AADARSH CHANDRA (Pi)

India | github.com/pie-314 | pidev.tech | LinkedIn | aadarsh.official.xz@gmail.com

SUMMARY

Software engineer with strong foundations in backend systems, scalable architecture, and performance-oriented design. Experience building storage engines, async data pipelines, developer tooling, and web backends using Rust, C, and Python.

EDUCATION

Indian Institute of Technology Madras Ongoing

BS in Data Science

Relevant Coursework: Data Structures, Algorithms, Operating Systems, Database Systems, Distributed Systems

TECHNICAL SKILLS

- **Languages:** Rust, C, Python, SQL, TypeScript, JavaScript
- **Backend & APIs:** REST API Design, Flask, Django, Async Services (Tokio), Authentication Flows, JSON/HTTP Handling
- **Systems & Concurrency:** Async Programming, Multithreading Concepts, File I/O, Write-Ahead Logging (WAL), Memory Management, Producer–Consumer Patterns
- **Databases:** SQLite, PostgreSQL, Supabase, Schema Design, Indexing, Query Optimization
- **Web & Frontend:** React, Next.js, Component-Based UI Architecture, State Management Basics
- **DevOps & Tooling:** Git, Docker, Linux (Arch/Hyprland), CLI/TUI Development, Benchmarking, Performance Profiling
- **Other Technologies:** Aptos Move (Smart Contracts), Firebase (Storage/Firestore), Basic Cloud Deployment

SELECTED PROJECTS

RadishDB (Persistent Key-Value Store) | *C* | GitHub

- Built a Redis-inspired persistent key-value store (2K+ LOC) with hash-indexed in-memory structures and append-only disk storage.
- Implemented Write-Ahead Logging (WAL) and crash-recovery mechanism to ensure durability and atomic writes.
- Benchmarked 100K+ key recovery in ¡60ms and sustained 8K+ operations/sec in local stress tests.
- Developed automated replay and integrity validation tests to verify recovery correctness.

Rustorian (Async Telemetry Backend) | *Rust, Tokio, SQLite* | GitHub

- Designed asynchronous ingestion service using bounded channels and producer-consumer pipelines.
- Sustained 2K+ records/sec throughput with sub-millisecond in-memory read latency.
- Implemented structured persistence and periodic data rotation for scalable long-term storage.
- Focused on modular separation between ingestion, storage, and query layers.

TRX (Terminal Package Explorer) | *Rust*

- Built a modular TUI integrating pacman and AUR backends for Arch Linux.
- Designed responsive async execution layer and state-driven rendering pipeline.
- Improved CLI usability through ergonomic navigation and structured command output.

Neural Network from Scratch | *C*

- Implemented a single-layer perceptron with custom gradient descent and manual weight updates.
- Built forward and backward propagation logic without external ML libraries.
- Optimized contiguous memory layout for efficient parameter updates and experimentation.

PUBLICATIONS

Author of engineering deep-dives at pidev.tech/blogs

Featured article: “How I Built a Crash-Safe Database Engine in C”

INTERESTS

Scalable backend architecture, systems design, distributed systems, developer tooling, and technical writing.