**POKHARA UNIVERSITY**

**CRIMSON COLLEGE OF TECHNOLOGY**

**Devinagar-11, Butwal**



## A Final Year Project Report On

**GharKhoje- An Online Platform for finding properties**

(PRJ– 451)

*A project report submitted on behalf of partial fulfillment of Bachelo Of Computer Application*

*Pokhara University, Nepal*

# Submitted To:

# Crimson College of Technology

Department of Science and Technology

**Submitted By:**

Kamal Subedi (Symbol No: 27680/077) Gaurav Acharya (Symbol No: 27676/077)

## September, 2024

i

# SUPERVISOR’S RECOMMENDATION

I hereby recommend that this report has been prepared under my supervision by Kamal Subedi (27680/077) and Gaurav Acharya (27676/077) entitled “GharKhoje” in partial fulfillment of the requirements for the degree of BCA(Bachelor of Computer Application) be processed for evaluation.

……...................................

### Mr. Abdul Hak

Project Supervisor

Department of Science and Technology

Crimson College of Technology

# ACKNOWLEDGEMENT

We express our sincere gratitude to Crimson College of Technology, Butwal for their invaluable resources and consistent support throughout our project. The college's commitment to academic excellence has significantly influenced our educational journey, and we appreciate the tools and guidance provided.

A special acknowledgement goes to **Mr. Abdul Hak** for his exceptional guidance and support throughout the project. His knowledge and prompt assistance have played a significant role in the success of our work. We extend our thanks to all other supervisors for their contributions to the project.

We are also thankful to our principal, Mr. Devi Prasad Bhandari for providing us with a work space where we could easily work in a group. We thank respected teachers for supporting us in understanding and implementing our algorithms.

We would also like to acknowledge the support from our peers and colleagues who provided valuable input and participated in the testing phases of the project. Their feedback and suggestions have been essential in refining and improving GharKhoje.

We would highly appreciate and heartily welcome suggestion for further improvement if any.

Kamal Subedi (Symbol No: 27680/077) Gaurav Acharya (Symbol No: 27676/077)

# ABSTRACT

GharKhoje is an easy-to-use online property listing portal designed for people looking to buy, sell, or rent properties in Nepal. It is a simple platform where users can browse through a variety of properties and contact sellers directly through a secure messaging system. The website allows users to obtain detailed property information, including location, price, and area, making the search process effortless. Whether you want to list a property or find one that suits your needs, GharKhoje provides a one-stop solution for all property-related transactions. Our platform simplifies property dealings by allowing sellers to easily post their listings while giving Seekers transparent information about the properties.

***Keywords:*** *gharkhoje, owner, Seekers, homesearch, propertydetails*

# TABLE OF CONTENT

### [SUPERVISOR’S RECOMMENDATION...................................................................](#_bookmark0)...

[**ACKNOWLEDGEMENT.............................................................................................**](#_bookmark1)**..**

### [ABSTRACT...................................................................................................................](#_bookmark2)...

[TABLE OF CONTENT v](#_bookmark3)

[LIST OF FIGURES.......................................................................................................](#_bookmark4)..

[LIST OF TABLES.........................................................................................................](#_bookmark5).

[LIST OF ABBREVIATIONS.......................................................................................](#_bookmark6)...

[CHAPTER 1: INTRODUCTION................................................................................](#_bookmark7)....

* 1. [Introduction 1](#_bookmark8)
  2. [Problem Statement 2](#_bookmark9)

[1.3. Objectives 3](#_bookmark10)

* 1. [Scope and Limitations 4](#_bookmark11)
     1. [Scope 4](#_bookmark12)
     2. [Limitations 5](#_bookmark13)

[1.5. Methodology 6](#_bookmark14)

[1.6: Report Organization 7](#_bookmark16)

[CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW..............](#_bookmark17)....

[2.1 Background Study 8](#_bookmark18)

[2.1 Literature Review 9](#_bookmark19)

[2.2.1 Study of existing systems 9](#_bookmark20)

[CHAPTER 3: SYSTEM ANALYSIS...........................................................................](#_bookmark21)..

* 1. [System Analysis 11](#_bookmark22)
  2. [Requirement Analysis 11](#_bookmark23)
     1. [Functional Requirements 11](#_bookmark24)
     2. [Non-Functional Requirements 17](#_bookmark33)
     3. [Technical Feasibility 17](#_bookmark34)
     4. [Economical Feasibility 17](#_bookmark35)
     5. [Legal Feasibility 18](#_bookmark36)
     6. [Operational Feasibility 18](#_bookmark37)
     7. [Schedule Feasibility 18](#_bookmark38)
  3. [Analysis 19](#_bookmark40)
     1. [Class Diagram 19](#_bookmark41)
     2. [Activity Diagram 20](#_bookmark43)

[CHAPTER 4: SYSTEM DESIGN................................................................................](#_bookmark47)..

* 1. [Design 23](#_bookmark48)
     1. [System Architecture Diagram 23](#_bookmark49)
     2. [Component Diagram 24](#_bookmark51)
  2. [Algorithm Details 25](#_bookmark53)

[CHAPTER 5: IMPLEMENTATION AND TESTING..............................................](#_bookmark54)..

* 1. [Implementing Tools 27](#_bookmark55)
  2. [: Testing 28](#_bookmark56)
     1. [: Unit Testing 28](#_bookmark57)
     2. [Test Cases for System Testing 32](#_bookmark61)
  3. [Result Analysis 33](#_bookmark65)

[Chapter 6: Conclusion and Future Recommendations..............................................](#_bookmark66)..

[6.1: Conclusions.............................................................................................................](#_bookmark67)..

[6.2: Future Recommendations 34](#_bookmark68)

[**References.......................................................................................................................**](#_bookmark69)**..**

[**Appendix.........................................................................................................................**](#_bookmark70)**..**

# LIST OF FIGURES

[Figure 1. 1: Agile Methodology 6](#_bookmark15)

[Figure 3. 1: Use Case Diagram 12](#_bookmark25)

[Figure 3. 2: Gantt chart of work schedule 18](#_bookmark39)

[Figure 3. 3: Class Diagram for GharKhoje 19](#_bookmark42)

[Figure 3. 4: Activity diagram of admin 20](#_bookmark44)

[Figure 3. 5: Activity diagram of Property Seller 21](#_bookmark45)

[Figure 3. 6: Activity Diagram of Property Seeker 22](#_bookmark46)

[Figure 4. 1: Repository System Architecture Design 23](#_bookmark50)

[Figure 4. 2: Component Diagram 24](#_bookmark52)

# LIST OF TABLES

[Table 3. 1: Register Use Case Description 13](#_bookmark26)

[Table 3. 2: Use Case Description of Admin 14](#_bookmark27)

[Table 3. 3: Use Case Description of Admin (Manage Properties) 14](#_bookmark28)

[Table 3. 4: Use Case Description of Owner (Login) 15](#_bookmark29)

[Table 3. 5: Use Case Description of Owner (Add Property) 15](#_bookmark30)

[Table 3. 6: Use Case Description of Seeker (Login) 16](#_bookmark31)

[Table 3. 7: Use Case Description of Seeker 16](#_bookmark32)

[Table 5. 1: Test case for User Login… 29](#_bookmark58)

[Table 5. 2: Test case for Admin Login 30](#_bookmark59)

[Table 5. 3: Test case for User Registration 31](#_bookmark60)

[Table 5. 4: Test Case for System Testing 32](#_bookmark62)

[Table 5. 5: Test Case for System Usability Testing 32](#_bookmark63)

[Table 5. 6: Test Case for Admin Functionality 33](#_bookmark64)

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| API | Application Programming Interface |
| DB | Database |
| JS | JavaScript |
| SDLC | Software Development Life Cycle |
| UI | User Interface |
| JSON | JavaScript Object Notation |
| JWT | JSON Web Token |
| TF-IDF | Term Frequency-Inverse Document Frequency |

# CHAPTER 1: INTRODUCTION

## Introduction

The real estate industry in Nepal has been growing rapidly in recent years, with an increasing demand for property buying, selling, and renting. However, the traditional methods of finding and listing properties are often time-consuming, inefficient, and limited in reach. This has created a need for a modern, user-friendly platform that can connect property Seekers, sellers, and effectively.

GharKhoje is an innovative online real estate portal designed to revolutionize the way people in Nepal search for, list, and manage properties. By leveraging the power of ReactJS, a modern JavaScript library for building user interfaces, GharKhoje aims to provide a seamless, responsive, and feature rich experience for users across various devices.

The platform will serve as a comprehensive marketplace for real estate transactions, offering features such as advanced property search, virtual tours, neighborhood insights, and direct communication between interested parties. For property owners and real estate agents, GharKhoje will provide tools to easily list and manage properties, track inquiries, and analyze market trends.

By bridging the gap between property seekers and providers, GharKhoje aspires to streamline the real estate process in Nepal, making it more accessible, transparent, and efficient for all stakeholders involved.

## Problem Statement

The real estate market in Nepal faces several challenges that hinder efficient transactions and limit market growth:

1. **Fragmented Information:** There is a lack of a centralized, reliable platform for property listings, making it difficult for Seekers and renters to find comprehensive information about available properties.
2. **Limited Reach:** Traditional methods of property advertising have limited reach, sellers and landlords from accessing a wider pool of potential Seekers or tenants.
3. **Inefficient Communication:** The process of arranging property viewings and negotiations is often time-consuming and inefficient, with multiple back-and-forth communications.
4. **Lack of Transparency:** There is often a lack of detailed information about properties, neighborhoods, and market trends, leading to uncertainty in decision- making.
5. **Outdated Listing Management:** Many real estate agents and property owners struggle with managing and updating their listings effectively, leading to inaccurate or outdated information.
6. **Limited Market Insights:** Both Seekers and sellers lack access to comprehensive market data and trends, making it challenging to make informed decisions.
7. **Accessibility Issues:** Traditional real estate processes can be particularly challenging for people with mobility issues or those living far from the property they are interested in. GharKhoje aims to address these problems by providing a modern, user-friendly platform that centralizes property information, facilitates efficient communication, and offers valuable insights to all users involved in real estate transactions in Nepal.

## 1.3. Objectives

The primary goal of GharKhoje is to create a comprehensive and user-friendly online real estate portal that caters to the needs of property Seekers, sellers, renters, and real estate professionals in Nepal. The specific objective of the project is:

1. Create a centralized platform for property listings with advanced search and filtering options based on location, price, property type, and amenities, enhancing user experience
2. Implement user authentication, role-based access control, and a secure messaging system to facilitate communication between property seekers, sellers, and agents while ensuring privacy
3. Develop a user-friendly interface for property owners and agents to efficiently manage and update property listings, supported by an admin panel for monitoring activities and generating reports.
4. Prioritize user data security by following best practices in web application security and privacy protection.

By achieving these objectives, GharKhoje aims to significantly improve the real estate transaction process in Nepal, making it more efficient, transparent, and accessible for all stakeholders involve.

## Scopes and Limitations

### Scopes

The scope of the GharKhoje project includes:

1. User Roles and Authentication:
   * Implement secure user registration and login functionality.
   * Create different user roles (e.g., property seeker, owner, agent, admin) with appropriate access rights.
2. Property Listing and Management:
   * Allow property owners and agents to create, edit, and manage property listings.
   * Implement a comprehensive property search feature with multiple filters.
   * Enable users to save favorite properties and set up alerts for new listings.
3. Communication System:
   * Develop a secure in-app messaging system for users to communicate about properties.
4. Map Integration:
   * Integrate interactive maps to display property locations.
   * Provide neighborhood information and nearby amenities through maps.
5. Admin Panel:
   * Create a comprehensive admin dashboard for managing users, listings, and site content.
6. Mobile Responsiveness:
   * Ensure the application is fully responsive and functions well on various devices and screen sizes.

### Limitations

The following limitations apply to the initial version of GharKhoje:

1. Geographic Coverage: The system will initially focus on properties within Nepal. International listings will not be supported in this version.
2. Language Support: The platform will primarily support English languages. Other languages will not be available in the initial release.
3. Offline Functionality: The application will require an internet connection to function. Offline mode will not be supported in this version.
4. Legal and Financial Services: The platform will not provide legal or financial services related to property transactions. Users will need to seek these services externally.
5. Real-time Data: While the system will strive to provide up-to-date information, real-time updates on property availability may not always be possible due to reliance on user inputs.
6. Verification Process: While basic verification measures will be in place, the platform cannot guarantee the accuracy of all user-provided information about properties or users.
7. Mobile Application: The initial version will be a web application optimized for mobile browsers. Native mobile apps for iOS and Android will not be developed in this phase.
8. Integration Limitations: The system may have limited integration with external real estate tools or services in its initial version.

These limitations provide a realistic scope for the initial version of GharKhoje, allowing for future enhancements and expansions based on user feedback and market demands.

## 1.5. Methodology

The development of the "GharKhoje" project we have selected the Agile methodology, it allowing for flexibility and continuous improvement throughout the development process. By working in iterative sprints, we were able to break down the project into manageable tasks, regularly delivering functional components such as property listings, secure messaging, and search features. Frequent user feedback was integrated into each iteration, enabling us to quickly adapt and refine the platform. This approach ensured a collaborative environment, reduced development risks, and resulted in a more user-centric, efficient, and reliable system for property transactions.

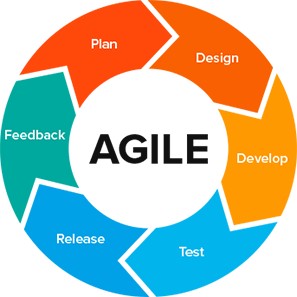


Figure 1. 1: Agile Methodology

## 1.6: Report Organization

This report is divided into six chapters:

**Chapter 1:** This chapter explains the reasons behind the development of the GharKhoje platform, including the problem it addresses, the project's objectives, its features, and the scope and limitations of the project.

**Chapter 2:** In this chapter, we review existing research and literature relevant to the project. We analyze journals, articles, and previous works to provide background and context for our project.

**Chapter 3:** This chapter focuses on the system analysis, including how we identified user requirements and assessed the feasibility of the platform.

**Chapter 4:** In this section, we cover the system design with detailed diagrams such as class diagrams, object diagrams, and activity diagrams, showing the structure of the platform.

**Chapter 5:** This chapter describes how the system was implemented and tested. We go over the tools and technologies used, along with the testing procedures to ensure functionality.

**Chapter 6:** Finally, we present the achievements of the project and provide recommendations for future improvements and potential developments.

# CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

## 2.1 Background Study

The real estate market in Nepal has undergone significant transformation with the adoption of online platforms for property transactions. These platforms have brought a new level of accessibility, transparency, and convenience to the property market, allowing users to engage in buying, selling, or renting properties with ease.

Studies show that online property platforms offer several advantages, such as cost- effectiveness, time efficiency, and extensive search capabilities. Users appreciate the transparency and ease of use provided by these platforms, where detailed property information is readily available, including prices, locations, and images. The ability to communicate directly with property owners has further enhanced the user experience by eliminating intermediaries and reducing transaction complexity.

Another crucial factor contributing to these platforms' success is their focus on security and reliability. Users from various backgrounds have expressed a growing trust in online platforms due to the secure handling of personal data and the reliability of the information provided. The availability of advanced services, such as legal assistance and property consultancy, adds to the credibility and professionalism of these platforms, making them a preferred choice for serious Seekers and sellers [1].

As a result, online property portals have become essential tools in the modern real estate market, offering a more efficient and transparent alternative to traditional property dealing methods. These platforms cater to a wide spectrum of users, from individuals looking for quick and informal property transactions to those seeking more formalized and professional services [2].

By streamlining the entire property transaction process, these platforms are helping to meet the demands of a growing, tech-savvy population, pushing the real estate market in Nepal towards further digitalization.

## 2.1 Literature Review

The real estate industry has undergone significant digital transformation in recent years. Real-estate portals have become increasingly popular, offering a convenient and efficient way for Seekers, sellers, and renters to connect and transact.

According to a study by the National Association of Realtors (2021), 97% of home Seekers used the internet in their home search process. This highlights the critical role that online platforms play in the modern real estate market [3].

In Nepal, the adoption of digital platforms for real estate has been growing, albeit at a slower pace compared to more developed markets. A report by the Nepal Land and Housing Association indicated that there is a significant opportunity for digital platforms to streamline the process in Nepal, particularly in urban areas where internet penetration is higher [4].

### 2.2.1 Study of Existing Systems

Some of the existing real estate platforms in Nepal include:

1. **Gharbazar:** Gharbazar.com provides services for those who want to buy, sell and rent their properties. It is an online marketplace for those who want to deal with properties in Nepal. It is a platform where the Seeker and the seller can deal with the properties directly with each other. It is simply a bridge between the Seekers and the sellers [5].
2. **Hamrobazar:** Hamrobazar.com is Nepal's No. 1 Marketplace which enables to list wide variety of new or used product online. While not exclusively a real estate platform, it has a significant section dedicated to property listings [6].
3. **NepalHomes:** NepalHomes is a pioneer prop-tech company that serves real estate businesses with a one-stop property marketplace, authentic real estate knowledge center, and an exclusive marketing platform [7].

After examining these systems, we identified several areas for improvement:

-Limited advanced search capabilities

* Lack of virtual tour features

– Absence of comprehensive market insights and trend analysis tools

* Limited integration of map-based search and neighborhood information GharKhoje aims to address these gaps while providing a more user-friendly and feature-rich experience.

## CHAPTER 3: SYSTEM ANALYSIS

## System Analysis

Here, the focus is on understanding and defining the system's requirements and feasibility. The section describes the functional requirements through the illustration of use case diagrams, portraying user-system interactions. Non-functional requirements, encompassing aspects like performance and security, are also explored. Additionally, the feasibility analysis examines the technical, operational, economic, and scheduling aspects of the project. The chapter employs an object-oriented approach, utilizing tools such as ER diagrams for data modeling, class diagrams for object modeling, and activity diagrams for process modeling, to comprehensively analyze and design the system.

## Requirement Analysis

The following are the key functional and non-functional requirements for GharKhoje:

### Functional Requirements

1. User Management:
   * User registration and authentication

* Role-based access control (property seeker, owner, admin)
* User profile management

1. Property Listing:
   * Create, edit, and delete property listings
   * Advanced search and filter functionality
2. Communication:
   * In-app messaging system
3. Map Integration:
   * Display property locations on interactive maps.
4. Admin Functions:
   * User management
   * Property listing moderation

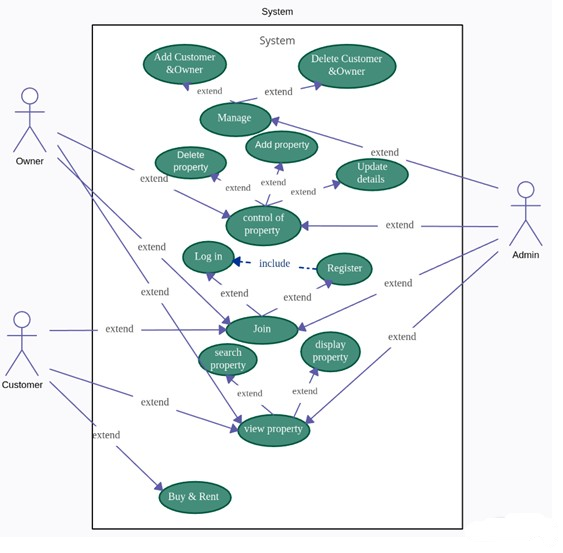


Figure 3. 1: Use Case Diagram

The figure above is the use case diagram of our proposed project, GharKhoje. There are three actors: owner, admin and Seeker (customer) along with their respective use cases.

### Use Case Description

Table 3. 1: Register Use Case Description

|  |  |  |
| --- | --- | --- |
| Use Case Name | | Register |
| Scenario | | Create user-account for Seeker |
| Triggering event | | User starts registration process |
| Brief description | | User should be able to create his/her new account in order to chat with owners. |
| Actors | | Seeker |
| Related use cases | | Login |
| Stakeholders | | Users (Seeker or Owner), Admin |
| Preconditions | | User should start the system |
| Preconditions | | New user account is created |
| Flow of Activities | 1. | The user accesses the Registration page. |
|  | 2. | The user enters their personal information |
|  |  | into the registration form. |
|  | 3. | After submitting the credentials, the system |
|  |  | creates a new account for the user. |
|  | 4. | The user is redirected to the login page |
| Exception conditions | | User provides incomplete or wrong information; the system must display suitable  error message to the user |

Table 3. 2: Use Case Description of Admin

|  |  |
| --- | --- |
| Use Case Name | Login |
| Actors | Admin |
| Preconditions | None |
| Primary Path | * Email address * Password * Click on Login |
| Exception conditions | Invalid Credentials |

Table 3. 3: Use Case Description of Admin (Manage Properties)

|  |  |
| --- | --- |
| Use Case Name | Login |
| Actors | Admin |
| Preconditions | Login |
| Primary Path | * Approve * Reject |
| Exception conditions |  |

Table 3. 4: Use Case Description of Owner (Login)

|  |  |
| --- | --- |
| Use Case Name | Login |
| Actors | Owner |
| Preconditions | None |
| Primary Path | * Email address * Password * Click on Login |
| Exception conditions | Fields are missing |

Table 3. 5: Use Case Description of Owner (Add Property)

|  |  |
| --- | --- |
| Use Case Name | Add property |
| Actors | Owner |
| Preconditions | Login |
| Primary Path | * Add property * Edit property |
| Exception conditions |  |

Table 3. 6: Use Case Description of Seeker (Login)

|  |  |
| --- | --- |
| Use Case Name | Login |
| Actors | Seeker |
| Preconditions | None |
| Primary Path | * Email address * Password * Click on Login |
| Exception conditions | Fields are missing |

Table 3. 7: Use Case Description of Seeker

|  |  |
| --- | --- |
| Use Case Name | Login |
| Actors | Seeker |
| Preconditions | Login |
| Primary Path | * View property * Message to owner |
| Exception conditions | User must be logged in to message owner |

### Non-Functional Requirements:

1. Security:
   * Secure user authentication (e.g., JWT (JSON Web Tokens), OAuth)
   * Data encryption for sensitive information
   * Protection against common web vulnerabilities (e.g., XSS, CSRF)
2. Usability:
   * Intuitive and responsive user interface
3. Compatibility:
   * Cross-browser compatibility (Chrome, Firefox, Safari, Edge)
   * Responsive design for mobile devices

## Feasibility Study

### Technical Feasibility

GharKhoje is technically feasible due to the following factors:

* ReactJS is a widely used, stable framework with extensive documentation and community support.
* The team has experience in web development and is familiar with JavaScript and React ecosystems.
* Required tools and technologies (e.g., Node.js, MongoDB) are readily available and well-documented.
* Integration of third-party services (e.g., mapping APIs) is possible through available APIs.

### Economical Feasibility

An economical feasibility study is an analysis of the costs and revenues of a project to determine whether it is logical and possible to complete it. It is a type of cost-benefit analysis that evaluates whether it is possible to implement the project. Our project is economically feasible as we are using freely available tools and software to build the project. Furthermore, after our project is successfully developed, collaborations with

recruiters and educational institutions further enhance revenue opportunities through strategic partnerships and targeted advertising.

### Legal Feasibility

Legal feasibility refers to the evaluation of whether a software project can be implemented successfully based on legal and ethical requirements. It involves analyzing the barriers to legal implementation of the project, data protection or social media laws, project certificates, licenses, copyrights, etc. The project that we are developing is legally feasible as it does not break any legal rules.

### Operational Feasibility

GharKhoje is operationally feasible because: - It addresses a clear need in the Nepali real estate market. - The user- friendly interface will encourage adoption by both tech- savvy and less technical users. - Regular updates and customer support will ensure smooth operation and user satisfaction.

### Schedule Feasibility

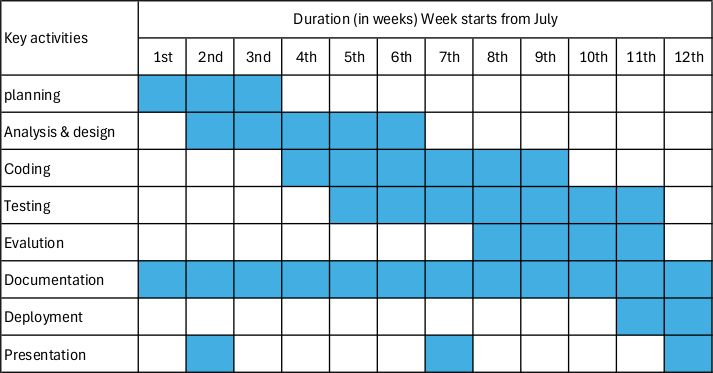
The project can be completed within the academic year, following this tentative schedule:

Figure 3. 2: Gantt chart of work schedule

## Analysis

### Class Diagram

A class diagram is a specific kind of UML (Unified Modelling Language) diagram that will be used to show how a system or software application is organized. It will provide a visual representation of a system's classes, interfaces, and objects, as well as their relationships, characteristics, methods, and activities.

For the GharKhoje project, a class diagram will be used to model the different entities involved in the system, such as users, orders, admin, etc.



Figure 3. 3: Class Diagram for GharKhoje

The diagram provides a clear overview of the system’s functionality, showing how different users (Admin, Seeker, and Seller) interact with the property listings. Admins

manage the system by adding or deleting properties and generating reports. Seekers can search for properties, view details. Sellers can list new properties, update existing listings, and manage their profiles.

### Activity Diagram

An activity diagram is a type of UML (Unified Modelling Language) diagram that will show the flow of actions and events in a system or process. It will be used to model the dynamic behaviour of a system, such as the interactions between actors, objects, and component.

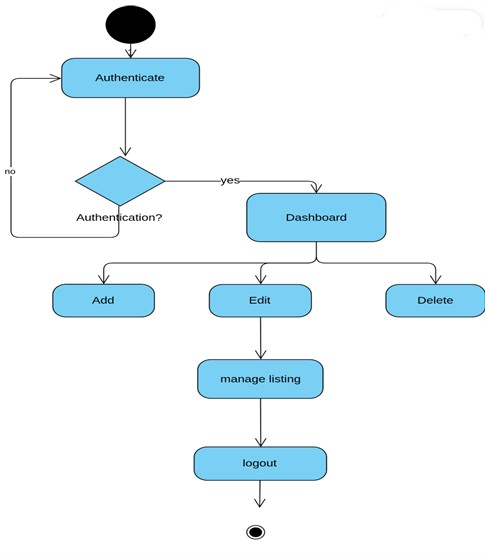


Figure 3. 4: Activity diagram of admin

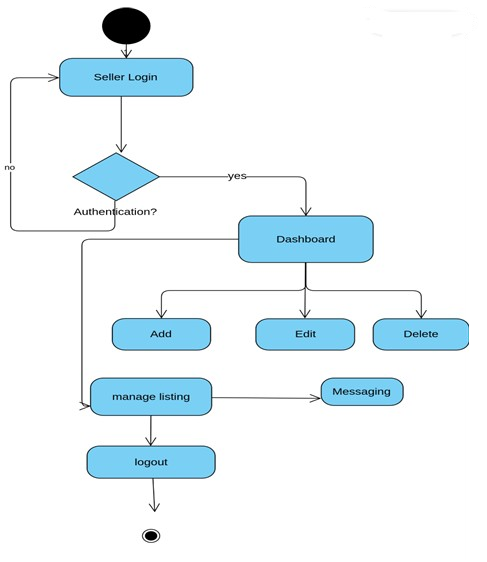


Figure 3. 5: Activity diagram of Property Seller

The above activity diagram for a property seller on a website outlines the seller’s journey from logging in to logging out. It includes actions like listing a new property, editing existing listings, deleting listings, and responding to Seeker inquiries. This visual representation helps understand the seller’s interactions and the system’s responses, ensuring a seamless and efficient experience for managing property listings.

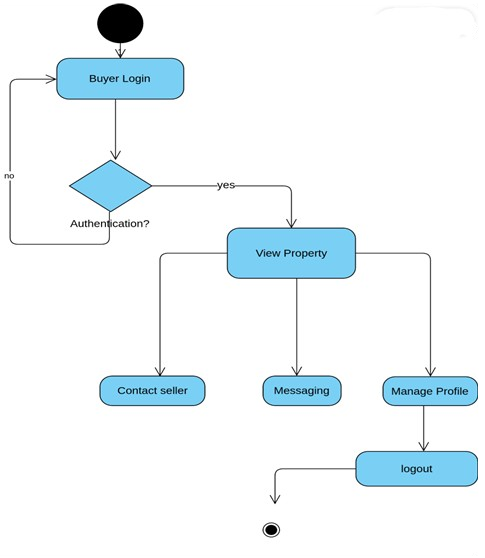


Figure 3. 6: Activity Diagram of Property Seeker

The above activity diagram for a property Seeker on outlines the Seeker’s journey from accessing the website to logging out. It includes actions like searching for properties, viewing property details, saving properties to a wish list, and contacting sellers. This visual representation helps understand the Seeker’s interactions and the system’s responses, ensuring a seamless and user-friendly experience.

# CHAPTER 4: SYSTEM DESIGN

## Design

### System Architecture Diagram

System architecture design refers to the conceptual model that defines the structure, behaviors, and more views of a system. It is a blueprint for the system and the development

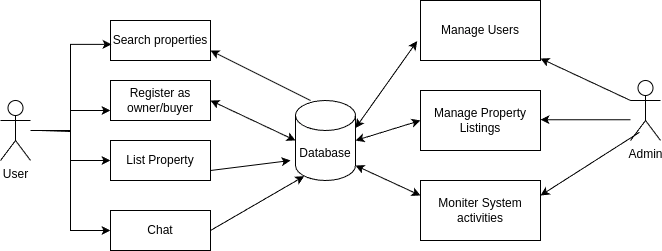


Figure 4. 1: Repository System Architecture Design

Figure above represents an interaction model between administrators, users, and the central database. Administrators have multiple functionalities: they can manage users, manage the property listings, and monitor the system activities. User begin with the registration process, after which they have the ability to search for Property, list Property for sell, and chat with owners.

### Component Diagram

A component diagram is a type of Unified Modelling Language (UML) diagram that depicts the high-level structure of a system, showcasing the components that make up the system and their relationships.

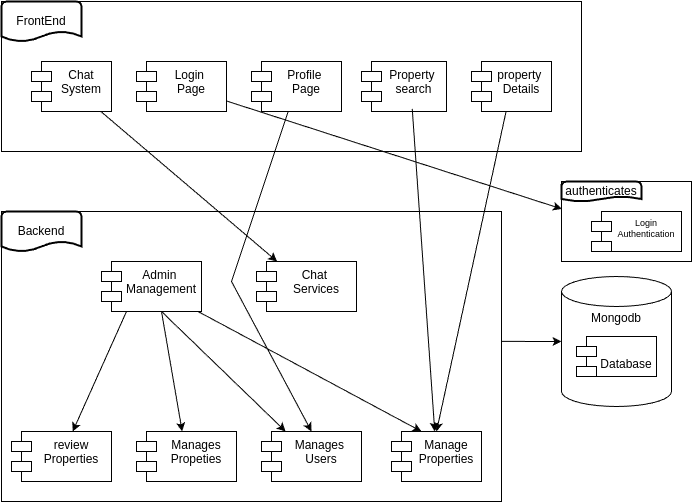


Figure 4. 2: Component Diagram

The component diagram outlines the architecture of a website system with interactions between various components. It shows that both Admin and User entities access the system through the Website. The main interface for both is the Homepage, which serves as a central hub, linking users to various functionalities such as Search, Register, Login, and Chat. Once a user logs in, they can List Property. The Admin Panel is accessible only after logging in and allows administrators to manage the website's content and operations. The diagram highlights how different components like the homepage, login system, and functionalities for users are interconnected, providing a clear flow of actions within the website system.

## Algorithm Details

We are going to use recommendation system to recommend property to seeker. For that we are going to use content-based recommendation system.

### Content-Based Algorithm in GharKhoje

Content-based filtering is an information retrieval method that uses item features to select and return items relevant to a user’s query. Content-based filtering in recommender systems uses the details of items (like location, property type and description) to match them with user preferences. It builds a temporary user profile based on the items the user has liked or is currently viewing, without needing them to sign in. The system then recommends similar items by comparing new items with those the user has shown interest in. This method is especially useful when we have detailed information about the items but little to no data about the user.

The step to implement content based recommendation system are:

### Collecting the data of user

* 1. The data of the user is collected through various interaction of the user like
  2. User interest and property preference
  3. Analyzing the user’s past searches and filters (e.g., price range, number of bedrooms)
  4. Tracking user interactions with the platform, including properties viewed, time spent on listings

All those data are collected and stored in a database. This is called user profile.

### 2 .Normalize the data

In this step, all the data collected are normalized in a particular range so that it will be suitable for processing. Data that are not necessary are removed.

### Vectorization of data for comparison

The normalize data is not suitable for comparing similarity. It is converted into numbers of each features. For vectorization we can use TF\*IDF (Term Frequency) \* (Inverse Document Frequency). For that first data is tokenized into words or tokens. It is arranged in a table having features and document and TF\*IDF is calculated.

*Term Frequency* (*TF* )=No. of repetition of wordsin sentence

*No. of wordsin sentence*

*Inverse Doc Frequency* (*IDF* )=log )

This both are vectors when done scalar product, produces TF\*IDF vector.

### Compute Cosine Similarity

In this step, we calculate angles between each vector above to all other vector. Cosine angle gives the angle between two vector that gives the insight on how much they are similar to each other.

Its formula is

*Cosine Angle*= A⃗ ∗ B⃗

A⃗ ∨ B⃗ ∨

After calculating the cosine angle of all the data above, we have a similarity vector for each vector.

### Recommend new posts

Once we have similarity matrix, we can calculate it to find other similar property that user will like. For that we first query property title, property description, property location. And then we find similarity of these data to using cosine and then compare it with above similarity matrix. The top most similar items are recommended to the user.

# CHAPTER 5: IMPLEMENTATION AND TESTING

## Implementing Tools

### Frontend:

* + 1. ReactJS: A JavaScript library for building user interfaces, providing a component-based architecture for efficient development.
    2. Redux: For state management across the application.
    3. React Router: For handling routing in the single-page application.

### Backend:

1. Node.js: A JavaScript runtime for building scalable network applications.
2. Express.js: A minimal and flexible Node.js web application framework.
3. MongoDB: A NoSQL database for flexible and scalable data storage.

**Authentication:** - JSON Web Tokens (JWT): For secure authentication and authorization.

### Testing:

1. React Testing Library: For testing React components

### Development and Deployment:

1. Configuration Management Tool GitHub: GitHub is a popular version control system used in software configuration management. It provides a hosting platform for Git repositories, allowing us to store and manage the source code. It keeps track of changes made to the codebase over time, allowing us to view the entire version history of a project.
2. Drawing Tool Draw.io: Draw.io, now known as diagrams.net, is a free and versatile online diagramming tool that allows users to create a wide range of diagrams, flowcharts, and other visual representations. It supports collaborative editing, integration with various cloud platforms, and provides an intuitive interface for creating and customizing diagrams. We used it for drawing use cases, activity, and class diagrams.

### Development Tools:

1. Visual Studio Code: As the primary code editor.
2. MongoDB Compass: For database management and visualization

## : Testing

Testing was a crucial part of our project to ensure that the platform functions as intended and provides a smooth user experience. We implemented various testing methods to catch and fix issues at different stages of development.

### : Unit Testing

We performed unit tests on individual components and functions to verify that each part of the code works correctly in isolation. This helped us catch bugs early and made debugging easier.

### Test Case 1: User Login

Table 5. 1: Test case for User Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SN | Test Inputs | Expected Outcomes | Actual Output | Results |
| 1 | When user provides correct email and password.  Email: [knowidont344@gmail.c](mailto:knowidont344@gmail.c) om  Password: abcd123456 | Login should be success | Login successful and redirected to home page | Test successful |
| 2 | When no email and password is provided: Email:  Password: | Invalid Credentials | Invalid Credentials | Test successful |
| 3 | When username is provided with incorrect password.  Email: [knowidont344@gmail.c](mailto:knowidont344@gmail.c) om  Password: abcd1234567 | Invalid Credentials | Invalid Credentials | Test successful |

### Test Case 2: Admin Login

Table 5. 2: Test case for Admin Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Test Inputs | Expected Outcomes | Actual Output | Results |
| 1 | When the user provides correct email and password.  Email: [knowidont344@gmail.c](mailto:knowidont344@gmail.c) om  Password: abcd123456 | Login success | Login successful and redirected to Admin dashboard | Test successful |
| 2 | When no email and password is provided:  Email: Password: | Invalid Credentials | Invalid Credentials | Test successful |
| 3 | When username is provided with incorrect password. | Invalid Credentials | Invalid Credentials | Test successful |

### Test Case 3: User Registration

Table 5. 3: Test case for User Registration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Test Inputs | Expected Outcomes | Actual Output | Results |
| 1 | Name: Kamal Subedi Email: [knowidont344@gmail.c](mailto:knowidont344@gmail.c) om  Password: abcd123456 Confirm Password:  abcd123456 | Successfully Registered. | Success (redirect to email verification page) | Test successful |
| 2 | Email: [knowidont344@gmail.c](mailto:knowidont344@gmail.c) om  Password: abcd123456 Confirm Password:  abcd123456a | Password do not match | Password do not match | Test successful |
| 3 | Email: Password:  Confirm Password: | Required Field | Email is invalid and Password must be at least 6 characters | Test successful |
| 4 | If already existing email is entered while  registration. | Error Email exists. | Email already exists | Test successful |

## Test Cases for System Testing

System testing is defined as testing of a complete and fully integrated software product. System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. system testing falls within the scope of black box testing, and as such should require no knowledge of inner design of the code or logic. One of the types of system testing is the usability testing which is performed in the system.

Table 5. 4: Test Case for System Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test  ID | Test Inputs | Expected Outcomes | Actual Output | Result |
| 1 | User login successful, uploads, views, chats. | Owner uploads the property details and Seeker views, and  chat to owner |  | Test Successful |

Table 5. 5: Test Case for System Usability Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test  ID | Test Inputs | Expected Outcomes | Actual Output | Result |
| 1 | Click on various links | Link should take | Users redirected | Test |
|  | on the system. | users to another | to expected | Successful |
|  |  | webpage according | webpages after |  |
|  |  | to the on-page | clicking the |  |
|  |  | URL. | links. |  |

Table 5. 6: Test Case for Admin Functionality

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test  ID | Test Inputs | Expected Outcomes | Actual Output | Result |
| 1 | Admin Login, Approve and Reject Listings. | Successful Completion of Admin Actions | Success | Test Successful |

## Result Analysis

The test results for the GharKhoje platform indicate that our system is robust in managing both admin and user interactions. The login functionalities are secure and user-friendly, with appropriate messages displayed for various scenarios such as invalid inputs or incomplete forms. The profile creation process is safeguarded against common issues, ensuring a smooth user experience.

All test cases marked as "Test Successful" demonstrate that the system behaves as expected according to the defined requirements. The system has passed tests for various scenarios, covering a wide range of user inputs and interactions, and ensuring that it reacts appropriately. This includes functionalities for Seekers and property owners and administrative tasks.

Overall, the test results suggest that the GharKhoje website is operational, user- centric, and ready for deployment. The system is prepared to offer a seamless user experience and effectively serve its intended purpose in the market

# Chapter 6: Conclusion and Future Recommendations

# : Conclusions

The development of the "GharKhoje" platform has effectively met the need for a more dynamic and user-friendly property listing experience. By incorporating features such as property viewing, listing capabilities, and secure messaging between Seekers and sellers, the portal provides a comprehensive solution for real estate transactions.

The Agile methodology employed allowed us to continuously refine and enhance the platform based on user feedback and iterative testing. After rigorous evaluation, we have confirmed that the system is both reliable and efficient, positioning "GharKhoje" as a valuable tool for the real estate market. The successful implementation of this platform is set to improve property searching and listing processes, offering an accessible and streamlined experience for all users.

## : Future Recommendations

* + - **Expand Features:** Consider adding advanced functionalities such as virtual property tours, augmented reality for property visualization, and integration with third-party services for mortgage and legal assistance to provide a richer user experience.
    - **Collect Feedback:** Continuously gather and analyze user feedback to identify areas for improvement and implement updates that align with user needs and preferences.
    - **Enhance Scalability:** As the platform's user base grows, focus on scaling the infrastructure to efficiently handle increased traffic and data while maintaining high performance and reliability.
    - **Develop a Mobile App:** Create a mobile application to offer users a convenient way to browse properties, list new ones, and communicate with sellers directly from their smartphones, increasing accessibility and engagement.

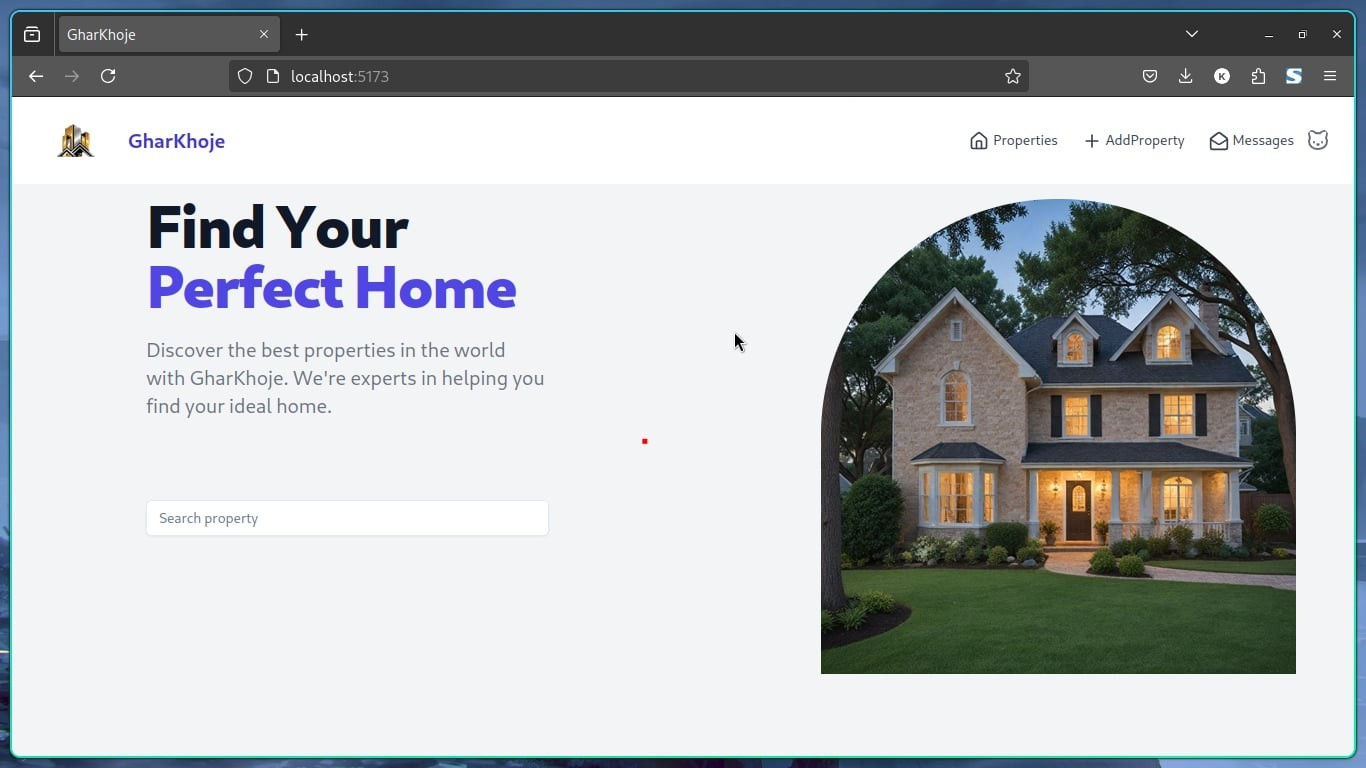
# References

1. "Fyafullaa | Ghar Jagga Nepal – Property Listings," Fyafullaa, <https://www.fyafullaa.com/>. [Accessed Sept 2024].
2. "GP Real Estate | Find Your Dream Property in Nepal," GP Real Estate, <https://gprealestate.com.np/>. [Accessed Sept 2024].
3. National Association of Realtors. (2021). Real Estate in a Digital Age 2021 Report. [National Association of REALTORS® (nar.realtor)](https://www.nar.realtor/). [Accessed Sept 2024].
4. Nepal Land and Housing Developers’ Association. (2022). Digital Trans- formation in Nepal’s Real Estate Sector. [Home - Nepal Land & Housing Developers](https://nlhdf.org.np/) [Federation (nlhdf.org.np)](https://nlhdf.org.np/). [Accessed Sept 2024].
5. "NepalHomes | Property in Nepal | Buy/Sell/Rent," NepalHomes, <https://www.nepalhomes.com/>[Accessed Sept 2024].
6. "HamroBazar: Online Marketplace and Property Listings in Nepal," HamroBazar, <https://www.hamrobazar.com/>[Accessed Sept 2024].
7. Nepal Homes: Home Page <https://www.nepalhomes.com/> . [Accessed Sept 2024].
8. How Recommendation Algorithms Work—And Why They May Miss the Mark. [https://www.scientificamerican.com/article/how-recommendation-algorithms-work-](https://www.scientificamerican.com/article/how-recommendation-algorithms-work-and-why-they-may-miss-the-mark/) [and-why-they-may-miss-the-mark/](https://www.scientificamerican.com/article/how-recommendation-algorithms-work-and-why-they-may-miss-the-mark/). [Accessed Sept 2024].

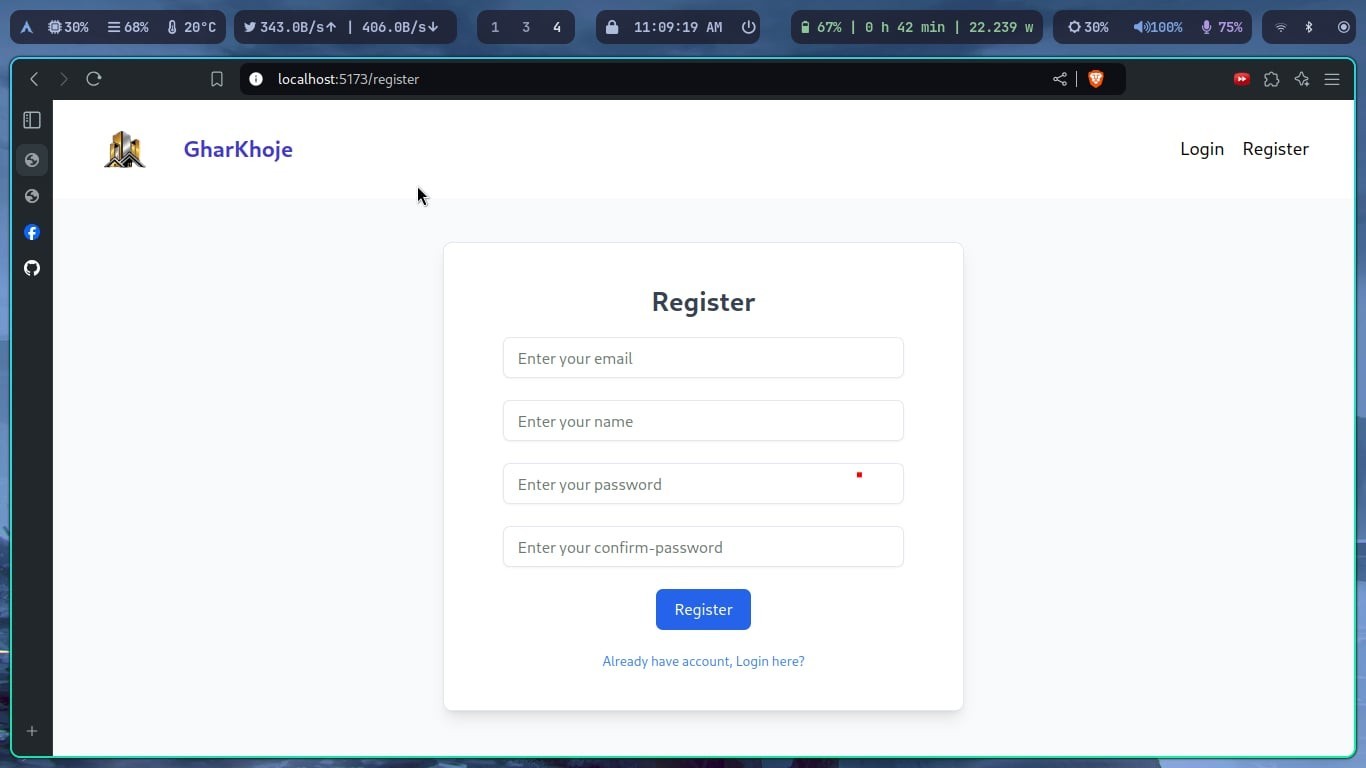
# Appendix

### Screenshot of GitHub Repository

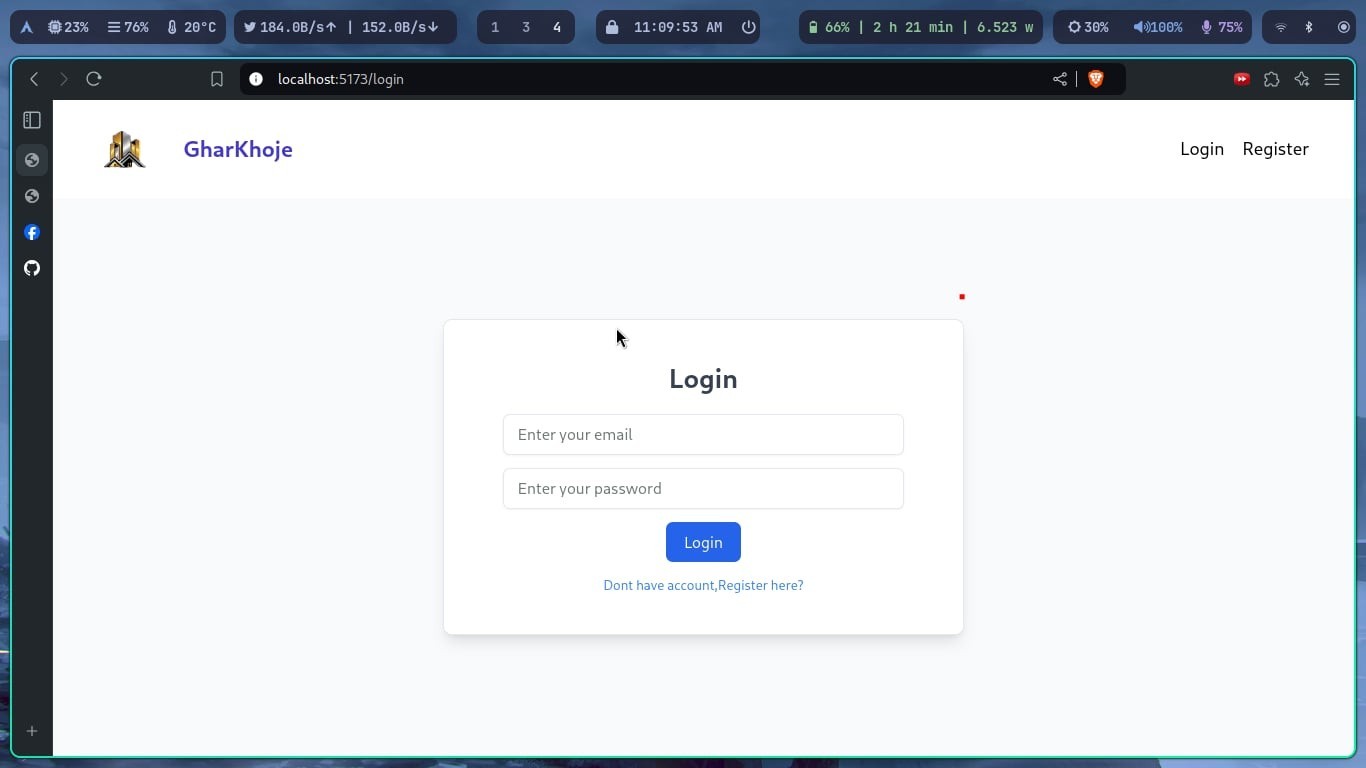
**Screenshot of Home Page**



### Screenshot of Register Page

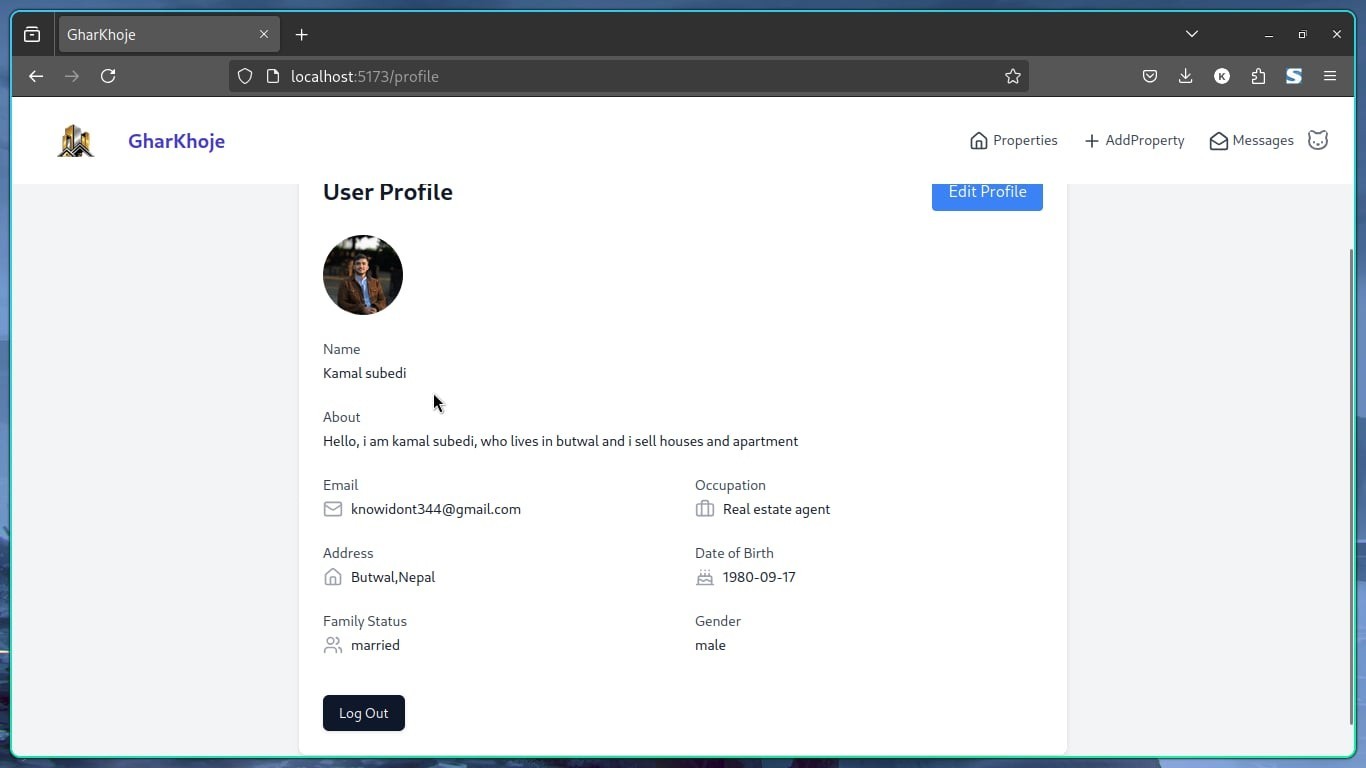


**Screenshot of Login Page**

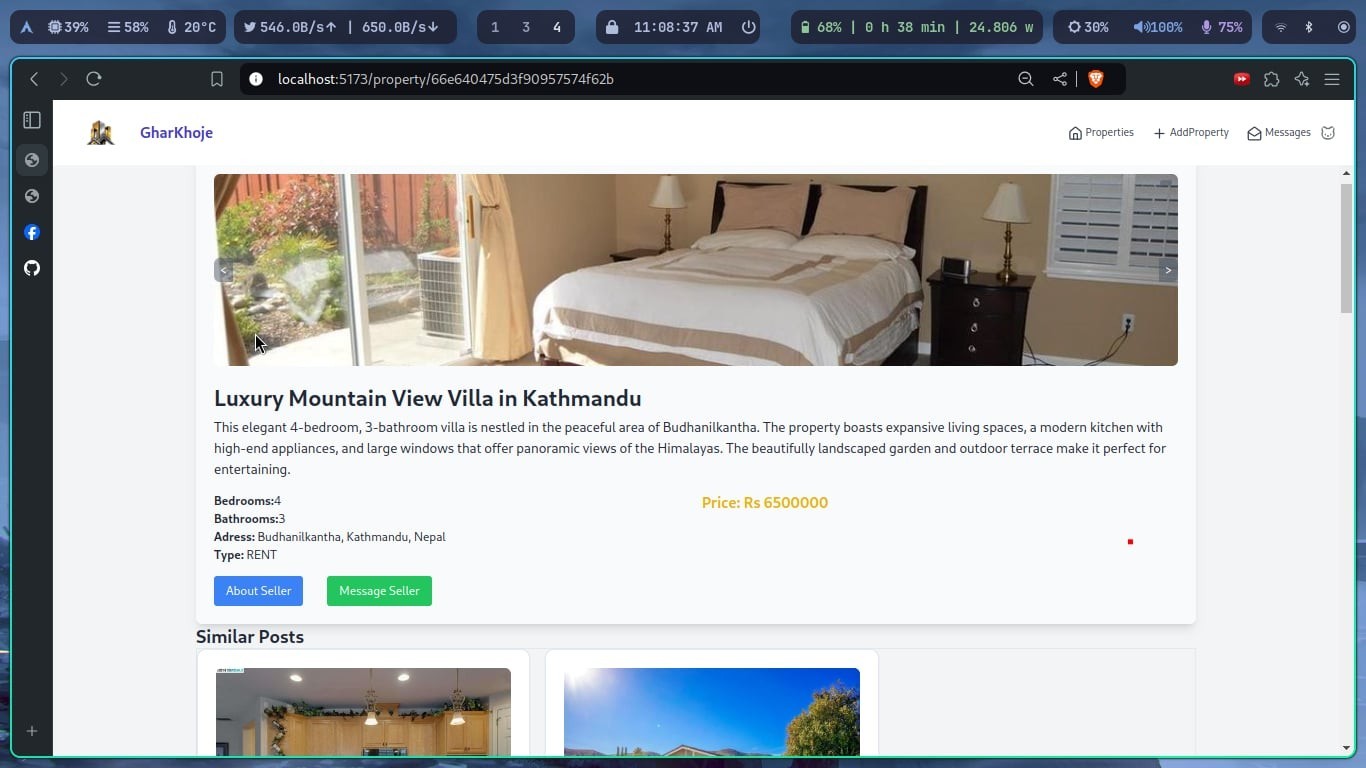


### Screenshot of Add Property Page

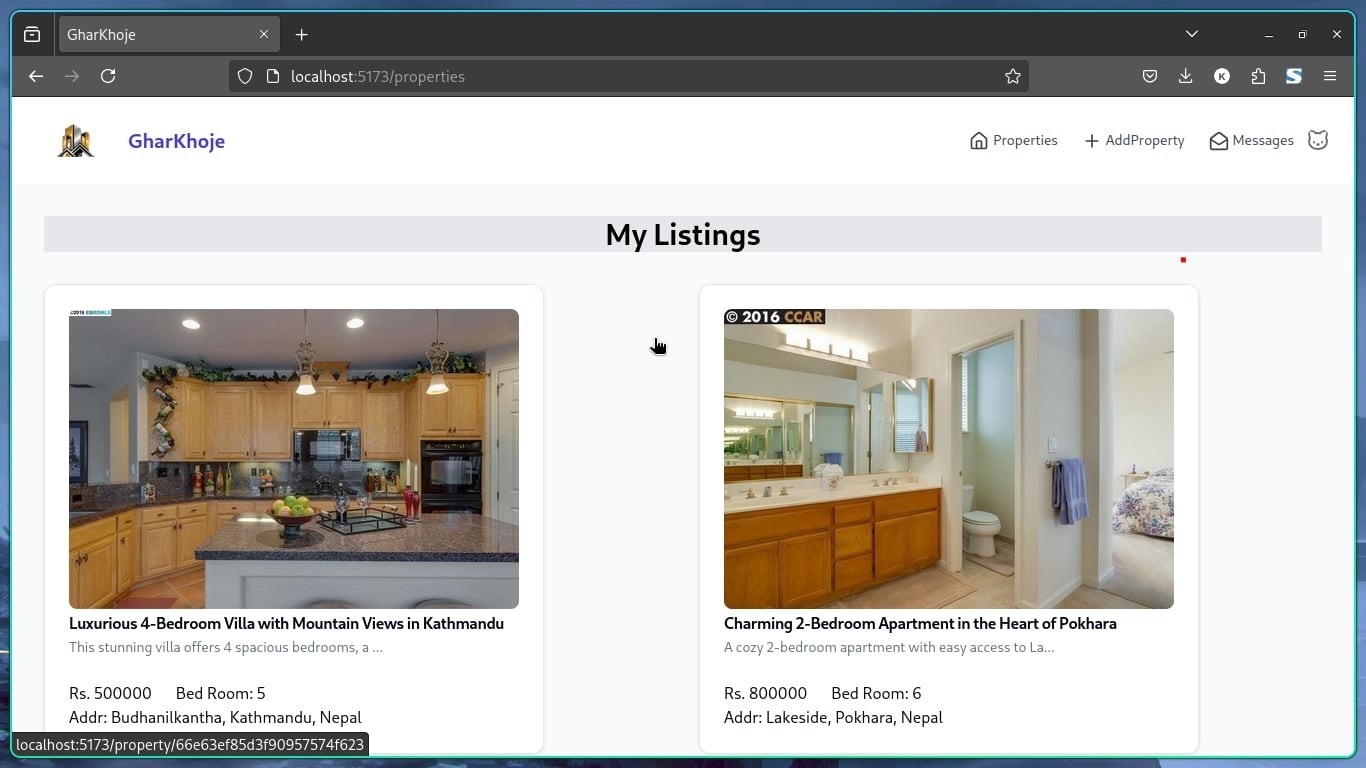
**Screenshot of Profile Page**



### Screenshot of Search Page

**Screenshot of Single Page (property recommendation)**

### Screenshot of Property Image

**Screenshot of Owner Listing page**

### Screenshot of Admin Panel Dashboard