

深圳大学实验报告

课程名称： Python程序设计

项目名称： 实验四：Blog应用程序设计

学 院： 人工智能学院

专 业： 计算机科学与技术

指导教师： 樊超

报告人： 陈怡婷 学号： 2024440124

实验时间： 2025年12月25日

提交时间： 2025年12月25日

流程图:

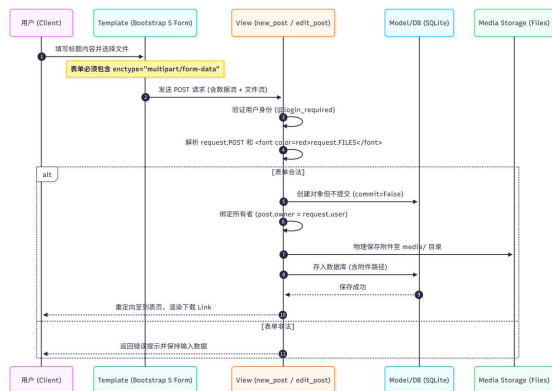


图 1 项目逻辑架构图

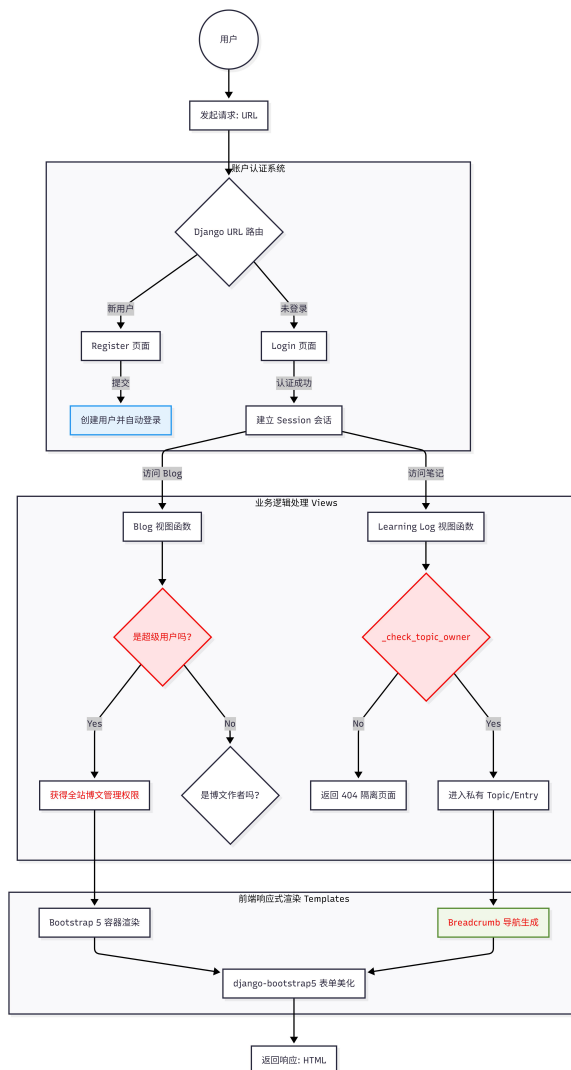


图 2 文件上传时序图

1 Objectives

本次实验旨在通过开发一个功能完备的 Web 应用程序，使学生达到以下目的：

- (1) 架构掌握：熟练运用 Django MTV 模式完成 Web 全流程开发。
- (2) 功能集成：掌握多媒体处理机制，实现博文附件上传与存储。
- (3) 权限认证：自主搭建注册登录系统，实现超管全局管理与多级权限控制。
- (4) 交互优化：集成 Bootstrap 5 实现响应式设计，优化前端交互体验。
- (5) 业务构建：开发并实现“公共博客”与“私密笔记”双重逻辑系统。

2 Overview

2.1 核心内容

认证系统：自主实现注册与登录，支撑全站权限控制。

Blog 系统（增强版）：支持附件上传（request.FILES 处理）、预览及下载。实现普通用户管理个人内容、超管全局管理的多级权限体系。

Learning Log 系统：实现 Topic 与 Entry 的一对多关联，确保用户数据私有化。

2.2 技术栈

后端：Python 3.13 / Django 6.0

前端：Bootstrap 5

数据库：SQLite3

3 Implementations

根据要求，以下部分为我独立实现的功能细节：

3.1 账户认证与多级权限系统

- (1) 自主搭建认证体系：利用 django.contrib.auth 实现注册、登录及注销功能；并在注册视图集成 login() 逻辑，实现注册后自动登录的顺滑体验。
- (2) 超级用户管理特权：引入 user.is_superuser 判定，赋予管理员全局数据清理权限，构建了“普通用户私有数据+管理员全局维护”的分层管理模式。

3.2 博客系统进阶开发 (19 - 1, 19 - 5)

- (1) 文件上传与存储：通过 request.FILES 与 FileField 实现博文附件上传，并配置 Media 资源管理路径，支持附件的持久化存储与在线下载。
- (2) 访问权限分离：实现“广场式”阅读逻辑，匿名用户仅可浏览，登录用户方可发帖，且仅文章所有者（Owner）具备编辑权限。
- 3.3 个人笔记系统私有化定制
- (1) 后端安全拦截逻辑：封装 _check_topic_owner 函数，针对手动修改 URL ID 的越权行为强制抛出 Http404，实现数据物理层面的绝对隔离。
- (2) 结构化归档：基于一对多(Topic - Entry)模型关系，支持用户在自定义主题下进行持续性笔记记录。
- 3.4 交互安全与 UI/UX 优化
- (1) 双重删除确认机制：为所有删除操作设计了中间确认视图（GET 跳转警告页，POST 执行删除），有效规避误操作风险。
- (2) 全站响应式美化：集成 Bootstrap 5，将原始列表重构为 Card（卡片）布局，并引入 Breadcrumbs（面包屑）导航优化深层页面的回溯体验。
- (3) 表单标准化：利用 django - bootstrap5 统一渲染全站表单，确保移动端与 PC 端的视觉风格一致。
- 3.5 全站 UI/UX 响应式美化
- (1) 响应式布局重构：全站集成 Bootstrap 5，将原始 HTML 列表重构为Card（卡片）与Badge（徽章）组合，显著提升视觉层次感。
- (2) 面包屑导航：在笔记编辑等深层页面引入 Breadcrumbs 插件，优化了复杂层级下的导航回溯体验。
- (3) 表单组件优化：利用 django-bootstrap5 统一渲染认证及业务表单，实现全站输入框与按钮的风格一致性，并确保移动端完美适配。

4 Results

4.1 账户认证与导航权限测试 (registration/login.html)

本模块验证了用户注册、登录后，导航栏根据身份动态切换的逻辑

场景描述	预期结果	实际结果
<div>响应式注册界面，如图 3所示</div> <div>主界面如图 5所示，登录状态导航栏，如图 6所示</div>	<div>注册表单在移动端正常缩放，通过 UserCreationForm 成功创建用户并自动登录。</div> <div>登录后，导航栏显示 "Hello, [user]"，隐藏 Login/Register，显示 Logout。</div>	<div>符合。</div> <div>符合。</div>

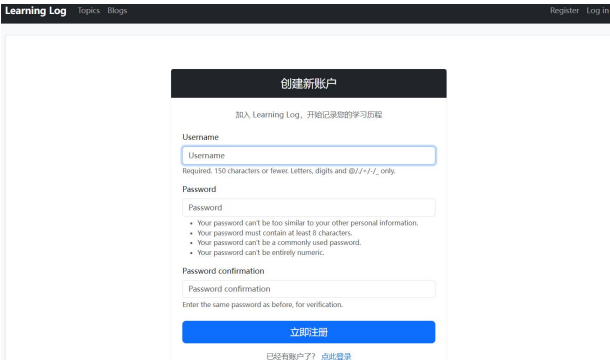


图 3 注册界面

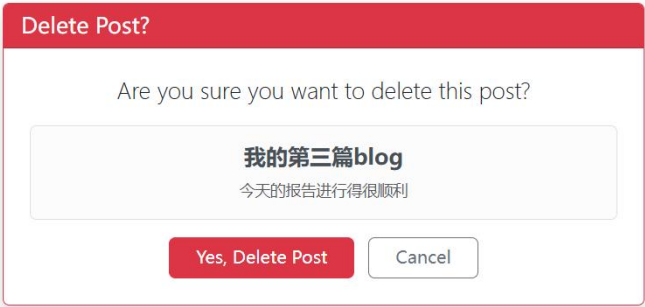


图 4 删除确认图



图 5 主页面



图 6 登陆界面

4.2 带附件的 Blog 系统与超级用户特权 (blogs.html)

验证 19-1、19-5 要求的实现，特别是文件上传与超级用户管理逻辑。

场景描述	预期结果	实际结果
<div>文件附件上传，如图 7</div>	<div>在发布博文时选择 PDF/图片文件，request.FILES 成功处理并存储至 media</div>	<div>符合。列表页成功显示附件下载链接。</div>

场景描述	预期结果	实际结果
所示	目录。	
19-5 普通用户权限	用户 A 无法看到用户 B 博文下的 Edit/Delete 按钮。	符合。
超级用户特权，如图 8 所示	超级用户登录后，可以对全站任何人的博文进行删除操作，用于全局维护。	符合。

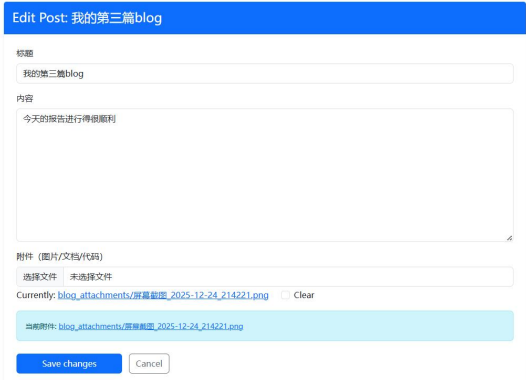


图 7 支持附件上传

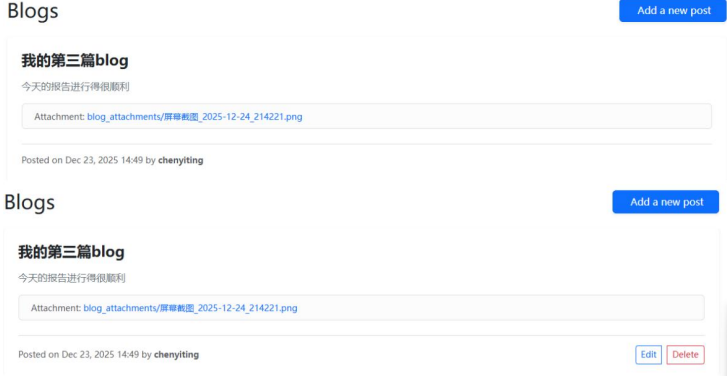


图 8 普通用户与超级用户对比图

4.3 私人笔记本系统与 UI 优化 (topics.html / topic.html)

验证个人笔记的私有性以及美化后的 Breadcrumbs 导航。

场景描述	预期结果	实际结果
私有权限拦截，如图 9 所示	用户 B 试图通过手动输入 URL 访问用户 A 的 Topic ID。	符合。抛出 Http404，数据被物理隔离。
面包屑导航功能，如图 10 所示	在 edit_entry 页面，用户能通过导航链接快速回退到对应的 Topic 详情页。	符合。增强了深度操作的路径感。

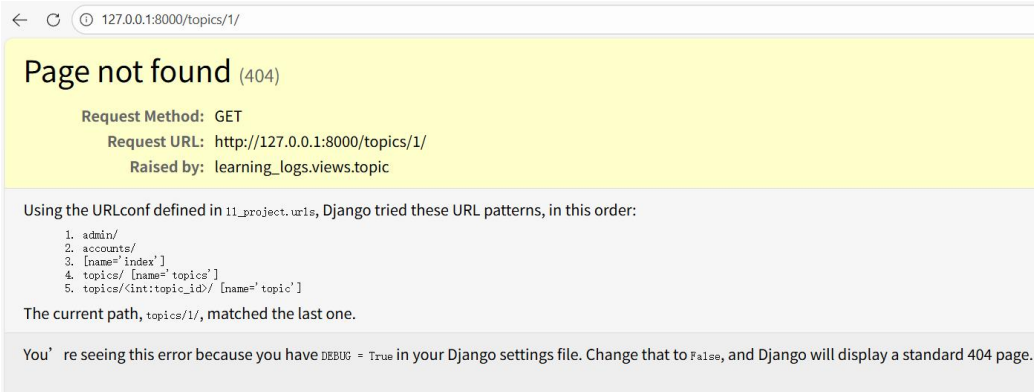


图 9 私有权限拦截

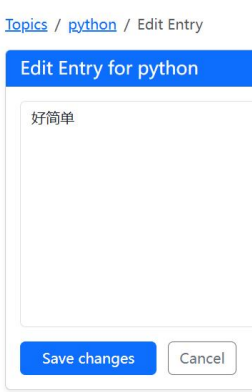


图 10 导航功能

4.4 安全交互设计测试 (delete_confirm.html)

验证你独立开发的二次确认删除机制。

场景描述	预期结果	实际结果
删除确认流	点击删除后，先进入中间确认页，展示要删除的内容标题，用户点击“确认”后执行删除。	符合。有效防止了误删操作。

5 Discussion

具体代码细节，登录GitHub 仓库地址：https://github.com/chenyitingkitty20051126-lgtm/Django_blog_system
技术实现与安全：基于 Django MTV 模式开发，通过后端私有函数与超管逻辑构建多级权限体系，严防越权访问。成功集成文件上传、Media 存储及自主身份认证模块，确保用户数据私有安全。
交互优化与体验：采用 Bootstrap 5 响应式布局，引入面包屑导航与删除二次确认机制，显著降低用户操作迷失感，提升系统工业化交互水准。
总结结论：项目圆满完成 Blog 与笔记系统各项任务，实现数据隔离与全局管理。通过实践，深刻掌握了 Django 认证系统、权限闭环及前端引导反馈的核心技术。

指导教师批阅意见：

成绩评定：

指导教师签字：

备注：

- 报告内的项目或内容设置，可根据实际情况加以调整和补充。
- 教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。