Mahesh Sutar

Software Engineer

7249654670

sutarmaheshgambhir@gmail.com

EXPERTISE

Programing Languages

Java
Advance Java
Java 8
SQL

Backend Technologies & Frameworks

Spring Boot
Spring MVC
Spring Security
Spring Data JPA
Hibernate
JWT (JSON Web Tokens)
ModelMapper
BCryptPasswordEncoder
Microservices

Database Technologies

MySQL

Web & API Technologies

RESTful Web Services
Swagger/ (API Documentation)
Postman (API Testing)
JSON/XML

Deployment Tools

Maven
Git, GitHub
Linux
AWS EC2,S3
Jenkins(CI/CD)
Docker
Kafka
Jira
JUnit
Mockito

Tools & IDEs

IntelliJ IDEA
Eclipse
STS

EDUCATION

BACHELOR'S DEGREE

SHIVAJI UNIVERSITY, KOLHAPUR.

PROFILE

I am a dedicated Java Developer with 3.2 years of experience in developing secure and scalable web applications using Java, Spring Boot, Spring Security, and MySQL. I have strong expertise in building RESTful APIs, implementing JWT-based authentication, and working with JPA/Hibernate for database operations. In my recent Hotel Management System project, I developed complete backend modules including user, property, booking, and admin functionalities. I'm proficient in tools like Postman, Swagger, Maven, Lombok, and follow clean code practices using layered architecture. I'm looking for opportunities to contribute to impactful backend development projects.

PROFESSIONAL EXPERIENCE

JAVA DEVELOPER

Tricon Infotech Pvt. Ltd. *Mar 2022-Present*

Hotel Management System is a monolithic Java Spring Boot web app that centralizes hotel bookings, listings, cities, reviews, and user accounts. It ensures a seamless experience for users and admins with a layered, maintainable architecture.

Technologies Used

- **Java, Spring Boot, Spring Security, Spring Data JPA**
- **MySQL** (Relational Database)
- **JWT** (JSON Web Tokens) for authentication
- **Lombok** for boilerplate reduction

Tools & Utilities

- **Postman** for API testing
- **Swagger** for interactive API documentation
- **BCryptPasswordEncoder** for secure password encryption

Architecture & Design

- **Monolithic** architecture with a single deployable unit
- **Layered structure**: Controller → Service → Repository → Database
- **Stateless authentication** using a **custom JWT filter**
- Clean code practices with **exception handling** and **modularity**

Security Features

- **JWT-based authentication** with **role-based access control**
- **BCrypt** for password hashing
- **Spring Security** integration for endpoint protection

Core Functionalities

- **User registration and login**
- **Hotel/property management** (create, update, delete)
- **City management** by admins
- **Booking system** with check-in/check-out and booking status
- **Review system** with ratings and comments

Software Tools

Spring Security + JWT – For securing REST APIs with token-based authentication and role-based access control.
Spring Data JPA + Hibernate – To interact with the database using object-oriented Java code (ORM).
Postman – To test and validate RESTful APIs during development.
Swagger (Springdoc OpenAPI) – For auto-generating and documenting APIs with an interactive UI.
Docker – To containerize the Spring Boot application for easier deployment and environment consistency.

Project Briefing

I developed a secure and scalable Hotel Management System backend using Spring Boot. The system supports user registration, hotel search by city, booking, and reviews, with secure login powered by Spring Security and JWT. I used Spring Data JPA with Hibernate to manage relational data in MySQL, and followed a layered architecture for clean separation of concerns. REST APIs were tested using Postman and documented with Swagger for easy access. The application was containerized using Docker for consistent deployment across environments. I also wrote unit tests using JUnit and Mockito to ensure code reliability.

Roles and Responsibilities

Designed and developed the Hotel Management System backend using **Spring Boot** with a modular architecture, including Property, Booking, City, and Review modules to ensure clean separation of concerns and scalability.

Built secure **authentication** and **authorization** mechanisms using **Spring Security** and **JWT**, enabling role-based access control for Admins, Owners, and Customers across all endpoints.

Hash passwords using **BCryptPasswordEncoder** to store user credentials securely. Prevented unauthorized access by ensuring passwords were never stored in plain text.

Developed **Restful APIs** for user registration, login, property listing, booking, and reviews, and documented them using Swagger for seamless integration with frontend and QA teams.

Implemented **MySQL** database schema and used **Spring Data JPA** for **CRUD** operations, writing custom queries where needed to optimize property search, city-based filters, and user-specific bookings.

Built Admin Panel functionalities including hotel property approval, user management, and system monitoring, ensuring administrative control over platform content and performance.

Integrated DTOs with **ModelMapper** to decouple internal entity structures from API response models, improving security and flexibility of API responses.

Managed bookings module **with date-based validation**, cancellation support, and conflict checking logic to prevent double bookings and ensure data accuracy.

Used **Kafka** to enable asynchronous communication between Booking and Notification microservices, improving scalability and decoupling service logic effectively.

Followed layered architecture (**Controller-Service-Repository**) to maintain clean code separation, making the system easier to **debug**, test, and extend in the future.

Managed and deployed -based applications on **Linux** servers using **Docker** and **shell scripting**.Performed routine Linux server administration tasks such as user management, process monitoring, and disk space management. Managed application lifecycle (**start/stop/status**) using shell scripts and Linux service management tools (systemctl, nohup).Monitored microservices logs and application health using Linux tools like **journalctl**, **tail**, **grep**, and **top**.

Enabled **CloudWatch monitoring** on **EC2** to track performance metrics and maintain high availability of backend services.

Deployed the Spring Boot application on **AWS EC2**, setting up security groups, SSH access, and automated startup scripts.

Integrated **AWS S3** to store hotel images and user uploads securely, using pre-signed URLs for controlled access and IAM roles for permission management.

Wrote unit and integration tests using **JUnit** and **Mockito** for service classes to ensure functionality and stability. Achieved higher code quality and ensured that critical features were thoroughly tested.

Maintained version control using **GitHub** and followed branching strategies for feature development, bug fixing, and release management, ensuring clean and manageable codebase.