VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

WEB PROGRAMMING PROJECT - WEATHER APP

Creators – Anupra Agrawal (22BCE0079) & Ramyani Palit (22BCE3935)

React: https://drive.google.com/drive/folders/1XNi7bNBCVugB9Fy35NhTPj-2BxA5fHZX

Java-Script: https://github.com/anupra28/Weather-Web-Application

---

Aim:

The Weather App is a React-based web application designed to provide real-time weather information for any location across the globe. It leverages external APIs to fetch and display data dynamically, offering users an intuitive and visually appealing interface to check the weather.

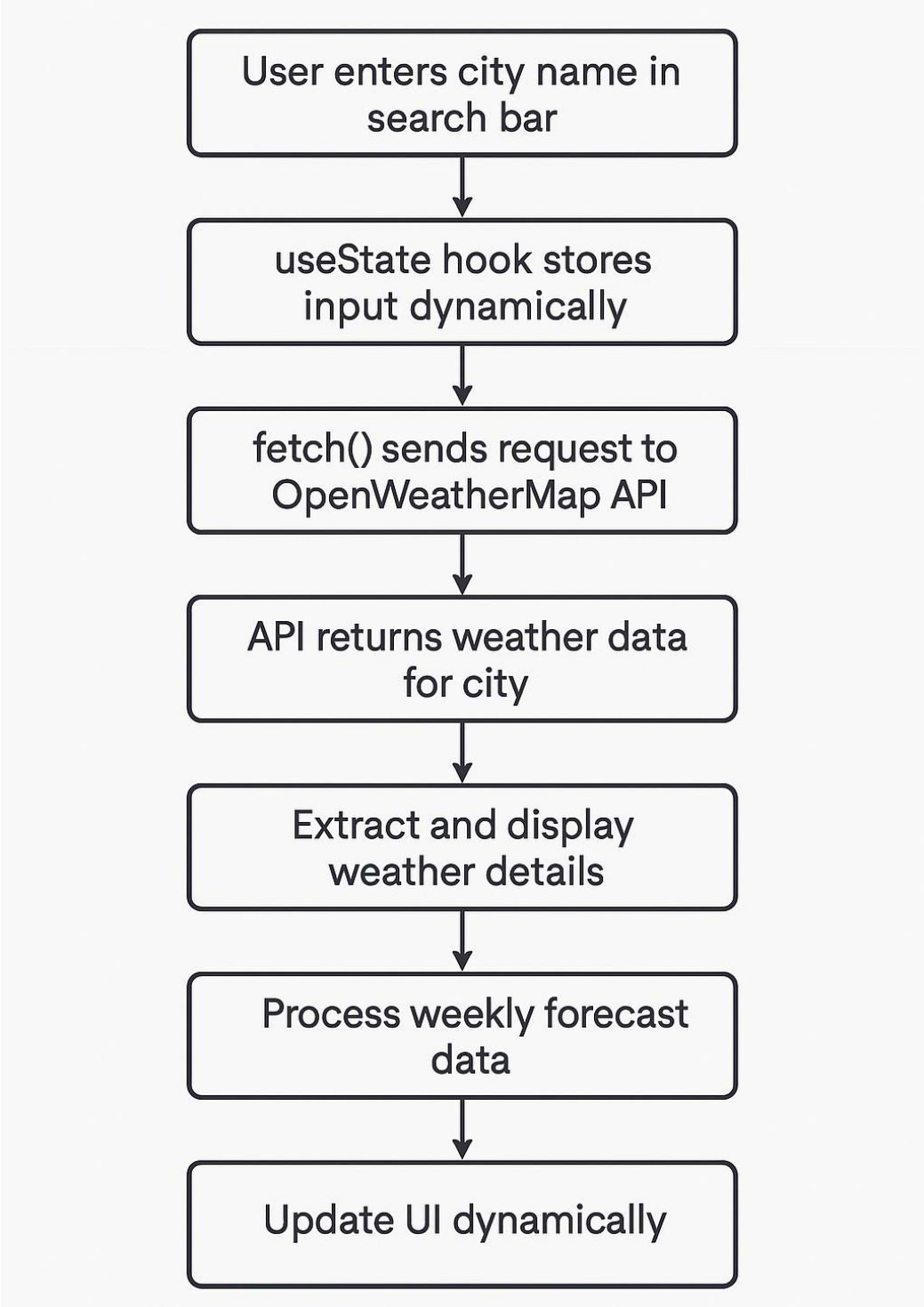
Objectives:

- Implement a real-time weather tracking system using an external API.
- Develop a responsive and interactive user interface using React.
- Enable temperature unit conversion between Celsius and Fahrenheit dynamically.
- Display detailed weather conditions, including temperature, humidity, wind speed, visibility, and atmospheric pressure.
- Provide an interactive weekly weather forecast for better planning.
- Implement error handling mechanisms for invalid inputs and API failures.
- Improve understanding of React, API handling, and UI responsiveness.

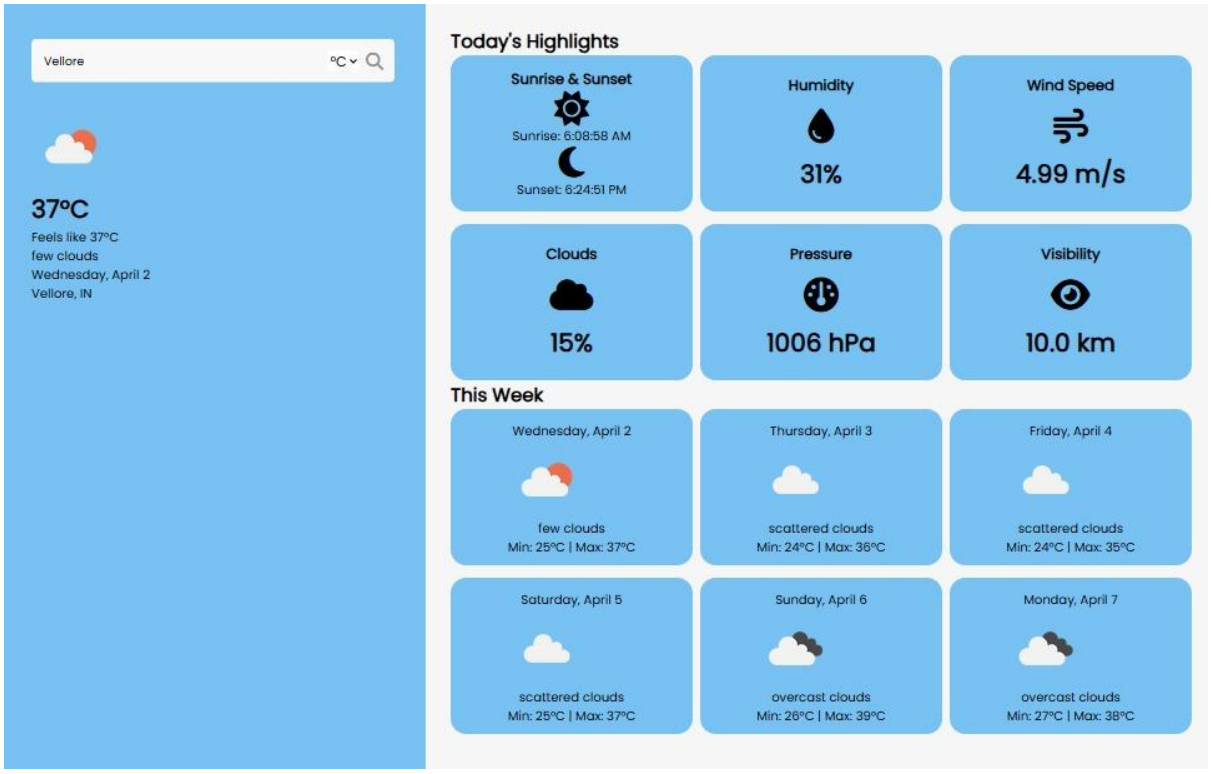
Project Workflow:

1. The user enters a city name in the search bar.
2. The React useState hook manages the location input and stores user input dynamically.
3. The fetch() function sends a request to the OpenWeatherMap API.
4. The API returns weather data for the searched city.
5. The app extracts and displays current weather details, including temperature, humidity, wind speed, and more.

6. The temperature unit conversion function updates values dynamically when the user toggles between Celsius and Fahrenheit.
7. The weekly forecast data is processed by filtering API data for daily trends.
8. All elements update dynamically without requiring a page refresh for a smooth user experience.

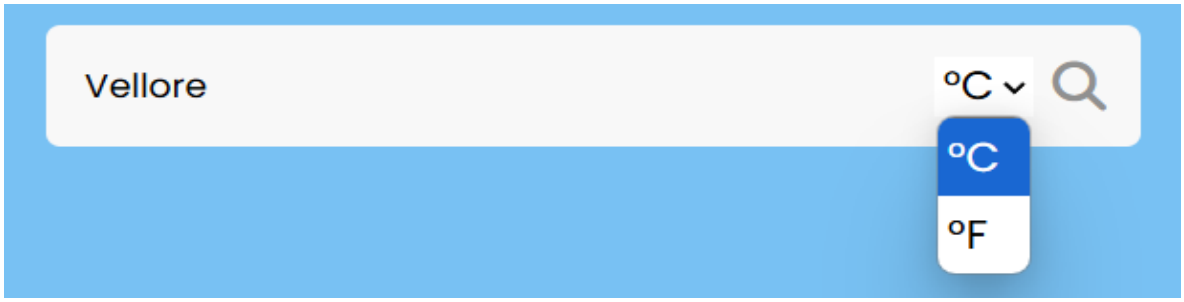


Modules & Functionalities



1. **Search & Temperature Conversion**

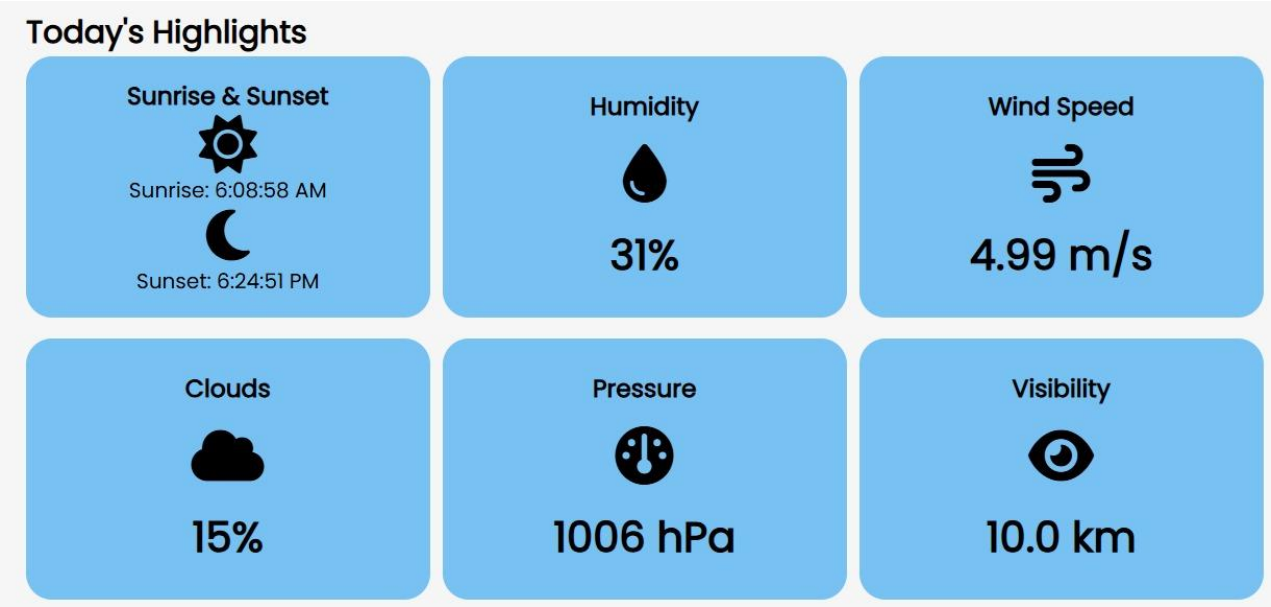
    - Users can enter a city name in the search bar to get instant weather updates.
    - The app retrieves and displays the current weather for the specified city using React state management.
    - A toggle button allows users to switch between Celsius and Fahrenheit dynamically, ensuring better usability across different regions.



2. **Today's Highlights Section**
   This section provides a detailed breakdown of the current weather conditions, including:

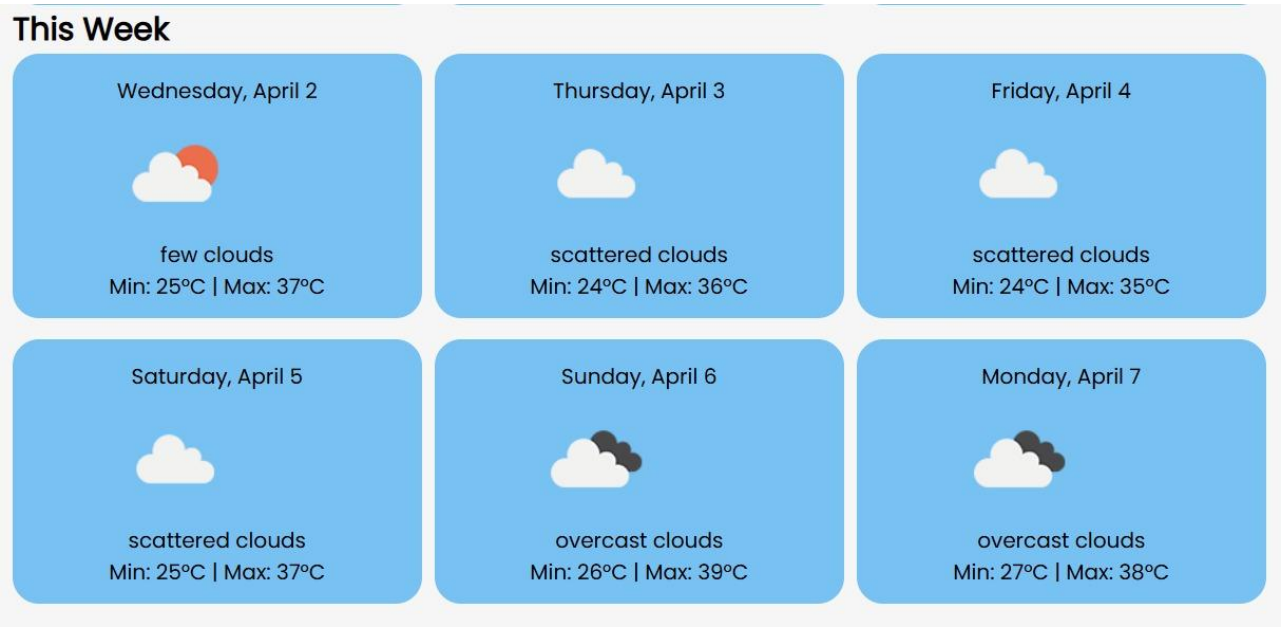
    - **Humidity:** Displays the air moisture level in percentage.
    - **Wind Speed:** Indicates wind speed measured in meters per second.
    - **Sunrise & Sunset:** Shows the local time for sunrise and sunset based on the city's time-zone.

- **Cloud Coverage:** Displays the percentage of cloud cover in the sky.
- **Visibility:** Indicates how clear or foggy the atmosphere is, measured in kilometres.
- **Pressure:** Shows the atmospheric pressure in hectopascals (hPa), which helps predict weather patterns.



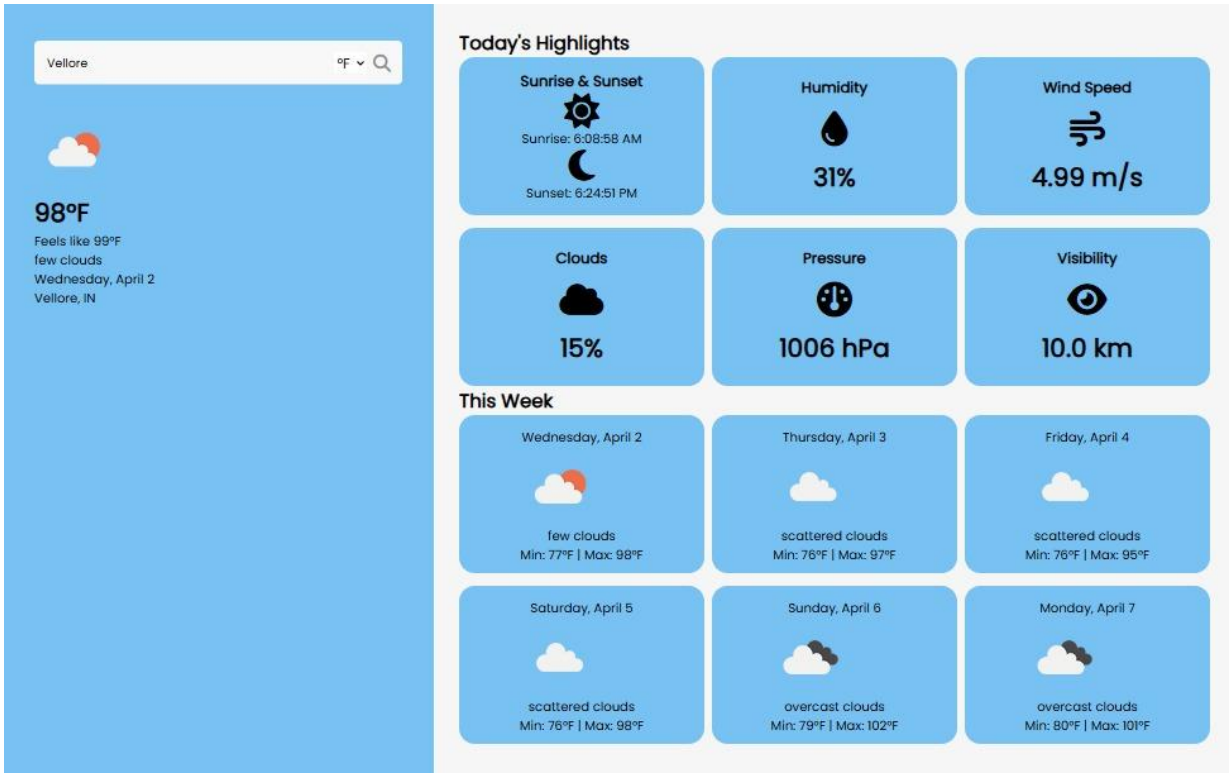
3. **Weekly Weather Forecast**

- The app fetches a seven-day weather forecast using the API.
- Each day in the forecast includes:
  1. Minimum and maximum temperatures.
  2. An icon representing the weather conditions (sunny, rainy, cloudy, etc.).
- The forecast helps users plan their week ahead effectively.



4. **Current Weather Display**

    - **Weather Icon & Description:** Displays an icon and text describing the weather conditions.
    - **Temperature:** Displays the actual temperature along with a "feels like" temperature for accuracy.
    - **Location & Date:** Shows the city name, country, and the current date in an easy-to-read format.



<u>Technologies Used:</u>

Frontend Technologies:

- **React.js:** Handles UI components and state management.
- **CSS3:** Manages the styling and layout using flexbox and grid for responsiveness.
- **JavaScript (ES6+):** Handles interactivity, API requests, and dynamic UI updates.

External APIs:

- **OpenWeatherMap API:** Fetches real-time weather and forecast data.
- **OpenWeather Icons:** Provides visual representations of weather conditions.

User Interface Enhancements:

- **React State Management** for efficient data handling.
- **CSS Grid & Flexbox** for responsive layouts.
- **FontAwesome Icons** to enhance UI with relevant icons.

- **Dark & Light Theme Colors** to improve visibility and readability.

Future Improvements & Enhancements:

1. **Geolocation Feature**: Enable auto-detection of the user's location using the browser's geolocation API for instant weather updates.

2. **Dark Mode Option**: Add a dark mode toggle to enhance user experience in low-light conditions.

3. **Hourly Forecast Feature**: Provide hourly weather updates for better short-term predictions.
4. **Weather-Based UI Themes**: Change background colours dynamically based on the current weather:

    - **Sunny:** Warm colours like yellow/orange.
    - **Rainy:** Cool blue tones.
    - **Snowy:** White and light blue shades.

5. **Air Quality Index (AQI) Integration**: Display air quality levels along with weather conditions to inform users about pollution levels.

What We Learned from This Project:

- **React & Component-Based Development:** Structuring a scalable and modular UI.

- **Working with APIs:** Fetching, handling, and filtering real-time data.

- **Asynchronous JavaScript:** Using fetch() API and promises for API handling.

- **State Management in React:** Effectively using useState and useEffect hooks.

- **Data Optimization:** Reducing unnecessary API calls for better efficiency.

- **UI/UX Principles:** Focusing on a clean and intuitive interface for an enhanced user experience.

- **Debugging & Error Handling:** Implementing effective ways to manage incorrect user inputs.