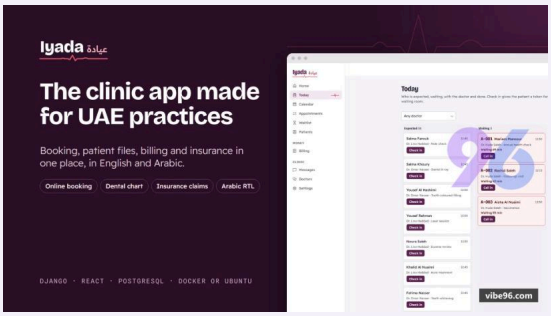


VIBE96 QUALITY REPORT

Iyada: Clinic Booking and Patient Management

Self-hosted English/Arabic clinic app for 1-10 doctor practices: booking, patient files, billing, insurance, queue.

Product ID	**B0612-89035**
Builder	**Bablu Mia**
Version reviewed	**1.0.0 · released 18 Nov 2026**
Stack	**Django 6.1 + Django REST Framework 3.18, React 19 + Vite 8 + Tailwind 4 · Python 3.12 + TypeScript 6 (React 19 web app) · PostgreSQL 16 (with Redis 7 for cache and background jobs)**
Built with	**Claude Code**
Review period	**5 Nov 2026 – 18 Nov 2026 · 1 round of changes**

83 / 100

Passed with 3 warnings

Pass mark 75 and no failing check.

Scores

Security	83
Code structure	88
Docs & completeness	73
Dependency health	82
Build & deploy	80

✓ **AI code audit**
6 checks · 5 Nov 2026

✓ **Installed & run**
18 Nov 2026

✓ **Demo vs delivered**
18 Nov 2026

✓ **Final sign-off**
18 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.
- ! 11 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-26-00146. 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 regex candidate is a form field name, an environment variable reference, or a randomly generated test value. The .env.example leaves all secrets empty and relies on an init_env command to generate them, and CI generates random values at runtime. A small residual risk remains because the full archive was not visible, notably a project file (4 MB) and the omitted files.	92	Pass
Dependency vulnerabilities	The backend pins every Python package to an exact version, and the web app has a package-lock.json that fixes its npm ranges. No known critical vulnerability could be confirmed from the visible versions. The remaining items are weaker pinning in package.json, no visible backend hash lockfile, and an unverified redis image tag.	82	Pass
Authorisation & access control	The digest shows a deliberate access design: ScopedViewSet with permission codes per action, an access-matrix test, JWT with MFA, hashed-password fields and scoped API keys. Only a handful of URL files were visible, so most endpoints could not be traced, and several public/token endpoints (demo-login, display token, patient token) need ownership checks confirmed.	78	Warning
Database access rules	The visible material shows a disciplined Django ORM backend. Docs state soft-delete, db_index on every FK, and encrypted ID numbers, and a test enforces permission scoping. The digest omitted all models, migrations, views and raw SQL, so none of these claims could be checked against code, and a 4 MB seed.sql plus a 347 KB schema.sql could not be checked either.	80	Warning
Code structure & duplication	The backend is split into clear apps (authentication, clinic, patients, clinical, billing, and so on) and the web app into feature folders with file-size and complexity limits enforced by lint. The digest shows no god files or copy-pasted modules, though it is too partial to confirm the limits are met everywhere.	88	Pass
Build reproducibility	Pinned dependencies, lockfile, Dockerfiles and a coherent compose file make a clean build likely. HowToSetup.md and the backend Dockerfile were not in the excerpt, so setup order and some references could not be confirmed, and a few small inconsistencies were found.	80	Warning

Open warnings

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

Warning 1 **Backend DB code was largely not visible; the conventions are strong, but the schema and queries could not be verified**

The visible material shows a disciplined Django ORM backend. Docs state soft-delete, db_index on every FK, and encrypted ID numbers, and a test enforces permission scoping. The digest omitted all models, migrations, views and raw SQL, so none of these claims could be checked against code, and a 4 MB seed.sql plus a 347 KB schema.sql could not be checked either.

Warning 2 **Largely reproducible build; setup doc and a few inconsistencies could not be fully verified**

Pinned dependencies, lockfile, Dockerfiles and a coherent compose file make a clean build likely. HowToSetup.md and the backend Dockerfile were not in the excerpt, so setup order and some references could not be confirmed, and a few small inconsistencies were found.

Warning 3 **Authz design looks solid but most views were not visible; a few unverifiable public and token endpoints**

The digest shows a deliberate access design: ScopedViewSet with permission codes per action, an access-matrix test, JWT with MFA, hashed-password fields and scoped API keys. Only a handful of URL files were visible, so most endpoints could not be traced, and several public/token endpoints (demo-login, display token, patient token) need ownership checks confirmed.

Requirement list

What the builder agreed to deliver before uploading — 11 of 15 met.

✓	Complete source for every component	Met
✓	No real credentials in the archive	Met
!	Builds from a clean machine	Partial
✓	Database script or migrations	Met
!	Authentication and roles on every mutating endpoint	Partial
!	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 18 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 18 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 13 Nov 2026; the builder fixed them and uploaded again.

Sign-off: approved for sale on 18 Nov 2026 with 3 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.