

Gubin Geramifard

647-787-8279 | geramifg@gmail.com | linkedin.com/in/GubinGeramifard | github.com/GubinGeramifard

EDUCATION

Athabasca University

Bachelor of Science in Computer Science, GPA: 3.3

Athabasca, AB

Expected April 2027

McMaster University

Honours Bachelor of Life Science

Hamilton, ON

2020

EXPERIENCE

Software Engineer Intern

ThePsychoEd Clinic

September 2025 to August 2026

Toronto, ON

- Built the clinic's report-generation engine: a multi-step Flask pipeline that ingests assessment documents and drafts clinical reports via the Claude API (Anthropic on Azure AI), with a chat interface for clinicians to refine output.
- Designed an automated verification step that cross-checks facts across documents, flagging date/age arithmetic errors, contradictions, and omissions by materiality, so clinicians catch discrepancies before generation.
- Engineered the async backend: background job processing with status-polling endpoints, client-info extraction, response parsing, and fallback logic across LLM calls.
- Built an automated speech-to-text pipeline (Python, Azure AI Speech Conversation Transcription) with speaker diarization for multi-speaker dialogue, integrated via Azure App Service, Power Automate, and SharePoint; benchmarked at ~98% transcription accuracy versus ~90% for Rev AI.

PROJECTS

SignD, Contract Explainer (RAG) | Cohere, FastAPI, Next.js, Supabase, pgvector

Aug 2026 to Sep 2026

- Built a retrieval-augmented (RAG) web app that reads leases and contracts, automatically flags risky clauses, and answers plain-language questions in any language with the exact source clause cited.
- Engineered the pipeline with Cohere embeddings, reranking, and generation over a per-user Postgres/pgvector store; benchmarking showed reranking improved top-1 retrieval accuracy by 17.5% and MRR@10 by 11.3% over embeddings alone (Wilcoxon $p = 0.0002$).
- Shipped multi-tenant authentication (Supabase) with per-user document isolation; deployed a FastAPI backend on Render and a Next.js/TypeScript frontend on Vercel, with a no-signup anonymous demo.

Tally, Double-Entry Ledger Engine | Go, Concurrency, Docker, GitHub Actions

Sep 2026

- Built a double-entry ledger engine in Go (standard library only) using a single-writer state machine, idempotency keys, and a write-ahead log with crash recovery; verified under the race detector with property-based and fuzz tests plus CI.
- Proved correctness with a live benchmark: a naive implementation lost millions of dollars to race conditions under concurrent load while the engine stayed exact at roughly 620,000 transfers per second; group commit made durable writes about 13x faster than fsync-under-a-lock.
- Scaled the write path across cores with a sharded engine, making cross-shard transfers atomic via a two-phase clearing protocol; deployed as a single-binary Docker service.

PriceMoose | Python, Flutter, FastAPI, SQLAlchemy, PostgreSQL, Riverpod

Jan 2026 to Apr 2026

- Built a cross-platform mobile app that scrapes and compares live grocery prices across major Canadian chains (Loblaws, Walmart, Sobeys, Metro), indexing 800+ products.
- Engineered a Python FastAPI backend with async web scrapers using the Flipp flyer API to extract real-time sale prices, persisted via SQLAlchemy ORM.
- Deployed the backend on Railway with PostgreSQL; the Flutter app targets Android and web with platform-aware API routing and Hive-backed local persistence.

TECHNICAL SKILLS

Languages: Java, Python, Go, C/C++, SQL (PostgreSQL), JavaScript, TypeScript, Dart, HTML/CSS, R

Frameworks: React, Next.js, Node.js, Express, FastAPI, Flask, Flutter, Three.js, Riverpod, JUnit

Developer Tools: Git, Docker, GitHub Actions, Azure (App Service, AI Speech), Google Cloud Platform, Supabase, Render, Railway, Vercel, Power Automate

Libraries: Cohere, pgvector, pandas, NumPy, Matplotlib, SQLAlchemy