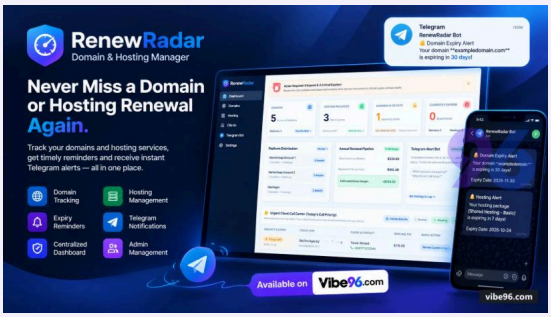


VIBE96 QUALITY REPORT

RenewRadar – Domain & Hosting Manager with Telegram Alerts

Domain, hosting and bundle expiry tracker with Telegram alerts for freelancers and web agencies.

Product ID	**B0699-54776**
Builder	**Naim Siddiqui**
Version reviewed	**1.0.0 · released 16 Nov 2026**
Stack	**React 18.3 with Vite 6 (frontend); Express 4.21 (backend) · JavaScript (Node.js 20+ backend, React 18 JSX frontend) · MariaDB 10.11 (MySQL-compatible via mysql2)**
Built with	**Google Antigravity**
Review period	**26 Oct 2026 – 16 Nov 2026 · 3 rounds of changes**

74 / 100

Listed after a manual review

Pass mark 75 and no failing check.

Scores

Security	77
Code structure	74
Docs & completeness	67
Dependency health	80
Build & deploy	55

✓ **AI code audit**
6 checks · 26 Oct 2026

✓ **Installed & run**
16 Nov 2026

✓ **Demo vs delivered**
16 Nov 2026

✓ **Final sign-off**
16 Nov 2026

At a glance

! Installed and run from the delivered code — 5 of 8 checks answered yes.

✓ Live demo compared with the delivered code — 7 of 7 checks answered yes.

! 10 of 15 promised items delivered (setup guide, example config and the rest).

✓ Security scored 77/100 with no failing check.

! 6 open warnings — explained in part 1.

Check this report is genuine. Scan the code or open vibe96.com/report/QR-26-00141. 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	All regex candidates are environment-variable reads, placeholders, or runtime token handling, with no real keys, tokens or connection strings committed. The .env.example uses clear placeholders. Warnings: docker-compose loads .env.example as the runtime env file, and the README exposes a live API hostname and a local file path.	**88**	Warning
Dependency vulnerabilities	Both the backend and frontend have package.json and package-lock.json, and no wildcard or pre-release versions are used. The main concerns are the node-telegram-bot-api dependency tree (historically pulls in deprecated 'request'-family packages, which the overrides only partly patch), unpinned caret ranges, and no documented audit result. No known critical RCE or auth-bypass in a production path could be confirmed from the manifests.	**80**	Warning
Authorisation & access control	Every route except /auth/login and /health sits behind authMiddleware, and queries in the visible controllers are scoped by user_id. Gaps: the Telegram settings read is open to any user, the mock-mode scoping lets records with no user_id be read or changed by anyone, and there is a single hardcoded admin with no password hashing. Several controllers were truncated, so ownership on some by-id mutations could not be fully verified.	**72**	Warning
Database access rules	Every visible SQL statement uses `?` placeholders and is scoped by `user_id`, and I found no injection path or unscoped delete in the digest. The schema scripts, the db.js inline schema and several controller bodies were not shown, so foreign keys, indexes and cascade rules could not be verified. Search and filter work is done in JavaScript after loading all rows, and the DB auth model is a single hardcoded user_id=1.	**72**	Warning
Code structure & duplication	The backend is split into controllers, services, middleware and routes, which is a sound boundary. However, every controller repeats the same user-scoping, mock-vs-DB branching and days-remaining status logic, and status thresholds and tier values are hard-coded rather than configured.	**74**	Warning
Build reproducibility	The source projects have package.json files, lockfiles and Dockerfiles, so npm install and build should work. The README, CLAUDE.md and the compose setup are inconsistent with the actual layout and env handling. A clean-machine Docker deploy would likely fail or run with placeholder credentials, and local setup commands point at folders that do not exist.	**55**	Warning

Open warnings

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

Warning 1 **No real credentials found; minor weak-default and config hygiene issues**

All regex candidates are environment-variable reads, placeholders, or runtime token handling, with no real keys, tokens or connection strings committed. The .env.example uses clear placeholders. Warnings: docker-compose loads .env.example as the runtime env file, and the README exposes a live API hostname and a local file path.

Warning 2 **Reasonable module layout, but duplicated filtering logic, mock/DB branching in every controller and a few hard-coded values**

The backend is split into controllers, services, middleware and routes, which is a sound boundary. However, every controller repeats the same user-scoping, mock-vs-DB branching and days-remaining status logic, and status thresholds and tier values are hard-coded rather than configured.

Warning 3 **Lockfiles present, caret ranges only; node-telegram-bot-api is deprecated-dependency heavy, no critical CVEs confirmed**

Both the backend and frontend have package.json and package-lock.json, and no wildcard or pre-release versions are used. The main concerns are the node-telegram-bot-api dependency tree (historically pulls in deprecated 'request'-family packages, which the overrides only partly patch), unpinned caret ranges, and no documented audit result. No known critical RCE or auth-bypass in a production path could be confirmed from the manifests.

Warning 4 **Folder names mismatch docs; compose env wiring and web build are doubtful; HowToSetup.md not reviewed**

The source projects have package.json files, lockfiles and Dockerfiles, so npm install and build should work. The README, CLAUDE.md and the compose setup are inconsistent with the actual layout and env handling. A clean-machine Docker deploy would likely fail or run with placeholder credentials, and local setup commands point at folders that do not exist.

Warning 5 **Queries are parameterised and user-scoped, but schema files were not reviewed and there are N+1/ in-memory filter hot paths**

Every visible SQL statement uses `?` placeholders and is scoped by `user_id`, and I found no injection path or unscoped delete in the digest. The schema scripts, the db.js inline schema and several controller bodies were not shown, so foreign keys, indexes and cascade rules could not be verified. Search and filter work is done in JavaScript after loading all rows, and the DB auth model is a single hardcoded user_id=1.

Warning 6 **Routes authenticated, but role checks are weak, no ownership model, and the user concept is hardcoded**

Every route except /auth/login and /health sits behind authMiddleware, and queries in the visible controllers are scoped by user_id. Gaps: the Telegram settings read is open to any user, the mock-mode scoping lets records with no user_id be read or changed by anyone, and there is a single hardcoded admin with no password hashing. Several controllers were truncated, so ownership on some by-id mutations could not be fully verified.

Requirement list

What the builder agreed to deliver before uploading — 10 of 15 met.

✓	Complete source for every component	Met
!	No real credentials in the archive	Partial
!	Builds from a clean machine	Partial
✓	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 16 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	No	
Every module in the feature list opens and works	No	
No real secrets in the archive	Yes	
Branding and company details can be changed without code	No	

Stage 2 — Live demo compared with the delivered code

Completed 16 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

Round 1: changes requested on 3 Nov 2026; the builder fixed them and uploaded again.

Round 2: changes requested on 10 Nov 2026; the builder fixed them and uploaded again.

Sign-off: approved for sale on 16 Nov 2026 with 6 warnings shown publicly.

What this report covers — and what it doesn't

- **It covers** version 1.0.0 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.