Venue Management System Project Document

Name: Dhanush T

Register no.: 7376211EE112

Seat no.: 383

Project ID: 12

Problem statement: Venue Management

1. Project Overview

- **Project Title:** Venue Management System
- **Project Description:** This project aims to develop a web application for colleges to manage venue bookings for various events. It facilitates an online booking process and streamlines approval workflows for both organizers and administrators.

2. Project Requirements

Functional Requirements:

- User Management:
  - Two user types: Organization member and Admin.
  - User registration and login functionalities.
- Venue Management:
  - Admin can add, update, and delete venues from the system.
  - System maintains a list of available venues with details like name, capacity, facilities, etc.
- Event Booking:
  - Users can search and view available venues for their desired dates.
  - Users can submit booking requests specifying venue preference, event details (name, description, participants, duration, contact person), and preferred date(s).
  - System validates date availability and notifies users of conflicts.

- Admin Approval:
    - Admin can review submitted booking requests.
    - Admin can approve or reject requests with an optional message.
    - Admin can propose alternative venues if the preferred venue is unavailable.
- Notification System:
    - System notifies users about booking status (approval/rejection) and assigned venue.

Non-Functional Requirements:

- Security: Secure user authentication and authorization mechanisms.
- Scalability: The system should be scalable to accommodate an increasing number of users and bookings.
- Performance: The system should provide a responsive user experience with fast loading times.

3. System Architecture

A high-level diagram will typically show the following components:

- **Front-End:** React for building the user interface for both Guest/Organization and Admin users.
- **Back-End:** Node.JS with Express.JS framework for handling server-side logic and API development.
- **Database:** MongoDB for storing user data, venue information, booking details, and event information.
- **API:** OpenAPI for defining and documenting the application's APIs for communication between front-end and back-end.

4. Implementation Details

Front-End Development: React will be used to create interactive UI components for users to search venues, submit booking requests, and view booking status.

Back-End Development: Node.JS with Express.JS will handle API requests for user management, venue management, booking processing, and notification generation. MongoDB will store user data, venue information, event details, and booking data.

6. Testing Plan

Testing Strategy:

- Unit Testing: Individual components and functionalities will be tested for expected behaviour.
- Integration Testing: Communication and data flow between front-end and back-end will be tested.
- User Acceptance Testing: Users will test the application for usability and functionality.

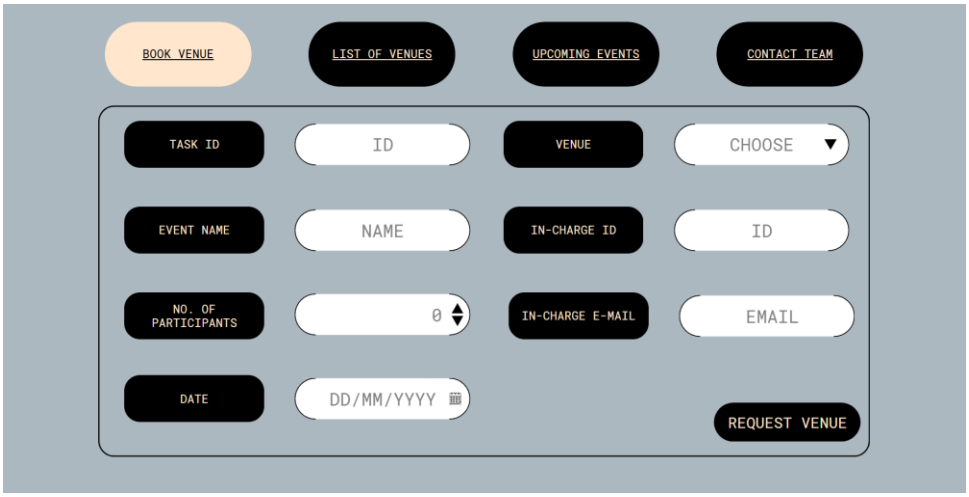
Testing Tools: Unit testing frameworks like Jest or Mocha will be used.

7. UI/UX

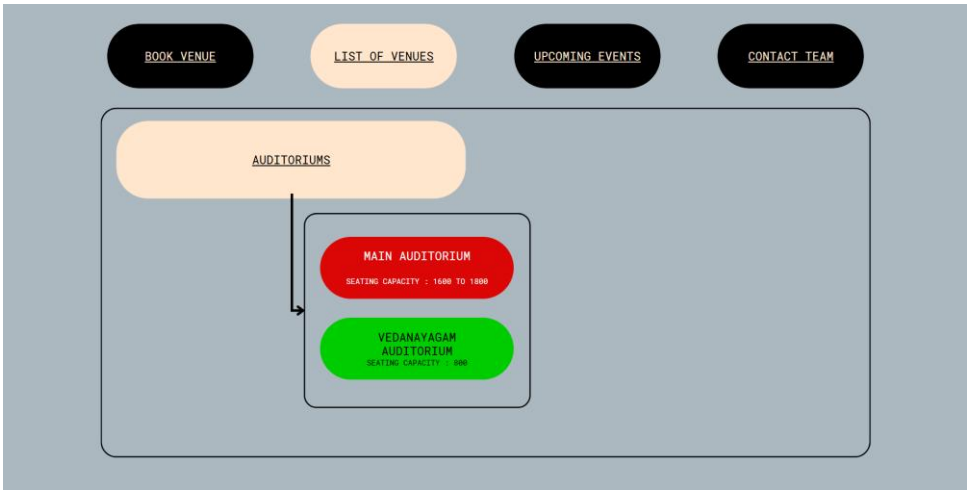
- Login page

- Home page for users

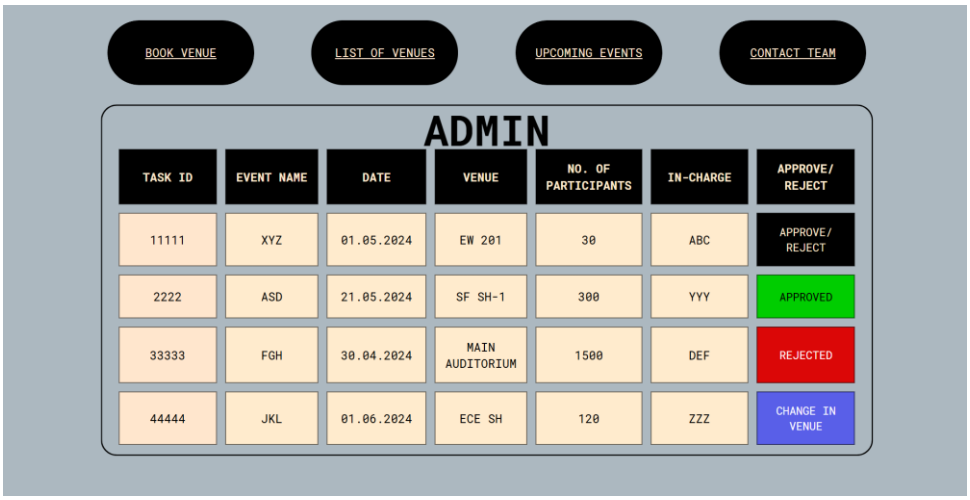
- Venue booking form



- List of all the available venues



- Admin page



- Sample of this website’s UI is available in this link: Venue management BIT

8. Deployment Plan

The application will be deployed on a cloud platform like Heroku or AWS. Version control will be implemented using Git for managing code changes.

9. Conclusion

This Venue Management System will streamline venue booking processes for colleges, allowing online submissions, efficient approval workflows, and clear communication channels.