Harsh Upadhyay

437-986-2285 upadhh1@mcmaster.ca Linkedin Github Personal Portfolio

Education

McMaster University

Honours Computer Science Co-op (B.A.Sc.) *Hamilton, ON*

- Presidents entrance scholarship valued at $5000 for incoming students with above an 96% average in high school
- Relevant Coursework: Applied Cryptography, Fundamentals of Machine Learning , Principles of Programming Languages, Operating Systems, Concurrent Systems, Data Structures and Algorithms, Databases.

Technical Skills

Programming: Python, JavaScript, TypeScript, C, C++, C#, Java, SQL, Haskell, MATLAB, R, Bash / Shell
Technologies: React.js, Next.js, Node.js, Django, Tailwind CSS, PyTorch, TensorFlow, OpenCV, DeepFace, GraphQL
Tools: Git, GitHub, Docker, JUnit, Cloudflare, PostgreSQL, MongoDB, APIs, Postman, Jira, Agile/Scrum, Linux, CI/CD

Experience

Government of Ontario – MPBSDP **May 2025 – December 2025**

Business Analyst Co-op *North York, Ontario*

- Enhanced Match and Merge Automation by raising automated throughput from **30% to 70%** through leading **Agile grooming/refinement**, translating BRs into technical tasks, and delivering implementation-ready **Jira** user stories.
- Validated system functionality by executing **REST API** and third-party interface tests using **Postman**, simulating scenarios to reproduce issues, identify defects, and ensure compliance with business rules.
- Used **SQL** and advanced analysis to assess postal code mapping reliability by comparing current records with historical data and GeoJSON boundaries, identifying discrepancies and evaluating **migration feasibility** for enhancements.

Sciencious - United Arab Emirates **April 2022 – August 2022**

Frontend Developer *Dubai, UAE*

- Spearheaded the development and deployment of the company's website using **Next.js**, achieving a **20%** boost in user engagement by delivering a responsive, mobile-friendly interface that increased average session duration by **35%**.
- Engineered and optimized **RESTful APIs** with **PostgreSQL** database integrations, reducing data retrieval times and improving system scalability for real-time analytics.
- Coordinated with cross-functional teams to design and integrate web applications, enhancing internal workflows and customer experience, resulting in a **15%** improvement in **project delivery efficiency**.

Projects

Workout Tracker | *Next.js, TypeScript, Tailwind CSS, PostgreSQL, Prisma, NextAuth.js*

- Implemented a personalized fitness dashboard with **data-driven visualizations**, enabling users to track workout volume and progress over time through interactive charts and **session-aware views**.
- Implemented a calendar-based workout history system supporting **CRUD operations** for exercises and sessions, reducing workout logging time by **40%** through state-consistent, session-aware workflows.
- Integrated an AI-powered workout assistant using **OpenRouter** and `meta-llama/llama-3.2-3b-instruct`, delivering context-aware recommendations by leveraging authenticated user data and historical activity.

PulseHTTP | *C++, Linux, epoll, Multithreading, Sockets*

- Built a high-performance **HTTP server** in **C++** using **POSIX sockets**, **epoll**, and **thread pools**, achieving **7.4× throughput** (**34K req/sec**) with **10K+ concurrent connections**.
- Implemented **event-driven epoll design** to minimize blocking I/O, reduce context switching, and improve CPU utilization.

LinkSnap | *Python, aiohttp, Redis, SQLite, Next.js*

- Developed an asynchronous link management service using **aiohttp** and non-blocking I/O, incorporating rate limiting (**10 req/min per IP**) and robust input validation to ensure safe, high-throughput request handling.
- Designed a cache-first resolution pipeline with **Redis** and SQLite persistence, reducing database hits by **60%** while guaranteeing deterministic link mappings, surfaced through a lightweight Next.js frontend.

ActiveTrack | *Python, Pandas, NumPy, Scikit-Learn, Matplotlib*

- Built a **machine learning model** with **98.9%** accuracy identifying barbell exercise types from **IMU sensor** data.
- Engineered **25+ features** from **accelerometer/gyroscope** time-series using **FFT**, statistics, and clustering.
- Optimized model performance with **grid search** across **Random Forest**, **Decision Trees**, and **SVM**.
- Developed automated rep-counting using **LowPassFilter** and peak detection, achieving **95%+** accuracy.