

AARIZ SHEIKH

Mumbai, India | arizsheikh17@gmail.com | +91 9137871445 Portfolio: aariz.vercel.app | GitHub: github.com/arino08 | LinkedIn

EDUCATION

M.H. Saboo Siddik College of Engineering — Mumbai, India Bachelor of Engineering in Information Technology | Expected Graduation: 2027

TECHNICAL SKILLS

- **Languages:** Rust (Tokio, Async), Java (Core), TypeScript, JavaScript, SQL.
- **Systems Engineering:** Multithreading, TCP/IP Networking, Memory Management, Linux (Nobara/Arch).
- **Web Technologies:** Next.js 14, React.js, Tailwind CSS, D3.js, WebSockets, Docker.
- **Tools & Cloud:** Git, GCP (Cloud Run), GitHub Actions, Postman, MongoDB.

KEY ENGINEERING PROJECTS

FlashKV | High-Performance In-Memory Database

(Tech Stack: Rust, Tokio, Bytes, TCP) [GitHub Repository](#)

- Handled 10,000+ concurrent connections with zero data races by engineering a sharded `RwLock` architecture (64 shards) using Tokio's asynchronous runtime.
- Reduced heap allocation overhead by 40% compared to standard string handling by implementing zero-copy byte manipulation via the `Bytes` crate.
- Enabled drop-in Redis compatibility by writing a custom TCP parser for the RESP protocol from scratch, supporting SET, GET, and INCR commands.

RustyLoad | Async HTTP & TCP Load Tester

(Tech Stack: Rust, Clap, Hyper, Ratatui) [GitHub Repository](#)

- Validated system stability under 50,000+ RPS load by building a high-concurrency CLI tool that outperforms Python-based alternatives by 18x.
- Identified tail-latency bottlenecks in distributed systems by implementing statistical analysis algorithms to calculate and report P95/P99 latency metrics.
- Unified full-stack benchmarking by architecting a dual-protocol engine capable of stress-testing both HTTP web servers and raw TCP database connections.

Mugen | Infinite Canvas Whiteboard

(Tech Stack: Next.js 14, Rough.js, FlashKV, MongoDB)

- Reduced diagram retrieval latency by 300ms (vs direct DB hits) by integrating the custom-built FlashKV database as a high-speed caching layer.
- Enabled infinite-canvas rendering at 60 FPS by engineering a 2D matrix transformation engine using Rough.js for smooth pan/zoom operations.
- Ensured zero-latency user interactions during network syncing by implementing optimistic UI updates and background persistence to MongoDB.

OPEN SOURCE & CERTIFICATIONS

- **Google Cloud Skills Boost:** Earned 9+ Badges including GenAI, Cloud Infrastructure, and Secure Networking (2023).
- **Learning in Public:** Authoring a weekly engineering series on LinkedIn documenting the transition from High-Level Web Dev to Low-Level Systems Programming in Rust.