

VIBE96 QUALITY REPORT

Haven — Small Hotel Front Desk & Room Booking

Front-desk software for independent hotels with 5–50 rooms, managing reservations, check-ins and guest payments.

Product ID	**B0183-36274**
Builder	**Redowan Ahmed Farhan**
Version reviewed	**1.4.0 · released 11 Nov 2026**
Stack	**React 19.3.1 + Vite 8.0.0 · JavaScript (Node.js 24+) · SQLite bundled with Node.js 24+; optional Neon PostgreSQL 17 (verified)**
Built with	**OpenAI Codex**
Review period	**9 Nov 2026 – 11 Nov 2026**

79 / 100

Passed with 3 warnings

Pass mark 75 and no failing check.

Scores

Security	79
Code structure	88
Docs & completeness	80
Dependency health	88
Build & deploy	70

✓ **AI code audit**
6 checks · 9 Nov 2026

✓ **Installed & run**
11 Nov 2026

✓ **Demo vs delivered**
11 Nov 2026

✓ **Final sign-off**
11 Nov 2026

At a glance

- ✓ Installed and run from the delivered code — 8 of 8 checks answered yes.
- ✓ Live demo compared with the delivered code — 7 of 7 checks answered yes.
- ! 12 of 15 promised items delivered (setup guide, example config and the rest).
- ✓ Security scored 79/100 with no failing check.
- ! 3 open warnings — explained in part 1.

Check this report is genuine. Scan the code or open vibe96.com/report/QR-26-00113. The online copy always shows the latest status.

PART 1

AI code audit

The full source archive was read by our AI auditor against each check. It reads the code — it does not run it; running it is part 2.

CHECK	WHAT WE FOUND	SCORE	RESULT
Secrets & credentials	Every secret-scan candidate is an environment variable reference, a code path reading from env, or a documentation example. Both .env.example files ship empty values for ADMIN_PASSWORD, SESSION_SECRET, DATABASE_URL, RESEND_API_KEY and DEMO_ACCESS_CODE, and no committed .env, keys or connection strings were seen. Only minor hygiene notes remain.	96	Pass
Dependency vulnerabilities	Node a project file pins every dependency to an exact version and package-lock.json is shipped, with a Dockerfile using `npm ci`. No known critical vulnerability could be confirmed. The README and demo probe disagree on the Vite version, and a few pinned versions could not be checked against advisories from the digest.	88	Pass
Authorisation & access control	The docs describe a single session guard in a project file: only health, public-config and login are reachable before it, and all data routes need a signed, unexpired cookie for the configured role. The route handlers, login and session verification code were not included in the digest, so none of these claims could be verified. The app is single-tenant with one shared admin role, so there is no per-record ownership to check.	70	Warning
Database access rules	The digest never shows a project file, a project file, or the SQL schema files, so the claims of bound queries, versioned writes and idempotent initialization cannot be verified. Nothing visible suggests string-built SQL, and the design (one versioned JSON workspace record) limits tenancy and delete risks. The score is capped because the core of this check was not inspectable.	70	Warning
Code structure & duplication	The project separates the API, storage, config, validation, and feature components into distinct modules, and the shared server app is reused by both Node and Workers. The remaining issues are small: the Dockerfile maintains a hand-written copy list that can drift, and the default port is repeated across several files.	88	Pass
Build reproducibility	The project has a lockfile, pinned dependencies, a multi-stage Dockerfile, and `Node Project/` as a clear build root. However, declared versions conflict across the stack line, README and package.json, and the runtime Docker image and compose file have coherence gaps. HowToSetup.md was not included in the digest, so setup order and prerequisites could not be verified.	70	Warning

Open warnings

Problems that did not stop the listing. They stay on the product page until a later version fixes them.

Warning 1 Database layer not fully visible; no injection seen, but storage and schema files were not provided for verification

The digest never shows a project file, a project file, or the SQL schema files, so the claims of bound queries, versioned writes and idempotent initialization cannot be verified. Nothing visible suggests string-built SQL, and the design (one versioned JSON workspace record) limits tenancy and delete risks. The score is capped because the core of this check was not inspectable.

Warning 2 Authz design looks sound, but api.js and storage code were not provided, so routes could not be traced

The docs describe a single session guard in a project file: only health, public-config and login are reachable before it, and all data routes need a signed, unexpired cookie for the configured role. The route handlers, login and session verification code were not included in the digest, so none of these claims could be verified. The app is single-tenant with one shared admin role, so there is no per-record ownership to check.

Warning 3 **Builds plausibly, but version drift and Docker/compose gaps; HowToSetup.md not verifiable**

The project has a lockfile, pinned dependencies, a multi-stage Dockerfile, and `Node Project/` as a clear build root. However, declared versions conflict across the stack line, README and package.json, and the runtime Docker image and compose file have coherence gaps. HowToSetup.md was not included in the digest, so setup order and prerequisites could not be verified.

Requirement list

What the builder agreed to deliver before uploading — 12 of 15 met.

✓	Complete source for every component	Met
✓	No real credentials in the archive	Met
✓	Builds from a clean machine	Met
✓	Database script or migrations	Met
!	Authentication and roles on every mutating endpoint	Partial
!	Pinned dependencies without critical CVEs (nice)	Partial
✓	Documentation set (nice)	Met
!	Working live demo	Partial
✓	README.md	Met
✓	HowToSetup.md	Met
✓	CLAUDE.md (or AGENTS.md) (nice)	Met
✓	CHANGELOG.md (nice)	Met
✓	.env.example	Met
✓	a project file	Met
✓	a project file (nice)	Met

PART 2

Human review

Reviewers from the Vibe96 team in Dubai worked through each question below; a final reviewer signed it off.

Stage 1 — Installed and run from the delivered code

Completed 11 Nov 2026

A reviewer installed the product on a clean machine, following only the guide that ships with it.

QUESTION	ANSWER	REVIEWER'S NOTE
Packages restored without errors	Yes	
Solution builds clean	Yes	
Database created from the shipped script or migrations	Yes	
First sign-in works with the documented steps	Yes	
Setup guide is accurate and complete	Yes	
Every module in the feature list opens and works	Yes	
No real secrets in the archive	Yes	
Branding and company details can be changed without code	Yes	

Stage 2 — Live demo compared with the delivered code

Completed 11 Nov 2026

A second reviewer checked that what the listing and demo show is what the buyer receives.

QUESTION	ANSWER	REVIEWER'S NOTE
Demo URL is reachable	Yes	
Demo credentials work	Yes	
Demo runs the same version as the archive	Yes	
All five differentiators are visible in the demo	Yes	
Screenshots match the demo	Yes	
Nothing in the demo is missing from the code	Yes	
Datasheet claims are accurate	Yes	

Changes asked for during review

None — approved on the first round.

Sign-off: approved for sale on 11 Nov 2026 with 3 warnings shown publicly.

What this report covers — and what it doesn't

- **It covers** version 1.4.0 exactly as the buyer downloads it: the source archive, the setup guide and the live demo on the dates above.

- **It is not** a penetration test, a performance or load test, or a legal review of third-party package licences.
- **Scores are a point in time.** Libraries age; a later version is audited again and gets its own report.
- **Your setup matters.** Hosting, configuration and changes made after purchase are outside this review.
- **How the score is made:** each AI check is scored 0–100 and weighted (security counts most). A listing needs 75 or more, no failing check, and the human review passed.
- **Always try the live demo** before buying, and ask the builder questions from the product page.

Report history

REPORT	VERSION	DATE	SCORE	STATUS
QR-26-00113	1.4.0	11 Nov 2026	**79**	Current
QR-26-00103	1.3.0	6 Nov 2026	**77**	Replaced
QR-26-00066	1.0.0	26 Oct 2026	**94**	Replaced