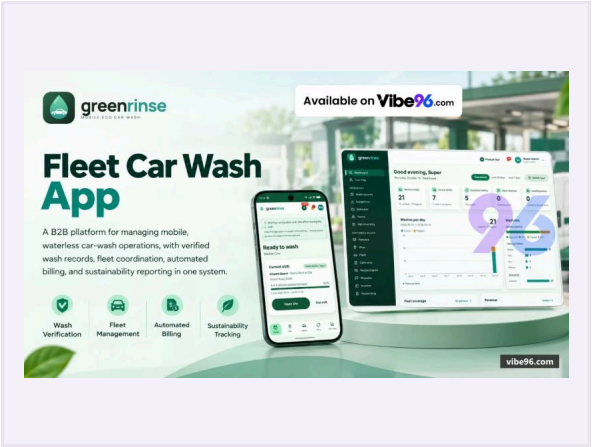


This report has been replaced by QR-26-00072 — a newer version of the product was reviewed.



VIBE96 QUALITY REPORT

GreenRinse

Mobile car-wash management for fleet operators: offline-first washer app, admin dashboard, partner portal.

Product ID	B0003-25486
Builder	Max code
Version reviewed	1.0.0 · released 20 Oct 2026
Stack	ASP.NET Core 10 + React 18.3 + TypeScript 5.9 · C# 13 / .NET 10 · SQL Server 2019+
Built with	Claude Code
Review period	20 Oct 2026 – 22 Oct 2026

86 / 100

Passed with 5 warnings

Pass mark 75 and no failing check.

Scores

Security	92
Code structure	72
Docs & completeness	100
Dependency health	85
Build & deploy	72

✓ AI code audit 6 checks · 20 Oct 2026	✓ Installed & run 20 Oct 2026	✓ Demo vs delivered 20 Oct 2026	✓ Final sign-off 22 Oct 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.
- ✓ 15 of 15 promised items delivered (setup guide, example config and the rest).
- ✓ Security scored 92/100 with no failing check.
- ! 5 open warnings — explained in part 1.



Check this report is genuine. Scan the code or open vibe96.com/report/QR-26-00051. 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	No real credentials were discovered. However, several example/template files contain weak or realistic-looking placeholder values that could be mistaken for production secrets. The docker-compose setup uses environment variables correctly, but .env.example contains instructional placeholders that should be more obviously fake.	92	Warning
Dependency vulnerabilities	The project uses .NET 10 (preview/RC) and npm packages with exact versions in package-lock.json. No known critical vulnerabilities found in the listed dependencies. Several packages are outdated (React 18, Vite 5, .NET packages from 2024) but not critically vulnerable. The use of .NET 10 preview in production is a concern.	85	Warning
Authorisation & access control	All mutating endpoints require authentication and role-based policies. Passwords use proper hashing (BCrypt). Global query filters enforce partner data isolation. Minor gaps: some read endpoints (GET /api/users/{id}, GET /api/partners/{id}, GET /api/vehicles/{id}) lack explicit ownership checks beyond query filters, and the Partner API v1 controller has weaker validation on some lookups.	92	Warning
Database access rules	All database access goes through EF Core with parameterised queries. Global query filters enforce tenant/partner scoping and soft-delete. Migrations are complete and reversible. Minor findings: a few foreign keys lack indexes (VehicleId on WashRecord, PartnerId on several tables), and one temporal table (FleetVehicle) could benefit from an index on the period columns for history queries.	92	Pass
Code structure & duplication	The backend follows clean layering (Domain → Application → Infrastructure → Api) with good separation of concerns. However, the frontend has significant duplication across three app folders (admin, partner, operator-pwa), and several files exceed 500 lines. Some hard-coded values and configuration belong in environment or settings tables.	72	Warning
Build reproducibility	The .NET backend builds cleanly with consistent SDK/framework versions and proper project references. However, the frontend npm workspaces lack a root workspace configuration file, making `npm ci` at the root ambiguous. HowToSetup.md jumps between backend and frontend without clarifying the frontend build must happen from the workspace root, and the Docker setup references environment variables not fully documented in .env.example.	72	Warning

Open warnings

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

Warning 1 Outdated dependencies present; no critical vulnerabilities detected

The project uses .NET 10 (preview/RC) and npm packages with exact versions in package-lock.json. No known critical vulnerabilities found in the listed dependencies. Several packages are outdated (React 18, Vite 5, .NET packages from 2024) but not critically vulnerable. The use of .NET 10 preview in production is a concern.

Warning 2 Weak default secrets in example files; no real credentials found

No real credentials were discovered. However, several example/template files contain weak or realistic-looking placeholder values that could be mistaken for production secrets. The docker-compose setup uses environment variables correctly, but .env.example contains instructional placeholders that should be more obviously fake.

Warning 3 Strong auth/authz with minor ownership gaps on read endpoints

All mutating endpoints require authentication and role-based policies. Passwords use proper hashing (BCrypt). Global query filters enforce partner data isolation. Minor gaps: some read endpoints (GET /api/users/{id}, GET /api/partners/{id}, GET /api/vehicles/{id}) lack explicit ownership checks beyond query filters, and the Partner API v1 controller has weaker validation on some lookups.

Warning 4 Build reproducible with gaps: missing npm workspace config, unclear frontend build order

The .NET backend builds cleanly with consistent SDK/framework versions and proper project references. However, the frontend npm workspaces lack a root workspace configuration file, making `npm ci` at the root ambiguous. HowToSetup.md jumps between backend and frontend without clarifying the frontend build must happen from the workspace root, and the Docker setup references environment variables not fully documented in .env.example.

Warning 5 Moderate duplication and some oversized files; business logic mostly well-layered but frontend has scattered concerns.

The backend follows clean layering (Domain → Application → Infrastructure → Api) with good separation of concerns. However, the frontend has significant duplication across three app folders (admin, partner, operator-pwa), and several files exceed 500 lines. Some hard-coded values and configuration belong in environment or settings tables.

Requirement list

What the builder agreed to deliver before uploading — 15 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	Met
✓ Pinned dependencies without critical CVEs (nice)	Met
✓ Documentation set (nice)	Met
✓ Working live demo	Met
✓ README.md	Met
✓ HowToSetup.md	Met
✓ CLAUDE.md (or AGENTS.md) (nice)	Met
✓ CHANGELOG.md (nice)	Met
✓ appsettings.example.json	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 20 Oct 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 20 Oct 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 22 Oct 2026 with 5 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.

Report history

REPORT	VERSION	DATE	SCORE	STATUS
QR-26-00072	1.2.0	27 Oct 2026	77	Current
QR-26-00051	1.0.0	22 Oct 2026	86	Replaced