***WEATHER FORECASTING***

Weather forecasting is the prediction of what the atmosphere will be like in a particular place by using technology and scientific knowledge to make weather observations. In other words, it is a way of predicting things like cloud cover, rain, snow, wind speed, and temperature before they happen.

Weather forecasters use all kinds of tools to achieve this goal. we have instruments called barometers to measure air pressure, radar to measure the location and speed of clouds, thermometers to measure temperature and computer models to process data accumulated from these instruments.

However, to this day, humans with good experience can still do a better job at predicting weather than computer models alone because humans are often involved in picking the most appropriate model for a situation.

Hazardous and severe weather forecasts are very important they can potential save lives,property and crops.



***EXPLANATION*:**

To create webpage to forecast the weather we need three different types of files. They are

1. Index.HTML
2. Style.CSS
3. Script.js

The main file is HTML file in which it includes the other two file.

**HTML Structure:**

The HTML sets up a simple structure with an input field for the city name, a button to fetch the weather, and a display area for the weather information. It also includes links to external stylesheets and scripts.

**CSS Styling:**

The CSS provides styling for the page elements to make them visually appealing. This includes a background gradient, styling for the weather card, input fields, and buttons, as well as transitions and hover effects.

**JavaScript Functionality:**

**API Setup:**

The script includes the URL for the OpenWeatherMap API and an API key for authentication.

**Document Ready:**

When the document is ready, it fetches the weather data for 'Pune' as a default.

**Weather Fetching Function:**

weatherFn(cName) constructs the API request URL with the city name and fetches the weather data. If the city is found, it calls weatherShowFn(data) to display the data; otherwise, it shows an alert.

**Display Function:**

Weather(data) updates the DOM elements with the fetched weather data (city name, date, temperature, description) and makes the weather info section visible using a fade-in animation

**Index.HTML**

<!DOCTYPE html>

<head>

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

<link rel="stylesheet" href=

"https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css">

<link rel="stylesheet" href=

"https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.1/css/all.min.css">

<link rel="stylesheet" href=

"https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&display=swap">

<title>GFG App</title>

</head>

<body>

<div class="container">

<div class="weather-card">

<h3>

Weather

</h3>

<input type="text" id="city-input"

placeholder="Enter city name">

<button id="city-input-btn"

onclick="weatherFn($('#city-input').val())">

Get Weather

</button>

<div id="weather-info"

class="animate\_\_animated animate\_\_fadeIn">

<h3 id="city-name"></h3>

<p id="date"></p>

<img id="weather-icon" src="https://cdn-icons-png.flaticon.com/128/12607/12607703.png" alt="Weather Icon">

<p id="temperature"></p>

<p id="description"></p>

<p id="wind-speed"></p>

</div>

</div>

</div>

<script src=

"https://code.jquery.com/jquery-3.6.0.min.js">

</script>

<script src=

"https://momentjs.com/downloads/moment.min.js">

</script>

<script src="script.js"></script>

</body>

</html>

**Styles.CSS:**

body {

margin: 0;

font-family: 'Montserrat', sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background: linear-gradient(to right, #4CAF50, #2196F3);

}

.container {

text-align: center;

}

.weather-card {

background-color: rgba(255, 255, 255, 0.95);

border-radius: 20px;

padding: 20px;

box-shadow: 0 0 30px rgba(0, 0, 0, 0.1);

transition: transform 0.3s ease-in-out;

width: 450px;

}

.weather-card:hover {

transform: scale(1.05);

}

#city-input {

padding: 15px;

margin: 10px 0;

width: 70%;

border: 1px solid #ccc;

border-radius: 5px;

font-size: 16px;

}

#city-input:focus {

outline: none;

border-color: #2196F3;

}

#city-input::placeholder {

color: #aaa;

}

#city-input-btn {

padding: 10px;

background-color: #2196F3;

color: #fff;

border: none;

border-radius: 5px;

font-size: 16px;

cursor: pointer;

}

#city-input-btn:hover {

background-color: #1565C0;

}

#weather-info {

display: none;

}

#weather-icon {

width: 100px;

height: 100px;

}

#temperature {

font-size: 24px;

font-weight: bold;

margin: 8px 0;

}

#description {

font-size: 18px;

margin-bottom: 10px;

}

#wind-speed {

font-size: 16px;

color: rgb(255, 0, 0);

}

#date {

font-size: 14px;

color: rgb(255, 0, 0);

}

**Script.js:**

const url =

'https://api.openweathermap.org/data/2.5/weather';

const apiKey =

'f00c38e0279b7bc85480c3fe775d518c';

$(document).ready(function () {

weatherFn('Pune');

});

async function weatherFn(cName) {

const temp =

`${url}?q=${cName}&appid=${apiKey}&units=metric`;

try {

const res = await fetch(temp);

const data = await res.json();

if (res.ok) {

weatherShowFn(data);

} else {

alert('City not found. Please try again.');

}

} catch (error) {

console.error('Error fetching weather data:', error);

}

}

function weatherShowFn(data) {

$('#city-name').text(data.name);

$('#date').text(moment().

format('MMMM Do YYYY, h:mm a'));

$('#temperature').

html(`${data.main.temp}°C`);

$('#description').

text(data.weather[0].description);

$('#weather-info').fadeIn();

}