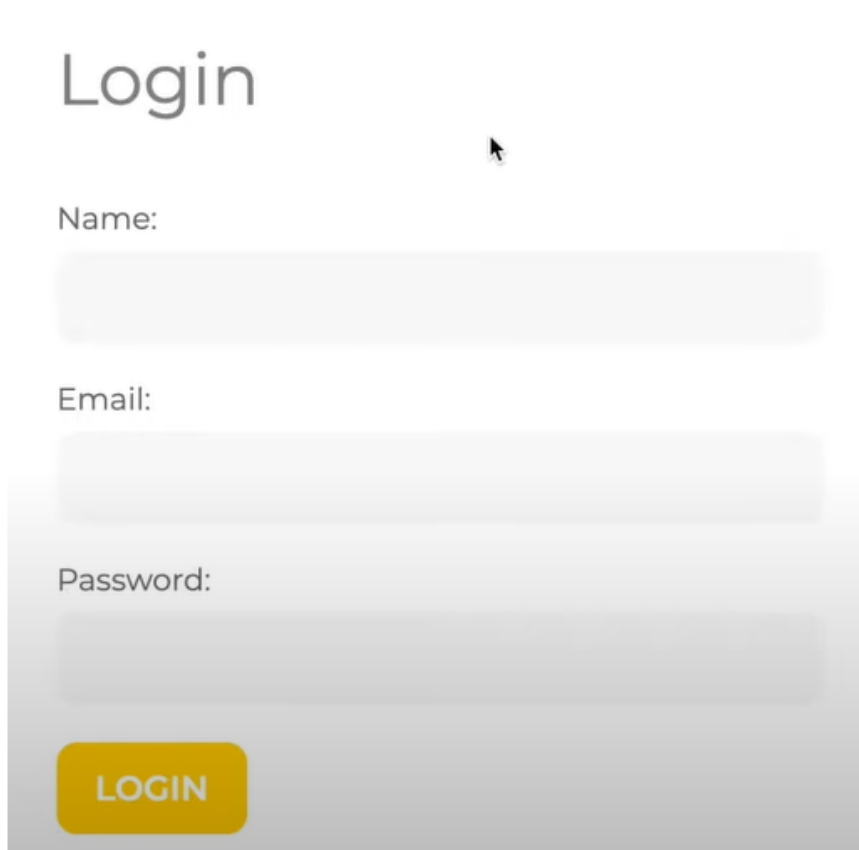


Ex. No: 7	React - Real Time Form Validation, and Submission Validation

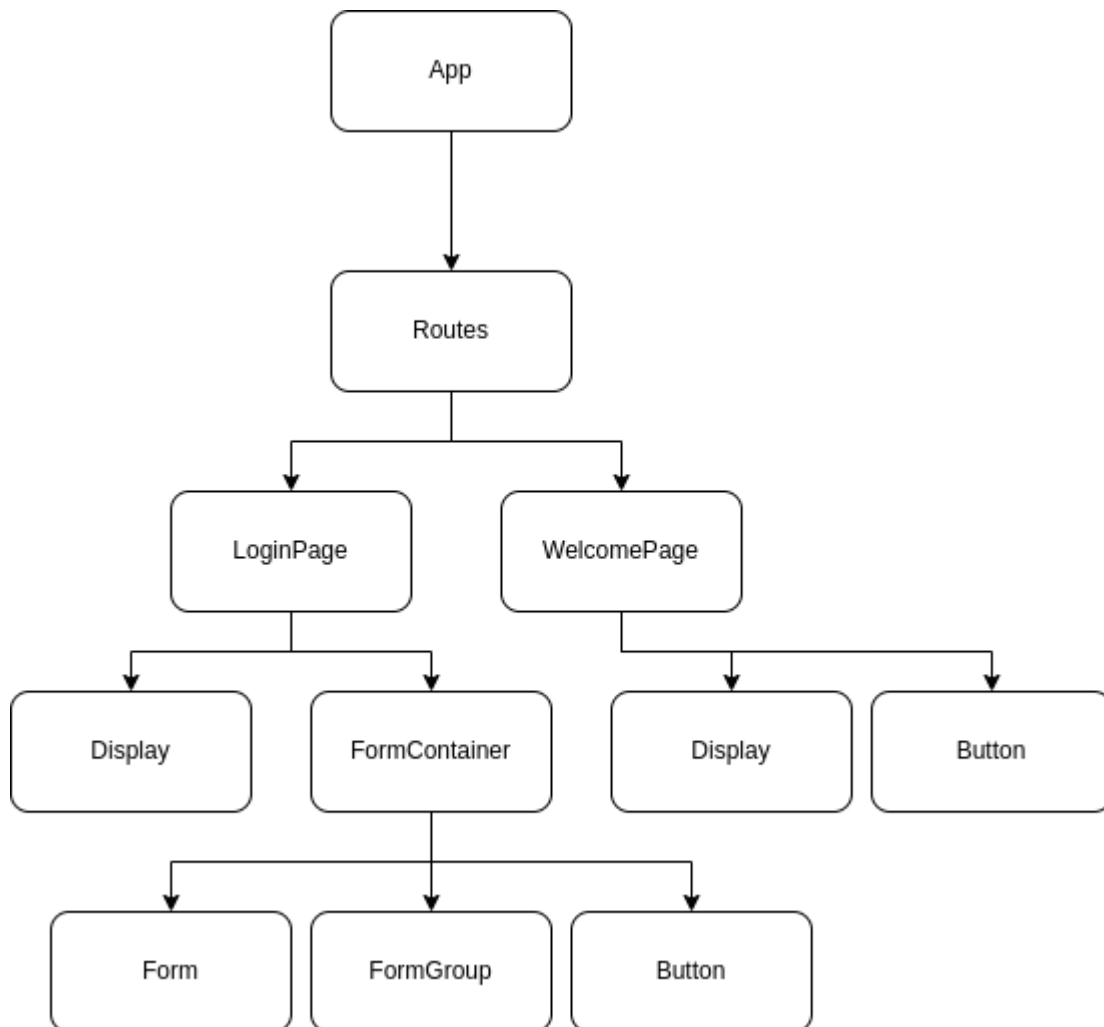
Aim: Implement a react app as below, (K4)

Mockup:



A login form mockup with a light gray background. At the top, the word "Login" is displayed in a large, dark gray font. Below it, there are three input fields: "Name:", "Email:", and "Password:". Each input field is a light gray rectangle with rounded corners. At the bottom left, there is a yellow button with the word "LOGIN" in white capital letters.

Component Hierarchy Diagram :



Program:

App.jsx

```
import './App.css';
import Login from './pages/login.jsx';
import Home from './pages/home.jsx';
import { BrowserRouter as Router ,Routes ,Route } from "react-router-dom";

function App() {
```

```

return (
  <div>
    <Router>
      <Routes>
        <Route path = "/" element = {<Login />} />
        <Route path = "/home" element = {<Home />} />
      </Routes>
    </Router>
  </div>
);
}

export default App;

```

components.jsx

```

export function Button({name, color, onClick}){
  return(
    <button
      name = {name}
      style = {{backgroundColor: color}}
      onClick= {onClick}
    >{name}</button>
  );
}

export function Input({name, onChange}){
  const type = name=="Password" ? "password" : "text";
  return(
    <label>
      {name} : <input name={name} type={type} onChange={onChange}
      placeholder={"Enter "+name} required />
    </label>
  );
}

export function Message({text}){
  return(
    <>
      {text.map((element) => (

```

```

        <label style={{color: element[0] ? "green" : "red"}}>
            {element[0] ? element[2] : element[1]}
        </label>
    )))
</>
);
}

```

./pages/home.jsx

```

export function Button({name, color, onClick}){
    return(
        <button
            name = {name}
            style = {{backgroundColor: color}}
            onClick= {onClick}
        >{name}</button>
    );
}

export function Input({name, onChange}){
    const type = name=="Password" ? "password" : "text";
    return(
        <label>
            {name} : <input name={name} type={type} onChange={onChange}
            placeholder={"Enter "+name} required />
        </label>
    );
}

export function Message({text}){
    return(
        <>
            {text.map((element) => (
                <label style={{color: element[0] ? "green" : "red"}}>
                    {element[0] ? element[2] : element[1]}
                </label>
            )))}
        </>
    );
}

```

./pages/login.jsx

```
import {Button, Input, Message} from '../components.jsx';
import { Link } from 'react-router-dom';
import { useState } from 'react';

const userDatabase = [
  {name:'Tom', email:'tomsayer.in', pass:'S2e'},
  {name:'Kira', email:'kira@L.in', pass:'L1g'},
  {name:'Harvey', email:'harvey@gmail.com', pass:'Y5f'},
];

export default function Login(){
  const [name, setName] = useState('');
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const reg1 = new RegExp("(?=.*[a-z])");
  const reg2 = new RegExp("(?=.*[A-Z])");
  const reg3 = new RegExp("(?=.*[0-9])");

  function success(){
    if(name && email && reg1.test(password) && reg2.test(password) &&
    reg3.test(password)){
      if(userDatabase.find((e) => e.name==name && e.email==email &&
      e.pass==password)) return true;
    }
    return false;
  }

  return(
    <form>
      <h1>Login</h1>
      <Input name="Name" onChange={(e) => setName(e.target.value)}/>

      <Input name="Email" onChange={(e) => setEmail(e.target.value)}/>

      <Input name="Password" onChange={(e) => setPassword(e.target.value)}/>
    </form>
  )
}
```

```

    <Link to={success() ? "/home" : "/"}>
      <Button name="Login" color="green" onClick={(e) =>
        {if(!success())alert("No such records found!")}}/>
    </Link>
  </form>
);
}

```

main.jsx

```

import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App.jsx'
import './index.css'

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
)

```

App.css

```

#root {
  max-width: 1280px;
  margin: 0 auto;
  padding: 2rem;
  text-align: center;
}
form{
  display: flex;
  flex-direction: column;
}
input{
  margin: 10px;
  font-size: 18px;
}
button{
  padding: 5px;
  margin: 15px;
}

```

```
width: 150px;
align-self: center;
font-size: 19px;
color: #fff;
}
```

index.css

```
:root {
  font-family: Inter, system-ui, Avenir, Helvetica, Arial, sans-serif;
  line-height: 1.5;
  font-weight: 400;

  color-scheme: light dark;
  color: rgba(255, 255, 255, 0.87);
  background-color: #242424;

  font-synthesis: none;
  text-rendering: optimizelegibility;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  -webkit-text-size-adjust: 100%;
}

a {
  font-weight: 500;
  color: #646cff;
  text-decoration: inherit;
}
a:hover {
  color: #535bf2;
}

body {
  margin: 0;
  display: flex;
  place-items: center;
  min-width: 320px;
  min-height: 100vh;
}
```

```
h1 {
  font-size: 3.2em;
  line-height: 1.1;
}

button {
  border-radius: 8px;
  border: 1px solid transparent;
  padding: 0.6em 1.2em;
  font-size: 1em;
  font-weight: 500;
  font-family: inherit;
  background-color: #1a1a1a;
  cursor: pointer;
  transition: border-color 0.25s;
}

button:hover {
  border-color: #646cff;
}

button:focus,
button:focus-visible {
  outline: 4px auto -webkit-focus-ring-color;
}

@media (prefers-color-scheme: light) {
  :root {
    color: #213547;
    background-color: #ffffff;
  }
  a:hover {
    color: #747bff;
  }
  button {
    background-color: #f9f9f9;
  }
}
```

Output:

Login

Name :

Email :

Password :

Login

Hello User

Logout

Result: Thus, form validation in react was studied and implemented.