VAISHNAV SHELKE

vaishnavshelke22@gmail.com • 9765233984 • https://www.linkedin.com/in/vaishnav-shelke

I am a passionate and results-driven software engineer. While working in payments industry I gained experience in building a **high-volume**, **low-latency multitenant** Spring Boot application for quick transaction routing. My work on payment services introduced me to **asynchronous** and **event-driven programming**, allowing me to build both **asynchronous-reactive** and **synchronous web APIs**. By troubleshooting production issues, I developed a deep understanding of the **microservice architecture** used by authentication services, including their intercommunication and distribution of responsibilities. My experience in the payment industry also taught me about data security measures such as **HTTP SSL** and payload-level security using **Message Level Encryption** and **JWE**

PROFESSIONAL EXPERIENCE

Associate Software Engineer @ **PayU Wibmo, Bengaluru**

June-2022 – Present

Project: Expresspay (New Product)

- Built a card payment routing product from scratch that authenticates cards on the merchant's page by accepting OTPs, eliminating the need for users to be redirected to the bank's page.
- Developed synchronous Java Spring boot RESTful APIs for payment aggregators and implemented SSL-secured HTTP clients to interface with card networks.
- Reduced transaction routing time by multithreading and caching using both distributed Redis and local Caffeine cache.
- Used the various encryption techniques to protect payload and card data.
- Created a multicast socket was to enable the application to receive messages sent by other services using the (UDP) Protocol.
- Implemented jobs to refresh the cache and to create and update tables.
- The product hence built was tested for max load of 300 requests per second on locust load testing tool.

Keywords: • Core Java • Spring boot • Redis • JPA/JDBC Template • MySQL • Caffeine • Multicast Messaging •SSL secured connections

Project: 3DSS (Legacy Product)

- Performed Junit testing on the transaction APIs. SQL clients, HTTP calls, Kafka publishers, and API calls to other services were mocked using Mockito to inject mock values into them, achieved 70%+ coverage.
- Built a reactive RESTful API using Java Vert.x, an asynchronous and event-driven programming library built in Java, that handled 170 requests per second during sale season.
- Resolved various issues that arose in the production environment. This involved identifying the root cause of the problem and implementing a solution to prevent it from happening again.
- Performed SonarQube analysis on the code to fix critical bugs.
- Helped various aggregators integrate with our payment gateway by providing them with product and API integration walkthroughs. I also provided on-call assistance to clients to help them overcome any difficulties they encountered.

Keywords: • Java Vert.x • Reactive • Event-Driven • Multithreaded • Kafka • Junit Testing • Mockito • Microservice Architecture • Distributed System

SKILLS

- Have deep interest in different data structures and their implementation with Algorithms to develop and optimize applications and features.
- Spring Boot, MySQL, Java , MySQL, Redis.
- Have exposure to working in Linux Shell Environment to deploy application in integration and UAT environments.
- Experience with version control systems like Git

EDUCATION

1. Bachelor of Technology in Mechanical Engineering, **National Institute of Technology** – Nagpur, 2022.
   CGPA: 8.65
2. Scored in the top **99.3 percentile** in JEE Mains 2018 and obtained a **90.32%** in Maharashtra 12th board exams.

PERSONAL PROJECTS

1. WhatsApp Analyzer:
   Built a tool that scans the chats on WhatsApp web to generate reports giving interesting insights into the user's behaviour pattern in group chats. Used python **web-application automation** library **Selenium** and **web scraping** library Beautiful Soup to automate logging in, scanning group chats to parse the **HTML** and to get a DOM Object to extract features such as chats, emojis, date etc, to perform analysis on. Developed **backend** in **Node.js** to run the **python** script and return the responses to the **REST** calls made by **React.js Frontend**. Used **Charts.js** library, to make the React frontend more appealing and infographic.
   GitHub: WhatsApp-Analyser