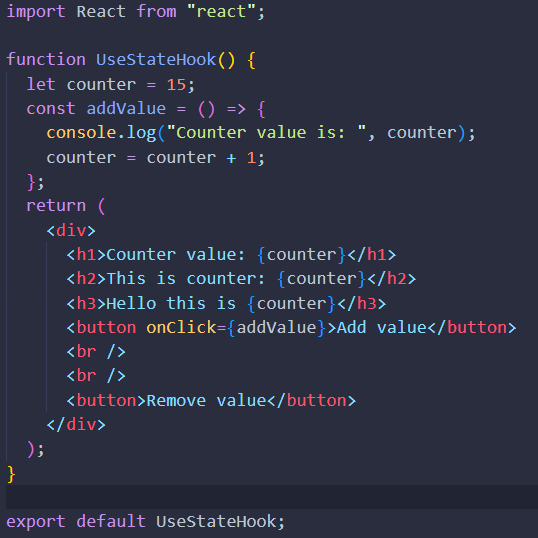
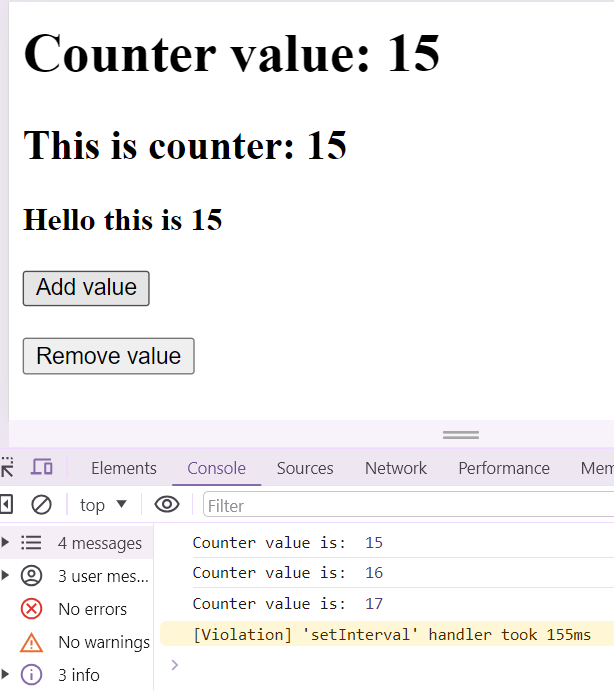
**Hooks in ReactJS**

Problem Statement

Here counter variable is at multiple places and if we want to update its value using javascript then we have to make a lot of variables and then take them using docummet.querySelector().

So, to make this complicated thing easy react give us a hook and say that UI updation ka kaam mera hoga.

Jaha Jaha phr ye counter hai mai har jagha usse update kr dunga.

UI Updation ko react control krta hai.

It can be done by react using useState Hook.

import React from "react";

iss line ko nhi bhi likhunga toh bhi chlega kyuki babel isko handle kr lega.

**useState Hook**

**useState**is React Hook that allows you to add state to a [functional component](https://blog.logrocket.com/fundamentals-functional-programming-react/). It returns an array with two values: the current state and a function to update it. The Hook takes an initial state value as an argument and returns an updated state value whenever the setter function is called. It can be used like this:

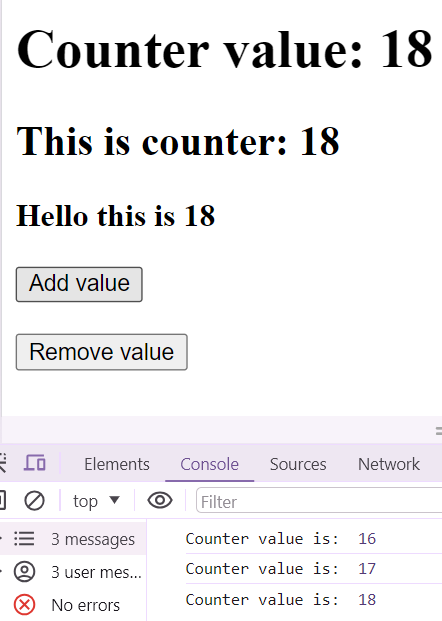
**const [state, setState] = useState(initialValue);**

Here, the **initialValue** is the value you want to start with, and state is the current state value that can be used in your component. The **setState** function can be used to update the state, triggering a re-render of your component.

**useState** can store any type of value, whereas the state in a class component is limited to being an object. This includes primitive data types like string, number, and Boolean, as well as complex data types such as array, object, and function. It can even cover custom data types like class instances.

We can import useState Hook like this

Import { useState } from “react”;

**Example of useState Hook**

Now we can clearly see that using this hook we have update the variable value in all the places where the variable is.