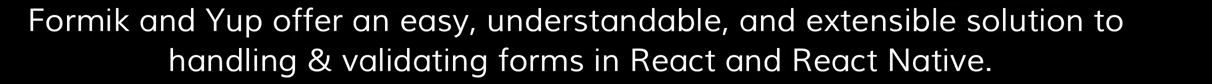
**Form Validation**



<https://www.npmjs.com/package/formik>

bun i formik

<https://www.npmjs.com/package/yup>

bun i yup

import {useFormik} from 'formik';

import {useFormik} from 'formik';

**Form handling with Formik**

**import { useFormik } from 'formik';**

**const initialValues = {**

**name: "",**

**email: "",**

**password: "",**

**con\_password: "",**

**}**

**export const Form = () => {**

***// const Formik = useFormik({***

***// initialValues: initialValues,***

***// onSubmit: (values) => {***

***// console.log(values);***

***// }***

***// })***

***// value={Formik.values.name}***

**const { values, errors, handleBlur, handleChange, handleSubmit } = useFormik({ *//destructured***

**initialValues: initialValues,**

**onSubmit: (*values*) => {**

**console.log(*values*);**

**}**

**})**

**return (**

**<>**

**<section>**

**<div *className*="container">**

**<div *className*="row">**

**<div *className*="col-10">**

**<form *onSubmit*={handleSubmit}>**

**<div *className*="mb-3">**

**<label *htmlFor*="name" *className*="form-label">Name</label>**

**<input *type*="text" *name*='name' *className*="form-control" *id*="name" *autoComplete*='false' *value*={values.name} *onChange*={handleChange} *onBlur*={handleBlur} />**

**</div>**

**<div *className*="mb-3">**

**<label *htmlFor*="email" *className*="form-label">Email address</label>**

**<input *type*="email" *name*='email' *className*="form-control" *id*="email" *autoComplete*='false' *value*={values.email} *onChange*={handleChange} *onBlur*={handleBlur} />**

**</div>**

**<div *className*="mb-3">**

**<label *htmlFor*="password" *className*="form-label">Password</label>**

**<input *type*="password" *name*='password' *className*="form-control" *id*="password" *autoComplete*='false' *value*={values.password} *onChange*={handleChange} *onBlur*={handleBlur} />**

**</div>**

**<div *className*="mb-3">**

**<label *htmlFor*="con\_password" *className*="form-label">Confirm Password</label>**

**<input *type*="con\_password" *name*='con\_password' *className*="form-control" *id*="con\_password" *autoComplete*='false' *value*={values.con\_password} *onChange*={handleChange} *onBlur*={handleBlur} />**

**</div>**

**<button *type*="submit" *className*="btn btn-primary">Submit</button>**

**</form>**

**</div>**

**</div>**

**</div>**

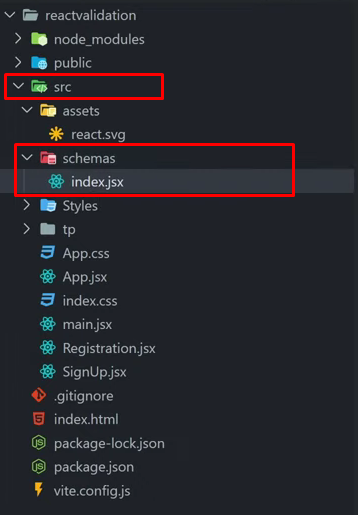
**</section>**

**</>**

**)**

**}**

**Create schema folder inside src folder 👇**



**Inside index.jsx 👇**

**import \* as Yup from "yup";**

**export const signUpSchema = Yup.object({**

**name: Yup.string().min(2).max(25).required("Please enter your name"),**

**email: Yup.string().email().required("Please enter your email"),**

**password: Yup.string().min(6).required("Please enter your password"),**

**con\_password: Yup.string().required().oneOf([Yup.ref("password"), null], "Password must match"),**

**});**

**Complete form.jsx 👇**

**import { useFormik } from 'formik';**

**import { signUpSchema } from "./schema"**

**const initialValues = {**

**name: "",**

**email: "",**

**password: "",**

**con\_password: "",**

**}**

**export const Form = () => {**

***// const Formik = useFormik({***

***// initialValues: initialValues,***

***// onSubmit: (values) => {***

***// console.log(values);***

***// }***

***// })***

***// value={Formik.values.name}***

**const { values, errors, handleBlur, touched, handleChange, handleSubmit } = useFormik({ *//destructured***

**initialValues: initialValues,**

**validationSchema: signUpSchema,**

**validateOnChange: true,**

**onSubmit: (*values*, { *resetForm* }) => {**

**console.log(*values*);**

**resetForm(); *// Call resetForm to reset the form fields***

**}**

**})**

**console.log(errors);**

**return (**

**<>**

**<section>**

**<div *className*="container">**

**<div *className*="row">**

**<div *className*="col-10">**

**<form *onSubmit*={handleSubmit}>**

**<div *className*="mb-3">**

**<label *htmlFor*="name" *className*="form-label">Name</label>**

**<input *type*="text" *name*='name' *className*="form-control" *id*="name" *autoComplete*='false' *value*={values.name} *onChange*={handleChange} *onBlur*={handleBlur} />**

**{errors.name && touched.name ? (<p>{errors.name}</p>) : null}**

**</div>**

**<div *className*="mb-3">**

**<label *htmlFor*="email" *className*="form-label">Email address</label>**

**<input *type*="email" *name*='email' *className*="form-control" *id*="email" *autoComplete*='false' *value*={values.email} *onChange*={handleChange} *onBlur*={handleBlur} />**

**{errors.email && touched.email ? (<p>{errors.email}</p>) : null}**

**</div>**

**<div *className*="mb-3">**

**<label *htmlFor*="password" *className*="form-label">Password</label>**

**<input *type*="password" *name*='password' *className*="form-control" *id*="password" *autoComplete*='false' *value*={values.password} *onChange*={handleChange} *onBlur*={handleBlur} />**

**{errors.password && touched.password ? (<p>{errors.password}</p>) : null}**

**</div>**

**<div *className*="mb-3">**

**<label *htmlFor*="con\_password" *className*="form-label">Confirm Password</label>**

**<input *type*="con\_password" *name*='con\_password' *className*="form-control" *id*="con\_password" *autoComplete*='false' *value*={values.con\_password} *onChange*={handleChange} *onBlur*={handleBlur} />**

**{errors.con\_password && touched.con\_password ? (<p>{errors.con\_password}</p>) : null}**

**</div>**

**<button *type*="submit" *className*="btn btn-primary">Submit</button>**

**</form>**

**</div>**

**</div>**

**</div>**

**</section>**

**</>**

**)**

**}**