Jonas Groening

jonasg@umich.edu | jonasiwnl.github.io | linkedin.com/in/jonasgroening | github.com/jonasiwnl

Education

University of Michigan Ann Arbor, MI

B.S.E, Computer Science *Graduating May 2026*

GPA: 4.00/4.00 | **Activities**: V1, UM Autonomous Robotic Vehicle
Coursework: Data Structures & Algorithms, Data Driven Systems, Theory of Computation, Discrete Math, Linear Algebra
Fellowships: SIG Discovery Day - Technology

Skills

Languages: Python, C++, Go, Typescript, Rust
Technologies: Git, Linux, Docker, Django, Flask, NextJS, MySQL, MongoDB, Terraform, Postman, Makefile
Interests: Keyboards, Soccer, Personal Finance, Hiking, Video Games

Experience

Vectra AI May 2024 – Present

Software Engineer Intern *Austin, TX*

- Leverage Terraform to orchestrate AWS S3 Access Point integrations and visualize time-series bucket usage data through Grafana, providing process-level cost observability and identifying areas for expense reduction.
- Pilot scalable event-driven architecture for high workload tasks with Python and Celery, improving fault tolerance, eliminating intermediate cron jobs and database tables, slashing latency by up to 80%.

CriTech Research May 2023 – August 2023

Software Engineer Intern *Saline, MI*

- Shipped redesigned endpoints for a medical patient portal using Python and Flask, removing unnecessary MySQL queries and accommodating a 10% growth in compliance data requests.
- Optimized API reliability by adding 100% coverage tests (unit, integration, blackbox) to a CI pipeline, saving ∼2 hours of manual testing weekly and providing high service availability.
- Collaborated with Senior Engineers to migrate backend infrastructure to Azure App Services, Blob Storage, and MySQL Database using Terraform, cutting hosting costs significantly and improving app uptime to 99.999%.

UM Autonomous Robotic Vehicle September 2022 – May 2023

Software Engineer - Sensors *Ann Arbor, MI*

- Deployed temporal, jitter, and transformative filters for an IMU sensor in C++ and Python to clean inputs for a SLAM (simultaneous location and mapping) algorithm, reducing noisy data by 40%.
- Implemented a robust Python logging system using Pub/Sub architecture to monitor robot metrics in real-time and alert engineers of potential errors, resulting in a diagnosis of malfunctioning sensors.
- Led architectural design reviews and communicated decisions across teams to ensure system reliability.

Projects

quarry.video | *NextJS, Python, Django, Go, Terraform, MongoDB* | Visit

- Architected a full-stack app with NextJS, MongoDB, and Prisma, providing a robust in-browser interface for short-form content generation, video editing, and data visualization.
- Leveraged Python and Django to build a reliable FFmpeg wrapper for the video processing pipeline.
- Authored and deployed a centralized logging service using Go, allowing the team to find and track anomalies.
- Automated deployment through a CI/CD pipeline built with Terraform and GitHub Actions, reducing manual testing and allowing features to reach production quicker.

Beehive | *C++, FFmpeg, Multithreading* | GitHub

- Engineered a cross-platform screen recording and streaming tool using C++, leveraging FFmpeg to encode and push video to an RTP server in real-time (60+ frames per second) or write to disk in multiple formats.

Embedchain | *Python, ChromaDB, PyTest* | GitHub

- Contributed Python features to create and reset multiple AI “brains” using ChromaDB collections.
- Iteratively improved code quality through open source maintainer feedback. Integrated extensive unit and end-to-end tests with PyTest for each feature.