**App.js**

import './App.css';

import UseReducer1 from './Reducer/UseReducer1';

function App() {

  return (

    <div className="App">

      <UseReducer1/>

    </div>

  );

}

export default App;

**UseReducer1.js**

import React,{useReducer} from 'react';

const initialState=0;

const reducer = (state, action) => {

    switch (action) {

      case "increment":

        return state + 1;

      case "decrement":

        return state - 1;

      case "reset":

        return initialState;

      default:

        return state;

    }

  };

function UseReducer1() {

    const [count, dispatch] = useReducer(reducer, initialState);

    return (

      <div>

          <h1>Count : {count}</h1>

        <button onClick={()=>dispatch('increment')}>increment</button>

        <button onClick={()=>dispatch('decrement')}>decrement</button>

        <button onClick={()=>dispatch('reset')}>reset</button>

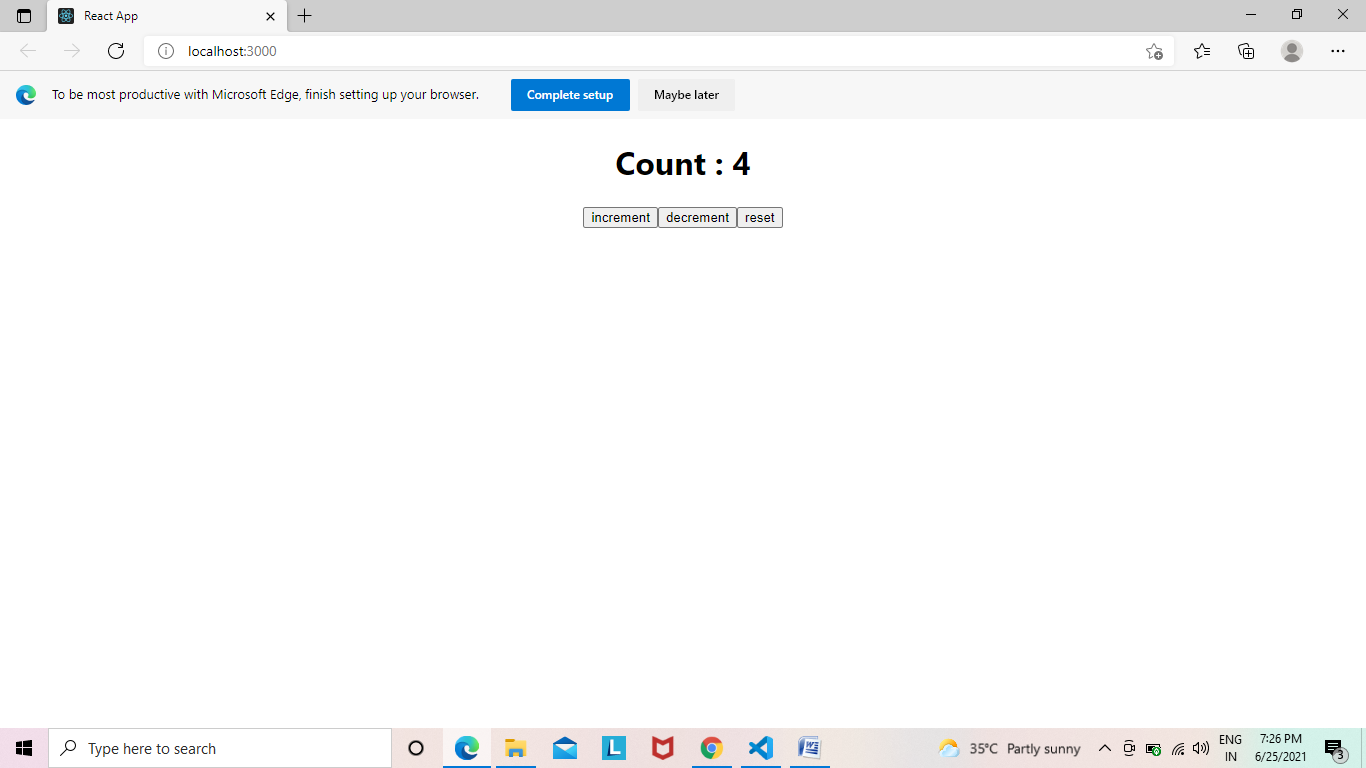
      </div>

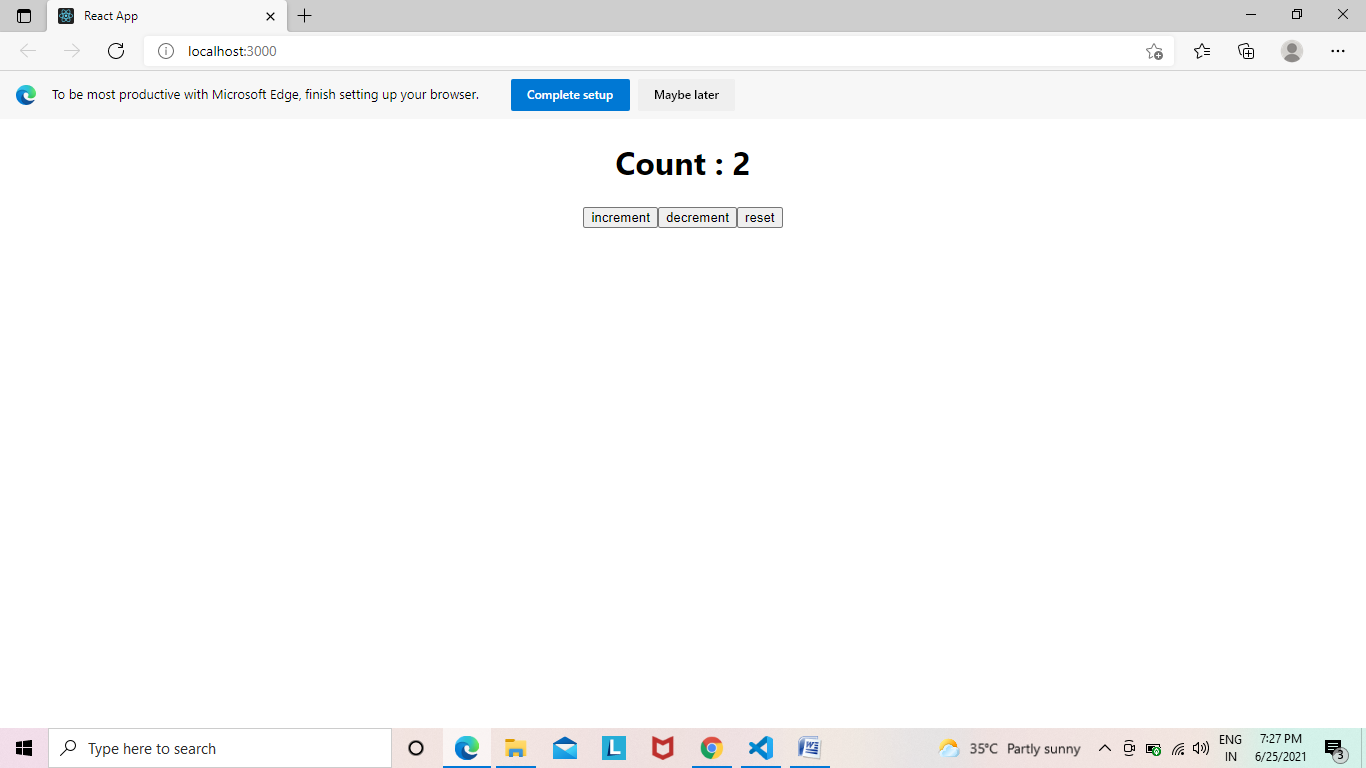
    );

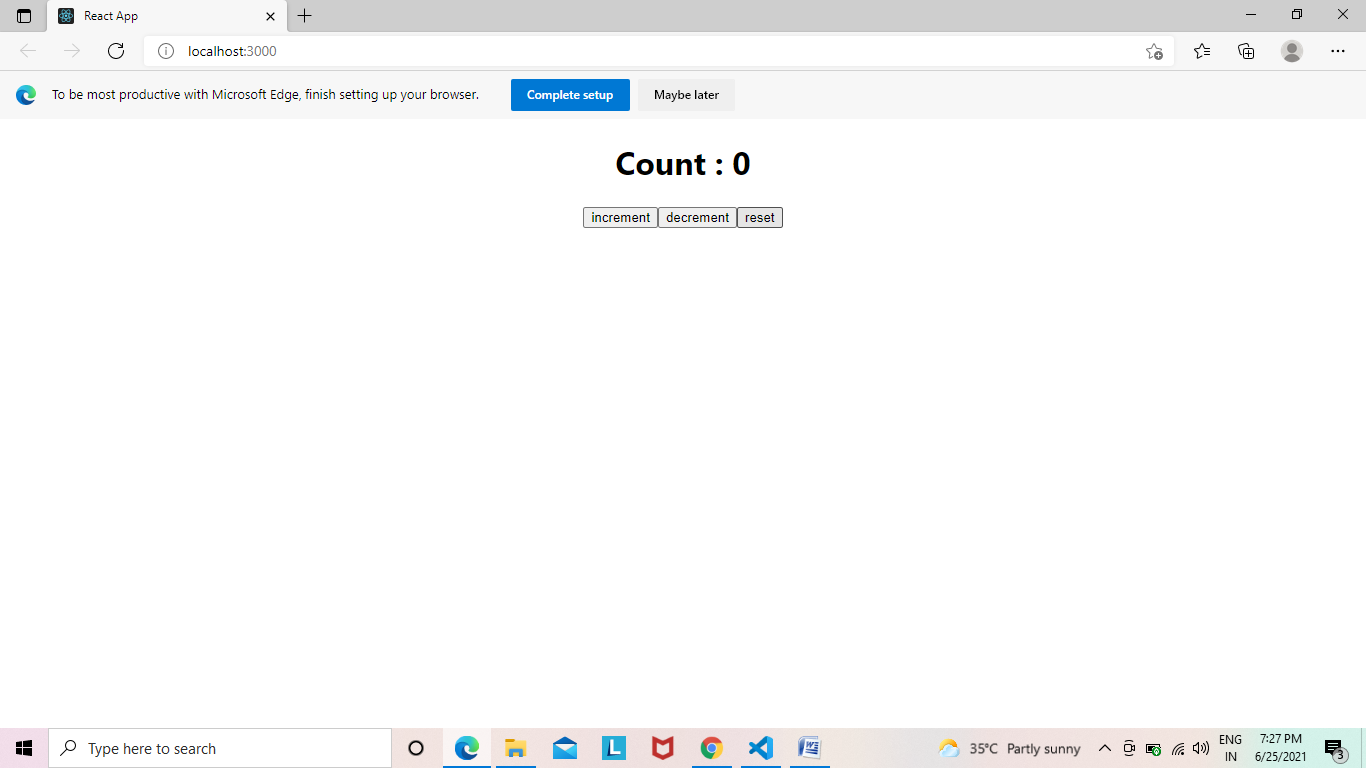
}

export default UseReducer1;

**Output:**

****

****

****

**App.js**

import './App.css';

import UseReducer2 from './Reducer/UseReducer2';

function App() {

  return (

    <div className="App">

      <UseReducer2/>

    </div>

  );

}

export default App;

**UseReducer2.js**

import React, { useReducer } from "react";

const initialState = {

    firstCounter: 0,

};

const reducer = (state, action) => {

    switch (action) {

      case "increment":

        return {

          firstCounter: state.firstCounter + 1,

        };

      case "decrement":

        return {

          firstCounter: state.firstCounter - 1,

        };

      case "increment5":

        return {

          firstCounter: state.firstCounter + 5,

        };

      case "decrement5":

        return {

          firstCounter: state.firstCounter - 5,

        };

      case "reset":

        return initialState;

      default:

        return state;

    }

  };

function UseReducer2() {

  const [count, dispatch] = useReducer(reducer, initialState);

  return (

    <div>

        <h1>Count : {count.firstCounter}</h1>

      <button onClick={()=>dispatch('increment')}>increment</button>

      <button onClick={()=>dispatch('decrement')}>decrement</button>

      <button onClick={()=>dispatch('increment5')}>increment5</button>

      <button onClick={()=>dispatch('decrement5')}>decrement5</button>

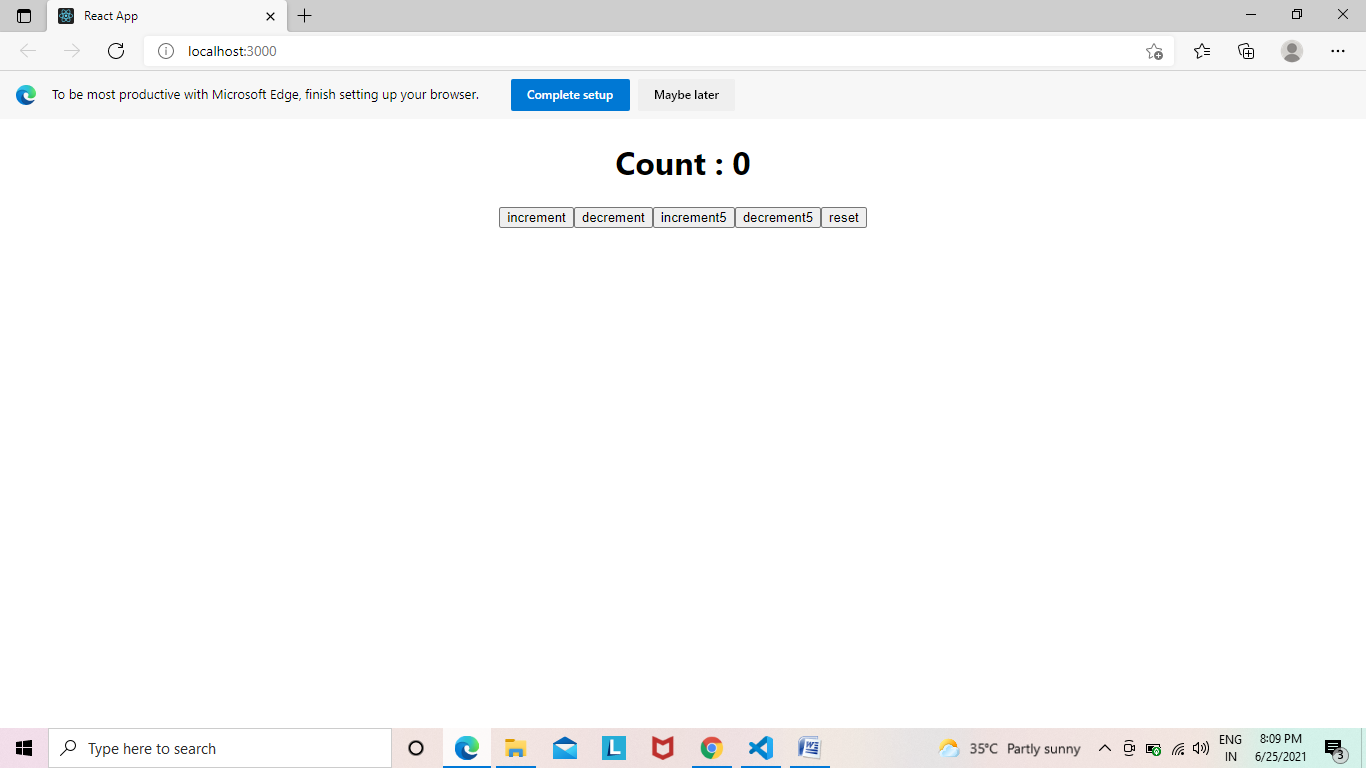
      <button onClick={()=>dispatch('reset')}>reset</button>

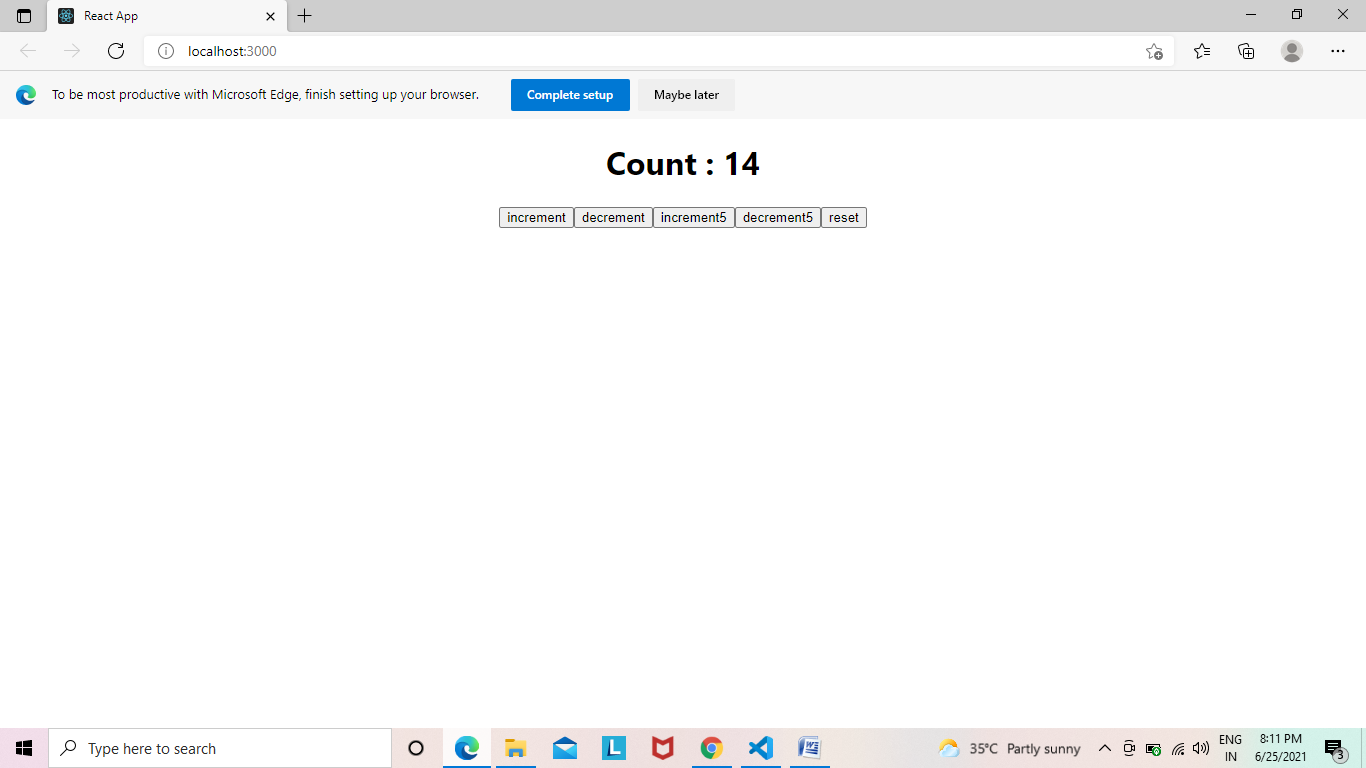
    </div>

  );

}

export default UseReducer2;

****

****

**UseReducer2.js**

import React, { useReducer } from "react";

const initialState = {

    firstCounter: 0,

};

const reducer = (state, action) => {

    switch (action.type) {

      case "increment":

        return {

          firstCounter: state.firstCounter + action.value,

        };

      case "decrement":

        return {

          firstCounter: state.firstCounter - action.value,

        };

      case "increment5":

        return {

          firstCounter: state.firstCounter +action.value,

        };

      case "decrement5":

        return {

          firstCounter: state.firstCounter - action.value,

        };

      case "reset":

        return initialState;

      default:

        return state;

    }

  };

function UseReducer2() {

  const [count, dispatch] = useReducer(reducer, initialState);

  return (

   <div>

        <h1>Count : {count.firstCounter}</h1>

   <button onClick={()=>dispatch({type:"increment",value:1})}>increment</button>

   <button onClick={()=>dispatch({type:"decrement",value:1})}>decrement</button>

   <button onClick={()=>dispatch({ type:"increment5", value:5})}>increment5</button>

      <button onClick={()=>dispatch({ type: "decrement5", value: 5 })}>decrement5</button>

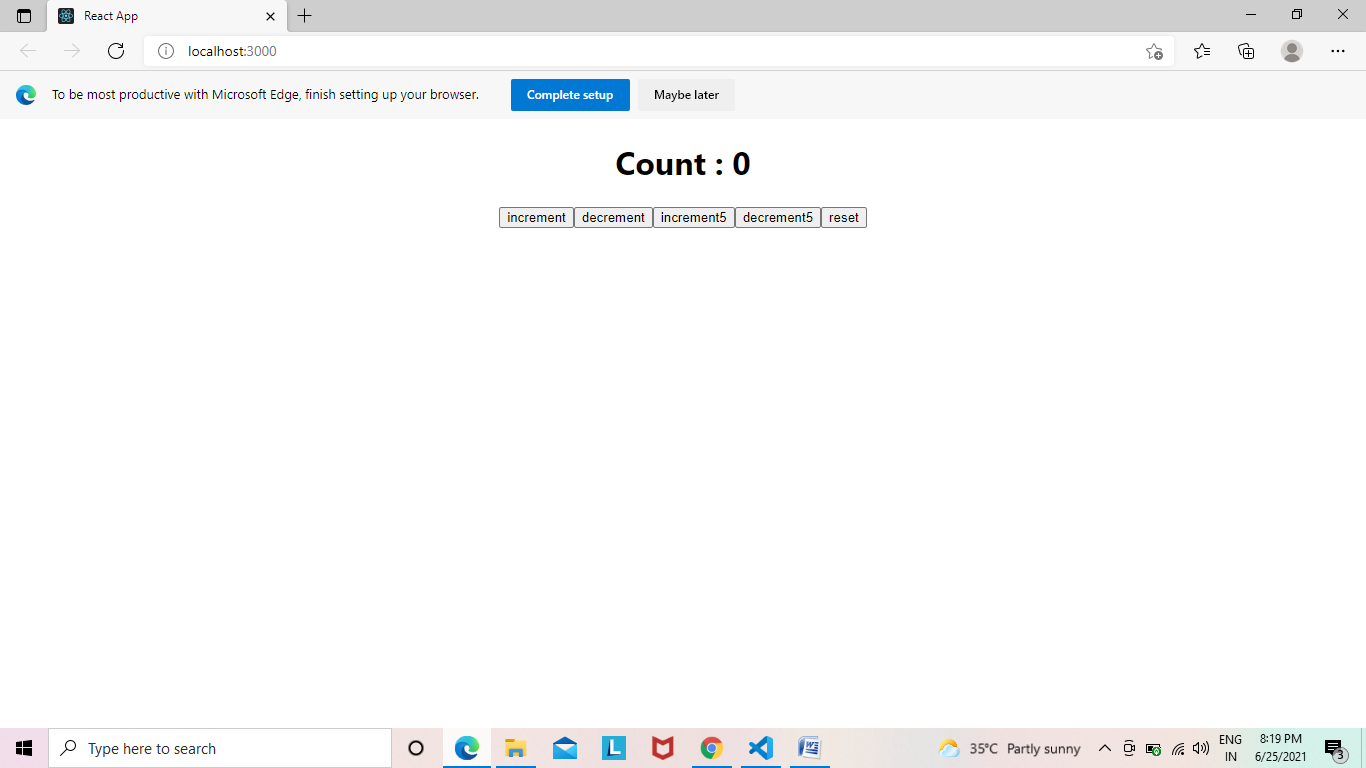
      <button onClick={()=>dispatch({ type: "reset", value: 1 })}>reset</button>

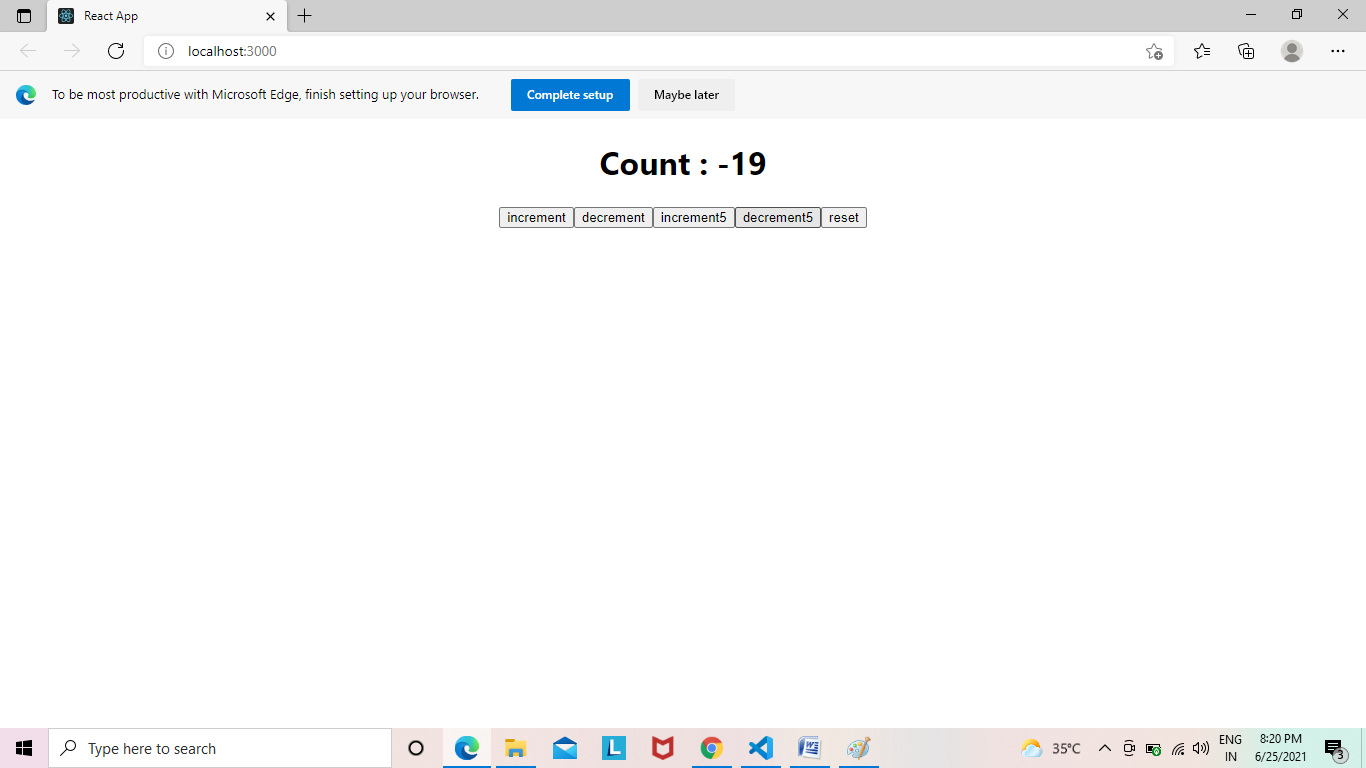
    </div>

  );

}

export default UseReducer2;

****

****

**App.js**

import './App.css';

import UseReducer3 from './Reducer/UseReducer3';

function App() {

  return (

    <div className="App">

      <UseReducer3/>

    </div>

  );

}

export default App;

**UseReducer3.js**

import React, { useReducer } from "react";

const initialState = {

    firstCounter: 0,

    secondCounter: 0,

};

const reducer = (state, action) => {

    switch (action.type) {

      case "increment":

        return {

          firstCounter: state.firstCounter + action.value,

        };

      case "decrement":

        return {

          firstCounter: state.firstCounter - action.value,

        };

      case "increment5":

        return {

          secondCounter: state.secondCounter +action.value,

        };

      case "decrement5":

        return {

          secondCounter: state.secondCounter - action.value,

        };

      case "reset":

        return initialState;

      default:

        return state;

    }

  };

function UseReducer3() {

  const [count, dispatch] = useReducer(reducer, initialState);

  const [countTwo, dispatchtwo] = useReducer(reducer, initialState);

  return (

    <div>

        <h1>Count : {count.firstCounter}</h1>

      <button onClick={()=>dispatch({ type: "increment", value: 1 })}>increment</button>

      <button onClick={()=>dispatch({ type: "decrement", value: 1 })}>decrement</button>

      <button onClick={()=>dispatch({ type: "reset", value: 1 })}>reset</button>

      <h1>CountTwo: {countTwo.secondCounter}</h1>

      <button onClick={()=>dispatchtwo({ type: "increment5", value: 5 })}>increment5</button>

      <button onClick={()=>dispatchtwo({ type: "decrement5", value: 5 })}>decrement5</button>

      <button onClick={()=>dispatchtwo({ type: "reset", value: 1 })}>reset</button>

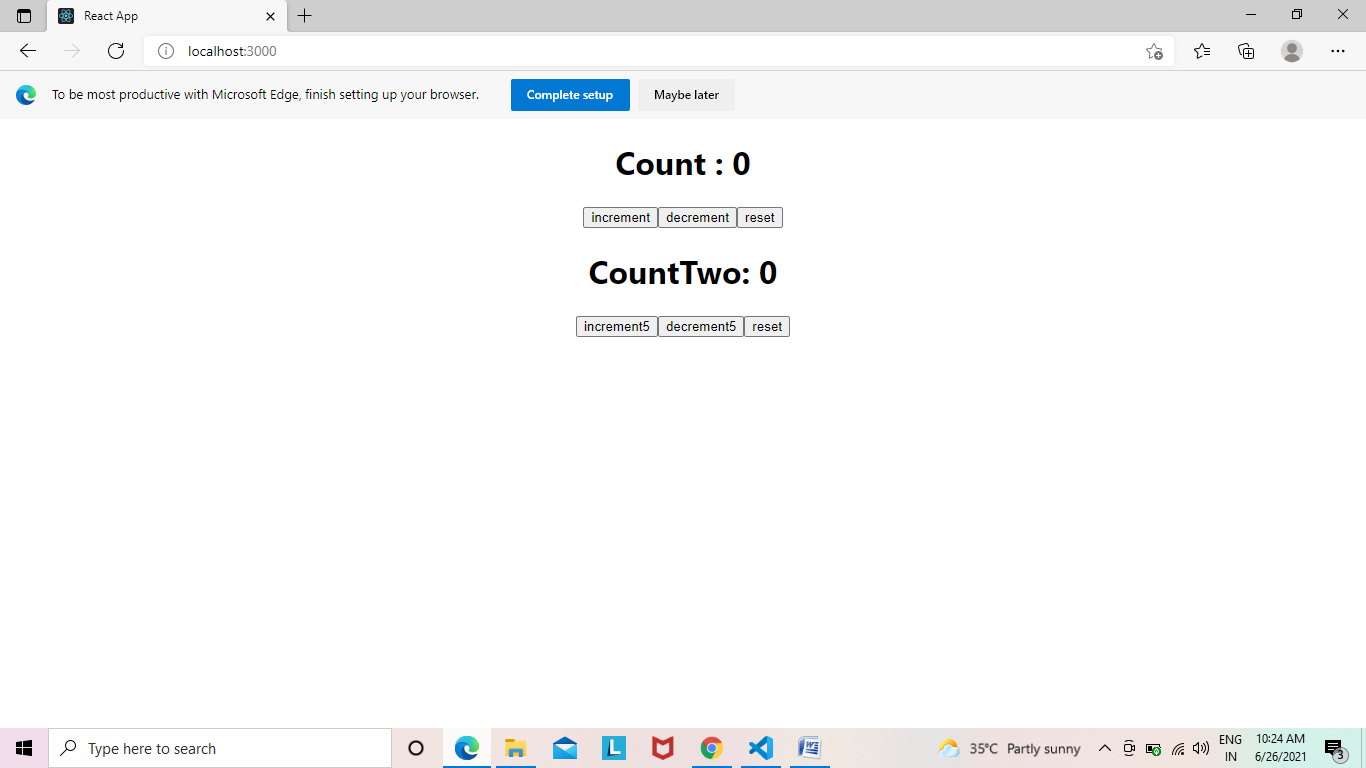
    </div>

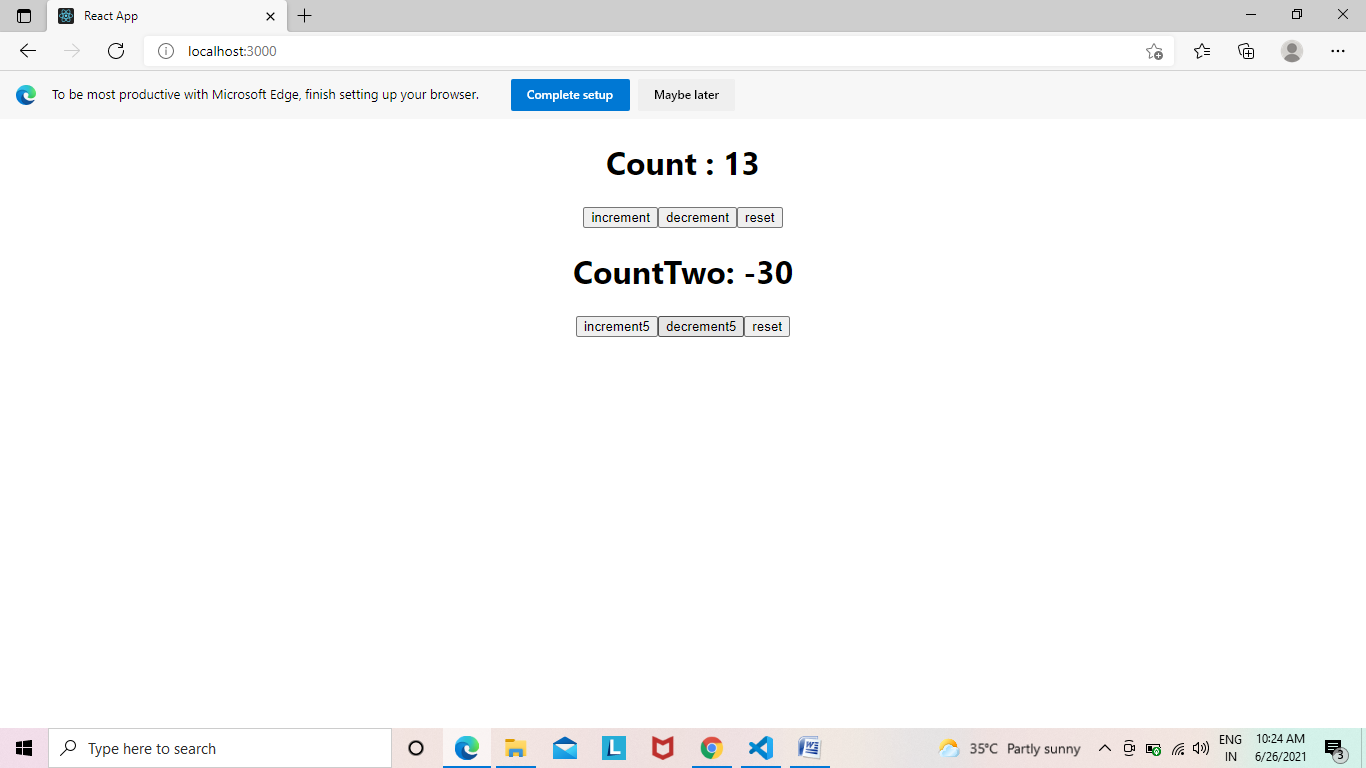
  );

}

export default UseReducer3;

**OutPut:**

****

****

**App.js**

import './App.css';

import UseReducer4 from './Reducer/UseReducer4';

function App() {

  return (

    <div className="App">

      <UseReducer4/>

    </div>

  );

}

export default App;

**UseReducer4.js**

import React, { useReducer } from "react";

const initialState = {

    firstCounter: 0,

    secondCounter: 0,

};

const reducer = (state, action) => {

    switch (action.type) {

      case "increment":

        return {

          firstCounter: state.firstCounter + action.value,

        };

      case "decrement":

        return {

          firstCounter: state.firstCounter - action.value,

        };

      case "increment5":

        return {

          secondCounter: state.secondCounter +action.value,

        };

      case "decrement5":

        return {

          secondCounter: state.secondCounter - action.value,

        };

      case "reset":

        return initialState;

      default:

        return state;

    }

  };

function UseReducer4() {

  const [count, dispatch] = useReducer(reducer, initialState);

  return (

    <div>

        <h1>Count : {count.firstCounter}</h1>

        <h1>CountTwo : {count.secondCounter}</h1>

      <button onClick={()=>dispatch({ type: "increment", value: 1 })}>increment</button>

      <button onClick={()=>dispatch({ type: "decrement", value: 1 })}>decrement</button>

      <button onClick={()=>dispatch({ type: "increment5", value: 5 })}>increment5</button>

      <button onClick={()=>dispatch({ type: "decrement5", value: 5 })}>decrement5</button>

      <button onClick={()=>dispatch({ type: "reset", value: 1 })}>reset</button>

    </div>

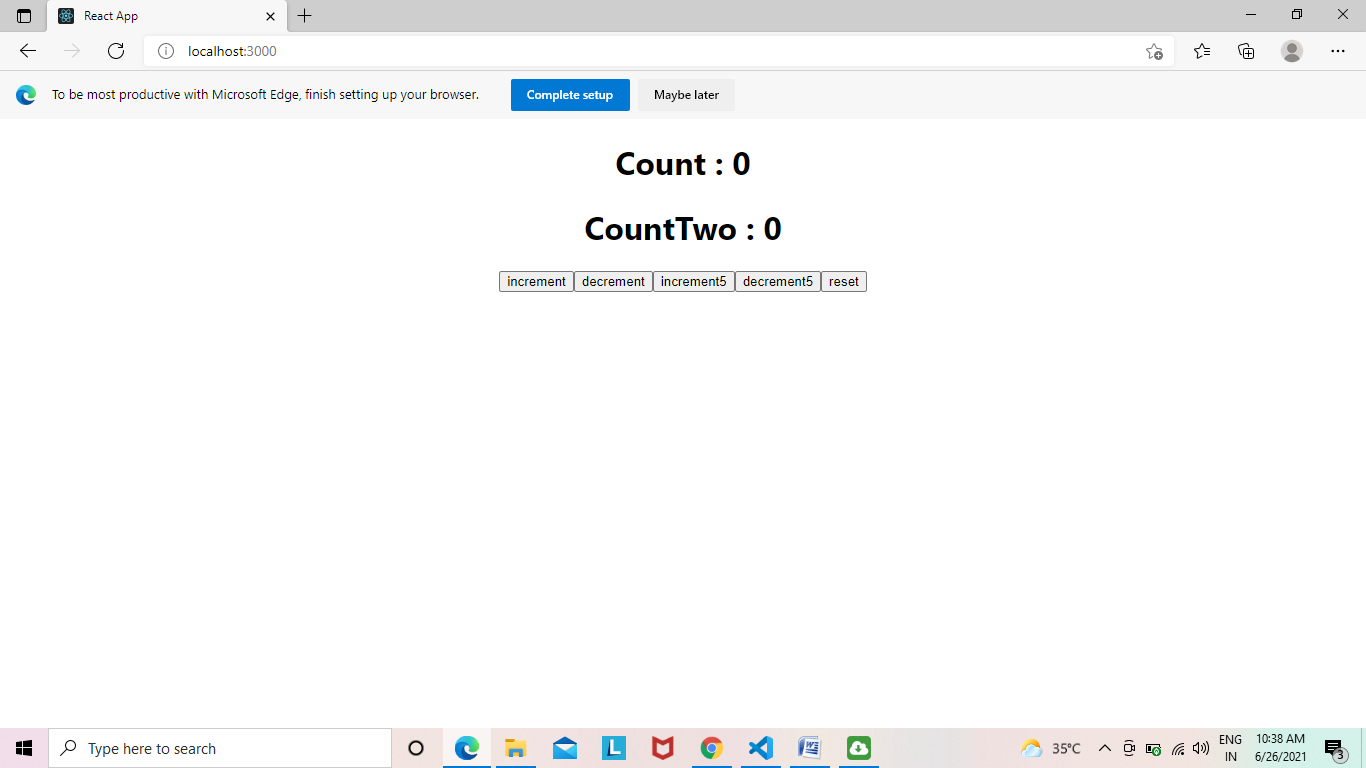
  );

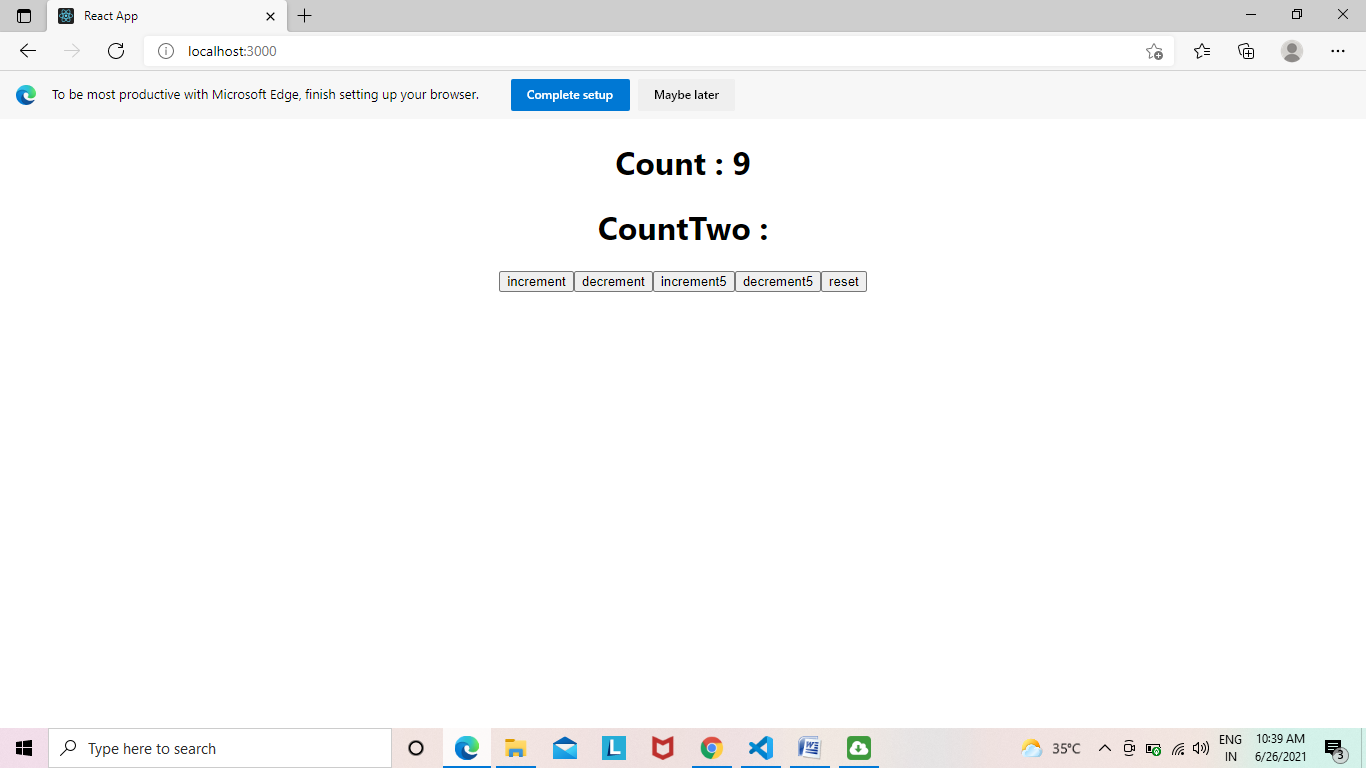
}

export default UseReducer4;

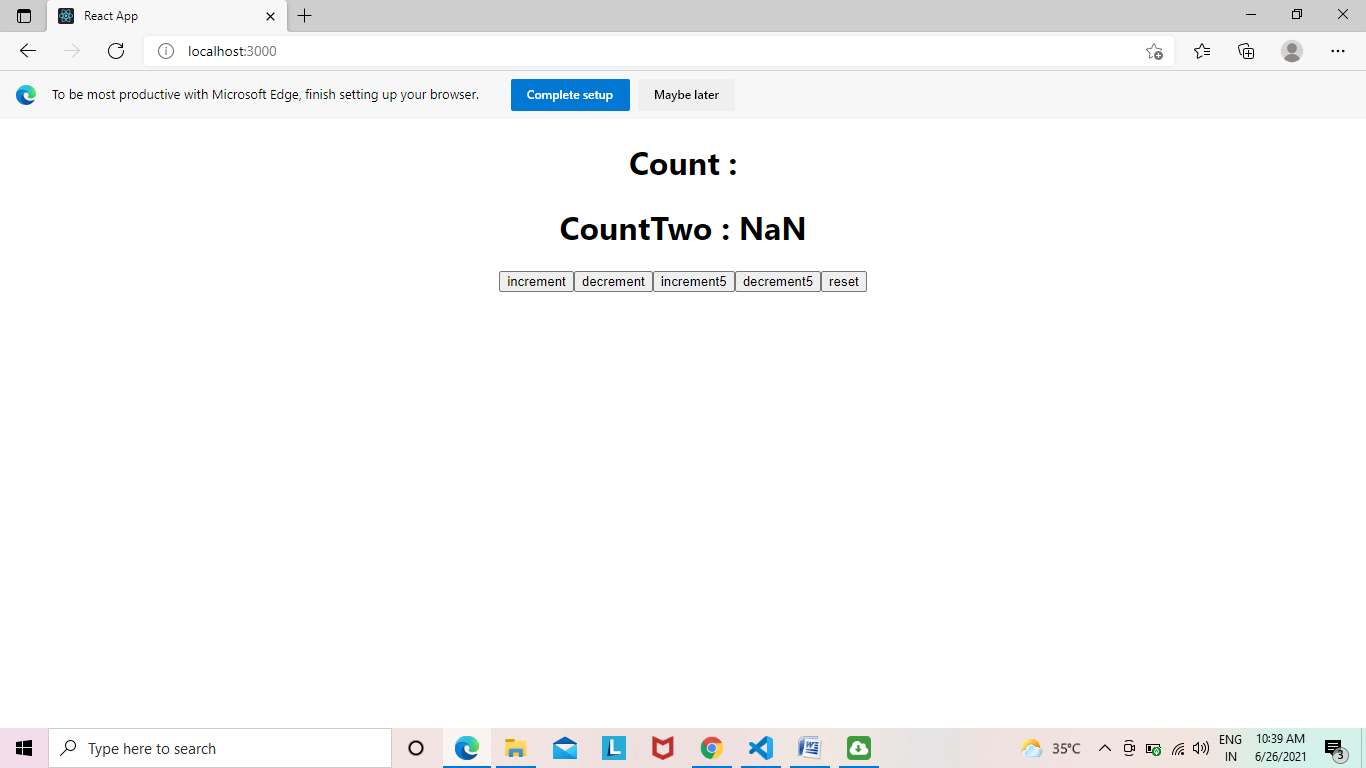
**Output:**

**Initial:**

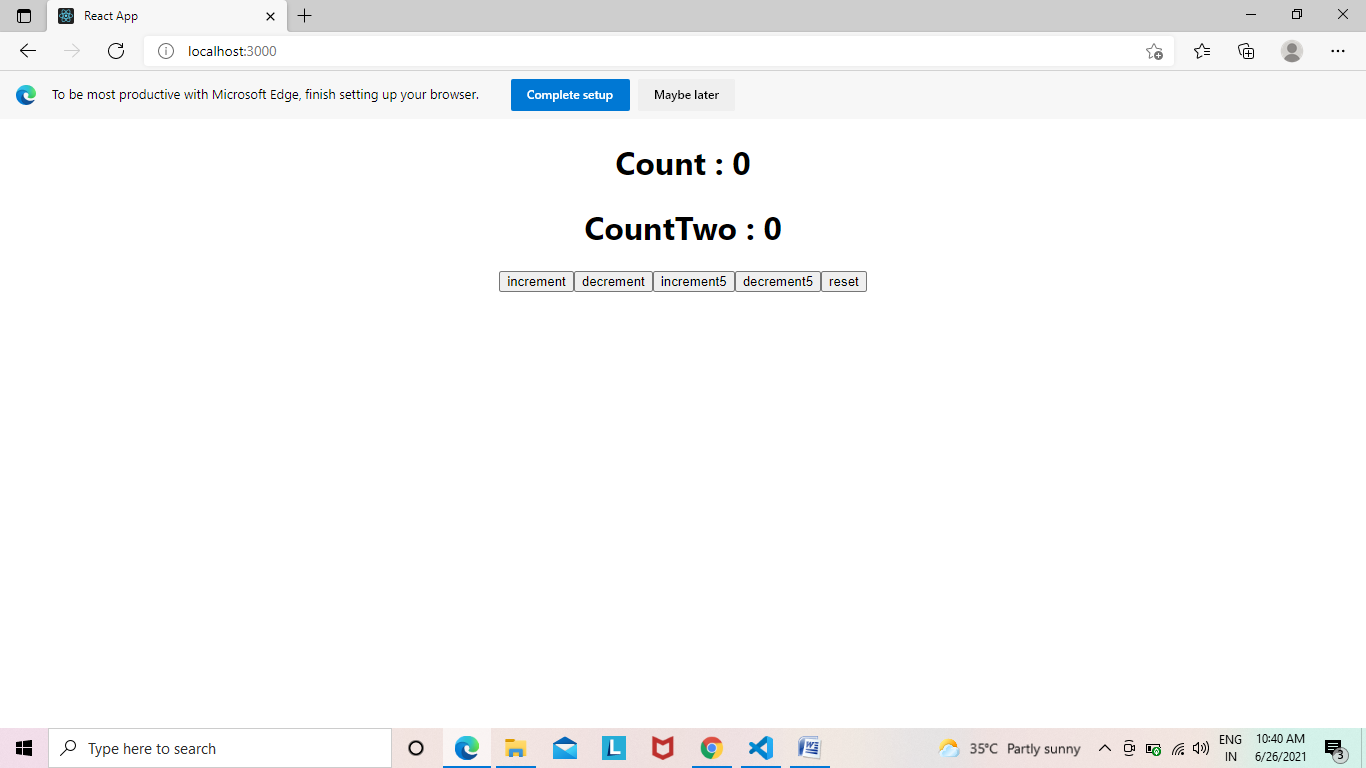
****

****

**After trying to perform countTwo related operations**

****

**Reset**

****

**App.js**

import './App.css';

import ComponentX from './Reducer/ComponentX';

function App() {

  return (

    <div className="App">

      <ComponentX/>

    </div>

  );

}

export default App;

**ComponentX.js**

import React from 'react'

import ComponentY from './ComponentY'

function ComponentX() {

    return (

        <div>

            <h1>This is component X</h1>

            <ComponentY/>

        </div>

    )

}

export default ComponentX

**ComonentY.js**

import React from 'react'

import ComponentZ from './ComponentZ'

function ComponentY() {

    return (

        <div>

            <h1>This is component Y</h1>

            <ComponentZ/>

        </div>

    )

}

export default ComponentY

**ComponentZ.js**

import React, { useContext } from "react";

function ComponentZ() {

  return (

    <div>

      <h1>This is component Z</h1>

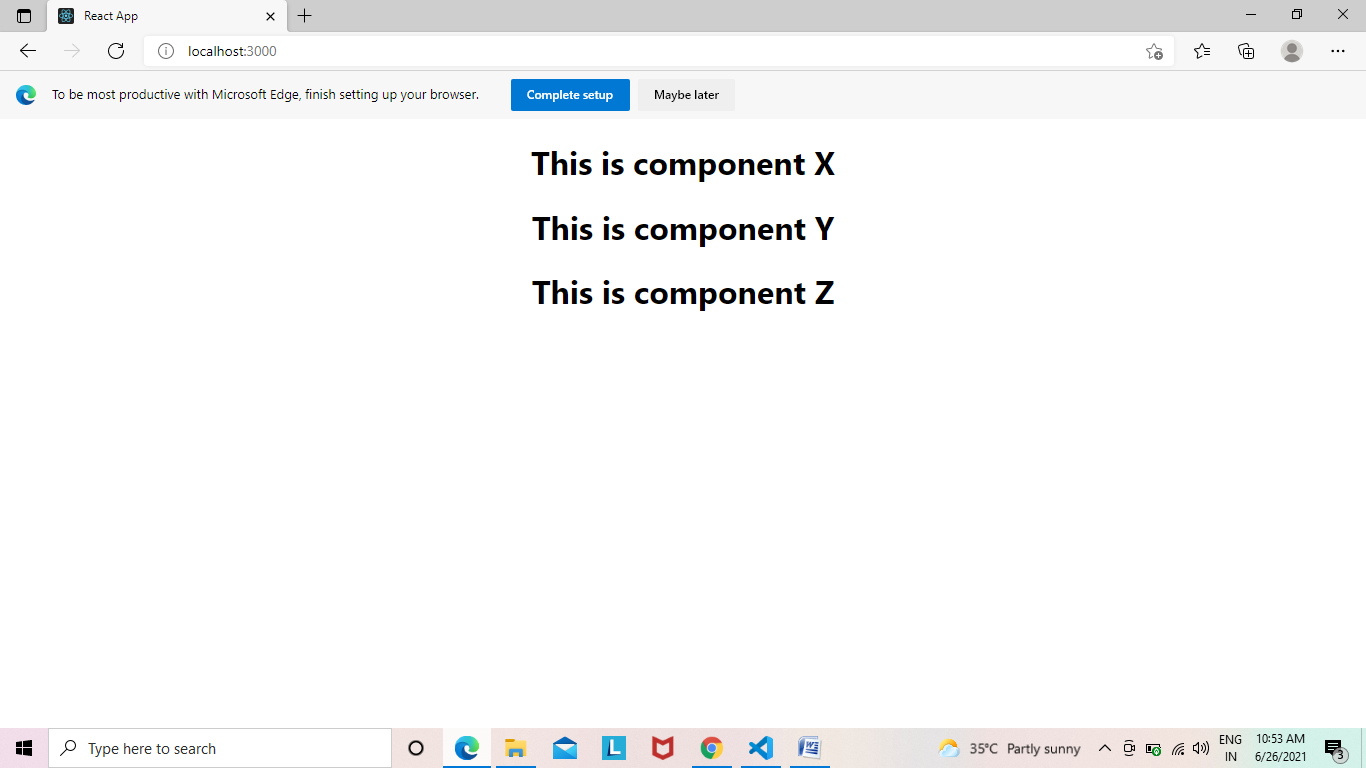
    </div>

  );

}

export default ComponentZ;

**Output:**

****

**App.js**

import './App.css';

import React,{ useReducer} from 'react'

import ComponentX from './Reducer/ComponentX';

import ComponentY from './Reducer/ComponentY';

import ComponentZ from './Reducer/ComponentZ';

export const CounterContext = React.createContext();

const initialState =0;

const reducer = (state, action) => {

  switch (action) {

    case "increment":

      return state + 1;

    case "decrement":

      return state - 1;

    case "reset":

      return initialState;

    default:

      return state;

  }

}

function App() {

  const [count,dispatch] = useReducer(reducer, initialState);

  return (

    <div className="App">

      <CounterContext.Provider value={{countState:count, countDispatch: dispatch}}>

         <ComponentX />

      </CounterContext.Provider>

    </div>

  );

}

export default App;

**ComponentX.js**

import React from 'react'

import ComponentY from './ComponentY'

function ComponentX() {

    return (

        <div>

            <h1>This is component X</h1>

            <ComponentY/>

        </div>

    )

}

export default ComponentX

**ComponentY.js**

import React from 'react'

import ComponentZ from './ComponentZ'

function ComponentY() {

    return (

        <div>

            <h1>This is component Y</h1>

            <ComponentZ/>

        </div>

    )

}

export default ComponentY

**ComponentZ.js**

import React, { useContext } from "react";

import { CounterContext } from "../App";

function ComponentZ() {

  const countContext = useContext(CounterContext);

  return (

    <div>

      <h1>This is component Z</h1>

      <h1>Count : {countContext.countState}</h1>

      <button onClick={()=>countContext.countDispatch('increment')}>increment</button>

      <button onClick={()=>countContext.countDispatch('decrement')}>decrement</button>

      <button onClick={()=>countContext.countDispatch('reset')}>reset</button>

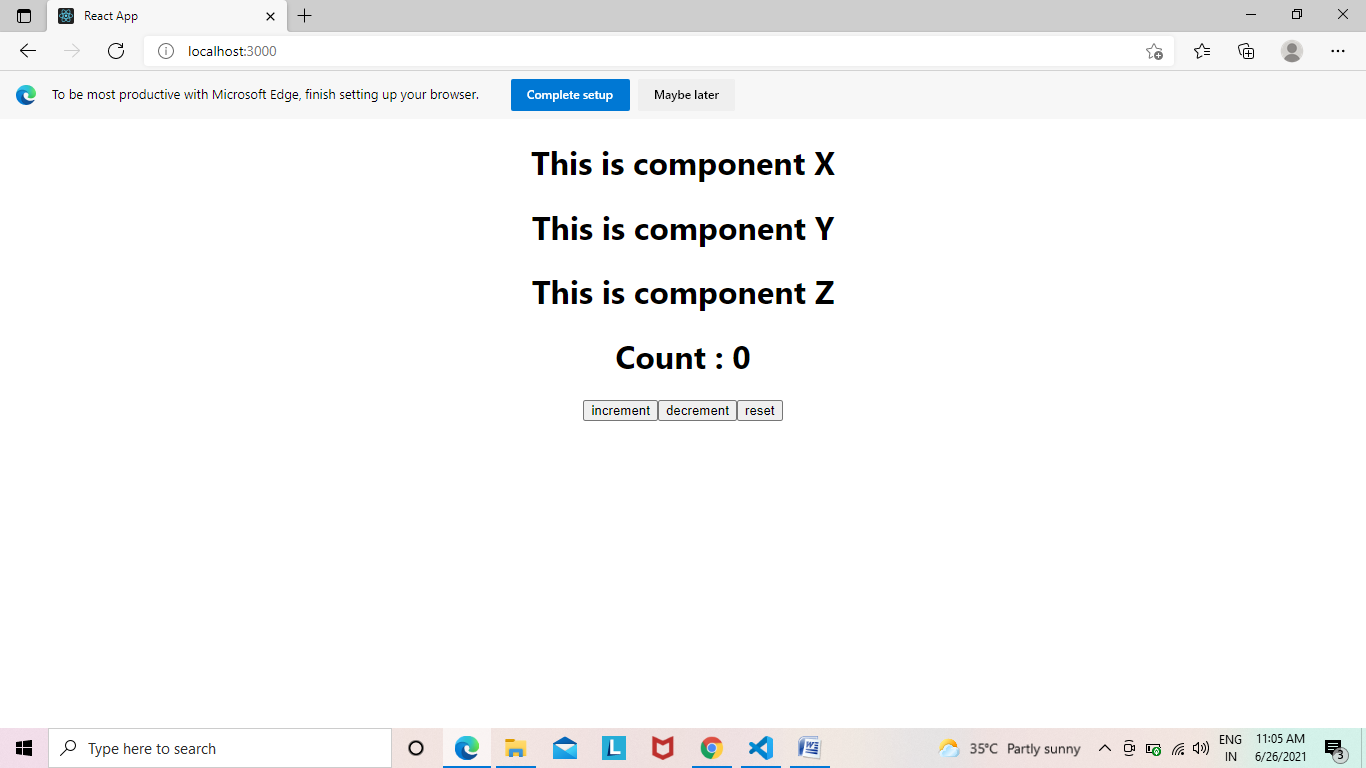
    </div>

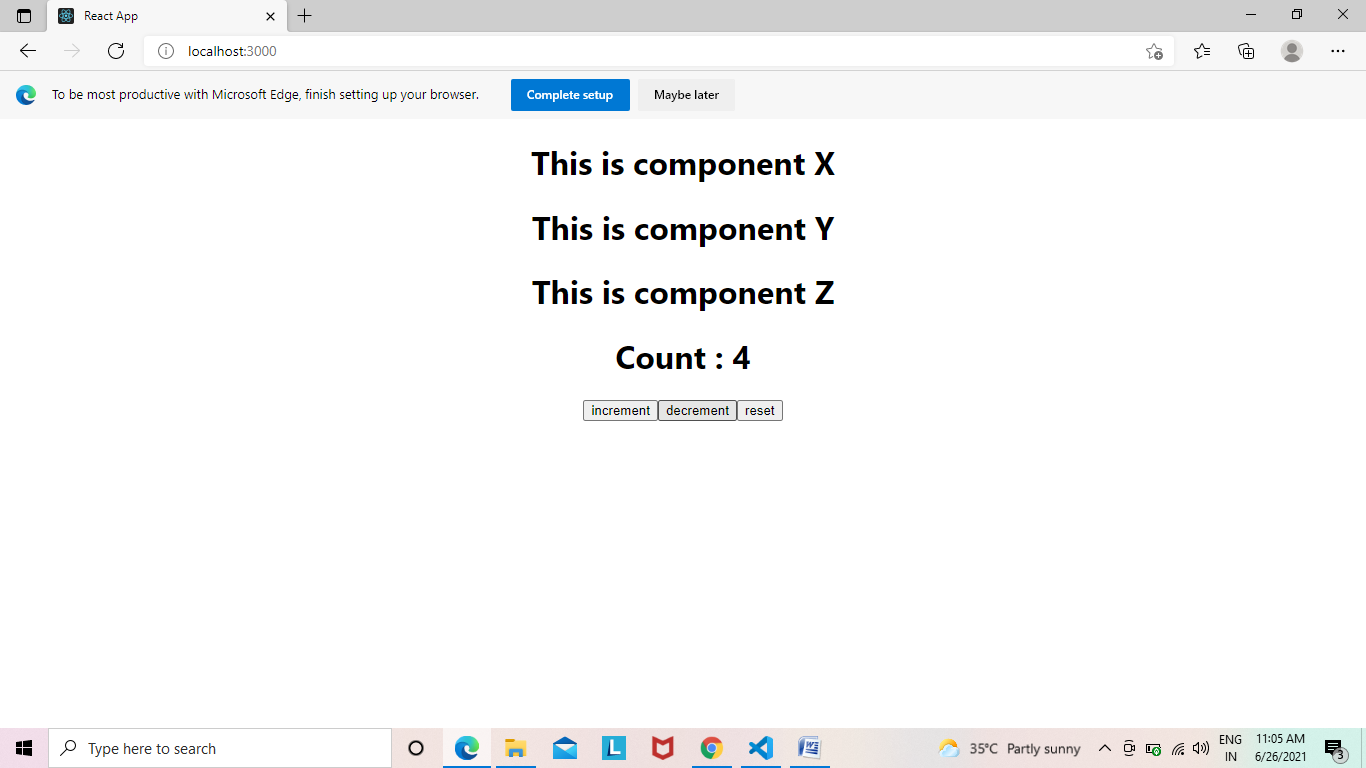
  );

}

export default ComponentZ;

**Output:**

****

****