

# Sudhanshu Shukla

Bhopal, Madhya Pradesh 462033

+91 8293041749 | [connect.sudhanshu.s@gmail.com](mailto:connect.sudhanshu.s@gmail.com) | [LinkedIn](#) | [GitHub](#) | [Codeforces](#)

## Achievements

---

- **iRage AlgoArena 2026:** Secured **3rd place** (All-India individual hackathon, UG students; quantitative trading / strategy design problem).

## Open Source Experience

---

**simdjson** – Fastest JSON Parser (**23k+** stars; used by Node.js, Meta Velox, ClickHouse, Microsoft FishStore) [GitHub](#) | **2026**

- Implemented architecture-specific SIMD optimizations for `find_next_json_quotable_character` in the JSON string builder.
- Added **LoongArch LSX** and **RISC-V RVV** intrinsics support following existing NEON/SSE2 patterns, enabling faster JSON parsing on these architectures.

## Projects

---

**Limit Order Book** | *C++20, Price-Time Priority, STL* [GitHub](#) | **Nov 2025 – Apr 2026**

- Built a **price-time priority LOB** in C++20 (`GoodTillCancel`, `FillAndKill`, `Market`) using `std::map<Price, std::list>` for **O(log n)** insert / **O(1)** best-price, and an `unordered_map` of iterators for **O(1)** cancel.
- Diagnosed and fixed a **heap-use-after-free** where structured bindings referenced an `unordered_map` node erased on the next line; collapsed 3 redundant hash lookups into 1.
- Eliminated an **O(n)** **per-insert** bug from `std::next` on a bidirectional iterator, and fixed `FillAndKill` cleanup that only inspected top-of-book.

**Live Strategy Backtester** | *C++20, Event-Driven Architecture* [GitHub](#) | **July 2025 – Apr 2026**

- Engineered a strictly decoupled event-driven trading engine using the **Observer pattern** and a custom **EventBus**.
- Profiled system latency via **core pinning**, hardware cache warmups, and **-O3/-march=native**, achieving **500k events/sec** throughput while keeping **p95 tail latencies under 150µs**.
- Integrated **simdjson** and **Boost.Beast** for high-throughput, async SSL WebSocket parsing of live Binance order books.
- Validated the event pipeline and modular components with isolated unit and integration tests using **GoogleTest**.

**FlashCache – In-Memory Key-Value Store** | *C++20, Linux epoll, RESP* [GitHub](#) | **Jan 2026**

- Built a single-threaded, edge-triggered `epoll` `RESP` server in C++20; benchmarked vs Valkey 8.0.7: **+11% on pipelined writes (1.22M req/s)** and **3× lower p99 latency**.
- Fixed silent I/O bugs: `partial-send()` desync (per-client write buffer with `EPOLLOUT`), 8KB read buffer wedging large payloads (dynamic, 64MB cap), `RESP` malformed-input injection.
- Cut `epoll_wait` syscalls **91%** (`strace`-verified) via batch tuning; added graceful shutdown via `signalfd` on `SIGINT/SIGTERM`.

## Technical Skills

---

**Languages:** C++20 (STL, Move Semantics, Smart Pointers), C++11/14, C, Python, SQL

**Systems:** Linux Epoll, Socket Programming (TCP/IP), UDP

**Tools:** Git, CMake, GDB, Valgrind, perf, Docker, Linux (Bash)

**Libraries:** Boost, Simdjson, Spdlog, GoogleTest, Redis

## Competitive Programming

---

- **ICPC Asia Amritapuri 2025:** Ranked **445** in Online Preliminary Round; qualified for Regionals.
- **Codeforces Expert:** Peak rating **1607** (Profile)
- **Meta Hacker Cup 2025:** Ranked **1605<sup>th</sup>** globally. (View)

## Education

---

**Rajiv Gandhi Technological University**

*B.Tech in Computer Science (AI & ML)*

**Bhopal, India**

*Oct 2022 – May 2026*

- **Relevant Coursework:** Operating Systems, Data Structures & Algorithms, Computer Networks, DBMS, OOP