**EXPERIMENT 9**

**AIM: Write a program which demonstrates Hooks(UseEffect), refs in React JS**

**CODE:**

**App.js:**

import React from 'react';

import CounterWithHooks from './CounterWithHooks';

function App() {

  return (

    <div className="App">

      <h1>React Hooks Example</h1>

      <CounterWithHooks />

    </div>

  );

}

export default App;

**CounterWithHooks.js:**

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

const CounterWithHooks = () => {

  // Declare state variable 'count'

  const [count, setCount] = useState(0);

  // useRef to store the reference of the input element

  const inputRef = useRef(null);

  // useEffect to trigger side effect when 'count' changes

  useEffect(() => {

    console.log(`Count has been updated to: ${count}`);

    // Example of DOM manipulation using refs - focus on input

    if (count === 5) {

      inputRef.current.focus();

      alert("The count is 5, focus is set on input!");

    }

    // Clean up function if required

    return () => {

      console.log("Cleaning up...");

    };

  }, [count]); // Dependency array - runs the effect when 'count' changes

  // Function to increment the count

  const increment = () => setCount(count + 1);

  return (

    <div style={{ textAlign: 'center' }}>

      <h2>Counter with useEffect and useRef</h2>

      <p>Current Count: {count}</p>

      <button onClick={increment}>Increment Count</button>

      <br />

      <br />

      <input ref={inputRef} placeholder="This input gets focus at count 5" />

    </div>

  );

};

export default CounterWithHooks;

**1. State management (useState)**:

* We're using the useState hook to create a state variable count and a function setCount to update it.

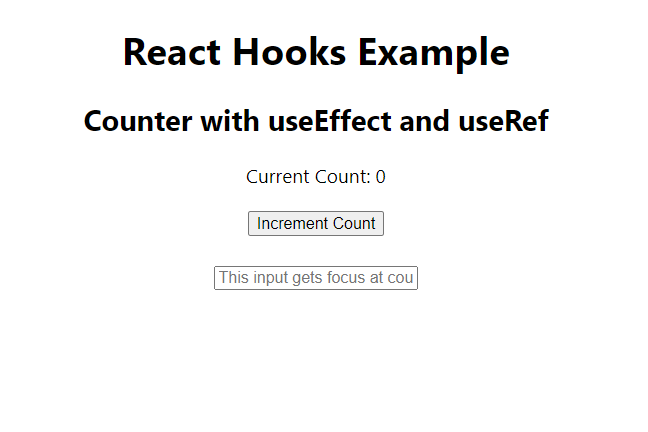
**2.** **useEffect**:

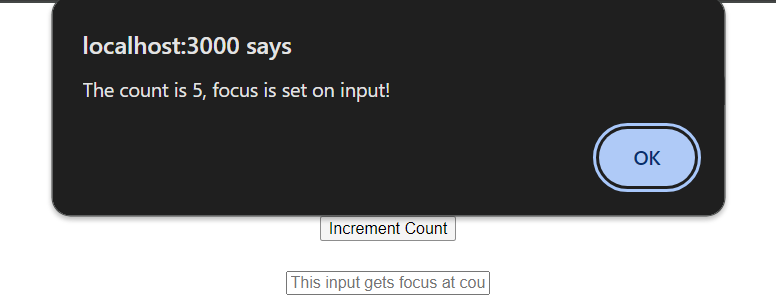
* This hook performs side effects (like logging and focusing an input field) whenever the count state changes.
* The dependency array [count] ensures the effect runs only when the count changes.

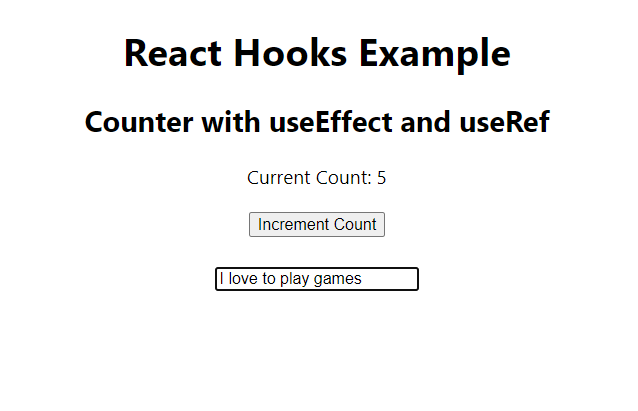
**3.** **useRef**:

* We use useRef to store a reference to the input element. We focus on the input when count reaches 5.

**OUTPUT:**

****

****

****

**CONCLUSION: Hence, we have successfully implemented a program to demonstrates Hooks(UseEffect), refs in React JS**