



สถาบันไอทีจีเนียส
IT GENIUS INSTITUTE

COURSE ID
AIC-07

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

AI / LLM ENGINEERING

On-Premise AI Chatbots with LangChain and LangGraph

สร้าง AI Chatbot ที่รันบนเซิร์ฟเวอร์องค์กร ข้อมูลไม่ออกไปนอกบ้าน

● 18 ชั่วโมง

● 3 วัน

● พื้นฐาน Python และ Docker

● ระดับ Intermediate



ระยะเวลา: 3 วัน

18 ชั่วโมง (3 วัน วันละ 6 ชม.) · 09:00-16:00 น.



รูปแบบ: Classroom / Hybrid

Onsite หรือ Online ผ่าน Microsoft Teams · Public / In-house



รายละเอียดหลักสูตร

www.itgenius.co.th

หลายองค์กรอยากใช้ AI Chatbot ตอบคำถามจากคู่มือ นโยบาย หรือเอกสารภายใน แต่ไม่สามารถส่งข้อมูลสำคัญออกไปประมวลผลบน Cloud API สาธารณะได้ ทั้งด้วยเหตุผลด้าน PDPA ความลับทางธุรกิจ และการควบคุมค่าใช้จ่าย ทางออกคือระบบ AI ที่รันบนเซิร์ฟเวอร์ขององค์กรเองทั้งหมด ตั้งแต่โมเดลภาษา ฐานข้อมูลเวกเตอร์ ไปจนถึงหน้าแชท ซึ่งวันนี้เครื่องมือโอเพนซอร์สพร้อมพอกที่จะทำได้จริง

หลักสูตรนี้พาผู้เรียนสร้าง AI Chatbot แบบ On-Premise ด้วย Python ตั้งแต่ต้นจนพร้อมใช้งาน เริ่มจากรันโมเดลโอเพนซอร์สด้วย Ollama และ vLLM ผ่าน API ที่เข้ากันได้กับ OpenAI ใช้ LangChain 1.x เชื่อมโมเดล นำเอกสารองค์กรเข้า Qdrant และค้นหาแบบ Hybrid จากนั้นสร้าง Agent ด้วย LangGraph 1.x ที่วนตรวจคำตอบเองได้ จำบทสนทนาด้วย Checkpointer และเรียก Tool หรือ MCP Server พร้อมจุดให้คนอนุมัติ แล้วต่อเป็นระบบจริงด้วย FastAPI แบบ Streaming หน้าแชท Chainlit การกำหนดสิทธิ์เข้าถึงเอกสาร และการวัดคุณภาพคำตอบ ปิดท้ายด้วยการรวมทุก Service ใน Docker Compose (อบรม 3 วัน วันละ 6 ชั่วโมง รวม 18 ชั่วโมง ระดับ Intermediate)

เครื่องมือ

LC LangChain

LG LangGraph

Py Python

OI Ollama

vec Vector DB

ราคาค่าอบรม · PUBLIC / IN-HOUSE

6,500 บาท / ท่าน

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสารประกอบการอบรม Lab Guide โค้ด Python ครบทุก Lab ไฟล์ docker-compose พร้อมใช้ และ Certificate · รับจัด In-house ขอใบเสนอราคาได้

สมัครผ่านเว็บไซต์ ลด 10%



เพิ่มเพื่อน
LINE

☎ 02-570-8449

☎ 088-807-9770

LINE @itgenius

🌐 www.itgenius.co.th



01 วัตถุประสงค์ Objectives

- เลือกและรันโมเดลภาษาโอเพนซอร์สบนเครื่ององค์กรด้วย Ollama และ vLLM ได้
- ใช้ LangChain 1.x เชื่อมโมเดลในเครื่องผ่าน API ที่เข้ากันได้กับ OpenAI
- สร้าง Pipeline นำเอกสารองค์กรเข้า Qdrant และค้นหาแบบ Hybrid ได้แม่นยำ
- ออกแบบ Agent ด้วย LangGraph 1.x ที่มี State, เชื้อไข และการวนตรรกะคำตอบ
- เก็บความจำบทสนทนาด้วย Checkpointer และเพิ่มจุดให้คนอนุมัติก่อนทำงานสำคัญ
- สร้าง Backend ด้วย FastAPI ที่ตอบแบบ Streaming และหน้าแชทด้วย Chainlit
- วาง Guardrail สกัดเข้าถึงเอกสาร และวัดคุณภาพคำตอบก่อนใช้งานจริง
- รวมทุก Service ใน Docker Compose และวางแผนติดตั้งบนเซิร์ฟเวอร์องค์กร

02 กลุ่มเป้าหมาย Target Audience

- Software Developer และ AI Engineer ที่ต้องการย้ายจาก Cloud API มาทำระบบ Self-hosted
- Data Scientist ที่ต้องการนำโมเดลภาษาไปใช้เป็นแอปพลิเคชันในองค์กร
- Web Developer สาย Backend หรือ Full-stack ที่ต้องการเพิ่มทักษะด้าน LLM
- Solution Architect และ Tech Lead ที่ต้องวางระบบ AI Chatbot ในองค์กร
- ทีม IT Infrastructure และ DevOps ที่ต้องดูแล AI Server ภายในองค์กร

03 จุดเด่นของหลักสูตร Highlights

- **ปรับปรุงใหม่ปี 2026** LangChain 1.x และ LangGraph 1.x
- **ข้อมูลไม่ออกนอกองค์กร** โมเดล Embedding และฐานข้อมูลรันในเครื่อง
- **RAG ที่ค้นหาแม่นยำขึ้น** Hybrid Search, Reranking และ Metadata
- **Agent ที่ทำงานตัวเอง** Agentic RAG, Tool, MCP และให้คนอนุมัติ
- **พร้อมขึ้นระบบจริง** FastAPI, Chainlit และ Docker Compose

04 ความรู้พื้นฐาน Prerequisites

- เขียน Python ได้ในระดับพื้นฐาน เช่น ฟังก์ชัน คลาส และการติดตั้งแพ็คเกจ
- เข้าใจแนวคิด REST API และรูปแบบข้อมูล JSON เบื้องต้น
- เคยใช้คำสั่ง Docker พื้นฐาน เช่น pull, run และ ps
- โน้ตบุ๊กที่ติดตั้ง Docker Desktop, Python และ VS Code ได้ พร้อมสิทธิ์ติดตั้งโปรแกรม

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
อุปกรณ์	Notebook RAM 16 GB ขึ้นไป เชื่อมต่ออินเทอร์เน็ต ติดตั้ง Docker Desktop ได้
รันโมเดล	Ollama บนโน้ตบุ๊ก และ vLLM บน Google Colab หรือ GPU ของผู้เรียนเอง
Framework	Python, LangChain 1.x, LangGraph 1.x, FastAPI และ Chainlit
ฐานข้อมูล	Qdrant สำหรับเวกเตอร์ และ PostgreSQL สำหรับเก็บความจำบทสนทนา
ข้อมูลตัวอย่าง	ชุดเอกสารนโยบายและคู่มือองค์กรสมมติภาษาไทยสำหรับทุก Lab

หมายเหตุ: ปรับปรุงใหม่ปี 2026 เทียบกับรุ่นเดิม: เปลี่ยนจาก Chain แบบเดิมมาใช้ LangChain 1.x กับ LangGraph 1.x เพิ่มรูปแบบเพิ่ม Hybrid Search, Agentic RAG, Checkpointer, MCP, Human-in-the-loop, Guardrail และการวัดคุณภาพคำตอบ เครื่องมือทั้งหมดเป็นโอเพนซอร์ส ผู้เรียนไม่จำเป็นต้องมี GPU เพราะใช้ Ollama กับโมเดลขนาดเล็กบนโน้ตบุ๊ก ส่วน Lab ที่ใช้ vLLM รับบน Google Colab หรือ GPU ที่ผู้เรียนจัดหาเอง ชื่อโมเดลและพีแอร์เปลี่ยนเร็ว เนื้อหาอ้างอิงเวอร์ชันล่าสุด ณ วันอบรม

06 ภาพรวมการเรียนรู้ 3 วัน Big Picture

วัน	หัวข้อหลัก
Day 1 Section 1-5	วางสถาปัตยกรรม AI ในองค์กร, รันโมเดลด้วย Ollama และ vLLM, ใช้ LangChain 1.x กับโมเดลในเครื่อง, ทำ Embedding ภาษาไทย และนำเอกสารองค์กรเข้า Qdrant
Day 2 Section 6-10	ค้นหาแม่นยำขึ้นด้วย Hybrid Search และ Reranking, สร้าง Agent ด้วย LangGraph 1.x, ทำ Agentic RAG, จำบทสนทนาด้วย Checkpointer และต่อ Tool, MCP พร้อม Human-in-the-loop
Day 3 Section 11-15	ทำ Backend ด้วย FastAPI แบบ Streaming, หน้าแชทด้วย Chainlit, วาง Guardrail และสิทธิ์เข้าถึงเอกสาร, วัดคุณภาพคำตอบ และรวมทุก Service ด้วย Docker Compose ใน Capstone



DAY
1

โมเดลในองค์กรและฐานความรู้

เนื้อหาการอบรม · Section 1-5

รวม
6 ชั่วโมง

1 สถาปัตยกรรม AI Chatbot แบบ On-Premise

- Cloud AI เทียบกับ On-Premise AI ทั้งด้านความเป็นส่วนตัว ค่าใช้จ่าย และการดูแลระบบ
- ส่วนประกอบของระบบ: Inference Engine, Embedding, Vector DB, Agent, API และ UI
- เลือกโมเดลโอเพนซอร์สที่รองรับภาษาไทย และประเมินขนาด GPU กับหน่วยความจำ
- Quantization และผลต่อความเร็ว ความแม่นยำ และทรัพยากรที่ใช้
- Lab: เตรียมเครื่องด้วย Docker, Python Virtual Environment และ uv

2 Lab: รันโมเดลด้วย Ollama และ vLLM

- Ollama สำหรับรันโมเดลขนาดเล็กบนโน้ตบุ๊กในช่วงพัฒนา
- vLLM สำหรับเซิร์ฟเวอร์: PagedAttention และ Continuous Batching
- รัน vllm serve ผ่าน Docker พร้อม API ที่เข้ากันได้กับ OpenAI
- เปิด Tool Calling และ Structured Output บนโมเดลที่รองรับ
- Lab: เรียกโมเดลในเครื่องด้วย OpenAI Python Client และวัดความเร็ว

3 LangChain 1.x กับโมเดลในเครื่อง

- ภาพรวม LangChain 1.x และสิ่งที่เปลี่ยนไปจาก Chain รุ่นเดิม
- เชื่อมโมเดลด้วย init_chat_model และ base_url ของ vLLM หรือ Ollama
- Message, Prompt Template และ Structured Output ด้วย Pydantic
- สร้าง Tool ด้วย @tool และ Agent อย่างง่ายด้วย create_agent
- Lab: Chatbot ถามตอบทั่วไปที่ไม่ส่งข้อมูลออกนอกเครื่อง

4 Embedding ภาษาไทยและ Qdrant

- หลักการ RAG และเหตุที่ช่วยลดการตอบมั่วของโมเดล
- เลือก Embedding Model หลายภาษาที่รันในเครื่อง เช่น BGE-M3
- ติดตั้ง Qdrant ด้วย Docker Compose และออกแบบ Collection กับ Payload
- Dense Vector, Sparse Vector และการเลือกค่าระยะทาง
- Lab: เก็บและค้นหาเวกเตอร์ด้วย Qdrant Python Client

5 Lab: Pipeline นำเอกสารองค์กรเข้าระบบ

- อ่านไฟล์ PDF, Word และหน้าเว็บภายใน แล้วแปลงเป็นข้อความที่สะอาด
- Chunking สำหรับภาษาไทย: ขนาด ส่วนซ้อน และการติดตามหัวข้อ
- เก็บ Metadata เช่น แหล่งที่มา หน้า หน่วยงาน และระดับความลับ
- อัปเดตเอกสารซ้ำโดยไม่เกิดข้อมูลซ้ำซ้อนใน Collection
- Lab: นำชุดเอกสารนโยบายองค์กรเข้า Qdrant ผ่าน LangChain



DAY
2

RAG ที่แม่นยำขึ้นและ Agent ด้วย LangGraph

เนื้อหาการอบรม · Section 6-10

รวม
6 ชั่วโมง

6 Lab: ค้นหาให้แม่นยำด้วย Hybrid Search

- ข้อจำกัดของ Similarity Search อย่างเดียวกับคำเฉพาะและรหัสเอกสาร
- Hybrid Search รวม Dense กับ Sparse ด้วย Query API และ RRF
- Reranking ด้วยโมเดลที่รันในเครื่องเพื่อจัดอันดับผลลัพธ์ใหม่
- กรองผลด้วย Metadata เช่น หน่วยงาน วันที่ และระดับความลับ
- Lab: เทียบผลค้นหาก่อนและหลังปรับ Retrieval

7 LangGraph 1.x พื้นฐาน

- ทำไมงาน Agent ต้องการ Graph แทน Chain ที่เป็นเส้นตรง
- StateGraph, State, Node, Edge และ Conditional Edge
- ออกแบบ State ด้วย TypedDict และ Reducer ของ Messages
- Compile Graph และแสดง Diagram เพื่อตรวจสอบ Logic
- Lab: Graph รับคำถาม ค้นหาข้อมูล และสรุปคำตอบพร้อมอ้างอิง

8 Lab: Agentic RAG

- ให้ Agent ตัดสินใจเองว่าต้องค้นหาข้อมูลหรือตอบได้ทันที
- ตรวจสอบความเกี่ยวข้องของเอกสารที่ค้นได้ก่อนนำไปตอบ
- เขียนคำถามใหม่แล้ววนค้นหาซ้ำเมื่อผลยังไม่ดีพอ
- ตรวจสอบคำตอบว่าอิงจากเอกสารจริง และกำหนดจำนวนรอบสูงสุด
- Lab: Agent ตอบคำถามนโยบายพร้อมแหล่งอ้างอิงทุกคำตอบ

9 ความจำและ Persistence

- Short-term Memory ด้วย Checkpointer และ thread_id
- เปลี่ยนจาก InMemorySaver เป็น PostgresSaver สำหรับใช้งานจริง
- จัดการประวัติแชทยาว: ตัด สรุป และควบคุมจำนวน Token
- Long-term Memory ด้วย Store สำหรับข้อมูลที่ผู้ใช้ให้ไว้
- Lab: Chatbot ที่จำบริบทได้แม้รีเซ็ตแชตเฟวเวอร์

10 Lab: Tool, MCP และ Human-in-the-loop

- ให้ Agent เรียก Tool ภายใน เช่น ค้นหาข้อมูลหรือระบบหลังบ้าน
- ต่อ Agent กับ MCP Server ภายในองค์กร
- หยุดรอให้คนอนุมัติด้วย interrupt แล้วทำต่อด้วย Command
- จำกัดสิทธิ์ของ Tool และบันทึกทุกการเรียกใช้
- Lab: Agent สร้างคำขอที่ต้องผ่านการอนุมัติก่อนบันทึก



DAY
3

จาก Prototype สู่ระบบที่ใช้งานจริง

เนื้อหาการอบรม · Section 11-15

รวม
6 ชั่วโมง

11 Lab: Backend API ด้วย FastAPI

- โครงสร้างโปรเจกต์แบบแยก Service และการตั้งค่าผ่าน Environment
- Endpoint สำหรับแชท ประวัติ และการนำเอกสารเข้า
- Async/Await และการเรียก Graph แบบ Asynchronous
- Streaming คำตอบทีละ Token ด้วย Server-Sent Events
- เอกสาร API อัตโนมัติด้วย OpenAPI และ Swagger UI

12 Lab: หน้าแชทด้วย Chainlit

- สร้างหน้าแชทด้วย Python โดยไม่ต้องเขียน HTML หรือ CSS
- เชื่อม Chainlit กับ FastAPI Backend แบบ Streaming
- แสดงแหล่งอ้างอิงและขั้นตอนการทำงานของ Agent
- อัปโหลดไฟล์จากหน้าแชท และเก็บคะแนนความพอใจคำตอบ
- ล็อกอินผู้ใช้และแยกประวัติแชทของแต่ละคน

13 ความปลอดภัยและ Guardrail

- กำหนดสิทธิ์เข้าถึงเอกสารตามผู้ใช้ด้วย Metadata Filter
- ป้องกัน Prompt Injection จากคำถามและจากเนื้อหาในเอกสาร
- กรองข้อมูลส่วนบุคคลในคำถามและคำตอบตามแนวทาง PDPA
- จำกัดอัตราการเรียกใช้และตั้ง Timeout ทุกชั้นของระบบ
- Lab: ทดสอบโจมตี Chatbot ของตัวเองแล้วปิดช่องโหว่

14 วัดคุณภาพและติดตามระบบ

- สร้างชุดคำถามทดสอบพร้อมคำตอบที่คาดหวังจากเอกสารจริง
- วัดความถูกต้อง การอ้างอิงตรงแหล่ง และอัตราที่ตอบว่าไม่ทราบ
- Tracing แบบ Self-hosted ด้วย Langfuse และเก็บ Log อย่างเป็นระบบ
- ดู Metrics ของ vLLM เช่น Latency และ Throughput
- Lab: วัดผลก่อนและหลังปรับ Prompt กับ Retrieval

15 Workshop: Capstone ระบบเต็มรูปแบบ

- รวม vLLM หรือ Ollama, Qdrant, PostgreSQL, Backend และ UI ใน docker-compose.yml
- Health Check, Volume และการสำรองข้อมูลของแต่ละ Service
- แนวทางขยายระบบเมื่อผู้ใช้เพิ่ม และการแยกเครื่อง GPU
- นำเสนอผลงาน รีวิวร่วมกัน และสรุป Checklist ก่อนขึ้นระบบจริง

07 เรียนจบแล้วได้อะไร Outcomes

- **รันโมเดลในองค์กรได้** Ollama, vLLM และ API แบบ OpenAI
- **สร้าง RAG ภาษาไทย** Qdrant, Hybrid Search และ Reranking
- **ออกแบบ Agent ได้** LangGraph, State และ Agentic RAG
- **จำและควบคุมได้** Checkpointer, MCP และให้คนอนุมัติ
- **ทำระบบครบวงจร** FastAPI, Streaming และ Chainlit
- **พร้อมใช้งานจริง** Guardrail, วัตถุ และ Docker Compose

08 สิ่งที่คุณเรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมฉบับสมบูรณ์ครบทั้ง 3 วัน และ Lab Guide ที่ละเอียด
- โค้ด Python ควบทุก Lab ทั้ง LangChain, LangGraph, FastAPI และ Chainlit
- ไฟล์ docker-compose สำหรับ Qdrant, PostgreSQL, Backend และ UI พร้อมเอกสารตัวอย่าง
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สามิตร โทยม ผู้สอนและนักพัฒนาที่สร้างระบบ AI Chatbot และ RAG สำหรับใช้งานภายในองค์กร ทำงานตามตอบจากเอกสารงานบริการภายใน และระบบที่ต้องเก็บข้อมูลไว้ในเซิร์ฟเวอร์ของตัวเอง จุดเด่นคือการสอนจากโค้ดที่ใช้ได้จริง อธิบายเหตุผลของการออกแบบแต่ละชั้น และพาผู้เรียนวัดผลทุกครั้งที่ได้รับระบบ เพื่อให้กลับไปสร้างและดูแล Chatbot ขององค์กรได้อย่างมั่นใจ

ราคาค่าอบรม 6,500 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

6,500 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

☎ 02-570-8449

📞 @itgenius

☎ 088-807-9770

✉ contact@itgenius.co.th



เพิ่มเพื่อน
LINE