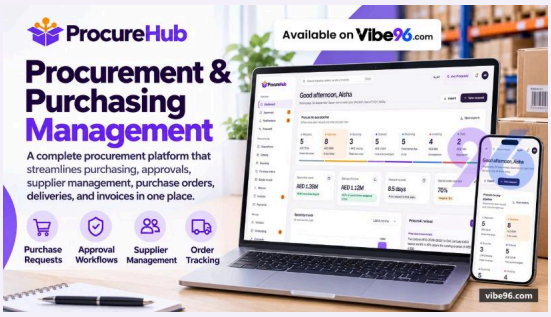


VIBE96 QUALITY REPORT

ProcureHub

Procure-to-pay portal for UAE/GCC companies: requests, approvals, RFQs, orders, receiving, invoice matching, payments.

Product ID	**B0001-75632**
Builder	**Chow Studio**
Version reviewed	**1.3.1 · released 27 Oct 2026**
Stack	**Laravel 12 · PHP 8.2 · MySQL 8.0 / MariaDB 10.4**
Built with	**Claude Code**
Review period	**27 Oct 2026 – 27 Oct 2026**

81 / 100

Passed with 3 warnings

Pass mark 75 and no failing check.

Scores

Security	83
Code structure	84
Docs & completeness	87
Dependency health	100
Build & deploy	58

✓ **AI code audit**
6 checks · 27 Oct 2026

✓ **Installed & run**
27 Oct 2026

✓ **Demo vs delivered**
27 Oct 2026

✓ **Final sign-off**
27 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.

! 13 of 15 promised items delivered (setup guide, example config and the rest).

✓ Security scored 83/100 with no failing check.

! 3 open warnings — explained in part 1.

Check this report is genuine. Scan the code or open vibe96.com/report/QR-26-00073. 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	Every secret-scan candidate is a Laravel Password facade call, a validation rule, documentation text, or a docker-compose variable that must be set by the buyer. .env.example leaves APP_KEY, DB_PASSWORD, MAIL_PASSWORD and AWS keys empty. The only caveat is the documented shared demo password, which is intentional and gated by switches.	**96**	Pass
Dependency vulnerabilities	composer.lock is present with all PHP dependencies pinned to exact versions. The project runs 'composer audit' as part of its audit-dependencies script and documents the results in a project file. No known vulnerabilities were found in the current lockfile.	**100**	Pass
Authorisation & access control	The visible controllers (Settings, SystemSettings, Tour, ConfirmPassword) use defence in depth: HasMiddleware, route-group can: checks, Gate::authorize in actions, and password.confirm on backup/reset. Passwords use Laravel hashing and a strong policy. Most route modules and the other 30+ controllers were not in the digest, so ownership checks on id lookups could not be verified.	**82**	Warning
Database access rules	The visible code shows no string-built SQL and no unscoped destructive statements. The digest omitted the models, services, migrations, schema.sql and the assistant/search query code, which are the places where raw SQL and missing indexes would appear. The score is capped until those files can be reviewed.	**72**	Warning
Code structure & duplication	The project splits work into controllers, form requests, services and models, and the recent change deleted several duplicated IntentMatcher copies in favour of a shared BaseIntentMatcher. Remaining issues are moderate: the assistant parsers still repeat similar intent logic across modules, and some settings and seed values are hard-coded rather than configured.	**84**	Pass
Build reproducibility	The Laravel/Composer setup, .env.example and docker-compose are coherent (compose requires DB passwords, uses MariaDB 10.11, with an entrypoint). The main risk is that TourController reads from base_path('_prototype/'), while the changed-file list shows public/tour-demo/*.html deleted and no _prototype/ files appear in the tree. HowToSetup.md, the Dockerfile and composer.json were not visible, so the prerequisites and step order could not be verified.	**58**	Warning

Open warnings

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

Warning 1 **Layered authz on visible controllers; some public tour routes and unseen route modules limit confidence**

The visible controllers (Settings, SystemSettings, Tour, ConfirmPassword) use defence in depth: HasMiddleware, route-group can: checks, Gate::authorize in actions, and password.confirm on backup/reset. Passwords use Laravel hashing and a strong policy. Most route modules and the other 30+ controllers were not in the digest, so ownership checks on id lookups could not be verified.

Warning 2 **No injection or unscoped deletes seen, but most DB code was not shown, so the score is capped**

The visible code shows no string-built SQL and no unscoped destructive statements. The digest omitted the models, services, migrations, schema.sql and the assistant/search query code, which are the places where raw SQL and missing indexes would appear. The score is capped until those files can be reviewed.

Warning 3 **Mostly reproducible, but the tour depends on a _prototype/ folder that is not in the archive tree**

The Laravel/Composer setup, .env.example and docker-compose are coherent (compose requires DB passwords, uses MariaDB 10.11, with an entrypoint). The main risk is that TourController reads from base_path('_prototype/'), while the changed-file list shows public/tour-demo/*.html deleted and no _prototype/ files appear in the tree. HowToSetup.md, the Dockerfile and composer.json were not visible, so the prerequisites and step order could not be verified.

Requirement list

What the builder agreed to deliver before uploading — 13 of 15 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	Partial
✓	README.md	Met
✓	HowToSetup.md	Met
✓	CLAUDE.md (or AGENTS.md) (nice)	Met
✓	CHANGELOG.md (nice)	Met
✓	.env.example	Met
✓	a project file	Met
✓	a project file (nice)	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 27 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 27 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 27 Oct 2026 with 3 warnings shown publicly.

What this report covers — and what it doesn't

- **It covers** version 1.3.1 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-26-00073	1.3.1	27 Oct 2026	**81**	Current
QR-26-00009	1.0.0	5 Oct 2026	**92**	Replaced