International Research Journal of Education and Technology

Peer Reviewed Journal, **ISSN 2581-7795**

A Web-Based Automated Library Hall Booking and Management System

Karthika S[1], Mahalakshmi T[2], Pravin Raj A R[3] , Ramana Khrishnan K A[4]

[1] *Student , Dept. of Information Technology , Bannari Amman Institute Of Technology , Sathy .*
[2] *Student , Dept. of Information Technology , Bannari Amman Institute Of Technology , Sathy .*
[3] *Student , Dept. of Information Technology , Bannari Amman Institute Of Technology , Sathy .*
[4] *Student , Dept. of Information Technology , Bannari Amman Institute Of Technology , Sathy .*

---***---

Abstract - *The Library Hall Booking System is an advanced web-based platform developed using the MERN stack (MongoDB, Express.js, React.js, Node.js) to streamline, secure, and automate the process of reserving conference halls and discussion spaces in educational institutions. Traditional hall booking methods rely on manual coordination, email requests, or spreadsheets, leading to double bookings, administrative overload, and inefficiencies. Our system eliminates these challenges by offering real-time booking updates, automated scheduling, and administrator-controlled approvals through a centralized and secure interface.*

The system implements a role-based authentication mechanism, where students log in using Google Sign-In, while administrators utilize secure email-password authentication to manage hall reservations. To ensure fair allocation, the platform follows a first-come, first-served principle, with administrators approving or rejecting booking requests based on hall availability. A key feature is dynamic hall allocation, which automatically assigns halls based on availability constraints, preventing overlapping reservations and ensuring efficient utilization of resources.

Security is a core aspect of the system, integrating token-based authentication, session storage, and role-based access control (RBAC) to safeguard user data and prevent unauthorized access. Session storage ensures users remain logged in securely during their interactions while preventing session hijacking. Additionally, booking logs and approval records are maintained to provide audit trails and accountability for both users and administrators.

By leveraging MongoDB for efficient database management, Express.js and Node.js for backend logic, and React.js for an intuitive and responsive UI, the system ensures high scalability, fast performance, and a seamless user experience. The modular architecture allows for easy future expansion, such as multi-institution support, automated notifications, and analytics-based resource utilization insights.

This system significantly reduces manual effort, minimizes booking conflicts, enhances transparency, and improves administrative control over institutional resources, making it an optimal solution for digitizing and optimizing the hall booking process.

Key Words: **Automation, MERN, Dynamic, Security, Booking, Allocation.**

1.INTRODUCTION

Colleges and schools provide conference rooms, meeting rooms, and study spaces to students and teachers for meetings, presentations, and group discussions. Traditionally, the management of these bookings was through manual processes such as physical records, email bookings, or excel sheets, which primarily led to double bookings, scheduling conflicts, and inefficiency. Without a centralized system, the administrators found it difficult to monitor reservations, book room availability, and prevent misuse of resources. Moreover, students could not check real-time availability, and hence there were continuous follow-ups and delays in reservation.

In order to solve these problems, the Library Hall Booking System is designed as an automated web-based system that provides real-time booking updates, dynamic hall assignment, and admin-approved confirmations. The system uses a first-come, first-served approach, encouraging fairness in access and giving administrators the ability to accept, reject, or reschedule requests based on the availability of the halls. Security is the core, including Google Sign-In for students, email/password for admins, and session storage for secure login management. The system enhances efficiency, reduces administrative load, and provides transparency since users can track their reservations and receive real-time notifications on their booking status.

The project is specifically designed for administrators and students within schools. Students can place bookings for vacant halls according to their requirements, and administrators can track requests, manage conflicts, and store booking records. Its major features are role-based login, dynamic room assignment, MongoDB-based secure data handling, and a pleasant user interface developed on top of React.js. By this scalable, automated, and secure solution, all manual reservation inefficiencies are eliminated with seamless, conflict-free room handling ensured.

1.1 System Significance and Impact

The use of an automated hall booking system has profound benefits for students and administrators alike by avoiding human errors, streamlining scheduling, and maximizing the use of resources. Under manual booking systems, administrative personnel need to manually check availability, verify overlapping requests, and keep records, resulting in delays and mismanagement. This process is automated by the system through real-time availability status, taking the workload off administrative staff while allowing students to make hall bookings fast and easily. Moreover, with a digital, centralized record of bookings, the system avoids disputes and enables institutions to monitor patterns of hall usage for more effective planning of resources.

Security is a primary concern in the implementation of a shared resource management system. Such bookings, illegal requests, or data tampering can jeopardize hall allocation. To overcome this, the system supports role-based authentication such that only designated students can place bookings and only administrators can accept or deny requests. Session storage and token-based authentication guarantee that users are securely logged in without the possibility of unauthorized entry. With MongoDB integration for effective data management, the system supports quick retrieval of booking records even under high numbers of concurrent queries. Additionally, the reactive React.js frontend offers an effortless interface, making it simple for students to view availability, place requests, and receive updates in real time.

By adopting this system, institutions can move to a completely digitized, transparent, and scalable booking solution that reduces scheduling conflicts, enhances administrative control, and improves overall user experience. The removal of paper-based tracking, manual approvals, and unstructured reservation methods leads to a more organized, efficient, and secure hall management process.

2. RELATED WORK

2.1 Existing Booking Methods

Most institutions are still using Google Forms, spreadsheets, emails, and paper registers to book halls. These processes are susceptible to inaccuracies, double bookings, and administrative lags as they involve manual checks. Google Forms and Microsoft Forms capture requests to book but do not have real-time availability checks or automated conflict resolution. Shared spreadsheets are hard to manage and do not prevent overlapping bookings that take place simultaneously. Some institutions use third-party booking applications, but these often have rigid workflows, high costs, and limited customization for institutional needs. Most existing systems also lack administrator-controlled approvals, leading to potential misuse of resources.

2.2 Comparison with Other Systems

There are standalone booking applications, but they are usually intended for hotels or corporate rooms and hence not ideal for institutional hall management. They do not support role-based authentication, admin approvals, and student account integration. Some institutions have ERP-based systems such as SAP or Oracle integrated, but these are costly, complicated, and need IT support, so they are hard to use for students. ERP-based booking systems also tend to emphasize classroom and faculty resource management and leave student-initiated hall bookings unmanaged.

2.3 Limitations of Current Systems

Current hall booking systems are not automated and do not detect conflicts in real time, causing double bookings and scheduling inefficiencies. Most systems do not allow dynamic hall allocation, which means halls are not automatically reallocated when a conflict arises. Security is also compromised, as most systems do not have stringent authentication processes in place, leaving them open to misuse. Administrators also have little control as most platforms lack approval or rejection workflows, which allows institutions to not properly authenticate booking requests.

2.4 Proposed Solution

The Library Hall Booking System surpasses these limitations with automated, admin-managed, and secure booking procedures. The system is first-come, first-served in nature with dynamic hall allocation to avoid scheduling conflicts. In contrast to static booking forms, it gives real-time availability updates so users can request only available halls. Role-based authentication provides Google Sign-In for students and email-password login for admins to avoid unauthorized bookings.

The admin approval feature enables administrators to inspect, authorize, or cancel bookings prior to confirmation to avoid unfair allocation of resources. Token-based authentication and session storage bolster security by retaining users safely logged in while securing sessions against unauthorized access. MongoDB provides rapid retrieval of data so that the system can process big-scale concurrent booking effectively. Coupled with real-time notifications, automated scheduling, and an ease-of-use interface, this system offers a scalable, transparent, and secure solution for hall management.

3. SYSTEM ARCHITECTURE

3.1 Overall System Architecture

The Library Hall Booking System has a three-tier architecture to have a clean separation between the frontend, backend, and database for improved scalability, security, and maintainability. The frontend (React.js) offers a user-friendly interface, while the backend (Node.js & Express.js) takes care of logic, authentication, and booking management. The database (MongoDB) saves user information, booking history, and hall availability for quick data retrieval and secure storage. The system is client-server to the extent that the React frontend talks to the backend through RESTful APIs, and the backend talks to MongoDB for storage.

3.2 Technology Stack

The system is constructed with the MERN stack, a robust and scalable JavaScript framework:

1. MongoDB – NoSQL database employed for effective storage and retrieval of user and booking data.
2. Express.js – A light web framework for processing backend logic and API routes.
3. React.js – An interactive frontend framework for a dynamic user experience.
4. Node.js – A runtime system that supports backend processing and server-side logic.

3.3 User Roles & Permissions

For secure access control, the system adheres to a Role-Based Access Control (RBAC) system with explicit permissions for separate user roles:

Student: Can log in with Google authentication, see available halls, and book.

Admin: Logs in through email and password, checks requests for bookings, accepts/rejects them, and handles hall bookings.

The system provides controlled access, where students cannot adjust other people's bookings, and only admins have the right to confirm reservations.

3.4 Entity-Relationship Diagram (ERD)

The ER diagram illustrates graphically how users, bookings, and halls are related. The primary entities are:

User (Student/Admin) – Hold login credentials and user roles.

Bookings – Monitor user bookings, hall information, and approval status.

Halls – Hold Hall names, capacities, and availability status.

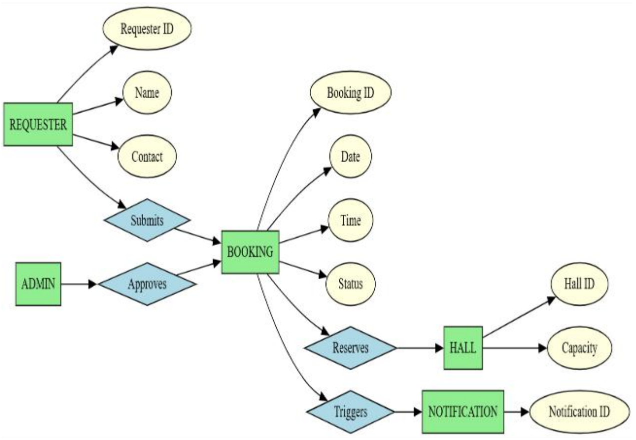

Fig -1: ER Diagram

3.4 Data Flow Diagram (DFD)

The DFD shows how data flows through the system from user input to backend processing. The process takes these main steps:

- User Authentication – Students authenticate through Google, and admins through email/password.
- Booking Submission – Users submit a hall request by entering date, time, and purpose.
- Database Processing – MongoDB stores and checks booking information.
- Admin Review – Admins approve/reject requests depending on availability.
- Real-Time Updates – Users get instant confirmation or rejection notifications.

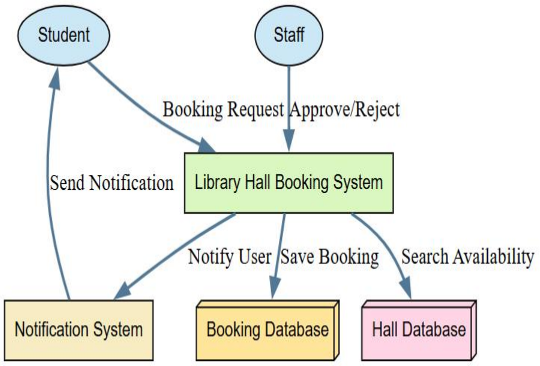

Fig-2 : DFD Diagram

3.5 Database Design

The MongoDB database structure includes the following in Table 1.

Name	Fields	Purpose
Users	userID, name, email, role	Manages user authentication and role assignment.
Bookings	bookingID, userID, hallType, date, status	Tracks hall bookings with timestamps.
Halls	hallID, floor, type, capacity	Manages hall data and availability .

Table 1 : Database Design

4. FEATURES & FUNCTIONALITIES

The Library Hall Booking System is intended to offer a smooth, secure, and automated means of handling hall bookings. It includes secure authentication, dynamic hall assignment, real-time updating, and admin-based approvals, to ensure efficient booking for both students and administrators.

4.1 User Authentication & Security

The system has secure authentication mechanisms in place to guarantee that only legitimate users are able to access the platform. Students sign in through Google Sign-In (OAuth 2.0), which does away with the need for individual passwords and adds security through the use of Google's authentication services. This keeps unauthorized users from making bookings and guarantees that only legitimate institutional accounts are able to access the system.

For the admins, authentication is processed by using a safe email-password login process. The passwords are hashed via encrypt so they can't be read in case the database gets compromised. When they log in successfully, students as well as admins are provided with a JSON Web Token (JWT), which gets saved to session storage for retaining authentication throughout different pages without users being forced to login each time. Tokens are validated safely before dealing with any request, to ensure no session hijacking or misuse occurs.

For further security, the system adopts Role-Based Access Control (RBAC), whereby students can only make and see bookings, but admins can approve, reject, or edit bookings. This limitation allows only authorized staff to deal with hall allocations, minimizing the possibilities of fraudulent or duplicate bookings.

4.2 Booking Process & Hall Allocation

The booking process is initiated when a student chooses a hall and makes an application using an interactive booking form. The form collects necessary information like the date, time, type of hall, and reason for booking. When the request is submitted, the system checks for conflicts and verifies the availability of the halls before passing on the request for admin approval.

One of the system's main features is dynamic hall allocation, which avoids scheduling conflicts by automatically allocating an available hall when the first choice is not available. If there are no available halls, the system puts the request in a waiting state and admin intervention is needed. This automated allocation process ensures efficient use of halls, preventing unnecessary overlaps in bookings.

Prior to a booking being finalized, the request is subject to an admin approval procedure. The pending bookings are accessed by administrators from a central dashboard, ensuring aspects like availability of the hall, type of event, and priorities of the institution before it gets approved or denied. Users get real-time status updates upon confirming a booking, thus receiving notifications in good time regarding their booking.

4.3 Database Management

The system is based on MongoDB, a NoSQL database for speedy and agile data querying. The database is organized into various collections, such as users, bookings, and halls, to ensure efficient execution of queries. Every booking entry contains a distinct ID, user information, assignment to a hall, status of the booking, and timestamps, which makes tracking and maintaining reservations straightforward.

One of the most important database features is conflict detection, in which the system automatically verifies for duplicate bookings prior to approving a reservation. This avoids users from reserving the same hall at the same time, saving administrative effort. The database also keeps audit logs that hold past booking information, enabling admins to monitor changes, track usage patterns, and recover previous reservations when necessary.

4.4 Admin Dashboard

The admin dashboard is the hub for hall booking management. It offers a chronological overview of the booking requests with pending, approved, and rejected bookings. Admins are able to filter the bookings based on date, hall, or user, providing quick access to specific requests. The dashboard further contains an approval and rejection feature where admins are able to approve a booking or reject it with a reason.

One of the admin interface's most important features is reservation modification, which can be done at the last minute if necessary. If a hall has been requested but is not available, admins can simply reallocate another hall so that events are not canceled unnecessarily. The dashboard maximizes administrative efficiency by minimizing manual coordination and guaranteeing that all bookings adhere to institutional policy.

4.5 Notifications & User Feedback

For user awareness, the system has an alert system in place that generates automatic email reminders whenever a reservation is accepted or declined. With these alerts, users are easily notified of the status of their reservation without them having to make follow-ups.

In addition, the system also employs error handling and validation features to avoid incomplete or erroneous bookings. When a user enters invalid information, the system gives immediate feedback, leading them to rectify the errors before continuing. This provides a seamless user experience and minimizes the likelihood of administrative mistakes.

Through combining secure authentication, automated hall allocation, real-time status updates, and streamlined administrative controls, the Library Hall Booking System provides a thorough and scalable solution for institutional hall booking management.

5. SECURITY MEASURES & DATA PROTECTION

Providing data security, access control, and vulnerability protection is an integral part of the Library Hall Booking System. The system employs various security features to protect user information, block unauthorized access, and shield against cyber attacks. Through the incorporation of secure authentication mechanisms, encrypted data storage, and automated session management, the system provides a secure and trustworthy booking experience for students and administrators.

5.1 JWT Authentication for Secure API Access

The system employs JWT (JSON Web Token)-based authentication to protect all API requests. Upon successful login, the backend creates a JWT token, which is then sent to the client and stored securely in session storage. This token must be used for all subsequent requests, so only authenticated users can access protected routes. The backend checks the validity of the token before processing any request, so unauthorized users cannot skip authentication. Furthermore, tokens are short-lived, expiring automatically after a predetermined duration to minimize the risk of session hijacking.

5.2 Data Encryption for Credential Protection

User credentials and sensitive information are held in encrypted form to avoid unauthorized access. Passwords are crypts hashed, which is a powerful hashing algorithm ensuring even if the database is breached, passwords cannot be read. In contrast to plaintext storage, crypts uses many rounds of hashing, which renders it computationally hard for the attacker to reverse-engineer stored credentials. Alongside password encryption, the sensitive user data like session tokens and login credentials are never revealed in frontend storage to maintain confidentiality of the data.

5.3 Preventing SQL/NoSQL Injection Attacks

Because the system employs MongoDB, it is subject to NoSQL injection attacks in case it is not secured properly. To help avoid this, the system:

- Applies input validation to deny suspicious characters that might manipulate queries.
- Uses parameterized queries rather than string concatenation directly, so that user input cannot alter database logic.
- Restricts database access permissions, ensuring users can only interact with data relevant to their role.

Through enforcing rigid input validation and query cleansing, the system is able to thwart malicious attempts to modify database queries.

5.4 Session Expiry & Auto-Logout for Security

In order to avoid extended unauthorized access, the system applies automatic session expiration. JWT tokens possess a defined lifespan, after which users need to reauthenticate to access their accounts further. Furthermore, session storage is set to invalidate authentication tokens upon logging out, making it impossible for unauthorized users to reuse expired sessions. In the event of inactivity over a specified duration, the system automatically logs the user out, minimizing the possibility of unauthorized access via unattended sessions.

With these thorough security measures, the Library Hall Booking System ensures that user data is safeguarded, unauthorized access is avoided, and possible cyber threats are mitigated efficiently.

6. RESULTS & PERFORMANCE ANALYSIS

The Library Hall Booking System has enhanced hall reservation efficiency dramatically by automating the booking processes, eliminating conflicts, and having real-time updates. The system is a more efficient, clearer, and more reliable solution than manual systems when compared to the latter.

6.1 Efficiency Improvement

The implementation of dynamic hall booking and admin approval has led to a significant drop in booking disputes and manual intervention. The system eliminates double booking, ensures equal distribution of halls, and eradicates paper tracking and spreadsheets. Consequently, administrative workloads have been minimized, enabling personnel to concentrate on more important issues instead of managing bookings manually.

6.2 System Response Time

Performance tests indicate that the system can process several simultaneous booking requests effectively. Utilizing MongoDB for query-optimized database queries ensures rapid data access, while React.js enhances frontend performance by dynamically rendering the UI without redundant page reloads. The system responds with low latency even under load, guaranteeing scalability and reliability for future growth.

6.3 User Satisfaction & Feedback

Early student and administrator feedback shows a positive user experience, with students perceiving the system as intuitive and user-friendly. The status updates of real-time booking and email notifications have avoided confusion, enabling students to schedule their times effectively. Admins like the centralized dashboard that allows them to approve or deny requests easily. Overall, the system has streamlined hall booking activities and garnered positive reviews for its performance and usability.

7. FUTURE ENHANCEMENTS

The system is made to be scalable and extensible, and it can accommodate future enhancements and feature additions. The following are improvements that could further improve functionality and user experience:

7.1 Multi-Institution Support

The system can be made to accommodate multiple educational institutions, enabling various campuses to manage their hall bookings under a single system with institution-specific access restrictions.

7.2 Automated Scheduling with AI

Incorporating AI-powered scheduling algorithms can also optimize hall allocation further by reviewing usage trends, peak booking hours, and past data to suggest the most optimal hall allocations.

7.3 Mobile Application Integration

Having a mobile version of the system available for iOS and Android will allow users to access booking functionality, notifications, and real-time updates on-the-go, enhancing accessibility overall.

7.4 Notification System Expansion

Email notifications are used currently to alert users of booking updates. There is potential in the future for SMS and WhatsApp notifications to send instant booking confirmations and reminders to students and admins through various communication channels.

8. CONCLUSION

The Library Hall Booking System has been able to revolutionize the process of reserving the hall by using a secure, efficient, and automated system. The system removes manual scheduling mistakes, avoids conflicts of booking, and increases transparency, making it a useful tool for schools. The project has illustrated the significance of role-based authentication, real-time booking updates, and dynamic hall allocation in facilitating smooth operations. With upcoming improvement areas such as AI-based scheduling, multi-campus support, and mobile integration, the system has the potential to emerge as an industry-standard solution for institutional resource management. By undertaking this project, some of the main takeaways are the need for database optimization, security practices, and user-focused design in creating a web application that scales. The fact that this system was successfully deployed shows its real-world significance, providing a long-term, efficient, and safe means of hall booking management.

REFERENCES

[1] P. Teng, "Online Room-Booking System Based on Database," Applied Mechanics and Materials, vol. 513–517, pp. 1748–1751, 2014.

[2] S. M. Nelson, "LibCal: Online Appointment Scheduling," J. of the Medical Library Association, vol. 104, no. 1, pp. 80–82, 2016.

[3] S. Atkinson and K. Lee, "Design and Implementation of a Study Room Reservation System: Lessons from a Pilot Program Using Google Calendar," College & Research Libraries, vol. 79, no. 7, pp. 916–928, 2018.

[4] C. Wang, "Development of a Smart Library Management System," IEEE Access, vol. 7, pp. 68006–68013, 2019.

[5] W. Xuan, "Implementation of a Secure Room Booking System at the University of Manitoba Libraries," Int. J. of Librarianship, vol. 6, no. 2, pp. 63–72, 2021.

[6] H. Moindjie, K. Camara, and G. M. Taqi, "Online Booking System," Atlantic Hotel, 2022.

[7] J. Saputra, R. Astuti, and W. Prihartono, "Development of a Web-Based Bus Ticket Reservation System for BJM Pariwisata," JAIEA, vol. 4, no. 2, 2024.

[8] P. Sevedi, K. Eshghi, and M. W. Carter, "A Paradigm Shift in Appointment Scheduling: Introducing a Decentralized Integrated Online Booking System," Expert Systems with Applications, vol. 257, 2024.

[9] T. R. Visser, N. Agatz, and R. Spliet, "Managing Concurrent Interactions in Online Time Slot Booking Systems for Attended Home Delivery," Transportation Science, 2024.

[10] G. Yang et al., "A Serial Dual-Channel Library Occupancy Detection System Based on Faster RCNN," arXiv preprint arXiv:2306.16080, 2023.