Weather App Tutorial

Let's build a **Weather App** step by step, ensuring you understand **API integration, DOM manipulation, and handling JSON data** properly.

Step 1: Understanding the Project

- The app should allow users to **input a city name** and fetch the **current weather** for that city.
- It should use **an external weather API** (e.g., OpenWeatherMap).
- The data should be displayed dynamically on the webpage.

Step 2: Setting Up the Project

Folder Structure

```rust
weather-app/
│── index.html   (HTML structure)
│── style.css    (CSS for styling)
│── script.js    (JavaScript for functionality)
```

Step 3: Get API Key

We will use **OpenWeatherMap API**.

1. Go to OpenWeatherMap.
2. Sign up and get an **API Key** (it's free).
3. The API endpoint to get weather data:

   ```bash
   https://api.openweathermap.org/data/2.5/weather?q=CITY_NAME&appid=YOUR_API_KEY&units=metric
   ```

   - **`q=CITY_NAME`**: City name entered by the user.
   - **`appid=YOUR_API_KEY`**: Your API key.
   - **`units=metric`**: To get temperature in Celsius.

Step 4: HTML Structure

This file provides:

- An input field to enter the city name.

- A button to fetch weather data.
- A section to display the fetched weather.

html

```html
<!-- index.html -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Weather App</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <h1>Weather App</h1>
        <div class="search-box">
            <input type="text" id="city" placeholder="Enter city name">
            <button id="searchBtn">Search</button>
        </div>
        <div class="weather-info">
            <h2 id="city-name"></h2>
            <p id="temperature"></p>
            <p id="description"></p>
            <img id="weather-icon" src="" alt="">
        </div>
    </div>
    <script src="script.js"></script>
</body>
</html>
```

Step 5: Styling the App (CSS)

This makes the app visually appealing.

css

```css
/* style.css */
body {
    font-family: Arial, sans-serif;
    background: linear-gradient(to right, #74ebd5, #acb6e5);
    text-align: center;
    padding: 50px;
}

.container {
    background: white;
    padding: 20px;
    border-radius: 10px;
    width: 300px;
    margin: auto;
    box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
}

.search-box {
    display: flex;
    gap: 10px;
    justify-content: center;
}

input {
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
    width: 60%;
}

button {
    padding: 10px;
    background: #3498db;
    color: white;
    border: none;
    cursor: pointer;
    border-radius: 5px;
}

button:hover {
    background: #2980b9;
}

.weather-info {
    margin-top: 20px;
}

.weather-info img {
    width: 100px;
}
```

Step 6: Fetching Weather Data with JavaScript

The `script.js` file:

1. **Captures user input** when the button is clicked.
2. **Sends an API request** to OpenWeatherMap.
3. **Extracts important weather data** from the JSON response.
4. **Displays the data dynamically** on the webpage.

```
js

// script.js

const API_KEY = 'YOUR_API_KEY'; // Replace with your OpenWeatherMap API key

document.getElementById('searchBtn').addEventListener('click', () => {
    const city = document.getElementById('city').value;
    if (city) {
        getWeather(city);
    } else {
        alert('Please enter a city name');
    }
});

async function getWeather(city) {
    const url = `https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${API_KEY}&units=m

    try {
        const response = await fetch(url);
        if (!response.ok) {
            throw new Error('City not found');
        }
        const data = await response.json();

        // Extract relevant information
        const cityName = data.name;
        const temp = data.main.temp;
        const description = data.weather[0].description;
        const icon = `https://openweathermap.org/img/wn/${data.weather[0].icon}.png`;

        // Display in the UI
        document.getElementById('city-name').textContent = cityName;
        document.getElementById('temperature').textContent = `Temperature: ${temp}°C`;
        document.getElementById('description').textContent = `Condition: ${description}`;
        document.getElementById('weather-icon').src = icon;
    } catch (error) {
        alert(error.message);
    }
}
```

Step 7: How the Code Works

1. Capturing User Input

- The `searchBtn` button listens for a click.
- When clicked, it fetches the text inside the input box.

2. Making API Request

- The `getWeather(city)` function builds the API URL and uses `fetch()` to request data.
- `await response.json()` converts the response to JSON.

3. Handling Errors

- If the response is not OK, an error message is shown.

4. Extracting Weather Data

- `data.main.temp`: Temperature
- `data.weather[0].description`: Weather condition (e.g., "clear sky")
- `data.weather[0].icon`: Fetches the corresponding weather icon.

5. Updating the DOM

- `document.getElementById('city-name').textContent = cityName;` updates the city name.
- Similarly, temperature, description, and weather icon are updated dynamically.

Step 8: Running the Project

1. Save the files in a folder.
2. Replace `YOUR_API_KEY` with your OpenWeatherMap API key.
3. Open `index.html` in a browser.
4. Enter a city name and get the weather details.

Bonus Enhancements

You can improve this app further: ✅ **Auto-detect user location** and fetch weather using `navigator.geolocation`.
✅ **Add background change** based on temperature (blue for cold, red for hot).
✅ **Display additional info** like humidity and wind speed.
✅ **Use local storage** to remember the last searched city.

Final Code

index.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Weather App</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <h1>Weather App</h1>
        <div class="search-box">
            <input type="text" id="city" placeholder="Enter city name">
            <button id="searchBtn">Search</button>
        </div>
        <div class="weather-info">
            <h2 id="city-name"></h2>
            <p id="temperature"></p>
            <p id="description"></p>
            <img id="weather-icon" src="" alt="">
        </div>
    </div>
    <script src="script.js"></script>
</body>
</html>
```

style.css

(Same as above)

script.js

(Same as above, replace `YOUR_API_KEY` with actual API key)

This project will help you **learn real-world API integration, JSON handling, and UI manipulation**! 🚀
Let me know if you need enhancements. 😊