Weather App - Documentation

## Introduction

This Weather App is a simple yet stylish weather information retriever using the OpenWeatherMap API.
The app allows users to enter a city name and fetch real-time weather details, including temperature, humidity, and wind speed.

## Features

- Search for any city's weather.
- Displays temperature, humidity, and wind speed.
- Dynamic weather icons based on conditions.
- Gradient background for a modern UI.
- Responsive design for all screen sizes.

## File Structure

```
/Weather-App
   |-- index.html  (Main HTML file)
   |-- style.css   (CSS styles)
   |-- script.js   (Weather fetch logic)
   |-- img/        (Icons and images)
```

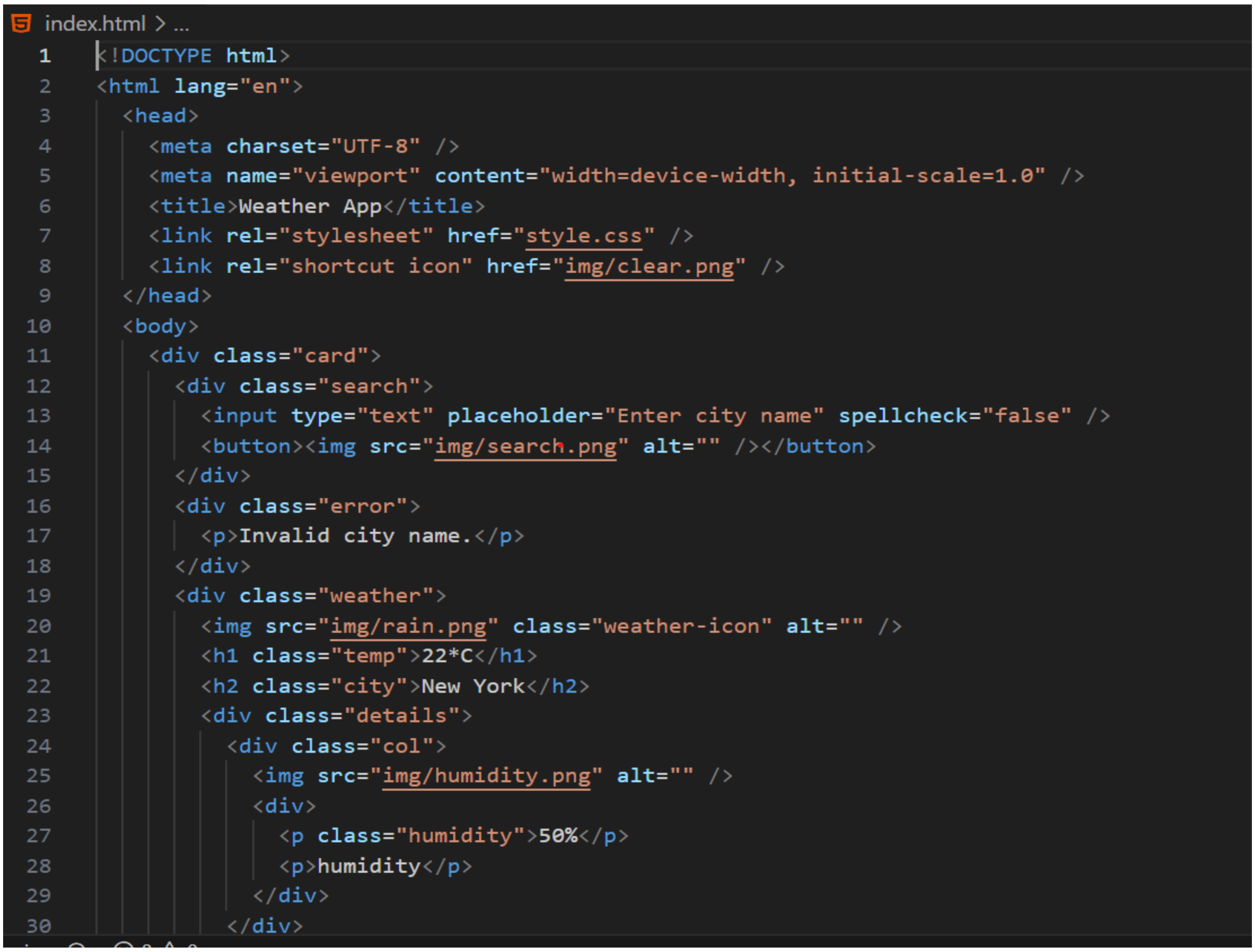
## Setup and Installation

- Clone or download the repository.
- Obtain an API key from OpenWeatherMap.
- Replace `const apiKey = "YOUR_API_KEY";` in script.js with your actual API key.
- Open index.html in a browser.

## Sections

### 1. HTML

The HTML file (index.html) defines the structure of the weather application. It includes input fields for city search,
display sections for weather details, and dynamic elements for icons and messages.

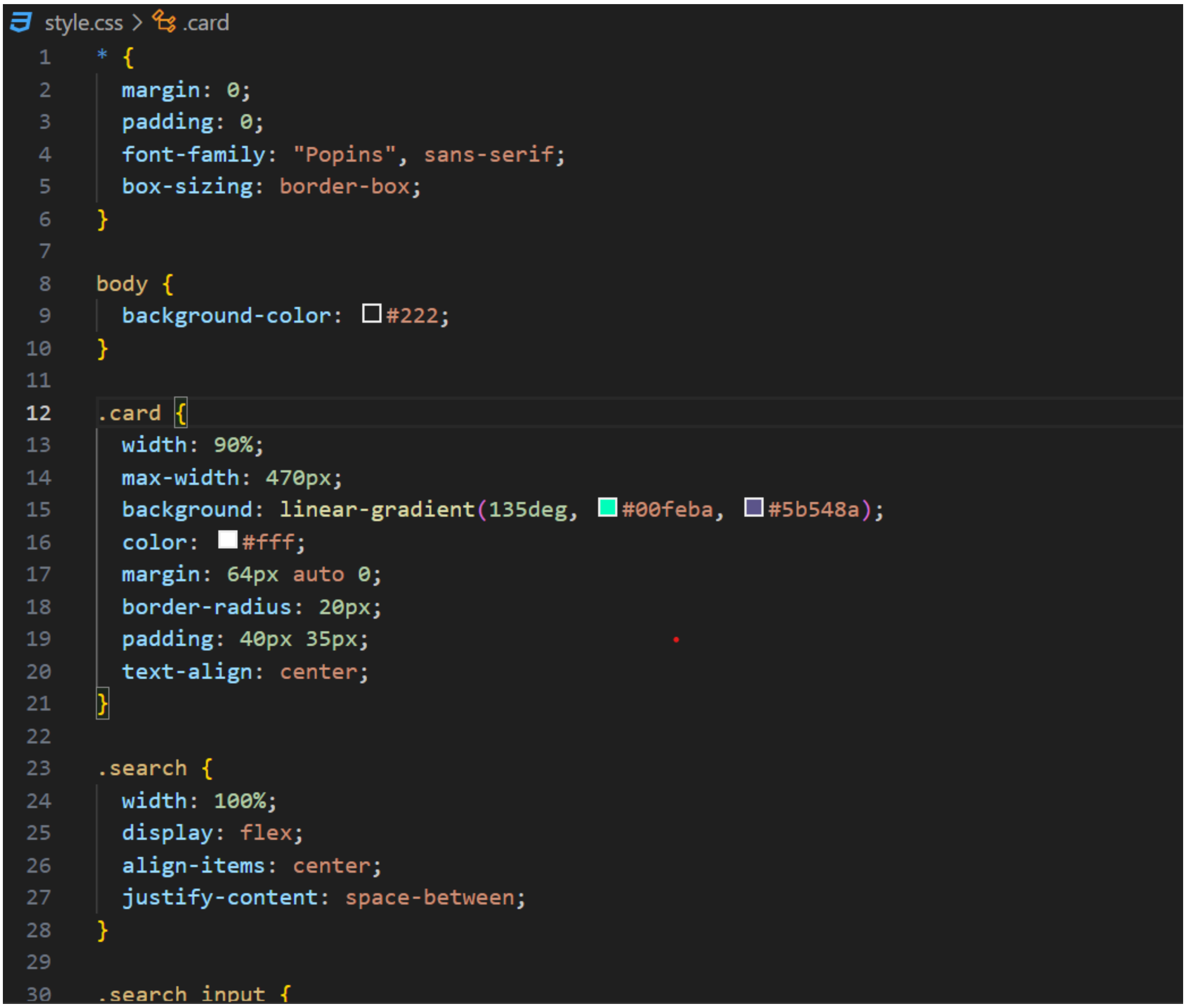


index.html > ...

```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" />
    <link rel="shortcut icon" href="img/clear.png" />
  </head>
  <body>
    <div class="card">
      <div class="search">
        <input type="text" placeholder="Enter city name" spellcheck="false" />
        <button><img src="img/search.png" alt="" /></button>
      </div>
      <div class="error">
        <p>Invalid city name.</p>
      </div>
      <div class="weather">
        <img src="img/rain.png" class="weather-icon" alt="" />
        <h1 class="temp">22*C</h1>
        <h2 class="city">New York</h2>
        <div class="details">
          <div class="col">
            <img src="img/humidity.png" alt="" />
            <div>
              <p class="humidity">50%</p>
              <p>humidity</p>
            </div>
          </div>
```

### 2. CSS

The CSS file (style.css) provides styling to the application, ensuring a modern and responsive design.
Tailwind CSS is used for layout and styling enhancements, while additional custom styles refine the user interface.

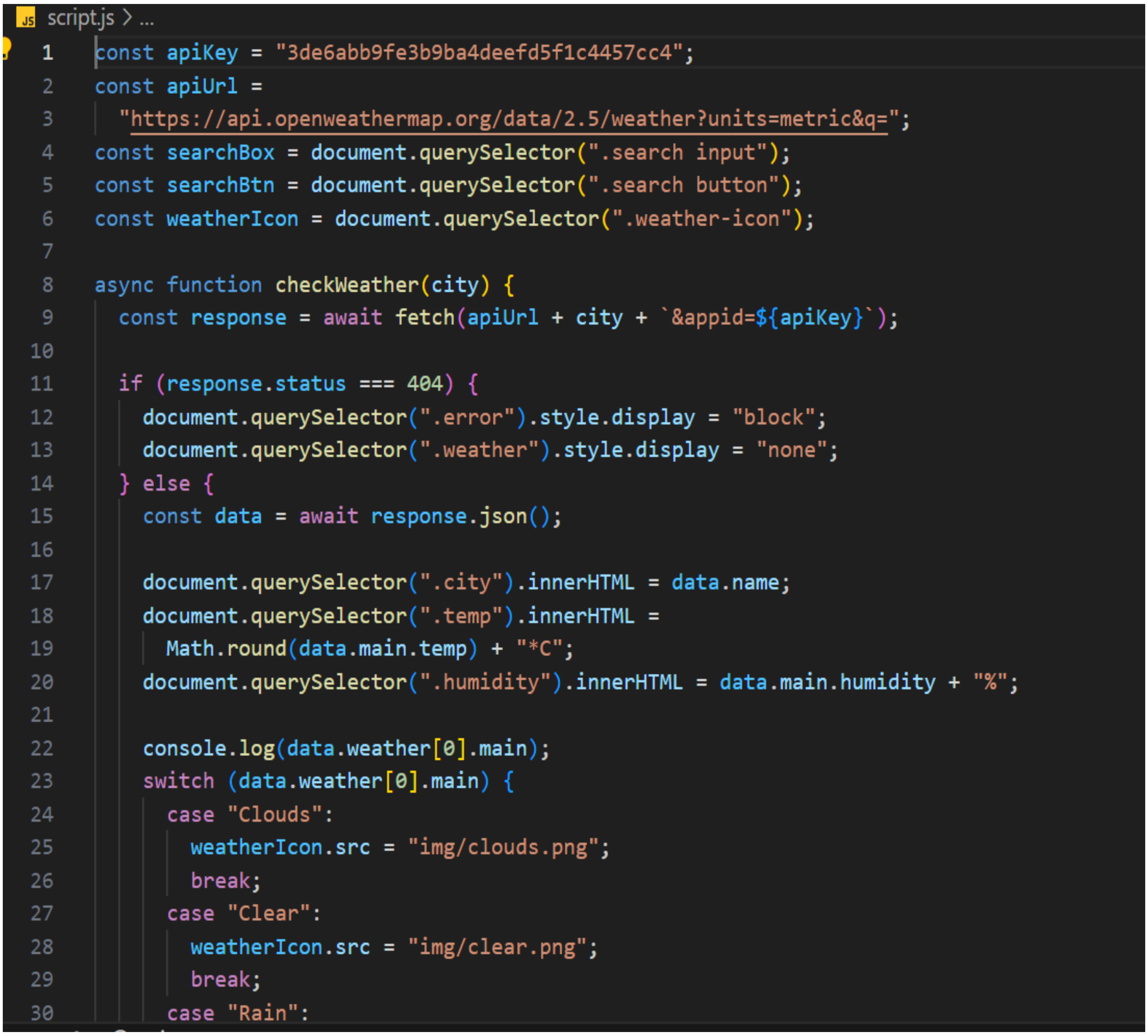

style.css > .card

```css
* {
  margin: 0;
  padding: 0;
  font-family: "Popins", sans-serif;
  box-sizing: border-box;
}

body {
  background-color: #222;
}

.card {
  width: 90%;
  max-width: 470px;
  background: linear-gradient(135deg, #00feba, #5b548a);
  color: #fff;
  margin: 64px auto 0;
  border-radius: 20px;
  padding: 40px 35px;
  text-align: center;
}

.search {
  width: 100%;
  display: flex;
  align-items: center;
  justify-content: space-between;
}

.search input {
```

### 3. JavaScript

The JavaScript file (script.js) handles API requests and dynamic updates to the DOM.

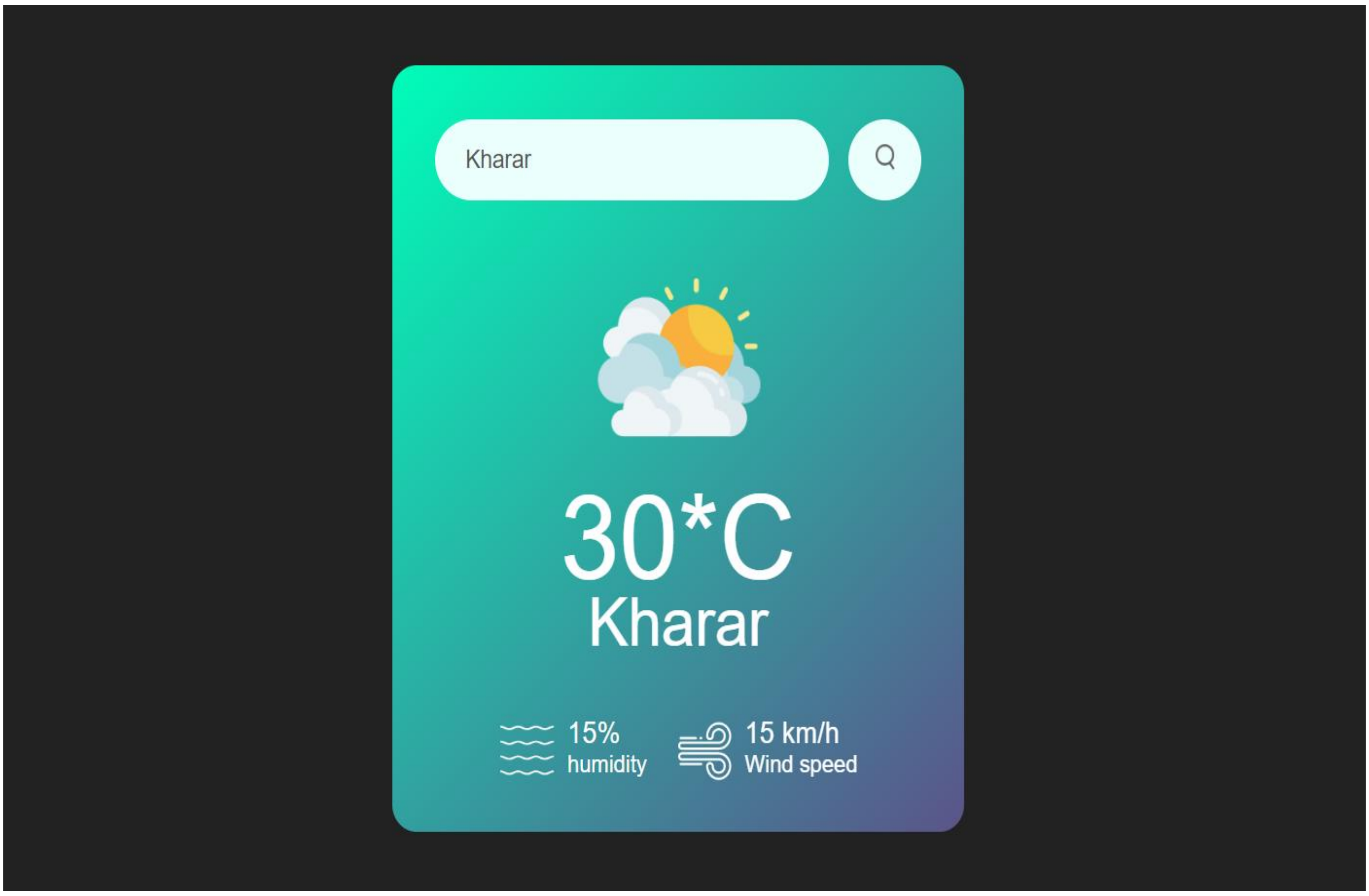
It fetches weather data from OpenWeatherMap, processes responses, and updates the UI accordingly.



script.js > ...

```javascript
const apiKey = "3de6abb9fe3b9ba4deefd5f1c4457cc4";
const apiUrl =
  "https://api.openweathermap.org/data/2.5/weather?units=metric&q=";
const searchBox = document.querySelector(".search input");
const searchBtn = document.querySelector(".search button");
const weatherIcon = document.querySelector(".weather-icon");

async function checkWeather(city) {
  const response = await fetch(apiUrl + city + `&appid=${apiKey}`);

  if (response.status === 404) {
    document.querySelector(".error").style.display = "block";
    document.querySelector(".weather").style.display = "none";
  } else {
    const data = await response.json();

    document.querySelector(".city").innerHTML = data.name;
    document.querySelector(".temp").innerHTML =
      Math.round(data.main.temp) + "*C";
    document.querySelector(".humidity").innerHTML = data.main.humidity + "%";

    console.log(data.weather[0].main);
    switch (data.weather[0].main) {
      case "Clouds":
        weatherIcon.src = "img/clouds.png";
        break;
      case "Clear":
        weatherIcon.src = "img/clear.png";
        break;
      case "Rain":
```

## Project Output :-



## Conclusion

This weather app provides real-time weather updates in a visually appealing and responsive UI.
You can further enhance it by integrating additional weather parameters or customizing UI elements to match your preferences.