

VIBE96 QUALITY REPORT

Serveo App

Multi-branch food delivery system with customer, rider, kitchen and HQ portals plus native mobile apps

Product ID	**B0007-19064**
Builder	**dev-studio**
Version reviewed	**1.0.0 · released 12 Oct 2026**
Stack	**ASP.NET Core 9 MVC + Flutter 3.24 · C# 12 (.NET 9) · SQL Server 2022**
Built with	**Claude Code, Cursor**
Review period	**11 Oct 2026 – 12 Oct 2026**

70 / 100

Listed after a manual review

Pass mark 75 and no failing check.

Scores

Security	71
Code structure	62
Docs & completeness	88
Dependency health	72
Build & deploy	72

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

✓ **Installed & run**
12 Oct 2026

✓ **Demo vs delivered**
12 Oct 2026

✓ **Final sign-off**
12 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 71/100.

! 5 open warnings — explained in part 1.

Check this report is genuine. Scan the code or open vibe96.com/report/QR-26-00036. 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 live API keys or production credentials were found. However, the codebase ships with weak default passwords ('Royex@2026') hardcoded in multiple files, including database seeds and configuration examples. These defaults are documented as demo credentials but ship to buyers unchanged, creating a security risk if deployed without modification.	75	Warning
Dependency vulnerabilities	The Flutter app has a proper pubspec.lock with pinned dependencies. The ASP.NET Core backend has no packages.lock.json, relying on floating NuGet package versions which can introduce breaking changes or vulnerabilities between builds. Several Flutter packages are outdated (google_maps_flutter 2.2.1 is 2+ years old, current is 2.9.0+). No known critical vulnerabilities detected in the versions present.	72	Warning
Authorisation & access control	Minor points were sent to the builder to improve; none of them stopped the listing.	45	Fail
Database access rules	All SQL uses parameterized queries via ADO.NET with no string concatenation of user input. Row-level security enforces branch isolation. DELETE/UPDATE statements are properly scoped. The schema is complete and can be created from empty. Minor issues: some foreign key columns lack explicit indexes (SQL Server auto-indexes FK constraints but explicit indexes improve query plans), and a few stored procedures could benefit from additional input validation.	92	Warning
Code structure & duplication	The project exhibits multiple structural issues: hard-coded demo passwords and connection strings appear in 15+ files instead of being centralized; several controller and service files exceed 500 lines with mixed concerns; SQL queries are duplicated across controllers; and configuration values are scattered rather than injected. While the architecture is sound, maintainability is compromised by these patterns.	62	Warning
Build reproducibility	The backend can build after database setup, but HowToSetup.md jumps straight to 'dotnet run' without 'dotnet restore', risking first-run failures. Flutter setup.sh/bat are present but the main README and HowToSetup skip the Flutter build entirely. The connection string example uses localhost\SQLEXPRESS but docker-compose uses 'db', creating confusion about which environment the docs target.	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 **Flutter lockfile present but .NET backend has no lockfile; several outdated packages**

The Flutter app has a proper pubspec.lock with pinned dependencies. The ASP.NET Core backend has no packages.lock.json, relying on floating NuGet package versions which can introduce breaking changes or vulnerabilities between builds. Several Flutter packages are outdated (google_maps_flutter 2.2.1 is 2+ years old, current is 2.9.0+). No known critical vulnerabilities detected in the versions present.

Warning 2 **Database access is parameterized and scoped; minor schema gaps in foreign key indexes**

All SQL uses parameterized queries via ADO.NET with no string concatenation of user input. Row-level security enforces branch isolation. DELETE/UPDATE statements are properly scoped. The schema is complete and can be created from empty. Minor issues: some foreign key columns lack explicit indexes (SQL Server auto-indexes FK constraints but explicit indexes improve query plans), and a few stored procedures could benefit from additional input validation.

Warning 3 **Significant code duplication, oversized files, and hard-coded values scattered across codebase**

The project exhibits multiple structural issues: hard-coded demo passwords and connection strings appear in 15+ files instead of being centralized; several controller and service files exceed 500 lines with mixed concerns; SQL queries are duplicated across controllers; and configuration values are scattered rather than injected. While the architecture is sound, maintainability is compromised by these patterns.

Warning 4 **Weak default credentials and demo passwords committed; no live secrets found**

No live API keys or production credentials were found. However, the codebase ships with weak default passwords ('Royex@2026') hardcoded in multiple files, including database seeds and configuration examples. These defaults are documented as demo credentials but ship to buyers unchanged, creating a security risk if deployed without modification.

Warning 5 **Build reproducible with gaps: missing .NET restore step, Flutter setup incomplete, connection string unclear**

The backend can build after database setup, but HowToSetup.md jumps straight to 'dotnet run' without 'dotnet restore', risking first-run failures. Flutter setup.sh/bat are present but the main README and HowToSetup skip the Flutter build entirely. The connection string example uses localhost\SQLEXPRESS but docker-compose uses 'db', creating confusion about which environment the docs target.

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	Partial
✓	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)	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 12 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 12 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 12 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.