Aarnawa Koirala

817-630-4442 | aarnawakoirala@gmail.com | linkedin.com/in/aarnawa | github.com/aarnawa05

Education

The University of Texas at Austin May 2027
Bachelor of Science in Computer Science, Minor in Business *Austin, TX*
GPA: 3.94 / 4.00
Relevant coursework: Data Structures, Operating Systems, Computer Vision, Software Engineering

Technical Skills

Languages: Rust, Java, C, Python, JavaScript
Frameworks: Spring, React, Flask
Databases: PostgreSQL
Developer Tools: Git, Docker, AWS (EC2, Amplify), Postman
Libraries: Apache POI, Numpy, SQLAlchemy

Experience

Software Engineering Intern May 2026 - August 2026
Progress Rail, a Caterpillar Company *Southlake, TX*

- Improved performance by 30-50% in critical areas by taking advantage of specialized CPU instructions.
- Extended the existing simulation engine with new physics capabilities, scaling it across hundreds of vehicles and multiple train systems without measurable performance degradation.
- Learned and applied rail dynamics and physics domain concepts to validate model accuracy, ensuring simulation output matched real-world vehicle behavior.
- Developed in Rust within a production, safety-critical simulation codebase, ramping into an unfamiliar language and domain to ship working features.

Projects

PintOS | *C, Concurrency, Kernel code*

- Designed and implemented priority scheduling to ensure high-priority threads received CPU time, preventing starvation through priority donation.
- Developed system calls (e.g., file operations, process management) to enable user programs to interact with the kernel safely and efficiently.
- Refactored the timer subsystem to block sleeping threads rather than busy-wait, eliminating wasted CPU cycles during sleep intervals.
- Debugged concurrency issues by analyzing race conditions and implementing robust synchronization techniques with semaphores and locks.

Memory Allocator | *C*

- Implemented a custom malloc/free using explicit free lists, coalescing, and 16-byte alignment for efficient memory management.
- Optimized heap allocation and fragmentation handling, improving memory utilization and allocation speed.
- Designed a heap consistency checker to detect memory corruption, alignment issues, and invalid free operations.
- Reached an average utilization of 85% and throughput of 1750 requests per second.

EducaNation - Full-Stack Website | *React, Flask, Docker*

- Built and deployed a full-stack web application serving 50k+ relational records across multiple entities, supporting filtered search, joins, and pagination with <300 ms average API response times.
- Designed and implemented RESTful API endpoints with server-side pagination and query filtering, reducing payload sizes by ∼70% and improving frontend load times and scalability, using Postman for validation.
- Collaborated in a 5 person team, contributing features through Git-based workflows, peer code reviews, and iterative releases while maintaining 90%+ backend test coverage, using GitLab, PyTest, and CI-style testing practices.