Aarjav Patni

☎ +1 437 484 2405 | ✉ aarjav.patni@uwaterloo.ca | in aarjavpatni | aarjavpatni | Website

Education

University of Waterloo Waterloo, ON
Bachelor of Software Engineering *Sep 2023 - Present*

Skills

Languages: C, C++, Python, Rust, Go, TypeScript, JavaScript, Elixir, Java, VHDL, Assembly, HTML, CSS
Technologies: Kubernetes, Terraform, Docker, React, React Native, NextJS, Django, PostgreSQL, AWS, GCP, Pandas

Experience

Voltra

Waterloo, ON
Software Engineer *Jan 2025 – Apr 2025*

- Architected and maintained modular **Terraform**-based **Kubernetes** clusters on AWS and GCP for the microservices architecture of chargepoint software
- Configured **ArgoCD**-driven GitOps pipelines and Helm charts to automate cluster and microservice deployments
- Implemented self-hosted distributed tracing using **SigNoz** and network-level observability with **Cilium Hubble**
- Developed an **ETL** microservice in **Elixir** using **AWS S3** and **DuckDB** to ingest, aggregate and normalize electrical grid data across multiple regions and data formats

WATonomous

Waterloo, ON
Software Engineer – Server Cluster Division *Aug 2024 – Dec 2024*

- Developed WATcloud CLI – a **Rust**-based **Linux** shell featuring easy access to info such as compute cluster status, daemon process status, and per-user usage quota for the cluster machines
- Enhanced system stability by resolving API integration issues in error reporting systems (**Prometheus**, **Sentry**)

Readwise

Toronto, ON
Software Engineer *May 2024 – Aug 2024*

- Developed an internal **Django** API to streamline feedback ticket triaging, handling **300+** daily requests, and integrated it with Slack for daily customer experience (CX) statistics
- Engineered a **TypeScript**-based Raycast tool to interact with the API, empowering the CX team to create task issues efficiently and save **50+** hours per month
- Implemented a document search feature in **React** for the **Rust** desktop app to overcome Safari Webkit limitations
- Refactored the Rust-based desktop app's auto-update mechanism to interoperate safely with the TypeScript frontend, preventing race conditions and update failures.
- Increased PDF to HTML conversion accuracy by **75%** using Diffbot parser APIs, greatly enhancing textbook and research paper readability

Projects

Distributed Filesystem Simulator | *GitHub*

- Developed a **Rust**-based distributed file store simulator with chunking, SHA-256 hashing, and LZ4 compression
- Implemented in-memory node replication and XOR-parity recovery to mimic fault-tolerant storage
- Built a `clap` CLI and `tokio` async backend for put/get operations, node management, and failure simulation
- Benchmarked and documented throughput and storage overhead with `hyperfine` and `flamegraph` analysis tools

Orderbook Proxy API Server | *GitHub*

- Developed a proxy API server in **Rust** to speed up the processing of sample trade data by **64%**
- Implemented a least recently used (LRU) caching strategy to reduce API calls by **83.6%**
- Engineered additional logging functionality to monitor query performance and track system metrics
- Documented the project's performance benchmarks, design decisions, and tradeoffs

Hostel Management System | *GitHub*

- Developed a full-stack web application to digitize and streamline student tenant management
- Built the frontend using **NextJS** and **Supabase PostgreSQL** for the backend
- Implemented an admin interface to view, add, and update student and billing records using the Supabase API
- Automated student billing process by integrating logic to dynamically calculate rent and utilities costs based on past monthly billing data, and generate invoices accordingly