**บทที่ 9 แบบฟอร์มใน React**

**การสร้างแบบฟอร์มด้วยวิธีอ้างอิงกับ state**

**ขั้นตอนที่ 1 สร้างคอมโปเน้นท์ LoginForm**

import React, { Component } from 'react';

class LoginForm extends Component {

render() {

return (

<div>

Form in here!!!!

</div>

)

}

}

export default LoginForm;

import React from 'react';

import 'bootstrap/dist/css/bootstrap.min.css';

import LoginForm from './Login-form';

function App() {

return (

<div className="container">

<LoginForm />

</div>

);

}

export default App;

**ขั้นตอนที่ 2 กำหนดหน้าตาแบบฟอร์มด้วย Bootstrap**

import React, { Component } from 'react';

class LoginForm extends Component {

render() {

return (

<div className ="col-6 mt-5 mx-auto card">

<div className ="card-body ">

<form>

<div className ="form-group">

<label htmlFor="username">User Name</label>

<input

type="text"

className ="form-control"

id="username" name="username" />

</div>

<div class="form-group">

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

<input

type="password"

className ="form-control"

id="password" name="password" />

</div>

<div className="text-center">

<button

type="submit"

className ="btn btn-primary my-1">Login</button>

</div>

</form>

</div>

</div>

)

}

}

export default LoginForm;

**ขั้นตอนที่ 3 กำหนด state สำหรับเก็บข้อมูลจากแบบฟอร์ม**

import React, { Component } from 'react';

class LoginForm extends Component {

state = {

username: '',

password: ''

}

onInputChange = (event) => {

this.setState({

[event.target.name]: event.target.value

})

console.log(this.state);

}

render() {

return (

<div className ="col-6 mt-5 mx-auto card">

<div className ="card-body ">

<form>

<div className ="form-group">

<label htmlFor="username">User Name</label>

<input

type="text"

className ="form-control"

id="username"

name="username"

onChange={this.onInputChange}

/>

</div>

<div className="form-group">

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

<input

type="password"

className ="form-control"

id="password"

name="password"

onChange={this.onInputChange}

/>

</div>

<div className="text-center">

<button

type="submit"

className ="btn btn-primary my-1">Login</button>

</div>

</form>

</div>

</div>

)

}

}

export default LoginForm;

**ขั้นตอนที่ 4 ส่งข้อมูลจากแบบฟอร์ม**

import React, { Component } from 'react';

class LoginForm extends Component {

state = {

username: '',

password: ''

}

onInputChange = (event) => {

this.setState({

[event.target.name]: event.target.value

})

console.log(this.state);

}

onLoginSubmit = (event) => {

event.preventDefault();

console.log(this.state);

}

render() {

return (

<div className ="col-6 mt-5 mx-auto card">

<div className ="card-body ">

<form onSubmit={this.onLoginSubmit}>

<div className ="form-group">

<label htmlFor="username">User Name</label>

<input

type="text"

className ="form-control"

id="username"

name="username"

onChange={this.onInputChange}

/>

</div>

<div className="form-group">

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

<input

type="password"

className ="form-control"

id="password"

name="password"

onChange={this.onInputChange}

/>

</div>

<div className="text-center">

<button

type="submit"

className ="btn btn-primary my-1">Login</button>

</div>

</form>

</div>

</div>

)

}

}

export default LoginForm;

**CSS คลาสของ Bootstrap ที่จำเป็นต้องใช้ในแบบฟอร์ม**

import React, { Component } from 'react';

class LoginForm extends Component {

state = {

username: '',

password: ''

}

onInputChange = (event) => {

this.setState({

[event.target.name]: event.target.value

})

}

onLoginSubmit = (event) => {

event.preventDefault();

console.log(this.state);

}

render() {

return (

<div className ="col-6 mt-5 mx-auto card">

<div className ="card-body ">

<form onSubmit={this.onLoginSubmit}>

<div className ="form-group">

<label htmlFor="username">User Name</label>

<input

type="text"

className ="form-control is-valid"

id="username"

name="username"

onChange={this.onInputChange}

/>

<div className="valid-feedback">พบชื่อผู้ใช้</div>

</div>

<div className="form-group">

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

<input

type="password"

className ="form-control is-invalid"

id="password"

name="password"

onChange={this.onInputChange}

/>

<div className="invalid-feedback">รหัสผ่านสั้นเกินไป</div>

</div>

<div className="text-center">

<button

type="submit"

className ="btn btn-primary my-1">Login</button>

</div>

</form>

</div>

</div>

)

}

}

export default LoginForm;

**ตัวอย่างแบบฟอร์มในทางปฏิบัติ**

**ขั้นตอนที่ 1 กำหนด Register คอมโปเน้นท์**

import React, { Component } from 'react';

class Register extends Component {

render() {

return(

<div className="row">

<div className ="col-sm-3 mt-5"></div>

<div className ="col-sm-6 mt-5 card">

<div className ="card-body ml-3 mr-3 mt-5 mb-1">

<form>

<div className ="form-group">

<label htmlFor="username">User Name</label>

<input

type="text"

className ="form-control"

id="username"

name="username"

/>

<div className="invalid-feedback">กรอกชื่อผู้ใช้อย่างน้อย 5 ตัวอักษร</div>

</div>

<div className="form-group">

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

<input

type="email"

className ="form-control"

id="email"

name="email"

/>

<div className="invalid-feedback">คุณกรอกอีเมล์ไม่ถูกต้อง</div>

</div>

<div className="form-group">

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

<input

type="password"

className ="form-control"

id="password"

name="password"

/>

<div className="invalid-feedback">ต้องกรอกอย่างน้อย 8 ตัวอักษร</div>

</div>

<div className="text-center">

<button

type="submit"

className ="btn btn-primary">Register</button>

</div>

</form>

</div>

</div>

<div className ="col-sm-3 mt-5"></div>

</div>

)

}

}

export default Register;

import React from 'react';

import 'bootstrap/dist/css/bootstrap.min.css';

import Register from './Register';

const App = () => {

return (

<div className="container">

<Register />

</div>

);

}

export default App;

**ขั้นตอนที่ 2 กำหนด state ให้สามารถติดตามค่าในอินพุตได้**

state = {

formElements: {

username: {

type: 'text',

value: '',

validator: {

required: true,

minLength: 5,

maxLength: 15

},

touched: false,

error: {status: true, message: ''}

},

email: {

type: 'email',

value: '',

validator: {

required: true,

pattern: 'email'

},

touched: false,

error: {status: true, message: ''}

},

password: {

type: 'password',

value: '',

validator: {

required: true,

minLength: 8

},

touched: false,

error: {status: true, message: ''}

}

},

formValid: false

}

username: {

type: 'text',

value: '',

validator: {

required: true,

minLength: 5,

maxLength: 15

},

touched: false,

error: {status: true, message: ''}

}

**ขั้นตอนที่ 3 กำหนดเมธอด onFormChange สำหรับติดตามค่าในฟอร์ม**

onFormChange = (event) => {

const name = event.target.name;

const value = event.target.value;

let updatedForm = {...this.state.formElements};

updatedForm[name].value = value;

updatedForm[name].touched = true;

const validatorObject = this.checkValidator(value, updatedForm[name].validator);

updatedForm[name].error = {

status: validatorObject.status,

message: validatorObject.message

}

let formStatus = true;

for (let name in updatedForm) {

if(updatedForm[name].validator.required === true){

formStatus = !updatedForm[name].error.status && formStatus;

}

}

this.setState({

...this.state,

formElements: updatedForm,

formValid: formStatus

});

}

**ขั้นตอนที่ 4 เพิ่มเมธอด checkValidator(value, rule) ไปยังไฟล์ Registr.js**

checkValidator = (value, rule) => {

let valid = true;

let message = '';

if(value.trim().length === 0 && rule.required) {

valid = false;

message = 'จำเป็นต้องกรอก';

}

if(value.length < rule.minLength && valid) {

valid = false;

message = `น้อยกว่า ${rule.minLength} ตัวอักษร`;

}

if(value.length > rule.maxLength && valid) {

valid = false;

message = `มากกว่า ${rule.maxLength} ตัวอักษร`;

}

if(rule.pattern === 'email' && valid){

if(/^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(value) === false) {

valid = false;

message = 'กรอกอีเมล์ไม่ถูกต้อง';

}

}

return { status:!valid, message:message};

}

**ขั้นตอนที่ 5 กำหนดคลาส is-invalid ที่อินพุตหากผู้ใช้กรอกข้อมูลไม่ตรงเงื่อนไข**

getInputClass = (name) => {

const elementErrorStatus = this.state.formElements[name].error.status;

return elementErrorStatus && this.state.formElements[name].touched ?

'form-control is-invalid' :

'form-control is-valid';

}

**ขั้นตอนที่ 6 แสดงข้อความเตือนเมื่อผู้ใช้กรอกข้อมูลไม่ถูกต้อง**

getErrorMessage = (name) => {

return this.state.formElements[name].error.message;

}

**ขั้นตอนที่ 7 ยกเลิกปุ่มส่งแบบฟอร์ม หากกรอกข้อมูลไม่ถูกต้อง**

<button

type="submit"

className="btn btn-primary"

disabled={!this.state.formValid}

>Register</button>

**ขั้นตอนที่ 8 กำหนดเมธอด สำหรับส่งแบบฟอร์ม**

onFormSubmit = (event) => {

event.preventDefault();

const formData = {};

for (let name in this.state.formElements) {

formData[name] = this.state.formElements[name].value;

}

console.log(formData);

}

**ขั้นตอนที่ 9 แก้ไขแบบฟอร์ม**

<div className="row">

<div className ="col-sm-3 mt-5"></div>

<div className ="col-sm-6 mt-5 card">

<div className ="card-body ml-3 mr-3 mt-5 mb-1">

<form onSubmit={this.onFormSubmit}>

<div className ="form-group">

<label htmlFor="username">User Name</label>

<input

type="text"

className={this.getInputClass('username')}

id="username"

name="username"

onChange={this.onFormChange}

/>

<div className="invalid-feedback">

{ this.getErrorMessage('username') }

</div>

</div>

<div className="form-group">

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

<input

type="email"

className={this.getInputClass('email')}

id="email"

name="email"

onChange={this.onFormChange}

/>

<div className="invalid-feedback">

{ this.getErrorMessage('email') }

</div>

</div>

<div className="form-group">

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

<input

type="password"

className={this.getInputClass('password')}

id="password"

name="password"

onChange={this.onFormChange}

/>

<div className="invalid-feedback">

{ this.getErrorMessage('password') }

</div>

</div>

<div className="text-center">

<button

type="submit"

className="btn btn-primary"

disabled={!this.state.formValid}

>Register</button>

</div>

</form>

</div>

</div>

<div className ="col-sm-3 mt-5"></div>

</div>

**ตัวอย่างโค้ดสำหรับแบบฟอร์มลงทะเบียน**

import React, { Component } from 'react';

class Register extends Component {

state = {

formElements: {

username: {

type: 'text',

value: '',

validator: {

required: true,

minLength: 5,

maxLength: 15

},

touched: false,

error: {status: true, message: ''}

},

email: {

type: 'email',

value: '',

validator: {

required: true,

pattern: 'email'

},

touched: false,

error: {status: true, message: ''}

},

password: {

type: 'password',

value: '',

validator: {

required: true,

minLength: 8,

maxLength: 20

},

touched: false,

error: {status: true, message: ''}

}

},

formValid: false

}

onFormChange = (event) => {

const name = event.target.name;

const value = event.target.value;

let updatedForm = {...this.state.formElements};

updatedForm[name].value = value;

updatedForm[name].touched = true;

const validatorObject = this.checkValidator(value, updatedForm[name].validator);

updatedForm[name].error = {

status: validatorObject.status,

message: validatorObject.message

}

let formStatus = true;

for (let name in updatedForm) {

if(updatedForm[name].validator.required === true){

formStatus = !updatedForm[name].error.status && formStatus;

}

}

this.setState({

...this.state,

formElements: updatedForm,

formValid: formStatus

});

}

checkValidator = (value, rule) => {

let valid = true;

let message = '';

if(rule.required) {

if (value.trim().length === 0) {

valid = false;

message = 'จำเป็นต้องกรอก';

}

}

if(value.length < rule.minLength && valid) {

valid = false;

message = `น้อยกว่า ${rule.minLength} ตัวอักษร`;

}

if(value.length > rule.maxLength && valid) {

valid = false;

message = `มากกว่า ${rule.maxLength} ตัวอักษร`;

}

if(rule.pattern === 'email' && valid){

if(/^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(value) === false) {

valid = false;

message = 'กรอกอีเมล์ไม่ถูกต้อง';

}

}

return { status:!valid, message:message};

}

getErrorMessage = (name) => {

return this.state.formElements[name].error.message;

}

getInputClass = (name) => {

const elementErrorStatus = this.state.formElements[name].error.status;

return elementErrorStatus && this.state.formElements[name].touched ?

'form-control is-invalid' :

'form-control is-valid';

}

onFormSubmit = (event) => {

event.preventDefault();

const formData = {};

for (let name in this.state.formElements) {

formData[name] = this.state.formElements[name].value;

}

console.log(formData);

}

render() {

return(

<div className="row">

<div className ="col-sm-3 mt-5"></div>

<div className ="col-sm-6 mt-5 card">

<div className ="card-body ml-3 mr-3 mt-5 mb-1">

<form onSubmit={this.onFormSubmit}>

<div className ="form-group">

<label htmlFor="username">User Name\*</label>

<input

type="text"

className={this.getInputClass('username')}

id="username"

name="username"

onChange={this.onFormChange}

/>

<div className="invalid-feedback">

{ this.getErrorMessage('username') }

</div>

</div>

<div className="form-group">

<label htmlFor="email">Email\*</label>

<input

type="email"

className={this.getInputClass('email')}

id="email"

name="email"

onChange={this.onFormChange}

/>

<div className="invalid-feedback">

{ this.getErrorMessage('email') }

</div>

</div>

<div className="form-group">

<label htmlFor="password">Password\*</label>

<input

type="password"

className={this.getInputClass('password')}

id="password"

name="password"

onChange={this.onFormChange}

/>

<div className="invalid-feedback">

{ this.getErrorMessage('password') }

</div>

</div>

<div className="text-center">

<button

type="submit"

className="btn btn-primary"

disabled={!this.state.formValid}

>Register</button>

</div>

</form>

</div>

</div>

<div className ="col-sm-3 mt-5"></div>

</div>

)

}

}

export default Register;