



## Day 58

# “Web Development + Security”

## Conditional Rendering & Rendering Lists in React:

### What is Conditional Rendering?

Conditional Rendering means showing (or hiding) components or elements based on a condition — just like using `if`, `else`, or the ternary operator in plain JavaScript. In React, it helps display different UI parts dynamically, depending on state, props, or data.

### What is Rendering Lists?

Rendering Lists means displaying multiple similar elements — like a list of users, products, or tasks — by looping over an array.

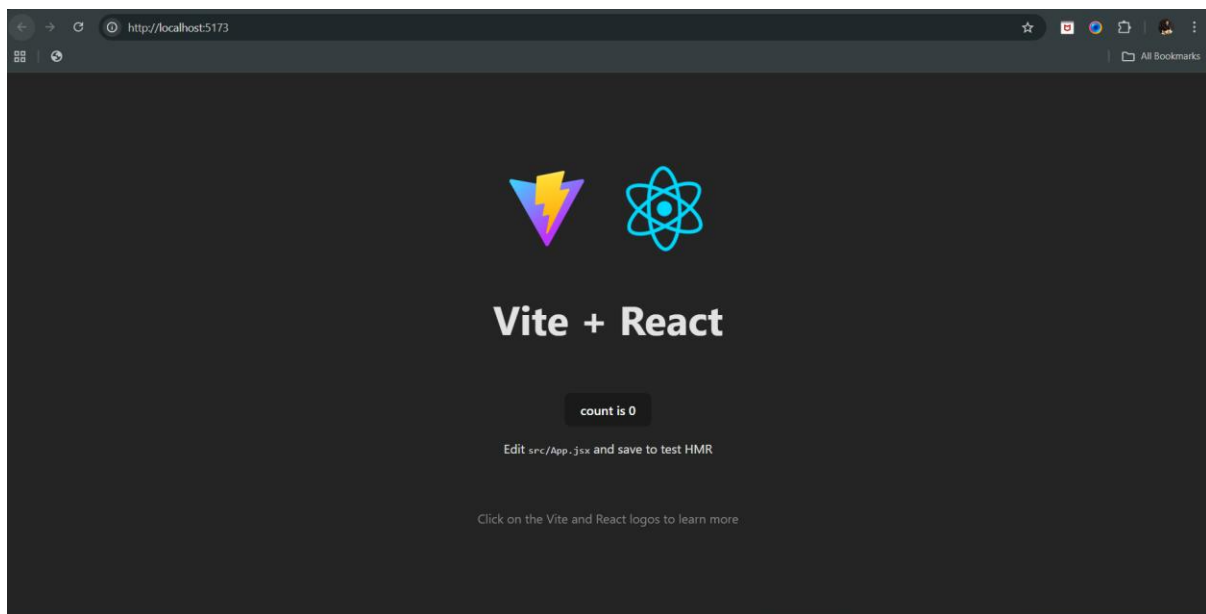
React does this using the `.map()` function.

### In Short

- Conditional rendering = *show/hide content dynamically*
- List rendering = *display many items efficiently*
- Together, they make React apps interactive and data-driven

To understand, we will run the `npm run dev`, as we always do from last few days:

Output:

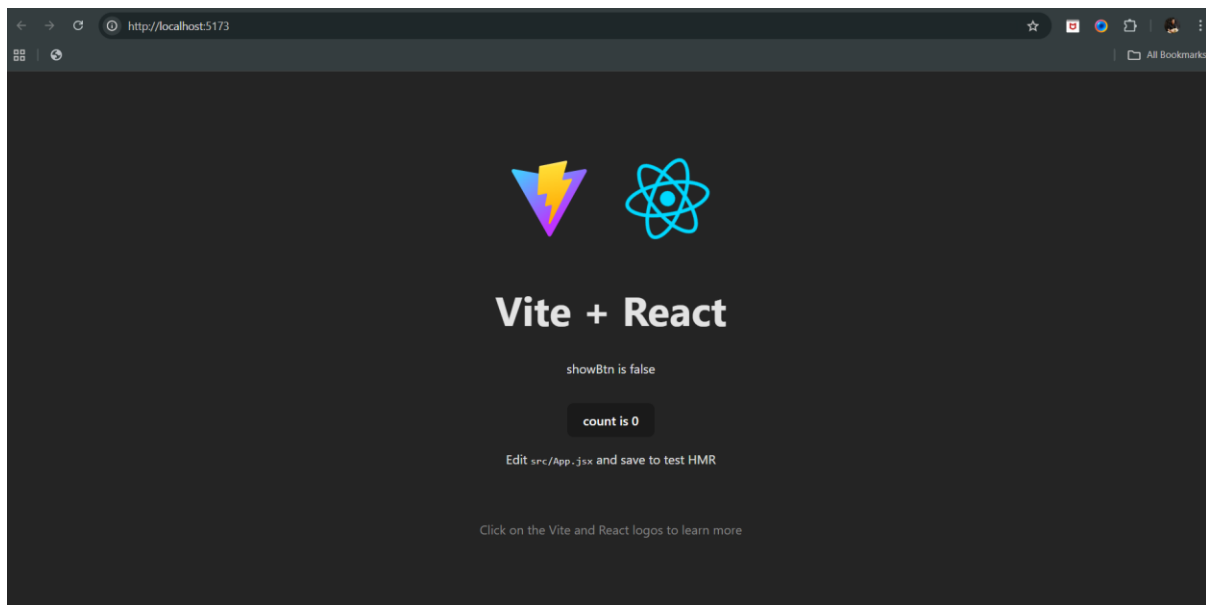


Now, suppose we want that a button above the count will show only if the showBtn which is a variable in the snippet be true, since by default its value is false. For this we will edit the App.jsx as shown below:

App.jsx: line number 8 and 21 added

```
App.jsx
src > App.jsx > App
1  import { useState } from 'react'
2  import reactLogo from './assets/react.svg'
3  import viteLogo from '/vite.svg'
4  import './App.css'
5
6  function App() {
7    const [count, setCount] = useState(0)
8    const [showBtn, setShowBtn] = useState(false) //default-false by us
9
10   return (
11     <>
12       <div>
13         <a href="https://vite.dev" target="_blank">
14           <img src={viteLogo} className="logo" alt="Vite logo" />
15         </a>
16         <a href="https://react.dev" target="_blank">
17           <img src={reactLogo} className="logo react" alt="React logo" />
18         </a>
19       </div>
20       <h1>Vite + React</h1>
21       {showBtn?<button>Displayed when showBtn is True</button>:"showBtn is false"}
22       <div className="card">
23         <button onClick={() => setCount((count) => count + 1)}>
```

Output: since showBtn in the code is written as false.

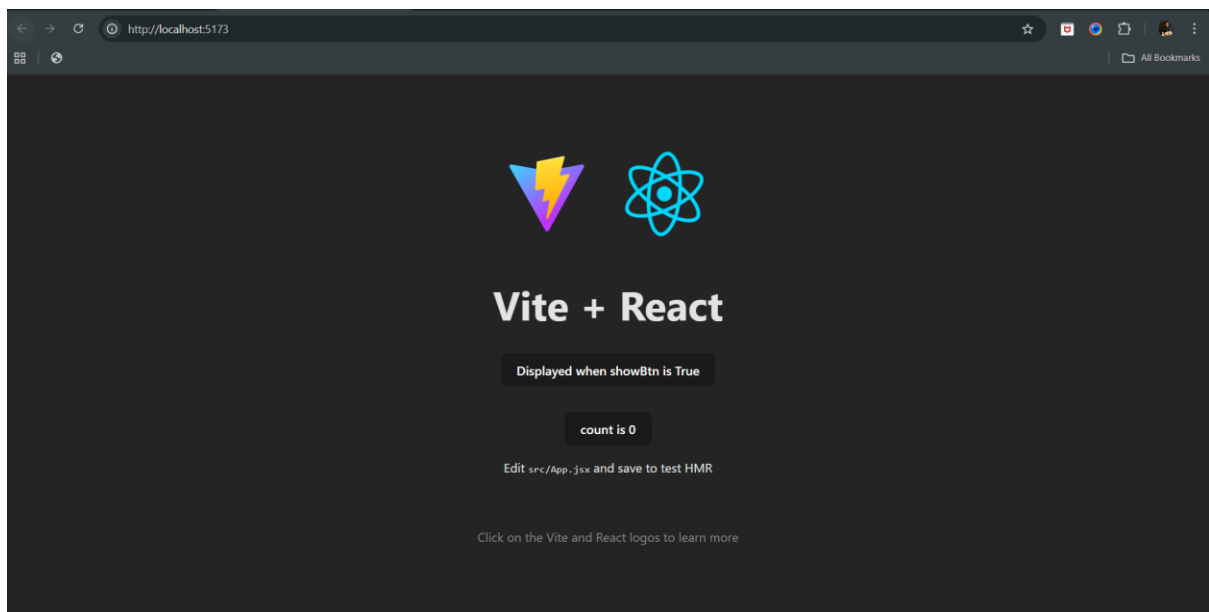


Now, in case we hard code the line number 8 to be true, then.

App.jsx: see line 8 changed from false to true.

```
App.jsx U X
src > App.jsx > App
1  import { useState } from 'react'
2  import reactLogo from './assets/react.svg'
3  import viteLogo from '/vite.svg'
4  import './App.css'
5
6  function App() {
7    const [count, setCount] = useState(0)
8    const [showBtn, setShowBtn] = useState(true)
9
10   return (
11     <>
12     <div>
13       <a href="https://vite.dev" target="_blank">
14         <img src={viteLogo} className="logo" alt="vite logo" />
15       </a>
16       <a href="https://react.dev" target="_blank">
17         <img src={reactLogo} className="logo react" alt="React logo" />
18       </a>
19     </div>
20     <h1>Vite + React</h1>
21     {showBtn?<button>Displayed when showBtn is True</button>:"showBtn is false"}
22     <div className="card">
```

Output: as expected button is visible.

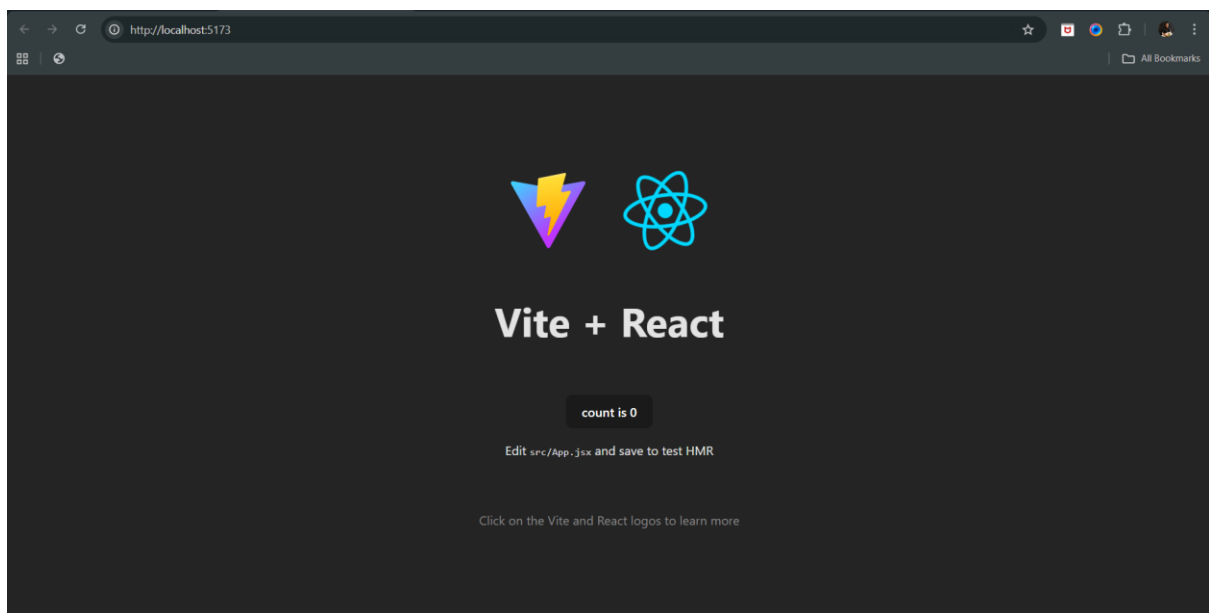


Is there any other standard way to do so? Yes, we can use `&&` in the `{}` to do so, as shown below:

App.jsx: notice line number 8 and 21.

```
App.jsx U x
src > App.jsx > App
1  import { useState } from 'react'
2  import reactLogo from './assets/react.svg'
3  import viteLogo from '/vite.svg'
4  import './App.css'
5
6  function App() {
7    const [count, setCount] = useState(0)
8    const [showBtn, setShowBtn] = useState(false)
9
10   return (
11     <>
12       <div>
13         <a href="https://vite.dev" target="_blank">
14           <img src={viteLogo} className="logo" alt="Vite logo" />
15         </a>
16         <a href="https://react.dev" target="_blank">
17           <img src={reactLogo} className="logo react" alt="React logo" />
18         </a>
19       </div>
20       <h1>Vite + React</h1>
21       {showBtn && <button>Displayed when showBtn is True</button>}
22     <div className="card">
```

Output: as it is false, button will not show.

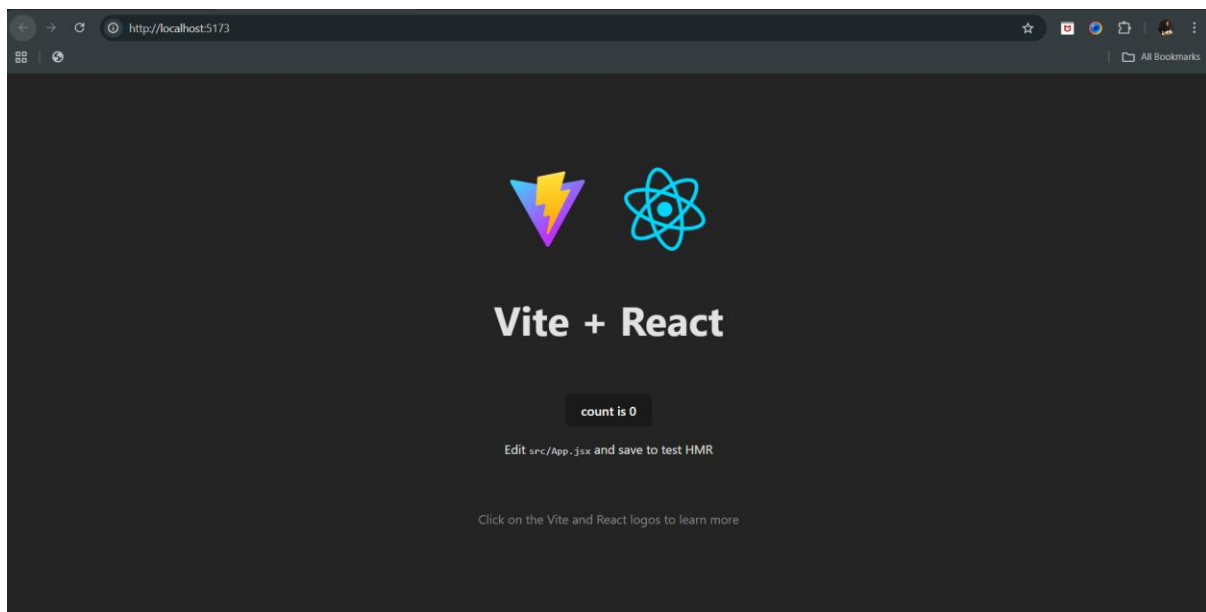


Now, can we make the changes such that when the count button will be clicked the button will get toggled? Yes, we can do so. See below:

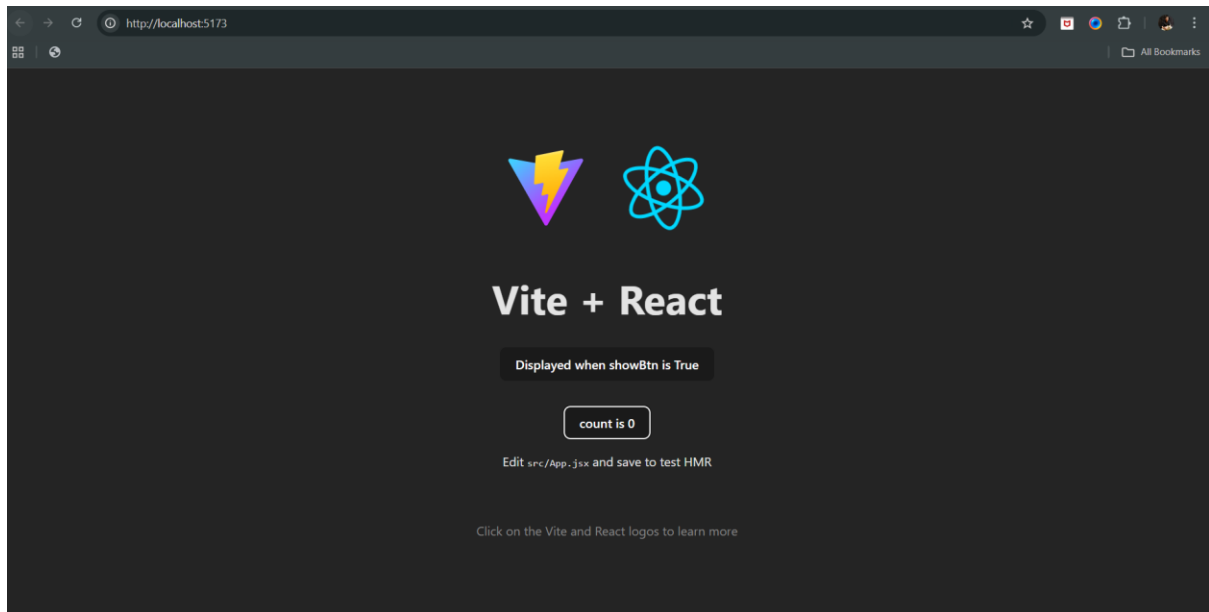
App.jsx: notice line 23.

```
App.jsx U X
src > App.jsx > App
6 function App() {
7   const [count, setCount] = useState(0)
8   const [showBtn, setShowBtn] = useState(false)
9
10  return (
11    <>
12      <div>
13        <a href="https://vite.dev" target="_blank">
14          <img src={viteLogo} className="logo" alt="Vite logo" />
15        </a>
16        <a href="https://react.dev" target="_blank">
17          <img src={reactLogo} className="logo react" alt="React logo" />
18        </a>
19      </div>
20      <h1>Vite + React</h1>
21      {showBtn && <button>Displayed when showBtn is True</button>}
22      <div className="card">
23        <button onClick={() => setShowBtn(() => setShowBtn(!showBtn))}>
24          count is {count}
```

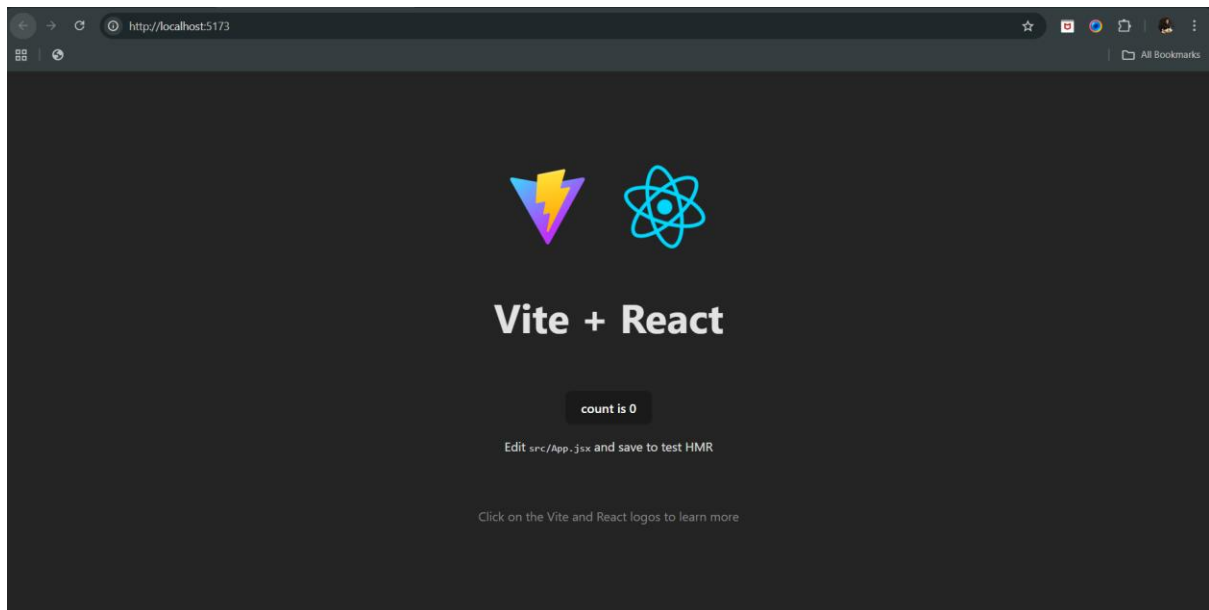
Output: since by default it is false, button will not be visible when loaded first



Output: when count button was clicked



Output: when button was clicked again



--The End--