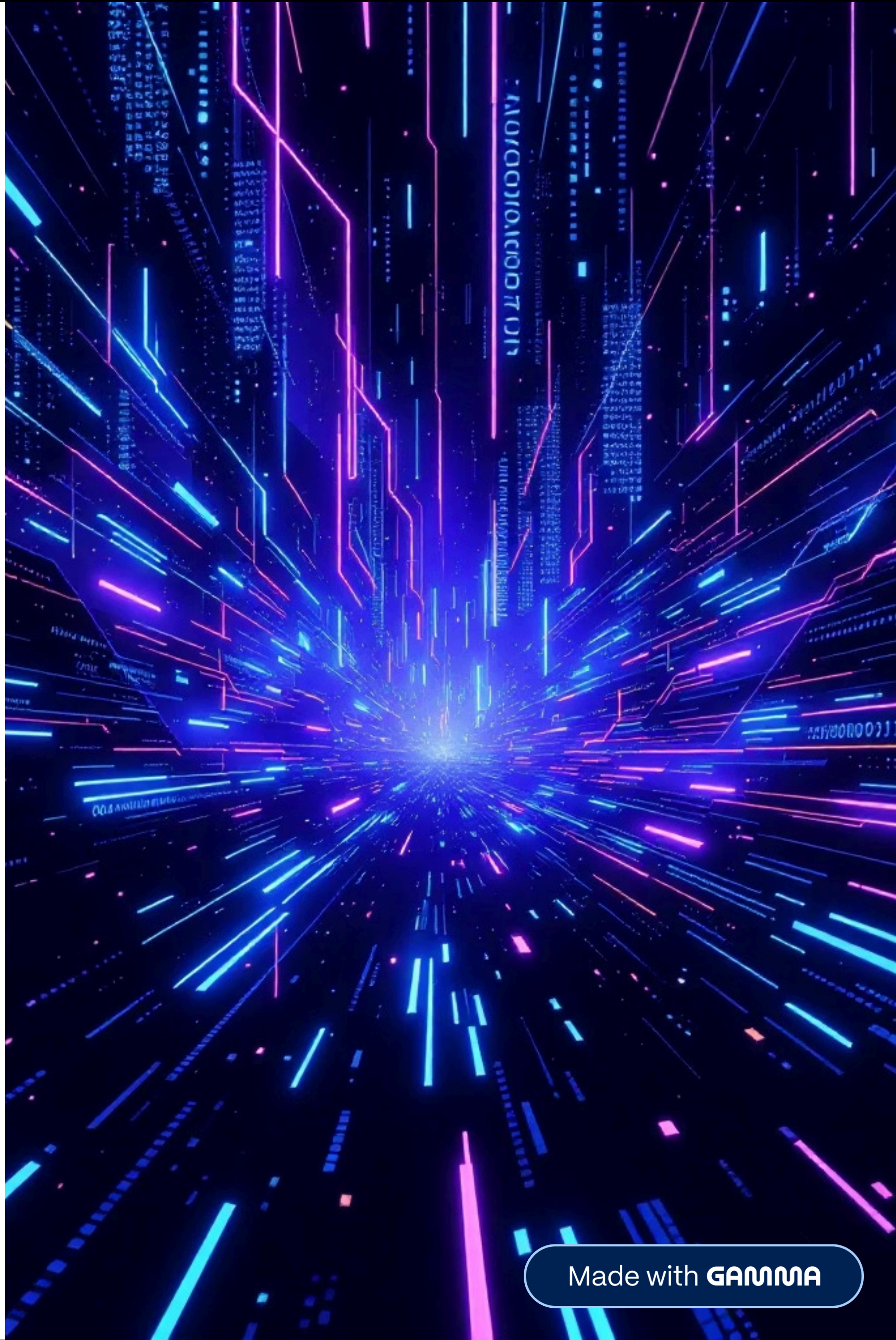
Chillax.AI

Stop Stressing, Start Understanding

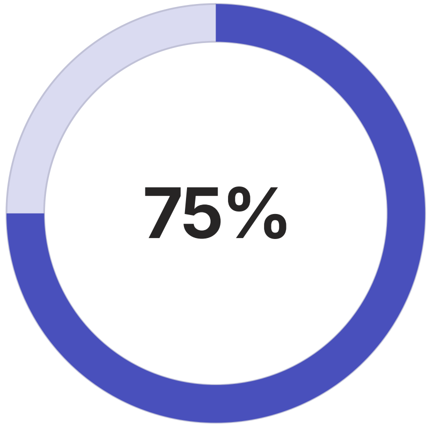
Your offline, AI-powered code companion for untangling legacy systems

ET Gen AI Hackathon — Open Novel Innovation Category | Built for Hackathons 🚀



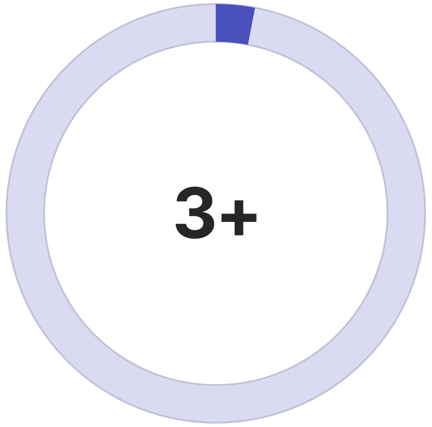
The Problem

Developers are drowning in code they didn't write.



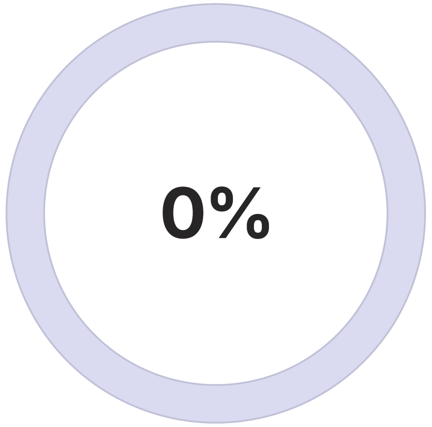
Time Reading Code

Not writing it



Weeks to Onboard

On legacy systems



Code Leaving Machine

Privacy-first

The Gap

Existing AI tools like GitHub Copilot and ChatGPT require sending proprietary code to external servers — unacceptable for enterprises in finance, healthcare, and defense.

> "Enterprise codebases can't afford cloud-based AI assistants"

Tools exist to help developers write faster, but nothing helps them **understand existing code at a system level**.

Our Solution

A standalone desktop IDE combining visual mapping, offline AI, and execution visualization.

Interactive Code Map

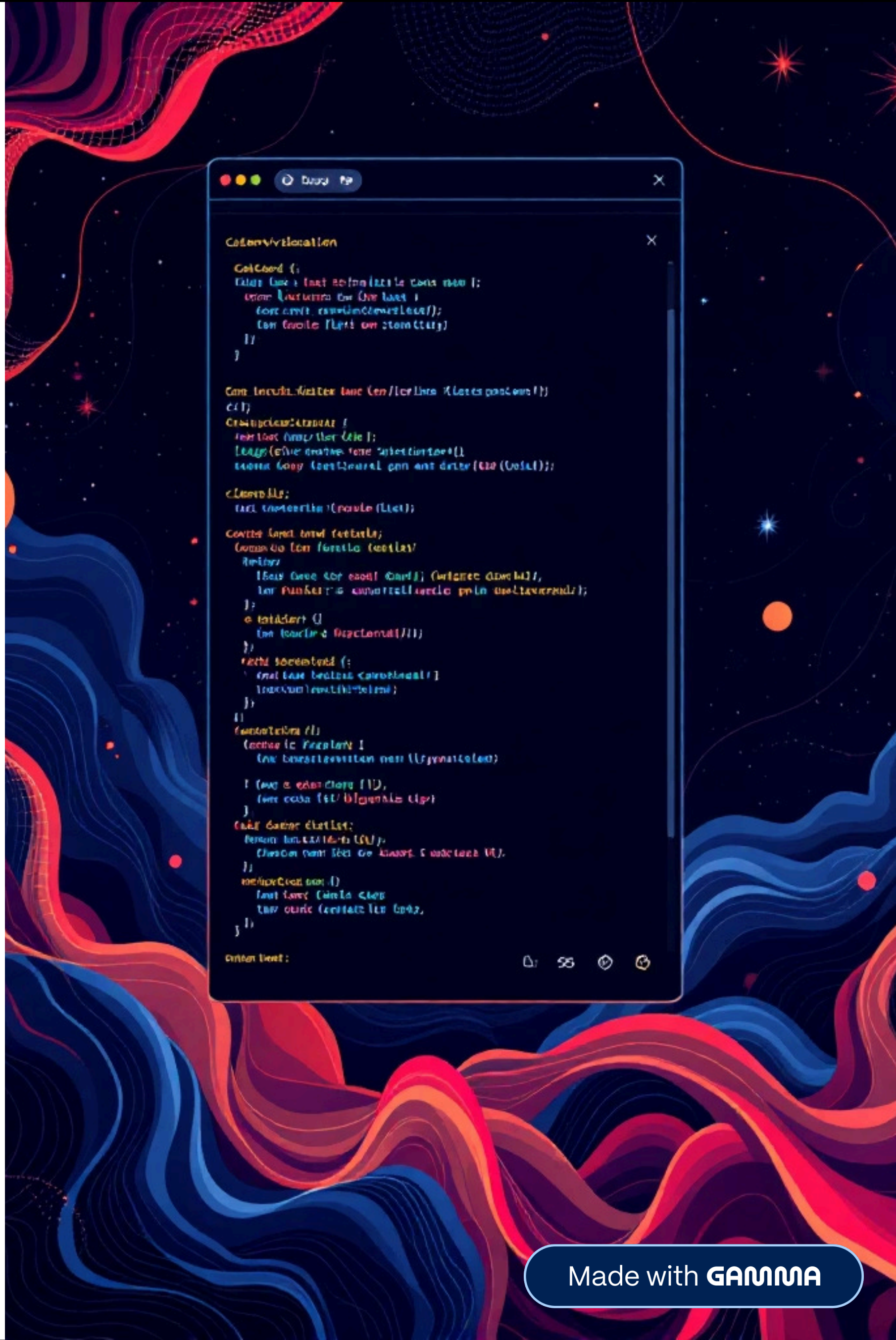
Visual graph showing all modules, functions, classes, and their relationships

Offline AI Assistant

LLaMA 3 running locally explains code in plain English with full context

Execution Visualizer

Animated walkthrough showing code flow with playback controls



How It Works

Visual Graph Theory + GenAI

Unlike chatbots that just output text, **Chillax.AI combines visual mapping with AI comprehension** — we show code execution visually.

Privacy-First Architecture

All code analysis and AI processing happens **100% offline** on the developer's local machine.

Zero data leaves the machine. No cloud dependencies. No proprietary code exposed.

Load Codebase

Analyze Structure

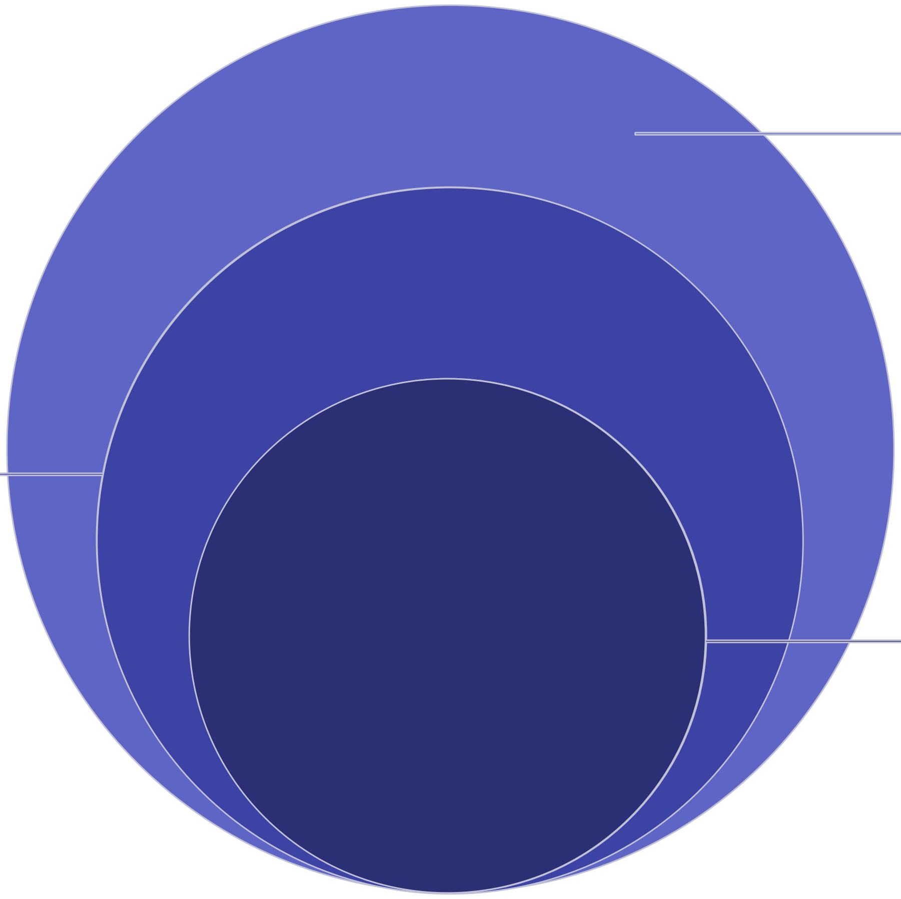
Ask Questions

Understand Instantly

Success metric: **60% reduction in code comprehension time**

Technical Architecture

Three-Layer Neuro-Symbolic Approach



GenAI + Viz

Generate explanations and interactive visualizations

Knowledge Graph

Represent relations and feed graph to GenAI

Symbolic Analysis

Parse code into AST and extract semantics

Tech Stack: Electron, React 18, Vite, Monaco Editor, FastAPI, Python AST, Ollama, React Flow, Custom dark theme CSS

Validation Strategy

Three complementary approaches to prove effectiveness.

1. **Accuracy Testing**

   Compare AI-generated explanations against ground-truth AST dependency graphs

2. **Performance Benchmarking**

   Measure inference latency targeting sub-5-second responses on standard laptops

3. **User Study**

   Simulated onboarding tasks measuring time-to-discovery with vs. without Chillax.AI

Target Users & Use Cases

Privacy-Sensitive Industries

Banking, healthcare, and defense teams working with proprietary, classified, or regulated codebases that cannot leave local machines.

Legacy System Maintainers

Engineers supporting decade-old codebases that lack documentation but power critical business operations.

Computer Science Educators

Professors and TAs using visualization to teach students how complex systems actually work.

Use Case Scenario

A developer joins a team with a massive, undocumented 10-year-old Python codebase. They open **Chillax.AI**, visualize the architecture as an interactive graph, and ask "How does the payment module work?" — **all without data leaving their machine**.

Why Chillax.AI?

Neuro-Symbolic Innovation

Combine deterministic AST parsing with probabilistic LLMs to reduce hallucinations and increase accuracy.

Truly Offline by Design

Privacy and security aren't features — they're fundamental to our architecture. No cloud dependencies.

Visual Execution Flow

Go beyond text explanations with animated code execution visualization showing exactly how systems work.

Language-Agnostic Future

Python-first, but architecture adapts to JavaScript, Java, C++, Go, and more. Universal legacy platform.

Novelty: Visual Graph Theory + GenAI

This hybrid approach allows us to show code execution visually within a unified interface where developers simultaneously explore architecture graphs, engage the AI assistant, and watch animated execution traces — all on their local machine.

GitHub: [your-github-link] | Team: [your-team-name] | Contact: [your-email]

Take the Stress Out of Legacy Code

Chillax.AI

Built for the ET Gen AI Hackathon — Open Innovation Category

View on GitHub

Watch Demo

Team: [Team Name] | Contact: [Team Email]