Will Beatty

331-307-9348 | wbeatty@protonmail.com | linkedin.com/in/willbeatty6 | github.com/wbeatty

Education

University of Michigan Ann Arbor, MI

Bachelor of Science in Computer Science, Minor in Mathematics *Aug. 2024 – Expected May 2028*

- **Coursework:** Computer Organization, Data Structures & Algorithms, Linear Algebra, Web Systems
- **Activities:** Michigan Student AI Lab, Sigma Chi Fraternity (President)

Experience

Software Engineer Intern July 2026 – Present

NOX METALS *Detroit, Michigan*

- Modeled RFQ win probability with a hierarchical Bayesian logistic regression in PyMC over ≈770 quotes, partially pooling alloy and customer intercepts, with a month-level random walk absorbing win-rate drift
- Engineered a commodity-adjusted price feature (log price vs. trailing 90-day same-alloy median) isolating markup from metal-market drift; validated on chronological splits via log-loss, AUC, calibration
- Propagated the full posterior through a revenue trade-off analysis, showing a 10% discount lifts win rate only ≈3% relative against an ≈11% break-even, ending the company's discount-to-win pricing
- Identified label censoring and exposure gaps from the experiment, then designed an immutable quote-exposure pipeline capturing every customer-visible offer and outcome across all quoting channels

Undergraduate Research Assistant June 2025 – Aug 2025

University of Chicago - Biological Sciences Division *Remote*

- Developed a Python Snakemake-based pipeline to process and analyze WGS samples using JabBa
- Packaged and deployed pipeline on a computing cluster using SLURM, increasing accessibility
- Gained experience with HPC workflows and troubleshooting distributed systems on a multi-node cluster

Projects

Strategy Backtesting Engine | *C++* June 2026

- Built an event-driven market-microstructure backtester in C++20 that replays historical exchange data tick-by-tick, with systemic strategy orders merged in, allowing interaction with the reconstructed book
- Implemented order-entry latency simulation by time-stamping submitted orders forward and merging them into the historical event stream through a monotonic clock, modeling realistic queue-positions
- Kept the replay path allocation-free with custom pool allocators and a lock-free SPSC ring buffer

High-Performance Limit Order Book | *C++* May 2026

- Engineered a low-latency limit order book matching engine in C++17 with price-time priority, achieving ≈ 10 mil orders/sec with O(1) best-bid/best-ask access and worst/best-case O(log M)/O(1) insertion
- Architected a lock-free three-stage pipeline (ingest, match, output) over custom bounded SPSC ring buffers with cache-aligned head/tail indices and acquire/release atomics to avoid false sharing
- Built a growable memory-pool allocator from scratch, bump-allocating order/limit objects from std::unique_ptr owned slabs and maintaining a manager-owned free list to re-use deallocated memory

Technical Skills

Languages: C++, Python, JavaScript, SQL, HTML/CSS
Developer Tools: Git/Github, Claude Code, Cursor, SLURM, Insomnia, Postman
Libraries & Frameworks: pandas, NumPy, React, Flask, Matplotlib