WEATHER.IO : Weather App

Team ID : LTVIP2023TMID08182

Team Size : 4

Team Leader : Cheepurupalli Sravani

Team Member : Bandaru Gayathri

Team Member : Reyya Renuka

Team Member : Pulamarasetti Leelavathi

ABSTRACT: This Project is based on the idea of making an application which makes it easier to find the weather of any location. The Application will be created by using HTML, CSS, Java Script, so that it remains platform independent.

KEYWORDS: Weather Application, Weather Report, Weather Forecast, Location based.

INTRODUCTION:

Weather forecasting has been one amongst the foremost scientifically and technologically difficult issues round the world within the last century. Weather Forecast systems are among the foremost advanced equation systems that computer needs to solve. This most advances scientifically and technological modification is attributable to two factors: 1st, it's used for several human activities and second, attributable to the resourcefulness created by the varied technological advances that are directly associated with this concrete analysis field, just like the evolution of computation and therefore the improvement in activity systems. Sometimes beauty cannot be defined in just words and statements. Those lovely flowers and blossoms in the spring, warm sunshiny days in the summer, freezing mornings and the beautiful snow in the winter. Certain changes in weather are always wonderful and Cherishable. In some or other way our daily lives are dependant on the weather conditions. It has been always essential to know the regular updates of weather, as it continuously varies with every passing day. No one can imagine how the weather is going to be on a subsequent day. Definitely, the Weather forecast is a big thing that enabled many of us to stay notified about the changes in climatic conditions beforehand. It can be said that it is one of the greatest advancements of all time, mothered by innovative technologies and creative thoughts. The furtherance of the weather forecasting is the weather app development.

PROJECT OBJECTIVE:

By the end of this project, we will:

- Create a user interface using HTML, CSS and JavaScript to display weather information.
- Utilize JavaScript to interact with the OpenWeatherMap API and fetch weather data.
- Dynamically update the UI with the fetched weather data.
- Allow users to search for weather information by city name or zip code

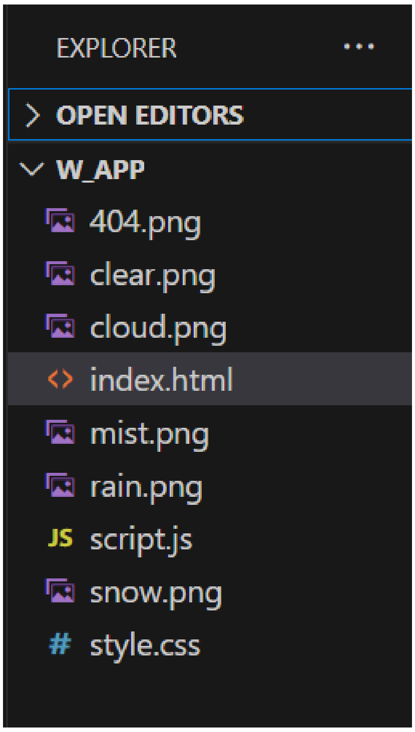
Project Flow:

To accomplish the objectives, we will complete the following activities:

- Set up the project structure
- Design and implement the user interface
- Connect to the OpenweatherMap API
- Fetch weather data based on user input
- Update the UI with the fetched weather data

Project Structure:

The project structure will include the following files:



```
EXPLORER ...
> OPEN EDITORS
v W_APP
  404.png
  clear.png
  cloud.png
  index.html
  mist.png
  rain.png
  script.js
  snow.png
  style.css
```

index.html: The main HTML file that contains the structure of the web page.

style.css: The CSS file that defines the styles for the user interface.

script.js: The JavaScript file that handles API calls and updates the UI.

images/ (optional): A folder to store any necessary images for the UI

Milestone 1: Set up the project structure

Create a new project folder for the Weather App.

Inside the project folder, create the following files/folders:

index.html

style.css

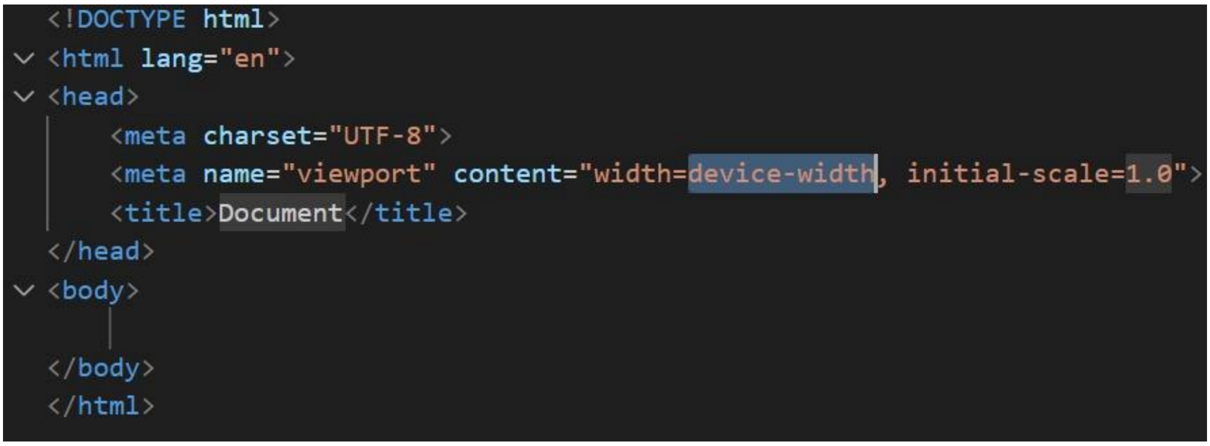
script.js

images

Milestone 2: Design and implement the user interface

Open index.html in your code editor.

Set up the basic HTML structure.



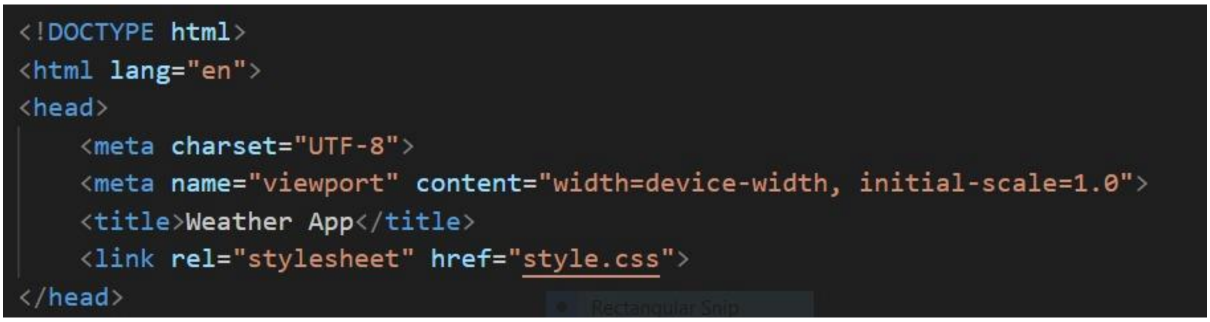
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>

</body>
</html>
```

Design the layout and structure of the user interface using HTML elements and CSS classes.

Label the title as "Weather App"

Apply styles to the UI elements using CSS in style.css.

Link style.css to 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>
```

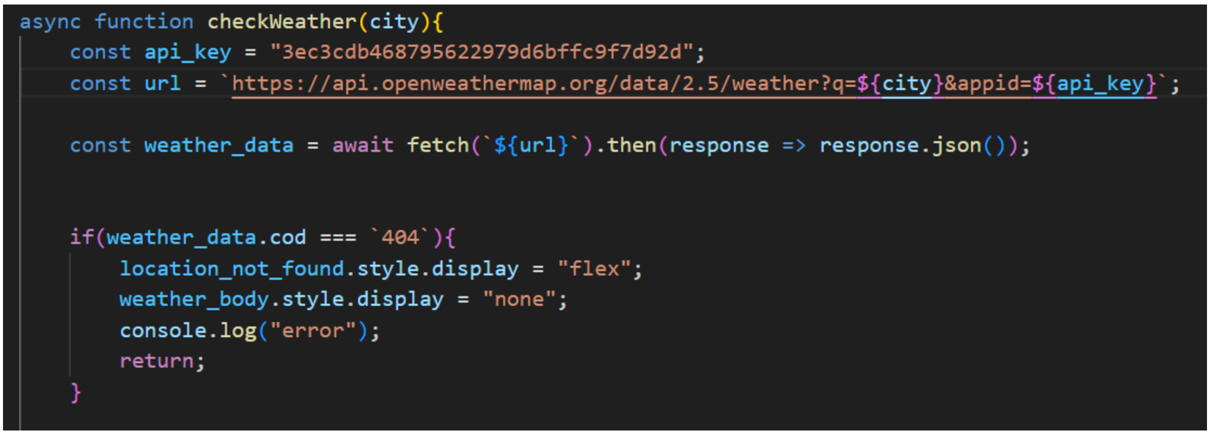
Milestone 3: Connect to the OpenWeatherMap API

In script.js, define a constant variable to store your OpenWeatherMap API key.

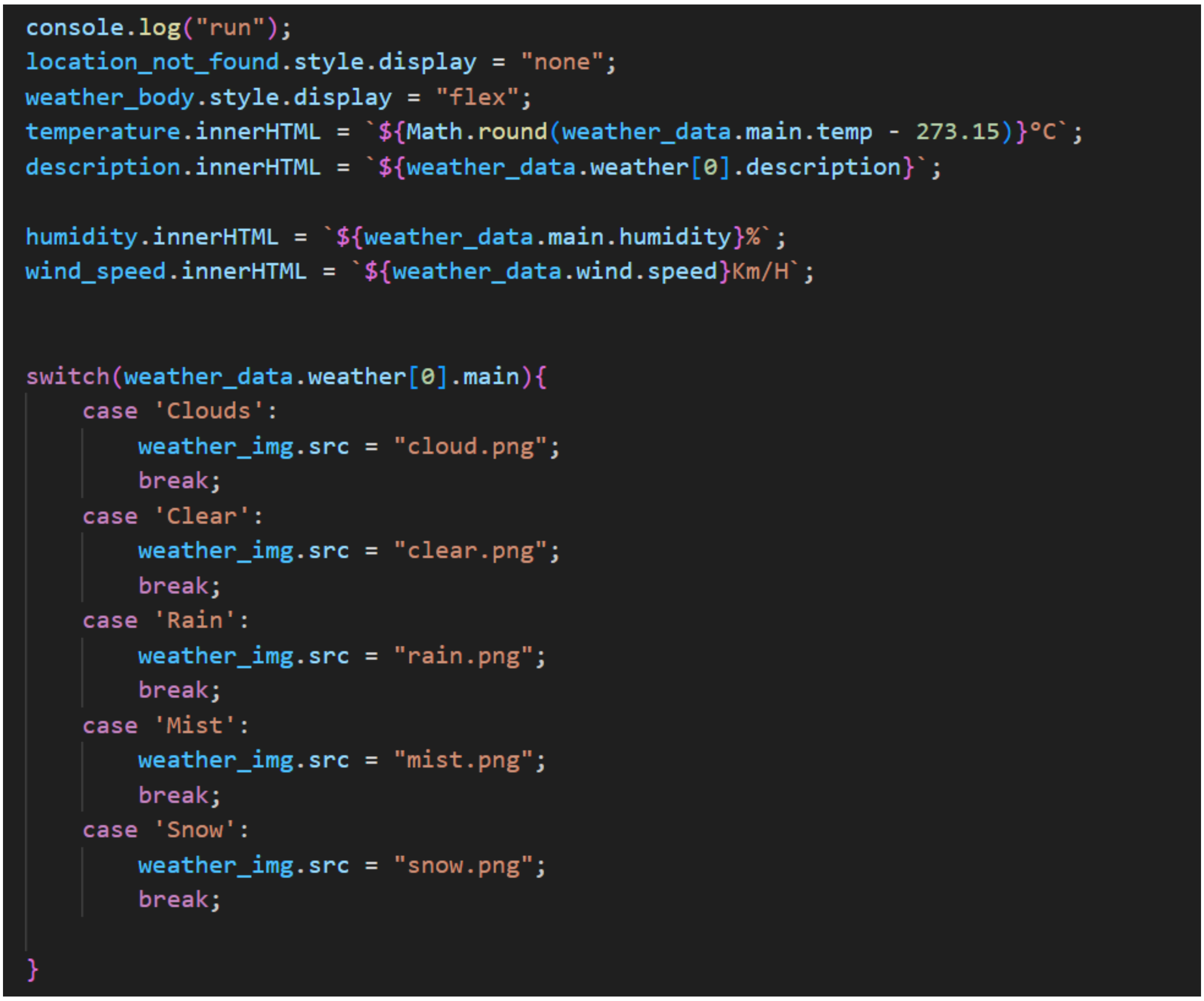
Create a function to handle API calls and fetch weather data from the OpenWeatherMap API.

Use the fetch() function or an AJAX library to make a GET request to the OpenWeatherMap API, passing the necessary parameters (e.g. city name).

Handle the API response and extract the relevant weather data.



```
async function checkWeather(city){
    const api_key = "3ec3cdb468795622979d6bffc9f7d92d";
    const url = `https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${api_key}`;

    const weather_data = await fetch(`${url}`).then(response => response.json());


    if(weather_data.cod === `404`){
        location_not_found.style.display = "flex";
        weather_body.style.display = "none";
        console.log("error");
        return;
    }
```



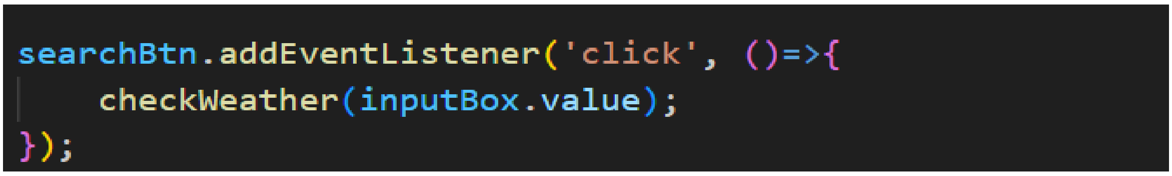
```
console.log("run");
location_not_found.style.display = "none";
weather_body.style.display = "flex";
temperature.innerHTML = `${Math.round(weather_data.main.temp - 273.15)}°C`;
description.innerHTML = `${weather_data.weather[0].description}`;

humidity.innerHTML = `${weather_data.main.humidity}%`;
wind_speed.innerHTML = `${weather_data.wind.speed}Km/H`;


switch(weather_data.weather[0].main){
    case 'Clouds':
        weather_img.src = "cloud.png";
        break;
    case 'Clear':
        weather_img.src = "clear.png";
        break;
    case 'Rain':
        weather_img.src = "rain.png";
        break;
    case 'Mist':
        weather_img.src = "mist.png";
        break;
    case 'Snow':
        weather_img.src = "snow.png";
        break;

}
```

Milestone 4: Fetch weather data based on user input

Add an input field and a button to the UI to allow users to enter a city name or zip code.

Add an event listener to the button to trigger the weather data fetch function when clicked.

Retrieve the user input from the input field.

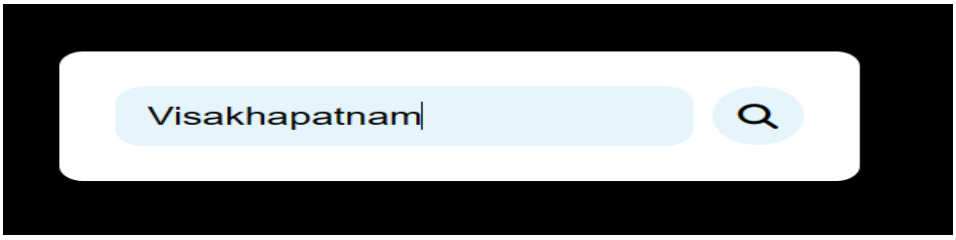


```
searchBtn.addEventListener('click', ()=>{
    checkWeather(inputBox.value);
});
```

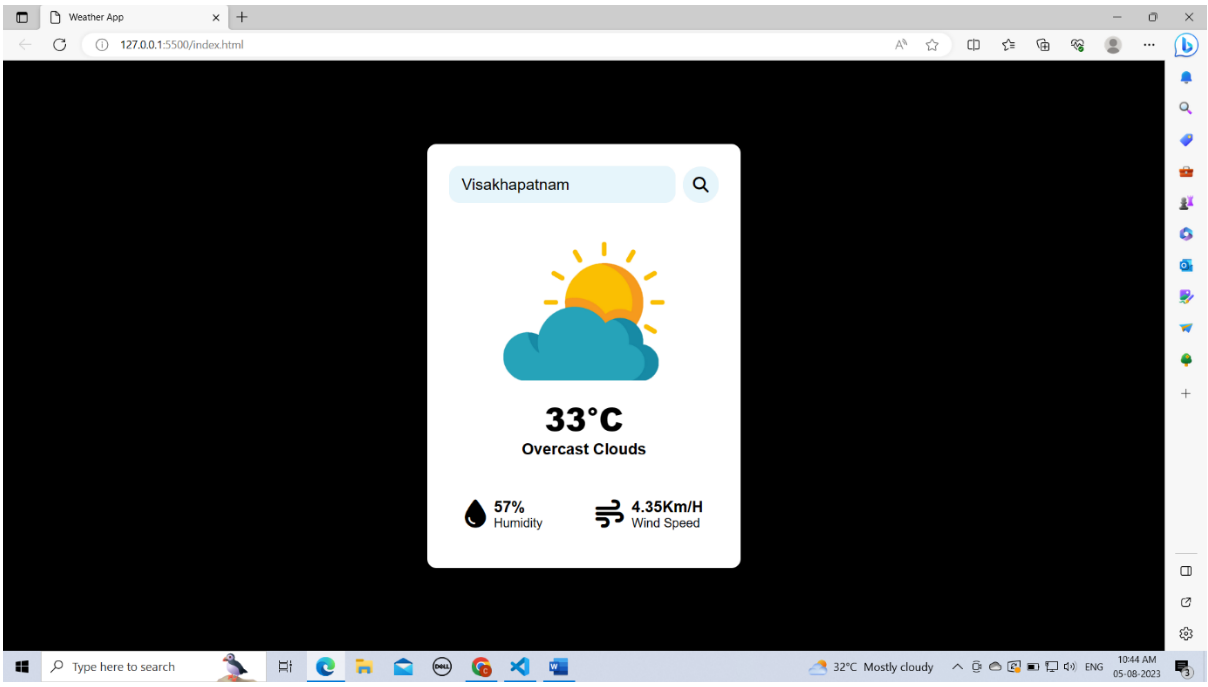
Milestone 5: Update the UI with the fetched weather data

Create functions to update the UI with the fetched weather data.

Select the necessary UI elements and modify the UI elements content or styles to display the weather information dynamically.



Visakhapatnam



Visakhapatnam

33°C

Overcast Clouds

57%
Humidity

4.35Km/H
Wind Speed

Conclusion:

The Weather App is a web application that provides real-time weather information to users. By integrating the OpenWeatherMap API and implementing an intuitive user interface, users can easily retrieve weather data for a specific location. The project's modular structure allows for easy maintenance and further enhancements, such as adding additional features or optimizing the UI.