MUSTAFA MOTIWALA

+1 (647) 915-5352

mustafa@motiwala.ca ⋄ linkedin.com/in/mustafa-motiwala ⋄ motiwala.ca ⋄ Github

SKILLS

Programming Languages	Go | Python | Haskell | Idris | Racket | Typescript/Javascript | Java | PHP | Lean Julia | C | C++
Technical Skills	ReactJS | NumPy | Pandas | SQL | HTML/CSS | SageMath MATLAB | Qt framework | Verilog

EXPERIENCE

Software Engineer (Intern) Tesla, Inc. *Palo Alto, CA, United States* May 2025 - Sep 2025

- Reduced space utilization of critical telemetry software by a **factor of 10**.
- Developed a **secure settings syncing** system for various electronic control units on the Megapack.
- Designed a generic service for syncing Tesla Megapack files to the cloud.

Software Engineer (Intern) Tesla, Inc. *Palo Alto, CA, United States* May 2024 - Sep 2024

- Improved battery calibration algorithms on the Tesla Powerwall 3 to **speed up calibration** by **300-800%**.
- Added revenue grade virtual meters to the Tesla Powerwall 3, allowing utilities customers to realize up to **millions of dollars** of revenue.
- Upgraded **bootloader software** written in **Go** on the Tesla Powerwall 3 and Tesla Megapack to make over the air firmware updates **more robust** and **improve uptime** by up to 50%.
- Designed a pipeline to automatically decode and upload CAN traces to the cloud, saving **2 hours** of debugging time **per service engineer per day**.

Software Engineer, Chirality Research, Inc. *Houston, TX, United States* Jun 2022 - May 2024

- Implemented custom protocols to **migrate 600 terabytes of data** between databases **8 times faster** than existing solutions. Used a modified CSV format and specialized software implemented in C++ to quickly upload files to Amazon S3.
- Supported **development of a web API** for reading and writing to an in-house database service. Implemented a custom Python client for the API.
- Maintained and **upgraded legacy PHP websites**, including full rewrites of **data visualization software** in modern Javascript, creating a more robust and maintainable platform for domain experts to analyze data.
- Designed compact file formats to **compress 400 gigabytes** worth of domain-specific data, halving storage costs and **doubling data transfer speeds**. Implemented serializers and deserializers for said format in Python.
- Automated several dataflows, ranging from SQL stored procedures to transfer data between databases to custom Python and Julia scripts which moved data between entirely different services.

CS Research: PL design; named effect handlers, University of Toronto Dec 2023 - Jun 2024

- Researching **algebraic effects** and named handlers, a technique which generalizes exceptions in a functional way.
- Implemented named handlers as a **Haskell library**.

CS Research: Supervised compilation, University of Toronto Aug 2023 - Dec 2023

- Researching supervised compilation: **optimizing programs** by representing data dependencies as graphs.
- **Implemented** supervised compilation techniques in Racket for a **subset of Scheme**, demonstrating its viability at a larger scale and opening the doors for applying supervised compilation in the real world.

Teaching Assistant, University of Toronto Sep 2023 - Present

- Helped deliver 6 courses covering theoretical computer science, programming languages, calculus, and linear algebra.

PROJECTS

KDE Connect (2021). Extended the open source, multi-platform Java & C++ desktop app KDE Connect with several features, including a **remote file manager**, the ability to execute **remote commands**, and **remote desktop** capabilities. The C++ app is developed in the Qt framework. (Repo here)

TBL Programming Language (2022). Created a dynamically typed, higher-order **functional programming language** with support for Lisp-style macros, implemented in Javascript. (Repo here)

Instagram Bots (2023). Created a TypeScript library for writing bots that **automate interactions on Instagram**. Integrated TBL so as to enable programmatic interactions within the app, allowing for a Wordle clone playable in Instagram chats.

Arcade Musti (2024). A **React+Typescript** web arcade, containing reproductions of classic games like 2048, Sudoku, Minesweeper, Flow, Wordle, and others. (Repo here); (Website here)

Rubik's Reverse (2025). A **Rubik's cube solver** which can run "in reverse", allowing users to **generate arbitrary patterns** on the Rubik's cube, written in **Haskell** and **React**. (Repo here) (Website here)

Rubiksify (2025). A **Rubik's cube mosaic** generator; a **Haskell** and **React** webapp to help recreate images in real life with Rubik's cubes. (Repo here) (Website here)

Simply Typed Lambda Calculus (2023). Implemented a **dependently-typed** evaluator and normalizer using Idris for the simply typed lambda calculus in which program representations were inherently well-typed. (Repo here)

Munchausen numbers (2022). Designed and implemented algorithms in **Python** & **Julia** to search for *Munchausen numbers* – numbers equal to the sum of their digits raised to their own power – **improving** upon currently known methods by a **factor of 10**. (Repo here)

RELEVANT COURSE WORK

Machine Learning.

- Implemented ML algorithms like decision trees, nearest neighbours, and neural networks to create **music synthesizers**, **handwriting recognizers**, and heart **disease predictors**.
- Used industry standard tools like NumPy, Pandas, and SciKit.

Electrical engineering.

- Designed **arithmetic circuits** which used 7 segment displays to show results. Implemented this circuit by hand on **breadboards**.
- Used hardware description languages like **Verilog** to synthesize a Morse decoder circuit which flashes Morse codes on LEDs.

Systems Programming

- Wrote C programs to add an **echo sound effect** to WAV audio files.
- Implemented a small **Unix shell** in **C**.
- Designed a custom **file system** in **C**.
- Created a **preemptive threading library** in **C**.

Mathematics

- Generated 3 dimensional phase portraits for systems of differential equations with **Matlab**.
- Implemented **elliptic curve cryptography**, compute **field quotients**, and automate common **linear algebra** routines with **Sage Math**.
- Used **Lean** and **Coq** to formalize theorems from number theory, linear algebra, and computer science.

EDUCATION

Bachelor of Science, Computer Science and Mathematics

University of Toronto — Sept. 2022 - Jun. 2026 (Expected)

- Dean's List Scholar & 3.95 GPA