Thanks for your interest in “Front-End Development Intern” position at Smart Decision Advisory Services.

Here are the next steps.

1. Please review the assignment below.
2. Submit your assignment on or before July 12, 2024, through the Form at https://forms.office.com/r/CPD0BAnFrc. Please note that the form also has few other questions to help us evaluate candidates.
3. Shortlisted candidates with good quality submissions will be invited for technical interview (starting Monday 8th July).

We will highly encourage your submit the form at the soonest as

- early submission of the form (including the complete assignment) will be indicative of your interest and competency.
- we may start the interview process early and select the candidate early, if we get good quality early submissions.

All the Best! Looking forward to your form & assignment submission.

****

Assignment: Weather Dashboard

Description:
Build a weather dashboard that provides current weather information and forecasts for different cities. The application should allow users to search for cities and display the current weather and a 7-day forecast. Use React for the front-end, Material-UI for styling, and a weather API (such as OpenWeatherMap) for data.
Objectives:

- Fetch and display real-time weather data.
- Implement data visualization for temperature trends.
- Ensure the application is responsive and mobile-friendly.

Requirements:

1. **Set up a React Application**:
   - Initialize a new React project using Create React App.
   - Set up Material-UI for styling the application.
2. **User Authentication (Optional)**:
   - Implement user authentication allowing users to sign up, log in, and log out.
   - Store user preferences, such as favorite cities, in the user profile.
3. **Search Functionality**:
   - Implement a search bar allowing users to search for different cities.
   - Fetch weather data for the searched city using a weather API (e.g., OpenWeatherMap).

    - Display a list of search suggestions as the user types.

4. **Display Current Weather**:
    - Display the current weather information for the selected city.
    - Include details such as temperature, humidity, wind speed, and weather conditions.
    - Use Material-UI components like `Card` and `Typography` for styling.
5. **7-Day Forecast**:
    - Display a 7-day weather forecast for the selected city.
    - Include details such as daily temperatures, weather conditions, and precipitation.
    - Use Material-UI components and icons to enhance the visual representation.
6. **Data Visualization**:
    - Implement charts to display temperature trends over the next 7 days.
    - Use a chart library like Chart.js or Recharts for visualization.
    - Display line charts or bar charts to represent temperature changes.
7. **Responsive Design**:
    - Ensure the application is fully responsive and works well on various screen sizes, including mobile devices.
    - Use Material-UI's grid system and responsive utilities to achieve a mobile-friendly layout.
8. **Error Handling**:
    - Handle errors gracefully, such as displaying appropriate messages when the weather data cannot be fetched.
    - Ensure the application handles edge cases, such as invalid city names.
9. **Optional Features**:
    - Allow users to save their favorite cities and quickly access weather information for those cities.
    - Implement a feature to display weather alerts or notifications.
    - Provide localization support to display weather information in different languages.

Deliverables:

- A link to the GitHub repository containing the source code.
- A brief document (README.md) explaining the project structure, setup instructions, and how to run the application.
- Any additional notes or considerations taken into account while completing the assignment.

Evaluation Criteria:

- **Correctness**: The application should meet the specified requirements and function correctly.
- **Code Quality**: The code should be clean, well-organized, and properly documented.

- **User Interface**: The UI should be visually appealing, user-friendly, and responsive.
- **Data Handling**: Proper handling of API data, including fetching, displaying, and error management.
- **Optional Features**: Implementation of additional features will be considered a plus.

****

Best Regards,

Nitin Acharekar