Pravallika Annapareddy
Software Development Engineer

Baltimore, MD USA | Phone no: (551) 465-6319 | E-mail: pravallika.jbm@gmail.com | LINKEDIN

SUMMARY

- Experienced Software Development Engineer with a 5+ years track record of designing, developing, and deploying scalable and secure applications using a range of technologies, including Java, Python, cloud services (AWS), and containerization technologies (Docker, Kubernetes).
- Proficient in microservices architecture, API development, and database management.
- Expertise in developing and implementing RESTful APIs using frameworks such as Spring Boot, Flask, with a focus on performance optimization, security best practices and seamless integration with diverse systems.
- Proficient in leveraging cloud platforms (AWS) for scalable deployments, with hands-on experience in containerization (Docker) and orchestration (Kubernetes) to streamline application management and improve deployment efficiency.
- Skilled in data modeling, querying, and ensuring data integrity across various database technologies, including relational (MySQL, PostgreSQL) and NoSQL (MongoDB, Elasticsearch) databases.
- Versatile leader skilled in collaborating with cross-functional teams to translate complex business requirements into innovative, mission-critical software solutions.

SKILLS

Back-End Technologies	Java, Python, RESTful API, Webservice, Microservices, Spring Boot, Spring MVC, Spring Data JPA, SQLAlchemy, Hibernate ORM, DSA, Django, Flask, Asyncio, Multiprocessing, Pandas, NumPy, PySpark, SciPy, Machine Learning.
Front-End Technologies	HTML, CSS, UI Material, JavaScript, React JS, React Native, React Hooks, Typescript, SASS, Angular, ES6
J2EETechnologies	Servlets, JSP, JDBC, JMS, Maven, Multi-threading
Architecture	Microservices, MVC, MVVM, TDD, BDD
Methodology	Agile, Scrum, Kanban, Waterfall.
IDEs	NetBeans, IntelliJ IDEA, Eclipse, Visual Studio, PyCharm
Databases	SQL, Elasticsearch, MongoDB (NoSQL), PostgreSQL, MySQL
CI/CD & Tools	Jenkins, Code Ship, GitLab CI/CD
Testing tools	Mockito, Jest, Selenium, JUnit, TestNG, Espresso, PyUnit, Pytest, xUnit, Moq.
Cloud & Deployment	AWS, Docker
Web Server	Apache Tomcat, WebLogic
Other Tools	Jira, GitHub, Bitbucket, Swagger, Open API, GraphQL, SonarQube, Postman, Kafka
Operating System	Windows, LINUX, Mac OS

EDUCATION

Master of Science in Systems Engineering - University of Maryland, Baltimore County, USA
Master of Technology in Mining Engineering – National Institute of Technology, Rourkela, India
Bachelor of Technology in Civil Engineering - Mahatma Gandhi Institute of Technology, JNTUH, India

CERTIFICATION

Scrum Master Certification from Scrum Alliance, Inc.
Link: https://bcert.me/bc/html/show-badge.html?b=krjrphlv

WORK EXPERIENCE

Software Development Engineer | Capital One, MD **July 2023 – Present**

- Designed and implemented microservices-based architecture using Python frameworks (Django) to decompose the monolithic financial application into independent, scalable, and maintainable services (account management, transaction processing, credit scoring), improving system flexibility and development speed.
- Wrote and optimized complex SQL queries in Python, using libraries such as SQL Alchemy to retrieve transaction history, customer account information, credit scores, and loan details, improving query performance by 30% using indexing, query optimization, and efficient joins.

- Engineered and maintained a mission-critical trading platform for the firm's equity derivatives desk—part of the "High- Frequency Trading Platform Enhancement" project—by developing core trading logic with Java Spring Boot. This solution delivered low-latency order execution, contributing to a 20% reduction in communication latency and a 10% increase in trading volume.
- Collaborated closely with quantitative analysts to refine pricing models and risk management tools. By using Java for high- performance core calculations alongside Python (with Pandas) for data analysis, you helped optimize algorithmic trading strategies, improving overall trading performance by 15%.
- Architected and implemented a distributed, fault-tolerant order management system, used microservices architecture for the firm's equities trading desk. Utilized Spring Cloud for service discovery and configuration management.
- Developed RESTful APIs using Spring Boot and Spring MVC to facilitate communication between different components of the trading platform, enabling seamless integration with frontend and external systems. Maintained some Python-based microservices and APIs using Flask.
- Developed a hybrid Java/Python solution for real-time option pricing, utilizing Java for core pricing calculations and Python for efficient data processing and model calibration, significantly improving pricing accuracy.
- Implemented and optimized algorithmic trading strategies for various asset classes, including equities, fixed income, and currencies. Utilized Java for core trading logic and employed Python with Pandas for data analysis and strategy back testing.
- Developed high-performance Java components within the trading platform using Spring Boot and Spring Data JPA, implementing efficient caching strategies with Redis to minimize database load for critical market data access.
- Utilized AWS Lambda for serverless computing to execute code in response to events, optimizing specific trading processes and reducing operational overhead by 25%, which enabled efficient and scalable processing of trading data.
- Created Python-based monitoring tools using libraries like Prometheus and Grafana to track the performance and health of the trading platform, providing real-time insights into key metrics and facilitating proactive issue resolution.

Software Developer | Infosys, India **April 2019 – Aug 2022**

- Designed and implemented dynamic, responsive user interfaces using React.js, enhancing the user experience and reducing page load times by 25%, which contributed to higher user satisfaction and engagement.
- Developed and maintained RESTful APIs and business logic using Java and Spring Boot, enabling secure user authentication, efficient transaction processing, and seamless integration with third-party financial systems.
- Optimized backend performance by implementing efficient data processing techniques and caching strategies, reducing response times and improving the scalability of financial transactions.
- Planned and implemented microservices architecture for digital banking application, breaking down complex business processes such as user account management, & fraud detection into independently deployable services.
- Sketched MongoDB collections and documents to efficiently store and query complex data, including user accounts, financial transactions, and customer activity, ensuring fast read and write operations.
- Applied best practices for API security, such as OAuth for secure user authentication, ensuring compliance with industry standards and safeguarding sensitive user financial data, resulting in a 30% reduction in security vulnerabilities and enhancing data protection protocols.
- Integrated Elasticsearch, Logstash, and Kibana (ELK stack) for real-time logging, error tracking, and monitoring, achieving 98% system reliability and reducing downtime, ensuring a seamless user experience.
- Implemented asynchronous processing in Spring Boot using @Async to handle long-running tasks like transaction validation and fraud detection in a non-blocking manner, improving system performance and responsiveness.
- Continuously enhanced the frontend and backend of the application, improving load times, reducing complaints by 25%, and enabling a smoother, more engaging user experience.
- Deployed the digital banking application on AWS to take advantage of scalable, reliable, and cost-effective cloud infrastructure, ensuring high availability and seamless performance under varying loads.
- Set up automated rollback mechanisms in the CI/CD pipeline, ensuring that if a deployment failed at any stage, the system would automatically revert to the previous stable version.
- Used Mockito to mock asynchronous operations, such as transaction validation or fraud checks, that might involve external systems or long-running processes.
- Implemented Test-Driven Development (TDD), ensuring code quality and reducing bugs during development.
- Developed Python modules for high-throughput data processing within the Kafka pipeline, improving real-time data ingestion and analysis by 20% for specific data streams.
- Utilized React components to create reusable, modular UI elements, contributing to a consistent and maintainable codebase.
- Utilized advanced SQL skills within Python applications for deep data analysis and data cleaning, crafting intricate PostgreSQL queries that significantly reduced data retrieval and processing times.