

VIBE96 QUALITY REPORT

Mixoff — Office Productivity Suite

Local-first Windows productivity suite with Documents, Sheets, Slides, Mixoff Flow, and optional AI integration.

Product ID	**B0814-15415**
Builder	**Zyphase**
Version reviewed	**1.0.0 · released 6 Nov 2026**
Stack	**React 18.3.1 with Tauri 2 (@tauri-apps/api 2.11.1) · TypeScript 5.9.3 (frontend) with Rust 1.99.1 (native layer) · None; optional managed-AI service uses SQLite via node: 24.22.1**
Built with	**OpenAI Codex**
Review period	**30 Oct 2026 – 6 Nov 2026 · 2 rounds of changes**

72 / 100

Listed after a manual review

Pass mark 75 and no failing check.

Scores

Security	74
Code structure	62
Docs & completeness	76
Dependency health	72
Build & deploy	72

✓ **AI code audit**
6 checks · 29 Oct 2026

✓ **Installed & run**
6 Nov 2026

✓ **Demo vs delivered**
6 Nov 2026

✓ **Final sign-off**
6 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.

! 16 of 21 promised items delivered (setup guide, example config and the rest).

! Security scored 74/100 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-26-00102. 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 was reviewed and none contains a real credential. The managed-ai .env.example uses clear REPLACE_WITH placeholders, and the other hits are secret-detection regexes, runtime handling of user-supplied credentials in test mocks, and tokenizer function names. Only minor hygiene notes remain.	96	Pass
Dependency vulnerabilities	Lockfiles exist for npm (package-lock.json) and Rust (Cargo.lock) and direct dependencies are exact-pinned, which is good. However, the xlsx dependency is installed from a SheetJS CDN tarball outside the npm registry, and the managed-ai service has no lockfile or declared dependencies. Several pinned versions (e.g. vitest 5.0.5, vite 8.3.4, @tiptap 3.31.3) could not be verified against known advisories from the digest, so the lockfile contents were not audited line by line.	72	Warning
Authorisation & access control	The product is a local-first Tauri desktop app, so its main attack surface is a small managed-AI gateway whose routes (gateway.mjs, server.mjs) were not included in the excerpts. The README describes bearer device tokens for customer routes and a separate admin secret for /admin/*, but I could not verify the guards, ownership checks or constant-time comparison in code. The demo is a static mock-IPC frontend with no data-changing backend.	55	Warning
Database access rules	The desktop app stores data in native files, not a database, so most SQL checks do not apply. The one database is the managed-ai gateway's SQLite ledger (gateway.mjs, server.mjs). The digest did not include its source, so injection, unscoped UPDATE/DELETE, per-user scoping, foreign keys and indexes could not be verified. The README also admits the ledger is not production-ready for multi-host use.	70	Warning
Code structure & duplication	The digest shows a sensible top-level split (src/modules, src/flow, src/storage, src-tauri, crates/file-safety), but several files are written as dense one-line code that is hard to review. The scripts folder and test harnesses also appear to contain large amounts of generated or duplicated tooling that buyers would have to maintain.	62	Warning
Build reproducibility	The main app has pinned dependencies, lockfiles, and documented npm/Tauri build steps, so a Windows machine can likely restore and build it. However, the archive is nested with a duplicated top-level a project file, README links point to files outside the digest, and several prerequisites (Python, Edge, Rust version, Node for the gateway) are not clearly pinned. Setup in HowToSetup.md could not be verified from the digest.	72	Warning

Open warnings

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

Warning 1 Structure is modular, but minified-looking single-line files and a dense scripts/ folder hurt maintainability

The digest shows a sensible top-level split (src/modules, src/flow, src/storage, src-tauri, crates/file-safety), but several files are written as dense one-line code that is hard to review. The scripts folder and test harnesses also appear to contain large amounts of generated or duplicated tooling that buyers would have to maintain.

Warning 2 **Database rules not verifiable: managed-ai SQLite gateway source was not shown; no injection evidence found**

The desktop app stores data in native files, not a database, so most SQL checks do not apply. The one database is the managed-ai gateway's SQLite ledger (gateway.mjs, server.mjs). The digest did not include its source, so injection, unscoped UPDATE/DELETE, per-user scoping, foreign keys and indexes could not be verified. The README also admits the ledger is not production-ready for multi-host use.

Warning 3 **Managed AI gateway auth is documented but its code was not visible; the desktop app has no web routes**

The product is a local-first Tauri desktop app, so its main attack surface is a small managed-AI gateway whose routes (gateway.mjs, server.mjs) were not included in the excerpts. The README describes bearer device tokens for customer routes and a separate admin secret for /admin/*, but I could not verify the guards, ownership checks or constant-time comparison in code. The demo is a static mock-IPC frontend with no data-changing backend.

Warning 4 **Locked and mostly pinned deps, but xlsx comes from a CDN tarball, plus pre-release/unverifiable versions**

Lockfiles exist for npm (package-lock.json) and Rust (Cargo.lock) and direct dependencies are exact-pinned, which is good. However, the xlsx dependency is installed from a SheetJS CDN tarball outside the npm registry, and the managed-ai service has no lockfile or declared dependencies. Several pinned versions (e.g. vitest 5.0.5, vite 8.3.4, @tiptap 3.31.3) could not be verified against known advisories from the digest, so the lockfile contents were not audited line by line.

Warning 5 **Mostly reproducible with lockfiles, but archive layout, setup doc and tooling prerequisites have gaps**

The main app has pinned dependencies, lockfiles, and documented npm/Tauri build steps, so a Windows machine can likely restore and build it. However, the archive is nested with a duplicated top-level a project file, README links point to files outside the digest, and several prerequisites (Python, Edge, Rust version, Node for the gateway) are not clearly pinned. Setup in HowToSetup.md could not be verified from the digest.

Requirement list

What the builder agreed to deliver before uploading — 16 of 21 met.

✓	Complete source for the desktop app	Met
!	Lockfiles and pinned dependencies	Partial
✓	No real credentials	Met
!	HowToSetup.md for each desktop platform	Partial
!	README.md with the five differentiating features	Partial
!	Reachable demo	Partial
✓	Example settings file if the app reads keys (nice)	Met
✓	CLAUDE.md for AI coding tools (nice)	Met
!	CHANGELOG.md starting at 1.0.0 (nice)	Partial
✓	Tests for core engines (nice)	Met
✓	Architecture documentation (optional)	Met
✓	README.md	Met
✓	HowToSetup.md	Met

✓	Mixoff a project file	Met
✓	Mixoff a project file	Met
✓	Mixoff a project file	Met
✓	Mixoff a project file	Met
✓	Mixoff a project filee (nice)	Met
✓	CLAUDE.md (nice)	Met
✓	CHANGELOG.md (nice)	Met
✓	LICENSE-NOTE.md	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 6 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 6 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 6 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.