Code folder will be founded into : State and life cycle. Component is **Form** .

**Note:** In html by default input field manage a state to taking input. Others tag has not any state. By default behavior. DOM control this default behavior

But react is not html, it is JSX element. So in general JSX input behavior is different. React each class component have **state,** so we can use state for input field. When we pass **value** attribute to the JSX, react can understand that we can manage input by react. If don’t pass then react don’t understand. When we pass value prop. Then we need to use **state** to change the input state.

When we pass value. This is called controlled input

 <form>

          <input

            type="text"

            value="masud"

            name=""

            id=""

            placeholder="Enter Name"

          />

        </form>

Now need to pass change method:

// state

state = {

    title: "JavaScript",

  };

// input handler

  handleInput = (e) => {

    this.setState({

      title: e.target.value,

    });

  };

  render() {

    return (

      <div>

        <form>

          <input

            type="text"

            name=""

            id=""

            value={this.state.title}

            placeholder="Enter Name"

            onChange={this.handleInput}

          />

        </form>

      </div>

    );

  }

When we pass value prop. We must should use onChange method. Now this **controlled** input field

**Multiple form field handle:**

To provide value to the textarea we need to pass as children. Or we can pass **value** prop to the react. Since we have use the value prop. We should use **onChange** handler.

  <textarea

            name="text"

            value={this.state.text}

            onChange={this.handleInput}

          >

            {" "}

          </textarea>

**Select and options form handle:** in select provide value with default value. Then provide **onChange()** handler. When any options is selected then this option will be showed as selected

 <select

            onChange={this.handleInput}

            value={this.state.library}

            name=""

            id=""

          >

            <option value="NodeJS">Node JS</option>

            <option value="React">React JS</option>

            <option value="Angular">Angular</option>

          </select>

Provide default value which must found in option value.if match then default selected will this option

**Checked:**

To handle like input we need to pass **checked** prop to the checkbox. Checkbox value is true or false. So first use state as default value then use checkbox handler.

**// state**

state = {

    title: "JavaScript",

    text: "JavaScript is awesome",

    library: "React",

    isAwesome: true,

  };

**// handler**

handleInput = (e) => {

    if (e.target.type === "text") {

      this.setState({

        title: e.target.value,

      });

    } else if (e.target.type === "textarea") {

      this.setState({

        text: e.target.value,

      });

    } else if (e.target.type === "select-one") {

      this.setState({ library: e.target.value });

    } else if (e.target.type === "checkbox") {

      this.setState({ isAwesome: e.target.checked });

    }

  };

**// checkbox**

 <input

            type="checkbox"

            onChange={this.handleInput}

            checked={this.state.isAwesome}

          />

Above all input handler was controlled form input. But if don’t handle like react way, called uncontrolled

**Form submit handler:**

<form onSubmit={handleFormSubmit}>

By default this onSubmit have a reload issue. When we click on submit button. Page is reloading because of html default behavior. We can prevent it by react way. Using **e.preventDefault();** method

After handling input. Now inside form submit we can do anything what we want to do with this form data.

 // handle form submit

  handleFormSubmit = (e) => {

    e.preventDefault();

    const { title, library, isAwesome, text } = this.state;

    console.log(title, library, isAwesome, text);

  };

Uncontrolled input: mainly default behavior of html form. When we want to use file type input. We must need to use DOM way. We cannot use react form handing. Because file has not state.

**Shortcut input handler like dynamic:**

Inside setState we need to change property name dynamic, because e.target.value is same for all the event. So inside input we can use name. and to set dynamic property we can use [e.target.name]=e.target.value; to set dynamic we use array like [].

  state = {

    title: "JavaScript",

    text: "JavaScript is awesome",

    library: "React",

    isAwesome: true,

  };

Handler dynamic:

[e.target.name] = e.target.value;

Input field with name similar as state:

<input

            type="text"

            name="title"

            value={this.state.title}

            placeholder="Enter Name"

            onChange={this.handleInput}

          />

          <br />

          <br />

          {/\* text area handle  \*/}

          <textarea

            name="text"

            value={this.state.text}

            onChange={this.handleInput}

          >

            {" "}

          </textarea>

We can use **value=null**. Then we can edit input field. But this is uncontrolled.

React popular library **FormMik** to handle react form. React give us as suggestion

Control means react control. React control this control form.