SAMEER NADAF

+919353015312• sameernadaf5761@gmail.com • www.linkedin.com/in/ sameer-nadaf-logged

SUMMARY

Backend Developer with 3.5+ years of experience in Spring Boot, Microservices, Hibernate, and AWS. Skilled in REST APIs, JWT, Kafka, and CI/CD with Jenkins. Experienced in Agile (Scrum) using JIRA. Delivered scalable, secure, cloud-native systems with integrations like Stripe, SendGrid, and Twilio.

EXPERIENCE

Java Developer , Peerbits Technologies 3.1 years

PROJECTS

1. PG Management System
   - **Tech Stack:** Angular, Spring Boot, MySQL, JWT
   - Multi-role PG management system for owners and tenants.
   - Role-based login, daily meal tracking, and notification system.
2. Portfolio Website.
   - **Tech Stack:** Angular, Bootstrap, HTML5, CSS3, Vercel
   - Designed and developed a responsive personal portfolio showcasing skills, projects, and contact information.
   - Hosted on a custom domain with live deployment and version control.
     live: https://www.iamsameer.in/

TECHNICAL SKILLS

Programming & Frameworks

- Java 8–17, Spring Boot, Spring MVC, Spring Security, Spring Cloud, Hibernate (JPA), Feign Client, RESTful APIs, Microservices Architecture

Database & Caching

- MySql
- ER Diagrams
- Redis Cache

Cloud & DevOps

- AWS (EC2, S3, IAM), Docker, Kubernetes (EKS), Terraform, Jenkins (CI/CD), GitHub Actions, Git etc.

Messaging & Communication

- Apache Kafka, REST API, Feign Client for communication in microservices, Eureka Server, API Gateway etc.

Authentication & Security

- JWT, Spring Security, Role-Based Access Control (RBAC)

Tools & Platforms

- JIRA, Postman, Swagger, IntelliJ IDEA, STS, Maven, Gradle, Git, GitHub.

Monitoring & Logging

- SLF4J, Logback, Zipkin, Spring Boot Admin, ELK

Development Process

- Agile Methodology (Scrum) – Sprint Planning, Daily Stand-ups, Reviews, Retrospectives

Zenhotels.com

Duration: 3.5 Years
Architecture: Microservices | Domain-Driven Design | Event-Driven
Methodology: Agile (Scrum) | Daily Standups | Sprint Planning via JIRA

Tech Stack

Spring Boot, Spring Security, JWT, Hibernate (JPA), Kafka, Feign Client, Stripe SDK, Docker, Kubernetes (EKS), AWS (EC2, S3), API Gateway, Eureka Server, Zipkin, Admin Server, Terraform, Redis, MySQL, ELK Stack, SendGrid, Twilio, GitHub, Maven, SLF4J

Modules & Responsibilities

1. **User Service**

- Implemented secure login, registration, password management
- Role-based access control (Admin/Customer) using Spring Security + JWT.
- Persistent session handling with token validation.

2. **Hotel Service**

- CRUD operations for hotel listings, metadata (images, rules, amenities).
- Managed image uploads via AWS S3 and exposed presigned URLs. Used Hibernate/JPA for relational mapping and validations.

3. **Room Inventory Service**

- Handled availability, room types, and dynamic pricing logic.
- Implemented date-based inventory tracking with optimized queries. Integrated Redis for caching frequently accessed inventory data.

4. **Booking Service**

- Developed a booking engine with validation, availability check, and confirmation number generation.
- Kafka used to broadcast booking events (to Payment, Notification services). Synchronized inventory
- update and cancellation workflows.

5. **Payment Service**

- Integrated with Stripe SDK (sandbox & production) for secure payments.

6. **Search & Filter Service**

- Implemented full-text search with Elasticsearch by hotel name, location, amenities. Applied multi-criteria
- filters: star rating, price range, facilities, room type.

7. **Review & Rating Service**

- Enabled users to add/edit/delete hotel reviews and ratings. Aggregated scores used
- in hotel ranking algorithm.

8. **Notification Service**

- Real-time communication via Kafka, triggered SMS/email using Twilio/SendGrid.

9. **Infrastructure & Integration**

- Implemented API Gateway for routing, security, and rate limiting.
- Eureka Server for service discovery and registration of microservices.
- Zipkin integration for distributed tracing and performance monitoring.
- Spring Boot Admin Server for real-time monitoring of service health.
- Terraform used for provisioning AWS infrastructure (EC2, S3, IAM roles).
- Dockerized all services; deployed via Kubernetes (EKS) for production scaling.
- Created ER Diagrams and UML class diagrams for design reviews.
- Used Feign Client for inter-service REST communication with fallback logic.