บริษัท ไอทีจีเนียส เอ็นจิเนียริ่ง จำกัด
IT GENIUS ENGINEERING CO., LTD.
Tel: 02-570-8449, Mobile: 088-807-9770
Line ID: @itgenius

COURSE OUTLINE
รายละเอียดหลักสูตรอบรม

สร้าง AI Chatbots สำหรับองค์กรด้วย LangChain.js & Next.js & Supabase

ในยุคดิจิทัลที่องค์กรต่างขับเคลื่อนด้วยข้อมูล การปลดล็อกศักยภาพของข้อมูลภายในองค์กรให้พนักงานสามารถเข้าถึงและใช้งานได้อย่างชาญฉลาดคือโจทย์สำคัญ AI Chatbot ไม่ได้เป็นเพียงเครื่องมือถาม-ตอบทั่วไปอีกต่อไป แต่ได้วิวัฒนาการมาเป็น "ผู้เชี่ยวชาญดิจิทัล" ที่สามารถตอบคำถามซับซ้อนโดยอ้างอิงจากเอกสาร คู่มือ หรือฐานความรู้ขององค์กรได้อย่างแม่นยำ

Workshop นี้ถูกออกแบบมาเพื่อพาคุณลงมือสร้าง AI Chatbot ที่ใช้งานได้จริงตั้งแต่ต้นจนจบ ด้วยสถาปัตยกรรมสมัยใหม่บน Next.js ที่รวมทั้ง Frontend และ Backend API ไว้ในที่เดียว เราจะใช้ขุมพลังของ LangChain.js เพื่อเชื่อมต่อกับ Generative AI, สร้างระบบ RAG (Retrieval-Augmented Generation) เพื่อให้ AI ตอบคำถามจากเอกสารของคุณโดยเฉพาะ พร้อมทั้งสร้างระบบสมาชิกและจัดเก็บประวัติการสนทนาด้วย Supabase จนสามารถนำขึ้น Production ให้ทีมของคุณใช้งานได้จริง

วัตถุประสงค์:

- เข้าใจสถาปัตยกรรมและหลักการทำงานของ AI Chatbot ที่ใช้เทคนิค RAG ร่วมกับ LangChain.js
- พัฒนา Full-Stack Application ด้วย Next.js โดยสร้างทั้งส่วนติดต่อผู้ใช้ (UI) และ REST API สำหรับ AI
- สร้างระบบ AI ที่ตอบคำถามจากไฟล์เอกสารขององค์กรได้ (PDF, TXT, etc.)
- ผสานระบบยืนยันตัวตน (Authentication) และฐานข้อมูล (Database) เข้ากับแอปพลิเคชันด้วย Supabase
- นำโปรเจกต์ที่สร้างขึ้นทั้งหมดไปเผยแพร่ (Deploy) บน Vercel หรือ Docker เพื่อการใช้งานจริง

กลุ่มเป้าหมาย:

- Web Developers (Frontend/Full stack) ที่ต้องการเพิ่มทักษะด้าน AI และ LLMs เข้าไปในสายงาน
- JavaScript/TypeScript Developers ที่ต้องการสร้างแอปพลิเคชัน AI ที่ใช้งานได้จริง
- Solution Architects หรือ Tech Leads ที่ต้องการเข้าใจกระบวนการสร้าง AI Chatbot เพื่อนำไปประยุกต์ใช้กับระบบขององค์กร
- ผู้ที่สนใจสร้าง Product AI และต้องการเรียนรู้ Tech Stack ที่ทันสมัยและครบวงจร
- ทีมภายในองค์กร (IT/ผลิตภัณฑ์/ซัพพอร์ต/เซลส์) ที่อยากมีบอทถาม-ตอบจากคู่มือ/สเปก/นโยบาย
- อาจารย์/วิทยากร/ที่ปรึกษา ที่ต้องการเวิร์กช็อป RAG แบบนำไปสอนและปรับใช้ได้ทันที

ความรู้พื้นฐาน:

- JavaScript/TypeScript และ Next.js เบื้องต้น
- พอเข้าใจ REST/HTTP, .env, Git
- SQL เบื้องต้นจะช่วยมาก (ตาราง/คีย์/ความสัมพันธ์)
- คอมพิวเตอร์ส่วนตัวสำหรับใช้ในการอบรม (Windows, macOS, หรือ Linux)

ระยะเวลาในการอบรม:

- 12 ชั่วโมง (2 วัน)

วิทยากรผู้สอน:

- อาจารย์สามิตร โกยม

เนื้อหาการอบรม:

Section 1: ภาพรวม AI Chatbot กับ Langchain.js

- AI Chatbot คืออะไร: ทำความเข้าใจ Generative AI และ Large Language Models (LLMs)
- รู้จัก LangChain.js: Framework ที่เปรียบเสมือน "กาวใจ" เชื่อมต่อส่วนประกอบต่างๆ ในการสร้าง AI App
- ภาพรวม Technology Stack: ทำความรู้จักเครื่องมือที่จะใช้ทั้งหมดใน Workshop (Next.js, LangChain.js, Supabase, Shadcn/UI, Vercel)
- การตั้งค่าสภาพแวดล้อม: ติดตั้ง Node.js, pnpm/npm/yarn และเครื่องมือที่จำเป็น
- การสร้างโปรเจกต์ Next.js: เริ่มต้นโปรเจกต์ด้วย create-next-app และเลือกใช้ TypeScript

Section 2: การพัฒนา Rest API ใน Next.js เพื่อใช้งานกับ Langchain.js

- รู้จัก App Router: ภาพรวมโครงสร้างของ Next.js App Router
- Route Handlers คืออะไร: วิธีการสร้าง Backend API ภายในโปรเจกต์ Next.js
- การสร้าง API Endpoint: ลงมือสร้างไฟล์ route.ts สำหรับเป็นช่องทางสื่อสารกับ AI
- การจัดการ Request และ Response: เรียนรู้การใช้งานอ็อบเจกต์ Request และ NextResponse
- การอ่านข้อมูลจาก Client: วิธีการรับข้อมูล JSON ที่ผู้ใช้ส่งมาจากหน้าเว็บ
- การตั้งค่า Environment Variables: การใช้ไฟล์ .env.local เพื่อเก็บข้อมูลสำคัญอย่างปลอดภัย
- แบบฝึกหัด: ทดสอบ API Endpoint ด้วยเครื่องมืออย่าง Thunder Client หรือ Postman

Section 3: พื้นฐาน Langchain.js เชื่อมต่อกับ Gen AI

- องค์ประกอบหลักของ LangChain.js: ทำความรู้จัก Models, Prompts, และ Chains
- การเลือกและเชื่อมต่อ Models: วิธีการเชื่อมต่อกับ LLMs ผ่าน Provider ต่างๆ (เช่น Groq, OpenAI, OpenRouter)
- การสร้าง Prompt Templates: ออกแบบโครงสร้างคำสั่ง (Prompt) เพื่อควบคุมการตอบสนองของ AI
- การสร้าง Chain แรก (LLMChain): การรวม Model และ Prompt เข้าด้วยกันเป็น Chain พื้นฐาน
- การผสาน Chain เข้ากับ API: นำ Chain ที่สร้างขึ้นมาใช้งานใน Route Handler ที่เตรียมไว้
- การทำ Streaming Response: ทำให้ Chatbot ตอบกลับแบบ Real-time เหมือนพิมพ์สด

Section 4: ระบบยืนยันตัวตนด้วย Supabase Auth

- ภาพรวม Supabase: รู้จักแนวคิด Backend-as-a-Service (BaaS)
- การตั้งค่าโปรเจกต์ Supabase: สร้างโปรเจกต์ใหม่และทำความรู้จักหน้า Dashboard
- รู้จัก Supabase Auth: ภาพรวมความสามารถของระบบยืนยันตัวตน
- การติดตั้ง Supabase Client: เพิ่ม Library@supabase/supabase-js ในโปรเจกต์
- การสร้าง Client และ Middleware: ตั้งค่าการเชื่อมต่อ Supabase และการป้องกัน Route ฝั่ง Server
- การสร้าง UI สำหรับ Login/Register: พัฒนาฟังก์ชันและหน้าจอสำหรับให้ผู้ใช้เข้าสู่ระบบและสมัครสมาชิก
- การจัดการ Session: วิธีการตรวจสอบสถานะการล็อกอินของผู้ใช้ฝั่ง Client

Section 5: UI Chatbot ด้วย Prompt-kit-UI Shadcn/UI

- รู้จัก Shadcn/UI: แนวคิด Component ที่ปรับแต่งได้สูงและวิธีการติดตั้ง
- การใช้งาน Component ที่จำเป็น: นำCard,Input,Button,Avatar มาประกอบเป็นหน้าแชท
- การจัดการ State ฝั่ง Client: รู้จักprompt-kit และการใช้ HookuseChat เพื่อจัดการข้อความ
- การเชื่อมต่อ UI กับ API: เขียนฟังก์ชันเพื่อส่งข้อความจาก Input ของผู้ใช้ไปยัง/api/chat
- การแสดงผลแบบ Streaming: ทำให้ UI สามารถรับและแสดงผลข้อความที่ AI ทยอยส่งมาได้
- การปรับปรุง UX: เพิ่มสถานะ Loading และการจัดการข้อผิดพลาดเบื้องต้น

Section 6: AI Chatbot มีการเก็บประวัติ (Chat History)

- การออกแบบตารางใน Supabase: สร้าง Table สำหรับเก็บข้อความ (messages) และความสัมพันธ์กับผู้ใช้ (users)
- รู้จัก Row Level Security (RLS): การตั้งค่านโยบายความปลอดภัยเพื่อให้ผู้ใช้เห็นเฉพาะข้อมูลของตนเอง
- การบันทึกประวัติการแชท: แก้ไขโค้ดใน API ให้บันทึกคำถามของผู้ใช้และคำตอบของ AI ลง Database ทุกครั้ง
- การดึงประวัติมาแสดงผล: เขียนฟังก์ชันเพื่อดึงประวัติการแชทเก่ามาแสดงเมื่อผู้ใช้เปิดหน้าแชท
- การใช้ประวัติเป็น Context: ส่งประวัติการสนทนาเก่าเข้าไปใน Prompt เพื่อให้ AI พูดคุยได้ต่อเนื่อง

Section 7: เชื่อมต่อ AI กับเครื่องมือภายนอก (Tool Calling)

- Tool Calling/Function Calling คืออะไร: แนวคิดการมอบ "เครื่องมือ" ให้ AI เพื่อทำงานที่นอกเหนือจากการตอบคำถาม
- ตัวอย่าง Use Case: การดึงข้อมูลสภาพอากาศ, การคำนวณเลข, การค้นหาข้อมูลจาก API อื่น
- การนิยาม Tool: รูปแบบการสร้าง Tool ที่ LangChain สามารถเข้าใจได้ (ชื่อ, คำอธิบาย, Schema ของ Input)
- รู้จัก LangChain Agents: Agent ที่ทำหน้าที่เป็น "ผู้จัดการ" คอยตัดสินใจว่าจะตอบเองหรือจะใช้ Tool
- การสร้าง Agent Executor: การรวม Agent, Tools, และ Model เข้าด้วยกัน
- แบบฝึกหัด: สร้าง Tool ง่ายๆ (เช่น ฟังก์ชันคืนค่าวันที่ปัจจุบัน) และให้ AI เรียกใช้งานผ่านการแชท

Section 8: Document Loader, Embedding , Vector Store

- เจาะลึกองค์ประกอบของ RAG: ทบทวนสถาปัตยกรรมและส่วนประกอบแต่ละชิ้น
- Document Loaders: วิธีการโหลดข้อมูลจากแหล่งต่างๆ เช่น ไฟล์ PDF, TXT, หรือเว็บไซต์
- Text Splitters: ความสำคัญของการแบ่งเอกสาร และกลยุทธ์การแบ่ง (เช่น RecursiveCharacterTextSplitter)
- Embeddings คืออะไร: อธิบายแนวคิดการแปลงข้อความเป็น "พิกัดของความหมาย" (Vector)
- Vector Stores คืออะไร: รู้จักฐานข้อมูลสำหรับค้นหาข้อมูล Vector และการใช้งานpgvector บน Supabase
- สร้าง Script สำหรับ Ingestion: เขียนโค้ดสำหรับประมวลผลเอกสาร (Load -> Split -> Embed -> Store)

Section 9: พัฒนา RAG เพื่อให้ AI ตอบคำถามจากข้อมูลในเอกสารขององค์กร

- การสร้าง Retriever: การเขียนโค้ดเพื่อค้นหา Chunks ของข้อมูลที่เกี่ยวข้องกับคำถามของผู้ใช้จาก Vector Store
- การสร้าง Retrieval Chain: รูปแบบ Chain ใน LangChain ที่ออกแบบมาสำหรับงาน RAG โดยเฉพาะ
- การปรับแก้ Prompt สำหรับ RAG: เทคนิคการเขียน Prompt โดยสอดแทรก "Context" ที่ได้จาก Retriever
- การรวม RAG เข้ากับ API: ปรับปรุง/api/chat ให้ทำงานแบบ RAG เต็มรูปแบบ
- การทดสอบระบบ RAG: ทดลองถามคำถามที่ AI ต้องใช้ข้อมูลจากเอกสารที่เราใส่เข้าไปเท่านั้นจึงจะตอบได้
- การแสดงแหล่งอ้างอิง (Source): เพิ่มความสามารถในการแสดงว่า AI นำข้อมูลจากส่วนไหนของเอกสารมาตอบ

Section 10: การเผยแพร่ (Deployment) โปรเจกต์ไปใช้งานจริง

- รู้จัก Vercel: แพลตฟอร์มสำหรับ Deploy โปรเจกต์ Next.js ที่ง่ายและทรงพลัง
- การเตรียมโปรเจกต์: ตรวจสอบความเรียบร้อยของโค้ดและ Dependencies
- การตั้งค่า Environment Variables บน Vercel: นำข้อมูลจาก.env.local ไปใส่ใน Dashboard ของ Vercel
- การเชื่อมต่อกับ GitHub: ตั้งค่า Continuous Deployment (CD) ให้ Deploy อัตโนมัติเมื่อมีโค้ดใหม่เข้ามา
- ขั้นตอนการ Deploy: การกด Deploy ครั้งแรก และการดู Log ระหว่าง Build
- การแก้ปัญหาที่พบบ่อย: แนวทางการตรวจสอบและแก้ไขเมื่อ Deployment ไม่สำเร็จ
- แนวทางการ Deploy บน Docker: การสร้าง Dockerfile และการตั้งค่า Container
- การทดสอบบน Production: ทดสอบการทำงานของ Chatbot บน URL จริง