

# Aryan Yadav

Blog: [www.aryanyadav.dev](http://www.aryanyadav.dev) | E-mail: [contact@aryanyadav.dev](mailto:contact@aryanyadav.dev) | Github: [Aryan77](https://github.com/Aryan77) | LinkedIn: [aryanyadav4](https://www.linkedin.com/in/aryanyadav4) | (607)342-1284 | Ithaca, NY

## EDUCATION

**Cornell University:** BS in Computer Science - Dec 2023, Masters in Computer Science - May 2024

## SKILLS

**Languages/Tools :** C++, C, Python, OCaml, Java, C#, AWS, SQL, Kafka, JavaScript, React, Angular

**Specialization:** HFT Systems, Low Latency Programming, Kernel Programming, Backend Development, Distributed Systems

## EXPERIENCE

**Millennium Management | Software Engineer Intern - Low Latency C++**

**Miami, FL | Jun 2023 – Aug 2023**

- Designed and implemented advanced lock-free MPMC and SPMC queue algorithms, leveraging atomic operations, memory barriers, cache locality, zero-copy deserialization, branch prediction to achieve ultra-high throughput and minimal latency.
- Designed a component test suite to validate daily startup messages between the order management system and clients, ensuring accurate exchanges.
- Developed robust automation scripts to streamline the component test environment setup, which were pivotal in rapidly provisioning remote servers with no prior configurations, ensuring a consistent and optimized environment for application testing.
- Conducted comprehensive performance benchmarking to evaluate the efficiency of the Aeron library against raw UDP and IPC. This involved creating various real-world scenarios to measure throughput, latency, and resource utilization, providing valuable insights to guide future architectural decisions.
- Collaborated closely with the Crossing Engine team, diagnosing and resolving intricate issues related to munmapping which was traced to a bug in specific versions of the Linux kernel.

**Bermen March | Software Engineer Intern**

**Brooklyn, NY | Jun 2022 – Aug 2022**

- Built a Windows service using .NET Core to fully automate a risk assessment Excel workflow for a typical hedge fund.
- Implemented the service as a background process using Hangfire, built a build-and-deploy pipeline in Azure DevOps to deploy the service across multiple Amazon EC2 instances.
- Implemented a Kafka consumer as part of the service to obtain real-time market data from multiple Kafka topics, enabling simultaneous data ingestion from diverse sources.
- Designed and implemented robust data categorization and normalization processes to save data to Postgres database, ensuring consistency and data integrity for downstream applications.
- Built a multi-page Angular frontend with an ASP.NET backend to allow users to select a subset of data and display relevant charts and tables; the tables implemented lazy-loading and virtual scrolling and allowed for advanced querying and filtering without refetching data.

**UniPantry | Software Engineer Intern**

**Ithaca, NY | Feb 2022 – Jun 2022**

- Developed an app with a TikTok-like UI in React Native, hosted on Firebase and using GCS for video data storage.
- Implemented machine-learning based user-optimized search and recommendations using Algolia Search and Recommend.
- Reduced database latency by over 30% by minimizing indexes, limiting client write rates, using async calls whenever possible, and refactoring database structure to be more denormalized.
- Designed a build pipeline for building testable versions and deployable binaries for both Android and iOS.
- Integrated Google Analytics to capture usage statistics for every user action, displayed using an interactive React UI using D3.

**AI Learners | Software Engineer Intern**

**Ithaca, NY | Dec 2021 – Feb 2022**

- Implemented major refactoring of our game engine to enable automatic generation of React components and PIXIJS animations from JSON representation of game data.
- Created robust and efficient APIs using Node.js to handle game data, user profiles, and score tracking, ensuring seamless gameplay experiences.
- Integrated Google Firestore as the primary database, optimizing queries to provide real-time data access, enhancing game state persistence and user progression tracking.

## PROJECTS

**[Distributed Database \(Go\)](#)** - A distributed database system similar to Amazon's DynamoDB or Google's BigTable that is persistent, fault tolerant and sharded, achieving consensus using my implementation of the Raft protocol and implementing sharding using the shard controller pattern. I use a custom RPC library to simulate node failure to test correctness.

**[Minimal Browser \(Python, Tkinter\)](#)** - A minimal browser engine and GUI implemented from scratch; is able to connect to web servers and properly download relevant information for displaying web pages and use my own HTML and CSS parser to provide a proper layout on a minimal tkinter window. Also implements tabbed browsing and hyperlink-based navigation for better UI.

**[Ten-Tac-Toe \(JQuery, Express.js, Google Cloud Run\)](#)** - A web app that allows users to play a deeply strategic and computationally unsolved extension of Tic-Tac-Toe. Users can play against a very strong AI built using minimax and alpha-beta pruning, or against other players online, connected in a 'room' that communicates state using web-sockets. The Express backend is hosted as a serverless platform on Google Cloud Run, allowing for arbitrary scaling.

**[OCaml Data Science Library \(OCaml, Jupyter Notebook\)](#)** - Single library that provides end-to-end functionality for a machine learning workflow in OCaml - from data importing and cleaning to exporting the final model. I provide data manipulation functions similar to Numpy and Pandas, along with machine learning algorithms to achieve feature parity with Sklearn.

**[MapReduce \(C++, Bash\)](#)** - A library that implements the MapReduce pattern, along with an example application using the library to present the most frequent n-grams from a massive number of text files after processing them in parallel; the worker threads maintaining their own local maps removes the locking overhead and makes it extremely performant.