# Lay Sheth

Software Engineer
+91 7000035904 | laysheth1@gmail.com | linkedin.com/in/laysheth | github.com/cloaky233 | leetcode.com/laysheth

## Education

**Vellore Institute of Technology** — Bhopal, Madhya Pradesh
*Bachelor of Technology in Computer Science; GPA: 8.89/10.00* — *Sept. 2022 – May 2026*

## Experience

**Software Engineering Intern** — Sept. 2024 – Dec. 2024
*PreProd Corp* — *Bengaluru, India (Remote)*

- Decreased event processing latency by **40%** (250ms → 150ms) by refactoring the Kafka consumer service to use **async batching** and connection pooling, resolving throughput bottlenecks in the real-time data pipeline.
- Engineered a semantic search microservice using **ChromaDB** vector embeddings, improving query result relevance by over **50%** compared to the legacy full-text search implementation for the internal document portal.

**Open Source Contributor** — Mar. 2025 – Present
*Zed Industries & Rust Foundation*

- Eliminated a critical **application panic** (crash) affecting the LSP store by backtracing invalid journal paths and implementing defensive error propagation across 3 Rust modules, restoring stability for users [**#42117**].
- Accelerated environment debugging for **10K+ active users** by implementing a verbose toolchain inspection feature in **rustup**, directly resolving a long-standing community feature request for better path visibility [**#4270**].

## Projects

**EmbraceDB: B+Tree KV Storage Engine** | *C++23, WAL, Snapshots* — Nov. 2025

- Achieved 421K point lookups/sec (2.37µs latency) by implementing cache-optimized B+Tree degree-4 structure with binary search on sorted keys
- Engineered crash-safe durability via WAL (CRC32 checksums) and atomic snapshots, enabling sub-second recovery for full state restoration
- Optimized mixed workload to 215K ops/sec (70% read, 20% write, 10% update) by buffering WAL writes in 4KB batches and implementing in-place updates

**Agentless Backend API for monitoring** | *Rust, Tokio, SSH, Docker, LTO* — Oct. 2025 – Present

- Achieved <6MB runtime footprint by designing agentless Rust monitoring system, eliminating deployment overhead vs Grafana/Prometheus
- Reduced SSH timeout failures from 15% to <2% at 100 concurrent connections by implementing layered timeout orchestration for async execution
- Decreased Docker image size by 94% (500MB to 30MB) using multi-stage builds and LTO compilation

## Technical Skills

**Languages**: C/C++ (Modern C++23), Rust, Python, SQL
**Systems Programming**: Memory Management, Multi-threading, File I/O, Binary Serialization
**Data Structures & Storage**: B+Trees, Hash Indexing, WAL, Checkpointing, Page-based Storage
**Core Concepts**: Distributed Systems, Crash Recovery, CAP Theorem, Memory Management
**Databases & Messaging**: PostgreSQL, Redis, SQLite, Kafka, SurrealDB
**DevOps & Infrastructure**: Docker, Ansible, Linux, Kubernetes, Git, CI/CD Pipelines, AWS (EC2, S3)
**Core Competencies**: Performance Profiling, Crash Recovery, Asynchronous I/O
**Testing & QA**: Unit Testing, Integration Testing, Performance Profiling, Debugging Tools

## Achievements

**LeetCode Weekly Contest 478** : Global Rank 244/25,246 (Top 1%), solved all 4 problems
**SurrealDB Technical Ambassador**: Recognized as official ambassador; authored "LLM-as-a-Judge" RAG implementation featured in official engineering blog as reference architecture.
**National Finalist, Smart India Hackathon**: Led 6-person team to prototype low-cost myoelectric prosthetic hand with EMG gesture recognition, achieving 85% cost reduction vs commercial systems.
**Winner, PreProdCorp Buildathon**: Developed ML model comparison platform with interactive visualizations and 8+ algorithms in 24 hours, securing 1st place among peer teams.