

POTHURAJU VYSHRAWAN

Backend Software Engineer

Java • Spring Boot • Kafka • Microservices • Distributed Systems • AI-Integrated Systems

+91-7671942792 | vyshrawanp@gmail.com | linkedin.com/in/vyshrawan-p-795b5b256 | github.com/VyshrawanP | vyshrawan.in

Summary

Final-year Computer Science and Business Systems (CSBS) student at VIT-AP passionate about building secure, scalable backend systems and distributed applications. Experienced with Java, Spring Boot, Microservices, Apache Kafka, PostgreSQL, Redis, Docker, and AWS, with hands-on experience implementing SAGA transactions, event-driven architectures, and AI-powered features using Spring AI and RAG. Seeking Backend Software Engineer opportunities for 2026/2027 to build reliable, high-performance systems and solve real-world engineering challenges.

Skills

Languages: Java, Python, SQL, HTML5, CSS3, JavaScript, TypeScript

Frameworks: Spring Boot, Spring MVC, Spring REST, Spring Security, Spring Data JPA, Spring AI, FastAPI, LangChain, LangGraph

Databases & Caching: MySQL, PostgreSQL, Redis, ChromaDB, JPA/Hibernate

Messaging & Streaming: Apache Kafka, Server-Sent Events (SSE)

Cloud & Infrastructure: AWS (EC2 fundamentals), Infrastructure as Code (IaC) basics, Docker, Git, GitHub Actions, Maven

System Design Fundamentals: Load Balancing, DNS / Route 53, Caching Strategies, Horizontal Scaling, Rate Limiting

Security & Standards: OWASP Top 10, Secure Coding Standards, JWT-based Auth, Input Validation & Sanitization

Testing & Tooling: JUnit, Mockito, pytest, Apache JMeter, Postman

Operating Systems: Windows, macOS, Linux

AI Tooling: Prompt Engineering, RAG, Spring AI, Multi-Agent Orchestration

Architecture & Patterns: RESTful APIs, Microservices, SAGA Orchestration, CQRS, Transactional Outbox, CI/CD

Projects

Digital Banking System – Microservices Clearing Platform | *Java, Spring Boot, Kafka, Redis, Docker* **Personal Project**

- Designed a microservices-based digital banking backend with six independent Spring Boot services (API Gateway, Account, Transaction, Payment, Fraud Detection, Notification), orchestrated via Docker Compose.
- Implemented a multi-step SAGA orchestration pattern for the transaction clearing engine, including compensatory rollback logic for failed or rejected transfers to guarantee consistency without distributed locking.
- Built a Kafka-driven fraud detection service performing live velocity and risk analysis, with an OTP-based step-up verification flow triggered for transfers exceeding compliance thresholds.
- Integrated a Razorpay-style payment hub simulation for checkout and webhook processing, and an API Gateway layer handling centralized routing and rate limiting across all services.

CodeSentinel – Java-Native Agentic RAG System | *Java, Spring Boot, ChromaDB, Groq* **Production-Deployed**

- Built an end-to-end Retrieval-Augmented Generation pipeline in Spring Boot using ChromaDB for vector storage and Groq Llama-3.3-70B for response generation, with Redis caching and Docker containerization.
- Implemented chunking, embedding, and semantic retrieval logic from scratch without LangChain, to maintain full control over the RAG pipeline and demonstrate first-principles system design.
- Engineered guest session isolation with automatic stale-session cleanup, plus custom rate limiting (per-endpoint request windows with proper 429/Retry-After handling) to harden upload and query endpoints against abuse.
- Exposed REST APIs for document ingestion, semantic query processing, and retrieval; deployed to production with CI/CD via GitHub Actions and Docker Compose.

AURA AI – Multi-Agent Research & Reasoning Platform | *Python, FastAPI, LangGraph, ChromaDB, React, Docker* **Personal Project**

- Built a LangGraph StateGraph orchestrator coordinating seven specialized agents (Planner, RAG, Research, Data Analyst, Code, Critic, Synthesizer), streaming live execution traces to a React 19 + TypeScript frontend.
- Implemented a hybrid RAG pipeline with query decomposition, BM25 + dense vector fusion over ChromaDB, and a weighted reranker, ingesting PDF, DOCX, CSV, XLSX, and Markdown with metadata-preserving chunking.
- Designed a sandboxed Python execution layer using AST static analysis to block dangerous imports (`os`, `subprocess`, `socket`, `eval`) with a hard 5s timeout, and wrapped retrieved content in delimited context boundaries as a prompt-injection defense.
- Added a critic loop that scores drafts against retrieved evidence and triggers re-retrieval below a 0.85 threshold, plus a citation validator tagging claims as supported, partially supported, or unsupported; covered by a pytest suite and a Docker Compose stack (FastAPI, PostgreSQL, ChromaDB, Redis).

Additional Engineering Practice

- **Authentication & Spring Security Internals:** Built a complete JWT authentication system from first principles – custom `UserDetailsService` backed by PostgreSQL, BCrypt password hashing, and a hand-written JWT validation filter wired into the Spring Security filter chain.
- **Industry-Standard Secure Coding:** Hands-on offensive-security practice via OWASP WebGoat and OWASP ZAP, covering SQL injection exploitation and remediation. Applies OWASP Top 10 guidelines, parameterized queries, input validation, and least-privilege database access as standard practice across all projects, not as an afterthought.
- **System Design Fundamentals:** Comfortable reasoning about load balancers, DNS resolution (Route 53), horizontal scaling, and caching strategies; authored a system-design breakdown of soft reservation with TTL-based rollback (Redis SETNX + TTL) to solve race conditions in high-concurrency booking flows, referencing patterns used by Ticketmaster and BookMyShow.
- **Cloud & Production Readiness:** Working knowledge of AWS EC2 for compute provisioning and Infrastructure as Code (IaC) fundamentals for repeatable deployments; comfortable operating across Windows, macOS, and Linux environments for development and deployment workflows.
- **AI-Assisted Engineering:** Practical experience with prompt engineering to accelerate development workflows (debugging, boilerplate, documentation) while maintaining full ownership of architecture, correctness, and production-readiness decisions.

Business & Product Perspective

- CSBS coursework blends core computer science with business systems, giving a working ability to reason about technical decisions in terms of cost, scalability, and time-to-market – not just correctness – and to communicate engineering trade-offs clearly to non-technical stakeholders.

Certifications & Achievements

- **JPMorgan Chase Virtual Software Engineering Internship** – completed via Forage, covering software engineering modules.
- **Security Research:** responsible disclosure of a Firebase API key exposure and a Supabase Row-Level Security misconfiguration on production applications.

Education

VIT-AP University, Amaravati

Bachelor of Technology, Computer Science and Business Systems (CSBS)

Sep 2023 – Oct 2027

Grade: 8.01

Sri Chaitanya Junior College, DD Colony, Hyderabad

Senior Secondary Education – Mathematics, Physics, Chemistry

Apr 2021 – Mar 2023

Grade: 9.08

St. Joseph's High School, Hyderabad

Secondary School Certificate (SSC)

Jun 2009 – Apr 2021

Grade: 10