



สถาบันไอจีเนียส
IT GENIUS INSTITUTE

COURSE ID
DOP-06

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

DEVOPS / LLMOPS

LLMOps: Evaluation, Observability and CI/CD

วัดผล ติดตาม และ Deploy แอป LLM ขึ้น Production อย่างมั่นใจ

● 12 ชั่วโมง

● 2 วัน

● พื้นฐาน Python, RAG และ Docker

● ระดับ Intermediate



ระยะเวลา: 2 วัน

12 ชั่วโมง (2 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

การสร้าง Chatbot หรือระบบ RAG ให้ตอบได้ดีในเครื่องของนักพัฒนาใช้เวลาไม่นาน แต่การดูแลให้ระบบตอบได้ดีอย่างสม่ำเสมอใน Production เป็นอีกเรื่องหนึ่ง แค่ Prompt นิดเดียวคำตอบบางกลุ่มก็แย่งโดยไม่มีใครรู้ เปลี่ยนรุ่นโมเดลแล้วค่าใช้จ่ายเพิ่มขึ้นหลายเท่า ผู้ใช้แจ้งว่าคำตอบผิดแต่ทีมหาไม่เจอว่าเกิดจากขั้นไหน เพราะแอป LLM ให้ผลไม่ตายตัว วิธีทดสอบและ Deploy แบบซอฟต์แวร์ทั่วไปจึงไม่พอ

หลักสูตรนี้ต่อยอดจากหลักสูตร RAG และ AI Chatbot โดยพาผู้เรียนนำแอป RAG ขึ้น Production อย่างมีระบบ เริ่มจากติดตั้ง Langfuse เพื่อทำ Tracing ทุกขั้นของแอป จัดการ Prompt แบบมีเวอร์ชัน สร้าง Dataset สำหรับวัดผลจากข้อมูลจริง ประเมินคุณภาพของ RAG ด้วย Ragas และทดสอบพฤติกรรมด้วย promptfoo จากนั้นนำการวัดผลไปเป็นด่านตรวจใน GitHub Actions Deploy ด้วย Docker ผ่าน LLM Gateway เฝ้าระวังคุณภาพ ความเร็ว และค่าใช้จ่ายหลังขึ้นระบบ และปิดท้ายด้วย Capstone ที่สร้าง Pipeline ครบตั้งแต่แค่ Prompt จนถึงเฝ้าระวังใน Production (อบรม 2 วัน วันละ 6 ชั่วโมง รวม 12 ชั่วโมง ระดับ Intermediate)

เครื่องมือ

Lf

Langfuse

Rg

Ragas

PF

promptfoo

GA

GitHub Actions

Dk

Docker

ราคาค่าอบรม · PUBLIC / IN-HOUSE

8,900 บาท / ท่าน

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสารประกอบการอบรม Lab Guide แอป RAG ตัวอย่าง ไฟล์ docker-compose ของ Langfuse ชุด Eval และ Workflow ของ GitHub Actions และ Certificate · รับผิดชอบ In-house ขอใบเสนอราคาได้

สมัครผ่านเว็บไซต์ ลด 10%



เพิ่มเพื่อน
LINE

☎ 02-570-8449

☎ 088-807-9770

LINE @itgenius

🌐 www.itgenius.co.th

01 วัตถุประสงค์ Objectives

- อธิบายวงจร LLMOps และความแตกต่างจาก MLOps และ DevOps แบบเดิม
- ติดตั้ง Langfuse และทำ Tracing ให้แอป RAG เห็นทุกขั้นตอนตั้งแต่ Retrieval ถึงคำตอบ
- จัดการ Prompt แบบมีเวอร์ชัน แยก Production กับ Staging และย้อนกลับได้
- สร้าง Dataset สำหรับวัดผลจาก Trace จริงและข้อมูลสังเคราะห์
- ประเมินคุณภาพ RAG ด้วย Ragas และทดสอบพฤติกรรมของแอปด้วย promptfoo
- ตั้ง Evaluation Gate ใน CI/CD ไม่ให้การเปลี่ยนแปลงที่ทำให้คุณภาพตกขึ้น Production
- Deploy แอป LLM ด้วย Docker และ LLM Gateway พร้อม Fallback และ Rate Limit
- ฝักระวังคุณภาพ Latency และค่าใช้จ่ายหลังขึ้นระบบ และวางมาตรการ Guardrails

02 กลุ่มเป้าหมาย Target Audience

- AI Engineer และนักพัฒนาที่สร้าง Chatbot หรือระบบ RAG แล้วต้องการนำขึ้น Production
- DevOps และ Platform Engineer ที่ต้องดูแลแอป LLM ร่วมกับทีมพัฒนา
- ML Engineer ที่ต้องการขยายจาก MLOps มาสู่งาน LLM
- Tech Lead ที่ต้องวางมาตรฐานการทดสอบและ Deploy แอป AI ของทีม
- ผู้ที่ผ่านหลักสูตร RAG & Knowledge Base หรือหลักสูตร AI Chatbot มาแล้ว

03 จุดเด่นของหลักสูตร Highlights

- **ต่อยอดจาก RAG** นำแอป RAG ที่สร้างแล้วขึ้น Production
- **เห็นทุกขั้นตอนด้วย Tracing** Langfuse แบบ Self-host uu Docker
- **วัดผลได้เป็นตัวเลข** Ragas, promptfoo และ LLM-as-a-Judge
- **Eval ใน CI/CD** GitHub Actions กันคุณภาพตก
- **คุมต้นทุนได้** Token, Caching และ Model Routing

04 ความรู้พื้นฐาน Prerequisites

- เขียน Python ได้ และเคยสร้างแอป LLM หรือระบบ RAG มาก่อน หรือผ่านหลักสูตร RAG & Knowledge Base
- ใช้ Git, GitHub และ Docker ขั้นพื้นฐานได้
- เข้าใจแนวคิด CI/CD เบื้องต้น
- เตรียม LLM API Key ที่มีเครดิต และบัญชี GitHub ของตนเอง พร้อมโน้ตบุ๊กที่ติดตั้ง Docker Desktop ได้

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
อุปกรณ์	Notebook RAM 16 GB ขึ้นไป ติดตั้ง Docker Desktop และ Docker Compose ได้
Observability	Langfuse แบบ Self-host บน Docker และแนวคิด OpenTelemetry สำหรับ GenAI
Evaluation	Ragas และ promptfoo บน Python และ Node.js
CI/CD และ Deploy	GitHub Actions, Docker และ LiteLLM เป็น LLM Gateway
LLM	LLM API Key ของผู้เรียน และโมเดลขนาดเล็กผ่าน Ollama สำหรับทดลอง

หมายเหตุ: ผู้เรียนเตรียม LLM API Key ที่มีเครดิตและบัญชี GitHub ของตนเองหรือของบริษัท ค่าใช้ API และนาทียของ GitHub Actions เป็นไปตามการใช้งานจริงและเงื่อนไขของแต่ละบริการ เครื่องมือหลักเป็นโอเพนซอร์สและรันบนเครื่องของผู้เรียนด้วย Docker หลักสูตรนี้ต่อยอดจากหลักสูตร RAG & Knowledge Base และหลักสูตร AI Chatbot โดยไม่สอนการสร้าง RAG ตั้งแต่ต้น และเน้นการดูแลคุณภาพ ส่วนการทดสอบเจาะระบบเชิงลึกอยู่ในหลักสูตร AI Security and LLM Red Teaming ชื่อเครื่องมือและฟีเจอร์ด้าน AI เปลี่ยนเร็ว เนื้อหาจะอัปเดตตามเวอร์ชันล่าสุด ณ วันอบรม

06 ภาพรวมการเรียนรู้ 2 วัน Big Picture

วัน	หัวข้อหลัก
Day 1 Section 1-6	วงจร LLMOps, ติดตั้ง Langfuse และทำ Tracing ให้แอป RAG, จัดการ Prompt แบบมีเวอร์ชัน, สร้าง Dataset สำหรับวัดผล, ประเมิน RAG ด้วย Ragas และทดสอบพฤติกรรมของแอปด้วย promptfoo
Day 2 Section 7-12	ตั้ง Evaluation Gate ใน GitHub Actions, Deploy ด้วย Docker และ LLM Gateway, Online Evaluation และ Monitoring, คุณด้นทุนและ Latency, Guardrails กับ Governance และ Capstone Pipeline ควบวงจร



DAY
1

Tracing, Prompt Versioning และ Evaluation

เนื้อหาการอบรม · Section 1-6

รวม
6 ชั่วโมง

1 LLMOps ต่างจาก MLOps และ DevOps อย่างไร

- ปัญหาเฉพาะของแอป LLM: ผลไม่ตายตัว คุณภาพวัดยาก และค่าใช้จ่ายผันตามการใช้งาน
- วงจร LLMOps: พัฒนา วัดผล Deploy ฝักระวัง และนำข้อมูลจริงกลับมาปรับปรุง
- สิ่งที่ต้องวัด: คุณภาพคำตอบ ความถูกต้องตามแหล่งข้อมูล Latency ค่าใช้จ่าย และความปลอดภัย
- Lab: ติดตั้งแอป RAG ตัวอย่างและลองหาคำตอบผิดเกิดจากข้อไหนโดยไม่มีเครื่องมือ

2 Lab: Tracing ด้วย Langfuse

- ติดตั้ง Langfuse แบบ Self-host ด้วย Docker Compose
- ใส่ Tracing ให้แอป RAG ด้วย Python SDK ให้เห็น Retrieval, Prompt และการเรียกโมเดล
- Trace, Span, Session, Token และค่าใช้จ่ายต่อคำขอ ตามแนวคิด OpenTelemetry สำหรับ GenAI
- Lab: ไล่ Trace ของคำตอบที่ผิดจนพบว่าปัญหาอยู่ที่ Retrieval

3 Lab: Prompt Management และ Versioning

- เก็บ Prompt ใน Git เทียบกับเก็บใน Prompt Management ของ Langfuse
- กำหนดเวอร์ชันและ Label อย่าง Production กับ Staging พร้อมค่า Config ของโมเดล
- ผูก Trace กับเวอร์ชันของ Prompt เพื่อเทียบผลก่อนและหลังแก้
- Lab: ปล่อย Prompt เวอร์ชันใหม่แล้วย้อนกลับเมื่อพบปัญหา

4 Lab: สร้าง Dataset สำหรับวัดผล

- Golden Dataset คืออะไร และควรมีคำถามแบบไหนบ้าง
- คัด Trace จริงที่มีปัญหาหรือได้ Feedback ไม่ดีเข้ามาเป็นกรณีทดสอบ
- สร้างชุดคำถามสังเคราะห์จากเอกสารด้วย Ragas แล้วให้คนตรวจทาน
- Lab: สร้าง Dataset 50 ข้อจากเอกสารและ Trace ของแอปตัวอย่าง

5 Lab: ประเมิน RAG ด้วย Ragas

- Metric หลักของ RAG: Faithfulness, ความเกี่ยวข้องของคำตอบ, Context Precision และ Context Recall
- LLM-as-a-Judge ทำงานอย่างไร และจุดอ่อนที่ต้องระวัง
- แยกปัญหาว่าอยู่ที่ Retrieval หรือ Generation จากคะแนนแต่ละด้าน
- Lab: วัดผลแอป RAG ก่อนและหลังปรับ Chunking แล้วสรุปว่าดีขึ้นจริงหรือไม่

6 Lab: ทดสอบพฤติกรรมด้วย promptfoo

- เขียน Test Case แบบ Assertion: ต้องมีคำสำคัญ รูปแบบ JSON ถูก และไม่หลุดหัวข้อ
- เทียบหลาย Prompt และหลายโมเดลในตารางเดียว
- ใช้ Assertion แบบ LLM-graded สำหรับเกณฑ์ที่วัดด้วยกฎไม่ได้
- Lab: เลือกโมเดลที่คุ้มที่สุดจากคุณภาพ ความเร็ว และค่าใช้จ่าย



DAY
2

CI/CD, Production และการคุมต้นทุน

เนื้อหาการอบรม · Section 7-12

รวม
6 ชั่วโมง

7 Lab: Evaluation Gate ใน GitHub Actions

- รับชุด Eval อัตโนมัติทุกครั้งที่เปิด Pull Request ที่แก้ Prompt หรือโค้ด
- กำหนดเกณฑ์ผ่าน และจัดการผลที่แคว้งจากความไม่ตายตัวของโมเดล
- แสดงผลเทียบกับรอบก่อนเป็นความเห็นใน Pull Request
- Lab: Pipeline ที่บล็อกการ Merge เมื่อคะแนน Faithfulness ต่ำกว่าเกณฑ์

8 Lab: Deploy ด้วย Docker และ LLM Gateway

- Containerize แอป RAG และจัดการ Config กับ Secret อย่างปลอดภัย
- ใช้ LiteLLM เป็น Gateway กลาง สลับโมเดลได้โดยไม่แก้โค้ด
- ตั้ง Fallback, Retry, Timeout และ Rate Limit ต่อทีมหรือต่อแอป
- Lab: Deploy แอปผ่าน Gateway แล้วจำลองโมเดลหลักล้มเพื่อทดสอบ Fallback

9 Lab: Online Evaluation และ Monitoring

- สุ่ม Trace จาก Production มาให้ LLM-as-a-Judge ให้คะแนนต่อเนื่อง
- Dashboard ของคุณภาพ Latency อัตรา Error และค่าใช้จ่าย
- ตั้งการแจ้งเตือนเมื่อคะแนนตก หรือ Latency และค่าใช้จ่ายสูงผิดปกติ
- Lab: ปิดวงจรจาก Feedback ไม่ดีใน Production สู่การทดสอบใหม่ใน CI

10 Lab: คุมต้นทุนและ Latency

- ติดตาม Token และค่าใช้จ่ายแยกตามผู้ใช้ ฟังก์ชัน และเวอร์ชันของ Prompt
- Prompt Caching ของผู้ให้บริการ และ Semantic Cache สำหรับคำถามซ้ำ
- Model Routing: ส่งงานง่ายให้โมเดลเล็ก งานยากให้โมเดลใหญ่
- Lab: ลดค่าใช้จ่ายต่อคำถามแล้ววัดผลยืนยันว่าคุณภาพไม่ลดลง

11 Guardrails, Safety และ Governance

- ปิดบังข้อมูลส่วนบุคคลใน Trace และกำหนดระยะเวลาเก็บข้อมูลตาม PDPA
- Guardrails ฝั่ง Input และ Output และการวัดผลกระทบต่อ Latency
- แผนรับมือเหตุการณ์เมื่อแอป LLM ตอบผิดหรือทำข้อมูลรั่ว
- Workshop: เขียน Release Checklist ของแอป LLM สำหรับทีม

12 Workshop: Capstone LLMOps Pipeline

- รับโจทย์ปรับปรุงแอป RAG ที่มีปัญหาคุณภาพและค่าใช้จ่ายสูง
- วิเคราะห์ด้วย Trace สร้าง Dataset และตั้งเกณฑ์วัดผล
- แก้ Prompt หรือ Retrieval ผ่าน Pull Request ที่ต้องผ่าน Evaluation Gate
- Deploy ผ่าน Gateway เปิด Monitoring และพิสูจน์ผลด้วยตัวเลข

07 เรียนจบแล้วได้อะไร Outcomes

- เห็นทุกชั้นของแอป Tracing ด้วย Langfuse
- จัดการ Prompt เป็นระบบ Versioning, Label และ Rollback
- วัดคุณภาพเป็นตัวเลข Dataset, Ragas และ promptfoo
- Eval ใน CI/CD GitHub Actions กันคุณภาพตก
- Deploy อย่างมั่นใจ Docker, LLM Gateway และ Monitoring
- คมด้นทุนได้ Caching, Routing และวัดผลย้อนขึ้น

08 สิ่งที่คุณเรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมฉบับสมบูรณ์ครบทั้ง 2 วัน และ Lab Guide ที่ละเอียด
- แอป RAG ตัวอย่างพร้อม Tracing และไฟล์ docker-compose ของ Langfuse และ LiteLLM
- ชุด Eval ของ Ragas และ promptfoo พร้อม Workflow ของ GitHub Actions
- Release Checklist และแม่แบบ Dataset สำหรับแอป LLM
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สามิตร โกยม ผู้สอนและวิศวกรที่พัฒนาและดูแลแอป LLM และระบบ RAG ที่ใช้งานจริงในองค์กร จุดเด่นคือการสอนให้ตัดสินใจจากตัวเลขแทนความรู้สึก ทุก Lab ให้วัดผลก่อนและหลังเปลี่ยนแปลง ทั้งคุณภาพ ความเร็ว และค่าใช้จ่าย เพื่อให้ผู้เรียนกลับไปวางกระบวนการ LLMOps ที่ทีมพัฒนาและทีมปฏิบัติการใช้ร่วมกันได้จริง

ราคาค่าอบรม 8,900 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

8,900 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

☎ 02-570-8449

☎ 088-807-9770

LINE @itgenius

✉ contact@itgenius.co.th



เพิ่มเพื่อน
LINE