Weather App Project Documentation

Overview

This project is a simple weather app built using ReactJS, which allows users to view real-time weather statistics of a city, including temperature, humidity, and wind speed. The app utilizes the OpenWeatherMap API to fetch data from the server.

Features

- Displays real-time weather statistics of a city
- Fetches data from the OpenWeatherMap API
- Displays weather icon according to the weather data
- Supports temperature in degrees Celsius and Fahrenheit
- Displays wind speed in kilometers per hour and miles per hour
- Displays humidity in percentage

Getting Started

❖ Prerequisites

- NodeJS
- npm
- OpenWeatherMap API Key

❖ Installation

1. Create a new React project in a folder named "weather-app":

```bash
npx create-react-app weather-app
```

2. Navigate to the project directory:

```bash
cd weather-app
```

3. Clear the contents of the `public/index.html` file and replace the title with "weather":

```html
<title>weather</title>
```

4. Create a folder structure as follows:

```bash
src
├── components
│   ├── assets
│   └── weather-app
│       └── weather.jsx
└── App.js
```

5. Download Thunder Client Extension to view the response data coming from the API.

❖ API Key

1. Obtain an API key from OpenWeatherMap.
2. Insert the API key into the `api` variable in the `weather.jsx` file.

❖ Running the App

1. Start the development server by running:

```sql
npm start
```

2. Open the app in a browser at `http://localhost:3000/`.

❖ Usage

1. Enter the name of a city in the search bar.
2. Click the search button to view the weather statistics of the city.
3. The temperature is displayed in degree Celsius by default. To switch to Fahrenheit, click the "F" button.
4. The wind speed is displayed in kilometers per hour by default. To switch to miles per hour, click the "mph" button.

Components

❖ Weather Component

The `weather.jsx` file contains the `Weather` component, which is responsible for rendering the user interface and fetching data from the OpenWeatherMap API.

Props

- `API`: The OpenWeatherMap API key.
- `tempUnit`: The temperature unit (`celsius` or `fahrenheit`).
- `windUnit`: The wind speed unit (`kmh` or `mph`).

Methods

- `searchWeather`: Fetches data from the OpenWeatherMap API using the `fetch()` method and updates the component state with the received data.

State

- `weatherData`: The weather data received from the OpenWeatherMap API.
- `error`: The error message received from the OpenWeatherMap API.
- `loading`: A boolean indicating whether the component is currently loading data from the API.

Files

- `K:\rt\src\Components\Assets/`: Contains the assets used in the app, including the weather icons.
- `K:\rt\src\Components\WeatherApp\WeatherApp.jsx`: Contains the weather component.
- `K:\rt\src\App.js`: Mounts the `Weather` component on the screen.

Source Code

- **WeatherApp.jsx**

```
//WeatherApp.jsx
// Type "rafc" and press Enter
import React, { useState } from 'react';
import './WeatherApp.css';
import search_icon from '../Assets/search.png';
import clear_icon from '../Assets/clear.png';
import cloud_icon from '../Assets/drizzle.png';
import rain_icon from '../Assets/rain.png';
import snow_icon from '../Assets/snow.png';
import wind_icon from '../Assets/wind.png';
import humidity_icon from '../Assets/humidity.png';
import drizzle_icon from '../Assets/drizzle.png';

const WeatherApp = () => {

  let api_key = "c358ffb5f6690d695999310474d666c7";

  const [wicon, setWicon] = useState(cloud_icon);

  const search = async () =>{
    const element =document.getElementsByClassName("cityInput")
    if(element[0].value=="")
    {
      return 0;
    }
    let url =
`https://api.openweathermap.org/data/2.5/weather?q=${element[0].value}&units=M
etric&appid=${api_key}`;

    let response = await fetch(url);
    let data = await response.json();
    const humidity = document.getElementsByClassName("humidity-percent");
    const wind = document.getElementsByClassName("wind-rate");
    const temprature = document.getElementsByClassName("weather-temp");
    const location = document.getElementsByClassName("weather-location");

    humidity[0].innerHTML = data.main.humidity+ "%";
    wind[0].innerHTML =Math.floor(data.wind.speed)+" km/h";
```

```
    temprature[0].innerHTML =Math.floor(data.main.temp)+"°C";
    location[0].innerHTML = data.name;

    if(data.weather[0].icon==="01d" ||data.weather[0].icon==="01n" )
    {
      setWicon(clear_icon);
    }
    else if(data.weather[0].icon==="02d" ||data.weather[0].icon==="02n")
    {
      setWicon(cloud_icon);
    }
    else if(data.weather[0].icon==="03d" ||data.weather[0].icon==="03n")
    {
      setWicon(drizzle_icon);
    }
    else if(data.weather[0].icon==="04d" ||data.weather[0].icon==="04n")
    {
      setWicon(drizzle_icon);
    }
    else if(data.weather[0].icon==="09d" ||data.weather[0].icon==="09n")
    {
      setWicon(rain_icon);
    }
    else if(data.weather[0].icon==="10d" ||data.weather[0].icon==="10n")
    {
      setWicon(rain_icon);
    }
    else if(data.weather[0].icon==="13d" ||data.weather[0].icon==="13n")
    {
      setWicon(snow_icon);
    }
    else{
      setWicon(clear_icon);
    }


  }
  return (
    <div className= "container">
      <div className="top-bar">
      <input type="text" className="cityInput" placeholder='Search' />
      <div className="search-icon" onClick={()=>{search()}}>
        <img src={search_icon} alt="" />
      </div>
      </div>
      <div className="weather-image">
       <img src={wicon} alt="" />
      </div>
      <div className="weather-temp">29°C</div>
```

```
      <div className="weather-location">Mumbai</div>
      <div className="data container">
        <div className="element">
          <img src={humidity_icon} alt="" className="icon" />
          <div className="data">
            <div className="humidity-percent">64%</div>
            <div className="text">Humidity</div>

          </div>
        </div>
        <div className="element">
          <img src={wind_icon} alt="" className="icon" />
          <div className="data">
            <div className="wind-rate">18 km/h</div>
            <div className="text">Wind Speed</div>

          </div>
        </div>
      </div>
    </div>
  );
};

export default WeatherApp;
```

- **App.js**

```
import './App.css';
import WeatherApp from './Components/WeatherApp/WeatherApp';;

function App() {
  return (
    <div className="App">
      <WeatherApp />
    </div>
  );
}

export default App;
```

- **WeatherApp.css**

```
// Type "rafc" and press Enter
import React, { useState } from 'react';
import './WeatherApp.css';
import search_icon from '../Assets/search.png';
import clear_icon from '../Assets/clear.png';
import cloud_icon from '../Assets/drizzle.png';
import rain_icon from '../Assets/rain.png';
import snow_icon from '../Assets/snow.png';
import wind_icon from '../Assets/wind.png';
import humidity_icon from '../Assets/humidity.png';
import drizzle_icon from '../Assets/drizzle.png';

const WeatherApp = () => {

  let api_key = "c358ffb5f6690d695999310474d666c7";

  const [wicon, setWicon] = useState(cloud_icon);

  const search = async () =>{
    const element =document.getElementsByClassName("cityInput")
    if(element[0].value=="")
    {
      return 0;
    }
    let url =
`https://api.openweathermap.org/data/2.5/weather?q=${element[0].value}&units=M
etric&appid=${api_key}`;

    let response = await fetch(url);
    let data = await response.json();
    const humidity = document.getElementsByClassName("humidity-percent");
    const wind = document.getElementsByClassName("wind-rate");
    const temprature = document.getElementsByClassName("weather-temp");
    const location = document.getElementsByClassName("weather-location");

    humidity[0].innerHTML = data.main.humidity+ "%";
    wind[0].innerHTML =Math.floor(data.wind.speed)+" km/h";
    temprature[0].innerHTML =Math.floor(data.main.temp)+"°C";
    location[0].innerHTML = data.name;

    if(data.weather[0].icon==="01d" ||data.weather[0].icon==="01n" )
    {
      setWicon(clear_icon);
    }
    else if(data.weather[0].icon==="02d" ||data.weather[0].icon==="02n")
```

```
    {
      setWicon(cloud_icon);
    }
    else if(data.weather[0].icon==="03d" ||data.weather[0].icon==="03n")
    {
      setWicon(drizzle_icon);
    }
    else if(data.weather[0].icon==="04d" ||data.weather[0].icon==="04n")
    {
      setWicon(drizzle_icon);
    }
    else if(data.weather[0].icon==="09d" ||data.weather[0].icon==="09n")
    {
      setWicon(rain_icon);
    }
    else if(data.weather[0].icon==="10d" ||data.weather[0].icon==="10n")
    {
      setWicon(rain_icon);
    }
    else if(data.weather[0].icon==="13d" ||data.weather[0].icon==="13n")
    {
      setWicon(snow_icon);
    }
    else{
      setWicon(clear_icon);
    }


  }
  return (
    <div className= "container">
      <div className="top-bar">
      <input type="text" className="cityInput" placeholder='Search' />
      <div className="search-icon" onClick={()=>{search()}}>
        <img src={search_icon} alt="" />
      </div>
      </div>
      <div className="weather-image">
       <img src={wicon} alt="" />
      </div>
      <div className="weather-temp">29°C</div>
      <div className="weather-location">Mumbai</div>
      <div className="data container">
        <div className="element">
          <img src={humidity_icon} alt="" className="icon" />
          <div className="data">
            <div className="humidity-percent">64%</div>
            <div className="text">Humidity</div>
```

```
          </div>
        </div>
        <div className="element">
          <img src={wind_icon} alt="" className="icon" />
          <div className="data">
            <div className="wind-rate">18 km/h</div>
            <div className="text">Wind Speed</div>

          </div>
        </div>
      </div>
    </div>
  );
};

export default WeatherApp;
```

- **Index.html**

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop. See
https://developers.google.com/web/fundamentals/web-app-manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the
build.
```

```
      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-root public URL.
      Learn how to configure a non-root public URL by running `npm run build`.
    -->
    <title>Weather App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
    <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.
      The build step will place the bundled scripts into the <body> tag.

      To begin the development, run `npm start` or `yarn start`.
      To create a production bundle, use `npm run build` or `yarn build`.
    -->
  </body>
</html>
```

OutPut