体验实习课题

1. 课题介绍

课题名称

产品说明书问答服务系统

背景说明

本课题要求以“后台软件开发”的视角，设计并实现一个面向浏览器的智能问答系统。用户上传 PDF/Markdown 产品说明书后，可针对该文档内容进行自然语言提问，系统需流式返回精准答案。

重点说明

该课题重点不在于前端页面美观，而在于 **AI 工程化（AI Engineering）** 能力：

- **后台接口设计与服务抽象**：如何优雅地集成不稳定的 LLM 服务。
- **长文本与上下文处理**：在大模型 Token 限制下的工程解决方案。
- **流式传输（Streaming）**：HTTP 分块传输或 SSE 的实现。
- **系统的可扩展性**：支持不同模型厂商或不同处理策略的切换。

---

2. 基础要求

功能目标

在浏览器中打开一个页面，实现以下能力：

1. 文档解析与管理 （可离线）

- 页面支持上传一份产品说明书（格式支持 PDF 或 Markdown）。
- 后台接收文件后，提取文本内容并进行必要的清洗/存储。
- 你可以只提取纯文本，**忽略图片、表格和复杂排版**。只要能拿到文字内容即可。

2. 基于文档的问答

- 用户输入与产品相关的问题。
- 系统仅依据上传的说明书内容进行回答（约束 AI 不许胡编乱造）。
- **关键要求**：若文档内容较多，需考虑如何构建 Prompt 以适配模型的上下文窗口（Context Window）。

3. 后台服务能力

- 提供后台服务（推荐 Java/Spring Boot 或 Python/FastAPI，语言不限）。
- **模型服务集成**：对接至少一种大模型 API (GPT/Gemini)
- **流式响应**：鉴于大模型生成速度较慢，后台接口必须支持**流式输出（Stream/SSE）**，实现前端“打字机”效果，而非等待全部生成完再返回。

4. 基本页面交互

- 文件上传区域与状态展示（解析中/就绪）。
- 对话框：发送问题，实时展示 AI 回复。

---

3. 进阶要求

在完成基础要求的前提下，可进一步实现以下能力：

1. 处理策略与模型抽象

- 支持通过配置切换不同的底层实现，且无需改动核心逻辑：
- **策略 A（Long Context）**：直接将全文放入 Prompt 发送给支持长文本的模型
- **策略 B（RAG 简易版）**：如果文本过长，先进行简单的关键词截取或分段，再发送给模型。
- 或者支持**多模型切换**：
- 在界面上选择“GPT-4.1（快速/低成本）”或“GPT-5（精准/高成本）”，后台动态路由。

2. 多轮对话上下文

- 支持“追问”场景。
- *示例*：
- 用户：“这款相机的电池容量是多少？” -> AI：“2000mAh。”
- 用户：“那充满电需要多久？” -> AI 需理解“那”指的是上述电池，并给出答案。
- 后台需维护会话的历史记录（History），并合理控制发送给模型的历史长度（防止 Token 溢出）。

3. 引用溯源

- 在返回答案的同时，告知用户答案出自文档的哪一部分。

- *形式*：返回答案文本 + 来源标记（如：“参考自第 3 页”或返回原文片段）。

4. 工程化稳定性与异常处理

- **网络与 API 异常**：
  - 模型 API 超时或由限流（Rate Limit）导致的 429 错误。
  - 后台需实现重试机制（Retry）或降级处理。
- **防幻觉机制**