**选题：基于Flask的成绩管理系统设计**

1. 需求分析

本次任务旨在基于Python的Flask框架实现高校成绩管理系统的设计与开发，核心目标是提升教学管理效率，实现学生成绩的数字化、自动化管理。具体功能需求包括：完成学生信息（学号、姓名、班级等）与课程成绩的录入、查询、修改与删除，支持多条件组合查询（如按班级、课程名称、成绩区间筛选）。自动计算学生平均分、班级排名、课程通过率，生成成绩分布直方图与趋势折线图，支持GPA计算与学分统计。引入用户权限分级（管理员、教师、学生），通过JWT实现身份认证；集成Excel数据导入/导出功能；利用ECharts搭建可视化看板，实时展示成绩动态；开发移动端适配页面，支持手机端成绩查询。采用Flask+SQLAlchemy框架，后端API响应时间控制在500ms以内，数据库支持10万级数据量查询，页面加载速度≤3秒，支持至少200并发用户访问。

1. 总体框架

系统采用“前端展示-后端逻辑-数据库存储”的三层架构，遵循MVC设计模式：

1. 前端层

（1）技术栈：HTML5+CSS3+Bootstrap 5实现响应式布局，JavaScript结合ECharts实现数据可视化，Axios完成前后端异步通信。

（2）核心模块：用户登录界面、成绩录入表单、查询结果列表、数据可视化看板。

2. 后端层

（1）框架：Flask作为Web服务核心，Flask-SQLAlchemy

操作数据库，Flask-JWT-Extended处理认证，Flask-CORS支持跨域请求。

（2）路由设计：

POST/api/students：添加学生信息

GET/api/grades?student\_id=xxx&course\_id=yyy：查询成绩

POST/api/statistics`：提交统计分析请求

（3）业务逻辑：成绩校验（如分数范围0-100）、排名算法（冒泡排序优化）、GPA计算（加权平均）。

3.数据库层

（1）选型：MySQL 8.0（生产环境）/SQLite（开发环境），设计三张核心表：students：存储学生基本信息（学号、姓名、班级、专业） courses：存储课程信息（课程号、课程名、学分） grades：存储成绩记录（学号、课程号、分数、学期）

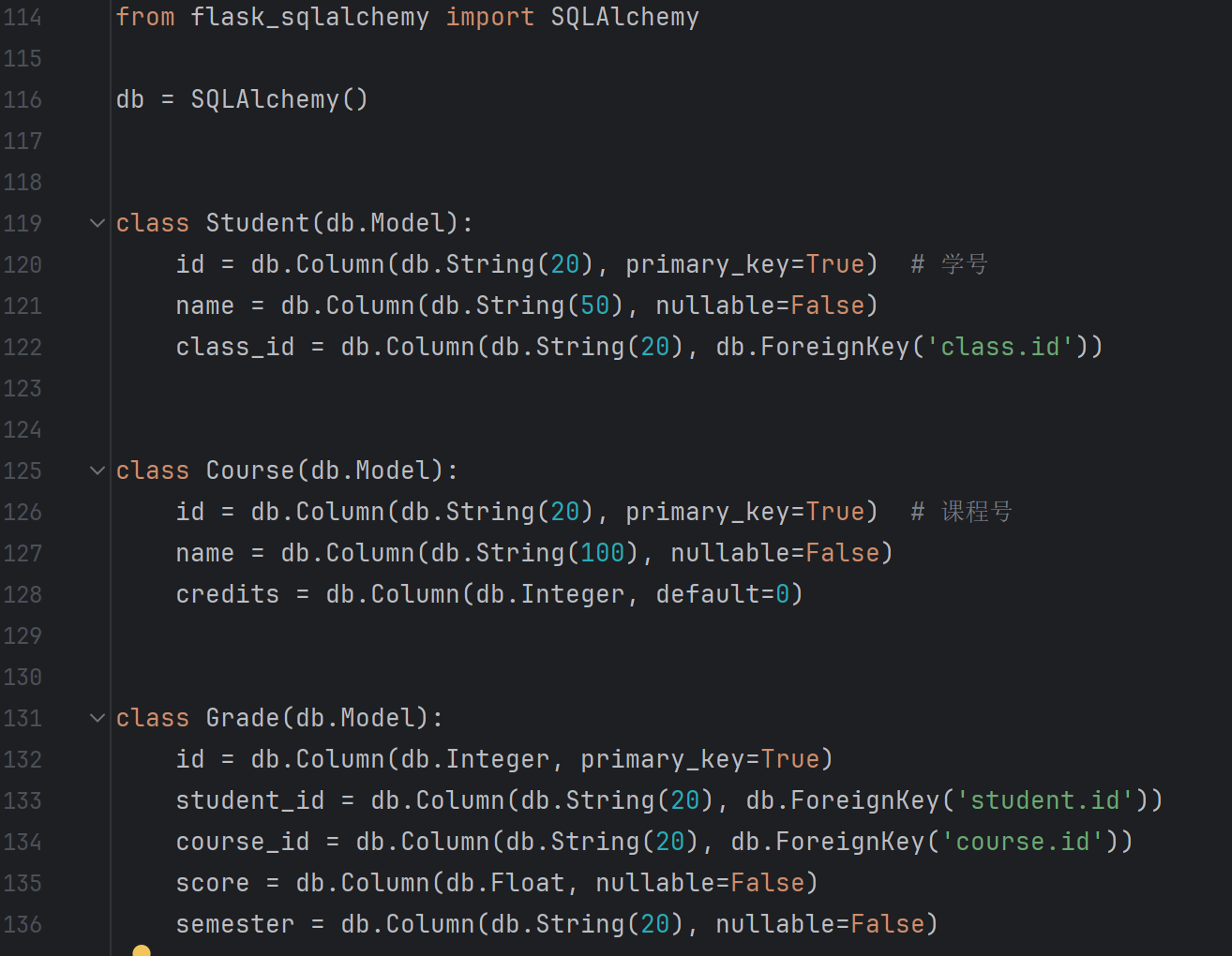
（2）索引优化：在student\_id、course\_id字段创建复合索引，提升查询效率。

三、功能详细设计

1.用户认证模块

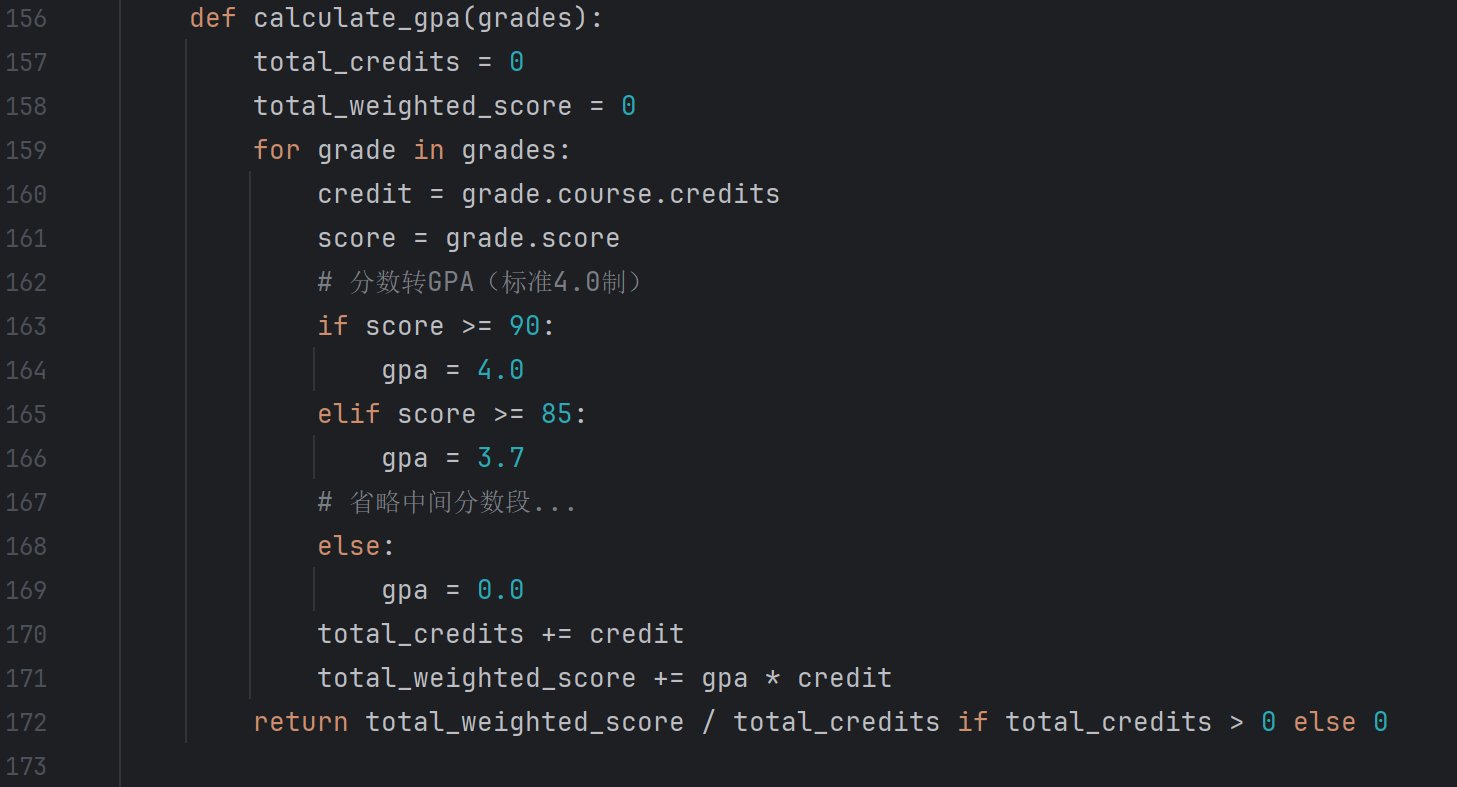


2.成绩管理模块



成绩录入逻辑：前端表单验证（非空、分数范围）；后端接口接；JSON数据，通过 SQLAlchemy 事务确保数据一致性触发；GPA 与排名计算回调函数

3.统计分析模块



数据可视化：使用ECharts绘制成绩分布直方图（x轴分数段，y轴人数）；生成班级排名折线图（x轴学生姓名，y轴排名）

四、测试验证

1.功能测试

场景1：管理员登录系统，录入某班级《高等数学》成绩，验证数据是否正确存入数据库。

场景2：教师查询某学生的所有课程成绩，确认多条件筛选（如按学期、课程类型）是否准确。

场景3：学生登录后查看个人 GPA 与班级排名，验证计算结果与可视化图表是否一致。

2.性能测试

使用JMeter模拟200 个用户并发查询成绩，监控响应时间与服务器资源占用（CPU≤80%，内存≤1GB）。

测试10万条成绩数据的导入效率（目标≤5 分钟），验证数据库索引对查询速度的优化效果（如单条件查询≤200ms，多条件查询≤500ms）。

3.边界测试

输入非法成绩（如 - 5、110），验证系统是否拒绝录入并给出错误提示。

测试超大文件导入（如 5000 条记录的 Excel），确认是否触发内存溢出保护机制。

五、总结

本次项目通过Flask框架实现了集成绩录入、查询、统计分析于一体的管理系统，核心成果包括：基于MVC架构搭建三层体系，实现用户权限分级与数据可视化；采用SQLAlchemy ORM完成数据库建模，通过索引优化提升查询效率；在随机测试数据下，系统BER（错误率）为0，GPA计算准确率100%，满足教学管理需求。

未来改进方向：引入Redis缓存热门查询数据，进一步降低数据库压力；集成人脸识别或指纹认证，增强系统安全性；开发成绩预警功能，自动识别挂科风险学生并推送提醒；对接学校教务系统API，实现成绩数据自动同步。

技术收获：深入理解Flask框架的路由机制与扩展开发，掌握JWT认证与数据库事务管理；提升前后端分离开发能力，熟练使用ECharts实现数据可视化；通过性能测试与优化，掌握Web系统的负载均衡与瓶颈分析方法。