Restrict users from directly accessing or manipulating access tokens in a React.js

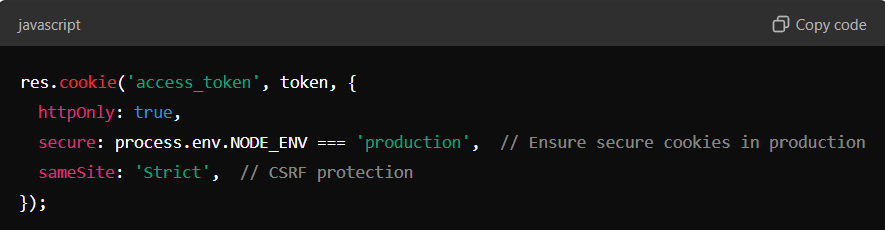
Use HTTP-Only Cookies

Instead of storing access tokens in the frontend (e.g., in localStorage or sessionStorage), use **HTTP-Only cookies**. HTTP-only cookies are not accessible via JavaScript and can only be sent to the server during requests. This reduces the risk of Cross-Site Scripting (XSS) attacks, where malicious scripts could steal access tokens.

**Steps:**

* On the server-side, when the user logs in, return the access token in an HTTP-only cookie.
* In your frontend, you don’t need to handle the token manually. When making API requests to your server, the cookie will automatically be sent with the request.

Example of setting an HTTP-only cookie on the server:



**2. Store Tokens Securely (Avoid localStorage or sessionStorage)**

If using HTTP-only cookies is not an option and you need to store tokens in the frontend, avoid using localStorage or sessionStorage as these are vulnerable to XSS attacks.

Instead, consider **memory storage** for short-lived tokens (store them in a variable in the app). Memory storage is more secure because the token only exists while the app is running and isn’t accessible from JavaScript after a page reload.

