

Aryan Mishra

aryanmishra4002@gmail.com | aryan-mishra.site | linkedin.com/in/aryan-mishra | github.com/aryan55254

PROFILE SUMMARY

Backend engineer with a systems programming foundation, focused on distributed infrastructure and protocol-layer design. Builds in C++ and TypeScript with an emphasis on correctness under concurrency and measurable performance.

EDUCATION

Guru Tegh Bahadur Institute of Technology
Bachelor of Technology in Information Technology

New Delhi, India
Aug. 2024 – June 2028

EXPERIENCE

Open Source Contributor

Remote

- **uWestJS (Performance):** Reduced peak heap usage by **12.5% (6MB)** and eliminated GC pressure via module-level static constants for empty JSON bodies. [PR #51](#)
- **uWestJS (SEO):** Expanded metadata coverage for high-performance NestJS adapter features including backpressure handling, improving discoverability for production use cases. [PR #46](#)
- **passmark:** Resolved **6 dependency vulnerabilities** (CVE-2026-23745, CVE-2026-24842) via npm package overrides in `package.json`. [PR #34](#)

PROJECTS

Sentinel | *TypeScript, PostgreSQL Wire Protocol, AWS EC2*

[GitHub](#)

- Built a protocol-layer SQL firewall that speaks native PostgreSQL wire protocol on both sides, intercepting Query messages before they reach any backend instance. Blocked queries receive valid PG error frames — client drivers cannot distinguish Sentinel from a native Postgres instance.
- Engineered a connection plane with warm SSL-terminated pool per backend instance, multiplexing **2,000 concurrent client sessions** across 20 pool connections with **14–15ms average proxy latency** measured over the open internet against EC2-hosted instances.

Chronos | *C++, Lock-free Programming, Multithreading*

[GitHub](#)

- Built a low-latency job scheduler using a Chase-Lev work-stealing deque (wait-free owner, lock-free thieves) and an MPSC injection mailbox with spinlock producers and wait-free consumers, achieving **sub-5µs P50 latency** and **1M+ ops/sec** on mixed workloads.
- Eliminated OS scheduling overhead via `__mm_pause()` spin-wait workers, prevented convoy effects with randomized victim selection, and enforced cache locality via `alignas(64)` structures achieving perfect **4x linear scaling** on 4 threads.

Vortex | *TypeScript, Node.js, BullMQ, Redis, Docker*

[GitHub](#)

- Built a video processing backend as a multi-process monolith where the API and worker run as decoupled Docker containers communicating exclusively via Redis, keeping the API non-blocking under **4,000+ queued jobs**.
- Implemented duration-based priority queuing (BullMQ), WebSocket job status broadcast per user room, and a repeatable cleanup job running every 10 minutes to prevent disk exhaustion.

Scruffy | *C++, Memory Management, Multithreading*

[GitHub](#)

- Implemented a stop-the-world Mark-and-Sweep garbage collector in C++ with multi-source root tracking across a VM stack and global root array, supporting cyclic object graphs across INT, PAIR, and STRING types.
- Validated correctness under concurrent allocation via multithreaded stress tests, with automatic heap threshold adjustment triggering collection before exhaustion.

TECHNICAL SKILLS

Languages: C++, TypeScript, JavaScript, SQL

Backend: Node.js, Express.js, WebSockets, PostgreSQL, Redis, BullMQ, Docker, AWS EC2

Systems: Lock-free concurrency, memory management, protocol design, multithreading

Frontend & Tools: React.js, Next.js, Git, Linux

TECHNICAL WRITING

Engineering Blog: Deep dives on systems internals : cache mechanics, memory hierarchies, and Node.js runtime architecture at aryan-mishra.site/blog.