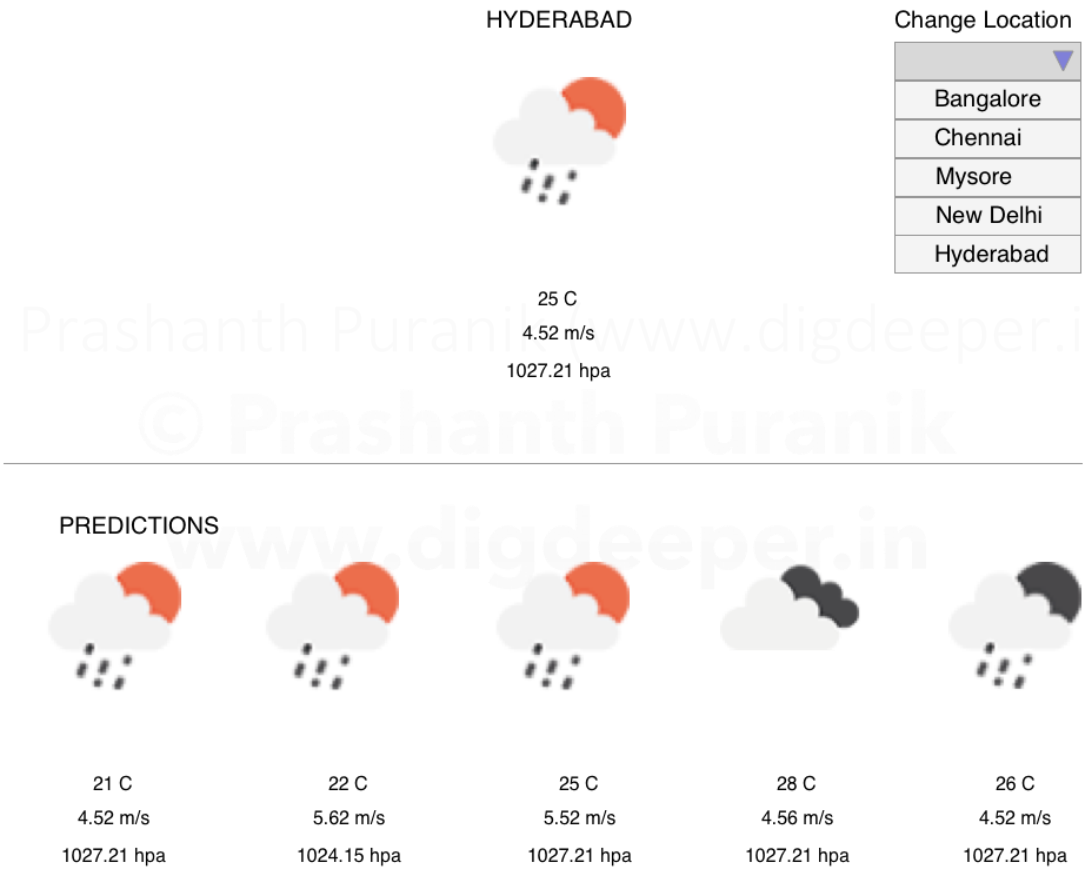
WEATHER APP

Use Open Weather Map API (https://openweathermap.org) to build a weather app.

Mock-ups

Page 1: Home



Page 2: Settings

Functionality

1. Shows the details of weather for the user's current location. It uses geolocation to get location. The app shows a nice picture indicating the weather.

Reference:
https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_geolocation

2. Shows the details of weather prediction for next 5 days at same time as the current time. These should be displayed as 5 cards next to each other. Again, use a picture to indicate state of weather.

3. Stores all previous user locations locally in the browser (using local storage). A list of all previous locations is displayed under Locations dropdown. When user selects a different city, the page is refreshed with weather for it.

Reference:
https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_webstorage_local

4. The user can also search for a city and see weather details change to the one for searched city.

5. The app also has a settings screen. This lets the user set the following.
   a) Add/Remove cities to/from the previous locations list
   b) Change the default location for which weather is displayed (may be set to 'Current' or one of the cities from the previous locations list)
   c) Change units for temperature display etc. - Metric/Imperial

Additional Requirements

UI to be designed and developed by your team

You need to come up with your own design of components / pages to **support at least 2 of the following 4 features**. Design user-friendly interfaces. Take care of all different application states (loading, error, success, no data fallback message), responsive web design (phone, desktop), SEO, performance (appropriate rendering model), web a11y. Conduct Lighthouse audit on your app, and check for web vitals.

NOTE: When developing these pages if you feel the need for more pages (like login page, authorization page etc., please develop those as well).

1. **Hourly Weather Forecast View**
   Description: Implement a detailed hourly forecast view for the next 48 hours.
   - **API Endpoint**: `GET /data/3.0/onecall?lat={lat}&lon={lon}&exclude=current,minutely,daily,alerts&appid={API key}`
   - **Details to Display**
     - Hourly temperature
     - Weather conditions (with appropriate icons)
     - Precipitation probability
     - Wind speed and direction
   - **Purpose**: Provides users with granular weather information to plan their activities effectively.

2. **Weather Alerts Page**
   Description: Create a dedicated page to display active weather alerts for the selected location.
   - **API Endpoint**: `GET /data/3.0/onecall?lat={lat}&lon={lon}&exclude=current,minutely,hourly,daily&appid={API key}`
   - **Details to Display**
     - Alert event name
     - Description of the alert
     - Start and end times
     - Issuing authority
   - **Purpose**: Keeps users informed about severe weather conditions and safety advisories.

3. **Historical Weather Data View**
   Description: Provide users with the ability to view historical weather data for a selected date and location.
   - **API Endpoint**: `GET /data/3.0/onecall/timemachine?lat={lat}&lon={lon}&dt={time}&appid={API key}`
   - **Details to Display**
     - Temperature trends throughout the day
     - Weather conditions
     - Precipitation levels
     - Wind patterns
   - **Purpose**: Allows users to analyze past weather conditions, which can be useful for various personal or professional reasons.

4. **Air Quality Index (AQI) Display**

Description: Integrate air quality information into the app to inform users about the pollution levels in their area.

- **API Endpoint**: `GET /data/2.5/air_pollution?lat={lat}&lon={lon}&appid={API key}`
- **Details to Display**
    - AQI value
    - Concentrations of pollutants like PM2.5, PM10, NO_2, SO_2, O_3, CO
    - Health recommendations based on AQI levels
- **Purpose**: Provides users with critical information about air quality, enabling them to take necessary precautions.