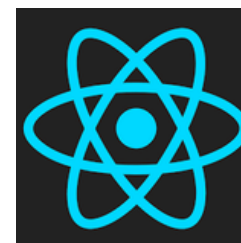


TIMER

INTERVIEW QUESTIONS-96



Follow on



@DUVVURU KISHORE



CODING QUESTION-2

REQUIREMENTS

- Create a React component that displays a **timer** with an initial value of **0**.
- The component should include two buttons: one to **start** the timer and the other to **stop** the timer. Additionally, include a **reset button** to reset the timer.
- Clicking on the **start button** will initiate the timer from 0, **increasing the timer value by 1 for every second**.

- Clicking on the **stop button will pause the timer** .
- If the timer is paused and the start button is clicked again, the timer should continue from the paused value.
- Clicking on the reset button will reset the timer to the initial value of 0.

Before going through the solution we need to understand **setInterval** and **clearInterval**

setInterval is a JavaScript function used to repeatedly execute a given function at specified intervals. It takes two parameters: the function to execute and the time interval in milliseconds.

For example, **setInterval(myFunction, 1000)** will execute **myFunction** every 1000 milliseconds (1 second).

CLEARINTERVAL

clearInterval is a function used to stop the interval set by **setInterval**. It takes the ID returned by **setInterval** as its parameter and clears the interval associated with that ID, effectively stopping the execution of the function.

CODE

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

const Timer = () => {
  const [timer, setTimer] = useState(0);
  const [intervalId, setIntervalId] = useState(null);

  const startTimer = () => {
    if (!intervalId) {
      const id = setInterval(() => {
        setTimer((prevTimer) => prevTimer + 1);
      }, 1000);
      setIntervalId(id);
    }
  };

  const stopTimer = () => {
    clearInterval(intervalId);
    setIntervalId(null);
  };

  const resetTimer = () => {
    clearInterval(intervalId);
    setIntervalId(null);
    setTimer(0);
  };
};
```

```
return (  
  <div>  
    <h2>Timer: {timer}</h2>  
    <button onClick={startTimer}>Start</button>  
    <button onClick={stopTimer}>Stop</button>  
    <button onClick={resetTimer}>Reset</button>  
  </div>  
);  
};  
  
export default Timer;
```

THE CODE IN THE COMMENTS

EXPLANATION

State Management: The component uses the `useState` hook to manage two states:

Timer: Holds the current value of the timer.

intervalId: Holds the ID returned by `setInterval`, allowing us to clear the interval when needed.

Timer Control Functions:

startTimer: Starts the timer by setting up an interval that **increments the timer value every second using setInterval**. It checks if `intervalId` is null before starting the timer to avoid starting multiple intervals.

stopTimer: Stops the timer by **clearing the interval using clearInterval** and setting `intervalId` to null.

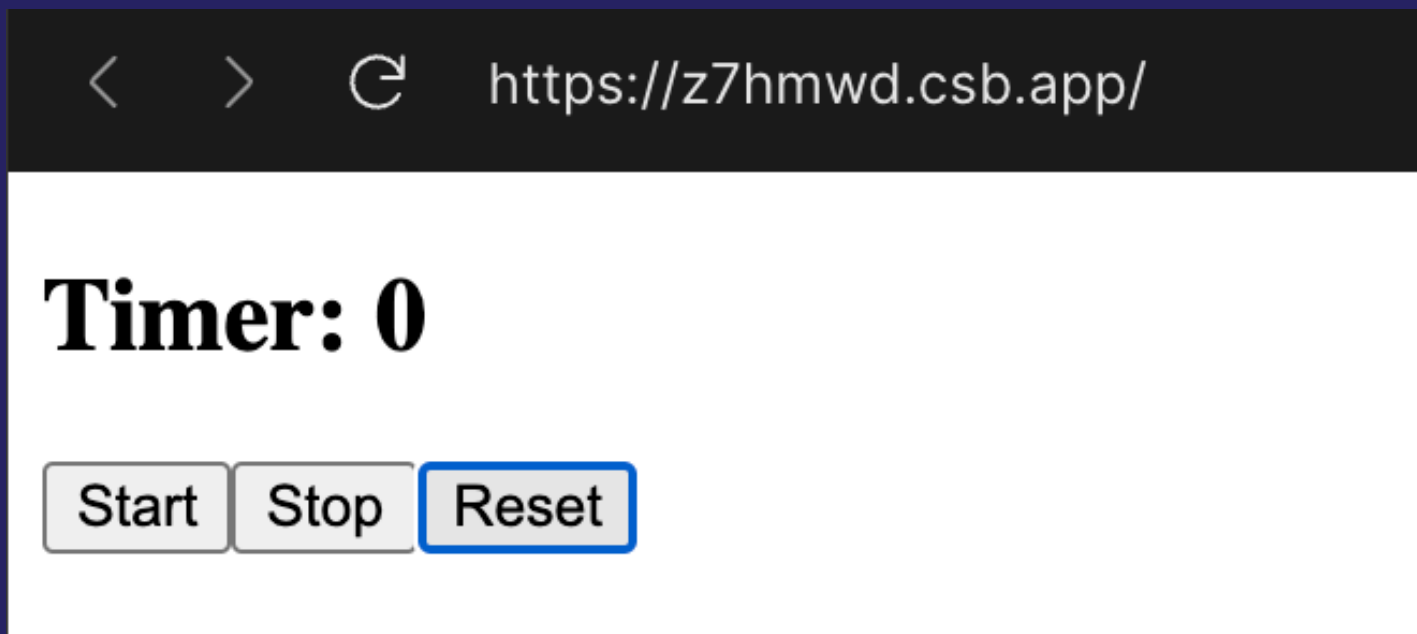
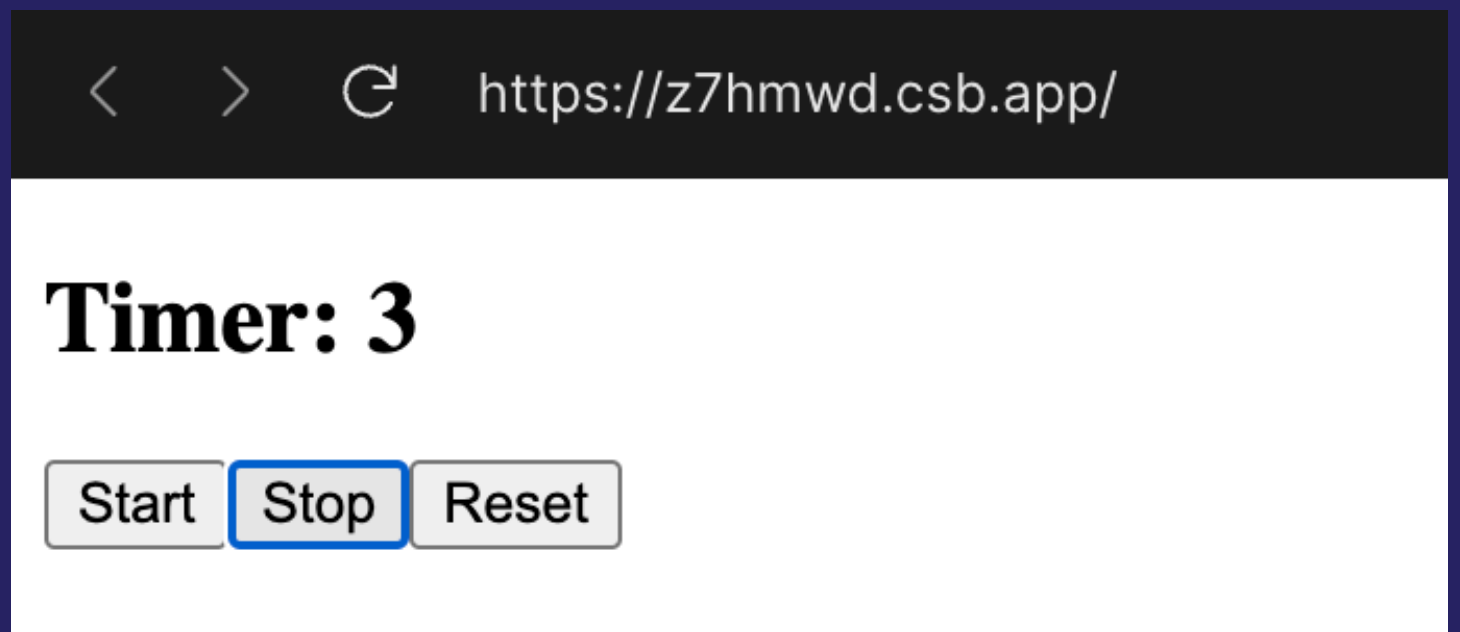
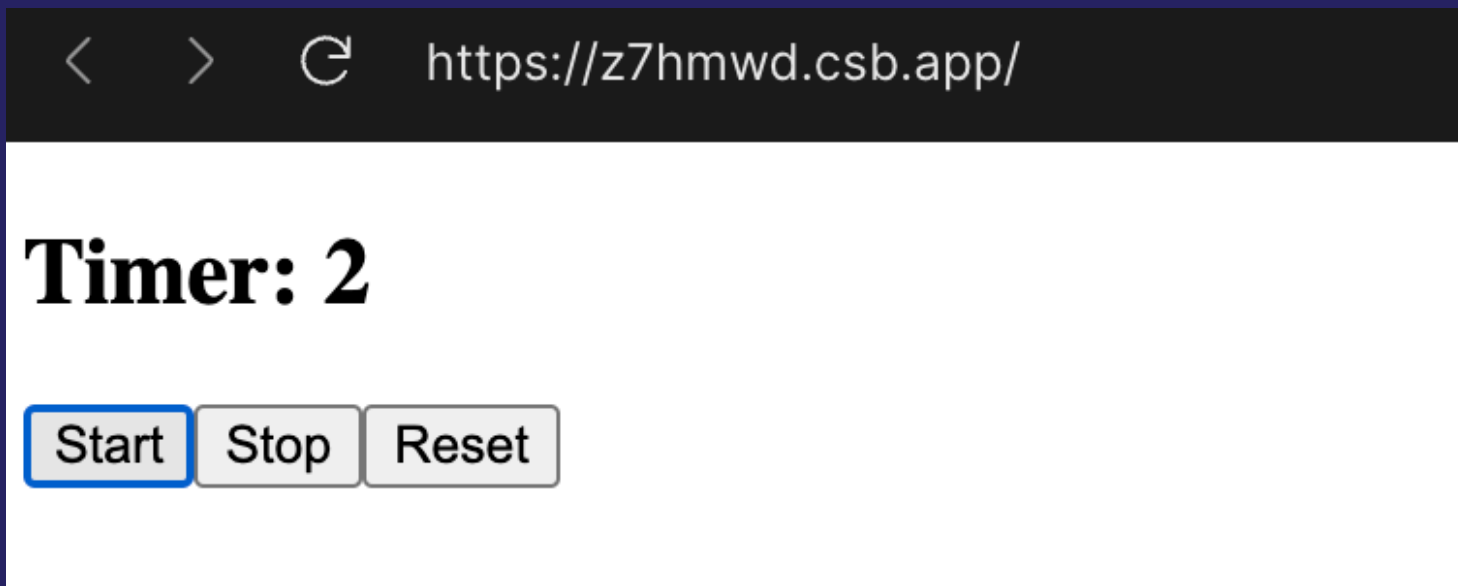
resetTimer: Resets the timer by clearing the interval, setting `intervalId` to null, and resetting the timer state to 0.

Rendering: The component renders the current timer value (timer) along with three buttons:

Start: Clicking this button starts the timer.

Stop: Clicking this button stops the timer.

Reset: Clicking this button resets the timer to 0.





Follow on  **@Duvvuru Kishore**

