**WEEK - 7(Additional HandsOn)**

**React**

**16. ReactJS-HOL**

**Objectives**

**• Explain React Forms validation**

Form validation in React ensures that users enter valid input before submitting.

**🔹 Types of validation:**

* **Client-side (manual)**: You write your own logic using if statements
* **Library-based**: Using tools like Formik, Yup, or React Hook Form

**Basic Validation Example:**

const handleSubmit = (e) => {

e.preventDefault();

if (email === '') {

alert("Email is required");

} else {

alert("Submitted: " + email);

}

};

**• Identify the differences between React Form and HTML Form**

| **Feature** | **HTML Form** | **React Form** |
| --- | --- | --- |
| Form Control | Browser-controlled | State-controlled (useState, handlers) |
| Validation | required, pattern attributes | Custom logic or libraries |
| Submission | Refreshes page | Prevents refresh using e.preventDefault() |
| Dynamic Fields | Difficult | Easy with state and JSX |
| Reusability | Low | High (reusable components) |

**• Explain about controlled components**

As explained before, a controlled component:

* Is an input element whose **value is tied to React state**
* Changes are handled through onChange events

**🔹 Example:**

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

<input type="text" value={name} onChange={(e) => setName(e.target.value)} />

**• Identify various React Form input controls**

| **Input Type** | **JSX Syntax Example** |
| --- | --- |
| Text | <input type="text" value={value} onChange={...} /> |
| Textarea | <textarea value={value} onChange={...} /> |
| Select | <select value={value} onChange={...}> <option>...</option> </select> |
| Checkbox | <input type="checkbox" checked={checked} onChange={...} /> |
| Radio | <input type="radio" value="..." checked={...} onChange={...} /> |
| Email | <input type="email" value={email} onChange={...} /> |

**Explain how to handle React Forms**

**🔹 Basic Steps:**

 Set up state with useState

 Bind state to input's value (or checked)

 Use onChange to update state

 Use onSubmit on <form> with a handleSubmit function

**Example:**

function FormExample() {

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

const handleChange = (e) => setEmail(e.target.value);

const handleSubmit = (e) => {

e.preventDefault();

alert("Email: " + email);

};

return (

<form onSubmit={handleSubmit}>

<input type="email" value={email} onChange={handleChange} />

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

</form>

);

}

**• Explain about submitting forms in React**

When you submit a React form:

* You use onSubmit={handleSubmit}
* You call e.preventDefault() to stop page reload
* You can then validate, send to API, or show alerts

**Example:**

const handleSubmit = (e) => {

e.preventDefault();

if (name === '') {

alert("Name is required!");

} else {

console.log("Submitting form:", name);

}

};

**CODE:**

**src/RegisterForm.js:**

import React, { useState } from 'react';

function RegisterForm() {

  const [fullName, setFullName] = useState('');

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

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

  // Basic email validation pattern

  function isValidEmail(email) {

    return /^\S+@\S+\.\S+$/.test(email);

  }

  const handleSubmit = (e) => {

    e.preventDefault();

    if (fullName.trim().length < 5) {

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

      return;

    }

    if (!isValidEmail(email)) {

      window.alert("Email is not valid!");

      return;

    }

    if (password.length < 8) {

      window.alert("Password must be 8 characters long!");

      return;

    }

    window.alert("Registration Successful!");

    setFullName('');

    setEmail('');

    setPassword('');

  };

  return (

    <div

      style={{

        height: "100vh",

        width: "100vw",

        display: "flex",

        alignItems: "center",

        justifyContent: "center",

      }}

    >

      <div>

        <h1

          style={{

            color: "red",

            fontWeight: "bold",

            fontSize: "2.5rem",

            textAlign: "center",

            marginBottom: "30px"

          }}

        >

          Register Here!!!

        </h1>

        <form onSubmit={handleSubmit} style={{ minWidth: 320 }}>

          <div style={{ marginBottom: "12px" }}>

            <label style={{ display: "inline-block", width: 70 }}>Name:</label>

            <input

              type="text"

              value={fullName}

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

              style={{ fontSize: "1rem", padding: "5px", width: 180 }}

            />

          </div>

          <div style={{ marginBottom: "12px" }}>

            <label style={{ display: "inline-block", width: 70 }}>Email:</label>

            <input

              type="text"

              value={email}

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

              style={{ fontSize: "1rem", padding: "5px", width: 180 }}

            />

          </div>

          <div style={{ marginBottom: "16px" }}>

            <label style={{ display: "inline-block", width: 70 }}>Password:</label>

            <input

              type="password"

              value={password}

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

              style={{ fontSize: "1rem", padding: "5px", width: 180 }}

            />

          </div>

          <div style={{ textAlign: "center" }}>

            <button

              type="submit"

              style={{

                fontSize: "1rem",

                padding: "5px 18px",

                cursor: "pointer"

              }}

            >Submit</button>

          </div>

        </form>

      </div>

    </div>

  );

}

export default RegisterForm;

**src/App.js:**

import React from 'react';

import RegisterForm from './RegisterForm';

function App() {

  return <RegisterForm />;

}

export default App;

**OUTPUT:**

