LLM + RAG 知識檢索服務

【說明】

- 請使用附件提供的資料建立一個地端可執行的簡易知識檢索服務，能接受自然語言問題並回覆答案與來源段落。整體流程需包含（不限於）：資料前處理 → 分塊（Chunk）→ 向量化（Embedding）→ 向量資料庫檢索 → 本地 LLM 產生回答 → 回傳答案與來源。

【必要要求】

- 只能使用地端模型：
  - 服務啟動後需完全可離線執行，不得呼叫任何雲端 LLM、Embedding API、向量庫託管服務。
  - 禁止使用：OpenAI、Anthropic、Cohere、AWS Bedrock 等雲端 API。
  - 可參考地端模型選項：Ollama、Hugging Face離線模型、llama.cpp（GGUF）、SentenceTransformers（本地 Embedding）等。
- 向量資料庫使用：
  - 可使用：FAISS（檔案型索引）、Qdrant（本地執行）、Milvus（本地執行）。
  - 請於說明文件中標示：
    - 向量維度與型態（是否正規化、float32/float16）。
    - 距離度量（cosine/L2/IP）。
    - 索引類型與參數（如HNSW, IVF等）。
- 檔案清洗 & Chunk & Embedding：
  - 請針對原始資料進行適當清洗與斷句處理。
  - 需說明Chunk切分策略（長度、重疊、標題維護等）。
  - 使用本地模型產生Embedding，並寫入向量資料庫。
- 檢索 + 本地 LLM 回答：
  - 可透過CLI或API提問。
  - 生成回答時應以檢索段落為唯一依據進行回覆。
    - 若有回覆時，需呈現檢索到的Chunk內文。
    - 若無檢索到足夠資訊，請回覆：「無法回覆該問題」，避免產生無根據的回答。

【交付內容】

- 建置方式/流程說明（可DOC/PDF/投影片等各式檔案）：請簡要說明整體設計邏輯與處理流程，包含以下面向：
  - 資料處理流程：清洗方式、Chunk切分策略、向量維度與模型選擇、向量庫設計邏輯與參數。
  - 問答流程設計：檢索方式（Retriever）、是否採用Re-ranker、Prompt 組裝策略、Token限制與截斷設定。

    - 模型與設計選擇原因:像是為何選擇某模型或索引方法？資源限制考量？效果最佳化？
- 程式碼涵蓋以下區塊:
    - 資料清洗。
    - Chunk切分。
    - 向量產生與向量庫寫入。
    - 問題接收與檢索輸出。

【交付時程】

- 請於信件所提供之繳交期限提交【交付內容】。
- 提交方式:回覆信件附上或雲端連結(開啟讀取權限)。