**Lesson 01 Demo 02**

**Creating Form validation using custom hook**

**Objective:** To demonstrate the form validation using custom hook.

**Tools required:** Node JS and React JS

**Prerequisites:** HTML, CSS, Bootstrap, JavaScript ES5/ES6, Basic React Concept

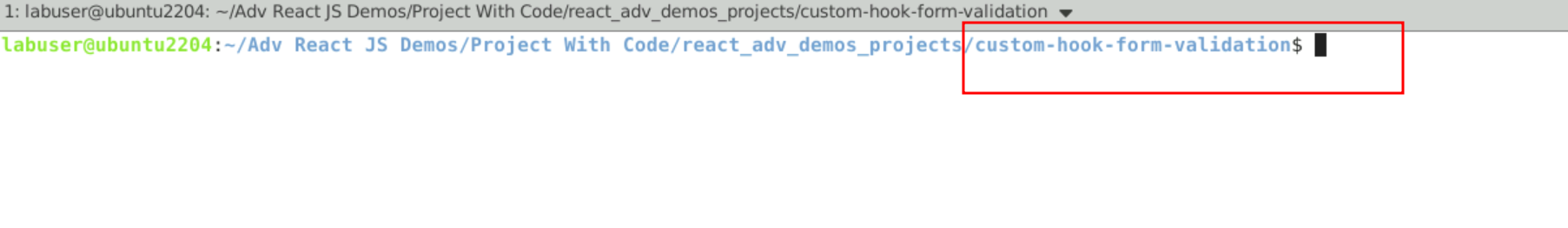
**Note** : All react js project already created with version 18.x with Sample App.js file

**Steps to be followed:**

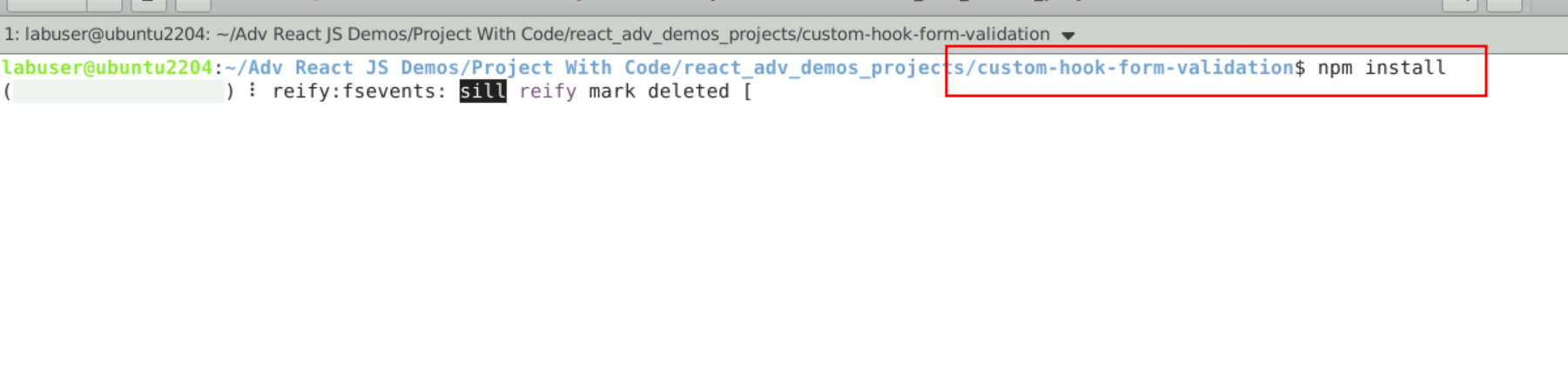
1. Set up for react js project
2. Creating custom hook in separate external file.
3. Use custom hook in user defined components
4. Importing user defined functional component in App.js entry file
5. Adding bootstrap features in index.html file
6. Step to run the application and check the output on browser.

**Step 1: Set up for react js project**

1. Open a terminal window inside a React JS pre-created project ie **custome-hook-form-validation**

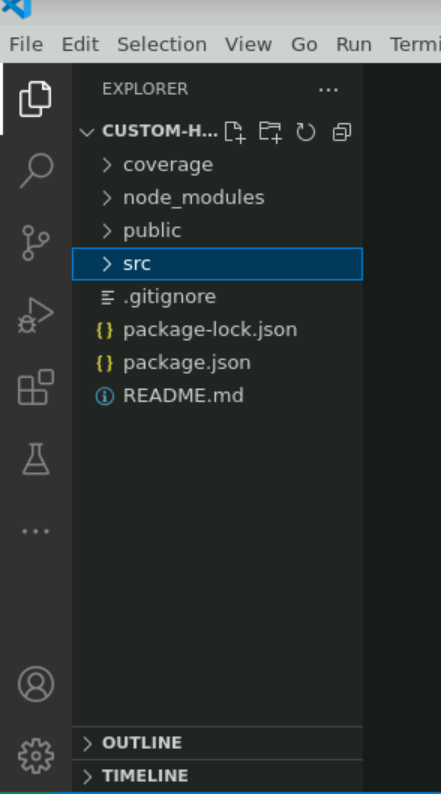


1. Now you need to run the command as **npm install.** This command helps us to installed all required dependencies mention in package.json file in local machine in the form of node\_module folder.



1. Now open **custom-hook-form-validation** folder in VS Code Editor

Note: short cut to open write **code .**



**Step 2: Creating custom hook in separate external file**

1. Now inside src folder create a **useFormValidation.js** file. Inside this file we are writing the logic to create the custom hook. The below Sample code take the help of useState as well as useEffect pre-defined hook function and creating custom hook. Which is responsible to do the form validation for username, emailed and password.

2.2. Below code you need to write inside **useFormValidation.js** file

**useFormValidation.js**

import { useState, useEffect } from 'react';

const useFormValidation = (initialState, validationRules) => {

const [values, setValues] = useState(initialState);

const [errors, setErrors] = useState({});

const [isSubmitting, setIsSubmitting] = useState(false);

useEffect(() => {

const validate = () => {

let valid = true;

const newErrors = {};

for (const fieldName in validationRules) {

if (Object.prototype.hasOwnProperty.call(validationRules, fieldName)) {

const rules = validationRules[fieldName];

const value = values[fieldName];

for (const rule of rules) {

if (rule.required && !value.trim()) {

newErrors[fieldName] = `${fieldName} is required`;

valid = false;

break;

}

if (rule.minLength && value.length < rule.minLength) {

newErrors[fieldName] = `${fieldName} must be at least ${rule.minLength} characters`;

valid = false;

break;

}

// Add more validation rules as needed

}

}

}

setErrors(newErrors);

return valid;

};

if (isSubmitting) {

const isValid = validate();

if (isValid) {

// Perform form submission, e.g., send data to server

console.log('Form data:', values);

setValues(initialState);

setIsSubmitting(false);

} else {

setIsSubmitting(false);

}

}

}, [isSubmitting, values, validationRules, initialState]);

const handleSubmit = (event) => {

event.preventDefault();

setIsSubmitting(true);

};

const handleChange = (event) => {

const { name, value } = event.target;

setValues({ ...values, [name]: value });

};

return {

values,

handleChange,

handleSubmit,

errors,

isSubmitting,

};

};

export default useFormValidation;

**Step 3: Use custom hook in user defined components**

3.1 Now we need to create user defined component. So create **LoginPage.js** file inside src folder and import custom hook code inside **LoginPage.js** file and use custom hook functionality. Inside this component we are using custom component which help to do the form validation.

3.2 Below code you need to write inside a **LoginPage.js** file

import React from 'react';

import useFormValidation from './useFormValidation'; // Import the custom hook

const LoginPage = () => {

const INITIAL\_STATE = {

username: '',

email: '',

password: '',

};

const VALIDATION\_RULES = {

username: [{ required: true, minLength: 3 }],

email: [{ required: true }],

password: [{ required: true, minLength: 6 }],

};

const {

values,

handleChange,

handleSubmit,

errors,

isSubmitting,

} = useFormValidation(INITIAL\_STATE, VALIDATION\_RULES);

return (

<div className='container'>

<form onSubmit={handleSubmit}>

<div className='row'>

<div className='col-4 offset-2'>

<input

type="text"

name="username"

value={values.username}

onChange={handleChange}

placeholder="Username"

/>

</div>

<div className='col-6'>

{errors.username && <span style={{"color":"red"}}>{errors.username}</span>}

</div>

</div>

<div className='row'>

<div className='col-4 offset-2'>

<input

type="email"

name="email"

value={values.email}

onChange={handleChange}

placeholder="Email"

/>

</div>

<div className='col-6'>

{errors.email && <span style={{"color":"red"}}>{errors.email}</span>}

</div>

</div>

<div className='row'>

<div className='col-4 offset-2'>

<input

type="password"

name="password"

value={values.password}

onChange={handleChange}

placeholder="Password"

/>

</div>

<div className='col-6'>

{errors.password && <span style={{"color":"red"}}>{errors.password}</span>}

</div>

</div>

<div className='row'>

<div className='col-4 offset-2'>

<button type="submit" disabled={isSubmitting} className='btn btn-success'>

{isSubmitting ? 'Submitting...' : 'Submit'}

</button>

</div>

</div>

</form>

</div>

);

};

export default LoginPage;

**Step 4: Importing user defined functional component in App.js entry file**

4.1 Now inside App.js file we need to import our user defined component ie LoginPage

**App.js file**

import logo from './logo.svg';

import './App.css';

import LoginPage from './LoginPage';

function App() {

return (

<div className='container'>

<h3>Custom Hook for Form Validation</h3>

**<LoginPage></LoginPage>**

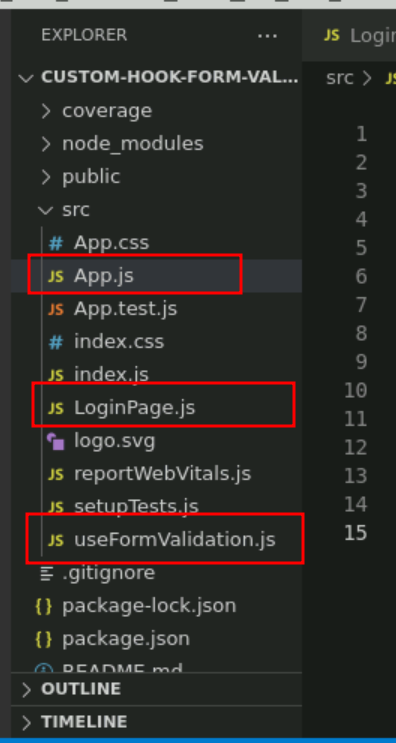
</div>

);

}

export default App;

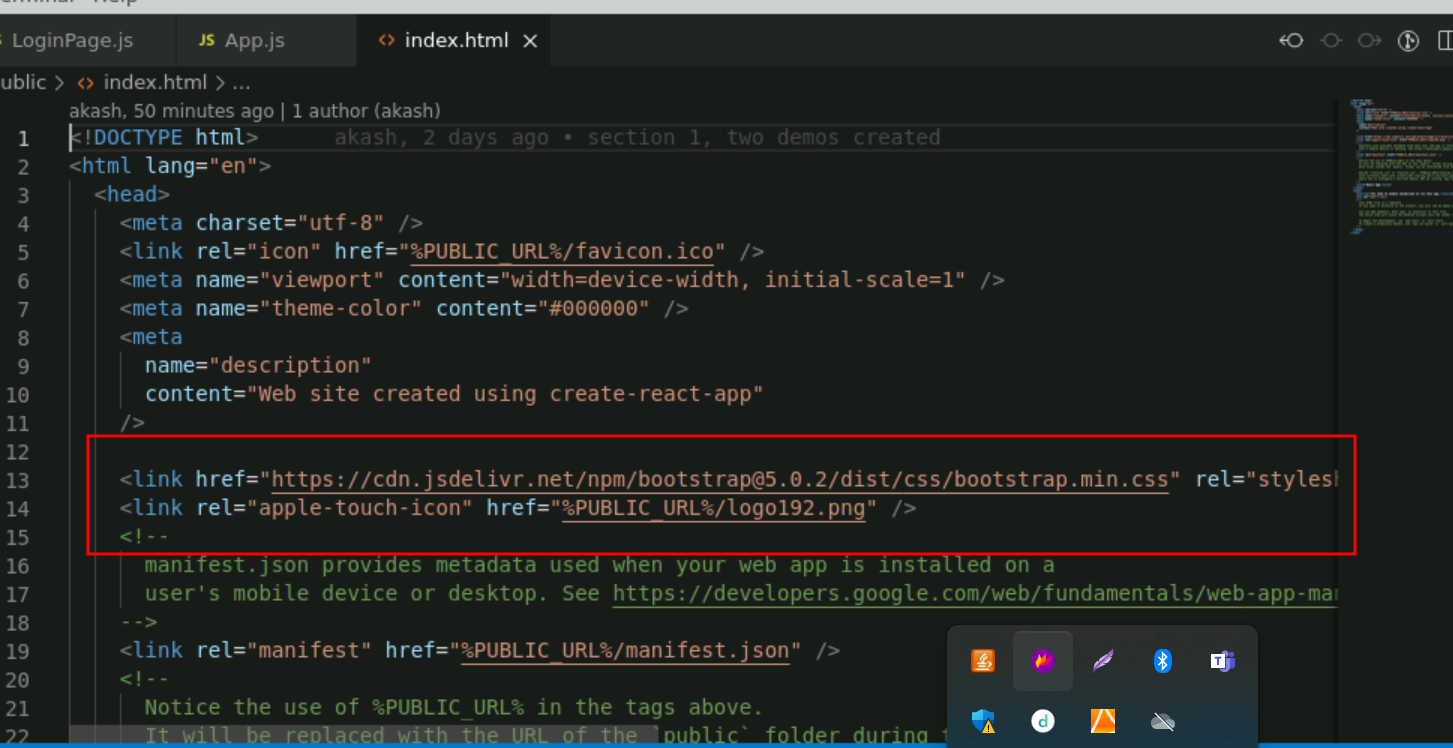
4.2 After all files created in project your project looks files below



**Step 5 :** Adding bootstrap features in index.html file

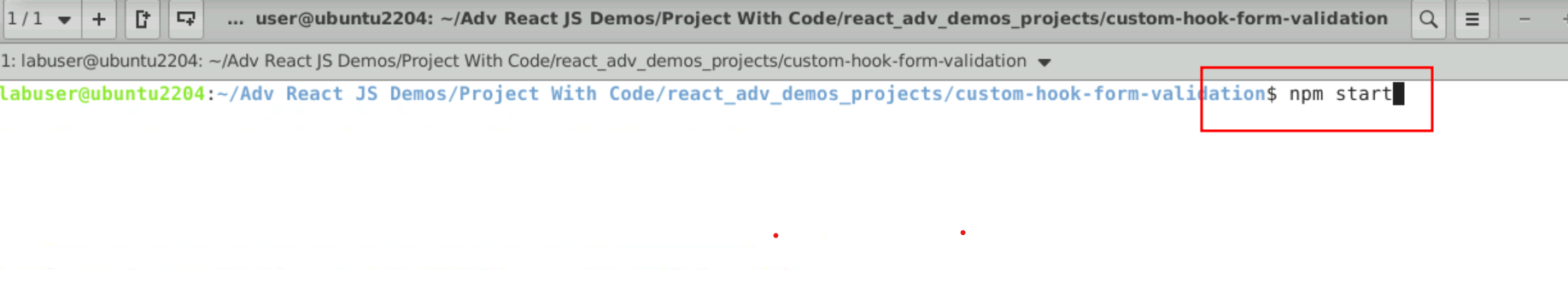
5.1 open the index.html file and add the CDN URL for bootstrap.

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

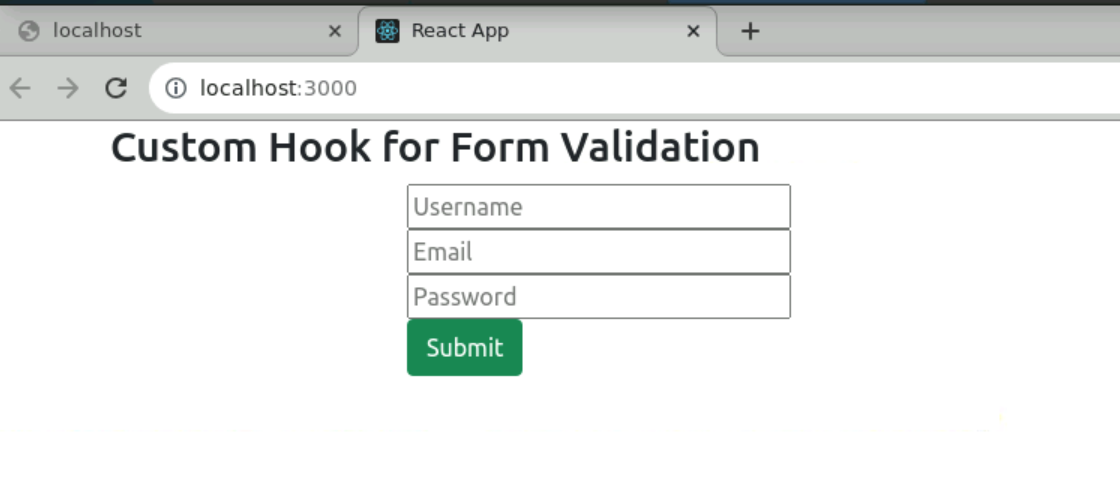


**Step 5 :** Adding bootstrap features in index.html file

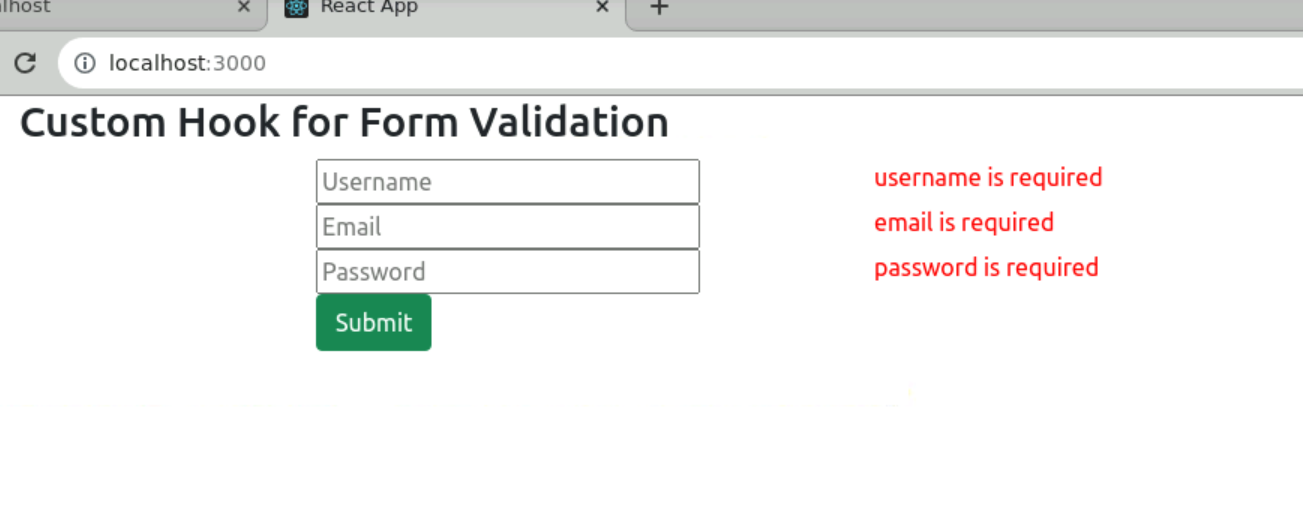
5.1 Now terminal window run the command as **npm start (**Make sure terminal open inside a project).



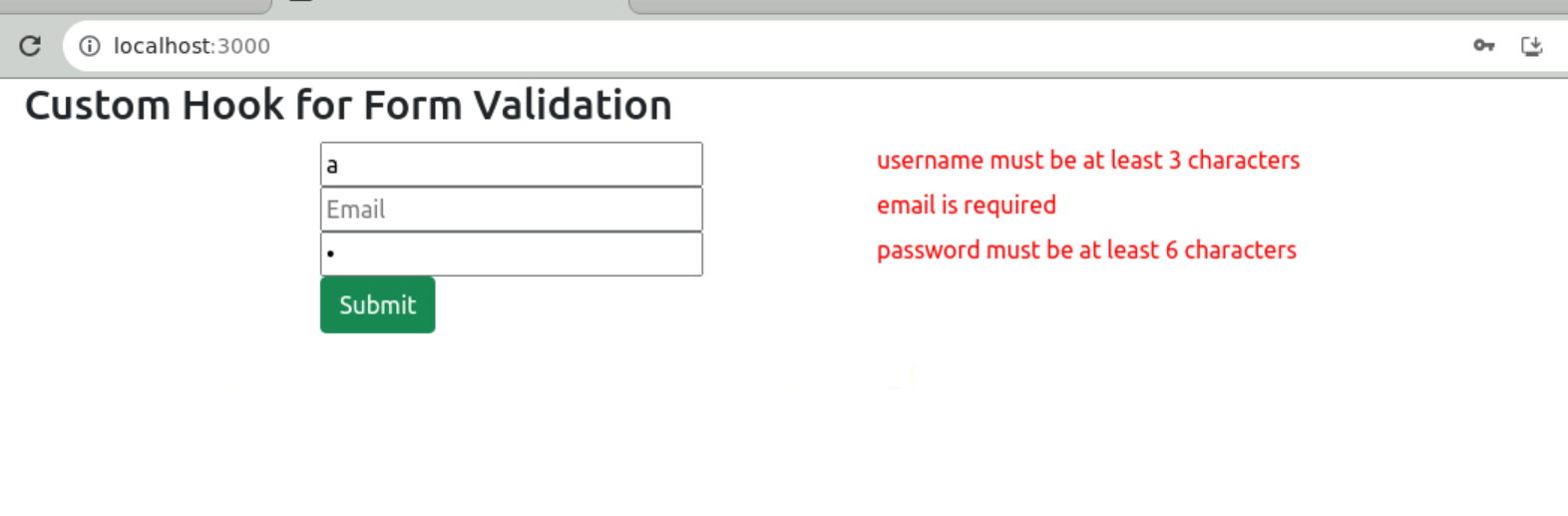
5.2 Now you project run on default browser with url as [**http://localhost:3000**](http://localhost:3000)



5.3 Now you can click on button without writing username, email or password. Then you can see the validation error.



5.4 Now username or password with 1 character. You can see the minimum length validation.



5.5 after written all valid information you can submit. And those information you can see in browser console.

