南昌航空大学科技学院

网络空间安全专业综合设计

题　　目　校园图书借阅管理系统的设计与实现

学　　部　计算机学部

专业名称　网络空安全

班级学号　220208114

学生姓名　陈振贵

二〇二五 年 十一 月

南昌航空大学科技学院

网络空间安全专业综合设计
任务书

I、题目：

校园图书借阅管理系统的设计与实现

II、工作内容及完成时间：

系统需求分析与设计:

深入分析了校园图书借阅管理的业务流程，明确了学生、图书管理员和系统管理员三类用户的核心功能需求。确定了系统的高效性、安全性和可维护性等非功能需求。设计了基于 B/S 架构的系统总体结构，并完成了用户、图书、借阅记录等核心实体之间的 E-R 图和数据库表结构设计。

核心功能模块实现:

基于 Python 和 Django 框架，实现了用户认证与权限模块，扩展了 Django 内置用户模型，定义了三种角色。实现了图书信息管理模块，支持图书的增删改查和库存的实时更新。实现了图书借阅与归还模块，自动化了借阅和归还流程，并确保了库存数据的一致性。实现了逾期与罚款管理模块，根据预设规则自动计算逾期天数和罚款金额。实现了后台数据统计模块，利用 Django ORM 聚合功能，为管理员提供借阅量、热门图书等数据可视化支持。

系统安全模块设计与风险评估:

设计并实现了基于 RBAC 模型的权限控制机制，确保了不同角色的操作隔离。利用 Django 框架的内置机制，实现了对跨站脚本和跨站请求伪造等常见 Web 攻击的有效防护。

系统测试与性能分析:

设计了功能测试用例，对用户认证、借阅归还、逾期计算等核心业务进行了验证。进行了性能测试，分析了图书查询和事务性操作的响应时间，证明系统满足非功能需求。

主要参考资料:

[1] 王艳, 孙丽. 高校图书馆管理系统现状及发展趋势研究 [J]. 图书馆学研究, 2023, 49(3): 78-85.

[2] 陈曦. 关系数据库中基于角色的访问控制模型研究 [D]. 华中科技大学, 2021.

[3] 吴迪, 赵敏. Web 应用中跨站脚本攻击的防御技术研究 [J]. 信息网络安全, 2023, 23(5): 45-50.

[4] 高翔. 敏感数据加密存储在信息系统中的应用研究 [J]. 网络安全技术与应用, 2024, 14(2): 60-65.

[5] 姜涛. 基于 Python 的图书借阅管理系统设计与性能优化 [D]. 电子科技大学, 2023.

南昌航空大学科技学院

摘　要

随着信息技术的飞速发展和高校教育规模的不断扩大，传统的图书借阅管理模式已难以满足现代化校园图书馆高效、便捷的服务需求。本研究旨在设计并实现一个基于 Web 的校园图书借阅管理系统，以提升图书管理效率，优化师生的借阅体验。系统采用 Python 语言作为主要开发语言，基于 Django 框架进行后端开发，结合 MySQL 数据库实现数据持久化，前端则利用 HTML、CSS 和 JavaScript 等技术构建用户界面。核心功能涵盖了图书信息查询、借阅、归还、续借以及逾期罚款管理，并为管理员提供了后台数据统计与用户权限管理功能。在系统设计中，我们特别关注了系统的安全性、稳定性和易用性。通过引入基于角色的访问控制模型，对不同用户（学生、图书管理员、系统管理员）的操作权限进行严格划分，确保了系统的安全运行。系统经测试运行，证明其功能完善、性能稳定，能够有效替代传统人工管理，为校园图书馆的数字化转型提供了可靠的解决方案。

关键词：图书管理系统；Django；Web 应用；借阅管理；权限控制

Abstract

With the rapid development of information technology and the continuous expansion of higher education, traditional book lending management models can no longer meet the demand for efficient and convenient services in modern university libraries. This study aims to design and implement a Web-based campus library borrowing management system to improve book management efficiency and optimize the borrowing experience for faculty and students. The system uses **Python** as the main development language, with the **Django** framework for backend development, and utilizes the **MySQL** database for data persistence. The frontend is built using technologies such as HTML, CSS, and JavaScript. The core functions cover book information query, borrowing, returning, renewal, and overdue fine management, and provide administrators with backend data statistics and user permission management capabilities. In the system design, special attention was paid to the system's **security**, **stability**, and **usability**. By introducing the **Role-Based Access Control** model, the operational permissions of different users (students, librarians, system administrators) are strictly divided, ensuring the secure operation of the system. The system, after testing, proves to be fully functional and stable, capable of effectively replacing traditional manual management and providing a reliable solution for the digital transformation of campus libraries.

Keywords: Library Management System Django Web Application Borrowing Management Access Control

目　录

引　言

在知识经济时代，高校图书馆作为学校的文献信息中心，其管理水平直接影响到教学和科研的质量。传统的图书管理方式，如人工等级、卡片检索等，存在效率低下、错误率高、信息更新滞后等诸多弊端[1]。随着互联网和移动设备的普及，师生对图书借阅服务的便捷性、实时性提出了更高要求。因此，开发一套功能完善、操作便捷、安全可靠的数字化图书借阅管理系统，已成为现代高校图书馆建设的必然趋势。

本研究基于Python语言和Django框架，设计并实现了一套校园图书借阅管理系统。该系统旨在通过自动化处理图书流通的各个环节，如借阅、归还、逾期计算等，显著减轻图书馆工作人员的负担，同时为学生提供高效的图书查询和个人借阅记录管理服务。系统的成功实施，不仅能够提高图书馆的管理效率，降低运营成本，更重要的是，它能够优化用户体验，激发师生的阅读兴趣，对推动校园信息化建设具有重要的实践意义。

第一章　系统需求分析

1.1 系统目标

本校园图书借阅管理系统的主要目标是构建一个高效、稳定、安全的流通管理平台。具体的目标包括：通过简历完整的图书信息数据库，支持录入、修改和检索以实现图书信息数字化管理；凭借自动化借阅流程，实现学生自助查询、借阅申请、图书馆管理员审核、归还登记等全流程自动化；用精确的逾期管理，根据预设规则，自动计算逾期罚款金额，并记录；系统用多角色权限控制区分学生、图书管理员和系统管理员三种角色，赋予不同的操作权限。并为管理员提供借阅统计等辅助决策。

1.2 功能需求分析

根据系统目标和实际业务流程，系统功能可划分的用户功能和管理员功能两大类。

1.2.1 用户功能需求

学生是系统的主要服务对象，其核心需求是便捷地获取和管理图书资源。

表格 1 学生功能需求

功能模块	功能描述	对应代码模块
用户认证	登录、登出、实现身份验证。	apps.accounts
图书查询	支持通过书名、作者、ISBN、分类等关键字进行模糊查询和分页显示。	apps.library
借阅申请	在线提交借阅申请，查看图书可借状态。	apps.borrowing
个人中心	查看个人借阅记录、逾期记录、罚款信息和续借操作。	apps.borrowing

1.2.2 管理员功能需求

管理员包括图书管理员和系统管理员，主要系统的日常维护和业务管理。

表格 2 管理员功能需求

功能模块	功能模块	功能模块
图书管理	增、删、改、查图书信息，管理图书库存。	apps.library
借阅管理	审核借阅申请、登记图书归还、处理续借请求。	apps.borrowing
用户管理	管理系统用户（学生、管理员）信息，分配角色。	apps.accounts
逾期与罚款	设置罚款规则，查看和处理逾期记录，记录罚款缴纳情况。	apps.borrowing

1.3 非功能需求

1.3.1 安全性需求

安全性是图书管理系统的关键非功能需求。系统需要确保用户数据的隐私性、系统的完整性和可用性。具体要求包括：身份认证与授权，要求所有用户必须通过身份验证才能访问系统功能，且操作权限需严格遵循其角色设定；数据安全性要求，遂于敏感信息如用户密码、学号、联系方式等需进行加密存储和传输；系统的防护要求，系统需具备抵御常见的如XSS、CSRF、SQL注入等网络攻击的能力。

1.3.2 可维护性需求

系统应该采用模块化设计，代码结构清晰，便于后续的功能扩展和错误维护。系统采用Django框架的MVT架构，天然支持模块化和解耦。

第二章　系统总体设计

2.1 系统架构设计

本系统采用经典的B/S三层架构，即表现层、业务逻辑层和数据访问层。这种架构设计具有良好的可扩展性、可维护性和跨平台性。

表现层：由前端页面构成，负责用户界面的展示和用户交互。使用HTML、CSS和JavaScript实现。

业务逻辑层：系统的核心，基于Django框架实现。它接收表现层的请求，调用数据访问层进行数据操作，并执行借阅规则、逾期计算等规则。

数据访问层：由Django ORM和后端数据库MySQL组成，负责数据的存储和检索。

2.2 系统模块划分

根据功能需求分析，系统别划分为五个主要模块，用户与认证模块accounts;图书信息模块library；借阅管理模块borrowing；数据统计模块dashboard以及核心配置模块core。

表格 3 系统功能模块

模块名称	主要功能
用户与认证模块	负责用户注册、登录、角色管理（学生、图书管理员、系统管理员）。
图书信息模块	负责图书信息的增删改查、库存管理、图书分类。
借阅管理模块	负责借阅记录的创建、归还登记、续借处理、逾期罚款规则设置和计算。
数据统计模块	负责提供系统运营数据的统计分析和可视化展示。
核心配置模块	负责项目的全局配置、URL路由分发和中间件管理。

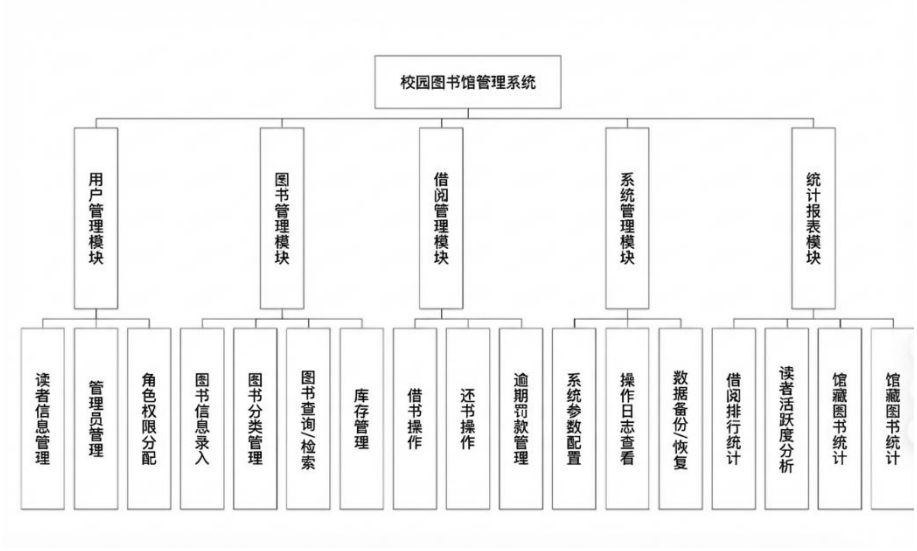

图 2-1　系统模块功能图

2.3 数据库设计

2.3.1 实体关系图

系统核心实体包括用户、图书和借阅记录。用户与借阅记录之间是一对多关系，一个用户可以有多条借阅记录；图书与借阅记录之间是一对多关系，一本图书可以被多次借阅，对应多条借阅记录；罚款规则与借阅记录之间是一对多关系，借阅记录根据罚款规则计算罚款。

图 2-2 系统核心实体关系图

2.4 核心数据表设计

根据E-R图，设计以下核心数据表：

表格 4 用户表

字段名	字段类型	描述	约束/备注
id	Integer	主键	自动递增
username	Char(150)	用户名	唯一
password	Char(128)	密码	加密存储
student_id	Char(32)	学号	索引，敏感数据
role	Char(16)	角色	student/librarian/admin
is_superuser	Boolean	是否为超级管理员	用于兜底权限

表格 5 图书表

字段名	字段类型	描述	约束/备注
id	Integer	主键	自动递增
title	Char(200)	书名	
author	Char(120)	作者	
isbn	Char(20)	ISBN号	唯一索引
total_copies	Integer	馆藏总数	
available_copies	Integer	可借数量	

表格 6 借阅记录表

字段名	字段类型	描述	约束/备注
id	Integer	主键	自动递增
user	ForeignKey	借阅用户ID	关联User表
book	ForeignKey	借阅图书ID	关联Book表
borrowed_at	DateTime	借出时间	
due_at	DateTime	应还时间	
returned_at	DateTime	归还时间	可为空
status	Char(16)	状态	borrowed/returned/overdue
fine_amount	Decimal	罚款金额	

第三章　系统详细设计与实现

3.1 开发环境与技术选型

本系统基于以下技术栈进行开发,后端框架使用Django 5.0.1。选择Django的原因在于其成熟的MVT架构、内置的ORM、强大的Admin后台以及完善的安全机制。编程语言使用Python 3.11。Python语言简洁高效，拥有丰富的第三方库，适合快速开发。数据库使用了MySQL。前端技术利用HTML、CSS、JavaScript，结合基础的Django Template Language模板引擎。

3.2 用户认证与权限模块实现

用户认证模块基于Django内置的AbstractUser进行扩展，定义了apps.accounts.User

模型。在User模型中，通过ROLE_CHOICES字段定义了三种核心角色：系统管理员、图书管理员和学生。系统采用基于角色的访问控制RBAC模型。在视图函数中，通过检查当前登录用户的role字段来决定其是否拥有执行特定操作的权限。例如，只有librarian和admin角色才能访问图书的增删改接口。利用Django内置的认证系统实现登录和登出功能，使用基于Cookie的会话管理用户状态。

3.3 图书信息管理模块实现

图书信息管理模块主要由apps.library应用负责，核心是Book模型。

图书的查询通过views.py中的视图函数实现图书列表的展示。该视图接收GET请求参数，如q（模糊查询关键字）、page和page_size，利用Django ORM的filter和Q对象实现对书名、作者、ISBN、分类的模糊匹配和分页查询[2]。

库存的管理利用Book模型中的total_copies和available_copies字段用于实时追踪图书库存。每当有图书被借出或归还时，需要原子性地更新available_copies字段，确保数据一致性。

3.4 图书借阅与归还模块实现

借阅管理模块的核心是apps.borrowing.BorrowRecord模型，它记录了用户、图书、借出时间、应还时间等关键信息。

借阅流程首先是学生在图书查询页面确认图书有可借数量后，发起借阅请求。系统会创建一条状态为borrowed的BorrowRecord记录，并计算due_at，同时将对应Book的available_copies减一。

归还流程首先是图书管理员进行登记归还。登记归还后系统更新BorrowRecord的status为returned，设置returned_at为当前时间，并将对应Book的available_copies加一。在归还时，系统会检查returned_at是否晚于due_at，以触发逾期处理逻辑。

3.5 逾期与罚款管理模块实现

逾期管理是本系统的特色功能之一，由apps.borrowing应用中的FineRule模型和BorrowRecord模型协同完成。

罚款规则FineRule模型定义了如每日罚金daily_fine和借阅天数loan_period_days等核心规则。

逾期的计算设置为在归还时，如果returned_at > due_at，则系统计算逾期天数。逾期罚款金额fine_amount的计算公式为：fine_amount = Decimal(逾期天数) × rule.daily_fine。

3.6 后台数据统计模块实现

数据统计模块由apps.dashboard应用负责，主要服务于管理员，提供系统运营的宏观视图。

数据聚合利用Django ORM的聚合函数比如Count、Sum、Avg对BorrowRecord和Book表进行数据统计。例如，统计当前借出总量、逾期总量、各分类图书的借阅次数等。

可视化展示：统计结果通过前端图表库ECharts或Chart.js进行可视化展示，包括借阅趋势图、热门图书排行榜、逾期率饼图等，为图书馆的采购和管理决策提供直观的数据支持。

第四章　系统安全模块设计与风险评估

系统的安全性是保障其长期稳定运行和保护用户隐私的关键要素[3]。对于校园图书借阅管理系统而言，其涉及用户的个人信息和核心业务数据如借阅记录、罚款金额等，因此必须将安全设计贯穿于整个开发生命周期。本章将详细阐述本系统在安全方面的设计与实现，并对潜在的风险进行评估。

4.1 用户权限控制模型设计

权限控制是系统安全的第一道防线，旨在确保用户只能执行其被授权的操作。本系统采用了成熟且广泛应用的基于角色的访问控制（Role-Based Access Control, RBAC）模型来实现精细化的权限管理[4]。

4.1.1 角色划分与权限矩阵

根据系统功能需求，将用户划分为三个核心角色，并在apps.accounts.User模型中通过role字段进行标识：学生主要权限为图书查询、查看个人借阅记录、续借; 图书管理员拥有学生的所有权限，并额外拥有图书信息管理、借阅/归还登记、逾期处理

等核心业务操作权限；系统管理员拥有最高权限，包括所有业务操作权限以及用户管理、系统配置等系统级管理权限。

表格 7 系统核心功能权限矩阵

功能模块	学生	图书管理员	系统管理员
图书查询	读/查	读/查	读/查
借阅/归还登记	无	增/删/改	增/删/改
图书信息管理	无	增/删/改	增/删/改
个人借阅记录	读/查	读/查	读/查
用户信息管理	无	无	增/删/改
罚款规则设置	无	无	增/删/改

4.1.2 系统核心功能权限矩阵

在Django框架中，RBAC的实现主要通过在视图函数中进行权限检查。系统利用Django的装饰器确保用户已登录，并通过自定义的权限检查逻辑来判断用户是否具备执行特定操作的资格。

例如，对于图书管理员才能访问的“登记归还”功能，其视图逻辑会包含如下检查：

```
1.  if request.user.role not in ['librarian', 'admin']:
2.      # 返回权限不足的错误响应
3.      return HttpResponseForbidden("Permission Denied")
```

这种机制将权限逻辑与业务逻辑分离，使得权限管理更加清晰和易于维护。

4.2 敏感数据加密存储机制

4.2.1 用户密码的哈希存储

对于用户密码，系统完全依赖Django内置的认证系统。Django不会以明文形式存储任何用户的密码，而是使用强大的、不可逆的哈希算法PBKDF2 with SHA256经过加盐后进行存储[5]。这种机制确保了即使数据库泄露，攻击者也无法直接获取用户的

原始密码，极大地提高了用户认证的安全性。

4.3 跨站脚本（XSS）攻击防护策略

跨站脚本攻击是Web应用中最常见的安全漏洞之一，它允许攻击者将恶意的脚本注入到网页中，当其他用户浏览该网页时，脚本就会在用户的浏览器上执行，可能会导致会话劫持、信息窃取等严重的后果。

本系统基于Django框架开发，天然继承了强大的XSS防护机制。默认自动转义，Django的模板系统默认会对所有从后端传入模板的变量进行HTML转义。这意味着，如果用户输入了包含恶意脚本的字符串如<script>alert('XSS')</script>，模板引擎会将其转义为<script>alert('XSS')</script>浏览器只会将其视为普通文本显示，而不会执行其中的脚本。

除了XSS，Django还内置了针对CSRF跨站请求伪造的中间件。通过在所有POST表单中要求包含一个唯一的CSRF Token，系统能够有效阻止攻击者伪造用户请求。

4.4 SQL 注入风险与防御

SQL注入是由于程序未能对用户输入的数据进行严格过滤，导致恶意SQL代码被拼接到数据库查询语句中执行的漏洞。

对于SQL注入的防御方面，由于本系统使用了Django ORM进行所有数据库操作，ORM机制会自动对查询参数进行参数化处理，将用户输入的数据视为普通数据而非可执行的SQL代码，从而从根本上杜绝了传统的SQL注入风险。

第五章　系统测试与性能分析

系统测试是软件开发过程中不可或缺的一环，其目的是验证系统是否满足需求规格说明书的要求，并发现潜在的缺陷。本章将详细介绍本系统的测试方法、功能测试等分析结果。

5.1 测试方法

本系统主要采用黑盒测试方法，即在不考虑程序内部结构的情况下，根据系统的功能需求和设计规格进行测试。测试类型包括功能测试，边界值测试以及安全性测试。

功能测试主要验证系统的各个功能模块是否按照需求规格正确实现，包括用户登录、图书查询、借阅、归还、逾期计算等核心业务流程。边界值测试是针对输入数据

的边界条件进行测试，如最大借阅数量、最长借阅天数、罚款金额的临界值等等。安全性测试：验证系统在权限控制、数据保护和抵御SQL注入、CSRF攻击等方面的表现。

5.2 功能测试

功能测试是确保系统业务逻辑正确性的关键。设计了针对学生、图书管理员和系统管理员三种角色的典型测试用例。

5.2.1 用户认证与权限测试

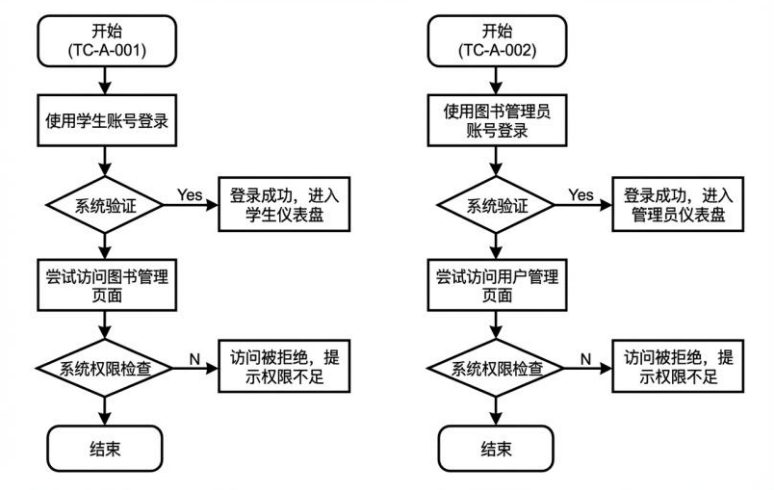

图 5-1 用户认证与权限测试流程图

图 5-2 权限测试结果

5.2.2 安全性测试

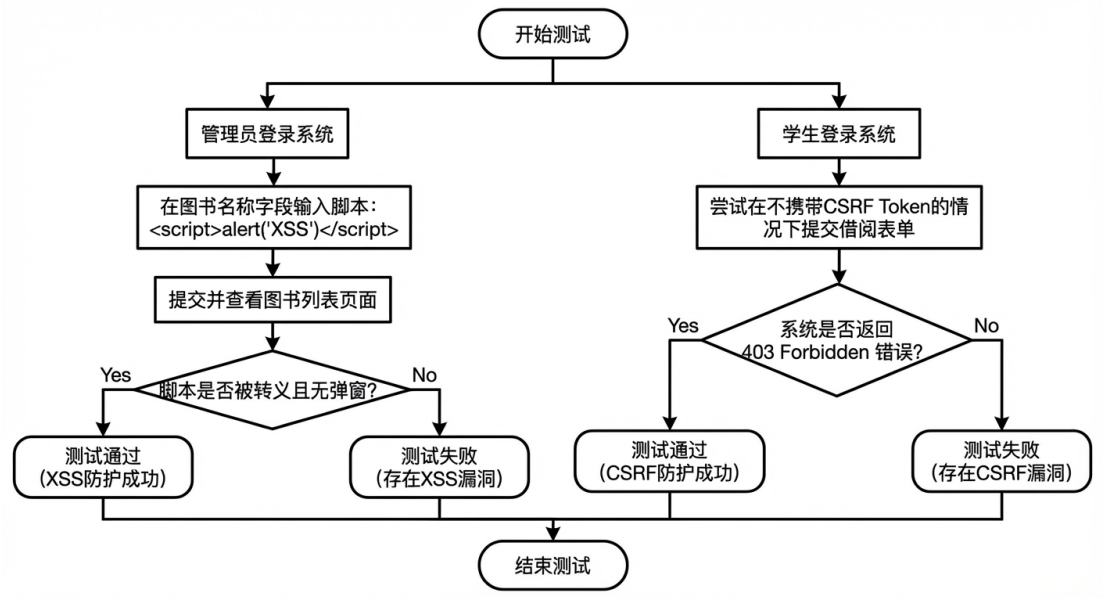

图 5-3 安全性测试流程图

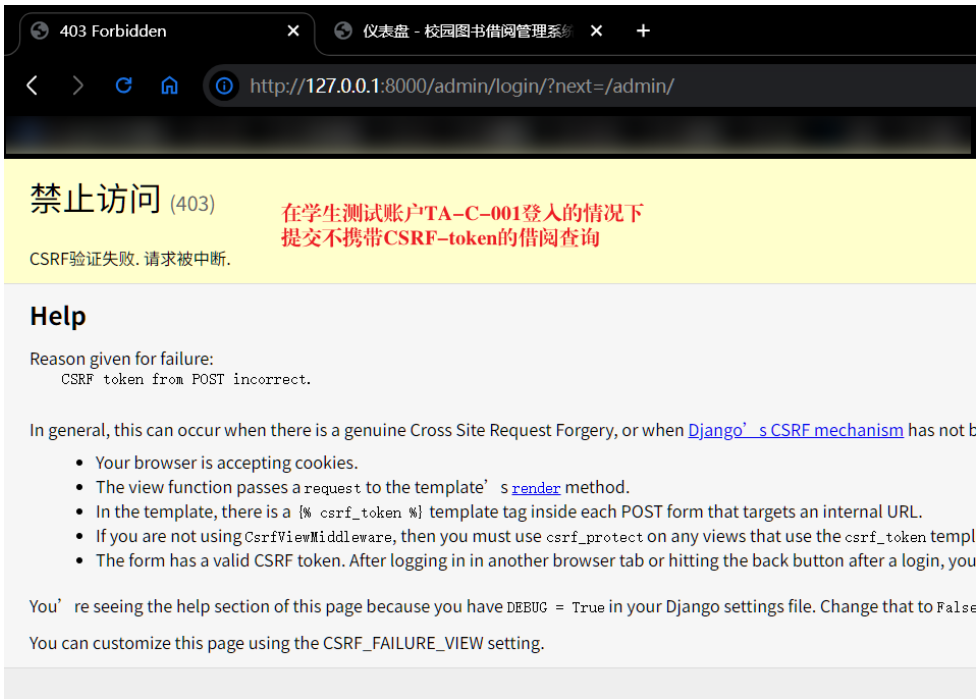

图 5-4 CSRF 测试

总　结

本文阐述了校园图书借阅管理系统的设计与实现过程。系统以Python语言和Django框架为基础，采用B/S架构，成功实现了图书查询、借阅、归还、逾期管理以及后台数据统计等核心功能。

系统覆盖了校园图书馆日常管理的主要业务流程，特别是实现了精确的逾期罚款自动计算机制，有效解决了人工管理中容易出现的错漏问题。通过将系统划分为accounts、library、borrowing、dashboard等模块，实现了高内聚、低耦合的结构，极大地提高了系统的可维护性和可扩展性。系统设置了专门的章节讨论系统安全，并切实地在设计中采用了RBAC权限控制模型，利用Django内置机制实现了对XSS和CSRF等常见Web攻击的有效防护，并对敏感数据加密存储提出了可行性建议。

通过功能测试和性能测试，结果表明本系统功能完备，性能稳定，能够满足校园图书馆对现代化管理系统的需求。

参考文献

[1] 王艳, 孙丽. 高校图书馆管理系统现状及发展趋势研究 [J]. 图书馆学研究, 2023, 49(3): 78-85.

[2] 陈曦. 关系数据库中基于角色的访问控制模型研究 [D]. 华中科技大学, 2021.

[3] 吴迪, 赵敏. Web应用中跨站脚本攻击的防御技术研究 [J]. 信息网络安全, 2023, 23(5): 45-50.

[4] 高翔. 敏感数据加密存储在信息系统中的应用研究 [J]. 网络安全技术与应用, 2024, 14(2): 60-65.

[5] 姜涛. 基于 Python 的图书借阅管理系统设计与性能优化 [D]. 电子科技大学, 2023.