

Product Requirements Document (PRD)

Product Name

Ally Lite

One-Line Description

A web app that turns Lighthouse reports into clear, human-readable performance, accessibility, and interaction-friction insights for non-experts.

Problem Statement

Website owners and non-frontend stakeholders struggle to understand Lighthouse reports. Scores and audit names do not explain: - what is actually broken, - who it impacts, - what should be fixed first.

As a result, performance and accessibility issues remain unfixed despite being measurable.

Target Users

Primary: - Small business owners - Founders - Designers / PMs

Secondary: - Junior developers - Freelancers auditing client sites

Not targeting: - Lighthouse power users - Senior frontend engineers

Goals

- Translate technical audits into **human consequences**
- Prioritize issues instead of listing everything
- Enable non-developers to understand web quality

Non-Goals (Explicit)

- Not replacing Lighthouse
 - Not fixing backend or infrastructure issues
 - Not claiming perfect UX analysis
-

Core User Flow

1. User enters a public URL
 2. App runs Lighthouse analysis
 3. Raw audits are normalized
 4. Gemini generates human-readable explanations
 5. User sees a prioritized report
 6. (Optional) User previews "Fix with Ally" concept
-

Key Features (MVP)

1. URL Scan

- Input: public website URL
- Validation for HTTPS / reachable domain

2. Performance Insights

Metrics covered: - FCP - LCP - CLS - TTI

For each metric: - Single primary root cause - Impact explanation (business + user) - One recommended fix

Example output:

Users on slow networks wait ~3s before seeing content due to an uncompressed hero image.

3. Accessibility Insights

Focus areas: - Missing accessible names - Low contrast text - Keyboard navigation blockers

Each issue includes: - Who is affected - Why the issue matters - Suggested fix (plain English + optional code)

Example:

Screen reader users cannot submit this form because the button has no accessible name.

4. Interaction Friction Detection

Signals (safe, measurable): - Layout shifts - Overlapping elements - Small tap targets

Terminology used: - "Interaction friction" instead of UX theory

5. Issue Prioritization

- Issues ranked by:
- Severity
- User impact
- Ease of fix

Displayed as: 1. Fix now 2. Fix next 3. Can ignore

6. Ally Extension Preview (Concept)

- Toggle showing how Ally *could* fix selected issues via DOM manipulation
- At least one real fix demonstrated (e.g. image dimensions to reduce CLS)

Purpose: - Bridge report → repair - Vision validation, not full implementation

Out of Scope (Hackathon)

- Authentication
 - Historical tracking
 - CI/CD or GitHub integration
 - Mobile app
 - Full accessibility remediation
-

Technical Architecture

Frontend

- Web app UI
- URL input + report viewer

Backend

- Cloud Function triggers Lighthouse
- Results normalized into simplified schema
- Gemini processes normalized data

Storage

- Firebase Firestore for reports
- Shareable report links

AI Usage (Gemini)

Used for: - Translating audits into plain language - Prioritization explanation - Fix suggestions

Not used for: - Raw metric calculation - Scoring logic

Success Metrics (Hackathon-Relevant)

- User understands top issue without explanation
 - Judges can explain report back in their own words
 - Live demo shows visible improvement
-

Risks & Mitigations

Risk: Overpromising UX analysis - Mitigation: Strictly frame as interaction friction

Risk: Gemini hallucination - Mitigation: Constrain prompts to provided audit data only

Risk: Slow scans - Mitigation: Pre-recorded demo fallback

Future Vision (Post-Hackathon)

- Ally browser extension auto-applies low-risk fixes
 - CI integration for non-regressive performance
 - Team-based reports
-

Positioning Statement

Lighthouse tells you what failed. Ally tells you what matters — and what to fix first.