

# AI实时协同文档库

---

## 1. 项目概述

---

- **项目的背景:** 对于大型的文档，很多时候往往不是完全由一个人完成，而是需要多个人的协作。一个实时同步文档库在这时会发挥出巨大的作用，能够实时显示别人的修改并进行自己的修改。
- **项目的核心目标:**
  1. 完成.txt, .md格式文档的同步编写以及渲染
  2. 接入AI大模型功能，能够自动获得当前文章的上下文，对用户的回答进行分析。(提供不同的大模型接入服务，用户能够配置不同的大模型进行回答)
  3. 完成.docx, .pptx等格式文档的同步编写
- **项目的关键问题:**
  1. **线程方面:**
    - 解决对同一个文件的读写的线程危险问题
    - 解决大量用户发起请求时的服务器负载功能，支持高并发功能(以1w~10w的同时连接为目标)
  2. **大模型接入方面:**
    - 提供多个现成的大模型接口，(要求用户提供API Key / 服务器自动提供) 待定
    - 实现大模型能够读取上下文的引用(考虑直接整篇读取可能会导致Token过大)
    - 实现Chat模式和Agent模式的切换 待定
  3. **文档内容方面:**
    - 对用户的文档大小进行限制——减少对服务器内存和流量的占用
    - 支持用户上传文档功能——对用户上传的文档大小进行检测
  4. **用户账号方面:**
    - 数据库中拒绝对密码的明文存储
    - 预留用户的支付接口，支持微信/支付宝支付功能等
- **项目地址:** <https://github.com/WingWR/SyncNote>

## 2. 项目技术栈

---

### 前端:

- **框架:** Vue3 + Vite + TypeScript
- **状态管理:** Pinia
- **网络通信:** Axios
- **实时协同文档:**
  - **TipTap**——Y.js实现的富文本编辑器，多人支持良好
  - **Y.js**——实现协同功能开发
  - **Y-websocket**——Y.js的WebSocket连接器, 负责实时同步文档内容

## 后端:

- 平台: JakartaEE
- 多LLM支持组件: Langchain4j
- 持久化层: MyBatis
- 消息队列: RabbitMQ

## 数据库:

- 主数据库: MySQL
- 缓存: Redis

## 3. 页面设计

---

### 编辑文稿页面

- 左侧边栏:

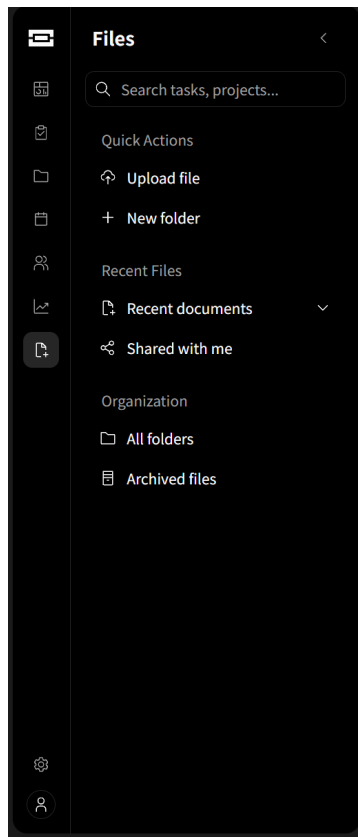
#### 上方:

- 回到主页:

点击后马上回到主页面，右边会显示一个欢迎界面，介绍软件的信息
- 文档管理操作:
  - 加入别人的共享文件
  - 新建文件
  - 导入文件
- 工作区界面:
  - 显示用户所有的文件，按照已开放权限以及没有开放权限来分类
- AI操控窗口:
  - 最上方显示AI和用户的聊天记录
  - 下方为交互区域，分为三个功能块
    - 选择使用的大模型
    - 选择使用的模式(Chat / Agent)
    - 信息输入框(圆角矩形)，信息发送按钮(圆形)

#### 底部:

- 系统设置
  - 显示当前用户信息
  - 支持修改用户信息
  - 退出登录



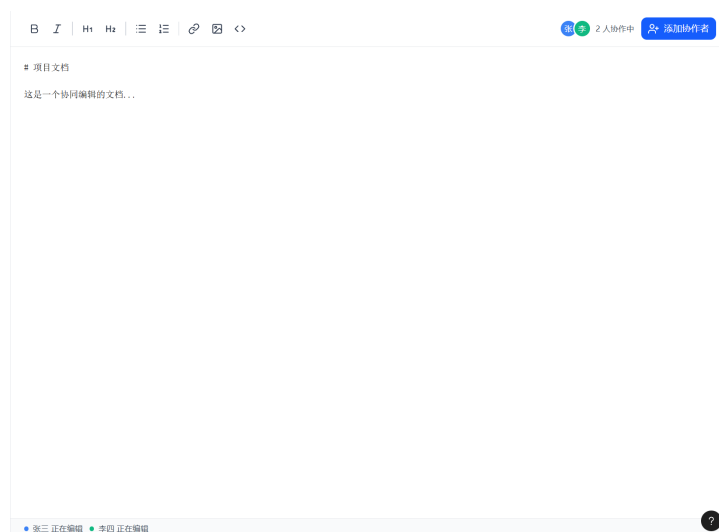
- **右侧界面:**

右侧根据当前的路由显示信息，如具体的文稿或者主页等等

**下方演示为.txt**

**上部分:**

- 工具栏以及显示协作人数(过多时只显示前几个)
- 添加协作者按钮



添加协作者

项目文档.md

✕

邮箱地址

✉ 输入协作者邮箱

权限设置

可编辑

发送邀请

或

分享链接

<https://collaborate.app/invite/abc123xyz>

复制

任何拥有此链接的人都可以编辑此文档

## 4. 表设计

### user表

字段名	类型	约束	说明
id	BIGINT	PK, AUTO_INCREMENT	用户ID, 用雪花算法生成
username	VARCHAR(50)	UNIQUE, NOT NULL	用户名
email	VARCHAR(100)	UNIQUE, NOT NULL	邮箱
avator	VARCHAR(100)	NOT NULL	用户的头像
password_hash	VARCHAR(255)	NOT NULL	哈希后的密码
salt	VARCHAR(100)	NOT NULL	密码盐（可选）
created_at	DATETIME	NOT NULL	创建时间

### document表

字段名	类型	约束	说明
id	BIGINT	PK, AUTO_INCREMENT	文档ID, 用雪花算法生成
owner_id	BIGINT	FK(user.id)	文档所有者
name	VARCHAR(255)	NOT NULL	文档名称
type	VARCHAR(20)	NOT NULL	文档类型 txt/md/docx/pptx
size_bytes	BIGINT	NOT NULL	文档大小（字节）
is_deleted	TINYINT(1)	DEFAULT 0	软删除标志
created_at	DATETIME	NOT NULL	创建时间
updated_at	DATETIME	NOT NULL	最后更新时间

## document\_chunk表

字段名	类型	约束	说明
id	BIGINT	PK, AUTO_INCREMENT	Chunk ID, 用雪花算法生成
document_id	BIGINT	FK(document.id)	文档ID
chunk_index	INT	NOT NULL	Chunk序号（升序）
content	MEDIUMTEXT	NOT NULL	分片内容
updated_at	DATETIME	NOT NULL	更新时间

## document\_collaborator表

字段名	类型	约束	说明
id	BIGINT	PK, AUTO_INCREMENT	主键, 用雪花算法生成
document_id	BIGINT	FK(document.id)	文档ID
user_id	BIGINT	FK(user.id)	用户ID
permission	VARCHAR(20)	NOT NULL	read / write
joined_at	DATETIME	NOT NULL	加入时间

## system\_setting表

字段名	类型	约束	说明
user_id	BIGINT	PK, AUTO_INCREMENT	用户ID
key_name	VARCHAR(100)	UNIQUE, NOT NULL	配置Key
key_value	TEXT	NULL	配置Value（加密存储）
updated_at	DATETIME	NOT NULL	更新时间

## 5. 预计使用要求技术

容器化（使用 Docker 和 Docker Compose 等工具）：

CI/CD 流水线实现

基于消息队列的事件驱动机制设计

全面的软件测试（例如单元测试、模拟测试等）

性能分析与优化

