สร้าง AI Chatbots On-Premise ด้วย LangChain และ LangGraph (Python) สำหรับองค์กร

ในยุคที่ Artificial Intelligence (AI) เข้ามามีบทบาทสำคัญในการขับเคลื่อนธุรกิจ หลายองค์กรเริ่มกังวลเรื่องความปลอดภัยและความเป็นส่วนตัวของข้อมูล (Data Privacy) เมื่อต้องส่งข้อมูลสำคัญไปประมวลผลบน Cloud API สาธารณะ หลักสูตรนี้จึงถูกออกแบบมาเพื่อแก้ปัญหานี้โดยเฉพาะ ด้วยการสอนสร้างระบบ "Local AI" ที่ทำงานได้ภายในเซิร์ฟเวอร์ขององค์กรเอง 100% ตั้งแต่ระบบสมองกล (Inference Engine) ไปจนถึงฐานข้อมูลความรู้ (Vector Database)

ผู้เรียนจะได้ลงมือปฏิบัติจริงในการพัฒนา AI Chatbot ที่มีความซับซ้อนระดับ Enterprise Grade โดยเปลี่ยนจากการเขียน Script ธรรมดา มาสู่การวางสถาปัตยกรรมแบบ Microservices ที่ยืดหยุ่นและขยายผลได้จริง เรียนรู้การใช้ vLLM เพื่อรันโมเดลภาษาขนาดใหญ่ให้ประมวลผลได้รวดเร็ว, การใช้ Qdrant จัดการความรู้เฉพาะทางขององค์กร (RAG), การใช้ LangGraph สร้าง Flow การตัดสินใจที่ชาญฉลาด แทนการเขียน Chain แบบเดิม และเชื่อมต่อระบบทั้งหมดผ่าน FastAPI และ Docker เพื่อให้พร้อมนำไปติดตั้งใช้งานจริง (Deployment) ได้ทันที

***** หมายเหตุ** หลักสูตรนี้เน้นการใช้งาน LangChain และ LangGraph ร่วมกับ Python เพื่อสร้าง AI Chatbot ที่ทำงานได้ภายในองค์กรอย่างปลอดภัยและมีประสิทธิภาพ โดยใช้ Ollama, vLLM และ Qdrant เป็นเทคโนโลยีหลัก ผู้เรียน ไม่จำเป็น ต้องมีคอมพิวเตอร์ที่มี GPU ก็สามารถเรียนรู้ได้ผ่าน Google Colab (ฟรี)

วัตถุประสงค์:

- มีความรู้ภาษา Python เบื้องต้น
- เข้าใจคอนเซปต์ของ Web API (REST) เบื้องต้น
- เคยใช้งานคำสั่ง Docker พื้นฐาน (pull, run, ps)
- คอมพิวเตอร์ส่วนตัวสำหรับใช้ในการอบรม (Windows, macOS, หรือ Linux)

กลุ่มเป้าหมาย:

- Software Developers / AI Engineers ที่ต้องการย้ายจาก OpenAI API มาทำระบบเอง (Self-hosted)
- Web Developers (Frontend/Full-stack) ที่ต้องการเพิ่มทักษะด้าน AI และ LLMs เข้าไปในสายงาน
- Data Scientists ที่ต้องการ Deploy โมเดลให้เป็น Application

- Solution Architects หรือ Tech Leads ที่ต้องการเข้าใจกระบวนการสร้าง AI Chatbot เพื่อนำไปประยุกต์ใช้กับระบบขององค์กร
- ผู้ที่สนใจสร้าง Product AI และต้องการเรียนรู้ Tech Stack ที่ทันสมัยและครบวงจร
- IT Infrastructure / DevOps ที่ต้องการเข้าใจการวางระบบ AI Server ในองค์กร

จุดเด่นของหลักสูตร

- เน้นสอนการใช้ Tech Stack ที่รันแบบ Offline ได้ทั้งหมด (Local LLM, Local Embeddings) ตัดความกังวลเรื่องข้อมูลรั่วไหล
- เปลี่ยนจาก LangChain Chains แบบเดิมที่ทำงานเป็นเส้นตรง มาสู่ LangGraph ที่รองรับการทำงานแบบ Loop และจดจำ Context ได้ดีกว่า (Stateful)
- สอนใช้ vLLM ซึ่งเป็น Inference Engine ที่เร็วที่สุดในปัจจุบัน และ FastAPI ที่เป็นมาตรฐานของ Python Backend
- เรียนรู้วิธีแยกส่วน Frontend (Chainlit), Backend (FastAPI) และ Database (Qdrant) ออกจากกัน เพื่อความง่ายในการดูแลรักษา
- ทุก Workshop รันบน Docker Environment จำลองสภาพแวดล้อมการทำงานจริงในระดับองค์กร

คอมพิวเตอร์และโปรแกรมที่รองรับการพัฒนา

- รองรับ Windows 10, 11
- รองรับ MacOS
- รองรับ Linux OS

ระยะเวลาในการอบรม:

- 18 ชั่วโมง (3 วัน)

ราคาคอร์สอบรม:

- ราคา 6,500 บาท / คน (ราคานี้ยังไม่ได้รวมภาษีมูลค่าเพิ่ม)

วิทยากรผู้สอน:

- อาจารย์สามิตร โกยม

เนื้อหาการอบรม:

1: ปูพื้นฐาน Local AI Infrastructure และ vLLM

- **On-Premise AI Overview:** ทำความเข้าใจความแตกต่างระหว่าง Cloud AI กับ Local AI และทำไมองค์กรถึงต้องการ Privacy
- **รู้จัก vLLM (Inference Engine):** เจาะลึกเครื่องมือรันโมเดลภาษาที่เร็วที่สุดในปัจจุบัน และสถาปัตยกรรม PagedAttention
- **Environment Setup with Docker:** การติดตั้ง Docker, NVIDIA Container Toolkit และเตรียม Python Virtual Environment
- **Running Local LLM:** เวิร์กชอปการรันโมเดล Open Source (เช่น Llama-3, Qwen, Typhoon) ผ่าน vLLM บนเครื่องตัวเอง
- **LangChain Integration:** การเขียน Python เพื่อเชื่อมต่อ vLLM ผ่าน OpenAI Compatible API

2: สร้างสมองความจำด้วย Vector Database และ RAG

- **RAG Architecture 101:** เข้าใจกระบวนการ Retrieval-Augmented Generation เพื่อลดอาการ Hallucination ของ AI
- **Vector Embeddings:** หลักการแปลงข้อความภาษาไทยเป็นตัวเลข (Vector) และการเลือก Embedding Model (HuggingFace)
- **Setting up Qdrant:** การติดตั้งและบริหารจัดการ Qdrant Vector Database ผ่าน Docker Compose
- **Data Ingestion Pipeline:** เขียน Script อ่านไฟล์เอกสารองค์กร (PDF/Text) ตัดคำ (Chunking) และบันทึกลงฐานข้อมูล
- **Similarity Search:** เทคนิคการเขียน Query เพื่อค้นหาข้อมูลที่เกี่ยวข้องจาก Qdrant มาใช้งาน

3: ยกระดับสู่ AI Agent ด้วย LangGraph

- **Why LangGraph?:** ข้อจำกัดของ LangChain แบบเดิม (Chains) และการเปลี่ยนผ่านสู่การทำงานแบบ Graph (Cyclic)
- **Core Concepts:** ทำความเข้าใจ Nodes (จุดทำงาน), Edges (เส้นเชื่อม), และ State (หน่วยความจำกลาง)
- **Building Graph Workflow:** เวิร์กชอปสร้าง Flow การทำงาน: รับคำถาม -> ค้นหาข้อมูล (Retrieve) -> สรุปคำตอบ (Generate)
- **Agent State Management:** การออกแบบโครงสร้างข้อมูลเพื่อเก็บ Chat History และ Context ระหว่างการสนทนา
- **Visualizing the Graph:** การ Compile และแสดงภาพ Diagram การทำงานของ Agent เพื่อ Debug Logic

4: พัฒนา Backend API มาตรฐานองค์กรด้วย FastAPI

- **Microservices Design:** การออกแบบโครงสร้างโปรเจกต์แบบแยกส่วน (Decoupled Architecture) เพื่อการขยายตัวในอนาคต
- **FastAPI Implementation:** การสร้าง RESTful API Endpoint เพื่อรับ-ส่งข้อความกับ LangGraph
- **Asynchronous Programming:** เทคนิคการเขียน Python แบบ Async/Await เพื่อรองรับผู้ใช้งานจำนวนมากพร้อมกัน
- **Streaming Response (SSE):** การทำ Server-Sent Events เพื่อส่งข้อความตอบกลับทีละตัวอักษร (Token Streaming) แบบ Real-time
- **API Documentation:** การจัดการ Swagger UI เพื่อให้ทีม Frontend หรือ Mobile นำ API ไปใช้งานต่อได้ง่าย

5: เชื่อมต่อ Frontend และ Deployment (Production Ready)

- **Introduction to Chainlit:** รู้จัก Python Framework สำหรับสร้าง Chat Interface สวยงามโดยไม่ต้องเขียน HTML/CSS
- **Frontend Integration:** การเชื่อมต่อ Chainlit UI เข้ากับ FastAPI Backend แบบ Streaming
- **Full Stack Docker Compose:** การมัดรวมทุก Service (vLLM, Qdrant, Backend, Frontend) ในไฟล์ docker-compose.yml เดียว
- **End-to-End Testing:** ทดสอบระบบจำลองสถานการณ์จริง ตั้งแต่ User พิมพ์ถาม จนถึง AI ค้นข้อมูลและตอบกลับ
- **Deployment & Scaling:** แนวทางการนำไปติดตั้งบน Server องค์กร และเทคนิคการขยาย Scaling เมื่อ User เพิ่มขึ้น