

Yash Bhalgat

Los Angeles, CA | (949) 745-7320 | yashbhalgat9657@gmail.com | linkedin.com/in/yashnbhalgat | github.com/yash9657

EDUCATION

University of California, Riverside

Riverside, USA

Master of Science in Computer Science

September 2023 – March 2025

GPA: 3.76 / 4 · *Graduate Teaching Assistant* — taught Physics 2LA and Chemistry 1LA labs, 45 students per term.

Pune Institute of Computer Technology

Pune, India

Bachelor of Engineering, Computer Engineering

August 2019 – July 2023

GPA: 9.2 / 10

TECHNICAL SKILLS

Languages: TypeScript, Python, Go, SQL, JavaScript, C++

Computer Science: Data Structures & Algorithms, Object-Oriented Design, Software Architecture & Design Patterns, DBMS (indexing, query optimization, transactions), Distributed Systems, Concurrency, Operating Systems, Computer Networks

Backend & Data: Node.js, Fastify, Express, Django, FastAPI, PostgreSQL (Row Level Security), Redis, MongoDB, Supabase, REST, WebSockets, Pub/Sub, microservices, event-driven pipelines

Frontend: React, Next.js, TypeScript, XState, Redux, Tailwind CSS

Cloud, DevOps & Testing: GCP (Cloud Run, GCS, Vertex AI), AWS (Lambda, SQS, Aurora, CloudFront, EventBridge, S3, EC2), Docker, Kubernetes, Terraform, GitHub Actions CI/CD, unit / integration / end-to-end testing (Vitest, Playwright), code review, technical design docs, Agile / Scrum

AI & Healthcare Interoperability: LLM orchestration (Claude, Gemini, OpenAI), agentic tool use, RAG, MCP, evaluation harnesses; HL7 FHIR R4, SMART on FHIR, ICD-10-CM / CPT coding, HIPAA, SOC 2

WORK EXPERIENCE

Stealth Startup

Los Angeles, USA

Founding Engineer

December 2025 – Present

- Architected a multi-pass GPU captioning pipeline — vocal isolation, WhisperX transcription, forced alignment, speaker diarization, sound-event detection, LLM authoring with an automated judge — that resolves **song lyrics, non-speech audio cues and code-switched dialogue**. Cut feature-film captioning and CCSL from a **multi-day vendor cycle to one automated run**.
- Built a deterministic caption **QC engine** (reading rate, line length, broadcast timing) with context-aware auto-fixers over a **12-format conversion layer** (SCC, TTML/IMSC, iTT, EBU-TT-D, SRT, WebVTT), every file **verified against a real decoder before delivery** — removing the manual conformance pass and the formatting rejections it missed.
- Rebuilt pre-release video delivery off public long-lived links: private storage behind an **authenticated streaming gateway, DRM-protected CMAF playback (Widevine / FairPlay)**, short-lived signed sessions, **per-viewer forensic watermarking** and an audit record per access — clearing the content-protection review that had been **gating studio and distributor work**.
- Re-architected feature-length packaging from a sequential encode that exceeded the platform request limit into a **parallel chunked transcode with an idempotent fan-in barrier**, keyframe-aligned so quality tiers stay switchable — enabling reliable packaging and **adaptive streaming with instant seek**.
- Built the end-credits builder and render pipeline (headless-browser compositing through FFmpeg) delivering **broadcast-spec MP4, ProRes and DPX** at exact delivery frame rates, plus a film tax-incentive compliance panel — moving a **per-title outsourced cost in-house**.
- Led **SOC 2 (Security and Confidentiality) readiness**: row-level tenant isolation on every table, an append-only audit log of every content mutation, static and dependency analysis in CI, and a quarterly risk register — taking the platform to **Type I ready**.

- Built the **ambient clinical documentation pipeline** — encounter audio through streaming speech recognition with **medical-lexicon biasing**, speaker-role attribution across clinician, patient, caregiver and interpreter, then LLM structuring into a clinician-signed draft. Every assertion **cites its transcript span**, and an **assertion classifier resolves negation, hypothetical and family-vs-patient context** so “denies chest pain” never becomes an active problem. Notes are **signed in the room**, not after hours.
- Replaced fixed SOAP templates with a **note-format engine**: a clinician composes their own format section by section — or uploads an existing note as an example — and it **compiles to a typed extraction schema** with per-section grounding rules, each section tagged by **LOINC document type**. SOAP, APSO, H&P, procedure, **DAP/BIRP behavioral health** and PT/OT evaluations run on one engine; **a new specialty ships in an afternoon with no code change**.
- Built the coding engine that closed the gap between the visit and the bill: **ICD-10-CM, CPT and HCPCS Level II** suggestions, **E/M level from the 2021 medical-decision-making grid, modifier and NCCI bundling checks, diagnosis-to-procedure medical-necessity linkage**, and **HCC recapture prompts backed by MEAT evidence** — each one click to accept with its supporting quote, exported as a **superbill ready for an 837P claim**. Eliminated the manual code lookup where **undercoding and denials originated**.
- Built the EHR interoperability layer on **SMART on FHIR** — **SMART App Launch** (EHR, standalone, and **Backend Services with asymmetric JWT client authentication**), **granular v2 CRUDS scopes**, notes written as a **FHIR DocumentReference** alongside **discrete Condition, Encounter and charge resources**, over a **SNOMED CT → ICD-10-CM and RxNorm** terminology layer. A **FHIRcast** subscription keeps the app on the clinician's open chart.
- Where vendors gate write scopes behind marketplace review — or expose **no write API at all** — shipped a **Chrome extension (MV3) as a universal write layer**: least-privilege per-system permissions, short-lived scoped tokens, **no PHI stored in the extension**, and a per-field write trail attributing **every entry to a clinician, note and timestamp**. New practices went live in **minutes instead of waiting on an API partnership**.
- Owned the engineering half of HIPAA and SOC 2: **append-only logging of every PHI access and change** (§164.312(b)), minimum-necessary role access, MFA, **per-state recording-consent capture**, and **42 CFR Part 2 segmentation** for substance-use content — the company **passed SOC 2 Type I and Type II**.

PERSONAL PROJECTS

- **Inkcast** — Compiles topics and documents into **narrated vector lessons** through a seven-stage pipeline, synchronizing drawings to measured speech and rendering with Chromium and FFmpeg. A Next.js studio dispatches stateless render containers through private Blob storage and SSE; scene caching and editable storyboards support **local re-rendering**. TypeScript, Fireworks, Kokoro/ONNX, Whisper. **inkcast.video** | [github](#)
- **Tallyweb** — **Keyboard-first accounting and ERP** with double-entry constraints, versioned hash-chained vouchers, company authorization and **FORCE RLS** in PostgreSQL. Deployed the Go core on ARM64 Lambda with CloudFront, Aurora Serverless v2 and scheduled idempotent jobs. Scoped agent/MCP proposals require approval; live AI, document and statutory integrations remain unconfigured on AWS. React, XState, Terraform. **tallyweb.world** | [github](#)
- **Finfloww** — Built reconciliation, invoice-to-PO review and variance analysis with **deterministic financial calculations**, concurrent workflow execution and **human approval gates**. Bounded investigations cite original source rows, run generated JavaScript in an isolated QuickJS sandbox and record append-only events. Organization-scoped Supabase data, encrypted provider keys and Realtime updates; currently pilot software. Next.js, TypeScript, Fireworks. **finfloww.xyz** | [github](#)
- **DentaBill AI** — Built dental claim imports, queued follow-ups, **live transcripts and structured next actions**. Fastify APIs and SQS workers run on Lambda with Aurora, claim-state guards, idempotent summaries, retry backoff, dead-letter handling and **organization-scoped PHI audits**. The live demo runs the complete pipeline with simulated payer calls; live telephony remains gated. Next.js, Prisma, Clerk, Terraform. **usedentabill.xyz** | [github](#)
- **Cursor for Resolve** — Agentic AI inside a professional edit bay: video matting, promptable segmentation and an **on-device vision-language model so no footage leaves the machine**, with an agent driving DaVinci Resolve for dailies, screener exports and VFX pulls.