

AMMAR ATEYA

[✉ ammarat@umich.edu](mailto:ammarat@umich.edu) | [in linkedin.com/in/ammarateya](https://linkedin.com/in/ammarateya) | [GitHub github.com/ammarateya](https://github.com/ammarateya)

EDUCATION

University of Michigan - Ann Arbor

B.S. in Computer Science and Linguistics, Minor in Mathematics

Expected Graduation: Apr 2027

Ann Arbor, MI

- **Relevant Coursework:** Data Structures and Algorithms, Advanced Operating Systems, Distributed Systems, Web Systems, Machine Learning, Computer Organization, Discrete Mathematics, Linear Algebra, Linux Programming
- **Activities and Societies:** V1 Entrepreneurship, Michigan Hackers, Linguistics Club
- **Awards/Programs:** National Merit Scholar, Michigan Seal of Biliteracy in Spanish, 2x Hackathon Winner

TECHNICAL SKILLS

Languages and Frameworks: C++/C, Python, JavaScript, TypeScript, Go/Golang, HTML/CSS, React.js, Next.js, Ruby

Tools and Technologies: Git, AWS, Bash/Linux, SQL, NoSQL, REST APIs, RPCs, PyTorch, TensorFlow, Docker/Kubernetes

EXPERIENCE

GitHub

May 2025 – Aug 2025

Software Engineer Intern

San Francisco, CA

- Spearheaded React rewrite of the Pull Requests Files Changed interface, slashing CPU usage by 12% per request.
- Identified and remediated critical accessibility issues across high-traffic React and Ruby on Rails pages, improving UI/UX for users relying on assistive technology, at least 7% of GitHub users, or 10.5 million active users.
- Led Vitest migration of React packages; identified and removed obsolete code to cut bundle size by 5%.

0xKnowledge

Jun 2024 – Aug 2024

Software Engineering Intern

Ann Arbor, MI

- Developed a full stack Notion-style knowledge base editor in Next.js for clients to manage documentation.
- Managed end-to-end cloud deployment on Vercel, automating the build process with CI/CD pipelines to streamline continuous integration, updates, and minimize downtime.
- Designed and implemented SQL schema, integrated with Supabase to handle real-time data and user authentication, improving query performance by 25% and ensuring efficient data management.

PROJECTS

Thread Library | C++, Multi-threading, Mutexes, Condition Variables, Semaphores, Unix

- Implemented a kernel C++ thread library on Unix, handling CPU booting, thread management, management of 50+ CPUs, interrupts, atomicity, and FIFO scheduling order.
- Designed spin-locks, mutexes, conditional variables utilizing advanced Unix context management.
- Enforced comprehensive unit tests to assure the library's soundness, identify and rectify errors, and guarantee optimal performance in diverse scenarios.

Sharded Fault-Tolerant Key/Value Service | Go, Paxos, RPC, Concurrency, Distributed Systems

- Implemented a distributed key/value storage system that partitions keys across replica groups ("shards"), with each group internally replicated via Paxos to tolerate server and network failures.
- Built a fault-tolerant shard master service (also replicated with Paxos) to manage reconfiguration and assign shards to replica groups, ensuring single-copy semantics and linearizable Get/Put/Append operations across reassessments.

Ann Arbor Parking Citations Visualizer | React, Python, Postgres, REST API, CI/CD

- Scaled to 3,000 users within the first week of deployment; built a high-performance interactive map using Leaflet.js and Canvas rendering to visualize 10,000+ data points with a responsive, Google Maps-inspired UI.
- Engineered a self-healing data pipeline via GitHub Actions to continuously scour citations; implemented optical character recognition (OCR) and fuzzy geocoding algorithms to enrich raw records with precise coordinates.
- Built a real-time subscription engine alerting users of new tickets via email and webhooks; developed comprehensive analytics dashboards for officer performance tracking while ensuring privacy through server-side image redaction.

PeteCode: LeetCode Solution Platform | React, FastAPI, Supabase, GroqCloud, ElevenLabs API, TypeScript

- Developed a full stack AI-powered platform using React and FastAPI to provide personalized, engaging coaching for LeetCode problems; integrated API responses from GroqCloud (LLM 3.2 70b model) and ElevenLabs APIs.

Computer Vision Fighting Game | Python, Pygame, OpenCV, MediaPipe

- Built a desktop game, leveraging OpenCV and MediaPipe to recognize real-time gestures via webcam, and map them to in-game actions such as healing and attacking; won an award at the annual MSU hackathon.
- Designed AI opponents with state-driven logic, allowing for single-player mode, enhancing user interactivity.