**Veer Narmad South Gujarat University, Surat.**

**Department of Information and Communication Technology**



# **Project Report**

**1st Semester**

**M.Sc. (Information and Communication Technology)**

## 2 Year Course

**Year 2023 – 2024**

“HouseMarket Place"

Guided By : Submitted By :

Dr. Veena Jokhakar(Guid) Shyam (R23110018000710062)

Shivani (R23110018000710054)

**Veer Narmad South Gujarat University, Surat.**

**Department of Information and Communication Technology**

**M.Sc. (Information and Communication Technology) Programme**

# **Certificate**

This is to certify that Mr./Ms. **Sadisatsowala shyam** with Exam Seat Number: **10054** Enrollment Number:**R23110018000710062** has worked on his/her project work entitled **HouseMarketPlace** at **J.P.Dawer Institute** as a partial fulfillment of the requirements for 1st ***Semester - M.Sc. (Information and Communication Technology)***, during the academic Year 2023-2024.

Date :

Place : Dept. of ICT, VNSGU, Surat.

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Internal Project Guide**  **M.Sc.(I.C.T.) 3rd Semester Department of I.C.T.**  **Veer Narmad** **South**  **Gujarat University, Surat** | **Course Coordinator**  **M.Sc. (I.C.T.) Programme Department of I.C.T.**  **Veer Narmad** **South**  **Gujarat University, Surat** | **Head of the Department Department of I.C.T.**  **Veer Narmad** **South**  **Gujarat University, Surat** |

**Veer Narmad South Gujarat University, Surat.**

**Department of Information and Communication Technology**

**M.Sc. (Information and Communication Technology) Programme**

# **Certificate**

This is to certify that /Ms. **,Patel Shivani**  with Exam Seat Number: **10048** and Enrollment Number:**R23110018000710054** has worked on his/her project work en titled **HouseMarketPlace** at **J.P.Dawer Institute** as a partial fulfillment of the requirements for 1st ***Semester - M.Sc. (Information and Communication Technology)***, during the academic Year 2023-2024.

Date :

Place : Dept. of ICT, VNSGU, Surat.

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Internal Project Guide**  **M.Sc.(I.C.T.) 3rd Semester Department of I.C.T.**  **Veer Narmad** **South**  **Gujarat University, Surat** | **Course Coordinator**  **M.Sc. (I.C.T.) Programme Department of I.C.T.**  **Veer Narmad** **South**  **Gujarat University, Surat** | **Head of the Department Department of I.C.T.**  **Veer Narmad** **South**  **Gujarat University, Surat** |

Index

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Particulars | Page No. | Sign |
| 1. | Introduction   * 1. Project Profile   2. Introduction   3. Scope & Objective   4. Project Goal   5. Project Category | 4-8 |  |
| 2. | Environment description   * 1. Hardware and software requirement   2. Application area of this system | 9-12 |  |
| 3. | Analysis Reports   * 1. Drawback of the current System   2. Requirement Analysis   3. Requirement specification   4. Proposed system   5. Advantage of the Proposed System | 13-22 |  |
| 4. | Software diagram  4.1 Admin & User activity diagram  4.2 Admin & User usecase diagram  4.3 Sequence diagram  4.4 E-R diagram  4.5 Class diagram | 23-31 |  |
| 5. | Database design | 32-35 |  |
| 6. | Testcase design | 36-37 |  |
| 7. | Project images | 38-45 |  |
| 8. | Future scope | 46-47 |  |
| 9 | Reference | 48-49 |  |

**1.Introduction**

1.1 Project Profile

|  |  |
| --- | --- |
| Project Name | “HouseMarket Place” |
| Project type | Full stack website |
| Duration | 5 month |
| Tools   * Front-End * Back-End * Code Editor * Rest Testing | * JSF , Bootstrap , CSS * JPA , EJB , Rest , Jersy Client * NetBeans * PostMan |
| Operating System | Windows 11 |
| Project Guide | Prof. Hiral ma’am |
| Submitted By | Shyam Sadisatsowala  Shivani Patel |
| Submitted To | “J.P.Dawer institute of Information Science and Technology - BSc (IT) & MSc (ICT)” |

1.2 Introduction

Welcome to the comprehensive documentation for HouseMarket Place, a cutting-edge real estate platform designed to seamlessly connect property seekers and property owners. This platform provides a robust and user-friendly experience for both administrators and users, offering a wide range of features to facilitate property management, booking transactions, and user interactions.

The administrative side of HouseMarket Place empowers administrators with powerful tools to efficiently manage the platform. Admins can create and define roles such as 'rent' and 'sale,' ensuring a streamlined categorization of properties. Through the intuitive admin interface, property creation, display, update, and deletion become seamless tasks. The administration dashboard consolidates key metrics and insights for effective decision-making. Admins can also manage bookings, including confirming and canceling reservations, and oversee user reviews to maintain the platform's integrity. The user management functionality allows administrators to efficiently handle user accounts, ensuring a secure and well-maintained user base.

On the user side, HouseMarket Place provides a user-friendly interface for property exploration and interaction. Users can easily browse and view available properties, accessing detailed information about each listing. The platform enables users to submit reviews, fostering a community-driven environment. Additionally, users can conveniently check the status of their bookings and update their passwords, ensuring a personalized and secure experience.

HouseMarket Place leverages a robust technology stack to deliver a seamless and secure experience. The database is powered by MySQL, providing a reliable and scalable foundation for data management. JPA (Java Persistence API), REST (Representational State Transfer), EJBs (Enterprise JavaBeans), Jersey Client, CDI beans (Contexts and Dependency Injection), and JSF (JavaServer Faces) collectively contribute to the platform's functionality and user interface. The implementation of JWT (JSON Web Token) Authentication Mechanism ensures secure and efficient user authentication, enhancing the overall security of the HouseMarket Place platform. This documentation serves as your comprehensive guide to understanding and maximizing the capabilities of HouseMarket Place, empowering you to navigate the platform with confidence and take full advantage of its features.

1.3 Scope & Objective

Scope:

The scope of the HouseMarket Place project is to provide a sophisticated and user-centric real estate platform that caters to the needs of both property administrators and users. The platform aims to streamline property management processes for administrators by offering features such as role creation, property CRUD operations, booking management, and user review oversight. On the user side, the scope includes intuitive property exploration, detailed property information access, seamless booking status checks, and password updates. The technological scope encompasses a robust stack featuring MySQL for data management, JPA, REST, EJBs, Jersey Client, CDI beans, JSF for enhanced functionality and user interfaces, and JWT Authentication for a secure authentication mechanism.

Objectives:

Efficient Property Management: Enable administrators to efficiently manage properties by providing tools for role creation, property CRUD operations, and streamlined booking management. This ensures a well-organized and up-to-date property listing.

Enhanced User Experience: Prioritize user experience by offering a user-friendly interface for property exploration, allowing users to easily access and understand property details. The platform aims to foster user engagement through a seamless review submission process.

Secure Authentication: Implement JWT Authentication Mechanism to ensure a secure and reliable user authentication process. This objective is crucial for maintaining the integrity and confidentiality of user data on the HouseMarket Place platform.

Comprehensive Documentation: Provide detailed and comprehensive documentation to empower users and administrators in navigating the platform effectively. This documentation will serve as a guide for understanding and maximizing the features of HouseMarket Place.

Technology Stack Optimization: Leverage a powerful technology stack, including MySQL, JPA, REST, EJBs, Jersey Client, CDI beans, and JSF, to deliver a robust and scalable platform. The objective is to ensure the seamless functioning of the platform and its adaptability to future enhancements.

Community Engagement: Encourage user participation and feedback through the review feature, fostering a sense of community on the HouseMarket Place platform. Admins can use user reviews to enhance the overall quality and credibility of the listed properties.

1.4 Project Goal

The overarching goal of the HouseMarket Place project is to revolutionize the real estate experience by creating a robust, user-friendly platform that seamlessly connects property administrators and users. The primary objectives supporting this goal are to streamline property management, enhance user experience, ensure secure authentication, provide comprehensive documentation, optimize the technology stack, and foster community engagement. By achieving these objectives, HouseMarket Place aims to attain the following project goals:

Platform Excellence: Strive for excellence in platform functionality, ensuring that property administrators have efficient tools for managing properties, bookings, and user interactions. The goal is to establish HouseMarket Place as a go-to platform for property management needs.

User Satisfaction: Prioritize user satisfaction by offering an intuitive and engaging interface for property seekers. The goal is to create an environment where users can easily discover, evaluate, and book properties, ultimately enhancing their overall experience on the platform.

Security Assurance: Guarantee the security and integrity of user data through the implementation of a robust JWT Authentication Mechanism. The goal is to build trust among users by ensuring that their personal information is handled with the utmost care and security.

Documentation Excellence: Provide comprehensive and user-friendly documentation that serves as a valuable resource for both administrators and users. The goal is to empower users to navigate the platform effectively, leading to a positive and efficient user experience.

Technological Innovation: Strive for technological innovation by optimizing the chosen stack, including MySQL, JPA, REST, EJBs, Jersey Client, CDI beans, and JSF. The goal is to create a scalable and adaptable platform that can evolve with emerging technologies and industry standards.

Community Building: Establish a vibrant and engaged community of users through the review feature, creating a collaborative environment. The goal is to foster a sense of community where users actively contribute to the platform's growth and improvement.

1.5 Project Category:

HouseMarket Place falls within the category of Real Estate Management Platforms, providing a comprehensive solution for property administrators and users alike. This project integrates advanced technologies and features to streamline property management processes, enhance user interactions, and establish a secure and efficient online marketplace for properties. The platform spans both the administrative and user sides, offering functionalities such as role creation, property CRUD operations, booking management, and user reviews. With a focus on user experience, security, and community engagement, HouseMarket Place contributes to the evolution of real estate management platforms by combining technological innovation with user-centric design. This project category encapsulates the broader scope of real estate technology, emphasizing the seamless integration of property management, user engagement, and secure authentication within a unified platform.

**2 Environment description**

2.1 Hardware and Software Requirement

For Admin

|  |  |
| --- | --- |
| **Software** | **Apache Netbean 17** |
| **Database** | Mysql |
| **Operating System** | Windows or any equv |
| **RAM** | 512 MB(Minimum) |
| **Microprocessor** | Any Microprocessor can be used but latest would be better for performance |
| **Hard Disk Space** | 1 GB (Minimum) |

**For User**

|  |  |
| --- | --- |
| **Operating System** | Windows or any equv |
| **RAM** | 512 MB(Minimum) |
| **Microprocessor** | Any Microprocessor can be used but latest would be better for performance |
| **Hard Disk Space** | 1 GB (Minimum) |

For documentation

|  |  |
| --- | --- |
| Software | MS Word |

2.2 Application Area Of System

Real Estate Agencies and Brokers:

Real estate agencies and brokers can leverage HouseMarket Place to efficiently manage and showcase their property listings.

The platform's role management and property management features are particularly valuable for organizing and categorizing listings.

Property Management Companies:

Companies that specialize in property management can use the system to streamline their operations, from creating and updating property details to managing bookings.

Property Rental Services:

Platforms offering property rental services, both short-term and long-term, can benefit from HouseMarket Place's booking management capabilities.

The role creation feature allows for clear differentiation between properties available for rent and those for sale.

Real Estate Developers:

Real estate developers can utilize the platform to display and promote their properties, manage bookings, and gather valuable user reviews for marketing purposes.

Real Estate Investment Platforms:

Investors and investment platforms interested in real estate can use HouseMarket Place to explore potential investment opportunities, view property details, and track booking status.

Community and Review Platforms:

The user review management feature makes HouseMarket Place suitable for community-driven real estate platforms, where users can share their experiences and insights.

Technology Providers in Real Estate:

Companies providing technology solutions to the real estate industry can integrate or customize HouseMarket Place to offer advanced property management features to their clients.

Educational Institutions and Research:

Academic institutions and researchers studying real estate trends and behaviors may find value in the platform's user data and review features for analysis.

Government and Regulatory Bodies:

Government entities involved in real estate regulation and oversight may use HouseMarket Place for monitoring and ensuring compliance within the industry.

General Real Estate Consumers:

Everyday users and property seekers looking for a user-friendly platform to explore, review, and book properties can directly benefit from HouseMarket Place.

1. **Analysis Report**

**3.1** Drawback of the current System:-

Booking Management Complexity:

Drawback: While the system includes booking management features, the complexity of managing bookings, especially in scenarios involving multiple properties or overlapping dates, could be a potential drawback. Administrators may find it challenging to efficiently handle intricate booking scenarios.

User Review Moderation Challenges:

Drawback: The ability for administrators to manage user reviews is crucial, but challenges may arise in terms of moderation. Without robust moderation tools, the system could face issues related to spam, inappropriate content, or biased reviews, affecting the overall credibility of the platform

Limited Property Details Customization:

Drawback: The system allows users to view property details, but there may be limitations in terms of customization. Property developers or agencies with specific requirements for showcasing property details may find the system restrictive.

Potential Database Performance Issues:

Drawback: Depending on the scale of data and user interactions, the use of MySQL as the database may encounter performance issues. In scenarios with a large number of concurrent users or extensive data, the system may face challenges in terms of responsiveness and query optimization.

Limited User Management Features:

Drawback: While the system allows administrators to manage users, the extent of user management features may be limited. Advanced user management functionalities, such as detailed user analytics, personalized user communication, or user segmentation, may be lacking.

3.2 Requirement Analysis:-

Admin Module:

1. Property Management:

a. Add Property:

- Admin should be able to add new properties for rent or sale.

- Specify details like property type (house, flat, etc.), location, size, price, and other relevant information.

b. Update Property Status:

- Enable admin to update the details of existing properties.

c. Delete Property:

- Admin should have the ability to remove a property from the system.

2. User Management:

a. View Users:

- Admin can view a list of registered users.

b. Manage Users:

- Ability to block or unblock users.

- View user details and history.

3. Booking Management:

a. View Bookings:

- Admin should be able to view a list of property bookings made by users.

b. Manage Bookings:

- Confirm or reject property bookings.

- View booking details and user information.

4. Feedback Management:

a. View Feedback:

- Admin can view feedback provided by users.

5. Security:

a. Admin Authentication:

- Secure login for admin with username and password.

b. Role-Based Access:

- Differentiate between admin and regular user roles.

6. System Management:

a. Dashboard:

- Provide a dashboard for a quick overview of property listings, user activity, and bookings.

5.2 User Module**:**

1. Property Browsing:

a. View Properties:

- Users should be able to browse and search for properties based on various criteria.

b. Property Details:

- Detailed information about each property including images, price, location, and contact details.

2. Booking:

a. Book Property:

- Users can request a booking for a specific property.

b. Booking Status:

- View the status of booking requests.

3. Feedback:

a. Provide Feedback:

- Users should be able to leave feedback for properties they have interacted with.

4. Account Management:

a. User Registration:

- Allow users to create accounts with a unique username and password.

b. Password Update:

- Users can update their passwords.

5. User Dashboard:

a. Personalized Dashboard:

- Users should have a dashboard displaying their bookings, feedback, and favorite properties.

6. Security:

a. User Authentication:

- Secure login for users.

b. Password Recovery:

- Option for users to recover their password..

3.3 Requirement Specification

1. Admin Side:

Role Management:

Admins can create roles (e.g., 'rent' and 'sale').

The role creation process is intuitive and customizable.

Property Management:

Admins can create, display, update, and delete properties.

Property management operations are responsive and user-friendly.

Booking Management:

Admins can confirm and cancel property bookings.

Detailed booking information is available for effective management.

User Review Management:

Admins have tools to manage user reviews (approval, moderation, removal).

User review management is efficient to maintain credibility.

2. User Side:

Property Exploration:

Users can browse and view available properties with ease.

Property exploration features are responsive and seamless.

Property Details Viewing:

Users have access to detailed property information when viewing a listing.

Property details viewing is quick and efficient.

Review Submission:

Users can submit reviews for properties.

Review submission is user-friendly with quality feedback validation.

Booking Status Check:

Users can check the status of their property bookings.

Booking status checks are real-time and easily accessible.

Password Update:

Users can update their passwords.

Password update processes adhere to security best practices.

3. Technology Stack:

Database:

The system uses MySQL for efficient data storage and retrieval.

Database operations are optimized for performance and scalability.

Backend Technologies:

JPA is implemented for Java-based persistence.

RESTful APIs are used for seamless client-server communication.

EJBs handle business logic.

Jersey Client manages RESTful communication.

CDI beans are applied for efficient bean management.

JSF is utilized for user interface development.

Backend technologies are selected for optimal performance and maintainability.

Authentication Mechanism:

The system implements JWT Authentication for secure user authentication.

Authentication processes adhere to industry standards.

3.4 Proposed system

1. Admin Side:

Role Management:

Admins can create roles (e.g., 'rent' and 'sale').

The role creation process is intuitive and customizable.

Property Management:

Admins can create, display, update, and delete properties.

Property management operations are responsive and user-friendly.

Booking Management:

Admins can confirm and cancel property bookings.

Detailed booking information is available for effective management.

User Review Management:

Admins have tools to manage user reviews (approval, moderation, removal).

User review management is efficient to maintain credibility.

User Management:

Admins can manage user accounts (create, modify, deactivate).

User management operations are secure and scalable.

Dashboard Management:

The system provides a comprehensive dashboard for administrators.

The dashboard offers real-time data updates and is visually intuitive.

2. User Side:

Property Exploration:

Users can browse and view available properties with ease.

Property exploration features are responsive and provide a seamless experience for users.

Property Details Viewing:

Users have access to detailed property information when viewing a listing.

Property details viewing is quick and efficient.

Review Submission:

Users must be able to submit reviews for properties.

Review submission is user-friendly with quality feedback validation.

Booking Status Check:

Users can check the status of their property bookings.

Booking status checks are real-time and easily accessible.

Password Update:

Users can update their passwords.

Password update processes adhere to security best practices.

3. Technology Stack:

Database:

The system uses MySQL for efficient data storage and retrieval.

Database operations are optimized for performance and scalability.

Backend Technologies:

JPA is implemented for Java-based persistence.

RESTful APIs are used for seamless client-server communication.

EJBs (Enterprise JavaBeans) handle business logic.

Jersey Client is utilized for handling RESTful communication.

CDI beans (Contexts and Dependency Injection) are applied for efficient bean management.

JSF (JavaServer Faces) is used for user interface development.

Backend technologies are selected for optimal performance and maintainability.

Authentication Mechanism:

The system implements JWT Authentication for secure user authentication.

Authentication processes adhere to industry standards.

3.5 Advantages of **Proposed** System

Efficient Property Management:

Advantage: Admins can easily create, update, and delete property listings, streamlining property management operations.

Benefit: Ensures a well-maintained and up-to-date property portfolio, enhancing the overall user experience.

Transparent Booking Processes:

Advantage: The system facilitates efficient booking management, allowing admins to confirm or cancel bookings.

Benefit: Increases transparency in property transactions, providing users with a clear understanding of their booking status.

Community-Driven Reviews:

Advantage: Admins can manage user reviews, ensuring authenticity and quality.

Benefit: Fosters a community-driven environment where users can provide valuable feedback, enhancing trust and credibility.

Flexible Role Management:

Advantage: Admins can create roles (e.g., 'rent' and 'sale'), allowing for flexible categorization of properties.

Benefit: Provides adaptability to different types of real estate offerings, catering to a diverse market.

Secure JWT Authentication:

Advantage: The implementation of JWT Authentication ensures secure user authentication.

Benefit: Enhances the overall security of user accounts and sensitive data, protecting against unauthorized access.

Real-Time Booking Status Checks:

Advantage: Users can check the real-time status of their property bookings.

Benefit: Enhances user satisfaction by providing instant access to booking information, reducing uncertainty.

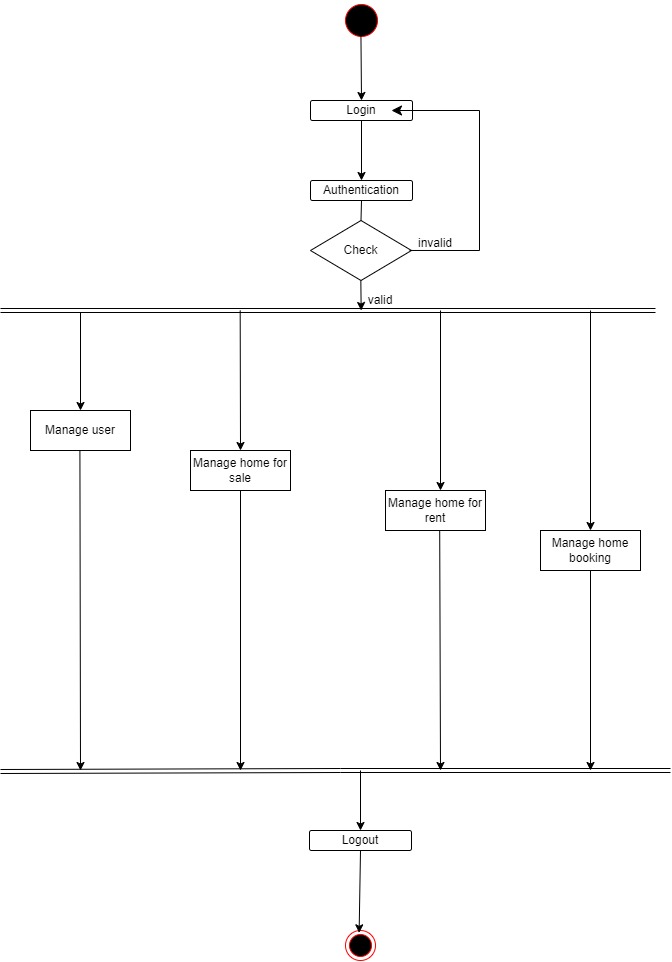
Detailed Property Information:

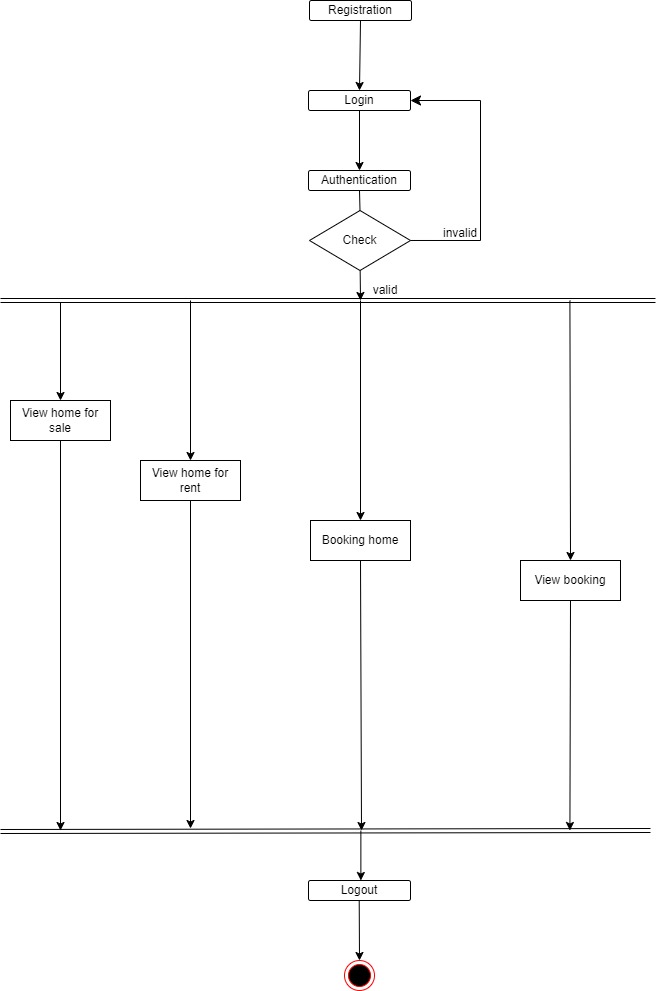
Advantage: Users have access to detailed property information when viewing specific listings.

Benefit: Enables users to make informed decisions by providing comprehensive details about each property.

**4 Software Diagram**

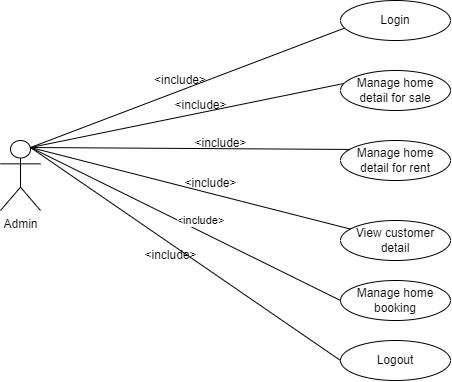
4.1 Admin & User Activity Diagram



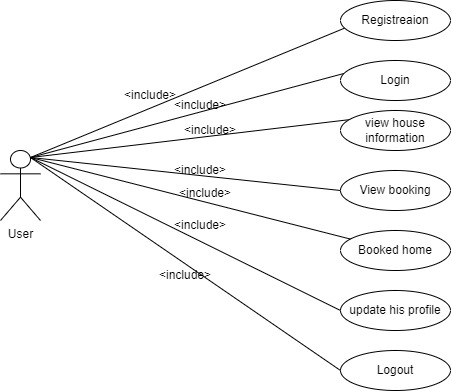


4.2 Admin & User usecase diagram

Admin:-

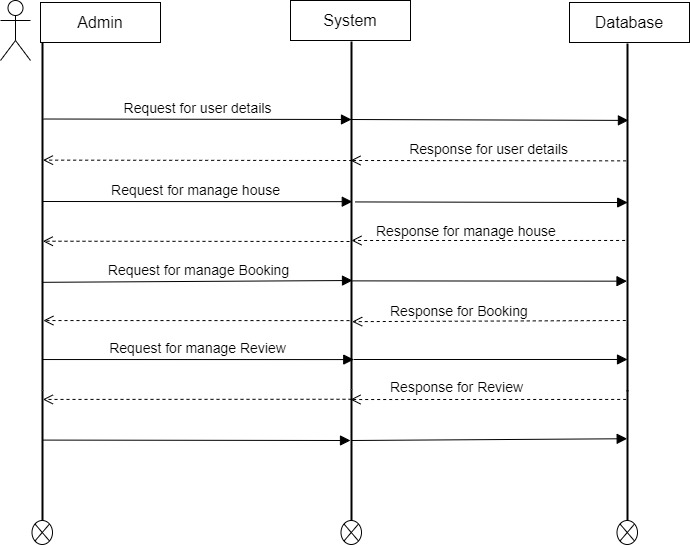


User:-

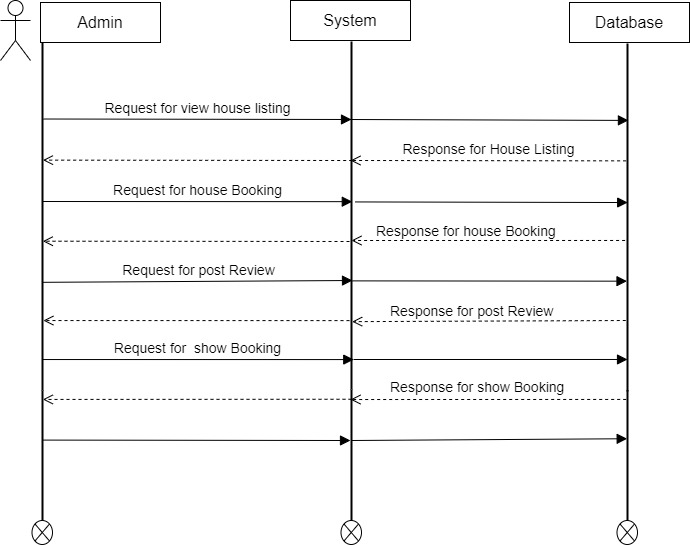


4.3 Admin & User Sequence Diagram

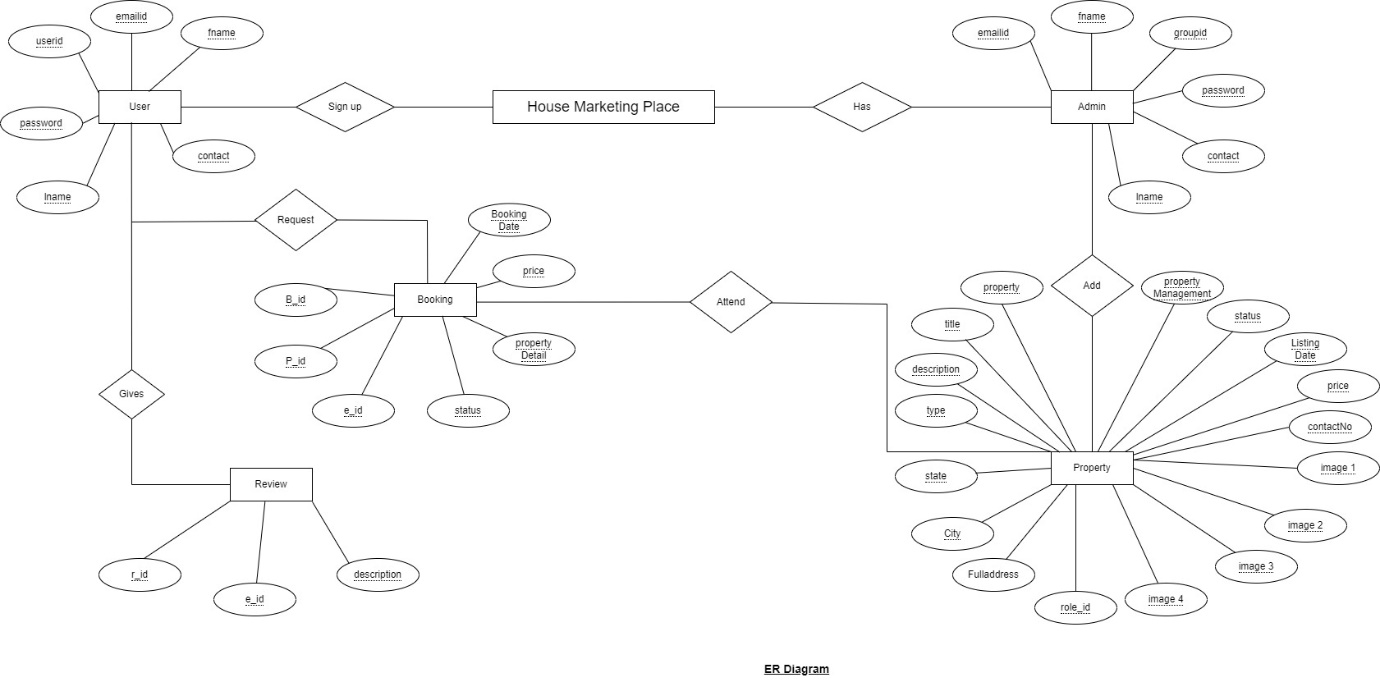
Admin:-



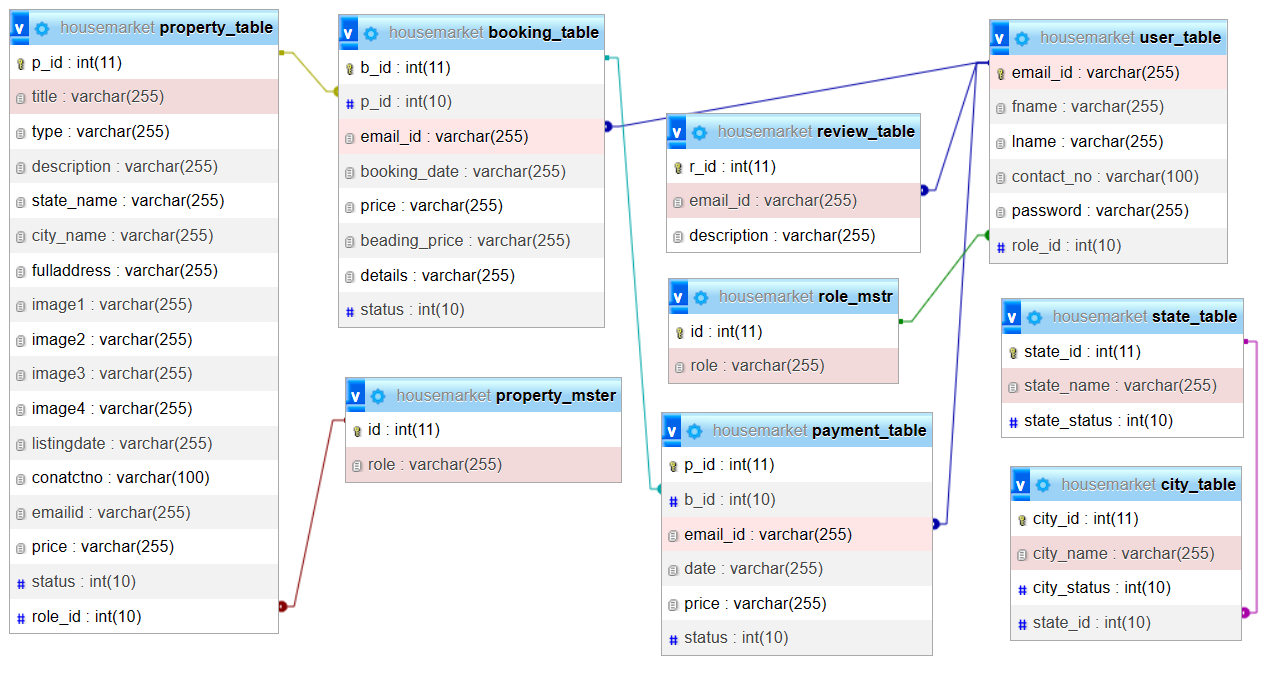
User:-



4.4 E-R Diagram



4.5 Class Diagram



**5.DataBase Design**

**DataBase Design**

Property\_table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Description | Constraint | Null/Not Null |
| p\_id | INT(11) | Unique identifier for the property | Primary Key, Auto Inc | Not Null |
| title | Varchar(255) | Title for property |  | Not Null |
| type | Varchar(255) | Type of property |  | Not Null |
| description | Varchar(255) | Description of the  property |  | Not Null |
| state\_name | Varchar(255) | State for property location |  | Not Null |
| city\_name | Varchar(255) | cityfor property location |  | Not Null |
| fulladdress | Varchar(255) | Full Address for property location |  | Not Null |
| image1 | Varchar(255) | Image1 for property |  | Not Null |
| image2 | Varchar(255) | image2 for property |  | Not Null |
| image3 | Varchar(255) | image3 for property |  | Not Null |
| image4 | Varchar(255) | Image4 for property |  | Not Null |
| listingdate | Varchar(255) | listingDate of property |  | Not Null |
| conatctno | Varchar(100) | Contactno of property Owner |  | Not Null |
| emailid | Varchar(255) | Emailid of property owner |  | Not Null |
| price | Varchar(255) | Price for property |  | Not Null |
| status | Int(10) | Status for property |  | Not Null |
| role\_id | Int(10) | Foreign key from property\_mster | Foreign Key | Not null |

**Property\_mster**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| column Name | Data Type | Description | Constraint | Null/Not Null |
| id | INT(11) | Unique identifier for property\_mster | Primary Key | Not Null |
| role | Varchar(255) | Role for property\_mster |  | Not Null |

**booking\_table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Description | Constraint | Null/Not Null |
| b\_id | int(11) | Unique identifier for booking\_table | Primary Key, Auto Inc | Not Null |
| P\_id | Int(11) | Foreign key for Property\_table | Foreign Key | Not Null |
| email\_id | Varchar(255) | Foreign Key from User\_table | Foreign Key | Not Null |
| booking\_date | Varchar(255) | BookingDate for Booking |  | Not Null |
| Price | Varchar(255) | Price for Booking |  | Not Null |
| beading\_price | Varchar(255) | Bprice for Booking |  | Not Null |
| details | Varchar(255) | Details for Booking |  | Not Null |
| status | Int(10) | Status for Booking |  | Not Null |

**City\_table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Description | Constraint | Null/Not Null |
| city\_id | int(11) | Unique identifier for city\_table | Primary Key, Auto Inc | Not Null |
| city\_name | Varchar(255) | name for city\_table |  | Not Null |
| city\_status | Int(10) | Status for city\_table |  | Not Null |
| state\_id | Int(10) | Foreign key from state\_id | Foreign Key | Not Null |

**Review\_table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Description | Constraint | Null/Not Null |
| r\_id | int(11) | Unique identifier for city\_table | Primary Key, Auto Inc | Not Null |
| email\_id | Varchar(255) | Foreign Key from User\_table | Foreign Key | Not Null |
| description | Varchar(255) | Description for review\_table |  | Not Null |

**Role\_mstr**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Description | Constraint | Null/Not Null |
| id | int(11) | Unique identifier for  Role\_mstr | Primary Key, Auto Inc | Not Null |
| role | Varchar(255) | Role for Role\_mstr |  | Not Null |

**State\_table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Description | Constraint | Null/Not Null |
| state\_id | int(11) | Unique identifier for  State\_table | Primary Key, Auto Inc | Not Null |
| state\_name | Varchar(255) | Name for state |  | Not Null |
| state\_status | Int(10) | Status for state |  | Not Null |

**User\_table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | Data Type | Description | Constraint | Null/Not Null |
| email\_id | Varchar(255) | Unique identifier for  User\_table | Primary Key, Auto Inc | Not Null |
| fname | Varchar(255) | Firstname of user |  | Not Null |
| lname | Varchar(255) | Lastname of user | Foreign Key | Not Null |
| contact\_no | Varchar(255) | Contact no of user |  | Not Null |
| Password | Varchar(255) | password no of user |  | Not Null |
| role\_id | Int(10) | Foreign key from role\_mstr | Foreign Key | Not Null |

6 Test Case Design

1. Register page

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | Inputted Value | Expected Result | Actual Result | Pass/Fail |
| 1 | Emailid(Required,formatted) | "john.doe@example" | Valid Emailid is accepted. | Redirect to login | Pass |
| 2 | Password (Min-6, Max-10) | "P@ss123" | Password meets criteria and is accepted. | Redirect to login | Pass |
| 3 | Firstname (Required,) | “johndoe” | Email is required and must be in a valid format. | Redirect to login | Pass |
| 4 | Lastname (Required) | "1doe" | LastName is Required | Redirect to login | Pass |
| 5 | Contactno (Required) | “1234567890” | Phone number must be 10 digits and is accepted. | Redirect to login | Pass |

1. Login page

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | Inputted Value | Expected Result | Actual Result | Pass/Fail |
| 1 | Emaild(Required) | "JohnDoe@gamil.com" | Valid emailid is accepted. | Redirect to Home page | Pass |
| 2 | Password (Required) | "P@ss123" | Valid password is accepted. | Redirect to Home page | Pass |
| 3 | Emailid Missing | "" | Display error: "emailid is required." | Give an error message | Fail |
| 4 | Password Missing | "" | Display error: "Password is required." | Give an error message | Fail |
| 5 | Incorrect Emailid | "User123" | Display error: "Invalid emailid or password." | Give an error message | Fail |
| 6 | Incorrect Password | "WrongP@ss" | Display error: "Invalid emailid or password." | Give an error message | Fail |
| 7 | Emailid with Spaces | "John Doe" | Display error: "emailid cannot contain spaces." | Redirect to Home page | Pass |

**7 Project Images**

**7.2 User Interface**

**Common Sign-in Page**

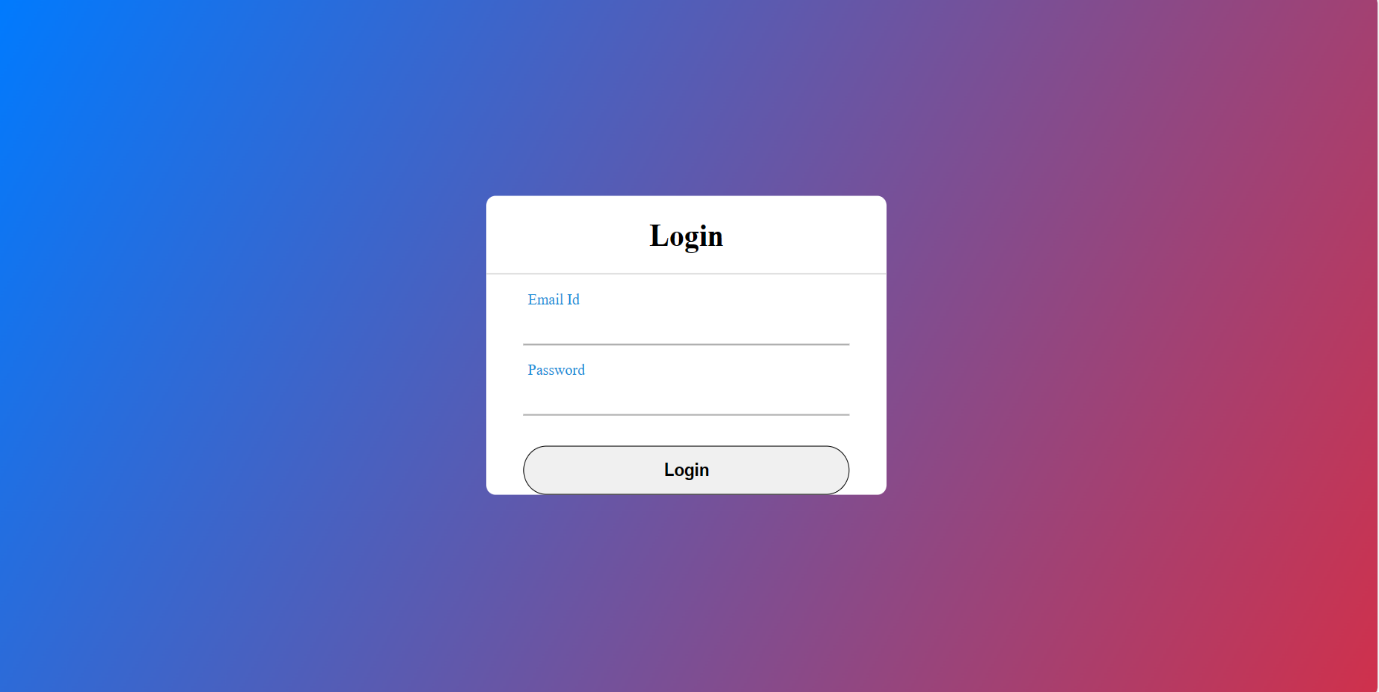


Figure 1: This is a common login page

**Register Page**

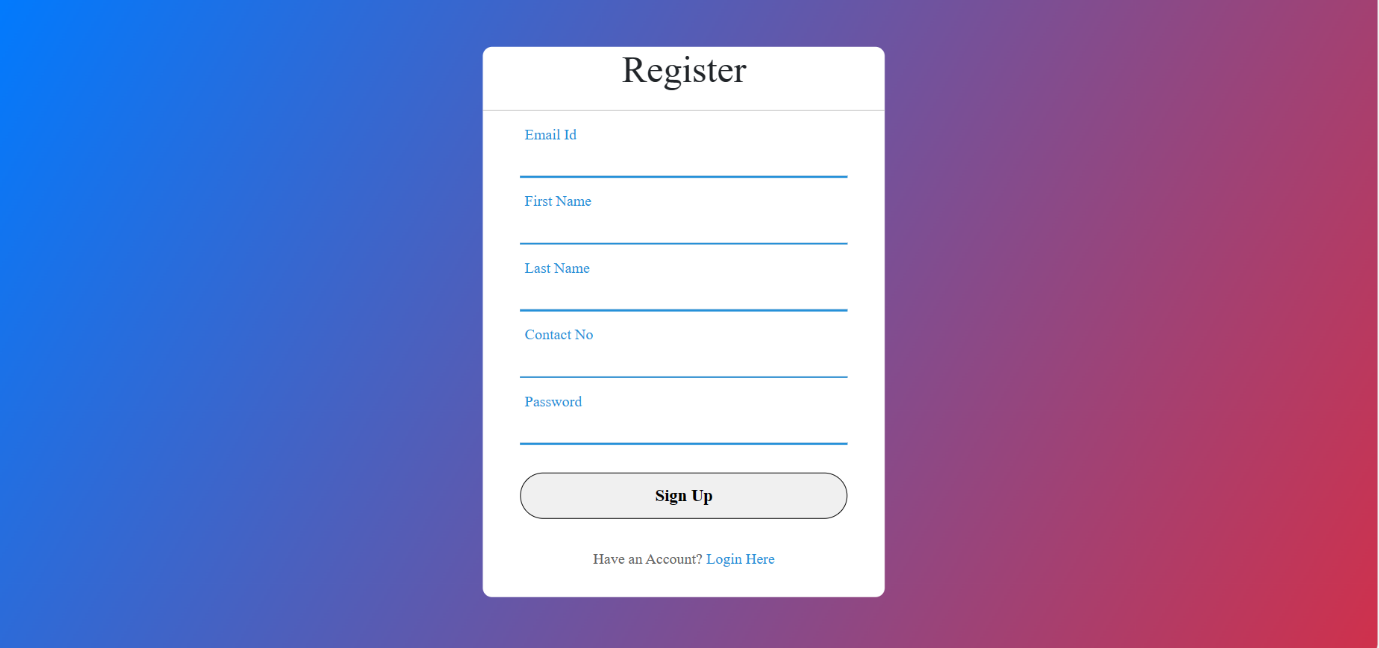
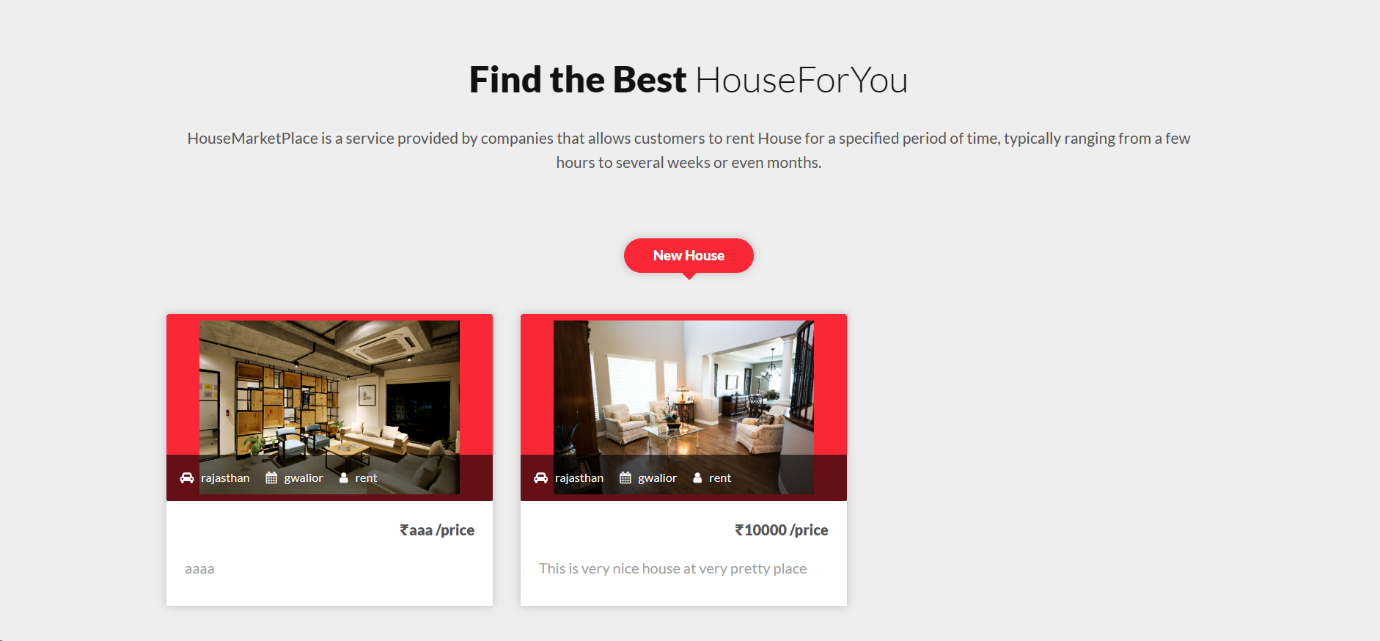
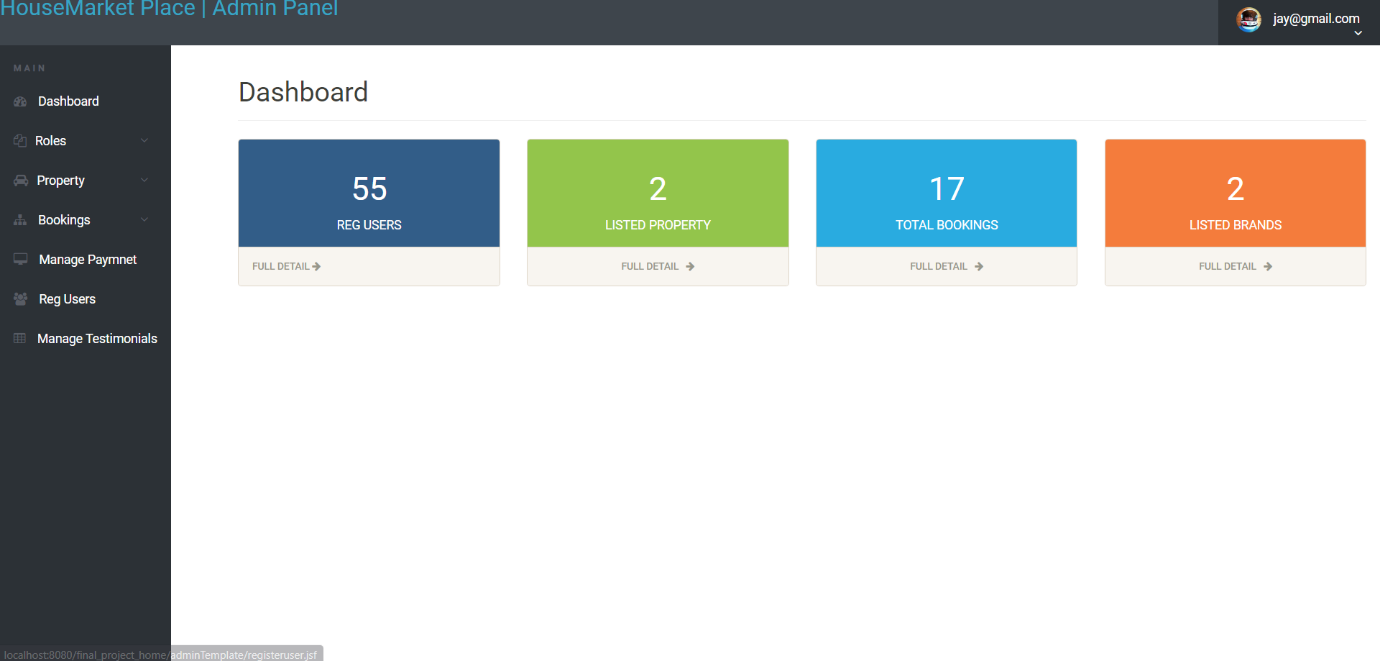


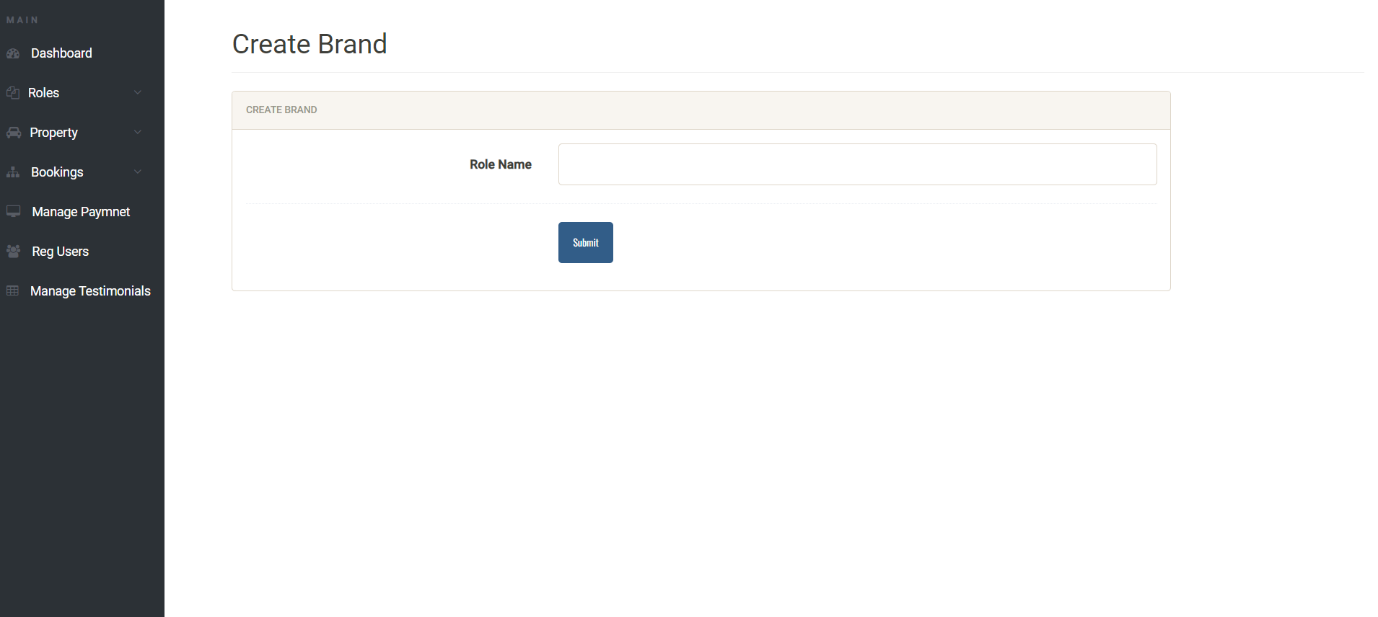
Figure 2: This is a Registration page



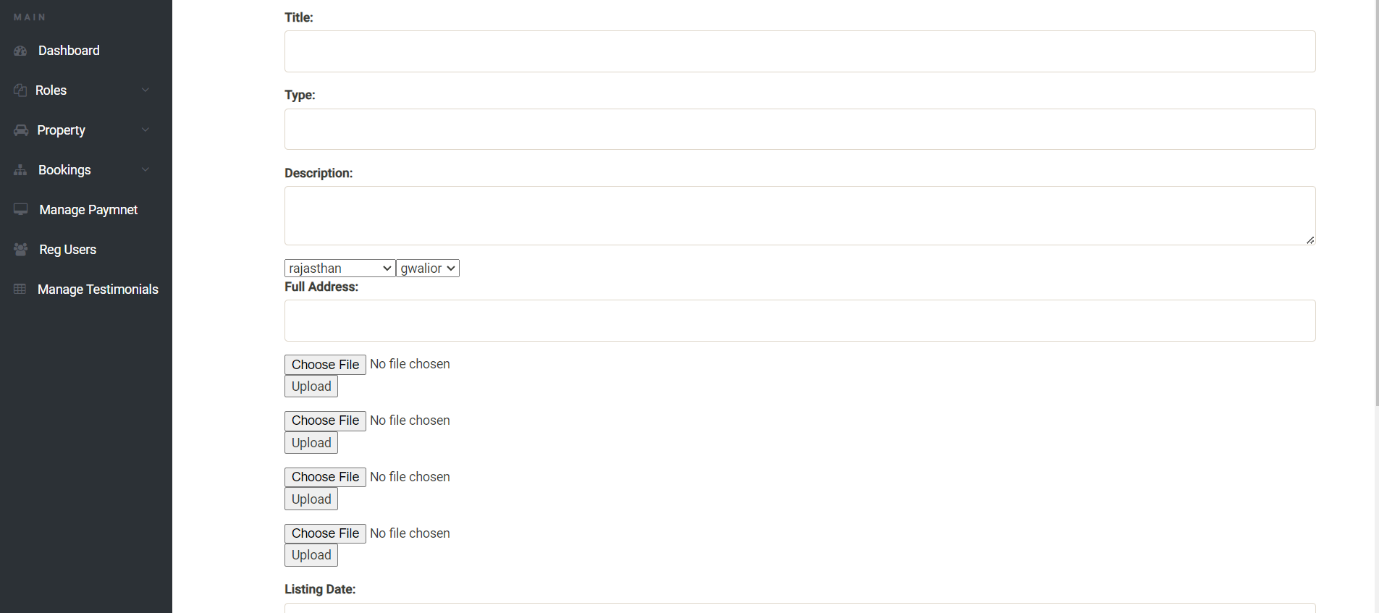
User: User home

Admin:DashBoard

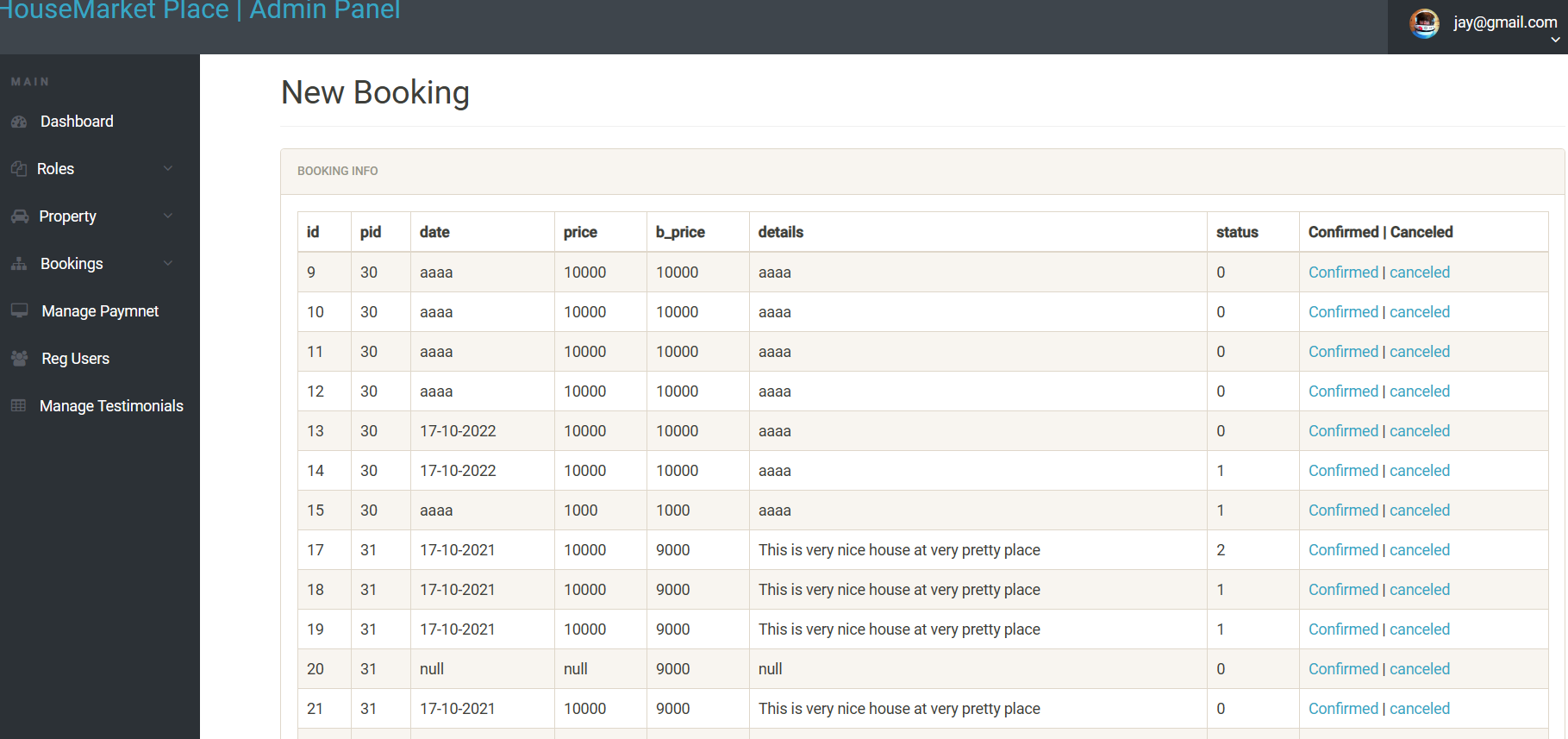




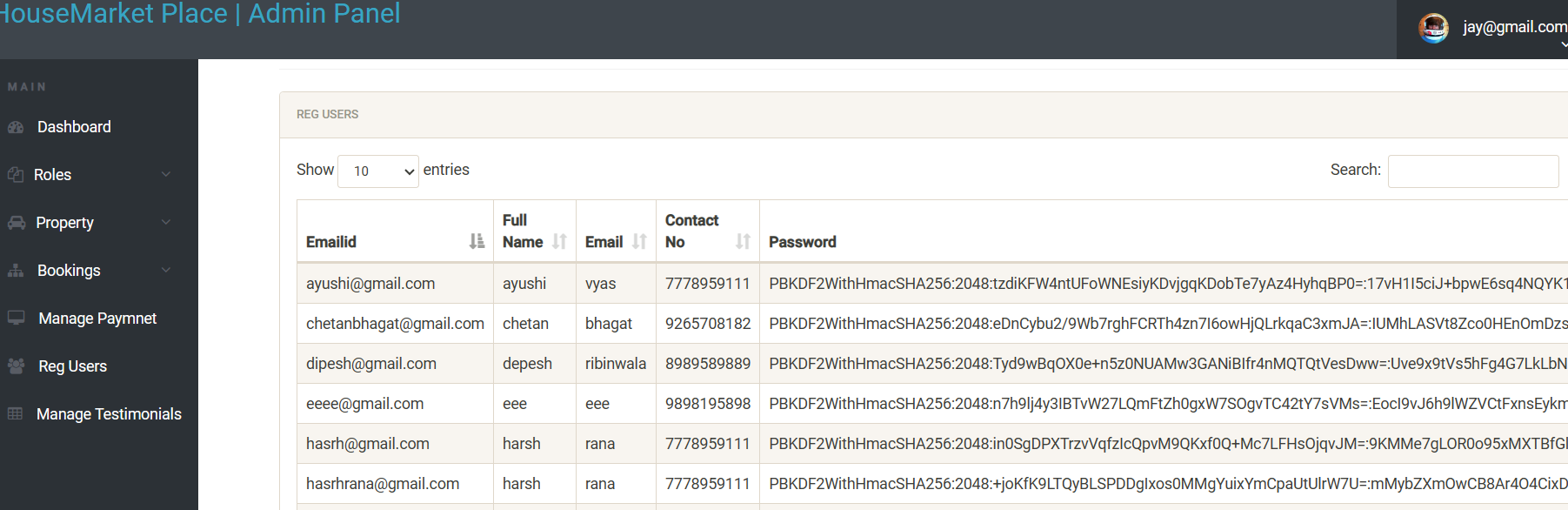
Admin:- createbrand(like rent and sale)



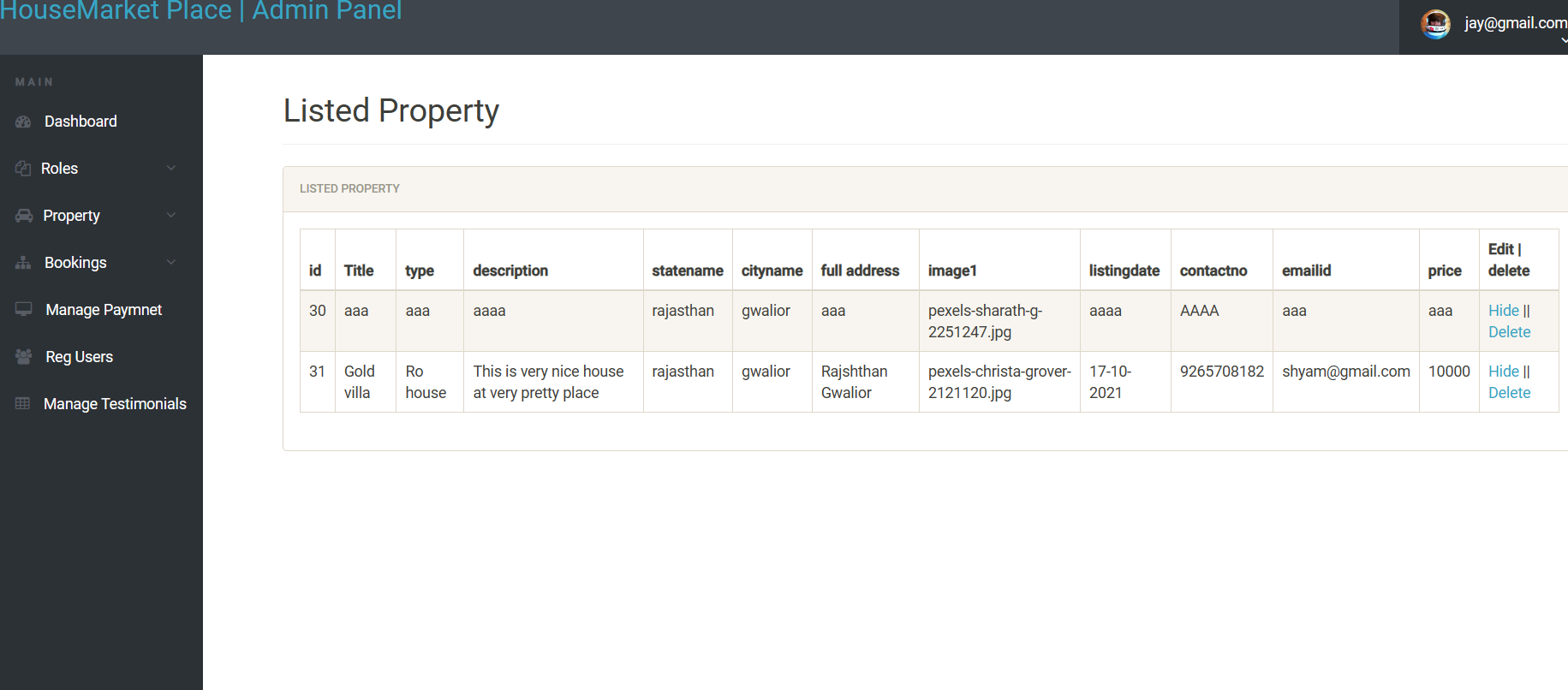
Admin: insert Property



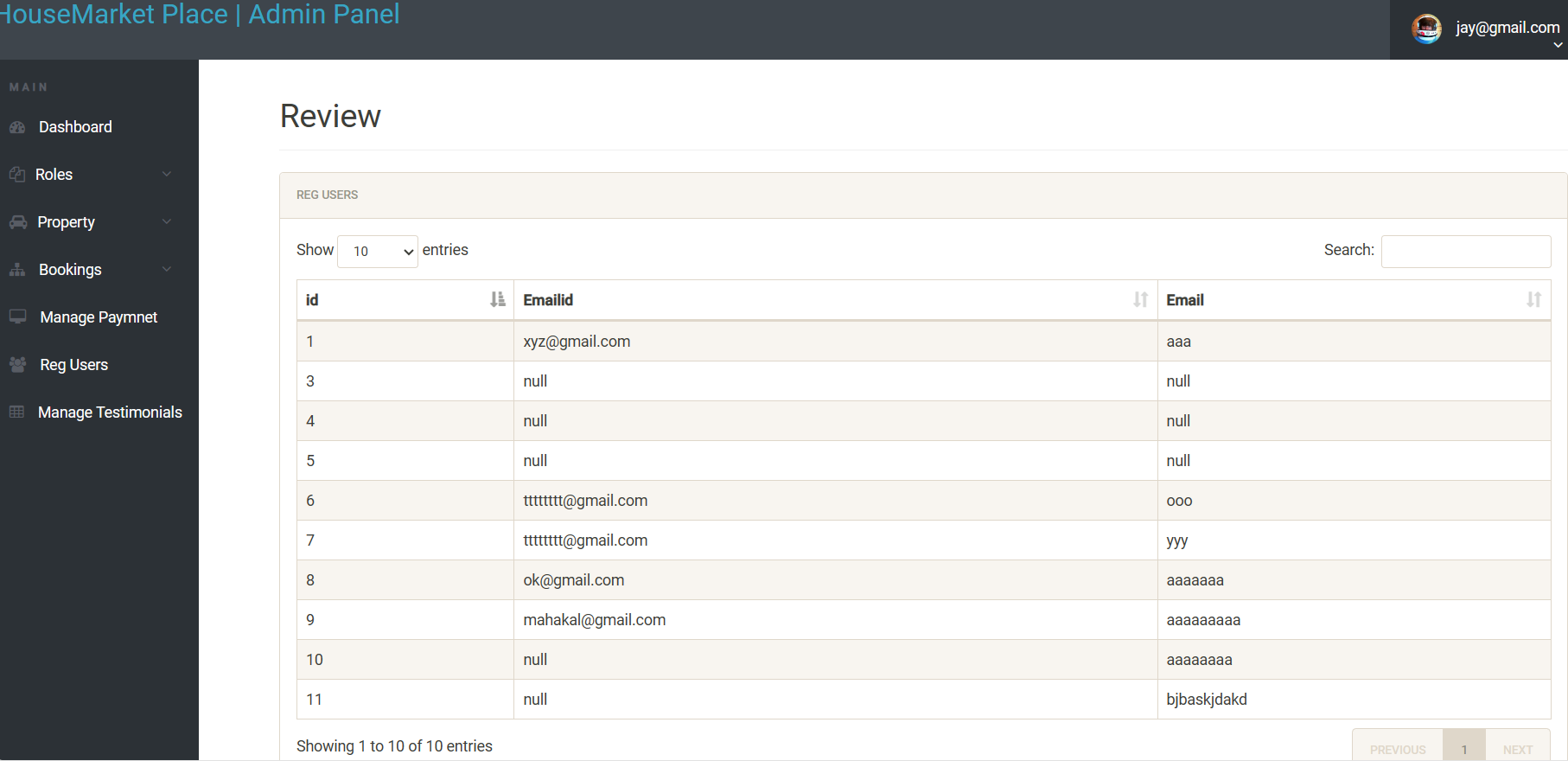
Admin:Manage Booking



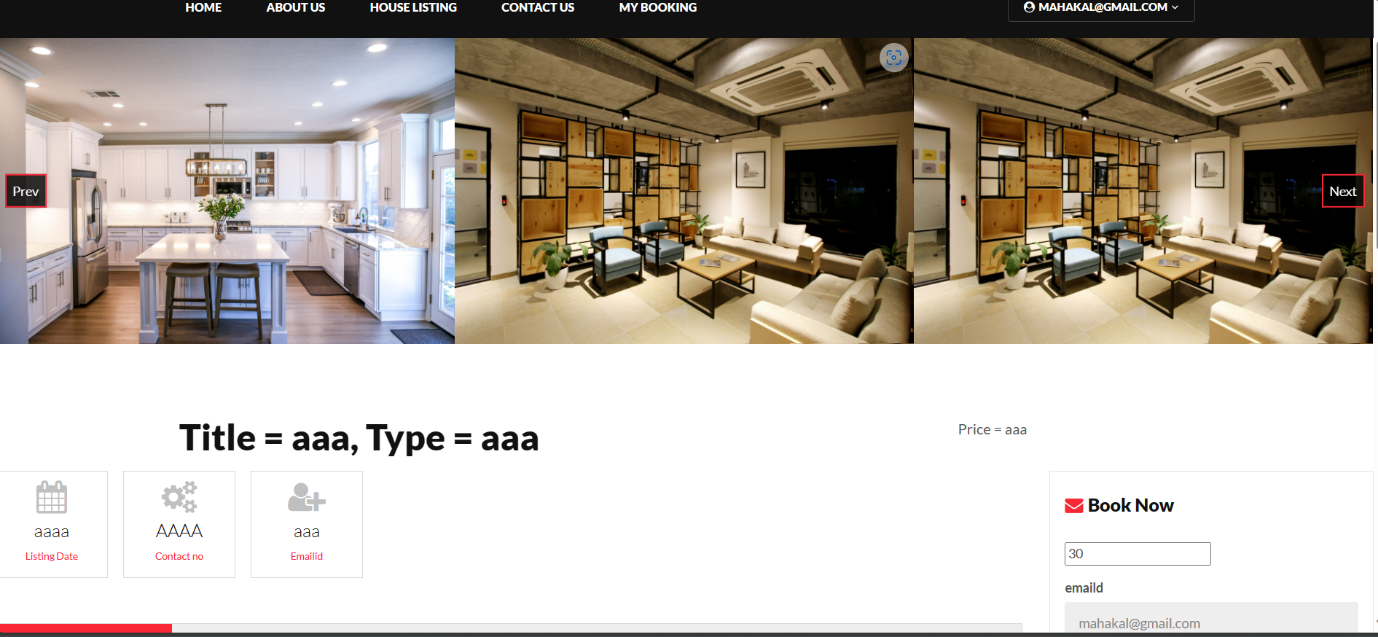
Admin:Manage User



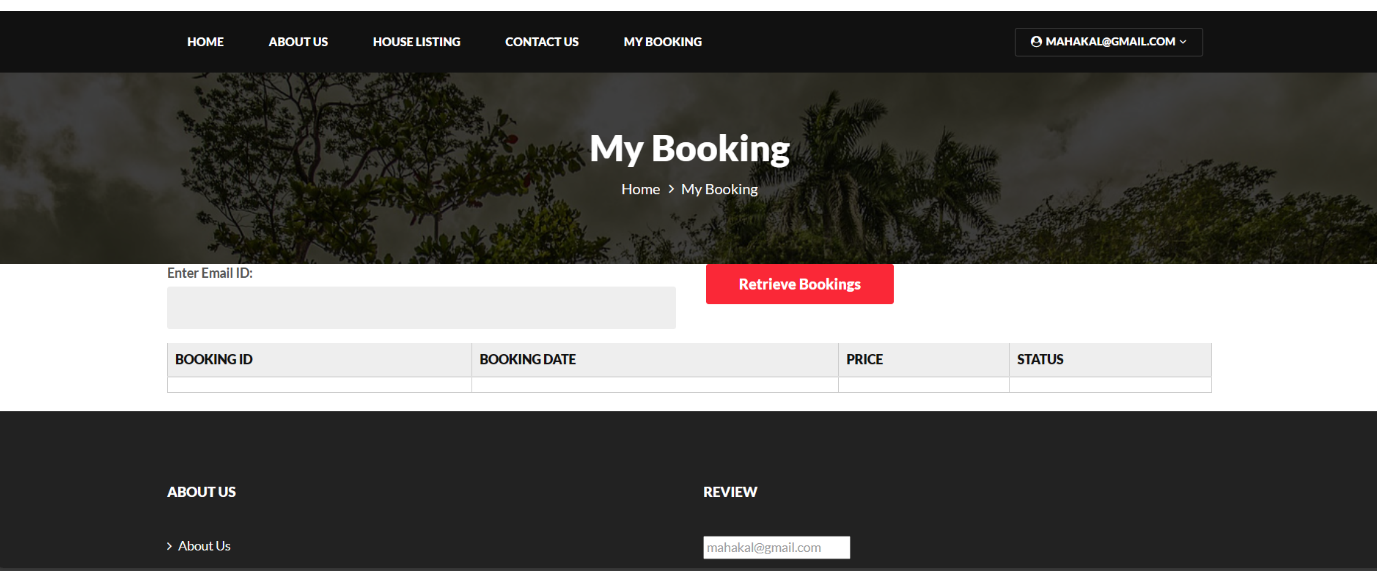
**Admin:Manage Listed Property**



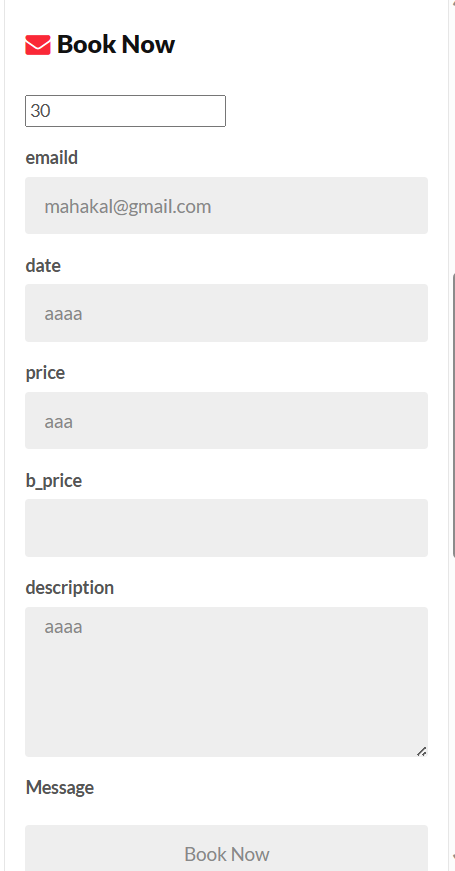
Admin:Show Review



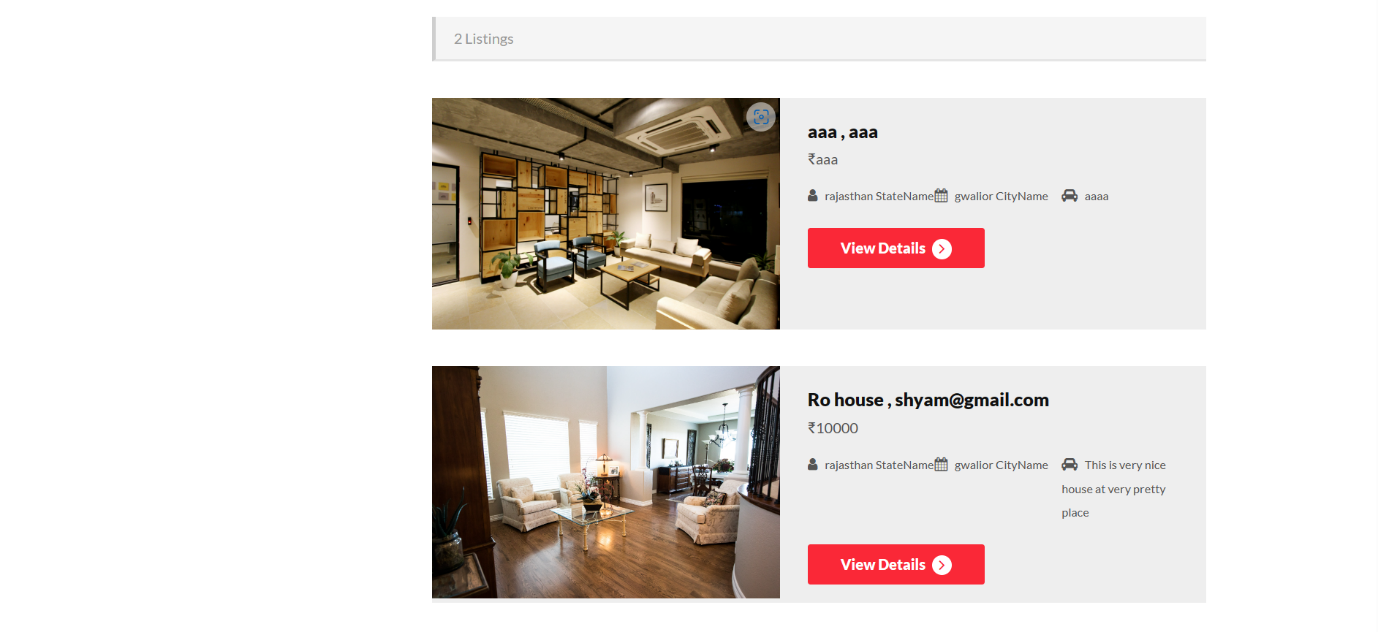
User:-Show Particular Property



User:-Show his booking



User:-Book Property



User:-Show House Listing

**8.Future Scope**

Advanced Search and Filters:

Enhance the property search functionality with advanced filters, allowing users to refine their search based on criteria such as price range, amenities, location, etc.

Real-Time Availability and Pricing:

Provide real-time updates on property availability and pricing to ensure users have the latest information when making booking decisions.

Mobile App Development:

Develop dedicated mobile applications for both users and administrators to enhance accessibility and convenience.

**9.References**

**References I have referred to during project development :**

* <https://github.com/kamlendu>
* <https://stackoverflow.com/>
* <https://netbeans.apache.org/tutorial/main/kb/docs/javaee/>
* <https://hiringrewards.com/>