Wheather App

MINOR PROJECT FILE

SCAN ME

Project Details

Project Name : WHEATHER-APP

PROJECT DESCRIPTION :The primary objective of the Weather App is to provide users with an intuitive and accessible platform to retrieve real-time weather information effortlessly. The application offers key weather details for the user's current location or any city they search for, including temperature, windspeed, humidity, and cloudiness. The app is designed to be a reliable, one-stop solution for daily weather needs, with a focus on simplicity and convenience.

Group members :

AZAD (10622057)

Faculty Name : Mr. TK BHAGAT

MINOR PROJECT

Wheather App

Introduction

In today's fast-paced world, staying up-to-date with weather conditions is essential for planning daily activities. However, accessing accurate weather information often requires navigating through multiple websites or apps. The Weather App aims to solve this problem by offering a simple, unified platform that provides real-time weather updates at the user's fingertips . Whether you need weather data for your current location or want to search for conditions in another city, the Weather App streamlines the process by offering all of these features in one place.

Developed using web technologies like HTML, CSS, and JavaScript, the .Weather App integrates with the OpenWeather Map API to provide accurate, real-time data, including temperature, humidity, wind speed, cloud cover, and more. The app's user-friendly design ensures that anyone, regardless of technical expertise, can easily navigate through its features. Whether you're a student looking to plan outdoor activities, a professional monitoring weather for work, or simply a casual user interested in daily weather updates, the Weather App is designed to meet the needs of all. By eliminating the need for multiple apps or complex processes, this platform offers a seamless experience, ensuring that
weather information is just a click away.

Description: The Weather App is a dynamic, user-friendly web platform that provides real-time weather updates and forecast information for users worldwide. Built with a responsive interface using HTML, CSS, and JavaScript, this project leverages weather APIs to deliver accurate and up-to-date data. Designed with ease of use in mind, it allows users to quickly check current weather conditions and plan their activities accordingly.

Key Features:

Real-Time Weather Updates:

1. Displays current temperature, humidity, wind speed, and atmospheric pressure.
2. Provides location-based weather data by accessing the user's geolocation or a manually entered city name.

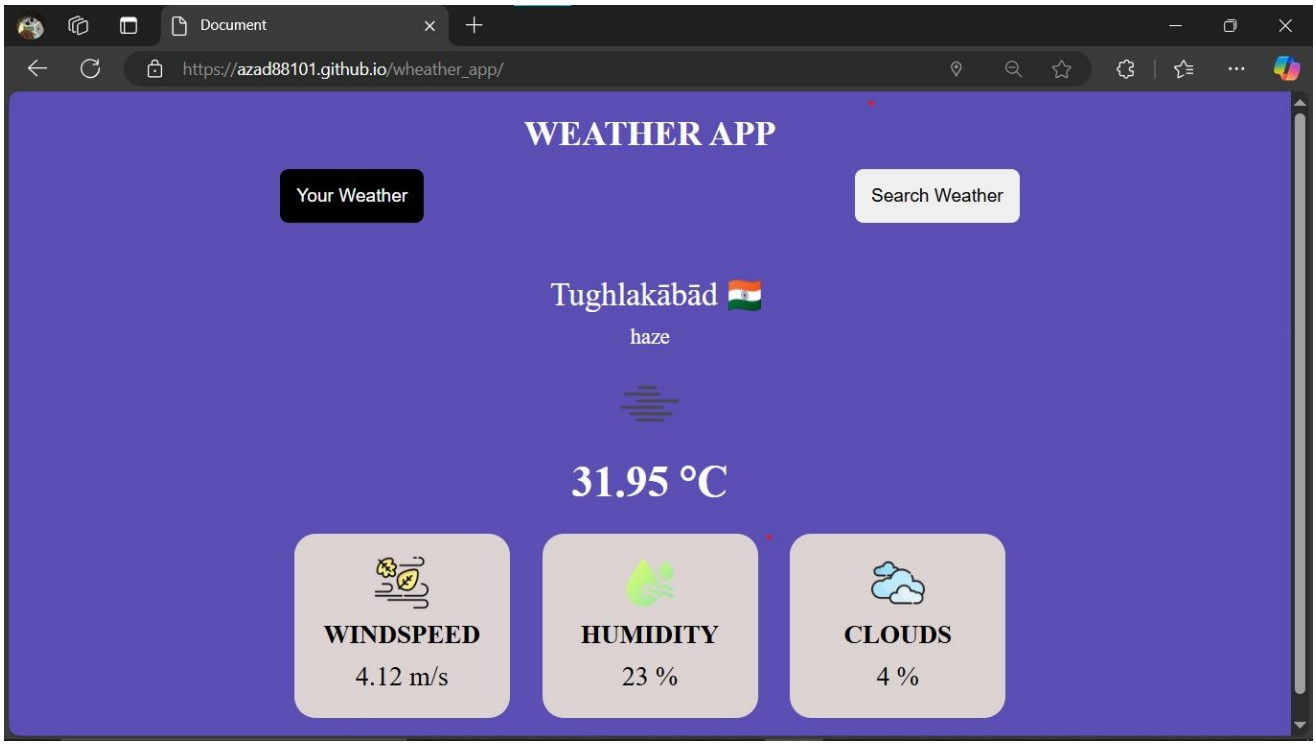
User-Friendly Interface:

1. Features a clean, responsive layout for easy navigation across various devices.
2. Optimized for mobile and desktop screens, ensuring a consistent user experience.

Search Functionality:

1. Allows users to search weather information by city, expanding its usability for travelers or individuals planning trips
2. Supports multiple city entries for a quick comparison of weather across locations.

HOME PAGE



The homepage presents a clean and minimalist design with a focus on providing essential weather information at a glance. The dark purple background creates a modern and sophisticated feel, while the use of white text and icons ensures readability.

Layout and Structure:

Header: The header section prominently displays the title "WEATHER APP" in bold white text, immediately establishing the purpose of the application.

Main Content: The main content area is divided into two sections

Your Weather: This section displays weather information for the user's current location.

It includes Location: The city and country name (e.g., Tughlakabad, India) are shown in white text.

Weather Condition: A brief description of the current weather (e.g., "haze") is displayed in white text.

Temperature: The current temperature is shown in a large, bold white font.

Weather Details: Three smaller boxes display additional weather information

1. Wind Speed: The wind speed is shown with a wind icon and the value (e.g., 4.12 m/s).
2. Humidity: The humidity level is displayed with a water droplet icon and the percentage (e.g., 23%).
3. Clouds: The cloud coverage is shown with a cloud icon and the percentage (e.g., 4%).

Search Weather: This section allows users to search for weather information for a specific location. It includes a search bar where users can enter the city name.

Design Elements:

Color Palette: The dark purple background provides a calming and professional look. White text and icons are used for high contrast and readability.

Typography: The font choices are clear and easy to read, with a larger font size for the temperature and a smaller size for other details.

Icons: Simple and recognizable icons are used to represent wind speed, humidity, and clouds.

Layout: The layout is well-organized and easy to navigate, with clear separation between the"Your Weather" and "Search Weather" sections.

Functionality:

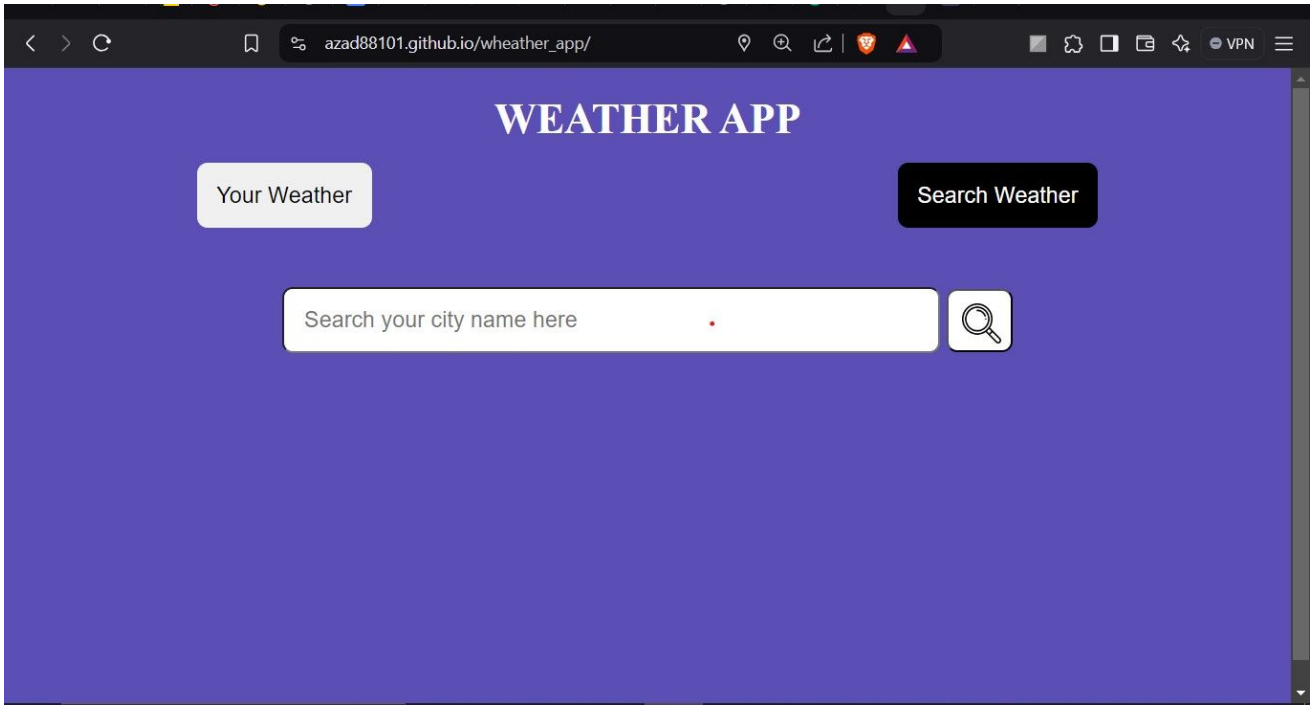
Real-time Weather Updates: The app displays current weather information for the user's location.

Location-Based Weather: The app automatically detects the user's location or allows them to manually enter a location.

Weather Details: The app provides detailed weather information, including temperature, wind speed, humidity, and cloud coverage.

Search Functionality: Users can search for weather information for anylocation by entering the city name.

Search page



"WEATHER APP" is displayed prominently at the top, indicating the purpose of the application.

Key Elements:

Buttons:

1. "Your Weather": This button likely displays weather inform for the user's current location.
2. "Search Weather": This button allows the user to search for weather information for a specific location.

Search Bar:

The search bar is located below the buttons and is used to input the name of the city for which the user wants to know the weather.

A magnifying glass icon to the right of the search bar indicat that it is used for searching.

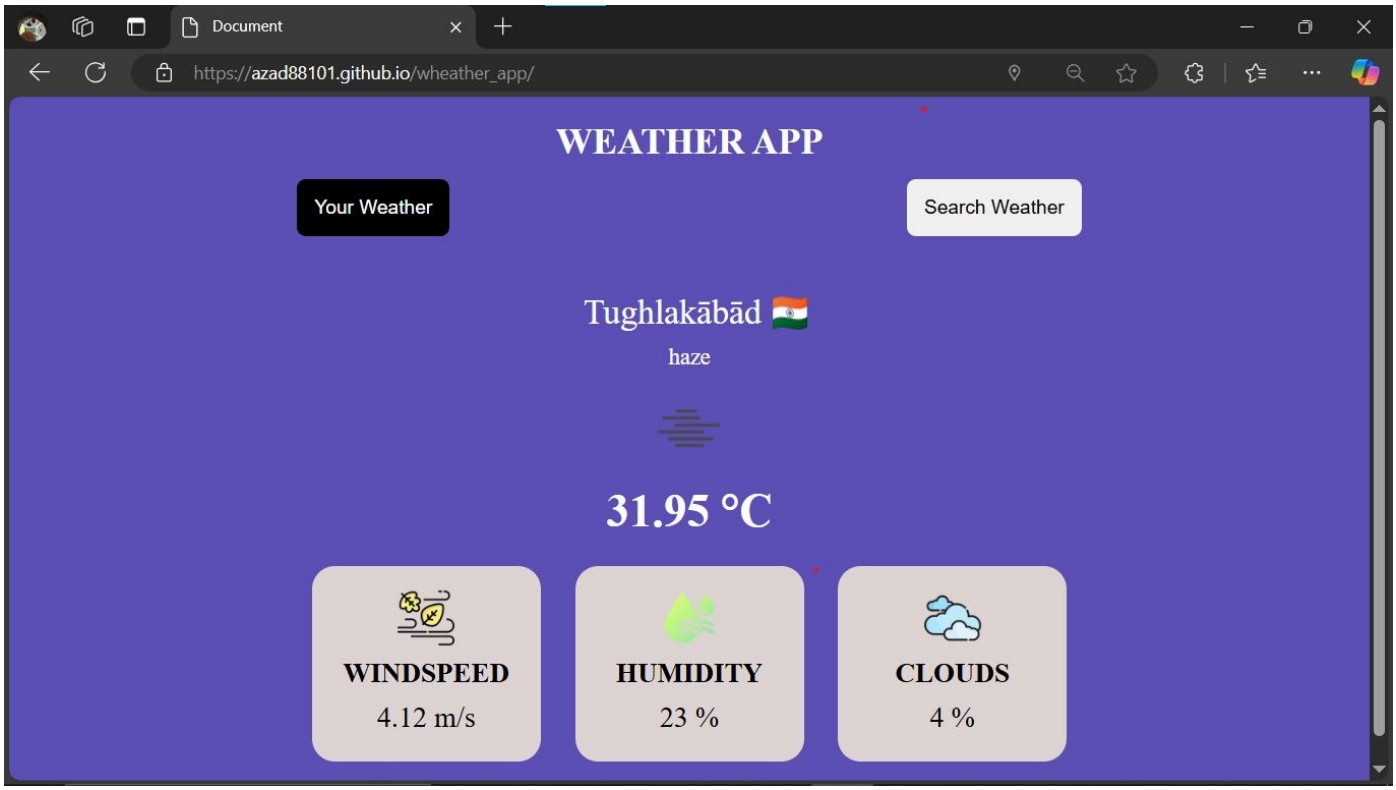
Design:

The overall design is simple and clean, with a dark purple background and white text.

The use of a dark background with light text enhances readability and provides a modern look.

This page perfectly aligns with the minimalist and functional design philosophy of the **Wheather-App** platform.

About



key Features

1. Title:

1.The page is clearly titled "Weather Forecast", immediately letting users know that they can access real-time weather data here.

2.Location Input Field:

1.Users are provided with an input field where they can enter a city or location name to get the weather forecast.
2.This field is designed to accept valid location names, ensuring easy and intuitive usability.

3.Get Weather Button:

1.After entering a location, users can click the “Get Weather” button to initiate the weather retrieval process.
2.The button is prominently displayed, making it easy for users to find and use.

Real-Time Weather Data:

1.Upon submission, the app fetches weather details from an API, providing live data such as temperature, humidity, wind speed, and weather conditions.
2.The process is fast, allowing users to quickly view the latest weather information.

Dynamic Weather Icons:

The app includes weather icons that change based on the current weather condition (e.g., sunny, rainy, cloudy), making the forecast more visually engaging.

Design and User Experience:

1.User-Friendly Interface: The layout is simple and clear, allowing users to navigate and use the app without any prior experience.

2.Responsive Design: The page is optimized for both desktop and mobile devices, ensuring accessibility across various platforms.

3.Quick Data Retrieval: The app is designed for minimal wait time, allowing users to quickly access up-to-date weather information.

Benefits:

1.Real-Time Weather Updates: Users receive immediate information on the current weather conditions and forecast helping them plan their day accordingly.

2.Wide Compatibility: The app is compatible with multiple devices, enabling users to check the weather from their phones, tablets, or computers.

3.No Installation Required: Since the app operates online, users don’t need to install additional software to access weather data.

Project Structure Overview

1. **`Index.html`**

- This is the main HTML file for the front-end interface of the weather app.
- It serves as the primary page where users can interact, input locations, and view weather data.
- The HTML structure is designed to display real-time weather information fetched from an external API, with sections for location-based data, weather details, and styling hooks.

2. **`assets(IMAGES)/`**

- This folder IMAGES holds any static files such as images, icons, or background images that enhance the app's visual presentation.

3. Style.Css

- This file provides styling to make the weather app visually appealing and consistent.
- It contains CSS rules for layout, color schemes, fonts, and responsive design elements, ensuring the app looks good on various devices.

4.Script.js

- This JavaScript file handles the client-side logic and interactions of the app.
- It includes functions to fetch data from the weather API, process the response, and dynamically update the HTML elements with real-time weather information.
- It may also include error handling for API requests and manage other interactive elements on the page.

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" /
    <title>Document</title>
    <link rel="stylesheet" href="style.css" />
    <link rel="shortcut icon" href="/images/clouds.png" type=
"image/x-icon" />
  </head>
  <body>
    <header>
      <div class="header">
        <div><h1>WEATHER APP</h1></div>
        <div class="pagebutton">
          <button class="pagebtn" id="home-pagebtn">Your Weather
            </button>

        <button class="pagebtn" id="search-pagebtn">Search Weather</button>
        </div>
      </div>
    </header>

    <!-- creation of search page  -->

    <div class="search-page" id="sp">
      <input
        type="text"
        placeholder="Search your city name here "
        id="search-bar"
      />
<button class="search-btn" id="search-city">
        <img src="images/search.png" id="search-icon" />
  </button>
    </div>
```

```
<div class="weather-page" id="wp">
    <div class="city">
      <p id="city-data"></p>
      <img src="" id="country-icon" />
    </div>

    <div class="wheather-condition">
      <p id="wheather-type"></p>
      <img src="" alt="" id="wheather-icon" />
      <h1 id="temperature"></h1>
    </div>

    <div class="cards">
      <div class="innerCards">
        <img src="images/wind.png" alt="" class="cards-icon" />
        <h2>WINDSPEED</h2>
        <p class="card-text" id="wind"></p>
      </div>
<div class="innerCards">
        <img src="images/humidity.png" alt="" class="cards-icon" />
        <h2>HUMIDITY</h2>
        <p class="card-text" id="humidity"></p>
      </div>

      <div class="innerCards">
        <img src="images/clouds.png" alt="" class="cards-icon" />
        <h2>CLOUDS</h2>
        <p class="card-text" id="clouds"></p>
      </div>
 </div>
  </div>

  <!-- end wheather app -->

  <!-- grant location access -->

  <div class="location-page" id="locationpage">
    <img src="images/location.png" id="location-icon" />
```

```
    <div class="location-para">
      <p>Grant location Acesss</p>
      <p>Allow to get weather Information</p>
    </div>

    <button id="location-btn">Grant Access</button>
    <p id="message"></p>
  </div>

  <!-- loading page  -->

  <div class="loading-conatiner" id="loadingconatiner">
    <img id="loading-gif" src="images/loading.gif" />
    <h1>Loading ...</h1>
  </div>

  <!-- error page  -->
  <div class="error-page" id="ep">
    <h1 id="error-text">404 NOT FOUND <br /></h1>
    <img id="error-gif" src="images/error.gif" alt="" />
  </div>
 <script src="index.js"></script>
</body>
</html>
```

STYLE.CSS

```
* {
  margin: 0%;
  padding: 0%;
  box-sizing: border-box;
}

body {
  height: 100vh;
  background-color: rgb(91, 78, 180);
  padding-bottom: 20px;
}

.pagebtn.current-page {
  background-color: rgb(0, 0, 0);
  color: white;
}

.header {
  display: flex;
  flex-direction: column;
  justify-content: space-between;
  align-items: center;
  margin: 20px;
}

/* weather app */
.weather-page {
  display: none;
}

.weather-page.active {
  display: flex;
  flex-direction: column;
  align-items: center;
  padding: 20px;
}

h1 {
  color: white;
}
```

```
.cards {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 30px;
}

.innerCards {
  height: 170px;
  width: 200px;
  display: flex;
  flex-direction: column;
  align-items: center;
  background-color: rgb(221, 211, 211);
  border-radius: 20px;
  padding: 10px;
}

.cards-icon {
  height: 50px;
  width: 50px;
  margin: 10px;
}

#country-icon {
  height: 30px;
  width: 30px;
}

.pagebutton {
  display: flex;
  grid-gap: 400px;
}

.pagebtn {
  font-size: 17px;
  padding: 15px;
  margin-top: 15px;
  border-radius: 8px;
  border: none;
}
/* location  */
.location-page {
  display: none;
}
```

```
.location-page.active {
  display: flex;
  flex-direction: column;
  align-items: center;
  padding: 20px;
}

#location-icon {
  height: 200px;
  width: 200px;
}

#location-btn {
  font-size: 17px;
  padding: 15px;
  margin: 20px;
  border-radius: 8px;
}

.location-para {
  font-size: 30px;
  color: white;
  display: flex;
  flex-direction: column;
  align-items: center;
  padding: 25px;
}
/* search bar  */
.search-page {
  display: none;
}

.search-page.active {
  display: flex;
  justify-content: center;
  align-items: center;
  padding: 25px;
}

.search-btn {
  height: 48px;
  width: 50px;
  background-color: white;
  border-radius: 8px;
  margin-left: 5px;
}
```

```
#search-icon {
  height: 30px;
  width: 30px;
  margin: 10px;
}

#search-bar {
  height: 50px;
  width: 500px;
  border-radius: 8px;
  font-size: 17px;
  padding: 15px;
}

/* loading page */

.loading-conatiner {
  display: none;
}

.loading-conatiner.active {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  padding: 25px;
  margin: 20px;
}
/* //API DATA  */
.city {
  display: flex;
  justify-content: center;
  align-items: center;
}

#city-data {
  font-size: 30px;
  margin: 10px;
  color: white;
}

.wheather-condition {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
```

```
color: white;
  font-size: 20px;
}

#temperature {
  margin-bottom: 25px;
}

.card-text {
  font-size: 25px;
  margin: 10px;
}

#message {
  font-size: 25px;
  margin: 10px;
  color: white;
  padding: 10px;
  border-radius: 10px;
}

/* errorpage */
.error-page {
  display: none;
}

.error-page.active {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  color: white;
  font-size: 20px;
}

#error-text {
  font-size: 40px;
  padding: 10px;
  border-radius: 10px;
  margin-top: 25px;
}

#error-gif {
  height: 300px;
  width: 400px;
}
```

Index.js

```
//pages
const searchpage = document.getElementById("sp");
const weatherpage = document.getElementById("wp");
const loadingpage = document.getElementById("loadingconatiner");
const locationpage = document.getElementById("locationpage");
const errorpage = document.getElementById("ep");

//header
const searchbtn = document.getElementById("search-pagebtn");
const homebtn = document.getElementById("home-pagebtn");

//weather app

//location pages
const locationbtn = document.getElementById("location-btn");
const messageDiv = document.getElementById("message");

//search bar
const inputbar = document.getElementById("search-bar");
const searchcity = document.getElementById("search-city");

// api key
const Api = "99c3df48061950d0ab6187a7e4f8f495";

let permission = "deny";

let currentpage = homebtn;

currentpage.classList.add("current-page");

locationpage.classList.add("active");

getLocation();

function changepage(page) {
  if (currentpage != page) {
    currentpage.classList.remove("current-page");
    currentpage = page;

    currentpage.classList.add("current-page");
  }
}
```

```
function yourpage() {
  if (permission == "deny") {
    locationpage.classList.add("active");
    weatherpage.classList.remove("active");
    searchpage.classList.remove("active");
    errorpage.classList.remove("active");
  } else {
    getLocation();
    weatherpage.classList.add("active");
    searchpage.classList.remove("active");
    errorpage.classList.remove("active");
  }
}

function searchpagebtn() {
  weatherpage.classList.remove("active");
  searchpage.classList.add("active");
  locationpage.classList.remove("active");
  errorpage.classList.remove("active");
}

homebtn.addEventListener("click", () => {
  changepage(homebtn);
  yourpage();
});
searchbtn.addEventListener("click", () => {
  changepage(searchbtn);
  searchpagebtn();
});

function definedata(data) {
  const city = document.getElementById("city-data");
  const wheather = document.getElementById("wheather-type");
  const wheathericon = document.getElementById("wheather-icon");
  const temperature = document.getElementById("temperature");

  const wind = document.getElementById("wind");
  const humidity = document.getElementById("humidity");
  const clouds = document.getElementById("clouds");
  const countryicon = document.getElementById("country-icon");

  city.innerText = data?.name;
  wheather.innerText = data?.weather?.[0]?.description;
  const tempkelvin = data?.main?.temp;
  temperature.innerText = (tempkelvin - 273.15).toFixed(2) + " °C";
```

```
// Call other functions

        permission = "allow";

        locationpage.classList.remove("active");
        weatherpage.classList.add("active");

        yourweather(latitude, longitude);
        senddata(latitude, longitude);
      },
      (error) => {
        // Set error message
        messageDiv.style.backgroundColor = "red";
        messageDiv.innerText = `Error: ${error.message}`;
      }
    );
  } else {
    console.log("Geolocation is not supported by this browser.");
  }
}

navigator.permissions
  .query({ name: "geolocation" })
  .then((permissionStatus) => {
    if (permissionStatus.state === "granted") {
      // getLocation();
      permission = "allow";
      yourpage();
      // } else if (permissionStatus.state === "prompt") {
      //   console.log("Permission prompt expected.");
      //   getLocation(); // Will prompt the user to allow/deny location
    } else {
      // messageDiv.style.backgroundColor = "red";
      // messageDiv.innerText = `Location permission denied.`;
    }
  });
async function yourweather(lat, lon) {
  weatherpage.classList.remove("active");

  loadingpage.classList.add("active");
  let reponse = await fetch(
    `https://api.openweathermap.org/data/2.5/weather?lat=${lat}&lon=${lon}&appid=
  );

  let data = await reponse.json();
  console.log(data);

  loadingpage.classList.remove("active");
```

Technologies and Tools Used

The **WHEATHER APP** project leverages a variety of modern technologies and libraries to deliver its diverse functionality:

1. **HTML5 :**

   • The backbone of the app's structure, HTML5 enables a clean and accessible layout for presenting content and forms, ensuring compatibility across different devices and screen sizes.

2. **CSS3** :

   - The app's styling is powered by CSS3, which is used to create an attractive and responsive design. Features like flexible grid layouts and media queries ensure the app adapts well to mobile and desktop screens alike.

3. **JavaScript:**

   - JavaScript is central to the app's interactivity, used for handling dynamic features like fetching real-time weather data from APIs, managing user inputs, and updating the UI based on the current location or search query. It ensures the app is responsive and user-friendly.

4. **Weather API :**

   • To provide accurate and up-to-date weather information, the app integrates with weather APIs like OpenWeatherMap or WeatherAPI. These APIs deliver comprehensive weather data, such as temperature, wind speed, humidity, and weathe conditions.

Future Improvements

While the Weather App is already an efficient and user-friendly platform for real-time weather information, several improvements can be implemented to increase its functionality, enhance user engagement, and meet evolving user needs:

1. **Personalized Weather Alerts and Notifications**

    • Introducing personalized weather alerts would allow users to receive notifications about significant weather changes in their preferred locations. This feature could be tailored to specific conditions such as storms, heavy rain, or extreme temperatures, providing timely and relevant updates. This enhancement would increase user satisfaction by offering real-time, location-specific notifications.

2. **User Accounts and Saved Locations**:

    • Adding user authentication would allow users to create accounts and store their preferred cities or locations. This feature would offer a more personalized experience, as users could quickly access weather data for their favorite locations without having to search each time. Furthermore, integration with cloud storage solutions could allow users to sync their preferences across multiple devices, enhancing accessibility and user convenience.

3. **Extended Forecast and Historical Data:**

    • Expanding the app to include extended weather forecasts for up to a week or more would provide users with a comprehensive view of upcoming weather patterns. Additionally, incorporating historical weather data would allow users to compare past conditions, which could be valuable for planning or analysis purposes.

Conclusion

The Weather App has been designed to provide a comprehensive and user-friendly solution for accessing real-time weather information. By integrating features such as location-based weather updates and manual city searches, the app allows users to quickly obtain accurate weather data for their daily needs. The integration of the OpenWeatherMap API ensures that users receive reliable and up-to-date forecasts, including details like temperature, humidity, wind speed, and cloud cover. This approach not only simplifies the user experience but also streamlines the process of obtaining weather information, which otherwise would require navigating through multiple sources.

Looking ahead, there are several opportunities to further improve the app's functionality. The addition of additional features, such as advanced weather alerts, extended forecast views, or integration with other weather data sources, could expand its usability. Moreover, incorporating personalized user settings, such as location preferences or saved cities, would enhance convenience. A potential future improvement could be making the app mobile-friendly or creating a dedicated mobile app for on-the-go use, ensuring that users can check the weather easily from any device.

In summary, the Weather App offers a straightforward yet powerful tool for obtaining real-time weather information. With its focus on simplicity and reliability, the app meets the needs of a broad range of users, from casual individuals planning their day to professionals who require accurate forecasts for their work. As the app continues to evolve and incorporate user feedback, it has the potential to become an indispensable part of users' daily routines, providing them with the weather data they need quickly and efficiently.