**Lesson 10 Demo 02**

**Creating a React Component Login Test**

**Objective:** To create a React component login test with user credentials to display the success or failure message

**Tools required:** Node.js and React.js

**Prerequisites:** None

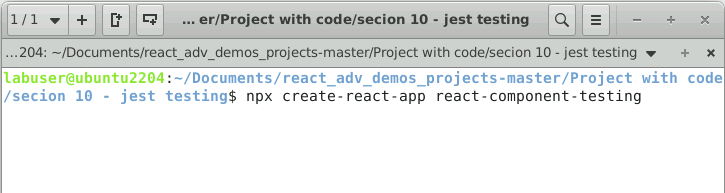
Steps to be followed:

1. Set up a React project
2. Verify test dependencies
3. Create the components
4. Test the application

**Step 1: Set up a React project**

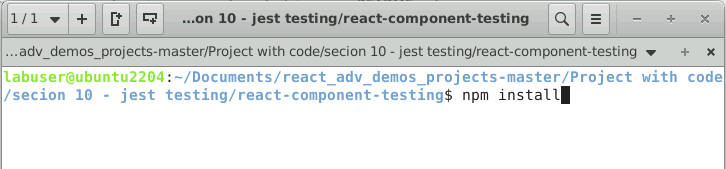
1. Open a terminal window to run the following command to create a React application:

**npx create-react-app react-component-testing**

****

1. Open the terminal window inside the **react-component-testing** folder and run the following command to download the dependencies:

**npm install**



1. Open the project **REACT-COMPONENT-TESTING** in the VS Code editor and view the project structure

A screenshot of a computer program

Description automatically generated

**Step 2: Verify test dependencies**

1. Verify the test dependencies in the **package.json** file as shown below:

**A screenshot of a computer program

Description automatically generated**

**Step 3: Create the components**

1. Create the **Login.js** file with the following code in the src folder:

**import { useState } from "react";**

**function Login() {**

**let[email,setEmailId]=useState("");**

**let[password,setPassword]=useState("");**

**let [msg,setMessage]=useState("");**

**let checkUser= (event)=> {**

**if(email=="admin@gmail.com" && password=="admin@123"){**

**setMessage("successfully login");**

**}else {**

**setMessage("failure try once again");**

**}**

**}**

**return(**

**<div>**

**<h3>Login Page</h3>**

**<form onSubmit={checkUser}>**

**<label>EmailId</label>**

**<input type="email" name="emailid" onChange={(event)=>setEmailId(event.target.value)} placeholder="enter emailid"/><br/>**

**<label>Password</label>**

**<input type="password" name="password" onChange={(event)=>setPassword(event.target.value)} placeholder="enter password"/><br/>**

**<input type="submit" value="SignIn" data-testid="submit-button"/>**

**<input type="reset" value="reset" data-testid="reset-button"/>**

**</form>**

**<br/>**

**<span data-testid="result">{msg}</span>**

**</div>**

**)**

**}**

**export default Login;**

A screen shot of a computer program

Description automatically generated

1. Create the **Login.test.js** file with the following code that is responsible for testing using Jest:

**import { fireEvent, render, screen } from '@testing-library/react';**

**import Login from './Login';**

**describe("Login Testing ",()=> {**

**test('React Login test with success', async() => {**

**render(<Login />);**

**// successfully login testing code**

**const email = screen.getByPlaceholderText('enter emailid');**

**fireEvent.change(email,{target:{value:"admin@gmail.com"}});**

**const password = screen.getByPlaceholderText('enter password');**

**fireEvent.change(password,{target:{value:"admin@123"}});**

**const submitButton = screen.getByTestId('submit-button');**

**fireEvent.submit(submitButton);**

**const result1 = screen.getByTestId("result");**

**console.log(result1.textContent);**

**expect(result1.textContent).toEqual("successfully login");**

**});**

**test('React Login test with failure', async() => {**

**render(<Login />);**

**// failure login testing code**

**const email = screen.getByPlaceholderText('enter emailid');**

**fireEvent.change(email,{target:{value:"admin@gmail.com"}});**

**const password = screen.getByPlaceholderText('enter password');**

**fireEvent.change(password,{target:{value:"admin@1234"}});**

**const submitButton = screen.getByTestId('submit-button');**

**fireEvent.submit(submitButton);**

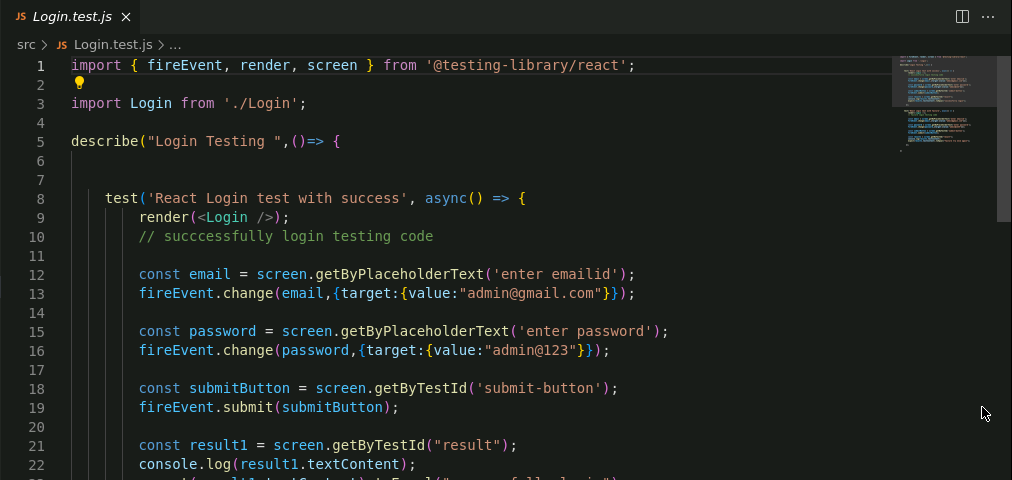
**const result1 = screen.getByTestId("result");**

**console.log(result1.textContent);**

**expect(result1.textContent).toEqual("failure try once again");**

**});**

**})**



**Step 4: Test the application**

1. Open the terminal inside the project and run the application with the help of the following command:

**npm test**

**A screenshot of a computer

Description automatically generated**

After successful execution, the output can be viewed on the terminal as shown below:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Press **a** key to run all test

A screenshot of a computer

Description automatically generated

1. To run the same application, use the command:

**npm start**

A screenshot of a computer

Description automatically generated

The output can be viewed as shown below:

A screenshot of a login page

Description automatically generated

1. Enter the user credentials given below for a successful login test and click on **SignIn**

**EmailId:** [**admin@gmail.com**](mailto:admin@gmail.com)

**Password: admin@123**

A screenshot of a computer

Description automatically generated

The output can be viewed as shown below:

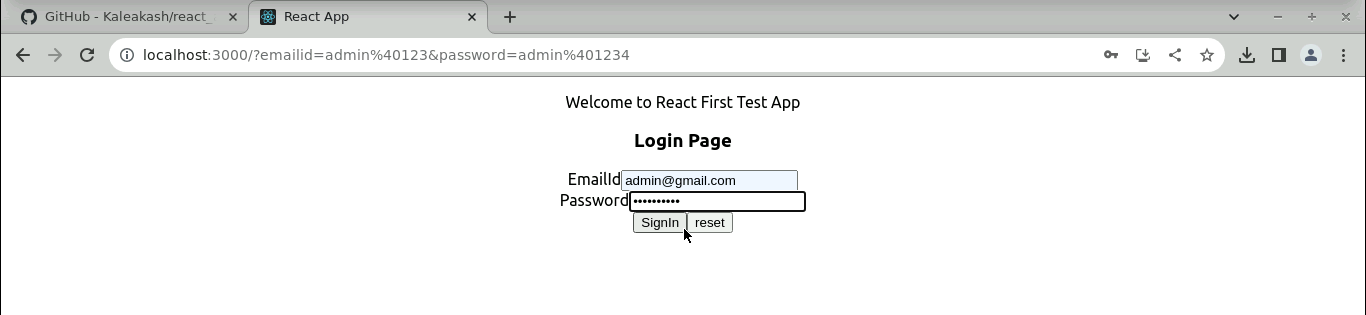
A screenshot of a computer

Description automatically generated

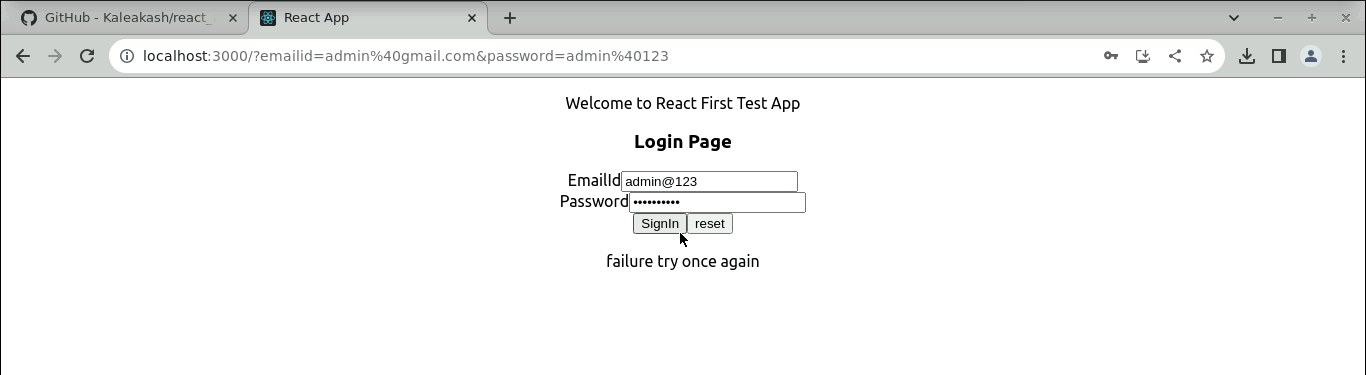
1. Enter the user credentials given below for another login test and click on **SignIn**

**EmailId: admin@gmail.com**

**Password: admin@1234**



The output can be viewed as shown below:



With this, you have successfully created a login component test to display the success or failure message.