



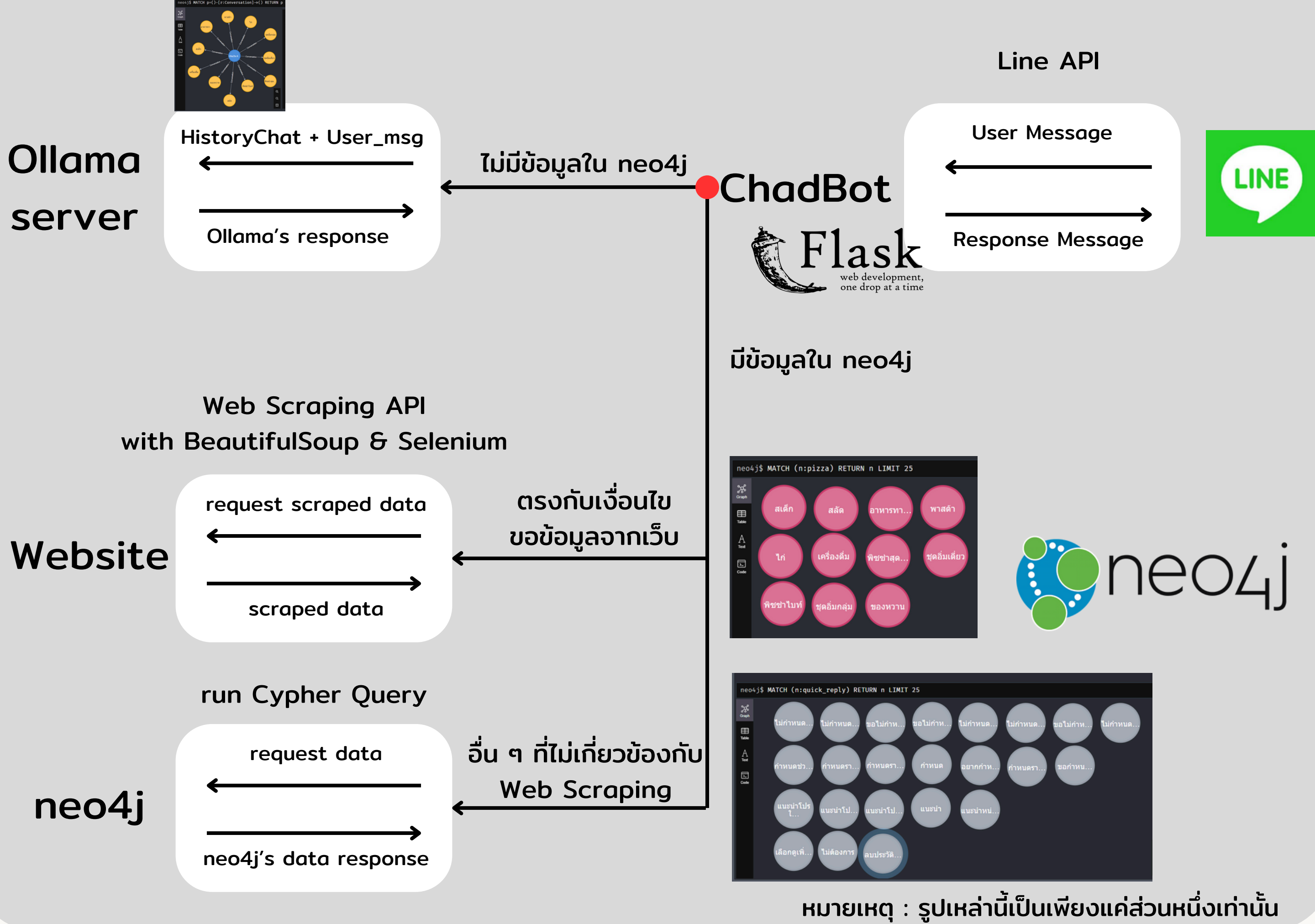
# ChadBot Pizza Company

## ที่มาและความสำคัญ



เนื่องจากสินค้าและโปรโมชั่นของร้าน pizza นั้นมีมากขึ้นเรื่อย ๆ การให้ตัวผู้ใช้มาเลื่อนหาข้อมูลให้ตัวเองนั้นอาจจะทำให้ยุ่งยาก ด้วยเหตุนี้จึงเกิดเป็น ChadBot ที่ทำงานผ่าน Line chatbot platform ที่ช่วยค้นหาสินค้า/โปรโมชั่น ตามหมวดและราคาตามความต้องการเพื่อความเหมาะสมของผู้ใช้

## โครงสร้าง ChatBot



## การตั้งค่าเริ่มต้นและการเริ่มต้นการเชื่อมต่อ

### Python Library

- faiss
- requests
- pyngrok import ngrok
- flask import Flask, request
- linebot import LineBotApi, WebhookHandler
- linebot.models import TextSendMessage, QuickReply, QuickReplyButton, MessageAction
- sentence\_transformers import SentenceTransformer
- neo4j import GraphDatabase
- import json
- bs4 import BeautifulSoup
- selenium import webdriver

### NLP Model

- sentence-transformers/distiluse-base-multilingual-cased-v2

### Configuration

#### Config ngrok

```
port = "5000"
ngrok.set_auth_token("<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}")
```



#### Config Line

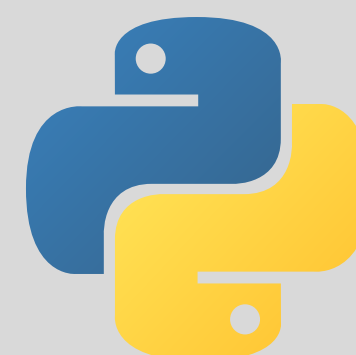
```
@app.route("/chat", methods=['POST'])
def linebot():
    body = request.get_data(as_text=True)
    try:
        # check if request from LINE
        json_data = json.loads(body)
        access_token = "<ACCESS_TOKEN>"
        secret = "<SECRET>"
        line_bot_api = LineBotApi(access_token)
        handler = WebhookHandler(secret)
        signature = request.headers['X-Line-Signature']
        handler.handle(body, signature)
        msg = json_data['events'][0]['message']['text']
        tk = json_data['events'][0]['replyToken']
        userid = json_data['events'][0]['source']['userId']
```



#### Config Flask

```
app = Flask(__name__)
app.config["BASE_URL"] = public_url
app.config["JSON_AS_ASCII"] = False
# app.config["SECRET_KEY"] = "secret"

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



#### Config Selenium

```
# setup chrome options
chrome_options = webdriver.ChromeOptions()
chrome_options.add_argument('--headless') # ensure GUI
chrome_options.add_argument('--no-sandbox')
chrome_options.add_argument('--disable-dev-shm-usage')

driver = webdriver.Chrome(options=chrome_options)
driver.get(url)
driver.implicitly_wait(5) # Wait for the page to load
html = driver.page_source
```



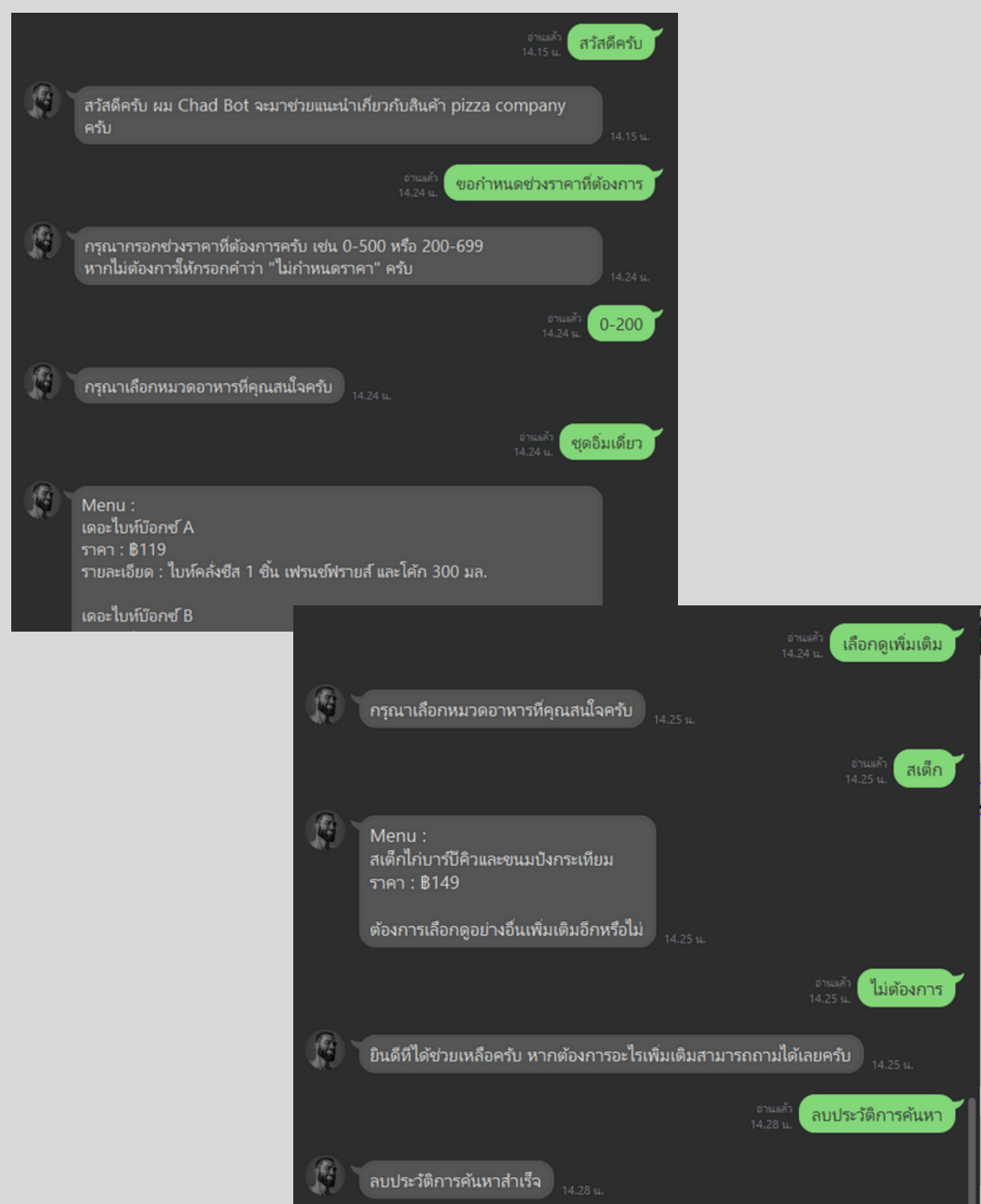
### Config neo4j with faiss

```
cypher_query = """
MATCH (n) WHERE (n:Greeting OR n:Grateful OR n:Goodbye OR n:pizza OR n:quick_reply) RETURN n.name as name, n.msg_reply as reply;
"""
input_corpus = []
results = run_query(cypher_query)
for record in results:
    print(record)
    input_corpus.append(record['name'])
input_corpus = list(set(input_corpus))
print(input_corpus)

# Encode the input corpus into vectors using the sentence transformer model
input_vecs = model.encode(input_corpus, convert_to_numpy=True, normalize_embeddings=True)

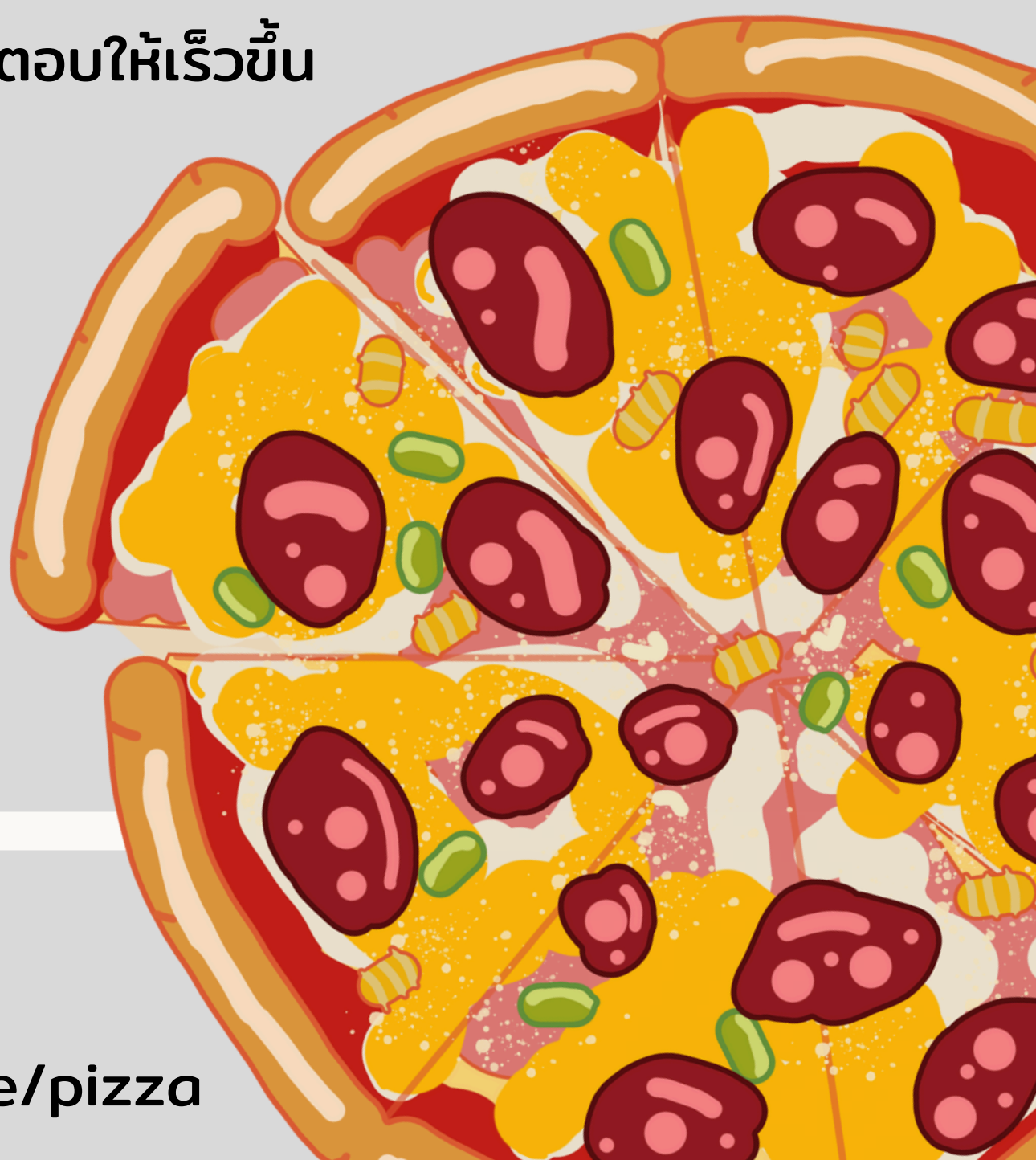
# Initialize FAISS index
d = input_vecs.shape[1] # Dimension of vectors
index = faiss.IndexFlatL2(d) # L2 distance index (cosine similarity can be used with normalization)
index.add(input_vecs) # Add vectors to FAISS index
```

## ตัวอย่างการทำงาน



## โอกาสในการพัฒนา

- เพิ่มตัวเลือกในการเลือกหน้าพิซซ่าเพิ่มขึ้น
- อาจจะมีการให้เลือกโปรโมชั่นต่าง ๆ ใส่ตะกร้าสินค้าแล้วรวมราคามาให้ผู้ใช้ เพื่อประกอบการตัดสินใจของผู้ใช้
- พัฒนาเพิ่มประสิทธิภาพความเร็วในการตอบให้เร็วขึ้น



## Python Code

<https://github.com/Poadkee/ChadBot/tree/pizza>