ISSN 2277-2685
IJESR/April-June. 2026/ Vol-16/Issue-2/1028-1036
G. Shirisha *et. al.*, / International Journal of Engineering & Science Research
Full Length Article

NEXGEN HOSTELS

B. Pravalika[1], G. Shirisha[2], K. Shirisha[3], E. Sowmya Sree[4]
[1] Associate Professor; Department of Computer Science and Engineering Bhoj Reddy Engineering College for Women Hyderabad India.
[2,3,4]B.Tech Student's; Department of Computer Science and Engineering Bhoj Reddy Engineering College for Women Hyderabad India.
Mail Id; shirishakunchala0@gmail.com[2]
Accepted 01-06-2026

Abstract
Finding suitable hostel accommodation is a major challenge for students relocating to cities for higher education. Information regarding hostel availability, rental fees, facilities, safety, and location is often scattered across multiple sources, making the search process time-consuming and unreliable. Traditional methods require students to visit hostels physically, make numerous inquiries, and depend on outdated information, leading to inconvenience and uncertainty.
NexGen Hostels is a web-based Hostel Management System developed to address these challenges through a centralized digital platform. The system enables hostel owners to efficiently manage property listings, update room availability, and maintain hostel information in real time. Students can search, filter, compare, and select hostels based on criteria such as budget, location, amenities, and occupancy. The platform enhances transparency, accessibility, and user convenience by providing accurate information, streamlined communication, and quick decision-making support. By integrating modern web technologies, database management, and online accommodation services, NexGen Hostels offers an efficient, reliable, and user-friendly solution for hostel discovery and management.

Keywords: *Hostel Management System, Student Accommodation, Web-Based Platform, Hostel Search, Room Availability, Digital Hostel Booking, Database Management, Real-Time Updates, Online Accommodation Services, User-Friendly Interface.*

Introduction

The increasing movement of students to different cities for higher education has created a substantial demand for secure, affordable, and convenient accommodation facilities. Hostels serve as one of the most preferred housing options for students due to their affordability and accessibility. However, identifying a suitable hostel remains a challenging task because information regarding available accommodations is often scattered across multiple sources. Students frequently rely on recommendations from friends, relatives, or local contacts, which may not always provide accurate or updated details. As a result, considerable time and effort are spent searching for accommodation before making a final decision.The advancement of web technologies has created opportunities to digitalize accommodation-related services. A centralized online platform can simplify the process by providing comprehensive information about hostels, including room availability, rental charges, facilities, food services, safety measures, and location details. Such a platform can also facilitate communication between students and hostel owners, reducing the dependence on traditional manual methods.To address these challenges, the NexGen Hostels system has been developed as a web-based hostel management platform. The system enables hostel owners to register their properties, maintain updated hostel information, and manage accommodation requests efficiently. Simultaneously, students can search, compare, and evaluate hostels based on their individual preferences. The platform aims to improve transparency, minimize manual effort, and provide a seamless user experience through a centralized digital environment.By integrating hostel discovery and management functionalities into a single platform, NexGen Hostels contributes to a more efficient accommodation ecosystem. The system benefits both students and hostel providers

by simplifying information access, improving decision-making, and enhancing operational efficiency.

Existing System

In the current accommodation environment, the hostel search process is predominantly manual and lacks a structured mechanism for information sharing. Students who relocate to unfamiliar cities often face difficulties in gathering reliable details about available hostels. Most accommodation decisions are based on personal recommendations, local inquiries, or individual visits to hostel premises. These approaches are often inefficient and may not provide complete information regarding room availability, pricing, facilities, or hostel policies.

The absence of a unified information system further complicates the process. Since hostel details are distributed across different channels, students find it difficult to compare multiple accommodation options effectively. In many situations, they must visit several hostels before identifying a suitable choice, resulting in increased travel expenses and delayed decision-making. These limitations highlight the need for a modern digital solution capable of providing centralized and real-time hostel information.

Proposed System

The proposed NexGen Hostels platform introduces a web-based solution that centralizes hostel-related information and management activities. The system provides a digital environment where students can search for hostels according to various criteria such as location, budget, room type, and available amenities. Detailed and verified information regarding room availability, rental costs, food facilities, hostel regulations, photographs, and user reviews is made available through an easy-to-use interface.

Hostel owners are provided with tools to manage hostel profiles, update accommodation details, and respond to user requests efficiently. The platform facilitates direct interaction between users and hostel providers while ensuring that information remains accurate and current. Through automation and digitalization, the system significantly reduces the effort involved in accommodation search and management processes.

REQUIREMENT ANALYSIS

Requirement analysis is a critical phase in software development that identifies and defines the functionalities expected from the system. The NexGen Hostels platform is designed to support three primary stakeholders: users, hostel owners, and administrators. Each stakeholder is provided with specific functionalities to ensure efficient hostel management and accommodation services.The User Module enables students and other accommodation seekers to create an account and securely access the system. After authentication, users can browse hostel listings, view detailed information regarding facilities and availability, submit accommodation requests, monitor responses from hostel owners, and provide ratings based on their experiences. Users also have the option to remove previously submitted ratings and securely terminate their sessions through the logout feature.The Hostel Owner Module allows accommodation providers to register their hostels and manage related information through a dedicated interface. Owners can update hostel details, monitor accommodation requests submitted by users, and either approve or reject requests based on room availability and hostel policies. This module streamlines communication between owners and prospective residents while ensuring that information remains current and accessible.

The Administrator Module serves as the supervisory component of the system. Administrators are responsible for managing user and hostel owner accounts, monitoring platform activities, reviewing ratings, handling complaints, and maintaining overall system integrity. Through centralized control, administrators ensure smooth operation and compliance with platform policies.

Computational Resource Requirements

The successful deployment of the NexGen Hostels platform requires appropriate software and hardware resources. The software environment includes Windows 11 as the operating system, which provides a stable platform for development and deployment activities. The user interface is developed using HTML, CSS, and JavaScript, enabling responsive and interactive web pages. Server-side functionalities are implemented using Python and Node.js, which support efficient data processing and application logic. MongoDB is utilized as the database management system due to its flexibility and capability to manage large volumes of structured and semi-structured data.From a hardware perspective, the system requires a computer equipped with at least an Intel Core i5 processor, 4 GB of RAM, and approximately

120 GB of storage capacity. These specifications provide sufficient computational power for development, testing, and operational activities associated with the platform.

Software Development Life Cycle Model

The development of NexGen Hostels follows the Waterfall Software Development Life Cycle (SDLC) model. The Waterfall model is a structured and sequential development methodology in which each phase is completed before progressing to the next stage. This approach is particularly suitable for projects with clearly defined requirements and limited scope for major changes during development.The process begins with problem identification, where challenges associated with hostel accommodation search and management are analyzed. This phase establishes the foundation for understanding user needs and project objectives. Requirement gathering follows, involving the collection of information from students, hostel owners, and other stakeholders through surveys, interviews, and domain research.The gathered information is then documented systematically to ensure clarity and consistency. Requirement analysis and prioritization are subsequently performed to evaluate feasibility, significance, and implementation complexity. Following this stage, a feasibility study is conducted to assess technical, economic, and operational viability.The final outcome of the analysis process is the Software Requirement Specification (SRS) document, which serves as the primary reference for system design and implementation. By following a structured development methodology, the project ensures effective planning, documentation, and quality assurance throughout its lifecycle.

SYSTEM DESIGN

System design represents the process of transforming requirements into a structured framework that guides software development. It defines how different components interact, process requests, and exchange information to achieve the desired functionality. A well-designed architecture improves system performance, maintainability, scalability, and reliability. In the NexGen Hostels platform, the design phase focuses on organizing software components and technical infrastructure to support efficient hostel management and accommodation services.

The architecture of the proposed system illustrates the flow of information among users, hostel owners, administrators, application services, and the database. Through this structured approach, the platform ensures smooth communication and efficient processing of user requests. The design is categorized into Software Architecture and Technical Architecture, each addressing different aspects of system implementation.

Software Architecture

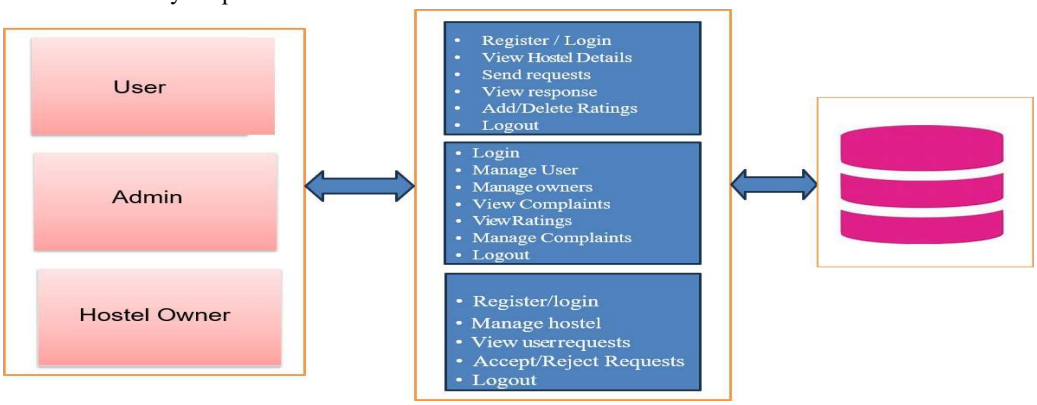


Fig 1 Software architecture

The software architecture defines the logical organization of the application and the interaction between its major modules. The NexGen Hostels platform follows a multi-tier architecture consisting of presentation, application, and data layers. The presentation layer provides interfaces through which users, hostel owners, and administrators interact with the system. This layer is responsible for collecting user input and displaying processed information.

The application layer contains the business logic that manages authentication, hostel search operations, request handling, rating management, complaint processing, and administrative functions. It acts as the intermediary between users and the database by validating requests and executing appropriate operations.

The data layer is responsible for storing and retrieving information related to users, hostel owners, hostel details, accommodation requests, ratings, and complaints. MongoDB serves as the primary database management system, enabling efficient storage and retrieval of data. This layered architecture promotes modularity, simplifies maintenance, and enhances overall system performance.

Technical Architecture

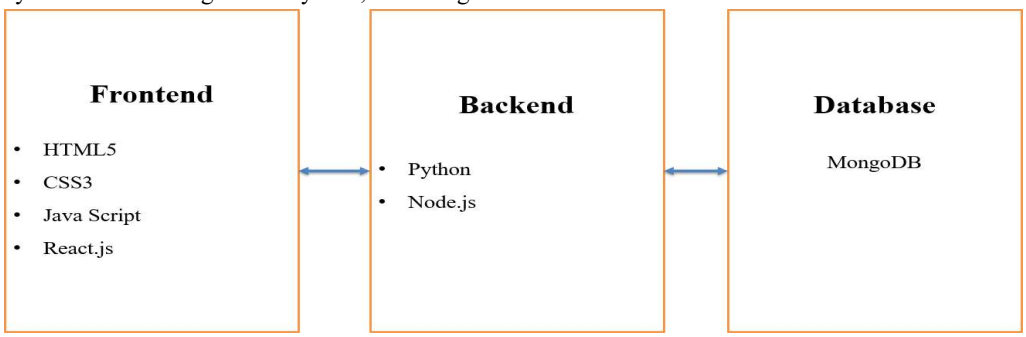


Fig 2 Technical architecture

The technical architecture describes the technological infrastructure required to implement and operate the system. The client side is developed using HTML, CSS, and JavaScript, providing an interactive and responsive user experience across different devices and browsers. The server side utilizes Python and Node.js to process requests, manage application logic, and coordinate communication between users and the database.

MongoDB functions as the backend data repository, storing all system-related information in a secure and organized manner. Communication between the client and server is performed through web-based protocols that ensure reliable data transmission. The architecture supports secure authentication, efficient resource utilization, and scalability for future expansion.

By integrating modern web technologies with a structured architectural framework, the NexGen Hostels platform delivers a reliable, scalable, and user-friendly solution for hostel discovery and management.

IMPLEMENTATION

The development of the NexGen Hostels platform is based on a modern web technology stack designed to ensure efficiency, scalability, and ease of use. The system integrates frontend, backend, and database technologies to deliver a seamless user experience and reliable data management.

HTML (HyperText Markup Language) is used as the structural foundation of the web application. It defines the layout and organization of web pages by creating essential elements such as forms, navigation menus, buttons, headings, and content sections. In this system, HTML is used to design core interfaces including the homepage, authentication pages, hostel listing pages, and booking-related forms.

CSS (Cascading Style Sheets) is utilized to enhance the visual presentation of the application. It controls styling aspects such as colors, typography, spacing, alignment, and responsive layout behavior. CSS ensures that the platform remains visually consistent and accessible across various screen sizes and devices, thereby improving user experience.

JavaScript is incorporated to provide dynamic behavior and interactivity within the system. It enables client-side validation, real-time filtering of hostel listings, form handling, and responsive user interactions. By reducing page reloads and improving responsiveness, JavaScript enhances the overall usability of the platform.

Node.js is employed as the server-side runtime environment. It is responsible for handling client requests, processing business logic, managing authentication processes, and facilitating communication between the frontend and database. Its event-driven and non-blocking architecture makes it suitable for developing scalable web applications such as NexGen Hostels.

MongoDB is used as the database management system for storing and managing application data. It follows a document-oriented structure, which allows flexible storage of hostel information, user profiles, booking requests, ratings, and related records. Its scalability and fast query performance make it well-suited for handling large volumes of dynamic data.

Pseudocode

The system workflow can be represented using a high-level pseudocode structure that outlines the major operations performed within the platform.

The general process begins with user authentication, followed by hostel browsing, request submission, and response handling. Hostel owners manage listings and respond to user requests, while administrators oversee system activities and ensure proper functioning. This logical flow ensures smooth interaction between all modules and maintains system consistency.

TESTING

Software testing is a critical phase in the software development life cycle that focuses on evaluating the system to ensure it meets specified requirements and functions as intended. It plays an essential role in identifying defects, improving software quality, and ensuring system reliability before deployment. In modern digital environments, where software applications are widely used for essential services, testing becomes a vital process to avoid failures that may lead to financial loss, data breaches, or reduced user trust.

The primary objective of software testing is to verify that the developed system performs correctly under different conditions and meets both functional and non-functional requirements. Effective testing contributes to improved system performance, enhanced security, and higher user satisfaction.

Dimensions of Testing

Software testing can be analyzed across multiple dimensions to ensure comprehensive evaluation of the system. These dimensions include different layers of the application such as the user interface layer, application programming interfaces (APIs), and database layer. Each layer requires specific validation to ensure proper functionality and integration.Testing is also categorized based on scale, which includes unit-level testing, module-level testing, integration testing, and system-level testing. These levels ensure that individual components as well as the complete system operate correctly when combined.Another important dimension is the type of testing performed. This includes functional testing, performance testing, security testing, and usability testing, each focusing on different quality attributes of the system. Additionally, testing methodologies such as manual testing, automated testing, scripted testing, and exploratory testing are applied depending on project requirements and complexity.

Stages of Testing

The testing process is carried out in several structured stages to ensure software quality and reliability.

Unit testing is the initial stage where individual components or functions of the system are tested independently. This phase ensures that each module performs correctly in isolation and helps identify defects early in the development process.Integration testing follows unit testing and focuses on verifying the interaction between different modules. It ensures that combined components work together as expected and that data flows correctly between system modules.

System testing evaluates the complete integrated system as a whole. It verifies whether the software meets all specified functional and non-functional requirements. This testing is typically performed in an environment that closely resembles real-world deployment conditions.Acceptance testing, also known as User Acceptance Testing (UAT), is the final stage of testing. In this phase, end users evaluate the system to determine whether it satisfies their requirements and is ready for deployment. Successful completion of this stage confirms that the system is suitable for production use.

Types of Testing

Two major approaches are commonly used in software testing: black-box testing and white-box testing.

Black-box testing evaluates the system based on its external behavior without considering internal code structure. Test cases are derived from system requirements and input-output specifications. This method is applicable across all levels of testing, including unit, integration, system, and acceptance testing.White-box testing, on the other hand, focuses on internal code structure and logic. It requires knowledge of programming and is primarily used at the unit testing level. Techniques such as statement coverage, branch coverage, and path coverage are used to ensure that all possible execution paths within the code are tested thoroughly.

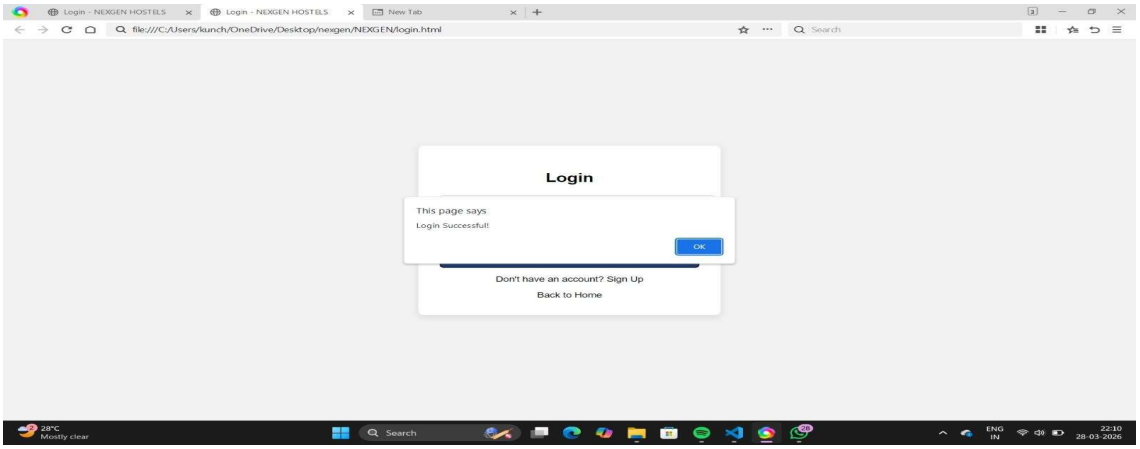
Test Cases Overview

Test cases are designed to validate the functionality of each module within the system. The homepage module is tested to ensure proper loading of interface elements and navigation functionality. The user login module is evaluated for authentication accuracy and security validation. The hostel owner login module is tested to confirm correct access control and dashboard redirection.Additional test cases include verification of hostel addition

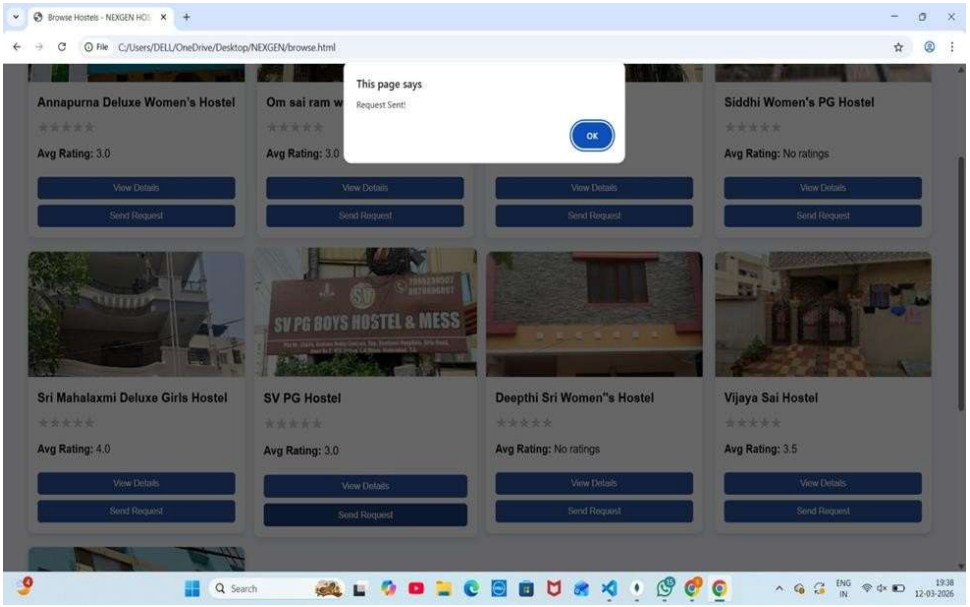
functionality, browsing and filtering of hostels, detailed hostel view operations, and logout functionality for all user roles. These test cases ensure that each module operates as expected and contributes to the overall reliability of the system.

Screenshots

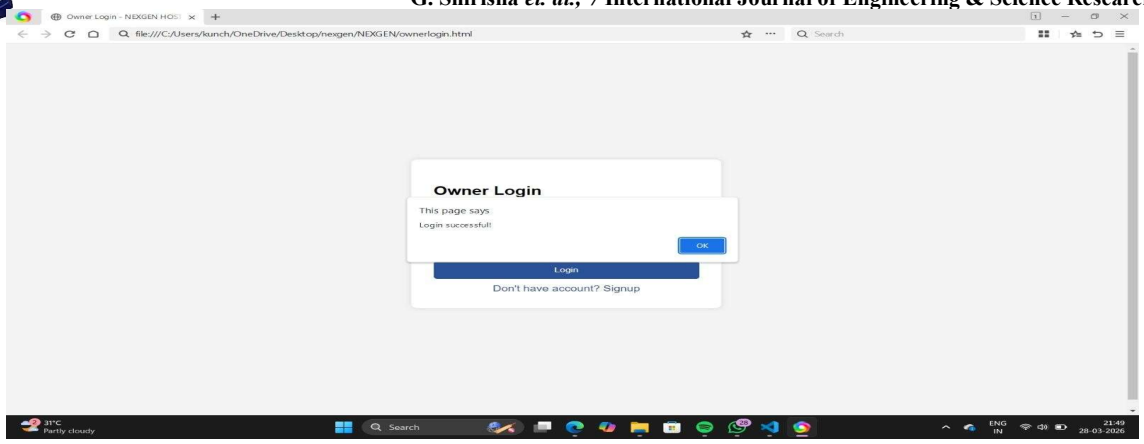
Screenshot 1 Home Page



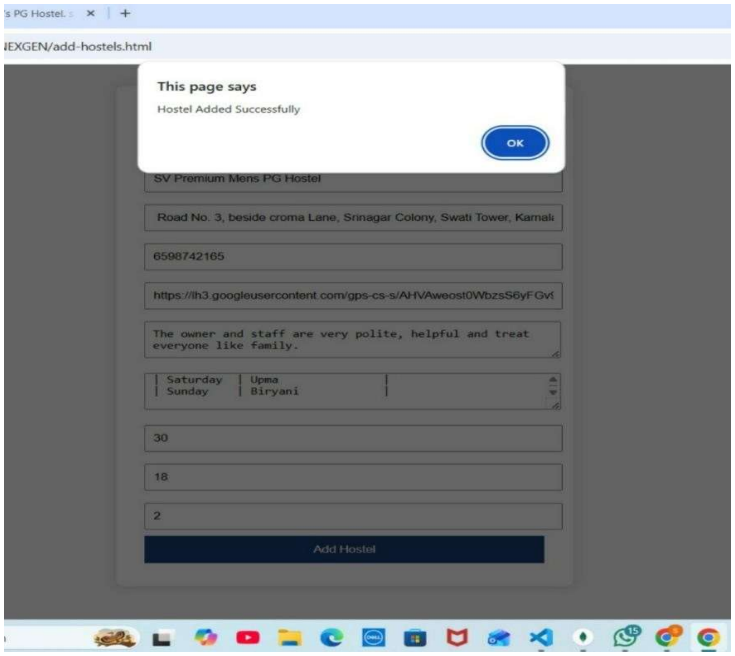
Screenshot 2 Login Page



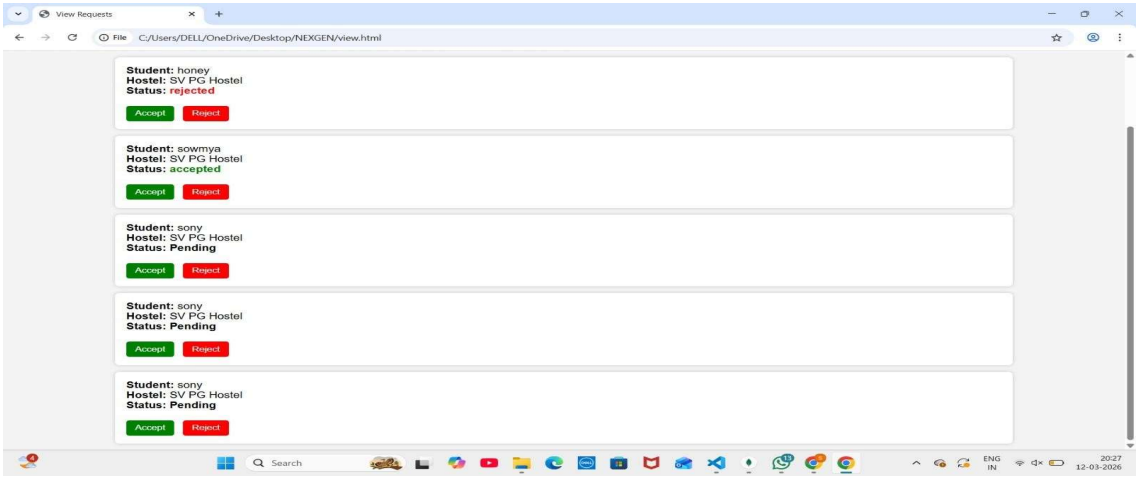
Screenshot 3 Browse Hostels page



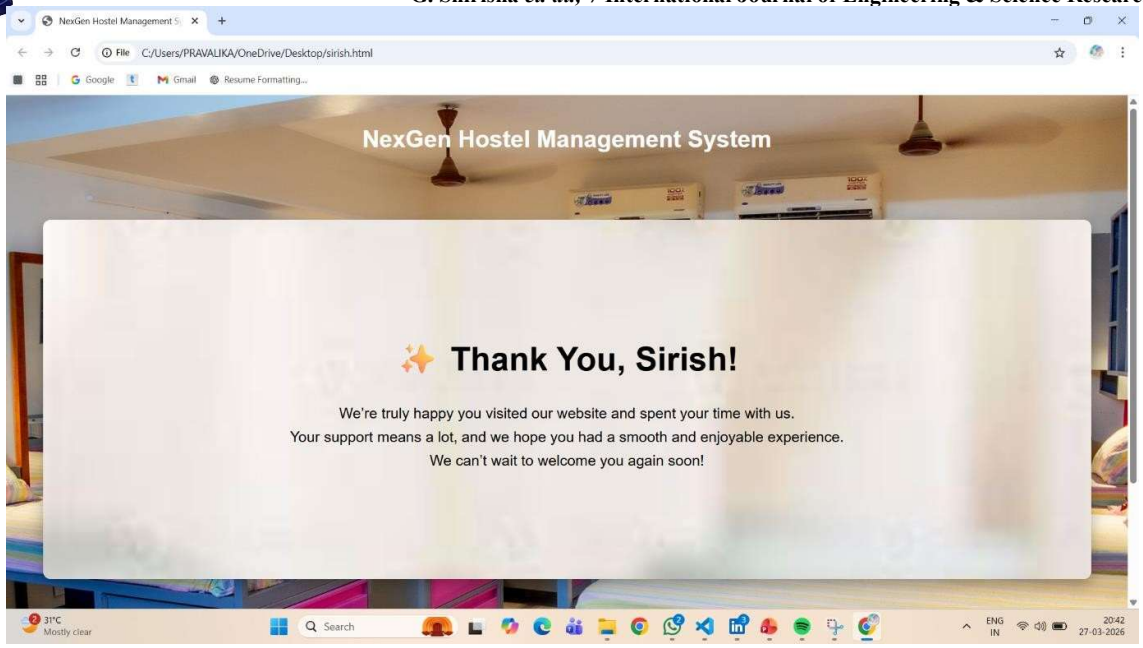
Screenshot 4 Owner Login page



Screenshot 5 Add Hostel page



Screenshot 6 View Requests page



Screenshot 7 Home page

Conclusion

The NexGen Hostels platform has been developed as a centralized web-based solution to address the challenges associated with traditional hostel searching and management processes. The system effectively consolidates hostel-related information into a single digital platform, enabling users to access accurate and updated details regarding accommodation options. By reducing dependency on manual inquiries and physical visits, the proposed system significantly improves efficiency, transparency, and communication between students and hostel owners. Furthermore, it minimizes administrative workload for hostel providers while enhancing the overall decision-making process for users. As a result, the platform contributes to a more organized, reliable, and user-friendly hostel management ecosystem.

Future Scope

The NexGen Hostels system has strong potential for further enhancement through the integration of advanced features and emerging technologies. One important improvement is the incorporation of secure online payment gateways, which would enable users to book hostel rooms and complete fee transactions directly through the platform, thereby increasing convenience and automation.

In addition, the system can be expanded to support multiple cities and regions, and in the long term, it may be extended to an international level to broaden its accessibility and user base. Such expansion would significantly increase the platform's applicability in diverse geographical contexts.

Another possible enhancement involves the introduction of subscription-based services for hostel owners. This model would allow premium listings, improved visibility, and targeted promotion of hostel facilities within the platform. These future improvements would further strengthen the system's scalability, functionality, and commercial viability while enhancing the overall user experience.

REFERENCES

1) Kübler-Ross, E., & Byock, I. (2025). *On Death & Dying: What the Dying Have to Teach Doctors, Nurses, Clergy & Their Own Families* (50th Anniversary ed.). Scribner. This work provides insights into human behavior and decision-making processes relevant to user interaction studies.
2) Watson, J. B., & Rayner, R. (2024). *Conditioned Emotional Reactions: The Case of Little Albert* (D. Webb, Ed.). CreateSpace Independent Publishing Platform. This classical psychological study explains behavioral conditioning and response patterns.
3) Kundariya, H. (2025). "Can AI Replace Web Developers: A Practical Look at Current Tools and Limitations." SitePoint. Retrieved March

10, 2026. This article discusses the role of AI in modern web development and its limitations.

4) Mozilla Developer Network (MDN). (2024). *The Web and Web Standards*. Retrieved from https://developer.mozilla.org. This source provides essential guidelines for HTML, CSS, and JavaScript development standards.
5) Krueger, A. (2025). "Remembering Duluth's Famous Mongoose, Mr. Magoo." Duluth News Tribune. Archived February 1, 2026. This article is part of general digital publication references.
6) Node.js Official Release Notes. (2026). Version 25.6.1 (Current). Retrieved February 15, 2026. This documentation explains backend runtime updates and performance improvements in Node.js.
7) ECMAScript Language Specification. (2025). JavaScript Standard Language Specification. Retrieved April 17, 2024. This defines standardized rules for JavaScript programming and ensures cross-platform compatibility.
8) Stack Overflow Developer Survey. (2025). Stack Overflow. Retrieved October 10, 2025. This survey provides statistical insights into developer tools, frameworks, and industry trends.
9) W3C Web Standards Documentation. (2024). World Wide Web Consortium. This defines global standards for web technologies and ensures interoperability across browsers.
10) Oracle. (2025). MongoDB Documentation and Database Design Guide. Retrieved 2026. This provides guidelines for NoSQL database design and scalable data management used in web applications.