DHRUVIN GANDHI

+1(413) 800-6652 | Amherst, MA | dhruvinrakes@umass.edu | linkedin.com/dhruvin-gandhi | Portfolio Website

EDUCATION

University of Massachusetts, Amherst   **Sep 2023 - May 2025**
Master of Science in Computer Science   **GPA: 3.95/4.0**
Coursework: Distributed Systems, Algorithms for Data Science, Advanced NLP, Machine Learning, Database Systems

Veermata Jijabai Technological Institute   **Aug 2019 - May 2023**
Bachelor of Technology in Computer Engineering   **GPA: 9.13/10**
Coursework: Operating Systems, Data structures, Algorithms, Object Oriented Programming, Big Data Analytics

SKILLS

Programming Language	C++ , Java, Python, Go, R, JavaScript, TypeScript, SQL, NoSQL
Technologies/Frameworks	Linux, Flask, Django, Node.js, Angular, React, Spring Boot, Kubernetes, AWS, Docker

EXPERIENCE

Cisco   Jan 2025 - Present
ML Student Researcher   *Amherst, MA, USA*

- Developing a novel RAG pipeline that enhances LLMs with user-specific knowledge graphs to personalize text generation.
- Incorporating agentic reasoning in RAG pipeline to update knowledge graphs and retrieve data to the current context.

Advanced Networked Systems Research Lab   Sep 2024 - Jan 2025
Graduate Student Researcher   *Amherst, MA, USA*

- Implemented consensus algorithms (Lazy and Chain Replication) to enhance fault tolerance in the XDN system.
- Proposed and built an optimized interoperability pipeline for Java and C# integration using CoreCLR and JVM.
- Reduced the cross language call latency in this pipeline by **40%** enabling XDN to support any consensus algorithm

Interactive Brokers   Jun 2024 - Aug 2024
Software Engineering Intern   *Greenwich, CT, USA*

- Developed the logic for the formal models of distributed trading protocols using P language, C# and state machines.
- Tested for consistency, fault tolerance, and latency specifications on these models for **1000 users**.
- Identified a critical vulnerability in real-time data access for multi-regional clients due to asynchronous network delays.
- Proposed a working solution with **Logical Lamport clocks**, ensuring 100% correctness across distributed environments.
- Automated Java method invocation in a P runtime by implementing a **C++ bridge**, boosting efficiency by **90%**.

Morgan Stanley   May 2022 - Jul 2022
Software Engineering Intern   *Mumbai, India*

- Created a full-stack internal web tool with **10+ features** using Spring Boot, Angular, TypeScript from scratch.
- Addressed the challenge of unauthorized database updates through this tool which is now being utilized by **7 teams**.
- Built a **reusable** Angular UI for efficiently filtering amongst **10,000+** rows to perform visual CRUD operations.
- Optimized Python script for syncing DEV Table with ~**28M rows**, reducing execution time from **8hrs to 15 mins**.
- Collaborated with the finance team to understand requirements and include their feedback in development process.

Jobdae (Startup)   May 2021 - Jul 2021
Software Engineering Intern   *Remote, India*

- Designed and implemented back-end web architecture using Python, Django, and PostgreSQL, Created **REST APIs** defining functionality for profile management and job application tracking.

PROJECTS

Scalable CI/CD | NextJS, Redis, AWS, ECS, ECR, S3, Docker, Git, Node

- Developed a scalable AWS ECS/ECR architecture supporting up to **5,000** concurrent Docker containers to automate GitHub repo cloning and executable builds. Artifacts stored in AWS S3 and served via a reverse proxy. (Code)

DocuCare - Full-Stack Patient Tracker System | Spring Boot, Angular, JavaScript, Postgres, Amazon S3, JUnit

- Led a test-driven development of a **MVC web application** using agile software design principles such as, version control with Git, organizing development into sprints with fixed durations of 2 weeks and conflict resolution (Code)

Fault-tolerant Distributed System | Java, Zookeeper, Cassandra, Concurrency

- Engineered a **multi-threaded** Fault-tolerant replicated datastore algorithm using Zookeeper as the centralized service, ensuring consistency across replicated servers experiencing **1000 requests/sec** from numerous clients (Code)