Aryan Awasthi

Phone: +1 438-855-6936 **Email:** Aryanbvp.09@gmail.com LinkedIn GitHub Portfolio **Location:** Montreal, QC

Summary

AI engineer who builds and ships production agentic systems—MCP servers, multi-agent plugins, and LLM-powered automation—that connect frontier models (Claude, Gemini) to live enterprise data and tools. At CSC Generation, delivered AI integrations across multiple retail brands from rapid prototype to production deployment. Master of Applied Computer Science, Concordia University.

Education

Concordia University **August 2023 – May 2025**
Master of Applied Computer Science *Montreal, Quebec*

Amity University **July 2017 – June 2021**
Bachelor of Technology in Computer Science and Engineering *Noida, India*

Skills

Generative AI & Agents: Claude (Anthropic API, MCP), Gemini API, Google AI Studio, Model Context Protocol (MCP), multi-agent systems, LangChain, LangGraph, RAG, prompt engineering, RLHF, AI security auditing, local LLM inference (Ollama, MLX).

Automation & Integration: Python, n8n, Microsoft Graph API, Outlook OAuth2, SharePoint, Workato, Wrike, Slack API, Canva API.

Cloud & Data: Google Cloud (Gemini API, GCP), AWS (SageMaker, Lambda, S3), Supabase (Storage + Postgres), Docker, Redis, FAISS, Pinecone, Meilisearch.

Frontend & Tooling: React, Vite, TypeScript, shadcn/ui, Chart.js, Vercel, Claude Desktop Extensions (.mcpb), Git, GitHub Actions, MLflow.

Experience

CSC Generation (Sur La Table) **October 2025 – Present**
AI Automation Manager *Montreal, QC*

- Built and deployed a suite of **6+ MCP servers** (Google Ads, GA4, Meta Ads, DMP, Image Generation) packaged as the **slt-marketing-analytics Claude Desktop extension**—the self-serve analytics layer for marketing teams across **3+ retail brands**; also built **Repo-Knowledge** (TypeScript MCP + Meilisearch + OpenAI embeddings) for semantic codebase search.
- Architected a production **purchase-order automation pipeline** (n8n + Gemini PDF extraction + Microsoft Graph) processing **∼50–100 POs and 25–50 vendor replies daily**; orchestrated **6 event-driven sub-workflows** (PDF parsing, 1,988-vendor master lookup, SharePoint CRUD, reply handling) with Slack error alerting and a **70-case test plan** across 12 suites; deployed to production May 2026.
- Architected an internal **analytics platform** (React + Vite + shadcn/ui + Vercel) with a department-owned data pipeline (SharePoint → n8n → DMP → serverless API) and JWT email-domain auth; integrated **Anthropic Claude** to generate AI-written weekly narrative summaries per department after each ingest, with a config-only multi-department architecture.
- Ran an independent **AI security audit** of a vendor SaaS tool via direct MCP API access—reviewed **388 screenshots** and surfaced an active AWS Access Key ID, live session tokens, employee PII, internal SharePoint paths, and a critical architectural flaw ("redaction" was metadata-only; S3 screenshots never visually blurred); delivered a **40+ page formal report** to executive and security leadership before broader rollout.
- Led the **Scribe Optimize pilot evaluation** across 9 employee value chains, quantifying **∼1,780 hrs/year** of addressable Tier-1 workflow savings (Bloomreach 482 hrs, email campaigns 363 hrs, SMS 289 hrs, PIM 170 hrs); designed a two-phase validation framework with a 1–5 scoring rubric across process accuracy and recommendation feasibility.
- Supported the **company-wide Claude AI rollout** at Sur La Table (Claude Day, March 2026)—delivered technical demos, ran cross-functional team training sessions, and led weekly AI office hours for Merch, Email, and Engineering as the primary CSC technical lead alongside executive stakeholders.

Projects

Red – Local LLM Assistant | *Ollama, llama.cpp, MLX, Apple Silicon, Python, MCP*

- Built a private AI assistant running **quantized open models locally** (Ollama / MLX on Apple Silicon), with tool-calling and **MCP** integrations so the assistant can act on local data and services without sending data to hosted APIs.
- Benchmarked models on **tokens/sec**, memory footprint, and tool-calling reliability under a 16 GB RAM constraint to choose the right model per task—practical experience with the cost, latency, and capability tradeoffs of self-hosted vs. hosted inference.

CLI LLM Council – Multi-Model Decision Stress-Testing | *Python, Claude, Gemini, Codex CLI, CLI orchestration*

- Built a **multi-model orchestration** workflow that routes a plan or decision to several frontier LLMs (Claude, Gemini, Codex) in parallel, then aggregates and contrasts their critiques to surface blind spots and stronger options.
- Designed the pipeline to expose where models **disagree**, using cross-model review as a lightweight evaluation layer before committing to a decision.

Multi-Agent LLM Chatbot with Reinforcement Learning | *Python, Llama 3, LangChain, FAISS, FastAPI, Streamlit, RLHF* | GitHub

- Built a **routing agent dispatching to specialist sub-agents** (hierarchical delegation), achieving **92% routing accuracy** and **87% response relevance** across diverse query types.
- Built a **RAG pipeline** with FAISS retrieval (**3× context speed**) and an RLHF reward model trained on **1,000+ feedback logs** that boosted engagement **35%** and cut latency **45%** via adaptive re-ranking.