Sai Varshini Thupakula

+1 (617) 671-4560 | saivarshini.thupakula@outlook.com | github.com/SaiVarshini1410 | linkedin.com/in/saivarshinithupakula

EDUCATION

Master's in Computer Science, Northeastern University | Boston, MA, United States Sep 2025 – May 2027

- Courses: Programming Design Paradigm | Database Management Systems

Bachelor's in Computer Science, Vasavi College of Engineering | India | GPA: 3.7/4.0 June 2019 – May 2023

- Courses: Data Structures and Algorithms | Design and Analysis of Algorithms | Object-Oriented Programming - Java | Software Engineering | Networking | Database Management Systems | Computer Organization and Architecture

EXPERIENCE

IBM, *Software Engineer* | Hybrid (Pune, India) Aug 2023 – June 2025

- Improved **system stability** by 25% by refactoring from **OSGi (Virgo)** modules to a consolidated **Spring Boot** monolith; resolved DI boundaries and rewrote shared services.
- Delivered 20% faster queries and simpler **failover** by migrating **PostgreSQL** persistence to a unified **MongoDB** collection (5M docs) and consolidating clusters into a single **replica set** with **rolling upgrades**.
- Cut maintenance downtime by 40% across 500 servers by leading an **RHEL 8 to 9.5** upgrade: scripted workflows, built a **custom ISO**, ported services to **Python 3.9**, and applied **kernel** patches.
- Raised **runtime reliability** by 15% and reduced memory footprint by 30% through leak mitigation, session logic optimization, and standardized **structured logging** & rotation.
- Resolved production-down incidents during customer **critsits**: root-caused **Virgo** crash loops, implemented permanent fixes, and restored service within hours while coordinating hotfix rollout and post-mortems.
- Automated ingestion and ops workflows (**Bash**, **Python**, JSON) and shipped **BIRT**-backed operational reports with optimized queries - automating 40% of manual processing and improving analytics transparency by 20%.

IBM, *Software Engineer Intern* | On-site (Pune, India) Jan 2023 – July 2023

- Cut resource allocation time by 25% by building a full-stack **Resource Management** platform with **RESTful APIs** (**Node.js**, **MySQL**) and containerized deployment via **Docker**; owned DB schema and backend endpoint design.
- Reduced environment setup time by 30% by deploying on **VMs** with optimized **Docker** compose/build caching.
- Improved operational efficiency by 30% by integrating **Python** collectors streaming **hypervisor/VM metrics** (CPU, memory, disk) into the backend for automated monitoring and reporting.
- Lowered manual QA cycles by 40% by automating high-priority test flows and extending the internal **CBA** automation framework, increasing backend validation coverage and reliability.

SKILLS

Backend:	**Java** (core, concurrency), Spring Boot, **Python**, **REST APIs**, JSON
Data & Persistence:	**MongoDB** (replica sets, rolling upgrades), PostgreSQL, **MySQL**, Schema/Version Migrations
Systems & Scripting:	**Linux (RHEL 8/9)**, **Bash**, Python 3.9, Package/Runtime Hardening, **Virtualization**
Build & Delivery:	**Git**, Maven/Gradle, Jenkins (CI/CD), Docker, ISO/OVA deployment, Release Packaging, **VMware**

RESEARCH PAPER

Automated Glaucoma Detection – Published Paper

- Engineered a diagnostic system for early glaucoma detection by integrating CNN, SVM, and KNN models, achieving 94% accuracy on the RIGA dataset (2,664 retinal fundus images), using advanced image preprocessing, supervised learning, and model evaluation techniques to enable scalable, AI-assisted screening and support vision-loss prevention.

PROJECTS

Track-It (Source)

- Developed a full-stack job application tracking system with a dynamic status workflow (Pending, Interview, Declined) and integrated data analysis visualizations (graphs/histograms) on the dashboard, built using ReactJS, HTML/CSS (frontend), Node.js, ExpressJS, and MongoDB (backend) for seamless tracking, organization, and insight generation.

Java CLI Food Ordering System (Source)

- Designed and implemented a modular, console-based food ordering system in pure Java, featuring user authentication, dynamic menu management, discount-based billing, using OOP principles, exception handling, modular classes, and interface-driven design, with a scalable architecture for future integration of GUI or database enhancements.

Covid Safe Room (Source)

- Implemented a state-machine–driven access-control backend on microcontroller hardware: separated business rules from device I/O, debounced concurrent PIR signals, validated temperature thresholds, and maintained an idempotent occupancy counter with clear fault paths—emphasizing state consistency, input validation, and reliability.