**React Formik Notes**

**Introduction**

* Forms are important for users to enter data
* Formik is a small library that helps you deal with forms in React
* Formik helps developers with the following:
  + Handle form data
  + Validation
  + Visual feedback with error messages
  + Form submission
* While you can handle forms with just React, Formik allows us to deal with forms in a scalable, performant and easy way.

**Creating an HTML Form**

* To start off, we will create a form that will be sent to YouTube that looks like:
* A picture containing text

  Description automatically generated
* When creating this form, we will be concerned with:
  + Managing the form state
  + Handling form submission
  + Validation and error messages
* To create this form, make a new react app as shown below:
* Text

  Description automatically generated Text

  Description automatically generated
* Text

  Description automatically generated Text

  Description automatically generated
* Focus on the Youtube Form component.
* Notice that since for is a reserved keyword in js, react uses ‘htmlFor’ instead of ‘for’ when dealing with forms. Recall that a label’s ‘htmlFor’ value should be the id of the input field it corresponds to. For example, the text field where the user can enter their channel name has an id of ‘channel’. Thus, the ‘Channel’ label’s htmlFor value is also ‘channel’. Not only is the htmlFor property is considered good practice to implement on a label tag, but the label’s corresponding input field is also focused when the user clicks on the lable.
* Recall that the name attribute specifies the name of an <input> element. The name attribute is used to reference elements in a JavaScript, or to reference form data after a form is submitted. Only form elements with a name attribute will have their values passed when submitting a form.

**useFormik Hook**

* To use the Formik library, we have to install it by running in cmd ‘npm i formik’
* This Formik library provides a hook called useFormik that we can import.
* The useFormik hook takes in an object as its parameter. This hook returns an object which contains a variety of user properties and methods that we can use with our form. This returned object will help us with managing the form state, handling form submission, validation and error messages.
* We can now use the useFormik hook as shown below.
* We can also log out the object that the useFormik hook returns.
* Text

  Description automatically generated Text

  Description automatically generated

**Managing Form State**

* Our youtube form has 3 input fields. However, we are not tracking the value of these three fields. When the user types in something, the value of the fields changes which means we need a state variable for that.
* Thus, we need state variables for the name, email, and channel input fields. Or collectively, we can call them as form state. The form state is an object that maintains the different form fields as shown below.
* Diagram

  Description automatically generated
* If we are able to manage the form state, we can submit this data when the user clicks on submit
* Formik helps us manage this form state.
* To do so, we will need to include an ‘initialValues’ property in the useFormik parameter object. The value of the ‘initialValues’ property is an object which contains the initial values for all our form fields. This ‘initialValues’ property’s object contains key-value pairs. The keys are names of the input field. The value is the initial value of that input field.
* Text

  Description automatically generated
* Notice how above we have an ‘initialValues’ property in the useFormik parameter function. This ‘initialValues’ property’s value is an object that contains the ‘name’, ‘email’, and ‘channel’ properties. Notice that there is a key-value pair of ‘name’ and ‘grant’. This mean that the input field with a name of ‘name’ has its initial value set to ‘grant’.
* Next, we must add the ‘onChange’ and ‘value’ props for each of the form input fields. This is required to ensure the form fields are tracked by Formik. The value of the input field’s ‘onChange’ prop will be formik.handleChange. The value of the input field’s ‘value’ prop will be formik.values.INPUT\_FIELD’S\_NAME\_PROP\_VALUE. An example of this is shown below
* Text

  Description automatically generated
* Notice that the onChange prop has a value of formik.handleChange. Notice that the value we pass to the ‘value’ prop is formik.values.email since the input field has a name of ‘email’. When we change the email input field, the formik.handleChange function is executed which updates formik.values and this update value is then passed to the value ‘prop’ in the email input field.
* We can also access the values from the form through formik.values which we can log to the console.
* Now if we run the following code, we get the following webpage output:
* Text

  Description automatically generated
* Webpage Output: Table

  Description automatically generated with low confidence
* Console output:Text

  Description automatically generated
* Notice in the console, we log out the formik.values object. This is an object where the key corresponds to name attribute of the form input field and the value corresponds to the value of the form field. The value of this formil.values object will be the initialValues object we specified in the useFormik parameter.
* Now, if we enter ‘a’ in the channel input field, we get the following:
* Graphical user interface, application

  Description automatically generated
* Notice how the updated formik.values are logged to the screen which means the component rerenders. The reason why there are two rerenders is explain later.

**Handling Form Submission**

* We used formik to keep track of the form state, but now we want to get hold of this form state when the user submits the button.
* To handle form submission, we need to first specify the ‘onSubmit’ property on the form tag. The value of this ‘onSubmit’ property is formik.handleSubmit.
* 
* We also need to add a new ‘onSubmit’ property to the object we pass to useFormik as a parameter. This ‘onSubmit’ property’s value is a function. This function automatically receives the form state as its argument. Since we set the ‘onSubmit’ property of the form to formik.handleSubmit, this function is executed whenever the form’s is submitted.
* Graphical user interface, text

  Description automatically generated
* Now, our code looks like the following:
* Text

  Description automatically generated
* Enter the following input: Graphical user interface, text, application, email

  Description automatically generated
* Now, we see that the following is logged to the console:
* Graphical user interface, text

  Description automatically generated
* In our example, the ‘onSubmit’ property’s value is a function that logs out the values. In real world cases, this function likely makes an API call to the server to post the form data.
* Notice that the webpage did not reload when we click submit.
* To get rid of the above warning, add type = ‘submit’ to the button.