

Internship Report: Custom Hooks in React (Basic Introduction)

Week 7 | Day 4

By: Hamza Rafique





Internship: ProSensia

What are Custom Hooks?

In React, **Custom Hooks** are JavaScript functions that **start with the word use** and allow you to **reuse stateful logic** across components. They help make your code **cleaner, modular, and more readable**.

Instead of repeating the same logic (like fetching data, handling form inputs, etc.) in multiple components, you can **create one hook** and use it wherever needed.

Why Use Custom Hooks?

-  **Reusability** – Write once, use anywhere
 -  **Cleaner Components** – Keeps logic out of the UI
 -  **Better Abstraction** – Focus on "what" not "how"
 -  **Easier Testing** – Hooks can be unit tested separately
-

Basic Syntax Example

js

CopyEdit

```
// useCounter.js
```

```
import { useState } from 'react';
```

```
function useCounter(initialValue = 0) {
```

```
  const [count, setCount] = useState(initialValue);
```

```
const increment = () => setCount(prev => prev + 1);
const decrement = () => setCount(prev => prev - 1);
const reset = () => setCount(initialValue);

return { count, increment, decrement, reset };
}
```

export default useCounter;

✅ Now, you can use this hook in **any component**:

js

CopyEdit

```
import useCounter from './useCounter';
```

```
const Counter = () => {
  const { count, increment, decrement, reset } = useCounter(5);

  return (
    <div>
      <p>{count}</p>
      <button onClick={increment}>+</button>
      <button onClick={decrement}>-</button>
      <button onClick={reset}>Reset</button>
    </div>
  );
};
```

Rules to Remember

- Custom Hooks **must start with use**
 - They can use other hooks inside (like useState, useEffect)
 - Should be called **only inside React components or other hooks**
-

Conclusion

Custom Hooks are a **powerful pattern** in React for extracting and reusing logic. They help make your components **more focused and maintainable**, and are especially useful as your app grows in size and complexity.