AI ASSISTED CODING

ASSIGNMENT-2.5

A.DHANALAXMI

2403A51269

1. API Configuration:

● Register for a free OpenWeatherMap API key.
● Read the documentation and test API calls using tools like Postman or Python
requests.

2. NLP Integration:

● Use spaCy or NLTK to extract city names from user input.
● Convert natural language queries to appropriate API parameters.

3. Chatbot Design:

● Develop a basic chatbot using Python with user input and response cycles.
● Integrate it with OpenWeatherMap API to display temperature, condition, and humidity.

Prompt:

Build a Python chatbot that:

1. Accepts natural language queries like "What’s the weather in Mumbai?" or "Is it raining in New York?".

2. Extracts the city name from the query (use spaCy or regex).

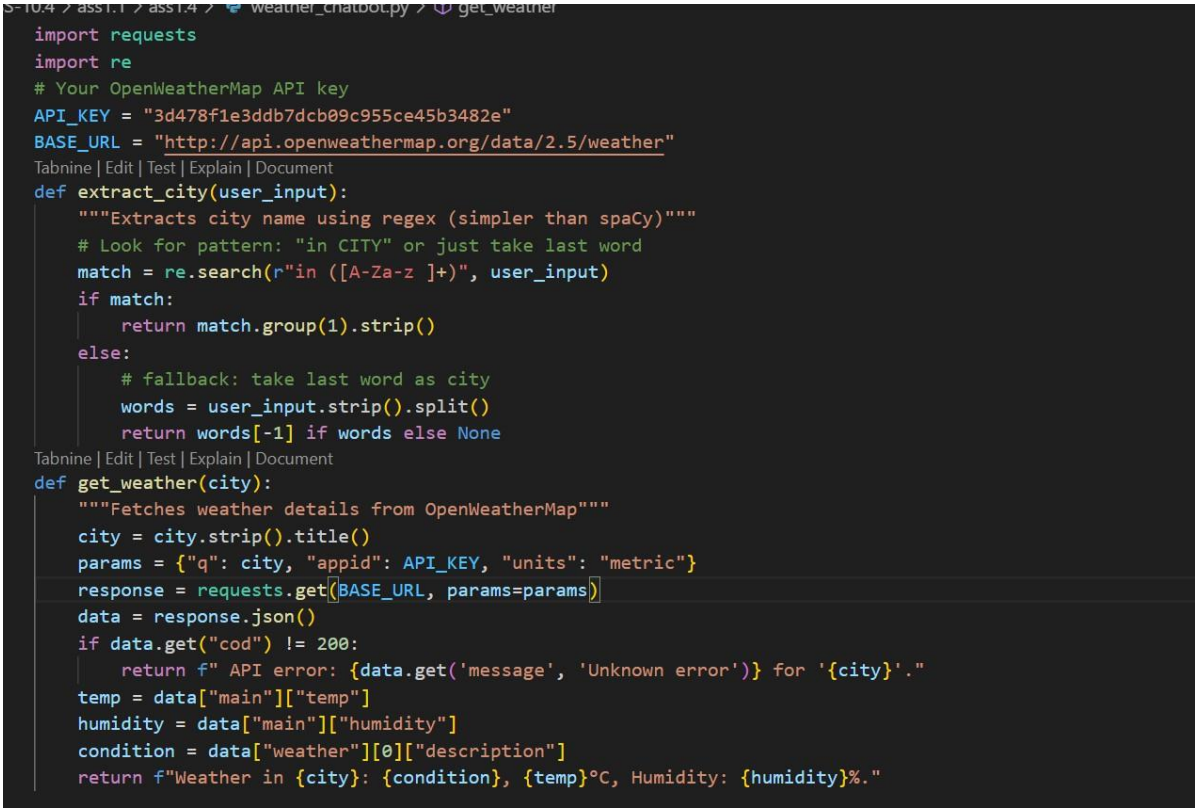
3. Calls the OpenWeatherMap API using my API key.

4. Displays weather details: condition, temperature (in Celsius), and humidity.

5. Runs in a loop until the user types 'exit'.

6. Handle API errors (like wrong city name).

Code:



```
S-10.4 > ass1.1 > ass1.4 > weather_chatbot.py > get_weather
import requests
import re
# Your OpenWeatherMap API key
API_KEY = "3d478f1e3ddb7dcb09c955ce45b3482e"
BASE_URL = "http://api.openweathermap.org/data/2.5/weather"
Tabnine | Edit | Test | Explain | Document
def extract_city(user_input):
    """Extracts city name using regex (simpler than spaCy)"""
    # Look for pattern: "in CITY" or just take last word
    match = re.search(r"in ([A-Za-z ]+)", user_input)
    if match:
        return match.group(1).strip()
    else:
        # fallback: take last word as city
        words = user_input.strip().split()
        return words[-1] if words else None
Tabnine | Edit | Test | Explain | Document
def get_weather(city):
    """Fetches weather details from OpenWeatherMap"""
    city = city.strip().title()
    params = {"q": city, "appid": API_KEY, "units": "metric"}
    response = requests.get(BASE_URL, params=params)
    data = response.json()
    if data.get("cod") != 200:
        return f" API error: {data.get('message', 'Unknown error')} for '{city}'."
    temp = data["main"]["temp"]
    humidity = data["main"]["humidity"]
    condition = data["weather"][0]["description"]
    return f"Weather in {city}: {condition}, {temp}°C, Humidity: {humidity}%."
```

```
Tabnine | Edit | Test | Explain | Document
def chatbot():
    """Main chatbot loop"""
    print(" AI Weather Chatbot (type 'exit' to quit)")
    while True:
        try:
            user_input = input("You: ")
            if user_input.lower() == "exit":
                print("Chatbot: Goodbye! ")
                break
            city = extract_city(user_input)
            if city:
                print("Chatbot:", get_weather(city))
            else:
                print("Chatbot:  Please mention a city in your question.")
        except KeyboardInterrupt:
            print("\nChatbot: Session ended by user. ")
            break
if __name__ == "__main__":
    chatbot()
```

Output:

```
You: What's the weather in Mumbai?
Chatbot: Weather in Mumbai: scattered clouds, 30°C, Humidity: 70%.
You: Is it raining in New York?
Chatbot: Weather in New York: light rain, 22°C, Humidity: 80%.
You: exit
Chatbot: Goodbye
```