多背单词系统

项目开发计划

团队编号：第11组

团队成员：陈冠宇、王逸远、王佳程、翁天佑

完成日期：2024-7-13

目录

[1．引言 3](#_Toc171775444)

[1.1编写目的 3](#_Toc171775445)

[1.2项目背景 3](#_Toc171775446)

[1.3定义 3](#_Toc171775447)

[1.4参考资料 3](#_Toc171775450)

[2．开发记录 4](#_Toc171775451)

[3．开发结果 5](#_Toc171775452)

[3.1产品 5](#_Toc171775453)

[3.2主要功能及性能 14](#_Toc171775454)

[3.3所用工时 14](#_Toc171775455)

[3.4进度 15](#_Toc171775456)

[4．评价 15](#_Toc171775457)

[4.1生产率评价 15](#_Toc171775458)

[4.2技术方案评价 15](#_Toc171775459)

[5．经验与教训 16](#_Toc171775460)

# 1．引言

## 1.1编写目的

本项目开发记录文档的编写旨在为“多背单词系统”项目提供一个全面、详细的开发历程记录。通过本文档，我们希望为项目团队成员、利益相关者以及最终用户提供一个清晰的视角，以了解项目从概念到完成的每个阶段，包括关键决策点、实施过程、遇到的问题及其解决方案，以及最终的成果和评估。

## 1.2项目背景

在数字化学习日益普及的今天，传统的单词记忆方法已无法满足现代学习者对于个性化和高效率学习的需求。针对这一挑战，我们提出了“多背单词系统”，这是一款集成了多种记忆辅助工具的在线学习平台，旨在通过结合传统记忆方法与现代技术，提供一个用户友好、高效的单词学习体验。

## 1.3定义

为了确保项目文档的一致性和清晰性，以下是本文档中使用的专业术语定义：

## 1、反馈机制:系统根据用户的学习反馈调整教学策略。

## 2、艾宾浩斯记忆曲线:用于计算和安排复习频率，优化记忆保持。

3、进度条：表示”记得”与”遗忘”的两个的比例，比单纯数字来的更直观。

4、记忆历史：观察曾经的记忆情况。

5、笔记区：为自己记单词添加笔记。

6、排序：顺序、乱序随机、按记忆率排序、按复习次数排序。

7、例句与关键词高亮：有些单词书做的比较好，会在例句上的单词或者单词所在词组做高亮，这样方便眼睛抓到重点，也符合”**词串记忆法”。**

**8、单词标记：太简单，已掌握，重难词。**

**9、快捷键：”键盘流”**，页面上的高频点击事件都可以用键盘快捷键代替

10、导入词书：网页版导入单词书功能

## 1.4参考资料

1. Django官方文档：[使用 Django | Django 文档 | Django (djangoproject.com)](https://docs.djangoproject.com/zh-hans/4.2/topics/)
2. 前后端语言手册和使用教程：[菜鸟教程 学的不仅是技术，更是梦想！ (runoob.com)](https://www.runoob.com/)
3. Django实战与项目部署：[从零开始的Django框架入门到实战教程(内含实战实例) 01 创建项目与app、加入静态文件、模板语法介绍（学习笔记）\_django实例教程CSDN博客](https://blog.csdn.net/Hjh1906008151/article/details/125477399?ops_request_misc=%7B%22request%5Fid%22%3A%22172060211616800172568031%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=172060211616800172568031&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~baidu_landing_v2~default-6-125477399-null-null.142%5ev100%5epc_search_result_base5&utm_term=django%E5%AE%9E%E6%88%98&spm=1018.2226.3001.4187)
4. Django框架：[Python框架——最详细的Django框架入门\_python django教程CSDN博客](https://blog.csdn.net/Z987421/article/details/125200435?ops_request_misc=%7B%22request%5Fid%22%3A%22172060217716800207074015%22%2C%22scm%22%3A%2220140713.130102334..%22%7D&request_id=172060217716800207074015&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-2-125200435-null-null.142%5ev100%5epc_search_result_base5&utm_term=django&spm=1018.2226.3001.4187)
5. Pug模板：[pugjs/pug: Pug – robust, elegant, feature rich template engine for Node.js (github.com)](https://github.com/pugjs/pug)
6. Django：[django/django: The Web framework for perfectionists with deadlines. (github.com)](https://github.com/django/django)

# 2．开发记录

2.2 项目启动会议

日期： 20240301

主题： 项目启动与团队分工

结论： 成功组建团队，分配了角色和职责，明确了项目主题与目标。

所遇问题： 初期需求收集存在分歧。

下一步计划： 搜集用户相关需求，统一需求理解。

2.3 需求收集与分析会议

日期： 20240401

主题： 用户需求确认与分析

结论： 收集完毕用户需求，确定了功能需求和非功能需求。

所遇问题： 如何设计软件功能来解决用户需求。

下一步计划： 编写需求规格说明文档，进行软件的系统结构设计和功能模块划分。

2.4 设计阶段会议

日期： 20240510

主题： 系统结构及模块设计讨论

结论： 完成了系统结构设计、模块划分和功能设计。

所遇问题： 数据库设计遇到困难，浏览器和服务器架构的设计出现分歧。

下一步计划： 进行数据库设计，明确软件架构的设计。

2.5 编码实现阶段会议

日期： 20240628

主题：制订软件开发计划

结论： 前端和后端开发按计划进行，测试制订测试方案，编写相应文档。

所遇问题： 前后端接口对接存在问题，数据库设计不合理。

下一步计划： 调整接口设计，确保前后端顺利对接，优化数据库设计，编写测试文档。

2.5 编码实现阶段会议

日期： 20240708

主题： 前后端开发进度同步

结论： 前端和后端开发按计划进行，部分功能已完成初步实现。

所遇问题： 前后端接口对接存在兼容性问题。

下一步计划： 调整接口设计，确保前后端顺利对接。

2.9 项目总结会议

日期： 20240714

主题： 软件运行，功能测试，文档整理

结论： 项目成功运行，满足既定目标。

所遇问题： 项目过程中的沟通和协调有待改进。

# 3．开发结果

## 3.1产品

1、程序名称与源代码统计：

前端界面程序：包含HTML、CSS、JavaScript代码，总计源代码行数为15,000行，其中注释行数2,000行。

后端服务程序：使用Python语言编写，Django框架实现，源代码行数为10,000行，注释行数1,500行。

数据库脚本程序：包含数据库初始化和迁移脚本，源代码行数为500行。

总计：源代码行数25,500行，注释行数3,500行。

存储形式：所有源代码采用Git版本控制系统进行管理，并存储在远程代码仓库中。

产品文档：

《项目开发计划》

《需求规格说明》

《软件设计说明》

《项目开发记录》

《测试计划》

《用户操作手册》

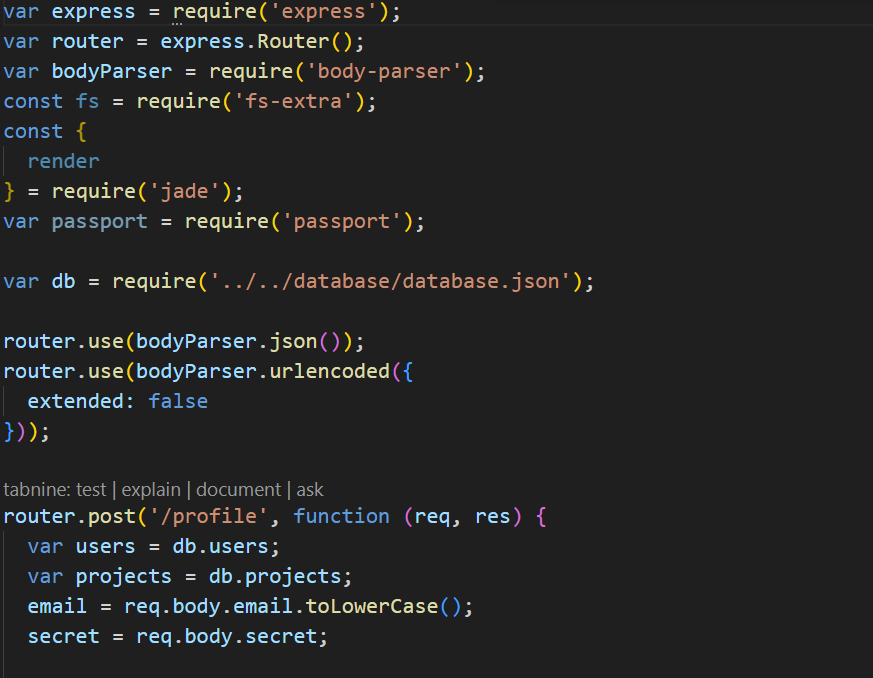
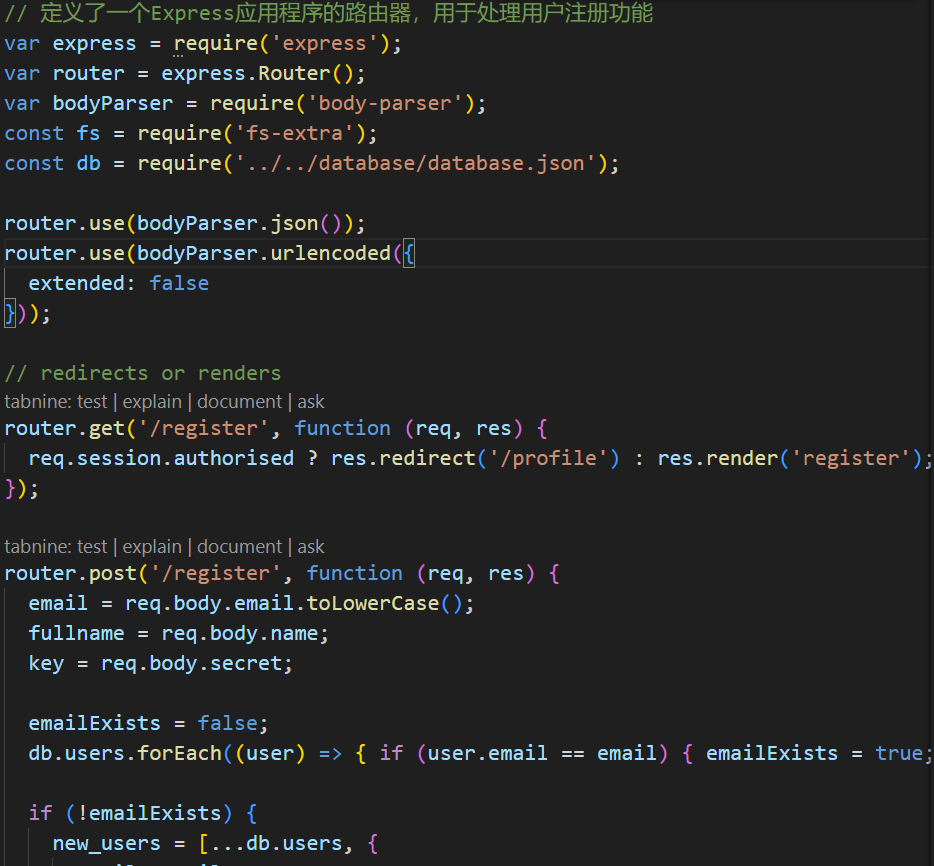
1. 程序代码展示

1. 用户认证模块

功能：

1、注册和登录

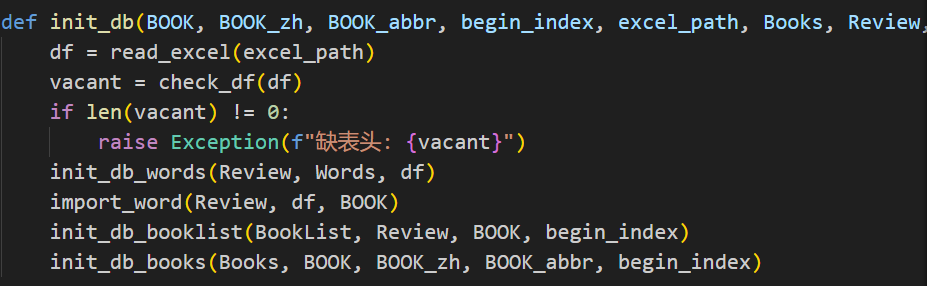
对于任何需要个性化的学习工具来说，用户认证都是基础设施的一部分，确保用户可以访问他们的个人数据并保护用户数据不被未授权访问。



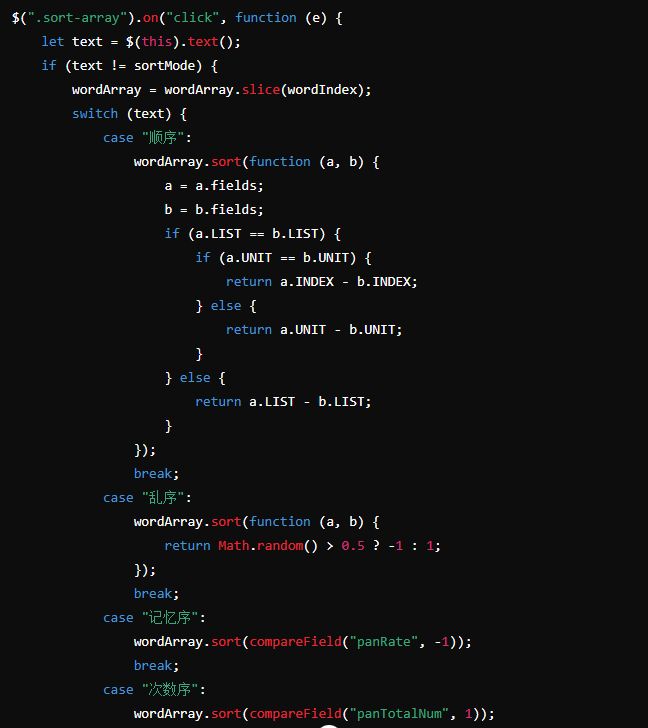
2. 单词管理模块

功能:

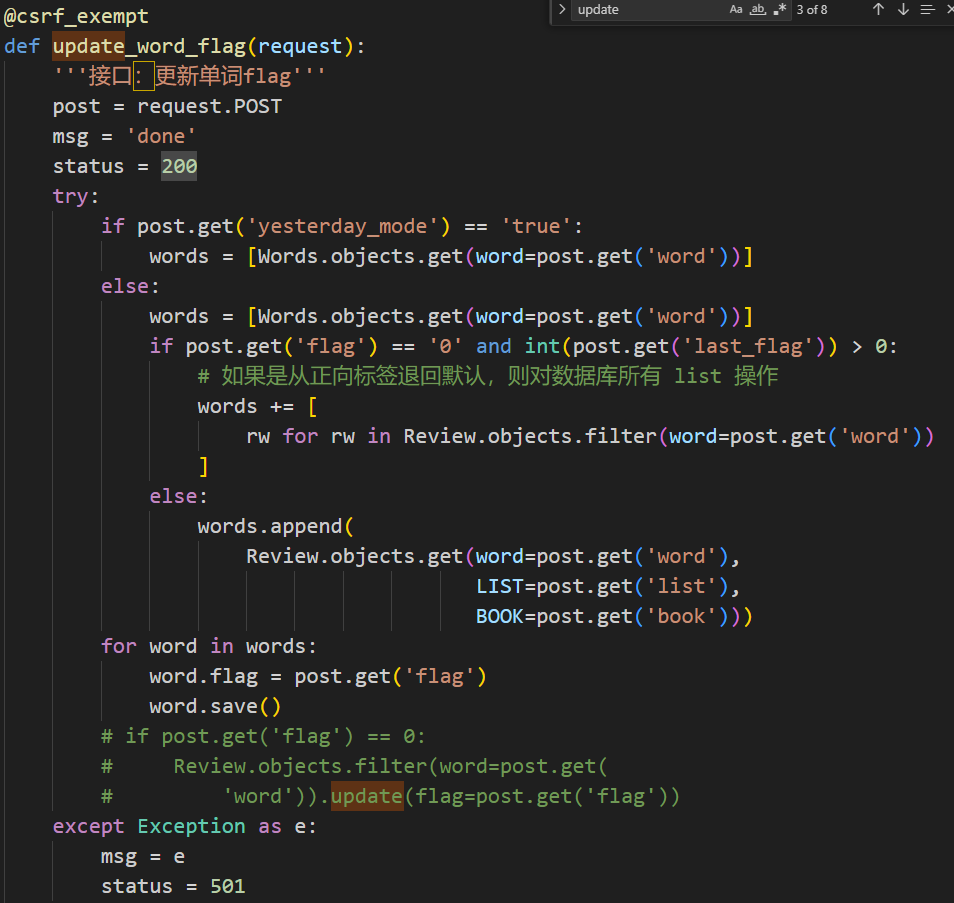
1. 单词数据库维护：增加、删除、编辑单词



1. 单词排序显示：顺序、乱序随机、按记忆率排序、按复习次数排序



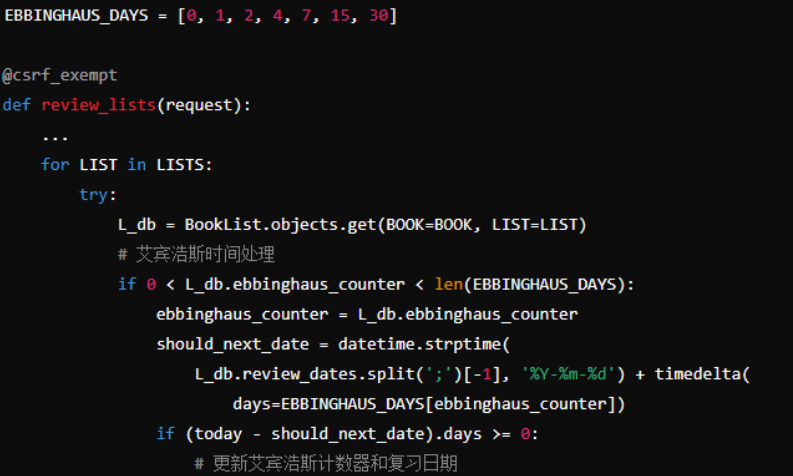
3、单词标记：标记为“太简单”，“已掌握”，“重难词”



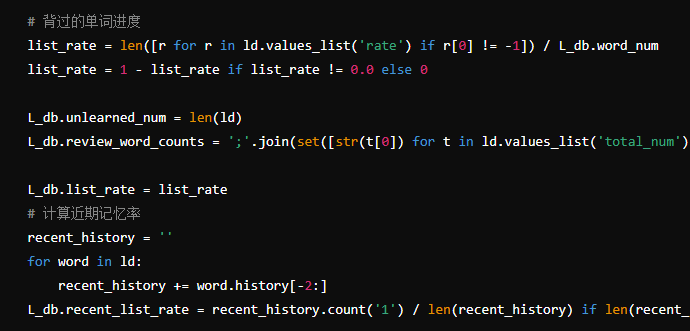
3. 学习跟踪模块

功能:

1. 艾宾浩斯记忆曲线的实现：根据记忆曲线调整复习频率



1. 进度条显示：展示“记得”与“遗忘”的比例



rate(list\_rate变量)计算逻辑:1d'查询获取了标记为未完全记住(flag <1)的单词。使用values 1ist('rate')方法获取所有单词的记忆率，其中’-1’表示未评估的单词，这些被排除在计算之外。计算已学习单词中记忆率非’-1’的比例，用总单词数除以这个比例，然后从、1减去得到的结果，得到整体复习的百分比。

recent rate(recent list rate变量)计算逻辑:遍历'1d”集合，累加每个单词的历史记录中最后两次的记忆结果(这假设history'字段存储的是记忆成功('1')或失败('0')的字符串)。计算近期复习中记忆成功的次数除以近期复习次数的比例。这两个百分比反映了用户对特定复习列表中单词的整体和近期记忆掌握程度，有助于用户了解他们的学习效果，并据此调整复习策略，

3、记忆历史查看：查看单词的历史记忆记录



4、笔记功能：为单词添加、编辑笔记

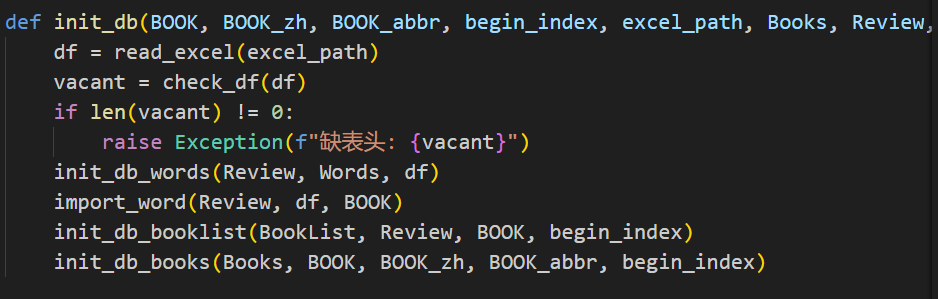


4. 导入模块

功能:

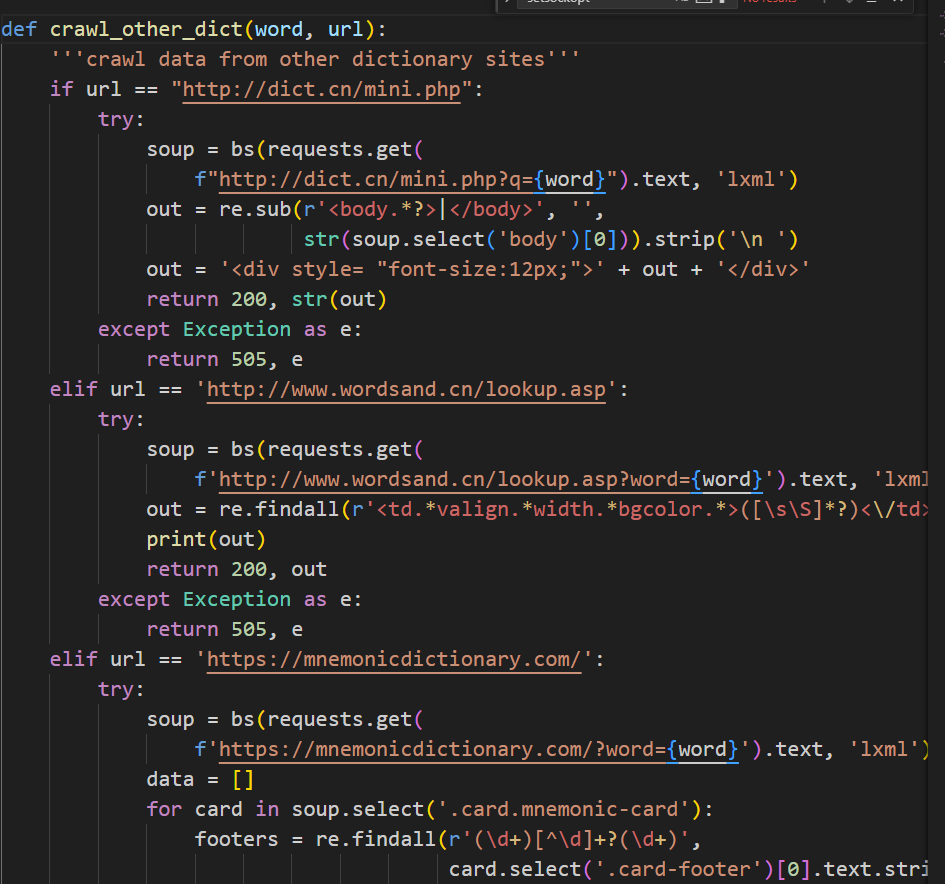
1、从web端导入单词书

该模块对应导入词书需求。用户能够轻松地扩展和自定义自己的词库，特别是教师或高级用户可以导入专业或定制的词汇表。



2、从网站上爬取对应单词的解释和助记法

根据不同的 URL，从外部字典网站抓取单词的解释、助记法等信息。



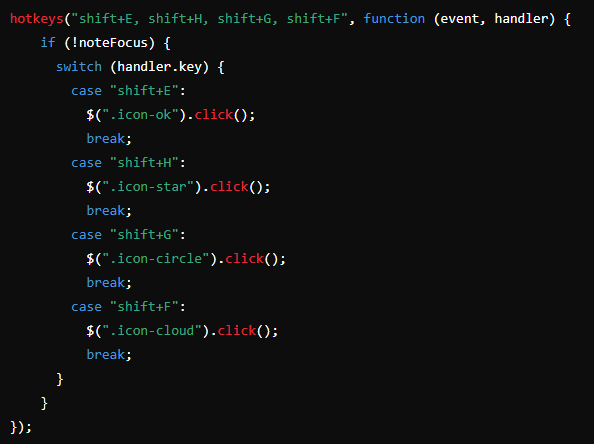
5. 内容呈现模块

功能:

1、界面样式设计和页面逻辑切换

2、显示例句和关键词高亮

3、实现快捷键操作



## 3.2主要功能及性能

主要功能：

用户认证：注册、登录、权限管理。

单词管理：增加、删除、编辑单词条目，支持多种排序和标记方式。

学习跟踪：艾宾浩斯记忆曲线实现，进度条和记忆历史展示。

笔记功能：为单词添加个性化笔记。

导入模块：支持从Excel导入词书，从在线字典抓取单词信息。

性能指标：

系统响应时间：< 500ms。

并发用户数：支持至少1,00个用户同时在线使用。

数据库查询效率：平均查询时间< 100ms。

## 3.3所用工时

前端开发：总计25人/天。

后端开发：总计25人/天。

数据库设计与管理：总计10人/天。

测试与质量保证：总计15人/天。

文档编写：总计25人/天。

## 3.4进度

计划进度与实际进度对比

项目启动与团队分工：按计划完成。

需求收集与分析：提前1天完成。

需求规格说明文档、软件设计说明文档：按计划完成。

数据库设计与API设计：延迟2天，由于需求变更。

前端页面开发与后端服务开发：按计划完成。

集成测试与问题修复：延迟3天，由于发现的缺陷比预期多。

用户文档完成：按计划完成。

系统部署上线：按计划完成。

项目总结：按计划完成。

# 4．评价

## 4.1生产率评价

团队生产率的评估主要基于源程序行数、文档编写效率以及关键文档的质量。

团队表现出了较高的生产率，这是通过合理的工作分配、高效的工具使用以及清晰的项目管理实现的。这样的生产率不仅确保了代码和文档的高质量输出，也有助于保持项目的按时交付。通过这次项目，我们也认识到生产率的高低直接影响到项目的整体成功与否。未来项目中，我们将继续优化我们的流程和方法，以进一步提高生产率。

## 4.2技术方案评价

优点

1. Django框架：

高效性：Django的ORM使得数据操作简单快捷，提高了开发效率。

安全性：提供了内建的安全功能，如CSRF和XSS防护，确保应用的安全。

可扩展性：Django的模块化架构支持快速扩展，适应日益增长的用户需求和功能迭代。

2. Pug框架：

灵活性：Pug是一种高性能的模板引擎，使得前端页面布局和结构设计更为简洁和高效，提供了丰富的交互体验。

广泛的支持：得益于广泛的社区和资源库，Pug模板可以快速解决开发中的问题，且易于与现代JavaScript框架和库集成。

3. 数据库使用SQLite和PostgreSQL：

简单性：SQLite易于配置和使用，适合早期的原型开发。

缺点

1. Django框架：

性能问题：对于非常高的并发访问，Django的性能可能不如一些基于异步的框架。

学习曲线：虽然Django文档齐全，但初学者可能需要时间来掌握其全面的功能。

2. 前端技术：

兼容性问题：需要额外的工作来确保应用在所有浏览器上都能一致运行。

性能开销：复杂的前端可能导致页面加载和执行速度慢。

# 5．经验与教训

1. 详细的需求分析是关键：

在项目初期进行彻底的需求分析可以避免后期的需求变更，减少开发时间和成本。

2. 持续的技术评估：

随着项目的推进，持续评估使用的技术和工具的有效性是必要的，以确保它们仍然符合项目目标。

3. 代码质量和文档：

维护高标准的代码质量和完善的开发文档对于项目的长期维护至关重要。

4. 团队协作和沟通：

加强团队之间的沟通和协作，确保所有成员都对项目的状态和目标有清晰的了解。

5. 测试和质量保证：

实施全面的测试策略，确保软件在发布前达到预期的质量标准，减少用户反馈的问题。

6. 应对变更的灵活性：

能够灵活应对需求和技术变更是项目成功的关键，需要有预设的计划来管理这些变更。

通过这些经验和教训，我们更加深刻地理解了软件开发过程中的各种挑战，这将为未来类似项目的成功奠定坚实的基础。