Collin Kokotas

Staff Software Engineer

Elk Grove, CA · collinkokotas@gmail.com · github.com/hoodiecollin · linkedin.com/in/collinkokotas · hoodiecollin.dev

SUMMARY

Staff engineer and product-minded builder with 11+ years shipping web applications and AI products end-to-end — increasingly at principal scope. I conceived, pitched, and single-handedly built Velo™ — Recentive's flagship AI predictive-analytics platform — grew it into the company's #1 revenue source, and architected it into a platform multiple teams now build on. Deep in TypeScript/Next.js and agentic AI; at my best owning ambiguous, high-impact problems from zero to production, setting the technical direction others build behind, and lifting teams through architecture, documentation, and mentorship.

SKILLS

Languages: TypeScript / JavaScript (11+ yrs), Rust, Go, Python

AI & Agents: Vercel AI SDK, LangGraph, OpenRouter, Anthropic, OpenAI, Ollama, tool-calling, structured-output extraction, schema-aware query generation, fact-grounding, prompt engineering

Frameworks & Infra: Node.js, Next.js, React, Redux, GraphQL, SQL, Snowflake, PostgreSQL, Docker, AWS, GCP, Redis, CI/CD, Jest, Cypress

Practices & Tools: Monorepo architecture, type-safe codegen, CI/CD (GitHub Actions), Storybook, technical leadership & mentorship, cross-functional Agile/SDLC

EXPERIENCE

Recentive Analytics — Staff Software Engineer Sep 2024 – Jul 2026

- Conceived, pitched, and built the proof-of-concept and v1 of Velo™, an AI-powered predictive-analytics and optimization platform for sports, media, and entertainment.
- Grew Velo™ into the company's #1 revenue source — generating more than all other revenue streams combined — scaling to 12+ enterprise clients (several multinational) and a few hundred executive and decision-maker users.
- Built the agentic-AI layer end-to-end (Vercel AI SDK, LangGraph, OpenRouter — Anthropic/OpenAI): tool-calling agents that generate and run warehouse queries via LLM-driven schema selection over a custom semantic-metadata layer I added to the Snowflake schema — an approach I adopted after vector RAG underperformed.
- Designed a generic file-ingestion engine (docx/csv/pdf/xlsx, a few MB to multi-GB) that uses an LLM to derive each file's purpose and semantic metadata and materialize it into dynamic Snowflake tables the agents query.
- Built “grounded facts,” a layer that mints verifiable facts from tool/query results and audits responses for unsupported figures — cutting responses containing ungrounded numbers from >60% to <10% across real customer threads.
- Architected the platform (monorepo, build tooling, type-safe codegen) and authored the architecture docs, design specs, and diagrams that onboarded data science, data engineering, and application teams to build on it.

Mayan — Staff Software Engineer May 2021 – Aug 2024

- Technical lead for the core web app; wrote custom build tools generating strongly typed SDKs from backend services, cutting front-end effort and catching breaking changes before deploy.
- Designed the Snowflake and PostgreSQL database schemas underpinning the product, and delivered third-party integrations across Amazon Ads / Seller-Central, GCS, Intercom, Looker, and Chargebee.
- Built a Figma-driven React component library with a Storybook showcase, and added runtime metrics and error logging that measurably reduced error frequency and improved performance.
- Established CI/CD (GitHub Actions) and a release process that eliminated broken deployments; guided QA to comprehensive Cypress end-to-end coverage, driving regressions per release to zero.
- Contributed directly to other engineering teams' Python and TypeScript projects, unblocking cross-team delivery beyond my own product area.
- As engineering & QA manager, led a daily cross-functional standup that cut support-ticket resolution from 1–2 weeks to 1–3 days, overhauled the technical hiring process, ran annual performance reviews, and coached struggling engineers back to standing through structured improvement plans.

AVB Marketing — Senior Engineer Apr 2018 – Mar 2021

- Front-end lead for an e-commerce platform powering 550+ websites; raised average Lighthouse performance from 12 to 92 and built a prerender service taking product-page SEO from 0 to 100.
- Integrated Amazon Payments, Synchrony, and PayPal (+33% cart conversion); Dockerized the project, cutting new-developer onboarding time by 25%.

KeeperSecurity — Senior Engineer Apr 2017 – Mar 2018

- Rewrote the browser-extension autofill algorithm (~2s → <100ms) and cut site-compatibility failures from 20% to 3%.
- Extended the autofill engine to support payment and address forms, broadening coverage across the sites users relied on.

VISA — Software Engineer Apr 2015 – Mar 2017

- Built a QA request-recording proxy that drove “cannot reproduce” tickets to near zero on the Express Checkout widget; optimized bundle size and client-side loading.

SELECTED PROJECTS

ForgeDB · forgedb.dev 2026

An application-database generator: write one declarative .forge schema and, at compile time, get a tailored Rust database, a typed TypeScript SDK, and a REST API with an OpenAPI 3.1 spec. It's a code generator, not an ORM or query engine — the output is specialized per schema over columnar storage, so there's no generic runtime to pay for.

typescript-to-rust (ttr) · github.com/HoodieCollin/typescript-to-rust 2026

Language-level translator that compiles a strict TypeScript dialect into idiomatic Rust with true ownership semantics (borrows vs. owned values, &self / &mut self methods), with output verified by a real cargo toolchain rather than string matching.

Optigon · github.com/hoodiecollin/optigon 2026

Packages several interchangeable implementations of an operation (sorting, dictionary lookup, more to come) behind one interface, then learns per workload which is fastest via regret-scored adaptive dispatch. One Rust core, shipped as native addons to TypeScript (Node + Bun) and Python.

checked-rs · github.com/HoodieCollin/checked-rs 2024

Rust library that encodes arbitrary validation logic into the type system, with a proc-macro that generates specialized, self-validating integer types.

use-quantum-state · github.com/HoodieCollin/use-quantum-state 2024

TypeScript / React library for fine-grained cross-component state: subscribers update without replacing the provider's value, so only components bound to a changed property rerender — cutting render cycles in high-frequency UI such as large tables.

EDUCATION

Hack Reactor — Advanced Software Engineering Immersive San Francisco, 2014