CASE STUDY

Vizuna

- Case Study



PROJECT OVERVIEW

Vizuna is an AI-powered workplace coaching platform designed to support continuous employee reflection, real-time behavioral insights, team feedback loops, and actionable workplace improvements. The client's initial prototype was built using rapid vibe coding and AI-assisted development (via Cursor and LLM code generators) to validate market fit. While this enabled fast prototyping, our engineering team was brought in to perform Vibe Coding Cleanup Services—taking the inherited AI-generated codebase and refactoring it into a secure, enterprise-ready architecture.

THE CHALLENGE

Input from the development team identified critical code-level defects, security flaws, and architectural limitations inherent to the AI-generated prototype:

- Client-Side Credential & API Exposure: Embedded LLM API keys and direct client-to-LLM REST calls in both mobile and web builds created immediate compliance and abuse risks.

- Sensitive Data Leakage: PII and confidential enterprise feedback were sent raw to external AI model endpoints without redaction or sanitization.

- Unenforced Multi-Tenant Isolation: The database schema lacked Row Level Security (RLS), risking cross-organization data leakage and improper seat/role allocation.

- Session Amnesia & Unhandled Errors: Unstructured prompt management resulted in context loss across coaching sessions, while unhandled async errors led to frequent client crashes.

OUR CLEANUP PROCESS

Through our Vibe Coding Cleanup Services, we executed a systematic audit, refactoring, security remediation, and validation program:

1. Zero-Trust Backend Proxying: Removed all client-side API credentials and established a serverless proxy layer using Supabase Edge Functions with custom middleware for automatic PII redaction and token rate-limiting.

2. Database Security & RLS Implementation: Audited the entire PostgreSQL schema, writing and verifying strict Row Level Security (RLS) policies for all multi-tenant tables.

3. Architectural & Context Optimization: Refactored mobile client layers (Android Kotlin/Compose and iOS Swift/SwiftUI) and the Next.js Admin Hub, introducing structured context-stitching pipelines for reflection history and pulse inputs.

4. Testing & QA Automation: Built regression test suites covering API security, edge function error handling, database permission checks, and core user reflection loops.

TECHNOLOGIES AND TOOLS

- Static Analysis & Debugging: Cursor IDE, ESLint, TypeScript Compiler Static Analysis, SonarQube.
- Backend & Security: Supabase Edge Functions (Serverless TypeScript), Node.js, Custom PII Redaction Middleware.
- Database & Data Access: Supabase PostgreSQL, Row Level Security (RLS), pgTAP (Database Unit Testing).
- Client Surfaces: Native Android (Kotlin/Compose), Native iOS (Swift/SwiftUI), Next.js Admin Hub.
- Testing & CI/CD: Jest, Playwright, GitHub Actions CI/CD pipelines.

RESULTS (QUANTIFIED EVIDENCE)

- 100% Client Credential Elimination: Removed 3 exposed API keys and 12 direct client-to-LLM endpoint calls from production mobile and web client bundles.
- 100% RLS Enforcement: Created, applied, and verified 24 Row Level Security (RLS) policies across all database tables, achieving 100% automated test pass rate for multi-tenant isolation scenarios.
- Security & Vulnerability Remediation: Identified and fixed 14 security findings (including 2 High-severity PII data exposure flaws and 5 Medium-severity authorization bypasses).
- Defect Resolution & Test Coverage: Resolved 18 code-level defects and introduced automated regression testing, increasing total test coverage from 0% to 82%.
- Performance & Reliability Gains: Reduced client crash rate from 4.2% to <0.1%, lowered average LLM response latency by 35% via context payload optimization, and eliminated 100% of PII exposure in outbound prompt payloads.

BUSINESS IMPACT

By applying our Vibe Coding Cleanup Services, Vizuna successfully transformed from an unoptimized AI prototype into a secure, enterprise-grade SaaS platform:

- Enterprise-Ready Compliance: Assured enterprise buyers that sensitive internal feedback remains strictly protected, sanitized, and isolated.

- Scalable Architecture: Established a maintainable, serverless infrastructure capable of handling high user concurrency without client stability issues.

- Technical Debt Reduction: Delivered a clean, fully typed codebase that reduced new feature onboarding time for engineers by 40%.