# Project Documentation: Hospital Management System

## 1. Project Overview

### Objective:

The objective of this project is to create a web-based application that simplifies hospital operations and enhances patient experience. The platform enables patients to explore hospital facilities, make payments, and manage their information. It also provides administrators with tools to oversee the hospital's operations effectively.

### Key Features:

- Facility Overview: Allows patients to browse hospital facilities, including doctors, departments, and available beds.  
- Patient Portal: Patients can register and log in to view personalized information, such as medical history and billing details.  
- Online Payment: Integration of secure payment methods for patients to pay bills.  
- Admin Dashboard: Provides hospital administrators with tools to manage users, facilities, and system resources.

## 2. Functional Requirements

### User Roles:

Patients:  
- Can browse hospital facilities and services.  
- Can create accounts to view personalized details (e.g., appointments, medical history).  
- Can pay bills online.  
  
Administrators:  
- Can manage the hospital system, including doctors, departments, and bed allocations.  
- Can view and manage patient accounts and records.  
- Can generate reports on hospital activities and resource utilization.

### Features:

- Facility Management: Displays detailed information about hospital facilities, such as doctor profiles, department services, and bed availability.  
- User Authentication: Secure login and registration system with password protection and validation.  
- Payment Integration: Allows patients to make online payments securely through integrated payment gateways.  
- Admin Tools: Administrators can manage users, update facility information, and view system analytics.

## 3. System Design and Architecture

### Application Architecture:

Backend:  
- Framework: ASP.NET Core for handling business logic, API endpoints, and database interactions.  
- Controllers: Manage HTTP requests and interact with services.  
  
Frontend:  
- Framework: Razor Pages for dynamic UI generation and user input handling.  
  
Database:  
- Technology: SQL Server for storing data, including user details, facilities, and payment transactions.  
- Entity Framework Core: ORM for seamless database interactions.

## 4. Technology Stack

Backend:  
- ASP.NET Core: Handles the core functionality and API interactions.  
  
Frontend:  
- Razor Pages: Renders dynamic content and provides user interaction capabilities.  
  
Database:  
- Entity Framework Core: Provides efficient database access.  
- SQL Server: Stores hospital and user data.  
  
Security:  
- ASP.NET Identity: Manages authentication and role-based access.

## 5. Database Design

Users:  
- UserID (Primary Key)  
- Name  
- Email  
- PasswordHash  
- RoleID (Foreign Key)  
  
Roles:  
- RoleID (Primary Key)  
- RoleName (Patient, Admin)  
  
Doctors:  
- DoctorID (Primary Key)  
- Name  
- Specialty  
- Department  
  
Facilities:  
- FacilityID (Primary Key)  
- Name  
- Description  
- Availability  
  
Payments:  
- PaymentID (Primary Key)  
- UserID (Foreign Key)  
- Amount  
- PaymentDate

## 6. Security Considerations

### Authentication:

Implemented using ASP.NET Identity for secure user login and role-based actions.

### Authorization:

Role-based access control ensures patients and administrators can only access authorized features.

### Data Protection:

Sensitive information, such as passwords and payment details, is encrypted for security.

## 7. Future Improvements

### Additional Features:

- Appointment Booking: Allow patients to schedule appointments with doctors.  
- Notifications: Send reminders for upcoming appointments or payments.  
- Reporting: Advanced analytics for hospital performance.

### Scalability:

Cloud hosting (e.g., Azure) and caching for better performance under high traffic.

## 8. Conclusion

This hospital management system provides a comprehensive solution to improve patient engagement and streamline hospital operations. Built on modern technologies like ASP.NET Core and SQL Server, the platform ensures scalability, security, and ease of use for all stakeholders.