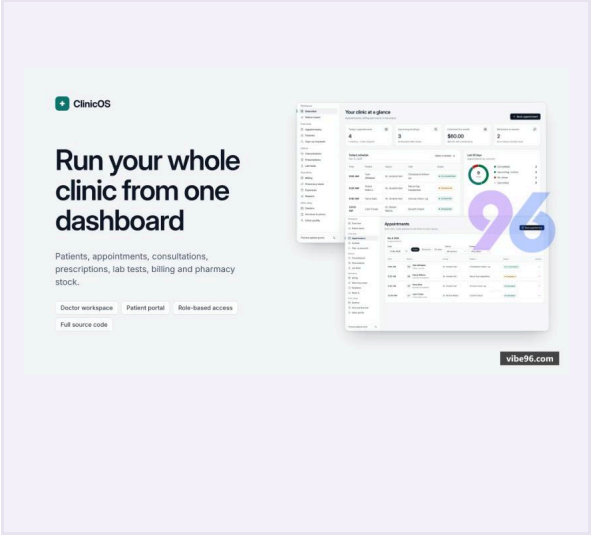


VIBE96 QUALITY REPORT

ClinicCare – Clinic Management System

Web app for single-site private clinics: patients, appointments, billing and pharmacy.

Product ID	**B0539-82689**
Builder	**Kamrul**
Version reviewed	**1.0.0 · released 11 Nov 2026**
Stack	**React 18.3 + Vite 6.4, with optional Supabase backend · TypeScript 5.9 (React 18.3 frontend) · PostgreSQL 16 (via Supabase; tested on PostgreSQL 16)**
Built with	**Claude Code**
Review period	**8 Nov 2026 – 11 Nov 2026**

83 / 100

Passed with 4 warnings

Pass mark 75 and no failing check.

Scores

Security	81
Code structure	88
Docs & completeness	60
Dependency health	88
Build & deploy	82

✓ **AI code audit**
6 checks · 8 Nov 2026

✓ **Installed & run**
11 Nov 2026

✓ **Demo vs delivered**
11 Nov 2026

✓ **Final sign-off**
11 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.
- ! 15 of 25 promised items delivered (setup guide, example config and the rest).
- ✓ Security scored 81/100 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-26-00116. 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	The regex candidates are mostly false positives: function parameters named password in App.tsx, authCore.ts and authService.ts. The only real item is a hard-coded demo password in demoData.ts, intentionally public and shown on the sign-in screen. The project guards against it being used with a real backend: auto-login refuses to build with Supabase configured, and demo logins are disabled when Supabase is configured. The .env.example contains only placeholders, and the code uses only the Supabase publishable key.	90	Warning
Dependency vulnerabilities	a project file pins all four runtime and six dev dependencies to exact versions, and package-lock.json is present. No wildcard, 'latest' or range versions were found, and no known critical advisory could be matched from the digest. The lockfile contents were not shown, so transitive dependencies could not be checked, and several pinned versions look unusual and need confirming with npm audit.	88	Pass
Authorisation & access control	The app has no custom server routes; all access goes through Supabase Auth and PostgreSQL row-level security, which the docs say is backed by 126 allow/deny tests covering ownership and role isolation. The policies themselves (schema.sql) were not in the excerpts, so enforcement could not be verified directly. Client-side role checks are explicitly UX-only, and a public demo password path exists that must stay disabled in production.	82	Warning
Database access rules	The visible code uses the Supabase client with parameterised calls (.eq, .rpc) and no string-built SQL. The docs describe RLS, a 126-check allow/deny suite and idempotent schema scripts. However, schema.sql, cloudStorage.ts and the test SQL were not in the excerpts, so foreign keys, indexes on foreign keys, cascade rules, unscoped deletes in the diff-based sync and plain-text sensitive columns could not be verified.	70	Warning
Code structure & duplication	The project separates pages, lib, features, components and styles, keeps SECTION_ROLES in one place, and keeps pure business rules in clinic.ts. The main costs are a few places where the auth flow and configuration values are repeated or hard-coded, which are easy to consolidate.	88	Pass
Build reproducibility	The project has a lockfile, pinned dependencies, pinned Node/npm engines, a coherent multi-stage Dockerfile and compose file whose env args match .env.example. Gaps: the authors state `npm ci`/`vite build` were never verified against the registry, HowToSetup.md was not available to check ordering, and some documentation and script leftovers are inconsistent.	82	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; one public demo password ships in source (weak default)**

The regex candidates are mostly false positives: function parameters named password in App.tsx, authCore.ts and authService.ts. The only real item is a hard-coded demo password in demoData.ts, intentionally public and shown on the sign-in screen. The project guards against it being used with a real backend: auto-login refuses to build with Supabase configured, and demo logins are disabled when Supabase is configured. The .env.example contains only placeholders, and the code uses only the Supabase publishable key.

Warning 2 **RLS-based authz is well designed, but the schema and policies were not visible and demo logins are public**

The app has no custom server routes; all access goes through Supabase Auth and PostgreSQL row-level security, which the docs say is backed by 126 allow/deny tests covering ownership and role isolation. The policies themselves (schema.sql) were not in the excerpts, so enforcement could not be verified directly. Client-side role checks are explicitly UX-only, and a public demo password path exists that must stay disabled in production.

Warning 3 **No injection paths visible, but schema and cloudStorage were not provided, so DB rules cannot be fully verified**

The visible code uses the Supabase client with parameterised calls (.eq, .rpc) and no string-built SQL. The docs describe RLS, a 126-check allow/deny suite and idempotent schema scripts. However, schema.sql, cloudStorage.ts and the test SQL were not in the excerpts, so foreign keys, indexes on foreign keys, cascade rules, unscoped deletes in the diff-based sync and plain-text sensitive columns could not be verified.

Warning 4 **Build mostly reproducible; minor doc and config inconsistencies, HowToSetup.md not verified**

The project has a lockfile, pinned dependencies, pinned Node/npm engines, a coherent multi-stage Dockerfile and compose file whose env args match .env.example. Gaps: the authors state `npm ci`/`vite build` were never verified against the registry, HowToSetup.md was not available to check ordering, and some documentation and script leftovers are inconsistent.

Requirement list

What the builder agreed to deliver before uploading — 15 of 25 met.

!	Complete web application source	Partial
!	Database schema creates from empty	Partial
!	Row-level security and role checks	Partial
✓	No real credentials	Met
!	Example environment file	Partial
!	Pinned dependencies and lockfile	Partial
✓	HowToSetup.md with two targets	Met
✓	Reachable demo with sample data	Met
!	CLAUDE.md architecture guide (nice)	Partial
✓	Seed data (nice)	Met
!	Docker setup (nice)	Partial
!	Keep files small and layered (nice)	Partial
!	API and ERD documentation (optional)	Missing
✓	README.md	Met
✓	HowToSetup.md	Met
✓	ClinicCare Web a project filee	Met
✓	ClinicCare Web a project file	Met
✓	ClinicCare Web App/.nvmrc	Met
✓	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 (optional)	Met
!	a project file (optional)	Missing

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 11 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 11 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

None — approved on the first round.

Sign-off: approved for sale on 11 Nov 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.