React State and Variable Declarations

Declaration	Scope	Re-render Behavior	Modern?
`var rotation = 0`	Function	❌ Reset to 0	❌ Avoid
`let rotation = 0`	Block	❌ Reset to 0	☑ Use
`const rotation = 0`	Block	❌ Reset to 0	☑ Use
`const rotation = useRef(0)`	Block	☑ Persists	☑ Use for refs

State Declaration

Components Explained:

```
const [user, setUser] = useState<string | null>(null);
```

`const [user, setUser]` - Array destructuring that extracts two values from useState:

- `user` - The current state value
- `setUser` - Function to update the state

`useState<string | null>` - The hook with TypeScript generic type:

- `string | null` means the state can hold either a string value or null
- This is a union type providing type safety

`(null)` - The initial state value

- State starts as null (no user)

Usage Pattern:

```
const [user, setUser] = useState<string | null>(null);


setUser(null);    // Update state back to null
setUser("Alice"); // Update state to "Alice"
console.log(user); // Initially null


#### Common Use Cases:
This pattern is typical for:
- Authentication state - null when logged out, username when logged in
- User profiles - null when not loaded, user data when available
- Optional data - Data that may or may not exist
```

```
**Gotcha:** Remember that state updates are asynchronous, so `user` won't
immediately reflect the new value after calling `setUser`.
```