📂 Project Structure

Climate-map/

│── frontend/

│ ├── index.html

│ ├── styles.css

│ ├── script.js

│── backend/

│ ├── app.py

│ ├── requirements.txt

│── README.md

🌍 Frontend Code

1️⃣ index.html (Main Web Page)

<!DOCTYPE html>

<html lang=”en”>

<head>

<meta charset=”UTF-8”>

<meta name=”viewport” content=”width=device-width, initial-scale=1.0”>

<title>Climate Change Impact Map</title>

<link rel=”stylesheet” href=”styles.css”>

<link rel=”stylesheet” href=<https://unpkg.com/leaflet/dist/leaflet.css>/>

</head>

<body>

<h1>Climate Change Impact Map</h1>

<div id=”info”>

<p><strong>Click on the map to get climate data</strong></p>

</div>

<div id=”map”></div>

<script src=<https://unpkg.com/leaflet/dist/leaflet.js>></script>

<script src=”script.js”></script>

</body>

</html>

2️⃣ styles.css (Styling)

Body {

Font-family: Arial, sans-serif;

Text-align: center;

Margin: 0;

Padding: 0;

}

H1 {

Margin: 10px 0;

}

#map {

Height: 500px;

Width: 90%;

Margin: auto;

Border-radius: 10px;

}

#info {

Margin: 10px auto;

Width: 90%;

Padding: 10px;

Background: #f4f4f4;

Border-radius: 8px;

}

3️⃣ script.js (Map & Data Fetching)

Const map = L.map(“map”).setView([20, 78], 4); // Center on India

L.tileLayer(https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png, {

Attribution: “&copy; OpenStreetMap contributors”

}).addTo(map);

Async function fetchClimateData(lat, lon) {

Const response = await fetch(`http://127.0.0.1:5000/api/climate?lat=${lat}&lon=${lon}`);

Return response.json();

}

Map.on(“click”, async function € {

Const { lat, lng } = e.latlng;

Const data = await fetchClimateData(lat, lng);

L.marker([lat, lng]).addTo(map)

.bindPopup(`<strong>Temperature:</strong> ${data.temp}°C<br>

<strong>Weather:</strong> ${data.weather}<br>

<strong>CO2 Levels:</strong> ${data.co2} ppm`)

.openPopup();

});

🚀 Backend Code (Flask)

4️⃣ app.py (API Server)

From flask import Flask, request, jsonify

Import requests

App = Flask(\_\_name\_\_)

API\_KEY = “your\_openweathermap\_api\_key”

@app.route(“/api/climate”, methods=[“GET”])

Def get\_climate\_data():

Lat = request.args.get(“lat”)

Lon = request.args.get(“lon”)

Weather\_url = f[https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={API\_KEY}&units=metric](https://api.openweathermap.org/data/2.5/weather?lat=%7blat%7d&lon=%7blon%7d&appid=%7bAPI_KEY%7d&units=metric)

Weather\_data = requests.get(weather\_url).json()

Co2\_levels = 420 # Mock CO2 value

Return jsonify({

“temp”: weather\_data[“main”][“temp”],

“weather”: weather\_data[“weather”][0][“description”],

“co2”: co2\_levels

})

If \_\_name\_\_ == “\_\_main\_\_”:

App.run(debug=True)

5️⃣ requirements.txt (Dependencies)

Flask

Requests

🔧 Setup & Run

📥 Install Dependencies

Pip install -r requirements.txt

🚀 Start Backend

Python backend/app.py

🌍 Open index.html in a Browser

📌 Features

✔️ Interactive Map – Click anywhere to get climate data

✔️ Real-time Weather Data – Uses OpenWeatherMap API

✔️ Mock CO2 Levels – Can be replaced with real API

✔️ Lightweight – No database needed