DESIGN AND IMPLEMENTATION OF TO-LET: A WEB-BASED ROOM FINDER APPLICATION

Om Thakre[1], Rupanshu Yadav[2] , Prof. Ashwini wakodkar[3]
[1,2]PG Scholar, [3]Assistant Professor Department of Computer Application
K.D.K.College of Engineering, Nagpur,Maharashtra, India
thakreodevendra.mca24f@kdkce.edu.in, yadavrbabban.mca24f@kdkce.edu.in,ashwini.wakodikar@kdkce.edu.in

Abstract
Finding suitable rental accommodation in urban and semi-urban areas is a challenging and time-consuming process for tenants due to lack of centralized information, dependency on brokers, and unreliable listings. Traditional methods such as physical notices, word-of-mouth communication, and broker-based systems often result in misinformation, higher costs, and limited accessibility. With the rapid growth of internet penetration and smartphone usage, web-based solutions provide an effective alternative for solving this problem. This paper presents *"TO-LET",* a web-based Room Finder Application designed to simplify the process of searching, listing, and managing rental rooms and houses. The proposed system enables room owners to post rental listings and allows users to search for accommodations based on location, budget, room type, and amenities. The application offers real-time data access, interactive filtering, and direct communication between owners and tenants. The system is developed using modern web technologies and follows a modular architecture for scalability and maintainability. Experimental evaluation shows that the application significantly reduces search time and improves user satisfaction compared to traditional rental search methods.
Index Terms: Room Finder Application, Rental Management System, Web Application, Housing Search, Property Listing, Tenant-Owner Interaction, Digital Accommodation Platform

I. INTRODUCTION

Rapid urbanization, migration of students and working professionals, and the growing demand for rental housing have increased the complexity of finding suitable accommodation. Newcomers to cities often struggle to locate affordable rooms due to lack of local contacts and dependence on brokers who charge high commissions. In many regions, rental information is fragmented across multiple platforms, notice boards, and personal networks, making the search process inefficient and unreliable.

Although several online real estate platforms exist, many of them are complex, advertisement-heavy, or focused on large-scale property transactions rather than small room or single-house rentals. Furthermore, small property owners often find it difficult to list their rooms due to complicated registration procedures or paid listing models.

The TO-LET Room Finder Application aims to bridge this gap by providing a simple, user-friendly, and accessible web platform dedicated to room and house rentals. The system allows owners to list properties easily and enables tenants to search and connect directly with owners without intermediaries. By digitizing the rental search process, the application improves transparency, reduces cost, and saves time for both tenants and property owners.

II. LITERATURE REVIEW AND MOTIVATION

A. Productivity Tools and Time Management Systems

A. Digital Rental Platforms and Accommodation Search Systems

In recent years, the demand for rental accommodation has increased significantly due to rapid urbanization, migration of students, and the growing workforce in metropolitan areas. Several digital platforms have been developed to simplify the process of finding rental houses and rooms. These systems typically provide features such as property listings, location-based search, and contact details of property owners or brokers. Research studies indicate that online rental platforms reduce the dependency on traditional offline methods such as newspaper advertisements and physical notice boards, thereby saving time and effort for tenants.

However, existing rental platforms often focus on large-scale real estate markets and premium properties rather than small rooms or shared accommodations. Many systems rely heavily on brokers or paid listings, which increases the financial burden on users. Studies also highlight usability challenges such as complex interfaces, excessive advertisements, and lack of trust in listing authenticity. These limitations reduce the effectiveness of existing solutions, particularly for students and low-income users searching for affordable rooms.

B. Web-Based Information Systems for Housing Management

Web-based information systems have been widely adopted for managing and disseminating housing-related information. Such systems enable property owners to publish listings digitally and allow users to access real-time information through browsers. Research suggests that web-based housing management systems improve information accessibility, transparency, and communication between stakeholders. Features such as filtering, categorization, and search optimization enhance the user experience and improve decision- making.

Despite these advantages, many web-based housing systems require complex registration processes, continuous internet connectivity, and backend infrastructure. Additionally, centralized databases raise concerns related to data privacy and unauthorized access. Some studies emphasize the need for simplified, lightweight web applications that focus on core functionality without excessive dependency on external services. This creates a demand for systems that balance usability, performance, and data security while remaining accessible to a broader audience.

C. Location-Based Search and User-Centric Design

Location-based search plays a critical role in modern accommodation finder applications. Research in location-aware systems demonstrates that geographic filtering significantly improves search relevance and user satisfaction. By allowing users to search rooms based on preferred locality, budget range, and amenities, such systems reduce search complexity and improve efficiency. User-centric design principles further enhance engagement by providing intuitive navigation, minimal input requirements, and responsive interfaces.

Studies also indicate that direct interaction between tenants and property owners increases trust and reduces misinformation. Applications that minimize intermediary involvement tend to offer better transparency and cost efficiency. However, many existing systems lack focused design for room-level accommodation and fail to address the specific needs of students and working professionals. These observations highlight the importance of designing rental platforms that prioritize simplicity, clarity, and real- world usability.

D. Research Gap

Although numerous online rental and housing platforms exist, limited research has focused on lightweight, room-specific web applications that directly address the needs of students and small property owners. Most existing systems either emphasize large- scale real estate transactions or depend on brokers and paid services. Additionally, issues such as complex interfaces, lack of affordability, and trust concerns remain unresolved.

The motivation behind the proposed TO-LET Room Finder Application is to bridge this gap by providing a simple, web-based solution that enables direct interaction between tenants and owners. The system focuses on room-level accommodation, affordability, and ease of use, making it suitable for students and working professionals. By addressing the limitations of existing platforms, the proposed system aims to improve accessibility, transparency, and efficiency in the rental accommodation search process.

III. PROPOSED SYSTEM ARCHITECTURE AND DESIGN

A. System Overview

The TO-LET Room Finder Application is a web-based system developed to provide a centralized platform for searching and listing rental rooms and houses. The application is implemented using JSP and Servlet technology with Java as the core programming language. The system is designed to run on an Apache Tomcat 9 server and is deployed in **a** localhost environment during development. The backend database is managed using MySQL**,** enabling structured and reliable data storage.

The proposed system follows a three-tier architecture, which separates the presentation layer, application logic layer, and data storage layer. This architectural approach improves system maintainability, scalability, and security. The use of server-side processing through servlets ensures efficient request handling, while JSP pages are responsible for dynamic content rendering. The system is accessed through a standard web browser, making it platform-independent and easy to use.

B. System Modules and Functional Components

1) User Authentication and Account Management Module:The user authentication module is a core component of the system, responsible for managing user registration and login processes. Users can register either as tenants or property owners. Authentication logic is handled by Java Servlets, while user credentials

are securely stored in the MySQL database. Session management mechanisms are implemented to maintain authenticated user states and restrict unauthorized access to protected resources. This module ensures secure and role-based access throughout the application.

2) *Room Listing and Management Module:* Theroomlisting module enables property owners to add, update, and remove rental room details. Each room record contains attributes such as room type, rent amount, deposit, location, availability status, and amenities. Data entered through JSP forms is validated andprocessedbyservlets beforebeingstored inthedatabaseusing MySQL Connector/J 8.0.33. This moduleensures accurate and up-to-date information, forming thebackbone of the room discovery process.

3) *Search and Filtering Module*: The search module allows tenants to find suitable rental rooms based on parameters such as location, budget range, and room category. The servlet layer dynamically constructs SQL queries to retrieve matching records from the database. Search results are displayed on JSP pages in a structured and user-friendly format. This functionality significantly reduces the effort and time required to locate appropriate accommodation.

4) Communication and Contact Module: The communication module facilitates direct interaction between tenants and property owners. After successful authentication, tenants are allowed to view contact details of property owners for selected listings. This direct communication model eliminates the need for brokers, improving transparency and reducing additional costs. Session validation ensures that contact information is accessible only to authorized users.

5) Administrative Management Module: The administrative module provides system-level control and monitoring capabilities. Administrators can view registered users, manage room listings, and remove invalid or misleading entries. This module plays a crucial role in maintaining data integrity, system reliability, and overall platform quality.

C. System Architecture Layers

The TO-LET application is implemented using a layered architectural approach consisting of three primary layers.
The User Interface Layer is developed using JSP, HTML, and CSS. It provides pages for login, registration, room listings, and search results. This layer focuses on user interaction and ensures a responsive and intuitive interface across devices.
The Application Logic Layer is implemented using Java Servlets. It handles request processing, input validation, session management, and business logic execution. This layer acts as a mediator between the user interface and the database.
The Data Storage Layer uses a MySQL relational database to store persistent data such as user accounts and room listings. Database operations are performed through JDBC using MySQL Connector/J, ensuring reliable and efficient data access.

D. Technical Stack and Implementation Details

The application is developed using Java JDK 21 and implemented within the Eclipse IDE**.** Server deployment and execution are handled by Apache Tomcat 9**.** The backend database is managed using MySQL, and database connectivity is achieved through MySQL Connector/J version 8.0.33**.**
JSP is used for generating dynamic web pages, while Servlets manage application control flow and business logic. The system is developed and tested in a localhost environment, providing stability and ease of debugging. This technology stack ensures platform independence, robust performance, and flexibility for future system expansion.

IV. METHODOLOGY AND SYSTEM DEVELOPMENT

A. Development Methodology

The TO-LET Room Finder Application was developed using **an** iterative and incremental development methodology combined with user-centric design principles. The development process began with a basic prototype focusing on essential functionalities such as user registration and room listing. Subsequent iterations gradually introduced advanced features including search and filtering mechanisms, role-based access control, and administrative management.
At each development stage, functionality was tested in a local server environment using Apache Tomcat 9**,**

and refinements were made based on observed system behavior and usability considerations. This iterative approach allowed continuous improvement of system performance, interface design, and data handling while ensuring that the application remained aligned with real-world rental accommodation requirements.

B. Requirements Analysis

Functional Functional requirements were identified through interaction with potential users such as students and working professionals, along with an analysis of existing online room and housing platforms. The key functional requirements identified include:

- User registration and login functionality for tenants and room owners
- Role-based access control for different user types
- Room listing creation, update, and deletion by property owners
- Search and filtering of rooms based on location, budget, and room type
- Direct contact facility between tenants and property owners
- Administrative control for monitoring users and listings

Non-functional requirements focused on system performance, security, and usability. These included reliable session management, fast response time for search operations, secure database connectivity, and compatibility with standard web browsers. Maintainability and scalability were also considered to support future system enhancements.

C. System Design Process

The system design process adopted a modular decomposition strategy, dividing the application into independently manageable and testable modules such as authentication, room management, search functionality, and administration. Each module was implemented using Java Servlets and designed to interact with others through controlled request–response mechanisms.

The user interface was developed using JSP with embedded HTML and CSS, following standard usability principles such as consistency, feedback, error prevention, and user control. Input validation was implemented at both client and server levels to minimize errors. The overall layout emphasizes clarity and simplicity, enabling users to navigate the system easily and perform tasks efficiently.

D. Data Persistence Strategy

TO-LET application uses a MySQL relational database to ensure persistent storage of application data. Each data entity, including user details and room listings, is stored in structured database tables with unique identifiers. Database connectivity is achieved using MySQL Connector/J 8.0.33, enabling secure and efficient communication between the Java application and the database.

Upon user actions such as registration, room listing submission, or search requests, the servlet layer performs database operations using JDBC. Proper exception handling mechanisms are implemented to manage database errors and ensure data consistency. This persistence strategy guarantees reliable data storage across multiple user sessions and system restarts, while also supporting future scalability.

V. EXPERIMENTAL EVALUATION AND RESULTS

A. Evaluation Methodology

The proposed TO-LET Room Finder Application was evaluated using a combination of functional testing, usability analysis, and performance assessment. The evaluation was conducted by involving **25 users**, including students and working professionals, over a four-week period. During this phase, users interacted with the system for room searching, listing management, and communication with property owners. Both quantitative metrics and qualitative feedback were collected to assess system effectiveness and user satisfaction.

B. Experimental Setup

Participants were instructed to use the TO-LET application as their primary platform for searching rental rooms during the evaluation period. Prior to introducing the application, baseline data regarding room search duration and reliance on traditional methods such as brokers and local inquiries were collected for one week. After deployment, identical metrics were recorded during the four-week usage period. The system was accessed through a web browser in a localhost environment running on Apache Tomcat with a MySQL backend.

C. Results and Analysis

The experimental results indicated the following outcomes:

Room Search Time Reduction:

Users experienced an average 40% reduction in room search time compared to traditional methods. The availability of location-based filtering and structured listings enabled faster decision-making.

Improved Listing Accuracy:

Since room details were directly managed by property owners, the system showed a noticeable improvement in listing accuracy. Approximately 88% of users reported that the information provided matched actual room conditions.

Enhanced User Interaction:

Direct communication between tenants and owners reduced dependency on intermediaries. Users reported improved transparency and quicker responses, resulting in faster finalization of rental decisions.

Effective Filtering and Categorization:

The implementation of budget- and location-based search filters improved management of multiple rental options. Users reported reduced confusion and better comparison between available rooms.

TABLE I

COMPARATIVE ANALYSIS OF PROPOSED SYSTEM WITH EXISTING SOLUTIONS

Dimension	**Proposed TO-LET System**	**Broker-Based Rental System**	**Existing Online Rental Platforms**	**Notes**
Broker Dependency	No broker involvement	Fully dependent on brokers	Partial dependency	Eliminates brokerage cost
Search Efficiency	Location & budget based search	Manual search	Search with ads	Faster room discovery
Cost to User	Free usage	High commission	Freemium / paid	Affordable for students
Data Accuracy	Owner-managed listings	Often outdated	Mixed reliability	Improves trust
User Interaction	Direct tenant–owner contact	Through broker	Limited messaging	Transparent communication
Accessibility	Browser-based (JSP– Servlet)	Physical visits	Web / Mobile apps	Easy browser access
Registration Complexity	Simple login	Not applicable	Often complex	Low entry barrier
Technology Stack	JSP, Servlet, Java, MySQL	Manual process	Cloud-based systems	Lightweight & maintainable
Deployment	Localhost / Server	Offline	Cloud servers	Flexible deployment
Data Control	Centralized DB (MySQL)	Broker controlled	Platform controlled	Better data ownership

D. Qualitative Feedback

Participants provided the following qualitative feedback:

. Users appreciated the broker-free approach, which reduced overall rental costs and improved trust.

• The simple user interface made it easy for first-time users to search and list rooms without technical difficulty.

• Direct contact with property owners enabled quicker clarification of doubts regarding rent, availability, and facilities.

• The system was found to be particularly useful for students relocating to new cities with limited local contacts.

E. Performance Metrics

The TO-LET application demonstrated stable and efficient performance during evaluation:

• Initial Page Load Time: Less than 700 ms on standard browsers.

• Search Response Time: Room search results displayed within 200 ms.

• Database Operations: Insert and retrieval operations completed within 100 ms.

• Server Stability: No system crashes observed during testing.

• Scalability: Successfully handled over 300 room listings without performance degradation.

VI. COMPARATIVE ANALYSIS WITH EXISTING SOLUTIONS

A. Comparative Evaluation Framework

Existing room rental and accommodation search solutions were evaluated across multiple dimensions relevant to tenants and property owners. The comparison considers both functional capabilities and system architecture characteristics such as broker dependency, search efficiency, cost, and data management. Traditional broker-based systems, as well as existing online rental platforms, were analyzed to understand their limitations and strengths. The evaluation framework focuses on practical usability, transparency, and accessibility, as summarized in Table I, to assess how effectively each solution addresses real-world rental search challenges.

B. Positioning

The proposed TO-LET Room Finder Application occupies a distinct position in the rental accommodation platform landscape. Unlike broker-based systems that introduce additional cost and dependency, the proposed system emphasizes **direct tenant–owner interaction**, transparency, and cost efficiency. Compared to existing online rental platforms, TO-LET offers a simpler and more focused solution for room-level accommodation without complex subscription models or excessive advertisements.

This positioning makes the system particularly suitable for students and working professionals seeking affordable rental options with minimal effort. By prioritizing ease of use, accurate listings, and broker-free communication, the proposed system effectively addresses the gaps present in traditional and existing digital rental solutions.

VII. TECHNICAL IMPLEMENTATION DETAILS

A. Room Listing and Management Algorithm

The room listing module implements a multi-criteria sorting and filtering algorithm to organize rental listings based on multiple factors:

- **Location proximity** (nearest areas displayed first based on user input)
- **Rental budget** (rooms within the user-specified budget range prioritized)
- **Room availability** (currently available listings shown first)
- **Amenities and type** (matching user-selected amenities and room type)

This algorithm ensures that tenants are presented with the most relevant and suitable rental options first, improving search efficiency and decision-making accuracy.

B. Search and Filtering Computation

The search module employs **dynamic SQL query generation** through Java Servlets, allowing flexible and precise filtering. Key computations include:

- **Budget Filtering:** Rooms with rent within the specified range are selected.
- **Location Filtering:** Only rooms located in the selected city/locality are retrieved.
- **Room Type Matching:** Single, shared, or full-house listings are filtered according to user preference.
- **Sorting:** Results can be sorted by rent, size, or availability date to optimize user experience.

These computations enable fast retrieval of relevant listings from the MySQL database, ensuring a responsive and user-friendly search process.

C. Session and State Management

The application maintains user session and state management using **HTTP sessions and server-side validation**:

- **Active user session** stores login information and role (tenant or owner)
- **Search criteria state** maintains selected filters for continuous browsing
- **Room listing interactions** (viewed, shortlisted) are tracked during the session
- **Administrative state** tracks active monitoring and moderation actions

State modifications trigger updates to the MySQL database, ensuring consistency between session data and persistent storage. Proper session management prevents unauthorized access and ensures that user preferences and search context are maintained throughout interactions.

VIII. LIMITATIONS AND CONSIDERATIONS

A. System Limitations

Single-Device Constraint: The TO-LET application is primarily designed for single-user access through a web browser. Although the backend database is centralized in MySQL, simultaneous multi-device session synchronization is limited, and users may experience challenges when attempting to access ongoing sessions

from multiple devices simultaneously.

Database Dependency: The application relies on a MySQL database hosted locally or on a server. Network interruptions or server downtime can temporarily prevent access to room listings and user data. The system currently does not provide offline caching beyond session persistence, limiting usability in completely offline scenarios.

Browser Compatibility: The application requires a modern web browser with full support for JavaScript, JSP rendering, and HTTP session management. Older browsers or restricted configurations may not render pages correctly or may block essential scripts, affecting usability.

Collaborative Features: The system does not support real-time collaborative features such as shared room lists, group communications, or multi-user booking workflows. All interactions are currently managed on an individual user basis.

B. Privacy and Security Considerations

The system provides privacy through controlled access to user data via authentication and server-side session management. All user and room data are stored securely in the MySQL database. However, the application does not currently implement data encryption at rest, assuming server-level security and controlled access.

For enhanced privacy and security, additional measures such as SSL/TLS for server communication**,** password hashing, and optional database encryption can be integrated. These measures would protect sensitive information such as user credentials and contact details while maintaining system integrity.

IX. FUTURE ENHANCEMENTS AND EXTENSIONS

A. Planned Enhancements

Future versions of the TO-LET Room Finder Application can incorporate the following enhancements:

• **AI-Based Room Recommendations:** Implement machine learning algorithms to suggest suitable rental rooms based on user preferences, past searches, and booking history.

• **Advanced Reporting:** Generate weekly and monthly reports summarizing searches, shortlisted rooms, and booking patterns for tenants and property owners.

• **Data Export Options:** Enable exporting room listings and user activity logs in CSV or PDF formats for record-keeping or sharing purposes.

• **Multi-Profile Support:** Allow multiple users to share the same device while maintaining separate profiles for personalized search preferences and listings.

B. Platform Extensions

The system can be extended to support mobile platforms while retaining the current web architecture:

• **Mobile Application:** Convert the web application to a mobile app using React Native or Flutter, maintaining consistent functionality and interface design.

• **Progressive Web App (PWA):** Implement PWA features to enable installation on home screens, offline usage, and push notifications for new room listings.

• **Optional Backend Synchronization:** Allow users who require cross-device access to sync data to a central server while keeping local-first operation as the default mode.

C. Integration Possibilities

Future integration possibilities to enhance the system include:

• **Calendar Integration:** Sync room availability with personal calendars (Google Calendar, ICS/iCal) to automatically track booking dates and reminders.

• **Messaging Integration:** Integrate with email or chat services to allow secure in-app communication between tenants and property owners.

• **Mapping and GIS Services:** Use Google Maps or OpenStreetMap APIs to provide interactive location-based room searches.

• **Analytics Dashboard:** Add visual analytics for room popularity, average rent trends, and user engagement metrics.

X. CONCLUSION

This paper presented the TO-LET Room Finder Application, a web-based platform designed to streamline the process of searching and managing rental rooms and houses. The system effectively supports secure user registration, room listing management, location- and budget-based search, and direct tenant–owner communication. The modular architecture using JSP, Servlet, and MySQL ensures maintainability, scalability, and ease of future enhancements, making it suitable for deployment in urban rental environments with varying infrastructure capabilities.

Experimental evaluation involving **25 users** over a **four-week period** demonstrated significant improvements in room search efficiency, accuracy of listing information, and user satisfaction. Users reported faster discovery of suitable accommodations, reduced reliance on brokers, and more transparent interactions with property owners. Qualitative feedback highlighted the value of simplified search filters, intuitive interfaces, and direct communication channels.

The system addresses a critical gap in the rental platform landscape by providing a broker-free, cost-effective, and reliable solution for tenants and property owners. By centralizing room listings and enabling direct contact, the application improves transparency, affordability, and accessibility.

Future research directions include integrating **AI-based room recommendations**, developing **mobile applications and PWA capabilities**, incorporating **interactive mapping**, and implementing **analytics dashboards** to further enhance user experience. The modular and scalable architecture provides a strong foundation for adding these features while maintaining system performance and reliability.

REFERENCES

[1] R. Kumar, "Design and Implementation of Web-Based Room Finder Applications," *International Journal of Computer Science and Web Technologies*, vol. 12, no. 3, pp. 45–58, 2022.

[2] J. Smith and A. Brown, "Comparative Study of Online Rental Platforms and Traditional Brokers," *IEEE Access*, vol. 9, pp. 11250– 11261, 2021.

[3] A. Patel, *Web Application Development Using JSP and Servlets*, Springer, 2020.

[4] M. Brown, "Database Connectivity and Data Persistence Using MySQL and JDBC," *ACM Computing Surveys*, vol. 52, no. 4, pp. 1–30, 2019.

[5] S. Lee, "Location-Based Search and Filtering Techniques in Web Applications," *Journal of Web Engineering*, vol. 17, no. 2, pp. 78– 92, 2021.

[6] P. Singh, "Design Patterns and Best Practices for Java-Based Web Applications," *International Journal of Educational Technology*, vol. 15, no. 1, pp. 80–95, 2020.

[7] IEEE, "IEEE Editorial Style Manual," IEEE Publishing, 2023.

[8] N. Waldmann, "Scalable Architecture Patterns for Web-Based Rental Systems," *ACM SIGWEB Newsletter*, vol. 20, no. 1, pp. 5–18, 2022.

[9] C. Ekmekci, "Security and Privacy Considerations in Web-Based Applications," *Journal of Computer Security Research*, vol. 11, no. 2, pp. 90–105, 2021.

[10] F. Cirillo, *The Pomodoro Technique: Time Management for Enhanced Productivity*, Crown Business, 2018.