**What Does Application Does ?**

I have designed an App that takes word and returns the smiley pertaining to word on Click of Button using React Hooks for State management and Form Created using React Form

**Prerequisites**

A)-**HTML, CSS, and JavaScript** (Fundamentals to get started)

Some Materials you can look - W3SCHOOLS, FREECODECAMP

B)-Some ES6 concepts of JavaScript like -

**Arrow Functions**

//The old way

function greet ()

{

console.log ('Hello');

}

var greet1 = function(){

console.log ('hello');

}

//The ES6 way

var greet2 = () => {

console.log ('hello');

}

C) **Let and Const**

Let — is used to define a local variable limiting their scope to the block .

Const — constant variable whose values cannot be changed.

D) **Code Editors**

1. Visual Studio Code — VSCODE

2. Sublime Text

3. Atom

**Steps To create React JS Project**

1) Install npm and Node

2) npx create-react-app my-app

cd my-app

npm start

**What is a Hook?** A Hook is a special function that lets you “hook into” React features. For example, useState is a Hook that lets you add React state to function components.

**When would I use a Hook?** We use Hook for Component Did Mount

**What is Component Did Mount() ?**

React has a lifecycle which you can monitor and manipulate during its three main phases.

The three phases are: **Mounting**, **Updating**, and **Unmounting**

React has four built-in methods that gets called, in this order, when mounting a component:

**render()**

The render() method is required, and is the method that actually outputs the HTML to the DOM. The render() method is required and will always be called, the others are optional and will be called if you define them.

**componentDidMount()**

The componentDidMount() method is called after the component is rendered.

**Forms in React**

Just like in HTML, React uses forms to allow users to interact with the web page.

Snippet:

class MyForm extends React.Component {

render() {

return (

<form>

<h1>Hello</h1>

<p>Enter your name:</p>

<input

type="text"

/>

</form>

);

}

}

ReactDOM.render(<MyForm />, document.getElementById('root'));

Source w3school

In HTML, form data is usually handled by the DOM.

In React, form data is usually handled by the components.

When the data is handled by the components, all the data is stored in the component state.

You can control changes by adding event handlers in the on Change attribute.

Conditional Rendering

If you do not want to display the h1 element until the user has done any input, you can add if statement.

{ val ?? null ?

     <> <h4 id ="emoji">Smiley Detected :  </h4> <h1>{smiley}</h1> </>: <h4> Please Enter a Word </h4>}

Val ?? null is check for null value ES2020 introduced the nullish coalescing **operator** denoted by the **double question marks** ( ?? )

On condition true it will return a statement and on being False it return other statement.

Entry point to Application is APP.js