Staysmart: A Web-Based Hostel Management System for Student Accommodation

Norzihani Yusof[1*], Nur Hani Ayesha Saiful Nizan[1], Nuzulha Khilwani Ibrahim[1], Zuraini Othman[1], Ahmad Syahrin Yusof[2]

[1]Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka (UTeM), 76100 Malacca, Malaysia

[2]Jabatan Keselamatan dan Kesihatan Pekerjaan Selangor, Tingkat 7 & 8, Wisma Consplant 2, Jalan SS 16/1, 47500 Subang Jaya, Selangor, Malaysia

DOI: https://dx.doi.org/10.47772/IJRISS.2026.100300027

Received: 07 March 2026; Accepted: 12 March 2026; Published: 24 March 2026

ABSTRACT

Hostel management in educational institutions often relies on manual or semi-digital processes, leading to inefficiency, data inconsistency, and delays in room allocation and complaint handling. This paper presents StaySmart, a web-based hostel management system developed using PHP, MySQL, HTML, CSS, and JavaScript under a WAMP server environment. The system adopts a three-tier client-server architecture with role-based access control (RBAC) for two user types: students and administrators. Students can submit hostel applications, view room assignment results, report maintenance complaints, and receive in-system notifications. Administrators manage applications, allocate rooms, monitor occupancy across five hostel blocks totalling 2,700 rooms, and resolve complaints through a centralized dashboard. The system incorporates AI-assisted logic using keyword-based classification to automatically categorize maintenance complaints into five issue types and assigns urgency levels (Urgent, Normal, Low) accordingly. System evaluation was conducted through unit testing, integration testing, system testing, and usability assessment with six participants. Testing confirmed accurate room allocation, correct complaint categorization with 93.3% category accuracy and 86.7% priority accuracy, real-time data synchronization, and intuitive user navigation with a 95% task completion rate. The findings demonstrate that StaySmart effectively reduces administrative workload, improves transparency, and enhances the overall efficiency of hostel management operations.

Keywords: Hostel Management System, Web-Based System, Student Accommodation, Keyword-Based Complaint Classification, Role-Based Access Control, AI-Assisted Categorization

INTRODUCTION

Hostel management is a critical component of student life in educational institutions. Providing safe and comfortable accommodation while supporting efficient administrative processes is essential for institutional effectiveness. However, many universities continue to rely on manual or semi-digital processes for student hostel applications, room allocation, and maintenance complaint handling. These approaches typically involve paper forms, spreadsheets, or standalone digital tools such as Google Forms, which lead to inefficiencies, data inconsistencies, and delays in processing [1][2].

The increasing number of students in higher education institutions has amplified these limitations. Manual data entry results in inconsistent records, while the absence of centralized tracking causes delays in room assignment and unresolved maintenance complaints. Students often lack real-time visibility into their application status, and administrators face growing workloads managing records across disparate systems [3][4].

To address these challenges, this paper presents StaySmart, a web-based hostel management system developed for Universiti Teknikal Malaysia Melaka (UTeM). The system centralizes hostel operations on a single platform accessible via any standard web browser. Key features include online hostel application submission, automated room allocation, maintenance complaint reporting with AI-assisted categorization, real-time room occupancy tracking, in-system notifications, and a comprehensive administrative dashboard. The system was implemented using PHP, MySQL, HTML, CSS, and JavaScript in a WAMP server environment, adopting a three-tier client-server architecture with RBAC for students and administrators [5][6].

The scope covers hostel application management, room allocation, AI-assisted complaint categorization, and administrative record management for five hostel blocks at UTeM encompassing 2,700 rooms. The remainder of this paper is organized as follows: Section II reviews related work, Section III describes the system architecture and methodology, Section IV presents experimental results, Section V discusses findings, and Section VI concludes the paper.

Related Work

Research on web-based hostel and facility management systems has grown steadily, driven by the need to modernize administrative processes in educational institutions. This section reviews key prior works that inform the design and evaluation of StaySmart.

Wong and Mahdin [11] developed an online hostel management system that digitizes student room applications and administrative assignment processes. Their system eliminated paper-based workflows and centralized student records, demonstrating improvements in administrative efficiency. However, the system lacked AI-assisted complaint categorization, real-time room availability tracking, and integrated dashboard analytics.

Azeeta et al. [3] proposed an intelligent student hostel allocation system using web technologies, highlighting the importance of automating room assignment based on predefined constraints such as gender-specific blocks and capacity limits. Their system did not address maintenance complaint management.

Diyaolu et al. [6] developed an e-based hostel management system that digitized complaint submission processes, demonstrating reduced processing times compared to manual methods. However, the system required administrators to manually sort and prioritize issues without automated classification.

In the domain of AI-assisted classification, Revina et al. [10] investigated keyword-based ticket classification in IT service management and found that simpler keyword-matching methods achieved comparable accuracy to more complex machine learning models for structured complaint categories. Al-Hawari and Barham [1] applied machine learning to IT helpdesk systems and reported significant improvements in response efficiency when complaints were automatically categorized.

Bouabdallaoui et al. [4] demonstrated the value of automated fault detection in building facility management, showing improved maintenance response times. Mohamed et al. [8] recommended RBAC as the most suitable model for web-based administrative systems where distinct user types require differentiated feature access, informing the dual-role architecture of StaySmart.

Table I. Comparison of Hostel Management Systems

Feature	**Manual**	**Wong & Mahdin [11]**	**Azeeta et al. [3]**	**StaySmart**
Online Application	✗	✓	✓	✓
Room Allocation	Manual	✓	✓	✓
Complaint Submission	Form/Call	✓	✗	✓
AI Categorization	✗	✗	✗	✓
Real-Time Room Status	✗	✗	✗	✓

Dual Roles (RBAC)	✗	✓	✓	✓
Integrated Dashboard	✗	Limited	✗	✓
In-System Notifications	✗	✗	✗	✓

As shown in Table I, StaySmart addresses all identified gaps by providing a complete, integrated solution combining online applications, automated room allocation, AI-assisted complaint categorization, real-time occupancy tracking, and comprehensive dashboard analytics.

SYSTEM ARCHITECTURE AND METHODOLOGY

The development of StaySmart followed the Structured Systems Analysis and Design Method (SSADM), which provides a systematic framework comprising feasibility study, requirements analysis, logical system specification, physical design, implementation, and testing phases [11].

A. System Architecture

StaySmart adopts a three-tier client-server web architecture: (1) presentation layer using HTML, CSS, and JavaScript; (2) application logic layer using PHP; and (3) data layer using MySQL. The system is hosted on a WAMP server and accessible via standard web browsers. RBAC differentiates system access between students (authenticated via matric number and password) and administrators (authenticated via institutional email credentials) [8].

B. System Modules

The system comprises four primary modules as illustrated in the Data Flow Diagram (DFD Level 0) in Figure 1.

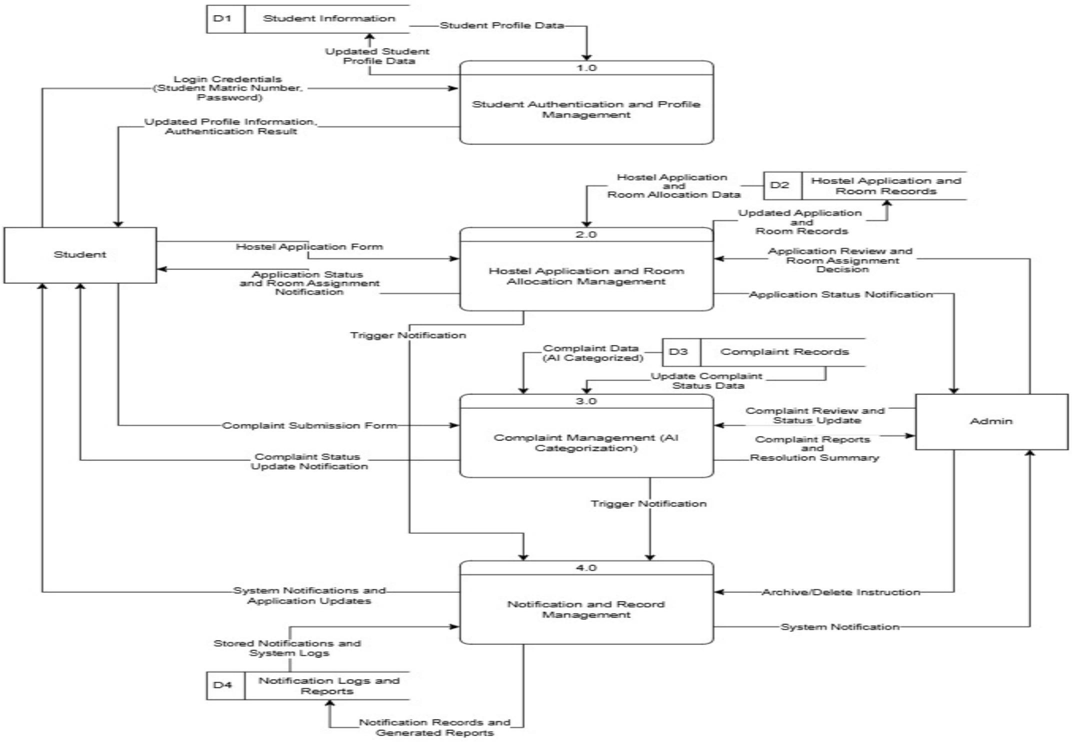
D1
Student Information
Student Profile Data
Updated Student Profile Data
1.0
Student Authentication and Profile Management
Login Credentials (Student Matric Number, Password)
Updated Profile Information, Authentication Result
Hostel Application and Room Allocation Data
D2
Hostel Application and Room Records
Updated Application and Room Records
Student
Hostel Application Form
Application Status and Room Assignment Notification
2.0
Hostel Application and Room Allocation Management
Application Review and Room Assignment Decision
Application Status Notification
Trigger Notification
Complaint Data (AI Categorized)
D3
Complaint Records
Update Complaint Status Data
3.0
Complaint Management (AI Categorization)
Complaint Submission Form
Complaint Status Update Notification
Complaint Review and Status Update
Complaint Reports and Resolution Summary
Admin
Trigger Notification
4.0
Notification and Record Management
System Notifications and Application Updates
Archive/Delete Instruction
System Notification
Stored Notifications and System Logs
D4
Notification Logs and Reports
Notification Records and Generated Reports

Fig. 1. StaySmart Data Flow Diagram (DFD Level 0)

Module 1 handles user login, session management, and profile updates. Module 2 validates room availability against gender-specific block constraints before confirming assignments. Module 3 processes maintenance complaints, applies keyword-based classification, and enables administrators to update complaint statuses. Module 4 generates in-system notifications triggered by application status changes and complaint updates and supports archiving of semester data.

C. Database Design

The StaySmart database comprises seven relational tables in MySQL. Figure 2 presents the Entity Relationship Diagram (ERD) and Table II summarizes the database tables.

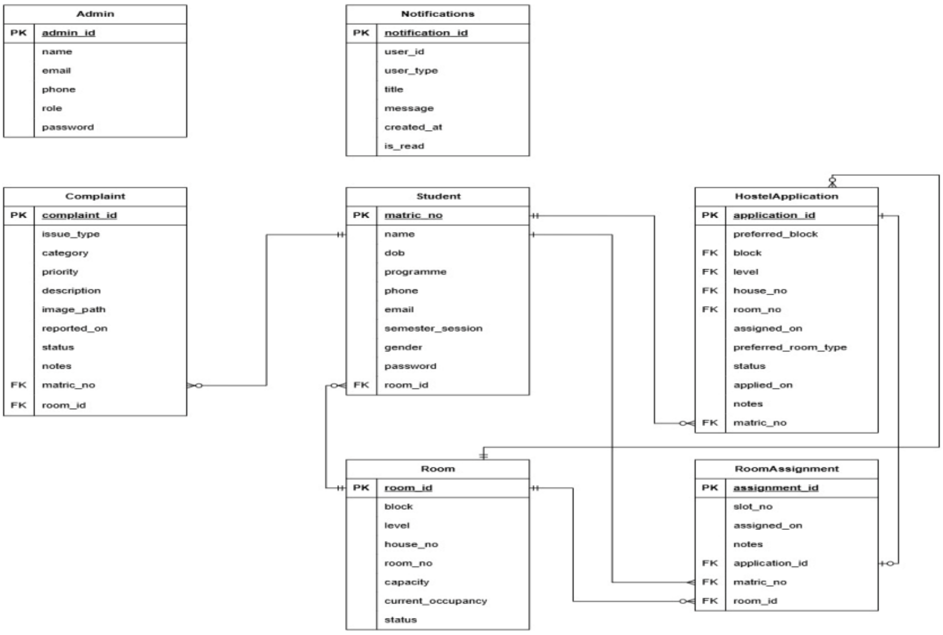

Fig. 2. Entity Relationship Diagram (ERD) of StaySmart

Table Ii. Database Tables and Descriptions

Table	PK	Function
Student	matric_no	Student profiles and login
Admin	admin_id	Admin accounts and roles
Room	room_id	2,700 rooms across 5 blocks
HostelApplication	application_id	Hostel application records
RoomAssignment	assignment_id	Room-student assignments
Complaint	complaint_id	Complaints + AI category/priority
Notification	notification_id	In-system notifications

D. AI-Assisted Complaint Classification

The AI-assisted classification module employs a keyword-based approach validated by Revina et al. [10] for structured service request categorization. This approach was selected for its interpretability, low computational overhead, and proven effectiveness [1][10].

The classification process: (1) accepts issue type from dropdown; (2) extracts complaint description text; (3) converts text to lowercase and tokenizes; (4) matches tokens against keyword dictionary; (5) assigns subcategory and priority based on first matched rule; (6) defaults to ‘Others’ category with LOW priority if no keywords match. Table III presents the classification rules.

Table Iii. Ai-Assisted Complaint Classification Rules

Issue Type	**Trigger Keywords (Examples)**	**Priority**
Electrical	blackout, short circuit, sparks, no power	URGENT
Electrical	flickering, bulb, socket, switch	NORMAL
Plumbing	flood, burst pipe, no water, sewage	URGENT
Plumbing	dripping, slow drain, tap, shower	NORMAL
Furniture/Equip.	broken, collapsed, damaged, unsafe	URGENT
Furniture/Equip.	loose, stuck, squeaky, worn	NORMAL
Hygiene/Cleaning	cockroach, rat, pest, mold	URGENT
Hygiene/Cleaning	dirty, uncleaned, trash, smell	NORMAL
Others	No keyword match	LOW

E. Development Environment

The system was developed using Visual Studio Code with PHP 7.4 as the server-side language, MySQL 5.7 for database management, and WAMP server for the local environment. HTML5, CSS3, and JavaScript provided the front-end. Open-source tools were selected to minimize development cost [6].

ANALYSIS AND RESULTS

This section presents results from system functional testing, AI classification accuracy evaluation, usability assessment, and system interface overview.

A. System Functional Testing

The system underwent unit testing, integration testing, and system testing using sample data representing realistic hostel operations. Table IV summarizes testing outcomes.

Table Iv. System Testing Outcomes

Test Case	**Result**
Student login – valid credentials	Pass
Student login – invalid credentials	Pass
Hostel application submission	Pass
Room allocation (gender-specific block)	Pass
Overbooking prevention (room → Full)	Pass

Complaint submission with AI classification	Pass
Admin updates complaint + notifies student	Pass
In-system notification on app approval	Pass
Admin approve/reject application	Pass
Admin views real-time occupancy dashboard	Pass

All ten test cases passed successfully. Room allocation correctly enforced gender-specific block constraints and prevented overbooking. Integration testing confirmed real-time data synchronization across modules and the database.

B. AI Classification Accuracy

The keyword-based classification was evaluated using 15 sample complaints across all five issue types, compared against ground truth labels from two independent reviewers. Table V presents the results.

Table V. Ai Classification Evaluation Results

Issue Type	**n**	**Cat. Acc.**	**Pri. Acc.**
Electrical	3	100%	100%
Plumbing	3	100%	67%
Furniture/Equip.	3	100%	100%
Hygiene/Cleaning	3	67%	67%
Others	3	100%	100%
Overall	15	93.3%	86.7%

Overall category accuracy was 93.3% and priority accuracy was 86.7%. The one misclassified category involved a Hygiene complaint using non-standard vocabulary. Two priority misassignments occurred for ambiguous descriptions. These findings are consistent with Revina et al. [10], who demonstrated high accuracy for structured complaint vocabulary using keyword-based methods.

C. Usability Assessment

Usability testing was conducted with six participants (four students, two administrators) performing predefined tasks. Table VI presents task completion results.

Table Vi. Usability Testing Task Completion

Task	**Users**	**Completion**
Student login & profile viewing	4 students	100%
Hostel application submission	4 students	100%
Complaint submission	4 students	75%

Admin room assignment	2 admins	100%
Admin complaint status update	2 admins	100%
Overall	6 participants	95%

Overall task completion rate was 95%. One student required guidance during complaint submission due to uncertainty about the issue type dropdown. Qualitative feedback indicated a clean interface and intuitive navigation. Participants highlighted real-time application status and automatic complaint categorization as particularly useful features [9].

D. System Interface

Figure 3 presents the administrator dashboard providing an overview of total students, room occupancy, pending applications, and active complaints, with gender distribution, top academic programmes, and 7-day trend charts.

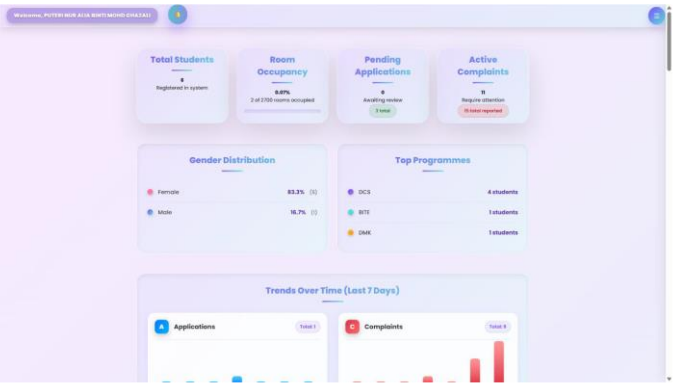

Fig. 3. Admin Das hboard – Overview Panel

Figure 4 presents the AI-Powered Complaints management interface, enabling administrators to filter and prioritize maintenance responses by category and urgency level.

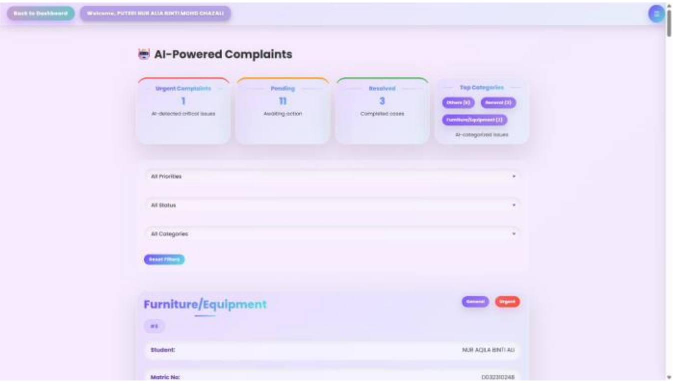

Fig. 4. AI-Powered Complaint Management Interface

DISCUSSION

The results demonstrate that StaySmart successfully addresses the limitations of existing manual and semi-digital hostel management approaches. The centralized web-based platform eliminated paper-based workflows and provided both students and administrators with real-time access to hostel information, consistent with findings from comparable systems [11][3].

The AI-assisted complaint classification achieved 93.3% category accuracy and 86.7% priority accuracy using a keyword-based approach, consistent with Revina et al. [10]. The interpretable nature of classification decisions is important in administrative contexts where administrators need to understand and trust automated prioritization [1].

The primary limitation is dependence on predefined vocabulary. Complaints using non-standard terms or mixed-language descriptions may be incorrectly classified. Future enhancement using machine learning models trained on larger datasets of real complaint data could address this limitation [1][10].

The 95% usability task completion rate indicates an accessible and intuitive interface. The one task requiring guidance suggests that clearer labelling for the issue type dropdown could improve user experience, aligned with usability principles for web-based administrative systems [9].

The system was evaluated using simulated data rather than in a live production environment. Deployment in a live university setting and longitudinal evaluation would provide stronger validation under real-world conditions including peak concurrent usage and the full diversity of actual student complaint descriptions.

CONCLUSION

This paper presented StaySmart, a web-based hostel management system designed to centralize and digitize hostel operations for educational institutions. The system adopted a three-tier client-server architecture with role-based access control and was implemented using PHP, MySQL, and standard web technologies. The AI-assisted keyword-based complaint classification module achieved 93.3% category accuracy and 86.7% priority accuracy across 15 evaluated complaints, demonstrating the viability of the approach for structured maintenance complaint categorization.

System testing confirmed successful room allocation with gender-specific constraint enforcement, real-time data synchronization, and correct role-based access control. Usability assessment with six participants yielded a 95% task completion rate. The findings support that StaySmart effectively reduces administrative workload, improves transparency, and enhances operational efficiency in hostel management compared to manual approaches.

Future work should prioritize deployment in a live university environment to validate system performance under real operational conditions. The AI module should be extended with machine learning models trained on real complaint data for improved robustness. Additional planned enhancements include a mobile application, real-time push notifications to external devices, an online payment module for hostel fees, two-factor authentication, and multi-campus support.

ACKNOWLEDGEMENT

The authors would like to thank the Centre of Research and Innovation Management (CRIM) of Universiti Teknikal Malaysia Melaka (UTeM) for sponsoring the publication fees under the Tabung Penerbitan CRIM UTeM.

REFERENCES

1. Al-Hawari, F., & Barham, H. (2021). A machine learning based help desk system for IT service management. Journal of King Saud University – Computer and Information Sciences, 33(6), 702–718. https://doi.org/10.1016/j.jksuci.2019.04.001
2. Almaiah, M. A., Al-Khasawneh, A., & Althunibat, A. (2020). Exploring the critical challenges and factors influencing the e-learning system usage during COVID-19 pandemic. Education and Information Technologies, 25, 5261–5280. https://doi.org/10.1007/s10639-020-10219-y
3. Azeeta, A., Misra, S., Odusami, M., Peter, O. U., & Ahuja, R. (2021). An intelligent student hostel allocation system based on web applications. Lecture Notes in Electrical Engineering, 701, 779–791. https://doi.org/10.1007/978-981-15-8297-4_62
4. Bouabdallaoui, Y., Lafhaj, Z., Yim, P., Ducoulombier, L., & Bennadji, B. (2021). Predictive maintenance in building facilities: A machine learning-based approach. Sensors, 21(4), 1044. https://doi.org/10.3390/s21041044

5. Kaur, M., & Bansal, S. (2022). Real-time database systems for resource management applications. International Journal of Information Management, 63, Article 102434.
6. Diyaolu, A. M., Abodunrin, O. B., Adedamola, A. A., Ogunode, R. S., & Omoloba, O. (2024). Development of an e-based hostel management system. International Journal of Innovative Science and Research Technology, 9(6). https://doi.org/10.38124/ijisrt/IJISRT24JUN147
7. Hosamo, H. H., Svennevig, P. R., Svidt, K., Han, D., & Nielsen, H. K. (2022). A digital twin predictive maintenance framework of air handling units. Energy and Buildings, 261, 111988. https://doi.org/10.1016/j.enbuild.2022.111988
8. Mohamed, A. K. Y. S., Auer, D., Hofer, D., & Küng, J. (2022). A systematic literature review for authorization and access control. International Journal of Web Information Systems, 18(2–3), 156–180. https://doi.org/10.1108/IJWIS-04-2022-0077
9. Qashou, A. (2021). Influencing factors in M-learning adoption in higher education. Education and Information Technologies, 26(2), 1755–1785. https://doi.org/10.1007/s10639-020-10323-z
10. Revina, A., Buza, K., & Meister, V. G. (2020). IT ticket classification: The simpler, the better. IEEE Access, 8, 193380–193395. https://doi.org/10.1109/ACCESS.2020.3032840
11. Wong, P. L., & Mahdin, H. (2022). Design and development of online hostel management system to improve application and management process. Applied Information Technology and Computer Science, 3(2), 356–369.
12. Yadav, A., & Vishwakarma, D. K. (2020). Sentiment analysis using deep learning architectures: A review. Artificial Intelligence Review, 53(6), 4335–4385. https://doi.org/10.1007/s10462-019-09794-5
13. Zhang, J., & Tao, D. (2021). Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things. IEEE Internet of Things Journal, 8(10), 7789–7817. https://doi.org/10.1109/JIOT.2020.3039359
14. Ahmad, A. F., & Noor, N. H. (2024). User acceptance of web-based administrative systems in higher education. Journal of ICT in Education, 11(1), 1–10.
15. Lim, S. Y., & Wong, K. T. (2024). Centralized database systems for administrative efficiency. Journal of Information Systems Research, 9(2), 61–69.
16. Kumar, P., Patel, D., & Shah, S. (2023). Design of role-based access control in web applications. Journal of Web Engineering, 22(1), 89–98.