

## ใบงาน Mini Project 1: การพัฒนา Q&A Chatbot ด้วย RAG

มอบหมายโดย อ.ปกรณ์ ยุบลโกศล (ผู้สอน)



**รายวิชา:** AIE322: การเรียนรู้ของเครื่องแบบมีผู้สอน (Supervised Machine Learning)



**กำหนดส่ง:** ภายในวันศุกร์ที่ 21 ก.พ. 2568



**ช่องทางส่งงาน:** Google form สำหรับกรอกรายละเอียด

1. เข้าไปกรอกข้อมูลใน google form <https://forms.gle/Yk6CJuUAumX64GqQ8>
2. สำหรับ นศ ที่ทำงานเป็นกลุ่ม ให้กรอกชื่อและรหัสนักศึกษาใน google form ของสมาชิกทุกคนให้ครบ
3. ส่ง ลิงก์ GitHub ที่เก็บโค้ด (ใส่ link ใน google form)
4. ส่ง ลิงก์ YouTube ที่บันทึกการนำเสนอ (ใส่ link ใน google form)
5. เมื่อ submit แบบฟอร์มแล้ว ให้กด turn in ใน ms team ด้วย

### วัตถุประสงค์ของโปรเจกต์

1. ศึกษาแนวคิด RAG (Retrieval-Augmented Generation) สำหรับพัฒนา Chatbot
2. พัฒนา Chatbot ที่สามารถตอบคำถามใน หัวข้อที่ผู้เรียนสนใจ (เช่น กีฬา แฟชั่น เทคโนโลยี ฯลฯ)
3. เรียนรู้การใช้งาน LLM (Large Language Model) และ RAG Framework ในภาษา Python
4. ฝึกทักษะการนำเสนอแนวคิดและการเขียนโค้ดผ่านวิดีโอ YouTube
5. ฝึกการใช้งาน GitHub ในการจัดเก็บและแชร์โค้ด

### ข้อกำหนดของโปรเจกต์

เรื่อง	รายละเอียด
หัวข้อของ Chatbot	นักศึกษาสามารถเลือกหัวข้อใดก็ได้ที่สนใจ เช่น กีฬา อาหาร เกม เทคโนโลยี ฯลฯ
ประเภทของ Chatbot	ใช้ RAG (Retrieval-Augmented Generation) ในการพัฒนา Chatbot เพื่อให้ตอบคำถามจากข้อมูลที่กำหนดเอง (Q&A) โดยการโต้ตอบระหว่างผู้ใช้กับ chatbot สามารถทำผ่าน command line ได้
เครื่องมือที่ใช้ (แล้วแต่สะดวก)	Python, LangChain หรือ FAISS/Pinecone สำหรับ RAG, OpenAI API หรือ Llama-Index, Flask (ถ้าต้องการทำเว็บ)
รูปแบบการส่งงาน	1. อัปโหลดโค้ดทั้งหมดลง GitHub พร้อม README อธิบายการใช้งาน 2. บันทึกวิดีโอลง YouTube (อธิบายคอนเซ็ปต์โดยอาจจะทำ slide นำเสนอ, โค้ด, และสาธิตการทำงาน) 3. ส่งลิงก์ GitHub และ YouTube
ความยาวของวิดีโอ	5-7 นาที (ไม่ต้องอธิบายโค้ดละเอียด เอาเฉพาะส่วนที่สำคัญหลักๆ แต่ต้องแสดงการทำงาน)
คะแนนพิเศษ	ถ้าพัฒนาเป็นเว็บแอป (เช่น Flask + HTML/CSS) จะได้รับ คะแนนพิเศษ

เกณฑ์การให้คะแนน (100 คะแนน)

หมวดหมู่	รายละเอียด	สัดส่วนคะแนน
แนวคิดและการออกแบบ (Concept & Design)	อธิบายแนวคิด RAG และโครงสร้าง Chatbot ของตนได้เข้าใจง่าย	20%
การพัฒนา Chatbot (Implementation)	Chatbot ทำงานได้จริง สามารถดึงข้อมูลจากฐานความรู้ได้	30%
การนำเสนอ (Presentation)	วิดีโออธิบายชัดเจน มีการสาธิตการทำงาน	20%
การใช้งาน GitHub	โค้ดมีการอัปโหลดลง GitHub และมี README อธิบาย	10%
ความคิดสร้างสรรค์	การออกแบบโครงสร้าง หรือฟีเจอร์พิเศษที่เพิ่มเข้ามา	10%
คะแนนพิเศษ	หากพัฒนาเป็น Web App (เช่น Flask)	+10%

แหล่งอ้างอิง / ตัวอย่าง

- Code a simple RAG from scratch (ระดับความยาก: กลาง)  
<https://huggingface.co/blog/ngxson/make-your-own-rag>
- Question Answering using Transformers Hugging Face Library (ระดับความยาก: กลาง)  
<https://www.youtube.com/watch?v=DNRIUGtKIVU>
- Build Your Own RAG System with HuggingFace and Open Source LLMs (ระดับความยาก: ซับซ้อน)  
<https://www.youtube.com/watch?v=SNpmkx9cpck>