Y Lakshmi Charan

+91 789 368 8725 | ylcharan.dev@gmail.com | ylcharan.dev | github.com/ylcharan

Education

Lovely Professional University Phagwara, Punjab
Bachelor of Computer Science and Technology, Minor: Web Development (7.0/10) *Aug. 2023 – Currently*
Relevant Coursework: Data Structures and Algorithms (Leetcode) , Object-Oriented Programming, Computer Networks, Database Management Systems (SQL, relational modeling), Web Development (MERN Stack)

Skills

Languages Java, Python, JavaScript, SQL, C/C++
Backend FastAPI, Node.js, Next.js, REST APIs, PostgreSQL, MySQL, Supabase, Clerk, Netty
Frontend React.js, Tailwind CSS, GSAP, HTML/CSS, Three.js
Core CS Data Structures, Algorithms, OOP, Complexity Analysis, OS Basics, CN, RL Basics, ML Basics

Experience

Freelance Frontend Developer — Remote June 2024 – present
Front-end Development

- Designed and implemented modular, scalable frontend architectures using React.js and Next.js, integrating RESTful APIs with clearly defined contracts to support maintainability and future feature expansion.

Projects

Mini Redis - Redis-Compatible Distributed Key-Value Store source code

- Built a Redis-compatible TCP server implementing the RESP protocol, enabling seamless interaction with `redis-cli` and standard Redis clients.
- Implemented 30+ commands covering strings, key management, TTL/expiry, and server operations using a thread-safe `ConcurrentHashMap` with O(1) reads.
- Designed WAL-based persistence using `FileChannel` with DSYNC for crash recovery and implemented LRU eviction via `LinkedHashMap(accessOrder)`.
- Extended system with distributed features including primary-replica replication (PSYNC) and consistent hashing with 150 virtual nodes for efficient key routing.

Advanced Thread Pool & Scheduler - Concurrency Engine source code

- Engineered a production-grade thread pool from scratch without relying on `ExecutorService` or `ForkJoinPool`.
- Implemented multiple scheduling strategies including FIFO, Priority, Scheduled, and Work-Stealing using appropriate concurrent data structures.
- Built a custom `TaskFuture<T>` using `LockSupport.park/unpark` with a lock-free waiting-thread mechanism.
- Added dynamic worker scaling based on queue depth with hysteresis control to prevent thrashing under burst workloads.

LearnSmart - AI-Powered Voice Learning Platform source code

- Built a full-stack AI learning platform where users create personalized voice companions with custom names, subjects, and personalities.
- Integrated VAPI for real-time voice conversations, enabling interactive subject-specific learning sessions.
- Implemented Clerk authentication and Supabase backend to persist user profiles, configurations, and session history.
- Added Sentry-based monitoring for client and edge runtime error tracking in production environments.

Beauty & Wellness - Full-Stack E-Commerce Platform | link source code

- Developed a full-stack MERN application with a decoupled architecture and RESTful APIs for product and service management.
- Implemented secure authentication and authorization workflows for protected user interactions and data access.
- Designed scalable API endpoints ensuring efficient client–server communication with MongoDB-backed persistence.
- Deployed the system with React frontend on Vercel and Node.js/Express backend on Render with environment-based configuration.