Que):Create a React app with a component called Counter, which displays a button and a counter value. When the button is clicked, the counter should increment by one using the useState hook. For example, A functional component called Counter. It uses the useState hook to manage the state of the counter. The count state variable is initialized to 0. When the button is clicked, the handleIncrement function updates the count state using the setCount function. It uses the functional update form of setState, providing a callback function that receives the previous state (prevCount) and returns the updated state (in this case, prevCount + 1).

ANS:

Counter.js

```
import React, { useState } from 'react';
const Counter = () => {
  // Initialize state using the useState hook
  const [count, setCount] = useState(0);

  // Function to handle incrementing the counter
  const handleIncrement = () => {
    // Use the functional update form of setState with the previous state (prevCount)
    setCount((prevCount) => prevCount + 1);
  };
  return (
    <div>
      <h1>Counter Value: {count}</h1>
      <button onClick={handleIncrement}>Increment</button>
    </div>
  );
};
export default Counter;
```

App.js

```
import React from 'react';

import Counter from './Counter';

function App() {

    return (

        <div>

            <Counter />

        </div>

    );

}

export default App;
```