ChatNCHU 開發簡介

ChatNCHU 是一個整合爬蟲、文件解析與 RAG
架構的校務規章與公告檢索問答平台，讓使用者以自然語言快速查找分散於各處室網站的公開資訊。
專案動機源於「資訊分散、版本混亂、缺乏引導」三大痛點，期望以自動化蒐整與語意檢索提升查詢效率與正確性。

目錄

- 平台目標
- 核心痛點
- 系統架構
- 技術棧
- 資料流程
- RAG 檢索流程
- 功能摘要
- 專案角色
- 名稱與範疇
- 開發與部署
- 目前進度與成果展示

平台目標

- 透過定期爬取公告、簡章與
  PDF，將非結構化內容轉為可檢索的結構化資料，解決資料來源分散與年度版本不明的問題。

- 結合向量資料庫與本地
  LLM，提供具來源脈絡的自然語言回答，提升決策信心並降低行政溝通成本。

核心痛點

- 資訊分散：規章與公告散落各系所與處室網站，人工搜尋成本高且容易遺漏。
- 文件紊亂：規章常有年度版本差異，難以確認是否為最新或適用版本。
- 缺乏指引：條文式資訊未結合情境，缺少一體化回覆與可驗證來源脈絡。

系統架構

資料層

- 定期爬取 HTML 與 PDF，解析、清理、切分並結構化為 JSON/CSV。

檢索層

- 以 Sentence Transformers 產生向量並寫入 Qdrant，提供相似度檢索（Cosine/Euclidean）。

生成層

- 以 LangChain 組合系統指令、使用者查詢與檢索片段，交由 Llama 3（vLLM 部署）生成回答。

應用層

- 前端以 Vue 3 + Vue Router 串接後端 API，提供查詢與可視化介面。

技術棧

- 前端：Vue 3、Vue Router、Axios；UI/UX 使用 Figma 原型製作。
- 後端：Java、Spring Boot、Maven；整合 Google OAuth 2.0；MySQL 儲存登入資料。
- 資料工程：Python、requests/httpx、BeautifulSoup/lxml、Selenium、pdfplumber、pandas、re；輸出 JSON/CSV。
- 向量與 LLM：Sentence Transformers、Qdrant、LangChain、Llama 3（vLLM 部署）。
- DevOps：Linux Ubuntu、GitHub Actions（CI/CD）、Docker、Kubernetes（部署與擴展）。

資料流程

- Web Scraping：以 requests + BeautifulSoup 取得校內公告與規章之 HTML/PDF。
- PDF Parsing：以 pdfplumber 抽取文字並搭配正則處理雜訊與格式。
- Cleaning & Chunking：去除 HTML 標籤與無效內容，依語意斷塊以利後續檢索。
- Structuring：轉換為標準化 JSON/CSV，納入欄位定義與版本控管。
- 排程更新：使用 Windows Task Scheduler 或 APScheduler 進行定期 ETL。

RAG 檢索流程

- Query 分析：將自然語言查詢以 Embedding Model 轉換為向量並生成檢索語句。
- 向量檢索：計算 Query 與資料向量相似度，擷取 Top-k 相關片段（Cosine/Euclidean）。
- Prompt 組合：以 LangChain PromptTemplate 整合系統指令、使用者需求與知識片段。
- 回答生成：Llama 3（vLLM）根據提供的上下文生成可追溯的自然語言回覆。

功能摘要

- 關鍵字與情境式問答：支援以自然語言查詢入學與行政規章資訊並回傳脈絡來源片段。
- 資料來源更新：定期爬取各單位最新公告，降低版本不一致風險。
- 身分驗證：支援 Google OAuth 2.0 登入與 Token 驗證流程（後端）。

專案角色

- PM：掌握進度、PRD 與文件中心（Jira、Notion、甘特圖、GitHub）（陳鉦元）。
- UI/Frontend：Figma 設計、Vue 3 開發、路由與 API 串接（洪慧珊）。
- Backend/DB：Java + Spring Boot、MySQL、Google Auth 整合（陳鉦元）。
- Data Engineer：爬蟲、解析、清理、結構化與排程（許巧琳）。
- LLM Engineer：Embedding、RAG、Prompt 建構與模型部署（張允）。
- DevOps：CI/CD、容器化與集群部署、伺服器維運（陳鉦元）。

名稱與範疇

- 平台名稱：ChatNCHU（校內公開資訊之檢索與問答）。
- 適用範圍：系所與處室公告辦法、規章條文、申請流程與相關公開文件。

開發與部署

- 版本控管：Git/GitHub 或 GitLab 管理程式與資料流程版本。
- 自動化：GitHub Actions 進行測試與部署，Docker/K8s 提供可移植與可擴充環境。
- 執行環境：Linux Ubuntu 作為開發、測試與部署的主要作業系統。

目前進度與成果展示

- 已完成需求定義、UI/UX 原型、核心技術選型與系統流程設計草圖（含資料與 RAG 流程圖）。
- 持續整合前後端與向量檢索，並完善自動化 ETL 與模型部署流程以支援穩定上線。