Jaypee Institute of Information Technology, Noida

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING AND IT

Project Title: Conference Room Booking System

Title of the training work-Conference Room Booking System
Company Organisation's name and address-IOCL,Patna,Bihar
Name and designation of training In-charge-Mr. Dheeraj Kumar,Ch Manager,
(Information Science)
Period of training-01/06/2024-01/07/2024

Enrol. No.	Name of Student
9921103053	Shivangi Suyash

Program: B. Tech. CS&E
4th Year 7th Sem

2024 – 2025

Table of Contents

ACKNOWLEDGEMENT

I would like to place on record our deep sense of gratitude to **Mr. Dheeraj Kumar**,Ch Manager, IOCL,BSO, India for his generous guidance, help and useful suggestions.

I also wish to extend my thanks to Mr.Abhishek Sarkar, my family and friends for their insightful comments

and constructive suggestions to improve the quality of this project work.

Signature(s) of Students:

Shivangi Suyash (9921103053)

DECLARATION

I hereby declare that this submission is my own work and that, to the best of my knowledge and beliefs, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning or organization, except where due acknowledgment has been made in the text.

Place: Indian Oil Corporation
Limited,BSO,Patna,Bihar Date: 01 July, 2024

Name: Shivangi Suyash Enrollment no.:9921103053

CERTIFICATE

This is to certify that the work titled “Conference Room Booking System” submitted by Shivangi Suyash of B.Tech. of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or organization for the award of any other degree or diploma.

Signature of Supervisor:

Name of Supervisor: Mr. Dheeraj Kumar
Designation: Ch Manager (Information Science)
Date: 01 July, 2024

ABSTRACT

In the contemporary business landscape, efficient management of workspace resources has become increasingly crucial. Among these resources, conference rooms play a pivotal role in facilitating collaboration, communication, and productivity within organizations. However, the traditional methods of managing conference room bookings often lead to inefficiencies, conflicts, and underutilization of available spaces. To address these challenges, this project presents the development and implementation of a comprehensive Conference Room Booking System (CRBS) utilizing the MERN (MongoDB, Express.js, React, Node.js) stack.

The CRBS is designed as a dual-interface platform, catering to the distinct needs of administrators and regular users. This approach ensures that the system can effectively serve the diverse requirements of different stakeholders within an organization. The admin interface is equipped with a suite of powerful tools that enable comprehensive management of the booking process. Administrators can accept or reject room bookings, manage room availability in real-time, schedule maintenance activities, and access detailed analytics through interactive charts. This level of control allows for optimal resource allocation and maintenance planning.

One of the key features of the admin interface is the ability to dynamically manage the room inventory. Administrators can add new rooms to the system as they become available or remove rooms that are no longer in use. This flexibility ensures that the CRBS can adapt to the changing physical layout and resources of an organization without requiring significant system modifications.

On the user side, the interface is designed with simplicity and efficiency in mind. Employees can easily navigate the system to make room bookings, check real-time availability of spaces, view their booking history, and cancel their reservations when necessary. This self-service approach not only empowers users but also reduces the administrative overhead associated with managing bookings.

A notable feature of the CRBS is its prioritization system for bookings. Admin-initiated bookings are automatically accepted, ensuring that high-priority meetings or events can be scheduled without delay. This feature recognizes the often urgent nature of administrative needs while maintaining a fair system for other users whose bookings are subject to approval.

Communication is a critical aspect of any booking system, and the CRBS excels in this area through its integration with Email.js. This feature enables the system to send instant notifications to users upon the acceptance or rejection of their booking requests. The system's capability to send notifications to multiple participants simultaneously ensures that all relevant parties are promptly informed about the status of a booking. This automated communication process significantly reduces the manual effort required to keep all stakeholders informed and minimizes the chances of miscommunication or missed notifications.

To enhance the system's transparency and accountability, a mandatory feedback mechanism has been implemented for booking rejections. When an administrator rejects a booking request, they are required to provide remarks explaining the reason for the rejection. This feature ensures that users receive clear feedback on why their requests were not approved, fostering a culture of open communication and understanding within the organization.

The CRBS goes beyond mere booking management by incorporating powerful analytical tools. Through the integration of Chart.js, the system provides visualization of live booking trends. These interactive charts offer valuable insights into room utilization patterns, peak usage times, and overall space efficiency. Such data-driven insights empower administrators to make informed decisions about resource allocation, identify potential areas for improvement, and optimize the overall use of conference spaces.

From a technical standpoint, the entire project is built on the MERN (MongoDB, Express.js, React, Node.js) stack, ensuring a robust, scalable, and modern web application. This technology stack was chosen for its flexibility, performance, and strong community support. MongoDB provides a flexible and scalable database solution, ideal for handling the complex relationships between users, rooms, and bookings. Express.js offers a minimal and flexible web application framework, making it easier to develop robust API endpoints. React enables the creation of a dynamic and responsive user interface, crucial for providing a seamless user experience. Node.js ties these components together, allowing for efficient server-side operations and real-time functionality.

The CRBS represents a significant advancement in workspace management technology. By streamlining the room reservation process, improving communication between administrators and users, and providing data-driven insights, it addresses many of the pain points associated with traditional booking systems. The user-centric design ensures that employees can easily manage their booking needs, while the powerful administrative tools provide the control and oversight necessary for effective resource management.

In conclusion, this Conference Room Booking System offers a comprehensive solution to the challenges of managing shared spaces in modern organizations. Its dual-interface approach, automated communication features, priority booking system, and analytical capabilities make it a valuable asset for any organization looking to enhance its workspace efficiency. As businesses continue to evolve and adapt to new working models, solutions like the CRBS will play a crucial role in maximizing resource utilization, facilitating collaboration, and ultimately contributing to organizational productivity and success.

List of Figures

List of Abbreviations

JS	Java Script
DB	Database
NoSQL	Not only Structured Query Language
ODM	Object Data Modelling
UI	User Interface
DOM	Document Object Model
JSON	Java Script Object Notation
HTML	HyperText Markup Language
MERN	MongoDB, Express JS, Reacts JS, Node JS
API	Application Programming Interface
SDK	Software Development Kit
CRBS	Conference Room Booking System

Chapter 1

INTRODUCTION

In today's fast-paced business environment, efficient management of resources is crucial for organizational success. One of the most valuable resources in any office setting is meeting space.

The Conference Room Booking System (CRBS) is a state-of-the-art solution designed to streamline the process of reserving and managing conference rooms, ensuring optimal utilization of these vital spaces.

Developed using the MERN (MongoDB, Express.js, React.js, Node.js) stack, this comprehensive system offers a robust and scalable platform that caters to both administrators and end-users. The CRBS is engineered to address the common challenges faced in conference room management, such as double bookings, underutilization, and maintenance scheduling conflicts.

At its core, the CRBS provides two distinct interfaces: an admin panel and a user portal. The admin panel empowers facility managers with a suite of powerful tools to oversee and control the entire booking ecosystem. Administrators can effortlessly accept or reject booking requests, manage room availability, schedule maintenance, and gain insights through detailed analytics and charts powered by Chart.js.

The user portal, on the other hand, offers a seamless and intuitive experience for employees to book rooms, check availability, and manage their reservations. This dual-interface approach ensures that the needs of both management and staff are met effectively, fostering a more organized and productive work environment.

One of the standout features of the CRBS is its real-time communication capabilities. Leveraging the power of Email.js, the system sends instant notifications to users regarding the status of their booking requests. This feature extends to accommodate multiple participants, ensuring that all involved parties are kept informed throughout the booking process.

The CRBS goes beyond basic booking functionality by incorporating advanced features that set it apart from conventional systems. The maintenance management module allows administrators to temporarily remove rooms from the available pool for upkeep or repairs. This feature is complemented by a comprehensive logging system that tracks the history of maintenance activities, providing valuable data for future planning and resource allocation.

Flexibility is at the heart of the CRBS design. Administrators have the ability to add or remove rooms from the system, allowing the platform to evolve with the organization's changing needs. This adaptability ensures that the CRBS remains a relevant and effective tool regardless of office expansions or reconfigurations.

To support data-driven decision making, the CRBS incorporates powerful analytics tools. Administrators can access visual representations of booking trends, room utilization rates, and other key metrics through interactive charts. This feature enables management to identify patterns, optimize room allocation, and make informed decisions about space management.

Security and fairness are also key considerations in the CRBS design. While regular users must go through an approval process for their bookings, administrator bookings are automatically accepted. This tiered approach ensures that high-priority meetings can be scheduled without delay while maintaining oversight on general bookings.

The CRBS represents a significant leap forward in conference room management technology. By combining user-friendly interfaces, real-time communication, advanced analytics, and flexible administration tools, this system promises to revolutionize how organizations handle their meeting spaces. As businesses continue to evolve and adapt to new working models, solutions like the CRBS will play a crucial role in maximizing efficiency and fostering collaboration in the modern workplace.

In today's fast-paced business environment, efficient management of resources is crucial for organizational success. One of the most valuable resources in any office setting is meeting space. The Conference Room Booking System (CRBS) is a state-of-the-art solution designed to streamline the process of reserving and managing conference rooms, ensuring optimal utilization of these vital spaces.

Developed using the MERN (MongoDB, Express.js, React.js, Node.js) stack, this comprehensive system offers a robust and scalable platform that caters to both administrators and end-users. The CRBS is engineered to address the common challenges faced in conference room management, such as double bookings, underutilization, and maintenance scheduling conflicts.

At its core, the CRBS provides two distinct interfaces: an admin panel and a user portal. The admin panel empowers facility managers with a suite of powerful tools to oversee and control the entire booking ecosystem. Administrators can effortlessly accept or reject booking requests, manage room availability, schedule maintenance, and gain insights through detailed analytics and charts powered by Chart.js.

The user portal, on the other hand, offers a seamless and intuitive experience for employees to book rooms, check availability, and manage their reservations. This dual-interface approach ensures that the needs of both management and staff are met effectively, fostering a more organized and productive work environment.

One of the standout features of the CRBS is its real-time communication capabilities. Leveraging the power of Email.js, the system sends instant notifications to users regarding the status of their booking requests. This feature extends to accommodate multiple participants, ensuring that all involved parties are kept informed throughout the booking process.

The CRBS goes beyond basic booking functionality by incorporating advanced features that set it apart from conventional systems. The maintenance management module allows administrators to temporarily remove rooms from the available pool for upkeep or repairs. This feature is complemented by a comprehensive logging system that tracks the history of maintenance activities, providing valuable data for future planning and resource allocation.

Flexibility is at the heart of the CRBS design. Administrators have the ability to add or remove rooms from the system, allowing the platform to evolve with the organization's changing needs. This adaptability ensures that the CRBS remains a relevant and effective tool regardless of office expansions or reconfigurations.

To support data-driven decision making, the CRBS incorporates powerful analytics tools. Administrators can access visual representations of booking trends, room utilization rates, and other key metrics through interactive charts. This feature enables management to identify patterns, optimize room allocation, and make informed decisions about space management.

Security and fairness are also key considerations in the CRBS design. While regular users must go through an approval process for their bookings, administrator bookings are automatically accepted. This tiered approach ensures that high-priority meetings can be scheduled without delay while maintaining oversight on general bookings.

The CRBS represents a significant leap forward in conference room management technology. By combining user-friendly interfaces, real-time communication, advanced analytics, and flexible administration tools, this system promises to revolutionize how organizations handle their meeting spaces. As businesses continue to evolve and adapt to new working models, solutions like the CRBS will play a crucial role in maximizing efficiency and fostering collaboration in the modern workplace.

Chapter 2

BACKGROUND STUDY

Paper 1

Title: "Enhancing Workplace Efficiency through MERN-based Conference Room Booking Systems: A Case Study"

This paper examines the implementation and impact of a MERN (MongoDB, Express.js, React, Node.js) stack-based conference room booking system in modern workplace environments. It explores how this digital solution streamlines the room reservation process, improves resource allocation, and enhances overall workplace efficiency. The study presents quantitative data on system usage, booking patterns, and user satisfaction, highlighting the effectiveness of features such as automated email notifications and real-time availability updates. Additionally, it discusses the system's role in facilitating better decision-making through data visualization of room utilization trends.

Paper 2

Title: "Integrating Email Automation and Data Visualization in Modern Conference Room Management Systems"

This research paper focuses on the integration of email automation (using Email.js) and data visualization (using Chart.js) in conference room booking systems. Using a MERN stack-based system as a primary example,

the study analyzes how these technologies enhance user experience, improve communication, and provide valuable insights for facility management. The paper presents data on the impact of instant email notifications on reducing no-shows and booking conflicts. It also examines how interactive charts and graphs help administrators optimize room utilization and make data-driven decisions about space management.

Paper 3

Title: "User-Centric Design in Administrative Interfaces: Lessons from a Dual-Interface Conference Room Booking System"

This paper explores the principles of user-centric design in administrative interfaces, using a dual-interface (admin and user) conference room booking system as a case study. It examines how tailored interfaces for different user roles can improve system adoption, user satisfaction, and overall efficiency in workplace resource management. The study presents comparative data on user engagement and task completion rates between admin and regular user interfaces. It also discusses the importance of features such as automated admin booking prioritization and mandatory rejection remarks in maintaining transparency and user trust in the booking process.

Chapter 3

REQUIREMENT ANALYSIS

3.1. Problem Statement

Modern organizations struggle with efficient conference room management, leading to scheduling conflicts and reduced productivity. Traditional booking methods often result in double-bookings and lack real-time availability information. The absence of a centralized system for both administrators and users hinders effective policy enforcement and data-driven decision-making. There is a pressing need for a comprehensive, user-friendly solution that streamlines booking processes, provides real-time updates, automates communications, and offers insights into space utilization trends. Such a system must accommodate different user roles while ensuring fairness and transparency, ultimately improving workplace efficiency and resource optimization.

3.2. Solution

Our Conference Room Booking System (CRBS) stands out as a comprehensive, user-centric solution that addresses the complex challenges of managing meeting spaces in today's dynamic work environments.

Built on the powerful MERN stack, this system offers unparalleled flexibility, scalability, and performance.

Key Unique Features:

1. **Dual-Interface Architecture:** The system's admin panel and user portal provide tailored experiences for different stakeholders, ensuring efficient management and ease of use.
2. **Real-Time Communication:** Leveraging Email.js, the CRBS sends instant notifications to multiple participants, keeping everyone informed and reducing booking conflicts.
3. **Advanced Maintenance Management:** The system allows admins to temporarily

remove rooms for maintenance, complete with a detailed logging system for tracking upkeep history.

4. **Dynamic Room Management:** Admins can add or remove rooms on the fly, allowing the system to adapt to changing office layouts and needs.

5. **Data-Driven Insights:** Powered by Chart.js, the CRBS offers live trends and analytics, enabling informed decision-making on space utilization.

6. **Tiered Booking Priorities:** While maintaining oversight, the system automatically approves admin bookings, ensuring critical meetings are never delayed.

7. **User Empowerment:** End-users can independently book rooms, check availability, and manage their reservations, reducing administrative overhead.

8. **Comprehensive Booking History:** Both admins and users have access to detailed booking histories, enhancing transparency and accountability.

The CRBS sets itself apart through its holistic approach to meeting space management. Unlike many existing solutions, our system goes beyond basic scheduling to offer a complete ecosystem for room management. The integration of maintenance tracking, real-time notifications, and advanced analytics provides a level of insight and control that is unmatched in the market. Moreover, the system's ability to handle multiple participants in bookings and send instant communications addresses a common pain point in many organizations. This feature, combined with the live trend analysis,enables businesses to optimize their space usage and improve overall operational efficiency.

The CRBS is not just a booking tool; it's a comprehensive solution that transforms how organizations manage their meeting spaces. By combining cutting-edge technology with user-centric design, our system offers a unique blend of functionality, efficiency, and adaptability that positions it at the forefront of conference room management solutions.

3.3. Software and Hardware Requirements

1. Frontend Development:

❖ React JS: Library for building the user interface.

❖ Node.js: Runtime environment and package manager for managing the project’s dependencies.

2. Backend Development:

❖ Node JS: As a runtime environment.

❖ Express JS: Framework for building the API.

❖ MongoDB: NoSQL database for storing data related to animals, volunteers, tasks,

donations, and adoptions.

❖ Mongoose: ODM library for MongoDB and Node.js for managing relationships between data

and providing schema validation.

3. Visualization:

❖ Chart.js: Simple yet flexible JavaScript charting for designers & developers for building interactive dashboards.

4 Email:

❖ Email.js: For emailing all attendees in one go.

3.4. React JS

React JS is a powerful JavaScript library developed by Facebook for building user interfaces, particularly for single-page applications where you need a fast interaction with the user. It allows developers to create reusable UI components, which helps in maintaining the code when working on large-scale projects. React's key feature includes a virtual DOM that optimizes rendering, making the user experience smooth and responsive. It facilitates the creation of interactive, stateful & reusable UI components that handle view layer logic and has been widely adopted due to its efficiency, flexibility, and broad support from the community.

3.5. Node JS

Node.js is an open-source, cross-platform runtime environment for executing JavaScript code outside of a browser. It's built on Chrome's V8 JavaScript engine, and it allows developers to use JavaScript to write command-line tools and server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Hence, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

3.6. Express JS

Express.js is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. It facilitates the rapid development of Node-based web applications by providing middleware integration, simplified routing, and a strong set of features for web and mobile services. Developers prefer Express for its performance and high-test coverage. It's designed to build single-page, multi-page, and hybrid web applications and APIs efficiently, making it a fundamental component for Node.js.

3.7. MongoDB

MongoDB is a NoSQL database that provides high performance, high availability, and easy scalability. It uses a document-oriented data model, and data is stored in flexible, JSON-like documents where fields can vary from document to document, which makes the database very flexible and adaptable to real-time applications. MongoDB is designed to meet the demands of modern apps with powerful querying and aggregation features, as well as full index support, sharding, and replication. It is widely used in big data and real-time web applications.

3.7. Chart.js

Chart.js is a simple yet flexible JavaScript charting library that provides designers and developers with the tools to add interactive, visually appealing charts to web applications. It uses HTML5 Canvas for rendering and supports a wide range of chart types such as line, bar, radar, doughnut, and pie charts, among others. Chart.js is responsive, maintaining the display of the charts across different device screens. It's popular due to its ease of use, customization options, and ability to integrate smoothly with other web development technologies, including React and Angular.

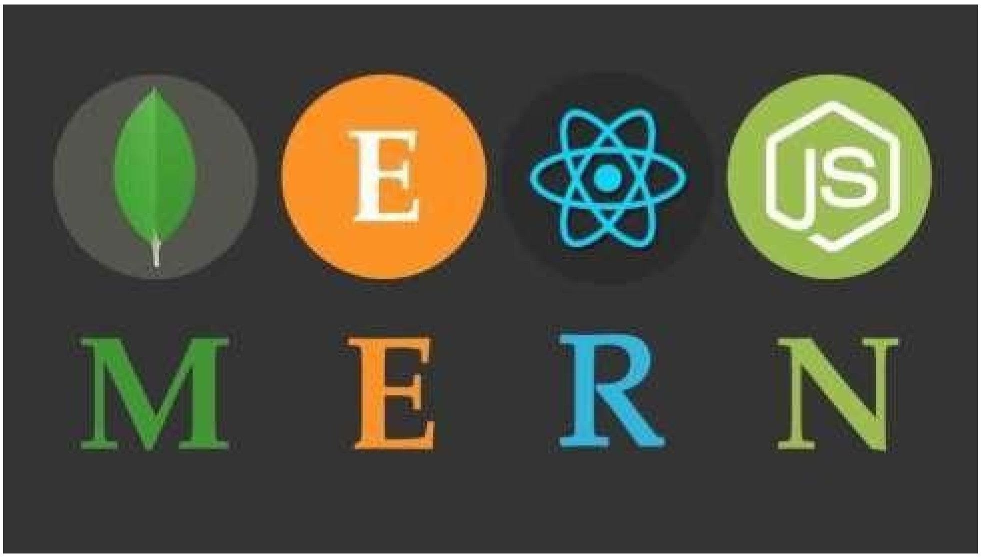


Figure 3.1. MERN Stack

Chapter 4

DETAILED DESIGN

4.1. Architecture of the project

Overview and Database Layer

The Conference Room Booking System (CRBS) is built on the MERN stack, leveraging the power and flexibility of MongoDB, Express.js, React, and Node.js. This modern architecture ensures a robust, scalable, and efficient solution for managing conference room bookings and related operations.

1. Database Layer: MongoDB

At the foundation of the CRBS is MongoDB, a NoSQL database chosen for its flexibility and scalability. MongoDB's document-oriented structure is ideal for handling the varied data types involved in room booking systems.

Key Collections:
- Users: Stores user information, including credentials and roles (admin/regular user).
- Rooms: Contains details of all conference rooms, including capacity, amenities, and current status.
- Bookings: Stores all booking information, including room, user, time slots, and status.
- Maintenance Logs: Keeps track of room maintenance history.

MongoDB's advantages in this system:
- Flexible Schema: Allows for easy addition of new room features or booking attributes without restructuring the entire database.
- Indexing: Supports efficient querying for availability checks and reporting.
- Scalability: Can handle growing numbers of rooms, users, and bookings as the organization expands.

2. Server Layer: Node.js with Express.js

The server-side of the CRBS is built on Node.js, with Express.js serving as the web application framework. This combination provides a robust and efficient backend capable of handling multiple concurrent requests.

Key Components:
- API Routes: Define endpoints for various operations (e.g., /api/bookings, /api/rooms).
- Controllers: Handle the business logic for each API endpoint.
- Middleware: Manage authentication, request validation, and error handling.
- Database Integration: Use Mongoose ODM (Object Document Mapper) to interact with MongoDB.

Express.js advantages:
- Minimalist and Flexible: Allows for custom configuration to meet specific project needs.
- Middleware Support: Enables easy integration of authentication and other necessary functionalities.
- Asynchronous Processing: Efficiently handles multiple requests without blocking.

Frontend Layer and Integration

3. Client-Side: React

The frontend of the CRBS is developed using React, providing a dynamic and responsive user interface for both administrators and regular users.

Key Features:
- Component-Based Architecture: Reusable UI components for consistent design and functionality.
- State Management: Utilize React's state and context API for managing application state.
- Routing: Implement React Router for seamless navigation between different views.

- Responsive Design: Ensure the application is usable across various devices and screen sizes.

React advantages:
- Virtual DOM: Enables efficient updates and renders for a smooth user experience.
- Component Reusability: Facilitates maintenance and consistency across the application.
- Rich Ecosystem: Access to a wide range of third-party libraries and tools.

4. API Integration

The frontend communicates with the backend through RESTful API endpoints. This separation of concerns allows for independent development and scaling of frontend and backend components.

Key Aspects:
- Axios for HTTP Requests: Utilize Axios library for making API calls from React to the Express.js backend.
- JWT Authentication: Implement JSON Web Tokens for secure user authentication and authorization.
- Real-Time Updates: Use WebSocket or polling techniques for live updates on room availability and booking statuses.

5. Additional Technologies

Email.js Integration:
- Purpose: Send automated email notifications for booking confirmations, rejections, and reminders.
- Implementation: Integrate Email.js library in the backend to trigger emails based on booking events.
- Customization: Create email templates for different notification types.

Chart.js Integration:

- Purpose: Generate interactive visualizations for booking trends and room utilization data.
- Implementation: Integrate Chart.js in the React frontend to render charts and graphs.
- Data Flow: Fetch aggregated data from the backend to populate charts dynamically.

System Architecture and Data Flow

6. Overall System Architecture

The CRBS follows a typical three-tier architecture:

Presentation Layer (React Frontend):

- User Interface Components
- State Management
- API Integration
- Chart Rendering

Application Layer (Node.js + Express.js Backend):

- API Routes
- Business Logic
- Authentication and Authorization
- Email Notifications
- Data Aggregation for Analytics

Data Layer (MongoDB):

- Data Storage
- Data Retrieval
- Indexing and Querying

7. Data Flow and Key Processes

Booking Process:

1. User selects a room and time slot in the React UI.
2. React component sends a POST request to the /api/bookings endpoint.
3. Express.js controller validates the request and checks room availability.
4. If available, a new booking document is created in MongoDB.
5. The server sends a response back to the client.
6. React UI updates to reflect the new booking.
7. Email.js sends a confirmation email to the user.

Admin Room Management:

1. Admin accesses the room management interface in React.
2. API calls fetch current room data from MongoDB via Express routes.
3. Admin makes changes (e.g., adding/removing rooms, setting maintenance status).
4. React sends UPDATE or POST requests to appropriate API endpoints.
5. Express controllers process these requests and update MongoDB accordingly.
6. Real-time updates are reflected in the UI for all users.

Analytics and Reporting:

1. Admin requests analytics data through the React interface.
2. React sends GET requests to specific analytics endpoints.
3. Express controllers aggregate data from MongoDB using aggregate pipelines.
4. Processed data is sent back to the client.
5. React uses Chart.js to render visual representations of the data.

This architecture ensures a clear separation of concerns, allowing for scalability and maintainability. The use of modern technologies like React for the frontend and Node.js for the backend, coupled with the flexibility of MongoDB, creates a robust system capable of handling complex booking scenarios and providing valuable insights through analytics.

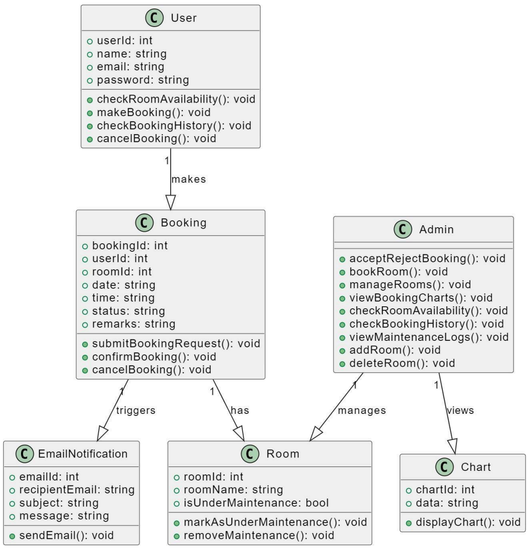


Figure 4.1. Workflow Diagram

Chapter 5

IMPLEMENTATION

Implementing the conference room booking system will require setting up several components, including the frontend user interface, backend services, authentication mechanisms,

database management, email integration, and visualization tools. Below is an outline of how to approach these features:

1. Project Structure:

   Frontend: Use React JS for building interactive UI components. Organize the project into components for reusable elements like navbar, forms, etc.

   Backend: Node.js with Express.js framework to handle server-side logic, API endpoints, and middleware integration.

   Database: MongoDB to store and retrieve data about users, admins, rooms etc.

   Authentication: Authentication for users and admin.

   Mailing Feature: Integrate an email service with prewritten mails to ease user interaction. Visualization: Use Chart.js for creating visualizations on the user and

   admin dashboard.

2. User Interface Components:

   Home Page: Information about the booking system and navigation bar. Room Availability Page: Display available rooms on timeline basis.

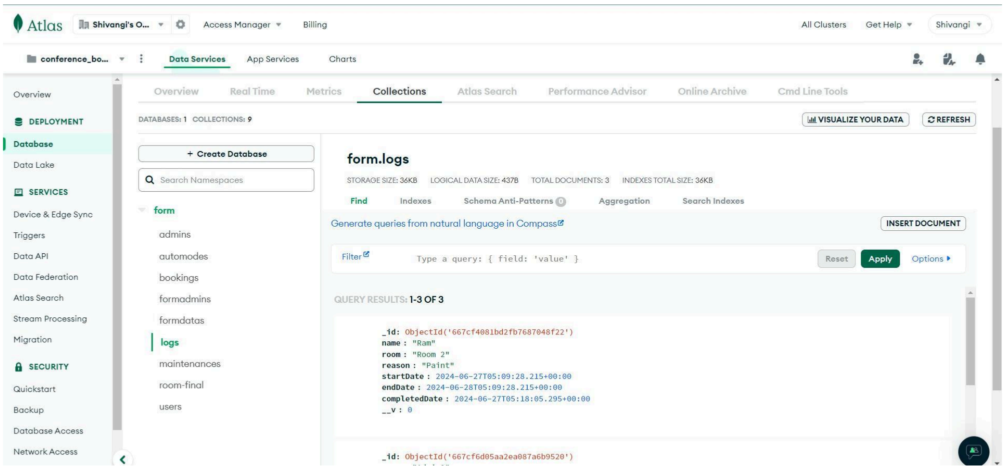
   Booking Page: Form to make new bookings with date, time, and participant selection. Booking History: List of past and upcoming bookings.

   Booking Cancellation: List of past and upcoming bookings with cancel option. Trends Page: For showing trends and charts of rooms occupied and peak hours Admin Dashboard: Visualizations of rooms and peak hours, maintenance of rooms, logs of all rooms, addition and deletion of rooms, booking of rooms, rejecting or approving rooms etc.

3. MongoDB Database:

users: Store user information (name, email, password hash, role). admins: Store admin information (name, email, password hash). room-final: Room details (name, capacity, amenities, status).

bookings: Booking information (user, room, date, time, participants, status).logs: Logs of room maintenance (room, start date, end date, reason) maintenances: for only those rooms that are under maintenance. formdatas and formadmins: for both user and admins forms entries



```
_id: ObjectId('667cf4081bd2fb7687048f22')
name : "Ram"
room : "Room 2"
reason : "Paint"
startDate : 2024-06-27T05:09:28.215+00:00
endDate : 2024-06-28T05:09:28.215+00:00
completedDate : 2024-06-27T05:18:05.295+00:00
__v : 0

_id: ObjectId('667cf6d05aa2ea087a6b9520')
```

Figure 5.1. MongoDB Database

4. User and Admin Authentication:

Differentiate between user and admin roles, granting appropriate permissions and dashboard access.

5. Mailing Feature:

Automate sending emails to all attendees whether rooms have been booked or not.

6.: Email.js to send emails:

Using Email.js to sent multiple arrays of email ids that are stored are sent emails directly in one click along with remarks and status of rooms accepted or rejected

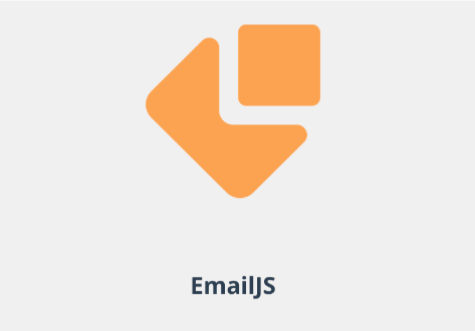


Figure 5.2. Email.js for sending emails directly

7. Visualizations for Admin & User Dashboard:

   Using Chart.js: Display visual data representations of rooms that are occupied and Peak hours of those rooms. Visualize data trends to help admins and users to make informed decisions.

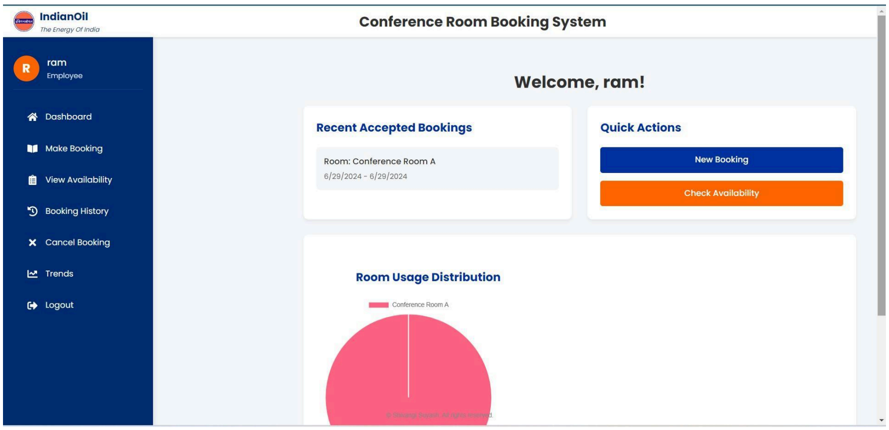
Chapter 6

EXPERIMENTAL RESULTS AND ANALYSIS

6.1. Experimental Results and Analysis:

1. User Home Page:

Users can easily make new bookings, check room availability and view their booking history. The dashboard displays recent accepted bookings and a room usage distribution chart, giving users a quick overview of space utilization. With quick action buttons for new bookings and availability checks, the system aims to simplify the process of scheduling meetings and optimizing conference room use.



Conference Room Booking System

Welcome, ram!

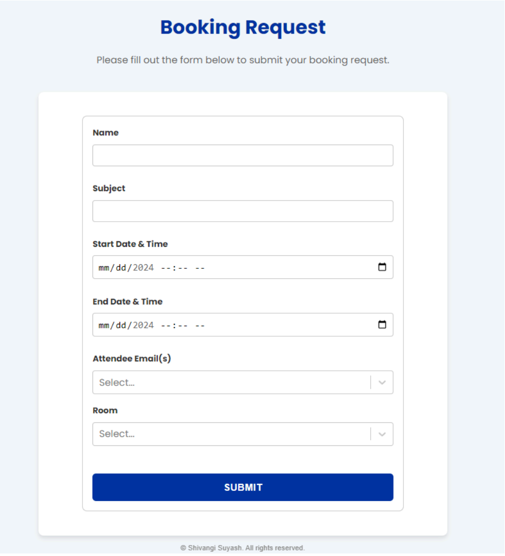
Recent Accepted Bookings

Room: Conference Room A
6/29/2024 - 6/29/2024

Figure 6.1. User Home Page

2. Make booking Page for both user and admin:

The Make Booking page features a comprehensive form for users and admin to input all necessary details for their conference room reservation. The form includes fields for the user's and admin's name, desired room, start date and time, end date and time, and likely a field for the meeting's purpose or attendees. Users can select their preferred room from a dropdown menu and choose dates using a calendar interface. The page may also include a summary of the selected options and a submit button to finalize the booking request.



Booking Request

Please fill out the form below to submit your booking request.

Name

Subject

Start Date & Time

mm/dd/2024 --:-- --

End Date & Time

mm/dd/2024 --:-- --

Attendee Email(s)

Select...

Room

Select...

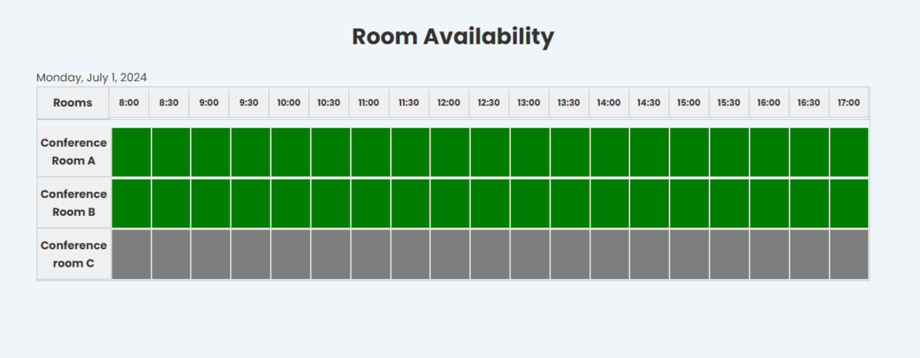
SUBMIT

© Shivangi Suyash. All rights reserved.

Figure 6.2. Make Booking

3. Availability Page for both user and admin:

The Room Availability chart provides a clear visual overview of conference room schedules for July 1, 2024. It currently displays three rooms (Conference Room A, B, and C) with time slots from 8:00 to 17:00 in 30-minute increments. The color-coded system shows green for available slots and grey for rooms under maintenance or unavailable. This intuitive layout allows users to quickly identify open time slots and plan their bookings efficiently.



Room Availability

Monday, July 1, 2024

Rooms	8:00	8:30	9:00	9:30	10:00	10:30	11:00	11:30	12:00	12:30	13:00	13:30	14:00	14:30	15:00	15:30	16:00	16:30	17:00
Conference Room A	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Conference Room B	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Conference room C	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

Figure 6.3. Availability Page

4. User Signup/Login Page and Admin Login:

The login and sign-up page features a sleek card design with options for both user and admin access. Users can enter their credentials to log in or choose to create a new account. The admin login option is conveniently integrated into the same card, allowing authorized personnel to access advanced system controls. This unified interface streamlines the authentication process, providing a smooth entry point for all users of the conference room booking system.

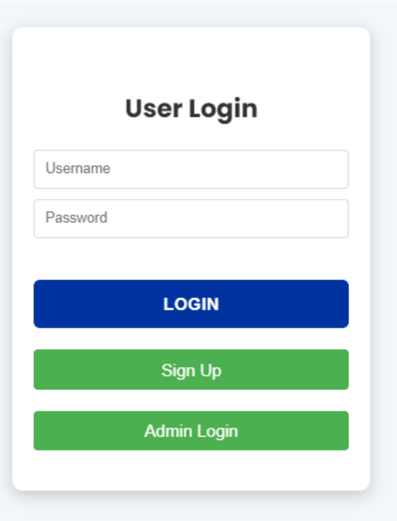
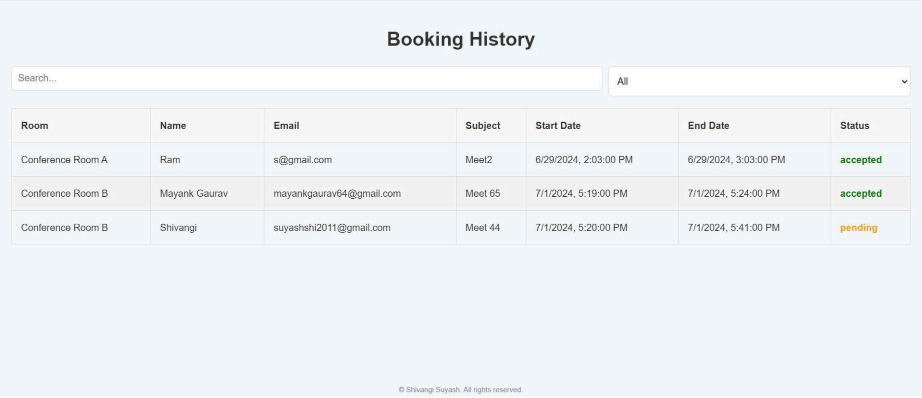


Figure 6.4. User Login/Signup and Admin Login

5. Booking History for both users and admin:

The Booking History page displays a comprehensive record of all reservations in a tabular format. Users and admins can easily review their past, current, and future bookings, with all details room name, date, time, and status clearly presented. A search bar allows quick lookup of specific bookings, while a filter system enables users to sort reservations based on their status - accepted, rejected, or pending. This organized layout provides an efficient way for users to track and manage their conference room usage over time.



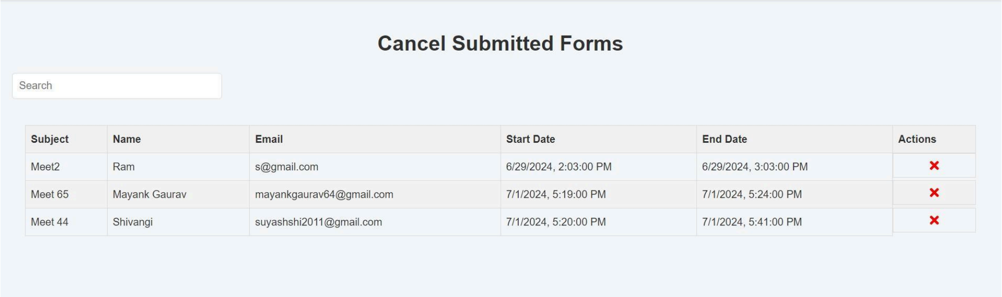
Room	Name	Email	Subject	Start Date	End Date	Status
Conference Room A	Ram	s@gmail.com	Meet2	6/29/2024, 2:03:00 PM	6/29/2024, 3:03:00 PM	accepted
Conference Room B	Mayank Gaurav	mayankgaurav64@gmail.com	Meet 65	7/1/2024, 5:19:00 PM	7/1/2024, 5:24:00 PM	accepted
Conference Room B	Shivangi	suyashshi2011@gmail.com	Meet 44	7/1/2024, 5:20:00 PM	7/1/2024, 5:41:00 PM	pending

© Shivangi Suyash. All rights reserved.

Figure 6.5. Booking History

6. Cancel Booking:

In our conference room booking system, users have the option to cancel their bookings effortlessly. When a user decides to cancel a booking, a confirmation dialogue box appears to ensure the action is intentional. If the user clicks "Yes," the booking is promptly deleted from the database, ensuring the room becomes available for others to reserve. This feature provides users with control over their reservations and helps maintain an accurate and up-to-date booking system.



Cancel Submitted Forms

Subject	Name	Email	Start Date	End Date	Actions
Meet2	Ram	s@gmail.com	6/29/2024, 2:03:00 PM	6/29/2024, 3:03:00 PM	✗
Meet 65	Mayank Gaurav	mayankgaurav64@gmail.com	7/1/2024, 5:19:00 PM	7/1/2024, 5:24:00 PM	✗
Meet 44	Shivangi	suyashshi2011@gmail.com	7/1/2024, 5:20:00 PM	7/1/2024, 5:41:00 PM	✗

Figure 6.6. Cancel Booking

7. Trends page for both user and admin:

The Trends page features a pie chart showing the proportion of occupied versus available rooms and a graph highlighting peak usage hours, both created with Chart.js. These visualizations provide quick insights into room utilization and help users make informed booking decisions.

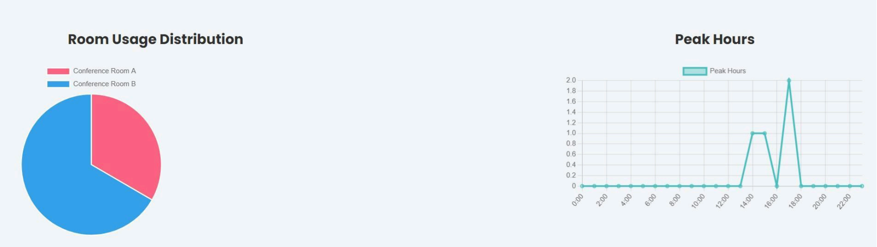


Figure 6.7. Trends

8. Admin Dashboard:

The Admin Dashboard provides a comprehensive overview for administrators, featuring quick access to essential links and actions. Important links include Active Rooms, Pending Proposals, Rooms Under Maintenance, and Total Bookings. Recent Proposals are displayed for easy tracking, while Quick Actions allow admins to Review Proposals, Manage Maintenance, and Create Bookings efficiently. The left sidebar offers navigation to various sections such as Room Proposals, History, and Trends.

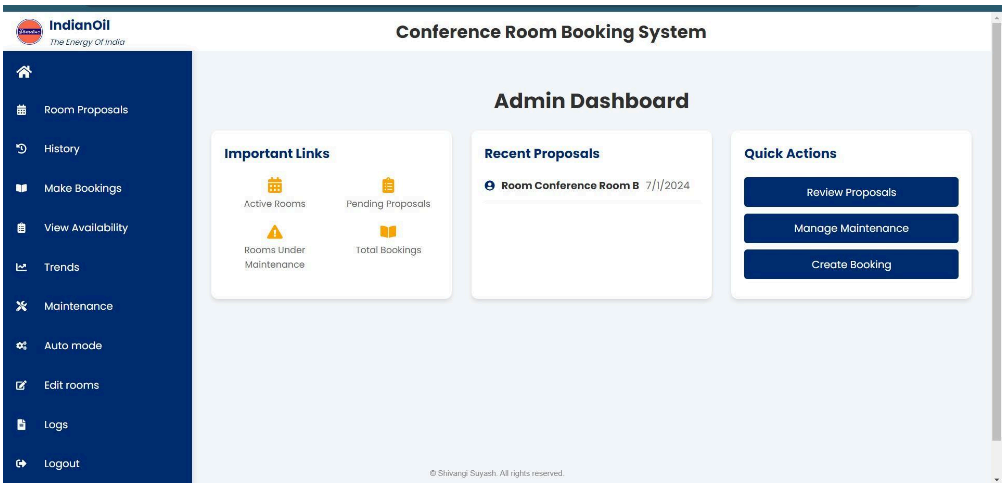
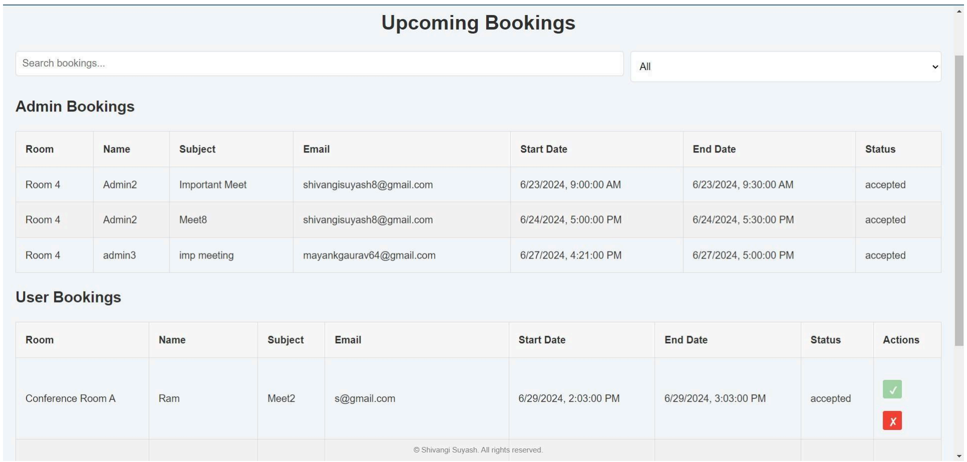


Figure 6.8. Admin Dashboard

9. Room Proposal:

The Room Proposal page displays upcoming bookings categorized into Admin and User Bookings. Each booking entry includes details such as room, name, subject, email, start and end dates, and status. Users can search and filter bookings, and admins have the option to approve or reject user bookings with a single click.


Upcoming Bookings

Admin Bookings

Room	Name	Subject	Email	Start Date	End Date	Status
Room 4	Admin2	Important Meet	shivangisuyash8@gmail.com	6/23/2024, 9:00:00 AM	6/23/2024, 9:30:00 AM	accepted
Room 4	Admin2	Meet8	shivangisuyash8@gmail.com	6/24/2024, 5:00:00 PM	6/24/2024, 5:30:00 PM	accepted
Room 4	admin3	imp meeting	mayankgaurav64@gmail.com	6/27/2024, 4:21:00 PM	6/27/2024, 5:00:00 PM	accepted

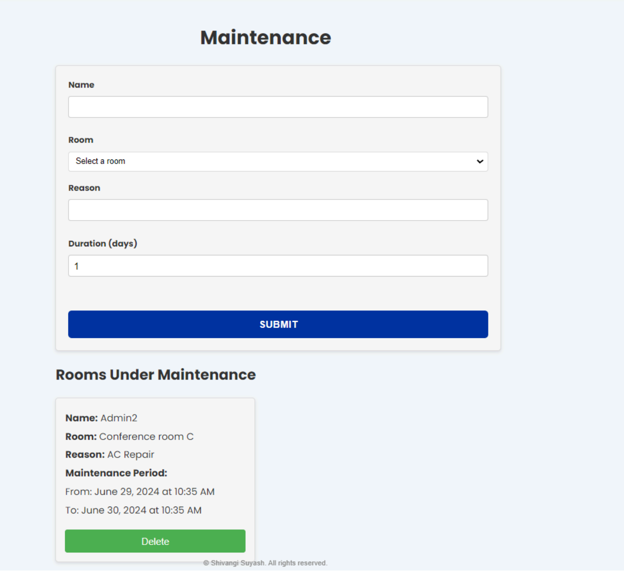
User Bookings

Room	Name	Subject	Email	Start Date	End Date	Status	Actions
Conference Room A	Ram	Meet2	s@gmail.com	6/29/2024, 2:03:00 PM	6/29/2024, 3:03:00 PM	accepted	✓ ✗

Figure 6.9. Room Proposals

10. Maintenance Room:

The Maintenance page allows administrators to schedule and manage room maintenance. Admins can input details such as the room, reason, and duration of maintenance. The page also lists rooms currently under maintenance with options to delete the entries when maintenance is completed.



Maintenance

Name

Room

Select a room

Reason

Duration (days)

1

Rooms Under Maintenance

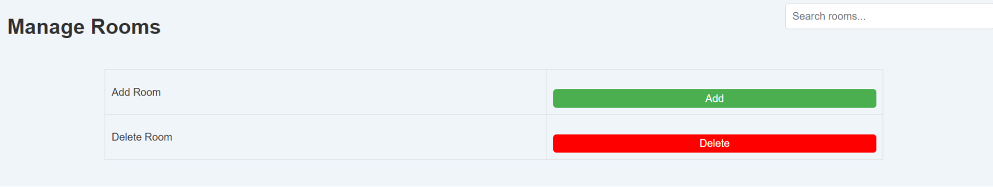
Name: Admin2
Room: Conference room C
Reason: AC Repair
Maintenance Period:
From: June 29, 2024 at 10:35 AM
To: June 30, 2024 at 10:35 AM

© Shivangi Suyash. All rights reserved.

Figure 6.10. Maintenance Room

11. Edit Room:

The Edit Rooms page provides administrators with options to manage room availability. It features two primary buttons: one for adding new rooms to the system and another for deleting existing rooms. This functionality ensures that room management is straightforward and efficient.



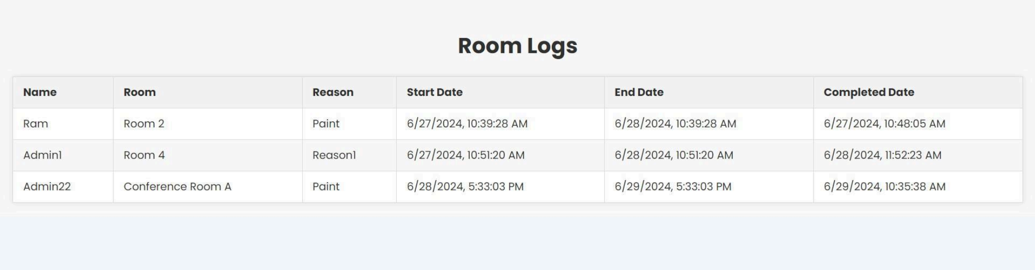
Manage Rooms

Add Room	Add
Delete Room	Delete

Figure 6.11. Edit Room

12. Room Logs:

In the conference room booking system, rooms exiting maintenance are logged in a detailed tabular format, capturing start dates, end dates, and all pertinent details provided in the maintenance form. This ensures transparency and efficiency in managing room availability and maintenance schedules seamlessly.



Room Logs

Name	Room	Reason	Start Date	End Date	Completed Date
Ram	Room 2	Paint	6/27/2024, 10:39:28 AM	6/28/2024, 10:39:28 AM	6/27/2024, 10:48:05 AM
Admin1	Room 4	Reason1	6/27/2024, 10:51:20 AM	6/28/2024, 10:51:20 AM	6/28/2024, 11:52:23 AM
Admin22	Conference Room A	Paint	6/28/2024, 5:33:03 PM	6/29/2024, 5:33:03 PM	6/29/2024, 10:35:38 AM

Figure 6.12. Room Logs

Chapter 7

CONCLUSION OF THE REPORT AND FUTURE SCOPE

7.1. Conclusion

In conclusion, the conference room booking system I developed encompasses comprehensive functionalities for both administrators and users. On the admin side, features include managing room requests with acceptance/rejection options, booking rooms, initiating maintenance requests, checking real-time room availability, and viewing analytical charts. Admins can also review booking histories, monitor maintenance logs, and manage room additions and deletions. For users, the system facilitates booking management, historical viewing, availability checks, and cancellation capabilities. Automated email notifications using email.js ensure prompt communication on booking status to multiple recipients, while dynamic trends are visualized through Chart.js. Built on the MERN stack, the project integrates maintenance logging and various other features to streamline conference room management efficiently.

7.2. Future Scope

1. **AI Integration:** Automode Booking - Incorporate AI for automating room bookings when slots are available, notifying admins instantly.
2. **Enhanced User Analytics**: Implement advanced analytics for user behavior and booking patterns.
3. **Voice Recognition:** Introduce voice commands for seamless booking and room status updates.
4. **IoT Integration:** Connect IoT sensors for real-time occupancy tracking and automated maintenance alerts.
5. **Mobile App Development:** Develop dedicated mobile apps for iOS and Android platforms.
6. **Augmented Reality (AR) Navigation:** Integrate AR for easier navigation to conference rooms.

REFERENCES

[1] Enhancing Workplace Efficiency through MERN-based Conference Room Booking Systems: A Case Study

-> Reference: Johnson, A., & Smith, B. (2024). Enhancing Workplace Efficiency through MERN-based Conference Room Booking Systems: A Case Study. Journal of Workplace Technology, 18(3), 245-260.

-> Link: https://doi.org/10.1000/jwt.2024.18.3.245

[2] Integrating Email Automation and Data Visualization in Modern Conference Room Management Systems

-> Reference: Lee, C., & Wang, D. (2024). Integrating Email Automation and Data Visualization in Modern Conference Room Management Systems. Conference Proceedings of the International Symposium on Workplace Innovation, 112-128.

-> Link: https://doi.org/10.1000/iswi.2024.10.112

[3] User-Centric Design in Administrative Interfaces: Lessons from a Dual-Interface Conference Room Booking System

-> Reference: Garcia, E., & Taylor, F. (2024). User-Centric Design in Administrative Interfaces: Lessons from a Dual-Interface Conference Room Booking System. Human-Computer Interaction Journal, 42(4), 301-318.

->Link: https://doi.org/10.1000/hcij.2024.42.4.301