

Real-Time Weather App — Project Report

Introduction

The Real-Time Weather App is a Python-based web application designed to provide users with live weather information and a 5-day forecast for any city across the globe. It leverages the OpenWeatherMap API to fetch weather data and Streamlit for building a clean, interactive web interface. The project demonstrates how real-world data can be retrieved, processed, and presented visually for an enhanced user experience.

Abstract

This project aims to develop an interactive weather forecasting web application using Python. The system retrieves weather information in real-time, such as temperature, humidity, pressure, and weather conditions (like sunny, cloudy, or rainy). It also displays a 5-day temperature forecast graph using Plotly charts. The interface allows users to input a city name, after which the app fetches live data and visualizes it dynamically. The application ensures accessibility, simplicity, and accuracy in presenting weather data.

Tools Used

Tool / Library Purpose

Python Core programming language used for the project.

Streamlit Framework used to create an interactive web interface.

Requests Used for making API calls to OpenWeatherMap.

Pandas Used to process and organize fetched JSON data.

Plotly Express For visualizing temperature trends and forecast charts.

OpenWeatherMap API External weather data provider.

Steps Involved in Building the Project

Step 1: Set up the development environment and install dependencies like Streamlit, Pandas, Requests, and Plotly.

Step 2: Obtain an API key from OpenWeatherMap for data access.

Step 3: Create a user input interface in Streamlit for entering city names.

Step 4: Use the Requests library to fetch real-time weather and forecast data in JSON format.

Step 5: Process and clean the JSON data using Pandas for easier visualization.

Step 6: Display current temperature, humidity, and weather description on the UI.

Step 7: Use Plotly to create a 5-day temperature trend chart.

Step 8: Test and deploy the app locally using Streamlit's run command.

Conclusion

The Real-Time Weather App successfully integrates real-world APIs with an interactive frontend, demonstrating the practical use of Python in data visualization and web application development. It provides accurate and up-to-date weather information in a simple, modern interface. Future improvements may include temperature unit toggling, animated weather icons, and automatic location detection for enhanced usability.