

# Online Marketplace for Homestays

## Project Report



Sri Lanka Institute of Information Technology  
IT2080 Information Technology Project

Group ID: ITP\_WD\_B04\_G09

Registration No:	Name:
IT21166488	Maleesha K.L.D.D.S
IT21183768	Heshan W.A.S
IT21233562	Ranawaka W.E.I
IT21186424	Bandara N.G.S.S.M
IT21184376	Jayathilaka A.W.N.W
IT21182846	Chamaleen D.B.N
IT21197796	Wasana K.H.I.M
IT21240706	Dhananjana B.K.T

May 2023

## Declaration

---

This project report is our original work, and the content is not plagiarized from any other resource. References for all the content taken from external resources are correctly cited. To the best of our knowledge, this report does not contain any material published or written by third parties, except as acknowledged in the text.

**Authors:**

Author SID	Author name	Signature
IT21166488	Maleesha K.L.D.D.S	
IT21183768	Heshan W.A.S	
IT21233562	Ranawaka W.E.I	
IT21186424	Bandara N.G.S.S.M	
IT21184376	Jayathilaka A.W.N.W	
IT21182846	Chamaleen D.B.N	
IT21197796	Wasana K.H.I.M	
IT21240706	Dhananjana B.K.T	

Date: 19/05/2023

# **Abstract**

---

This document provides a comprehensive overview of our group project for the Information Technology Project module in the second year, second semester. The project focuses on the development of a MERN web application that enables users to buy and sell properties, including houses, rooms, and apartments. Additionally, the web application provides additional features such as finding the nearest furniture showrooms, booking transport services, and locating lawyers for legal assistance.

The document begins with an abstract, followed by a table of contents. In Chapter 1, the project background, problem identification, and motivations are discussed. A literature review is conducted to explore existing research and solutions relevant to the project. The aim and objectives of the project are outlined, and the chosen methodology for implementation is described. Additionally, a clickable link to the Git repository is provided for convenient access to the project's source code.

Chapter 2 covers the requirements analysis, including stakeholder analysis and requirements modeling. This chapter aims to capture the needs and expectations of the application's users, including the ability to find the nearest furniture showrooms, book transport services, and locate lawyers for legal assistance. The requirements for these additional features are defined and integrated into the project scope.

The subsequent chapter, Chapter 3, focuses on the design and development aspects of the web application. Various diagrams, such as components, processes, workflows, and databases, are presented to provide a visual representation of the system's architecture and functionality, including the features related to furniture showrooms, transport booking, and lawyer search.

Chapter 4 details the testing phase, including the definition of test cases and the presentation of test results. It covers the testing of the core functionalities as well as the additional features, ensuring that the web application operates smoothly and provides accurate results for finding showrooms, booking transport, and locating lawyers.

In Chapter 5, the evaluation and conclusion chapter, the project is assessed based on the test results and user or expert feedback. The extent to which the objectives of the project were achieved, including the successful integration of the furniture showroom, transport booking, and lawyer search functionalities, is determined. A concluding statement is provided summarizing the achievements and potential areas for future enhancements.

Finally, the document concludes with a references section, where all sources consulted during the project are cited using MS Word's referencing tool.

Overall, this document serves as a comprehensive resource for understanding our group project for the second year, second semester of the Information Technology Project module. It covers the project's background, problem identification, literature review, objectives, methodology, requirements analysis, design and development, testing, evaluation, and conclusion. Additionally, it highlights the additional features incorporated into the web application, such as finding furniture showrooms, booking transport services, and locating lawyers, enhancing the overall user experience.

## Acknowledgement

---

The work described in this document was conducted as part of our second-semester group project for the Information Technology Project module during our second year. We, the members of the Y2\_S2\_B04\_G09 group, would like to express our heartfelt gratitude to all those who provided us with valuable guidance and necessary support, enabling us to complete our project.

We extend special thanks to all the lecturers and instructors associated with the Information Technology Project (ITP) module, particularly Mr. MR Harshanath S.M.B. Their advice and encouragement have been instrumental in our project's success. Their unwavering support has helped us meet all the requirements effectively. We would also like to acknowledge the dedication and hard work of all the group members. Each member contributed maximum effort and demonstrated strong commitment throughout the semester, which culminated in the development of this website. Their collective efforts have led to the successful completion of this project.

Additionally, we would like to express our gratitude to our university colleagues for their unwavering support and guidance throughout our project. Your contribution has been invaluable, and we are grateful for your assistance in making this endeavor a success.

We have strived to ensure that this paragraph is both accurate and relevant to our group's achievements, excluding any irrelevant information.

# Table of Contents

---

<b>Declaration</b> .....	i
<b>Abstract</b> .....	ii
<b>Acknowledgement</b> .....	iii
<b>Chapter 01 - Introduction</b> .....	1
1.1 Background.....	1
1.2 Problems and Motivations .....	1
<b>1.2.1 Problems</b> .....	1
<b>1.2.2 Motivation</b> .....	2
1.3 Literature review.....	3
1.4 Aim and objectives. ....	3
<b>1.4.1 Aim:-</b> .....	3
<b>1.4.2 Objectives:-</b> .....	3
1.5 Solution overview .....	4
1.6 Methodology .....	4
1.7 The Structure of the report .....	5
<b>1.7.1 Requirements</b> .....	5
<b>1.7.2 Design and development</b> .....	5
<b>1.7.3 Testing</b> .....	5
<b>1.7.4 Conclusion</b> .....	5
<b>1.7.5 References</b> .....	5
<b>1.7.6 Appendixes</b> .....	5
1.8 A clickable link to Git repository.....	5
<b>Chapter 02 - Requirements</b> .....	7
2.1 Stake Holder Analysis.....	7
2.2 Requirement Analysis .....	7
<b>2.2.1 Listing Management System</b> [IT21166488 - Maleesha K.L.D.D.S] .....	7
<b>2.2.2 Payment Management System</b> [IT21183768 – Heshan W.A.S].....	8
<b>2.2.3 Blog Management System</b> [IT21186424 – Bandara N.G.S.S.M].....	9
<b>2.2.4 Customer Profile</b> [IT21233562 – Ranawaka W.E.I.].....	10
<b>2.2.5 Furniture Showroom Management System</b> [IT21184376 – Jayathilaka A.W.N.M] .....	11
<b>2.2.6 Transport Management System</b> [IT21182846 – Chamaleen D.B.N].....	12
<b>2.2.7 Wasana K.H.I.M</b> [IT21197796 – Wasana K.H.I.M].....	13
<b>2.2.8 Legal Contact Management System</b> [IT21240706 – Dhananjana B.K.T].....	14
2.3 Requirement Modeling .....	16

<b>2.3.1 Use case Diagram of the system .....</b>	16
<b>2.3.2 Use case Diagram of the system .....</b>	25
<b>Chapter 03 – Design and Development .....</b>	74
3.1 ER Diagram .....	74
3.2 High level Architecture.....	75
3.3 Class diagram.....	76
3.4 User Interfaces .....	77
<i>Listing Management System [IT21166488 – Maleesha K.L.D.D.S]</i> .....	77
<i>Payment Management System [IT21183768 – Heshan W.A.S]</i> .....	83
<i>Blog Management System [IT21186424 – Bandara N.G.S.S.M]</i> .....	85
<i>Customer Profile [IT21233562 – Ranawaka W.E.I]</i> .....	87
<i>Showroom Management System [IT21184376 – Jayathilaka A.W.N.W]</i> .....	91
<i>Vehicle Management System [IT21182846 – Chamaleen D.B.N]</i> .....	95
<i>User Management and Pending Listing Management System [IT21197796 – Wasana K.H.I.M]</i> .....	100
<i>Legal Contact and Support Center Managements [IT21240706 – Dhananjana B.K.T]</i> .....	105
3.5 Databases .....	110
<b>3.5.1 Listings .....</b>	110
<b>3.5.2 Customer .....</b>	110
<b>3.5.3 Todo .....</b>	110
<b>3.5.4 Users .....</b>	110
<b>3.5.5 Vehicles.....</b>	110
<b>3.5.6 Revenue .....</b>	110
<b>3.5.7 Bookings .....</b>	110
<b>3.5.8 Card Details .....</b>	110
<b>3.5.9 Payment Details .....</b>	110
<b>3.5.10 Blogs.....</b>	110
<b>3.5.11 Lawyers.....</b>	110
<b>3.5.12 Support .....</b>	111
<b>3.5.13 Showrooms.....</b>	111
<b>3.5.14 ShowroomsImages .....</b>	111
<b>3.5.15 Furnitures .....</b>	111
3.6 Development Aspects .....	111
<b>3.6.1. M – MongoDB.....</b>	111
<b>3.6.2 E – Express .....</b>	112
<b>3.6.3 R – React .....</b>	112

<b>3.6.4 N – NodeJS .....</b>	113
<b>3.6.5 – Figma.....</b>	113
<b>3.6.6 – ClickUp .....</b>	113
<b>3.6.7 – Insomnia .....</b>	114
<b>3.6.8 – GitHub.....</b>	114
<b>3.6.9 – Postman.....</b>	114
<b>3.6.10 – Visual Code .....</b>	114
<b>Chapter 04 – Testing.....</b>	115
4.1 Testing Methods.....	115
<b>4.1.1 Unit Testing .....</b>	115
<b>4.1.2 Integration Testing.....</b>	115
4.2 Listing Management System.....	116
4.3 Payment Management.....	118
4.4 Blog Management.....	119
4.5 Customer Management .....	120
4.6 Furniture Showroom Management .....	122
4.7 Transport Management System .....	124
4.8 User Management .....	125
4.9 Pending List Management .....	127
4.10 Legal Contact Management .....	128
4.11 Support Center Management .....	130
<b>Chapter 05 – Evaluation and Conclusion .....</b>	90
5.1 Evaluation .....	90
<b>5.1.1 Assessment of the Project results.....</b>	90
5.2 Conclusion .....	90
<b>References.....</b>	92
References .....	92
<b>Appendix.....</b>	I

# Table of Figures

---

## Chapter 02

FIGURE 2. 1 – USE CASE DIAGRAM OF THE SYSTEM .....	16
FIGURE 2. 2 – USE CASE DIAGRAM OF THE LISTING MANAGEMENT .....	17
FIGURE 2. 3 - USE CASE DIAGRAM OF THE PAYMENT MANAGEMENT <b>ERROR! BOOKMARK NOT DEFINED.</b>	
FIGURE 2. 4 - USE CASE DIAGRAM OF THE BLOG MANAGEMENT .....	19
FIGURE 2. 5 - USE CASE DIAGRAM OF THE CUSTOMER MANAGEMENT.....	20
FIGURE 2. 6 - USE CASE DIAGRAM OF THE FURNITURE SHOWROOM MANAGEMENT .....	21
FIGURE 2. 7 - USE CASE DIAGRAM OF THE TRANSPORT MANAGEMENT .....	22
FIGURE 2. 8 - USE CASE DIAGRAM OF THE USERS AND PENDING LIST MANAGEMENT .....	23
FIGURE 2. 9 - USE CASE DIAGRAM OF THE LEGAL CONTACT AND SUPPORT CENTER MANAGEMENT .....	24
FIGURE 2. 10 - ACTIVITY DIAGRAM OF THE LISTING MANAGEMENT .....	25
FIGURE 2. 11 - ACTIVITY DIAGRAM OF THE PAYMENT MANAGEMENT .....	26
FIGURE 2. 12 - ACTIVITY DIAGRAM OF THE BLOG MANAGEMENT.....	27
FIGURE 2. 13 - ACTIVITY DIAGRAM OF THE CUSTOMER INFORMATION UPDATE .....	28
FIGURE 2. 14 - ACTIVITY DIAGRAM OF THE CUSTOMER PASSWORD UPDATE .....	29
FIGURE 2. 15 - ACTIVITY DIAGRAM OF THE FURNITURE SHOWROOM MANAGEMENT .....	30
FIGURE 2. 16 - ACTIVITY DIAGRAM OF THE VEHICLE MANAGEMENT .....	31
FIGURE 2. 17 - ACTIVITY DIAGRAM OF THE USER MANAGEMENT .....	32
FIGURE 2. 18 - ACTIVITY DIAGRAM OF THE PENDING LISTS MANAGMENTS .....	33
FIGURE 2. 19 - ACTIVITY DIAGRAM OF THE LOGIN.....	34
FIGURE 2. 20 - ACTIVITY DIAGRAM OF THE REGISTER .....	35
FIGURE 2. 21 -ACTIVITY DIAGRAM OF THE LEGAL CONTACT MANAGEMENT .....	36
FIGURE 2. 22 - ACTIVITY DIAGRAM OF THE SUPPORT CENTER MANAGEMENT .....	37

## Chapter 03

FIGURE 3. 1 -ER DIAGRAM OF THE SYSTEM .....	74
FIGURE 3. 2 - HIGH LEVEL ARCHITECTURE OF THE SYSTEM .....	75
FIGURE 3. 3 - CLASS DIAGRAM OF THE SYSTEM .....	76
FIGURE 3. 4 - HOME PAGE .....	77
FIGURE 3. 5 -ALL PROPERTIES PAGE .....	77
FIGURE 3. 6 - SINGLE PROPERTY PAGE.....	78
FIGURE 3. 7 - LANDLORD DASHBOARD .....	78
FIGURE 3. 8 - DISPLAY ALL LISTING PAGE .....	79
FIGURE 3. 9 - ADD NEW LISTING PAGE .....	79
FIGURE 3. 10 - UPDATE LISTING PAGE .....	80
FIGURE 3. 11 - LISTING REPORT GENERATE .....	80
FIGURE 3. 12 - LANDLORD RESERVATION PAGE.....	81
FIGURE 3. 13 - LANDLORD REVENUE PAGE .....	81
FIGURE 3. 14 - LANDLORD PROFILE PAGE.....	82
FIGURE 3. 15 - LANDLORD PROFILE SETTINGS .....	82
FIGURE 3. 16 - PAYMENT DASHBOARD .....	83
FIGURE 3. 17 - PAYMENT CARD ADDING PAGE .....	83
FIGURE 3. 18 - NEW CARD ADDING PAGE .....	84
FIGURE 3. 19 - CHECKOUT PAGE .....	84
FIGURE 3. 20 - BLOG MANAGEMENT PAGE .....	85
FIGURE 3. 21 - NEW BLOG ADDING PAGE.....	85
FIGURE 3. 22 - BLOG UPDATE PAGE.....	86
FIGURE 3. 23 - ALL BLOG VIEW PAGE .....	86
FIGURE 3. 24 - CUSTOMER PROFILE .....	87

FIGURE 3. 25 - CUSTOMER SETTING PAGE.....	87
FIGURE 3. 26 - CUSTOMER PROFILE INFO PAGE .....	88
FIGURE 3. 27 - CUSTOMER LOGIN AND SECURITY PAGE .....	88
FIGURE 3. 28 - CUSTOMER PRIVACY AND POLICY PAGE .....	89
FIGURE 3. 29 – CUSTOMER VIEW BOOKING PAGE .....	89
FIGURE 3. 30 - CUSTOMER GENERATE REPORT PAGE .....	90
FIGURE 3. 31 - SHOWROOM DASHBOARD PAGE.....	91
FIGURE 3. 32 - ADD SHOWROOM PAGE.....	91
FIGURE 3. 33 - ADD NEW FURNITURE PAGE .....	92
FIGURE 3. 34 - SHOWROOM SUBSCRIPTION PAGE .....	92
FIGURE 3. 35 - SHOWROOM PROFILE SETTING PAGE.....	93
FIGURE 3. 36 - SHOWROOM PROFILE PAGE .....	93
FIGURE 3. 37 - ALL SHOWROOM PAGE .....	94
FIGURE 3. 38 - VEHICLE DASHBOARD PAGE .....	95
FIGURE 3. 39 - ALL VEHICLE LISTING PAGE .....	95
FIGURE 3. 40 - ADD VEHICLE PAGE .....	96
FIGURE 3. 41 - UPDATE VEHICLE PAGE.....	96
FIGURE 3. 42 - VIEW VEHICLE BOOKINGS PAGE .....	97
FIGURE 3. 43 - VEHICLE PROFILE PAGE.....	97
FIGURE 3. 44 - UPDATE VEHICLE OWNER PROFILE PAGE .....	98
FIGURE 3. 45 - ALL TRANSPORT VIEW PAGE .....	98
FIGURE 3. 46 - SINGLE VEHICLE DETAILS PAGE.....	99
FIGURE 3. 47 - LOGIN PAGE .....	100
FIGURE 3. 48 - SIGNUP PAGE .....	100
FIGURE 3. 49 - SITE OWNER DASHBOARD .....	101
FIGURE 3. 50 - ALL USERS DISPLAY PAGE .....	101
FIGURE 3. 51 - ADD NEW USER PAGE .....	102
FIGURE 3. 52 - UPDATE USER PAGE .....	102
FIGURE 3. 53 - PENDING PROPERTIES PAGE .....	103
FIGURE 3. 54 - PENDING VEHICLES PAGE .....	103
FIGURE 3. 55 - GENERATE ALL USER REPORT .....	104
FIGURE 3. 56 - SUPPORT CENTER ADD NEW RECORD FORM .....	105
FIGURE 3. 57 - SUPPORT CENTER PAGE .....	105
FIGURE 3. 58 - SUPPORT CENTER REPORT GENERATING .....	106
FIGURE 3. 59 - PROFILE Q&A PAGE .....	106
FIGURE 3. 60 - PROFILE CONTACT SUPPORT PAGE .....	107
FIGURE 3. 61 - VIEW ALL LAWYERS PAGE .....	107
FIGURE 3. 62 – LAWYER PROFILE PAGE .....	108
FIGURE 3. 63 – LAWYER UPDATE INFO UPDATEE PAGE.....	108
FIGURE 3. 65 – LAWYER SERVICE LISTING PAGE.....	109

## List of Tables

---

TABLE 4. 1 – LISTING MANAGEMENT TEST CASES .....	116
TABLE 4. 2 – LISTINGS MANAGEMENT TEST CASES .....	117
TABLE 4. 3 - PAYMENT MANAGEMENT TEST CASES .....	118
TABLE 4. 4 - BLOG MANAGEMENT TEST CASES.....	119
TABLE 4. 5 - CUSTOMER MANAGEMENT TEST CASES .....	120
TABLE 4. 6 - FURNITURE SHOWROOM MANAGEMENT TEST CASES.....	122
TABLE 4. 7 - TRANSPORT MANAGEMENT TEST CASES .....	124
TABLE 4. 8 - USER MANAGEMENT TEST CASES .....	125
TABLE 4. 9 - PENDING LIST MANAGEMENT TEST CASES .....	127
TABLE 4. 10 - LEGAL CONTACT MANAGEMENT TEST CASES .....	128
TABLE 4. 11 - LEGAL CONTACT TEST CASES .....	130

## **List of Abbreviations**

---

MERN - Mongo DB Express React NodeJS .....	ii, 4
UI - User Interface.....	5
UML - Unified Modeling Language .....	5

# **Chapter 01 - Introduction**

---

## **1.1 Background**

Rent Mate is an all-in-one online marketplace that is designed to simplify the process of booking a property, whether it be for a short-term stay or a long-term rental. It is a web application that enables landlords to list their properties, and customers can easily browse through the listings to find their perfect match.

In addition to property listings, Rent Mate offers a range of other services that make the property booking experience even more convenient. Customers who are moving their belongings from one place to another can book transportation through the platform, making it easy to transport their items with ease. For those who book unfurnished properties, Rent Mate provides a list of nearby furniture showrooms where they can find the perfect pieces to furnish their new space.

If customers require legal support when booking a property, they can find a lawyer for that task through the platform. Rent Mate focuses on providing the best services to its customers and bringing everything needed for property booking into one place. The platform's search box and filtering options enable customers to quickly and easily find the properties that best fit their needs and preferences.

The platform also ensures the safety and security of its users by validating user accounts and removing any invalid users. The site owner has access to manage property listing availability, and customers can manage their account information with ease. Rent Mate provides a cancellation feature, which allows customers to cancel their bookings if needed.

For landlords, the platform offers a dashboard where they can see the status of their listings and incomes and manage details about their properties. Vehicle owners who offer transportation services can also manage their vehicle listings and get details about their bookings. Furniture showroom owners can add details about their showrooms to attract more customers.

Customers can get help through the support center, and the site owner has a dashboard to manage details about customer problems and their status. The platform also provides a list of lawyers and their services, enabling customers to feel confident in their booking decisions and know they have the support they need if any legal issues arise.

Rent Mate offers an online payment gateway that allows customers to pay for property and transport bookings instantly, and using a data encryption method, the platform gives security to customers' sensitive card details. The platform also provides refunding methods for customers' booking cancellations.

Overall, Rent Mate is a comprehensive and convenient platform that provides a range of services related to property booking. With its focus on customer satisfaction and commitment to safety and security, it is a top choice for anyone looking to book a property, transportation, or furniture showroom.

## **1.2 Problems and Motivations**

### **1.2.1 Problems**

- Physical payment: Physical payments are not secure and fast. Most people are now moving with new technologies and moving to online payment methods. Due to the pandemic situation and travel restrictions happening from time to time, people have faced a lot of difficulties when making payments.
- Time-Consuming: It's very difficult to find a good property rental these days and very time-consuming to find the proper one. When customers are at a long distance from the property

place going there physically is very time-consuming. when customers need more services like legal support and find transport they must go to different people in different places.

- Competition in the property renting market: Competition in the property renting market is another problem for developing the system, In some time periods mainly January, December months people are relaxing their minds and they are organizing journeys then they forced to find homestays or transportation then they have to face competition to find homestays, but actually, some people fail that competition then they wasting their time, so then the seasoning period is mainly involved for that.
- Lack of trust: In most of the sites, we are giving the information to fulfill our requirements which means if we want to find homestays, we find the platform by using browsers we give our valid information, but that site is fake, we must waste our time and money to involving that find that sites. we cannot find trustworthy platforms that are helping illegal things also.
- Fast Tech Growth: when developing a system, we cannot guarantee to all the people who are using this system are technological people, when they are using the new technology, they feel they do not anything about that and they feel they are isolated to that, and they reduce social interaction and social skills.

### **1.2.2 Motivation**

- Secure: The frequency of security breaches in company systems has significantly increased over the last couple of years, mostly due to a lack of awareness of the sensitive information that users have but it may also be very important financially.
- Usability: Using an enterprise solution like this can have a big impact for anyone interested in improving their user experience. According to the survey, because of automation, users are looking for a consistent experience and even making sure that back-end operations are as efficient as feasible. Anyone can easily browse via this program with just a little understanding of technology. Despite a shortage of staff, the system is accessible every day of the week and has an unlimited usage limit.
- Fast and reliable: An enterprise system can lessen not only information technology and training costs but also the time and effort needed by employees to complete daily duties and obligations. Due to its effectiveness, the Enterprise system can eliminate labor-intensive, repetitive tasks, freeing up team members' time and attention for other, more crucial tasks. Technically speaking, this system is faultless, has no glitches, and has great dependability.
- Competitors online: If don't have a website, it's very probable that their rival has. If this is the case, you are missing out on opportunities to attract new clients and stay at the top of their minds. It is essential that no chances are lost and taken advantage of by the opposition.
- Creditability: Having an online presence is essential for providing clients with important information and responding to their questions, as well as providing a high-quality, user-friendly website.
- Growth opportunity: Websites are generally useful for creating a location where possible investors may be directed. It displays the goals of our business, its accomplishments to date, and its potential for the future.

### **1.3 Literature review**

Existing system has different type of feature but looking at those features customer cannot satisfying to using for those things. Most of the systems are developing to include common features, but according to the user requirements they have not to proper way to rent their property and get a profit through that. We will examine the existing research on online marketplaces for homestays in this review of literature and highlight their essential characteristics, advantages, and difficulties.

Users find online homestay markets appealing since they offer a wide range of functions. The capability to look for and reserve a home based on standards like location, cost, and facilities is one of the essential advantages. These sites also give landlords the ability to construct thorough listings with descriptions, images, and reviews from previous visitors. Online markets for homestays frequently offer a safe payment method.

**Benefits:** Using online marketplaces for homestays has several advantages. First, they provide a distinctive and genuine travel encounter that enables visitors to fully experience the local culture. Homestays also provide a more individualized experience because they give visitors the chance to speak with their hosts and get tailored recommendations and transport for local sights and activities. Homestays also frequently cost less than standard lodging and provide more flexible check-in and check-out periods.

**Challenges-**Despite their many advantages, online marketplaces for homestays still confront several difficulties. Making sure that users are safe and secure is one of the major challenges. This entails confirming the individuals' identity and making sure the lodgings adhere to a set of safety requirements. Since visitors might not always be aware of regional norms or may have different expectations for their stay, there is also the risk of cultural mistakes or conflicts.

Then we are developing the system to solve those issues, so developing that system we can give the customer to trustworthy system when the adding homestays or vehicles then approving it to from the admin, we can get the trustworthy system and customer and work with through the system efficiently.

### **1.4 Aim and objectives.**

#### **1.4.1 Aim:-**

- to provide a comprehensive platform that connects tenants with various home stay providers, transport providers, and furniture shop owners to make the rental process more convenient and hassle-free.

#### **1.4.2 Objectives:-**

- Providing tenants with a user-friendly platform where they can search and book homestays that meet their needs and preferences, including transportation and furniture options.
- Offering a wide variety of homestay options from a variety of providers, allowing renters to choose the one that best suits their budget and lifestyle.
- Providing a secure and reliable payment system that allows tenants to pay for their chosen home accommodation and related services easily and securely.
- To collect feedback from tenants, housing providers, transport providers and furniture shoppers to continuously improve the platform's features and services.
- Establishing a reputation as a trusted and reputable platform that provides high quality home stay options and related services to tenants.

## **1.5 Solution overview**

Our property rental platform is a comprehensive solution designed to meet the needs of individuals seeking rental properties and property owners. Built on the MERN stack, our user-friendly platform streamlines the property search process and provides additional services for a convenient and hassle-free experience. Property seekers can easily find their ideal rental properties . Property owners can efficiently list their properties, attracting potential tenants with detailed descriptions, photos, and contact information.

But our platform goes beyond property listings. We integrate additional services to enhance the rental experience. Users can access transportation services to facilitate their move, legal services for guidance throughout the rental process, and a furniture showroom locator to conveniently furnish their new rental home.

In addition to these features, our platform also includes a blog management system. Users can access a variety of informative and engaging blog posts related to property rental, tips for tenants and landlords, market trends, and more. The blog provides valuable insights and resources to educate and assist users in making informed decisions.

Our platform offers numerous benefits. It saves time by streamlining the property search process, eliminates the hassle of searching for additional services separately, and enhances decision-making through detailed listings, reviews, and ratings. Comprehensive support is provided, ensuring a smooth rental process with readily available legal services, transport options, and access to valuable blog content. The user experience is seamless, with a user-friendly interface and personalized profiles.

Furthermore, our platform includes a payment system that allows users to make payments securely and conveniently for rental properties, additional services, and any applicable fees. The payment methods available on our platform provide users with a secure and seamless transaction experience.

## **1.6 Methodology**

- The following steps would be taken in the technique for building an online marketplace for homestays:
- Define the scope: Choose the features you want to include in the online marketplace and the types of homestays you want to provide. This could entail investigating current marketplaces to discover what features they provide.
- Create a platform: Choose a platform that will work for your purposes and use it to build the online marketplace. This can entail creating a unique platform from scratch or utilizing an already-existing platform like Airbnb or VRBO.
- Build a user-friendly interface: Develop a user-friendly interface that makes it easy for owners to list their homestays, vehicle, furniture and get the lawyer facility and for users to search and book homestays, reserve vehicles and search lawyers.
- Implement payment system: Implement a secure payment system that allows users to pay for their bookings online and ensures that hosts receive their payment in a timely manner.
- Develop policies and guidelines: Develop policies and guidelines for hosts and guests to follow to ensure a safe and pleasant experience for all parties involved.
- Implement marketing strategy: Implement a marketing strategy to attract user platform. This may involve advertising using the blogs.

- Monitor and evaluate: Monitor the performance of online marketplace and evaluate the feedback from users to make improvements and ensure the platform continues to meet the needs of its users. [1]

## **1.7 The Structure of the report**

The remaining sections of the project report consist of six main parts, each of which includes more detailed subsections.

### **1.7.1 Requirements**

- The project encompasses the documentation of user stories, use case diagrams, and activity diagrams to elucidate the functional requirements. In addition, there is a consideration of non-functional requirements to address various aspects of the project's performance, usability, security, and other relevant factors.

### **1.7.2 Design and development**

- The design aspect of the project is presented here, utilizing various diagrams to provide clarity. UML diagrams, including class diagrams, activity diagrams, and high-level architecture diagrams, are employed to illustrate the software design. An ER diagram is employed to showcase the database design of the software. To enhance understanding of the UI design, user interface designs are utilized. This section also delves into the implementation details of the project, including the selection of the database management system, front-end and back-end technologies, and the programming language chosen for implementation.

### **1.7.3 Testing**

- This section outlines the test plan for the software, which involves using test cases to verify the proper functioning of all main functions along with their respective validations.

### **1.7.4 Conclusion**

- This section provides an overview of the entire project, including a discussion of the limitations and weaknesses of the proposed methodology and solution.

### **1.7.5 References**

- This section includes a compilation of all the source materials used in the creation of the reports.

### **1.7.6 Appendixes**

- This section contains additional information about the individual contributions made by each group member.

## **1.8 A clickable link to Git repository**

- [Clickable Link to Git repo](#)

# **Chapter 02 - Requirements**

---

## **2.1 Stake Holder Analysis**

The stakeholders in the system include customers looking for short-term home stays, property owners or hosts offering accommodations, company staff managing operations and support, suppliers or service providers offering transportation, furniture showrooms, and legal services, and system owner overseeing system functionality and security. They all depend on the system to facilitate smooth interactions and transactions, enabling customers to find accommodations, property owners to manage listings, staff to provide support, suppliers to deliver services, and administrators to maintain a secure platform. Understanding and addressing their needs is crucial for a successful and satisfying experience for everyone involved.

## **2.2 Requirement Analysis**

### **2.2.1 Listing Management System** [IT21166488 - Maleesha K.L.D.D.S]

#### **2.2.1-1 Functional Requirements**

The main purpose of the Listing management system is property information management. This system allows property owners to provide details about their property, including name of the property, category, location , and images, as well as key details. Property owners can update their listings as needed, ensuring that the information is always accurate and up to date. If needed they can delete the property as well through their dashboard.

As well as property owners can search for their added listings through their dashboard. Another important aspect of a listing management system is the ability to view reservations made through the website. This feature helps property owners know when a booking is made, so they can get ready to make sure the renter has a good experience. Through their dashboard they can view the revenue that they have made.

Additionally, through the listing management system it can generate personalized income reports. These reports provide property owners with a clear and summary overview of their rental income, it helps property owners to keep track of their money and make smart business choices.

overall, the listing management system is very important for our website. It makes managing listings easier, helps track rental income, and ensures a good experience for renters and property owners.

- As a landlord, I want to be able to add my property to the website, so I can advertise my properties to the audience.
- As a landlord, I want to be able to generate income report at the end of the month, so I can overall idea about net income of the month.
- As a landlord, I want to be able to update the listing for my properties as needed, so I can add new information, ensuring that the details are accurate and up to date.
- As a landlord, I want to be able to delete the listing, so I can remove sold properties.
- As a landlord, I want to be able to view my listing, so I can view how it look like to customers.

### 2.2.1-2 Non- Functional Requirements

- **User Friendly:** The system should be user friendly, with clear messages, helpful friendly with a clear interface, informative error documentation. This will attract and retain customers for the startup.
- **Compatibility:** Be compatible with a range of web browsers and devices to ensure that all users can access it.
- **Easy to Access:** Easy to access and use from anywhere, whether on a desktop or mobile device. This includes ensuring the website is responsive and optimized for mobile devices.
- **Availability:** The system should always be maintained with minimum periods of downtime or maintenance. Must take backup for recovery purpose if there is any data lost.

### 2.2.1-3 Technical Requirements

- Application Programming Interface (API)
  - Express – RESTful APIs
- Database Management
  - Mongo DB
- Responsive Design
- Frontend
  - React
  - Prime React – UI Library
- Backend
  - Nodejs

## **2.2.2 Payment Management System [IT21183768 – Heshan W.A.S]**

### 2.2.2-1 Functional Requirements

In this function manage all reservation payments, service provider charges and advertising charges also site Owner's commission of our web application. who are looking for a rental house or rental vehicle if they are interested in any services in our web application, they can reserve that through our web application. Customers can reserve that by adding valid payment details and Booking details. In this scenario they want payment gateway. Customers can pay their reservation payments via the credit/Debit card or Pay here Option. They add their card, update card details, and remove their existing credit or debit card from our system. After the payment process when done, the customer can view their payment report. After a customer makes a reservation, it is updated on their profile page. When the customer cancels the booking, the payment made by him will be refunded. It is done through a third-party payment method. Once a refund is issued the system admin will update their profile page.

After the customer pays and completes their booking, the amount due to the service provider and the amount due to the site owner will be updated on his page. Apart from that, in our web application, advertisements about places where Lawyer service and Furniture Service can be obtained as additional services have been included. There a subscription fee is charged from the service providers. The service provider of lawyer service and furniture service will have to

pay the annual charge at the time of placing their advertisement. All these payments are updates on site owner dashboard.

As a payment admin, he or she can search for the transaction they want. They can also filter the transaction report and get the specific transaction details. Property owner, Vehicle Owner, Payment admin can generate the previous transaction Details Report.

- As a site owner I want to interface to payment handling so that I can update payment details, search payment details, and get reports of financial transactions.
- As a customer I want to payment gateway so that I can pay for my reservation.

#### 2.2.2-2 Non-Functional Requirements

- **Security:** In managing payments we take appropriate measures to protect user data and prevent unauthorized access using encryption and authentication protocols.
- **Usability:** The payment system should be user-friendly and intuitive, with easy-to-use navigation and clear instructions. The system should also provide users with feedback on the status of their payments, such as confirmation messages.
- **Performance:** Our payment system should be fast and responsive, with quick processing times and minimal lag.
- **Reliability:** Establish clear payment terms. As an example, clearly define payment terms, including payment methods, cancellation policy, and refunds.
- **Responsible:** We make sure our pricing is transparent and clearly stated.

#### 2.2.2-3 Technical Requirements

- Web-Based System
- Back end – Node JS & Express
- Front end - React.
- Database - MongoDB

### **2.2.3 Blog Management System [IT21186424 – Bandara N.G.S.S.M]**

#### 2.2.3-1 Functional Requirements

Information about all the services available to customers through our website is published in the blog site. The information required by the customer in an online marketplace includes information about the home stays facilities, show rooms, transportation facilities as well as the information required in the legal framework. A blog post is written by the system site owner. The articles for all these aspects are also updated by admin according to the changes on the respective days. When admin creates the blog, and all the blogs are display with in site owner dashboard then if site owner wants to delete any blog post admin can click the delete icon and do the own task. Using the topic given during the creation of the blog post, the admin can get reports of the dates that published the posts and comments written by customers related to the

published blogs. Admin can search relevant categories as want by using the topics of blogs then admin can view relevant blog post.

- As a system admin I want to add blog according to the categories acknowledgement to the user.
- As a system admin I want to so that I can give a update created blogs according to the new updates can publish new things to the customers through blogs.

#### 2.2.3-2 Non-Functional Requirements

- **Usability:** Websites should maintain user-friendly designs and be accessible by the users.
- **Availability:** Ability of the users to access the web site and use that as they expect.
- **Accurate:** Contents that are publish through the web site should be reliability, truthfulness and correctness.it makes a better attraction with the users.
- **Performance:** Better performance web site allows users to interact with it quickly. Important is minimizing the loading and response time.
- **Efficiency:** If web site does what should it do utility and effectiveness goals are fulfilled.
- **User-friendly:** A web site should make it easy for users to interact by any device and get information as effortlessly as possible.

#### 2.2.3-3 Technical Requirements

- **Back end** – Node.js, express
- **Front end** – React, Bootstrap
- **Database** – Mongo DB

### 2.2.4 Customer Profile [IT21233562 – Ranawaka W.E.I.]

#### 2.2.4-1 Functional Requirements

The system provides several options for managing personal information and bookings. Users can easily navigate through these features to ensure their preferences are accurately reflected. Starting with personal information, individuals can view and update their details, ensuring that the information stored is up-to-date and relevant. Additionally, users can change their password whenever necessary, maintaining the security of their accounts. For those who wish to discontinue their membership, the option to delete their account is available.

Moving on to property bookings, the system offers comprehensive functionality. Users can view their current property bookings, accessing important details such as reservation dates and accommodation specifics. If any changes need to be made, individuals have the flexibility to

modify their property booking details as required. Moreover, users also have the option to cancel a property booking altogether if the need arises.

In terms of transport bookings, similar capabilities are provided. Users can conveniently view their transport bookings, gaining access to essential information related to their travel plans. If modifications are necessary, individuals can easily change their transport booking details to align with their preferences. Furthermore, if a transport request is no longer needed, users have the option to delete it from their records.

Lastly, the system caters to reporting needs by enabling users to generate a booking details report [2]. This feature allows individuals to obtain a comprehensive summary of their bookings, facilitating better organization and planning.

- AS a user I want generate a report about my booking so that I can use the pdf for legal purposes
- As a user I want to update my personal information so that I can correct them whenever I change them

#### 2.2.4-2 Non-Functional Requirements

- User Friendly: The system is designed with a user-friendly interface, making it easy for customers to navigate and perform tasks efficiently.
- Easy to access: The system can be easily accessed from various devices, allowing customers to connect and manage their information and bookings conveniently.
- Availability: The system is available 24/7, ensuring that customers can access its features at any time that suits them, promoting flexibility and efficiency.

#### 2.2.4-3 Technical Requirements

- back end - express, NodeJS
- Front end - react, prime react.
- data base - mongo DB

### **2.2.5 Furniture Showroom Management System [IT21184376 – Jayathilaka A.W.N.M]**

#### 2.2.5-1 Functional Requirements

Furniture showroom management system designed with the interests of both customers and showroom owners. The system works by charging showroom owners a subscription fee for the ability to advertise their showrooms on the network. After paying the fee, the owner gains access to an add showroom form where they may add information about their showroom, such as the type of furniture they sell, where they are located, when they are open, how to contact them, and other relevant details. Owners can add a showroom to their profile. Showroom owners can enter details such as the name, location, images, contact numbers. Clicking on the add

furniture button allows the owner to add furniture to their showroom dashboard. The furniture should be categorized and displayed in the place reserved for the showroom so that customers can easily browse the furniture. Within the function, the owner can perform a variety of operations, as well as deleting or modifying specific furniture items. The owner can also add new furnishings to the system as needed and includes a search option where the owner can enter any data of furniture and obtain information about that item. Finally, the owner can generate a report that shows what types of furniture were installed within a specific time.

- As a furniture showroom owner, I want to publish my showroom so that I can sell my own property and earn profit.
- As a furniture showroom owner, I want to add my furniture items so that I can provide acknowledgement to the customer about that item in my showroom.

#### 2.2.5-2 Non-Functional Requirements

- **Maintainability:** The system should be easy to maintain and update, to ensure that it remains up to-date and relevant.
- **Usability:** The system should be user-friendly and easy to use.
- **Availability:** The system should be always available or have minimal downtime to ensure that showroom operations are not interrupted.
- **Performance:** The system should be responsive and perform well
- **Compatibility:** The system should be compatible with the showroom's other systems and tools.

#### 2.2.5-3 Technical Requirements

- **Back end** – Node.js, express
- **Front end** – React js
- **Database** – Mongo DB

### **2.2.6 Transport Management System [IT21182846 – Chamaleen D.B.N]**

#### 2.2.6-1 Functional Requirements

Transport Management System is a software solution that is designed to manage and optimize the transportation of goods. In this function all the owners can add their vehicle through the system, owner must enter their credentials and log in to their own profile. Then they click the add vehicle button and give the vehicle information and fill that details and submits the form then vehicle owner must wait to be getting verification status from site owner. Within vehicle owner dashboard it displays as the activate but the admin not giving the access the profile display as pending. After the site owner accepts, the vehicle will be published in the system. If the owner wants to delete or update any information from the vehicle, they can do it manually. Then the customer can view details through the web site and view the number of reservations

and search vehicle if they are giving the any data, then vehicle owner finally generates revenue report as well.

- As a User Stories vehicle owner, I want to add my own property of vehicles so that I can assign services to the customer, and I can earn profit for giving services to them.
- As a vehicle owner requirement as the I want to change vehicle requests so that I can fulfil customer requirements as they want, and I can delete requests and generate revenue report.

#### 2.2.6-2 Non-Functional Requirements

- **Efficiency:** Can help ensure that the transport management system is reliable, scalable, secure, usable, and maintainable. This, in turn, can help improve the overall performance of the system, reduce costs, and enhance the user experience.
- **Availability:** The TMS should always be available for use, with minimal downtime for maintenance or upgrades.
- **Accuracy:** The TMS should be able to handle a large volume of data and transactions without slowing down or crashing
- **Maintainability:** A TMS should be designed with a modular and scalable architecture that allows for easy maintenance and upgrades

#### 2.2.6-3 Technical Requirements

- Back end – Node.js, express
- Front end – React, Bootstrap
- Database – Mongo DB

### **2.2.7 Wasana K.H.I.M [IT21197796 – Wasana K.H.I.M]**

#### 2.2.7-1 Functional Requirements

Access any of the system's features, users must login to the system via sign in page. If login, Users must first sign up and require having a user account. All users have options to register as a normal user, property owners, showroom owners, vehicle owners and lawyers. In this function logging authentication is the way of confirming the identity of a user while they access their profile on a particular platform. The system checks whether the data provided by the users is correct and the validity of the data is also checked. After signing users, all user details will be updated in the user management system. Admin can view all categories of registered users and the details about them. If there are any requirements for adding new users or admins, users can be added by giving users' first name, last name, NIC, phone number, address and can be made changes if have some updates about user details. Furthermore, can be deleted user account If user violated system rules and policies or if they need to remove their account. Searching any

data can get results. Provide a monthly report containing data on all users who have registered within a given month for the system.

- As a system admin, I want to manage all the system users after signing up system so that I can analyze how many users are registered to the system.

#### 2.2.7-2 Non-Functional Requirements

- **User-friendly:** The user management system should be user-friendly and intuitive, with an easy-to-use interface and clear instructions.
- **Accuracy:** The system should be able to validate separate categories of the user details when they sign up to the system.
- **Maintainability:** Admin can access user accounts. If violated system rules or polices, admin can delete those user's account.
- **Security:** The system can use key derivation to password encryption and encourage users to create strong passwords that avoid reusing passwords across multiple accounts. The system can enforce password complexity requirements, such as a minimum length, a mix of uppercase and lowercase letters, numbers, and special characters.

#### 2.2.7-3 Technical Requirements

- Back end – Node.js, express.
- Front end – React, Bootstrap
- Data base – Mongo DB

### **2.2.8 Legal Contact Management System** [IT21240706 – Dhananjana B.K.T]

#### 2.2.8-1 Functional Requirements

Legal contact can be reached through our system, as legal documents are necessary when renting a home. According to the Legal Contact Management provides to users for contact lawyers physically because not satisfaction security purpose for legal documents. A lawyer is the primary user of this function. After lawyers have registered to the system who want to publish their personal qualifications and contact details before publishing, they must pay a subscription fee. After paying, the service listing form will be enabled for lawyers. The lawyer fills and submits the form. The service will be published in the system. Then represent all details and user can contact lawyers by that information. The lawyer can update their personal qualifications and contact details.

- As a lawyer I want to publish my personal qualifications and contact details on the system so that I can increase my client base and revenue.

#### 2.2.8-2 Non-Functional Requirements

- **User friendly:** provide a user-friendly environment interface for lawyers to access their details easily and users can contact lawyers quickly.
- **Easy to access:** Lawyer can easily change and add their details through the system.
- Availability: There are numerous lawyers available for users to contact and provide their services.
- **Usability:** The system should be easy to use and intuitive, allowing users to access the information and support they need quickly and easily.

#### 2.2.8-3 Technical Requirements

- **Back end** – Node.js, express
- **Front end** – React, Bootstrap
- **Database** – Mongo DB

## 2.3 Requirement Modeling

### 2.3.1 Use case Diagram of the system

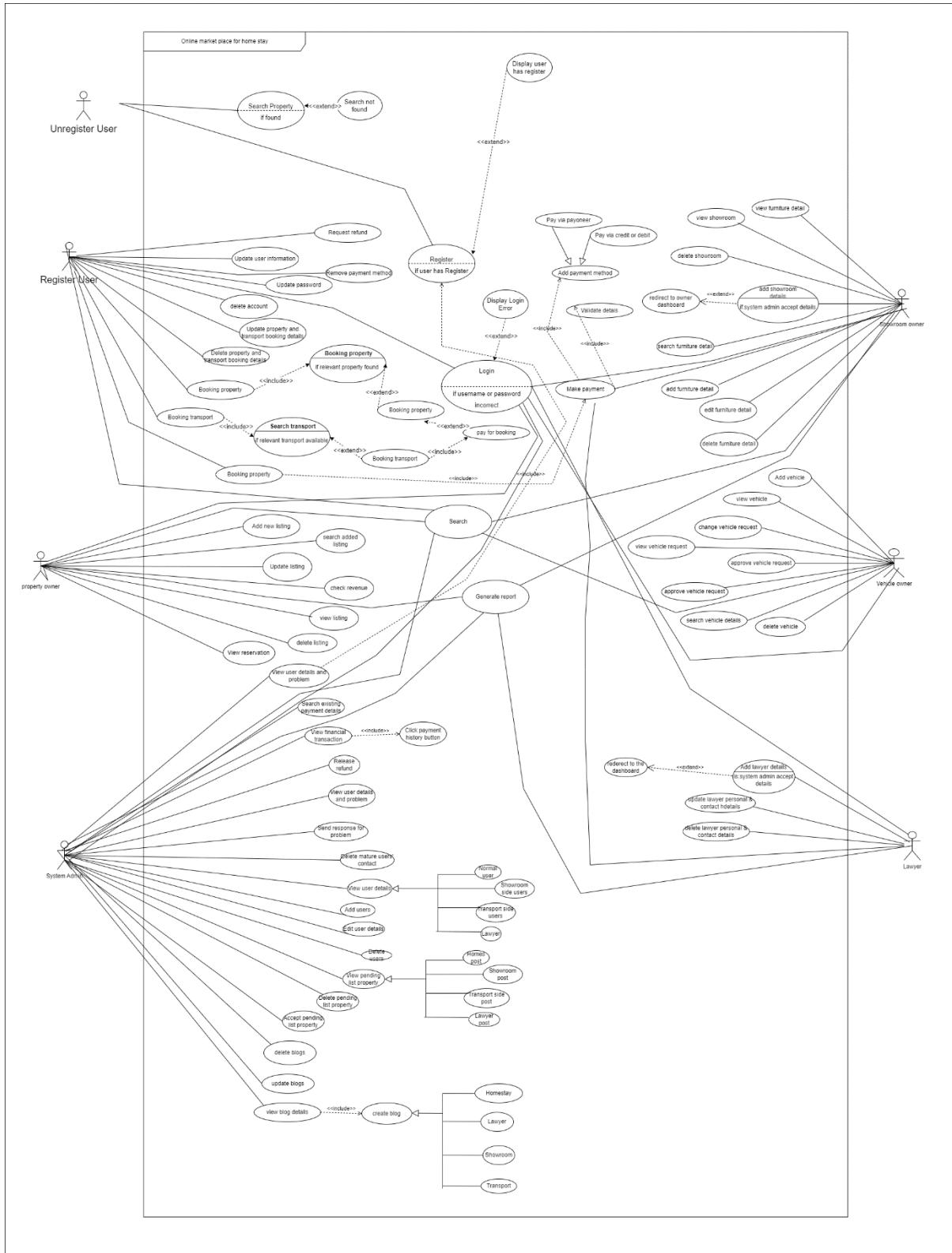


Figure 2. 1 – Use case diagram of the system.

## 2.3.1-1 Use Case Diagram of the Listing Management [IT21166488 - Maleesha K.L.D.D.S ]



Figure 2. 2 – Use case diagram of the Listing Management

## 2.3.1-2 Use Case Diagram of the Payment Management [IT21183768 – Heshan W.A.S ]

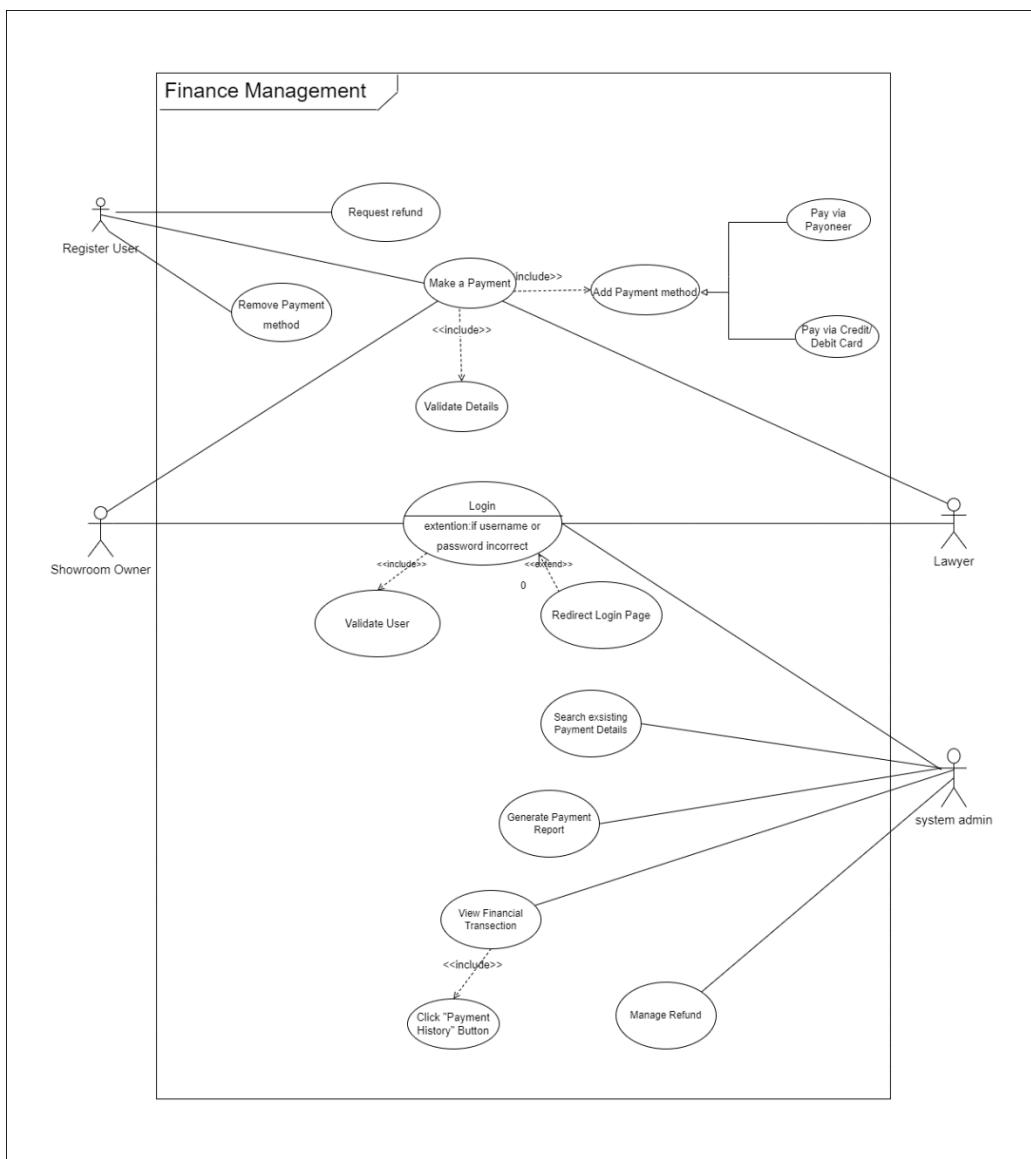


Figure 2. 3 - Use Case Diagram of the Payment Management

## 2.3.1-3 Use Case Diagram of the Blog Management [IT21186424 – Bandara N.G.S.S.M ]

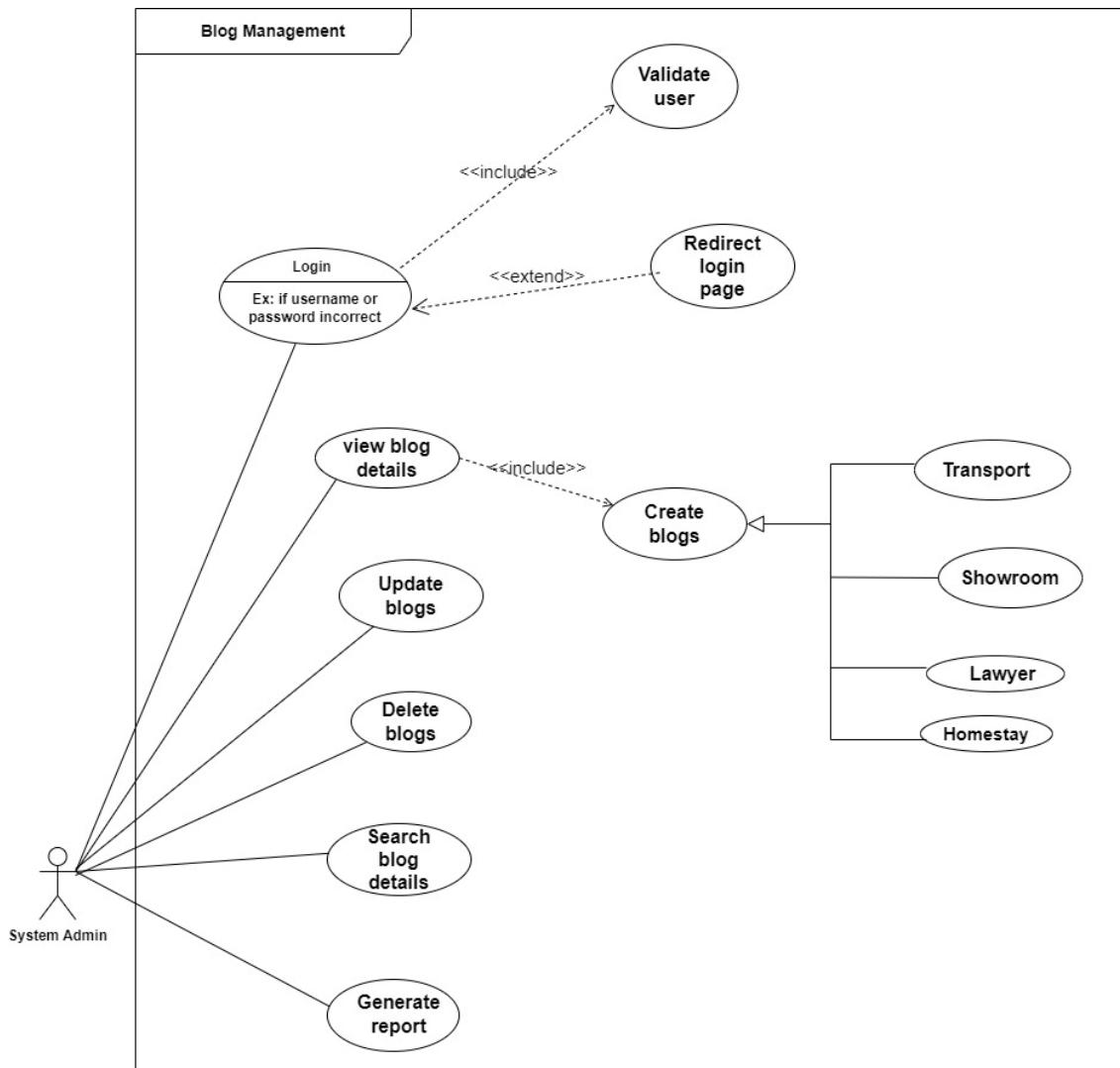


Figure 2. 4 - Use Case Diagram of the Blog Management

## 2.3.1-4 Use Case Diagram of the Customer [IT21233562 – Ranawaka W.E.I ]

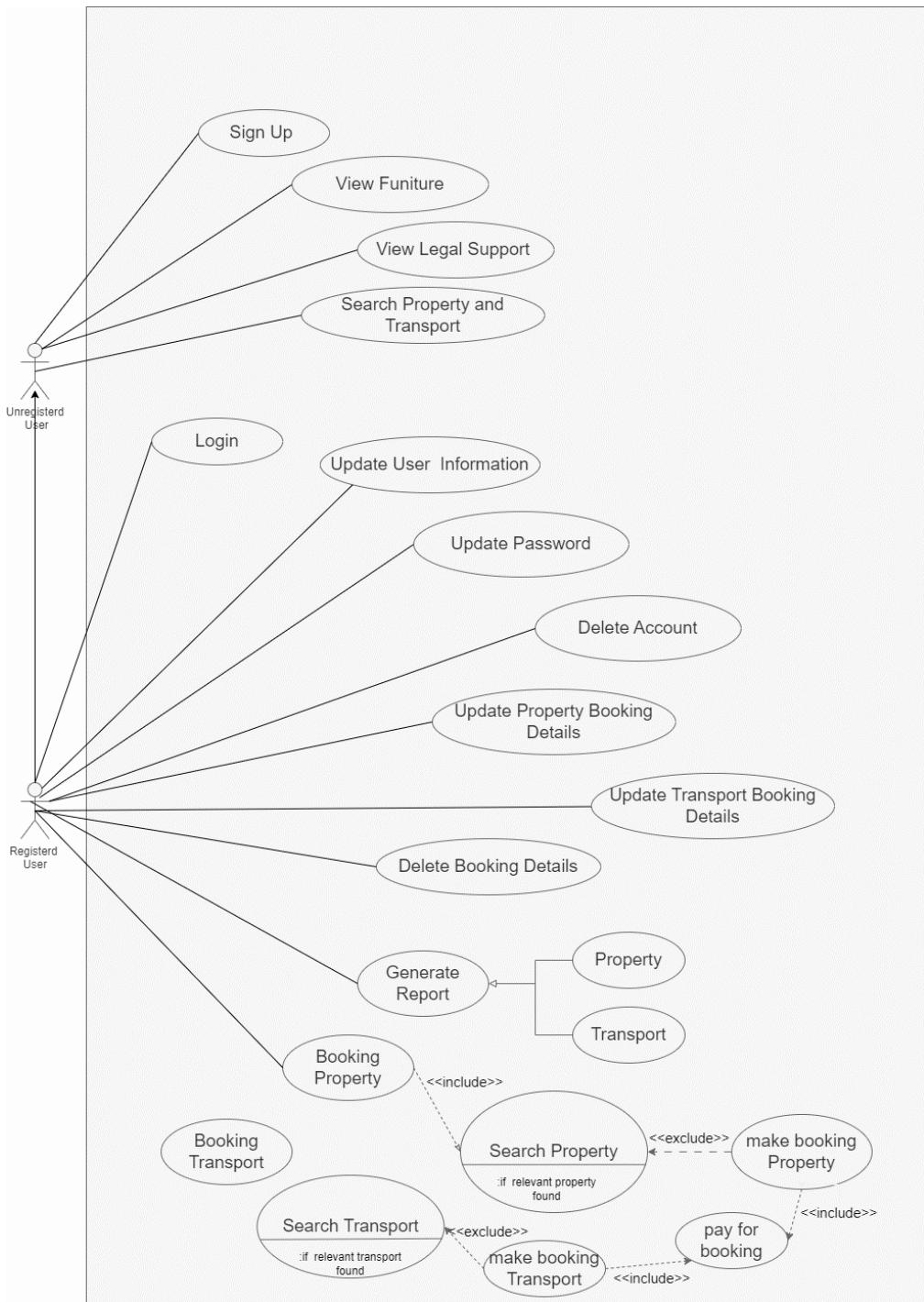


Figure 2. 5 - Use Case Diagram of the Customer Management

*2.3.1-5 Use Case Diagram of the Furniture Showroom Management  
[IT21184376 – Jayathilaka A.W.N.M ]*



*Figure 2. 6 - Use Case Diagram of the Furniture Showroom Management*

## 2.3.1-6 Use Case Diagram of the Transport Management [IT21182846 – Chamaleen D.B.N ]

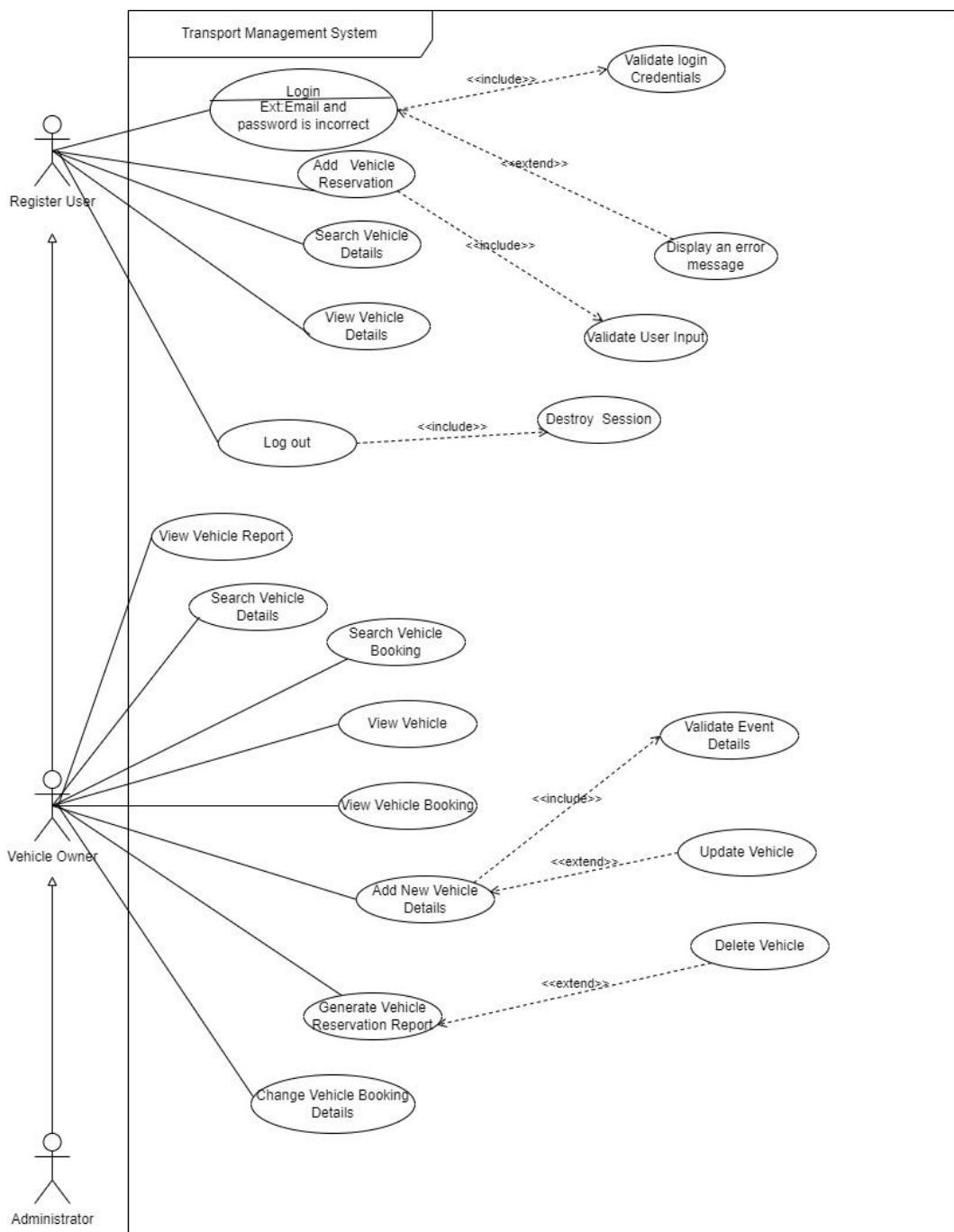
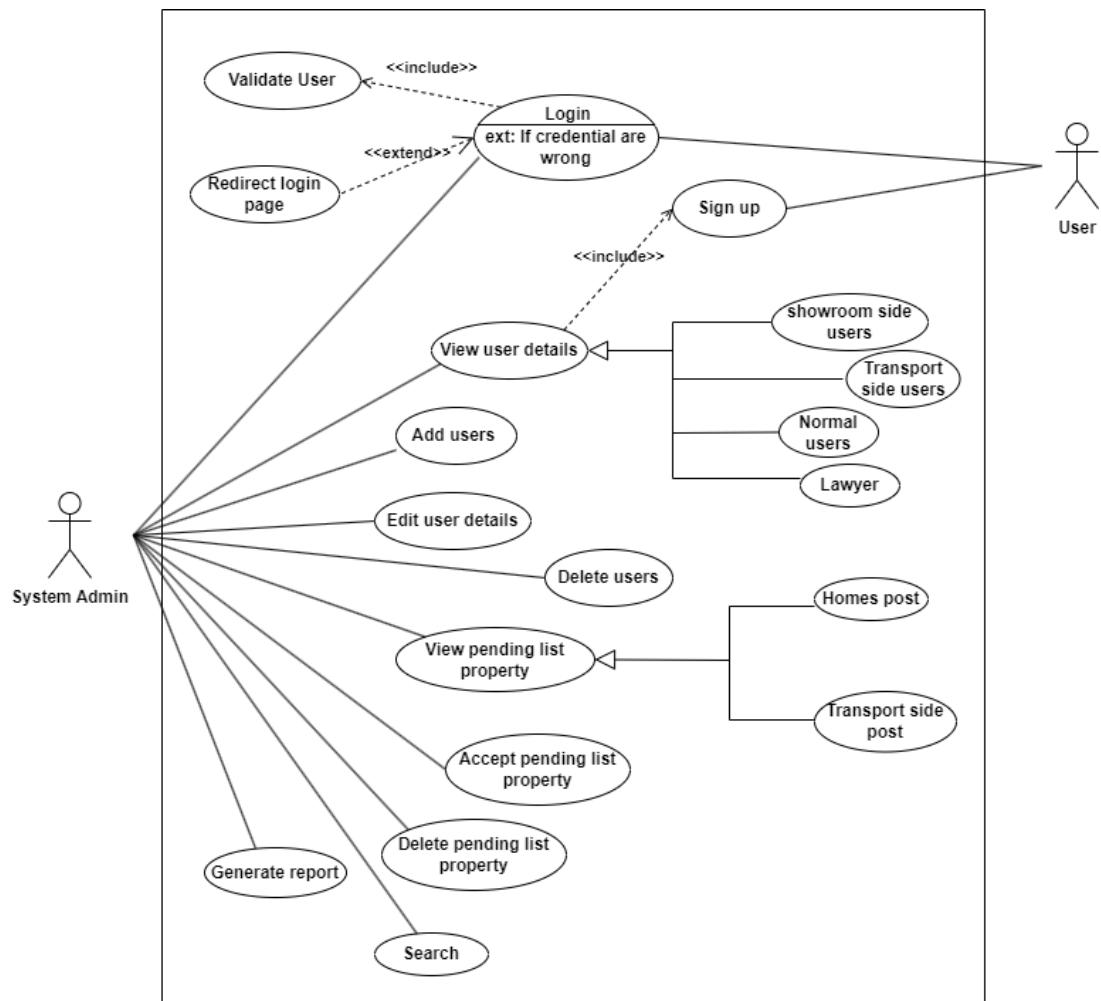


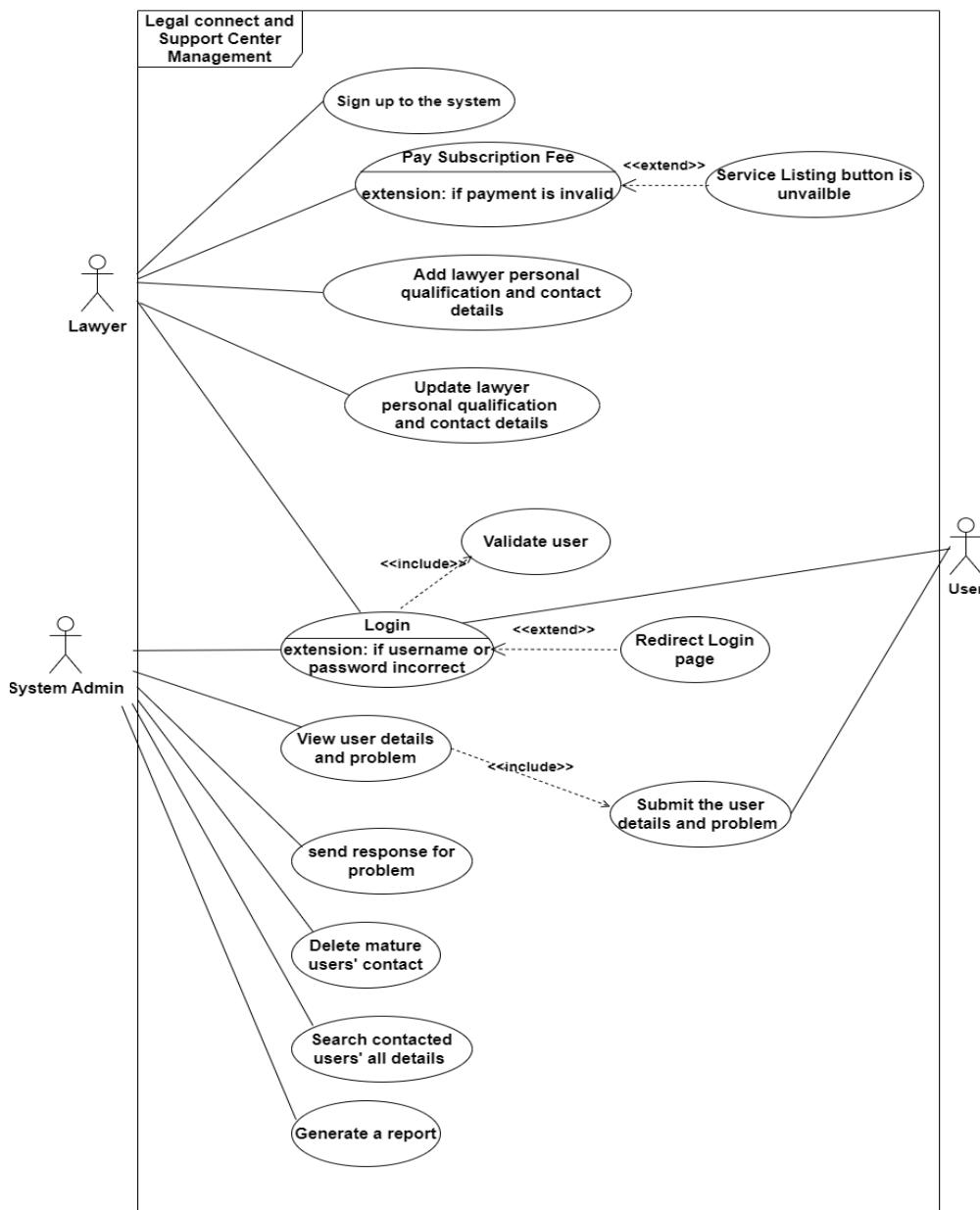
Figure 2. 7 - Use Case Diagram of the Transport Management

*2.3.1-7 Use Case Diagram of the User Management and Pending List Management  
[IT21197796 – Wasana K.H.I.M ]*



*Figure 2. 8 - Use Case Diagram of the Users and Pending List Management*

*2.3.1-8 Use Case Diagram of the Legal contact and Support Center Management  
[IT21240706 – Dhananjana B.K.T]*



*Figure 2. 9 - Use Case Diagram of the Legal Contact and Support Center Management*

### 2.3.2 Use case Diagram of the system

2.3.2-1 Use Case Diagram of the Listing Management [IT21166488 - Maleesha K.L.D.D.S]

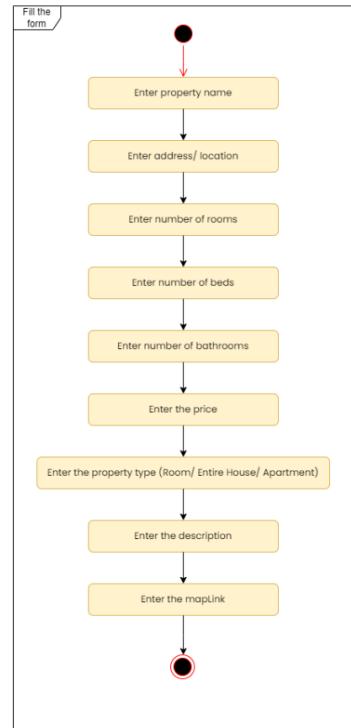
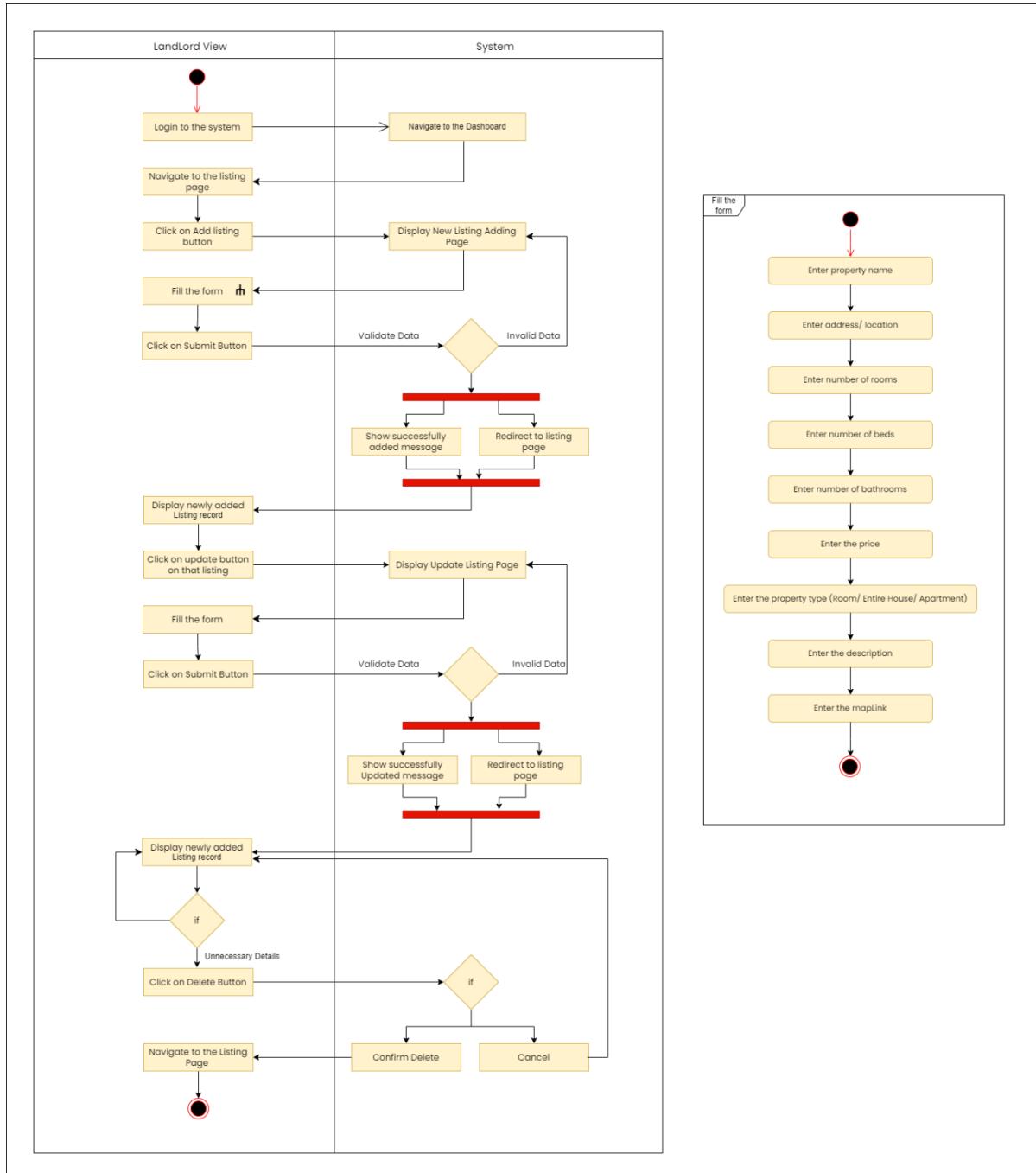


Figure 2. 10 - Activity Diagram of the Listing Management

## 2.3.2-2 Use Case Diagram of the Payment Management [IT21183768 – Heshan W.A.S ]

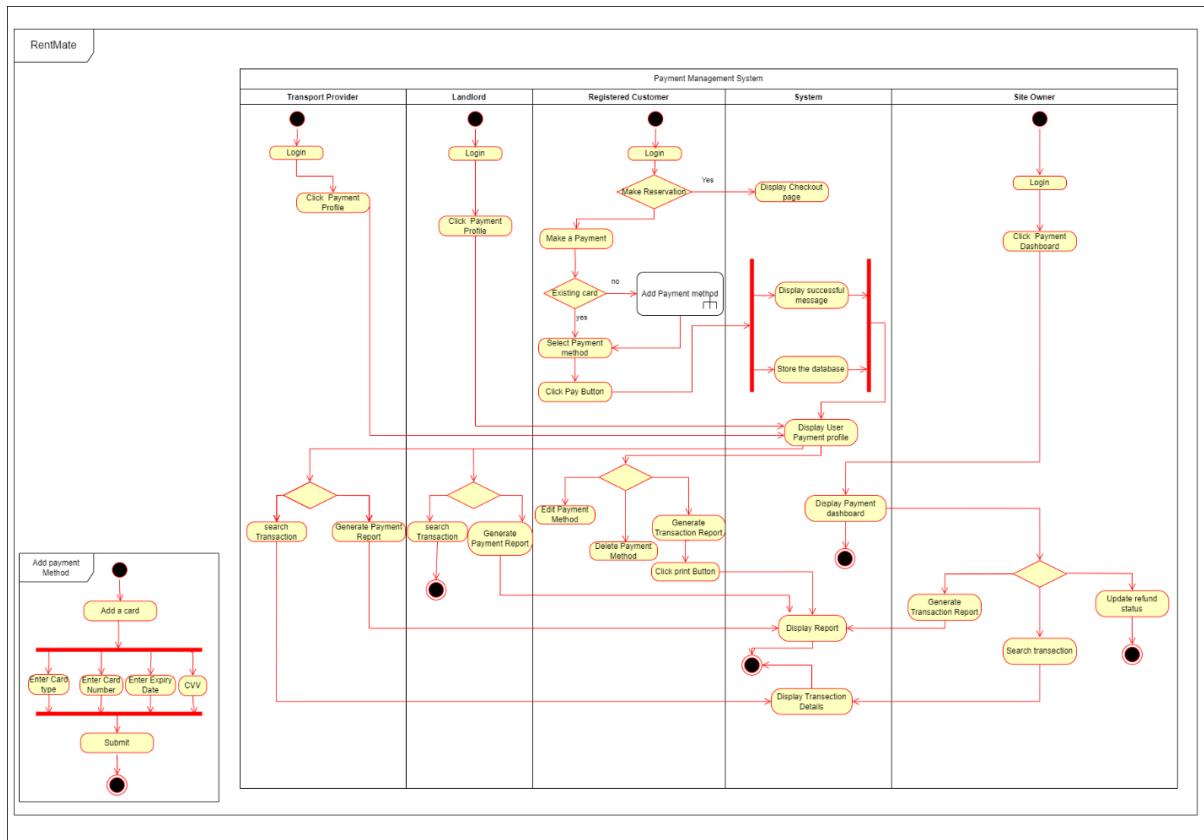


Figure 2. 11 - Activity Diagram of the Payment Management

## 2.3.2-3 Use Case Diagram of the Blog Management [IT21186424 – Bandara N.G.S.S.M ]

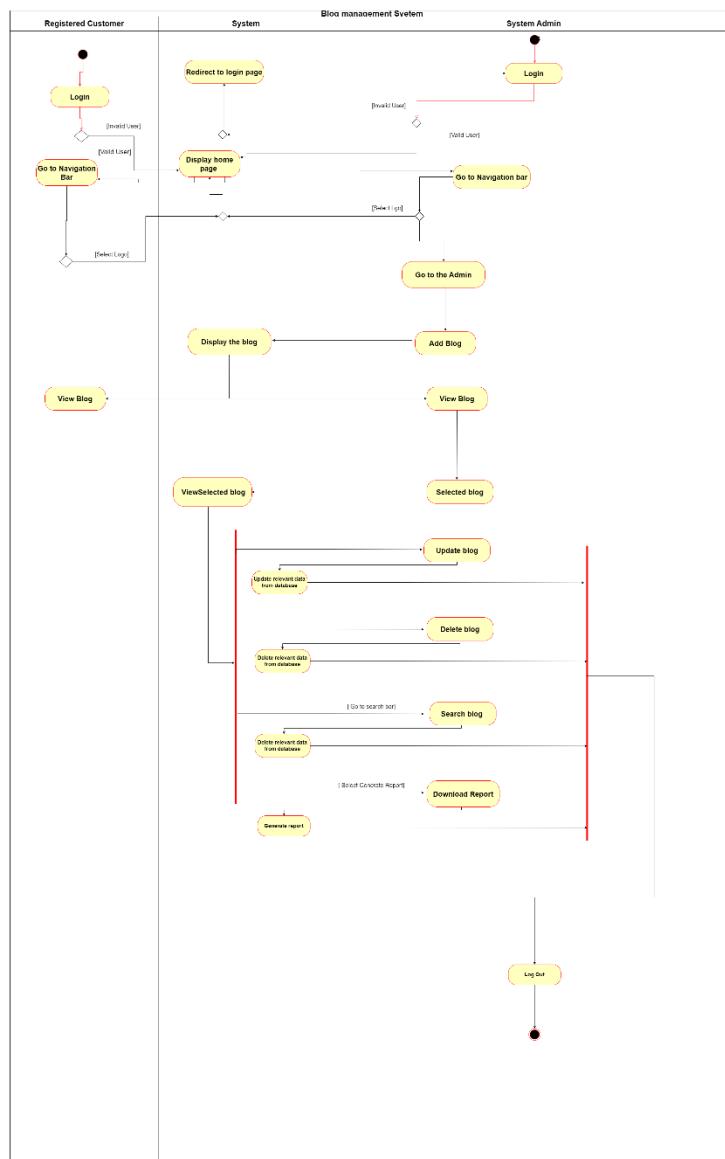


Figure 2. 12 - Activity Diagram of the Blog Management

## 2.3.2-4 Use Case Diagram of the Customer [IT21233562 – Ranawaka W.E.I ]

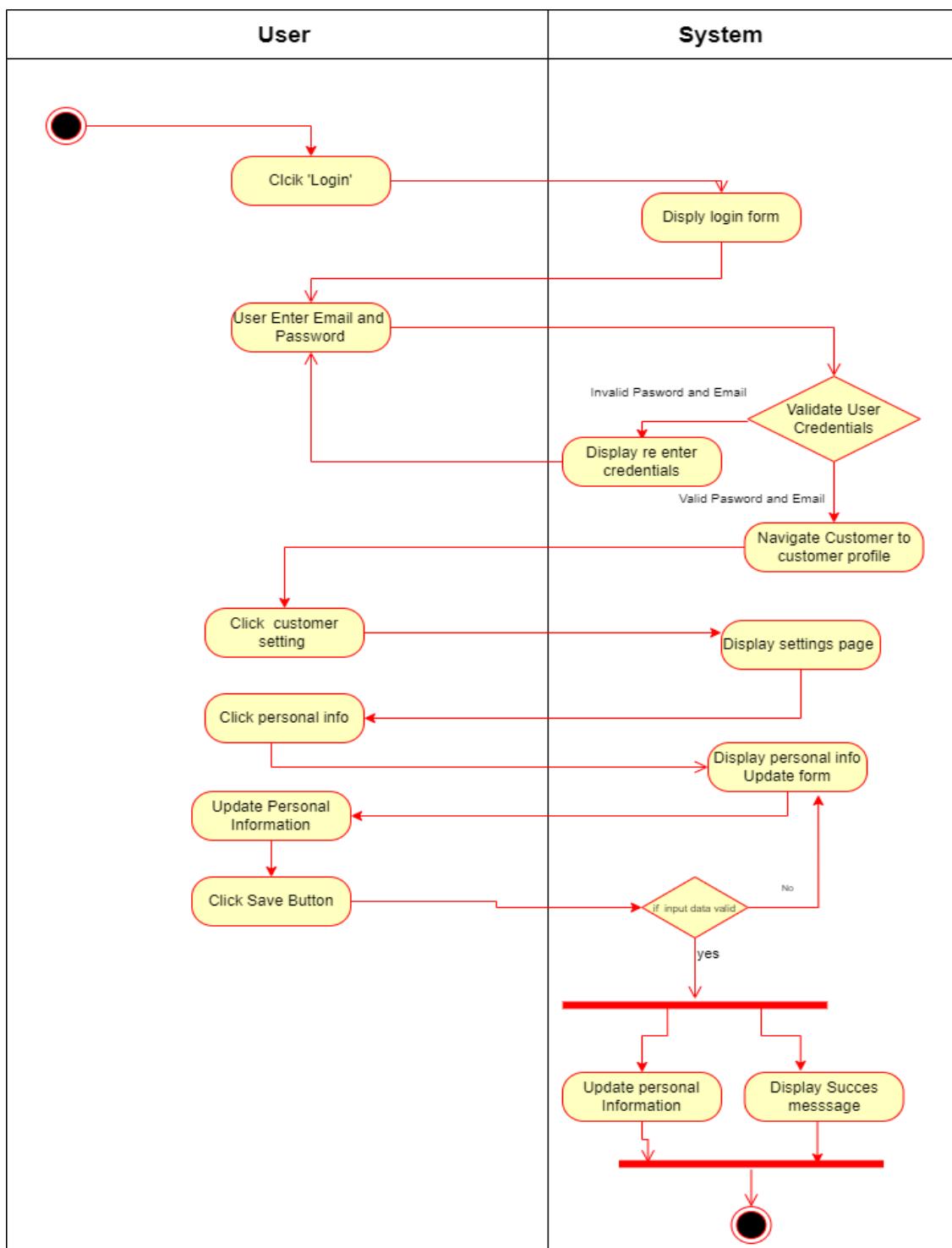


Figure 2. 13 - Activity Diagram of the Customer Information Update

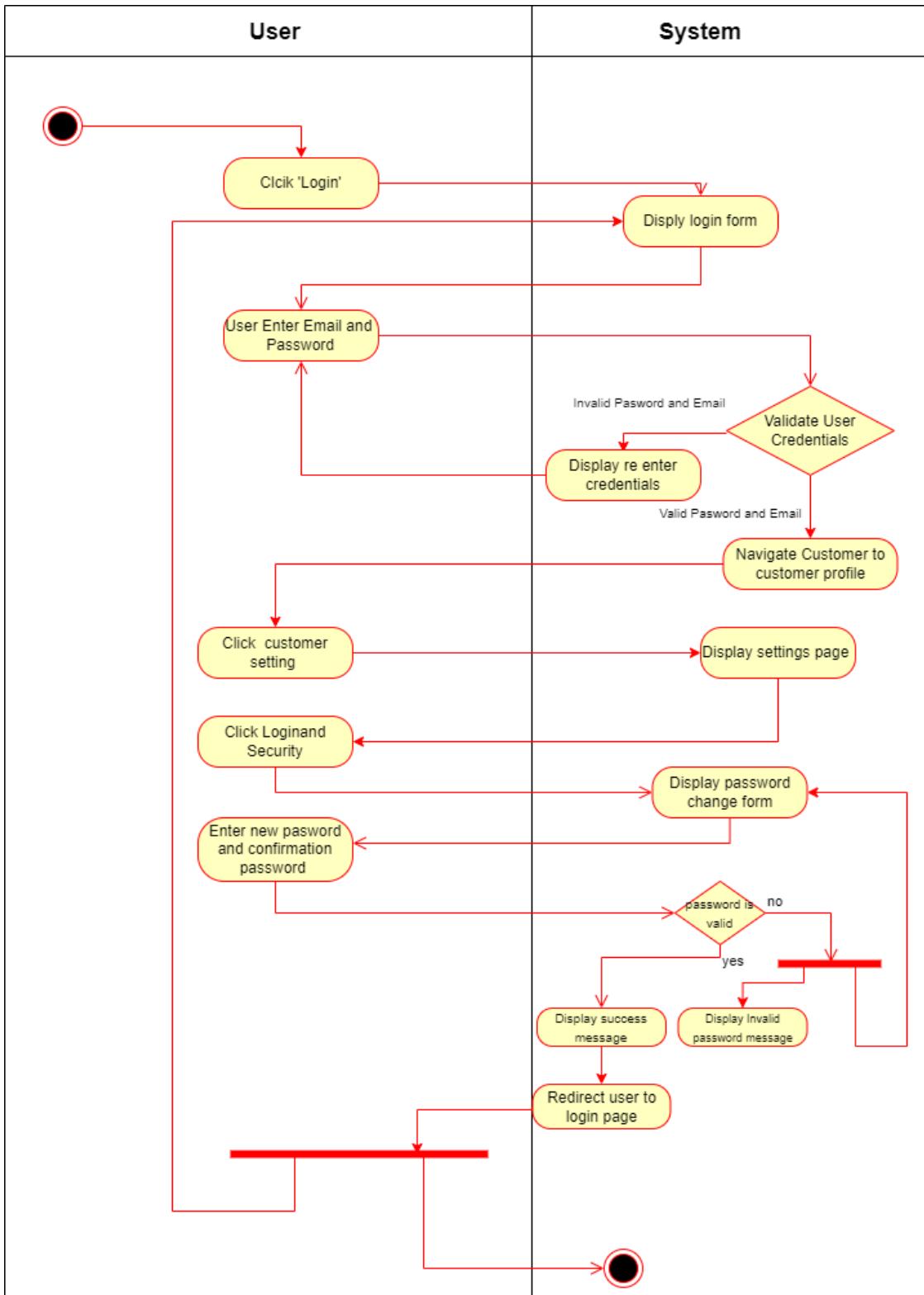
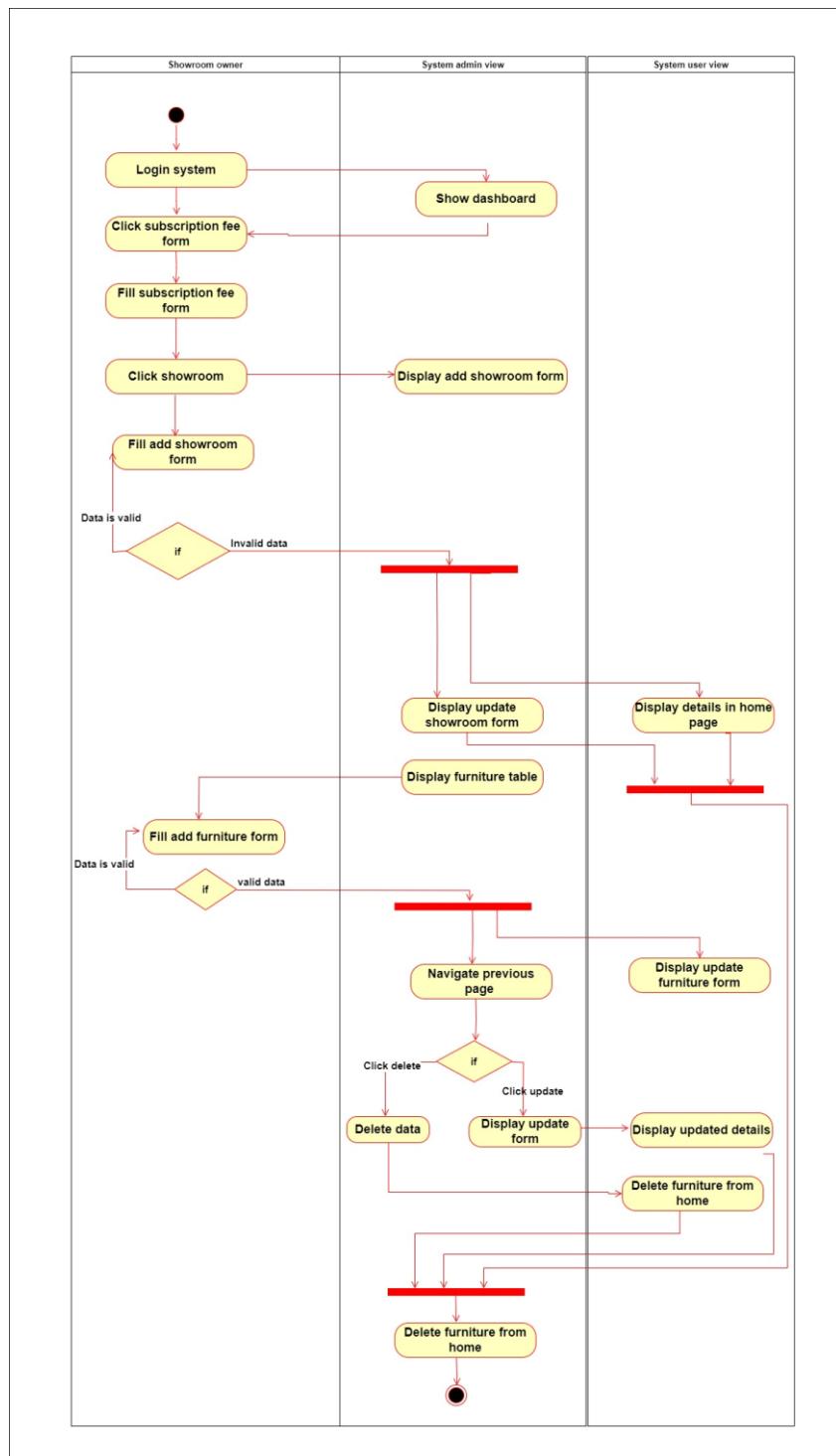


Figure 2. 14 - Activity Diagram of the Customer Password Update

*2.3.2-5 Use Case Diagram of the Furniture Showroom Management [IT21184376 – Jayathilaka A.W.N.M ]*



*Figure 2. 15 - Activity Diagram of the Furniture Showroom Management*

## 2.3.2-6 Use Case Diagram of the Transport Management [IT21182846 – Chamaleen D.B.N ]

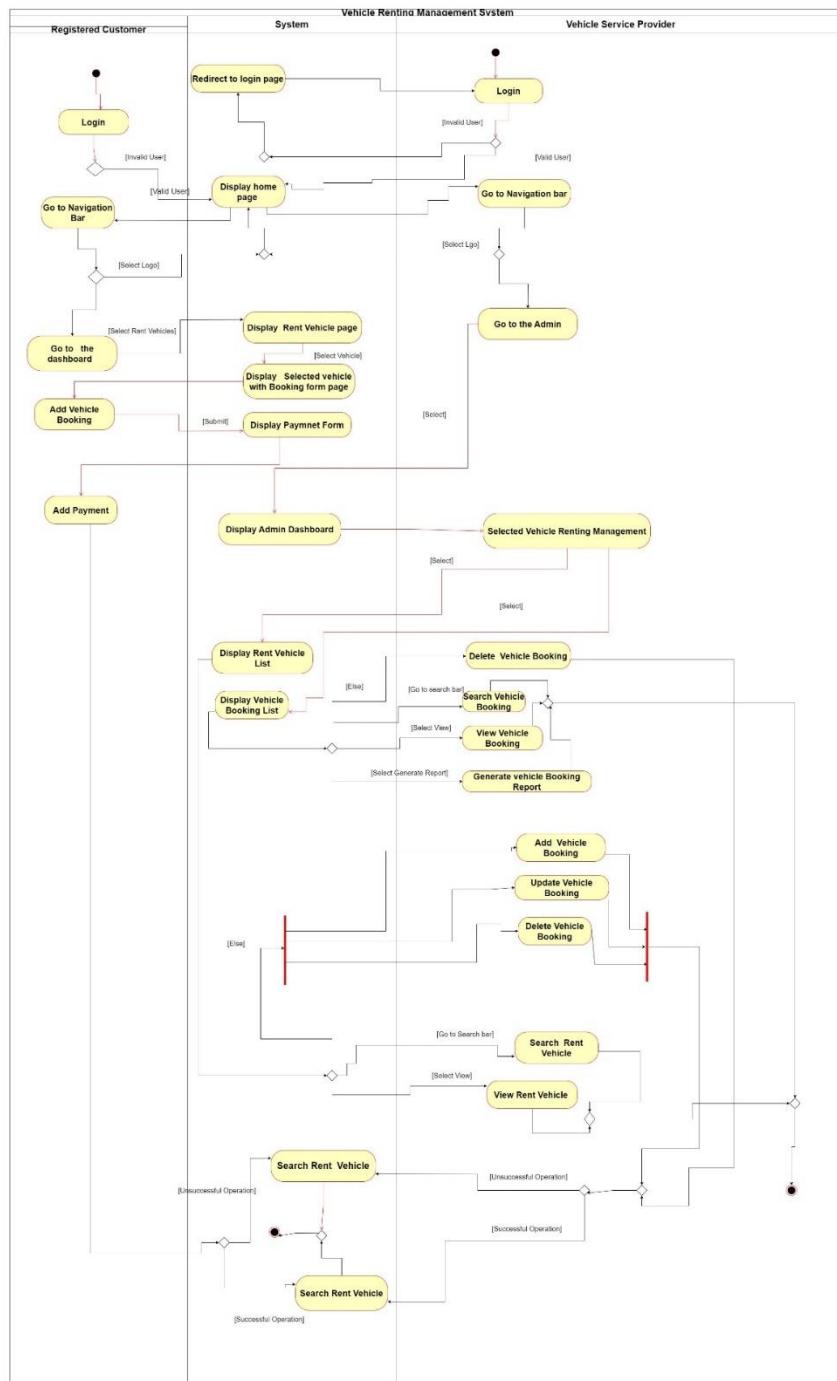


Figure 2. 16 - Activity Diagram of the Vehicle Management

**2.3.2-7 Use Case Diagram of the User Management and Pending List Management**  
**[IT21197796 – Wasana K.H.I.M ]**

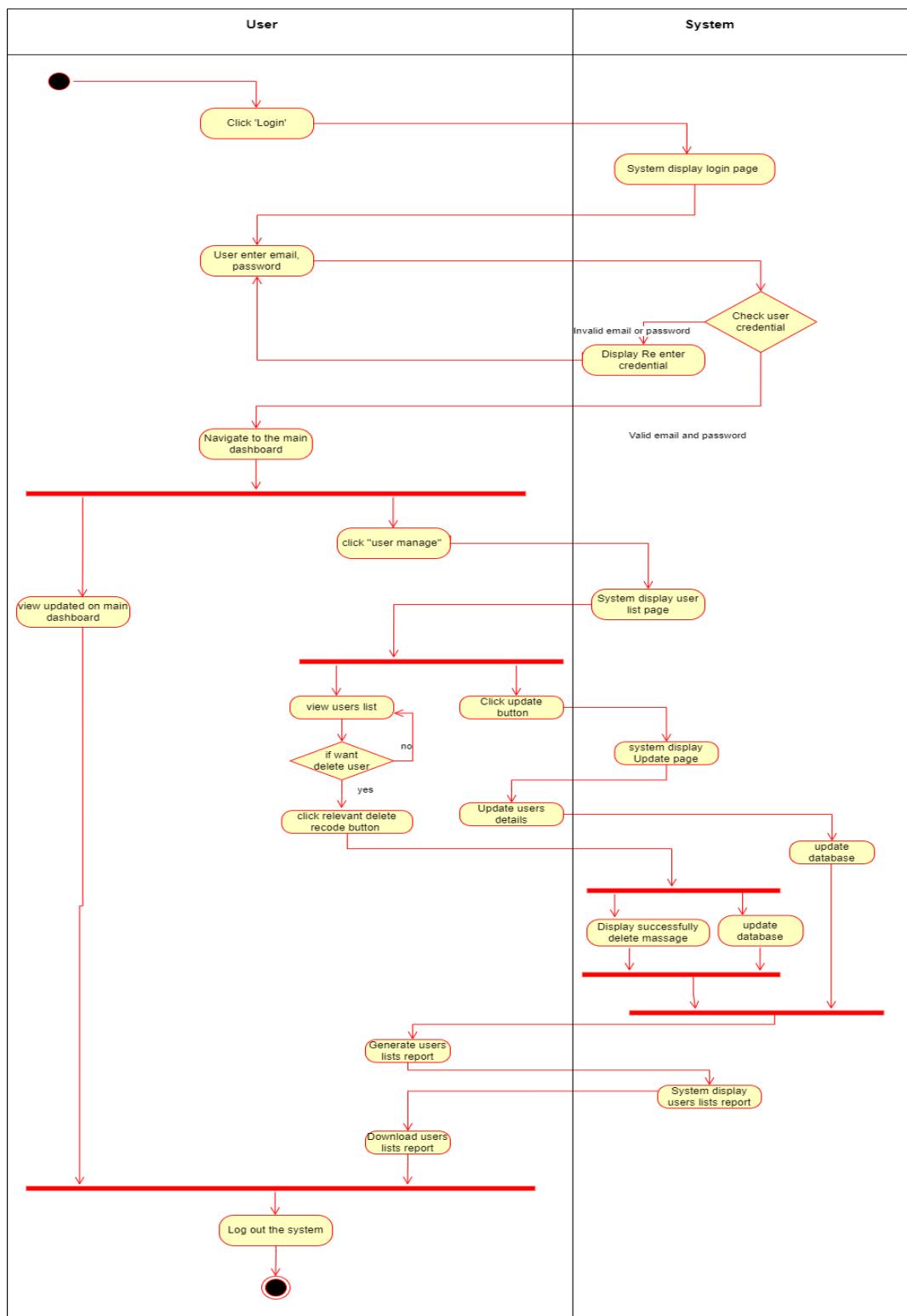


Figure 2. 17 - Activity Diagram of the User management

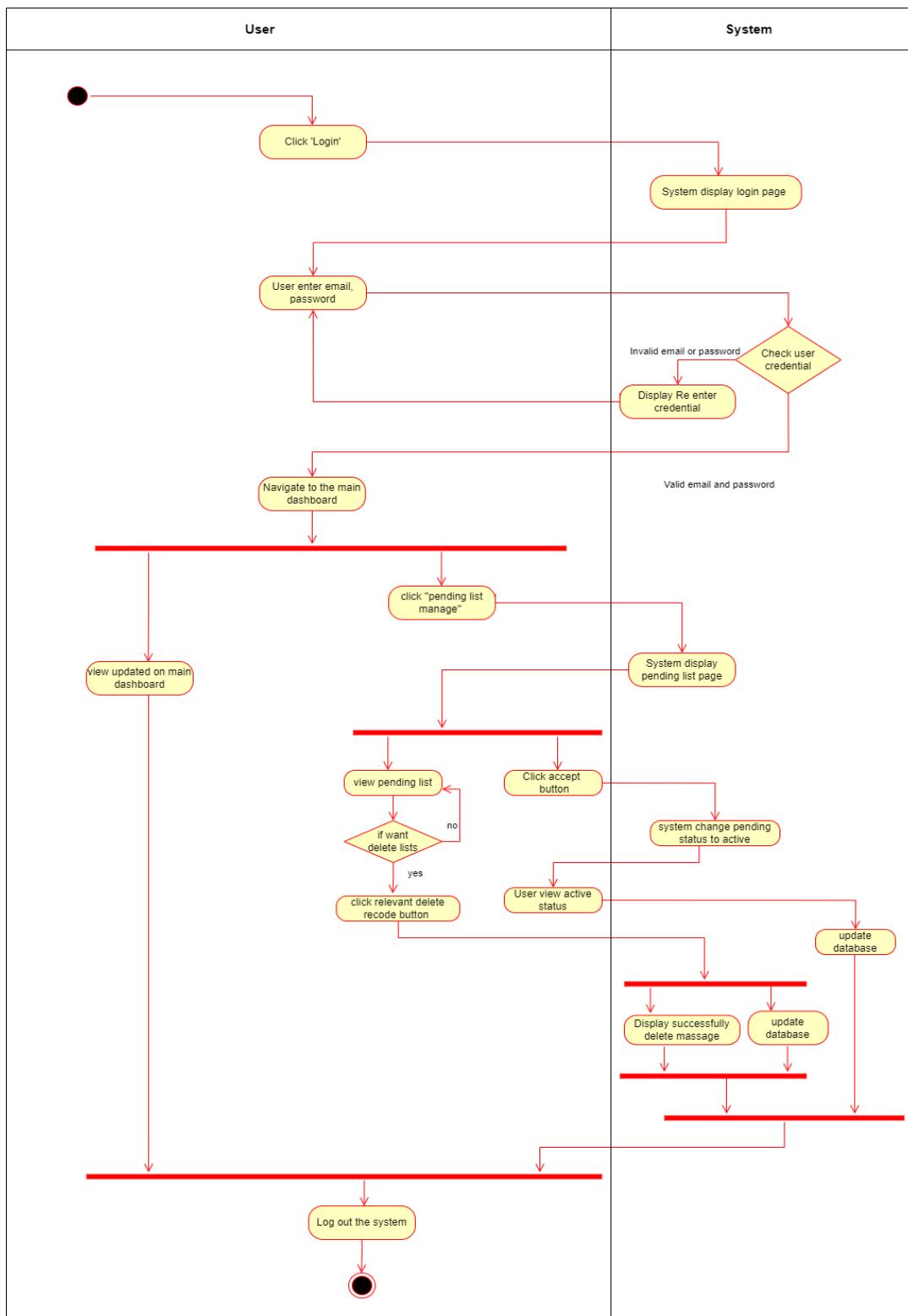


Figure 2. 18 - Activity Diagram of the Pending lists managements

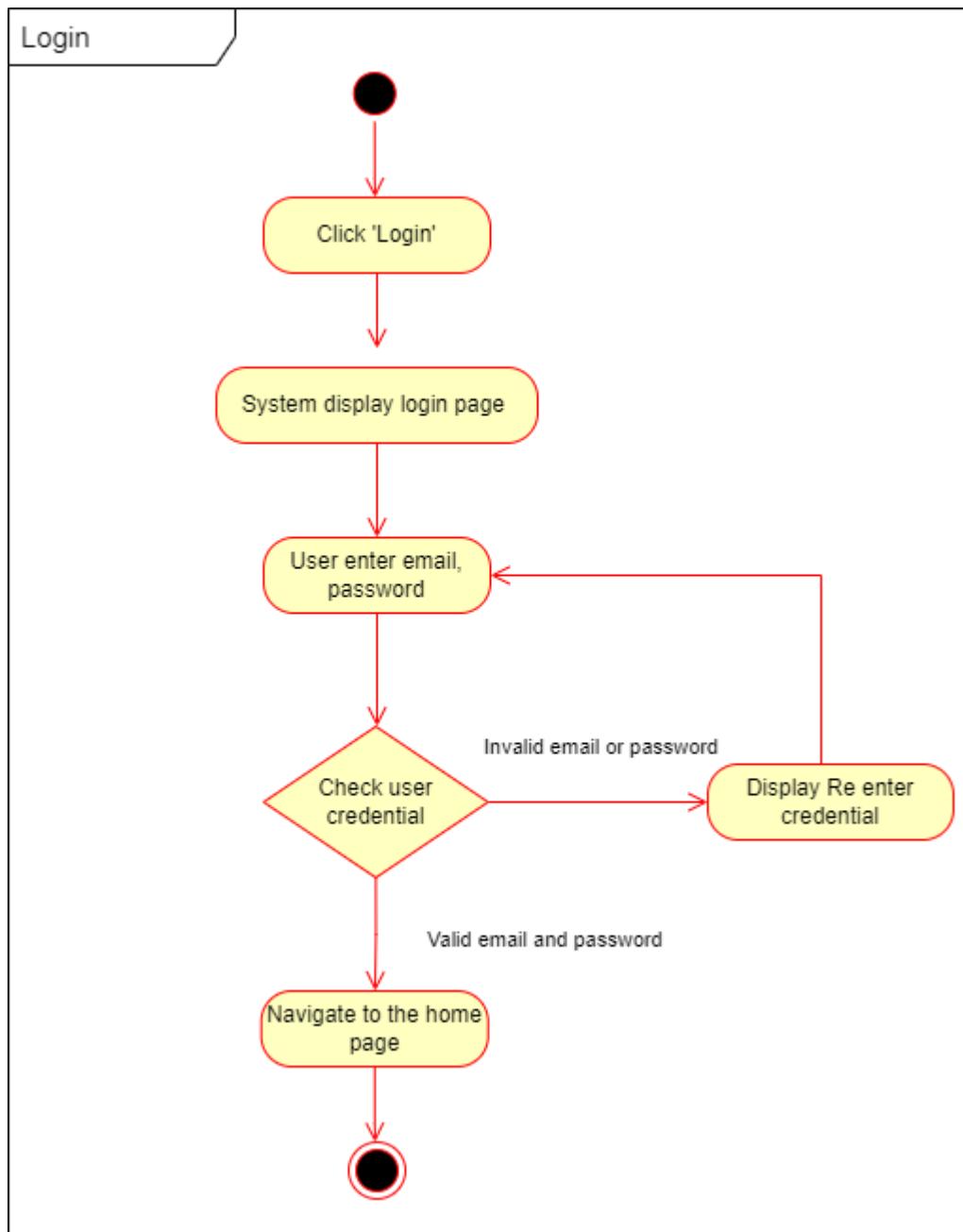


Figure 2. 19 - Activity Diagram of the Login

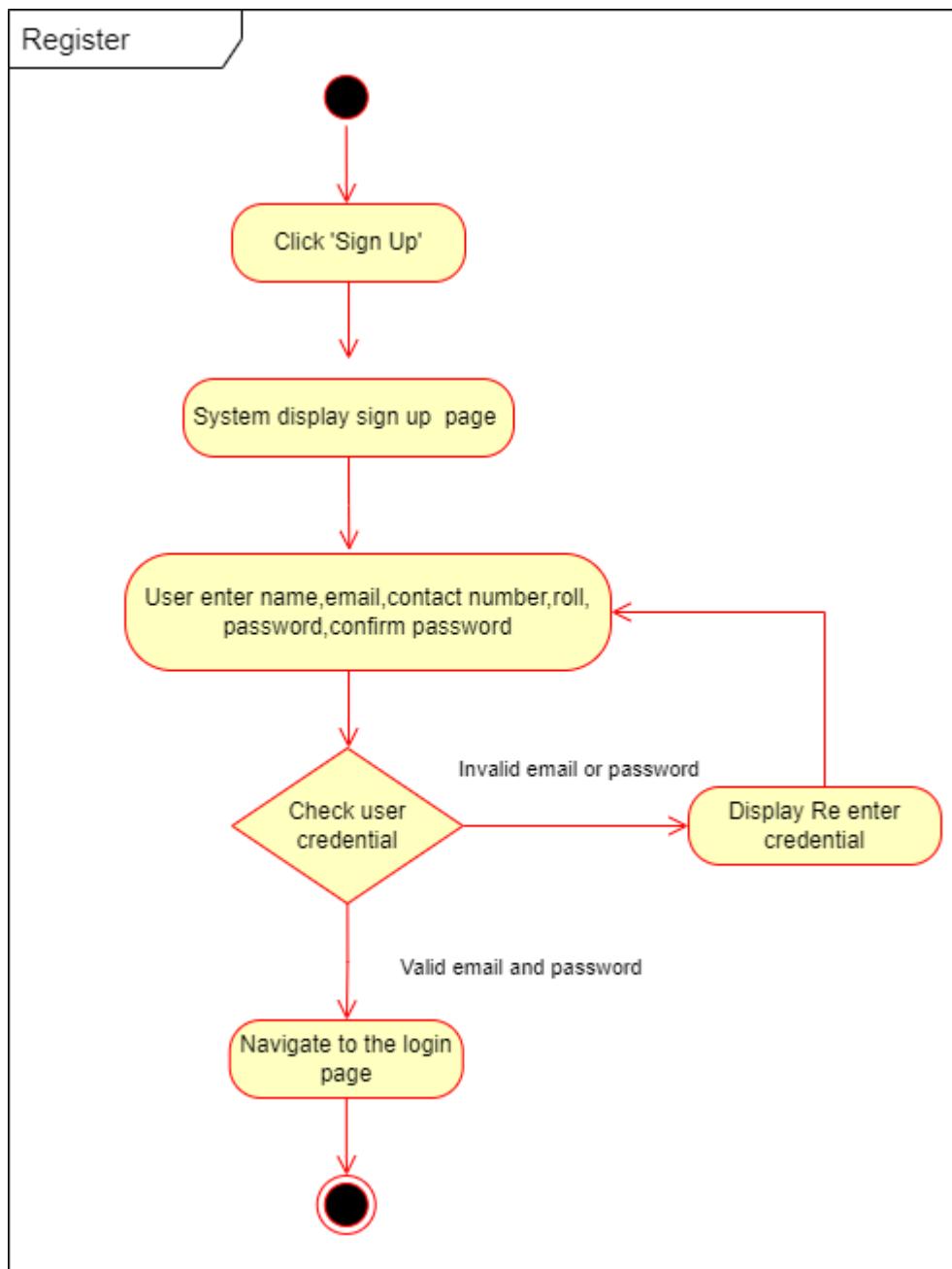


Figure 2. 20 - Activity Diagram of the Register

*2.3.1-8 Use Case Diagram of the Legal contact and Support Center Management  
[IT21240706 – Dhananjana B.K.T]*

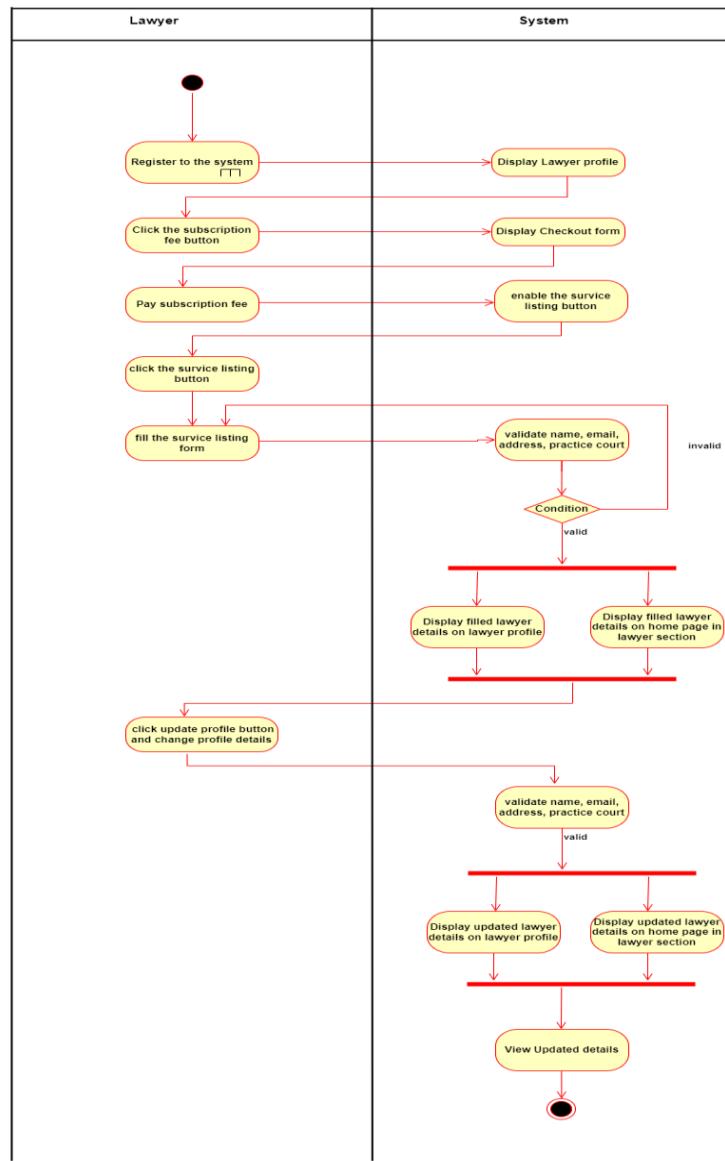


Figure 2. 21 -Activity Diagram of the Legal Contact Management

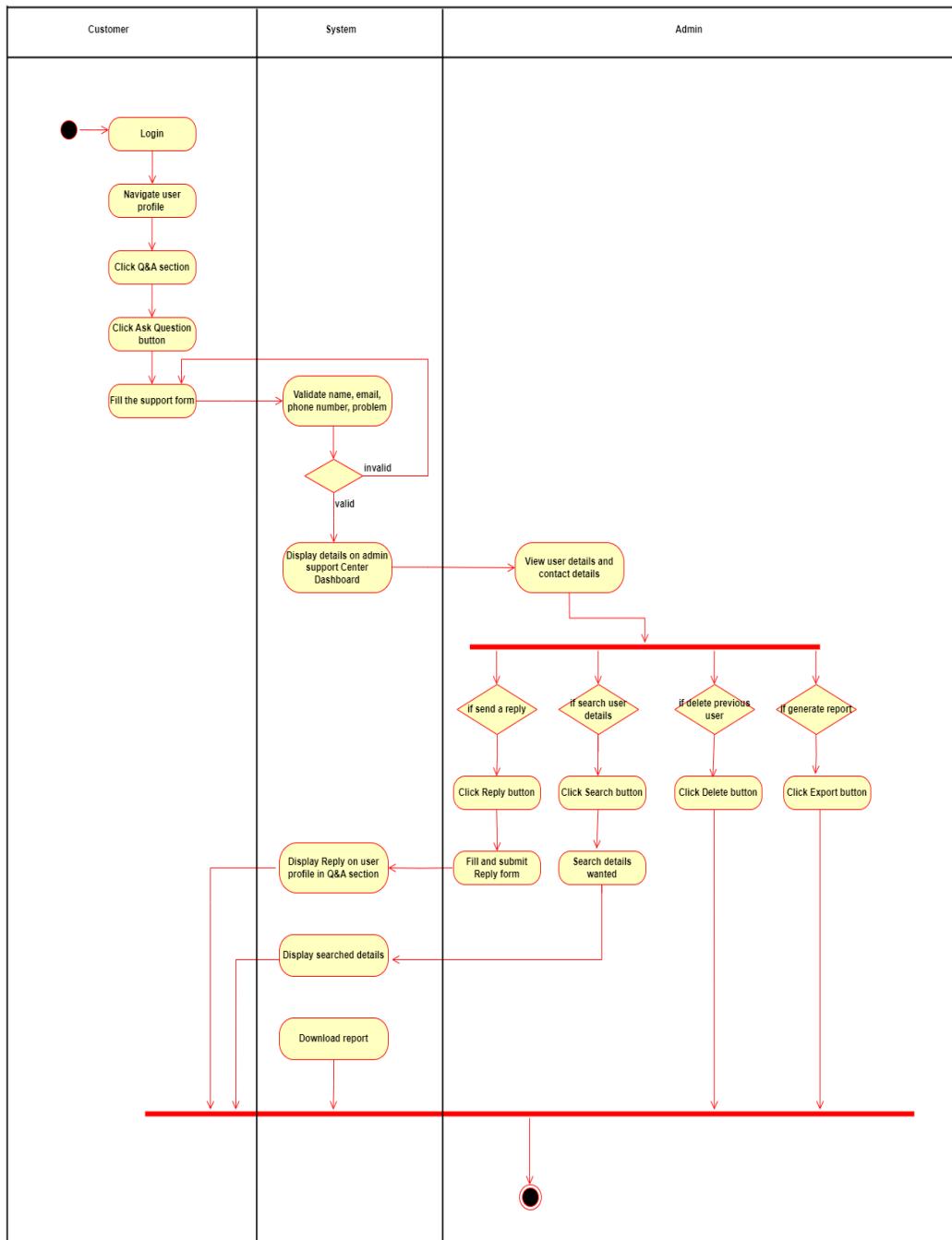


Figure 2. 22 - Activity Diagram of the Support Center Management

# Chapter 03 – Design and Development

## 3.1 ER Diagram

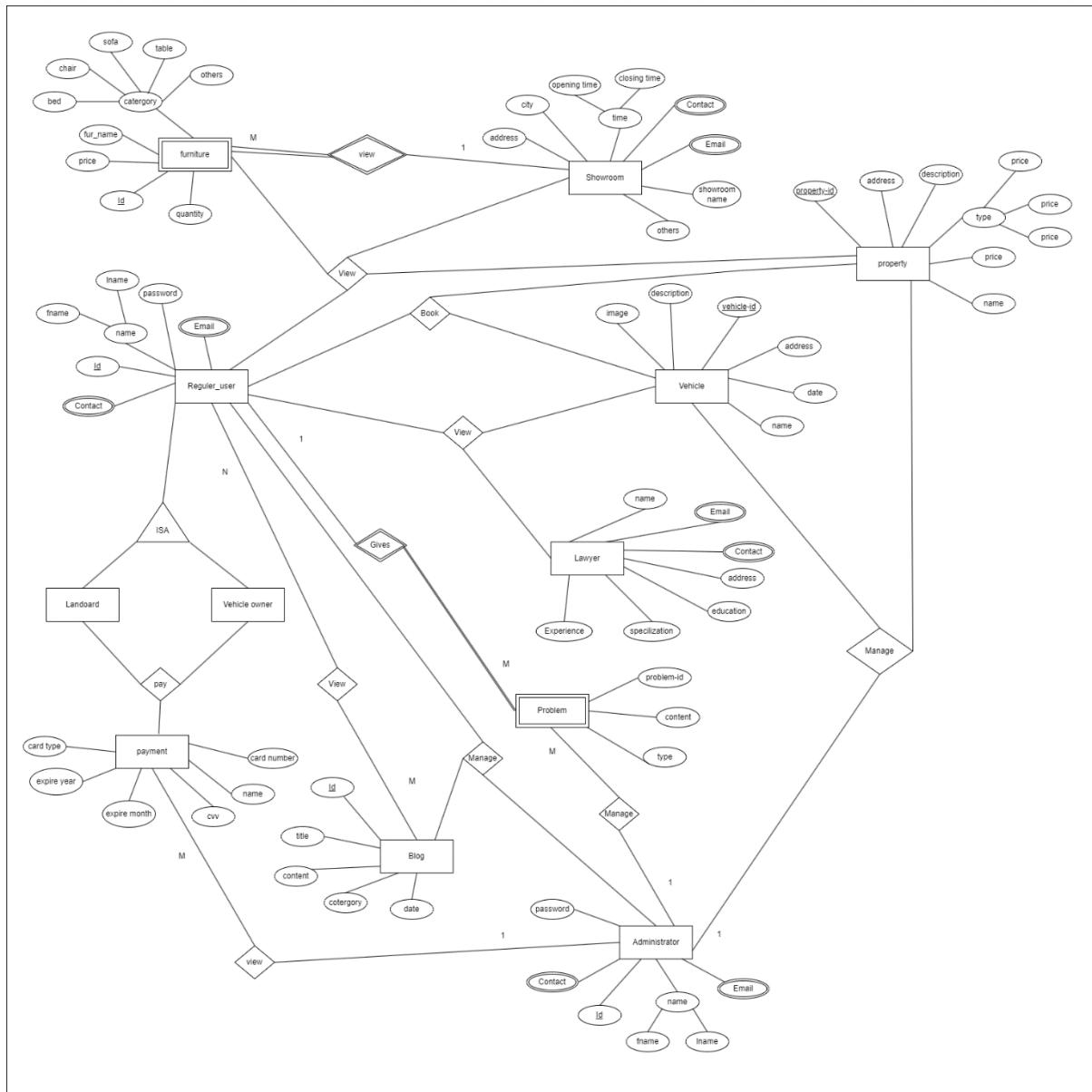


Figure 3. 1 -ER Diagram of the system

### 3.2 High level Architecture

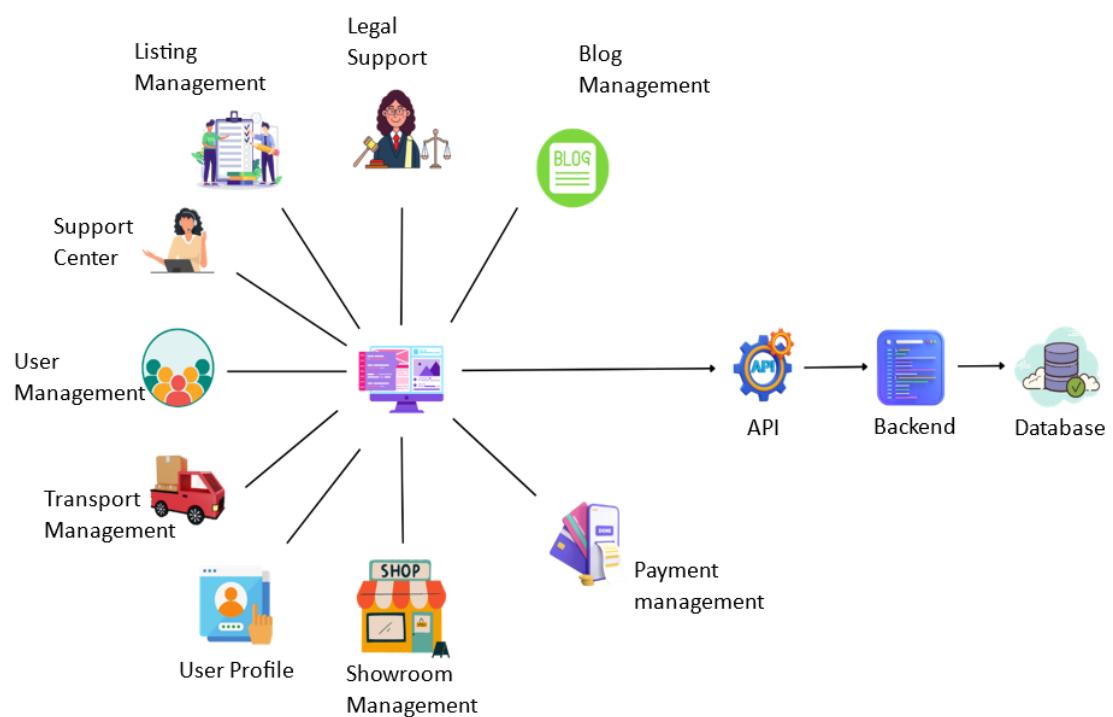


Figure 3. 2 - High level architecture of the system

### 3.3 Class diagram

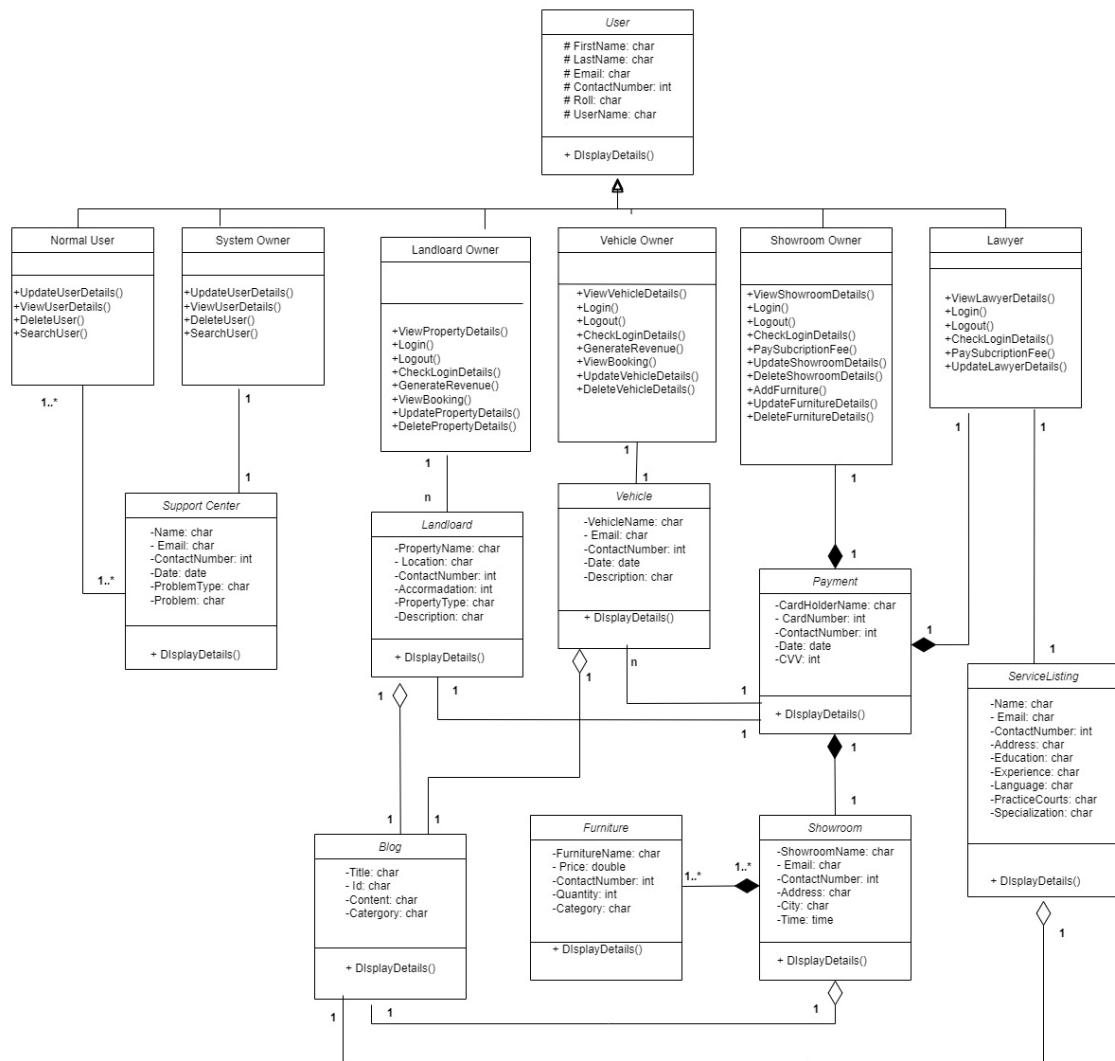


Figure 3. 3 - Class Diagram of the system

### 3.4 User Interfaces

*Listing Management System [IT21166488 – Maleesha K.L.D.D.S]*

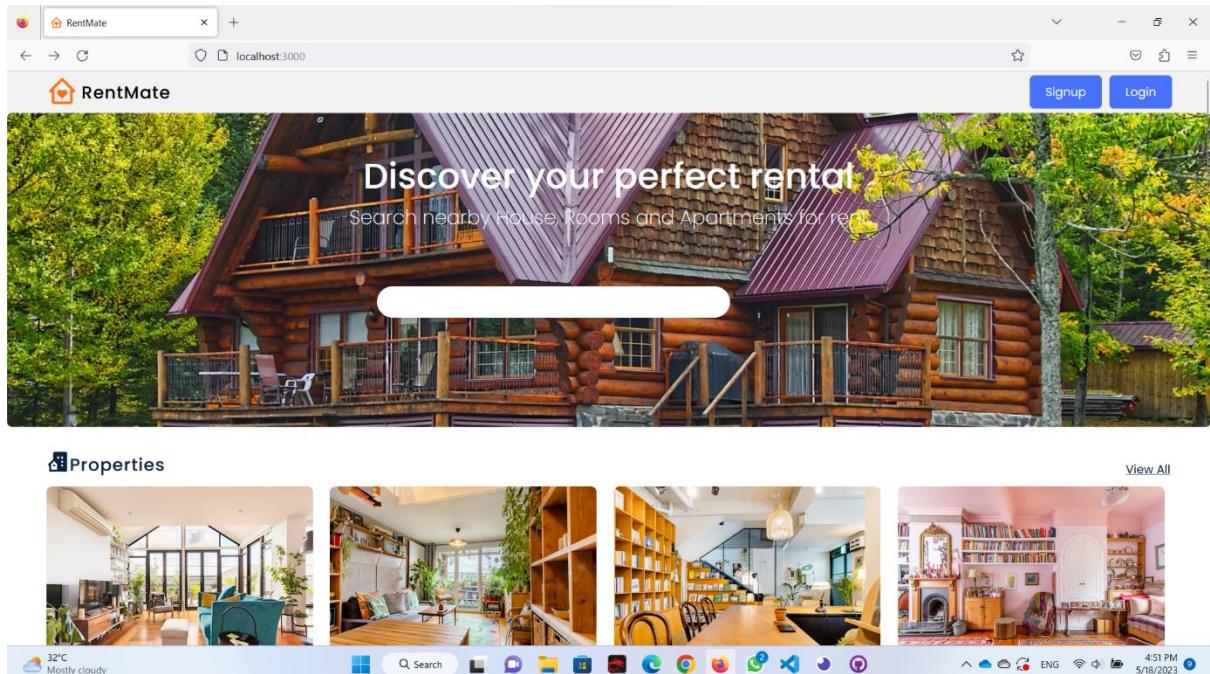


Figure 3. 4 - Home Page

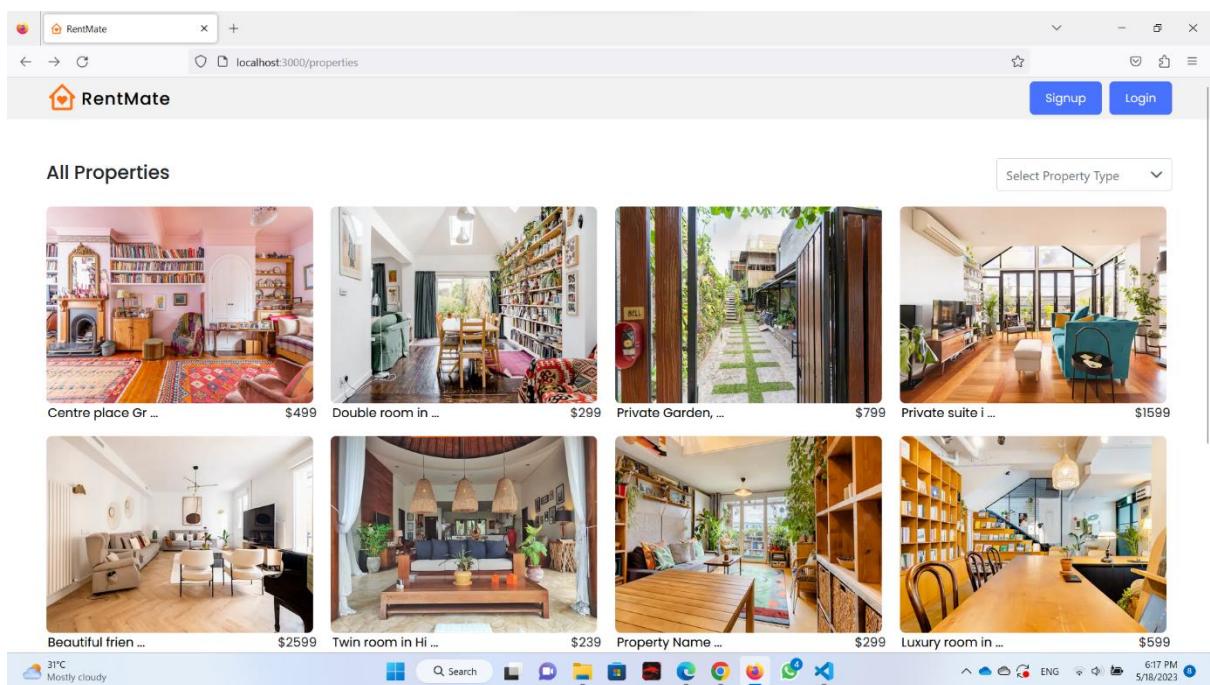


Figure 3. 5 -All properties page

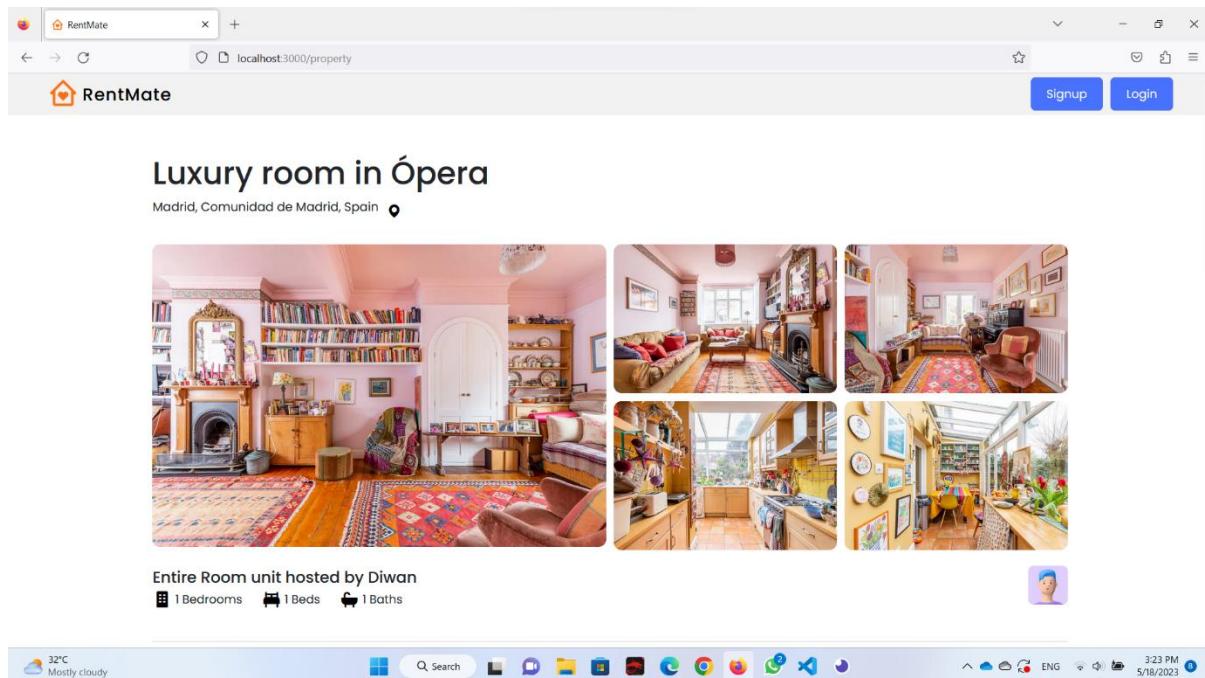


Figure 3. 6 - Single Property Page

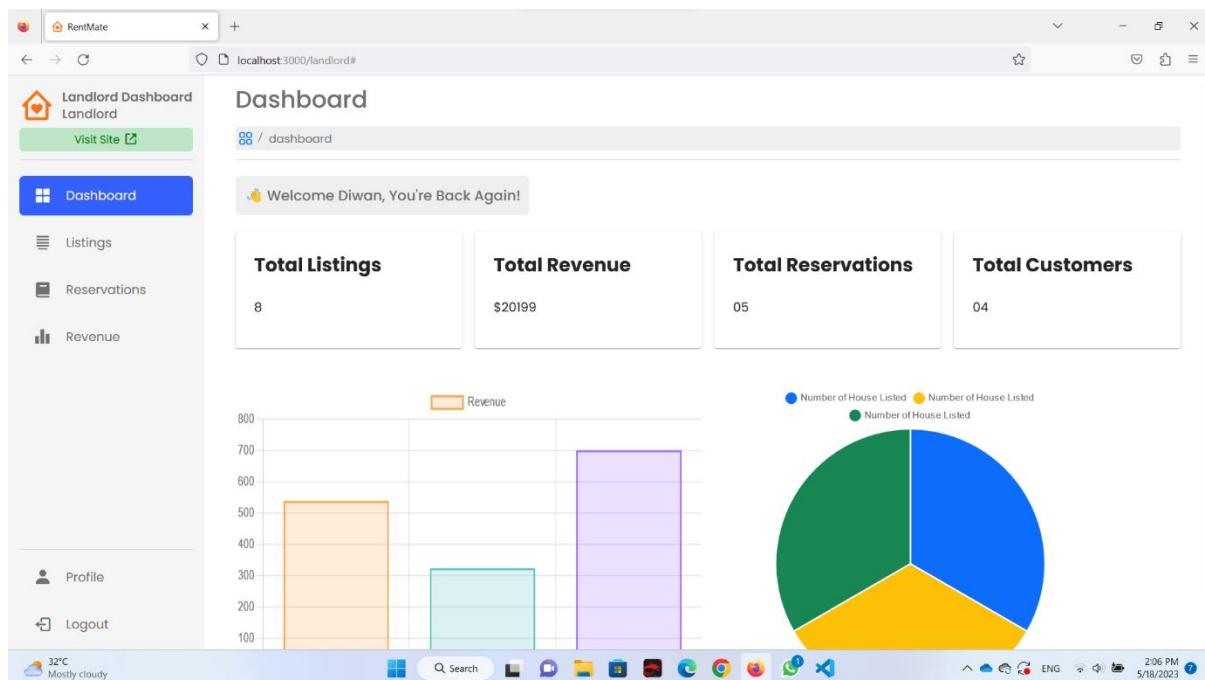


Figure 3. 7 - LandLord Dashboard

Name	Location	Bedrooms	Beds	Baths	Status	Operation
Private Garden, Char ...	Lyon, Auvergne- ...	4	5	2	✓ Active	<button>Edit</button> <button>Delete</button>
Treehouse De Valenti ...	Balamban, Centr ...	1	1	1	✓ Active	<button>Edit</button> <button>Delete</button>
RiverTree Duplex- Ri ...	Kottathara, Ker ...	2	3	2	∅ Pending	<button>Edit</button> <button>Delete</button>
Luxury room in Ópera ...	Madrid, Comunid ...	1	1	1	∅ Pending	<button>Edit</button> <button>Delete</button>
Luxury Surry Hills B ...	Surry Hills, Ne ...	2	2	1	∅ Pending	<button>Edit</button> <button>Delete</button>

Figure 3. 8 - Display All listing page

Add New Listing

Property Name:

Location:

Bedrooms:  Beds:  Baths:

Property Type:  Price:

Description:

MapLink:

Figure 3. 9 - Add New Listing Page

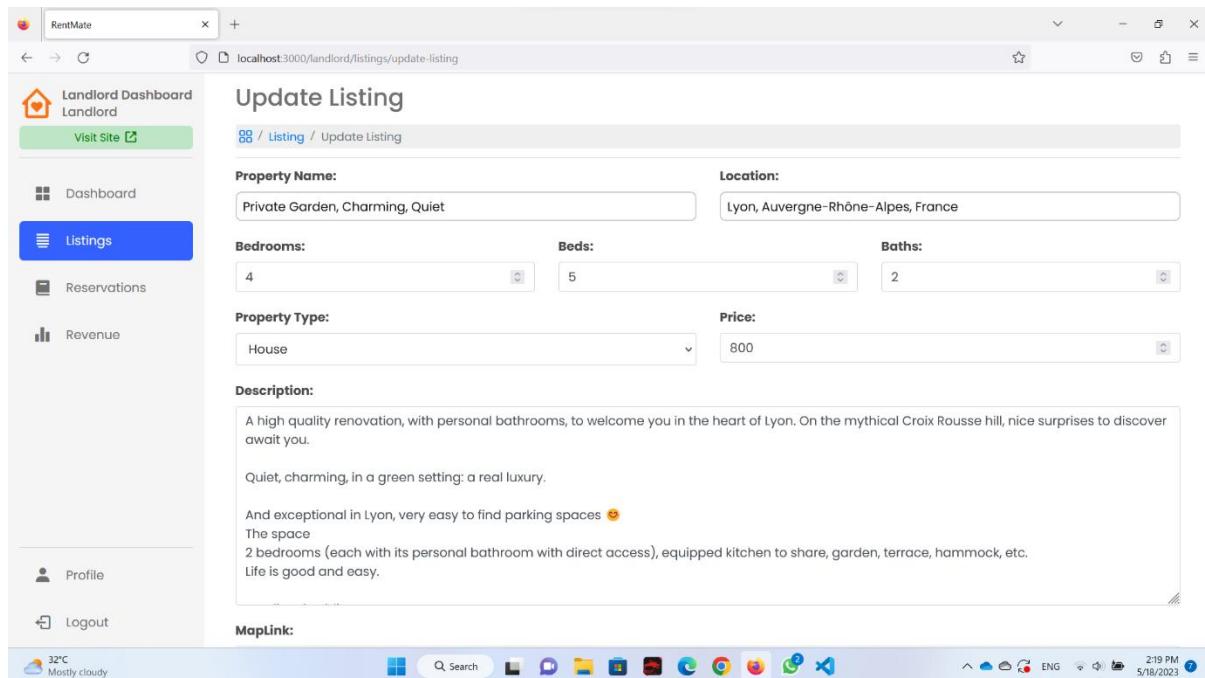


Figure 3. 10 - Update Listing Page

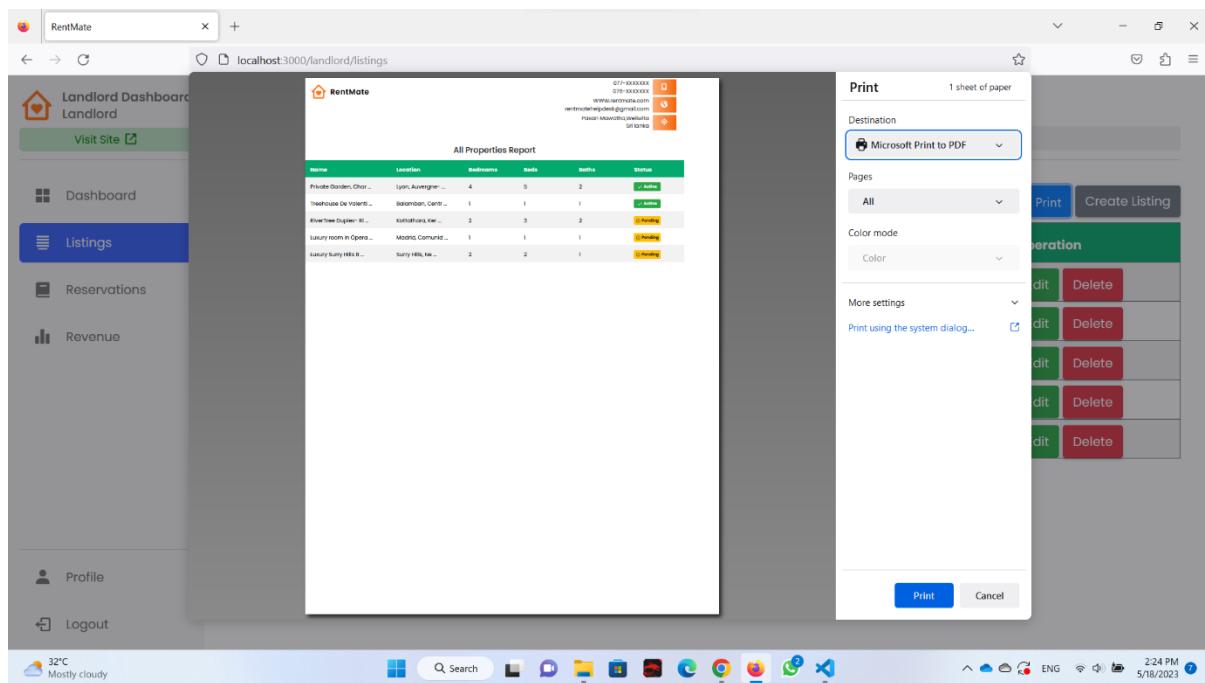


Figure 3. 11 - Listing Report Generate

**Reservations**

Property Name	Cu. Name	Cu. Number	Address	Booked Date	End Date	Duration Period
The Divine Villa Diwan ...	Diwan ...	0766598143	Colombo	08-02-2023	08-02-2024	12 Months
The Divine Villa Sachidu ...	Sachidu ...	0756598143	Gampaha	10-02-2022	08-02-2024	24 Months
The Divine Villa Mallesha ...	Mallesha ...	0716598143	Kaluthara	07-02-2020	08-02-2024	06 Months
The Divine Villa Kasun ...	Kasun ...	0725498143	Galle	05-01-2023	08-02-2024	06 Months
The Divine Villa Chamara ...	Chamara ...	0778598143	Kandy	19-02-2019	08-02-2024	12 Months
The Divine Villa Supun ...	Supun ...	0777598143	NuwaraEliya	18-02-2019	08-02-2024	06 Months
The Divine Villa Shehan ...	Shehan ...	0718998143	Mathara	12-04-2023	08-02-2024	36 Months

Figure 3. 12 - Landlord Reservation Page

**Revenue**

Homes	Rooms	Apartments	Total Revenue
\$499	\$499	\$499	\$499

Property Name	Cu. Name	Cu. Number	Address	Property Type	Price	Status
The Divine Villa Diwan ...	Diwan ...	0766598143	Colombo	House	\$5999	✓ Booked
The Divine Villa Sachidu ...	Sachidu ...	0756598143	Gampaha	Room	\$4999	✓ Booked
The Divine Villa Mallesha ...	Mallesha ...	0716598143	Kaluthara	Apartment	\$10999	✓ Booked
The Divine Villa Kasun ...	Kasun ...	0725498143	Galle	House	\$8999	✓ Booked
The Divine Villa Chamara ...	Chamara ...	0778598143	Kandy	Apartment	\$499	✓ Booked
The Divine Villa Supun ...	Supun ...	0777598143	NuwaraEliya	House	\$999	✓ Booked
The Divine Villa Shehan ...	Shehan ...	0718998143	Mathara	Room	\$1499	✓ Booked

Figure 3. 13 - Landlord Revenue Page

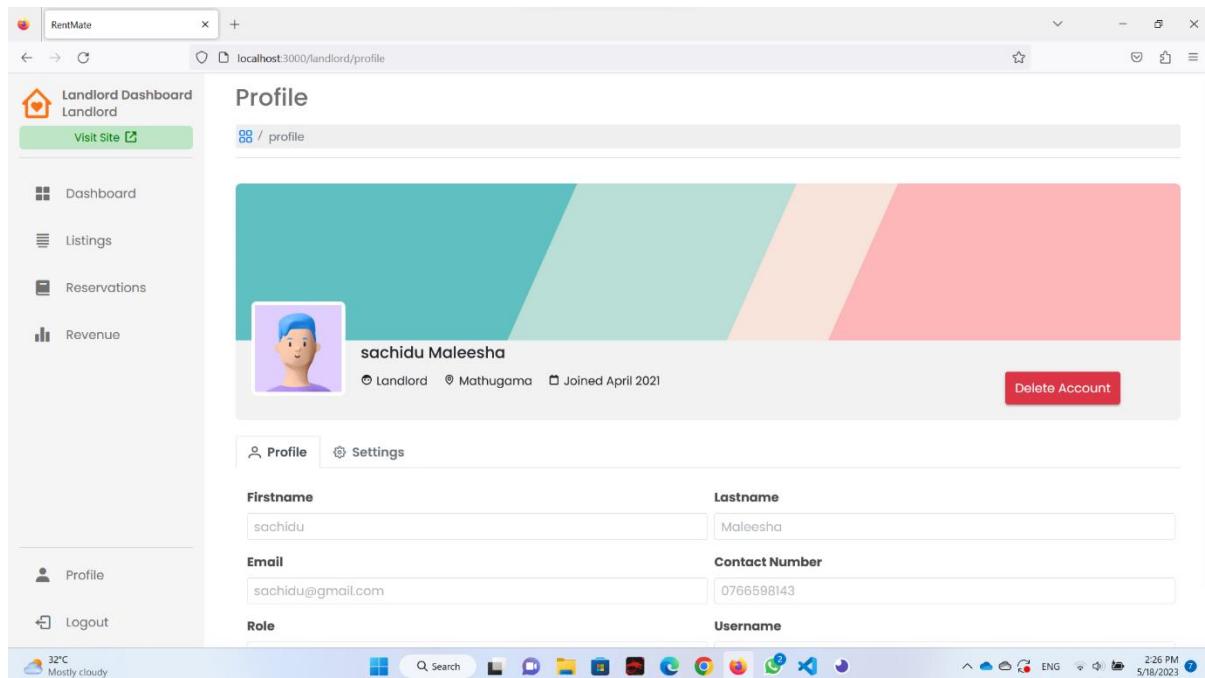


Figure 3. 14 - Landlord Profile Page

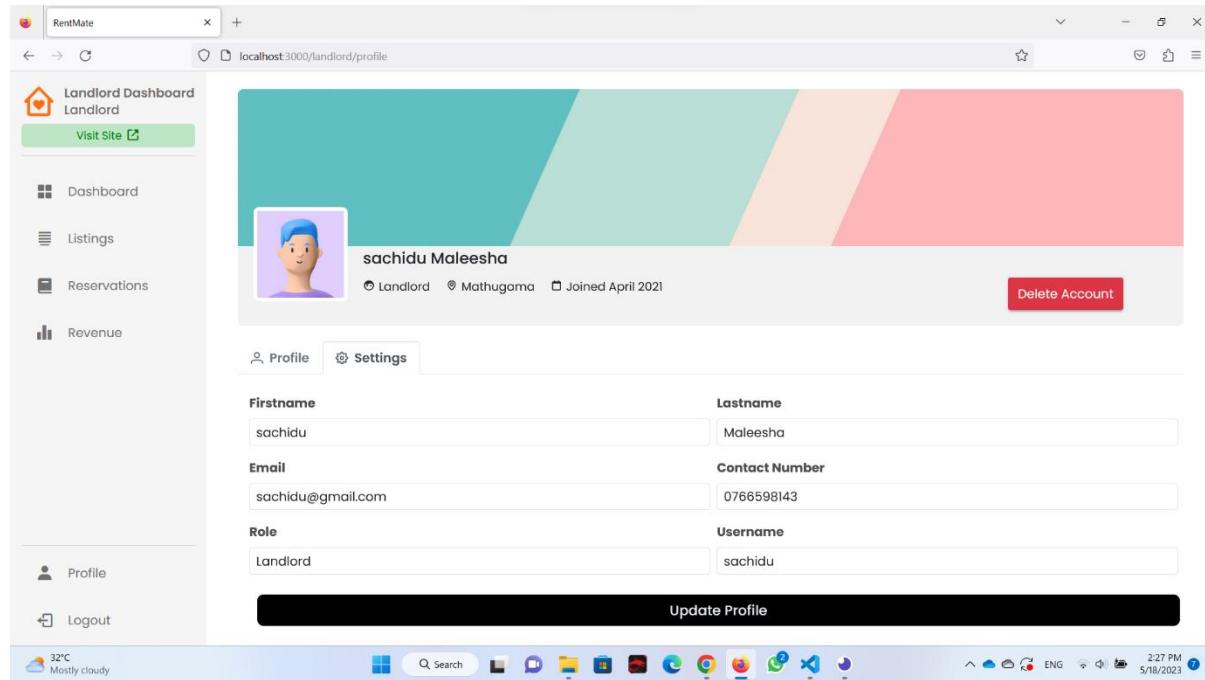


Figure 3. 15 - Landlord Profile Settings

*Payment Management System [IT21183768 – Heshan W.A.S]*

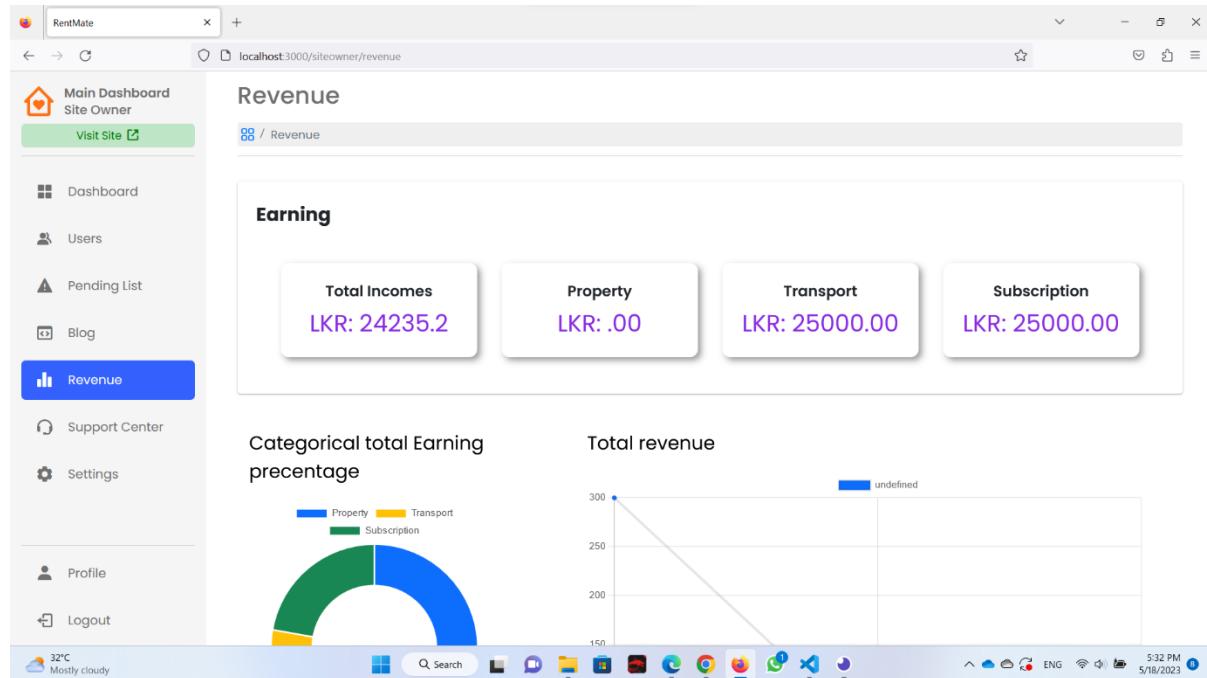


Figure 3. 16 - Payment Dashboard

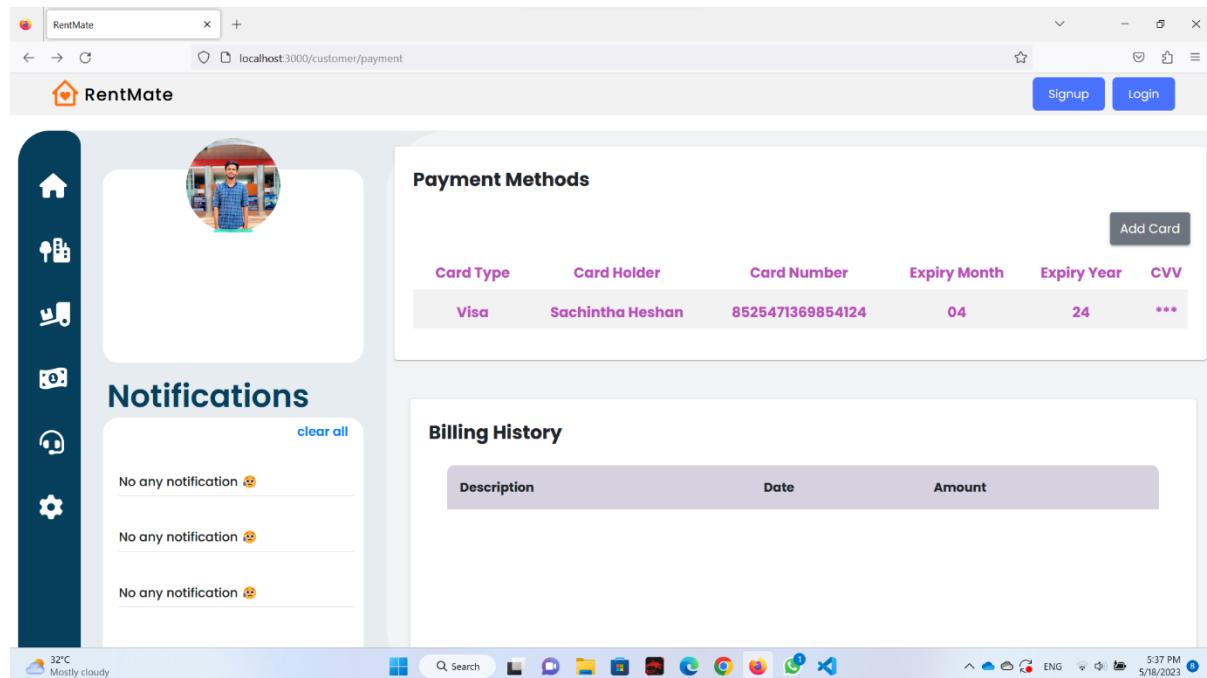


Figure 3. 17 - Payment Card Adding Page

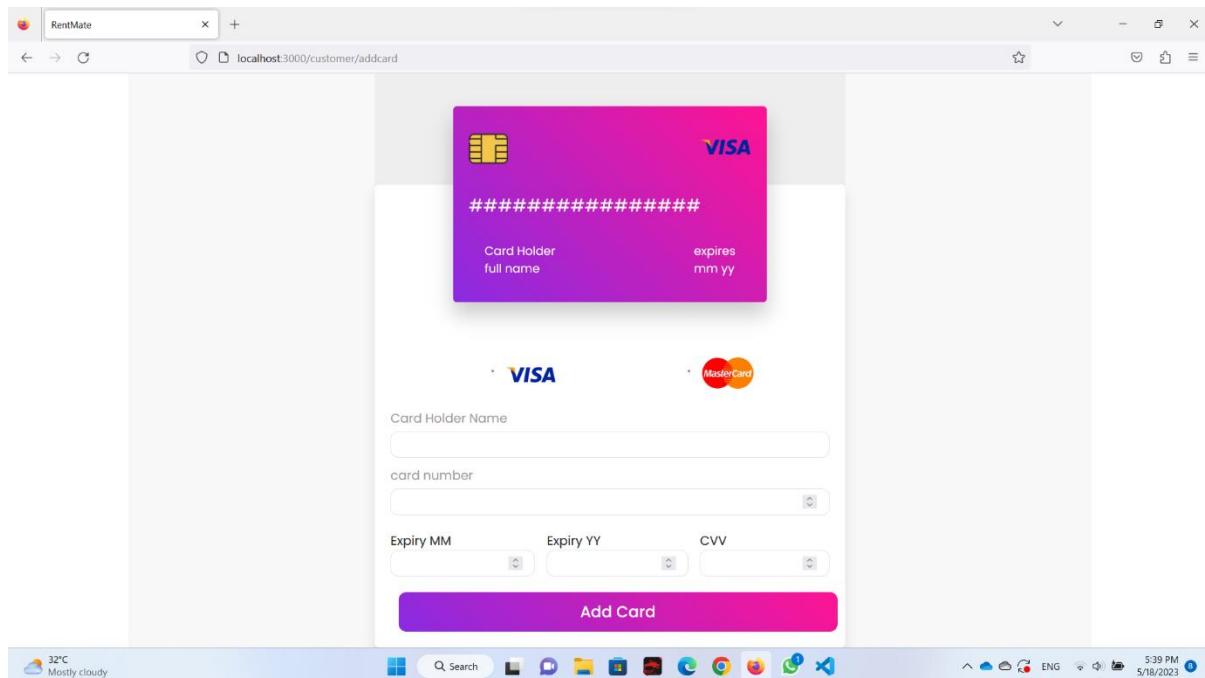


Figure 3. 18 - New Card Adding Page

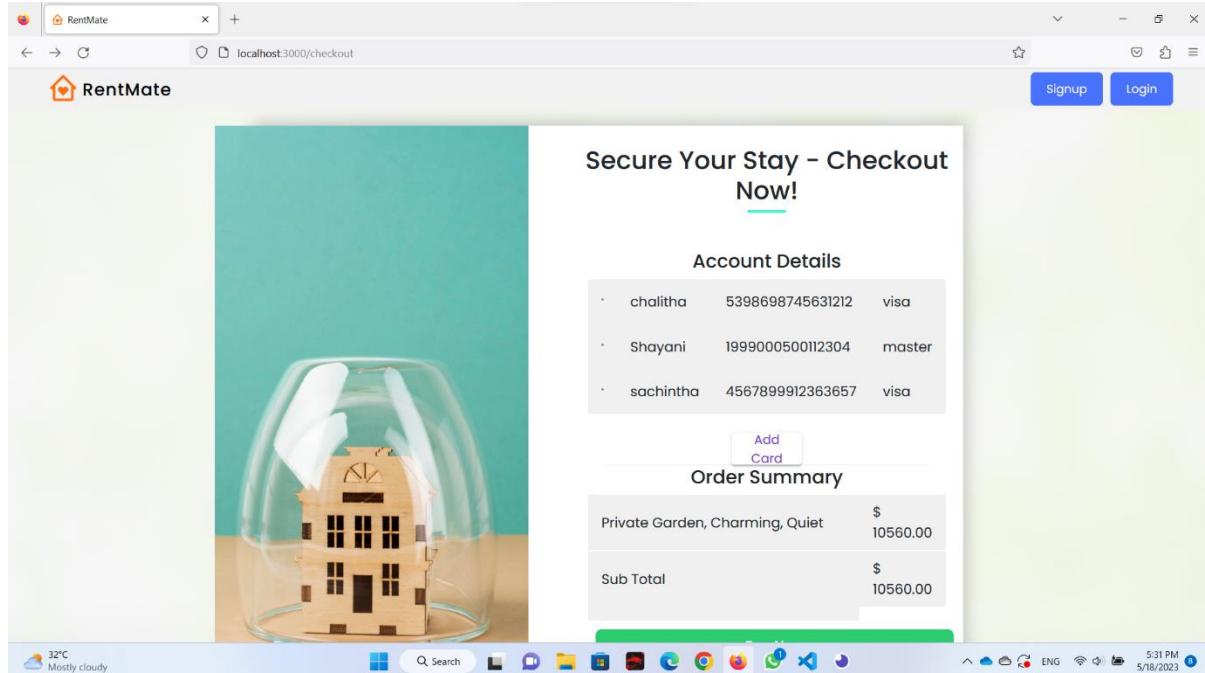


Figure 3. 19 - Checkout Page

*Blog Management System [IT21186424 – Bandara N.G.S.S.M]*

The screenshot shows a web browser window titled "RentMate" with the URL "localhost:3000/siteowner/blog". The left sidebar, titled "Main Dashboard Site Owner", includes links for Dashboard, Users, Pending List, Blog (which is highlighted in blue), Revenue, Support Center, Settings, Profile, and Logout. The main content area is titled "Dashboard" and shows a table of blog posts:

Title	Id	Content	Date	Image	Operation
10 Best Homes in Colombo	45	Best places to visit in srilanka	2023-05-19	Image	<button>Update</button> <button>Delete</button>
Best homes in Malabe	78	Malabe	2023-05-28	Image	<button>Update</button> <button>Delete</button>
Luxuary rooms in Sri Lanka	56	Luxury Rooms	2023-05-13	Image	<button>Update</button> <button>Delete</button>

The status bar at the bottom shows "29°C Mostly cloudy" and the system date/time as "9:42 PM 5/18/2023".

Figure 3. 20 - Blog Management Page

The screenshot shows a web browser window titled "RentMate" with the URL "localhost:3000/siteowner/blog/add-blog". The left sidebar is identical to Figure 3.20. The main content area is titled "Add Blog" and contains the following form fields:

- title**: An input field with placeholder text "Enter Title here".
- id**: An input field with placeholder text "Enter ID".
- Text Here**: A large text area labeled "Text Here" with a placeholder text "Text Here".

The status bar at the bottom shows "29°C Mostly cloudy" and the system date/time as "9:43 PM 5/18/2023".

Figure 3. 21 - New Blog Adding Page

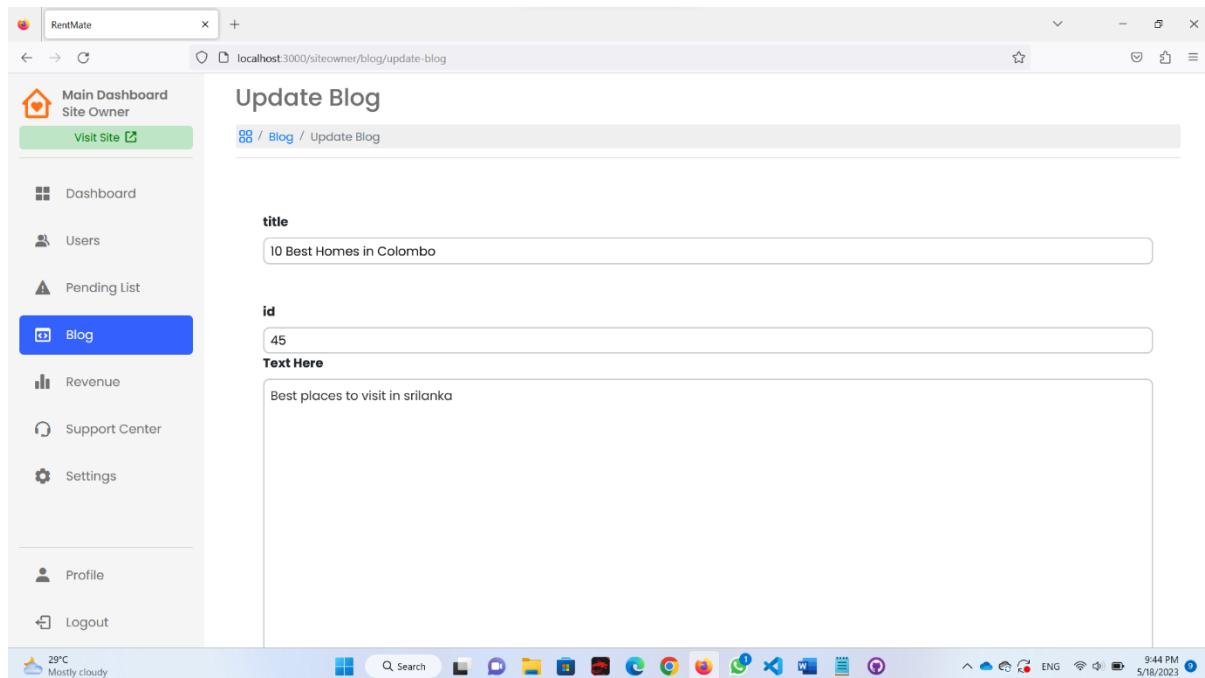


Figure 3. 22 - Blog Update Page

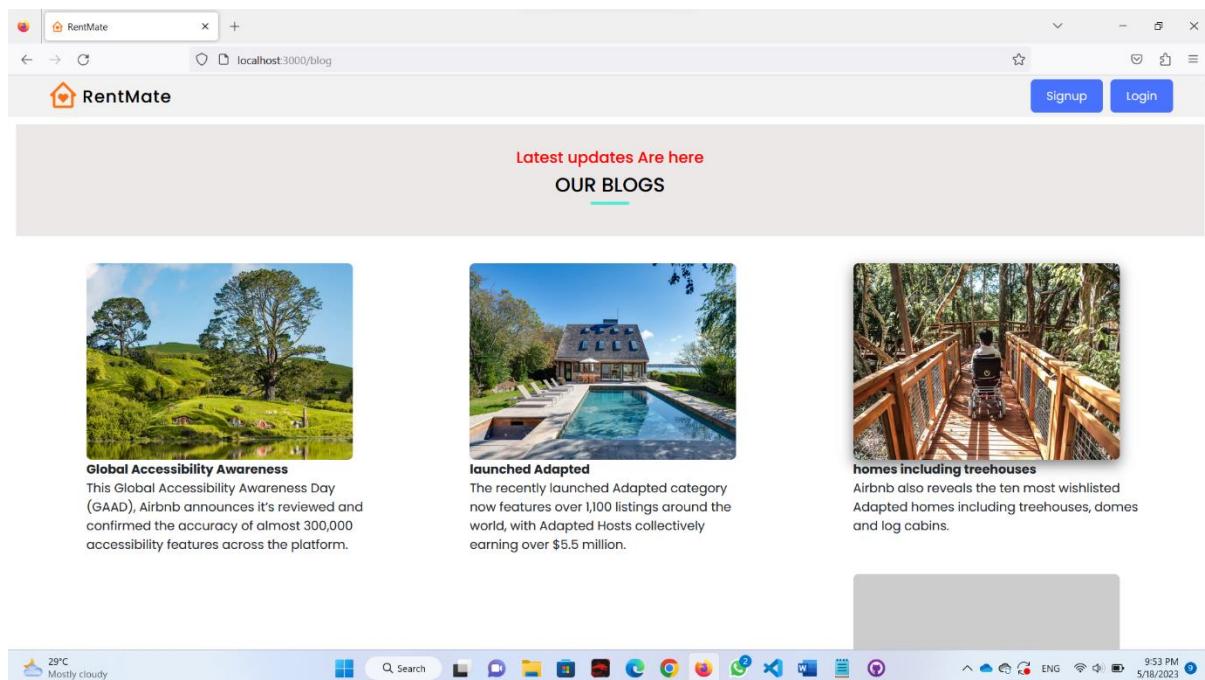


Figure 3. 23 - All Blog View Page

*Customer Profile [IT21233562 – Ranawaka W.E.I]*

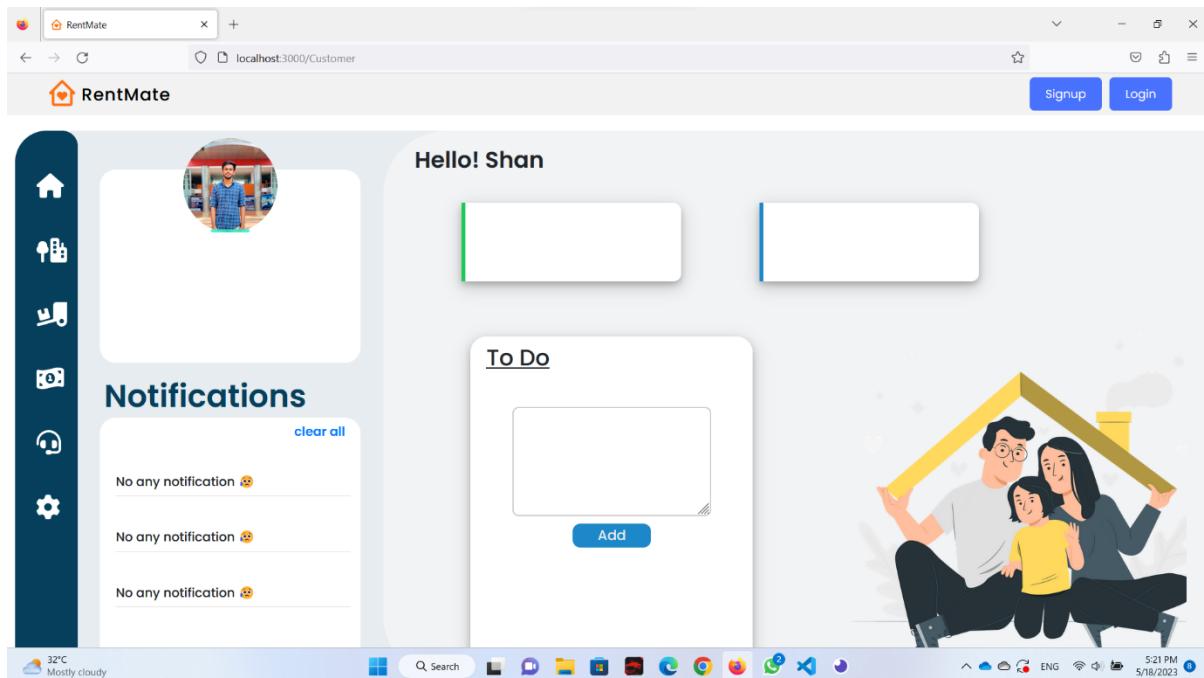


Figure 3. 24 - Customer Profile

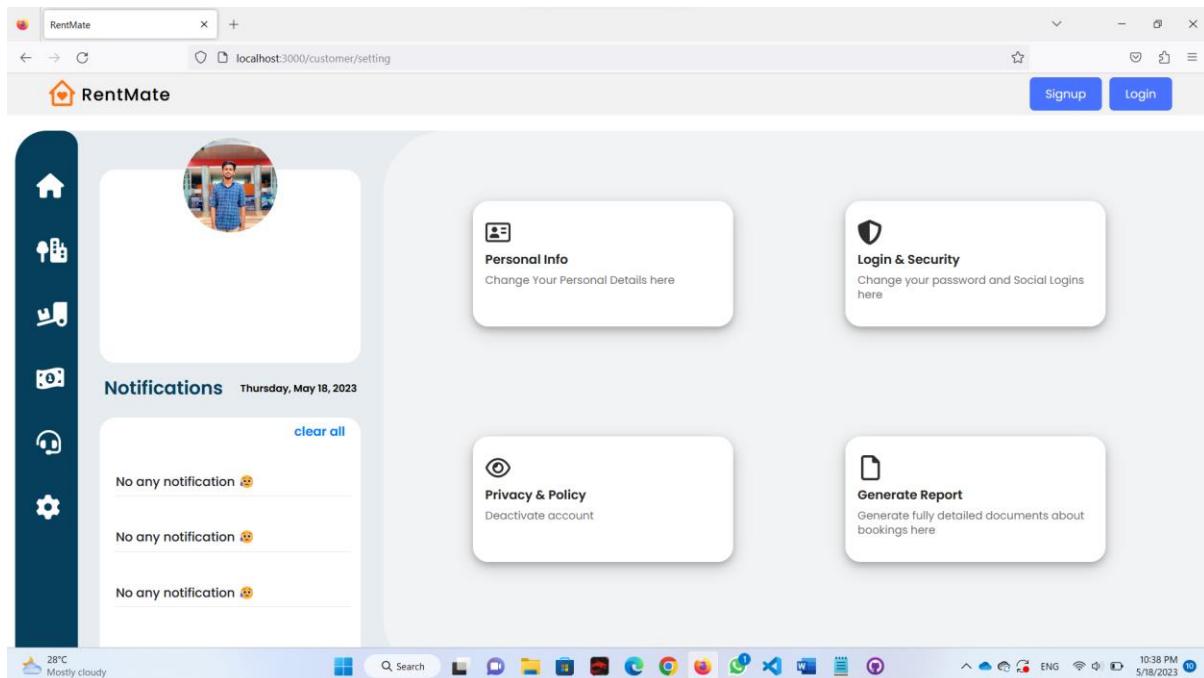


Figure 3. 25 - Customer Setting Page

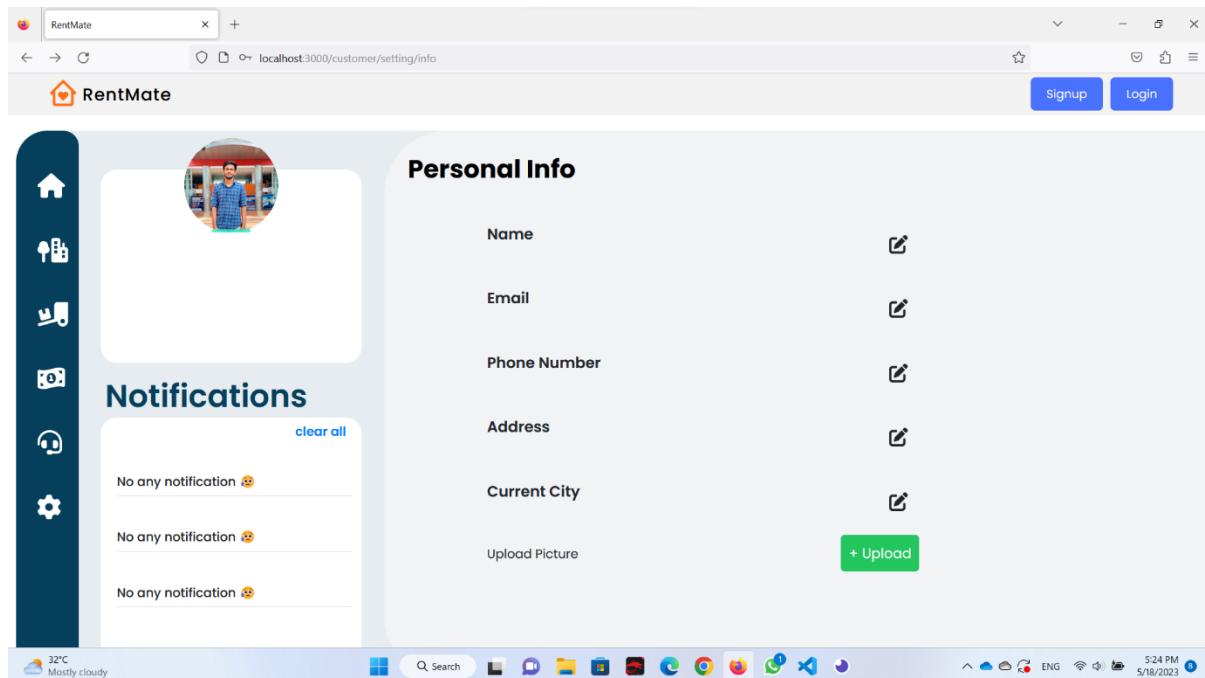


Figure 3. 26 - Customer Profile Info Page

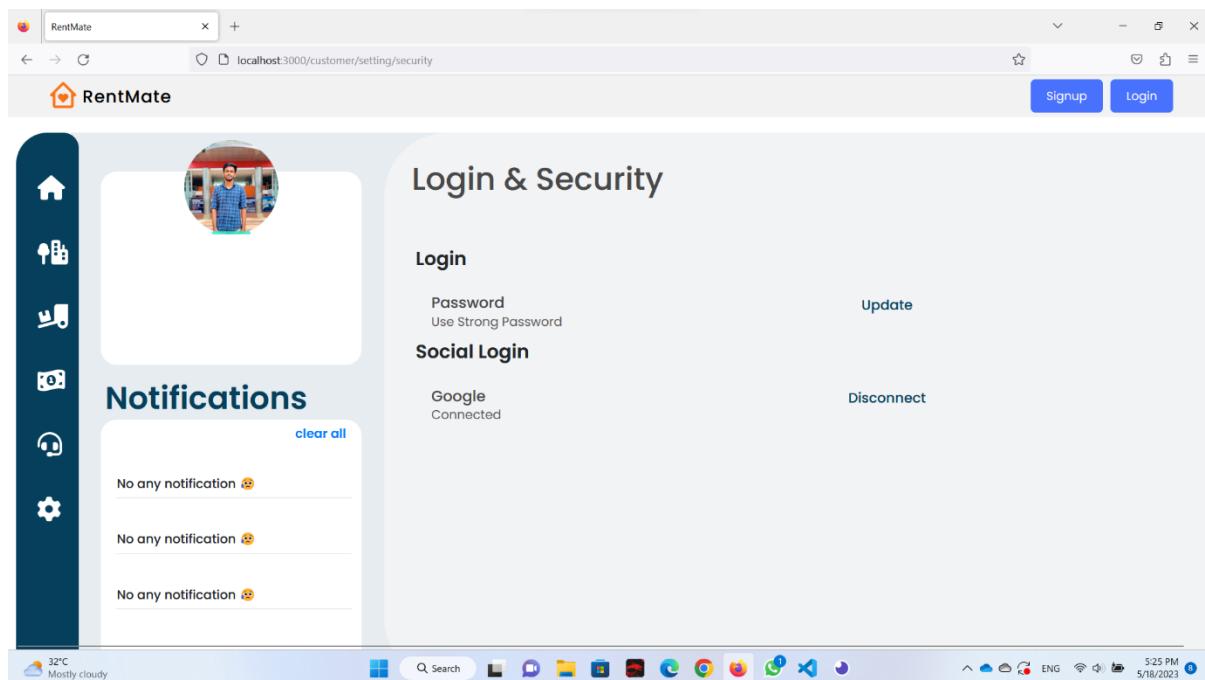


Figure 3. 27 - Customer Login and Security Page

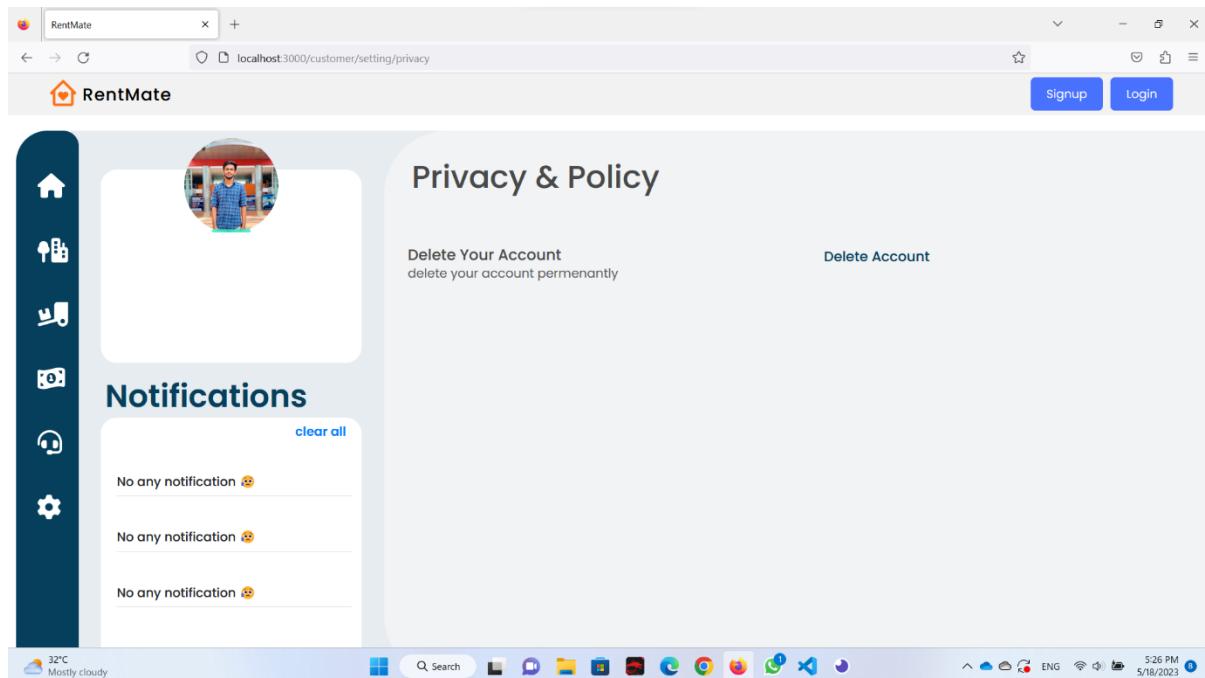


Figure 3. 28 - Customer Privacy and Policy Page

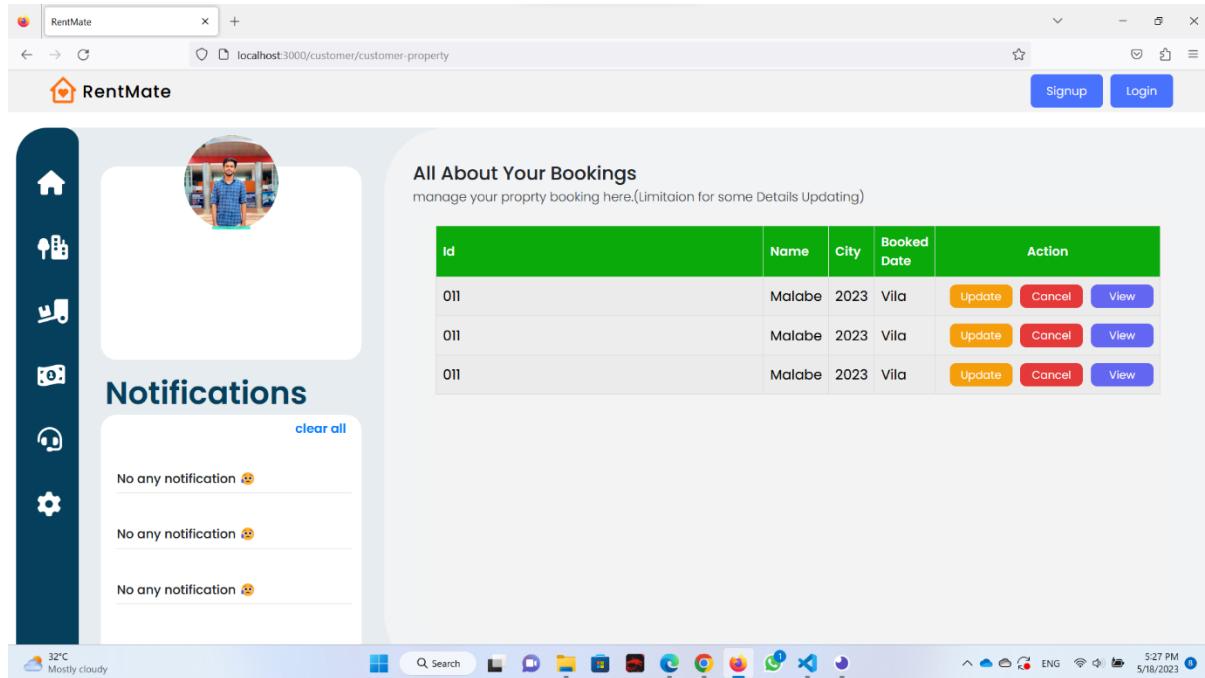


Figure 3. 29 – Customer view Booking page

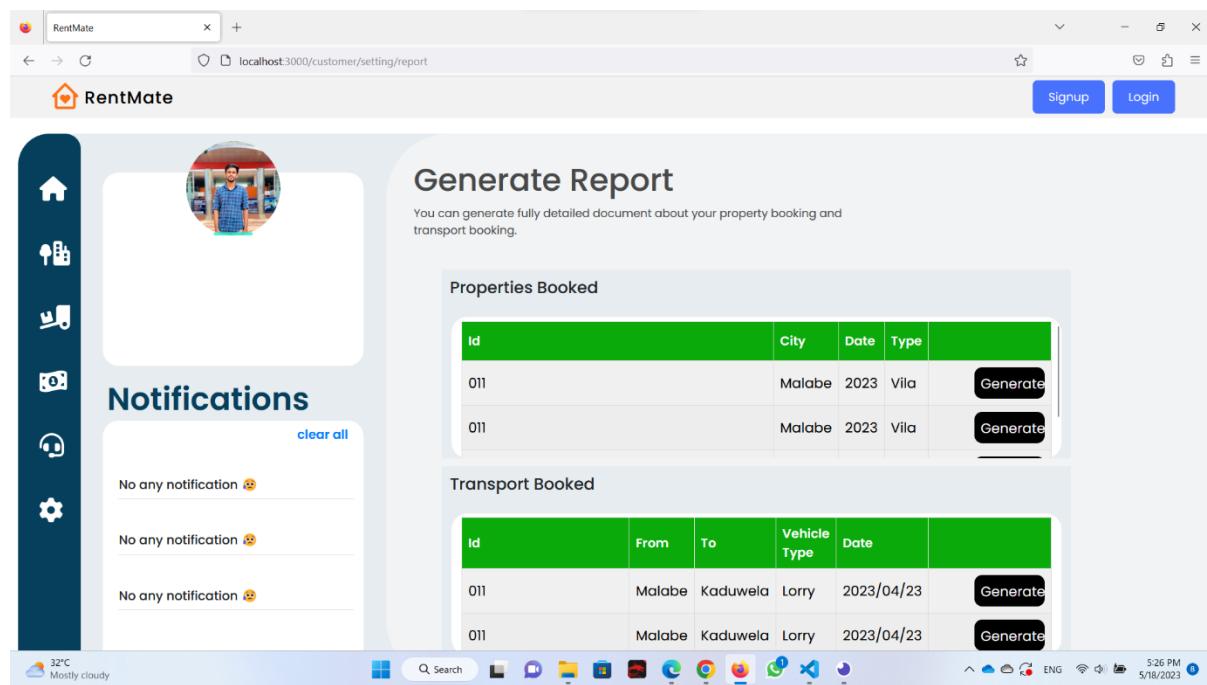


Figure 3. 30 - Customer Generate Report Page

*Showroom Management System [IT21184376 – Jayathilaka A.W.N.W]*

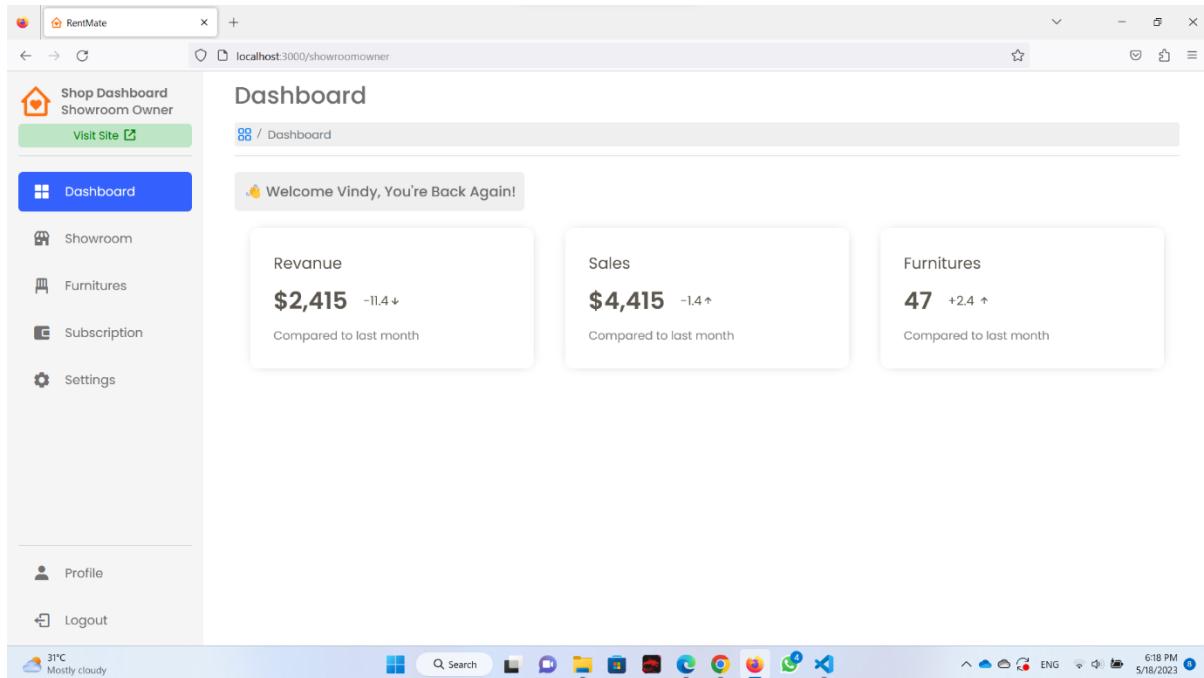


Figure 3. 31 - Showroom Dashboard Page

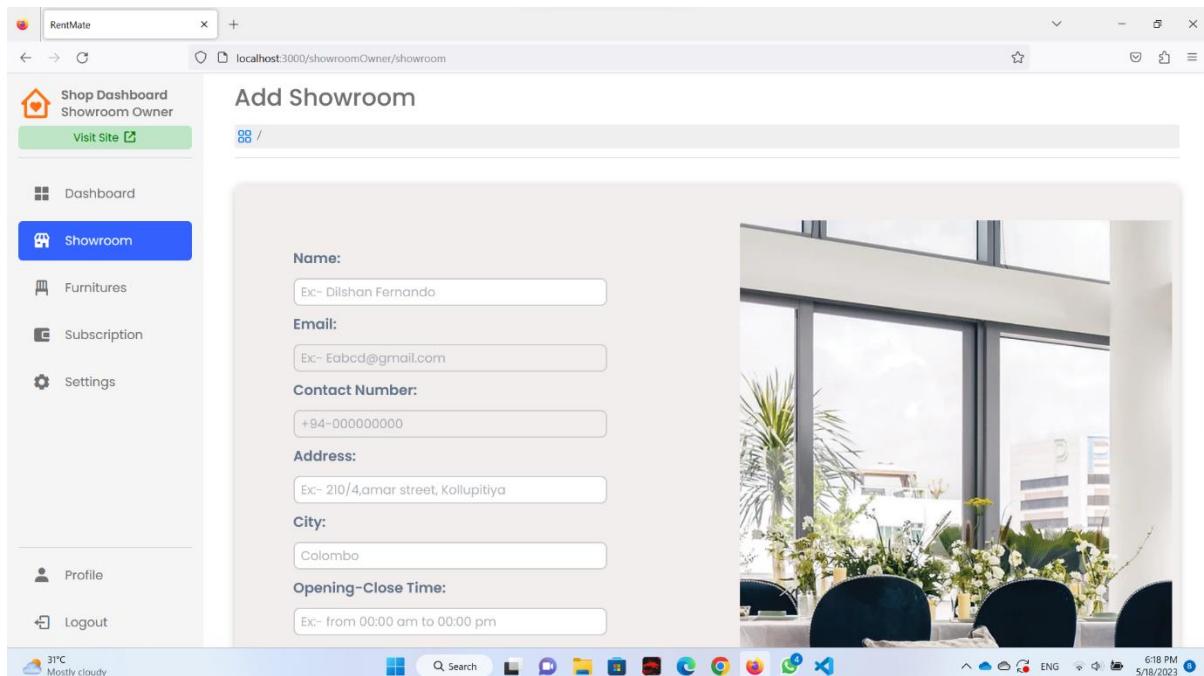


Figure 3. 32 - Add Showroom Page

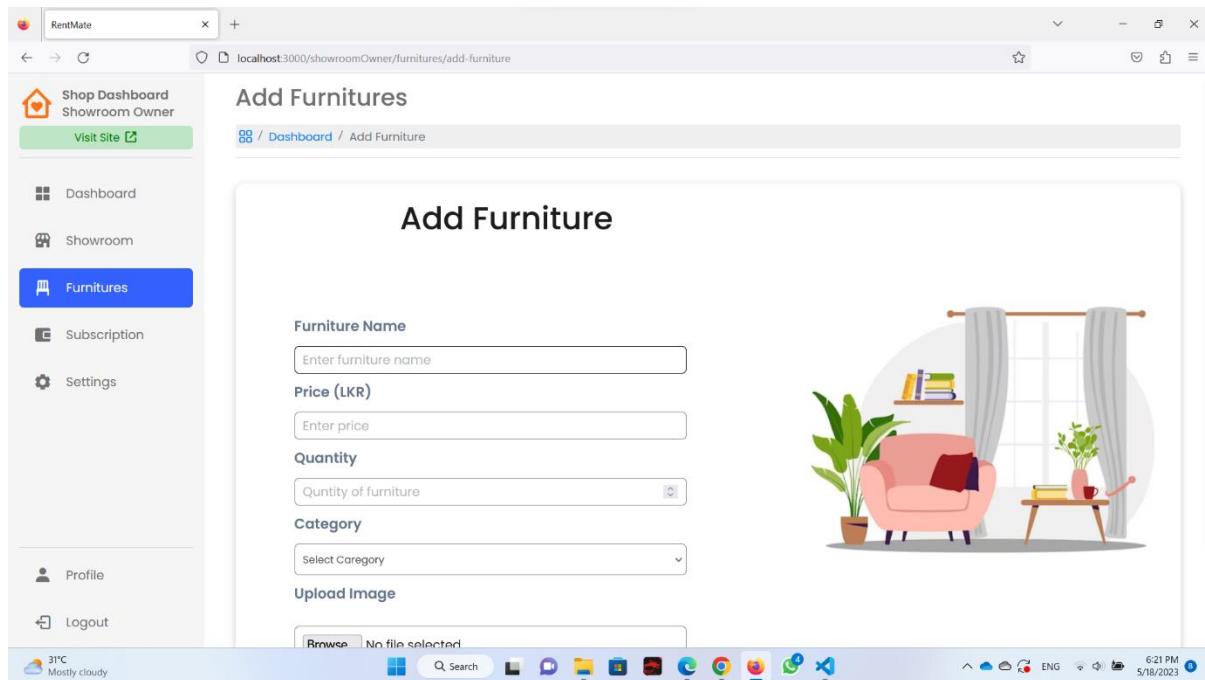


Figure 3. 33 - Add New Furniture Page

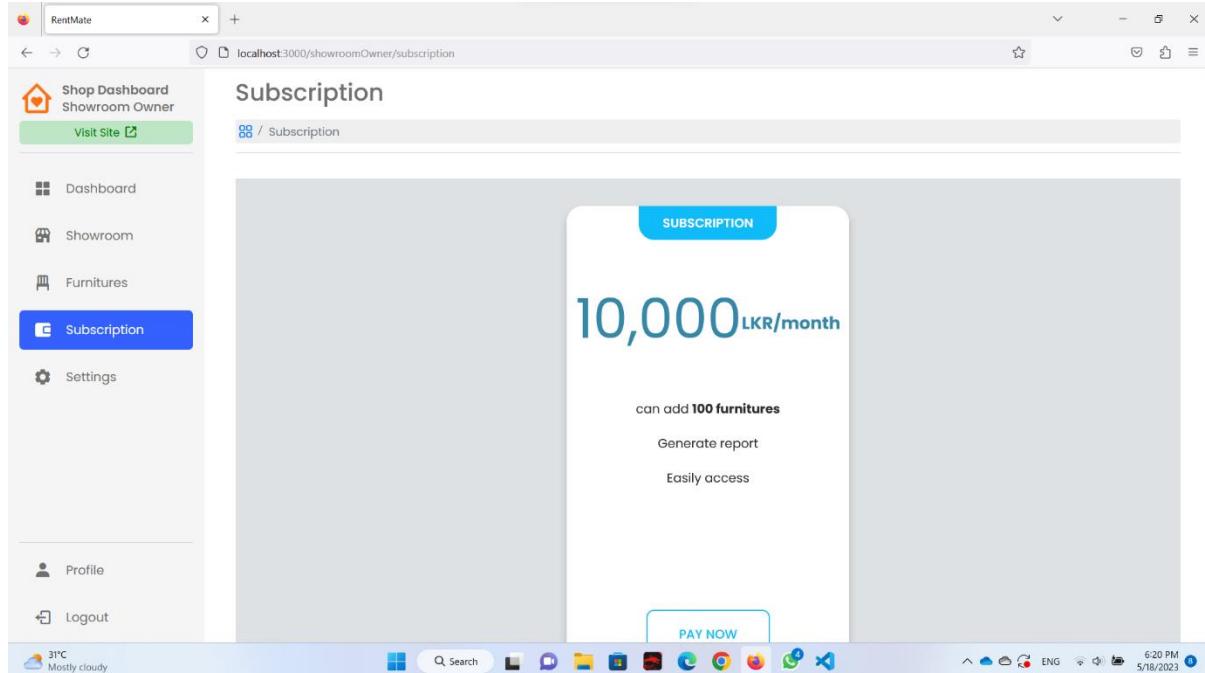


Figure 3. 34 - Showroom Subscription Page

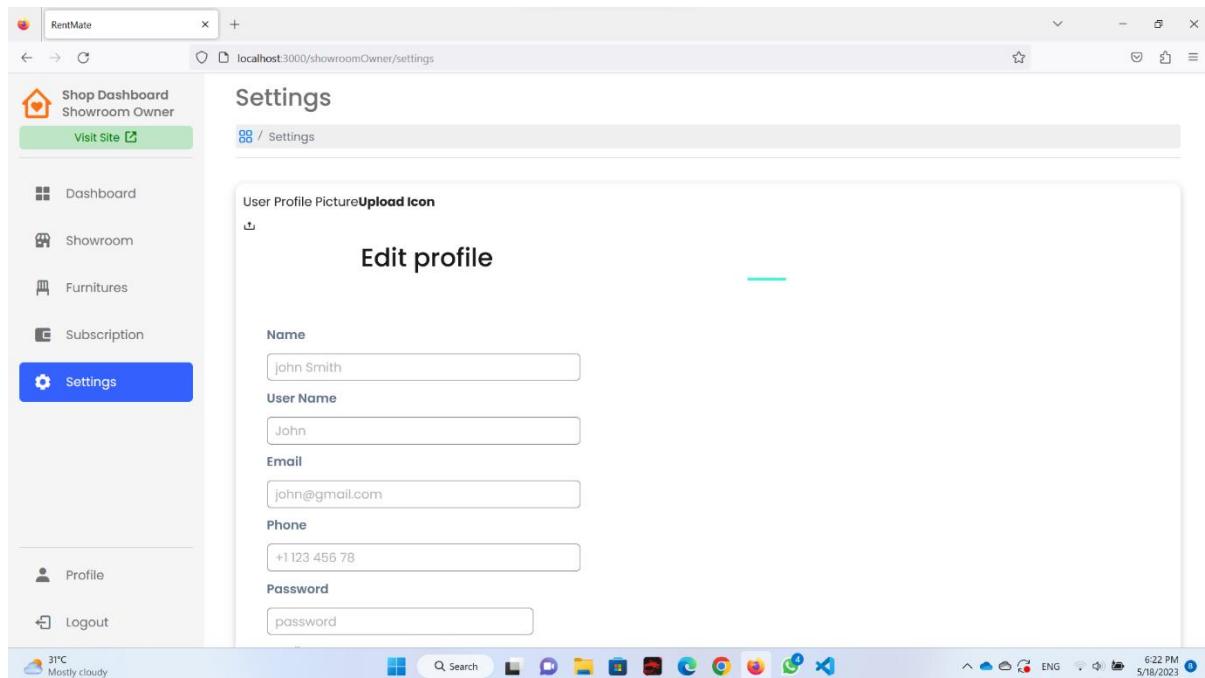


Figure 3. 35 - Showroom Profile Setting Page

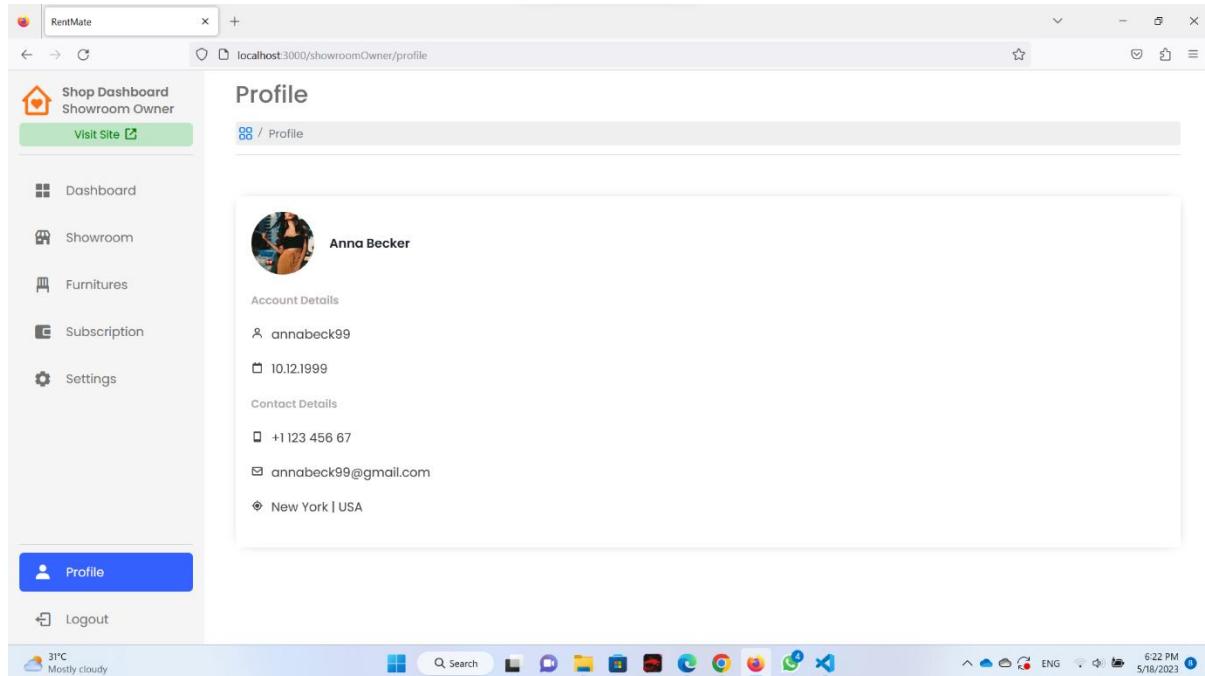


Figure 3. 36 - Showroom Profile Page

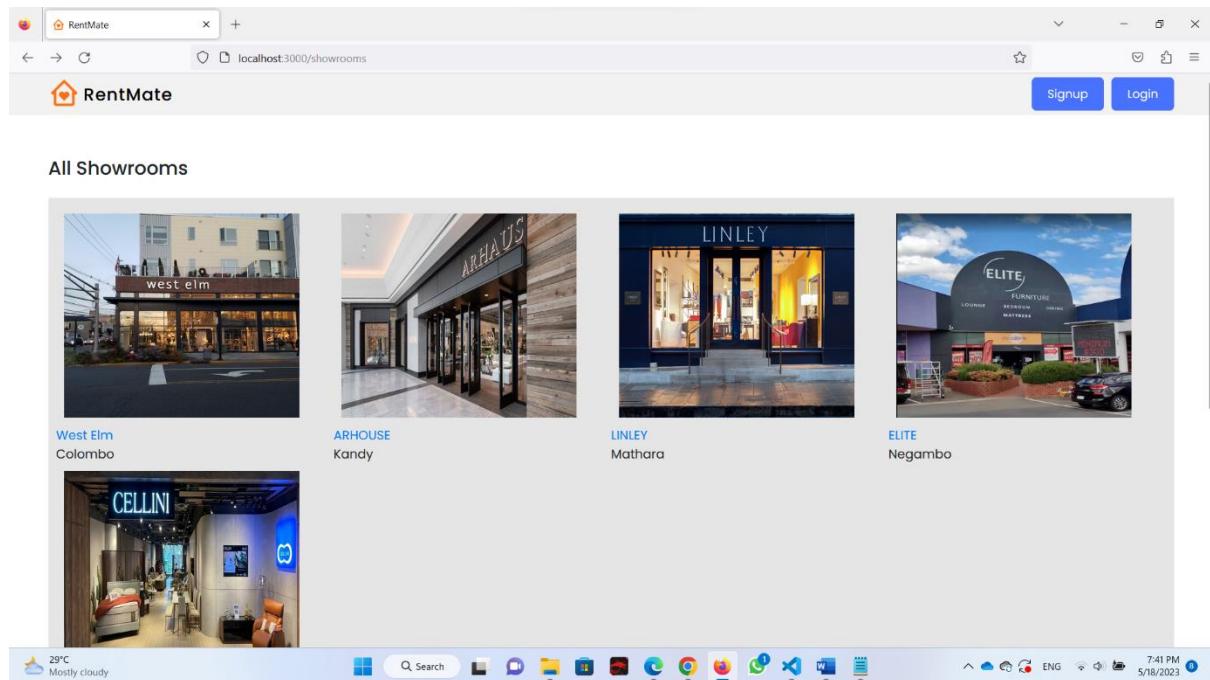


Figure 3. 37 - All Showroom Page

*Vehicle Management System [IT21182846 – Chamaleen D.B.N]*

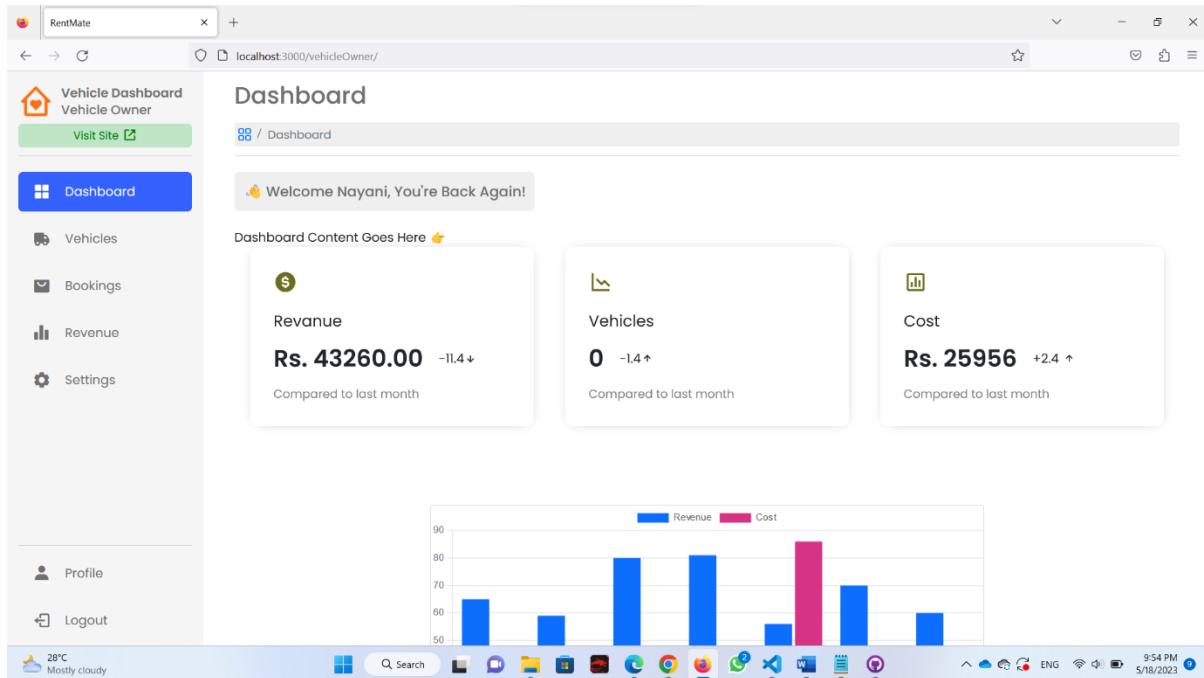


Figure 3. 38 - Vehicle Dashboard Page

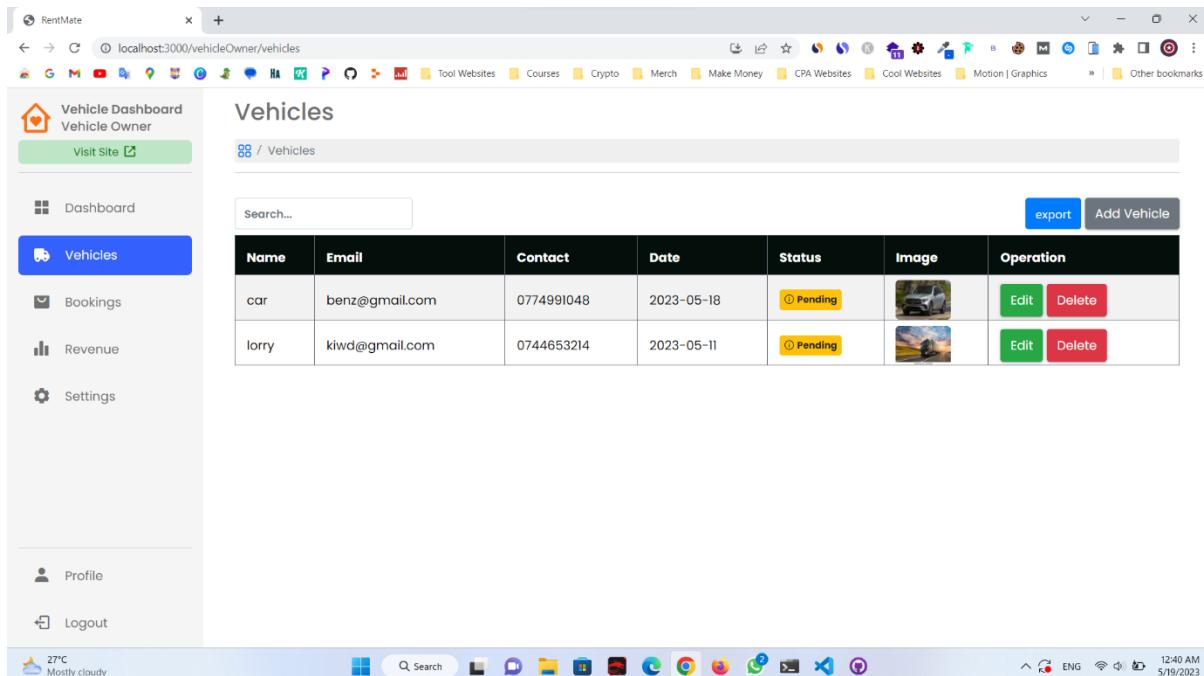


Figure 3. 39 - All vehicle listing page

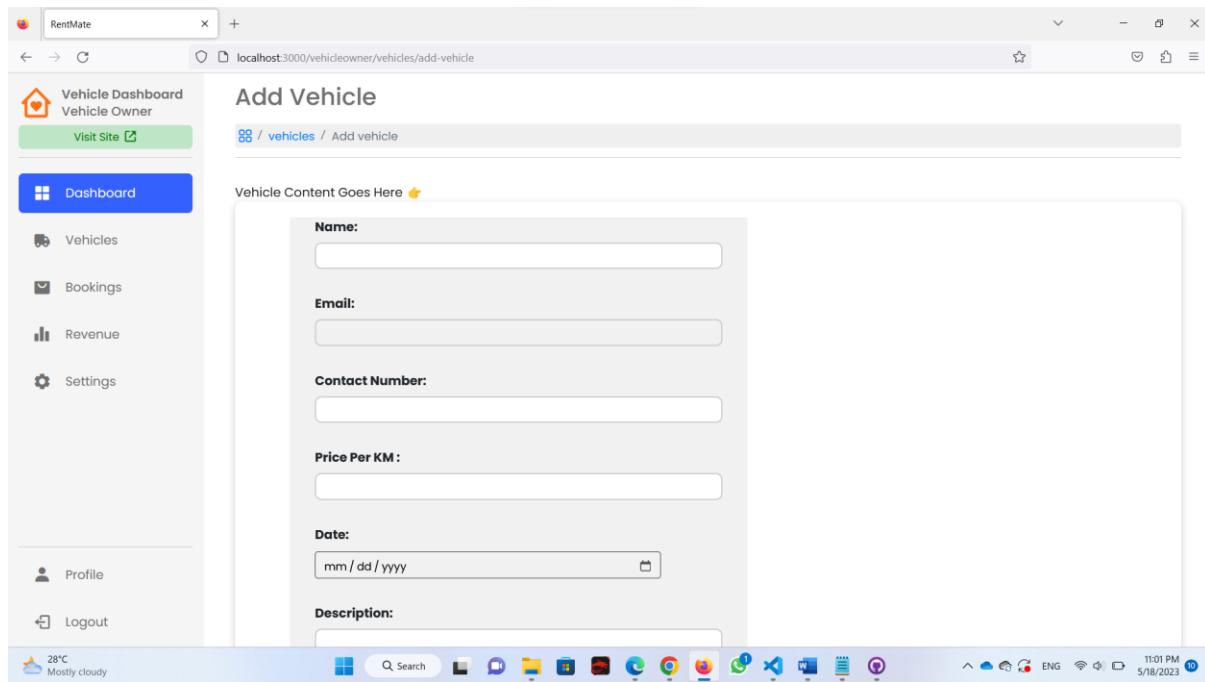


Figure 3. 40 - Add Vehicle Page

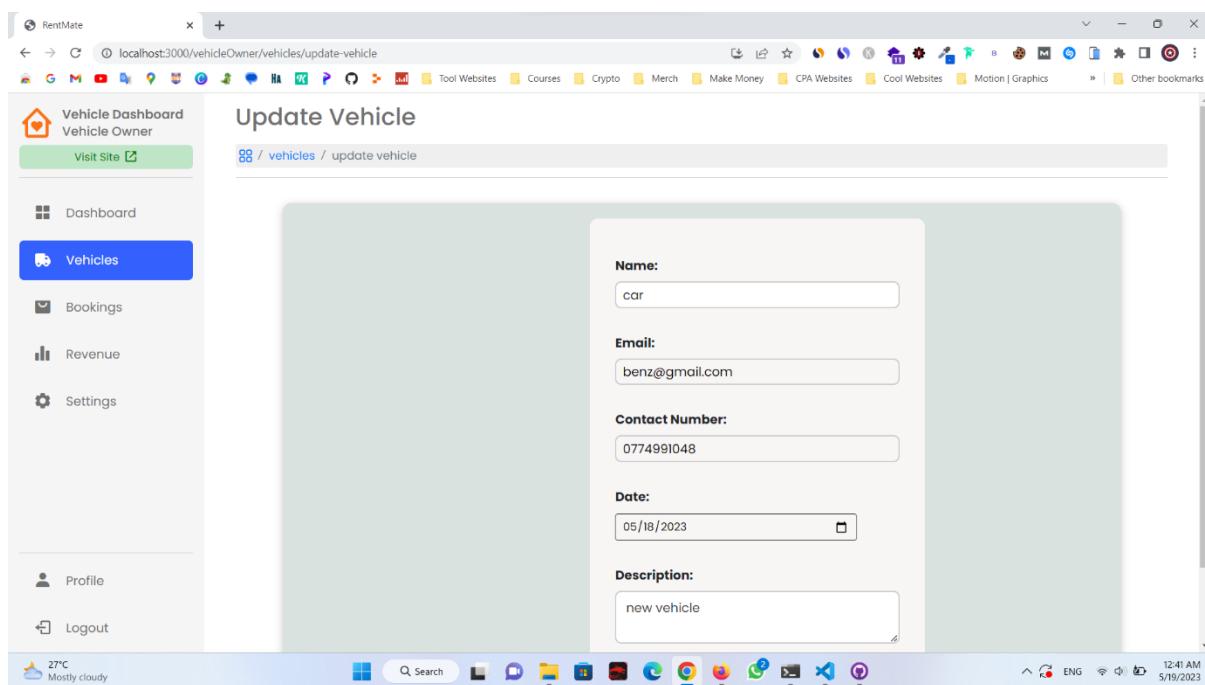


Figure 3. 41 - Update Vehicle Page

The screenshot shows a web browser window titled "RentMate" with the URL "localhost:3000/vehicleOwner/bookings". The left sidebar has a "Bookings" button highlighted in blue. The main content area is titled "Bookings" and displays a table of vehicle bookings:

Delivery Location	Delivery Date	Return Location	Distance	Price
31231	2023-05-18	123	123	Rs.2460
123	2023-05-21	555	34	Rs.40800
2	2023-05-23	67	11	Rs.374000

The browser toolbar at the top includes various icons for file operations, search, and bookmarks. The taskbar at the bottom shows the Windows Start button, a search bar, and several pinned application icons.

Figure 3. 42 - View Vehicle Bookings Page

The screenshot shows a web browser window titled "RentMate" with the URL "localhost:3000/vehicleOwner/profile". The left sidebar has a "Profile" button highlighted in blue. The main content area is titled "Profile" and displays the user's profile information:

**sachidu Maleesha**  
Landlord

**Account Details**

- sachidu**
- 10.12.1999**

**Contact Details**

- 0766598143**
- sachidu@gmail.com**
- New York | USA**

The browser toolbar and taskbar are visible at the top and bottom of the screen respectively.

Figure 3. 43 - Vehicle Profile Page

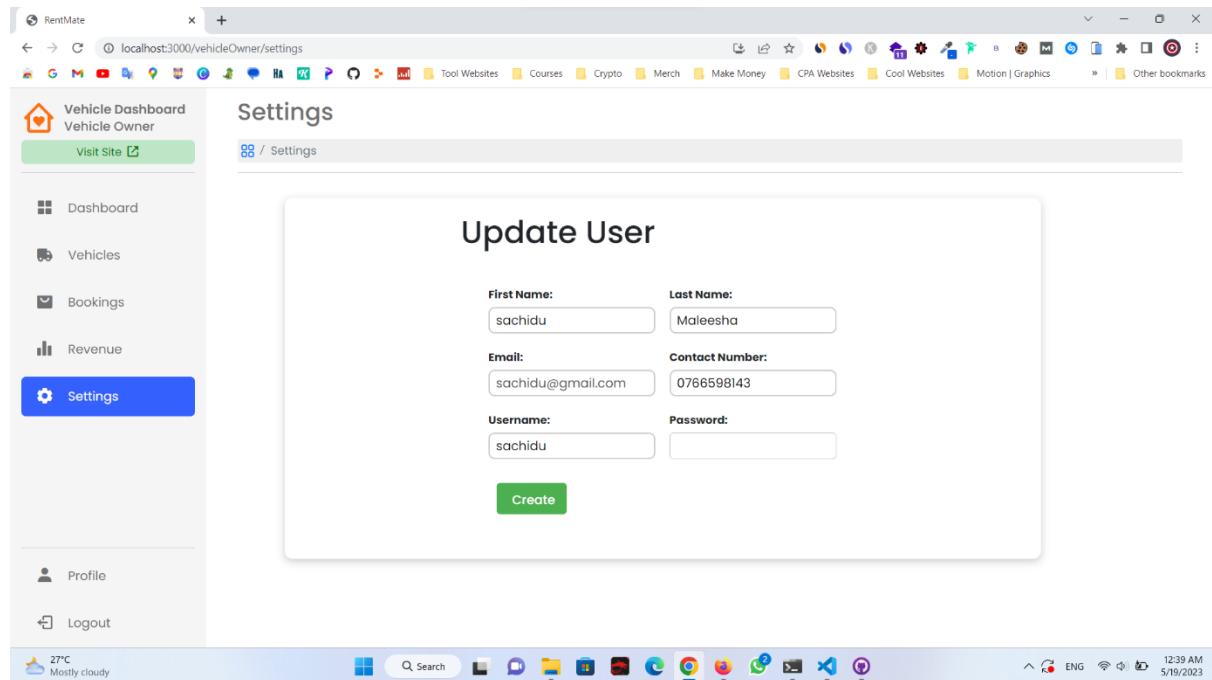


Figure 3. 44 - Update Vehicle Owner Profile Page

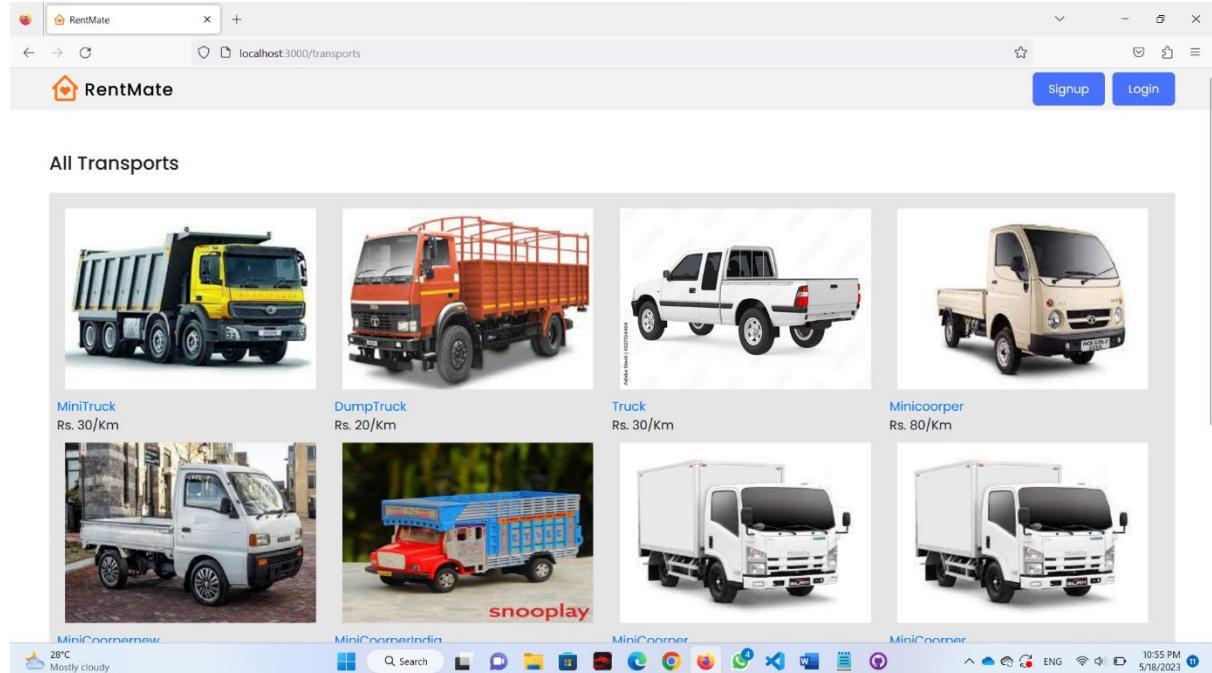


Figure 3. 45 - All transport view page

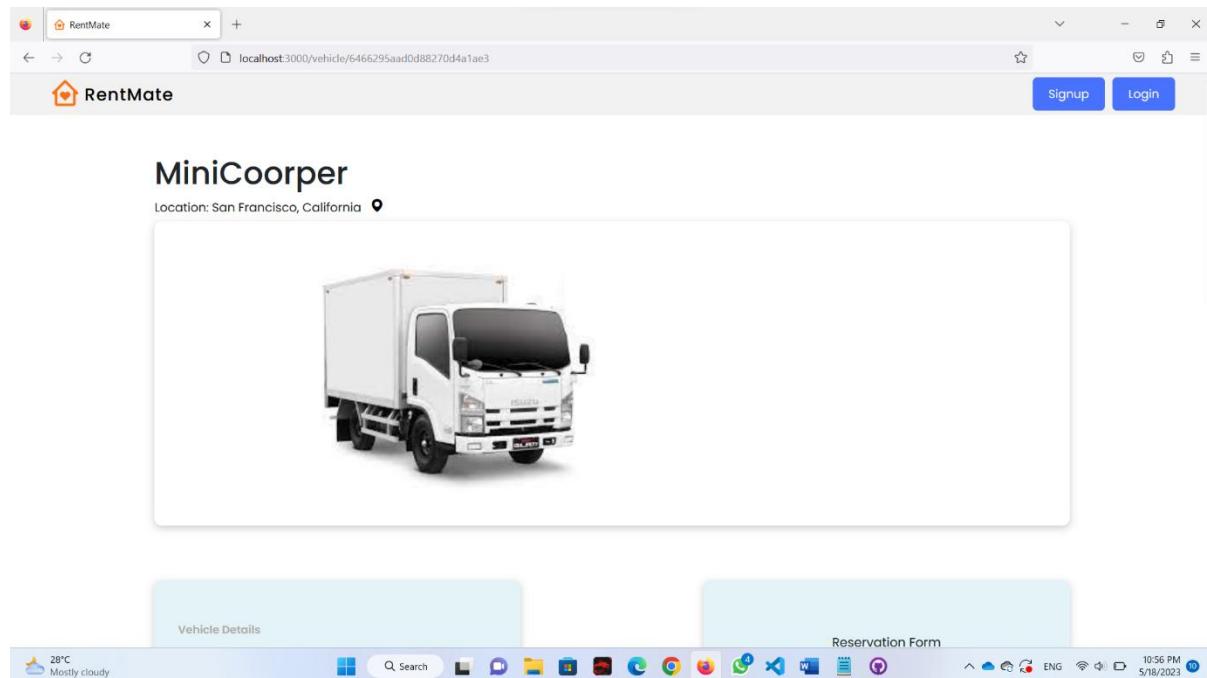


Figure 3. 46 - Single Vehicle Details Page

*User Management and Pending Listing Management System [IT21197796 – Wasana K.H.I.M]*

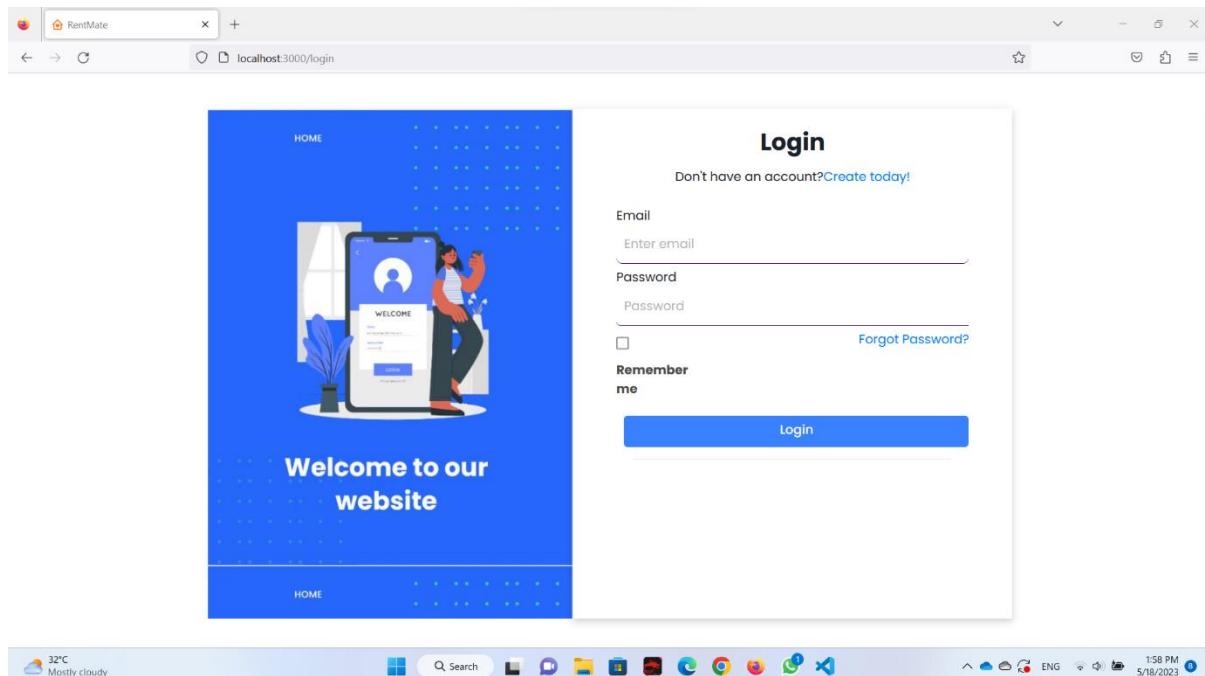


Figure 3. 47 - Login Page

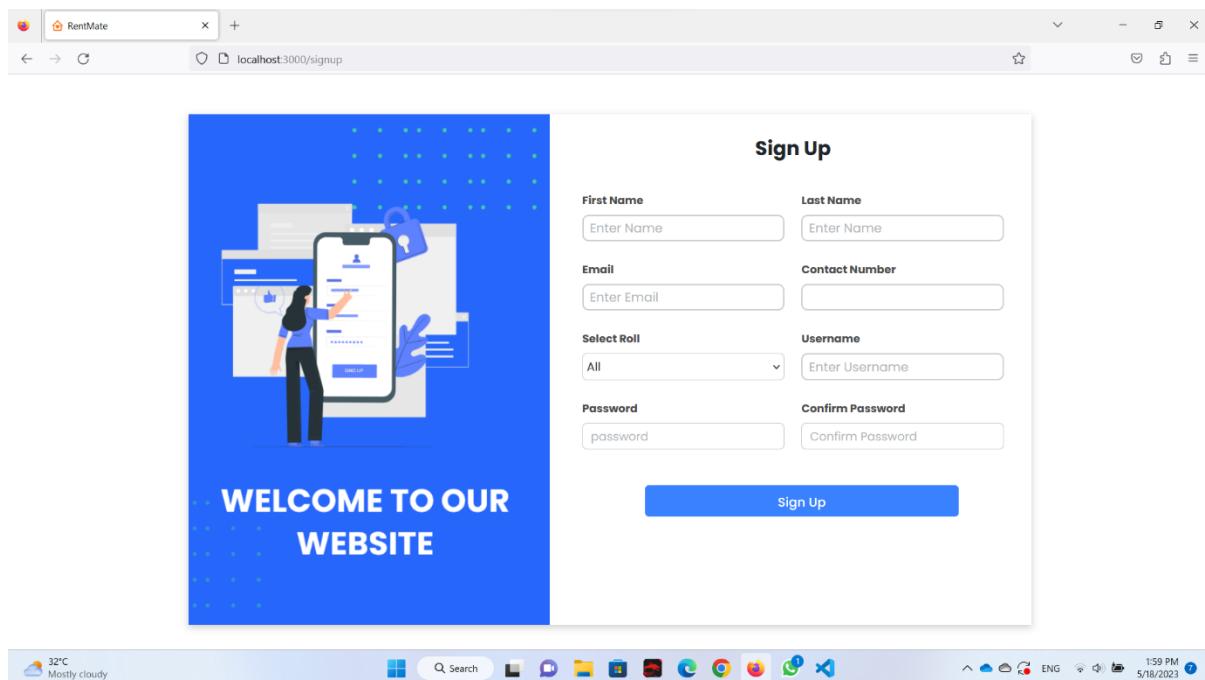


Figure 3. 48 - SignUp Page

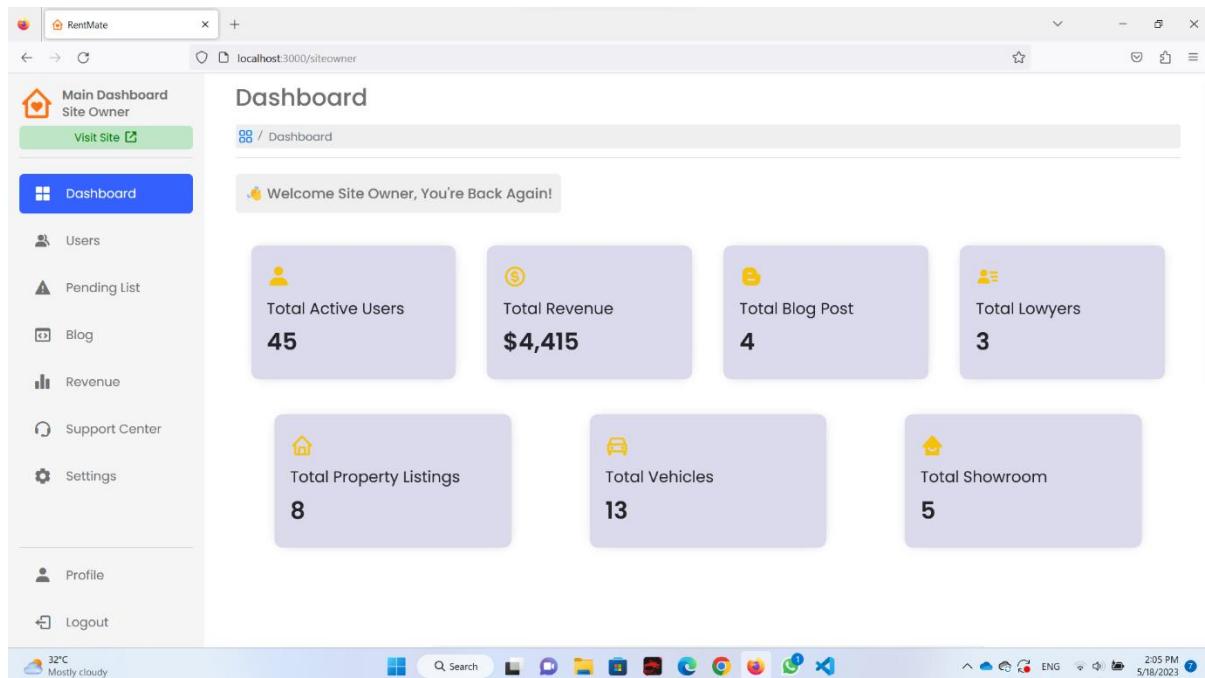


Figure 3. 49 - Site Owner Dashboard

The screenshot shows the RentMate All Users Display Page. The sidebar includes: Main Dashboard, Visit Site, Dashboard, **Users**, Pending List, Blog, Revenue, Support Center, Settings, Profile, and Logout. The main area is titled 'Dashboard' and shows a search bar, a 'Print' button, and a 'New Users' button. Below is a table listing users:

First Name	Last Name	Email	Contact Number	Role	Username	Operation
vindy	hewage	vindy@gmail.com	9874890257	Landlord	vindy	<button>Update</button> <button>Delete</button>
nayani	chamleenn	ny56ni@gmail.com	1234567867	Showroom Owner	nayani	<button>Update</button> <button>Delete</button>
isuwash	hewage	isuuh@gmail.com	0703636127	Site Owner	isuuh	<button>Update</button> <button>Delete</button>
Diwan	Sachindu	diwan@gmail.com	0766598143	Landlord	diwansachidu	<button>Update</button> <button>Delete</button>
roottoor	I232312	randula981@gmail.com	0713861718	Vehical Owner	randula98	<button>Update</button> <button>Delete</button>
Eshan	rweffg	D@gmail.com	0774991084	Regular User	eshan	<button>Update</button> <button>Delete</button>
Kasun	Chamara	kasun@gmail.com	0781236785	Regular User	kasun	<button>Update</button> <button>Delete</button>
Sithum	Fernando	sithum@gmail.com	0762313775	Showroom Owner	sithum99	<button>Update</button> <button>Delete</button>
Test	TEst	randula98@gmail.com	0713861718	Showroom Owner	randula98	<button>Update</button> <button>Delete</button>

The status bar at the bottom shows the weather as 32°C Mostly cloudy, the date as 5/18/2023, and the time as 2:00 PM.

Figure 3. 50 - All Users Display Page

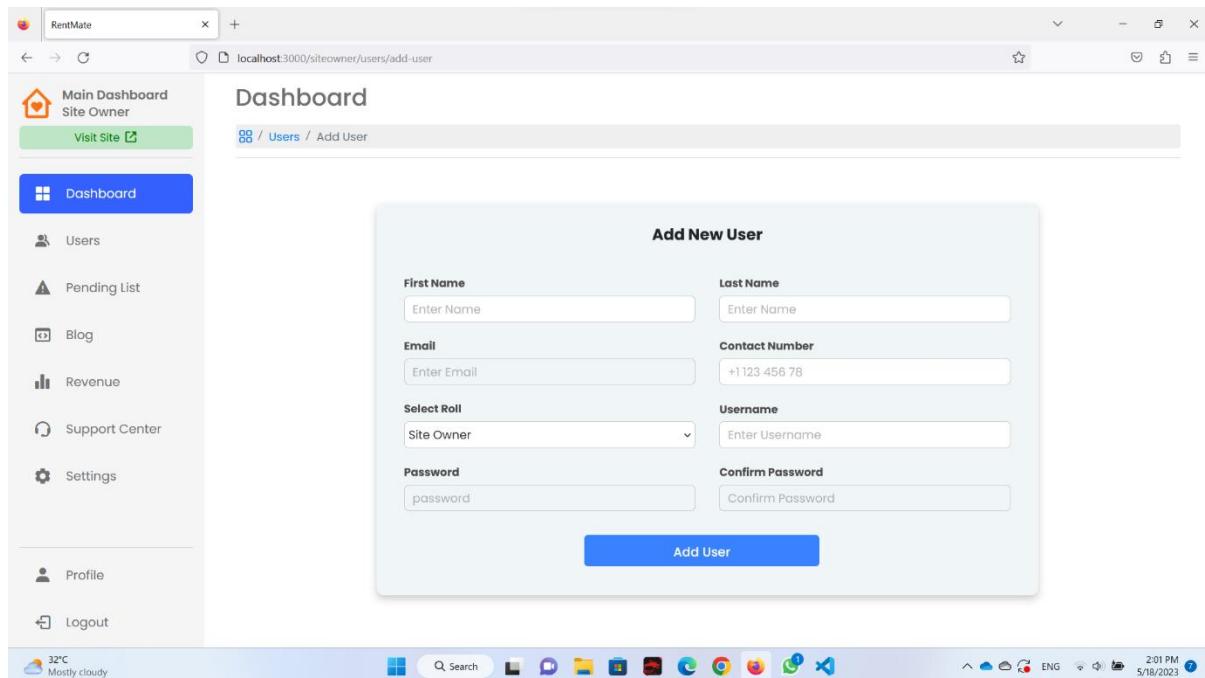


Figure 3. 51 - Add new user page

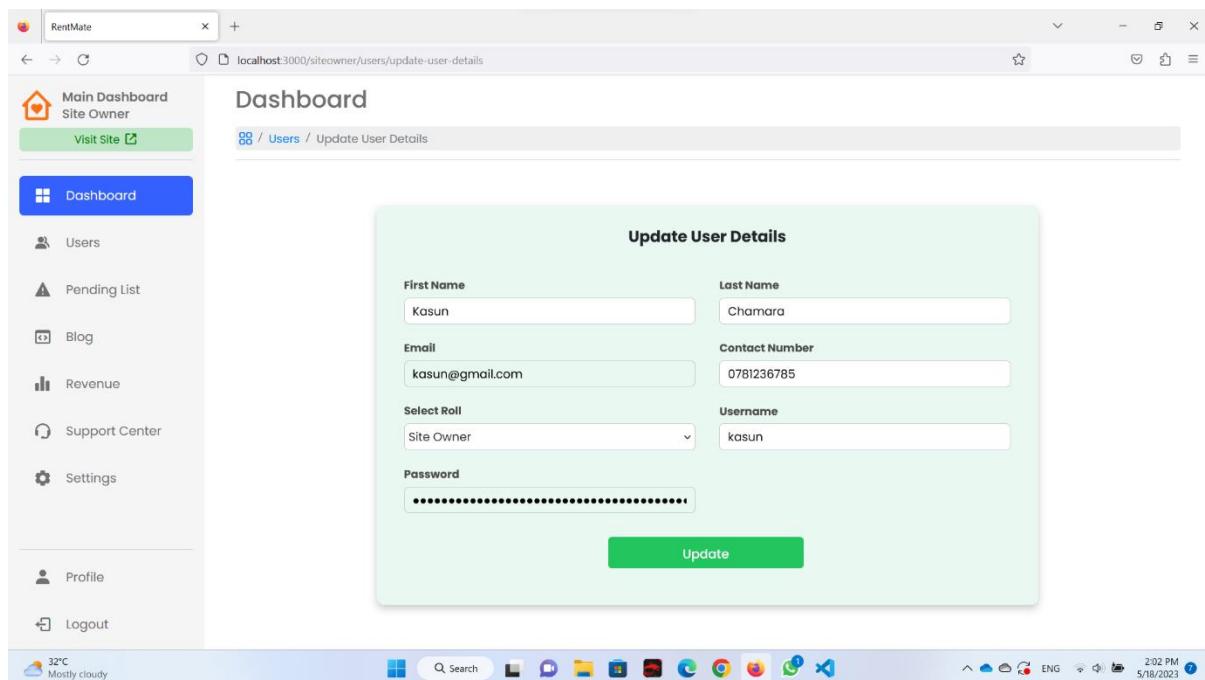


Figure 3. 52 - Update user page

**Pending Properties**

Name	Location	Bedrooms	Beds	Baths	Status	Operation
Dsm	Tambon Thung Kwo, Chang Wat Phrae, Thailand	1	1	1	✓ Active	Accept Delete
OceanEye bed (room 5)	Beliatta, Southern Province, Sri Lanka	12	2	1	✓ Active	Accept Delete
Test 09	Location 09	5	4	2	✓ Active	Accept Delete
d	df	5	5	5	✓ Active	Accept Delete
Test 10	Location 10	10	5	3	✓ Active	Accept Delete
Test 11	Test 11	5	4	2	⌚ Pending	Accept Delete
acdc	Test 11	5	4	2	⌚ Pending	Accept Delete

Figure 3. 53 - Pending Properties Page

**Pending Vehicles**

Name	Email	Contact	Date	Status	Operation
asdasd	randula98@gmail.com	0713861718	2023-05-11	✓ Active	Accept Delete
MiniTruck	perera@gmail.com	0765678781	2023-05-19	✓ Active	Accept Delete
DumpTruck	nimal@gmail.com	0734567890	2023-05-23	✓ Active	Accept Delete
Truck	nisal@gmail.com	0785667890	2023-05-26	✓ Active	Accept Delete
Minicooper	suresh@gmail.com	0673456789	2023-05-20	✓ Active	Accept Delete
dutyTruck	vije@gmail.com	0678978671	2023-05-30	⌚ Pending	Accept Delete
sdf	sdf@SF.com	0713861718	2023-05-18	⌚ Pending	Accept Delete

Figure 3. 54 - Pending Vehicles Page

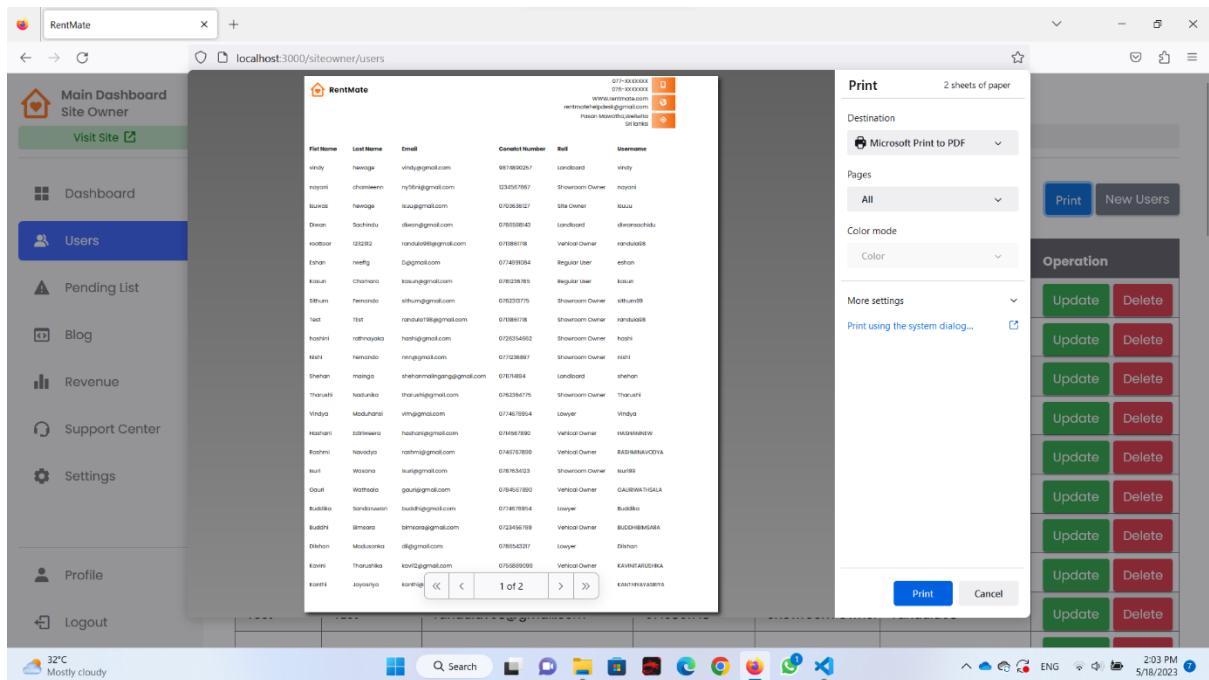


Figure 3. 55 - Generate all user report

*Legal Contact and Support Center Managements [IT21240706 – Dhananjana B.K.T]*

The screenshot shows a web browser window titled "RentMate" with the URL "localhost:3000/siteowner/support-center". The left sidebar, titled "Main Dashboard Site Owner", includes links for Dashboard, Users, Pending List, Blog, Revenue, Support Center (which is highlighted in blue), and Settings. The main content area is titled "Support Center" and shows a table of support tickets. The table columns are Name, Email, Contact Number, Date, Type, Problem Description, Status, and Operation. The data in the table is as follows:

Name	Email	Contact Number	Date	Type	Problem Description	Status	Operation
Nayani	nay@gmail.com	776857556	2023-05-04	Payment	How payment	<span>Solved</span>	<span>Reply</span> <span>Delete</span>
Nayani	nay@gmail.com	786548364	2023-05-04	SignUp	How sign up	<span>Pending</span>	<span>Reply</span> <span>Delete</span>
Nayani	nay@gmail.com	776543244	2023-05-03	SignUp	How sign up	<span>Pending</span>	<span>Reply</span> <span>Delete</span>
Nayani	nay@gmail.com	774678954	2023-05-03	Payment	How do payment	<span>Pending</span>	<span>Reply</span> <span>Delete</span>
Vindy	vin@gmail.com	786548364	2023-05-02	Payment	How do payment	<span>Pending</span>	<span>Reply</span> <span>Delete</span>

The bottom status bar shows weather (28°C, Mostly cloudy), system icons, and the date/time (9:57 PM, 5/18/2023).

Figure 3. 57 - Support Center Page

The screenshot shows a web browser window titled "RentMate" with the URL "localhost:3000/siteowner/support-center/add-record". The left sidebar is identical to Figure 3.57. The main content area is titled "Support Center" and shows a form titled "Add New Record Form". The form fields are:

- Name: (input field)
- Email: (input field)
- Contact Number: (input field)
- Date: (input field with placeholder "mm / dd / yyyy")
- What Kind of Problem: (dropdown menu with placeholder "-- Select an option --")
- Problem: (text area)

A green "Add Record" button is at the bottom of the form.

The bottom status bar shows weather (28°C, Mostly cloudy), system icons, and the date/time (9:58 PM, 5/18/2023).

Figure 3. 56 - Support Center Add New Record Form

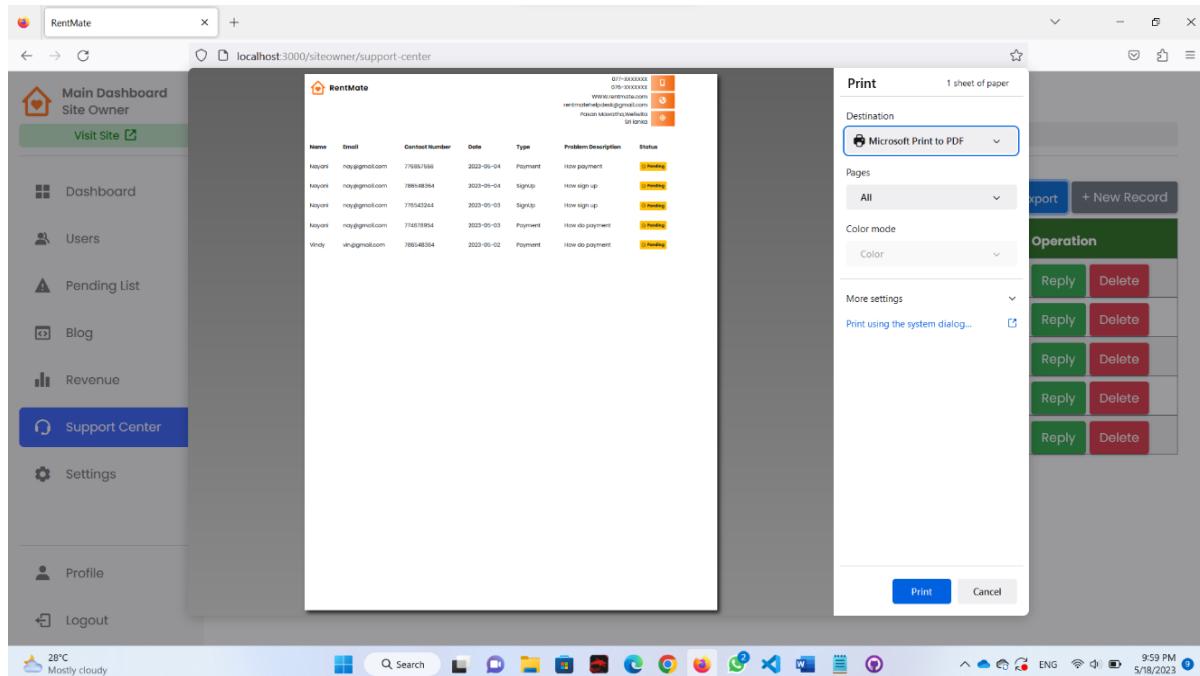


Figure 3. 58 - Support Center report generating

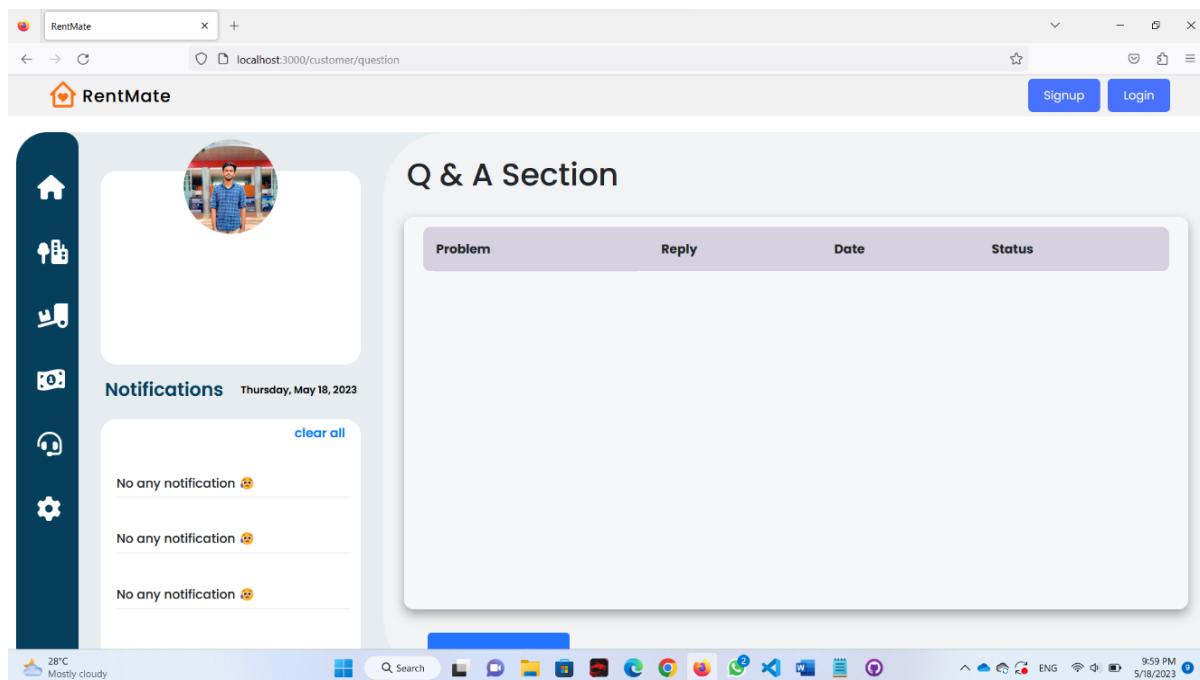


Figure 3. 59 - Profile Q&A Page

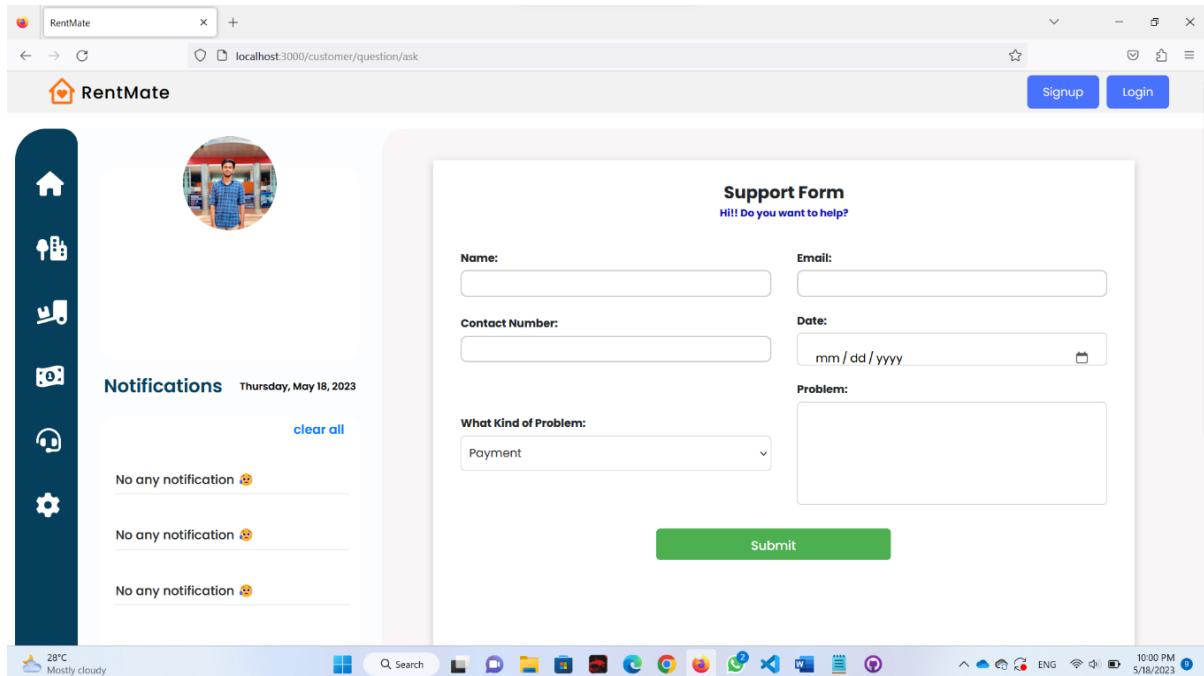


Figure 3. 60 - Profile Contact Support Page

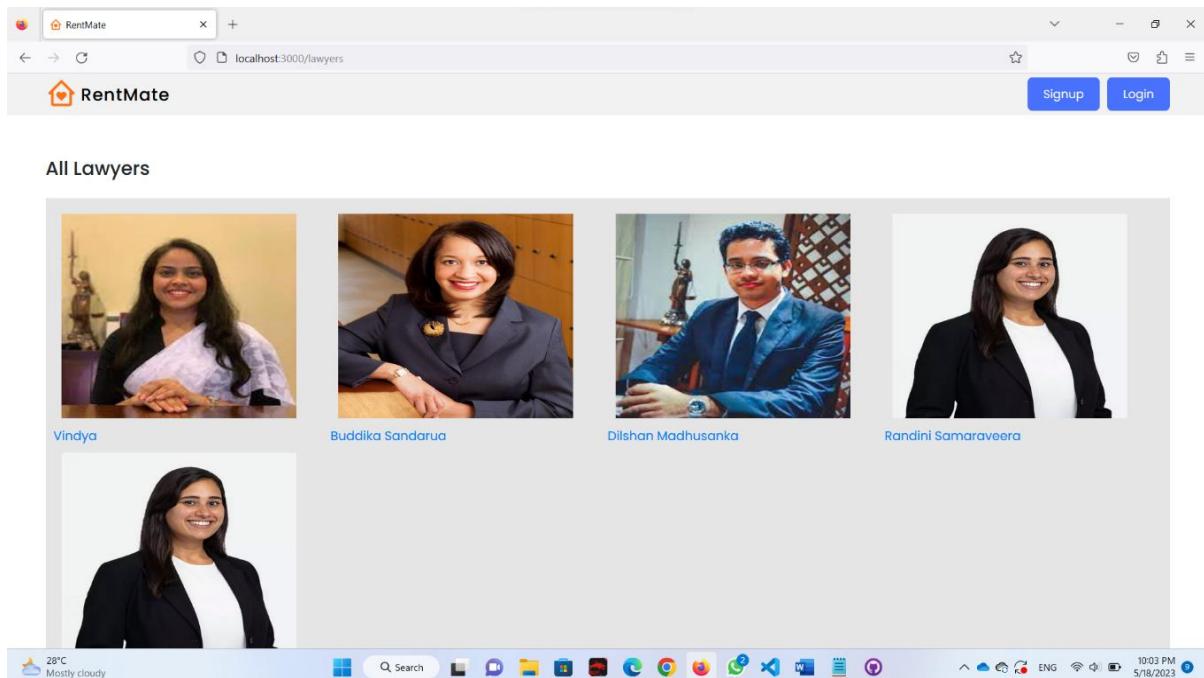


Figure 3. 61 - View All Lawyers Page

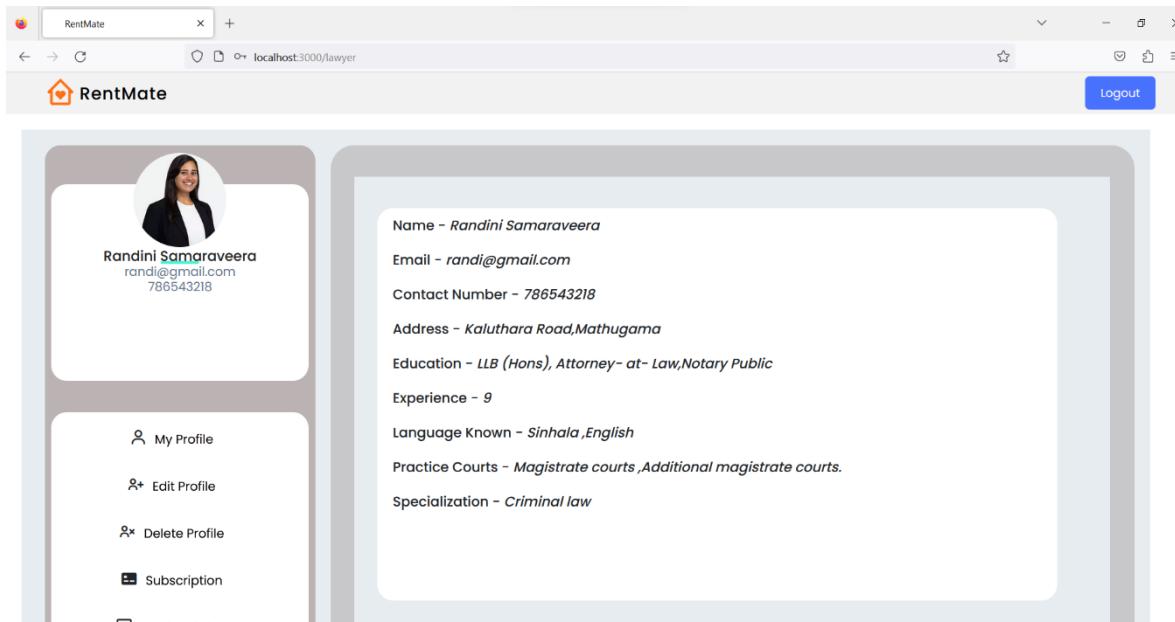


Figure 3. 62 – Lawyer Profile Page

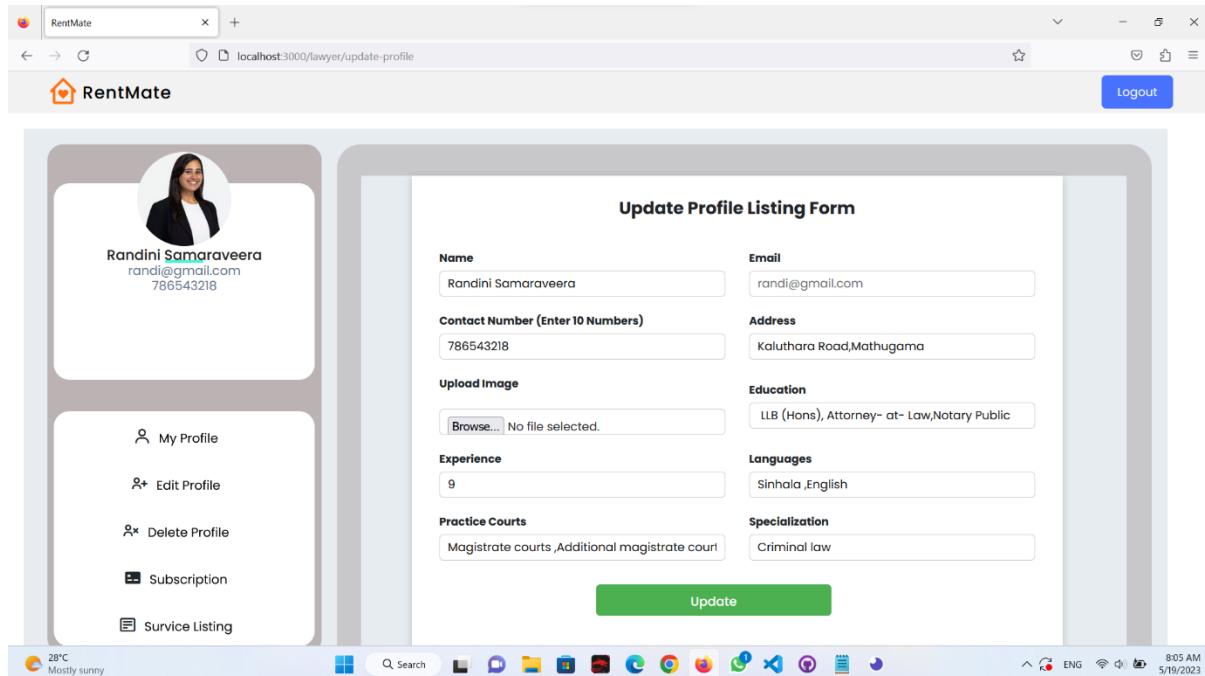


Figure 3. 63 – Lawyer Update Info Updatee Page

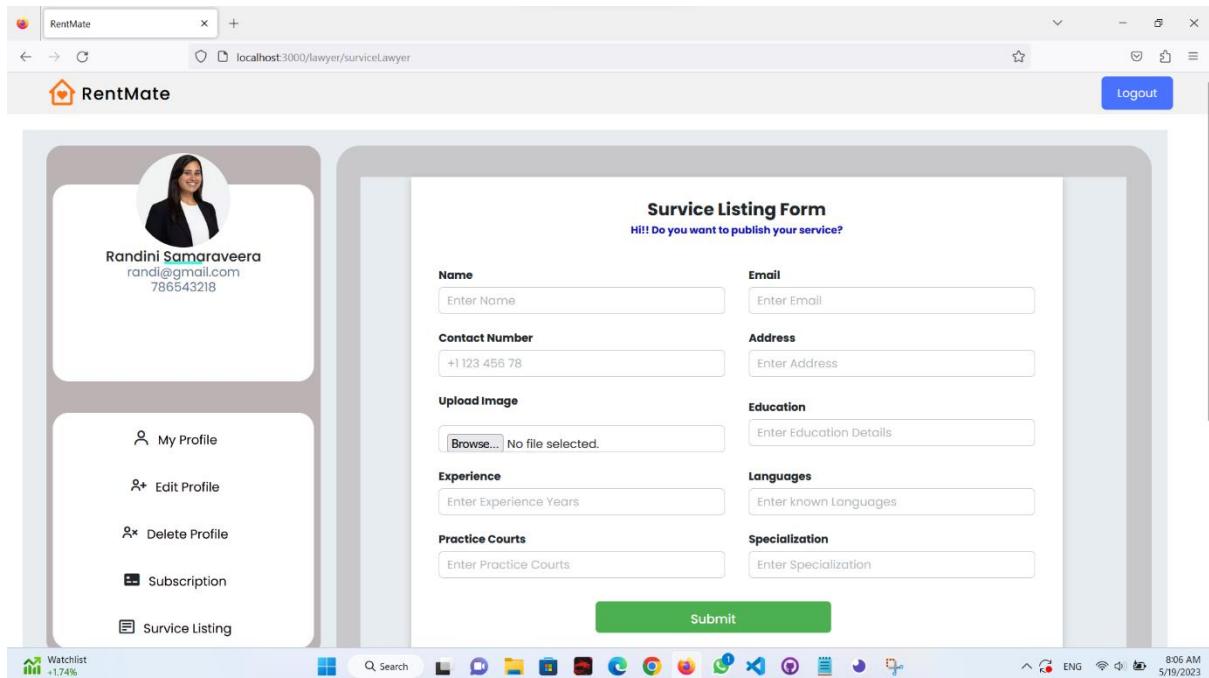


Figure 3. 64 – Lawyer Service Listing Page

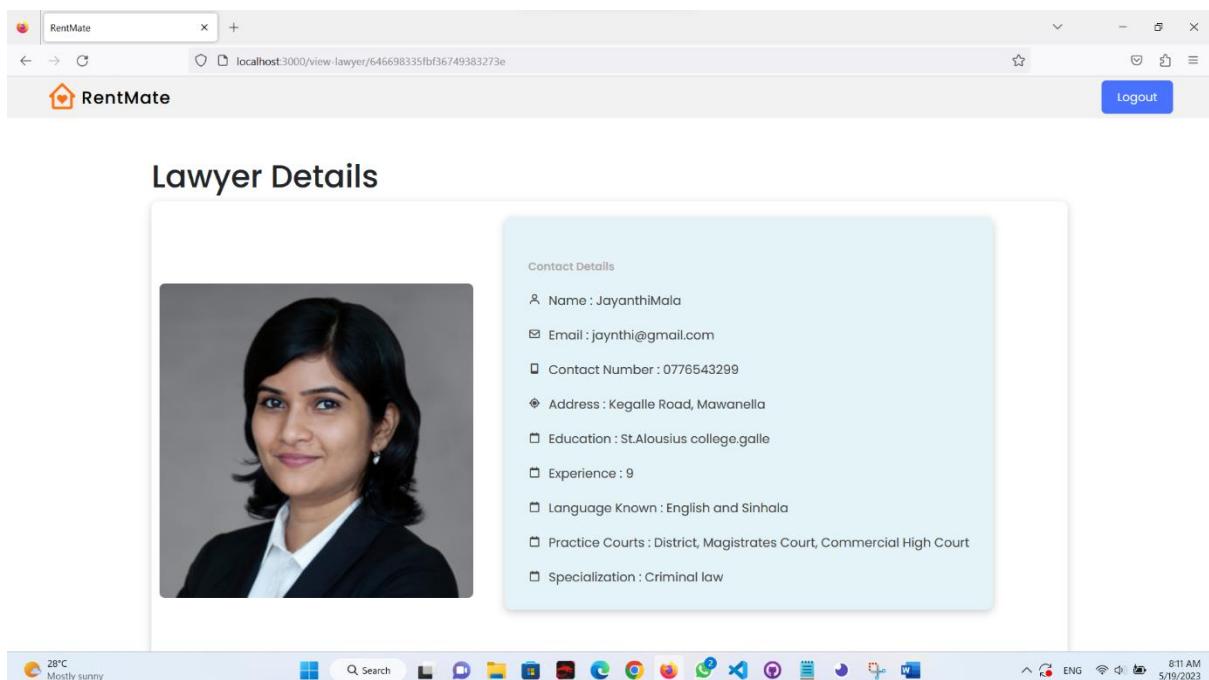


Figure 3.65 - Single Layer Page

## **3.5 Databases**

### **3.5.1 Listings**

propertyID	PropertyName	address	rooms	beds	baths	price	type	description	map Link
------------	--------------	---------	-------	------	-------	-------	------	-------------	----------

### **3.5.2 Customer**

userId	fname	lname	email	phone	address	city	avatar
--------	-------	-------	-------	-------	---------	------	--------

### **3.5.3 Todo**

UserId	content
--------	---------

### **3.5.4 Users**

userID	fName	lName	email	contactNo	roll	username	password
--------	-------	-------	-------	-----------	------	----------	----------

### **3.5.5 Vehicles**

name	email	contact	date	status	image
------	-------	---------	------	--------	-------

### **3.5.6 Revenue**

name	vin	contact	date	price
------	-----	---------	------	-------

### **3.5.7 Bookings**

date	vin	pick Up Location	drop Off Locations
------	-----	------------------	--------------------

### **3.5.8 Card Details**

cardType	name	cardNumber	Exp_month	Exp_year	cvv
----------	------	------------	-----------	----------	-----

### **3.5.9 Payment Details**

PaymentID	paymentType	UserID	ContactNo	reservervationDetail	PaymentDate	reservationClosingDate
vehicleID	sLocation	Location	distance	cardNumber	amount	

### **3.5.10 Blogs**

title	id	content	image	date	idNo
-------	----	---------	-------	------	------

### **3.5.11 Lawyers**

ID	name	email	contactNumber	address	education	image	experience	languages	practiceCourt	specialization
----	------	-------	---------------	---------	-----------	-------	------------	-----------	---------------	----------------

**3.5.12 Support**

ID	name	email	contactNumber	date	problemtype	problem	status
----	------	-------	---------------	------	-------------	---------	--------

**3.5.13 Showrooms**

ID	ShowroomName	Email	ContactNumber	Address	City	OpencloseTime	City
----	--------------	-------	---------------	---------	------	---------------	------

**3.5.14 ShowroomsImages**

ID	Image1	Image2	Image3
----	--------	--------	--------

**3.5.15 Furnitures**

ID	FurnitureName	Price	Quantity	Category	Image
----	---------------	-------	----------	----------	-------

**3.6 Development Aspects**

This report provides an overview of the development aspects of our project, which utilized the MERN stack. The MERN stack, consisting of MongoDB, Express.js, React, and Node.js, offers a powerful framework for developing full-stack web applications. Throughout this report, we will explore the key elements of our development process, including technology selection, architecture and design, front-end development, back-end development, and deployment. By harnessing the capabilities of the MERN stack, we successfully built a robust and user-friendly application that met our project's requirements. This report serves as a documentation of our journey, highlighting the methodologies and techniques we employed in leveraging the MERN stack for our project.

**3.6.1. M – MongoDB**

MongoDB is a database system that stores data in flexible and dynamic JSON-like documents. This allows for varying fields between documents and the ability to modify the data structure as needed. The document model in MongoDB aligns with the objects used in the application code, making data management straightforward. With MongoDB, you can perform ad hoc queries, create indexes, and perform real-time aggregations, enabling efficient data access and analysis. Moreover, MongoDB is designed to be a distributed database, providing inherent features for high availability, horizontal scalability, and geographic distribution, which can be easily utilized. [3]

### 3.6.2 E – Express

Express is a web application framework for Node.js that offers a wide range of features and is known for its versatility and lightweight nature. It caters to the needs of both web and mobile applications, providing comprehensive functionality.

One of the notable strengths of Express is its seamless API creation capabilities. It provides an abundance of HTTP utility methods and middleware options, allowing developers to build robust APIs quickly and effortlessly.

Additionally, Express excels in performance by adding a minimal layer of essential web application features without disrupting the core functionalities of Node.js. This enables efficient and high-performing web applications without sacrificing the benefits and capabilities of the underlying Node.js platform. [4]

### 3.6.3 R – React

React is a JavaScript library that serves as a powerful tool for building user interfaces. It offers several notable features that contribute to its popularity and effectiveness in UI development.

Firstly, react adopts a declarative approach, which simplifies the creation of interactive user interfaces. By defining straightforward views for each stage of your application, react intelligently updates and renders only the essential components when data changes. This declarative nature enhances the predictability, comprehension, and debuggability of your code, making it easier to develop and maintain.

Secondly, react follows a component-based architecture, allowing you to create encapsulated components that manage their own state. These components can then be combined to build complex user interfaces. Unlike traditional template-based approaches, React's component logic is written in JavaScript, enabling the seamless passage of rich data throughout your application while keeping the state separate from the Document Object Model (DOM).

Lastly, react adheres to the principle of "learn once, write anywhere." This means that react does not impose constraints on the rest of your technology stack. You can easily incorporate React into your existing codebase without the need for extensive rewrites. Moreover, react can be used to render components on the server side with Node.js and empowers the development of mobile applications through React Native.

In summary, react is a versatile JavaScript library for building user interfaces. Its declarative approach, component-based architecture, and platform flexibility make it a popular choice among developers seeking efficient and scalable UI development solutions. [5]

### 3.6.4 N – NodeJS

Node.js is a free and open-source server environment that runs on different platforms like Windows, Linux, Unix, and Mac OS X. It uses JavaScript on the server side. One of its key strengths is its ability to handle multiple requests simultaneously without waiting. This makes it highly responsive and scalable. Additionally, Node.js is known for its efficient memory usage, making it optimized for resource utilization. In summary, Node.js is a versatile and efficient server environment that is widely used for building responsive applications. [6]

Node.js is capable of:

- Building web servers and web applications efficiently.
- Developing real-time applications like chat or streaming platforms.
- Creating RESTful APIs for smooth communication.
- Building command-line tools and scripts.
- Supporting backend services and microservices for scalable systems.
- Handling data streaming, file processing, and IoT applications.

In summary, Node.js offers a versatile set of capabilities for developing web servers, real-time applications, APIs, command-line tools, backend services, data streaming, and IoT applications.

### 3.6.5 – Figma

Figma is a widely used tool for designing and presenting prototypes. It lets you create interactive prototypes that show how your design will work. With Figma, multiple team members can work together on the same project at the same time. You can design interfaces, share your prototypes with others, and get feedback. Figma has easy-to-use design tools, supports reusable components, and works well with other tools. It's great for presenting and improving your designs, making it popular among designers and product teams.

### 3.6.6 – ClickUp

Clickup is a user-friendly project management tool that simplifies team collaboration and task tracking. Its customizable Kanban board allows you to visually organize and manage your projects. With Clickup, you can create tasks, set deadlines, assign team members, and track progress effortlessly. It seamlessly integrates with popular tools like Slack, ensuring smooth communication and workflow. Click Up's mobile apps enable you to stay connected and manage your projects on the go. Overall, Clickup streamlines project management, making it a valuable solution for teams to stay productive and organized.

### 3.6.7 – Insomnia

Insomnia is a user-friendly tool for testing and debugging API requests. It helps developers and QA professionals send and analyze HTTP requests easily. With Insomnia, you can organize and manage API request collections, authenticate requests, and validate responses. It supports team collaboration, offers advanced features like environment variables, and allows for customization through plugins. Insomnia simplifies API testing, ensuring the reliability and functionality of your APIs.

### 3.6.8 – GitHub

GitHub is a widely used platform for version control and collaboration in software development. It leverages Git, a distributed version control system, to track changes and manage code repositories. With GitHub, teams can easily collaborate, review, and improve code, and manage project tasks. It integrates seamlessly with a variety of tools, enabling streamlined workflows. GitHub's extensive documentation and active community make it a valuable resource for developers worldwide. It helps maintain code quality, facilitates efficient collaboration, and ensures the integrity of software projects.

### 3.6.9 – Postman

Postman is a user-friendly tool for testing and documenting API requests. It allows you to send various types of requests, including GET, POST, PUT, and DELETE, to API endpoints. With Postman, you can easily customize headers, handle authentication, and validate responses. It provides features for organizing requests, automating tests, and collaborating with team members. Postman's extensive documentation and community support make it a valuable resource for API development and testing. It ensures the reliability and accuracy of your APIs.

### 3.6.10 – Visual Code

Visual Studio Code, also known as VS Code, is a popular and user-friendly code editor. It offers a lightweight and customizable environment with powerful features for writing code efficiently. With intelligent code completion and error checking, it helps improve code quality. Visual Studio Code supports collaboration through Git integration and real-time sharing. Its clean interface can be customized with themes and shortcuts. It provides excellent debugging support with breakpoints and variable inspection. Available on multiple platforms, Visual Studio Code is widely used by developers worldwide.

## Chapter 04 – Testing

---

### 4.1 Testing Methods

#### 4.1.1 Unit Testing

Unit testing is an important part of software development. It involves testing small parts of the application, known as units, to make sure they work correctly. Software engineers create automated tests for these units and run them to check if the code functions as expected. The goal of unit testing is to ensure that each individual part of the software performs its intended tasks accurately.

#### 4.1.2 Integration Testing

Integration testing is an essential phase in software testing. It involves combining and testing individual software modules as a group. The main purpose of integration testing is to determine if a system or component meets a set of functional requirements. This testing occurs after unit testing but before system testing, ensuring that all the integrated parts work together correctly.

## **4.2 Listing Management System**

*Table 4. 1 – Listing Management Test Cases*

<b>Project ID: ITP_WD_B04_G09</b>										
<b>Project Name: Online Marketplace for Homestays</b>										
<b>Testing function:</b> Adding a New Listing										
<b>Test case ID:</b> 01	Test case designed by, ID: IT21166488 Name: Maleesha K.L.D.D.S									
<b>Test Priority</b> <b>(High/Medium/Low)</b>	High									
<b>Test description:</b> This test case verifies the functionality of the listing management system by testing the process of adding a new listing to the website.										
<b>Preconditions (if there are any): Landlord must be logged into the system</b>										
<b>Test Steps:</b>										
<ol style="list-style-type: none"> <li>1. Visit the homepage.</li> <li>2. Log onto the website and navigate to the dashboard.</li> <li>3. Navigate to the Listing Page.</li> <li>4. Click on the "Add New Listing" button.</li> <li>5. Fill in all the required details, such as title, description, location, price, and key details.</li> <li>6. Upload relevant images for the listing.</li> <li>7. Click on the "Add Property" button to add the listing.</li> <li>8. Display successful message and redirect to the listing page.</li> </ol>										
Test ID	Test Inputs	Expected Outputs	Actual Output	Result	Comments					
FTI001_01	Title, description, location, price, and other key details.	The listing should be successfully added to the marketplace, and the user should be able to view the newly added listing in the list of available rentals.	The listing is successfully added to the marketplace, and the user can see the newly added listing in the list of available rentals.	Pass	<ul style="list-style-type: none"> <li>• All the validations are working properly.</li> <li>• Data successfully inserted to the database.</li> <li>• Redirect to the listing page, place where display all the added listings.</li> </ul>					

Table 4. 2 – Listings Management Test Cases

<b>Project ID: ITP_WD_B04_G09</b>										
<b>Project Name: Online Marketplace for Homestays</b>										
<b>Testing function:</b> Updating an Existing Listing										
<b>Test case ID: 01</b>	Test case designed by, ID: IT21166488 Name: Maleesha K.L.D.D.S									
<b>Test Priority (High/Medium/Low)</b>	High									
<b>Test description:</b> This test case verifies the functionality of the listing management system by testing the process of updating an existing listing on the website. This help to make the listing up to date.										
<b>Preconditions (if there are any): Landlord must be logged into the system</b>										
<b>Test Steps:</b>										
<ol style="list-style-type: none"> <li>1. Visit the homepage.</li> <li>2. Log onto the website and navigate to the dashboard.</li> <li>3. Navigate to the Listing Page.</li> <li>4. Click on the update button on the specific listing.</li> <li>5. Redirect to the listing updating page.</li> <li>6. Update necessary details.</li> <li>7. Click on the “update button”.</li> <li>8. Redirect to the listing page.</li> </ol>										
Test ID	Test Inputs	Expected Outputs	Actual Output	Result	Comments					
FTI001_01	Details that needed to be updated	The changes made to the listing should be saved successfully, and the updated information should be updated and display on the website.	The changes made to the listing are saved successfully, and the updated information is display on the website.	Pass	<ul style="list-style-type: none"> <li>• Verify updated information inserted to the database successfully.</li> <li>• All the validations are working properly.</li> <li>• Redirect to the listing page, place where display all the added listings.</li> </ul>					

#### 4.3 Payment Management

Table 4. 3 - Payment Management Test Cases

<b>Project ID: ITP_WD_B04_G09</b>										
<b>Project Name: Online Marketplace for Homestays</b>										
<b>Testing function:</b> Enter a new Card Detail										
<b>Test case ID: 01</b>	Test case designed by, ID: IT21183768 Name: W.A.S. Heshan									
<b>Test Priority (High/Medium/Low)</b>	Medium									
<b>Test description:</b> when a customer wants to reserve a property or service, they should add card details to the system and pay										
<b>Test Steps:</b>										
1: Customer should be registered 2: Click reserve a property. 3: Click Add card Button 4: Enter New Card Details 5: Click Submit 6: System prompts added successful message										
Test ID	Test Inputs	Expected Outputs	Actual Output	Result	Comments					
01	Card Number: - 4605592014 635289	Display ‘Check the Car Number length’ message at the top of the relevant field	Display ‘Check the Car Number length’ message at the top of the relevant field	Pass	Data successfully inserted to the database.					
02	Card Number: - 4605592014 6352	Display ‘Card Number should be 16 digits’ message at the top of the relevant field	Display ‘Card Number should be 16 digits’ message at the top of the relevant field	Pass						
03	Expiry Month: - 15	Display ‘Month between 1-12’ message at the top of the relevant field	Display ‘Month between 1-12’ message at the top of the relevant field	Pass						
04	CVV: - 1452	Display ‘Check the CVV Number length’ message at the top of the relevant field	Display ‘Check the CVV Number length’ message at the top of the relevant field	Pass						

#### **4.4 Blog Management**

*Table 4. 4 - Blog Management Test Cases*

<p>Project ID: ITP_WD_B04_G09</p> <p>Project Name: Online Marketplace for Homestays</p> <p><b>Testing function: Blog Management - Add New blog post.</b></p>										
<b>Test case ID: 01</b>	Test case designed by, ID:IT21186424 Name: N.G.S.S.M Bandara									
<b>Test Priority (High/Medium/ Low)</b>	Medium									
<b>Test description:</b> Create a new Blog Post										
<p><b>Test Steps:</b></p> <p>1: Admin Login to the system using login credentials.</p> <p>2: Navigate to the blog list page.</p> <p>3: Click the add new button.</p> <p>4: Navigate to the add blog page.</p> <p>5: Add necessary details.</p> <p>6: Click on POST button</p> <p>7: Blog post added to the blog page</p>										
Test ID	Test Inputs	Expected Outputs	Actual Output	Result	Comments					
Test_001	<u>Inputs</u> Title Id Content Date Image category	Display ‘all fields should be filled’ message	Display ‘all fields should be filled’ message	Pass						
Test_002	<u>Inputs</u> Title Id Content Date Image Category	If successfully added blog details, that blog record will be displayed in blog page when clicked view the blog page	If successfully added blog details, that blog record will be displayed in blog page when clicked view the blog page	Pass	Data successfully inserted to the database.  Added blog post are displaying in the blog page					

#### **4.5 Customer Management**

*Table 4. 5 - Customer Management Test Cases*

<b>Project ID: ITP_WD_B04_G09</b>										
<b>Project Name: Online Marketplace for Homestays</b>										
<b>Testing function:</b> Add customer personal details										
<b>Test case ID:</b> <b>UM001</b>	Test case designed by, ID: IT21233562 Name: Ranawaka W. E. I									
<b>Test Priority (High/Medium/Low)</b>	High									
<b>Test description:</b> After Customer register to system customer should enter their personal details into system.										
<b>Test Steps:</b>										
<ol style="list-style-type: none"> <li>1. Visit customer profile.</li> <li>2. Go to settings and click personal info.</li> <li>3. Click edit icon in each field.</li> <li>4. Fill all field in the form with valid details.</li> <li>5. Click on save button in each field</li> </ol>										
<b>Test ID</b>	<b>Test Inputs</b>	<b>Expected Outputs</b>	<b>Actual Output</b>	<b>Result</b>	<b>Comments</b>					
<b>CD001_01</b>	All fields are kept empty:  <ul style="list-style-type: none"> <li>▪ Name</li> <li>▪ Email</li> <li>▪ Phone No</li> <li>▪ Address</li> <li>▪ Current city</li> <li>▪ Profile picture</li> </ul>	Display ‘Can’t be empty fields’(All fields are required)	Display ‘Can’t be empty fields’ (All fields are required)	Pass						
<b>CD001_02</b>	Email: Example2@gmail.com	Input field border colour should be change in to red	Input field border colour change in to red	Pass						
<b>CD003_03</b>	Email: user@gmail.com	Input field border colour should be change in to green.  Display successfully updated alert.	Input field border colour should be change in to green.  Display successfully updated alert.	Pass						

<b>Testing function:</b> Change customer password										
<b>Test case ID:</b> <b>UM001</b>	Test case designed by, ID: IT21233562 Name: Ranawaka W. E. I									
<b>Test Priority</b> <b>(High/Medium/Low)</b>	High									
<b>Test description:</b> when customer need to change their password to protect their personal details										
<b>Test Steps:</b>										
<ol style="list-style-type: none"> <li>1. Visit customer profile.</li> <li>2. Go to settings and click login and security section.</li> <li>3. Click update password button.</li> <li>4. Fill all field in the form with valid and strong password.</li> <li>5. Click update button</li> </ol>										
<b>Test ID</b>	<b>Test Inputs</b>	<b>Expected Outputs</b>	<b>Actual Output</b>	<b>Result</b>	<b>Comments</b>					
<b>CD001_01</b>	All fields are kept empty:	Display ‘Cannot be empty fields’ (All fields are required)	Display ‘Cannot be empty fields’ (All fields are required)	Pass						
<b>CD001_02</b>	New Password: Imesh2001 Confirmation Password: Eshan2001	Display alert “password miss matched”	Display alert “password miss matched”	Pass						
<b>CD003_03</b>	New Password: Imesh\$2001 Confirmation Password: Imesh\$2001	Display “password successfully updated “alert	Display “password successfully updated “alert.	Pass						

#### **4.6 Furniture Showroom Management**

*Table 4. 6 - Furniture Showroom Management Test Cases*

<b>Project ID: ITP_WD_B04_G09</b>										
<b>Project Name: Online Marketplace for Homestays</b>										
<b>Testing function:</b> Check contact number in the add showroom form.										
<b>Test case ID:</b> FSM001	Test case designed by, ID: IT21184376 Name: Jayathilaka A.W.N.M									
<b>Test Priority</b> <b>(High/Medium/Low)</b>	Medium									
<b>Preconditions (if there are any):</b> User owner must log in to the system as a Showroom Owner.										
<b>Test description:</b> Add new furniture showroom										
<b>Test Steps:</b>										
1: Navigate to the showroom owner dashboard. 2: Pay subscription fee. 3: Click showroom button. 4: Enter showroom details. 5: Click add button.										
<b>Test ID</b>	<b>Test Inputs</b>	<b>Expected Outputs</b>	<b>Actual Output</b>	<b>Result</b>	<b>Comments</b>					
FSM001_01	All fields are kept empty.	Display “All fields are required” message.	Display “All fields are required” message.	Pass						
FSM001_02	09873242ab	Display “Enter only numbers” message at the bottom of the relevant field.	Display “Enter only numbers” message at the bottom of the relevant field.	Pass						
FSM001_03	07623132245	Display “Contact number should contain exactly 10 numbers” message at the bottom of the relevant field.	Display “Contact number should contain exactly 10 numbers” message	Pass						

			at the bottom of the relevant field.		
FSM001_04	0762313	Display “Contact number should contain exactly 10 numbers” message at the bottom of the relevant field.	Display “Contact number should contain exactly 10 numbers” message at the bottom of the relevant field.	Pass	
FSM001_05	All input fields are filled accurately.	Display “Showroom added successfully” massage.	Display “Showroom added successfully” massage.	Pass	Data successfully added into the database.

#### **4.7 Transport Management System**

*Table 4. 7 - Transport management Test Cases*

<b>Project ID: ITP_WD_B04_G09</b>										
<b>Project Name: Online Marketplace for Homestays</b>										
<b>Testing function:</b> Register a new user to the system.										
<b>Test case ID:</b> <b>01</b>	Test case designed by, ID: IT21197796 Name: Chamaleen D.B. N									
<b>Test Priority (High/Medium/Low)</b>	Medium									
<b>Test description:</b> When a vehicle owner adds a vehicle to the system, the owner should register to the system.										
<b>Preconditions (if there are any): Vehicle owner must be logged into the system</b>										
<b>Test Steps:</b> 1: Vehicle owner should be logged into the system 2: Click on the add vehicle button 3: Enter new vehicle details 4: Click on add button 5: System prompts added successful message										
<b>Test ID</b>	<b>Test Inputs</b>	<b>Expected Outputs</b>	<b>Actual Output</b>	<b>Result</b>	<b>Comments</b>					
FTI001_01	07123456763	Display ‘Check the contact no length’ message at the bottom of the relevant field.	Display ‘Check the contact no, length’ message at the bottom of the relevant field	Pass						
FTI001_02	0712345678Bc	Display ‘Invalid input format’ message at the bottom of the relevant field	Display ‘Invalid input format’ message at the bottom of the relevant field	Pass						
FTI002_01	@112gmail.com	Display ‘Invalid input format’ message at the email should not starting the ‘@’ symbol	Display ‘Invalid input format’ message at the email should not starting the ‘@’ symbol	Pass						
FTI002_02	AB99@gmail.com	Display ‘Invalid input format’ message at the email should not starting the ‘AB’ like that capital letters	Display ‘Invalid input format’ message at the email should not starting the ‘AB’ like that capital letters	Pass						

#### **4.8 User Management**

*Table 4. 8 - User Management Test Cases*

<b>Project ID: ITP_WD_B04_G09</b>										
<b>Project Name: Online Marketplace for Homestays</b>										
<b>Testing function:</b> Register a new user to the system.										
<b>Test case ID:</b> <b>UM001</b>	Test case designed by, ID: IT21197796 Name: Wasana K.H.I.M									
<b>Test Priority</b> <b>(High/Medium/Low)</b>	High									
<b>Test description:</b> When a user visits to our site and needs to sell a home or vehicle and add showroom or lawyer, the user should register to the system.										
<b>Test Steps:</b>										
<ol style="list-style-type: none"> <li>1. Visit to the home page.</li> <li>2. Click on the sign-up button to register the system.</li> <li>3. Fill all the fields in the registration form with valid details.</li> <li>4. Click on the Sign-up button.</li> </ol>										
<b>Test ID</b>	<b>Test Inputs</b>	<b>Expected Outputs</b>	<b>Actual Output</b>	<b>Result</b>	<b>Comments</b>					
<b>UM001_01</b>	All fields are kept empty:  1. Name 2. Email 3. Contact No 4. Roll 5. username 6. Password 7. Confirm Password	Display ‘Can’t be empty fields’ (All fields are required)	Display ‘Can’t be empty fields’ (All fields are required)	Pass						

<b>UM001_02</b>	All input fields are filled accurately:  1. Name 2. Email 3. Contact No 4. Roll 5. username 6. Password 7. Confirm Password	All the user details should be successfully validated and passed to the database.  "Successfully Registered" alert will be show up on the screen and navigate to the login page.	All the user details should be successfully validated and passed to the database.  "Successfully Registered" alert will be show up on the screen and navigate to the login page.	Pass	Data successfully inserted to the database.
<b>UM001_03</b>	Password:  aaaa123	Display Invalid massage. 'Enter strong password with minimum eight characters'	Display Invalid massage. 'Enter strong password with minimum eight characters'	Pass	
<b>UM001_04</b>	Password:  aaaa123qgd	Display Invalid massage. 'Enter strong password'	Display Invalid massage. 'Enter strong password'	Pass	
<b>UM001_05</b>	Password:  abQW123@#	Successfully validated password	Successfully validated password	Pass	Validate the password

#### **4.9 Pending List Management**

*Table 4. 9 - Pending List Management Test Cases*

<b>Project ID: ITP_WD_B04_G09</b>										
<b>Project Name: Online Marketplace for Homestays</b>										
<b>Testing function:</b> Pending status										
<b>Test case ID:</b> <b>PM001</b>	Test case designed by, ID: IT21197796 Name: Wasana K.H.I.M									
<b>Test Priority</b> <b>(High/Medium/Low)</b>	Medium									
<b>Pre-condition:</b> Site owner should be logged in to the site.										
<b>Test description:</b> When a landlord or vehicle owner add places or vehicle it should come pending list and get approval from site owner.										
<b>Test Steps:</b> 01. Navigate to pending list page. 02. Select a specific listing. 03. Search content 04. Click Accept button. 05. ‘Pending’ status change to ‘active’ successfully.										
Test ID	Test Inputs	Expected Outputs	Actual Output	Result	Comments					
<b>PM001_01</b>	Click ‘Accept’ button	If owners entered details are valid. Then click the Accept button with inside the action bar that ‘pending’ status change to ‘active’ and it also visible in owners’ dashboard as the active successfully.	If owners entered details are valid. Then click the Accept button with inside the action bar that ‘pending’ status change to ‘active’ and it also visible in owners’ dashboard as the active successfully.	Pass	Update the database status as ‘Pending’ to ‘Active’					

#### **4.10 Legal Contact Management**

*Table 4. 10 - Legal Contact Management Test Cases*

<b>Project ID: ITP_WD_B04_G09</b>					
<b>Project Name: Online Marketplace for Homestays</b>					
<b>Testing function:</b> Legal Contact Management – Add lawyer details					
<b>Test case ID: LW001</b>	Test case designed by, ID: IT21240706 Name: Dhananjana B.K.T				
<b>Test Priority (High/Medium/Low)</b>	Medium				
<b>Test description:</b> Add lawyer details through site.					
<b>Precondition:</b> Lawyer must be logged into the system					
<b>Postcondition:</b> “Added successful” message should appear					
<b>Test Steps:</b> 1: Visit to the home page. 2: Lawyer should be logged into the system 3: Click on the subscription fee button and pay amount on lawyer profile 4: Click Service Listing button 5: Fill all the fields in the Service Listing form with valid details. 6: Click on the Add button.					
Test ID	Test Inputs	Expected Outputs	Actual Output	Result	Comment
LW001_01	All field are kept empty: 1.Name 2.Email 3.Contact Number 4.Address 5.Education 6.Experience 7.Language Known 8.Practice Courts 9.Specialization	Display “All field are required” on the form field	Display “All field are required” on the form field	Pass	

LW001_02	All input field are filled accurately: 1.Name 2.Email 3.Contact Number 4.Address 5.Education 6.Experience 7.Language Known 8.Practice Courts 9.Specialization	All the user details should be successfully validated and passed to the database. "Successfully Added" alert will be show up on the screen.	All the customer details should be successfully validated and passed to the database. "Successfully Added" alert will be show up on the screen.	Pass	Data successful inserted to the database.
LW001_03	abc@gmail.com	Email is successfully validated	Email is successfully validated	Pass	
LW001_04	abc.gmail.com	Display invalid massage “Email is invalid”	Display invalid massage “Email is invalid”	Pass	

#### 4.11 Support Center Management

Table 4. 11 - Legal Contact Test Cases

<b>Project ID: ITP_WD_B04_G09</b>										
<b>Project Name: Online Marketplace for Homestays</b>										
<b>Testing function:</b> Support Centre Management – Delete previous user										
<b>Test case ID: A001</b>	Test case designed by, ID: IT21240706 Name: Dhananjana B.K.T									
<b>Test Priority (High/Medium/Low)</b>	Medium									
<b>Test description:</b> Delete previous user.										
<b>Precondition:</b> Site owner must be logged into the system										
<b>Postcondition:</b> “Deleted Successfully” message should appear										
<b>Test Steps:</b> 2: Site owner should be logged into the system 3: Click on the Delete button available in the support centre dashboard 4: Select the record to be deleted 5: Click relevant record’s delete button 6: Check the previous record list after it got deleted.										
Test ID	Test Inputs	Expected Outputs	Actual Output	Result	Comments					
A001_01		If successfully executed previous user record and disappear popup message, click yes option and it should be deleted and “Successfully Deleted” message should be displayed	Disappear popup message, click yes option and it should be deleted and “Successfully Deleted” message should be displayed	Pass	Deleted button is working as expected. When clicked delete button that relevant record only gets deleted. Alert message is successfully displaying.					

## **Chapter 05 – Evaluation and Conclusion**

---

### **5.1 Evaluation**

#### **5.1.1 Assessment of the Project results**

Prior to designing the system, the group members study the functioning of house renting. Once the

requirements were identified the team came together to design the system. The system was designed to make it as user-friendly as possible to the users, so that no prior training will be needed to use the system. Certain functions proved to be more complicated than expected. As a result, a basic implementation of such functions was developed for the time being, and we hope to further develop the system in the future with the technology MERN [7].

The website provides an intuitive and visually appealing interface that allows users to easily navigate through rental listings, search for properties, vehicles, lawyers and find furniture showroom near rental house and perform other necessary actions. We gathered thorough and precise data regarding the rental properties that were readily available. This contains information about the area, cost, type of property, amenities, and excellent visuals.

#### **5.1.2 Lessons Learned**

- The importance of developing a complete product equipped with validations prior to release to the market.
- The power and importance of teamwork.
- This project taught us the importance of time management and proper planning to complete the work in time prior to the deadline.

### **5.2 Conclusion**

RENTMATE was created with the intention of increasing rental process efficiency and giving landlords, vehicle owners, attorneys, showroom owners, and tenants a satisfying experience. The project's goal was to develop an intuitive system that makes it easy for users to find suitable rental properties and streamline transactions. A limitation of the current system is that only authorized users, such as landlords and property managers, can manage tasks involving property listings, reservations, and

communications. These functionalities are not directly available to prospective tenants. The incorporation of an interface made especially for tenants is one suggested remedy for this restriction. They would be able to browse and search for rental homes, as well as send rental inquiries, all without involving staff. It is crucial to check that the interface gives tenants access to the appropriate tasks while preventing them from accessing administrative or management tasks.

## References

---

### References

- [1] [Online]. Available: <https://www.hindawi.com/journals/misy/2022/8179161/>.
- [2] A. Adeniran, "How to create pdf reports in react," [Online]. Available: <https://www.freecodecamp.org/news/how-to-create-pdf-reports-in-react/>.
- [3] "MongoDB Full workflow," [Online]. Available: <https://medium.com/@ashiqgiga07/working-with-mongodb-afee14557a92>.
- [4] [Online]. Available: <https://kinsta.com/knowledgebase/what-is-express-js/>.
- [5] [Online]. Available: <https://medium.com/@alfredodiling/react-from-zero-to-hero-9b5bb0de6025>.
- [6] [Online]. Available: <https://www.tutorialsteacher.com/nodejs>.
- [7] [Online]. Available: <https://blog.logrocket.com/mern-stack-tutorial/>.

## Appendix

---

	<b>Student ID and Name with Initials</b>	<b>Brief Description of the function</b>
01	IT21166488 – Maleesha K.L.D.D.S	<p><b><u>Listing Management System</u></b></p> <ul style="list-style-type: none"><li>• <b>Create</b> - Adding new listing to system.</li><li>• <b>Retrieve</b> - Display added listing on the dashboard page.</li><li>• <b>Update</b> – Update the previous listings as needed.</li><li>• <b>Delete</b> - Delete the listings from the system if the listing is not important.</li><li>• <b>Search</b> added listings on the dashboard.</li><li>• <b>Generate</b> personalized income report.</li></ul>
02	IT21183768 – Heshan W.A.S	<p><b><u>Payment Management System</u></b></p> <ul style="list-style-type: none"><li>• <b>Create</b> - Adding a Credit/Debit Card</li><li>• <b>Retrieve</b> - Displaying transection History, Display Card Details</li><li>• <b>Update</b> - Updating Card Details, Update Refund status.</li><li>• <b>Delete</b> - Delete Existing card Details.</li><li>• <b>Search</b> by using date Get transection Details.</li><li>• <b>Generate</b> Payment Details report.</li></ul>
03	IT21186424 – Bandara N.G.S.S.M	<p><b><u>Blog Management System</u></b></p> <ul style="list-style-type: none"><li>• <b>Create</b> – Adding new blog post to the system.</li><li>• <b>Retrieve</b> – Displaying blog post and comment section.</li><li>• <b>Update</b> - Updating new information to the blog post.</li><li>• <b>Delete</b> - Delete published blog post.</li><li>• <b>Search</b> - blog id and get blog details.</li><li>• <b>Generate</b> monthly reports on blog posts.</li></ul>

04	IT21233562 – Ranawaka W.E.I	<p><b><u>Customer</u></b></p> <ul style="list-style-type: none"> <li>• <b>Create</b> – Add new personal information.</li> <li>• <b>Retrieve</b> – Display user's booking details.</li> <li>• <b>Update</b> – Change property booking details, transport booking details.</li> <li>• <b>Delete</b> – Delete user account, cancel bookings.</li> <li>• <b>Search</b> users previous booking details.</li> <li>• <b>Generate</b> fully detailed report about bookings</li> </ul>
05	IT21184376 – Jayathilaka A.W.N.M	<p><b><u>Furniture Showroom Management System</u></b></p> <ul style="list-style-type: none"> <li>• <b>Create</b> - Adding new furniture showroom and furniture to the system.</li> <li>• <b>Retrieve</b> - Displaying showroom and furniture details.</li> <li>• <b>Update</b> - Updating showroom and furniture details.</li> <li>• <b>Delete</b> - Delete showroom and furniture.</li> <li>• Search for furniture id and get furniture details.</li> <li>• <b>Generate</b> report on all furniture of relevant showroom.</li> </ul>
06	IT21182846 – Chamaleen D.B.N	<p><b><u>Transport Management System</u></b></p> <ul style="list-style-type: none"> <li>• <b>Create</b> - adding new vehicles to the system.</li> <li>• <b>Retrieve</b> - displaying vehicle details.</li> <li>• <b>Update</b> - updating vehicle information.</li> <li>• <b>Delete</b> - removing vehicle information.</li> <li>• <b>Search</b> vehicle ID</li> <li>• <b>Print</b> Vehicle ID , name, location</li> </ul>
07	IT21197796 – Wasana K.H.I.M	<p><b><u>User Management System and Pending Listing Management System</u></b></p> <ul style="list-style-type: none"> <li>• <b>Create</b> - adding new user to the system.</li> <li>• <b>Retrieve</b> - displaying user details , display pending lists.</li> <li>• <b>Update</b> - updating user information.</li> <li>• <b>Delete</b> - removing violated users , remove pending lists.</li> <li>• <b>Searching</b> relevant role &amp; get users details.</li> <li>• <b>Generate</b> monthly reports about user details.</li> </ul>

08	IT21240706 – Dhananjana B.K.T	<p><b><u>Legal Contact and Support Center Management</u></b></p> <ul style="list-style-type: none"> <li>• <b><u>Create</u></b> -adding new lawyer details to the system, adding new problems.</li> <li>• <b><u>Retrieve</u></b> - displaying lawyer details and profiles, display problems and user details.</li> <li>• <b><u>Update</u></b> - updating the lawyer details , updating the solution status.</li> <li>• <b><u>Delete</u></b> - deleting the lawyer , deleting previous user details.</li> <li>• <b><u>Search</u></b> user details and problem details through the any details.</li> <li>• <b><u>Generate</u></b> a report user details and problem details</li> </ul>