## **Objectives**

* Explain about React forms
* Define controlled components
* Explain about various input controls
* Explain about handling forms
* Explain about submitting forms

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

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

## **Prerequisites**

The following is required to complete this hands-on lab:

* Node.js
* NPM
* Visual Studio Code

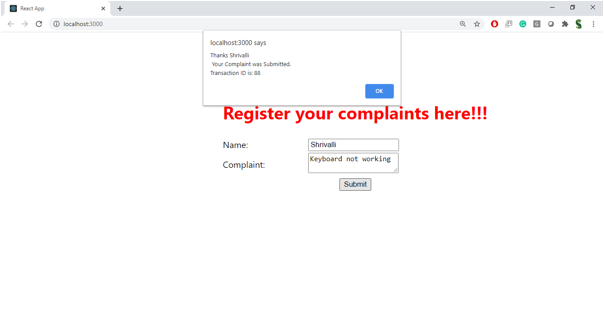
## **Notes**

Estimated time to complete this lab: **60 minutes.**

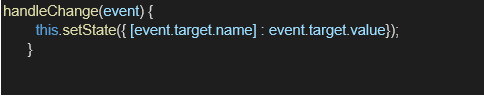
Create a React App named “ticketraisingapp” which will help to raise a complaint and get it resolved.

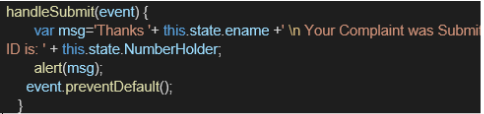
Create a component named “ComplaintRegister” with a form containing a textbox to enter the employee name and a textarea to enter the complaint. Use “handleSubmit” event of the button to submit the complaint and generate a Reference number for further follow ups in the alert box.

**Output Expected:**



**Hint:**





**SOURCE CODE**

**File: ComplaintRegister.js**

import React, { useState } from 'react';

function ComplaintRegister() {

const [ename, setEname] = useState('');

const [complaint, setComplaint] = useState('');

const [referenceNumber, setReferenceNumber] = useState('');

const handleChange = (event) => {

const { name, value } = event.target;

if (name === 'ename') {

setEname(value);

} else if (name === 'complaint') {

setComplaint(value);

}

};

const handleSubmit = (event) => {

event.preventDefault();

// Generate a reference number

const reference = Math.floor(Math.random() \* 1000000);

setReferenceNumber(reference);

const msg = `Thanks ${ename},\nYour Complaint was submitted successfully. \nReference ID: ${reference}`;

alert(msg);

// Clear the form

setEname('');

setComplaint('');

};

return (

<div className="complaint-container">

<h1 className="heading">Register Your Complaints Here!!</h1>

<form onSubmit={handleSubmit} className="complaint-form">

<div className="form-field">

<label htmlFor="ename">Employee Name:</label>

<input

type="text"

id="ename"

name="ename"

value={ename}

onChange={handleChange}

required

placeholder="Enter your name"

/>

</div>

<div className="form-field">

<label htmlFor="complaint">Complaint:</label>

<textarea

id="complaint"

name="complaint"

value={complaint}

onChange={handleChange}

required

placeholder="Enter your complaint"

></textarea>

</div>

<div className="form-field">

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

</div>

</form>

{referenceNumber && (

<div className="reference">

<p>Reference Number: {referenceNumber}</p>

</div>

)}

</div>

);

}

export default ComplaintRegister;

**File: App.js**

import React, { useState } from 'react';

function ComplaintRegister() {

const [ename, setEname] = useState('');

const [complaint, setComplaint] = useState('');

const [referenceNumber, setReferenceNumber] = useState('');

const handleChange = (event) => {

const { name, value } = event.target;

if (name === 'ename') {

setEname(value);

} else if (name === 'complaint') {

setComplaint(value);

}

};

const handleSubmit = (event) => {

event.preventDefault();

// Generate a reference number

const reference = Math.floor(Math.random() \* 1000000);

setReferenceNumber(reference);

const msg = `Thanks ${ename},\nYour Complaint was submitted successfully. \nReference ID: ${reference}`;

alert(msg);

// Clear the form

setEname('');

setComplaint('');

};

return (

<div className="complaint-container">

<h1 className="heading">Register Your Complaints Here!!</h1>

<form onSubmit={handleSubmit}>

<div className="form-field">

<label htmlFor="ename">Employee Name:</label>

<input

type="text"

id="ename"

name="ename"

value={ename}

onChange={handleChange}

required

placeholder="Enter your name"

/>

</div>

<div className="form-field">

<label htmlFor="complaint">Complaint:</label>

<textarea

id="complaint"

name="complaint"

value={complaint}

onChange={handleChange}

required

placeholder="Enter your complaint"

></textarea>

</div>

<div className="form-field">

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

</div>

</form>

{referenceNumber && (

<div className="reference">

<p>Reference Number: {referenceNumber}</p>

</div>

)}

</div>

);

}

function App() {

return (

<div className="App">

<ComplaintRegister />

</div>

);

}

export default App;

// Adding minimal styles within the same file

const style = document.createElement("style");

style.innerHTML = `

/\* Centering the content with a little top space \*/

.complaint-container {

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

height: 100vh;

padding-top: 50px; /\* Space from top \*/

background-color: #f4f4f4;

}

/\* Red heading style \*/

.heading {

color: red; /\* Changed to red \*/

font-size: 2.5rem;

margin-bottom: 20px; /\* Space below the heading \*/

}

/\* No box format, just a simple layout \*/

form {

display: flex;

flex-direction: column;

width: 250px; /\* Narrower form \*/

}

/\* Spacing for each form field \*/

.form-field {

margin-bottom: 15px;

width: 100%;

}

.form-field label {

font-weight: bold;

}

.form-field input,

.form-field textarea {

width: 100%;

padding: 8px;

border: 1px solid #ccc;

border-radius: 4px;

font-size: 1rem;

}

/\* Red submit button \*/

.form-field button {

width: 100%;

padding: 10px;

background-color: red; /\* Changed to red \*/

border: none;

border-radius: 4px;

font-size: 1.2rem;

cursor: pointer;

color: white;

}

.form-field button:hover {

background-color: darkred; /\* Darker shade of red on hover \*/

}

.reference {

margin-top: 20px;

font-size: 1.1rem;

font-weight: bold;

}

`;

document.head.appendChild(style);

**OUTPUT:**

