React:

Task -1

- ➢ You have to create a 2-page Web-App
- ➢ Page1: with Register/Login user. Page2: Shows users Name, Phone no, address, email, etc. all details of the user after logging in
- ➢ On the 1st page, the person should be able to Register - by filling in information like: Name, Email id, password, Address, phone number, ROLE (ADMIN or GUEST).
- ➢ After logging in, the 2nd page should show all the details of the user logged in with logout button. It should have an edit icon for each field & the user should be able to edit that information. The second page CSS should be different for Admin & Guest. (Background, Font style, etc.)
- ➢ The Frontend can be made in React.
- ➢ Use Firebase for Database, you can interact with the database from the Client-side itself.
- ➢ Host the site using neltify also keep the complete project in GitHub.
- ➢ The UI for both pages should look good & professional.

Task – 2

Create a React application using the below API endpoint:

https://datapeace-storage.s3-us-west-2.amazonaws.com/dummy_data/users.json

Sample Data:

```
{
 id: 1,
 first_name: "James", last_name:
 "Butt",
 company_name: "Benton, John B Jr",
 city: "New Orleans", state: "LA",
 zip: 70116,
 email: "jbutt@gmail.com",
 web: "http://www.bentonjohnbjr.com", age:
 70
}
```

Application functions: (Use own CSS properties / styling library for creating proper UI)

Home Page and Users Page:

1. **Home Page:** Should have a Registration form and Login form and a tab to navigate to **Users Page**. Registration form should have all the fields equivalent to the property names given in the above API endpoint (sample data provided for property name reference). Login form should have username and password fields. Provide 'username' as 'first_name' and 'password' as 'email'.

2. Details entered in the Registration form should be stored in Firebase, as well as displayed in *Users page* as a table with column first_name, last_name, email, web, city, state along with a tab to navigate to **Home Page**.

3. Once the user logged in with the corresponding username (first_name) and password (email), user should redirect to *Details* page on a separate route with LOGOUT option. 4. Website links (web) should open in a new browser tab

Users Page:

This page should have the following options:

Search: Allow to search using first_name or last_name

Pagination: Data should be paginated and the UI should look like: ‹ 1 2 3 ... 15 ›

Sort: All columns should be sortable in both ascending and descending order

Details Page:

1. Details page should show all fields of user with edit option in all the fields. (Updating the field details should reflect in database immediately)

2. Clicking on LOGOUT button should redirect to Login (Home) page

Routing:

Home Page: **'/'**

Users Page: **'/users'**

Details Page: **'/users/<user id>'**

1. Users page and Detail pages should be on their respective routes:
2. Route for Users page should be /users

Route for Details page should /users/<id> (e.g: /users/2, if id of user is 2)