INNOVASTAY HMS (HOTEL MANAGEMENT SYSTEM) 🏨

**Technology Stacks :**- Spring Boot | Spring Data JPA | Spring Web | Hibernate | XAMPP -MySQL | Swagger/OpenAPI | Postman | JUnit | Mockito  
**Type:** Backend Project

**NAME :-** SHOUVIK BANIK

**EMPID :-** 30108683

CAPGEMINI TECHNOLOGY SERVICES (IND)

JAVA Foundation PT Batch SEP 2, CHE

# 1. ABSTRACT

The InnovaStay Hotel Management System (HMS) is a backend-driven enterprise application designed to automate and streamline hotel operations. Built using the Spring Boot framework, it integrates multiple core hotel management functionalities—such as room booking, amenities tracking, payments, and customer reviews—within a scalable microservice-ready architecture. The system provides robust RESTful APIs tested with Postman, documented using Swagger/OpenAPI, and validated through JUnit and Mockito test cases. This backend-first approach ensures reliability, modularity, and data consistency across all modules, enabling future frontend or mobile integration.

# 2. PURPOSE OF THE PROJECT

In the hospitality industry, data consistency and operational efficiency are critical. Many existing systems are monolithic or outdated, leading to poor scalability and high maintenance costs. The motivation behind developing InnovaStay HMS was to:

* Learn and implement Spring Boot enterprise-level backend development.
* Create a modular, scalable, and testable system following modern REST API design.
* Understand entity relationships and data persistence through JPA/Hibernate.
* Apply software engineering principles and real-world testing methodologies.

This project bridges theoretical knowledge and real-world hotel management challenges, providing hands-on exposure to full backend lifecycle development.

# 3. SCOPE OF THE PROJECT

The InnovaStay HMS automates and manages hotel data operations across seven major interconnected modules :-

|  |  |  |
| --- | --- | --- |
| **Modules** | **Description** | **Relationships** |
| Hotel Module | Stores hotel ID, name, description, and location details. | Linked to Amenity Module. |
| Amenity Module | Tracks hotel amenities such as Wi-Fi, parking, or pool access. | Linked to Hotel Module. |
| Room Module | Maintains details of individual rooms, including availability, price, and type. | Many-to-One with Room Type; One-to-Many with Reservation. |
| Room Type Module | Defines room categories with pricing and occupancy limits. | One-to-Many with Room. |
| Review Module | Captures user reviews and ratings for each reservation. | One-to-One with Reservation. |
| Reservation Module | Manages check-in/out, guest data, and booking details. | Many-to-One with Room; One-to-Many with Payment. |
| Payment Module | Handles payment transactions, status, and linkage to reservations. | Many-to-One with Reservation. |

# 4. DEVELOPMENT OF THE PROJECT

The development process followed a Sprint-based agile methodology, emphasizing modular design and continuous testing. Key Steps are as follows :-

* **Project Setup :-** Initialized Spring Boot project with Maven dependencies for JPA, Web, JUnit, Mockito and Validation through Spring Initializer.
* **Database Integration :-** Configured XAMPP MySQL for persistent data storage and schema creation.
* **Entity Design :-** Modeled entities using Hibernate annotations for one-to-one, one-to-many, and many-to-one relationships.
* **Repository Layer :-** Used Spring Data JPA repositories for CRUD operations and custom query methods.
* **Service Implementation Layer :-** Implemented business logic and integrated JUnit + Mockito for service-level testing.
* **Controller Layer :-** Designed RESTful APIs for all CRUD and functional operations.
* **Validation &** **Exception Handling :-** Validated Using @Valid, @Size & @NotBlank . Developed Global Exception Handling using Controller Advice and custom exceptions.
* **API Documentation :-** Integrated Swagger/OpenAPI to auto-generate REST documentation.
* **Testing :-** Validated endpoints with Postman, covering all positive and negative test cases.

# 5. KEY LEARNINGS AND TAKEAWAYS

This project was instrumental in developing the following technical competencies:-

* **System Architecture:**- Gained in-depth experience in designing and implementing layered backend systems using **Spring Boot**.
* **Data Persistence :-** Mastered advanced **JPA/Hibernate mappings** and techniques for **query optimization** to enhance database performance.
* **Testing & Quality Assurance :-** Developed comprehensive unit and integration tests using **JUnit and Mockito** to ensure code reliability and maintainability.
* **API Development :-** Applied **REST API best practices** for building clean, consistent, and intuitive web services.
* **Error Management :-** Implemented robust **exception handling** strategies with reusable service components for improved application stability.
* **Enterprise Application Design :-** Strengthened the ability to architect **scalable and modular enterprise applications** ready for production environments.

# 6. DIAGRAMS :-

**FIG:- 1 :- USE CASE IMAGE**

# 

**FIG 2 :- CLASS DIAGRAM**

# 7. CONCLUSION

The InnovaStay HMS project is a robust backend solution built with modern Java frameworks to manage hotel operations efficiently. Through this project, I gained significant exposure to real-world backend development cycles, industry-standard design patterns, and testing methodologies. In the future, the project can be extended to include frontend integration, authentication via Spring Security & JWT, and cloud deployment with Docker. This project not only enhanced my technical skillset but also reinforced my understanding of software design, testing, and documentation in professional enterprise environments.

Ultimately, InnovaStay HMS represents a successful synthesis of modern technology and disciplined engineering practices. It stands as a testament to my capability to translate complex business requirements into a well-architected, test-driven, and scalable solution. The experience has solidified my readiness to contribute effectively to challenging backend development projects and deliver software that is not only functional but also robust, maintainable, and aligned with industry best practices.