RV COLLEGE OF ENGINEERING®, BENGALURU-560059

(Autonomous Institution Affiliated to VTU, Belagavi)

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

****

TITLE OF THE PROJECT

Real Estate Management System

***Mini - Project Report***

***Submitted by***

**SKANDA PR USN 1RV22CS199**

**SUHAS RAJ HR USN 1RV22CS209**

**SHREERAM SB USN 1RV22CS188**

***in partial fulfillment for the requirement of 5th Semester DATABASE MANAGEMENT SYSTEMS (CD252IA)***

**Under the Guidance of**

Dr. Hemavathy. R

Professor

### Department of Computer Science and Engineering

**Academic Year 2024 - 2025**

**RV COLLEGE OF ENGINEERING®, BENGALURU 560059**

**(Autonomous Institution Affiliated to VTU, Belagavi)**

### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

****

**CERTIFICATE**

Certified that the project work titled **‘Real Estate Management System’** is carried out by Suhas Raj HR(1RV22CS209), Skanda PR(1RV22CS199), Shreeram SB(1RV22CS188), who are Bonafide students of R. V. College of Engineering, Bengaluru, in partial fulfillment of the curriculum requirement of 5th Semester **Database Management Systems (CD252IA)** Laboratory Mini Project during the academic year **2024-2025**. It is certified that all corrections/suggestions indicated for the internal Assessment have been incorporated in the report. The report has been approved as it satisfies the academic requirements in all respect laboratory mini-project work prescribed by the institution.

#### Signature of Faculty In-charge Head of Dept CSE Department

**External Examination**

**Name of Examiners Signature with date**

**1**

**2**

**Acknowledgement**

We are indebted to our Faculty (Theory)**, Dr. Hemavathy. R,** Professor**, Dept of CSE, RV College of Engineering** for her wholehearted support, suggestions and invaluable advice throughout our project work and helped in the preparation of this report.

We also express our gratitude to our theory Faculty (Lab) **Dr. Hemavathy. R**, Professor**,** Department of Computer Science and Engineering for their valuable comments and suggestions.

Our sincere thanks to **Dr. Shantha Rangaswamy,** Professor and Head, Department of Computer Science and Engineering, RVCE for her support and encouragement.

We express sincere gratitude to our beloved Principal, **Dr. K. N. Subramanya** for his appreciation towards this project work.

We thank all the **teaching staff and technical staff** of Computer Science and Engineering department, RVCE for their help.

**Abstract**

The Real Estate Management System is a comprehensive, database-driven application designed to streamline the complexities of property management. By integrating a robust relational database management system (RDBMS), this platform efficiently handles the storage, retrieval, and processing of extensive real estate data. It connects buyers, sellers, and real estate agents on a single centralized platform, ensuring seamless and secure management of property listings, transactions, and user interactions.

This project serves as a complete solution for real estate professionals and clients, offering functionalities such as property search and filtering by location, price, size, and amenities. Sellers and agents can list properties with detailed descriptions, images, and pricing information, while buyers can browse listings, express interest, and initiate transactions effortlessly. The system also features an admin panel for user management, property verification, and transaction tracking. Advanced SQL queries and normalization techniques are employed to maintain data integrity and optimize performance, ensuring that the platform remains scalable, secure, and efficient even as the volume of data grows.

One of the standout innovations in our project is the integration of an AI-powered chatbot developed using the Llama model. This chatbot provides real-time responses to user queries related to real estate and property details, enhancing user engagement and support. Additionally, we have implemented a machine learning model that predicts property prices based on factors such as location, property type (e.g., apartment or villa), and historical data from Bangalore. These innovative features not only streamline the property search process but also empower users with predictive insights, helping them make well-informed decisions in the dynamic real estate market.

**Table of Contents**

Page No.

|  |  |
| --- | --- |
| Acknowledgement | i |
| Abstract | ii |
| Table of Contents | iii |

List of Figures iv

1. Introduction 7-8
   1. Objective 8
   2. Scope 8
2. Software Requirement Specification 9-10
   1. Software Requirements. 9
   2. Hardware Requirements 9
   3. Functional Requirements 9,10
   4. Non-Functional Requirements 10
3. Entity Relationship Diagram for Real Estate Management System 11-13
4. Detailed Design 14-19
   1. DFD Level 0 for Real Estate Management System 16
   2. DFD Level 1 for Real Estate Management System 16,17
   3. DFD Level 2 for Real Estate Management System 18,19
5. Relational Schema and Normalization for Real Estate Management System 20-22
6. Forms, security, and Validation 23
7. Innovative Experiment 24
8. Conclusion 25
9. References 26
10. Appendix: Snapshots 27-31

**List of Figures**

|  |  |  |
| --- | --- | --- |
| Sl. No | Index | Page No. |
| 1. | Fig 3.1 Entity-Relationship (ER) Diagram for Real Estate Management System | 11 |
| 2. | Fig 4.1 DFD Level 0 for Real Estate Management System | 16 |
| 3. | Fig 4.2 DFD Level 1 of Admin for Real Estate Management System | 16 |
| 4. | Fig 4.3 DFD Level 1 of Owner for Real Estate Management System | 17 |
| 5. | Fig 4.4 DFD Level 1 for Real Estate Management System | 17 |
| 6. | Fig 4.5 DFD Level 2 of Property and Booking management for Real Estate Management System | 18 |
| 7. | Fig 4.6 DFD Level 2 of User and Account management for Real Estate Management System | 19 |
| 8. | Fig 5.1 Relational Database Schema for Real Estate Management System | 20 |
| 9. | Fig 5.2 Normalized Schema for Real Estate Management System | 22 |
| 10. | Fig 8.1 Website Interface | 27 |
| 11. | Fig 8.2 View Properties’ Details | 27 |
| 12. | Fig 8.3 Search Option | 28 |
| 13. | Fig 8.4 Login Portal | 28 |
| 14. | Fig 8.5 Owner Dashboard | 28 |
| 15. | Fig 8.6 Customer Dashboard | 29 |
| 16. | Fig 8.7 Admin Dashboard | 29 |
| 17. | Fig 8.8 Registration Portal | 30 |
| 18. | Fig 8.9 Chatbot | 30 |
| 19. | Fig 8.10 Real Estate Price Prediction Page | 31 |
| 20. | Fig 8.11 List of Tables in DB | 31 |
| 21. | Fig 8.12 Booking Table Contents | 31 |
| 22. | Fig 8.13 Property Table Contents | 31 |
| 23. | Fig 8.14 Users Table Content | 31 |

# INTRODUCTION

The **Real Estate Management System** is designed to simplify and automate the real estate transaction process by providing a comprehensive digital platform. Traditional real estate management involves multiple intermediaries, paperwork, and manual tracking, leading to inefficiencies and delays. This system aims to address these issues by leveraging database management technology, enabling real-time updates, and facilitating transparent communication among stakeholders. With urbanization and an increasing demand for real estate, managing property transactions has become more complex. Buyers and sellers need a reliable platform that ensures security, efficiency, and transparency. Additionally, real estate agents require a structured system to manage listings and transactions effectively. This project provides a robust database-driven solution to integrate and streamline these processes. The system will support features such as property listing, advanced search functionality, transaction tracking, and payment management. Users can register, list properties, negotiate deals, and finalize transactions within the system. Moreover, administrators can monitor and manage system operations to ensure compliance with industry standards.

By implementing this system, real estate professionals and customers can experience a seamless, efficient, and secure platform for conducting real estate transactions. This project highlights the importance of integrating technology into real estate management to enhance user experience and operational efficiency.

One of the standout innovations in our project is the integration of an AI-powered chatbot developed using the Llama model. This chatbot provides real-time responses to user queries related to real estate and property details, enhancing user engagement and support. Additionally, we have implemented a machine learning model that predicts property prices based on factors such as location, property type (e.g., apartment or villa), and historical data from Bangalore. These innovative features not only streamline the property search process but also empower users with predictive insights, helping them make well-informed decisions in the dynamic real estate market.

**1.1 Objective**

The primary objective of the **Real Estate Management System** is to develop a structured, database-driven platform that simplifies real estate operations. The system aims to eliminate the inefficiencies of traditional property transactions and provide a secure and user-friendly interface for all stakeholders.

Specific objectives include:

1. **Streamlining Property Listings**: Allow sellers and agents to list properties with detailed information, including images, pricing, and location.
2. **Enhancing Property Search and Filtering**: Enable buyers to search and filter properties based on multiple parameters, such as location, price range, property type, and amenities.
3. **User Management and Authentication**: Implement secure user registration, authentication, and role-based access control for buyers, sellers, and agents.
4. **Transaction Management**: Facilitate secure transactions between buyers and sellers, ensuring transparency and accountability.
5. **Payment Processing and Financial Tracking**: Integrate secure payment gateways for seamless financial transactions and maintain a record of all payments and invoices.
6. **Real-time Notifications and Alerts**: Notify users about new listings, price updates, pending transactions, and system changes.
7. **Reporting and Data Analytics**: Provide insightful reports on market trends, property demand, and transaction history to help users make informed decisions.
8. **Scalability and Security**: Ensure that the system can handle large volumes of data efficiently while maintaining high security standards to protect user information and transactions.

By achieving these objectives, the system will create a seamless, efficient, and reliable platform for real estate management.

**1.2 Scope**

The **Real Estate Management System** is designed to cater to multiple stakeholders, including buyers, sellers, agents, and administrators. The scope of the system covers various functional and non-functional aspects to ensure a comprehensive and efficient solution for managing real estate transactions.

**Functional Scope:**

1. **User Registration and Role-based Access**: The system will support multiple user roles, including buyers, sellers, real estate agents, and administrators. Each role will have specific permissions and functionalities.
2. **Property Listings**: Sellers and agents can add detailed property listings, including descriptions, images, location, and pricing.
3. **Advanced Search and Filtering**: Buyers can search for properties based on customized criteria, making it easier to find suitable options.
4. **Inquiry and Communication**: Users can contact sellers or agents through an integrated messaging system to ask questions and negotiate deals.
5. **Transaction Management**: The system will support offer submissions, price negotiations, and agreement finalization.
6. **Payment Gateway Integration**: Secure payment options will be included to facilitate down payments, installment tracking, and full payments.
7. **Notifications and Alerts**: Users will receive automated notifications about listing updates, new messages, and transaction status changes.
8. **Reporting and Analytics**: The system will generate reports on sales, user activities, and market trends to help users make data-driven decisions.

# SOFTWARE REQUIREMENT SPECIFICATIONS

* 1. **Software Requirements**
* **Operating System:** Windows, macOS, or Linux
* **Database Management System (DBMS):** MySQL, PostgreSQL, or MongoDB
* **Backend Technologies:** Python (Flask/Django) or Node.js (Express)
* **Frontend Technologies:** HTML, CSS, JavaScript (React, Angular, or Vue.js)
* **Server:** Apache, Nginx, or Node.js-based server
* **Development Tools:** Visual Studio Code, PyCharm, or IntelliJ IDEA
* **Other Dependencies:** Bootstrap, jQuery, API integrations for maps (Google Maps API, HERE Maps)

**2.2 Hardware Requirements**

* **For Development:**
  + Processor: Intel Core i5 (or equivalent)
  + RAM: 8GB (minimum), 16GB (recommended)
  + Storage: 500GB SSD (recommended for faster performance)
  + Internet Connection: Required for database and API access
* **For Deployment (Server Requirements):**
  + Processor: Intel Xeon or AMD Ryzen (multi-core)
  + RAM: 16GB or higher
  + Storage: 1TB SSD
  + Bandwidth: High-speed internet for handling multiple concurrent users
  + Cloud Hosting: AWS, Azure, or DigitalOcean for scalability

**2.3 Functional Requirements**

* **User Management:**
  + User registration and authentication (Buyers, Sellers, Agents, and Admins)
  + Role-based access control (RBAC)
* **Property Management:**
  + Add, update, and delete property listings
  + Upload property images and videos
  + Advanced search and filtering options
* **Transaction Management:**
  + Buyers can send purchase requests
  + Sellers can accept/reject offers
  + Contract finalization and document handling
* **Payment and Billing:**
  + Integration with secure payment gateways (Stripe, PayPal)
  + Transaction history and invoice generation
* **Notifications and Alerts:**
  + Email/SMS notifications for new listings, transactions, and messages
* **Admin Controls:**
  + Dashboard for monitoring activities
  + Verification of property listings
  + User management and fraud detection

**2.4 Non-Functional Requirements**

* **Scalability:**
  + The system should handle increased users and transactions efficiently.
* **Security:**
  + Implement encryption, authentication, and access controls to protect user data.
* **Performance:**
  + Page load times should be under 2 seconds, and queries should be optimized for speed.
* **Reliability:**
  + The system should have 99.9% uptime and include backup mechanism.
* **Usability:**
  + User-friendly UI with an intuitive interface for seamless navigation.
* **Compliance:**
  + Ensure adherence to real estate regulations and data protection laws (GDPR, CCPA).

# ER Diagram

# 

# Preamble*:* The ER Diagram is the cornerstone of our database design for the Real Estate Management System. It meticulously outlines the structural blueprint that connects various data entities through defined relationships and attributes. This visual representation clarifies how critical components such as Property, User, Transaction, and other entities interrelate. It highlights essential aspects like cardinality, constraints, and dependencies, ensuring that the design is both robust and scalable. By mapping these relationships, the diagram aids in maintaining data integrity and consistency throughout the system. Its clarity also fosters improved communication among developers, database administrators, and stakeholders. Overall, the ER Diagram lays a solid foundation for efficient data management and system performance.

#### Fig 3.1 Entity-Relationship (ER) Diagram for Real Estate Management System

ER diagram has 8 entities namely Testimonial, Owner, Loans, Registration, Property, Cancellation, Customer and Transaction records.

### Entities and Attributes:

1. **Customer**
   * Customer ID (Primary Key)
   * First Name
   * Middle Name
   * Last Name
   * DOB
   * Email ID
   * Phone Number
   * Occupation
   * Address
   * Customer Password
2. **Owner**
   * Owner ID (Primary Key)
   * First Name
   * Middle Name
   * Last Name
   * Email ID
   * Phone Number
   * Owner Password
3. **Property**
   * Property ID (Primary Key)
   * Property Name
   * Property Type
   * Property Status
   * Location
   * Price
   * Area
   * BHK
   * Date
4. **Registration**
   * Registration ID (Primary Key)
   * Registration Date
   * Registration Status
   * Reference Number
5. **Cancellation**
   * Cancellation ID (Primary Key)
   * Cancellation Date
   * Amount Refunded
   * Reference Number
6. **Transaction**
   * Transaction ID (Primary Key)
   * Transaction Date
   * Transaction Mode
   * Amount Paid
   * Transaction Status
   * Bank Name
7. **Loans**
   * Loan ID (Primary Key)
   * Bank Name
   * Rate of Interest
   * Processing Fee
   * Tenure
   * Maximum Loan Amount
   * Number of Installments
8. **Testimonial**
   * Testimonial ID (Primary Key)
   * Customer Satisfaction
   * Customer Description
   * Site Link

**Relationships:**

**1. One-to-One (1:1) Relationships**

* **Property - Registration** → A Property is linked to one Registration, and each Registration corresponds to one Property.
* **Transaction - Registration** → A Transaction is related to one Registration, and each Registration has one associated Transaction.

**2. One-to-Many (1:N) Relationships**

* **Owner - Property** → An Owner can own multiple Properties, but each Property belongs to a single Owner.
* **Customer - Property** → A Customer can register for multiple Properties, but each Property can have only one Customer associated with it at a time.
* **Customer - Cancellation** → A Customer can cancel multiple Property registrations, but each Cancellation is linked to only one Customer.
* **Customer - Transaction** → A Customer can perform multiple Transactions, but each Transaction belongs to a single Customer.
* **Customer - Testimonial** → A Customer can write multiple Testimonials, but each Testimonial belongs to a single Customer.

**3. Many-to-Many (M:N) Relationships**

* **Customer - Loans** → A Customer can apply for multiple Loans, and each Loan can be taken by multiple Customers.
* **Property - Loans →** A Property can have multiple Loans taken against it, and a Loan can be associated with multiple Properties.

## Data Flow Diagram

## Preamble:

## The Data Flow Diagram (DFD) is an essential tool that illustrates the dynamic movement of data within the Real Estate Management System. It offers a detailed view of how information flows between external entities and internal processes, ensuring every operation is accounted for. The DFD captures interactions among users (like Admin, Customer, and Owner) and system processes such as property management, user authentication, and booking. It provides insight into the continuous flow of data—from input through processing to storage—highlighting key operations at each level. This visualization helps identify potential bottlenecks and areas for process optimization. It also supports a deeper understanding of the system’s operational workflow. Ultimately, the DFD serves as a roadmap to ensure that all functional components work seamlessly together.

**1.Actors and External Entities**

* **Customers**
  + Register and log in to the system.
  + View, edit, or delete account details.
  + Book, confirm, or cancel property bookings.
* **Owners**
  + Register and log in to the system.
  + View, edit, or delete account details.
  + Add, edit, view, or delete properties.
* **Admins**
  + Manage user accounts (approve, add, edit, delete users).
  + Manage properties (add, edit, delete properties).

**2. Processes**

* **User and Account Management**
  + Customers, Owners, and Admins register and log in.
  + Users can edit or delete their accounts.
  + Admins can manage user approvals.
* **Property Management**
  + Owners and Admins can add, edit, view, and delete properties.
  + Customers can view listed properties.
* **Booking Management**
  + Customers can book, confirm, or cancel property bookings.
  + Booking details are stored and updated accordingly.

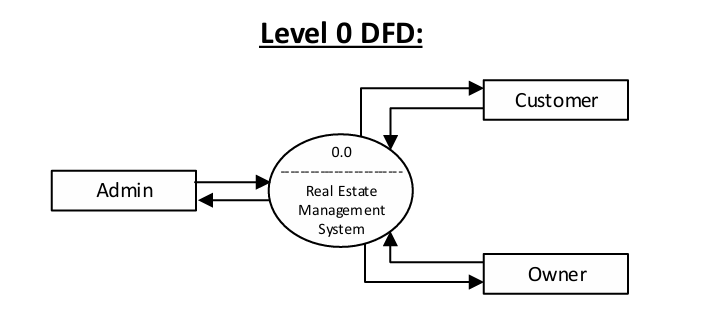
**3. Data Stores**

* **User Data** – Stores details of customers, owners, and admins, including names, contact details, login credentials, and roles.
* **Property Data** – Stores property details such as property name, type, location, area, price, BHK, and owner information.
* **Booking Data** – Stores booking records including customer details, booked property, booking date, status (confirmed/canceled), and payment reference.
* **Transaction Data** – Stores financial transactions related to property bookings, including transaction ID, amount paid, payment method, and status.
* **Registration Data** – Stores registration details of customers and properties, including registration ID, date, and approval status.

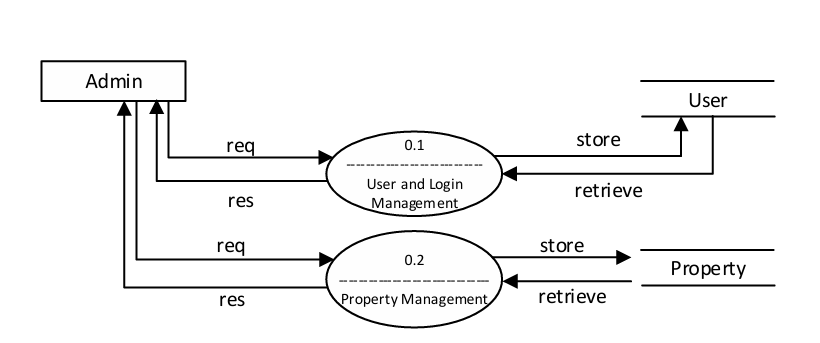
**4.Summarized Description**

* Level 0 offers a high-level overview, showing the primary external entities and their direct interactions with the system.
* Level 1 decomposes the system into specific modules such as User and Login Management, Property Management, and Booking Management, detailing operations like adding, editing, reading, and deleting records.
* Level 2 dives deeper into the internal processes, emphasizing the detailed flow of data through various retrieval and storage mechanisms. This layered approach highlights how user actions are systematically captured, processed, and recorded, ensuring operational efficiency and data integrity throughout the system.

## DFD Level 0

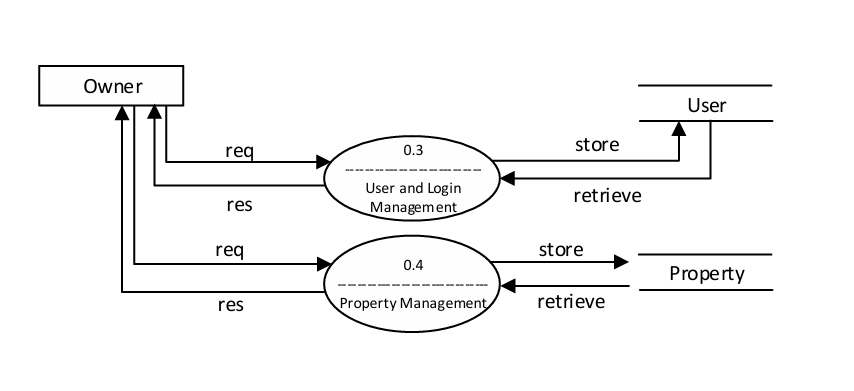
****

**Fig 4.1 DFD Level 0 for Real Estate Management System**

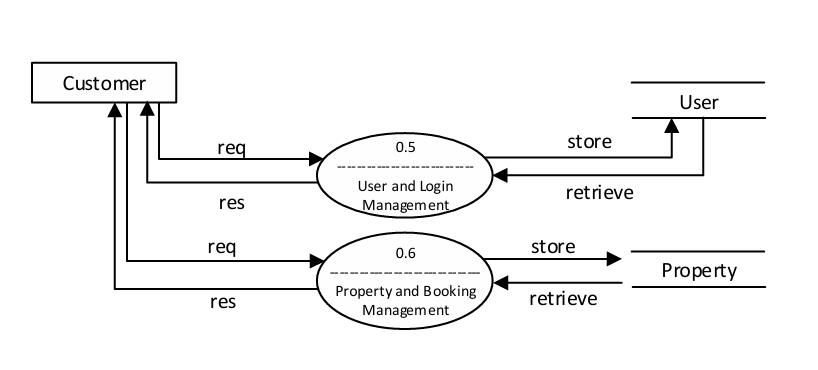
****

**Fig 4.2 DFD Level 1 of Admin for Real Estate Management System**

## DFD Level 1

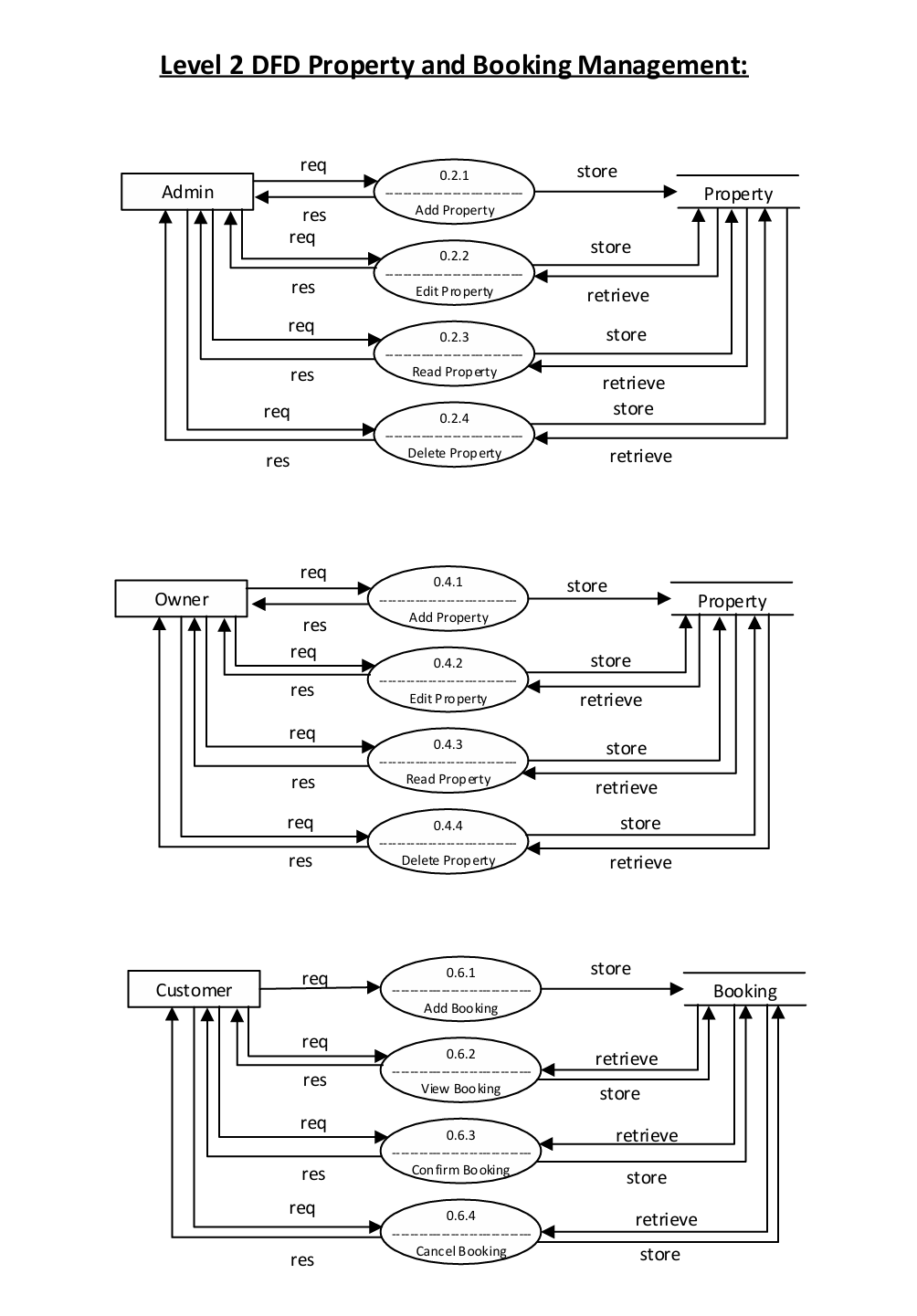
****

**Fig 4.3 DFD Level 1 of Owner for Real Estate Management System**

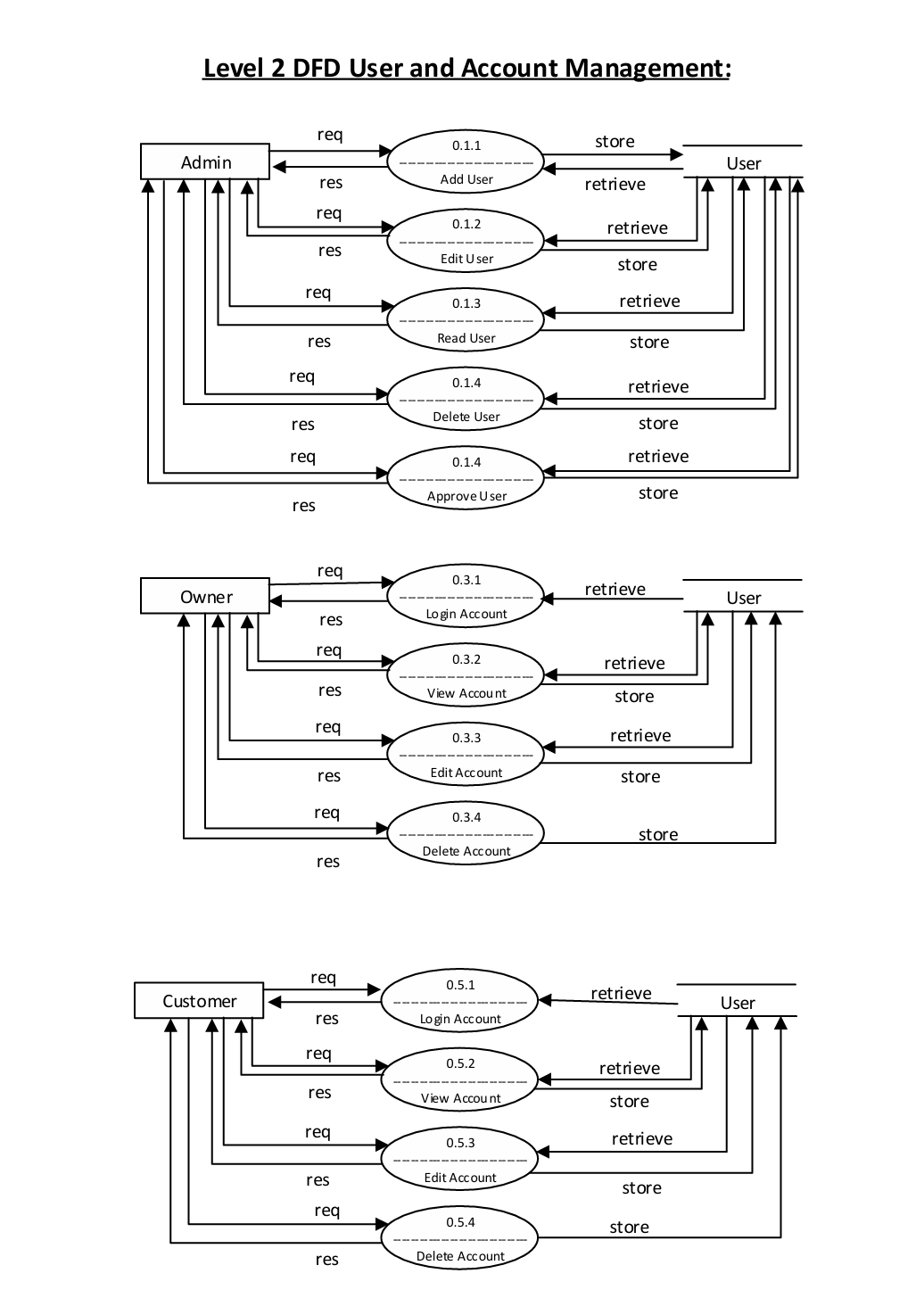
****

**Fig 4.4 DFD Level 1 for Real Estate Management System**

## DFD Level 2

****

**Fig 4.5 DFD Level 2 of Property and Booking management for Real Estate Management System**

****

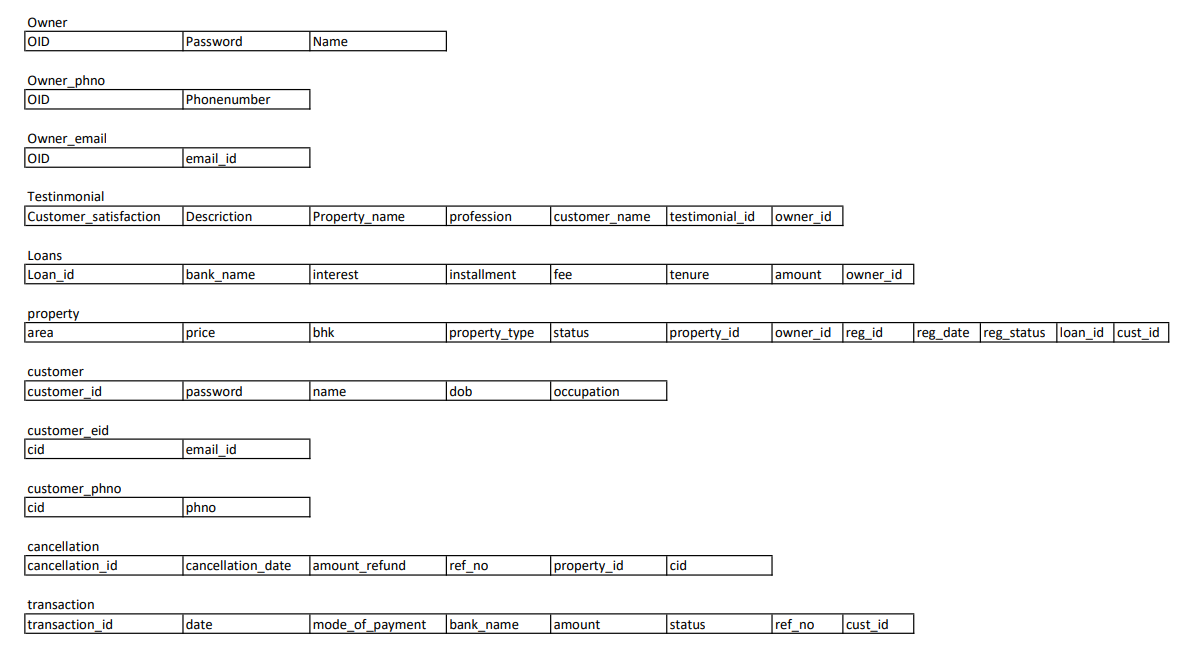
**Fig 4.6 DFD Level 2 of User and Account management for Real Estate Management System**

## Relational database structure (Schema) and Normalization

## Preamble:

## It outlines the Relational database structure and normalized structure of the Real Estate Management System’s database, showcasing a design that applies fundamental normalization principles to ensure data consistency and integrity. The schema is methodically organized into distinct tables to reduce redundancy and prevent update anomalies, ultimately enhancing system performance. By segregating related data into specialized tables, each handling a single concept—such as customer details, property information, transactions, and loan management—the design fosters modularity and scalability. The separation also aids in clear data management and improves query efficiency by localizing information to the most relevant tables. With referential integrity maintained through well-defined keys and relationships, the structure lays a robust foundation for an efficient real estate management platform. This approach not only streamlines data retrieval and updates but also aligns with best practices in database design, ensuring that the system remains agile and responsive to evolving needs.

**Diagram:**

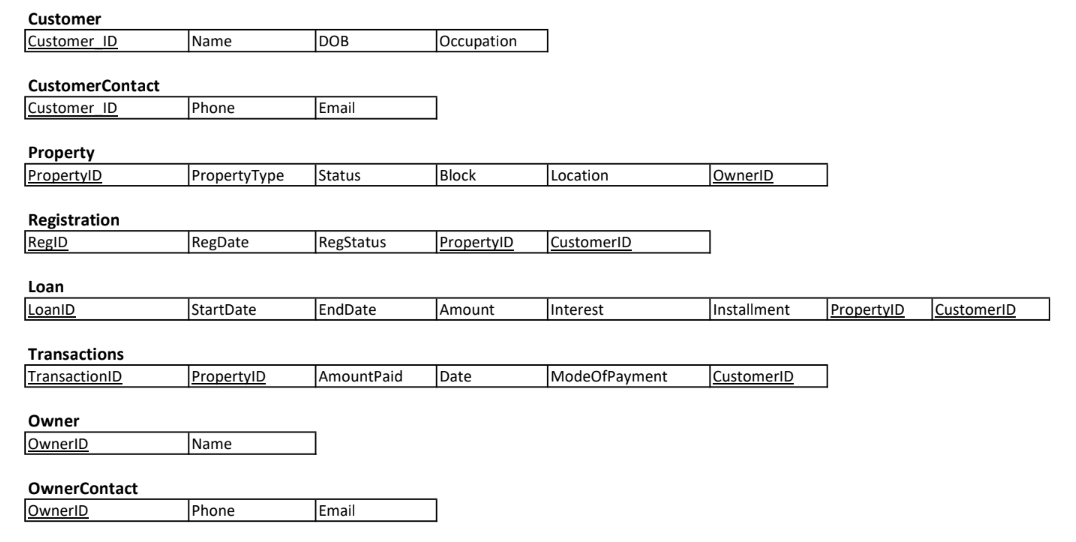
****

#### Fig 5.1 Relational Database Schema for Real Estate Management System

The relational schema for the project represents a structured database design for a real estate management system. It consists of multiple interconnected tables, each storing relevant information about property owners, customers, properties, transactions, loans, testimonials, and cancellations.

1. **Owner Table**: Stores details of property owners, including their unique ID (OID), password, and name. Additional contact details such as phone numbers and email IDs are stored in separate tables (Owner\_phno and Owner\_email) linked via OID.
2. **Customer Table**: Maintains records of customers with fields such as customer ID (customer\_id), password, name, date of birth (dob), and occupation. Contact information, including email and phone numbers, is stored in customer\_eid and customer\_phno tables, linked via customer ID (cid).
3. **Property Table**: Stores details of properties, including area, price, number of bedrooms (bhk), property type, and status. It is linked to owners, customers, and loans via owner\_id, cust\_id, and loan\_id. Additional registration details such as registration ID, date, and status are also included.
4. **Loans Table**: Captures loan-related details such as loan\_id, bank name, interest rate, installment amount, tenure, fee, and total amount. It is linked to the respective property owner via owner\_id.
5. **Transaction Table**: Maintains financial transactions, including transaction\_id, date, mode of payment, bank name, amount, status, reference number, and associated customer ID (cust\_id).
6. **Testimonial Table**: Records customer feedback with fields for customer satisfaction level, description, property name, profession, customer name, and owner\_id. Each testimonial is uniquely identified by testimonial\_id.
7. **Cancellation Table**: Stores information about property cancellations, including cancellation\_id, date, amount refunded, reference number, associated property\_id, and customer ID (cid).

## Normalized Schema



#### Fig 5.2 Normalized Schema for Real Estate Management System

**First Normal Form:**

A relation is in 1NF if values in each column are atomic, and there are no repeating groups. In our tables, there are no columns which contain multiple values or non-atomic values.

**Second Normal Form:**

A relation is in 2NF if it in 1NF, and all non-prime attributes are fully functionally dependant on primary key.

In our relation, for the Property table, it has columns of Registration related. So composite primary key will be Property\_id and Reg\_id, but the attributes like reg\_date, reg\_status depends only on reg\_id, and not on Property\_id. Also, Property\_type and location depend on Property\_id, but not on Reg\_id.

So, these two relations are split into separate relations, Property relation and Registration relation.

**Third Normal Form:**

A relation is in 3NF if it’s in 2NF, and there are no transitive dependencies.

If any non-key attribute depends on another non-key attribute rather than directly on primary key, then the relation does not meet 3NF.

In our relation, for customer relation, Phone and Email depends on Name attribute, which is not primary key. So, we split it into customer and customer contact relations.

The same for owner relation is followed as well.

1. **Forms, Security, and Validation**

Our Real Estate Management System incorporates robust form validation and security measures to ensure a safe and user-friendly experience for all users.

**6.1 Form Validation**

To enhance data integrity and user experience, we have implemented stringent validation checks on key forms, particularly the Login and Register pages:

**6.2 Email Validation**

The email field ensures users enter a valid email format (e.g., @gmail.com, @yahoo.com, etc.), preventing incorrect or fake email entries.

**Password Strength Requirements:**

Passwords must meet high-security criteria to prevent unauthorized access. The requirements include:

* Minimum 10 characters
* At least one uppercase letter
* At least one special character (e.g., @, #, $)
* A mix of letters and numbers

These validations are enforced both on the client side (using JavaScript) and server side (via Flask) to prevent bypassing the security measures.

**6.3 Security Measures**

Our platform integrates multiple security features to protect user data and ensure secure authentication:

Password Encryption & Secure Storage:

* User passwords are encrypted using bcrypt hashing before being stored in the database.
* This ensures that even if the database is compromised, raw passwords remain protected.

Decryption for Authentication:

* When a user logs in, the entered password is hashed and compared with the stored hash to authenticate the user securely.
* The system never stores plain-text passwords, minimizing the risk of data breaches.

SQL Injection Prevention:

All form inputs are sanitized, and parameterized queries are used to prevent SQL injection attacks.

Cross-Site Scripting (XSS) and CSRF Protection:

Input fields are validated and escaped to prevent malicious script execution.

CSRF tokens are implemented to prevent unauthorized form submissions.

By incorporating these validation and security mechanisms, our platform ensures a safe, reliable, and efficient user authentication process while protecting sensitive data.

## Innovative Experiment

## Our Real Estate Management System incorporates several innovative features to enhance performance, user experience, and predictive analytics. These optimizations make the platform faster, smarter, and more efficient in handling large-scale real estate data.

## Database Indexing for Faster Data Retrieval

## To improve the efficiency of search and retrieval operations, we have implemented indexing on key database fields:

## Property Name & Price Indexing:

## By creating indexes on the name and price columns of the property table, search queries execute significantly faster.

## This ensures that properties are quickly fetched and displayed on the Home Screen, even as the dataset grows.

## User Email & Password Indexing:

## We have indexed the email field to speed up user authentication.

## The hashed passwords are also indexed for optimized verification, improving login performance while maintaining security.

## These indexing strategies reduce query execution time and enhance system responsiveness, especially for large databases with numerous property listings.

## AI-Powered Chatbot for Real Estate Assistance

## To provide real-time assistance and improve user engagement, we have integrated an AI-powered chatbot developed using the Llama model. This chatbot:

## Answers user queries about property details, pricing, and locations.

## Provides recommendations based on user preferences.

## Improves customer support, reducing reliance on manual assistance.

## By leveraging Natural Language Processing (NLP), the chatbot enhances the overall user experience by delivering instant, intelligent responses.

## Machine Learning-Based Property Price Prediction

## Our platform integrates a machine learning model that predicts property prices based on:

## Location

## Property type (apartment, villa, etc.)

## Historical pricing trends

## By analysing market data, this model provides users with predictive insights, helping them make informed decisions in real estate transactions. Buyers can estimate property values, while sellers can determine competitive pricing.

## Additionally, we have deployed our Real Estate Management System using ngrok, enabling secure and instant public access to our locally hosted application. This deployment method allows users to interact with the platform without requiring complex server configurations. By leveraging ngrok, we can securely expose our Flask application to the internet for testing and demonstration purposes. This setup ensures seamless accessibility while maintaining security and performance.

## These innovative implementations—database indexing for optimized performance, an AI chatbot for user interaction, and predictive analytics for real estate pricing—make our system a cutting-edge solution in the real estate domain.

## Conclusion

The **Real Estate Management System** is a comprehensive, database-driven application designed to enhance and streamline the process of buying, selling, and managing real estate properties. In today's fast-paced and highly competitive real estate industry, technology plays a crucial role in ensuring efficiency, transparency, and security. This system is developed to address the limitations of traditional property management processes, which often involve excessive paperwork, manual coordination, and a lack of centralized data management. By leveraging advanced database technologies, user-friendly web interfaces, and secure transaction mechanisms, this system provides a reliable solution that benefits all stakeholders involved in real estate transactions.

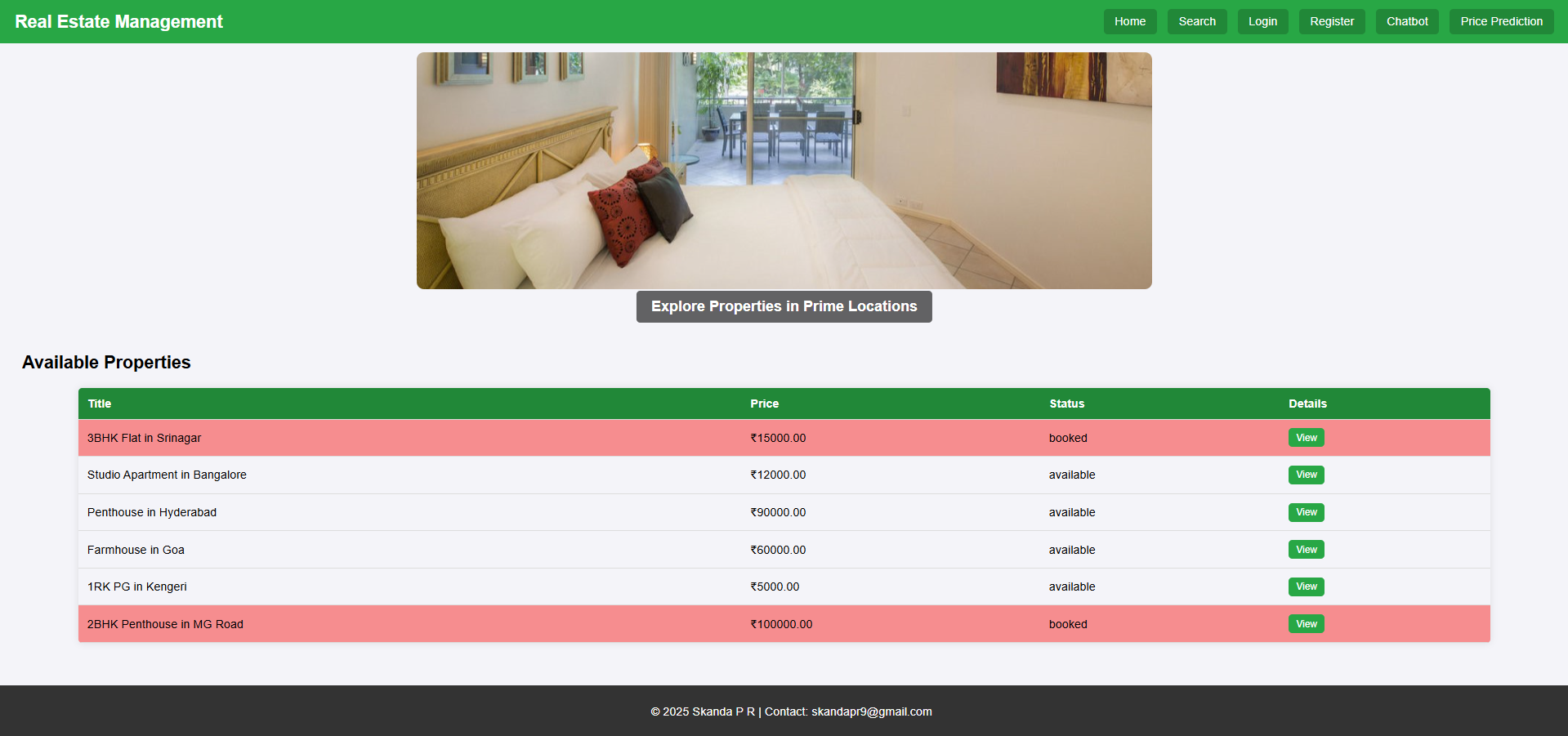
One of the most significant advantages of this system is its ability to **centralize property listings and user interactions**. In traditional real estate transactions, buyers and sellers often rely on physical visits, third-party brokers, and classified advertisements to connect and negotiate deals. This method is not only time-consuming but also lacks transparency. With the Real Estate Management System, all property listings are maintained in a structured database, making it easy for buyers to browse available properties based on specific filters such as location, price, size, and amenities. Sellers and real estate agents can list properties with detailed descriptions, images, and pricing information, ensuring that potential buyers have all the necessary details at their fingertips.

Another critical feature of the system is **role-based access control**, which ensures that different users have specific permissions based on their roles. Buyers can search for and inquire about properties, sellers can list and update property details, and real estate agents can facilitate transactions. Administrators oversee the entire platform, verifying property listings, managing user access, and ensuring compliance with legal regulations. This structured approach prevents unauthorized access and enhances the overall security of the system.

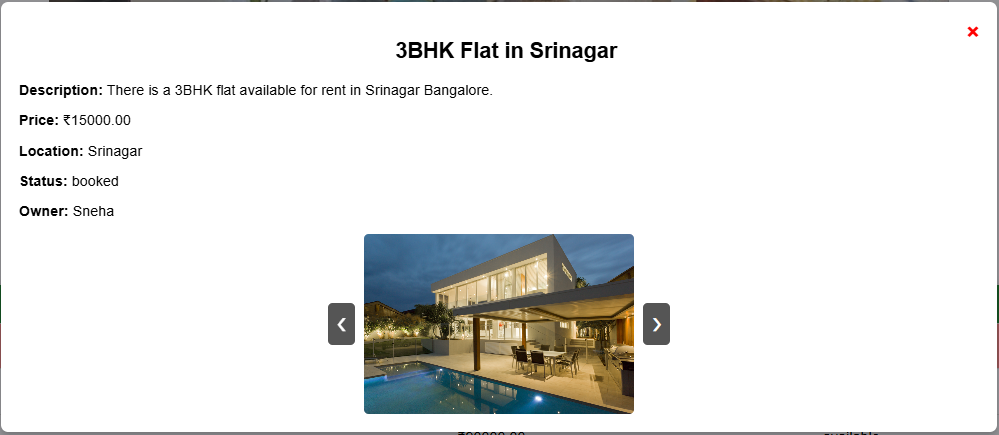
## References

* + 1. Kumar, R., & Sharma, P. A database approach to real estate management using SQL and NoSQL. *International Journal of Computer Applications*, 2018.
    2. Smith, J. R., & Gupta, A. The role of information technology in modern real estate transactions. *Journal of Real Estate Research*, 2019.
    3. Anderson, B., & Lee, K. Enhancing property listing efficiency through data-driven algorithms. *IEEE Transactions on Smart Cities*, 2020.
    4. Choudhury, S., & Banerjee, A. Blockchain for real estate: A new paradigm for secure transactions. *Elsevier Computers & Security*, 2021.
    5. Martin, D., & Fernandez, C. The evolution of digital real estate marketplaces. *Harvard Business Review*, 2016.
    6. Williams, T., & Zhao, Y. Implementing cloud-based property management systems. *Springer International Journal of Information Systems*, 2020.
    7. Patel, R., & Khan, A. Analyzing the impact of AI on real estate valuation and predictions. *Journal of Artificial Intelligence and Real Estate*, 2022.
    8. Roberts, E., & Lang, M. User experience design in online property management systems. *ACM Transactions on Human-Computer Interaction*, 2018.
    9. Brown, H., & Cooper, S. Big Data applications in real estate analytics. *Wiley Journal of Data Science*, 2019.
    10. Lin, Y., & Thompson, J. Mobile applications in property search and their effects on market efficiency. *Journal of Mobile Technology in Real Estate*, 2021.
    11. Singh, V., & Raj, P. The impact of digitalization on real estate customer behavior. *Elsevier Journal of Digital Business Trends*, 2020.
    12. Turner, L., & White, R. Integrating Geographic Information Systems (GIS) for smarter real estate decisions. *International Journal of Spatial Data Management*, 2017.
    13. Scott, P., & Evans, N. Role-based access control in real estate databases. *IEEE Journal of Database Security*, 2019.
    14. Harris, J., & Nelson, G. Cloud computing and database management for large-scale property systems. *Springer Journal of Cloud Computing Applications*, 2022.

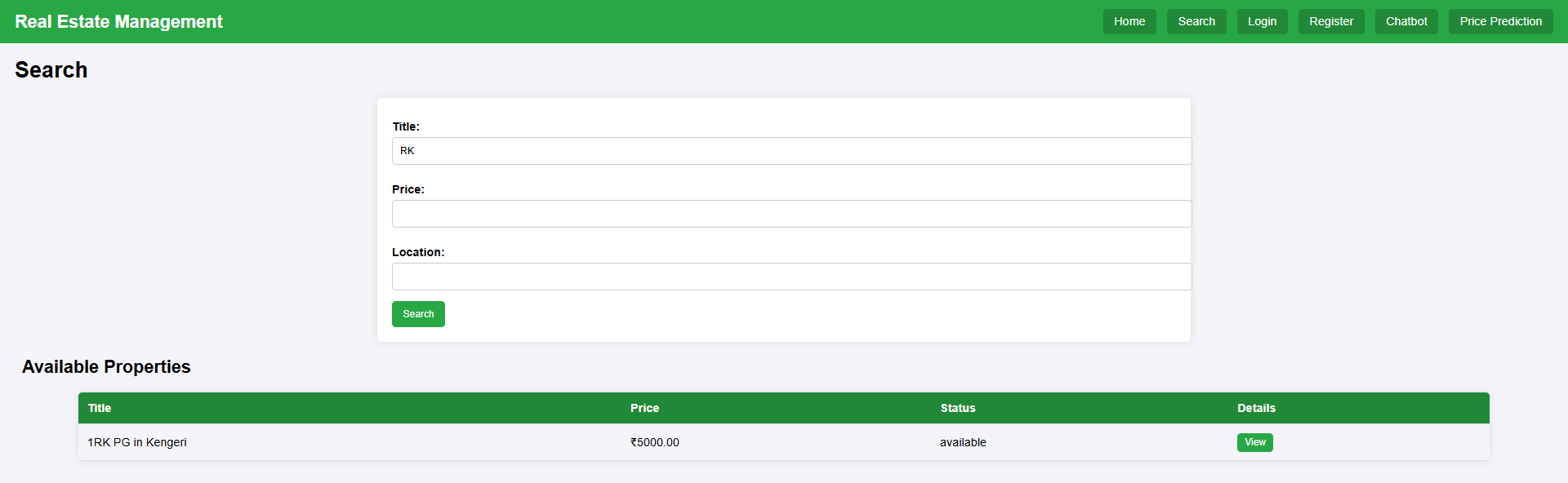
## Appendix: Snapshots

****

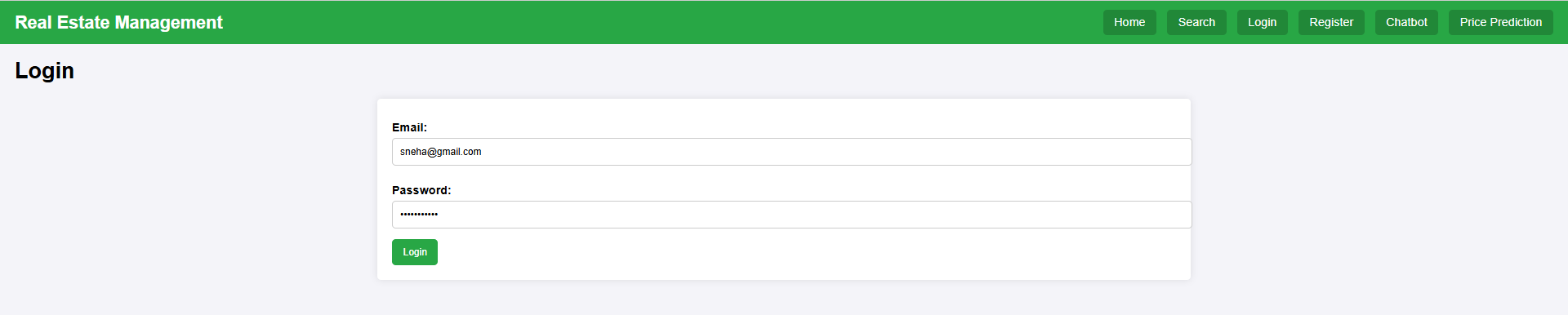
**Fig 8.1 Website Interface**

****

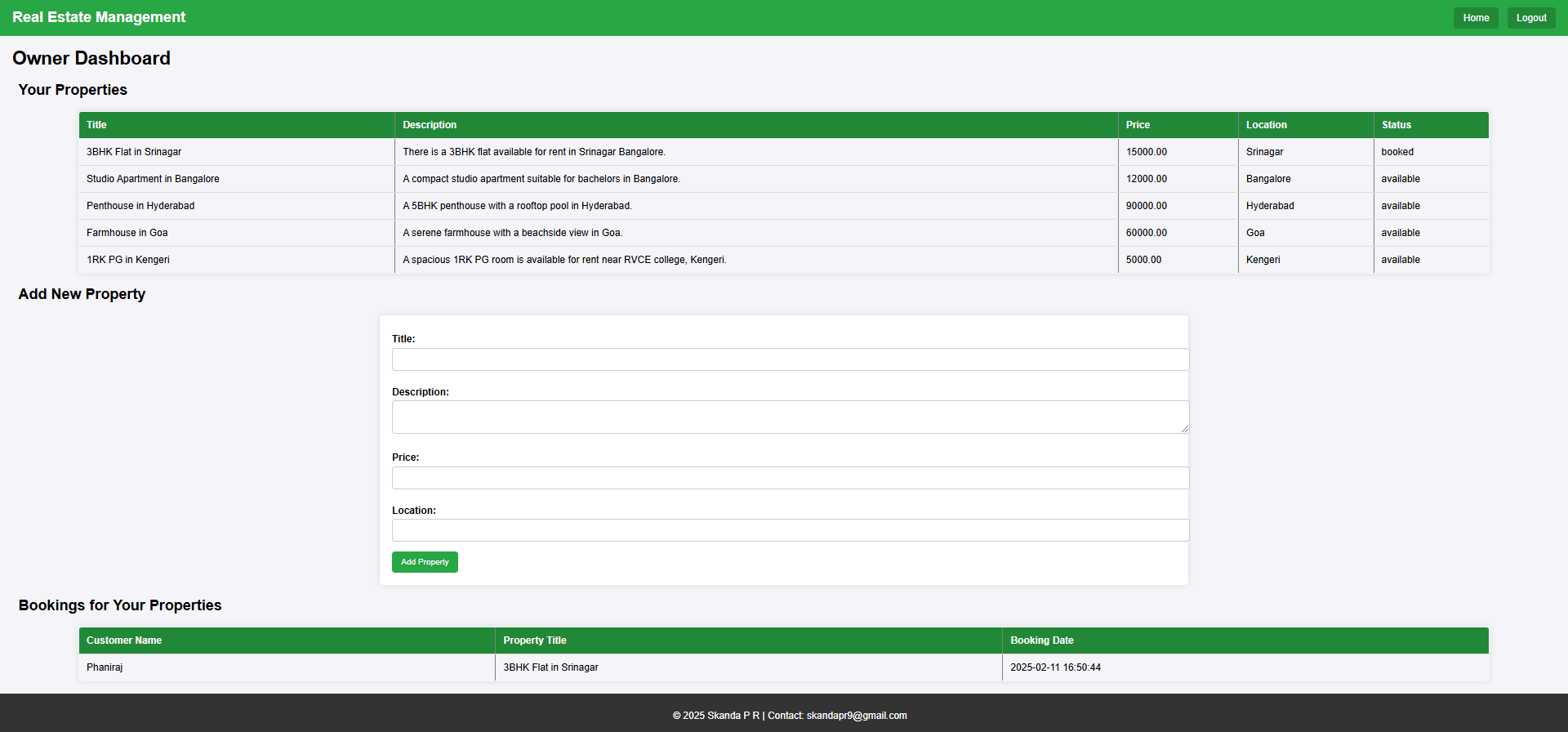
**Fig 8.2 View Properties’ Details**



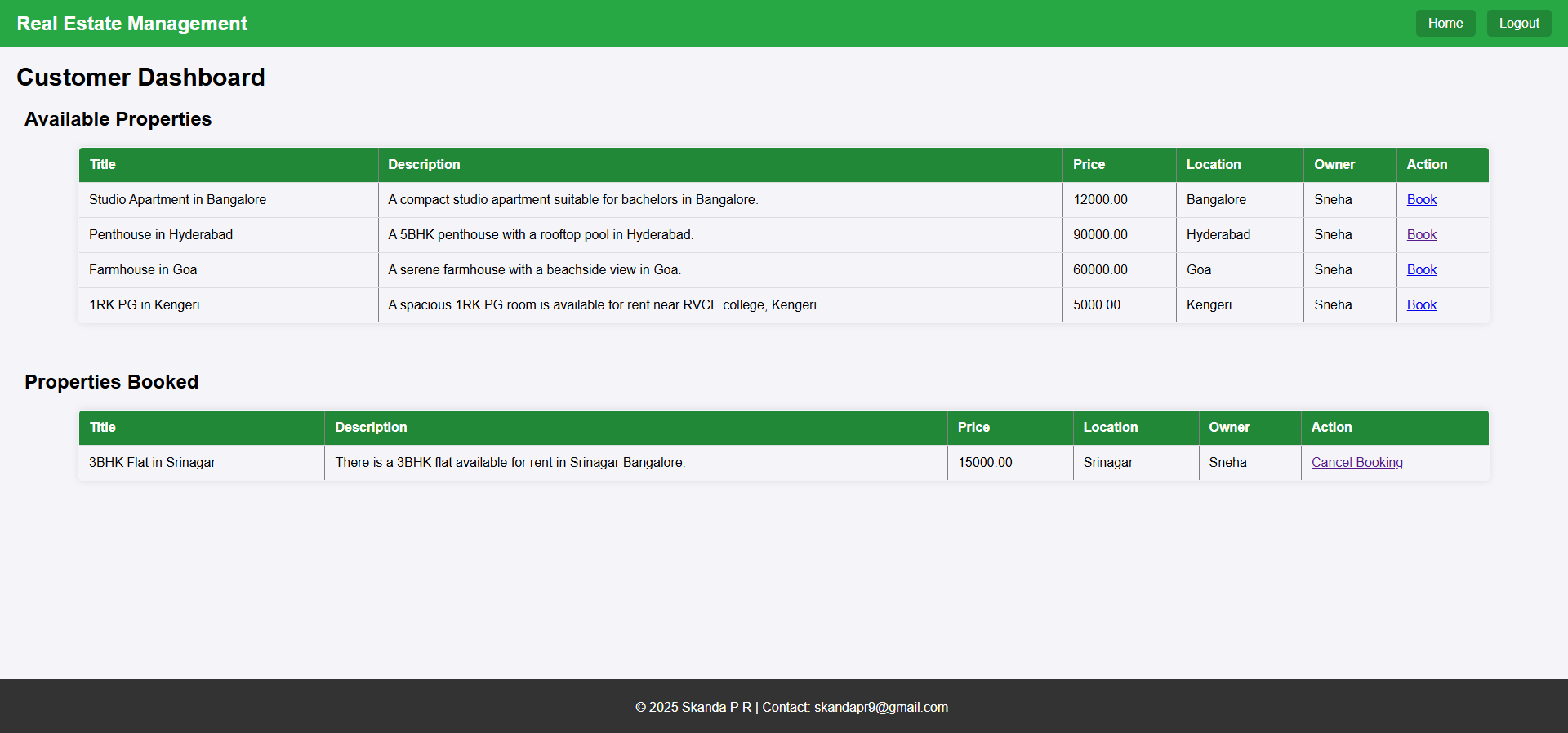
**Fig 8.3 Search Option**

****

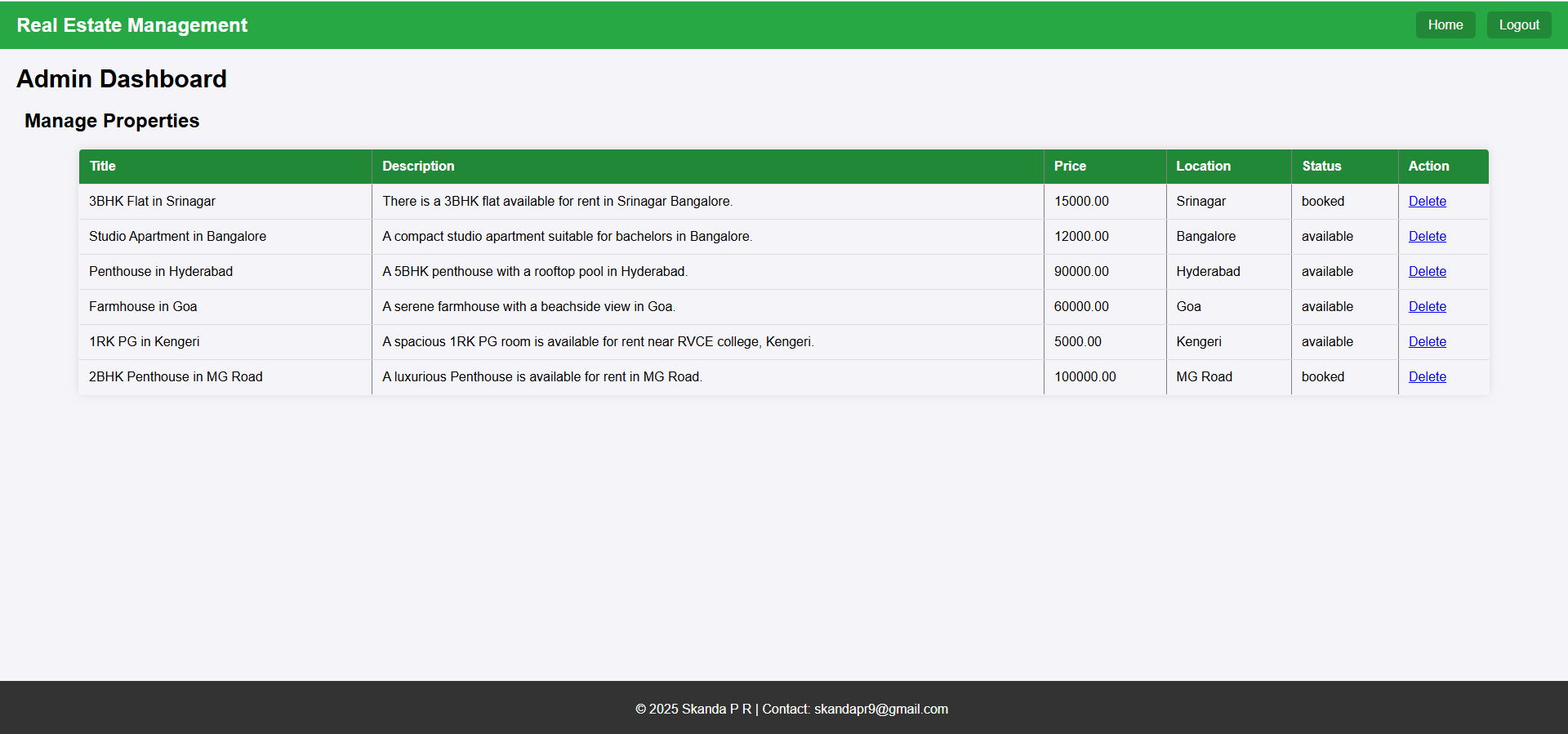
**Fig 8.4 Login Portal**

****

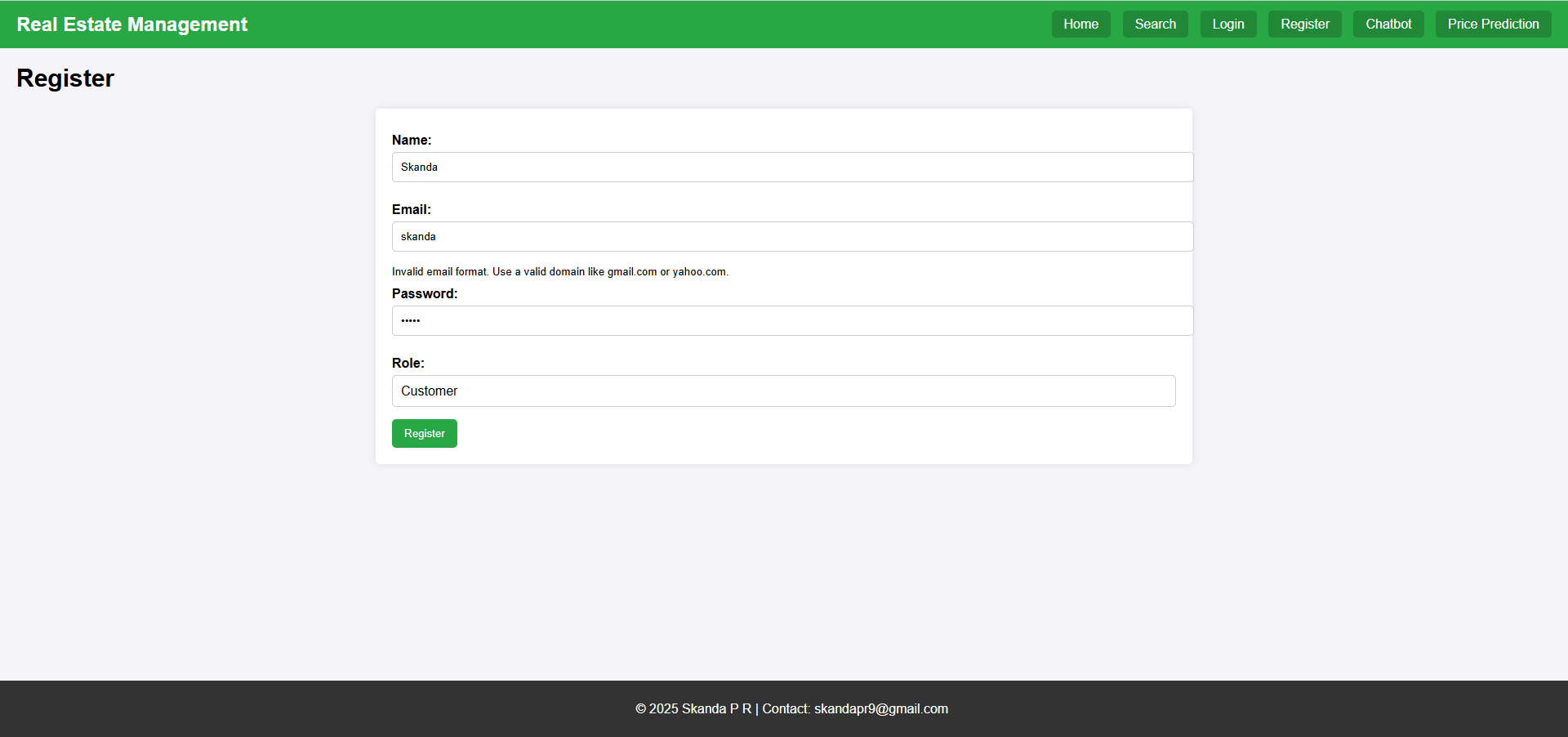
**Fig 8.5 Owner Dashboard**

****

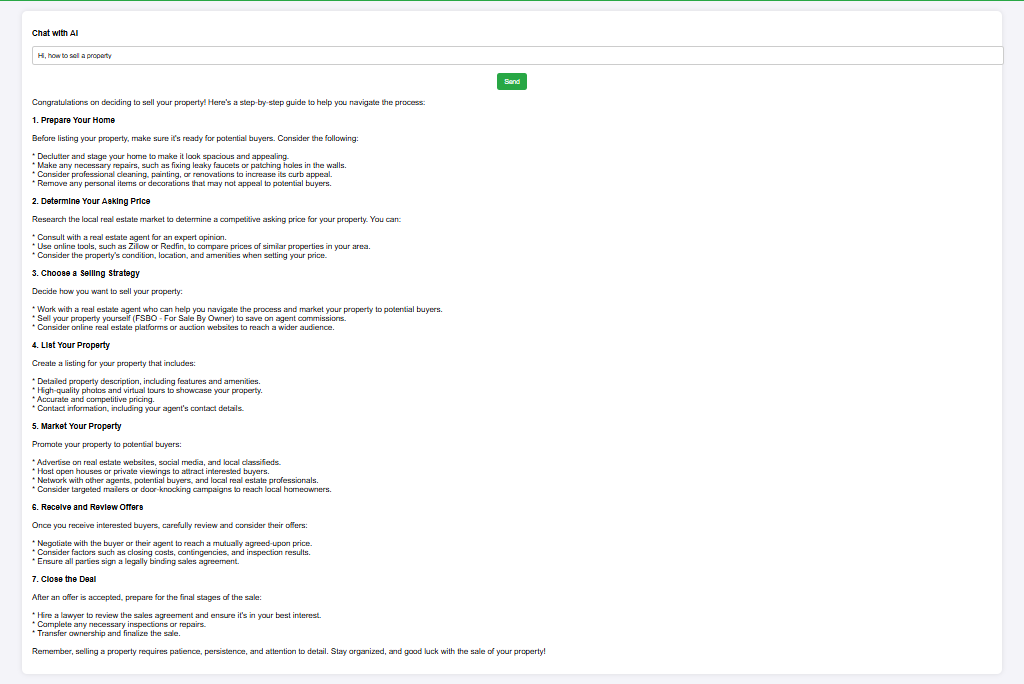
**Fig 8.6 Customer Dashboard**

****

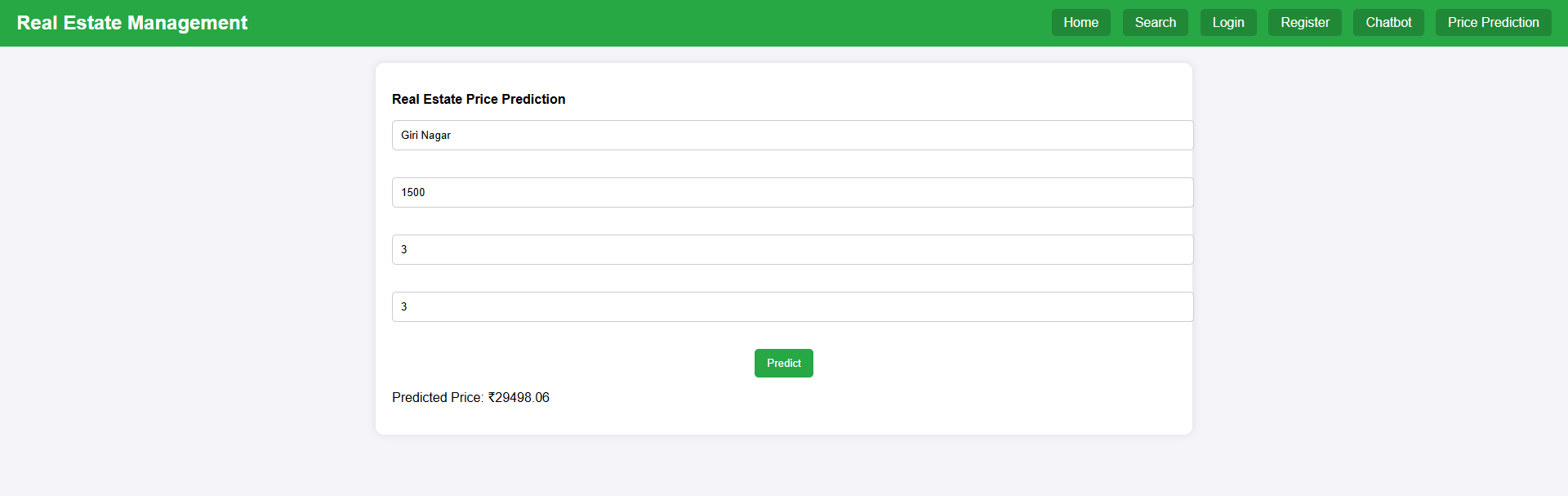
**Fig 8.7 Admin Dashboard**

****

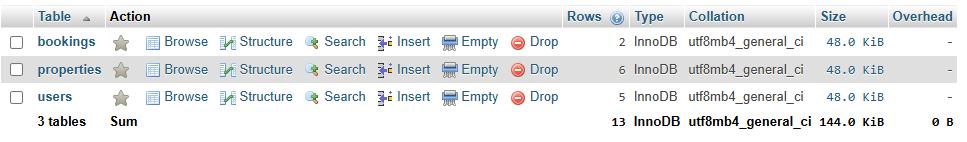
**Fig 8.8 Registration Portal**

****

**Fig 8.9 Chatbot**

****

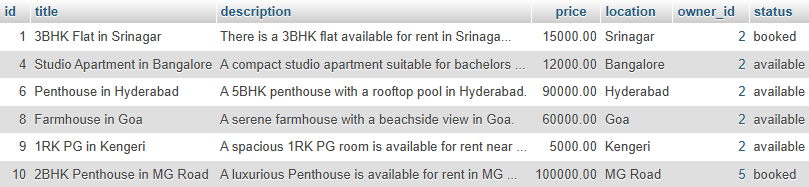
**Fig 8.10 Real Estate Price Prediction Page**

****

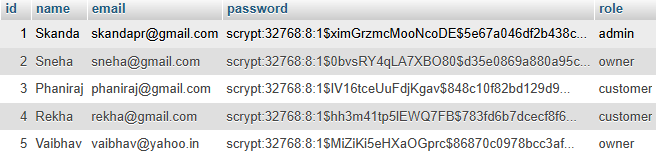
**Fig 8.11 List of Tables in DB**

****

**Fig 8.12 Booking Table Contents**

****

**Fig 8.13 Property Table Contents**

****

**Fig 8.14 Users Table Content**