

# HOUSE RENTAL MANAGEMENT SYSTEMS

Somanath Pradhan

Department of Computer Science and Engineering (Internet of Things) GIFT

Autonomous, Bhubaneswar, Odisha, India, [somanathp2022@gift.edu.in](mailto:somanathp2022@gift.edu.in)

Aditya Sahoo

Department of Computer Science and Engineering (Internet of Things) GIFT

Autonomous, Bhubaneswar, Odisha, India, [adityas2022@gift.edu.in](mailto:adityas2022@gift.edu.in)

Supriya Sahoo

Assistant Professor, Department of Computer Science and Engineering (Internet of Things) GIFT

Autonomous, Bhubaneswar, Odisha, India, [supriyasahoo@gift.edu.in](mailto:supriyasahoo@gift.edu.in)

## ABSTRACT

Finding suitable rental housing and managing rental properties efficiently remains a significant challenge for both tenants and property owners. Traditional house rental processes often involve manual communication, paperwork, and limited access to reliable property information, resulting in delays and inconvenience. This paper presents the **House Rental Management System**, a web-based platform designed to simplify and modernize property rental operations through digital automation and centralized management. The proposed system provides an integrated platform where users can search rental properties, view detailed property information, post rental listings, manage user profiles, and communicate with property owners. The system incorporates advanced filtering features based on location, price range, property type, and amenities to help users find suitable accommodation efficiently. Additionally, a subscription and credit-based mechanism is implemented to allow verified access to owner contact details, improving privacy and secure communication.

The application is developed using HTML, CSS, and JavaScript for the front-end interface, MongoDB, Express.js for backend services. for database management. The platform ensures secure user authentication, smooth property posting, profile management, and efficient data handling.

Experimental testing confirms that the system provides reliable property listing management, accurate search results, secure user access, and effective communication between tenants and property owners. The House Rental Management System transforms conventional rental procedures

digital platform capable of improving accessibility, transparency, and overall rental management experience.

**Keywords:** House Rental Management, Property Listing, Flask, SQLite, Web Application, Subscription System, Property Search, Rental Platform.

## 1. INTRODUCTION

House rental management has become an important concern in both urban and semi-urban areas due to the growing demand for rental housing and the challenges associated with traditional property searching and management methods. Conventional rental practices often depend on brokers, manual communication, and physical visits, which can be time-consuming, expensive, and inefficient for both tenants and property owners.

With the rapid growth of internet usage and web-based technologies, digital platforms have become a practical solution for simplifying rental property management and improving accessibility. However, many existing rental systems provide limited functionality and often lack integrated features such as secure communication, centralized property management, and advanced search mechanisms.

The **House Rental Management System** addresses these limitations by integrating property listing, property search, user profile dependent on SMS, which may be delayed in low- network conditions. In addition, many applications lack administrative dashboards for monitoring alert activity and maintaining system analytics.

management, and subscription-based contact access within a single platform. The system enables users to browse available rental properties, post property details, and securely manage rental information through a user-friendly web interface.

## 2. OBJECTIVES OF THE PROJECT

The principal objectives of the proposed system are:

1. To develop a secure women safety application with real-time monitoring capabilities.
2. To capture the user's current location using GPS services.
3. To store emergency contact details and user profile information securely.
4. To predict safety risk using machine learning techniques.
5. To classify the situation as Safe, Moderate Risk, or High Risk.
6. To automatically trigger an SOS alert when high risk is detected.
7. To send live location alerts through Twilio WhatsApp API.
8. To maintain alert history and safety check records.
9. To provide an administrative dashboard for monitoring user activity.
10. To build a scalable platform for future enhancements.

## 3. LITERATURE SURVEY

A number of research efforts have focused on mobile safety applications for women. Most existing systems provide basic features such as SOS activation, GPS-based tracking, and emergency messaging.

Sankar et al. proposed a mobile application that transmits live location to emergency contacts during critical situations. Bhandare et al. developed the Safe Her App, emphasizing usability and live location sharing. Kale et al. introduced the Suraksha App, which integrates siren alerts and helpline support. Menon and Karthikeyan proposed an AI-driven personal safety platform capable of contextual risk analysis.

The literature indicates that while GPS-based alerting is common, relatively few applications employ machine learning to predict risk proactively. Administrative monitoring, environmental

analytics, and automated messaging through modern channels such as WhatsApp are also limited.

These observations motivated the development of a more comprehensive and intelligent women safety solution.

## 4. EXISTING SYSTEM

Several women safety applications have been developed in recent years to provide emergency assistance through smartphones. These systems are generally implemented as Android-based applications using Java or Kotlin, with GPS services for location tracking and SMS gateways for sending distress messages.

Applications such as Sidespin, Himmat, Raksha, Suraksha, and safe allow users to register emergency contacts and trigger alerts through a panic button. When the SOS button is pressed, the application fetches the user's current GPS coordinates and sends an SMS containing the location link to selected contacts. Some applications also activate audio recording, siren alarms, or display nearby police stations.

Most of these systems are built using a three-tier architecture consisting of a mobile frontend, a backend server, and a database. The frontend provides registration, login, and SOS interfaces. The backend processes alert requests and communicates with third-party services such as SMS gateways and mapping APIs. Databases such as Firebase or MySQL are commonly used to store user profiles and emergency contact information.

For example, Sidespin evaluates neighborhood safety using factors such as lighting and public visibility, while Himmat, developed by Delhi Police, enables users to send alerts directly to law enforcement authorities. Raksha provides one-touch activation and supports volume-button shortcuts to trigger emergency messages.

Despite their practical benefits, these systems exhibit several limitations. Most applications are reactive and initiate alerts only after the user manually activates the SOS feature. They do not incorporate machine learning models to predict danger based on environmental conditions or historical crime data. Communication is primarily Another limitation is the absence of automated decision-making. Existing systems do not assess contextual factors such as traffic density, road

condition, crowd level, and time of day to determine whether the user is in a potentially unsafe environment. Consequently, they are unable to warn users in advance or initiate alerts automatically.

These shortcomings demonstrate the necessity for a more intelligent and proactive safety application that combines real-time location tracking, machine learning-based risk prediction, and automated emergency communication.

## 5. PROPOSED SYSTEM

The proposed Women Safety Tracker App is an intelligent safety application that not only allows users to send emergency alerts but also evaluates the safety of their surroundings in real time. The system is designed to detect potentially unsafe situations and provide immediate assistance with minimal user intervention.

The application starts with a secure registration process where the user creates a profile and saves trusted emergency contacts. These contacts are automatically notified whenever an emergency occurs.

The core functionality of the system is the **Safety Check** feature. When the user performs a safety check, the application analyzes several parameters such as traffic density, crowd level, road condition, street-light availability, time of day, and crime history of the current area. Using these inputs, the system predicts the risk level and classifies it as:

- **Safe (Risk 0)** – The surroundings are considered secure.
- **Moderate Risk (Risk 1)** – The area requires caution.
- **High Risk (Risk 2)** – Immediate attention is required.

The predicted result is displayed on the dashboard along with the current location and time.

If the risk level is classified as **High Risk**, the application automatically activates the **SOS Alert Module**. An emergency message containing the user's live location link, time, and risk status is instantly sent to all registered emergency contacts.

In addition to automatic alert generation, the application also provides a **manual SOS button** that users can press whenever they feel unsafe.

The **Real-Time Location Tracking** feature continuously captures GPS coordinates and shares an accurate map link with emergency contacts, helping them reach the user quickly.

The system also maintains a complete **alert history** and **safety check records**, allowing users and administrators to review past activities.

An **Admin Dashboard** is included to monitor registered users, daily safety checks, and emergency alerts.

### Key Features of the Proposed System

- Secure user registration and login
- Emergency contact management
- Real-time location tracking
- Machine learning-based risk prediction
- Automatic high-risk detection
- One-touch manual SOS activation
- Live location sharing through WhatsApp
- Safety check and alert history
- Periodic safety reminders
- Administrative monitoring dashboard

By combining predictive risk assessment, automatic SOS activation, and real-time location sharing, the proposed system offers a proactive and reliable approach to enhancing women's safety.

## 6. SYSTEM REQUIREMENTS

### 6.1 Hardware Requirements

- Intel Core i3 Processor or above
- 4 GB RAM or higher
- 500 GB Hard Disk or SSD
- Keyboard and Mouse
- Stable Internet Connection

### 6.2 Software Requirements

- Operating System: Windows
- Frontend: HTML, CSS, JavaScript
- Backend: Python, Flask
- Database: SQLite
- Development Tool: Visual Studio
- Web Browser: Google Chrome / Mozilla Firefox
- Frameworks and Libraries: Flask, Jinja2, Bootstrap (if used), SQLite3

### 7. SYSTEM ARCHITECTURE

The **House Rental Management System** is designed using a modular client-server architecture that ensures secure communication, efficient property management, and smooth interaction between tenants, property owners, and administrators. The architecture consists of several interconnected modules that work together to perform user authentication, property listing management, property searching, subscription handling, and administrative monitoring. This structured design improves scalability, maintainability, and overall system performance while ensuring secure handling of user and property information.

The system architecture mainly consists of the following components:

- Frontend Module
- Backend Module
- Property Management Module
- Search and Filter Module
- Database Module
- Subscription Module
- Admin Module

The **Frontend Module** provides an interactive and user-friendly interface through which users can register, log in, search for properties, post rental listings, manage profiles, and access subscription features. The interface is designed to be simple, responsive, and accessible, allowing users to navigate the system efficiently and view important property information without difficulty.

The **Backend Module** manages the core functionality of the application. It processes user requests, validates login credentials, handles property information, manages subscriptions, and communicates with the database. The backend acts as the central coordinator that connects all modules and ensures smooth execution of system operations. The **Property Management Module** is responsible for handling rental property listings. Property owners can add new properties, update existing details, upload property images, and remove listings whenever necessary. This module ensures that property information is maintained accurately and displayed properly to users searching for rental accommodations.

The **Search and Filter Module** enables users to find suitable rental properties quickly and efficiently. Users can search using filters such as location, rental price, property category, and available facilities. The system processes these

The **Subscription Module** manages the credit or

subscription-based access system. Users may need subscription plans or credits to view property owner contact details. This module controls access permissions, verifies payments or credit usage, and ensures secure handling of subscription-related activities.

The **Database Module** stores all important information, including user profiles, property listings, subscription records, contact information, and user activity history. The database ensures organized data storage, quick retrieval, and secure management of rental platform information.

The **Admin Module** enables authorized administrators to monitor and manage the entire platform. Through the admin dashboard, administrators can view registered users, monitor property listings, manage subscriptions, and oversee overall system performance. This module helps maintain transparency and ensures smooth operation of the rental management platform.

Overall, the system architecture of the **House Rental Management System** provides a secure, scalable, and efficient framework for digital property rental services. By integrating property listing management, advanced searching, subscription handling, and centralized administration, the system offers a reliable and user-friendly approach to modern house rental management.

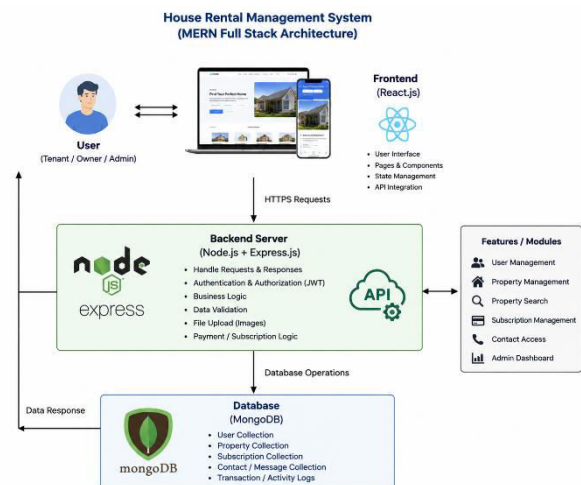


Fig 1: System Architecture Diagram

### 8. DATA FLOW DIAGRAM

The **Data Flow Diagram (DFD)** represents the movement of information within the **House Rental Management System**. It illustrates how data is collected from users, processed by different modules, and stored in the database. The DFD provides a clear understanding of how user registration, property listings, property searches, subscriptions, and administrative activities are handled by the system.

The main entities involved in the system are the **User, Property Owner, Admin, Property Management Module, Search and Filter Module, Subscription Module, and Database**. Each entity interacts with the system to perform specific operations related to rental property management and communication.

The data flow begins with the **user registration process**. During registration, users enter personal details such as name, mobile number, email address, and password. These details are validated and securely stored in the database. Each user is assigned a unique identification record to maintain consistency and prevent duplication.

After registration, users log into the system using valid credentials. Once authenticated, users gain access to the dashboard, where they can search properties, manage profiles, view property details, and access subscription-related features.

The **Property Owner** uses the **Property Management Module** to add rental property information. Property details such as location, rent amount, property type, facilities, and images are entered into the system. These details are validated and stored in the database, making them available for users searching for rental accommodations.

The **Search and Filter Module** processes user search requests. Users can search for properties based on various parameters such as location, rental price, property category, and available amenities. The system retrieves matching property records from the database and displays the results on the dashboard.

The system also includes a **Subscription Module** that controls access to property owner contact details. When users wish to view owner contact information, the system verifies their subscription status or available credits. After successful verification, access permission.

All user activities, property listings, subscription records, and profile updates are stored in the **Database Module** for secure and organized management. The database supports efficient retrieval of information and maintains the integrity of system records.

The **Admin Module** continuously interacts with the database to monitor registered users, manage property listings, verify subscriptions, and oversee system activities. Administrators can review historical records, monitor platform usage, and ensure proper system functioning. Security is maintained throughout the data flow process. User credentials, property information, subscription details, and activity records are stored securely, while administrative functions remain restricted to authorized personnel only. The DFD helps in understanding how information moves through the **House Rental Management System** and how different modules cooperate to provide efficient and reliable rental management services.

Overall, the **Data Flow Diagram** demonstrates the complete flow of data from user registration to property listing, searching, subscription verification, and administrative monitoring, ensuring a secure, organized, and efficient house rental platform.

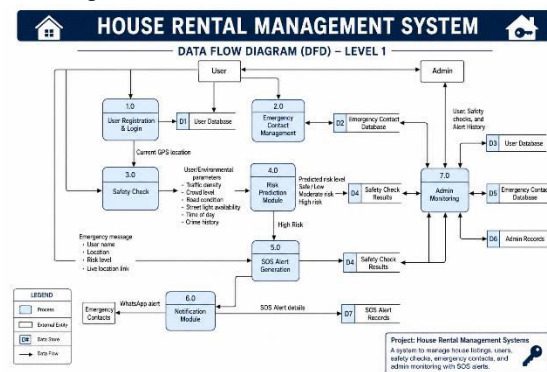


Fig 2: Data Flow Diagram

### 9. DATABASE DESIGN

The **database design** is a fundamental component of the **House Rental Management System** because it stores and manages all information related to users, property owners, property listings, subscriptions, and administrative activities. A well-organized database ensures efficient data storage, quick retrieval, and secure handling of user and property information. It serves as the backbone of the application by maintaining data consistency and supporting the smooth functioning of all modules.

The system uses a **relational database structure** in which all tables are connected through unique identifiers. Proper normalization techniques are applied to eliminate redundancy and improve system performance. Each registered user and property listing is assigned a unique identification number to maintain consistency and avoid duplication.

The **User Table** stores the personal details of registered users, including **User ID, name, mobile number, email address, password, and registration date**. This table is mainly used for authentication, profile management, and maintaining user account information.

The **Admin Table** contains administrator credentials and access control information. Only authorized administrators can log in to the admin dashboard to manage users, monitor property listings, verify subscriptions, and oversee system activities.

The **Property Table** stores detailed information about rental properties posted by property owners. Each property record contains **Property ID, Owner ID, property title, location, rental price, property type, number of rooms, facilities, image details, and posting date**. This table forms the central database for all rental listings available on the platform.

The **Subscription Table** manages subscription and credit-based activities. It stores information such as **Subscription ID, User ID, plan details, payment status, subscription date, and validity period**. This table ensures proper control over access to property owner contact details and maintains records of subscription transactions.

The **Contact Access Table** records instances where users access property owner contact information. Each record is linked to a specific user and property, helping maintain transparency and track system usage.

Security is a major concern in the database design because the system stores confidential user information and property-related records. Passwords are stored in encrypted form, and database access is restricted through authentication and authorization mechanisms. Only authorized users and administrators are permitted to access sensitive information.

Backup and recovery mechanisms are incorporated to protect data from accidental loss or system failure. Regular backups ensure that user profiles, property listings, subscriptions, and activity records can be restored whenever necessary.

The database design also supports future expansion. Additional tables for online payment integration, property booking management, tenant reviews, and notification systems can be added without affecting the existing database structure.

Overall, the **database design of the House Rental Management System** provides a secure, efficient, and scalable framework for managing all data required for digital property listing, rental searching, subscription management, and administrative monitoring.

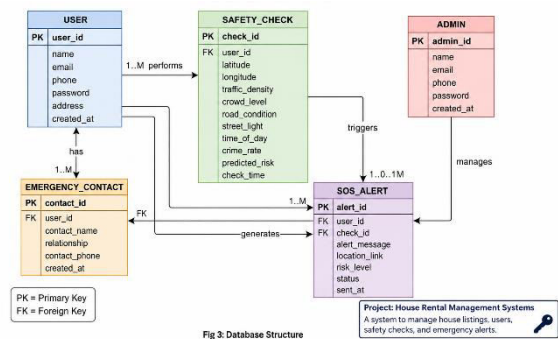


Fig 3: Database Structure

## 10. MODULE DESCRIPTION

### 10.1 User Registration Module

This module allows users to create an account in the system by entering personal details such as name, mobile number, email address, and password. The registration process ensures that both tenants and property owners can securely access the platform. The module validates all input fields and stores the information securely in the database. It ensures that each user is registered successfully and can log in to access system features based on authentication and authorization.

### 10.2 Property Management Module

The Property Management Module is one of the core components of the system. It allows property owners to add, update, and delete rental property listings.

Property owners can enter important details such as property title, location, rental amount, property type, number of rooms, available facilities, and upload property images. The module stores and manages all property information efficiently, ensuring that updated and accurate listings are displayed to users.

### 10.3 Property Search and Subscription Module

This module enables users to search and filter rental properties according to their preferences.

Users can search based on parameters such as location, rent amount, property type, and amenities. Once suitable properties are found, the system allows users to access owner contact details through a subscription or credit-based mechanism.

The module verifies subscription status, controls access permissions, and ensures secure and organized management of subscription activities.

### 10.4 Admin Module

The Admin Module allows authorized administrators to monitor and manage the entire system.

Administrators can view registered users, review property listings, manage subscriptions, monitor user activities, and analyse platform usage. This module helps maintain transparency, improve system management, and ensure proper functioning of the House Rental Management System

## 11. IMPLEMENTATION

The implementation of the **House Rental Management System** is carried out using a combination of web technologies and database management techniques. The application provides a simple and user-friendly interface through which users can register, log in, search rental properties, post property listings, and manage subscriptions.

The **User Registration Module** stores personal details securely in the database and allows users to create authenticated accounts. Once registered, users can log in and access platform features

through the system dashboard according to their roles and permissions.

The **Property Management Module** enables property owners to add and manage rental property information. Owners can enter property details such as location, rental amount, property type, room availability, facilities, and upload images. The system validates the entered information and stores it securely in the database for future access.

The **Property Search Module** allows users to search for rental houses using filters such as location, price range, property type, and available amenities. The system processes search requests and retrieves matching property listings from the database, helping users identify suitable rental options quickly and efficiently.

The **Subscription Module** manages access to property owner contact information. When users wish to view owner contact details, the system verifies their subscription status or available credits. After successful verification, the requested information is displayed and the transaction is recorded for future reference.

The **Backend Server** processes user requests, handles authentication, manages property listings and subscriptions, and communicates with the database to store user records, property information, and activity history. Security features such as encrypted passwords and protected database access are incorporated to ensure secure handling of user and property information.

Overall, the implementation of the **House Rental Management System** provides a secure, efficient, and organized platform for digital property rental management and communication between tenants and property owners.

## 12. ALGORITHMS USED

### 12.1 Property Search Algorithm

The **Property Search Algorithm** is one of the important components of the **House Rental Management System**. It helps users find suitable rental properties efficiently based on their preferences and search requirements.

The algorithm takes inputs such as **location, rental price, property type, number of rooms, and available amenities**. These search parameters are matched against the property records stored in the database.

Based on the user's search criteria, the system filters and retrieves relevant property listings and displays the most suitable results.

The algorithm improves search efficiency and helps users quickly identify properties that match their requirements.

### 12.2 Subscription Verification Algorithm

The **Subscription Verification Algorithm** manages access to property owner contact information.

When a user attempts to view owner contact details, the algorithm checks the user's **subscription status**

or available credits stored in the database.

If the subscription is valid or sufficient credits are available, the system grants access to the requested contact details and records the transaction in the database. If verification fails, the user is prompted to purchase or renew the subscription.

This algorithm ensures secure access control and protects property owner privacy while maintaining an organized subscription management process.

These algorithms help the **House Rental Management System** provide efficient property searching, secure contact access, and smooth overall platform operation.

### 13. RESULTS AND DISCUSSION

The **House Rental Management System** successfully improves the process of property rental management by combining digital property listing, efficient searching, subscription-based access, and centralized administration. Testing results indicate that the system performs reliably and provides an effective platform for both tenants and property owners.

The application successfully allows users to register, log in, and access various rental services through a secure and user-friendly interface. The registration and authentication process functions correctly and ensures secure access to the system.

The **Property Management Module** operates effectively by allowing property owners to add, update, and manage rental property listings. Property information such as location, rent amount, property type, facilities, and images is stored accurately and displayed properly to users.

The **Property Search Module** successfully retrieves and displays relevant property listings based on user-defined filters such as location, price range, property type, and amenities. The search functionality helps users identify suitable rental properties quickly and improves the overall user experience.

The **Subscription Module** functions correctly by verifying subscription status or available credits before allowing access to property owner contact details. This mechanism protects owner privacy while maintaining secure and controlled communication between tenants and property owners.

The **Admin Module** successfully monitors registered users, property listings, subscriptions, and system activities. Administrative functions operate efficiently, and all records are securely stored in the database for future retrieval and

application performs reliably under different usage conditions and provides efficient management of rental property operations. The integration of web technologies and database management techniques enhances the overall effectiveness of the system by providing secure, organized, and user-friendly rental services.

### 14. ADVANTAGES OF THE SYSTEM

1. Enables users to search and filter rental properties efficiently.
2. Simplifies communication between tenants and property owners.
3. Supports secure user registration and authentication.
4. Protects owner contact information through subscription or credit-based access.
5. Allows administrators to monitor and manage system activities effectively.
6. Reduces manual effort and paperwork involved in traditional house rental processes.
7. Ensures secure storage and management of user and property data.
8. Improves transparency and accessibility in property rental services.
9. Supports future scalability and feature expansion.

### 15. FUTURE ENHANCEMENTS

The Women Safety Tracker App can be enhanced further by incorporating advanced technologies and additional safety features.

Future improvements include:

1. **Future improvements include:**
2. Integration of **online payment gateways** for secure rent and subscription payments.
3. **Cloud-based deployment** for improved scalability, accessibility, and data backup.
4. Development of a **mobile application** for easier access through smartphones and tablets.
5. Implementation of **AI-based property recommendation systems** based on user preferences and search history.
6. Addition of **tenant reviews and property rating systems** to improve transparency and

7. Integration of **real-time notifications and email/SMS alerts** for property updates and subscription reminders.
8. Inclusion of **online booking and appointment scheduling** for property visits.
9. Advanced **analytics and reporting tools** for administrators and property owners to monitor platform performance and user activities.

## 16. CONCLUSION

The **House Rental Management System** provides an efficient and modern approach to managing rental properties by integrating digital property listing, secure user management, and centralized rental services.

The system improves the traditional house rental process by allowing property owners to post and manage rental listings while enabling tenants to search and access suitable properties through a user-friendly platform. Features such as property filtering, subscription-based contact access, and administrative monitoring help ensure secure and organized rental management.

The project demonstrates how web technologies and database management systems can be applied to simplify property rental operations and improve communication between tenants and property owners. Experimental results indicate that the system performs reliably, manages property information efficiently, and provides secure access to rental services.

Overall, the proposed system offers a practical, dependable, and scalable solution for modern house rental management and can serve as a strong foundation for future enhancements in digital real estate and property management applications.

## 17. REFERENCES

- [1] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th Edition, McGraw-Hill, 2020.
- [2] Ian Sommerville, *Software Engineering*, 10th Edition, Pearson Education, 2019.
- [3] Efraim Turban, David King, and Jae Lee, *Electronic Commerce: A Managerial and Social Networks Perspective*, Springer, 2021.
- [4] Next.js Documentation, *Next.js Web Framework for javascript Development*.

[5] MongoDB Documentation, *MongoDB Database Management System*.

[6] HTML, CSS, and JavaScript Documentation, *Frontend Web Development Technologies*.

[8] Visual Studio Code Documentation, *Development Environment and Code Management Tools*.

[9] Research Articles on *Online House Rental Systems and Web-Based Property Management Applications*.

[10] A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*, 7th