

Kabir Jain

6474469190 | k56jain@uwaterloo.ca | [linkedin.com/in/kabirjain25/](https://www.linkedin.com/in/kabirjain25/) | github.com/nointro3493 | kabirjain.netlify.app

EDUCATION

University of Waterloo

Waterloo, ON

Bachelors of Computer Science (BCS)

Sep. 2023 - April 2027

- Relevant Coursework: Data Structures and Algorithms, Object-oriented design, OS, Machine Learning

SKILLS

Languages: Python, C/C++, GoLang, Bash, Typescript, Scala Javascript, Java, PHP, SQL, Ruby on Rails, Rust

Technologies: React, Redux, Three.js, Next.js, Flask, FastAPI, gRPC, NoSQL, Supabase, Redis, GraphQL

Tools: AWS, Docker, Kubernetes, Datadog, Grafana, CI/CD

EXPERIENCE

Software Engineer

May 2026 – Present

Tesla

Palo Alto, CA

- Incoming Summer 2026 focusing on **Fleetnet** systems

Software Engineer

Sept. 2025 – April 2026

Shopify

Toronto, ON

- Designed and deployed **load-shedding** cooldown logic to mitigate thundering-herd behavior during service restarts, stabilizing dependency recovery and **preventing cascading outages** under burst traffic.
- Hardened **payment security** boundaries by re-architecting **Google Pay** token flows to use session-scoped indirection and in-service decryption, eliminating propagation of encrypted tokens
- Debugged and fixed payment flow issues in Shopify's CardServer (transaction processing) and CardSink (settlement pipeline), resolving **gRPC** and database-level errors that caused failed charges using **Ruby** and **GoLang**.
- Designed and maintained internal service **APIs** used across payment infrastructure, ensuring backward compatibility and safe rollouts across dependent services at **99.9%** uptime

Founding Engineer

November 2025 – February 2026

Heavy Influence

San Francisco, CA

- Developed **React Native** creator app, shipping production-ready MVP in 6 weeks and onboarding 120+ creators
- Added structured logging, error monitoring, and alerting (**Sentry/Cloud Logging**) to reduce production incidents
- Designed event-driven **Firebase** backend handling scheduled jobs, and async post processing for **300K+** posts.

Software Engineer

May 2025 – Aug. 2025

Global X ETFs

Toronto, ON

- Developed ML-assisted **C++/SQL** market simulators using NumPy/Pandas-validated datasets, enabling ETF teams to backtest strategies and cutting analysis time by **40%**.
- Built predictive trading features with **Python**, using **Pandas** for feature engineering and **PyTorch** prototypes for modeling volatility, order-flow behavior, and risk signals.
- Designed **Airtable** bases and automations for research/ops tracking, linking datasets, experiments, and model outputs to streamline coordination between engineering and quantitative teams.

PROJECTS

EasyASL - HTV8 Winner [Learn more.](#) | *NodeJS, OpenAI, PicsArt API, Tailwind, FireBase, TypeScript*

- Created the **first** English to American Sign Language translator (before **Sign Gemma**) using **Typescript**
- Improved EasyASL's image translation from American Sign Language to English by **fine-tuning LLM** data, and optimizing **deep learning** network, reducing processing time by **45%** and improving overall accuracy.

Lunero [Learn more.](#) | *Supabase, TypeScript, PostgreSQL, RESTful, React, OpenAI*

- **Co-Founded** a real estate platform, allowing buyers and sellers to find the perfect realtor based on client's desires by implementing a **proprietary matching algorithm** and **OpenAI**, achieving **300+ users** in our waiting list
- Pitched to top VC firms such as **Sequoia Capital**, and **a16z** as **1 of 8 teams** selected from **350+** applicants

Investment Portfolio Optimizer [Learn more.](#) | *Python, yFinance, Numpy, Matplotlib, Pandas*

- Utilized **Python** and **yfinance** to optimize portfolios by analyzing trends, **reducing analysis time by 70%**
- Applied the **Monte Carlo** method to simulate over **1,000** portfolio combinations, utilizing **multithreading** and random weight distributions to assess potential outcomes, and identify the most efficient portfolios.
- Leveraged **NumPy** for numerical computations and **Matplotlib** to visualize portfolio distributions, risk-return tradeoffs, and performance trends, enhancing data-driven decision-making.