🚘

Tesla India

Instructions

- You should use json-server for this.
- Use react-router for routing.
- Use redux and thunk for dispatching asynchronous actions.
- Keep the code clean, commented and documented. Maintain feature based coding. (separate action and reducers - feature wise folders) example : all the action and reducer related to authentication in one folder etc.
- You are free to use any css solutions as long as it looks good. Remember this also has score
- You will have to also deploy both JSON server and app.

Problem Statement:

- Tesla company is planning to open their sales all over India.
- So they are planning to make a pre-booking sale, but before that they wanted to know how many people around India are showing interest in this.
- So you need to build an app where they can collect market response and plan their sales accordingly.
- The task is to build two sections.
    - Tesla section
    - User section
- In home screen make two buttons for these two sections, on clicking on buttons move to that particular section.
- Details of each section is given below

Tesla section:

- It will have a pre-booking registration form with following fields
    1. Name
    2. Age
    3. State (make a dropdown of at-least 5 States)
    4. Year of planning (dropdown of at-least 5 years)
- After submitting form, show an alert or popup saying that "successfully registered", also you need to store this information in JSON server for admin to access it

Admin Section:

- This section will have Navbar with three different routes
    - Login `(/login)`
    - List page `(/list)`
    - Reports `(/report)`
- Data and reports should be private routes.

Login page (/login)

- Use reqres API to login as admin (https://reqres.in/) - LOGIN - SUCCESSFUL
- After successful login move admin to data page.

List page (/list)

- In this page all the data which was being filled in user section should be displayed in form of table
- This page should also have filter and sort functionality
  - Filter by State
  - Filter by year of planning
  - Sort by Age
- Each row should also have `delete` icons.
- Admin should be able to `delete` any user who has registered (his entry in json server should also get deleted)
- Each row should also have `edit` icon.
- When Admin clicks on `edit`, Admin should be able to edit/change name of particular user in JSON data (you can use modal for this) and display updated data.

Reports Page (/report)

- In this page admin should have useful reports regarding the registrations. The reports you'll need to build are
  1. Number of people registered in a given age range (13-18, 18-25 and 25+).
  2. Average group size of people registered for Tesla.
  3. No of users planning for particular year.
- Display these reports in form of table.

Bonus:

- Display above reports in form of charts.
- You can use any external library of your choice to achieve charts.

Note:

- Maintain flow of app as mentioned.
- Error messages should be shown, make use of any CSS library of your choice (chakra and MUI preferred)
- Use loaders.
- Good designs will fetch bonus marks.
- Submitting local host links for mock server will lead to disqualification.

Submission

- Please submit deployed link and Github link of code.
- **Push your code into `masai-repo`, don't submit personal repo links (submitting personal repo links will lead to disqualification)**
- Please double check if deployed version works or not (run deployed version on your laptop and then submit it).
- Any issues in deployed link, will be considered null and void.
- Please verify your submissions are correct.
- Make sure you follow all instructions carefully.
- Submit before deadline.

Rubrics / Criteria to be judged upon

- HTML, CSS,JSON server, React, **Redux**
- Filtering, sorting.