# Tribhuvan University

# Academia International College

**Final Year Project Report**

**On**

**Room Finder System**

**Under the supervision of**

**“Mr. Ananda Adhikari”**

**Submitted by**

**Nafisha Maharjan (T.U. Exam Roll No. 29016/078)**

**Prastab Maharjan (T.U. Exam Roll No. 29022/078)**

**Sresta Sharma (T.U. Exam Roll No. 29032/078)**

**Submitted to**

**Department of Computer Science and Information Technology**

**Academia International College**

**Institute of Science and Technology**

**Tribhuvan University**

**June, 2025**

# Tribhuvan University

# Academia International College

**Final Year Project Report**

**On**

Room Finder System

A final year project submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Information Technology awarded by Tribhuvan University


Submitted by

Nafisha Maharjan (T.U. Symbol No. 29016/078)

Prastab Maharjan (T.U. Symbol No. 29022/078)

Sresta Sharma (T.U. Symbol No. 29032/078)

Submitted to

Department of Computer Science and Information Technology

Academia International College

Institute of Science and Technology

Tribhuvan University


June 2025

# Tribhuvan University

**Institute of Science and Technology**

# Academia International College

**Department of Computer Science and Information Technology**

Email: mail@academiacollege.edu.np


## Supervisor's Recommendation

I hereby recommend that the project work report prepared under my supervision by Ms. Nafisha Maharjan (29016/078), Mr. Prastab Maharjan (29022/078) and Ms. Sresta Sharma (29032/078) entitled "Room Finder System" be accepted as fulfilling in partial requirements for the degree of Bachelors of Science in Computer Science and Information Technology. In my best knowledge, this is an original work in Computer Science and Information Technology.

……..

Mr. Ananda Adhikari

Project Supervisor

Department of Computer Science and Information

Technology Academia International College

Gwarko, Lalitpur

# Tribhuvan University

## Department of Computer Science and Information Technology

# Academia International College

## Certificate of Approval

This is to certify that this project prepared by Ms. Nafisha Maharjan, Mr. Prastab Maharjan and Ms. Sresta Sharma entitled "Room Finder System" in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and quality as for the required degree.

| | |
|---|---|
| ……..<br>Mr. Ananda Adhikari<br>Project Supervisor<br>Department of Computer Science and IT<br>Academia International College | ……..<br>Mr. Bishwas Mathema<br>HOD/Program Coordinator<br>Department of Computer Science and IT<br>Academia International College |
| ……..<br>Internal Examiner<br>Academia International College | ……..<br>External Examiner<br>Central Department of CSIT<br>Tribhuvan University |

# Acknowledgement

We want to sincerely thank Academia International College, Lalitpur, for their invaluable resources and continuous support during the course of our project. Our educational experience has been greatly impacted by the college's constant dedication to academic success, and we sincerely appreciate the support and resources provided.

We would like to express our deepest and sincere thanks to our highly respected supervisor Mr. Ananda Adhikari for his valuable guidance, encouragement and help for supporting the work. His useful suggestion for this work and cooperative behavior are sincerely acknowledged.

We would also like to thank our BSc.CSIT coordinator sir Mr. Bishwas Mathema for supporting and guiding us throughout the project. We are also grateful to our teachers for their constant support and guidance. At the end we would like to express our sincere thanks to all the friends and others who helped us directly or indirectly during this project work.

Regards,

Nafisha Maharjan (29016/078)

Prastab Maharjan (29022/078)

Sresta Sharma (29032/078)

# Abstract

Room Finder System is a Web based application that helps the renters, book rooms and house owners to list available spaces for rent. People need to invest a lot of time and effort to find room in Kathmandu Valley. The Room Finder System helps to solve these problems by connecting the house owner with tenant. The Room Finder system enforced, greatly facilitates the tenant who can book their room according to their requirement from anywhere inside the Kathmandu Valley. This application has been developed using React as frontend tool and Django for backend. We used SQLite as a database. The Room Finder system hence designed will be of great importance towards the tenants and landlords with quick access and functionalities reducing time and money on both sides.

Tenants can find their ideal rooms by using advanced filters and interactive map with the help of OpenStreetMap. To improve search efficiency system, uses TF-IDF with Cosine Similarity and fuzzy matching algorithms, allowing for accurate results even when search term includes typos or incomplete phrases. System also contains features like user review and ratings, as well as admin dashboard for managing room listings and user data.

The proposed solution significantly reduces the time required for searching rooms in Kathmandu Valley. By simplifying the room finding process, it enhances the user experience, improves decision making and provides a cost-effective approach for tenants.

***Keywords: Room Finder system, Web based application, Kathmandu Valley, Available spaces for rent, Book Rooms, Time Saving, TF-IDF, Cosine Similarity, Fuzzy Matching.***

# Table of Contents

Supervisor's Recommendation ..... i

Certificate of Approval ..... ii

Acknowledgement ..... iii

Abstract ..... iv

List of Figures ..... vii

List of Tables ..... viii

List of Abbreviations ..... ix

Chapter 1 : Introduction ..... 1

1.1 Introduction ..... 1

1.2 Problem Statement ..... 2

1.3 Objectives ..... 2

1.4 Scope and Limitations ..... 2

1.4.1 Scope ..... 2

1.4.2 Limitations ..... 3

1.5 Methodology ..... 3

1.6 Report Organization ..... 5

Chapter 2 : Background and Literature Review ..... 6

2.1 Background Study ..... 6

2.2 Literature Review ..... 6

2.2.1 Existing System ..... 7

Chapter 3 : System Analysis ..... 8

3.1 System Analysis ..... 8

3.1.1 Requirement Analysis ..... 8

3.1.2 Feasibility Analysis ..... 11

3.1.3 Analysis Data Modeling using ER Diagram ..... 14

3.1.4 Process Modeling using DFD ..... 15

Chapter 4 : System Design....17
4.1 System Design....17
4.1.1 Architectural-Design....17
4.2 Algorithm Details....18
4.2.1 TF-IDF + Cosine Similarity....18
4.2.2 : Fuzzywuzzy Algorithm with Levenshtein Distance....19
4.2.3 Haversine Formula....20
Chapter 5 : Implementation and Testing....21
5.1 Implementation....21
5.1.1 Tools used....21
5.1.2 Implementation Details of modules....21
5.2 Testing....22
5.2.1 Test Case for Unit Testing....22
5.2.2 Integration Testing....24
5.2.3 System Testing....25
5.3 Result Analysis....26
Chapter 6 : Conclusion and Future Recommendation....27
6.1 Conclusion....27
6.2 Future Recommendation....27
References....29
Appendices....30

# List of Figures

Figure 1.1: Incremental Model ..... 4
Figure 3.1: Use Case Diagram for Admin ..... 9
Figure 3.2: Use Case Diagram for User ..... 10
Figure 3.3 : GANTT CHART ..... 13
Figure 3.4: ER Diagram of Room Finder System ..... 14
Figure 3.5: Level 0 DFD (Context Diagram) of Room Finder System ..... 15
Figure 3.6: Level-1 DFD of Room Finder System ..... 16
Figure 4.1: 3-Tier Architecture ..... 17

# List of Tables

Table 3.1: Project Schedule Table ..... 12
Table 5.1: Test Case for Registration/Login ..... 22
Table 5.2: Test Case for Room and Review Model ..... 23
Table 5.3: Test Case for Integration Testing ..... 24
Table 5.4: Test Case for Responsiveness of System ..... 25
Table 5.5: Test Case for Usability and UX ..... 25

# List of Abbreviations

| | |
|---|---|
| CSS | Cascading Style Sheet |
| HTML | Hypertext Markup Language |
| SQLite | Structured Query Lite |
| TFIDF | Term Frequency Inverse Document Frequency |
| VSCode | Visual Studio Code |

# Chapter 1 : Introduction

## 1.1 Introduction

In today's world, the growth of technology related content and services is unstoppable. This trend not only brings new opportunities but also inspires the creation of businesses that offer entirely fresh and helpful content and services for the community. The housing sector, in particular, is adapting to the challenges of this trend by adopting new strategies. One such strategy involves the development of software for managing rental rooms and flats.

Finding rental rooms in Kathmandu Valley is a complex task for students, working professionals, and families due to the absence of a centralized digital platform. Traditional methods, such as word of mouth, street posters, and social media posts, are fragmented and inefficient. These methods often lead to wasted time and resources, especially for people new to the city. To solve this issue, this project proposes the development of a web-based platform called "Room Finder System" specifically targeted for the Kathmandu Valley. This platform will allow users to search for available rooms based on area names, room types, price ranges, and map locations using geolocation services and efficient search algorithms.

In response to the changing needs of housing management in Kathmandu Valley, the Room Finder System is developed. The system aims to simplify the communication between landlords and tenants by combining advance search algorithms, geospatial visualization and safe, responsive online technology into a single digital platform. In addition to addressing the shortcomings of traditional, distributed room-search methods, this strategy supports current efforts in improving the accessibility, transparency and effectiveness in the rental housing industry.

## 1.2 Problem Statement

There is no any proper dedicated, centralized, and structured platform for discovering rental rooms in Kathmandu Valley. There are various platforms, however they are either not updated and lag a lot, or is not centralized to only rooms or flats. Users currently rely on unreliable, scattered sources such as social media groups, personal networks, or physical visits, which are not only time-consuming, but also inefficient. This problem is increased for students, job seekers, and new residents who are unfamiliar with the geography of the city. As a result, users face difficulties in making informed decisions about rentals due to lack of real-time, location-based, and filterable data.

## 1.3 Objectives

The primary goal of the project is to develop a comprehensive Room Finder System with the following key focuses:

- To design and develop a centralized and responsive web application for room search and discovery within the Kathmandu Valley.
- To integrate an interactive map view using OpenStreetMap via Leaflet.js.
- To implement algorithms for text-based fuzzy search and geospatial distance calculations.

## 1.4 Scope and Limitations

### 1.4.1 Scope

The scope of Room Finder System includes:

- Geographical Coverage: The system is designed especially for the Kathmandu Valley, focusing on the needs of local renters in this area.
- User Friendly interface: A secure mobile responsive platform with the real time map-based listing.
- Search Capabilities: Helps renters to find the room via type, price and distance.
- Basic Security Implementation: The system contains advance security features like two factor authentication.

### 1.4.2 Limitations

Limitations of the Room Finder System are:

- Limited to Kathmandu Valley: The system is geographically limited and cannot be extended outside the Kathmandu Valley without any major changes.
- Internet Dependence: Requires internet for all functionalities since, its web based.
- Map API Limitation: Currently relies in open-source APIs like Nominatim and Leaflet.js which may lead to slow performance or less accurate location data, especially during high traffic period.

## 1.5 Methodology

The Incremental Model is a Software Development approach where the system is built and delivered in small, functional parts (increments). Each increment adds new features or improvements, allowing for continuous refinement based on feedback and testing. We used the Incremental Model to develop the Room Finder System, enabling flexibility and step-by-step progress. Regular supervision and feedback sessions played a crucial role in shaping the system as it evolved.

In this model, the project was divided into multiple increments. The first increment focused on setting up the core backend features, including user authentication, room listing, and review submission. This provided a functional foundation to build upon. In subsequent increments, we introduced more advanced functionalities. We enhanced authentication by switching to JWT for improved security and implemented room creation, editing, and deletion features with appropriate permissions. Later, we integrated a recommendation system based on TF-IDF and cosine similarity, along with advanced filtering options for location, budget, and room type. Each increment was tested and refined based on supervisor feedback to ensure smooth functionality and user-friendliness. The next phase involved frontend development using React, where the system was integrated with dynamic room listings, search capabilities, and user authentication flows. We also designed a visually appealing interface with a responsive layout that reflected real-time data from the backend. In later increments, we prepared for map integration using the Haversine formula, refined the user profile section, and polished the interface to align with our project's goals.

Once all the increments were successfully integrated, we conducted comprehensive testing across both backend and frontend components, resolved any remaining issues, and prepared the project for deployment on GitHub. This incremental approach allowed us to manage complexity effectively, deliver core features early, and continuously enhance the system based on practical feedback.

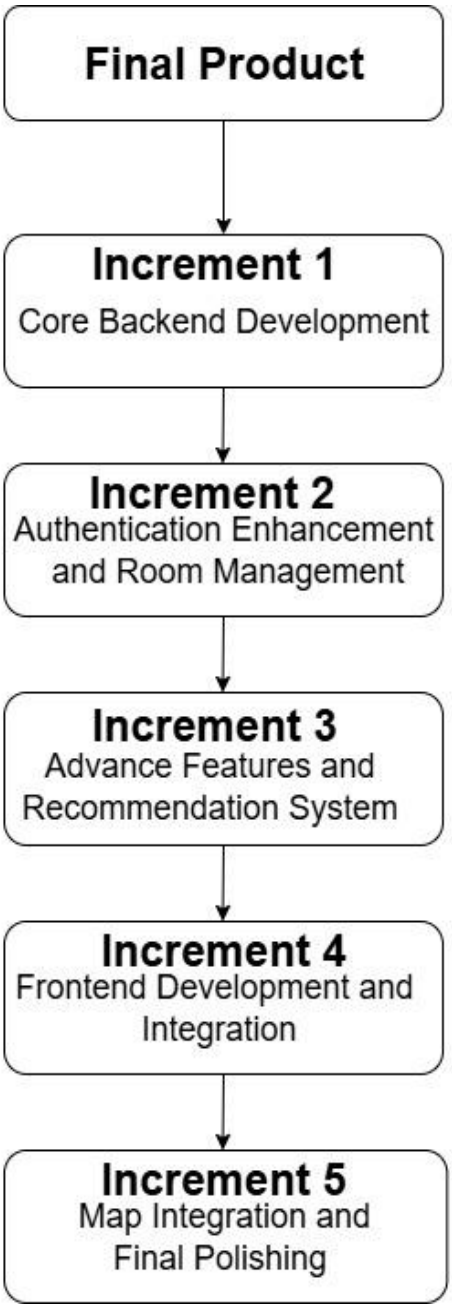


**Figure 1.1: Incremental Model**

## 1.6 Report Organization

This report is divided into six main chapters, each representing specific aspect of project. The chapters are outlined as follows:

Chapter 1: Introduction

The first chapter provides an overview of the project, including its background, the problem being addressed, the objectives, and the scope and limitations. It also outlines the methodology used during the development of the project.

Chapter 2: Background Study and Literature Review

This section is all about the background study and literature review of the existing same system.

Chapter 3: System Analysis

The third chapter focuses on the system development process. It outlines the functional and non-functional requirements and includes the feasibility study.

Chapter 4: System Design

This section illustrates the system with the help of the diagrams. It is all about the design and algorithm details. Here we specify the algorithm that we are going to use and the design we are building.

Chapter 5: Implementation and Testing

This chapter explains how the system was developed using various tools and technologies. It also details the testing process, including unit testing and system testing.

Chapter 6: Conclusion and Future Recommendation

The final chapter summarizes the project's key achievement and findings. It provides the recommendation and suggestion for the future improvements.

# Chapter 2 : Background and Literature Review

## 2.1 Background Study

Conducting a background study for a room finder system in Nepal involves researching existing system, understanding the needs of customer and property owners, analyzing the local industry, and considering the cultural landscape. It enables organization to develop a tailored system meeting the unique needs of the Nepalese market.

The room finder system has witnessed the significant growth, facilitated by online platforms, making it easier for the renters. It is preferred by the locals seeking temporary house due to job transfers, education, especially in Kathmandu Valley where many residents, including students and those in low-paying jobs, find traditional apartment unaffordable. Consequently, room finder has become a popular and affordable alternative. Online platform enables to search room based on preference like location and price.

## 2.2 Literature Review

In Nepal, due to rapid urbanization, population growth, and the absence of the centralized digital platform have made it more difficult to find the suitable rental housing in urban areas like the Kathmandu Valley. The room seekers typically rely on unofficial sources like street posters, word of mouth and unorganized social media posts, especially job seekers, students, and recently moved residents. These approaches frequently lead to frustration and wasted effort as they are not only ineffective but also time consuming.

The system assists the users in finding rooms that matches their preference. One effective method is cosine similarity, which allows system to rank and recommend the rooms that are most relevant to the user's search query. Moreover, text-based algorithm play an essential role in managing the data effectively. The TF-IDF technique transforms textual information into numerical vectors. When combined with cosine similarity, this method provides accurate pairs of search terms to enhance the results, leading more accurate outcome. Additionally, fuzzy matching algorithm with levenshtein distance is implemented to improve search flexibility by addressing spelling errors or incomplete queries [1] [2].

### 2.2.1 Existing System

**Gharbeti .com**

Gharbeti.com is the real state website in Nepal, facilitates the buying selling, and renting properties with a user-friendly platform. Users can search based on price, location, and connect directly with the agents for assistance. With a focus on comprehensive services, Gharbeti.com emerges as a valuable resource for an individuals engaged in property transaction in Nepal [3].

**Daley Bhai**

Daley Bhai is a room rental platform in Nepal, featuring shared rooms, private properties, and apartments. While daleybhai is a useful service, there are concerns about the property details, geographic accuracy, and high price range [4].

**Mero Property**

The platform is user friendly and informative, providing the latest real state listing, market trends, and news. Users can search for properties using location, price and other criteria with detailed information and photos available for each listing. Although it is functional, the platform's overall efficiency is limited because it lacks property information [5].

**Rental Nepal**

Rental Nepal offers a range of properties such as a rooms, apartments, lands, and houses. With user-friendly website, it facilitates easy searches and browsing of various rental option based on user preference [6].

**Magic Bricks**

Magic Brick is user friendly and informative platform in India, facilitates the buying, selling and renting properties with user-friendly platform. User can search for properties using location, price and other criteria with the detailed information and photos available [7].

# Chapter 3 : System Analysis

## 3.1 System Analysis

The Room Finder System is designed to help users choose the most convenient room for their living. Built using Django, the system is powered by the Cosine Similarity and fuzzy algorithm. To do system analysis, we need to know what the Room Finder System needs, what parts it has, and how it works. This includes specifying the users, data flow, system functionality, and technical architecture and understanding the database and relationship among various entities.

### 3.1.1 Requirement Analysis

The system requirement can be functional requirements and non-functional requirements.

#### i. Functional Requirements

The system takes into account the following functional requirements:

- User Registration and Login: The users are able to register through web platform using email and password.
- Admin Panel: Provides admin dashboard for providing functionality to create, edit and delete the room information.
- Room Search Functionality: The users are able to search the room based on the area, room type, price and distance.
- Room Detail View: The system should provide comprehensive information about each room including description, location, pricing, and any other additional services offered.
- Contact Information: The system provides contact information about the room owner for further details.
- Review and Ratings: The users should be able to leave review and ratings for the room they have rented, providing feedback for other potential customers.

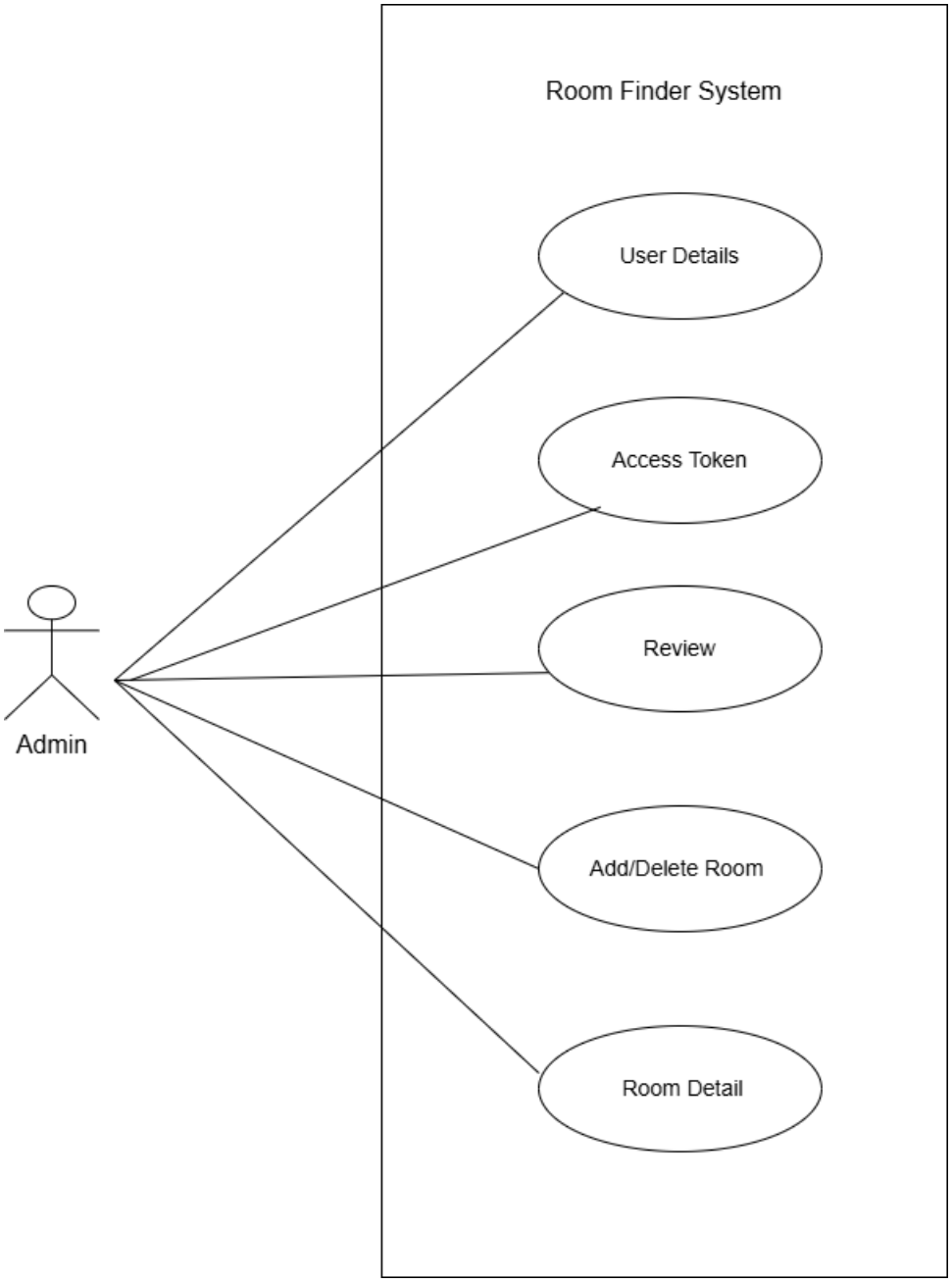


**Figure 3.1: Use Case Diagram for Admin**

The use case diagram indicates interaction between the admin and the system. The admin can view the user details, add or delete the room. They can also view the token which is generated while user is registered.

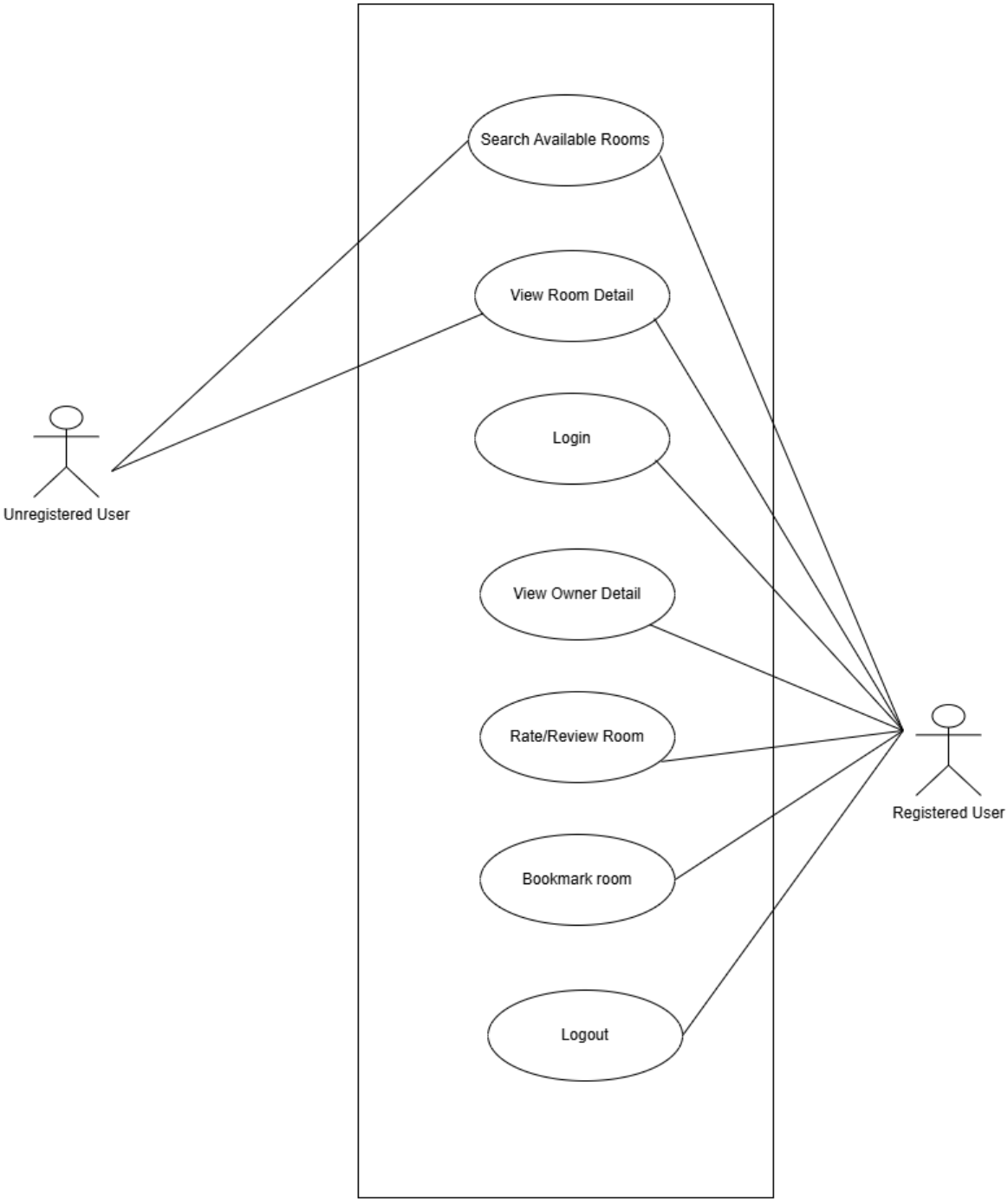


**Figure 3.2: Use Case Diagram for User**

The use case diagram indicates the interaction between the user and the system. The registered user can use all the functions of the system. The process begins with the user registering or logging into the system. Once the user logs into the system, they can search for the relevant room and view the owner details. User can also rate/review the rooms. The unregistered user can only view the room details.

**ii. Non-Functional Requirements**

The system takes into account the following non-functional requirements:

- Performance: The system responds to the user queries and provides recommendations within few seconds.
- Usability: The system is easy to use for both technical and non-technical users.
- Maintainability: The system extends easily. The code is written in a way that it favors implementation of new functionalities.
- Security: The system prevents unauthorized access to the admin panel and listing feature management.
- Responsiveness: All features render correctly on the user devices within few seconds.

### 3.1.2 Feasibility Analysis

Feasibility study helps to determine whether the project is practically achievable within the available resources, time, and technology.

**i. Technical Feasibility**

The project is technically feasible due to open-source technologies requiring minimal hardware and software costs. The technology stack includes Django for backend, React for the frontend, and SQLite as the database. These components can be hosted locally for development and demonstration purposes.

**ii. Operational Feasibility**

The Room Finder system's web based, user-friendly and simple design makes it operationally feasible. It can be accessed from any internet connected device and doesn't require any software to be installed. With features like map views, filters, and a

straightforward admin panel, the platform is easy to use for both the room finder and owner.

### iii. Economic Feasibility

The economic feasibility of a room finder system evaluates its financial viability, analyzing initial and operating costs and cost benefit analysis to determine if the project is economically viable. The entire system was built using open-source technologies, such as Django, SQLite, HTML, JavaScript, Leaflet.js and OpenStreetMap, which are all free and doesn't require any license. It is an effective choice for this project because many websites provide free website hosting as long as storage limit is met.

### iv. Schedule Feasibility

**Table 3.1: Project Schedule Table**

| Task | Start Date | End Date | Duration(days) |
|---|---|---|---|
| Initial Setup and Planning | 5/1/2025 | 5/7/2025 | 6 |
| Models and APIs | 5/8/2025 | 6/3/2025 | 26 |
| Recommendation System and Search | 6/4/2025 | 7/1/2025 | 27 |
| Frontend Recommendation | 7/2/2025 | 7/28/2025 | 26 |
| Review System and User Dashboard | 7/29/2025 | 8/19/2025 | 21 |
| Testing (Pre-final) | 6/4/2025 | 8/25/2025 | 82 |
| Documentation and Final Testing | 5/6/2025 | 8/29/2025 | 111 |

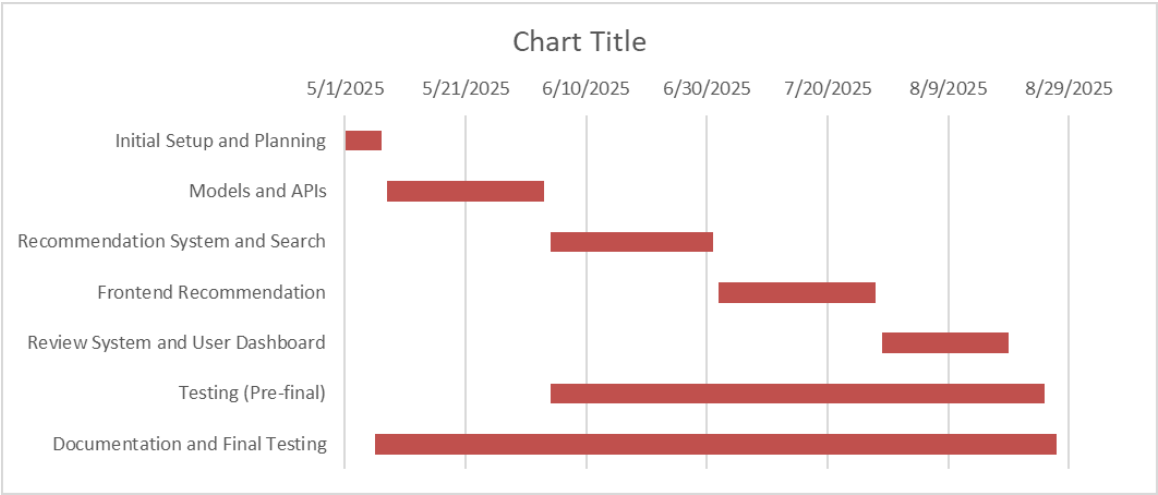


**Figure 3.3 : GANTT CHART**

### 3.1.3 Analysis Data Modeling using ER Diagram

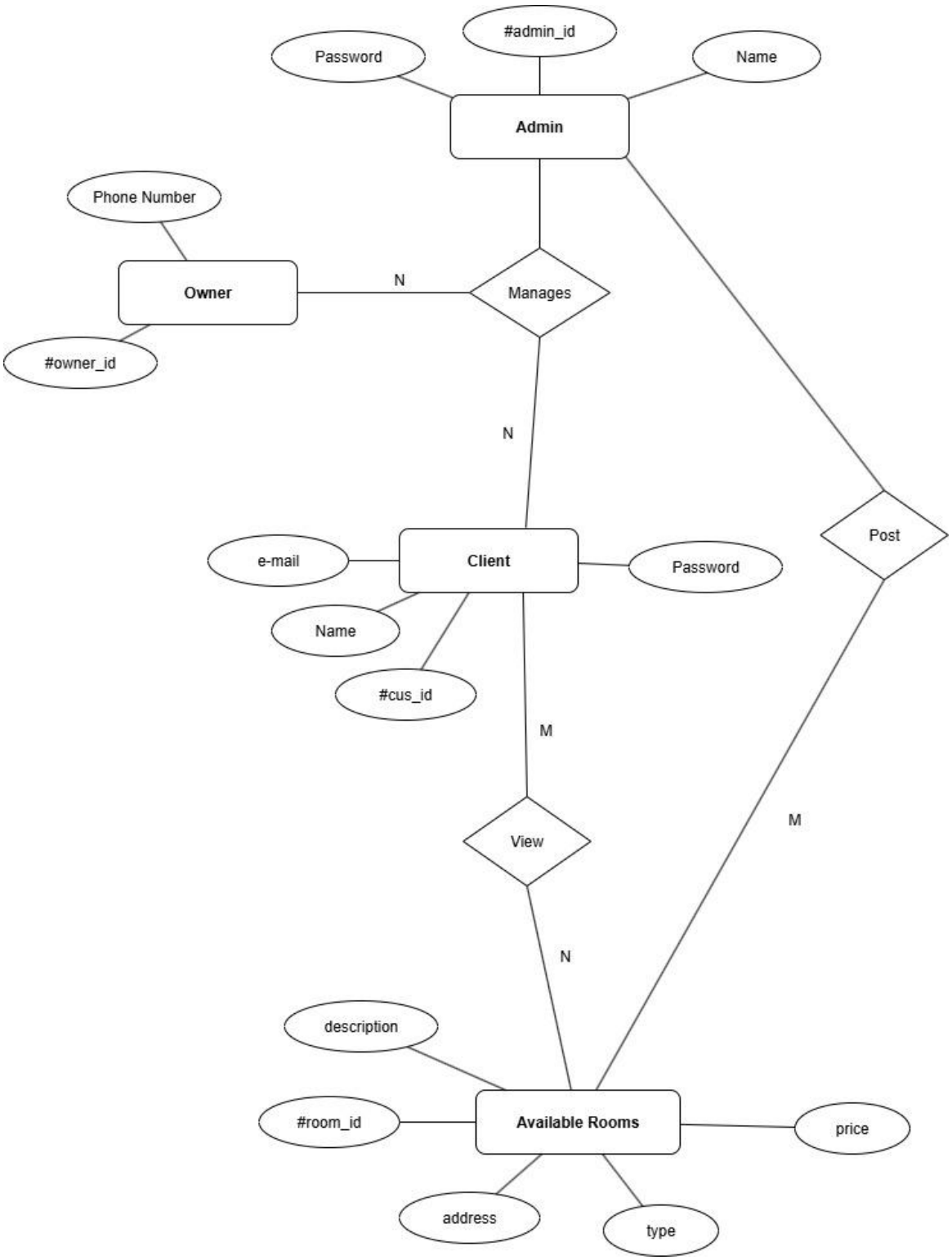


**Figure 3.4: ER Diagram of Room Finder System**

The ER diagram of the Room Finder System illustrates the relationship between the key entities: Admin, Owner, Client, and Available rooms. The admin manages both owner and clients, who are identified by the unique ids and have associated attributes.

### 3.1.4 Process Modeling using DFD

The processes such as input of room details, room recommendation, and flow of result between the user and system are illustrated below in the form of a Data Flow Diagram.

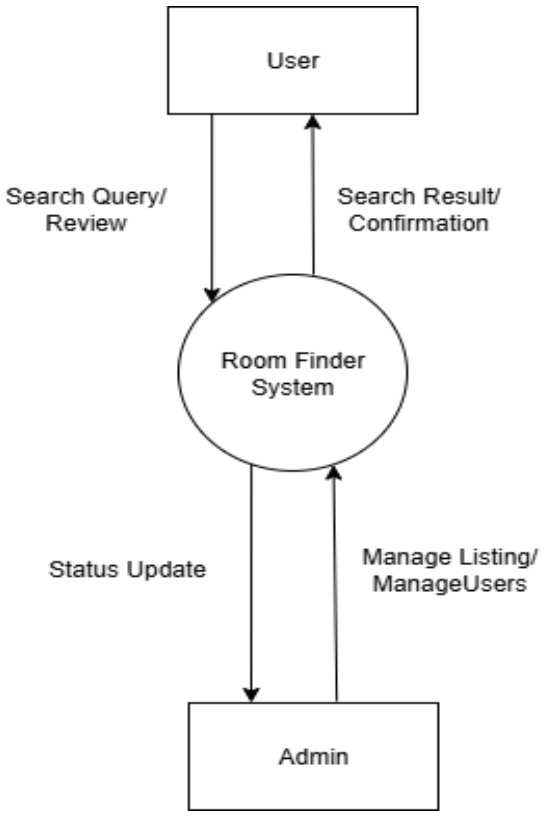


**Figure 3.5: Level 0 DFD (Context Diagram) of Room Finder System**

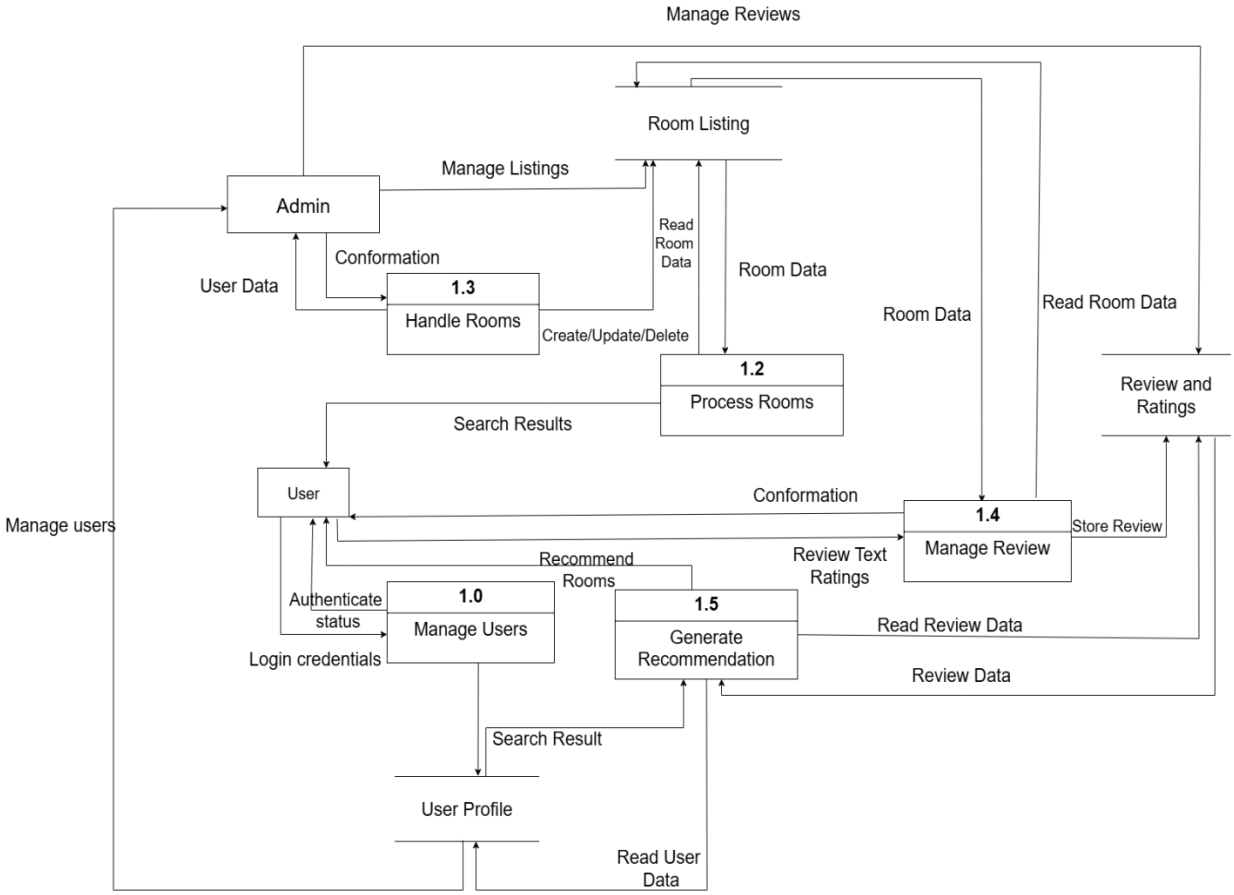


**Figure 3.6: Level-1 DFD of Room Finder System**

The above level-1 DFD of Room Finder System presents a structured view of system's functional components and their interactions with external entities. It illustrates how system manages user registration, processes and updates room listings, administrates user view and generates personalized room recommendations. Data flows between users, admin and the system data.

# Chapter 4 : System Design

## 4.1 System Design

As discussed in the analysis chapter, the system design also follows the structured approach.

### 4.1.1 Architectural-Design

Our system is structured upon 3-tier architecture, dividing it into presentation, application, and data layers. This client-server model ensures independent development and maintenance of process logic, data access, storage, and user-interface, promoting modularity and scalability. The architecture facilitates a clear and organized division of functional responsibilities, enhancing the overall efficiency and maintainability of the system.

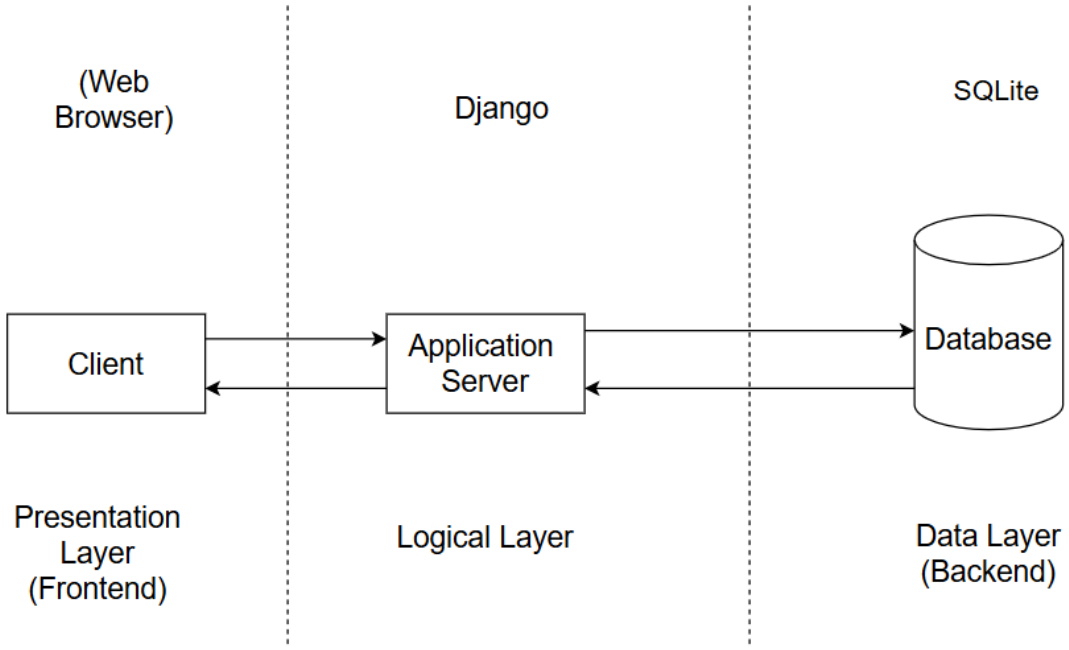


**Figure 4.1: 3-Tier Architecture**

## 4.2 Algorithm Details

### 4.2.1 TF-IDF + Cosine Similarity

TF-IDF (Term Frequency-Inverse Document Frequency) is a statistical algorithm used to evaluate how important a word is to a document in a collection. In the context of Room Finder System, TF-IDF is used to convert a textual data such as room titles, description, and location into numerical vectors that can be compared for similarity.

The Cosine Similarity measures the similarity between the two vectors of an inner product space. It is measured by the cosine of angle between two vectors and determines whether two vectors are roughly pointing in the same direction. Cosine similarity is commonly used in text analysis to measure the similarity between documents based on the frequency of words or phrase they contain.

In Room Finder System, we specially use cosine similarity algorithm for suggesting the similar listings when a room is unavailable. It is used by comparing the title, description, and area of unavailable room with other listings in database. Then the system finds the room that is most semantically similar.

Cosine similarity is calculated as the dot product of two vectors divided by the product of their magnitudes. As an equation, this would look like:

Cosine Similarity =

dot product of vector A and vector B / (magnitude of vector A * magnitude of vector B)

The resulting cosine similarity value ranges from -1(completely dissimilar) to 1(identical). A value of 0 indicates orthogonality (no similarity). In the context of NLP, cosine similarity is commonly used to compare the similarity of document vectors, word embeddings, or any future vectors in a multi-dimensional space, making it a valuable metric for tasks like document retrieval, clustering, and information retrieval [8].

### 4.2.2 : Fuzzywuzzy Algorithm with Levenshtein Distance

Fuzzy matching is a technique used to find similar strings even if they are not exactly the same. It's like finding matches that are close enough, even if they have some differences. This can be useful for searching for similar words, correcting spelling mistakes, or comparing names.

The Levenshtein Distance is a string metric that measures the minimum number of single character edits required to change one string into another. FuzzyWuzzy is essentially a high-level wrapper around the Levenshtein Distance algorithm. It uses Levenshtein in background to measure how similar two strings are, but it also adds smarter ways to handle real-life cases like word in different order, partial matches, or repeated words.

In Room Finder System, FuzzyWuzzy is used to improve the search functionality, especially for matching user input location names with stored data. If a user searches for "Gworko" instead of "Gwarko", the system can still identify the correct area and return relevant room listing. This is particularly useful for handling misspelling and incomplete input.

### 4.2.3 Haversine Formula

The haversine formula determines the great-circle distance between two points on a sphere given their longitudes and latitudes. It is important in navigation. It is a special case of a more general formula in spherical trigonometry, the law of haversines that relates the sides and angles of spherical triangles.

To accurately determine the location between the rooms and the main street of the given location, the haversine formula was embedded into the backend computation logic of the system. The version of haversine formula implemented in our system is as follows:

$\Delta$ lat = lat 2 – lat 1

$\Delta$ long = long2 – long1

$\alpha$ = sin ^ 2 ( $\Delta$ lat / 2) + cos (lat1) . cos (lat2) . sin ($\Delta$ long / 2)

c= 2 . atan2 ($\sqrt{\alpha}$ , $\sqrt{1-\alpha}$ )

d = R * c

where:

- R is the radius of Earth
- Lat1 lat2 are the latitude of the first and second locations
- Long1 long2 are the longitudes of the first and second locations
- d is the resulting distances between the two locations

This formula, which accounts for the Earth's curvature, enables the calculation of great circle distances based on latitude and longitude coordinates [9].

# Chapter 5 : Implementation and Testing

## 5.1 Implementation

Various development technologies and tools have been used for the application development.

### 5.1.1 Tools used

Following are the tools used for the development of our project:

- React js: Frontend development JavaScript Library
- Tailwind CSS for designing and styling
- JWT for session handling and secure API request
- SQLite: It is lightweight, file-based relational database.
- Django for handling backend logic
- Django REST Framework : helps in building restful APIs in Django
- Leaflet.js/OpenStreet map for displaying room location in map.
- Git: A distributed version control system for tracking changes in source code during software development.
- Github: A web-based platform for version control and collaborative development using Git.
- Draw.io: To create diagrams.
- Vscode for writing and running the code

### 5.1.2 Implementation Details of modules

Some of the module included in the project is:

- Admin Module: The admin module is the default module of the room finder system which is provided in default by Django.
- Userauth Module: The userauth module is responsible for handling all user authentication related functionality in room finder system. It includes feature like user login, signup, and authentication using JSON Web tokens (JWT).
- Rooms Module: This module is responsible for managing all the operations related to the rooms. Here rooms can be added, deleted and updated by the owner. It allows users to view available rooms. It also includes review system where users can submit feedback and rate the rooms.

## 5.2 Testing

The following steps are taken during the testing phase.

- Unit Testing: Unit testing is done in backend to verify the working of APIs.
- Integration Testing: Integration testing is performed manually to verify the frontend, backend and database worked seamlessly with each other.
- System Testing: In system testing, the entire system is tested from end-to-end to identify the potential errors and bugs.

### 5.2.1 Test Case for Unit Testing

Unit Testing ensures that each module or component perform its functionality correctly.

**Table 5.1: Test Case for Registration/Login**

| Test Case Id | Test Case Description | Steps Performed | Expected Outcome | Actual Outcome | Status |
|---|---|---|---|---|---|
| UT-01 | Test user registration | 1. Open register page.<br>2. Enter valid details.<br>3. Submit the form. | User is registered and redirected to the login page. | User is registered successfully and redirected to the login page. | Pass |
| UT-02 | Test login | 1. Open login page.<br>2. Enter valid credentials.<br>3. Submit. | User is logged in and redirected to the homepage. | User is logged in and redirected to the homepage. | Pass |
| UT-03 | Test invalid login attempt | 1. Enter wrong credentials.<br>2.Submit | Error message is displayed: "Invalid email | Error message is displayed: "Invalid email or password." | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | “Invalid email or password” | or password." | | |
| UT-04 | Test JWT token storage | 1.Login successfully.<br>2.Check local storage. | Access token is stored in the browser for use in API. | Access token is stored in browser. | Pass |

**Table 5.2: Test Case for Room and Review Model**

| **Test Case Id** | **Test Case Description** | **Steps Performed** | **Expected Outcome** | **Actual Outcome** | **Status** |
|---|---|---|---|---|---|
| UT-05 | Create a new room | 1. Login.<br>2.Goto “Add Room” page.<br>3.Submit Details | Room is created and listed. | Room is created and listed. | Pass |
| UT-06 | Submit Review | 1.Login.<br>2.Visit a room page.<br>3.Leave review and rating. | Review is added below the star rating. | Review is added below the star rating. | Pass |
| UT-07 | Prevent Duplicate Review | 1.Submit a second review for same room. | Error: “you have already reviewed this room.” | Error displayed. | Pass |
| UT-08 | Fetch recommendat | 1.Visit a room page. | Recommendation is displayed. | Recommendations are displayed. | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | ion | 2.Scroll to "Similar Rooms You Might Like." | | | |

### 5.2.2 Integration Testing

**Table 5.3: Test Case for Integration Testing**

| Test Case Id | Test Case Description | Steps Performed | Expected Outcome | Actual Outcome | Status |
|---|---|---|---|---|---|
| IT-01 | User Registration +JWT Authentication | 1. Register a new account.<br>2. Log in with the registered credentials.<br>3.Browse and leave the review. | JWT token works and user remains authenticated. | Token is valid upon request. | Pass |
| IT-02 | Search + Filter + Recommendation | 1. Use Search bar on homepage. | Rooms are filtered and recommendation matches the query. | Bookmarked rooms appear in the list. | Pass |
| IT-03 | Bookmark toggle | 1.Bookmark a room.<br>2.Vist bookmark | Bookmarked rooms are displayed. | Bookmarked rooms appear in the list. | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | | section. | | | |

### 5.2.3 System Testing

**Table 5.4: Test Case for Responsiveness of System**

| Test Case Id | Test Case Description | Steps Performed | Expected Outcome | Actual Outcome | Status |
|---|---|---|---|---|---|
| ST-01 | Test on desktop | 1. Open all the pages on 1220px screen. | Layout renders cleanly. | Works well on desktop. | Pass |
| ST-02 | Test on tablet | 1. Open the application on tablet browser use chrome DevTools. | Pages are displayed correctly without layout issues. | Pages are displayed correctly without layout issues. | Pass |
| ST-03 | Test on mobile | 1.Open application on a phone/DevTools. | All components are scrollable and readable. | Looks clean and scrolls correctly. | Pass |

**Table 5.5: Test Case for Usability and UX**

| Test Case Id | Test Case Description | Steps Performed | Expected Outcome | Actual Outcome | Status |
|---|---|---|---|---|---|
| ST-04 | Test navigation | 1. Use navbar links. 2. Test routing. 3.Test backward/forward. | Pages load correctly and stay consistent. | Navigation between home, detail, login works fine. | Pass |

| | | | | | |
|---|---|---|---|---|---|
| ST-05 | Test form validation | 1. Leave form blank.<br>2. Observe error messages. | Appropriate error messages are displayed. | Appropriate error messages are displayed. | Pass |
| ST-06 | Test user feedback | 1. Perform actions like login, logout, bookmark, review. | Feedback messages are displayed for all actions. | Feedback messages are displayed for all actions. | Pass |

## 5.3 Result Analysis

The Room Finder System was found to be functional, user-friendly during testing. Key observation includes:

- Accuracy: The cosine-similarity based recommendation returned relevant room suggestions based on the title and description.
- User Experience: The interface was responsive, easy to navigate and visually appealing across the devices.
- Performance: The room details and recommendation loads within 1-2 seconds, even during consecutive routing.
- Validation: The form validation, login tracking and feedback messages were performed consistently.
- Feedback: User provided the positive feedback on the ability to leave reviews and bookmark functionality.

# Chapter 6 : Conclusion and Future Recommendation

## 6.1 Conclusion

In conclusion, the designed room finder system provides a centralized, user-friendly, and effective platform for room discovery and listing management, filling the major gap in rental housing landscape in Kathmandu Valley. The traditional room finding system like word-of-mouth, social media and posters are unreliable. This system provides the integration of search filters, geological features, and recommendation algorithms into a secure and responsive web application.

The system is built using robust, open-source technologies, including Django for the backend, React for the frontend, and SQLite as the development database. Security and session management are handled via JWT-based authentication, ensuring that only authorized users can post or manage room listings. Additional features such as room reviews, rating mechanisms, and TF-IDF + Cosine Similarity-based recommendations significantly enhance the user experience by enabling informed decision-making and intelligent suggestions. However, it is crucial to acknowledge the system's limitation, including dependence on accurate information and the necessity for a stable online infrastructure.

## 6.2 Future Recommendation

As the room finder system matures, there are strategic areas for improvement and expansion to ensure its continued effectiveness and alignment with evolving user needs in the market. There are some areas for improvement in non-functional requirements as well as functional requirements to enhance better user experience.

There are some areas on which it can be done better.

- Real Time Messaging: To enhance user experience and communication between the renters and property owner real time messaging feature is recommended which allows user to engage in directly one-to-one conversation within the platform.
- Multilingual Support: Provide the local language options, such as Nepali, to increase accessibility for local Nepalese who don't understand English.

- Payment and Booking: We can enable the online booking or reservation system for rooms and add the payment integration system to support advance payment or deposit.

# References


[1] Ahmed, Alqurashi and Aljohani, "Hybrid recommender system for rental property search using collaborative filtering and content-based similarity," *Journal of Intelligent Information Systems,* vol. 62(3), pp. 451-467, 2024.

[2] Calsamiglia, Dupraz and Pardo, "Ranking algorithms and congestion on two-sided housing platforms," arXiv.org, arXiv, 2023.

[3] "Real Estate & Housing in Nepal - Ghar & Jagga Rental Solution," [Online]. Available: https://gharbheti.com/. [Accessed 30 April 2025].

[4] [Online]. Available: https://daleybhai.com/. [Accessed 30 April 2025].

[5] [Online]. Available: https://www.meroproperty.com/. [Accessed 1 May 2025].

[6] [Online]. Available: https://www.rentalnepal.com/. [Accessed 1 May 2025].

[7] "Real Estate | Property in India | Buy/Sale/Rent Properties," [Online]. Available: https://www.magicbricks.com/. [Accessed 30 April 2025].

[8] R. S. Srivastava and G. , "Prime video insightful recommender: Unraveling patterns with TF-IDF and cosine similarity," *Journal of Information & Optimization Sciences,* vol. 46, no. 3, pp. 793-802, 2025.

[9] A.-T. Amda, Safwandi and Z. Fitri, "Boarding House Search in Lhokseumawe and North Aceh Using Haversine Formula with Geographic Information System," *JIMU: Jurnal Ilmiah Multidisipliner,* vol. 3, no. 4, pp. 2375-2385, 2025.

# Appendices

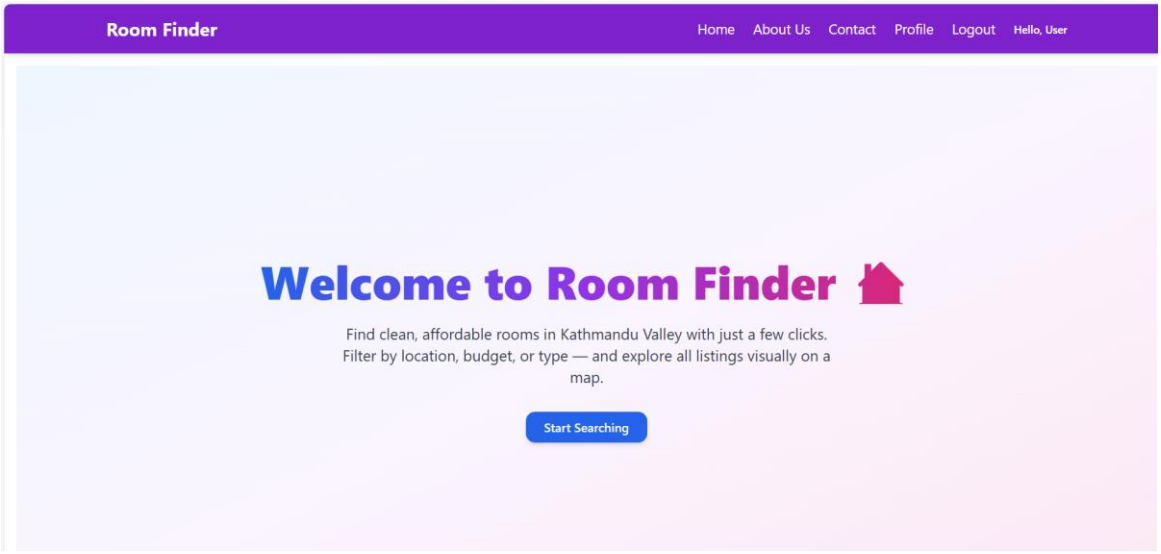


**Figure 1: Landing Page**

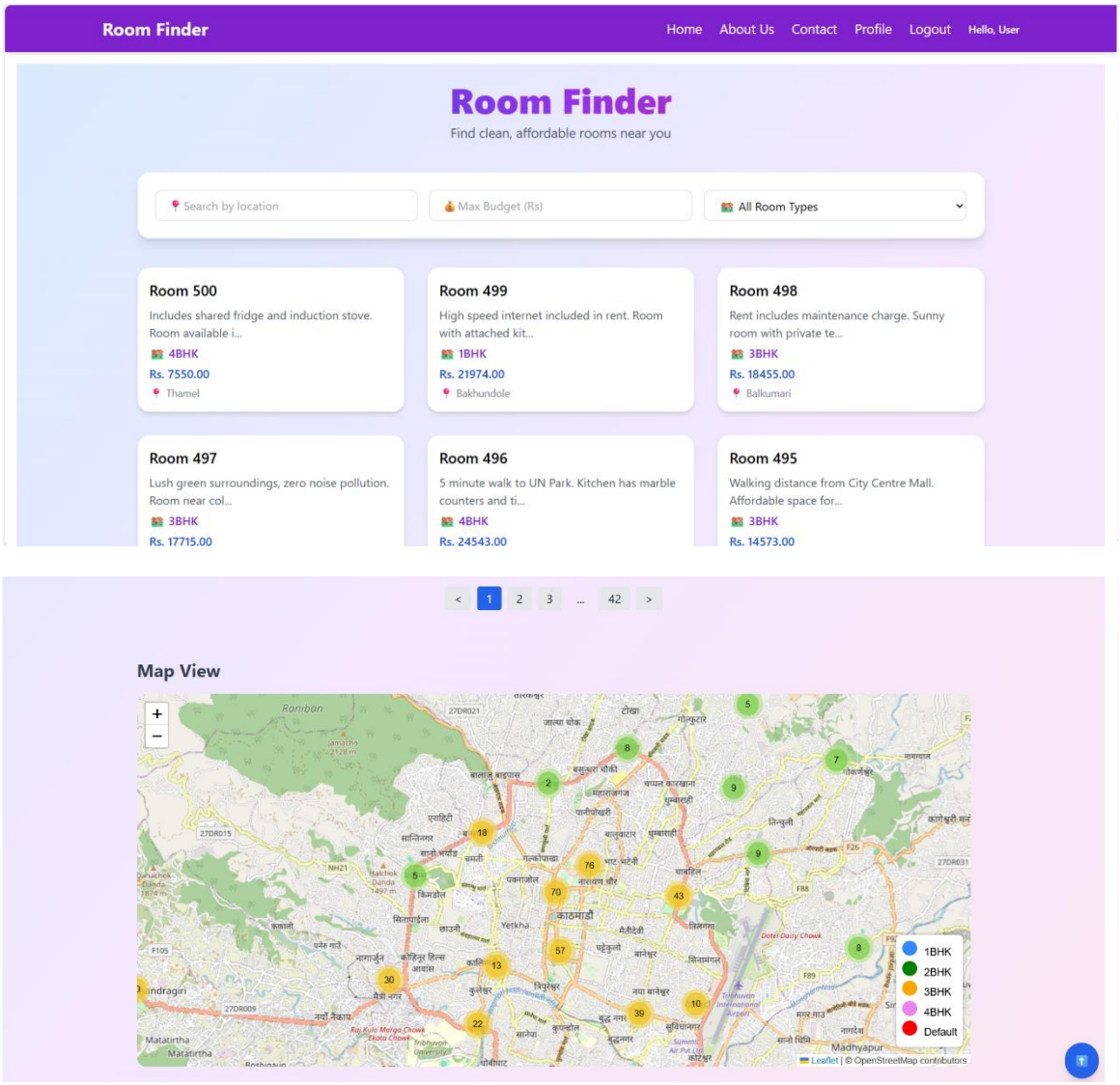


**Figure 2: Home Page**


**Figure 3: About Page**

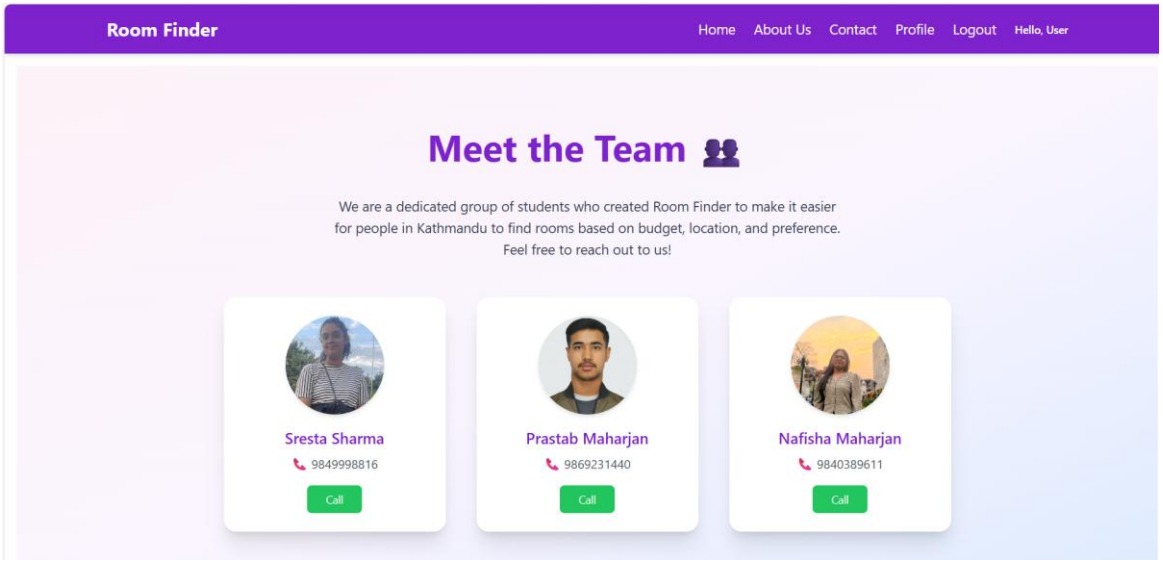


**Figure 4: Contact Page**

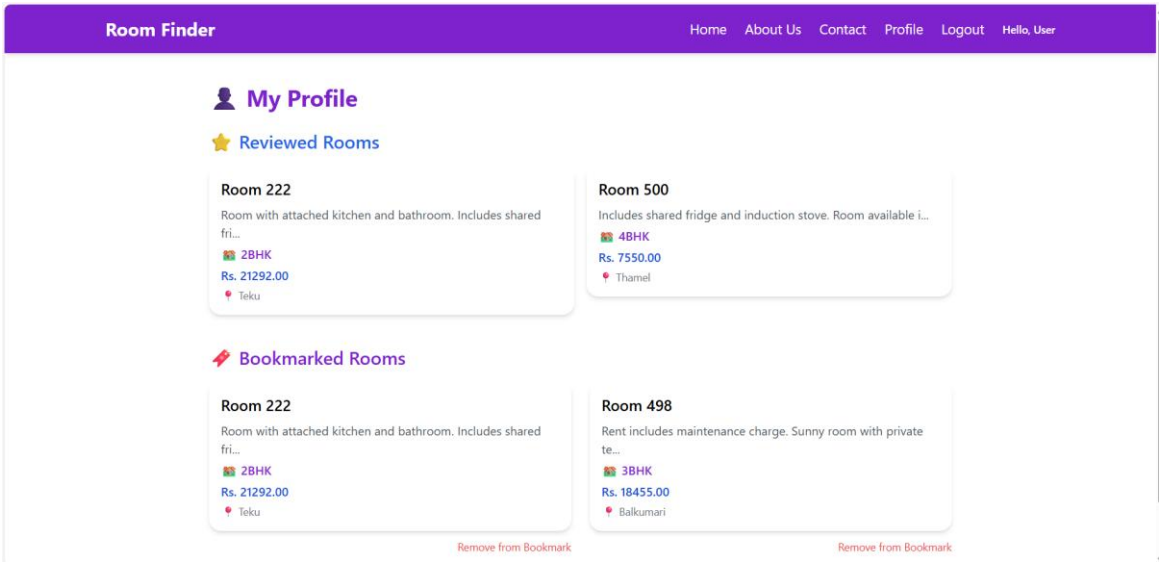


**Figure 5: Profile Page**

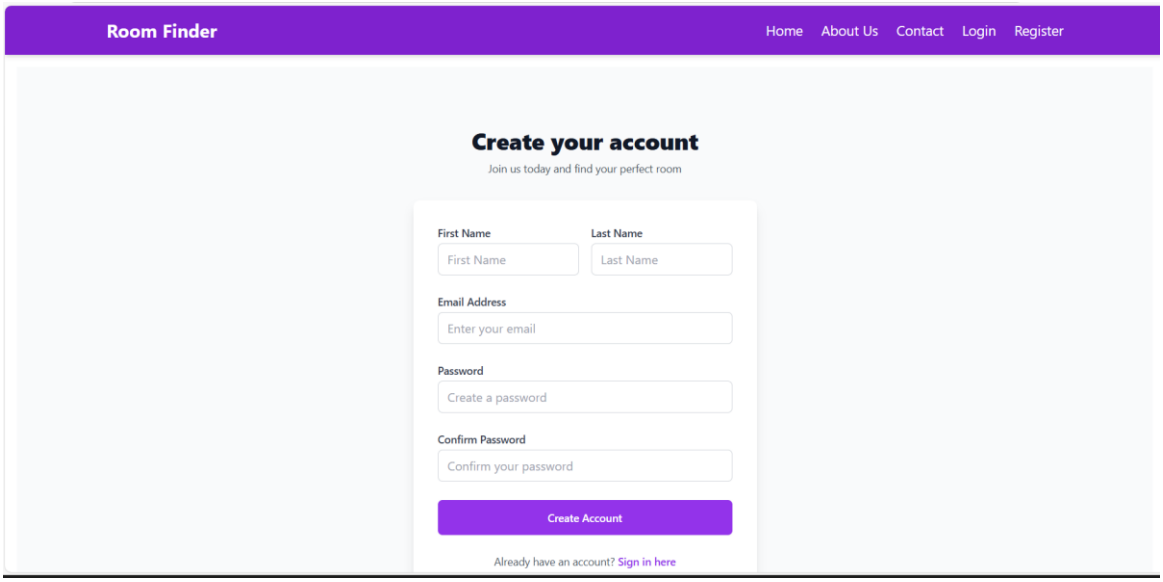


**Figure 6: Register Page**

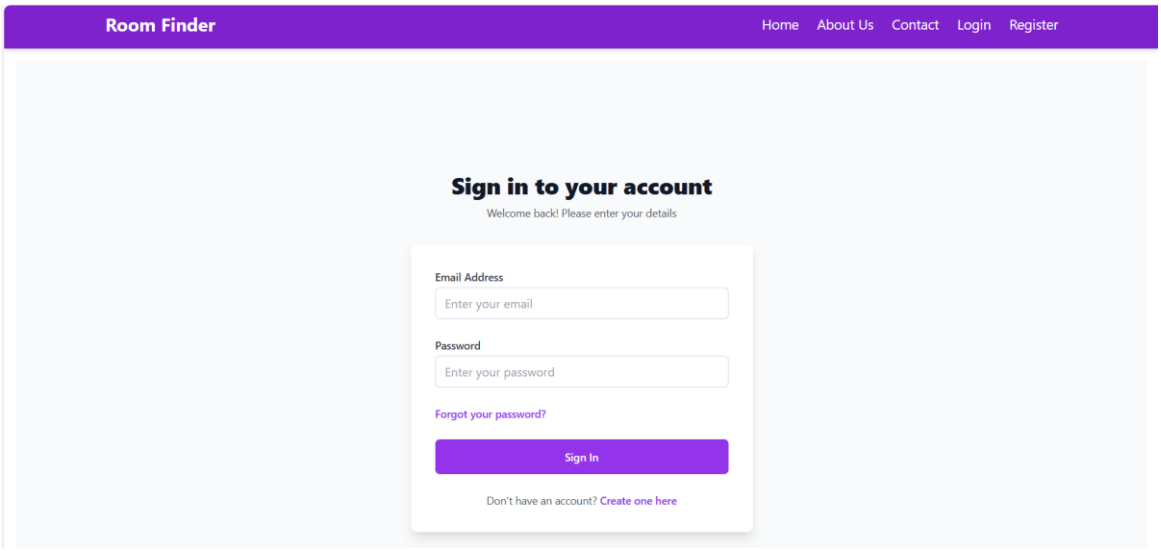


**Figure 7: Login Page**

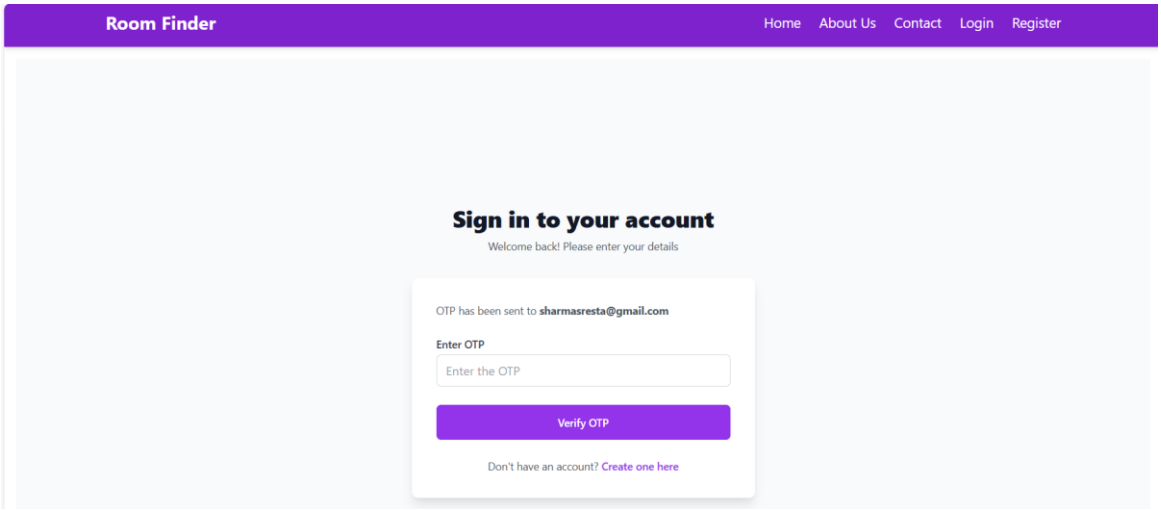


**Figure 8: OTP Verification**

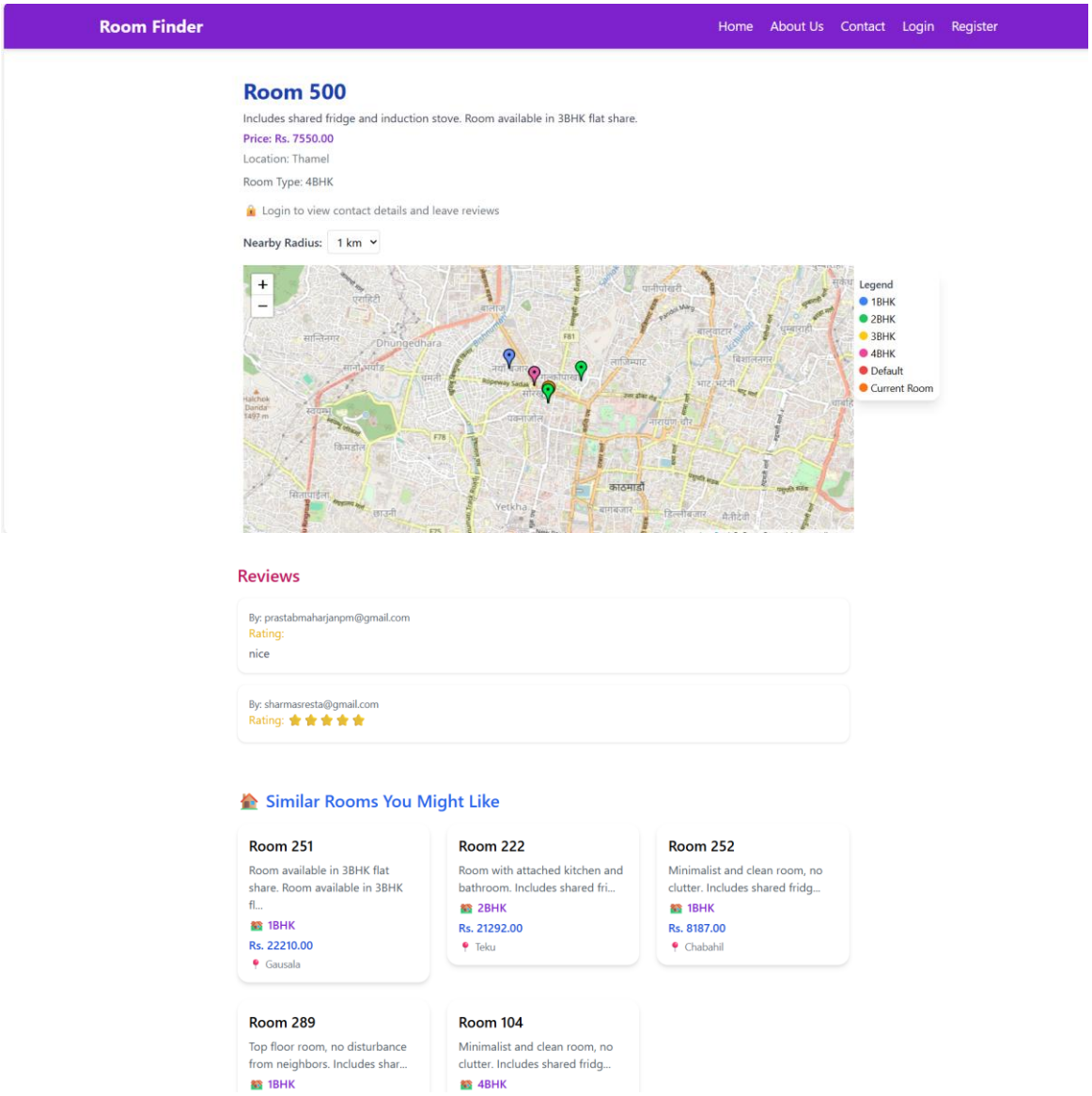


**Figure 9: Room Detail Page**