**Practical No 3**

**AIM: Using States in Functional components**

**Q. 1 Create Counter using useState hook**

🡺 Count1.js

import React, { useState } from 'react';

const Count1 = () => {

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

return (

<div>

<p>You clicked {count} times</p>

<button onClick={() => setCount(count + 1)}>

Click me

</button>

</div>

);

}

export default Count1

🡺App.js

import Count1 from "./components/counter1";

function App() {

return (

<div className="App">

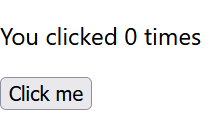
<Count1/>

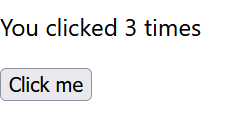
</div>

);

}

export default App;

O/p:



**Q2 Increment the age using useReducer hook**

🡺Functional\_age.js

import { useReducer } from 'react';

function reducer(state, action) {

if (action.type === 'incremented\_age') {

return {

age: state.age + 1

};

}

throw Error('Unknown action.');

}

export default function Counter1() {

const [state, dispatch] = useReducer(reducer, { age: 42 });

return (

<>

<button onClick={() => {

dispatch({ type: 'incremented\_age' })

}}>

Increment age

</button>

<p>Hello! You are {state.age}.</p>

</>

);

}

🡺App.js

import Counter1 from "./components/functional\_age";

function App() {

return (

<div className="App">

<Counter1/>

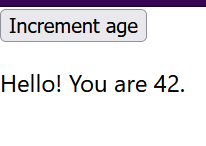
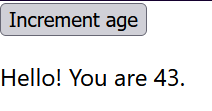
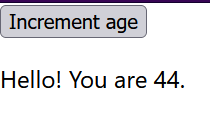
</div>

);

}

export default App;

O/p:



**Q. 3 Perform incrementation & decrementation using useReducer hook**

**🡺** incre\_decr.js

import { useReducer } from 'react';

const initialState = {count: 0};

function reducer(state, action) {

switch (action.type) {

case 'increment':

return {count: state.count + 1};

case 'decrement':

return {count: state.count - 1};

default:

throw new Error();

}

}

function Incredecre() {

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

return (

<div>

Count: {state.count} <br></br>

<button onClick={() => dispatch({type: 'increment'})}>Increment by 1</button><br></br>

<button onClick={() => dispatch({type: 'decrement'})}>Decrement by 1</button>

</div>

);

}

export default Incredecre;

🡺App.js

import Incredecre from "./components/incr\_decr";

function App() {

return (

<div className="App">

< Incredecre />

</div>

);

}

export default App;

**Q. 4 Perform increment decrement and reset operation using useReducer hook**

**🡺**Incrdecr.js

import { useReducer } from 'react';

const initialState = 0;

const reducer = (state, action) => {

switch (action) {

case "add":

return state + 1;

case "subtract":

return state - 1;

case "reset":

return 0;

default:

throw new Error("Error");

}

};

const Arithmetic = () => {

// Initialising useReducer hook

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

return (

<div>

<h2>{count}</h2>

<button onClick={() => dispatch("add")}>

Increment

</button>

<button onClick={() => dispatch("subtract")}>

decrement

</button>

<button onClick={() => dispatch("reset")}>

reset

</button>

</div>

);

};

export default Arithmetic;

🡺App.js

import Arithmetic from "./components/incredecr";

function App() {

return (

<div className="App">

< Arithmetic/>

</div>

);

}

export default App;

