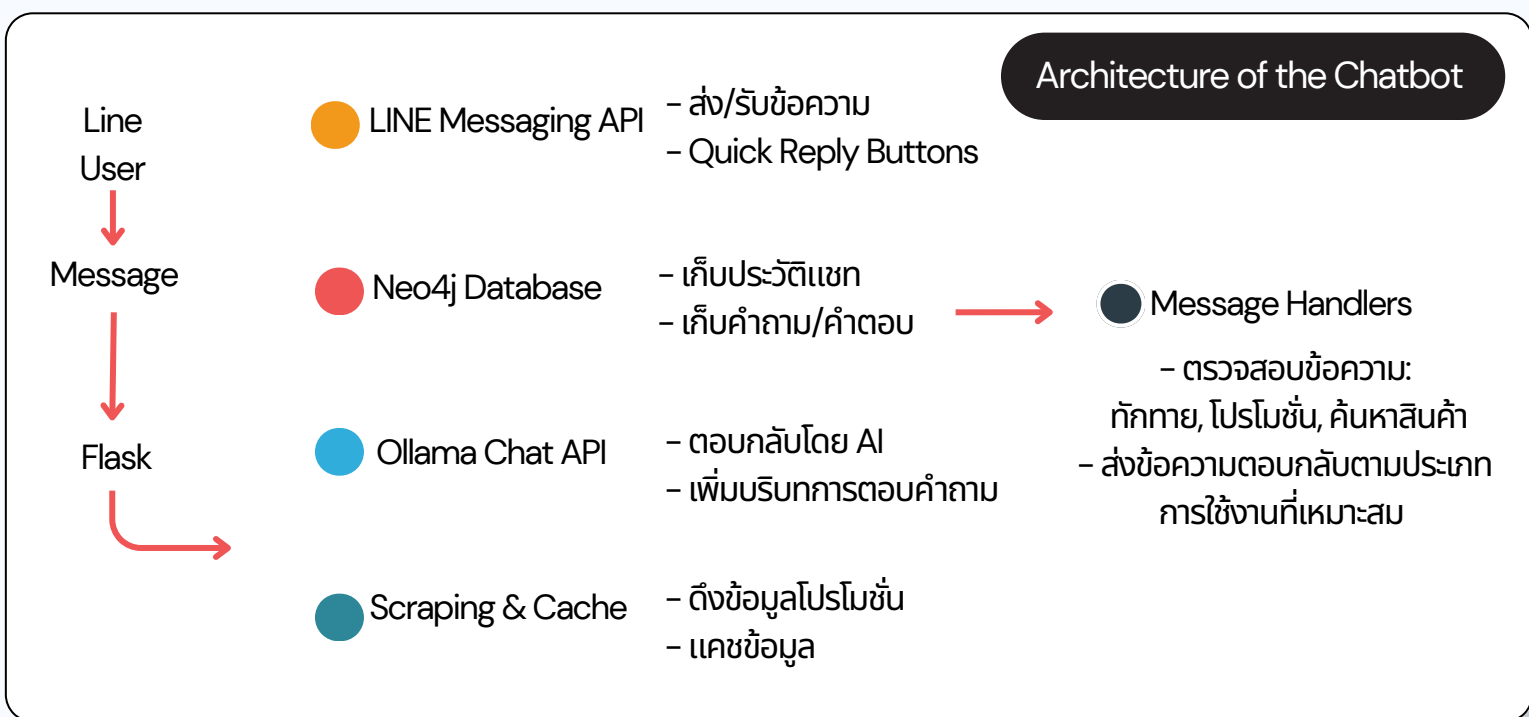




Chatbot: Lotus's Discount Bot



ปัจจุบันลูกค้าต้องการความสะดวกรวดเร็วในการค้นหาสินค้าลดราคาและโปรโมชั่นที่น่าสนใจ Lotus's Discount Bot จึงถูกพัฒนาขึ้นมาเพื่อให้ลูกค้าสามารถค้นหาข้อมูลสินค้าและโปรโมชั่นได้ง่ายๆ ผ่านการพิมพ์ชื่อหรือคำค้นหา ช่วยเพิ่มความสะดวก ประหยัดเวลา และทำให้ลูกค้าได้รับข้อเสนอที่คุ้มค่ามากยิ่งขึ้น



Set Up the Development Environment

Python Libraries ที่ใช้ในการพัฒนา

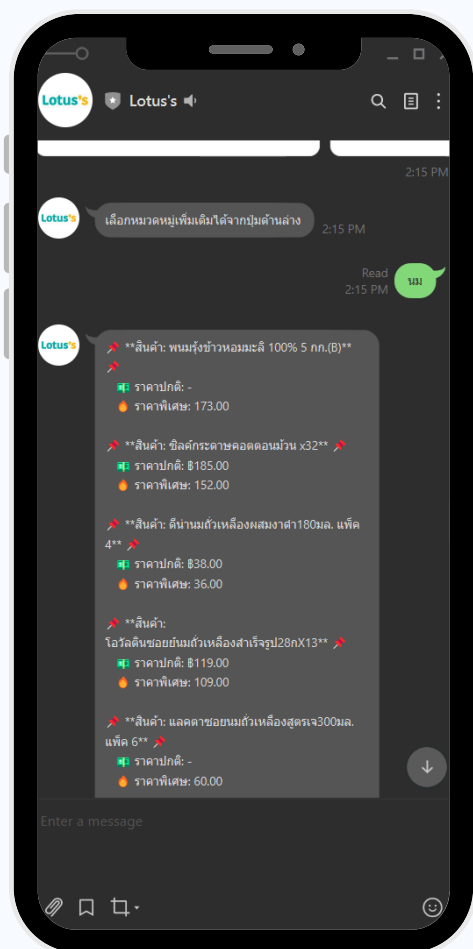
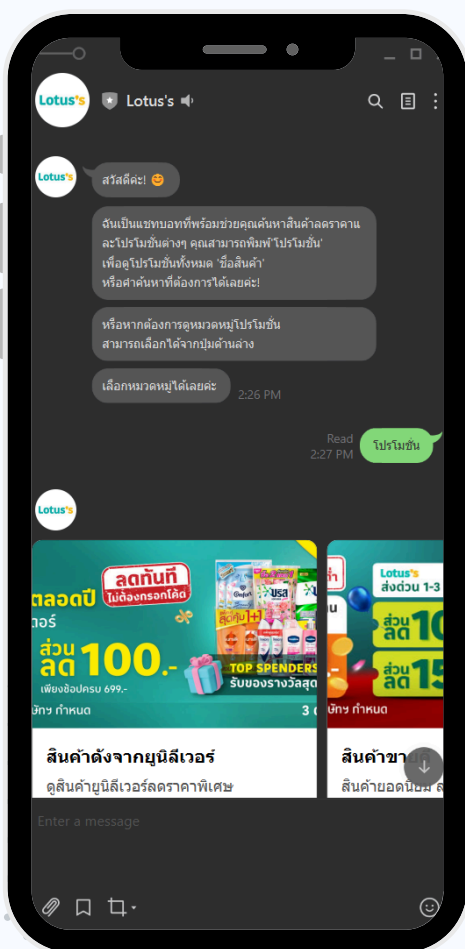
- **Flask** : ใช้สร้าง Web Application และ API
- **LineBotApi** และ **WebhookHandler** จากไลบรารี linebot : ใช้เชื่อมต่อและจัดการ Line Bot
- **BeautifulSoup** (จาก bs4) : ใช้ในการดึงข้อมูล (scrape) จากหน้าเว็บ
- **Selenium** : ใช้สำหรับกร Automation การทำงานของ Web Browser เช่นการดึงข้อมูลหน้าเว็บ
- **chromedriver_autoinstaller** : ใช้ในการติดตั้ง ChromeDriver อัตโนมัติสำหรับ Selenium
- **Neo4j** (จาก neo4j) : ใช้เชื่อมต่อและทำงานกับฐานข้อมูลกราฟ Neo4j
- **SentenceTransformer** (จาก sentence_transformers) : ใช้สำหรับการประมวลผลภาษาธรรมชาติ (NLP) ในการเปรียบเทียบประโยคหรือข้อความ
- **NumPy** : ใช้สำหรับการจัดการข้อมูลที่เป็นตัวเลข
- **ollama** : ใช้ในการประมวลผลแชทผ่านโมเดล AI

Model NLP ที่ใช้

SentenceTransformer รุ่น 'sentence-transformers/distiluse-base-multilingual-cased-v2' : ใช้ในการแปลงประโยคเป็นเวกเตอร์เพื่อนำไปเปรียบเทียบความคล้ายคลึงกันของประโยค

Configure the Chatbot for Testing

- **Neo4j** : ใช้จัดเก็บข้อมูลการถามตอบของผู้ใช้
- **Flask** : เว็บเซิร์ฟเวอร์ที่ให้บริการ API สำหรับรับข้อมูลจาก Line Webhook และเชื่อมต่อกับ Line API เพื่อส่งข้อความ
- **Line Bot API** และ **Line Webhook** : ใช้รับและตอบข้อความผ่าน Line Bot โดยต้องตั้งค่า access_token และ secret เพื่อเชื่อมต่อกับ Line API
- **Selenium** : ตั้งค่าการใช้งานโดยใช้ ChromeOptions เพื่อตั้งค่าการทำงานของเบราว์เซอร์ในโหมด headless (ไม่มี UI) สำหรับการดึงข้อมูลจากเว็บ



- ### แนวคิดสำหรับการสร้างเป็น Smart Chatbot
1. ตอบคำถามเกี่ยวกับโปรโมชั่นและส่วนลด ของ Lotus's ได้อย่างรวดเร็ว
 2. สามารถ ค้นหาสินค้าที่มีส่วนลด และ แนะนำโปรโมชั่นพิเศษ
 3. เชื่อมต่อกับฐานข้อมูลสินค้าและโปรโมชั่น ของ Lotus's
 4. ใช้ AI เพื่อปรับแต่งคำแนะนำ ตาม ความสนใจของลูกค้า
 5. รองรับ การใช้งานหลายภาษา และทำงานแบบเรียลไทม์

code : <https://github.com/Naowan/Mini-project-Recommendation-chatbot-Lotus>



PDPA CHATBOT

การคุ้มครองข้อมูลส่วนบุคคลเป็นประเด็นสำคัญในยุคดิจิทัล ซึ่งองค์กรต้องปฏิบัติตามข้อกำหนด PDPA เพื่อปกป้องสิทธิ์ของผู้ใช้ Chatbot ออกแบบเพื่อตอบคำถามเกี่ยวกับ พระราชบัญญัติคุ้มครองข้อมูลส่วนบุคคล (PDPA) ให้ผู้ใช้เข้าใจเกี่ยวกับการจัดการข้อมูลส่วนบุคคลและความเป็นส่วนตัวได้ง่ายขึ้น

ARCHITECTURE OF THE CHATBOT

User Interface



LINE platform

- เมื่อเชื่อมต่อกับแพลตฟอร์มช่วยให้สามารถถามคำถามได้โดยตรง
- ใช้ LINE API ในการจัดการข้อความและส่งคำตอบให้ผู้ใช้

Backend Components

Flask Web Server

Neo4j Database

Ollama API

Sentence Transformers Model

- จัดการการรับ-ส่งข้อมูลระหว่างผู้ใช้และเซิร์ฟเวอร์
- เก็บข้อมูลเกี่ยวกับ PDPA และคำตอบสำหรับคำถามทั่วไป
- ให้ข้อมูลในกรณีที่ไม่มีคำตอบในฐานข้อมูล
- วิเคราะห์ข้อความเพื่อค้นหาคำตอบที่ใกล้เคียงที่สุด

SET UP THE DEVELOPMENT ENVIRONMENT

Python Libraries Used

- **requests** : สำหรับการส่งและรับข้อมูลผ่าน HTTP requests เช่น การเรียกใช้ API ของ Ollama
- **json** : ใช้จัดการข้อมูลในรูปแบบ JSON เช่น ข้อมูลจาก Neo4j หรือ Line API
- **neo4j** : สำหรับเชื่อมต่อและรับ query บนฐานข้อมูล Neo4j
- **flask** : ใช้สร้างและรับเว็บเซิร์ฟเวอร์ในการจัดการกับ Line Bot API
- **linebot** : ไดรเวอร์ของ Line Bot API ที่ช่วยในการตอบข้อความกลับไปยังผู้ใช้
- **sentence-transformers** : ใช้สำหรับ NLP Model ในการจับคู่คำถามกับฐานข้อมูลคำถามที่เตรียมไว้
- **numpy** : สำหรับการคำนวณเชิงคณิตศาสตร์ เช่น การคำนวณ cosine similarity ของคำถาม

NLP Model

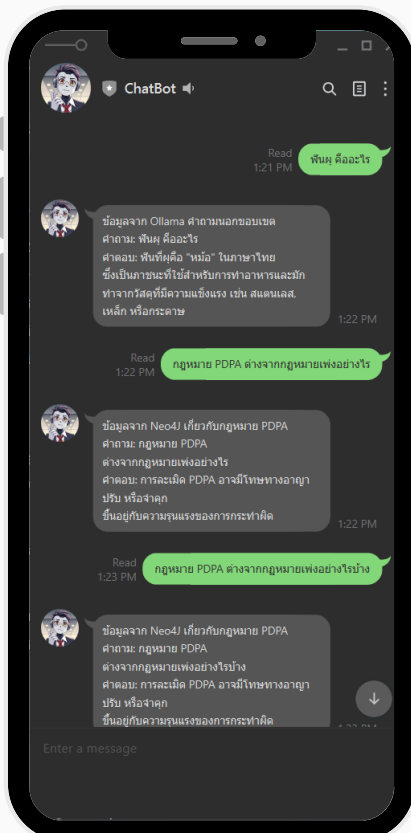
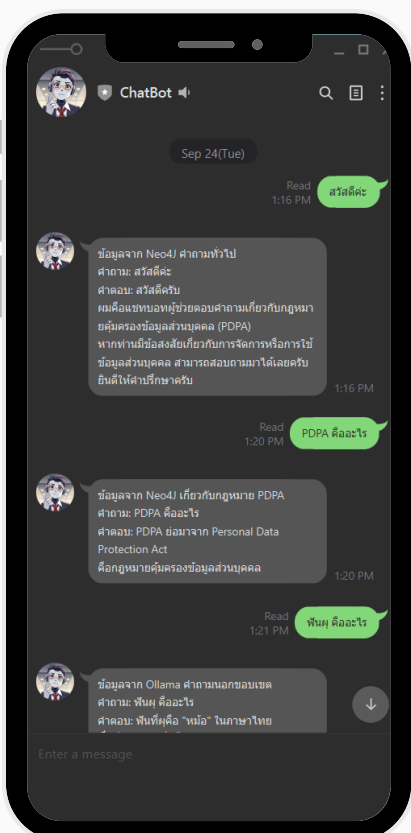
SentenceTransformer สำหรับการจับคู่คำถามกับฐานข้อมูล

CONFIGURE THE CHATBOT FOR TESTING

- **Neo4j Database**: เชื่อมต่อฐานข้อมูลสำหรับจัดเก็บข้อมูลคำถามและคำตอบที่เกี่ยวข้องกับ Greeting และ PDPA
- **Flask**: สร้าง API Endpoint ที่จะรับข้อความจาก Line Bot และประมวลผลคำตอบ
- **Line API**: ใช้ Line Bot API และ Webhook Handler เพื่อให้บอทสามารถตอบกลับผู้ใช้

การประมวลผลและตอบคำถาม

- **Corpus จาก Neo4j**: โหลดฐานข้อมูล Greeting และ PDPA จาก Neo4j เพื่อตรวจสอบคำถาม
- **Cosine Similarity**: คำนวณความใกล้เคียงระหว่างคำถามที่ได้รับและข้อมูลในฐานข้อมูล
- **Fallback ไปที่ Ollama API**: หากไม่มีคำถามที่ตรงกับฐานข้อมูล ระบบจะส่งคำถามไปยัง Ollama API เพื่อรับคำตอบเพิ่มเติม



แนวคิดการสร้าง SMART CHATBOT

- ใช้เทคโนโลยี machine learning (ML) ด้วย Sentence Transformers เพื่อตรงกับคำถามของผู้ใช้กับฐานข้อมูล.
- สร้างคำตอบด้วย AI ผ่าน Ollama API สำหรับคำถามที่ไม่ครอบคลุม.
- เป็น โซลูชันแบบผสมผสาน: ใช้ฐานข้อมูลสำหรับคำถามที่ทราบและใช้ AI สำหรับคำถามใหม่หรือซับซ้อน.

code : <https://github.com/Naowan/PDPA-chatbot>