



# ผู้เชี่ยวชาญทันตกรรม



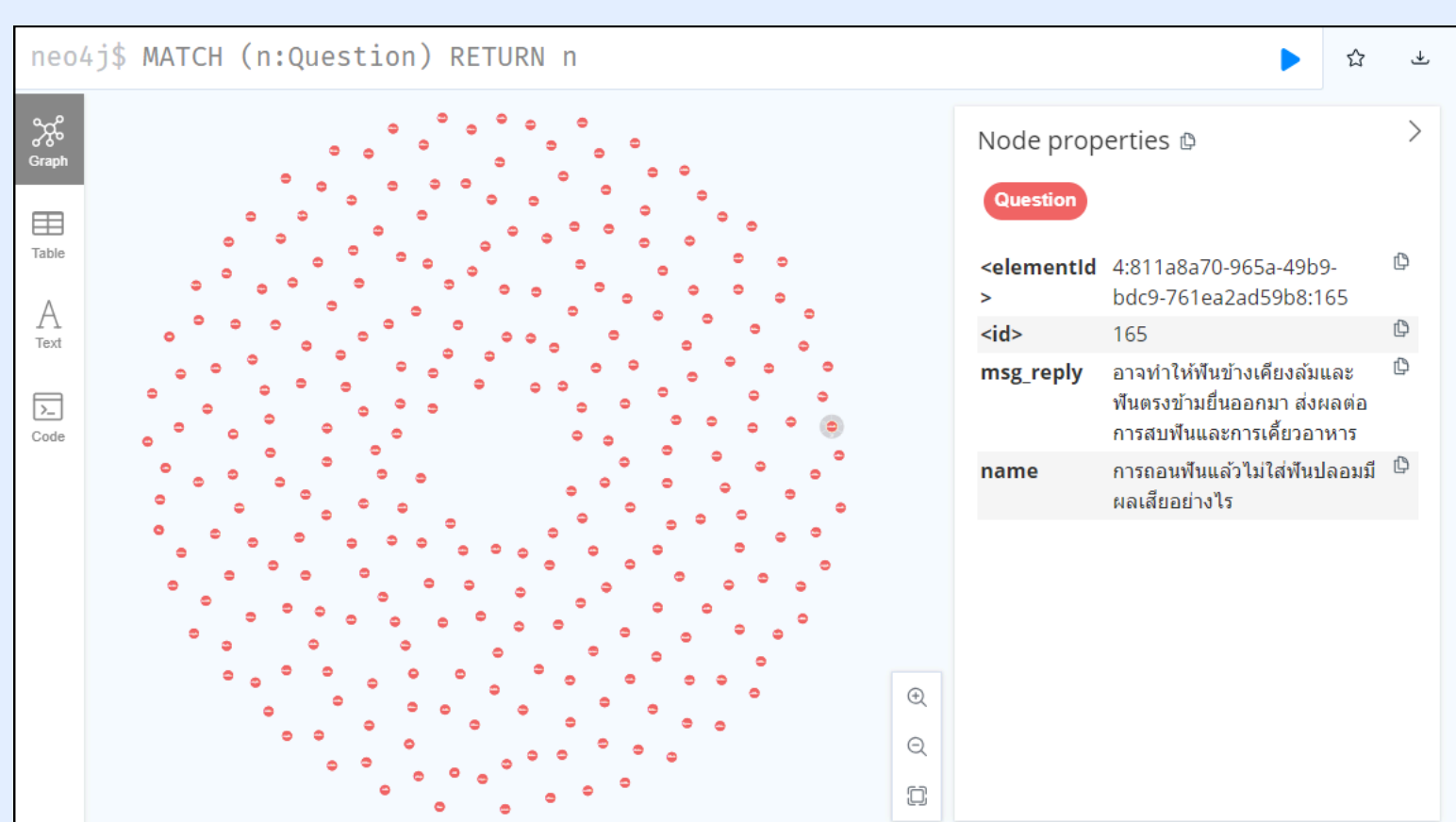
# DentChat



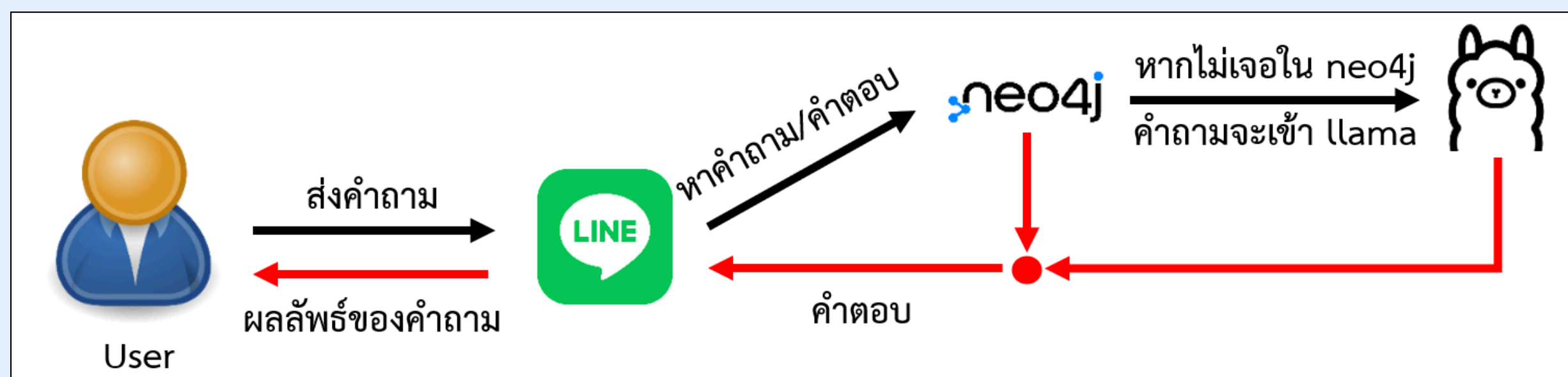
## ที่มาและความสำคัญ



แชทบอทที่ให้คำปรึกษาเกี่ยวกับทันตกรรมเบื้องต้นเป็นเครื่องมือที่ช่วยให้ผู้ใช้สามารถเข้าถึงข้อมูลด้านการดูแลฟันได้อย่างสะดวกและรวดเร็ว ไม่ว่าจะเป็นการดูแลฟัน การรักษารากฟัน ถอนฟัน ผ่าฟันคุด หรือการฟอกสีฟัน แชทบอทนี้ช่วยลดข้อจำกัดในการรอฟบทันตแพทย์และให้ข้อมูลเบื้องต้นที่แม่นยำ ผู้ใช้สามารถถามคำถามและรับคำตอบได้ตลอดเวลา ทำให้มีความมั่นใจมากขึ้นก่อนเข้ารับการรักษาจริง ทั้งยังส่งเสริมการดูแลสุขภาพฟันที่ถูกต้องวิธีในชีวิตประจำวัน



## การทำงาน Chatbot และ Neo4j



- FAISS
- FastAPI
- Pydantic
- Requests
- Pyngrok
- Flask
- LINE Bot SDK
- Sentence Transformers
- Torch
- Neo4j
- Numpy
- JSON
- Time
- model = SentenceTransformer('sentence-transformers/distiluse-base-multilingual-cased-v2')

```

cypher_query = '''
MATCH (n) WHERE (n:Question) RETURN n.name as name, n.msg_reply as reply;
'''
input_corpus = []
results = run_query(cypher_query)
for record in results:
    input_corpus.append(record['name'])
input_corpus = list(set(input_corpus))
print(input_corpus)
  
```

URI = "neo4j://localhost"  
AUTH = ("neo4j", "password")

Neo4j

```

def llama_response(msg):
    print("====this is llama====")
    payload = {
        "model": llama_model,
        "prompt": f"{msg} ตอบเป็นภาษาไทยใช้ข้อมูลวิชาการ ตอบสั้นๆไม่เกิน 20 คำ",
        "stream": False
    }
    response = requests.post(url, headers=headers, data=json.dumps(payload))
    if response.status_code == 200:
        res_JSON = json.loads(response.text)
        res_text = res_JSON["response"]
        return f"(ตอบโดย Ollama): {res_text}"
    else:
        print("error:", response.status_code, response.text)
        return "ขอภัย ไม่สามารถเชื่อมต่อกับเซิร์ฟเวอร์ได้"
  
```

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

Llama

```

Ngrok
port = "5000"
ngrok.set_auth_token("-----")
public_url = ngrok.connect(port).public_url
# Open a ngrok tunnel to the HTTP server
print(f"ngrok tunnel {public_url} -> http://127.0.0.1:{port}")
  
```

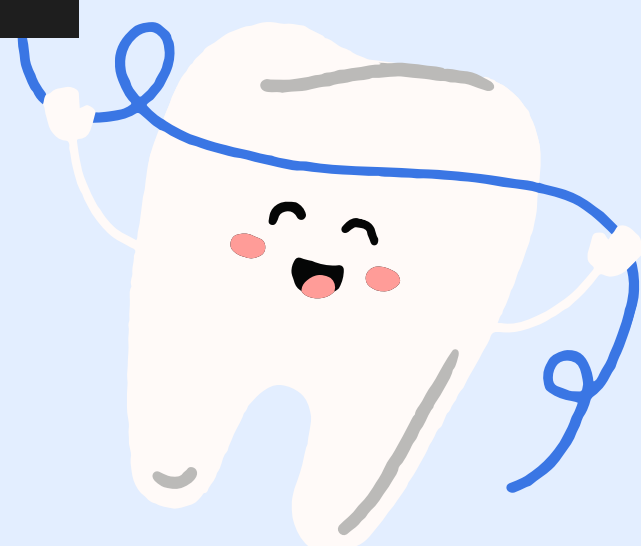
```

app = Flask(__name__)
app.config["BASE_URL"] = public_url
app.config["JSON_AS_ASCII"] = False

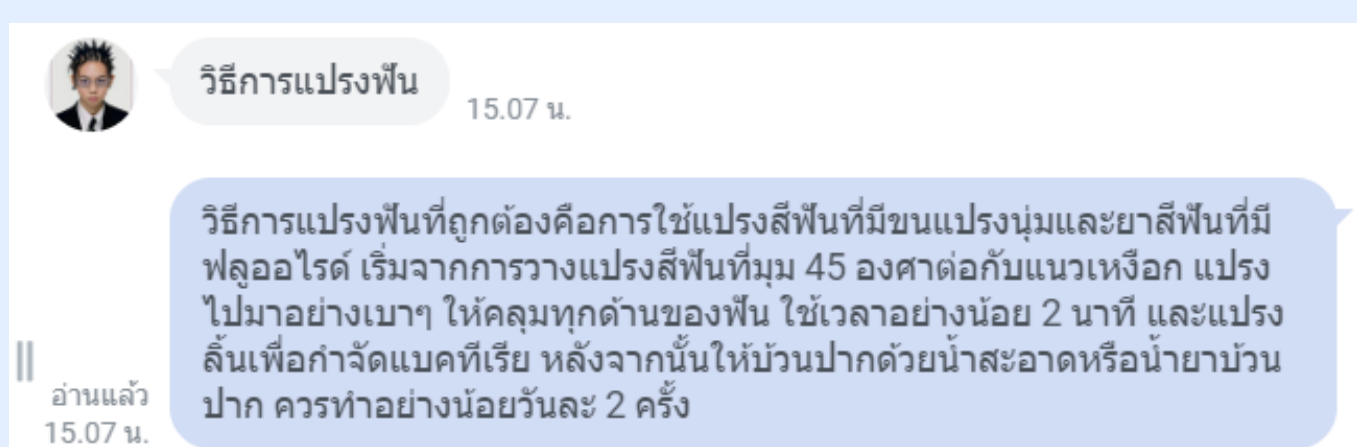
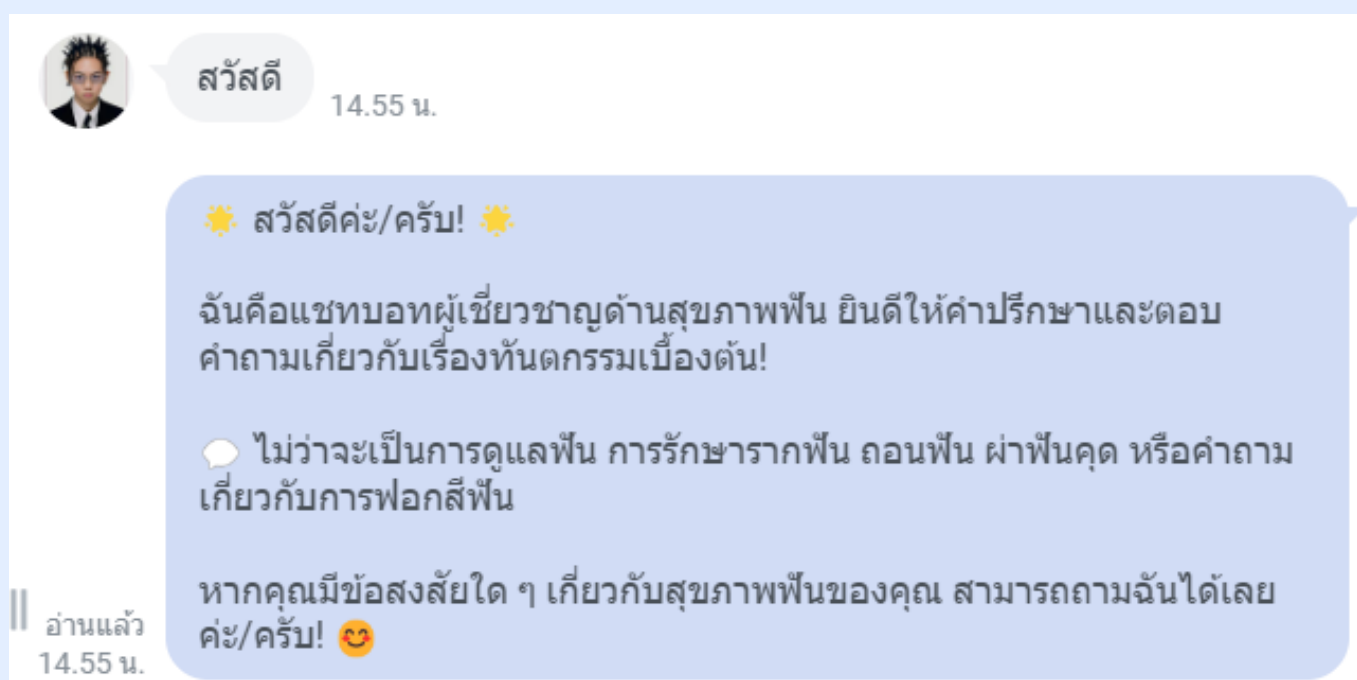
if __name__ == '__main__':
    app.run(port=5000)
  
```

```

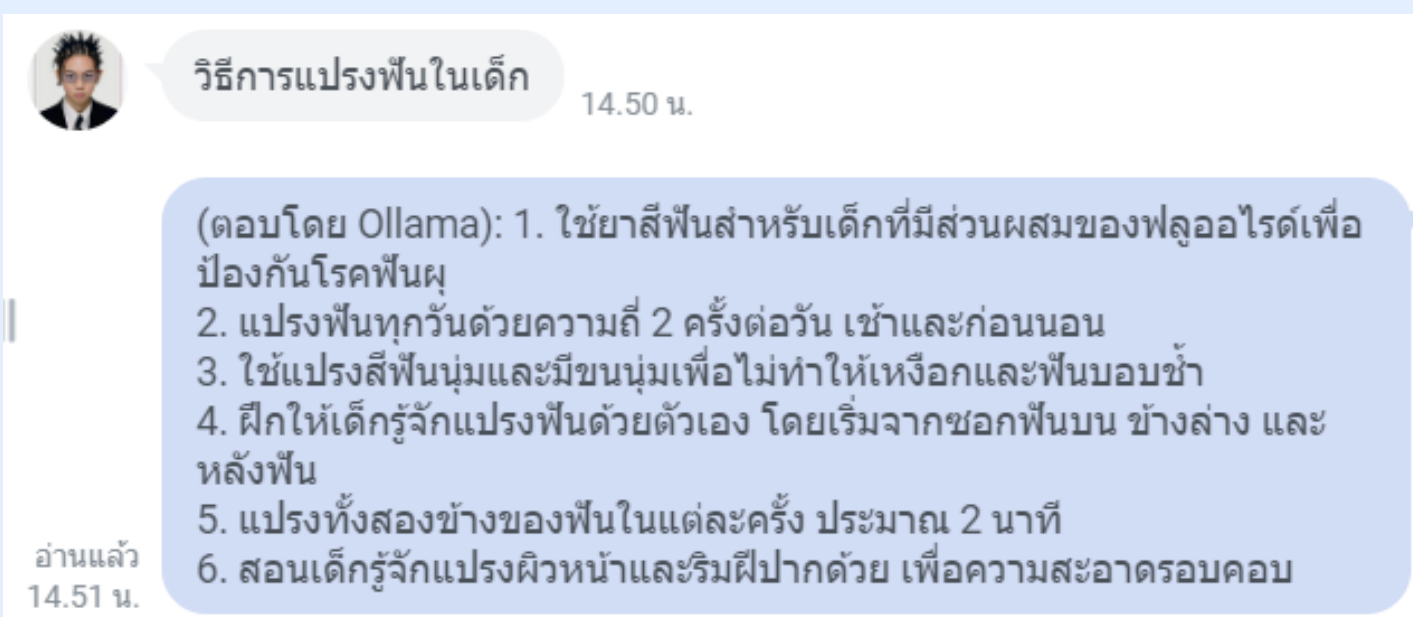
LINE
@app.route("/chat", methods=['POST'])
def linebot():
    body = request.get_data(as_text=True)
    try: # check if request from LINE
        json_data = json.loads(body)
        access_token = '-----'
        secret = '-----'
        line_bot_api = LineBotApi(access_token)
        handler = WebhookHandler(secret)
  
```



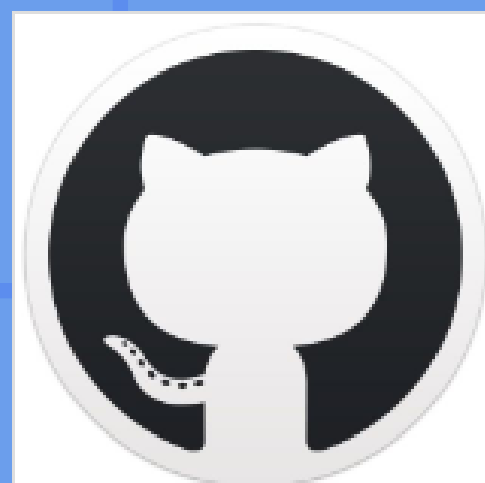
## ตัวอย่างคำตอบจาก Neo4j



## ตัวอย่างคำตอบจาก Ollama



- จัดการบริบทของแชทบอทในทสนทนา เพื่อให้แชทบอทเข้าใจการโต้ตอบแบบหลายประโยคได้ดียิ่งขึ้น และปรับปรุงการเก็บประวัติการสนทนาเพื่อใช้ประโยชน์จากข้อมูลเดิม
- เพิ่มฟีเจอร์การให้คำปรึกษาแบบเชิงลึก
- จัดการนัดหมายกับทันตแพทย์



**madmax-mc/project01-dentchat**  
 Contribute to madmax-mc/project01-dentchat development by creating an account on GitHub.  
 GitHub

<https://github.com/madmax-mc/project01-dentchat.git>