**Evnet Handler**

**onclick**

import React, { useState } from 'react'

const Conclik = () => {

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

    const handleClick = () => {

        setCount(count + 1);

    };

    return (

        <div>

            <button onClick={handleClick}>Click Me!</button>

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

        </div>

    )

}

export default Conclik

**onclick event**

import React from 'react'

const Conclicktwo = () => {

    const handleButtonClick = (event) => {

        event.stopPropagation(); // Stops the event

        console.log("Button Clicked");

    };

    return (

        <div style={{

            backgroundColor: "aqua",

            width: "300px"

        }} onClick={() => console.log("Parent Div Clicked")}>

            <button onClick={handleButtonClick}>

                Click Me

            </button>

        </div>

    )

}

export default Conclicktwo

**onMouseEnter onMouseLeave**

import React, { useState } from 'react'

const Conmouseriverleave = () => {

    const [isHovered, setIsHovered] = useState(false);

    const handleMouseEnter = () => {

        setIsHovered(true);

    };

    const handleMouseLeave = () => {

        setIsHovered(false);

    };

    console.log(isHovered);

    return (

        <div className='App'>

            <div>

                <h1>onMouseEnter and onMouseLeave</h1>

                <br /><br />

            </div>

            <div

                style={{

                    display: "flex",

                    justifyContent: "center"

                }}>

                <div

                    onMouseEnter={handleMouseEnter}

                    onMouseLeave={handleMouseLeave}

                    style={{

                        width: '200px',

                        height: '200px',

                        backgroundColor: isHovered ? 'lightgreen' : 'lightblue',

                        textAlign: 'center',

                        lineHeight: '200px',

                    }}

                >

                    {isHovered ? 'Mouse is inside!' : 'Hover over me!'}

                </div>

            </div>

        </div>

    )

}

export default Conmouseriverleave

**onSubmit simple State**

import React, { useState } from "react";

const Customerform = () => {

    const [firstName, setfirstName] = useState('');

    const [lastName, setLastName] = useState('');

    const [email, setEmail] = useState('');

    const handleSubmit = (event) => {

        console.log(event);

        event.preventDefault();

        console.log('first name:', firstName);

        console.log('last name:', lastName);

        console.log('email:', email);

    };

    return (

        <form onSubmit={handleSubmit}>

            <input type="text" name='firstName' onChange={(e) => setfirstName(e.target.value)} value={firstName} />

            <input type="text" name='lastName' onChange={(e) => setLastName(e.target.value)} value={lastName} />

            <input type="email" name='email' onChange={(e) => setEmail(e.target.value)} value={email} />

            <button type="submit">Submit</button>

        </form>

    )

}

export default Customerform

**onSubmit destructuring**

import React, { useState } from "react"

const Customerform = () => {

    const [formData, setformData] = useState({

        firstName: '',

        lastName: '',

        email: ''

    })

    const handleChange = (event) => {

        const { name, value } = event.target

        setformData({

            ...formData,

            [name]: value

        })

    }

    const handleSubmit = (event) => {

        console.log(event);

        event.preventDefault();

        console.log('Form submitted with data:', formData);

    };

    return (

        <form onSubmit={handleSubmit}>

            <input type="text" name='firstName' onChange={handleChange} value={formData.firstName} />

            <input type="text" name='lastName' onChange={handleChange} value={formData.lastName} />

            <input type="email" name='email' onChange={handleChange} value={formData.email} />

            <button type="submit">Submit</button>

        </form>

    )

}

export default Customerform