



(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 5, May 2025

⊕ www.ijirset.com 🖂 ijirset@gmail.com 🖄 +91-9940572462 🕓 +91 63819 07438





www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405134|

Hotel Booking Web Application using MERN Stack

Dr. S.N. Gunjal, Piyush Mohojkar, Dnyaneshwar Raundal

Assistant Professor, Department of Computer, SRES'S Sanjivani College of Engineering Kopargaon,

Maharashtra, India

Student, Department of Computer, SRES'S Sanjivani College of Engineering Kopargaon, Maharashtra, India

Student, Department of Computer, SRES'S Sanjivani College of Engineering Kopargaon, Maharashtra, India

ABSTRACT: This research presents the design and implementation of a modern, scalable Hotel Management System (HMS) using the MERN stack—MongoDB, Express.js, React.js, and Node.js. The paper outlines how the system addresses limitations of traditional hotel booking platforms by offering a seamless user experience, real-time booking capabilities, administrative management tools, and high performance through full-stack JavaScript architecture. The methodology follows a systematic software engineering approach, including requirement gathering, system design, development, testing, deployment, and evaluation. The research further evaluates the system's usability, scalability, and performance, positioning the MERN stack as an effective solution for digital transformation in the hospitality industry.[1]

KEYWORDS: - MERN stack, MongoDB, Express.js, React.js, Node.js, Full-stack development, Hotel Management System, Real-time booking, Web application, Hospitality industry.

I. INTRODUCTION

With the growing dependency on online systems across industries, the hospitality sector is evolving rapidly through digital transformation. Traditional hotel management systems often lack scalability, responsiveness, real-time updates, and user-centric design. These limitations can affect customer satisfaction and operational efficiency. The emergence of full-stack web development technologies, particularly those based on JavaScript, offers a cohesive environment for building powerful, interactive, and scalable web applications.

This paper presents a complete Hotel Management System (HMS) built on the MERN stack, which enables a robust backend, dynamic frontend, and flexible database design. This research not only elaborates on the implementation process of the system but also provides an evaluation of its real-world applicability, performance, and user experience.

II. LITERATURE SURVEY

2.1 Evolution of Hotel Booking Platforms

Hotel booking systems have evolved from manual entries to sophisticated software platforms. Earlier client-server applications and desktop-based software solutions posed limitations in multi-user environments and required manual intervention. Web-based systems emerged in the 2000s, driven by increased internet access and globalization.

As per research by Lama Dev (2024), web-based hotel systems improve operational efficiency and customer satisfaction by providing features like dynamic room pricing, real-time availability, and secure payments. Modern systems integrate multiple components such as booking engines, payment gateways, and CRM tools.[2]





www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405134|

III. PROPOSED METHODOLOGY

PROBLEM STATEMENT

In the current digital era, travelers expect to browse, compare, and book hotels seamlessly online. However, many existing hotel booking systems suffer from outdated interfaces, inefficient backend operations, and lack of real-time availability. This leads to poor user experience, booking conflicts, and increased administrative overhead for hotel staff.

The objective of this project is to design and develop a full-stack Hotel Booking System using the MERN stack that provides a responsive and user-friendly platform for customers to search, view, and book hotel rooms based on real-time availability. Simultaneously, hotel administrators should be able to manage room listings, availability, pricing, and bookings through a secure dashboard.

Key features will include:

- User authentication and role-based access (Admin/Customer)
- Room browsing with filters (price, availability, type, etc.)
- Real-time booking and payment integration
- Admin panel for room and booking management
- MongoDB for storing user, hotel, and booking data efficiently

IV. METHODOLOGY



Figure 1: System Architecture.

Figure 1 illustrates the system architecture of the Hotel Booking System implemented using the MERN (MongoDB, Express.js, React.js, Node.js) stack. The architecture is divided into three primary components:

Client Machine (Frontend):

This layer is built using React.js, which runs in the user's web browser. It provides an interactive and responsive user interface where customers can search for hotels, view room details, and make bookings. The application communicates with the backend server through HTTP requests (typically via RESTful APIs).

Back-End Server:

The server is built using Node.js with the Express.js framework. It acts as a middle layer that handles client requests, processes business logic, and communicates with the database. The server exposes API endpoints for various

IJIRSET©2025





www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405134|

operations such as user authentication, room listings, booking handling, and more.

Database (MongoDB):

MongoDB is used as the primary database to store all persistent data, including user details, hotel information, room availability, and booking records. The server uses the MongoDB driver to interact with the database for CRUD operations.

This architecture ensures a clear separation of concerns, promotes scalability, and enables efficient communication between the client and server, making it ideal for developing a robust and modern hotel booking system.

The system is built using a client-server architecture, enabling efficient interaction between users and the backend server. The client side is developed with React.js, ensuring a smooth and interactive user experience, while the backend comprises Node.js and Express.js managing APIs, logic, and database interactions.

Database Design

The database schema is designed using MongoDB collections:

- Users: Information such as name, email, role, and hashed password.
- Hotels: Metadata like hotel name, location, amenities, pricing, and images.
- Rooms: Linked with hotels, containing type, price, availability.
- Bookings: User ID, room ID, dates, and payment status.
- 3.3 Authentication and Security

Authentication is handled using JWT (JSON Web Tokens), ensuring secure sessions. Passwords are encrypted using bcrypt. Middleware in Express is ensures that only authorized users access specific routes.

RESTful API Endpoints

The backend provides the following APIs: /api/register: Register a new user. /api/login: Login user and return JWT. /api/hotels: Get, post, update, delete hotel data. /api/bookings: Manage room bookings.

Implementation:

Frontend:

Developed using React.js State management through Redux Styled using Tailwind CSS

Backend:

Developed using Express.js and Node.js APIs tested with Postman JWT for authentication

Database: Collections created in MongoDB Mongoose used for ODM

4.3 Testing

Unit Testing: For individual components and functions. Integration Testing: Ensured smooth interaction between frontend, backend, and database. User Testing: Conducted usability tests with selected users.





|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405134|

4.4 Deployment

The system was deployed on Heroku with MongoDB Atlas as a cloud database. CI/CD pipelines ensured smooth deployment and updates.

Role of JavaScript Frameworks in Web Applications

JavaScript has become a core technology in building responsive, high-performance web applications. The MERN stack—a combination of MongoDB, Express.js, React.js, and Node.js—has gained significant traction for full-stack development due to its unified language support, asynchronous capabilities, and real-time updates.

MongoDB supports flexible, schema-less design suitable for storing user profiles, booking details, and room metadata. React.js allows modular UI development, while Node.js ensures event-driven performance and Express.js facilitates secure, scalable REST APIs.

Advantages of MERN Stack in the Hospitality Sector

•Unified Development Environment*: JavaScript across frontend and backend reduces development complexity.

•Component-Based UI*: React enables reusable components and dynamic interfaces.

•Scalable Database*: MongoDB handles increasing user and booking data efficiently.

•Real-Time Features*: Node.js supports concurrent processing and real-time updates.

These advantages make MERN an ideal stack for building modern hotel management systems.

V. FLOWCHART



MERN Stack Development

Frontend Engineering

UI and UX Design

Using Tailwind CSS and responsive design principles, the UI adapts to devices of all sizes. Navigation was enhanced using React Router. Key UI components include:

- Hotel Cards
- Availability Calendar
- Booking Modal
- Analytics Graphs (via Chart.js)

State and Data Management

Redux was employed to manage global state. Local component states used hooks (useState, useEffect). API integration with Axios ensured smooth client-server communication.





|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405134|

5.3 Accessibility and Optimization

The UI meets WCAG guidelines for accessibility. Lazy loading and code splitting reduced initial load time.[8]

VI. BACKEND ENGINEERING

6.1 API Architecture

The system exposes RESTful endpoints for:

- User actions (/api/users)
- Hotel and room data (/api/hotels, /api/rooms)
- Booking and payments (/api/bookings)

6.2 Database Schema and Relationships

Collections:

- Users: Basic info, JWT token, role
- Hotels: Name, amenities, rating, location
- Rooms: Linked via hotel ID
- Bookings: With references to user and room

6.3 Security Considerations

- Authentication via JWT
- Passwords hashed using bcrypt
- Input validated with Joi
- Rate limiting and CORS enabled[4].

VII. FEATURES AND FUNCTIONALITY

7.1 User Features

- Signup and login with secure token storage
- Search hotels by location, price, rating
- Filter rooms based on type and availability
- Book rooms and make payments

7.2 Admin Features

- Add/update/delete hotels and rooms
- View customer bookings and revenue
- Analyze occupancy trends

7.3 Real-Time Availability Room inventory updates

Immediately upon booking or cancellation, ensuring accurate availability.

7.4 Analytics and Reporting Admins can view statistics such as:

- Total bookings
- Revenue generated
- Peak booking hours

VIII. RESULTS

Usability The system received positive feedback for its clean UI, simple workflows, and responsiveness across devices. **Performance** Performance testing with simulated loads showed consistent response times and minimal lag. The use of lazy loading and pagination improved frontend efficiency.





www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405134|

Security Implemented best practices: Encrypted passwords HTTPS with SSL Rate limiting to prevent DDoS Input validation with Joi

Scalability The use of MongoDB and stateless REST APIs ensures horizontal scalability. The system can handle increasing traffic with minimal adjustments.

The MERN-based Hotel Management System demonstrates how modern technologies can address legacy issues in the hospitality sector. The unified language stack improves developer productivity and reduces integration challenges.

Key takeaways:

- Real-time data improves customer experience
- Modular design allows for easy expansion
- Cloud deployment ensures accessibility and fault tolerance
- Challenges included:
- Managing state across components (solved using Redux)
- Ensuring consistent data updates (handled via MongoDB triggers and real-time APIs)

Results:



Fig. Hotel Admin Page

User-Friendly Interface:

- A modern, responsive UI using React.js.
- Easy navigation with sections for searching hotels, booking, and managing user profiles.
- Mobile-first design ensures usability across various devices.
- Real-Time Search and Filtering:
- Users can filter hotels by location, price, ratings, amenities, and availability.
- Dynamic results that update as filters are applied.
- Booking Functionality:
- A seamless booking process with a step-by-step workflow:
- Choose hotel and room type.
- Select check-in and check-out dates.
- Confirm guest details and make payments.
- Integrated payment gateway for secure transactions.
- User Authentication:
- Secure user sign-up and login using JWT (JSON Web Tokens).

Т

IJIRSET©2025





www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405134|

- Role-based access for users and administrators.
- Database Management:
- MongoDB stores:
- User information (e.g., profiles, booking history).
- Hotel details (e.g., descriptions, images, room types).
- Booking records for tracking and management.
- Real-time updates on room availability.
- Admin Dashboard:

Admins can:

- Add, update, or delete hotel details.
- Monitor bookings.
- View user analytics and revenue statistics.[12]



Fig. Booking Page

IX. CONCLUSION

The project successfully delivers a scalable, secure, and user-friendly hotel booking system using the MERN stack. By addressing traditional shortcomings and incorporating modern development principles, the platform sets a strong foundation for commercial deployment.

REFERENCES

[1] G. Bovet & J. Gornostaev, "Mastering React: A Comprehensive Guide to Modern Web Development," Packt Publishing, 2020.

[2] V. Kantor & R. Simpson, "Node.js Design Patterns: Unlocking Robust and Scalable Node.js Applications," O'Reilly Media, 2021.

[3] React.js Documentation, https://reactjs.org/docs

[4] Node.js Documentation, https://nodejs.org/en/docs

- [5] MongoDB Documentation, https://www.mongodb.com/docs
- [6] Express.js Guide, https://expressjs.com

[7] J. Smith & L. Brown, "A Comparative Study of Full-Stack Development Frameworks," Journal of Web Engineering, 18(3), 2020.

[8] R. Patel & S. Gupta, "The Role of MERN Stack in Scalable Web Application Development," International Journal of Computer Science Trends and Technology, 7(6), 2019.

[9] Traversy Media, "MERN Stack Crash Course for Beginners," https://www.traversymedia.com

| An ISO 9001:2008 Certified Journal |

12344





www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 5, May 2025

|DOI: 10.15680/IJIRSET.2025.1405134|

[10] FreeCodeCamp, "Building a Full-Stack Application with MERN," https://www.freecodecamp.org

 [11] Statista, "Digital Transformation in the Hospitality Industry: Trends and Insights," <u>https://www.statista.com</u>
[12] Attaluri, V., & Aragani, V. M. (2025). Sustainable Business Models: Role-Based Access Control (RBAC)
Enhancing Security and User Management. In Driving Business Success Through Eco-Friendly Strategies (pp. 341-356). IGI Global Scientific Publishing

[13] 3C, "Web Standards for Application Development," https://www.w3.org









INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY





www.ijirset.com