

37014328文档

1. 使用方式

1.1 安装和启动

- 1. 解压缩源码文件：下载并解压缩源码文件。
- 2. 进入后端文件：

```
cd backend
conda env create -f environment.yml // 执行环境安装
activate webCourse // 测试Python环境安装是否成功
```

- 3. 进入前端文件：

```
cd ../frontend
pnpm install // 安装所有依赖
pnpm dev // 启动Vue
```

- 4. 启动Flask服务器：

```
cd ../backend
python app.py // 启动Flask服务器
```

1.2 源码地址

源码地址：github.com/Zhong0118/RichEditor

1.3 数据库安装

数据库使用的是MongoDB，这是一个非关系型数据库。首先需要安装好MongoDB数据库，并且安装对应的mongosh终端工具。

1.4 部署网站

部署网站IP：[110.42.192.232/](#)

1.5 用户用例

- 1. 用户名：user1 密码：123456 邮箱：1@1.com
- 2. 用户名：user2 密码：123456 邮箱：2@2.com

2. 产品说明

在数字化时代，用户在日常学习和工作中常常需要编辑和处理大量的多模态信息。我们的富文本编辑器结合生成式AI能力，能够处理图片、文本、表格、代码甚至音频，并辅助用户进行内容创作，极大地提升学习效率 and 用户体验。

2.1 AI智能交互功能

此产品以富文本编辑器为载体，加装了许多AI智能交互功能，如：

- 选中润色、续写、翻译、总结
- AI问答
- OCR识别、语音转文字识别
- 一键排版文本
- 自定义上传本地知识库
- 自定义表格生成
- 思维导图内容提炼
- UML生成

对于AI功能，都有prompt的设置，可以事先为AI设定前置条件。

2.2 用户体验

- **伪支付条件**：提供二维码扫描，模拟VIP用户，提供更多功能。
- **共享文档**：利用每个文档独立的share_id实现多人共同编辑的逻辑。
- **模板文档**：后端默认生成数个特定情境下的模板文档，如日记模板、会议模板、计划书模板。
- **自定义模板**：用户可以自己自定义模板并导出，建议使用一键排版功能之后再进行模板设置。
- **markdown**：编辑器文本可使用markdown快捷方式，并且AI返回的文本是基于markdown语法的，编辑器可一键复制
- **明暗切换**：用户可自行切换明亮或暗夜模式
- **热力图**：用户可明确知道自己创建笔记的频率

2.3 编辑器功能

为了增加用户体验，本项目对于文档列表的虚拟化加载，根据多标签进行排序，搜索筛查，自动定位做出了很多配置，同时保证了大纲效果。针对编辑器本身的开发实现了富文本编辑器该有的所有功能，包括字体的改变，字体颜色背景颜色的设置，表格设置，并且配置了markdown的快速符号以及编辑器的快捷键，从而帮助用户使用项目。针对文本结果还实现了一键导出pdf文件，从而方便使用自己的文本内容。针对一些特定的文档类型，项目给出公共的模板库帮助快速构造文本内容，同时每个用户也可以独立保存自己设计的模板从而方便使用。每个用户都可以自定义共享文档，他人可使用共享码协同编辑对应的文档。用户可获得想要的mermaid代码从而可自己设计各种不同的UML图，比如甘特图、用例图、流程图等。用户可提炼文档大纲，从而形成思维导图。

2.4 响应式设计

项目实现了部分响应式设计以及明暗切换功能，设置大量优美的设计以及动画，保证用户有良好的体验。

2.5 未来目标

- 实现AI知识库的上传，用户可以通过上传文件来设置AI的背景内容。
- 实现笔记之间的关联性，利用双链多链逻辑来有机结合多个文档内容，从而打造用户自己的知识体系。
- 已完成

3. 功能架构

3.1 后端架构

- **技术栈**：Python Flask框架
- **数据库**：MongoDB（非关系型数据库）

- **交互**: 使用axios进行前后端交互, 解决CORS跨域问题

3.2 前端实现

- **UI框架**: Tailwind CSS和Element Plus
 - **前端框架**: Vue.js
 - **响应式设计**: 实现多端响应式设计, 支持明暗模式切换
1. 对于AI模型会提前进入运行, 有一个全局状态从而保证整体系统的性能; 把路由与具体数据库操作区分开, 从而保证项目文件的内聚性; 针对用户的密码实行hash加密从而保证安全性。
 2. 用MongoDB存储数据来保证在数据频繁更改的条件下维持高性能, 将文本内容以json的格式存储在数据库中, 减少数据体积

3.3 AI模型集成 (星河社区服务)

- **大模型**: 基于文心ERNIE SDK, 提供智能润色、续写、翻译、总结等功能
- **小模型**: PP-ASR 语音识别相关, PP-OCRv4 OCR识别

4. 产品创新点和优势

- **技术优势**: 结合大模型与小模型, 提供高效的AI辅助功能
 - **AI大模型与小模型的融合**: 通过结合大模型的深度理解和小模型的快速响应, 我们的产品能够在各种复杂场景下提供高效的AI辅助功能。
 - **高性能处理能力**: 产品在后端采用了优化的Flask框架和MongoDB数据库, 确保了数据处理的高效率和稳定性。同时, 前端的Vue.js和Tailwind CSS框架使得用户界面响应迅速, 提供了流畅的用户体验。
- **用户体验**:
 - **简洁直观的界面设计**: 我们的产品界面设计注重简洁性和直观性, 用户可以轻松上手, 快速找到所需的功能。界面布局合理, 操作逻辑清晰, 减少了用户的学习成本。保证用户体验。
 - **高效的交互逻辑**: 产品在交互设计上注重用户的操作习惯, 通过智能提示和快捷操作, 提升了用户的操作效率。例如, 通过一键排版和模板应用, 用户可以快速生成符合需求的文档。有很多弹窗设计, 保证了在所有情况下用户都能知道发生了什么事情。

产品运营和推广策略

- **市场定位**: 目标用户为需要高效文档编辑和AI辅助的专业人士
- **推广渠道**: 通过社交媒体、广告和合作伙伴推广, 以及同学之间的使用
- **运营策略**: 提供持续的技术支持和更新, 维护用户关系, 听取建议, 不断优化项目, 让项目真正的可以落地

5. 技术流程

5.1 后端实现

- **服务稳定性**: 通过Flask框架实现高并发处理, 确保服务的稳定性和响应速度。
- **数据存储**: 使用MongoDB存储用户数据, 保证数据的安全性和可扩展性。通过优化数据库查询和索引, 提升数据处理的效率。

5.2 前端实现

- **界面设计**: 使用Tailwind CSS和Element Plus实现美观的界面设计, 提供用户友好的视觉体验。

- **交互逻辑**: 通过Vue.js实现流畅的交互体验, 确保用户操作的连贯性和响应性。

5.3 数据处理

- **数据安全**: 采用加密技术保护用户数据不被泄露, 确保用户信息的安全。
- **数据存储**: 使用MongoDB进行数据存储, 确保数据的一致性和可靠性。通过合理的数据结构设计, 优化数据的存储和检索效率。

5.4

所有功能的实现以及布局以及策略都是基于最大程度上保证用户的体验, 让用户真正感受到便捷