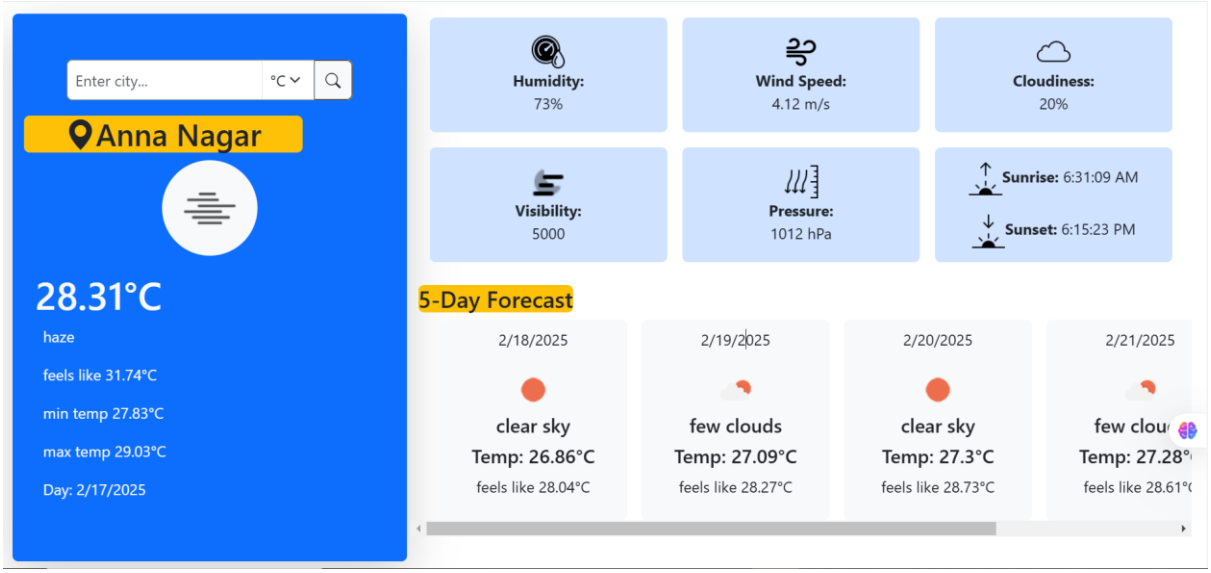
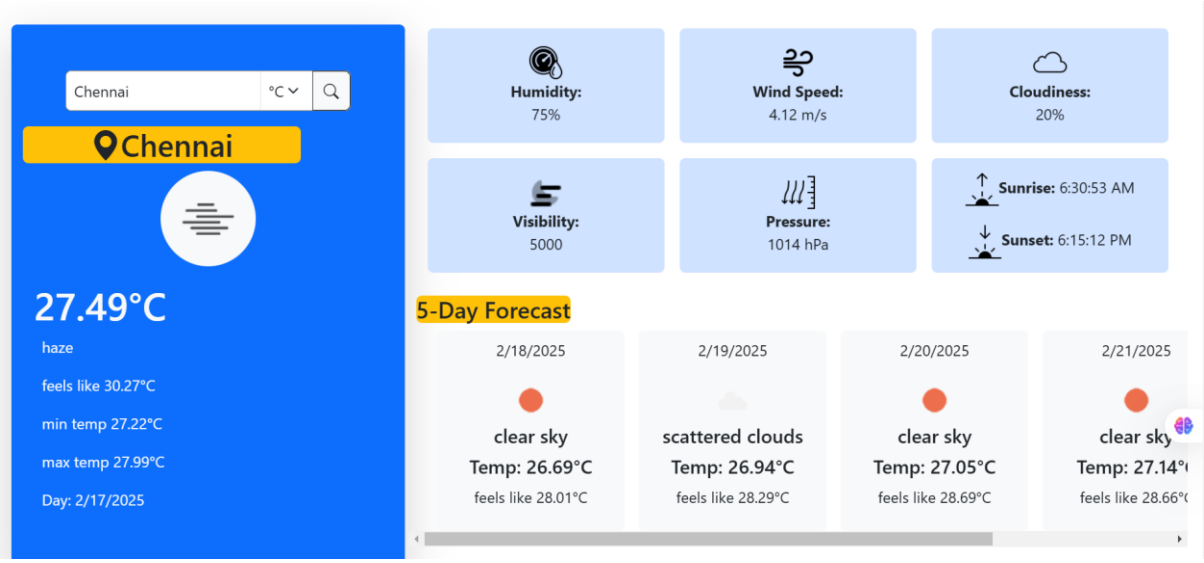
Assignment-06

Create a weather forecast application

Try out my page: https://anyday-weather.web.app/

Output:





Code:

App.js;

```
import React from "react";
import WeatherApp from "./WeatherApp";

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

export default App;
```

weatherApp.js:

```
import WeatherDetails from "./WeatherDetails";
import Forecast from "./Forecast";
import React, { useState, useEffect, useCallback } from "react";
import axios from "axios";
import './index.css'
import "bootstrap/dist/css/bootstrap.min.css";
import 'bootstrap-icons/font/bootstrap-icons.css';

const API_KEY = "45fa98156fff63d1d43fdbdad56bc903";
const WEATHER_URL = "https://api.openweathermap.org/data/2.5/weather";

function WeatherApp() {
  const [city, setCity] = useState("");
  const [unit, setUnit] = useState("metric");
  const [weatherData, setWeatherData] = useState(null);
  const [lat, setLat] = useState(null);
  const [lon, setLon] = useState(null);
  const [loading, setLoading] = useState(false);

  // Function to fetch weather data by coordinates
  const fetchWeatherByCoords = useCallback(async (latitude, longitude) => {
    if (!latitude || !longitude) return;
    setLoading(true);
    try {
      const response = await axios.get(
        `${WEATHER_URL}?lat=${latitude}&lon=${longitude}&appid=${API_KEY}&unit
s=${unit}`
```

```
      );
      setWeatherData(response.data);
    } catch (error) {
      console.error("Error fetching weather data", error);
    } finally {
      setLoading(false);
    }
  }, [unit]);

  // Get user location when the component mounts
  useEffect(() => {
    navigator.geolocation.getCurrentPosition(
      (position) => {
        setLat(position.coords.latitude);
        setLon(position.coords.longitude);
        fetchWeatherByCoords(position.coords.latitude,
position.coords.longitude);
      },
      (error) => console.error("Error getting location", error)
    );
  }, [fetchWeatherByCoords]);

  // Fetch weather when lat/lon changes
  useEffect(() => {
    if (lat && lon) fetchWeatherByCoords(lat, lon);
  }, [lat, lon, fetchWeatherByCoords]);

  // Handle city search
  const handleSearch = async () => {
    if (!city) return;
    setLoading(true);
    try {
      const locationRes = await
axios.get(`${WEATHER_URL}?q=${city}&appid=${API_KEY}`);
      const { lat, lon } = locationRes.data.coord;
      setLat(lat);
      setLon(lon);
      fetchWeatherByCoords(lat, lon);
    } catch (error) {
      alert("City not found");
    } finally {
      setLoading(false);
    }
  };

  return (
    <div className="container-fluid py-2 px-4">
      <div className="row">
```

```
        {/* Search Bar & Current Weather */}
        <div className="con col-md-4 bg-primary text-light py-5 rounded
shadow-lg d-flex flex-column">
          <div className="input-group d-flex bg-light border border-secondary
rounded align-self-center"
               style={{width: '300px'} }>
            <input
                type="text"
                className="form-control p-2 "
                placeholder="Enter city..."
                value={city}
                onChange={(e) => setCity(e.target.value)}
                style={{width: '190px'}}
            />
            <select
                className="form-select p-2"
                value={unit}
                style={{width: '40px'}}
                onChange={(e) => setUnit(e.target.value)}
            >
                <option value="metric">°C</option>
                <option value="imperial">°F</option>
            </select>
            <button className="btn p-2 btn-outline-dark"
onClick={handleSearch} style={{width: '40px'}}>
                <i className="bi bi-search"></i>
            </button>
          </div>

          {loading ? (
            <p>Loading weather...</p>
          ) : weatherData ? (
            <div className="cont">
              <div className="text-center my-3" d-flex flex-column>
                <h2 className="bg-warning d-flex rounded justify-content-
center text-dark"  style={{width: '75%'}}><i class="bi bi-geo-alt-
fill"></i>{weatherData?.name || "Loading..."}</h2>
                <img
                    className="bg-light img-fluid rounded-circle"
                    src={`http://openweathermap.org/img/wn/${weatherData?.weat
her[0].icon}@2x.png`}
                    alt="weather-icon"
                />
              </div>
              <div>
                <h1>{weatherData?.main.temp}°{unit === "metric" ? "C" :
"F"}</h1>
              <div className="det">
```

```
                <p>{weatherData?.weather[0].description}</p>
                <p>feels like {weatherData?.main.feels_like}°{unit ===
"metric" ? "C" : "F"}</p>
                <p>min temp {weatherData?.main.temp_min}°{unit === "metric" ?
"C" : "F"}</p>
                <p>max temp {weatherData?.main.temp_max}°{unit === "metric" ?
"C" : "F"}</p>
                <p>Day: {new Date(weatherData?.dt *
1000).toLocaleDateString()}</p>
              </div>
              </div>
            </div>
          ) : (
            <p>No data available</p>
          )}
        </div>

        {/* Weather Details & Forecast */}
        <div className="col-md-8 justify-self-center">
          {weatherData && <WeatherDetails data={weatherData} />}
          {weatherData && <Forecast lat={lat} lon={lon} unit={unit} />}
        </div>
      </div>
    </div>
  );
}

export default WeatherApp;
```

WeatherDetails.js:

```
import React from "react";
import './new.css';
const WeatherDetails = ({ data }) => {
  return (
    <div className="container mt-4">
      <div className="row g-3 justify-content-center">
        <div className="col-6 col-md-4">
          <div className="bg-primary-subtle rounded p-3 text-center weather-
box">
          <img width="35" height="35" src="https://img.icons8.com/ios-
filled/50/humidity.png" alt="humidity"/>
            <strong>Humidity:</strong> {data.main.humidity}%
          </div>
```

```
        </div>
        <div className="col-6 col-md-4">
          <div className="bg-primary-subtle rounded p-3 text-center weather-
box">
          <img width="35" height="35" src="https://img.icons8.com/ios-
filled/50/wind--v1.png" alt="wind--v1"/>
            <strong>Wind Speed:</strong> {data.wind.speed} m/s
          </div>
        </div>
        <div className="col-6 col-md-4">
          <div className="bg-primary-subtle rounded p-3 text-center weather-
box">
          <img width="35" height="35"
src="https://img.icons8.com/ios/50/cloud--v1.png" alt="cloud--v1"/>
            <strong>Cloudiness:</strong> {data.clouds.all}%
          </div>
        </div>
        <div className="col-6 col-md-4">
          <div className="bg-primary-subtle rounded p-3 text-center weather-
box">
          <img width="35" height="35"
src="https://img.icons8.com/plumpy/24/fog-night--v1.png" alt="fog-night--v1"/>
            <strong>Visibility:</strong> {data.visibility}
          </div>
        </div>
        <div className="col-6 col-md-4">
          <div className="bg-primary-subtle rounded p-3 text-center weather-
box">
          <img width="35" height="35"
src="https://img.icons8.com/ios/50/atmospheric-pressure.png" alt="atmospheric-
pressure"/>
            <strong>Pressure:</strong> {data.main.pressure} hPa
          </div>
        </div>
        <div className="col-6 col-md-4">
          <div className="bg-primary-subtle rounded p-3 text-center weather-
box-1">
            <div>
              <img width="35" height="35" src="https://img.icons8.com/ios-
filled/50/sunrise.png" alt="sunrise"/>
              <strong>Sunrise:</strong> {new Date(data.sys.sunrise *
1000).toLocaleTimeString()} <br />
            </div>
           <div>
            <img width="35" height="35" src="https://img.icons8.com/ios-
filled/50/sunset.png" alt="sunset"/>
            <strong>Sunset:</strong> {new Date(data.sys.sunset *
1000).toLocaleTimeString()}
```

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

export default WeatherDetails;
```

Forecast.js:

```
import React, { useEffect, useState } from "react";
import axios from "axios";
import './new.css';

const API_KEY = "cbb28006f792f3d036949baf4ecddd6d"; // Replace with your API
key

const Forecast = ({ lat, lon, unit }) => {
  const [forecastData, setForecastData] = useState([]);

  useEffect(() => {
    if (!lat || !lon) {
      console.error("Latitude or Longitude is missing! API request skipped.");
      return;
    }

    const fetchForecast = async () => {
      try {
        const response = await axios.get(
          `https://api.openweathermap.org/data/2.5/forecast?lat=${lat}&lon=${l
on}&appid=${API_KEY}&units=${unit}`
        );

        const dailyForecast = extractDailyForecast(response.data.list);
        setForecastData(dailyForecast);
      } catch (error) {
        console.error("Error fetching forecast data", error);
      }
    };

    fetchForecast();
  }, [lat, lon, unit]);

  const extractDailyForecast = (data) => {
```

```
    const dailyData = {};

    data.forEach((item) => {
      const date = item.dt_txt.split(" ")[0];
      const time = item.dt_txt.split(" ")[1];

      if (!dailyData[date] || time === "12:00:00") {
        dailyData[date] = item;
      }
    });

    return Object.values(dailyData).slice(1, 6);
  };

  return (
    <div className="mt-4">
      <h4 className="bg-warning d-inline-flex rounded">5-Day Forecast</h4>
      {forecastData.length === 0 ? (
        <p>Loading or No Data Available...</p>
      ) : (
        <div className="deta row g-3 d-flex">
          {forecastData.map((day, index) => (
            <div key={index} className="col-md-3 p-2 bg-light text-center
rounded">
              <p>{new Date(day.dt * 1000).toLocaleDateString()}</p>
              <img
                src={`http://openweathermap.org/img/wn/${day.weather[0].icon}.
png`}
                alt="weather-icon"
              />
              <h5>{day.weather[0].description}</h5>
              <h5>Temp: {day.main.temp}°{unit === "metric" ? "C" : "F"}</h5>
              <p>feels like {day.main.feels_like}°{unit === "metric" ? "C" :
"F"}</p>
            </div>
          ))}
        </div>
      )}
    </div>
  );
};

export default Forecast;
```

new.css:

```css
.weather-box {
    height: 120px;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
  }
  .weather-box-1{
    height: 120px;
    display: flex;
    flex-direction: column;
    gap: 20px;
    align-items: center;
    justify-content: center;
  }

  /* Media Query for screens below 360px */
  @media (max-width: 360px) {
    .weather-box {
      height: 90px; /* Reduce height */
      font-size: 14px; /* Decrease font size */
      padding: 8px; /* Reduce padding */
    }

    .input-group{
        scale:0.87;
        margin-bottom: 0;
    }

    .cont{
        margin: 0;
        scale: 0.7;
        display: flex;
        align-self: center;
        gap: 20px;
        width: 100vw;
    }

    .det{
        display: flex;
        flex-direction: row;
        flex-wrap: wrap;
    }
    .con{
        padding-top: 0;
        height: 300px !important;
    }
}
```

```css
@media (max-width: 320px){
    .cont{
        scale: 0.65;
    }
}

@media (min-width: 360px){
    .det{
        margin-left: 20px;
    }
    .con{
        margin-top: 20px;
    }
    H1{
        margin-left: 12px;
    }
}

.deta{
    flex-wrap: nowrap !important;
    gap: 15px;
    overflow-x: auto !important;
    padding-left: 30px;
}
```