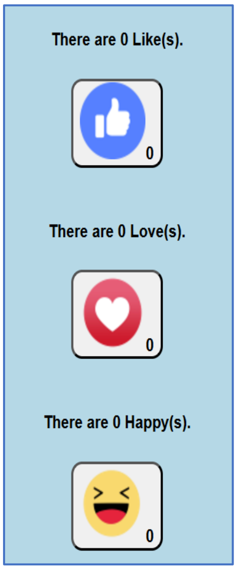
Portfolio Week 9

Task 1:

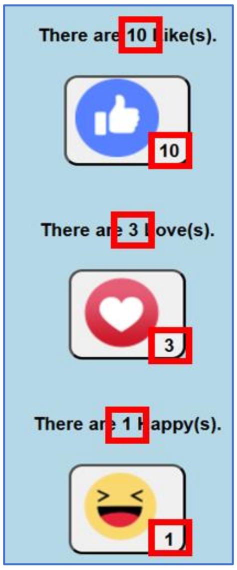
Output:



There are 0 Like(s).

There are 0 Love(s).

There are 0 Happy(s).



There are 10 Like(s).

There are 3 Love(s).

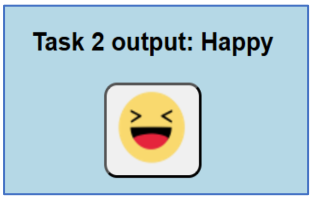
There are 1 Happy(s).

Explanation:

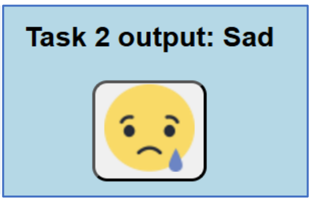
I created a class component that, via the constructor, initialises props and states with the desired default values. The state of ‘number’ is set to 0 and ‘pic’ property is set to null. A set of conditionals within the constructor then check the value of ‘type’ and compares it to a string that matches the imported emoji pictures. If a match is found, set the ‘pic’ to that particular image. An increment method updates the number’s state using ‘.setState’. The increment method is called on a button click.

Task 2:

Output:



Task 2 output: Happy



Task 2 output: Sad

Explanation:

I created a class component similar to the first task, except the managed state is the emoji image itself. Default state of ‘pic’ is set to ‘Happy’. A toggle method updates the ‘pic’ state using ‘.setState’. Conditionals compare the state of the ‘pic’ and set the new state to the opposite image (e.g. happy to sad or sad to happy). The toggle method is called on a button click.

Reflection:

I learned how to implement class components, their props, and their states. Although the results achieved between class and functional components are the same, how they are implemented differs significantly. Class components require inheriting from the React library in order to use its various methods. One such example is the render method, which is absent from functional components. Class components also have constructors for the initialisation of states and props. States are managed differently as class components managed them with methods from the React library, such as setState, whereas functional components managed them through the Hook API library. ‘this’ is also often needed with class components so as to remove ambiguity by directly stating that particular class instance. Overall, class components appear to be harder to implement and require more code to achieve the same results.