**Module 10 – Single Page Application framework – React**

**16.ReactJS-HOL**

**1. React Forms Validation**

React form validation is the process of checking input values before submitting a form. It ensures the data entered by users is correct, complete, and safe. Validation can be done:

* While typing (real-time)
* When the form is submitted

**2. React Form vs HTML Form**

| **Feature** | **React Form** | **HTML Form** |
| --- | --- | --- |
| Control of data | Controlled via state | Browser handles it directly |
| Validation | Programmatically using JS | Basic validation via attributes |
| Behavior customization | Fully customizable | Limited customization |
| Dynamic input handling | Easily manageable | Not easy without JS |

**3. Controlled Components**

A controlled component in React is one where form data is handled by the component’s state. This gives more control over the input and enables validation, formatting, and dynamic behavior.

**4. React Form Input Controls**

* <input type="text"> for name/email
* <input type="password"> for passwords
* <textarea> for long text
* <button> to submit

**5. Handling React Forms**

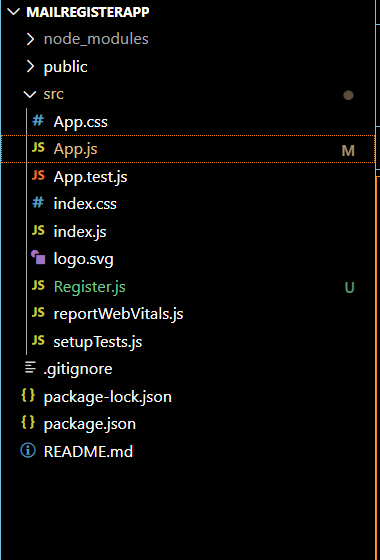
React uses onChange to update the form's state in real time. This way, you always have the latest values in state, and you can validate them or modify them as needed.

**6. Submitting Forms in React**

When the form is submitted using onSubmit, you can:

* Prevent page reload with e.preventDefault()
* Check validations
* Show success/failure message
* Process data (send to server or display)

**Structure:**



**Register.js:**

import React, { useState } from 'react';

function Register() {

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

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

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

  const validate = () => {

    if (name.trim().length < 5) {

      alert("Full Name must be 5 characters long!");

      return false;

    }

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

      alert("Email is not valid!");

      return false;

    }

    if (password.length < 8) {

      alert("Password must be at least 8 characters long!");

      return false;

    }

    return true;

  };

  const handleSubmit = (e) => {

    e.preventDefault();

    if (validate()) {

      alert(`Thanks ${name}, your registration is successful!`);

      // Optional: clear the form after successful submission

      setName('');

      setEmail('');

      setPassword('');

    }

  };

  return (

    <div style={{ textAlign: 'center', marginTop: '50px' }}>

      <h2 style={{ color: 'red', fontWeight: 'bold' }}>Register Here!!!</h2>

      <form onSubmit={handleSubmit}>

        <div>

          <label>Name: </label>

          <input

            type="text"

            value={name}

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

          />

        </div>

        <br />

        <div>

          <label>Email: </label>

          <input

            type="text"

            value={email}

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

          />

        </div>

        <br />

        <div>

          <label>Password: </label>

          <input

            type="password"

            value={password}

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

          />

        </div>

        <br />

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

      </form>

    </div>

  );

}

export default Register;

**App.js:**

import React from 'react';

import Register from './Register';

function App() {

  return (

    <div>

      <Register />

    </div>

  );

}

export default App;

**Output:**

