

Aaryan Patel

SYSTEMS, SOFTWARE & ML INFRASTRUCTURE ENGINEER

Bogart, Georgia | aaryansp26@gmail.com | linkedin.com/in/aaryanpatelsystems
Portfolio + full project evidence | github.com/asp53826

PROFESSIONAL PROFILE

Computer Science senior graduating December 2026 with more than one year of industrial data and IoT engineering experience. Builds production data workflows at MP Equipment and public systems software from first principles, including Raft/MVCC databases, vectorized execution, LLM inference scheduling, retrieval infrastructure, and financial-data services. Work emphasizes reproducible benchmarks, failure injection, independent correctness oracles, and clearly documented limits.

CORE TECHNICAL SKILLS

Languages: C++17, Python, SQL, TypeScript/JavaScript, R
Systems: Distributed systems, Raft, MVCC/OCC, WAL and LSM trees, concurrency, vectorized execution, linearizability
ML / Data: PyTorch, HNSW and FAISS, LLM inference, ETL, data modeling, sensor analytics, signal processing
Platforms: Linux, Git and GitHub Actions, Docker, Power BI, M1 ERP, PLC and Industrial IoT integration

PROFESSIONAL EXPERIENCE

- MP Equipment**
Industrial IoT & Data Engineer Intern
Promoted from Computer Science Intern into a broader Industrial IoT and data engineering role.

Gainesville, Georgia
May 2025 - Present
- Build Python and SQL pipelines that combine IoT sensor streams, equipment alarm history, and M1 ERP production records into reliable operational datasets.
 - Standardize asset identifiers, measurement units, missing-value handling, and outlier treatment so engineering and operations reports remain consistent across sources.
 - Develop predictive-maintenance workflows from vibration, temperature, humidity, and machine-state signals to surface abnormal equipment behavior.
 - Design Power BI dashboards for OEE, downtime, on-time delivery, vendor performance, production KPIs, and equipment health.
 - Connect PLC-controlled machinery to real-time monitoring workflows that expose machine state and alarm behavior for troubleshooting.
 - Automate recurring operational reports to reduce manual data preparation and keep performance views current.
- Georgia State University - Department of Computer Science**
Undergraduate Research Assistant

Atlanta, Georgia
Aug 2022 - May 2023
- Analyzed Reddit social-network data to study group relationships and maintained research data models used in the analysis.
 - Built date-range controls and public-facing views for machine-learning and data-mining research data.

EDUCATION

- University of Georgia**
Bachelor of Science in Computer Science
Systems-focused engineering across distributed computing, ML infrastructure, data engineering, and signal processing.

Expected December 2026
- Georgia State University**
Prior coursework in Data Science

Aug 2022 - Apr 2024

SELECTED ENGINEERING PROJECTS

Public repositories include source, tests, benchmark commands, and documented failure boundaries.

Raft + MVCC Database Lab | C++17

- Implemented Raft elections and conflict repair with MVCC snapshots, serializable optimistic concurrency control, safe-point garbage collection, and P-compositional linearizability checking.
- Verified 598 assertions across seeded partitions, stale candidates, minority leaders, transaction conflicts, and invalid histories; five-node failover completed in 11 logical ticks without minority commits.

Paged LLM Inference Scheduler | Python, PyTorch

- Built paged KV caching, continuous batching, chunked prefill, prefix caching, and speculative decoding around a small-model inference simulator.
- Reached 94% KV-slot utilization versus 21%, 2.9x sequential throughput, and 5.5x better median time-to-first-token across 90 tests under a controlled CPU-scale benchmark.

SEC EDGAR MCP Server | Python, MCP, XBRL

- Developed issuer identity resolution, filing history, bounded text windows, XBRL discovery, per-host cache validation, and a global SEC request pacer.
- Measured a 138x warm 10-K read improvement with 32 offline tests while keeping peak request windows at the SEC 10-requests-per-second ceiling.

Vectorized Columnar Engine | C++17

- Implemented null bitmaps, selection vectors, vectorized filters, joins, aggregation, and TPC-H Q1 with scalar reference implementations as independent oracles.
- Measured 53.1 million TPC-H Q1 rows per second and 1.94x scalar-filter throughput with 6,288 assertions covering null and selection semantics.

Multi-Target Tracking and Sensor Fusion | Python

- Built an IMM motion bank, exact JPDA enumeration with bounded fallback, Wald track scoring, and OSPA decomposition for estimation, association, and track-existence analysis.
- Reduced localization error by 47% and raised IMM coverage from 0.893 to 0.960 across 67 tests; published the turn-rate regimes where the configured model bank loses.

HNSW Vector Search Engine | C++17, Python

- Implemented a layered HNSW graph, diversity-aware neighbor selection, SIMD distance kernels, persistence, Python bindings, and GIL-free concurrent search.
- Reached 1.83x FAISS queries per second at 0.999 recall; at 98.5% recall, performed 26x fewer distance computations than exact scan across 51 tests.

ENGINEERING PRACTICE

- Verification: fault injection, differential testing, linearizability, exact-recall baselines, paired seeds, and controlled performance comparisons.
- Delivery: Linux development, Git-based workflows, automated tests and GitHub Actions, Docker packaging, documentation, and reproducible commands.
- Target roles: systems engineering, software engineering, ML infrastructure, data engineering, database internals, signal processing, and reliability-focused development.