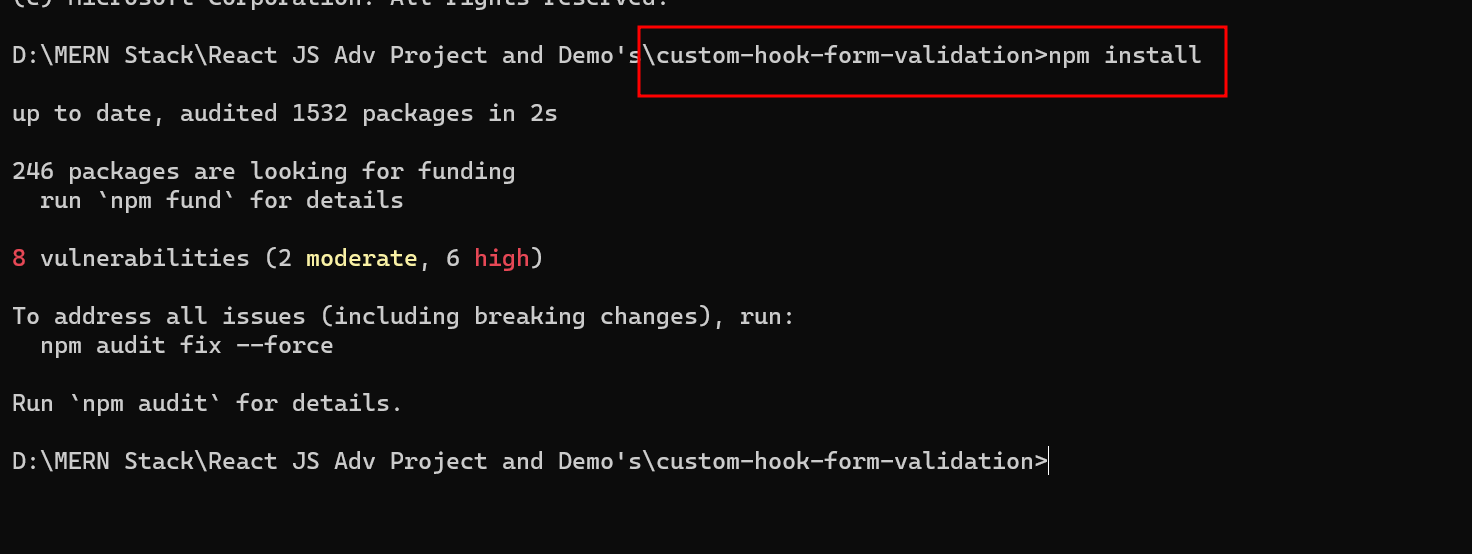
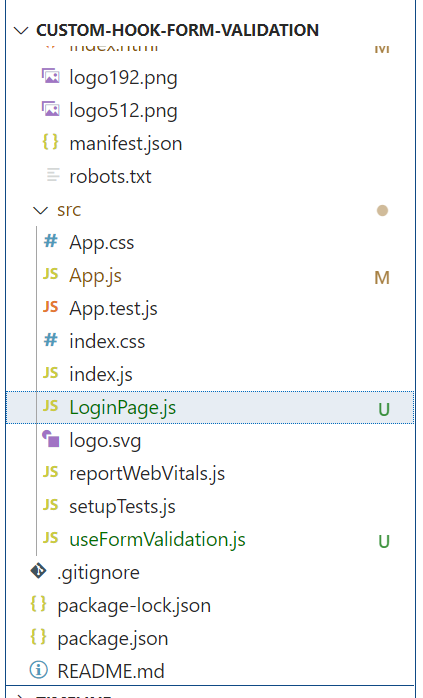
**Creating custom hook to do react form validation**

1. Node JS and React JS CLI must be installed.
2. Open the terminal or command prompt inside project ie **simple-custom-hook**
3. Then run npm install to download the required dependencies.



1. Then open the project in VC Code.



1. Now create **userFormValidation.js** file which contains the logic to create Custom hook for form validation

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;

1. Now we need to use this useFormValidation custom hook in our any component depending upon our requirements.

So now we will create the **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;

1. Now we need to import the LoginPage component in 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;

1. In Index.js file please add the bootstrap CDN link

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

**index.html**

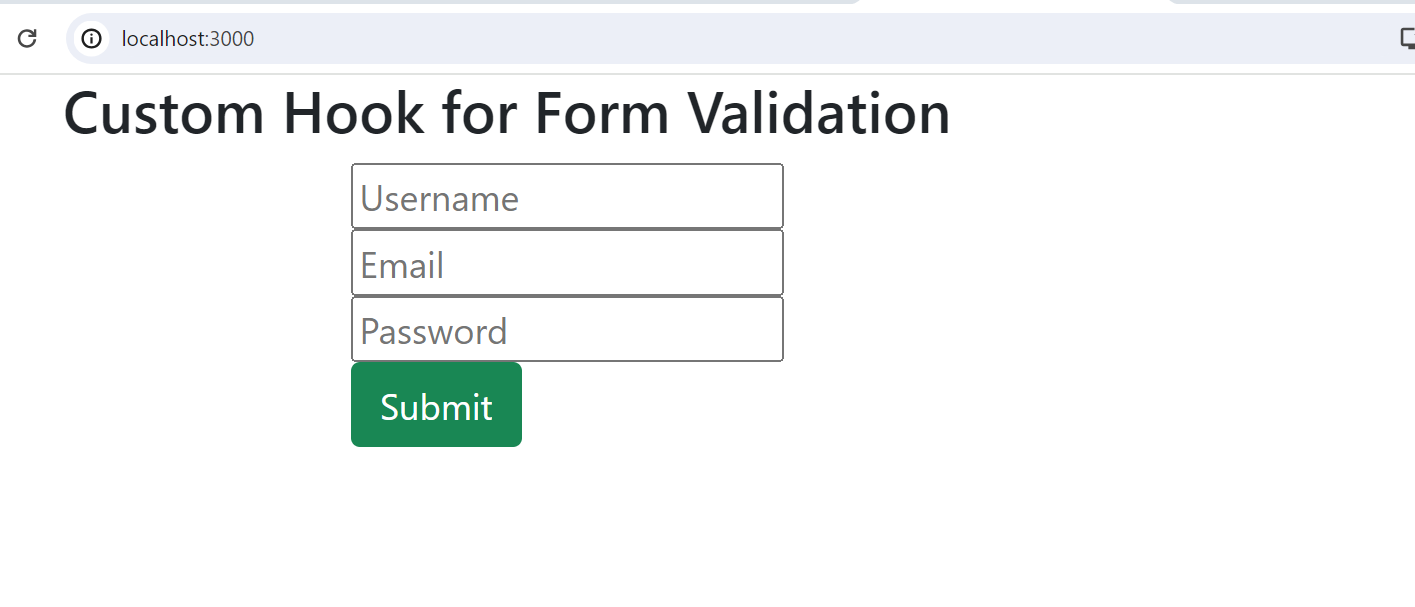


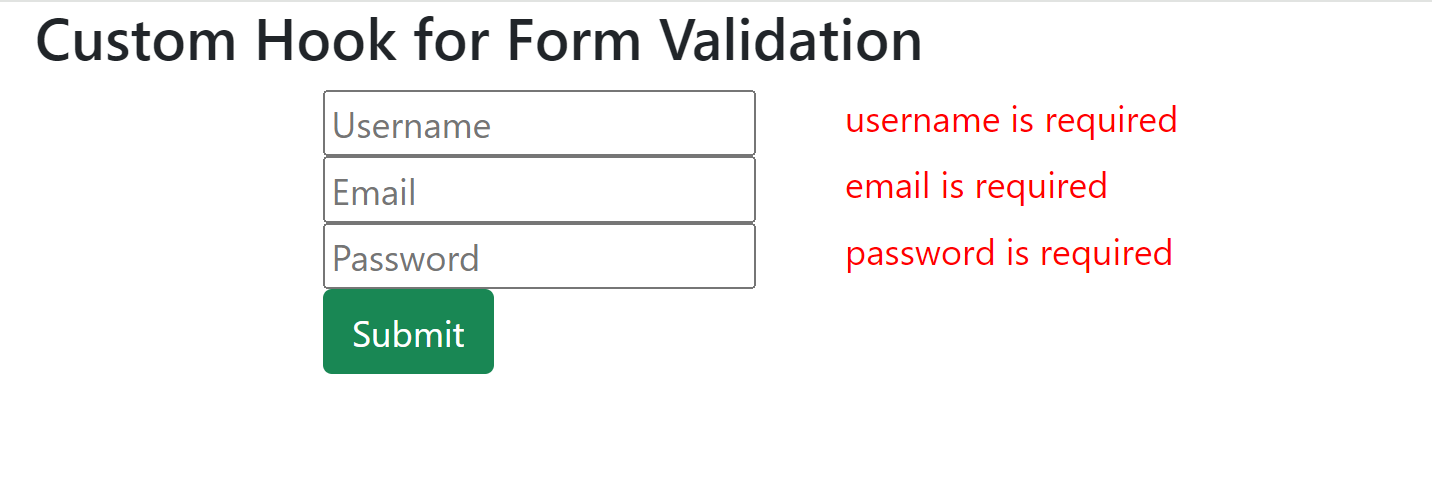
1. Now we need to run the application using the command as **npm start** and view the application in browser.



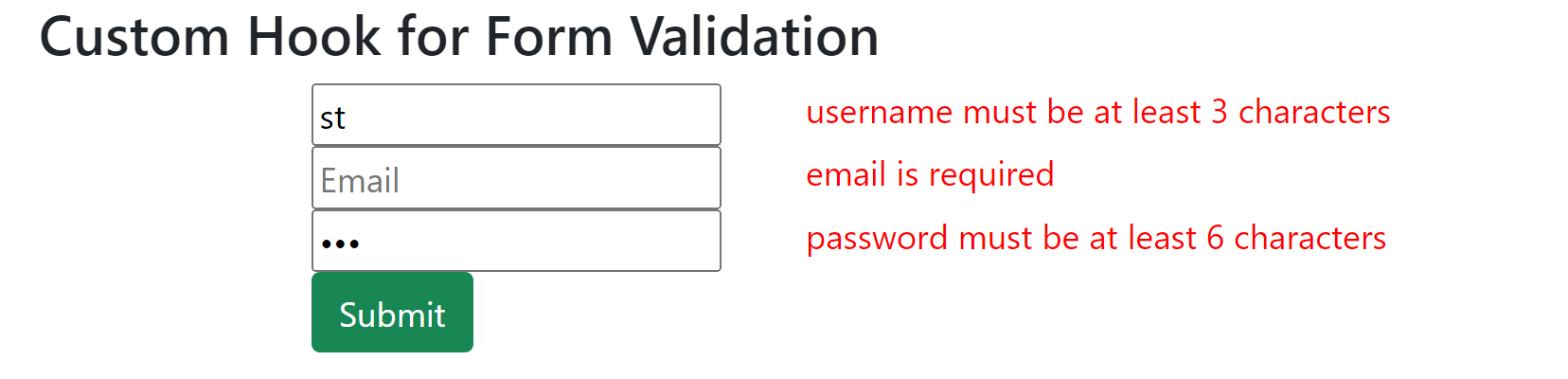
Now you can see the application in browser.

1. Now without writing username, emailed or password you can check the validation rules.





After filled partial information as



1. After filled the proper information you can see the output in browser console.

