Nathan Monroy

monroynathan31@gmail.com | linkedin.com/in/monroyn | github.com/NM711

Summary

Software developer with experience building full-stack applications, automation tooling, and systems-level projects using **TypeScript**, **Node.js**, **C++**, and **SQL**. Comfortable working in **Git**-based collaborative workflows, debugging unfamiliar codebases, and translating business requirements into shipped features. Interested in backend systems, data platforms, and scalable developer tooling.

Skills

Languages:	TypeScript, JavaScript, C++, Java, SQL
Frontend:	React, SvelteKit, Tailwind CSS, HTML/CSS
Backend & Data:	Node.js, PostgreSQL, MySQL, SQLite, MongoDB
Developer Tools:	Git, GitHub, Docker, Bash, SSH

Experience

CSIS Tutor | Grossmont College | *El Cajon, CA* October 2025 – Present

- Tutor students across the CSIS curriculum, including **CompTIA A+** and **Security+** coursework, intermediate **Linux** administration, **Microsoft Server**, and programming courses in **C++**, **Java**, **Python**, **JavaScript** and more
- Debug student programming assignments and walk through **Linux** and **Windows Server** configuration and troubleshooting, explaining the underlying concepts so students can solve similar problems on their own
- Adapt explanations and troubleshooting approaches to different experience levels and learning styles
- Diagnose issues with limited context by asking targeted clarifying questions and reasoning through unfamiliar code, systems, and configurations

Automation Software Developer Intern | ULimo | *Remote, US* September 2025 – November 2025

- Developed **TypeScript**/**Node.js** automation tooling to streamline operational workflows and improve scalability of social media management systems
- Built **Discord** bots with **discord.js** to automate HR onboarding tasks such as role assignment, welcome flows, reminders, and request intake, reducing manual work for the HR team
- Collaborated with HR and operations stakeholders to gather requirements and translate them into onboarding automation features
- Participated in iterative development and weekly review meetings to refine feature priorities and improve tooling reliability
- Used **Git**-based workflows to implement, test, and ship production-facing automation updates

Full Stack Developer | Midtown Moto Garage | *San Diego, CA* January 2025 – July 2025

- Designed and deployed a full-stack platform with **SvelteKit**, **Node.js**, and **MySQL**, enabling small businesses to manage inventory listings and customer-facing service information
- Built an administrative CMS and responsive UI components in **SvelteKit** and **Tailwind CSS**
- Containerized and deployed multiple application services with **Docker** across VM environments, managing service dependencies and configuration
- Integrated headless **Shopify** to let the business sell products sourced from its data providers
- Built **Node.js** automation tooling to sync up-to-date provider data with the store database
- Integrated **Umami** analytics to give the business visibility into site traffic and support data-driven decisions
- Translated business requirements and design mockups into production-ready features

Projects

MIPS I Emulator | *C++, Assembly, ELF* github.com/NM711/MIPS-Emulator

- Implemented a **MIPS I** CPU emulator in **C++** covering arithmetic, logical, memory, and control-flow instructions, including **HI/LO registers** and **syscalls**
- Built an **ELF** loader that parses **ELF32** headers and program segments and maps executables directly into emulated memory
- Developed a disassembler to analyze instruction execution and verify correctness

DHT22 Sensor Protocol | *C++, Arduino, ESP32* github.com/NM711/DHT22Arduino

- Implemented a **C++** driver library for the **DHT22/AM2303** sensor on an **ESP32**, performing the single-wire handshake by driving and releasing a shared data pin with a pull-up resistor
- Decoded the sensor's timing-based bitstream into temperature and humidity readings using bit manipulation and microsecond pulse-width measurements to distinguish 0 and 1 bits
- Validated each transmission with a checksum to reject corrupted reads, as with **TCP** packet validation

SimpleDoc Markup Language | *Parsing, Language Design, Nim* github.com/NM711/SimpleDocParser

- Designed and implemented a lightweight markup language with a custom lexer and parser pipeline in **Nim**
- Built structured document transformation tooling for programmatic content generation
- Focused on maintainable parsing architecture, extensibility, and predictable structured output

Education

A.S. Computer Science | Grossmont College Expected Fall 2027
Transferring to pursue a Bachelor's degree in Computer Science