**ReactJS Hands-On Lab Report: Mail Register App with Validation**

**Objectives**

* Explain React Forms validation
* Identify the differences between React Form and HTML Form
* Explain about controlled components
* Identify various React Form input controls
* Explain how to handle React Forms
* Explain about submitting forms in React

**In this hands-on lab, you will learn how to:**

* Implement React forms validation
* Use various input controls like textbox, button and textarea

**Prerequisites**

* Node.js
* NPM
* Visual Studio Code

**Step-by-Step Instructions**

**1. Create the React App**

npx create-react-app mailregisterapp

cd mailregisterapp

code .

**2. Clean the Project**

Delete the following files from src/:

* App.test.js
* logo.svg
* reportWebVitals.js
* setupTests.js

**3. Modify App.js**

import React from 'react';

import Register from './Register';

function App() {

return (

<div className="App">

<h1>Mail Register App</h1>

<Register />

</div>

);

}

export default App;

**4. Create Register.js**

Create src/Register.js and paste the following code:

import React, { useState } from 'react';

const Register = () => {

const [name, setName] = useState('');

const [email, setEmail] = useState('');

const [password, setPassword] = useState('');

const [error, setError] = useState('');

const handleSubmit = (e) => {

e.preventDefault();

if (name.length < 5) {

setError('Name should have at least 5 characters');

return;

}

if (!email.includes('@') || !email.includes('.')) {

setError('Email should contain @ and .');

return;

}

if (password.length < 8) {

setError('Password should have at least 8 characters');

return;

}

alert('Form submitted successfully!');

setName('');

setEmail('');

setPassword('');

setError('');

};

return (

<form onSubmit={handleSubmit}>

<div>

<label>Name:</label><br />

<input

type="text"

value={name}

onChange={(e) => setName(e.target.value)}

required

/>

</div>

<br />

<div>

<label>Email:</label><br />

<input

type="email"

value={email}

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

required

/>

</div>

<br />

<div>

<label>Password:</label><br />

<input

type="password"

value={password}

onChange={(e) => setPassword(e.target.value)}

required

/>

</div>

<br />

<button type="submit">Register</button>

{error && (

<p style={{ color: 'red' }}>{error}</p>

)}

</form>

);

};

export default Register;

**5. Run the Application**

npm start

**Concepts Explained**

* **Controlled Components**: Each input field is managed using useState.
* **Form Handling**: Uses onChange for updates, onSubmit for validation logic.
* **Validation**: Handled inside handleSubmit() with custom logic.
* **HTML vs React Form**: HTML forms submit directly; React uses state and JS to control behavior.





