# INTRODUCTION TO PROJECT

As urban areas expand and property markets evolve, managing real estate properties effectively becomes increasingly important. A **Real Estate Management System** (REMS) is designed to simplify and improve the management of various property types, including residential, commercial, and industrial. This system handles essential tasks such as listing properties, managing owners and users, overseeing finances, and scheduling maintenance.

The Real Estate Management System we’ve developed aims to consolidate these functions into a single, user-friendly platform. By automating repetitive tasks and centralizing information, it seeks to make property management more efficient for managers, landlords, tenants, and investors.

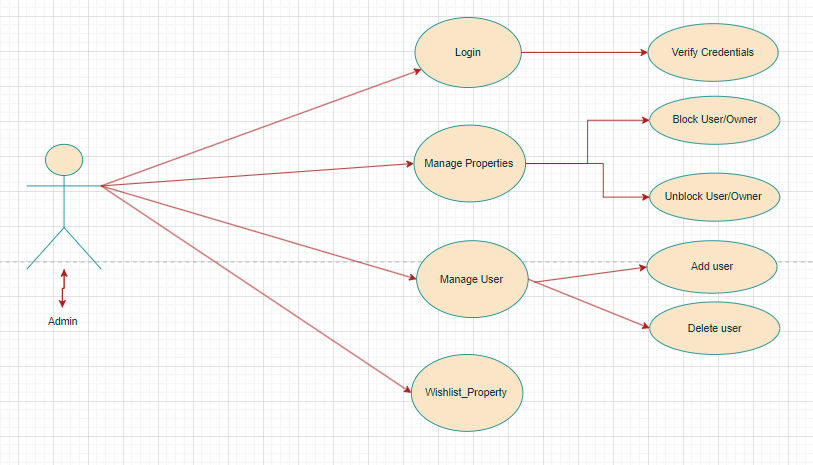
This report offers a summary of the Real Estate Management System. It covers the system’s design, main features, and implementation process. Additionally, the report assesses the system’s performance, gathers user feedback, and suggests areas for future improvement.

Our project is focused on addressing common challenges in property management and providing a solution that simplifies operations, supports better decision-making, and improves interactions among all parties involved.

**2.REQUIREMENTS**

**2.1 FUNCTIONAL REQUIREMENTS**

**2.1 Admin Account**

****

**1. Accessing the Admin Panel**

* **Admin Login:** The admin accesses the platform using a secure login process, typically requiring a unique username and password. To enhance security, the platform may employ two-factor authentication (2FA), which requires the admin to enter a code sent to their mobile device or email in addition to their password. This step ensures that only authorized personnel can access the admin panel.

**2. Dashboard Overview**

* **Admin Dashboard:** Upon logging in, the admin is directed to a dashboard that provides a high-level overview of the platform’s activity. The dashboard displays key metrics such as the total number of active users and property owners, the number of properties currently listed, and recent transactions. It might also highlight pending actions that require the admin’s attention, such as new property listings awaiting approval or user accounts flagged for review. The dashboard serves as a central hub where the admin can quickly assess the platform’s status and navigate to different management sections.

**3. User Management**

* **View and Access User Properties:** The admin has the capability to view all properties associated with any user on the platform. This includes detailed information about each property, such as its listing status (e.g., available, under offer, sold), location, price, and any associated documents. The admin can review these details to ensure compliance with platform rules and address any issues that arise.
* **Block/Unblock Users:** The admin can block users who violate platform policies or exhibit suspicious behavior. Blocking a user restricts their access to the platform, preventing them from logging in or interacting with properties. If a user resolves the issue or if the block was applied in error, the admin can unblock the user, restoring their access and permissions. This feature helps maintain the platform’s integrity and security.

**4. Owner Management**

* **View and Access Owner Properties:** Similar to user management, the admin can view all properties listed by a specific property owner. This feature allows the admin to monitor the owner’s activities, including the accuracy of property listings, adherence to platform guidelines, and overall conduct. The admin can take action if any of the owner’s properties violate the platform’s terms.
* **Block/Unblock Owners:** If a property owner is found to be non-compliant with the platform’s rules—such as listing fraudulent properties or engaging in unethical practices—the admin can block their account. Blocking an owner prevents them from adding or managing properties on the platform. If the situation is resolved or if further investigation shows the block was unnecessary, the admin can unblock the owner and restore their account privileges.

**5. Monitoring Activities**

* **Track User and Owner Actions:** The admin has access to logs or reports that detail the actions taken by both users and owners on the platform. These logs include activities like property listings, edits, purchases, and communication between users and owners. Monitoring these activities allows the admin to detect any unusual behavior, address potential security threats, and ensure that all interactions on the platform are in line with its policies.
* **Review Transactions:** The admin can review transactions between users and owners, ensuring that payments are processed correctly and that there are no discrepancies or issues. This includes checking payment records, reviewing the transfer of property ownership, and handling any disputes that arise between buyers and sellers. The admin’s role in overseeing transactions helps maintain trust and transparency on the platform.

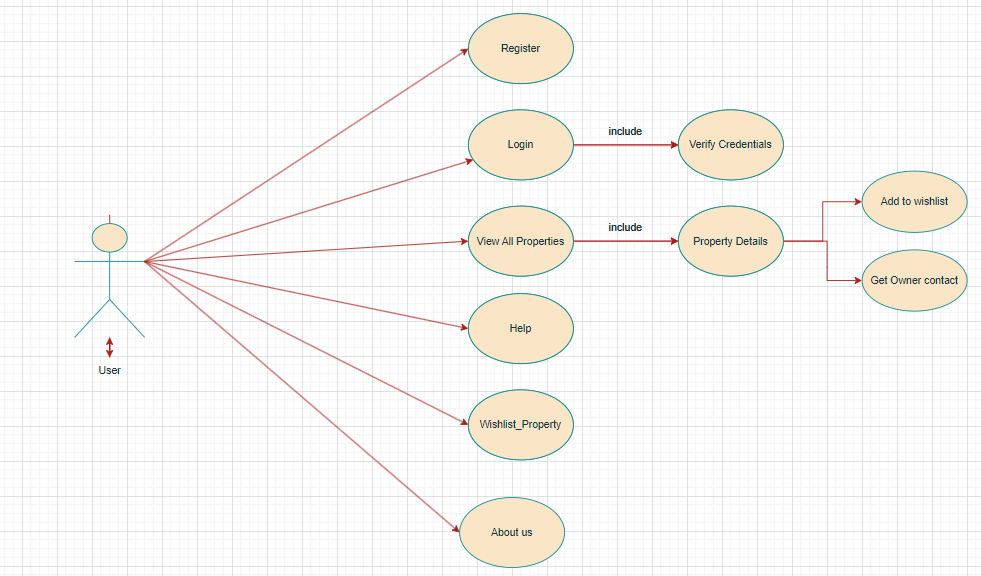
**6. Content and Platform Management**

* **Manage Content:** The admin has the ability to update or create content on the platform, such as modifying the terms of service, privacy policies, FAQs, and other informational pages. They can also post announcements or updates that are visible to all users, such as introducing new features, notifying users of scheduled maintenance, or communicating important policy changes. This ensures that all platform content remains current and relevant.

**7. Exiting the Platform**

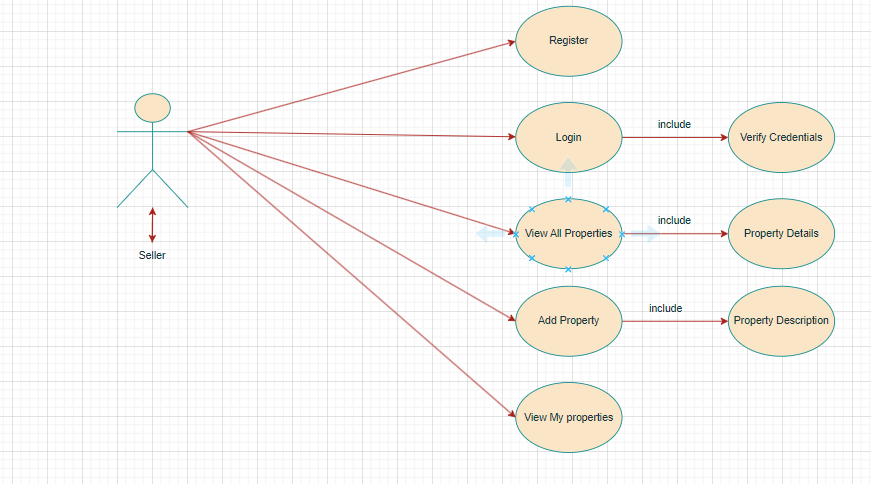
* **Log Out:** After completing their tasks, the admin securely logs out of the platform. This step is crucial to prevent unauthorized access, especially if the admin is using a shared or public computer. The platform may also include an automatic logout feature that activates after a period of inactivity, further enhancing security by reducing the risk of unauthorized access.

**2.2 User Account**

****

* + 1. **1. Account Creation and Access**
    2. **Register:** Users start by signing up on the platform, where they provide necessary information such as their name, email address, and a secure password. The platform sends a confirmation link or verification code to their email or phone, which they must confirm to complete the registration. This step ensures that only valid users gain access to the platform.
    3. **Login:** Once registered, users can log in using their email and password. If they forget their password, they can use the password recovery option, which sends a reset link or code to their registered email or phone, allowing them to create a new password and regain access to their account.
    4. **2. Explore Properties**
    5. **Browse Listings**: After logging in, users are presented with a range of properties listed by different owners. These listings are displayed in an organized manner, typically featuring key details like the property name, location, type (e.g., apartment, house), and price. Users can scroll through the listings to explore their options.
    6. **Search and Filters:** To help users find exactly what they’re looking for, the platform offers a search bar and various filters. Users can type in specific keywords (like a city or neighborhood) or apply filters based on criteria such as price range, property type, number of bedrooms, or other features. This functionality allows users to quickly narrow down the list of properties to those that match their preferences.
    7. **3. Property Details**
    8. **View Details:** When a user finds a property that interests them, they can click on it to view more detailed information. The property’s detail page includes a comprehensive description, multiple high-quality images, the price, and details about amenities (e.g., parking, swimming pool, garden). The page may also provide information on nearby facilities like schools, hospitals, or shopping centers, helping users assess whether the property meets their needs.
    9. **Owner Information**: On the same property detail page, users can access information about the property owner. This may include the owner’s profile, their contact details (if shared), and possibly ratings or reviews from previous interactions with other users. This transparency helps build trust and confidence in the transaction process.
    10. **4. Wishlist Management**
    11. **Add to Wishlist:** If a user is interested in a property but isn’t ready to make a purchase, they can save it to their wish-list. This feature allows users to bookmark properties they like, so they can easily find and review them later. The wish-list is especially useful for users who are comparing multiple properties or waiting for further information before making a decision.
    12. **Manage Wishlist:** Users can access their wish-list through their account dashboard. From there, they can review the properties they’ve saved, compare them, or remove any properties they’re no longer interested in. When they’re ready, they can click on a saved property to view its details again or proceed to purchase it.
    13. **5. Purchasing a property**
    14. **Start Purchase:** Once a user decides to buy a property, they initiate the purchase process by clicking the “Buy Now” button on the property’s detail page. This action takes them to the payment process, where they can finalize their purchase.
    15. **Payment Process**: During the payment process, users are guided through a series of steps to complete the transaction. They can select their preferred payment method, which might include options like credit/debit cards, bank transfers, or other payment gateways. The platform ensures that all payment transactions are processed securely to protect user information and funds.
    16. **Purchase Confirmation:** After the payment is successfully processed, the user receives a confirmation message both on the platform and via email. This confirmation includes details of the transaction, such as the property purchased, the amount paid, and any next steps (e.g., arranging property visits or receiving documents).

**2.3 Owner Account**



**1. Account Setup and Access**

* **Registration:** Property owners start by signing up on the platform. They provide their personal details, including their name, email address, and a password. A verification link or code is sent to their email or phone to confirm their registration, ensuring the validity of the account.
* **Login:** After successfully registering, owners log in using their email and password. If they forget their login credentials, they can use the password recovery option to reset their password via an email or phone code.

**2. Property Management**

* **Add Property:** Owners can list new properties by filling out a form with necessary details. This includes the property’s name, location, type (e.g., house, apartment), price, and a detailed description. Owners can upload images and any relevant documents to enhance the property listing. The new listing may go through an approval process before it is visible to other users on the platform.
* **Update Property**: Owners can modify details of their existing properties if there are any changes. This could involve updating the property’s price, revising the description, or adding new images. Once updated, the changes are saved and reflected on the property’s detail page for potential buyers to see.
* **View Property**: Owners can view a list of all properties they have added to the platform. This view includes information like the property’s current status (e.g., available, under offer), its location, and any images or descriptions associated with it. This overview helps owners keep track of their listings.
* **Remove Property:** If an owner decides to remove a property from the platform, they can select the property from their list and choose to delete or deactivate it. This action will remove the property from public view, but it may still be retained in the owner’s account history for record-keeping purposes.

**3. Wishlist Functionality**

* **Add to Wishlist:** Owners can also utilize a wishlist feature to save properties of interest. This might include properties they wish to monitor, compare with their own, or consider for future transactions. They can add properties to their wishlist directly from the property’s detail page.
* **Manage Wishlist:** The wish-list can be accessed from the owner’s account dashboard. Here, owners can review properties they have saved, manage their preferences by removing properties they are no longer interested in, or follow up on properties they are considering.

**4. Optional Notifications and Alerts**

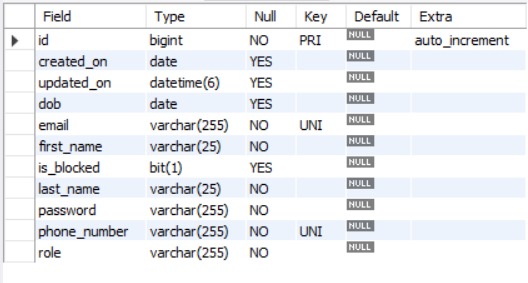
* **Receive Updates:** Depending on the platform’s features, owners may receive notifications about updates related to their properties or wishlist items. This could include alerts about changes in property status, price adjustments, or other relevant information.
  1. **NON-FUNCTIONAL REQUIREMENTS**
* **1. Performance Requirements**
* **Response Time:** The system should display pages and respond to user actions (such as property searches and purchases) within 2 seconds to ensure a smooth experience.
* **Scalability:** The platform should efficiently support up to 10,000 simultaneous users without performance degradation, even during peak times.
* **2. Usability Requirements**
* **User Interface (UI):** The design should be straightforward and easy to navigate for all users, including those with varying levels of technical expertise.
* **Accessibility:** The platform must meet accessibility guidelines (such as WCAG 2.1) to ensure it is usable by people with disabilities. This includes features like compatibility with screen readers and full keyboard navigation.
* **Ease of Learning**: New users should be able to perform basic tasks and navigate the platform effectively within 5 minutes of their first use.
* **3. Security Requirements**
* **Data Protection:** User data, including sensitive information like payment details, must be protected through encryption both during transmission and while at rest, using modern security protocols.
* **Authentication:** The system should implement robust authentication methods, such as strong password policies and multi-factor authentication (MFA), to secure user accounts.
* **Authorization:** Access controls must be in place to ensure that users, property owners, and administrators can only access features and data relevant to their roles.
* **4. Reliability Requirements**
* **Availability:** The platform should maintain an uptime of 99.9%, excluding scheduled maintenance periods.
* **Backup and Recovery:** Regular backups should be taken, and there should be a disaster recovery plan in place to restore service within 1 hour in the event of a major disruption.
* **5. Compatibility Requirements**
* **Browser Support:** The platform should be compatible with the latest versions of major web browsers, including Chrome, Firefox, Safari, and Edge.
* **Device Support:** The system should work seamlessly across a range of devices, including desktops, tablets, and smartphones, adapting to different screen sizes.
* **6. Compliance Requirements**
* **Regulatory Compliance:** The platform must adhere to relevant legal and regulatory requirements, such as GDPR for data protection and CCPA for privacy.
* **7. Maintainability Requirements**
* **Code Quality:** The codebase should be well-organized and documented to facilitate maintenance and future updates. Best practices for coding standards should be followed.
* **Upgrades:** The platform should be designed to support easy updates and integration of new features without causing significant disruptions or requiring extensive rework.
* **8. Interface Requirements**
* **User Interface (UI) Consistency:** Design elements like colors, fonts, and buttons should be consistent throughout the platform to provide a cohesive user experience.
* **System Interfaces:** The platform should integrate smoothly with external services, such as payment processors and data providers, through standardized APIs.

**3. DESIGN**

**3.1 Database Design**

The following table structures depict the database design.

# Table1: User Information



**Table 2: Owner Information**

A screenshot of a computer

Description automatically generated

**Table 3: Property Information**

**A screenshot of a computer

Description automatically generated**

**Table 4: Wishlist Information**

**A screenshot of a computer

Description automatically generated**

**E-R Diagram, Dataflow diagram and Class Diagram:**

Go to Appendix A

**4. CODING STANDARDS IMPLEMENTED**

### Naming and Capitalization

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | **Case** | **Examples** | **Additional Notes** |
| Class | Pascal | Person, Bank Vault, SMS Message, Dept | Class names should be based on "objects" or "real things" and should generally be **nouns**. No ‘\_’ signs allowed. Do not use type prefixes like ‘C’ for class. |
| Method | Camel | getDetails, updateStore | Methods should use **verbs** or verb phrases. |
| Parameter | Camel | personName, bankCode | Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in most scenarios. |
| Interface | Pascal with "I" prefix | Disposable | Do not use the ‘\_’ sign |
| Property | Pascal | ForeColor, BackColor | Use a noun or noun phrase to name properties. |
| Associated private member variable | \_camelCase | \_foreColor, \_backColor | Use underscore camel casing for the private member variables |
| Exception Class | Pascal with "Exception" suffix | WebException, |  |

### Comments

* Comment each type, each non-public type member, and each region declaration.
* Use end-line comments only on variable declaration lines. End-line comments are comments that follow code on a single line.
* Separate comments from comment delimiters (apostrophe) or // with one space.
* Begin the comment text with an uppercase letter.
* End the comment with a period.
* Explain the code; do not repeat it.

**5. TEST REPORT**

**Another group called Linux did the testing and the report of the testing is given hereunder.**

**GENERAL TESTING:**

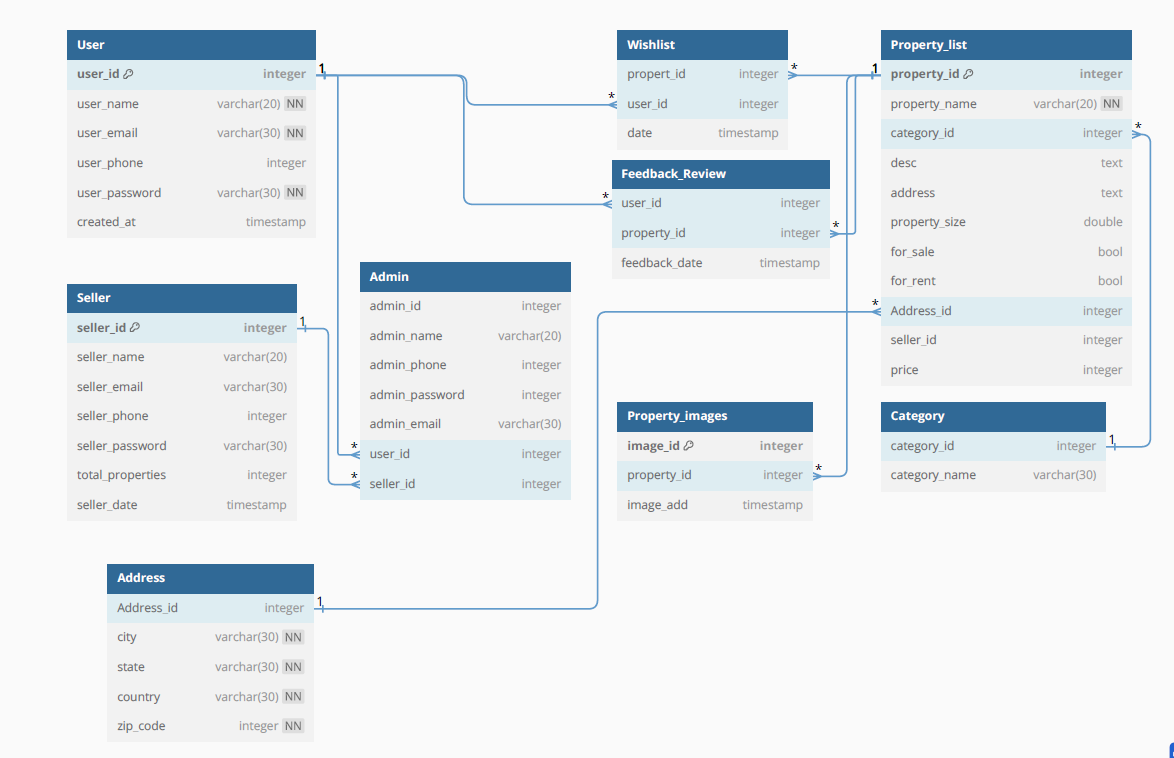
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR-NO** | **TEST CASE** | **EXPECTED RESULT** | **ACTUAL RESULT** | **ERROR MESSAGE** |
| 1 | Register Page | Redirected to Next page | OK | Nothing |
| 2 | Login Page | Pop-up will come | Ok | Please enter username and password again . |
| 3 | Reset login | Only user’s password will be reset | Ok | Nothing |
| 4 | Quick search property | Gives all property details | Ok | Nothing |
| 5 | Booking Property | All the fields should be filled for submission | Ok | Nothing |
| 6 | Checking login or not | User is logged in or not | Ok | Nothing |
| 7 | Goto property page | Set added information about user’s or owner’s property | Ok | Nothing |
| 8 | Add information in property table | Save this all data into property table | Ok | Nothing |
| 9 | Transaction | On back it should be reverted to previous page | Ok | Nothing |
| 10 | View transaction done | It shows you all transactions done previously | Ok | Nothing |
| 11 | Logout | It will logout from user profile. | Ok | Nothing |
|  | **STATIC TESTING** |  |  |  |
| **SR-NO** | **Deviation** | **Program** |  |  |
| 1 | Commenting not followed | All Web Application |  |  |

**6. PROJECT MANAGEMENT RELATED STATISTICS**

|  |  |  |  |
| --- | --- | --- | --- |
| **DATE** | **WORK PERFORMED** | ****SLC Phase**** | **Additional Notes** |
| APRIL 18,2024 | Project Allotment and User Requirements Gathering | Feasibility Study | Our team met the client Mr. Nitin Kudale (CEO, SIIT Pune) to know his requirements. |
| APRIL 24,2024 | Initial SRS Document Validation  And Team Structure Decided | Requirement Analysis  (Elicitation) | The initial SRS was presented to the client to understand his requirements better |
| APRIL 24,2024 | Designing the use-cases, Class Diagram, Collaboration Diagram, E-R Diagram and User Interfaces | Requirement Analysis &  Design Phase | Database Design completed |
| MAY 10,2024 | Business Logic Component design Started | Design Phase | ---------------------- |
| MAY 10,2024 | Coding Phase Started | Coding Phase | 70% of Class Library implemented. |
| MAY 15,2024 | Implementation of Web Application and Window Application Started | Coding Phase | Class Library Development going on. |
| MAY 25,2024 | Implementation of Web Application and Window Application Continued | Coding Phase and Unit Testing | Class Library Modified as per the need. |
| MAY 30,2024 | Implementation of Web Application and Window Application Continued | Coding Phase and Unit Testing | -- |
| JUNE 30,2024 | After Ensuring Proper Functioning the Required Validations were Implemented | Coding Phase and Unit Testing | Module Integration was done by the Project Manager |
| JULY 10,2024 | The Project was Tested by the respective Team Leaders and the Project Manager | Testing Phase (Module Testing) | -- |
| AUG 10,2024 | The Project was Submitted to Other Project Leader of Other Project Group for Testing | Testing Phase (Acceptance Testing) | The Project of Other Team was Taken up by the Team for Testing |
| AUG 12-13,2024 | The Errors Found were Removed | Debugging | The Project was complete for submission |
| AUG 16,2024 | Final Submission of Project |  |  |

Appendix A

Entity Relationship Diagram



**Data Flow Diagram:**

**A diagram of a diagram

Description automatically generated**

**A diagram of a diagram

Description automatically generated**

**A diagram of a company

Description automatically generated**

**Class Diagram**

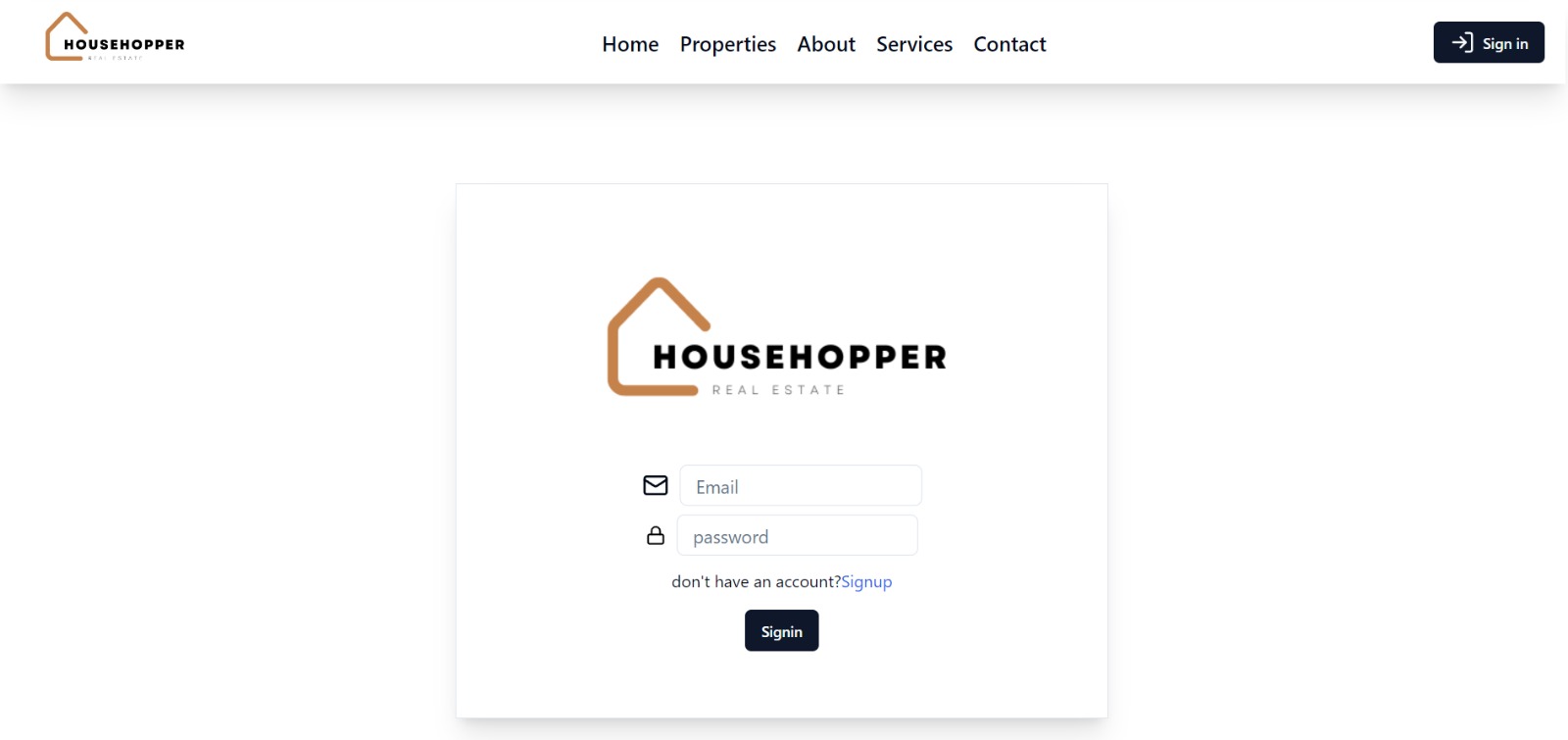
A diagram of a diagram

Description automatically generated

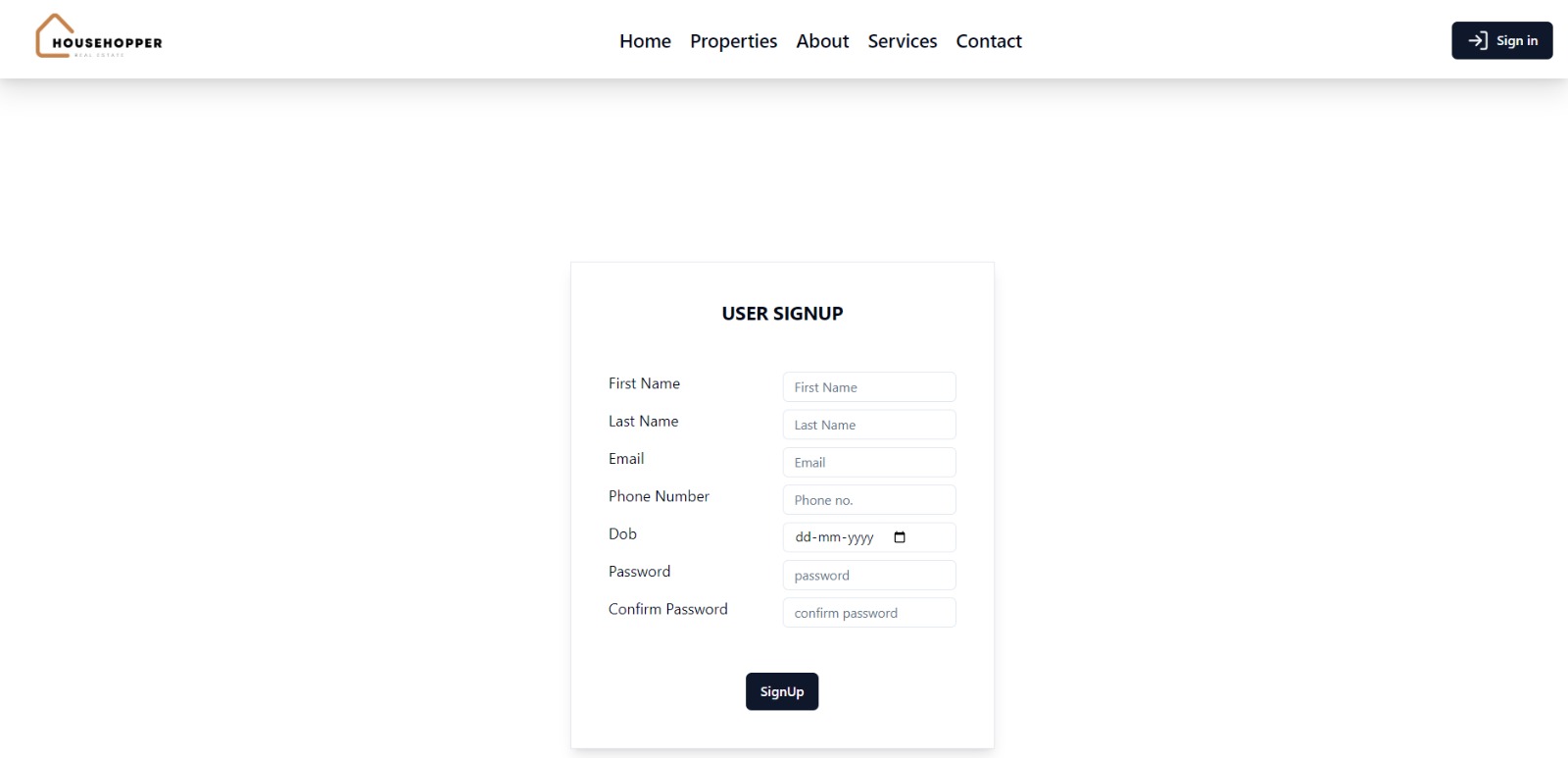
Appendix B

Homepage:

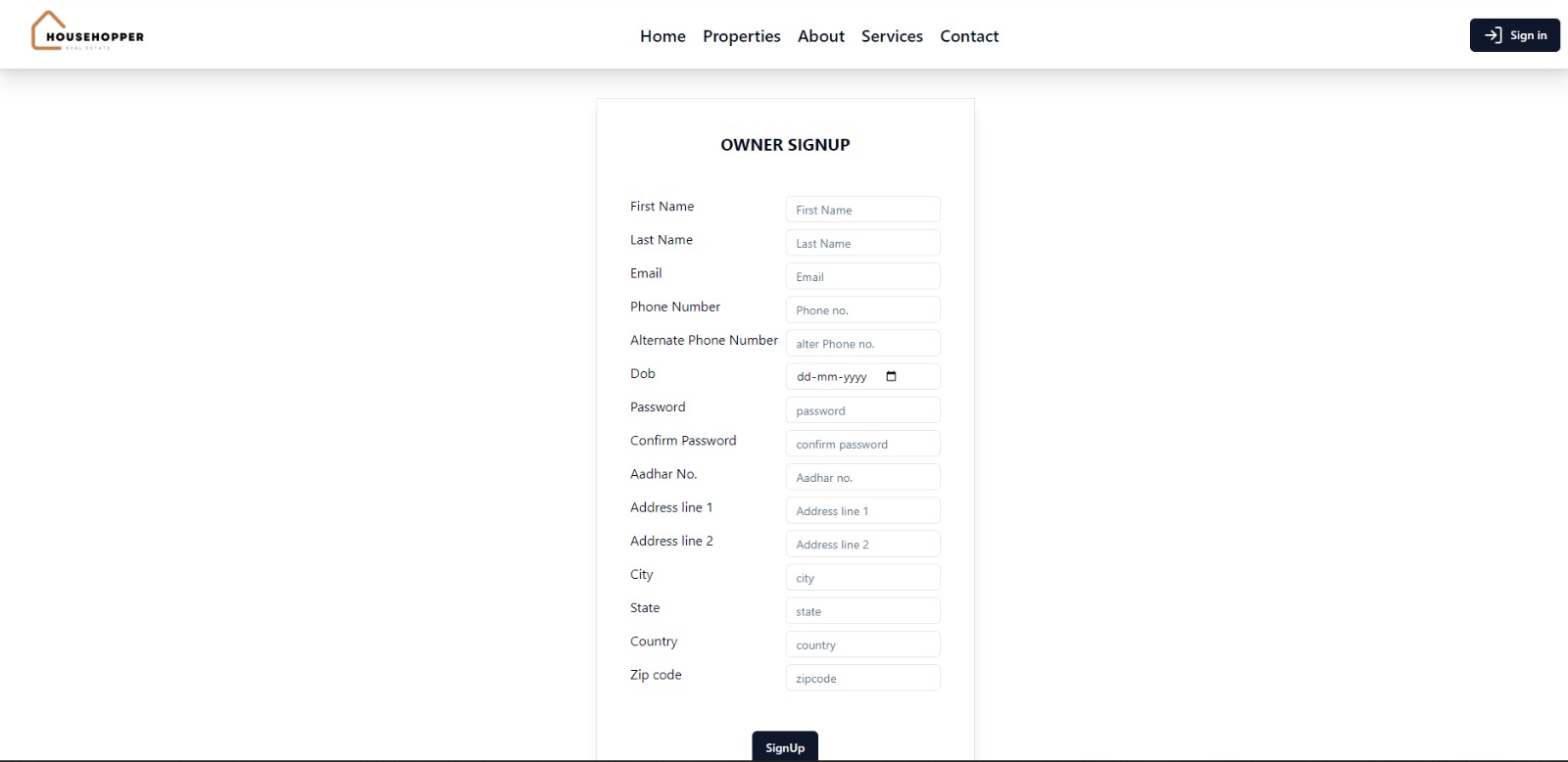


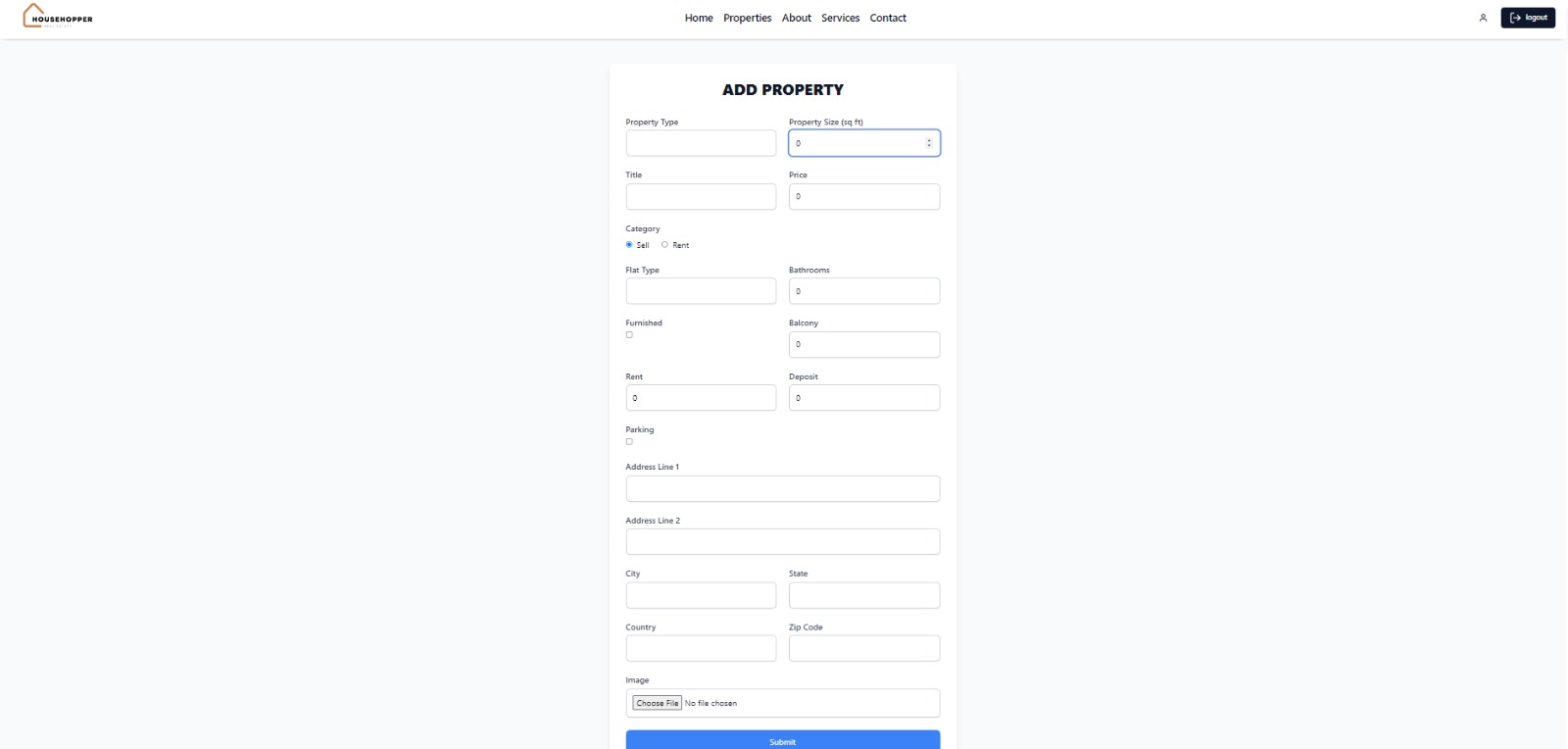
LoginPopup:

User Registration:

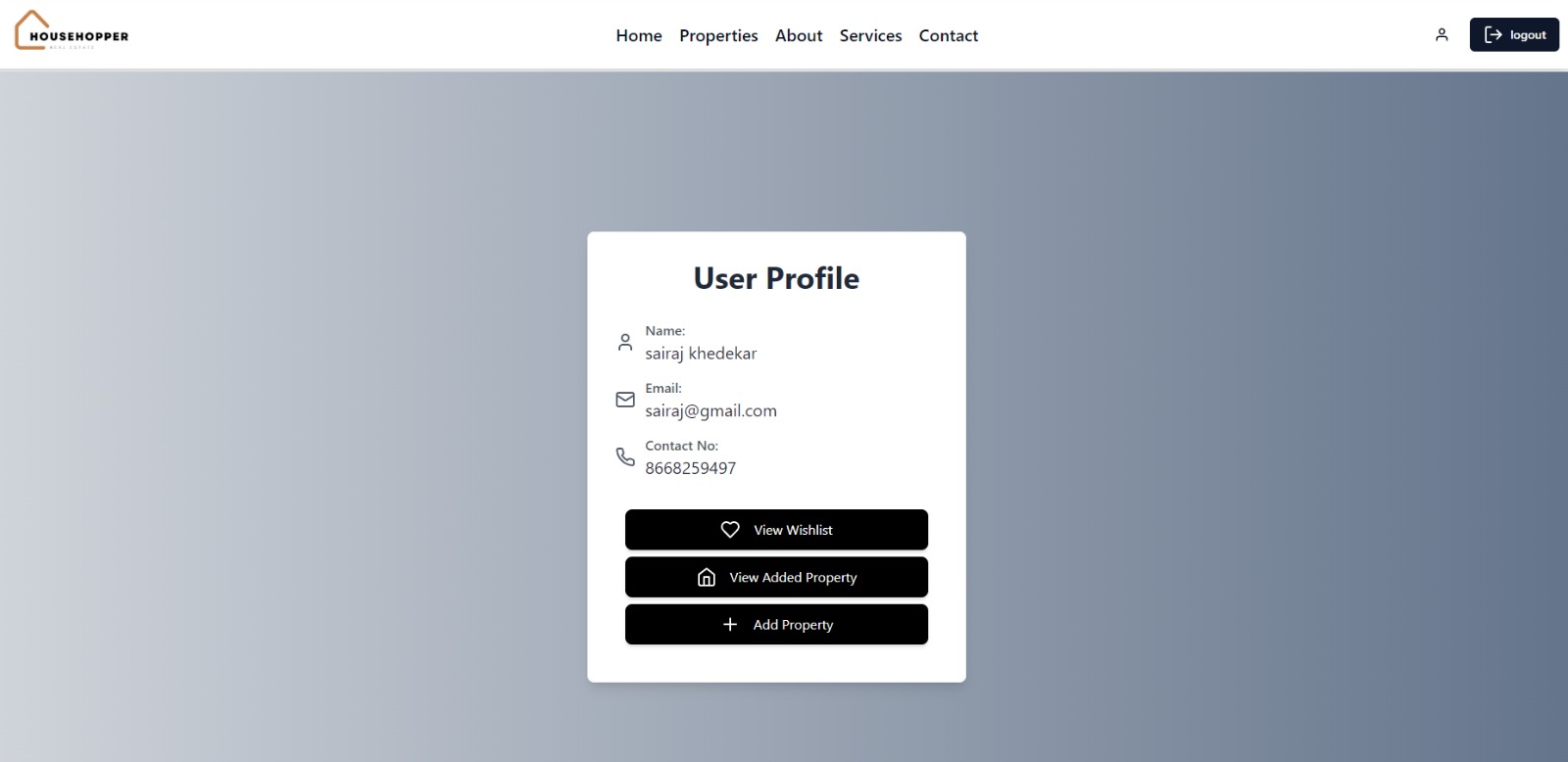
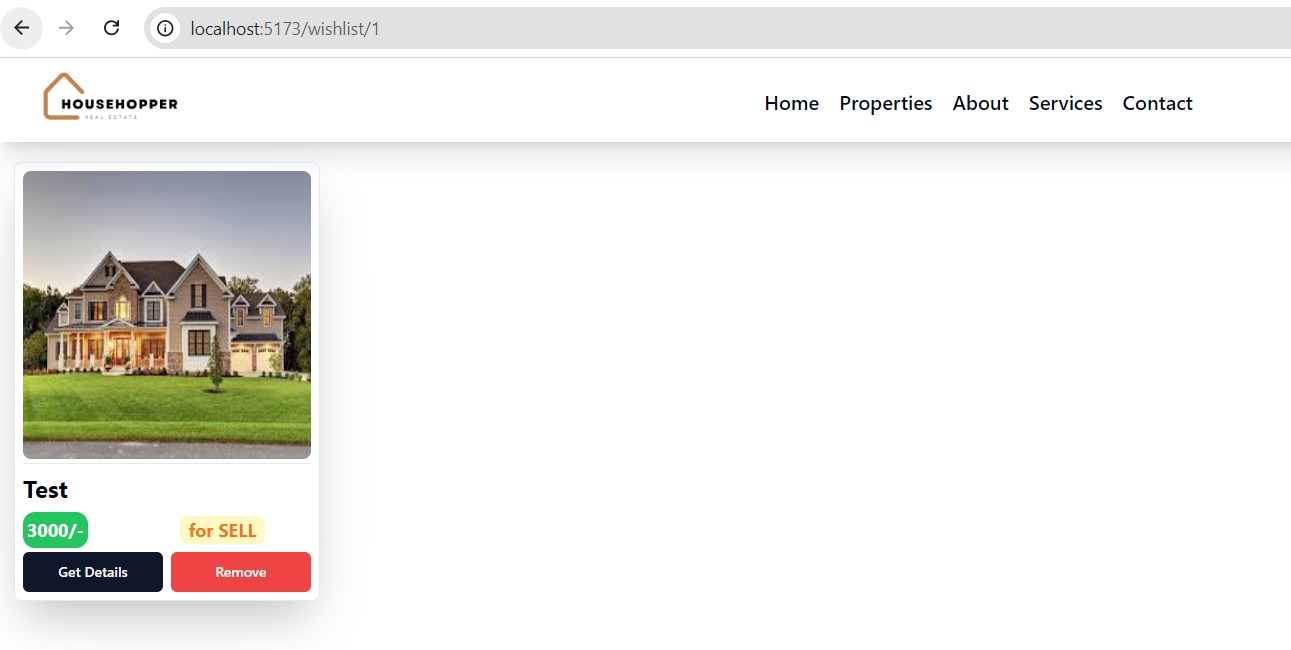


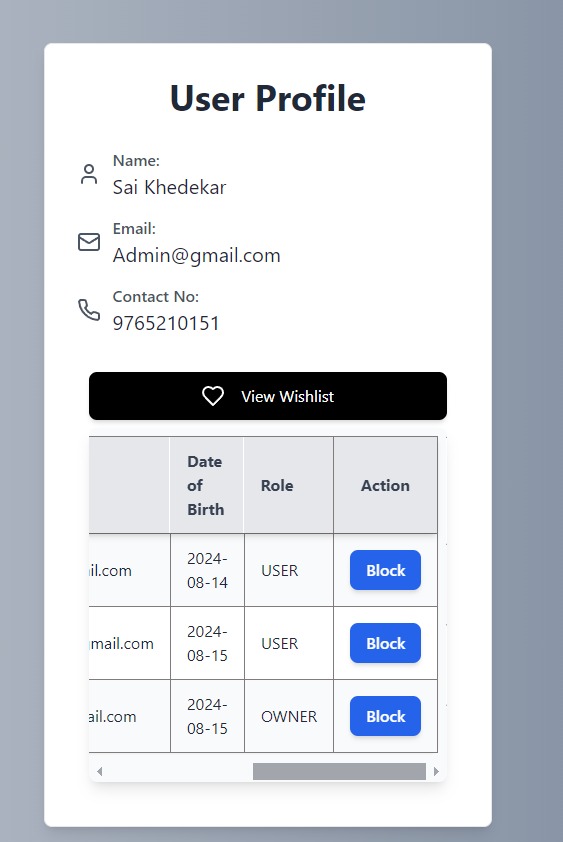
**PropertyOwner Registeration**

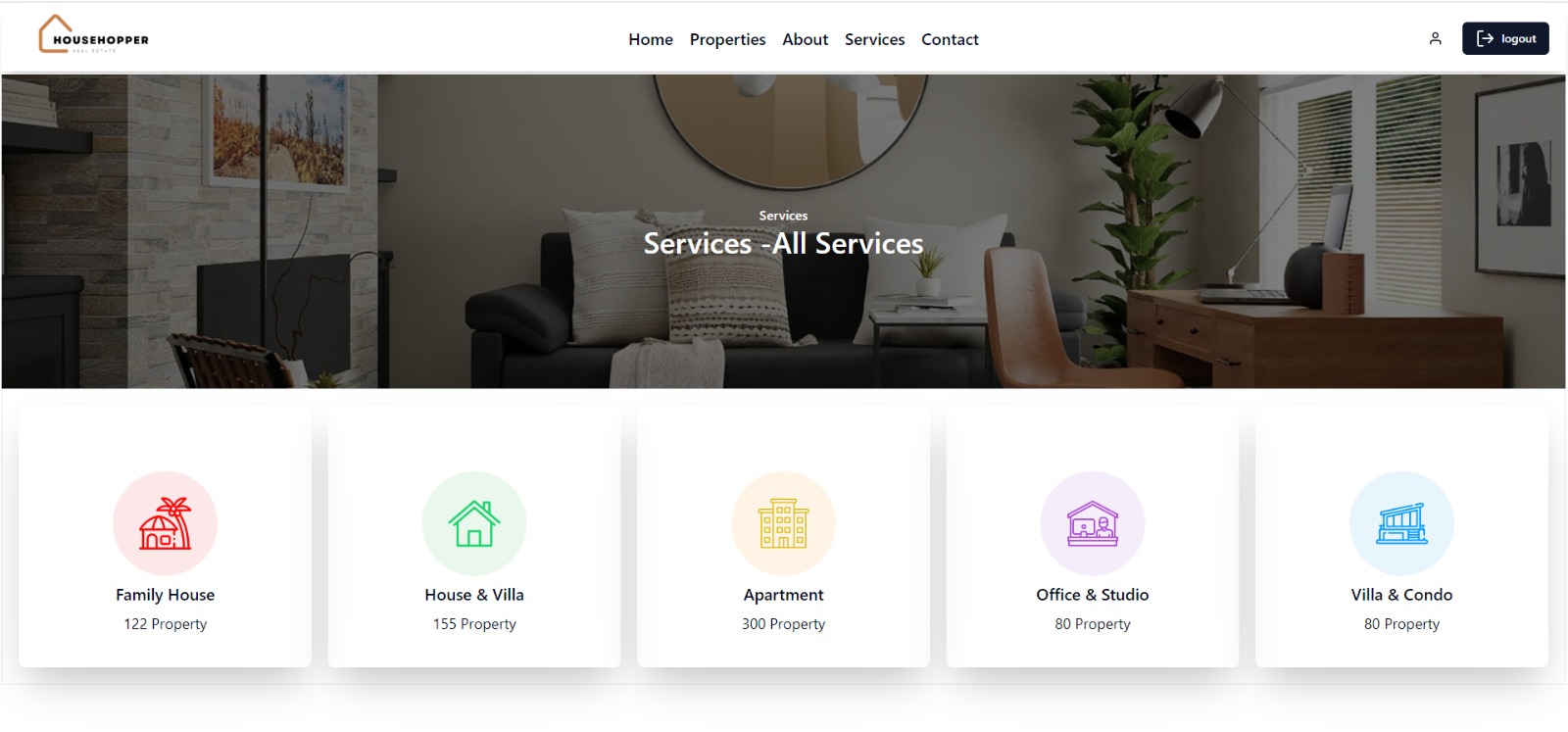


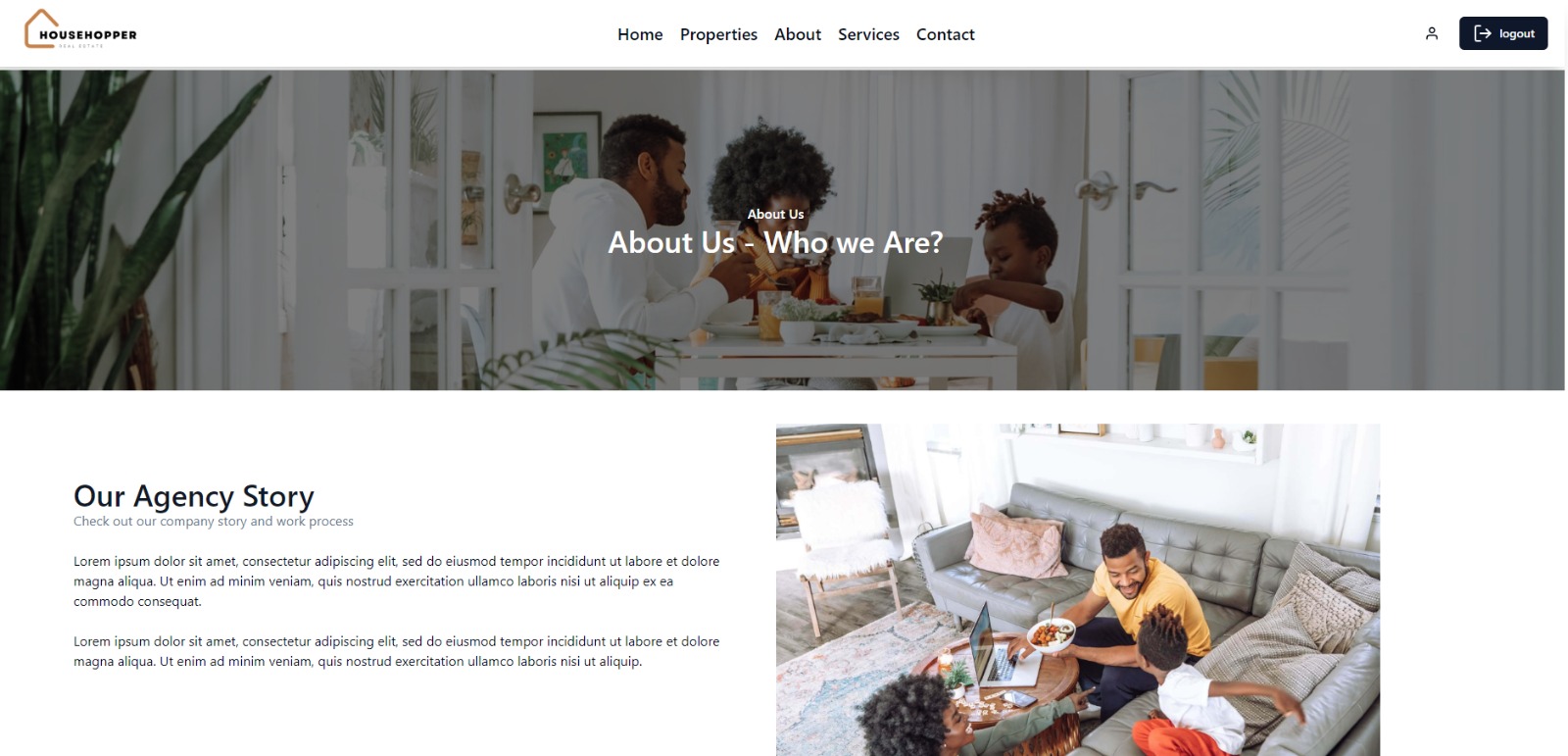
**AddProperty**

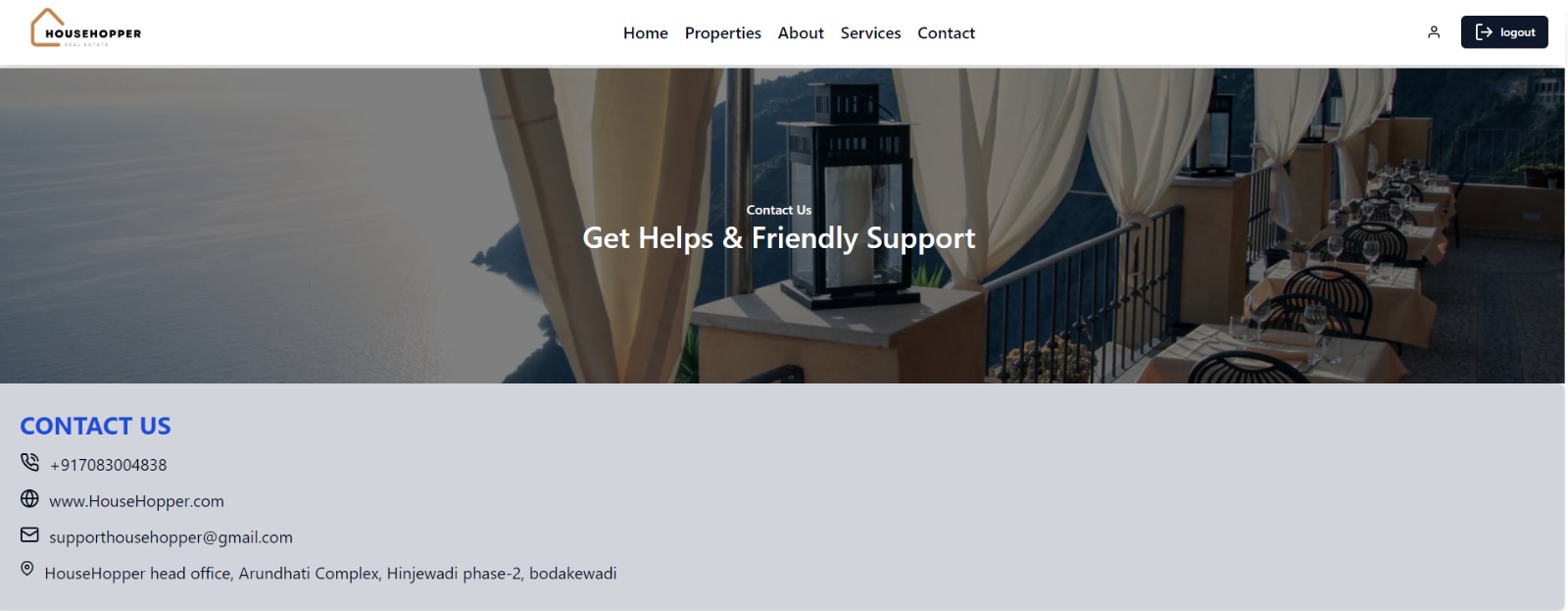
**User Profile**

**WishListPage:**

AdminPage:

Services 

AboutUs:

**ContactUs:**

**7.REFERENCES:**

<http://www.google.com>

[http://](http://www.xml101.com:8081/xml/)www.magicbricks.com

http://www.webdevelopersjournal.com

http://www.w3.org

http://www.wikipedia.org

http://www.delta.com

http://www.tailwind.com

http://www.w3school.com