Sourish Das

705-970-5318 | s4das@uwaterloo.ca | linkedin/sourishdas | github/sourishdas07 | sourishdas.app

Education

University of Waterloo **Waterloo, ON**

Honours Bachelor of Computer Science (Co-op) *Exp. Grad Date: 2025*

- **Clubs:** Computer Science Club · Data Science Club · Blockchain Club
- **Relevant Coursework:** OOP in C++ · Compilers · Data Structures · Linux · Functional Programming · Linear Algebra · Calculus 2

Experience

Backend Software Engineer **May 2023 - Sept 2023**

Baraka (YC 21) *Dubai, UAE*

- Refactored various legacy APIs to help the company move towards microservices using **TypeScript**, **NestJS**, **Java** and **Springboot**
- Implemented a cutting-edge semantic search feature to enhance the user experience by tailoring investment opportunities to their creative financial inquiries and boosting total search requests by **17%** , leveraging **OpenAI API**, **TypeScript**, **NestJS**, and **Milvus DB**
- Built the backend logic, including a **RESTFUL API** with **CRON** jobs that update specific columns in the **Postgres DB** for a new passive income-generating discovery page to boost investments in dividend-paying stocks by **21.5%** using **Typescript** and **NestJS**
- Reduced search latency by up to **60%** for both standard and semantic search with the help of **Meillisearch** and **Milvus**
- Designed a CLI microservice to allow developers to easily perform **CRUD** operations in **Milvus DB** with **Java**, **Gradle**, and **Springboot**
- Monitored logs to swiftly resolve production issues and optimized S3 image uploads, boosting reliability and user experience in **AWS**

Quantitative Developer **May 2023 - Sept 2023**

WatStreet *Waterloo, ON*

- Developed an algorithm capable of identifying **Triangular Arbitrage** opportunities across both centralized and decentralized cryptocurrency exchanges utilizing **Python** and **Poloniex API**
- Researched profitable **Cross-Exchange Arbitrage** opportunities between **Poloniex** and **Binance** to find over **25** daily trades that would be profitable with the help of **Jupyter Notebook**, **Python**, **Poloniex API**, **Binance API** and **Pandas**
- Developed a daily changing graph to track the correlation between different coins on various exchanges to identify Arbitrage opportunities using **Jupyter Notebook**, **Python**, **Pandas** and **Matplotlib**

Full-Stack Developer **Sept 2022 - May 2023**

UW Orbital *Waterloo, ON*

- Developed a **MySQL** database that is in charge of managing **500+** records of satellite telemetry data
- Implemented a connection from the Mission Control Website to a Firebase DB using **React** and **Firebase DB**
- Designed a homepage for the ARO Website to allow users to send information to our satellite using **React**

Projects

Bull Forecast | *Python, Streamlit, Yahoo Finance, Prophet, Pandas, Matplotlib*

- Developed a web app that previews price forecasts on their stock and crypto holdings using **Streamlit**
- Analyzed the relative return of different entities over a given time period using **Python**, **Pandas** and **Matplotlib**
- Allows users to compare the price trends over a given amount of time to future lower-bound, upper-bound and mean predictions using **Python**, **Pandas** and **Matplotlib** with a **81% accuracy**

INV$T | *Python, OpenCV, Mediapipe, ReactJS, TailwindCSS*

- Created a script that sends users an email notification if a specific stock in your created portfolio reaches a certain target price, or a likely possibility of providing a certain return, with the help of **Python**, **Pandas** and **Smtplib**
- Users are able to preview their portfolio and witness real-time price changes using **Yahoo Finance**, and **Python**
- Implemented mean regression to make the forecasts

Stock Correlation Calculator | *Python, OpenCV, Mediapipe, ReactJS, TailwindCSS*

- Implemented image processing with **OpenCV** and **Python** to accurately track reps of over 6 exercises in video streams
- Optimized **Python** scripts to achieve a rep counting accuracy of over **95%** through extensive testing and optimization
- Utilized **Flask** as the backend framework and **MongoDB** as the database to efficiently store and retrieve user data

RepMe | *Python, OpenCV, Mediapipe, ReactJS, TailwindCSS*

- Using **OpenCV** and **MediaPipe**, developed an algorithm responsible for counting reps for various home workouts
- With the help of **Flask** established a connection between the Python back-end and the front-end
- Designed and built a front-end using **React** allowing users to choose a suitable exercise based on their skill level

Netflix2.0 | *React, Firebase DB, Stripe Checkouts, TMDB API*

- Developed a full-stack web app replicating the current look and feel of Netflix, using **React** and **TMDB API**
- Implemented a login feature to allow users to create accounts and personalize their experience using **Firebase DB**

Technical

Languages: TypeScipt, JavaScript, Java, Python, SQL, C++, C, Scheme, R
Databases: PostgreSQL, MySQL, MongoDB, Firebase
Frameworks: AWS, NestJS, Express, Flask, Springboot, FastAPI, ReactJS, NextJS, Node.js, Tailwind
Libraries: pandas, NumPy, plotly, Matplotlib, scikit-learn, Tensorflow, OpenCV, Poloniex API, Binance API, Finnhub API, yfinance