

SOFTWARE REQUIREMENT SPECIFICATIONS

STUDENT NAME : SELVAKUMAR R

SEAT NO : 136

PROJECT ID : 16

PROJECT TITLE : HOSTEL MANAGEMENT PORTAL

1. INTRODUCTION:

Purpose:

The purpose of the hostel management portal is to automate and streamline administrative tasks related to hostel facilities within educational institutions. It aims to optimize room management, enhance user experience, improve administrative efficiency, manage data effectively, and enhance accountability.

Scope:

The project includes functionalities for user roles and authentication, room management (CRUD operations, booking, and change requests), hostel block management, reporting and analytics, user interface design, security and access control, integration and deployment.

2. FUNCTIONAL REQUIREMENTS:

User Management:

- User registration, login, and profile management.

- Admin and user roles with different access levels.

Room Management:

- Add, edit, and delete room details.

- Allocate rooms to students/staff.

- Change room status based on occupancy.

Hostel Block Management:

- Add/edit/delete hostel block details including area and capacity.

Booking System:

- Room booking for outsiders with approval process.

- Room change requests for hostellers.

Reporting:

- Generate reports on occupancy, room types, etc.

Settings:

- Admin panel for system configuration.

3. NON-FUNCTIONAL REQUIREMENTS:

Performance:

- System should handle concurrent users efficiently.

Security:

Secure user authentication and authorization mechanisms.

Data encryption for sensitive information.

Scalability:

Ability to scale the system as the number of users or hostels increases.

Usability:

Intuitive user interface for easy navigation.

Reliability:

Minimal downtime and data integrity assurance.

4. SYSTEM ARCHITECTURE:

MEAN stack architecture with MongoDB, Express.js, Angular.js, and Node.js.

Frontend (client-side):

Angular.js is used to create the user interface and experience. It manages client-side interactions, UI rendering, and data presentation.

Backend (Server Side):

Node.js is the runtime environment for the backend server. It handles server-side logic, routing, and database communication.

Express.js is a web application framework for Node.js that makes it easier to create sophisticated online apps. It includes middleware for managing requests, routing, and other server-side processes.

Database:

MongoDB is a NoSQL database that stores hostel-related data such as room details, block information, user profiles, booking records, and occupancy status. It provides flexibility and scalability, making it suited for managing a wide range of data types.

API:

APIs (Application Programming Interfaces) enable communication between frontend and backend components. RESTful APIs are often used to define endpoints for data access and manipulation.

5. USER INTERFACE DESIGN:

The application features a clean, organized layout with intuitive navigation, user-friendly forms and input fields, responsive design across devices, a dashboard overview, interactive components, visual feedback, consistent design elements, and accessibility by following web

accessibility standards. It also provides a dashboard overview, interactive components, and visual feedback upon completion. The application ensures a seamless user experience across various screen sizes.

6. TESTING:

Unit testing is a process used by developers to validate individual components of an application, such as a hostel management portal.

Integration testing ensures seamless integration and data flow.

User Acceptance Testing (UAT) involves real-world user interactions.

Regression testing ensures code changes don't introduce unintended side effects.

Performance testing evaluates the portal's responsiveness, scalability, and stability.

Security testing identifies vulnerabilities to prevent unauthorized access.

Usability testing assesses the portal's user interface design and experience.

Accessibility testing ensures the portal is usable by individuals with disabilities and complies with accessibility standards.

7. DEPLOYMENT:

The hostel management portal must be deployed by setting up the hosting environment, packaging the application, implementing CI/CD pipelines for automated releases, selecting a deployment strategy, performing database migration if necessary, configuring monitoring and security, and conducting post-deployment testing to ensure functionality and reliability in the production environment.

TECHNOLOGICAL STACK:

Frontend: Angular.js

Backend: Node.js, Express.js

Database: MongoDB

API: RESTful APIs / GraphQL APIs

FLOW CHART:

