REDDY BALAJI MADHA

Backend Software Developer · Edge AI · Distributed Systems

Fremont, CA · 805-732-2532 · reddybalajimadha@gmail.com · LinkedIn · GitHub

SUMMARY

Backend software developer with 4+ years of experience building production APIs, AI-powered clinical data pipelines, automation tooling, and full-stack analytics platforms for multi-tenant EMS and healthcare environments. Delivered end-to-end systems in Java/Spring Boot and Python/FastAPI spanning NEMSIS 3.5.1 SOAP integration, AI-assisted PCR generation, edge AI deployment on ARM64 IoT devices, Snowflake-backed operational dashboards, and real-time Elasticsearch search infrastructure. Strong foundation in infrastructure automation, reliability engineering, and production operations in regulated environments.

CORE SKILLS

Languages: Java, Python, SQL, Bash, JavaScript
Backend: Spring Boot, FastAPI, REST APIs, JWT, JSON, XML, JAXB, SOAP, schema validation
Healthcare & AI: NEMSIS 3.5.1, HL7 v2, FHIR, ICD-10/RxNorm/SNOMED CT mapping, GPT integration, Whisper ASR, MedGemma
Edge AI & IoT: Whisper.cpp, YOLOv8, ONNX Runtime, MedGemma 4B, ARM64 deployment, Ericsson Cradlepoint R1900, Docker buildx/QEMU
Data & Analytics: Snowflake, Snowpipe, Elasticsearch, SQL, data modeling, clinical dashboard design
NLP & Audit: Sentence Transformers, semantic similarity, confidence scoring, audit automation, JSON artifact generation
Cloud & Infra: AWS (EC2, EKS, ECS), Docker, Kubernetes, GitHub Actions, Linux, CI/CD
Reliability: Structured logging, async workflows, retries/backoff, health checks, integration testing

EXPERIENCE

Atian.ai | Software Developer

Dec 2024 – Present · Fremont, CA

Edge AI Inference Platform (Cradlepoint R1900 · Whisper.cpp · MedGemma · YOLOv8)

- Architected and deployed an edge AI system on an Ericsson R1900 Cradlepoint router (ARM64, 4GB RAM) for real-time EMS medical documentation without cloud dependency — Whisper.cpp at 14s inference per 30s audio, YOLOv8 via ONNX Runtime at 750ms for scene awareness.
- Built a fully automated encounter pipeline — YOLO triggers start/stop on person detection, Whisper transcribes, and MedGemma 4B generates SOAP notes, PCR reports, and ED handoffs with zero manual input across a 17-task NLP pipeline.
- Implemented audio noise suppression (FFT-based filtering), medical terminology correction (dictionary + Whisper prompt biasing), heuristic speaker diarization, and benchmarked multiple models (MedGemma 4B, Gemma 4, Gemma 1B, YOLO-World) for edge accuracy vs resource trade-offs.
- Cross-compiled Docker containers for ARM64 using buildx/QEMU, managing image size constraints (500MB vs 2.5GB) on router storage. Stress-tested for 10 hours: 1,015 transcriptions at 103/hour with zero crashes.
- Built real-time dashboard with SSE streaming, SQLite persistence, and REST APIs connecting router AI to edge PC intelligence.

EMS Intelligence Analytics Dashboard (Snowflake · Python · FastAPI)

- Designed and built 'First Mile Intelligence Analytics' — a multi-tenant, multi-tab EMS operations platform deployed for live ambulance districts, surfacing real-time metrics across EMS operations, clinical intelligence, service area analytics, revenue & billing, and AI documentation scoring.
- Built the full backend data pipeline from EMS incident sources through Snowpipe-based continuous ingestion into Snowflake, powering dashboard KPIs including response time by priority, dispatch processing lags, fleet GPS consistency, clinical protocol compliance, and estimated billing revenue.
- Implemented backend APIs feeding live dashboard panels: paramedic/EMT performance tables, medication/procedure frequency rankings, incident heat maps, patient demographics, service line distribution, and payor mix analytics.
- Integrated AI Documentation Intelligence panel tracking documentation accuracy (89.5% field completion rate), PCR completion before hospital, GCS/trauma scale capture, and protocol adherence — giving EMS directors real-time quality visibility.

NEMSIS 3.5.1 AI-Powered EMS Submission Pipeline (Java · Spring Boot · GPT · Kubernetes)

- Designed and built an end-to-end NEMSIS 3.5.1 XML submission pipeline in Java/Spring Boot converting AI-extracted PCR JSON from EMS audio transcripts into XSD-compliant SOAP envelopes submitted to state and vendor endpoints in real time.
- Integrated multi-stage AI enrichment pipeline: ASR transcription → GPT-based structured PCR extraction (vitals, medications, procedures, clinical impressions) → NEMSIS code mapping (ICD-10, RxNorm, SNOMED CT) → XML builder → vendor SOAP submission.

- Built intelligent fallback chains for data completeness: age extraction from clinical narrative when DOB unavailable, plain-text impression fallback when ICD-10 unresolvable, RxNorm resolution from drug name strings, and SNOMED procedure mapping from free-text descriptions.
- Reduced vendor rejection rate by implementing empty section suppression — dynamically omitting XML sections (eVitals, eMedications, eHistory, eScene, eDisposition, eNarrative) when no clinical data exists rather than emitting empty/default elements.
- Developed two-pass AI prompt engineering: primary extraction prompt for structured PCR data from EMS audio, and a reconciliation mapper prompt cross-referencing narrative against structured fields to fill coverage gaps.
- Deployed on Kubernetes (3-replica) with live vendor integration, iterating on extraction accuracy through real EMS encounter feedback loops.

QR-Based Clinical Handoff Token System (Java · Spring Boot · JWT)

- Designed and implemented a secure, zero-login patient handoff system: paramedic generates a short-lived JWT token at handoff, QR code is displayed on the ePCR device, ED staff scans on arrival to instantly access the live patient care report.
- Built generateHandoffToken() and validateHandoffToken() in JwtToken.java (8-hour expiry), POST /api/v1/patients/{uuid}/handoff-token endpoint in PatientController.java, and public GET /api/v1/handoff/summary?token=xxx endpoint — bypassing authentication for ED staff while maintaining patient-scoped security via token claims.

Clinical Audit & Confidence-Scoring Pipeline (Python · NLP · Sentence Transformers · Gemini)

- Built a Python audit pipeline comparing EMS audio transcripts against clinical and billing reports (PDF, HTML, TXT) to generate consistency confidence scores — enabling automated quality gating of patient care documentation.
- Designed weighted overall_consistency scoring using semantic similarity, bidirectional coverage (A2R/R2A), hallucination rate, field-level consistency, and contradiction penalties — with configurable OK/WARN/ALERT decision thresholds.
- Implemented strict clinical field validation for key EMS indicators (GCS, IV/fluids, oxygen, bleeding control, ETA, splinting) to reduce false-positive quality scores on incomplete documentation.
- Integrated embedding-based similarity using sentence-transformers (e5-large-v2, MiniLM) with chunk-level matching and low-match diagnostics for explainable, interpretable audit outputs.
- Generated machine-readable audit artifacts (audit_result.json) with per-metric scores, item-level hallucination severity, and decision outputs for downstream QA and clinical review workflows.

Infrastructure & Integration (Kubernetes · AWS · Linux · Ericsson R1900)

- Configured and integrated Ericsson R1900 media server containers for streaming EMS data to AWS, supporting real-time data transport for platform ingestion workflows.
- Managed Linux-based service environments, CI/CD pipelines via GitHub Actions, and Kubernetes deployments (kustomize/kubectl) across ECS microservice architecture.
- Diagnosed and resolved Cassandra quorum loss and Kubernetes pod networking failures on AWS EKS, restoring platform availability during production incidents.
- Built and maintained Python/FastAPI ingestion and validation services handling HL7 v2 and FHIR payloads with schema validation and structured error handling.

Tata Consultancy Services (TCS) | System Administrator — Automation & Infrastructure

Feb 2020 – Aug 2022 · Bangalore, India

- Designed and built internal automation tools using Python, Bash, and PowerShell to eliminate recurring manual tasks and streamline operational workflows across support processes.
- Developed SQL-based diagnostic scripts and tooling to investigate production incidents, identify root causes, and help application teams restore services faster.
- Built monitoring and health-check scripts across Linux, Windows, and macOS environments for proactive detection of reliability issues before they impacted operations.
- Collaborated with application development teams on production issues, debugging application-layer failures using log analysis, network diagnostics, and container-level troubleshooting.

EDUCATION

Jessup University
M.S. in Computer Science · In Progress (Expected 2027)

California Lutheran University
M.S. in Information Technology · 2022–2024 · GPA: 3.74/4.0
Coursework: Distributed Systems, Data Communications & Networking, Information Security, Data Warehouse & BI, Project Management