Tuition Student Records (Frontend)

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 and similarly for others.
- 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:

- Create login and signup so that each teacher must be able to login into his account and should be able to perform the below functionalities.
- After logging in the main page should show the list of student details along with the total no of tests taken by that particular student (BONUS: Pictures for the students).
- Main page should display Teacher name on Navbar and Teacher should also have ability to signout.
- The main page should have the pagination functionality with the accurate page numbers, use API endpoint for pagination.

- Teacher should also have the ability to filter by gender and sort by age, use relevant API endpoint for sorting and filtering.
- The page should also have the ability to search the Students by name, use API endpoint for searching particular patient by name.
- On clicking on a particular Student it must show a page containing the list of all the tests taken by that particular student in tabular format.
- The Teacher can also see and manage - add/remove tests for a particular student
- Teacher should also have ability to add new student details and tests (you can use modal for that) and delete existing patient.
  - Student - Name, Gender & Age (Can have multiple Tests)
  - Test - Name, Subject, Marks & Date (Belongs to a particular Student)

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.
- ***Deploy JSON server too.
- 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.
- Make sure you follow all instructions carefully.
- Submit before deadline.

Rubrics / Criteria to be judged upon

- HTML, CSS, React
- Redux
- JSON server
- MUI / Chakra UI
- Pixel perfect design
- Code cleanliness

Time Limit - `4 Hours`