

Devanshu Sharma

Dehradun, Uttarakhand

✉ devanshusharma658@gmail.com [in linkedin.com/in/devanshu-sharma-a1baaa243](https://www.linkedin.com/in/devanshu-sharma-a1baaa243) github.com/Devanshusharma2005

SUMMARY

Systems-focused software engineer with experience in distributed systems, backend performance optimization, and open source contribution. Proven ability to ship production-impacting work and collaborate in large codebases. Passionate about building correct, high-performance infrastructure and developer-facing systems.

EDUCATION

Graphic Era Hill University, Dehradun

Bachelor of Technology in Computer Science and Engineering

SKILLS

Languages: Go, C, C++, SQL

Backend & Systems: Distributed Systems, REST APIs, Algorithm Design, Concurrency, gRPC

Databases & Storage: PostgreSQL, Redis, etcd, MongoDB

Infrastructure & Tools: Docker, Git, Linux, Bleve, pgx, libpg_query

CS Fundamentals: Data Structures & Algorithms, Operating Systems, Computer Networks, DBMS, System Design

WORK EXPERIENCE

Clockwork Labs – Software Engineer Intern

Jun 2026 – Present

Working on **SpacetimeDB**, a distributed realtime database that unifies the database, application server, and synchronization layer into a single high-performance runtime.

Contributing to the open-source SpacetimeDB ecosystem, focusing on correctness, low-latency data access, and scalability for modern multiplayer and realtime applications.

Authoring technical blogs and educational content to demystify complex distributed systems concepts and drive developer adoption for the SpacetimeDB ecosystem.

Diving deeper into databases, distributed systems, and infrastructure for high-throughput, low-latency workloads as part of a small, systems-focused team.

Ashiba Research – Research Engineer Intern

May 2026 – Jun 2026

Built **ARC (Attestation Receipt Compiler)**, a deterministic claim-verification engine that ingests raw GPU telemetry, NVIDIA attestation tokens, and deployment logs to emit cryptographically-bounded **receipts** with explicit verdicts: **supported**, **contradicted**, **unknown**, or **not applicable**.

Designed and implemented a **GPU acceptance pass pipeline** that cross-validates declared SKU identity (`hwmodel`, `ueid`) against live **NVIDIA SPDM/EAT attestation tokens** collected via `nvattest` corelib mode, catching false-supported cases including SXM/PCIe topology mismatches and MIG configuration without GI/CI evidence.

Integrated **NVIDIA Remote Attestation Service (NRAS)** with a **device-bound nonce anti-replay layer**, ensuring each receipt is cryptographically fresh and tied to a specific evidence collection event rather than a cached or replayed attestation quote.

Authored an **evidence importer** converting heterogeneous raw sources (`nvidia-smi`, **NVML**, **DCGM**, OTEL spans, CloudTrail, and NVIDIA EAT JWTs) into a canonical artifact bundle, with each field carrying a provenance label and custody chain for downstream pass consumption.

Enforced strict **claim boundary discipline**: every receipt explicitly declares what the evidence does *not* prove (e.g., memory encryption status in CC-off mode), preventing false-confidence propagation to lenders and auditors relying on GPU collateral claims.

Cal.com – Software Engineer Intern

June 2025 – Oct 2025

Patched a critical **Stored XSS vulnerability** by enforcing strict file-type validation and sanitization for user-uploaded assets, hardening platform security.

Identified and eliminated multiple **$O(n^2)$ scheduling bottlenecks** by redesigning algorithms using **sorting**, **early-exit scans**, and **two-pointer techniques**, reducing complexity to **$O(n \log n)$** and improving performance by up to **25x+** in production workloads.

Redesigned **time-slot conflict detection** by pre-sorting intervals and eliminating redundant comparisons, significantly reducing computation for large scheduling datasets.

Optimized **date range intersection and filtering** using **two-pointer traversal** and **interval-based data structures**, enabling near-linear performance for multi-user availability queries.

Resolved critical edge cases including **booking duration overflow (99M+ minutes)** and cross-locale inconsistencies in scheduling flows.

Maintained a large **TypeScript monorepo** by reviewing **200+ PRs**, upholding high standards for performance, correctness, and code quality across a distributed team.

PROJECTS

Distributed Search Engine | *Go, etcd, Redis, Bleve, Embeddings, Docker*

Dec 2025 – Feb 2026

Built a **distributed search system** in **Go** indexing **24K+ Wikipedia documents** across **8 shards**, achieving **28ms p99 latency** at **7K+ QPS** via parallel query execution and multi-layer caching.

Designed a **hybrid ranking pipeline** combining **BM25** and **vector similarity** (384-dim embeddings), improving relevance for fuzzy and semantic queries.

Implemented an **8-node sharded architecture** with **etcd**-based service discovery and fault-tolerant coordination, maintaining availability during node failures.

Engineered a **parallel fan-out / Top-K merge** query engine, reducing end-to-end latency by **12x** vs single-node search.

Built a **Redis caching layer** with **distributed locking (SetNX)** to prevent thundering herd, achieving **sub-ms cache hits** under high concurrency.

Deployed a **15-service Docker cluster** (shards, coordinator, Redis, etcd, Ollama) for reproducible, production-like environments.

Diffo: Git-like Schema Versioning System | *Go, PostgreSQL, pgx, libpg_query*

Feb 2026 – Present

Built a **Git-inspired schema versioning system** using a **commit DAG**, enabling branching, merging, and reproducible database history beyond linear migrations.

Implemented core **graph algorithms** (DFS cycle detection, merge-base computation) to ensure correctness under complex and adversarial histories.

Designed a **deterministic ancestry engine** (BFS with stable ordering), eliminating nondeterministic schema resolution across environments.

Developed a **three-way merge engine** with structured diffs, producing **typed conflicts** instead of fragile text-based merges.

Built a **schema diff + migration engine** with **topological sorting** over FK dependencies, generating safe, minimal, and idempotent migrations.

Implemented **live schema capture** via **pg_catalog** introspection and a canonical DDL pipeline for consistent schema reconstruction.

OPEN SOURCE COLLABORATIONS

Vitess (CNCF Distributed SQL): Fixed a **race condition in vtgate** where concurrent goroutines accessed **PreSessions**, **ShardSessions**, and **PostSessions** during Rollback/Release flows. Introduced **safe snapshot helpers** to ensure thread-safe iteration; validated with Go's `-race` flag.

Supabase: Resolved a critical **frontend performance issue** caused by excessive `history.replaceState()` calls during rapid scrolling, triggering browser **SecurityError** limits and crashes. Designed a **rate-limited history wrapper** (50 calls / 10s) with **debouncing and deduplication**, eliminating crashes while preserving URL sync and TOC highlighting.

Enhanced **docs navigation UX** by adding **dropdown indicators** for nested sidebar items; fixed incorrect API duration example (`100y` → `876000h`) to ensure compatibility with supported formats.

RELEVANT COURSEWORK

- | | | |
|----------------------------------|-------------------------------|-------------------|
| • Data Structures and Algorithms | • Computer Networks | • Cloud Computing |
| • Object-Oriented Programming | • Computer Architecture | |
| • Operating Systems | • Database Management Systems | • Web Development |