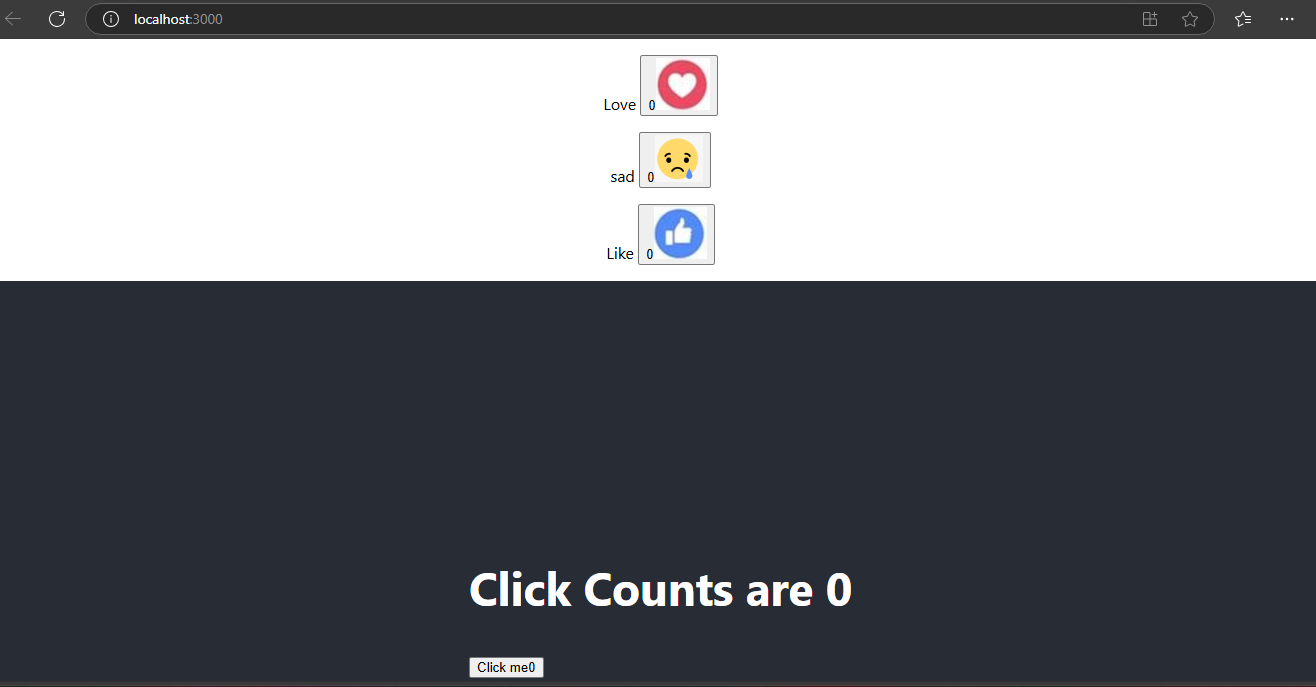
Reflection(week 7):-

In week 7 lab, the lab tutor gave us ideas about HOOKAPI. We should create a functional react component using Hook, develop a simple click counter component, and develop an emoji counter for likes, love, and sad emotions. While using Hook, we set up a new React project with hooks, and again we set up and used React state with useState Hook as guided in the tutorial. After setting up we did some coding for counter.js. In the Emoji counter.js file we create a functional component EmojiCounter and add an image of love, like, or sad emotions by import Love from './Love.png' , import Sad from './sad.png' , import Like from './like.png' from the src folder. At last, we call this functional component in index.js file by using: import EmojeeCounter from './EmojeeCounters' also need to add some code in ReactDOM.render as guided in lab tutorial. At last, we can see the following output in the browser window;



Also, all the code performed in this lab tutorial is in the GitHub link at the top of this Portfolio pdf.

Answer the following :-

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

Ans:- The name of the component created in EmojeeCounters.js is EmojeeCounter.

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

Ans:- <EmojeeCounter pic='Love' />

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

Ans:- const [pic, setPic] = useState(Love);

const [count, setCount] = useState(0);

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

Ans:- <button onClick={ClickHandle}

const ClickHandle = () => {

setCount(count + 1); };

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

Ans:- In the line <EmojeeCounter pic='Love' />, pic='Love' is a prop being passed to the

EmojeeCounter component. The pic prop is a custom property that the parent

component (likely index.js) sends to EmojeeCounter. The value 'Love' is a string, which

indicates that the initial image to be displayed in the EmojeeCounter component will be

the "Love" emoji image (defined as Love.png).

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

Ans:- useEffect is a React hook that is used to perform side effects in function

components. It is similar to lifecycle methods in class components, such as

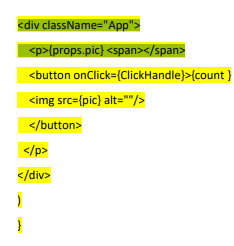
componentDidMount or componentDidUpdate. The code in useEffect will run

after the component renders, and you can specify dependencies to control when

the effect should run.

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

Return(



Ans:- This JSX returns a div container with a class of App that contains a paragraph

displaying the value of the pic prop (like "Love"), a button displaying the count, and

an image displaying the emoji corresponding to the pic state (like Love.png). When

the button is clicked, it triggers the ClickHandle function to increment the count,

and the UI updates to reflect the new count. The image displayed also changes

dynamically based on the value of the pic state.

Q3. Create a code for a Component that takes two HTML one text box and one label. Label will be used to display the images. So it should be like this

If I write “Happy” in the text box the label should show happy face (You can use any image)

If I write “Like” in the text box the label should show Like icon

If I write “sad “ the label should show sad emoji.

Run this component take the screen shot of your newly run component and write a paragraph how did you develop this component

Ans:- I developed this React component by using the useState hook to manage the input text

and image source. I created an input field where users can type "Happy", "Like", or "Sad",

triggering corresponding images to display based on conditional logic. The images

update dynamically as the input changes.