Kiran S

6380733284 | kiranvelan444@gmail.com | GitHub | LinkedIn | Chennai, Tamil Nadu, India

Summary

Website

Senior year Computer Science student specializing in backend development and system design. Proficient in Spring Boot and FastAPI with experience building scalable full-stack applications through innovative projects. Looking for a software developer role to contribute technical expertise and drive future-ready software solutions.

Education

B. Tech in Computer Science Engineering 2022 - 2026
Indian Institute of Technology, Jodhpur
GPA: 8.35/10.00

Skills

Languages: Python, Java, JavaScript, SQL
Frameworks: Spring Boot, FastAPI, React, Node.js, Retrieval-Augmented Generation (RAG)
Databases: MySQL, Redis, Appwrite Collections
Tools & Practices: Git, Docker, Postman, IntelliJ, VS Code, Appwrite
Software Engineering Concepts: System Design, Concurrency, Unit Testing, Microservices, Version Control System, CICD pipeline, RESTful API Design, Performance Tuning.

Experience

SwarmIQ – Real-Time Group Decision-Making and Opinion Gathering Platform GitHub
Founder & Full-Stack Developer | Mar 2025 - Present **(Self-Initiated Start Project)**
Technologies used: Java, Spring Boot, WebSocket, Rest API, Thymeleaf, JavaScript, FastAPI Microservice, MySQL

- **Problem**: Traditional surveys and polling tools lack real-time collaboration and often fail to capture consensus effectively.
- **Solution**: Developed a full-stack real-time platform for decentralized group decision-making, leveraging Artificial Swarm Intelligence and Particle Swarm Optimization (PSO) to simulate swarm behavior and guide user consensus.
- Built backend services in Spring Boot and Python (FastAPI) for modular communication between WebSocket-based real-time control and PSO computation microservices.
- Engineered a multi-round question engine supporting up to 20 options per question, with customizable option counts per round (default 6), and built-in convergence scoring to guide collective decisions.
- Maintained API response times under 150ms with CPU usage below 35% during peak sessions, optimized performance using MySQL indexing and modular MVC structure.
- Enhanced decision quality and usability for facilitators through clear visualizations and session reports.
- Collaborating with an Associate Professor Mr. Krishna Kumar Balaraman from the SME IIT Jodhpur to align product goals with real-world group decision-making research.
- Project is not open-sourced due to startup confidentiality; currently exploring productization and planning a potential launch.

Projects

SmartScribe – AI-Powered Media Transcription and Interactive Q&A Platform GitHub
Technologies used: Python, Fast API, Node.js, React, Tailwind CSS, Whisper, Appwrite, Docker, Hugging Face, Ollama LLM

- **Problem**: Learners often struggle to retain or revisit important concepts from long videos due to lack of searchability and interaction.
- **Solution**: Created a web-based tool that transcribes YouTube/audio/video content and supports interactive question-answering using context-aware retrieval.
- Implemented a FastAPI backend with modular routes for processing uploads, generating context, and handling retrieval queries. Used Appwrite to manage user sessions and chat history and containerized the app with Docker for deployment.
- Refined a RAG (Retrieval-Augmented Generation) pipeline with Hugging Face embeddings and optimized chunked vector search to improve context relevance.
- Used OpenAI Whisper for transcription and local LLMs (Mistral via Ollama) for question answering, achieving 70–90% accuracy on 15+ videos of 5–20 minutes.
- Reduced end-to-end response times to under 90 seconds, with live Q&A responses under 5 seconds on CPU.

API Rate Limiting Gateway GitHub
Technologies used: Spring Boot, Redis, React, Chart.js, Middleware Architecture

- **Problem**: APIs exposed to clients, risk abuse and uneven resource consumption without a robust throttling mechanism.
- **Solution**: Built a configurable API rate-limiting gateway with real-time monitoring and multiple throttling strategies.
- Designed middleware architecture to allow pluggable custom strategies and seamless integration with backend services.
- Implemented Fixed Window and Token Bucket algorithms using Redis for time-bucket tracking and counter resets.
- Developed a secure admin dashboard using React and Chart.js for visualization and live configuration of limits, and Redis persisted request logs and usage data for analytics and audit purposes.
- Supported over 3,200 requests per minute under stress tests with decision latency under 10ms. Designed with modularity and extensibility in mind to support future integration with authentication and analytics systems

Relevant Coursework

Data Structures and Algorithms – A
Software Engineering
Database Systems – A-
Computer Networks – A
Operating System – A-
Cybersecurity