

# SWETHA REDDY SINGIREDDY DAMYELLA

San Jose, CA ◊ +1(669) 335-7888 ◊ [swethareddy.singireddydamyella@sjtu.edu](mailto:swethareddy.singireddydamyella@sjtu.edu) ◊ [linkedin.com/in/swetha-reddy-singireddy-damyella-495353298/](https://linkedin.com/in/swetha-reddy-singireddy-damyella-495353298/)

## EDUCATION

Master of Science in Software Engineering, San Jose State University Relevant Coursework: Cloud Computing, Distributed Systems, Software Systems Engineering, Data Mining	Jan 2024 – Dec 2025   San Jose, CA
Bachelor of Technology in Computer Science, Malla Reddy College of Engineering and Technology Relevant Coursework: Machine Learning, Artificial Intelligence, Data Structures & Algorithms, Database Systems.	Jun 2018 – May 2022   Hyderabad, India

## TECHNICAL IMPACT SUMMARY

- Designed and implemented an on-demand evidence capture system for Amazon's internal compliance automation, replacing a rigid weekly batch pipeline with a self-service, API-driven workflow for auditors and security teams.
- Modernized customer-facing applications and backend microservices for AstraZeneca by migrating to a cloud-native architecture on Azure, improving API performance by ~30%, reducing query latency by ~50%, and maintaining ~97% application uptime through caching, event-driven workflows, and enhanced observability.
- Built and deployed end-to-end AI/ML systems through production-grade projects, spanning deep learning, Retrieval-Augmented Generation with LLMs, and classical machine learning, with containerized pipelines supporting real-time inference and scalable experimentation.

## TECHNICAL SKILLS

Languages: Python, Java, Go, C++, JavaScript, TypeScript, HTML, CSS

Frameworks: Spring Boot, Django, Flask, Node.js, ReactJS, Vue.js, Selenium

Cloud & DevOps: AWS, Azure, GCP, Docker, Kubernetes, Terraform, Jenkins, Jira

Data, Databases & Observability: Kafka, Apache Spark, Hadoop, PostgreSQL, MySQL, MongoDB, DynamoDB, CloudWatch, Prometheus, Grafana, ELK Stack

## EXPERIENCE

### Software Development Engineer Intern, Amazon Web Services (AWS)

Compliance and Security Team

May 2025 – Aug 2025

Seattle, USA

- Architected an **on-demand, serverless, event-driven evidence capture system** using **API Gateway, AWS Lambda, AWS Step Functions, DynamoDB, Amazon S3, and SQS Dead Letter Queues**, extending an **existing scheduled batch workflow** to enable faster, **on-demand audit and compliance requests**.
- Designed and implemented **RESTful APIs** secured with **IAM role-based authorization** and **SigV4 authentication**, applying **schema-level input validation** and authorization checks to safely trigger evidence generation and track request lifecycles using **UUID-based correlation identifiers** across distributed services.
- Engineered **asynchronous, state-machine-driven orchestration** using **AWS Step Functions** to coordinate screenshot capture, redaction, PDF transformation, and publishing, staging artifacts in **Amazon S3** and configuring **SQS DLQs** and retries to ensure **non-blocking execution**, fault isolation, and near real-time evidence generation.
- Designed **scalable DynamoDB schemas** with optimized **partition keys, Global Secondary Indexes (GSIs), TTL-based lifecycle management**, and **AWS KMS encryption** to persist request metadata, execution states, correlation identifiers, and compliance artifact references with strong durability guarantees.
- Implemented **reliability and observability layers** using **CloudWatch metrics, dashboards, and alarms** with **SQS DLQs**, along with **testing (JUnit, Mockito)** and **ReactJS components (TypeScript, Hooks)** for real-time evidence status tracking and operational visibility.
- Eliminated **100% manual evidence generation**, reducing audit request turnaround time by **~95% (hours → <1 minute)** and enabling access to **compliance evidence**.

### Software Engineer, Cognizant Technology Solutions

Client: AstraZeneca (Global Biopharmaceutical Company)

Mar 2022 – Dec 2023

Hyderabad, India

- Designed and developed customer-facing web applications by building **Java Spring Boot microservices** and **RESTful APIs** with **DTO-based request/response models** and **JDBC-based database access**, supporting high-traffic public endpoints in a distributed cloud environment.
- Contributed to implementing **persistent data storage** using **Azure SQL Database** as the system of record for transactional and business-critical data, applying **normalized schemas, clustered and non-clustered indexing strategies, query optimization**, and **ACID guarantees** to ensure consistency and data integrity.
- Improved **API throughput** by **~30%** by optimizing REST endpoint contracts, introducing **Redis caching (Azure Cache for Redis)** with TTL-based eviction policies, and offloading long-running synchronous operations to **Kafka-based asynchronous event workflows** in a distributed microservices architecture.
- Supported the integration of **Apache Kafka**, designing topic structures and **partitioning strategies** to enable event-driven processing, service decoupling, scalable inter-service communication, and reliable message delivery with retries and back-pressure handling under high request volumes.
- Containerized and deployed **cloud-native microservices on Microsoft Azure AKS** using **Docker** and core **Kubernetes primitives** (Deployments, Services, Ingress, ConfigMaps, Secrets), and supported **CI/CD pipelines** using **Azure DevOps and Jenkins** to automate build, test, and AKS deployments across environments.
- Supported **Terraform-based infrastructure-as-code** to provision and manage Azure resources, integrating **Azure Key Vault** for secrets management, configuring **Azure Monitor metrics and alerts**, reducing manual subscription administration and configuration drift by **~75%**, and delivering features through **Agile sprints tracked in Jira**.

### Software Engineer Intern, Cognizant Technology Solutions

Cloud Monitoring, Security, and Analytics Team

Oct 2021 – Mar 2022

Hyderabad, India

- Contributed to a cloud-based monitoring and analytics platform by analyzing **DNS traffic patterns, load balancer behavior**, and **system telemetry** to support security and operations teams in identifying anomalies and validating mitigation strategies.
- Assisted in developing **GraphQL APIs** to aggregate and query monitoring data from multiple backend sources, improving query flexibility and reducing over-fetching compared to traditional REST-based integrations.
- Worked on log ingestion and health monitoring pipelines, processing large-scale DNS and access logs using **Hadoop** and building **Splunk dashboards and alerts** to track latency, error rates, and traffic anomalies across application load balancers.

## PROJECTS

### ImageCaptionAI – Deep Learning Image Captioning System

Python, PyTorch, CNN (ResNet50), LSTM, GloVe, NLP, Flask, Docker

- Built an end-to-end image captioning pipeline using a CNN-based encoder (ResNet50) for high-level visual feature extraction and an LSTM-based decoder trained with teacher forcing to generate fluent, context-aware natural language descriptions.
- Integrated pretrained **GloVe embeddings** to improve semantic representation, implemented beam-style decoding for inference stability, containerized training and inference using **Docker**, and exposed real-time caption generation through a **Flask API**.

### Domain-Aware Customer Support AI using LLM + RAG

Python, PyTorch, Transformers, SBERT, FAISS, LangChain, LLaMA, NLP

- Designed a domain-aware **Retrieval-Augmented Generation (RAG)** pipeline using **LangChain** to orchestrate document ingestion, chunking, metadata enrichment, SBERT-based embedding generation, semantic retrieval, and prompt assembly for **LLaMA**-based inference.
- Implemented a **Docker**-containerized inference service that embeds user queries, retrieves top-K relevant context from **FAISS**, injects grounded evidence into prompts to reduce hallucinations, and serves low-latency responses via a **Flask API**.

### Intelligent Customer Churn Prediction System

Python, Scikit-learn, Logistic Regression, SVM, Random Forest, KNN, Matplotlib

- Developed a classical machine learning pipeline to predict customer churn from structured data, performing feature engineering, handling class imbalance, and comparing Logistic Regression, SVM, Random Forest, and KNN models.
- Containerized model training and inference using **Docker**, evaluated models using precision, recall, F1-score, and ROC-AUC metrics, and generated probability-based churn risk scores with performance visualizations using **Matplotlib**.