**Step 1: Obtain an OpenWeatherMap API Key**

1. Go to the OpenWeatherMap website: <https://openweathermap.org/>
2. Sign up for a free account.
3. Once logged in, navigate to the "API keys" section (usually under your profile or settings).
4. Generate a new API key. Copy this key; you will need to paste it into the Python script later.

**Step 2: Install Required Python Libraries**

1. Open your terminal or command prompt.
2. Run the following command to install the necessary libraries:

Bash

pip install requests pandas matplotlib seaborn streamlit

This command will download and install requests (for making API calls), pandas (for data manipulation), matplotlib and seaborn (for creating visualizations), and streamlit (for building the web-based dashboard).

**Step 3: Create the Python Script (**weather\_dashboard.py**)**

1. Open a text editor or a code editor like VS Code.
2. Copy and paste the following Python code into the editor:

Python

import requests

import json

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import streamlit as st

# Replace with your actual API key

api\_key = "YOUR\_OPENWEATHERMAP\_API\_KEY"

city\_name = st.sidebar.text\_input("Enter City Name:", "Kolkata") # Default city in India

base\_url = "http://api.openweathermap.org/data/2.5/weather?"

complete\_url = base\_url + "appid=" + api\_key + "&q=" + city\_name + "&units=metric" # Use metric for Celsius

st.title(f"Weather Insights Dashboard for {city\_name.capitalize()}")

try:

response = requests.get(complete\_url)

response.raise\_for\_status()

data = response.json()

if data["cod"] != "404":

main\_data = data["main"]

wind\_data = data.get("wind", {})

weather\_data = {

"temperature": main\_data.get("temp"),

"feels\_like": main\_data.get("feels\_like"),

"humidity": main\_data.get("humidity"),

"pressure": main\_data.get("pressure"),

"wind\_speed": wind\_data.get("speed", 0),

"description": data["weather"][0].get("description", "").capitalize(),

"city": data["name"],

"country": data["sys"].get("country")

}

st.subheader("Current Weather Overview:")

st.write(f"\*\*City:\*\* {weather\_data['city']}, {weather\_data['country']}")

st.write(f"\*\*Description:\*\* {weather\_data['description']}")

st.write(f"\*\*Temperature:\*\* {weather\_data['temperature']} °C")

st.write(f"\*\*Feels Like:\*\* {weather\_data['feels\_like']} °C")

st.write(f"\*\*Humidity:\*\* {weather\_data['humidity']}%")

st.write(f"\*\*Pressure:\*\* {weather\_data['pressure']} hPa")

st.write(f"\*\*Wind Speed:\*\* {weather\_data['wind\_speed']} m/s")

st.subheader("Visualizations:")

# 1. Bar Chart using Seaborn

st.subheader("Temperature and Feels Like Comparison")

temp\_df = pd.DataFrame({

'Parameter': ['Temperature', 'Feels Like'],

'Value': [weather\_data['temperature'], weather\_data['feels\_like']]

})

plt.figure(figsize=(6, 4))

sns.barplot(x='Parameter', y='Value', data=temp\_df, palette="viridis")

plt.ylabel("Temperature (°C)")

st.pyplot(plt)

plt.close()

# 2. Scatter Plot using Matplotlib

st.subheader("Humidity vs. Pressure")

plt.figure(figsize=(6, 4))

plt.scatter(weather\_data['humidity'], weather\_data['pressure'], color='skyblue', alpha=0.7)

plt.xlabel("Humidity (%)")

plt.ylabel("Pressure (hPa)")

plt.title("Relationship between Humidity and Pressure")

plt.grid(True)

st.pyplot(plt)

plt.close()

# 3. Pie Chart using Matplotlib

weather\_params = ['Humidity', 'Pressure', 'Wind Speed']

values = [weather\_data['humidity'], weather\_data['pressure'], weather\_data['wind\_speed']]

plt.figure(figsize=(6, 6))

plt.pie(values, labels=weather\_params, autopct='%1.1f%%', startangle=140, colors=sns.color\_palette('pastel'))

plt.title("Distribution of Key Weather Parameters")

st.pyplot(plt)

plt.close()

else:

st.error("City not found. Please enter a valid city name.")

except requests.exceptions.RequestException as e:

st.error(f"Error fetching data: {e}")

except KeyError as e:

st.error(f"Error accessing data: Key '{e}' not found in the API response.")

1. **Replace the API Key:** Find the line api\_key = "YOUR\_OPENWEATHERMAP\_API\_KEY" and replace "YOUR\_OPENWEATHERMAP\_API\_KEY" with the actual API key you obtained in Step 1.
2. **Save the File:** Save the edited file as weather\_dashboard.py.

**Step 4: Run the Visualization Dashboard**

1. Open your terminal or command prompt.
2. Navigate to the directory where you saved the weather\_dashboard.py file using the cd command.
3. Run the Streamlit application using the command:

Bash

streamlit run weather\_dashboard.py

This command will launch the Streamlit dashboard in your default web browser.

**Step 5: Interact with the Dashboard**

1. A web page will open, displaying the "Weather Insights Dashboard" for the default city (Kolkata).
2. On the left sidebar, you will see a text input field labeled "Enter City Name:".
3. Type the name of any city in India (or any city worldwide) into the text box and press Enter.
4. The dashboard will update, displaying the current weather information for the entered city and the three visualizations:
   * A bar chart comparing temperature and feels-like temperature.
   * A scatter plot showing the relationship between humidity and pressure.
   * A pie chart showing the distribution of humidity, pressure, and wind speed.

**Step 6: Deliverables**

Your deliverables for this task are:

1. **The Python Script (**weather\_dashboard.py**):** This is the code you created in Step 3. Make sure it includes your OpenWeatherMap API key.
2. **The Visualization Dashboard:** This is the web application that you run using Streamlit (as described in Step 4 and interact with in Step 5). You can demonstrate this by:
   * Taking screenshots of the dashboard showing weather data and the visualizations for a few different cities.
   * If possible, you could also provide a live demonstration by running the script.

By following these steps, you will have successfully fetched data from the OpenWeatherMap API using Python and created a visualization dashboard using Matplotlib and Seaborn, presented through a Streamlit web application. This covers both the script and the interactive dashboard deliverables. Good luck with your internship!