MARASANIGE SAMARTH MAHENDRA

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

EDUCATION

Northeastern University **January 2024 - December 2025**
Master's, Computer Science
- Relevant coursework: Programming Design, Software Engineering, Database Systems, Computer Systems, Algorithms, NLP/ML, Mobile Development

Dayananda Sagar College of Engineering **August 2018 - July 2022**
Bachelor's, Computer Science

SKILLS

- **Languages:** Python, Java, C/C++, SQL, NoSQL, JavaScript, TypeScript, HTML/CSS
- **Frameworks & Libraries:** Django REST Framework, Flask, React.js, Node.js
- **Databases:** PostgreSQL, Elasticsearch, MongoDB, Redis
- **Cloud & DevOps:** AWS, Terraform, Docker, Kubernetes, Prometheus, Datadog, Celery
- **Systems:** TCP/IP, Locking, Profiling, CPU microarchitecture, lock-free concurrency, cache optimization (L1/L2/L3), async I/O, Multi-threaded Design
- **System Design & Concepts:** Microservices, REST APIs, System Design, scalable backend, distributed systems, Data structures and Algorithms, Solved 400+ problems on LeetCode (Top 1.6% of all LeetCode accounts (by global rank)

PROFESSIONAL EXPERIENCE

Jobstats.fyi **Remote**
Founder & Developer *October 2025 - Present*
- Built JobStats.fyi, a real-time analytics platform visualizing 15K+ interview submissions from 3.7K+ candidates across 1K+ companies (1K+ sessions in 48 hours, all organic).
- Added full observability with Datadog RUM and custom dashboards to track ingestion lag, API latency, and user funnels.
- Designed ingestion + aggregation pipelines (Discord harvesting, ETL into MongoDB, Redis caching) with a Flask API layer and chart-based analytics.

Draup **Bengaluru, KA, India**
Associate Software Development Engineer *August 2022 - November 2023*
- Maintained and enhanced core platform modules across digital stack, outsourcing, customer, and university features, ensuring reliable operation and scalability.
- Designed a dynamic query framework enabling real-time data aggregation, improving chatbot performance by 60% and cutting new entity development time by 80%.
- Revamped filtering logic to support boolean modifers operators and nested conditions, unlocking complex query composition for users.
- Developed 100+ modular REST APIs using Python, Django REST Framework, and PostgreSQL, with Redis caching and Celery workers for async task execution.
- Implemented subscription-based access control to manage app-level permissions and feature visibility.
- Led migration from PostgreSQL to Elasticsearch, enabling real-time analytics with 5× faster queries and higher scalability.
- Applied query optimization (partitioning, indexing, view creation) to achieve 400% faster execution and 50% lower operational cost.
- Built Datadog + AWS CloudWatch dashboards to monitor system health, reducing downtime from 4% to 1% and accelerating issue resolution by 75%.

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 Buffered B-Tree Key-Value Store (C, Python) **Remote**
Python, (C, Concurrency, WAL) *October 2025 - Present*
- 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 ~8× 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**
Tech stack: Puppeteer, Python, Redis, Celery, MongoDB, Grafana, Prometheus, GPT-4o *December 2024 - Present*
- Designed a producer-consumer architecture using Celery, integrated with Prometheus and Grafana, achieving 99.9% 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%.
- *Link to project*

Real-Time AI Voice Assistant & Intelligent Agent Platform | OpenAI GPT-4o, Twilio, Discord, FastAPI **Boston, MA, USA**
Tech stack: OpenAI GPT-4O, Twilio, Celery, FastAPI, Discord, Websockets, Render (deployment), Redis *April 2025 - May 2025*
- 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.
- *Link to project*

StackOverflow *February 2025 - April 2025*

Full-Stack Q&A Platform (StackOverflow Clone) | TypeScript, JavaScript, React.js, Node.js, MongoDB, Cypress, Jest, CodeQL, DevOps

- Developed a full-stack Q&A system using React + Node.js/TypeScript with MVC backend, comprehensive Cypress/Jest testing, and CI/CD via GitHub Actions + CodeQL.
- *Link to project*

Myocardium Wall motion and wall thickness map, Patent Pending :- App-no : 202341086278 (India) (Co-inventor/author) **Bengaluru, KA, India**

Research Assistant | Python, Dicom, MRI , numpy, Tkinter, image processing *November 2021 - September 2023*

- Engineered a Python+DICOM cine-MRI processing system with custom wall-thickness algorithms, improving precision by 50% and accelerating runtime by 60× via NumPy + multiprocessing.