TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES

1338 Arlegui St., Quiapo, Manila

CribConnect: Mobile-Compliant Web-based Apartment Searching, Listing, and Sharing System

By

Carpio, Kristine Shyra A.

Dela Cruz, Gabriel D.

Diaz, John Mikael R.

Dida-agun, Abdul Rahman A.

Javier, Beejay B.

Villegas, Eugene D.

B.S Computer Engineering

Technological Institute of the Philippines

A Final Documentation Submitted to the Department of Computer Engineering, in Partial Fulfillment of the Requirements of Software Design Course

May2024

TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES
1338 Arlegui St., Quiapo, Manila

DEPARTMENT OF COMPUTER ENGINEERING

APPROVAL SHEET

The proposed project entitled "**CribConnect: Mobile-Compliant Web-based Apartment Searching, Listing, and Sharing System**", which was presented on **6th of May 2024** by the proponents:

Carpio, Kristine Shyra A.
Dela Cruz, Gabriel D.
Diaz, John Mikael R.
Dida-agun, Abdul Rahman A.
Javier, Beejay B.
Villegas, Eugene D.

is hereby **APPROVED** by the following members of the committee:

Mr. Joselito San Juan
Head, Panel Member

Engr. Mon Arjay F. Malbog
Project Adviser

Mr. Dennis Nava
Class Adviser

TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES
1338 Arlegui St., Quiapo, Manila

DEPARTMENT OF COMPUTER ENGINEERING

ACCEPTANCE SHEET

The project entitled "**CribConnect: Mobile-Compliant Web-based Apartment Searching, Listing, and Sharing System",** has been prepared and submitted by the proponents:

Abdul Rahman A. Dida-agun
Beejay B. Javier
Eugene D. Villegas
Gabriel D. Dela Cruz
John Mikael R. Diaz
Kristine Shyra A. Carpio

for approval to the committee for the Software Design..

After thorough review and evaluation of the proposed project, the committee has accepted the presented proposed design based on the required criteria.

The acceptance is valid to the information being presented. Accepted this **6th of May 2024, 2nd semester 2023-2024**.

Mr. Joselito San Juan
Head, Panel Member

Engr. Mon Arjay F. Malbog
Project Adviser

Mr. Dennis Nava
Class Adviser

ACKNOWLEDGEMENT

This project wouldn't have been finished without a lot of people's help. We want to thank our amazing professors. To our class adviser, Mr. Dennis Nava, MIT, and to our project adviser, Engr. Mon Arjay F. Malbog, for their incredible knowledge and advice. Their guidance made our project a reliable and successful system.

We also want to express our deepest appreciation to our families. Their love, encouragement, and support kept us going during the tough parts. Their belief in us made us stay focused and determined. We're also grateful to our friends and colleagues for their constant encouragement and help overcoming obstacles.

We would like to recognize the entire academic community and all the researchers who influenced our work. Their work laid the groundwork for our project, and we appreciate the knowledge they shared. We can't thank everyone by name, but please know that every contribution, big or small, mattered. We are truly thankful to each and every one of you.

Finally, we are incredibly grateful to God for giving us the knowledge, strength, and determination to keep going throughout this project. His presence and guidance helped us see it through to the end.

Thank you!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The process of finding suitable rental accommodations, particularly for students and young professionals, has become increasingly challenging due to factors such as rising costs and varying living conditions. This study explores the complexities of off-campus housing, emphasizing the importance of affordability, safety, and convenience. Through surveys and interviews, insights were gathered from both renters and landlords, highlighting key considerations and concerns. In response, the study proposes the development of a web-based application tailored to streamline the rental search process, prioritizing user validation, reviews, and verified listings. By providing a centralized platform, the system aims to enhance the overall experience for both tenants and landlords, promoting efficiency, resource utilization, and cost-effectiveness. The study's findings underscore the significance of such a solution in addressing the evolving needs of the rental market, particularly in urban areas like Metro Manila.

I. INTRODUCTION

Moving away from the comfort of one's home is never easy, especially when they are going to an unfamiliar place. Apart from preferences, there are more factors to consider with regards to searching for a place to rent. According to Wohletz (2013), the process of renting has changed over the years and finding a place to rent is not as cheap or easy as it used to be. Furthermore, according to the study conducted by educations.com (2021), for international students, the Philippines has one of the most affordable cost of living compared to the likes of New York, Boston, or London. However, that might not be the case for a local student as according to Wise (2023), the top five (5) most expensive cities to live in are Manila, Cebu City, Quezon City, Davao City, and Cagayan de Oro. The total living expenses per month as a student in Manila and Cebu City averages to 20,000PHP and 18,000PHP respectively. The average monthly cost of a student dorm room in Manila, Cebu City, and Quezon City are 18,000PHP, 16,000PHP, and 11,000PHP respectively which indicates that a significant portion of a student's expenses would include housing. To reduce the monthly expenses, consider seeking affordable options when it comes to accommodation like sharing a rented apartment or finding a roommate as splitting utilities and rent can significantly lower the monthly expenses. In addition, off-campus housing options are usually more affordable compared to on-campus accommodations (Little, 2023).

College students are a prime target market for apartment owners and developers. In the past few years, many cities containing a university have experienced the addition of apartments built specifically for students (Imperiale & Vanclay, 2016). With this type of surge in the market, a large portion of students will be able to be accommodated when the apartments available fit their budget. According to Oh & Kim (2021), for college students on a tight budget, house sharing has long been considered an affordable accommodation choice. Recently, house-sharing has become popular among young single-person households, including those in professional and management roles as well as those in the economically disadvantaged sector, as a way to reduce rent and increase social networks. Velasquez-Garcia & Garcia (2016) stated in their paper that since private ownership owns the majority of boarding homes and dorms in the Philippines, living conditions and facility quality are unpredictable and mostly unregulated. Accrediting private dorms and boarding houses is a standard procedure for universities in response to the growing concern over the standard of student living next to campuses. In addition, they are not endorsed by any

government bodies that oversee housing or higher education, meaning that their certification is narrowly focused and solely serves the purposes and interests of particular colleges. This might pose a problem for the safety of the students. According to an article published by Morris (2014), apartment complexes learned how to take advantage of college students with their advertisements and students who do not know any better often fall into their money trap. He advised the students through his post to put extra effort in researching before deciding where to stay rather than just looking at the price to not suffer the same experience he did. Based on a joint study conducted by Apartments.com and Google (2015), 71% of respondents agree that the internet plays an important role in searching apartments than three years ago as 67% of the respondents found their apartment through the internet, which from their perspective the most effective source in terms of searching for apartments. Searching online for an apartment is the most viable option there is, and the most convenient. While the internet proved to be an important aspect in terms of searching for an apartment, it is not as easy as it seems when factors such as limited budget and location are to be considered.

Students and apartment seekers alike face challenges when it comes to off-campus housing. The problem of poor living circumstances is one of the main issues off-campus housing tenants must deal with. The location and accessibility of off-campus housing options present another difficulty. Another matter of issue is how boarders/tenants are treated. Some students claim that they have experienced unjust treatment by boarding house or landlord owners, which has a detrimental effect on their well-being and sense of security. In addition, there are legitimate worries among students over the hygiene and condition of off-campus housing. Maintaining a healthy living space requires regular maintenance and hygienic practices (Office of Outreach and Relationships, 2024). Furthermore, the survey conducted by the researchers showed that the price is one of the major factors that students and employees consider when choosing a living space. As shown in ‘Figure 1.1 Survey Result: Factors to consider’, 90% answered price as one of the factors they consider when searching for a living space. As shown in “Figure 1.2 Survey Result: Lease’s Price Opinion”, 20% of the respondents which are currently renting a place claim that their apartment is rather expensive, and 20% of respondents which are currently seeking to rent a place answered that most of the living spaces that they encounter are expensive. On the other hand, the researcher interviewed an apartment owner near a University Campus named Marian Zaragosa regarding the problems that the apartment has to face. She indicated that as a self-managing owner, she has to be

more careful with accepting tenants because if there is any damage made in the apartments then it will become an additional operational expense.

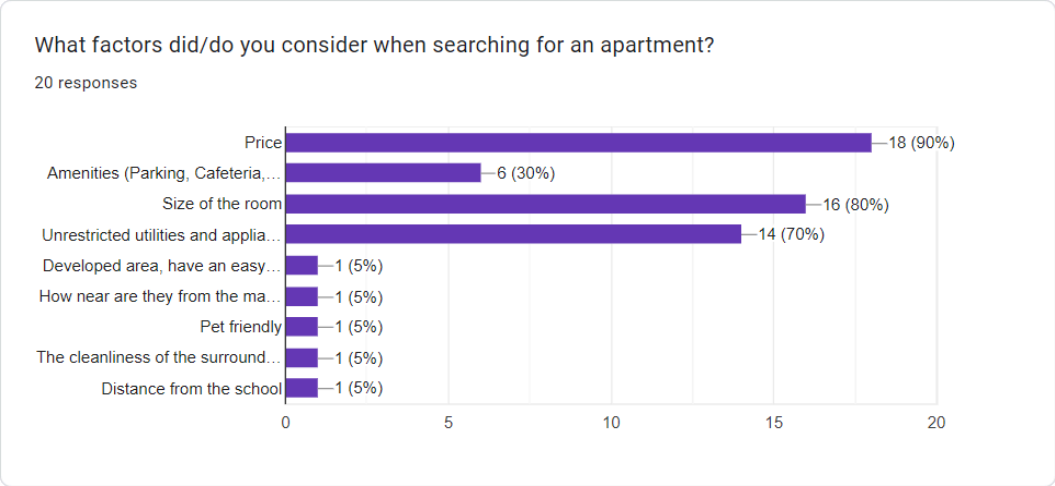


Figure 1.1 Survey Result: Factors to consider

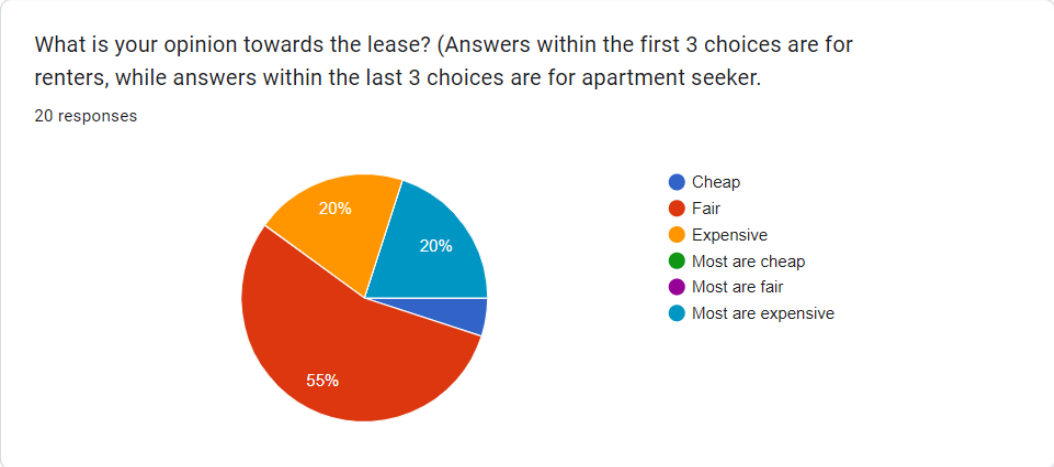


Figure 1.2 Survey Result: Lease's Price Opinion

This system aims to help the students and apartment seekers alike by creating a web-based application that can assist them in their problem of searching for a living space that fits their preferences. Specifically, the proponents plan to create a web-based application that focuses on profiling property searchers using validation and reviews, enabling increased security and reliability for both the tenants and the landowners. In addition, create a seamless property listing and inquiry request to both property searcher and owner which is limited to verified users. On the searchers' end, the team plans to provide a

simple layout of the listed properties that prioritizes only the necessary information based on the factors that the respondents of the survey primarily consider.

This study considered both aspects with regards to renting, where the team took a survey for the searcher and an interview for the landholder. The survey took both male and female respondents ranging from 18 to 30 years of age that are currently living in a rented property within Metro Manila. The team managed to accumulate 20 respondents in total. This survey intends to assess the experience and opinion of the respondents in searching for an apartment to rent. On the other hand, the team interviewed an apartment owner nearby Polytechnic University of the Philippines - Parañaque, this interview intends to resolve a problem that the interviewee raised with regards to accommodating apartment searchers online. The information that is gathered will be used by the team to create a web-based listing application that can enhance the long-term lodging experience of both tenants and landlords through a rating and review system, while providing the searcher a seamless experience in terms of searching a property to rent. Furthermore, the system is limited to rooms, apartments, and dorms which are for long term use, therefore any short term listings such as hotels are subject to be removed.

This study holds significant advantages for both landlords and tenants alike. It offers convenience by providing a centralized platform for listing, searching, and checking about available apartments, accessible anytime and anywhere. By encouraging apartment sharing and roommate matching, the system promotes efficient resource utilization, reducing housing expenses and minimizing environmental impact by decreasing the demand for new construction. This study is also significant to the students that will be doing a system related to the study. To further prove the value of this study, the result of the survey that the proponents conducted are shown below. The proponents accumulated a total of 20 responses; 35% are employees and the rest are students as shown in ‘Figure 1.3 Survey Result: Respondent’s Employment Status’. In ‘Figure 1.7 Survey Result: Reason for renting’, it shows that to relocate near school is the main reason that they look for a living space. Among them, 60% already found a living space while the majority of the remaining percentage have been looking for more than a month now as shown in ‘Figure 1.4 Survey Result: How long have you been searching’. Out of the 60% that already found a place to stay, the majority of them took a few weeks to a month to find a suitable place to rent as shown in ‘Figure 1.5 Survey Result: How long did it take’. And as for the mean they use to look for an apartment, 50% of the respondents answered that they base it on the recommendation of their families and/or friends, and 35% use social media apps to search as shown in ‘Figure 1.6 Survey Result: How did you find your apartment’ below.

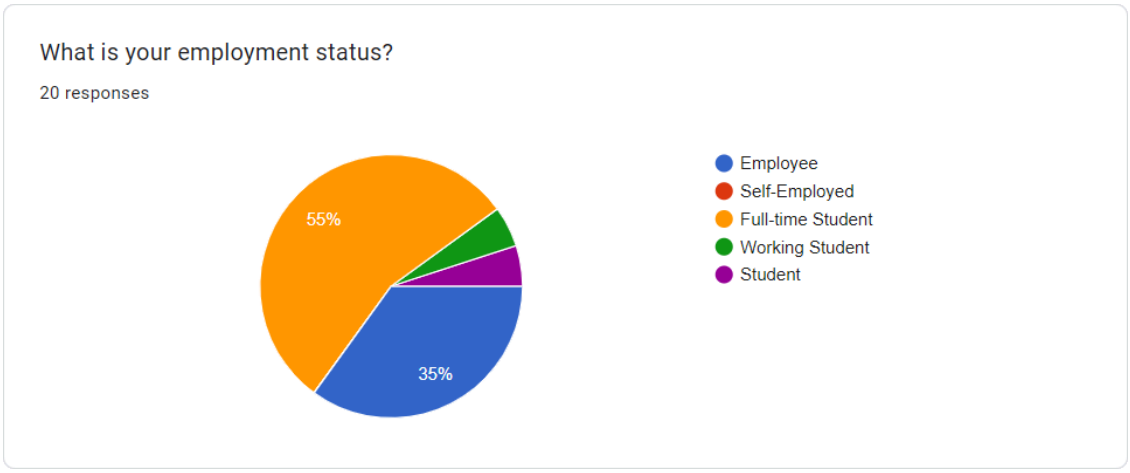


Figure 1.3 Survey Result: Respondent's Employment Status

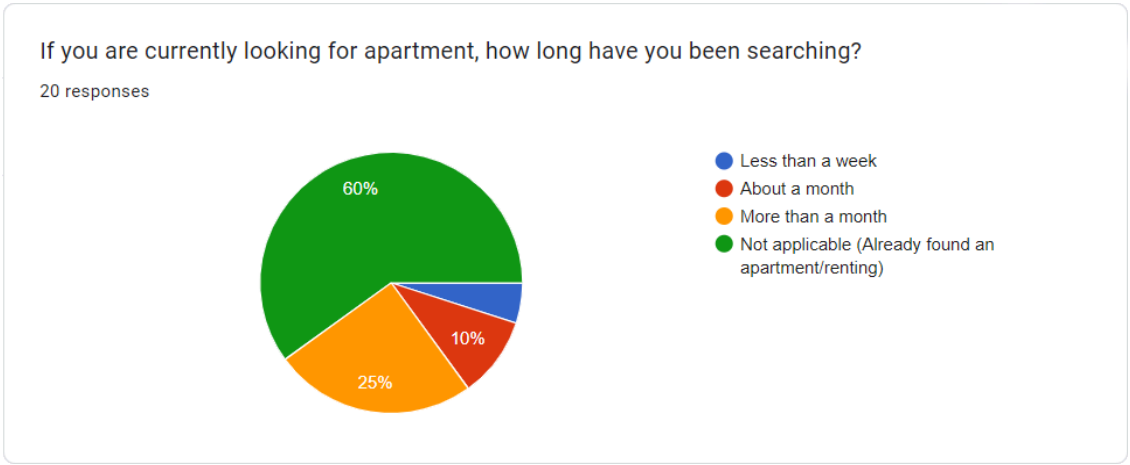


Figure 1.4 Survey Result: How long have you been searching

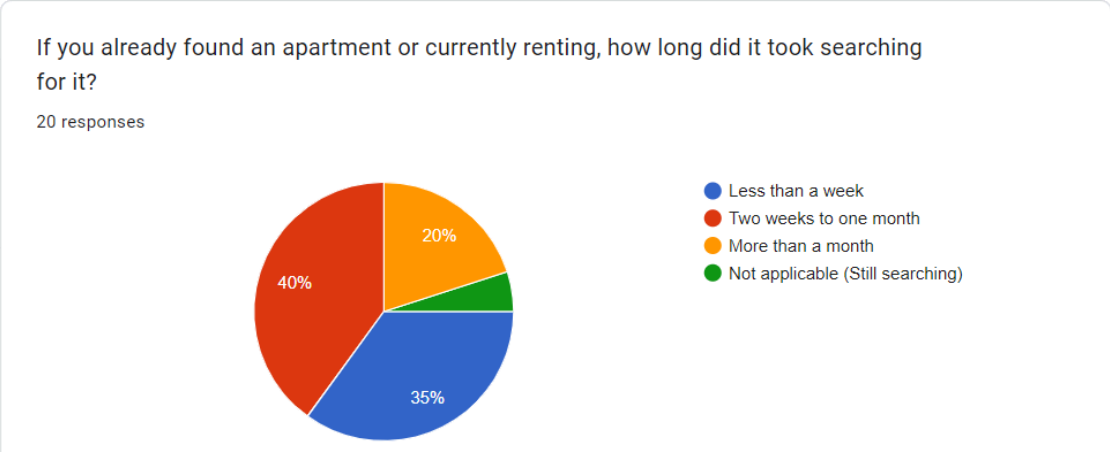


Figure 1.5 Survey Result: How long did it take

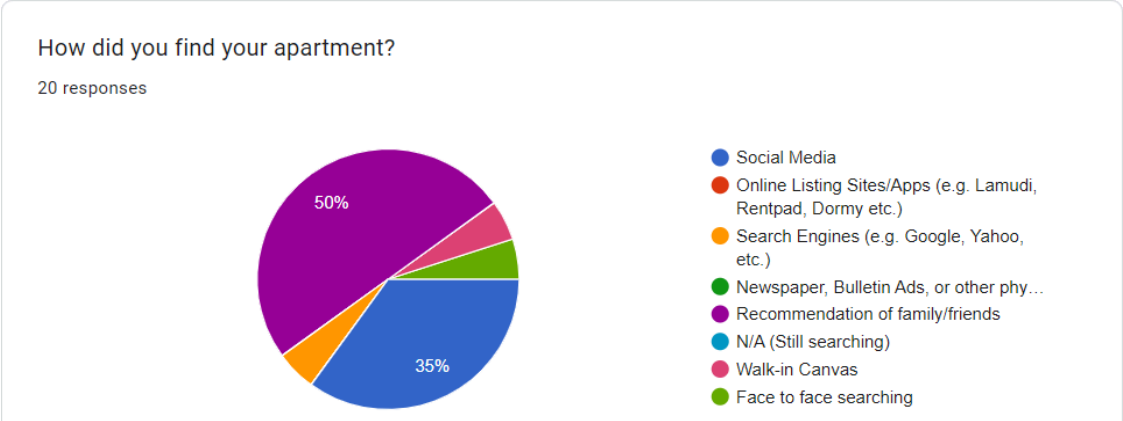


Figure 1.6 Survey Result: How did you find your apartment

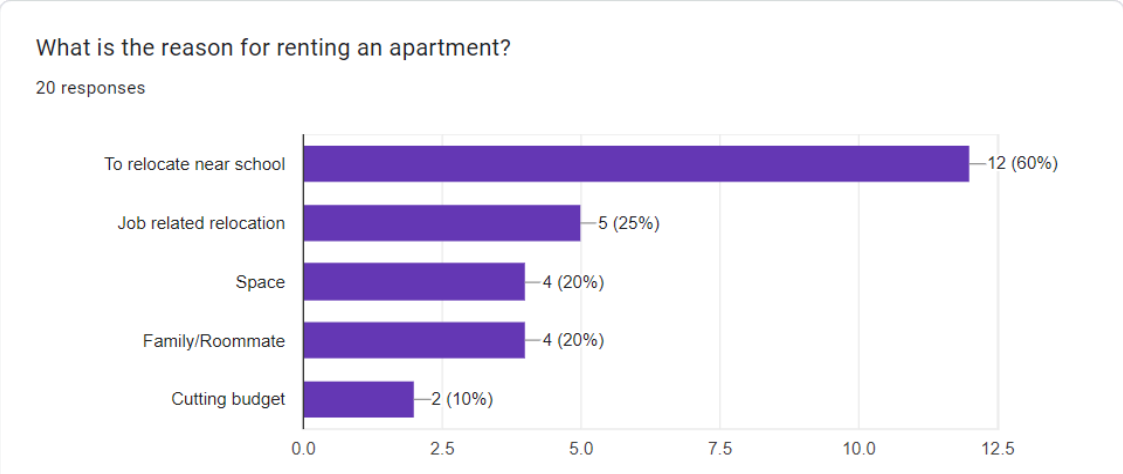


Figure 1.7 Survey Result: Reason for renting

II. SOFTWARE NEEDS VALIDATION

As a metropolitan area is expanding and developing, the demand for more reasonably priced housing is increasing (Abad et al., 2016). According to McKee (2023), renting an apartment in the Philippines can be a great choice, but only if you make the effort to choose the ideal residence for your needs and budget. To avoid any unpleasant surprises, don't forget to do extensive research, evaluate costs, and thoroughly read the lease agreement.

Over time, renting an apartment online has been easier and easier. A database of apartments available for rent is provided by various real estate firms and agencies on their websites. You can also reserve your apartment online, read other people's reviews, and view images of the properties (nyhabitat.com, 2013).

There are several existing real estate websites in the Philippines made specifically to cater buying, selling, and renting properties. Students or apartment searchers can use these websites to look for living space that they can rent with their preference in mind, however, these web apps do not offer the option to look for roommates to share the cost with. Examples are listed below.

Lamudi Philippines

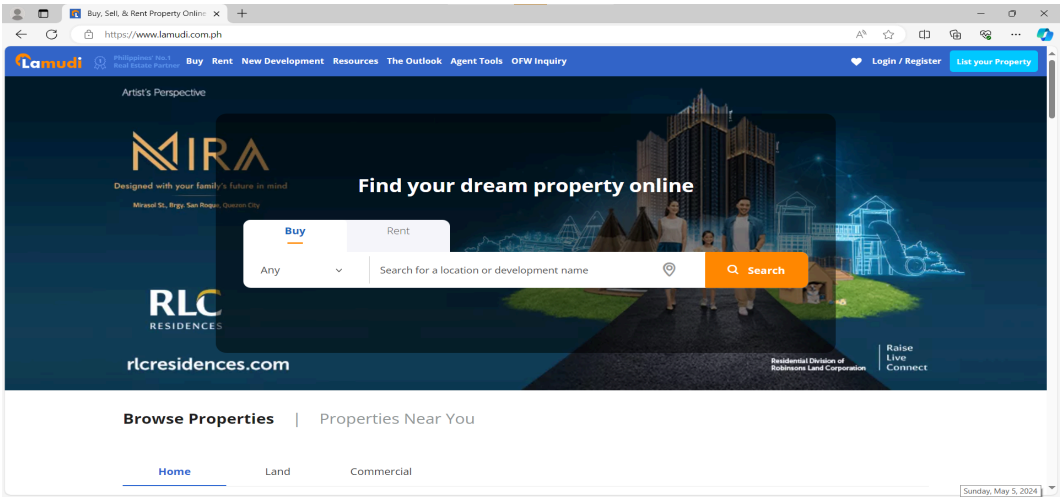


Figure 2.1 Lamudi homepage

Lamudi is a global real estate platform that operates in several countries, including the Philippines. It offers listings for residential and commercial properties, as well as land for sale or rent.

Dot Property Philippines

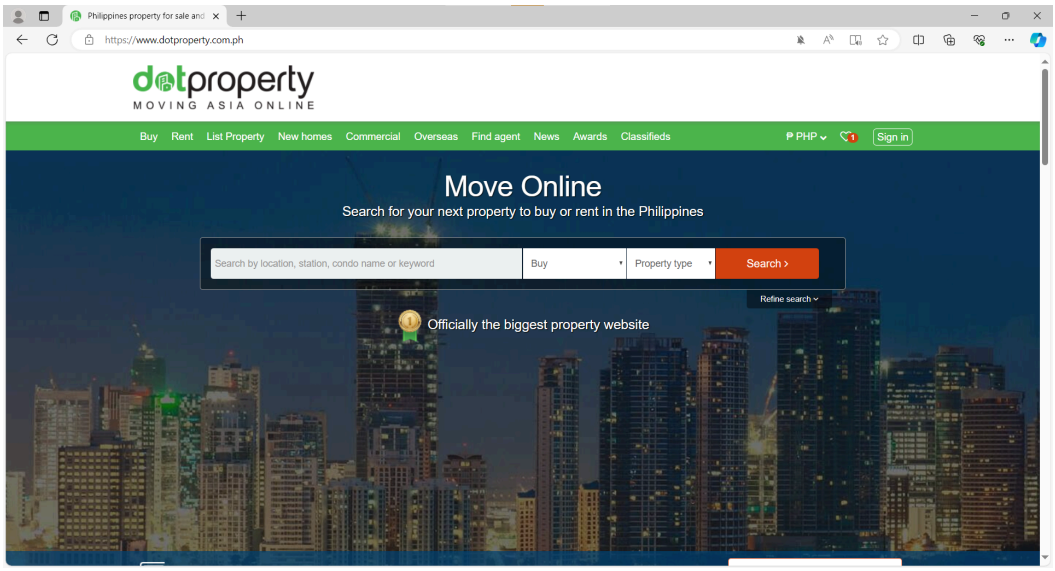


Figure 2.2 Dot Property homepage

Dot Property is a real estate marketplace that offers listings for residential, commercial, and industrial properties in the Philippines. It also provides news and insights about the real estate market in the country.

MyProperty Philippines

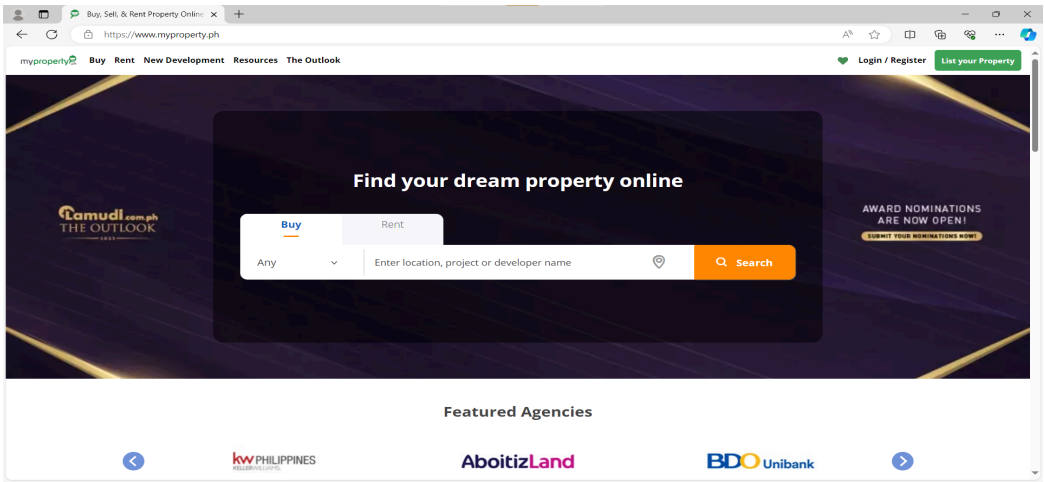


Figure 2.3 MyProperty homepage

MyProperty is a real estate platform that features listings for residential and commercial properties, as well as land for sale or rent. It also offers tools and resources for property buyers and sellers.

Rentpad Philippines

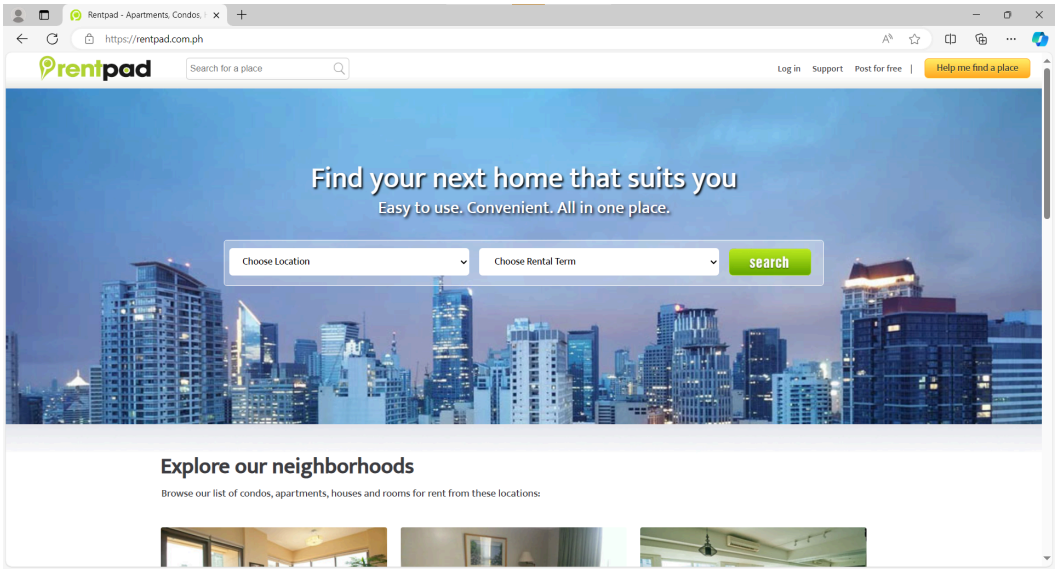


Figure 2.4 Rentpad homepage

Rentpad is a real estate platform in the Philippines that focuses on rental properties. It allows users to search for apartments, condos, houses, and other types of properties that are available for rent.

Carousell Philippines

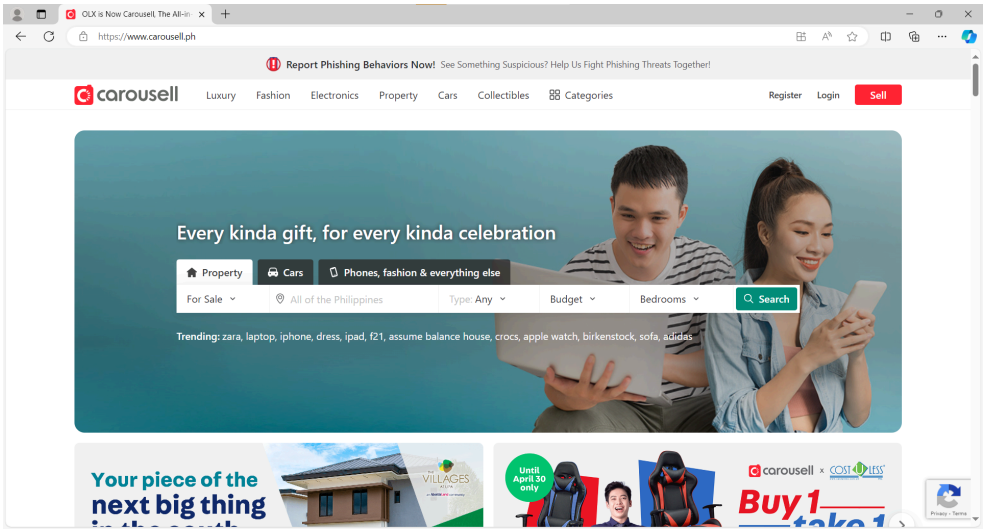


Figure 2.5 Carousell homepage

Carousell.com, formerly known as OLX, is a platform where users can buy, sell, and rent various items, including housing and property. It allows individuals and real estate agents to list properties for sale or rent, including apartments, houses, condos, and commercial spaces. Users can browse listings, filter search results based on their preferences, and contact sellers or landlords directly through the platform. Carousell.com provides a convenient and accessible way for people to find housing and property listings in their area.

The current trend involves housing expansion and rapid enrollment, which have an unpleasant impact on students' well-being. Prior to the term "student housing" being recognized as an area of knowledge in housing research, studies regarding concerns pertaining to college and university students' choices for housing were mostly overlooked. Now co-housing, also known as collective housing, has grown in popularity because it improves quality of life by fostering social, economic, and environmental sustainability (Alcaraz et al., 2020). Ballesteros et al. (2022) conducted a study regarding the measurement of housing affordability in the Philippines, and according to the authors, by comparison with the Residual Income Method, the 30 percent of income standard overestimates home affordability among those with low incomes and underestimates among the upper income levels. That is to say, housing priced at 30% of income is out of reach for the poor and low-income households, but housing priced at 30% or more is within the reach of the rich and middle-class. What this shows is that renting a decent apartment can be costly,

especially for low-income households. Most students are left with no choice but to share a room with other students to afford the place.

According to OpenAI (2024), students can find roommates for room-sharing through various methods. These methods include: (a) social media platforms such as Facebook, Twitter, or Instagram which often have groups or pages dedicated to housing at specific universities or cities in which students can post about their search for roommates; (b) online roommate finders such as Roomsurf, Roommates.com, and Roomgo.net designed specifically for finding roommates, however, these services are only available in the US; and lastly (c) word of mouth where students can ask friends, classmates, or acquaintances if they know anyone looking for a roommate or if they can recommend someone.

Websites, like Roomsurf, were designed like social media in which people can communicate with each other, however, these websites do not have an option for homeowners or landowners to lease their living spaces.

Roomsurf

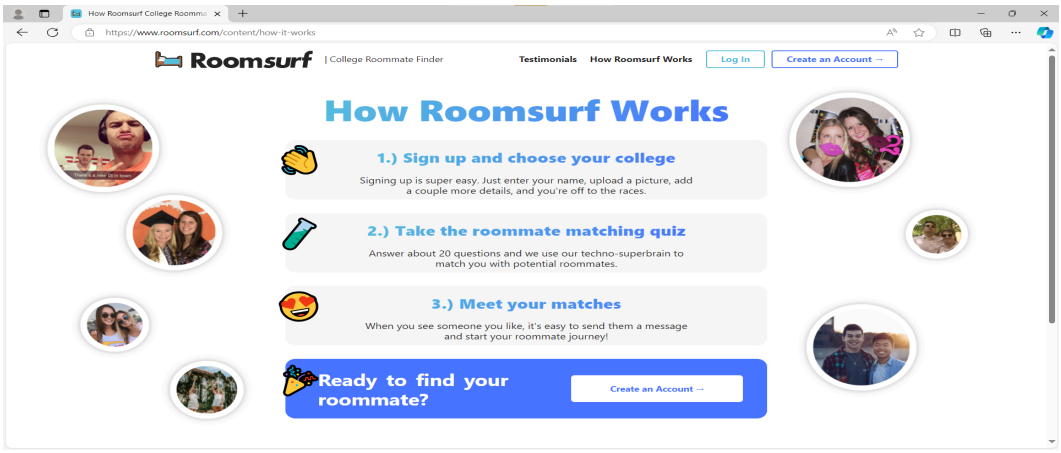


Figure 2.6 How Roomsurf Works | Roomsurf.com

Roomsurf.com is a website that helps college students find compatible roommates. It uses a matching algorithm to connect students who are attending the same university or college and are looking for roommates. Users create a profile and answer questions about their habits, preferences, and lifestyle, and then Roomsurf suggests potential roommates based on compatibility. It's a tool designed to make the roommate search process easier and more efficient for students.

The websites that are closest to what CribConnect is aspiring to be are: Roomgo.net, formerly known as Easyroommate, Roommates.com, and Dormy.ph. These websites allow users to list their living space in hopes of finding a roommate. Among the three, Dormy.ph is the only website that is available in the Philippines.

Roommates.com

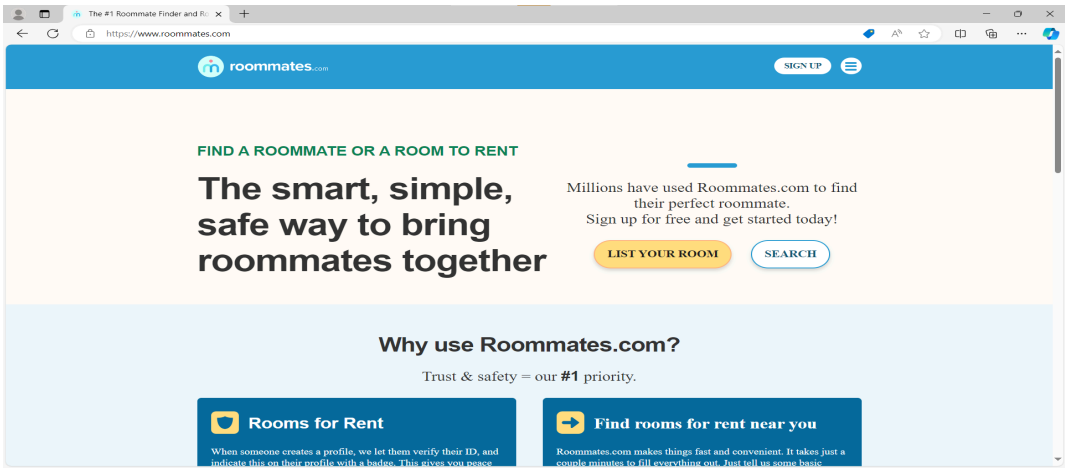


Figure 2.7 Roommates.com homepage

Roommates.com is a website that helps people find roommates or rooms for rent. It allows users to create profiles, search for compatible roommates based on various criteria such as location, rent budget, lifestyle preferences, and communicate with potential matches. The website aims to make the process of finding a roommate or a place to live easier and more efficient.

Roomgo.net

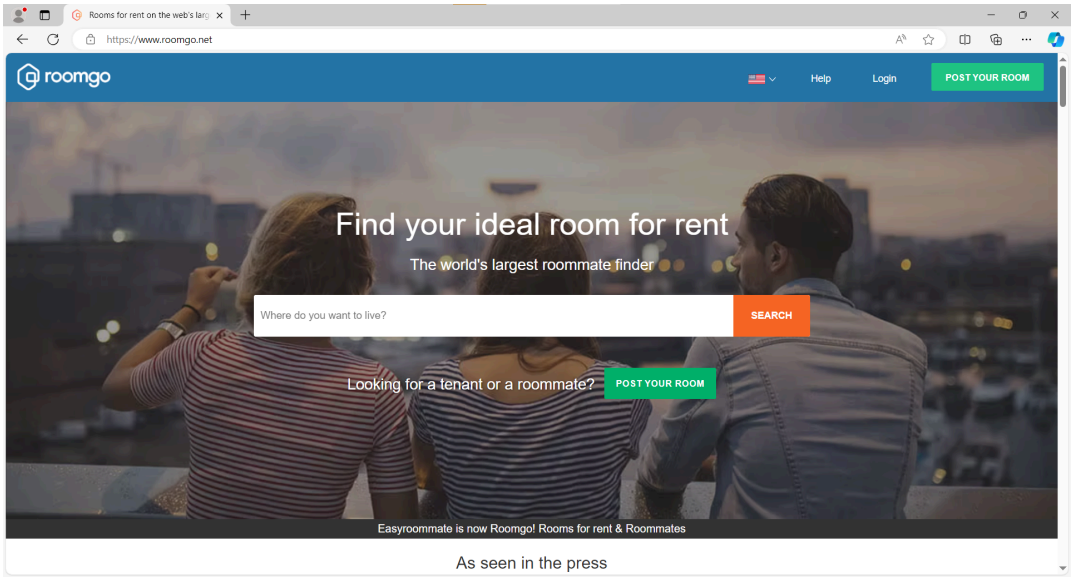


Figure 2.8 Roomgo.net homepage

Roomgo.net is a website that helps people find roommates and rental accommodations. It's essentially a platform that connects individuals looking to rent out rooms with those seeking to rent them. Users can create profiles, search for available rooms or roommates based on location, budget, and preferences, and communicate with potential matches through the website. Roomgo.net typically provides features such as messaging, filtering options, and profile verification to facilitate safe and efficient roommate and rental searches.

Dormy.ph

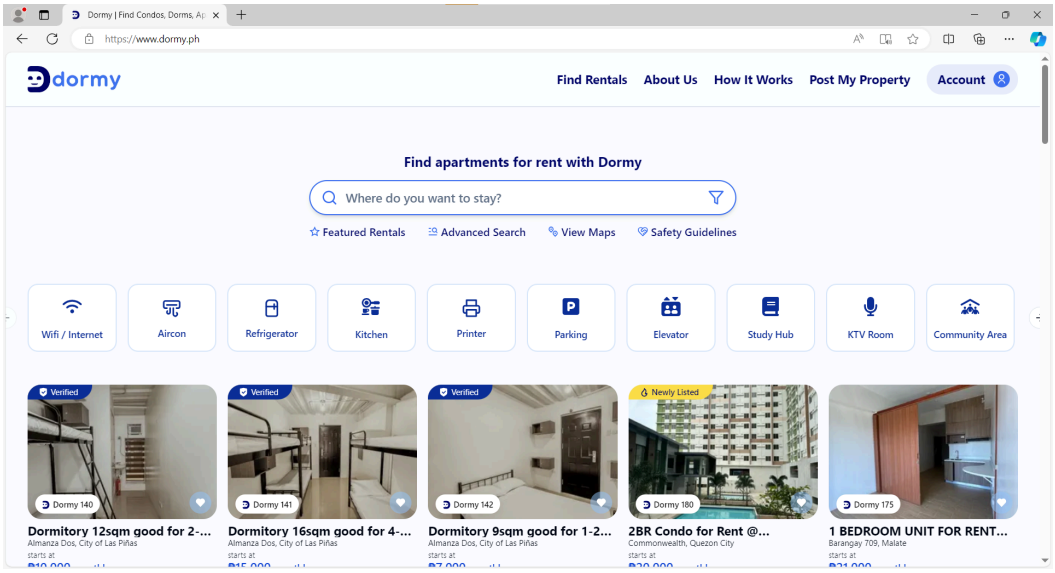


Figure 2.9 Dormy.ph homepage

Dormy.ph is a marketplace for residential stays, where renters can easily find residential spaces for rent in one platform. It also offers scheduling for a property viewing, and roommate matching.

The main difference between CribConnect and the existing solutions, particularly Dormy.ph, is that CribConnect includes a rating system. With this system in place, landowners can choose whether to accept a user's request to rent their property or not which will give landowners a chance to filter out those with bad reputations. The user's rating is based on their previous landlords. This will also benefit other users as they will not have to deal with a roommate with bad behavior.

III. SOFTWARE REQUIREMENT SPECIFICATIONS (SRS)

A. External Interface Requirements

1. User Interface

The user must be able to understand the overall aesthetics, the functionality must be easily accessible, and the design must be straightforward but effective. The UI needs to offer helpful shortcuts that are significantly connected to longer processes. The system's GUI has to be simple to use and intuitive. In order to optimize the hardware's available resources and enhance user-friendliness, a basic design approach is required.

2. Hardware Interfaces

The user should at least use the required specifications listed on Table 3.1 and Table 3.2 below since the developers aim to make the system functional for these specifications at the very least.

Table 3.1 Computer Specification

Hardware	Specification
Processor	At least Intel Pentium 4 or AMD Athlon XP for Windows/Linux; at least Intel Core Duo for Mac
Memory	At least 4GB
Network Connectivity	Ethernet, WiFi

Table 3.2 Smartphone Specification

Hardware	Specification
Processor	At least Snapdragon S4 Play MSM8225 for Android; at least Apple A4 for iPhone

Memory	At least 2GB
Network Connectivity	WiFi, Mobile Data

3. Software Interfaces

- Operating System - a software program that acts as an intermediary between computer hardware and the applications running on it. It manages computer hardware resources and provides common services for computer programs. The OS typically handles tasks such as memory management, process scheduling, file management, and device management. These are the operating systems that we recommend: Windows 7, 8, 8.1, Windows 10, macOS, Linux distro, Android, iOS.

- Browser - a software application used to access information on the World Wide Web. It retrieves and displays web pages, allowing users to navigate the internet and interact with various online content. These are the browsers that we recommend: Chromium-based, Gecko-based, Goanna-based, Trident-based, Webkit-based, Blink-based, KHTML-based, EdgeHTML-based, MSHTML-based, and Presto-based browser.

- PythonAnywhere - a cloud-based Python development and hosting platform that allows users to write, run, and host Python applications entirely in the cloud. It provides an integrated development environment (IDE) where you can write and debug your Python code directly in your web browser. PythonAnywhere also offers web hosting capabilities, allowing users to deploy their Python web applications and websites easily.

- Illustrator - a vector graphics editor developed and marketed by Adobe Inc. It is primarily used to create logos, icons, drawings, typography, and complex illustrations. It provides a wide range of tools for drawing, painting, and manipulating objects, allowing users to create scalable, high-quality artwork.

- Figma - a cloud-based design tool used for creating user interfaces, websites, and app prototypes. It is known for its collaborative features, which allow multiple users to work on the same design in real-time.

- HTML - the standard markup language used to create and design documents on the World Wide Web. It provides the structure for web pages by using a system of tags and attributes to define elements within a document, such as headings, paragraphs, links, images, and more.

- CSS - a style sheet language used to describe the presentation of a document written in HTML or XML (including XML dialects). It describes how elements should be rendered on screen, on paper, in speech, or on other media.

- Javascript - a high-level, dynamic, and interpreted programming language primarily used for building interactive web pages and web applications. It is often referred to as the "language of the web" because it is widely used to enhance web pages with dynamic and interactive features.

- Python - a high-level, interpreted programming language known for its simplicity and readability. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

- Django - a high-level web framework written in Python that encourages rapid development and clean, pragmatic design. It follows the model-view-controller (MVC) architectural pattern. Django's primary goal is to ease the creation of complex, database-driven websites by emphasizing reusability and "pluggability" of components.

- SQL - a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS) or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data, where there are relations between different entities/variables of the data.

- SQLite - a lightweight, serverless, self-contained, high-reliability, full-featured, and public-domain SQL database engine. It's often used in mobile apps, IoT devices, and desktop software where a full-scale client-server database system like MySQL or PostgreSQL might be overly complex or unnecessary.

4. Communication Interfaces

The table below shows the communication interfaces that the system will be requiring in order to function properly

Table 3.3 Communication Interfaces of the System

HTTPS	The system will specifically use HTTPS to communicate with the web server. It is the foundation of data communication on the World Wide Web. It is an application-layer protocol that defines the format for messages sent between web browsers and web servers.
WebSocket	It is a protocol that provides full-duplex communication channels over a single TCP connection, commonly used for real-time web applications.
Ethernet	This is a widely used standard for wired networking. It allows devices to communicate over a local area network (LAN) using a protocol such as TCP/IP.
Wi-Fi	Wireless communication technology that allows devices to connect to a network using radio waves. It is commonly used for internet access and local network connections.

B. Non-Functional Requirements

1. **Performance Requirement**

The interface should be user-friendly for property seekers and landlords, ensuring smooth navigation. It must scale effectively with increased users and listings without sacrificing performance. Additionally, it should be mobile-responsive for smartphone and tablet users. Lastly, the feedback system should enable users to rate properties, landlords, and tenants.

2. **Safety Requirement**

As part of the safety and security of the owners and searchers, the system will require the users to undergo a user verification process before they can access the full functionality of the application to prevent malicious behaviors from bad actors.

3. Security Requirement

Under the Data Privacy Act of the Philippines, the project must obtain user consent before disclosing any personal information. The system only uses data for its intended purpose and complies with Republic Act No. 10173, safeguarding sensitive details like names and emails. This act protects all forms of information and applies to individuals and organizations processing personal data. All data collected from respondents is kept confidential.

4. Software Quality Attributes

- **Compatibility**

  This system, being a web-based application, should be able to run on any operating system, be it on a desktop, laptop, or any smartphone so long as they have access to the internet.

- **Acceptability**

  The system must be acceptable to the types of users for which it is designed.

- **Simplicity**

The proponents designed the system in a way that the users will be able to use the application with ease. The developers made the user interface user-friendly and less complicated as possible.

- **Efficiency**

The system was designed in a way that will not consume too much resources from the user's device.

IV. SOFTWARE DESIGN EVALUATION

A. Abstract

This project is a listing site that permits its users to inquire to rental properties of landholders. These listings are viewable to guest users with images and details of the apartment while leaving the landholder's contact information hidden, once a user is logged in into an account, the user landholder's contact information will become visible and the user will now be able to send requests. SQLite is the choice for this application's database and uses a web framework called Django for its development. The whole software consists of multiple languages with 53.2% HTML, 27.9% Python, 18.8% CSS and a few JS for UI/UX purposes. The project aims to make a visually appealing website that is not lacking with its functionalities, providing an easy navigation module for users. Moreover, a key function of the software called Cribshare should allow apartment seekers to share a certain apartment they want with others. This functionality could only be accessed once an account has been verified. Lastly, the rating and review system will only become relevant once the system is used recurrently, this could potentially lessen the user's perception of the software in general but the functionalities offered shall be enough to address most of the problems they're facing with regards to apartment searching.

B. Introduction

In creating a well designed software, the developers must consider different factors that affect the whole functionality of the system. Using a framework such as Django could minimize a lot of issues that can potentially arise during the development phase of a project. Additionally, the database to be used is also a key consideration that might impair the system's functionality. With this in mind, this project is meant to provide apartment seekers and landlords with a software that is essentially oriented towards the long term lodging experience through a bilateral rating and review system in addition to rent sharing.

As lodging becomes more expensive overtime it becomes quite difficult to adjust expenses accordingly, especially for students. Thus, most students that are far from their school compromise by commuting regardless of the time it may consume. Splitting the rent with others is one of the solutions most apartment seekers opted into, but finding a trustworthy roommate is quite difficult specifically through online inquiry. Apart from that, landholders' major problem in accepting tenants online is the credibility of the

tenants who inquire as they are essentially putting their trust in them for maintaining their properties. Furthermore, there is no such thing as a comprehensive criteria of properties for rent that apartment seekers can rely upon, leading to them wasting time in personally checking apartments for rents that do not fit their preferences. In any case, these problems haven't yet been addressed by most of the apartment listing sites. Furthermore, with the daily traffic reaching 3.63 million vehicles in Metro Manila (Statista, 2024), traveling to seek an apartment could be physically tasking to many. Clearly, there is a need for a software that can provide a solution that can precede the traditional apartment searching method in addition to the problems previously discussed.

On the other hand, there are a number of listing sites that address the problem of seeking an apartment, but there are aspects that can be further improved. Additionally, platforms such as Facebook marketplace which is quite popular for its convenience, does not have any criterion for its apartment listings that could potentially help apartment seekers to identify or validate the apartment they're interested in. The rating is limited to the products and does not include apartment listings. There is also a lack of options with regards to understanding what are the necessary changes towards the apartment listed or its tenant, hence a review seems to be an appropriate solution towards that.

Therefore, the team concludes that a listing software with bilateral rating and review system would be beneficial towards both the apartment seeker and landlords, providing them the opportunity to change how they manage their apartments or how they maintain the apartment they rent. A verification of users is also an essential part of the software for security measures, especially for the crib share feature that lets the user request to split the rent with others. Thus the software must include the following features: Apartment listing, Apartment requesting, Verification request, Cribsharing, and Rating and Reviews. The software is expected to have a minimal range of traffic with these functionalities, therefore a database that is fast and efficient in terms of queries such as SQLite are preferable. Lastly, to be efficient with the codes a framework is essential, since it can provide a wide range of options with regards to queries, variables, and layers towards the development process. Django provides different layers to its users such as models, templates, and views that could be used to simply organize the whole software that boosts the efficiency of its developers.

C. Review of Related Literature

The system is expected to receive small to medium traffic since the main functionality of the software is the inquiry requests which ideally won't be used often, this is one of the reasons for the choice of database management to use. SQLite is the database used for this project, as an embedded database that is commonly used for mobile applications and web applications with low to medium level of traffic queries. SQLite offers exactly what the system requires, a sql-based database that offers a unique feature which is that the entire database is contained within a single file. For this reason, most developers had historically seen it as the simplest database for simple use cases, and within the recent years, SQLite has received a great amount of attention from its developers to make it the simplest database with advanced use cases. Hence for most developers, SQLite is the best balance of trade-offs (K. Dodds, 2024).

Web frameworks are proven effective in streamlining the web development process by providing the developers an access to its pre-written code libraries and modules that improves the working efficiency of developers, therefore the project used a back-end framework called Django. Back-end frameworks handle the server side of web applications. It handles HTTP requests, manages the system's databases, and improves the overall security of the web application. Benefits of using web frameworks includes: Easier development process, Eases debugging and application maintenance, Reduces code length, Improves database proficiency, and Reinforces security (Sencha, 2023). The system uses the user authentication provided by Django, it uses the PBKDF2 algorithm with a SHA256 hash which is a password stretching mechanism that is recommended by NIST. Nevertheless, on the function side, consumer reviews help in determining the issues of certain products, shedding light on new use cases, and provide ideas for future innovation. Online shoppers are expecting a change in businesses as a response to their reviews., both positive (25%) and negative (38%), in creating a better product (G. Cadorette, 2023). Apartments can be viewed as products that require constant improvement in order to stay relevant, but these reviews are not limited to properties as feedback is used towards employees as well, this method is used to determine the effectiveness of their staff. A similar approach can be done towards tenants, this is to ensure that the tenants do not induce any damage or ignore the basic maintenance needed of the apartment they rent, this is also to reflect with themselves what are the things to improve in order to create a maintainable relationship with their landlords.

In contradiction, the database used SQLite provides the essential functions for the software's minimal operation but the long term operability could be in question. In terms of a better choice in database MySQL seems to be a better option for the database of the software and with this regards the database used can be changed once the software obtains a larger market. Furthermore, using the authentication that a framework provides could be further improved by implementing additional security measures towards the software. Lastly, the bilateral rating and review system doesn't have any certified criteria that can objectify each input leading to a subjective factor which makes it hard to validate.

D. Methodology

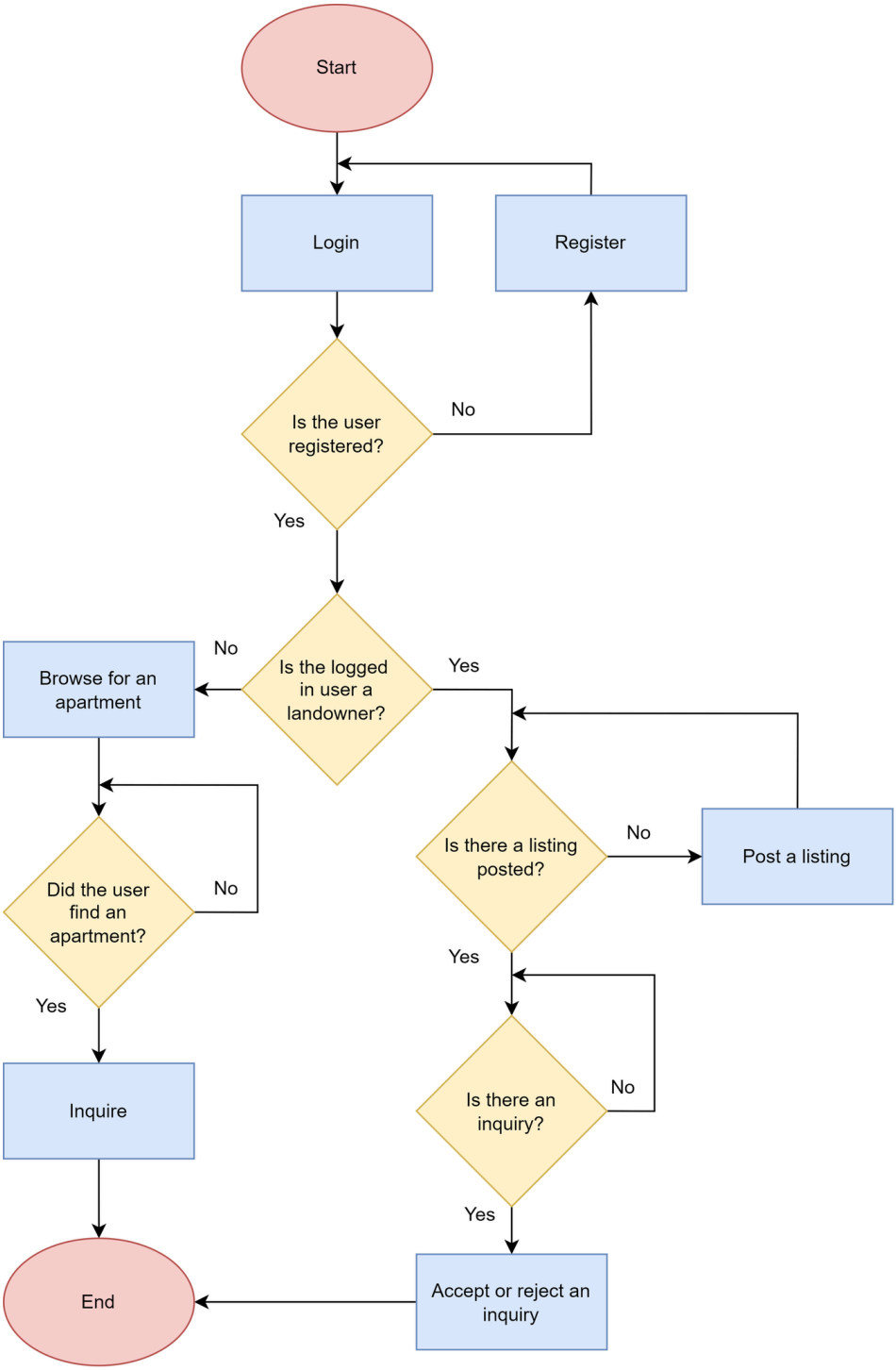


Figure 4.1 Flowchart of the General Process of the System

Register

Registering on our system is the first process. Both landowners and tenants can register on our system to access the features and functions of it. They must have to provide information or documents in order to proceed.

Login

At this stage of the process, either landowners or tenants must log in using a registered account. Users must enter their email address and the associated password into the system. Following that, the system will check the user account database to determine if the email and password combination already exists. The system will display the user's homepage if it is located in the database entries.

Homepage

This is the section of the system where the user can see all of the system's features and functions. Users will have access to the homepage property listings and the property ratings.

Property Listing

This is the section of the system, users can see the key feature of this system by seeing all of the photographic amenities of the property, including the property rating, tenant ratings and the landlord ratings on the tenants.

Chat

Both landlords and tenants can start a conversation with each other in regards of the property concerns.

E. Conclusion and Recommendation

Various challenges developed throughout the creation of the system. However, all project requirements were met within the research scope and constraints, specifically focusing on maximizing user engagement for both tenants and landowners. A key feature implemented is a reputation rating system. Landowners can leverage this system to filter potential tenants with negative ratings, minimizing the risk of bad experiences. Conversely, tenants can view past ratings left by previous landlords on the tenants, allowing them to avoid problematic living situations. This two-sided rating system builds trust and creates a more secure platform for both parties.

In terms of building this system it is a combination of freely available algorithms and development techniques gleaned from online tutorials and our knowledge of Database Management Systems (DBMS). While this approach proved functional for the initial prototype, further research could explore the integration of more sophisticated algorithms for improved rating accuracy and user matching.

By implementing these features, the platform can transition beyond simply connecting tenants and landowners. It can become a comprehensive resource for renters, providing them with the necessary tools to make informed decisions and fostering a sense of community within the properties it manages. This holistic approach will undoubtedly enhance user satisfaction and contribute to the platform's long-term success.

Furthermore, this system can be enhanced by boosting user engagements in both landowners and tenants thus offering virtual tours of different room types and common areas could be a great way to showcase the landlords offered properties and amenities whilst the user could have more clearer picture on the unit that they think suitable for their comfortability stays and needs; it is also best to highlight nearby landmarks, attractions and showcasing upcoming events and activities can further highlight the vibrant community within the scope of the place that the landowners property is built.

V. SOFTWARE DESIGN DESCRIPTION (SDD)

A. System Architecture

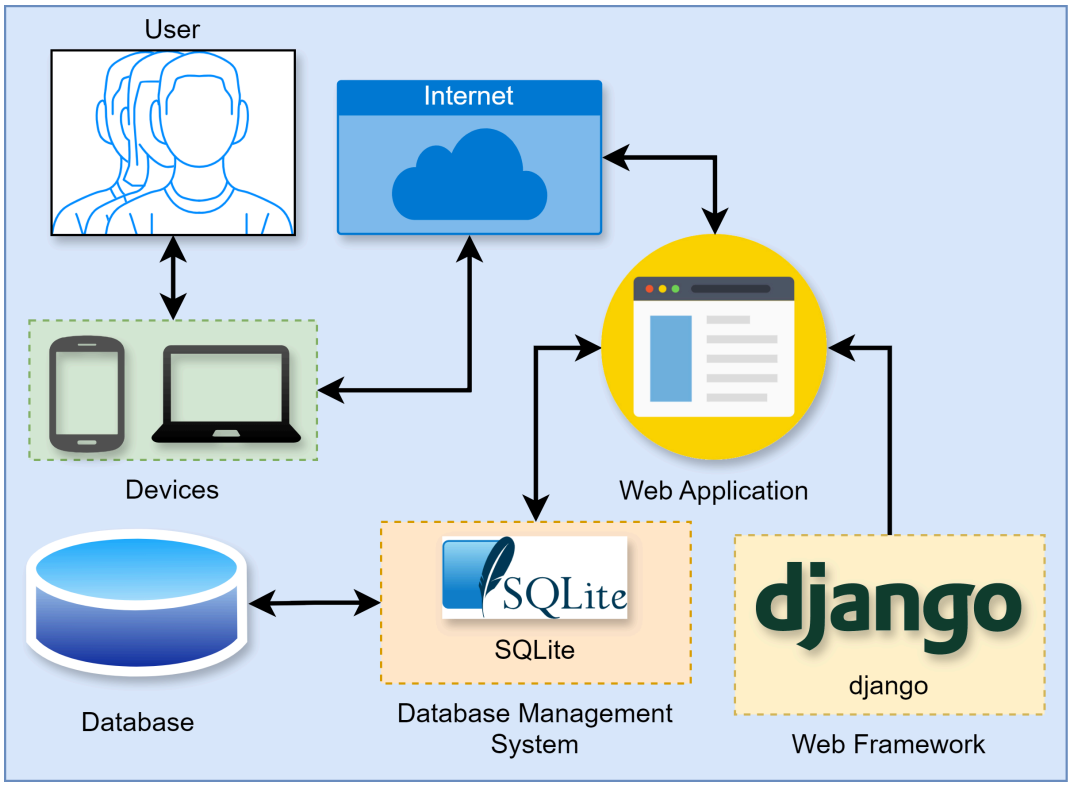


Figure 5.1 CribConnect System Architecture

B. Constraints

Table 5.1 System Constraint

Design Constraints	Design Criteria
Security	• The system shall provide security, enough to prevent bad actors from performing any actions that can do harm or breach the privacy of the users
Functionality	• The system shall work as the developers intended and shall be user-friendly to be easily understood by the users.
Maintainability	• The system shall easily be fixed and restored to a working state should an error occur.

C. Functional View

Figure 5.2 Apartment Inquiring System

All users shall be able to login using their account's username and password which the system will verify, a login error will pop up if the login is unsuccessful. The system shall also provide the users (excluding admin) a rating for both apartment and tenants which will be updated by the system. The inquiring process includes showing the user information to the landowner. The

landowners can accept the inquiry request which the system will handle by promoting the account as tenant as well as the apartment information.

D. Data Flow Diagram

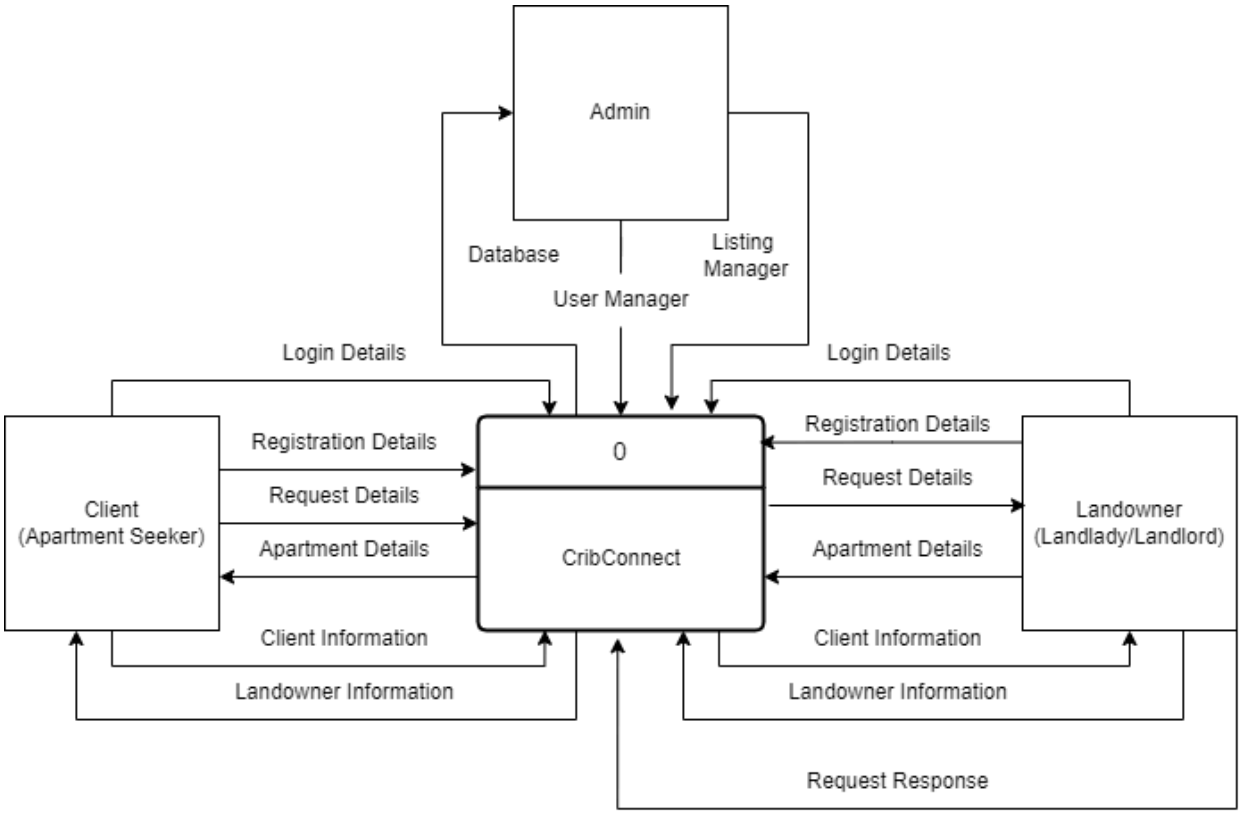


Figure 5.3 Data Flow Diagram Level 0

The diagram shown in Figure 5.3 provides the basic functionality of the system, the Admin will be provided with the whole database of the system for the managerial tasks. Both Landowner and Client can login and register an account, Landowners can create an apartment listing while the Client can view the apartment listed with the Landowners information, the Client can send a request towards the Landowner, the request contains the Client information which the Landowner can view before responding into.

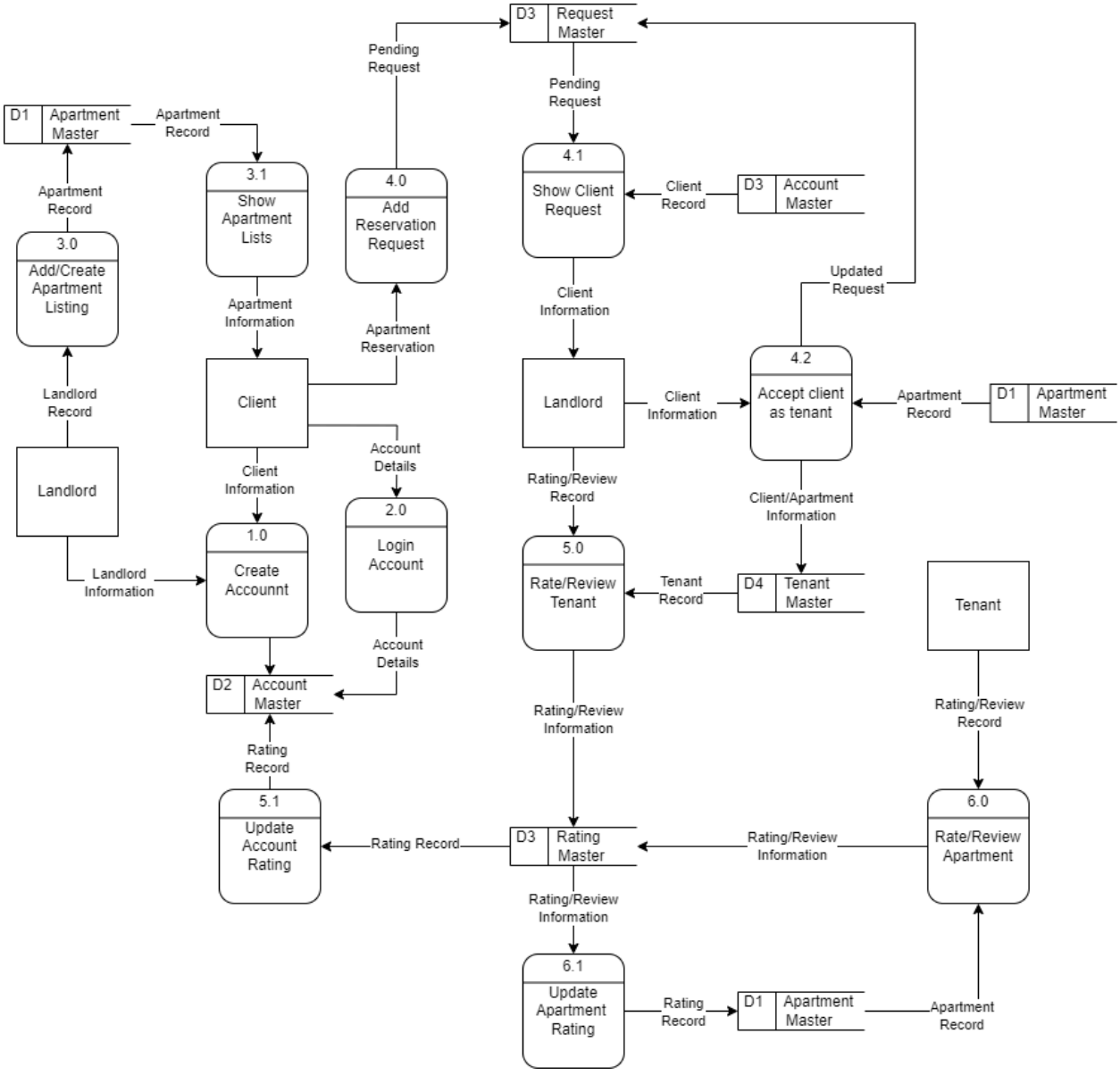


Figure 5.4 Data Flow Diagram Level 1

The system contains 6 main processes. Process 1 and 2 consist of the basic functions such as account registration and login. Process 3 is designated for Landlords to list their property and to show it to possible Clients. The reservation is within Process 4, which has 3 sub-processes involved, 1. Client adding a request, 2. Showing the request to the Landlord, and 3 handling the response of the Landlord. Lastly, process 5 and 6 are for rating and reviewing, which is a two way rating, process 5 is for rating and reviewing a tenant, while process 6 is for rating and reviewing an apartment.

E. System Flow Charts

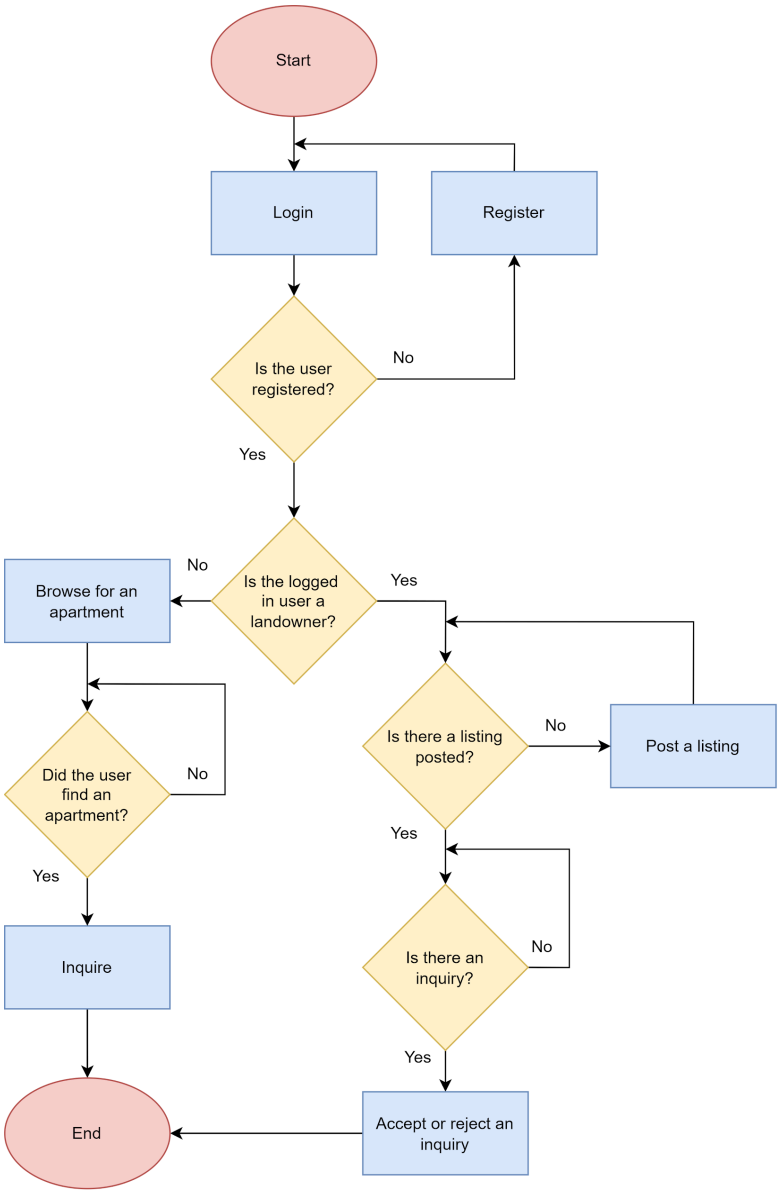


Figure 5.5 System Flowchart

F. Entity Relationship Diagram

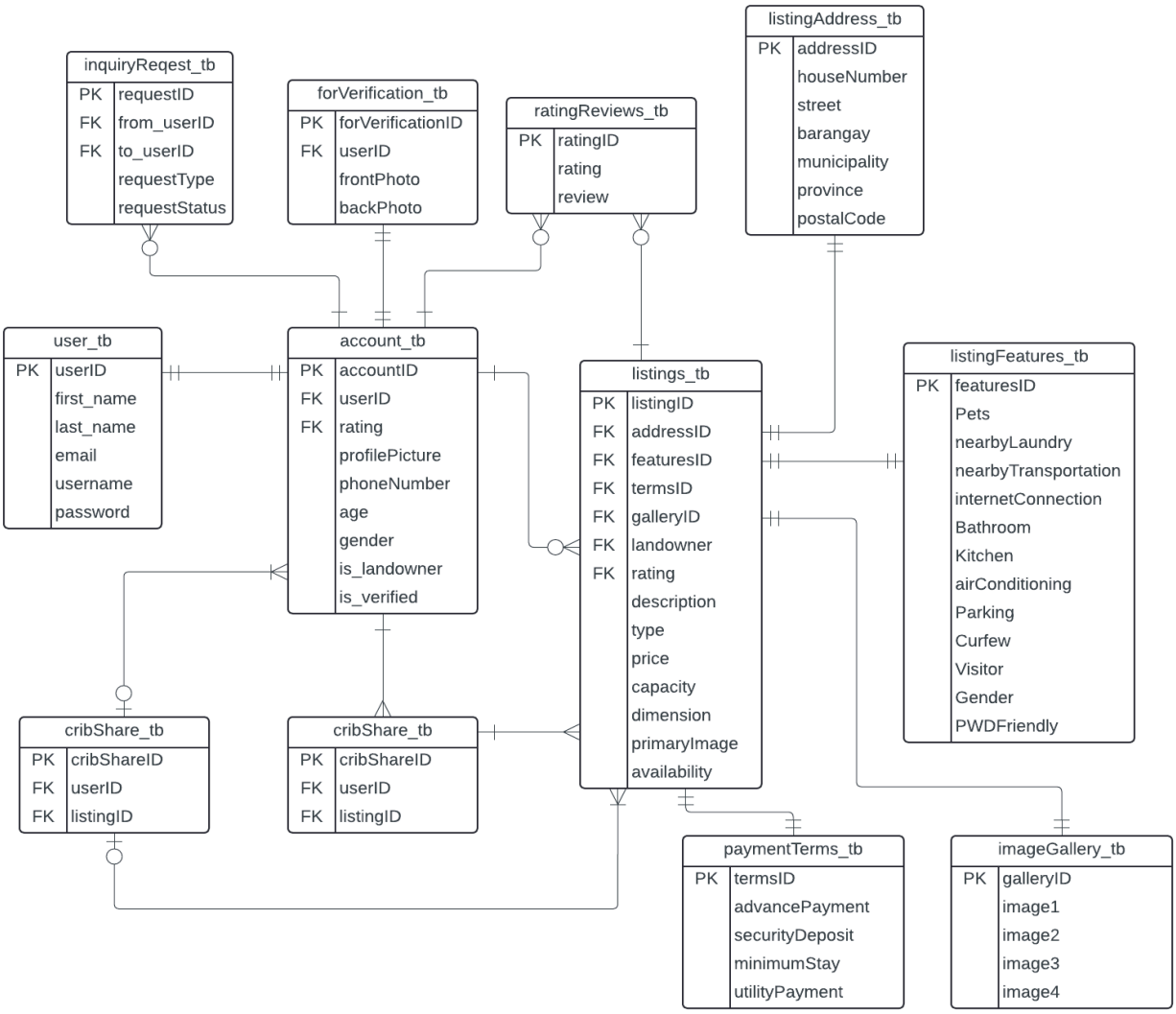


Figure 5.6 Entity-Relationship Diagram

G. Data Dictionary

user_tb					
Column Name	**Data Type**	**Field Size**	**Nullable**	**Key**	**Description**
firstName	varchar	64	No		First Name of the user
lastName	varchar	64	No		Last Name of the user
lastName	varchar	64	No		Last Name of the user
username	varchar	64	No		The username for login
password	varchar	64	No		The password for login
emailAddress	varchar	64	No		Email address of the user

Figure 5.7 user_tb Data Dictionary

ratingReviews_tb					
Column Name	**Data Type**	**Field Size**	**Nullable**	**Key**	**Description**
ratingID	int	8	No	PK	The ID of the rating and reviews
rating	int	1	No		1-5 star rating towards the user/apartment
review	varchar	255	Yes		Optional review towards the user/apartment

Figure 5.8 ratingReviews_tb Data Dictionary

account_tb					
Column Name	**Data Type**	**Field Size**	**Nullable**	**Key**	**Description**
accountID	int	8	No	PK	account ID of the user
userID	int	8	No	FK	User details for authentication
rating	varchar	64	No	FK	The account of account for logging in
phoneNumber	int	16	No		Phone Number of the user
age	int	3	No		Age of the user
gender	varchar	6	No		Gender of the user
is_verified	boolean	2	No		Verification Status of the user
is_landowner	boolean	2	No		Verification Status of the user
profilePicture	file	-	Yes		The profile picture of the user

Figure 5.9 account_tb Data Dictionary

listingAddress_tb					
Column Name	**Data Type**	**Field Size**	**Nullable**	**Key**	**Description**
addressID	int	8	No	PK	ID of the address
houseNumber	varchar	16	No		The house number of the listing
street	varchar	16	No		The street of the listing
barangay	varchar	16	No		The barangay of the listing
municipality	varchar	16	No		The municipality of the listing
province	varchar	16	No		The province of the listing
postalCode	int	8	No		The postal code of the listing

Figure 5.10 listingAddress_tb Data Dictionary

listingFeatures_tb					
Column Name	**Data Type**	**Field Size**	**Nullable**	**Key**	**Description**
featuresID	int	16	No	PK	ID of the features of a listing
pets	boolean	2	No		If pets are allowed
nearbyLaundry	boolean	2	No		If listing has nearby laundry shops
nearbyTransit	boolean	2	No		If listing is near public transportation
wifi	boolean	2	No		If listing has internet connection
bathroom	boolean	8	No		If bathroom is private, communal, or none
kitchen	boolean	8	No		If kitchen is private, communal, or none
airConditioning	boolean	2	No		If listing has airconditioning
parking	boolean	2	No		If listing has nearby parking space
curfew	boolean	2	No		If listing has a curfew
visitor	boolean	2	No		If visitors are allowed
maleOnly	boolean	2	No		If limited to male occupants
femaleOnly	boolean	2	No		If limited to female occupants
accessibility	boolean	2	No		If listing is accessible to PWD

Figure 5.11 listingFeatures_tb Data Dictionary

paymentTerms_tb					
Column Name	**Data Type**	**Field Size**	**Nullable**	**Key**	**Description**
termsID	int	8	No	PK	The ID of the payment term
advancePayment	int	16	Yes		The amount of the advance payment
securityDeposit	int	16	Yes		The payment amount for the deposit
minimumStay	int	16	Yes		The number of minimum stay in years
utilityPayment	int	16	Yes		The payment amount for the utility

Figure 5.12 paymentTerms_tb Data Dictionary

imageGallery_tb					
Column Name	**Data Type**	**Field Size**	**Nullable**	**Key**	**Description**
galleryID	int	8	No	PK	The ID of the image gallery for the listing
image1	file	-	No		The image of the property
image2	file	-	No		The image of the property
image2	file	-	No		The image of the property
image2	file	-	No		The image of the property

Figure 5.13 imageGallery_tb Data Dictionary

listing_tb					
Column Name	**Data Type**	**Field Size**	**Nullable**	**Key**	**Description**
listingID	int	8	No	PK	The ID of the listing
addressID	int	8	No	FK	ID of the address
featuresID	int	16	No	FK	ID of the features of a listing
termsID	int	8	No	FK	The ID of the payment term
rating	int	1	Yes	FK	The ID of the ratingReviews
landowner	int	8	No	FK	The accountID of the owner
description	varchar	255	Yes		Description of the listing
type	varchar	16	No		Type of living space of the listing
price	int	16	No		The price per month of the listing
capacity	int	3	No		The number of occupants allowed in the listing
dimension	varchar	16	No		The size of the living space of the listing
primaryImage	varchar	255	No		The directory of the pictures of the listing
is_available	boolean	1	No		Is the listing available or not

Figure 5.14 listing_tb Data Dictionary

inquiryRequest_tb					
Column Name	**Data Type**	**Field Size**	**Nullable**	**Key**	**Description**
requestID	int	8	No	PK	ID of the request
from_userID	int	8	No	FK	UserID from the sender of the request
to_userID	int	8	No	FK	UserID of the requester
requestType	varchar	16	No		The type of the request
requestStatus	boolean	1	No		The status of the request

Figure 5.15 inquiryRequest_tb Data Dictionary

VI. SOFTWARE TESTING AND EVALUATION

A. Test Approach

User or customer testing is a crucial phase in the testing procedure, involving users or customers providing feedback on system performance. This could be formal, where a system from an external supplier is rigorously tested, or informal, with users exploring a new software to assess its suitability. Even after comprehensive system and release testing, user testing remains vital as it uncovers how the system operates in the user's environment, impacting reliability, performance, usability, and robustness.

B. Test Plan

The test will be performed in such a way that users can utilize the system for the tenants to view listings, inquire that fits their preference for the living set-up, and get a listing. While for the landlords, they will be able to create their listing for the tenants to view and accommodate, approve or disapprove an inquiry request from the tenant and lastly to rate their tenants so that the other tenants can see the other tenants behavior and performance during the stay in the rented property.

C. Testing Requirements

Table 6.1 System Requirements

Requirements ID	Name	Description
SR001	View Website Home Page (Guest)	The users should be able to view the listings available and open the login form.
SR002	View Website Home Page (Registered)	The users should be able to view the listings available and access the user module.
SR003	Login with incorrect Username/password	The users should be able to see if the username/password is incorrect
SR004	Login with correct Username/password	The users should be able to login their account and be redirected to the homepage
SR005	Register/Create Account	The users should be able to register their username and password

SR006	Fill information for account creation	The users should be redirected to the fill-up page and input their personal information
SR007	View Registered Account	The users should be able to view their personal information.
SR008	Edit information (limited)	The users should be able to edit the information in their registered account, specifically the email, contact number and profile picture.
SR009	Delete Account	The admin should be able to delete an account.
SR010	Create/Add listing	The landlord should be able to create and add listing including the pictures of the property
SR011	View listing	The users should be able to see the listing details such as image, address, etc. while verified users can view the contact details of the owner and cribsharers.
SR012	Delete existing listing	The landlord should be able to delete the existing listings.
SR013	Add inquiry request	The searchers should be able to add an inquiry request to the landlords
SR014	View inquiry request	The users should be able to see the inquiry requests sent and received.
SR015	Accept/Reject inquiry request	The landlords should be able to accept and reject the request of the searchers.
SR016	Rate tenant	The landlords should be able rate the tenant
SR017	Rate listing	The tenants should be able to rate listings that they currently occupy.
SR018	View rating and reviews	The user should be able to view the ratings and reviews of their property or account.
SR019	Upload ID for User Validation	The users should be able to upload an ID for their account validation
SR020	View validation request	The admin should be able to see the validation requests of the users.
SR021	Edit Verification Status	The admin should be able to approve or reject the validation request.

D. Testing Cases Summary

Table 6.2 Test Cases

Test Case ID	Name	Requirement to be tested
TC001	View Website Home Page (Guest)	SR001
TC002	View Website Home Page (Registered)	SR002
TC003	Login with incorrect Username/password	SR003
TC004	Login with correct Username/password	SR004
TC005	Register/Create Account	SR005
TC006	Fill information for account creation	SR006
TC007	View Registered Account	SR007
TC008	Edit information (limited)	SR008
TC009	Delete Account	SR009
TC010	Create/Add listing	SR010
TC011	View listing	SR011
TC012	Delete existing listing	SR012
TC013	Add inquiry request	SR014
TC014	View inquiry request	SR015
TC015	Accept/Reject inquiry request	SR016
TC016	Rate tenant	SR017
TC017	Rate listing	SR018
TC018	View rating and reviews	SR019
TC019	Upload ID for User Validation	SR020
TC020	View validation request	SR021

TC021	Edit Verification Status	SR022

E. Result of the Testing

Table 6.3 Test Cases Results

Test Case ID	Result	Remarks
TC001	Passed	
TC002	Passed	
TC003	Passed	
TC004	Passed	
TC005	Passed	
TC006	Passed	
TC007	Passed	
TC008	Passed	
TC009	Passed	
TC010	Passed	
TC011	Passed	
TC012	Passed	
TC013	Passed	
TC014	Passed	
TC015	Passed	
TC016	Passed	
TC017	Passed	
TC018		Review is not still visible
TC019	Passed	

TC020	Passed	
TC021	Passed	

VII. CONCLUSION AND RECOMMENDATION

In conclusion, a well-designed and efficiently implemented online listing and rental platform is paramount for the success of any property management business. It serves as the backbone of operations, facilitating seamless management of property listings, tenant inquiries, and overall rental processes. An effective online platform provides real-time visibility into property availability, reduces the risk of vacancies, and enhances decision-making by providing accurate and up-to-date information.

Furthermore, although the platform requires property owners to manually post their listings, it still significantly increases operational efficiency. It enables businesses to optimize their resources, streamline workflows, and allocate capital more effectively. Additionally, the integration of user-friendly features such as advanced search filters, secure communication channels, and data analytics enhances the overall user experience and management processes.

Ultimately, a robust online rental platform not only contributes to cost savings but also enhances tenant satisfaction by ensuring property availability and timely responses to inquiries. As the real estate market continues to evolve, investing in and maintaining a state-of-the-art online listing and rental platform is essential for staying competitive in today's dynamic and fast-paced market.

For future studies, several recommendations can be made to enhance the development and effectiveness of an online listing and rental platform. Integrating AI and machine learning can improve search functionality and provide personalized property recommendations, enhancing user experience. Additionally, implementing advanced data analytics to study user behavior will offer insights for continuous improvement of the platform’s interface and functionality. Enhanced security measures, such as blockchain for secure transactions and advanced encryption, should be researched to ensure user data privacy and protection. The inclusion of virtual and augmented reality for property tours can significantly enhance the user experience by allowing potential tenants to explore properties remotely in an immersive way. Furthermore, optimizing the platform for mobile use and developing dedicated mobile applications can increase accessibility and convenience for users.

Other recommendations include researching the integration of sustainability features into property listings to appeal to environmentally conscious users, and implementing AI-powered chatbots to provide immediate customer support. Encouraging user-generated content, such as reviews and ratings, can enhance trust and credibility. Providing detailed local market insights, including neighborhood statistics and nearby amenities, can add significant value to users. Finally, developing features that allow for customization and personalization of search criteria and notifications will improve user satisfaction. By focusing on these areas, future studies can contribute to the continuous improvement and innovation of online listing and rental platforms, ensuring they meet the evolving needs of both property owners and tenants.

Appendices

Appendix A: Software Project Management Plan (SPMP)

A. Software Process Model

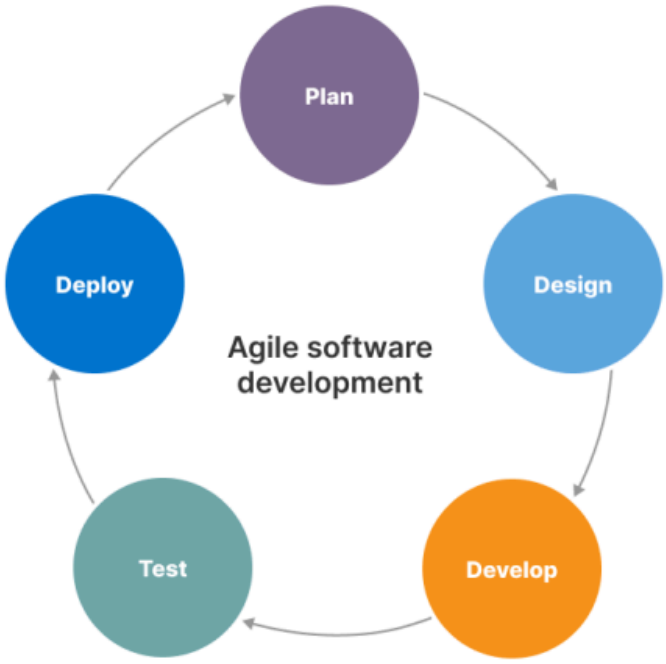


Figure A.1 Agile Software Development

B. Roles and Responsibilities

Table A.1 Roles and Responsibilities

Roles	Responsibilities	
Programmer	Is responsible for the programming of the system and ensuring that the system is fully functional	Carpio, Kristine Shyra A. Villegas, Eugene D.
UI/UX Designer	Is responsible for preparing and improving the user interface into a user-friendly and easy to navigate and understand.	Carpio, Kristine Shyra A. Villegas, Eugene D.
Database Designer	Is responsible for designing the database of the system.	Dela Cruz, Gabriel D. Dida-Agun, Abdul Rahman A.
System Analyst	Identifies the improvements needed, design systems to implement those changes and to motivate other members to work.	Villegas, Eugene D.

Technical Writer	Is responsible for writing documents that explains the system functionalities and requirements.	Carpio, Kristine Shyra A. Dela Cruz, Gabriel D. Diaz, John Mikael R. Dida-Agun, Abdul Rahman A. Javier, Beejay B. Villegas, Eugene D.
Software Tester	Identifies test conditions and creates test designs, test cases, test procedure specifications and test data, and may automate or help to automate the tests.	Dela Cruz, Gabriel D. Diaz, John Mikael R. Dida-Agun, Abdul Rahman A. Javier, Beejay B.

C. Time Table

Table A.2 Time Table

PHASE	TASK NO.	ACTIVITIES/TASK	RESPONSIBILITY	DURATION	DELIVERABLES	WEIGHT
PROJECT PLANNING	1	DETERMINING THE PROJECT GOAL	ALL	5 DAYS	ANSWERS TO THE QUESTION OF WHAT THE PROJECT IS ALL ABOUT	3
	2	RESEARCH FOR RELATED LITERATURE	ALL	3 WEEKS	RELATED RESEARCH ABOUT THE PROBLEM AND THE EFFECTIVENESS OF THE SOLUTION	3
	3	IDENTIFYING POTENTIAL CUSTOMERS	ALL	1 DAY	LIST OF POSSIBLE USERS	1
	4	SURVEY	JAVIER, VILLEGAS	1 WEEK	ANSWERED ONLINE SURVEYS	1

	5	FEASIBILITY STUDY	ALL	5 DAYS	INFORMATION REGARDING THE PROJECT	3
	6	REQUIREMENT ANALYSIS	DELA CRUZ, VILLEGAS	1 DAY	PARTIAL DOCUMENTATION FOR THE REQUIREMENTS OF THE PROJECT	3
SYSTEM DESIGN	7	DFD LVL 0	ALL	1 DAY	GRAPHICAL REPRESENTATIO N OF DATA FLOW	2
	8	DFD LVL 1	ALL	2 DAY	GRAPHICAL REPRESENTATIO N OF DATA FLOW	2
	9	ENTITY RELATIONSHIP DIAGRAM	JAVIER	2 DAY	GRAPHICAL REPRESENTATIO N OF ENTITY RELATIONSHIPS	2
	10	DATA DICTIONARY	DIDA-AGUN	2 DAY	STRUCTURED TABLE OF THE DATABASE	2
	11	USER INTERFACE	CARPIO	3 WEEKS	INTERFACE LAYOUT	3
TESTING	12	BETA TEST	ALL	3 DAY	INTRODUCE IT TO A LARGER GROUP , GET FEEDBACK AND REPORT BUGS	3
	13	BUG FIX	ALL	5 DAYS	FIXED ERRORS, MEET ENDS WITH CUSTOMER/USER SUGGESTIONS	1
DEPLOYM ENT	14	LAUNCH	ALL	1 DAY	LAUNCHED SYSTEM	1
MAINTENA NCE	15	MAINTENANCE	ALL	CONTINU OUS	MINOR ERROR CHECKUPS	3

D. Task Dependencies

Table A.3 Task Dependencies

Tasks	Assigned Members
Chapter 1 **Introduction**	Carpio Dela Cruz Diaz Dida-agun Javier Villegas
Chapter 2 **Software Needs Validation**	Dela Cruz Javier Villegas
Chapter 3 **Software Design Evaluation** **A. External Interface Requirements** **B. Non Functional Requirement**	Dela Cruz Javier

Chapter 4 **Technology and Techniques Evaluation** **A. Abstract** **B. Introduction** **C. RRL** **D. Methodology** **E. Results and Discussion** **F. Conclusion and Recommendation**	Carpio Villegas
Chapter 5 **Software Design Description** **A. System Architecture** **B. Constraints** **C. Functional View** **D. DFD** **E. System Flowchart** **F. ERD** **G. Data Structure**	Javier Villegas

Chapter 6 **Software Testing and Evaluation** **A. Test Approach** **B. Test Plan** **C. Testing**	Dela Cruz Villegas
Chapter 7 **Conclusion and Recommendation**	Dela Cruz
Appendix **A. Software Process Model** **B. Roles And Responsibilities** **C. Time Table** **D. Task Dependencies** **E. Resources Needed** **F. Gantt Chart**	Dela Cruz Javier Villegas

E. Gantt Chart

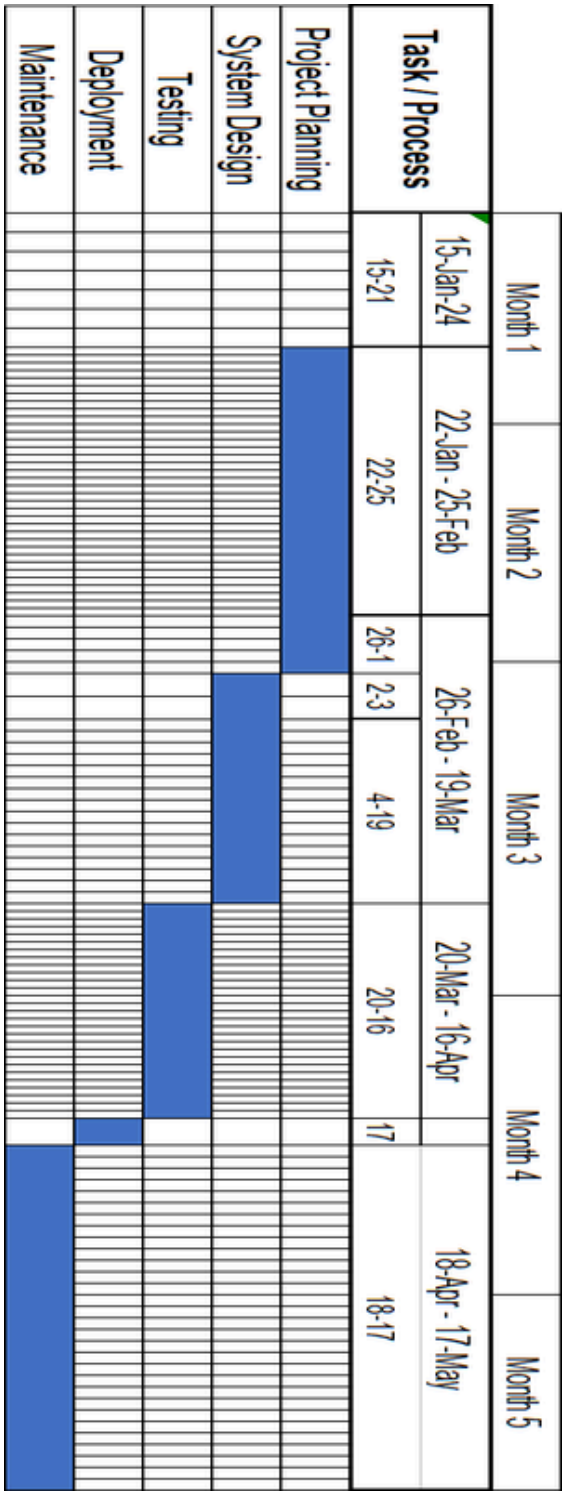
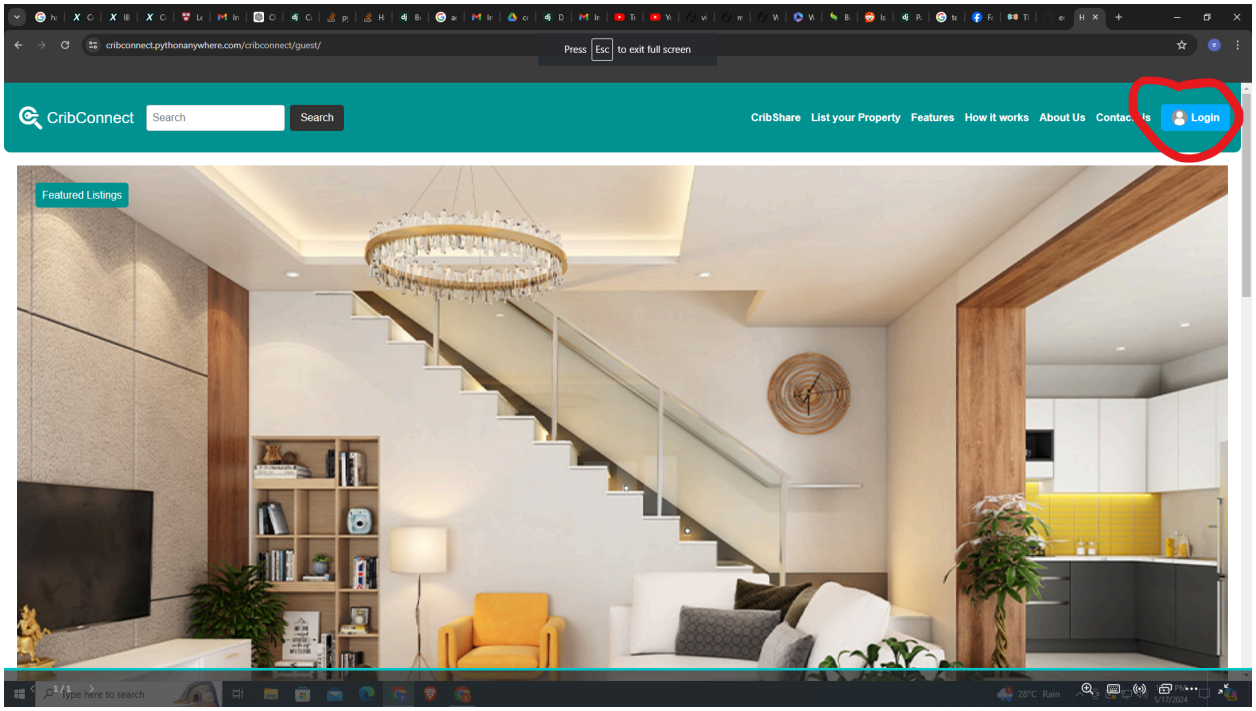
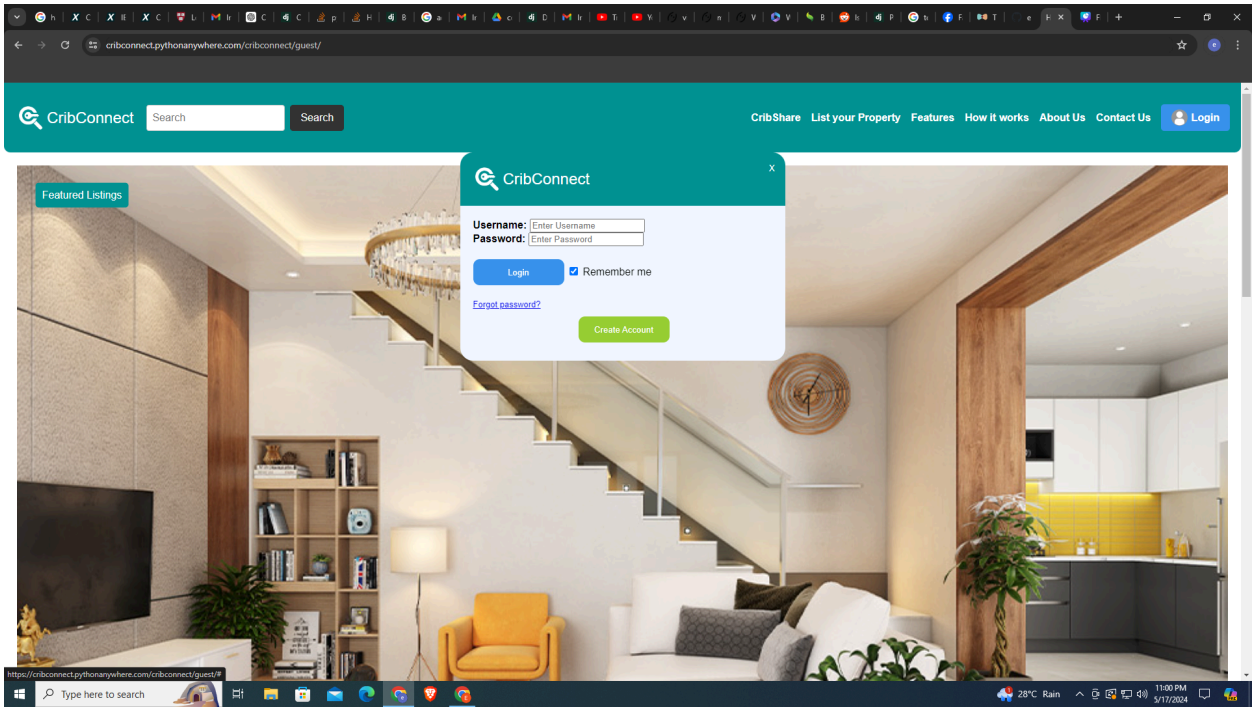


Figure A.2 Gantt Chart

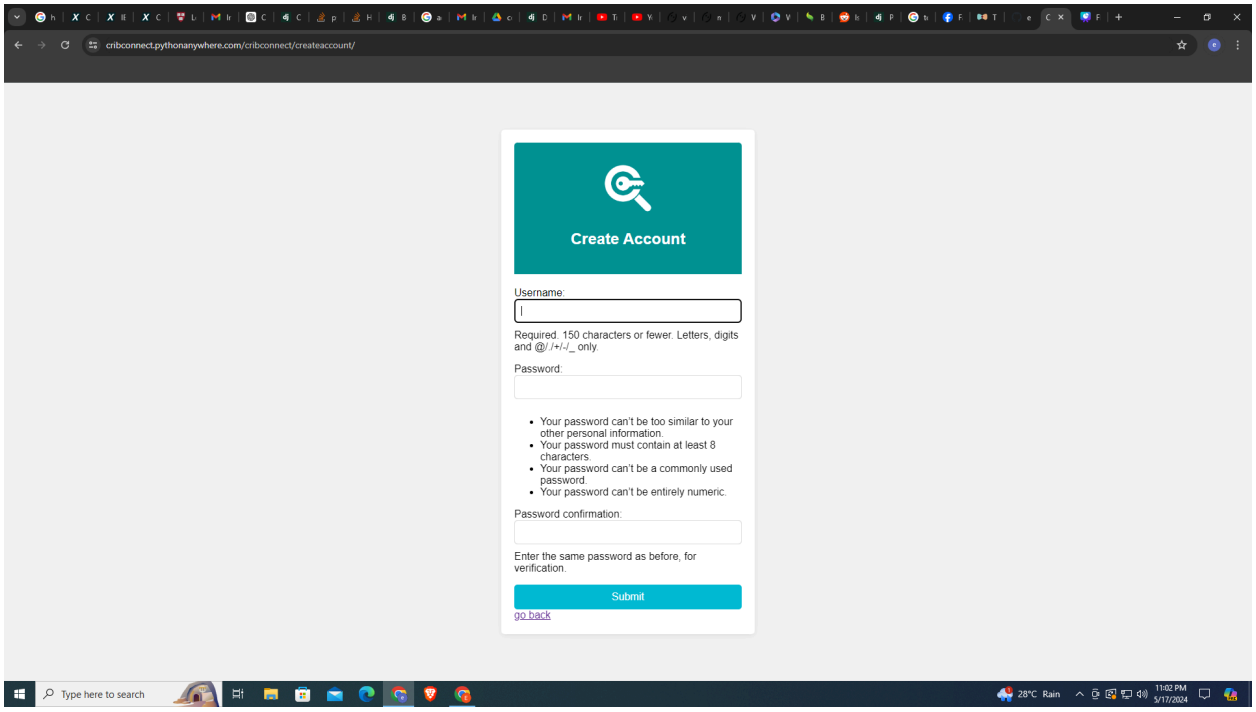
Appendix B: User's Manual



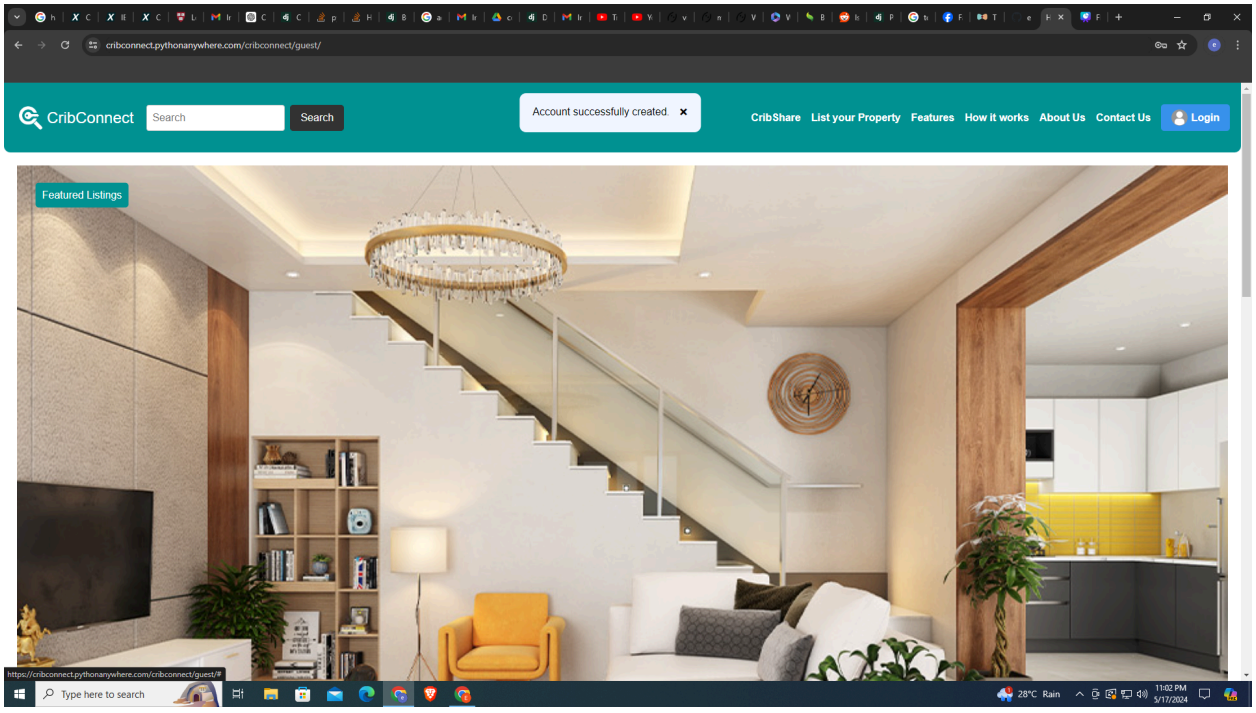
On the homepage click the login button on the upper right corner.



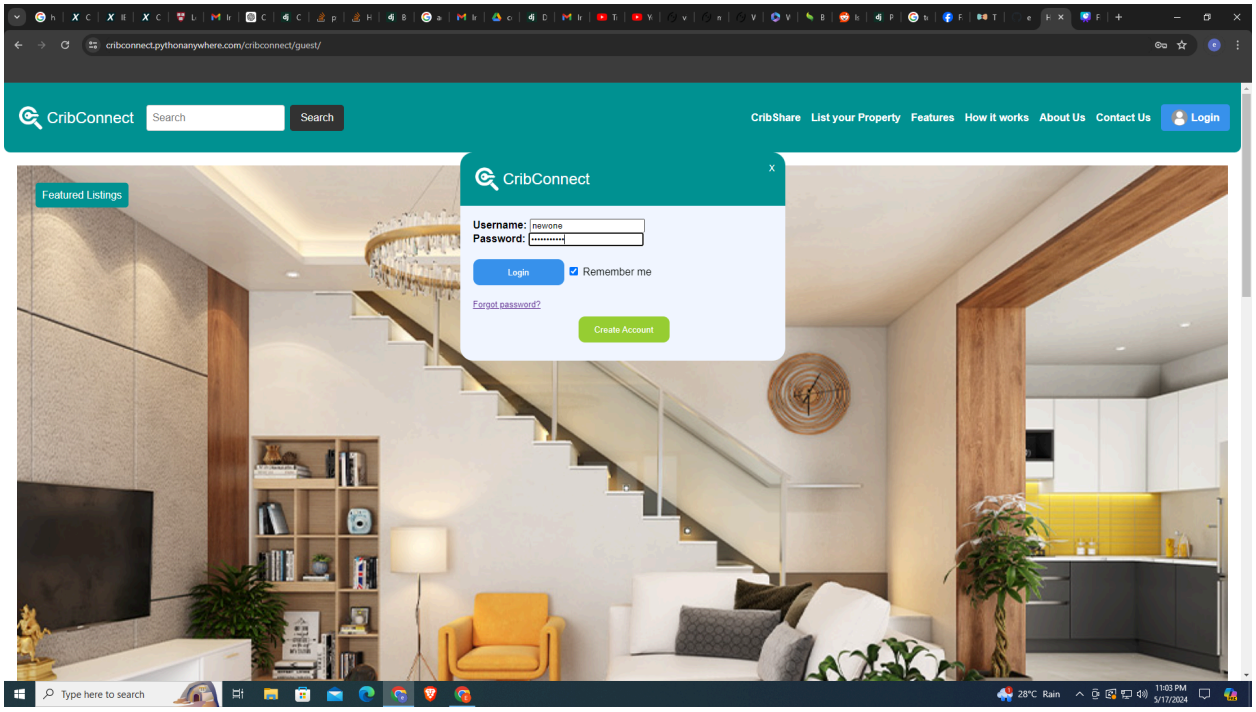
Create an Account by clicking the button “Create Account”.



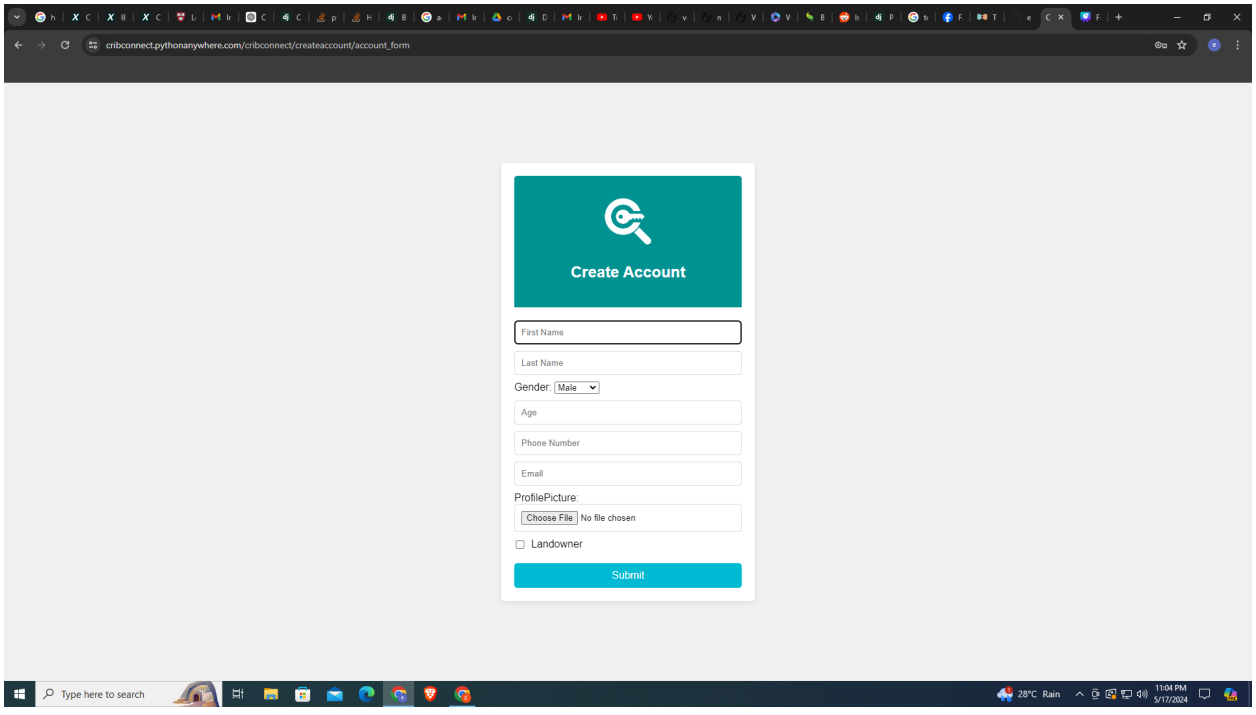
Enter your desired username and password. Make sure to match the password confirmation. Click submit.



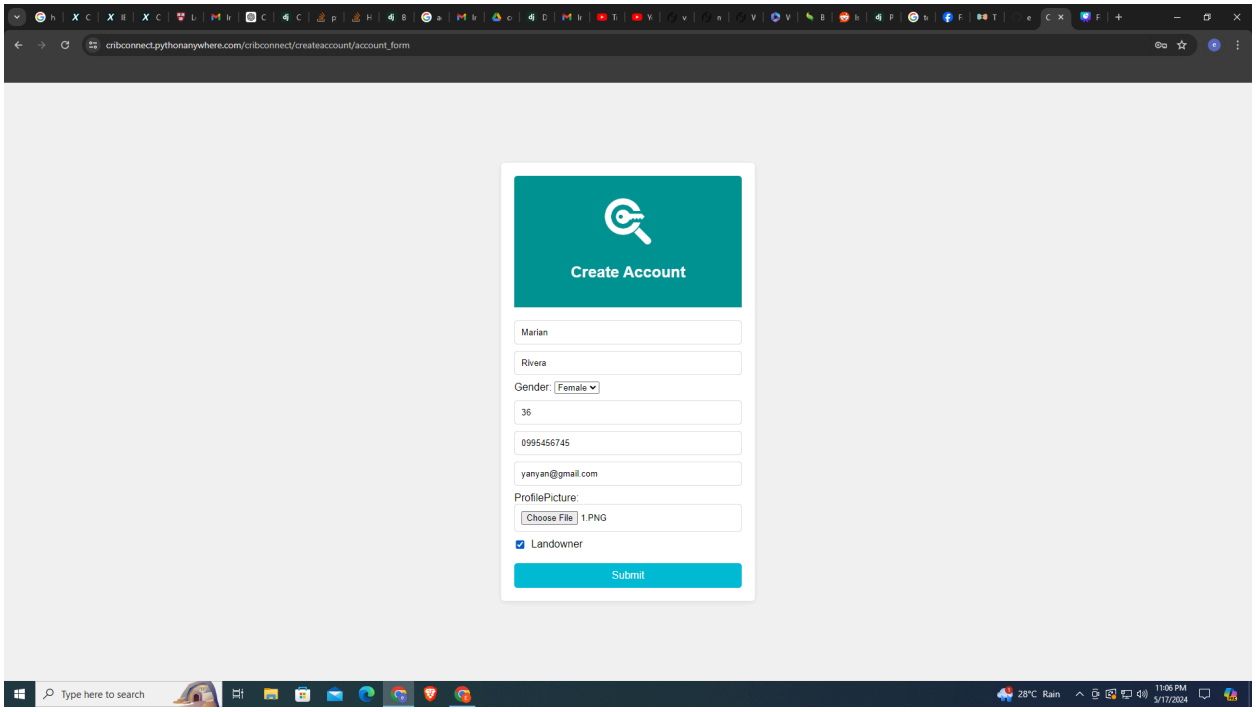
When an account is created, click the “Login” button and input username and password that was created earlier.



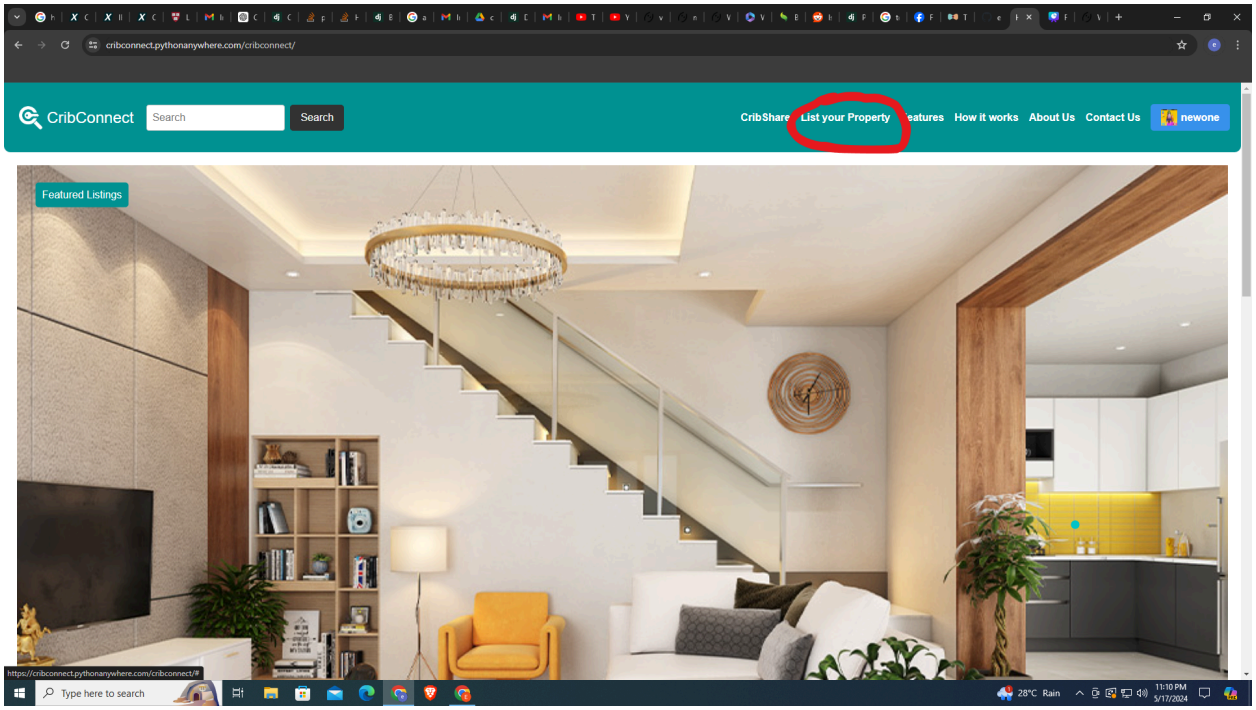
Click the login button. The user will be redirected to the filling up of information form



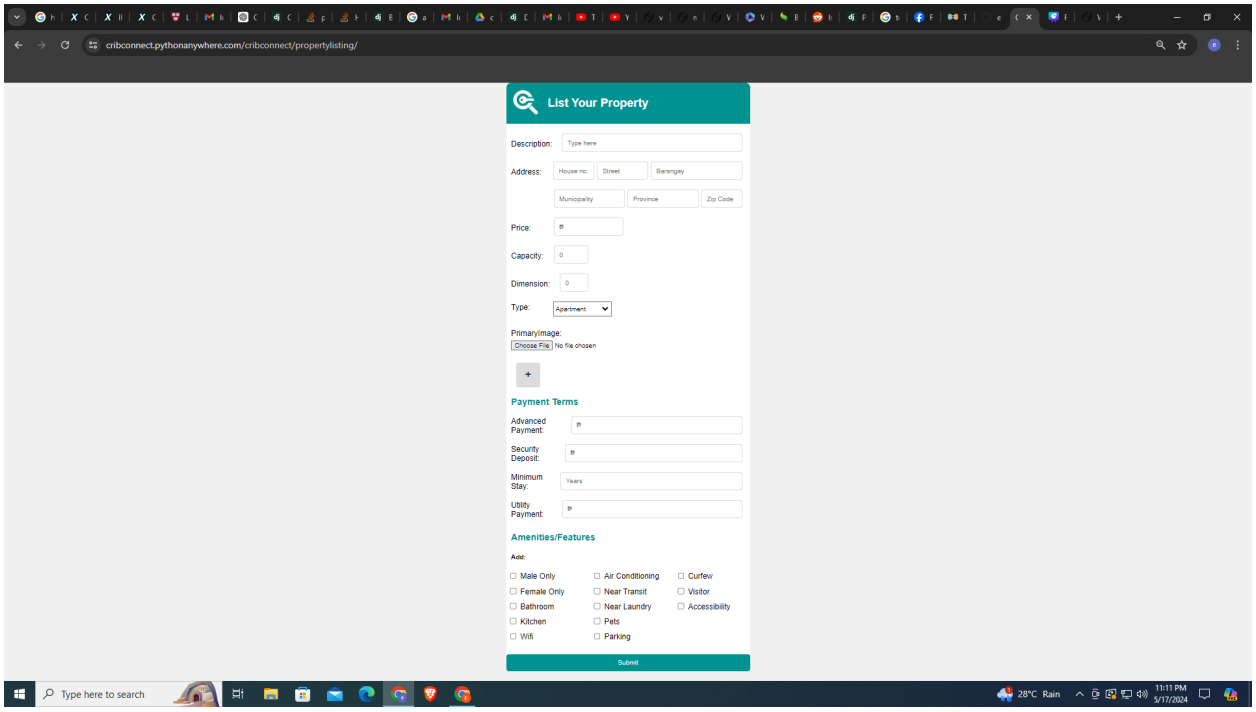
Fill up the information form.



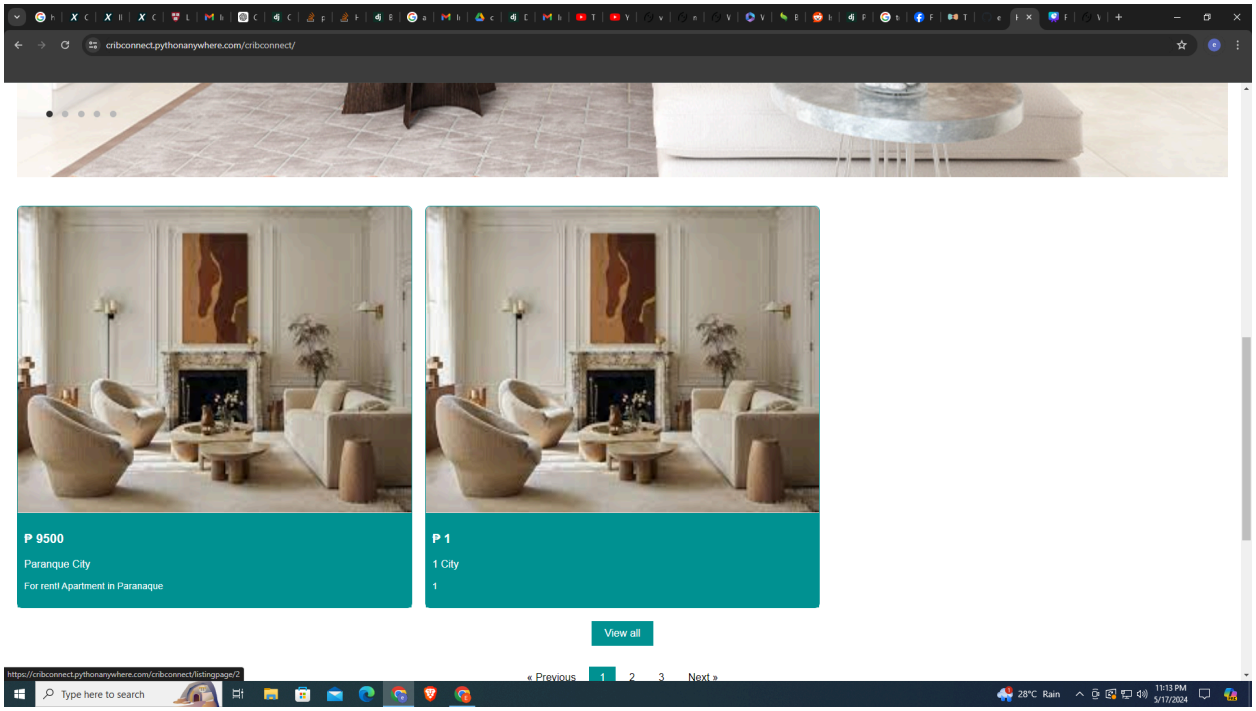
Check “Landowner” if landowner.



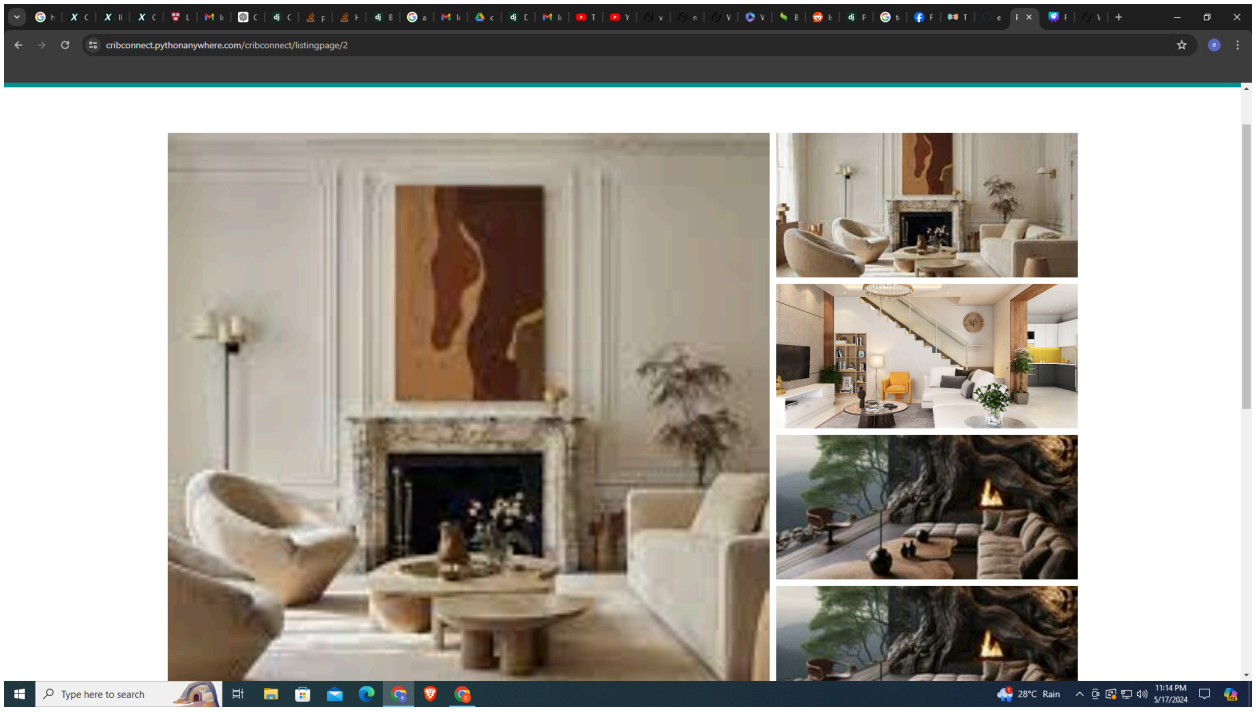
Once logged in, click list your property to create a listing



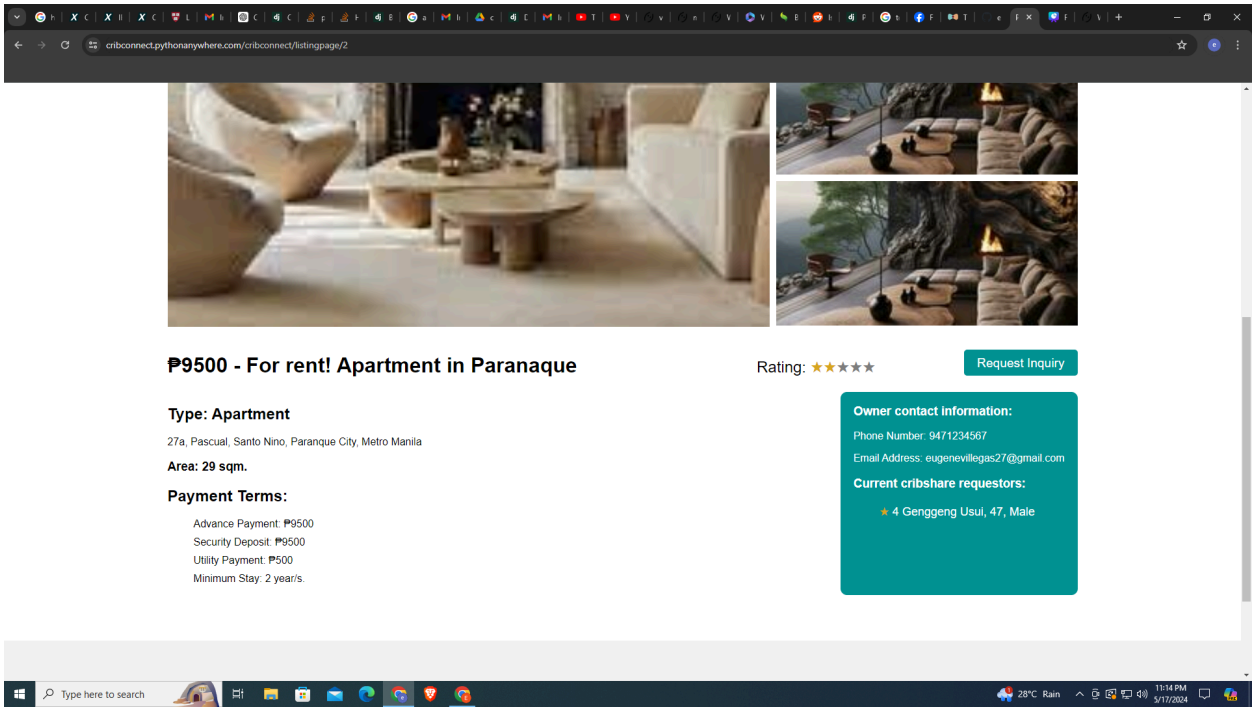
Fill up the required information about the listing that the landowner wants to be displayed.



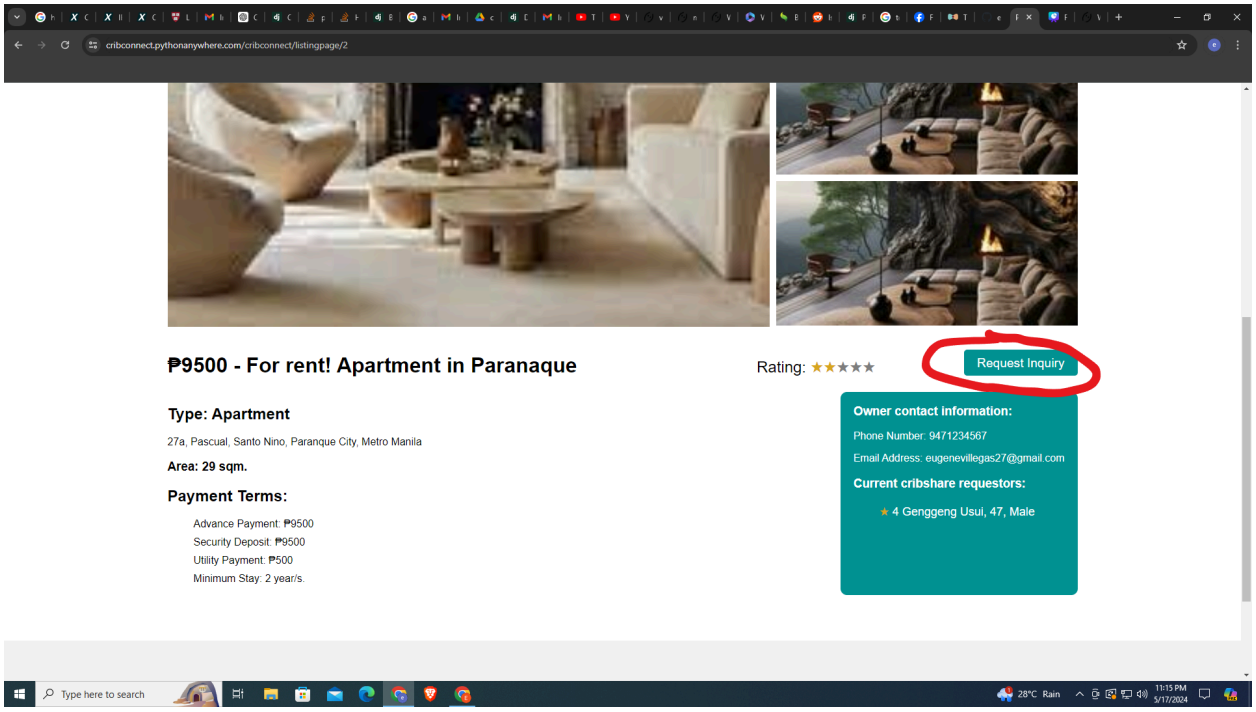
Once the listing has been created, go back to the menu to see if the listing has been displayed to the menu. Click the listing you want to view.



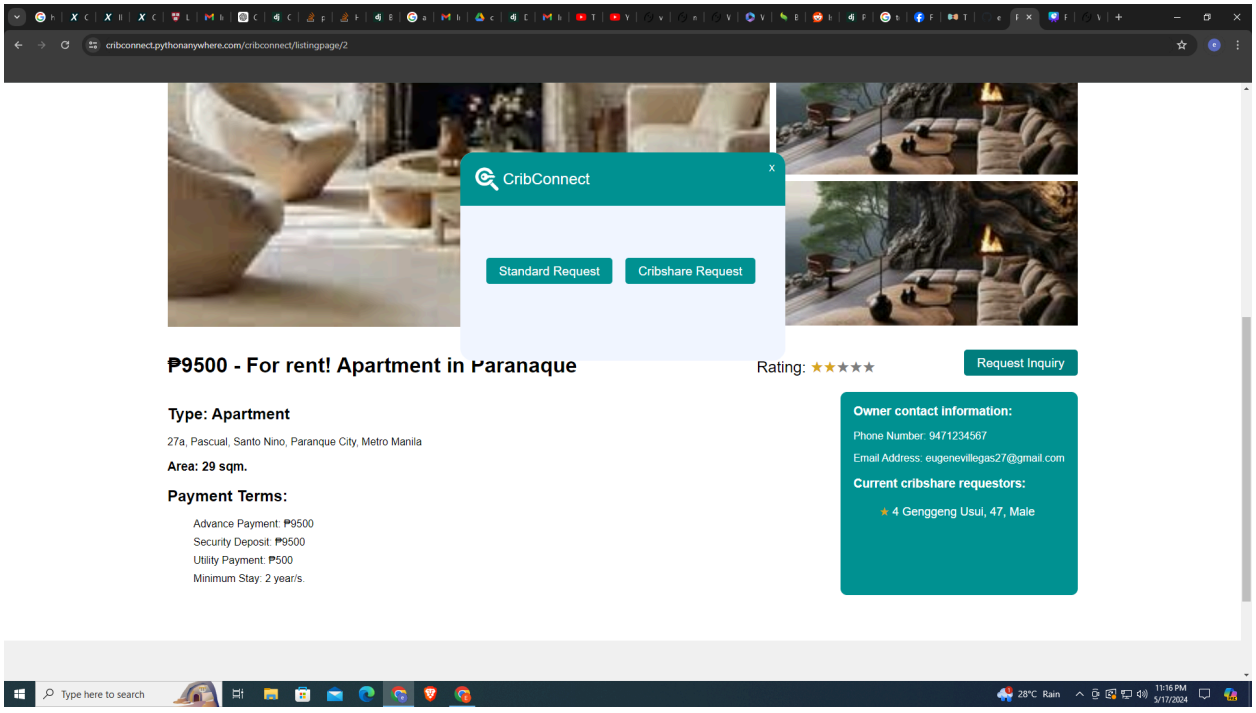
Scroll down to the bottom to check details



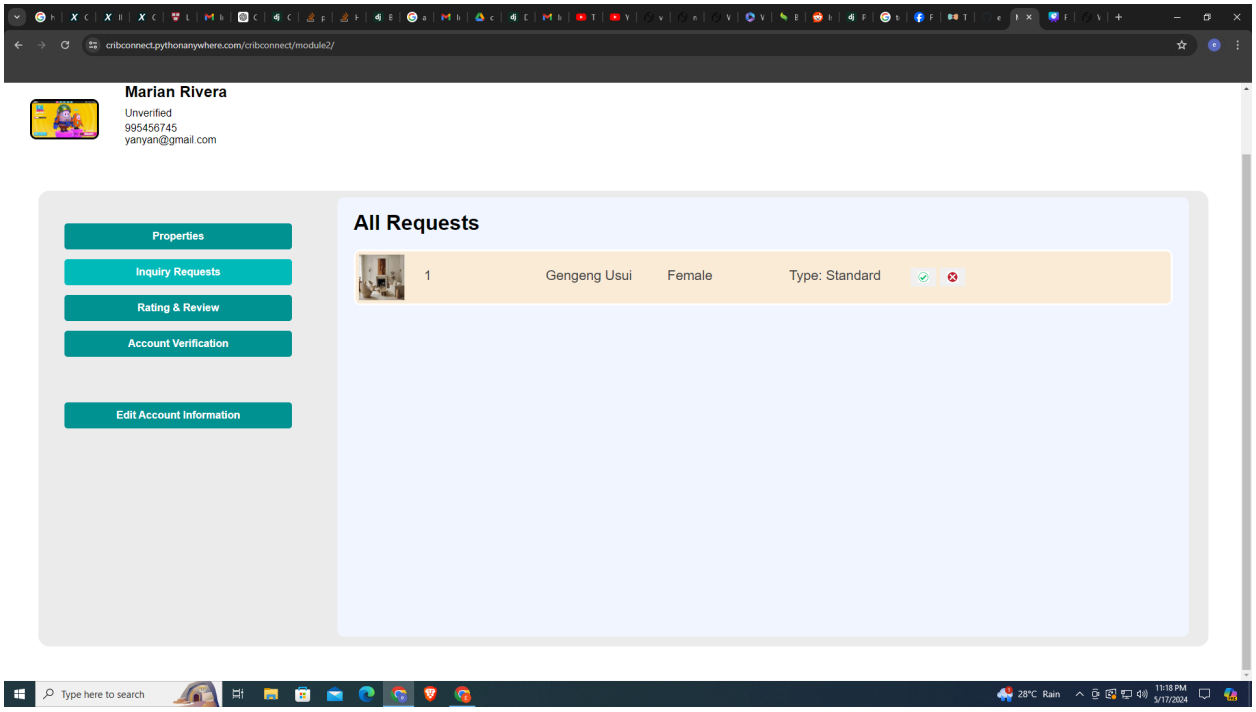
This is the overview of the listing together with the rating of the listing and the tenants that are currently living in the property.



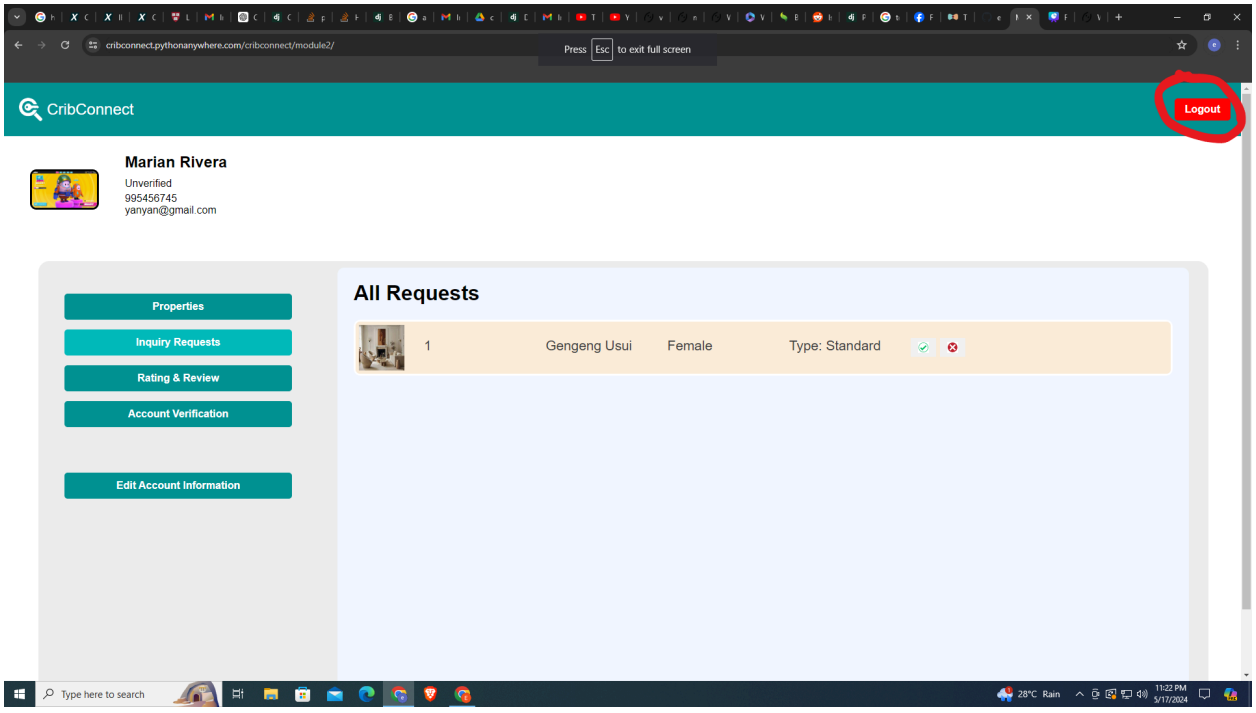
By clicking the “Request Inquiry” button an option will pop up.



Apartment searchers can select between cribshare or standard request.



After receiving the request, landowners can accept or decline by clicking the buttons beside the request details.



To log out, simply click the red button on the upper right corner.

Appendix C: Actual Testing Result

Respondent 1: Che Dela Rosa (landowner)

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users			✓	
Availability	The system should be available for web and mobile with the minimum required system		✓		
Correctness	The system should function as its purpose				✓
Portability	The users should be able to use the system anywhere				
Usability	The effectiveness of the system should e			✓	
Reliability	The information of the listing should be accurate and true			✓	

Respondent 2: Lulu Paguia (landowner)

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users		✓		
Availability	The system should be available for web and mobile with the minimum required system		✓		
Correctness	The system should function as its purpose			✓	
Portability	The users should be able to use the system anywhere			✓	
Usability	The effectiveness of the system should e			✓	
Reliability	The information of the listing should be accurate and true			✓	

Respondent 3: Marilou Co (landowner)

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users			✓	
Availability	The system should be available for web and mobile with the minimum required system			✓	
Correctness	The system should function as its purpose			✓	
Portability	The users should be able to use the system anywhere				✓
Usability	The effectiveness of the system should e				✓
Reliability	The information of the listing should be accurate and true				✓

Respondent 4: Czar Mercado

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users			✓	
Availability	The system should be available for web and mobile with the minimum required system				✓
Correctness	The system should function as its purpose			✓	
Portability	The users should be able to use the system anywhere			✓	
Usability	The effectiveness of the system should e		✓		
Reliability	The information of the listing should be accurate and true		✓		

Respondent 5: Jett Espiritu

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users			✓	
Availability	The system should be available for web and mobile with the minimum required system			✓	
Correctness	The system should function as its purpose		✓		
Portability	The users should be able to use the system anywhere			✓	
Usability	The effectiveness of the system should e				✓
Reliability	The information of the listing should be accurate and true			✓	

Respondent 6: Tyra Prix Zaraspe

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users			✓	
Availability	The system should be available for web and mobile with the minimum required system			✓	
Correctness	The system should function as its purpose			✓	
Portability	The users should be able to use the system anywhere			✓	
Usability	The effectiveness of the system should e			✓	
Reliability	The information of the listing should be accurate and true			✓	

Respondent 7: Tristan Yanoria

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users				✓
Availability	The system should be available for web and mobile with the minimum required system			✓	
Correctness	The system should function as its purpose			✓	
Portability	The users should be able to use the system anywhere			✓	
Usability	The effectiveness of the system should e			✓	
Reliability	The information of the listing should be accurate and true				✓

Respondent 8: Genaro Carigma (landowner)

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users		✓		
Availability	The system should be available for web and mobile with the minimum required system			✓	
Correctness	The system should function as its purpose			✓	
Portability	The users should be able to use the system anywhere				✓
Usability	The effectiveness of the system should e			✓	
Reliability	The information of the listing should be accurate and true			✓	

Respondent 9: Efraim Baculi

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users			✓	
Availability	The system should be available for web and mobile with the minimum required system			✓	
Correctness	The system should function as its purpose			✓	
Portability	The users should be able to use the system anywhere			✓	
Usability	The effectiveness of the system should e				✓
Reliability	The information of the listing should be accurate and true				✓

Respondent 10: Justin Victolero

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users				✓
Availability	The system should be available for web and mobile with the minimum required system				✓
Correctness	The system should function as its purpose				✓
Portability	The users should be able to use the system anywhere				✓
Usability	The effectiveness of the system should e				✓
Reliability	The information of the listing should be accurate and true				✓

Respondent 11: Jho-annes Victolero

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users			✓	
Availability	The system should be available for web and mobile with the minimum required system			✓	
Correctness	The system should function as its purpose				✓
Portability	The users should be able to use the system anywhere				✓
Usability	The effectiveness of the system should e			✓	
Reliability	The information of the listing should be accurate and true			✓	

Respondent 12: Gallen Rose Blosa

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users			✓	
Availability	The system should be available for web and mobile with the minimum required system			✓	
Correctness	The system should function as its purpose		✓		
Portability	The users should be able to use the system anywhere		✓		
Usability	The effectiveness of the system should e		✓		
Reliability	The information of the listing should be accurate and true		✓		

Respondent 13: RC Lloyd Concepcion

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users			✓	
Availability	The system should be available for web and mobile with the minimum required system			✓	
Correctness	The system should function as its purpose			✓	
Portability	The users should be able to use the system anywhere				✓
Usability	The effectiveness of the system should e				✓
Reliability	The information of the listing should be accurate and true				✓

Respondent 14: Felicity Kaye Paulino

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users			✓	
Availability	The system should be available for web and mobile with the minimum required system			✓	
Correctness	The system should function as its purpose				✓
Portability	The users should be able to use the system anywhere			✓	
Usability	The effectiveness of the system should e				✓
Reliability	The information of the listing should be accurate and true			✓	

Respondent 15: Ernesto Calmante

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users				✓
Availability	The system should be available for web and mobile with the minimum required system			✓	
Correctness	The system should function as its purpose		✓		
Portability	The users should be able to use the system anywhere				✓
Usability	The effectiveness of the system should e				✓
Reliability	The information of the listing should be accurate and true			✓	

Respondent 16: Necisita Dela Cruz (landowner)

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users			✓	
Availability	The system should be available for web and mobile with the minimum required system		✓		
Correctness	The system should function as its purpose			✓	
Portability	The users should be able to use the system anywhere		✓		
Usability	The effectiveness of the system should e			✓	
Reliability	The information of the listing should be accurate and true			✓	

Respondent 17: Auryl Marco Javier

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users			✓	
Availability	The system should be available for web and mobile with the minimum required system			✓	
Correctness	The system should function as its purpose				✓
Portability	The users should be able to use the system anywhere				✓
Usability	The effectiveness of the system should e			✓	
Reliability	The information of the listing should be accurate and true				✓

Respondent 18: Shey Dangupon

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users			✓	
Availability	The system should be available for web and mobile with the minimum required system			✓	
Correctness	The system should function as its purpose			✓	
Portability	The users should be able to use the system anywhere			✓	
Usability	The effectiveness of the system should e		✓		
Reliability	The information of the listing should be accurate and true			✓	

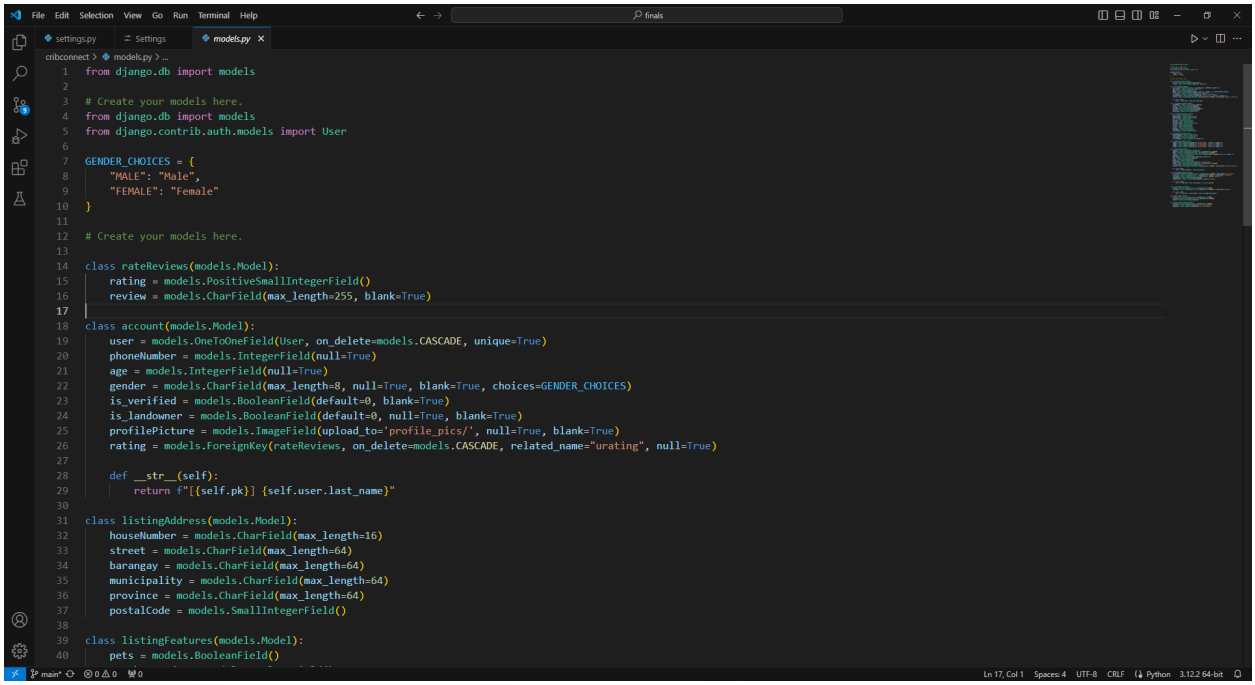
Respondent 19: Ruff Soriano

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users		✓		
Availability	The system should be available for web and mobile with the minimum required system			✓	
Correctness	The system should function as its purpose			✓	
Portability	The users should be able to use the system anywhere			✓	
Usability	The effectiveness of the system should e			✓	
Reliability	The information of the listing should be accurate and true		✓		

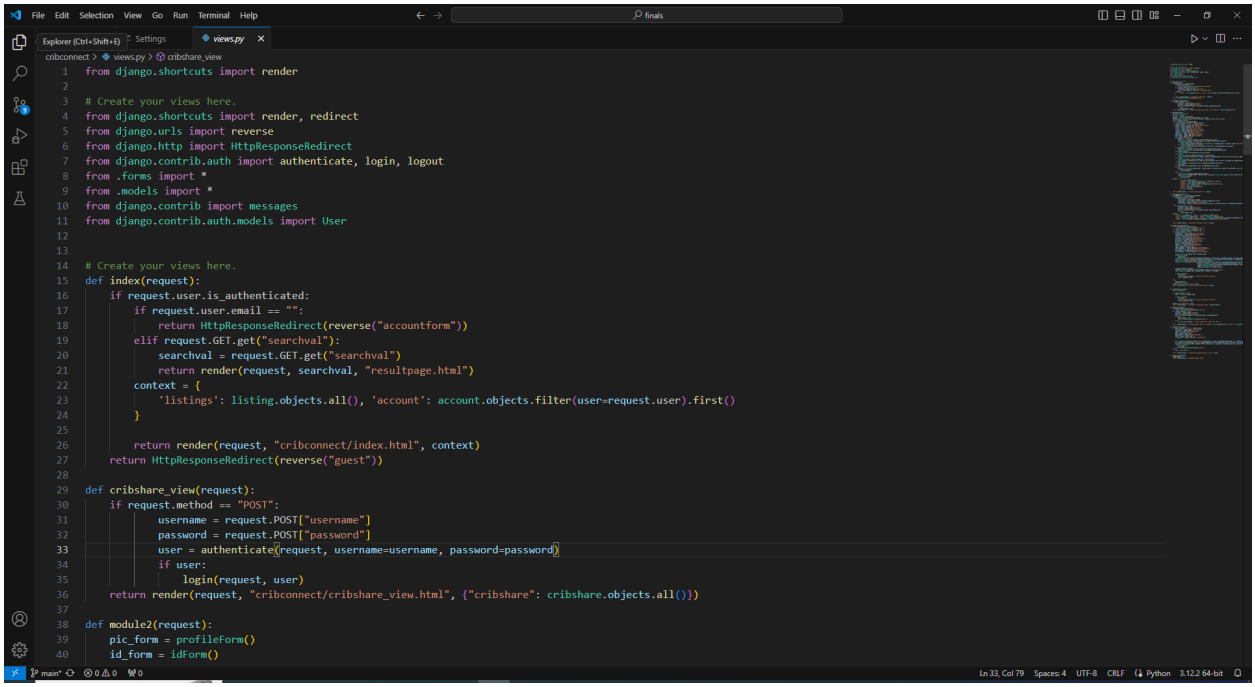
Respondent 20: Rashmin Driza

Attributes	Description	1 (poor)	2 (fair)	3 (good)	4 (very good)
Acceptability	The user interface and the functionality of the system must be user friendly and easy to understand for the target users			✓	
Availability	The system should be available for web and mobile with the minimum required system			✓	
Correctness	The system should function as its purpose			✓	
Portability	The users should be able to use the system anywhere			✓	
Usability	The effectiveness of the system should e				✓
Reliability	The information of the listing should be accurate and true				✓

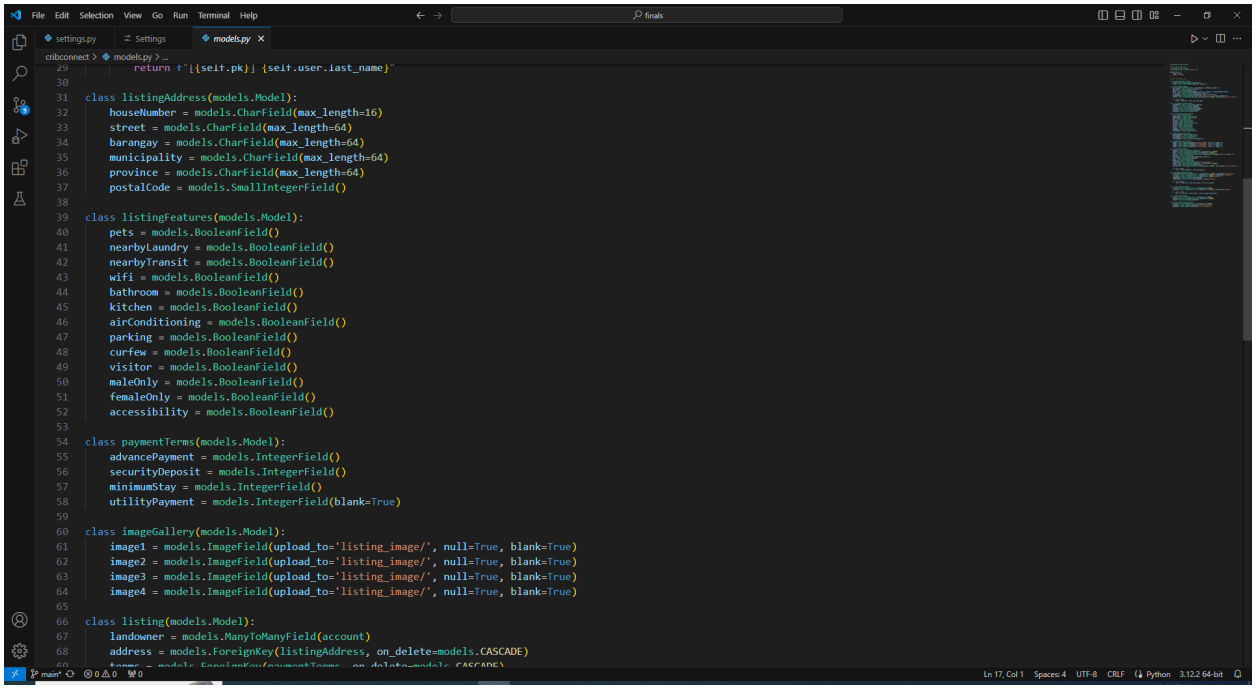
Appendix D: Source Code



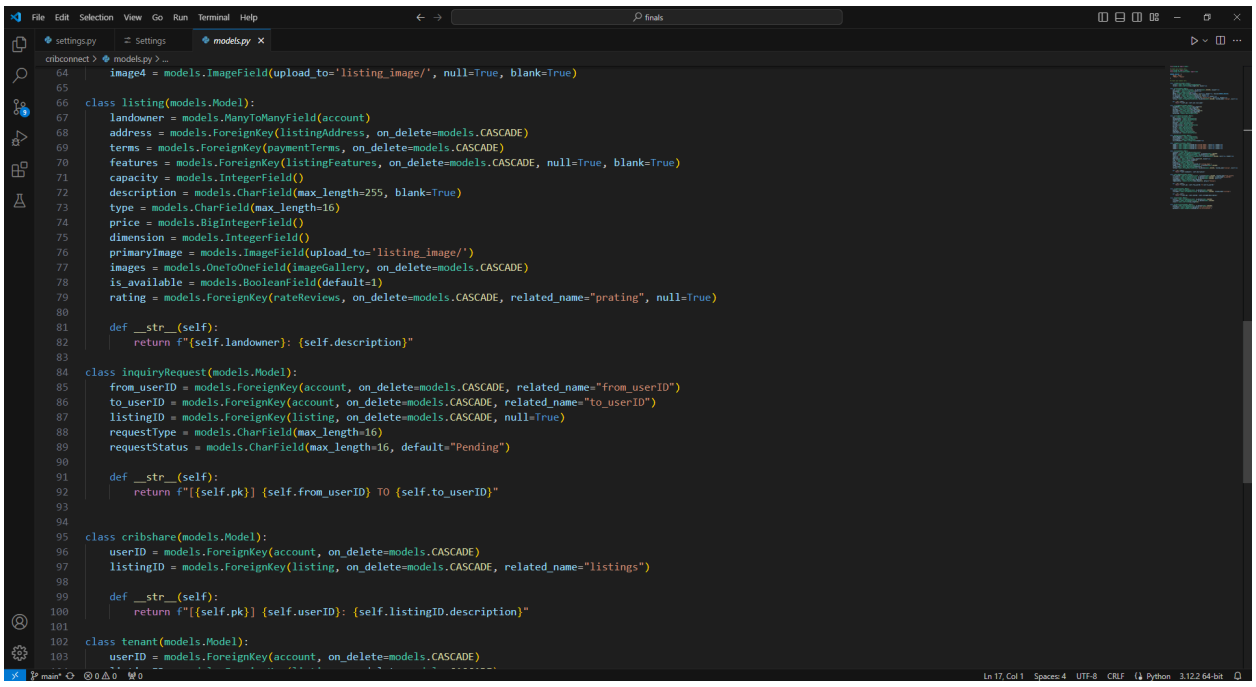
```python
from django.db import models

# Create your models here.
from django.db import models
from django.contrib.auth.models import User

GENDER_CHOICES = {
    "MALE": "Male",
    "FEMALE": "Female"
}

# Create your models here.

class rateReviews(models.Model):
    rating = models.PositiveSmallIntegerField()
    review = models.CharField(max_length=255, blank=True)

class account(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE, unique=True)
    phoneNumber = models.IntegerField(null=True)
    age = models.IntegerField(null=True)
    gender = models.CharField(max_length=8, null=True, blank=True, choices=GENDER_CHOICES)
    is_verified = models.BooleanField(default=0, blank=True)
    is_landowner = models.BooleanField(default=0, null=True, blank=True)
    profilePicture = models.ImageField(upload_to='profile_pics/', null=True, blank=True)
    rating = models.ForeignKey(rateReviews, on_delete=models.CASCADE, related_name="urating", null=True)

    def __str__(self):
        return f"[{self.pk}] {self.user.last_name}"

class listingAddress(models.Model):
    houseNumber = models.CharField(max_length=16)
    street = models.CharField(max_length=64)
    barangay = models.CharField(max_length=64)
    municipality = models.CharField(max_length=64)
    province = models.CharField(max_length=64)
    postalCode = models.SmallIntegerField()

class listingFeatures(models.Model):
    pets = models.BooleanField()
```



```python
from django.shortcuts import render

# Create your views here.
from django.shortcuts import render, redirect
from django.urls import reverse
from django.http import HttpResponseRedirect
from django.contrib.auth import authenticate, login, logout
from .forms import *
from .models import *
from django.contrib import messages
from django.contrib.auth.models import User


# Create your views here.
def index(request):
    if request.user.is_authenticated:
        if request.user.email == "":
            return HttpResponseRedirect(reverse("accountform"))
        elif request.GET.get("searchval"):
            searchval = request.GET.get("searchval")
            return render(request, searchval, "resultpage.html")
        context = {
            'listings': listing.objects.all(), 'account': account.objects.filter(user=request.user).first()
        }

        return render(request, "cribconnect/index.html", context)
    return HttpResponseRedirect(reverse("guest"))

def cribshare_view(request):
    if request.method == "POST":
            username = request.POST["username"]
            password = request.POST["password"]
            user = authenticate(request, username=username, password=password)
            if user:
                login(request, user)
    return render(request, "cribconnect/cribshare_view.html", {"cribshare": cribshare.objects.all()})

def module2(request):
    pic_form = profileForm()
    id_form = idForm()
```

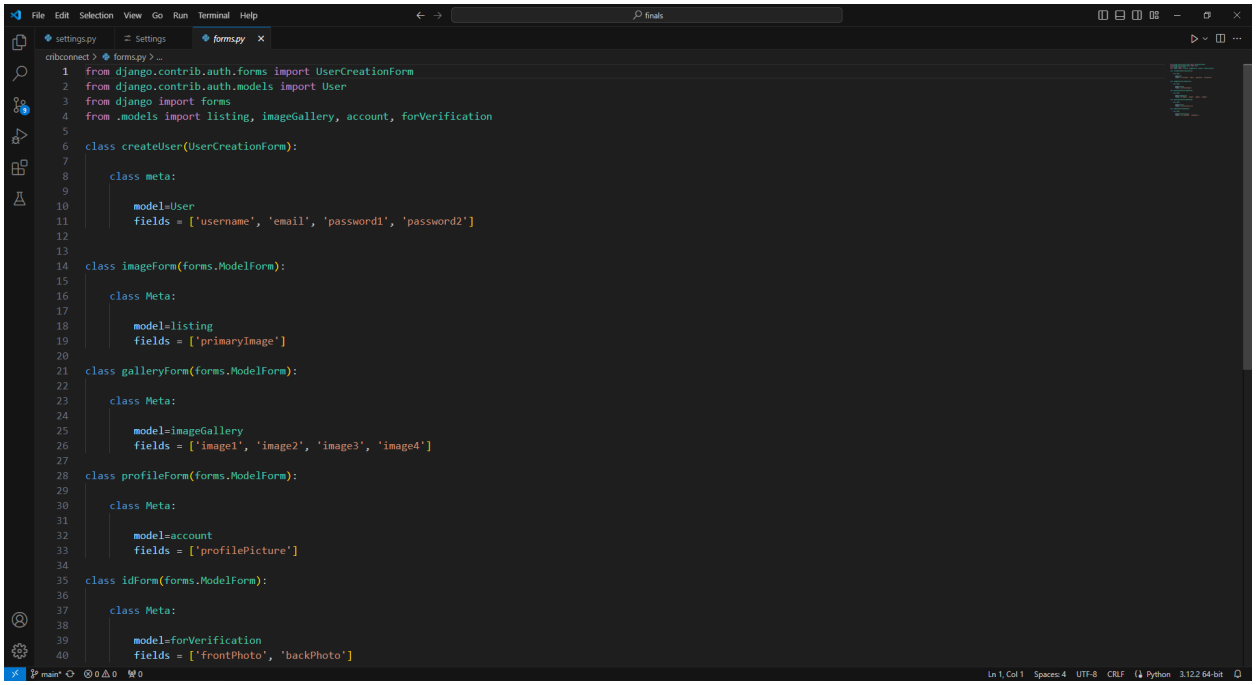


```
        return f"{self.pk} {self.user.last_name}"

class listingAddress(models.Model):
    houseNumber = models.CharField(max_length=16)
    street = models.CharField(max_length=64)
    barangay = models.CharField(max_length=64)
    municipality = models.CharField(max_length=64)
    province = models.CharField(max_length=64)
    postalCode = models.SmallIntegerField()

class listingFeatures(models.Model):
    pets = models.BooleanField()
    nearbyLaundry = models.BooleanField()
    nearbyTransit = models.BooleanField()
    wifi = models.BooleanField()
    bathroom = models.BooleanField()
    kitchen = models.BooleanField()
    airConditioning = models.BooleanField()
    parking = models.BooleanField()
    curfew = models.BooleanField()
    visitor = models.BooleanField()
    maleOnly = models.BooleanField()
    femaleOnly = models.BooleanField()
    accessibility = models.BooleanField()

class paymentTerms(models.Model):
    advancePayment = models.IntegerField()
    securityDeposit = models.IntegerField()
    minimumStay = models.IntegerField()
    utilityPayment = models.IntegerField(blank=True)

class imageGallery(models.Model):
    image1 = models.ImageField(upload_to='listing_image/', null=True, blank=True)
    image2 = models.ImageField(upload_to='listing_image/', null=True, blank=True)
    image3 = models.ImageField(upload_to='listing_image/', null=True, blank=True)
    image4 = models.ImageField(upload_to='listing_image/', null=True, blank=True)

class listing(models.Model):
    landowner = models.ManyToManyField(account)
    address = models.ForeignKey(listingAddress, on_delete=models.CASCADE)
```

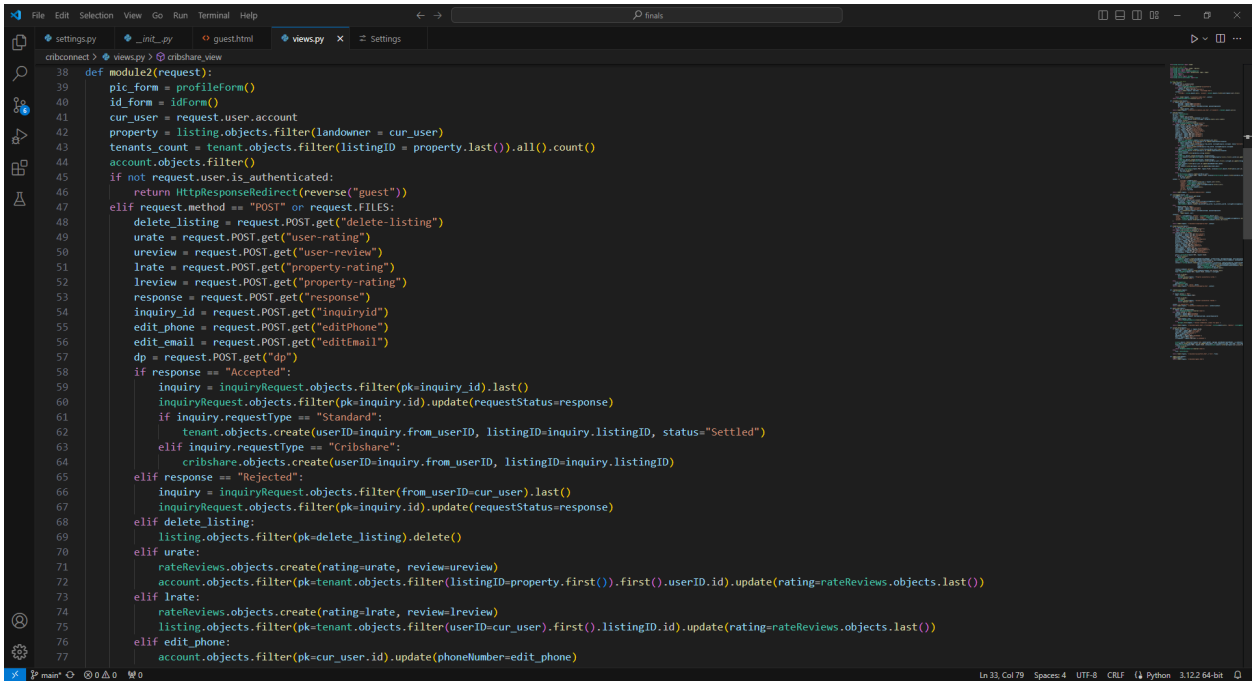


```
    image4 = models.ImageField(upload_to='listing_image/', null=True, blank=True)

class listing(models.Model):
    landowner = models.ManyToManyField(account)
    address = models.ForeignKey(listingAddress, on_delete=models.CASCADE)
    terms = models.ForeignKey(paymentTerms, on_delete=models.CASCADE)
    features = models.ForeignKey(listingFeatures, on_delete=models.CASCADE, null=True, blank=True)
    capacity = models.IntegerField()
    description = models.CharField(max_length=255, blank=True)
    type = models.CharField(max_length=16)
    price = models.BigIntegerField()
    dimension = models.IntegerField()
    primaryImage = models.ImageField(upload_to='listing_image/')
    images = models.OneToOneField(imageGallery, on_delete=models.CASCADE)
    is_available = models.BooleanField(default=1)
    rating = models.ForeignKey(rateReviews, on_delete=models.CASCADE, related_name="prating", null=True)

    def __str__(self):
        return f"{self.landowner}: {self.description}"

class inquiryRequest(models.Model):
    from_userID = models.ForeignKey(account, on_delete=models.CASCADE, related_name="from_userID")
    to_userID = models.ForeignKey(account, on_delete=models.CASCADE, related_name="to_userID")
    listingID = models.ForeignKey(listing, on_delete=models.CASCADE, null=True)
    requestType = models.CharField(max_length=16)
    requestStatus = models.CharField(max_length=16, default="Pending")

    def __str__(self):
        return f"[{self.pk}] {self.from_userID} TO {self.to_userID}"


class cribshare(models.Model):
    userID = models.ForeignKey(account, on_delete=models.CASCADE)
    listingID = models.ForeignKey(listing, on_delete=models.CASCADE, related_name="listings")

    def __str__(self):
        return f"[{self.pk}] {self.userID}: {self.listingID.description}"

class tenant(models.Model):
    userID = models.ForeignKey(account, on_delete=models.CASCADE)
```

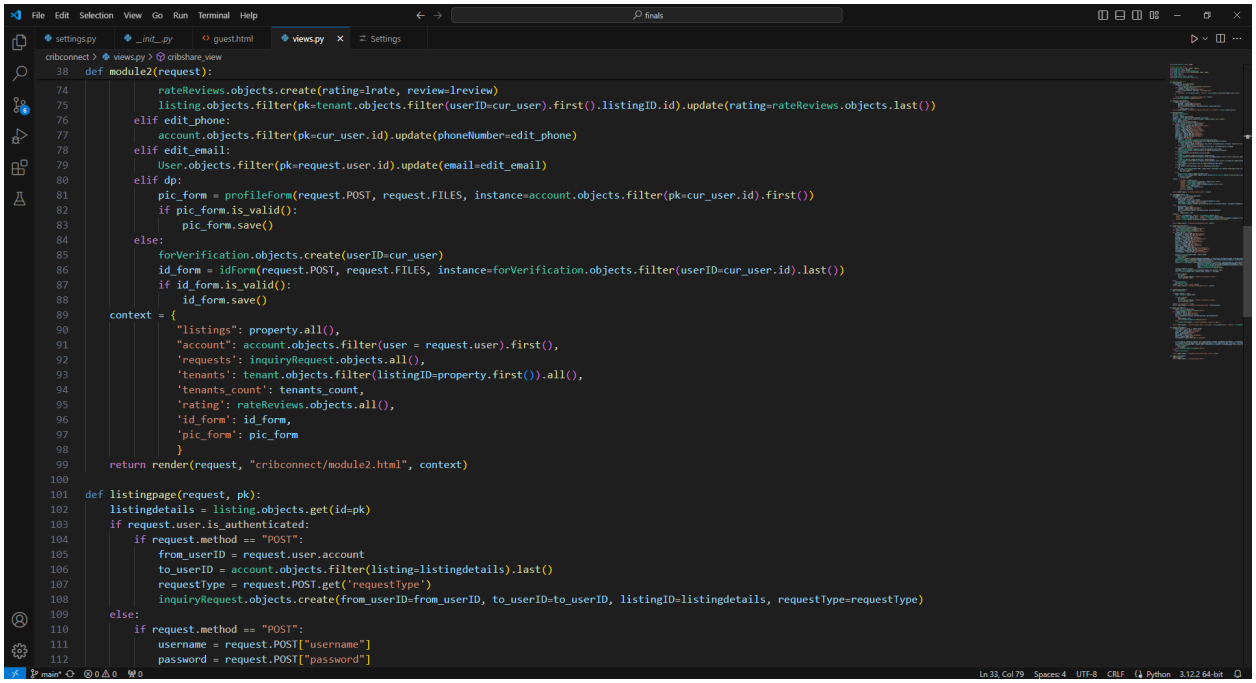
```python
class tenant(models.Model):
    userID = models.ForeignKey(account, on_delete=models.CASCADE)
    listingID = models.ForeignKey(listing, on_delete=models.CASCADE)
    status = models.CharField(max_length=16)

class forVerification(models.Model):
    userID = models.ForeignKey(account, on_delete=models.CASCADE)
    frontPhoto = models.ImageField(upload_to='for_verification/')
    backPhoto = models.ImageField(upload_to='for_verification/')
```



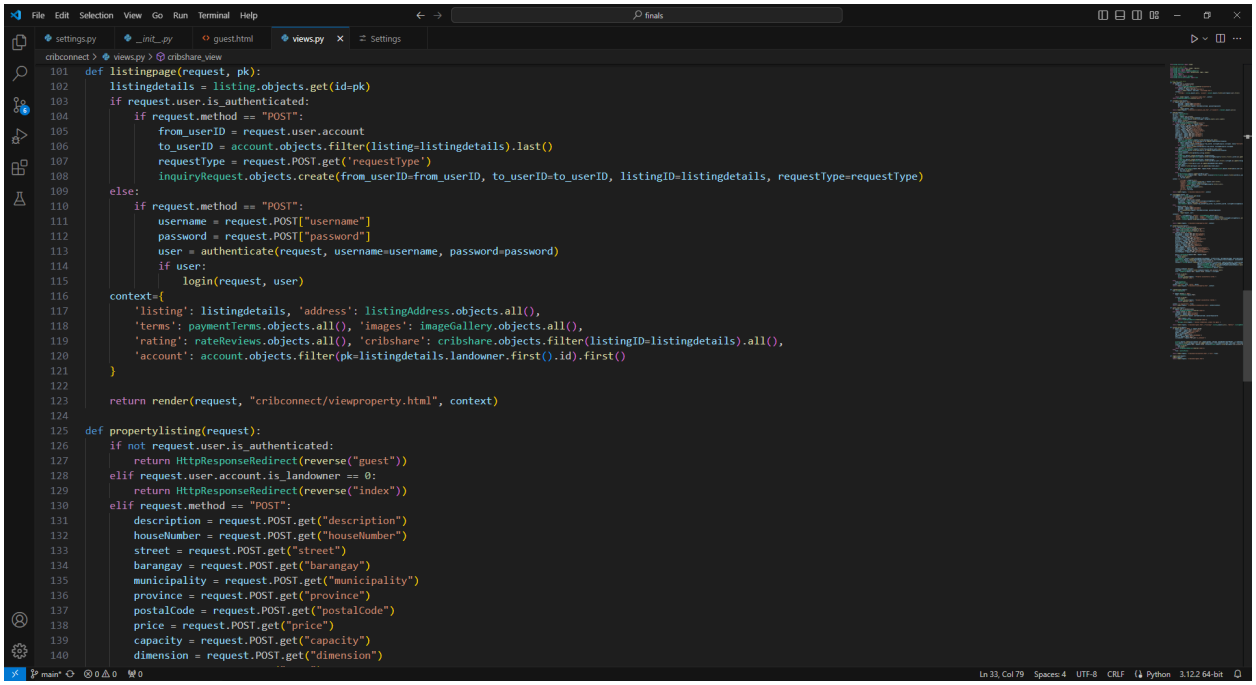
```python
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth.models import User
from django import forms
from .models import listing, imageGallery, account, forVerification

class createUser(UserCreationForm):

    class meta:

        model=User
        fields = ['username', 'email', 'password1', 'password2']


class imageForm(forms.ModelForm):

    class Meta:

        model=listing
        fields = ['primaryImage']

class galleryForm(forms.ModelForm):

    class Meta:

        model=imageGallery
        fields = ['image1', 'image2', 'image3', 'image4']

class profileForm(forms.ModelForm):

    class Meta:

        model=account
        fields = ['profilePicture']

class idForm(forms.ModelForm):

    class Meta:

        model=forVerification
        fields = ['frontPhoto', 'backPhoto']
```



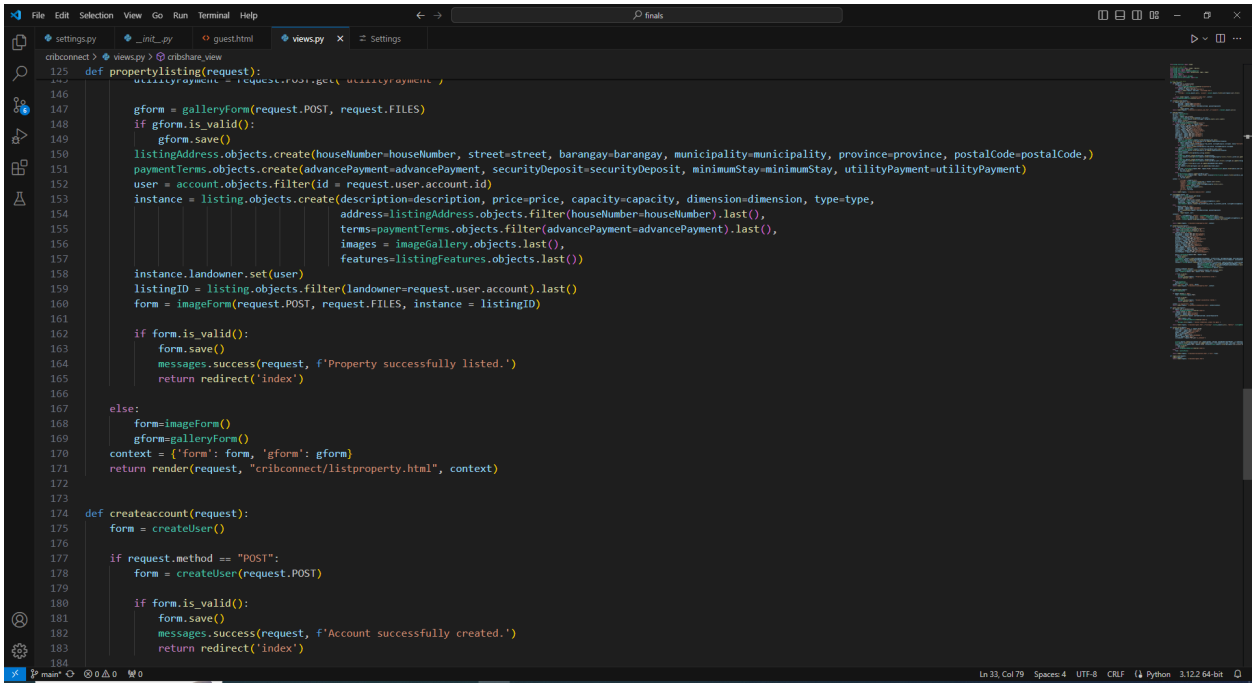
```python
def module2(request):
    pic_form = profileForm()
    id_form = idForm()
    cur_user = request.user.account
    property = listing.objects.filter(landowner = cur_user)
    tenants_count = tenant.objects.filter(listingID = property.last()).all().count()
    account.objects.filter()
    if not request.user.is_authenticated:
        return HttpResponseRedirect(reverse("guest"))
    elif request.method == "POST" or request.FILES:
        delete_listing = request.POST.get("delete-listing")
        urate = request.POST.get("user-rating")
        ureview = request.POST.get("user-review")
        lrate = request.POST.get("property-rating")
        lreview = request.POST.get("property-rating")
        response = request.POST.get("response")
        inquiry_id = request.POST.get("inquiryid")
        edit_phone = request.POST.get("editPhone")
        edit_email = request.POST.get("editEmail")
        dp = request.POST.get("dp")
        if response == "Accepted":
            inquiry = inquiryRequest.objects.filter(pk=inquiry_id).last()
            inquiryRequest.objects.filter(pk=inquiry.id).update(requestStatus=response)
            if inquiry.requestType == "Standard":
                tenant.objects.create(userID=inquiry.from_userID, listingID=inquiry.listingID, status="Settled")
            elif inquiry.requestType == "Cribshare":
                cribshare.objects.create(userID=inquiry.from_userID, listingID=inquiry.listingID)
        elif response == "Rejected":
            inquiry = inquiryRequest.objects.filter(from_userID=cur_user).last()
            inquiryRequest.objects.filter(pk=inquiry.id).update(requestStatus=response)
        elif delete_listing:
            listing.objects.filter(pk=delete_listing).delete()
        elif urate:
            rateReviews.objects.create(rating=urate, review=ureview)
            account.objects.filter(pk=tenant.objects.filter(listingID=property.first()).first().userID.id).update(rating=rateReviews.objects.last())
        elif lrate:
            rateReviews.objects.create(rating=lrate, review=lreview)
            listing.objects.filter(pk=tenant.objects.filter(userID=cur_user).first().listingID.id).update(rating=rateReviews.objects.last())
        elif edit_phone:
            account.objects.filter(pk=cur_user.id).update(phoneNumber=edit_phone)
```



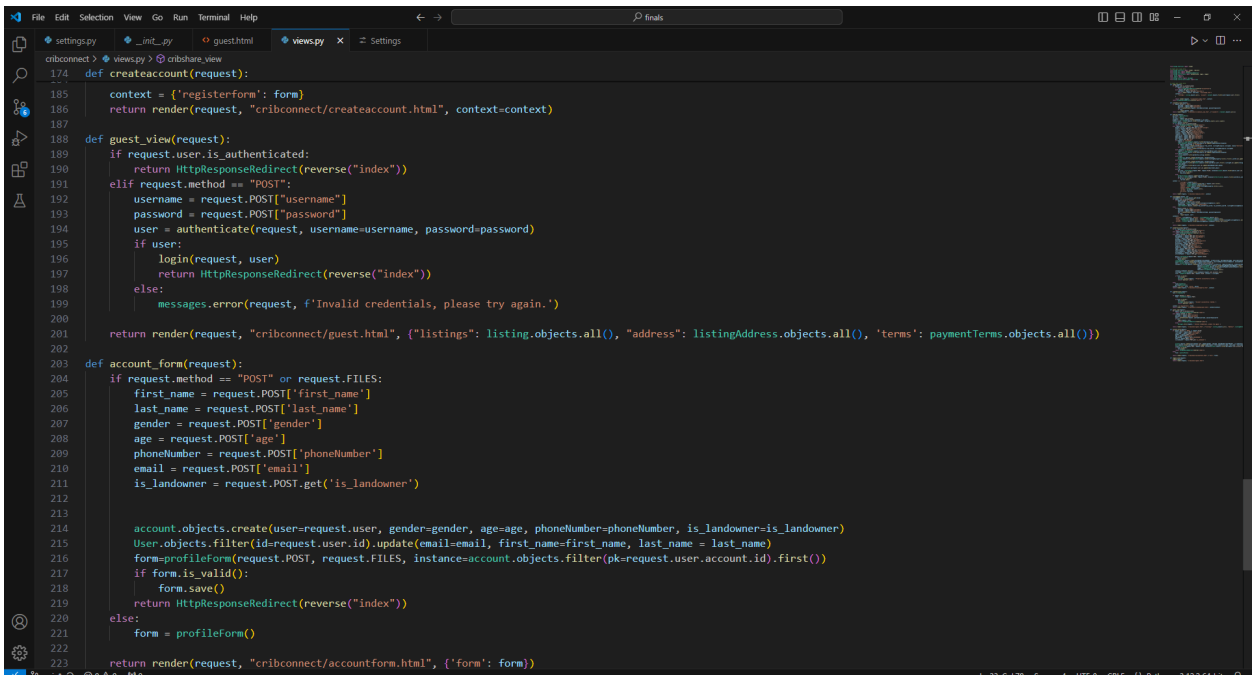
```python
def module2(request):
            rateReviews.objects.create(rating=lrate, review=lreview)
            listing.objects.filter(pk=tenant.objects.filter(userID=cur_user).first().listingID.id).update(rating=rateReviews.objects.last())
        elif edit_phone:
            account.objects.filter(pk=cur_user.id).update(phoneNumber=edit_phone)
        elif edit_email:
            User.objects.filter(pk=request.user.id).update(email=edit_email1)
        elif dp:
            pic_form = profileForm(request.POST, request.FILES, instance=account.objects.filter(pk=cur_user.id).first())
            if pic_form.is_valid():
                pic_form.save()
        else:
            forVerification.objects.create(userID=cur_user)
            id_form = idForm(request.POST, request.FILES, instance=forVerification.objects.filter(userID=cur_user.id).last())
            if id_form.is_valid():
                id_form.save()
    context = {
                "listings": property.all(),
                "account": account.objects.filter(user = request.user).first(),
                'requests': inquiryRequest.objects.all(),
                'tenants': tenant.objects.filter(listingID=property.first()).all(),
                'tenants_count': tenants_count,
                'rating': rateReviews.objects.all(),
                'id_form': id_form,
                'pic_form': pic_form
                }
    return render(request, "cribconnect/module2.html", context)

def listingpage(request, pk):
    listingdetails = listing.objects.get(id=pk)
    if request.user.is_authenticated:
        if request.method == "POST":
            from_userID = request.user.account
            to_userID = account.objects.filter(listing=listingdetails).last()
            requestType = request.POST.get('requestType')
            inquiryRequest.objects.create(from_userID=from_userID, to_userID=to_userID, listingID=listingdetails, requestType=requestType)
    else:
        if request.method == "POST":
            username = request.POST["username"]
            password = request.POST["password"]
```



```python
def listingpage(request, pk):
    listingdetails = listing.objects.get(id=pk)
    if request.user.is_authenticated:
        if request.method == "POST":
            from_userID = request.user.account
            to_userID = account.objects.filter(listing=listingdetails).last()
            requestType = request.POST.get('requestType')
            inquiryRequest.objects.create(from_userID=from_userID, to_userID=to_userID, listingID=listingdetails, requestType=requestType)
    else:
        if request.method == "POST":
            username = request.POST["username"]
            password = request.POST["password"]
            user = authenticate(request, username=username, password=password)
            if user:
                login(request, user)
    context={
        'listing': listingdetails, 'address': listingAddress.objects.all(),
        'terms': paymentTerms.objects.all(), 'images': imageGallery.objects.all(),
        'rating': rateReviews.objects.all(), 'cribshare': cribshare.objects.filter(listingID=listingdetails).all(),
        'account': account.objects.filter(pk=listingdetails.landowner.first().id).first()
    }

    return render(request, "cribconnect/viewproperty.html", context)

def propertylisting(request):
    if not request.user.is_authenticated:
        return HttpResponseRedirect(reverse("guest"))
    elif request.user.account.is_landowner == 0:
        return HttpResponseRedirect(reverse("index"))
    elif request.method == "POST":
        description = request.POST.get("description")
        houseNumber = request.POST.get("houseNumber")
        street = request.POST.get("street")
        barangay = request.POST.get("barangay")
        municipality = request.POST.get("municipality")
        province = request.POST.get("province")
        postalCode = request.POST.get("postalCode")
        price = request.POST.get("price")
        capacity = request.POST.get("capacity")
        dimension = request.POST.get("dimension")
```

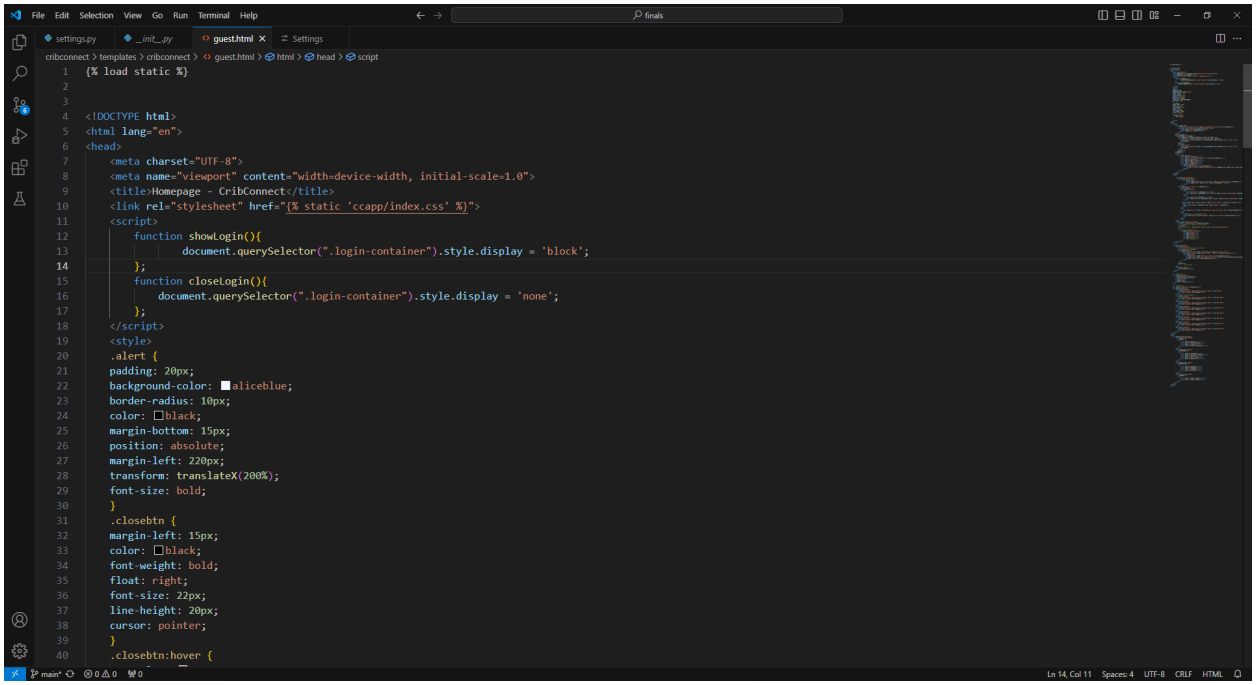


```python
def propertylisting(request):
        utilityPayment = request.POST.get('utilityPayment')

        gform = galleryForm(request.POST, request.FILES)
        if gform.is_valid():
            gform.save()
        listingAddress.objects.create(houseNumber=houseNumber, street=street, barangay=barangay, municipality=municipality, province=province, postalCode=postalCode,)
        paymentTerms.objects.create(advancePayment=advancePayment, securityDeposit=securityDeposit, minimumStay=minimumStay, utilityPayment=utilityPayment)
        user = account.objects.filter(id = request.user.account.id)
        instance = listing.objects.create(description=description, price=price, capacity=capacity, dimension=dimension, type=type,
                                          address=listingAddress.objects.filter(houseNumber=houseNumber).last(),
                                          terms=paymentTerms.objects.filter(advancePayment=advancePayment).last(),
                                          images = imageGallery.objects.last(),
                                          features=listingFeatures.objects.last())
        instance.landowner.set(user)
        listingID = listing.objects.filter(landowner=request.user.account).last()
        form = imageForm(request.POST, request.FILES, instance = listingID)

        if form.is_valid():
            form.save()
            messages.success(request, f'Property successfully listed.')
            return redirect('index')

    else:
        form=imageForm()
        gform=galleryForm()
    context = {'form': form, 'gform': gform}
    return render(request, "cribconnect/listproperty.html", context)


def createaccount(request):
    form = createUser()

    if request.method == "POST":
        form = createUser(request.POST)

        if form.is_valid():
            form.save()
            messages.success(request, f'Account successfully created.')
            return redirect('index')
```

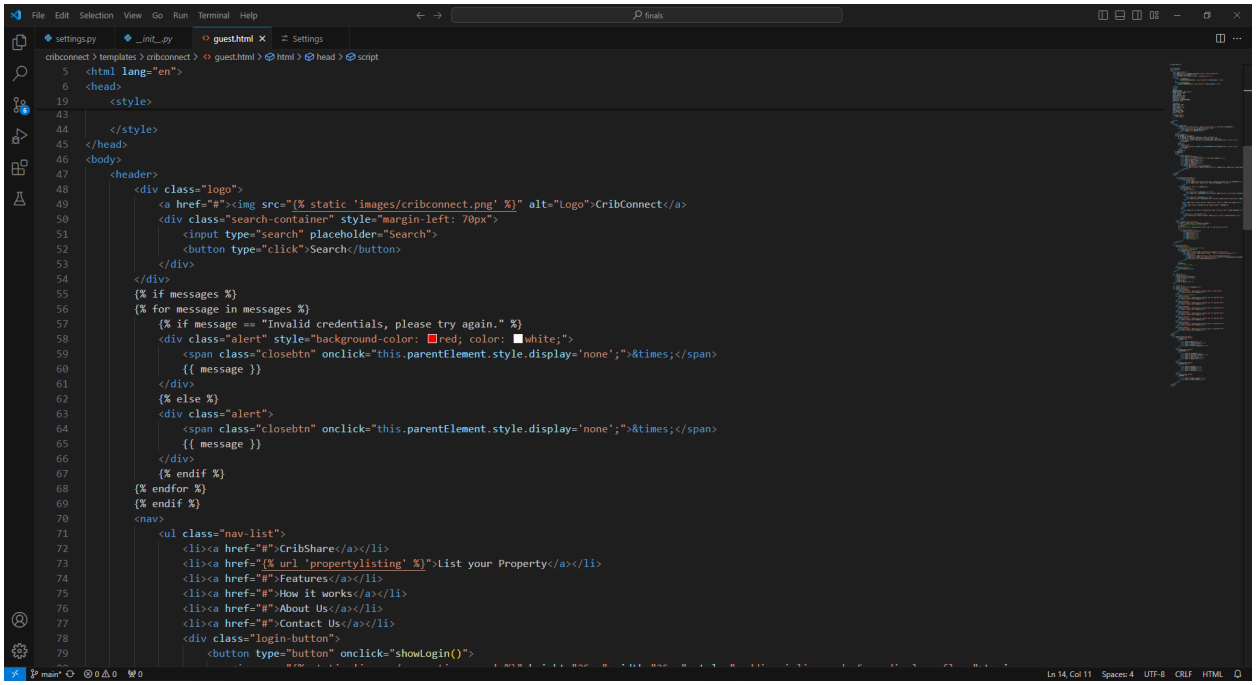


```python
def createaccount(request):
    context = {'registerform': form}
    return render(request, "cribconnect/createaccount.html", context=context)

def guest_view(request):
    if request.user.is_authenticated:
        return HttpResponseRedirect(reverse("index"))
    elif request.method == "POST":
        username = request.POST["username"]
        password = request.POST["password"]
        user = authenticate(request, username=username, password=password)
        if user:
            login(request, user)
            return HttpResponseRedirect(reverse("index"))
        else:
            messages.error(request, f'Invalid credentials, please try again.')

    return render(request, "cribconnect/guest.html", {"listings": listing.objects.all(), "address": listingAddress.objects.all(), 'terms': paymentTerms.objects.all()})

def account_form(request):
    if request.method == "POST" or request.FILES:
        first_name = request.POST['first_name']
        last_name = request.POST['last_name']
        gender = request.POST['gender']
        age = request.POST['age']
        phoneNumber = request.POST['phoneNumber']
        email = request.POST['email']
        is_landowner = request.POST.get('is_landowner')


        account.objects.create(user=request.user, gender=gender, age=age, phoneNumber=phoneNumber, is_landowner=is_landowner)
        User.objects.filter(id=request.user.id).update(email=email, first_name=first_name, last_name = last_name)
        form=profileForm(request.POST, request.FILES, instance=account.objects.filter(pk=request.user.account.id).first())
        if form.is_valid():
            form.save()
        return HttpResponseRedirect(reverse("index"))
    else:
        form = profileForm()

    return render(request, "cribconnect/accountform.html", {'form': form})
```

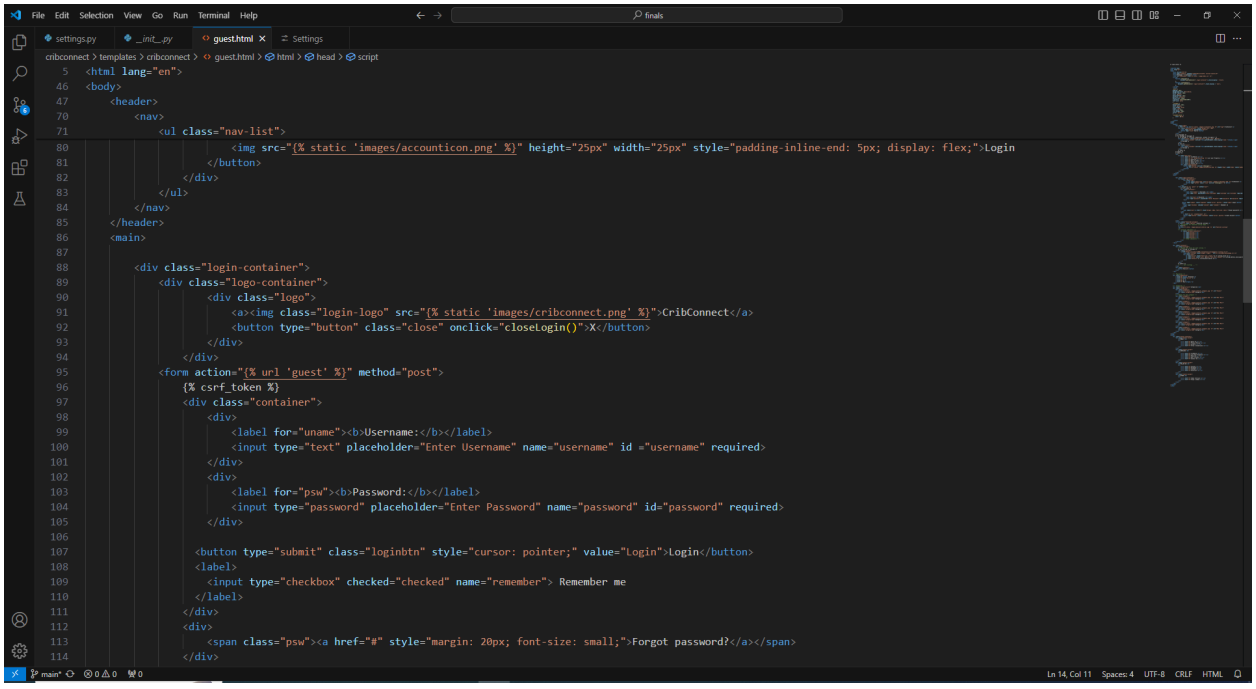
```python
            form.save()
        return HttpResponseRedirect(reverse("index"))
    else:
        form = profileForm()

    return render(request, "cribconnect/accountform.html", {'form': form})

def logout_view(request):
    logout(request)
    return render(request, "cribconnect/guest.html")
```



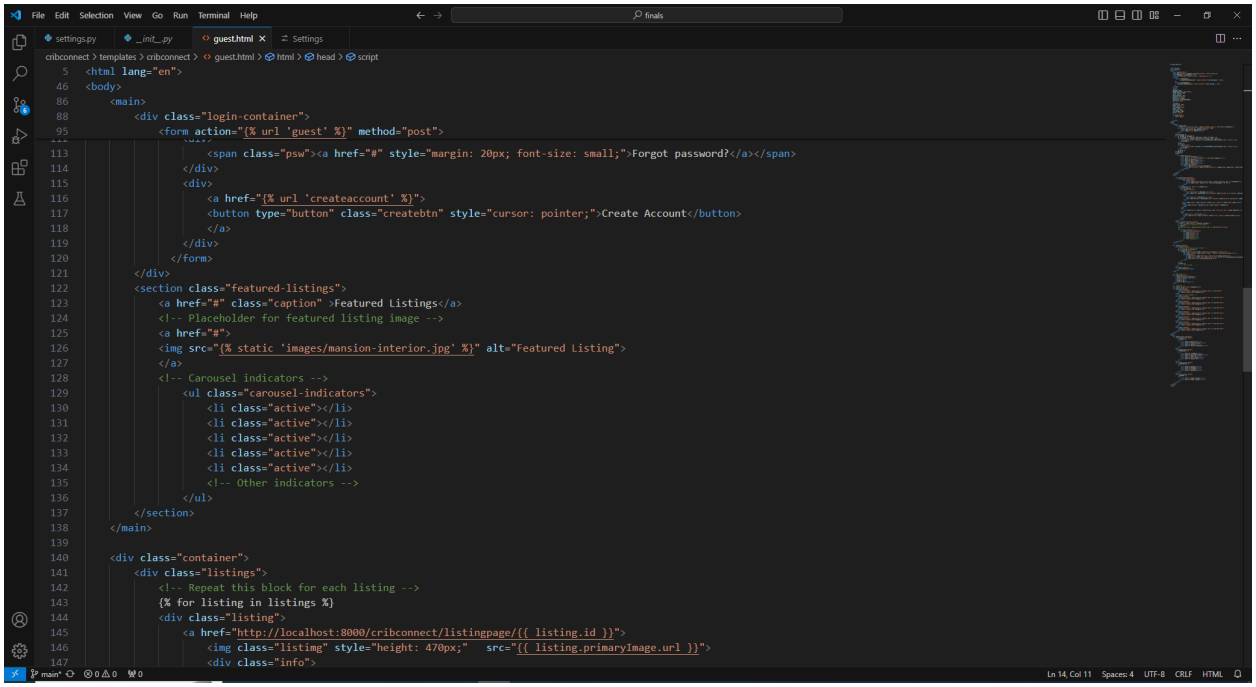
```
{% load static %}


<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Homepage - CribConnect</title>
    <link rel="stylesheet" href="{% static 'ccapp/index.css' %}">
    <script>
        function showLogin(){
                document.querySelector(".login-container").style.display = 'block';
        };
        function closeLogin(){
            document.querySelector(".login-container").style.display = 'none';
        };
    </script>
    <style>
    .alert {
    padding: 20px;
    background-color: aliceblue;
    border-radius: 10px;
    color: black;
    margin-bottom: 15px;
    position: absolute;
    margin-left: 220px;
    transform: translateX(200%);
    font-size: bold;
    }
    .closebtn {
    margin-left: 15px;
    color: black;
    font-weight: bold;
    float: right;
    font-size: 22px;
    line-height: 20px;
    cursor: pointer;
    }
    .closebtn:hover {
```



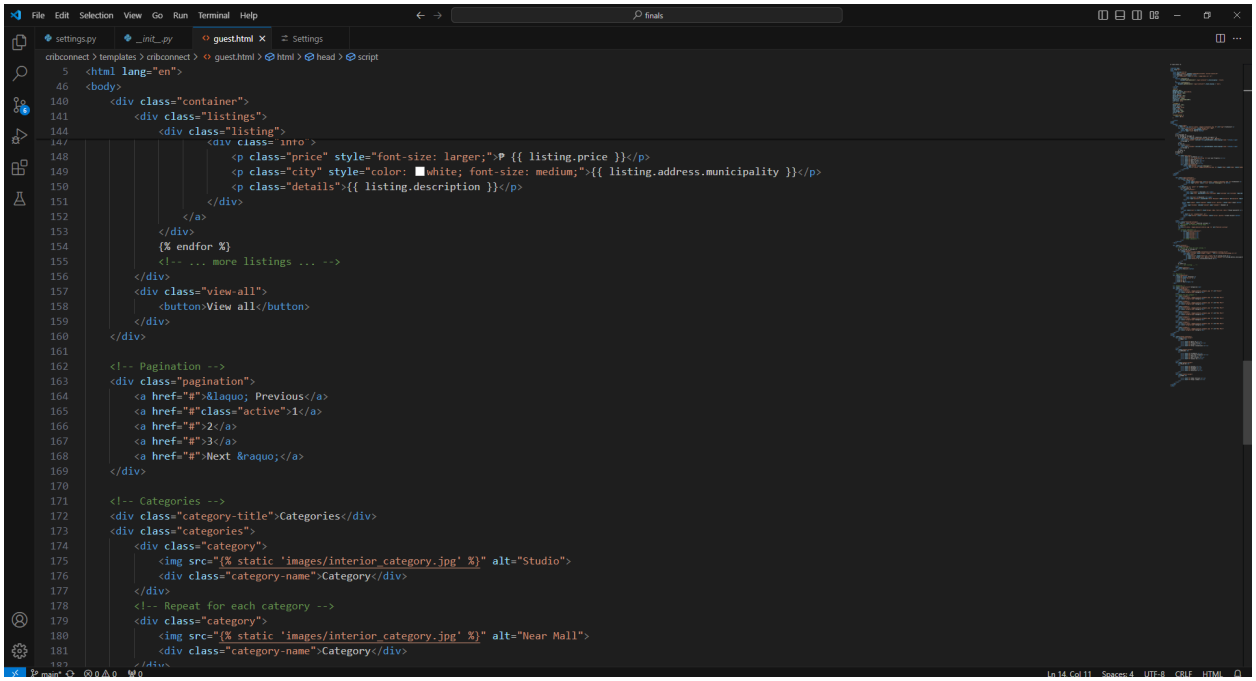
```
<html lang="en">
<head>
    <style>

    </style>
</head>
<body>
    <header>
        <div class="logo">
            <a href="#"><img src="{% static 'images/cribconnect.png' %}" alt="Logo">CribConnect</a>
            <div class="search-container" style="margin-left: 70px">
                <input type="search" placeholder="Search">
                <button type="click">Search</button>
            </div>
        </div>
        {% if messages %}
        {% for message in messages %}
            {% if message == "Invalid credentials, please try again." %}
            <div class="alert" style="background-color: red; color: white;">
                <span class="closebtn" onclick="this.parentElement.style.display='none';">&times;</span>
                {{ message }}
            </div>
            {% else %}
            <div class="alert">
                <span class="closebtn" onclick="this.parentElement.style.display='none';">&times;</span>
                {{ message }}
            </div>
            {% endif %}
        {% endfor %}
        {% endif %}
        <nav>
            <ul class="nav-list">
                <li><a href="#">CribShare</a></li>
                <li><a href="{% url 'propertylisting' %}">List your Property</a></li>
                <li><a href="#">Features</a></li>
                <li><a href="#">How it works</a></li>
                <li><a href="#">About Us</a></li>
                <li><a href="#">Contact Us</a></li>
                <div class="login-button">
                    <button type="button" onclick="showLogin()">
```



```
<html lang="en">
<body>
    <header>
        <nav>
            <ul class="nav-list">
                        <img src="{% static 'images/accounticon.png' %}" height="25px" width="25px" style="padding-inline-end: 5px; display: flex;">Login
                    </button>
                </div>
            </ul>
        </nav>
    </header>
    <main>

        <div class="login-container">
            <div class="logo-container">
                <div class="logo">
                    <a><img class="login-logo" src="{% static 'images/cribconnect.png' %}">CribConnect</a>
                    <button type="button" class="close" onclick="closeLogin()">X</button>
                </div>
            </div>
            <form action="{% url 'guest' %}" method="post">
                {% csrf_token %}
                <div class="container">
                    <div>
                        <label for="uname"><b>Username:</b></label>
                        <input type="text" placeholder="Enter Username" name="username" id ="username" required>
                    </div>
                    <div>
                        <label for="psw"><b>Password:</b></label>
                        <input type="password" placeholder="Enter Password" name="password" id="password" required>
                    </div>

                  <button type="submit" class="loginbtn" style="cursor: pointer;" value="Login">Login</button>
                  <label>
                    <input type="checkbox" checked="checked" name="remember"> Remember me
                  </label>
                </div>
                <div>
                    <span class="psw"><a href="#" style="margin: 20px; font-size: small;">Forgot password?</a></span>
                </div>
```



```
<html lang="en">
<body>
    <main>
        <div class="login-container">
            <form action="{% url 'guest' %}" method="post">
                    <span class="psw"><a href="#" style="margin: 20px; font-size: small;">Forgot password?</a></span>
                </div>
                <div>
                    <a href="{% url 'createaccount' %}">
                    <button type="button" class="createbtn" style="cursor: pointer;">Create Account</button>
                    </a>
                </div>
            </form>
        </div>
        <section class="featured-listings">
            <a href="#" class="caption" >Featured Listings</a>
            <!-- Placeholder for featured listing image -->
            <a href="#">
            <img src="{% static 'images/mansion-interior.jpg' %}" alt="Featured Listing">
            </a>
            <!-- Carousel indicators -->
                <ul class="carousel-indicators">
                    <li class="active"></li>
                    <li class="active"></li>
                    <li class="active"></li>
                    <li class="active"></li>
                    <li class="active"></li>
                    <!-- Other indicators -->
                </ul>
        </section>
    </main>

    <div class="container">
        <div class="listings">
            <!-- Repeat this block for each listing -->
            {% for listing in listings %}
            <div class="listing">
                <a href="http://localhost:8000/cribconnect/listingpage/{{ listing.id }}">
                    <img class="listimg" style="height: 470px;"   src="{{ listing.primaryImage.url }}">
                    <div class="info">
```


File Edit Selection View Go Run Terminal Help
finals
settings.py
__init__.py
guest.html
Settings
cribconnect > templates > cribconnect > guest.html > html > head > script
<html lang="en">
<body>

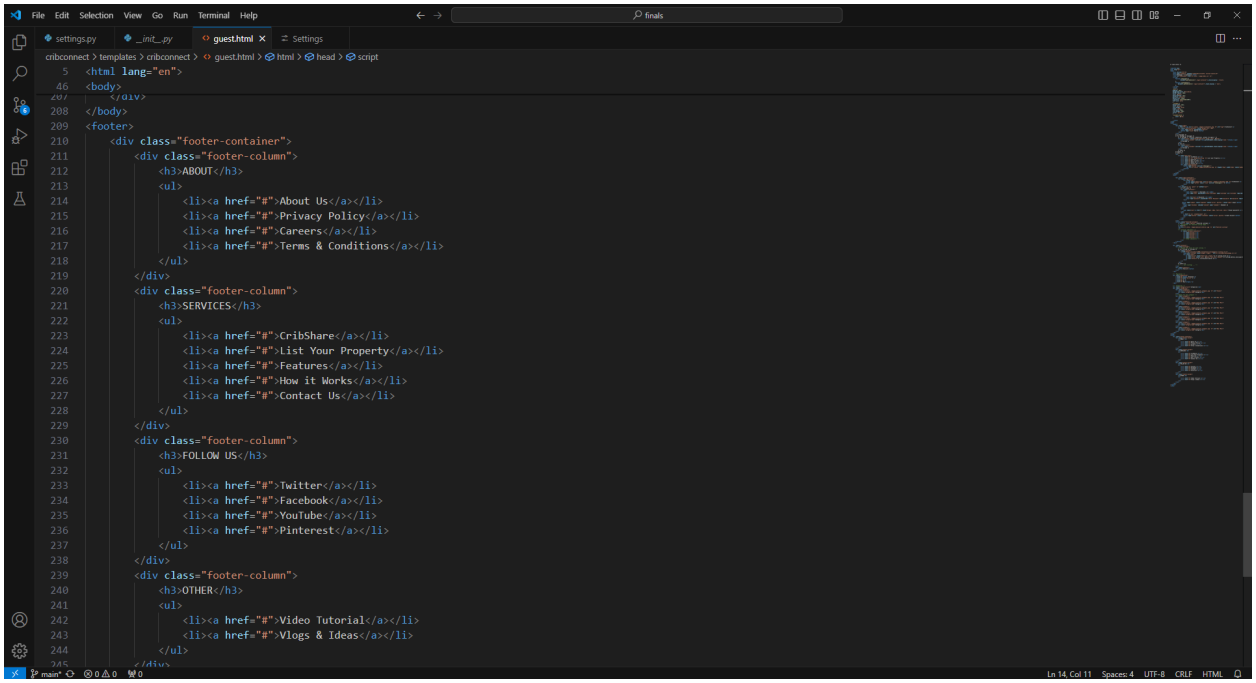
P {{ listing.price }}
{{ listing.address.municipality }}
{{ listing.description }}
{% endfor %}
<!-- ... more listings ... -->

<button>View all</button>
<!-- Pagination -->

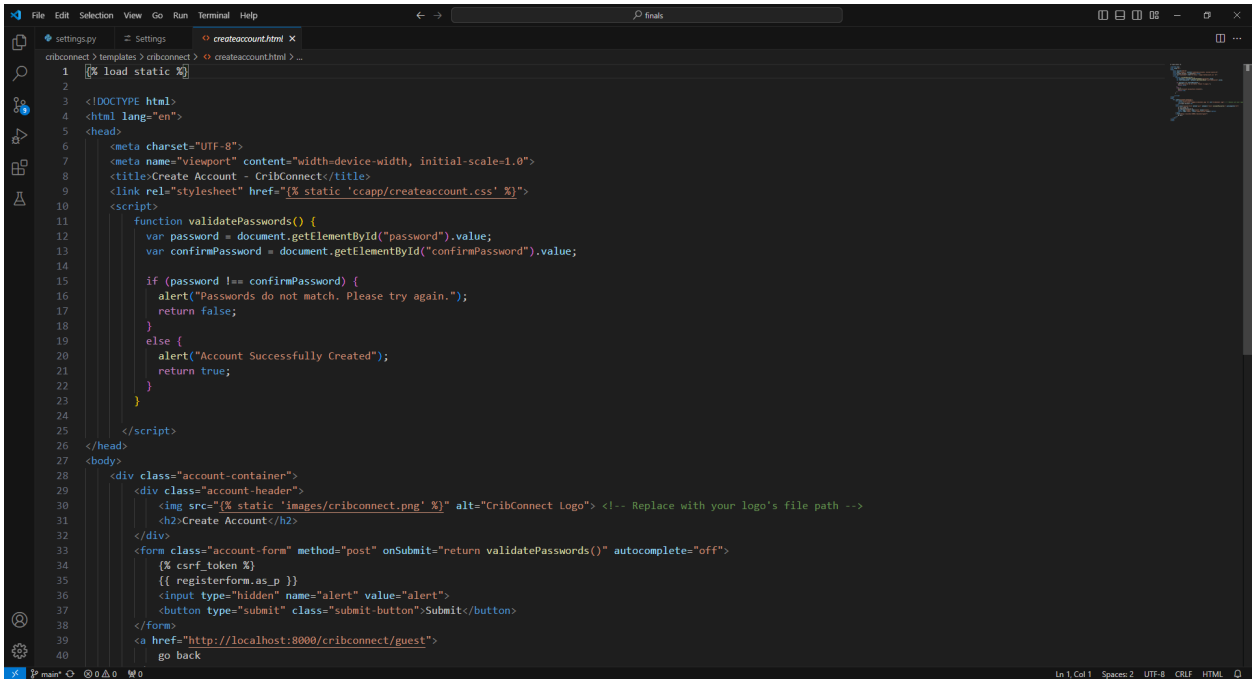
« Previous
1
2
3
Next »
<!-- Categories -->
Categories

<img src="{% static 'images/interior_category.jpg' %}" alt="Studio">
Category
<!-- Repeat for each category -->

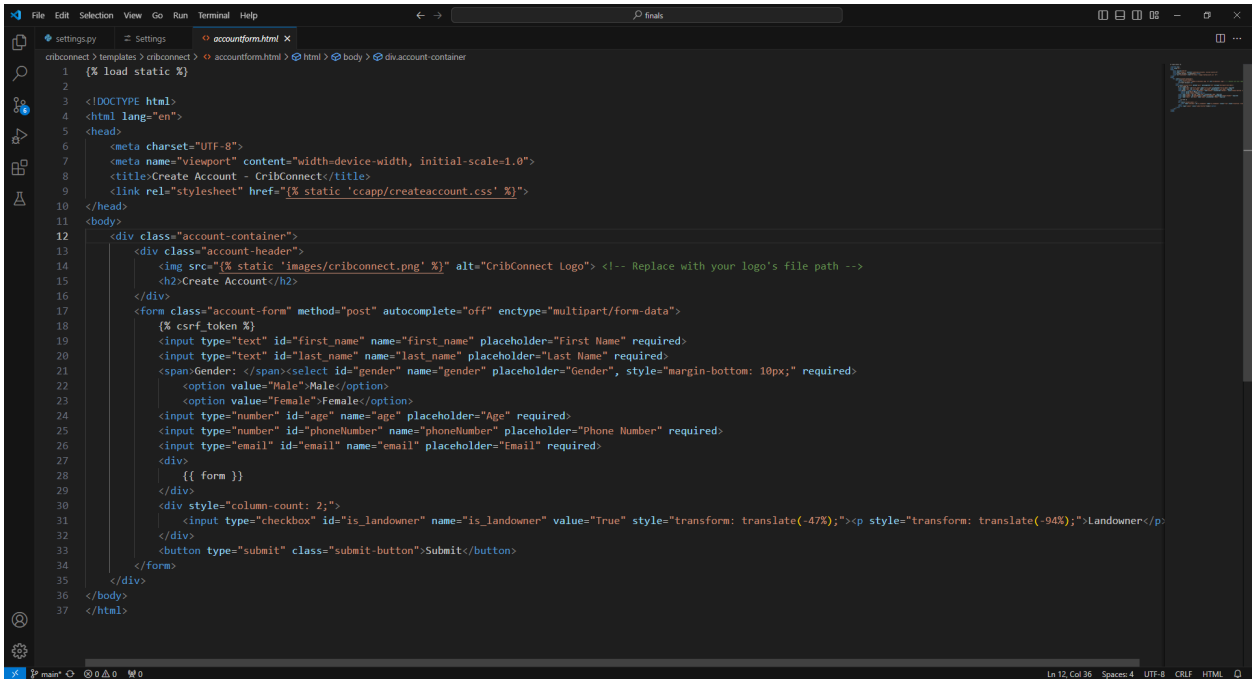
<img src="{% static 'images/interior_category.jpg' %}" alt="Near Mall">
Category
Ln 14, Col 11
Spaces: 4
UTF-8
CRLF
HTML



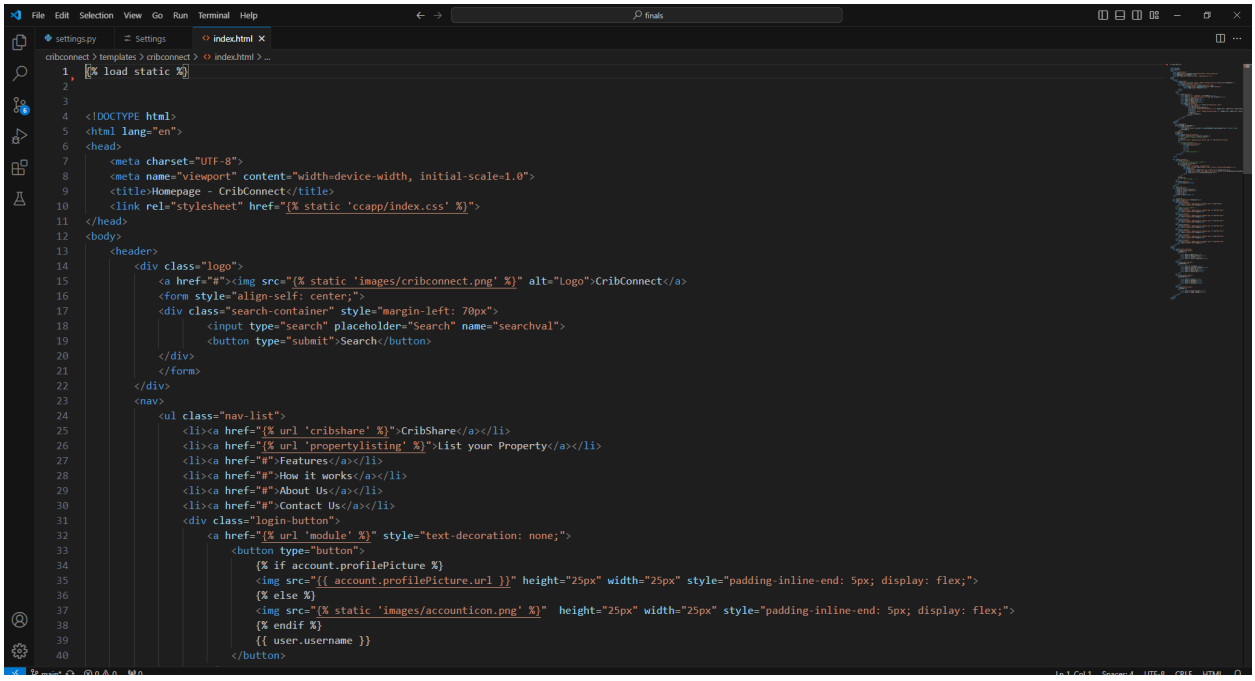
```
<html lang="en">
<body>
    </div>
</body>
<footer>
    <div class="footer-container">
        <div class="footer-column">
            <h3>ABOUT</h3>
            <ul>
                <li><a href="#">About Us</a></li>
                <li><a href="#">Privacy Policy</a></li>
                <li><a href="#">Careers</a></li>
                <li><a href="#">Terms & Conditions</a></li>
            </ul>
        </div>
        <div class="footer-column">
            <h3>SERVICES</h3>
            <ul>
                <li><a href="#">CribShare</a></li>
                <li><a href="#">List Your Property</a></li>
                <li><a href="#">Features</a></li>
                <li><a href="#">How it Works</a></li>
                <li><a href="#">Contact Us</a></li>
            </ul>
        </div>
        <div class="footer-column">
            <h3>FOLLOW US</h3>
            <ul>
                <li><a href="#">Twitter</a></li>
                <li><a href="#">Facebook</a></li>
                <li><a href="#">YouTube</a></li>
                <li><a href="#">Pinterest</a></li>
            </ul>
        </div>
        <div class="footer-column">
            <h3>OTHER</h3>
            <ul>
                <li><a href="#">Video Tutorial</a></li>
                <li><a href="#">Vlogs & Ideas</a></li>
            </ul>
        </div>
```



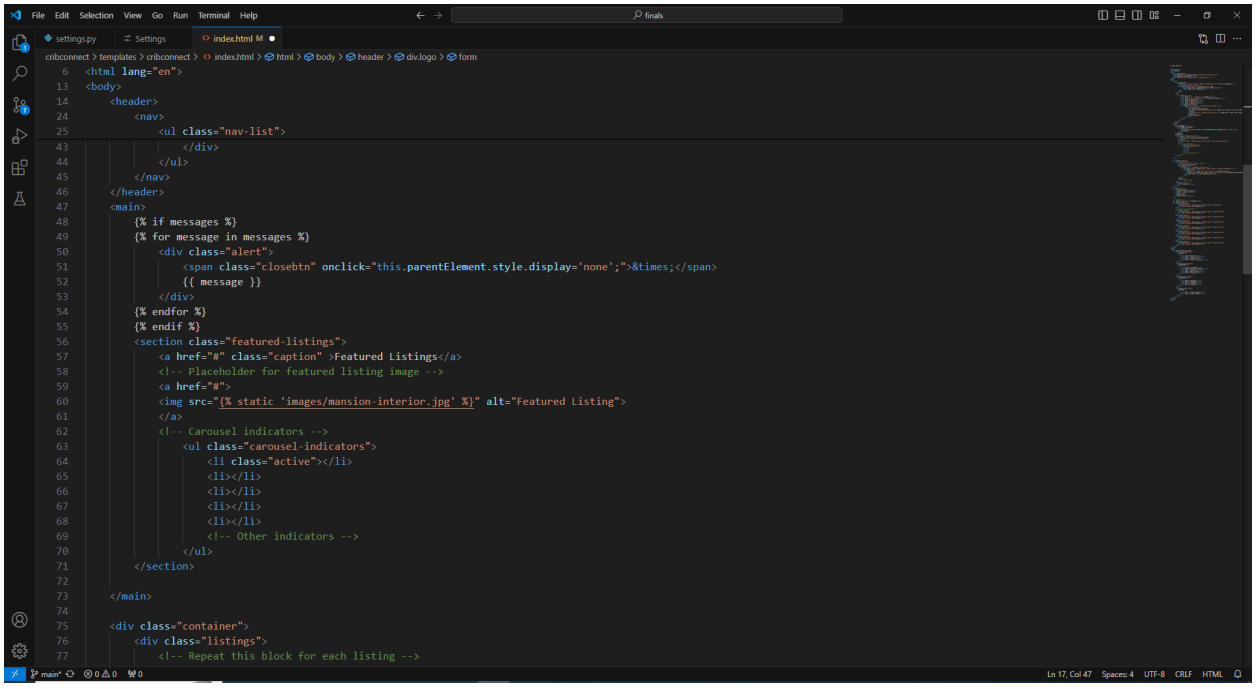
```
{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Create Account - CribConnect</title>
    <link rel="stylesheet" href="{% static 'ccapp/createaccount.css' %}">
    <script>
        function validatePasswords() {
          var password = document.getElementById("password").value;
          var confirmPassword = document.getElementById("confirmPassword").value;

          if (password !== confirmPassword) {
            alert("Passwords do not match. Please try again.");
            return false;
          }
          else {
            alert("Account Successfully Created");
            return true;
          }
        }

    </script>
</head>
<body>
    <div class="account-container">
        <div class="account-header">
            <img src="{% static 'images/cribconnect.png' %}" alt="CribConnect Logo"> <!-- Replace with your logo's file path -->
            <h2>Create Account</h2>
        </div>
        <form class="account-form" method="post" onSubmit="return validatePasswords()" autocomplete="off">
            {% csrf_token %}
            {{ registerform.as_p }}
            <input type="hidden" name="alert" value="alert">
            <button type="submit" class="submit-button">Submit</button>
        </form>
        <a href="http://localhost:8000/cribconnect/guest">
            go back
```



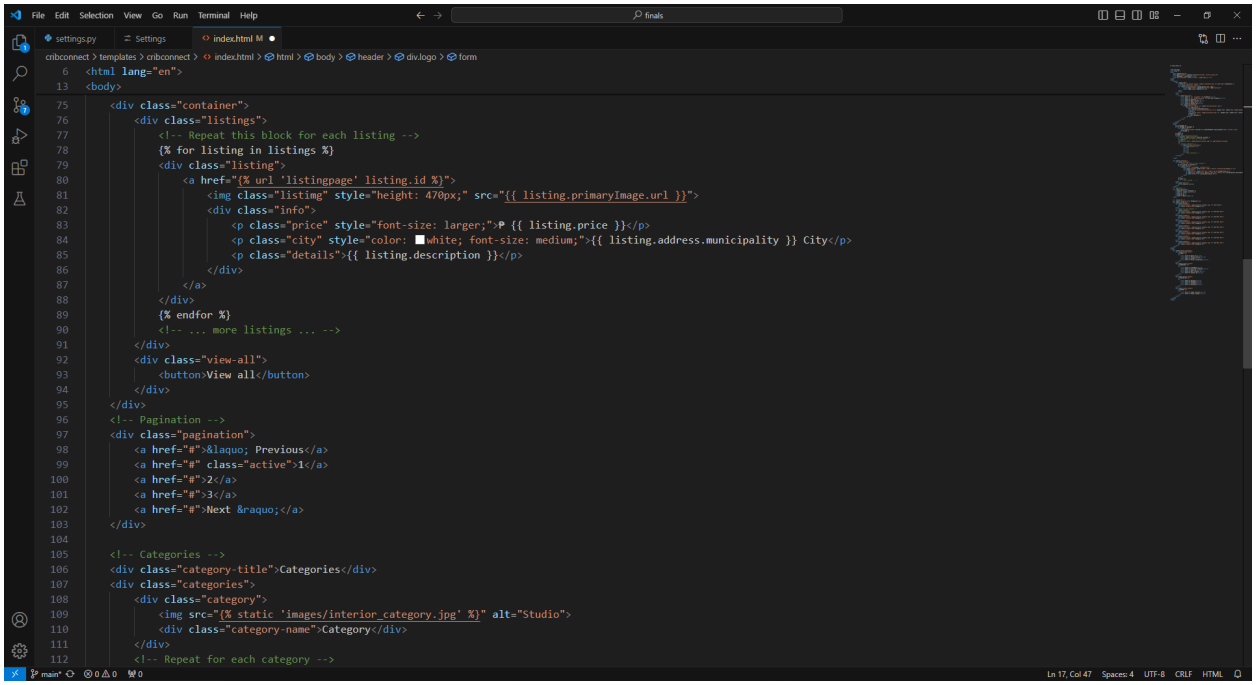
```
{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Create Account - CribConnect</title>
    <link rel="stylesheet" href="{% static 'ccapp/createaccount.css' %}">
</head>
<body>
    <div class="account-container">
        <div class="account-header">
            <img src="{% static 'images/cribconnect.png' %}" alt="CribConnect Logo"> <!-- Replace with your logo's file path -->
            <h2>Create Account</h2>
        </div>
        <form class="account-form" method="post" autocomplete="off" enctype="multipart/form-data">
            {% csrf_token %}
            <input type="text" id="first_name" name="first_name" placeholder="First Name" required>
            <input type="text" id="last_name" name="last_name" placeholder="Last Name" required>
            <span>Gender: </span><select id="gender" name="gender" placeholder="Gender", style="margin-bottom: 10px;" required>
                <option value="Male">Male</option>
                <option value="Female">Female</option>
            <input type="number" id="age" name="age" placeholder="Age" required>
            <input type="number" id="phoneNumber" name="phoneNumber" placeholder="Phone Number" required>
            <input type="email" id="email" name="email" placeholder="Email" required>
            <div>
                {{ form }}
            </div>
            <div style="column-count: 2;">
                <input type="checkbox" id="is_landowner" name="is_landowner" value="True" style="transform: translate(-47%);"><p style="transform: translate(-94%);">Landowner</p>
            </div>
            <button type="submit" class="submit-button">Submit</button>
        </form>
    </div>
</body>
</html>
```



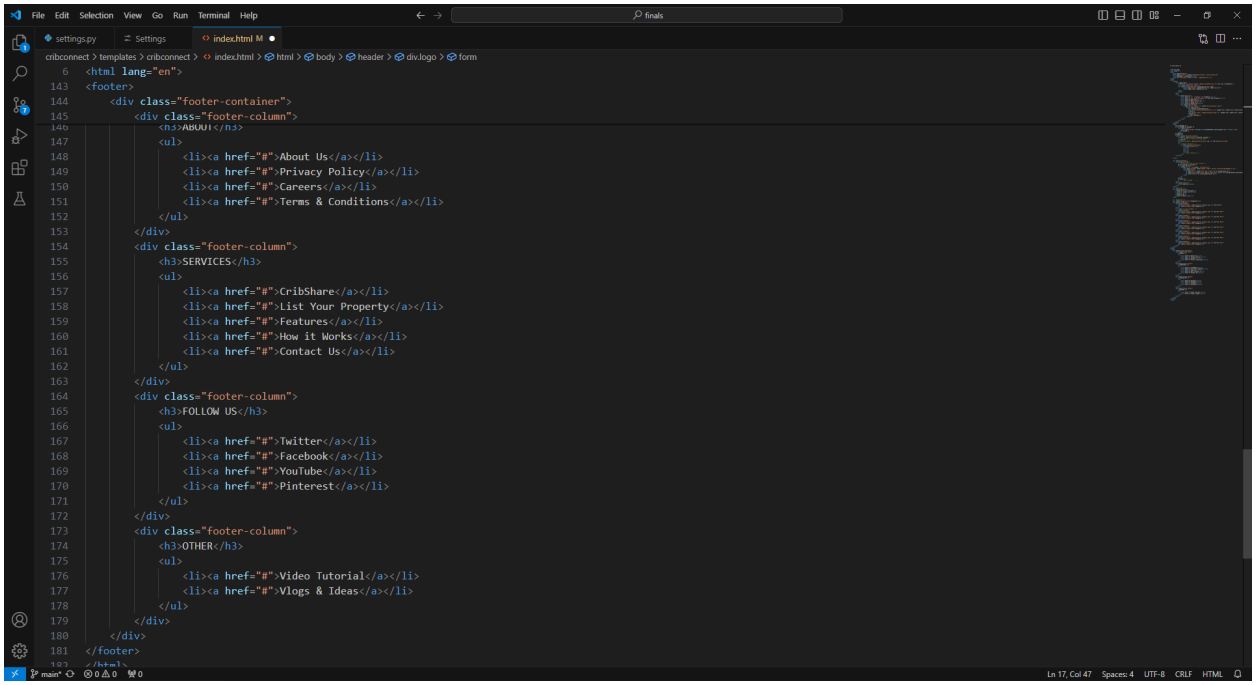
```
{% load static %}


<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Homepage - CribConnect</title>
    <link rel="stylesheet" href="{% static 'ccapp/index.css' %}">
</head>
<body>
    <header>
        <div class="logo">
            <a href="#"><img src="{% static 'images/cribconnect.png' %}" alt="Logo">CribConnect</a>
            <form style="align-self: center;">
            <div class="search-container" style="margin-left: 70px">
                    <input type="search" placeholder="Search" name="searchval">
                    <button type="submit">Search</button>
            </div>
            </form>
        </div>
        <nav>
            <ul class="nav-list">
                <li><a href="{% url 'cribshare' %}">CribShare</a></li>
                <li><a href="{% url 'propertylisting' %}">List your Property</a></li>
                <li><a href="#">Features</a></li>
                <li><a href="#">How it works</a></li>
                <li><a href="#">About Us</a></li>
                <li><a href="#">Contact Us</a></li>
                <div class="login-button">
                    <a href="{% url 'module' %}" style="text-decoration: none;">
                        <button type="button">
                            {% if account.profilePicture %}
                            <img src="{{ account.profilePicture.url }}" height="25px" width="25px" style="padding-inline-end: 5px; display: flex;">
                            {% else %}
                            <img src="{% static 'images/accounticon.png' %}"  height="25px" width="25px" style="padding-inline-end: 5px; display: flex;">
                            {% endif %}
                            {{ user.username }}
                        </button>
```

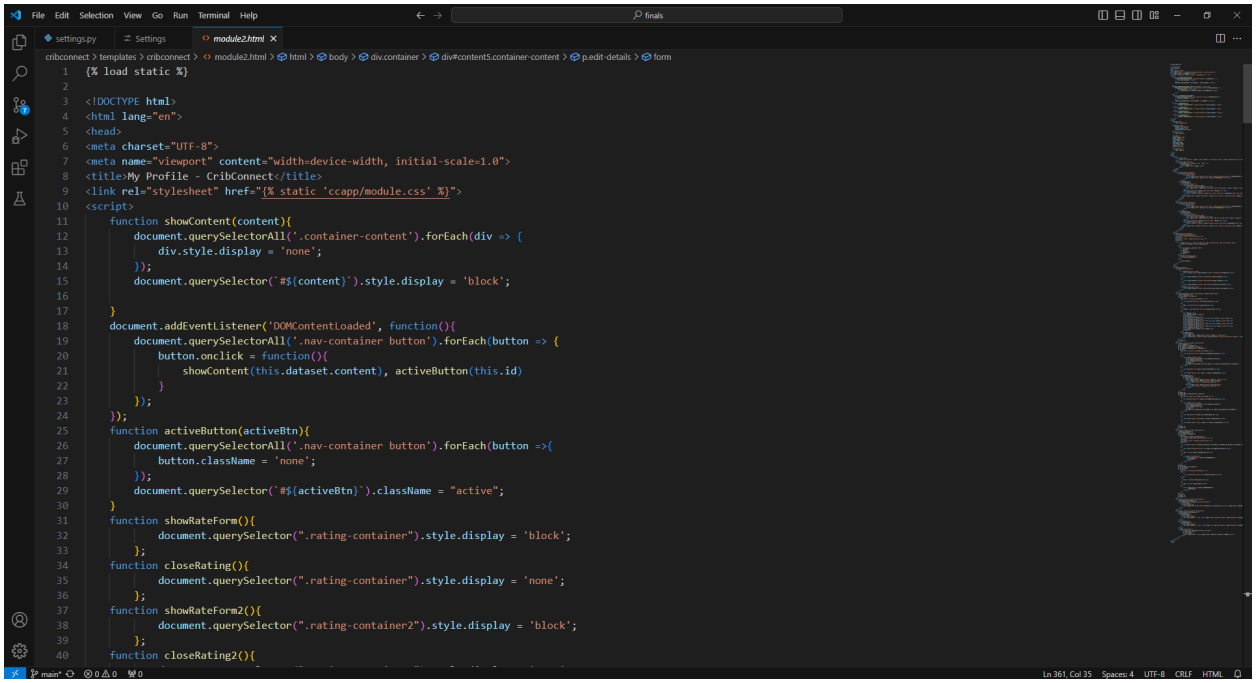


```
<html lang="en">
<body>
    <header>
        <nav>
            <ul class="nav-list">
                </div>
            </ul>
        </nav>
    </header>
    <main>
        {% if messages %}
        {% for message in messages %}
            <div class="alert">
                <span class="closebtn" onclick="this.parentElement.style.display='none';">&times;</span>
                {{ message }}
            </div>
        {% endfor %}
        {% endif %}
        <section class="featured-listings">
            <a href="#" class="caption" >Featured Listings</a>
            <!-- Placeholder for featured listing image -->
            <a href="#">
            <img src="{% static 'images/mansion-interior.jpg' %}" alt="Featured Listing">
            </a>
            <!-- Carousel indicators -->
                <ul class="carousel-indicators">
                    <li class="active"></li>
                    <li></li>
                    <li></li>
                    <li></li>
                    <li></li>
                    <!-- Other indicators -->
                </ul>
        </section>

    </main>

    <div class="container">
        <div class="listings">
            <!-- Repeat this block for each listing -->
```

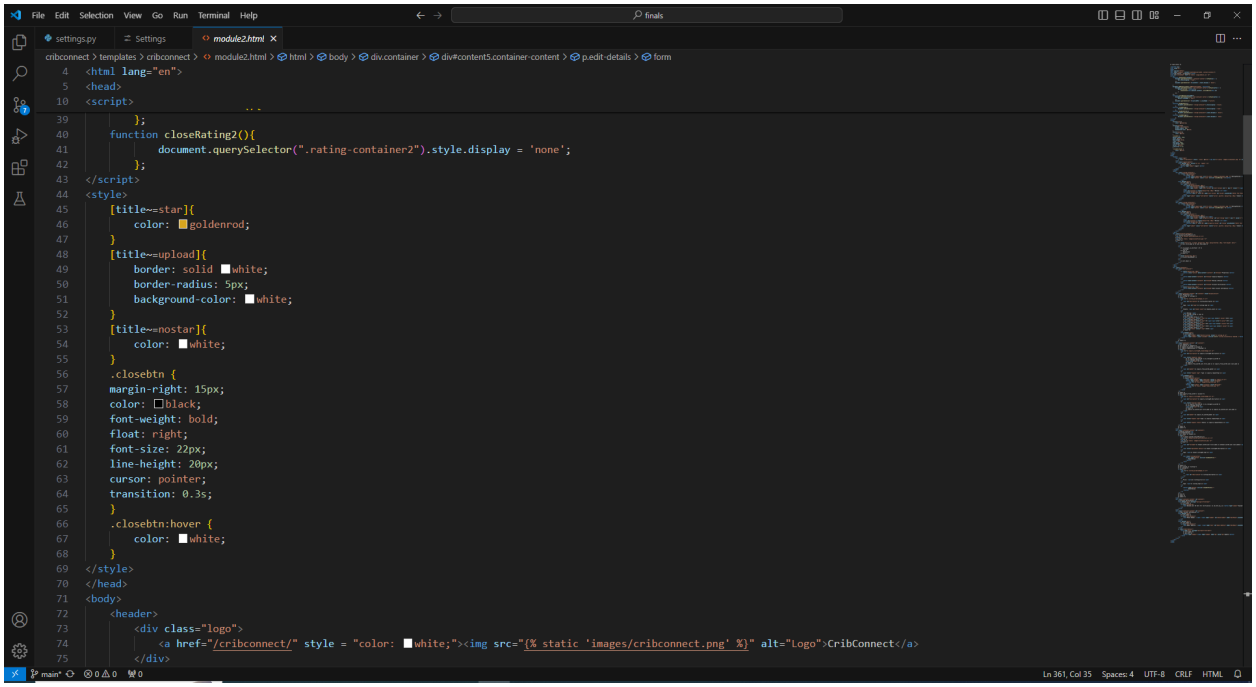


```
<html lang="en">
<body>
    <div class="container">
        <div class="listings">
            <!-- Repeat this block for each listing -->
            {% for listing in listings %}
            <div class="listing">
                <a href="{% url 'listingpage' listing_id %}">
                    <img class="listimg" style="height: 470px;" src="{{ listing.primaryImage.url }}">
                    <div class="info">
                        <p class="price" style="font-size: larger;">₱ {{ listing.price }}</p>
                        <p class="city" style="color: white; font-size: medium;">{{ listing.address.municipality }} City</p>
                        <p class="details">{{ listing.description }}</p>
                    </div>
                </a>
            </div>
            {% endfor %}
            <!-- ... more listings ... -->
        </div>
        <div class="view-all">
            <button>View all</button>
        </div>
    </div>
    <!-- Pagination -->
    <div class="pagination">
        <a href="#">&laquo; Previous</a>
        <a href="#" class="active">1</a>
        <a href="#">2</a>
        <a href="#">3</a>
        <a href="#">Next &raquo;</a>
    </div>

    <!-- Categories -->
    <div class="category-title">Categories</div>
    <div class="categories">
        <div class="category">
            <img src="{% static 'images/interior_category.jpg' %}" alt="Studio">
            <div class="category-name">Category</div>
        </div>
        <!-- Repeat for each category -->
```

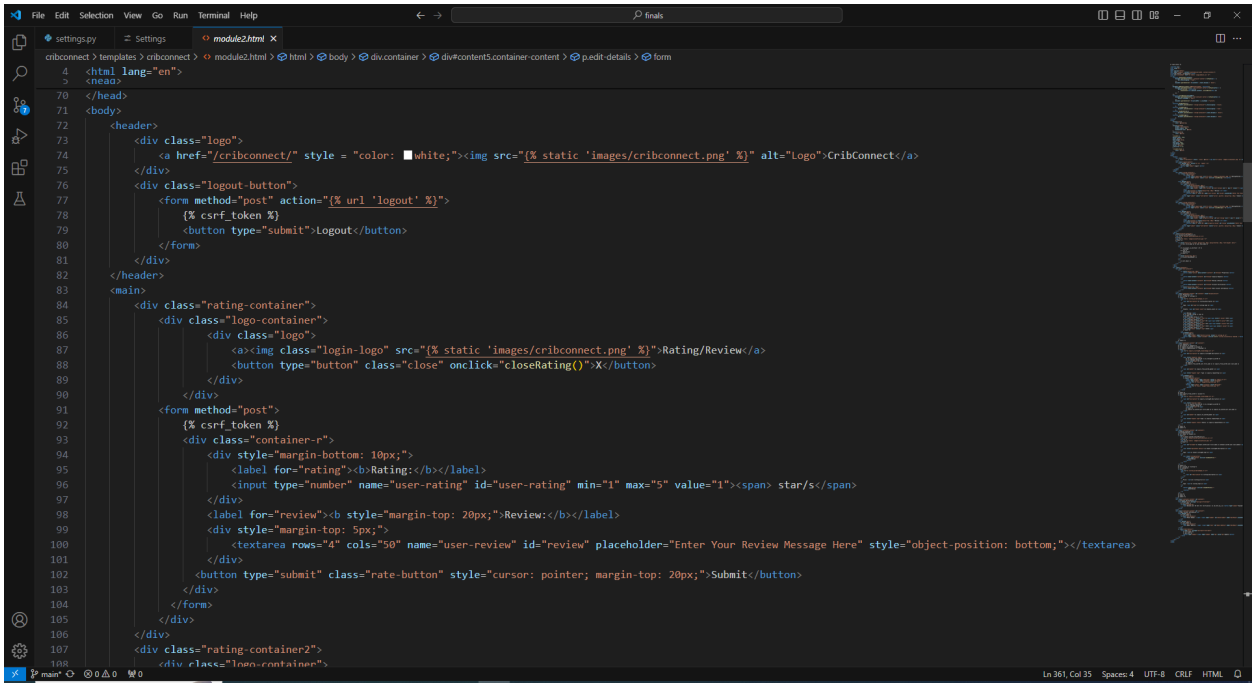

File Edit Selection View Go Run Terminal Help
finals
settings.py
Settings
index.html M
cribconnect > templates > cribconnect > index.html > html > body > header > div.logo > form
6 <html lang="en">
143 <footer>
144
145
146 <h3>ABOUT</h3>
147
148 About Us
149 Privacy Policy
150 Careers
151 Terms & Conditions
152
153
154
155 <h3>SERVICES</h3>
156
157 CribShare
158 List Your Property
159 Features
160 How it Works
161 Contact Us
162
163
164
165 <h3>FOLLOW US</h3>
166
167 Twitter
168 Facebook
169 YouTube
170 Pinterest
171
172
173
174 <h3>OTHER</h3>
175
176 Video Tutorial
177 Vlogs & Ideas
178
179
180
181 </footer>
main*
Ln 17, Col 47
Spaces: 4
UTF-8
CRLF
HTML



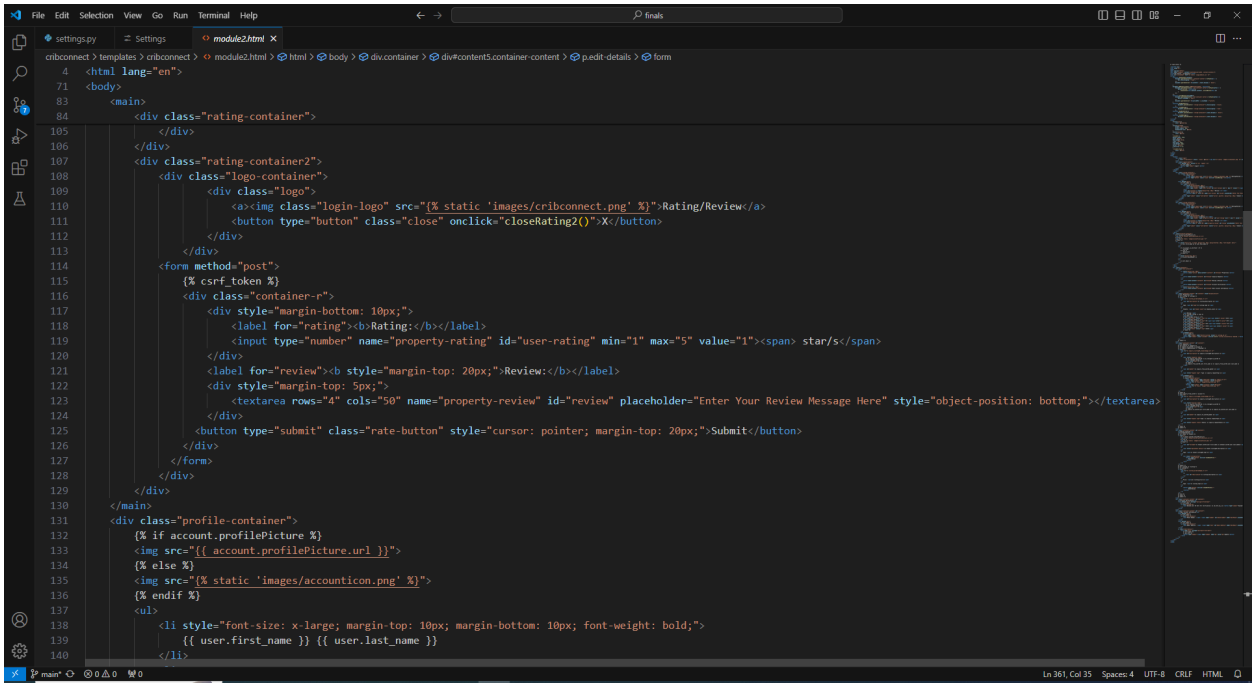
```
{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>My Profile - CribConnect</title>
<link rel="stylesheet" href="{% static 'ccapp/module.css' %}">
<script>
    function showContent(content){
        document.querySelectorAll('.container-content').forEach(div => {
            div.style.display = 'none';
        });
        document.querySelector(`#${content}`).style.display = 'block';

    }
    document.addEventListener('DOMContentLoaded', function(){
        document.querySelectorAll('.nav-container button').forEach(button => {
            button.onclick = function(){
                showContent(this.dataset.content), activeButton(this.id)
            }
        });
    });
    function activeButton(activeBtn){
        document.querySelectorAll('.nav-container button').forEach(button =>{
            button.className = 'none';
        });
        document.querySelector(`#${activeBtn}`).className = "active";
    }
    function showRateForm(){
            document.querySelector(".rating-container").style.display = 'block';
        };
    function closeRating(){
            document.querySelector(".rating-container").style.display = 'none';
        };
    function showRateForm2(){
            document.querySelector(".rating-container2").style.display = 'block';
        };
    function closeRating2(){
```



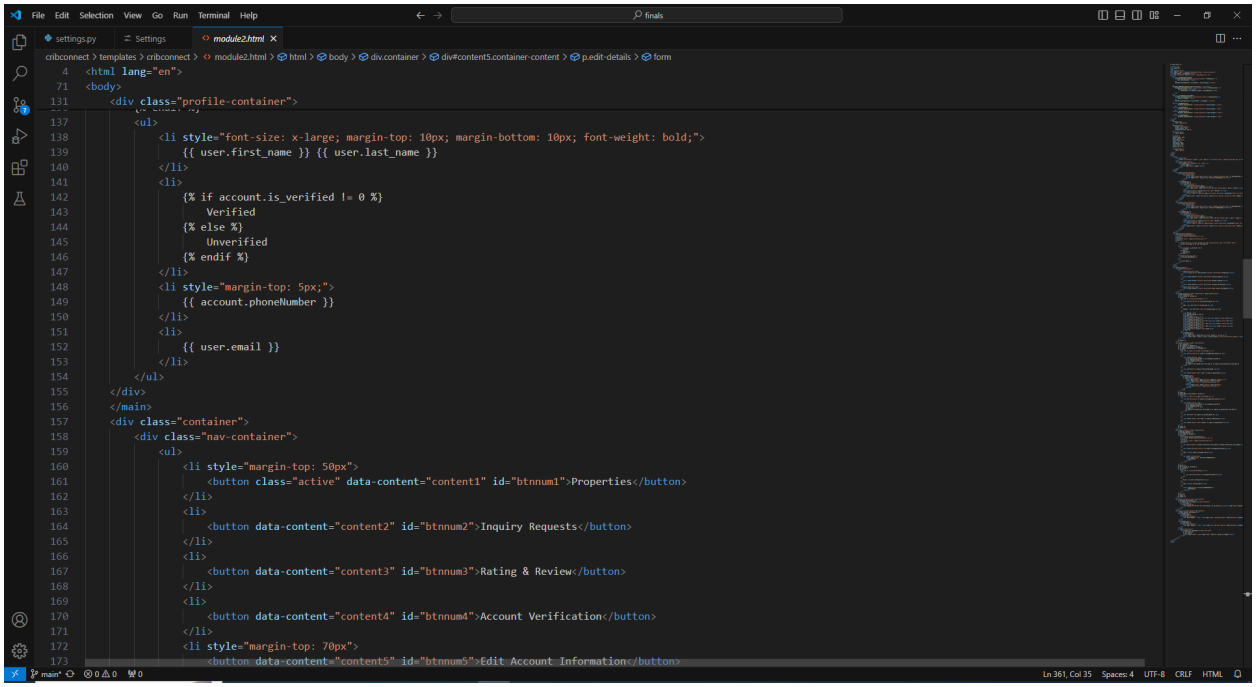
```
<html lang="en">
<head>
<script>
        };
    function closeRating2(){
            document.querySelector(".rating-container2").style.display = 'none';
        };
</script>
<style>
    [title~=star]{
        color: goldenrod;
    }
    [title~=upload]{
        border: solid white;
        border-radius: 5px;
        background-color: white;
    }
    [title~=nostar]{
        color: white;
    }
    .closebtn {
    margin-right: 15px;
    color: black;
    font-weight: bold;
    float: right;
    font-size: 22px;
    line-height: 20px;
    cursor: pointer;
    transition: 0.3s;
    }
    .closebtn:hover {
        color: white;
    }
</style>
</head>
<body>
    <header>
        <div class="logo">
            <a href="/cribconnect/" style = "color: white;"><img src="{% static 'images/cribconnect.png' %}" alt="Logo">CribConnect</a>
        </div>
```



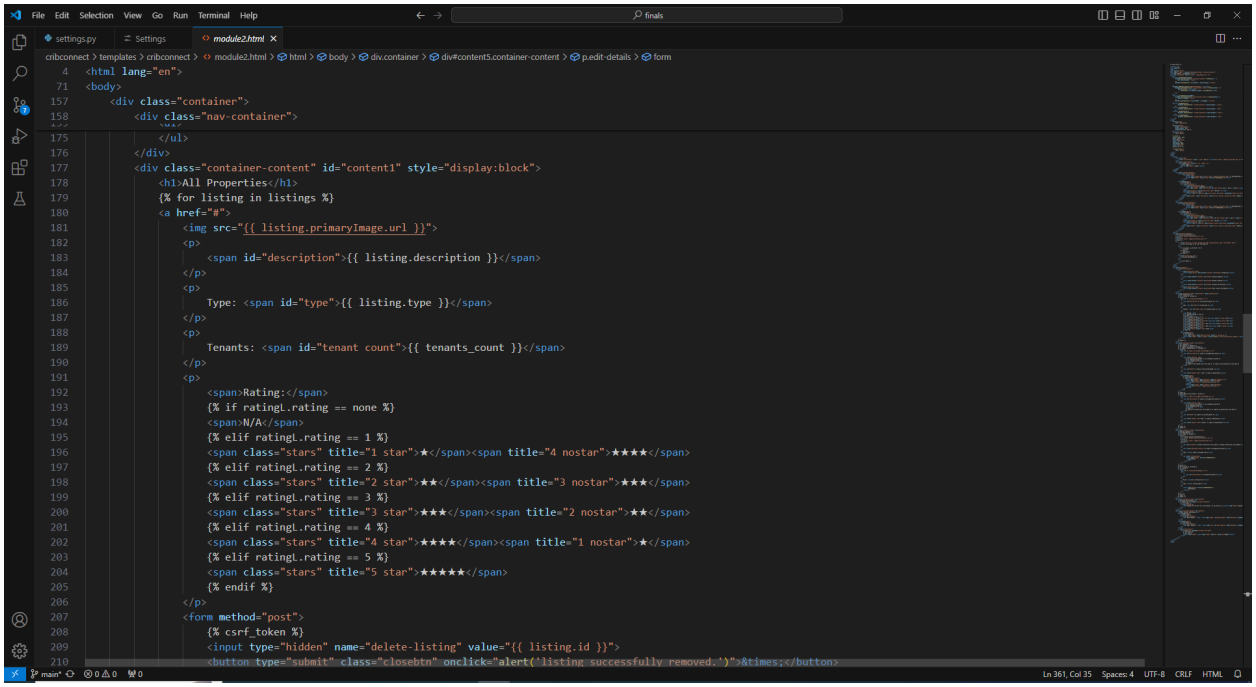
```
<html lang="en">
<head>
</head>
<body>
    <header>
        <div class="logo">
            <a href="/cribconnect/" style = "color: white;"><img src="{% static 'images/cribconnect.png' %}" alt="Logo">CribConnect</a>
        </div>
        <div class="logout-button">
            <form method="post" action="{% url 'logout' %}">
                {% csrf_token %}
                <button type="submit">Logout</button>
            </form>
        </div>
    </header>
    <main>
        <div class="rating-container">
            <div class="logo-container">
                    <div class="logo">
                        <a><img class="login-logo" src="{% static 'images/cribconnect.png' %}">Rating/Review</a>
                        <button type="button" class="close" onclick="closeRating()">X</button>
                    </div>
                </div>
            <form method="post">
                {% csrf_token %}
                <div class="container-r">
                    <div style="margin-bottom: 10px;">
                        <label for="rating"><b>Rating:</b></label>
                        <input type="number" name="user-rating" id="user-rating" min="1" max="5" value="1"><span> star/s</span>
                    </div>
                    <label for="review"><b style="margin-top: 20px;">Review:</b></label>
                    <div style="margin-top: 5px;">
                        <textarea rows="4" cols="50" name="user-review" id="review" placeholder="Enter Your Review Message Here" style="object-position: bottom;"></textarea>
                    </div>
                  <button type="submit" class="rate-button" style="cursor: pointer; margin-top: 20px;">Submit</button>
                </div>
              </form>
            </div>
        </div>
        <div class="rating-container2">
            <div class="logo-container">
```



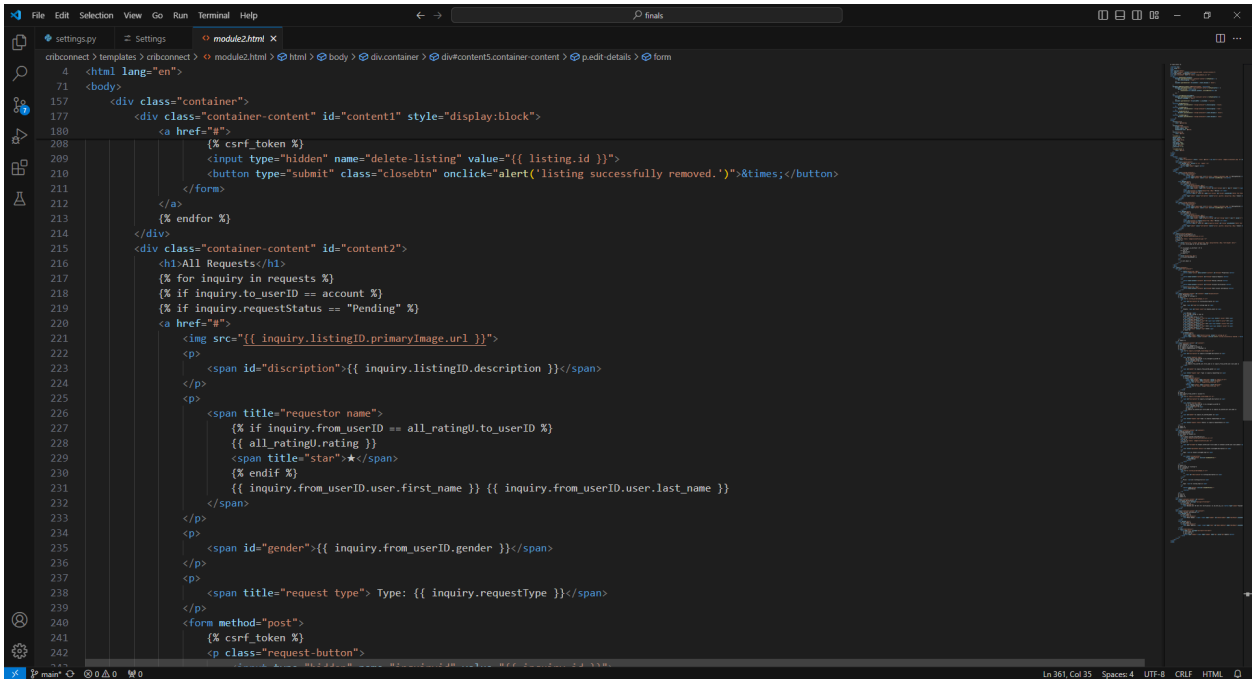
```
<html lang="en">
<body>
    <main>
        <div class="rating-container">
            </div>
        </div>
        <div class="rating-container2">
            <div class="logo-container">
                    <div class="logo">
                        <a><img class="login-logo" src="{% static 'images/cribconnect.png' %}">Rating/Review</a>
                        <button type="button" class="close" onclick="closeRating2()">X</button>
                    </div>
                </div>
            <form method="post">
                {% csrf_token %}
                <div class="container-r">
                    <div style="margin-bottom: 10px;">
                        <label for="rating"><b>Rating:</b></label>
                        <input type="number" name="property-rating" id="user-rating" min="1" max="5" value="1"><span> star/s</span>
                    </div>
                    <label for="review"><b style="margin-top: 20px;">Review:</b></label>
                    <div style="margin-top: 5px;">
                        <textarea rows="4" cols="50" name="property-review" id="review" placeholder="Enter Your Review Message Here" style="object-position: bottom;"></textarea>
                    </div>
                  <button type="submit" class="rate-button" style="cursor: pointer; margin-top: 20px;">Submit</button>
                </div>
              </form>
            </div>
        </div>
    </main>
    <div class="profile-container">
        {% if account.profilePicture %}
        <img src="{{ account.profilePicture.url }}">
        {% else %}
        <img src="{% static 'images/accounticon.png' %}">
        {% endif %}
        <ul>
            <li style="font-size: x-large; margin-top: 10px; margin-bottom: 10px; font-weight: bold;">
                {{ user.first_name }} {{ user.last_name }}
            </li>
```



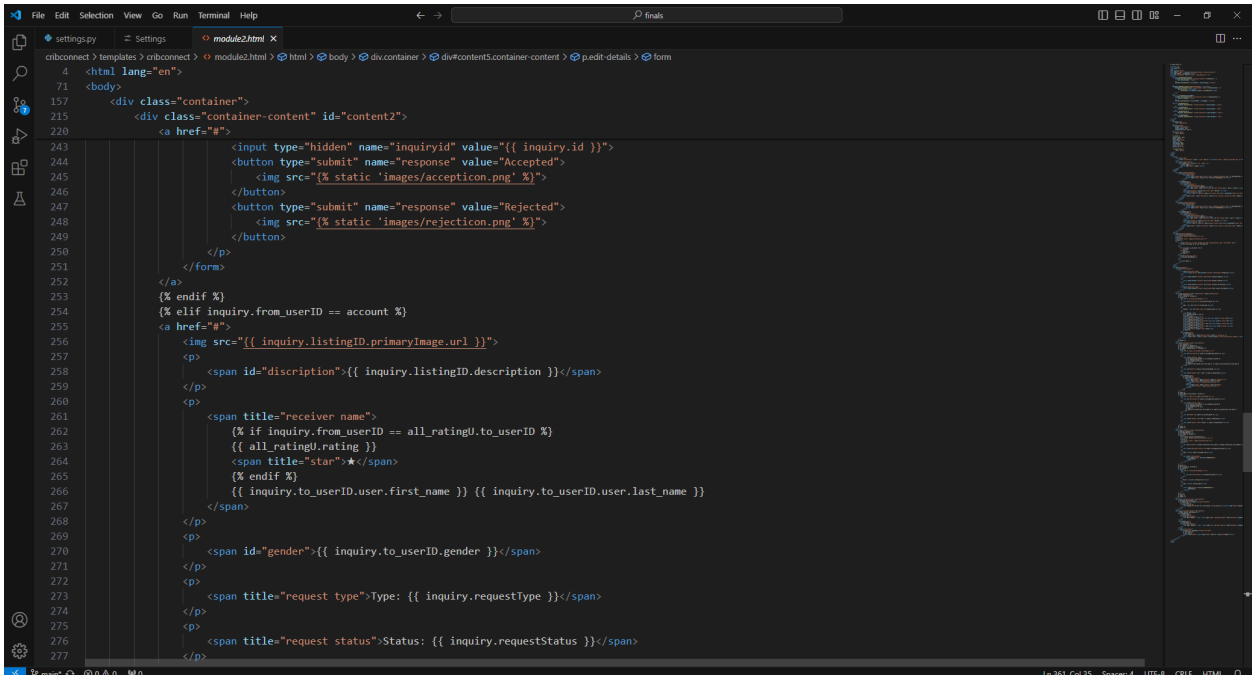
```
<html lang="en">
<body>
    <div class="profile-container">
        <ul>
            <li style="font-size: x-large; margin-top: 10px; margin-bottom: 10px; font-weight: bold;">
                {{ user.first_name }} {{ user.last_name }}
            </li>
            <li>
                {% if account.is_verified != 0 %}
                    Verified
                {% else %}
                    Unverified
                {% endif %}
            </li>
            <li style="margin-top: 5px;">
                {{ account.phoneNumber }}
            </li>
            <li>
                {{ user.email }}
            </li>
        </ul>
    </div>
    </main>
    <div class="container">
        <div class="nav-container">
            <ul>
                <li style="margin-top: 50px">
                    <button class="active" data-content="content1" id="btnnum1">Properties</button>
                </li>
                <li>
                    <button data-content="content2" id="btnnum2">Inquiry Requests</button>
                </li>
                <li>
                    <button data-content="content3" id="btnnum3">Rating & Review</button>
                </li>
                <li>
                    <button data-content="content4" id="btnnum4">Account Verification</button>
                </li>
                <li style="margin-top: 70px">
                    <button data-content="content5" id="btnnum5">Edit Account Information</button>
```



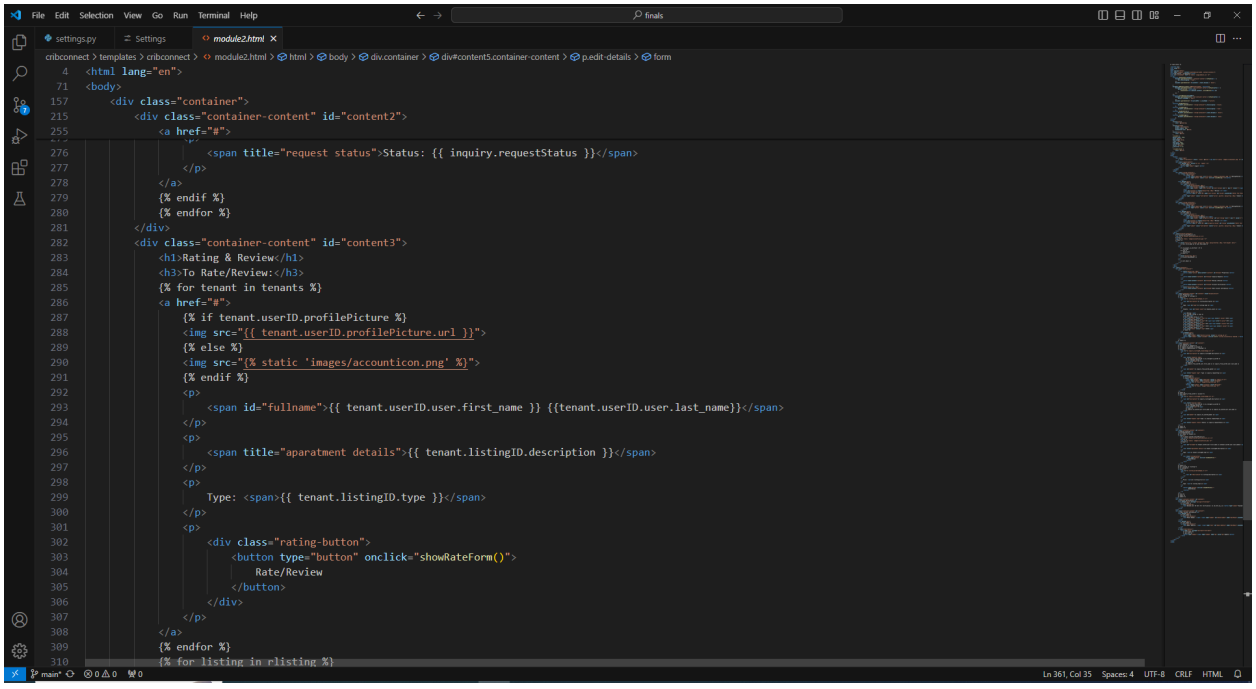
```
<html lang="en">
<body>
    <div class="container">
        <div class="nav-container">
            </ul>
        </div>
        <div class="container-content" id="content1" style="display:block">
            <h1>All Properties</h1>
            {% for listing in listings %}
            <a href="#">
                <img src="{{ listing.primaryImage.url }}">
                <p>
                    <span id="description">{{ listing.description }}</span>
                </p>
                <p>
                    Type: <span id="type">{{ listing.type }}</span>
                </p>
                <p>
                    Tenants: <span id="tenant count">{{ tenants_count }}</span>
                </p>
                <p>
                    <span>Rating:</span>
                    {% if ratingL.rating == none %}
                    <span>N/A</span>
                    {% elif ratingL.rating == 1 %}
                    <span class="stars" title="1 star">★</span><span title="4 nostar">★★★★</span>
                    {% elif ratingL.rating == 2 %}
                    <span class="stars" title="2 star">★★</span><span title="3 nostar">★★★</span>
                    {% elif ratingL.rating == 3 %}
                    <span class="stars" title="3 star">★★★</span><span title="2 nostar">★★</span>
                    {% elif ratingL.rating == 4 %}
                    <span class="stars" title="4 star">★★★★</span><span title="1 nostar">★</span>
                    {% elif ratingL.rating == 5 %}
                    <span class="stars" title="5 star">★★★★★</span>
                    {% endif %}
                </p>
                <form method="post">
                    {% csrf_token %}
                    <input type="hidden" name="delete-listing" value="{{ listing.id }}">
                    <button type="submit" class="closebtn" onclick="alert('listing successfully removed.')">&times;</button>
```



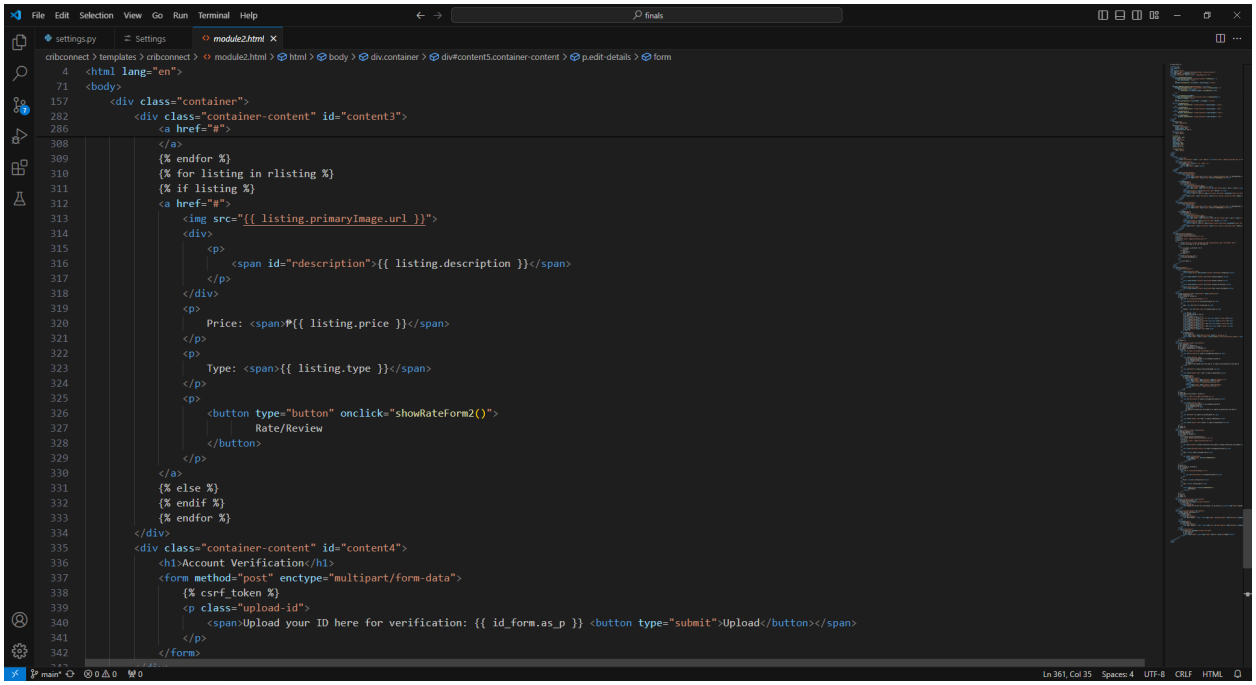
```
File Edit Selection View Go Run Terminal Help
finals
settings.py | Settings | module2.html
cribconnect > templates > cribconnect > module2.html > html > body > div.container > div#content5.container-content > p.edit-details > form
  4   <html lang="en">
 71   <body>
157       <div class="container">
177           <div class="container-content" id="content1" style="display:block">
180               <a href="#">
208                   {% csrf_token %}
209                   <input type="hidden" name="delete-listing" value="{{ listing.id }}">
210                   <button type="submit" class="closebtn" onclick="alert('listing successfully removed.')">&times;</button>
211               </form>
212           </a>
213           {% endfor %}
214       </div>
215       <div class="container-content" id="content2">
216           <h1>All Requests</h1>
217           {% for inquiry in requests %}
218           {% if inquiry.to_userID == account %}
219           {% if inquiry.requestStatus == "Pending" %}
220           <a href="#">
221               <img src="{{ inquiry.listingID.primaryImage.url }}">
222               <p>
223                   <span id="discription">{{ inquiry.listingID.description }}</span>
224               </p>
225               <p>
226                   <span title="requestor name">
227                       {% if inquiry.from_userID == all_ratingU.to_userID %}
228                       {{ all_ratingU.rating }}
229                       <span title="star">★</span>
230                       {% endif %}
231                       {{ inquiry.from_userID.user.first_name }} {{ inquiry.from_userID.user.last_name }}
232                   </span>
233               </p>
234               <p>
235                   <span id="gender">{{ inquiry.from_userID.gender }}</span>
236               </p>
237               <p>
238                   <span title="request type"> Type: {{ inquiry.requestType }}</span>
239               </p>
240               <form method="post">
241                   {% csrf_token %}
242                   <p class="request-button">
```



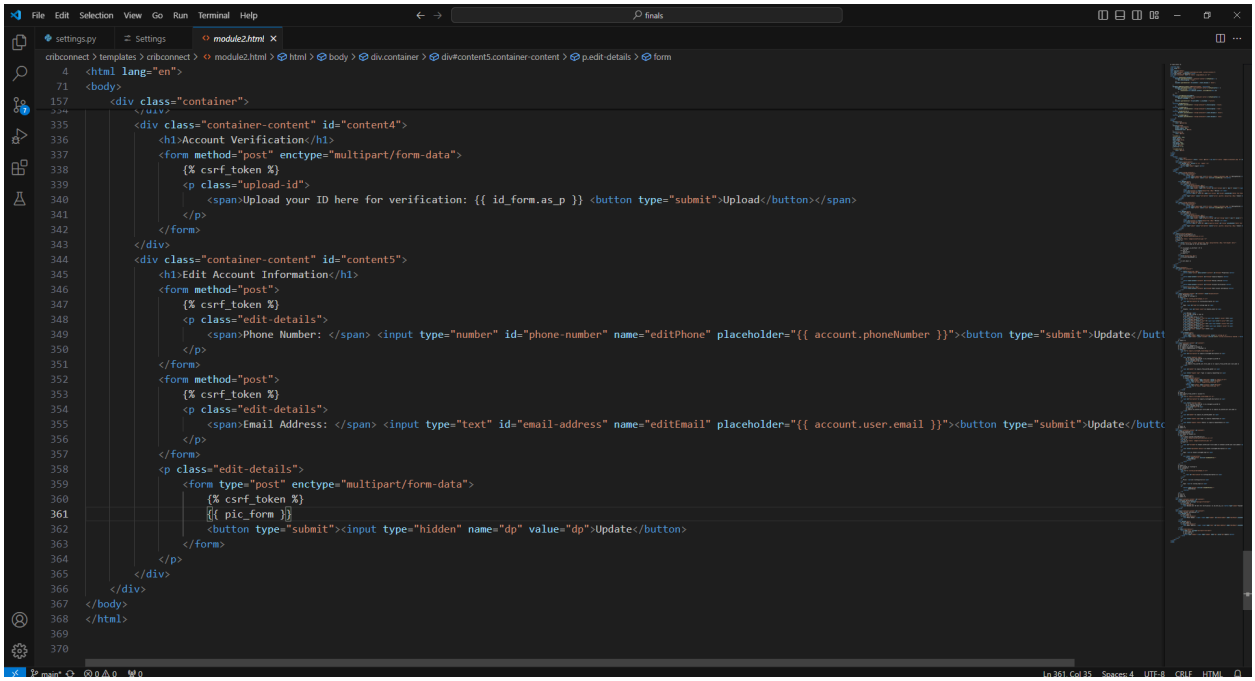
```
File Edit Selection View Go Run Terminal Help
finals
settings.py | Settings | module2.html
cribconnect > templates > cribconnect > module2.html > html > body > div.container > div#content5.container-content > p.edit-details > form
  4   <html lang="en">
 71   <body>
157       <div class="container">
215           <div class="container-content" id="content2">
220               <a href="#">
243                   <input type="hidden" name="inquiryid" value="{{ inquiry.id }}">
244                   <button type="submit" name="response" value="Accepted">
245                       <img src="{% static 'images/accepticon.png' %}">
246                   </button>
247                   <button type="submit" name="response" value="Rejected">
248                       <img src="{% static 'images/rejecticon.png' %}">
249                   </button>
250               </p>
251           </form>
252       </a>
253       {% endif %}
254       {% elif inquiry.from_userID == account %}
255       <a href="#">
256           <img src="{{ inquiry.listingID.primaryImage.url }}">
257           <p>
258               <span id="discription">{{ inquiry.listingID.description }}</span>
259           </p>
260           <p>
261               <span title="receiver name">
262                   {% if inquiry.from_userID == all_ratingU.to_userID %}
263                   {{ all_ratingU.rating }}
264                   <span title="star">★</span>
265                   {% endif %}
266                   {{ inquiry.to_userID.user.first_name }} {{ inquiry.to_userID.user.last_name }}
267               </span>
268           </p>
269           <p>
270               <span id="gender">{{ inquiry.to_userID.gender }}</span>
271           </p>
272           <p>
273               <span title="request type">Type: {{ inquiry.requestType }}</span>
274           </p>
275           <p>
276               <span title="request status">Status: {{ inquiry.requestStatus }}</span>
277           </p>
```



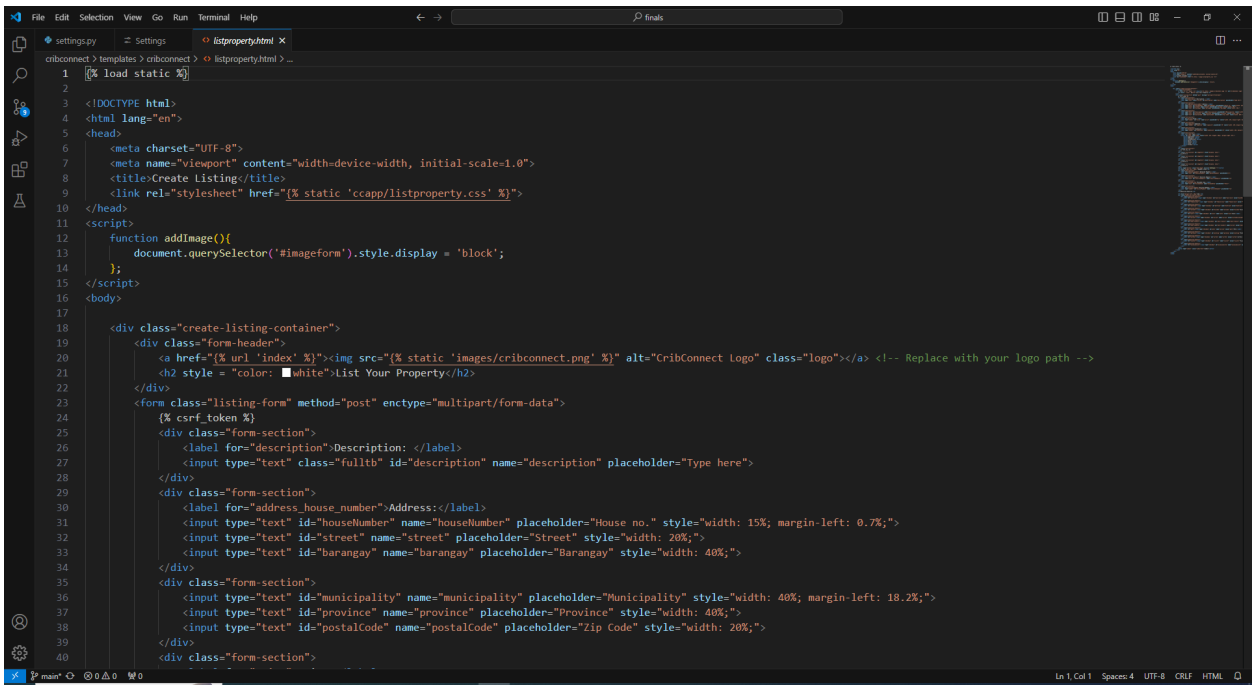
```
<html lang="en">
<body>
    <div class="container">
        <div class="container-content" id="content2">
            <a href="#">
                <p>
                    <span title="request status">Status: {{ inquiry.requestStatus }}</span>
                </p>
            </a>
            {% endif %}
            {% endfor %}
        </div>
        <div class="container-content" id="content3">
            <h1>Rating & Review</h1>
            <h3>To Rate/Review:</h3>
            {% for tenant in tenants %}
            <a href="#">
                {% if tenant.userID.profilePicture %}
                <img src="{{ tenant.userID.profilePicture.url }}">
                {% else %}
                <img src="{% static 'images/accounticon.png' %}">
                {% endif %}
                <p>
                    <span id="fullname">{{ tenant.userID.user.first_name }} {{tenant.userID.user.last_name}}</span>
                </p>
                <p>
                    <span title="aparatment details">{{ tenant.listingID.description }}</span>
                </p>
                <p>
                    Type: <span>{{ tenant.listingID.type }}</span>
                </p>
                <p>
                    <div class="rating-button">
                        <button type="button" onclick="showRateForm()">
                            Rate/Review
                        </button>
                    </div>
                </p>
            </a>
            {% endfor %}
            {% for listing in rlisting %}
```



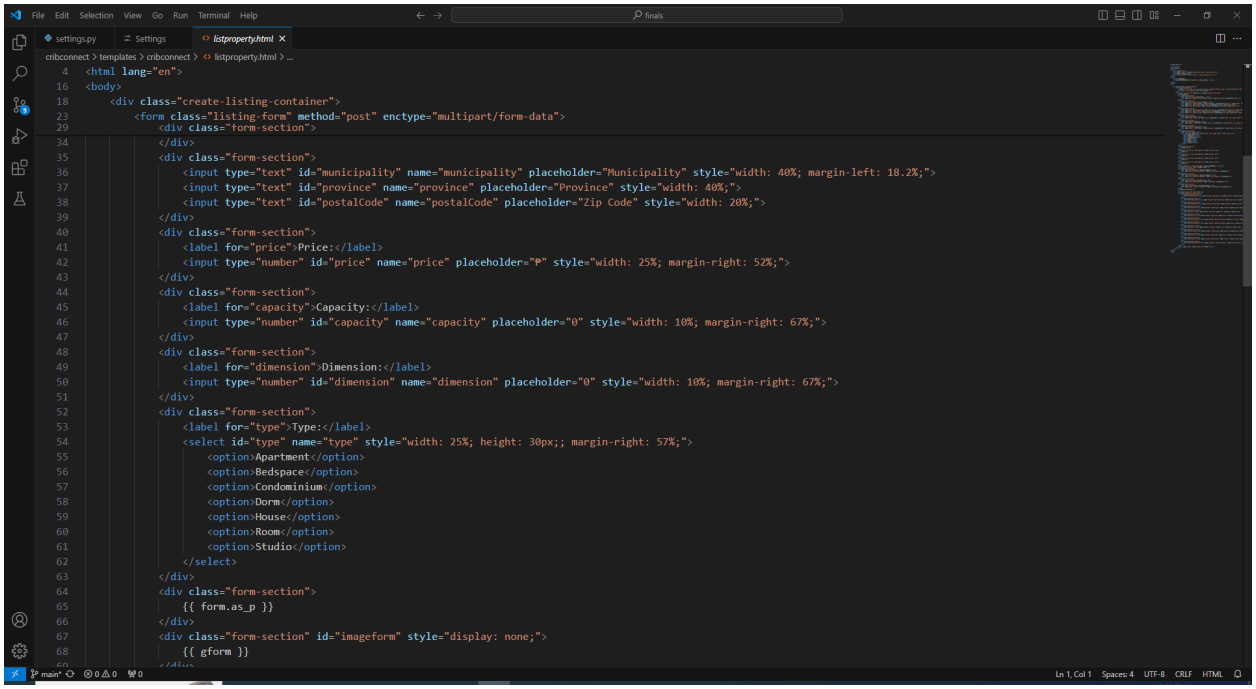
```
<html lang="en">
<body>
    <div class="container">
        <div class="container-content" id="content3">
            <a href="#">
            </a>
            {% endfor %}
            {% for listing in rlisting %}
            {% if listing %}
            <a href="#">
                <img src="{{ listing.primaryImage.url }}">
                <div>
                    <p>
                        <span id="rdescription">{{ listing.description }}</span>
                    </p>
                </div>
                <p>
                    Price: <span>₱{{ listing.price }}</span>
                </p>
                <p>
                    Type: <span>{{ listing.type }}</span>
                </p>
                <p>
                    <button type="button" onclick="showRateForm2()">
                            Rate/Review
                    </button>
                </p>
            </a>
            {% else %}
            {% endif %}
            {% endfor %}
        </div>
        <div class="container-content" id="content4">
            <h1>Account Verification</h1>
            <form method="post" enctype="multipart/form-data">
                {% csrf_token %}
                <p class="upload-id">
                    <span>Upload your ID here for verification: {{ id_form.as_p }} <button type="submit">Upload</button></span>
                </p>
            </form>
```



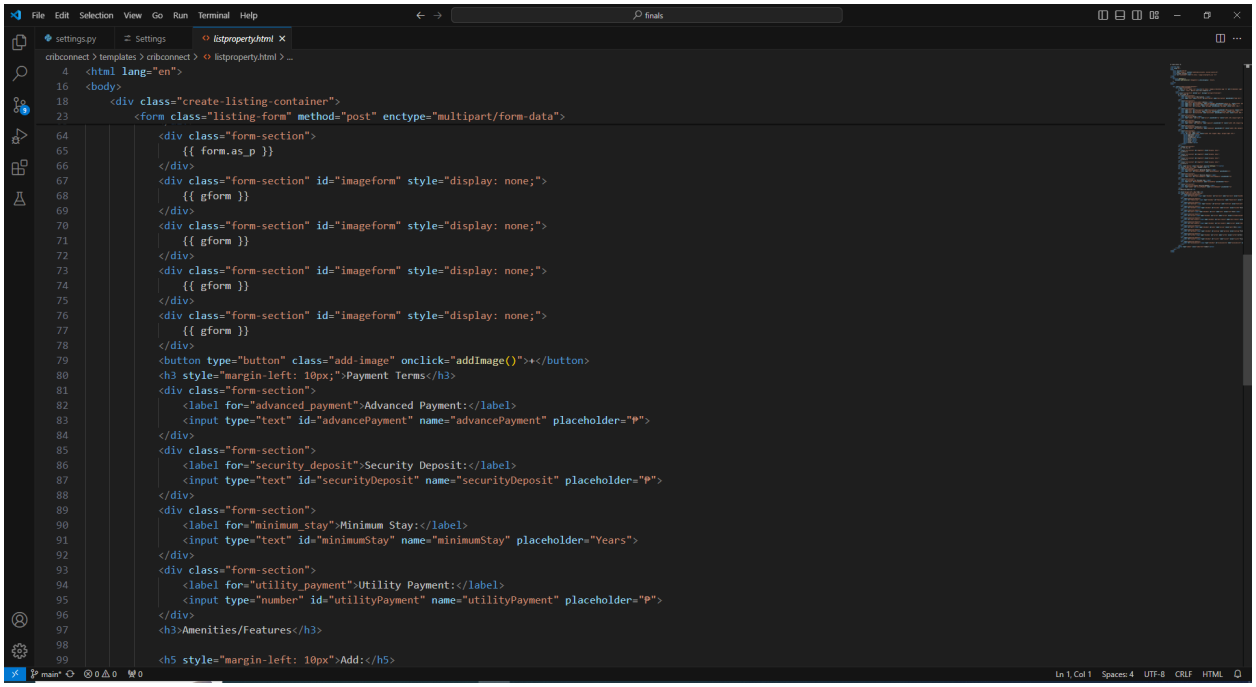
```
<html lang="en">
<body>
    <div class="container">
        <div class="container-content" id="content4">
            <h1>Account Verification</h1>
            <form method="post" enctype="multipart/form-data">
                {% csrf_token %}
                <p class="upload-id">
                    <span>Upload your ID here for verification: {{ id_form.as_p }} <button type="submit">Upload</button></span>
                </p>
            </form>
        </div>
        <div class="container-content" id="content5">
            <h1>Edit Account Information</h1>
            <form method="post">
                {% csrf_token %}
                <p class="edit-details">
                    <span>Phone Number: </span> <input type="number" id="phone-number" name="editPhone" placeholder="{{ account.phoneNumber }}"><button type="submit">Update</butt
                </p>
            </form>
            <form method="post">
                {% csrf_token %}
                <p class="edit-details">
                    <span>Email Address: </span> <input type="text" id="email-address" name="editEmail" placeholder="{{ account.user.email }}"><button type="submit">Update</butto
                </p>
            </form>
            <p class="edit-details">
                <form type="post" enctype="multipart/form-data">
                    {% csrf_token %}
                    {{ pic_form }}
                    <button type="submit"><input type="hidden" name="dp" value="dp">Update</button>
                </form>
            </p>
        </div>
    </div>
</body>
</html>
```



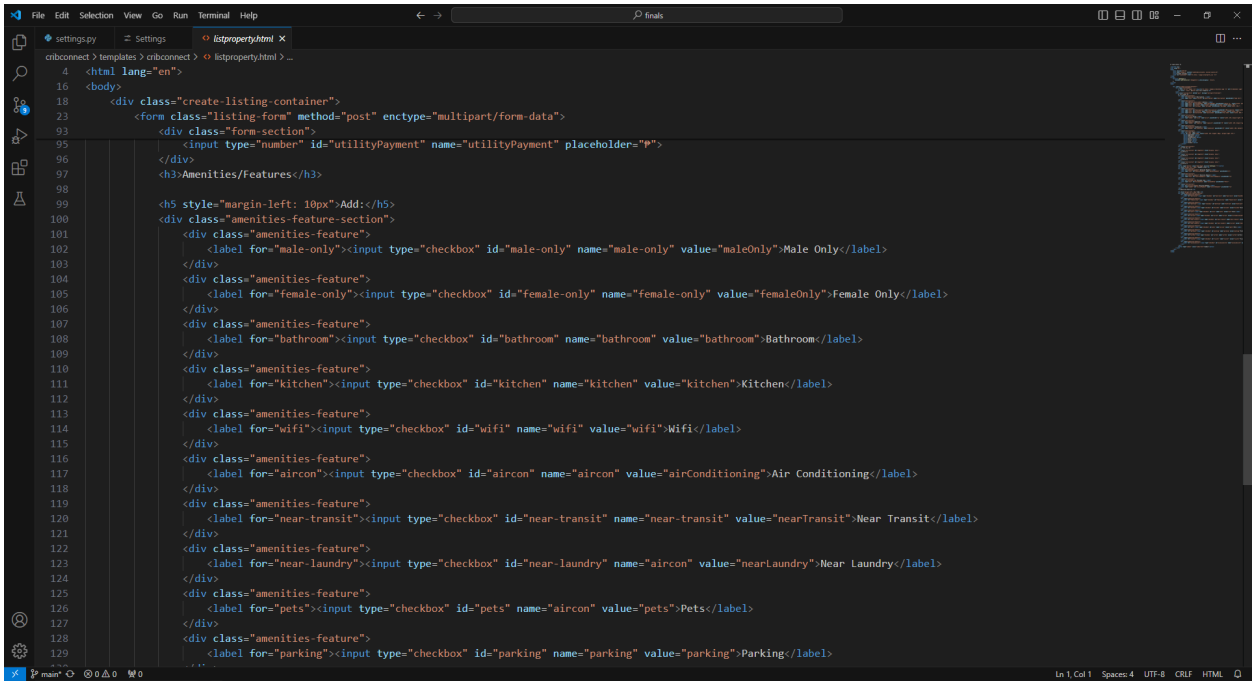
```
{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Create Listing</title>
    <link rel="stylesheet" href="{% static 'ccapp/listproperty.css' %}">
</head>
<script>
    function addImage(){
        document.querySelector('#imageform').style.display = 'block';
    };
</script>
<body>

    <div class="create-listing-container">
        <div class="form-header">
            <a href="{% url 'index' %}"><img src="{% static 'images/cribconnect.png' %}" alt="CribConnect Logo" class="logo"></a> <!-- Replace with your logo path -->
            <h2 style = "color: white">List Your Property</h2>
        </div>
        <form class="listing-form" method="post" enctype="multipart/form-data">
            {% csrf_token %}
            <div class="form-section">
                <label for="description">Description: </label>
                <input type="text" class="fulltb" id="description" name="description" placeholder="Type here">
            </div>
            <div class="form-section">
                <label for="address_house_number">Address:</label>
                <input type="text" id="houseNumber" name="houseNumber" placeholder="House no." style="width: 15%; margin-left: 0.7%;">
                <input type="text" id="street" name="street" placeholder="Street" style="width: 20%;">
                <input type="text" id="barangay" name="barangay" placeholder="Barangay" style="width: 40%;">
            </div>
            <div class="form-section">
                <input type="text" id="municipality" name="municipality" placeholder="Municipality" style="width: 40%; margin-left: 18.2%;">
                <input type="text" id="province" name="province" placeholder="Province" style="width: 40%;">
                <input type="text" id="postalCode" name="postalCode" placeholder="Zip Code" style="width: 20%;">
            </div>
            <div class="form-section">
```



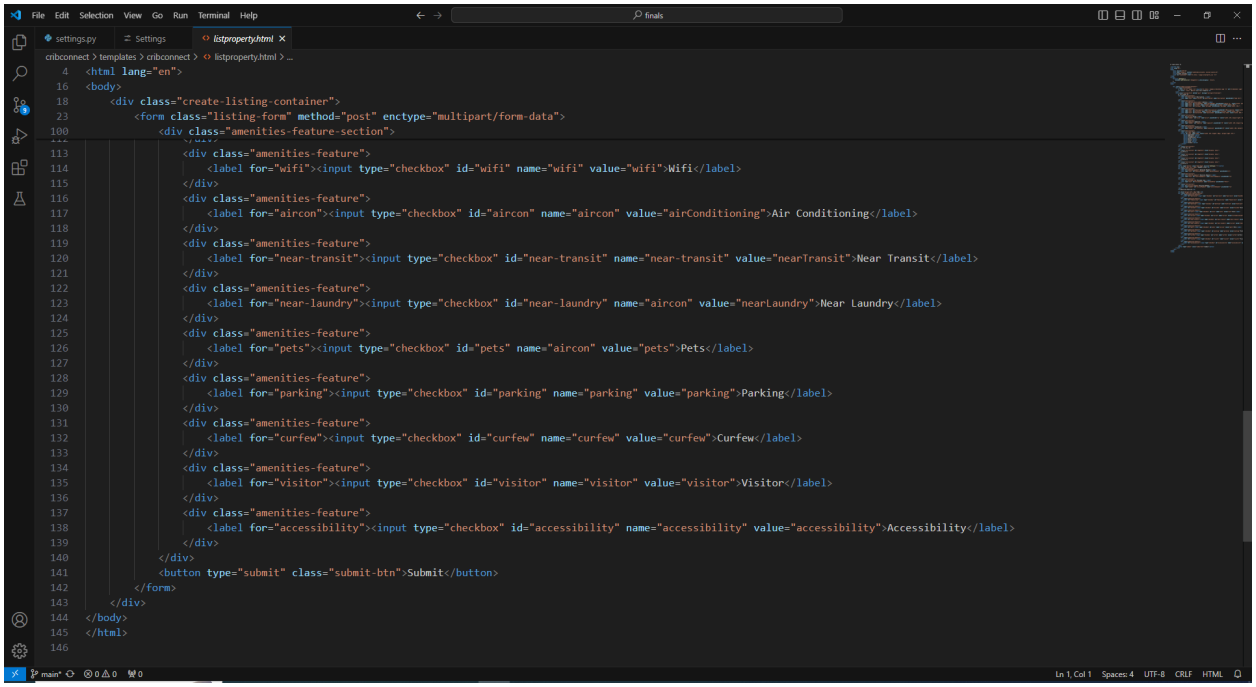
```
<html lang="en">
<body>
    <div class="create-listing-container">
        <form class="listing-form" method="post" enctype="multipart/form-data">
            <div class="form-section">
            </div>
            <div class="form-section">
                <input type="text" id="municipality" name="municipality" placeholder="Municipality" style="width: 40%; margin-left: 18.2%;">
                <input type="text" id="province" name="province" placeholder="Province" style="width: 40%;">
                <input type="text" id="postalCode" name="postalCode" placeholder="Zip Code" style="width: 20%;">
            </div>
            <div class="form-section">
                <label for="price">Price:</label>
                <input type="number" id="price" name="price" placeholder="₱" style="width: 25%; margin-right: 52%;">
            </div>
            <div class="form-section">
                <label for="capacity">Capacity:</label>
                <input type="number" id="capacity" name="capacity" placeholder="0" style="width: 10%; margin-right: 67%;">
            </div>
            <div class="form-section">
                <label for="dimension">Dimension:</label>
                <input type="number" id="dimension" name="dimension" placeholder="0" style="width: 10%; margin-right: 67%;">
            </div>
            <div class="form-section">
                <label for="type">Type:</label>
                <select id="type" name="type" style="width: 25%; height: 30px;; margin-right: 57%;">
                    <option>Apartment</option>
                    <option>Bedspace</option>
                    <option>Condominium</option>
                    <option>Dorm</option>
                    <option>House</option>
                    <option>Room</option>
                    <option>Studio</option>
                </select>
            </div>
            <div class="form-section">
                {{ form.as_p }}
            </div>
            <div class="form-section" id="imageform" style="display: none;">
                {{ gform }}
```



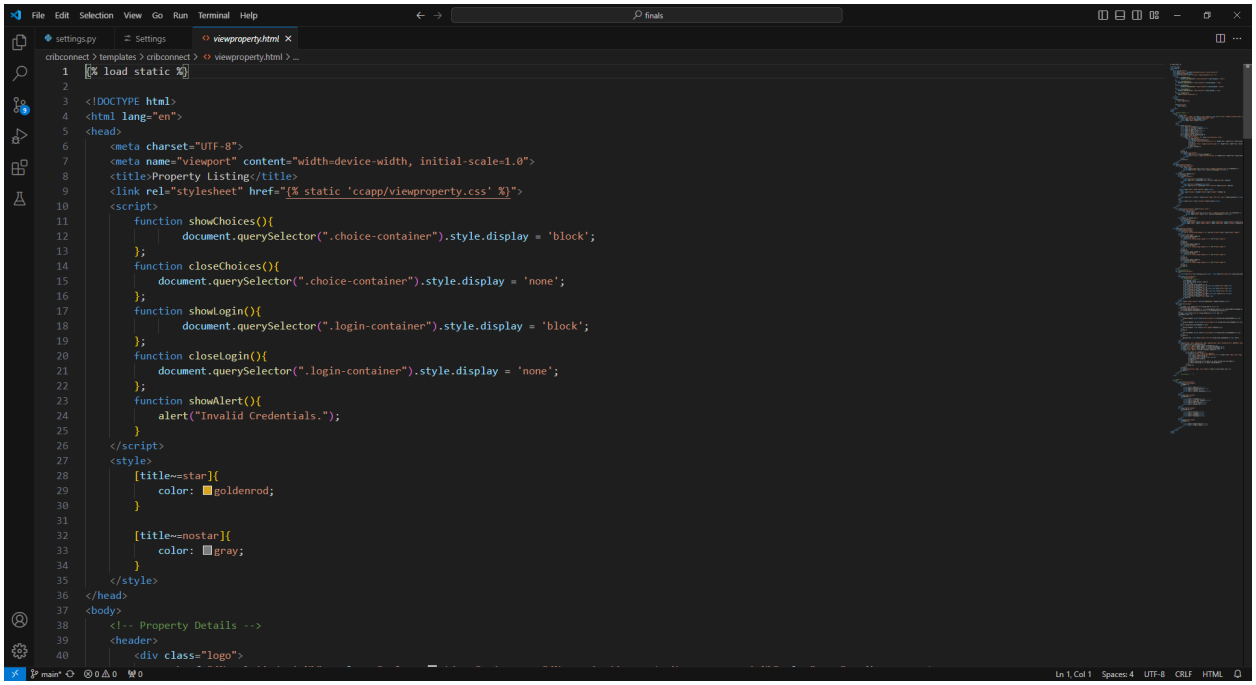
```
<html lang="en">
<body>
    <div class="create-listing-container">
        <form class="listing-form" method="post" enctype="multipart/form-data">
            <div class="form-section">
                {{ form.as_p }}
            </div>
            <div class="form-section" id="imageform" style="display: none;">
                {{ gform }}
            </div>
            <div class="form-section" id="imageform" style="display: none;">
                {{ gform }}
            </div>
            <div class="form-section" id="imageform" style="display: none;">
                {{ gform }}
            </div>
            <div class="form-section" id="imageform" style="display: none;">
                {{ gform }}
            </div>
            <button type="button" class="add-image" onclick="addImage()">+</button>
            <h3 style="margin-left: 10px;">Payment Terms</h3>
            <div class="form-section">
                <label for="advanced_payment">Advanced Payment:</label>
                <input type="text" id="advancePayment" name="advancePayment" placeholder="₱">
            </div>
            <div class="form-section">
                <label for="security_deposit">Security Deposit:</label>
                <input type="text" id="securityDeposit" name="securityDeposit" placeholder="₱">
            </div>
            <div class="form-section">
                <label for="minimum_stay">Minimum Stay:</label>
                <input type="text" id="minimumStay" name="minimumStay" placeholder="Years">
            </div>
            <div class="form-section">
                <label for="utility_payment">Utility Payment:</label>
                <input type="number" id="utilityPayment" name="utilityPayment" placeholder="₱">
            </div>
            <h3>Amenities/Features</h3>

            <h5 style="margin-left: 10px">Add:</h5>
```



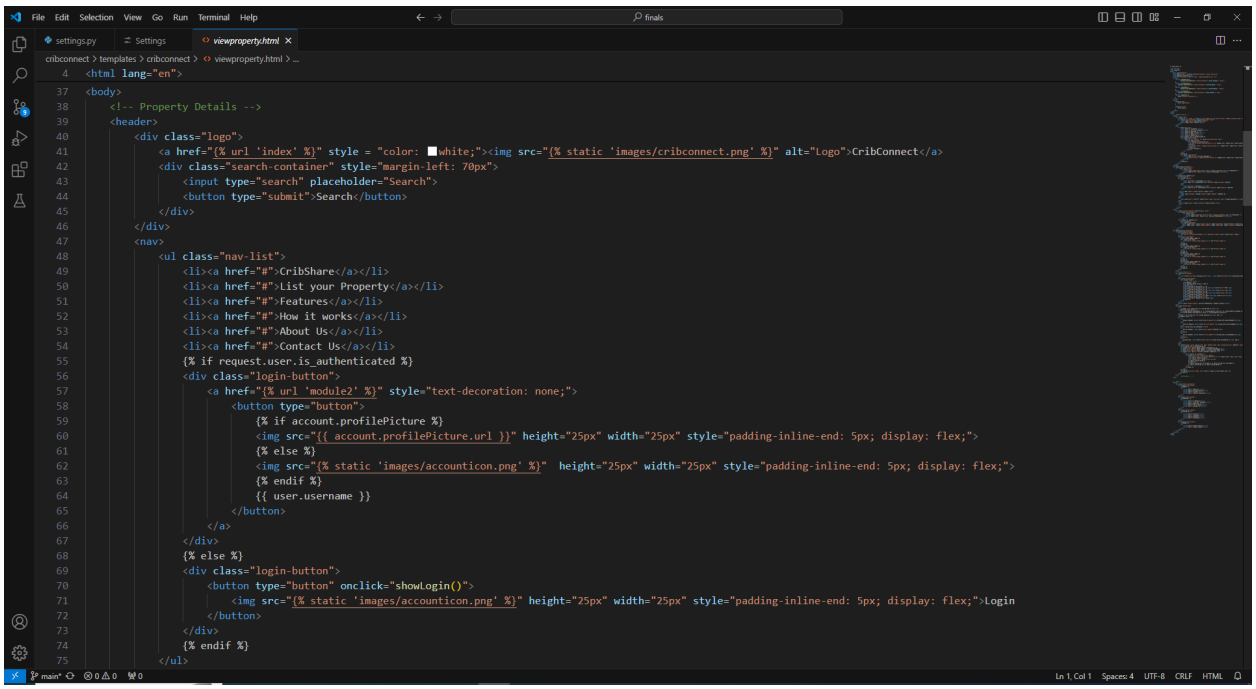
```
<html lang="en">
<body>
    <div class="create-listing-container">
        <form class="listing-form" method="post" enctype="multipart/form-data">
            <div class="form-section">
                <input type="number" id="utilityPayment" name="utilityPayment" placeholder="₱">
            </div>
            <h3>Amenities/Features</h3>

            <h5 style="margin-left: 10px">Add:</h5>
            <div class="amenities-feature-section">
                <div class="amenities-feature">
                    <label for="male-only"><input type="checkbox" id="male-only" name="male-only" value="maleOnly">Male Only</label>
                </div>
                <div class="amenities-feature">
                    <label for="female-only"><input type="checkbox" id="female-only" name="female-only" value="femaleOnly">Female Only</label>
                </div>
                <div class="amenities-feature">
                    <label for="bathroom"><input type="checkbox" id="bathroom" name="bathroom" value="bathroom">Bathroom</label>
                </div>
                <div class="amenities-feature">
                    <label for="kitchen"><input type="checkbox" id="kitchen" name="kitchen" value="kitchen">Kitchen</label>
                </div>
                <div class="amenities-feature">
                    <label for="wifi"><input type="checkbox" id="wifi" name="wifi" value="wifi">Wifi</label>
                </div>
                <div class="amenities-feature">
                    <label for="aircon"><input type="checkbox" id="aircon" name="aircon" value="airConditioning">Air Conditioning</label>
                </div>
                <div class="amenities-feature">
                    <label for="near-transit"><input type="checkbox" id="near-transit" name="near-transit" value="nearTransit">Near Transit</label>
                </div>
                <div class="amenities-feature">
                    <label for="near-laundry"><input type="checkbox" id="near-laundry" name="aircon" value="nearLaundry">Near Laundry</label>
                </div>
                <div class="amenities-feature">
                    <label for="pets"><input type="checkbox" id="pets" name="aircon" value="pets">Pets</label>
                </div>
                <div class="amenities-feature">
                    <label for="parking"><input type="checkbox" id="parking" name="parking" value="parking">Parking</label>
```



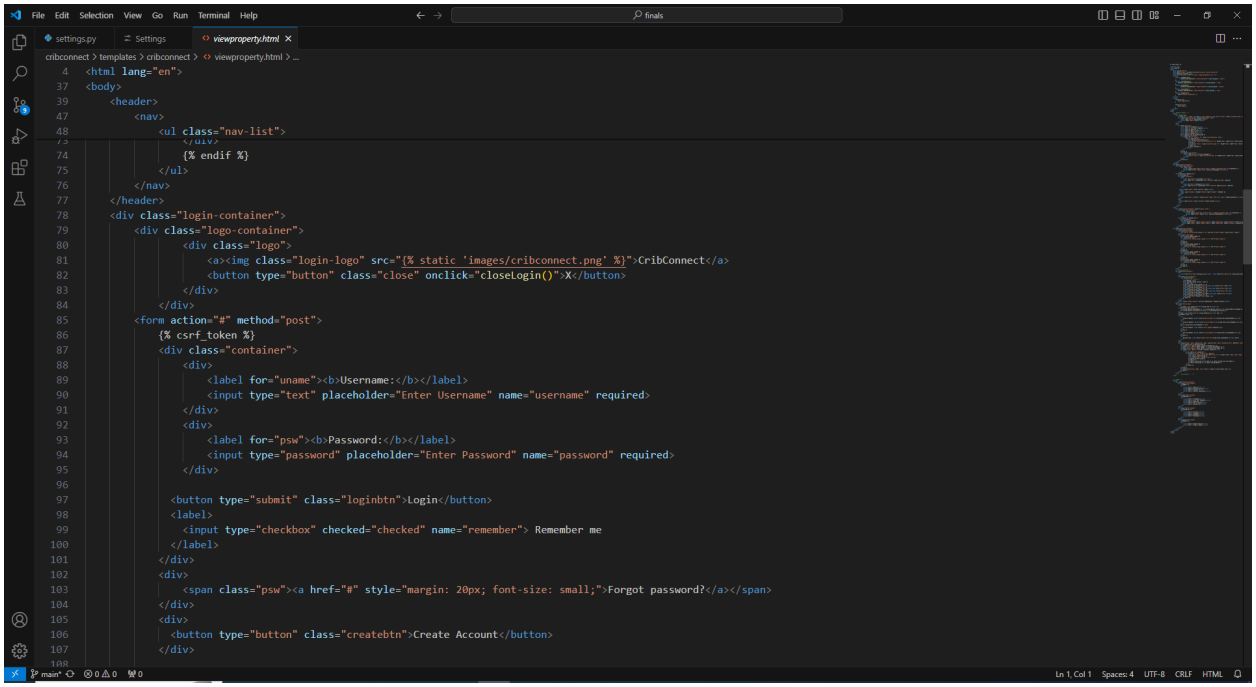
```
<html lang="en">
<body>
    <div class="create-listing-container">
        <form class="listing-form" method="post" enctype="multipart/form-data">
            <div class="amenities-feature-section">
                <div class="amenities-feature">
                    <label for="wifi"><input type="checkbox" id="wifi" name="wifi" value="wifi">Wifi</label>
                </div>
                <div class="amenities-feature">
                    <label for="aircon"><input type="checkbox" id="aircon" name="aircon" value="airConditioning">Air Conditioning</label>
                </div>
                <div class="amenities-feature">
                    <label for="near-transit"><input type="checkbox" id="near-transit" name="near-transit" value="nearTransit">Near Transit</label>
                </div>
                <div class="amenities-feature">
                    <label for="near-laundry"><input type="checkbox" id="near-laundry" name="aircon" value="nearLaundry">Near Laundry</label>
                </div>
                <div class="amenities-feature">
                    <label for="pets"><input type="checkbox" id="pets" name="aircon" value="pets">Pets</label>
                </div>
                <div class="amenities-feature">
                    <label for="parking"><input type="checkbox" id="parking" name="parking" value="parking">Parking</label>
                </div>
                <div class="amenities-feature">
                    <label for="curfew"><input type="checkbox" id="curfew" name="curfew" value="curfew">Curfew</label>
                </div>
                <div class="amenities-feature">
                    <label for="visitor"><input type="checkbox" id="visitor" name="visitor" value="visitor">Visitor</label>
                </div>
                <div class="amenities-feature">
                    <label for="accessibility"><input type="checkbox" id="accessibility" name="accessibility" value="accessibility">Accessibility</label>
                </div>
            </div>
            <button type="submit" class="submit-btn">Submit</button>
        </form>
    </div>
</body>
</html>
```



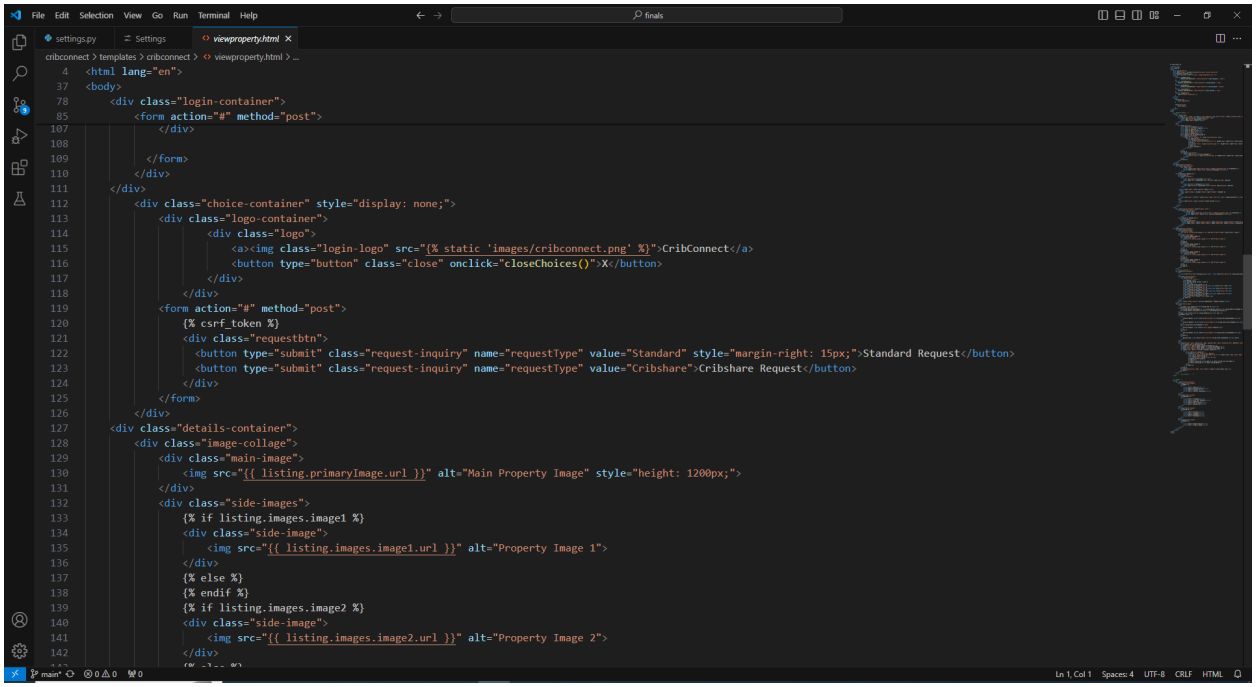
```
{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Property Listing</title>
    <link rel="stylesheet" href="{% static 'ccapp/viewproperty.css' %}">
    <script>
        function showChoices(){
                document.querySelector(".choice-container").style.display = 'block';
        };
        function closeChoices(){
            document.querySelector(".choice-container").style.display = 'none';
        };
        function showLogin(){
                document.querySelector(".login-container").style.display = 'block';
        };
        function closeLogin(){
            document.querySelector(".login-container").style.display = 'none';
        };
        function showAlert(){
            alert("Invalid Credentials.");
        }
    </script>
    <style>
        [title~=star]{
            color: goldenrod;
        }

        [title~=nostar]{
            color: gray;
        }
    </style>
</head>
<body>
    <!-- Property Details -->
    <header>
        <div class="logo">
```



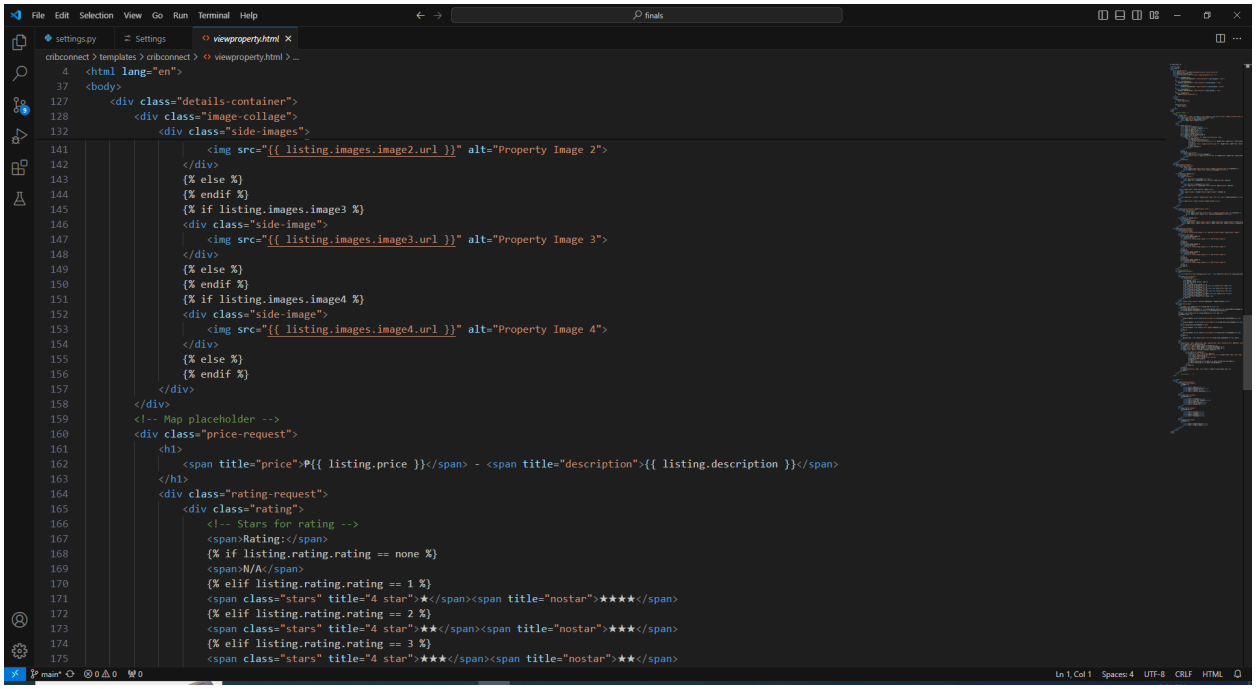
```
<html lang="en">
<body>
    <!-- Property Details -->
    <header>
        <div class="logo">
            <a href="{% url 'index' %}" style = "color: white;"><img src="{% static 'images/cribconnect.png' %}" alt="Logo">CribConnect</a>
            <div class="search-container" style="margin-left: 70px">
                <input type="search" placeholder="Search">
                <button type="submit">Search</button>
            </div>
        </div>
        <nav>
            <ul class="nav-list">
                <li><a href="#">CribShare</a></li>
                <li><a href="#">List your Property</a></li>
                <li><a href="#">Features</a></li>
                <li><a href="#">How it works</a></li>
                <li><a href="#">About Us</a></li>
                <li><a href="#">Contact Us</a></li>
                {% if request.user.is_authenticated %}
                <div class="login-button">
                    <a href="{% url 'module2' %}" style="text-decoration: none;">
                        <button type="button">
                            {% if account.profilePicture %}
                            <img src="{{ account.profilePicture.url }}" height="25px" width="25px" style="padding-inline-end: 5px; display: flex;">
                            {% else %}
                            <img src="{% static 'images/accounticon.png' %}"  height="25px" width="25px" style="padding-inline-end: 5px; display: flex;">
                            {% endif %}
                            {{ user.username }}
                        </button>
                    </a>
                </div>
                {% else %}
                <div class="login-button">
                    <button type="button" onclick="showLogin()">
                        <img src="{% static 'images/accounticon.png' %}" height="25px" width="25px" style="padding-inline-end: 5px; display: flex;">Login
                    </button>
                </div>
                {% endif %}
            </ul>
```



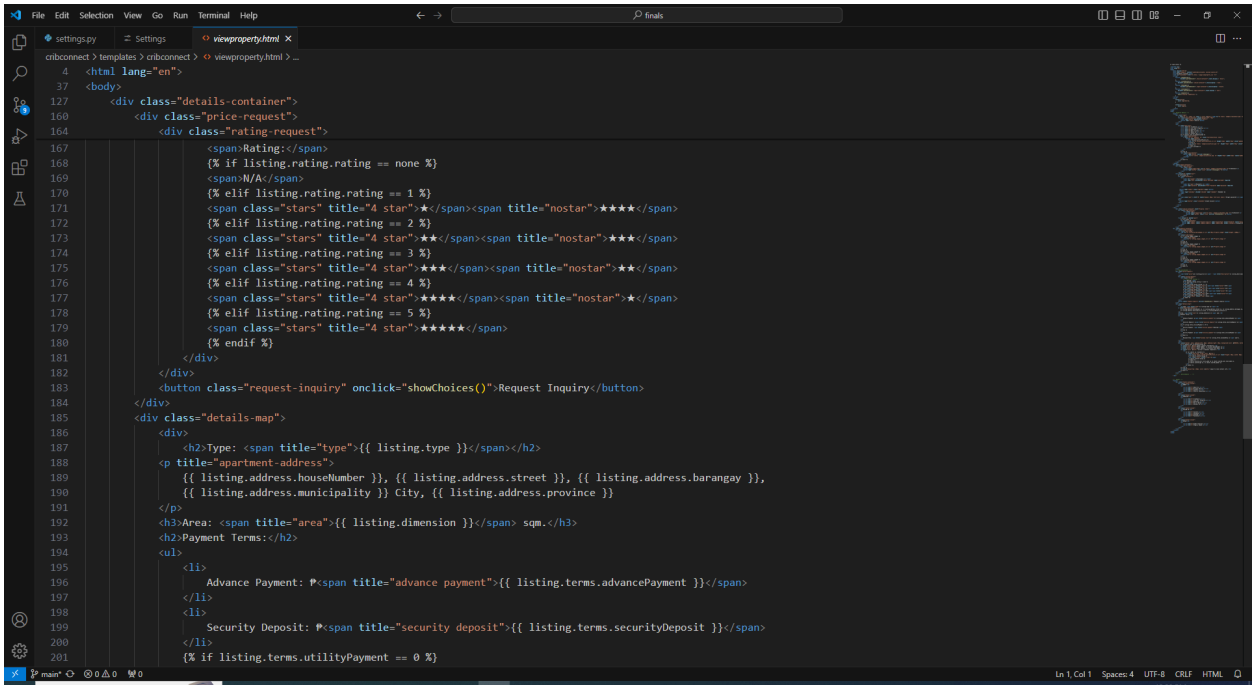
```
<html lang="en">
<body>
    <header>
        <nav>
            <ul class="nav-list">
                {% endif %}
            </ul>
        </nav>
    </header>
    <div class="login-container">
        <div class="logo-container">
            <div class="logo">
                <a><img class="login-logo" src="{% static 'images/cribconnect.png' %}">CribConnect</a>
                <button type="button" class="close" onclick="closeLogin()">X</button>
            </div>
        </div>
        <form action="#" method="post">
            {% csrf_token %}
            <div class="container">
                <div>
                    <label for="uname"><b>Username:</b></label>
                    <input type="text" placeholder="Enter Username" name="username" required>
                </div>
                <div>
                    <label for="psw"><b>Password:</b></label>
                    <input type="password" placeholder="Enter Password" name="password" required>
                </div>

              <button type="submit" class="loginbtn">Login</button>
              <label>
                <input type="checkbox" checked="checked" name="remember"> Remember me
              </label>
            </div>
            <div>
                <span class="psw"><a href="#" style="margin: 20px; font-size: small;">Forgot password?</a></span>
            </div>
            <div>
              <button type="button" class="createbtn">Create Account</button>
            </div>
```



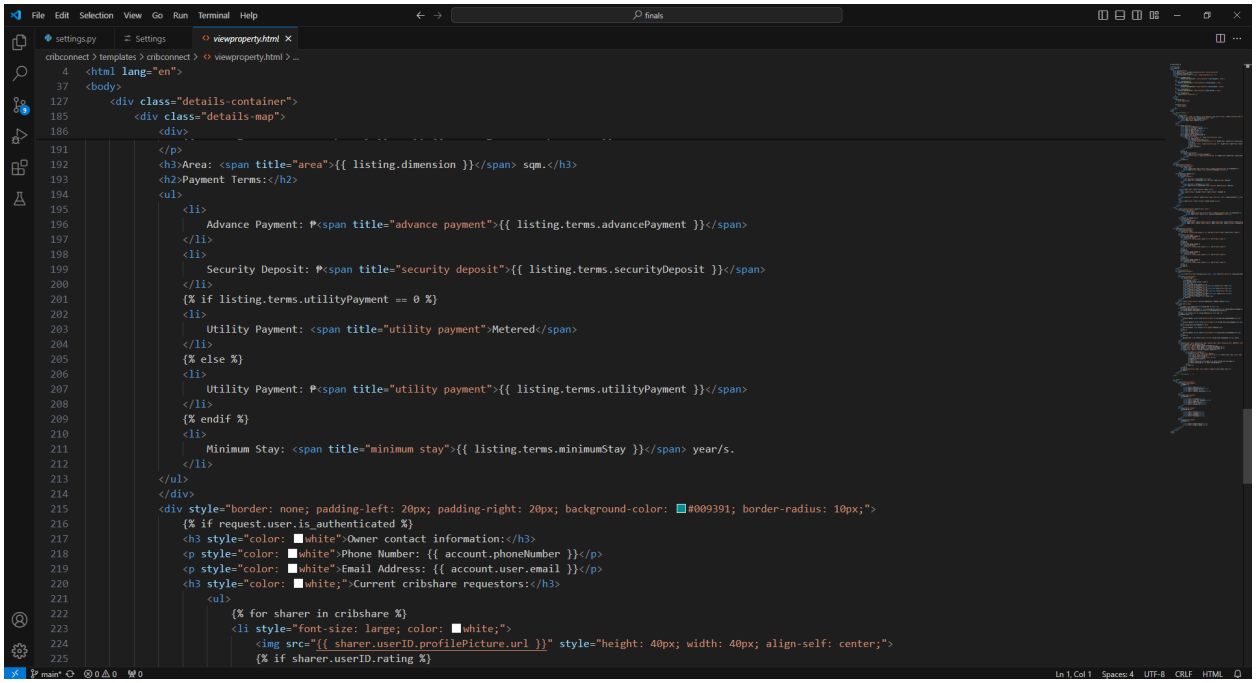
```
<html lang="en">
<body>
    <div class="login-container">
        <form action="#" method="post">
            </div>

          </form>
        </div>
    </div>
        <div class="choice-container" style="display: none;">
            <div class="logo-container">
                <div class="logo">
                    <a><img class="login-logo" src="{% static 'images/cribconnect.png' %}">CribConnect</a>
                    <button type="button" class="close" onclick="closeChoices()">X</button>
                </div>
            </div>
        <form action="#" method="post">
            {% csrf_token %}
            <div class="requestbtn">
              <button type="submit" class="request-inquiry" name="requestType" value="Standard" style="margin-right: 15px;">Standard Request</button>
              <button type="submit" class="request-inquiry" name="requestType" value="Cribshare">Cribshare Request</button>
            </div>
        </form>
    </div>
<div class="details-container">
    <div class="image-collage">
        <div class="main-image">
            <img src="{{ listing.primaryImage.url }}" alt="Main Property Image" style="height: 1200px;">
        </div>
        <div class="side-images">
            {% if listing.images.image1 %}
            <div class="side-image">
                <img src="{{ listing.images.image1.url }}" alt="Property Image 1">
            </div>
            {% else %}
            {% endif %}
            {% if listing.images.image2 %}
            <div class="side-image">
                <img src="{{ listing.images.image2.url }}" alt="Property Image 2">
            </div>
```



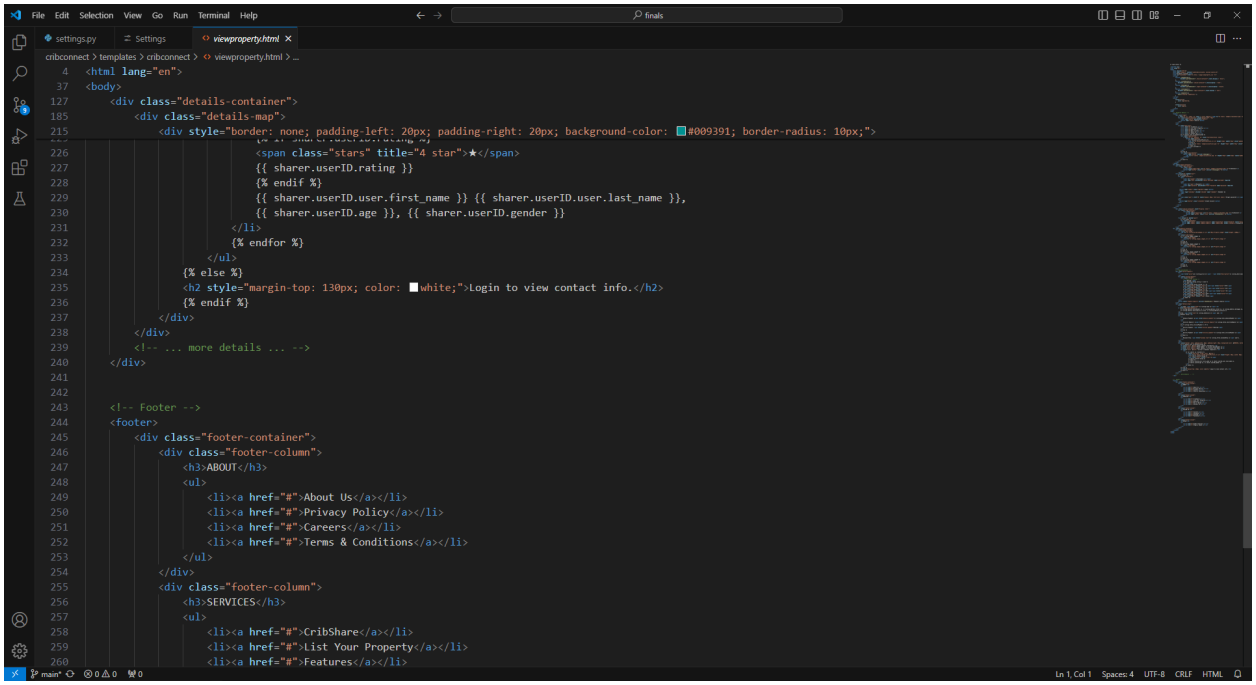
```
<html lang="en">
<body>
    <div class="details-container">
        <div class="image-collage">
            <div class="side-images">
                <img src="{{ listing.images.image2.url }}" alt="Property Image 2">
            </div>
            {% else %}
            {% endif %}
            {% if listing.images.image3 %}
            <div class="side-image">
                <img src="{{ listing.images.image3.url }}" alt="Property Image 3">
            </div>
            {% else %}
            {% endif %}
            {% if listing.images.image4 %}
            <div class="side-image">
                <img src="{{ listing.images.image4.url }}" alt="Property Image 4">
            </div>
            {% else %}
            {% endif %}
        </div>
    </div>
    <!-- Map placeholder -->
    <div class="price-request">
        <h1>
            <span title="price">P{{ listing.price }}</span> - <span title="description">{{ listing.description }}</span>
        </h1>
        <div class="rating-request">
            <div class="rating">
                <!-- Stars for rating -->
                <span>Rating:</span>
                {% if listing.rating.rating == none %}
                <span>N/A</span>
                {% elif listing.rating.rating == 1 %}
                <span class="stars" title="4 star">★</span><span title="nostar">★★★★</span>
                {% elif listing.rating.rating == 2 %}
                <span class="stars" title="4 star">★★</span><span title="nostar">★★★</span>
                {% elif listing.rating.rating == 3 %}
                <span class="stars" title="4 star">★★★</span><span title="nostar">★★</span>
```



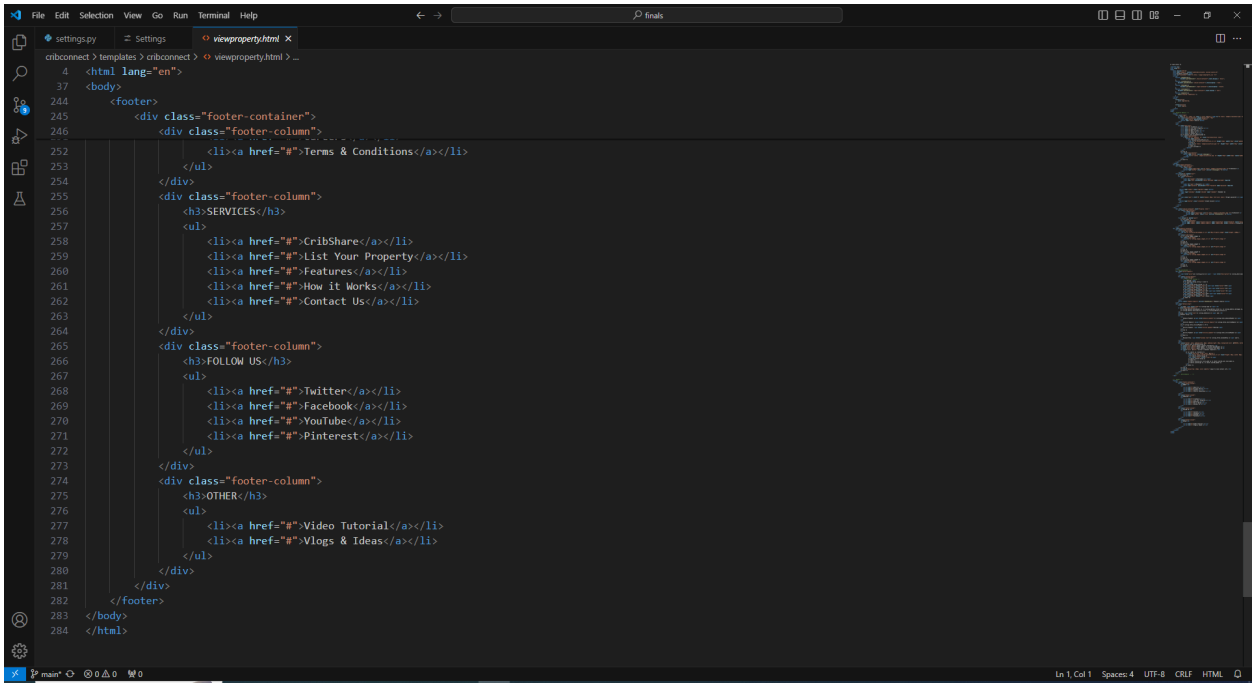
```
<html lang="en">
<body>
    <div class="details-container">
        <div class="price-request">
            <div class="rating-request">
                <span>Rating:</span>
                {% if listing.rating.rating == none %}
                <span>N/A</span>
                {% elif listing.rating.rating == 1 %}
                <span class="stars" title="4 star">★</span><span title="nostar">★★★★</span>
                {% elif listing.rating.rating == 2 %}
                <span class="stars" title="4 star">★★</span><span title="nostar">★★★</span>
                {% elif listing.rating.rating == 3 %}
                <span class="stars" title="4 star">★★★</span><span title="nostar">★★</span>
                {% elif listing.rating.rating == 4 %}
                <span class="stars" title="4 star">★★★★</span><span title="nostar">★</span>
                {% elif listing.rating.rating == 5 %}
                <span class="stars" title="4 star">★★★★★</span>
                {% endif %}
            </div>
        </div>
        <button class="request-inquiry" onclick="showChoices()">Request Inquiry</button>
    </div>
    <div class="details-map">
        <div>
            <h2>Type: <span title="type">{{ listing.type }}</span></h2>
        <p title="apartment-address">
            {{ listing.address.houseNumber }}, {{ listing.address.street }}, {{ listing.address.barangay }},
            {{ listing.address.municipality }} City, {{ listing.address.province }}
        </p>
        <h3>Area: <span title="area">{{ listing.dimension }}</span> sqm.</h3>
        <h2>Payment Terms:</h2>
        <ul>
            <li>
                Advance Payment: ₱<span title="advance payment">{{ listing.terms.advancePayment }}</span>
            </li>
            <li>
                Security Deposit: ₱<span title="security deposit">{{ listing.terms.securityDeposit }}</span>
            </li>
            {% if listing.terms.utilityPayment == 0 %}
```



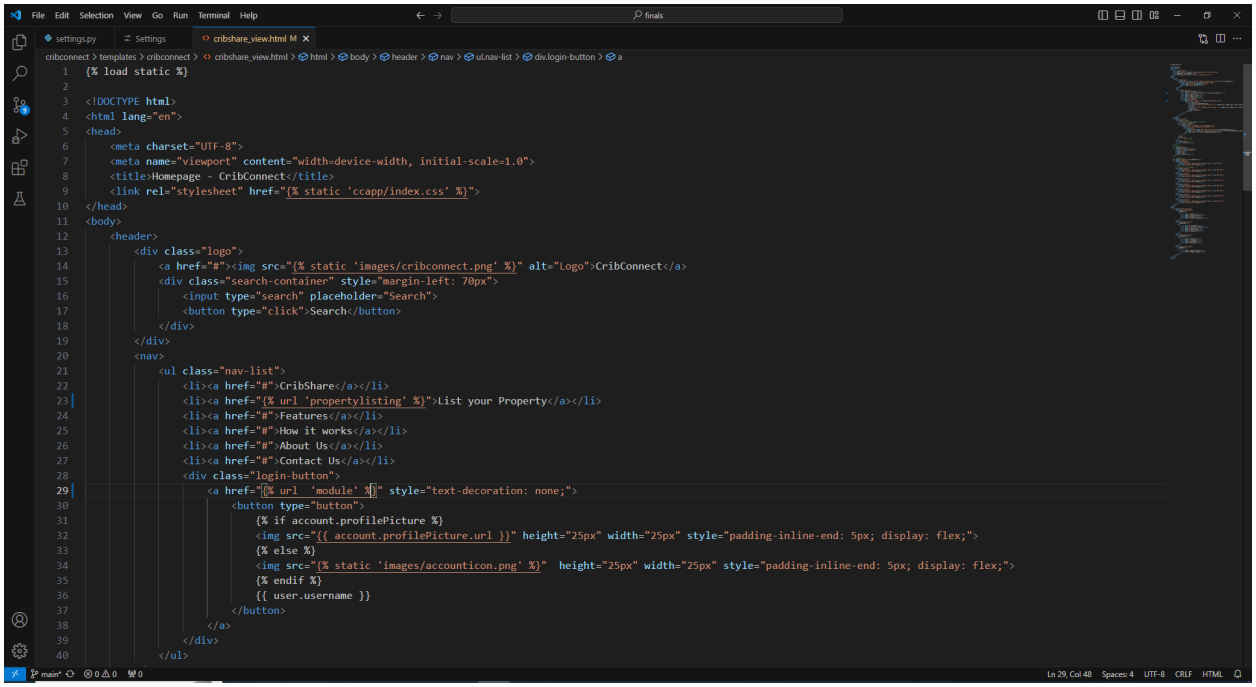
```
<html lang="en">
<body>
    <div class="details-container">
        <div class="details-map">
            <div>
            </p>
            <h3>Area: <span title="area">{{ listing.dimension }}</span> sqm.</h3>
            <h2>Payment Terms:</h2>
            <ul>
                <li>
                    Advance Payment: ₱<span title="advance payment">{{ listing.terms.advancePayment }}</span>
                </li>
                <li>
                    Security Deposit: ₱<span title="security deposit">{{ listing.terms.securityDeposit }}</span>
                </li>
                {% if listing.terms.utilityPayment == 0 %}
                <li>
                    Utility Payment: <span title="utility payment">Metered</span>
                </li>
                {% else %}
                <li>
                    Utility Payment: ₱<span title="utility payment">{{ listing.terms.utilityPayment }}</span>
                </li>
                {% endif %}
                <li>
                    Minimum Stay: <span title="minimum stay">{{ listing.terms.minimumStay }}</span> year/s.
                </li>
            </ul>
            </div>
            <div style="border: none; padding-left: 20px; padding-right: 20px; background-color: #009391; border-radius: 10px;">
                {% if request.user.is_authenticated %}
                <h3 style="color: white">Owner contact information:</h3>
                <p style="color: white">Phone Number: {{ account.phoneNumber }}</p>
                <p style="color: white">Email Address: {{ account.user.email }}</p>
                <h3 style="color: white;">Current cribshare requestors:</h3>
                    <ul>
                        {% for sharer in cribshare %}
                        <li style="font-size: large; color: white;">
                            <img src="{{ sharer.userID.profilePicture.url }}" style="height: 40px; width: 40px; align-self: center;">
                            {% if sharer.userID.rating %}
```



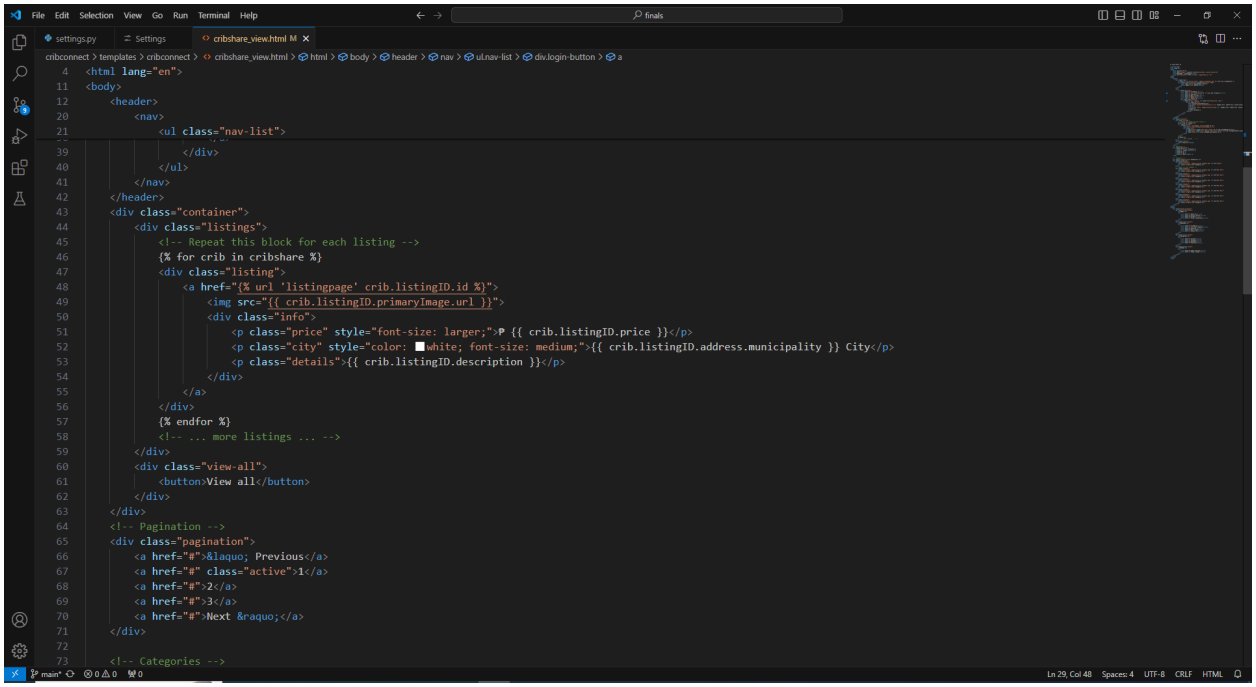
```
<html lang="en">
<body>
    <div class="details-container">
        <div class="details-map">
            <div style="border: none; padding-left: 20px; padding-right: 20px; background-color: #009391; border-radius: 10px;">
                            <span class="stars" title="4 star">★</span>
                            {{ sharer.userID.rating }}
                            {% endif %}
                            {{ sharer.userID.user.first_name }} {{ sharer.userID.user.last_name }},
                            {{ sharer.userID.age }}, {{ sharer.userID.gender }}
                        </li>
                        {% endfor %}
                    </ul>
                {% else %}
                <h2 style="margin-top: 130px; color: white;">Login to view contact info.</h2>
                {% endif %}
            </div>
        </div>
        <!-- ... more details ... -->
    </div>


    <!-- Footer -->
    <footer>
        <div class="footer-container">
            <div class="footer-column">
                <h3>ABOUT</h3>
                <ul>
                    <li><a href="#">About Us</a></li>
                    <li><a href="#">Privacy Policy</a></li>
                    <li><a href="#">Careers</a></li>
                    <li><a href="#">Terms & Conditions</a></li>
                </ul>
            </div>
            <div class="footer-column">
                <h3>SERVICES</h3>
                <ul>
                    <li><a href="#">CribShare</a></li>
                    <li><a href="#">List Your Property</a></li>
                    <li><a href="#">Features</a></li>
```

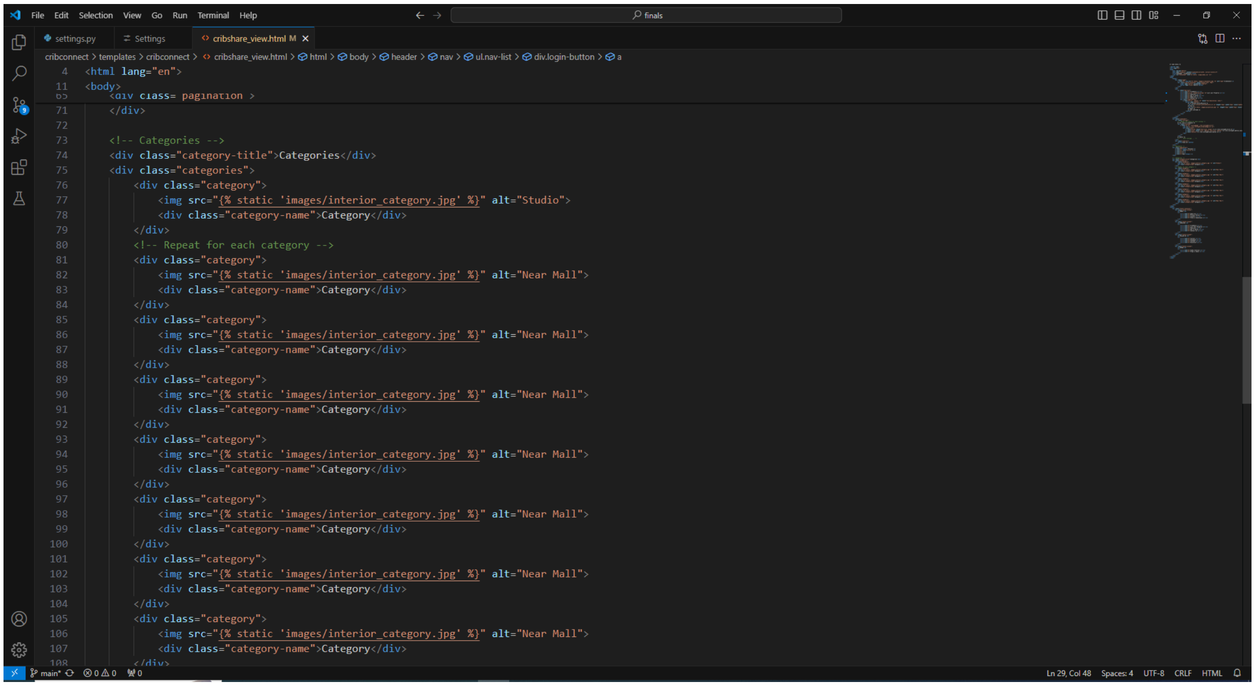


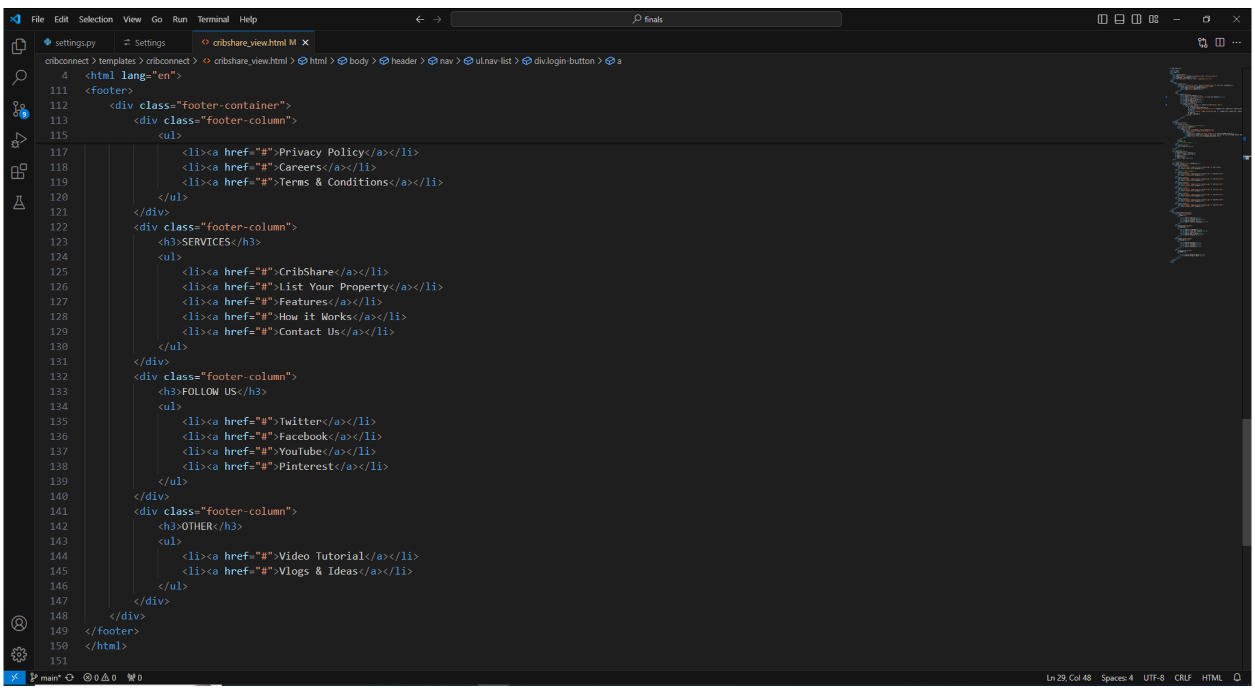
```
<html lang="en">
<body>
    <footer>
        <div class="footer-container">
            <div class="footer-column">
                    <li><a href="#">Terms & Conditions</a></li>
                </ul>
            </div>
            <div class="footer-column">
                <h3>SERVICES</h3>
                <ul>
                    <li><a href="#">CribShare</a></li>
                    <li><a href="#">List Your Property</a></li>
                    <li><a href="#">Features</a></li>
                    <li><a href="#">How it Works</a></li>
                    <li><a href="#">Contact Us</a></li>
                </ul>
            </div>
            <div class="footer-column">
                <h3>FOLLOW US</h3>
                <ul>
                    <li><a href="#">Twitter</a></li>
                    <li><a href="#">Facebook</a></li>
                    <li><a href="#">YouTube</a></li>
                    <li><a href="#">Pinterest</a></li>
                </ul>
            </div>
            <div class="footer-column">
                <h3>OTHER</h3>
                <ul>
                    <li><a href="#">Video Tutorial</a></li>
                    <li><a href="#">Vlogs & Ideas</a></li>
                </ul>
            </div>
        </div>
    </footer>
</body>
</html>
```



```
{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Homepage - CribConnect</title>
    <link rel="stylesheet" href="{% static 'ccapp/index.css' %}">
</head>
<body>
    <header>
        <div class="logo">
            <a href="#"><img src="{% static 'images/cribconnect.png' %}" alt="Logo">CribConnect</a>
            <div class="search-container" style="margin-left: 70px">
                <input type="search" placeholder="Search">
                <button type="click">Search</button>
            </div>
        </div>
        <nav>
            <ul class="nav-list">
                <li><a href="#">CribShare</a></li>
                <li><a href="{% url 'propertylisting' %}">List your Property</a></li>
                <li><a href="#">Features</a></li>
                <li><a href="#">How it works</a></li>
                <li><a href="#">About Us</a></li>
                <li><a href="#">Contact Us</a></li>
                <div class="login-button">
                    <a href="{% url 'module' %}" style="text-decoration: none;">
                        <button type="button">
                            {% if account.profilePicture %}
                            <img src="{{ account.profilePicture.url }}" height="25px" width="25px" style="padding-inline-end: 5px; display: flex;">
                            {% else %}
                            <img src="{% static 'images/accounticon.png' %}"  height="25px" width="25px" style="padding-inline-end: 5px; display: flex;">
                            {% endif %}
                            {{ user.username }}
                        </button>
                    </a>
                </div>
            </ul>
```


File Edit Selection View Go Run Terminal Help
finals
settings.py
Settings
cribshare_view.html M
cribconnect > templates > cribconnect > cribshare_view.html > html > body > header > nav > ul.nav-list > div.login-button > a
4 <html lang="en">
11 <body>
12 <header>
20 <nav>
21 <ul class="nav-list">
39
40
41 </nav>
42 </header>
43
44
45 <!-- Repeat this block for each listing -->
46 {% for crib in cribshare %}
47
48
49 <img src="{{ crib.listingID.primaryImage.url }}">
50
51 P {{ crib.listingID.price }}
52 {{ crib.listingID.address.municipality }} City
53 {{ crib.listingID.description }}
54
55
56
57 {% endfor %}
58 <!-- ... more listings ... -->
59
60
61 <button>View all</button>
62
63
64 <!-- Pagination -->
65
66 « Previous
67 1
68 2
69 3
70 Next »
71
72
73 <!-- Categories -->
main*
Ln 29, Col 48
Spaces: 4
UTF-8
CRLF
HTML


File Edit Selection View Go Run Terminal Help
finals
settings.py
Settings
cribshare_view.html M
cribconnect > templates > cribconnect > cribshare_view.html > html > body > header > nav > ul.nav-list > div.login-button > a
<html lang="en">
<body>
<div class="pagination">
</div>
<!-- Categories -->
<div class="category-title">Categories</div>
<div class="categories">
<div class="category">
<img src="{% static 'images/interior_category.jpg' %}" alt="Studio">
<div class="category-name">Category</div>
</div>
<!-- Repeat for each category -->
<div class="category">
<img src="{% static 'images/interior_category.jpg' %}" alt="Near Mall">
<div class="category-name">Category</div>
</div>
<div class="category">
<img src="{% static 'images/interior_category.jpg' %}" alt="Near Mall">
<div class="category-name">Category</div>
</div>
<div class="category">
<img src="{% static 'images/interior_category.jpg' %}" alt="Near Mall">
<div class="category-name">Category</div>
</div>
<div class="category">
<img src="{% static 'images/interior_category.jpg' %}" alt="Near Mall">
<div class="category-name">Category</div>
</div>
<div class="category">
<img src="{% static 'images/interior_category.jpg' %}" alt="Near Mall">
<div class="category-name">Category</div>
</div>
<div class="category">
<img src="{% static 'images/interior_category.jpg' %}" alt="Near Mall">
<div class="category-name">Category</div>
</div>
<div class="category">
<img src="{% static 'images/interior_category.jpg' %}" alt="Near Mall">
<div class="category-name">Category</div>
</div>
main* 0 0 0
Ln 29, Col 48 Spaces: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run Terminal Help
finals
settings.py
Settings
cribshare_view.html M
cribconnect > templates > cribconnect > cribshare_view.html > html > body > header > nav > ul.nav-list > div.login-button > a
<html lang="en">
<footer>
<div class="footer-container">
<div class="footer-column">

Privacy Policy
Careers
Terms & Conditions

</div>
<div class="footer-column">
<h3>SERVICES</h3>

CribShare
List Your Property
Features
How it Works
Contact Us

</div>
<div class="footer-column">
<h3>FOLLOW US</h3>

Twitter
Facebook
YouTube
Pinterest

</div>
<div class="footer-column">
<h3>OTHER</h3>

Video Tutorial
Vlogs & Ideas

</div>
</div>
</footer>
</html>
main* 0 0 0
Ln 29, Col 48 Spaces: 4 UTF-8 CRLF HTML

Appendix E: Curriculum Vitae

KRISTINE SHYRA CARPIO
Blk 88 Lot 5 Phase 5 Upper Bicutan, Taguig City, 1633
(+63) 948 082-8438 • shycarp@gmail.com
https://www.linkedin.com/in/kristine-shyra-carpio-618a18214/

KEY SKILLS:

Customer Service • Technical Support • Data Entry • Email Management • Admin Tasks • Photo Editing • Content Writing • AI Tool/Websites

---

TOOLS USED:

Salesforce, Azure, Avaya, Twilio, Customer Service Tool, Canva, Photoshop, AI Websites

---

WORK EXPERIENCE:

Customer Service Representative **April 2021 – Present**

FIS Global Solutions Philippines Inc. – Remote

- Assisted customers through phone calls, providing prompt and friendly assistance for financial account inquiries, issue resolution, and complaint handling.
- Managed account maintenance tasks, including updating personal information, processing account changes, and addressing address/name changes or beneficiary updates.
- Educated customers about various financial products and services, explaining account types, interest rates, fees, policies, and financial documents.
- Processed financial transactions such as deposits, withdrawals, fund transfers, bill payments, and account closures.
- Identified and resolved customer issues and discrepancies, investigating account problems, initiating corrections, and ensuring satisfactory resolution.
- Ensured compliance with security, privacy protocols, and applicable laws and regulations, safeguarding customer information and preventing fraudulent activities.

Administrative Associate **March 2019 – May 2019**

Bureau of Fire Protection – Alabang (On-site)

- Managing correspondence: ensuring timely responses or forwarding to the appropriate individuals.
- Document preparation and organization: Drafting, editing, and proofreading various documents, reports, and presentations. Maintaining electronic and physical filing systems to ensure easy retrieval of information.
- Data entry and record-keeping: Updating and maintaining databases, spreadsheets, and other records. Ensuring accuracy and confidentiality of data
- Office supply management: Ordering and managing office supplies, equipment, and inventory. Ensuring adequate stock levels and coordinating with vendors or suppliers.
- Communication and collaboration: Acting as a liaison between various departments, teams, and external parties. Facilitating effective communication and collaboration within the organization.
- Administrative support: Providing general administrative support to the team or department as needed. Assisting with miscellaneous tasks, such as filing expenses, processing invoices, or preparing reports.

Technical Support Representative **January 2020 – March 2020**

Digibytes Internet Café – SM Bicutan (On-site)

- Troubleshooting and Issue Resolution: Diagnosed and resolved technical issues with hardware, software, and IT systems for customers. Provided step-by-step instructions and utilized troubleshooting techniques to find solutions.
- Customer Assistance and Support: Interacted with customers through various channels to understand their technical concerns and deliver exceptional customer service. Addressed inquiries, resolved problems, and ensured customer satisfaction.
- Software and Hardware Installation: Assisted customers with the installation, configuration, and setup of software applications, operating systems, drivers, and hardware components. Provided guidance and troubleshooting support during installations.
- Remote Support: Utilized remote desktop tools to connect to customers' systems and provided real-time assistance. Diagnosed issues remotely, made configuration changes, and resolved technical problems without on-site visits.
- Training and Knowledge Sharing: Conducted training sessions and created instructional materials to educate customers on software applications, hardware devices, and IT systems. Shared knowledge and best practices to enhance technical support capabilities.
- System Maintenance and Updates: Assisted in routine maintenance tasks, system upgrades, and software patches to ensure optimal performance and security. Conducted periodic checks, performed software updates, and implemented security measures.
- Continuous Learning: Stayed up-to-date with emerging technologies, software applications, and IT trends. Actively pursued self-study, training programs, and certifications to continuously improve technical knowledge and skills.

EDUCATION:

- **Bachelor of Science in Computer Engineering**
  Technological Institute of the Philippines
- **STEM – Senior High School**
  West Bay College – Alabang

CERTIFICATIONS:

- **CCNAv7: Introduction to Networks** **May 2023**
  Cisco Networking Academy
- **Cybersecurity Essentials** **May 2023**
  Cisco Networking Academy

Gabriel Dela Cruz

Computer Engineering

A skilled and qualified computer engineering student with a comprehensive knowledge in computer programming, seeking a professional experience to utilize the knowledge I acquired during the course of my undergraduate education.

Address Karuhatan, Valenzuela City **Phone** 0945 634 1554

E-mail gabdrdc@gmail.com

mgdelacruz01@tip.edu.ph

Skills

- Programming skills
- Leadership skills
- Communication skills
- Management skills
- Goal-oriented
- Self-motivated
- Team player
- Detail-oriented

Education

2018-2020	University of the East - Caloocan Science, Technology, Engineering, and Mathematics Track
2020-2021	MOL Magsaysay Maritime Academy Bachelor of Science in Marine Transportation
2021-Present	Technological Institute of the Philippines Bachelor of Science in Computer Engineering

JOHN MIKAEL R. DIAZ

COMPUTER ENGINEERING

Technological Institute of the Philippines (TIP) M

Address: **015 Road 5 cor. Pacheco Drive, Dalandanan, Valenzuela City**

Email Address: **mjmdiaz@tip.edu.ph**

Cellular No.: **+639455421498**

CAREER OBJECTIVE

I am highly motivated and dedicated seeking opportunities to apply and expand technical skills, gain practical experience in software design, and troubleshooting, while contributing to innovative projects. Always aiming to enhance my knowledge, work collaboratively with a dynamic team, and ultimately pursue a successful career in the said field of technology.

DESIGN PROJECT COMPLETED/ RESEARCH

BuoyWatch: Practical FAD Management System and Automated Fish Detection Device for Fishermen
Project Design | May 2024

KNOWLEDGE, SKILLS, AND ATTITUDE

Having graduated from TIP with its orientation towards outcome-based education, I have acquired and can demonstrate the following student acquire outcomes (knowledge, skills, and attitudes) necessary to the practice of the computing profession:

- *Analyze complex problems and identify and define the computing requirements appropriate for solution.*
- *Use modern techniques and tools of the computing practice in complex activities.*
- *Understand professional, ethical, legal, security and social issues and responsibilities relevant to professional computing.*

LEADERSHIP ACTIVITIES

- D' Alpha Allegiance (Dance varsity)
  - Treasurer (2019 – 2020)
  - President (2020 – 2022)

SEMINARS AND TRAININGS ATTENDED

- CCNA R&S: Introduction to Networks; Routing and Switching Essentials; Scaling Networks; Enterprise Networking, Security, and Automation (Certificate of Completion)
  Technological Institute of the Philippines-Manila
  15 Oct 2018 – 22 Mar 2020

EXTRA AND CO-CURRICULAR ENGAGEMENT AND VOLUNTEER WORK

- Computer Engineering Student Society (CoESS)
  Member
  July 01, 2019 - July 31, 2020

OTHER SKILLS

- **Computer Troubleshooting**
- **Designing Database Structures and Tables**
- **Using Various Operating Systems** (Windows, Linux)
- **Programming Languages** (C, Assembly Language, Python)
- **Proficient in Microsoft Office** (Word, Excel, PowerPoint)

REFERENCES

Chona Opeña
Secretary, Computer Engineering
Technological Institute of the Philippines – Manila
openacp.rdmo@tip.ph
09397536379

ABDUL RAHMAN A. DIDA-AGUN
6 P Road 7, 1st West, Brgy. West Crame, San Juan City, 1500
(+63) 926-150-1527 • didaagunabdulrahman@gmail.com

KEY SKILLS:

Operating Systems • Programming • Data Entry • Email Management • Computer Networks • Project Management

---

TOOLS USED:

Kali Linux, DevC++, Cisco, PyCharm, IntelliJ

EDUCATION

- **Bachelor of Science in Computer Engineering**
  Technological Institute of the Philippines

- **STEM – Senior High School**
  Centro Escolar Integrated High School

- **Junior High School**
  Roosevelt College Cubao

CERTIFICATIONS

- **CCNAv7: Introduction to Networks Nov 2022**
  Cisco Networking Academy

- **Cybersecurity Essentials May 2023**
  Cisco Networking Academy

BEEJAY B. JAVIER

COMPUTER ENGINEER

PERSONAL PROFILE

I am an undergraduate B.S. Computer Engineering student with an unwavering passion for computing technology. I am a quick learner with the ability to handle pressure, as well as find creative solutions in solving problems; striving to improve and develop my skills and learn along the way.

SKILLS AND ABILITIES

- CISCO Networking
- Proficient in Microsoft Office (Word, Excel, PowerPoint)
- Programming Language: Python, C#, C++, Assembly Language
- Machine Learning: PyTorch, TensorFlow
- Proficient in MySQL
- Computer Repair and Troubleshooting

CONTACT ME AT

Acacia Drive B5 L4 PH1 Vista Bonita Subd. Brgy. San Jose Dasmariñas City, Cavite

bjjavier1@gmail.com

@bjjavier

(+63) 946 379 5637

WORK EXPERIENCE

Customer Service Associate
Sykes Asia, Inc. | Jul 2022 - Aug 2023

Data Encoder
Philippine Statistics Authority | Jan 2021 - Feb 2021

Statistical Researcher
Philippine Statistics Authority | Sep 2020 - Oct 2020

Statistical Researcher Team Supervisor
Philippine Statistics Authority | Sept 2020

Cadet Engineer
Experts Academy · OJT | May 2020 - Aug 2020

EDUCATIONAL HISTORY

Technological Institute of the Philippines
BS Computer Engineering | Jun 2017 - Present

- Auditor, Computer Engineering Student Society (2019 - 2020)

CERTIFICATIONS

- Splunk 7.x Fundamentals Part 1 (eLearning) (Jul 2020)
- FORTINET: NSE 2 Network Security Associate (Jun 2020 - Jun 2022)
- FORTINET: NSE 1 Network Security Associate (Jun 2020 - Jun 2022)
- Sophos Certified Technician (Jun 2020)
- Sophos Certified Engineer (Jun 2020)
- CCNAv7: Enterprise Networking, Security and Automation (Mar 2020)
- CCNA Routing and Switching: Scaling Networks (Dec 2019)
- CCNA Routing and Switching: Routing and Switching Essentials (Jun 2019)
- CCNA Routing and Switching: Introduction to Networks (Oct 2018)

EUGENE VILLEGAS

(+63) 976-489-9792

mevillegas@tip.edu.ph

6a Pascual Street, Greenheights Village, Parañaque City

CAREER OBJECTIVE

To use my current knowledge and skills in providing software solutions that addresses the problems that are concurrent in our community, and to further enhance this set of skills and knowledge using systematic approach towards any challenge that is to come.

CERTIFICATES

- **2024**
  Udemy

  HTML5 AND PYTHON3 COMPLETE COURSE 2023
  - Provided the essential coding techniques that help me with understand web development in a deeper sense.

- **2024**
  Udemy

  CSS, BOOTSTRAP ,JAVASCRIPT, PHP FULL STACK CRASH COURSE
  - Provides a basic functionalities of CSS, BOOTSTRAP, JAVA and PHP that is useful towards web development.

- **2024**
  Udemy

  C++ TRAINING CRASH COURSE 2022
  - A programming training using C++ that provides the basic fundamentals of coding.

EDUCATION

- **2022-PRESENT**
  Manila

  TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES
  - Bachelor of Science in Computer Engineering

- **2020-2021**
  Parañaque

  OLIVAREZ COLLEGE
  - Bachelor of Science in Radiologic Technology

SKILLS SUMMARY

Programming	92%
Project Management	47%
UI/UX Design	82%
Database Design	75%

OTHER SKILLS

- Data Analysis
- Project Presentation
- Project Documentation

REFERENCES

Abad, R. P. B., Banister, D., Biona, J. B. M. M., Fillone, A. M., Hickman, R. (2016, August 8). *Investigating the relationship between housing affordability and mobility in Metro Manila, Philippines*. Proceedings of the 23rd Annual Conference of the Transportation Science Society of the Philippines (2016). https://ncts.upd.edu.ph/tssp/wp-content/uploads/2016/08/Abad-et-al.pdf

Alcaraz, A. Z., Ison, J. Y., Jimenez, C. J., Marcelo, A. R., Alonsagay, A., Pelias, R., Dela Cruz, C., Alas, J. A., Santiago, A. V., Mesana, J. C. (2020, March). *ACCOMMODATION PREFERENCES AMONG A SELECT GROUP OF FILIPINO COLLEGE STUDENTS*. Antorcha, *7*(2). https://research-manila.letran.edu.ph/read/165

Apartments.com & Google. (2015). *Online Search Behavior and Trends of Apartment Renters*. https://www.shelterrealty.com/docs/Apartments_Google_WhitePaper.pdf

Ballesteros, M. M., Ramos, T. P., Ancheta, J. A. (2022, August 18). *Measuring Housing Affordability in the Philippines. Philippine Institute for Development Studies*. https://www.pids.gov.ph/publication/discussion-papers/measuring-housing-affordability-in-the-philippines

educations.com. (2021, January 5). *Study in the Philippines: Housing & Living Costs*. https://www.educations.com/study-guides/asia/study-in-the-philippines/student-housing-19879

Little, N. (2023, May 15). *8 ways to beat the cost of living as a student*. Barossa Regional University Campus. https://www.barossacampus.com.au/post/beat-the-cost-of-living-as-a-student

McKee, A. J. (2023). *Renting Housing in the Philippines as an Expat (2023)*. docmckee. https://docmckee.com/travel/renting-housing-in-the-philippines-as-an-expat-2023/

mommybloggersphilippines. (n.d.). *What Students Need to Consider When Choosing a Boarding House for Rent*. https://mommybloggersphilippines.com/2022/06/28/students-need-consider-choosing

-boarding-house-rent/

Morris, A. (2014, October 21). *Apartment complexes too often take advantage of college students*. The Crimson White. https://thecrimsonwhite.com/21573/opinion/apartment-complexes-too-often-take-advantage-of-college-students/

nyhabitat.com. (2013, January 28). *10 Things to Avoid When Renting an Apartment Online*. https://www.nyhabitat.com/blog/2013/01/28/10-things-avoid-renting-apartment-online/

Office of Outreach and Relationships. (2024, January 31). *Moving Off Campus in the Philippines: Housing and Financial Considerations*. The Continents State University. https://www.continents.us/blog/moving-off-campus-housing-and-financial-considerations-philippines

Oh, J. & Kim, J. (2021, March 3). *Relationship between Mental Health and House Sharing: Evidence from Seoul*. International Journal of Environmental Research and Public Health, *18*(5). https://doi.org/10.3390/ijerph18052495

OpenAI. (2024). *ChatGPT* (May 5 version) [Large language model]. https://chatgpt.com/

Velasquez-Garcia, Z. & Garcia, J. A. S. (2016). *On-Campus Living Experiences among Filipino University Students*. Philippine Journal of Counseling Psychology, *18*(1). https://ejournals.ph/article.php?id=17291

Wise. (2023). *Cost of living in the Philippines: Your guide*. https://wise.com/us/blog/cost-of-living-in-the-philippines

Wohletz, J. (2013, August 5). *Five reasons why renting an apartment is a huge pain*. Westworld. https://www.westword.com/arts/five-reasons-why-renting-an-apartment-is-a-huge-pain-5799227