A Project Report

On

“RESPONSIVE REAL ESTATE PORTAL”

Submitted to the

Department of Computer Applications

In partial fulfilment of the Course

Integrated Master of Computer Applications

Under the guidance of

Mr. MARIADAS RONNIE C P

Project Done by

TANIA SHINE

(Reg no: 193241010113)

DEPARTMENT OF COMPUTER APPLICATIONS

SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT

December-2023

SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT

BONAFIDE CERTIFICATE

Certified that the Project Work entitled

“RESPONSIVE REAL ESTATE PORTAL”

is a bonafide work done by

Tania Shine

In partial fulfillment of the requirement for the Award of

INTEGRATED MASTER OF COMPUTER APPLICATIONS

Degree From
Mahatma Gandhi University, Kottayam
(2019-2024)

Head of Department *Project Guide*

Submitted for the Viva-Voce Examination held on…………………………………….

External Examiner1 **External Examiner2**

(Name & Signature) (Name & Signature)

SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT

CERTIFICATE

This is to certify that the project entitled **"RESPONSIVE REAL ESTATE PORTAL"** has been successfully carried out by **TANIA SHINE** (Reg no: 193241010113) in partial fulfilment of the Course **INTEGRATED MASTER OF COMPUTER APPLICATIONS.**

Date: **HEAD OF DEPARTMENT**

SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT

CERTIFICATE

This is to certify that the project entitled **“RESPONSIVE REAL ESTATE PORTAL”** has been successfully carried out by **TANIA SHINE** (Reg no: 193241010113) in partial fulfilment of the course **INTEGRATED MASTER OF COMPUTER APPLICATIONS** under my guidance.

Date:

Mariadas Ronnie C P
INTERNAL GUIDE

SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT

DECLARATION

I, **TANIA SHINE**, hereby declare that the project work entitled **“RESPONSIVE REAL ESTATE PORTAL”** is an authenticated work carried out by me under the guidance of **Mr. Mariadas Ronnie C P** for the partial fulfilment of the course **INTEGRATED MASTER OF COMPUTER APPLICATIONS**. This work has not been submitted for similar purpose anywhere else except to **SCMS SCHOOL OF TECHNOLOGY AND MANAGEMENT**, affiliated to **M.G.UNIVERSITY, KOTTAYAM** .

I understand that detection of any such copying is liable to be punished in any way the school deems fit.

Date:

Place: **TANIA SHINE**

ACKNOWLEDGEMENT

An endeavour over a long period can be successful with the advice, support and blessings of many well-wishers. To acknowledge all of them is a blissful opportunity showered upon me by the Almighty. With great pleasure and privilege, I present here with full satisfaction, Project report on “RESPONSIVE REAL ESTATE PORTAL”.

I take this Opportunity to express my gratitude and sincere thanks to all who helped me to complete this project successfully. I first thank God Almighty, who showered his immense blessing on my effort. I express my gratitude and sincere thanks to the Director Dr. Indu Nair for her kind consideration and valuable guidelines throughout the course of our project work.

I sincerely express my gratitude to Dr. Anjana S Chandran, HOD MCA. I also extend my sincere gratitude to Mr. Mariadas Ronnie C P my project guide, for his valuable guidance, cooperation, and constant encouragement throughout the project work.

I also thank my friends and well-wishers, who have provided their wholehearted support to me in this exercise. I believe that this endeavour has prepared me for taking up new challenging opportunities in future.

TANIA SHINE

Table of Contents

1. EXECUTIVE SUMMARY

‘ESquare – A Responsive Real Estate Portal’ is designed to make the existing manual system- automatic, by using computerized and full-edged computer software, so that all the valuable data and information can be stored for a longer period with easy access and manipulation. This application can maintain and view computerized records without getting redundant entries. The project helps to manage properties posted, providing better services for the client.

It is a software application that manages the operational activities of a real estate business. It is an online real estate software application that manages the overall operational activities and processes, starting from the management of the property, to the management of real estate agents and clients. It provides comprehensive reports for managing the Real Estate agency performance and efficiency, and enables the management for a better decision-making.

Users of the System

- ADMINISTRATOR
- USERS

Main Functions

Admin Features:

- Manage all users.
- Manage properties posted by users.
- Approve properties to ensure authenticity.
- Add and manage other admins.
- View messages submitted by users.
- Generate various reports based on criteria.
- See an overview of total users, admins, properties posted, and new messages.

User Features:

- Register and create an account.
- Pay for subscription and post properties.
- Manage posted properties.
- Send messages, suggestions, or feedback to the admin.
- Send inquiries to property owners.
- View inquiries received for each property.
- View and edit their profile.
- Change password and phone number visibility.

2. BACKGROUND

2.1. Existing System

Presently, popular Real Estate websites like OLX, helloaddress.com, and zillow.com are utilized for property transactions. Currently, real estate operations rely heavily on manual record-keeping and limited digitalization. Property details and user interactions are managed through traditional means, leading to inefficiencies, potential errors, and delayed responses. The absence of a centralized system for property verification and user communication poses challenges in ensuring accuracy and trust.

The proposed Real Estate Website seeks to enhance user experience with a user-friendly and responsive platform. It builds on existing features and introduces innovations - including video uploading, improved privacy controls, and a 360-degree virtual tour. These enhancements aim to elevate the overall user experience and provide a comprehensive solution to the existing shortcomings in real estate websites.

2.2. Definition of Problem

1. Storing the data manually is difficult and time-consuming.
2. Manually stored data may contain errors or invalid data.
3. May affect the business productivity.

2.3. Proposed System

Esquare.com is a Responsive Real Estate Portal poised to revolutionize property transactions. It addresses the shortcomings of existing systems by delivering an exceptionally user-friendly and responsive web application. Users can effortlessly navigate and search for properties based on their preferences and budget with a single click, ensuring an enhanced and streamlined experience.

Building upon existing features, Esquare.com introduces cutting-edge innovations, including video uploading, improved privacy controls, and a 360-degree virtual tour, elevating the platform's functionality and providing users with a comprehensive real estate solution. Admins have full control for managing users, properties, and messages, ensuring authenticity through property approval, and gaining insights with comprehensive reports. Whereas Users have a seamless experience, allowing them to easily register, post properties, manage inquiries, and communicate while enjoying innovative features like video uploads and enhanced privacy controls.

Advantages of Proposed System:

- **User-Friendly Interface:** Esquare.com offers an intuitive and user-friendly interface, ensuring easy access and navigation for both admins and users.
- **Enhanced Efficiency and Flexibility:** The platform is designed to be highly efficient and flexible, allowing admins and users to log in and interact seamlessly, enhancing overall user experience.
- **24/7 Accessibility:** Esquare.com provides round-the-clock accessibility, allowing users and admins to engage with the platform anytime, anywhere globally.
- **Detailed Property Profiles:** Provides users with detailed property information, including pricing, features, and other essential details, facilitating informed decision-making during the property search.
- **Accurate Property Transactions:** Esquare.com employs automated processes, ensuring precision in property transactions, including posting, approval, and financial details, enhancing the accuracy and reliability of the real estate platform.
- **Security Measures:** Esquare.com prioritizes security and confidentiality, with unique usernames and passwords ensuring that only authorized admins and registered users can access the platform.
- **Interactive and Time-Saving:** The platform is highly interactive, saving time for both admins and users by streamlining processes and reducing paperwork.
- **Reduced Environmental Impact:** By minimizing paper usage, Esquare.com contributes to a more sustainable and eco-friendly approach to real estate transactions.

3. PROJECT OVERVIEW

3.1.Objective of the project

Esquare.com was created to modernize the real estate industry by moving from manual processes to a computer-based system. The goal was to simplify and automate the complicated steps involved in real estate transactions. By using computers, all aspects of property management, from registering users to listing properties and approving transactions, were handled smoothly. A major focus was on giving users a simple interface for easy property searches, while also introducing new features like video uploads and stronger privacy settings. The main goal was to make things more efficient, save money, and manage real estate records better. Administrators could easily manage user information, property approvals, and user interactions, while users could access detailed property information and submit feedback to help the system get better.

3.2.Stakeholders

- Admin

The Admin is in charge of running the entire real estate platform. They log in using a username and password, manage users, oversee property postings, approve transactions, and view user feedback. Admins play a key role in ensuring the smooth operation of Esquare.com.

- Users (Property Seekers and Sellers)

Users, including property seekers and sellers, log in using their username and password. They can search for properties, post their own properties, manage inquiries, and utilize innovative features like video uploads and enhanced privacy controls. Users are required to pay for a subscription if they have posted a property in the same month. Users are at the heart of Esquare.com, driving the dynamic real estate interactions on the platform.

3.3.Scope of project

The scope of Esquare.com is defined after the initial investigation, emphasizing seamless property transactions. The project primarily revolves around assisting users in buying and selling properties, with the admin overseeing data updates.

3.4. Feasibility Analysis

3.4.1 Feasibility study

Every project is feasible for given unlimited resources and infinitive time. Feasibility study is an evaluation of the proposed system regarding its workability, impact on the organization, ability to meet the user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development. Here the resourcesavailability and requirements are said to feasible to create the proposed system.

3.4.2. Technical Feasibility

Technical feasibility assesses whether the current technical resources are sufficient for the new system. If they are not available, can they be upgraded to provide the level of technology necessaryfor the new system? It checks whether the proposed system can be implemented in the present system without supporting the existing hardware. Currently,

- Technology exists to develop a system.
- The proposed system can hold data to be used.
- The proposed system can provide adequate response. Hence, we can say that the proposed system is technically feasible.

3.4.3. Operational Feasibility

Operational feasibility determines if the human resources are available to operate the system once it has been installed. The resources that are required to implement or install are already available with the Breakdown Assist. The persons of this Assist need no exposure to computer but have to be trained to use this particular software. The project is optimally feasible.

3.4.4. Schedule Feasibility

An evaluation of the time needed for the development of this project. The time schedule requiredfor the development of this project is very important, since more development time effects machinetime, costs and delays in the development of the other systems. So the project should be completewithin affixed schedule time as far as this is concerned.

Schedule feasibility study for the design is shown below

Problem identification	5
Requirement analysis	10
Overall design	20
Construction	22
Testing	15

3.4.5. Economic Feasibility

Economic feasibility determines whether the time and money are available to develop the system. It also includes the purchase of new equipment, hardware and software. Since software product must be cost effective in the development, on maintenance and in the use. It is affordable to allocatethe required resources.

4. OVERALL PROJECT PLANNING

4.1. Development Environment

Hardware Specifications

- Intel i3 or above
- Memory: at least 4GB
- Display: Color monitor
- Keyboard: Windows Compatible
- Mouse: Windows Compatible

Software Specifications

Technology used:

i. Server side

● Front end	: PHP, HTML, CSS, Bootstrap
● IDE	: Visual Studio Code
● Back end	: SQL server
● Operating System	: Windows

4.2. Constraints

The set of constraints that we come across this system is as follows

- User Interface is only in English i.e.no other language option is available.
- Admin can login with his assigned username and password i.e. no guest facility is available.

4.3. Deliverables

List of documents that shall be delivered are User Manual

- System maintenance documentation.
- Application archive with source code.
- Database backup and DDL script.
- Complete source code.

4.4. Assumptions and Dependencies

a) Assumptions

- All roles are created in the system already but further registration of users on given roles can be done.
- Roles and tasks are predefined and are made known to the administrator.
- The code should be free of compilation errors/syntax errors.
- The product must have an interface which is simple enough to understand.
- Roles and tasks are predefined and are made known to the administrator.
- End users should have basic knowledge of computer.

b) Dependencies

- All necessary hardware and software are available for implementing and use of the tool.
- All roles are created in the system already.
- The proposed system should be designed, developed and implemented based on the software requirements specifications document.

4.5. Risks

Some of the risks are follows

- Database crash will cause heavy data loss
- Wrong input will cause discrepancies in data
- Availability of the network.

4.6. Process Model

The process model for developing the project is agile model.

The phases are: -

- Requirement analysis
- System study
- Designing
- Coding
- Testing
- Maintenances

4.7. Test Strategy

4.7.1. System Testing

When a system is developed, it is hoped that it performs properly. In practice however some errors always occur. The main purpose of testing and information system is to find the errors and correct them. A successful test is one which finds an error. The main objectives of system testing are:

- To ensure during operation the system will perform as per specifications.
- To make sure that the system meets the requirements during operation.

- To verify that the controls incorporated in the system function as intended.
- To see that when correct inputs are fed to the system the outputs are correct.
- To make sure that during operation incorrect input and output will be deleted.

The scope of a system test should include both manual operations and computerized. Operation system testing is a comprehensive evaluation of the programs, manual procedures, computer operations and controls. System testing is the process of checking if the developed system is working according to the original objectives and requirements. All testing needs to be conducted in accordance with the test conditions specified earlier.

4.7.2 Types of testing

Unit Testing

Unit Testing will be done to test field validations, navigation, functionality of the programs and its block. These tests are applied on various functions within each program and other critical program blocks.

Module Testing

Module Testing will be each program done to test the interaction between the various programs within one module. It checks the functionality of with relation to other programs within the same module. It then tests the overall functionality of each module.

Integration Testing

The major concerns of integration testing are developing an incremental strategy that will limit the complexity of entire actions among components as they are added to the system. Developing a component as they are added to the system, developing an implementation and integration schedules that will make the modules available when needed, and designing test cases that will demonstrate the viability of the evolving system. Though each program works individually they should work after linking them together. This is also referred to as interfacing. Data may be lost across interface and one module can have adverse effect on another. Subroutines after linking may not do the desired function expected by the main routine. Integration testing is a systematic technique for constructing program structure while at the same time conducting tests to uncover errors associated with the interface. In the testing, the programs are constructed and tested in small segments.

Validation Testing

This provides the final assurance that the software meets all the functional, behavioral and performance requirements. The software is completely assembled as a package. Validation succeeds when the software functions in a manner in which user wishes. Validation refers to the process of using software in live environment in order to find errors. During the course of validation, the system failure may occur and sometime the coding has to be hanged according to the requirement. Thus the feedback from the validation phase generally produces changes in the software. Once the application was made of all logical and interface errors, inputting dummy data ensure that the software developed satisfied all the requirements of the user. The dummy data is known as test cases.

Output Testing

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in the specific format. Asking the users about the format of output they required, tests the output generated in two ways. One is on screen and another is printed format. The output format on the screen found to be correct as the format was designed in the system design phase according to the user needs. For the hard copy also, the output comes out as the specified requirement by the user. Hence output testing does not result in any correction in the system.

Acceptance Testing

Acceptance testing (also known as user acceptance testing) is a type of testing carried out in order to verify if the product is developed as per the standards and specified criteria and meets all the requirements specified by customer. This type of testing is generally carried out by a user/customer where the product is developed externally by another party. Acceptance testing falls under black box testing methodology where the user is not very much interested in internal working/coding of the system, but evaluates the overall functioning of the system and compares it with the requirements specified by them. User acceptance testing is considered to be one of the most important testing by user before the system is finally delivered or handled over to the end user. Acceptance testing is also known as validation testing, final testing, QA testing, factory acceptance testing and application testing etc. And in software engineering, acceptance testing may be carried out at two different levels; one at the system provider level and another at the end user level (hence called user acceptance testing, field acceptance testing or end-user testing). Acceptance test refers to the acceptance of data into the system for processing. The acceptance test contributes to the consistency and smooth working of the system. The system under consideration is tested for users at a time for developing and making changes whenever required.

4.8. Testing environment and tools

The hardware specification used for testing:

Operating system	Windows 11
Memory	8 GB
Hard Disk	512 GB

The software specification used for testing:

Front End	PHP, HTML, CSS, Bootstrap
Back End	XAMPP
Operating System	Windows 11

5. ITERATION PLANNING

5.1. Schedule

SERIAL NO.	TASK	DURATION
1	Problem identification	5 days
2	Requirement Specification	10 days
3	Database Design and Analysis	12 days
4	Design Analysis	9 days
5	Coding	24 days
6	Testing	10 days
	Total	70 days

5.2. Risk

- Wrong input
- Software installation issues.
- Database crash will cause heavy data loss

6. HIGH LEVEL SYSTEM ANALYSIS

6.1. User Characteristics

All users of the system are expected to have basic knowledge of using a computer and basic knowledge in English language.

Users of the system:

- Admin
- Users

6.2. Summary of system features/Functional requirements

Manage Users

Admin can view, and delete user accounts.

Manage Properties

Admin can oversee and control properties posted by users.

Property Approval

Admin has the authority to approve properties for authenticity.

Admin Management

Admin can add, update, view, and delete other admin accounts.

Message Handling

Admin can view and manage messages submitted by users.

Report Generation

Admin can generate various reports based on specified criteria.

Account Registration

Users can register and create their accounts.

Subscription Payment

Users can pay for subscriptions to post properties.

Property Management

Users can manage the properties they have posted.

Send Messages

Users can send messages, suggestions, or feedback to the admin.

Enquiries

Users can send inquiries to property owners and view received inquiries.

Profile Management

Users can view, edit, and manage their profiles. The users can also change password and phone number visibility settings.

6.3. Non Functional Requirements/Supplementary Specifications

The non-functional requirements which define the system performance are:

Accuracy:

The level of accuracy in the proposed system will be high. All operations would be done correctly and it ensures that whatever information that comes from the center is accurate.

Reliability:

The reliability of the proposed system will be high. The reason for the increased reliability of the system is that system uses correct formulas to calculate the results.

Immediate Response:

The system is highly responsive because it uses well accurate formulas to calculate required results provided the user should enter the valid input data.

Easy to Operate:

The system should be easy to operate and should be such that it can be developed within a short period of time and fit in the limited budget of the user.

The other non-functional requirements are:

- Security
- Maintainability
- Extensibility
- Reusability
- Resource utilizations

6.4. Glossary

Admin	Administrator
Users	Property Seekers and Sellers

6.5. Business Rules

The information should be correct and valid.

6.6. Use Case

Manage & Approve Listings

Admins can view, delete, and approve posted listings to ensure their authenticity.

Post Properties

Users can submit property listings with detailed descriptions, images, videos, and 360-degree view links for virtual tours.

Manage Posted Properties

Users can view, update, and delete their listings, as well as check the approval status of their posted properties.

Manage Admins

Admins can add, delete and search for other administrators in the system. They also have options to update their account.

Manage Users

Admins can add, delete, view, and search for existing users in the system.

Send Property Enquiry

Users can submit enquiries to the property owner regarding their desired listings.

Manage Enquiry Requests

Users can review all requets received from other users interested in their posted properties.

Send Messages

Users can communicate queries and suggestions directly to the admin through the Contact Us page.

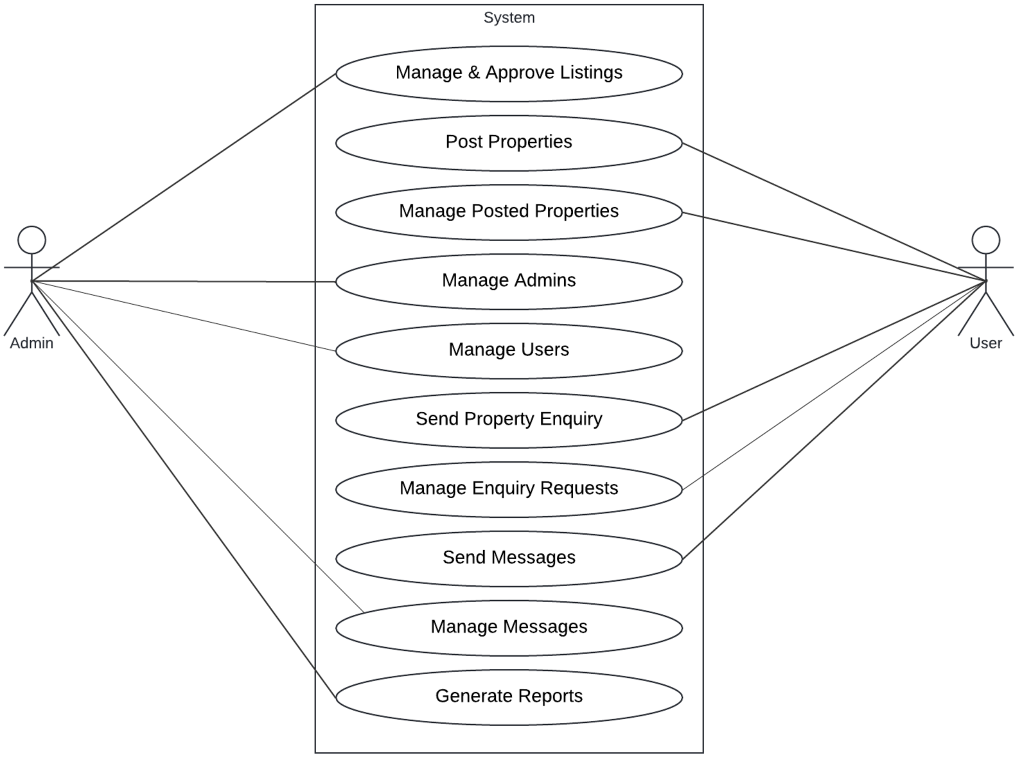
Manage Messages

Admins can view, search, and delete user-sent messages.

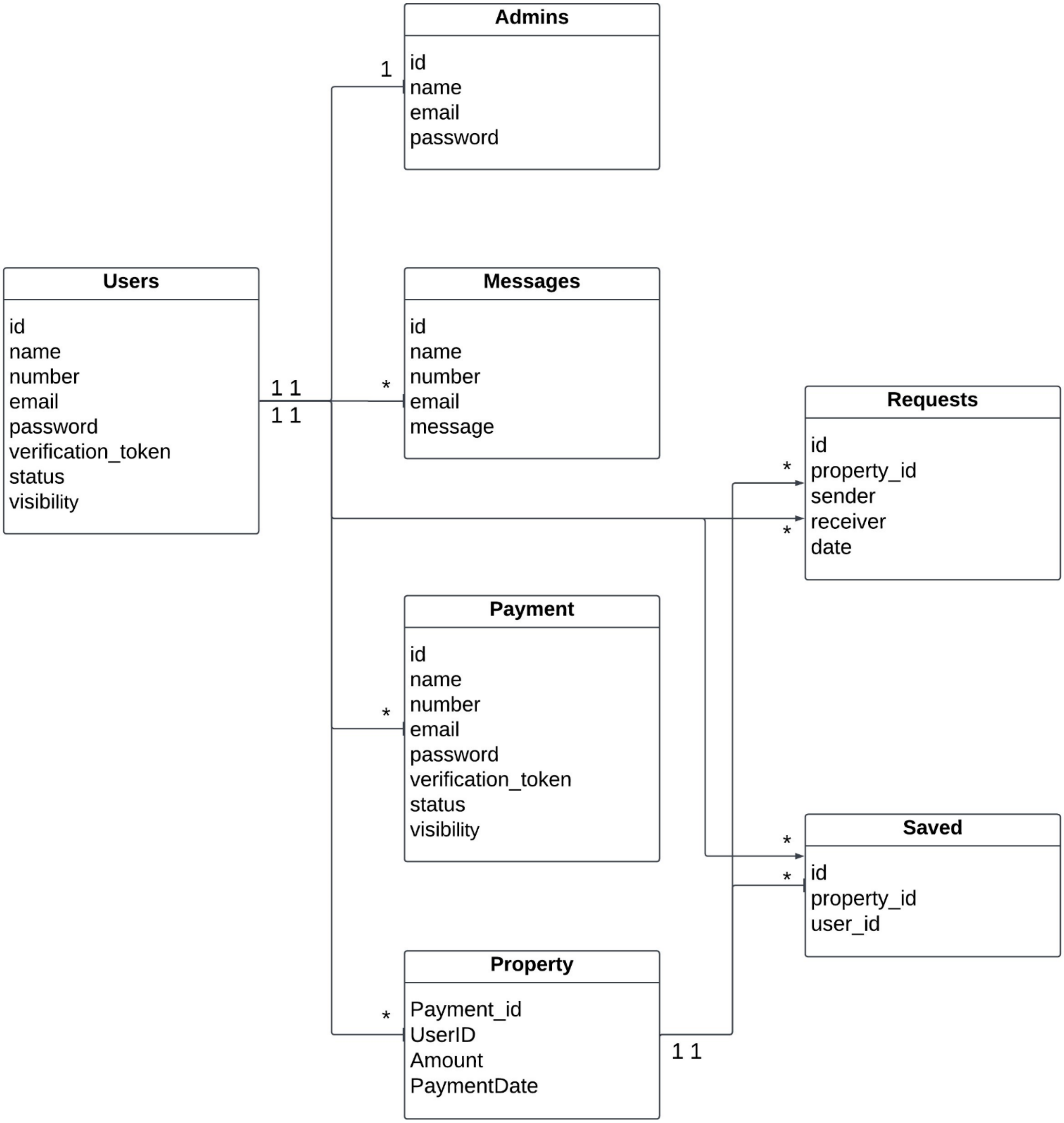
Generate Reports

Admins can generate various reports with visualizations based on different criteria.

6.7. Use-Case Diagram



7. DOMAIN MODEL



8. USE CASE MODEL

8.1 Use case text

Scope: Real Estate Management

Primary Actor: Admin

Stakeholders and Interests:

- **Users**: Users can be the one who posts properties or a person who wishes to acquire a property posted by another user. Users can log in to the system, search and post properties. They can manage their properties, and view enquiry requests for the properties they posted. They will also have to pay a subscription fee of 1000rs for posting an unlimited number of properties for 5 months if they have already posted a property in the month. They can also send an enquiry request for a property posted by another user. They can also send their messages to the admin.
- **Administrator:** Admin is responsible for operating the whole system. The admin gets logged in by a valid username and password. Admin can add and manage other Admins, Manage and Approve listings, View Messages from users as well as generate reports.

Preconditions:

No user can use the system without logging into the system.

Success Guarantee (Post conditions):

Really easy to use the system.

Main Success Scenario:

Use Case: Admin - Manage & Approve Listings

1. System redirects to admin's listing page.
2. The admin views all the listings and their details.
3. Admin clicks on 'View Property' button for viewing property.
   a. The system redirects to a new page where the admin can view the property along with the description, images, and videos.

    b. The admin can also delete the property from the same page
        i. System validates the property information and if there’s no issue, the data is deleted from the database.

4. Admin clicks on the ‘Delete Listing’ button.
    a. System prompts for confirmation and validates the property information and if there’s no issue, the data is deleted from the database.
5. Admin clicks on the ‘Approve Listing’ button.
    a. System prompts for confirmation and validates the property information and if there’s no issue, the data is updated in the database making the property status as approved.
6. Admin types in the search query and clicks on the search button.
    a. System validates the entered information and if there exists such a property, the data is retrieved from the database.

Use Case: Admin - Manage Admins

1. System redirects to admin’s page where they can manage administrators.
2. The admin views all the admins and their details.
3. Admin clicks on the ‘Update Profile’ button to update their profile.
    a. The system redirects to a new page where the admin can enter their new account details and update their account.
        i. The system validates the account information and if there’s no issue, the data is updated in the database.
4. Admin clicks on the ‘Register New’ button to add a new admin to access the system.
    a. The system redirects to a new page where the admin can enter the details of the new admin account.
        i. The system validates the account information and if there’s no issue, the data is added to the database.
5. Admin clicks on the ‘Delete Admin’ button.
    a. System prompts for confirmation and validates the admin information and if there’s no issue, the data is deleted from the database.

6. Admin types in the search query and clicks on the search button.
    a. System validates the entered information and if there exists such a admin, the data is retrieved from the database.

Use Case: Admin - Manage Users

1. System redirects to admin’s page where they can manage users.
2. The admin views all the users and their details.
3. Admin clicks on the ‘Delete User’ button.
    a. System prompts for confirmation and validates the information and if there’s no issue, the data is deleted from the database.
4. Admin types in the search query and clicks on the search button.
    a. System validates the entered information and if there exists such a user, the data is retrieved from the database.

Use Case: Admin – Manage Messages

1. System redirects to admin’s page where they can manage messages.
2. The admin views all the messages from the users and their details.
3. Admin clicks on the ‘Delete Message’ button.
    a. System prompts for confirmation and validates the information and if there’s no issue, the data is deleted from the database.
4. Admin types in the search query and clicks on the search button.
    a. System validates the entered information and if there exists such a message, the data is retrieved from the database.

Use Case: User- Post Property

1. User clicks the ‘Post Property’ button.
2. If the user has already posted a property in the same month, the system redirects to the payment page to pay the subscription fee.
    a. The system prompts the user to enter their payment details including card number and cvv.
        i. System validates the entered information and if there’s no issue, the payment details are entered into the database.

3. The system redirects to the post property page.
    a. System prompts the user for entering details of the property, upload the images and videos of the property.
    b. Users enters the property details and clicks ‘Post Property’ button.
        i. System validates the entered information and if there’s no issue, the details are inserted in the database and shows a success message.

Use Case: User- Manage Posted Property

1. System redirects to user’s ‘My Listings page’.
2. The user views all the listings posted by them and their details.
3. User clicks on the ‘Update Property’ button to update their property.
    a. The system redirects to a new page where the user can update their property and change their descriptions, images and videos.
        i. The system validates the property information and if there’s no issue, the data is updated in the database.
4. User clicks on the ‘Delete Property’ button.
    a. System prompts for confirmation and validates the property information and if there’s no issue, the data is deleted from the database.
5. User clicks on ‘View Enquiry’ button for viewing property requests.
    a. The system redirects to a new page where the user can view all the enquiry requests from users for that property.
    b. The user can click on ‘Delete Request’ button for deleting the request.
        i. System prompts for confirmation and validates the property information and if there’s no issue, the data is deleted from the database.
    c. The user can click on ‘View Property button’ to view the property for which the request was received.
        i. The system redirects to a new page where the user can view the property along with the description, images, and videos.
6. User clicks on ‘View Property’ button for viewing property.
    a. The system redirects to a new page where the user can view the property

along with the description, images, and videos.

b. The user can also view enquiries from the same page by clicking ‘View Enquiries’ button.

    i. Goto step 5

Use Case: User – Send Property Enquiry

1. The user clicks on the ‘All listings’ page and the system redirects to a new page where they can view all the approved properties and its details.
2. User clicks on the ‘Send Enquiry’ button.
    a. System prompts for confirmation and validates the information and if there’s no issue, the request is sent to the user who posted the property.

Use Case: User – Manage All Enquiry Requests

1. The user clicks on the ‘My Requests’ option and the system redirects to a new page.
2. The user views all the requests from other users and their details.
3. The user can click on ‘Delete Request’ button for deleting the request.
    a) The system prompts for confirmation and validates the property information and if there’s no issue, the data is deleted from the database.
4. The user can click on ‘View Property button’ to view the property for which the request was received.
    a) The system redirects to a new page where the user can view the property along with the description, images, and videos.

Use Case: User – Send Messages

1. The user clicks on the ‘Contact Us’ page and the system redirects to a new page where they can view a contact us form.
2. The user fills in all the fields with their personal details along with the message they wish to send to the admin.
3. User clicks on the ‘Send Message’ button.
    a. System prompts for confirmation and validates the information and if there’s no issue, the request is sent to the user who posted the property.

Use Case: Admin- Generate Reports

1. System redirects to admin's generate reports page.
2. The admin click on the button appropriate for generating the desired report.
3. Based on the selected button, the system redirects to the appropriate reports page.
4. Admin can generate various reports based on different criteria.

Extensions:

Admin - Manage Admins

3. a. The new password and password retyped for confirmation isn't same.

1. System tells the Admin that the passwords entered doesn't match.
2. The use case continues at step 3.

4. a. The new admin information entered doesn't contain all the necessary details.

1. System tells the Admin that some information is missing.
2. The use case continues at step 4.

4. b. The entered admin information is in incorrect format.

1. System tells the admin that some information is incorrect.
2. The use case continues at step 4.

User- Post Property

2. a. The entered payment information doesn't contain all the necessary details.

1. System tells the User that some information is missing.
2. The use case continues at step 2.

2. b. The entered payment information is incorrect.

1. System tells the User that some information is incorrect.
2. The use case continues at step 2.

3. a. The entered property information doesn't contain all the necessary details.

1. System tells the User that some information is missing.
2. The use case continues at step 3.

3. b. The entered property information is in incorrect format.

1. System tells the User that some information is incorrect.
2. The use case continues at step 3.

User- Manage Posted Properties

3. The entered property information is in incorrect format.

    1. System tells the user that some information is incorrect.
    2. The use case continues at step 3.

User- Send Messages

3. a. The entered message information doesn’t contain all the necessary details.

    1. System tells the User that some information is missing.
    2. The use case continues at step 2.

3. b. The entered message information is in incorrect format.

    1. System tells the User that some information is incorrect.
    2. The use case continues at step 2.

Special Requirements:

1. Text must be visible from 1 meter.
2. We want robust recovery when the system fails.
3. Language internationalization on the text displayed.

Frequency of Occurrence:

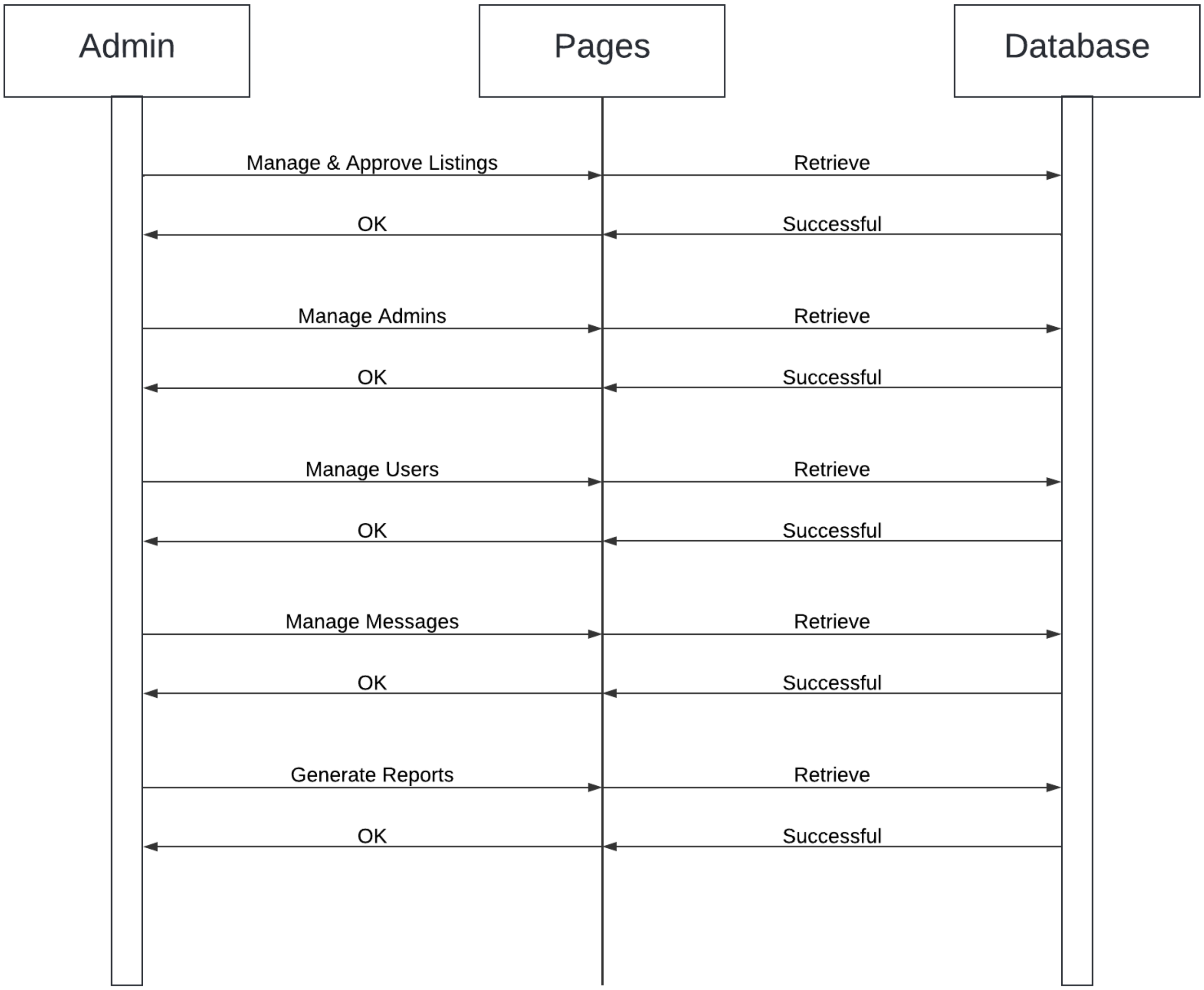
Could be nearly continuous.

Open Issues:

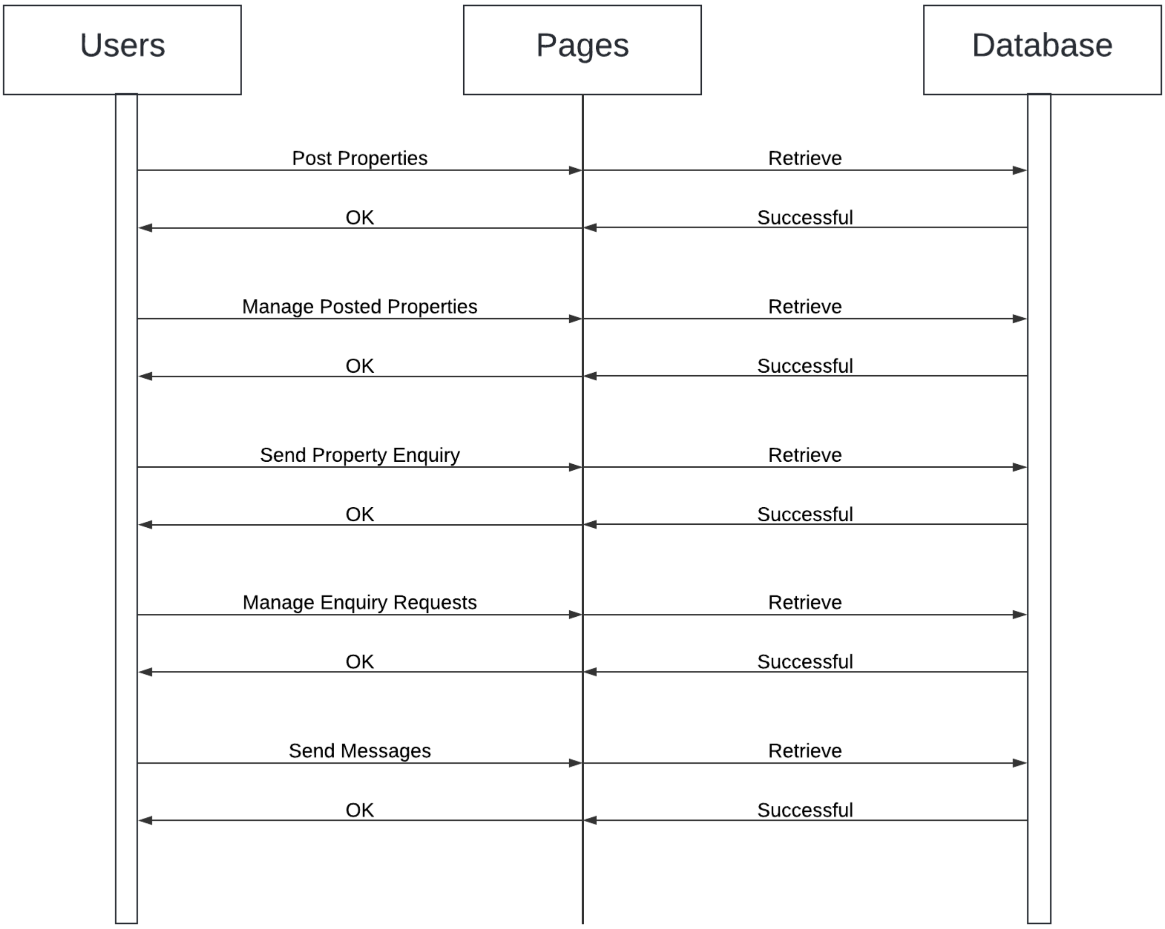
1. Explore the recovery issues.

8.2 System Sequence Diagram

Admin Side



User Side



8.3 Operation Contracts

Operation: Registered User (username: string, password: string)

Cross Reference: Use case: User login

Preconditions: Proper communication between pages

Conditions :

- A new login instance was created was created by the admin/user.
- Accepted username and password and stored it in respective attributes.
- After validation, the user gets logged in to the system.

Operation: Admin (username: string, password: string)

Cross Reference: Use case: Admin login

Preconditions: Proper communication between pages

Conditions :

- An admin login instance was defined.
- Accepted username and password and stored it in respective attributes.
- After validation, the admin gets logged in to the system.

8.4 Reports

Subscriber Payment Report: Admin can generate reports based on Subscription payment done by the users between the selected date range.

Property Owner Report: Admin can generate reports based on the property owner type – Owner, Builder, Dealer who posted it.

Property Price Report: Admin can generate reports based on properties whose pricing is between the entered budget range.

Property Type Report: Admin can generate reports based on property type – Flat, House or Office.

Property Posted Report: Admin can view data reports based on properties that are posted between the selected date range.

Monthly Payment Report: Admin can generate reports based on Subscription payment done by the users between the selected month and year.

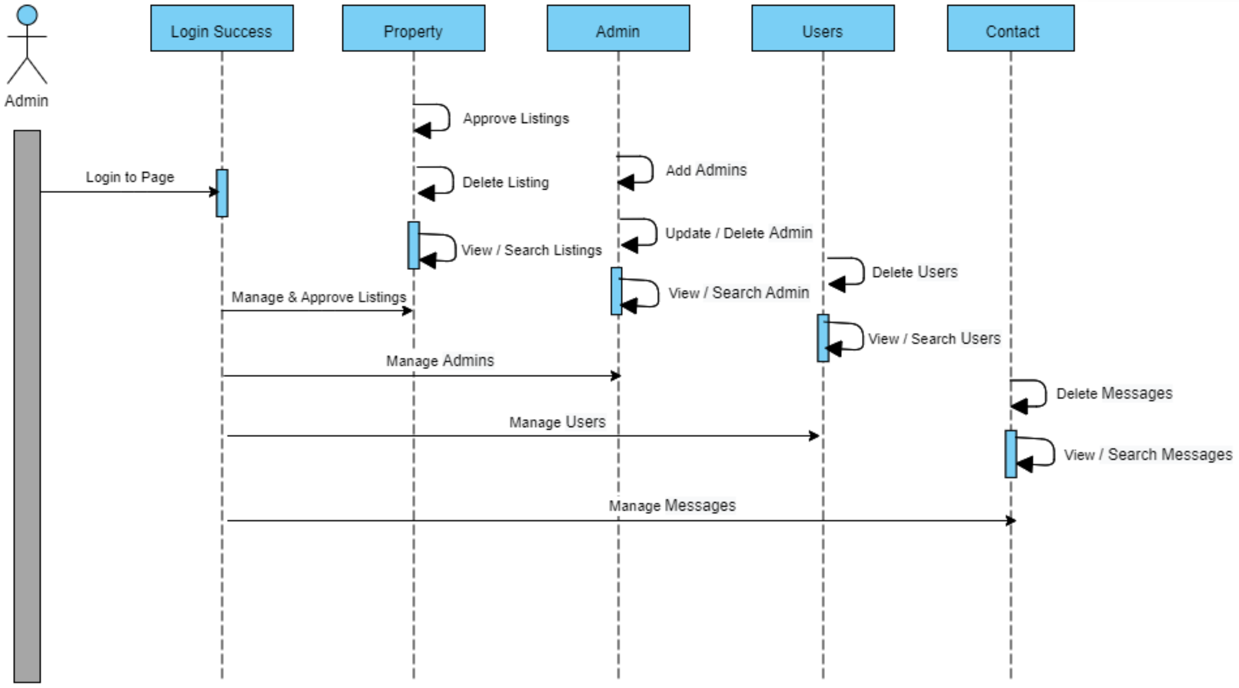
Yearly Payment Report: Admin can generate reports based on Subscription payment done by the users between the selected year.

Property Type Visualization Report: Admin can generate a bar chart report based on property type – Flat, House or Office.

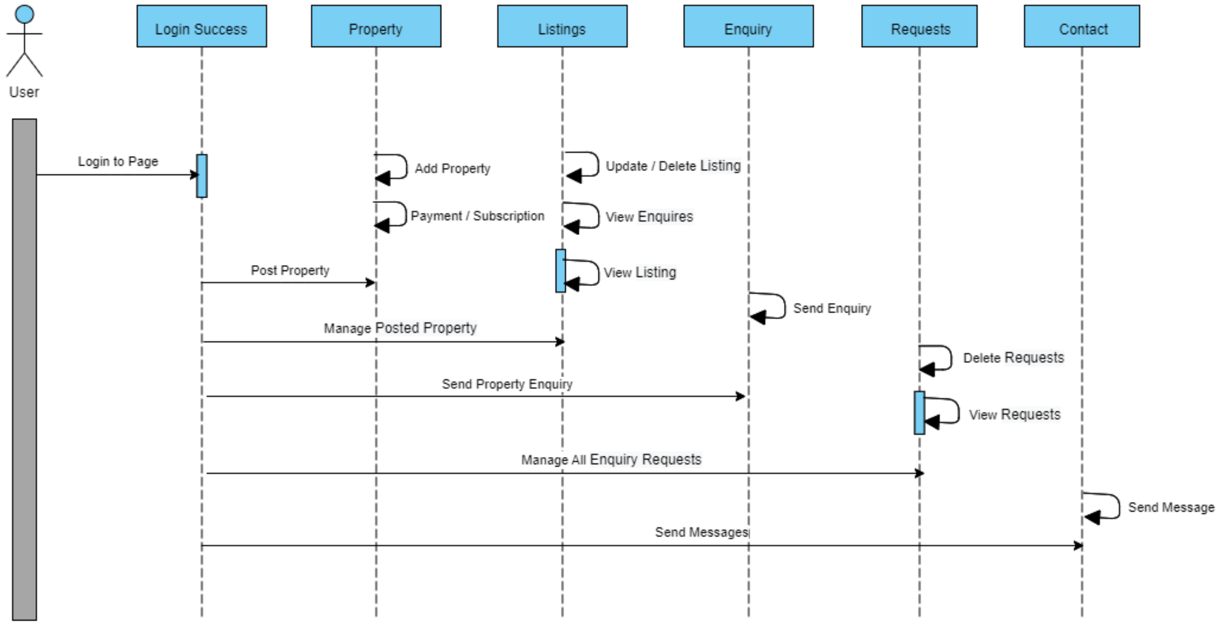
9. DESIGN MODEL

9.1 Sequence Diagram

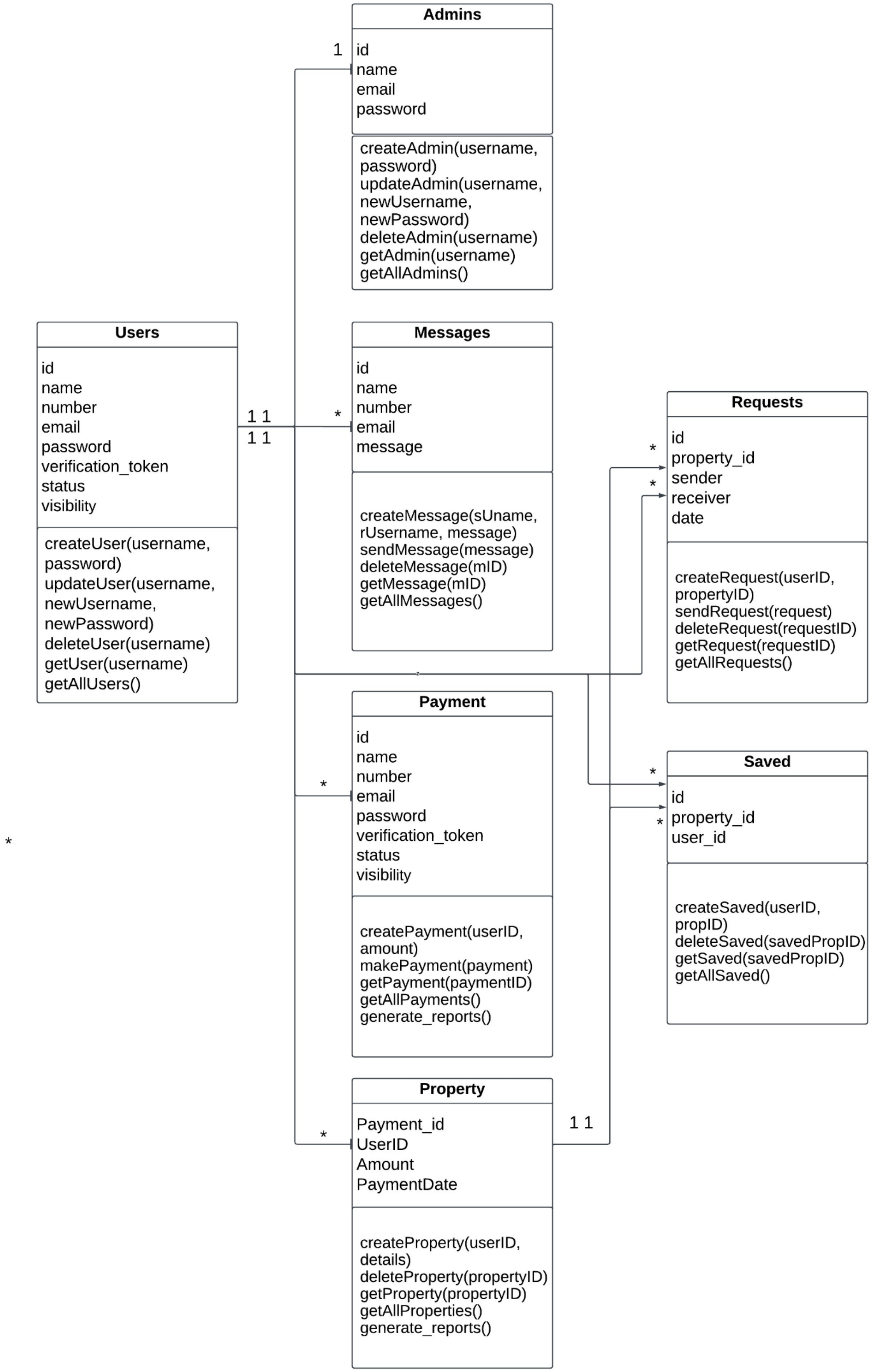
Admin



User



9.2 Class Diagram



9.3 UI Design

ADMIN:

Login Page: The admin must enter a username and a password to login to the application.

Home Page: Displays the total count of users, admins, properties posted, and new messages.

Listings Page: Admin can view all the properties posted by the users and will have the options to delete and approve them.

Approval Waiting Page: Admin can view all the unapproved properties and approve them.

Users Page: Admin can view/search or delete users.

Admins Page: Admin can add/view/search or delete admins. They also have the option to update their profile and change the password.

Messages Page: Admin can view/search or delete Messages sent by users to admin.

Reports Page: Admin can generate various reports based on Subscriber Payment, Property Owner, Property Pricing, Property Type, Monthly and Yearly Payment as well as a Visualization on Property Types.

Logout Page: By clicking this button, the admin gets logged out of the application.

USER:

Login Page: The user must enter a username and a password to login to the application.

Home Page: The user can view all the existing properties and also allows them to search for property based on different criterias.

Post Property Page: Enables the user to post a property and share property details, including descriptions, images, videos, and 360-degree views. The user is redirected to pay for a subscription before posting a property if the user has already posted a property in the same month.

My Listings Page: The user can view, update or delete all the properties they have posted.

My Requests Page: The user can view or delete all the enquiry requests for the properties they have posted.

All Listings Page: It allows the user to view the properties posted by other users. It also allows the user to send an enquiry to the user who posted it.

Dashboard Page: It gives the user an overview on the count of the Properties Listed and Saved, Requests Sent and Received by them.

Contact Us: It allows the user to send messages and their valuable suggestion to the administrator personally.

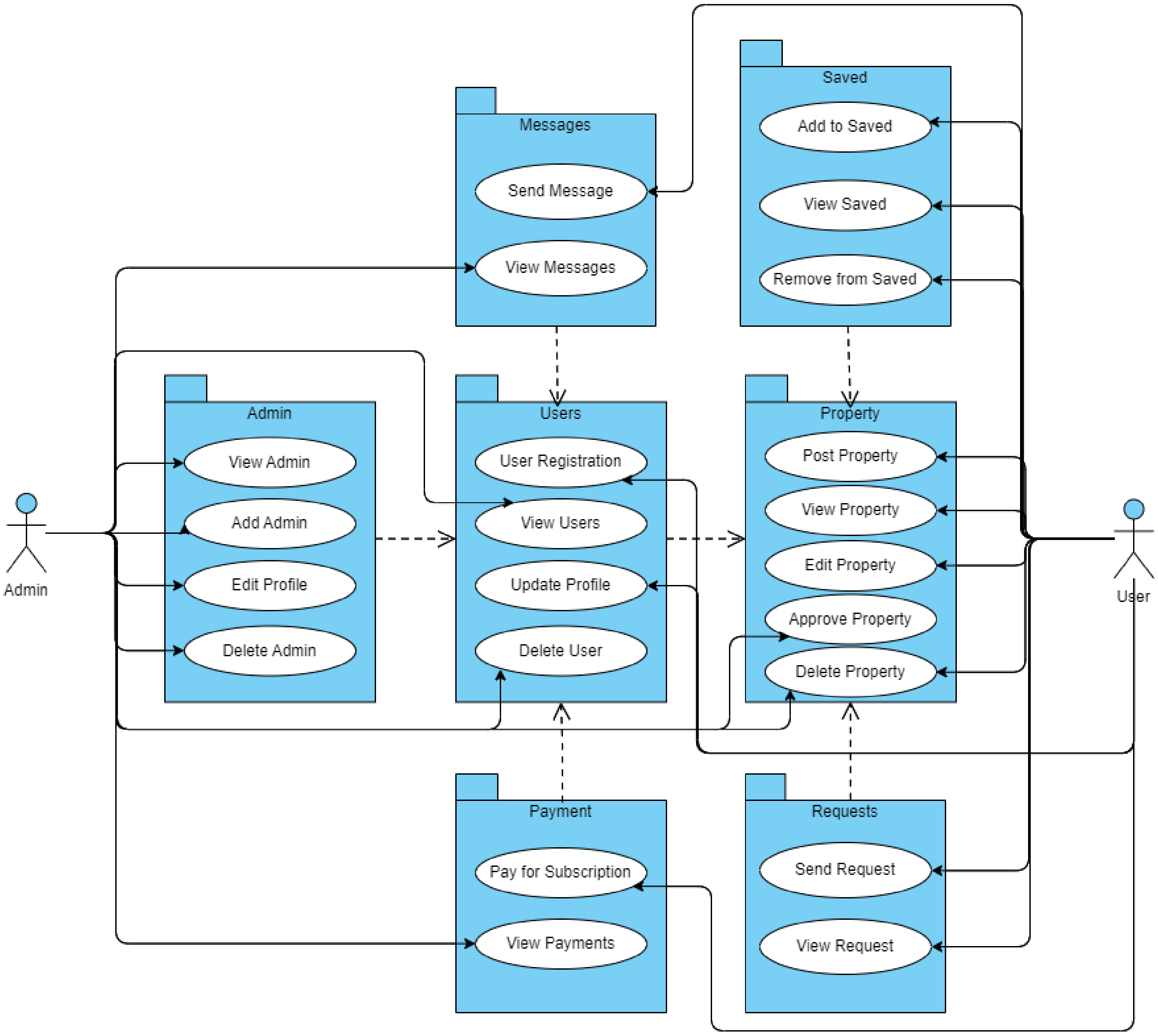
Logout Page: By clicking this button, the user gets logged out of the application.

9.4 Theoretical Background

Esquare – The Real Estate Portal is developed using one of the widely used front-end tools PHP, HTML, CSS and Bootstrap and at the back-end, we used SQL Server. The Visual Studio Code is used as the Integrated Development Environment(IDE). The Operating System used to develop this application is Windows 11.

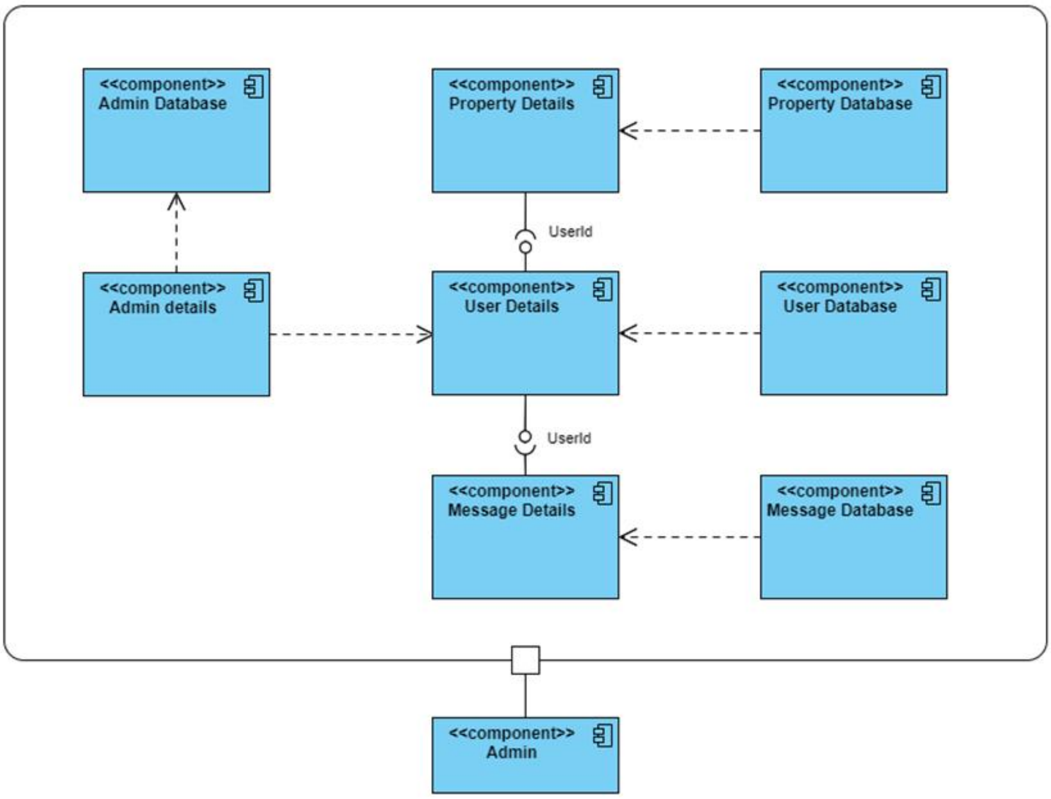
9.5 Architecture

9.5.1 Package diagram



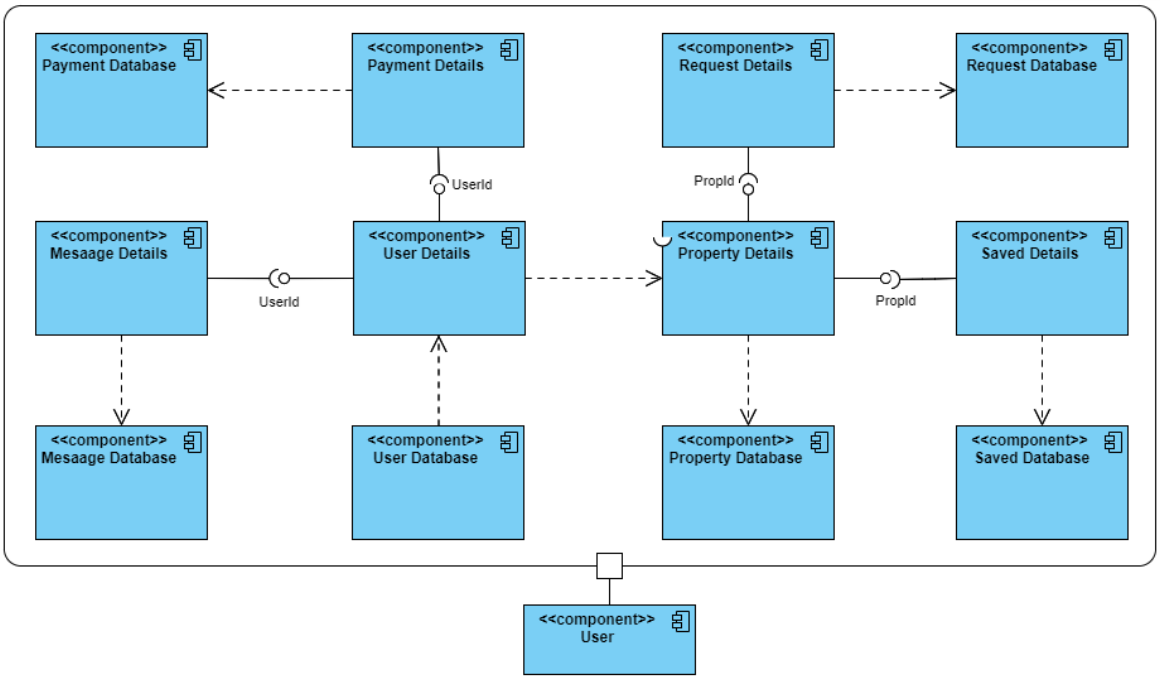
9.5.2 Component diagrams

Admin Side



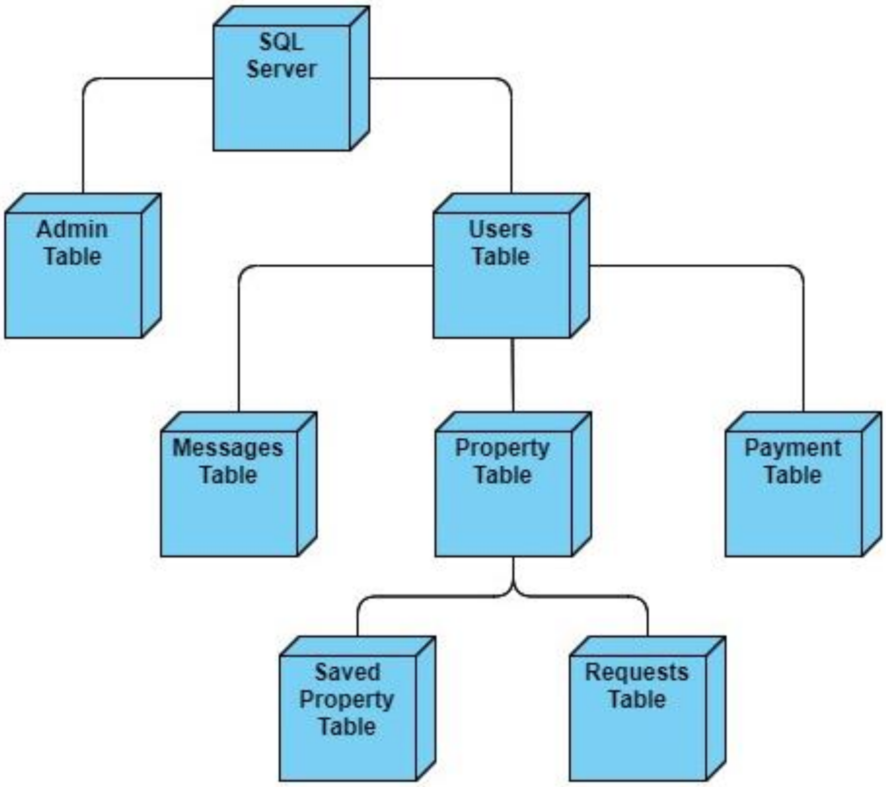
<<component>> Admin Database
<<component>> Property Details
<<component>> Property Database
UserId
<<component>> Admin details
<<component>> User Details
<<component>> User Database
UserId
<<component>> Message Details
<<component>> Message Database
<<component>> Admin

User Side



<<component>> Payment Database
<<component>> Payment Details
<<component>> Request Details
<<component>> Request Database
UserId
PropId
<<component>> Mesaage Details
UserId
<<component>> User Details
<<component>> Property Details
PropId
<<component>> Saved Details
<<component>> Mesaage Database
<<component>> User Database
<<component>> Property Database
<<component>> Saved Database
<<component>> User

9.5.3 Deployment diagram



9.6 Database Design

Data Base Name : home_db

USER INFO TABLE

S.No	Field name	Data Type	Description	Constraints
1	id	varchar(10)	ID of user	Primary Key, Not Null
2	name	varchar(50)	Name of user	Not Null
3	number	varchar(10)	Phone number of user	Not Null
4	email	varchar(50)	Email Id of user	Not Null
5	password	varchar(50)	Password of user	Not Null
6	verification_token	varchar(30)	Verification Id for user	Not Null
7	status	varchar(20)	Account status	Not Null
8	visibility	int(2)	Account visibility	Not Null

ADMIN INFO TABLE

S.No	Field name	Data Type	Description	Constraints
1	id	varchar(10)	ID of Admin	Primary Key, Not Null
2	name	varchar(20)	Name of Admin	Not Null
3	password	varchar(20)	Password of Admin	Not Null

MESSAGES TABLE

S.No	Field name	Data Type	Description	Constraints
1	id	varchar(20)	ID of message	Primary Key, Not Null
2	name	varchar(50)	Name of sender	Not Null
3	email	varchar(50)	Email ID of sender	Not Null
4	number	varchar(10)	Phone number of sender	Not Null
5	message	varchar(1000)	Message of sender	Not Null

PROPERTY TABLE

S.No	Field name	Data Type	Description	Constraints
1	id	varchar(20)	ID of property	Primary Key, Not Null
2	user_id	varchar(20)	ID of user	Foreign Key, Not Null
3	property_name	varchar(50)	Name of property	Not Null
4	address	varchar(100)	Address of property	Not Null
5	price	varchar(10)	Price of property	Not Null
6	type	varchar(10)	Type of property	Not Null
7	offer	varchar(10)	Offer of property	Not Null
8	status	varchar(50)	Status of property	Not Null
9	furnished	varchar(50)	If property is furnished	Not Null
10	bhk	varchar(10)	BHK of property	Not Null

11	deposite	varchar(10)	Deposit Amount for property	Not Null
12	bedroom	varchar(10)	Bedroom Count of property	Not Null
13	bathroom	varchar(10)	Bathroom Count of property	Not Null
14	balcony	varchar(10)	Balcony Count of property	Not Null
15	carpet	varchar(10)	Carpet Area of property	Not Null
16	age	varchar(2)	Age of property	Not Null
17	total_floors	varchar(2)	Total Floor Count of property	Not Null
18	room_floor	varchar(2)	Room Count of property	Not Null
19	loan	varchar(20)	Loan Availability of property	Not Null
20	lift	varchar(3)	Lift Availability of property	Not Null
21	security_guard	varchar(3)	Security Guard Availability for property	Not Null
22	play_ground	varchar(3)	Playground Availability of property	Not Null
23	garden	varchar(3)	Garden Availability of property	Not Null
24	water_supply	varchar(3)	Water Availability of property	Not Null
25	power_backup	varchar(3)	Power Backup Facility of property	Not Null
26	parking_area	varchar(3)	Parking Availability of property	Not Null
27	gym	varchar(3)	Gym Availability of property	Not Null
28	shopping_mall	varchar(3)	Mall Availability near property	Not Null
29	hospital	varchar(3)	Hospital Availability near property	Not Null
30	school	varchar(3)	School Availability near property	Not Null
31	market_area	varchar(3)	Market Availability near property	Not Null
32	image_01	varchar(50)	Image 1 of property	Not Null
33	image_02	varchar(50)	Image 2 of property	Not Null
34	image_03	varchar(50)	Image 3 of property	Not Null
35	image_04	varchar(50)	Image 4 of property	Not Null
36	image_05	varchar(50)	Image 5 of property	Not Null

37	video_name	varchar(255)	Video of property	Not Null
38	view	varchar(200)	360-degree view link of property	Not Null
39	description	varchar(1000)	Property description	Not Null
40	date	date	Date posted of property	Not Null
41	approved	int(2)	Approval Status of property	Not Null

PROPERTY REQUESTS TABLE

S.No	Field name	Data Type	Description	Constraints
1	id	varchar(20)	ID of Request	Primary Key, Not Null
2	property_id	varchar(20)	ID of Property	Foreign Key, Not Null
3	sender	varchar(20)	ID of Sender	Foreign Key, Not Null
4	receiver	varchar(20)	ID of Reciever	Not Null
5	date	date	Date of Request	Not Null

SAVED PROPERTY TABLE

S.No	Field name	Data Type	Description	Constraints
1	id	varchar(20)	ID of Saved Property	Primary Key, Not Null
2	property_id	varchar(20)	ID of Property Saved	Foreign Key, Not Null
3	user_id	varchar(20)	User ID who Saved the Property	Foreign Key, Not Null

USER SUBSCRIPTION TABLE

S.No	Field name	Data Type	Description	Constraints
1	Payment_id	int(3)	ID of Payment	Primary Key, Not Null
2	UserID	varchar(20)	User ID of Payment	Foreign Key, Not Null
3	Amount	int(10)	Amount of Payment	Not Null
4	PaymentDate	varchar(25)	Date of Payment	Not Null

10. TESTING

10.1 Test cases

Test Scenario: Checking Login Functionality

<u>Test Case 1: Invalid User Login</u>

An unregistered or unauthorized login attempt must be blocked.

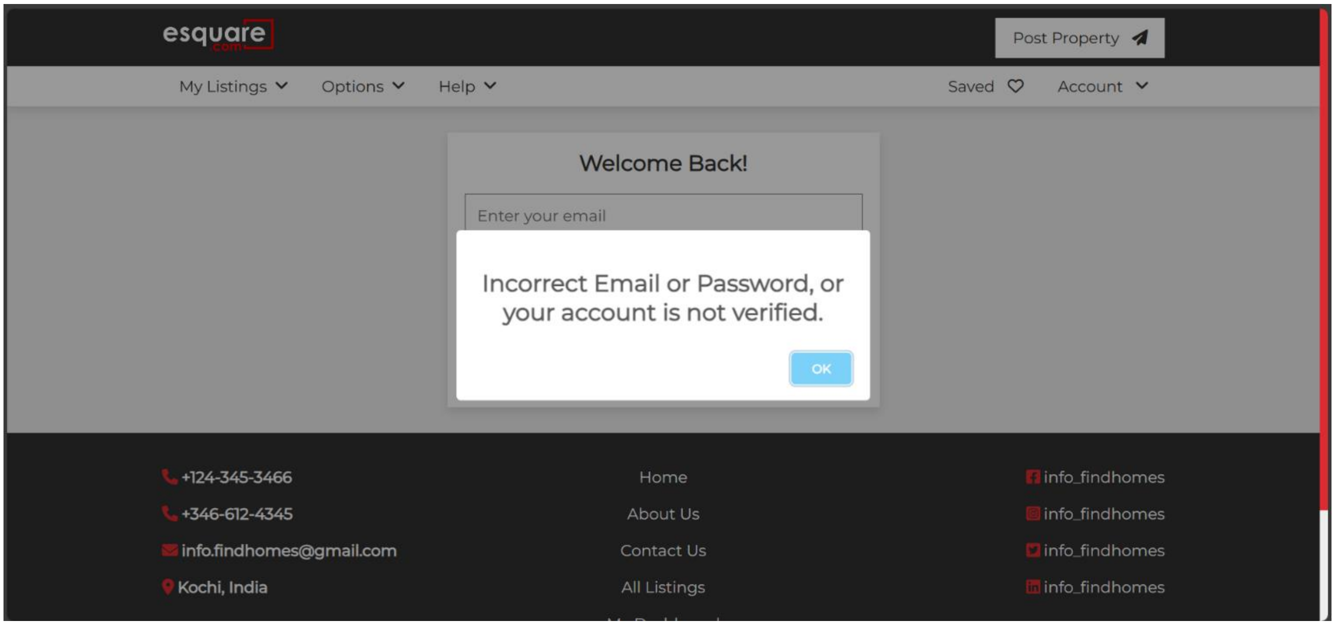
Precondition: Unauthorized users do not have valid credentials to log in.

Assumption: Only authorized users have access to valid credentials for logging in.

Test Steps:

1. Go to Login Page
2. Enter credentials
3. Submit the credentials

Expected Result: A login attempt with invalid or wrong input results in an unsuccessful login message.



Incorrect Email or Password, or your account is not verified.

Test Case 2: Invalid Admin Login

An unregistered or unauthorized login attempt must be blocked.

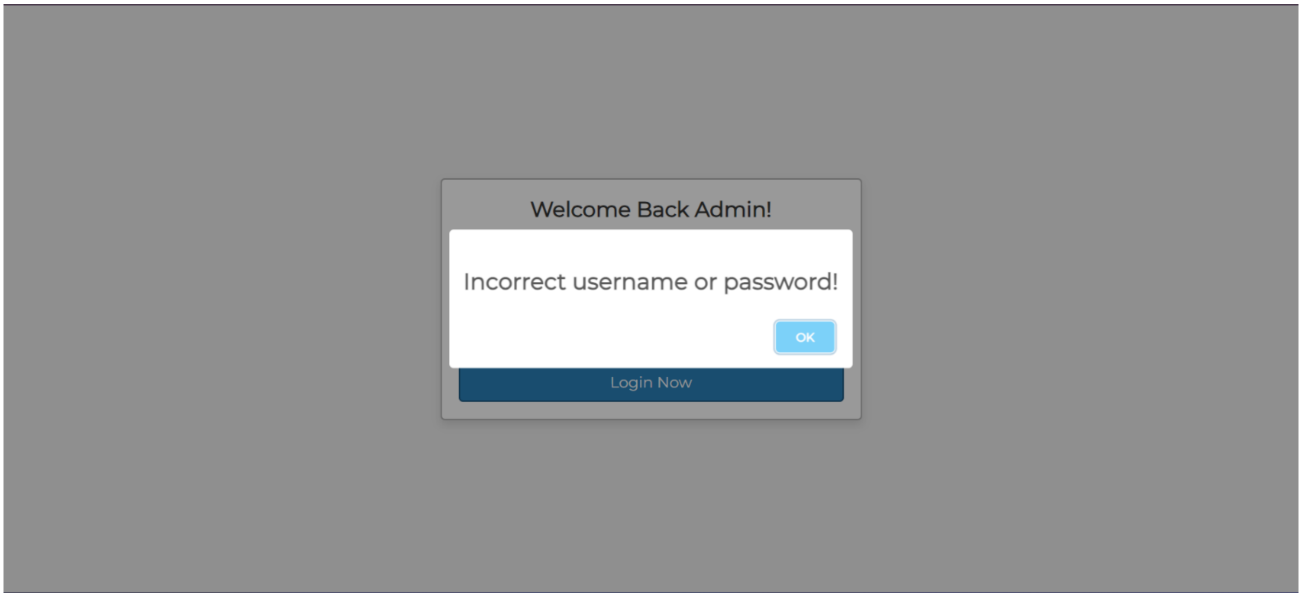
Precondition: Unauthorized admin do not have valid credentials to log in.

Assumption: Only authorized admin have access to valid credentials for logging in.

Test Steps:

1. Go to Login Page
2. Enter credentials
3. Submit the credentials

Expected Result: A login attempt by admin with invalid or wrong input results in an unsuccessful login message.



Welcome Back Admin!

Incorrect username or password!

Test Case 3: Check results on not entering new matching passwords by User at Update Profile page

All the attempts to update profile must be blocked if the new passwords entered by the user don’t match.

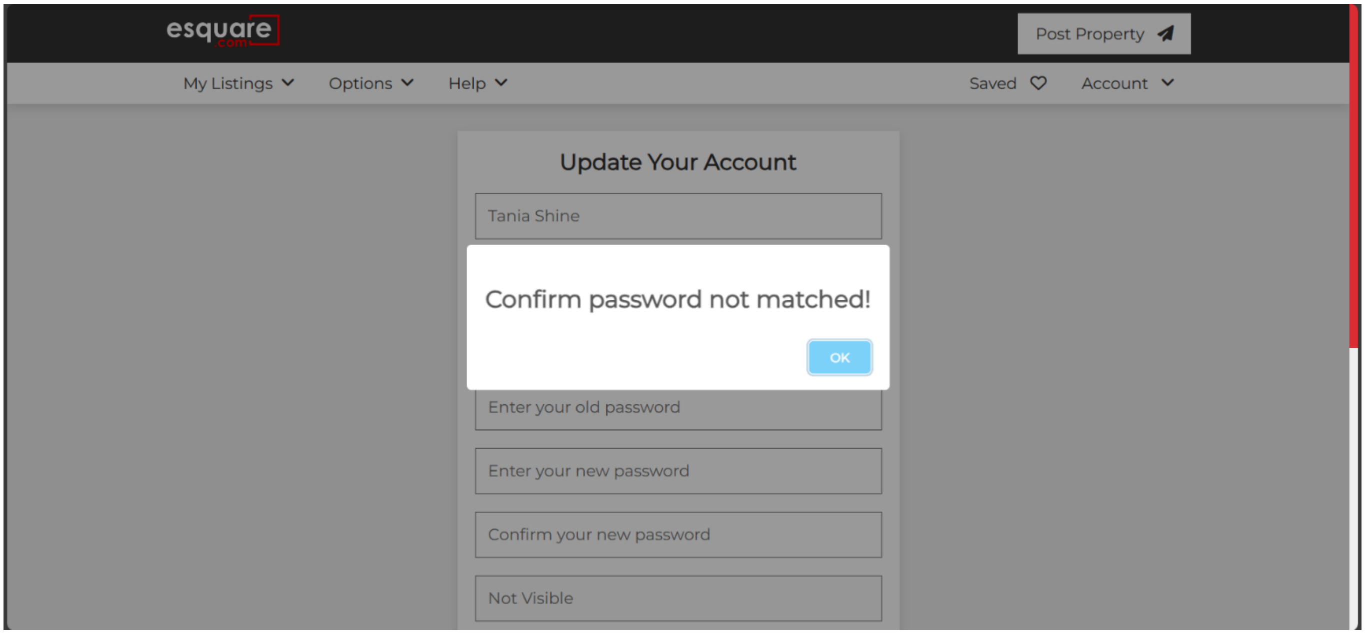
Precondition: User accidentally forgets the new password entered.

Assumption: Only authorized users have access to update their profile.

Test Steps:

1. Log in to the system as User using the correct user’s credentials.
2. Go to the Update Profile page.
3. Submit the updated details of the profile by entering mismatched new passwords.

Expected Result: An attempt by the user to update their profile by entering mismatched new passwords, results in an unsuccessful error message.



Update Your Account

Tania Shine

Confirm password not matched!

Test Case 4: Check results on not correctly entering old password by User at Update Profile page

All the attempts to update profile must be blocked if the old password entered by the user is wrong.

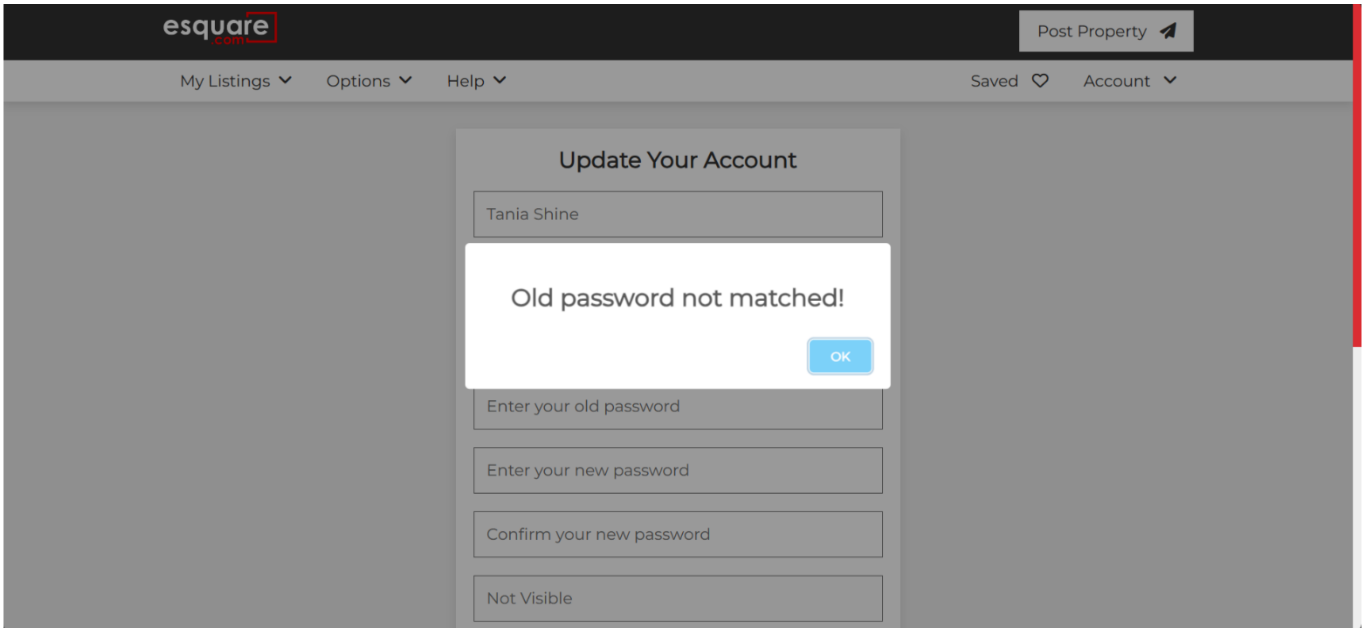
Precondition: User accidentally mistypes or forgets the old password.

Assumption: Only authorized users have access to update their profile.

Test Steps:

1. Log in to the system as User using the correct user's credentials.
2. Go to the Update Profile page.
3. Submit the updated details of the profile by entering wrong old password.

Expected Result: An attempt by the user to update their profile by entering wrong old password, results in an unsuccessful error message.



Tania Shine

Old password not matched!

Test Case 5: Check results on not storing all the required fields by User at Contact Us page

All the attempts to send messages must be blocked if all the required fields aren't filled by the user.

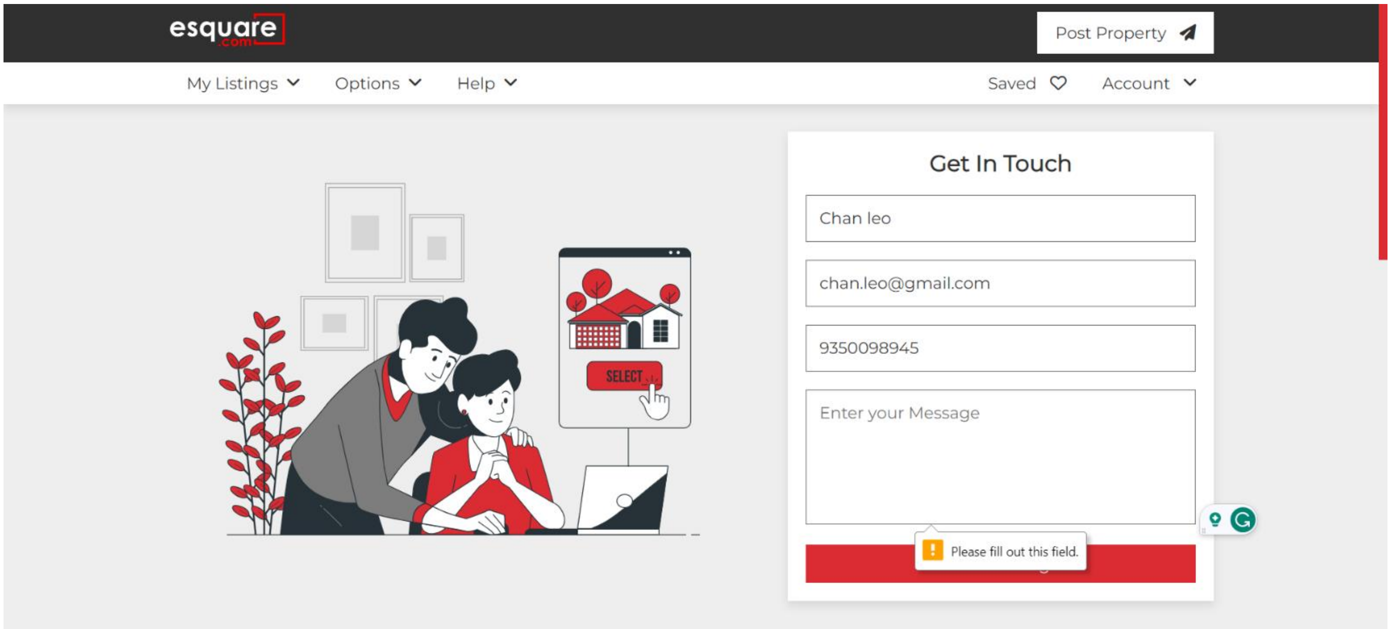
Precondition: User accidentally forgets or does not have all the necessary data to send a message.

Assumption: Only authorized users have access to send messages.

Test Steps:

1. Log in to the system as User using the correct user's credentials.
2. Go to the Contact Us page.
3. Submit the details of the message without filling all or many of the required fields.

Expected Result: An attempt by the user to send a message without filling all or many of the required fields, results in an unsuccessful error message.



Get In Touch

Chan leo

chan.leo@gmail.com

9350098945

Enter your Message

Please fill out this field.

Test Case 6: Check results on validating details entered by User at Contact Us page

All the attempts to send messages must be blocked if any of the fields are filled in an incorrect format.

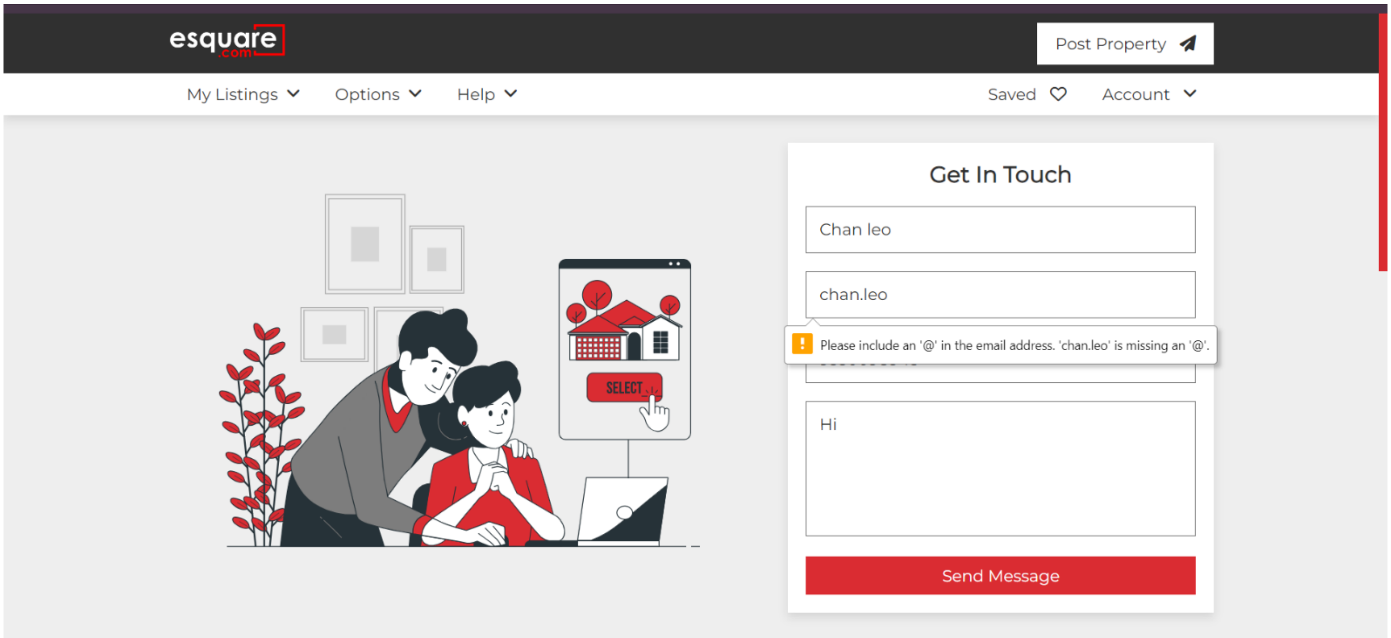
Precondition: User accidentally forgets or does not have all the necessary data to send a message.

Assumption: Only authorized users have access to send messages.

Test Steps:

1. Log in to the system as User using the correct user’s credentials.
2. Go to the Contact Us page.
3. Submit the details of the message by passing data in incorrect format.

Expected Result: An attempt by the user to send a message by filling all or many of the fields in an incorrect format, results in an unsuccessful error message.



Get In Touch

Chan leo

chan.leo

Please include an '@' in the email address. 'chan.leo' is missing an '@'.

Hi

Test Case 7: Check results on not entering new matching passwords by Admin at Update Profile page

All the attempts to update profile must be blocked if the new passwords entered by the Admin don’t match.

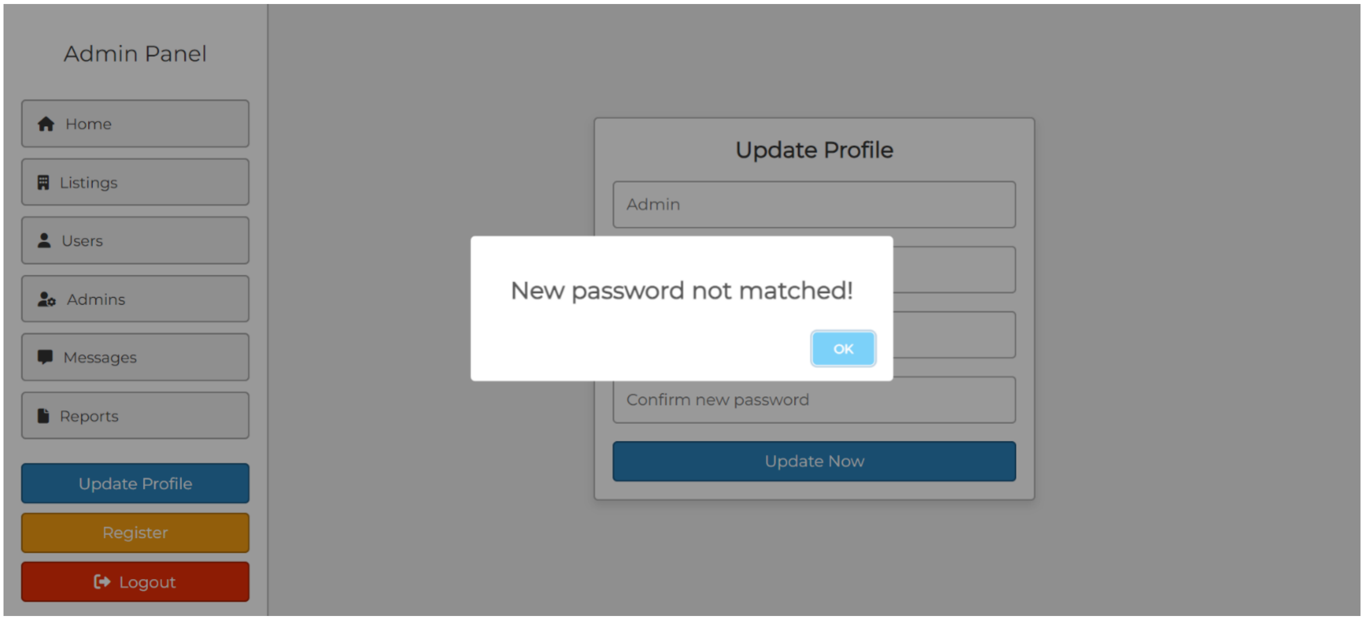
Precondition: Admin accidentally forgets the new password entered.

Assumption: Only authorized Admin have access to update their profile.

Test Steps:

1. Log in to the system as Admin using the correct user’s credentials.
2. Go to the Update Profile page.
3. Submit the updated details of the profile by entering mismatched new passwords.

Expected Result: An attempt by the Admin to update their profile by entering mismatched new passwords, results in an unsuccessful error message.



New password not matched!

Test Case 8: Check results on not correctly entering old password by Admin at Update Profile page

All the attempts to update profile must be blocked if the old password entered by the admin is wrong.

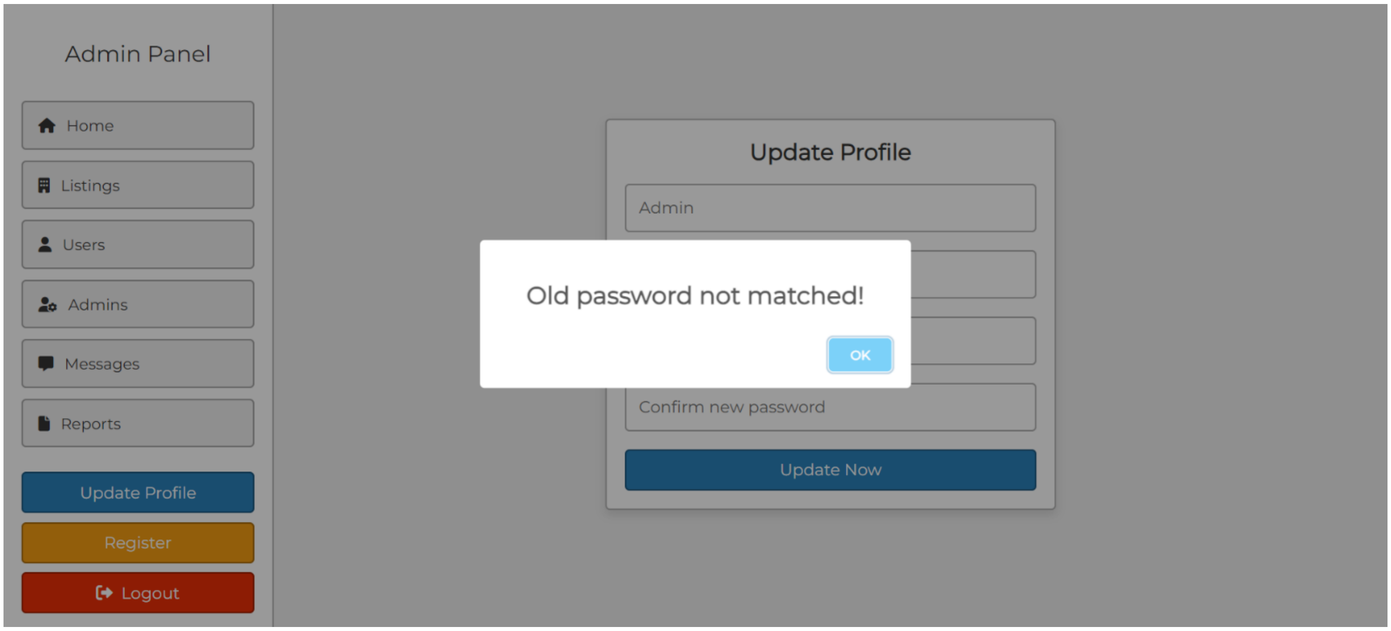
Precondition: Admin accidentally mistypes or forgets the old password.

Assumption: Only authorized Admin have access to update their profile.

Test Steps:

1. Log in to the system as Admin using the correct user's credentials.
2. Go to the Update Profile page.
3. Submit the updated details of the profile by entering wrong old password.

Expected Result: An attempt by the Admin to update their profile by entering wrong old password, results in an unsuccessful error message.



Old password not matched!

Test Case 9: Check results on validating details entered by User at Update Profile page

All the attempts to update their profile must be blocked if any of the fields are filled in an incorrect format.

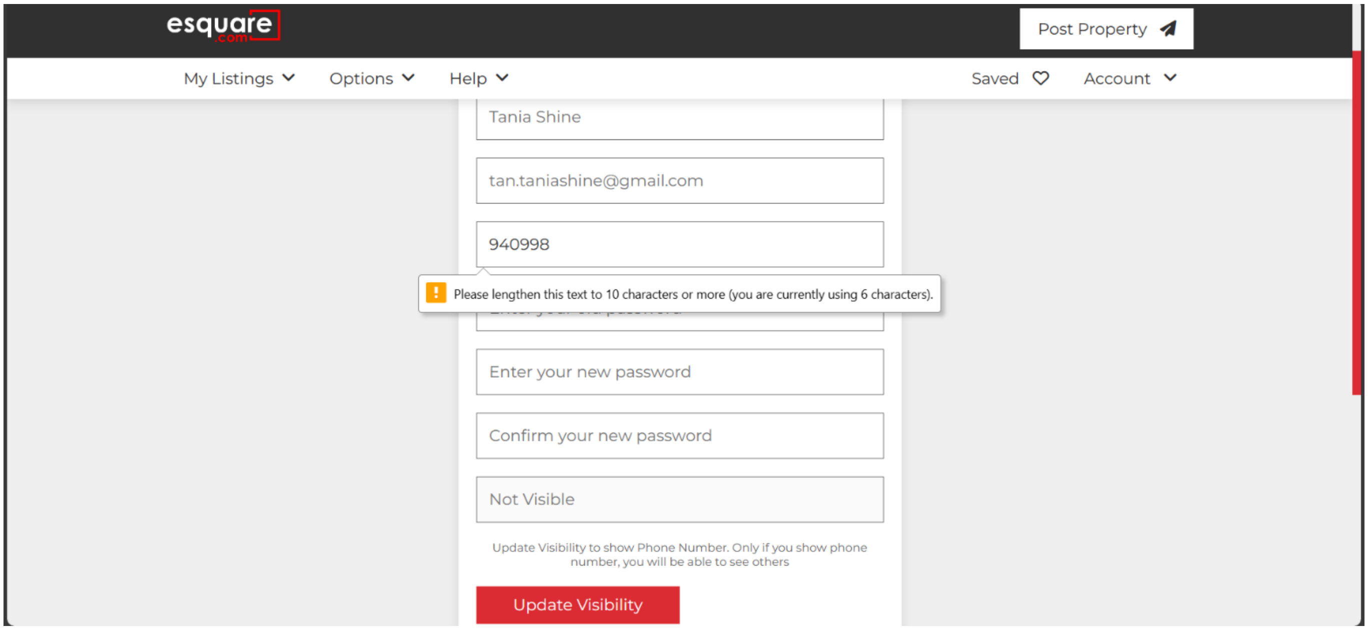
Precondition: User accidentally forgets or does not have all the necessary data to send a message.

Assumption: Only authorized users have access to send messages.

Test Steps:

1. Log in to the system as User using the correct user's credentials.
2. Go to the Update Profile page.
3. Submit the details of the profile by passing data in incorrect format.

Expected Result: An attempt by the user to update their profile by filling all or many of the fields in an incorrect format, results in an unsuccessful error message.



Tania Shine

tan.taniashine@gmail.com

940998

! Please lengthen this text to 10 characters or more (you are currently using 6 characters).

Enter your new password

Confirm your new password

Not Visible

Update Visibility to show Phone Number. Only if you show phone number, you will be able to see others

Test Case 10: Check results on checking files uploaded by User at Post Property page

All the attempts to post property must be blocked if any of the files uploaded exceeds the size limit.

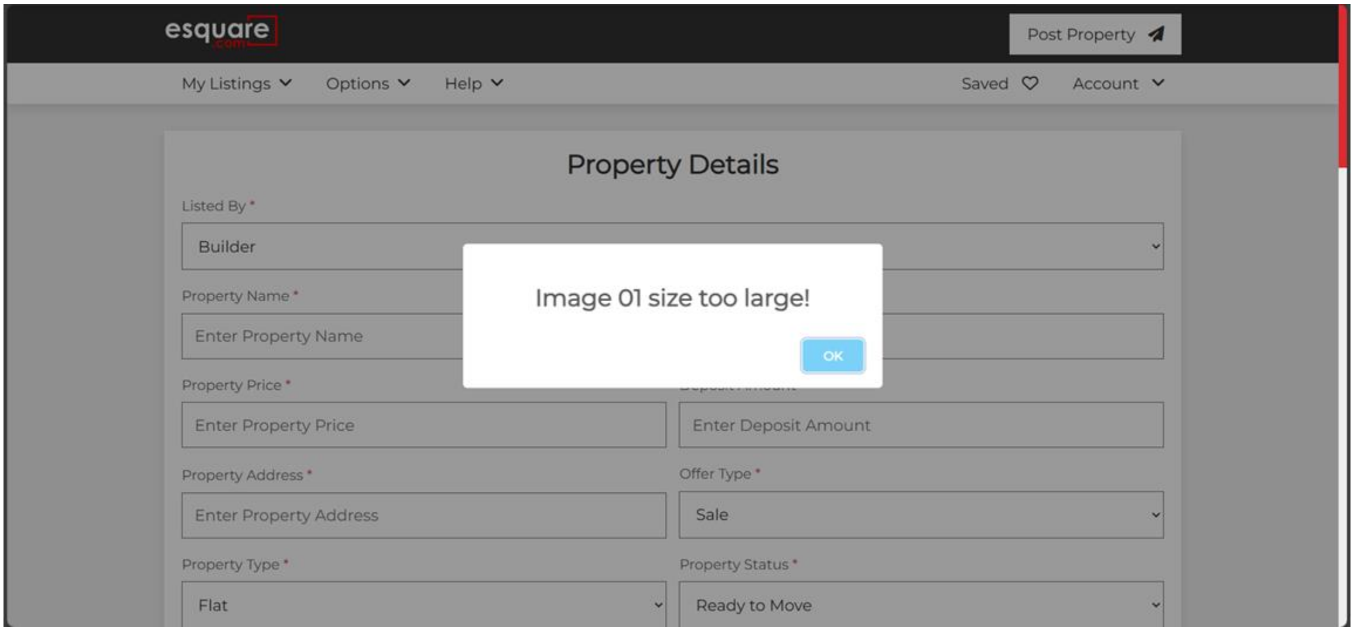
Precondition: User uploads an image of huge file size.

Assumption: Only authorized users have access to post property.

Test Steps:

1. Log in to the system as User using the correct user’s credentials.
2. Go to the Post Property page.
3. Submit the details of the property by uploading a file that exceeds the size limit.

Expected Result: An attempt by the user to post their property by uploading a file that exceeds the size limit, results in an unsuccessful error message.



Property Details

Listed By * Builder

Property Name * Enter Property Name

Image 01 size too large!

Property Price * Enter Property Price

Enter Deposit Amount

Property Address * Enter Property Address

Offer Type * Sale

Property Type * Flat

Property Status * Ready to Move

10.2 Test Report

In all the test cases, as we got the expected result, we can easily come to the conclusion that the testing process was successful and hence proved that our application is efficient enough to store and process data. Almost about 99.9% of test cases were passed successfully.

10.3 Sample Code used for testing

Test Case 1: Invalid User Login

```
if(isset($_POST['submit'])){

    $email = $_POST['email'];

    $email = filter_var($email, FILTER_SANITIZE_STRING);

    $pass = sha1($_POST['pass']);

    $pass = filter_var($pass, FILTER_SANITIZE_STRING);

    $verify_users = $conn->prepare("SELECT * FROM `users` WHERE email = ? AND
password = ? AND status = 'verified' LIMIT 1");

    $verify_users->execute([$email, $pass]);

    $row = $verify_users->fetch(PDO::FETCH_ASSOC);

    if($verify_users->rowCount() > 0){

        setcookie('user_id', $row['id'], time() + 60*60*24*30, '/');

        header('location:home.php');

    } else {

        $warning_msg[] = 'Incorrect Email or Password, or your account is not verified.';

    }

}
```

Test Case 2: Invalid Admin Login

```
  if(isset($_POST['submit'])){
     $name = $_POST['name'];
     $name = filter_var($name, FILTER_SANITIZE_STRING);
     $pass = ($_POST['pass']);
     $pass = filter_var($pass, FILTER_SANITIZE_STRING);
     $select_admins = $conn->prepare("SELECT * FROM `admins` WHERE name =
 ? AND password = ? LIMIT 1");
     $select_admins->execute([$name, $pass]);
     $row = $select_admins->fetch(PDO::FETCH_ASSOC);
     if($select_admins->rowCount() > 0){
        setcookie('admin_id', $row['id'], time() + 60*60*24*30, '/');
        header('location:dashboard.php');
     }else{
       $warning_msg[] = 'Incorrect username or password!';
     }
  }
```

Test Case 3: Check results on not entering new matching passwords by User at Update Profile page

```
if($empty_pass != $old_pass){
      if($old_pass != $prev_pass){
         $warning_msg[] = 'Old password not matched!';
      }
      elseif($c_pass != $new_pass){
         $warning_msg[] = 'Confirm password not matched!';
      }
}
```

Test Case 4: Check results on not correctly entering old password by User at Update Profile page

```
if($empty_pass != $old_pass){
        if($old_pass != $prev_pass){
            $warning_msg[] = 'Old password not matched!';
        }
        elseif($c_pass != $new_pass){
            $warning_msg[] = 'Confirm password not matched!';
        }
}
```

Test Case 5: Check results on not storing all the required fields by User at Contact Us page

```
  <form action="" method="post">
            <h3>Get in Touch</h3>
                         <input type="text" name="name" required maxlength="50"
  placeholder="Enter your Name" class="box">
                       <input type="email" name="email" required maxlength="50"
  placeholder="Enter your Email" class="box">
                   <input type="number" name="number" required maxlength="10"
  max="9999999999" min="0" placeholder="Enter your Number" class="box">
              <textarea name="message" placeholder="Enter your Message" required
  maxlength="1000" cols="30" rows="10" class="box"></textarea>
            <input type="submit" value="Send Message" name="send" class="btn">
        </form>
```

Test Case 6: Check results on validating details entered by User at Contact Us page

```
<form action="" method="post">
            <h3>Get in Touch</h3>
                         <input type="text" name="name" required maxlength="50"
  placeholder="Enter your Name" class="box">
                       <input type="email" name="email" required maxlength="50"
  placeholder="Enter your Email" class="box">
                    <input type="number" name="number" required maxlength="10"
  max="9999999999" min="0" placeholder="Enter your Number" class="box">
              <textarea name="message" placeholder="Enter your Message" required
  maxlength="1000" cols="30" rows="10" class="box"></textarea>
           <input type="submit" value="Send Message" name="send" class="btn">
        </form>
```

Test Case 7: Check results on not entering new matching passwords by Admin at Update Profile page

```
if($old_pass != $empty_pass){
    if($old_pass != $prev_pass){
      $warning_msg[] = 'Old password not matched!';
    }elseif($c_pass != $new_pass){
      $warning_msg[] = 'New password not matched!';
    }else{
      if($new_pass != $empty_pass){
        $update_password = $conn->prepare("UPDATE `admins` SET password = ? WHERE
id = ?");
        $update_password->execute([$c_pass, $admin_id]);
        $success_msg[] = 'Password updated!';
      }else{
        $warning_msg[] = 'Please enter new password!';
      }
    }
  }
```

Test Case 8: Check results on not correctly entering old password by Admin at Update Profile page

```
if($old_pass != $empty_pass){

    if($old_pass != $prev_pass){

      $warning_msg[] = 'Old password not matched!';

    }elseif($c_pass != $new_pass){

      $warning_msg[] = 'New password not matched!';

    }else{

      if($new_pass != $empty_pass){

        $update_password = $conn->prepare("UPDATE `admins` SET password = ? WHERE
id = ?");

        $update_password->execute([$c_pass, $admin_id]);

        $success_msg[] = 'Password updated!';

      }else{

        $warning_msg[] = 'Please enter new password!';

      }

    }

  }
```

Test Case 9: Check results on validating details entered by User at Update Profile page

```
<form action="" method="post">
     <h3>Update your Account</h3>
     <input type="tel" name="name" maxlength="50"
placeholder="<?=$fetch_account['name'] ?>" class="box">
     <input type="email" name="email" maxlength="50"
placeholder="<?=$fetch_account['email'] ?>" class="box">
     <input type="text" name="number" min="0" max="9999999999" minlength="10"
placeholder="<?=$fetch_account['number'] ?>" class="box">
     <input type="password" name="old_pass" maxlength="20" placeholder="Enter
```

```
your old password" class="box">
      <input type="password" name="new_pass" maxlength="20" placeholder="Enter
your new password" class="box">
      <input type="password" name="c_pass" maxlength="20" placeholder="Confirm
your new password" class="box">
      <input type="text" placeholder="<?php echo ($fetch_account['visibility'] == 0) ?
'Visible' : 'Not Visible'; ?>" class="box" disabled>
      <p style="font-size: small; position:left">Update Visibility to show Phone Number.
Only if you show phone number, you will be able to see others</p>
      <input type="hidden" name="visibility" value="<?php echo
($fetch_account['visibility'] == 0) ? 'Visible' : 'Not Visible'; ?>">
      <input type="submit" value="Update Visibility" name="show" class="btn"
style="width: 50%;" align="left">
      <input type="submit" value="Update Profile" name="submit" class="btn">
   </form>
```

Test Case 10: Check results on checking files uploaded by User at Post Property page

```
if(!empty($image_01)){
  if($image_01_size > 2000000){
    $warning_msg[] = 'Image 01 size is too large!';
  }else{
    move_uploaded_file($image_01_tmp_name, $image_01_folder);
  }
}else{
  $rename_image_01 = ";
}
```

11. TRANSITION

11.1 System Implementation

Implementation is the process of having the system personal checks out and put equipment's to use, train the users to use the new system and construct any file that are needed to see it. The final and important phases in the system lifecycle are the implementation of new system. The file conversion is the most time consuming and expensive activity in the implementation stage. System implementation refers to the step necessary to install a new system to put into the operation. The implementation has different meaning, raining from the conversion of basic application to complete replacement of computer system. Implementation includes all these activities that take place to covert from old system to new system. The new system may be totally new replacing and existing manual or automated system or it may be major modification to an existing system. The method of implementation and time scale adopted is found out initially. The system is tested properly and at the same time the users are trained in new procedure. Proper implementation is essential to provide a reliable system to meet organization requirements. Successful implementation may not guarantee improvement in the organization using the new system, but it will prevent improper installation. The implementation involves the following things:-

1. Careful planning.
2. Investigation of the system and constrains
3. Design the methods to achieve the changeover.
4. Train the staff in the changed phase.
5. Evaluation of change over method.

After converting as a package, it has been delivered to the customers where it is implemented and tailored to meet the specific requirements.

11.2 System Maintenance

Like housework, dirty clothes and weeds, system work never seems to an end; users almost always want changes or encounter problems. Thus the system maintenance part of the system process deserves special attention. It is during system maintenance that the analyst:

1. Resolves necessary changes
2. Correct errors.
3. Enhance or modifies the system.
4. Assign staff to perform maintenance activities.
5. Provides for scheduled maintenance.

Most system spends the bulk of their time in the maintenance phase, with constant enhancements and repairs. Studies show that more money is spent in this forth phase than in all of the others combined. Writing system is that require as little maintenance as possible is one of the primary goal as well as one of the benefits of today’s modern methodology of software development. Maintenance ids divided into three categories.

1. Corrective maintenance.
2. Adaptive maintenance.
3. Preventive maintenance.

11.2.1 Corrective maintenance

It has to do with the removal of residual errors present in the product when it is delivered as well as errors introduced into the software during its maintenance accounts for about 20% of the maintenance cost.

11.2.2 Adaptive maintenance

It involves adjusting the application to changes in the environment, that is a new release of the hardware or the operating system or anew database system. It also accounts for nearly 20% of the maintenance cost.

11.2.3 Preventive maintenance

It involves changing the software to improve some qualities. It accounts for over 50% of maintenance costs. Here changes are due to the functions offered by the application, and new functions, improve the performance of application etc.

Maintenance is not such a difficult task. The above three maintenance tasks can be easily carried out under this system.

12. ANNEXURE

12.1 References

Websites

[1] YouTube

[2] Random sites

https://W3schools.com/
http://www.stackoverflow.com/questions
http://tutorialspoint.com/

https://www.geekforgeeks.com

12.2 Annexure I: User Interview Questionnaires

1. How would you approach the system?

2. What about usability of this system?

3. What are normal project requirements?

12.3 CONCLUSION

The development of Esquare.com marks a significant leap in the real estate domain by introducing an automated platform for property transactions. Unlike manual processes, the automated real estate system accelerates property postings and enhances accuracy, streamlining the overall experience for users. By incorporating features such as detailed property profiles, video uploads, and improved privacy controls, the platform strives to simplify property exploration and interaction.

Real estate transactions may seem straightforward but are intricate due to various factors such as property details, inquiries, and negotiations. Esquare.com aims to alleviate the challenges associated with property transactions, making it accessible and efficient for users of all scales—small, middle-sized, and large businesses.

The software undergoes rigorous testing, employing validation techniques to eliminate errors and ensure seamless operation. The successful design and implementation of the system demonstrate its capability to provide a reliable and user-friendly platform for real estate dealings.

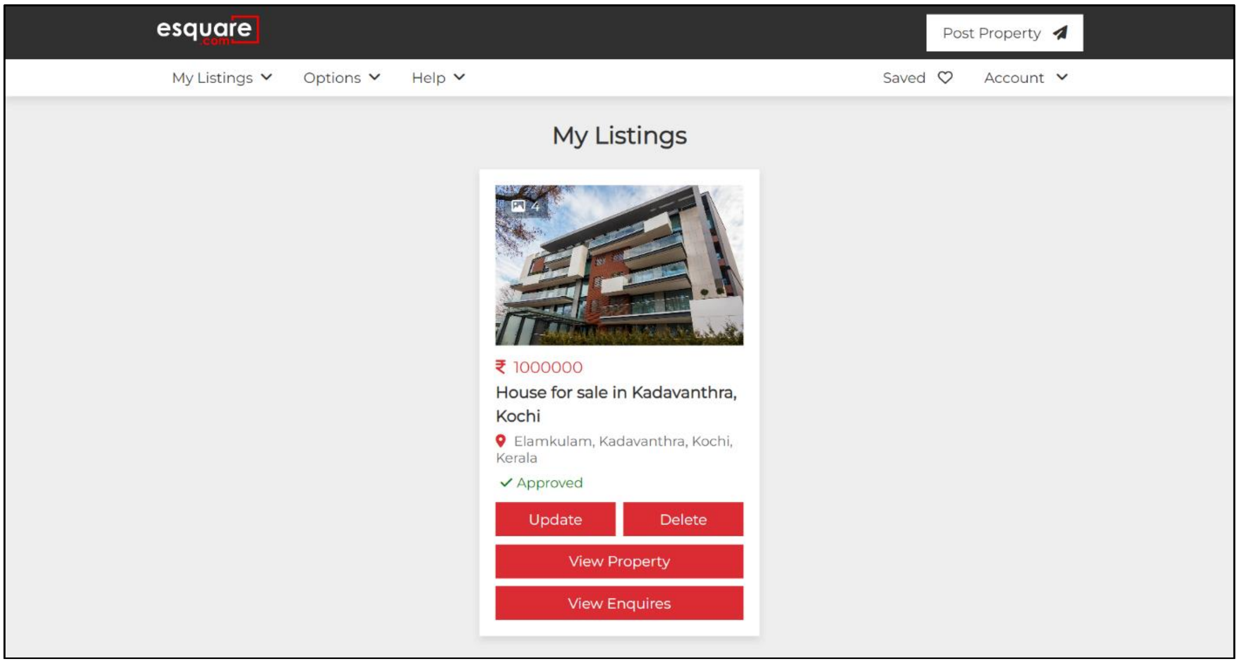
Embarking on this project has not only contributed to the enhancement of our technical skills but has also instilled confidence in our abilities as future IT professionals. We extend our gratitude to all those who supported us in the successful completion of this project.

12.4 SAMPLE CODE

- Screenshots

12.4.1. Main

USER MY LISTINGS PAGE



My Listings

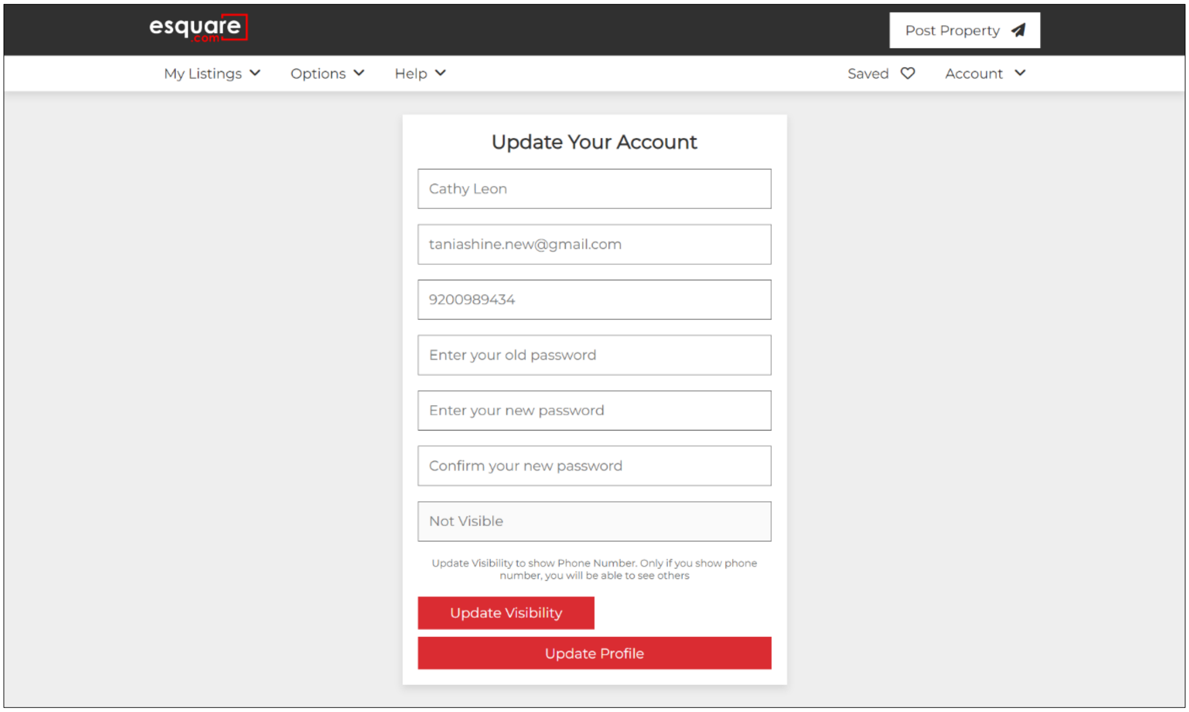
₹ 1000000

House for sale in Kadavanthra, Kochi

Elamkulam, Kadavanthra, Kochi, Kerala

✓ Approved

USER UPDATE PROFILE PAGE



Update Your Account

Cathy Leon

taniashine.new@gmail.com

9200989434

Enter your old password

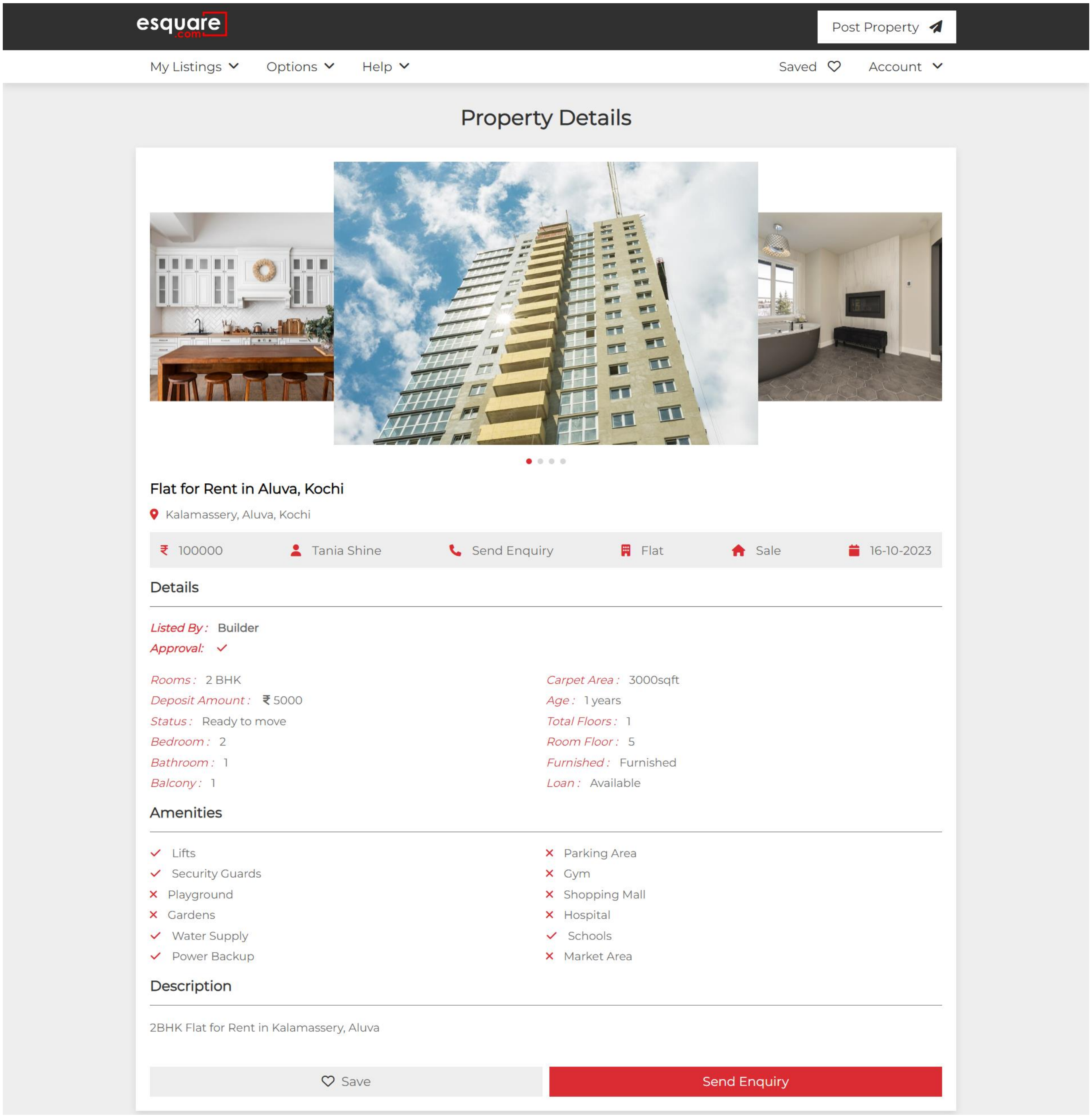
Enter your new password

Confirm your new password

Not Visible

Update Visibility to show Phone Number. Only if you show phone number, you will be able to see others

USER VIEW PROPERTY PAGE



Property Details

Flat for Rent in Aluva, Kochi

Kalamassery, Aluva, Kochi

₹ 100000 | Tania Shine | Send Enquiry | Flat | Sale | 16-10-2023

Details

Listed By : Builder
Approval: ✓

Rooms : 2 BHK
Deposit Amount : ₹ 5000
Status : Ready to move
Bedroom : 2
Bathroom : 1
Balcony : 1

Carpet Area : 3000sqft
Age : 1 years
Total Floors : 1
Room Floor : 5
Furnished : Furnished
Loan : Available

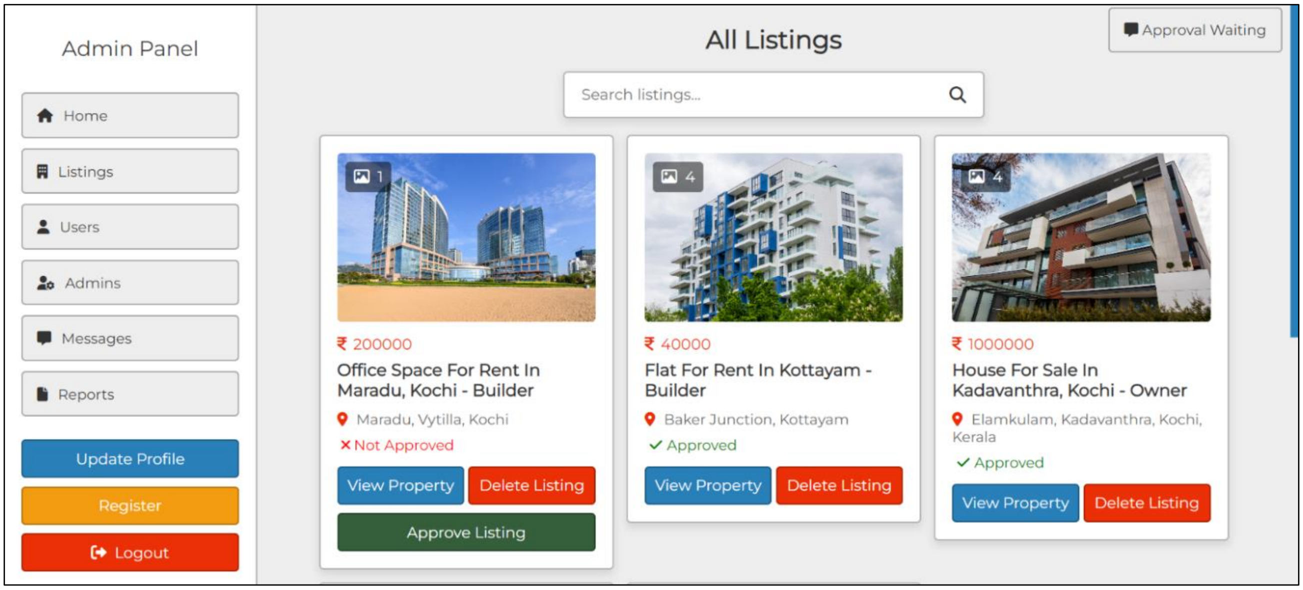
Amenities

- ✓ Lifts
- ✓ Security Guards
- ✗ Playground
- ✗ Gardens
- ✓ Water Supply
- ✓ Power Backup
- ✗ Parking Area
- ✗ Gym
- ✗ Shopping Mall
- ✗ Hospital
- ✓ Schools
- ✗ Market Area

Description

2BHK Flat for Rent in Kalamassery, Aluva

ADMIN LISTINGS PAGE

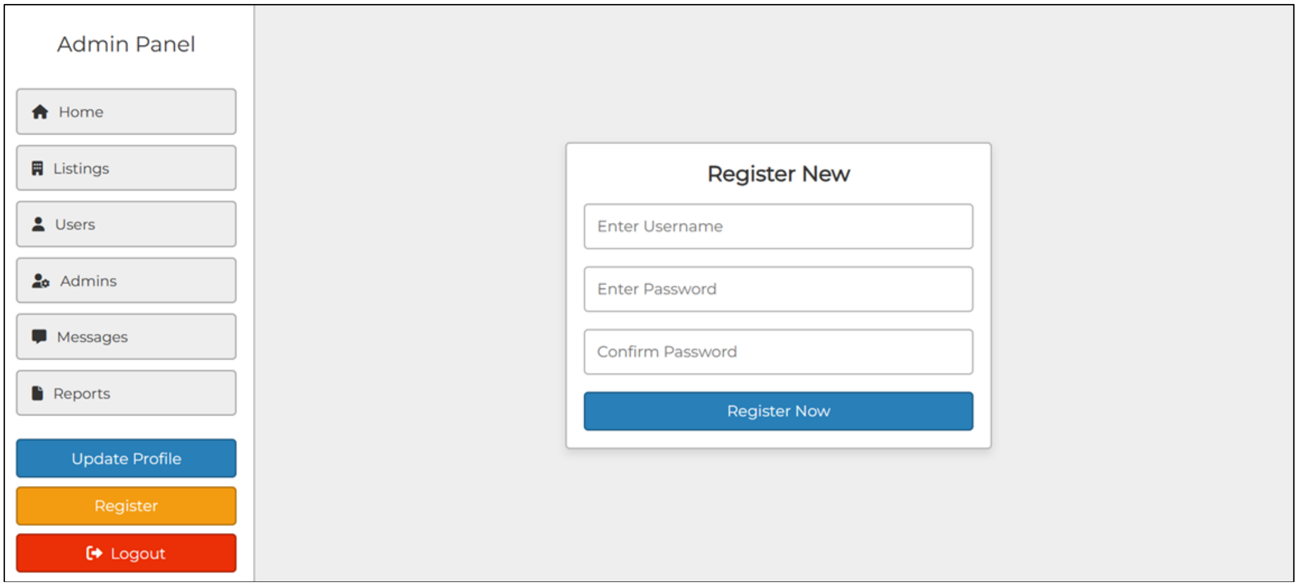


₹ 200000
Office Space For Rent In Maradu, Kochi - Builder
Maradu, Vytilla, Kochi
✕ Not Approved

₹ 40000
Flat For Rent In Kottayam - Builder
Baker Junction, Kottayam
✓ Approved

₹ 1000000
House For Sale In Kadavanthra, Kochi - Owner
Elamkulam, Kadavanthra, Kochi, Kerala
✓ Approved

ADMIN REGISTER NEW PAGE



Sample Code

12.4.2. Main

<u>My Listings Page of User</u>

my_listings.php

```
<?php
include 'components/connect.php';
if(isset($_COOKIE['user_id'])){
   $user_id = $_COOKIE['user_id'];
}
else{
   $user_id = '';
   header('location:login.php');
}

if(isset($_POST['delete'])){

   $delete_id = $_POST['property_id'];
   $delete_id = filter_var($delete_id, FILTER_SANITIZE_STRING);

   $verify_delete = $conn->prepare("SELECT * FROM `property` WHERE id = ?");
   $verify_delete->execute([$delete_id]);

   if($verify_delete->rowCount() > 0){
      $select_images = $conn->prepare("SELECT * FROM `property` WHERE id = ?");
      $select_images->execute([$delete_id]);
      while($fetch_images = $select_images->fetch(PDO::FETCH_ASSOC)){
         $image_01 = $fetch_images['image_01'];
         $image_02 = $fetch_images['image_02'];
         $image_03 = $fetch_images['image_03'];
         $image_04 = $fetch_images['image_04'];
         $image_05 = $fetch_images['image_05'];
```

```
       unlink('uploaded_files/'.$image_01);
       if(!empty($image_02)){
         unlink('uploaded_files/'.$image_02);
       }
       if(!empty($image_03)){
         unlink('uploaded_files/'.$image_03);
       }
       if(!empty($image_04)){
         unlink('uploaded_files/'.$image_04);
       }
       if(!empty($image_05)){
         unlink('uploaded_files/'.$image_05);
       }
     }
     $delete_saved = $conn->prepare("DELETE FROM `saved` WHERE property_id = ?");
     $delete_saved->execute([$delete_id]);
     $delete_requests = $conn->prepare("DELETE FROM `requests` WHERE property_id = ?");
     $delete_requests->execute([$delete_id]);
     $delete_listing = $conn->prepare("DELETE FROM `property` WHERE id = ?");
     $delete_listing->execute([$delete_id]);
     $success_msg[] = 'Listing Deleted Successfully!';
   }else{
     $warning_msg[] = 'Listing Deleted Already!';
   }
}
?>
<body>
   <?php include 'components/user_header.php'; ?>
<section class="my-listings">
      <h1 class="heading">My Listings</h1>
```

```
        <div class="box-container">
        <?php
            $total_images = 0;
                 $select_properties = $conn->prepare("SELECT * FROM `property` WHERE
user_id = ? ORDER BY date DESC");
            $select_properties->execute([$user_id]);
            if($select_properties->rowCount() > 0){
                while($fetch_property = $select_properties->fetch(PDO::FETCH_ASSOC)){

                $property_id = $fetch_property['id'];
                //counting images uploaded
                if(!empty($fetch_property['image_02'])){
                    $image_count_02 = 1;
                }else{
                    $image_count_02 = 0;
                }

                if(!empty($fetch_property['image_03'])){
                    $image_count_03 = 1;
                }else{
                    $image_count_03 = 0;
                }

                if(!empty($fetch_property['image_04'])){
                    $image_count_04 = 1;
                }else{
                    $image_count_04 = 0;
                }

                if(!empty($fetch_property['image_05'])){
                    $image_count_05 = 1;
                }else{
                    $image_count_05 = 0;
                 }

                 $total_images = (1 + $image_count_02 + $image_count_03 + $image_count_04
+ $image_count_05);

        ?>
            <form accept="" method="POST" class="box">
                <input type="hidden" name="property_id" value="<?= $property_id; ?>">
                <div class="thumb">
                    <p><i class="far fa-image"></i><span><?= $total_images; ?></span></p>
                    <img src="uploaded_files/<?= $fetch_property['image_01']; ?>" alt="">
                </div>
                <div class="price">
                      <i class="fas fa-indian-rupee-sign"></i><span><?= $fetch_property['price'];
?></span>
                </div>

                <h3 class="name"><?= $fetch_property['property_name']; ?></h3>

                            <p  class="location"><i  class="fas  fa-map-marker-alt"></i><span><?=
$fetch_property['address']; ?></span></p>
                <p style="padding: 5px;">
                    <?php if ($fetch_property['approved'] == '1') : ?>
                        <span style="font-size: 17px; color:green;"><i class="fas fa-check"></i>
                        Approved</span>
                    <?php else : ?>
                        <span style="font-size: 17px; color:red;"><i class="fas fa-times"></i>
                        Not Approved</span>
                    <?php endif; ?>
                </p>
            <div class="flex-btn">
                  <a href="update_property.php?get_id=<?= $property_id; ?>" class="btn">update</a>
                 <input type="submit" name="delete" value="delete" class="btn" onclick="return confirm('Delete this Listing?');">
            </div>
             <a href="view_property.php?get_id=<?= $property_id; ?>" class="btn">view property</a>
                 <?php if ($fetch_property['user_id'] == $user_id) { ?><a href="view_enquiry.php?get_id=<?= $property_id; ?>" class="btn">View Enquires</a>
           <?php } ?>
          </form>
      <?php
             }
          }else{
             echo '<p class="empty">No properties added yet! <a href="post_property.php" style="margin-top:1.5rem;" class="btn">add new</a></p>';
            }
      ?>
      </div>  </section>
<?php include 'components/footer.php'; ?>
</body>
```

Update Profile Page of User

update.php

```
<body>
  <?php include 'components/user_header.php'; ?>
  <section class="form-container">
  <form action="" method="post">
    <h3>Update your Account</h3>
    <input type="tel" name="name" maxlength="50" placeholder="<?=$fetch_account['name'] ?>" class="box">
    <input type="email" name="email" maxlength="50" placeholder="<?=$fetch_account['email'] ?>" class="box">
    <input type="text" name="number" min="0" max="9999999999" minlength="10" placeholder="<?=$fetch_account['number'] ?>" class="box">
    <input type="password" name="old_pass" maxlength="20" placeholder="Enter your old password" class="box">
    <input type="password" name="new_pass" maxlength="20" placeholder="Enter your new password" class="box">
    <input type="password" name="c_pass" maxlength="20" placeholder="Confirm your new password" class="box">
    <input type="text" placeholder="<?php echo ($fetch_account['visibility'] == 0) ? 'Visible' : 'Not Visible'; ?>" class="box" disabled>
    <p style="font-size: small; position:left">Update Visibility to show Phone Number. Only if you show phone number, you will be able to see others</p>
    <input type="hidden" name="visibility" value="<?php echo ($fetch_account['visibility'] == 0) ? 'Visible' : 'Not Visible'; ?>">
    <input type="submit" value="Update Visibility" name="show" class="btn" style="width: 50%;" align="left">
    <input type="submit" value="Update Profile" name="submit" class="btn">
  </form> </section>
  <?php include 'components/footer.php'; ?>
</body>
```

View Property Page of User

view_property.php

```
  <?php include 'components/user_header.php'; ?>

  <section class="view-property">
    <h1 class="heading">Property Details</h1>
    <?php
    $select_properties = $conn->prepare("SELECT * FROM `property` WHERE id = ?
ORDER BY date DESC LIMIT 1");
    $select_properties->execute([$get_id]);
    if($select_properties->rowCount() > 0){
      while($fetch_property = $select_properties->fetch(PDO::FETCH_ASSOC)){

      $property_id = $fetch_property['id'];

      $select_user = $conn->prepare("SELECT * FROM `users` WHERE id = ?");
      $select_user->execute([$fetch_property['user_id']]);
      $fetch_user = $select_user->fetch(PDO::FETCH_ASSOC);

      $select_our = $conn->prepare("SELECT * FROM `users` WHERE id = ?");
      $select_our->execute([$user_id]);
      $fetch_our = $select_our->fetch(PDO::FETCH_ASSOC);

      $select_saved = $conn->prepare("SELECT * FROM `saved` WHERE property_id
= ? and user_id = ?");
      $select_saved->execute([$fetch_property['id'], $user_id]);
    ?>
    <div class="details">
    <div class="swiper images-container">
      <div class="swiper-wrapper">
        <img src="uploaded_files/<?= $fetch_property['image_01']; ?>" alt=""
class="swiper-slide">
```

```
        <?php if(!empty($fetch_property['image_02'])){ ?>
        <img src="uploaded_files/<?= $fetch_property['image_02']; ?>" alt=""
class="swiper-slide">
        <?php } ?>
        <?php if(!empty($fetch_property['image_03'])){ ?>
        <img src="uploaded_files/<?= $fetch_property['image_03']; ?>" alt=""
class="swiper-slide">
        <?php } ?>
        <?php if(!empty($fetch_property['image_04'])){ ?>
        <img src="uploaded_files/<?= $fetch_property['image_04']; ?>" alt=""
class="swiper-slide">
        <?php } ?>
        <?php if(!empty($fetch_property['image_05'])){ ?>
        <img src="uploaded_files/<?= $fetch_property['image_05']; ?>" alt=""
class="swiper-slide">
        <?php } ?>
      </div>
      <div class="swiper-pagination"></div>
    </div>
    <h3 class="name"><?= $fetch_property['property_name']; ?></h3>
    <p class="location"><i class="fas fa-map-marker-alt"></i><span><?=
$fetch_property['address']; ?></span></p>
    <div class="info">
      <p><i class="fas fa-indian-rupee-sign"></i><span><?= $fetch_property['price'];
?></span></p>
      <p><i class="fas fa-user"></i><span><?= $fetch_user['name']; ?></span></p>
      <?php
      if ($fetch_our['visibility'] == 0 && $fetch_user['visibility'] == 0) {
      ?>
        <p><i class="fas fa-phone"></i><a href="tel:<?= $fetch_user['number'];
?>"><?= $fetch_user['number']; ?></a></p>
      <?php
      } else if ($fetch_user['visibility'] != 0) {
      ?>
        <p><i class="fas fa-phone"></i><a
href="mailto:tan.taniashine@gmail.com">Send Enquiry</a></p>
      <?php
      } else {
      ?>
        <p><i class="fas fa-phone"></i><a href="update.php">Not Visible</a></p>
      <?php
      }
      ?>

      <p><i class="fas fa-building"></i><span><?= $fetch_property['type'];
?></span></p>
      <p><i class="fas fa-house"></i><span><?= $fetch_property['offer'];
?></span></p>
      <p><i class="fas fa-calendar"></i><span><?= date("d-m-Y",
strtotime($fetch_property['date']));?></span></p>
    </div>
    <h3 class="title">Details</h3>
    <div class="flex">
      <div class="box">
      <p><b><i>Listed By :</i><span><?= $fetch_property['listed_by'];
?></span></b></p>
      <p>
        <i><b>Approval: </b></i>
        <?php if ($fetch_property['approved'] == '1') : ?>
          <i class="fas fa-check"></i>
        <?php else : ?>
          <i class="fas fa-times"></i>
          <span style="color:brown">Wait for admins approval. <a
href="mailto:tan.taniashine@gmail.com">Contact Support</a></span>
```

```
        <?php endif; ?>
      </p>
      </div>
    </div>
    <div class="flex">
      <div class="box">
        <p><i>Rooms :</i><span><?= $fetch_property['bhk']; ?> BHK</span></p>
        <p><i>Deposit Amount : </i><span><span class="fas fa-indian-rupee-sign"
style="margin-right: .5rem;"></span><?= $fetch_property['deposite']; ?></span></p>
        <p><i>Status :</i><span><?= $fetch_property['status']; ?></span></p>
        <p><i>Bedroom :</i><span><?= $fetch_property['bedroom']; ?></span></p>
        <p><i>Bathroom :</i><span><?= $fetch_property['bathroom']; ?></span></p>
        <p><i>Balcony :</i><span><?= $fetch_property['balcony']; ?></span></p>
      </div>
      <div class="box">
        <p><i>Carpet Area :</i><span><?= $fetch_property['carpet'];
?>sqft</span></p>
        <p><i>Age :</i><span><?= $fetch_property['age']; ?> years</span></p>
        <p><i>Total Floors :</i><span><?= $fetch_property['total_floors'];
?></span></p>
        <p><i>Room Floor :</i><span><?= $fetch_property['room_floor'];
?></span></p>
        <p><i>Furnished :</i><span><?= $fetch_property['furnished']; ?></span></p>
        <p><i>Loan :</i><span><?= $fetch_property['loan']; ?></span></p>
      </div>
    </div>
    <h3 class="title">Amenities</h3>
    <div class="flex">
      <div class="box">
        <p><i class="fas fa-<?php if($fetch_property['lift'] == 'Yes'){echo
'check';}else{echo 'times';} ?>"></i><span>Lifts</span></p>
        <p><i class="fas fa-<?php if($fetch_property['security_guard'] == 'Yes'){echo
```

```
'check';}else{echo 'times';} ?>"></i><span>Security Guards</span></p>
        <p><i class="fas fa-<?php if($fetch_property['play_ground'] == 'Yes'){echo
'check';}else{echo 'times';} ?>"></i><span>Playground</span></p>
        <p><i class="fas fa-<?php if($fetch_property['garden'] == 'Yes'){echo
'check';}else{echo 'times';} ?>"></i><span>Gardens</span></p>
        <p><i class="fas fa-<?php if($fetch_property['water_supply'] == 'Yes'){echo
'check';}else{echo 'times';} ?>"></i><span>Water Supply</span></p>
        <p><i class="fas fa-<?php if($fetch_property['power_backup'] == 'Yes'){echo
'check';}else{echo 'times';} ?>"></i><span>Power Backup</span></p>
      </div>
      <div class="box">
        <p><i class="fas fa-<?php if($fetch_property['parking_area'] == 'Yes'){echo
'check';}else{echo 'times';} ?>"></i><span>Parking Area</span></p>
        <p><i class="fas fa-<?php if($fetch_property['gym'] == 'Yes'){echo
'check';}else{echo 'times';} ?>"></i><span>Gym</span></p>
        <p><i class="fas fa-<?php if($fetch_property['shopping_mall'] == 'Yes'){echo
'check';}else{echo 'times';} ?>"></i><span>Shopping Mall</span></p>
        <p><i class="fas fa-<?php if($fetch_property['hospital'] == 'Yes'){echo
'check';}else{echo 'times';} ?>"></i><span>Hospital</span></p>
        <p><i class="fas fa-<?php if($fetch_property['school'] == 'Yes'){echo
'check';}else{echo 'times';} ?>"></i><span>Schools</span></p>
        <p><i class="fas fa-<?php if($fetch_property['market_area'] == 'Yes'){echo
'check';}else{echo 'times';} ?>"></i><span>Market Area</span></p>
      </div>
    </div>
    <h3 class="title">Description</h3>
    <p class="description"><?= $fetch_property['description']; ?></p>
    <br><br>
    <!-- Video section -->
    <?php if (!empty($fetch_property['video_name'])) { ?>
      <h3 class="title">Video Tour</h3>
      <section class="video-section">
```

```
            <div class="video-container">
               <video controls width="100%">
                  <source src="uploaded_files/<?= $fetch_property['video_name']; ?>"
type="video/mp4">
                  Your browser does not support the video tag.
               </video>
            </div>
            </section>
         <?php } ?>

         <?php if (!empty($fetch_property['view'])) { ?>
            <h3 class="title">Virtual Tour</h3>
            <section class="section">
            <div class="container">
               <iframe width="100%" height="300px" src="<?= $fetch_property['view'];
?>"></iframe>
            </div>
            </section>
         <?php } ?>

         <form action="" method="post" class="flex-btn">
            <input type="hidden" name="property_id" value="<?= $property_id; ?>">
            <?php
               if($select_saved->rowCount() > 0){
            ?>
            <button type="submit" name="save" class="save"><i class="fas fa-
heart"></i><span>Saved</span></button>
            <?php
               }else{
            ?>
            <button type="submit" name="save" class="save"><i class="far fa-
heart"></i><span>Save</span></button>
```

```
            <?php
               }
            ?>
            <?php if ($fetch_property['user_id'] != $user_id) { ?>

               <input type="submit" value="send enquiry" name="send" class="btn">
              <?php } ?>
              <?php if ($fetch_property['user_id'] == $user_id) { ?><a
href="view_enquiry.php?get_id=<?= $property_id; ?>" class="btn">View Enquires</a>
              <?php } ?>
         </form>
         </div>
         <?php
         }
         }else{
         echo '<p class="empty">property not found! <a href="post_property.php"
style="margin-top:1.5rem;" class="btn">add new</a></p>';
         }
         ?>
   </section>

<?php include 'components/footer.php'; ?>

</body>
```

Listings Page of Admin

admin/listings.php

```
<body>
<?php include '../components/admin_header.php'; ?>

<section class="listings" style="text-transform: capitalize;">
  <h1 class="heading">All Listings</h1>
  <div class="bar">
    <a href="approval.php"><i class="fas fa-message"></i><span> Approval
Waiting</span></a>
  </div>

  <form action="" method="POST" class="search-form">
    <input type="text" name="search_box" placeholder="Search listings..." maxlength="100"
required>
    <button type="submit" class="fas fa-search" name="search_btn"></button>
  </form>

  <div class="box-container">
  <?php
    if(isset($_POST['search_box']) OR isset($_POST['search_btn'])){
      $search_box = $_POST['search_box'];
      $search_box = filter_var($search_box, FILTER_SANITIZE_STRING);
      $select_listings = $conn->prepare("SELECT * FROM `property` WHERE property_name
LIKE '%{$search_box}%' OR address LIKE '%{$search_box}%' ORDER BY date DESC");
      $select_listings->execute();
    }else{
      $select_listings = $conn->prepare("SELECT * FROM `property` ORDER BY date
DESC");
      $select_listings->execute();
    }
    //for counting total no of imgs uploaded
    $total_images = 0;
     if($select_listings->rowCount() > 0){
      while($fetch_listing = $select_listings->fetch(PDO::FETCH_ASSOC)){
      $listing_id = $fetch_listing['id'];
      $select_user = $conn->prepare("SELECT * FROM `users` WHERE id = ?");
      $select_user->execute([$fetch_listing['user_id']]);
      $fetch_user = $select_user->fetch(PDO::FETCH_ASSOC);

      if(!empty($fetch_listing['image_02'])){
        $image_count_02 = 1;
      }else{
        $image_ count _02 = 0;
      }
      if(!empty($fetch_listing['image_03'])){
        $image_ count _03 = 1;
      }else{
        $image_ count _03 = 0;
      }
      if(!empty($fetch_listing['image_04'])){
        $image_ count _04 = 1;
      }else{
        $image_ count _04 = 0;
      }
      if(!empty($fetch_listing['image_05'])){
        $image_ count _05 = 1;
      }else{
        $image_ count _05 = 0;
      }

      $total_images = (1 + $image_ count _02 + $image_ count _03 + $image_ count _04 +
$image_ count _05);   ?>
```

```
  <div class="box">
    <div class="thumb">
      <p><i class="far fa-image"></i><span><?= $total_images; ?></span></p>
      <img src="../uploaded_files/<?= $fetch_listing['image_01']; ?>" alt="">
    </div>
    <p class="price"><i class="fas fa-indian-rupee-sign"></i><?= $fetch_listing['price']; ?></p>
    <h3 class="name"><?= $fetch_listing['property_name']; ?> - <?= $fetch_listing['listed_by'];
?></h3>
    <p class="location"><i class="fas fa-map-marker-alt"></i><?= $fetch_listing['address'];
?></p>
    <p style="padding: 5px;">
             <?php if ($fetch_listing['approved'] == '1') : ?>
                <span style="font-size: 17px; color:green;"><i class="fas fa-check"></i>
                Approved</span>
             <?php else : ?>
                <span style="font-size: 17px; color:red;"><i class="fas fa-times"></i>
                Not Approved</span>
             <?php endif; ?>
          </p>

    <form action="" method="POST">
      <input type="hidden" name="delete_id" value="<?= $listing_id; ?>">
      <input type="hidden" name="approve_id" value="<?= $listing_id; ?>">
      <div class="button-container" style=" display: flex; gap:5px;">
        <a href="view_property.php?get_id=<?= $listing_id; ?>" class="btn">view property</a>
        <input type="submit" value="delete listing" onclick="return confirm('Delete this
Listing?');" name="delete" class="delete-btn">
      </div>
      <?php if ($fetch_listing['approved'] == '0') { ?><input type="submit" value="approve
listing" onclick="return confirm('Approve this Listing?');" name="approve" class="approve-
btn">
      <?php } ?>
```

```
        </form>
    </div>
    <?php
        }
      }elseif(isset($_POST['search_box']) OR isset($_POST['search_btn'])){
        echo '<p class="empty">No results Found!</p>';
      }else{
        echo '<p class="empty">No property posted yet!</p>';
      }
    ?>
    </div>
</section>
</body>
```

Register Admin Page of Admin

admin/register.php

```
    <body>
    <?php include '../components/admin_header.php'; ?>
    <section class="form-container" style="text-transform: capitalize;">
      <form action="" method="POST">
        <h3>register new</h3>
        <input type="text" name="name" placeholder="Enter Username" maxlength="20" class="box" required oninput="this.value = this.value.replace(/\s/g, '')">
        <input type="password" name="pass" placeholder="Enter Password" maxlength="20" class="box" required oninput="this.value = this.value.replace(/\s/g, '')">
        <input type="password" name="c_pass" placeholder="Confirm Password" maxlength="20" class="box" required oninput="this.value = this.value.replace(/\s/g, '')">
        <input type="submit" value="register now" name="submit" class="btn">
      </form>
    </section>
    </body>
```