

# Software Requirements Specification (SRS)

## 1. Introduction

### 1.1 Purpose

The purpose of this project is to develop a simple hotel booking system for a small hotel with 15 rooms. The system will allow customers to book rooms online and enable the hotel owner to manage bookings efficiently.

### 1.2 Scope

- Customers can browse available rooms, book a stay, and receive confirmation.
- The hotel admin can manage room availability and view bookings.
- A simple authentication system will be implemented for secure access.
- The system will be web-based and responsive.

## 2. Requirements Prioritization

### ### Essential Features

- User Authentication (Signup/Login)
- Room Booking System (Check-in, Check-out)
- Booking Confirmation via Email
- Admin Panel for Managing Rooms & Bookings
- Room Details Page
- Room Availability Management

### ### Important Features

- Basic Payment System (Cash on Arrival)
- Search & Filter by Room Type
- Simple Admin Dashboard UI

### ### Optional Features

- Customer Reviews & Ratings
- Google Maps Integration
- SMS Notifications

### ### Future Enhancements

- Role-Based Access (Multiple Admins)

- Full Payment Integration
- Complex Search Filters

### 3. Functional Requirements

#### 3.1 User Module:

- Users must be able to register, log in, and log out.
- Users should be able to browse available rooms.
- Users can book a room by selecting check-in and check-out dates.
- Users receive a booking confirmation via email.

#### 3.2 Admin Module:

- The admin can add, edit, or remove rooms.
- The admin can view and manage bookings.
- The admin can update room availability.

### 4. Non-Functional Requirements

- Performance: The system should load pages within 2 seconds.
- Security: Basic security with hashed passwords (bcrypt) and JWT authentication.
- Scalability: Designed for a single hotel but should allow room for future expansion.
- Usability: The UI should be simple and mobile-friendly.
- Deployment:
  - Frontend: Hosted on Vercel/Netlify.
  - Backend: Hosted on AWS/DigitalOcean/Render.

### 5. Technology Stack

- Frontend: HTML, CSS, JavaScript, React.js (Optional)
- Backend: Node.js, Express.js
- Database: MongoDB
- Authentication: JWT (JSON Web Token)
- Deployment: Netlify (Frontend), Render/AWS (Backend)

### 6. Conclusion

This SRS document outlines a simple yet functional hotel booking system. By following this plan, we can ensure a structured development process while keeping the project manageable within 20 days.