

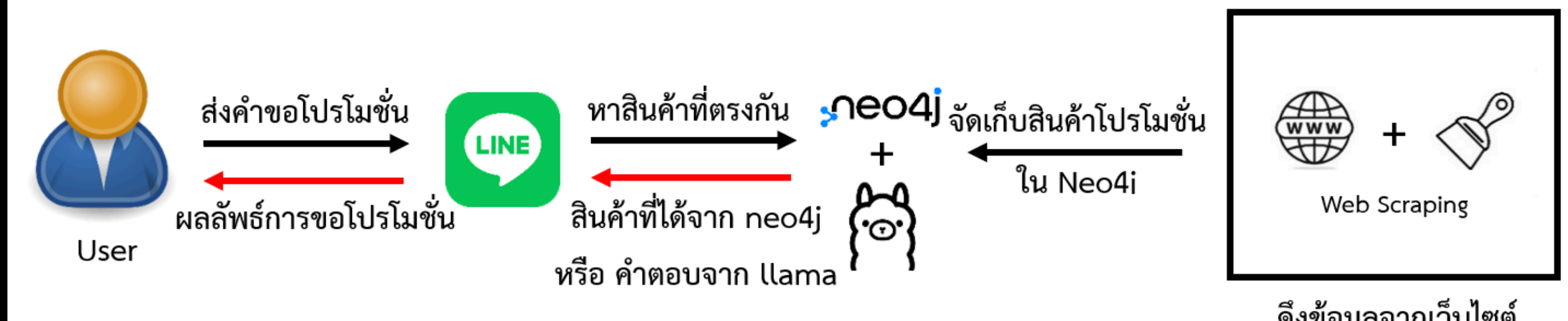
Sephora Thailand

Online Customer Service

ที่มาและความสำคัญ

แขกบอกผู้ช่วย Sephora ช่วยให้ผู้ใช้งานสามารถเข้าถึงโปรโมชั่นและผลิตภัณฑ์ที่ต้องการได้อย่างสะดวกและรวดเร็ว ผ่านการเลือกดูผลิตภัณฑ์ตามแบรนด์ ส่วนลด และคะแนน นอกจากนี้ยังเสริมประสบการณ์ด้วยการตอบคำถามเฉพาะจาก Ollama ที่ช่วยให้ข้อมูลเชิงลึกเกี่ยวกับเครื่องสำอางและผลิตภัณฑ์ความงาม ซึ่งช่วยเพิ่มความมั่นใจให้กับผู้ใช้ในการเลือกซื้อ

การทำงานของ ChatBot



Set Up the Development Environment

1. pip install Flask
 - from flask import Flask
2. pip install neo4j
 - from neo4j import GraphDatabase
3. pip install pyngrok
 - pyngrok import ngrok
4. pip install line-bot-sdk
 - from linebot import LineBotApi, WebhookHandler
 - from linebot.exceptions import InvalidSignatureError, LineBotApiError
 - from linebot.models import MessageEvent, TextMessage, TextSendMessage, QuickReply, QuickReplyButton, MessageAction, FlexSendMessage
5. pip install requests
 - requests
6. pip install fuzzywuzzy
 - fuzzywuzzy import fuzz
7. pip install bs4
 - from bs4 import BeautifulSoup
8. pip install selenium
 - from selenium import webdriver
 - from selenium.webdriver.common.by import By
 - from selenium.webdriver.support.ui import WebDriverWait
 - from selenium.webdriver.support import expected_conditions as EC
9. pip install chromedriver-autoinstaller
 - import chromedriver_autoinstaller

Configure the Chatbot

Setup Chrome

```
# Setup Chrome options
chrome_options = webdriver.ChromeOptions()
chrome_options.add_argument('--headless')
chrome_options.add_argument('--no-sandbox')
chrome_options.add_argument('--disable-dev-shm-usage')
chrome_options.add_argument('user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36')
```

Neo4j

```
# Neo4j credentials
URI = "neo4j://localhost"
AUTH = ("neo4j", "password")

# Initialize Flask app
app = Flask(__name__)

# Neo4j connection
def run_query(query, parameters=None):
    with GraphDatabase.driver(URI, auth=AUTH) as driver:
        with driver.session() as session:
            result = session.run(query, parameters)
            return [record for record in result]
```

api scrap

```
@app.route('/api/products', methods=['GET'])
def get_products():
    page = 1
    all_products = []

    while True:
        url = f"https://www.sephora.co.th/sale?page={page}"
        print(f"Scraping page: {url}")
        products_on_page = scrape_products(url)

        if not products_on_page:
            break

        print(f"Found {len(products_on_page)} products on page {page}")
        all_products.extend(products_on_page)
        page += 1
        time.sleep(2)
```

Llama

```
def llama_response(msg, history_chat):
    print("====this is llama====")

    # ตรวจสอบว่าคำถามเกี่ยวข้องกับหรือไม่
    if not is_relevant_question(msg):
        return "ขออภัยครับ ฉันสามารถตอบคำถามเกี่ยวกับเครื่องสำอาง ผลิตภัณฑ์ความงาม และน้ำหอมเท่านั้นครับ"

    # Combine user prompt with chat history
    history = "\n".join(history_chat) # Join the history into a single string
    full_prompt = f"{history}\nUser: {msg}\nBot (male customer service agent): " # Format the prompt with history

    payload = {
        "model": llama_model,
        "prompt": full_prompt + "ตอบคำถามเกี่ยวกับเครื่องสำอาง ผลิตภัณฑ์ความงาม และน้ำหอม ให้สุภาพ ใจกว้างไทยสันถวไมตรี และลงท้ายด้วยคำว่า 'ครับ'",
        "stream": False,
        "options": {
            "num_predict": 100, # Adjust as needed
            "num_ctx": 1024,
            "temperature": 0.8,
        }
    }
```

```
url = "http://localhost:11434/api/generate"
headers = {"Content-Type": "application/json"}
llama_model = "supachai/llama-3-typhoon-v1.5"
```

Flask&Line

```
# Flask app
app = Flask(__name__)

# Start ngrok tunnel
port = "5000"
ngrok.set_auth_token("-----")
public_url = ngrok.connect(port).public_url
print(f"ngrok tunnel {public_url} -> http://127.0.0.1:{port}")

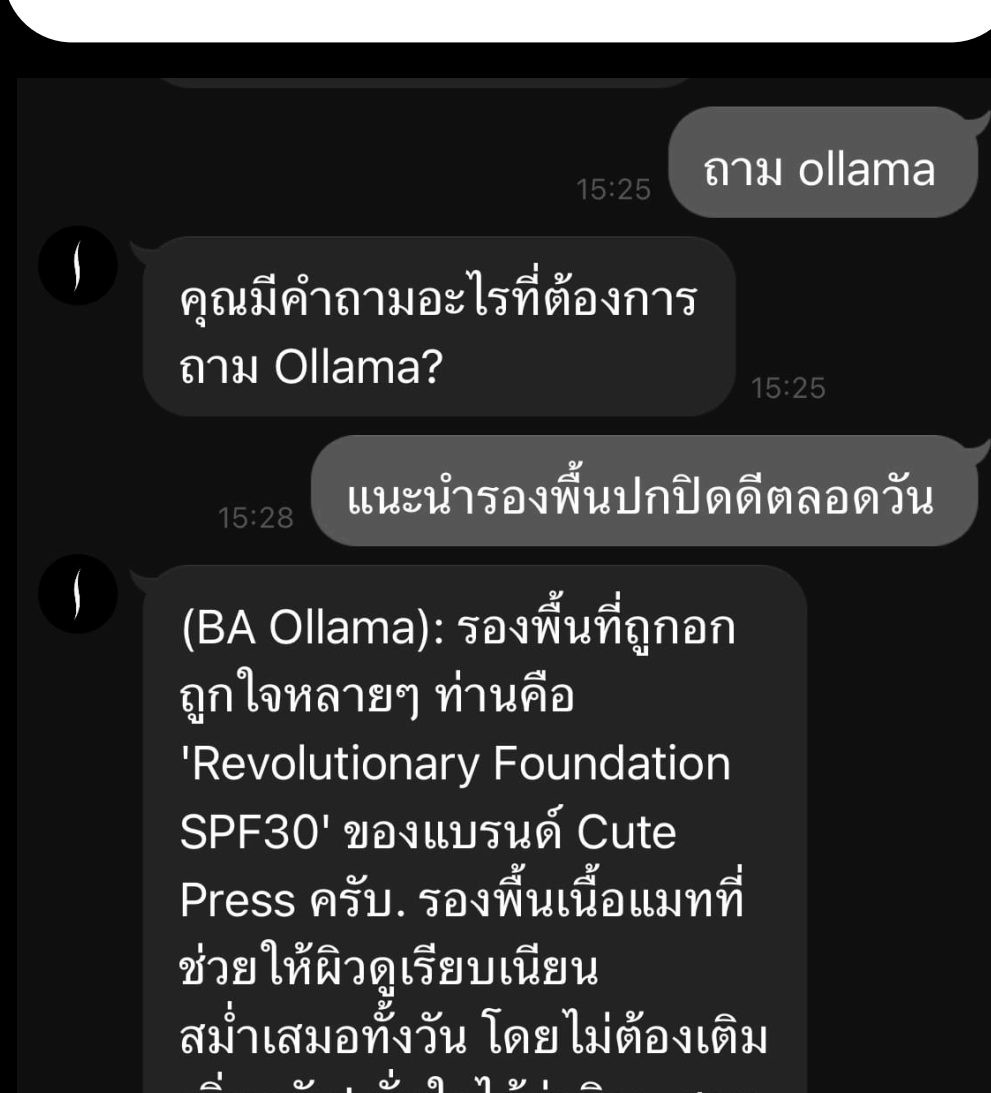
@app.route("/webhook", methods=['POST'])
def webhook():
    body = request.get_data(as_text=True)
    signature = request.headers['X-Line-Signature']

    try:
        handler.handle(body, signature)
    except InvalidSignatureError:
        return "Invalid signature", 400

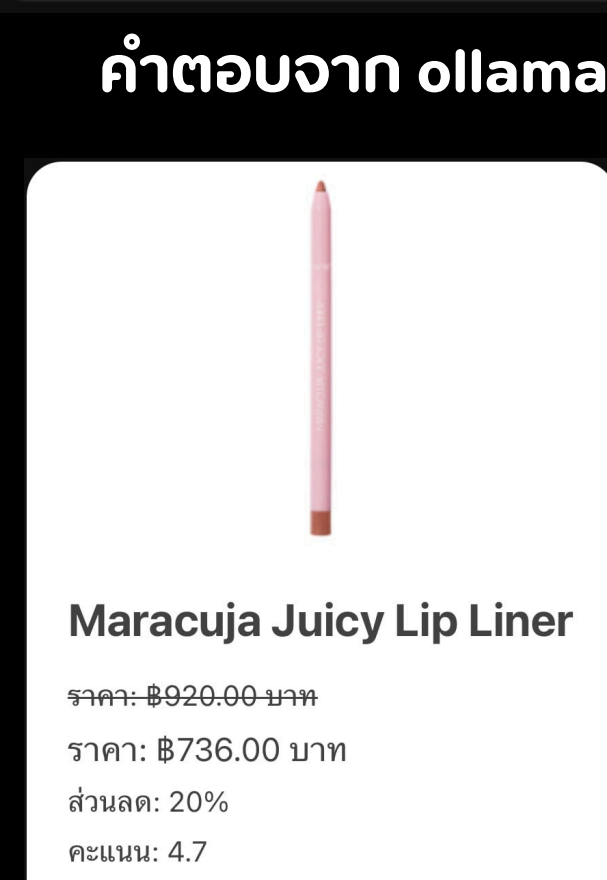
    return 'OK'

if __name__ == "__main__":
    app.run(port=5000)
```

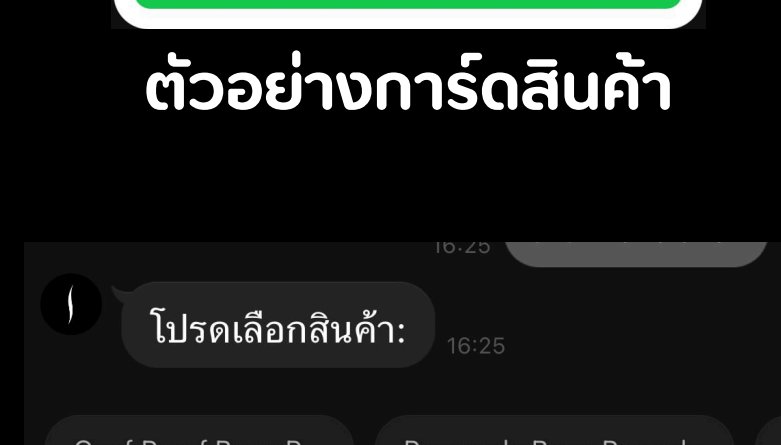
ตัวอย่างบทสนทนา



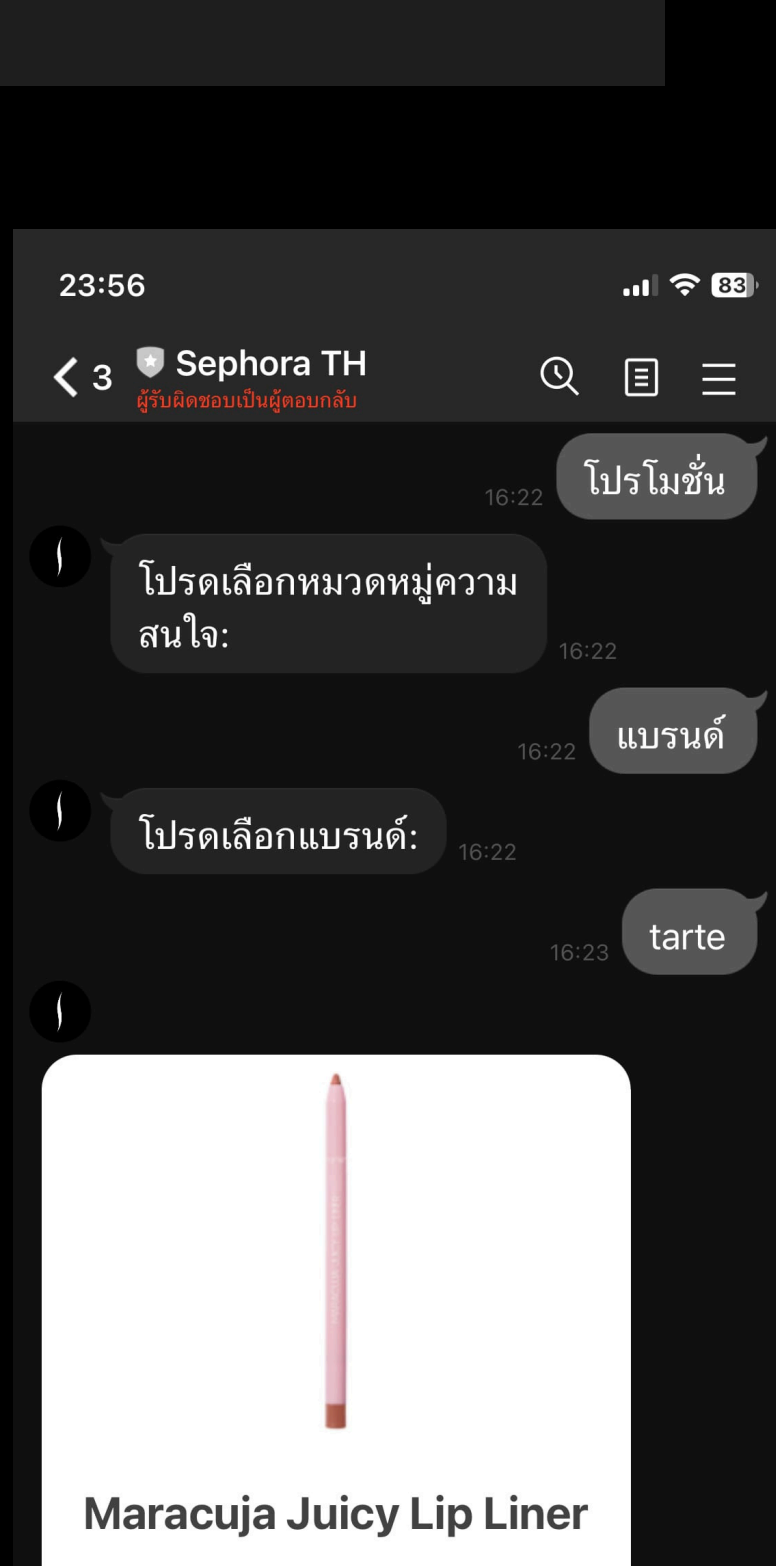
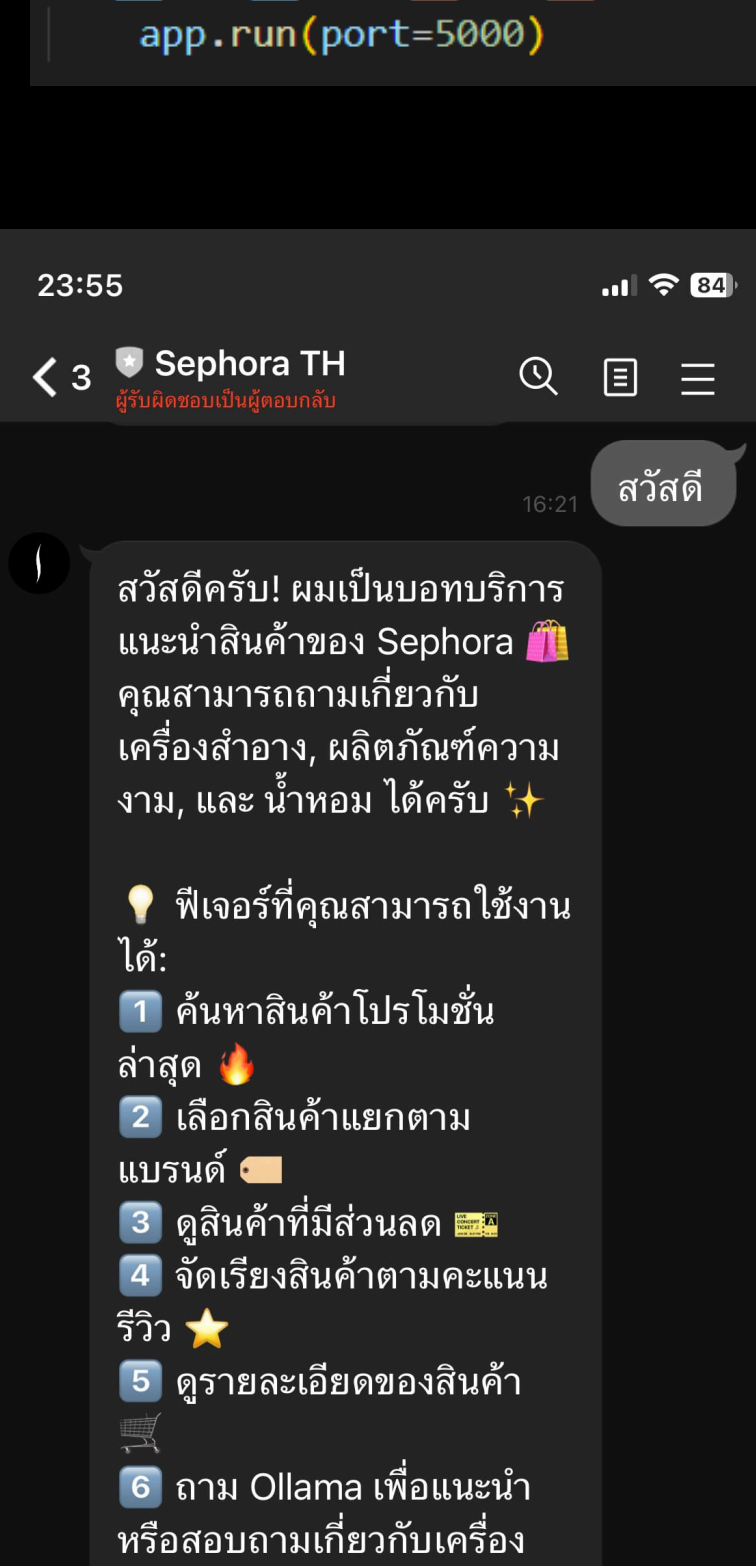
คำตอบจาก ollama



ตัวอย่างการ์ดสินค้า



ตัวอย่าง quick reply



โอกาสในการพัฒนา

- **Real-Time Product Updates:** เชื่อมต่อกับระบบข้อมูลผลิตภัณฑ์ของ Sephora แบบ real-time เพื่อให้แขกบอกสามารถดึงข้อมูลผลิตภัณฑ์โปรโมชั่น และคะแนนรีวิวดั้แบบอัตโนมัติ และแสดงข้อมูลที่เป็นปัจจุบันที่สุด
- **User Behavior Analytics:** เก็บข้อมูลพฤติกรรมของผู้ใช้ เช่น การค้นหาผลิตภัณฑ์และการสนทนา จากนั้นวิเคราะห์เพื่อสร้างประสบการณ์เฉพาะบุคคล แขกบอกสามารถแนะนำผลิตภัณฑ์ที่ตรงกับความต้องการได้โดยอิงจากพฤติกรรมก่อนหน้า
- **Sentiment Analysis:** ใช้การวิเคราะห์อารมณ์ (Sentiment Analysis) เพื่อตรวจสอบอารมณ์ของผู้ใช้ในบทสนทนา และปรับโทนการตอบสนองเพื่อให้ผู้ใช้รู้สึกว่าได้รับการดูแล

<https://github.com/madmax-mc/project02-Chatbot-Recommendation.git>

