

LAB -8

Name -Utkarsh Raj

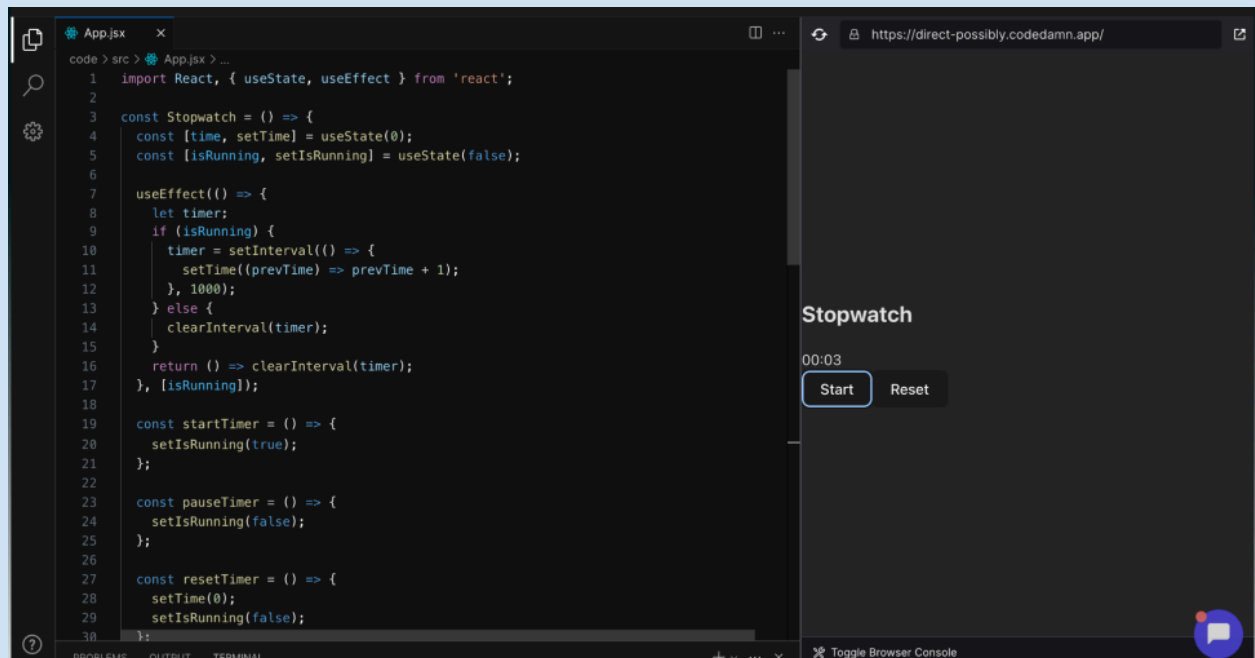
Roll - 22CS2024

Q1. Develop a currency converter application that allows users to input an amount in one currency and convert it to another. For the sake of this challenge, you can use a hard-coded exchange rate. Take advantage of React state and event handlers to manage the input and conversion calculations.

```
1 import React, { useState } from 'react';
2
3 const CurrencyConverter = () => {
4   const [amount, setAmount] = useState('');
5   const [fromCurrency, setFromCurrency] = useState('USD');
6   const [toCurrency, setToCurrency] = useState('EUR');
7   const exchangeRate = 0.85;
8
9   const handleAmountChange = (e) => {
10     setAmount(e.target.value);
11   };
12
13   const handleFromCurrencyChange = (e) => {
14     setFromCurrency(e.target.value);
15   };
16
17   const handleToCurrencyChange = (e) => {
18     setToCurrency(e.target.value);
19   };
20
21   const convertCurrency = () => {
22     const convertedAmount = parseFloat(amount) * exchangeRate;
23     return convertedAmount.toFixed(2);
24   };
25
26   return (
27     <div>
28       <h2>Currency Converter</h2>
29       <div>
30         <label htmlFor="fromCurrency">From:</label>
```

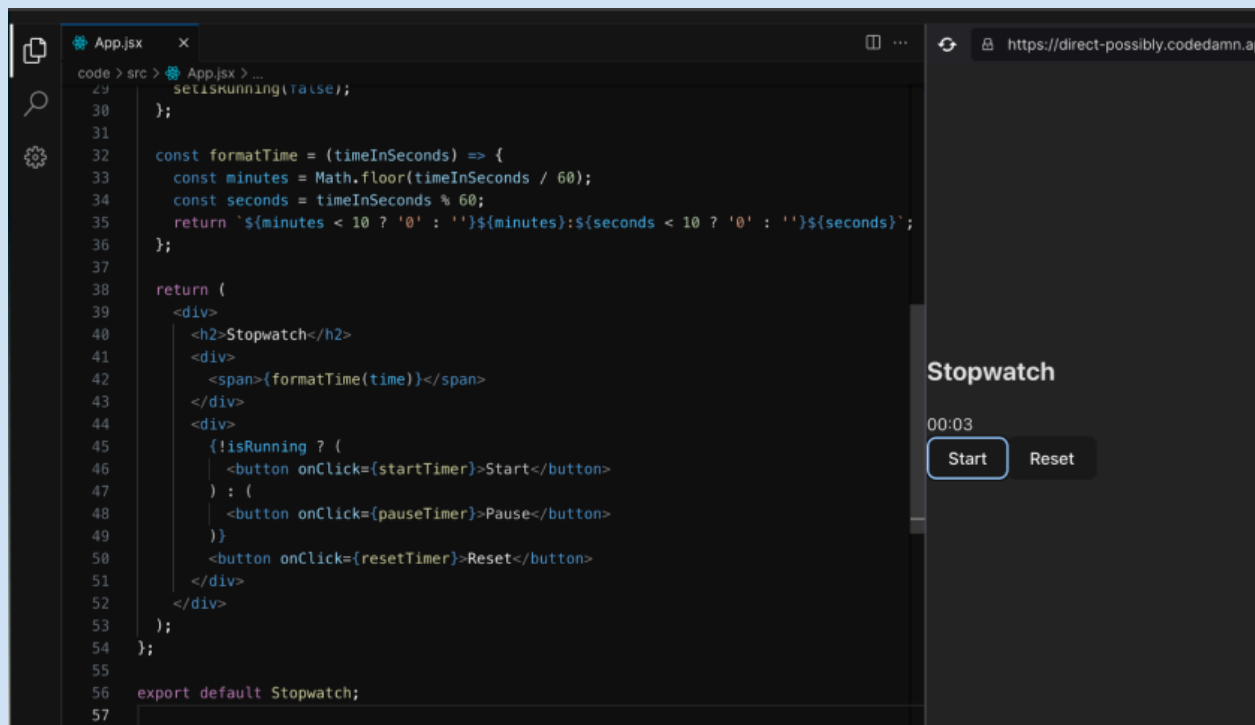
```
23     return convertedAmount.toFixed(2);
24   };
25
26   return (
27     <div>
28       <h2>Currency Converter</h2>
29       <div>
30         <label htmlFor="fromCurrency">From:</label>
31         <select id="fromCurrency" value={fromCurrency} onChange={handleFromCurrencyChange}>
32           <option value="USD">USD</option>
33           <option value="EUR">EUR</option>
34         </select>
35         <input type="number" value={amount} onChange={handleAmountChange} />
36       </div>
37       <div>
38         <label htmlFor="toCurrency">To:</label>
39         <select id="toCurrency" value={toCurrency} onChange={handleToCurrencyChange}>
40           <option value="USD">USD</option>
41           <option value="EUR">EUR</option>
42         </select>
43         <button onClick={convertCurrency}>Convert</button>
44       </div>
45       <span>{convertCurrency()}</span>
46     </div>
47   );
48 };
49
50 export default CurrencyConverter;
```

Q2. Create a stopwatch application through which users can start, pause and reset the timer. Use React state, event handlers and the setTimeout or setInterval functions to manage the timer's state and actions.



The screenshot shows a VS Code editor with a file named `App.jsx`. The code defines a `Stopwatch` component using `useState` and `useEffect`. It includes functions for starting, pausing, and resetting the timer. The browser preview on the right shows the rendered application with the title "Stopwatch", a timer display showing "00:03", and two buttons: "Start" and "Reset".

```
1 import React, { useState, useEffect } from 'react';
2
3 const Stopwatch = () => {
4   const [time, setTime] = useState(0);
5   const [isRunning, setIsRunning] = useState(false);
6
7   useEffect(() => {
8     let timer;
9     if (isRunning) {
10       timer = setInterval(() => {
11         setTime((prevTime) => prevTime + 1);
12       }, 1000);
13     } else {
14       clearInterval(timer);
15     }
16     return () => clearInterval(timer);
17   }, [isRunning]);
18
19   const startTimer = () => {
20     setIsRunning(true);
21   };
22
23   const pauseTimer = () => {
24     setIsRunning(false);
25   };
26
27   const resetTimer = () => {
28     setTime(0);
29     setIsRunning(false);
30   };
31 }
```



The screenshot shows the same VS Code editor with the `App.jsx` file, now including the `formatTime` function and the JSX for the stopwatch UI. The browser preview on the right shows the updated application with the title "Stopwatch", a timer display showing "00:03", and three buttons: "Start", "Pause", and "Reset".

```
29   setIsRunning(false);
30 };
31
32 const formatTime = (timeInSeconds) => {
33   const minutes = Math.floor(timeInSeconds / 60);
34   const seconds = timeInSeconds % 60;
35   return `${minutes < 10 ? '0' : ''}${minutes}:${seconds < 10 ? '0' : ''}${seconds}`;
36 };
37
38 return (
39   <div>
40     <h2>Stopwatch</h2>
41     <div>
42       <span>{formatTime(time)}</span>
43     </div>
44     <div>
45       {!isRunning ? (
46         <button onClick={startTimer}>Start</button>
47       ) : (
48         <button onClick={pauseTimer}>Pause</button>
49       )}
50       <button onClick={resetTimer}>Reset</button>
51     </div>
52   </div>
53 );
54 };
55
56 export default Stopwatch;
57
```