INTRODUCTION

React testing with testing-library and Jest -

1- when we create our project with npx create-react-app command then react pre-install these things for us.

2- it improves the code stability of our project

3- we can perform unit testing, mock data testing, snapshot testing with help of this.

4- its have its own code watch mode.

5 - when we modified the code then all test cases running accordingly.

6- More code writing but can give a benefit in the long term project.

CONFIGURATION

Configuration of testing-library and Jest -

1- it detects the suffix of .test.js

2- Generally create test files with the name of the component files.

3- we can store the test pages in the component folder and in a separate folder as per standard.

4- Run npm run test For run test.

IMPLEMENTATION

1- import required methods from testing lib.

Ex-

import { render, screen } from "@testing-library/react";

import userEvent from "@testing-library/user-event";

2- import the component that needs to test.

Ex-

import { render, screen } from "@testing-library/react";

import userEvent from "@testing-library/user-event";

import Login from "./Login";

3- then follow the syntax for testing.

describe(“text of the testings”, ()=> {

Test or it(“text of the testings”, ()=> {

render(component);

//testing code.

}

}

4- give a data-id in an attribute that we want to test.

Data-id is a special type of attribute in Html when it is custom and globally accessible and ignored by the end user.

Ex-

<input

*type*="email"

*placeholder*="Enter email"

*data-testid*="email-input"

*value*={email}

*onChange*={(*e*) => setEmail(*e*.target.value)}

/>

5- then we mock the data and test the input and output of desired.

Ex-

test("pass valid email to test email input field", () => {

render(<*Login* />);

// for email validattion

const inputEl = screen.getByTestId("email-input");

userEvent.type(inputEl, "test@mail.com");

expect(screen.getByTestId("email-input")).toHaveValue("test@mail.com");

expect(screen.queryByTestId("error-msg")).not.toBeInTheDocument();

});