

Ryleigh Shikhanova

607-699-1626 | ryleigh.shikhanova@gmail.com | <https://www.linkedin.com/in/ryleigh-shikhanova/> | <https://github.com/ryleigh-shikhanova>

EDUCATION

East Stroudsburg University

Bachelor of Science in Computer Science, Minor in Mathematics

East Stroudsburg, PA

Aug 2022 – May 2027

EXPERIENCE

Fulfillment Associate

Amazon

May 2023 – Present

Pittston, PA

- Investigate inventory and workflow exceptions, identify root causes, correct system or physical discrepancies, and document resolutions using internal tools.
- Work 30–40 hours per week on an overnight schedule while completing a B.S. in Computer Science.

PROJECTS

Intel 8080 Emulator | C++, CMake, Catch2, Git

May 2026 – Present

- Developing an Intel 8080 emulator implementing instruction decoding and execution across the processor's instruction set, including undocumented opcodes
- Modeled CPU state, registers, flags, memory access, and register-pair operations using modular C++ components
- Implemented instruction cycle accounting and opcode metadata to reproduce Intel 8080 timing behavior
- Built automated Catch2 test suites covering arithmetic, data-transfer, flag, and CPU-state behavior
- Configured a portable CMake build and development workflow for compilation and automated testing

Intel 8080 Hardware Test Harness | C/C++, AVR Assembly, Arduino, Digital Logic

Sept 2026 – Present

- Developing a breadboard-based Intel 8080 test system designed to generate authentic CPU execution traces for validating a cycle-aware 8080 emulator
- Developed an AVR assembly power sequencer to control the 8080's required +5 V, -5 V, and +12 V rails, including voltage validation, startup/shutdown ordering, fault handling, and relay control
- Integrating SRAM, Intel 8224/8228 support logic, bus transceivers, and microcontrollers for program injection and processor-state capture
- Designing a pipeline to convert physical CPU traces into automated Catch2 emulator tests

Unix Shell | C, Linux

Sept 2026 – Oct 2026

- Developed a Unix-style command shell in C with command parsing, built-in commands, and executable resolution through the system PATH
- Created and managed child processes using `fork()`, `execve()`, and `waitpid()`, including argument and environment propagation
- Implemented shell built-ins for directory navigation, environment management, process signaling, executable lookup, and directory listing
- Used Linux/POSIX system interfaces including `getdents`, `kill`, `access`, `chdir`, and environment APIs with syscall-level error handling

Photography Filter Application | C++, CMake, Git

Aug 2024 – Dec 2024

- Led a three-person development team and coordinated integration with other project teams to deliver a photography filtering application
- Implemented an image-import interface and integrated JPEG-processing components into the application workflow
- Developed unit and integration tests to validate component behavior and cross-team integration

GNOME - Open Source Contributor | JavaScript

Oct 2026 – Present

- Authored and submitted a merge request to the upstream GNOME Shell project, currently undergoing maintainer review
- Modified GNOME Shell's brightness management functionality to add an automatic-brightness toggle to the system status interface

TECHNICAL SKILLS

Languages: C++, C, x86 Assembly, AVR, Rust, Go, Python, Java, SQL, JavaScript

Developer Tools: Git, CMake, Catch2, Maven, JUnit, Docker, Linux, Visual Studio, VS Code

Frameworks: React, Node.js, Swing