

# Jonas Groening

[jonasg@umich.edu](mailto:jonasg@umich.edu) | [jonasiwnl.github.io](https://github.com/jonasiwnl) | [linkedin.com/in/jonasgroening](https://linkedin.com/in/jonasgroening) | [github.com/jonasiwnl](https://github.com/jonasiwnl)

## EDUCATION

### University of Michigan

B.S.E, Computer Science

Ann Arbor, MI

Graduating May 2026

**GPA:** 4.00 | **Activities:** UM Autonomous Robotic Vehicle, V1 @ Michigan, Michigan Hackers

**Coursework:** Data Structures & Algorithms, Advanced Algorithms, Database Management Systems, Computer Architecture, Foundations of Computer Science

## TECHNICAL SKILLS

**Languages:** Python, Go, C++, Javascript, Typescript, C#

**Technologies:** Git, Linux, Docker, Django, .NET, Flask, NextJS, Terraform, Postman, MySQL, MongoDB, Makefile

## EXPERIENCE

### Vectra AI

Incoming Software Engineer Intern

May 2024 – August 2024

Austin, TX

### UM Autonomous Robotic Vehicle

Software Engineer

August 2023 – May 2024

Ann Arbor, MI

- Developed a **ROS2** node in **Python** to accurately read, process, and publish IMU sensor data, improving robot localization and orientation precision.
- Deployed temporal and jitter filters for the IMU sensor in **C++** and **Python** to clean inputs for a SLAM (simultaneous location and mapping) algorithm, reducing noisy data by 50%.
- Implemented a robust logging system using Pub/Sub architecture to monitor robot metrics in real-time and alert the team of potential errors.

### CriTech Research

Software Engineer Intern

May 2023 – August 2023

Saline, MI

- Shipped 3 modified endpoints for a medical patient portal using **C#** and **.NET**, reducing unnecessary **MySQL** queries and accommodating a 12% growth in interactions.
- Engineered a **Python** API using the **Flask** framework to receive and process x-ray scans, creating an interactive 3-dimensional point cloud that enables medical professionals to analyze bone anomalies.
- Optimized API reliability by adding 100% coverage tests (unit, integration, blackbox) to a CI pipeline, saving ~2 hours of manual testing weekly.

## PROJECTS

### Quarry.video

January 2023 – May 2023

- Architected a full-stack tool with **Next.js**, providing a robust interface for in-browser video editing, data visualization, and user authentication.
- Pioneered a custom ORM adapter for **MongoDB** that improves session data retrieval times by over 5 seconds.
- Authored and deployed a centralized API logging service with **Go** and **Typescript**, allowing the team to find anomalies more frequently.
- Automated deployment with a CI/CD pipeline built with **Terraform**, **Docker**, and GitHub Actions, reducing manual testing and allowing features to see production quicker.

### Embedchain | Open Source Contributor

July 2023 – August 2023

- Shipped a **Python** feature to create multiple AI “brains” using **ChromaDB** collections, resolving 3 github issues.
- Integrated extensive unit and end-to-end tests with **PyTest** increasing coverage by 15%.
- Improved API ergonomics by writing a comprehensive database reset function.
- Extended app functionality using **TypeScript** by adding runtime flags for custom model and database options.

### Zipnotes

October 2023 – December 2023

- Engineered a locally stored notes app in **Rust** through the **Yew** framework, using **Nginx** as a static web server.
- Wrote a comprehensive **Dockerfile** to containerize the application, allowing for easy deployment. Files are built with a Rust image, then copied into a slimmer Nginx image for minimal image size.