### Redux Toolkit

#### Objectives and Outcomes

In this exercise you will learn to use Form validation by Formik. You will install and configure Formik, Yup and use it within your React application. At the end of this exercise, you will be able to:

* Install and configure Formik, Yup within your React application

#### Install and Configure Formik, Yup

* As a first step you will create a new React application. Then install Formik and Yup into your application as follows:

npx create-react-app ReactForm

npm install formik yup

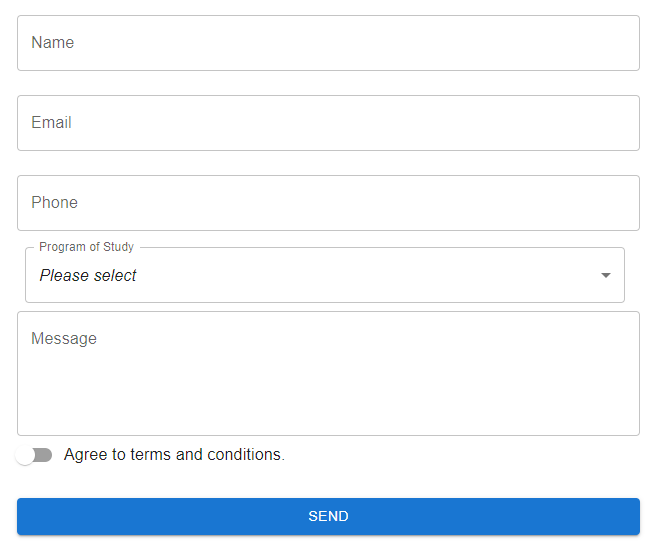
Your package.json file like:

"formik": "^2.2.9",

"yup": "^0.32.11" ,

#### Create Contact components for display the view

* Create the Contact component (named Contact.js) to add the new contact by yourself (use Materialize, Material UI or some libraries which support Material design), the screen display as follow:



#### Import useFormik and create initial values

* Open the Contact.js file, import useFormik and write codes as follow:

…

import { useFormik } from 'formik';

…

const formik = useFormik({

initialValues:{

name:"",

email:"",

phone:"",

program:0,

message:"",

agree:false

},

});

#### Update the initial values and get the values of controls (textbox, select,…)

<form onSubmit={formik.handleSubmit}>

<TextField

label="Name"

name="name"

value={formik.values.name}

onChange={formik.handleChange}

/>

<TextField

label="Email"

name="email"

value={formik.values.email}

onChange={formik.handleChange}

/>

<TextField

label="phone"

name="phone"

value={formik.values.phone}

onChange={formik.handleChange}

/>

<FormControl sx={{ m: 1, minWidth: 600 }}>

<InputLabel id="demo-simple-select-autowidth-label">Program of Study</InputLabel>

<Select

labelId="demo-simple-select-autowidth-label"

id="demo-simple-select-autowidth"

label="Program of Stydy"

name="program"

value={formik.values.program}

onChange={formik.handleChange}

>

<MenuItem value={0}>

<em>Please select</em>

</MenuItem>

<MenuItem value={1}>Software Engineering</MenuItem>

<MenuItem value={2}>Information System</MenuItem>

<MenuItem value={3}>Information Assurance</MenuItem>

<MenuItem value={4}>Internet of Things</MenuItem>

<MenuItem value={5}>Artificial Intelligence</MenuItem>

<MenuItem value={6}>Digital Art & Design</MenuItem>

</Select>

</FormControl>

<TextField

id="outlined-multiline-static"

label="Message"

multiline

name='message'

rows={4}

value={formik.values.message}

onChange={formik.handleChange}

/>

<FormControlLabel control={<Switch/>} label="Agree to terms and conditions." name='agree'

value={formik.values.agree} onClick={formik.handleChange} />

<Button

type='submit'>

Send

</Button>

</form>

**Validate to form values by Yup**

* Next, we need to import the Yup to validate the values of form.
* Write codes as follow:

….

import \* as Yup from 'yup';

…

const formik = useFormik({

initialValues:{

name:"",

email:"",

phone:"",

program:0,

message:"",

agree:false

},

onSubmit: (values)=>{

},

validationSchema: Yup.object({

name: Yup.string().required("Required.").min(2, "Must be 2 characters or more"),

email: Yup.string().required("Required.").email("Invalid email"),

phone: Yup.number().integer().typeError("Please enter a valid number"),

program: Yup.number().integer().typeError("Please select a program."),

message: Yup.string().required("Required.").min(10, "Must be 10 characters or more"),

agree: Yup.boolean().oneOf([true], "The terms and conditions must be accepted.")

}),

});

#### Show the errors of control values

* Update codes as follow:

. . .

<Textfield

name="name"

. . .

/>

{formik.errors.name && (<Typography variant="caption" color="red">{formik.errors.name}</Typography>)}

<Textfield

name="email"

. . .

/>

{formik.errors.email && (<Typography variant="caption" color="red">{formik.errors.email}</Typography>)}

<Textfield

name="phone"

. . .

/>

{formik.errors.phone && (<Typography variant="caption" color="red">{formik.errors.phone}</Typography>)}

<select

name="program"

>

. . .

</select>

{formik.errors.program && (<Typography variant="caption" color="red">{formik.errors.program}</Typography>)}

<TextField

name='message'

...

/>

{formik.errors.message && (<Typography variant="caption" color="red">{formik.errors.message}</Typography>)}

<FormControlLabel control={<Switch/>} label="Agree to terms and conditions." name='agree' value={formik.values.agree} onClick={formik.handleChange}/>

{formik.errors.agree && (<Typography variant="caption" color="red">{formik.errors.agree}</Typography>)}

* Submit form when all data had validated:

onSubmit: (values)=>{alert(JSON.stringify(formik.values))},

//Convert to JSON string and show in alert box.

#### Run your React app and see the result

#### Conclusions

In this exercise you learnt to use Formik and Yup in React application.