FrontEnd Performance Checklist

From Phong Trinh Ha Cameron Proulx Jacob Beaumont Gabriel Laboy

Date Mar 30, 2025

---

1. 1. Component Optimization

- ☐ **Lazy Loading:** Use React's `React.lazy()` for components that aren't immediately needed.
- ☐ **Minimize Re-renders:** Use `React.memo()` to avoid re-rendering components unnecessarily and `useMemo()` for expensive calculations.

- [ ] **Efficient Routing:** Use React Router's `lazy()` to pre-load routes only when necessary.

2. Asset Optimization

- [ ] **Images:** Use compressed image formats like WebP and lazy-load images.
- [ ] **Fonts:** Load only the necessary font subsets and use system fonts when possible.
- [ ] **Defer Scripts:** Use the `async` or `defer` attribute for non-essential JavaScript to load it after the page content.

3. Efficient State & Event Handling

- [ ] **State Management:** Keep state local where possible, use `useReducer` for more complex state, and avoid unnecessary context updates.
- [ ] **Event Handling:** Use event delegation when possible, and avoid making multiple API calls or performing expensive calculations on every event.

4. Performance Profiling & Debugging

- [ ] **DevTools:** Use React Developer Tools to identify unnecessary renders and performance bottlenecks.
- [ ] **React Profiler:** Regularly use the React Profiler to see which components are rendering often and optimize them.

5. Search & Filtering Performance

- [ ] **Pagination:** For large datasets, use server-side pagination to avoid loading everything at once.

6. Deployment Best Practices

- [ ] **Code Splitting:** Use `React.lazy()` and `Suspense` to load only the necessary code for each page.

- [ ] **CDN for Assets:** Serve static assets (images, JS, CSS) from a CDN for faster delivery.