Luxmikant

Aspiring Software Development Engineer

luxmikant@outlook.com — 7018209392 — github.com/luxmikant — linkedin.com/in/luxmikant

Professional Summary

Software Development Engineer with strong experience in designing, developing, and deploying scalable applications across the full technology stack. Proficient in modern frameworks and tools, with a solid foundation in algorithms, data structures, and system design.

Education

VIT Bhopal University, MP **CGPA: 8.38 — 2022 – 2026**

B.Tech in Computer Science Engineering, Specialization in Health Informatics

Relevant Coursework: Data Structures & Algorithms, Operating Systems, Database Management Systems, Computer Networks, Object-Oriented Programming, Software Engineering, Computer Architecture

Technical Skills

- **Languages:** C++, Rust,Python **Operating Systems:** Linux (command line, shell scripting)
- **Core Software Development:** Object-Oriented Programming (OOP), System Design, Software Development Life Cycle (SDLC), Application Definition, System Modeling, Release Planning and Development, Support and Maintenance
- **Backend & Systems:** REST API design, Actix-web (Rust), Flask (Python), Microservices, Concurrency (Tokio, Go routines), Memory Management, Docker, Containerization, API security (OAuth2/JWT)
- **Databases:** PostgreSQL, MySQL, MongoDB, query optimization,
- **Machine Learning:** PyTorch (basics), ML models (classification, regression), data analysis,EDA.
- **DevOps & Testing:** Experienced with Git/GitHub fundaments , CI/CD automation (GitHub Actions),

Experience

Open Source Contribution: Flamenco Web Configuration Manager **Jan 2025 - May 2025**

Core Contributor

- Collaborated a responsive Vue.js interface with Vuex, REST APIs, and WebSockets connected to a Go backend, reducing configuration time by 40%.
- Implemented an MQTT communication layer with MQTT.js and a Go broker for real-time messaging across 50+ nodes with 99.9% uptime.
- Orchestrated a task distribution system utilizing Go's concurrent processor pools and optimized thread allocation, diminishing average task completion latency by 150ms across 50+ nodes .
- Designed a cron-based Go scheduler with timezone awareness, optimizing global resource utilization by 25%.
- Created a cross-platform path translation layer for seamless deployment across Windows, Linux, and macOS.

Projects

FinBash - Full-Stack Financial Dashboard **Feb 2025 - Present**

Open Source Contributor

- Engineered a high-performance Rust backend with zero-cost abstractions, achieving 40% faster calculations with guaranteed memory safety.
- Implemented a concurrent architecture processing 500+ CSV records/sec, using Rust's borrow checker to eliminate data races at compile-time.
- Developed 7 modular components including a financial calculator, CSV parser, portfolio tracker, and async API handlers with Serde for serialization.
- Built an async RESTful API with Tokio, achieving ¡100ms response times for real-time portfolio updates.
- Established coding standards and documentation for integrating Rust backend that were adopted by 10+ engineers; elevated team code quality.

DICOM Image Processing Application **Dec 2024 – Mar 2025**

Full-Stack Developer

- Architected a full-stack DICOM platform (Python/Flask, React) processing 1000+ images, reducing radiologist workflow time by 30%.
- Transformed a secure microservices architecture for real-time image processing and comparison using Cornerstone.js.
- Modeled a responsive Material UI interface with intuitive tools, improving diagnostic efficiency by 40%.
- Containerized the application with Docker and MongoDB, ensuring HIPAA-compliant deployment.

Achievements

- Authored an AI research paper on Alzheimer's detection (published in Springer), developing a model that achieved a 92% accuracy rate in identifying early-stage biomarkers.
- Secured a Top 10 finish among 100+ teams at the Hack2Byte 3.0 hackathon through collaborative rapid prototyping.
- Solved 450+ algorithmic challenges on LeetCode, demonstrating strong command of data structures and algorithms.

Certifications

- Introduction to Rust Training — Ferrous Systems
- Applied Machine Learning in Python — University of Michigan
- Data Structures and OOP with C++: CS104, CS105 Masterclass