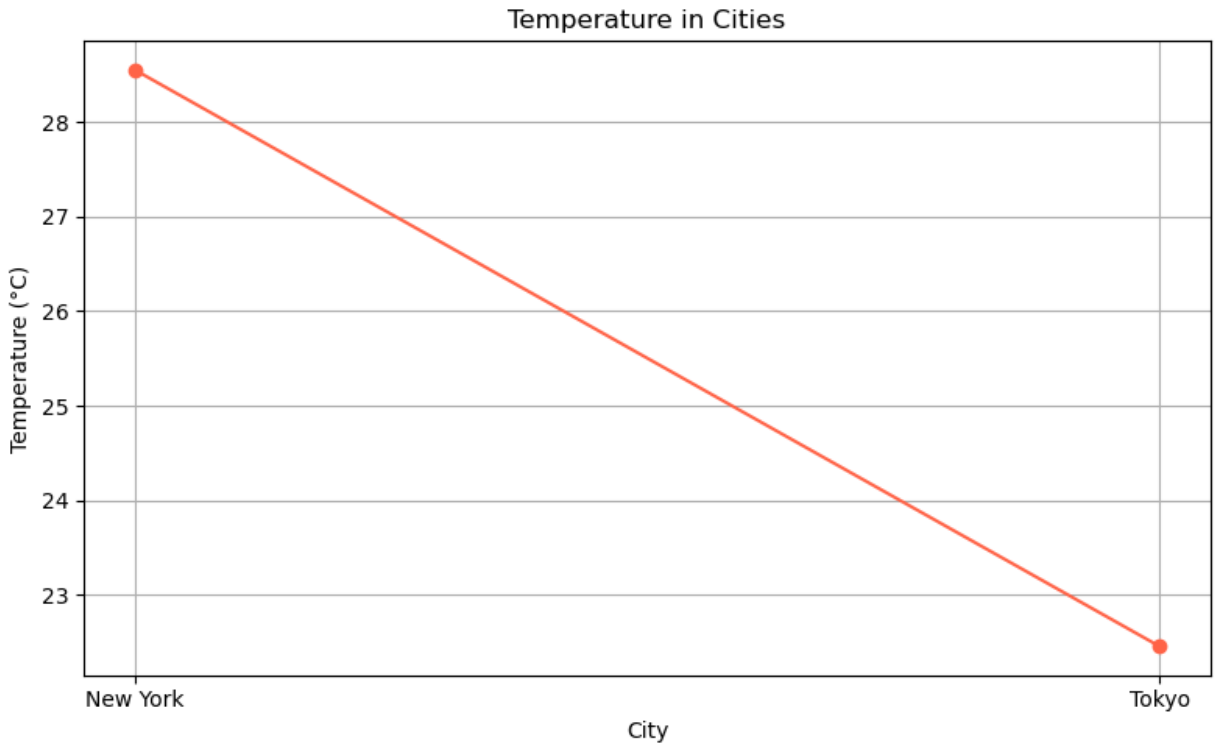
Task Overview & Explanations

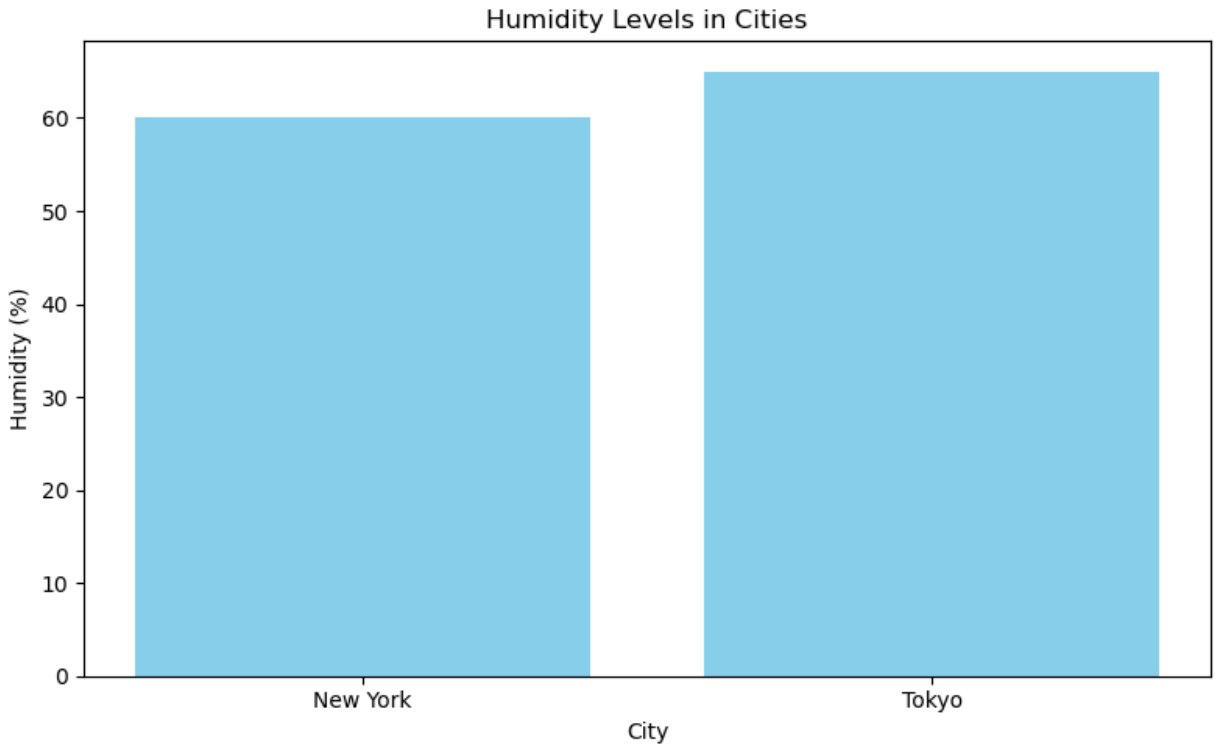
1. **API Setup**
    - Signed up at OpenWeather and obtained a free API key for authentication.
2. **API Request**
    - Used the requests library to make a GET request for **current weather data** for:
        - New York
        - Tokyo
    - Queried temperature and humidity in metric units (°C, %).
3. **Error & Rate Limit Handling**
    - Implemented try/except block for network or city errors.
    - Handled API rate limits (HTTP 429) using a delay-and-retry mechanism.
4. **Data Extraction**
    - Parsed JSON responses to extract:
        - Temperature (main.temp)
        - Humidity (main.humidity)
5. **Data Conversion**
    - Stored data in a clean pandas DataFrame for easy manipulation and plotting.
6. **Visualization**
    - Created:
        - **Line Plot** for Temperature across cities.
        - **Bar Chart** for Humidity levels.
    - Used matplotlib for basic plot styling with titles and labels.

📊 **Visualizations**

1. **Temperature Line Plot**



2. Humidity Bar Chart



📌 Observations

- **New York** was warmer than Tokyo at the time of the request.
- **Tokyo** had higher humidity.
- The OpenWeather API was responsive and easy to use for basic weather metrics.

Conclusion

This assignment demonstrated how to:

- Access real-time data via APIs,
- Handle responses and errors effectively,
- Visualize results clearly using Python libraries.

This is an essential workflow for data analysts working with external APIs.