

## Day 59

# “Web Development + Security”

## Handling Events in React:

### What Are Events?

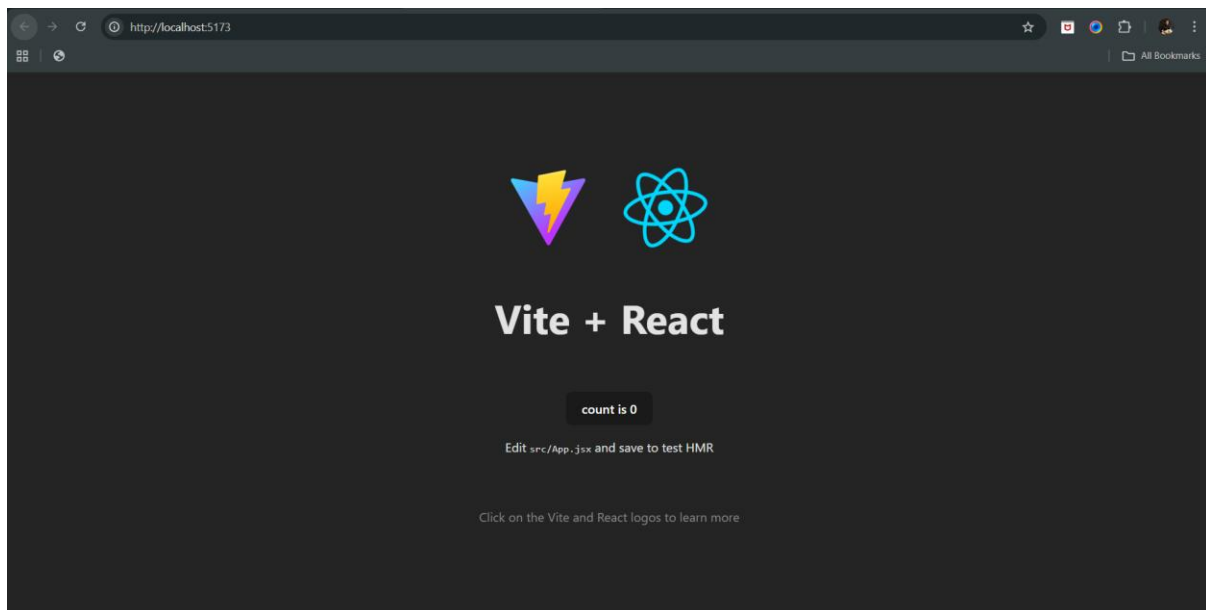
In web development, events are actions or occurrences that happen in the browser —like a click, key press, mouse move, form submit, etc.

In React, event handling is similar to vanilla JS, but with some differences:

JavaScript	React
Uses lowercase event names (onclick)	Uses camelCase (onClick)
Uses strings in HTML (onclick="fn()")	Uses functions (onClick={fn})
this refers to element	In React, handled inside components

To understand it, we will again create the folder using `npm create vite@latest`, and then by running the `npm run dev`, we will get the below output:

Output:

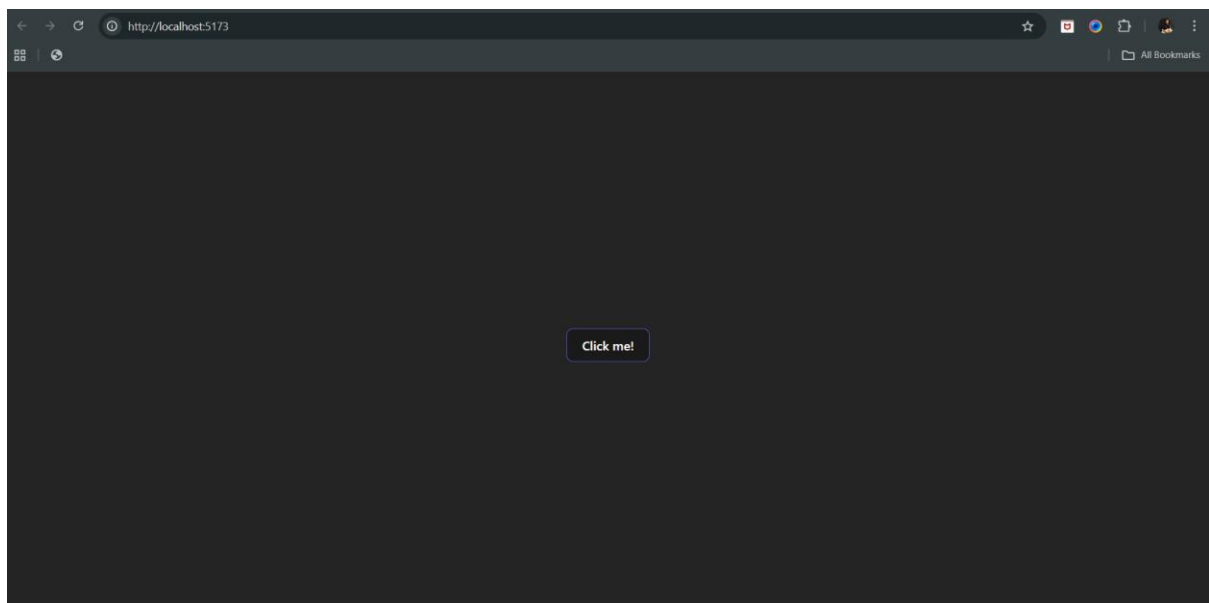


Now, we can create a simple event, where the alert box will come when someone will click the button, for which we will edit the `App.jsx` and remove the previous functions and code, and write the following code:

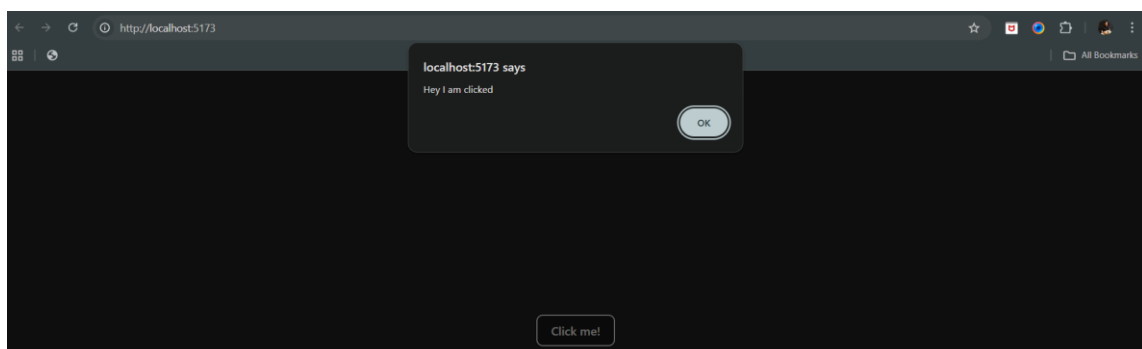
App.jsx:

```
.gitignore U App.jsx U X
src > App.jsx > default
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
9   const handleClick = ()=>{
10     alert("Hey I am clicked");
11   }
12
13   return (
14     <>
15       <div className="button">
16         <button onClick={handleClick}>Click me!</button>
17       </div>
18     </>
19   )
20 }
21
22 export default App
```

Output: when loaded.



Output: when clicked



So, basically it was a very simple example of onClick event, which worked as expected. Now, let's create another event, where on mouse over, the alert box will come:

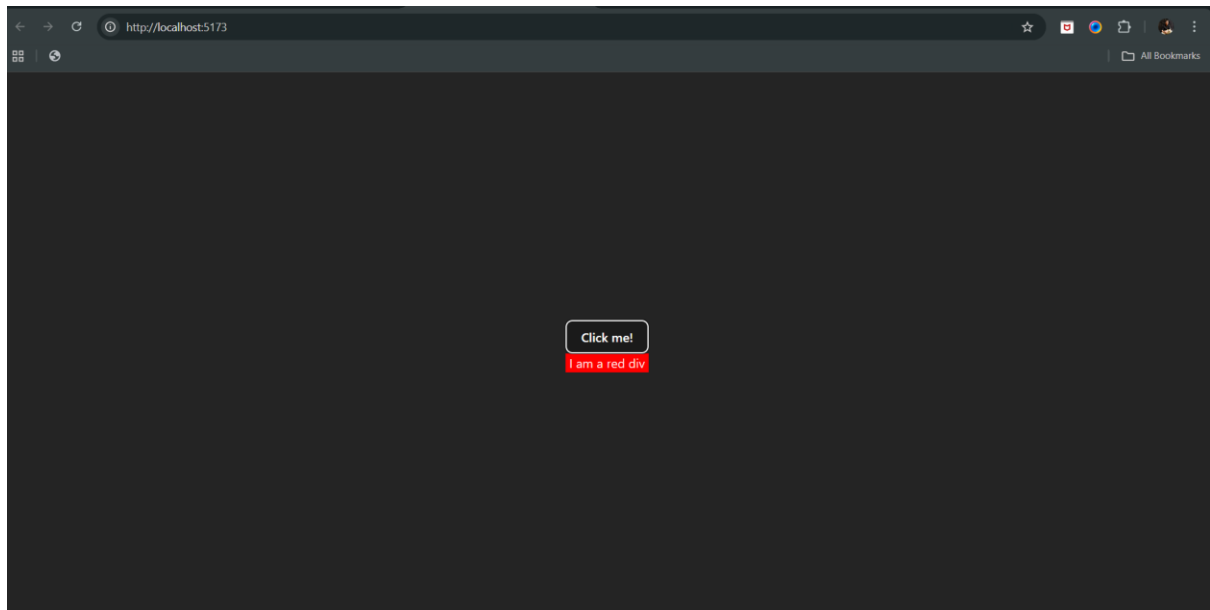
App.jsx:

```
.gitignore U x App.jsx U x
src > App.jsx > ...
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
9   const handleClick = ()=>{
10     alert("Hey I am clicked");
11   }
12   const handleMouseOver = ()=>{
13     alert("I am mouse over");
14   }
15
16   return (
17     <>
18       <div className="button">
19         <button onClick={handleClick}>Click me!</button>
20       </div>
21       <div className="red" onMouseOver={handleMouseOver}>
22         I am a red div
23       </div>
24     </>
25   )
26 }
27
28 export default App
```

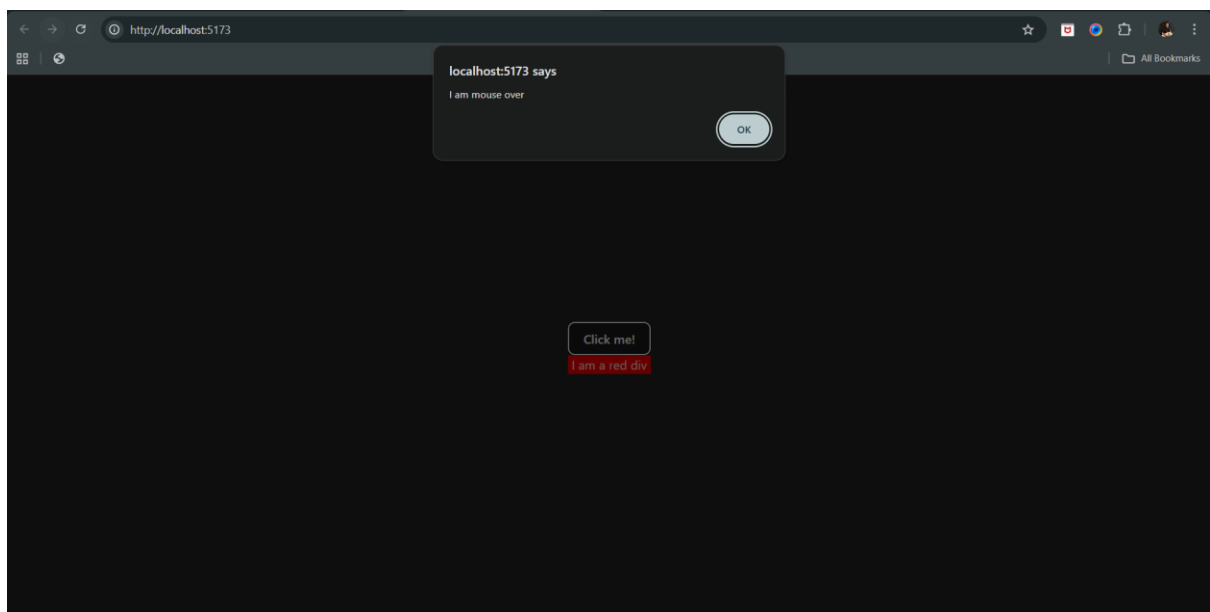
Index.css: add line 16-18

```
File Edit Selection View Go Run Terminal Help Aditya
node_modules public src
assets
App.css U
App.jsx U
index.css U
main.jsx U
.gitignore U
eslint.config.js U
index.html U
package-lock.js... U
package.json U
README.md U
vite.config.js U
src > index.css > .red
1 :root {
2   font-family: system-ui, Avenir, Helvetica, Arial, sans-serif;
3   line-height: 1.5;
4   font-weight: 400;
5
6   color-scheme: light dark;
7   color: rgba(255, 255, 255, 0.87);
8   background-color: #242424;
9
10  font-synthesis: none;
11  text-rendering: optimizeLegibility;
12  -webkit-font-smoothing: antialiased;
13  -moz-osx-font-smoothing: grayscale;
14 }
15
16 .red{
17   background-color: red;
18 }
19 }
```

Output:



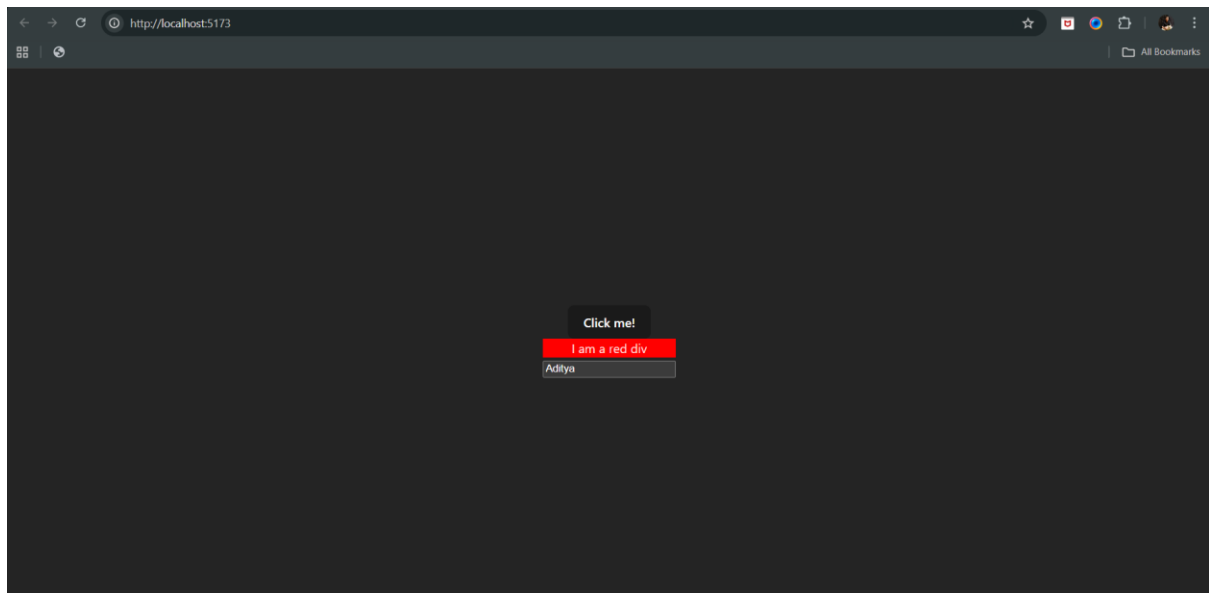
Output: when mouse went over the red box. Clearly, it worked as expected.



Now, what next? Now, we will see the events on the input field, say we added the input field, then App.jsx:

```
.gitignore U App.jsx U index.css U
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 [name, setName] = useState("Aditya")
9
10  const handleClick = ()=>{
11    alert("Hey I am clicked");
12  }
13  const handleMouseOver = ()=>{
14    alert("I am mouse over");
15  }
16  return (
17    <>
18      <div className="button">
19        <button onClick={handleClick}>Click me!</button>
20      </div>
21      <div className="red" onMouseOver={handleMouseOver}>
22        I am a red div
23      </div>
24      <input type="text" value={name}/>
25    </>
26  )
27 }
28 export default App
```

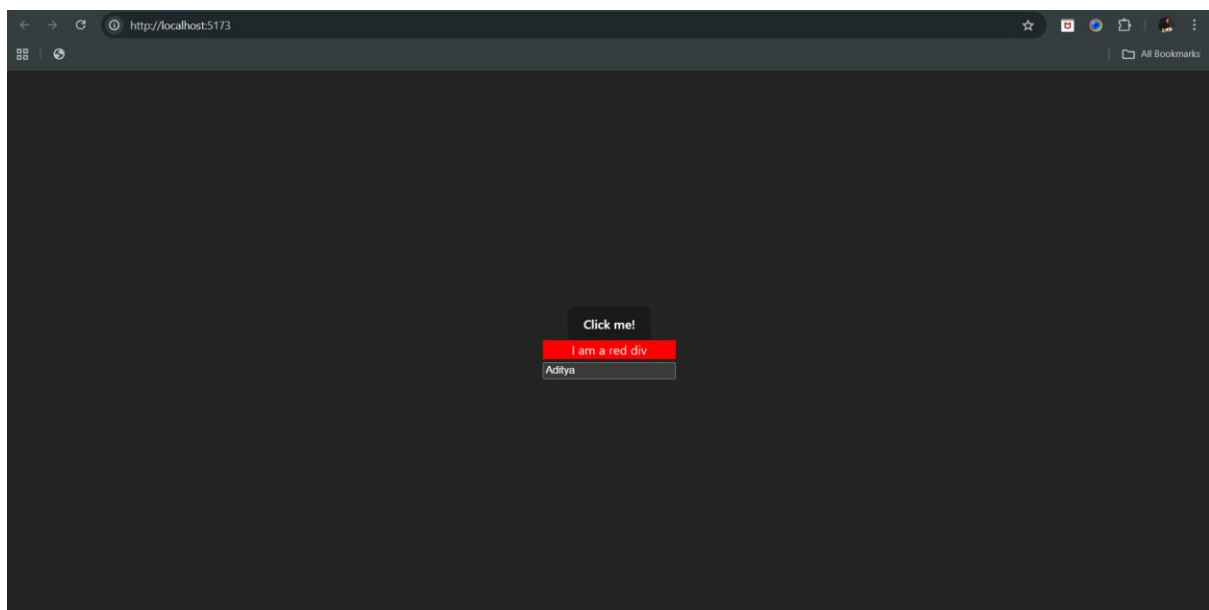
Output: you will observe that the text inside the text box doesn't changes. So, how to solve this issue? We will use onChange event.



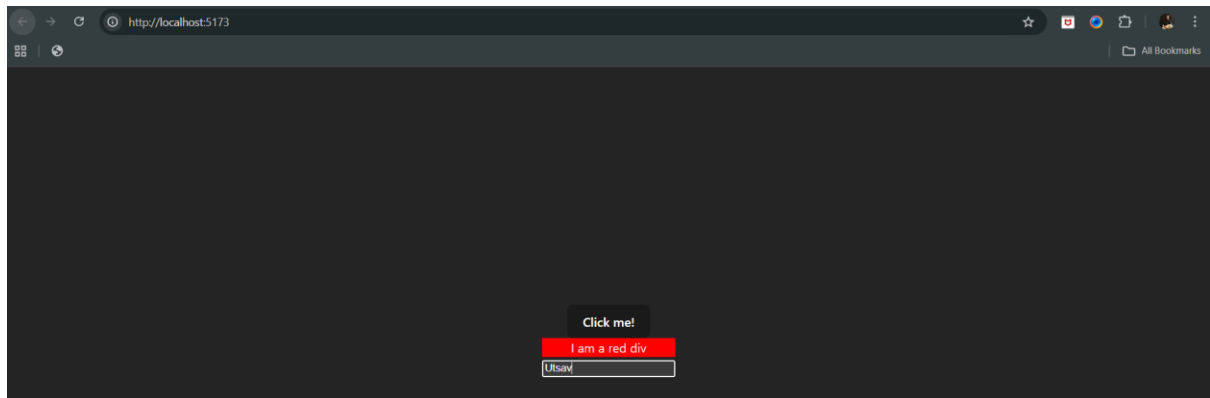
So, updated App.jsx: line 16 to 18 added, and 28 changed.

```
.gitignore U  App.jsx U x  index.css U
src > App.jsx > App
import './App.css'
5
6 function App() {
7   const [count, setCount] = useState(0)
8   const [name, setName] = useState("Aditya")
9
10  const handleClick = ()=>{
11    alert("Hey I am clicked");
12  }
13  const handleMouseOver = ()=>{
14    alert("I am mouse over");
15  }
16  const handleChange = (e)=>{
17    setName(e.target.value)
18  }
19
20  return (
21    <>
22      <div className="button">
23        <button onClick={handleClick}>Click me!</button>
24      </div>
25      <div className="red" onMouseOver={handleMouseOver}>
26        I am a red div
27      </div>
28      <input type="text" value={name} onChange={handleChange}/>
29    </>
30  )
31 }
32 export default App
```

Output: on load



Output: on changing the input field, clearly the text inside the form field is now changed.



--The End--