

Yiyuan Li

Singapore, River Valley | yy@eyuan.me | +65 895 234 47 | website: eyuan.me | linkedin.com/in/yyuanl
github.com/yuann3

Summary

Software engineer and full-stack engineer who likes taking things apart. 2+ years of experience, mostly in Rust and C, with Python, TypeScript, and Go when the job calls for it. I've built a SQLite engine from scratch, a Redis server, a GPU shader renderer that runs inside Emacs buffers, and a high-performance AI multi-agent swarm runtime that can spawn dozens of concurrent agents. I'm happiest figuring out how something works one layer deeper than I need to. I think computer and Emacs is cool, and so are house music and movies.

Technical Skills

Languages: Rust, C, C++, Python, SQL, JavaScript, TypeScript, Emacs Lisp, GLSL, Bash, Go (Golang)

Frameworks & Libraries: Tokio, wgpu, Vue 3, Hono, Mastra AI, React.js, Next.js, FastAPI, Flask, Node.js, REST API, JWT

Databases & Tools: Supabase, Redis, SQLite, PostgreSQL, MSSQL, ChromaDB, AWS, Turborepo, Docker, Git, Linux, CI/CD

Concepts: Multi-Agent AI Orchestration, Actor Model, Distributed Systems, Microservices, RAG, API Design, Multithreading, Concurrency, FFI/Dynamic Modules, TDD, Agile

Experience

Software Engineer – AI, ClassDo Pte. Ltd. – Singapore Jan 2026 – Present

- Pair-programmed with a small team to design and build a multi-agent AI curriculum platform from scratch – owned the software architecture decisions on how agents coordinate, validate, and hand off work
- Built the backend in a Turborepo monorepo: Hono middleware with JWT auth, RBAC, typed RPC client, and 809+ tests across 59 files. Deployed with AWS
- Integrated Singapore's SkillsFuture Skills Framework into Supabase (PostgreSQL), seeding 269K rows across 10 tables with multi-country schema support
- Built the Vue 3 frontend with Vike SSR, Pinia state management, and PrimeVue, with Storybook for the design system
- Tech Stack: TypeScript, Mastra AI, Hono, Vue 3, Supabase, PostgreSQL, AWS, Turborepo, Vitest, Zod

Software Engineering Intern, Newcastle Australia IHE Pte. Ltd. – Singapore Jan 2025 – Sept 2025

- Designed and developed a full-stack RAG AI learning platform using FastAPI, React 19, and Ollama, enabling intelligent document Q&A for 500+ users
- Implemented document parsing pipeline with LlamaParse and vector search using ChromaDB, reducing query response time by 40%
- Built OAuth 2.0 authentication system and RESTful API endpoints handling 1,000+ daily requests
- Tech Stack: Python, TypeScript, React 19, Next.js, FastAPI, Flask, Tailwind CSS, JWT, ChromaDB

Technical Leader, Data Structures & Algorithms, The University of Newcastle – Singapore Feb 2024 – Nov 2024

- Led intensive algorithm workshops breaking down complex topics: dynamic programming state transitions, graph traversal edge cases, and recursive backtracking optimization
- Developed custom teaching materials for difficult concepts—visualizing B-tree rebalancing, explaining amortized complexity, debugging off-by-one errors in binary search

Projects

MRA: Multi-Agent Runtime for Rust

[View on GitHub](#) ↗

- Designed an Erlang/OTP-inspired agent runtime where each AI agent runs as a Tokio actor with bounded mpsc mailboxes, zero shared mutable state, and transparent restarts via ArcSwap mailbox slots
- Implemented a supervisor with OneForOne/OneForAll restart strategies, exponential backoff, hang detection,

and peer injection – agents survive crashes without callers noticing

- Technologies: Rust, Tokio, async/await, Actor Model, Serde, Figment, OpenRouter API

Emaclaude: Claude Code Orchestrator

[View on GitHub ↗](#)

- Built a 3-agent coding workflow (planner, coder, reviewer) inside Doom Emacs split buffers, coordinated by a ~130-line Rust CLI and an Elisp orchestrator piping JSON state and effects
- Implemented a review loop with double-confirmation passes to catch hallucinated approvals, plus inline diff comments and one-command PR creation via Magit and gh CLI
- Technologies: Rust, Emacs Lisp, Claude Code CLI, Magit, Serde, Clap, GitHub CLI

ShaderView: GLSL Shader Engine for Emacs

[View on GitHub ↗](#)

- Built a GPU-accelerated GLSL shader renderer that runs inside Emacs buffers using wgpu (Vulkan/Metal), with hot-reload on save and Shadertoy-compatible uniforms (iResolution, iTime, iMouse)
- Wrote a Rust dynamic module (cdylib FFI) bridging Emacs Lisp to a pure-Rust GPU core, supporting concurrent shader sessions, PNG export, and configurable frame rates
- Technologies: Rust, wgpu, Emacs Lisp, GLSL, FFI/Dynamic Modules, Vulkan, Metal

Sequel: SQLite Database Engine

[View on GitHub ↗](#)

- Built a SQLite-compatible database engine from scratch in Rust, implementing B-tree data structures and binary file parsing to read .db files without external dependencies
- Implemented SQL query execution including SELECT with WHERE clauses, COUNT aggregation, and index optimization achieving 3x faster lookups through B-tree traversal
- Technologies: Rust, B-tree, Binary Parsing, Database Internals, File I/O

Rego: Redis Server Implementation

[View on GitHub ↗](#)

- Implemented Redis-compatible server in Go with core commands (GET, SET, EXPIRE) and WAIT replication, passing 100% of Codecrafters integration tests
- Optimized memory allocation using slice pooling, reducing garbage collection overhead and improving throughput by 30%
- Technologies: Go (Golang), TCP/IP Networking, Concurrency, Memory Optimization

Ruskey: Programming Language Interpreter

[View on GitHub ↗](#)

- Developed a complete interpreter for Monkey language in Rust: lexer, recursive descent parser, AST, and tree-walking evaluator
- Implemented closures, first-class functions, and hash maps using Test-Driven Development (TDD) methodology
- Technologies: Rust, Compiler Design, Abstract Syntax Trees (AST), TDD

Education

Singapore University of Technology and Design (SUTD), 42 Singapore
Computer Science

Nov 2025 – Nov 2026

- Core Curriculum: Peer-to-peer software engineering program focused on C, Unix, and systems programming
- Completed 16+ projects: libft, ft_printf, get_next_line, push_swap, pipex, Born2beRoot

The University of Newcastle, Bachelor of Information Technology

Jan 2024 – Sep 2025

- High Distinction: Object-Oriented Programming (OOP)
- Distinction: Data Structures and Algorithms (DSA), Advanced Database Systems