## **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**

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

* Node.js
* NPM
* Visual Studio Code

## **Notes**

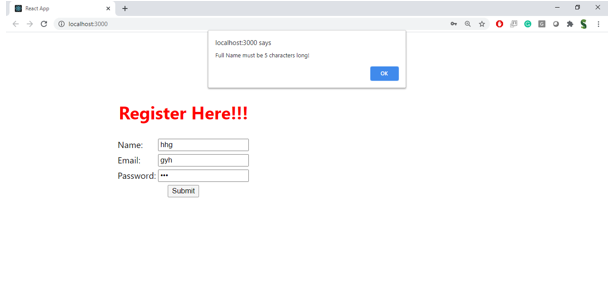
Estimated time to complete this lab: **60 minutes.**

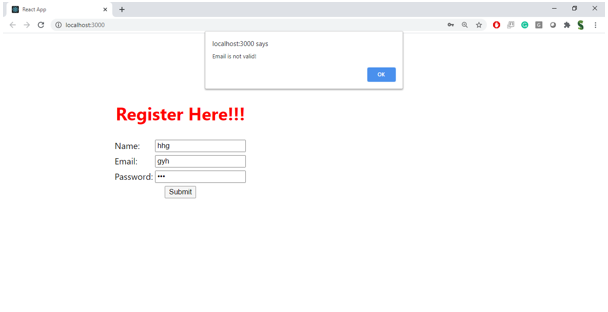
Create a React App named “mailregisterapp” which will have a component named “register.js”. Create a form which accepts the name, email and password and validate the fields as per the following:

1. Name should have atleast 5 characters
2. Email should have @ and .
3. Password should have atleast 8 characters.

Ensure that validations are implemented through eventhandle and eventsubmit of a form.

**Output Expected:**





**SOURCE CODE**

**File: Register.js**

import React, { useState } from 'react';

// Register Component

const Register = () => {

const [formData, setFormData] = useState({

fullName: '',

email: '',

password: '',

});

const [errors, setErrors] = useState({

fullName: '',

email: '',

password: '',

});

// Validate the form fields

const validateForm = () => {

let formErrors = { ...errors };

let isValid = true;

// Validate fullName

if (formData.fullName.length < 5) {

formErrors.fullName = 'Full Name must be 5 characters long!';

isValid = false;

} else {

formErrors.fullName = '';

}

// Validate email

const emailRegex = /^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;

if (!emailRegex.test(formData.email)) {

formErrors.email = 'Email is not valid!';

isValid = false;

} else {

formErrors.email = '';

}

// Validate password

if (formData.password.length < 8) {

formErrors.password = 'Password must be at least 8 characters long!';

isValid = false;

} else {

formErrors.password = '';

}

setErrors(formErrors);

return isValid;

};

// Handle the form input change

const handleChange = (e) => {

const { name, value } = e.target;

setFormData({

...formData,

[name]: value,

});

};

// Handle form submit

const handleSubmit = (e) => {

e.preventDefault();

if (validateForm()) {

alert('Form submitted successfully!');

// Reset form data after successful submission

setFormData({

fullName: '',

email: '',

password: '',

});

} else {

// Show validation error messages as alerts

if (errors.fullName) {

alert(errors.fullName);

}

if (errors.email) {

alert(errors.email);

}

if (errors.password) {

alert(errors.password);

}

}

};

return (

<div>

<h1>Register Here</h1>

<form onSubmit={handleSubmit}>

<div>

<label htmlFor="fullName">Name: </label>

<input

type="text"

id="fullName"

name="fullName"

value={formData.fullName}

onChange={handleChange}

required

/>

</div>

<div>

<label htmlFor="email">Email: </label>

<input

type="email"

id="email"

name="email"

value={formData.email}

onChange={handleChange}

required

/>

</div>

<div>

<label htmlFor="password">Password: </label>

<input

type="password"

id="password"

name="password"

value={formData.password}

onChange={handleChange}

required

/>

</div>

<div>

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

</div>

</form>

</div>

);

};

export default Register;

**File: App.js**

import React from 'react';

import './App.css';

import Register from './Register';

function App() {

return (

<div className="App">

<Register />

</div>

);

}

export default App;

**File: App.css**

.App {

font-family: Arial, sans-serif;

max-width: 500px;

margin: 0 auto;

padding: 20px;

margin-top: 125px;

}

form div {

margin-bottom: 15px;

}

label {

font-weight: bold;

}

input {

width: 100%;

padding: 8px;

margin-top: 5px;

font-size: 14px;

}

button {

padding: 10px 20px;

background-color: #4CAF50;

color: white;

border: none;

cursor: pointer;

}

button:hover {

background-color: #45a049;

}

span {

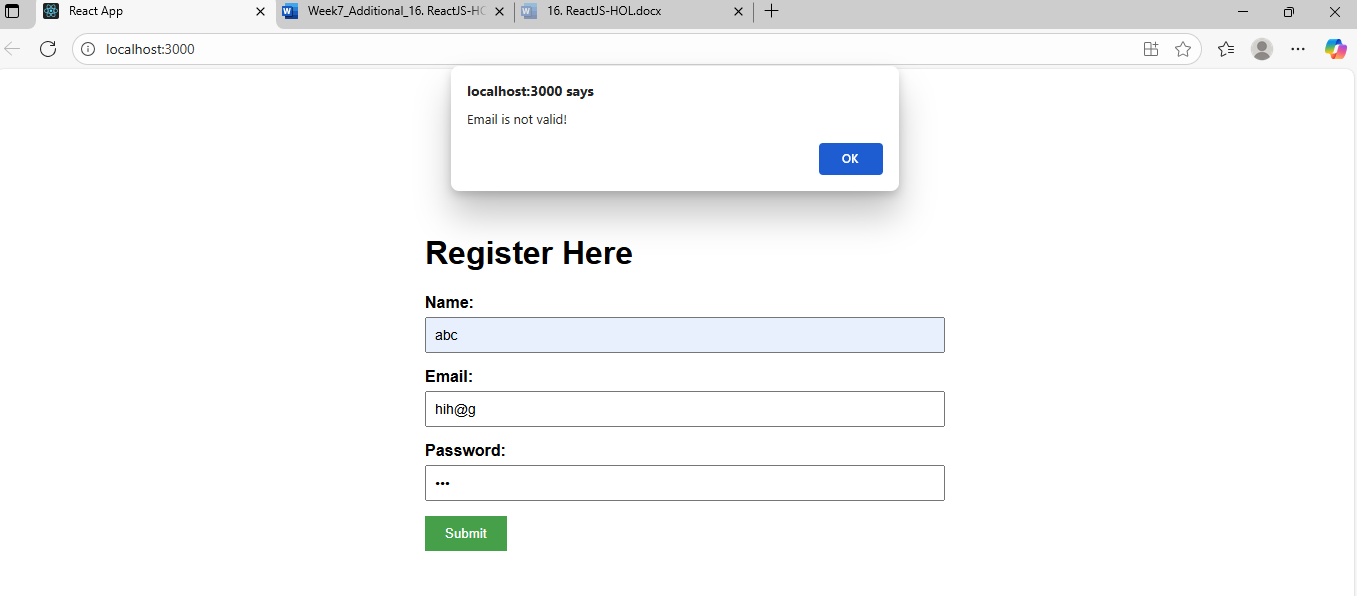
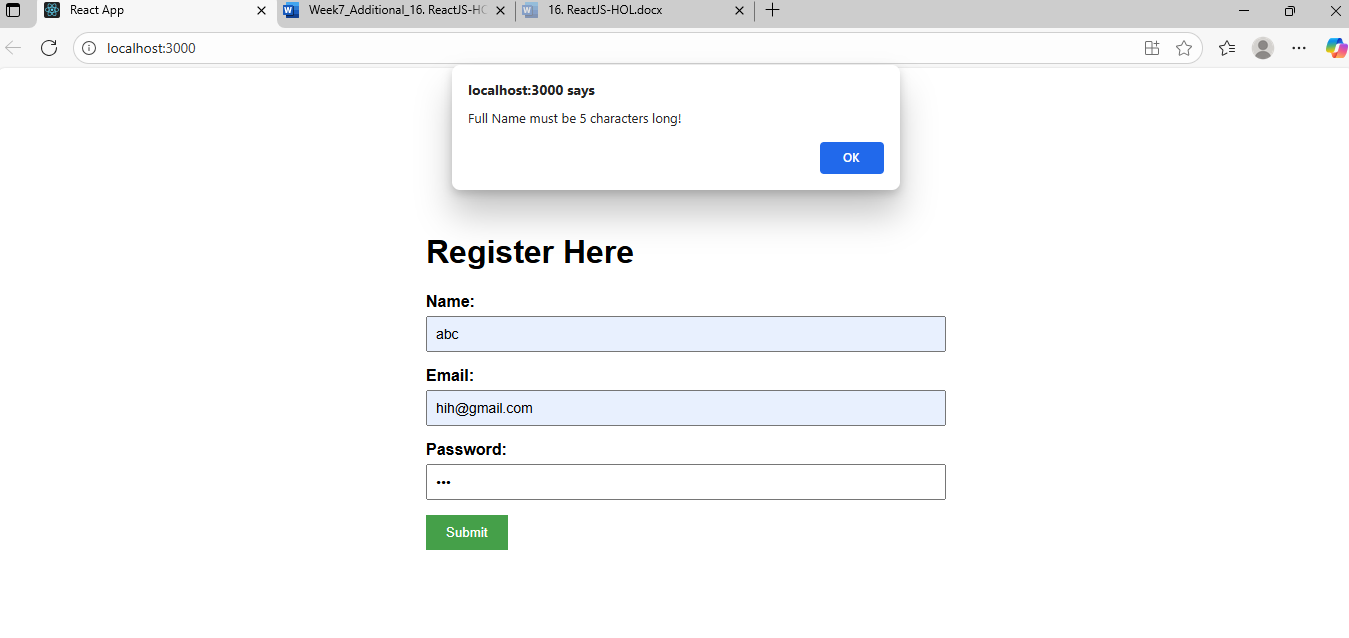
color: red;

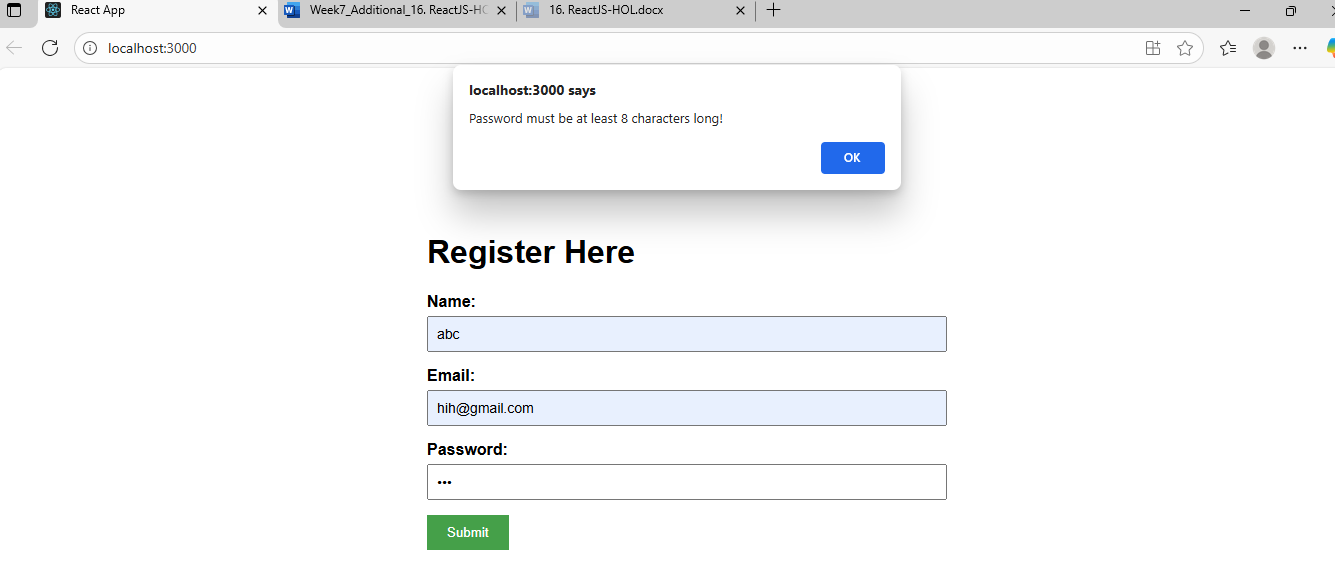
font-size: 12px;

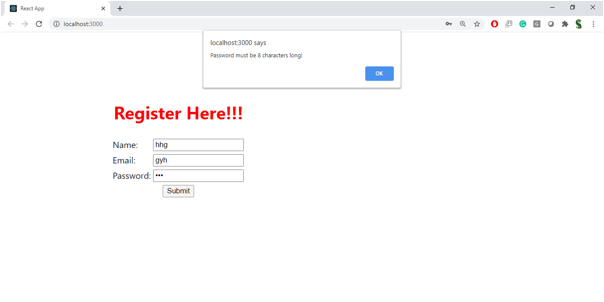
margin-left: 10px;

}

**OUTPUT:**







**Hint:**

