**Department of Information Technology**

**Academic Year: 2023-24 Name of Student:**

**Semester: V Student ID:**

**Class / Branch/ Div: TE- IT A/B Roll No.**

**Subject: IP Lab Date of Submission:**

**Name of Instructor: Prof. Yaminee Patil**

**Experiment No.:10**

**Aim:**  Write a program to demonstrate React Flux.

**Software Used:** Visual Studio Code

**Code:**

**App.js**

import React from "react";

import Counter from "./Counter";

function App() {

return (

<div>

<h1>Counter App</h1>

<Counter />

</div>

);

}

export default App;

**Counter.js**

import React, { Component } from "react";

import \* as CounterActions from "../actions/CounterActions";

import CounterStore from "../stores/CounterStore";

class Counter extends Component {

constructor() {

super();

this.state = {

count: CounterStore.getCount(),

};

}

componentDidMount() {

CounterStore.on("change", this.updateCount);

}

componentWillUnmount() {

CounterStore.removeListener("change", this.updateCount);

}

updateCount = () => {

this.setState({

count: CounterStore.getCount(),

});

};

handleIncrement = () => {

CounterActions.incrementCounter();

};

handleDecrement = () => {

CounterActions.decrementCounter();

};

render() {

return (

<div>

<p>Count: {this.state.count}</p>

<button onClick={this.handleIncrement}>Increment</button>

<button onClick={this.handleDecrement}>Decrement</button>

</div>

);

}

}

export default Counter;

**CounterActions.js**

import dispatcher from "../dispatcher/AppDispatcher";

export function incrementCounter() {

dispatcher.dispatch({

type: "INCREMENT",

});

}

export function decrementCounter() {

dispatcher.dispatch({

type: "DECREMENT",

});

}

**AppDispatcher.js**

import { Dispatcher } from "flux";

const dispatcher = new Dispatcher();

export default dispatcher;

**CounterStore.js**

import { EventEmitter } from "events";

import dispatcher from "../dispatcher/AppDispatcher";

class CounterStore extends EventEmitter {

constructor() {

super();

this.count = 0;

}

getCount() {

return this.count;

}

increment() {

this.count++;

this.emit("change");

}

decrement() {

this.count--;

this.emit("change");

}

handleActions(action) {

switch (action.type) {

case "INCREMENT":

this.increment();

break;

case "DECREMENT":

this.decrement();

break;

default:

// do nothing

}

}

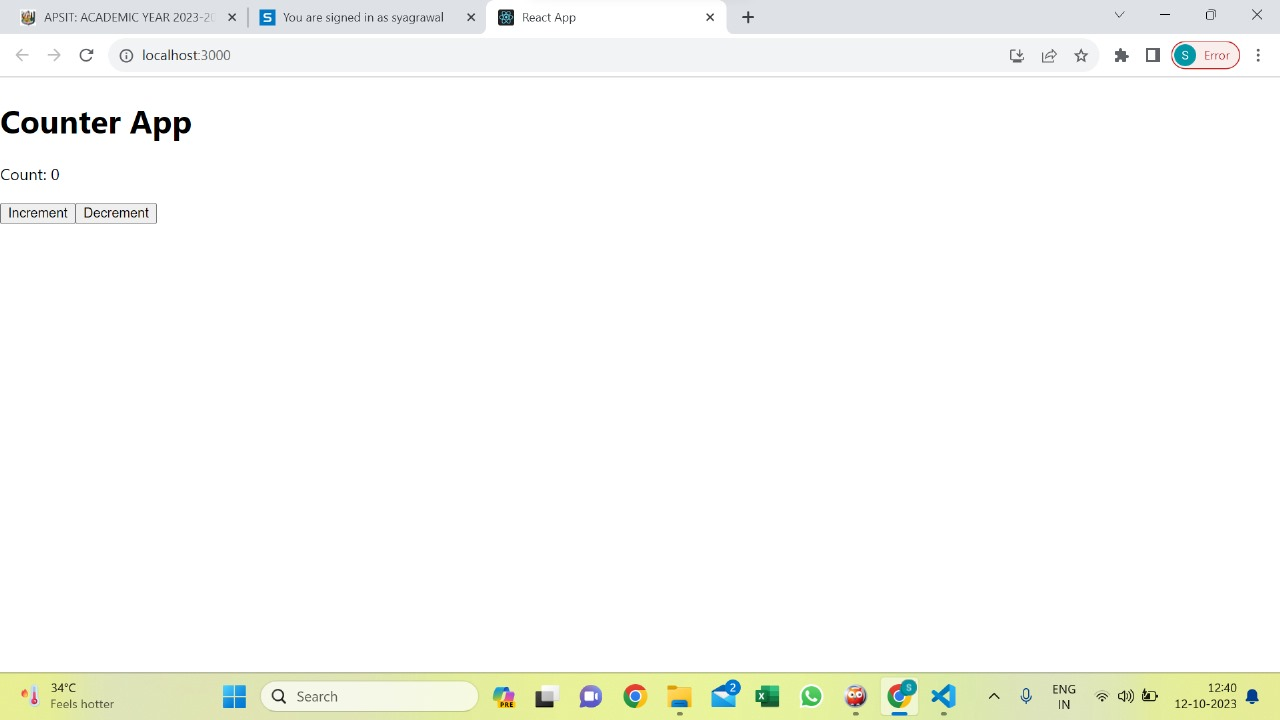
}

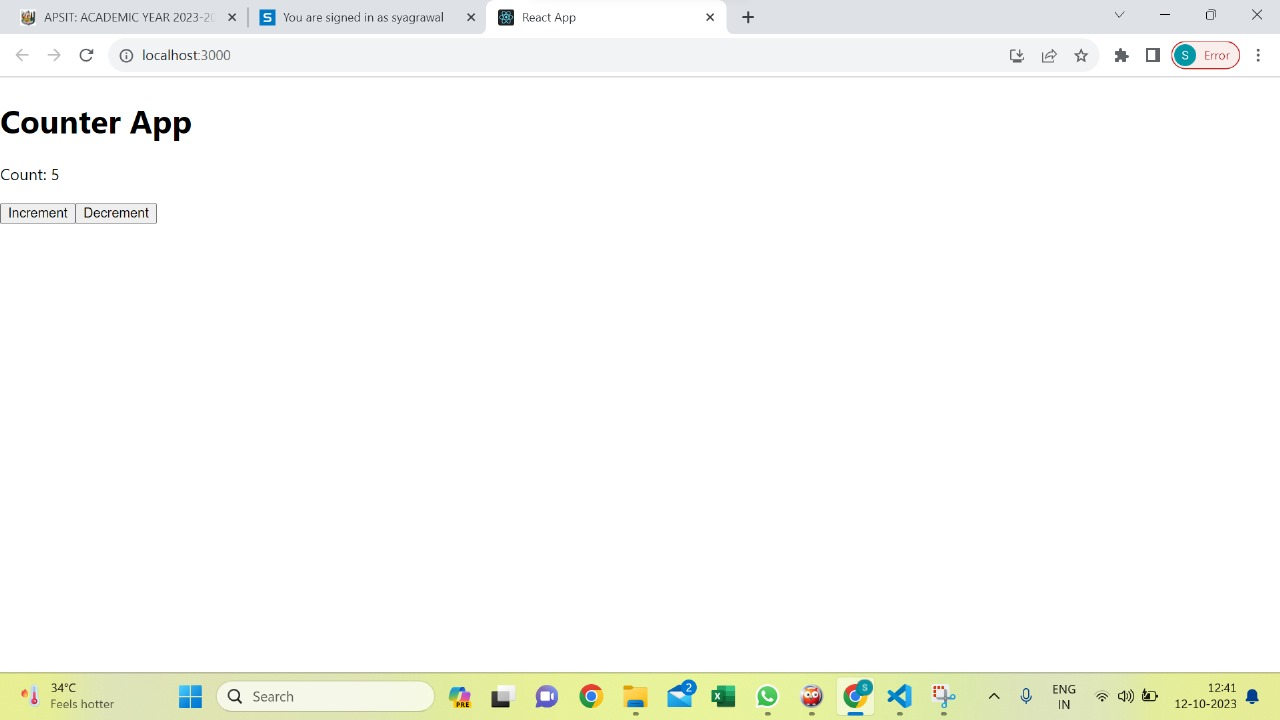
const counterStore = new CounterStore();

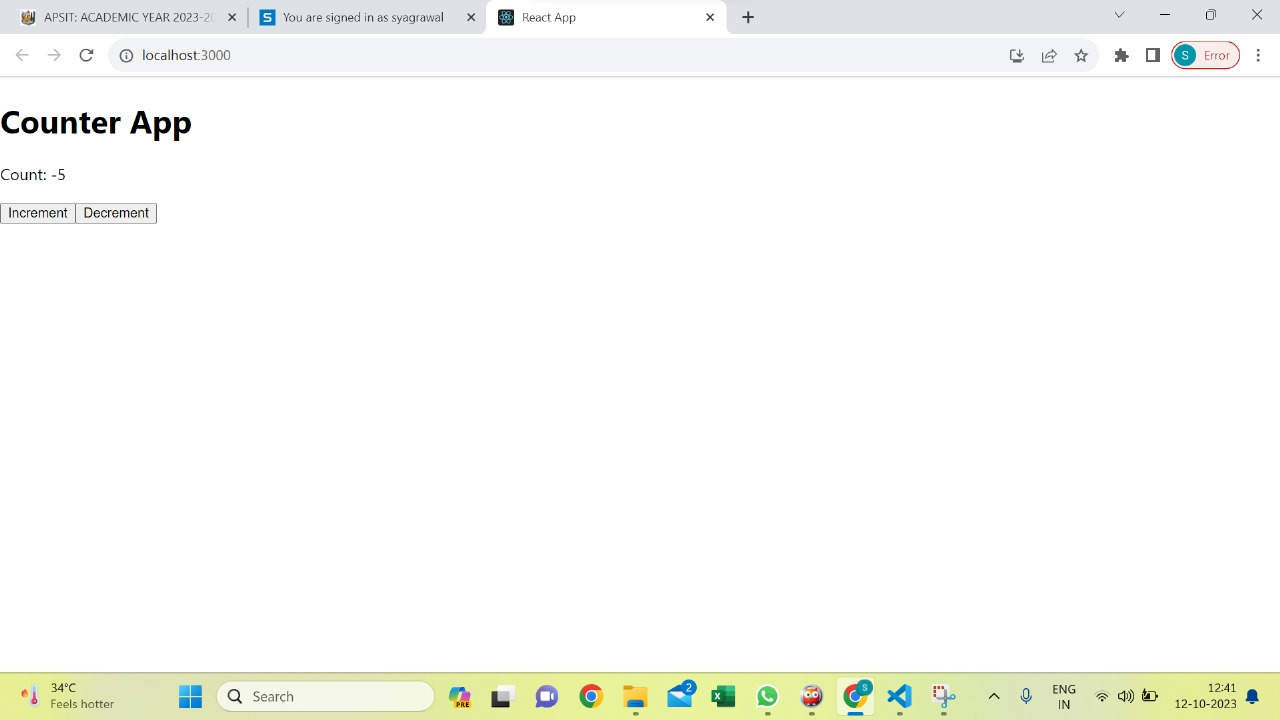
dispatcher.register(counterStore.handleActions.bind(counterStore));

export default counterStore;

**Output:**







**Conclusion:** In this experiment we have implemented program to demonstrate react flux which is a powerful architecture for managing state in ReactJS applications. The program helps us understand unidirectional data flow, centralized control, and immutable data, using Flux and helps keep application's state management organized and maintainable.

**Department of Information Technology**

**Academic Year: 2023-24 Name of Student:**

**Semester: V Student ID:**

**Class / Branch/ Div: TE- IT A/B Roll No.**

**Subject: IP Lab Date of Submission:**

**Name of Instructor: Prof. Yaminee Patil**

**Experiment No.:10**

**Aim:**  Write a program to demonstrate React Flux.

**Software Used:** Visual Studio Code