

```

import { Oval } from 'react-loader-spinner';
import React, { useState } from 'react';
import axios from 'axios';
import { FontAwesomeIcon } from '@fortawesome/react-fontawesome';
import { faFrown } from '@fortawesome/free-solid-svg-icons';
import './App.css';

function GetWeather() {
  const [input, setInput] = useState('');
  const [weather, setWeather] = useState({
    loading: false,
    data: {},
    error: false,
  });

  const currentDateFunction = () => {
    const months = [
      'January', 'February', 'March', 'April', 'May', 'June',
      'July', 'August', 'September', 'October', 'November', 'December',
    ];
    const weekdays = [
      'Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
      'Saturday',
    ];
    const currentDate = new Date();
    const date = `${weekdays[currentDate.getDay()]}, ${
      months[currentDate.getMonth()] ${currentDate.getDate()} `;
    return date;
  };

  const fetchWeather = async (event) => {
    if (event.key === 'Enter') {
      event.preventDefault();
      setInput('');
      setWeather({ ...weather, loading: true });
      const url = 'https://api.openweathermap.org/data/2.5/weather';
      const api_key = '91628efe4c4a076b71277d3b0981c0e3';
      await axios
        .get(url, {
          params: {
            q: input,
            units: 'metric',
            appid: api_key,
          },
        })
        .then((res) => {
          console.log('res', res);
          setWeather({ data: res.data, loading: false, error:
false });
        })
        .catch((error) => {
          setWeather({ ...weather, data: {}, error: true });
          setInput('');
          console.log('error', error);
        });
    }
  };

  return (
    <div className="App">
      <h1 className="app-name">Weather App</h1>
      <div className="search-bar">
        <input
          type="text"

```

```

        className="city-search"
        placeholder="Enter city name"
        name="query"
        value={input}
        onChange={(event) => setInput(event.target.value)}
        onKeyDown={fetchWeather}
      />
    </div>
    {weather.loading && (
      <>
        <br />
        <br />
        <Oval type="Oval" color="black" height={100} width={100} />
      </>
    )}
    {weather.error && (
      <>
        <br />
        <br />
        <span className="error-message">
          <FontAwesomeIcon icon={faFrown} />
          <span style={{ fontSize: '20px' }}>City not found</span>
        </span>
      </>
    )}
    {weather && weather.data && weather.data.main && (
      <div>
        <div className="city-name">
          <h2>
            {weather.data.name},
            <span>{weather.data.sys.country}</span>
          </h2>
        </div>
        <div className="date">
          <span>{currentDateFunction()}</span>
        </div>
        <div className="icon-temp">
          <img
            className=""
            src={`https://openweathermap.org/img/wn/${
              weather.data.weather[0].icon}@2x.png`}
            alt={weather.data.weather[0].description}
          />
          {Math.round(weather.data.main.temp)}<sup
            className="deg">&deg;C</sup>
        </div>
        <div className="des-wind">
          <p>{weather.data.weather[0].description}</p>
          <p>Wind Speed: {weather.data.wind.speed} m/s</p>
        </div>
      </div>
    )}
  </div>
);
}

export default GetWeather;

```