Weather App using React or Vue.js
Upload a html file named USPnumber1.USPnumber2.zip. If you need to upload a file after the deadline use late.USPnumber1.USPnumber2.zip

Remove the node_modules folder before sending the zip.

Activity: Building a Simple Weather App using React or Vue.js

Objective: This activity will challenge you to create a simple weather application using either React or Vue.js. This app will make use of component lifecycle methods (or the Composition API in Vue.js), state management, and data flow between components.

Instructions:

Setup: Create a new React Hooks or Vue.js app using Create React App, Vue CLI, or your preferred setup.

Components: Create the following components:

App: This is the main component that will hold other components.
SearchBar: This component should have an input field for the user to enter a city name.
WeatherDisplay: This component will display the weather information returned by the API.
State Management and Data Flow: Implement the following requirements:

The App component should maintain the state of the current city searched by the user.
When the user types a city name into the SearchBar and presses Enter, this should trigger an update to the state in the App component.
The WeatherDisplay component should receive the city name as a prop and fetch weather data for that city.
Component Lifecycle / Composition API: Implement the following requirements:

If you're using React: Use useEffect() in the WeatherDisplay component to fetch data when the component first mounts or whenever the city name prop changes. In React with classes, useEffect is equivalent to componentDidMount() and componentDidUpdate().
If you're using Vue.js: Use the mounted lifecycle hook to fetch data when the component is first rendered, and a watch or the Composition API's watchEffect to fetch new data whenever the city name prop changes.
Weather API: Use the Fetch API to send GET requests to the Open-Meteo public APIs. To find the city's latitude and longitude, use the endpoint https://geocoding-api.open-meteo.com/v1/search?name={city}&count=1 (API Documentation) OR create a list of, at least, 5 cities with their latitudes and longitudes. To find the weather, use the endpoint:
https://api.open-meteo.com/v1/forecast?latitude={lat}&longitude={lon}&current_weather=true (API Documentation).

Error Handling: Implement error handling for the case when the user searches for a city that the API doesn't recognize.
Style your components using CSS to improve the user experience.

Finally, test your application to ensure it behaves as expected. Check that the data flow is correct and that the component lifecycle methods (or the Composition API in Vue.js) are working as they should.

This activity touches on several key concepts of both React and Vue.js including component structure, state management, handling user input, and lifecycle methods (or the Composition API). Take your time to understand each concept. Happy coding!