Rehan Shah

+1 732-890-3454 | rehanfshah@gmail.com | linkedin.com/in/rehanshah17 | github.com/rehanshah17

Education

University of Michigan, Ann Arbor Ann Arbor, MI

B.S. in Computer Science, Expected May 2027 *GPA: 3.78*

- **Relevant Coursework:** Data Structures and Algorithms, Object-Oriented Programming, Computer Architecture, Theory of Computability, Discrete Mathematics, Distributed Systems, Web Systems, Open Source Development
- **Involvements:** Alpha Kappa Psi, Michigan Community Scholars Program, Michigan Student AI Lab

Experience

Software Engineer May 2025 – Aug 2025

Sample Scout *Harlem, NY*

- Built large-scale audio data ingestion pipelines using Python, Pandas, and BeautifulSoup to process 60,000+ audio files into training-ready datasets with schema validation and repeatable data quality checks
- Generated similarity search embeddings using CLAP and MERT models with Pinecone vector indexing, improving audio retrieval relevance by 25% through optimized feature extraction and nearest-neighbor search
- Improved classification accuracy by training a second-stage logistic regression model that combined Siamese network similarity scores with other acoustic similarity features like chroma and MFCC to refine predictions
- Designed and deployed an API via FastAPI which handled song specific metadata and reranking using Python and Uvicorn, exposed endpoints for embedding lookup and search results that handled 200,000+ requests

Software Engineering Intern May 2024 – Aug 2024

Siftd.ai *San Francisco, CA*

- Captured browser navigation and user interaction events using TypeScript content scripts, message passing, and IndexedDB (Dexie.js), collecting and storing hundreds of structured workflow records per browsing session
- Integrated Helicone into Next.js/TypeScript API routes and added Zustand state management in React to track tool calls, enhancing observability for backend engineers and reducing concurrency errors across sessions
- Implemented a Go request interceptor to prevent sensitive fields like API keys, auth tokens from being logged or indexed by sanitizing inputs, truncating user-provided values, and parsing DOM snapshots within logging

Projects

Soteric | Open Source *Python, Rust*

- Building an open-source, process based file protection system that dynamically encrypts specified files to prevent unauthorized access by AI tools and automated processes, enforcing application-specific access policies at runtime
- Developed a macOS proof-of-concept using macFUSE and Python to intercept filesystem read operations, inspect calling processes via PID tracking, and implement a conditional encryption layer that selectively scrambles protected files for blacklisted applications while preserving seamless plaintext access for trusted workflows
- Designed a configurable profile engine supporting application level protection rules and began re-architecting the system in Rust to deliver a fast and native implementation with cross-platform support

subcheck *Go, Docker, Linux, Makefiles, CLI Tooling*

- Built a developer CLI tool in Go that reproduces the University of Michigan CAEN autograder environment locally using Docker, enabling students to validate builds and runtime behavior before submission
- Implemented environment parity checks for compiler flags, Makefile targets, and runtime dependencies, catching submission-breaking issues such as missing flags, toolchain mismatches, and environment-specific failures
- Integrated automated build, execution, and Valgrind memory-check workflows into a single command interface, allowing students to compile, run, and debug programs locally with the same tooling used by the autograders

WebDJController *Javascript*

- Built a web-based DJ controller with a dual-deck mixing UI, crossfader, looping, and transport controls, supporting both Spotify playback and local audio uploads
- Implemented Spotify OAuth and a Node.js/Express API proxy to keep client secrets server-side, exchange auth codes for tokens, and forward authenticated Spotify Web API requests with rate-limit handling
- Added DJ-assist features such as BPM sync via playback-rate adjustment for local files and automated crossfade transitions using an equal-power curve

Technical Skills

Languages: GoLang, Python, JavaScript, TypeScript, C/C++, Java, SQL
Machine Learning & Data: PyTorch, scikit-learn, Pandas, CLAP, MERT, Pinecone
Systems & Tools: Linux, Docker, PostgreSQL, Elasticsearch, FastAPI, Node.js, Express, React, Git, GitHub Actions