WEATHER FORECAST APP

**Project Name:**

**Basic Weather App**

**Aim:**

The aim should be to implement a **Basic Weather App** that provides predefined weather information for a few cities without the need for an external API.

**Description:**

This project is a simple weather application that allows users to input a city name and view its corresponding weather details. The app uses hardcoded weather data stored within the code to display information such as temperature and weather conditions. It is designed to provide basic functionality and is suitable for beginners learning web development concepts.

**Features:**

1. **City Search:**
   * Users can type the name of a city into an input field.
2. **Weather Details:**
   * Displays temperature and weather conditions for predefined cities.
3. **Case Insensitivity:**
   * Accepts city names in any letter case (e.g., "Chennai" or "chennai").
4. **Friendly Messages:**
   * If weather data for a city is unavailable, it informs the user.

**Technologies Used:**

1. **HTML:**
   * For structuring the webpage.
2. **CSS:**
   * For styling the application.
3. **JavaScript:**
   * For functionality, including processing user input and displaying weather data.

**How It Works:**

1. The user enters a city name in the input box and clicks the **Get Weather** button.
2. The JavaScript code checks if the entered city exists in the hardcoded weather data object.
3. If the city is found, the weather details (temperature and condition) are displayed.
4. If the city is not found, the app displays a message indicating that weather data for the city is unavailable.

**Limitations:**

1. The app does not fetch real-time weather data because it does not use an API.
2. The weather information is limited to a predefined set of cities and static data.

**Sample Data:**

**City:** Chennai

* **Temperature:** 30°C
* **Condition:** Hot and Humid

**City:** London

* **Temperature:** 15°C
* **Condition:** Cloudy

**City:** New York

* **Temperature:** 20°C
* **Condition:** Sunny

**Learning Outcomes:**

* Understanding the basics of HTML, CSS, and JavaScript integration.
* Implementing input handling and dynamic content rendering in JavaScript.
* Learning to manage case sensitivity and provide user-friendly feedback.

CODE

Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Basic Weather App</title>

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

</head>

<body>

<h1>Weather App</h1>

<div>

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

<button id="getWeather">Get Weather</button>

</div>

<div id="weatherResult">

<!-- Weather details will appear here -->

</div>

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

</body>

</html>

CSS

styles.css

body {

font-family: Arial, sans-serif;

text-align: center;

padding: 20px;

background-color: #f0f8ff;

}

h1 {

color: #333;

}

input, button {

padding: 10px;

margin: 5px;

font-size: 16px;

}

button {

background-color: #007bff;

color: white;

border: none;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

#weatherResult {

margin-top: 20px;

font-size: 18px;

color: #555;

}

JAVA SCRIPT

script.js

const button = document.getElementById("getWeather");

const weatherResult = document.getElementById("weatherResult");

// Sample weather data for demonstration

const sampleWeatherData = {

"New York": { temp: 20, condition: "Sunny" },

"London": { temp: 15, condition: "Cloudy" },

"Paris": { temp: 18, condition: "Rainy" },

"Tokyo": { temp: 22, condition: "Clear" }

};

button.addEventListener("click", () => {

const city = document.getElementById("city").value.trim();

if (sampleWeatherData[city]) {

// Fetch the weather data from the sample object

const { temp, condition } = sampleWeatherData[city];

weatherResult.innerHTML = `

<p>City: ${city}</p>

<p>Temperature: ${temp}°C</p>

<p>Condition: ${condition}</p>

`;

} else {

weatherResult.innerHTML = `<p>Weather data for "${city}" is not available.</p>`;

}

});

OUTPUT:

