**16. ReactJS-HOL**

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.

**App.js**

import React from 'react';

import Register from './Register';

import './App.css';

function App() {

  return (

    <div className="App">

      <Register />

    </div>

  );

}

export default App;

**Register.js**

import React from 'react';

const validEmailRegex = RegExp(

  /^(([^<>()\[\]\.,;:\s@\"]+(\.[^<>()\[\]\.,;:\s@\"]+)\*)|(\".+\"))@(([^<>()[\]\.,;:\s@\"]+\.)+[^<>()[\]\.,;:\s@\"]{2,})$/i

);

const validateForm = errors => {

  let valid = true;

  Object.values(errors).forEach(val => val.length > 0 && (valid = false));

  return valid;

};

class Register extends React.Component {

  constructor(props) {

    super(props);

    this.state = {

      fullName: '',

      email: '',

      password: '',

      errors: {

        fullName: '',

        email: '',

        password: '',

      }

    };

  }

  handleChange = (event) => {

    event.preventDefault();

    const { name, value } = event.target;

    let errors = this.state.errors;

    switch (name) {

      case 'fullName':

        errors.fullName =

          value.length < 5

            ? 'Full Name must be 5 characters long!'

            : '';

        break;

      case 'email':

        errors.email =

          validEmailRegex.test(value)

            ? ''

            : 'Email is not valid!';

        break;

      case 'password':

        errors.password =

          value.length < 8

            ? 'Password must be 8 characters long!'

            : '';

        break;

      default:

        break;

    }

    this.setState({ errors, [name]: value });

  }

  handleSubmit = (event) => {

    event.preventDefault();

    if (validateForm(this.state.errors)) {

      alert('Valid Form');

    } else {

      if (this.state.errors.fullName) {

        alert(this.state.errors.fullName);

      }

      if (this.state.errors.email) {

        alert(this.state.errors.email);

      }

      if (this.state.errors.password) {

        alert(this.state.errors.password);

      }

    }

  }

  render() {

    const { errors } = this.state;

    return (

      <div className='wrapper'>

        <h1 style={{color: 'red'}}>Register Here!!!</h1>

        <form onSubmit={this.handleSubmit} noValidate>

          <div className='form-group'>

            <label>Name</label>

            <input type='text' name='fullName' onChange={this.handleChange} noValidate />

          </div>

          <div className='form-group'>

            <label>Email</label>

            <input type='email' name='email' onChange={this.handleChange} noValidate />

          </div>

          <div className='form-group'>

            <label>Password</label>

            <input type='password' name='password' onChange={this.handleChange} noValidate />

          </div>

          <div className='form-group'>

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

          </div>

        </form>

      </div>

    );

  }

}

export default Register;

**App.css**

.App {

  font-family: sans-serif;

  display: flex;

  justify-content: center;

}

.wrapper {

  margin-top: 50px;

  padding: 20px;

  border: 1px solid #ccc;

  border-radius: 8px;

  width: 300px;

}

.form-group {

  margin-bottom: 15px;

}

label {

  display: block;

  margin-bottom: 5px;

}

input {

  width: 95%;

  padding: 8px;

  border-radius: 4px;

  border: 1px solid #ccc;

}

button {

  width: 100%;

  padding: 10px;

  background-color: #007bff;

  color: white;

  border: none;

  border-radius: 4px;

  cursor: pointer;

}

button:hover {

  background-color: #0056b3;

}

**Output:**

