Suraj Agri **Date:-18/11/2024**

Question 1:

output:-

You clicked 5 times.(on every click it increase the number by 1)

Question 2 :

Output:

```
import './App.css';
import React from "react";
function App() {
  const names = ["Brian", "Paul", "Krug", "Halley"];
//  const listItems = names.map((name.index) =>
<li key={index}>{name}</li>); //
  return <ul>{listItems}</ul>;
}
export default App;
```

Question 2 (Question3):-

Output:-

// using setSate directly inside the render() is causing to call state again and again which makes the UI render on infinity loop .

```
import React, { Component } from "react";
 class App extends Component {
  state = {
        counter: 0,
  };

  render() {
        this.setState({ counter: this.state.counter + 1 });
        return <div>Counter: {this.state.counter}</div>;
  }
}
export default App;
```

Solution:-

```
import React, { Component } from "react";
 class App extends Component {
  state = {
          counter: 0,
  };
Count = ()=>{this.setState({ counter: this.state.counter + 1 });}
  render() {
          return (
<div>Counter: {this.state.counter}</div>;
<button onClick={this.count}> count by 1 </button>
)
  }
}
export default App;
```

Question3:-

Output:-

```
import { useState } from "react";
import React from "react";
function Counter() {
  const [count, setCount] = useState(0);
  const counts = () => {
    setCount(count + 1);
  };

  return (
     <div>
        <p>Count: {count}</p>
        <button onClick={counts}>Increment</button>
     </div>
  );
}

export default Counter;
```

or

```
import React, { Component } from "react";
 class App extends Component {
  state = {
        counter: 0,
  };
Count = ()=>{
this.setState({ counter: this.state.counter + 1 });
}
  render() {
        return (
<div>Counter: this.state.counter}</div>;
<button onClick={this.count}> count by 1 </button>
)
  }
}
export default App;
```

Question 4:-

Live-app: Currency-converter

Git-link: currency-converter/currency-converter-app/src/App.js at main · Surajagarwal09/currency-converter

Output:

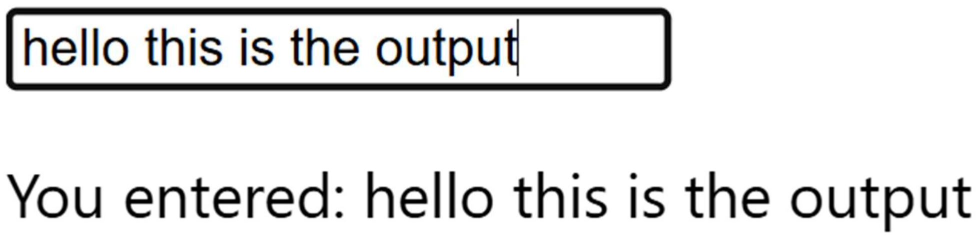
```
import './App.css';

import React, { useState } from 'react';

import { FontAwesomeIcon } from "@fortawesome/react-
fontawesome";

import { faArrowRightArrowLeft } from "@fortawesome/free-
solid-svg-icons";


function App() {

 const [fromcurrency, setFromcurrency] = useState("India");

 const [tocurrency, setTocurrency] = useState("America");

 const [currency, setCurrency] = useState(0);

 const [convertedcurrency, setConvertedcurrency] =
useState(0);


 const handlecurrency = (event) => {

  setCurrency(event.target.value);

 }

 const handlefromcurrency = (event) => {
  setFromcurrency(event.target.value)
 }
 const handletocurrency = (event) => {
  setTocurrency(event.target.value)
 }
 const convertcurrency = (event) => {
  event.preventDefault();
  if (isNaN(currency)) {
   window.alert("Please enter a valid number");
  }
  if (fromcurrency === "India" && tocurrency === "America") {
   setConvertedcurrency(currency * 0.0118488);
  } else if (fromcurrency === "India" && tocurrency === "China") {
   setConvertedcurrency(currency * 0.085773988)
  } else if (fromcurrency === "India" && tocurrency === "Japan") {
   setConvertedcurrency(currency * 1.8249042);
  } else if (fromcurrency == "America" && tocurrency == "India") {
   setConvertedcurrency(currency / 0.0118479);
  } else if (fromcurrency === "America" && tocurrency ===
"China") {
   setConvertedcurrency(currency * 7.2394187);
  } else if (fromcurrency === "America" && tocurrency ===
"Japan") {
      setConvertedcurrency(currency * 154.07395);
    } else if (fromcurrency === "China" && tocurrency == "India") {
      setConvertedcurrency(currency * 11.656517)
    } else if (fromcurrency === "China" && tocurrency ===
"America") {
      setConvertedcurrency(currency * 0.13813264);
    } else if (fromcurrency === "China" && tocurrency === "Japan")
{
      setConvertedcurrency(currency * 21.290936);
    } else if (fromcurrency === "Japan" && tocurrency === "India") {
      setConvertedcurrency(currency * 0.54762059);
    } else if (fromcurrency === "Japan" && tocurrency ===
"America") {
      setConvertedcurrency(currency * 0.006486789);
    } else if (fromcurrency === "Japan" && tocurrency === "China")
{
      setConvertedcurrency(currency * 0.046961805);
    } else if (fromcurrency === tocurrency) {
      setConvertedcurrency(currency);
    } else if (isNaN(currency)) {
      window.alert("Please enter a valid number");
    }
  }
```

```
 return (
  <div className="App">


   <h1 className='h1'>Currency Exchange Rates</h1>


   <div className='contents'>
    <form onSubmit={convertcurrency}>
     <div className='form-group'>
      <select value={fromcurrency}
onChange={handlefromcurrency}>
       <option value="India">India</option>
       <option value="America">America</option>
       <option value="China">China</option>
       <option value="Japan">Japan</option>
      </select>
     <FontAwesomeIcon icon={faArrowRightArrowLeft}
className='icon'/>
      <select value={tocurrency}
onChange={handletocurrency}>
       <option value="Japan">Japan</option>
       <option value="China">China</option>
       <option value="America">America</option>
```

```
          <option value="India">India</option>

         </select>

        </div>  <br />

        <div className='input'>

         <input value={currency} onChange={handlecurrency} />

         <FontAwesomeIcon
icon={faArrowRightArrowLeft}  className='icon'/>

         <input type="number" value={convertedcurrency}
readOnly />

        </div><br />

        <button id='submit' type="submit">Submit</button>

       </form>

     </div>

   </div>

  );

}


export default App;
```

Question 5:-

Output:



```
hello this is the output

You entered: hello this is the output
```

// it takes the value or text that has been entered in the input box and displays the entered or given value or text.

Question 6:-

Output:

Question 7:-

Live -app: Stop-Watch

Gitlink: stop-watch/src/App.js at main · Surajagarwal09/stop-watch

Output:

```
import './App.css';
import { useEffect, useRef, useState } from 'react';
function App() {
  const [watch, setWatch] = useState("00:00:00:000");
  const [isRunning, setIsRunning] = useState(false);
  const [time, setTime] = useState(0);
  const intervalRef = useRef(0);

  useEffect(() => {
    if (isRunning) {
      intervalRef.current = setInterval(() => {
        setTime((prevTime) => prevTime + 10);
      }, 10);
    } else {
      clearInterval(intervalRef.current);
    }
    return () => {
```

```
    clearInterval(intervalRef.current);
  }
}, [isRunning])

function handleStart() {
  setIsRunning(true)
};
function timeData(time) {
  let hours = Math.floor(time / (1000 * 60 * 60));
  let minute = Math.floor(time / (1000 * 60) % 60);
  let seconds = Math.floor(time / (1000) % 60);
  let milliseconds = Math.floor(time % 1000);
  hours = String(hours).padStart(2, "0");
  minute = String(minute).padStart(2, "0");
  seconds = String(seconds).padStart(2, "0");
  milliseconds = String(milliseconds).padStart(3, "0");

  return `${hours}:${minute}:${seconds}:${milliseconds}`;
}

useEffect(() => {
  setWatch(timeData(time));
}, [time])
```

```
  function handleStop() {
    setIsRunning(false)
  }

  function handleReset() {
    setIsRunning(false);
    setTime(0);
    setWatch("00:00:00:000");
  }

  return (
    <div className="App">
      <h1 className='h1'>Stop Watch</h1>
      <div className="stop-watch">
        <div className='flex'>
          <h1 className='watch'>{watch}</h1>
          <div className="buttons">
            <button className="reset-button"
onClick={handleReset}>Reset</button>
            <button className="start-button"
onClick={handleStart}>Start</button>
            <button className="stop-button"
onClick={handleStop}>Stop</button>
```

```
          </div>
        </div>
      </div>
    </div>
  );
}

export default App;
```

<u>Question 8:-</u>

<u>Output:</u>

Hello, *Claire*!

//Because the name Claire is passed down using prop.

Question 9:-

Output:

```
import React, { Component } from "react";
class App extends Component {
  constructor(props) {
    super(props);
    this.state = { name: ""};  }
   handleSubmit = (event) => {
    event.preventDefault();
    console.log("Submitted Name:", this.state.name);
  };
//  handlechange = (Event) => {
    this.setState({ name: Event.target.value })
    console.log(this.state.name);
  }//
  render() {
    return (
      <form onSubmit={this.handleSubmit}>
        <label>
          Name:
```

```
        // <input type="text" value={this.state.name}
onChange={this.handlechange} />//
       </label>
       <button type="submit">Submit</button>
     </form>
   );
 }
}
export default App;
```

<u>Question 10:-</u>

Output:

```
import React, { useState } from "react";
function App() {
 const [counter, setCounter] = useState(0);
 function incrementCounter() {
  setCounter(counter + 1);
 }
 return (
  <div>
   <button onClick={incrementCounter}> Increment
</button>
   <p>Counter: counter}</p>
  </div>
 );
}
export default App;
```

<u>// the issue was the fixed value given in the <p> tag ,which will not show the current value on click.</u>