

# Jonathan Thomas

Toronto, ON | 365-341-2579 | [Jonathanthomasv6@gmail.com](mailto:Jonathanthomasv6@gmail.com) | [LinkedIn](#) | [GitHub](#)

## EDUCATION

### Toronto Metropolitan University

Expected Dec 2027

*Bachelor of Science (Honours) in Computer Science*

*Toronto, ON*

- **Relevant Coursework:** Software Engineering, Database Systems, Artificial Intelligence, Data Structures and Algorithms, Machine Learning, Computer Security, Web Systems Development, Operating Systems

## EXPERIENCE

### Software Engineering Intern

Jan 2026 to Aug 2026

*Ontario Public Service*

*Toronto, ON*

- Automated multi-step enterprise processes using AI agent workflows and n8n, reducing manual work by over **70%**
- Implemented an orchestrator agent for an enterprise **RAG** chatbot, reducing token usage and retrieval calls by **60%** through intent-based routing, with changes deployed to production
- Developed **C#** and **Oracle** functionality to link functional requirements (FRs) to business rule changes, designing a new relational table and modifying database triggers to enable end-to-end requirements traceability
- Improved chatbot reliability through citation validation, context retention, and prompt tuning, validated across **1000+** email queries using Playwright

## PROJECTS

### Wallex | *Java, Spring Boot, PostgreSQL, Docker, React*

- Built a personal finance application for managing transactions, importing bank statements, and tracking spending
- Developed a PDF bank statement pipeline to extract transaction data into a spending dashboard
- Implemented **REST APIs** for transaction CRUD operations with validation and PostgreSQL storage
- Added Docker containerization, **CI/CD** workflows on GitHub Actions, and backend tests with MockMvc and JUnit to automate builds and test transaction APIs

### KV Server | *Rust, TCP, Concurrency*

- Built a multi-client Rust key-value server with a custom **TCP protocol**, parser, and in-memory storage
- Implemented append only logging, checksums, snapshots, crash recovery, and log compaction for durable storage
- Designed concurrent request handling with worker threads, bounded queues, backpressure, and graceful shutdown

### Recipe Hub | *React, FastAPI, PostgreSQL, pgvector, LangChain, LangGraph*

- Built a RAG recipe recommender that matches user ingredients and cooking requests across **13,000+** recipes
- Embedded user queries with Qwen3 and retrieved relevant recipes using pgvector **similarity search**
- Used LangChain and LangGraph for recipe search, follow up questions, and conversation context
- Built batch embedding and indexing workflows and evaluated retrieval quality using Recall@K

## TECHNICAL SKILLS

**Languages:** Java, Python, C#, SQL, Rust, C++, JavaScript, Bash

**Technologies:** Spring Boot, .NET, FastAPI, React, PostgreSQL, Oracle, pgvector, LangChain, LangGraph

**Developer Tools:** Docker, Git, GitHub Actions, AWS, Azure, Linux, Playwright

## ACTIVITIES

### SolutionHacks Winner – Best Use of MongoDB

June 2025

*BridgeAI Project*

- Built an AI career platform for immigrants that parses resumes and generates qualification plans, placing among 200+ hackathon participants
- Implemented JWT authentication, PDF resume processing, progress tracking, and MongoDB backed user profiles