Learner Assignment Submission Format

Learner Details

- **Name: Nikil G S**
- **Enrollment Number: su625mr014**
- **Batch / Class: MERN STACK**
- **Assignment: Weather API**
- **Date of Submission:30-7-2025**

Problem Solving Activity 1.1

1. Program Statement

Build a weather checking application where the user can enter a city name, and the app will display: place name, Temperature in Celsius, Weather description

---

2. Algorithm

- Wait for the user to input a city name and click the "Check" button.
- Fetch weather data for that city from the **OpenWeatherMap API**.
- Extract:
- Display the extracted information on the webpage.
- Based on the weather:
  - Add CSS class for cloudy, rainy, clear, or default background.
- If the city is not found or an error occurs, show an alert.

---

3. Pseudocode

START

User inputs city name in the textbox

User clicks "Check" button

↓

```
CALL getWeather function
    SET city = value entered in the textbox
    TRY
        SEND request to OpenWeatherMap API with city
        RECEIVE JSON weather data
        EXTRACT:
            - city name
            - temperature
            - weather description
        DISPLAY:
            - name in element with id 'name'
            - temperature in 'temp'
            - description in 'desc'
        IF weather contains 'Cloud'
            ADD 'cloudy' class to body
        ELSE IF weather contains 'Rain'
            ADD 'rainy' class to body
        ELSE IF temperature > 30
            ADD 'clear' class to body
        ELSE
            ADD 'default' class to body
    CATCH any error
        SHOW alert "city is not found"
END
```

1. **Program Code**

HTML

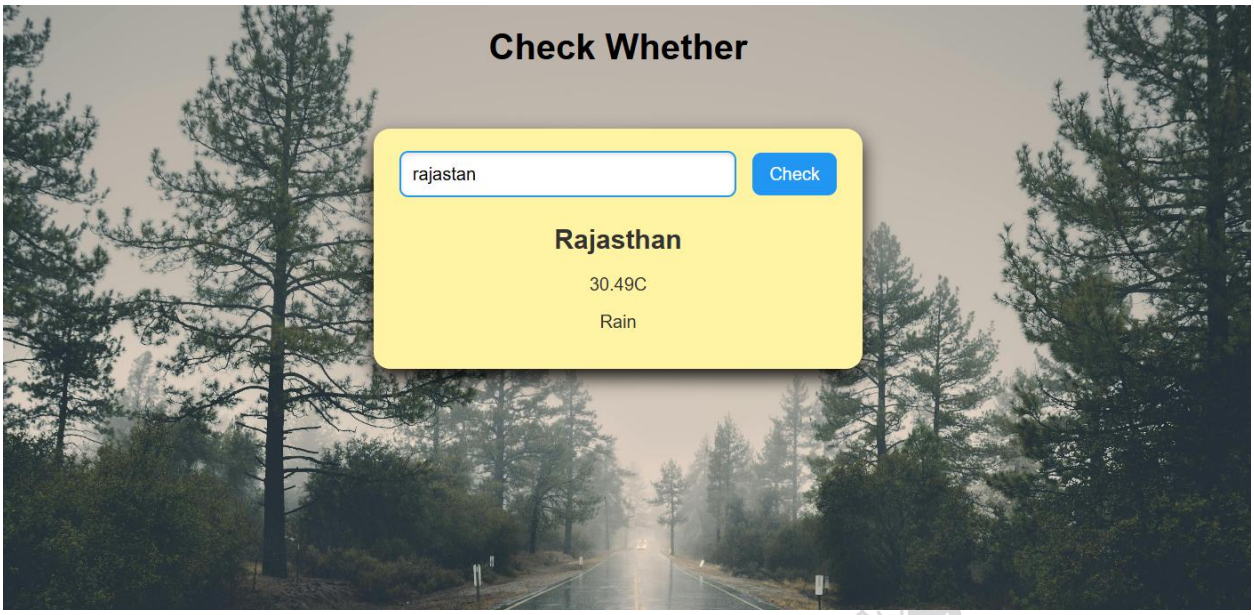
```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>whether</title>
   <link rel="stylesheet" href="style.css">
</head>
<body>
   <center><h1>Check Whether </h1></center>
    <div class="a">
      <div class="input">
         <input type="text" id="inputValue" placeholder="Enter city">
         <button id="button" class="button">Check</button>
      </div>
      <div class="b">
        <div class="ab"><h2 id="name"></h2></div>
        <div class="bc"><p id="temp"></p></div>
         <div class="bc"><p id="desc"></p></div>
      </div>
    </div>

   <script src="api.js"></script>
</body>
</html>
```

API.js

```
const getWeather = async () => {
   const city = document.getElementById('inputValue').value;
   const nameval = document.getElementById('name');
   const temp = document.getElementById('temp');
   const desc = document.getElementById('desc');
   const body = document.body;
   try {
```

```
        const response = await
fetch(`https://api.openweathermap.org/data/2.5/weather?q=${city}&units=metric&APPI
D=51b2167f2d4528282cde0f0c4bce52c7`)
        const data = await response.json()
        const weather = data.weather[0].main;

        nameval.innerText = data.name;
        temp.innerText = data.main.temp + "C";
        const weatherCondition = data.weather[0].main;
        desc.innerText = data.weather[0].main;

          if (weatherCondition.includes('Cloud')) {
            body.classList.add('cloudy');
        } else if (weatherCondition.includes('Rain')){
            body.classList.add('rainy');
        } else if ((data.main.temp > 30)){
            body.classList.add('clear');
        } else {
            body.classList.add('default');
        }

    } catch (error) {
        alert("city is not found");
    };
}

document.getElementById('button').addEventListener('click', getWeather)
```

2. **Screenshots of Output**



---

7. Observation / Reflection

 Good Use of API: The code successfully demonstrates how to fetch data using fetch() and work with asynchronous JavaScript using async/await.

 Conditional UI Styling: Weather conditions are used effectively to apply different styles via CSS.

 Improvement Area:

- Clear old classes before adding new ones to avoid class stacking.
- Validate the input to avoid unnecessary API calls on empty input.