计算机与软件学院

**《Python程序开发课程设计》报告**

基于Django+Sql的图书管理系统

专 业： 软件工程

班 级：

学 号：

姓 名：

指导老师：

报告日期： 2022 年 06 月 08 日

目录

[1 项目简介 2](#_Toc105787424)

[1.1 前言 2](#_Toc105787425)

[1.2 开发环境 2](#_Toc105787426)

[1.3 开发与部署 2](#_Toc105787427)

[2 需求分析 3](#_Toc105787428)

[2.1 需求分析 3](#_Toc105787429)

[2.2 功能模块 3](#_Toc105787430)

[3 系统设计 4](#_Toc105787431)

[3.1 系统整体架构设计 4](#_Toc105787432)

[3.2 系统核心模块过程设计 4](#_Toc105787433)

[3.2.1 基本设置 4](#_Toc105787434)

[3.2.2 用户登陆模块 5](#_Toc105787435)

[3.2.3 图书添加模块 6](#_Toc105787436)

[3.2.4 图书删除模块 8](#_Toc105787437)

[3.2.5 图书查询模块 9](#_Toc105787438)

[4 编码与实现 12](#_Toc105787439)

[4.1 前端 12](#_Toc105787440)

[4.2 后端 16](#_Toc105787441)

[5 总结与展望 18](#_Toc105787442)

[5.1 总结 18](#_Toc105787443)

[5.2 展望 18](#_Toc105787444)

# 项目简介

## 前言

进入21世纪以来，信息技术从根本上推动了图书馆的飞速发展，计算机和计算机管理系统已成为图书馆进行图书管理的主要设备和系统。虽然目前很多大型的图书馆已经有一整套比较完善的管理系统，但是在一些中小型的图书馆中，大部分工作仍需手工完成，工作起来效率比较低，不便于动态、及时地调整图书结构。为了更好地适应当前图书馆的管理需求，解决手工管理中存在的弊端，越来越多的中小型图书馆正在逐步向计算机信息化管理转变。

## 开发环境

编译器；PyCharm2021.2.3

解释器：conda4.12

Python：3.8.8

Django：3.2.5

Pymysql：1.0.2

MySQL：8.0

Navicat：15

## 开发与部署

本地编程开发Django图书管理系统，用python manage.py makemigrations与python manage.py migrate 进行数据库迁移，用Django自带server启动程序，python manage.py runserver。

# 需求分析

## 需求分析

登陆页面：进行用户登陆验证。

主页：分页显示数据库中的图书信息，用户可以对图书进行添加、修改、删除、查询。

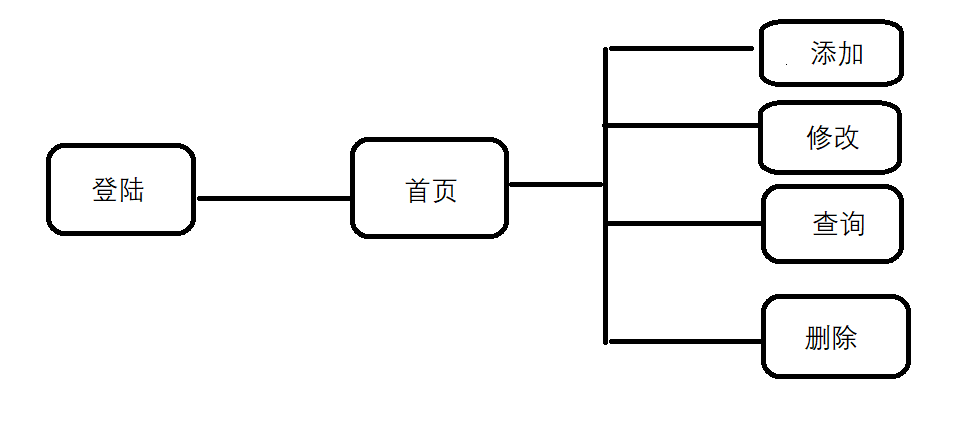
查看图书：用户可以在此界面上查看数据库中所有的图书。

修改图书：用户可以对图书修改，如：书名、作者、价格、出版社、ISBN、能否在线阅读、图书编号。

删除图书：用户可以对图书进行删除。

查询图书：用户可以根据作者、书名、是否能在线阅读对图书进行进行查询

## 功能模块



# 系统设计

## 系统整体架构设计

新建项目并新建app，在app的同级目录创建一个static文件夹，这里面主要是一些css和js的包，在app下新建templates，用于存放html文件。在app目录下新建urls.py 文件，并在其中设置路由，在项目的总urls.py中引入app模块。

## 系统核心模块过程设计

### 基本设置

在setting中对项目进行基本设置，添加静态加载代码：

STATIC\_URL = '/static/'  
STATICFILES\_DIRS = *[* os.path.join*(*BASE\_DIR, 'static'*)  
]*

添加数据库连接代码：

DATABASES = *{* 'default': *{* 'ENGINE': 'django.db.backends.mysql',  
 'NAME': 'booklibrary', # 数据库名  
 'USER': 'root', # 数据库账号

'PASSWORD': 'aktgw33', # 数据库密码

'HOST': 'localhost', # 数据库位置

'PORT': '', # 数据库端口（可空缺）

*}*

### 用户登陆模块

前端使用from表单通过post请求去获取用户输入的账号密码，同时用csrf\_token去验证并保护提交数据：

*<*h2*>*Sign In*</*h2*>  
<*form action="#" method="post"*>* {% csrf\_token %}  
 *<*input type="text" name="UserName" placeholder="Username" required=""*>  
 <*input type="password" name="PassWord" placeholder="Password" required=""*>  
 <*ul class="tick w3layouts agileits"*>  
 <*li*>  
 <*input type="checkbox" id="brand1" value=""*>  
 <*label for="brand1"*><*span*></*span*>*Remember me*</*label*>  
 </*li*>  
 </*ul*>  
 <*div class="send-button w3layouts agileits"*>  
 <*input type="submit" value="Sign In"*>  
 </*div*>  
</*form*>*

后端使用get方法得到前端页面中提交的数据，并从数据库中查找是否有改用户存在，如果存在进行密码验证，验证通过进行页面重定向进入图书管理系统，否则返回登陆页面。

def login*(*request*)*:  
 if request.method == 'POST':  
 user\_name = request.POST.get*(*'UserName'*)* user\_passwd = request.POST.get*(*'PassWord'*)* users\_obj = models.User.objects.get*(*UserName=user\_name*)* if not users\_obj is None:  
 if users\_obj.PassWord == user\_passwd:  
 return redirect*(*'/app/booklist'*)* return render*(*request, 'index.html'*)* else:  
 return render*(*request, 'index.html'*)*

### 图书添加模块

前端使用from表单通过post请求去获取需要添加的图书信息，同时用csrf\_token去验证并保护提交数据：

*<*form action="" method="post"*>* {% csrf\_token %}  
 <!-- 加密系统 -->  
 *<*p*>* id *<*input class="btn btn-primary-outline radius" type="text" name="id"*>  
 </*p*>  
 <*p*>* 书名 *<*input class="btn btn-primary-outline radius" type="text" name="BookName"*>  
 </*p*>  
 <*p*>* 作者 *<*input class="btn btn-primary-outline radius" type="text" name="Author"*>  
 </*p*>  
 <*p*>* 价格 *<*input class="btn btn-primary-outline radius" type="text" name="Price"*>  
 </*p*>  
 <*p*>* 出版社 *<*input class="btn btn-primary-outline radius" type="text" name="Publisher"*>  
 </*p*>  
 <*p*>* 版号 *<*input class="btn btn-primary-outline radius" type="text" name="ISBN"*>  
 </*p*>  
 <*p*>* 是否能在线阅读 *<*input class="btn btn-primary-outline radius" type="text" name="IsReadOnline"*>  
 </*p*>  
 <*p*>* 书编号 *<*input class="btn btn-primary-outline radius" type="text" name="BookNum"*>  
 </*p*>  
 <*p*>  
 <*button class="btn btn-warning-outline radius" type="submit" class="btn"*>*提交*</*button*>  
 </*p*>  
 </*form*>*

后端使用get方法得到前端页面中提交的数据，并保存到数据库中进行页面重定向：

def addbook*(*request*)*:  
 if request.method == 'POST':  
 # 获取数据  
 book\_id = request.POST.get*(*'id'*)* book\_name = request.POST.get*(*'BookName'*)* book\_author = request.POST.get*(*'Author'*)* book\_price = request.POST.get*(*'Price'*)* book\_publisher = request.POST.get*(*'Publisher'*)* book\_isbn = request.POST.get*(*'ISBN'*)* book\_isreadonline = request.POST.get*(*'IsReadOnline'*)* book\_booknum = request.POST.get*(*'BookNum'*)* # 保存到数据库中  
 models.Book.objects.create*(*id=book\_id, BookName=book\_name, Author=book\_author,  
 Price=book\_price, Publisher=book\_publisher,  
 ISBN=book\_isbn, IsReadOnline=book\_isreadonline,  
 BookNum=book\_booknum*)* return redirect*(*'/app/booklist'*)* return render*(*request, 'addbook.html'*)*

### 图书删除模块

获取需要删除图书的id，在数据库中利用id进行查询并删除：

def del\_book*(*request*)*:  
 # 获取要删除的id  
 id = request.GET.get*(*"id"*)* # 根据id删除出局库中的记录  
 models.Book.objects.filter*(*id=id*)*.delete*()* return redirect*(*"/app/booklist"*)*

### 图书查询模块

判断查询的是书名还是作者还是在线是否能在线阅读，然后查询数据库，将查询到的信息返回到前端页面：

def find*(*request*)*:  
 if request.method == 'POST':  
 # 获取数据  
 book\_name = request.POST.get*(*'BookName'*)* book\_author = request.POST.get*(*'Author'*)* book\_isreadonline = request.POST.get*(*'IsReadOnline'*)* if book\_name != '':  
 book\_obj = models.Book.objects.filter*(*BookName=book\_name*)* return render*(*request, 'find.html', *{*"book\_obj\_list": book\_obj*})* # 获取数据库中数据  
 if book\_author != '':  
 book\_obj = models.Book.objects.filter*(*Author=book\_author*)* return render*(*request, 'find.html', *{*"book\_obj\_list": book\_obj*})* if book\_isreadonline != '':  
 book\_obj = models.Book.objects.filter*(*IsReadOnline=book\_isreadonline*)* return render*(*request, 'find.html', *{*"book\_obj\_list": book\_obj*})* return render*(*request, 'find.html'*)*

前端通过for循环循环读取从后端得到的字典，并将内容在页面中展示出来：

*<*form action="" method="post"*>* {% csrf\_token %}  
 <!-- 加密系统 -->  
  
 *<*p*>* 书名查找 *<*input class="btn btn-primary-outline radius" type="text" name="BookName" *>  
 <*button class="btn btn-primary-outline radius" type="submit" class="btn"*>*查找*</*button*>* 作者查找 *<*input class="btn btn-primary-outline radius" type="text" name="Author" *>  
 <*button class="btn btn-primary-outline radius" type="submit" class="btn"*>*查找*</*button*>* 是否能在线阅读 *<*input class="btn btn-primary-outline radius" type="text" name="IsReadOnline" *>  
 <*button class="btn btn-primary-outline radius" type="submit" class="btn"*>*查找*</*button*>  
 </*p*>  
  
</*form*>  
 <*div class="panel-body"*>  
 <*table class="table table-border table-bordered table-striped mt-20" id="table"*>  
 <*thead*>  
 <*tr*>  
 <*th class="col1" style="text-align:center"*>*id*</*th*>  
 <*th class="col1" style="text-align:center"*>*书名*</*th*>  
 <*th class="col2" style="text-align:center"*>*作者*</*th*>  
 <*th class="col3" style="text-align:center"*>*价格*</*th*>  
 <*th class="col1" style="text-align:center"*>*出版社*</*th*>  
 <*th class="col2" style="text-align:center"*>*版号*</*th*>  
 <*th class="col2" style="text-align:center"*>*在线阅读*</*th*>  
 <*th class="col3" style="text-align:center"*>*编号*</*th*>  
 <*th class="col3" style="text-align:center"*>*操作*</*th*>  
 </*tr*>* {% for book\_obj in book\_obj\_list %}  
 *<*tr*>  
 <*th class="col1" style="text-align:center"*>*{{book\_obj.id}}*</*th*>  
 <*th class="col1" style="text-align:center"*>*{{book\_obj.BookName}}*</*th*>  
 <*th class="col2" style="text-align:center"*>*{{book\_obj.Author}}*</*th*>  
 <*th class="col3" style="text-align:center"*>*{{book\_obj.Price}}*</*th*>  
 <*th class="col1" style="text-align:center"*>*{{book\_obj.Publisher}}*</*th*>  
 <*th class="col2" style="text-align:center"*>*{{book\_obj.ISBN}}*</*th*>  
 <*th class="col1" style="text-align:center"*>*{{book\_obj.IsReadOnline}}*</*th*>  
 <*th class="col3" style="text-align:center"*>*{{book\_obj.BookNum}}*</*th*>  
 <*th class="col3" style="text-align:center"*><*a class="btn btn-primary-outline radius"  
 href="/app/edit\_book/?id={{book\_obj.id}}"*>*修改*</*a*>  
 <*a class="btn btn-warning-outline radius"  
 href="/app/del\_book/?id={{book\_obj.id}}"*>*删除*</*a*></*th*>  
 </*tr*>* {% endfor %}  
 *</*thead*>  
 <*tbody id="tablebody"*>  
 </*tbody*>  
 </*table*>  
 </*div*>* <!--end表格-->

# 编码与实现

## 前端

页面主体部分：

{% load static %}  
*<!DOCTYPE* html>  
*<*html lang="en"*>  
  
<*head*>  
 <*meta http-equiv="Content-Type" content="text/html; charset=UTF-8"*>  
  
 <*meta http-equiv="X-UA-Compatible" content="IE=edge"*>  
 <*meta name="viewport" content="width=device-width, initial-scale=1"*>  
 <*meta name="description" content=""*>  
 <*meta name="author" content=""*>  
 <*link rel="icon" href="https://v3.bootcss.com/favicon.ico"*>  
  
 <*title*>*{% block title %}{% endblock title %}*</*title*>  
 <*link href="{% static 'bootstrap/css/bootstrap.min.css' %}" rel="stylesheet"*>  
 <*link href="{% static 'dashboard.css' %}" rel="stylesheet"*>  
 <*link rel="stylesheet" href="{% static 'fontawesome/css/font-awesome.min.css' %}"*>  
 <*link rel="stylesheet" type="text/css" href="{% static 'css/H-ui.css'%}" *>  
</*head*>  
<*body style="background-image: url({%static 'images/bg-main.png' %})" *>*<!--分页名称-->  
<!--导航栏-->  
{% include "nav.html" %}  
*<*div class="container-fluid"*>  
 <*div class="row"*>  
 <*div class="col-md-3"*>  
 <*ul class="nav nav-sidebar"*>* {% block nav %}  
 {% endblock %}  
 *</*ul*>  
 </*div*>*<!--主程序-->  
  
 *<*div class="col-md-9" style="justify-content:flex-end;"*>  
 <*h1 class="page-header" style="text-align:center"*>*图书管理系统*</*h1*>  
 <*div class="panel panel-primary" *>  
 <*div class="panel-header" style="text-align:center"*>*{% block name %}{% endblock name %}*</*div*>  
 <*div class="panel-body" style="background-color:#eae5d8"*>* {% block main %}  
 {% endblock %}  
 <!-- 页码 -->  
 *<*nav aria-label="Page navigation"*>  
 <*ul class="" style=""*>* {% if book\_obj\_list.has\_previous %}  
 *<*li style="display: inline"*>  
 <*a href="?page={{ book\_obj\_list.previous\_page\_number }}" class="btn btn-danger-outline radius" aria-label="Previous"*>  
 <*span aria-hidden="true"*>*&laquo;*</*span*>  
 </*a*>  
 </*li*>* {% endif %}  
 *<*li style="display: inline"*><*a href="#" class="btn btn-danger-outline radius"*>*Page {{ book\_obj\_list.number }} of {{ book\_obj\_list.paginator.num\_pages }}*</*a*></*li*>  
 <*li style="display: inline"*><*a href="#" class="btn btn-danger-outline radius"*>*2*</*a*></*li*>  
 <*li style="display: inline"*><*a href="#" class="btn btn-danger-outline radius"*>*3*</*a*></*li*>  
 <*li style="display: inline"*><*a href="#" class="btn btn-danger-outline radius"*>*4*</*a*></*li*>  
 <*li style="display: inline"*><*a href="#" class="btn btn-danger-outline radius"*>*5*</*a*></*li*>* {% if book\_obj\_list.has\_next %}  
 *<*li style="display: inline"*>  
 <*a href="?page={{ book\_obj\_list.next\_page\_number }}" class="btn btn-danger-outline radius" aria-label="Next"*>  
 <*span aria-hidden="true"*>*&raquo;*</*span*>  
 </*a*>  
 </*li*>* {% endif %}  
 *</*ul*>  
 </*nav*>  
 </*div*>  
 </*div*>  
 </*div*>  
 </*div*>  
</*div*>  
</*body*>  
</*html*>*

## 后端

图书列表和编辑：

def booklist*(*request*)*:  
 # 获取数据库中的数据并保存到字典中传入页面  
 book\_list = models.Book.objects.all*()* # 测试分页功能  
 paginator = Paginator*(*book\_list, 5*)* page = request.GET.get*(*'page', 1*)* books = paginator.page*(*page*)* return render*(*request, 'booklist.html', *{*"book\_obj\_list": books*})*def edit\_book*(*request*)*:  
 if request.method == 'POST':  
 # 获取id  
 book\_id = request.POST.get*(*'id'*)* book\_name = request.POST.get*(*'BookName'*)* book\_author = request.POST.get*(*'Author'*)* book\_price = request.POST.get*(*'Price'*)* book\_publisher = request.POST.get*(*'Publisher'*)* book\_isbn = request.POST.get*(*'ISBN'*)* book\_isreadonline = request.POST.get*(*'IsReadOnline'*)* book\_booknum = request.POST.get*(*'BookNum'*)* # 根据id去数据库中找对象  
 book\_obj = models.Book.objects.get*(*id=book\_id*)* # 修改  
 book\_obj.BookName = book\_name  
 book\_obj.Author = book\_author  
 book\_obj.Price = book\_price  
 book\_obj.Publisher = book\_publisher  
 book\_obj.ISBN = book\_isbn  
 book\_obj.IsReadOnline = book\_isreadonline  
 book\_obj.BookNum = book\_booknum  
 # 保存  
 book\_obj.save*()* # 重定向到列表  
 return redirect*(*'/app/booklist'*)* else:  
 # 获取id  
 id = request.GET.get*(*'id'*)* # 去数据库中查找对应的数据  
 book\_obj = models.Book.objects.get*(*id=id*)* book\_obj\_list = models.Book.objects.all*()* # 返回页面  
 return render*(*request, 'edit\_book.html',  
 *{*"book\_obj": book\_obj, "book\_obj\_list": book\_obj\_list*})*

# 总结与展望

## 总结

从这个大作业开始构建框架到最后的完成，在这几天中，我学到了很多东西，我也遇到过很多的问题难题，百度查博文看视频，甚至查阅官方文档。诸如我在一个函数中得到本页面提交的信息，而我想把这个信息通过页面重定向的方式传递到另一个页面（函数）中，但是我在另一个页面中却没有得到这个数据，程序也没有报错。经查阅得知，是重定向函数本身是不能携带参数的，即便不会报错。而要解决的方法也很简单，将信息存储到cookie或者存储到session中，通过cookie/session去获取数据。然而这只是理想情况，实际报了一个格式错误，再引入序列化函数后解决了数据格式错误的问题，但是，依然拿不到想要的数据。在换了八百种方法、一番斗智斗勇后，终于实现了图书管理系统的基本功能。在几轮测试后，也没有大的问题出现。

## 展望

时间的紧缺导致只能实现基础功能，拓展功能实现的很少，下一步可以实现拓展功能。