Adarsh Kumar

519-575-6199 | adk2k6@gmail.com | adarshkumar.ca | linkedin.com/in/a369kuma | github.com/a369kuma

Education

University of Waterloo Waterloo, ON

Bachelor of Applied Science in Management Engineering, Minor in Artificial Intelligence *Sep. 2023 – Apr. 2028*

Relevant Coursework: Data Structures & Algorithms, Databases & Software Design, Probability & Statistics, Optimization

Technical Skills

Languages: Python, Java, C++, SQL, TypeScript, JavaScript, Kotlin, Bash
Frameworks & Data: React, Node.js, Flask, REST APIs, PostgreSQL, Redis, Redshift, Elasticsearch, RabbitMQ
Cloud & Systems: GCP (Apigee), AWS (S3, Glue, Lambda, API Gateway), Kubernetes, Docker, Ubuntu Linux, GitHub Actions

Experience

Software Engineering Intern Sep. 2026 – Present

Ford Motor Company *Waterloo, ON*

- Reduced vehicle-data API latency by **25%** by migrating legacy endpoints to **GCP** behind **Apigee**, redesigning request routing while automating backend processing and deployment workflows with **Python** and **Bash**.
- Built **Java** and **Kotlin** tooling over **Elasticsearch** to correlate cloud feedback with production vehicle signals, surface actionable patterns, and inform feature eligibility, rollout, and prioritization decisions.
- Automated privacy checks for **1,000+ Product Development employees** by engineering a production VIN lookup system on **Ubuntu Linux** with **SQL**-backed validation, audit logging, and traceable access controls across internal workflows.

Data Engineering Intern Jan. 2026 – Apr. 2026

Canso Investment Counsel Ltd. *Toronto, ON*

- Processed **500K+** daily trade and position records by building **Python** data pipelines with **AWS Glue** and **S3**, validating transaction fields and loading normalized datasets for downstream investment analysis.
- Improved recurring analytical query performance by **20%** by restructuring **Redshift** tables and optimizing **SQL** query paths, eliminating redundant joins and full-table scans across large production datasets.

Software Engineering Intern Sep. 2025 – Dec. 2025

CareMetric Solutions Inc. *Waterloo, ON*

- Supported production healthcare workflows across **8+ clinics** by developing **React** and **TypeScript** interfaces that connected form validation, user actions, and application state with internal review systems.
- Reduced recurring manual data entry by **95%** while processing **100K+ patient records per week** through **Python** and **Flask** services that validated fields, rejected malformed records, and persisted clean data.

Software Engineer Sep. 2024 – Present

UW Formula Electric *Waterloo, ON*

- Improved vehicle-system debugging by developing **C++** tooling that decoded **300+ CAN signals per test session**, validating sensor values, communication timing, and controller-state transitions across repeated on-vehicle tests.
- Reduced manual fault isolation across **dozens of recorded vehicle sessions** by building **Python** log-replay workflows on **Ubuntu Linux** that reproduced system behavior and traced failures to sensor data, message timing, or controller behavior.

Projects

TapInPass | *Python, Flask, AWS, PostgreSQL, NFC, REST APIs, GitHub Actions*

- Replaced manual attendance workflows for **85+ paid center orders** across Canada and the U.S. by building an **NFC**-based platform with **Python**, **Flask**, and **PostgreSQL** for real-time student check-in and check-out.
- Improved backend reliability by developing **REST APIs** on **AWS** that validate tap state, prevent duplicate check-ins, persist attendance events, and reject invalid workflow states before database writes.
- Streamlined production releases by automating deployment through **GitHub Actions** and structuring backend workflows around validation, persistence, and failure handling across deployed customer environments.

Linkarena | *TypeScript, Node.js, PostgreSQL, Redis, AWS*

- Built a stadium engagement platform with **paid active users**, used by **Minnesota Twins Minor League Baseball affiliate teams**, coordinating real-time check-ins, payments, vouchers, and fulfillment through production backend workflows.
- Improved reliability across asynchronous workflows by designing a **Node.js** and **TypeScript** API with **PostgreSQL** for transactional state, **Redis** for low-latency reads, and idempotent queued processing for retries and concurrent requests.