**姓名：刘长鑫 学号：171002609 班级：计算机17-2班**

1. **总体设计**

前端：使用React库，并使用material ui作为UI。

后端：使用Django框架，数据库使用PostgreSQL。

设计原则：前后端分离，前后端只使用JSON格式进行数据传输，前后端交互使用的API遵循RESTful API的设计原则。

1. **前后端交互API**
2. **登录**

Request如下：

POST /api/session

|  |
| --- |
| {      "username":"<用户名>",      "password":"<密码>"  } |

如果登录信息正确，服务器的Response的Body为：

|  |
| --- |
| {      "success": true  } |

否则为：

|  |
| --- |
| {      "success": false  } |

1. **分页显示所有图书**

Request如下：

GET /api/users?per\_page=<page size>&page=<current page number>

服务器Response的Body如下：

|  |
| --- |
| {      "page": <current page number>,      "per\_page": <page size>,      "total": <total item number>,      "total\_pages": <total page number>,      "data": [          {              "id": <book id>,              "name": "<book name>",              "price": <book price>          }      ]  } |

1. **添加图书**

Request如下：

POST /api/book/

|  |
| --- |
| {      "name": "<book name>",      "price": <book price>  } |

服务器Response的Body如下：

|  |
| --- |
| {      "id": <book id>,      "name": "<book name>",      "price": <book price>  } |

1. **更新图书**

Request如下：

PUT /api/book/<book id>/

|  |
| --- |
| {      "name": "<book name>",      "price": <book price>  } |

服务器Response的Body如下：

|  |
| --- |
| {      "id": <book id>,      "name": "<book name>",      "price": <book price>  } |

1. **删除图书**

Request如下：

PUT /api/book/<book id>/

如果删除成功，服务器Response的Body为空。

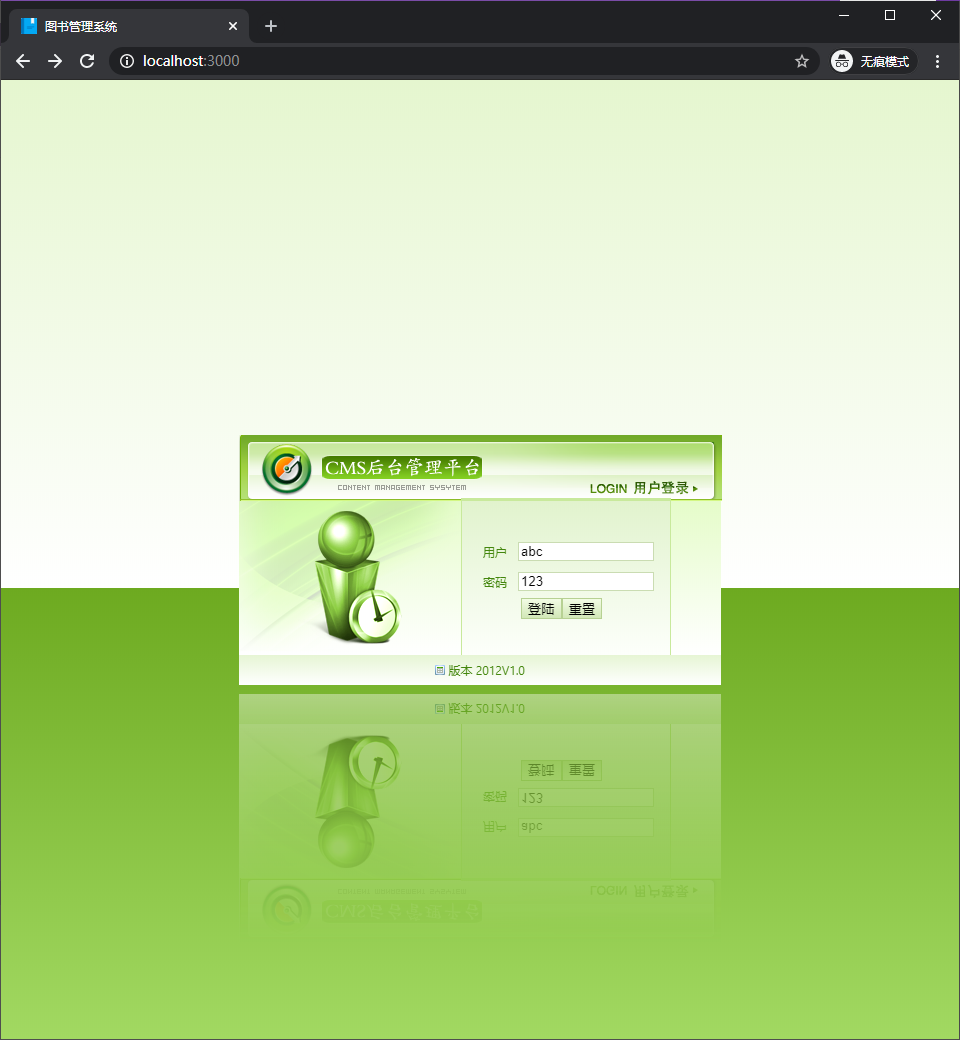
1. **条件查询**

GET /api/search?id=<book id>&name=<book name>&price\_lower\_bound=<minimum price>&price\_upper\_bound=<maximum price>

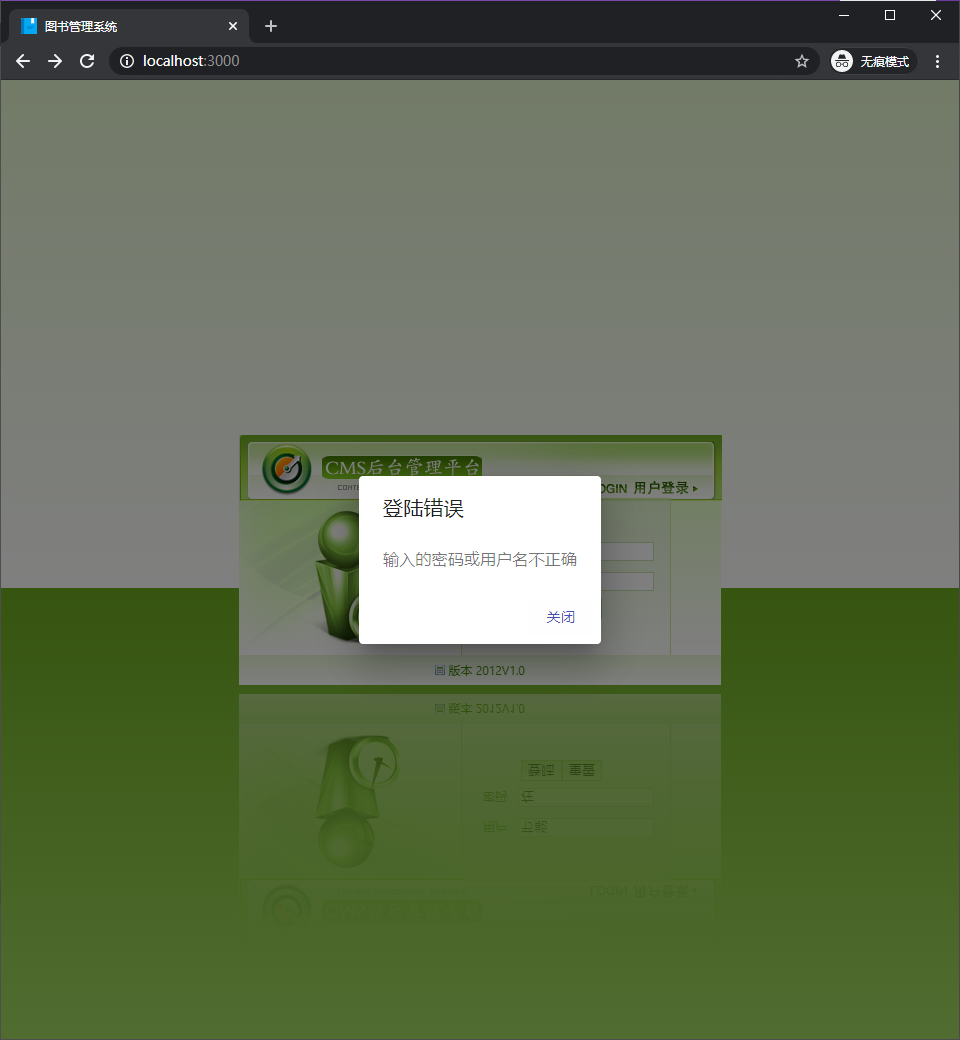
服务器Response的Body如下：

|  |
| --- |
| [      {          "id": <book id>,       "name": "<book name>",       "price": <book price>      }  ] |

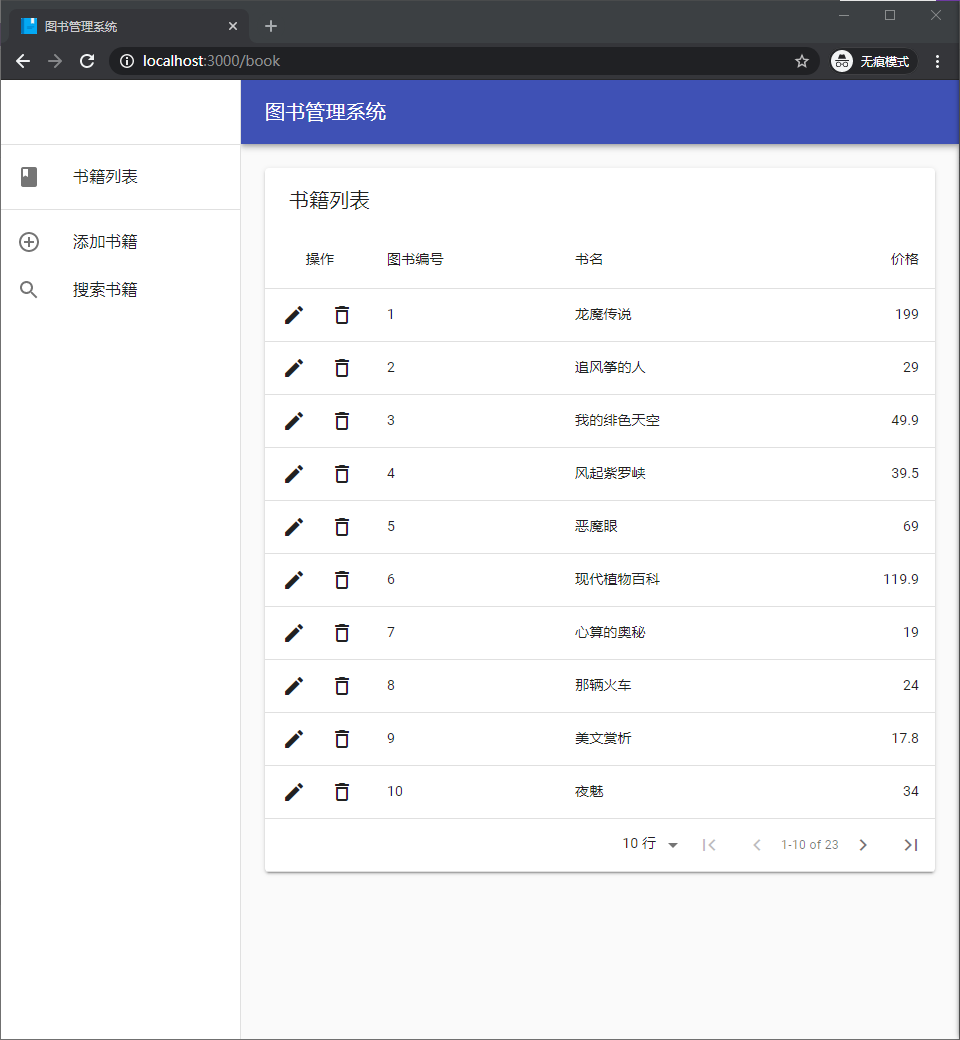
1. **用户指南**
2. **“登录”页面**

****

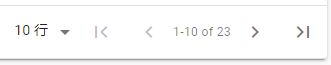
点击“登陆”后，如果用户名与密码正确进入“书籍列表”页面；否则弹出以下提示：



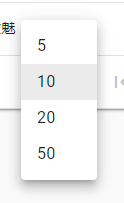
1. **“书籍列表”页面**



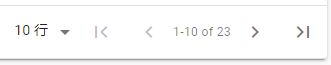
1. 在该页面，可以使用分页列表查看书籍：



可以选择一页显示的项目数量：



默认一页显示10条，可以将值选为5、10、20或50条。



可以在选择按 进入第一页，按进入上一页，进入下一页，到最后一页。

1. 可以对任一书籍进行修改操作：



点击按钮，可对书籍进行修改：

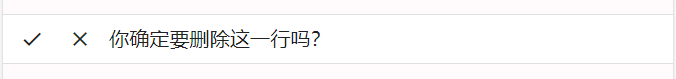


确认修改，点击，否则点击

1. 可以对任一书籍进行删除操作：

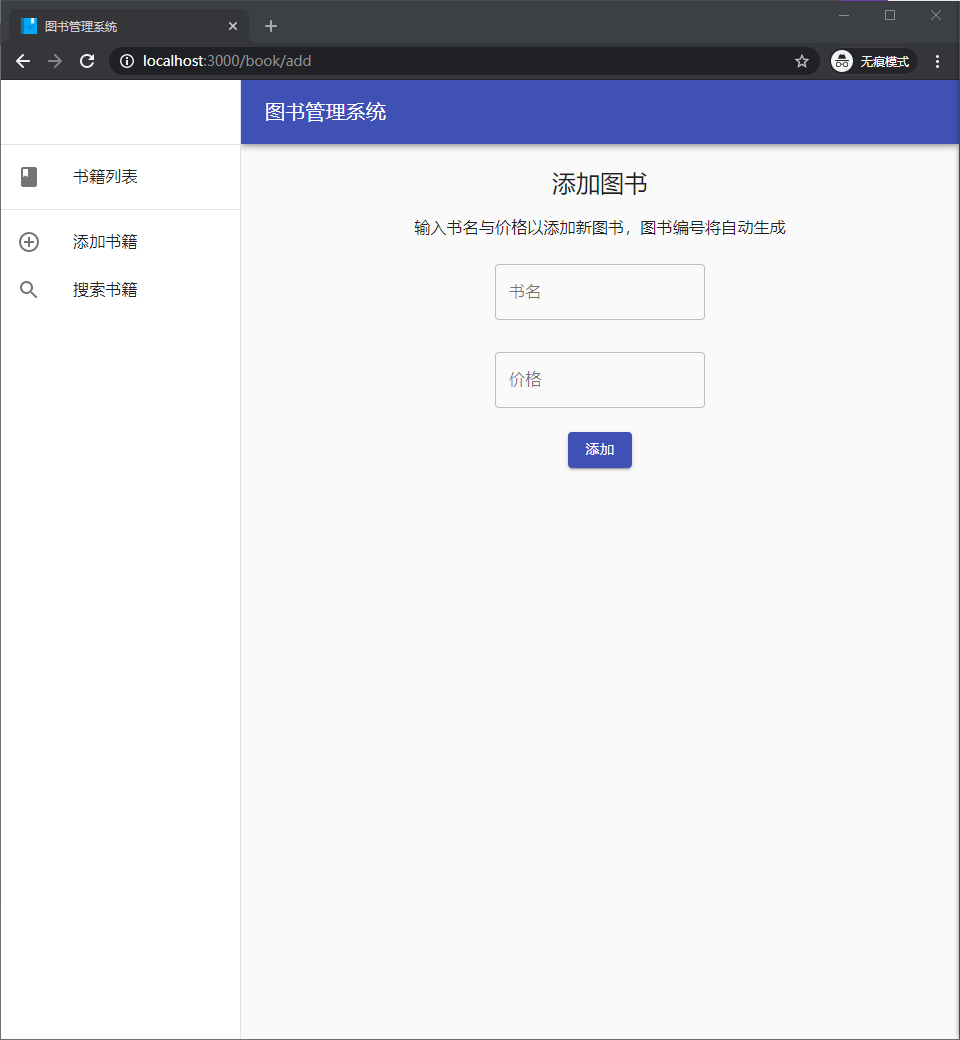


点击按钮，可对书籍进行修改：



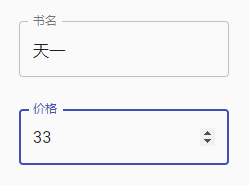
确认修改，点击，否则点击

1. **“添加书籍”页面**



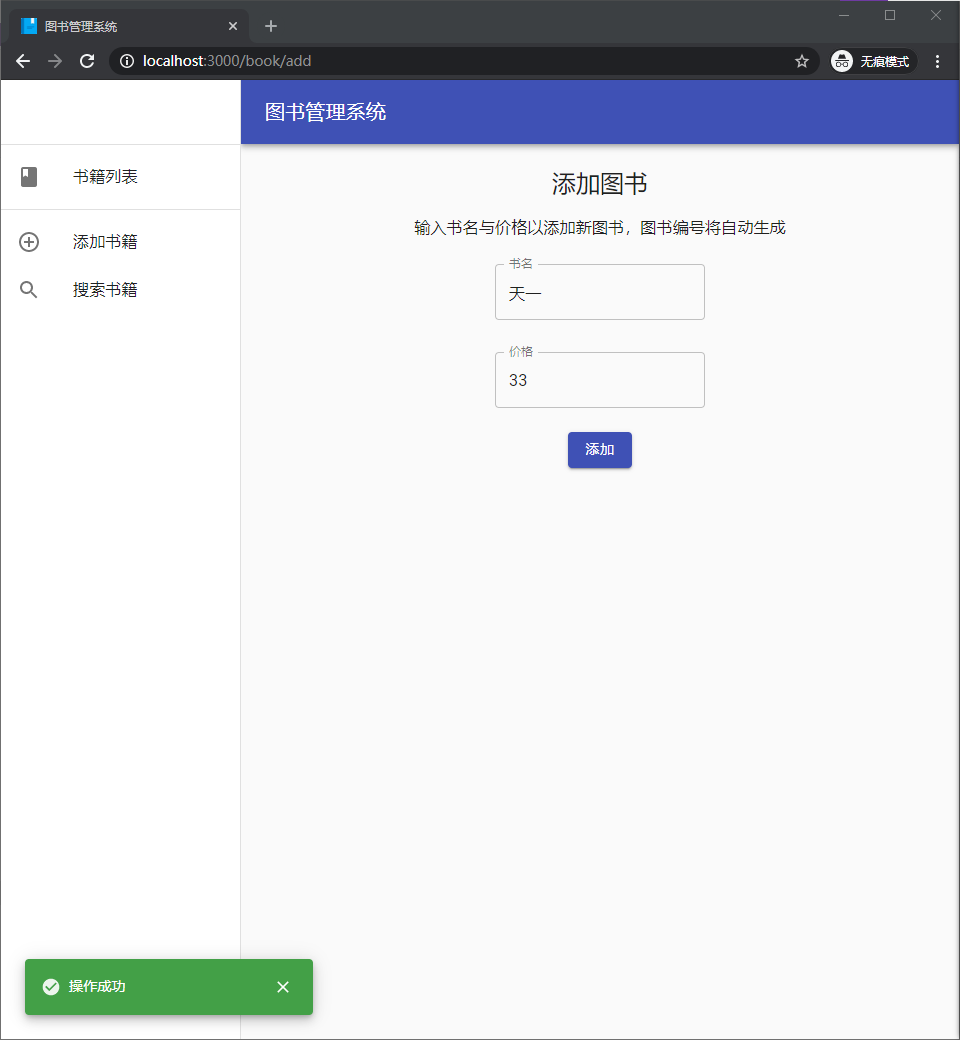
在其中输入书名和价格，点击“添加”按钮后书籍将被添加。

当输入框被激活时，效果如下：

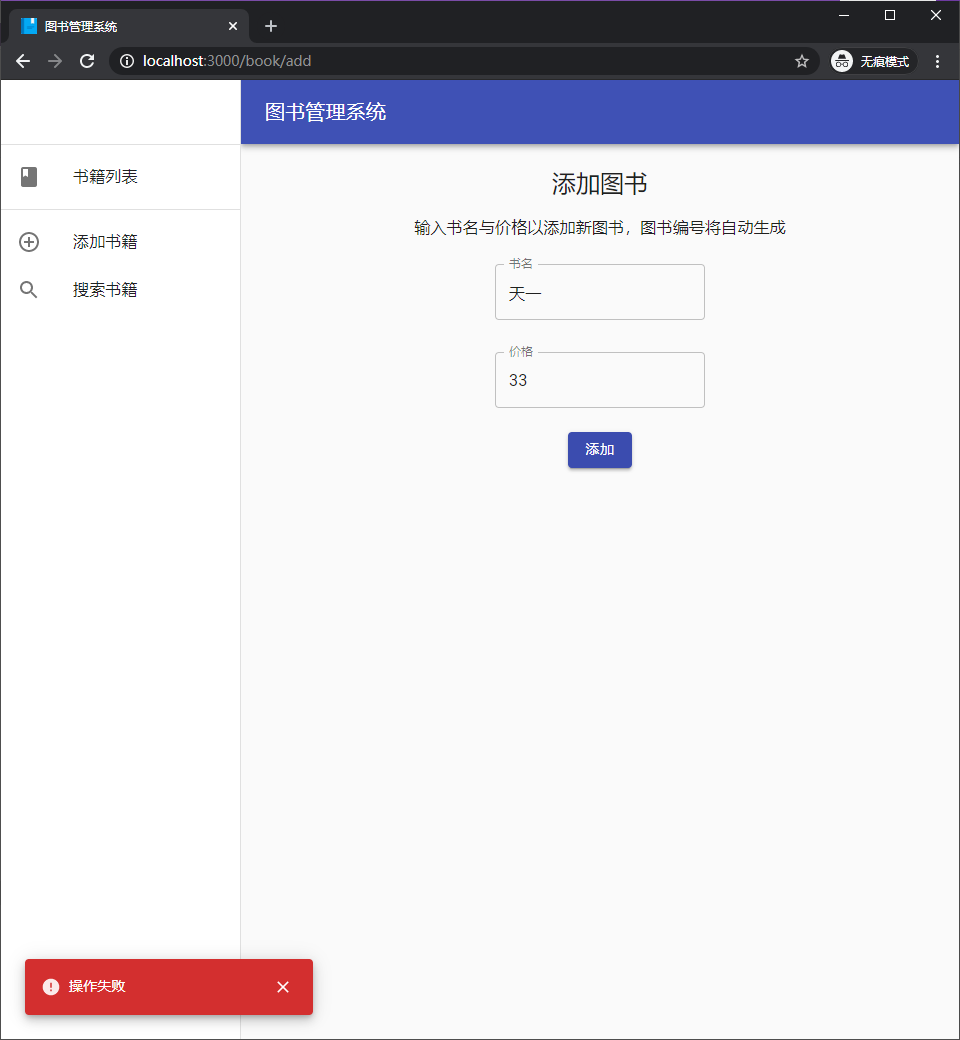
当添加成功，在页面左下角显示如下“操作成功”，如图：



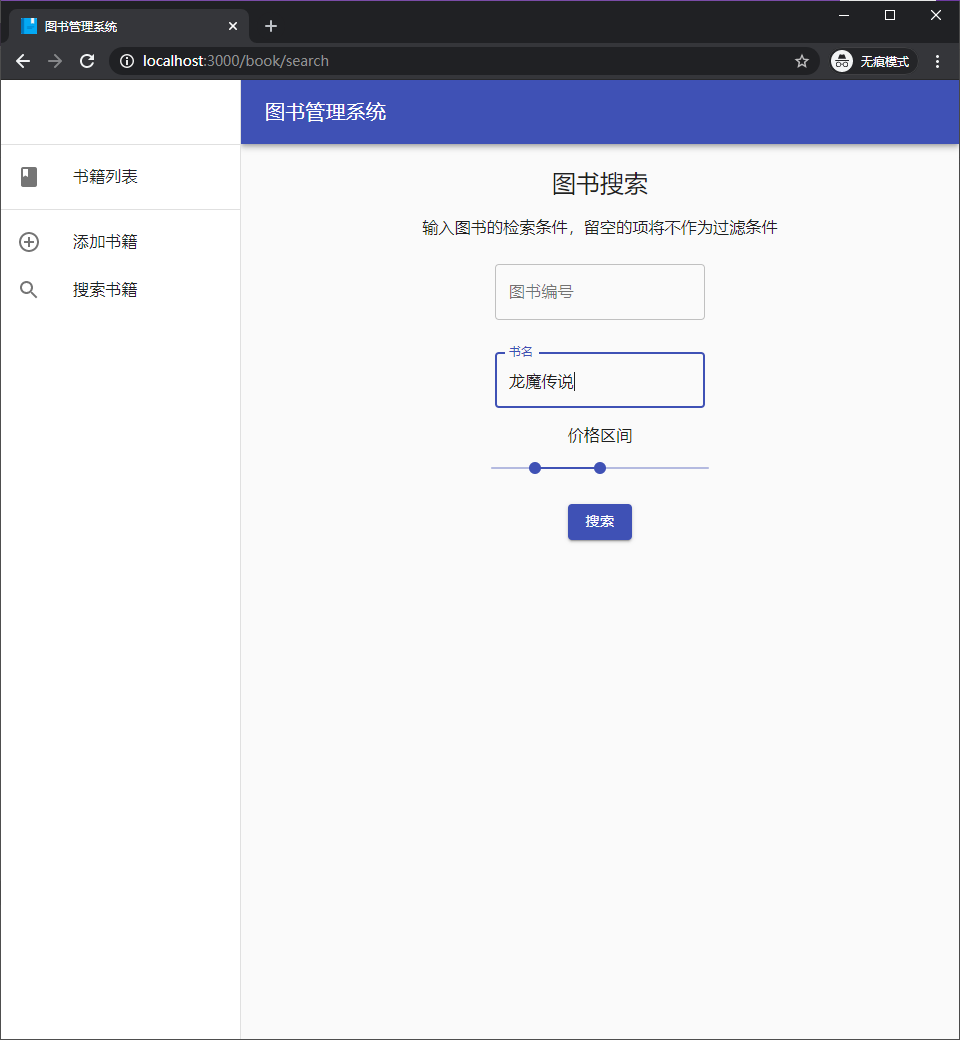


否则显示“操作失败”（如无法连接到后端服务器的情况），如下图：

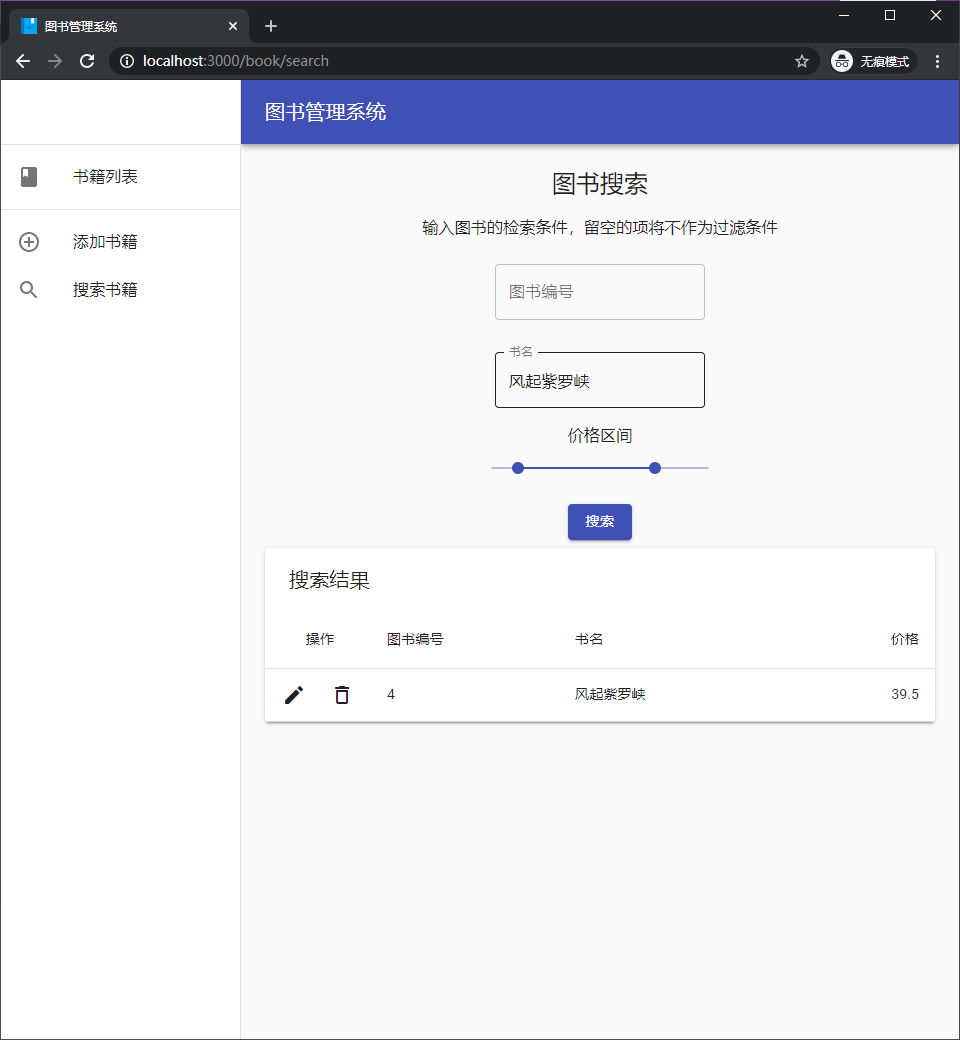




1. **“图书搜索”页面**

****

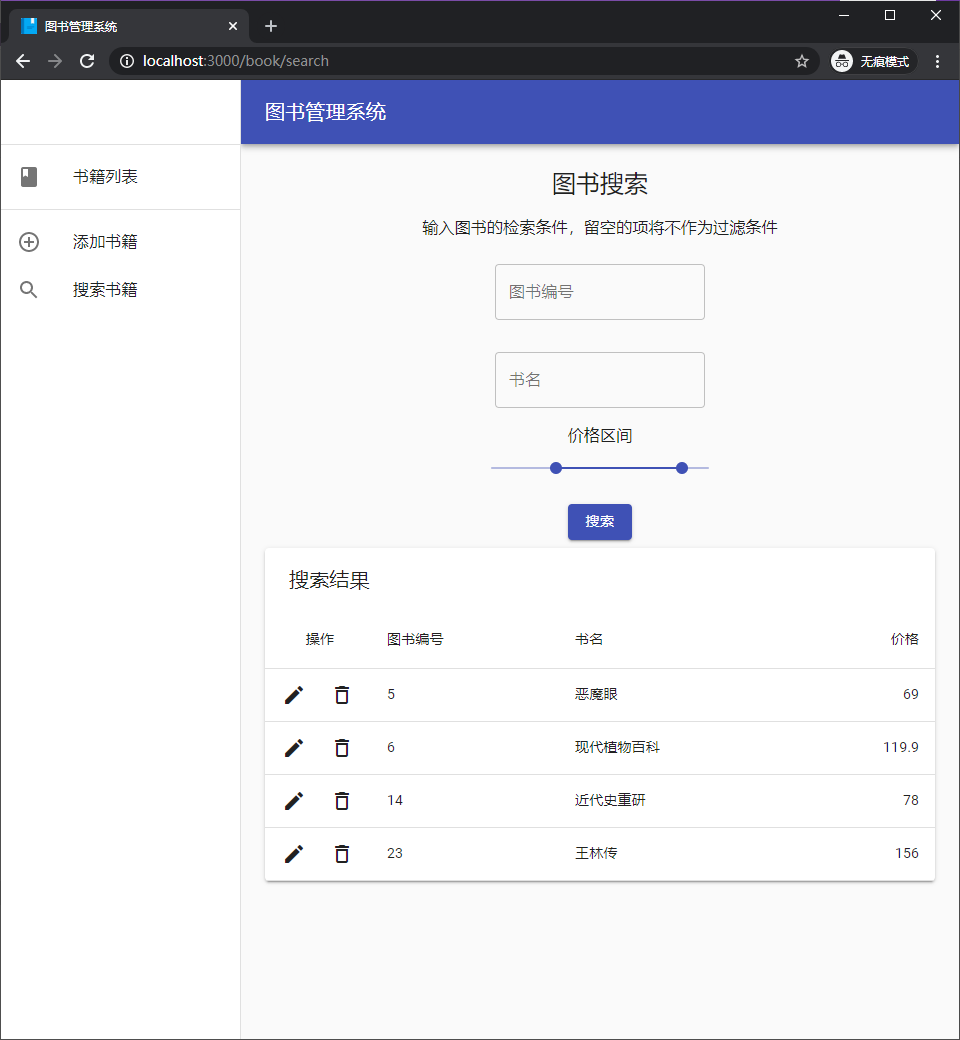
当搜索结果返回后，页面如下：



使用方法：

输入图书的检索条件，留空的项将不作为过滤条件。

如：不限定图书编号与书名，搜索价格在60-175之间的图书，则将“图书编号”与“书名”留空，将价格区间的滑动条选为60-175，如下图：



1. **总结**

该图书管理系统拥有登录与对图书的增删改查功能：

* 增：在“添加书籍”页面完成书籍的添加
* 删：在“书籍列表”页面或在“搜索书籍”页面完成书籍的删除
* 改：在“书籍列表”页面或在“搜索书籍”页面完成书籍的修改
* 查：在“书籍列表”页面通过分页列表显示所有书籍信息，在“搜索书籍”页面完成条件搜索（可以搜索图书编号、书名以及限定价格区间）

该图书管理系统使用React作为前端框架，并使用了material ui。使用Django作为后端框架，并使用PostgreSQL数据库来进行数据持久化。同时前后端分离，只使用JSON格式进行数据传输，前后端交互使用的API遵循RESTful API的设计原则。