NITHISH SURESH BABU

Seattle, WA
+1 2062268935 | nithish952001@gmail.com | linkedin.com/in/nithish-suresh-babu
github.com/nithish-95 | portfolio.nithish.net

SUMMARY

Full-Stack Software Engineer with experience delivering GenAI platforms and scalable solutions. Led the design of an AI-powered job matching platform that integrated Gemini 2.5 Flash, AWS S3, and Bedrock, boosting recommendation accuracy and enabling fast releases via automated CI/CD. Delivered a document-generation system for an aviation insurance product, improving reliability and reducing processing errors through a Temporal-based workflow. Seeking to apply deep expertise in secure backend architecture, React/Next.js front-ends, and cloud-native AWS services to accelerate AI-driven product outcomes.

EDUCATION

University Of Michigan **Aug 2023 - Apr 2025**
Master's, Computer and Information Science *Michigan, USA*

SKILLS

- **LANGUAGES**: Go, Python, TypeScript, JavaScript, C++, HTML, CSS, SQL
- **FRONTEND**: Tailwind-CSS, Shadcn/ui, Next.js, React, React Native, TanStack, Flutter
- **BACKEND**: Go, FastAPI, Node.js, Supabase, Firebase, Convex, Pocketbase
- **DATABASE**: PostgreSQL, SQLite3, AWS RDS, DynamoDB, Redis
- **CLOUD**: AWS, GCP, EC2, CloudFront, Amplify, Fargate, Apprunner, AWS rekognition, S3, Sagemaker
- **DEVOPS**: CI/CD, Docker, Git, GitHub Actions, Route53, Kubernetes
- **AI**: LangChain, Bedrock, Ollama, RAG, AI Agents

OPEN SOURCE CONTRIBUTIONS

Go library for accessing the GitHub v3 API | go-github
Contributor

- **Engineered and implemented new API methods** to manage **GitHub Enterprise App** repository installations, successfully enhancing the functionality of the widely-used library.
- Authored **production-quality** Go code and comprehensive unit tests to guarantee the reliability and stability of new library features.
- Collaborated with other maintainers through the Pull Request (PR) review process, refactoring code to align with established project conventions and Go best practices.

Time Tracking & Project Management Platform | Chrono
Project Maintainer

- **AI-Powered Project Management Suite:** Architected a unified **Time Tracking and Kanban** platform for small teams, featuring a custom **AI Analytics layer** that ingests workflow data to generate productivity insights using multiple LLM models.
- **Modern Full-Stack Implementation:** Built a high-performance application using **Next.js 16** and **React 19**, leveraging Server Actions and Server-Side Rendering (SSR) for optimal SEO and speed.
- **Data & Schema Design:** Designed a relational schema on /**PostgreSQL** to manage complex relationships between users, timesheets, and project tasks, ensuring data integrity and efficient querying.

EXPERIENCE

SQUARE RESULTS **Aug 2025 - Present**
Software Development Engineer - GenAI *Remote, WA*

- **Architected RAG based job matching pipeline** using **Gemini 2.5 Flash** that ingests user resume from **AWS S3** and queries relevant job descriptions from **AWS Bedrock** knowledge base (a vector database), generating personalised job recommendations for the user. This pipeline runs periodically and the results are stored in **Supabase** for fast retrieval to be shown on the user home page. This approach improves match accuracy through semantic understanding of skills and job requirements over traditional text based search.
- Automated deployment by building a **GitHub Actions workflow** that containerises backend services with Docker, pushes images to **AWS ECR**, deploys them to **AWS App Runner**, and deploys the frontend to **AWS Amplify**, resulting in faster releases and fewer errors.
- Built a secure, multi-tenant backend with FastAPI (Python) and responsive frontend interfaces with Next.js, ensuring strict data isolation, reducing data-leak risks, and enabling safe access for all users.
- **Collaborated with stakeholders** across product and data teams to demo real-time recommendation capabilities, translating technical AI concepts into actionable business insights for leadership.

CLASSA.AI **Jul 2025 - Aug 2025**

Software Developer - Contract *Remote, WA*

- **Re-architected a document generation workflow** for an aviation insurance platform using **Temporal** to manage a complex **30-branch decision tree** for private jet insurance policies.
- **Replaced a fragmented, database-backed state machine** with a **durable workflow model**, eliminating the need for manual state persistence and complex database triggers across a multi-step user wizard.
- **Enhanced data consistency and fault tolerance** by leveraging **Temporal's** stateful history, providing a deterministic way to resume interrupted document generations and automate policy delivery to high-net-worth clients.

CENTRAL MARINE RESEARCH INSTITUTE **Jan 2025 - Jul 2025**

Full Stack Engineer *Remote, WA*

- Transformed unstructured PDFs of Marine institute's internal encyclopaedia into a structured SQLite3 database by extracting images and metadata with Python, enabling **efficient local-first** data access for **100+ researchers** in low-connectivity environments
- Built responsive web application using **Vue.js** and **Go** backend, delivering real-time visualisation and search of marine research data, which reduced data retrieval time for researchers
- Developed cross-platform mobile app using Flutter with offline-first architecture, allowing researchers to capture and sync complex datasets without data loss in remote marine research settings.

UNIVERSITY OF MICHIGAN **Sep 2024 - Dec 2024**

Research Assistant -GenAI DeepFake *Dearborn, MI*

- **ML Model Evaluation:** Developed benchmarking pipelines using **Python** and **TensorFlow** to evaluate eight Deepfake detection models. This benchmarking pipeline was used to evaluate our custom model against existing models achieving 88.37% accuracy on diverse GenAI image datasets.
- **Engineered** a continuous feedback loop by deploying a full-stack Next.js prototype for public testing, utilising a custom SQLite schema to capture user data that **directly informed iterative model retraining and improved detection performance.**

ACADEMIC PROJECTS

ADVANCED OPERATING SYSTEM **Oct 2024 - Feb 2025**

SCALABLE CHAT APPLICATION

- **Engineered a real-time, low-latency** chat application using **Go and WebSockets** to support bi-directional communication across **multiple concurrent clients.** I designed a centralised "hub" to manage the lifecycle of active connections, ensuring that incoming messages are broadcasted across the participant pool with minimal delay.
- Implemented a **thread-safe** connection manager using Go's concurrency primitives **goroutines and channels** to handle simultaneous messaging streams without **race conditions.** By decoupling message ingestion from the broadcasting logic, I ensured that the system remains responsive even under **high-concurrency loads**, preventing blocking operations in the main execution path.
- Focused on operational stability by incorporating heartbeat monitoring and graceful cleanup of stale **WebSocket** sessions to mitigate **memory leaks and resource exhaustion.** I utilised a clean, modular architecture that separates the transport layer from the business logic, making the system easy to extend with features like private rooms or persistent message storage.

CLOUD COMPUTING **Nov 2024 - Jul 2025**

MULTIPLAYER TIC-TAC-TOE

- Architected and deployed a real-time Tic-Tac-Toe application using Go and WebSockets on **AWS Fargate,** leveraging a **serverless container model** to eliminate infrastructure overhead. I containerised the service using a **multi-stage Docker build** to minimise the attack surface and image size, ensuring a lean production environment by separating the Go build pipeline from the final Alpine runtime.
- Designed a secure networking stack featuring an **Application Load Balancer (ALB) and Route 53** for **custom domain mapping and traffic orchestration.** I implemented granular IAM role separating execution permissions from task logic-and configured isolated security groups to ensure least-privilege access, allowing only necessary traffic from the ALB to the containerised application.
- Established a **scalable deployment workflow** using **Amazon ECR and ECS,** enabling **automated lifecycle management** and **cost-efficient resource scaling.** By moving to a Fargate-based architecture, I transitioned the application from a manual setup to a production-ready environment that supports durable WebSocket connections and automatic health monitoring.

ACHIEVEMENTS

- **People's Choice of the Year 2025 (Google Cloud Developers):**Designed and Developed AI Voice agent that Automates the Insurance Quoting via Phone calls