useState Hook

Question 1: 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).

Question 2: Create a React app with a component called Toggle, which displays a button to toggle the visibility of a message using the useState hook.

For example, a functional component called Toggle. It uses the useState hook to manage the state of the visibility of the message. The showMessage state variable is initialized to false. When the button is clicked, the handleToggle function updates the showMessage state using the setShowMessage function, toggling its value between true and false. The message is displayed conditionally using the logical AND (&&) operator. If showMessage is true, the message is shown; otherwise, it remains hidden.

Question 3: Create a React app with a component called FormInput, which displays an input field and updates its value using the useState hook.

For example, a functional component called FormInput. It uses the useState hook to manage the state of the input field. The inputValue state variable is initialized to an empty string. When the user types in the input field, the handleChange function updates the inputValue state using the setInputValue function, reflecting the changes in the input field value. The current value of the input field is displayed below it using an inline paragraph (<p>) element.