Assignment Title: Understanding React JS State with Functional Components

Objective: The objective of this assignment is to help you understand the concept of state in React JS with functional components and how to work with it. You will learn how to create and manage state in a React functional component and update it when needed.

Instructions:

1. Create a new React JS project using your preferred development environment and setup.
2. Create a new functional component named "Counter" that displays a count value and two buttons to increment and decrement the count.
3. Implement the counter logic using React state with the "useState" hook.
4. When the increment or decrement button is clicked, update the count value in the component state accordingly.
5. Ensure that the count value displayed in the component is always in sync with the state value.
6. Add a button to reset the count to 0.
7. Ensure that the reset button updates the component state and resets the count value displayed in the component.
8. Style the component using CSS to make it look presentable.

Requirements:

1. Use React functional components to implement the Counter component with the "useState" hook for state management.
2. Use the "setState" method to update the component state.
3. Use inline styles or CSS modules to style the component.
4. Ensure that the component is fully functional and the count value is updated correctly.
5. Submit the completed project as a ZIP file, including all project files.

Bonus:

1. Add validation to prevent the count value from going below 0.
2. Implement a "step" feature to allow users to increment or decrement the count by a specified amount.

Tips:

1. Use console.log statements to debug and troubleshoot issues.

Here's a template code for the Counter component without any business logic:

```
import React, { useState } from 'react';
import 'bootstrap/dist/css/bootstrap.min.css';

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

  const handleIncrement = () => {
```

```
  // TODO: Implement increment logic
 };

 const handleDecrement = () => {
  // TODO: Implement decrement logic
 };

 const handleReset = () => {
  // TODO: Implement reset logic
 };

 return (
  <div >
   <h2>Counter</h2>
   <div>{count}</div>
   <div>
    <button onClick={handleIncrement}>+</button>
    <button onClick={handleDecrement}>-</button>
    <button onClick={handleReset}>Reset</button>
   </div>
  </div>
 );
}
export default Counter;
```

Sample output