## **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**

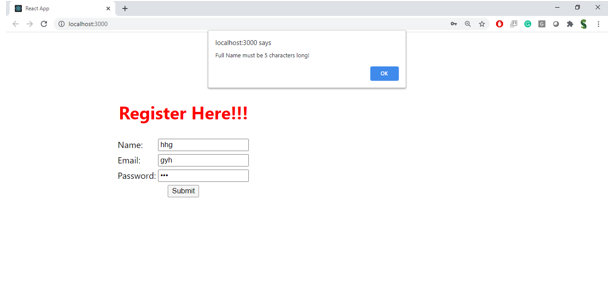
Estimated time to complete this lab: **60 minutes.**

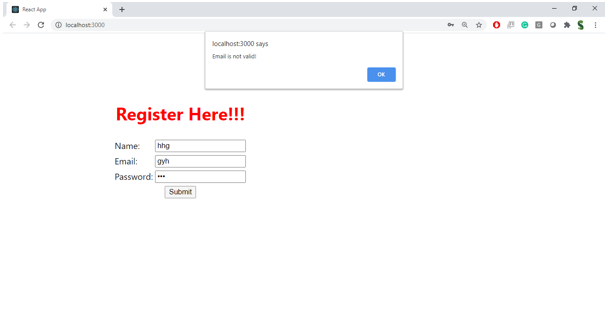
Create a React App named “mailregisterapp” which will have a component named “register.js”. Create a form which accepts the name, email and password and validate the fields as per the following:

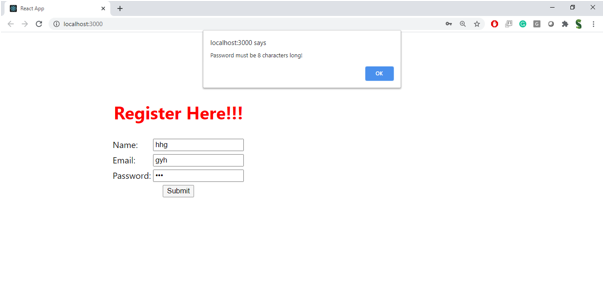
1. Name should have atleast 5 characters
2. Email should have @ and .
3. Password should have atleast 8 characters.

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

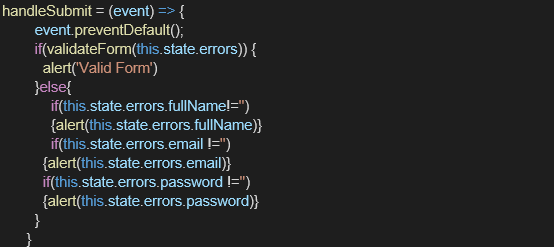
**Output Expected:**

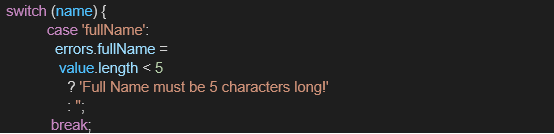


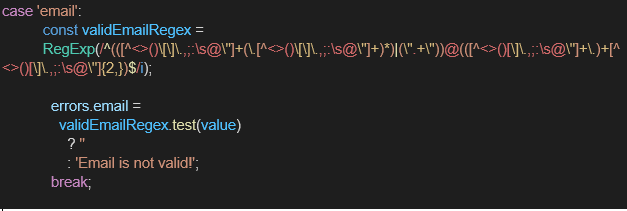


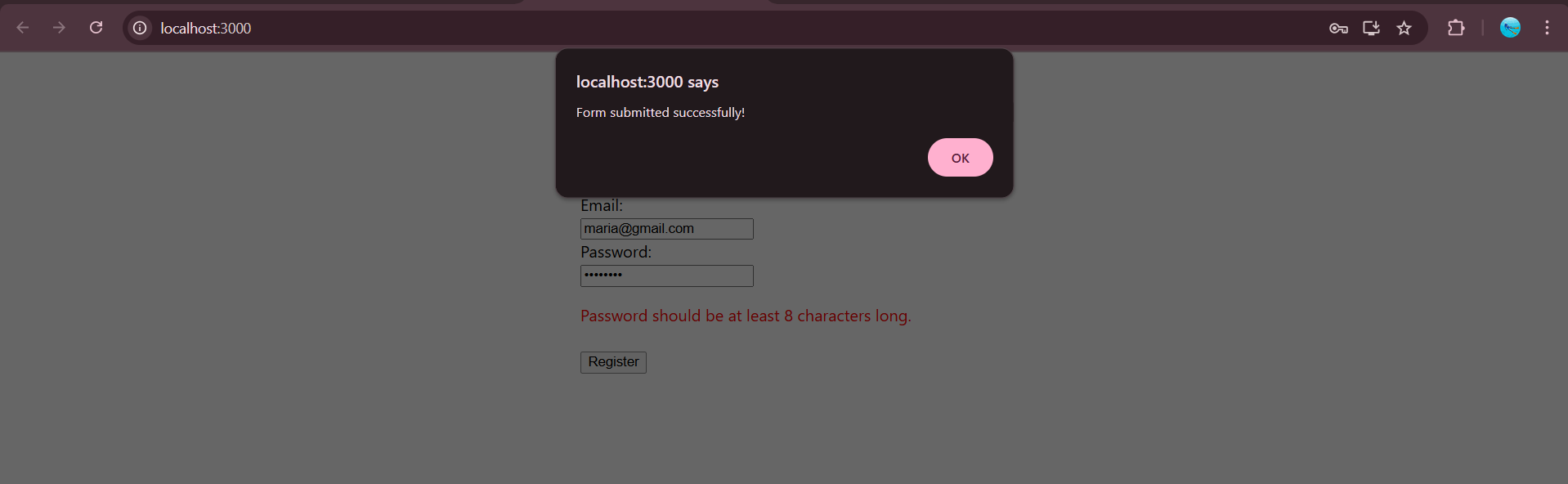


**Hint:**









Register.js

import React, { useState } from 'react';

function Register() {

const [formData, setFormData] = useState({

name: '',

email: '',

password: ''

});

const [errors, setErrors] = useState({});

const validate = () => {

const errors = {};

if (formData.name.trim().length < 5) {

errors.name = 'Name should be at least 5 characters long.';

}

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

errors.email = 'Email should contain "@" and "."';

}

if (formData.password.length < 8) {

errors.password = 'Password should be at least 8 characters long.';

}

setErrors(errors);

return Object.keys(errors).length === 0;

};

const handleChange = (e) => {

setFormData({

...formData,

[e.target.name]: e.target.value

});

};

const handleSubmit = (e) => {

e.preventDefault();

if (validate()) {

alert('Form submitted successfully!');

console.log(formData);

}

};

return (

<div style={{ padding: "20px", maxWidth: "400px", margin: "auto" }}>

<h2>Register</h2>

<form onSubmit={handleSubmit} noValidate>

<div>

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

<input

type="text"

name="name"

value={formData.name}

onChange={handleChange}

/>

{errors.name && <p style={{ color: 'red' }}>{errors.name}</p>}

</div>

<div>

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

<input

type="text"

name="email"

value={formData.email}

onChange={handleChange}

/>

{errors.email && <p style={{ color: 'red' }}>{errors.email}</p>}

</div>

<div>

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

<input

type="password"

name="password"

value={formData.password}

onChange={handleChange}

/>

{errors.password && <p style={{ color: 'red' }}>{errors.password}</p>}

</div>

<button type="submit" style={{ marginTop: "10px" }}>Register</button>

</form>

</div>

);

}

export default Register;