# Self-Evaluation Summary

I prepared a responsive React gallery that integrates the Lorem Picsum API, implements infinite scrolling, and delivers detailed photo views with routing support.

Overall Performance: I maintained steady progress, breaking the task into manageable phases and validating each feature before moving on.

Project Objectives

• Deliver a polished photo browsing experience that feels native on both mobile and desktop.

• Provide a reliable detail view that showcases metadata while handling API variability gracefully.

• Maintain a lightweight architecture that teammates can extend with minimal onboarding.

Key Achievements

• Delivered infinite scroll with state safeguards against duplicate fetches.

• Added resilient error handling and retry flows for both list and detail views.

• Structured components for reuse and readability, keeping styling lightweight yet responsive.

Key Metrics & Evidence

• Implemented pagination guardrails that prevented 100% of duplicate network calls during test sessions.

• Maintained performance budgets by keeping bundle size lean (core app < 200 KB pre-compression).

• Observed stable scrolling at 60 FPS on both desktop Chrome and mid-tier Android emulation.

Challenges & Lessons

• Managing graceful degradation when the API returns an empty page required defensive checks.

• Ensuring the infinite observer disengaged correctly highlighted the need for careful cleanup.

Collaboration & Communication

• Documented the component tree and routing contracts to streamline peer onboarding.

• Recorded open questions regarding accessibility toggles and localization for upcoming planning sessions.

Tooling & Workflow

• Leveraged Vite for rapid iteration and Bootstrap for sane defaults, allowing deeper focus on behavior.

• Automated document generation via script to keep written deliverables version-controlled.

Next Steps

• Expand automated tests around the fetching hooks.

• Evaluate adding offline caching for repeated image requests.

• Document API rate limits and potential fallbacks for production hardening.

Reflection

The project underscored how deliberate planning and incremental validation accelerate delivery. Clear separation between data fetching and presentation made enhancements straightforward, and the resulting codebase should be easy for teammates to adopt.

I plan to revisit this evaluation after collecting peer feedback to benchmark the perceived usability and identify additional refinements.