



Generative AI for Software Developer



ในยุคที่ Generative AI เข้ามามีบทบาทอย่างลึกซึ้งในวงการพัฒนาซอฟต์แวร์ นักพัฒนายุคใหม่ไม่เพียงแต่ต้องเข้าใจภาษาโปรแกรม แต่ยังต้องสามารถใช้ภาษาอังกฤษเพื่อสื่อสารกับ AI อย่างมีประสิทธิภาพ หลักสูตร "Generative AI for Software Developer" นี้จึงถูกออกแบบมาเพื่อสอนนักพัฒนาให้สามารถใช้ Prompt Engineering ร่วมกับเครื่องมือ AI ชื่อนำอย่าง ChatGPT และ Claude เพื่อเพิ่มประสิทธิภาพในการพัฒนา เขียนโค้ด ออกแบบระบบ และสร้างเอกสารได้อย่างมืออาชีพ

อาชีพ

หลักสูตรนี้จะพาผู้เรียนก้าวเข้าสู่การใช้ AI ในทุกขั้นตอนของวงจรชีวิตการพัฒนาซอฟต์แวร์ ตั้งแต่การร่างไอเดีย การออกแบบสถาปัตยกรรมระบบ การสร้างโค้ดต้นแบบ การจัดการ legacy code ไปจนถึงการสร้างเว็บไซต์และเชทบอท โดยเน้นการลงมือปฏิบัติจริงในรูปแบบ workshop เพื่อให้ผู้เรียนสามารถนำความรู้ไปใช้ได้ทันทีในงานจริง

วัตถุประสงค์ของหลักสูตร (Course Objectives):

- เข้าใจพื้นฐานของ Generative AI, LLM, NLP และ Transformer Model ที่อยู่เบื้องหลัง ChatGPT และ Claude
- ฝึกใช้ Prompt ภาษาอังกฤษเพื่อสั่งงาน AI ให้ช่วยออกแบบ เขียนโค้ด แก้ปัญหา และวิเคราะห์โค้ด
- ใช้ AI ในการช่วยออกแบบฐานข้อมูล ระบบซอฟต์แวร์ และวาด Diagram
- สร้างแอปต้นแบบ (Prototype), เว็บไซต์, และ Chatbot ด้วย AI โดยไม่ต้องเขียนโค้ดทั้งหมดเอง
- เรียนรู้แนวทางปฏิบัติที่ดีที่สุด (Best Practices) ในการใช้ AI ร่วมกับกระบวนการพัฒนาซอฟต์แวร์จริง

จุดเด่นของหลักสูตร (Course Highlights):

- เน้นการใช้ Prompt ภาษาอังกฤษ เพื่อฝึกการสื่อสารกับ AI อย่างเป็นระบบ
- เรียนรู้ผ่านเครื่องมือจริง ได้แก่ ChatGPT, Claude, GitHub Copilot, และ Plugin/Extension ที่ใช้ในการพัฒนา
- Workshop เน้นลงมือปฏิบัติ เช่น การสร้าง Chatbot, ออกแบบ Database, วาด Diagram, และจัดการ Legacy Code
- รวมเทคนิค Prompt Engineering ที่ใช้ได้ผลจริงในงานพัฒนา เช่น Dev Prompt, Refactor Prompt, Unit Test Prompt
- อัปเดตสำหรับปี 2025 ครอบคลุม LLM รุ่นล่าสุด และเครื่องมือใหม่ เช่น Claude 3, GPT-4o, Copilot Workspace

กลุ่มเป้าหมายของหลักสูตร (Target Audience):

- นักพัฒนาซอฟต์แวร์ (Software Developers)
- Full Stack / Backend / Frontend Developers
- DevOps Engineers และ Software Architects
- AI Enthusiasts และ Technical Leads



- ผู้เรียนสาย Programming ที่กำลังเปลี่ยนผ่านสู่ยุค AI-first
- ผู้พัฒนาแอป หรือฟรีแลนซ์ที่ต้องการลดเวลาและต้นทุน
- อาจารย์ / ผู้ฝึกอบรม / Mentor ด้าน Software Development

ผู้เรียนต้องมีพื้นฐานอะไรบ้าง? (Prerequisites):

- มีพื้นฐานภาษาอังกฤษระดับอ่านออกเขียนได้ (ไม่ต้องถึงขั้นคล่อง)
- มีประสบการณ์เขียนโปรแกรมมาแล้ว เช่น Python, JavaScript, TypeScript, Java หรือภาษาใดก็ได้
- เข้าใจหลักการพื้นฐานของซอฟต์แวร์ เช่น MVC, API, Database, Git
- เคยใช้งาน VS Code หรือ IDE อื่นๆ สำหรับพัฒนาโปรแกรม
- มีความคุ้นเคยกับระบบปฏิบัติการ เช่น Windows, macOS หรือ Linux

ระยะเวลาในการอบรม:

- 18 ชั่วโมง (3 วัน)

ราคาคอร์สอบรม:

- 8,500 บาท / คน (หรือแบบเหมาจ่ายตามตกลงกัน)

วิทยากรผู้สอน:

- อาจารย์สาวิตร ไทยม

คอร์สที่ควรอบรมก่อนหน้า:

- ไม่มี

คอร์สต่อเนื่องที่แนะนำ:

- ไม่มี



เนื้อหาการอบรม:



วันที่ 1 Introduction to Generative AI for Developers (6 ชั่วโมง)

Section 1: Generative AI 101

- แนวคิดของ Generative AI และความหมายที่แท้จริง
- ตัวอย่างการใช้งานจริงในสาย Dev
- ความแตกต่างของ AI แบบ Rule-based vs LLM-based
- Tools เปรียบเทียบ: GPT-4o vs Claude 3 vs Copilot

Section 2: เข้าใจ LLMs และ Transformers

- โครงสร้างเบื้องหลัง LLMs (Transformer, Tokenization)
- RLHF คืออะไร? ทำให้ AI ฉลาดได้อย่างไร
- Prompt Engineering เบื้องต้น: Prompt = Programming

Section 3: Prompt ภาษาอังกฤษสำหรับ Developer

- วิธีตั้ง Prompt ที่ดี: Clear, Specific, Role-based
- โครงสร้าง Prompt ที่ควรใช้กับ ChatGPT และ Claude
- ตัวอย่าง Prompt สำหรับ Coding, Debugging, Explain Code



Hands-on Labs:

- สมัครและตั้งค่า ChatGPT, Claude, OpenAI API Key
- ทดลอง Prompt เบื้องต้น: Generate, Refactor, Translate Code

วันที่ 2 AI for Coding, Architecture & Debugging (6 ชั่วโมง)

Section 4: AI Coding Assistant อย่างมีประสิทธิภาพ

- แนะนำ GitHub Copilot / ChatGPT Code Interpreter
- แนะนำ Claude กับฟีเจอร์ Code Understanding
- การใช้ Prompt เพื่อ Generate Function, Unit Test, API Docs

Section 5: AI กับ Software Design & System Architecture

- Generating UML, ERD, Architecture Diagram จาก Prompt
- ใช้ AI ออกแบบ API ด้วย RESTful / GraphQL
- เทคนิค Craft Prompt ให้ AI ช่วยเขียนโครงสร้างระบบ

Section 6: จัดการ Legacy Code ด้วย AI

- ให้ AI อธิบายโค้ดเก่า (Explain Code)
- Refactor / Simplify / Add Comment
- Use case: อ่านโค้ด Java / Python ที่ไม่มีเอกสาร

Section 7: การใช้งาน React Router

- การติดตั้งและตั้งค่า React Router v6
- การสร้าง Route และ Navigation
- การใช้งาน Link, NavLink และ Nested Routes
- การจัดการ Dynamic Routes และ Route Parameters
- การปฏิบัติ: สร้างแอปพลิเคชัน Multi-page ด้วย React Router

Hands-on Labs:

- Generate S-UU API + Database Schema + Diagram
- สั่ง AI ทำ Reverse Engineering จากโค้ดที่ให้ได้
- Workshop: เขียน Prompt ให้ Claude อ่านและสรุปโค้ด



วันที่ 3 AI Prototyping, Automation & Best Practices (6 ชั่วโมง)

Section 8: Prototyping & UI Generation

- ใช้ AI สร้าง HTML, Tailwind UI, Bootstrap Template
- Generate Static Web + Hosting on GitHub Pages / Vercel
- AI tools: Framer AI, Uizard, Locofy, Relume

Section 9: AI-powered Chatbot & Assistant

- ใช้ ChatGPT สร้าง Chatbot ตอบคำถาม
- ใช้ Claude ช่วยสร้าง FAQ Bot / Coding Assistant
- แนวคิด RAG (Retrieval Augmented Generation)

Section 10: Best Practices & Limitations

- วิธีเชื่อม AI เข้า Dev Lifecycle: Plan → Build → Test → Deploy
- การวัดคุณภาพ Prompt / การปรับใช้ AI ในองค์กร
- ข้อควรระวังเรื่อง Security, Hallucination, Copyright

Hands-on Labs:

- สร้าง Static Website ด้วย Prompt เดียว
- ทดลองสร้าง Chatbot ตามโจทย์ Workshop
- สรุป Prompt ที่ใช้ได้จริง พร้อมรวมเป็น Prompt Library ส่วนตัว