Mansur Nafis

Buffalo, NY | mansurna@buffalo.edu | 716-247-2920
linkedin.com/in/mansurnafis36

Education

University at Buffalo **Buffalo, NY**
B.S. in Computer Science | GPA: 3.6/4.0 *Expected December 2026*

Relevant Coursework: Systems Programming, Programming Languages (OCaml), Data Structures, Algorithms and Complexity, Computer Organization, Discrete Structures, Linear Algebra

Technical Skills

Languages: C, Java, Python, TypeScript, JavaScript, OCaml
Frameworks: React, Node.js
Tools: Git, Firebase, Google Cloud
Concepts: Data Structures & Algorithms, Functional Programming, OOP

Experience

Finta **Buffalo, NY**
Software Engineering Intern *January 2026 – Present*
- Optimized backend data fetching and caching logic to make core platform workflows **10–20x faster** under real user load.
- Refactored data access patterns and improved request handling to reduce latency and improve scalability across business dashboards.
- Develop full-stack production features using TypeScript, React, and REST APIs in a fast-paced startup environment.
- Build AI-driven product features using OpenAI APIs to automate fundraising workflows and enhance user experience.
- Collaborate in Git-based workflows with code reviews, documentation, and rapid iteration based on customer feedback.

Empathy Engineer **New York, NY**
Co-Founder *May 2024 – August 2024*
- Built an AI chatbot prototype leveraging OpenAI APIs to simulate structured virtual therapy conversations.
- Designed prompt workflows and response handling logic to improve contextual accuracy and conversation flow.

University at Buffalo **Buffalo, NY**
Teaching Assistant — CSE 305: Programming Languages (OCaml) *Spring 2026 – Present*
- Lead weekly recitations on functional programming, type systems, recursion, and interpreter design.
- Hold office hours assisting students with OCaml debugging and semantic reasoning.
- Evaluate programming assignments and provide structured technical feedback.
- Develop supplemental practice problems and walkthroughs to reinforce functional programming concepts.

Projects

Statically-Typed Mini Language Interpreter — *OCaml, Functional Programming*
- Implemented an interpreter for a statically-typed functional language supporting lexical scoping, closures, recursion, and higher-order functions.
- Designed and evaluated abstract syntax trees (ASTs) using environment-based semantics and recursive pattern matching in OCaml.
- Implemented closure environments to correctly capture variable bindings during function evaluation.
- Built a static type checker to enforce type safety prior to program evaluation.

Map Route Optimizer — *C, BFS, Dijkstra's*
- Implemented BFS and Dijkstra's algorithms to compute shortest paths on complex graph-based map inputs.
- Designed adjacency list structures to improve runtime efficiency and memory usage for large datasets.

Message Encoder/Decoder — *C*
- Built byte-shifting encoder/decoder to securely transform user messages for transmission and storage.
- Implemented real-time decoding workflow to support seamless server-side interpretation.

State Pattern Calculator — *Java, OOP*
- Implemented the State design pattern to eliminate conditional branching in scientific calculator logic.
- Designed modular class architecture enabling clean extension of mathematical operations.

Interests

Building Computers, Basketball, Weightlifting, Soccer, Football, Video Games