Rajdeep Gill

hi.rajdeepgill@gmail.com | LinkedIn | GitHub

Education

University of Manitoba June 2026
Bachelor of Science in Computer Engineering **4.46/4.50 GPA**

Skills

Programming Languages: TypeScript, JavaScript, Python, C++, Java
Frameworks and Libraries: React, NextJS, PyTorch, OpenMPI, Node.js, Express, Flask
Tools: Git, Docker, PostgreSQL, Github Actions, Bazel, gRPC, Protocol Buffers (Protobuf)

Experience

Robinhood | *Software Engineer* **Sep 2025 - Current**

- Led a team of 3 engineers to build a tax lot selling feature, implementing observables for real-time market data updates and virtualization to ensure performant rendering for users with thousands of positions, driving $55M in notional volume
- Gained ownership of the entire margin web experience as the team's sole frontend engineer, driving feature launches through A/B testing and collaborating directly with backend, mobile, and design partners to ship features end-to-end
- Enhanced an internal cross-platform library (Kotlin → JavaScript/Swift) to bridge React context and Redux stores into Web Workers, enabling shared state and cache access for parallelized processing without blocking the main thread
- Built a highly composable set of React components for rendering interactive data tables, accelerating future feature development while supporting virtualization, sorting, filtering, column pinning, and pagination
- Built a native margin onboarding flow for Robinhood Legend, eliminating a redirect to the classic app, reducing onboarding time by 70% and increasing margin enrollment rates by 12%
- Redesigned the margin call interface with intuitive risk indicators, detailed requirement breakdowns, and actionable recovery steps to help users better understand their margin obligations and reduce overall fund recovery time

Robinhood | *Software Engineer Intern* **May 2025 - Sept 2025**

- Owned end-to-end development and delivery of a real-time margin maintenance table built with React and TypeScript, featuring embedded trade execution and risk indicators
- Overhauled the margin experience by expanding the server-driven UI to include futures traders and unifying business logic across Web, iOS, and Android, driving a 43% increase in visitors to the margin experience
- Enhanced a Protobuf-based server-driven UI architecture by implementing 12 foundational components in Python and Typescript along with a logging framework to increase system usage and maintainability
- Repaired the core integration suite for the margin experience and added 20+ test suites, achieving 93% code coverage

Biosensors Research Lab | *Machine Learning Research Intern* **May 2024 - Dec 2024**

- Created a custom convolutional neural network to predict cell viability from optical microscopy images, achieving 85% accuracy, establishing a non-invasive alternative to traditional staining methods for assessing cell health
- Developed a model validation GUI in Python displaying SHAP-based feature attributions, and visualizing cell images to debug model predictions
- Created a parallelized data processing pipeline with a new morphological parameter extraction algorithm, improving accuracy by 17% and cutting processing runtime by 75%.
- Replaced a high-risk system of manually-shared hard drives with a centralized MySQL database, allowing for cross-experiment queries and easy data sharing

Biosensors Research Lab | *Machine Learning Research Intern* **May 2023 - Aug 2023**

- Built a framework for electronic testing equipment, allowing for automation of entire experimental sequences through Python scripts and cutting manual setup time
- Programmed an FPGA to coordinate microfluidic device operations, integrating di-electrophoresis control with synchronized LED illumination and cell imaging

Projects

Distributed Image Processing *(C++, OpenMPI)*

- Accelerated the processing of gigapixel-scale images by 3.8x on 4 nodes by designing a scalable OpenMPI pipeline allowing data to be partitioned across an N-node cluster

AI Bug Trace *(Python, FastAPI, LangChain, Docker)*

- Built a dockerized backend service to track and evaluate developer debugging workflows by capturing file operations, terminal commands, test results and pull requests to collect better data for fine-tuning LLM models

Chess *(React, TypeScript, Socket.io, Flask)*

- Built a 2 player chess game with real-time updates and 250 millisecond synchronization latency through WebSockets