**Real-Time Weather App**

* **Abstract**

The Weather Monitoring App is a Python-based project developed using **Streamlit** and the **OpenWeatherMap API**. It provides real-time weather information and a 5-day forecast in an interactive format. Users can enter a city name, choose temperature units (Celsius or Fahrenheit), and view details such as temperature, humidity, sunrise, sunset, and weather descriptions with icons. The forecast is also visualized through a line chart for trend analysis. The app is simple, responsive, and user-friendly, but it can only fetch data for cities available in the API’s database. This project demonstrates practical API integration and visualization in Python for real-world use.

* **Introduction**

Weather strongly impacts daily activities, travel, agriculture, and planning. This project presents a **Weather Monitoring App** that provides up-to-date weather information using a clean and interactive interface. Built with **Python and Streamlit**, the app integrates with the **OpenWeatherMap API** to fetch accurate data for cities in its database.

The application displays current conditions such as temperature, humidity, sunrise, sunset, and general weather descriptions. It also includes a **5-day forecast**, represented in cards and a temperature trend chart using **Matplotlib**. With its lightweight design and ease of use, the app is a practical example of combining data APIs with visualization tools to create meaningful applications for everyday users.

* **Tools Used**
* **Programming Language:** Python
* **Framework:** Streamlit
* **Visualization Library:** Plotly
* **Data Visualisation:** **Matplotlib**
* **API Service:** **OpenWeatherMap API**
* **Steps Involved in Building the Project**

1. **Requirement Analysis** – Identified the need for a lightweight, interactive weather monitoring tool.
2. **Environment Setup** – Installed Python, Streamlit, Requests, and Matplotlib libraries.
3. **API Integration** – Registered with OpenWeatherMap API and obtained an API key for fetching live weather data.
4. **Backend Development** – Implemented functions in Python to fetch current weather and forecast data from the API.
5. **Frontend Development** – Designed the user interface using Streamlit with sidebar input, styled weather cards, and icons.
6. **Data Visualization** – Used Matplotlib to plot a line graph showing forecast temperature trends over 5 days.
7. **Error Handling** – Added logic to handle invalid city names or unavailable data in the API database.
8. **Testing & Deployment** – Tested with different cities and deployed locally as a Streamlit web app.

* **Conclusion**

The Weather Monitoring App provides real-time weather updates in an interactive and visually appealing format. It demonstrates how Python can integrate APIs and visualization tools to create useful applications. Though limited to cities listed in the API’s database, it serves as a practical tool for daily weather monitoring and can be extended with additional features like historical data, multiple city tracking, or severe weather alerts.

* **GitHub Repository Link:**

<https://github.com/Arpita368/Python-Internship-Project>

* **App Link:**

<https://python-internship-project-lcvveo38tuahfmjdurju4f.streamlit.app/>

* **LinkedIn Profile:**

<https://in.linkedin.com/in/arpita-sonparote-9b73a1313>