Group Project Report

ITE 2952 - Programming Group Project

University of Moratuwa

Blue Haven Rentals

(Boarding House Finder Website)

Prepared by

CodeCove

Kriyanjala W – E2340426

Hatharasinghe AIN – E2340065

Jayangani WMGS – E2340450

Abstract

The Blue Heaven Rentals online application is created to empower the increasing demands on the university students, employees, and job seekers to locate credible and inexpensive boarding homes in Sri Lanka. In the past, people used informal channels, like social media, social conversation, or the event boards that tended to give them an obsolete, inaccurate, or fragmented information. Due to the necessity to create a centralized and reliable platform, Blue Heaven Rentals was created to simplify the search mechanism, posting and control of the online listing of boarding houses by creating a user-friendly and interactive online platform.

With the system, the boarding seekers will be able to search the location, price range and facilities of accommodations and boarding house owners will be able to pool, post and manage their listings effectively. React and JavaScript were used in creating the front end of the application to allow it to be responsive, interactive, and compatible across devices. Firebase was used to power the backend services and Firestores was used to operate real-time database, Firebase Authentication to provide secure user. Altogether, these technologies offer flawless synchronization, high-security, and ability to be expanded in the future.

Unit and black-box testing procedures were performed in order to ensure the system was accurate, reliable and performing well. The findings established that Blue Heaven Rentals is successful in achieving its goals, due to the creation of a contemporary, efficient and secure platform to both strikers and proprietors. Overall, this is a key project that aims to make service finding (boarding house) in Sri Lanka far easier and more convenient in terms of integrating technology and comfort as well as providing the basis towards further innovations (mobile applications, online payments and recommendations, based on an artificial intelligence utilization).

Table of Content

List of Table

List of Figure

Chapter 01 – Introduction

1.1. Overview

Sometimes we find that in our present times of the ever-growing mobility of students, movement of the workforce is a basic need to people moving to study or work. In Sri Lanka, college students and the newly famed employees tend to look to uninhabited areas in the large cities in search of boarding houses. However, the familiar accommodation search techniques of word-of-mouth, display boards, or unauthenticated online searches are less likely to provide quality service, are time consuming and are not transparent enough.

Blue Heaven Rentals has emerged to solve these problems and hence has created a centralized digital service along which boarding seekers and boarding house owners will be matched. The system allows the owners to post their properties with the complete details with properties, rent, location and other equipment and searchers can use the system and filter the listings on their terms. The platform is offering an administrator position as well, to check profiles and guarantee the reliability of data to reduce the possibility of misguided ads and fraud.

Fundamentally, the purpose of this project is to substitute the already archaic, semi-traditional manual procedure with a new, validated, and effective eLearning approach. Not only does it help economy of efforts and time, but also builds confidence between boarding seekers and owners. The system is therefore instrumental in ensuring that the accommodation search process is more transparent, structured and affordable to all users in Sri Lanka.

1.2. Background and Motivation

Background

The concept of accommodation is an essential requirement to all the students, employees and job seekers who move to new cities or towns in pursuit of education or job opportunities. The informal and decentralized means of locating appropriate boarding houses have long been used in Sri Lanka through the newspaper, word-of-mouth, notice boards, or unverified social media groups in a decentralized way. These approaches are disjointed, haphazard and unreliable.

Misleading information, expired listing, and lack of comprehensive description of the facilities and pricing of available accommodations are some of the challenges faced by boarding seekers. Conversely, the owners of the boarding houses are usually unable to access a large audience and only do local advertisements. This imbalance between demand and supply points out the necessity of a centralized and reliable digital platform that will be used to match seekers and boarding providers.

Motivation

The reason behind this project is due to the fact that students and employees are actually facing challenges of not finding safe, affordable and convenient boarding houses. The existing practices may require a lot of time, funds, and even bodily effort with the seekers being compelled to traverse various places with no assurance of success. Moreover, the number of fraudulent or unfinished advertisements on the Internet is widespread in the social media, which causes frustration and mistrust in the users.

In boarding house owners, the lack of a centralized system means that they are not able to market the properties effectively. The owners are not able to access potential tenants outside of their immediate area, resulting in empty properties even when they are needed.

Thus, the objective of this project is to overcome these challenges with the help of a centralized web-based solution. The system will not only save the seekers time and effort that would otherwise be spent trying to find a boarding house as it will offer a secure, scalable, and easy to use boarding house finder site but will also give the owners a good avenue to manage and advertise their properties. Finally, the project is supposed to develop a dependable online connection between the seekers and providers to decrease inefficiencies and build trust.

1.3. Aim and Objectives

Aim

This project is intended to design and come up with a web based boarding house finding system which will simplify process of searching, advertising and managing boarding house hence minimizing inefficiencies and increase access by students, employees and owners.

Objectives

To achieve the above aim, the project focuses on the following objectives.

1. To investigate the current issues and shortcomings of the current procedures of locating and marketing boarding houses.
2. To create and build an easy-to-use, mobile web tool to serve the needs of a wide range of devices.
3. To introduce search and filtering feature which gives seekers the opportunity to narrow the results of boarding houses according to their specifications like the region, prices, and amenities.
4. To accommodate layer elements giving boarding house owners a dashboard to add, edit and delete their listing.
5. To make its system secure by means of user authentication, role based access control and encrypted communications.
6. To spend less time, resources, and money on seekers with procured information in a single platform.

1.4. Summary

The following chapter described the Blue Heaven Rentals project, including its context and background along with its impetus. The discussed issues with current manual and online methods of finding accommodation highlighted the inadequacy of the current methods and the necessity of a more systematic nature of the available means.

With the goal and the precise data, the project establishes its mission of digitizing the boarding-house search process, incorporating cloud computing, real-time databases, and responsive web technologies. The system is not only a convenient tool but also a reliable platform which will help all students, employees and property owners.

Blue Heaven Rentals in short is a new project aimed at making an outmoded, yet socially important, process smarter, scalable and more user-friendly by releasing it online - to establish the modernized accommodation management in Sri Lanka.

Chapter 02 – Related Research

2.1. Introduction

Prior to the creation of Blue Heaven Rentals, it was important to examine solutions that exist and how the clients are finding boarding houses at present. This analysis forms the basis of finding the functional deficiencies, user dissonance and improvement opportunities. The senescence has been checked on both the manual platforms; as well as the digital platforms to analyze the strengths and techniques involved.

This chapter identifies the existing systems all the traditional offline methods to the most well-known international online ways of booking, including **Facebook Marketplace, Ikman.lk, Room.lk, and booking.com,** and explains how each system facilitates or falls short to the needs of the Sri Lankan boarding seekers and property owners. The comparison gives feedback on the reason behind the need to set Blue Heaven Rentals and makes the argument as to how the service differentiates itself in terms of verification, reliability, and usability.

Also, the associated work analysis facilitates the conditions of ensuring that the suggested system will not be merely duplicative of already existing ideas but will be based on them to provide increased functionality. This review provides a clear rationale of choices in the design made in Blue Heaven Rentals by reiteratively listing the flaws including absence of authentication, unconfirmed listing and ineffective local targeting. Therefore the chapter acts as a literature review and a comparative analysis upon which the development strategy of the project was built.

2.2. Manual System

Individuals in the manual system normally locate boarding houses with friends or members of their family, on a bulletin board of the universities or in a written advert in the newspapers or via local agents. These are very easy and inexpensive methods, but severe limitations exist with them.

First, when seeking through the manual search the searchers **physically use to spend time** and energy because frequently they wander through multiple places before discovering the right place. Second, **no sufficient verification** of the information is done, including prices, facilities, and even availability can be incorrect or dated. Third, listing, and the tracking of listings are **not stored centrally in a database**, so the data are not organized and become always loose.

There are also some challenges faced in the manual system by the boarding house owners. They have to rely on personal contacts or words and this may take weeks or months to get tenants. It is not a structured manner of popularizing listings and multiple inquiries.

In general, the paper-based system proves to be ineffective, unclear, and does not correspond to a modern user who wishes to get quick and digital access to credible information fast. Blue Heaven Rentals can directly cope with these shortcomings by providing a **verified, cloud-based, and real-time platform** that does not require the inconvenience of a physical search.

2.3. Similar System 01 – Facebook Marketplace

One of the most popular online models of purchasing, selling, renting things such as boarding houses is Facebook Marketplace. Although it enables property owners to access a large number of potential customers in a short time, it was **not developed with the accommodation management** in mind. Advertisements are combined with irrelevant sections like furniture or car, and seekers cannot filter out relevant ones quite easily.

The second issue is **the shortage of verification**. Listings may be left unverified by anyone and this predisposes the element of **fraudulent or false advertisements**. Also, the location-related search filters do not apply to the universities or places of work. The direct message is the system of communication between the seekers and the owners, and it is not screened concerning security and reliability.

Blue Heaven Rentals on the other hand concentrates on proven listing in boarding. It offers easy to follow types, filters and administrative validation to create authenticity, resolving the trust and usability problems of Facebook Marketplace.

2.4. Similar System 02 – Ikman.lk

Ikman.lk is a popular Sri Lankan classified site which contains information about property and rentals. It offers geographic search features and price filter, and consequently users are able to browse accommodations rather easily. Nevertheless, it does not specifically concentrate on the interests of the boarding house seekers like students, or those seeking employment but rather on **general real estate markets.**

Furthermore, Ikman.lk does **not update in real-time**, with taken or unfree boarding houses continuing to display, wasting time of the users. It is also **lacking an integrated communication system**, and users need other ways to communicate with each other externally, such as phone calls or SMS.

Blue Heaven Rentals advances this to a greater level by firebase **updating demands in Firebase Firestore.** Once the listing is changed or deleted, the new information appears immediately to all the users. Also, there is control of communication via authenticated user accounts, improving security and user experience.

2.5. Similar System 03 – Room.lk

Room.lk is a domestic platform that deals with temporary and permanent rentals of accommodation in Sri Lanka. It enables people to search among rooms, annexes, and houses by either district or price. Nevertheless, the design and structure of the site is old-fashioned and functions limitedly.

No classification of listings in terms of target group (e.g. students or employees), and in majority the listings do not include detailed descriptions or proper mapping of their location. It is further characterized by the lack of user registration and authentication where any individual will be able to post without being verified. This makes it less reliable and may end up spamming or listing the same things more than once.

Blue Heaven Rentals offers a friendly, modern interface by comparison and identifies users with Firebase. This is to guarantee that both seekers and owners have verified accounts making the environment safer and controlled.

2.6. Similar System 04 – Booking.com

Booking.com is a global platform that is popular in making hotel and accommodation reservations. It offers superior amenities like tracking of availability, online payment and reviews. Nevertheless, it aims at business accommodation, but not individual or student lodgings.

The lengthy and commission payment based system and the expensive nature of the platform also make it uneconomical to local boarding owners who usually offer low-cost rentals. Also, the majority of users have to undergo formal registration to the position of property managers, which may dishearten the small scale owners.

The Blue Heaven Rentals addresses this gap by making the registration of the many locals free and straightforward so that they do not have to contend with complicated registration demands and expenses. It gives the characteristics of local classified sites the confidence and the structure of professional websites.

2.7. Similar System 05 – Bodima.lk

Bodima.lk is a Sri Lankan site which focuses on renting rooms and boarding. It enables users to either place or view adverts in boarding house, annex and rooms anywhere. The mission of the site is to relate the owner of property with those who may require a pocket friendly accommodation, especially the students and working people.

Bodima.lk represents a simple but effective presentation, with every listing containing rent and photos, together with a description of the facilities (furnishing, the number of rooms, and availability of a parking place). The ease of the system and the fact that people do not need to have much technical understanding to make postings and navigate the site make it simple to use. Nevertheless, the platform does not have any advanced features like user authentication check, Admin authentication check or real-time updates. Structured user role separation is also absent, that is, there is no administrative control between seekers and owners who may use the same interface.

Blue Heaven Rentals builds on the ideas of Bodima.lk by including secure Firebase authentication, real time Firestore database syncing, and admin verification system. It also offers higher level of filters of search, role-using dashboard, and mobile/ desktop friendly construction. In resolving the flaws of Bodima.lk, Blue Heaven Rentals instills a more effective, reliable, and expansive boarding system management in Sri Lanka.

2.8. Comparison Table

Features	Manual System	Facebook Marketplace	Ikman.lk	Room.lk	Booking.com	Bodima.lk	Blue Heaven Rentals
Verified listings	×	×	×	×	√	×	√
Real-time Updates	×	×	×	×	√	×	√
Admin Monitoring	×	×	×	×	√	×	√
User role separation	×	×	×	×	√	×	√
Advanced Search Filters	×	√	√	×	√	√	√
Security & Authentication	×	×	×	×	√	×	√
Platform Focus	√	×	×	×	×	√	√

Table 2.8-0-1 Comparison Table

2.9. Summary

This chapter has examined manual and online systems that are presently being utilized in finding and managing boarding accommodations. It discovered the principal flaws of the current approaches such as unverified information, numerous data are so old, searching is restricted, and roles cannot be managed easily. Marketplace platforms like Facebook and ikmann.lk are highly visible (but not trustworthy). Although Bodima.lk is localized, it does not have the structural and security measures that come with a dependable platform.

Blue Heaven Rentals was created as an eradication of these shortcomings. It takes the simplicity of the local systems and integrates the technological advancement of the contemporary web platforms. It provides a secure and verified authentication system, live updates, and administrative controls so that the boarding seekers and owners can engage in a transparent and reliable setup. This platform forms a new, locally applicable, and scalable platform to the Sri Lankan accommodations market.

Chapter 03 – System Analysis and Design

3.1. Introduction

The developments behind the development of Blue Heaven Rentals which is an online boarding house finding system includes system analysis and design.

This chapter explains the analytical processes, identification of requirements and system modeling applied in coming up with a fuller, efficient, and easy to operate solution. It covers both the functional and non-functional analysis, and it is backed by graphic means including the UML diagrams and the ER model.

This chapter will be aimed at having an overall awareness of the relationship among the system elements and how they interact towards realizing the ultimate goal which is the linking of boarding seekers to proven boarding house owners using a quality, real-time, and conveniently considered web platform.

3.2. Requirement Analysis

3.2.1. Requirement Gathering Techniques

The process of requirement gathering is an important process of the system development, which facilitates knowing the needs, expectations, and difficulties of the potential users of the system, in advance of the system implementation.

In the case of Blue Heaven Rentals, the primary and secondary research methods were adopted in order to gather extensive and trusted information about actual users and market surveys.

1. Surveys & Questionnaires (Primary data collection)

To understand the real problems faced by people searching for boarding places, a **Google Form** was design and distributed among **university students, employees, and job seekers** across difference regions. The questionnaire was structured with **multiple choice** allowing respondents to express their experiences in detail.

The survey covered the following key areas:

- Difficulties faced when searching for boarding houses (e.g., fake ads, lack of photos, unclear pricing).
- Preferred features in an online platform (e.g., verified listings, contact options, search filters).
- Opinions on using technology or websites to find accommodation.
- Expectations for safety, user-friendliness, and communication features.

The result revealed that:

- ✓ 82% of respondents faced difficulties finding reliable boarding houses online.
- ✓ 76% preferred a verified system with real-time availability updates.

- ✓ 69% mentioned that current platforms lack communication options between seekers and owners.

These findings strongly supported the need for a more interactive and trustworthy solution, leading to the concept of Blue Heaven Rentals.

2. Interviews with Boarding House Owners.

Local boarding house owners were interviewed directly to get to know their struggles when listing and managing their properties.

Most owners complained about existing sites because of:

- Problem updating the availability status after rooms had been reserved.
- Time wasted answering repetitive inquiries.
- Lack of trust in online seekers due to fake profiles.

Based on this understanding, certain characteristics such as **owner dashboards, administrator verification** and **automatic availability updates** were incorporated within the proposed system.

3. Observation of Existing Systems

Current web systems like **Bodima.lk** and **Ikman.lk** was surveyed to denote usability concern and weaknesses.

Even though they offer a simple listing platform, they are not **up-to-date in real time**, do not have **role based access** and **built-in messaging functions -** the most important elements in the expectation of modern usage.

It is based on this analysis that the state of the art features like real-time Firebase syncing, and secure chats were included in Blue Heaven Rentals.

4. Document and Literature Review

Secondary sources were to be used including analysis of **academic articles, technology web blogs and case studies** pertaining online rental.

These sources served to learn best practices concerning:

- Incorporating firebase-based scalable systems.
- Creating user-friendly user interfaces to rental websites.
- Multi-user platform data integrity and security.

Another interesting outcome that was verified by the review was the increasing tendency toward mobile-first web administration and real-time information management that would also be in line with the technological emphasis in the project.

3.3. Use Case

The Use Case diagram visually represents user interactions with the system. It helps to identify all external entities (actors) and their relationships with internal system functions.

In Blue Heaven Rentals, there are three main actors:

- Boarding Seeker
- Boarding Owner
- Administrator

3.3.1 High level Use Case Diagram for System

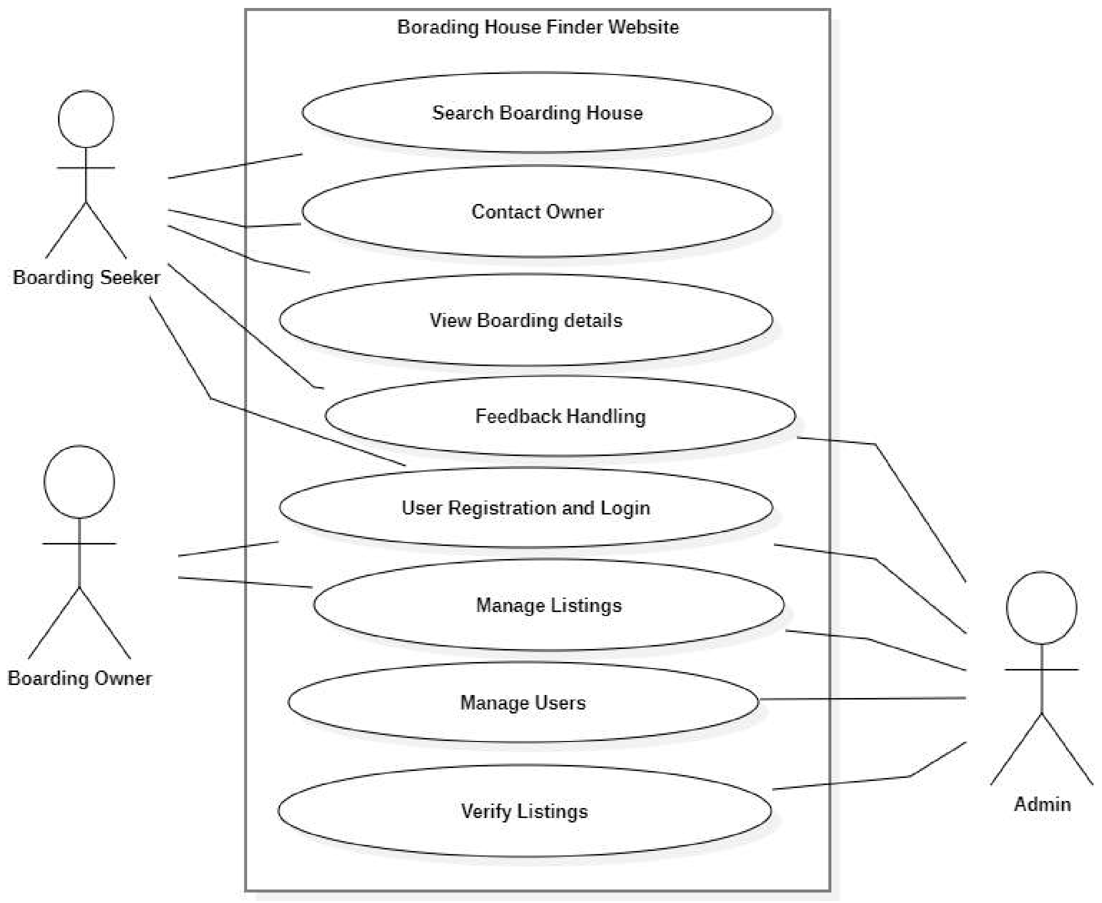


Figure 0-1 High level Use Case Diagram for System

3.3.2 Use Case Diagram for User Registration

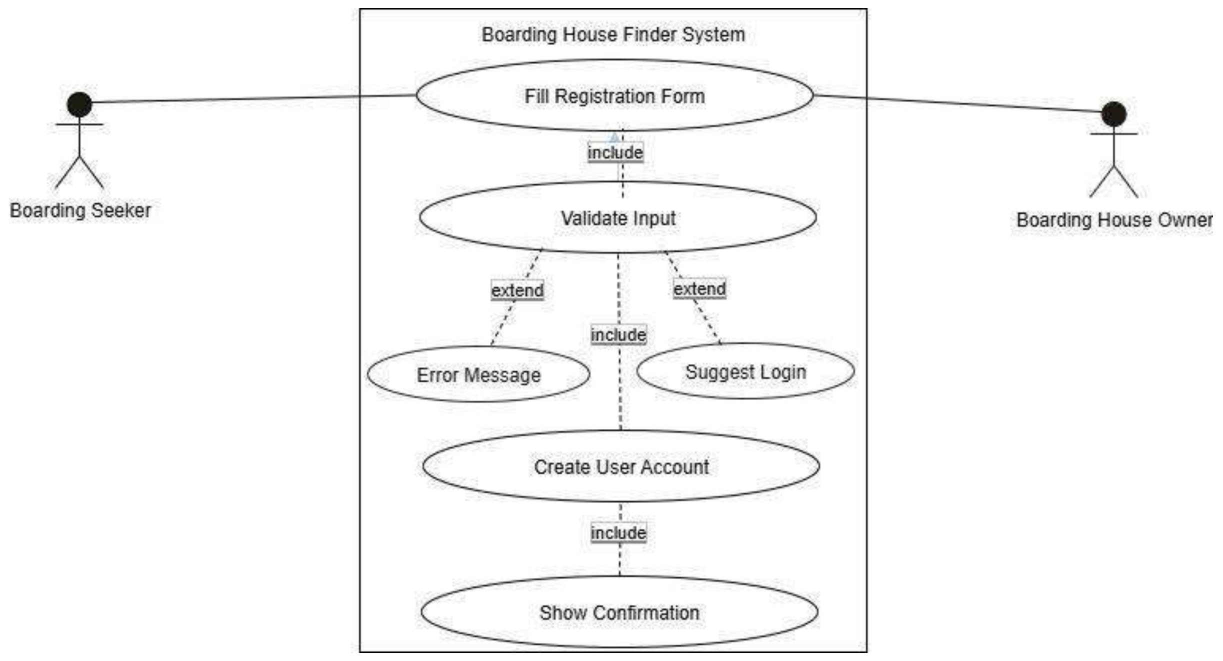


Figure 0-2 Use Case Diagram for User Registration

3.3.3. Use Case Diagram for Login

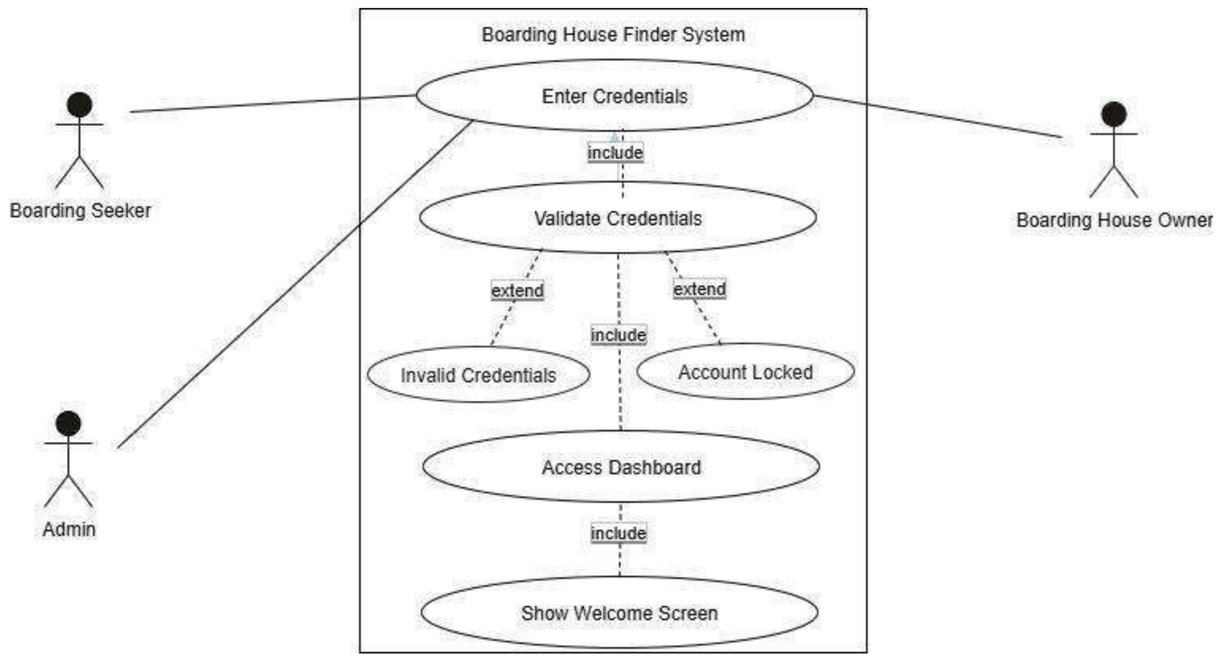


Figure 0-3 Use Case Diagram for Login

3.3.4 Use Case Diagram for Search Boarding

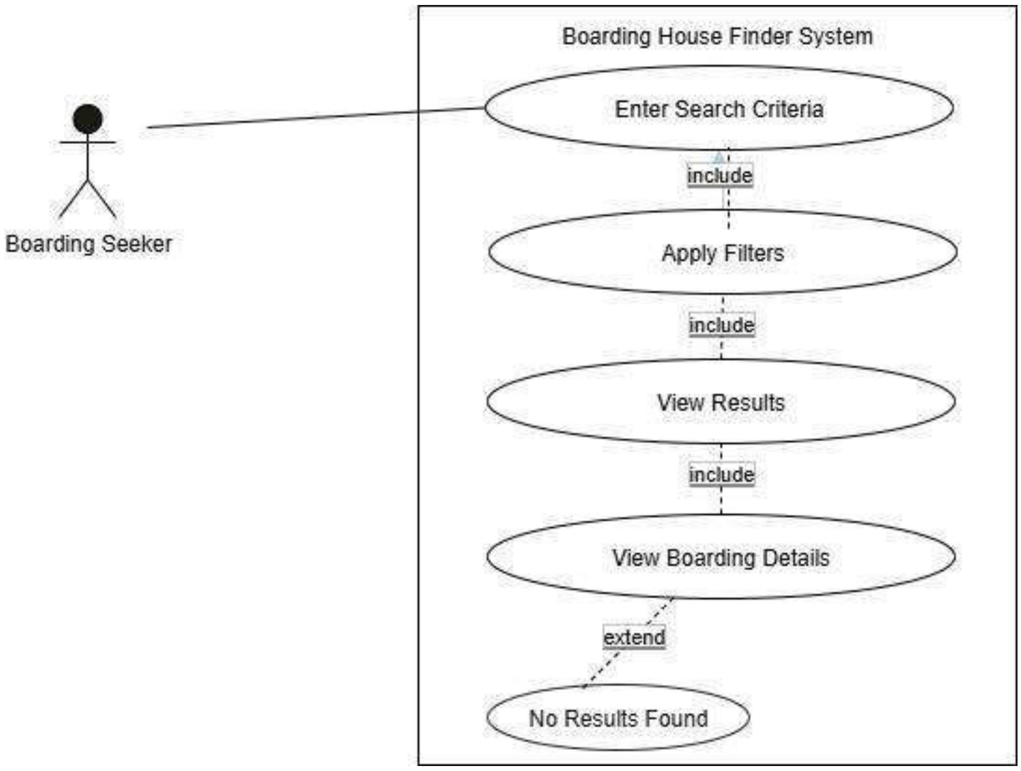


Figure 0-4 Use Case Diagram for Search Boarding

3.3.5. Use Case Diagram for Manage Listings

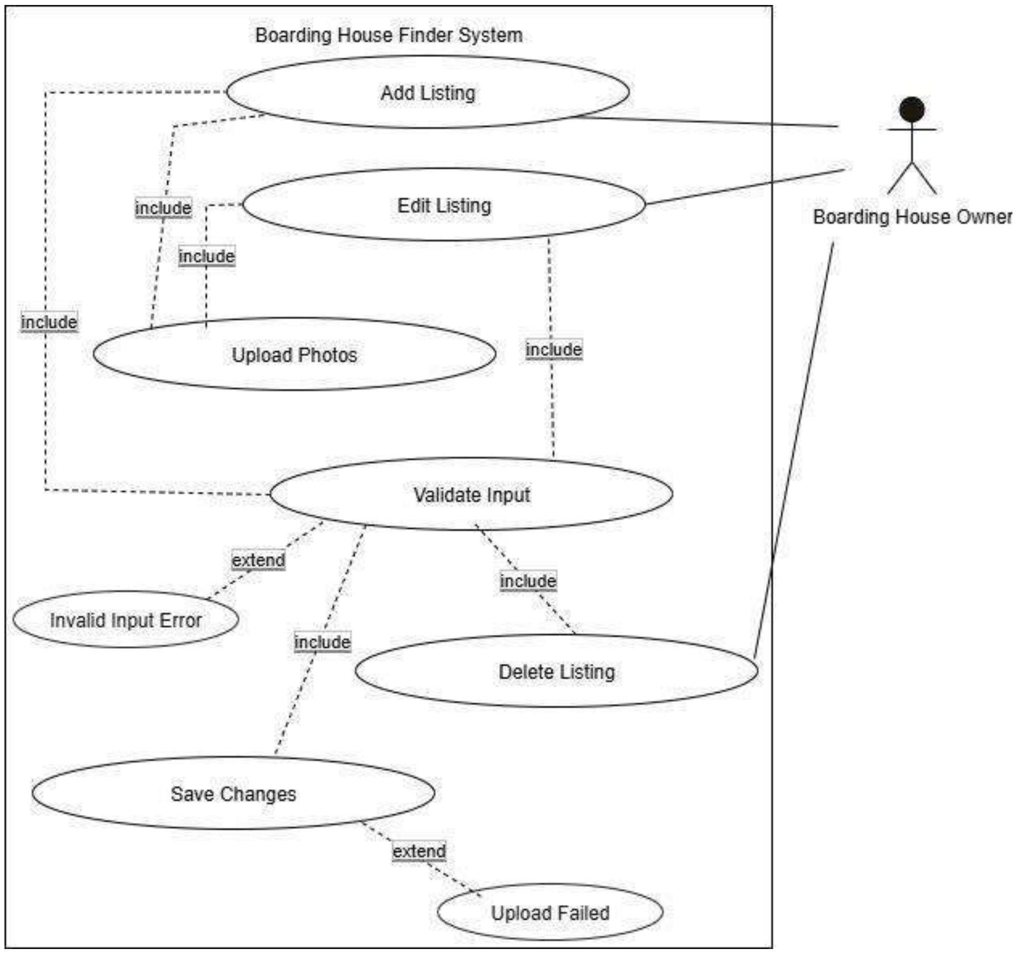


Figure 0-5 Use Case Diagram for Manage Listings

3.3.6. Use Case Diagram for Verify Listings

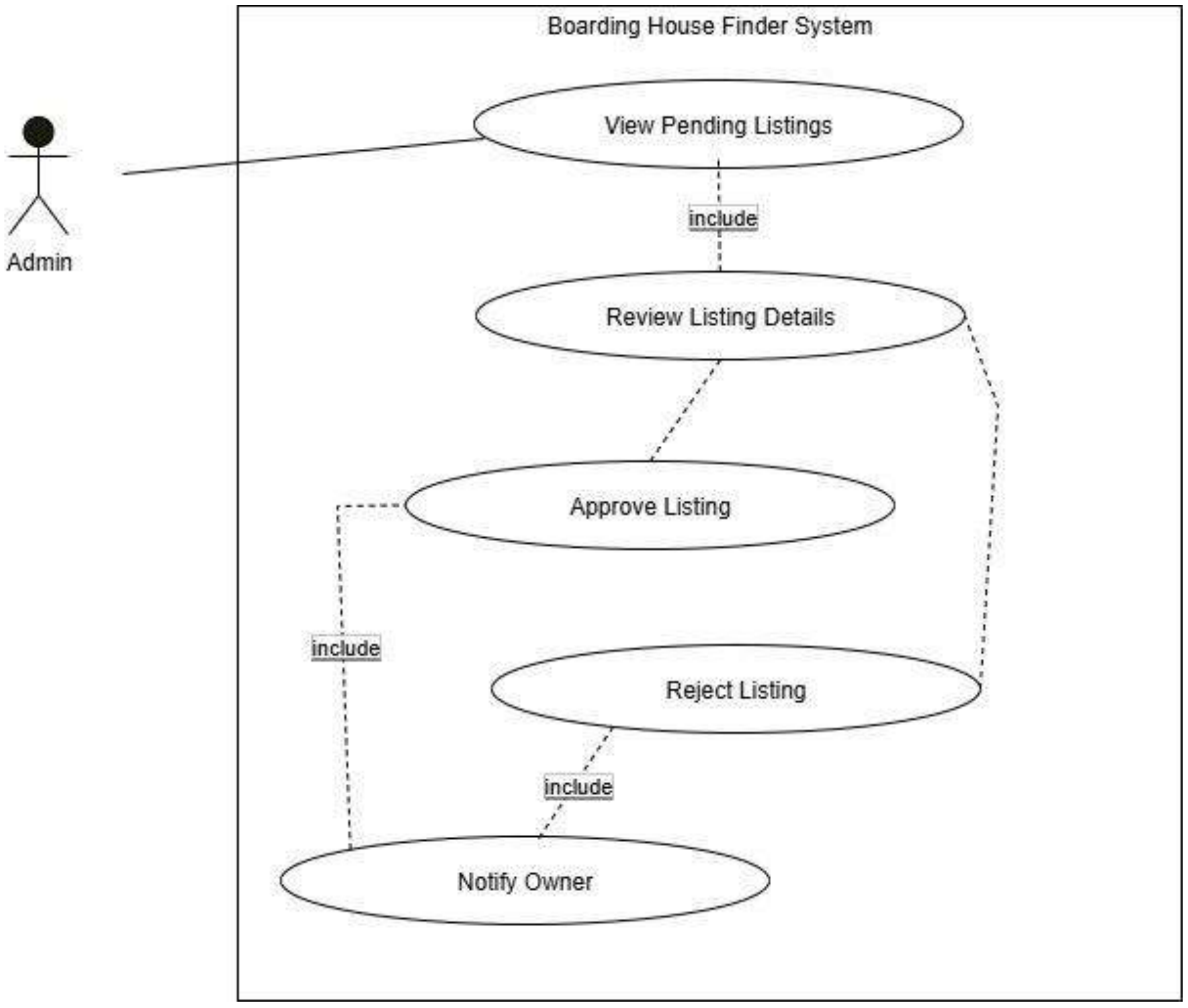


Figure 0-6 Use Case Diagram for Verify Listings

3.4. Activity Diagrams

Activity diagram describes the workflow of important processes within the system. Representing the home page actions, choice point, and control flow of the user activities, it could be the procedure of their processing by logging into the system and selecting the boarding house one wants or the description of their transactions with the listings management. Through visualization of these activities, the diagram elucidates the active mechanism incorporated in the system and the roles of every user role.

3.4.1 Activity Diagram for Boarding House Reservation

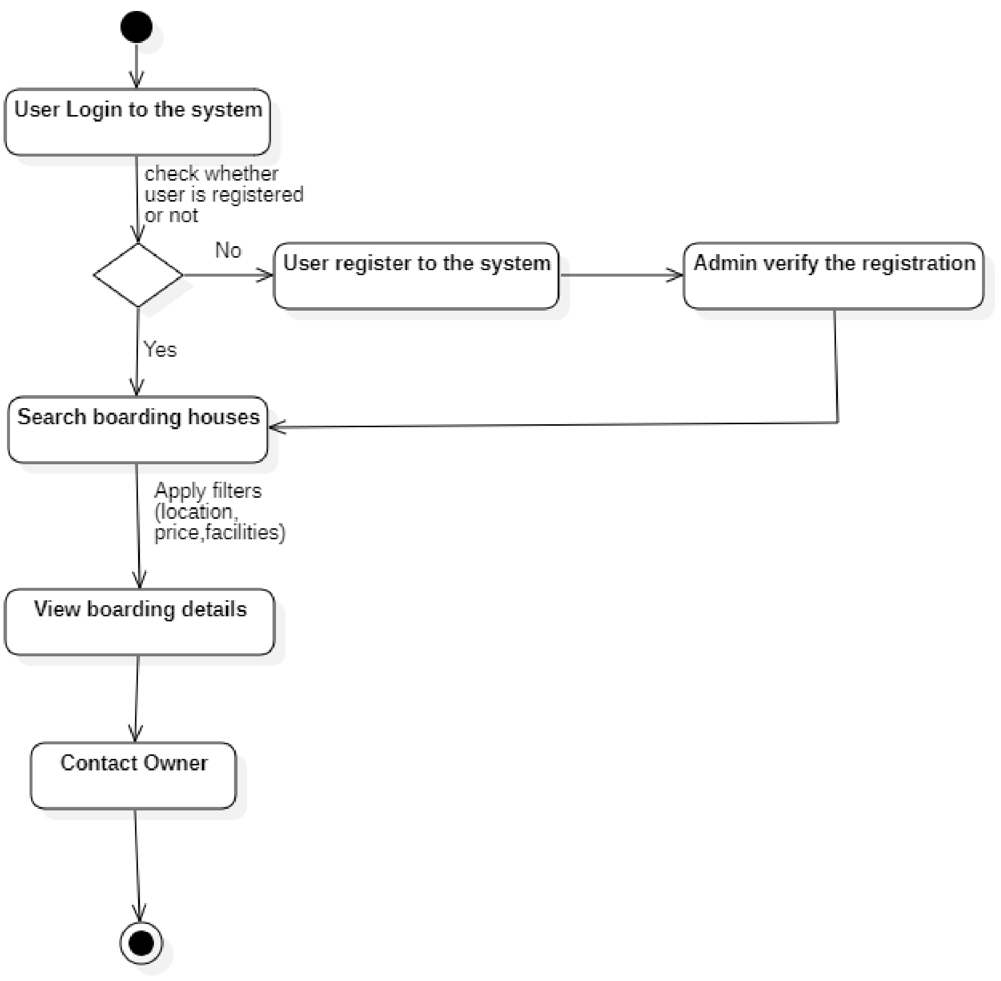

User Login to the system
check whether user is registered or not
No
User register to the system
Admin verify the registration
Yes
Search boarding houses
Apply filters (location, price,facilities)
View boarding details
Contact Owner

Figure 0-7 Activity Diagram for Boarding House Reservation

3.4.2 Activity Diagram for Add listings

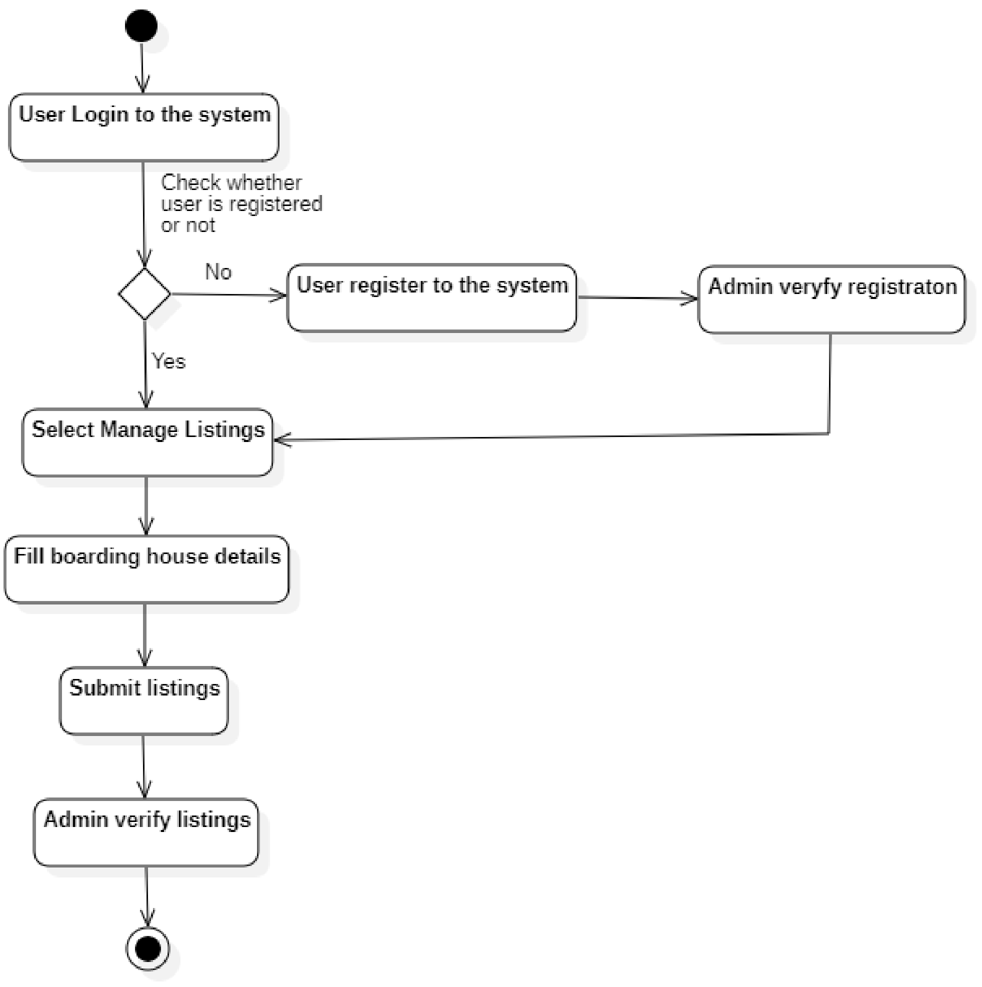


Figure 0-8 Activity Diagram for Add listings

3.4.3 Activity Diagram for Feedback Handling

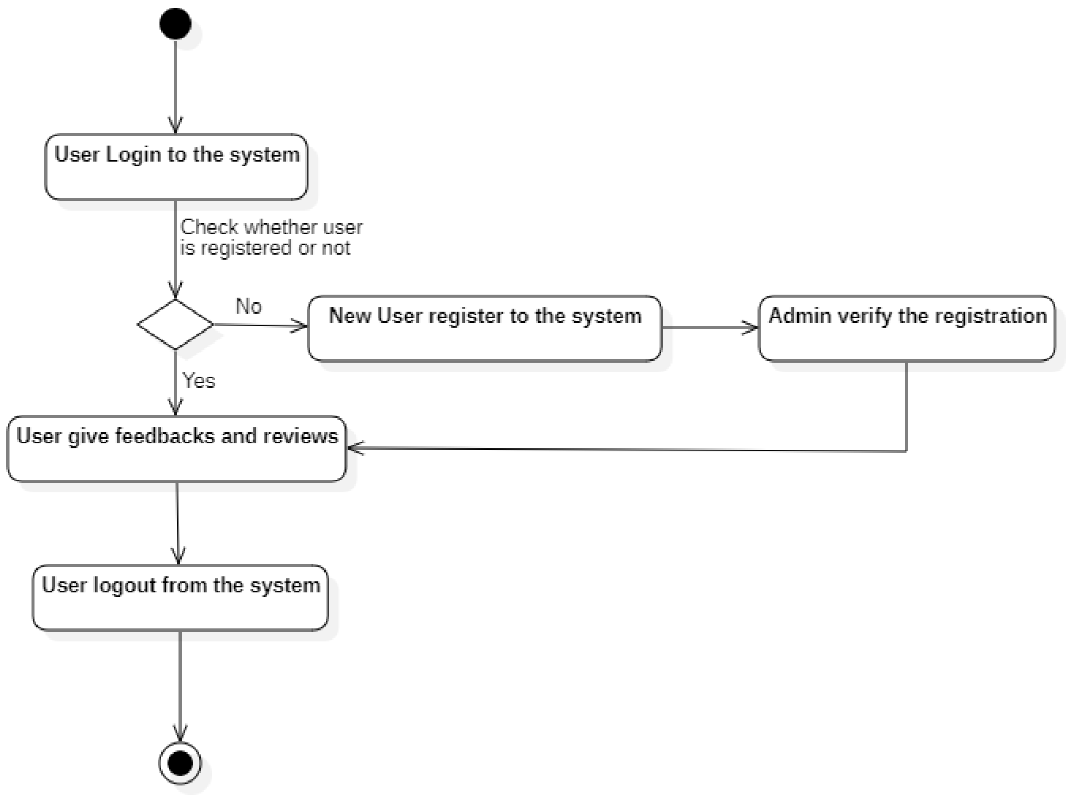

User Login to the system
Check whether user is registered or not
No
New User register to the system
Admin verify the registration
Yes
User give feedbacks and reviews
User logout from the system

Figure 0-9 Activity Diagram for Feedback Handling

3.5 Class Diagram

Class diagram displays the static format of the system comprising of the classes, their attributes, methods and their relationships. It is an indication of the ways other entities in the system, including users, seekers, owners, admins and boarding houses, relate to one another in the system through inheritance and associations. This is because this diagram helps in informing the object oriented design of the system.

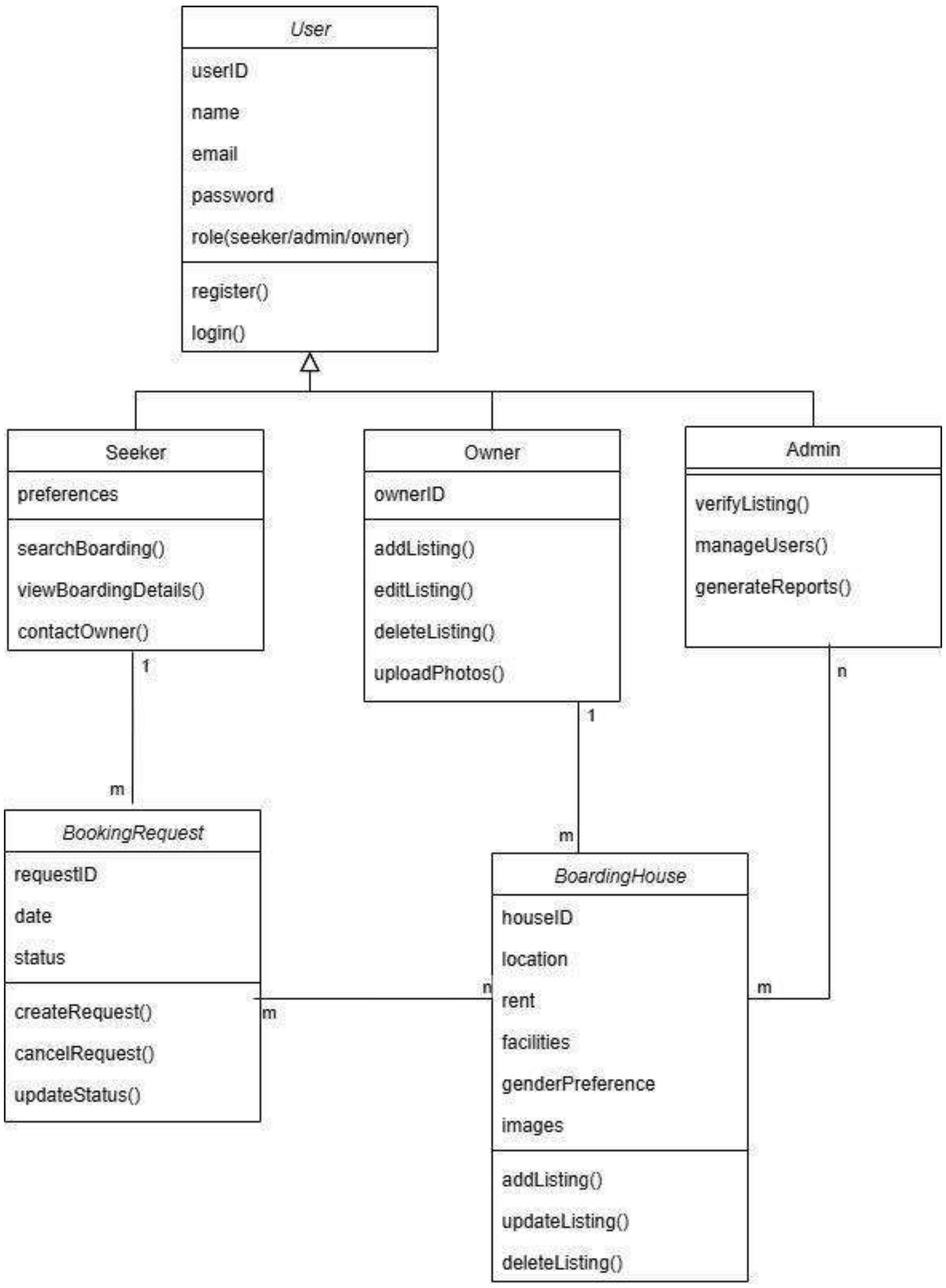


Figure 0-10 Class Diagram

3.6. Sequence Diagram

3.6.1 Sequence diagram for User Login

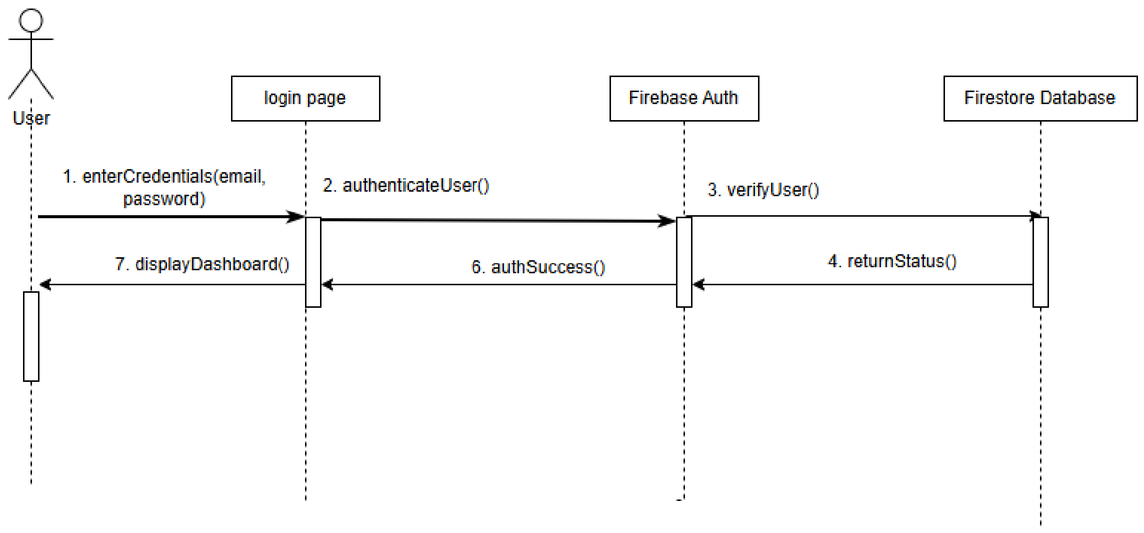


Figure 0-11 Sequence Diagram for User Login

3.6.2 Sequence diagram for Add Listing

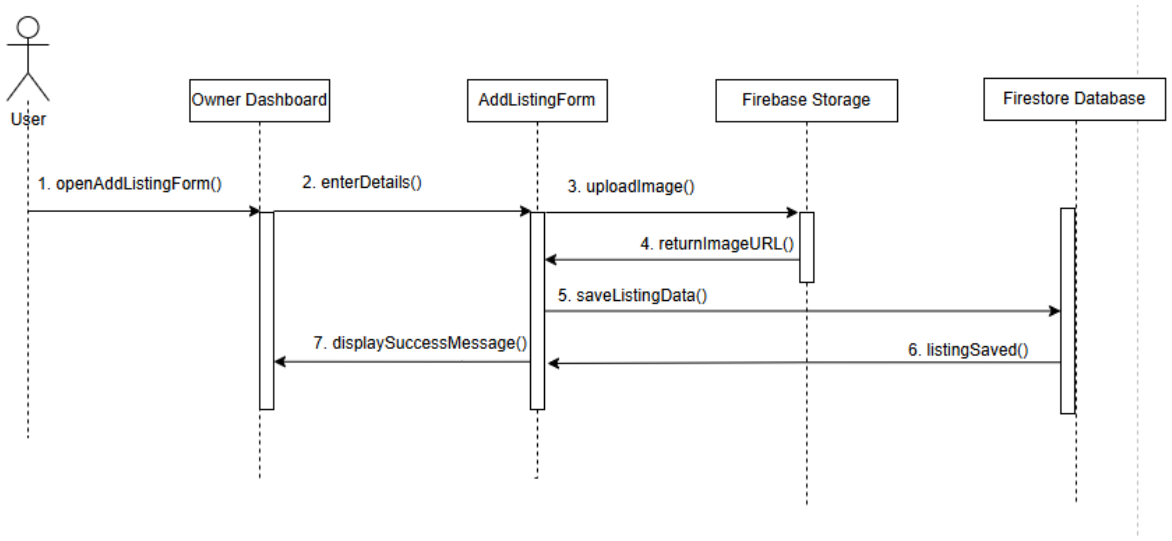


Figure 0-12 Sequence diagram for Add Listing

3.7. ER Diagram

The ER diagram indicates the logical design of the database; it implies the relationships among entities (the user, boarding houses, owner and admin), and also shows the attributes of the entities and the relations between them. It makes certain that required data of the system is captured to the end and has a base that allows database design and formation.

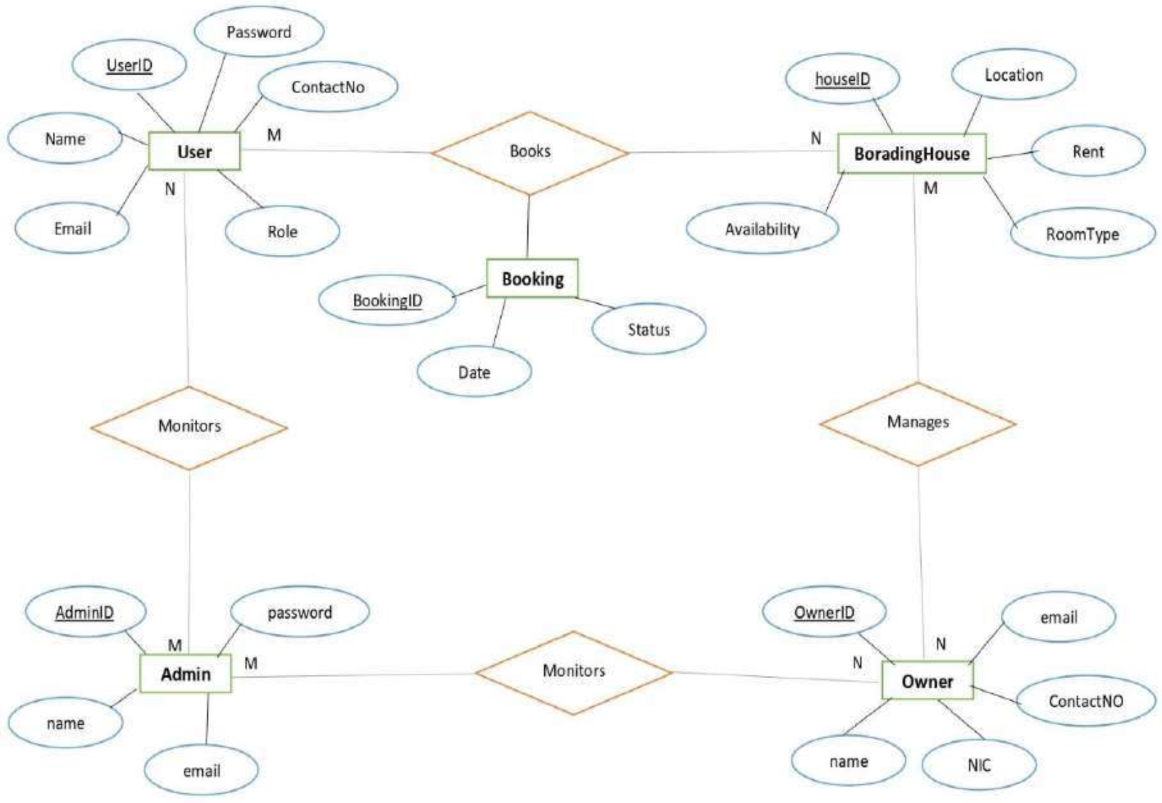


Figure 0-13 ER Diagram

3.8. Relational Schema

1. User(<u>UserID</u>, Name, Email, Password, ContactNo, Role)
2. Owner(<u>OwnerID</u>, name, NIC, ContactNo, email)
3. BoardingHouse(<u>houseID</u>, <u>OwnerID</u> (Foreign Key), Location, Rent, RoomType, Availability)
4. Booking(<u>BookingID</u>, <u>UserID</u> (Foreign Key), <u>houseID</u> (Foreign Key) , Date, Status)
5. Admin(<u>AdminID</u>, name, email, password)

3.9. Summary

Overall, this chapter provided detailed analysis and design of the business of Blue Heaven Rentals.

The chapter not only discussed functional and non-functional requirements but also went through user interactions using UML diagrams and gave the logical structure using the ER model. Firebase integration and modular design of the system guarantee flexibility, performance and scalability.

In general, the system design will have a strong base in the implementation with a hope of providing secure and convenient boarding house finder platform that will overcome the real-life issues of the students, staff, and job seekers in Sri Lanka.

Chapter 04 – Technologies Adapted

4.1. Introduction

In this chapter, the kind of technologies, tools and platforms, which were used in the development of Blue Heaven Rentals web based boarding house finding system, are described.

Each technology was chosen depending on the requirements of the system that included scalability, performance, security and ease of development.

Because the offered system is dedicated to keeping real-time updates, cloud integration as well as user interactivity, the technology stack will incorporate front-end web technologies (react, JavaScript) in addition to a cloud-based solution Firebase, offered by Google.

Below are the hardware, software, and detailed technologies embraced in the development process as well as deployment.

4.2. Software Requirements

The software requires explain the necessary software elements that were involved in the creation, creation and implementation of the Blue Heaven Rentals software.

Software Components	**Description**
Operating System	Windows , MacOs and any operating systems
Web Browser	Google chrome / Opera / Mozilla firefox (for testing and running the web app)
Front-end Language	React , JavaSript
Back-end Service	Firebase Authentication and fire store for managing data and user session
Database	Firebase cloud store
Development Tool	Firebase console for coding, testing and database configuration
Version Control	GitHub maintained to keep a version of the project and to share the latest updates
Design Tools	Figma, photoshop

Table 4.2-0-1 Software Requirements

4.3. Hardware Requirements

The Blue Heaven Rentals system is a web-based application which is to be said the fully web based application and this implies that it can be accessed by any such modern device that has a web browser and is connected to the internet.

Thus, the system does not demand advanced hardware and localizations. The system is accessible to both users and administrators through the cloud services.

User Type	Device Requirements	Description
Client (Boarding seekers / Owners)	Any smartphone, tablet, laptop or desktop with a web browser	The system is responsive and browser-based so it can be accessed using any device, and not installed.
Administrator	Standard PC or laptop with strong internet access	Used to provide user management, listing verification and activity monitoring using the administration panel.

Table 4.3-0-2 Hardware Requirements

4.4. Technologies

4.4.1. Front-end Technologies

The web interface of Blue heaven rentals is built with the help of React and Javascript to provide a responsive and interactive front end.

- React is a component-based JavaScript library that lets developers create interactive applications that are loaded evolves in a single-page version with a fast renderer in a Virtual DOM.
- JavaScript (ES6) includes applications in the client side (JavaScript), user interaction and communication with Firebase back-end.

A combination of these technologies ensures a smooth user experience so that the system works effectively across the devices and browsers.

4.4.1. Back-end Technologies

Firebase, which is a product created by Google to create cloud-based applications, is used as the backend.

- Firebase Authentication:
  Provides email/ password verification to users on secure sign in and registration. This is to guarantee that system functionalities are only accessed by authorized users.
- Firebase Firestore:
  It is the major real-time database to store and control listing, user profile, message, and booking information. The structure of the NoSQL permits adaptable, swift data queries.
- Firebase Cloud storage(Optional Integration):
  Have the option to store multimedia files such as images uploaded to it by the owners as part of their boarding listings.

This is a server less backend architecture that is scalable and provides maintenance-free and device-synchronization.

4.5. Other Requirements

In addition to the hardware and software equipment, the system had a few non-technical requirements to make implementation easy:

- Internet Connection: Customers require a stable Internet connection in order to access Firebase real-time services.
- Google Account: Essential in the resource of Firebase console and hosting.
- Browser Compatibility: Browser is compatible on Google Chrome, Microsoft edge and Mozilla Firefox.
- Security Compliance: The encryption of data over the communication between the users and the server is secured by the use of SSL (HTTPS).

4.6. Summary

This chapter has talked of the major technologies and tools that were utilized in developing Blue Heaven Rentals. It was developed using the frontend with React and Java-Script and Firebase as the backend service and includes authentication service and real time database management.

The technologies selected have guaranteed scalability, security, and responsiveness which make the system reliable and available using any web browser on any device.

Chapter 05 – Implementation

5.1. Introduction

This chapter provides an account of the implementation phase of Blue Heaven Rentals where a functional web application was created out of the designed system.

The key consideration in the implementation was to make the frontend interface (React + JavaScript) in line with the Firebase backend to experience real-time, secure, and user-friendly.

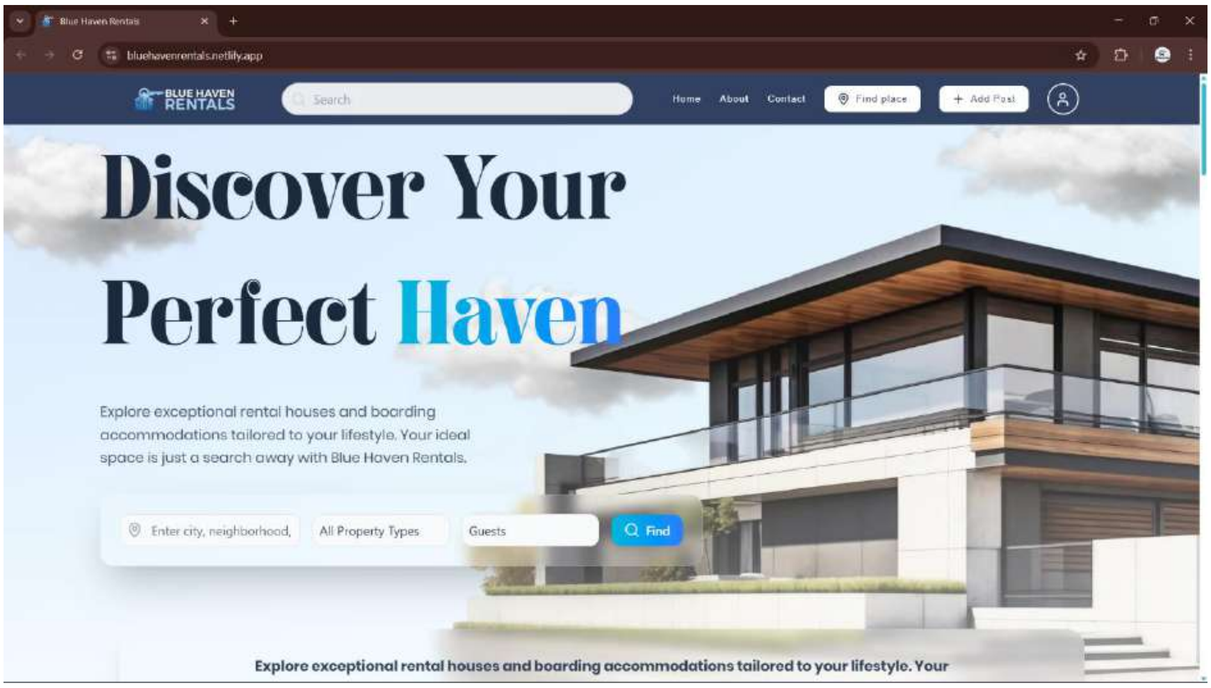
The process involved development of interactive user interfaces, combining Firebase Authentication and Firestore and also testing its functionality to be reliable and easy to use.

5.2. Front-end User Interfaces

The frontend implementation entails the development of an interactive and responsive interface in which the users are able to execute the important functions of the system without any problems.

In all interfaces, simplicity, ease of use and navigation were kept in the system.

Home Page



Discover Your Perfect Haven

Explore exceptional rental houses and boarding accommodations tailored to your lifestyle. Your ideal space is just a search away with Blue Haven Rentals.

Explore exceptional rental houses and boarding accommodations tailored to your lifestyle. Your

Figure 0-1 Home Page

User Registration Page

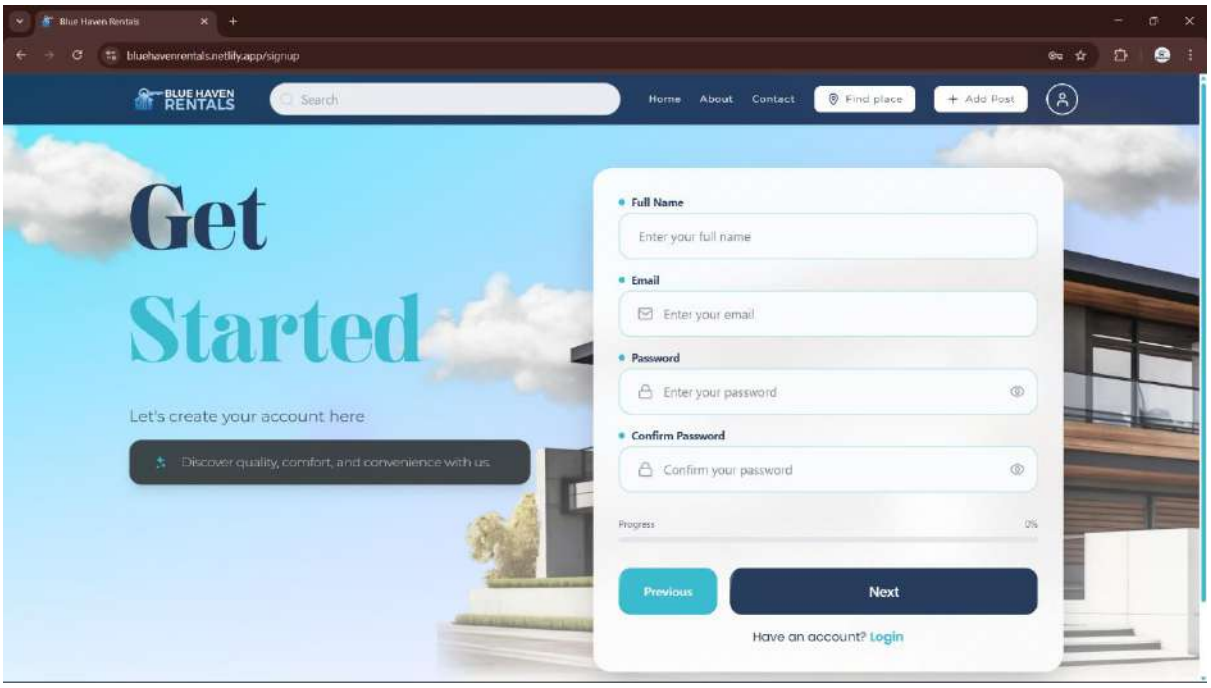


Figure 0-2 User Registration Page

Login Page

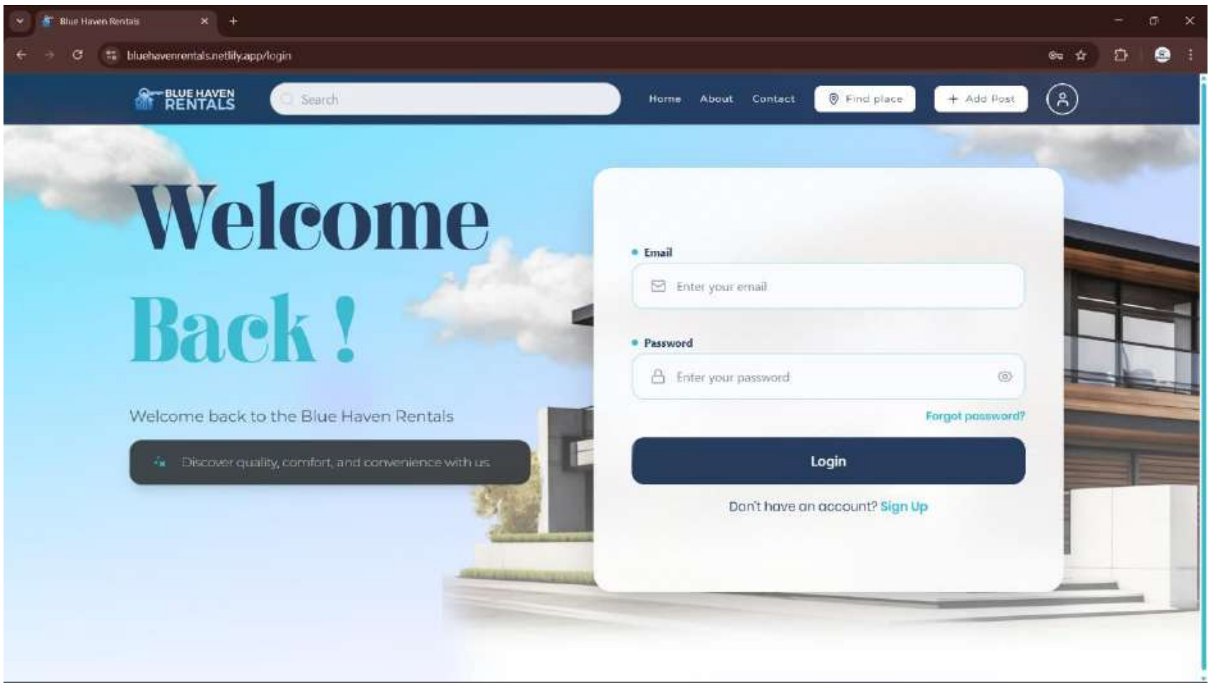


Figure 0-3 Login Page

Search & Filter Page

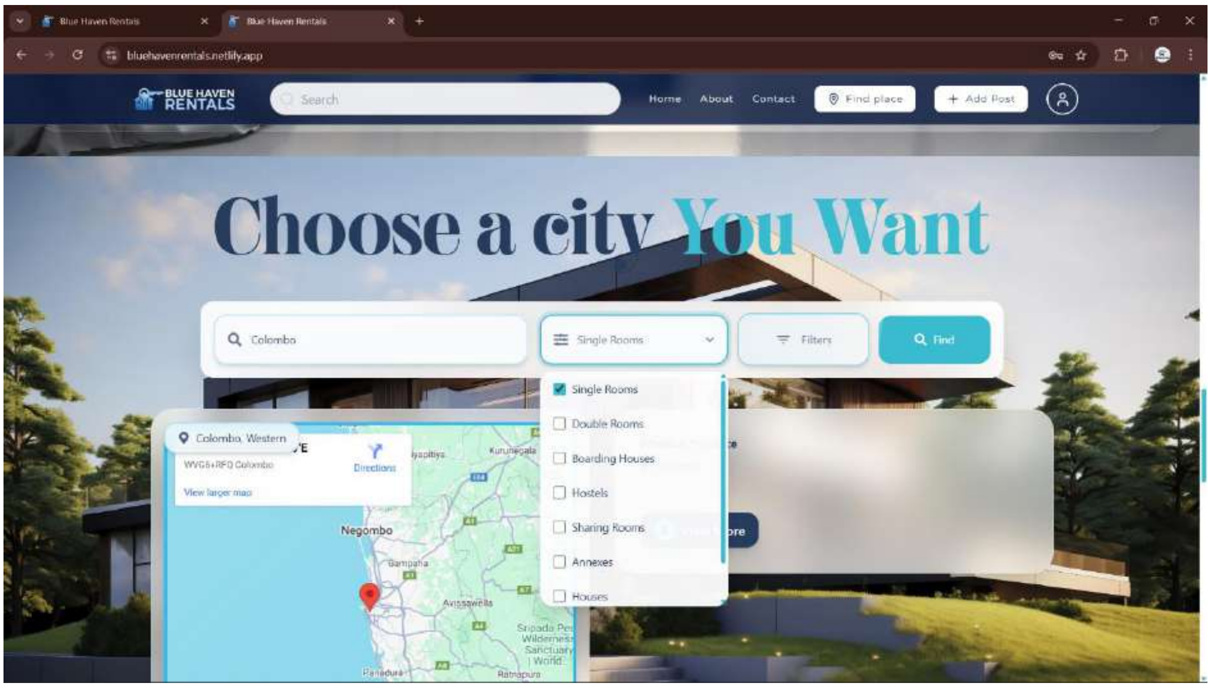


Figure 0-4 Search & Filter Page

Search Results Page

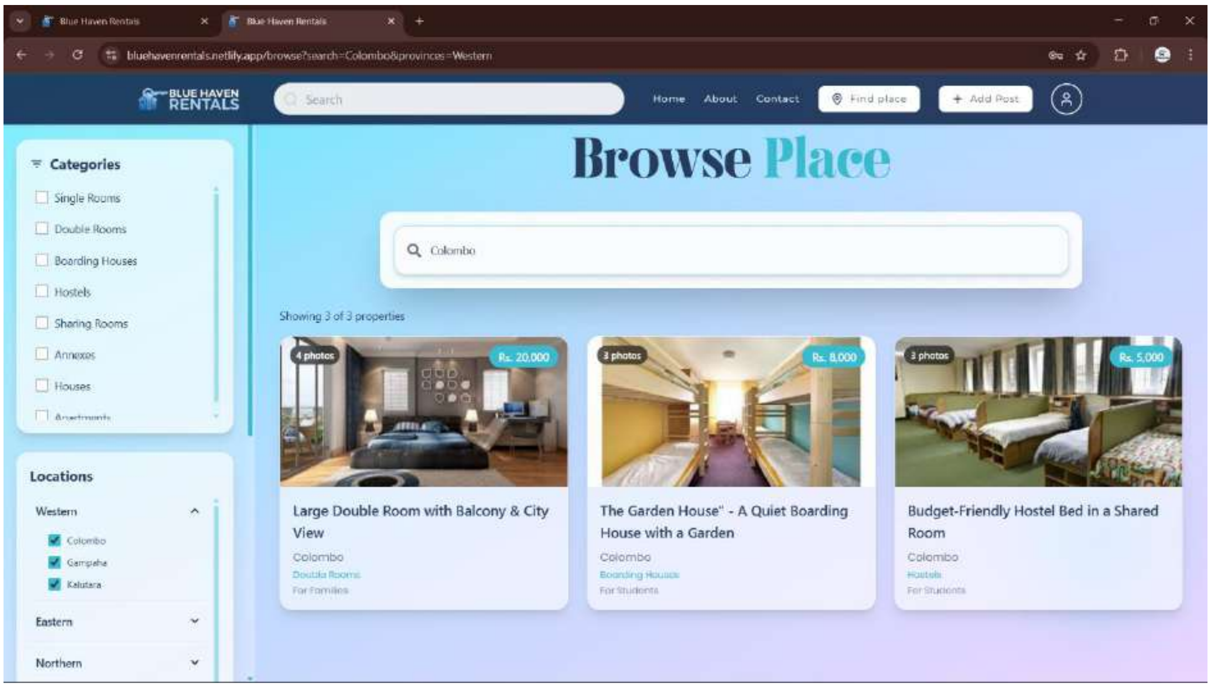
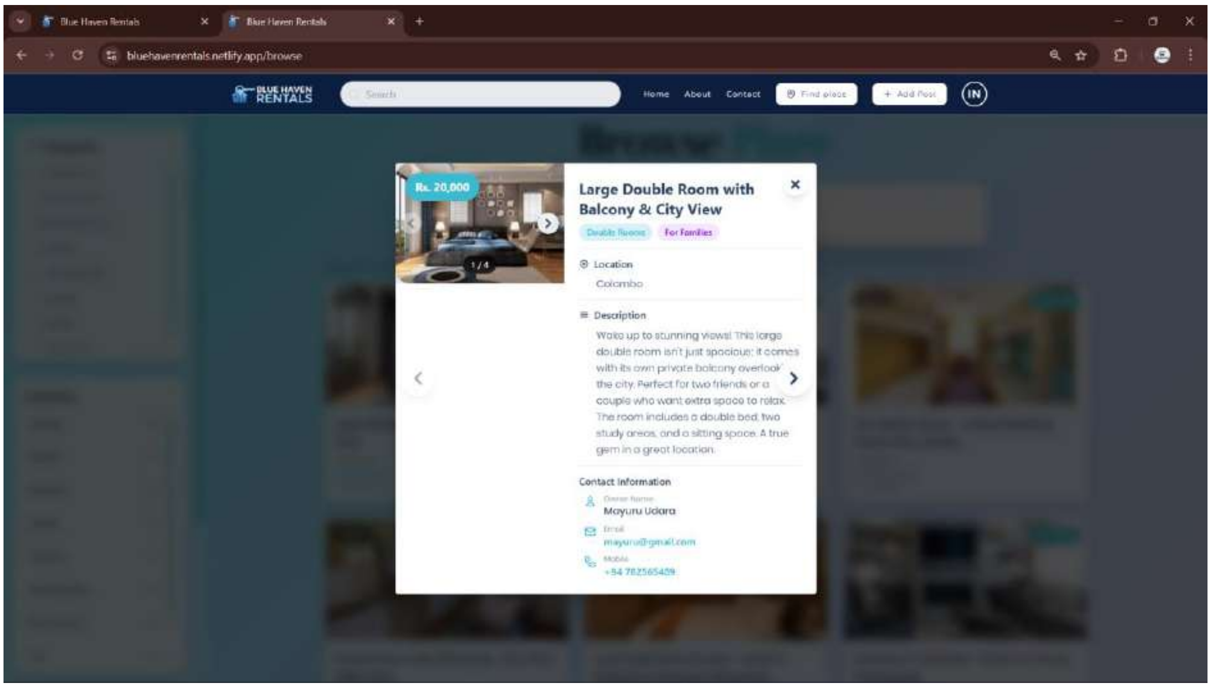


Figure 0-5 Search Results Page

Boarding Details Page



Large Double Room with Balcony & City View

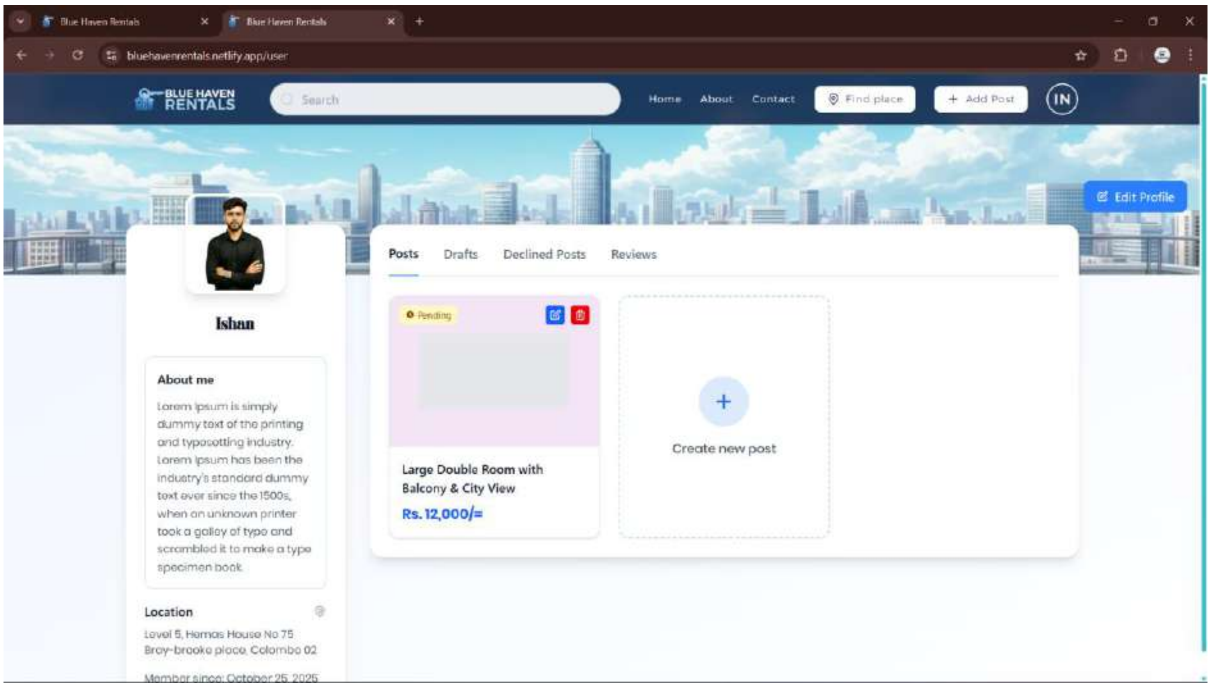
Location
Colombo

Description
Wake up to stunning views! This large double room isn't just spacious; it comes with its own private balcony overlook the city. Perfect for two friends or a couple who want extra space to relax. The room includes a double bed, two study areas, and a sitting space. A true gem in a great location.

Contact Information
Owner Name
Mayuru Udara
Email
mayuru@gmail.com
Mobile
+94 782565409

Figure 0-6 Boarding Details Page

Owner Dashboard



Ishan

About me
Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Location
Level 5, Hemas House No 75 Bray-brooke place, Colombo 02

Member since: October 25, 2025

Large Double Room with Balcony & City View
Rs. 12,000/=

Figure 0-7 Owner Dashboard

Add New Listing Form

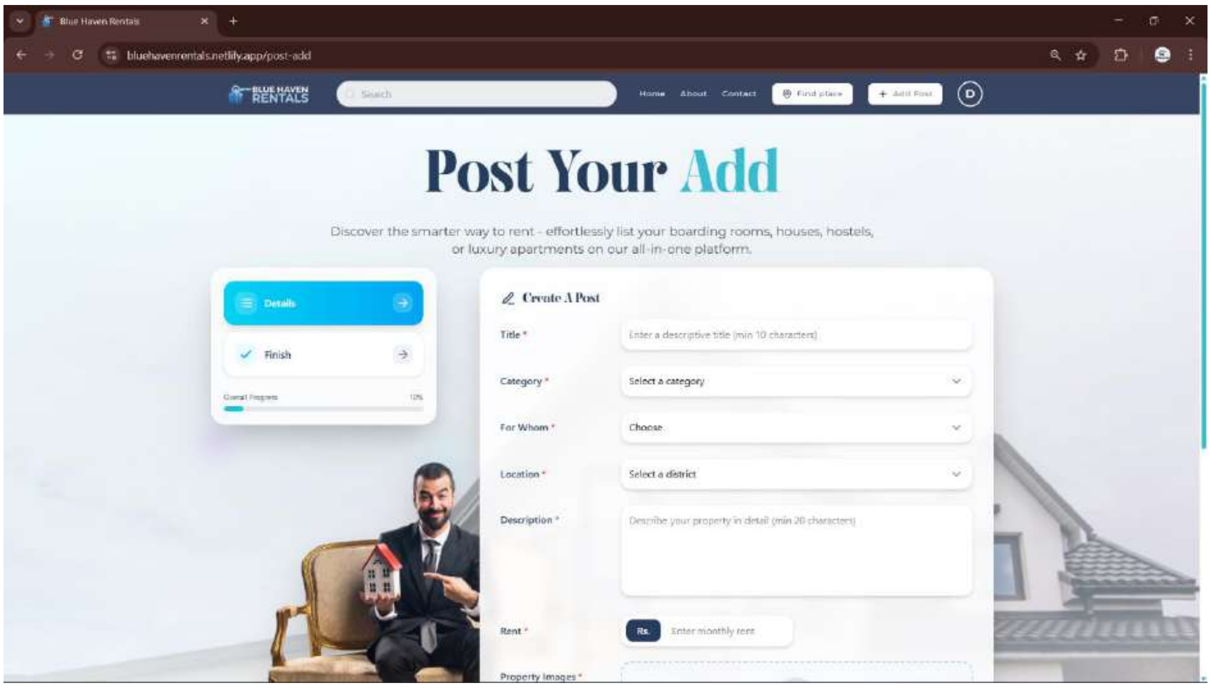

bluehavenrentals.netlify.app/post-add
Home
About
Contact
Find place
Add Post
Post Your Add
Discover the smarter way to rent - effortlessly list your boarding rooms, houses, hostels, or luxury apartments on our all-in-one platform.
Details
Finish
Overall Progress
Create A Post
Title *
Enter a descriptive title (min 10 characters)
Category *
Select a category
For Whom *
Choose
Location *
Select a district
Description *
Describe your property in detail (min 20 characters)
Rent *
Rs.
Enter monthly rent
Property Images *

Figure 0-8 Add New Listing Form

Feedback Form

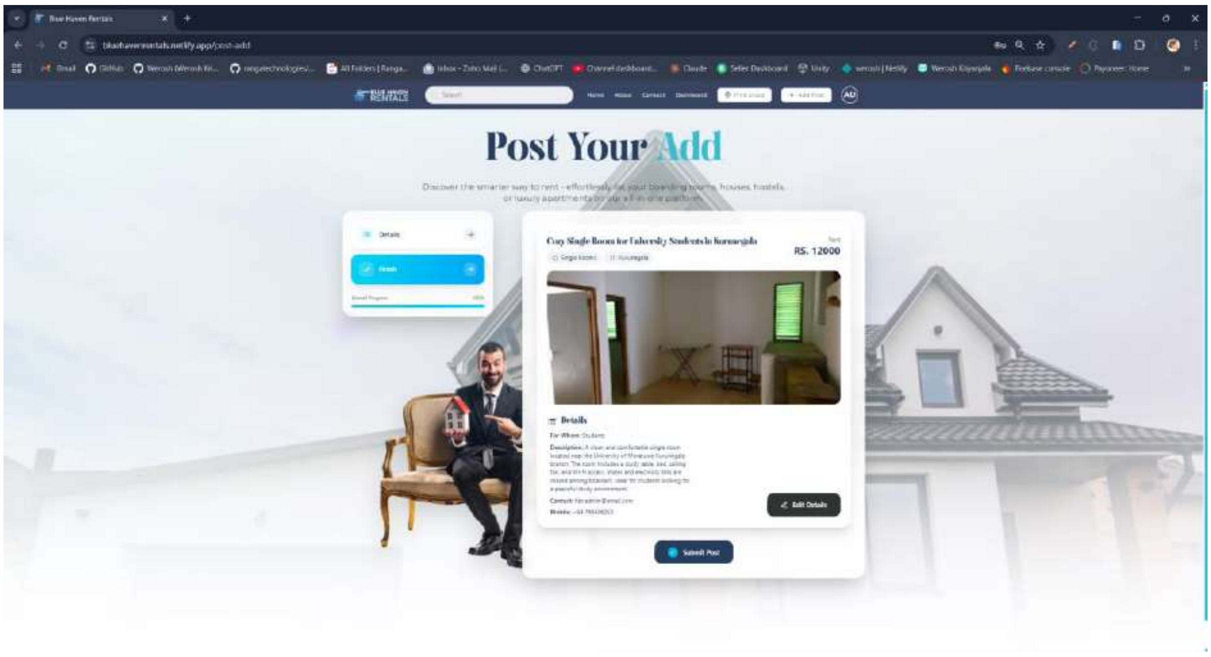
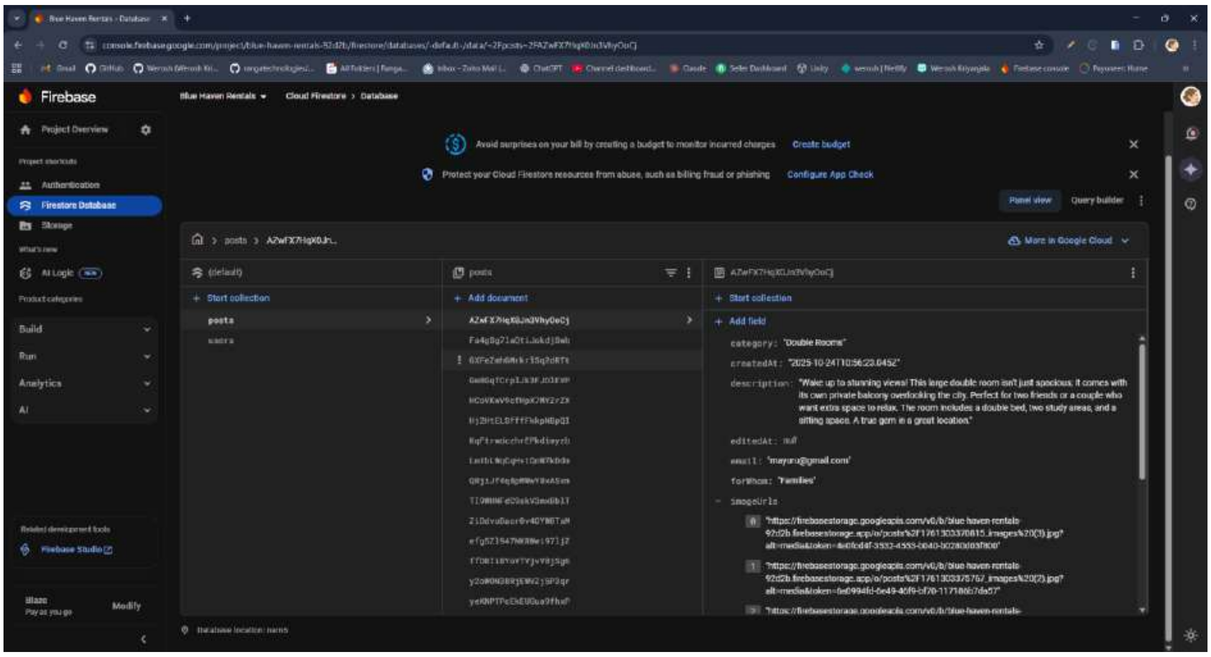


Figure 0-9 Feedback Form

5.3. Back-end User Interfaces

The backend implementation was handled entirely through Firebase, a server less cloud platform offering real-time database and authentication service.

Firebase Firestore Database View



Blue Haven Rentals > Cloud Firestore > Database

Avoid surprises on your bill by creating a budget to monitor incurred charges — Create budget

Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing — Configure App Check

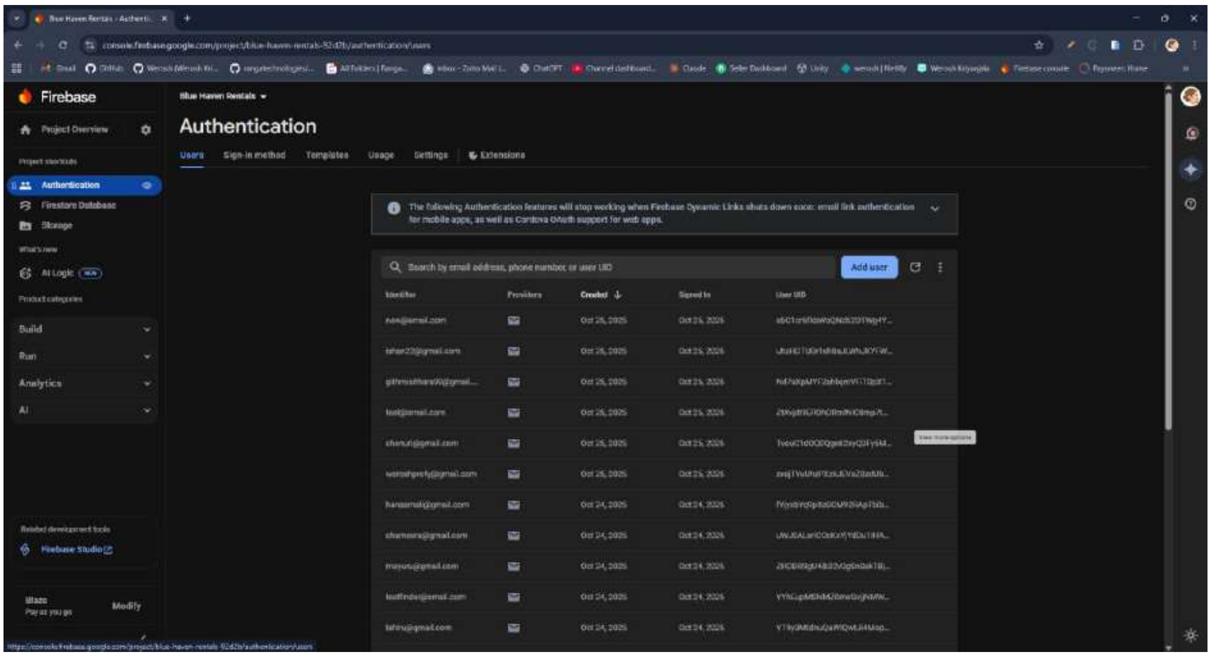
posts > AZwFX7HqX0Jn...

(default)	posts	AZwFX7HqX0Jn3VhyOoCj
posts	AZwFX7HqX0Jn3VhyOoCj	category: "Double Rooms"
users	Fa4gBg71aOtiJokdjBwb	createdAt: "2025-10-24T10:56:23.045Z"
	6XFeZeh6Mrkr1SqZdRTt	description: "Wake up to stunning views! This large double room isn't just spacious; it comes with its own private balcony overlooking the city. Perfect for two friends or a couple who want extra space to relax. The room includes a double bed, two study areas, and a sitting space. A true gem in a great location."
	GwmGqfCrpIJk3FJOIRWP	editedAt: null
	HCoVKwV9cfHpX2NY2rZX	email: "mayuru@gmail.com"
	HjZHtELDfffFkkpNBpQ3	forWhom: "Families"
	KqftrwdcchrEPkdiwyzb	imageUrls
	LmtbLNpCqHv1QnW7kDde	0 "https://firebasestorage.googleapis.com/v0/b/blue-haven-rentals-92d2b.firebasestorage.app/o/posts%2F1761303370815_images%20(3).jpg?alt=media&token=4e0fcd4f-3532-4553-b040-b0280d03f800"
	QRj1Jf4qRpMNwY8xASxm	1 "https://firebasestorage.googleapis.com/v0/b/blue-haven-rentals-92d2b.firebasestorage.app/o/posts%2F1761303375767_images%20(2).jpg?alt=media&token=0e0994fd-0e49-46f9-bf70-11718667da57"
	TI9MHNFd2OskV3mdGb1T	2 "https://firebasestorage.googleapis.com/v0/b/blue-haven-rentals-
	ZiDdvuDacr0v4OYNBTaM	
	efg5Z1S47NKRBwi971j2	
	fTOBI1BYoYTYjvV8jSgn	
	y2oWON3BRjEWV2jSP3qr	
	yeKNPTPcEkEUGua9fhxP	

Database location: nam5

Figure 0-10 Firebase Firestore Database View

Firebase Authentication Console



Blue Haven Rentals

Authentication

Users · Sign-in method · Templates · Usage · Settings · Extensions

The following Authentication features will stop working when Firebase Dynamic Links shuts down soon: email link authentication for mobile apps, as well as Cordova OAuth support for web apps.

Search by email address, phone number, or user UID — Add user

Identifier	Providers	Created	Signed In	User UID
new@email.com	✉	Oct 25, 2025	Oct 25, 2025	[illegible]
ishan22@gmail.com	✉	Oct 25, 2025	Oct 25, 2025	[illegible]
githmadharaX@gmail...	✉	Oct 25, 2025	Oct 25, 2025	[illegible]
test@email.com	✉	Oct 25, 2025	Oct 25, 2025	[illegible]
shenuri@gmail.com	✉	Oct 25, 2025	Oct 25, 2025	[illegible]
weroshprofy@gmail.com	✉	Oct 25, 2025	Oct 25, 2025	[illegible]
hansamal@gmail.com	✉	Oct 24, 2025	Oct 24, 2025	[illegible]
shamsera@gmail.com	✉	Oct 24, 2025	Oct 24, 2025	[illegible]
mayuru@gmail.com	✉	Oct 24, 2025	Oct 24, 2025	[illegible]
testfinder@email.com	✉	Oct 24, 2025	Oct 24, 2025	[illegible]
tahiru@gmail.com	✉	Oct 24, 2025	Oct 24, 2025	[illegible]

Figure 0-11 Firebase Authentication Console

Admin Dashboard

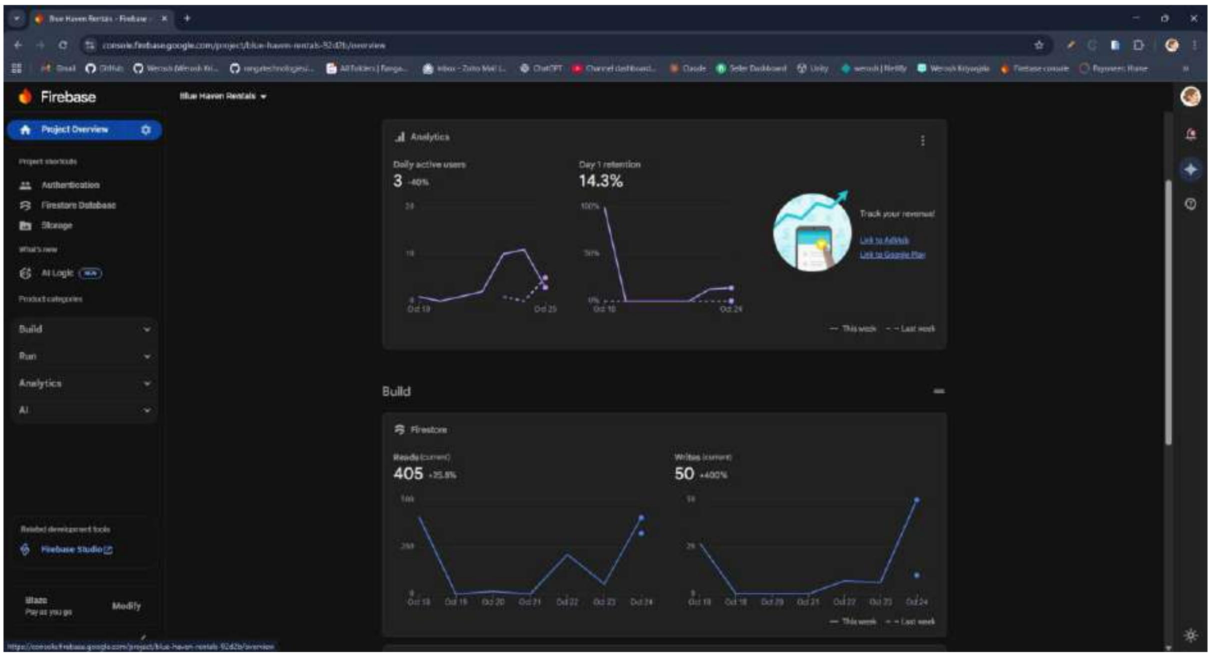


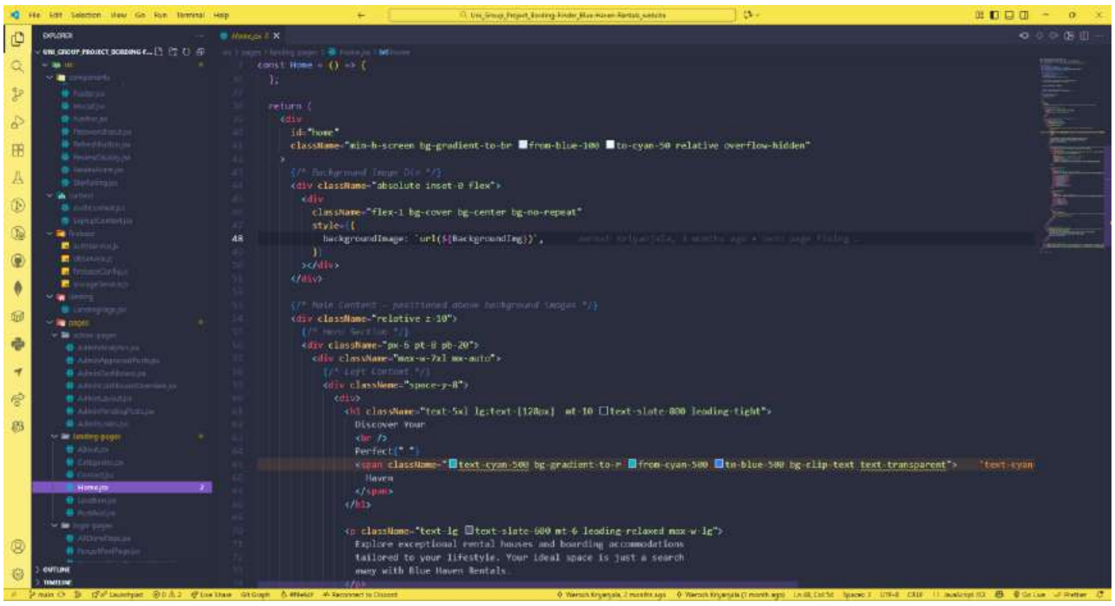
Figure 0-12 Admin Dashboard

5.4. Code Segment

This section highlights important parts of the implementation. That provide the logic and functionality needed to support the system's features and capabilities. Here important code segments for front end and back end are included.

Front-end

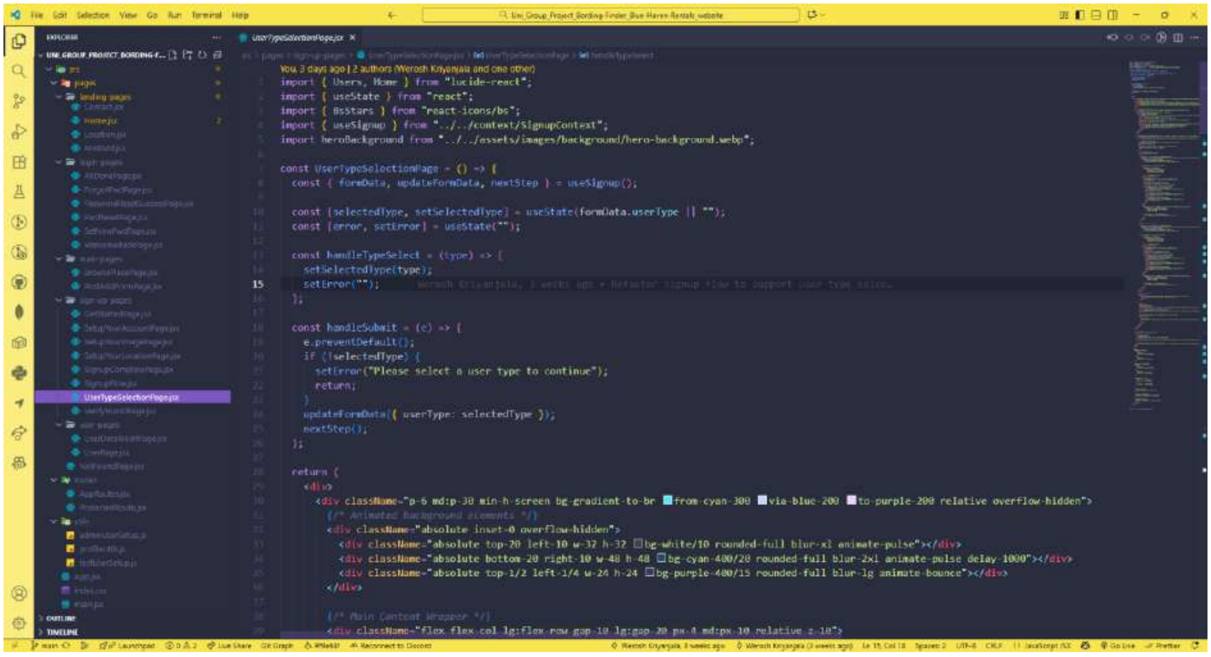
Home Page



```
const Home = () => {
  };

  return (
    <div
      id="home"
      className="min-h-screen bg-gradient-to-br from-blue-100 to-cyan-50 relative overflow-hidden"
    >
      {/* Background Image Div */}
      <div className="absolute inset-0 flex">
        <div
          className="flex-1 bg-cover bg-center bg-no-repeat"
          style={{
            backgroundImage: `url(${BackgroundImg})`,
          }}
        ></div>
      </div>

      {/* Main Content - positioned above background images */}
      <div className="relative z-10">
        {/* Hero Section */}
        <div className="px-6 pt-8 pb-20">
          <div className="max-w-7xl mx-auto">
            {/* Left Content */}
            <div className="space-y-8">
              <div>
                <h1 className="text-5xl lg:text-[128px]  mt-10 text-slate-800 leading-tight">
                  Discover Your
                  <br />
                  Perfect{" "}
                  <span className="text-cyan-500 bg-gradient-to-r from-cyan-500 to-blue-500 bg-clip-text text-transparent">
                    Haven
                  </span>
                </h1>

                <p className="text-lg text-slate-600 mt-6 leading-relaxed max-w-lg">
                  Explore exceptional rental houses and boarding accommodations
                  tailored to your lifestyle. Your ideal space is just a search
                  away with Blue Haven Rentals.
                </p>
```

Figure 0-13 Home Page

User Registration Page

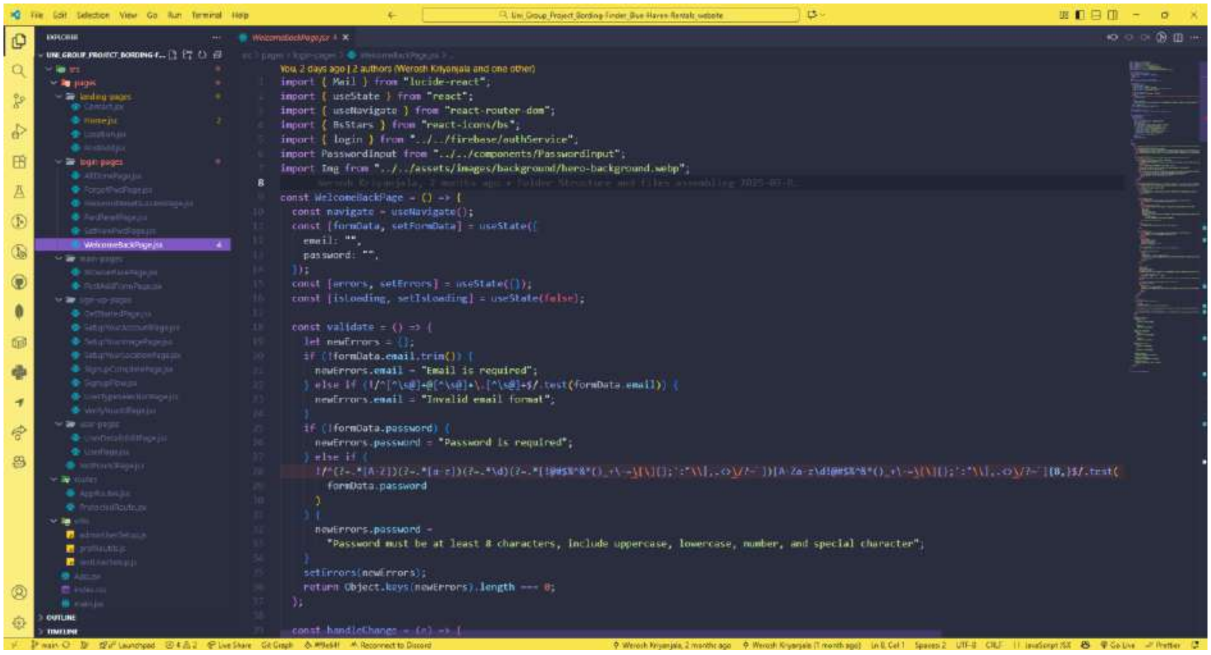


UserTypeSelectionPage.jsx

```
You, 3 days ago | 2 authors (Werosh Kriyanjala and one other)
import { Users, Home } from "lucide-react";
import { useState } from "react";
import { BsStars } from "react-icons/bs";
import { useSignup } from "../../context/SignupContext";
import heroBackground from "../../assets/images/background/hero-background.webp";

const UserTypeSelectionPage = () => {
  const { formData, updateFormData, nextStep } = useSignup();

  const [selectedType, setSelectedType] = useState(formData.userType || "");
  const [error, setError] = useState("");

  const handleTypeSelect = (type) => {
    setSelectedType(type);
    setError("");
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    if (!selectedType) {
      setError("Please select a user type to continue");
      return;
    }
    updateFormData({ userType: selectedType });
    nextStep();
  };

  return (
    <div>
      <div className="p-6 md:p-30 min-h-screen bg-gradient-to-br from-cyan-300 via-blue-200 to-purple-200 relative overflow-hidden">
        {/* Animated background elements */}
        <div className="absolute inset-0 overflow-hidden">
          <div className="absolute top-20 left-10 w-32 h-32 bg-white/10 rounded-full blur-xl animate-pulse"></div>
          <div className="absolute bottom-20 right-10 w-48 h-48 bg-cyan-400/20 rounded-full blur-2xl animate-pulse delay-1000"></div>
          <div className="absolute top-1/2 left-1/4 w-24 h-24 bg-purple-400/15 rounded-full blur-lg animate-bounce"></div>
        </div>

        {/* Main Content Wrapper */}
        <div className="flex flex-col lg:flex-row gap-10 lg:gap-20 px-4 md:px-10 relative z-10">
```

Figure 0-14 User Registration Page

Login Page

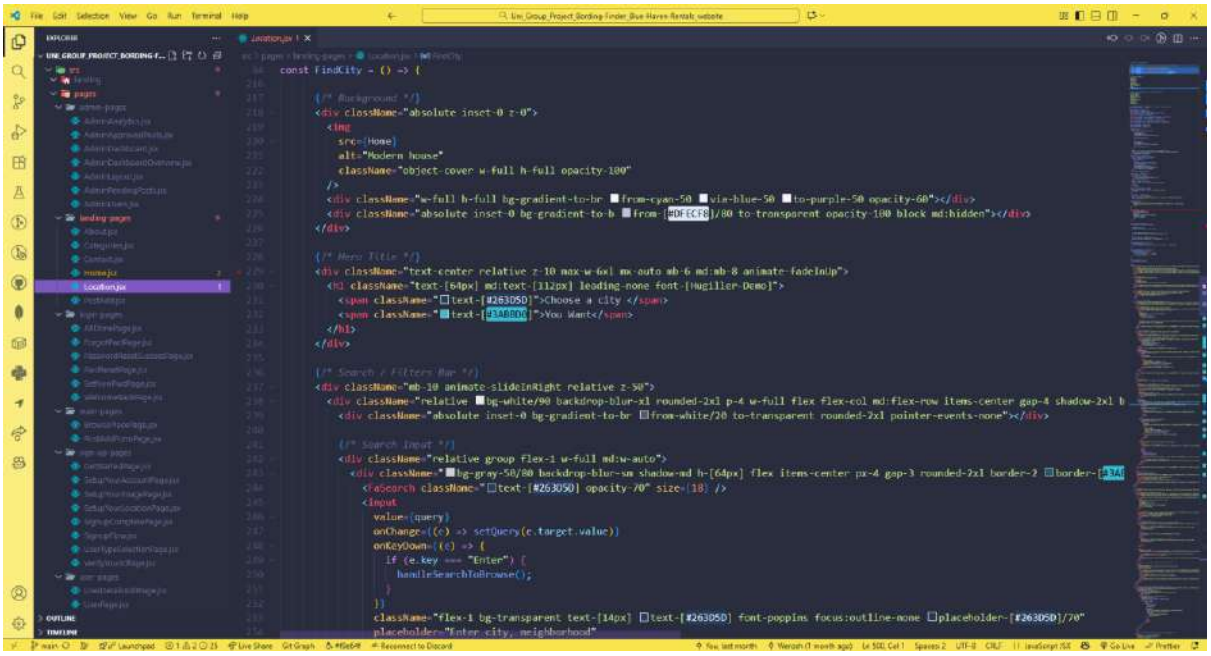


WelcomeBackPage.jsx

```
You, 2 days ago | 2 authors (Werosh Kriyanjala and one other)
import { Mail } from "lucide-react";
import { useState } from "react";
import { useNavigate } from "react-router-dom";
import { BsStars } from "react-icons/bs";
import { login } from "../../firebase/authService";
import PasswordInput from "../../components/PasswordInput";
import Img from "../../assets/images/background/hero-background.webp";

const WelcomeBackPage = () => {
  const navigate = useNavigate();
  const [formData, setFormData] = useState({
    email: "",
    password: "",
  });
  const [errors, setErrors] = useState({});
  const [isLoading, setIsLoading] = useState(false);

  const validate = () => {
    let newErrors = {};
    if (!formData.email.trim()) {
      newErrors.email = "Email is required";
    } else if (!/^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(formData.email)) {
      newErrors.email = "Invalid email format";
    }
    if (!formData.password) {
      newErrors.password = "Password is required";
    } else if (
      !/^(?=.*[A-Z])(?=.*[a-z])(?=.*\d)(?=.*[!@#$%^&*()_+\-=\[\]{};':"\\|,.<>\/?~])[A-Za-z\d!@#$%^&*()_+\-=\[\]{};':"\\|,.<>\/?~]{8,}$/.test(
        formData.password
      )
    ) {
      newErrors.password =
        "Password must be at least 8 characters, include uppercase, lowercase, number, and special character";
    }
    setErrors(newErrors);
    return Object.keys(newErrors).length === 0;
  };

  const handleChange = (e) => {
```

Figure 0-15 Login Page

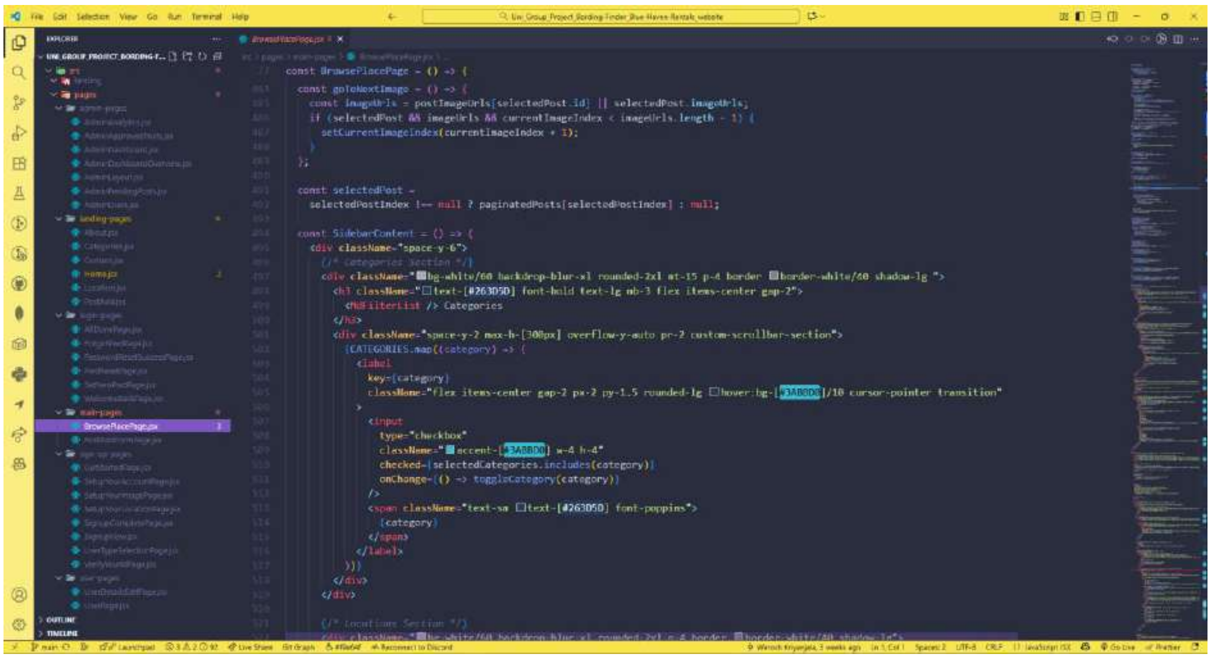
Search & Filter Page



```
const FindCity = () => {

      {/* Background */}
      <div className="absolute inset-0 z-0">
        <img
          src={Home}
          alt="Modern house"
          className="object-cover w-full h-full opacity-100"
        />
        <div className="w-full h-full bg-gradient-to-br from-cyan-50 via-blue-50 to-purple-50 opacity-60"></div>
        <div className="absolute inset-0 bg-gradient-to-b from-[#DFECF8]/80 to-transparent opacity-100 block md:hidden"></div>
      </div>

      {/* Menu Title */}
      <div className="text-center relative z-10 max-w-6xl mx-auto mb-6 md:mb-8 animate-fadeInUp">
        <h1 className="text-[64px] md:text-[112px] leading-none font-[Mugiller-Demo]">
          <span className="text-[#263D5D]">Choose a city </span>
          <span className="text-[#3ABBD0]">You Want</span>
        </h1>
      </div>

      {/* Search / Filters Bar */}
      <div className="mb-10 animate-slideInRight relative z-50">
        <div className="relative bg-white/90 backdrop-blur-xl rounded-2xl p-4 w-full flex flex-col md:flex-row items-center gap-4 shadow-2xl b
          <div className="absolute inset-0 bg-gradient-to-br from-white/20 to-transparent rounded-2xl pointer-events-none"></div>

          {/* Search Input */}
          <div className="relative group flex-1 w-full md:w-auto">
            <div className="bg-gray-50/80 backdrop-blur-sm shadow-md h-[64px] flex items-center px-4 gap-3 rounded-2xl border-2 border-[#3A
              <FaSearch className="text-[#263D5D] opacity-70" size={18} />
              <input
                value={query}
                onChange={(e) => setQuery(e.target.value)}
                onKeyDown={(e) => {
                  if (e.key === "Enter") {
                    handleSearchToBrowse();
                  }
                }}
                className="flex-1 bg-transparent text-[14px] text-[#263D5D] font-poppins focus:outline-none placeholder-[#263D5D]/70"
                placeholder="Enter city, neighborhood"
```

Figure 0-16 Search & Filter Page

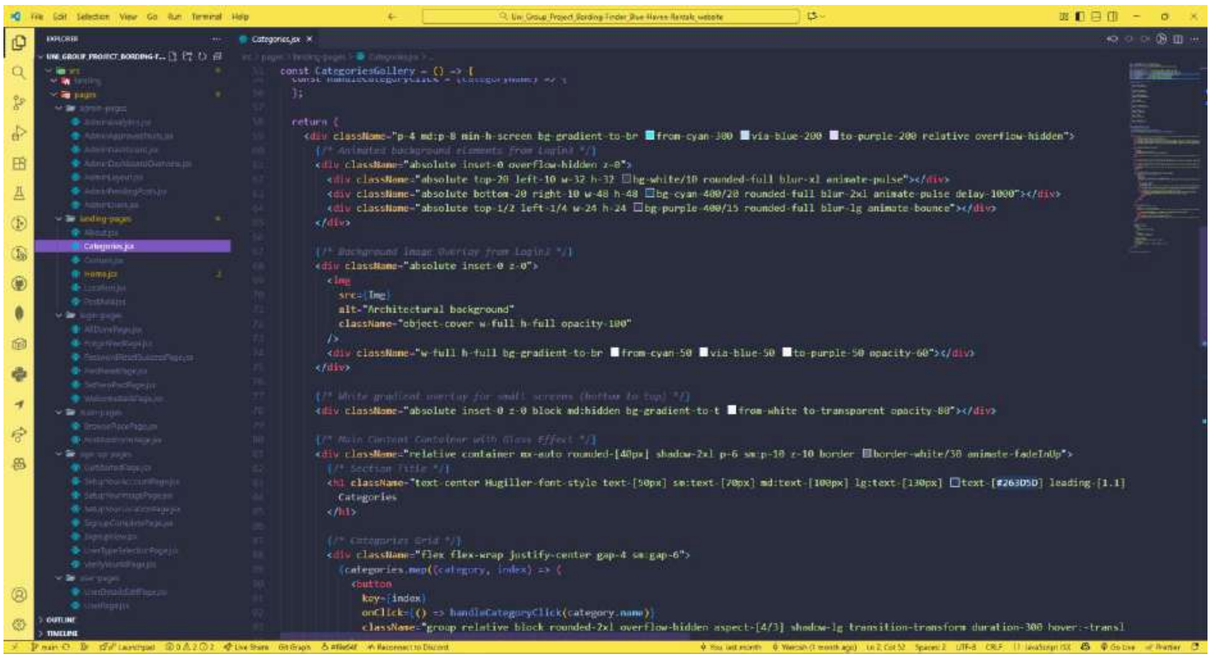
Search Results Page



```
const BrowsePlacePage = () => {
  const goToNextImage = () => {
    const imageUrls = postImageUrls[selectedPost.id] || selectedPost.imageUrls;
    if (selectedPost && imageUrls && currentImageIndex < imageUrls.length - 1) {
      setCurrentImageIndex(currentImageIndex + 1);
    }
  };

  const selectedPost =
    selectedPostIndex !== null ? paginatedPosts[selectedPostIndex] : null;

  const SidebarContent = () => (
    <div className="space-y-6">
      {/* Categories Section */}
      <div className="bg-white/60 backdrop-blur-xl rounded-2xl mt-15 p-4 border border-white/60 shadow-lg ">
        <h3 className="text-[#263D5D] font-bold text-lg mb-3 flex items-center gap-2">
          <MdFilterList /> Categories
        </h3>
        <div className="space-y-2 max-h-[300px] overflow-y-auto pr-2 custom-scrollbar-section">
          {CATEGORIES.map((category) => (
            <label
              key={category}
              className="flex items-center gap-2 px-2 py-1.5 rounded-lg hover:bg-[#3ABBD0]/10 cursor-pointer transition"
            >
              <input
                type="checkbox"
                className="accent-[#3ABBD0] w-4 h-4"
                checked={selectedCategories.includes(category)}
                onChange={() => toggleCategory(category)}
              />
              <span className="text-sm text-[#263D5D] font-poppins">
                {category}
              </span>
            </label>
          ))}
        </div>
      </div>

      {/* Locations Section */}
```

Figure 0-17 Search Results Page

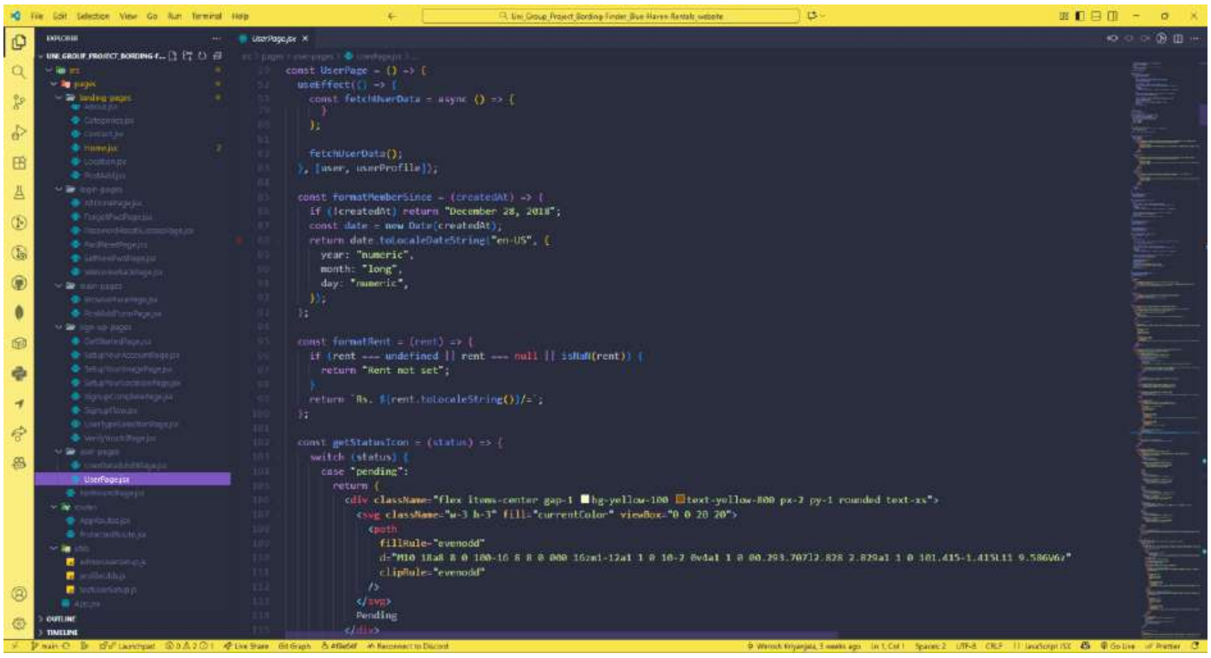
Boarding Details Page



```
const CategoriesGallery = () => {
  };

  return (
    <div className="p-4 md:p-8 min-h-screen bg-gradient-to-br from-cyan-300 via-blue-200 to-purple-200 relative overflow-hidden">
      {/* Animated background elements from Login1 */}
      <div className="absolute inset-0 overflow-hidden z-0">
        <div className="absolute top-20 left-10 w-32 h-32 bg-white/10 rounded-full blur-xl animate-pulse"></div>
        <div className="absolute bottom-20 right-10 w-48 h-48 bg-cyan-400/20 rounded-full blur-2xl animate-pulse delay-1000"></div>
        <div className="absolute top-1/2 left-1/4 w-24 h-24 bg-purple-400/15 rounded-full blur-lg animate-bounce"></div>
      </div>

      {/* Background Image Overlay from Login1 */}
      <div className="absolute inset-0 z-0">
        <img
          src={Img}
          alt="Architectural background"
          className="object-cover w-full h-full opacity-100"
        />
        <div className="w-full h-full bg-gradient-to-br from-cyan-50 via-blue-50 to-purple-50 opacity-60"></div>
      </div>

      {/* White gradient overlay for small screens (bottom to top) */}
      <div className="absolute inset-0 z-0 block md:hidden bg-gradient-to-t from-white to-transparent opacity-80"></div>

      {/* Main Content Container with Glass Effect */}
      <div className="relative container mx-auto rounded-[40px] shadow-2xl p-6 sm:p-10 z-10 border border-white/30 animate-fadeInUp">
        {/* Section Title */}
        <h1 className="text-center Hugiller-font-style text-[50px] sm:text-[70px] md:text-[100px] lg:text-[130px] text-[#263D5D] leading-[1.1]
          Categories
        </h1>

        {/* Categories Grid */}
        <div className="flex flex-wrap justify-center gap-4 sm:gap-6">
          {categories.map((category, index) => (
            <button
              key={index}
              onClick={() => handleCategoryClick(category.name)}
              className="group relative block rounded-2xl overflow-hidden aspect-[4/3] shadow-lg transition-transform duration-300 hover:-transl
```

Figure 0-18 Boarding Details Page

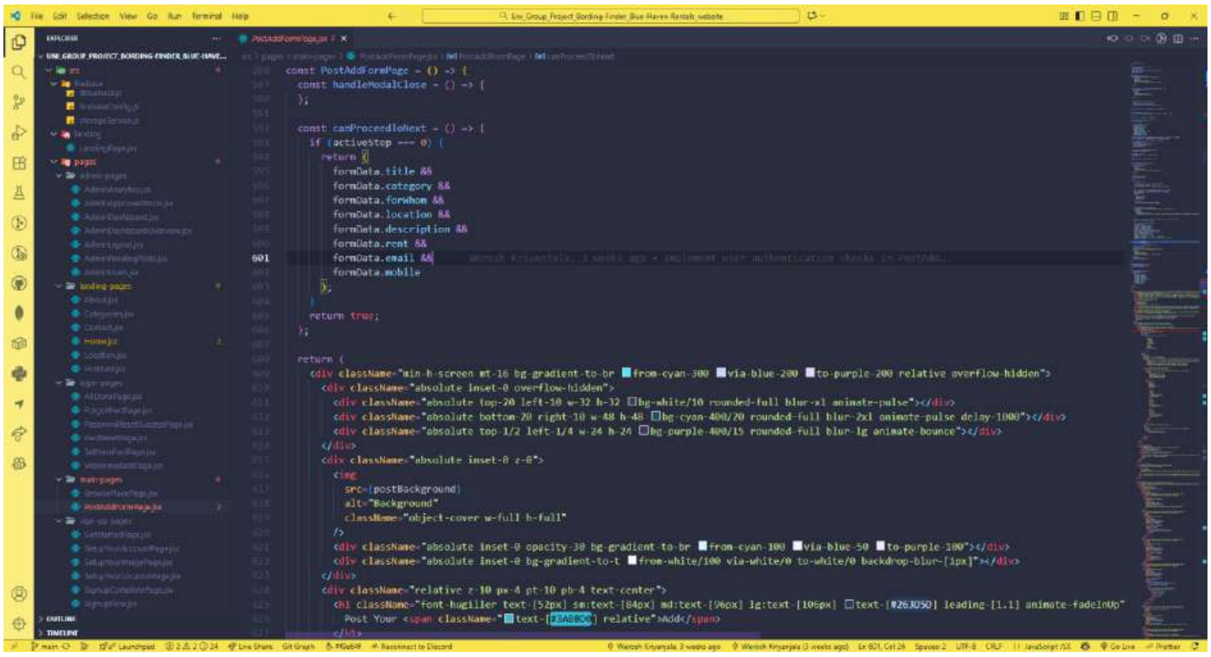
Owner Dashboard



```
const UserPage = () => {
  useEffect(() => {
    const fetchUserData = async () => {
      }
    };

    fetchUserData();
  }, [user, userProfile]);

  const formatMemberSince = (createdAt) => {
    if (!createdAt) return "December 28, 2018";
    const date = new Date(createdAt);
    return date.toLocaleDateString("en-US", {
      year: "numeric",
      month: "long",
      day: "numeric",
    });
  };

  const formatRent = (rent) => {
    if (rent === undefined || rent === null || isNaN(rent)) {
      return "Rent not set";
    }
    return `Rs. ${rent.toLocaleString()}/=`;
  };

  const getStatusIcon = (status) => {
    switch (status) {
      case "pending":
        return (
          <div className="flex items-center gap-1 bg-yellow-100 text-yellow-800 px-2 py-1 rounded text-xs">
            <svg className="w-3 h-3" fill="currentColor" viewBox="0 0 20 20">
              <path
                fillRule="evenodd"
                d="M10 18a8 8 0 100-16 8 8 0 000 16zm1-12a1 1 0 10-2 0v4a1 1 0 00.293.707l2.828 2.829a1 1 0 101.415-1.415L11 9.586V6z"
                clipRule="evenodd"
              />
            </svg>
            Pending
          </div>
```

Figure 0-19 Owner Dashboard

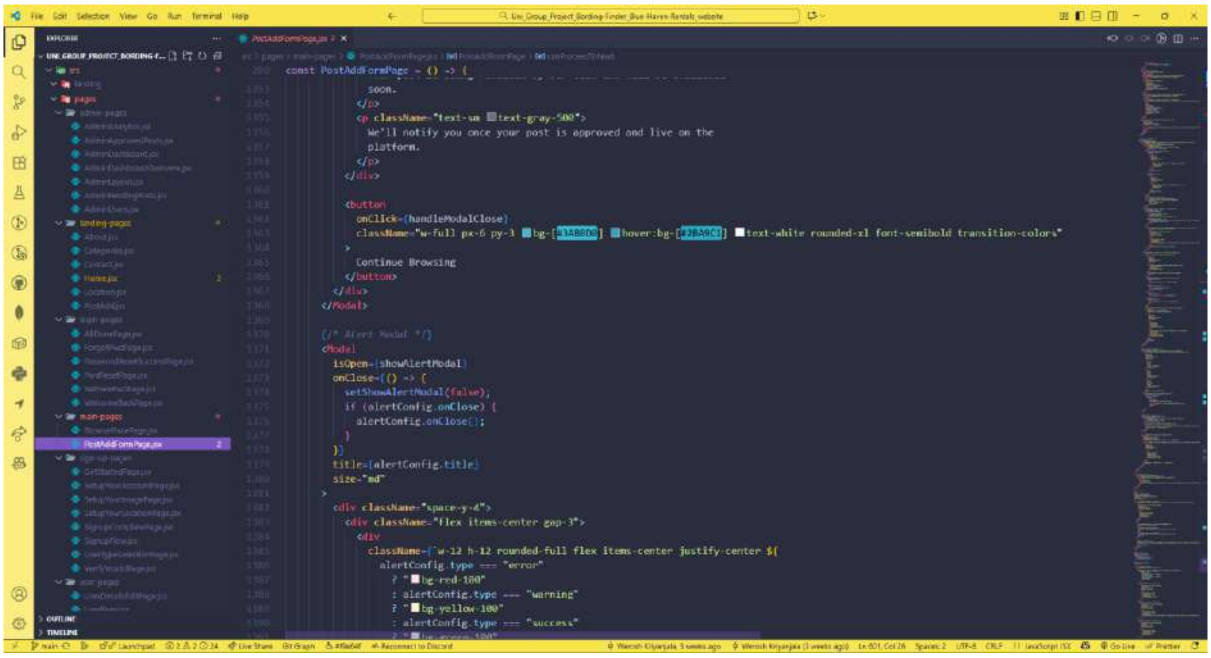
Add New Listing Form



```
const PostAddFormPage = () => {
  const handleModalClose = () => {
  };

  const canProceedToNext = () => {
    if (activeStep === 0) {
      return (
        formData.title &&
        formData.category &&
        formData.forWhom &&
        formData.location &&
        formData.description &&
        formData.rent &&
        formData.email &&
        formData.mobile
      );
    }
    return true;
  };

  return (
    <div className="min-h-screen mt-16 bg-gradient-to-br from-cyan-300 via-blue-200 to-purple-200 relative overflow-hidden">
      <div className="absolute inset-0 overflow-hidden">
        <div className="absolute top-20 left-10 w-32 h-32 bg-white/10 rounded-full blur-xl animate-pulse"></div>
        <div className="absolute bottom-20 right-10 w-48 h-48 bg-cyan-400/20 rounded-full blur-2xl animate-pulse delay-1000"></div>
        <div className="absolute top-1/2 left-1/4 w-24 h-24 bg-purple-400/15 rounded-full blur-lg animate-bounce"></div>
      </div>
      <div className="absolute inset-0 z-0">
        <img
          src={postBackground}
          alt="Background"
          className="object-cover w-full h-full"
        />
        <div className="absolute inset-0 opacity-30 bg-gradient-to-br from-cyan-100 via-blue-50 to-purple-100"></div>
        <div className="absolute inset-0 bg-gradient-to-t from-white/100 via-white/0 to-white/0 backdrop-blur-[1px]"></div>
      </div>
      <div className="relative z-10 px-4 pt-10 pb-4 text-center">
        <h1 className="font-hugiller text-[52px] sm:text-[84px] md:text-[96px] lg:text-[106px] text-[#263D50] leading-[1.1] animate-fadeInUp"
          Post Your <span className="text-[#3AB8DC] relative">Add</span>
```

Figure 0-20 Add New Listing Form

Feedback Form

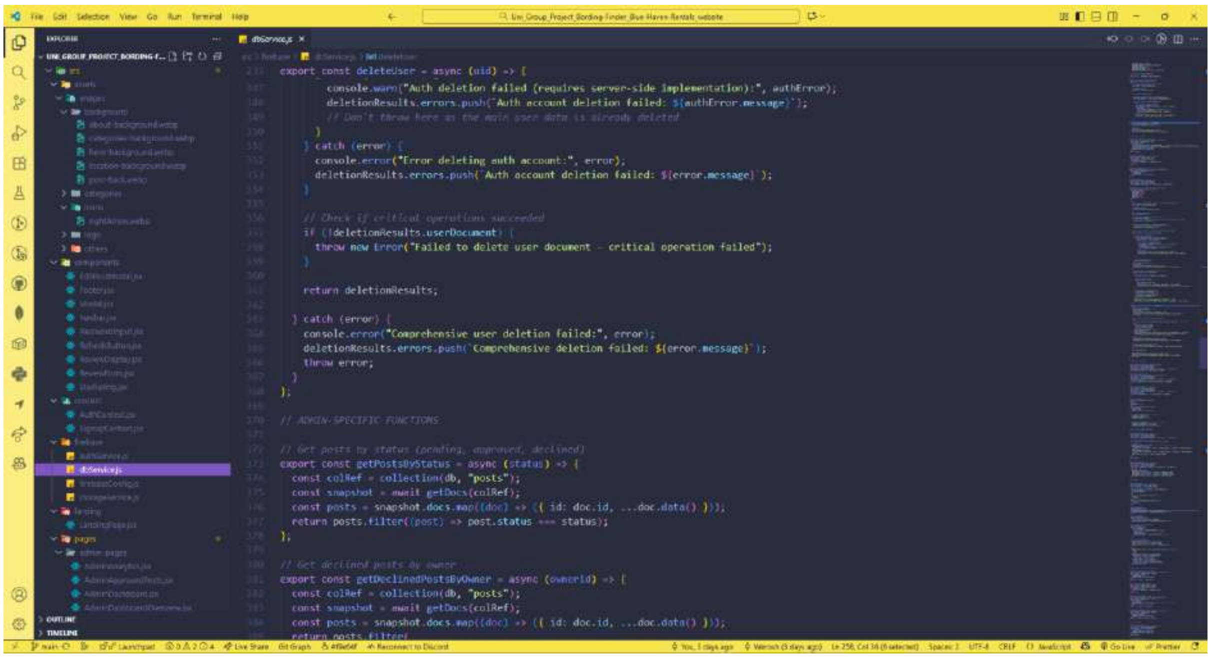


```
const PostAddFormPage = () => {
                  soon.
                </p>
                <p className="text-sm text-gray-500">
                  We'll notify you once your post is approved and live on the
                  platform.
                </p>
              </div>

              <button
                onClick={handleModalClose}
                className="w-full px-6 py-3 bg-[#3AB8D0] hover:bg-[#2BA9C1] text-white rounded-xl font-semibold transition-colors"
              >
                Continue Browsing
              </button>
            </div>
          </Modal>

          {/* Alert Modal */}
          <Modal
            isOpen={showAlertModal}
            onClose={() => {
              setShowAlertModal(false);
              if (alertConfig.onClose) {
                alertConfig.onClose();
              }
            }}
            title={alertConfig.title}
            size="md"
          >
            <div className="space-y-4">
              <div className="flex items-center gap-3">
                <div
                  className={`w-12 h-12 rounded-full flex items-center justify-center ${
                    alertConfig.type === "error"
                      ? "bg-red-100"
                      : alertConfig.type === "warning"
                      ? "bg-yellow-100"
                      : alertConfig.type === "success"
```

Figure 0-21 Feedback Form

Back-end

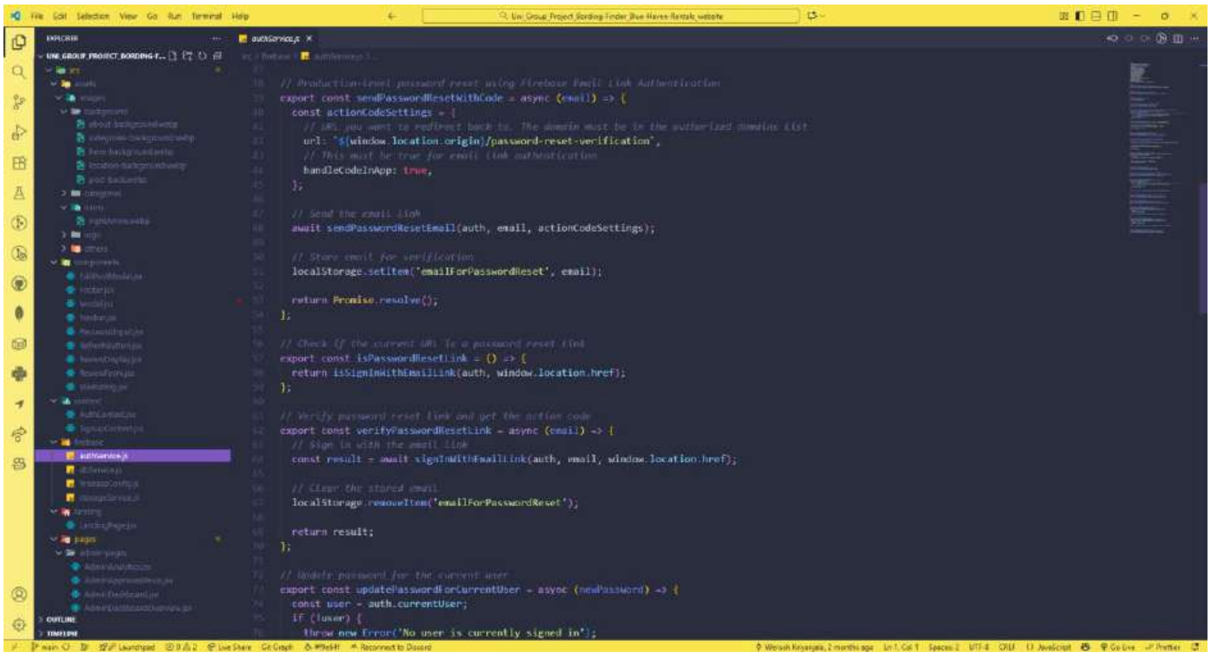
Firebase Firestore Database View



```
export const deleteUser = async (uid) => {
        console.warn("Auth deletion failed (requires server-side implementation):", authError);
        deletionResults.errors.push(`Auth account deletion failed: ${authError.message}`);
        // Don't throw here as the main user data is already deleted
      }
    } catch (error) {
      console.error("Error deleting auth account:", error);
      deletionResults.errors.push(`Auth account deletion failed: ${error.message}`);
    }

    // Check if critical operations succeeded
    if (!deletionResults.userDocument) {
      throw new Error("Failed to delete user document - critical operation failed");
    }

    return deletionResults;

  } catch (error) {
    console.error("Comprehensive user deletion failed:", error);
    deletionResults.errors.push(`Comprehensive deletion failed: ${error.message}`);
    throw error;
  }
};

// ADMIN-SPECIFIC FUNCTIONS

// Get posts by status (pending, approved, declined)
export const getPostsByStatus = async (status) => {
  const colRef = collection(db, "posts");
  const snapshot = await getDocs(colRef);
  const posts = snapshot.docs.map((doc) => ({ id: doc.id, ...doc.data() }));
  return posts.filter((post) => post.status === status);
};

// Get declined posts by owner
export const getDeclinedPostsByOwner = async (ownerId) => {
  const colRef = collection(db, "posts");
  const snapshot = await getDocs(colRef);
  const posts = snapshot.docs.map((doc) => ({ id: doc.id, ...doc.data() }));
  return posts.filter(
```

Figure 0-22 Firebase Firestore Database View

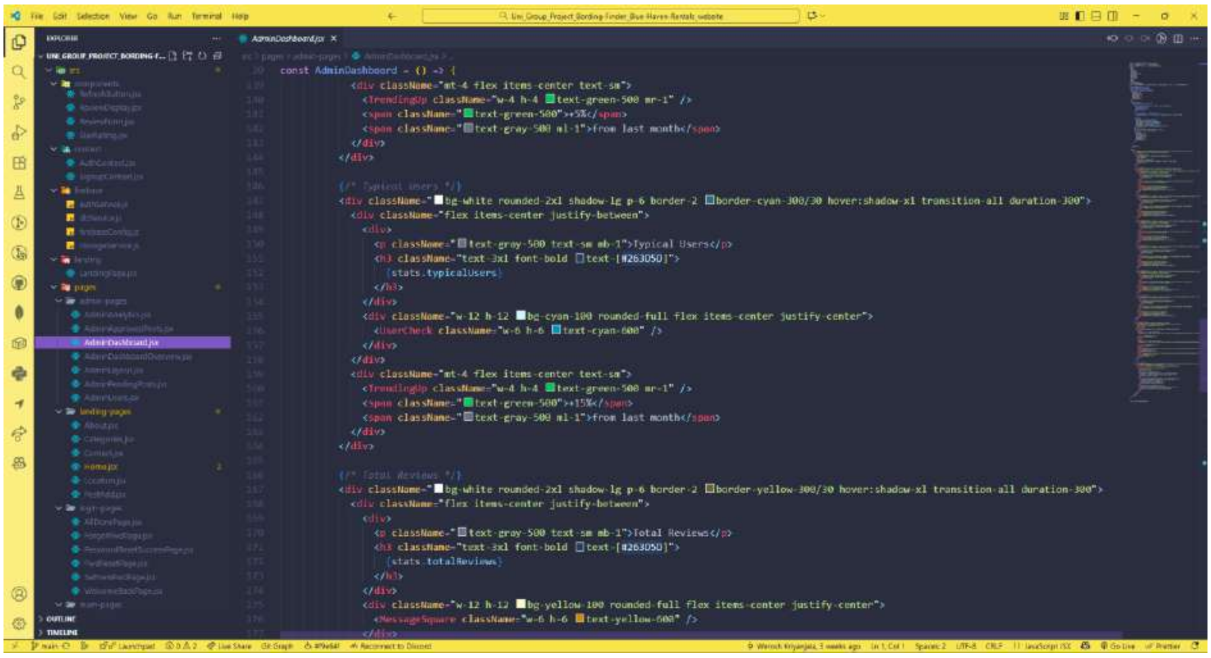
Firebase Authentication Console



```
// Production-level password reset using Firebase Email Link Authentication
export const sendPasswordResetWithCode = async (email) => {
  const actionCodeSettings = {
    // URL you want to redirect back to. The domain must be in the authorized domains list
    url: `${window.location.origin}/password-reset-verification`,
    // This must be true for email link authentication
    handleCodeInApp: true,
  };

  // Send the email link
  await sendPasswordResetEmail(auth, email, actionCodeSettings);

  // Store email for verification
  localStorage.setItem('emailForPasswordReset', email);

  return Promise.resolve();
};

// Check if the current URL is a password reset link
export const isPasswordResetLink = () => {
  return isSignInWithEmailLink(auth, window.location.href);
};

// Verify password reset link and get the action code
export const verifyPasswordResetLink = async (email) => {
  // Sign in with the email link
  const result = await signInWithEmailLink(auth, email, window.location.href);

  // Clear the stored email
  localStorage.removeItem('emailForPasswordReset');

  return result;
};

// Update password for the current user
export const updatePasswordForCurrentUser = async (newPassword) => {
  const user = auth.currentUser;
  if (!user) {
    throw new Error('No user is currently signed in');
```

Figure 0-23 Firebase Authentication Console

Admin Dashboard



```
const AdminDashboard = () => {
          <div className="mt-4 flex items-center text-sm">
            <TrendingUp className="w-4 h-4 text-green-500 mr-1" />
            <span className="text-green-500">+5%</span>
            <span className="text-gray-500 ml-1">from last month</span>
          </div>
        </div>

        {/* Typical Users */}
        <div className="bg-white rounded-2xl shadow-lg p-6 border-2 border-cyan-300/30 hover:shadow-xl transition-all duration-300">
          <div className="flex items-center justify-between">
            <div>
              <p className="text-gray-500 text-sm mb-1">Typical Users</p>
              <h3 className="text-3xl font-bold text-[#263D5D]">
                {stats.typicalUsers}
              </h3>
            </div>
            <div className="w-12 h-12 bg-cyan-100 rounded-full flex items-center justify-center">
              <UserCheck className="w-6 h-6 text-cyan-600" />
            </div>
          </div>
          <div className="mt-4 flex items-center text-sm">
            <TrendingUp className="w-4 h-4 text-green-500 mr-1" />
            <span className="text-green-500">+15%</span>
            <span className="text-gray-500 ml-1">from last month</span>
          </div>
        </div>

        {/* Total Reviews */}
        <div className="bg-white rounded-2xl shadow-lg p-6 border-2 border-yellow-300/30 hover:shadow-xl transition-all duration-300">
          <div className="flex items-center justify-between">
            <div>
              <p className="text-gray-500 text-sm mb-1">Total Reviews</p>
              <h3 className="text-3xl font-bold text-[#263D5D]">
                {stats.totalReviews}
              </h3>
            </div>
            <div className="w-12 h-12 bg-yellow-100 rounded-full flex items-center justify-center">
              <MessageSquare className="w-6 h-6 text-yellow-600" />
            </div>
```

Figure 0-24 Admin Dashboard

5.5. Summary

According to the above chapter user interfaces include the front-end interface, which is the part of the system that users interact with directly, as well as the back-end interface, which is used by administrators to manage and maintain the system. Codes segments describe in the chapter are blocks of code that perform main functions in the system.

In next chapter testing is the process of verifying that the system functions as intended, while evaluation is the process of assessing the system's performance and usability.

Chapter 06 – Testing and Evaluation

6.1. Introduction

Testing and evaluation are essential part of the software development life cycle that guarantee the correctness, success, and dependability of the system.

In the case of Blue Heaven Rentals, various testing procedures were used to ensure that all the functions including user registration, user logging in, search, listing management and user management as an administrator worked properly and in a consistent manner.

All the modules were put to test with the help of both individual testing and testing with the rest of the system to ensure that they integrate well with the real time back-end of Firebase.

This chapter explains the chief categories of software testing that can be performed, and the findings derived by systematic testing in case of an evaluation of the test cases.

6.2. Type of Software Testing

6.2.1. Unit Testing

Unit testing focuses on the functionality testing of a system to ensure that the units or parts of the system work. Each component in Blue Heaven Rentals and Firebase function was tested in isolation, to confirm the component works properly in isolation.

Indicatively, the registration and login modules would be tested by entering correct and incorrect data to ensure that Firebase authentication was managing the two appropriately.

Also, the AddListing() and fetchListings functions were tested separately to make sure that the data was properly added to and withdrawn out of Firestore.

This also assisted in detecting and correcting initial problems like false data validation, fields and delays in responses prior to integration.

6.2.2. Black Box Testing

The black box testing was done to test the external functionality of the system without the need to look at the internal code structure.

It was concerned with ensuring that every feature was generating the right output under given input circumstances.

In this case, an example of this is that when the user keyed in boarding houses in the city of Kandy, the system should have displayed all the listings within this city. Equally, once wrong details were typed in the logs, the system would present an error.

This method was used to make sure that the system acted as desired and verified that interfaces reacted appropriately to different actions of the user.

6.2.3. White Box Testing

White box testing was used to verify the internal logic of system.

This consisted of verification of condition flow and making sure that every conceivable code path had been run to completion successfully.

The console logs and Firebase Emulator tools were used to test internal Firebase processes, including identification of user roles, and update conditions of lists in Blue Heaven Rentals.

It enabled developers to confirm the correct functionality of the backend logic of ownership verification, modification of Firestore documents, and presentation of filtered listings.

It was also used to guarantee that no redundant and unreachable code paths were left in the final implementation.

6.2.4. Integration Testing

Integration testing was conducted to ensure that the system modules interaction went well.

After the modules were tested unitarily (such as registration, listing management and search), they were integrated into one another and implemented as a whole workflow.

As an example, an integration test was done to ensure that when an owner inserted a new listing, the new data was saved into Firestore, immediately shown in the search page, and was still edible via the owner dashboard.

This testing confirmed that each of the modules was able to communicate with each other using Firebase and that the frontend-backend linkage was a consistency without failure.

6.2.5. Security Testing

Security testing was conducted to verify that Blue Heaven Rentals was in charge of data integrity as it ensured privacy of the user.

The Firebase Authentication and HTTPS were tested to avoid the possibility of leakage of data and unauthorized access.

Attackers tried to get access to the restricted areas without the required login credentials, which was an attempt to gain access to the admin panel area, and the system redirected the unauthorized user to the login system.

This has verified that Firebase security setting and access control policies were installed properly.

To further ensure secure transfer of data between clients and the server all the database communications were checked to ensure that it was encrypted.

6.2.6. User Acceptance Testing (UAT)

User Acceptance Testing (UAT) has been done with a group of real users that were selected to use the system, who are the target audience of the system and they include university students, owners of boarding houses and working professionals.

The users were requested to engage with the system, register, use the search, and contact the owner and leave a feedback using a Google Form.

The outcomes were very positive: the system appeared easy, responsive and efficient in locating checked boarding locations to the user.

The highlight of the application that most users liked was the real-time update feature, and the application interface was easy to comprehend.

Such testing ensured that the system was as per the expectations of the users and attained the intended goals.

6.3. Test Cases

Having completed all kinds of testing, the following significant test cases were carried out to confirm the functionality of the system and the quality of results.

Project Details		**Project Name:**	Blue Heaven Rentals							
		Module Name:	**User Registration**							
		Created By:								
		Received By:								
		Created Date:			**Received Date:**					
Test Case ID	**Test Description/Test case name**	**Prerequisite**	**Test Steps**	**Input Data**	**Expected Result**	**Actual Result**	**Status**	**Severity**	**Priority**	**Executed by**
TC-UR-001	Register New User	Browser opened, Internet connected	1. Navigate to Feedback page 2. Enter details 3. Click submit	Name, Rating, Comment	Feedback Saved and confirmation message displayed	As expected	Pass	Low	Medium	Githmi
TC-UR-002	Register with existing email	User already exists	1. Enter same email as an existing user 2. Submit form	Email = test@gmail.com	System shows “Email already in use ”	As expected	Pass	Low	Medium	Githmi
TC-UR-003	Register with invalid email	None	1. Enter invalid email format 2. Submit	Email = test@	Error message “ Invalid email ”	As expected	Pass	Low	Medium	Githmi

TC-UR-004	Register with missing fields	None	1. Leave password field empty 2. Submit form	Email = user@gmail.com	Validation message " All fields are required"	As expected	Pass	Low	High	Githmi

Table 6.3-0-1 User Registration use case

Project Details		**Project Name:**	Blue Heaven Rentals							
		Module Name:	**User Login**							
		Created By:								
		Received By:								
		Created Date:			**Received Date:**					
Test Case ID	**Test Description/Test case name**	**Prerequisite**	**Test Steps**	**Input Data**	**Expected Result**	**Actual Result**	**Status**	**Severity**	**Priority**	**Executed by**
TC-UL-01	Login with valid credentials	User account created	1. Go to login page 2. Enter valid email & password 3. Click login	Email & Password	User redirected to dashboard	As expected	Pass	High	High	Githmi
TC-UL-02	Login with invalid credentials	None	1. Enter incorrect password 2. Click login	Email = user@gmail.com, Password = 123	System displays error “Invalid credentials”	As expected	Pass	Low	Medium	Githmi
TC-UL-03	Login with empty fields	None	1. Leave email empty 2. Click login	Empty email	Validation message “Please enter your email”	As expected	Pass	Low	Medium	Githmi

TC-UL-04	Logout function	User logged in	1. Click logout 2. Confirm logout	None	User session cleared, redirected to login page	As expected	Pass	Medium	High	Githmi

Table 6.3-0-2 User Login Use case

Project Details		**Project Name:**	Blue Heaven Rentals							
		Module Name:	**Boarding House Reservation**							
		Created By:								
		Received By:								
		Created Date:			**Received Date:**					
Test Case ID	**Test Description/Test case name**	**Prerequisite**	**Test Steps**	**Input Data**	**Expected Result**	**Actual Result**	**Status**	**Severity**	**Priority**	**Executed by**
TC-BR-01	Search available boarding houses	User logged in	1. Enter location “Colombo” 2. Click search	Location = Colombo	List of available boarding displayed	As expected	Pass	Medium	High	Githmi
TC-BR-02	View boarding details	Search completed	1. Select one boarding from search results	Boarding ID	Full details shown (photos, rent, owner info)	As expected	Pass	Low	Medium	Githmi
TC-BR-03	Contact owner	Boarding details open	1. Click “Contact Owner” button 2. Send message	Message = “Is this still available?”	Message delivered successfully	As expected	Pass	Medium	High	Githmi
TC-BR-04	Book unavailable listing	Listing marked unavailable	1. Try to reserve unavailable house	Booking attempt	System shows “This boarding is currently unavailable”	As expected	Pass	Low	Medium	Githmi

Table 6.3-0-3 Boarding House Reservation Use case

Project Details		**Project Name:**	Blue Heaven Rentals							
		Module Name:	**User Feedback Form**							
		Created By:								
		Received By:								
		Created Date:			**Received Date:**					
Test Case ID	**Test Description/Test case name**	**Prerequisite**	**Test Steps**	**Input Data**	**Expected Result**	**Actual Result**	**Status**	**Severity**	**Priority**	**Executed by**
TC-UF-01	Submit feedback successfully	User logged in	1. Navigate to Feedback Page 2. Enter details 3. Click submit	Name, Rating, Comment	Feedback saved and confirmation message displayed	As expected	Pass	Low	Medium	Githmi
TC-UF-02	Submit empty feedback	None	1. Leave comment field blank 2. Submit	Empty comment	Error “Feedback cannot be empty”	As expected	Pass	Low	Medium	Githmi
TC-UF-03	Submit long feedback	None	1. Enter comment > 500 characters 2. Submit	Long comment	Warning “Comment too long”	As expected	Pass	Low	Medium	Githmi

Table 6.3-0-4 User Feedback Form Use Case

Project Details		Project Name:	Blue Heaven Rentals							
		Module Name:	**Add Listing**							
		Created By:								
		Received By:								
		Created Date:			Received Date:					
Test Case ID	**Test Description/Test case name**	**Prerequisite**	**Test Steps**	**Input Data**	**Expected Result**	**Actual Result**	**Status**	**Severity**	**Priority**	**Executed by**
TC-AL-01	Add new listing	Owner logged in	1. Go to “Add Listing” 2. Enter all fields 3. Submit	Title, Rent, Location, Description	Listing added to database and visible on main page	As expected	Pass	High	High	Githmi
TC-AL-02	Add listing with missing fields	None	1. Leave rent field empty 2. Submit	Missing rent	Validation “All fields required”	As expected	Pass	Medium	High	Githmi
TC-AL-03	Edit existing listing	Listing available	1. Go to “My Listings” 2. Edit rent field 3. Save	Rent = 12000	Listing updated successfully	As expected	Pass	Medium	High	Githmi
TC-AL-04	Delete listing	Listing available	1. Click delete icon 2. Confirm	Listing ID	Listing removed from database	As expected	Pass	Medium	Medium	Githmi

Table 6.3-0-5 Add Listing use case

6.4. Summary

This chapter described testing and evaluation procedure that was conducted on Blue Heaven Rentals.

All the functional and non-functional aspects of the system were verified using various methods of testing such as unit, black box, white box, integration, security and user acceptance testing.

The positive outcomes of these tests prove that the system is strong, effective, and complies with all the requirements of the users, and it can be deployed and be applied in reality.

Chapter 07 – Conclusion and Future work

7.1. Introduction

This chapter will be the conclusion of the Blue Heaven Rentals web-based system and work on the results, lessons learned, and the efficiency of the developed solution. It also marks possible improvement of the future and problem encountered during its development stage. The chapter ends with the contribution made by the team members towards the successful project completion.

7.2. Conclusion

The system Blue Heaven Rentals was built and created with the aim of offering a new, quality, and easy to use program to avoid and locate boarding houses in Sri Lanka. It particularly targets the challenges that university students, employees, and job seekers go through because it is very difficult to find an apt boarding place using the traditional means as advertised on newspapers, posted on boards, or social media pages. Such traditional approaches are usually inefficient, time consuming and credibility or real-time deficient.

This web based system has been effective in filling the gap between boarding seekers and owners of boarding houses by means of online centralized platform. Location, price, and facilities can be easily searched and accommodations found by seekers, and the owners can sign up and manage their listings. Firebase Firestore makes sure that the information is updated in real-time so that a user could get the latest data and not wait due to the network territories. In addition, the use of Firebase Authentication software makes this more trustworthy and safe as only registered users can handle listings and interact in the system.

The front end is built using React and JavaScript and provides a user-friendly and responsive interface that can be accessed not only through desktop but also mobile devices to ensure the interface is made accessible to a broad group of users. In the meantime, the backend services in Firebase make data management easy, which makes the system lightweight but powerful. This was developed in phases and phases of analysis, designing, implementation, and testing making sure that there were all the requirements fulfilled in the proper and efficient way.

To sum up, Blue Heaven Rentals achieves its major objective of making the search and operation of a boarding house process easier. It improves access, trust and openness in the matching of seekers and owners and implements a contemporary and technology-driven solution to an ancient social issue. The system can be extended to offer additional features in the future such as integration of payment, AI-suggestions and expansion of mobile application as a developer that will enhance the system's presence as a new solution in the Sri Lankan rental ecosystem.

7.3. Future Work

Although the current version of Blue Heaven Rentals meets fulfills its core objectives, there are several potential improvements to enhance its functionality in future version:

- **Online Payment Integration:** Implement online payment gateway arrangements which are secure, booking and booking confirmation.
- **Mobile Application:** A cross-platform mobile application should be created based on React Native to make accessibility easier.
- **Recommendation System:** Introducing suggestions of AI-driven recommendations to users depending on their interests and past searches
- **User Verification System:** Add document verification to enhance responsibility and credibility of advert.
- **Rating & Review System:** This is a feature that will enable users to rate and review boarding houses to enable them make superior choices.

These improvements will make the systems reliable, usable and relevant in the market.

7.4. Challenged Faced

The team faced the following challenges in the development process:

- Adding real time data synchronization through Firebase Firestore.
- Creating an interface that is responsive and graphically appealing and compatible with various gadgets.
- Efficient time management between the documentation, design, and the coding.
- Training on new tools and transitioning to cloud-based architecture of Firebase.

However, the challenges were successfully resolved because of teamwork, active communication, and problem-solving.

7.5. Contribution of Team Members

Blue Heaven Rentals was successfully developed with equal contributions of all the three members. The implementation was done by each as well as documentation and testing. The contributions were fairly divided to provide equal participation and cooperation in development.

Hatharasinghe AIN	Assisted in the front-end development in React and JavaScript, developed project documentation (SRS and report chapters), and took part in the testing and debugging activities. Assisted in drawing out diagrams like use case diagram and activity diagram as well.
Kriyanjala W	Helped to set up the backend with firebase, firestore database, and authentication modules. Helped in the integration of the frontend and user interface design and documentation preparation. Participated in test and concluding deployment.
WMGS Jayangani	Helped with system design presentation (class, sequence, ER), helped with front-end and back-end component development, and helped with writing the final report and reviewing it. Other tasks performed were on testing, debugging and demonstrating the system.

Table 7.5-1Contribution of Team Members

The project was developed by all the group members who were collaborative and equally shared the responsibilities of planning, developing, and evaluating the project so that they could have a balanced workload and share the achievements of the project.

7.6. Summary

The chapter was a summary of the Blue Heaven Rentals project results but provided the project with details of the success, difficulties, and possible improvements in the system in the future. The project has managed to solve the specified issue of the inefficient search of boarding houses by using a credible and interactive online solution. This system was successfully completed because of the hard work of the team, their contribution and effective cooperation.

8. References

[1] “React,” React – A JavaScript library for building user interfaces, [Online]. Available: https://react.dev/. [Accessed 26 Oct 2025].

[2] “Getting started with React – Learn web development,” Mozilla Developer Network (MDN), [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/web-development/Frameworks_and_libraries/React. [Accessed 26 Oct 2025].

[3] “Cloud Firestore | Firebase,” Google Firebase Documentation, [Online]. Available: https://firebase.google.com/docs/firestore. [Accessed 26 Oct 2025].

[4] “Cloud Firestore Data model | Firebase,” Google Firebase Documentation, [Online]. Available: https://firebase.google.com/docs/firestore/data-model. [Accessed 26 Oct 2025].

[5] “Get data with Cloud Firestore,” Google Firebase Documentation, [Online]. Available: https://firebase.google.com/docs/firestore/query-data/get-data. [Accessed 26 Oct 2025].

[6] “How to make a sequence diagram | UML diagram tutorials | Gliffy,” Gliffy Blog, [Online]. Available: https://www.gliffy.com/resources/how-make-sequence-diagram-uml-diagram-tutorials. [Accessed 26 Oct 2025].

[7] “UML Sequence Diagram Tutorial,” EdrawMax Guide, [Online]. Available: https://edraw.wondershare.com/video/edrawmax/how-to-draw-sequence-diagram.html. [Accessed 26 Oct 2025].

[8] “UML Sequence Diagram Tutorial | Lucidchart,” Lucidchart Blog, [Online]. Available: https://www.lucidchart.com/pages/uml/uml-sequence-markup. [Accessed 26 Oct 2025].

[9] “Sequence Diagrams – Unified Modeling Language (UML),” GeeksforGeeks, [Online]. Available: https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-sequence-diagrams/. [Accessed 26 Oct 2025].

[10] “Responsive Web Design: Master Responsive CSS For Best User Experience,” Hackr.io, [Online]. Available: https://hackr.io/tutorial/responsive-web-design-master-responsive-css-for-best-user-experience. [Accessed 26 Oct 2025].

[11] “Responsive Web Design Tutorial | LearnWebCode,” LearnWebCode, [Online]. Available: https://learnwebcode.com/responsive-web-design-tutorial/. [Accessed 26 Oct 2025].

[12] “Create deploys | Netlify Docs,” Netlify Documentation, [Online]. Available: https://docs.netlify.com/site-deploys/create-deploys/. [Accessed 26 Oct 2025].

[13] “CSS Responsive Web Design Introduction,” TechGeekBuzz, [Online]. Available: https://www.techgeekbuzz.com/tutorial/css/css-responsive-web-design-introduction/. [Accessed 26 Oct 2025].

[14] “The Living Conditions of Boarding Houses around the University of Sri Jayewardenepura,” Journal of Real Estate Studies, University of Sri Jayewardenepura, [Online]. Available: https://journals.sjp.ac.lk/index.php/SLJRE/article/view/7458. [Accessed 26 Oct 2025].

[15] “Determinants of rental value for residential properties: A land owner’s perspective for boarding homes,” Built-Environment Sri Lanka, [Online]. Available: https://besl.sljol.info/articles/10.4038/besl.v12i1.7612. [Accessed 26 Oct 2025].

Responsibility Matrix

Index Number	**Name**	**Work Done**
E2340426	Kriyanjala W	Abstract, Chapter 01, Chapter 02
E2340065	Hatharasinghe AIN	Chapter 05, Chapter 03, Chapter 04
E2340450	Jayangani WMGS	Chapter 06, Chapter 07, Finalize the Report