

# **Real-Time Weather App**

## **Introduction**

The Real-Time Weather App is a lightweight, interactive web application designed to provide users with current and forecasted weather information for any global city. Developed using Python and Streamlit, the app connects to the OpenWeatherMap API to display real-time temperature, humidity, weather conditions, sunrise and sunset times, and a 5-day weather forecast. The application includes a user-friendly interface that allows toggling between Celsius and Fahrenheit units, and uses weather icons for enhanced visualization.

## **Abstract**

This project demonstrates how to combine APIs, data visualization, and web development tools to create an efficient and visually engaging weather monitoring application. The main goal was to fetch live weather data using OpenWeatherMap's RESTful API and present it in a concise, informative format using Streamlit. Additional features such as weather condition icons, unit toggling, and a temperature forecast chart help provide users with useful insights for planning daily activities, travel, or events. This project emphasizes the importance of building practical, real-world applications with minimal dependencies.

## **Tools Used**

- Python 3.8+ – Core programming language
- Streamlit – Frontend and UI development
- OpenWeatherMap API – Real-time and forecast weather data
- requests – HTTP requests to the API
- matplotlib – Temperature trend chart visualization
- pytz – Time zone management for sunrise/sunset times

## Steps Involved in Building the Project

1. **User Input & Interface:** Created a Streamlit interface with input fields for city name and unit toggle (Celsius/Fahrenheit).
2. **API Integration:** Fetched live weather and 5-day forecast data using OpenWeatherMap's API with the requests library.
3. **Data Parsing & Display:** Extracted and displayed key metrics including temperature, humidity, sunrise/sunset, and general weather condition.
4. **Icon Mapping:** Mapped weather descriptions (like "Rain", "Clear", "Clouds") to dynamic icons for better UX.
5. **Forecast Chart:** Used matplotlib to visualize the 5-day forecast as a line chart, with temperature values over time.
6. **Error Handling:** Added basic error checking for invalid city names or API failures.

## Conclusion

The Real-Time Weather App provides a functional and visually engaging way to access current and future weather data. It integrates third-party APIs, data parsing, and charting libraries to deliver real-time updates with minimal latency. The project highlights practical skills in working with APIs, handling time zones, and developing responsive user interfaces in Python. It serves as a strong example of how to transform raw API data into actionable and attractive web content.