axios for HTTP requests

- **What it is: A promise-based HTTP client for making API calls (GET, POST, PUT, DELETE).**
- **Why it matters: You'll use it to send login and registration data to your backend API, and to fetch protected data after authentication.**
- **Key points:**
  - **How to send POST requests with request bodies.**
  - **How to handle responses and errors with .then() and .catch() or async/await.**
  - **Adding authentication headers (e.g., Authorization: Bearer <token>).**

localStorage for storing tokens

- **What it is: Browser storage that persists even after refreshing the page.**
- **Why it matters: You can store JWT tokens or session tokens here to keep users logged in.**
- **Key points:**
  - **localStorage.setItem("token", value) to store.**
  - **localStorage.getItem("token") to retrieve.**
  - **localStorage.removeItem("token") for logout.**
- **Note: Do not store highly sensitive data in localStorage (like passwords). It's fine for JWT tokens, but consider security implications.**

react-router-dom for navigation and route protection

- **What it is: Library for client-side routing in React.**
- **Why it matters: Lets you define routes (e.g., /login, /dashboard) and control access to them.**
- **Key points:**
  - **BrowserRouter, Routes, and Route components for routing.**
  - **useNavigate() for redirecting users after login or logout.**
  - **Protected routes: Wrap routes in a component that checks for a token before granting access.**

React's useState and useEffect hooks

- **useState: To manage form input fields (e.g., email, password) and authentication state (logged in / logged out).**
- **useEffect: To run side effects like:**
  - **Checking if a user is logged in on app load.**
  - **Fetching user data after authentication.**

Input sanitization and client-side validation

- **What it is: Preventing bad or insecure inputs before sending them to the server.**
- **Why it matters: Improves user experience and security.**
- **Key points:**
  - **Use HTML5 attributes:**
    - **required → ensures a field is filled.**
    - **type="email" → enforces email format.**
    - **pattern="[A-Za-z0-9]{6,}" → for custom regex validation (e.g., password length).**
  - **Additional validation in React (checking passwords match, trimming whitespace, etc.).**
  - **Always validate again on the backend for security.**