

Weather Report Dashboard - Technical Report

Overview

This project is a real-time weather monitoring web application developed using Python and Streamlit. It leverages the OpenWeatherMap API to provide current weather insights with dynamic visualizations, supporting multiple cities and automated analytics.

1. Approach

1.1 Architecture

- **Language & Frameworks:** Python, Streamlit (for UI), Pandas, Plotly, Matplotlib, Seaborn
- **API Integration:** Uses OpenWeatherMap API for fetching live weather data
- **Modular Design:** Codebase is organized into modules for API communication, data handling, and UI rendering

1.2 Data Flow

1. **Fetch:** Real-time data is fetched from the OpenWeatherMap API
2. **Process:** Data is cleaned and organized with Pandas
3. **Visualize:** Interactive graphs and charts are generated with Plotly, Matplotlib, and Seaborn
4. **Display:** Everything is presented via an intuitive Streamlit interface

1.3 Key Features

- Multi-city weather tracking
 - Dynamic dashboard with real-time updates
 - Rich visualizations (line charts, bar graphs, etc.)
 - Auto-generated weather summary reports
 - Basic statistical trends and temperature analysis
-

2. How to Run the Code

2.1 Prerequisites

- Python 3.x installed
- An active internet connection
- OpenWeatherMap API key (included or prompted via the code)

2.2 Installation

```
bash
CopyEdit
# Clone the repository (if not already done)
git clone <repo-url>
cd <project-folder>

# Install required packages
pip install -r requirements.txt
```

2.3 Running the Application

```
bash
CopyEdit
streamlit run streamlit_app.py
```

2.4 Accessing the Dashboard

- Open a browser and go to: <http://localhost:8501>
 - The dashboard will load automatically and begin fetching live weather data
-

3. Assumptions

3.1 Environment & System

- User has Python 3.x and necessary libraries installed
- The system can process moderate amounts of data efficiently
- The Streamlit port (8501) is available and not blocked

3.2 API Usage

- API key has sufficient quota and privileges
- API responses are valid and in JSON format
- The `cities.txt` file contains valid city names or coordinates

3.3 Connectivity

- Persistent internet access is available throughout use
- The OpenWeatherMap API remains operational

3.4 User Permissions

- Users have permission to read/write files in the project directory
 - Log files and generated reports are stored locally
-

4. Limitations & Future Scope

Limitations

- Limited by OpenWeatherMap API rate limits
- May lag with too many cities due to local processing
- Doesn't yet include historical data or forecasting

Future Enhancements

- Add weather forecast support (e.g., 7-day prediction)
- Integrate map-based UI for better city selection
- Enable export to PDF/Excel
- Add email alerts for extreme weather conditions