**Software Development GAPS  
  
1. AS-IS   : Defines what your technology is capable of.  
2. To-BE  : Defines what client requirements are.  
3. GAP     : Defines the functionalities not supported by your technology  
                and client needs them.  
  
How to fill the GAP?  
- By using 3rd Party Implementations.  
  
Why React Needs 3rd Party?  
- It is not designed for what you are using.  
- Hence lot of GAPS.  
- 3rd Party can fill the GAPS.  
  
What are the popular 3rd Party for Forms in React?  
- Formik                            [Popular]  
- KendoForms  
- React-Hook-Forms            [Trending] React 16+ version  
- React-Final-Forms etc..  
  
                "npmjs.com"  
  
1. Install Formik Library for your project  
  
        > npm install  formik --save                    [MIT]  
  
  
2. Import "useFormik" hook  
  
   import { useFormik } from "formik"  
  
3. Configure formik form with values and submit event.  
  
    const  formik = useFormik({  
               initialValues: { },  
            onSubmit : function() { }  
    })  
  
    \* initialValues is an object that defines the values to use for form.  
    \* onSubmit defines the actions to perform on form submit  
  
4. You have to bind the formik values and events to form elements.  
  
     <input type="text"  value={formik.values.UserName} onChange={formik.handleChange} />  
  
     <form onSubmit={formik.handleSubmit}>  
  
  
5. You can present the values or submit  
  
        <dd> { fromik.values.UserName } </dd>  
  
        onSubmit : values => {  
             alert(JSON.stringfy(values);  
          }  
  
Ex: FormikDemo.js  
  
  
import { useFormik } from "formik"  
  
export default function FormikDemo()  
{  
    const formik = useFormik({  
        initialValues : {  
            UserName: '',  
            Password: '',  
            City: ''  
        },  
        onSubmit: values => {  
            alert(JSON.stringify(values));  
        }  
    })  
    return(  
        <div className="container-fluid">  
            <form onSubmit={formik.handleSubmit}>  
                <h2>Register User</h2>  
                <dl>  
                    <dt>User Name</dt>  
                    <dd><input name="UserName" onChange={formik.handleChange} value={formik.values.UserName} type="text"/></dd>  
                    <dt>Password</dt>  
                    <dd><input name="Password" onChange={formik.handleChange} value={formik.values.Password} type="password"/></dd>  
                    <dt>City</dt>  
                    <dd>  
                        <select onChange={formik.handleChange} value={formik.values.City} name="City">  
                            <option>Delhi</option>  
                            <option>Hyd</option>  
                        </select>  
                    </dd>  
                </dl>  
                <button>Register</button>  
            </form>  
        </div>  
    )  
}  
  
  
Ex: With Boolean  
  
import { Formik, useFormik } from "formik"  
  
export default function FormikDemo()  
{  
    const formik = useFormik({  
        initialValues : {  
            UserName: '',  
            Password: '',  
            City: '',  
            Subscribe: true  
        },  
        onSubmit: values => {  
            alert(`${values.UserName}\nSubscription: ${(values.Subscribe==true)?"Subscribed":"Not Subscribed"}`);  
        }  
    })  
    return(  
        <div className="container-fluid">  
            <form onSubmit={formik.handleSubmit}>  
                <h2>Register User</h2>  
                <dl>  
                    <dt>User Name</dt>  
                    <dd><input name="UserName" onChange={formik.handleChange} value={formik.values.UserName} type="text"/></dd>  
                    <dt>Password</dt>  
                    <dd><input name="Password" onChange={formik.handleChange} value={formik.values.Password} type="password"/></dd>  
                    <dt>City</dt>  
                    <dd>  
                        <select onChange={formik.handleChange} value={formik.values.City} name="City">  
                            <option>Delhi</option>  
                            <option>Hyd</option>  
                        </select>  
                    </dd>  
                    <dt>Subscribe</dt>  
                    <dd className="form-switch"><input className="form-check-input" name="Subscribe" onChange={formik.handleChange} checked={formik.values.Subscribe} type="checkbox" /> Yes</dd>  
                </dl>  
                <button>Register</button>  
            </form>  
        </div>  
    )  
}**