Rithvik Sunil

+1 (437) 662-7064 | rithvik11sunil@gmail.com | linkedin.com/in/rithvik-sunil | Personal Website

Education

University of Toronto Toronto, ON

Bachelor of Science in Computer Science, Minor in Statistics *2027*

- Relevant Coursework: Database Management Systems, Machine Learning, Computer Architecture, Operating Systems, Principles of Programming Languages, Paralleling Programming, Systems Programming, Data Structures and Algorithms, Object-Oriented Programming, Data Analysis, Software Design

Technical Skills

Languages: C, C++, Haskell, Python, Go, Java, JavaScript, MIPS Assembly, R, mySQL, Dart
Developer Tools: Git, VS Code, Cursor, Saturn, Maven, Docker, Adobe After Effects
Libraries & Frameworks: Pandas, Numpy, Matplotlib, Tensorflow, Sklearn, Flutter, Django, Flask, Firebase, JUnit, Node.js, Express.js, MongoDB, React, Next.js, Vite

Experience

Director of Technology September 2024 – April 2026

University of Toronto Scientific Research Initiative (UTSRI) *Toronto, Ontario*

- Built a responsive, multi-page web application using **Next.js**, **React**, and **Tailwind CSS**, implementing server and client components for fast loads and clean routing.

Software Engineering Intern September 2024 - November 2024

Coubon *Toronto, Ontario*

- Contributed to a cross-platform **Flutter application** (iOS, Android, Web) using **Riverpod** for state management and **Firebase** services (Auth, Firestore, Storage) for backend integration. Implemented key UI features.

Projects

CodeAtlas AI | *Next.js, TypeScript, FastAPI, PostgreSQL/pgvector, Redis, OpenAI* May 2026

- Engineered a **full-stack AI repository analysis platform** as measured by end-to-end indexing of public GitHub repositories, by building a **Next.js/TypeScript** dashboard, **FastAPI** backend, **PostgreSQL/pgvector** schema, and **Docker Compose** development environment.
- Implemented an **asynchronous indexing pipeline** as measured by tracked repository status, file counts, line counts, and re-index support, by using **Redis Queue (RQ)**, cloning public repos, applying ignore rules, chunking source files, generating **OpenAI embeddings**, and storing vectorized code chunks.
- Built **retrieval-augmented generation (RAG)** features as measured by architecture summaries, semantic code search, file explanations, and repository-aware chat with **cited source files**, by combining **vector similarity search**, **OpenAI chat models**, dependency graph parsing, and a polished developer dashboard.

NeuroCPP | *C++20, CMake, GoogleTest, Machine Learning, Autograd* July 2025

- Built a **machine learning framework from scratch** as measured by a custom **N-dimensional Tensor engine**, matrix operations, activation functions, and benchmark executables, by implementing contiguous memory storage, shape/stride tracking, indexing, reshaping, transposition, and **matrix multiplication** in **modern C++20**.
- Implemented **reverse-mode automatic differentiation** as measured by gradient-tested tensor operations and end-to-end training examples, by building a dynamic computation graph with topological backpropagation for arithmetic, reductions, **matmul**, **softmax**, and neural-network activations.
- Developed a tested **neural network and classical ML library** as measured by **50 GoogleTest tests**, XOR learning, multiclass classification, and regression/clustering examples, by implementing **Linear**, **Sequential**, **SGD**, loss functions, accuracy metrics, **linear regression**, **logistic regression**, **k-means**, and **k-nearest neighbors**.

UofTGuessr | *Java* October 2024 – December 2024

- Built a Java Swing GeoGuessr-style application structured with **Clean Architecture**, separating **entities**, **interactors**, and **adapters** for maintainability and scalability.
- Implemented **real-time multiplayer** via **raw sockets** with matchmaking and round synchronization, and integrated **Firebase Realtime Database** for **authentication**, **account management**, and **live stat tracking**.
- Designed a **distance-based scoring pipeline** using the **Haversine formula**, added **community photo ingestion** through an **Imgur/OkHttp integration**, and ensured reliability with **unit and integration tests** in **JUnit/Mockito**.