主導課程：生成式AI應用系統與工程(Generative AI Application Systems and Engineering)

課程基本資料

開設學校：國立成功大學

開授教師：莊坤達

班級人數：聯盟校每校上限100 人

開課級別：研究所 開放大學部選修

授課語言： 中文

授權方式：封閉式

同步遠距上課時間：每週三 14:00-17:00

是否接受非同步授課：是

遠距上課位置：YouTube

課程網頁：NTU Cool

聯盟學校修課人數與助教比例：每__25__名學生需__1__名助教

課程概述

本課程以「生成式 AI 應用系統」開發為核心，帶領學生從需求分析、系統設計，到實作與部署，完成一個生成式 AI Web服務。內容涵蓋軟體工程SDLC、前後端技術、資料流程與工作流、開源CI/CD與 MLOps/LLMOps工具、AWS 雲端環境、分散式 AI Infra、LLM 微調與 local LLM 評測、Agent 工作流與 MCP/ADK/agent-to-agent架構、LiteLLM/OpenRouter 等模型代理工具，以及 token 經濟學、Prompt 優化、多輪對話設計、幻覺減少與 LLM 資安議題。

學生將透過至少六次程式作業與期末專題，實作一個具完整工程思維的生成式AI應用系統

課程目標

修課學生預期能達成：

- 理解生成式 AI 應用系統的整體架構，包括前端、後端、LLM 層、Agent 工作流與資料流。
- 熟悉SDLC在生成式AI專案中的實務流程，能撰寫系統需求與高階架構設計。能運用 Web架構、AWS平台、資料庫與向量檢索技術，可建構具RAG能力的應用系統。
- 了解並整合 MLOps/LLMOps toolchain，包括 CI/CD、模型部署、評估與觀測。
- 掌握 Agent workflow、MCP、ADK 與 agent-to-agent 等概念，並透過 callbacks 實作安全與審計邏輯。
- 理解 token 經濟學、多輪對話設計與幻覺減少策略，並融入系統設計中。
- 能設計與實作基本的 LLM 資安防護，包括 prompt injection 防禦與 response auditing。
- 完成一項可展示的生成式 AI 應用系統期末專題。

參考書目

- 講義、程式碼示例、AWS 教材
- MCP、LLMOps、Ray 等官方文件

課程內容大綱

週次	日期	課程內容	備註
1	2/23	課程介紹、修課要求與評分方式：	HW 1： 前端 streaming Chat/ 建議介面

		• 生成式AI應用系統典型架構：前端、後端、LLM 層、Agent workflow、資料流、系統觀測機制等介紹。	React/Next.js + SSE/WebSocket，實作 LLM 互動介面
2	3/2	SDLC、需求分析與系統架構設計概念： • SDLC 在生成式 AI 專案中的應用。 • 撰寫 System Requirement、Use Case • Microservice、高階架構圖規劃、服務邊界與模組切分	
3	3/9	前端 Web 技術與生成式介面設計： • Next.js / React / Tailwind • gen-AI UI patterns • SSE/WebSocket streaming	HW 2： Streaming LLM Chat UI 實作
4	3/16	後端架構、微服務設計與 LLM Proxy Gateway： • API Server：FastAPI / Node.js • REST / WebSocket / SSE 實作 • Microservice 架構核心 • GenAI 微服務拆分 • LLM Gateway核心微服務：Routing、Token Logging	
5	3/23	事件驅動資料流架構： • ETL / background tasks • Airflow DAG、Task、Scheduler • Kafka → Airflow → Iceberg 的典型資料處理流程	HW 3： Kafka ETL Pipeline 實作

		• Iceberg Data Lake • Microservice 與 Background Worker 整合	
6	3/30	CI/CD、MLOps、LLMOps： • GitHub Actions／GitLab CI • Docker化、環境建置 • MLflow/promptfoo • 模型行為監控、Regression Test • Microservice deployment pipeline	
7	4/6		清明連假
8	4/13	AWS 開發與部署、AWS Kiro介紹： • AWS EC2 / ECS / Lambda • IAM、S3、RDS • Kiro 架構 • 成本估算與auto scaling • CDN/Cache	
9	4/20	資料庫系統與向量資料庫： • Embedding、HNSW、Retrieval • RAG pipeline：chunking、index、rerank • Retrieval microservice	HW 4：RAG
10	4/27	模型微調、Local LLM、行為檢測： • Fine-tuning • vLLM、Ollama 本地部署 • promptfoo / eval harness 進行模型行為檢測	

		• Model Service as Microservice	
11	5/4	分散式 AI Infra 與工作流程： • Ray tasks、actors • Ray Serve + scaling • 多檔案、多任務平行LLM pipeline • Ray Worker microservice	HW 5：Ray-based 生成式工作流實作
12	5/11	Agent Workflow、MCP、ADK、Agent-to-Agent 協作： • Agent 架構：Planner、Tool、Critic • MCP：工具抽象層與資源管理 • ADK / Vertex AI Agent Builder • agent-to-agent workflow	
13	5/18	ADK Callbacks與Sanitize/Policy Check/ Audit Middleware： • ADK Agent生命周期與callback流程 • 使用ADK callbacks做安全防護與審計	HW 6：LLM Security Layer
14	5/25	Other LLM Security、Jailbreak、與 Response Auditing： • Prompt Injection • Jailbreak 防禦 • ADK-based Policy Engine • API key 保護與最小權限原則	
15	6/1	Observability： • LLM UX/error recovery • 延遲優化、Queue	

		• logs / metrics / distributed tracing • token 成本／效能指標 • Online A/B testing	
16	6/8	Project Demo	
17	6/15	(Optional) Supplementary Materials	
18	6/22	(Optional) Supplementary Materials	

成績評量方式

- HW 1-6：10% each HW
- Final Project：40%
    - Ranking based on 系統架構圖（含微服務拆分、Agent workflow）、服務流程圖與 API 設計、Demo Presentation and Slides、GitHub Source and Technical Report

對學生修課的課程要求

- 具備基本程式設計能力。
- 具備基本Web 技術（HTML/CSS/JS）與GitHub使用經驗。
- 對雲端服務有初步認識者佳（不為必要條件）。