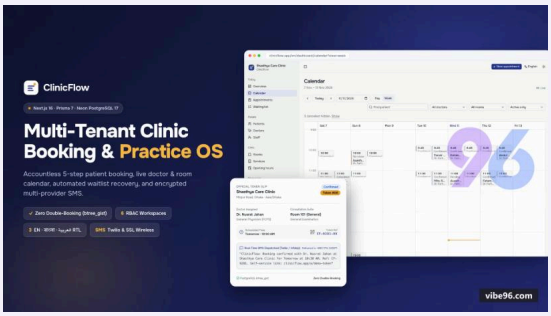


VIBE96 QUALITY REPORT

ClinicFlow — Multi-Tenant Clinic Management & Appointment System

Multi-tenant clinic booking management system clinics and hospitals, with accountless patient booking and SMS reminders.

Product ID	**B0097-37124**
Builder	**MD FAHIM**
Version reviewed	**1.0.0 · released 16 Nov 2026**
Stack	**Next.js 16 (App Router) with React 19 · TypeScript 5.9 on Node.js 22+ · PostgreSQL 17 (Neon) with Prisma 7; btree_gist and pg_trgm extensions**
Built with	**Claude Code, Google Antigravity**
Review period	**11 Nov 2026 – 16 Nov 2026**

68 / 100

Listed after a manual review

Pass mark 75 and no failing check.

Scores

Security	69
Code structure	78
Docs & completeness	83
Dependency health	62
Build & deploy	62

✓ **AI code audit**
6 checks · 10 Nov 2026

✓ **Installed & run**
16 Nov 2026

✓ **Demo vs delivered**
16 Nov 2026

✓ **Final sign-off**
16 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.
- ! 20 of 24 promised items delivered (setup guide, example config and the rest).
- ! Security scored 69/100 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-26-00140. 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 live API keys, private keys, or real database URLs were found. The scan hits are placeholders, docker-compose local dev values, and a publicly documented demo password hardcoded in the seed and landing page. These are weak defaults that could reach production if the buyer does not override them.	**85**	Warning
Dependency vulnerabilities	The only manifest in the digest is Next.js Web a project file. Most production dependencies use caret ranges, and the file tree shows no lockfile (lockfiles may be omitted from the digest, so this is unconfirmed). Versions can't be matched against advisories with confidence, and some oddities (a `cn` package, `shadcn` as a runtime dependency) need cleanup.	**62**	Warning
Authorisation & access control	The digest only exposes docs, config and generated Prisma code, not the server actions (src/actions), the /cron/run route, /media/[id] or the dashboard/platform pages, so guards and ownership checks could not be traced. Documentation claims guarded, tenant-scoped actions, Argon2 hashing and HMAC-signed patient tokens, and a tenant-isolation test exists, which is encouraging but unverified. Some surfaces are inherently high-risk and need confirmation before production use.	**60**	Warning
Database access rules	The digest omits all server code, Prisma query sites and the schema contents, so injection and unscoped-delete checks could not be verified directly. The project uses Prisma 7 with migrations, a seed and an SQL schema, and documents tenant-isolation tests and AES-GCM credential encryption, which is a good sign. Scoring is held down because the key files were unseen, the schema is duplicated in three places, and the seed path ships a known default password.	**62**	Warning
Code structure & duplication	The App Router layout separates actions, server services, and UI components sensibly, which is a good base. However, the submission duplicates the whole project (root ClinicFlow-Submission vs. 'Next.js Web App'), carries roughly 50+ Claude skill reference files and internal tooling residue, and repeats the demo password in several places, which makes it harder for a buyer to know which copy is authoritative.	**78**	Warning
Build reproducibility	The Next.js app has a package.json, a Prisma migration and an .env.example, and the live demo shows it builds and runs on Vercel. However, the archive duplicates docs/DB/README at two levels, commits generated Prisma output and tool folders, and the digest cannot confirm that setup steps, versions and prerequisites are correct or that a lockfile exists. A clean-machine build is plausible but not proven.	**62**	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; shared demo password and dev-default secrets ship to buyers**

No live API keys, private keys, or real database URLs were found. The scan hits are placeholders, docker-compose local dev values, and a publicly documented demo password hardcoded in the seed and landing page. These are weak defaults that could reach production if the buyer does not override them.

Warning 2 **Structure is organized by feature, but the project ships duplicated trees, a large vendored skills folder, and generated/tokenade artifacts**

The App Router layout separates actions, server services, and UI components sensibly, which is a good base. However, the submission duplicates the whole project (root ClinicFlow-Submission vs. 'Next.js Web App'), carries roughly 50+ Claude skill reference files and internal tooling residue, and repeats the demo password in several places, which makes it harder for a buyer to know which copy is authoritative.

Warning 3 **Database layer not verifiable from digest; schema and duplication concerns, no confirmed injection**

The digest omits all server code, Prisma query sites and the schema contents, so injection and unscoped-delete checks could not be verified directly. The project uses Prisma 7 with migrations, a seed and an SQL schema, and documents tenant-isolation tests and AES-GCM credential encryption, which is a good sign. Scoring is held down because the key files were unseen, the schema is duplicated in three places, and the seed path ships a known default password.

Warning 4 **Duplicated app tree, committed generated code and unverified setup doc leave build reproducibility unclear**

The Next.js app has a package.json, a Prisma migration and an .env.example, and the live demo shows it builds and runs on Vercel. However, the archive duplicates docs/DB/README at two levels, commits generated Prisma output and tool folders, and the digest cannot confirm that setup steps, versions and prerequisites are correct or that a lockfile exists. A clean-machine build is plausible but not proven.

Warning 5 **Caret ranges everywhere and no lockfile visible; dependency set can't be verified as free of known CVEs**

The only manifest in the digest is Next.js Web a project file. Most production dependencies use caret ranges, and the file tree shows no lockfile (lockfiles may be omitted from the digest, so this is unconfirmed). Versions can't be matched against advisories with confidence, and some oddities (a `cn` package, `shadcn` as a runtime dependency) need cleanup.

Warning 6 **Authz not verifiable: server actions and route handlers were not shown; design claims guards but evidence is missing**

The digest only exposes docs, config and generated Prisma code, not the server actions (src/actions), the /cron/run route, /media/[id] or the dashboard/platform pages, so guards and ownership checks could not be traced. Documentation claims guarded, tenant-scoped actions, Argon2 hashing and HMAC-signed patient tokens, and a tenant-isolation test exists, which is encouraging but unverified. Some surfaces are inherently high-risk and need confirmation before production use.

Requirement list

What the builder agreed to deliver before uploading — 20 of 24 met.

✓	Complete source of the web app	Met
!	No real credentials: config files may be present but must contain placeholders or local defaults only	Partial
!	Lockfile and pinned dependencies	Partial
✓	Schema creates from empty	Met
!	Tenant isolation and role-gated access	Partial
✓	HowToSetup.md with at least two targets	Met
✓	Working demo with sample data	Met
!	Developer-friendly structure (nice)	Partial
✓	CLAUDE.md with architecture and test commands (nice)	Met

✓	Seed data and docker-compose (nice)	Met
✓	README and licence note (nice)	Met
✓	API and deployment docs (optional)	Met
✓	README.md	Met
✓	HowToSetup.md	Met
✓	Next.js Web a project filee	Met
✓	Next.js Web a project file	Met
✓	a project file	Met
✓	Next.js Web 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 (nice)	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 16 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 16 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 16 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.