

HostelHQ-An Intelligent Hostel Management System

Anushka Kamble , Suyash Kadam, Devika Patil, Sanjana Kusalkar

Anushka Kamble, Computer & Zeal college of Engineering

Suyash Kadam, Computer & Zeal college of Engineering

Devika Patil, Computer & Zeal college of Engineering

Sanjana Kusalkar , Computer & Zeal college of Engineering

Abstract - Efficient hostel management is very important for educational institutions because more students are needing accommodation. Traditional methods, which depend on manual tasks, often lead to delays, errors, and trouble in keeping records, assigning rooms, tracking fees, and handling complaints. These issues cause student dissatisfaction and make the work of wardens and administrators harder. HostelHQ solves these problems by offering a strong, web-based hostel management system built with the Django framework and SQLite database. The system uses role-based access control, so students, wardens, and managers can use the platform in ways that fit their responsibilities. One of the main features of HostelHQ is an AI-powered chatbot that helps students with common questions and allows them to report complaints quickly. The chatbot also has an escalation process to make sure unresolved issues are passed on to wardens or managers in a timely manner. Features such as tracking fee status, attendance, and room allocation have been included to improve transparency and efficiency. By combining automation with smart support, HostelHQ reduces administrative work, lowers human mistakes, and improves communication among all users. This paper presents the method, design, and architecture of HostelHQ, showing it as a scalable and flexible solution that can support future enhancements like IoT integration, predictive analytics, and biometric authentication.

However, many hostels still rely on old manual methods for daily operations, leading to scattered data, slow communication, and poor transparency. Existing systems have tried to fix some of these problems by introducing digital tools for admissions, fee collection, or room allocation [1][2]. Yet, they often lack proper integration, scalability, or smart features that can meet the needs of students who are used to digital tools. These shortcomings create gaps in user experience and operational efficiency, making administrators overwhelmed and students unhappy. HostelHQ was created to address these challenges. The system is designed to offer a complete, digital hostel management platform where all stakeholders can interact through a single interface. Students can access personalized dashboards to check their accommodation status, report issues, and track their fee status. Wardens are given tools to monitor attendance, assign rooms, and handle complaints. Managers oversee the overall performance, ensuring smooth coordination between wardens and students. By dividing the system into three distinct roles, HostelHQ ensures both efficiency and security through role-based access control..

2. Functional And Non Functional Requirements Of HostelHQ

2.1 Functional Features

- User Registration:** The system will allow new users (students, managers, administrators) to register with unique email addresses and strong passwords.

User Login: The system will give a secure login option for all registered users, verifying their credentials against the database.

1.INTRODUCTION

The need for student housing is increasing because higher education institutions are expanding around the world.

3. METHODOLOGY

- **Role-Based Access Control (RBAC):**

The system will enforce RBAC, ensuring that users can only access functions and data relevant to their assigned role (Student, Manager, Administrator).

- **Traffic Hotspot Detection:** The platform performs clustering analysis (e.g., DBSCAN) on location data to highlight areas with high accident or congestion frequency.

- **Password Management:** The system will allow users to securely reset forgotten passwords and change existing ones.

- **Session Management:** The system will manage user sessions securely, including automatic logout after inactivity.

2.2 Non-Functional Features

- **Response Time:** The system will respond to 90% of user requests within 3 seconds under peak load conditions (e.g., 500 concurrent users)

- **Data Retrieval:** Complex data queries (e.g., generating monthly reports) will complete within 10 seconds.

- **Data Encryption:** All sensitive data, including user credentials, personal information, and financial details, will be encrypted both during transit (using HTTPS/SSL/TLS) and when stored (database encryption).

- **Access Control:** The RBAC mechanism will be strong, preventing unauthorized access to data and functionalities across all user roles.

The development of HostelHQ followed the Agile Scrum methodology, an iterative and incremental approach ideal for web application creation.

Agile was chosen because it allows for flexibility and continuous feedback, which is needed given the various user roles and features. The methodology ensured that the project was completed in manageable sprints, with each focusing on specific modules such as authentication, fee management, chatbot integration, and reporting. Regular sprint reviews and retrospectives helped the team refine features based on user testing and feedback.

The requirement analysis phase was a key starting point. It involved identifying functional requirements like user authentication, online payments, complaint handling, and role-based dashboards. Non-functional requirements, such as scalability, security, and performance, were also defined to ensure the system remains robust under heavy load. A lot of attention was given to security, with measures like password hashing, role-based access control, and encryption of sensitive data included in the design.

Once requirements were finalized, the system design phase used UML diagrams, entity-relationship models, and data flow diagrams to visualize processes and database interactions.

Django was selected as the backend framework because of its modularity and built-in support for secure authentication and database management. SQLite was initially chosen as the database for its lightweight nature, although the architecture was made scalable to allow future upgrades to more advanced databases. The frontend was created using HTML, CSS, and JavaScript to ensure compatibility across different platforms.

Finally, testing was done using both functional and non-functional test cases. Unit testing checked the functionality of individual modules, while integration testing ensured that

modules worked well together. The chatbot was tested with various user queries to assess response accuracy and escalation procedures. Performance testing confirmed the system could handle multiple users at once, and security testing was carried out to address risks like SQL injection and XSS attacks.

4. SYSTEM ARCHITECTURE AND DASHBOARD

The HostelHQ is built on a three-tier architecture:

- Frontend:** Developed using HTML,CSS,JS for interactive web.
- Backend:** Built on Django Python Framework for handling APIs, rule-based logic, and notifications.
- Database:** SQLite for storing all data in it .



5. CONCLUSION

HostelHQ is a major step forward in hostel management by replacing inefficient manual systems with a digital, automated, and smart solution.

The system brings together multiple hostel functions—room allocation, fee management, attendance tracking, and complaint resolution—into one platform, offering a smooth experience for students, wardens, and managers. Its AI chatbot, a key innovation, introduces intelligent automation for real-time support and issue escalation, reducing administrative workload and improving response times.

ACKNOWLEDGEMENT

The authors express their sincere gratitude to Prof. Dhanashri R. Londhe for her consistent guidance and valuable insights. The team also extends appreciation to the Department of Computer Engineering, Zeal College of Engineering & Research, Pune, for providing infrastructure and technical support throughout this project.

REFERENCES

- [1] P. Harina, K. K. Kavya, R. Sharmikha Sree, and S. Meera, “Hostel Management,” in 2022 1st Int. Conf. on Computational Science and Technology (ICCST), 2022. doi: 10.1109/ICCST55948.2022.10040481.
- [2] T. C. Chua and S. A. Mubin, “APResidence: Development of Online Student Accommodation Management System for Asia Pacific University,” in 2023 3rd Int. Conf. on Mobile Networks and Wireless Communications (ICMNWC), 2023. doi: 10.1109/ICMNWC60182.2023.10435913.
- [3] S. K. Yadav, U. Tyagi, and N. Singh, “Smart Hostel Booking System for Efficient Room Allocation and Management,” in 2025 3rd Int. Conf. on Communication, Security, and Artificial Intelligence (ICCSAI), 2025. doi: 10.1109/ICCSA164074.2025.11064697.
- [4] K. Chaudhri and R. Kevat, “Study of Digitalized Hostel Management System,” Int. J. of Scientific Research in

Computer Science, Engineering and IT, vol. 7, no. 2, pp. 366–371, 2021. doi: 10.32628/CSEIT217280.

[5] A. M. Diyaolu et al., “Development of an E-Based Hostel Management System,” Int. J. of Innovative Science and Research Technology, vol. 9, no. 6, pp. 1000–1009, 2024. doi: 10.38124/ijisrt/IJISRT24JUN147.

[6] Eweoya et al., “Development of Web-Based Hostel Management System,” British Journal of Computer, Networking and IT, vol. 8, no. 1, pp. 30–41, 2025.

[7] R. Heda et al., “DHMS: A Digital Hostel Management System Integrating Campus ChatBot, Predictive Intelligence, and Real-Time Automation,” arXiv, 2025. [Online]. Available: <https://arxiv.org/abs/2507.17759>.

[8] G. L. Borhade et al., “Smart Energy Management and Overload Control of Hostel Management using IOT,” Int. J. of Advanced Research in Science, Communication and Technology, vol. 3, no. 4, pp. 168–174, 2023. doi: 10.48175/568.

[9] V. R. Sugumaran et al., “SMART HOSTEL AUTHENTICATION SYSTEM USING ANDROID,” Int. Research Journal of Modernization in Engineering Technology and Science, vol. 6, no. 4, pp. 78–83, 2024.

[10] S. Bhardwaj et al., “Hybrid Technology Based Smart Hostel Management System Using AI and IoT,” in 2022 Fourth Int. Conf. on Emerging Research in Electronics, Computer Science and Technology (ICERECT), 2022. doi: 10.1109/ICERECT56837.2022.10059715.