**PROGRAM**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Weather Dashboard</title>

<style>

body {

font-family: "Poppins", sans-serif;

background: linear-gradient(135deg, #74ebd5 0%, #acb6e5 100%);

color: #333;

text-align: center;

margin: 0;

padding: 20px;

transition: background 0.5s;

}

h1 { color: #222; }

#container {

max-width: 700px;

margin: auto;

background: white;

padding: 25px;

border-radius: 15px;

box-shadow: 0 4px 10px rgba(0,0,0,0.1);

}

input, button {

padding: 10px;

border-radius: 8px;

border: 1px solid #ccc;

margin: 5px;

}

button {

background-color: #007BFF;

color: white;

cursor: pointer;

transition: 0.3s;

}

button:hover { background-color: #0056b3; }

#weatherInfo, #forecast, #airQuality { margin-top: 20px; }

.forecast-container {

display: flex;

flex-wrap: wrap;

justify-content: center;

gap: 10px;

}

.forecast {

border: 1px solid #ddd;

border-radius: 8px;

padding: 10px;

background: #f9f9f9;

width: 130px;

}

.history button {

background: #eee;

color: #333;

}

.history button:hover {

background: #ddd;

}

#loader {

display: none;

margin-top: 15px;

}

.spinner {

border: 4px solid #f3f3f3;

border-top: 4px solid #007BFF;

border-radius: 50%;

width: 30px;

height: 30px;

animation: spin 1s linear infinite;

margin: auto;

}

@keyframes spin {

0% { transform: rotate(0deg); }

100% { transform: rotate(360deg); }

}

#themeToggle {

background: #222;

color: #fff;

}

.dark {

background: linear-gradient(135deg, #1f1f1f 0%, #434343 100%);

color: #eee;

}

</style>

</head>

<body>

<div id="container">

<h1>🌤️ Weather Dashboard</h1>

<input type="text" id="cityInput" placeholder="Enter city name">

<button onclick="getWeather()">Search</button>

<button onclick="getLocationWeather()">📍 Current Location</button>

<button id="themeToggle" onclick="toggleTheme()">🌗 Toggle Theme</button>

<div id="loader"><div class="spinner"></div></div>

<div id="weatherInfo"></div>

<div id="airQuality"></div>

<div id="forecast"></div>

<div class="history">

<h3>Search History</h3>

<div id="historyList"></div>

</div>

</div>

<script>

const apiKey = "YOUR\_OPENWEATHERMAP\_API\_KEY";

const AQI\_API\_KEY = "YOUR\_AIRVISUAL\_API\_KEY"; // optional Air Quality API

async function getWeather(city = null) {

if (!navigator.onLine) {

alert("You are offline! Please check your internet connection.");

return;

}

if (!city) city = document.getElementById("cityInput").value.trim();

if (!city) return alert("Please enter a city name");

document.getElementById("loader").style.display = "block";

document.getElementById("weatherInfo").innerHTML = "";

document.getElementById("forecast").innerHTML = "";

document.getElementById("airQuality").innerHTML = "";

try {

// Current weather

const response = await fetch(

`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}&units=metric`

);

if (!response.ok) throw new Error("City not found");

const data = await response.json();

const { name, main, weather, wind, coord } = data;

document.getElementById("weatherInfo").innerHTML = `

<h2>${name}</h2>

<img src="https://openweathermap.org/img/wn/${weather[0].icon}@2x.png" alt="icon">

<p><b>Temperature:</b> ${main.temp} °C</p>

<p><b>Feels Like:</b> ${main.feels\_like} °C</p>

<p><b>Weather:</b> ${weather[0].description}</p>

<p><b>Humidity:</b> ${main.humidity}% | <b>Wind:</b> ${wind.speed} m/s</p>

`;

saveToHistory(name);

await getForecast(city);

await getAirQuality(coord.lat, coord.lon);

} catch (err) {

document.getElementById("weatherInfo").innerHTML =

`<p style="color:red;">❌ ${err.message}</p>`;

} finally {

document.getElementById("loader").style.display = "none";

}

}

async function getForecast(city) {

try {

const res = await fetch(

`https://api.openweathermap.org/data/2.5/forecast?q=${city}&appid=${apiKey}&units=metric`

);

const data = await res.json();

let forecastHTML = '<h3>5-Day Forecast</h3><div class="forecast-container">';

data.list.filter((\_, i) => i % 8 === 0).forEach(item => {

forecastHTML += `

<div class="forecast">

<p>${new Date(item.dt\_txt).toLocaleDateString()}</p>

<img src="https://openweathermap.org/img/wn/${item.weather[0].icon}.png" alt="icon">

<p>${item.main.temp} °C</p>

<p>${item.weather[0].main}</p>

</div>

`;

});

forecastHTML += "</div>";

document.getElementById("forecast").innerHTML = forecastHTML;

} catch {

document.getElementById("forecast").innerHTML = "<p>Forecast not available.</p>";

}

}

async function getAirQuality(lat, lon) {

try {

const res = await fetch(`https://api.openweathermap.org/data/2.5/air\_pollution?lat=${lat}&lon=${lon}&appid=${apiKey}`);

const data = await res.json();

const aqi = data.list[0].main.aqi;

const levels = ["Good 🌱", "Fair 🙂", "Moderate 😐", "Poor 😷", "Very Poor ☠️"];

document.getElementById("airQuality").innerHTML = `

<h3>Air Quality Index</h3>

<p>${levels[aqi - 1]} (Level: ${aqi})</p>

`;

} catch {

document.getElementById("airQuality").innerHTML = "<p>Air quality data unavailable.</p>";

}

}

function getLocationWeather() {

if (!navigator.geolocation) {

alert("Geolocation not supported by your browser.");

return;

}

navigator.geolocation.getCurrentPosition(

async pos => {

const { latitude, longitude } = pos.coords;

const res = await fetch(`https://api.openweathermap.org/data/2.5/weather?lat=${latitude}&lon=${longitude}&appid=${apiKey}&units=metric`);

const data = await res.json();

getWeather(data.name);

},

() => alert("Unable to retrieve your location.")

);

}

function saveToHistory(city) {

let history = JSON.parse(localStorage.getItem("weatherHistory")) || [];

if (!history.includes(city)) {

history.push(city);

localStorage.setItem("weatherHistory", JSON.stringify(history));

}

displayHistory();

}

function displayHistory() {

let history = JSON.parse(localStorage.getItem("weatherHistory")) || [];

document.getElementById("historyList").innerHTML =

history.map(c => `<button onclick="getWeather('${c}')">${c}</button>`).join("");

}

function toggleTheme() {

document.body.classList.toggle("dark");

}

window.onload = displayHistory;

</script>

</body>

</html>