AI标书生成系统项目任务书

项目周期：4月15日 – 6月15日（开发完成）

一、目标

构建一套智能标书生成系统，支持上传PDF/Word格式标书文档，通过OCR提取结构与语言风格，用模型生成包括简易结果、意见书及整本标书（含100页+）等文档。前后端部署，开发Web端与微信小程序（app？），数据库支持用户文档管理。

二、核心模块设计

1. 用户系统模块  
- 用户注册 （费用 cloudflore租用？百度网安租用？）登录（手机号+验证码（接口费用））  
- 权限管理（普通用户 / 管理员）（数据库租用（费用））  
- 用户使用配额、上传记录跟踪  
- 支持微信登录（小程序端）  
技术栈：Vue + uni-app + FastAPI（费用） + PostgreSQL

2. 文档上传与解析模块  
- 支持上传 .pdf / .docx  
- 提取正文内容、段落结构、表格、标题层级  
- 文档初步分类识别  
技术栈：PyMuPDF, python-docx, OSS存储或MinIO （费用）

# 3. OCR识别与结构提取模块  
- 针对扫描件或图像型PDF进行结构化提取  
- 支持表格、段落、标题、页眉页脚识别  
技术栈：PaddleOCR（费用） + 后处理 + MongoDB结构中间件

# 4. （投标文件后考虑）风格分析与Prompt构建模块（可能导致生成内容单一）  
- 分析用户历史文档的风格特征，聚类为风格向量  
- 构造Prompt模板支持风格一致生成  
技术栈：BERT（待定）, Faiss, LangChain（是否需要对话式生成）

1. 大语言模型微调模块（租用大模型（费用），未完成api接口替代（费用））  
   - 部署自有大模型（文心/ deepseek）  
   - LoRA微调任务模型支持填空/全生成两类文档（决定单一模型多模态or多个模型）  
   技术栈：LoRA, PEFT, vLLM

模型聊天使用场景

6. Word文档生成模块  
- 支持填空生成、自由结构生成、风格一致性  
- 支持目录、图表、页眉页脚、封面、导出.docx  
技术栈：python-docx-template + Jinja2

7. 数据库与用户标书记录模块  
- 存储用户文档、风格向量、生成记录等  
- 支持至少保存一年历史、用户反馈跟踪  
数据库：PostgreSQL + MongoDB

8. Web前端 + 小程序开发模块  
- 页面：注册/上传/生成/预览（费用）/历史/模板选择（？）等  
- 响应式设计，支持PC与微信小程序使用（技术手段待定）  
技术栈：Vue 3 + uni-app + Element Plus + Axios

# 9. 后台管理模块  
- 用户/文档/生成记录管理（租用网盘（费用））  
- 日志、模型状态监控、模板配置  
技术栈：FastAPI Admin 或 Vue Admin Template

三、时间表

第一阶段：需求 + 基础模块（4.15 – 4.30）  
4.15-4.16：模块分工  
4.16-4.25：上传与解析  
4.26-4.30：OCR

第二阶段：模型部署 + 生成系统（5.1 – 5.20）  
5.1-5.5：风格提取（多个？）、Prompt构建开发  
5.6-5.10：模型部署（多个？） + 简易文档生成功能  
5.11-5.15：填空式与自由式生成模块完成（合并or集成）  
5.16-5.20：Word结构与模版生成系统（多个？）开发

第三阶段：前后端整合 + 完整测试（5.21 – 6.15）  
5.21-5.28：Web与小程序端整合联调  
5.29-6.5：测试（单元测试 之前完成 ） + Bug修复  
6.6-6.10：UI优化、样式完善  
6.11-6.15：内部验收，准备上线包

四、后续 非开发  
6.16–6.30：采集反馈  
7.1–7.31：Prompt增强  
8.1–8.31：补丁优化  
9.1–10.15：合同/协议文档支持

五、交付内容

- 任务书  
- 系统源代码（前后端 + 模型微调）  
- 完整部署脚本 + 模型运行文档  
- 用户手册

需确定：

是否需要对话式生成

需要多用户同步使用

费用相关确定：

模型数量（服务器大小）

需微调模型数量 取决于模型多模态需求（应对不同风格任务）

模型参数量（决定效果）

服务器GPU

大小（决定训练速度）

网盘空间（用户存储大小）

文心一言X1模型 token数（模型开发未完成前）