**Exp-8 Using useState in ReactJS**

**Aim:** To make a counter app using ReactJS.

**Theory:**

This React application demonstrates the use of the **useState** hook for managing state in functional components. It implements a simple counter that allows users to increment or decrement the count by a specified step value. The step value can be adjusted using an input field. The application consists of three main functions:

1. **Increment (inc)** – Increases the counter by the step value.
2. **Decrement (dec)** – Decreases the counter by the step value.
3. **Reset (reset)** – Resets the counter to its initial value.

The component maintains two state variables: counter for the count and step for the step size. The setStep function ensures that an empty input defaults to 1 to prevent errors. Inline styles are used for basic styling of the UI.

**Code:**import React, { useState } from "react";

export default function App() {

  const initialCount = 10;

  const [counter, setCounter] = useState(initialCount);

  const [step, setStep] = useState(5);

  const inc = () => {

    setCounter(counter + step);

  };

  const dec = () => {

    setCounter(counter - step);

  };

  const reset = () => {

    setCounter(initialCount);

  };

  return (

    <div

      className="main"

      style={{ display: "flex", justifyContent: "center", margin: "60px" }}

    >

      <div

        className="counter1"

        style={{

          display: "flex",

          flexDirection: "column",

          alignItems: "center",

          border: "solid 2px black",

          width: "300px",

          paddingBottom: "25px",

          borderRadius: "10px",

          backgroundColor: "#f0f0f0",

        }}

      >

        <span style={{ display: "flex", alignItems: "center", gap: "20px" }}>

          <p>Count : </p>

          <p style={{ fontSize: "28px" }}>{counter}</p>

        </span>

        <p>

          Step :{" "}

          <input

            type="number"

            value={step}

            onChange={(e) => setStep(e.target.value || 1)}

            style={{ width: "40px", border: "1px solid", borderRadius: "4px" }}

          />

        </p>

        <div className="buttons" style={{ display: "flex", gap: "10px" }}>

          <button

            onClick={inc}

            style={{ backgroundColor: "#0bd70b", borderRadius: "5px" }}

          >

            Increment

          </button>

          <button

            onClick={dec}

            style={{ backgroundColor: "#ff2a2a", borderRadius: "5px" }}

          >

            Decrement

          </button>

          <button

            onClick={reset}

            style={{ backgroundColor: "#b2b2b2", borderRadius: "5px" }}

          >

            Reset

          </button>

        </div>

      </div>

    </div>

  );

}

**Output:**

