

AAROHI GANDHI

Seattle, WA | U.S. Citizen | (206) 756 5000 | aarohigandhi@hotmail.com
aarohigandhi.vercel.app | github.com/aarohigandhi | linkedin.com/in/aarohigandhi

EDUCATION

University of Washington, Seattle | Paul G. Allen School of Computer Science and Engineering

Expected June 2028

B.S. Computer Science | GPA 3.80 / 4.00

- Admitted to UW at age 15 through the Robinson Center Early Entrance Program, two years ahead of cohort. Dean's List, 4.0 quarter.
- Coursework: Data Structures and Algorithms, Databases and Concurrency, Object Oriented Design, Machine Learning, Statistical Inference, Applied Probability, Linear Algebra, Optimization.

PROJECTS

Zenith, Downlink Scheduling for Satellite Constellations

2026

Python | Java | SGP4 orbital mechanics | simulated annealing | branch and bound | github.com/aarohigandhi/zenith

- Recovered 83% of generated data, 12.94 TB against a greedy baseline's 69%, across 12 Earth observation satellites and 12 real ground station sites by designing a simulated annealing scheduler whose ejection moves escape the local optima single swap heuristics get stuck in.
- Built the orbital mechanics pipeline from scratch, SGP4 propagation from real TLEs through coordinate transforms to visibility window and link budget computation, validated to 7 nanometre worst case agreement against published AIAA vectors across 10,916 active satellites and 98,241 propagations.
- Proved the scheduler optimal rather than asserting it, with a branch and bound solver that closed six instances and found a 0.00% gap on every one, where greedy fell 21 to 31% short.

Redline, Launch Vehicle Fault Detection During Ascent

2026

Python | physics simulation | extended Kalman filter | CUSUM | fault injection | github.com/aarohigandhi/redline

- Cut false engine shutdowns from 130 to 6 while finding 2.3 times as many real faults, 70% against the flown redline scheme's 30%, by arbitrating four detectors that fail in different ways instead of trusting any one, scored over 904 simulated flights with zero false alarms on the healthy ones.
- Explained why the baseline fails and not only that it does: three transducers share one chamber tap, so a partly blocked port makes all three read low together and voting cannot help, and a common mode feed loss droops every engine at once, which a per engine monitor answers by shutting them down one after another.

Nanobook, Zero Allocation Limit Order Book and Matching Engine

2026

Java 21 | Nasdaq TotalView ITCH 5.0 | JMH | Epsilon GC | github.com/aarohigandhi/nanobook

- Parsed 268.7M real Nasdaq ITCH messages at 9.8M per second and rebuilt the book with zero unknown order references and zero crossed quotes, the check that proves the parser byte correct, since one wrong field offset would have orphaned millions of orders.
- Ran the full 8.25 GB session in 95 MB under a collector that never reclaims, where one 16 byte object per message would have needed 4.3 GB, at 300 ns median and 1.2 us p99 service latency under 200k orders per second with no GC pauses.

Drift, Bit Exact Arithmetic Attribution for Low Precision Training

2026

Java | Python | FP8 and FP4 numerics | github.com/aarohigandhi/drift

- Built an arithmetic lab that makes every decision inside a matmul an independent variable, number format, scaling, rounding, accumulator width and summation order, verified against exact rational arithmetic on all 65,536 FP8 value pairs; the cross check caught 154 signed zero errors in my own reference on its first run.
- Validated it against real FP8 hardware, reproducing a training divergence at steps 452, 494 and 553 against 446, 453 and 497 in simulation, and traced it to the backward matmuls by showing it appears only when both operands are cast, after two earlier tests had wrongly shown nothing.

Screenshot Brain, Semantic Search Over Screenshots

2026

TypeScript | Next.js | Postgres and pgvector | Claude API | embeddings

- Shipped a full stack app that makes any screenshot searchable in plain language, owning ingest, storage schema, auth and search UI end to end, with a labeled evaluation harness scoring top 3 retrieval accuracy so quality was measured rather than assumed.

Also: InjectEval, an open source harness measuring how often prompt injection works against agents with tool access. | Adaptive KV cache compression, a custom decode loop cutting peak GPU memory 40% at under 2% quality loss. | PhishGuard AI, led a 5 person team on an ML threat detection pipeline for the Microsoft Imagine Cup, breach rate cut 45%. | CareVision, real time multimodal vision, 2nd of 200+ teams at DubHacks 2024. | PillPal, iOS medication app shipped solo to the App Store. | MIT Beaver Works Summer Institute 2023, embedded systems and hardware security.

EXPERIENCE

Founder and Lead Engineer, CyberMinds

2022 to Present

Production platform on Linux | 5,000+ monthly active users across 12 countries | 50+ contributors

- Scaled from zero to 5,000+ monthly active users across 12 countries as sole architect of the entire system, request gateway through model inference layer, with no team, no funding and no prior codebase.
- Cut request failure rate from 30% to under 3% through repeated upstream outages by building a fault tolerant multi provider proxy with exponential backoff, per provider rate limiting, idempotent writes and automatic failover.
- Instrumented the platform with telemetry and threshold alerting so failures surfaced before users reported them, and cut dashboard load time over 50% by profiling the slow path instead of adding hardware.
- Made code review and an automated test suite a requirement on every commit for 50+ mostly non technical contributors, on a two week release cycle.

Founder, Blu Birds

2023 to Present

Technology access initiative for elderly communities | Seattle, WA

- Delivered technology training to 200+ elderly participants, publishing a white paper on senior usability barriers from structured interviews at retirement communities and rebuilding the curriculum around the failure modes they surfaced.

TECHNICAL SKILLS

Languages: Python, Java, SQL, TypeScript and JavaScript, Swift, R, OCaml, Bash.

Systems: Linux, concurrency and locking, memory management and allocation free design, performance profiling, latency measurement, JMH benchmarking, ACID transactions, Docker, Git, GitHub Actions, CI/CD, REST APIs, PostgreSQL.

ML and numerics: PyTorch, HuggingFace Transformers, floating point and low precision arithmetic, NumPy, pandas, scikit learn, embeddings and vector search, evaluation harness design.

Mathematics: Linear algebra, applied probability, statistical inference, optimization and linear programming, discrete math, combinatorics, orbital mechanics.