Q1 Write one page reflective what did you learn about React Hook API during this week

This week, I learned how React Hook API simplifies state management and side effects in functional components, focusing on useState and useEffect. Using useState, I created dynamic components like a click counter and an emoji counter, where the state was updated and reflected in real time. The useEffect hook was particularly useful for handling side effects, such as updating images based on props, and for initializing components. Additionally, I gained a better understanding of passing and utilizing props to make components reusable and dynamic. Overall, these tasks reinforced the importance of modular and efficient component design in React.

Q2. Study the code in EmojeeCounters.js, Please note, You Do not need to submit the full code rather you need to answer the following questions for your this week portfolio

• What is Name of the Component you have created in EmojeeCounters.js

EmojeeCounter

• Identify the line of code that uses the EmojeeCounter in index.js

<EmojeeCounter pic='Love' />

<EmojeeCounter pic='sad' />

<EmojeeCounter pic='Like' />

• Declares the states of each of the html elements defined in the EmojeeCounters.js ( identify these lines and explain only those lines )

const [pic, setPic] = useState(Love);  
This initializes the pic state to hold the default image (Love).

const [count, setCount] = useState(0);  
This initializes the count state to track the number of button clicks.

• Lines of codes that are used to associate the event handler used.

<button onClick={ClickHandle}>{count} <img src={pic} alt="" /></button>

• Explain the line : <EmojeeCounter pic=’Love’/>, what is pic=’Love’ means in this line.

The pic='Love' is a prop passed to the EmojeeCounter component. It specifies which emoji image should be displayed by the component. In this case, the Love emoji is used.

• What is useEffect and why you think we have used it in the Component.

useEffect is a React Hook that performs effects in functional components. It is used here to update the pic state based on the pic prop whenever the prop changes. This ensures the correct emoji image is displayed dynamically.

• Explain these line of the codes in functional component EmojeeCounter.js:

return(

<div className=”App”>

<p>{props.pic}<span></span>

<button onClick={ClickHandle}>{count}

<img src={pic} alt””/>

</button>

</p>

</div>

)}

This block of code renders the EmojeeCounter components. Displays the prop value as text, alongside with a button. The button shows the click count and an image that cnages based on a state.