

Bhavani Shankar Mukka

✉ shankarbhavani862@gmail.com ☎ +91 6302336156 📍 Visakhapatnam 🔗 LinkedIn 🐙 Github
🔗 LeetCode

EDUCATION

B.Tech, Computer Science & Engineering

2024 – 2028

Anil Neerukonda Institute Of Technology & Sciences

Current CGPA : 9.0

SKILLS

- **Core CS** : DSA, OOP, Networking, System Designing
- **Languages**: Python, Java, C++, Golang
- **Backend & Distributed Systems**: FastAPI, REST APIs, gRPC, WebSockets, Microservices, net/ http (Golang), RabbitMQ
- **DevOps / Tools**: Docker Compose, Git, GitHub Actions (CI/CD), Linux/Ubuntu, Prometheus, Grafana
- **Databases**: MySQL, PostgreSQL, MongoDB, Redis, Firebase
- **Cloud**: Microsoft Azure, AWS (fundamentals)
- **Testing** : pytest, k6 tests
- **Web/Mobile Frontend (Basics)**: JavaScript, React.js, Android (Java/XML)

PROJECTS

All projects are actively maintained on GitHub, Visit my Github for other Amazing Ai Driven + Networking Projects.

Social Media Backend (50+ Github Stars) [🔗](#)

- Production-grade backend with **60+** REST endpoints and real-time chat/notifications, **using FastAPI, WebSockets, Redis, PostgreSQL, RabbitMQ, Docker Compose, and Azure deployment**, designed to handle **1000+ concurrent users**.
- Implemented **CI and CD pipeline** with GitHub Actions, running **80+ pytest checks** so every production release **follows a tested path to Azure VM**.
- Decoupled Background Jobs with **RabbitMQ and Celery**, Decreasing API time on background jobs to **<100ms** from **2-5s**.
- Added production-grade **observability and benchmarking (Prometheus, Grafana, k6)** for the Azure-hosted backend, **ran smoke/load tests**, and found real infra bottlenecks.

Minicorn (30+ Github Stars) [🔗](#)

- **A lightweight zero-dependency Python HTTP server with 2 interface standards** (WSGI and ASGI), enabling developers to run synchronous and asynchronous python apps through one single interface for faster development.
- Added **WebSocket support for bidirectional communication**.
- **minicorn is standalone which means it comes with 0 extra dependencies**.

Microservices [🔗](#)

- Designed and implemented a **polyglot microservices architecture** with **Go and Python** services communicating via **gRPC**.
- **Containerized 5+ Services** using **Docker Compose** for reproducible local deployment and **service orchestration**.

ACHEIVMENTS

- Solved **250+ DSA problems** across major coding platforms **LeetCode, HackerRank** and consistently applied **problem-solving patterns to backend architecture** and optimization tasks.
- Developed and contributed to **multiple backend systems on Github** as a second-year student, with emphasis on clean API design, modularity, and scalability.