

SIVA SANKAR REDDY ASAM

+91 8688660794 | aasamsivasankarreddynani143@gmail.com | [Linkedin](#) | [GitHub](#) | [Portfolio](#)

Technical Highlights

- Ranked **Top 2%** on LeetCode (500+ problems) with expertise in **distributed systems** and **low-latency design**
- Built **Aether**, a real-time incident response platform (NestJS/Next.js) deployed on AWS Fargate, handling **10k+ concurrent WebSocket connections**
- **60% performance gains** at Nirmaan via Redis caching and SQL optimization for large datasets
- Proficient in **Go, Python, React/TypeScript**, and **AWS**. Contributor to **LLM-powered workflows** and **WebSocket-heavy architectures**

Work Experience

Software Engineer | Nirmaan (Stratos Infra Technologies Pvt. Ltd.)
Bengaluru, India | July 2024 - Present

- Architected and implemented multi-layer caching (**React Query** client-side, **Redis server-side** via **Frappe framework**), slashing **API latency by 55%** and improving overall application responsiveness by **60%**
- Overhauled foundational codebase modules, **refactoring ~80%** of prior code using **SOLID principles** and **modular design**, reducing technical debt by an **estimated 45%** and enhancing maintainability
- Designed and implemented a **high-performance data table** component featuring **server-side pagination**, **dynamic filtering**, and **optimized SQL queries** via the **Frappe framework**, improving rendering **speed for large datasets by up to 60%**.
- Took ownership of and developed a **real-time internal communication system** using **WebSockets (Socket.IO)** and architected a **context-aware chatbot** leveraging **database integration** to provide instant answers to **domain-specific queries**
- Spearheaded integration of **Python-based AI analytics** for real-time insights into projects, orders, and vendor metrics, enabling **data-driven operational decisions**
- Developed and optimized **RESTful APIs** within the **Frappe framework** and refined frontend state management for complex workflows, contributing to a **35% reduction** in specific component load times

Full Stack Engineer | Attica Gold Company
Bengaluru, India | March 2024 - July 2024

- Developed a **CCTV management system** PoC using **MERN stack**, **OpenCV**, and **Flask**, handling streaming from multiple sources
- Utilized **AWS (S3, Lambda)** for secure footage storage and **JWT/RBAC** for access control
- Implemented **WebSocket-based communication** for aspects of the streaming pipeline

Skills

- **Languages:** JavaScript/TypeScript, Python, Go, SQL
- **Frontend:** React, Next.js, Redux/Zustand, Tanstack Query, Tailwind CSS, HTML/CSS
- **Backend:** Node.js (NestJS, Express.js), Python (Flask, FastAPI, Django), Go, RESTful APIs, WebSockets (Socket.IO), Microservices (Conceptual), Message Queues (BullMQ, Redis Streams), JWT Authentication
- **Databases:** PostgreSQL (Prisma), MongoDB, MySQL, Redis (Caching, Pub/Sub)
- **Cloud & DevOps:** AWS (EC2, S3, Lambda, RDS, ECR, ECS Fargate, CloudWatch), Docker, Terraform (IaC), CI/CD (GitHub Actions), Serverless (Conceptual)
- **Testing:** Jest, Supertest, React Testing Library (RTL), Cypress/Playwright (Conceptual) - *Add specific testing experience*
- **Architecture & Design:** System Design (HLD, LLD), API Design, Database Modeling, SOLID Principles, Design Patterns, OOP/OOD
- **Tools:** Git, GitHub, Docker Desktop, VS Code, Postman, Insomnia REST, Figma (collaboration), Jira, Confluence

Projects

Aether – Collaborative Real-Time Incident Response Platform | [GitHub Link](#)

Tech: TypeScript, NestJS, Next.js, PostgreSQL, Prisma, Redis, WebSockets, Docker, GitHub Actions, AWS Fargate, Terraform

- Engineered a **full-stack, real-time incident management** platform from concept to MVP deployment within an 8-week sprint, simulating enterprise development cycles. ([Live Demo Link](#))
- Built a scalable backend using **NestJS (Node.js/TypeScript)**, implementing modular design, **JWT authentication (Passport.js)**, and **Prisma ORM** for PostgreSQL database interactions
- Developed a real-time collaborative "**war room**" leveraging **WebSockets (Socket.IO)** for live chat, status updates, and participant tracking, ensuring low-latency communication during critical incidents. Solved WebSocket authentication race conditions
- Built a responsive frontend with **Next.js (React/TypeScript)**, utilizing **Tanstack Query** for server state management, **Zustand** for global UI state, and **Tailwind CSS** with shadcn/ui for the component library
- Implemented robust data modeling for **users, teams, incidents, messages, and tasks**, establishing clear **relationships and indexing** for performance
- **Containerized** application services (**Frontend, Backend, DB, Redis**) using **Docker** and established a **CI/CD pipeline** with **GitHub Actions** for automated **testing, building, and deployment**
- Deployed the application to **AWS ECS Fargate** using managed services (**RDS, ElastiCache**), demonstrating understanding of cloud infrastructure and deployment patterns