MINIWORLD TECHNOLOGY

- Maganti Murali Krishna
- muralikrishna1828@gmail.com

Task:

Build a Weather Chatbot

Description:

Create a chatbot that can provide users with weather information for a given location.

Requirements:

1. The chatbot should be able to understand user queries about the weather, such as "What's the weather like in New York today?" or "Will it rain in London tomorrow?"

2. Integrate an API or a weather service to retrieve weather data based on user queries.

3. Parse the user's query to extract the location and date for which the weather information is requested.

4. Display the weather information in a user-friendly format, including details like temperature, humidity, wind speed, and weather conditions (e.g., sunny, cloudy, rainy).

5. Handle cases where the location is not recognized or the weather data is unavailable.

6. Implement error handling and appropriate error messages for any failures or unexpected situations.

7. Test the chatbot with various weather queries to ensure accurate and relevant responses.

8. Optionally, you can incorporate natural language processing (NLP) techniques to improve the chatbot's understanding of user queries.

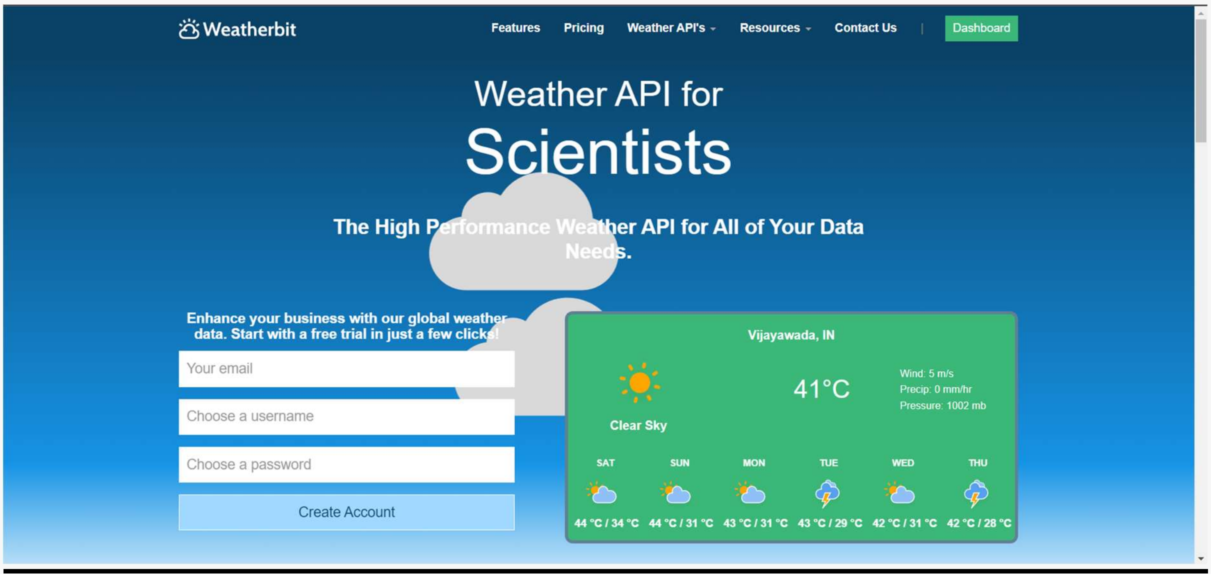
Introduction:

The Weather Information Chatbot is a program designed to provide users with weather information for a specific location. It integrates with the OpenWeatherMap API to retrieve weather data based on user queries. The chatbot is built using Python and utilizes natural language processing techniques to understand user queries.

Requirements:

1. Python 3.x
2. Requests library (`pip install requests`)
3. Weatherbit API (https://www.weatherbit.io/)
4. IDE: Google Colab

Weatherbit.io Interface:



Weather API for Scientists

The High Performance Weather API for All of Your Data Needs.

Enhance your business with our global weather data. Start with a free trial in just a few clicks!

Vijayawada, IN

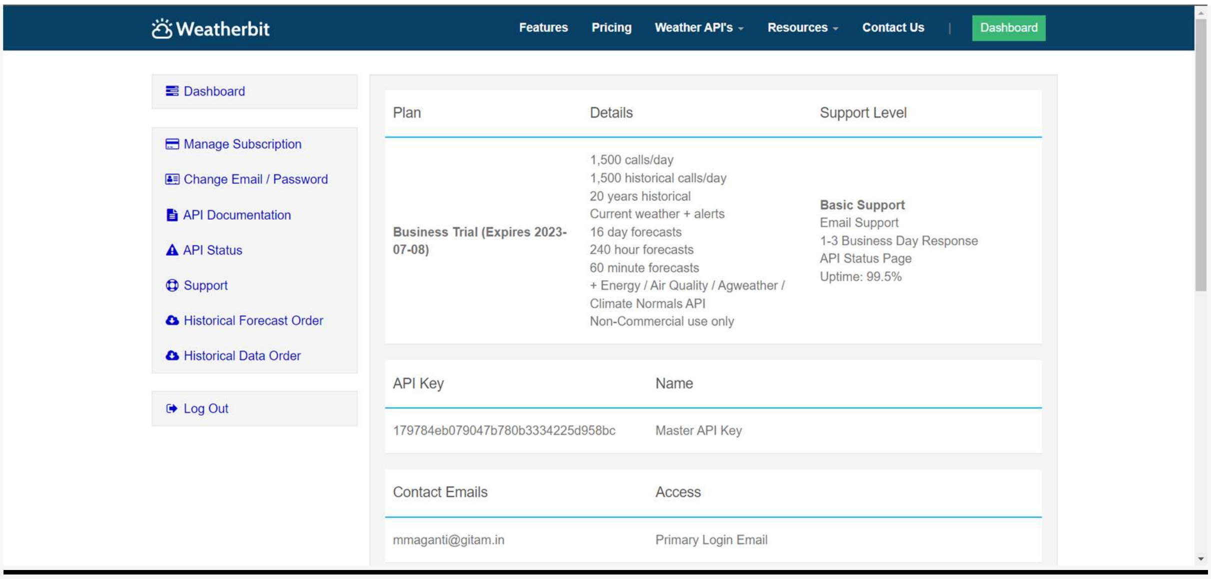
41°C

Wind: 5 m/s
Precip: 0 mm/hr
Pressure: 1002 mb

Clear Sky

SAT	SUN	MON	TUE	WED	THU
44 °C / 34 °C	44 °C / 31 °C	43 °C / 31 °C	43 °C / 29 °C	42 °C / 31 °C	42 °C / 28 °C

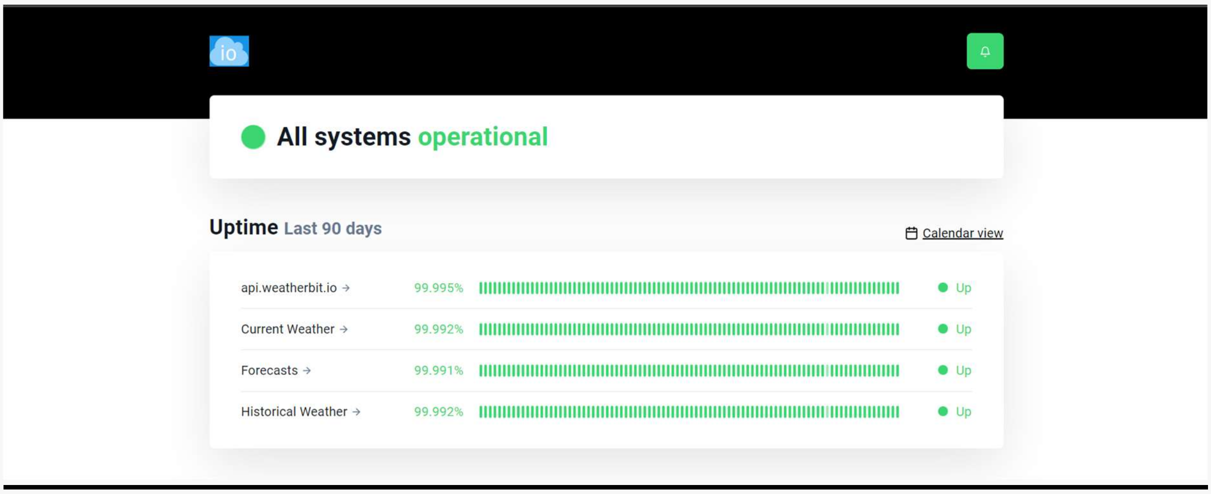
- Create an account in Weatherbit.io
- Open Dashboard there we will see our API KEY
- After acquiring the key paste the key in code.
- We can check the status of our API key in the dashboard



Plan	Details	Support Level
Business Trial (Expires 2023-07-08)	1,500 calls/day 1,500 historical calls/day 20 years historical Current weather + alerts 16 day forecasts 240 hour forecasts 60 minute forecasts + Energy / Air Quality / Agweather / Climate Normals API Non-Commercial use only	**Basic Support** Email Support 1-3 Business Day Response API Status Page Uptime: 99.5%

API Key	Name
179784eb079047b780b3334225d958bc	Master API Key

Contact Emails	Access
mmaganti@gitam.in	Primary Login Email



All systems operational

Uptime Last 90 days

api.weatherbit.io →	99.995%	Up
Current Weather →	99.992%	Up
Forecasts →	99.991%	Up
Historical Weather →	99.992%	Up

Setup and Usage

1. Obtain a Weatherbit.io API key by signing up on their website.

2. Install the required dependencies by running `pip install requests` in your terminal.

4. Open the chatbot code in a text editor or IDE of your choice.

5. Locate the `API_KEY` variable and replace `"YOUR_API_KEY"` with your actual Weatherbit.io API key.

6. Save the changes in the code file.

7. Run the chatbot by executing the script in your Python environment (`python chatbot.py`).

8. The chatbot will greet you and prompt you to enter a location for weather information.

9. Enter a location, such as a city name, and press Enter.

10. The chatbot will display the weather information for the specified location, including temperature, humidity, wind speed, and weather conditions.

11. To exit the chatbot, type 'exit' and press Enter.

Project Source Code:

```
import requests

# Weatherbit API endpoint and API key
API_ENDPOINT = "https://api.weatherbit.io/v2.0/current"
API_KEY = "179784eb079047b780b3334225d958bc"

def get_weather(location):
    # Make a GET request to the Weatherbit API
    params = {
        "city": location,
        "key": API_KEY,
        "units": "M"
    }
    response = requests.get(API_ENDPOINT, params=params)

    # Check if the request was successful
    if response.status_code == 200:
        weather_data = response.json()
        return weather_data
    else:
        return None

def parse_weather_data(weather_data):
    # Extract relevant weather information from the API response
    # Adjust the parsing logic based on the Weatherbit API response
structure
    temperature = weather_data["data"][0]["temp"]
    humidity = weather_data["data"][0]["rh"]
    wind_speed = weather_data["data"][0]["wind_spd"]
    weather_conditions =
weather_data["data"][0]["weather"]["description"]
    rain = "rain" in weather_data["data"][0]

    return temperature, humidity, wind_speed, weather_conditions, rain
```

```python
def format_weather_response(location, temperature, humidity,
wind_speed, weather_conditions, rain):
    # Format the weather information into a user-friendly response
    response = f"Weather information for {location}:"
    response += f"\nTemperature: {temperature}°C"
    response += f"\nHumidity: {humidity}%"
    response += f"\nWind Speed: {wind_speed} m/s"
    response += f"\nWeather Conditions: {weather_conditions}"

    if rain:
        response += "\nIt will rain today."
    else:
        response += "\nNo rain is expected today."

    return response

def chatbot():
    print("Welcome to the Weather Information Chatbot!")
    print("You can ask for the weather by typing a location.")
    print("To exit, simply type 'exit'.\n")

    while True:
        user_input = input("Please enter a location: ")

        if user_input.lower() == "exit":
            print("Goodbye!")
            break

        weather_data = get_weather(user_input)

        if weather_data is not None and "data" in weather_data and
len(weather_data["data"]) > 0:
            temperature, humidity, wind_speed, weather_conditions, rain
= parse_weather_data(weather_data)

            response = format_weather_response(user_input, temperature,
humidity, wind_speed, weather_conditions, rain)
            print(response)
        else:
            print("Sorry, weather information for that location is
unavailable. Please try again.\n")

# Start the chatbot
chatbot()
```

Sample Output:

```
Welcome to the Weather Information Chatbot!
You can ask for the weather by typing a location.
To exit, simply type 'exit'.

Please enter a location: new delhi
Weather information for new delhi:
Temperature: 36°C
Humidity: 39%
Wind Speed: 5.1 m/s
Weather Conditions: Haze
No rain is expected today.
Please enter a location: London
Weather information for London:
Temperature: 19.4°C
Humidity: 57%
Wind Speed: 2.57 m/s
Weather Conditions: Few clouds
No rain is expected today.
Please enter a location: exit
Goodbye!
```

Error Handling

The chatbot handles the following error scenarios:

1. If the location is not recognized, the chatbot displays an error message indicating that weather information for that location is unavailable.

2. If there is a failure in retrieving weather data from the API, the chatbot displays an error message and suggests trying again.

Limitations

1. The chatbot relies on the Weatherbit.io API for weather data, and therefore, its availability and accuracy are dependent on the API's performance.

2. The chatbot may not handle complex queries or non-standard input formats.

3. The chatbot's response may be limited to basic weather information and may not cover all possible weather attributes or conditions.

Troubleshooting

If you encounter any issues or errors while using the Weather Information Chatbot, please check the following:

1. Ensure that you have a stable internet connection.

2. Verify that your Weatherbit.io API key is correctly set in the code.

3. Double-check that you have installed the `requests` library.

Github Repository link:

https://github.com/ALANWALKERMK7/internship-miniworld/tree/main

References:

- Open AI – ChatGpt
- Github
- Towardsdatascience - https://towardsdatascience.com/how-to-create-a-weather-chatbot-b8ef1b1d6703
- Twilio - https://www.twilio.com/blog/build-weather-chatbot-sms-python-flask
- Medium - https://medium.com/analytics-vidhya/building-a-simple-weather-chatbot-using-rasa-54eaf97daa82