

## **Major technologies used:**

### **The web app uses Geolocation API to get location of end-user:**

When a user opens the app, user is asked for permission to report location information. The Geolocation API provides us with latitude, longitude and accuracy of measurement. This task is asynchronous, so the app cannot initially start without location of user.

### **This weather app uses openweathermap API to fetch data and displays it to user:**

Open Weather is a weather API that feeds data to our application. We are using the one call API which when called provides us with minute forecast for 1-hour, hourly forecast for 48 hours, daily forecast for 7 days, historical data for 5 previous days and national weather alerts. The data is provided to us in JSON format.

### **Implementing search:**

When user enters city name in search bar and presses search button, the data is sent to geocoding API provided by open weather API to fetch latitude and longitude of city. Another fetch request is sent to API to fetch weather data using the fetched location data. The fetched data replaces the current data being shows.

### **Displaying weather icons:**

For displaying respective weather icons for each weather daily, hourly or current weather, the API provides us with a weather code. Several weather icons with respective name for each weather condition are saved in a static folder. After receiving weather code, respective source image is set using JavaScript.

### **Dynamically changing background image:**

To change background image according to current weather condition of that place, when current weather data is fetched, the weather icon code is used for our purpose. Several background images for specific weather conditions are saved in a static folder. The images are names with respective weather icon code for each weather code. The background image is then set via JS using respective weather icon code for current weather.

## **General working procedure:**

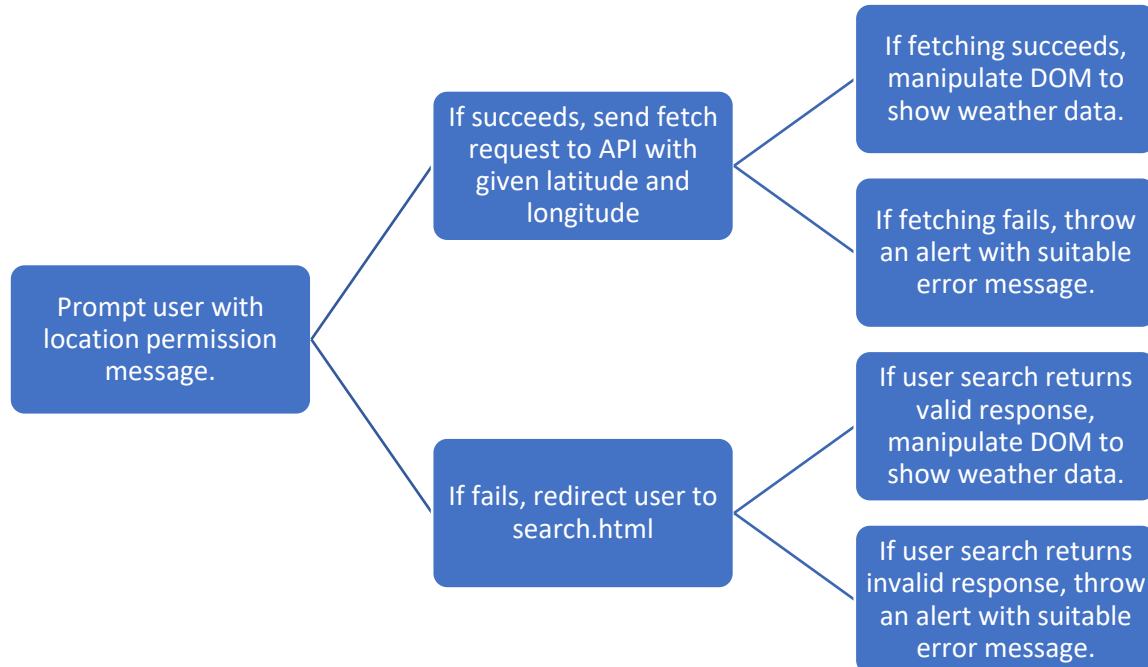
After the location data is retrieved from user's browser, the location is embedded to Open Weather API link and fetch request is sent to Open Weather API Servers. This is also an asynchronous task, so promises are used to store data after data is received. The received JSON data is then parsed and stored in a variable. The data now needs to be injected to DOM elements. The required window elements are imported from window object for DOM manipulation. The suitable data is injected using functions. To implement search, when user searches for a specific data, weather data for that city is fetched and replaces the currently shown data.

## **Algorithm:**

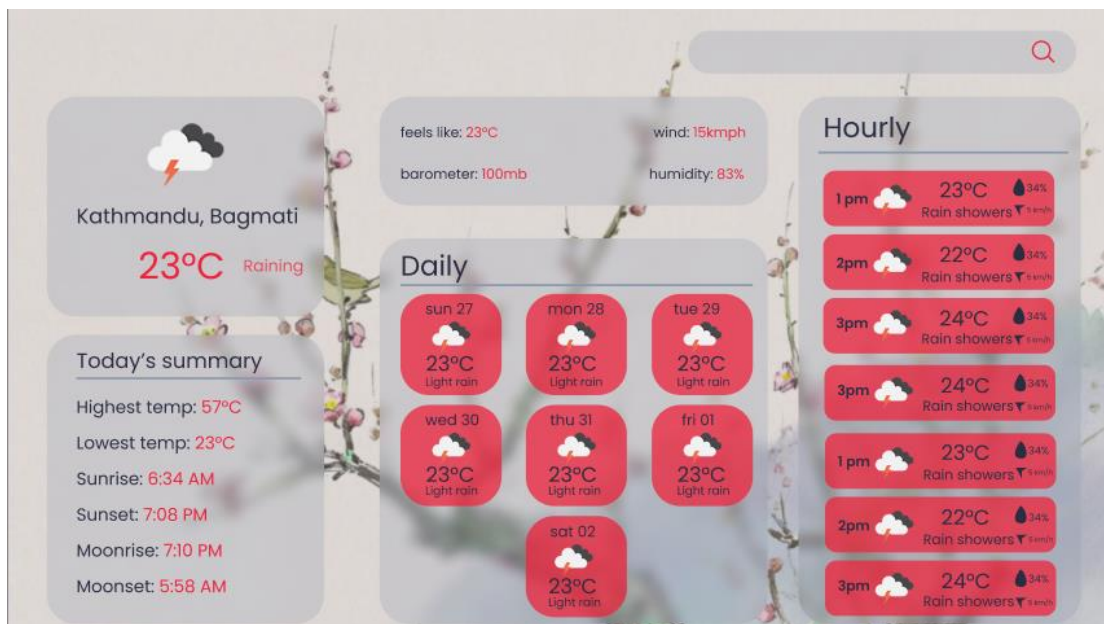
1. Fetch user location data from geolocation API.
  - I. Fetch weather data from Open Weather API.
    - i. Manipulate DOM to display the suitable data.

- a. To implement search, when user searches for specific place, send fetch request to weather API again with updated location data of that place.
  - b. If fetching fails, throw an alert with suitable error message.
- II. If fetching data from API fails, throw an alert with suitable error message.
2. If fetching location data fails, redirect user to search.html.

### Flow chart:



### UI:



## Folder Structure:

Weather-app:.

```
| index.html
| search.html
|
|——scripts
|   | favoritesHandler.js
|   | fetch.js
|   | helpers.js
|   | search.js
|   |
|   |——components
|       renderCurrentWeather.js
|       renderDailyWeather.js
|       renderHourlyWeather.js
|       renderTodaySummary.js
|
|——styles
    bootstrap.min.css
    style.css
```