Aaryan V Bhalla

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

Education

University of Waterloo Expected Graduation: Apr 2030
Bachelor of Computer Science (Honours, Co-Op) – President's Scholarship of Distinction *Waterloo, ON, CA*

AI & Machine Learning Skills

Frameworks: PyTorch, TensorFlow, HuggingFace Transformers, scikit-learn, YOLO11, MediaPipe, OpenCV, LightGBM, MuJoCo, ONNX Runtime, PPO/reinforcement learning
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 Sep 2026 – Present
Xsolla *Los Angeles, CA, USA*

- Built and shipped an internal Systems Registry dashboard in React 19, TypeScript and Neo4j through about 20 production releases, tracking stewards, builders, operators, risk scores and outstanding actions for 50+ company systems
- Ran a column-by-column data-accuracy audit of the dashboard against its graph sources, finding 11 issues; fixed the 4 app defects and routed the data gaps to their owners
- Designed a rule-based red/yellow/green risk score from Slack, meeting and weekly-update activity; removing a false-positive signal took the dashboard from 48 of 51 programs green to an accurate 47 green / 3 yellow / 1 red split
- Found missing Slack thread replies in the company knowledge graph and drove a pipeline fix that refreshes reply counts, now running in production
- Built OneBox Viewer, a sandbox and recipe-catalog app with a Cloud Run back end that hides secrets from every response, tested with Playwright against recorded real data
- Designed the architecture for Prototype Overlays, a 6-phase system of chained sandboxes, and wrote a Go control-plane feature that saves sandbox runs to an experiments repository for review
- Shipped every change through a gated pipeline (preview, stakeholder approval, pull request, CI/CD, production check and rollback), building alongside AI coding agents while owning the specs and review

Artificial Intelligence Engineer May 2026 – Aug 2026
Decimal Technologies *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

Countersign – AI Payments Gatekeeper for One-Person Companies | *TypeScript, Next.js, Convex, Claude API, Cartesia, ECDSA*

- Won the Moss track and placed Top 20 in the Hyphenate track at Cal Hacks 13.0, built with a team of four in about 35 hours: AI agents take payment requests by phone, web chat and email, call the real vendor back on the number already on file, and only a rules engine can release money; I owned the Policy Engine, the case pipeline, the sandbox bank and the scenario answer key
- Achieved 0 false payments across a 39-scenario answer key (26/26 scams blocked, 13/13 legitimate bills paid or escalated) and 28/28 scenarios on a live deployed pipeline by designing an 18-rule deterministic TypeScript Policy Engine as the only path to PAY, with Claude limited to parsing requests and writing replies
- Rejected 4/4 forged, tampered or replayed transfers by building a sandbox bank that moves money only with an ECDSA P-256 countersignature over canonical JSON, single-use nonces and a 2-minute expiry, plus import-guard tests that let only one module touch the signing key

- Fixed pipeline runs that died silently in a live rehearsal by building an idempotent Convex step function with a watchdog (90 s, at most 2 restarts, then fail closed) and race-based LLM timeouts, and wrote the Policy Engine's 53 tests and 27 pipeline integration tests in a 1,126-test CI suite

Deja – Real-Time Free-Trial Abuse Decision Service | *Java 21, PostgreSQL, LightGBM, Python, TypeScript, Next.js*

- A service that catches people farming free trials with multiple accounts: it links accounts that share a device, card, phone or email pattern into identity clusters as they form, scores them with a machine-learning model and rules, and returns allow, challenge, review or block in milliseconds; built solo, ∼34,000 lines across Java, Python and TypeScript with 485 passing tests
- Made every live decision reproducible exactly offline - a 100% match on 100,000 events from 32 concurrent clients, identical on x86 and ARM - by running one pure-function engine against Postgres online and in memory offline, with lock-ordered transactions that commit interacting events in log order
- Cut expected abuse cost per 1,000 sign-ups by 68% against a rules-only baseline in backtests on simulated data (PR-AUC 0.78, 84% recall at 1% false positives) by training a calibrated LightGBM model on identity-graph features that more than doubled PR-AUC over per-event features alone
- Hardened it for production before deployment, verified by 21/21 failure modes passing under fault injection, 90,800 generated API contract tests with 0 failures, and 300+ dependency vulnerabilities cut to 0, by building the checks into CI on every push

ApexTrace – F1 Safety-Warning Simulator | *TypeScript, three.js, Web Workers, Python, PyTorch*

- A Formula 1 driving simulator with a safety system on board that tells the driver when to brake, built by a team of three in 48 hours that won 1st place at Formula Tech Hacks 2026: you drive real-layout Monza and Baku in the browser, while the bench deliberately breaks the telemetry link to test whether the warning still arrives in time
- Built the core simulator and safety logic - 2026-regulation car dynamics (bicycle model, Pacejka-style tyres, active aero, hybrid power unit), the look-ahead BRAKE warning that never trusts stale data, and a seeded fault-injection test runner - leading the team with 62 of 111 commits
- Found 3.5× more safety failures than random testing on the same 50-simulation budget (47.0 vs 13.6) by training a PyTorch TCN that forecasts lap failure from the first 12 s of telemetry (ROC-AUC 0.918) and using it to screen scenarios for a Soft Actor-Critic search
- Took the app to zero infrastructure after the event, deleting 26.9k lines of backend, by porting the Python simulator line for line to TypeScript in a Web Worker and proving it tick-for-tick against recorded Python sessions

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

Marionette – Real-Time Webcam Motion Capture Driving a Simulated Humanoid | *TypeScript, Python, Go, MuJoCo, PyTorch, ONNX Runtime Web, MediaPipe, three.js*

- Built a 14,790-line browser app where a MuJoCo-simulated Unitree H1 humanoid copies your upper body from a webcam in real time, entirely on-device: pose estimation, 2D-to-3D lifting, retargeting and IK in a WebWorker, and 500 Hz control and physics on the main thread
- Trained a 1.1M-parameter residual MLP that lifts 2D landmarks to 3D on 1.04M CMU mocap poses, using a randomised virtual camera as augmentation; 74.8 mm MPJPE on 40,000 held-out-subject samples, measured on the exported ONNX model, at 0.3–0.5 ms per inference in the browser
- Wrote closed-form retargeting from human limb axes to H1 joints, verified against MuJoCo forward kinematics to under 10^{-6} on 200 sampled configurations, plus damped least-squares IK that cuts hand position error 28% (83.6 vs 116.7 mm) in 2.3 iterations on average
- Measured that H1 cannot stay upright on planted feet while its arms follow a person (centre of mass 12 mm from the support edge, falls in 3.2 s), then added centre-of-mass feedback inside the 500 Hz physics loop - stable at 500 Hz, unstable at 50 Hz - reaching 83% push recovery on a disclosed support rig that applies 0 N when undisturbed
- Cut knee sag 10× (0.206 to 0.021 rad) with a contact-aware gravity feed-forward, low-pass filtered after finding that the raw term formed a positive feedback loop with ground contact and made the robot fall faster than no feed-forward at all
- Trained a PPO residual balance policy over 37M CPU-only environment steps with domain randomisation; it beat PD alone 3.2× on push recovery but added only 4.4 points over the classical controller against a 5-point bar, so it ships switched off, after fixing clip-fraction, reward-scale and truncation-bootstrap bugs found in the diagnostics
- Held 19 ms p95 camera-to-screen latency and 500 Hz physics even at 8× CPU throttle, cut cold load from 4.34 s to 2.85 s by removing a duplicate 13.6 MB WASM download, and compiled a mesh-free physics model from 56 MB to 51 KB with 0.0 measured dynamics divergence

- Backed it with 110 unit and 9 Playwright end-to-end tests in CI, including an automated check that no request leaves the origin during a live camera session, and cross-language fixtures that caught a gravity-projection bug (matrix column instead of row) before it reached training

Crucible – Compute-in-the-Loop Regression Testing for Autonomous Driving | *C++20, Go, Python, TypeScript, NATS JetStream, PostgreSQL, ONNX Runtime, React*

- Built a regression-testing platform for self-driving planners solo (∼12,000 lines across four languages) that runs each planner as a CPU-quota'd OS process under an enforced millisecond deadline inside a deterministic simulator, then decides with paired statistics whether a code change made driving better or worse - live at crucible.bhalla.info
- Achieved bit-exact reproducibility across four platforms - 0 fingerprint mismatches over 40,000 runs on macOS/arm64, glibc, musl and emulated x86-64 - by writing an IEEE-exact math library, eliminating compiler-dependent argument-evaluation order, and quantising measured planner latency into the virtual clock
- Simulated driving at 501× real time per core (1,904× across 10 cores) with a zero-allocation tick loop enforced by a test that overrides `operator new`, and a POSIX shared-memory SPSC ring at 2μs p99 round trip holding 100 ms deadlines to 1.037 ms p99 overshoot
- Designed the statistics that call a regression: 10,000-sample paired bootstrap over 11 driving metrics with Benjamini-Hochberg false-discovery control and a sequential probability ratio test reaching the same verdict on 84% fewer scenarios, with power and false-alarm rate validated against injected regressions of known size rather than asserted
- Built the distributed execution path in Go - NATS JetStream work queue with leases and redelivery, content-addressed result cache (1,000 scenarios × 2 planner versions in 21.3 s cold, 0.6 s warm), idempotent Postgres writes, S3-compatible trace storage - and ran untrusted planner code under cgroup v2 quotas and a seccomp-locked container with an automated escape-probe test in CI
- Used the platform to find four real planner defects and one map-fidelity defect the metrics had been misattributing, each now covered by a regression test verified to fail when the bug is reintroduced

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*
1st Place, Formula Tech Hacks - built ApexTrace with a team of three *2026*
2nd Place, PIVOTHACKS - built Verte with a team of three *2026*
President's Scholarship of Distinction - University of Waterloo *2026*
Also: Google "All Things Agentic" | Hack Canada | 2× HackTrack winner | ShriTeq winner