REF

Refs provide a way to directly interact with the DOMof components when necessary. However, they should be used sparingly as they bypass the usual React data flow and can make code harder to understand and maintain. In most cases, you should prefer using React's state and props system for managing component behavior.

e.g. – 1

```
import React,{createRef} from 'react';
import User from "./User"
class App extends React.PureComponent {
constructor()
  {
super()
this.inputref=createRef()
  }
componentDidMount()
{
  console.log(this.inputref.current.value=1000)
}

render() {
    return (
<div>
<h1> ref in react</h1>
<input type="text" ref ={this.inputref} />
</div>
    );
  }
}
export default App;
```

e.g.-2

```
import React, { createRef } from "react";
import User from "./User";
class App extends React.PureComponent {
constructor() {
super();
this.inputref = createRef();
 }
componentDidMount() {
  //  console.log((this.inputref.current.value = 1000));
 }
getval() {
  //  console.log((this.inputref));
  //console.log((this.inputref.current.value = 1000));
this.inputref.current.style.color ="red"
this.inputref.current.style.backgroundColor ="black"

 }
```

```
render() {
   return (
<div>
<h1> ref in react</h1>
<input type="text" ref={this.inputref} />
<button onClick={() =>this.getval()}> ref test</button>
</div>
   );
  }
}
export default App;
```

USEREF

In React, the useRef hook is used to create a mutable reference that persists across renders of a functional component. It allows you to reference and store a value that persists between renders without triggering a re-render when the value changes.

The useRef hook returns a mutable ref object with a .current property that can be used to store and access the current value. The .current property can be updated directly without causing a re-render.

e.g.

```
import React, { useRef } from "react";
import User from "./User";
function  App(){
  let inputref = useRef(null)
  function test_ref(){
inputref.current.value = 100
inputref.current.focus()
inputref.current.style.color="red"

  }
  return (
<div>
<h1> ref in react</h1>
<input type="text" ref={inputref} />
<button onClick={test_ref}> ref test</button>
<button onClick={()=>{inputref.current.style.display="none"}}> hide</button>
</div>
   );
}

export default App;
```