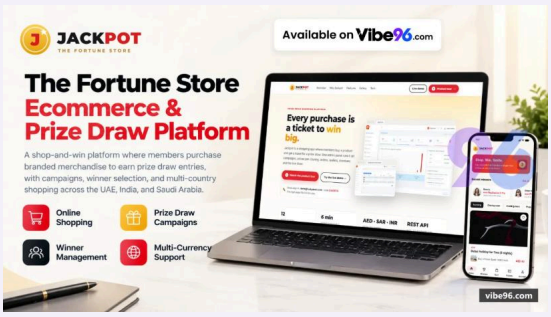


VIBE96 QUALITY REPORT

Jackpot

Prize-draw e-commerce platform where product purchases become raffle tickets; ASP.NET Core + Flutter + SQL Server.

Product ID	**B0005-89299**
Builder	**Chow Studio**
Version reviewed	**1.0.0 · released 23 Oct 2026**
Stack	**ASP.NET Core 10 (MVC + Minimal API) · C# 12 / .NET 10 · SQL Server 2019+**
Built with	**Claude Code**
Review period	**23 Oct 2026 – 23 Oct 2026**

89 / 100

Passed with 4 warnings

Pass mark 75 and no failing check.

Scores

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

✓ **AI code audit**
6 checks · 23 Oct 2026

✓ **Installed & run**
23 Oct 2026

✓ **Demo vs delivered**
23 Oct 2026

✓ **Final sign-off**
23 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.
- ✓ 17 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-26-00060. 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. The demo admin password 'Admin@123' is weak and ships in seed.sql; buyers must change it immediately. Example files (.env.example, appsettings.example.json) contain placeholder secrets that are correctly excluded from deployment but should be clearer. All other secrets use environment variables or user input as intended.	92	Warning
Dependency vulnerabilities	All dependencies are pinned in lockfiles. .NET packages are current (.NET 10, latest stable versions). Flutter dependencies are 6-12 months old but no known critical vulnerabilities found. Some minor updates available (riverpod 2.6.1→latest, go_router 14.6.2→latest). No wildcard versions or missing lockfiles.	85	Warning
Authorisation & access control	All mutating and admin endpoints require authentication with JWT (API) or cookie (admin panel). Role checks are enforced via policies. Passwords use [hidden] with 100k iterations. Minor gaps: member read endpoints (orders, tickets, wallet) lack explicit ownership checks in some service methods, relying on controller-level filtering. No critical exposure foun	92	Warning
Database access rules	All SQL uses Dapper with parameters only—no concatenation or injection paths found. Delete and update statements correctly scope by ID or member. The schema is complete and can be created from empty. Minor issues: a few foreign keys lack indexes (Country on Member, Product, Campaign), and one wallet update uses string interpolation for the column name (safe but inconsistent).	92	Pass
Code structure & duplication	The .NET project has good separation of concerns (Domain/Application/Infrastructure/Web layers) and Flutter follows feature-based organization. However, there is notable duplication in password handling across Flutter screens, some large service files, and hard-coded configuration values scattered in multiple places. The admin panel views reuse partials well, but the codebase would benefit from extracting common patterns.	72	Warning
Build reproducibility	The project is well-structured with consistent .NET 10 and Flutter SDK references, proper lock files, and a comprehensive setup guide. Docker Compose and database scripts are present and coherent. Minor issues: the setup document doesn't specify the exact Flutter version needed, and the iOS build prerequisites could be clearer about requiring a Mac with Xcode.	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 **Dependencies mostly current; Flutter packages slightly outdated, no critical vulnerabilities**

All dependencies are pinned in lockfiles. .NET packages are current (.NET 10, latest stable versions). Flutter dependencies are 6-12 months old but no known critical vulnerabilities found. Some minor updates available (riverpod 2.6.1→latest, go_router 14.6.2→latest). No wildcard versions or missing lockfiles.

Warning 2 **Weak default admin password and example secrets need attention before production**

No real credentials found. The demo admin password 'Admin@123' is weak and ships in seed.sql; buyers must change it immediately. Example files (.env.example, appsettings.example.json) contain placeholder secrets that are correctly excluded from deployment but should be clearer. All other secrets use environment variables or user input as intended.

Warning 3 **Strong authentication and role checks; minor ownership gaps on read endpoints**

All mutating and admin endpoints require authentication with JWT (API) or cookie (admin panel). Role checks are enforced via policies. Passwords use [hidden] with 100k iterations. Minor gaps: member read endpoints (orders, tickets, wallet) lack explicit ownership checks in some service methods, relying on controller-level filtering. No critical exposure foun

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

The .NET project has good separation of concerns (Domain/Application/Infrastructure/Web layers) and Flutter follows feature-based organization. However, there is notable duplication in password handling across Flutter screens, some large service files, and hard-coded configuration values scattered in multiple places. The admin panel views reuse partials well, but the codebase would benefit from extracting common patterns.

Requirement list

What the builder agreed to deliver before uploading — 17 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)	Met
✓	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 23 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 23 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 23 Oct 2026 with 4 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.