Roszhan Raj Meenakshi Sundhresan

San Jose, CA +1-657-319-6341 roszhan23@gmail.com github.com/roszhan2684 linkedin.com/in/roszhanraj2684

Education

California State University, Fullerton — Fullerton, CA
M.S. in Computer Science, GPA: ***3.96/4.0*** — *Jan 2024 – Dec 2025*

Anna University — Chennai, India
B.E. in Computer Science and Engineering, GPA: 3.8/4.0 — *Aug 2019 – Jun 2023*

Technical Skills

Languages: TypeScript, JavaScript, Python, Java, C++, Swift, SQL
Frontend / Backend: React, Next.js, React Native, SwiftUI, Node.js, Express.js, FastAPI, Django REST Framework,GraphQL
Data / Systems: PostgreSQL, MongoDB, Redis, Firebase, Kafka, event-driven systems, async workers, idempotency, caching
AI / ML: LLMs, RAG, agentic workflows, Bedrock/Claude, Gemini, MLflow, Core ML, ONNX, PyTorch
Cloud / Engineering: AWS, GitHub Actions, CI/CD, Linux, OpenTelemetry, system design, debugging, automated testing, Git

Experience

INSAIT Solutions LLC — Fremont, CA
Software Engineer — *Feb 2026 – Present*

- **Built and shipped 10+ production workflows end-to-end** using **React, TypeScript, Django REST Framework, PostgreSQL, and AWS**, owning frontend interfaces, backend APIs, data models, testing, deployment, and post-release iteration.
- **Developed reusable product architecture and API integrations** spanning **40+ REST endpoint modules and 30+ PostgreSQL-backed models**, supporting multi-role workflows across intake, encounters, transcription, AI-generated documentation, approvals, and reporting.
- **Improved reliability across 5+ critical workflows** by reproducing production defects, performing root-cause analysis, and hardening **authentication, async state, API error handling, validation, loading states, and backend logic**.
- **Reduced manual clinical documentation by ∼25%** by engineering an AI-assisted workflow integrating **AWS Transcribe and Bedrock/Claude** with structured data, human review, and approval controls before generated content entered downstream workflows.
- **Translated stakeholder and customer feedback into shipped features**, collaborating across engineering, product, and clinical teams while contributing to code reviews, technical documentation, testing, debugging, and release workflows.

Projects

Orbis Relay – AI Agent Security & Trust Platform *Python, TypeScript, React, FastAPI, PostgreSQL, Core ML, C++*

- **Built and deployed an end-to-end AI-native security platform** that evaluates human and autonomous-agent actions using deterministic policy and ML risk signals, with human approval workflows and auditable decision receipts.
- Built a reproducible ML pipeline over **33,274 security events** and a **93-scenario hand-written gold set**, comparing four model families and using calibration, adversarial evaluation, drift detection, retraining, and **19 automated release gates**.
- Productionized local inference with **Core ML** on a native Swift macOS endpoint, achieving **57 μs P95 inference on an M4 Pro** and sustaining **50 events/sec for 10 minutes with zero failures**; profiling improvements made event processing **4.6× faster**.
- Built signed model updates, atomic activation, rollback, kill switch, canary rollback, review-driven retraining, and a **C++20 feature-calculator port** matching Python across all 33,274 dataset events.

OpsPilot – AI-Powered Incident Response & Reliability Platform *TypeScript, Next.js, Python, FastAPI, PostgreSQL, Redis, Kafka*

- **Built a full-stack incident-response platform** that ingests application events, deployment metadata, logs, metrics, and traces to help engineers investigate production failures and prioritize likely root causes.
- Designed an **event-driven backend architecture** using **FastAPI, PostgreSQL, Redis, Kafka, and asynchronous workers** to coordinate incident ingestion, enrichment, correlation, background analysis, and retry-safe processing.
- Built AI-assisted investigation workflows that correlate recent deployments, service dependencies, telemetry anomalies, and historical incidents while keeping remediation actions behind **explicit human approval and auditable execution controls**.
- Added reliability mechanisms including **idempotent event processing, bounded retries, failure isolation, structured logging, metrics, tracing, and rollback-safe workflows**, making incident analysis observable and resilient to partial service failures.

ExpenseIQ – AI-Powered Personal Finance Platform *TypeScript, Next.js, React, Node.js, MongoDB, Plaid API*

- **Built and deployed a full-stack financial platform** using **TypeScript, Next.js, React, Node.js, and MongoDB**, spanning **38 API routes and 7 data models** for bank synchronization, transaction processing, expense tracking, and group payments.
- Eliminated duplicate and lost transactions during bank synchronization by engineering a **cursor-based delta-sync pipeline** with idempotent upsert-safe writes using the **Plaid API**.
- Built voice and natural-language transaction entry, spending Q&A, and group bill splitting with a multi-provider AI fallback layer using **Gemini, Groq, and Ollama**, while requiring confirmation before sensitive database writes.