Report Day: 7.4

Submitted By: Ayesha Abid

Submitted To: Ali Hyder Hidayat

Group: Frontend Development @ProSensia

Dated: 07/08/25

Task:

1. What Are Hooks in React?

Hooks are **functions** that let you “hook into” React features like **state** and **lifecycle** in functional components.
Examples: useState, useEffect, useContext, etc.

Introduced in React 16.8, hooks allow writing clean, reusable code **without using classes**.

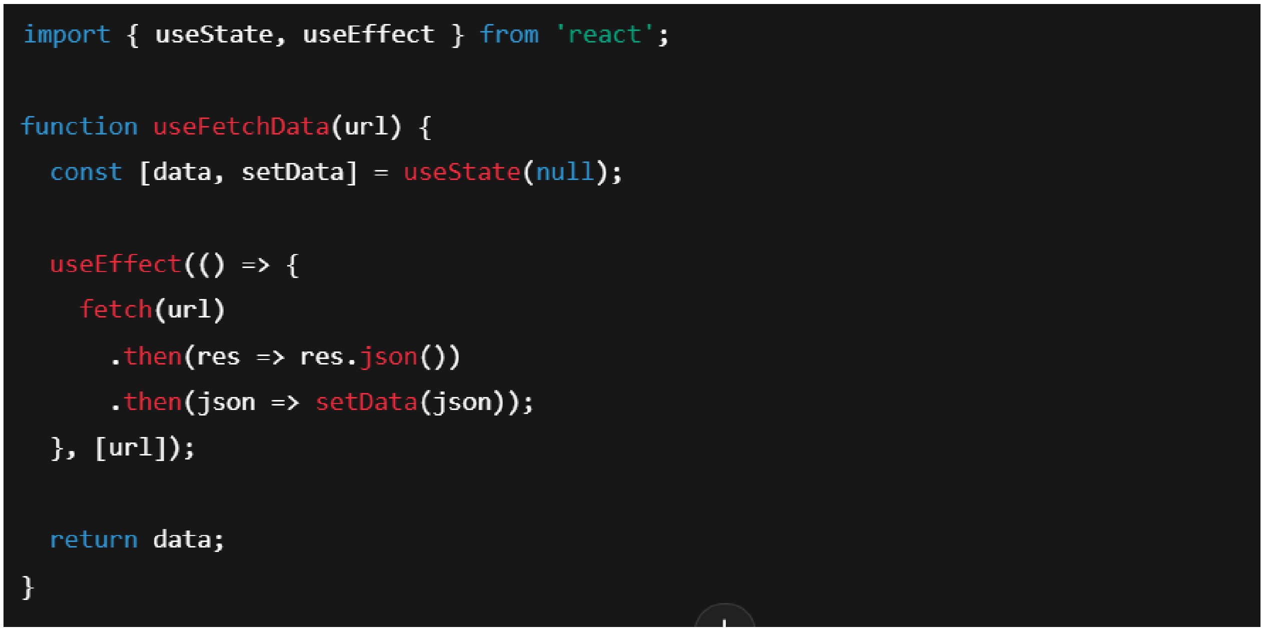
2. What Are Custom Hooks?

Custom Hooks are **user-defined functions** that start with the word use and can **use other hooks inside them**.

They help you **extract and reuse logic** across multiple components.

Why Use Custom Hooks?

- Code reusability
- Cleaner components
- Separation of concerns
- Avoid duplication



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

function useFetchData(url) {
  const [data, setData] = useState(null);

  useEffect(() => {
    fetch(url)
      .then(res => res.json())
      .then(json => setData(json));
  }, [url]);

  return data;
}
```

5. Common Use Cases for Custom Hooks

Hook Name	Use Case
useFetchData	Reusable logic for API calls
useForm	Manage form inputs and validation
useLocalStorage	Sync state with local storage
useAuth	Handle authentication logic
useWindowWidth	Track screen size for responsiveness

Conclusion

Custom Hooks in React are a powerful way to make your components **cleaner**, **shorter**, and **easier to manage**. They allow for better **code reuse**, especially for logic like API calls, form handling, and local storage access.