

Shreyas Ganesh

(352) 246-0133 | vansi.ganeshs@gmail.com | [linkedin.com/in/shreyas-ganesh](https://www.linkedin.com/in/shreyas-ganesh) | github.com/shreyasganesh0 | shreyas-portfolio.vansi-ganeshs.workers.dev

EDUCATION

University of Florida

Gainesville, FL

Master of Electrical & Computer Engineering

May 2026

- GPA 3.9/4.0. Coursework: Distributed Operating Systems, Parallel Computer Architecture, Advanced Systems Programming, Machine Learning.

PES University

Bengaluru, India

Bachelor of Computer Science & Engineering

May 2023

- GPA 8.24/10.0. Coursework: Operating Systems, Networks & Communications, Big Data, Computer Vision.

TECHNICAL SKILLS

Programming: C++ (C++20), C, Go, Zig, Python, Java, SQL, Assembly (ARM Cortex-M, x86)

Systems & Concurrency: Lock-free programming (atomics, memory ordering, CAS), Linux internals, device drivers (kernel modules), io_uring, async I/O, shared-memory IPC, performance profiling

Databases & Storage: PostgreSQL, MySQL, Redis, B-trees, LSM trees, WAL, hybrid BM25+vector search (Weaviate), page-format parsing, storage-engine internals

Distributed Systems: Replication protocols, consensus (Raft concepts), DHT (Chord), event-driven messaging (Kafka wire protocol, RESP), gRPC concepts

Cloud & DevOps: AWS (S3, Lambda, RDS, IAM), Terraform, Docker, Kubernetes, CI/CD (GitHub Actions with perf-regression gating), Linux/Unix

EXPERIENCE

Wabtec Corporation (Rail & Transit Manufacturing)

Bengaluru, India

Backend Software Engineer

Jan 2023 – Jul 2024

- Replaced record-by-record XML processing in a Java 8 + Apache Camel ingestion pipeline with batch-buffered loading, cutting query load ~20% across a 198,000-entries/day system
- Migrated 12 data-warehouse tables to an OLTP schema, eliminating full-table scans in ETL queries and cutting execution time by 20 seconds (5×) and 6 hours of monthly downtime
- Retuned Hazelcast cache eviction TTLs to match upstream recovery windows, reducing retry-induced latency spikes ~20% in a distributed caching layer
- Migrated 100,000+ XML payloads from database blob storage to AWS S3, decoupling storage I/O from query paths and improving response times ~23%

PROJECTS

Lock-Free MPMC Queue | C++, concurrency, CI/CD

github.com/shreyasganesh0/lockfree-queue

- Implemented a Vyukov-style bounded MPMC queue using `std::atomic` and per-slot sequence numbers; validated CAS loops on Ryzen 9 7940HS under producer/consumer saturation
- Designed false-sharing experiments demonstrating cache-line-aligned padding's impact on L1 miss rate; found that CCX-local sharing matches 64-byte padding at 1-byte granularity, recovering memory overhead
- Proved Store-Load reordering on x86 TSO via a Dekker's-algorithm stress test detecting violations across CPU saturation, cache thrashing, and concurrent I/O loads

SQLite Clone (Zig) | Zig, database internals

github.com/shreyasganesh0/sqlite-clone-zig

- Implemented a SQLite file-format parser and B-tree walker from scratch: 100-byte header validation, page-type discrimination (interior 0x05 vs leaf 0x0D), varint key decoding, schema traversal via DFS
- Walks the `sqlite_schema` table (always on page 1) to extract table metadata directly from `.db` byte streams

Python Redis Clone (PYDIS) | Python, asyncio, distributed systems

github.com/shreyasganesh0/python-redis-clone

- Built an asynchronous Redis clone using `asyncio`: handcrafted RESP-protocol parser, key expiry (PX with Unix-timestamp normalization), `CONFIG GET` introspection, and RDB-file parsing for cold-start persistence
- Shipped single-leader replication including the full handshake (PING → REPLCONF listening-port → REPLCONF capa → PSYNC), master-side full resync via RDB transfer, and write-command propagation

OPEN SOURCE

ClickHouse PR #89826 (MERGED) – Added `arrow_flight_request_descriptor_type` setting for Dremio compatibility via Arrow Flight Command descriptors (+231/−18, 15 files); merged by maintainers in 4 days

ScyllaDB PR #26920 (closed) – Eliminated duplicate JSON parsing on `put_item` LWT shard bounces in the alternator (DynamoDB-compatible) path via a `put_item_lwt_args` struct (+58/−11); 45/45 alternator tests pass (upstream #13254)