Ma Toan Bach

bachmatoan2910@gmail.com | linkedin.com/in/matoanbach | github.com/matoanbach | mbach@myseneca.ca

Education

Seneca Polytechnic — Toronto, ON
Bachelor of Engineering in Software Engineering — *Sept 2022 – August 2026*

Experience

Research Assistant — May 2025 – Present
Seneca Applied Research — *Toronto, ON*

- Designed and deployed an automated firmware unit-test generation workflow using Python, LangChain/LangGraph and FastMCP, cutting manual setup time by 90%.
- Architected and containerized a microservices test-generation FastMCP server to expose 12+ test tools via an internal API platform, enabling parallel and sandboxed test runs with robust concurrency controls.
- Implemented a RAG pipeline and a parser that indexed over 4,000 code symbols and knowledge-base records, boosting unit-test coverage from 84% to 99% while reducing token usage by about 32%.
- Integrated end-to-end observability with OpenTelemetry, Grafana, Tempo and Loki, decreasing debugging time by 30% and improving reproducibility.
- Partnered with AMD firmware engineering teams to refine prompts, guardrails and platform features, improving AI-generated test quality and driving adoption across the organization.

AI Engineer Intern — December 2025 – April 2026
Pathway Communications — *Toronto, ON*

- Built an AI browser automation system to extract and capture information from third-party portals using Playwright, Typescript, OpenCode SDK with sandboxed execution, Jenkins CI/CD and Docker for containerization, reducing manual coding time by 96%.
- Assisted in testing AI automation workflows for ticket processing in production OTRS system using the Google Agent Development Toolkit and Python.
- Built a Django-based observability dashboard to support debugging and testing of AI automation workflows; collaborated with ITSM, DevOps teams and stakeholders to gather feedback, ensuring alignment with project requirements and goals.
- Architected and maintained a FastMCP server platform that exposed database tools across MySQL, PostgreSQL and SQLite for AI agents to query client information.
- Integrated and configured Langfuse, Grafana, Prometheus and OpenTelemetry to monitor LLM agent workflows and MCP server metrics to track cost, latency, token usage and error rates, improving reliability and speeding up troubleshooting.

Certifications

Certified Kubernetes Application Developer (CKAD)	Issued: 2025-01 — Verify
HashiCorp Certified: Terraform Associate (003)	Issued: 2025-10 — Verify
Red Hat Certified System Administrator (RHCSA)	Issued: 2025-09 — Verify
Red Hat Certified Engineer (RHCE)	Issued: 2026-04 — Verify

Technical Skills

Languages: Python, Go, TypeScript, SQL
DevOps: Linux, RHEL, Proxmox, Bash, Docker, Kubernetes, Ansible, Terraform, GitHub Actions, Jenkins
Observability: Langsmith, Langfuse, Grafana, Prometheus
Cloud: AWS (EC2, EKS, ECR, IAM, S3, RDS)
Full-stack: FastAPI, FastMCP, OpenAPI, REST API, gRPC, Gin, React, Tailwind CSS, HTML
Databases/AI: LangChain/LangGraph, Google ADK, ChromaDB, PostgreSQL, MongoDB, DynamoDB, OpenAI models

Projects

datatrace | *TypeScript, React, Next.js, AWS (EKS/ECR), Docker, Kubernetes, GitHub Actions* *Link, GitHub*

- Collaborated in a 5-person hackathon team to prototype a food-industry analytics + POS dashboard for inventory, sales trends, and near-expiry alert workflows.
- Led the MVP front end in Next.js/React, shipping company vs. supplier views, KPI dashboards, certification workflow UI, and PDF report generation.
- Implemented CI/CD with GitHub Actions to build and push Docker images to AWS ECR and deploy the container to Amazon EKS with Kubernetes manifests and ingress.

leetyap.com | *TypeScript, React, Next.js, Tailwind CSS, Auth0, CodeMirror, OpenAI Realtime* *Link, GitHub*

- Built a solo LeetCode-style interview practice platform with an in-browser problem workspace (problem view, Python editor, and execution output) plus timers and settings.
- Integrated an AI voice interviewer using OpenAI Realtime to guide users through a structured interview flow (approach, implementation, testing, and complexity review).
- Implemented authentication with Auth0 and connected client-side code execution via Judge0, deploying the web app to Vercel with GitHub-based deployments.

Homelab | *Proxmox, RHEL, Networking, Terraform, Ansible, Kubernetes, OpenShift, Jenkins* *GitHub*

- Built and maintained a personal Proxmox-based homelab (64GB RAM, dual Xeon E5-2660 v3) to practice infrastructure, systems, and automation through repeatable build–break–fix cycles.
- Provisioned and rebuilt Linux VMs to practice Red Hat/RHCSA-style administration and troubleshooting (users/permissions, systemd services, storage/filesystems, SSH, and DNS/network fundamentals).
- Planned and documented a compact OpenShift-on-Proxmox deployment using the agent-based installer, including static IP/DNS mapping and Terraform-driven provisioning workflows.
- Explored Kubernetes cluster operations (Kubespray, Helm) and CI/CD experiments with Jenkins, capturing configs and lessons learned as a reusable knowledge base.

simple-bank | *Go, Gin, PostgreSQL, sqlc, JWT, Docker, GitHub Actions* *GitHub*

- Built a Go (Gin) REST API simulating core banking workflows—user registration/login, account management, and money transfers—backed by PostgreSQL.
- Implemented secure authentication with bcrypt password hashing, JWT middleware, session records, and account-ownership checks for protected resources.
- Designed transactional transfer logic that records transfers and accounting entries and updates balances atomically, including stable ordering to reduce deadlock risk under concurrent transfers.
- Added database-backed automated tests and GitHub Actions CI that starts PostgreSQL, runs migrations, and executes the test suite on every push.

LinkedIn Job Bot | *JavaScript, Node.js, Playwright, OpenAI API, DynamoDB, Express, Discord Webhooks* *GitHub*

- Built a LinkedIn job monitoring bot that scrapes new internship/co-op postings and delivers shortlisted roles to Discord channels for student job hunting.
- Automated LinkedIn login and UI-based job filtering with Node.js + Playwright, extracting job title, company, location, and apply URLs from dynamic pages.
- Applied OpenAI-based validation to filter for CS-related internship/co-op roles in English before publishing to Discord.
- Implemented deduplication with DynamoDB using a 14-day TTL to suppress repeats and broadcasted approved jobs to two Discord channels serving 144 students.

turtle programming language (Go) | *Go, GitHub Actions* *GitHub*

- Implemented a small programming language in Go, including a lexer, Pratt parser, AST, runtime object system, and built-in functions.
- Built two execution engines: a tree-walking evaluator for interpreting the AST and a bytecode compiler that lowers AST nodes into instructions for a stack-based virtual machine.
- Implemented VM features including globals, call frames, closures with free-variable capture, and symbol resolution across global/local/builtin scopes, with unit tests across parser, compiler, and VM behavior.

HTTP/1.1 Server | *C++23, POSIX Sockets, pthreads, zlib, CMake, vcpkg* *GitHub*

- Built a lightweight HTTP/1.1 server from scratch in C++23 using TCP sockets, implementing request parsing, response construction, and routing for GET/POST endpoints.

- Implemented basic file upload and download support, using a configurable storage directory and appropriate success/error responses.
- Added response compression and multi-client request handling to improve performance and scalability under concurrent traffic.

DNS Server | *C++23, UDP Sockets, DNS, CMake* *GitHub*

- Built a lightweight DNS server in C++ that receives and responds to queries over UDP, implementing binary message parsing and serialization for DNS headers, questions, and answers.
- Implemented domain-name encoding and support for compressed names (pointer-based labels) to correctly parse real DNS packets and construct valid responses.
- Enabled the server to forward queries to a public DNS provider and return the resolved results, validating end-to-end behavior using `dig`.