**App.js**

import './App.css';

import RefFocusInput from './useCallback/RefFocusInput';

function App() {

  return (

    <div className="App">

      <RefFocusInput/>

    </div>

  );

}

export default App;

**RefFocusInput:**

import React, { useEffect, useRef } from "react";

function RefFocusInput() {

  const inputRef = useRef(null);

  useEffect(() => {

    inputRef.current.focus();

  },[]);

  return (

    <div>

      <input ref={inputRef} type="text" />

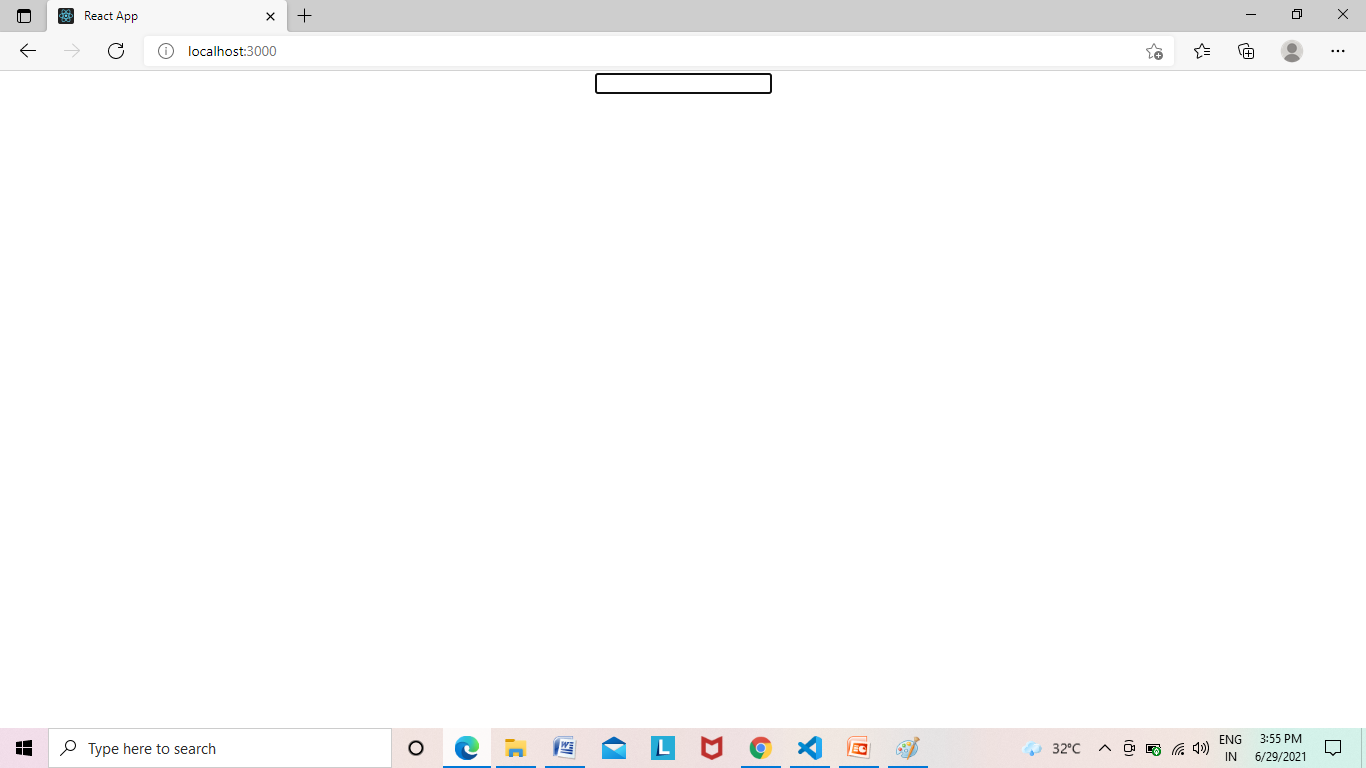
    </div>

  );

}

export default RefFocusInput;

**Output:**

****

**App.js**

import './App.css';

import ClassTimer from './useCallback/ClassTimer';

function App() {

  return (

    <div className="App">

      <ClassTimer/>

    </div>

  );

}

export default App;

**ClassTimer.js**

import React, { Component } from "react";

class ClassTimer extends Component {

  interval;

  constructor(props) {

    super(props);

    this.state = {

      timer: 0,

    };

  }

  componentDidMount() {

    this.interval = setInterval(() => {

      this.setState((prevState) => ({

        timer: prevState.timer + 1,

      }));

    }, 1000);

  }

  render() {

    return <div>Class Timer - {this.state.timer}

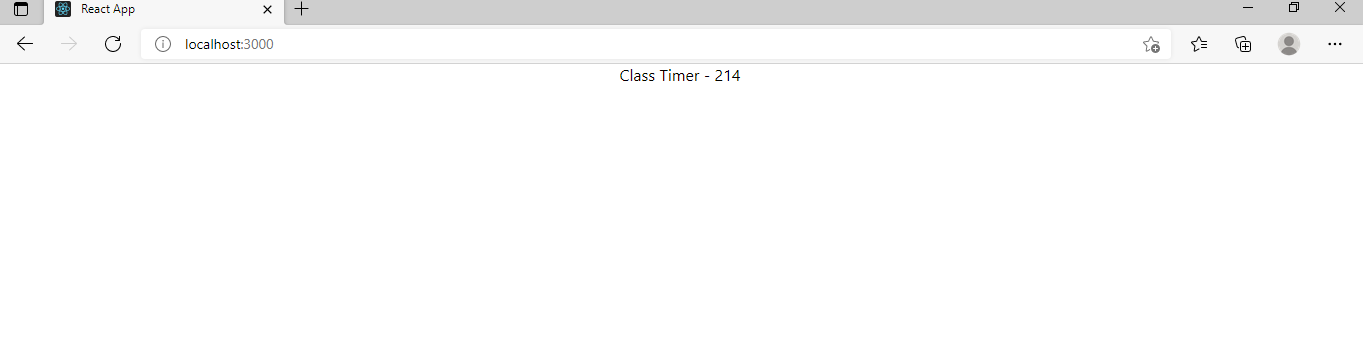
   </div>

  }

}

export default ClassTimer;

**Output**

****

**ClassTimer.js**

import React, { Component } from "react";

class ClassTimer extends Component {

  interval;

  constructor(props) {

    super(props);

    this.state = {

      timer: 0,

    };

  }

  componentDidMount() {

    this.interval = setInterval(() => {

      this.setState((prevState) => ({

        timer: prevState.timer + 1,

      }));

    }, 1000);

  }

componentWillUnmount = () =>{

   clearInterval(this.interval);

}

  render() {

    return <div>Class Timer - {this.state.timer}

    <button onClick={this.componentWillUnmount}>Clear Timer</button>

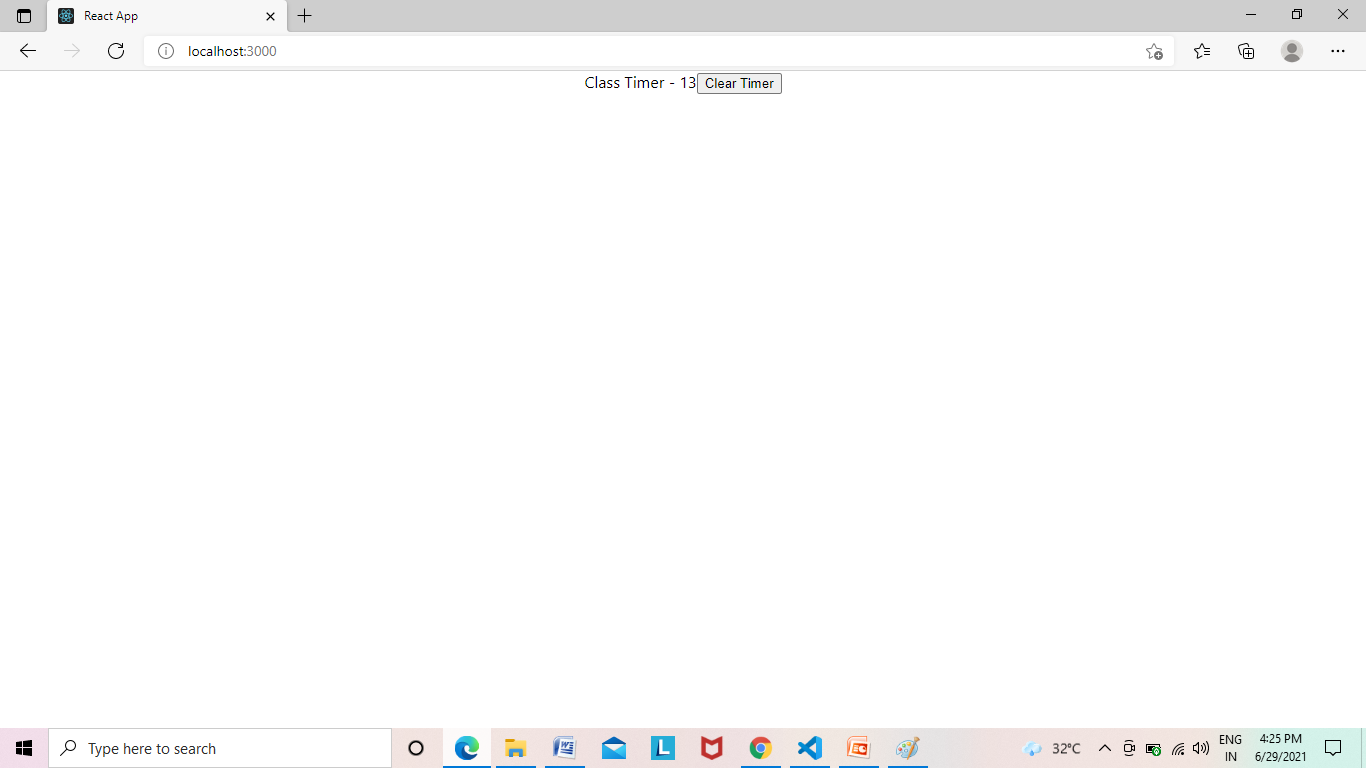
    </div>

  }

}

export default ClassTimer;

**Output:**

****

**ClassTimer.js**

import React, { Component } from "react";

class ClassTimer extends Component {

  interval;

  constructor(props) {

    super(props);

    this.state = {

      timer: 0,

    };

  }

  componentDidMount() {

    this.interval = setInterval(() => {

      this.setState((prevState) => ({

        timer: prevState.timer + 1,

      }));

    }, 1000);

  }

  render() {

    return <div>Class Timer - {this.state.timer}

    <button onClick={()=>clearInterval(this.interval)}>Clear Timer</button>

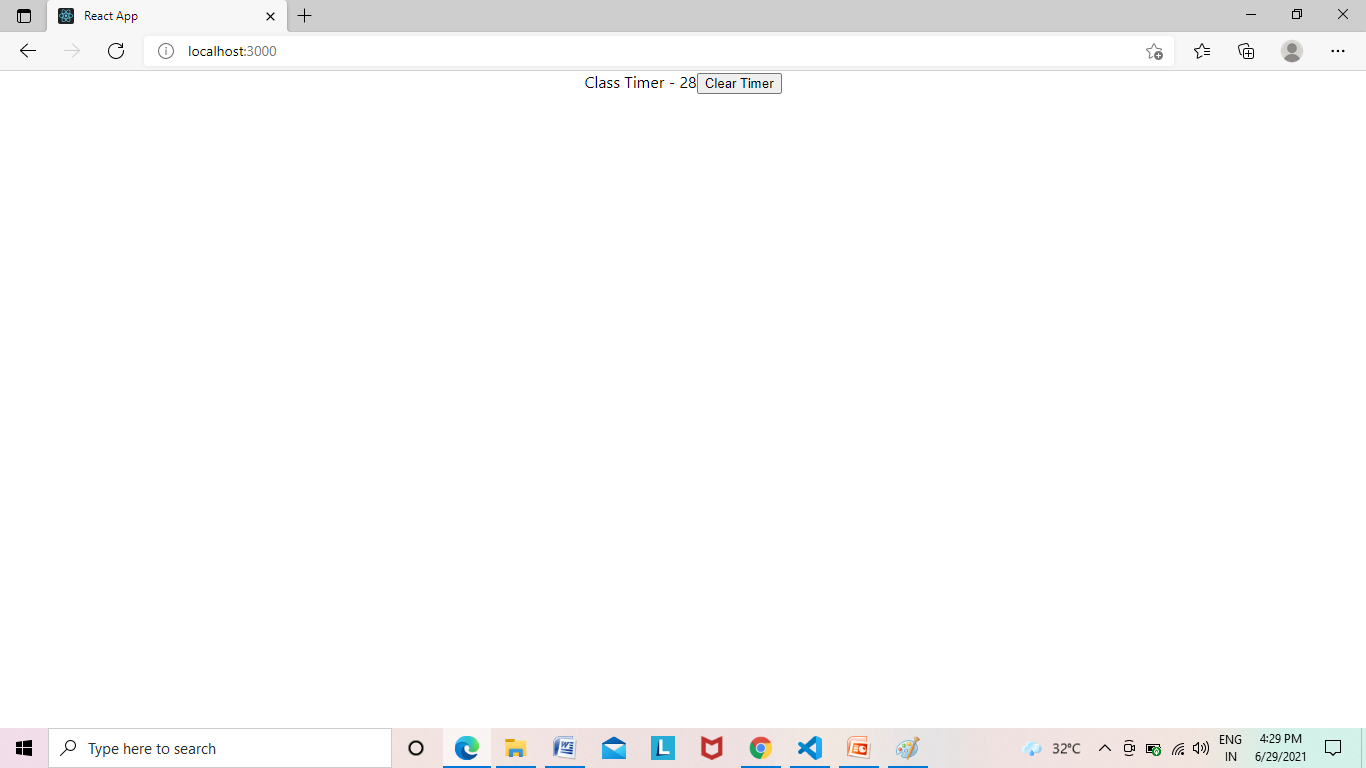
    </div>

  }

}

export default ClassTimer;

**Output:**

****

**App.js**

import './App.css';

import HookFunTimer from './useCallback/HookfunTimer';

function App() {

  return (

    <div className="App">

      <HookFunTimer/>

    </div>

  );

}

export default App;

**HookfunTimer.js**

import React, { useEffect, useRef, useState } from "react";

function HookFunTimer() {

  const [timer, setTimer] = useState(0);

  const intervalRef= useRef();

  useEffect(() => {

    intervalRef.current = setInterval(() => {

      setTimer((prevTimer) => prevTimer + 1);

    }, 1000);

    return () => {

      clearInterval(intervalRef.current);

    };

  }, []);

  return <div>

      Hook timer - {timer}

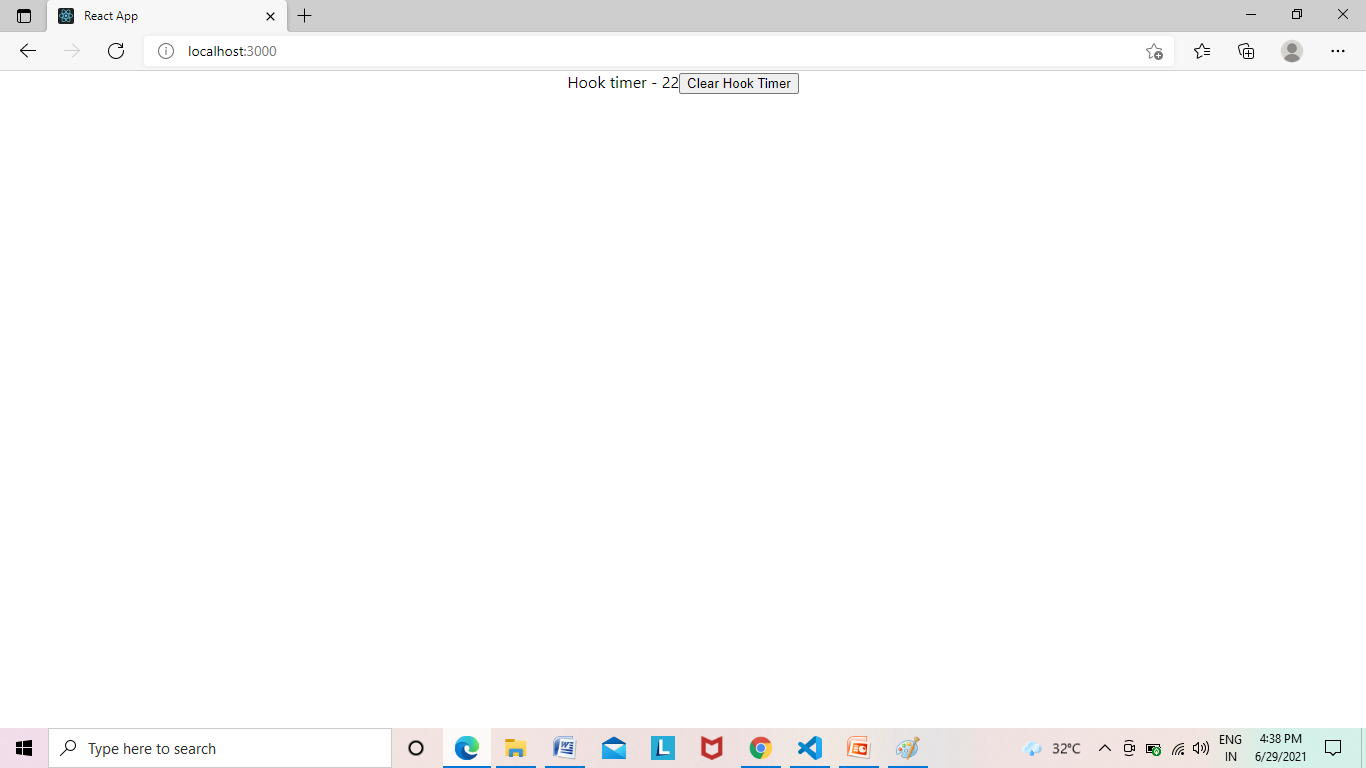
      <button onClick={()=>clearInterval(intervalRef.current)}>Clear Hook Timer</button>

  </div>;

}

export default HookFunTimer;

**Output:**

****