TASK-10

10. Use React to Render Nested Components:

– *Render multiple `WeatherInfo` components for different cities inside `WeatherApp`.*

Steps for Task 10:

1. *Ensure that WeatherApp Fetches Data for Multiple Cities:*
   - *You should already have an array of cities (e.g., ['London', 'New York', 'Tokyo']) in your WeatherApp.jsx component.*
   - *The fetchWeatherData function will loop through these cities and fetch the weather data for each one.*
2. *Render Multiple WeatherInfo Components:*
   - *You will pass each city's weather data as props to a separate WeatherInfo component.*
   - *You can map over the weatherData array, which contains the weather information for each city, and render a WeatherInfo component for each item in the array.*

– ***Updated Code for Task 10:***

Update WeatherApp.jsx to Render Multiple WeatherInfo Components:

```
// src/components/WeatherApp.jsx
import React, { useState, useEffect } from 'react';
import WeatherInfo from './WeatherInfo'; // Import the WeatherInfo component

const API_KEY = 'your-api-key'; // Replace with your OpenWeather API key

const WeatherApp = () => {
  // State to store the weather data
  const [weatherData, setWeatherData] = useState([]);

  useEffect(() => {
    const cities = ['London', 'New York', 'Tokyo']; // Example cities
    const fetchWeatherData = async () => {
      const weatherPromises = cities.map(async (city) => {
        // Fetch data from OpenWeather API
        const response = await
fetch(`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${API_KEY}&units=
metric`);
        const data = await response.json();
        // Return the data in a format we can use
        return {
          city: data.name,
          temperature: data.main.temp,
          description: data.weather[0].description,
        };
      });

      // Wait for all fetches to complete
      const results = await Promise.all(weatherPromises);
      setWeatherData(results); // Set the weather data in the state
    };

    fetchWeatherData();
  }, []);

  return (
    <div className="weather-app">
      <h1>Weather App</h1>
      {weatherData.length === 0 ? (
        <p>Loading...</p> // Show a loading message until data is fetched
      ) : (
        weatherData.map((weather, index) => (
          // Render a WeatherInfo component for each city's data
          <WeatherInfo
            key={index}
            city={weather.city}
            temperature={weather.temperature}
            description={weather.description}
          />
        ))
      )}
    </div>
  );
};

export default WeatherApp;
```

- ***Run the App:***

```
npm start
```