



Tribhuvan University
Faculty of Humanities and Social Sciences

“Room Rental System”

A PROJECT REPORT

Submitted to:

Department of Computer

ApplicationPascal

National College

In partial fulfillment of the requirements for the Bachelors in Computer Application

Submitted by:

Sese Sambahangphe

BCA 6th Semester

University SN: 6-2-1226-25-2019

Symbol No.: 122628582079/05/24

Under the Supervision of

Sanjivan Satyal



Tribhuvan University
Faculty of Humanities and Social Sciences
Pascal National College
Near SwimmingPool, Lalitpur
Bachelor in Computer Applications (BCA)

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project prepared under my supervision by **Sanjivan Satyal** entitled “**Room Rental System**” in the Partial Fulfillment of requirement for the degree of Bachelor in Computer Application is recommended for that final evaluation.

Sanjivan Satyal
Project Supervisor
BCA Department
Pascal National College



Tribhuvan University

Faculty of Humanities and Social Sciences

Pascal National College

Satdobato, Lalitpur

Bachelor in Computer Applications

(BCA)

LETTER OF APPROVAL

This is certify that this project prepared by **Sese Sambahangphe** entitled “**Room Rental System**” in the Partial Fulfillment of requirement for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

Sanjivan Satyal

Supervisor

BCA Department

Pascal National College

Kabindra Koirala

Program Coordinator

Pascal National College

External Examiner

Internal Examiner

ABSTRACT

The Room Rental System is an innovative solution designed to overcome the problem and process of renting rooms/houses/apartments. In today's face-paced world, finding suitable Accommodations can be time consuming and tedious. This system aims to alleviate such problems and challenges by using technology to provide an efficient, user-friendly and secure platform at certain level. The Room Rental System offers attractive set of features to facilitate seamless interactions between landlords and potential tenants. Landlords can effortlessly list their available and vacant property to the system with proper property details like descriptions, images, location, pricing and much more. The Room Rental System is incorporated with advances search algorithms, enabling tenants to quickly find rooms that match their search profile and shows property according to their nearby locations.

Keyword: Room Rental System, React, MongoDB, MS Word

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my supervisor Mr. **Sanjivan Satyal** who gave me the golden opportunity to do this wonderful project on the title Nepali Cuisine Information Management System, which also helped us in doing a lot of research and we came to know about so many new tools and technologies.

I **Sese Sambahangphe** express gratefulness to Pascal National Campus for their guidance and constant supervision as well as for providing necessary information regarding the Project and support in the completion of the project in the given time schedule.

In the end, I would also like to thanks Tribhuvan University for giving me this opportunity via the course of Computer Application to help me to understand the project ethics at this early stages of our upcoming professional journey and helped us to evaluate my knowledge and expand it a little more distance.

Yours sincerely,
Sese Sambahangphe

TABLE OF CONTENTS

SUPERVISOR'S RECOMMENDTATION

LETTER OF APPROVAL

ABSTRACT.....i

ACKNOWLEDEMENT.....ii

TABLE OF CONTENTS.....iii

LIST OF ABBREVIATION.....v

LIST OF FIGURES.....vi

LIST OF TABLES.....vii

CHAPTER: 1 INTRODUCTION1

1.1 Introduction.....1

1.2 Problem Statement.....1

1.3 Objectives.....2

1.4 Scope and Limitations.....2

1.4.1 Scope.....2

1.4.2 Limitation.....3

1.5 Development Methodology.....3

1.6 Report Organization.....4

CHAPTER: 2 BACKGROUND STUDY AND LITERATURE REVIEW4

2.1 Background Study.....5

2.2 Literature Review.....5

CHAPTER: 3 SYSTEM ANALYSIS AND DESIGN6

3.1 System Analysis7

3.1.1 Requirement Analysis.....8

3.1.2 Feasibility Analysis.....9

3.1.3 Data Modeling (ER Diagram).....	11
3.1.4 Process Modeling (DFD Diagram).....	12
3.2 System Design	12
3.2.1 Architectural Design.....	13
3.2.2 Database Schema Design.....	14
3.2.3 Interface Design (UI/UX).....	16
3.3 Algorithm Details	17
CHAPTER: 4 IMPLEMENTATION AND TESTING	20
4.1 Implementation.....	20
4.1.1. Tools Used (CASE tools, programming language, Database platforms.....)	20
4.1.2 Implementation Details of Modules (Description of procedures/functions)	21
4.2 Testing.....	24
4.2.1 Test Case for Unit Testing.....	24
4.2.2 Test Cases for System Testing.....	25
CHAPTER: 5 CONCLUSION AND FUTURE RECOMMENDATIONS	27
5.1 Lesson Learnt/ Outcome.....	27
5.2 Conclusion.....	27
5.3 Future Recommendations.....	28
REFERENCES.....	29
APPENDIX: SYSTEM SCREENSHOTS	

LIST OF ABBREVIATIONS

CRUD	Create, Read, Update, Delete
CSS	Cascade Style Sheet
Express Js	Express Javascript
React Js	React Javascript
MongoDB	Mongo Database
Node Js	Node Javascript

LIST OF FIGURES

Figure 1.1: Waterfall model of Room Rental System.....	3
Figure 3.1: Use Case Diagram of Room Rental System	9
Figure 3.2: Gantt Chart of Room Rental System	10
Figure 3.3: ER Diagram of Room Rental System.....	11
Figure 3.4: Context Level Diagram of Room Rental System	12
Figure 3.5: Level 1DFD of Room Rental System.....	12
Figure 3.6 Layered Architecture Design of Room Rental System.....	13
Figure 3.7 Database Schema Design of Room Rental System.....	14
Figure 3.8 Interface Design of Room Rental System.....	16

LIST OF TABLES

Table 4.2.1 Test Case for User Registration of Room Rental System.....	24
Table 4.2.2 Test Case for adding Property to Room Rental System.....	25

CHAPTER: 1

INTRODUCTION

1.1 Introduction:

With the increase in migration, rapid urbanization and over population, the demand of other necessity things as well as homes for shelter has been increased rapidly recently. In the present situation, to find the rooms, houses as you want at wanted location is very difficult and for searching physically its very tedious work especially in big cities like: Kathmandu, Pokhara, Dharan and so on. In this populated and highly demand of rooms for shelter situation, it is hassle and tedious to wondering around each city, each home in search of rooms. Even though, they may find rooms according to their requirements, sometimes it may not feasible in terms of economic or the location and sometimes tenant has to consider.

So, to overcome such day to day problems related to tenants like mainly students like myself, I chose room rental system related project named as "KothaNepal" for this semester to assist tenants searching rooms as of requirement through online web without stepping outside for searching rooms physically as before. For this project system will be divided into User side and Admin Side, where users can surf rooms according to the location, budget and rooms needed from the user interface whereas admin/landlord can post his/her property to the system through admin side.

1.2 Problem Statement:

With the rapid population growth, various aspect of human needs has been impacted all over the world. House accommodation is one of the aspect that has been impacted badly. When we talk about accommodation condition of Nepal, the condition of accommodation system is very poor. Especially students face significant challenges while seeking house for accommodation. Finding rooms is tedious work due to limited availability, manual search process that might involve physical visit of homes. Many tenants resort brokers, pay certain amount of fees to secure rooms in densely populated areas. Certain times cultural barriers further complicate the search of rooms in densely populated areas.

1.3 Objectives

Objectives of the system are:

- To develop a system that allows users to get information about vacant rooms through online.
- To provide facility to landlord to advertise their property recently on vacant by providing facility to add, update and delete the property details.

1.4 Scope and Limitation

1.4.1 Scope

This system can be very useful and helpful for those who are eagerly searching flat, rooms, apartment or even whole house in this hustling and bustling cities. This system has provided leverage and advantage to post free advertisement of rooms, apartment, and houses that are vacant. Normal user can post ads by simply creating an account with no cost at all. As this system has likely being completed, it has huge potential and room to be improved as the technology is being developing and changing at its highest speed. Various features can be added to the system to make the system more practical according to the time and situation.

1.4.2 Limitation

As system has been developed by aiming the crucial objectives of the system, but can be enhanced by adding more additional features, which has become the limitation of the system. Some of the limitations of the system are as follows:

- It doesn't provide any notification to the users/landlord who advertised their property until users/tenant manually contact the landlord for more update like whether particular property is still vacant or not.
- Users aren't able to wish list their favorite property for future purpose.

1.5 Development Methodology

For the development of this system, functional oriented programming approach is used that includes data flow diagram, entity-relational diagram, use case diagram and much more. Waterfall software development methodology is implemented in this system as development methodology, which consists various steps and levels.

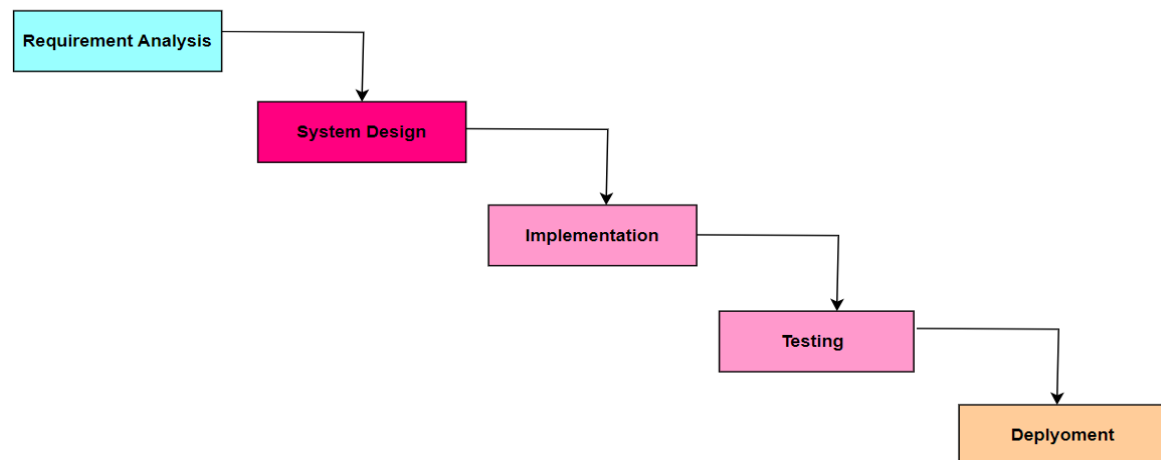


Figure 1.1: Waterfall Model of Room Rental System

1.6 Report Organization

The report has been structured as the guidelines provided by the Tribhuvan University. This report is divided into different chapters, each chapters consists various sub chapters and its content. It is divided into five different chapters as following:

Chapter 1: Introduction

This chapter deals with the introduction of the system with its objectives and limitations along with the reason why the system is made.

Chapter 2: Background Study and Literature Review

This chapter summarizes the work that has been carried out in the field of data collecting and also describes the features about some existing applications related to the. Room Rental System that are on existence at the market.

Chapter 3: System Analysis and Design

This chapter focuses on the different requirement of the system, which describes about the functional, non-functional, feasibility analysis, Entity Relational diagram, Data Flow Diagram, design of the system with system architecture, database schema, and interface design.

Chapter 4: Implementation and Testing

This chapter emphasizes tools used in system development, implementing details and result of test performed.

Chapter 5: Conclusion and Future Recommendation

This chapter highlights brief summary of lesson learnt, outcome and conclusion of the whole project and explain what have been done and what further improvements could be done.

The final portion of the report consists of References and Appendices. The References are listed according to the IEEE referencing standards.

CHAPTER: 2

BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

“Renting Accommodation” is defined as property owned by someone other than the resident or by legal entity for which the resident pays a periodic rent to the owner; simply a formal or informal agreement between a tenant and landlord to rent dwelling for a certain period of time at a predetermined price.

As migration to the developed city from remote place has been increasing, it leading to overpopulation at certain big and developed cities directly effecting various factors required for living like health, education, accommodation and so on. Accommodation is the one of the factors that is being hugely effected by over population. As houses and land for the shelter are limited for certain physical place and as population grows more it directly effects the price of the accommodation and over on that finding accommodation according to one’s budget becomes more difficult.

2.2 Literature Review

Many property rental systems similar to the Room Rental System are providing their services within Nepal. I searched and tried to explore some of few systems that are similar to this system. Some of them are:

Initially, found website named “**rentalnepal.com**”. This website is providing facility to rent from whole house to flat. Not only houses but this system is providing information about shutters as well as lands that are available to be rented. It has normal User Interface design but not as modern day’s website design, which I found con side of this rental system. But overall, it is well managed as it has different categories related to real estate along with facility to search particular property as of user needs. [1]

Another similar system named “**gharbeti.com**”. It has simple User Interface design. It is also providing rental service of buildings, flats, rooms, space, land and hostel as well. It has provided various query option where user can easily search properties of their choices by filtering query dropdown button. As far I experienced, it is providing facility as some broker but through online platform. [2]

Other different rental system that has most beautiful User Interface compared to other above

listed system. Properties are managed very properly and information are shown in very detail with high quality images and with rich information related to the properties. It is showing properties that are not only on rent but also those are in the sell. It is providing 'pricing' service through which landlord can purchases the subscription through which one can list ads of property in this system. Subscriptions were according to the certain amount of money like normal listing, premium listing and top listing and the price was in ascending order. [3]

CHAPTER: 3

SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

For this project, waterfall model will be used as Software Development Life Cycle. Waterfall model has five phases: Requirements analysis and specification, design, implementation and unit testing, deployment and maintenance. In this model, one phase doesn't overlap another phase and always starts with top to bottom approach. It starts with requirements analysis where the customer and software developer work together so as document all the function, performance of the software which is followed by designing phase, implementation, testing, deployment and maintenance phase. Output of one phase becomes the input of the next phase. All requirements are fixed for this project as initial phase considering all the resources are available.

3.1.1 Requirement Analysis

The requirement analysis is the initial step that helps to define, document, validate, manage and fulfill the expectations and the needs of users and stakeholders for the software or the application. In other terms, the requirement analysis is basically process of analyzing and refining the requirements gathered from various resources to ensure that the final system meets the requirements, needs and the expectations.

i. Functional Requirements:

Different functional requirements of the system have been identified as below:

For Users:

- The system displays properties/room to the users according to their queries.
- The system allows user to query properties according to their geo-location.

For Administration:

- The system allows admin to add, update, and delete properties/rooms and their detail information.
- The system allows access to the admins with proper authorization for performing CRUD operation through admin dashboard page.

ii. Non-Functional Requirements:

Different non-functional requirements of the system have been identified as below:

Availability:

The system is available 24/7 and should be accessible through online and from anywhere.

Security:

The system should have strong authentication and authorization process to prevent unauthorized access to the system.

Performance:

The system should be able to handle large number of volume of data and traffics at same time.

Compatibility:

The system should be compatible with different browsers, operating systems.

USECASE DIAGRAM

The Use case Diagram consist of a user who is allowed to create an account. After

registration user can login to the system, at this point user can add property to the system if one wishes to add property to the system and act as the admin of the system to manipulate the information of the property one can add to the system and can search it.

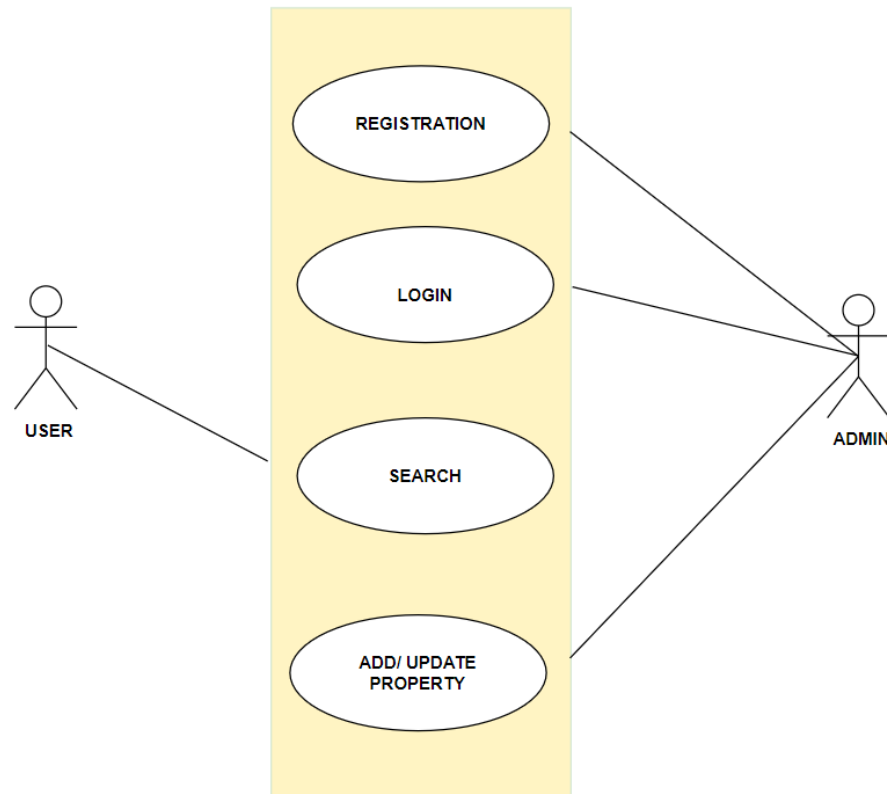


Figure 3.1: USECASE DIAGRAM

3.1.2 Feasibility Study

A feasibility study is an analysis that consider all of a project's affecting factors like economic, technical, legal and scheduling considerations. The feasibility study was conducted and was planned.

i. Technical Feasibility Study

The system is technically feasible as the requirements for the development of the system id easily accessible. The necessary hardware and software required for the implementation of the system was available. The academic programming language was enough to develop this system with available hardware resources like our laptopsand the internet and software that was available open source in the internet.

ii. Economic Feasibility:

The system is economic and cost effective. As all the tools and software resources

are free and open source for use. The existing software was enough for this system. No monetary cost was required to develop this program. All the basic resources required for this system were available for and with us so no extra money was required.

iii. Operational Feasibility:

The system is very easy and user friendly. It can be operated with the basic knowledge of the computer and internet. Users can access the system as long as they are online. Only an authenticated administrator is authenticated to make any changes to the system. Users are required to login to the system to save their wish listed rooms for future reference.

iv. Schedule Feasibility:

The system is completed within scheduled time and does not exceed the projected schedule time.

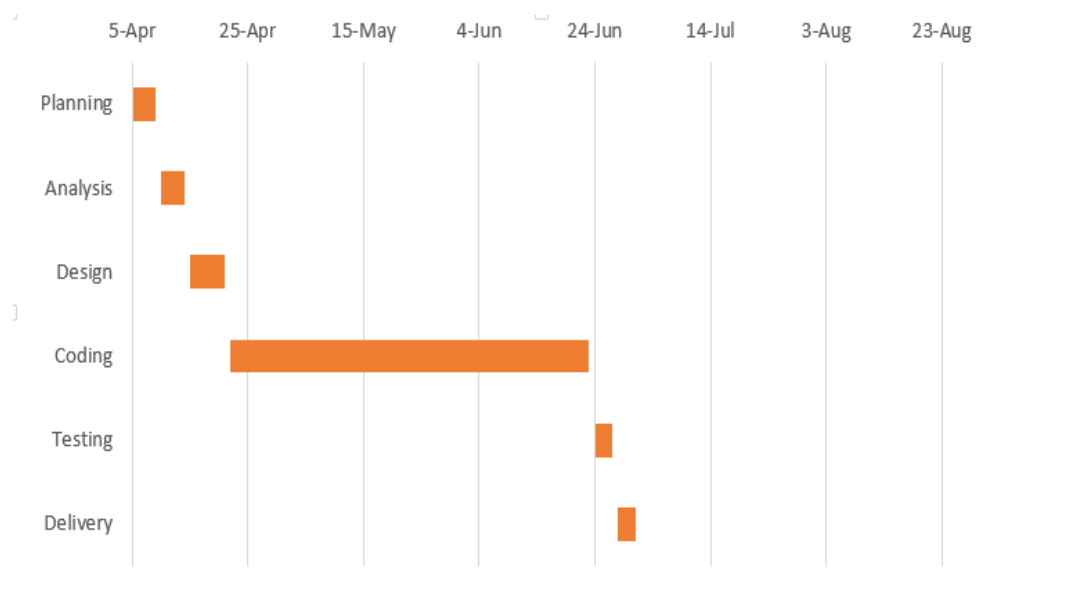


Figure 3.2: Gantt chart of Room Rental System

3.1.3 Data Modeling: (ER Diagram)

ER Diagram stands for Entity-Relationship Diagram. It is a high level data model which is used to define the data elements and relationship for the system. It develops a conceptual design for the databases which develops a very simple and easy to view the data.

For this system, admin/user and property are the entities and each entity has their own attributes like admin/user has attributes such as: phone, first_name, last_name, and password. Likewise, property has type, location, price, kitchen, bedrooms as shown in the figure.

In this system user and admin have similar attributes, as user can register and login to the system and if user successfully login to the system then user is authorized to act as the admin to the system. It is basically like C2C.

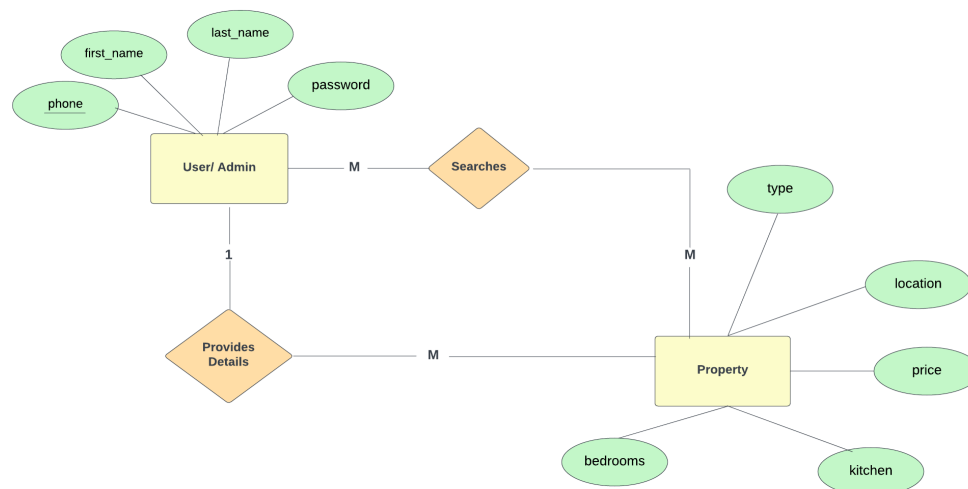


Figure 3.3: ER Diagram of Room Rental System

3.1.4 Process Modeling: (DFD Diagram)

A Data Flow Diagram shows how information does flow for any process or the system. It uses symbols like rectangles, circle and arrows to represent the flow of the data. DFD describes the processes that are involved in a system to transfer the data from the input to the file storage.

This system consists of 2 levels of DFD diagram i.e. Level 0(context level) and Level 1 which explains the flow of data in system in more detail.

Context Level Diagram:

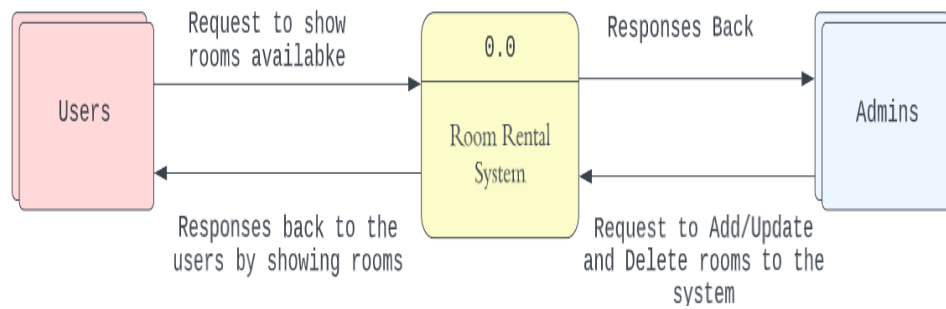


Figure 3.4: Context level diagram of Room Rental System

Context level represents the flow of the data in the system in contextual level. In this level, user can request the system to show the property according to the interest of a user and admin can simply manipulate the information and details of the property s/he added to the system. In this system normal user can become admin by just logging to the system and add and update the property details. But the catch is admin can only manipulate the information and details of the property that s/he added to the system and isn't allowed to manipulate the details of the property of other user who became admin to post property to the system.

First Level Diagram:

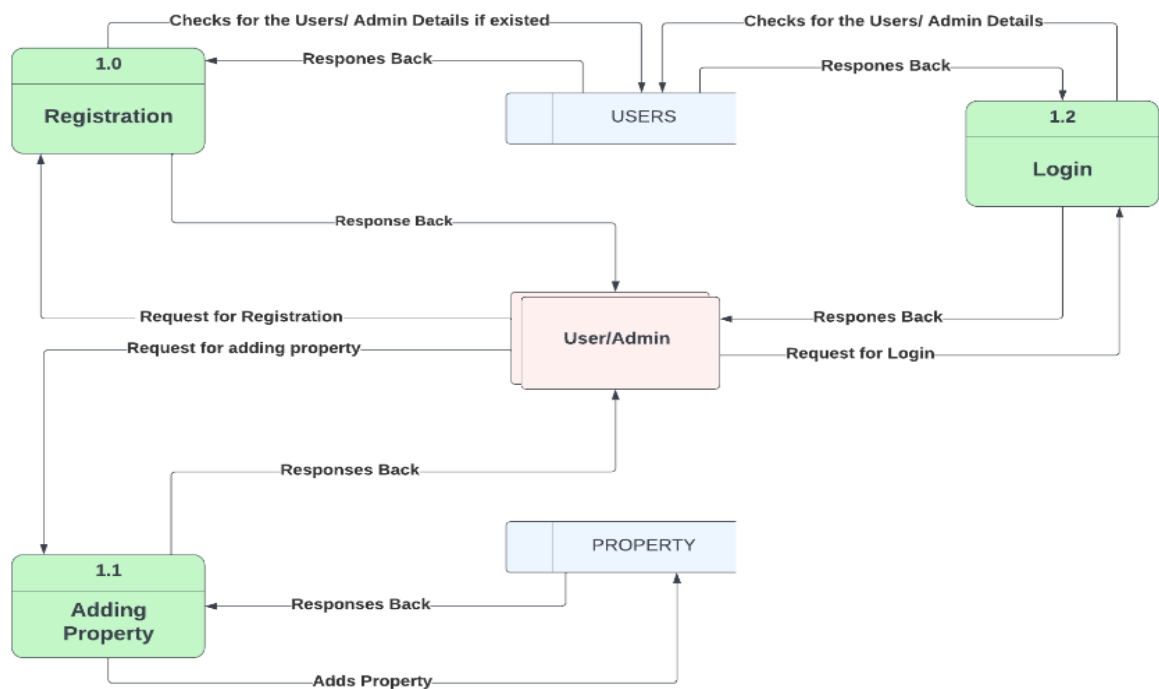


Figure 3.5: Level 1DFD diagram of Room Rental System

In the level 1 DFD of the system, there are three processes like: 'Registration' and 'Login' processes for user/admins, and 'Adding Property' process.

In this system, normal user/ visitor can become an admin just by logging to the system after successful registration process. Normally, user can request the system for the registration which enables to access more features system is capable of providing. If registration process goes smooth, then user can easily login to the system by simply requesting system for login by providing authenticated login information. After successful login to the system, user becomes one of the admin to the system, as this system is based on customer- customer concept any one can become an admin of the system, but one authorized admin can't manipulate details and information about property of other admins.

3.2 System Design

System Design is the process of designing the elements for the system, and the process of defining the architecture, components, interfaces, and data for a system that meets the needs and requirements of the particular organization. The Room Rental System is designed using various schema designs, CASE tools.

3.2.1 Architectural Design

Architectural Design in software development involves creating a structured plan or blueprint for the system's organization, components, and their interactions. It does addresses the high-level structural elements of a software system and how they work together to achieve the desired functionality, performance and the other things.

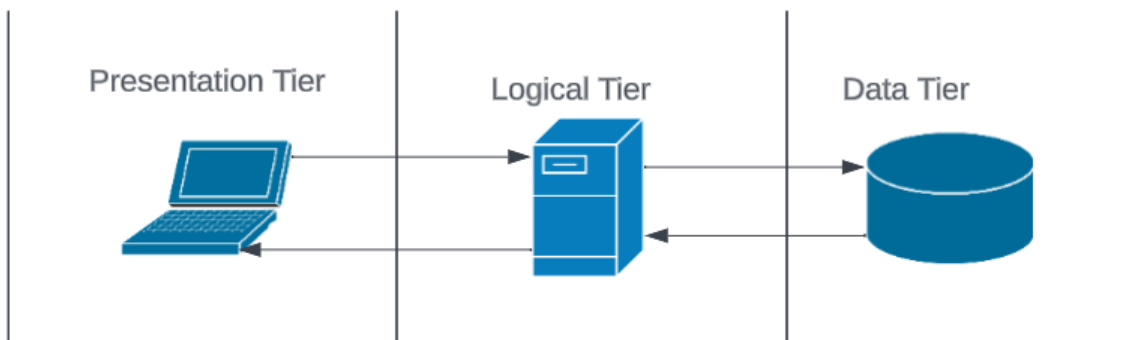


Figure 3.6: Layered Architectural Design of Room Rental System

3.2.2 Database Schema

Basically a database schema is a logical framework that defines the structure and organization of database which acts as a plan for how data is organized, stored and accessed within a database management system.

For creating a diagram for MongoDB database schema, one can be very helpful for visualizing the structure of data and understanding the relationships between different collections.

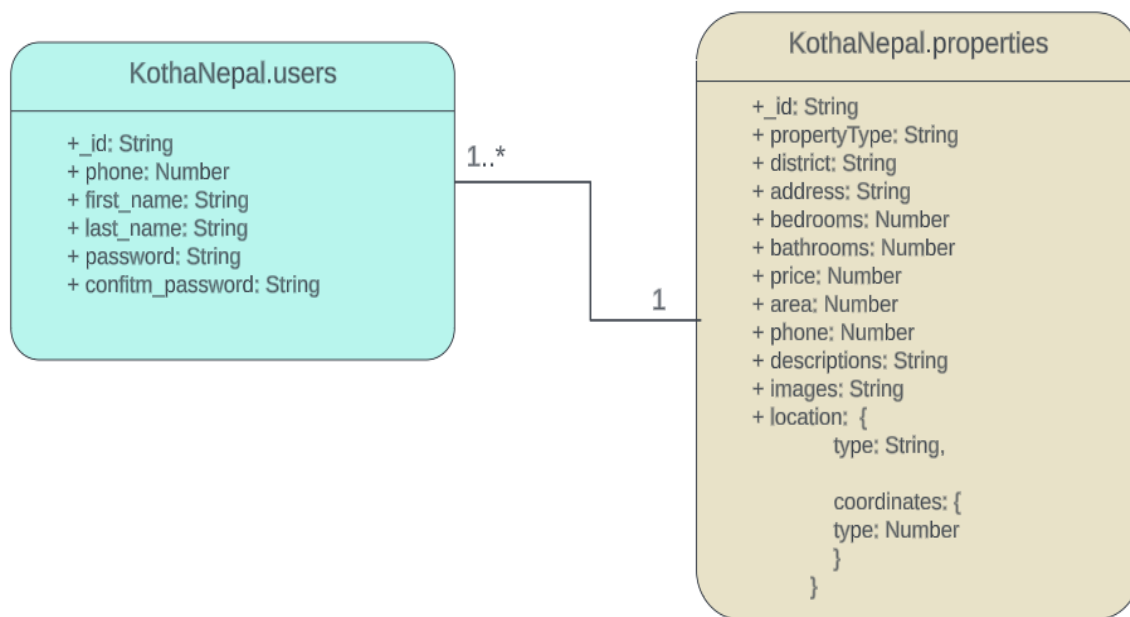


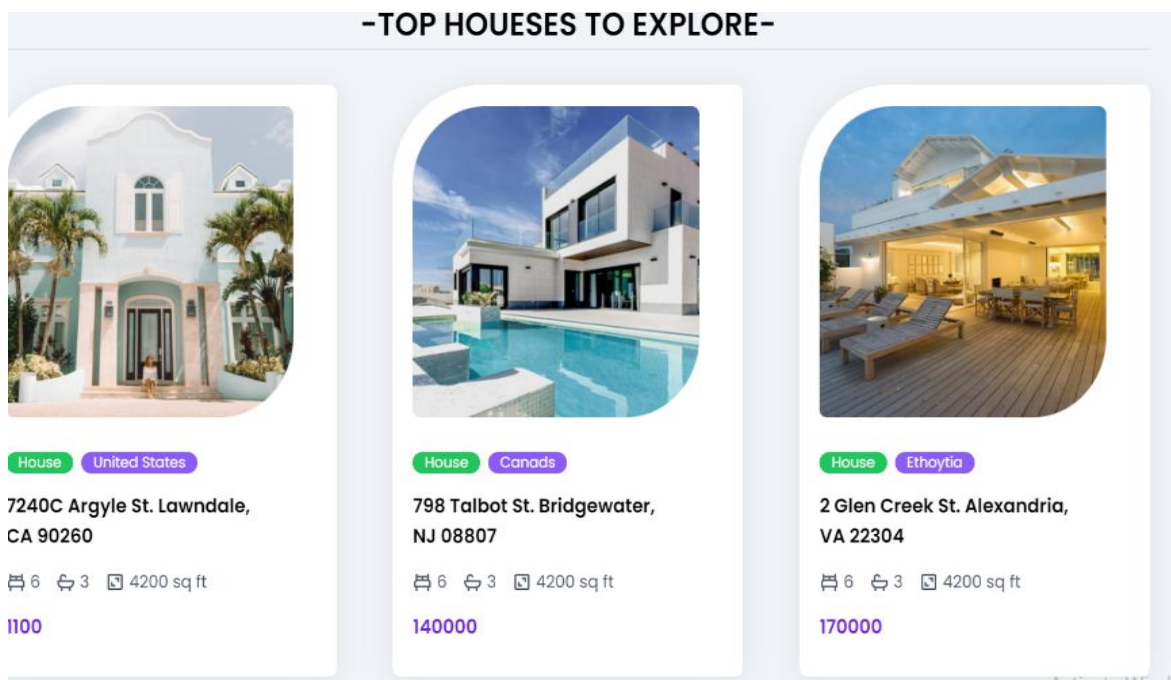
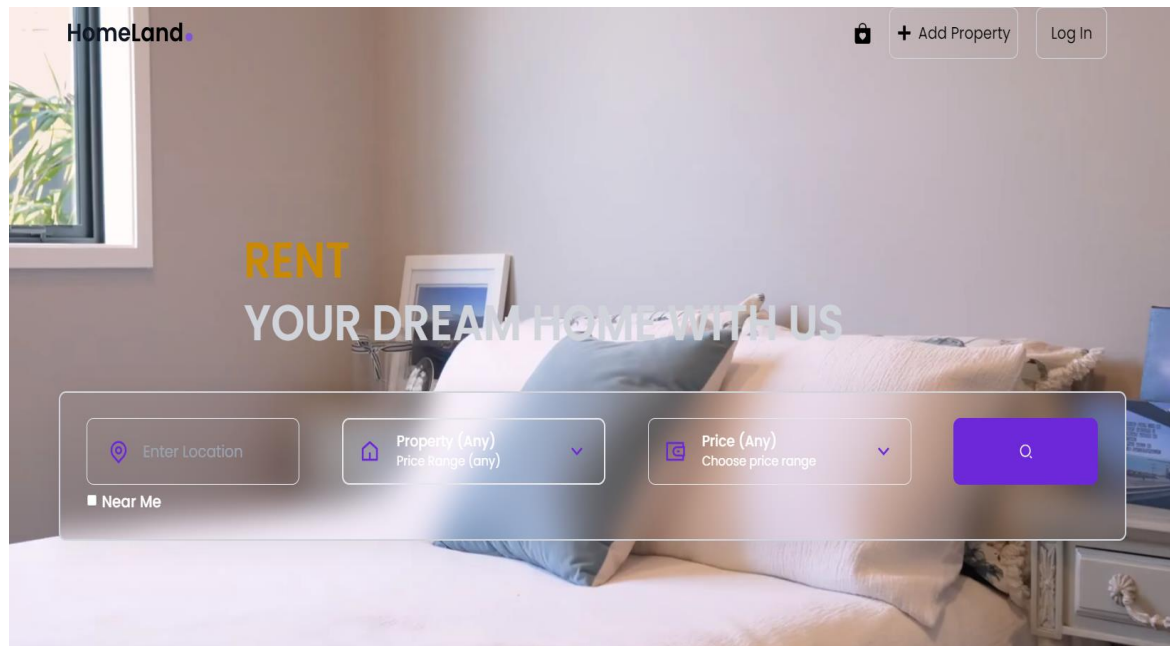
Figure 3.7: Database Schema Design of Room Rental System

In this above figure, two collections named 'users' and 'properties' are shown visually. In users and properties collection a list of fields along with their respective data types are shown.

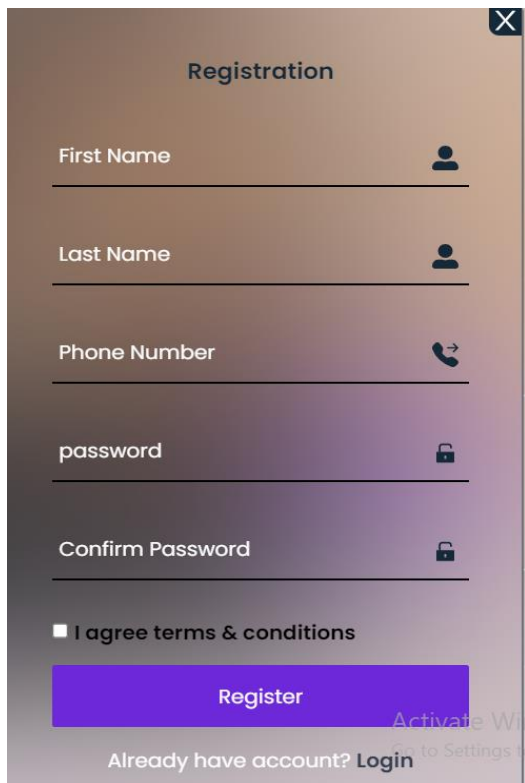
For 'users' collection the fields and their respective data types are: '_id, phone, first_name, last_name, password, confirm_password' and for properties: '_id, propertyType, district, addresses, bathrooms, price, areas, phone, descriptions, images and location'.

3.2.3 Interface Design (UI/ UX)


Interface Design, often referred as UI/UX Design, is very crucial aspect of creating digital websites and applications. It mainly focuses on creating user interface that are user-friendly, visually appealing and provide seamless user experience.





HomePage UI





Registration

First Name 

Last Name 

Phone Number 

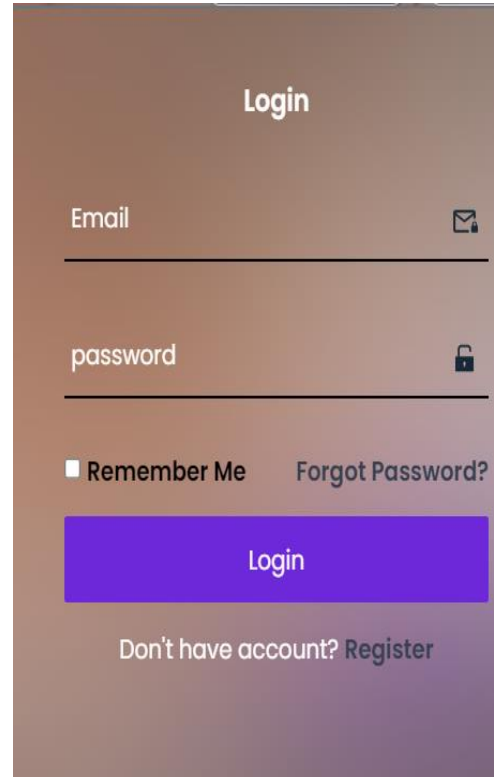
password 

Confirm Password 


☐ I agree terms & conditions


Register

Already have account? [Login](#)



Login

Email 

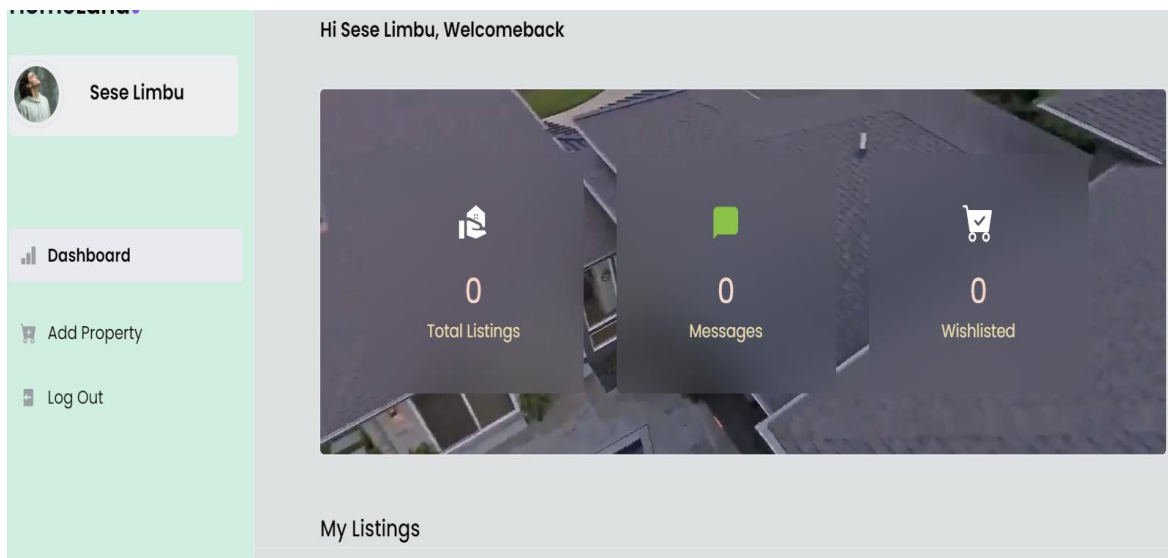
password 

☐ Remember Me [Forgot Password?](#)


Login


Don't have account? [Register](#)


Registration & Login UI





Hi Sese Limbu, Welcomeback


 **Sese Limbu**


 **Dashboard**

 Add Property

 Log Out

 **0**
Total Listings

 **0**
Messages

 **0**
Wishlisted

My Listings

Admin Dashboard UI

Figure 3.8 Interface Design of Room Rental System

3.3 Algorithm Details

Geospatial Distance Calculation Algorithm

Geospatial Distance Calculation Algorithm measures distance between two geographic points of the Earth. It represents a complex solution to precisely compute distances between two points of the Earth. At its core, Haversine formula is crafted for geospatial distance measurement. Haversine formula based algorithm is widely used for calculating great circle distance on the sphere.

Overview of an Algorithm

Geospatial distance calculation algorithm is based on Haversine Formula. This formula is widely used to compute distance between two points on the surface of a sphere, such as the Earth because of its precision and accuracy. The algorithm generally use following formula:

$$a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2\left(\frac{\Delta long}{2}\right)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

Where,

- **Δlat** is the difference between latitudes of two points,
- **$\Delta long$** is the difference between longitudes of two points,
- **lat1 , lat 2** are the latitudes of the two points,
- **R** is the radius of the Earth

Working mechanism of an Algorithm

Let,

- lat_1 and lon_1 be the latitude and longitude of the first point of the radius,
- lat_2 and lon_2 be the latitude and longitude of the second point of the radius,
- $\Delta lat = lat_2 - lat_1$,

- $\Delta \text{lon} = \text{lon_2} - \text{lon_1}$,
- **R** be the radius of the Earth

The formula given by the algorithm,

$$a = \sin^2\left(\frac{\Delta \text{lat}}{2}\right) + \cos(\text{lat_1}) \cdot \cos(\text{lat_2}) \cdot \sin^2\left(\frac{\Delta \text{long}}{2}\right)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

Lets take an example:

Let, Point 1: Latitude (lat_1) = 37.7749 degree, longitude (lon_1) = -122.449 degree

Point 2: Latitude (lat_2) = 34.0552 degree, longitude (lon_2) = -188.2437 degree

- **Converting Coordinates to Radians:**

Convert the latitude and longitude degree values to radians.

$$\text{lat_1} = 37.7749 \times (\pi/180) = 0.6594$$

$$\text{lon_1} = 122.4194 \times (\pi/180) = -2.134$$

$$\text{lat_2} = 34.0522 \times (\pi/180) = 0.5947$$

$$\text{lon_2} = 118.2437 \times (\pi/180) = -2.066$$

- **Calculate the differences:**

Compute the difference between Δlat and Δlon

$$\Delta \text{lat} = (\text{lat_2} - \text{lat_1}) = 0.0647$$

$$\Delta \text{lon} = (\text{lon_2} - \text{lon_1}) = 0.068$$

- **Use Algorithm formula:**

Calculate **a** using the Haversine formula based algorithm

$$a = \sin^2\left(\frac{\Delta \text{lat}}{2}\right) + \cos(\text{lat_1}) \cdot \cos(\text{lat_2}) \cdot \sin^2\left(\frac{\Delta \text{long}}{2}\right) = 0.0071$$

- **Central Angle (c):**

Compute the central angle

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) = 0.1333$$

- **Calculate Distance:**

Finally, use central angle to calculate the distance using the Earth's Radius (**R**):

$$\text{distance} = \mathbf{R} \cdot \mathbf{c} = 6371 \times 0.1333 = 849.4 \text{ km away}$$

CHAPTER: 4

IMPLEMENTATION AND TESTING

4.1. Implementation

4.1.1. Tools Used (CASE tools, Programming language, Database platforms)

Following are the tools and framework used for the accomplishment of this project:

Front End Tools

➤ React Js

For this system, React Js is used as the front end technology. React Js is a front- end library and framework developed by Facebook used for building interactive user interfaces and web applications quickly and more effectively with less amount of code compared to vanilla Js.

Back End Tools

➤ NodeJs (Express Js)

For this system, Node Js is used as back end technology. Node Js is simply an open-source and cross platform java runtime environment. Node.js run JavaScript code at backend by providing runtime environment outside of the normal browser.

Database

➤ MongoDB

For this system, MongoDB is used as database to store data. It is an open source NoSQL database management program. NoSQL is used as an alternative to traditional relational database. I t manages document –oriented information, store or retrieve data that are stored in JSON format.

Documentation Tools

1. MS Office

This office package is used for presenting, writing the documents for this system.

2. Draw.io

Draw.io is used to draw various diagrams like ER, DFD and others required for this system documentation.

4.1.2. Implementation details of modules

```
const allProperties = await property.find();
console.log(allProperties)

const nearbyProperties = allProperties.map(property => {
  const plainProperty = property.toObject();
  const propertyLatitude = plainProperty.location.coordinates[1];
  const propertyLongitude = plainProperty.location.coordinates[0];

  const distance = haversine(latitude, longitude, propertyLatitude, propertyLongitude);

  return {
    ...plainProperty,
    distance,
  };
}).filter(entry => entry.distance <= 4);
```

The above code snippet is equivalent to the Haversine formula based algorithm. It basically takes two parameters named **longitude** and **latitude** from user and calculate distance on the basis of the Haversine algorithm within max distance of 4000m/ 4km and return the calculated data through local API.

The sample data it has returned:

```
Array(2) 1
  0:
    address: "Kathmandu"
    area: 1221
    bathrooms: 1212
    bedrooms: 12
    descriptions: "1232132"
    dist: {calculated: 2032.9181674084216}
    district: "Lalitpur"
    images: "1700968567587-images_4.jpg"
    location: {type: 'Point', coordinates: Array(2)}
    price: 1220
    propertyType: "House"
    __v: 0
    _id: "6562b8772a0fd72798409d4e"
    [[Prototype]]: Object
  1:
    address: "Ranigaun"
    area: 122
    bathrooms: 12
    bedrooms: 12
    descriptions: "1221323"
    dist: {calculated: 2032.9181674084216}
    district: "Panchthar"
    images: "1700969218445-images_1.jpg"
    location: {type: 'Point', coordinates: Array(2)}
    price: 300000
    propertyType: "Apartment"
    __v: 0
    _id: "6562bb023d8c949f5ed16989"
    [[Prototype]]: Object
length: 2
```

[NearMe.js](#)

Activate Windows

Here, we can see two array of objects containing certain content in **key: value** pair. This code snippet library works similar as Haversine formula as mentioned in the algorithm details.

```

if (navigator.geolocation) {
  try {
    navigator.geolocation.getCurrentPosition(
      (position) => {
        const latitude = position.coords.latitude;
        const longitude = position.coords.longitude;
        console.log(`latitude: ${latitude}, longitude: ${longitude}`);

        setProperty({
          ...property,
          latitude: latitude,
          longitude: longitude,
        });

        setStates(true);
      }
    );
  }
}

```

The above code snippet shows access of user current user location. It has two variables latitude and longitude which stores current longitude and latitude and updates to the property named variable.

The snippet program show result as:

27.6476968 85.3446221

```

▼ {longitude: 85.3446221, latitude: 27.6476968} ⓘ
  latitude: 27.6476968
  longitude: 85.3446221
  ► [[Prototype]]: Object

```

Here, we can see the object data containing latitude and longitude with each containing respective values.

```

const GetData = async (e) => {
  e.preventDefault();

  const userValidation = await userFormValidation(loggedInUser);

  if (Object.keys(userValidation).length === 0) {
    setDataErrors([]);
  } else {
    setDataErrors(userValidation);
  }
};

```

The above code snippet shows the login module logic to check whether the user's data is valid or not by asynchronously fetching data from the database through local API.

The snippet program shows end result as:

[UserDatasContext.js:7](#)

```
{_id: '64b25c80a9d90219e61679fc', first_name: 'sese', last_name: 'eses', phone: 9840
288392, password: 'il', ...} i
  confirmpassword: "il"
  first_name: "sese"
  last_name: "eses"
  password: "il"
  phone: 9840288392
  __v: 0
  _id: "64b25c80a9d90219e61679fc"
  ► [[Prototype]]: Object
```

```
const submitForm = async () => {
  console.log("button is working well bth")
  const formData = new FormData();

  formData.append("propertyType", property.propertyType);
  formData.append("district", property.district);
  formData.append("address", property.address);
  formData.append("bathrooms", property.bathrooms);
  formData.append("bedrooms", property.bedrooms);
  formData.append("price", property.price);
  formData.append("area", property.area);
  formData.append("descriptions", property.descriptions);
  formData.append("images", property.images);
  formData.append("latitude", property.latitude);
  formData.append("longitude", property.longitude);

  try {
    const res = await fetch("/properties", {
      method: "POST",
      body: formData,
    });

    if (!res.ok) {
      console.log("Error from server");
    }

    const data = await res.json();

    if (data.status === 422 || !data) {
      window.alert("Unsucessfull");
      console.log("Registration incomplete!");
    }
  }
}
```

The above code snippet, shows the function to post data to the database by binding in FormData() through “/properties” named API.

4.2. Testing

System testing is basically done to check whether our system is working as a whole system as planned. Testing is the crucial steps in software development. System is tested time and again to increase the accuracy on different situations that system may face in near future.

4.2.1. Test cases for Unit Testing

Table 4.2.1: Test case for User Registration to the system

S.N	Test Name	Input	Expected Output	Actual Output	Test Result
1	Open Room Rental System	http://localhost:3000/register	Register component of the room rental system	Redirected to the Register form of the system.	Pass
2	Enter first name, last name, phone number, password and confirm password	First Name: Sese Last Name: Limbu Phone: 9804979923 Password: limbusese123 Confirm Password: limbusese123	Redirected to the home page.	Redirected to the home page of the room rental system.	Pass
3	Enter phone number and valid password	Phone Number: 9940288393 Password: il	Redirect to the home page	Redirect to the home page with new user profile dashboard	Pass

4.2.2. Test cases for System Testing

Table 4.2.2: Test case for adding new property to the system successfully

Test Case 1	Adding property to the system successfully!
Test Data	PropertyType: Apartment District: Lalitpur Address: Kharibot, Imadole Bedrooms: 12 Bathrooms: 2 Price: Rs: 12000 Area: 1200 sq.ft Description: This is the property Image: property_1.png
Expected Result	Browser should show an alert message mentioning “Successful addition of the Property Details”
Tested Result	Added Successfully

Table 4.2.3: Test case of failure for adding new property to the system

Test Case 1	Failure of adding property to the system
Test Data	PropertyType: Apartment District: Lalitpur Address: Kharibot, Imadole Bedrooms: Bathrooms: 2 Price: Rs: 12000 Area: 1200 sq.ft

	<p>Description: This is the property</p> <p>Image:</p>
Expected Result	Form should display an error mentioning “Please enter valid number of bedrooms” for Bedrooms field and “please insert an image” for Image field.
Tested Result	Form displayed an error mentioning “Please enter valid number of bedrooms” for Bedrooms field and “please insert an image” for Image field.

CHAPTER: 5

CONCLUSION AND FUTURE RECOMMENDATIONS

5.1. Lesson Learnt / Outcome

Every project makes us to learn and gain the knowledge in different aspects. Likewise other projects, I have learned lots of problem solving skills and ways to tackle it from different prospective, proper use of guidance, communication along with writing skills and management of time for each task.

➤ Problem Solving Skills

This project helped to learn lots of problem solving skills related to software development practice and to recognize different errors occur while developing system and solve it.

➤ Writing Skills

This project helped to learn to prepare documents related to the project with more systematic and grammatically error free than before as well as learned to use different case tools for drawing diagrams.

➤ Management of Time

This project helped to learn the most important aspect of software development process i.e management of time according to the complexity of the system modules and components.

➤ Learned New Programming Language and Framework

This project helped and gave opportunity to learn new programming language and its related framework to complete project.

5.2 Conclusion

This project is developed focusing on tenants who are searching accommodation at particular place according to the interest and budget of the tenant. As population is increasing at particular physical place, the availability of accommodation according to the location and budget of tenant at same has become extremely difficult. Over on that, searching flat/room in each home asking landlord is more tedious. To overcome such problem and eradicate such problem at certain amount, this system might be very helpful.

Tenant can just query according to the budget, location and type of the property through online just by visiting this system.

5.3 Future Recommendation

In the near future this system might become more helpful and useful as population increases at certain places like developing and developed cities. The future recommendation of this system are as below:

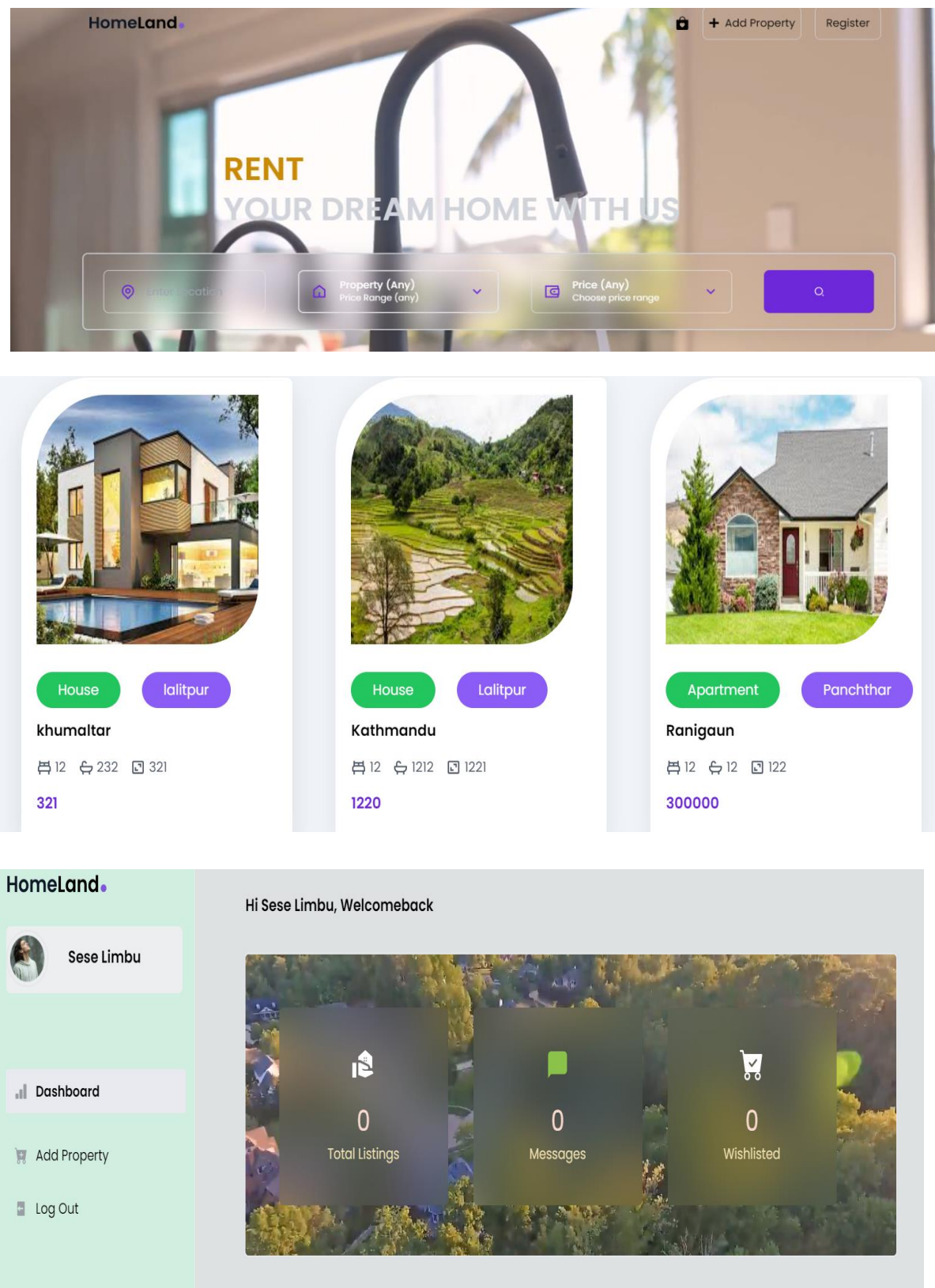
- More Interactive and user friendly.
- User can wishlist their liked property for future reference.
- Landlord can get notification about who liked and how many visitor visited his/her posted property on the system.
- OTP (One Time Password) feature can be added to the system.
- More Scalable

References

- [1] 2020. [Online]. Available: www.roomrentalnepal.com. [Accessed 2023].
- [2] 2019. [Online]. Available: <https://www.gharbeti.com>. [Accessed 2023].
- [3] P. C.L, "The ecological footprint of hostel tourists in Ontorio and Quebec," 2008.
- [4] I. G. P. C. Taffin, "Support Measures to Promote Rental Accomodation for Low-Income Groups," 2013.

APPENDIX: SYSTEM SCREENSHOTS

Screenshots of the system are:




```
import React, { createContext, useState, useEffect } from "react";

export const UserDatasContext = createContext({ matchedDatas: null, updateMatchedData: () => {} });

const UserDetails = ({ children, matchedDatas }) => {
  console.log(matchedDatas);

  return (
    <UserDatasContext.Provider value={{ matchedDatas }}>
      {children}
    </UserDatasContext.Provider>
  );
};
```

```
return (
  <div className="bg-white rounded-lg shadow-1 p-5 rounded-tl-[90px] w-full max-w-[352px] mx-auto cursor-pointer hover:shadow-2xl transition">
    <div className="w-[18rem] rounded-tl-[90px] rounded-br-[70px] bg-transparent">
      <img
        className="w-[18rem] h-[18rem] object-fill mb-8 rounded-tl-[90px] rounded-br-[70px] bg-transparent"
        src={require(`../images/${images}`)}
        alt="property images"
      />
    </div>
    <div className="flex gap-x-7 mt-6 items-center">
      <div className="bg-green-500 rounded-full px-7 py-2 text-white text-[17px] text-center">
        {propertyType}
      </div>
      <div className="bg-violet-500 rounded-full px-7 py-2 text-white text-[17px]">
        {district}
      </div>
    </div>
    <div className="text-lg mt-3 font-semibold max-w-[260px]">{address}</div>
  </div>
);
```

```
return (
  <>
    <div className="text-center uppercase text-[30px] my-8 text-black font-semibold border-b-2">
      <h3>Near Me Houses To Explore</h3>
    </div>
    <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-5 lg:gap-14">
      {houseData && houseData.length > 0
        ? houseData.map((house, index) => (
          <Link to={`/nearmeproperty/${house._id}`} key={index}>
            <NearMeHouse key={index} nearMeHouseDatas={house} />
          </Link>
        ))
        : "Please Click on Near Me "}
    </div>
  </>
);
```

```
exports.getAllProperties = async (req, res) => {
  try {
    const properties = await Property.find({ bedrooms: 12 });
    console.log(properties);
    res.json(properties);
  } catch (error) {
    res.status(500).send(error);
  }
};

exports.getMainProperties = async (req, res) => {
  try {
    const allProperties = await Property.find();
    console.log(allProperties);
    res.json(allProperties);
  } catch (error) {
    res.status(500).send(error);
  }
};
```