Yasen Behiri

yasen.ca ybehiri@uwaterloo.ca linkedin.com/in/yasenbehiri github.com/smolbrainer

Education

University of Waterloo September 2025 - April 2030

Bachelor of Software Engineering - Honours/Co-op, Joint Honours Computing & Optimization

- Relevant Coursework: Object Oriented Programming, Programming Principles, Embedded Systems, Data Structures

Experience

Software Engineering Intern September 2025 – Present

Heavy Influence *Remote*

- Developing a mobile-first app for creators to schedule and publish videos from a single platform.
- Supporting cross-platform posting across **TikTok, Instagram Reels, YouTube Shorts, and Facebook Reels**.
- Integrated posting APIs with **OAuth**, enabling **500MB+ uploads** and multiple accounts per platform.
- Built scheduling engine for **365-day publishing** with timezone-aware triggers and caption customization.
- Designed backend using **Firebase** for user data, account linking, subscription tiers, and scheduled post management.
- Building analytics dashboard to visualize cross-platform metrics like views, likes, and engagement growth.

Software Engineering Intern May 2025 – August 2025

Savi Finance *Toronto, ON*

- Launched **Social Insights** dashboard in React/Redux, rapidly scaling to **1K+ weekly active users** in 6 weeks.
- Built **Chrome** extension to auto-sync checkout data across platforms, cutting manual transaction entry by **90%**.
- Engineered GraphQL API with Apollo, achieving **170ms p95 latency** while handling **50K+ transactions/month**.
- Integrated **Coinbase Wallet** sync for **1.3K+ accounts**, achieving **95.2% automated balance accuracy**.
- Implemented tests with **Jest** and **Cypress**, increasing deployment reliability and maintaining **92%** code coverage.

Projects

TradeSmart AI | *React, Flask (Python), Random Forest, Chart.js, Firebase*

- Built a full-stack trading platform with **React** and **Flask**, enabling real-time stock and trade tracking and simulations.
- Designed dynamic, interactive financial charts using **Chart.js** and used **Firebase** for seamless real-time data syncing.
- Improved performance with **24-hour prediction caching**, lazy loading, and a responsive glassmorphism UI.
- Implemented **Random Forest** models using **25+ features** to generate **1-year price forecasts** with confidence scoring.

StepGuide | *Python, PyTorch, OpenCV, YOLOv11, COCO, Coqui TTS, TensorRT*

- Built an assistive wearable using **Python**, **PyTorch**, and **OpenCV** for the visually impaired.
- Enabled **30+ FPS** real-time object detection and accurate audible navigation through integrated speech output.
- Trained and deployed a custom **YOLOv11** model on **COCO**, reaching **96%** accuracy across **80+ classes**.
- Integrated **Coqui TTS** for continuous, context-aware real-time narration of detected surroundings.
- Optimized inference via **PyTorch → ONNX → TensorRT** pipeline with **FP16** precision.
- Achieved **120ms** latency on **Jetson Orin Nano** using parallel GPU acceleration for efficient edge deployment.

Get "Stuff" Done | *React Native, FastAPI, Stripe, Twitter/X API, OpenAI, Gemini, MongoDB*

- Developed a full-stack **React Native** + **FastAPI** app that gamifies habit tracking for better accountability.
- Implemented penalties like spam texts, AI calls, donations, and public **Twitter/X** posts when tasks are missed.
- Built secure payment flows with **Stripe** for auto donations, resolving OAuth and webhook issues for reliable execution.
- Designed **MongoDB** schema and event-driven backend for habits, penalties, and uploads, ensuring **24/7** operation.
- Leveraged **OpenAI & Gemini** LLMs with **Fetch.ai** agents for dynamic text generation and SMS voice prompts.

Cryptocurrency Twitter Bot | *Python, Tweepy, RegEx*

- Developed a **Python** Twitter bot that searches for tweets containing contract addresses of new cryptocurrencies
- Automated data extraction using the **Tweepy** library to help users discover emerging crypto projects.
- Implemented **RegEx** to filter and validate contract addresses from tweets, streamlining crypto project discovery.

Dino Game Auto-Jump System | *Arduino, C, Ultrasonic Sensor*

- Engineered an **ultrasonic sensor** system to detect object speed and trigger the Dino jump in Google's offline game.
- Programmed the system using **Arduino** and **C**, integrating analog sensor data with game controls.

Technical Skills

Languages: Python, C, C++, Java, JavaScript, Ruby, SQL, HTML, CSS, MATLAB
Frameworks/Libraries: React, React Native, Node.js, Ruby on Rails, GraphQL, Firebase, MongoDB
Tools & Platforms: Git, Linux, Docker, MySQL, Jira, Postman, GTest, Google Cloud