Final Project Report

# Project Title:

EnviroTrack: Real-Time Weather and Air Quality Monitoring System

# Objective:

To develop a real-time dashboard that fetches and visualizes weather and air quality data using public APIs. The goal is to provide users with insightful data visualization and environmental awareness for any location.

# Tools and Technologies Used:

Frontend: HTML, CSS, JavaScript (or React.js)

Backend: Python (Flask) or Node.js (Express)

Data Visualization: Chart.js / Plotly / Streamlit

APIs: OpenWeatherMap API, IQAir API

Deployment: Streamlit Cloud / Render / GitHub Pages

# Key Features:

🔍 Search by City/Location

🌡️ Display Real-Time Weather Info: Temperature, humidity, wind, cloud coverage

🌫️ Display Air Quality (AQI) with health recommendation tags

📈 Visualize Historical Trends using line/bar charts

📦 Save Search History (optional)

🎨 Responsive & Clean UI using Bootstrap or Tailwind CSS

# System Design:

1. User inputs city name.

2. Backend fetches weather and air quality data from respective APIs.

3. Parsed data is sent to frontend.

4. Frontend displays real-time values and visualizes trends with charts.

# Sample Python API Integration (Flask):

import requests  
from flask import Flask, request, jsonify, render\_template  
  
app = Flask(\_\_name\_\_)  
WEATHER\_API = "YOUR\_OPENWEATHERMAP\_KEY"  
AIR\_API = "YOUR\_IQAIR\_API\_KEY"  
  
@app.route('/')  
def home():  
 return render\_template('index.html')  
  
@app.route('/get\_data', methods=['POST'])  
def get\_data():  
 city = request.form['city']  
 weather\_url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={WEATHER\_API}&units=metric"  
 air\_url = f"http://api.airvisual.com/v2/city?city={city}&key={AIR\_API}"  
  
 weather\_res = requests.get(weather\_url).json()  
 air\_res = requests.get(air\_url).json()  
  
 return jsonify({  
 'temperature': weather\_res['main']['temp'],  
 'humidity': weather\_res['main']['humidity'],  
 'wind': weather\_res['wind']['speed'],  
 'aqi': air\_res['data']['current']['pollution']['aqius']  
 })  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)

# Expected Output:

A real-time dashboard that shows:

- Current temperature, humidity, and wind speed

- Current AQI with health warnings

- Interactive charts for AQI and temperature trends

# Conclusion:

This project demonstrates how public APIs can be integrated into real-time applications for meaningful insights. It highlights the power of data visualization and web development to build informative dashboards.