ADITYA TANEJA

Low-Latency Software Engineer with strong background in C++, multithreaded systems and quantitative financial modeling

Chicago, IL 8722941220 ataneja@depaul.edu Portfolio Github

EDUCATION

Masters in Computer Science | DePaul University September 2023 - June 2025

- Relevant Coursework :- Optimized C++, Optimized C++ Multithreading, Architecture of Real-Time Systems, Distributed Systems, Applied Algorithms, Real-Time Multithreaded Architecture, Real-Time Software Development I & II, Real-Time Networking.

Bachelors of Technology in Computer Science and Business Systems | NMIMS July 2019 - April 2023

- Relevant Coursework :- Computational Finance and Modeling, Financial & Cost Accounting, Probability and Statistics, Machine Learning, Artificial Intelligence, Data structures and Algorithms, Computer Networks, Advanced Social Media and Text Analytics, Operations Research, Cognitive Science and Analytics

PROJECTS

Zero-Latency Order Book Matching Engine

C++, Sockets, Multithreading, ZeroMQ, React.js, AWS

- Engineered a real-time matching engine using C++ and skip lists, handling huge quantities of orders in under 2s.
- Integrated a C++-to-Python data pipeline using ZeroMQ and JSON serialization, enabling real-time analytics and visualizations like order book heatmaps and depth charts with sub-500ms update intervals.
- Built and deployed an interactive front-end using React and WebSockets, providing live visualizations of market depth and order flow for simulated HFT environments

Custom Heap Allocator & Visual Debugger

C++, Memory Management, Debugging, Python (matplotlib)

- Developed a C++ allocator with free-list management, metadata encoding, and coalescing logic.
- Overloaded global new/delete operators to intercept and trace all heap activity.
- Created a Python-based animated debugger showing fragmentation, pointer links, and block states.

Real-Time Market Signal Forecaster

C++ (Server), Python (Forecasting), ARIMA/GARCH, Sockets

- Built a real-time TCP server in C++ to stream simulated market data under latency constraints.
- Python client consumed data, performed rolling statistical forecasts, and plotted live signal dashboards.
- Logged latency and forecast error metrics, applying time-series models in streaming mode.

Graphics Engine

C++, Test-Driven Development (TDD), Data-Driven Messaging Systems, System Optimization

- Engineered and implemented custom real-time memory and file management systems, reducing data access latency and enabling efficient handling of over 100,000 objects within strict time constraints.
- Developed and optimized real-time graphics engine components, integrating custom math library and object hierarchies, improving rendering performance.
- Utilized design patterns to abstract system components, facilitating a modular architecture that allowed for easy integration of new libraries, reducing system integration time.

Audio Engine

C++, Multithreaded Systems Design, Thread Synchronization

- Engineered a multithreaded audio engine using XAudio2, optimizing audio playback for multiple concurrent sound sources and reducing latency under real-time constraints.
- Designed and implemented thread-safe data management systems, improving asynchronous file loading performance, ensuring non-blocking synchronization for real-time audio playback.
- Developed a custom inter-thread communication system, improving data throughput and reducing resource contention across multiple threads.

Live 24/7 Algorithmic Trading Bot

Python, Multi-threading, Web Scraping, MongoDB, Flask, React

- Developed and deployed a fully automated trading bot capable of executing 24/7 live trades using multi-threading and event-driven architecture. Engineered a real-time market data pipeline using Python and APIs, processing live prices at high frequency for dynamic trade execution.
- Built and back tested multiple trading strategies including machine learning models like regression, classification, DNN, analyzing 6+ years of historical data to optimize trade decision-making.

- Designed and implemented technical indicators (MACD, RSI, Keltner Channels) and integrated web scraping for sentiment analysis (economic calendars, headlines).
- Developed a Flask API to serve trading insights and a React-based dashboard for monitoring bot status, price trends, and strategy performance.

AI-Powered Quantitative Alpha Research & Smart Beta Optimization Platform

Python, Pandas, Alphalens, NLP

- Engineered cross-sectional momentum and breakout factors; backtested against S&P 500 using factor rank autocorrelation, quantile returns, and performance attribution.
- Designed a smart beta optimizer with VaR and CVaR constraints to dynamically allocate capital across uncorrelated signals.
- Developed a robust backtesting framework simulating realistic execution, transaction costs, and slippage to validate out-of-sample performance.
- Built a fundamental NLP alpha extracting 10-K sentiment (Loughran-McDonald) and drift detection using Jaccard and Cosine similarity to forecast return anomalies.

Space Invaders

C#, Test-Driven Development (TDD), Data-Driven Messaging, Object Oriented Design Patterns

- Designed and developed a real-time Space Invaders clone, applying modern object-oriented design principles and game architecture patterns.
- Implemented key software design patterns (Factory, Observer, Strategy, Flyweight, Singleton, Object pooling, Adapter, Iterative, Data Driven) to enhance modularity and scalability.
- Engineered collision detection, sprite management, game loop architecture, and real-time input handling.

PROFESSIONAL EXPERIENCE

Backend Developer | Applied Philosophy of Science *December 2024 - Present*

- Leading the migration of an existing codebase from C++ to Java, enhancing maintainability and cross-platform compatibility.
- Developing a **distributed knowledge-based** operating system, allowing users to upload, share, and access research projects efficiently.
- Implementing **Java networking** solutions to facilitate seamless data exchange between researchers.
- Wrote **multi-threaded** backend services using OOP design principles.

Quantitative Researcher Intern | Wall Street Quants *Jun 2024 – August 2024*

- Developed and back tested statistical arbitrage strategies in cryptocurrency markets using Python, Pandas, and NumPy. Integrated performance evaluation metrics (Sharpe, drawdown, alpha) and simulated execution with market impact models.
- Engineered momentum and mean-reversion models, achieving a Sharpe ratio of 1.8 and simulated annualized returns of 15-20%.
- Implemented high-frequency data processing pipelines to analyze price-volume relationships and predict returns.
- Optimized execution strategies to reduce trading costs by ~20bps per trade, incorporating market impact analysis.
- Built back testing frameworks for performance evaluation, tracking metrics such as volatility, max drawdown, and alpha/beta.

Software Developer Research Assistant | DePaul University *March 2024 – Jun 2024*

- Designed and developed a distributed K-V storage system in Go with 99.9% availability, supporting 1M QPS with linearizable read / write operations. Leveraged consistent hashing for even data distribution and dynamic load balancing with data migration across Raft groups.
- Optimized performance using FollowerReads, RocksDB integration, and MVCC — reduced read latency by 42%.

SKILLS

Language: Python | C++ | C# | Java | Go | Scala | C#

Frontend: Tailwind CSS, JavaScript, TypeScript, HTML5, CSS3, Bootstrap, jQuery, AJAX, XML

Frameworks & Runtime Environments: SpringBoot | Django | Express.js | React | Next.js | Node.js | | Express.js

Database: MongoDB | PostgreSQL | MySQL |

Design Patterns: Factory, Observer, Singleton, Strategy, Adapter, Flyweight, Singleton, Object pooling, Adapter, Iterative, Data Driven

Finance & Trading: Quantitative Research, Statistical Arbitrage, Algorithmic Trading, Backtesting, Risk Management, Mean, Financial Modeling

Tools: Git | Perforce | AWS | Kafka