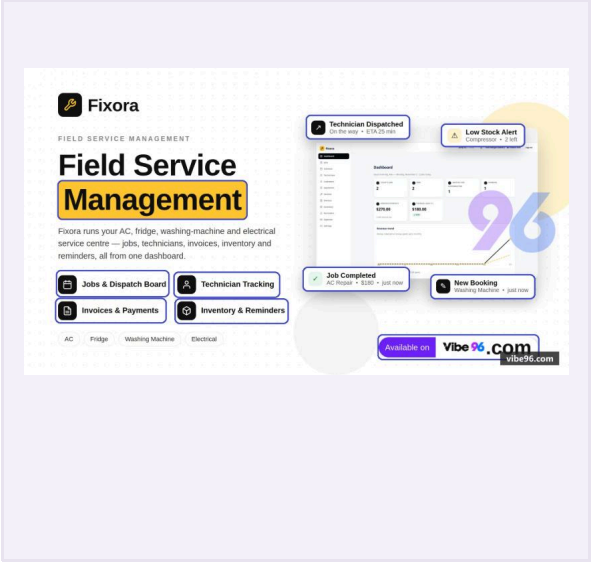


VIBE96 QUALITY REPORT

Fixora — Field Service Management SaaS

All-in-one field service management for AC, fridge, washer & electrical repair businesses.

Product ID	**B0308-51959**
Builder	**Jehan Ahmed**
Version reviewed	**1.0.0 · released 4 Nov 2026**
Stack	**None: zero-dependency Node.js server, vanilla JS frontend, no build step · JavaScript (Node.js 20+, no npm packages) · Supabase (PostgreSQL); no minimum version stated in the digest**
Built with	**OpenAI Codex**
Review period	**2 Nov 2026 – 4 Nov 2026**

73 / 100

Listed after a manual review

Pass mark 75 and no failing check.

Scores

Security	67
Code structure	72
Docs & completeness	60
Dependency health	96
Build & deploy	80

✓ **AI code audit**
6 checks · 2 Nov 2026

✓ **Installed & run**
4 Nov 2026

✓ **Demo vs delivered**
4 Nov 2026

✓ **Final sign-off**
4 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 67/100 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-26-00095. 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 code references to environment variables, user-supplied tokens, or test fixtures, not live credentials. .env.example is blank by design and .env is excluded from the archive and Docker image. The only concern is that the same demo passwords are repeated across the test suite, and a buyer who copies them into a public demo would be using guessable credentials.	90	Warning
Dependency vulnerabilities	package.json declares an empty dependencies object, so Fixora ships no third-party npm packages and has no known-CVE exposure from dependencies. package-lock.json is present, and the Dockerfile uses the maintained node:20-alpine image. Remaining concerns are minor: the base image tag is a floating minor and Node 20 is approaching end-of-life.	96	Pass
Authorisation & access control	The visible code requires a Bearer token on authenticated routes, with documented exceptions for login and booking only. The route table in server.js is truncated, so per-route role checks and ownership checks on id lookups could not be traced. Password storage is delegated to Supabase Auth, but the in-memory store handles passwords in a way that could not be verified.	55	Warning
Database access rules	The data layer (a project file) talks to Supabase over REST, which usually avoids raw SQL concatenation, but the excerpts show none of the query construction, delete/update scoping, or schema/migration files. Tenant scoping, foreign keys, indexes and cascade rules cannot be confirmed, and plaintext password handling in the store is a concern.	55	Warning
Code structure & duplication	The top-level split (lib/ for store, rules, whatsapp and demo data; public/ for the frontend; test/ and scripts/) is readable and keeps business rules out of the UI. The main costs are that server.js is described as the whole backend and mixes routing, auth, and invoice and reminder logic, and that app.js is a single 122 KB file. Schema, migrations and setup docs are also repeated across several files, and the demo credentials are copied into four test files.	72	Warning
Build reproducibility	The project has no npm dependencies, a pinned Node 20+ engine, and a README that runs database creation, then config, then start in the right order. A clean machine can very likely run it. The Dockerfile and compose healthcheck have small gaps, and HowToSetup.md was not visible in the excerpts, so its steps could not be checked.	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; demo passwords hardcoded in tests are weak defaults**

The regex candidates are code references to environment variables, user-supplied tokens, or test fixtures, not live credentials. .env.example is blank by design and .env is excluded from the archive and Docker image. The only concern is that the same demo passwords are repeated across the test suite, and a buyer who copies them into a public demo would be using guessable credentials.

Warning 2 **Zero-dependency Node app is mostly reproducible; Dockerfile and setup docs have small gaps**

The project has no npm dependencies, a pinned Node 20+ engine, and a README that runs database creation, then config, then start in the right order. A clean machine can very likely run it. The Dockerfile and compose healthcheck have small gaps, and HowToSetup.md was not visible in the excerpts, so its steps could not be checked.

Warning 3 **Database access cannot be verified: schema, migrations and query code were not in the excerpts**

The data layer (a project file) talks to Supabase over REST, which usually avoids raw SQL concatenation, but the excerpts show none of the query construction, delete/update scoping, or schema/migration files. Tenant scoping, foreign keys, indexes and cascade rules cannot be confirmed, and plaintext password handling in the store is a concern.

Warning 4 **Authz design looks sound, but most routes and password handling are not visible, so it cannot be confirmed**

The visible code requires a Bearer token on authenticated routes, with documented exceptions for login and booking only. The route table in server.js is truncated, so per-route role checks and ownership checks on id lookups could not be traced. Password storage is delegated to Supabase Auth, but the in-memory store handles passwords in a way that could not be verified.

Warning 5 **Layered layout is sensible, but server.js and app.js are oversized and the release duplicates migrations and docs**

The top-level split (lib/ for store, rules, whatsapp and demo data; public/ for the frontend; test/ and scripts/) is readable and keeps business rules out of the UI. The main costs are that server.js is described as the whole backend and mixes routing, auth, and invoice and reminder logic, and that app.js is a single 122 KB file. Schema, migrations and setup docs are also repeated across several files, and the demo credentials are copied into four test files.

Requirement list

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

!	Complete source of the web app and API	Partial
✓	No real credentials: config files may be present but must contain placeholders or local defaults only	Met
!	Exactly one example config file: .env.example	Partial
!	Database schema that creates from empty	Partial
!	HowToSetup.md with at least two install targets	Partial
✓	Pinned dependencies and lockfile	Met
!	Role-gated routes and per-user data isolation	Partial
!	Live demo with sample data	Partial
✓	README.md, CLAUDE.md and CHANGELOG.md at root (nice)	Met
✓	Seed data for demo (nice)	Met
!	Code structure without giant files (nice)	Partial
!	Parameterised database queries	Partial
!	Docs folder with deployment notes (optional)	Missing
✓	README.md	Met
✓	HowToSetup.md	Met
✓	CLAUDE.md (nice)	Met
✓	CHANGELOG.md (nice)	Met
✓	LICENSE-NOTE.md	Met
✓	Web a project filee	Met

✓	Web a project file	Met
✓	Web a project file	Met
✓	Web a project file (nice)	Met
✓	a project file	Met
✓	a project file (nice)	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 4 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 4 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 4 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.