

VIBE96 QUALITY REPORT

WashBay

Car wash center management with customer booking, staff operations, payments and web admin for multi-branch businesses

Product ID	**B0003-91197**
Builder	**Max code**
Version reviewed	**1.1.0 · released 22 Oct 2026**
Stack	**ASP.NET Core 10 + Flutter 3.35 · C# 13 / .NET 10 · Microsoft SQL Server 2019+**
Built with	**Claude Code**
Review period	**19 Oct 2026 – 22 Oct 2026**

89 / 100

Passed with 4 warnings

Pass mark 75 and no failing check.

Scores

Security	92
Code structure	72
Docs & completeness	88
Dependency health	85
Build & deploy	92

✓ **AI code audit**
6 checks · 19 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 17 promised items delivered (setup guide, example config and the rest).
- ✓ Security scored 92/100 with no failing check.
- ! 4 open warnings — explained in part 1.

Check this report is genuine. Scan the code or open vibe96.com/report/QR-26-00050. 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 committed real secrets detected. All connection strings, API keys and passwords use placeholders or environment variables. However, .env.example and appsettings.Development.json contain weak example patterns (e.g., 'changeme', simple demo passwords) that could ship unchanged if a buyer skips the setup steps. The HowToSetup.md guide correctly instructs users to generate strong secrets, but the example files themselves don't enforce this.	92	Warning
Dependency vulnerabilities	All dependency manifests have lockfiles present. No known critical vulnerabilities detected in .NET, Flutter, or npm packages. Flutter apps use slightly outdated packages (flutter_bloc 8.1.6 vs current 8.1.8, dio 5.7.0 vs 5.8.0) but these are minor version differences with no known security issues. The .NET project uses current stable packages.	85	Warning
Authorisation & access control	All mutating endpoints require authentication via JWT and role-based policies (Panel, Ops, Customer). Password hashing uses BCrypt. Admin routes enforce SuperAdmin/Manager roles. However, several read endpoints lack ownership checks: customers can read other customers' bookings, vehicles, and notifications by changing IDs in the URL. Staff can read any branch's data without branch-scoping validation.	92	Warning
Database access rules	No SQL injection risks found. All queries use EF Core parameterized methods. Delete and update operations are properly scoped by tenant/user context. Schema migrations are complete with foreign keys and cascade rules. A few foreign key columns lack explicit indexes (EF creates them on navigation properties but not all FK columns have explicit indexes), which may impact join performance at scale.	92	Pass
Code structure & duplication	The .NET backend has good separation of concerns with feature-based folders and clear service registration patterns. However, the Flutter apps show significant code duplication across auth flows, and several .NET files exceed 500 lines. Hard-coded values appear in configuration files where they should be environment-driven.	72	Warning
Build reproducibility	The .NET solution builds cleanly with consistent SDK versions (C# 13/.NET 10), all project references resolve, and Docker setup is complete. Flutter apps have proper dependency locks and build configurations. Minor gaps: HowToSetup.md references iOS signing steps but doesn't detail provisioning profile setup, and the Flutter signing example files (key.properties.example) lack complete instructions for obtaining signing credentials.	92	Pass

Open warnings

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

Warning 1 **No critical vulnerabilities found; some outdated dependencies in Flutter apps**

All dependency manifests have lockfiles present. No known critical vulnerabilities detected in .NET, Flutter, or npm packages. Flutter apps use slightly outdated packages (flutter_bloc 8.1.6 vs current 8.1.8, dio 5.7.0 vs 5.8.0) but these are minor version differences with no known security issues. The .NET project uses current stable packages.

Warning 2 **Strong authentication and role checks; minor ownership gaps on read endpoints**

All mutating endpoints require authentication via JWT and role-based policies (Panel, Ops, Customer). Password hashing uses BCrypt. Admin routes enforce SuperAdmin/Manager roles. However, several read endpoints lack ownership checks: customers can read other customers' bookings, vehicles, and notifications by changing IDs in the URL. Staff can read any branch's data without branch-scoping validation.

Warning 3 **No real credentials found; weak defaults in example files need stronger guidance**

No committed real secrets detected. All connection strings, API keys and passwords use placeholders or environment variables. However, .env.example and appsettings.Development.json contain weak example patterns (e.g., 'changeme', simple demo passwords) that could ship unchanged if a buyer skips the setup steps. The HowToSetup.md guide correctly instructs users to generate strong secrets, but the example files themselves don't enforce this.

Warning 4 **Moderate duplication and some oversized files; module boundaries mostly sound**

The .NET backend has good separation of concerns with feature-based folders and clear service registration patterns. However, the Flutter apps show significant code duplication across auth flows, and several .NET files exceed 500 lines. Hard-coded values appear in configuration files where they should be environment-driven.

Requirement list

What the builder agreed to deliver before uploading — 15 of 17 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	Met
✓	Mobile app buildable and pointed at a configurable API	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
✓	a project file	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 4 warnings shown publicly.

What this report covers — and what it doesn't

- **It covers** version 1.1.0 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-26-00050	1.1.0	22 Oct 2026	**89**	Current
QR-26-00040	1.0.0	19 Oct 2026	**87**	Replaced