

# Luxmikant

## Intern Analyst Developer

✉ luxmikant@outlook.com — 📞 7018209392 — 🌐 github.com/luxmikant — 💬 luxmikant

## Luxmikant

- **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: 8.35  
B.Tech in Computer Science Engineering  
Specialization in Health Informatics  
2022–2026

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

## Leadership

- ★ Technical Lead, College Project Team
- ★ Organized technical workshops
- ★ Team Leader, Karnataka Police Datathon
- ★ Research Team Lead, Project Expo

## Achievements

- 🏆 Achieved "Specialist" rank on Codeforces with a peak rating of 1496
- 🏆 Early detection of alzheimer disease using ai research paper publish in springer
- 🏆 Solved 450+ problems on LeetCode
- 🏆 Under Top 10 teams in Hack2Byte 3.0 hackathon

## Certifications

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

## Skills

**Frontend:** React, JavaScript, HTML/CSS, Tailwind CSS, Material-UI, Redux

**Backend:** C++ (STL, Boost, Templates)

**Rust Programming:** Ownership & Borrowing, Lifetimes, Traits, Generics, Pattern Matching, Error Handling, Concurrency, Async (Tokio), Actix-web, , Cargo, Unit Testing, Docker

**Languages:** C++, Rust, python

**Soft Skills:** Leadership, Communication, Teamwork, Adaptability

## Experience

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

- Developed responsive Vue.js configuration interface with Vuex state management, integrating RESTful APIs and WebSocket connections to communicate with Go backend, reducing configuration time by 40%.
- Implemented MQTT-based communication layer using MQTT.js client and Go broker integration, enabling real-time bi-directional messaging between manager and 50+ distributed worker nodes with 99.9% uptime.
- Engineered dynamic task distribution system using Go's concurrent processor pools and worker capability matching, achieving 35% improved parallel rendering task allocation efficiency.
- Designed cron-based worker scheduling system in Go with configurable sleep/wake cycles and timezone awareness, optimizing resource utilization across global render farm deployments by 25%.
- Created cross-platform path translation layer handling Windows/Linux/macOS file systems through unified variable substitution, ensuring seamless deployment across heterogeneous environments.

## Projects

### MSKJ CRM Platform

Full-Stack Developer May 2024 – Present

- Engineered AI-powered CRM using React/Vite, Express/Node.js, and MongoDB with intelligent customer segmentation, campaign management, and OpenAI/Gemini integration for automated content generation.
- Built visual segment builder with drag-and-drop interface supporting complex rule-based filtering across 15+ customer attributes, reducing segment creation time by 70%.
- Implemented secure Google OAuth authentication, JWT token management, and RESTful API architecture with CORS configuration for distributed microservices deployment.
- Deployed containerized application on Render with automated CI/CD, achieving 99% uptime and serving production traffic at frontend-crm-u9i2.onrender.com.

### FinBash - Full-Stack Financial Dashboard

Open Source Contributor Nov 2024 – Feb 2025

- Engineered high-performance Rust backend using ownership system and zero-cost abstractions, achieving 40% faster financial calculations with guaranteed memory safety and compile-time error prevention.
- Implemented concurrent architecture processing 500+ CSV records/second using Rust's borrow checker, eliminating data races and null pointer dereferences at compile-time.
- Developed 7 modular components: financial calculator, CSV parser, portfolio tracker, price alerts, async API handlers, CLI interface, and JSON serialization with serde.
- Built async RESTful API with Tokio runtime achieving  $\leq 100\text{ms}$  response times for real-time portfolio updates and concurrent user session handling.
- Integrated full-stack solution connecting Rust backend with React/TypeScript frontend, demonstrating systems programming in modern web architecture.