Sai Prakash Pathuru

Software Developer

sppathuru@gmail.com
+1 (720) 316-5281
Portfolio
LinkedIn

PROFESSIONAL SUMMARY

Experienced Software Developer with 7 years of expertise in designing and implementing scalable web applications and microservices leveraging Java/J2EE, Spring Boot, and modern JavaScript frameworks including React, Angular, and Next.js. Proficient in developing robust, user-centric solutions with efficient RESTful APIs, event-driven architectures using Apache Kafka, and cloud ecosystems such as AWS and Azure. Skilled in managing data persistence with relational databases (MySQL, PostgreSQL, Oracle) and NoSQL systems (MongoDB, DynamoDB), alongside advanced CI/CD pipelines utilizing Jenkins and Docker. Adept at securing applications with Spring Security, JWT, and OAuth, ensuring comprehensive quality throughout the software development lifecycle.

TECHNICAL SKILLS

Programming & Scripting Languages	Java 8/11/17, Python, C++, JavaScript, SQL, HTML5, CSS3, PL/SQL, Ruby, C, PowerShell, Bash
Enterprise Java Technologies	JAX-WS, JAX-RS, JSTL, JMS, JavaBeans, Servlets, JSP, JDBC, EJB, JNDI
Software Frameworks	Spring (Boot, MVC, Data JPA, Security, AOP, Cloud, Batch, IOC), JUnit, Hibernate, Struts, React.js, Vue.js, Next.js, Node.js, Angular, Cucumber, Mockito, JMockit, Selenium
Web Development Technologies	RESTful APIs, SOAP, XML, JSON
Database Management Systems	MySQL, MS SQL Server, Oracle, MongoDB, DynamoDB
SDLC	Agile (Scrum, JIRA), Waterfall
Cloud Computing Platforms & Services	AWS (EC2, S3, RDS, Lambda, API Gateway, IAM, ECS, EKS, EBS, VPC, ELB, ECR, SQS, SNS), Docker, Terraform, Azure, Kubernetes, Google Cloud Platform (GCP)
Message Queue & Event-Driven Services	Apache Kafka, RabbitMQ
Software Architecture & Design Patterns	Microservices, MVC, SOA, ORM
Application & Web Server Technologies	Apache Tomcat, WebSphere, JBoss, WebLogic
Integrated Development Environments	IntelliJ IDEA, Eclipse, VSCode
Version Control & Code Repositories	Git, GitHub
Build Automation & CI/CD Tools	Maven, Jenkins
Configuration & Source Code Management	Git, Jenkins, SVN
Testing & Monitoring	GraphQL, Prometheus, Grafana, ELK Stack, Postman, Log4J, Jasper Reports

PROFESSIONAL EXPERIENCE

Marathon Petroleum – Houston, TX **July 2024 – Present**
Software Developer

Project Description: Architected the Tesoro Gift Card platform, a sophisticated full-stack system integrating Java 17, Spring Boot, React, and AWS to orchestrate gift card lifecycle management (activation, redemption, balance inquiry) across Marathon's retail ecosystem. Engineered a high-performance solution with PostgreSQL for persistent storage, Apache Kafka for real-time transaction event streaming, and a PCI-compliant payment gateway, utilizing Spring Security

with JWT for secure API access and Redis caching to optimize balance lookup latency, ensuring scalability and reliability under transactional loads.

Frontend:

- Crafted a responsive single-page application using React and Redux, implementing reusable components for gift card sales and management with Typescript for type safety and seamless state management.
- Engineered interactive forms for activation and balance checks with JavaScript client-side validation, enhancing user feedback loops and input integrity via HTML5 and CSS3.
- Integrated RESTful APIs with Axios for real-time UI updates (e.g., balance post-redemption), incorporating error handling and notifications using Bootstrap for consistent styling.
- Optimized UI performance with lazy loading and Webpack code-splitting, reducing load times and enhancing responsiveness across devices.
- Ensured cross-browser compatibility and mobile responsiveness using HTML5, CSS3, and Bootstrap, delivering a uniform experience.
- Implemented role-based UI access with conditional rendering in React, displaying admin tools (e.g., bulk issuance) via context-driven logic.
- Wrote unit and integration tests with Jest and React Testing Library, validating core workflows (purchase, activation, balance check) for high reliability.

Backend:

- Designed Spring Boot microservices in Java 17 to manage gift card operations, enforcing atomic transactions with robust business logic and Spring Data JPA.
- Built a persistence layer with Hibernate and PostgreSQL, optimizing relational schemas and indexing for rapid balance retrievals.
- Integrated Apache Kafka for asynchronous transaction event publishing and RESTful APIs for POS/payment system interoperability, ensuring seamless downstream processing.
- Secured APIs with Spring Security and JWT, validating inputs to mitigate fraudulent requests and ensure system integrity.
- Developed Spring Batch jobs for nightly transaction reconciliation, generating automated financial reports with PostgreSQL aggregations.
- Implemented Redis caching for high-frequency balance lookups, reducing database strain and improving API response times.
- Documented endpoints with Swagger/OpenAPI, supporting API versioning and team integration with RESTful standards.

Cloud:

- Containerized microservices with Docker and deployed on AWS EKS using Kubernetes, enabling auto-scaling and high availability with rolling updates.
- Provisioned AWS infrastructure (e.g., RDS, S3, IAM) via Terraform, securing credentials in AWS Secrets Manager within a custom VPC.
- Configured AWS RDS (PostgreSQL) with multi-AZ failover and CloudWatch monitoring, tuning for low-latency performance under load.
- Leveraged AWS SQS for bulk task queuing and AWS SNS for transaction alerts, decoupling workloads for enhanced responsiveness.

Environment: Java 17, Spring Boot, Spring Data JPA, Spring Security, Spring Batch, Apache Kafka, React, Redux, Typescript, HTML5/CSS3, Bootstrap, PostgreSQL, Redis, RESTful APIs, Swagger/OpenAPI, Docker, Kubernetes (EKS), AWS (RDS, S3, SQS, SNS, CloudWatch, Secrets Manager, VPC), Jenkins, Terraform, JUnit, Mockito, Jest

Entergy – Frisco, TX **Jan 2023 – May 2024**
Software Developer

Project Description: Engineered the "Events and Presentations" platform, a high-throughput, event-driven web application designed to facilitate real-time investor interactions. The solution delivered dynamic content, including

comprehensive event schedules, detailed presentation materials, and live webcasts. Leveraged microservices architecture, developed robust RESTful APIs, deployed scalable cloud infrastructure, and adhered to Agile methodologies. Ensured secure, high-performance operations by implementing continuous integration and deployment (CI/CD) pipelines, comprehensive security strategies, effective caching mechanisms, and optimized load balancing to handle peak traffic during financial events.

Frontend:

- Developed an interactive, component-based React UI utilizing hooks and the context API for efficient state management.
- Ensured robust mobile responsiveness, cross-browser compatibility, and compliance with WCAG 2.1 accessibility standards through HTML5 and CSS3.
- Integrated asynchronous data retrieval from RESTful APIs using JavaScript fetch API and Axios, significantly enhancing user experience.
- Implemented client-side navigation with React Router, enabling smooth transitions and deep linking across various application sections.
- Utilized Redux for centralized state management, ensuring consistent and predictable state transitions across interactive elements.
- Conducted rigorous unit and integration testing with Jest and React Testing Library to validate functionality and component reliability.
- Improved frontend performance through optimization techniques including Webpack code splitting and lazy loading for faster load times.

Backend:

- Built comprehensive RESTful APIs using Java 17, Spring Boot, Spring MVC, and Spring Data JPA for efficient interaction with PostgreSQL databases.
- Architected a scalable service-oriented architecture (SOA) with Spring Cloud, leveraging Eureka for service discovery and Ribbon for load balancing.
- Implemented strong security measures with Spring Security, OAuth 2.0, and OpenID Connect (OIDC), providing secure role-based access control.
- Enhanced application performance through caching frequently accessed data with Ehcache.
- Scheduled and automated routine tasks like data archiving and updates through Spring Batch, ensuring consistent data management.
- Validated backend functionality through extensive unit testing with JUnit and Mockito, ensuring accurate and reliable service behavior.
- Integrated Amazon S3 for scalable, efficient file storage and retrieval, utilizing AWS SDK for Java.

Cloud:

- Deployed the application onto AWS Elastic Beanstalk, employing auto-scaling groups and load balancers to achieve high availability and reliability.
- Managed presentation file storage efficiently on Amazon S3, applying version control and lifecycle policies for optimized storage management.
- Configured detailed monitoring with Amazon CloudWatch, integrating automated incident response mechanisms through AWS Lambda.
- Ensured secure management of AWS resource access through detailed IAM roles and policies.
- Automated infrastructure provisioning and deployments using AWS CloudFormation, enabling streamlined and error-free environment setups.

Environment: Java 17, Spring Boot, Spring MVC, Spring Security, Spring Cloud (Eureka, Ribbon), Spring Batch, React, Redux, HTML5, CSS3, PostgreSQL, Ehcache, AWS (Elastic Beanstalk, S3, CloudWatch, Lambda, IAM, CloudFormation), OAuth 2.0, OpenID Connect, Jest, React Testing Library, JUnit, Mockito, AWS SDK for Java.

KeyBank – Cleveland, OH **Sep 2019 – Aug 2022**
Software Developer

Project description: Developed the Personalized Wealth Planning application, a secure web platform leveraging Java 11, Spring Boot, Angular, and Azure to enable high-net-worth clients and advisors to collaboratively manage financial plans. Integrated SOAP and RESTful APIs for real-time banking/market data, implemented Hibernate with Oracle for persistence, and utilized Spring Batch for scheduled data updates, delivering complex financial analytics (e.g., Monte Carlo simulations) with JasperReports for client-facing outputs.

Frontend:

- Built dynamic Angular components with Typescript for goal planning and portfolio dashboards, using Reactive Forms for multi-step input validation.
- Integrated Highcharts with custom JavaScript for interactive financial visualizations (e.g., cash flow projections, asset allocation).
- Ensured responsive design with Bootstrap and CSS3, supporting cross-device access for clients and advisors.
- Handled OAuth 2.0 SSO integration in Angular, securing token management for seamless authentication.
- Wrote Jasmine/Karma unit tests and Protractor end-to-end tests, ensuring robust user flows (plan creation, reporting).
- Optimized performance with AOT compilation and OnPush change detection, enhancing responsiveness for data-intensive views.
- Implemented role-based UI controls with Angular route guards, restricting advisor tools via permission directives.

Backend:

- Developed Spring Boot services in Java 11 for financial calculations (e.g., net worth, forecasting) with RESTful endpoints.
- Integrated JAX-WS SOAP services and REST APIs for banking/market data, optimizing Java Monte Carlo simulations for accuracy.
- Scheduled Spring Batch jobs for nightly data imports (e.g., market indices), generating audit logs with PL/SQL.
- Persisted data with Hibernate and Oracle 12c, writing PL/SQL procedures for aggregate computations.
- Generated JasperReports PDFs for financial summaries, assembling data for offline client review.
- Secured APIs with Spring Security, encrypting data with TLS and implementing caching via Ehcache for performance.
- Containerized services with Docker, ensuring consistency across development and production environments.

Cloud:

- Deployed on Azure App Service, automating CI/CD with Azure DevOps for staging/production slot deployments.
- Configured Azure SQL Database with Key Vault for secure credential management, migrating Oracle schemas seamlessly.
- Monitored performance with Azure Application Insights, setting custom dashboards/alerts for real-time issue tracking.
- Ensured HA/DR with multi-region Azure Traffic Manager and geo-replicated databases, securing traffic with VNETs.

Environment: Java 11, Spring Boot, Spring MVC, Spring Security, Spring Batch, Hibernate, Angular 8+, Typescript, RxJS, HTML5/CSS3, Bootstrap, Highcharts, Oracle 12c, MS SQL Server, SOAP (JAX-WS), RESTful APIs, Ehcache, JasperReports, Docker, Azure (App Service, SQL Database, Key Vault, Application Insights, DevOps Pipelines), JUnit, Mockito, Jasmine, Karma, Protractor

Zomato – Bengaluru, India **Oct 2017 – Aug 2019**
Software Developer

Project Description: Enhanced the Online Ordering & Payments system, a high-throughput platform built with Java 8, Spring Boot, Angular, and AWS, managing end-to-end food ordering and payment workflows. Integrated RabbitMQ for decoupled order events, MySQL for transactional data, and Redis for caching, while leveraging WebSockets and RESTful APIs to ensure real-time status updates and seamless payment gateway integrations (e.g., Razorpay, Stripe) under peak loads.

Frontend:

- Developed an Angular ordering interface with Typescript and HTML5/CSS3, supporting menu browsing and checkout flows.
- Implemented a dynamic checkout with JavaScript, integrating Google Maps API for address validation and real-time updates.
- Integrated payment SDKs (e.g., Razorpay) with JavaScript, handling 3-D Secure verification in the UI.
- Utilized WebSockets via Socket.io for real-time order status updates, enhancing user engagement without refreshes.
- Optimized performance with service workers and local storage caching, lazy-loading assets for faster page loads.
- Ensured responsive design across browsers with CSS3, testing compatibility with Selenium automation.
- Instrumented analytics with JavaScript tools (e.g., Google Analytics), identifying UX improvements for higher conversions.

Backend:

- Built Spring Boot microservices in Java 8 for order lifecycle management, exposing RESTful APIs for seamless processing.
- Integrated payment gateways (e.g., Razorpay) with RESTful handlers and webhooks, managing edge cases like timeouts/refunds.
- Configured RabbitMQ for event-driven order processing, decoupling services for scalability and fault tolerance.
- Designed and optimized MySQL schemas/queries for orders and payments, indexing for high-volume performance.
- Cached order data with Redis, reducing MySQL load and accelerating status retrievals.
- Developed a delivery integration module with RESTful APIs, syncing with external partners and triggering notifications.
- Implemented retry mechanisms and Log4J logging, publishing analytics events to Kafka for business insights.

Cloud:

- Deployed on AWS EC2 with ELB and Auto Scaling, later transitioning to Docker and AWS EKS for orchestration.
- Managed AWS RDS (MySQL) with read replicas and ElastiCache (Redis) for distributed caching, optimizing latency.
- Configured ELK Stack on AWS for centralized logging and monitoring, building Kibana dashboards for diagnostics.
- Automated CI/CD with Jenkins, performing JMeter load tests to tune performance for peak traffic (e.g., 5× normal).

Environment: Java 8, Spring Boot, Spring MVC, Angular 5+, JavaScript/Typescript, HTML5/CSS3, MySQL, Redis, RabbitMQ, Apache Kafka, RESTful APIs, Docker, Kubernetes (EKS), AWS (EC2, ELB, Auto Scaling, RDS, ElastiCache), Jenkins, ELK Stack, Log4J, JUnit, TestNG, Selenium

EDUCATION

Master of Science in Computer Science – University of Colorado Colorado Springs, USA
Bachelor of Technology in Computer Science & Engineering – Jawaharlal Nehru Technological University, IN

CERTIFICATIONS

- **Certified Developer Associate –** AWS
- **AI for ALL: From Basic to GenAI Practice –** NVIDIA
- **Getting Started With Deep Learning –** NVIDIA
- **Building RAG Agents with LLMs –** NVIDIA