Project Title	Weather App
Technologies	HTML, CSS, JavaScript
Project Difficulties level	Hard

Project Description

Objective: In this project, you will create a web-based weather application that allows users to enter a location and retrieve the current weather conditions for that location. The app should display details such as temperature, weather description, and an icon representing the current weather.

Skills to Practice:

- HTML for structuring the app.
- CSS for styling and layout.
- JavaScript for fetching weather data and interactivity.
- Integrating with a public weather API.

Project Requirements

Your weather app should include the following features:

User Input: Provide a text input field where users can enter the name of a city or location for which they want to check the weather.

Submit Button: Include a "Submit" button that, when clicked, triggers the weather data retrieval.

Display Weather Data: After a user submits a location, display the current weather conditions for that location. This should include temperature (in Celsius), weather description (e.g., "Sunny," "Cloudy," "Rainy"), and an icon representing the current weather condition.

Styling: Apply CSS to make your weather app visually appealing. Ensure it's responsive and user-friendly.

Use of Public Weather API: You are required to use a public weather API to fetch weather data for the entered location. One popular option is the OpenWeatherMap API (https://openweathermap.org/), which provides a free tier.

Error Handling: Implement error handling in case the user enters an invalid location or if there are issues with the API request. Provide appropriate error messages to the user.

Project Guidelines

- Start by creating the HTML structure for your app. Include the necessary input fields and placeholders for displaying weather data.
- Style your app using CSS to make it visually appealing and user-friendly.
- Use JavaScript to handle user interactions and to fetch weather data from the chosen weather API. Ensure that the data is displayed clearly and comprehensively to the user.
- Consider responsive design to make your app look good on various screen sizes.
- Test your app thoroughly to ensure it functions correctly. Pay special attention to error handling and providing informative messages to users.
- You may choose to use asynchronous JavaScript (Promises or async/await) for making API requests.
- Ensure that you are complying with the terms and conditions of the chosen weather API, especially if you plan to deploy the app publicly.

Submission

Submit your project as a zip file containing all HTML, CSS, and JavaScript files. Include a README file with instructions on how to run the app and any additional notes you'd like to provide.

Grading Criteria

Your project will be evaluated based on the following criteria:

- Functionality: Does the app correctly display weather data based on user input?
- User Interface: Is the app visually appealing, responsive, and user-friendly?
- Error Handling: Does the app handle errors gracefully and provide helpful messages to users?
- Code Quality: Is the code well-structured, organized, and commented?

Additional Tips

- Use the OpenWeatherMap API documentation to understand how to make API requests and retrieve weather data.
- Feel free to explore additional features like displaying forecasts, changing temperature units (e.g., Fahrenheit), or adding more weather details.

Good luck with your Weather App project!