



EXAMPLE REPORT: STANDARD PACKAGE

Code Audit Report: Alpine Guide

Project:

Alpine Guide - Professional Mountain Guide Marketplace

Audit Scope:

High-level code review, structure analysis, and critical risk identification

Auditors:

Rocksoft Team

Date: February 13, 2026

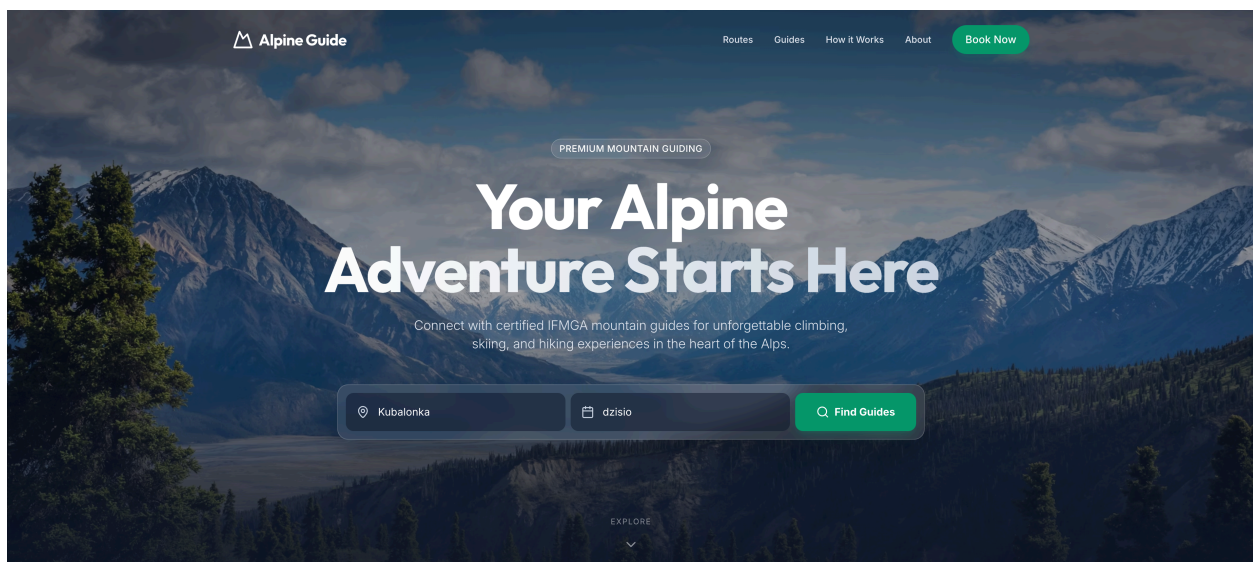
Table of Contents

1. Project overview	3
Project history and context	3
2. Audit objective	4
3. Executive summary	4
Key findings:	4
4. Scope of the audit	5
4.1 Infrastructure and backend	5
4.2 Frontend and SEO	5
4.3 Security and stability	5
5. Infrastructure and backend (Directus CMS)	6
5.1 Server configuration and performance	6
5.2 Directus data modeling	6
5.3 Custom hooks and extensions	7
6. Service and SEO (Next.js Frontend)	7
6.1 Redirect loops and URL structure	7
6.2 Security headers	7
6.3 SEO and Pagination	8
6.4 Media optimization (Cloudinary integration)	8
7. Accessibility and performance	9
7.1 Accessibility (A11y)	9
7.2 Code quality and dependencies	10
8. Conclusion and next steps	10

1. Project overview

The Alpine Guide project is a specialized digital platform designed to connect professional mountain guides with climbers and hikers in the Alpine region.

The system architecture is built on a Headless CMS (Directus) acting as the central data hub, with a Next.js frontend providing a high-performance user interface for booking and route discovery.



Project history and context

Initial phase (3 years ago): The project started as a technical prototype (Proof of Concept) to test the feasibility of real-time guide availability tracking.

Commercial launch (2 years ago): The platform transitioned into a commercial service. While it successfully processed hundreds of bookings, it has been plagued by performance issues and synchronization errors.

Current state: The project was maintained by a single freelancer who is no longer available. The owners now seek to prepare for a potential expansion.

2. Audit objective

The primary goal of this audit is to evaluate the **technical health and scalability** of the Alpine Guide platform. We aim to identify **critical security vulnerabilities**, pinpoint performance bottlenecks. Ultimately, this report provides a technical roadmap to ensure the codebase is stable, maintainable, and ready for a seamless handover to a new development team.

3. Executive summary

The Alpine Guide platform is a successful product that has outgrown its initial prototype architecture. While the choice of Directus and Next.js remains optimal, the system now faces "growth pains" that impact its reliability and speed.

Key findings:

1: System reliability



Core architecture is solid, but legacy "quick fixes" (e.g., synchronous hooks and missing data validation) cause intermittent errors.

2: Performance



Desktop performance is good, but the mobile experience is poor due to unoptimized assets and heavy payloads, affecting SEO and UX.

3: Security



Missing security headers and permissive database policies pose a moderate risk that needs immediate attention.

4: Handover readiness



The code is professional and ready for a new team, though minor documentation gaps exist.

Overall verdict: **"Let's fix the foundation before you go further".**

4. Scope of the audit

This audit covers a deep-dive analysis of the entire application stack:

4.1 Infrastructure and backend

- Analysis of the Directus CMS configuration (collections, relationships, and roles).
- Review of custom Directus extensions, hooks, and endpoints.
- Evaluation of the hosting environment (DigitalOcean/Vercel) and database performance.

4.2 Frontend and SEO

- Full review of the Next.js (App Router) source code and state management.
- Performance testing using Google Lighthouse (Desktop and Mobile).
- Audit of SEO-critical elements: Meta tags, Schema.org implementation, and Core Web Vitals.
- Accessibility (A11y) check for compliance with basic standards.

4.3 Security and stability

- Review of API authentication and Authorization policies (Directus RLS).
- Inspection of third-party integrations (e.g., Stripe for payments, weather API).
- Verification of environment variable management and sensitive data handling.

5. Infrastructure and backend (Directus CMS)

5.1 Server configuration and performance

The Directus instance is hosted on a managed DigitalOcean Droplet.

- **Positive:** The use of Docker containers for the CMS and PostgreSQL database is a great practice, ensuring environment consistency.
- **Issue:** We noticed a lack of Redis caching for API queries. As a result, every request from the Next.js frontend hits the database directly, which causes noticeable latency (TTFB > 800ms) during peak booking hours.



RECOMMENDATION:

Implement Redis as a cache layer for Directus to reduce database load.

5.2 Directus data modeling

The architecture of collections (Guides, Routes, Bookings) is logically designed and uses appropriate relational types (M2M for Guides to Routes).

- **Positive:** Excellent use of 'Directus Translations' interface, which makes the platform ready for multi-language support (German, Italian, French).
- **Issue:** Some critical fields in the 'Bookings' collection lack "Required" validation at the CMS level, leading to corrupted data entries (e.g., bookings without a linked Guide ID).



RECOMMENDATION:

Add field-level validation and "Archive" states to prevent accidental deletion of historical booking data.

5.3 Custom hooks and extensions

- **Issue:** A custom hook responsible for sending email notifications via SendGrid is written in a synchronous way. If the SendGrid API responds slowly, it blocks the entire Directus process, leading to timeout errors for users.



RECOMMENDATION:

Refactor custom logic to use asynchronous background jobs or Directus Flows with error handling.

6. Service and SEO (Next.js Frontend)

6.1 Redirect loops and URL structure

- **Issue:** We identified a redirect loop (301) between /guides and /guides/ when accessed via certain mobile browsers. This negatively impacts both user experience and SEO crawl budget.



RECOMMENDATION:

Normalize trailing slashes in the next.config.js file.

6.2 Security headers

- **Issue:** The application is missing several critical Security Headers (Content-Security-Policy, X-Frame-Options, Permissions-Policy).



RECOMMENDATION:

Configure security headers in the Next.js middleware to protect against XSS and clickjacking attacks.

6.3 SEO and Pagination

- **Issue:** The "Route Discovery" page has ~50 pages of pagination. Currently, all these pages have the same Meta Title ("Discover Alpine Routes"), which Google flags as "Duplicate Content".
- **Issue:** Lack of rel="prev/next" tags and proper canonical URL handling on filtered pages.

RECOMMENDATION:



Dynamically inject the page number into Meta Titles (e.g., "Discover Alpine Routes - Page 2") and implement a robust canonical URL strategy.

6.4 Media optimization (Cloudinary integration)

- **Positive:** The integration with Cloudinary for guide profile photos is excellent. It correctly uses responsive breakpoints and WebP format.
- **Issue:** However, the main hero images on the landing page are still served as 4MB JPEGs directly from the Directus assets folder without any optimization.

RECOMMENDATION:



Pass all CMS-hosted images through an image optimization loader (e.g., Next/Image with a custom loader).

7. Accessibility and performance

We conducted a series of performance tests for key pages. While the Accessibility scores are high (95/100), the Performance on mobile devices is a major concern.

Mobile performance breakdown:

- [Homepage: 42/100](#) (Main issue: Large Hero Image & Unused JS)
- [Guide Profile: 58/100](#) (Main issue: Cumulative Layout Shift from dynamic reviews)
- [Search/Filters: 35/100](#) (Main issue: Heavy Directus API payloads)

Desktop performance breakdown:

- [Homepage: 88/100](#)
- [Guide Profile: 91/100](#)
- [Search/Filters: 72/100](#)

7.1 Accessibility (A11y)

- **Positive:** Excellent use of ARIA labels in the booking form.
- **Issue:** The "Guide Search" filter uses a custom dropdown that is not keyboard-navigable. Users relying on screen readers or keyboards cannot select a region.



RECOMMENDATION:

Replace custom dropdowns with a headless UI library (like Radix UI) that handles accessibility out of the box.

7.2 Code quality and dependencies

- **Issue:** The 'package.json' includes several outdated versions of Next.js and React, containing known security vulnerabilities (CVEs).
- **Issue:** Extensive use of 'any' in TypeScript interfaces for the Directus SDK, which makes the codebase fragile during refactoring.

8. Conclusion and next steps

The Alpine Guide platform is a solid project with a well-chosen technology stack.

While our audit has highlighted areas requiring immediate attention – specifically mobile performance, security headers, and Directus configuration – none of these issues are structural deal-breakers.

The codebase is professional and provides a reliable foundation for future growth.

NEXT STEP:



60-minute online session with our software engineer and technical consultant. During this call, we will walk you through the critical red flags and establish a clear action plan for your development team.

→ [Schedule a meeting](#)