**Write a program to Illustrate Controlled and Uncontrolled Components**

**Controlled Components:**

Controlled components have their state managed by React. The component's state is updated through React's state management.

**Code:**

//ControlledComponent

import React  from 'react';

import { useState } from 'react';

const ControlledComponent=()=>{

    const [inputText,setInputText]=useState("");

    const handleSubmit=(e)=>{

        alert(inputText);

    }

    const handleChange=(e)=>

    {

        setInputText(e.target.value);

    }

    return (

        <><form>

            <p>Enter the name<input type="text" value={inputText} onChange={handleChange}/></p>

            <button type="button" onClick={handleSubmit}>Submit</button>

        </form></>

    );

}

export default ControlledComponent;

**Work:**

* This program creates a controlled component in React that manages user input through the useState hook.
* It renders a form with an input field and a button, where the input field's value is bound to the component's state.
* When the user types in the input field, the state is updated, and when the button is clicked, the current input value is alerted.
* This demonstrates a basic controlled component that synchronizes user input with React's state management.

**Uncontrolled Components:**

Uncontrolled components manage their own state internally. React doesn't control the input data, and the component's state is updated directly.

**Code:**

//UncontrolledComponent

import React ,{useRef} from 'react';

const Uncontrolled=()=>{

    const inputRef=useRef(null);

    const handleSubmit=(e)=>{

      //  e.preventDefault();

        alert(inputRef.current.value);

    }

    return(

        <><form>

            <p>Enter the name :<input type='text' ref={inputRef}/></p>

            <button  onClick={handleSubmit}>Submit</button>

        </form></>

    );

}

export default Uncontrolled;

**Work:**

* This program creates an uncontrolled component in React using the useRef hook to manage user input.
* It renders a form with an input field and a button, where the input field is assigned a reference using inputRef.
* When the button is clicked, the handleSubmit function alerts the current value of the input field using inputRef.current.value.
* This demonstrates an uncontrolled component that accesses the input field's value directly using a reference, without relying on React's state management.