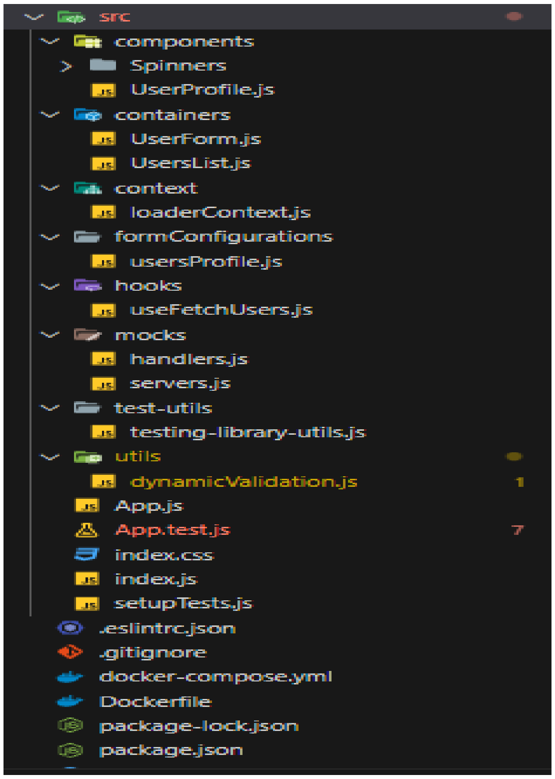
HOW TO RUN THE APP:

1) npm install & npm start   (dev mode)
2) npm install & npm run build & npm install –g server & npm run buildVersion (build mode)
3) If user not connected to docker hub and no login/ authentication required with two possible ways
   - npm run docker-build & npm run docker-run
   - npm run docker-build & docker-compose up

HOW TO TEST THE APP:

1) npm run test



```
src
  components
    Spinners
    UserProfile.js
  containers
    UserForm.js
    UsersList.js
  context
    loaderContext.js
  formConfigurations
    usersProfile.js
  hooks
    useFetchUsers.js
  mocks
    handlers.js
    servers.js
  test-utils
    testing-library-utils.js
  utils
    dynamicValidation.js                1
  App.js
  App.test.js                           7
  index.css
  index.js
  setupTests.js
.eslintrc.json
.gitignore
docker-compose.yml
Dockerfile
package-lock.json
package.json
```

Some useful info for the assessment.

I. React:
II. Tailwind for css
III. Jest for tests
IV. Context for loadingSpinner
V. Formik to handle Form
VI. Yup for formValidation

The project's folder structuring is as appears in the photo provided above.

App.js has two main Containers:

1. **UserForm** (right section of the scrollbar)
2. **UsersList** (left section of the scrollbar)

A custom hook to render the users, handle loading and error state is used in App.js. Can be found under hooks folder.

The **configuration** of the **userForm** can be found **under formConfigurations folder.**

The **validation** applied is **dynamic**. The validation script can be found **under utils folder.**

There are two spinners in the app.

- The first spinner is applied only on the left section when users are loading. This is called customLoadingSpinners. Spinners can be found under compnents/Spinners folder.
- The second spinner is global in the app. Actually a context loader have been implemented under context folder. This spinner applies when the user submit a form to change anther user'

All **the tests for the Five scenarios(user stories/requirements)** requested can be found in **App.test.js.**

Better folder structuring for tests could be applied, but skipped due to limited test cases.