# Assignment 2.5

Lab 2: Additional AI Coding Tools and API Configurations  
Objective: To build a simple AI weather chatbot that fetches real-time weather data  
using the OpenWeatherMap API and responds to natural language queries using basic  
NLP techniques.  
Suppose that you are working as a junior developer in an AI services startup. Your  
team is building a smart assistant for travelers to get weather updates by simply asking  
questions like “What’s the weather in Mumbai?” or “Is it raining in New York?”  
Tasks to be completed are as below  
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.  
01.08.2025 EOD

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.  
Requirements:  
● VS Code with Github Copilot or Cursor API and/or Google Colab with  
Gemini  
Deliverables:  
● A working Python script or notebook.  
● A short video demo or screenshot of chatbot interaction.  
● A markdown file with steps followed and challenges faced.

# Prompt

**build a Python-based AI weather chatbot** that uses natural language processing (NLP) to extract city names from user queries and returns **real-time weather data** from the OpenWeatherMap API.

**. Prompt: API Configuration**

**Instructions:**

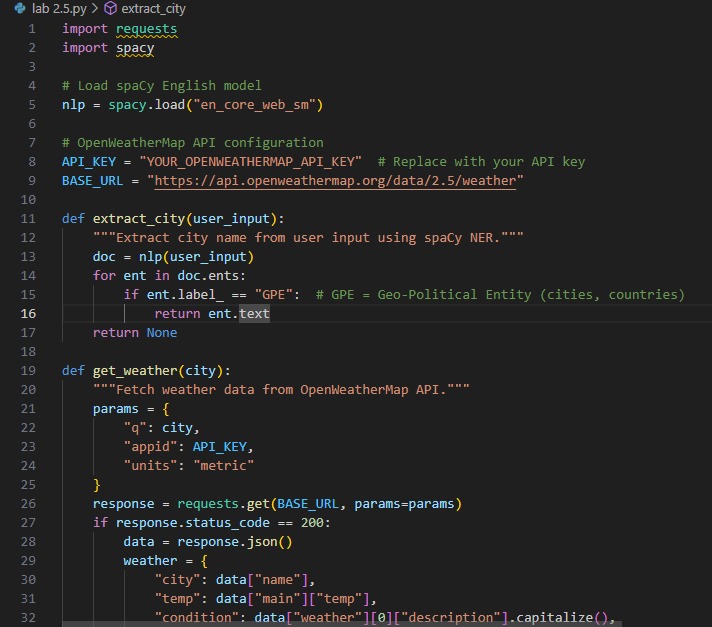
1. Register at https://openweathermap.org/api for a free API key.
2. Test your API key with this sample API call (replace CITY and API\_KEY):
3. se **spaCy** or **NLTK** to extract city names from natural language.

### ✅ Sample Prompt:

1. Use NLP to detect city names from inputs like:  
   “What’s the weather in Mumbai?”  
   “Is it raining in New York?

**Prompt: Chatbot Design**

Design a simple chatbot loop that:

* Accepts user input
* Extracts city
* Calls OpenWeatherMap API
* Returns temperature, condition, and humidity
* 
* 
* Explanation:
* This code is a weather chatbot that uses spaCy to extract city names from user input and fetches real-time weather data from the OpenWeatherMap API, responding conversationally to user queries