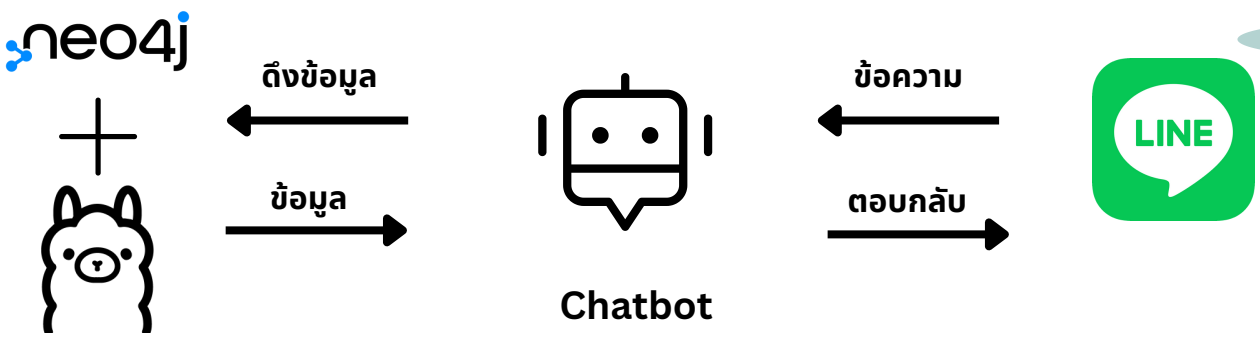
นายเจตนิพัทธ์ แสงค่าณี 6410110081

AI CHATBOT KIRITO_VINEERADVISER

ที่มาและความสำคัญ

การทำ veneer และการเคลือบฟันเป็นที่นิยมในปัจจุบันVeneerAdviser จึงถูกพัฒนาขึ้นเพื่อตอบคำถามเกี่ยวกับการทำ veneer ได้ตลอด 24 ชั่วโมง ช่วยให้ข้อมูลที่ถูกต้องและสะดวกในการเข้าถึง ลดภาระของทันตแพทย์ เพิ่มความมั่นใจให้กับผู้สนใจ และส่งเสริมการดูแลฟันอย่างมีประสิทธิภาพ

โครงสร้าง CHATBOT



การตั้งค่าเริ่มต้น

Python Library

- pandas
- faiss
- numpy
- neo4j
- flask
  import Flask, request, jsonify
- LineBotApi
- WebhookHandler
- linebot.exceptions import InvalidSignatureError
- linebot.models import MessageEvent, TextMessage, TextSendMessage
- SentenceTransformer, util
- requests
- json

Model

- SentenceTransformer('sentence-transformers/distiluse-base-multilingual-cased-v2')

การเริ่มต้นการเชื่อมต่อ

Config Neo4j

```
URI = "neo4j://localhost:7687"
AUTH = ("neo4j", "1          _  ")
```

```
cypher_query_greeting = '''
MATCH (n:Greeting) RETURN n.name as name, n.msg_reply as reply;
'''
greeting_corpus = []
greeting_replies = {}
results = run_query(cypher_query_greeting)
for record in results:
    greeting_corpus.append(record['name'])
    greeting_replies[record['name']] = record['reply']

greeting_corpus = list(set(greeting_corpus))  # เอาข้อความมาใส่ใน corpus
print(greeting_corpus)
```

```
cypher_query_question = '''
MATCH (n:Question) RETURN n.question as question, n.answer as answer;
'''
question_corpus = []
question_replies = {}
results = run_query(cypher_query_question)
for record in results:
    question_corpus.append(record['question'])
    question_replies[record['question']] = record['answer']

question_corpus = list(set(question_corpus))  # เอาข้อความมาใส่ใน corpus
print(question_corpus)
```

Config ngrok

```
PS C:\Users\OOM> ngrok config add-authtoken 2kE                                HB
Authtoken saved to configuration file: C:\Users\OOM
```

```
PS C:\Users\OOM> ngrok http http://localhost:5000
```

```
Session Status                online
Account                       Jatnipat (Plan: Free)
Update                        update available (version 3.17.0, Ctrl-U to update)
Version                       3.14.1
Region                        Asia Pacific (ap)
Web Interface                 http://127.0.0.1:4040
Forwarding                                                  -free.app -> http://localhost:500

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

Config Flask

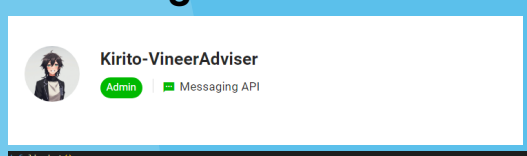
```
# สร้าง Flask app
app = Flask(__name__)
```

```
if __name__ == '__main__':
    # For Debugging
    app.run(port=5000)
```

Config Ollama

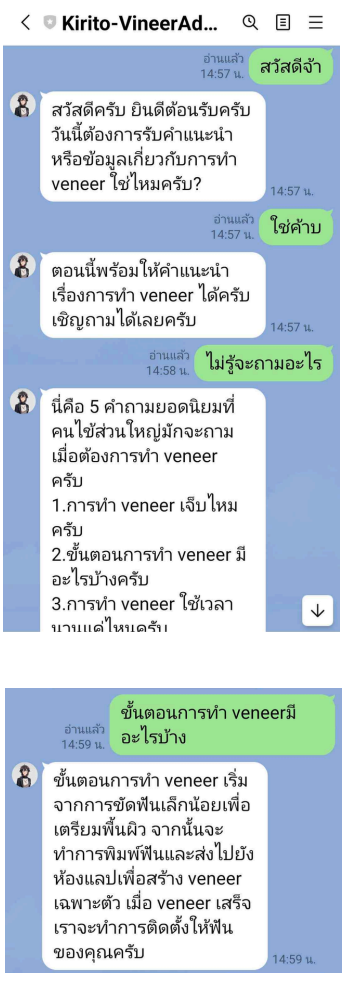
```
OLLAMA_API_URL = "http://localhost:11434/api/generate"

headers = {
    "Content-Type": "application/json"
}

def llama_generate_response(prompt):
    # Prepare the request payload for the supachai/llama-3-typhoon-v1.5 model
    payload = {
        "model": "supachai/llama-3-typhoon-v1.5",  # Adjust model name as needed
        "prompt": prompt+"ตอบไม่เกิน 20 คำและเป็นภาษาไทยเท่านั้น",
        "stream": False,
        "max_tokens": 60  # Limit the response to 100 tokens
    }

    # Send the POST request to the Ollama API
    response = requests.post(OLLAMA_API_URL, headers=headers, data=json.dumps(payload))
```

Config Line



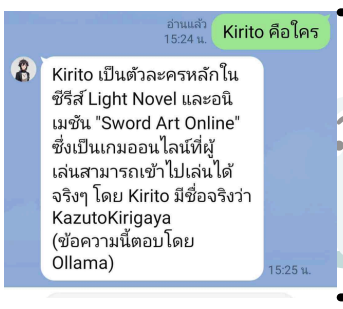

```
def linebot():
    body = request.get_data(as_text=True)  # รับข้อมูลจาก Line API
    try:
        json_data = json.loads(body)  # แปลงข้อมูลที่รับมาเป็น JSON
        access_token = '/1Khrjr                                                    5oZTos
        secret = '                                    '
        line_bot_api = LineBotApi(access_token)
        handler = WebhookHandler(secret)
        signature = request.headers['X-Line-Signature']
        handler.handle(body, signature)
```

ตัวอย่างการทำงาน



ทักทาย

ขอคำแนะนำ



Ollama

สอบถาม

การพัฒนาต่อ

- พัฒนาความเร็วในการตอบโดยหาโมเดลและวิธีการหมาะสม
- เพิ่มฟีเจอร์เรื่องการตอบโต้และแจ้งเตือนหากมีโปรโมชั่นต่างๆ
- พัฒนาให้สามารถวิเคราะห์พฤติกรรมและประวัติการใช้งานของผู้ใช้ เพื่อให้คำแนะนำที่เฉพาะเจาะจงมากขึ้น
- เพิ่มฟีเจอร์การแสดงภาพ วีดีโอเพื่ออธิบายขั้นตอนการทำ Vineer เพื่อให้ผู้ใช้เข้าใจมากยิ่งขึ้น

Github

https://github.com/JatnipatSangkanee/Ai-Chatbot-Veneer