Codebase Audit Report

Client: [REDACTED — Sample Report]
Project: Enterprise SaaS Platform
Date: 27 June 2026
Auditor: Thuban Code Health Engine v0.3.2

CONFIDENTIAL

thuban.dev

Generated by Thuban Pro

This report contains sensitive findings about the client's codebase.
Distribution restricted to authorised personnel only.

1. Executive Summary

This codebase has **87 issues across 1,547 files**: 1 deprecated API, 59 hardcoded localhost URLs, and 27 placeholder content instances. All 87 are auto-fixable. The pattern suggests rapid AI-assisted development without adequate review gates.

Verdict: Not production-ready. Deploying without remediation creates upgrade risk, deployment failures, and ongoing maintenance cost of **£3,840/month** (£46k/year). Estimated fix time: **under 45 minutes** with Thuban auto-fix.

Metric	Value	Status
Files Scanned	1,547	—
Health Score	52 / 100 (D+)	CRITICAL
Total Issues	87	HIGH
Auto-fixable	87 (100%)	POSITIVE
Deprecated APIs	1	HIGH
Hardcoded URLs	59	HIGH
Placeholder Content	27	MEDIUM
AI Hallucination Risks	0	CLEAR
Circular Dependencies	0	CLEAR
Scan Time	1,453ms	—
Est. Monthly Tech Debt	£3,840	HIGH

2. Issue Breakdown

2.1 Deprecated APIs

Deprecated APIs are functions or methods that still work in the current version but are scheduled for removal. Using them creates a ticking time bomb — the code will break silently on the next runtime upgrade.

File	Line	Issue	Severity
[redacted]/server.js	24	url.parse() — deprecated since Node 11	HIGH

Fix: Replace `url.parse()` with `new URL()`. Thuban can auto-fix this.

What this means in practice: When you upgrade Node.js (which you will need to do for security patches), this API call will start throwing warnings and eventually break entirely. Your server will crash on a routine upgrade.

2.2 Hardcoded Localhost URLs (59 instances)

59 files contain hardcoded `localhost` or `127.0.0.1` URLs. These will fail in any non-local environment — staging, production, CI/CD, Docker containers, cloud deployment.

File	Line	Pattern Found
[redacted]/cdp-connector.js	21	`ws://localhost:9222`
[redacted]/cdp-connector.js	25	`http://localhost:9222`
[redacted]/debug-cdp.js	22	`ws://localhost:9222`
[redacted]/takeover.js	45	`http://localhost:3000`
[redacted]/takeover.js	78	`http://localhost:8080`
...	...	`54 more instances`

Fix: Replace hardcoded URLs with environment variables: `process.env.CDP_URL || 'ws://localhost:9222'`. Thuban auto-fixes all 59 instances.

What this means in practice: If you deploy this codebase to AWS, Azure, Docker, or any cloud environment, 59 connections will silently fail. Your app will appear to work locally but break in production. This is the number one cause of 'it works on my machine' failures.

2.3 Placeholder/Example Content (27 instances)

27 files contain example domains (example.com), placeholder emails, or test data that should not appear in production code. These indicate incomplete cleanup after development or AI-generated scaffolding that was never reviewed before merge.

What this means in practice: Placeholder content in production signals to customers, investors, and partners that the software was not professionally built. In regulated industries, test data in production code can trigger compliance failures.

3. Tech Debt Cost Analysis

Thuban estimates the ongoing cost of unresolved technical debt by calculating the developer time likely spent working around, debugging, or being blocked by each issue category.

Category	Issues	Est. Hours/Month	Est. Cost/Month
Hardcoded localhost URLs	59	12	£960
Deprecated APIs	1	2	£160
Placeholder content	27	6	£480
Architecture drift	—	8	£640
Missing documentation	—	10	£800
Onboarding overhead	—	10	£800
TOTAL	**87**	**48**	**£3,840**

Methodology: Cost estimates assume a blended developer rate of £80/hour. Hours include time spent debugging, working around issues, explaining to new team members, and incident response. Estimates are conservative.

Annual projection: At current levels, this technical debt costs approximately **£46,080 per year** in lost developer productivity.

4. AI Hallucination Analysis

Thuban's hallucination detector scans for phantom imports, invented API methods, non-existent modules, and deprecated patterns commonly produced by AI coding assistants (GitHub Copilot, Cursor, Claude, ChatGPT).

Check	Result	Status
Phantom imports (non-existent modules)	0 found	CLEAR
Invented API methods	0 found	CLEAR
Non-existent npm packages	0 found	CLEAR
Deprecated Node.js APIs	1 found	WARNING
Fabricated function calls	0 found	CLEAR

This codebase shows **no AI hallucination risks**, which is positive. The deprecated API issue (url.parse) may have been introduced by an AI assistant suggesting outdated patterns, but does not constitute a hallucination.

5. AI Slop Index

The AI Slop Index measures indicators of AI-generated code that was merged without adequate human review. These are patterns commonly produced by AI coding assistants that suggest scaffolding, placeholder content, or copy-paste artifacts were shipped to production.

Indicator	Count	Risk Level
Placeholder/example content	27	MEDIUM
Hardcoded demo values (localhost)	59	HIGH
Scaffold residue (TODO/temporary)	0	CLEAR
Duplicate code blocks	0	CLEAR
Unused/dead imports	0	CLEAR
Inconsistent naming patterns	0	CLEAR
Tests that assert nothing	0	CLEAR
Fabricated API methods	0	CLEAR
Config drift	0	CLEAR

AI SLOP INDEX: 86 / 87 issues are AI slop indicators

Interpretation: 99% of issues in this codebase are AI slop indicators — patterns that suggest code was generated by an AI assistant and merged without adequate review. This is not unusual for AI-heavy development, but it requires immediate cleanup before the codebase is shown to investors, clients, or new team members.

6. Mother Code DNA Coverage

Mother Code DNA is Thuban's proprietary system for making codebases self-documenting. Each file receives a structured header that declares its purpose, dependencies, what depends on it, and what breaks if it changes.

In plain terms: Mother Code DNA means every file in your codebase can answer four questions: What do I do? What do I need? Who needs me? What breaks if I change? This reduces onboarding time for new developers, eliminates accidental breakage during refactors, and removes single-developer dependency — your codebase explains itself instead of relying on one person's memory.

Metric	Value
Files with DNA headers	0 / 1,547 (0%)
Orphan files (no imports or exports)	[analysis available in Pro]
Files with unclear purpose	[analysis available in Pro]

Recommendation: Run `npx thuban fix . --fix` to inject Mother Code DNA headers into all files. This is a safe, non-breaking change that adds structured comments — no runtime code is modified.

6. Fix Priority Roadmap

Issues ranked by business impact. Thuban can auto-fix all items marked as auto-fixable.

Priority	Issue	Count	Auto-fix	Effort
P1 - CRITICAL	Deprecated API (url.parse)	1	Yes	5 mins
P1 - CRITICAL	Hardcoded localhost URLs	59	Yes	10 mins
P2 - HIGH	Placeholder/example content	27	Yes	10 mins
P3 - MEDIUM	Mother Code DNA (0% coverage)	1,547	Yes	15 mins
P4 - LOW	Architecture documentation	—	Partial	2-4 hours

Total estimated fix time: under 45 minutes using Thuban auto-fix. All fixes create individual git commits that can be reverted independently.

7. Recommendations

1. **Immediate (today):** Run `npx thuban fix . --fix --commit` to auto-fix all 87 issues. Review the git diff. This takes under 15 minutes and immediately raises the score from D+ to B or higher.

2. **This week:** Install the pre-commit gate (`npx thuban gate`) to prevent new issues from entering the codebase. Every commit is checked in under 3 seconds.

3. **This month:** Inject Mother Code DNA headers into all files. This makes the codebase self-documenting and dramatically reduces onboarding time for new developers.

4. **Ongoing:** Run `npx thuban baseline .` and integrate Thuban into your CI/CD pipeline. Only alert on NEW issues — no noise from existing debt.

5. **Quarterly:** Run a full Thuban audit and compare against the baseline. Share the health score trend with stakeholders. Target: A grade (90+) within 3 months.

8. Benchmark Comparison

How does this codebase compare? The table below shows this project's scores against Thuban's internal benchmark set, based on comparable codebases scanned across similar technology stacks and team sizes.

Metric	This Project	Thuban Benchmark	Target
Health Score	**D+ (52)**	C+ (72)	A (95+)
Issues per 1,000 files	**56**	28	< 5
Deprecated APIs	**1**	3	0
Hardcoded URLs	**59**	8	0
AI Hallucination Risks	**0**	4	0
Mother Code DNA	**0%**	15%	90%+
Auto-fix Coverage	**100%**	68%	100%
Est. Monthly Debt Cost	**£3,840**	£1,200	< £200

9. Codebase MOT Certificate

Like a vehicle MOT, the Thuban Codebase MOT gives a clear Pass, Advisory, or Fail verdict on whether your codebase is fit for purpose — safe to deploy, safe to scale, safe to invest in.

THUBAN CODEBASE MOT CERTIFICATE	
Overall Result	**FAIL**
Health Score	**D+ (52/100)**
Production Readiness	**Not Ready**
Investor Readiness	**Not Ready**
Deployment Risk	**HIGH**
Maintenance Risk	**HIGH**
Security Review Required	**YES**
AI Slop Indicators	27 instances detected
Critical Issues	1 (deprecated API — will break on upgrade)
High Priority Issues	59 (hardcoded URLs — will fail in production)
Medium Priority Issues	27 (placeholder content — unprofessional)
Auto-fixable	**87 / 87 (100%)**
Estimated Fix Time	**Under 45 minutes**
Next Review Due	90 days after remediation

What Breaks First

Risk	Trigger	Impact
Production deployment failure	Any non-localhost environment	59 connections fail silently
Upgrade failure	Node.js upgrade to v24+	Server crashes on line 24
Client embarrassment	User sees example.com content	Trust destroyed
Developer slowdown	New hire joins the team	No docs, no DNA, no architecture map
Investor due diligence failure	Technical review requested	D+ score visible immediately

Rebuild vs Repair Verdict

REPAIRABLE

This codebase is **repairable**. The core product architecture is sound. All 87 issues are auto-fixable and the estimated repair time is under 45 minutes using Thuban. A full rebuild is not required. However, continued development without addressing these issues will push the codebase toward the **Refactor Required** threshold within 3-6 months.

10. Board Verdict

This section provides a clear, actionable summary for founders, board members, and investors who need to make decisions about this codebase.

Assessment	Verdict
Production Readiness	**NOT READY — deploy at risk**
Investor Risk Level	**MEDIUM-HIGH**
Remediation Required	**YES — before scaling, raising, or onboarding new developers**
Estimated Repair Cost	£2,500 – £5,000 (remediation sprint)
Estimated Rebuild Cost	£40,000 – £80,000 (if left unaddressed)
Time to Fix (with Thuban)	5 working days
Should You Keep Funding This?	YES — but fix the foundation first

	BEFORE (Current)	AFTER (Post-Remediation)
Health Score	**D+ (52/100)**	**B+ (85/100) estimated**
Issues	**87**	**0**
Monthly Tech Debt	**£3,840**	**< £200**
Production Ready	**NO**	**YES**
Investor Confidence	**LOW**	**HIGH**

Bottom line: This codebase is salvageable. The issues are real but fixable. However, every week you continue building on top of this foundation without addressing the underlying problems increases your remediation cost and deployment risk. The sooner you act, the cheaper and safer it is.

11. Recommended Next Step

Thuban Remediation Sprint

Duration	5 working days
Scope	Fix all P1 and P2 issues identified in this audit
Deliverables	Clean codebase, Thuban gate installed, baseline set, post-fix report
Outcome	Raise health score from D+ (52) to B or higher (80+)
Ongoing Protection	Pre-commit hook blocks new issues from entering the codebase
Investment	From £5,000 depending on codebase size and complexity

All Thuban Services

Service	Price	Best For
Thuban CLI (Free)	**Free**	Developers — scan your own code
Thuban Pro	**£19/mo**	Solo devs & freelancers — full reports, auto-fix, CI gate
Thuban Team	**£99/mo**	Startups & agencies — team dashboards, baseline tracking
Snapshot Report	**£495 one-off**	Quick automated health check with PDF
Founder Audit	**£2,500 one-off**	Full audit + walkthrough + fix roadmap
Investor Audit	**£7,500 one-off**	Due diligence report for boards & investors
Remediation Sprint	**From £5,000**	We fix the issues — not just find them

To discuss this report or book a Remediation Sprint:

craig@thuban.dev | thuban.dev | Book a call

Thuban runs 100% locally. No code was transmitted externally during this audit.

Thuban | thuban.dev | *AI wrote the code. Thuban checks the damage.*