Front End Development

Last updated by | Perez Martinez, Sheila | Oct 22, 2025 at 9:46 PM GMT+5:30

Child Pages

- Projects
  - Q-web-app

Global Standards

Area	✅ Must	❌ Must Not
Accessibility	Follow https://www.w3.org/WAI/ARIA/apg/ , use semantic HTML, ensure keyboard navigation	Use non-semantic tags for structure (e.g. `<div>` for everything), ignore screen reader support
Input Validation	Validate inputs both client-side and server-side	Rely solely on client-side validation or skip validation entirely
Error Handling	Implement graceful error handling with user-friendly messages	Show raw error messages or fail silently
Performance Optimisation	Minify assets, lazy-load images, reduce re-renders, use CDNs	Load large assets upfront, ignore performance audits
Security & Compliance	Follow OWASP guidelines, sanitise inputs, use HTTPS	Store sensitive data in plain text, ignore XSS/CSRF risks
Documentation	Maintain clear, version-controlled documentation	Leave code undocumented or rely on tribal knowledge
Logging, Monitoring & Observability	Use structured logging, set up alerts and dashboards	Log sensitive data, skip runtime monitoring
Testing	Use unit, integration, and E2E testing with coverage reports	Skip tests or rely only on manual QA
Integration & Compatibility	Ensure compatibility with Edge and other major browsers	Assume all users use Chrome, ignore cross-browser testing
Knowledge Sharing	Share learnings via wikis, workshops, and mentoring	Hoard knowledge or skip onboarding documentation
3rd Party Libraries	Vet libraries for security, size, and maintenance	Use outdated or bloated libraries without review

✅ Must Do

Area	Practice	Why It Matters
Responsive Design	Implement mobile-first design with media queries and fluid layouts	Ensures usability across devices
Performance	Minify assets, lazy-load images, use CDNs	Enhances speed and user experience
Accessibility	Use alt text, ARIA roles, and keyboard navigation	Makes your site usable for all users
Testing	Use E2E tools like Puppeteer	Validates real-world user flows
Version Control	Use Git with clear commit messages and branching strategy	Enables collaboration and rollback safety

❌ Must Not Do

Pitfall	Why to Avoid
Use `<div>` s for everything	Reduces semantic clarity and accessibility
Hard-code styles inline	Makes maintenance harder and breaks separation of concerns
Ignore performance bottlenecks	Leads to slow load times and poor SEO
Skip cross-browser testing	Causes inconsistent behaviour across platforms
Overuse DOM manipulation	Can cause rendering bottlenecks
Neglect accessibility	Excludes users with disabilities and violates standards

System Specific Standards

- React JS library - Overview