

7. Non-Functional Requirements (continued)

- **Performance:**

- Client-side caching for repeated API calls.
- Pagination and infinite scroll for large tables.
- Code splitting and lazy loading for heavy modules (e.g., charts, forms).

- **Security:**

- JWT-based authentication, tokens stored securely (httpOnly cookies preferred).
- Role-based access enforcement at both route and component level.
- Input sanitization to prevent XSS.

- **Scalability:**

- Designed to support multiple roles (admin, user, reviewer).
- Modular architecture to allow addition of new rule categories or workflows.
- Support for future i18n integration (keep UI text externalized).

- **Testing:**

- Unit tests using Jest + React Testing Library for components.
- API mocks for integration testing.
- E2E coverage using Cypress for critical flows (login, rule creation, approval).

- **Maintainability:**

- Clear folder and module structure.
- Consistent coding conventions with ESLint + Prettier.
- Strong typing with TypeScript for all props and API responses.

- **Accessibility:**

- All forms and tables must meet WCAG 2.1 AA standards.
- Semantic HTML for screen reader compatibility.
- Keyboard navigability for all interactive elements.

✓ This completes the **NFR section** of the LLD. Remaining enhancements can include sample component hierarchies or TypeScript types for API integration, depending on project needs.