Global Journal of Engineering and Technology Advances
eISSN: 2582-5003
Cross Ref DOI: 10.30574/gjeta
Journal homepage: https://gjeta.com/

(RESEARCH ARTICLE)

Check for updates

Design and Implementation of a Hostel Allocation System

Adebanjo, Adedoyin S [1], Josiah Solomon Kelechi [2], Olawunmi Asake Adebanjo [1,*], Awoniyi, Amos [1] and Adewuyi Oluwaseyi J [2]

[1] *Department of Software Engineering, Babcock University, Ilishan-Remo, Ogun State, Nigeria.*
[2] *Department of Computer Science, Babcock University, Ilishan-Remo, Ogun State, Nigeria.*

Global Journal of Engineering and Technology Advances, 2026, 28(02), 193–207

Publication history: Received on 20 June 2026; revised on 26 July 2026; accepted on 28 July 2026

Article DOI: https://doi.org/10.30574/gjeta.2026.28.2.0171

Abstract

The Hostel Management System (HMS) is a computerized solution developed to streamline and automate the management of hostel operations in educational institutions. Traditional hostel management methods rely heavily on manual processes, which are often inefficient, time-consuming, and prone to errors such as data loss, misallocation of rooms, and poor record-keeping. This project focuses on the design and implementation of a centralized system that manages key hostel activities, including student registration, room allocation, fee management, visitor records, and maintenance requests. The system provides a secure and organized database for storing student and hostel information, ensuring easy access and retrieval when needed. The proposed Hostel Management System aims to improve administrative efficiency, reduce human errors, and enhance transparency in hostel operations. It also enables better monitoring of occupancy and resource utilization while providing a user-friendly interface for both administrators and students. Overall, the system offers a reliable, scalable, and effective solution for modern hostel management in academic environments.

Keywords: Hostel Management; Student Accommodation; Web Application; Room Allocation; Bed Assignment; Analytics Dashboard

1. Introduction

The efficient management and allocation of hostel accommodations remain a critical challenge in many higher educational institutions worldwide. Traditionally, hostel assignment processes have relied heavily on manual procedures, which are often resource-intensive, time-consuming, and susceptible to biases and inaccuracies [1]. Efficient hostel management involves the allocation of rooms and beds, monitoring occupancy status, maintaining accurate student records, and providing timely administrative oversight.

With the advancement of Information and Communication Technology (ICT), universities are increasingly adopting web-based management systems to automate core administrative processes. Such systems enable centralized data storage, real-time access to information, improved transparency, and enhanced decision-making. Babcock University operates multiple hostels distributed across different organizations and branches, making manual tracking and reporting increasingly complex as the student population grows. This study therefore focuses on the development of a Hostel Allocation System that automates hostel operations and provides analytical insights to support informed administrative decisions.

* Corresponding author: Olawunmi Asake Adebanjo.

1.1. Statement of the Problem

Babcock University's hostel management is largely manual and lacks a fully integrated digital platform. Hostel allocation, bed occupancy tracking, and student registration are often handled using paper records or basic spreadsheets, introducing several challenges: delays in hostel assignment, difficulties in tracking occupied and vacant beds, inconsistencies in records, and limited visibility into hostel utilization. Manual record-keeping also poses risks related to data inaccuracies, unauthorized access, and difficulty in managing student preference data on a large scale [2]. The absence of analytical tools makes it difficult for administrators to assess hostel usage patterns and evaluate resource utilization, necessitating a robust web-based hostel allocation system.

The aim of the study is to design and develop a comprehensive web-based Hostel Allocation System with analytical capabilities for efficient hostel allocation, monitoring, and administration at Babcock University. The specific objectives are:

- To analyze existing hostel allocation procedures and identify their limitations.
- To design and develop a centralized system that automates hostel allocation and bed assignment, manages hostels, floors, rooms, beds, and student records, while providing real-time tracking and analytical dashboards for hostel occupancy, bed utilization, and user distribution.
- To evaluate the system's effectiveness in improving efficiency, accuracy, and transparency.

2. Literature Review

2.1. Conceptual Framework

Hostel accommodation plays a pivotal role in student welfare and effective campus management. Well-administered hostel facilities enhance students' quality of life, foster community engagement, and simplify administrative processes [3]. Traditional hostel room allocation has been managed through manual methods such as direct assignment by staff, first-come-first-served policies, and lottery-based selections. These conventional approaches are limited in their ability to maintain accurate records and handle growing student populations efficiently [4]. Several persistent problems afflict traditional hostel management systems: they are inefficient and time-consuming, vulnerable to human bias, and prone to data inconsistency [5].

Web-based hostel management systems are typically built on multi-tier architectures consisting of a client-facing interface, a server-side application layer, and a relational database for persistent storage. This architecture supports multi-user access, centralized data control, and secure role-based interactions for both administrators and students, as well as the integration of analytics modules that generate visual reports on occupancy and bed utilization.

2.2. Review of Related Works

Several web-based hostel management systems have been developed and studied in the literature. Khamis et al. [6] developed a PHP and MySQL-based hostel management portal that enabled real-time room allocation and improved record management. While effective for basic operations, the system had limited security features and did not include analytics for administrative reporting. Elakkiya et al. [7] similarly developed a PHP-based system with enhanced security features for student records, though scalability remained a concern for larger institutions.

Aswar et al. [8] built a hostel management system using .NET technology and SQL Server, providing a user-friendly graphical interface that addressed several shortcomings of manual systems. In the Nigerian context, hostel management systems have often been developed as institutional projects addressing specific administrative needs, but most rely heavily on manual data entry and lack automation, real-time updates, or analytics dashboards [1]. The present study addresses these gaps by delivering a fully integrated, role-aware, and analytically enriched system tailored to Babcock University.

2.3. Comparative Analysis of Existing Solutions

The comparative analysis of existing systems reveals that while most successfully automate basic hostel operations such as room booking and student registration, few integrate real-time occupancy tracking with analytics dashboards. Systems developed in the Nigerian context tend to lack the web accessibility and data visualization features found in more advanced international implementations. Additionally, most reviewed systems do not enforce hierarchical data structures that separately manage hostels, floors, rooms, and beds, which limits the granularity of occupancy tracking. The proposed system directly addresses all of these gaps.

3. Methodology

3.1. Software Development Lifecycle Model

The system was developed using the Agile Software Development Lifecycle model. Agile emphasizes flexibility, iterative development, and continuous improvement, making it suitable for evolving system requirements. Features such as gender restriction, disability constraints, and audit logging were added incrementally. Agile allowed flexibility, continuous stakeholder feedback, and rapid improvement through the following phases:

- Requirement Analysis — understanding existing hostel management processes and user needs.
- System Design — defining architecture, database models, and UML diagrams.
- Implementation — developing system modules incrementally in iterative sprints.
- Testing — continuous testing during and after each development sprint.
- Deployment — running the system in a live institutional environment.
- Review and Improvement — refining features based on user feedback.

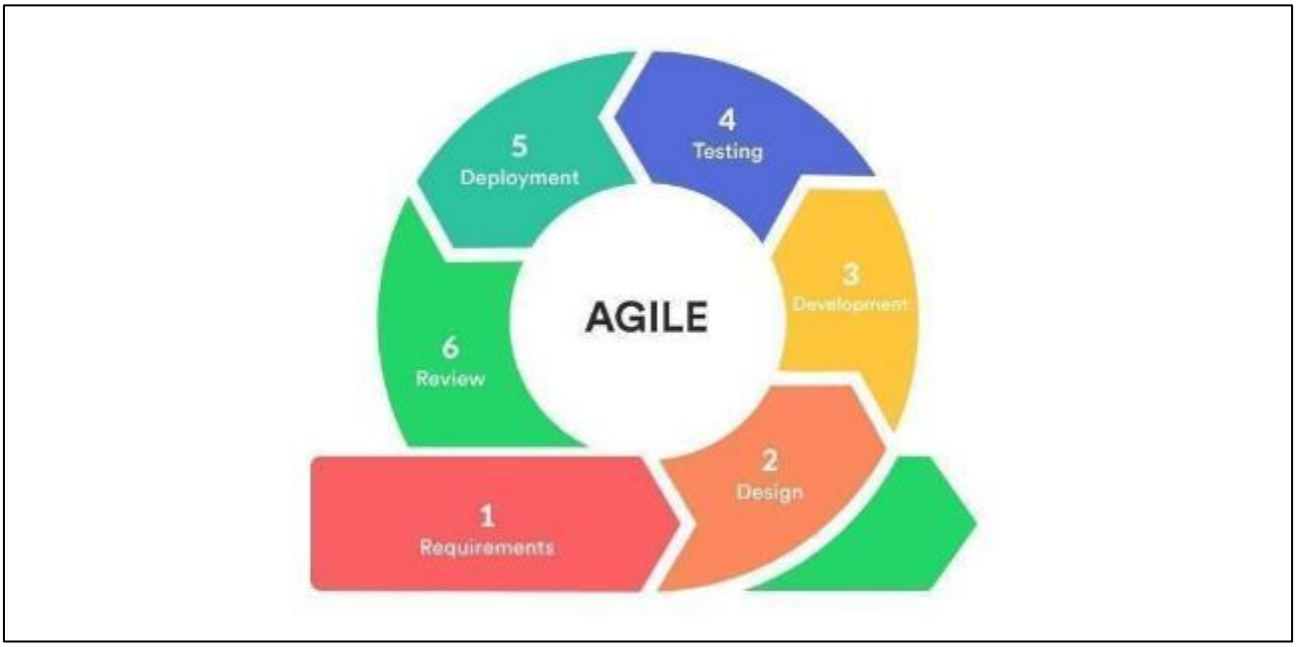

Figure 1: Agile SDLC [9]

3.2. System Requirements

3.2.1. Hardware Requirements

The system is designed to run on standard hardware to ensure affordability and ease of deployment: Processor — Intel Core i3 or higher; RAM — minimum 2 GB; Disk Storage — at least 2 GB free; Display — 1024×768 resolution or higher; Network — stable internet connection for client-server communication.

3.2.2. Software Requirements

The system relies on widely adopted open-source technologies: Operating System — Windows, Linux, or macOS; Programming Language — Python 3.11; Web Framework — Django 4.2.6; Database — SQLite (development and testing); Frontend — HTML5, CSS3, JavaScript; Visualization — Plotly; Data Processing — Pandas; Development Environment — Visual Studio Code.

3.2.3. Functional Requirements

- Hostel Management: administrators can create, update, and manage hostel records including gender classification and category restrictions.
- Structural Configuration: administrators define hierarchical structures of floors, rooms, and beds within each hostel.

- Bed Occupancy Monitoring: the system automatically tracks and updates occupancy status of each bed in real time.
- Student Registration Management: the system supports registration of students with unique identifiers including matriculation number, name, course, gender, and special needs status.
- Hostel Allocation Process: students can apply for allocation by selecting their preferred hostel, floor, room, and bed through a guided interface.
- Validation and Constraint Enforcement: the system prevents occupied bed selection, gender mismatches, and duplicate registrations.
- Analytics and Reporting: dashboards display occupancy statistics and analytical charts to support administrative decision-making.

3.2.4. Non-Functional Requirements

- Performance: fast response times for all user interactions, including hostel allocation, data retrieval, and dashboard visualization.
- Reliability: data consistency, integrity, and accuracy at all times with mechanisms to prevent data corruption or duplication.
- Security: access to administrative functionalities restricted through authentication and authorization; sensitive data protected against unauthorized access.
- Scalability: capable of handling increasing numbers of users, hostels, rooms, and allocation records without significant performance degradation.
- Maintainability: modular, well-documented, and extensible codebase to facilitate future updates and feature enhancements.

3.3. System Design

3.3.1. Use Case Diagram

The Use Case Diagram illustrates the interaction between the system and its two primary actors: Administrator and Student. Student use cases include: register in the system, select a hostel and bed, change room, and view personal profile and allocation status. Administrator use cases include: create and manage hostel structures; approve or reject room change requests; override student allocations; and monitor hostel occupancy and analytics.

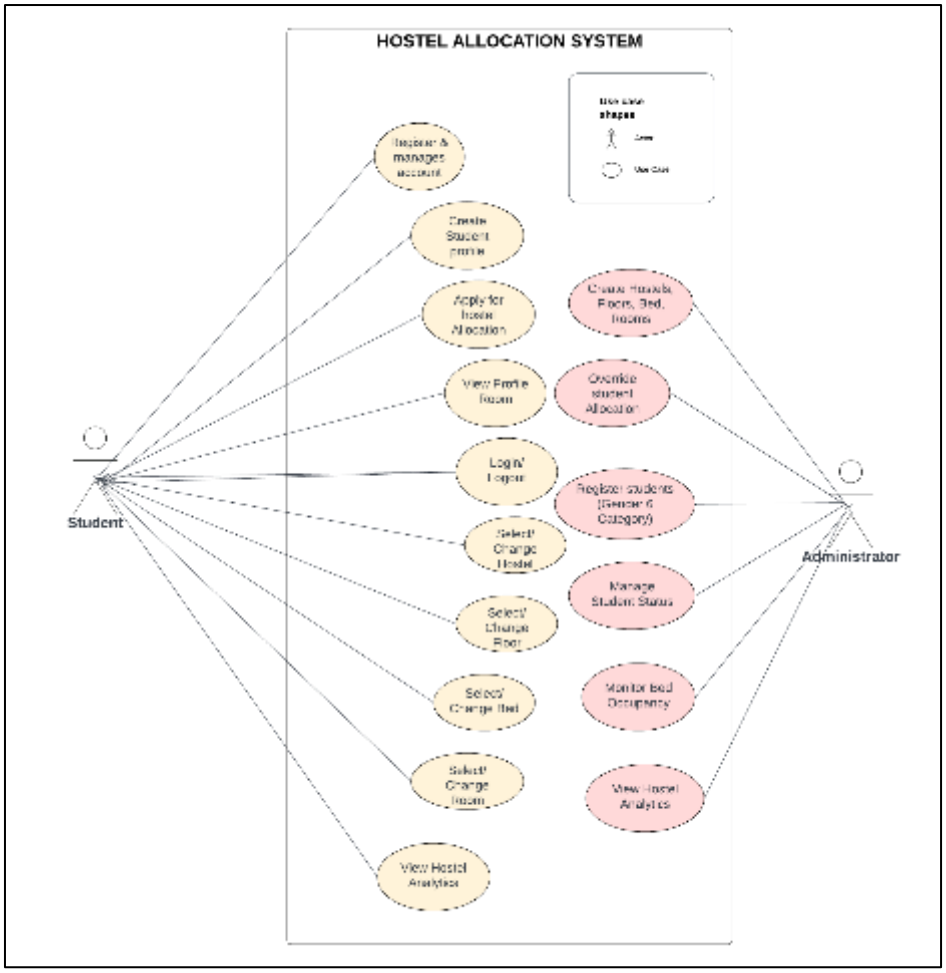

Figure 2: Use Case Diagram

3.3.2. Activity Diagram

The Activity Diagram represents the workflow of the hostel allocation process. The process follows a logical validation sequence to ensure fairness and compliance with defined constraints, enforcing integrity and transparency throughout the allocation process.

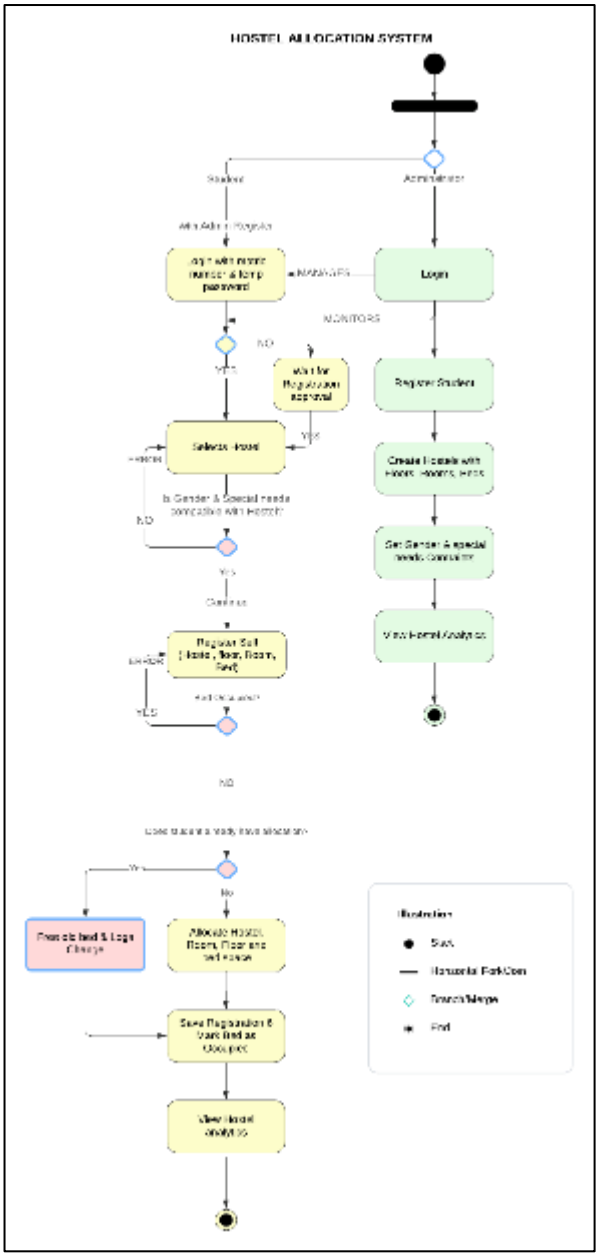

Figure 3: Activity Diagram

3.3.3. Flowchart Diagram

The Flowchart Diagram illustrates the decision-making logic of the allocation module. The system evaluates the following sequential conditions: Is the selected bed available? Does the student's gender match the hostel's gender classification? Does the student meet the hostel category requirements? Do disability or accessibility constraints apply? Does the student already have an existing allocation? If an existing allocation is detected, the system updates the record and logs the modification.

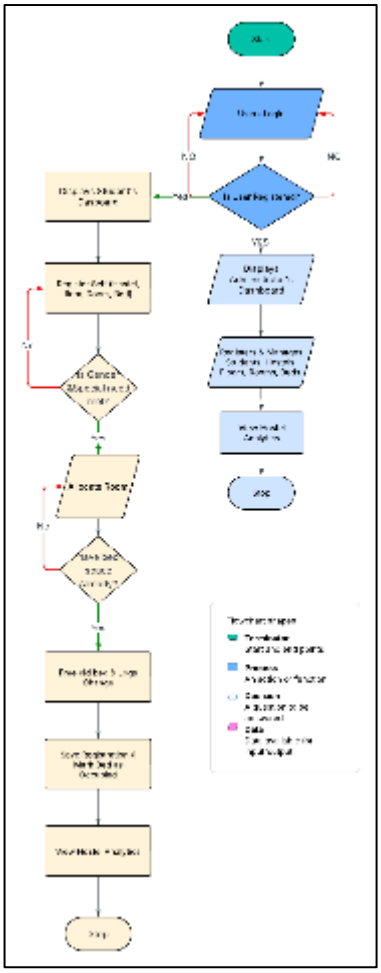

Figure 4: Flowchart Diagram

3.3.4. System Architecture

The system adopts a three-tier architecture ensuring separation of concerns and improved maintainability. The Presentation Layer is implemented using HTML, CSS, and Django template rendering. The Application Layer is developed using Django views and business logic components responsible for request handling, validation, constraint enforcement, and allocation processing. The Data Layer is managed through a relational database integrated via Django ORM, ensuring structured storage, data integrity, and efficient querying.

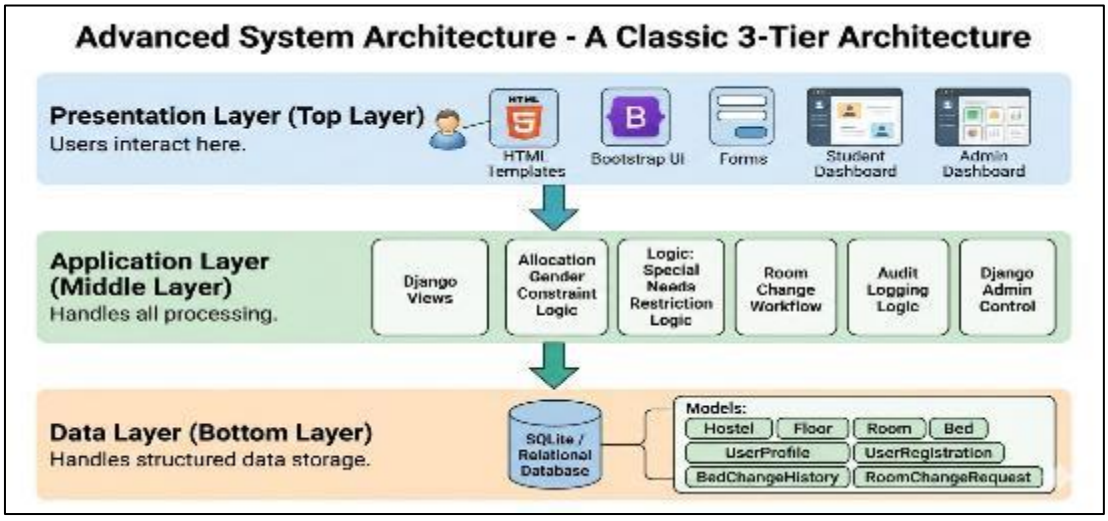

Figure 5: System Architecture Diagram

4. Implementation and Testing

4.1. User Authentication and Access Control

User authentication was implemented using Django's built-in authentication framework. The system distinguishes between two primary user roles: Student Users and Administrator Users (Staff Privileges). Access control mechanisms were enforced using login-required decorators, role-based conditional checks, and staff-only view restrictions. Administrative features such as hostel creation and override allocation are accessible only to users with staff privileges.

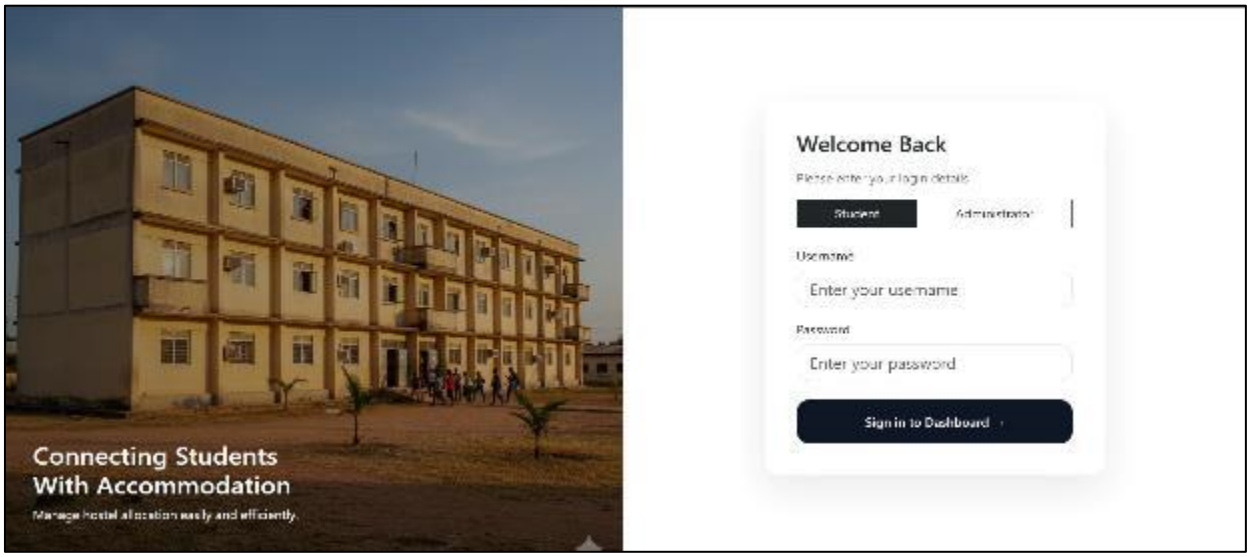

Figure 6: Role-Based Login Modules

4.2. Hostel Structure Management Module

Administrators can create and manage hostels including floors, rooms, and beds. Each hostel is defined with attributes such as gender classification (Male, Female, Mixed) and category restrictions (Undergraduate, Postgraduate, Staff, General). The system implements a relational hierarchy where a Hostel contains multiple Floors, each Floor contains multiple Rooms, and each Room contains multiple Beds. When a bed is allocated, its status is automatically updated; if deallocated, it reverts to available status, ensuring real-time data accuracy.

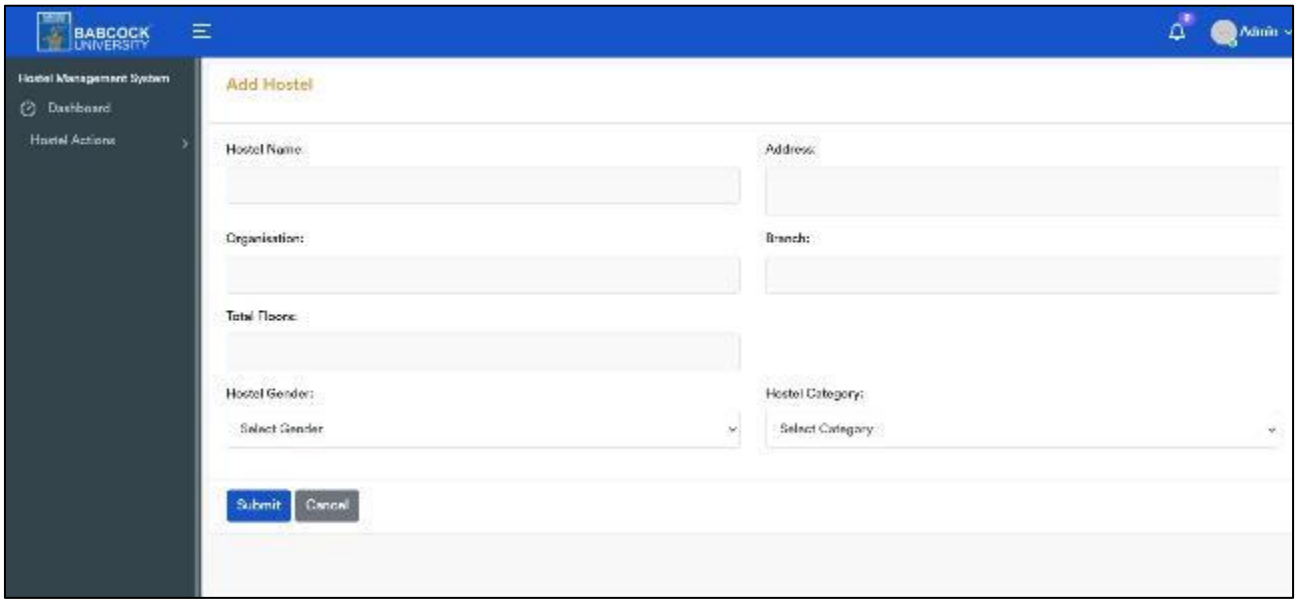

Figure 7: Add Hostel Page

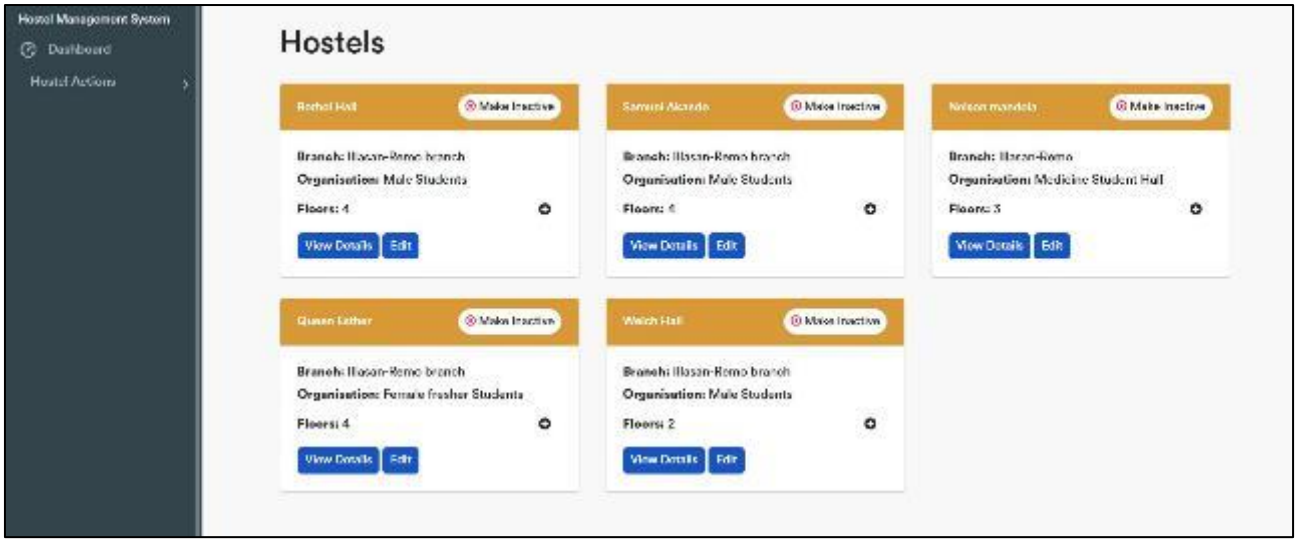

Figure 8: View Hostel Page

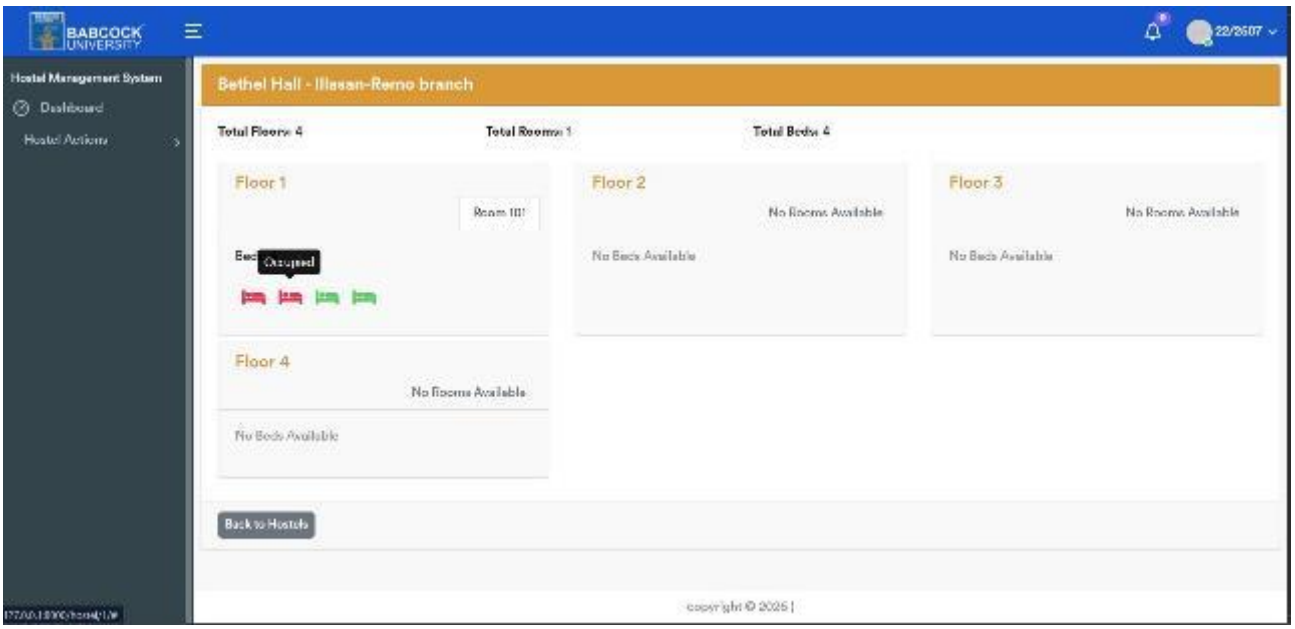

Figure 9: View Bethel Hall Hostel Page

4.3. Student Profile and Registration Module

Administrators can register students directly into the system, particularly for special cases or administrative overrides. The system validates gender, category, and accessibility constraints during registration to prevent invalid allocations. Each student profile contains personal and academic information: matriculation number, full name, course of study, gender, category (level), and special needs or disability status. If a student already has an allocation, the system frees the previous bed, updates the allocation record, and logs the change in the history table.

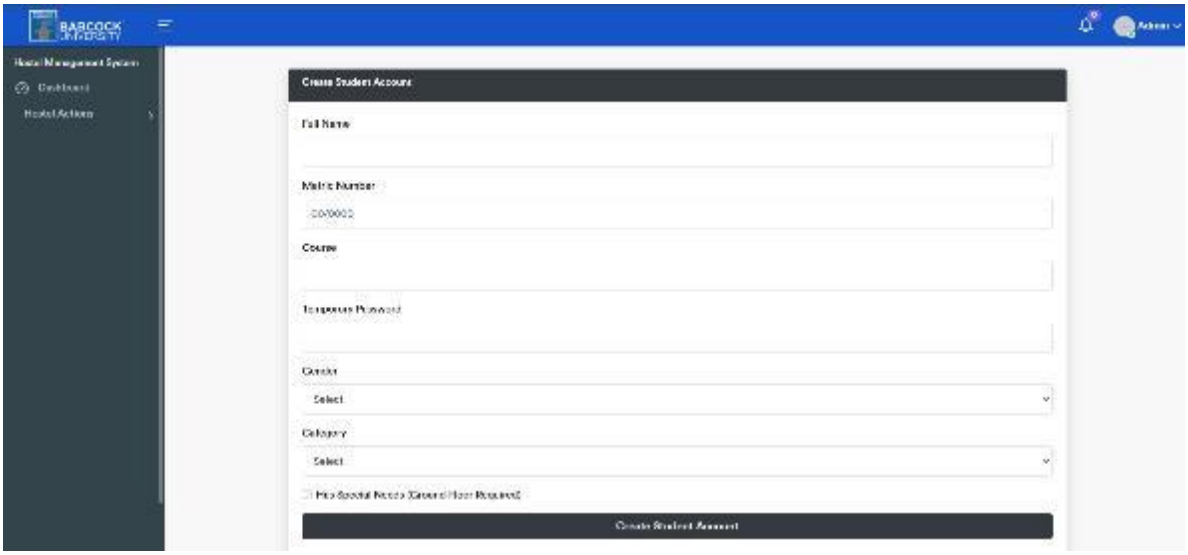

Figure 10: Admin Register Student Page

4.4. Student Dashboard and Allocation

Students interact with the system primarily for registration, allocation, and profile management. The dashboard provides a personalized overview of the student's current room assignment, upcoming payment due dates, and pending notifications. The system enforces constraints to ensure fair and accurate bed assignments throughout the allocation process.

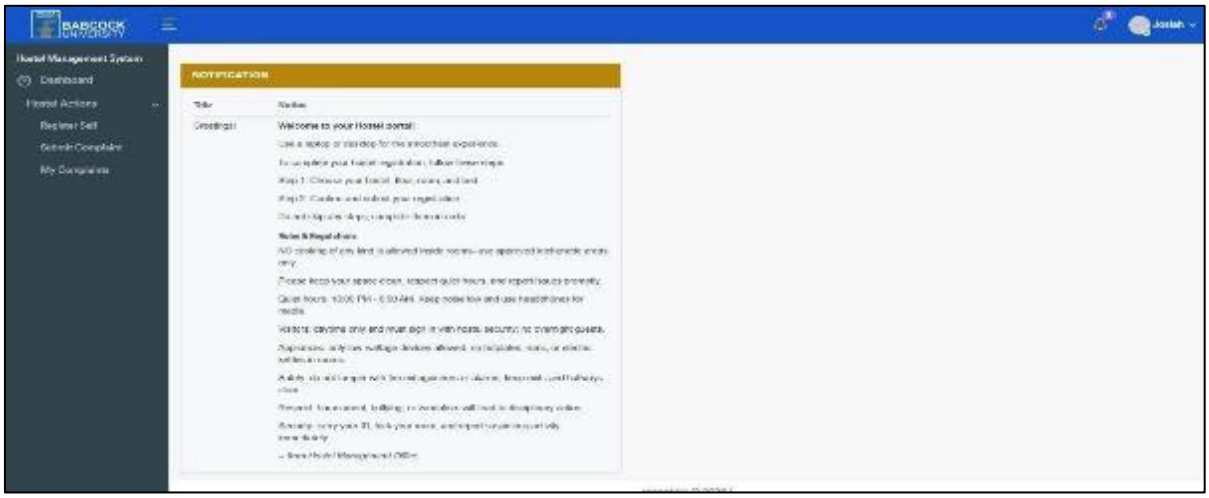

Figure 11: Student Dashboard

The registration interface guides students through the sequential selection of a hostel, floor, room, and bed. The system performs validation checks on bed availability, gender compatibility, category eligibility, and accessibility constraints. Duplicate allocations are prevented, and any existing allocation is automatically updated if a student changes their bed.

Figure 12: Student Registration and Bed Selection Page

4.5. Student Profile Page

The profile page allows students to view their personal details, allocated hostel, room, and bed. This interface also includes an option to change their password securely.

Figure 13: Student Profile Page

4.6. Hostel Analytics Module

The analytics module presents visual insights into hostel utilization. Graphs and charts display occupied versus available beds, allocations by gender, and distribution across student categories. These tools aid administrators in decision-making and capacity planning. The analytics module processes stored hostel and occupancy data to generate charts such as hostel-wise bed utilization, room occupancy status, and user distribution by organization and branch.

Figure 14: Hostel Analytics Dashboard

4.7. System Testing and Validation

The system was tested using a structured unit and functional testing approach to validate core modules including authentication, hostel management, allocation logic, and analytics. A total of eight (8) test cases were executed, covering both backend logic and user interface behaviour. All test cases passed successfully, achieving a 100% success rate with no failures recorded.

Table 1: System Integration & Functional Test Results

Item	Actual Result	Notes
python manage.py check	System check identified no issues (0 silenced)	Confirms project configuration is valid
python manage.py test hostelapp --verbosity 2	All tests executed successfully (OK)	Backend logic validated with unit tests
Step 1: Launch application	Homepage loads successfully	System accessible via browser
Step 2: User login	Authentication successful	Login system working correctly
Step 3: Add hostel	Hostel created successfully	Hostel management module functional
Step 4: Add floors, rooms, beds	Structure created successfully	Hierarchical setup validated
Step 5: Register student	Student record saved	Registration module working
Step 6: Allocate bed	Bed assigned correctly	Allocation logic enforced
Step 7: Submit complaint	Complaint stored successfully	Complaint module functional
Step 8: View complaints	Complaints displayed correctly	Tracking system works
Step 9: Open dashboard	Dashboard loads with data	Aggregated metrics displayed
Step 10: Open analytics page	Real analytics displayed with chats	Analytics module partially implemented

Table 2: System Unit Test Results

Test Case	Description	Expected Result	Actual Result	Status
test_hostel_aggregates	Validate hostel structure calculations	Correct counts of floors, rooms, beds	Accurate values returned	Pass
test_dashboard_view_ requires_login	Check access restriction	Unauthenticated users blocked	Access denied correctly	Pass
test_dashboard_view_ for_authenticated_use r	Verify dashboard access	Logged-in user sees dashboard	Dashboard loads successfully	Pass
test_complaint_creati on	Test complaint submission	Complaint saved in database	Successfully stored	Pass
test_notification_creat ion	Test notification system	Notification created and stored	Successfully stored	Pass
test_user_registration _allocation	Test registration and bed allocation	User assigned valid bed	Allocation successful	Pass
test_hostel_page_vie w	Test hostel listing page	Hostels displayed correctly	Page loads with data	Pass
test_analytics_placeho lder_view	Test analytics without data	Placeholder displayed	Placeholder shown correctly	Pass

Table 3: System Test Metrics Summary

Metric	Value	Description
Total Test Cases	8	Number of test scenarios executed
Passed Tests	8	All test cases passed successfully
Failed Tests	0	No failures recorded
Success Rate	100%	Indicates full system reliability
Execution Time	~7.3 seconds	Total time taken to run all tests
Allocation Accuracy	100%	No duplicate or invalid allocations
Data Integrity	High	Database consistency maintained
System Reliability	High	Stable performance under test conditions

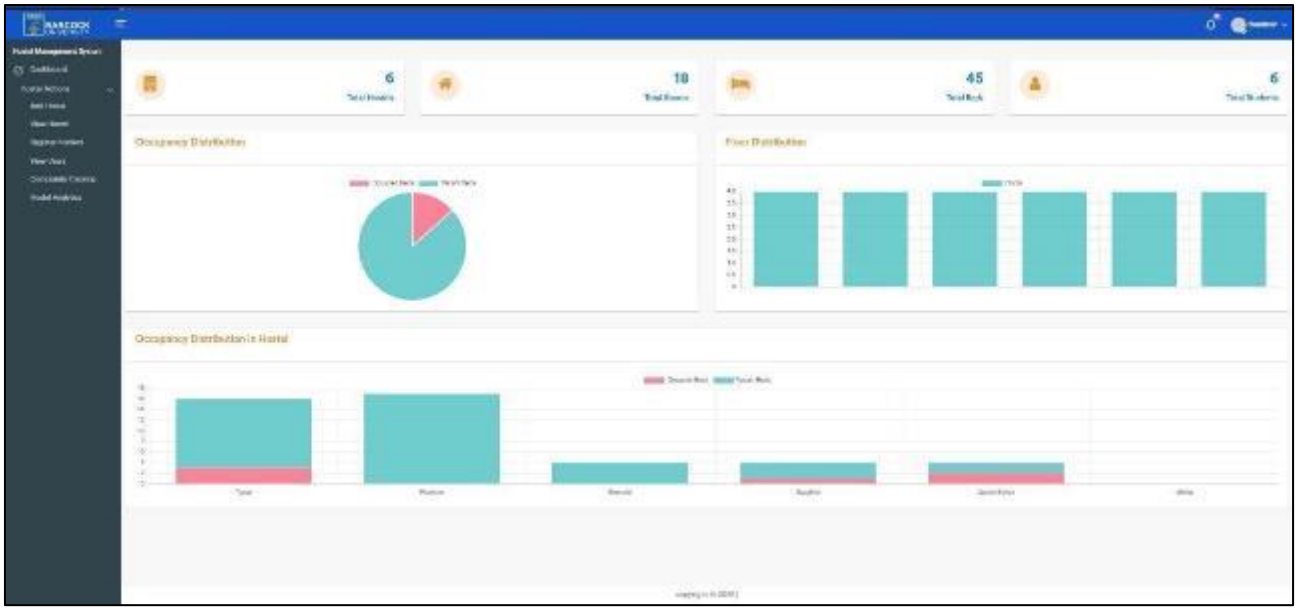

Figure 15: Admin Hostel Analytics Dashboard

The testing results confirmed that the system successfully enforces logical constraints, maintains database integrity, and prevents inconsistent allocations. The results confirm that the system is correctly configured, functionally reliable, and fully integrated. Core modules — authentication, hostel management, allocation, complaints, and analytics — worked as expected, demonstrating readiness for deployment.

5. Conclusion

This paper has presented the design and implementation of a web-based Hostel Allocation System for Babcock University. The system addresses longstanding inefficiencies in manual hostel administration by providing a centralized digital platform that automates room allocation, occupancy monitoring, and administrative reporting. Built on Python's Django framework with SQLite and Plotly for visualization, the system delivers a responsive, secure, and scalable solution accessible from any internet-enabled device.

The system successfully enforces intelligent allocation constraints including gender compatibility, category eligibility, and accessibility checks. Comprehensive testing with a 100% pass rate confirmed the reliability and correctness of all core modules. Administrators benefit from real-time occupancy dashboards and audit logs, while students gain a transparent, guided self-service allocation portal. Future work will focus on integrating online payment processing, developing a dedicated mobile application, migrating to PostgreSQL for production scalability, and implementing AI-driven allocation optimization.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] A. Adamu and S. Salihu, "Challenges of manual hostel management in Nigerian universities," Journal of Educational Administration, vol. 12, no. 2, pp. 45–58, 2016.

[2] O. Ojo and T. Oladele, "Hostel management challenges in Nigerian universities: A case study," Nigerian Journal of Educational Technology, vol. 8, no. 2, pp. 112–128, 2020.

[3] O. Balogun, J. Smith, and T. Adeyemi, "Enhancing student welfare through optimized hostel allocation," Journal of Higher Education Management, vol. 10, no. 1, pp. 45–62, 2024.

[4] M. R. Anderson-Rowland, "Allocation of dormitory rooms in universities," Journal of Educational Administration, vol. 36, no. 3, pp. 256–264, 1998.

[5] S. Hammad, R. Ahmed, and S. Khan, "Manual hostel allocation: Issues and challenges," International Journal of Education and Management Engineering, vol. 3, no. 5, pp. 25–32, 2013.

[6] A. Khamis, M. Jovanovic, and A. Abdullah, "Web-based hostel management portal: A case study," Applied Soft Computing, vol. 52, pp. 190–200, 2017.

[7] R. Elakkiya, K. Selvamani, and S. Vijayalakshmi, "Web-based hostel management system with enhanced security features," International Journal of Pure and Applied Mathematics, vol. 119, no. 12, pp. 1381–1389, 2018.

[8] G. Aswar, S. Patil, and P. More, "Hostel management system using .NET technology and SQL Server," International Journal of Computer Applications, vol. 178, no. 12, pp. 15–20, 2019.

[9] Tutorials Point, "Agile software development lifecycle." Available: https://www.tutorialspoint.com/agile/

[10] L. von Bertalanffy, General System Theory: Foundations, Development, Applications. New York, NY, USA: George Braziller, 1968.

[11] Django Software Foundation, Django Documentation, Version 4.2, 2023. Available: https://docs.djangoproject.com/

[12] N. Megahed and A. Mosa, "Optimization algorithms for hostel room allocation in universities," Computers & Education, vol. 180, pp. 104–132, 2022.

[13] T. Nguyen, H. Tran, and M. Le, "Web-based information systems in educational administration: A review," Educational Technology & Society, vol. 21, no. 3, pp. 45–58, 2018.

[14]

[15] V. Perumal, R. Krishnan, and K. Selvam, "Online entry management and gate pass system for hostels," International Journal of Engineering Research & Technology, vol. 7, no. 4, pp. 210–215, 2018.

[16] Plotly Technologies, "Plotly Python graphing library documentation," 2023. Available: https://plotly.com/python/

[17] P. Ugochukwu, C. Eze, and B. Okonkwo, "Data privacy and security in Nigerian university information systems," African Journal of Information Systems, vol. 15, no. 1, pp. 78–95, 2023.