

Aaryan V Bhalla

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

EDUCATION

University of Waterloo

Bachelor of Computer Science (Honours, Co-Op)

Sep 2025 – Present

Waterloo, ON, CA

AI & MACHINE LEARNING SKILLS

Frameworks: PyTorch, TensorFlow, HuggingFace Transformers, scikit-learn, YOLO11, MediaPipe, OpenCV

Agents & LLM Systems: LangGraph, LangChain, Google ADK, OpenAI API (Chat Completions, Realtime, Responses), Anthropic API, Google Gemini, Vertex AI, tool-calling agents, multi-agent orchestration, agent evaluation & guardrails, prompt engineering

Retrieval & NLP: RAG pipelines, embedding generation, FAISS, pgvector, semantic similarity search, sentence embeddings, tokenization

Modelling & Evaluation: CNN architecture design, training & hyperparameter tuning, held-out evaluation, benchmarking across model architectures, numerical optimization (Powell, homography/SVD), real-time inference profiling

Speech & Multimodal: Realtime speech-to-text, TTS (ElevenLabs, Google TTS), vision-language grounding, pose estimation, audio pipelines (PCM16, Web Audio API)

Data & Infrastructure: Python (pandas, NumPy), PostgreSQL, data pipelines, FastAPI, Flask, Docker, AWS, GCP (Cloud Run, Firestore, Vertex AI), Git, GitHub Actions (CI/CD)

EXPERIENCE

Software Engineer

Xsolla

Sep 2026 – Present

Los Angeles, CA, USA

- Building on the payments infrastructure behind Roblox, Epic Games, Rockstar Games, Valve, and Twitch

Artificial Intelligence Engineer

Decimal Technologies

May 2026 – Aug 2026

Delhi, India

- Designed and built AI-powered automation tools in Python - LLM agent pipelines using LangGraph and LangChain that replace manual multi-step workflows with production-grade automations
- Built and maintained RAG systems end-to-end: data ingestion, embedding generation, pgvector indexing, and retrieval-augmented generation grounded in live knowledge bases integrated with PostgreSQL
- Developed and optimized data pipelines and SQL queries, transforming structured and unstructured operational data into clean, queryable formats supporting data-informed decision making
- Evaluated where a model genuinely improves a workflow against where it adds unneeded complexity, documenting the reasoning behind each decision so results stayed reproducible and reviewable

AI & MACHINE LEARNING PROJECTS

Weaver – Conversational Multi-Agent Workflow Builder | Python, Google ADK, Cloud Run, Firestore, Vertex AI, Gemini

- Built a conversational multi-agent system solo in 10 days for Google's "All Things Agentic" hackathon: 15 Google ADK agents across 16 private Cloud Run services, ~12,250 lines, 87 automated tests, deployed to production and used by real users
- Designed a two-agent architecture separating user interviewing from system design, with a deterministic, model-free validator between them catching dependency cycles, dangling references, and unknown step types - on the principle that asking a model to check its own work is a second opinion from the same source, not validation
- Grounded agent capabilities in a live Firestore registry rather than prompt content, so adding video (Veo 3.1) and music (Lyria) generation required a database row and one service each with zero prompt changes
- Applied deliberate model tiering by task complexity (Gemini 3.7-flash for open-ended interviewing, 3.5-flash for structured DAG generation, 3.5-flash-lite for bounded single-purpose steps), and diagnosed a silent-failure class where the orchestrator reported HTTP 200 as success while the underlying action had failed, by parsing raw tool-response objects instead of model-generated prose

Verte – AI Sustainability Shopping Assistant | JavaScript, React, Node.js, Express, Snowflake, OpenAI (GPT-4o-mini, Whisper), Chrome MV3

- Placed 2nd at PivotHacks (12-hour, team of three): a Chrome Manifest V3 extension and Node/Express backend that surfaces secondhand and lower-carbon alternatives on Amazon and Best Buy product pages
- Led the architecture refactor (~2,000 lines) that consolidated the team's parallel workstreams into the final codebase, built around four agreed data contracts so mid-event pivots changed the data producers without touching the UI
- Built an anti-hallucination verification pipeline: every GPT-4o-mini product suggestion is checked against live Amazon.ca search results via a custom HTML parser and token-based matching that rejects sponsored listings and accessories, with budgets enforced against real prices and a retry loop that tells the model what was unavailable - nothing unverified ever reaches the user
- Built hands-free voice search with OpenAI Whisper and a Web Audio silence detector that calibrates to the room's noise floor and stops recording ~1.2s after speech ends, skipping the transcription call entirely when nothing was said

- Integrated Snowflake (key-pair JWT auth) as the source of a 24-category buy-used knowledge base with row validation, caching and a built-in fallback table, and wrote a rate-limited Amazon.ca scraper (custom HTTPS client, gzip, concurrency queue, 30-minute cache)

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

- 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

BagCheck – Computer Vision Cabin-Baggage Sizing System | *Python, PyTorch, YOLO11, OpenCV, FastAPI, SQLite, Open3D*

- Built a PyTorch segmentation and pose pipeline measuring hand-luggage dimensions from a single camera - ~8,300 lines of production Python plus 1,500 lines of tests (134 passing), with three interchangeable estimation backends behind one shared interface, selected by camera geometry rather than preference
- 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 projected-silhouette IoU with a scaled, coarse-swept, restarted Powell optimizer - validated to ± 3 cm against a synthetic ground-truth harness
- Designed a three-verdict compliance model 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

VFC – Virtual Fighting Championship | *TypeScript, MediaPipe, three.js, vitest*

- Engineered a real-time computer-vision pipeline (MediaPipe pose landmarker) classifying boxing techniques from live webcam video entirely client-side, holding detection accuracy across camera frame rates from 15-60fps - roughly 2-3 samples per event at the low end - through empirically tuned adaptive (One Euro) filtering
- Diagnosed accuracy regressions by direct before/after measurement rather than intuition, including a filtering bug silently discarding 65% of peak velocity, with parameter sweeps plotting signal fidelity against noise rejection and regression tests written to fail against the prior wrong values
- Built an adaptive performance governor that profiles live model inference cost and paces inference frequency against measured device capability with hysteresis, plus confidence-aware tracking that checks landmark confidence continuously through an event rather than only at onset, since lost landmarks scatter rather than freeze and can otherwise score as exceptionally fast motion

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

- Built dual-channel real-time transcription - microphone and system output audio captured separately and transcribed over independent OpenAI Realtime WebSocket sessions for speaker-attributed conversation history - with cross-stream echo suppression solved at the text layer via normalization (Latin and Devanagari) and a three-part similarity test
- Engineered a multi-source grounding pipeline combining speaker-attributed transcript, a live full-resolution screenshot passed to a vision model, and a persistent user reference library, after finding that pre-summarizing the screen into text lost exactly the details that mattered
- Built a priority-ordered, question-aware token budgeter deciding which reference material survives a fixed context ceiling (labeling anything truncated), and fixed a failure where concurrent auto-answers exceeded the provider's tokens-per-minute limit and hung every request, replacing it with a serialized, capped queue with per-request cancellation preserving partial output

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

- 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 actual tool results before responding
- Gated all mutating agent actions behind a preview-confirm-undo workflow built on idempotency keys and recorded inverse operations, so the agent can take real actions on user data without requiring blind trust

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

- Designed and built a large-scale semantic analysis pipeline over arXiv abstracts: 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
- Built QA into every pipeline handoff - each stage validated against expected distributions before the next ran, with every configuration decision logged for reproducibility

Grounded – Real-Time Panic Intervention System | *TypeScript, Gemini 2.5 Flash, ElevenLabs, face-api.js, Web Audio API*

- Built at Hack Canada 2026 in under 24 hours - a real-time multimodal application reading facial stress through the webcam (face-api.js), reasoning over it with Gemini 2.5 Flash, and delivering spoken breathing guidance via ElevenLabs, integrated into a single live pipeline running entirely in the browser

Disease Detection | *Python, TensorFlow, Flask, scikit-learn, Pandas, OpenCV, Raspberry Pi*

- 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 for real-time inference on Raspberry Pi, optimized for low-resource settings

EcoPulse – Carbon Footprint Intelligence Platform | *TypeScript, PostgreSQL, DeepSeek, Google Vision*

- Built a carbon footprint tracking platform pairing LLM reasoning (DeepSeek) with image recognition (Google Vision), pulling live emissions data from the ClimaTiq, ElectricityMaps, and IATA APIs to turn everyday activity into measured environmental impact

ProTrack | *React, Node.js, SQLite, Gemini API, Chart.js*

- Built a productivity platform driven by natural-language conversation with a Gemini-powered assistant rather than forms, over a relational SQLite schema with SQL aggregation and reporting across task, habit, and calendar metrics

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

- Built an autonomous agent integrating the Lichess Board API with a local Stockfish engine over UCI, streaming live game state via NDJSON long-polling through a multi-threaded pipeline with dedup locking against race conditions from redelivered events
- 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

CERTIFICATIONS & ACHIEVEMENTS

Claude Code 101 Anthropic	2026
Model Context Protocol: Advanced Topics Anthropic	2026
National Winner, IIT Delhi Youth Ideathon - placed first nationally against university teams across India	2024