IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, December 2025
Impact Factor: 7.67

Full-Stack Development of a Global Stay Management System

Prof. Shweta Hedaoo[1], Miss. Pragati Jambhule[2], Miss. Ashwini Vaidya[3], Miss. Aditi Neware[4]
Guide, Computer Science and Engineering Department[1]
Student,Computer Science and Engineering Depaartment[2-4]
Tulsiramji Gaikwad-Patil College of Engineering and Technology, Nagpur, India

Abstract: *This paper presents the development of the "Global Stay Management System", a full-stack web application designed to replicate the core functionalities of a modern vacation rental platform. The system employs an architecture centered on Node.js, Express.js, and MongoDB (MERN elements), utilizing EJS templates for the frontend interface. The primary technical achievements include the implementation of secure user authentication using JSON Web Tokens (JWT) and bcrypt for password encryption. Furthermore, the system incorporates structured MongoDB schema management to facilitate efficient CRUD (Create, Read, Update, Delete) operations for property listings, alongside responsive design techniques (Flexbox, Grid, media queries) to ensure seamless user experience across all devices. The developed system provides a functional, secure, and intuitive platform for listing, browsing, and booking accommodations.*

Keywords: Full-Stack Development, Node.js, MongoDB, JWT Authentication, Responsive Design, EJS Templating

I. INTRODUCTION

The exponential growth of the travel and tourism industry has driven the reliance on digital platforms for accommodation booking, emphasizing convenience and personalization. The Global Stay Management System was developed as a full-stack web application to emulate the essential operational features of popular vacation rental platforms.

The objective of this project was to establish a functional system that enables users to securely create, manage, and view property listings, and facilitates the booking of stays through a responsive interface. The scope of the work focused on developing robust user authentication, efficient data management, and creating a user interface optimized for cross-device compatibility. The backend is powered by Node.js and Express.js, while data persistence is handled by a MongoDB database. The frontend is rendered using EJS templates, enhanced by Tailwind CSS and JavaScript for a clean, responsive design.

SYSTEM ARCHITECTURE

The project is built upon a client-server architecture, ensuring a clear separation of the user interface from the backend logic, which allows for independent scaling and maintenance.

A. Architectural Stack

The system utilizes a MERN-based architectural stack, with EJS serving as the templating engine for the presentation layer. The core components are:

Backend (Server-Side): Managed by Node.js and the Express.js framework, responsible for handling user requests, server-side operations, and listing management.

Database (Data Layer): MongoDB is employed as a NoSQL database, leveraging its document-based model for efficient storage and retrieval of dynamic data, such as user and listing profiles.

Frontend (Client-Side): Designed using EJS, HTML, CSS, and JavaScript to manage user interaction and display dynamic listing data.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 2, December 2025 Impact Factor: 7.67

B. Data Flow and Logic

The logic layer utilizes Express.js routes to manage request-response flows. Middleware is implemented for crucial security tasks, specifically route protection, which ensures that only authenticated and authorized users can access features like listing creation and booking.

IMPLEMENTATION

Implementation focused on secure user identity management, persistent data handling, and delivering an optimal user experience.

A. Secure Authentication and Authorization

User security was achieved through the implementation of a secure authentication process:

Password Encryption: The bcrypt library was used to hash and securely store user passwords, protecting credentials from unauthorized access.

Session Management: JSON Web Tokens (JWT) are utilized for session handling, providing a secure method for login and signup functionalities.

Access Control: Middleware checks are performed on server-side routes to enforce authorization and prevent unauthenticated access to sensitive resources.

B. Database Operations (CRUD)

Mongoose, an Object Data Modeling (ODM) library, was integrated to manage interactions with the MongoDB database. This allowed for structured schema design for users and listings, enabling reliable CRUD operations (Create, Read, Update, Delete) to manage property inventory dynamically.

C. Frontend and Responsive Design

The frontend was developed using dynamic EJS templates for pages like Home, Listings, and User Dashboards.

Form Validation: Client-side JavaScript validation was implemented to check for empty fields and valid email formats on forms, providing immediate user feedback and reducing invalid submissions to the server.

Responsiveness: CSS Flexbox, Grid systems, and media queries were extensively applied to ensure that the layout, components, and text scale seamlessly across desktop, tablet, and mobile resolutions.

RESULTS

The developed platform successfully delivered a functional and secure user experience, meeting all core project objectives.

A. Functional Accuracy

The system demonstrated high functional accuracy, particularly in its security and data handling layers:

Secure Access: The JWT and bcrypt-based authentication system successfully restricted access to privileged features, confirming the secure implementation of login/signup flows.

Data Integrity: Client-side validation prevents the submission of incomplete or incorrect data, while CRUD operations on the backend maintain data consistency across listings.

B. User Experience and Performance

The frontend implementation provided a positive user experience:

Seamless Navigation: The use of EJS templates and modular components resulted in smooth page transitions and optimized load times.

Adaptive Layout: The responsive design ensures that the layout adapts effectively to various screen sizes, enhancing accessibility and usability

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, December 2025
Impact Factor: 7.67

II. CONCLUSION

The development of the Global Stay Management System successfully showcased the integration of full-stack technologies—Node.js, Express.js, MongoDB, and EJS—to create a functional, secure, and responsive online accommodation booking platform. The project achieved its goals in secure authentication, efficient database management, and implementing an intuitive user interface. This effort provided valuable practical experience in modern web development methodologies, including security best practices, UI/UX principles, and maintaining code modularity. Future work can extend the platform by integrating external features such as third-party login (e.g., Google OAuth), map functionalities, and payment gateways.

REFERENCES

[1]. Mozilla Developer Network (MDN). HTML, CSS, and JavaScript documentation.
[2]. W3Schools. Tutorials and references for web development technologies.
[3]. Nielsen Norman Group. 10 Usability Heuristics for User Interface Design.
[4]. Google Developers. Responsive Web Design Basics.
[5]. WebAIM (Web Accessibility in Mind). Introduction to Web Accessibility.
[6]. Additional sources on Node.js, Express.js, MongoDB, and JWT/bcrypt as cited in the thesis's Literature Review/body chapters