# Hardik Singh

972-330-1278 | hardikhssingh@gmail.com | linkedin.com/in/hardiksingh-hs/ | github.com/Hardik-Singh | hardiksingh.com

## Education

**University of Texas at Austin** — Austin, TX

*Bachelor of Science in Computer Science, Minor in Business* — *Aug. 2023 – May 2027*

- Relevant Coursework: Linear Algebra, Operating Systems, Probability & Stats, Algorithms, Diff. Equations

## Experience

**Incoming Software Engineering Intern** — Aug 2025 – Nov 2025

*Google* — *Seattle, WA*

- Will contribute to core systems development for Fuchsia OS in Rust, focusing on OS internals.

**Software Development Intern** — May 2025 – Aug 2025

*Amazon Web Services* — *Seattle, WA*

- Added raw query string support to CloudFront (CF) Functions by updating NGINX/Quickstep servers with FlatBuffer schema and serialization changes to meet Fortune 100 client requirements for migration.
- Enhanced Rust-based string parsing for CloudFront Functions using byte-level libraries, achieving 46% faster processing across 700+ edge locations..

**Software Engineering Intern** — May 2024 – Aug 2024

*John Deere Financial* — *Des Moines, IA*

- Designed a high-performance REST API with Python, FastAPI, and GraphQL to stream over 12 million telemetry data points from New Relic via AWS Kinesis; deployed using Terraform.
- Developed lightweight Java classes using Spring and Maven to implement structured JSON logging and traceability in AWS Lambda and ECS pipelines, efficiently capturing over 1M daily logs with 0.32% performance overhead.

**Software Engineering Intern** — May 2022 – August 2022

*Crestron Electronics* — *Plano, TX*

- Developed C++ libraries for automated control of Crestron DSP and control systems, enabling direct hardware interfacing and eliminating repetitive command execution across 30+ devices.
- Built a TensorFlow-based object detection pipeline to automate audiovisual device testing, improving validation speed by 330%.

## Projects

**Low-Latency Trading Exchange** | *C++, Python, Boost.Asio, pybind11, QuantLib, spdlog* — Nov 2023 – Present

- Created a C++ trading exchange with Boost.Asio async networking, lock-free queues, and DPDK for kernel bypass, achieving 5K orders/sec with sub-millisecond latency on an 8-core Apple M2.
- Built a custom order book using cache-friendly binary heap structures with memory pooling and price-time priority logic, ensuring consistent sub-millisecond execution in simulated trading workloads.

**Options Pricing Model** | *Python, C++, pybind11, Eigen, Accelerate* — March 2024 – Present

- Constructed a Python research framework with a C++ backend for Black-Scholes pricing via pybind11, enabling 100× speedup and vectorized Greeks computation to price 500K+ vanilla options/sec on Apple M2.
- Integrated a C++ order matching core from a custom low-latency trading exchange into a Python backtesting engine, enabling microsecond-level trade simulation and multi-asset PnL evaluation.
- Integrated time series models (GARCH, ARIMA) for volatility forecasting and strategy validation.

**Poker Bot** | *Rust, Python, NumPy, PyTorch* — Feb 2024 – Present

- Engineered a Rust-based poker engine with Monte Carlo rollouts and PyTorch RL agents, optimizing a custom game state and scheduler that supported 8K+ rollouts/sec for strategy inference in adversarial markets.
- Developed async human-bot gameplay with multithreaded task scheduling, validating 7+ RL models via a custom multi-agent tournament framework and deterministic Rust test harnesses.

## Technical Skills

**Languages**: Rust, C++, C, Python, Java, SQL, Go, JavaScript, TypeScript, R, HTML/CSS
**Frameworks**: Spring, React, Node.js, React-Native, FastAPI, Terraform, Django, Express
**Developer Tools**: AWS, Azure, Apache, Docker, Kubernetes, PyCharm, IntelliJ, Git, Jira, VS Code, Expo
**Libraries**: OpenCV, TensorFlow, Pandas, NumPy, Matplotlib, SciPy, PyTorch, Keras