**WEATHER IO: WEATHER APP**

Offered by **Smart Internz**



**TEAM ID:** LTVIP2023TMID08235

Kakinada Rishitha Priya (21NM5A0408)-Team Leader

Gorle Chandini (21NM5A0406)

Ommi Sandhya (20NM1A04C2)

Nandavarapu Shyam Susmitha Rani (20NM1A04B6)

**INTRODUCTION:**

The weather app is a digital application designed to provide users with real-time weather information for specific locations. Using HTML, CSS, and JavaScript, the app offers a simple and user-friendly interface that allows users to enter the name of a city and retrieve relevant weather data. The app utilizes an external weather API, such as OpenWeatherMap or Weather API, to fetch up-to-date weather details.

The user interface of the weather app consists of a container that houses an input field for entering the city name and a "Search" button to initiate the weather data retrieval process. Upon clicking the button, the JavaScript code sends a request to the weather API with the specified city name and API key.

The API returns a JSON response containing various weather-related parameters, including temperature, humidity, wind speed, and weather description. The JavaScript code processes the received data and updates the weather information section on the user interface.

The weather information displayed on the app includes the name of the city, the current temperature in Celsius, and a description of the weather conditions (e.g., cloudy, sunny, rainy). The app uses CSS to style the elements and create a visually appealing and responsive layout.

By providing users with instant access to accurate weather data, the weather app proves to be a useful tool for planning daily activities, traveling, or staying informed about weather conditions in different locations. The simplicity of the app's design and the convenience it offers make it an efficient and practical solution for users seeking quick weather updates.

**PROJECT STRUCTURE:**

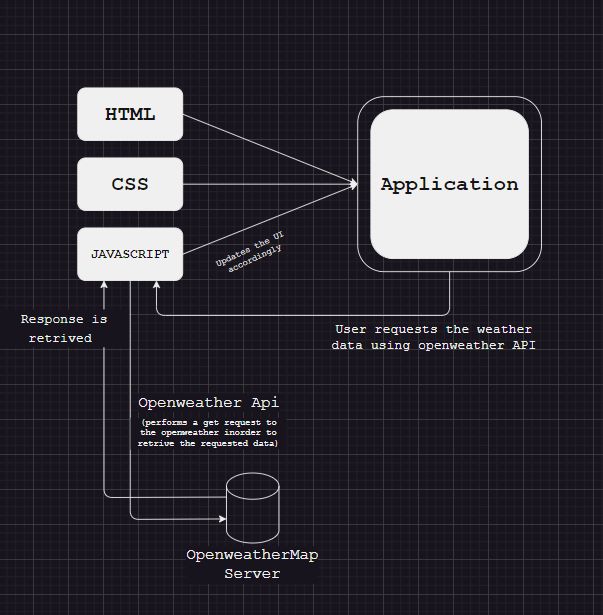


Fig 1. Design flow of weather application

The structure of a weather app involves several components that work together to provide users with real-time weather information. Below is an outline of the key components and their functionalities:

1. User Interface (UI):

* **Input Field**: Allows users to enter the name of the city they want to check the weather for.
* **Search Button**: Triggers the weather data retrieval process when clicked.
* **Weather Information Section**: Displays the retrieved weather data, including temperature, description, humidity, etc.

1. HTML File (index.html):

* Contains the basic structure of the app's user interface.
* Includes references to the CSS and JavaScript files.

The below Fig 2. Shows the html code used to create the weather app

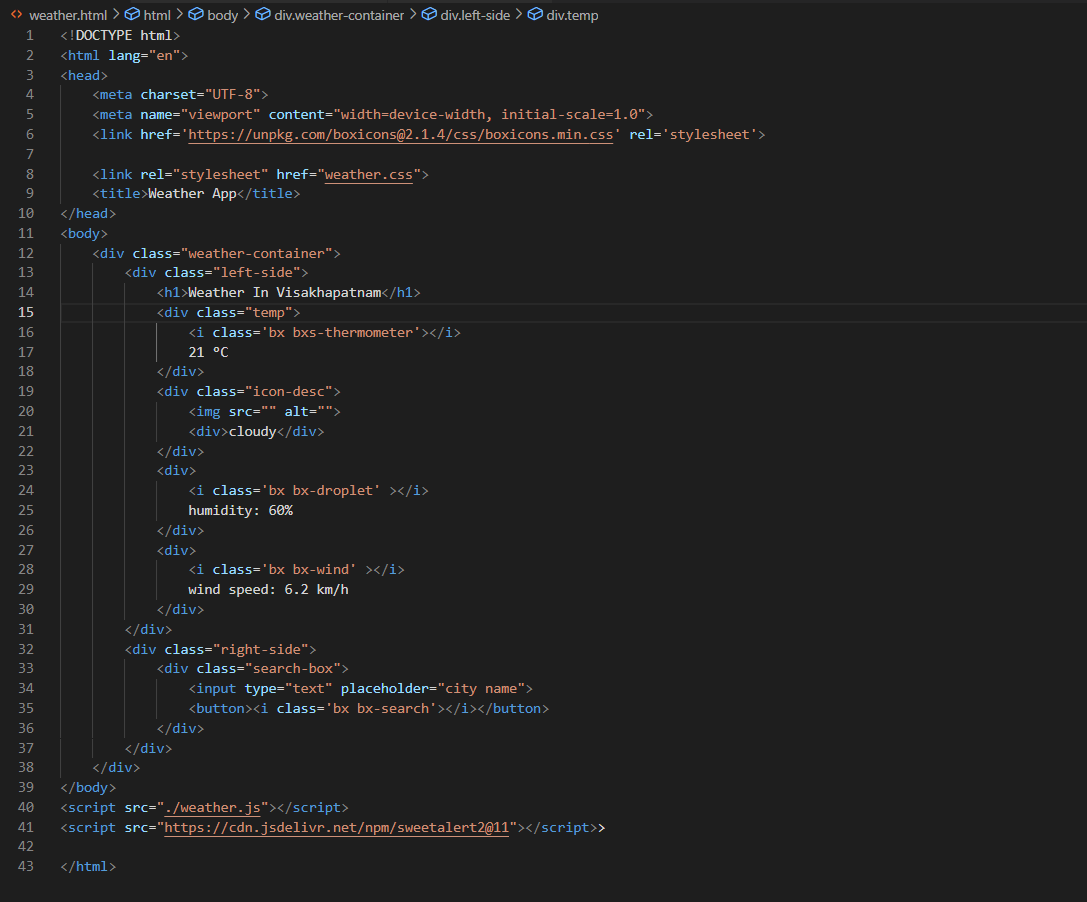


Fig 2. HTML Code

3. CSS File (styles.css):

* Defines the styling and layout of the user interface.
* Formats fonts, colors, backgrounds, buttons, and input fields.

The below fig 3 shows the CSS code i.e., is used to create the weather app

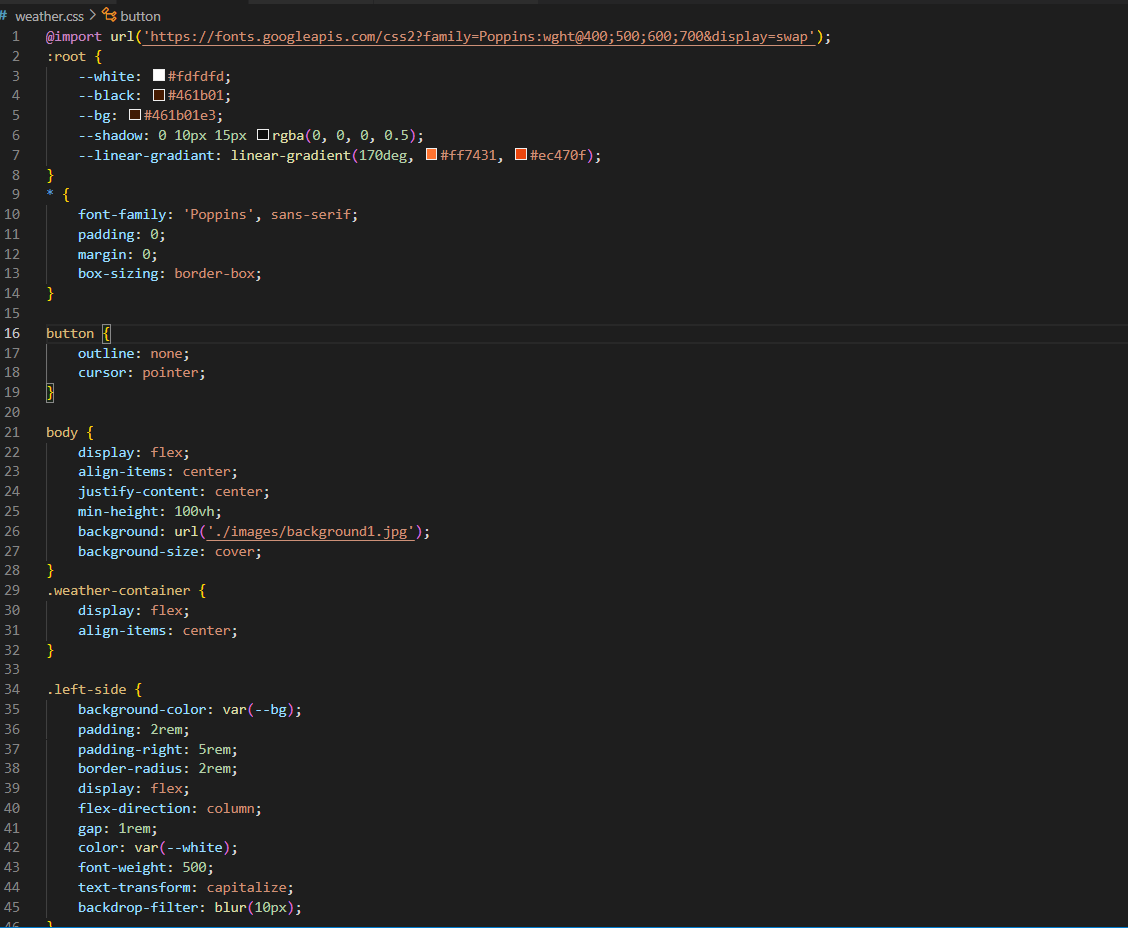


Fig 3. CSS code

4. JavaScript File (script.js):

* Implements the functionality of the app, handling user interactions and data retrieval.
* Contains event listeners to capture user actions, such as clicking the search button.
* Uses Fetch API or XML HTTP Request to fetch weather data from the weather API.
* Processes the retrieved data and updates the user interface with the relevant weather information.

The Fig 4. shows the java script code used to create the weather app

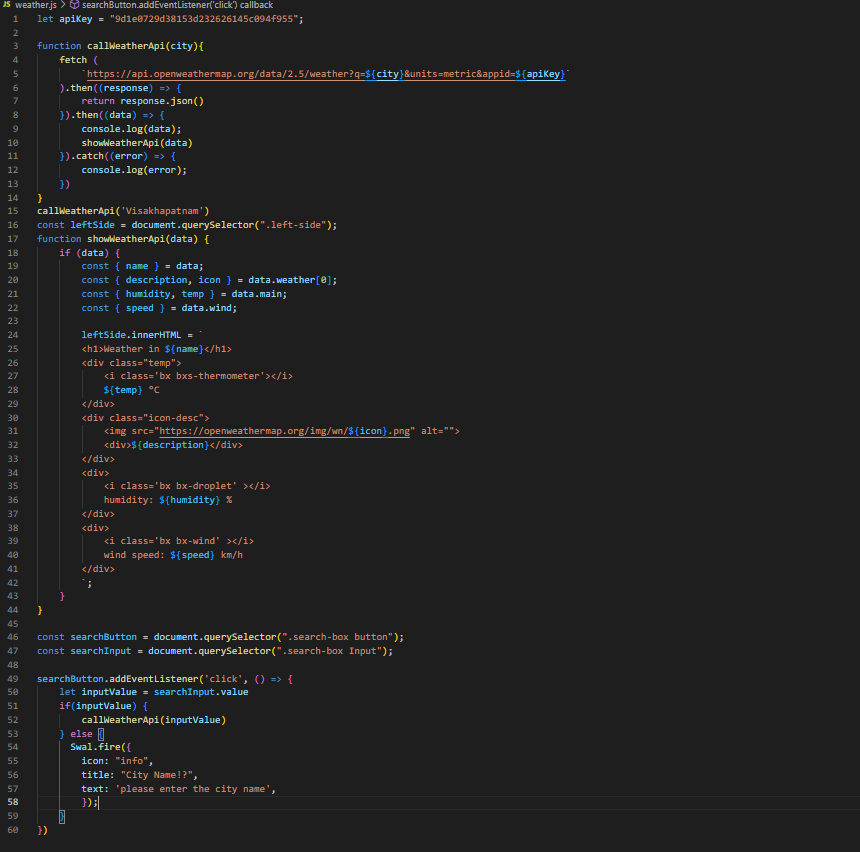


Fig 4. JS Code

5. Weather API:

* External API provided by weather service providers like OpenWeatherMap or Weather API.
* Requires an API key for authentication and usage limits.
* Accepts requests with a city name as a parameter and returns weather data in JSON format.

6. API Key:

* A unique identifier used to access the weather API.
* Must be obtained from the weather service provider by signing up and creating an account.

**The interaction flow of the weather app can be summarized as follows:**

1. User enters the name of a city into the input field.

2. User clicks the search button to initiate the weather data retrieval process.

3. JavaScript code sends a request to the weather API with the provided city name and API key.

4. The API responds with weather data in JSON format.

5. JavaScript processes the JSON response and extracts relevant weather information.

6. The app updates the weather information section on the user interface with the fetched data.

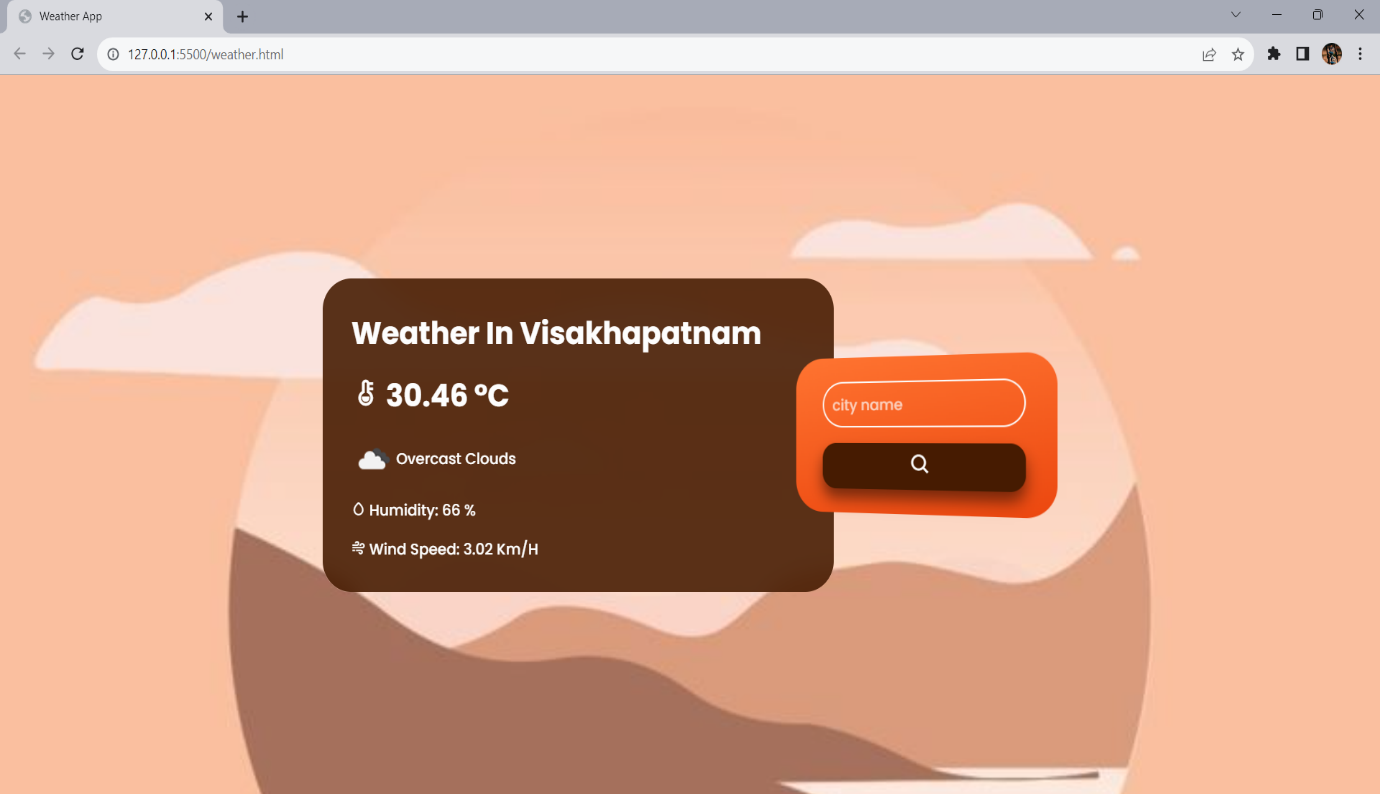


Fig 5. The output of the weather app is a simple webpage

**GITHUBLINK:** <https://rishithapriya.github.io/weather_forecastapp/Weather%20Forecast%20app/>

<https://github.com/RishithaPriya/weather_forecastapp>