

Gabriel Hassan

Ann Arbor, MI | gjhassan@umich.edu
linkedin.com/in/gabehassan | github.com/gabehassan | gabehassan.com

EDUCATION

University of Michigan

Bachelor of Science in Computer Science

Ann Arbor, MI

Expected May 2028

- Coursework: Data Structures and Algorithms (EECS 281), Discrete Math, Programming and Data Structures

Henry Ford College

Computer Science coursework

Dearborn, MI

Transferred Jan 2025

- Dean's List; Henry Ford II Honors Program; Programming Club President, Vice President prior year
- Community Service Club: volunteered at a retirement home supporting residents with memory loss, organizing events and regular game nights

EXPERIENCE

Owner and Operator

2019 – Present

Independent Resale Business

Detroit, MI

- Sold \$70,000+ in footwear and computer hardware across StockX, GOAT, Grailed, and private marketplaces, handling sourcing, pricing, and fulfillment
- Built a Node.js orchestrator that launched parallel automation tasks from JSON profiles, cutting setup for a new product release to seconds
- Reverse engineered an undocumented retail cart API from captured browser traffic, adding items to cart the instant a release went live

Co-Founder

2021

JustMonitors

Remote

- Wrote the Node.js Twitter monitor behind a paid alerts product from scratch, spanning OAuth 1.0a request signing to Discord alert delivery on AWS EC2
- Shipped a Discord bot for staff to retune tracked accounts and poll interval at runtime, commands gated by role
- Grew to 80 monthly active users and 6 paying organizations, led a support team of three, and wrote the docs

PROJECTS

Chrome Hearts Monitor | *JavaScript, Cloudflare Workers, Durable Objects*

2026

- Built a serverless Cloudflare Workers monitor that watches a luxury retailer's store and posts a Discord alert seconds after a new product goes live
- Cut detection latency 4x below Cloudflare's 60 second scheduling floor by driving the poll from a Durable Object alarm that reschedules itself every 15 seconds
- Scanned all 200+ product categories per check by splitting work across parallel worker invocations, with 83 unit tests over detection and alerting

Watchlist Match | *JavaScript, Vercel Serverless Functions* | watchlistmatch.com

2026

- Built a web tool that picks the movie for a group by comparing everyone's Letterboxd watchlists and sorting the overlap by rating
- Scraped paginated watchlist pages through serverless functions with caching and rate limiting, since Letterboxd has no public API and CORS blocks browser requests

table42 / noresi | *Go, Reverse Engineering, HTTP/2*

2026

- Built Go clients that book Resy and OpenTable reservations through undocumented APIs: availability polling, slot parsing, token exchange, and booking
- Reverse engineered OpenTable's private mobile API by decompiling its Android app, recovering the OAuth client setup, and mapping endpoints with mitmproxy
- Reproduced Chrome's TLS, HTTP/2, and header fingerprints from live browser captures to match real browser traffic at the protocol level
- Benchmarked against open source booking tools: 200ms end to end versus about 2.5 seconds

TECHNICAL SKILLS

Languages: JavaScript, Python, Go, C++, HTML/CSS

Platforms: Node.js, Cloudflare Workers, Durable Objects, Cloudflare KV, Vercel, AWS EC2, Linux

Practices: Git, REST APIs, OAuth, mitmproxy, Unit Testing, AI coding tools (Claude Code, Codex)