Python 学习平台项目文档

**一、项目概述**

**1.1 项目背景**

为 Python 学习者提供论坛交流、资源共享的在线平台，支持用户发帖、评论、上传下载学习资料，包含权限管理（普通用户 / 管理员）。

**1.2 核心功能**

**用户系统**：注册 / 登录 / 退出，管理员权限

**论坛模块**：帖子发布、排序（时间热度）、评论回复

**资源中心**：文件上传、分类管理、下载统计

**权限控制**：管理员后台、帖子编辑删除

**二、使用技术**

| **模块** | **技术 / 工具** | **说明** |
| --- | --- | --- |
| 后端 | Flask + SQLAlchemy | 轻量级框架，ORM 数据库管理 |
| 前端 | HTML + Bootstrap | 模板引擎，响应式 UI |
| 数据库 | Mysql | 轻量数据库，支持快速迭代 |
| 状态管理 | Flask-Login | 用户会话管理 |
| 数据库迁移 | Flask-Migrate | 版本控制，表结构同步 |
| 依赖管理 | virtualenv + requirements.txt | 虚拟环境隔离，依赖统一管理 |

**三、项目结构**

python\_learning\_platform/

├── app/ # 核心代码

│ ├── \_\_init\_\_.py # 应用初始化

│ ├── models.py # 数据库模型（User/Post/Resource）

│ ├── routes.py # 路由逻辑

│ ├── forms.py # 表单（注册/登录）

│ └── templates/ # 模板文件

│ ├── layout.html # 基础模板（导航栏）

│ ├── auth/ # 认证页面（login/register）

│ ├── forum/ # 论坛模块（posts.html）

│ └── resources/ # 资源模块（upload/index）

├── config.py # 配置文件（密钥/数据库URI）

├── requirements.txt # 依赖清单

├── run.py # 启动入口

└── migrations/ # 数据库迁移脚本（自动生成）

**四、部署指南**

**4.1 环境准备**

1. **安装 Python**：3.9+（推荐 3.11）
2. **创建虚拟环境**：

python -m venv venv

# 激活：

venv\Scripts\activate.bat # Windows

source venv/bin/activate # macOS/Linux

1. **安装依赖**：

pip install -r requirements.txt

**4.2 初始化数据库**

**迁移表结构**：

flask db init # 初始化迁移环境（首次）

flask db migrate # 生成迁移脚本

flask db upgrade # 应用迁移（创建表）

**4.3 运行项目**

python run.py

# 访问：http://localhost:5000

**五、功能模块详解**

**5.1 用户系统**

| **路由** | **方法** | **功能描述** | **权限** |
| --- | --- | --- | --- |
| /auth/register | GET/POST | 注册（邮箱唯一，密码加密） | 游客 |
| /auth/login | GET/POST | 登录（记住我功能待实现） | 游客 |
| /logout | GET | 退出登录 | 登录用户 |

**5.2 论坛模块（forum路由）**

**核心逻辑**：

帖子排序：最新 / 最旧 / 最多投票（通过 URL 参数?sort=控制）

权限控制：作者和管理员可编辑 / 删除帖子

**模板变量**：

posts：帖子列表（含用户、评论关联）

sort\_option：当前排序方式（模板中高亮显示）

**资源系统和代码运行暂未实现**

**六、数据库设计**

**6.1 表结构**

**1. User（用户）**

| **字段名** | **类型** | **描述** | **约束条件** |
| --- | --- | --- | --- |
| id | Integer | 用户唯一标识 | 主键，自增 |
| username | String(150) | 用户名 | 唯一，非空 |
| email | String(150) | 用户邮箱 | 唯一，非空 |
| password\_hash | String(255) | 加密后的用户密码 | 非空 |
| role | String(20) | 用户角色（如普通用户、管理员） | 默认值为 'user' |

**2. Post（帖子）**

| **字段名** | **类型** | **描述** | **约束条件** |
| --- | --- | --- | --- |
| id | Integer | 帖子唯一标识 | 主键，自增 |
| title | String(200) | 帖子标题 | 非空 |
| content | Text | 帖子内容 | 非空 |
| user\_id | Integer | 发布帖子的用户 ID | 外键，关联 User 表的 id，非空 |
| votes | Integer | 帖子的投票数 | 默认值为 0 |
| created\_at | DateTime | 帖子创建时间 | 默认值为当前时间 |

**3. Comment（评论）**

| **字段名** | **类型** | **描述** | **约束条件** |
| --- | --- | --- | --- |
| id | Integer | 评论唯一标识 | 主键，自增 |
| content | Text | 评论内容 | 非空 |
| user\_id | Integer | 发表评论的用户 ID | 外键，关联 User 表的 id，非空 |
| post\_id | Integer | 评论所属的帖子 ID | 外键，关联 Post 表的 id，非空 |
| parent\_comment\_id | Integer | 父评论 ID（用于实现评论的回复功能） | 外键，关联 Comment 表的 id |
| created\_at | DateTime | 评论创建时间 | 默认值为当前时间 |

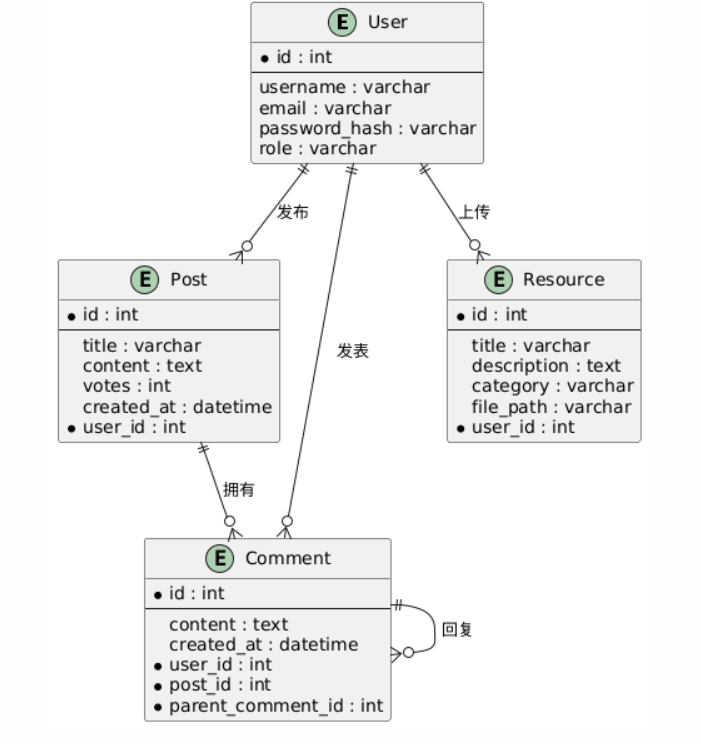
**4. Resource（资源）**

| **字段名** | **类型** | **描述** | **约束条件** |
| --- | --- | --- | --- |
| id | Integer | 资源唯一标识 | 主键，自增 |
| title | String(200) | 资源标题 | 非空 |
| description | Text | 资源描述 | 可选 |
| category | String(100) | 资源分类 | 可选 |
| file\_path | String(200) | 资源文件的存储路径 | 可选 |
| user\_id | Integer | 上传资源的用户 ID | 外键，关联 User 表的 id，非空 |

**6.2 外键约束**

Post.user\_id → User.id（级联删除）

Resource.user\_id → User.id（级联删除）



**七、附录**

**依赖清单（requirements.txt）**

Flask==2.3.2

Flask-SQLAlchemy==3.0.3

Flask-Login==0.6.2

Flask-Migrate==4.0.4

Werkzeug==2.3.7

WTForms==3.0.1

python-dotenv==1.0.0