

# MARASANIGE SAMARTH MAHENDRA

+1 (857) 707-1671 | samarth.mahendragowda@gmail.com | Boston, MA, USA | [LinkedIn](#) | [GitHub](#) | [Portfolio](#)

## EDUCATION

### Northeastern University

*Master's, Computer Science*

January 2024 - December 2025

- Courses: Programming Design, Software Engineering, Databases, Computer Systems, Algorithms, NLP/ML, Mobile Development

### Dayananda Sagar College of Engineering

*Bachelor's, Computer Science*

August 2018 - July 2022

## SKILLS

**Languages:** Python, Java, C/C++, SQL, NoSQL, JavaScript, TypeScript, HTML/CSS, Rust

**Frameworks & Libraries:** Django REST Framework, Flask, React.js, Node.js

**Databases:** PostgreSQL, Elasticsearch, MongoDB, Redis

**Cloud & DevOps:** AWS, Terraform, Docker, Kubernetes, Prometheus, Datadog, Celery, Linux/Unix

**System Design & Concepts:** Microservices, REST APIs, System Design, scalable backend, distributed systems, Operating Systems, Data Structures & Algorithms: Solved 400+ LeetCode problems (top 1.6% globally)

## PROFESSIONAL EXPERIENCE

### Draup (B2B AI/SaaS, AI-Driven Sales, Talent Intelligence Platform)

Bengaluru, KA, India

*Associate Software Development Engineer*

August 2022 - November 2023

- Owned and evolved **core backend** and data-platform systems in a **5-engineer team**, building mission-critical, **customer-facing** production workloads using **Python, Django REST Framework, PostgreSQL, and Celery-based async pipelines**, supporting **high-traffic, latency-sensitive APIs and data workflows**.
- Led **PostgreSQL → Elasticsearch** migration for real-time analytics and applied advanced **query optimization** techniques (partitioning, indexing, materialized views), delivering **5x faster queries, 400% execution speedups, and 50% lower operational costs** at scale.
- Designed a **dynamic query framework** and advanced filtering engine (boolean operators, nested conditions) using **PostgreSQL + Elasticsearch**, with **Redis caching**, improving chatbot performance by **60%** and reducing new entity development time by **80%**.

*Associate Software Development Engineer Intern*

April 2022 - July 2022

- Built **Datadog** dashboards, integrated **AWS CloudWatch** alarms to **monitor platform health**, reducing issue resolution time by **30%**.
- Implemented caching to improve efficiency of image requests, resulting in a **70% reduction** in load times.
- Developed **self-running Jenkins jobs** for **database cleanup**, cutting manual effort and improving efficiency by **25%**.

## PROJECTS & OUTSIDE EXPERIENCE

### ButterDB — High-Performance B-Tree Key-Value Store (C, Python)

Remote

*C Programming · B-Tree Engine · WAL · Page-Based I/O · Concurrency*

October 2025 - November 2025

- Built a concurrent **single-node TCP** key-value database with custom on-disk B-Tree engine and write buffering (Be-Tree).
- Implemented fixed-page file IO, per-node locking, **Write-Ahead Logging (WAL)**, and crash recovery for durability and concurrency.
- Designed benchmarking and metrics collection to evaluate **throughput, latency, and flush efficiency** under mixed read/write workloads.
- Achieved **~8x higher** insert throughput via **batched buffer propagation** and **IO batching**; code structured for systems-level performance experiments.

### Open Jobs Analytics Platform – Backend Infra + Monitoring

Boston, MA, USA

*Puppeteer, Redis, Celery, MongoDB, Grafana, Prometheus, GPT-4o, AWS*

December 2024 - December 2024

- Designed a **producer-consumer** architecture using **Celery**, integrated with **Prometheus** and **Grafana**, achieving 99% uptime.
- Scraped dynamic web pages with **Playwright** and **Puppeteer**, harvesting **1000+ data points** daily.
- LLM-powered CSS selector extraction reduced new-site onboarding by **90%**.
- Enhanced stealth capabilities with **random headers, user agents, referrer headers, and OS configurations**, reducing bot detection by up to **90%**.

### StackOverflow Full-Stack Q&A Platform

*TypeScript, JavaScript, React.js, Node.js, MongoDB, Cypress, Jest*

February 2025 - April 2025

- Developed Q&A web application using **React (frontend)** and **Node.js** with **TypeScript (backend)**.
- Architected **backend controllers and models** following the **MVC** pattern, leveraging design patterns such as **Facade, Validator, Strategy, and Factory** to ensure **modularity, maintainability, and scalability**.
- Built comprehensive **end-to-end and integration** test suites using **Cypress** and **Jest**, automating user flows (e.g., posting questions/answers, commenting, voting) and achieving high code coverage and reliability.
- Integrated CI with **GitHub Actions, Cypress, and CodeQL** for PR-based linting and coverage checks.

## Stock Market Simulation Application - Java MVC

Boston, MA, USA

February 2024 - April 2024

- Developed a **Java MVC** system managing investment portfolios, integrating a stock **API**, **test coverage**, and buy/sell on specified dates.
- Integrated **data visualization** to plot portfolio performance over time, utilizing bar and line charts for investment growth trends, stock gains/losses analysis, and moving average calculation.

## Operating Systems & Systems Programming Projects (C, x86-64)

September 2025 - December 2025

- User-Space Unix File System (FUSE): Implemented a **Unix-like file system** in user space using **FUSE and C**, supporting core POSIX operations. Designed on-disk structures (**superblock, block bitmap, inodes, directories**) with 4KB block management, path resolution, and persistent storage over a disk image.
- **Virtual Memory & fork()** : Extended a teaching OS WeensyOS kernel to support full virtual memory and process isolation on x86-64. Implemented **per-process page tables**, MMU permission enforcement, virtual page allocation, overlapping address spaces, and a fork() system call that duplicates page tables and copies writable user pages.
- **Unix Shell** Implementation: Implemented a Unix-style shell in C supporting command execution, **pipes**, I/O redirection, background jobs, and conditional chaining . Built a parser and **execution engine** for pipelines and subshells, with process control via fork(), exec\*(), waitpid(), and file descriptor manipulation.
- 

## Real-Time AI Voice Assistant & Intelligent Agent Platform

Boston, MA, USA

April 2025 - May 2025

Tech stack: *OpenAI GPT-4O, Twilio, Celery, FastAPI, Discord, Websockets, Render (deployment), Redis, AWS*

- Architected an agent system integrating **OpenAI GPT-4 + Google Gemini** with modular tools, dynamic function calling, and profile-aware responses via **MongoDB** and **Discord**.
- Built a scalable async backend with **FastAPI + WebSockets**, deployed on **Render** with a **Celery** worker handling long-running tool calls and real-time audio coordination.
- Deployed live demo via public phone number (833) 970-3274 using **Twilio**, showcasing job-query answering, system prompts with resume context, and cross-platform communication.