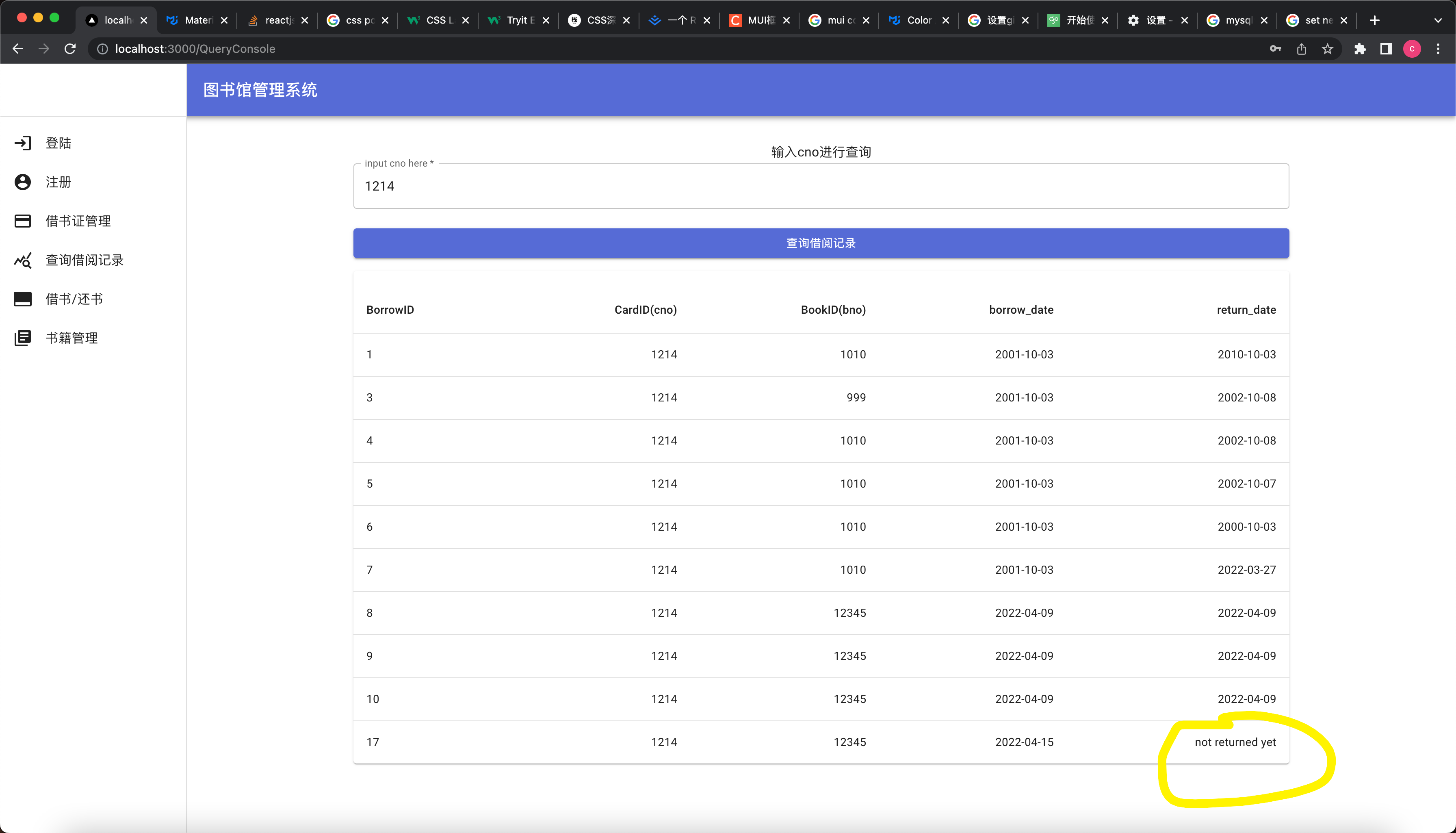
借阅前



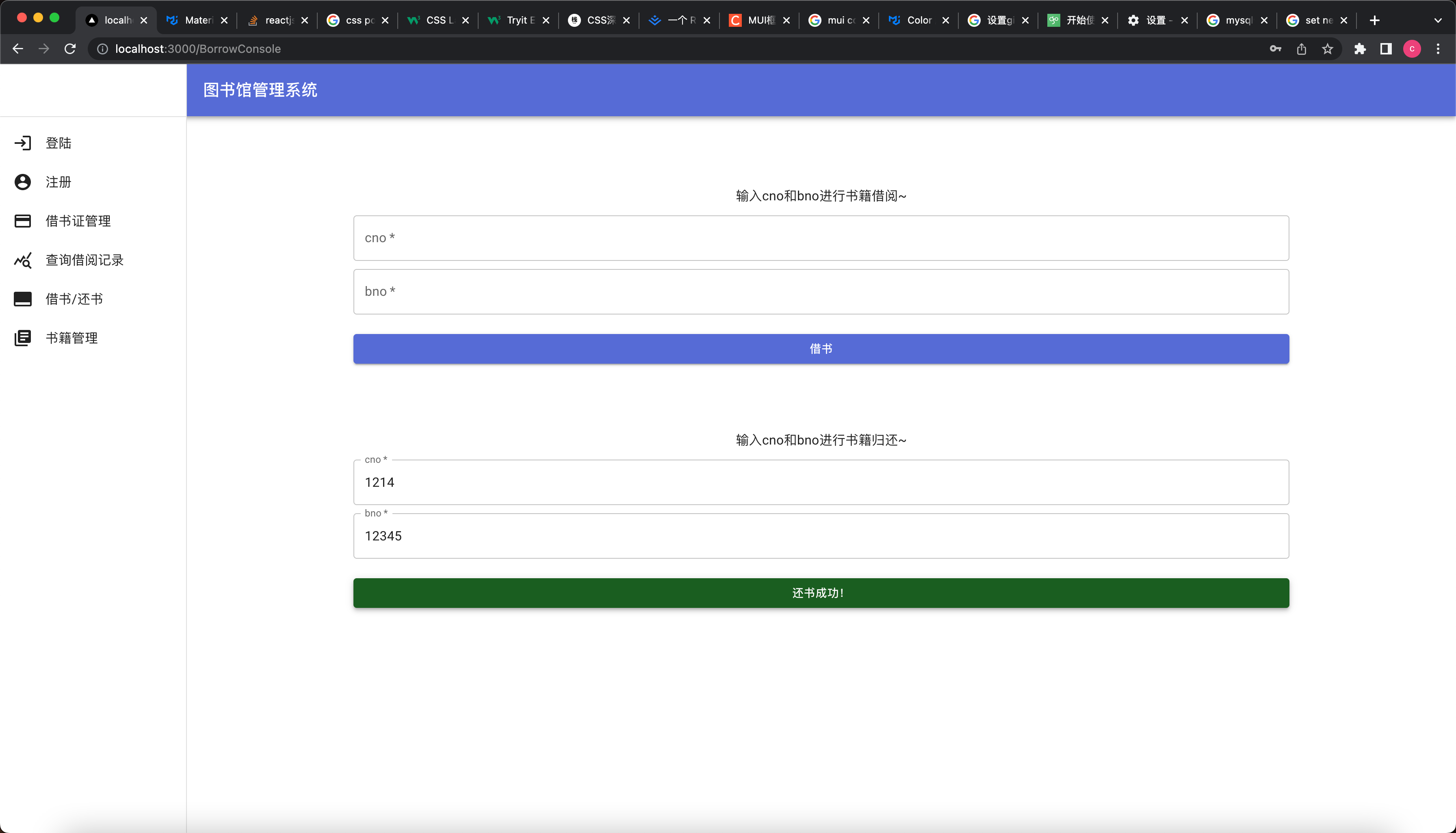
借阅新书籍



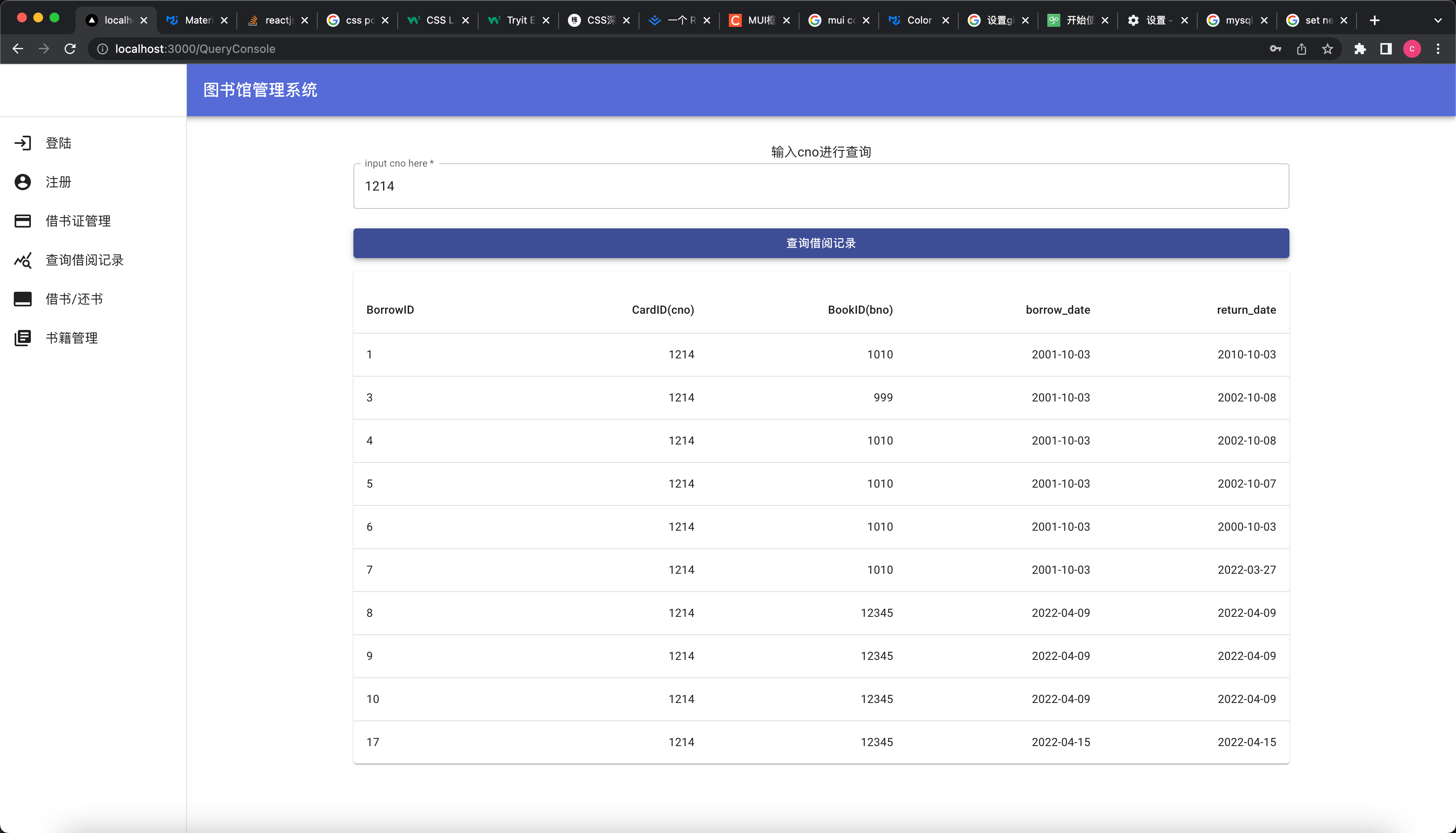
借阅之后



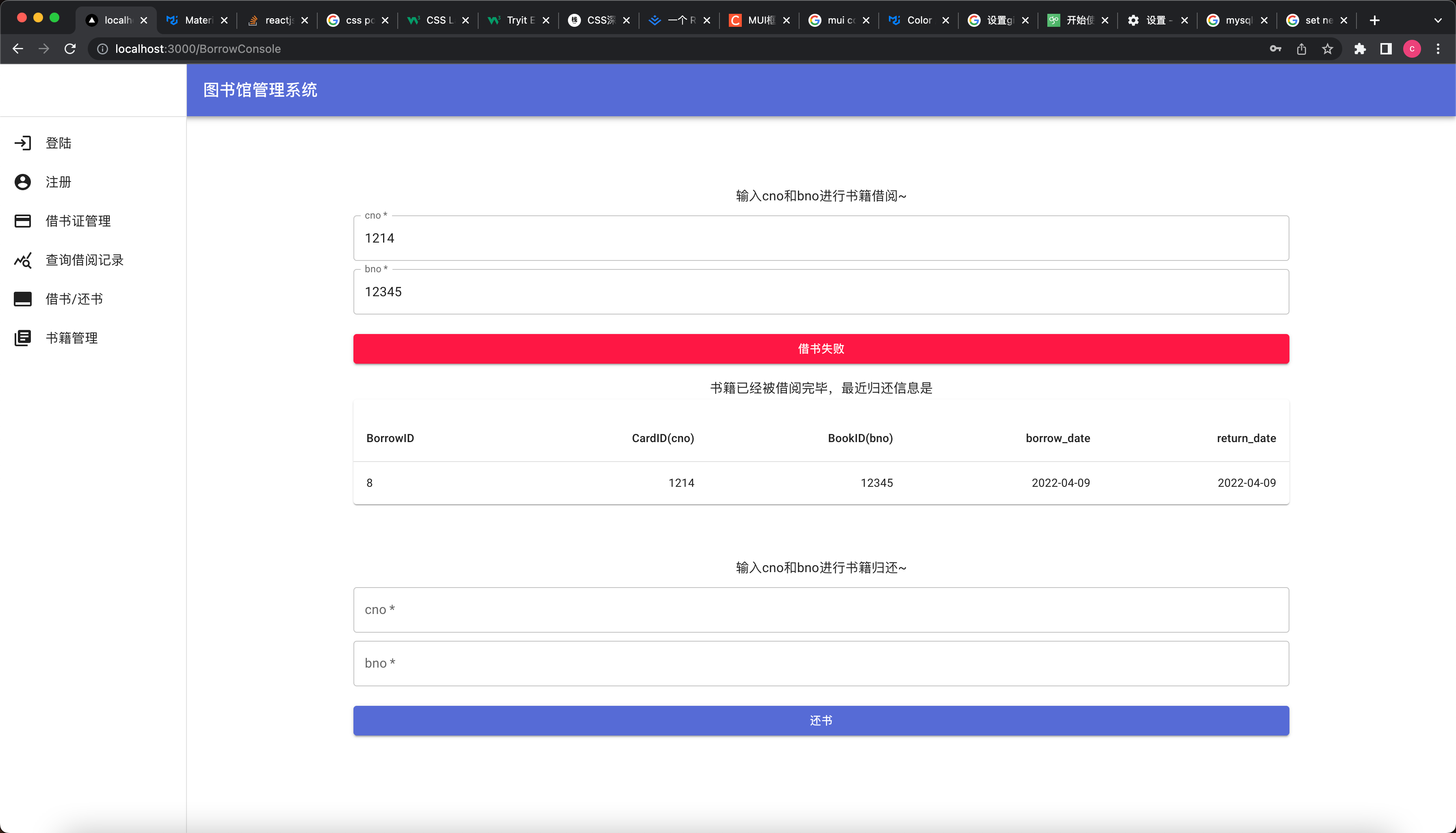
归还书籍



归还后

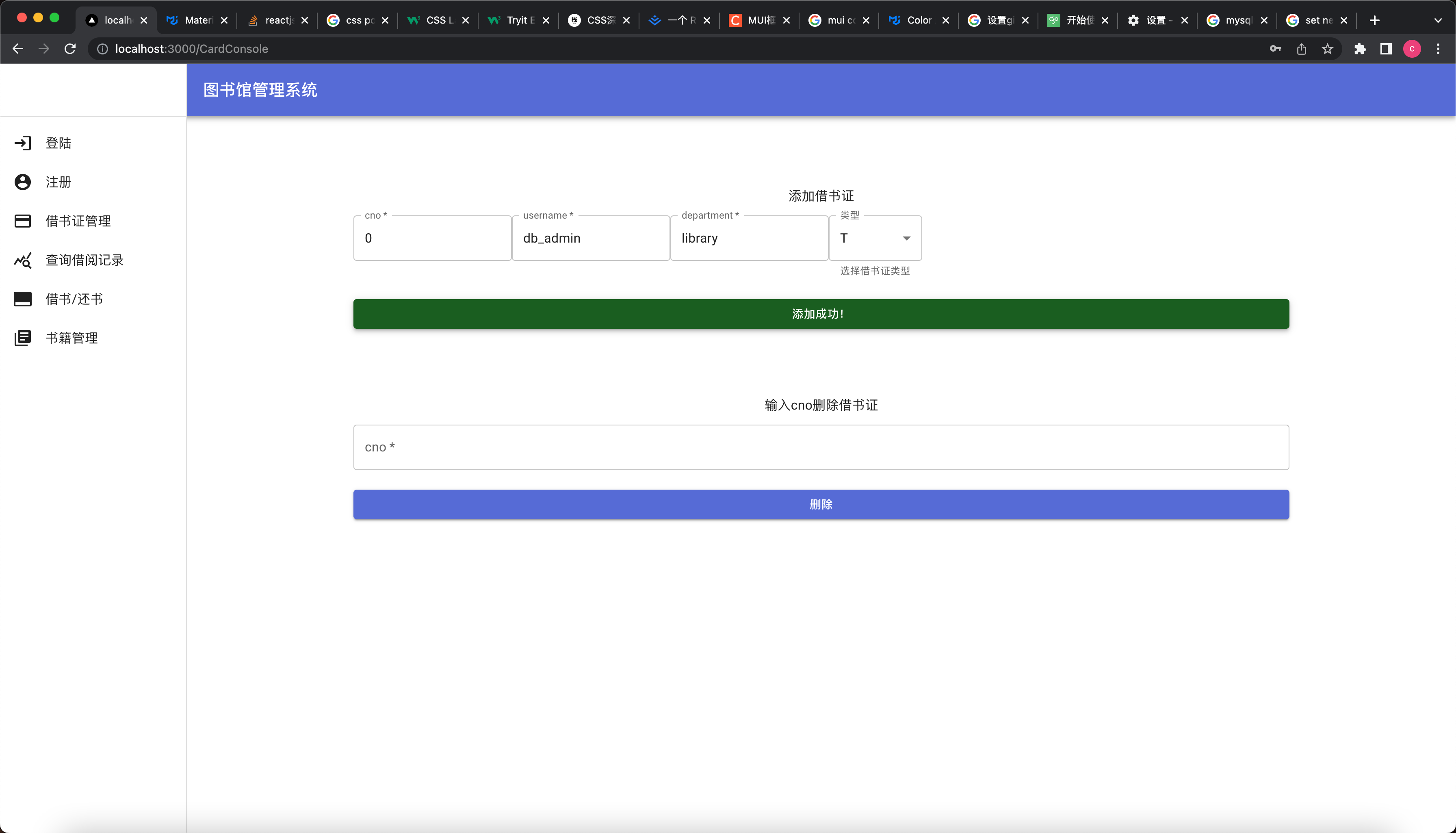


如果书籍借阅完毕，输出最近一次归还信息



**5.5 借书证管理**

**5.5.1 添加借书证**



**5.5.2 删除借书证**



1. **实验心得**

这一次大作业对我而言有点难度。在技术选型方面，本来选择的是C++中的qt框架进行可视化界面的搭建，但是由于qt creator工具下载频繁失败，最终选择使用网页进行gui搭建。后端由于之前学习过golang语言，所以选择gin+gorm框架，前端选择的是Nextjs in typescript + material ui。 网页端开发相对比较简单，安全性方面使用jwt进行加强。

这是我第一次搭建一个这样的简易项目，因此在搭建后端框架的时候遇到了一些问题，比如gorm框架对于如何实现外键约束，给出的文档比较简略，后来通过stackoverflow才终于找到对应的解答。这次项目也促使我更加深入了解orm框架内部机理，锻炼了我阅读文档的能力。

在搭建前端的时候，刚开始是通过React + ReactRouter进行搭建，但是后来在切换路由的时候网页频繁报错，发现是忘记push history的原因，为了避免频繁的书写路由声明，于是切换到了集成化程度更高的nextjs进行开发。

在网络方面，一开始因为跨域问题，前端一直拿不到后端的数据，后来在询问同学之后，在后端加入对应的跨域配置中间件才解决了问题。

在实现整个项目之后，我对于数据库的了解有了一定的加深，未来还需要继续学习，编写出性能更好的代码。