

# Riyasat Zaman

Edmonton, AB | riyasatzaman@gmail.com | +1 780-655-6808 | [www.riyasatzaman.dev](http://www.riyasatzaman.dev) | [github.com/riyasatzaman](https://github.com/riyasatzaman) | [linkedin.com/in/riyasatzaman](https://linkedin.com/in/riyasatzaman)

---

## EDUCATION

**University of Alberta** — BSc Computing Science, Statistics Minor

Expected Nov 2026

**Relevant coursework:** Software Engineering, Algorithms, Artificial Intelligence, Machine Learning, Database Systems, Human-Computer Interaction, Web Applications, Operating Systems, Object-Oriented Programming

---

## TECHNICAL SKILLS

**Languages:** Python, TypeScript, JavaScript, Java, C++, R, SQL

**Frontend:** React, Zustand, Tailwind CSS, Material UI, Jinja2

**Backend & APIs:** Flask, Django, Node.js, Express.js, REST APIs

**Data & Database:** PostgreSQL, MySQL, MongoDB, Firebase, Snowflake, Airflow, Streamlit

**AI & CV:** PyTorch, OpenCV, Hugging Face, LLMs

**DevOps:** Git, GitHub Actions, Docker, Docker Compose, Linux, NGINX, CI/CD

---

## TECHNICAL EXPERIENCE

**Software Developer** — Space Exploration Alberta Robotics (SPEAR)

Sep – Dec 2023

*Autonomous rover navigation — sensor fusion, localization, terrain classification*

- Integrated SLAM workflows synchronizing LiDAR and IMU data for precise rover localization across 7+ navigation tests, reducing localization error by ~35% over the previous GPS-only approach.
- Built computer vision pipelines using OpenCV and PyTorch to automate terrain classification and real-time obstacle detection.
- Implemented real-time path planning and sensor fusion in Python and C++ with ROS, contributing to the full autonomous navigation stack used in live field tests.

**Product Developer** — UofA Blueprint

May – Dec 2022

*Production full-stack application for ISA iCard rewards platform — API design, database architecture, QA*

- Shipped a production React/TypeScript/Tailwind frontend for ISA's iCard rewards app serving 10,000+ international students, implementing 10+ feature screens for vendor verification and discount redemption.
- Developed RESTful APIs with Node.js, Express.js, and PostgreSQL for user registration, content management, and admin workflows.
- Led QA and usability testing across 3 user evaluation rounds, improving Lighthouse performance score from ~65 to 90+.

---

## PROJECTS

**Verdigris — Distributed Social Network** | Django, React, PostgreSQL, Docker | [Video Demo](#)

Apr 2025

- Built a Django/React distributed social network across 5 independently hosted nodes, enabling real-time cross-server posts, comments, likes, and follows between users on different servers.
- Designed REST API contracts for cross-node synchronization of authors, posts, and follows, ensuring consistent state across all nodes.
- Deployed 5 cloud-hosted nodes with Docker, NGINX, and PostgreSQL; secured node-to-node requests with HTTP Basic Authentication.

**TTC Transit Reliability Monitor** | Python, Airflow, Snowflake, dbt, Docker, Streamlit | [GitHub](#) | [Demo](#)

May 2026

- Built an end-to-end Python data pipeline ingesting live TTC vehicle-location JSON every 15 minutes, landing raw data in Snowflake and transforming through dbt staging, intermediate, and mart layers.
- Orchestrated ingestion and transformation workflows with Apache Airflow and Docker Compose, processing 73,924+ vehicle observations across 218+ routes and 3,235+ distinct vehicles.
- Implemented 44 automated dbt and pytest checks validating source keys, timestamps, route relationships, and data quality contracts.

**FocusPilot — Gamified Productivity App** | React, TypeScript, Zustand | [GitHub](#) | [Demo](#)

Mar 2026

- Built a modular React/TypeScript productivity app with Pomodoro sessions, task management, notes, XP-based progression, and persistent local session analytics.
- Engineered a background timer system that maintains accurate session state across navigation, reloads, and app state changes.
- Designed a gamified progression system with XP, levels, and 5 tracked user stats; users self-reported a 40–50 minute improvement in focus duration across evaluation sessions.

**Capto — AI Image Captioning Tool** | Flask, Hugging Face, GitHub Actions | [GitHub](#) | [Demo](#)

Mar 2025

- Built an AI captioning app using Flask and Qwen3-VL via Hugging Face Inference Providers, supporting image upload, caption generation, regeneration, editing, and one-click download.
- Deployed on Render with Gunicorn, environment-based configuration, fallback handling, and a GitHub Actions CI/CD pipeline.