**Overview**

This documentation covers the setup and functionality of a simple weather application that fetches weather data based on user input. The app displays current weather conditions along with an appropriate image representing the weather.

## Table of Contents

1. Technologies Used
2. File Structure
3. HTML Setup
4. CSS Styling
5. JavaScript Functionality
6. Image Assets
7. Creating a Folder on GitHub
8. Contributing

## Technologies Used

* HTML5
* CSS3
* JavaScript (ES6)
* GitHub (for version control)

## File Structure

bash

/weather-app

├── index.html # Main HTML file

├── styles.css # CSS styles

├── script.js # JavaScript functionality

├── /images # Folder for weather images

├── mostly-cloudy.jpg

├── partly-cloudy.jpg

├── cloudy.jpg

├── fair.jpg

├── rainy.jpg

├── snowy.jpg

## HTML Setup

Create an index.html file with the following content:

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="styles.css">

</head>

<body>

<div class="container">

<h1>Current Weather</h1>

<label for="cityInput">City:</label>

<input type="text" id="cityInput" placeholder="Enter city..." aria-label="Enter city name">

<button id="checkTimeBtn">Check Weather</button>

<h2 id="result" style="margin-top: 20px;"></h2>

</div>

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

</body>

</html>

## CSS Styling

Create a styles.css file with the following content:

css

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: Arial, sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background-image: url('pngtree-white-cloud-on-blue-sky-weather-background-image\_410050.jpg');

background-size: cover;

background-position: center;

color: white;

}

.container {

text-align: center;

background: rgba(0, 0, 0, 0.5);

padding: 20px;

border-radius: 10px;

}

input {

padding: 10px;

margin: 10px;

border: none;

border-radius: 5px;

width: 200px;

}

input:focus {

outline: none;

border: 2px solid #007bff;

}

button {

padding: 10px 15px;

border: none;

border-radius: 5px;

cursor: pointer;

background-color: #007bff;

color: white;

transition: background-color 0.3s;

}

button:hover {

background-color: #0056b3;

}

button:disabled {

background-color: #555;

cursor: not-allowed;

}

h1, h2 {

margin: 10px 0;

}

## JavaScript Functionality

Create a script.js file with the following content:

javascript

document.getElementById('checkTimeBtn').addEventListener('click', async () => {

const city = document.getElementById('cityInput').value.trim();

if (!city) {

document.getElementById('result').innerText = "Please enter a city name.";

return;

}

const url = "https://yahoo-weather5.p.rapidapi.com/weather";

const headers = {

"x-rapidapi-key": "YOUR\_API\_KEY\_HERE",

"x-rapidapi-host": "yahoo-weather5.p.rapidapi.com"

};

document.getElementById('result').innerText = "Fetching data...";

try {

const response = await fetch(`${url}?location=${encodeURIComponent(city)}&format=json&u=f`, {

method: 'GET',

headers: headers

});

if (!response.ok) {

throw new Error(`HTTP error! status: ${response.status}`);

}

const weatherData = await response.json();

if (weatherData.location && weatherData.current\_observation) {

const location = weatherData.location.city;

const temperature = weatherData.current\_observation.condition.temperature;

const condition = weatherData.current\_observation.condition.text;

const conditionToImage = {

"Mostly Cloudy": "images/mostly-cloudy.jpg",

"Partly Cloudy": "images/partly-cloudy.jpg",

"Cloudy": "images/cloudy.jpg",

"Fair": "images/fair.jpg",

"Rainy": "images/rainy.jpg",

"Snowy": "images/snowy.jpg"

};

document.getElementById('result').innerText = `Weather in ${location}: ${temperature}°F, ${condition}`;

const imageUrl = conditionToImage[condition] || "default-image.jpg";

const weatherImage = document.createElement('img');

weatherImage.src = imageUrl;

weatherImage.alt = condition;

weatherImage.style.width = '300px';

weatherImage.style.marginTop = '10px';

const resultDiv = document.getElementById('result');

resultDiv.innerHTML = '';

resultDiv.appendChild(weatherImage);

resultDiv.prepend(document.createTextNode(`Weather in ${location}: ${temperature}°F, ${condition}`));

} else {

document.getElementById('result').innerText = 'Weather data not available. Please try a different city.';

}

} catch (error) {

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

document.getElementById('result').innerText = 'Error fetching weather data. Please check your connection and try again.';

}

});

## Image Assets

Ensure you have the following images saved in the images folder:

* mostly-cloudy.jpg
* partly-cloudy.jpg
* cloudy.jpg
* fair.jpg
* rainy.jpg
* snowy.jpg