# ASHESI UNIVERSITY

## A FREE WEB APPLICATION FOR THE GHANAIAN REAL-ESTATE MARKET

APPLIED PROJECT

B. Sc. Management Information Systems

Kweku Amankwa Odum

2021

**ASHESI UNIVERSITY**

# A FREE WEB APPLICATION FOR THE GHANAIAN REAL-ESTATE MARKET

## APPLIED PROJECT

Applied Project submitted to the Department of Computer Science and Information Systems, Ashesi University in partial fulfillment of the requirements for the award of Bachelor of Science degree in Management Information Systems.

**Kweku Amankwa Odum**

**April 2021**

# DECLARATION

I hereby declare that this Applied Project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……………………………………………………………………………………..…

Candidate's Name:

**Kweku Amankwa Odum………………..…………………………………………**

Date:

May 14, 2021**……………………………………………………………..………**

I hereby declare that preparation and presentation of this Applied Project were supervised in accordance with the guidelines on supervision of Applied Project laid down by Ashesi University.

Supervisor's Signature:

**…………………………………………………………………………………………**

Supervisor's Name:

Akua O. Ahenkorah **……………………………………………………………………**

Date:

May 14, 2021 **…………………………………………………………………**

# Acknowledgement

I would first like to thank God for the successful completion of this project. I would like to show my appreciation to my supervisor Akua Ahenkorah for her support, guidance, and advice throughout this project.

I would also like to thank my father, Mr Kwesi Odum, my mother, Mrs Joyce Odum, my aunt, Mrs Dorothy Nketsia-Tabiri and my best friend Sally Anorkor Abbey for their support in helping me navigate school and project work during this COVID-19 pandemic.

This project would not have been possible without the knowledge I acquired from my lecturers throughout my undergraduate program. I want to express my sincere gratitude to all faculty, lecturers in the Computer Science and Information Systems Department.

# Abstract

Technology has had a significant impact on the world and revolutionized all its industries. The real-estate industry has incorporated technology in its activities by using the internet and services like social media, web applications, and websites to ease accessibility and widen the reach of advertisements. The steady growth of Ghana's real estate industry has seen a rise in online property advertisements. This move online has introduced some challenges in terms of the propriety of online advertising mediums. As a primary advertising medium, social media ensures a broad reach, but competition from other content is likely to stifle engagements if you are not a recognized brand. Web applications and websites will be the ideal alternatives, but most of the existing ones are poorly managed; layouts are rough, content has disparities and advertisers are charged exorbitant fees. This project aims to create a web application that will aid Ghana's real-estate industry in an advertisement without worrying about efficiency and expenses. The web application will be developed using HTML, CSS, JavaScript, PHP, and SQL.

# Table of contents

# Chapter 1: Introduction

## 1.1 Background – The Internet and Real Estate

The growing popularity of the internet has had an impact on a lot of industries. The real estate industry, for one, with the help of the internet, has broadened its market and increased its reach by moving property advertisements online. Real-estate websites and web applications are an investment in advertising that benefit realties and individual property owners with their broad reach, ease of accessibility and ability to overcome geological constraints. According to the website www.owners.com in the year 2000, as cited by Delcoure and Miller, more than one million Americans sold their homes by themselves due to the real estate industry becoming more web-based[6]. The majority of realties develop custom websites or web applications. Individual property owners will then have to consult realties to sell their property and at a fee before charging commission when the property is sold. Websites and web applications that are not custom-built and allow both property owners and realties to advertise their property are a great alternative.

## 1.2 Problem Description

In Ghana, the real estate market has made a progressive move online. Realtors and property owners advertise their listings on social media platforms like Twitter, Facebook, Instagram, and Snapchat. Realties have also set up websites and web applications to promote their property. The websites have problems that make the overall user experience not a good one. Some listings have the same images but different descriptions, a generally lousy layout and google map implementations that do not work accurately. These firms also charge users exorbitant fees to advertise their property listings.

### 1.3 Proposed Solution

Abode is a web application that seeks to create a more user-friendly house-searching experience. The web application caters to three different user groups. The user looking to buy or rent a house, and the property owner looking to sell a house or put it up for rent, and the realty looking for a space to upload their listings and manage their agents and properties.

Users that are putting properties up for rent will have an admin dashboard that will allow them to manage the listings and all related details. Realties will be able to manage their agents and the properties they oversee. This project aims to provide a web application that caters to the needs of all user groups that will interact with it. The web application will also tackle the issue of content discrepancies with restb.ai's real estate solutions.

The web application will be developed with a focus on reliability, security, scalability, and maintainability to make it easy to service the web application.

#### 1.3.1 Restb.ai

Restb.ai is a company that uses AI to solve issues with real estate. It was launched in 2015 when founder Angel Estaban was looking for a home for his wife and newborn son and browsing through properties, noticed the inconsistencies with the quality of the listings and other difficulties he went through. Using his extensive background in AI, he started to create solutions for real estate by focusing on property photos, using the latest developments in computer vision[2].

### 1.3.2 Sherlock

Sherlock is an API-based tool developed by restb.ai to recognize visual similarities in property images. Sherlock allows you to scan, analyze and compare all the property images in your database. Its visual similarities solution works by breaking down each photo in the database into their unique digital fingerprint, which is made up of over 1000 visual markers. The AI then determines the images with the most similar fingerprints or look the most alike.[10]

Restb.ai and its solutions, when implemented in the Abode web application, will make it one of the best real estate web applications in the Ghanaian market.

## 1.4 Existing Solutions

**SellRent Ghana** is a brokerage firm that offers comprehensive real estate services to property owners, home buyers, tenants, real estate developers and investors. They also provide market research, property management and valuation. To advertise your property at SellRent, you will have to send your details and that of your property to them and have them upload it for you. They do not give you an option to manage your property. [1]

**Meqasa** is a website that advertises properties for sale or rent and land for sale. Their platform is for realties, property developers, and individual property owners. They also provide real estate advice. Meqasa offers packages to users who want to upload listings, with the lowest being GHS 495.00 and the highest being GHS 7450.00.[4]

**Ghanaprimeproperties** is a website that advertises properties that are available for rent and sale. Users must sign up and sign in to upload a listing, but the register page produced an error.[12]

### 1.5 Motivation

I have always wanted to work in the real estate industry, and I wanted to apply for my capstone project. This motivated me to research how I could do something that will benefit Ghanaian real estate.

In Ghana, real estate marketing is mainly done on social media by realtors new to the business or individual property owners. This is not the wrong way to advertise; however, with the frequency with which users post on social media, the advertisements have a high chance of getting lost. Websites that provide the avenue to advertise property listings also charge high rates that realtors and property owners would most likely have trouble paying.

And that is what motivated me to create a system that will allow realtors and individual property owners to advertise their listings without worrying about paying fees and is also just as good as the ones on the market.

I chose to build a web application because it is easy to access on several devices; the only thing required is an internet connection and a browser.

# Chapter 2 – Requirements

## 2.1 Requirements gathering

For the Abode web application, the primary stakeholders are realtors and property owners who are looking to sell and secondary, are people looking to buy property. A google form was sent out to inquire about people, their experience with online property shopping. The survey asked them about their experiences with property websites, property being advertised on social media, and challenges they experienced (if any).

After viewing and analyzing the results, the indication was that people looking to buy houses liked to visit property websites because of convenience. Realtors and property owners looking to sell liked the idea of property websites because they allowed them to reach many people. The majority of people looking to buy houses complained about inconsistencies in the content. Images and descriptions do not match, and some reported that the pictures on some of the websites did not correspond with the actual property.

Both parties complained about the layout and appearance of some websites, saying they are not good enough and are difficult to navigate.

Figure 6.1 in the appendix shows a picture of the requirements gathering form.

Surfing through existing websites and web applications and noting the issues and problems allowed me to draw up the requirements document for the web application. Making it a point to cover all the issues noticed in the existing web applications and websites.

### 2.2 Features of The Web Application

Users will be required to register and log in to access the different aspects of the web application based on their user privileges. Abode will allow clients to view listings, indicate interest in property listings and manage the properties they are interested in, but only when they are logged in.

Likewise, property owners and realties can upload listings, edit the details of a listing, check the details of a client who has shown interest in a listing.

### 2.3 Operating Environments

Abode is a web application and was developed using HTML (Hypertext Markup Language), CSS (Cascading Style Sheet), JavaScript, PHP (Hypertext Preprocessor), Ajax (Asynchronous JavaScript and XML) and MySQL (Structured Query Language).

### 2.4 Functional and Non-Functional Requirements

#### 2.4.1 Functional Requirements

The two stakeholder parties can perform the following functions:

- Ability to create an account and log in to that account.
- Ability to upload property listings.

- Ability to delete property listings.
- Ability to edit property listings.
- Ability to manage agents (Realties only).

Potential buyers will be able to:

- View property listings
- Interact with content, i.e., view location in google maps, view pictures.
- Indicate interest in a listing.
- Report a listing

### 2.4.2 Non-Functional Requirements

Security – Information collected from realtors, property owners and potential buyers for identification and verification purposes will be stored and protected against unauthorized access. A potential buyer will not be able to access any of the admin dashboards. The web application encrypts passwords and stores the encrypted password. The developer cannot access content uploaded by realtors and property owners.

Reliability – The Abode web application will always be accessible and fully operational. At every point in time, the web application will be running efficiently and consistently, performing all functions without failure.

Scalability – The web application will handle a growing number of users without suffering any form of stress or strain.

Maintainability – The web application will be easy to maintain. The developer can rectify any issues encountered quickly and efficiently without interfering with content on the web application or user interaction.

# Chapter 3 – System Architecture

The Abode web application will be using the Three Tier Architecture and the Model-View-Controller design.

## 3.1 Three Tier Architecture

**Presentation Tier –** This layer of the architecture focuses on the front-end of the web application and consists of the user interface. The user interface refers to the parts of the web application that the user interacts with, which is mainly the web content.[7] This tier will be developed using web technologies like HTML, CSS, Javascript, Ajax and the Bootstrap framework.

**Application Tier –** This tier contains the functional logic of the web application; its back-end processes. It is responsible for the core functionalities and capabilities of the application. [7] This tier will be developed using PHP.

**Data Tier –** This tier comprises the database and the data access layer. This layer will be developed using MYSQL. [7]The presentation tier will access its content from the data tier through API calls.

## 3.2 Model-View-Controller (MVC) Design

The MVC is a design model that allows the developer to reuse object code. It is essential to develop applications that involve user interfaces because it takes shorter times compared to other models.[9]

**Model –** This represents the logical structure of data in the application software. It defines where the data objects are stored. It is independent of the View and Controller.

**View –** This is responsible for all elements of the user interface. It concerns itself with the part of the web application the end-user will interact with.

**Controller –** This acts as a bridge between the model and the view. The controller passes data between the two, getting data from the model and giving it to the view for display.

The MVC model was used because all three components are independent of each other, allowing them to simultaneously reduce the time taken to develop the system and maintenance.

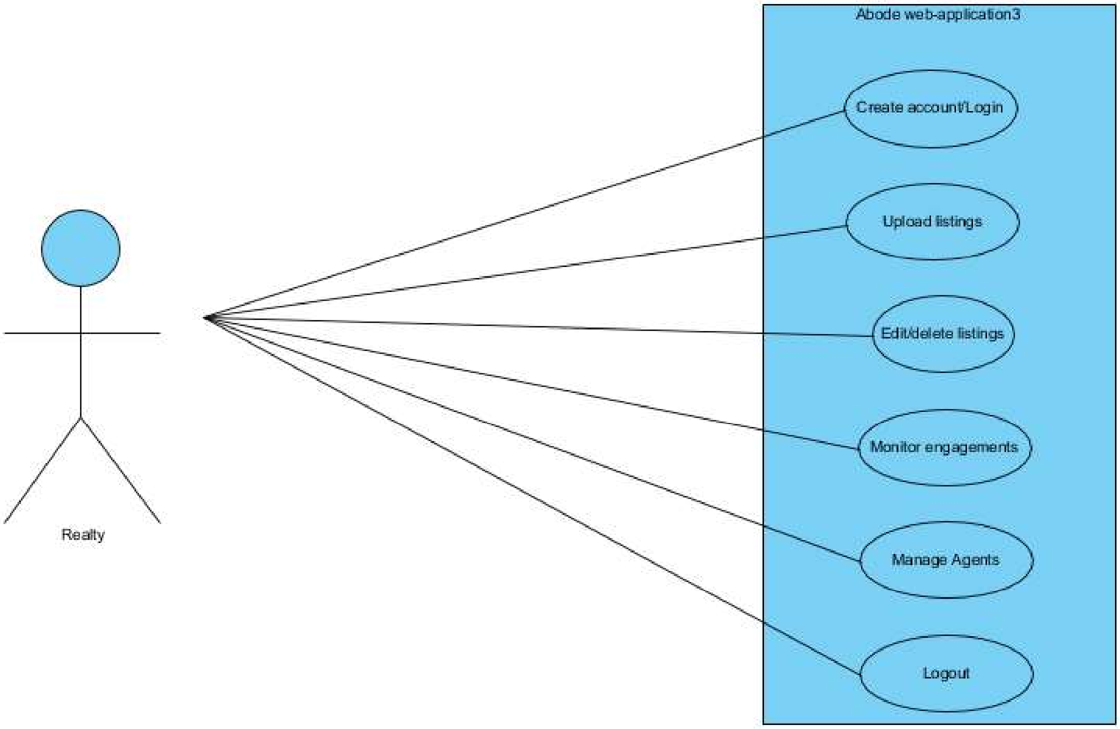
### 3.3 Use Case Diagram

A use case diagram is a visual representation of the user's expected interaction with a system.

Below are the use case diagrams for realtors/property owners, and potential buyers, respectively.

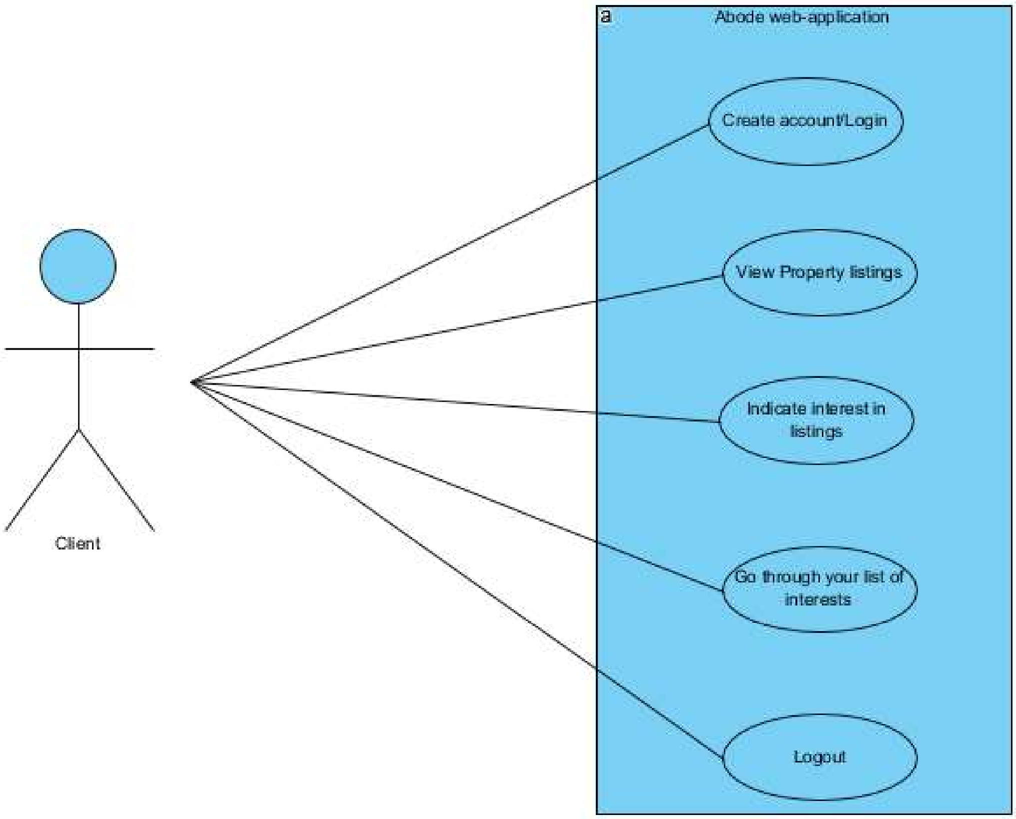
### 3.3.1 Use Case Diagram for Realtors

---



The Realtor can register, upload property listings, remove them when the property has been sold and manage agents (add/remove/assign to property). They can send emails to potential buyers and can monitor engagements with their listings.

### 3.3.2 Use Case Diagram for Potential Buyers



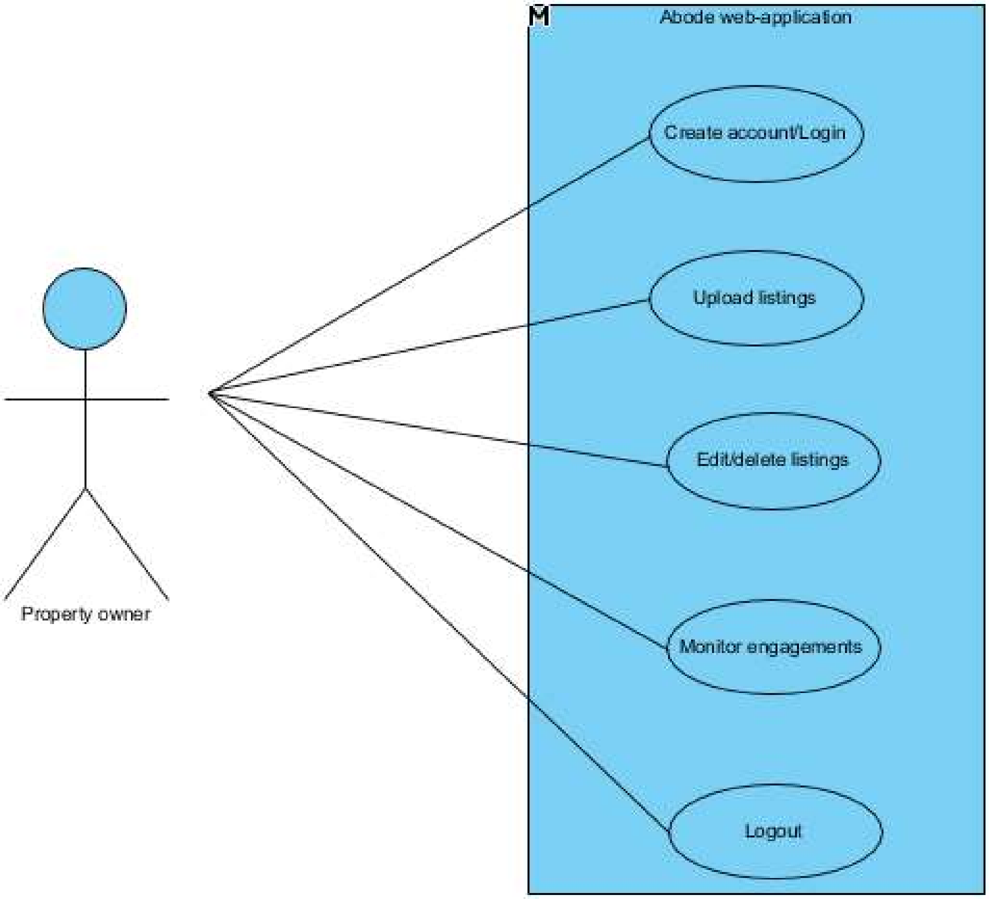
Potential buyers will be able to register and log in, indicate their interest in any property listing; they will also be able to go through the listings they showed interest in and log out when done.

### 3.1.3 Use Case for Property Owners

### 3.4 Entity Relationship Diagram

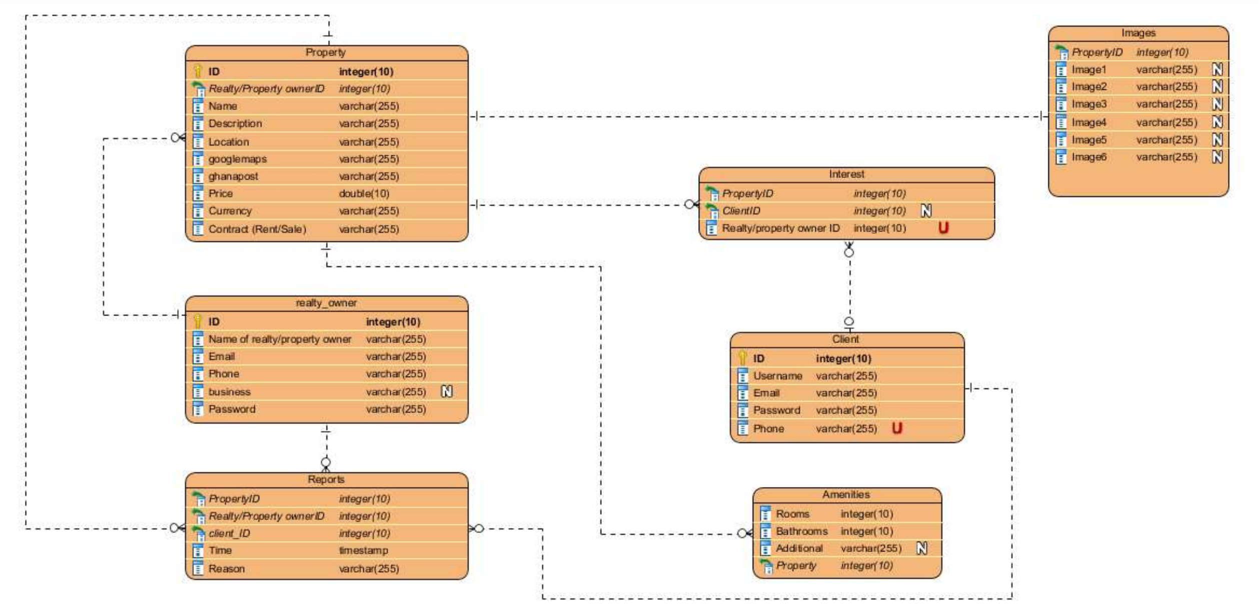
### 3.3.3 Use Case for Property Owners

---



Property owners will be able to register, login, add, edit and delete listings. They will also be able to monitor engagements with their listings and logout.

### 3.4 Entity Relationship Diagram



This shows the relationship between the entities of the Abode database, the primary keys and foreign key constraints. The diagram shows the attributes of the tables in the database.

### 3.5 Application Programming Interface (APIs)

APIs are software intermediaries that enable applications to interact with each other [8]. They allow web applications to access data from external systems.

- Google Maps Embed API – This will allow property owners/realtors to show the listings' location on google maps [3]. The property location allows potential buyers to know where the property is located and, if possible, access a street view image of the property.

# Chapter 4 – Implementation

## 4.1 Techniques and Constraints

Abode was developed as a web application to make it accessible for anybody with access to the web regardless of the device being used.

### 4.1.1 Techniques for Implementation

The technique was developed using the incremental model [5]. This is a process of software development where the system is broken down and developed into multiple standalone modules. The admin side of the web application was developed and implemented with a template. The implementation was carried out using frameworks and APIs.

### 4.1.2 Implementation Constraints

A major constraint that affected the completion of the web application was the inability to implement Sherlock. Restb.ai requires that potential clients contact them to build custom versions of Sherlock for their systems. Potential clients will have to pay for consultation and the custom version of Sherlock needed for the system. This version of Abode is going to be used for the final project presentation, and paying for consultation and a custom version of Sherlock for this version will not be a practical endeavour. In the future, when the web application is ready to be launched, then Restb.ai will be contacted, and Sherlock will be integrated into the system.

Another constraint was the database and its structure. The database was updated and edited as the project progressed. The structure was changed every time a problem was encountered with implementing a module that worked with the database. The database structure was never fixed and underwent changes throughout the implementation process.

### 4.2 System Modules

The Abode web application was broken down into the following modules:

#### 4.2.1 Sign Up

This module deals with the web application's registration processes for Clients, property owners and Realties.

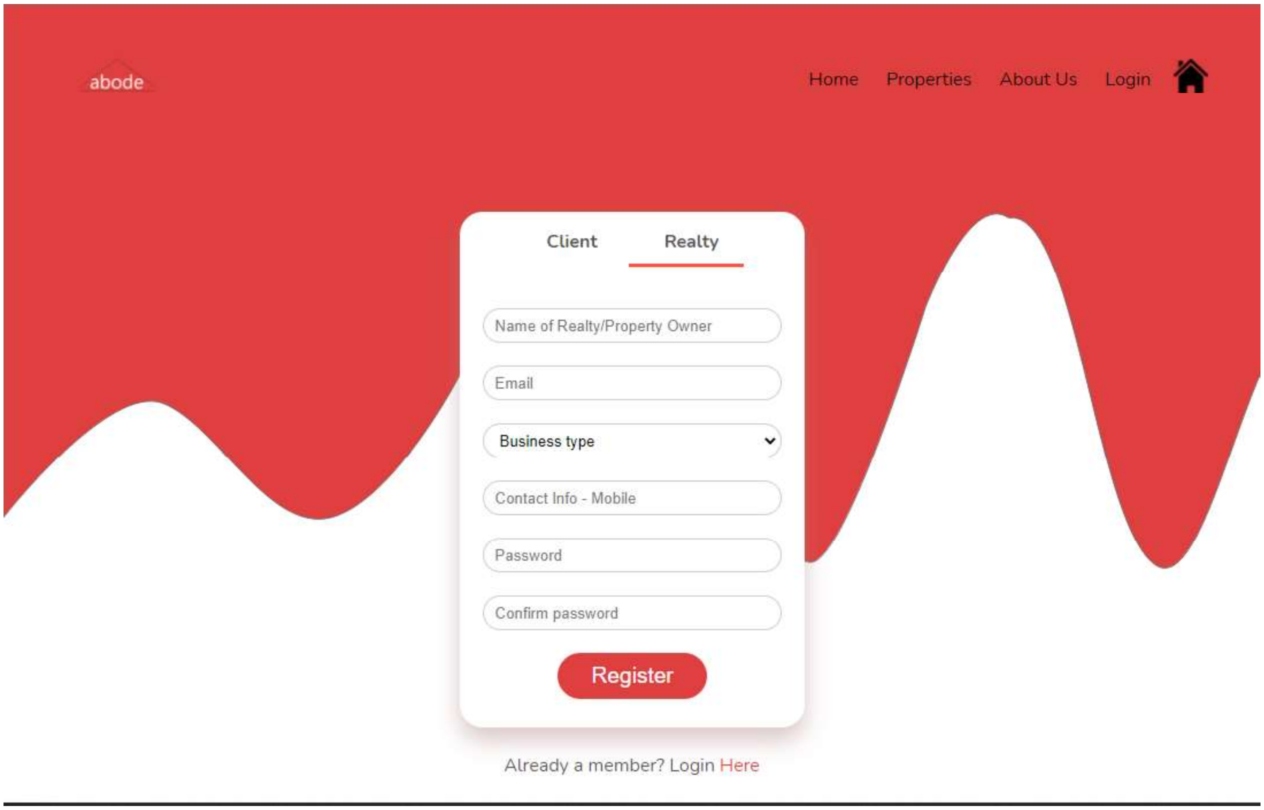


*Figure 4.2.1.1 Sign Up for Realty*

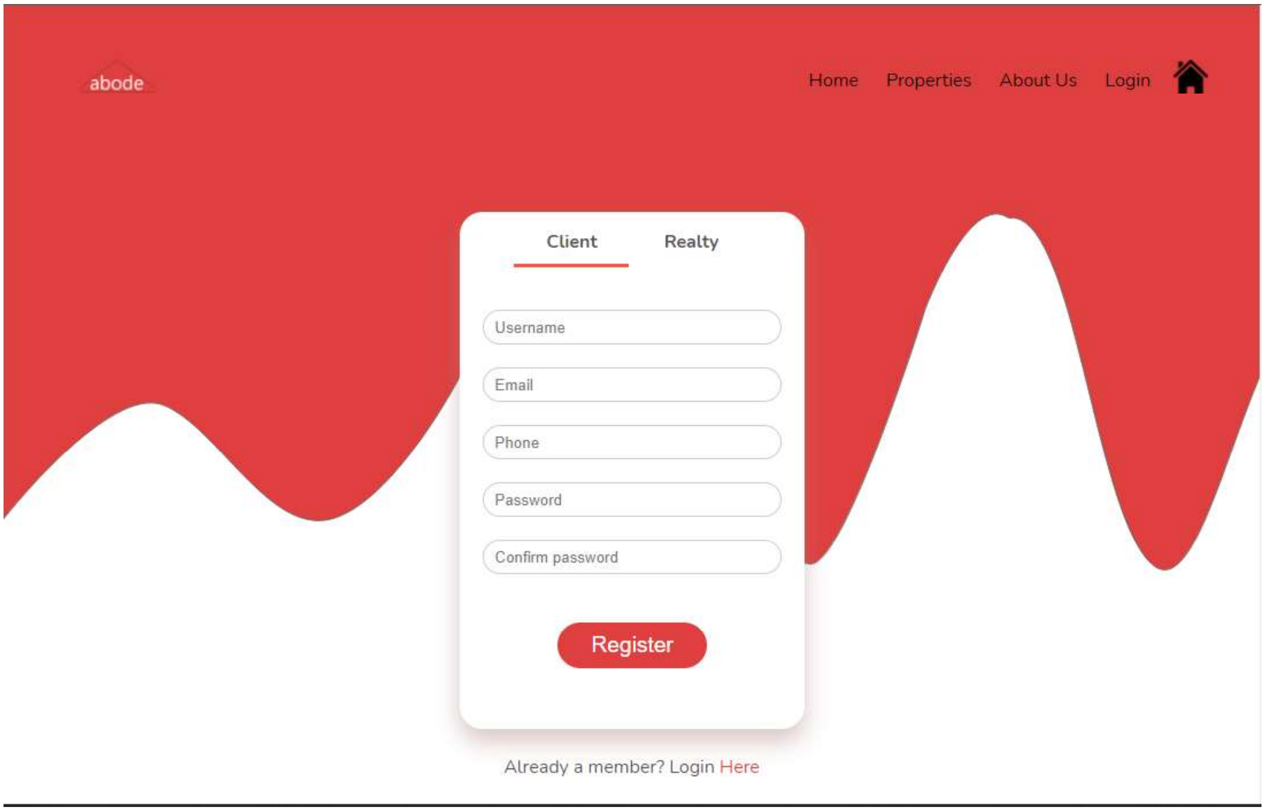


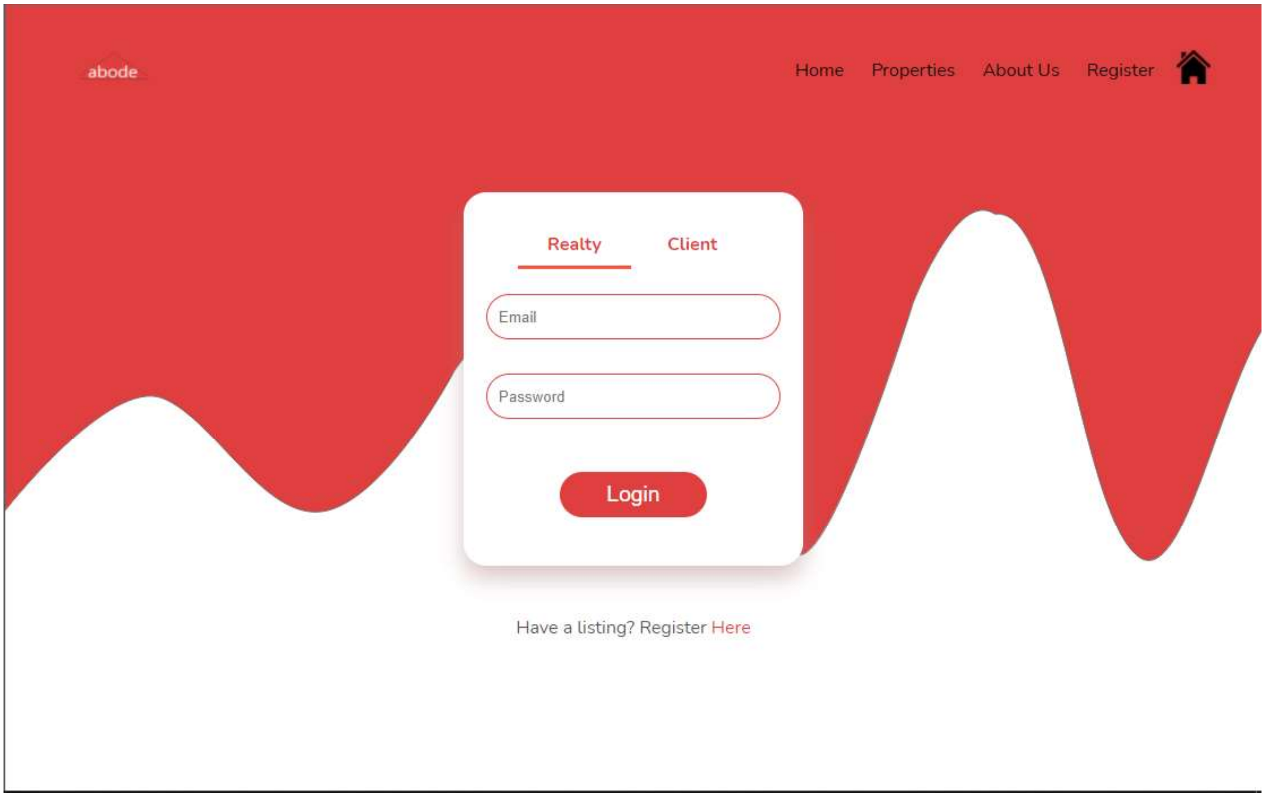
*Figure 4.2.1.2 Sign Up for Clients*

The snippet of code for client sign up is Figure 1.1in the appendix.

The snippet of code for realty and property owner sign up is Figure 1.2 in the appendix.

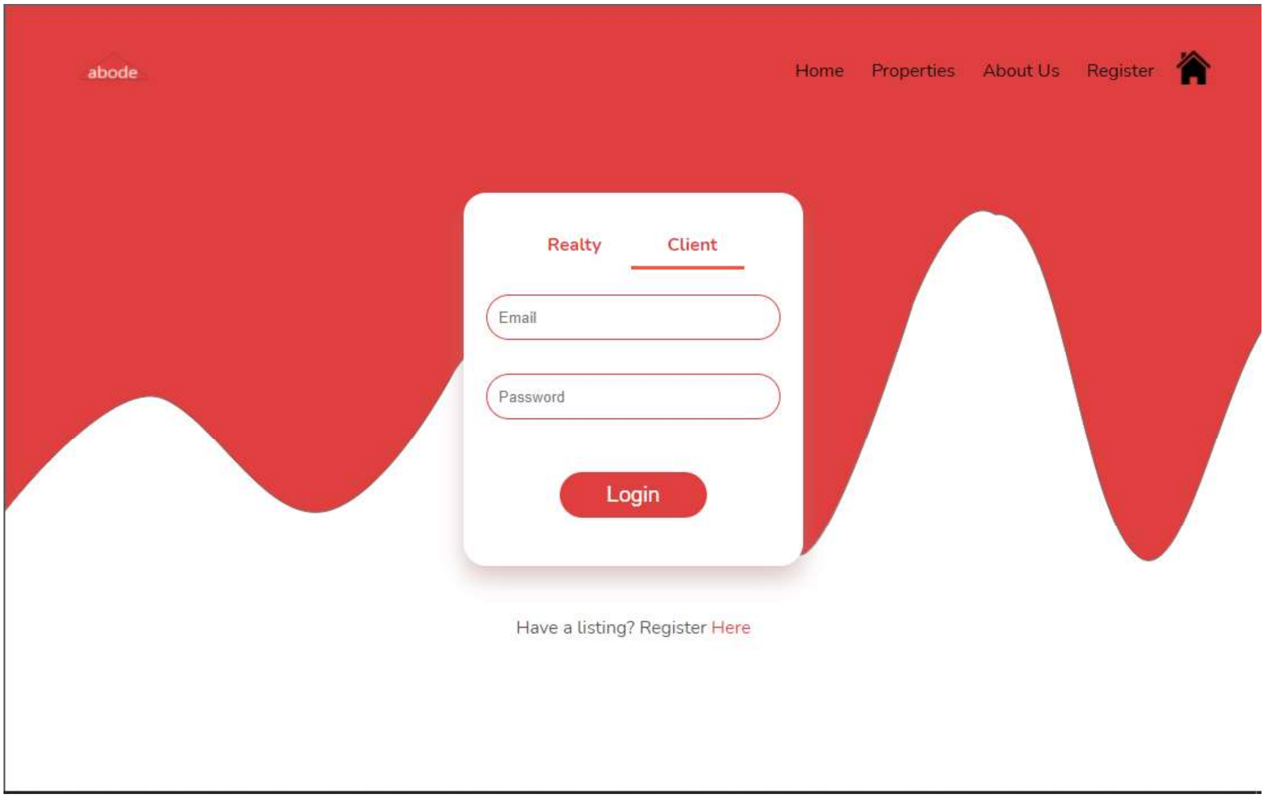
### 4.2.2 Login

This module deals with login and all login processes for all user groups.



Have a listing? Register Here

*Figure 4.2.2.1 Login for Realty*



Have a listing? Register Here

*Figure 4.2.2.2 Login for clients*

Figure 2.1 in the appendix shows the login process for clients.

Figure 2.2 in the appendix shows the login process for realties and property owners.

### 4.2.3 Property Management

This module handles all property-related processes, i.e., adding a listing, editing a listing, removing a listing, viewing a listing.

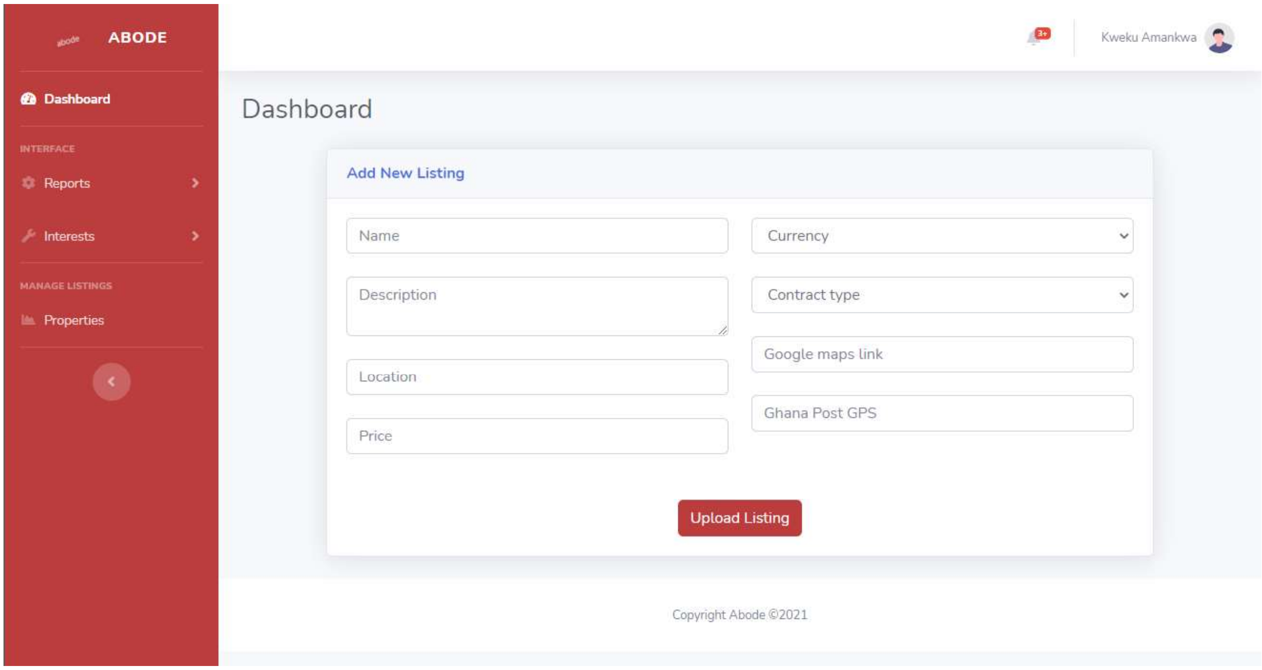


*Figure 4.2.3.1 Add Property for Property Owners*

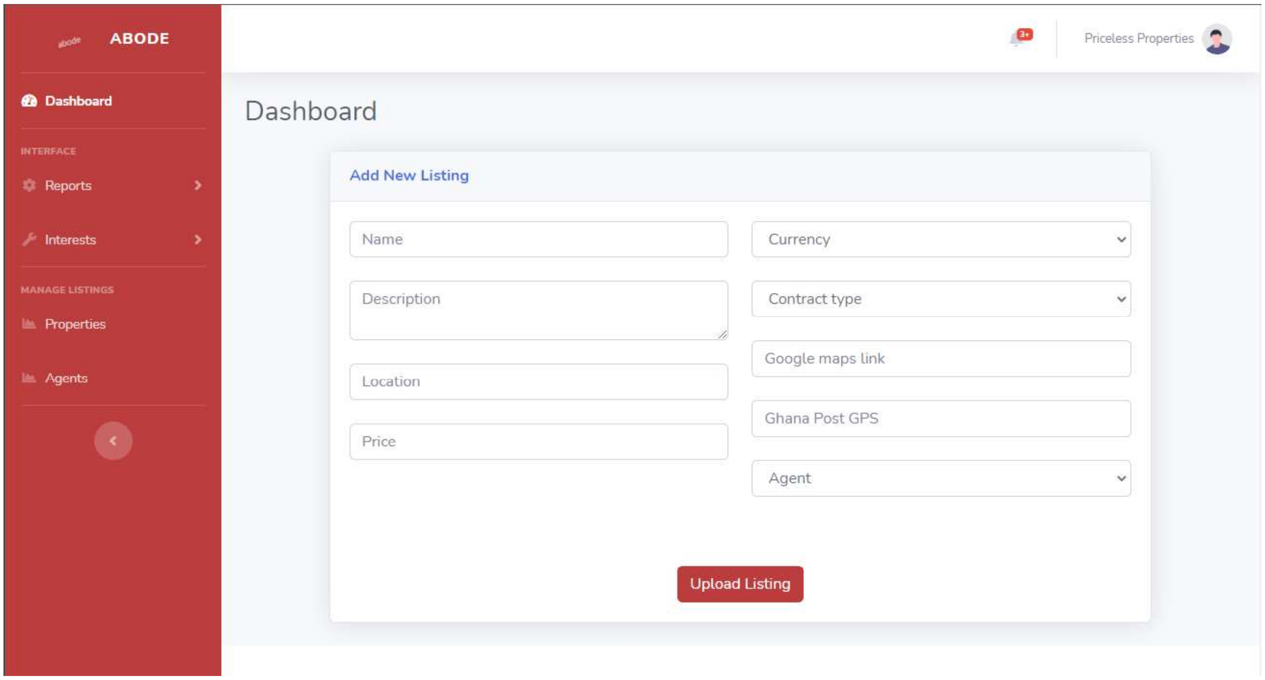
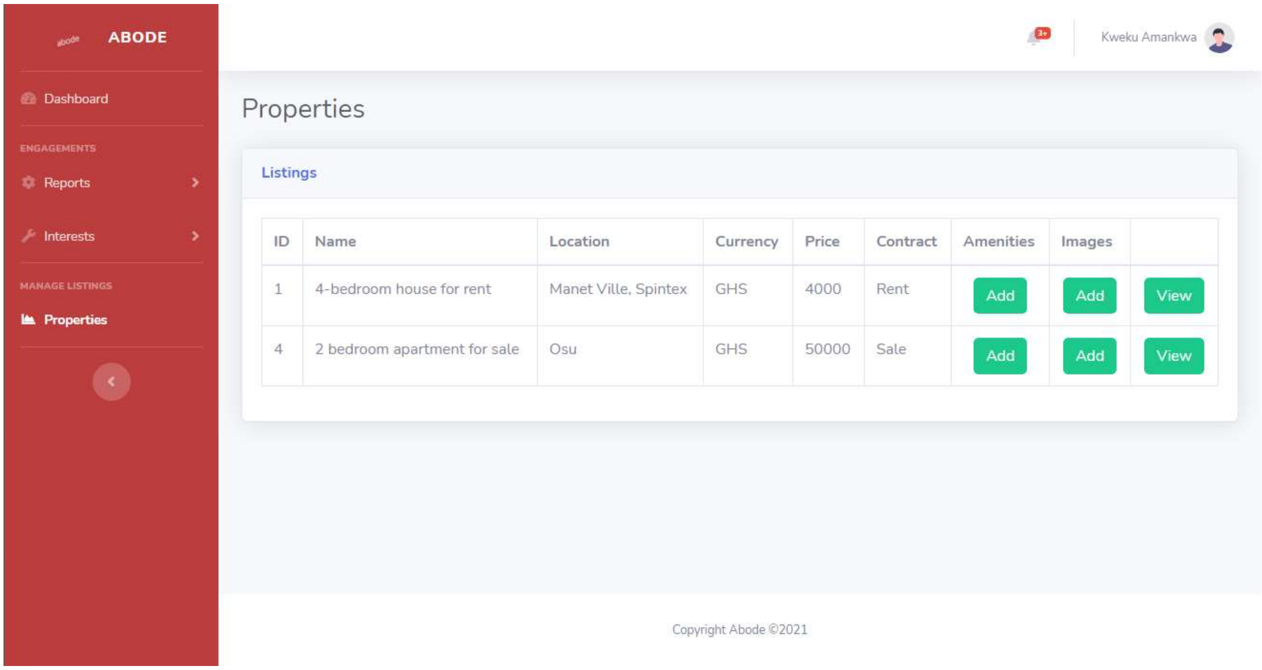


*Figure 4.2.3.2 Add Property for Realties*



| ID | Name | Location | Currency | Price | Contract | Amenities | Images | |
|---|---|---|---|---|---|---|---|---|
| 1 | 4-bedroom house for rent | Manet Ville, Spintex | GHS | 4000 | Rent | Add | Add | View |
| 4 | 2 bedroom apartment for sale | Osu | GHS | 50000 | Sale | Add | Add | View |

*Figure 4.2.3.3 View Property for Admins*

Figure 3.2 in the appendix shows the add property process for realties.

Figure 3.1 in the appendix shows the add property process for property owners.

Figure 3.3 in the appendix shows the view property code.

### 4.2.4 Agent Management

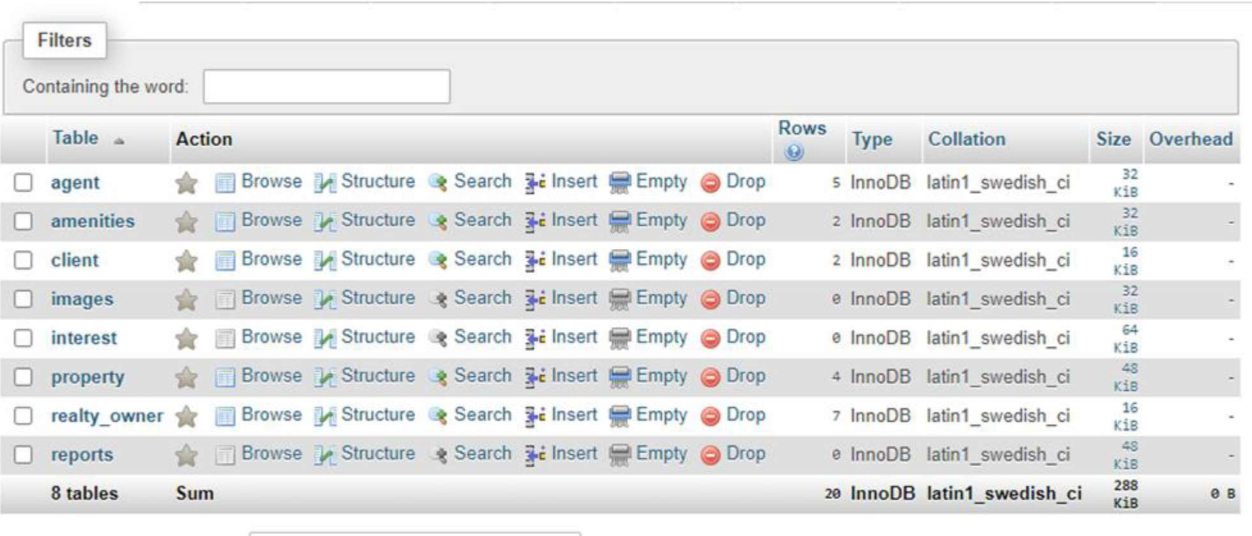
This module handles all agent-related processes, i.e., adding an agent, editing agent details, removing an agent, and assigning an agent to a property.

The picture of the code snippet for the add agent process is Figure 4.1 in the appendix.

The picture of the code snippet for the edit agent process is Figure 4.2 in the appendix.

## 4.3 Database

The database stores all system and user information. The database is a relational database because all the modules relate to each other. The database was developed using SQL.



| | Table | Action | Rows | Type | Collation | Size | Overhead |
|---|---|---|---|---|---|---|---|
| ☐ | agent | ☆ Browse Structure Search Insert Empty Drop | 5 | InnoDB | latin1_swedish_ci | 32 KiB | - |
| ☐ | amenities | ☆ Browse Structure Search Insert Empty Drop | 2 | InnoDB | latin1_swedish_ci | 32 KiB | - |
| ☐ | client | ☆ Browse Structure Search Insert Empty Drop | 2 | InnoDB | latin1_swedish_ci | 16 KiB | - |
| ☐ | images | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | latin1_swedish_ci | 32 KiB | - |
| ☐ | interest | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | latin1_swedish_ci | 64 KiB | - |
| ☐ | property | ☆ Browse Structure Search Insert Empty Drop | 4 | InnoDB | latin1_swedish_ci | 48 KiB | - |
| ☐ | realty_owner | ☆ Browse Structure Search Insert Empty Drop | 7 | InnoDB | latin1_swedish_ci | 16 KiB | - |
| ☐ | reports | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | latin1_swedish_ci | 48 KiB | - |
| | **8 tables** | **Sum** | **20** | **InnoDB** | **latin1_swedish_ci** | **288 KiB** | **0 B** |

*Figure 4.3 Abode Database*

# Chapter 5 – Testing and Results

Testing is carried out to ensure that all components and units of the system work as required and meet all user expectations. It is also conducted to help developers and users identify and fix any bugs with the system. Three types of testing were carried out: unit testing, user testing and system testing.

## 5.1 Unit Testing

Unit testing is carried out to ensure that the individual units of the system are working satisfactorily. The following are some of the important units of the system that were tested in the early development phase:

- **Account creation**

  **Pre-condition:** User data is collected using a form.

  **Expected Results**: The form successfully takes user data and inserts it into the database.

*Table 5.1 Test for account creation unit*

| Valid Input | Result |
|---|---|
| • Submit user details | Data is inserted into the database.<br>The user is redirected to the login page |
| **Invalid Input** | **Result** |
| • User details are already in the database | An alert box is displayed to inform the user.<br>The user is redirected to the register page |

- **Login**
  **Pre-condition:** User's account details are taken with a form.
  **Expected results:** Form accepts user's details.

*Table 5.2 Test for login unit*

| Valid Input | Result |
|---|---|
| • Submit user account details | User exists in the database.<br>Login is successful. |
| **Invalid Input** | **Result** |
| • User submits wrong details | An alert box is displayed to inform the user.<br>The user is redirected to the login page. |

- **Uploading a property listing**
  **Pre-Condition:** User submits property details.
  **Expected Results:** Property details are collected and inserted into the database.

*Table 5.3 Test for listing upload unit*

| **Valid Input** | **Result** |
|---|---|
| • Submit property listing details | Data is inserted into the database.<br>Successful. |
| **Invalid Input** | **Result** |
| • A field/fields in the form are left empty | An alert is displayed to inform the user.<br>The form is not submitted. |

- **Reporting a listing**
  **Pre-Condition**: User submits reason by form.
  **Expected Results**: Reason is collected and stored in the database.

*Table 5.4 Test for reporting unit*

| **Valid Input** | **Result** |
|---|---|
| • User submits reason for the report | The reason is inserted into the database.<br>The report is successful. |
| **Invalid Input** | **Result** |
| • User leaves field empty | An alert box is displayed to inform the user.<br>No report is made. |

- **Interest indication**
  **Pre-Condition:** User clicks indicate interest button.
  **Expected Results:** User successfully indicates interests.

*Table 5.5 Test for interest indication unit*

| **Valid Input** | **Result** |
|---|---|
| • User clicks indicate interest button | Property and user details are inserted into the database.<br>Interest is successfully indicated. |
| **Invalid Input** | **Result** |

| | |
|---|---|
| • User clicks “you liked this property.” | Nothing happens; the user has already indicated interest. |

### 5.2 User Testing

The system was tested by users to gain feedback on their interaction with the system in a bid to determine if the web application would be something they would use when it officially launches. To facilitate testing, the web application was hosted on a live server. A link to the web application was sent to some Ashesi students and some realtors to use and assess the system and give feedback using a google form. A total of 27 people tested the system. I was expecting a higher number, but unfortunately, not everyone who received the link filled the form.

After using the system, on a scale of 1-10, where one is “not great”, and ten is “really great”, 12 people selected 8, 8 people selected 9, and 3 people selected ten that is 44.4%, 29.6% and 11.1% respectively. Out of the 27 people, on a scale of 1-5, where one is extremely difficult, and five is extremely easy, 12 people (44.4%) selected 4, 10 people (37%) selected 5, saying that majority of users found the system easy to use.

When asked if they would use the system if it was ever launched, 21 (77.8%) said yes, and the remainder said no.

There was a section asking users for features to add, and there were only three responses asking for AR projections of the house, users paying for the house via the web application and a mobile application of the system.

Figure 5.1 in the appendix is a picture of user testing feedback form

### 5.3 System Testing

“System testing is defined as the testing of a complete and fully integrated software product.” [11] The primary purpose of this testing method was to perform a holistic evaluation of the system. The main areas were usability, performance, functional testing.

- **Usability Testing**

  This is carried out to determine how easy it is to use the system. The system made use of navigation bars and side menus with aptly described buttons and labels to enable users to navigate the system without hassle. 23 out of the 25 people who used the system found the system easy to use.

- **Performance Testing**

  This is carried out to fully determine that the web application is working as it should and satisfies all non-functional requirements: reliability, security, scalability, and maintainability. The web application was available whenever users used the link; their passwords were encrypted, users were able to use the website with no loading issues.

- **Functional Testing**

  All system functions worked as expected. Users were able to create accounts and log in properly. All property management functionalities worked without a hitch.

# Chapter 6 – Conclusions and Recommendations

This chapter covers the overview of the system, how the system satisfied requirements, recommendations and future works, limitations, and conclusion.

## 6.1 Overview of the project

Abode is a real-estate web application that was designed to help realtors and property owners advertise their property listings at no charge. The requirements for the system were divided into functional and non-functional requirements. The non-functional requirements of the system were reliability, security, scalability, and maintainability. The system's main modules were account creation, login, property management, i.e., uploading, editing and deleting listings, indicating interest in listings and reporting listings. The system was tested using unit testing, user testing and system testing.

## 6.2 Limitations

- A significant limitation was not being able to implement the Sherlock system from Restb.ai. This was supposed to deal with the content discrepancy issue that most real-estate websites struggle with.
- The pandemic made it difficult to get responses and feedback from questionnaires because we could not engage with them stakeholders physically. This also affected user testing because the people who tested the system were less than anticipated, even though the links were sent to a significantly large number of people.

## 6.3 Future Works

- The plan is to develop Abode with Sherlock fully integrated and then go ahead to officially launch it and make it available to the public.
- If the web application is as successful as we expect it to be, we are going to take it a step further and develop a mobile application to make it even more accessible and easier to use.

## 6.4 Conclusion

The web application provides users with an alternative that is easy to use and just as good, if not better, than the already existing real estate web application in the Ghanaian market. It was developed with input from a wide range of sources (Family, friends, students). Requirements were gathered from stakeholders to create the requirements statement, which guided and structured the development phase. The system includes property management features and an agent management feature for realties. Realties and property owners will not have to worry about expenses when looking for a place to advertise their property listings, and clients looking to buy houses will not have to worry about struggling with duplicates and content discrepancies.

# References

[1] Aviators http://themes.byaviators.com. 2013. About Us | SellRent Ghana. Retrieved March 31, 2021 from https://sellrentghana.com/about-us/

[2] Company | Restb.ai - Specialized Image Recognition for Real Estate. Retrieved April 26, 2021 from https://restb.ai/about/company/

[3] Get Started | Maps Embed API | Google Developers. Retrieved December 4, 2020 from https://developers.google.com/maps/documentation/embed/get-started

[4] Houses for sale, for rent or to Lease in Ghana | meQasa. Retrieved April 25, 2021 from https://meqasa.com/

[5] Incremental Model in SDLC: Use, Advantage & Disadvantage. Retrieved April 1, 2021 from https://www.guru99.com/what-is-incremental-model-in-sdlc-advantages-disadvantages.html

[6] Natalya Delcoure and Norm Miller. International Residential Real Estate Brokerage Fees and Implications for the US Brokerage Industry. 29.

[7] What is a 3-Tier Application Architecture? - Definition from WhatIs.com. Retrieved December 4, 2020 from https://searchsoftwarequality.techtarget.com/definition/3-tier-application

[8] What is an API? (Application Programming Interface). *MuleSoft*. Retrieved December 4, 2020 from https://www.mulesoft.com/resources/api/what-is-an-api

[9] What is model-view-controller (MVC)? - Definition from WhatIs.com. WhatIs.com. Retrieved April 1, 2021 from https://whatis.techtarget.com/definition/model-view-controller-MVC

[10] Visual Similarities Powered by Sherlock | Restb.ai. *Specialized Image Recognition for Real Estate*. Retrieved April 26, 2021 from https://restb.ai/solutions/visual-similarities/

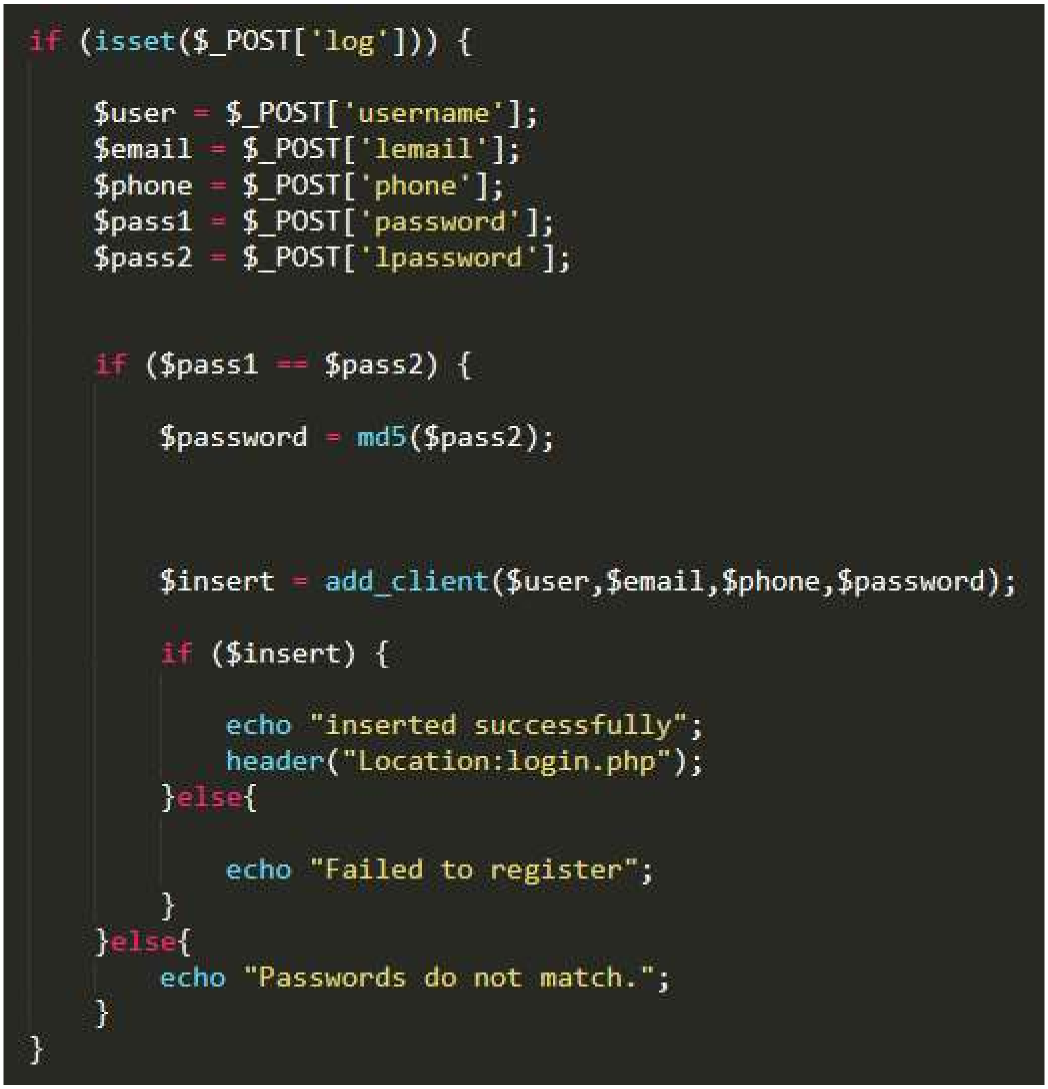
[11] What is System Testing? Definition of System Testing, System Testing Meaning - The Economic Times. Retrieved April 27, 2021 from https://economictimes.indiatimes.com/definition/system-testing

[12] Ghana Prime Properties. Real Estate Companies, Agents And Developers In Ghana | Properties For Sale. *Ghana Prime Properties*. Retrieved April 27, 2021 from https://www.ghanaprimeproperties.com/

# Appendix

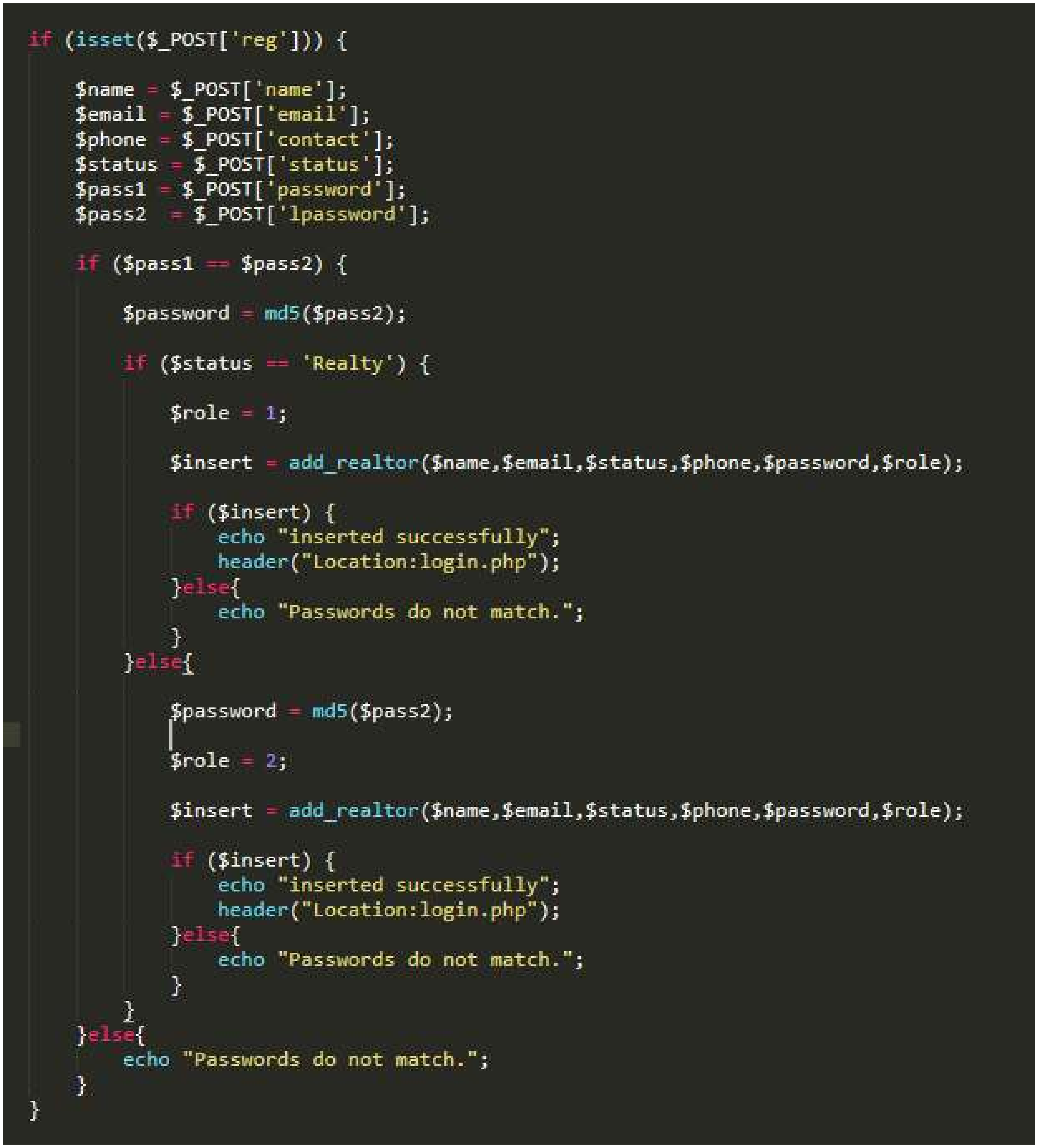
## 1.0 Sign Up

### 1.1 Client Sign Up



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

    $user = $_POST['username'];
    $email = $_POST['lemail'];
    $phone = $_POST['phone'];
    $pass1 = $_POST['password'];
    $pass2 = $_POST['lpassword'];


    if ($pass1 == $pass2) {

        $password = md5($pass2);



        $insert = add_client($user,$email,$phone,$password);

        if ($insert) {

            echo "inserted successfully";
            header("Location:login.php");
        }else{

            echo "Failed to register";
        }
    }else{
        echo "Passwords do not match.";
    }
}
```

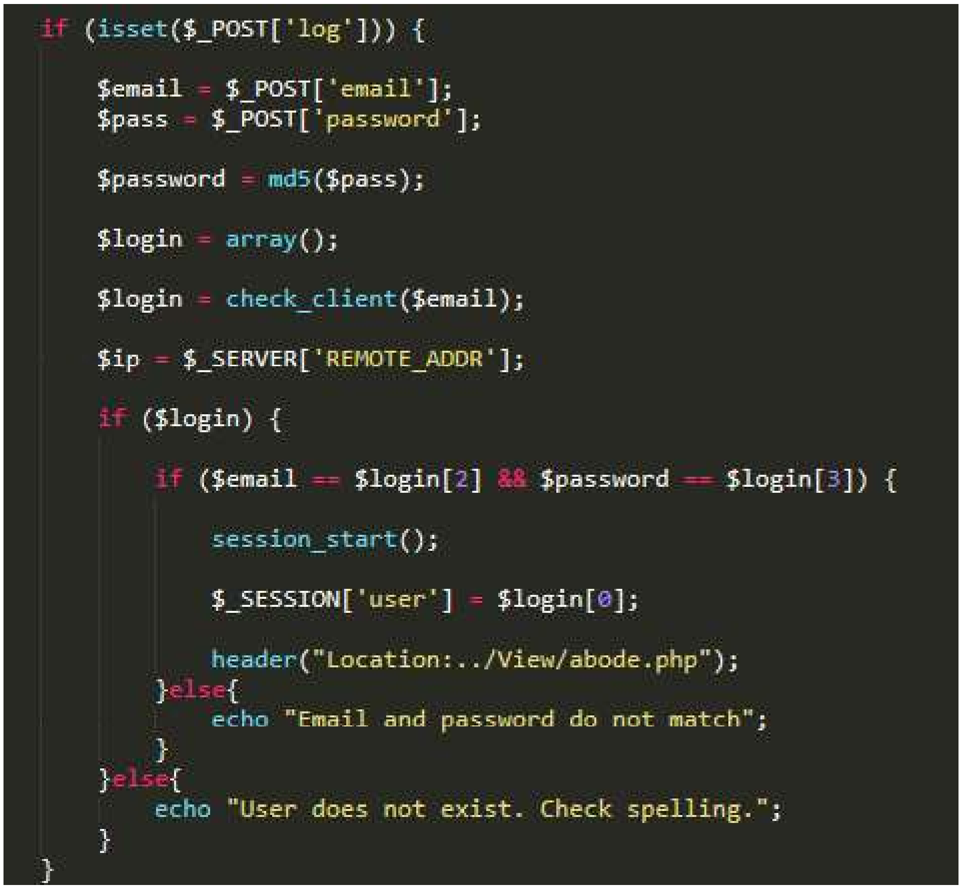
## 1.2 Realty and Property Owner Sign Up



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

    $name = $_POST['name'];
    $email = $_POST['email'];
    $phone = $_POST['contact'];
    $status = $_POST['status'];
    $pass1 = $_POST['password'];
    $pass2  = $_POST['lpassword'];

    if ($pass1 == $pass2) {

        $password = md5($pass2);

        if ($status == 'Realty') {

            $role = 1;

            $insert = add_realtor($name,$email,$status,$phone,$password,$role);

            if ($insert) {
                echo "inserted successfully";
                header("Location:login.php");
            }else{
                echo "Passwords do not match.";
            }
        }else{

            $password = md5($pass2);

            $role = 2;

            $insert = add_realtor($name,$email,$status,$phone,$password,$role);

            if ($insert) {
                echo "inserted successfully";
                header("Location:login.php");
            }else{
                echo "Passwords do not match.";
            }
        }
    }else{
        echo "Passwords do not match.";
    }
}
```

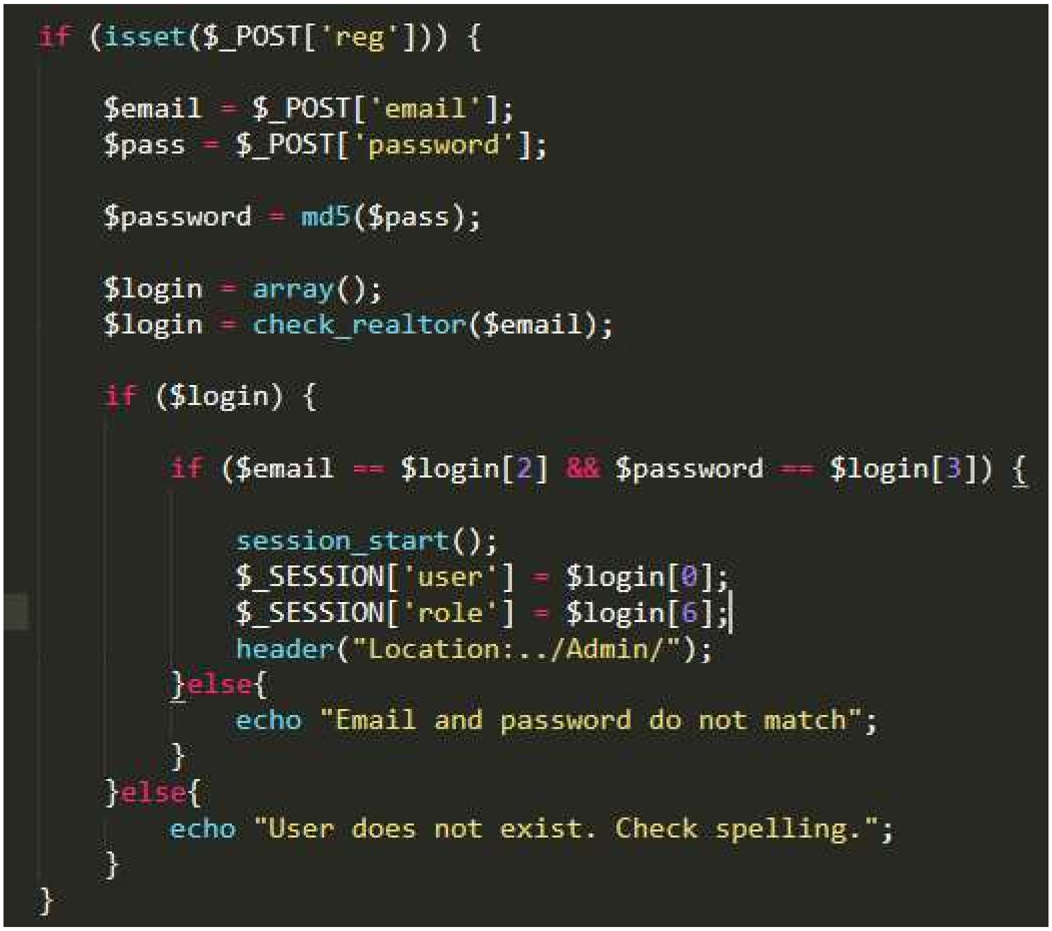
## 2.0 Login

## 2.1 Client Login



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

    $email = $_POST['email'];
    $pass = $_POST['password'];

    $password = md5($pass);

    $login = array();

    $login = check_client($email);

    $ip = $_SERVER['REMOTE_ADDR'];

    if ($login) {

        if ($email == $login[2] && $password == $login[3]) {

            session_start();

            $_SESSION['user'] = $login[0];

            header("Location:../View/abode.php");
        }else{
            echo "Email and password do not match";
        }
    }else{
        echo "User does not exist. Check spelling.";
    }
}
```

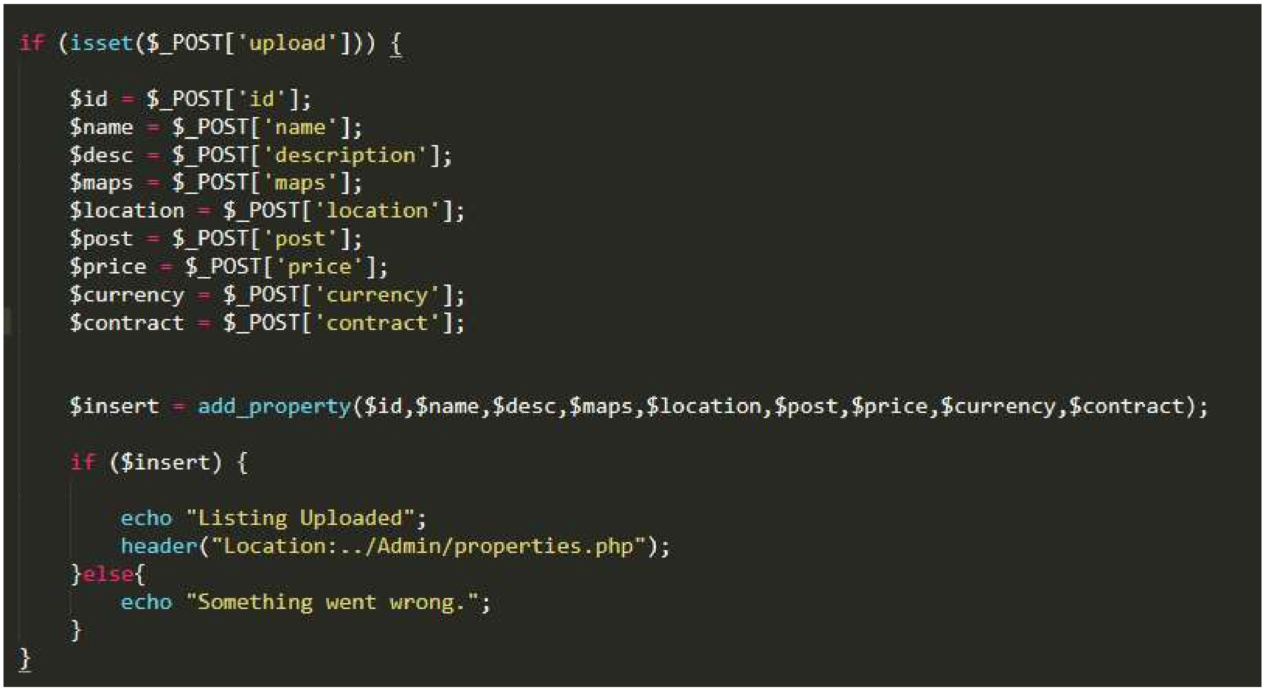
## 2.2 Realty and Property Owner Login



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

    $email = $_POST['email'];
    $pass = $_POST['password'];

    $password = md5($pass);

    $login = array();
    $login = check_realtor($email);

    if ($login) {

        if ($email == $login[2] && $password == $login[3]) {

            session_start();
            $_SESSION['user'] = $login[0];
            $_SESSION['role'] = $login[6];
            header("Location:../Admin/");
        }else{
            echo "Email and password do not match";
        }
    }else{
        echo "User does not exist. Check spelling.";
    }
}
```

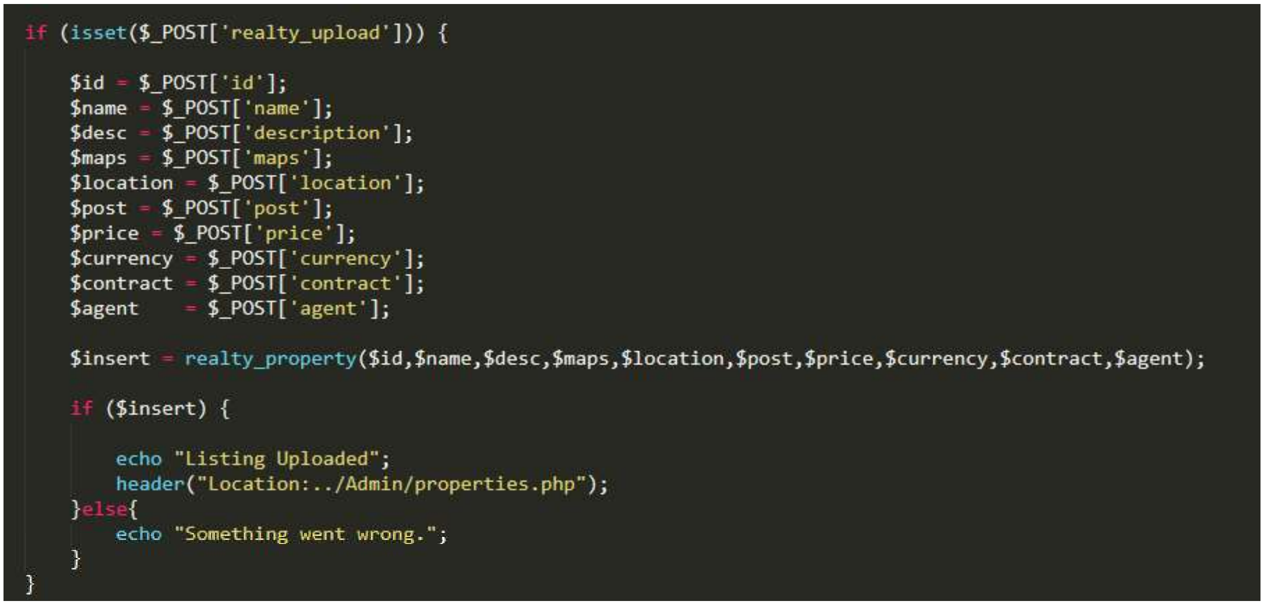
## 3.0 Property Management

### 3.1 Add Property for Property Owner



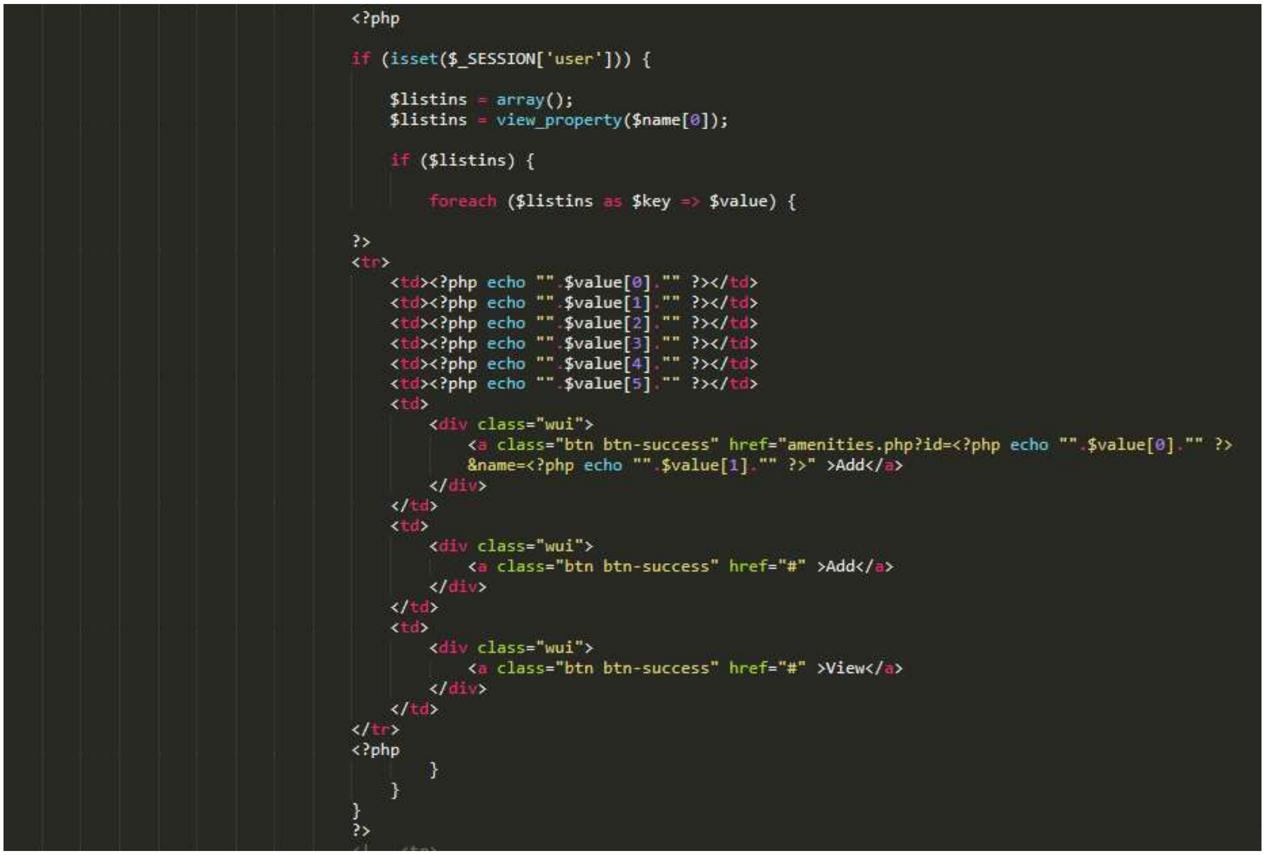
```
if (isset($_POST['upload'])) {

    $id = $_POST['id'];
    $name = $_POST['name'];
    $desc = $_POST['description'];
    $maps = $_POST['maps'];
    $location = $_POST['location'];
    $post = $_POST['post'];
    $price = $_POST['price'];
    $currency = $_POST['currency'];
    $contract = $_POST['contract'];


    $insert = add_property($id,$name,$desc,$maps,$location,$post,$price,$currency,$contract);

    if ($insert) {

        echo "Listing Uploaded";
        header("Location:../Admin/properties.php");
    }else{
        echo "Something went wrong.";
    }
}
```

### 3.2 Add Property For Realty



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

    $id = $_POST['id'];
    $name = $_POST['name'];
    $desc = $_POST['description'];
    $maps = $_POST['maps'];
    $location = $_POST['location'];
    $post = $_POST['post'];
    $price = $_POST['price'];
    $currency = $_POST['currency'];
    $contract = $_POST['contract'];
    $agent    = $_POST['agent'];

    $insert = realty_property($id,$name,$desc,$maps,$location,$post,$price,$currency,$contract,$agent);

    if ($insert) {

        echo "Listing Uploaded";
        header("Location:../Admin/properties.php");
    }else{
        echo "Something went wrong.";
    }
}
```

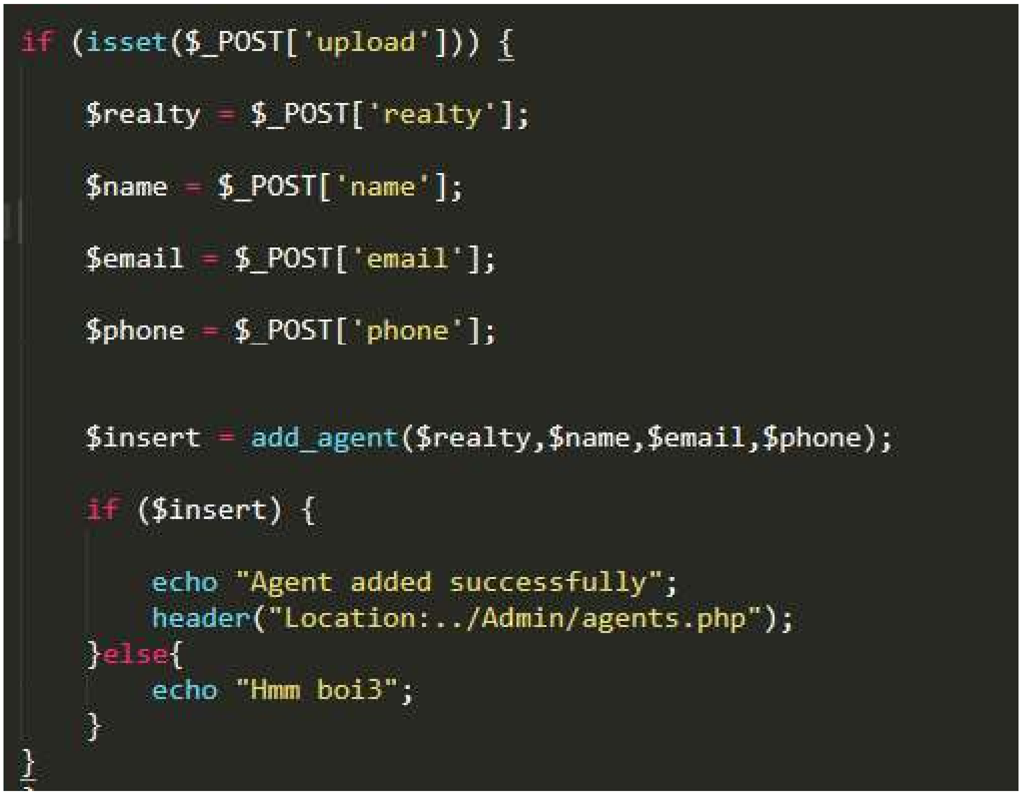
## 3.3 View Property



```
<?php

if (isset($_SESSION['user'])) {

    $listins = array();
    $listins = view_property($name[0]);

    if ($listins) {

        foreach ($listins as $key => $value) {

?>
<tr>
    <td><?php echo "".$value[0]."" ?></td>
    <td><?php echo "".$value[1]."" ?></td>
    <td><?php echo "".$value[2]."" ?></td>
    <td><?php echo "".$value[3]."" ?></td>
    <td><?php echo "".$value[4]."" ?></td>
    <td><?php echo "".$value[5]."" ?></td>
    <td>
        <div class="wui">
            <a class="btn btn-success" href="amenities.php?id=<?php echo "".$value[0]."" ?>
            &name=<?php echo "".$value[1]."" ?>" >Add</a>
        </div>
    </td>
    <td>
        <div class="wui">
            <a class="btn btn-success" href="#" >Add</a>
        </div>
    </td>
    <td>
        <div class="wui">
            <a class="btn btn-success" href="#" >View</a>
        </div>
    </td>
</tr>
<?php
        }
    }
}
?>
```

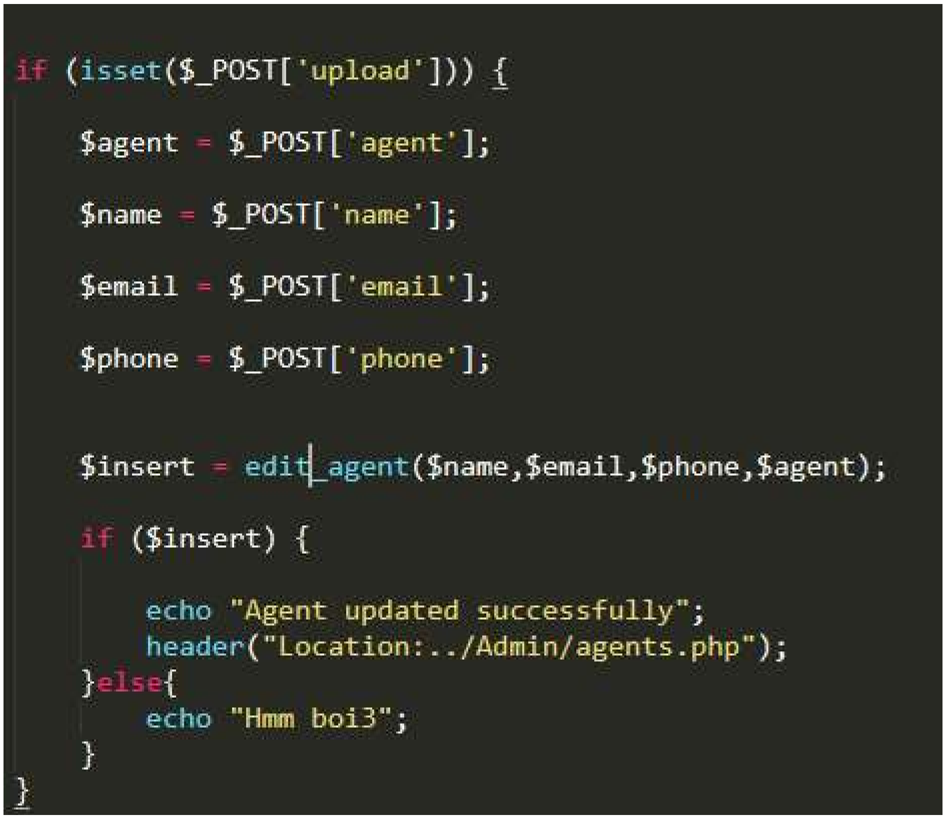
## 4.0 Agent Processes

### 4.1 Add Agent



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

    $realty = $_POST['realty'];

    $name = $_POST['name'];

    $email = $_POST['email'];

    $phone = $_POST['phone'];


    $insert = add_agent($realty,$name,$email,$phone);

    if ($insert) {

        echo "Agent added successfully";
        header("Location:../Admin/agents.php");
    }else{
        echo "Hmm boi3";
    }
}
```

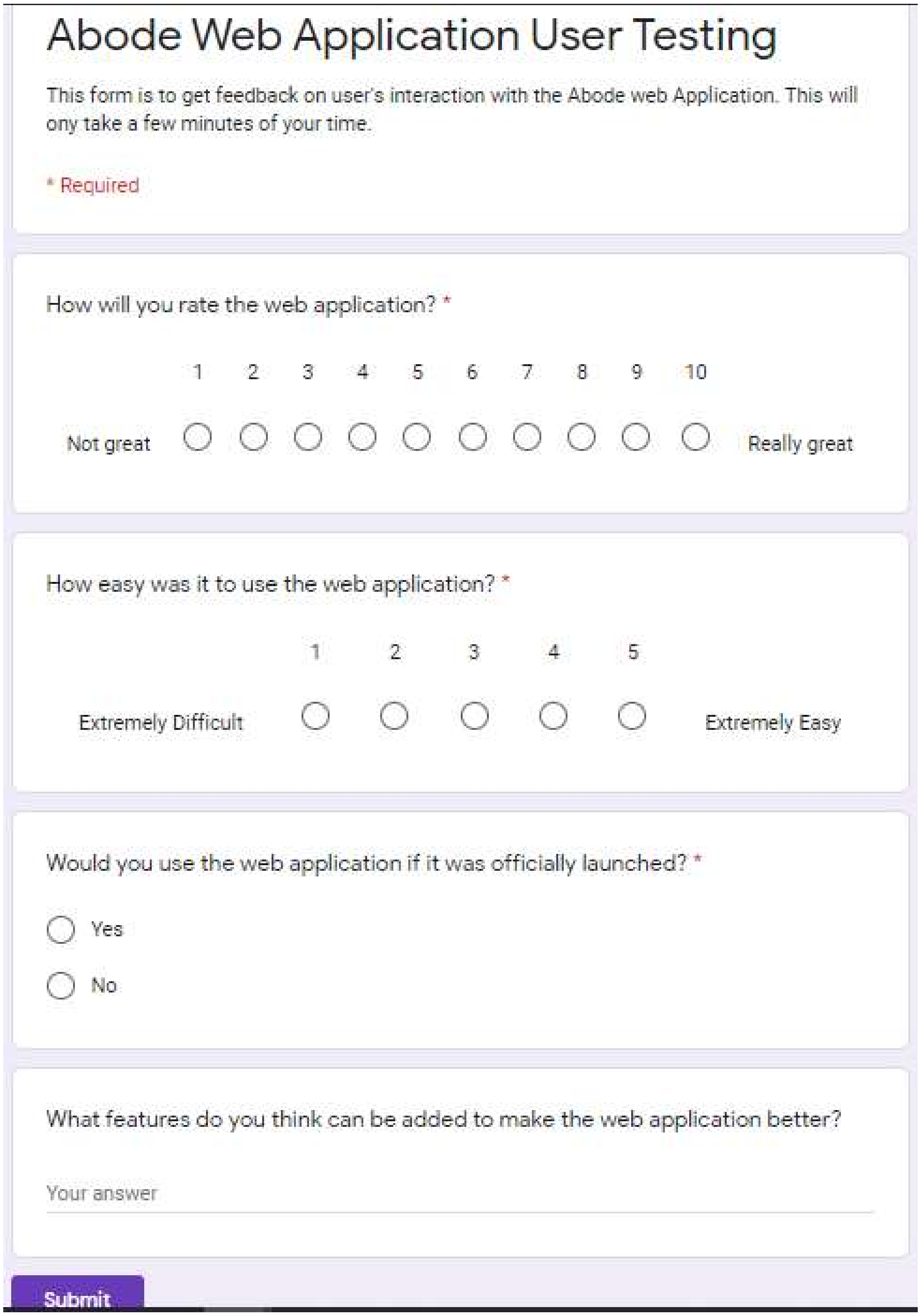
## 4.2 Edit Agent Details



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

    $agent = $_POST['agent'];

    $name = $_POST['name'];

    $email = $_POST['email'];

    $phone = $_POST['phone'];


    $insert = edit_agent($name,$email,$phone,$agent);

    if ($insert) {

        echo "Agent updated successfully";
        header("Location:../Admin/agents.php");
    }else{
        echo "Hmm boi3";
    }
}
```

## 5.0 Testing

## 5.1 User Testing Form



# Abode Web Application User Testing

This form is to get feedback on user's interaction with the Abode web Application. This will ony take a few minutes of your time.

* Required

How will you rate the web application? *

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Not great | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | Really great |

How easy was it to use the web application? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Extremely Difficult | ○ | ○ | ○ | ○ | ○ | Extremely Easy |

Would you use the web application if it was officially launched? *

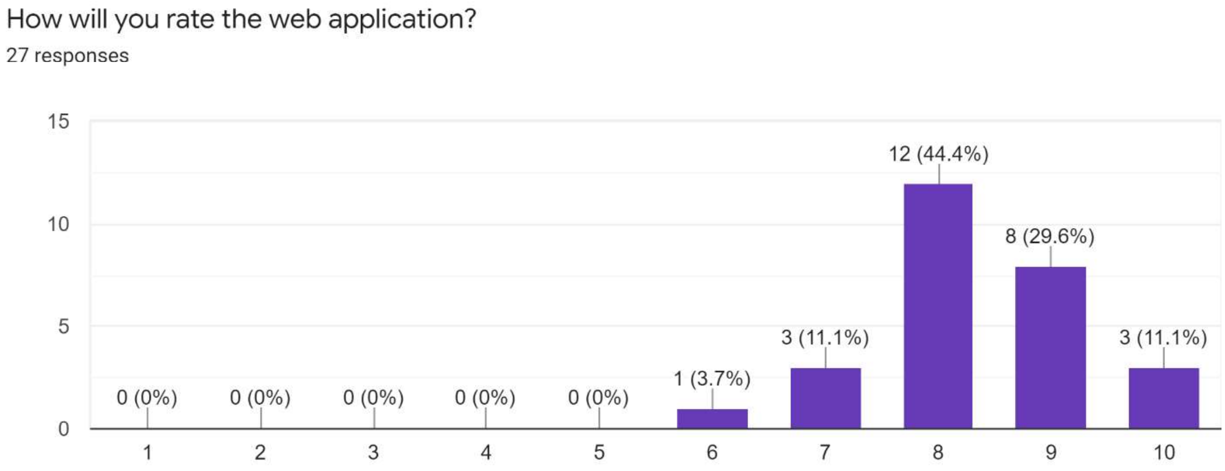
○ Yes

○ No

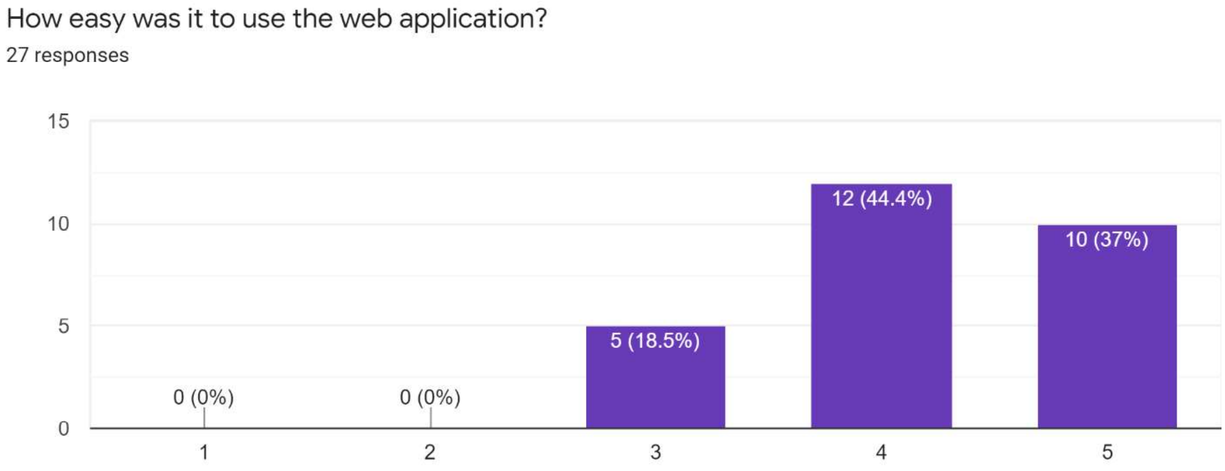
What features do you think can be added to make the web application better?

Your answer

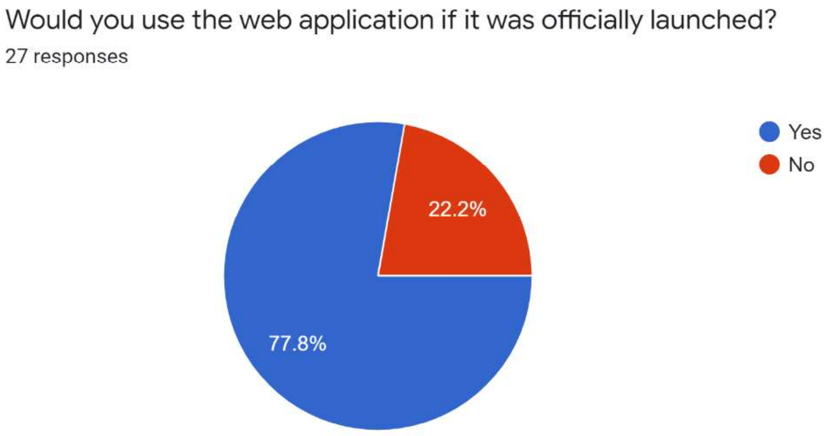
## 5.2 User Testing Results


How will you rate the web application?

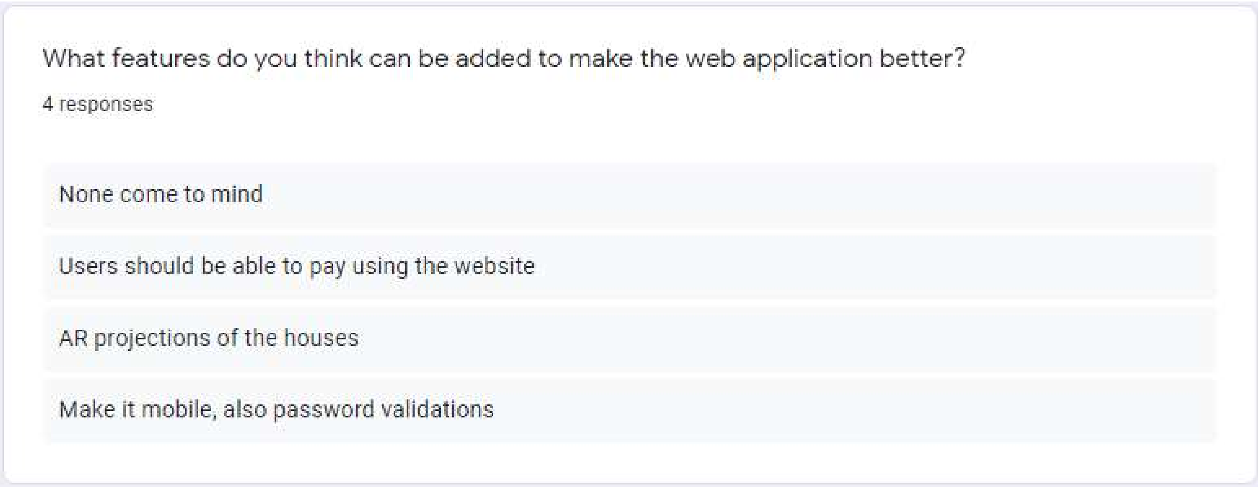
27 responses


How easy was it to use the web application?

27 responses


Would you use the web application if it was officially launched?

27 responses


What features do you think can be added to make the web application better?

4 responses

None come to mind

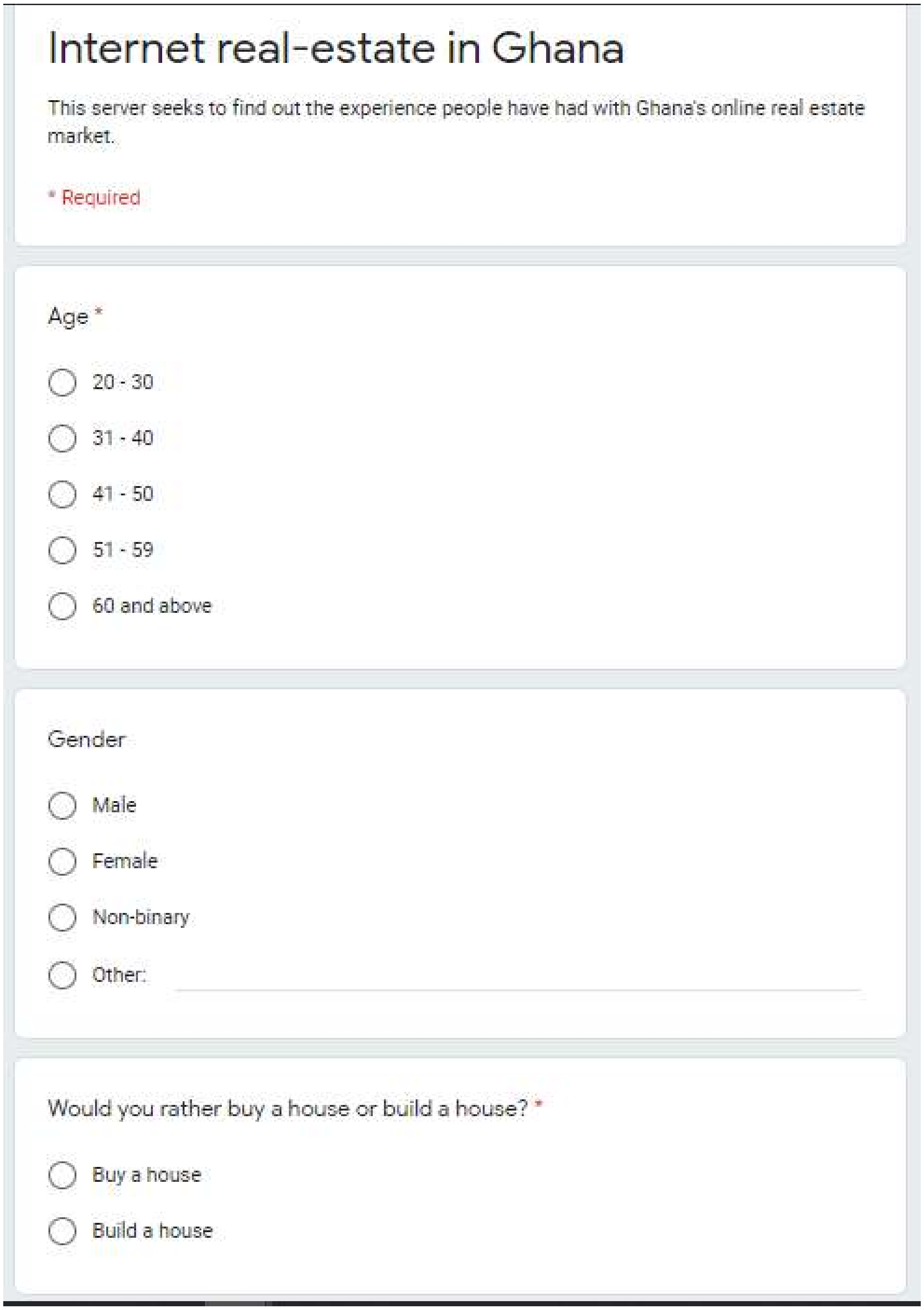
Users should be able to pay using the website

AR projections of the houses

Make it mobile, also password validations

## 6.0 Requirements

## 6.1 Requirements gathering form



Internet real-estate in Ghana

This server seeks to find out the experience people have had with Ghana's online real estate market.

* Required

Age *

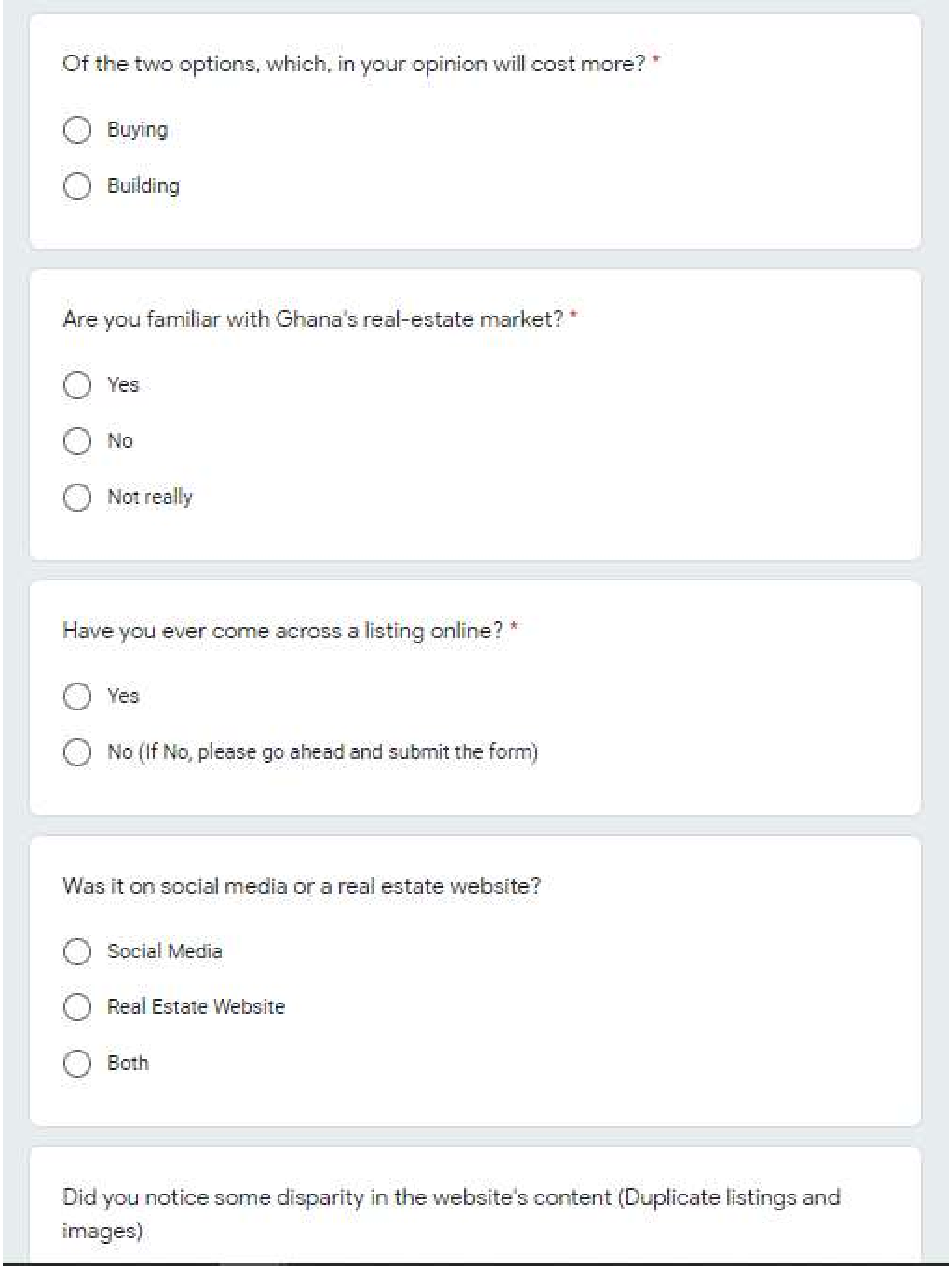
○ 20 - 30
○ 31 - 40
○ 41 - 50
○ 51 - 59
○ 60 and above

Gender

○ Male
○ Female
○ Non-binary
○ Other: ____________________

Would you rather buy a house or build a house? *

○ Buy a house
○ Build a house



Of the two options, which, in your opinion will cost more? *

○ Buying

○ Building

Are you familiar with Ghana's real-estate market? *

○ Yes

○ No

○ Not really

Have you ever come across a listing online? *

○ Yes

○ No (If No, please go ahead and submit the form)

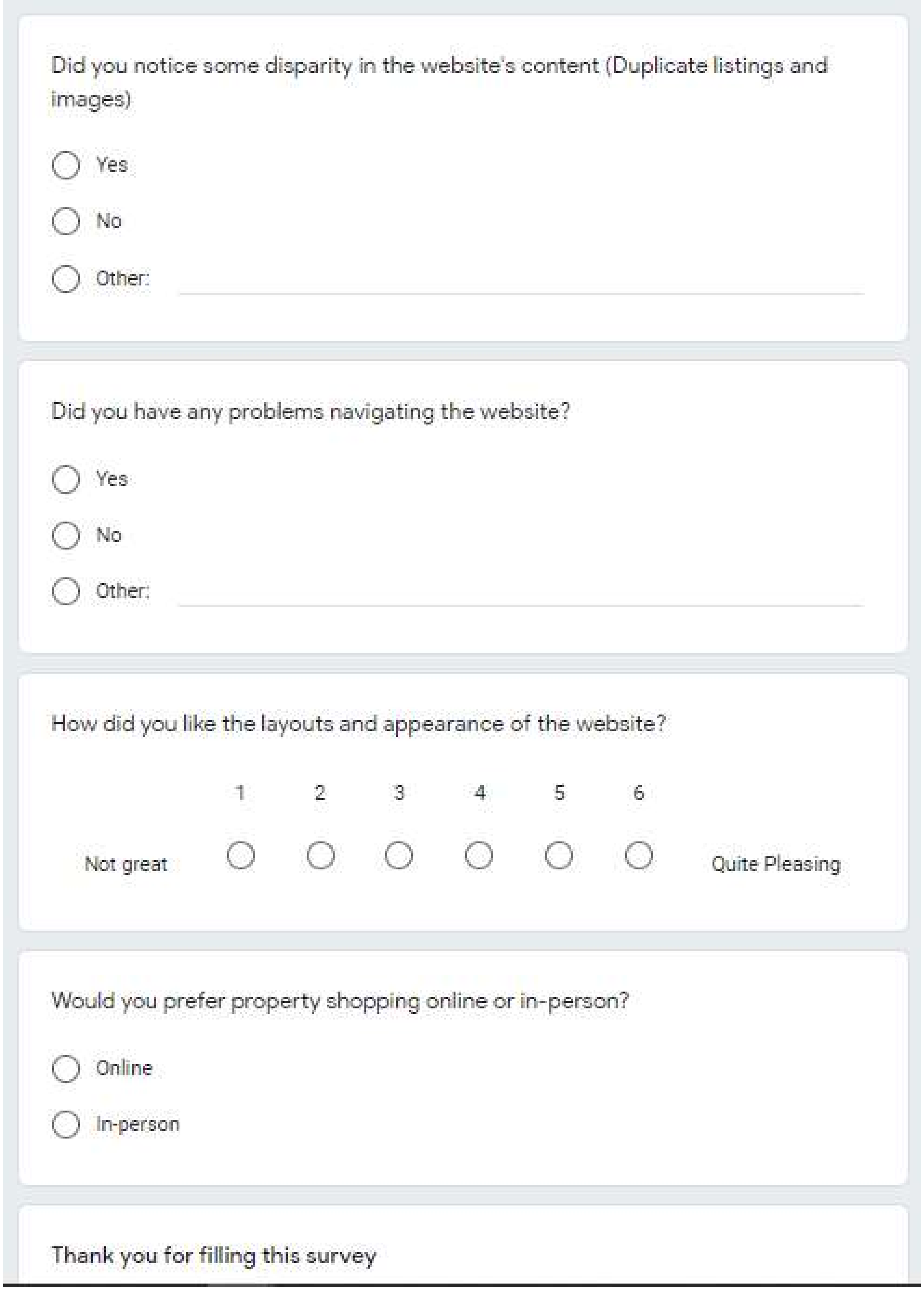
Was it on social media or a real estate website?

○ Social Media

○ Real Estate Website

○ Both

Did you notice some disparity in the website's content (Duplicate listings and images)



Did you notice some disparity in the website's content (Duplicate listings and images)

○ Yes

○ No

○ Other: ____________________

Did you have any problems navigating the website?

○ Yes

○ No

○ Other: ____________________

How did you like the layouts and appearance of the website?

|  | 1 | 2 | 3 | 4 | 5 | 6 |  |
|---|---|---|---|---|---|---|---|
| Not great | ○ | ○ | ○ | ○ | ○ | ○ | Quite Pleasing |

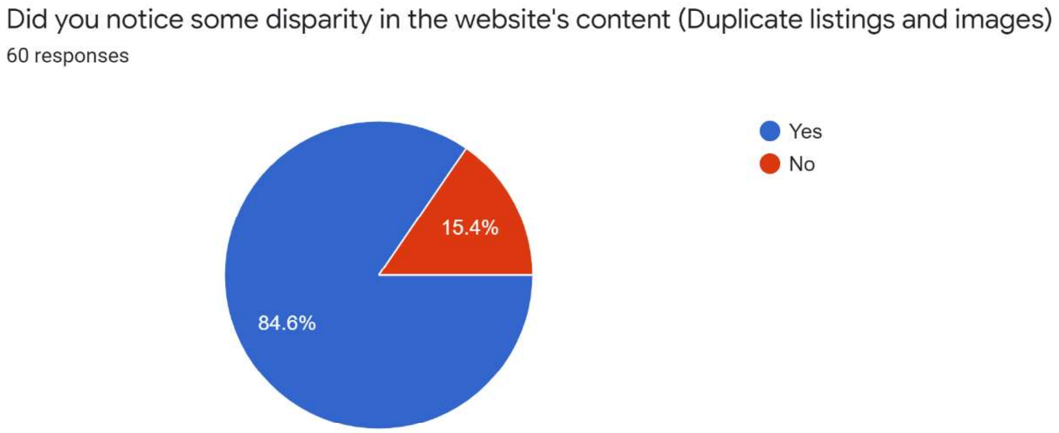
Would you prefer property shopping online or in-person?

○ Online

○ In-person

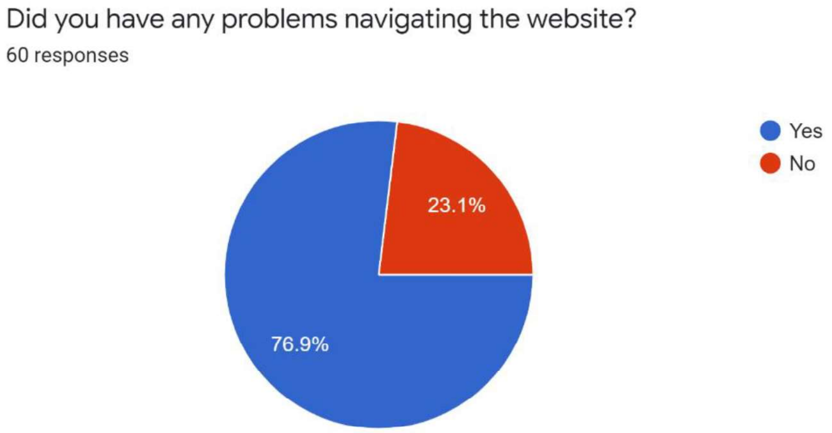
Thank you for filling this survey

## 6.2 Requirement form results



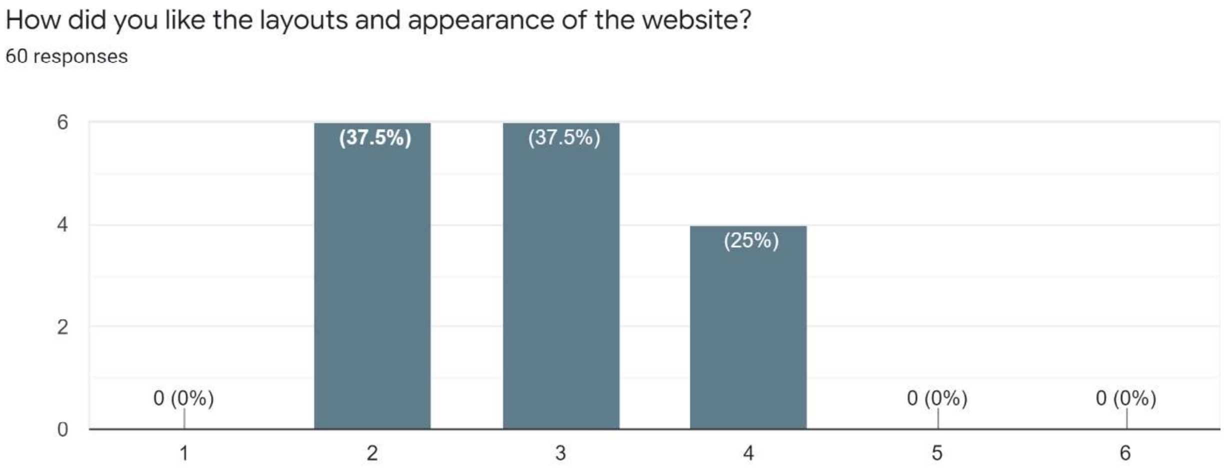
Did you notice some disparity in the website's content (Duplicate listings and images)

60 responses



Did you have any problems navigating the website?

60 responses



How did you like the layouts and appearance of the website?

60 responses