Prarup Manandhar
LinkedIn | prarupmanandhar98@gmail.com | **718 412 8521**

PROFESSIONAL SUMMARY

Java Full Stack Developer with 5+ years of experience delivering enterprise web and application solutions using **Java** and modern frameworks. Strong expertise in backend development, REST API design, and distributed systems with **Spring Boot**. Proven ability to build responsive and data-driven user interfaces using **Angular** across multiple versions. Hands-on experience working in **Agile** environments, contributing to sprint planning, feature development, and code reviews. Demonstrates leadership through ownership of features, collaboration with cross-functional teams, and mentoring support. Known for clear communication with stakeholders, QA teams, and developers to translate business requirements into functional solutions. Brings a balanced full-stack approach with experience in cloud-based deployments and scalable platforms using **AWS**.

TECHNICAL TOOLS

Backend: Java (8/11/17/21), Spring Boot (Spring MVC, Spring Security, Spring Profiles, Spring Actuator), RESTful APIs, Hibernate, JPA, JDBC, Microservices Architecture (Circuit Breaker, Fallback Logic), Resilience4j, Servlets, DTO Mapping, AOP, SOAP/XML, WSDL, XSD
Frontend: Angular (8/10/12/16/19), TypeScript, JavaScript, RxJS, HTML5, CSS3, SCSS, Material UI, Responsive Design, Angular Interceptors, Lazy Loading
Databases: PostgreSQL, Oracle, MySQL, MongoDB, DynamoDB, Redis
Messaging & Event Streaming: Apache Kafka (Producers/Consumers, Schema Validation, Topic Monitoring), Zookeeper, Consumer Groups, Retry Logic, Schema Registry, Event-Driven Workflows, Audit Trails
Security & Authentication: Spring Security (Custom Filters, RBAC), OAuth2, JWT, API Gateway, Route Guards, LDAP, Parameter Store
Testing & Quality Assurance: JUnit, Mockito, TestNG, Cucumber, Postman, Swagger, JMeter, K6, Contract Testing, SonarQube
Deployment Strategies: Docker, Jenkins (CI/CD, Automated Deployments, Pipelines), Blue-Green, Canary, WAR Packaging
Cloud & DevOps: AWS (ECS, EC2, S3, RDS, Lambda, API Gateway, CloudWatch, Parameter Store, Rekognition, Comprehend), Terraform, Docker, Kubernetes, CloudFormation, GitHub Actions, Bitbucket Pipelines
Monitoring & Observability: OpenTelemetry, Splunk, Distributed Tracing, CloudWatch, Dashboards, New Relic, Logging & Metrics
Automation & Scheduling: Linux Automation, Cron Jobs, Batch Jobs, Background Pipelines, XML Audit Reports
Tools & Collaboration: Git, GitHub, Bitbucket, IntelliJ IDEA, Maven, NPM, JIRA, Confluence, Agile/Scrum, Code Reviews

EDUCATION

Clark University
Master of Science in Information Technology

PROFESSIONAL EXPERIENCE

USAA

June 2024 – Current | St Louis, Mo, USA

Project: Policy Operations Modernization Portal

The Policy Operations Modernization Portal was created to improve how internal teams interact with policy data and batch processing results. Worked on developing backend services with **Spring Boot** and added frontend features in **Angular** such as policy search, workflow views, and status indicators. The project addressed business needs related to reducing manual effort and improving visibility across insurance operations.

Responsibilities:

- Brought legacy modules up to date with **Java 21** so that applications ran more efficiently.
- Applied **MVC architecture** in **Spring Boot and Angular** applications to ensure clean separation between data models, controllers, and UI components.
- Developed enterprise web and batch applications using **Java, Spring Boot, Hibernate, JPA,** and **REST APIs**, deployed on **Tomcat** and **WildFly** servers.
- Designed and implemented event-driven java microservices using **Apache Kafka** for asynchronous communication across policy processing systems.
- Used **Docker** to containerize services and deployed them on **Amazon ECS** via a Jenkins powered **CI/CD** pipeline.
- Implemented **JWT** and **OAuth2** authentication with **Spring Security** to protect APIs and maintain secure communication.
- Migrated and optimized database workflows by transitioning **batch jobs** and applications from **Oracle** to **PostgreSQL**, refining **SQL** queries for compatibility and performance improvements.
- Set up **ECS** load balancing with an Application Load Balancer using **Terraform** and AWS Console to improve performance and ensure high availability.
- Improved **frontend** user experience with **Angular 16** through advanced form validations, enhanced routing, and route guards for secure navigation.
- Configured and maintained **PostgreSQL RDS instances on AWS**, including backups, snapshots, and performance monitoring.
- Gained hands-on exposure to **Angular 19**, exploring its new standalone components and enhanced signals for state management.
- Developed **Kafka producers and consumers using Spring Kafka**, integrated into Spring Boot microservices.
- Designed **Kafka** topics with appropriate **partitioning and replication** to support high-throughput and fault-tolerant processing.
- Configured kafka **consumer groups** to enable parallel message consumption and horizontal scalability.
- Integrated **Angular build pipelines** into **Jenkins CI/CD**, automating testing and deployment of frontend assets to AWS S3 and CloudFront for scalable delivery.
- **Configured Swagger UI in Spring Boot applications** to auto-generate API docs and support seamless collaboration between microservice teams.
- Configured **AWS API** Gateway to securely manage and expose **RESTful APIs** built in **Java**.

- Deployed containerized services on **ECS** backed by **EC2** capacity with **S3** and **RDS** integrations so that applications ran reliably in production environments.
- Applied **Dependency Injection** and **Inversion of Control** in Spring using annotations to minimize coupling across components.
- Conducted migration of **Java** applications, **batch jobs**, and scripts from **Oracle** to **PostgreSQL**, ensuring compatibility, data integrity, and a smooth transition.
- Wrote **unit** and **integration tests** using **JUnit** and **Mockito** to validate code quality and maintain system reliability.
- Monitored **Kafka** consumer lag and broker health to maintain system stability.
- Worked with **Zookeeper-managed Kafka clusters** to ensure distributed coordination and availability.
- Configured IAM roles and security groups in **AWS ECS** using Terraform to secure communication between AWS services.
- Implemented role-based access and security in **PostgreSQL**.
- Integrated **AWS Lambda** with **API Gateway** and **S3** to handle serverless requests and data processing efficiently.
- Used **Git** with **Bitbucket** for version control, collaborated through pull requests, and automated deployment pipelines via Jenkins.
- Used Core **Java** Collections, **Generics**, and **Multithreading** to build reusable and concurrent components for scalable applications.
- Implemented Core **Java** performance tuning with **thread pooling**, **synchronization**, memory management, and **JVM** monitoring to optimize throughput and reduce latency.
- Integrated **SonarQube** into the CI pipeline to detect bugs, code smells, and vulnerabilities, to achieve continuous code quality
- Used **AWS AI** services like **Comprehend** and **Rekognition** to add text and image analysis features to applications.
- Contributed to Java code reviews, addressed defects, delivered **bug** fixes, and presented working solutions during **UAT** to stakeholders.
- Used **JIRA** to manage Agile sprints, create and update user stories, and collaborate with cross-functional teams for issue tracking and release planning.
- Applied **Agile** principles to manage iterative development, prioritize backlog items, and deliver incremental product releases on schedule.

Technological Environment:
Java 21, Core Java, Spring Boot (Spring MVC, Spring Security, Spring Profiles, Spring Actuator), Hibernate, JPA, RESTful APIs, Angular16/19, TypeScript, JavaScript, Kafka, schema validation, Zookeeper, Microservices, PostgreSQL, Oracle, AWS (ECS, EC2, S3, RDS, Lambda, API Gateway, Parameter Store), Docker, Jenkins CI/CD, Terraform, SonarQube, JUnit, Mockito, OpenTelemetry, Splunk, Resilience4j, Agile/Scrum, OAuth2, JWT, RBAC, AI/ML, JIRA, Bitbucket, Distributed Tracing

Nike
March 2022 – May 2024 | Hillsboro, OR, USA

Project: Product Lifecycle Tracking Platform

Built as a centralized Product Lifecycle Tracking Platform, this project helped business users follow product progress across planning and release phases. Contributed by developing interactive screens in **Angular** and supporting backend data services using **Java microservices** for dashboard metrics and product timelines. The platform supported faster decision making and better coordination across product teams.

Responsibilities:

- Designed and developed enterprise-level web applications using **Java 11**, **Spring Boot**, and **microservices architecture**, in order to modular, scalable solutions.
- Implemented RESTful APIs with DTO mapping, layered configurations, and exception handling so that backend logic stayed consistent and client integration worked smoothly.
- Built and integrated custom **Angular 12 components** with TypeScript, Material UI, and lazy-loaded modules, enabling faster load times and a more responsive user experience.
- Developed **HTTP interceptors** and reusable Angular services for centralized API handling and authentication.
- Created and tuned relational schemas and queries in **MySQL** for transaction-heavy modules in order to maintain data consistency and improve execution speed.
- Created and tuned **MongoDB collections** for high-performance document queries, applying indexing strategies to optimize lookups.
- Designed **DynamoDB tables** with partition and sort key strategies for event-driven workflows, to keep predictable performance and scalability for large datasets.
- Integrated **Apache Kafka** with producer/consumer groups to handle distributed event streaming, retries, offsets, and audit trails.
- Integrated optimized MySQL queries within **Spring Data JPA repositories**, improving API response times and maintaining transaction safety across services.
- Ensured event ordering and data consistency by designing **Kafka** topics with appropriate partition keys aligned to business entities.
- Migrated a legacy monolithic application to **Spring Boot microservices**, significantly improving scalability, reducing downtime during deployments, and enhancing feature delivery speed.
- Implemented event driven and background job pipelines using **messaging patterns** to support **asynchronous** workflows and audit compliance.
- Automated builds and deployments using **Docker** containers and **Jenkins CI/CD pipelines**, to maintain environment-based configurations.
- Customized **Swagger** configuration to include **JWT** authentication headers, improving API testing efficiency and developer experience during QA validation.
- Validated **Kafka** event flows using integration tests and mocked producers/consumers to ensure message integrity across microservices.
- Deployed microservices on **Kubernetes** using **Terraform** for automated infrastructure provisioning, integrating with AWS **API Gateway** and **S3** to achieve scalable and resilient deployments.
- Developed AWS **Lambda** functions using **Python** and **Terraform** to handle data processing, log cleanup, and routine maintenance automatically.
- Managed **AWS RDS** instances, configuring backups, parameter groups, and read replicas for high availability and disaster recovery.
- Integrated **Kubernetes** with **Jenkins** pipelines to automate microservice deployments, configured rolling updates, and implemented rollback policies.

- Configured centralized monitoring and alerting with **Splunk** and **AWS CloudWatch** to improve root cause analysis, enabling proactive error resolution.
- Conducted **performance/load testing with JMeter** to simulate production workloads and validate responsiveness under peak demand.
- Collaborated in **Agile sprints**, including story refinement, sprint reviews, QA validation, and continuous integration planning.
- Maintained version control and branching strategies in **GitHub**, including CI triggers via **GitHub Actions** for automated builds and tests.

Technological Environment:
Java 11, Spring Boot, Spring Data JPA, Spring Profiles, Spring Beans, Spring Actuator, RESTful APIs, DTO Mapping, Microservices Architecture, Angular 12, TypeScript, Angular Interceptors, Lazy Loading, MySQL, MongoDB, DynamoDB, Apache Kafka, Zookeeper, Docker, Jenkins CI/CD Pipelines, GitHub Actions, Kubernetes, Terraform, AWS Lambda, AWS API Gateway, AWS S3, AWS RDS, AWS CloudWatch, Splunk, Swagger, JWT Authentication, Python, Resilience4j, OpenTelemetry, JMeter, Agile Sprints, QA Validation, GitHub

Farm Bureau Mutual Insurance Company
April 2020 – Feb 2022 | Idaho, USA

Project: Claims Data Management Application
The Claims Data Management Application supported internal claim review and historical data analysis. Developed backend logic using **Java** and implemented frontend pages with **Angular** to display claim summaries, transaction history, and reporting views. The application met business needs focused on improving claim handling efficiency and data accuracy for operational teams.

Responsibilities:

- Built **Java 8 applications** with **Spring Boot, JSP,** and **Servlets**, integrating with **Angular 8** for improved data processing and platform efficiency.
- Developed **Kafka consumer** groups with retry logic and offset management to handle real-time ingestion so that analytics modules processed high volume daily events without data loss.
- Maintained reliability by writing **unit tests with JUnit** and **integration tests** with **Mockito**, validating service functionality.
- Secured **Java** applications by implementing **Spring Security** with **OAuth2** and **JWT**, to maintain strong authentication, authorization, and user data protection
- Applied **Aspect-Oriented Programming** to modularize cross-cutting concerns such as logging and transaction management, improving maintainability and performance.
- Designed and developed responsive **Angular 8 applications with TypeScript**, enhancing the user experience through interactive and dynamic components.
- Developed responsive UI components using **HTML5** and **JavaScript** to enhance interactivity and performance.
- Applied cross-browser compatibility methods in **CSS** in order to maintain consistent rendering across Chrome, Firefox, and IE11.

- Managed data storage and retrieval with **Oracle** and **PostgreSQL**, optimizing **SQL queries,** stored procedures**,** and **indexing** strategies for performance improvements.
- Managed **MongoDB** collections and indexes to support high-volume, schema-flexible data processing in microservice modules.
- Integrated with **legacy SOAP/XML-based services**, generating and parsing **WSDL** and **XSD** schemas for enterprise interoperability.
- Managed cloud deployments on **AWS**, combining EC2 and S3 services with Linux-based automation and scheduling.
- Used **AWS CloudFormation** templates to provision and manage **ECS** clusters and related infrastructure as code (IaC).
- Monitored **Java** application system health using **AWS CloudWatch** and custom **JVM** monitoring scripts for proactive detection and resolution of production issues.
- Scheduled **cron jobs on Linux** for Oracle data cleanup and generating XML-based audit reports.
- Collaborated in **Agile development** with cross-functional teams, implementing CI/CD pipelines through Jenkins, Git, and GitHub to streamline workflows and accelerate delivery timelines.

Technological Environment:
Java 8, Spring Boot, JSP, Servlets, Struts, Spring Security, OAuth2, JWT, Apache Kafka, SOAP, XML, WSDL, XSD, WAR deployments, Log4J, LDAP, ESB, Angular 8, TypeScript, HTML5, CSS3, JavaScript, Oracle 11g, PostgreSQL, MongoDB, SQL, AWS EC2, AWS S3, AWS CloudFormation, AWS CloudWatch, Linux Shell Scripting, Cron Jobs, JUnit4, Mockito, Git, GitHub, Jenkins Pipelines, Maven, Agile/Scrum