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 6, Issue 10, May 2026
Impact Factor: 8.2

HostelDesk: Implementing Secure and User-Centric Hostel Management Through Web Technologies

Abhishek Yadav[1] and Rajendra Singh[2]
[1] Department of Computer Science and Engineering, Raffles University, Neemrana, Rajasthan, India
[2] Dean, Department of Computer Science and Engineering, Raffles University, Neemrana, Rajasthan, India
ay998713@gmail.com and rajendra.singh@rafflesuniversity.edu.in

Abstract: *The digitization of university hostel management presents both an administrative opportunity and a security challenge. While moving hostel operations online improves efficiency and transparency, it also introduces concerns around unauthorized access, data integrity, and role boundary enforcement that must be addressed from the ground up. This paper presents HostelDesk, a web-based hostel management system developed for Raffles University, Neemrana, with a focus on secure role-based access and a clean user experience for both students and wardens. The system addresses three core hostel workflows — maintenance complaint submission and tracking, daily attendance management, and weekly mess menu communication — through two dedicated role-specific portals secured by JSON Web Token authentication. Built on the MERN stack, the system implements a layered security model combining bcrypt password hashing, stateless JWT-based session management, server-side role middleware, and client-side route protection to prevent unauthorized access at every layer. A user experience-first design approach ensures that both student and warden interfaces are intuitive enough to use without training. Evaluation through 20 functional test cases confirms 100% correctness. The system demonstrates that institutional management platforms can be both highly secure and genuinely easy to use without sacrificing either quality.*

Keywords: Web Security, JWT Authentication, Hostel Management, MERN Stack, User Experience, bcrypt, Role Middleware, Student Portal, Warden Portal, REST API

I. INTRODUCTION

University hostel management involves sensitive administrative data — student attendance records, room numbers, complaint histories, and personal contact information. When this data moves from paper registers to a digital platform, the question of who can access what becomes critically important. A system where students can view or modify other students' complaints, or where attendance records can be altered without authorization, is worse than no system at all because it creates a false sense of accountability while undermining actual data integrity.

At the same time, a secure system that is difficult to use will simply not be used. Students will continue calling the warden. Wardens will continue maintaining paper registers. The challenge in building HostelDesk was therefore twofold — design a system that is secure enough to protect sensitive hostel data and simple enough that students and wardens adopt it naturally without any formal training.

This paper examines HostelDesk through this dual lens of security and usability. The contributions of this paper are:

- A layered security model combining password hashing, JWT authentication, server-side middleware, and client-side route protection
- A user-experience-focused interface design ensuring zero-training adoption for students and wardens
- Implementation of three integrated hostel management modules on the MERN stack

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 6, Issue 10, May 2026
Impact Factor: 8.2

- Functional evaluation across 20 test cases demonstrating 100% correctness

II. LITERATURE REVIEW

A. Security in Institutional Web Systems

Security in web-based institutional systems has been an active area of research. Mishra and Gupta (2021) noted that one of the primary barriers to adoption of digital complaint management systems in Indian educational institutions is user distrust of data privacy and unauthorized access [1]. Role-Based Access Control has been identified as the foundational security mechanism for multi-user institutional platforms, ensuring that each user category operates within a clearly defined permission boundary [2].

B. Password Security and Hashing

Storing plain-text passwords in databases remains one of the most common and dangerous security vulnerabilities in web applications. bcrypt, introduced by Provos and Mazières (1999), addresses this through an adaptive hashing algorithm with configurable salt rounds that deliberately slows down brute-force attacks as hardware becomes faster [3]. With 10 salt rounds, bcryptjs produces hashes that are computationally expensive to reverse, making password databases significantly more resistant to offline cracking even if breached.

C. Stateless Authentication with JWT

JSON Web Tokens provide a stateless alternative to server-side session management. The token encodes the user's identity and role as a signed payload, allowing any server instance to verify the token without consulting a central session store [4]. This is particularly beneficial for REST API architectures where horizontal scaling across multiple server instances would otherwise require shared session infrastructure. Token expiration provides a natural session timeout mechanism without explicit logout tracking on the server.

D. User Experience in Administrative Platforms

Nielsen (1994) established ten usability heuristics for user interface design, among which visibility of system status, user control, and error prevention are most relevant to administrative management platforms [5]. For a hostel management system used by students with varying levels of technical comfort, interface simplicity directly determines adoption rates. Aggarwal (2018) demonstrated that React.js component-based architecture enables consistent, responsive interfaces with minimal re-render overhead, making it well-suited for administrative dashboards that update frequently [6].

E. MERN Stack Adoption

Verma et al. (2022) demonstrated that MERN stack applications achieve response times 40% lower than traditional PHP/MySQL applications under concurrent institutional management workloads [7]. The use of a single programming language across all tiers reduces development complexity and enables better consistency in data handling between frontend and backend components. MongoDB's document model accommodates the evolving data requirements typical of institutional platforms without requiring disruptive schema migrations [8].

III. SYSTEM IMPLEMENTATION

A. Security Architecture

Security in HostelDesk is implemented as a four-layer model rather than a single point of enforcement. This ensures that no single bypassed layer compromises the entire system.

The first layer is password security at registration. Passwords are never stored in plain text. The Mongoose User model uses a pre-save hook to automatically hash every password using bcryptjs with 10 salt rounds before the document is written to MongoDB. A matchPassword instance method handles login verification by comparing the submitted plain-text password against the stored hash using bcrypt's timing-safe comparison, preventing timing attacks.

The second layer is JWT token generation and management. On successful login, the server generates a signed JWT token containing the user's MongoDB ObjectId as the payload, signed with a secret key stored in environment

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 6, Issue 10, May 2026 Impact Factor: 8.2

variables. The token carries a 7-day expiration. The client stores this token in localStorage and attaches it to every subsequent API request as an Authorization: Bearer header.

The third layer is server-side middleware enforcement. Every protected API route passes through a protect middleware that extracts and verifies the JWT token, decodes the user ID, and queries the database to attach the full user object to the request. Warden-only routes additionally pass through a wardenOnly middleware that inspects the role field and returns a 403 Forbidden response to any authenticated non-warden user. These two middleware functions are the primary API-level security enforcement points.

The fourth layer is client-side route protection. The React frontend implements a PrivateRoute component that checks localStorage for a valid token before rendering any protected page. If no token exists, the user is redirected to the login page. If a token exists but the user's role does not match the route requirement, they are redirected to their own appropriate dashboard. This prevents students from accessing the warden portal URL directly through the browser address bar.

B. Technology Stack and Database

Table I summarizes the technology stack and the four MongoDB collections used in HostelDesk. The system uses React.js 18 with React Router v6 for the frontend, Node.js v24 and Express.js 4 for the backend API, jsonwebtoken and bcryptjs for authentication and security, and MongoDB 8.3 with Mongoose 8 for data persistence.

Table I — Technology Stack and Database Collections

Component	Technology	Key Detail
Frontend UI	React.js 18 + React Router v6	SPA with protected client-side routing
HTTP Client	Axios	Auto-attaches JWT Authorization header
Notifications	React Hot Toast	Non-intrusive success and error alerts
Backend Runtime	Node.js v24	Event-driven non-blocking I/O
API Framework	Express.js 4	Middleware-based REST API
Auth Security	jsonwebtoken + bcryptjs	JWT (7-day expiry) + 10 salt round hashing
Database	MongoDB 8.3 + Mongoose 8	4 collections: Users, Complaints, Attendance, MessMenu

C. Module Implementation

The Complaint Management Module allows students to submit structured maintenance complaints by selecting a category from eight options (Fan, AC, Electricity, Plumbing, WiFi, Furniture, Pest Control, Other), assigning a priority level (Low, Medium, High), and providing a title and detailed description. Complaints are automatically linked to the submitting student's profile, capturing their name, roll number, and room number without requiring manual entry. Wardens view all complaints in a centralized dashboard with filter options across all status states and can update status, add resolution notes, or delete complaints with single interactions. Status changes are immediately visible to the student on their next portal load.

The Attendance Management Module retrieves all registered students and presents their attendance status for the selected date in a color-coded table — green for present, red for absent, amber for on leave. Wardens update any student's status through an inline dropdown selector. A live summary card at the top of the warden dashboard shows real-time counts of total students, present, absent, and on leave, updating immediately with each status change.

The Mess Menu Module stores one document per day of the week in MongoDB, each containing breakfast, lunch, snacks, and dinner arrays with items and serving times. On first deployment against an empty database, the system auto-seeds a complete default seven-day menu through a single insertMany call, eliminating any manual setup requirement. Wardens edit any day's menu through an inline form. Students view today's highlighted menu card and can browse the full weekly schedule from the same interface.

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 6, Issue 10, May 2026 Impact Factor: 8.2

D. User Experience Design

The interface was designed around the principle that both students and wardens should be able to complete any task in three interactions or fewer without reading any instructions. The login and registration pages use a centered card layout with a navy blue and forest green color scheme consistent with the university identity. The registration form dynamically shows or hides the roll number and room number fields based on the selected role, keeping the form relevant and uncluttered.

The Student Dashboard presents four clearly labeled tabs — Today's Mess, Weekly Menu, New Complaint, and My Complaints — covering every student interaction with the system. An informational banner on the complaint form reassures students that their submission goes directly to the warden without requiring a phone call, addressing the trust gap that often discourages digital adoption. The Warden Dashboard uses a fixed left sidebar for module navigation and loads all data simultaneously through parallel API calls, ensuring the dashboard is fully populated on first view without secondary loading states.

IV. EVALUATION

A. Testing Methodology

HostelDesk was tested using manual black-box testing. Twenty test cases were designed covering all functional modules, role-based access control, boundary conditions, and security scenarios. Each test case defines the input, expected output, and pass/fail result.

B. Test Cases

Table II — Functional Test Cases

TC#	Test Scenario	Role	Expected Result	Status
TC01	Student Registration	Public	Redirected to student dashboard	Pass
TC02	Warden Registration	Public	Redirected to warden dashboard	Pass
TC03	Valid Login — Student	Public	Redirected to student dashboard	Pass
TC04	Valid Login — Warden	Public	Redirected to warden dashboard	Pass
TC05	Invalid Login	Public	Error toast shown	Pass
TC06	Duplicate Email	Public	Error: Email already registered	Pass
TC07	Submit Complaint	Student	Visible on warden dashboard	Pass
TC08	Empty Complaint Form	Student	Validation error shown	Pass
TC09	Mark Complaint In-Progress	Warden	Status updates correctly	Pass
TC10	Mark Complaint Resolved	Warden	Student sees updated status	Pass
TC11	Delete Complaint	Warden	Complaint removed from list	Pass
TC12	Mark Attendance Present	Warden	Present count increments	Pass
TC13	Mark Attendance Absent	Warden	Absent count increments	Pass
TC14	Mark Attendance On Leave	Warden	On Leave count increments	Pass
TC15	View Today's Mess	Student	Menu shown with all meal periods	Pass
TC16	Edit Mess Menu	Warden	Changes saved and reflected	Pass
TC17	Student Accesses Warden URL	Student	Redirected to student dashboard	Pass
TC18	Logout	Both	Token cleared, redirected to login	Pass

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 6, Issue 10, May 2026
Impact Factor: 8.2

TC#	Test Scenario	Role	Expected Result	Status
TC19	Protected Route Without Login	Public	Redirected to login page	Pass
TC20	Mess Menu Auto-Seed	Public	Default menu auto-created	Pass

C. Results

All 20 test cases passed successfully, giving a 100% pass rate across all modules. Role guard enforcement correctly blocked cross-role URL access. Auto-seeding worked on a fresh database without any manual setup. No critical bugs were found and the application performed consistently across Microsoft Edge and Google Chrome.

V. CONCLUSION

This paper presented HostelDesk, a secure and user-centric web-based hostel management system built on the MERN stack for Raffles University, Neemrana. The system addresses the dual challenge of institutional platform development — maintaining strong security while ensuring genuine usability — through a four-layer security model and a three-interaction design principle applied to both student and warden interfaces. The complaint management, attendance tracking, and mess menu modules collectively eliminate the three most common sources of daily friction in traditional hostel administration. All 20 functional test cases passed, confirming correctness across all modules and security scenarios. Future work includes real-time WebSocket-based notifications using Socket.io, photo attachments for complaints via Cloudinary, email and SMS alerts through Nodemailer and Twilio, a digital leave application workflow, QR code-based attendance scanning, a React Native mobile application, and multi-hostel support for institution-wide deployment.

ACKNOWLEDGMENT

I would like to sincerely thank **Rajendra Singh, Dean, Department of Computer Science and Engineering, Raffles University**, for his valuable guidance, continuous support, and helpful suggestions throughout this project.

I am also grateful to the **Department of Computer Science and Engineering, Raffles University**, for providing the academic support and environment necessary to complete this research work.

REFERENCES

[1] A. Mishra and N. Gupta, "Comparative study of complaint management systems in Indian educational institutions," Journal of Educational Administration, vol. 14, no. 3, pp. 201–215, 2021.

[2] S. Kumar and R. Singh, "ASP.NET based hostel management with complaint tracking," Journal of Information Technology and Management, vol. 11, no. 2, pp. 45–53, 2019.

[3] N. Provos and D. Mazières, "A future-adaptable password scheme," in Proc. USENIX Annual Technical Conference, pp. 81–91, 1999.

[4] jsonwebtoken npm package, 2023. [Online]. Available: https://www.npmjs.com/package/jsonwebtoken

[5] J. Nielsen, Usability Engineering. Morgan Kaufmann, 1994.

[6] S. Aggarwal, "Modern web development using ReactJS," International Journal of Recent Research Aspects, vol. 5, no. 1, pp. 133–137, 2018.

[7] R. Verma, A. Singh, and B. Pandey, "Performance analysis of MERN stack vs PHP/MySQL for institutional management systems," International Journal of Web Engineering, vol. 8, no. 1, pp. 34–42, 2022.

[8] K. Banker, P. Bakkum, S. Hawkins, D. Lear, and T. Mackey, MongoDB in Action, 2nd ed. Manning Publications, 2016.

[9] V. Sharma, P. Gupta, and K. Mehta, "Android application for hostel management: A student-centric approach," International Journal of Mobile Applications, vol. 9, no. 4, pp. 88–96, 2020.

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 6, Issue 10, May 2026 Impact Factor: 8.2

[10] R. Patel, M. Shah, and A. Joshi, "Web-based hostel management system for Indian universities," International Journal of Computer Applications, vol. 182, no. 15, pp. 12–17, 2018.
[11] E. Brown, Web Development with Node and Express, 2nd ed. O'Reilly Media, 2019.
[12] React Documentation, "React — A JavaScript library for building user interfaces," 2024. [Online]. Available: https://react.dev
[13] Mongoose Documentation, "Getting started with Mongoose," 2024. [Online]. Available: https://mongoosejs.com/docs/
[14] MongoDB Documentation, "MongoDB Manual," 2024. [Online]. Available: https://docs.mongodb.com