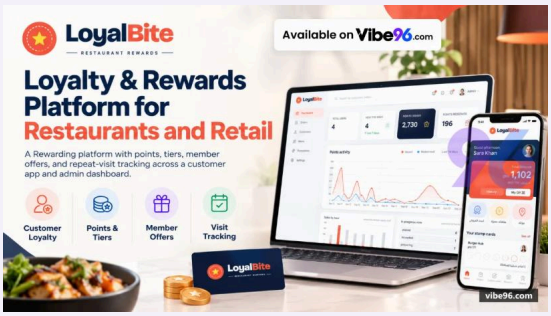
This report has been replaced by QR-26-00074 — a newer version of the product was reviewed.



VIBE96 QUALITY REPORT

LoyalBite

Multi-brand restaurant loyalty platform with POS-initiated transactions and mobile customer app

Product ID	**B0003-72659**
Builder	**Max code**
Version reviewed	**1.0.0 · released 6 Oct 2026**
Stack	**ASP.NET Core 10 MVC + EF Core 10 · C# 12 (.NET 10) · SQL Server 2019**
Built with	**Claude Code**
Review period	**6 Oct 2026 – 6 Oct 2026**

86 / 100

Passed with 5 warnings

Pass mark 75 and no failing check.

Scores

Security	93
Code structure	72
Docs & completeness	94
Dependency health	72
Build & deploy	78

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

✓ **Installed & run**
6 Oct 2026

✓ **Demo vs delivered**
6 Oct 2026

✓ **Final sign-off**
6 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.
- ! 16 of 17 promised items delivered (setup guide, example config and the rest).
- ✓ Security scored 93/100 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-26-00016. 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 found. However, the package ships with a placeholder Firebase google-services.json that will cause runtime errors if not replaced, and demo seed data uses weak passwords ('[hidden]'). The .env.example and appsettings.example.json correctly use placeholders, and the actual secrets files are properly gitignored.	**92**	Warning
Dependency vulnerabilities	The .NET project lacks a packages.lock.json, making builds non-reproducible. Both Flutter apps use outdated packages (some 1-2 years old) but no known critical vulnerabilities were identified in the versions present. The customer app's firebase_messaging and other Firebase packages are significantly outdated.	**72**	Warning
Authorisation & access control	All mutating and admin endpoints are properly guarded with authentication and role checks. The POS API correctly restricts staff to their own restaurant. Minor gaps: customer read endpoints (favorites, notifications, stamp history) trust the JWT uid claim without explicit ownership validation in queries, and the POS / transaction/history endpoint lacks pagination which could expose large datasets.	**92**	Warning
Database access rules	All database access uses EF Core LINQ with parameterized queries. No raw SQL, string concatenation or injection paths found. Delete and update operations are properly scoped by user/restaurant. Schema has comprehensive foreign keys and indexes. Minor: a few foreign key columns lack explicit indexes in the migration, though EF Core creates them automatically.	**95**	Pass
Code structure & duplication	The project has good separation of concerns overall (controllers, services, repositories), but contains significant duplication in Flutter apps (especially Dio client setup and token handling), oversized service files in the backend, and hard-coded configuration scattered across multiple files. Several utility and API files exceed 300 lines and mix concerns.	**72**	Warning
Build reproducibility	The project can be built and run from source with clear prerequisites. However, HowToSetup.md has ordering issues (references Firebase before explaining it), the customer app's Firebase placeholder will cause runtime failures, and some environment variable examples could be clearer. Docker compose works but the manual setup path has small gaps.	**78**	Warning

Open warnings

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

Warning 1 **Weak default passwords and placeholder Firebase config present**

No real credentials found. However, the package ships with a placeholder Firebase google-services.json that will cause runtime errors if not replaced, and demo seed data uses weak passwords ('[hidden]'). The .env.example and appsettings.example.json correctly use placeholders, and the actual secrets files are properly gitignored.

Warning 2 **Flutter apps have outdated dependencies; .NET project has no lockfile**

The .NET project lacks a packages.lock.json, making builds non-reproducible. Both Flutter apps use outdated packages (some 1-2 years old) but no known critical vulnerabilities were identified in the versions present. The customer app's firebase_messaging and other Firebase packages are significantly outdated.

Warning 3 **Strong authentication and authorization with minor ownership check gaps on read endpoints**

All mutating and admin endpoints are properly guarded with authentication and role checks. The POS API correctly restricts staff to their own restaurant. Minor gaps: customer read endpoints (favorites, notifications, stamp history) trust the JWT uid claim without explicit ownership validation in queries, and the POS /transaction/history endpoint lacks pagination which could expose large datasets.

Warning 4 **Moderate code duplication and some oversized files; structure generally sound but needs refactoring**

The project has good separation of concerns overall (controllers, services, repositories), but contains significant duplication in Flutter apps (especially Dio client setup and token handling), oversized service files in the backend, and hard-coded configuration scattered across multiple files. Several utility and API files exceed 300 lines and mix concerns.

Warning 5 **Build reproducible with minor setup gaps and missing Firebase configuration**

The project can be built and run from source with clear prerequisites. However, HowToSetup.md has ordering issues (references Firebase before explaining it), the customer app's Firebase placeholder will cause runtime failures, and some environment variable examples could be clearer. Docker compose works but the manual setup path has small gaps.

Requirement list

What the builder agreed to deliver before uploading — 16 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	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 6 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 6 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 6 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-26-00074	1.3.0	27 Oct 2026	**85**	Current
QR-26-00016	1.0.0	6 Oct 2026	**86**	Replaced