Projects from Anil Prathipati

Product Vision

Three Products That Plug Real Gaps in the Dev Workflow

Security, reliability, and traceability — the three things every engineering team needs and none of the existing tools nail all at once.

01	02	03
PendulineVault	Functional Test Suite Platform	Database Schema Change Tracker

Anil Prathipati

> ***"Secrets should be locked down. Deployments should never break what already works. And nobody should have to ask 'who changed that column' ever again."***

PendulineVault

Secrets management that's actually built for teams that aren't enterprise.

THE PROBLEM

Every application has secrets — API keys, database passwords, certificates, tokens. And most small-to-mid-size teams are managing them the wrong way: hardcoded in config files, copy-pasted into environment variables, or sitting in a shared spreadsheet somewhere. It's one leaked key away from a serious breach.

The enterprise tools that solve this — HashiCorp Vault, Akeyless — are built for organizations with a dedicated DevSecOps team. They're overkill for a 10-person startup or a mid-size company that just wants their secrets safe without a six-month implementation project.

WHAT WE'RE BUILDING

A lightweight, developer-friendly secrets management platform that gives teams enterprise-grade security without enterprise-grade complexity. AES-256-GCM encryption, JWT-based auth, role-based access control, and client libraries in Java, Python, .NET, and Node.js — so any app can pull secrets safely in minutes, not weeks.

- ✓ **AES-256-GCM Encryption at Rest**
  Every secret is encrypted with industry-standard encryption before it ever touches storage.
- ✓ **Multi-Language Client Libraries**
  Java, Python, .NET, Node.js — plug it into whatever stack the team is running.
- ✓ **Role-Based Access Control**
  Control exactly who and what can access which secrets. Audit who accessed what and when.
- ✓ **Zero-Config Local Dev Mode**
  Developers get a frictionless local setup. No DevOps overhead just to run the app locally.
- ✓ **Open-Core Pricing Model**
  Free tier for small teams. Paid tiers unlock SSO, audit logs, and compliance features.

WHO NEEDS THIS	WHAT THEY GET
• Startups outgrowing .env files	• Secrets managed centrally, not scattered everywhere
• Teams handling client data or PII	• Compliance-ready audit trail out of the box

• Companies pursuing SOC 2 or ISO 27001	• Drop-in replacement for environment variables
• Dev teams tired of secret-related incidents	• Path to FIPS 140-2 when you need it

83%	**< 1 day**	**Free Tier**
of breaches involve credentials or keys	to integrate with an existing app	for teams under 5 applications

The Vision: Any team — 3 developers or 300 — should be able to manage their secrets with the same discipline as a Fortune 500 security team. No consultants, no six-month rollout. Just plug it in and your secrets are safe.

Functional Test Suite Platform

Don't let new rules kill old business logic.

THE PROBLEM

Every team I've talked to has the same story — someone adds a new business rule, pushes it to production, and three days later support tickets start coming in. The new rule quietly broke something from six months ago that nobody thought to test. That's not a developer problem, that's a tooling problem.

Most existing testing tools are either too low-level (unit tests that developers have to write themselves) or too heavy (full QA platforms that need a dedicated team). There's nothing in the middle that just works out of the box for functional coverage.

WHAT WE'RE BUILDING

A ready-made application testing framework where teams can define their business rules and the platform automatically validates that existing functionality still holds every time a change is introduced. No writing test scripts from scratch. No maintaining massive test suites manually.

- ✓ **Rule-Based Test Generation**
  Define your business rules once — the platform builds the test scenarios around them automatically.
- ✓ **Regression Guard on Every Deploy**
  Every new rule gets validated against existing functionality before it reaches production.
- ✓ **Plain English Rule Definitions**
  Non-technical stakeholders can describe business rules. The platform translates them into tests.
- ✓ **Pass/Fail Reports with Context**
  When something breaks, you know exactly which rule caused it and which business case it affects.
- ✓ **CI/CD Integration**
  Plugs into existing pipelines — GitHub Actions, Jenkins, whatever the team already uses.

WHO NEEDS THIS	WHAT THEY GET
• Dev teams shipping rule changes frequently	• Stop breaking existing customers with new features
• Teams with complex conditional business logic	• Deploy on Friday without the anxiety

• Organizations with no dedicated QA headcount	• Catch regressions before production, not after
• Companies doing rapid product iteration	• Cut down support tickets from logic failures

~60%	3–5 days	Zero
of production bugs are regression failures	avg. time to catch a regression in prod	test scripts needed to get started

The Vision: A team shouldn't need a QA department or 10,000 unit tests to have confidence in their deployments. Build the rule, deploy it, and know nothing else broke. That's it. That's the product.

Database Schema Change Tracker

Know exactly who changed what in your database, and when.

THE PROBLEM

Database schema changes are one of the most dangerous and least tracked things in any software project. A column gets dropped, an index disappears, a data type changes — and unless someone kept notes, you're playing detective trying to figure out who did it and why the application broke.

Git tracks code. Tickets track features. But nobody's tracking schema changes with the same rigor. That gap causes outages, compliance headaches, and a lot of "I thought someone else was handling that" conversations.

WHAT WE'RE BUILDING

A schema management platform that watches your database and logs every structural change — table added, column modified, index dropped, constraint changed — along with who made the change and exactly when. Full history. Full accountability. No manual changelog required.

- ✓ **User-Level Change Attribution**
  Every schema change is tied to a specific user or service account. No more mystery modifications.
- ✓ **Full Structural Diff History**
  See exactly what the schema looked like before and after every change, side by side.
- ✓ **Alerts on Critical Changes**
  Get notified when columns are dropped, tables are renamed, or other high-impact changes happen.
- ✓ **Multi-Database Support**
  Works across PostgreSQL, MySQL, MSSQL — wherever your data lives.
- ✓ **Audit-Ready Export**
  One-click export of the full change log for compliance reviews, SOC 2 audits, or post-mortems.

WHO NEEDS THIS	WHAT THEY GET
• Teams with multiple devs touching the same database	• Full change history with user attribution
• Organizations with compliance or audit requirements	• Instant answers during incident investigations

• Companies running microservices with shared schemas	• Compliance-ready audit trail without manual effort
• Any team that's ever had an unexplained prod outage	• Confidence that schema changes are tracked and reversible

100%	Per User	Minutes
of schema changes captured automatically	attribution on every structural change	to set up vs. weeks of manual logging

The Vision: Your database schema should have the same version control discipline as your codebase. Every change should be traceable, every decision should be documented, and when something breaks, you should know in two minutes — not two days.

Anil Prathipati · Product Vision