WEATHER APP

Name: K.R.DeviSree

Internship Project

Abstract

The **Weather App** is a web-based application designed to provide real-time weather updates for any location around the world. Built using HTML, CSS, and JavaScript, the app leverages data from a public weather API to display current weather conditions such as temperature, humidity, wind speed, and weather descriptions. The user-friendly interface allows users to enter a city name and instantly view the relevant weather information, enhancing accessibility and user engagement.

This project aims to demonstrate the integration of third-party APIs in modern web development and showcases how front-end technologies can be used to fetch, process, and display real-time data. The application is responsive and compatible with various devices, making it practical for everyday use. By combining aesthetic design with functional programming, the Weather App exemplifies a real-world solution built through core web development skills.

Introduction

Weather plays a vital role in our daily lives, influencing everything from personal plans to business operations. With the advancement of technology, accessing real-time weather information has become more convenient and reliable. The **Weather App** is a simple yet powerful web application developed to provide users with accurate and up-to-date weather information based on their input location.

This project is built using fundamental web technologies — **HTML**, **CSS**, and **JavaScript** — and integrates a third-party **weather API** to fetch live data. The app allows users to enter the name of a city, after which it displays the current temperature, weather condition, humidity, wind speed, and other relevant information in a visually appealing format.

The purpose of developing this application is to understand and implement concepts like API integration, asynchronous data handling, and responsive web design. It also highlights how front-end technologies can be utilized to solve real-world problems by creating interactive and informative user experiences.

Objectives

1. **To Develop a Real-Time Weather Application**  
   Create a functional and user-friendly web application that can display current weather conditions for any user-specified location.
2. **To Integrate a Public Weather API**  
   Utilize a third-party API (such as OpenWeatherMap) to fetch and display live weather data.
3. **To Practice Front-End Web Development**  
   Apply HTML, CSS, and JavaScript skills to design and develop an interactive and responsive user interface.
4. **To Understand API Integration and Data Handling**  
   Gain hands-on experience with fetching external data asynchronously using JavaScript and displaying it dynamically in the web interface.
5. **To Enhance User Experience**  
   Ensure that the app is intuitive, visually appealing, and responsive across different devices and screen sizes.

**Code Implementation**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>Weather App</title>

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

</head>

<body>

<div class="weather-container">

<h2>Weather App</h2>

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

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

<div class="result" id="result"></div>

<div class="error" id="error"></div>

</div>

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

</body>

</html>

CSS

body {

font-family: Arial, sans-serif;

background: linear-gradient(to right, #83a4d4, #b6fbff);

margin: 0;

padding: 0;

display: flex;

height: 100vh;

justify-content: center;

align-items: center;

}

.weather-container {

background: white;

padding: 2rem;

border-radius: 12px;

box-shadow: 0 8px 20px rgba(0,0,0,0.1);

width: 300px;

text-align: center;

}

input[type="text"] {

width: 80%;

padding: 0.5rem;

font-size: 1rem;

border-radius: 6px;

border: 1px solid #ccc;

margin-bottom: 1rem;

}

button {

padding: 0.5rem 1rem;

font-size: 1rem;

border: none;

border-radius: 6px;

background-color: #3498db;

color: white;

cursor: pointer;

}

button:hover {

background-color: #2980b9;

}

.result {

margin-top: 1rem;

}

.error {

color: red;

margin-top: 1rem;

}

JavaScript

document.getElementById('getWeatherBtn').addEventListener('click', getWeather);

async function getWeather() {

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

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

const errorDiv = document.getElementById('error');

resultDiv.innerHTML = '';

errorDiv.innerHTML = '';

if (!city) {

errorDiv.textContent = 'Please enter a city name.';

return;

}

const apiKey = '9f0e6511a34d88112cffb6009f6fb60a'; // Replace with your OpenWeatherMap API key

const url = `https://api.openweathermap.org/data/2.5/weather?q=${encodeURIComponent(city)}&units=metric&appid=${apiKey}`;

try {

const response = await fetch(url);

if (!response.ok) {

throw new Error('City not found');

}

const data = await response.json();

resultDiv.innerHTML = `

<h3>${data.name}, ${data.sys.country}</h3>

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

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

<p><strong>Humidity:</strong> ${data.main.humidity}%</p>

<p><strong>Wind Speed:</strong> ${data.wind.speed} m/s</p>

`;

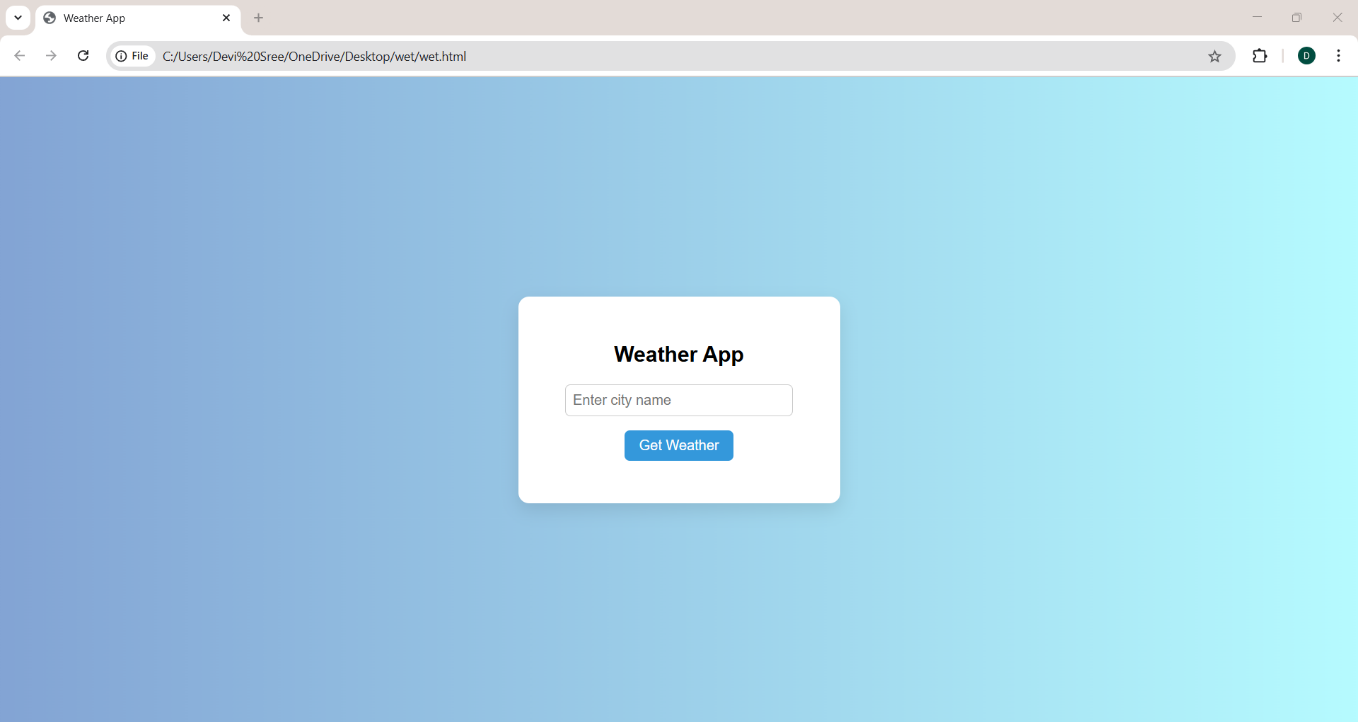
} catch (error) {

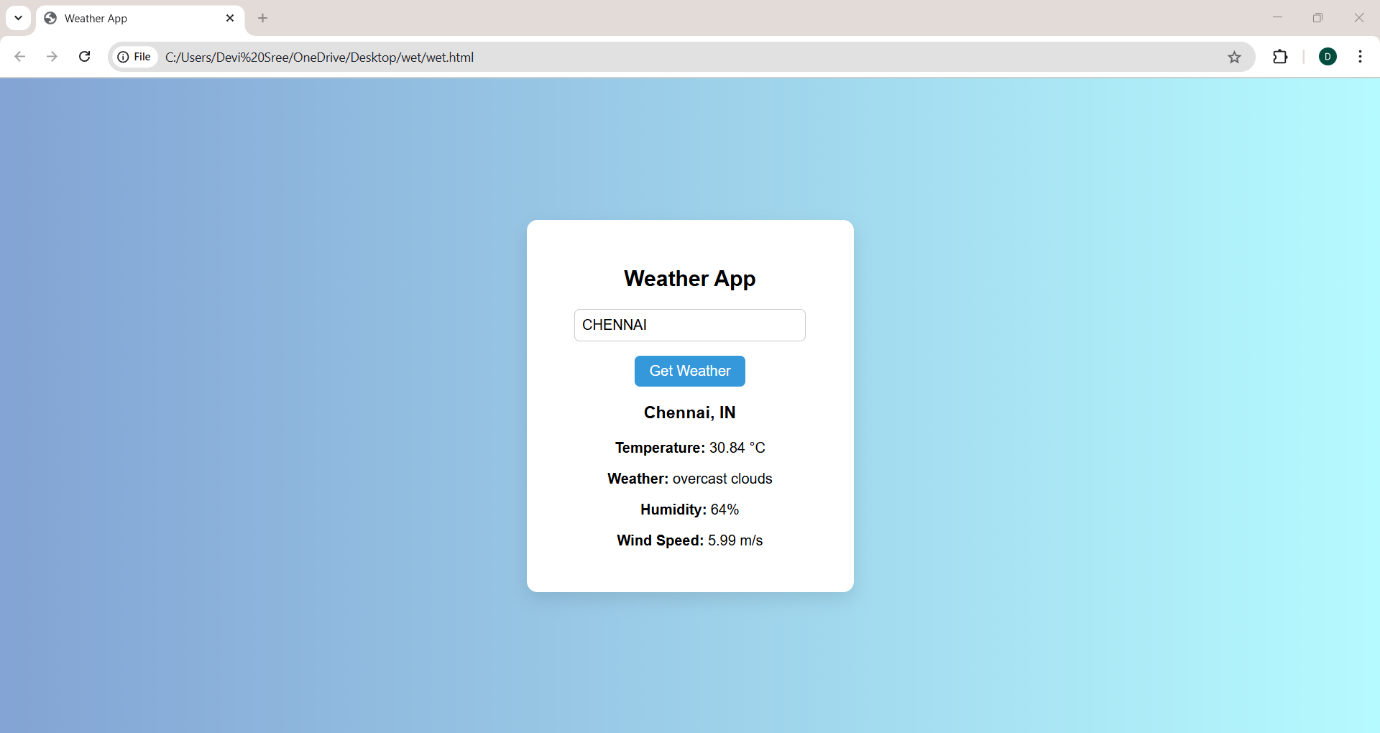
errorDiv.textContent = error.message;

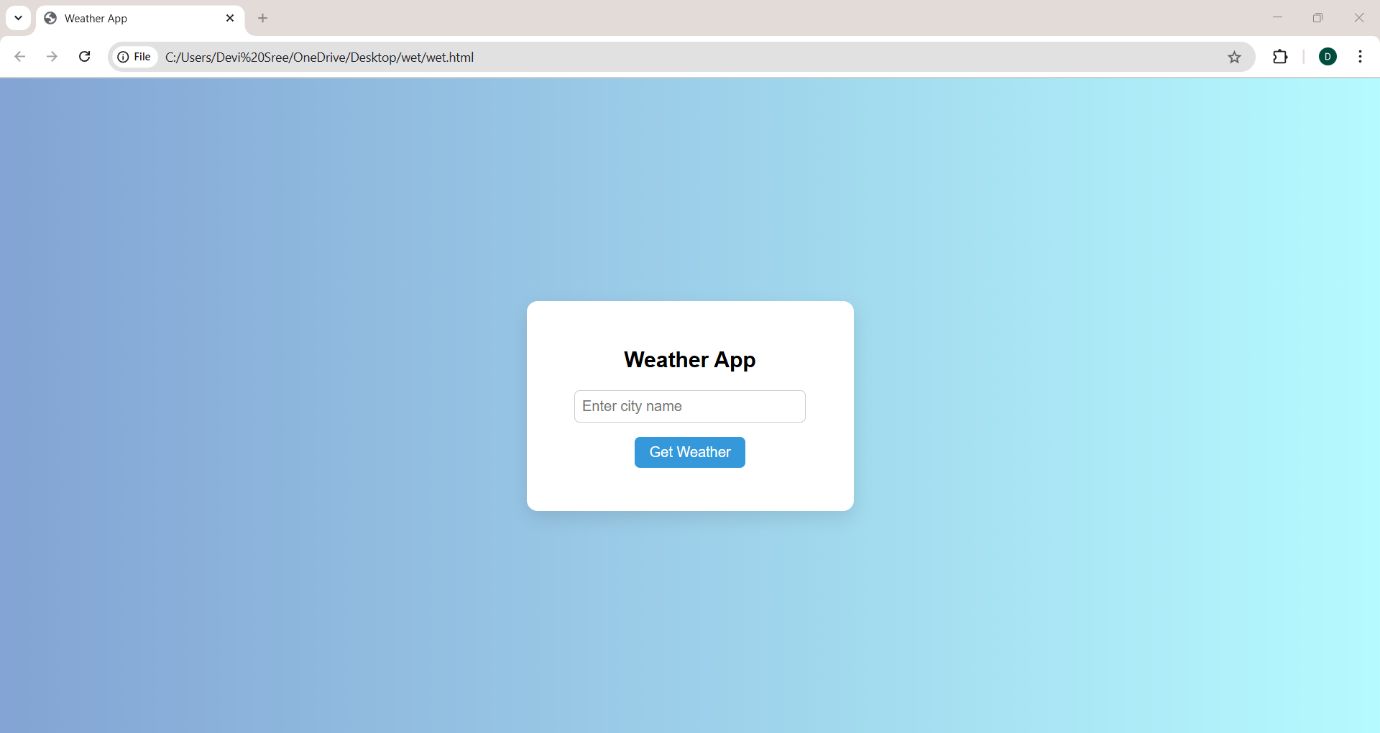
}

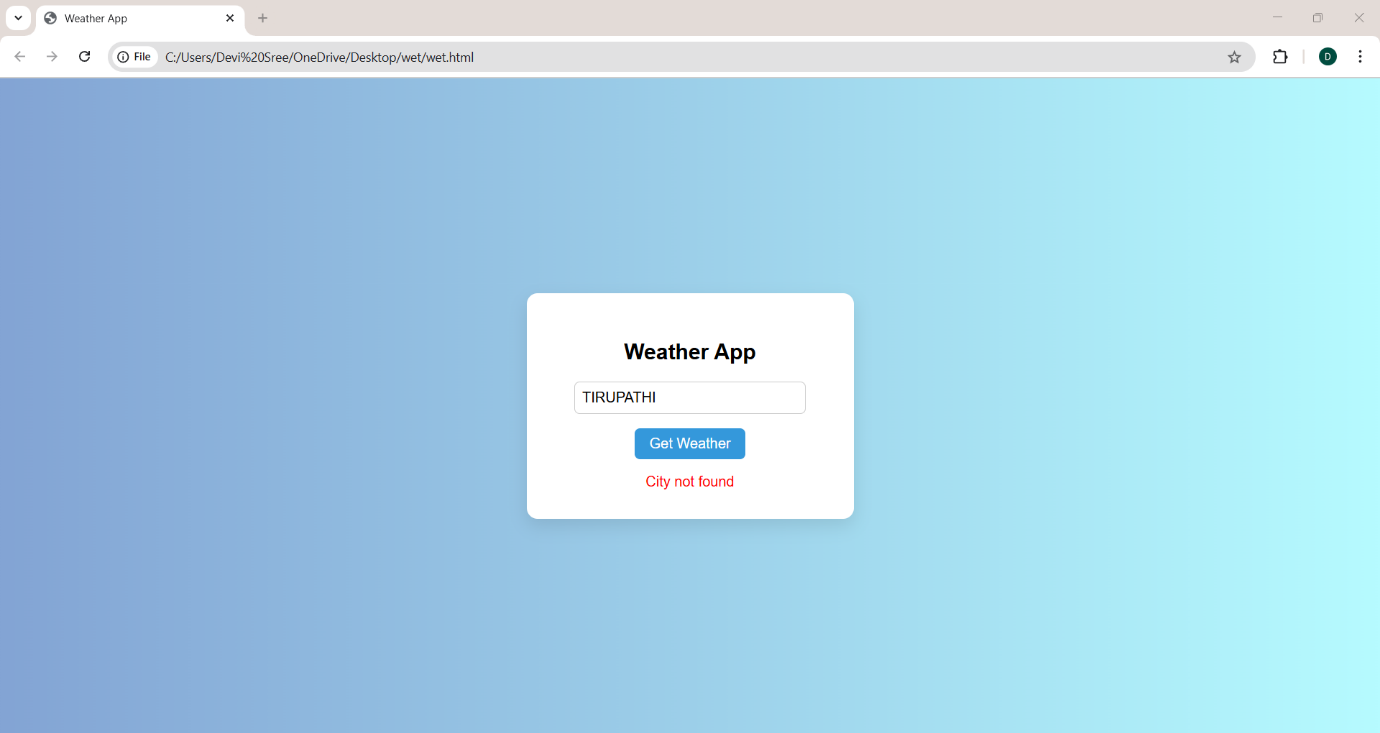
}

OUTPUT









CONCLUSION

The Weather App project successfully demonstrates the integration of real-time weather data with a user-friendly interface. By utilizing APIs, the app provides accurate and up-to-date weather information for any location worldwide. This project enhanced my understanding of asynchronous programming, API consumption, and responsive web design. Overall, the Weather App serves as a practical and interactive tool that can be further expanded with additional features such as forecasts, alerts, and user preferences. This internship project has strengthened my skills in web development and API integration, preparing me for more complex applications in the future.