Q.4. List down and explain the 3 ways in which useEffect() hook can be used, with relevant examples.

useEffect hook is an extremely powerful a versatile tool, allowing you to even create your own, custom hooks.

Hooks are available for functional components. Let’s introduce a simple useEffect to our functional component:

```
import React, { useEffect } from "react";

function SimpleUseEffect() {

useEffect(() => {
 alert("Component Rendered")
 });

return (
 <div>
 <b>A Simple use of useEffect...</b>
 </div>
 )
}
```

In the code above, we used the useEffect hook. It takes a function as input, which is executed after the initial rendering, as well as re-rendering, of the component. After each rendering, one the DOM has been updated and the function passed to useEffect is invoked. In the above scenario, the component gives an alert after the initial rendering of the component.