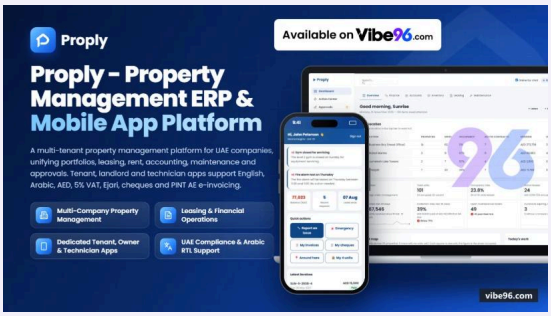


VIBE96 QUALITY REPORT

Proply Property ERP

Multi-company property management ERP on .NET 10 and SQL Server, with a React Native app.

Product ID	**B0004-24086**
Builder	**CodeHarbor**
Version reviewed	**1.0.0 · released 20 Nov 2026**
Stack	**ASP.NET Core 10 MVC + Razor, with a React Native 0.88 / Expo SDK 58 mobile app · C# / .NET 10 (ASP.NET Core 10 web app) · Microsoft SQL Server 2022 (2019 or newer works)**
Built with	**Claude Code**
Review period	**18 Nov 2026 – 20 Nov 2026 · 4 rounds of changes**

72 / 100

Listed after a manual review

Pass mark 75 and no failing check.

Scores

Security	71
Code structure	78
Docs & completeness	72
Dependency health	72
Build & deploy	72

✓ **AI code audit**
6 checks · 17 Nov 2026

✓ **Installed & run**
19 Nov 2026

✓ **Demo vs delivered**
19 Nov 2026

✓ **Final sign-off**
20 Nov 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.
- ! 18 of 25 promised items delivered (setup guide, example config and the rest).
- ! Security scored 71/100 with no failing check.
- ! 6 open warnings — explained in part 1.

Check this report is genuine. Scan the code or open vibe96.com/report/QR-26-00147. 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	None of the regex candidates is a real credential. Compose and the connection strings use required environment variables or placeholders, and appsettings.json has empty keys for encryption and signing. The remaining issues are a documented default admin password in a code comment and a default WhatsApp verify token that ships in appsettings.json.	88	Warning
Dependency vulnerabilities	The .NET project ships packages.lock.json, which is good, but its contents were not in the digest so installed versions could not be checked against advisories. The React Native app has package.json with caret ranges and no lockfile visible in the tree or digest, and the pinned versions (Expo 58, RN 0.88, TypeScript 6) could not be verified against registries.	72	Warning
Authorisation & access control	Passwords use a salted hasher, the mobile API is [Authorize] by default with role pinning, and login is throttled. However, the admin AccountController documents a GET seed route with default credentials, and most of the 57 portal controllers and the ownership checks on id lookups were not visible, so they could not be verified.	62	Warning
Database access rules	The visible architecture uses ADO.NET with stored procedures or parameterised text, per-company IDs passed to repositories, and a full installer schema with upgrades. The digest omitted all repository code and the schema contents, so injection, unscoped deletes, FK indexes and plain-text columns could not be verified. A few concrete issues surfaced: a documented default admin seed endpoint, a hard-coded verify token, and a schema split that may not create cleanly from empty.	62	Warning
Code structure & duplication	The project separates Controllers, Repositories, Infrastructure and Areas along clear lines, which is a good base. However, the digest shows very large single files (ImportEngine.cs at 98 KB, a 965 KB generated schema) and several hard-coded defaults that belong in configuration.	78	Warning
Build reproducibility	The .NET 10 project, installer, lockfile and docker-compose look coherent, and the Deploy README gives a clear order (database, then app settings, then first login). However, CLAUDE.md and the Deploy README point at paths that don't exist in the archive, HowToSetup.md could not be verified, and some prerequisites are not stated with versions.	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 **No real credentials found; a documented default admin password and a default webhook verify token ship**

None of the regex candidates is a real credential. Compose and the connection strings use required environment variables or placeholders, and appsettings.json has empty keys for encryption and signing. The remaining issues are a documented default admin password in a code comment and a default WhatsApp verify token that ships in appsettings.json.

Warning 2 **Large but organised monolith; some oversized files and hard-coded config defaults**

The project separates Controllers, Repositories, Infrastructure and Areas along clear lines, which is a good base. However, the digest shows very large single files (ImportEngine.cs at 98 KB, a 965 KB generated schema) and several hard-coded defaults that belong in configuration.

Warning 3 **Lockfile exists for .NET; React Native lockfile not visible; several unverifiable far-future version pins**

The .NET project ships packages.lock.json, which is good, but its contents were not in the digest so installed versions could not be checked against advisories. The React Native app has package.json with caret ranges and no lockfile visible in the tree or digest, and the pinned versions (Expo 58, RN 0.88, TypeScript 6) could not be verified against registries.

Warning 4 **Mostly reproducible, but doc drift, missing referenced files and unverifiable pieces create gaps**

The .NET 10 project, installer, lockfile and docker-compose look coherent, and the Deploy README gives a clear order (database, then app settings, then first login). However, CLAUDE.md and the Deploy README point at paths that don't exist in the archive, HowToSetup.md could not be verified, and some prerequisites are not stated with versions.

Warning 5 **Digest too thin to confirm SQL safety; schema and tenant scoping look sound, but several concerns remain**

The visible architecture uses ADO.NET with stored procedures or parameterised text, per-company IDs passed to repositories, and a full installer schema with upgrades. The digest omitted all repository code and the schema contents, so injection, unscoped deletes, FK indexes and plain-text columns could not be verified. A few concrete issues surfaced: a documented default admin seed endpoint, a hard-coded verify token, and a schema split that may not create cleanly from empty.

Warning 6 **Auth model looks solid, but a public seed-admin endpoint and unverified controllers cap the score**

Passwords use a salted hasher, the mobile API is [Authorize] by default with role pinning, and login is throttled. However, the admin AccountController documents a GET seed route with default credentials, and most of the 57 portal controllers and the ownership checks on id lookups were not visible, so they could not be verified.

Requirement list

What the builder agreed to deliver before uploading — 18 of 25 met.

!	Complete source for every part	Partial
✓	No real credentials	Met
!	Example configuration file	Partial
✓	Database creates from empty	Met
!	HowToSetup.md with two targets	Partial
!	Role-based authorisation on every write	Partial
✓	Pinned dependencies with lockfiles	Met
!	Reachable demo with sample data	Partial
✓	Documentation set (nice)	Met
✓	Seed data script (nice)	Met
!	Automated tests (nice)	Missing
✓	Docker setup for the API (nice)	Met
!	Differentiating features in code	Partial
✓	README.md	Met
✓	HowToSetup.md	Met
✓	NET a project file	Met

✓	a project file	Met
✓	NET a project file	Met
✓	React Native a project file	Met
✓	CLAUDE.md (nice)	Met
✓	CHANGELOG.md (nice)	Met
✓	a project file (nice)	Met
✓	docker-compose.yml (nice)	Met
✓	LICENSE-NOTE.md	Met
✓	a project file (optional)	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 19 Nov 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 19 Nov 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

Round 1: changes requested on 19 Nov 2026; the builder fixed them and uploaded again.

Round 2: changes requested on 19 Nov 2026; the builder fixed them and uploaded again.

Sign-off: approved for sale on 20 Nov 2026 with 6 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.