

Aaryan V Bhalla

+1 (942) 444 1412 | avbhalla@uwaterloo.ca | bhalla.info | github.com/aaryanved | in/aaryanved

SUMMARY

Second-year Computer Science student at the University of Waterloo with hands-on production experience building AI systems. Currently working as an AI Engineer at Decimal Technologies, designing LLM agent pipelines, RAG systems, and workflow automation tools in Python. National hackathon winner, Anthropic-certified, and actively involved in the UWaterloo engineering and residence communities. Strongest in AI and ML engineering, full-stack development, and rapid prototyping.

EDUCATION

University of Waterloo <i>Bachelor of Computer Science (Honours, Co-Op)</i>	Sep 2025 – Present Waterloo, ON, CA
– Relevant Coursework: CS135 (Designing Functional Programs) CS136 (Elementary Algorithm Design and Data Abstraction) CS245 (Logic and Computation) CS246 (Object-Oriented Software Development) – Activities: Member, Computer Science Club Chassis Design Team, UW Formula Electric Undergraduate Residence Advisor, Campus Housing	
Scottish High International School <i>ISC - Computer Science, Physics, Chemistry, Mathematics, English, Hindi, Spanish</i>	Apr 2011 – May 2025 Delhi, India

SKILLS

Languages: Python, C++, C, JavaScript, TypeScript, Java, C#, Bash
AI & ML: LangGraph, LangChain, OpenAI API, Anthropic API, Google Gemini, Vertex AI, HuggingFace Transformers, TensorFlow, PyTorch, scikit-learn, FAISS, RAG pipelines, pgvector, Pydantic, prompt engineering
Frontend: React, Angular, TypeScript, JavaScript, Next.js, Three.js, Tailwind CSS, HTML, CSS, face-api.js, Web Audio API
Backend & APIs: FastAPI, Flask, Node.js, Fastify, REST APIs, SSE, asyncio
Applied Cryptography & Security: RSA-OAEP/PSS, AES-256-GCM, WebCrypto/SubtleCrypto, JWT (HS256), hybrid encryption protocol design, forward secrecy
Compilers & Systems: C++, LLVM, CMake, lexing/parsing, static analysis, bytecode interpreters, GoogleTest
Databases: PostgreSQL, SQLite, Redis
DevOps & Cloud: Docker, AWS, Vercel, Git, GitHub Actions, Jenkins, CI/CD, Linux, macOS, Windows
AI Dev Tools: Claude Code, Cursor, Codex, Antigravity
Data & Analysis: Pandas, NumPy, data pipelines, ETL, vector indexing
Languages Spoken: English (Native), Hindi (Native), Spanish (Working)

CERTIFICATIONS

Claude Code 101 Anthropic	2026
Model Context Protocol: Advanced Topics Anthropic	2026

EXPERIENCE

Artificial Intelligence Engineer Co-op <i>Decimal Technologies</i>	May 2026 – Aug 2026 Delhi, India
– Design and build AI-powered automation tools in Python - developing LLM agent pipelines using LangGraph and LangChain that replace manual multi-step workflows with production-grade automations with measurable efficiency gains – Build and maintain RAG systems end-to-end: data ingestion, embedding generation, pgvector indexing, and retrieval-augmented generation grounded in live knowledge bases integrated with PostgreSQL – Develop and optimize data pipelines and SQL queries, transforming structured and unstructured operational data into clean, queryable formats supporting data-informed decision making – Collaborate cross-functionally with developers and stakeholders to identify automation opportunities, translate requirements into technical solutions, and communicate findings clearly to both technical and non-technical audiences	
Undergraduate Residence Advisor <i>University of Waterloo</i>	Sep 2025 – Apr 2026 Waterloo, ON, CA
– Supported a residential community of undergraduate students as a first point of contact for academic, personal, and community concerns - developing strong skills in conflict resolution, crisis response, and peer mentorship – Organized and facilitated community programming and events, collaborating with housing administration and fellow advisors to maintain a safe and inclusive residence environment	
Founder and Executive Director <i>SHISTECH</i>	Apr 2021 – Apr 2025 Delhi, India
– Co-founded and, after the founding Executive Director exited early, took over as sole Founder and Executive Director of SHISTECH, a Delhi-NCR high school technology symposium launched online during COVID and later run in-person - grew from 1,100+ to 6,000+ annual participants across 40+ schools in India and Oceania over four years, running 10+ event categories spanning competitive coding, robotics, hackathons, esports, graphic design, and music production	

- Led a 70-member organizing team across four domains (IT, Social Media, Design, Marketing) and structured a zero-participant-fee model funded by sponsorships from GitHub, Replit, Taskade, .xyz, Zebronics, Stellar, Gradusure, and Athena totaling CAD \$490,000+ over three years, growing cash prize pools from CAD \$490 to CAD \$1,600; marketing efforts drove 100,000+ online views in a single week at points
- Built and maintained the entire technical stack as sole engineer - full-stack systems in Node.js and JavaScript, registration and participant databases, and relational database design supporting the event's scale
- When a from-scratch gaming lab suffered widespread hardware failures two days before the event's first offline esports tournament, secured approval to work six hours past facility closing and led the team overnight diagnosing and rebuilding machines so the tournament ran on schedule the next morning

Web Development Intern

Jun 2023

Stellar Information Technology Pvt. Ltd.

Delhi, India

- Developed and maintained web application features in a professional software environment, gaining early exposure to production development workflows and client-facing web delivery

ICT Infrastructure and Community Technology Contributor

2023

Bagiya NGO

Delhi, India

- Designed and built an E-Library and established ICT Labs across two NGO branches, making educational technology accessible to underserved children
- Conducted stakeholder discovery with community members and leadership before specifying any technology, delivering tools that served real community outcomes rather than assumed ones

PROJECTS

Order Book Matching Engine | *Price-Time Priority, Distributed Systems Design, Zero-Knowledge Settlement* Aug 2026

- Designed a price-time priority matching engine architecture: a sorted structure over distinct price levels for $O(\log n)$ best-price lookup, a FIFO queue per price level for $O(1)$ time-priority matching, and an order-ID-to-node hashmap for $O(1)$ cancellation
- Modeled the full order lifecycle - validation, insertion, cross detection, partial and full fill handling, and cancellation - with explicit state transitions for each outcome
- Mapped the design onto a real off-chain sequencer / zero-knowledge prover / on-chain settlement architecture, reasoning through the trade-offs between off-chain matching speed and the trust model a zk-proof and L1 settlement layer remove
- Documented the full system as a technical architecture diagram to communicate design decisions and data structure trade-offs clearly

BagCheck – Computer Vision Cabin-Baggage Sizing System | *Python, PyTorch, YOLO11, OpenCV, FastAPI, SQLite, Open3DAu*

- Built a computer-vision system that measures hand-luggage dimensions from a single camera and flags oversized bags against airline cabin limits - ~8,300 lines of production Python plus 1,500 lines of tests (134 passing) across a segmentation/pose pipeline, three interchangeable measurement backends, and a FastAPI web app
- Designed three swappable dimension-estimation backends behind one shared interface (depth, reference-object homography, and monocular 3D pose recovery) chosen by camera geometry, so adding a fourth backend requires implementing one abstract class rather than touching the pipeline
- Solved full camera-pose recovery from a single image via homography decomposition (SVD re-orthonormalization, mirror-ambiguity resolution) and fit 3D bounding boxes by maximizing silhouette IoU with a scaled, restarted Powell optimizer - validated to ± 3 cm on a synthetic ground-truth harness
- Designed a three-verdict compliance model (compliant / oversized / needs manual check) that propagates measurement uncertainty into every decision, explicitly refusing to fail a bag on a number the system isn't confident about, backed by a SQLite audit log for full traceability

Clueless-ly – Real-Time AI Assistant Overlay | *TypeScript, Electron, React, OpenAI Realtime API, ScreenCaptureKit* 2026

- Built a cross-platform (macOS/Windows) Electron desktop app with a strict main/renderer/preload process boundary - the renderer never touches Node or OS APIs directly, only a typed IPC surface exposed via `contextBridge`
- Designed a platform-contract abstraction so all OS-specific behavior (permissions, window management, audio capture) is isolated behind a single adapter-resolver file, keeping 100% of feature code identical across macOS and Windows
- Integrated real-time transcription of microphone and system audio (via ScreenCaptureKit) alongside on-demand screen-context capture, streamed through OpenAI's realtime and vision APIs to answer questions in context
- Implemented OS-level content protection (`NSWindowSharingNone` / `WDA_EXCLUDEFROMCAPTURE`) so the overlay is excluded from screen recordings, with secrets never logged and API keys stored via Electron `safeStorage` backed by the OS keychain

KnightVision – Autonomous Chess Engine & Lichess Bot | *Python, python-chess, Stockfish (UCI), threading, REST/NDJSON stre*

- Built an autonomous chess-playing bot integrating the Lichess Board API and a local Stockfish engine over UCI - streaming live game state via NDJSON long-polling, converting FEN/UCI positions, and submitting moves through a multi-threaded seek/play/log pipeline with dedup locking to prevent race conditions
- Designed a human-latency model layering time-control-aware statistical delay distributions (uniform and right-skewed triangular) on top of engine move selection, so move timing reflects realistic human think-time rather than instant engine responses
- Built a 410-line opening book across 16 named repertoires as a flat UCI-move-history-keyed dictionary for $O(1)$ lookup, with graceful fallback to engine analysis when a position falls outside the book

- Instrumented the system with structured, append-only logs (games, moves, results, API calls, errors) for full post-hoc traceability, and identified and documented a platform-policy compliance gap (Board API vs. the sanctioned Bot API) as part of a full architecture review

Melodex – AI-Powered Spotify Intelligence Layer | *TypeScript, Node.js, Fastify, React, PostgreSQL, OpenAI, Vercel, Supabase* 2026

- Architected a full-stack pnpm monorepo (Fastify backend, React/Vite frontend, shared type-contract package) deployed as a single serverless function on Vercel with Supabase Postgres - designing around hard function timeouts and a shared connection budget via checkpointed resumable jobs, idempotent writes, and fast-fail connection pooling
- Built a resumable Spotify sync engine as an explicit state machine: a persisted checkpoint advances through phases (liked songs, playlists, albums, artists) across multiple function invocations, with per-phase retry budgets, OAuth token lifecycle handling, and Retry-After-aware backoff
- Designed an OpenAI tool-calling agent grounded in real user data through structured tools (deterministic filters, not free-text matching) and a post-generation validator that cross-checks every claim against tool results, with all mutating writes gated behind a preview-confirm-undo workflow built on idempotency keys and recorded inverse operations
- Integrated both custom security SDKs in production - encrypting every sensitive API call end-to-end via the RSA/AES handshake protocol and securing sessions via the JWT/refresh-rotation library - and did substantial live production debugging (mis-mapped errors, silent upstream API schema drift) from logs

MiniLang – Compiler and Static Analyzer | *C++, LLVM, CMake, GoogleTest, ANTLR* Jul 2026

- Built a from-scratch compiler: hand-wrote a lexer and recursive-descent parser producing an AST, a type checker catching mismatches and undefined-variable errors at compile time, and a stack-based bytecode interpreter for execution
- Designed a static analysis pass detecting use-before-initialization, unreachable code, and integer overflow risks, modeled on tools like Clang-Tidy, and extended the project with an experimental LLVM IR backend for native code generation
- Wrote 100+ unit tests with GoogleTest and set up a CMake build with GitHub Actions CI so every commit was automatically tested; documented trade-offs (e.g., recursive-descent vs. table-driven parsing) throughout

Custom End-to-End Encryption & Auth SDKs | *WebCrypto, RSA, AES-256-GCM, JWT, Redis, Postgres* Jun 2026

- Designed and built a versioned hybrid encryption protocol layering application-level encryption on top of TLS, with a byte-identical wire format implemented across TypeScript, Python, and Java so any app in that stack can adopt it: an RSA-4096 OAEP/PSS handshake establishes an ephemeral per-session AES-256-GCM key with RSA-PSS server-signature verification for anti-MITM authentication, giving every request forward secrecy
- Implemented a browser handshake state machine with concurrent-request coalescing and a pluggable session-store strategy (In-Memory/Redis/Postgres) with typed expiry errors
- Built a companion stateless JWT auth library (HS256) with refresh-token rotation - single-use tokens cryptographically bound to their session to prevent replay - and httpOnly/secure/sameSite cookie hardening against XSS; shipped as semver-versioned npm packages with Jest/Vitest test suites

Custom Git-Backed CMS | *TypeScript, Next.js, React, GitHub REST/Git Data API* May 2026 – Present

- Built a headless CMS with no database - all content reads/writes go directly through the GitHub Contents and Git Data APIs, using the repository's git history itself as the content store
- Designed a draft/publish workflow modeled on git branching: edits land on an auto-created draft branch, a diff view lists pending commits with per-commit and bulk revert, and a publish action performs an API merge plus a Vercel deploy-hook rebuild
- Hand-wrote a LaTeX parser from scratch (no external libraries) that recursively resolves brace-nested macros and infers the source document's font package to render pixel-faithful HTML from raw $\LaTeX$ source, with append-only version snapshots on every save

Pulse AI | *Python, LangGraph, LangChain, OpenAI API, FastAPI, PostgreSQL, pgvector, Docker, AWS* May 2026

- Designed and built a production AI agent system using LangGraph - structuring multi-step reasoning as task graphs with defined triggers, routing logic, and failure handling, integrated with PostgreSQL and pgvector for structured and semantic retrieval
- Built a full RAG pipeline end-to-end: data ingestion, chunking, embedding generation, pgvector indexing, and retrieval-augmented generation that grounds LLM outputs in structured evidence
- Deployed on AWS with Docker and CI/CD automation, maintaining observability across latency, token usage, and retrieval quality to measure and improve performance in production

Grounded | *TypeScript, JavaScript, Gemini 2.5 Flash, ElevenLabs, face-api.js, Web Audio API* Mar 2026

- Built at Hack Canada 2026 in under 24 hours - a real-time multimodal AI application integrating computer vision (face-api.js), LLM reasoning (Gemini 2.5 Flash), and text-to-speech (ElevenLabs) into a single live pipeline with real-time state management

The Originality Engine | *Python, HuggingFace Transformers, FAISS, Pandas, NumPy* Feb 2025

- Designed and built a large-scale semantic analysis pipeline: data ingestion and preprocessing, embedding generation using open-source HuggingFace transformer models, FAISS high-dimensional vector indexing, and similarity-scored output with systematic benchmarking across multiple model architectures

Disease Detection | *Python, TensorFlow, Flask, scikit-learn, Pandas, Raspberry Pi, Linux* Aug 2023

- Trained a TensorFlow CNN for medical image classification end-to-end - data preprocessing and augmentation, architecture design, hyperparameter tuning, and held-out evaluation - deployed as a Flask REST API on Linux for real-time inference on a Raspberry Pi

AWARDS & ACHIEVEMENTS

SRCC FIC Fellowship Programme

2025

Selected for the Fellowship and Incubation Centre programme at Shri Ram College of Commerce, University of Delhi - one of India's most prestigious undergraduate institutions.

National Winner - IIT Delhi Youth Ideathon

2024

Placed first nationally at the Indian Institute of Technology Delhi's Youth Ideathon, competing against university teams across India in a technology and innovation challenge.

Certification of Distinction - FERMAT Mathematics Olympiad

Awarded a Certification of Distinction in the FERMAT Mathematics Olympiad, recognizing exceptional performance in advanced mathematics.

HACKATHONS & COMPETITIONS

HackTrack

2-time consecutive winner

Scottish High International School

ShriTeq Hackathon

3-time participant (1-time winner)

Shri Ram School, Moulisari

Hack Canada

Participant - Built Grounded

Wilfrid Laurier University

- Designed and shipped a fully functional multimodal AI application in under 24 hours - integrating computer vision, LLM reasoning, and real-time text-to-speech into a unified live pipeline

Ctrl + Hack + Del 2.0

Participant

York University

CXC Data Science Hackathon

Participant

University of Waterloo