Chidambar Rao Serusanagandla
chidambarraos12@gmail.com | **+ 1 (940)-290-1606**
Linkedin | GitHub | Portfolio | Medium
Location: Texas, United States

Professional Summary:

- Experienced Software Developer with around 8 years of expertise in designing and building enterprise-level applications across banking, healthcare, retail, and financial services.
- Proficient in Java, Spring Boot, Microservices, RESTful APIs, ReactJS, and Angular, developing scalable and high-performance applications.
- Strong expertise in cloud technologies including AWS, GCP, and Azure, with hands-on experience in serverless computing, Lambda functions, and cloud-based data storage.
- Experienced in microservices architecture, using Spring Cloud, Kubernetes, and Docker to develop modular and scalable applications.
- Designed and implemented CI/CD pipelines using Jenkins, GitHub Actions, and AWS CodePipeline, optimizing deployment workflows and reducing time-to-market.
- Extensive experience with relational and NoSQL databases (PostgreSQL, MySQL, MongoDB, and Cassandra), ensuring optimized performance through indexing and query tuning.
- Implemented real-time messaging systems using Apache Kafka and RabbitMQ, ensuring efficient event-driven architecture.
- Strong background in security implementation using OAuth2, JWT authentication, Spring Security, and RBAC (Role-Based Access Control).
- Applied Agile methodologies including Scrum and Kanban, actively participating in sprint planning, backlog grooming, and daily stand-up meetings.
- Experienced in Test-Driven Development (TDD) using JUnit, Mockito, Cypress, Jest, and Selenium to maintain high-quality and defect-free applications.
- Designed API Gateway solutions, optimizing request handling, security, and performance for large-scale distributed applications.
- Implemented caching strategies using Redis, improving application performance and reducing latency.
- Developed feature toggling mechanisms for phased deployments and A/B testing, ensuring smooth feature rollouts.
- Collaborated with cross-functional teams, including UI/UX designers, DevOps engineers, product managers, and QA teams, to deliver robust and scalable solutions.
- Passionate about continuous learning, staying updated with emerging technologies and best coding practices to improve development efficiency.

Technical Skills:

Programming Languages	Java 8/11/17, JavaScript (ES6+), TypeScript, Python
Frontend	ReactJS, Angular, HTML5, CSS3, Bootstrap, Redux
Backend	Spring Boot, Spring MVC, Spring REST, Spring Data JPA, Hibernate, Node.js
Databases	PostgreSQL, MySQL, Oracle, MongoDB, Cassandra.
Cloud Platforms	AWS, GCP, Azure
Messaging & Streaming	Apache Kafka, RabbitMQ
Containerization & DevOps	Docker, Kubernetes, Jenkins, Terraform, AWS CodePipeline
Testing	JUnit, Mockito, Cypress, Jest, Selenium
Version Control & CI/CD	Git, Bitbucket, SVN (one per project), GitHub Actions
Build Tools	Maven, Gradle
Agile & Project Management	Jira, Scrum, Confluence

Professional Experience:

Client: First Citizens Bank, Los Angeles, CA **April 2023 - Present**
Role: Senior Java Full stack Developer

Project Description - Secure Banking Platform

At **First Citizens Bank**, I worked on developing a **Secure Banking Platform**, a robust and scalable banking application designed to handle **customer transactions, fund transfers, loan processing, and account management** securely and efficiently. The project focused on **modernizing the bank's digital infrastructure** by leveraging **microservices architecture, cloud computing, and real-time event-driven processing** to enhance transaction security, improve performance, and ensure compliance with financial regulations.

Responsibilities:

- Involved Designed a scalable microservices architecture using Spring Boot and Spring Cloud, ensuring seamless, modular, and loosely coupled service communication for various banking operations.
- Developed and deployed RESTful APIs for handling customer transactions, fund transfers, account creation, and loan processing, optimizing data flow between front-end and back-end services.
- Integrated OAuth2 authentication and JWT tokens to ensure secure user sessions, role-based access control (RBAC), and multi-factor authentication (MFA), enhancing banking security compliance.
- Implemented event-driven architecture using Apache Kafka for real-time transaction updates, fraud detection, and account activity notifications, reducing latency in financial transactions.
- Built a ReactJS-based banking dashboard for real-time monitoring of transactions, fund transfers, and fraud alerts, improving customer and employee user experiences.
- Utilized PostgreSQL with query optimization techniques (indexing, partitioning, caching) to ensure fast retrieval of transactional data, maintaining ACID compliance and high availability.
- Deployed microservices on AWS Kubernetes (EKS) using Terraform for infrastructure automation, ensuring efficient container orchestration and high system uptime.
- Automated CI/CD pipelines using Jenkins, incorporating automated testing, security scans, and deployment rollbacks, significantly reducing deployment errors and downtime.
- Developed feature toggling using LaunchDarkly, enabling A/B testing and controlled rollouts of new features without system downtime, improving user experience.
- Integrated Redis as an in-memory caching solution, reducing database load and improving API response times for frequently accessed user and transaction data.
- Secured APIs using API Gateway and rate-limiting strategies, preventing security vulnerabilities such as SQL injection and distributed denial-of-service (DDoS) attacks.
- Implemented financial reconciliation and reporting services, allowing seamless integration with third-party banking partners and regulatory agencies.
- Utilized GraphQL to optimize API calls, reducing payload size and improving front-end performance by fetching only required banking data.
- Designed and implemented monitoring solutions using AWS CloudWatch and ELK Stack (Elasticsearch, Logstash, Kibana), enabling real-time logging, alerting, and system health checks.
- Developed batch processing jobs using Spring Batch to handle large-scale financial data processing, ensuring high availability and consistency of batch transactions.
- Integrated third-party payment gateways (Stripe, PayPal, and Plaid) to allow seamless online transactions and direct bank integrations for automated fund transfers.
- Led sprint planning, backlog grooming, and daily stand-ups in Agile methodology, ensuring smooth project execution and continuous delivery.
- Optimized front-end performance using code-splitting, lazy loading, and Redux state management, ensuring a seamless user experience on web and mobile platforms.
- Conducted unit testing and integration testing using JUnit, Mockito, Cypress, and Jest, improving application stability and reducing production issues.
- Provided mentorship and knowledge-sharing sessions on best practices in Java development, microservices, and DevOps tools, enhancing team productivity and skill development.

Environment: **Backend:** Java, Spring Boot, Spring Cloud, Spring Security, Hibernate, RESTful APIs, GraphQL, ReactJS, Redux, JavaScript, Material UI, PostgreSQL, Redis (caching), AWS (EKS, S3, RDS, Lambda, CloudWatch), Kubernetes, Docker, Terraform, Jenkins, Apache Kafka, RabbitMQ, OAuth2, JWT, API Gateway, Role-Based Access Control (RBAC), JUnit, Mockito, Cypress, Jest, ELK Stack (Elasticsearch, Logstash, Kibana).

Client: NextGen Healthcare, Atlanta, GA **February 2021 - March 2023**
Role: Java Full Stack Developer

Project Description - Healthcare Management System

At **NextGen Healthcare**, I worked on the development of a **Healthcare Management System**, an advanced platform designed to streamline **patient management, appointment scheduling, medical records storage, billing, and insurance claim processing**. The system was built to **enhance healthcare accessibility, improve operational efficiency, and ensure compliance with HIPAA regulations** by leveraging **microservices architecture, cloud computing, and secure API integrations**.

Responsibilities:

- Designed and implemented a scalable microservices architecture using Spring Boot and Spring Cloud, ensuring modular, secure, and high-performance healthcare services.
- Developed RESTful APIs for patient registration, appointment scheduling, medical record retrieval, billing, and insurance claim processing, enabling seamless healthcare operations.
- Built an Angular-based UI for doctors, patients, and healthcare providers, offering an intuitive interface for managing patient appointments and records.
- Integrated AWS Lambda functions to automate appointment scheduling, patient notifications, and medical reminders, improving patient engagement and reducing manual intervention.
- Implemented OAuth2 authentication and JWT-based access control to ensure HIPAA compliance, encrypting patient health information and safeguarding sensitive data.
- Optimized PostgreSQL database performance by implementing partitioning, indexing, and query tuning, reducing data retrieval time for large-scale medical records.
- Deployed containerized microservices using Docker and Kubernetes on AWS, ensuring scalability, fault tolerance, and high availability of healthcare applications.
- Integrated Apache Kafka for asynchronous messaging between patient records, appointment scheduling, and billing systems, ensuring real-time data consistency.
- Automated CI/CD pipelines with AWS CodePipeline and Jenkins, enabling seamless deployments with version control, rollback mechanisms, and automated testing.
- Implemented Elasticsearch for fast and efficient medical record searches, reducing query execution time and improving the efficiency of patient data retrieval.
- Designed and implemented role-based access control (RBAC) to restrict access to patient data based on roles such as doctors, nurses, admin staff, and insurance providers.
- Integrated RabbitMQ as a message broker to handle asynchronous tasks, such as sending lab results, prescription updates, and real-time notifications to patients.
- Developed GraphQL APIs to optimize data fetching, allowing the frontend to request only the required data, improving application responsiveness.
- Implemented a secure API Gateway to manage authentication, rate limiting, and logging of API requests, ensuring protection against unauthorized access.
- Designed and implemented a data archival strategy for long-term storage of medical records using AWS S3 and Glacier, ensuring compliance with healthcare regulations.
- Developed unit and integration tests using JUnit, Mockito, and Cypress, ensuring robust testing coverage and minimizing production issues.
- Created monitoring dashboards using AWS CloudWatch and ELK Stack (Elasticsearch, Logstash, Kibana) to track API performance, error rates, and system health.
- Optimized UI performance by implementing lazy loading, server-side pagination, and state management using Redux, improving application speed and usability.

- Developed batch processing jobs with Spring Batch to handle bulk patient data processing, insurance claims, and report generation.
- Collaborated with cross-functional teams, including doctors, healthcare administrators, and data security teams, to gather requirements and ensure application compliance with medical industry standards.

Environment: Java, Spring Boot, Spring Cloud, Spring Security, Hibernate, RESTful APIs, GraphQL, Angular, TypeScript, Material UI, PostgreSQL, Elasticsearch (search optimization), AWS (Lambda, S3, RDS, CloudWatch), Docker, Kubernetes, Terraform, Jenkins, AWS CodePipeline, Apache Kafka, RabbitMQ, OAuth2, JWT, API Gateway, HIPAA-compliant encryption, JUnit, Mockito, Cypress, Jest, ELK Stack (Elasticsearch, Logstash, Kibana).

Client: Office Depot, Boca Raton, FL **January 2019 – January 2021**
Role: Full Stack Developer

Project Description: E-commerce Order Processing System

At **Office Depot**, I worked on developing an **E-commerce Order Processing System**, a high-performance platform designed to manage **customer orders, payment processing, inventory tracking, and real-time order status updates**. The goal of the project was to enhance the **efficiency, scalability, and user experience** of the order processing system by leveraging **microservices architecture, event-driven processing, and cloud deployment**.

Responsibilities:

- Developed a microservices-based order processing system using Spring Boot and Spring Cloud, ensuring high availability, scalability, and modular services for handling orders, payments, and inventory.
- Designed and implemented RESTful APIs for managing customer orders, product inventory, and user authentication, enabling seamless integration between the front-end and back-end services.
- Created a ReactJS-based order tracking dashboard, allowing customers to monitor real-time order status, estimated delivery times, and payment history, improving customer engagement.
- Integrated Apache Kafka as an event-driven messaging system to handle real-time order status updates, ensuring asynchronous communication between microservices such as order fulfillment, inventory management, and payment processing.
- Deployed containerized microservices on AWS ECS (Elastic Container Service) with auto-scaling capabilities, ensuring efficient resource utilization and system performance.
- Implemented MongoDB indexing and sharding techniques, optimizing query performance for order retrieval, customer purchase history, and product catalog management.
- Automated deployment pipelines using GitHub Actions, enabling continuous integration (CI) and continuous deployment (CD), reducing manual intervention and deployment downtime.
- Developed an authentication and authorization system using OAuth2 and JWT, securing APIs and ensuring role-based access control (RBAC) for different types of users (customers, admins, and vendors).
- Integrated Stripe and PayPal payment gateways, allowing customers to make secure transactions using multiple payment methods, enhancing the e-commerce platform's revenue capabilities.
- Designed and optimized API Gateway configurations, enabling load balancing, request throttling, and security enforcement, improving API response time and reliability.
- Implemented Redis caching to optimize session management, reducing load on MongoDB and improving response times for frequently accessed customer order data.
- Designed and built an inventory synchronization service that updates real-time stock availability across multiple warehouses, preventing order failures due to inventory mismatches.
- Developed a fraud detection system leveraging machine learning APIs, identifying suspicious transactions based on order patterns, flagged locations, and abnormal purchasing behaviors.
- Optimized ReactJS front-end performance using lazy loading, component-based architecture, and Redux state management, ensuring a seamless shopping experience.

- Integrated RabbitMQ for asynchronous processing of background tasks such as email notifications, invoice generation, and order confirmation, improving system efficiency.
- Created automated test suites using JUnit, Mockito, Cypress, and Jest, ensuring high test coverage and reducing regression issues before production releases.
- Implemented monitoring and logging using AWS CloudWatch and ELK Stack (Elasticsearch, Logstash, Kibana), allowing real-time tracking of system errors and application health metrics.
- Developed batch processing jobs using Spring Batch to handle bulk order processing, discount calculations, and scheduled email notifications, improving system efficiency.
- Led Agile sprint planning, backlog grooming, and daily stand-ups, working collaboratively with product managers, QA engineers, and DevOps teams to deliver high-quality e-commerce solutions.
- Provided technical mentorship to junior developers, guiding them in best practices related to ReactJS development, Java microservices, database optimization, and DevOps automation.

Environment: Java, Spring Boot, Spring Cloud, Spring Security, Hibernate, RESTful APIs, ReactJS, Redux, Material UI, MongoDB, Redis (caching), AWS (ECS, S3, CloudWatch), Docker, Kubernetes, Terraform, GitHub Actions, Apache Kafka, RabbitMQ, OAuth2, JWT, API Gateway, Role-Based Access Control (RBAC), JUnit, Mockito, Cypress, Jest, ELK Stack (Elasticsearch, Logstash, Kibana).

Client: Lister Technologies, Chennai, India. **June 2017 - December 2018**
Role: Java Developer

Project Description - FinTech Payment Gateway

At **Lister Technologies**, I worked on the development of a **FinTech Payment Gateway**, a secure and scalable platform designed to **process online transactions, manage digital payments, and handle financial settlements** across multiple banking institutions and merchants. The goal was to **enhance transaction security, improve processing speed, and ensure compliance with financial regulations** by leveraging **microservices architecture, cloud deployment, and real-time event-driven systems**.

Responsibilities:

- Designed and developed a secure online payment processing system using Spring Boot microservices architecture, ensuring modularity, scalability, and fault tolerance for high transaction volumes.
- Implemented RESTful APIs for handling user authentication, payment processing, transaction history, and refund mechanisms, integrating with multiple third-party banking services.
- Built an Angular-based UI for financial transactions, invoice generation, and real-time payment tracking, improving user engagement and efficiency.
- Integrated MongoDB as a NoSQL database to store and retrieve high-velocity financial transactions, ensuring quick and scalable data access.
- Implemented Apache Kafka as a message broker to process millions of real-time transactions asynchronously, reducing latency in fund transfers and fraud detection.
- Deployed and managed containerized services using Docker and Kubernetes on Google Cloud Platform (GCP), ensuring high availability, disaster recovery, and auto-scaling of services.
- Developed and optimized transaction reconciliation jobs using Spring Batch, automating settlement processing between merchants and banks.
- Integrated OAuth2 and JWT-based authentication mechanisms to secure APIs and user sessions, preventing unauthorized access to financial data.
- Implemented multi-factor authentication (MFA) for users and administrators, ensuring enhanced security compliance for payment processing.
- Designed and implemented role-based access control (RBAC) to restrict access based on user roles such as merchants, customers, and administrators.
- Developed and implemented caching strategies using Redis, reducing redundant database queries and improving API response times.
- Optimized MongoDB queries using indexing and sharding techniques, improving transaction retrieval speeds for large datasets.

- Integrated third-party financial services (Visa, MasterCard, UPI, PayPal) via secure APIs, ensuring smooth payment processing across different platforms.
- Designed and implemented automated fraud detection algorithms that analyze transaction patterns and anomalies, flagging suspicious activities for further investigation.
- Developed a notification system using RabbitMQ to send real-time alerts for successful transactions, failed payments, and account security warnings via SMS and email.
- Implemented comprehensive logging and monitoring using ELK Stack (Elasticsearch, Logstash, Kibana) to track API performance, system errors, and transaction failures.
- Automated CI/CD pipelines using Jenkins and Terraform, ensuring rapid and secure deployments of microservices without manual intervention.
- Conducted performance testing using JMeter, simulating high-load scenarios to ensure the payment gateway can handle peak transaction volumes efficiently.
- Collaborated with DevOps teams to configure and optimize cloud infrastructure on GCP, ensuring cost-effective cloud resource allocation.
- Worked in an Agile environment, participating in daily stand-ups, sprint planning, and retrospective meetings, ensuring timely project delivery and continuous integration.

Environment: Java, Spring Boot, Spring Security, Spring Data JPA, Hibernate, RESTful APIs, Angular, TypeScript, Material UI, MongoDB, Redis (caching), Google Cloud Platform (GCP), Docker, Kubernetes, Terraform, Jenkins, Apache Kafka, RabbitMQ, OAuth2, JWT, Role-Based Access Control (RBAC), Multi-Factor Authentication (MFA), JMeter, JUnit, Mockito, ELK Stack (Elasticsearch, Logstash, Kibana).

Education:

Master's in computer science, State University of New York Polytechnic Institute, Utica, New York
Bachelor's in computer science, JNTUH, Hyderabad, India