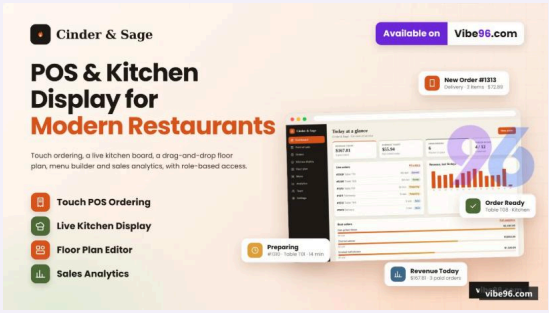


VIBE96 QUALITY REPORT

Resturant management system

Restaurant POS, kitchen display and floor-plan system on Next.js and Supabase, for independent restaurants and cafés.

Product ID	**B0539-42968**
Builder	**Kamrul**
Version reviewed	**1.0.0 · released 12 Nov 2026**
Stack	**Next.js 15.5 (App Router) with React 19.1 · TypeScript 5.9 (strict) on Node.js 20.9+ · PostgreSQL via Supabase (RLS, SECURITY DEFINER functions, Realtime); minimum version not pinned in digest**
Built with	**Claude Code**
Review period	**9 Nov 2026 – 12 Nov 2026 · 1 round of changes**

78 / 100

Passed with 5 warnings

Pass mark 75 and no failing check.

Scores

Security	77
Code structure	78
Docs & completeness	80
Dependency health	82
Build & deploy	80

✓ **AI code audit**
6 checks · 9 Nov 2026

✓ **Installed & run**
12 Nov 2026

✓ **Demo vs delivered**
12 Nov 2026

✓ **Final sign-off**
12 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.
- ! 12 of 15 promised items delivered (setup guide, example config and the rest).
- ✓ Security scored 77/100 with no failing check.
- ! 5 open warnings — explained in part 1.

Check this report is genuine. Scan the code or open vibe96.com/report/QR-26-00132. 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	All regex hits for Stripe and Supabase keys are placeholders (whsec_REPLACE_ME, sk_test_REPLACE_ME, YOUR_*) or environment variable reads, so no live credential is committed. The only concern is a hardcoded demo password for fictional demo accounts that ships to buyers and is published in the README.	90	Warning
Dependency vulnerabilities	Node-Project has package.json and a package-lock.json (193 KB), and every dependency is pinned to an exact version with no wildcards or pre-release tags. The only loose spot is the postcss override range, and some pinned versions could not be checked against advisories. No known critical vulnerability could be confirmed from the digest.	82	Pass
Authorisation & access control	The architecture puts authorization in Postgres RLS and SECURITY DEFINER functions, and the Stripe checkout route is documented as auth + role guarded via authorizeRestaurant(). However, the checkout and webhook route bodies and schema.sql were not provided for review, so ownership checks on order/payment ids and the webhook's signature handling cannot be confirmed. The demo mode and a publicly documented demo password are a deployment risk.	72	Warning
Database access rules	The digest shows no string-built SQL and no unscoped deletes. Data access goes through Supabase's query builder, RLS and SECURITY DEFINER functions, and the one visible server query is scoped by restaurant, user and active flag. a project file, a project file and the RLS tests were not in the excerpts, so foreign keys, indexes, cascade rules and function safety could not be confirmed.	70	Warning
Code structure & duplication	The project follows a sensible folder map (routes, views, lib/data stores, lib/server). The main structural risks are the DataStore pattern being implemented twice with likely duplicated logic, and hard-coded protected route and config lists that must be kept in sync by hand. Only the digest excerpts were available, so file-length and dead-code claims are partly unverified.	78	Warning
Build reproducibility	Manifests, Dockerfile and compose are coherent: versions are pinned, a lockfile exists, `npm ci` is used, and the standalone output is copied correctly. The main gaps are that HowToSetup.md was not available to verify prerequisites and step order, and that the README image paths, `next.config.ts` and the `.dockerignore` contents are unverified or mismatched.	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 **No real credentials found; one shipped demo password and a documented demo login are weak defaults**

All regex hits for Stripe and Supabase keys are placeholders (whsec_REPLACE_ME, sk_test_REPLACE_ME, YOUR_*) or environment variable reads, so no live credential is committed. The only concern is a hardcoded demo password for fictional demo accounts that ships to buyers and is published in the README.

Warning 2 **Layered structure is clear, but several files and checks are hard-coded or mix concerns**

The project follows a sensible folder map (routes, views, lib/data stores, lib/server). The main structural risks are the DataStore pattern being implemented twice with likely duplicated logic, and hard-coded protected route and config lists that must be kept in sync by hand. Only the digest excerpts were available, so file-length and dead-code claims are partly unverified.

Warning 3 **Mostly reproducible Next.js build; setup doc unseen and a few coherence gaps remain**

Manifests, Dockerfile and compose are coherent: versions are pinned, a lockfile exists, `npm ci` is used, and the standalone output is copied correctly. The main gaps are that HowToSetup.md was not available to verify prerequisites and step order, and that the README image paths, `next.config.ts` and the `.dockerignore` contents are unverified or mismatched.

Warning 4 **Database layer looks well designed, but schema, supabase-store and RLS tests were not provided, so it can't be fully verified**

The digest shows no string-built SQL and no unscoped deletes. Data access goes through Supabase's query builder, RLS and SECURITY DEFINER functions, and the one visible server query is scoped by restaurant, user and active flag. a project file, a project file and the RLS tests were not in the excerpts, so foreign keys, indexes, cascade rules and function safety could not be confirmed.

Warning 5 **Authz is well-designed (RLS + role checks) but key routes and the SQL boundary could not be verified**

The architecture puts authorization in Postgres RLS and SECURITY DEFINER functions, and the Stripe checkout route is documented as auth + role guarded via authorizeRestaurant(). However, the checkout and webhook route bodies and schema.sql were not provided for review, so ownership checks on order/payment ids and the webhook's signature handling cannot be confirmed. The demo mode and a publicly documented demo password are a deployment risk.

Requirement list

What the builder agreed to deliver before uploading — 12 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)	Met
✓	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 12 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 12 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 12 Nov 2026 with 5 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.