

# DAFFODIL INTERNATIONAL UNIVERSITY



**DHAKA, BANGLADESH**

**January, 2022**

## **Development of a Web Application for House Rent Named “Here is your Home”**

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of  
Science in Computer Science and Engineering

**Supervised By**

**Taslima Ferdaus Shuva**

Sr. Lecturer  
Department of CSE  
Daffodil International University

**Submitted By:**

**Tahmid Ahmed**

ID:171-15-884

**Taufiq Rahman**

ID: 163-15-802

## APPROVAL

This Project/internship titled titled “Development of a Web Application for House Rent Named Here is your Home”, submitted by Tahmid Ahmed, ID No: 171-15-884 and Taufiq Rahman, ID No: 163-15-802 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 18.09.2021.

### BOARD OF EXAMINERS

*Tania Khatun*

---

**Tania Khatun**

**Senior Lecturer**

Department of Computer Science and Engineering  
Daffodil International University

**Internal Examiner**

*Jahangir*

---

**Mohammad Jahangir Alam**

**Lecturer**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Internal Examiner**

*Farid*

---

**Dr. Dewan Md. Farid**

**Associate Professor**

Department of Computer Science & Engineering  
United International University, Bangladesh

**External Examiner**

## DECLARATION

We thus declare that we completed this project under the supervision of Taslima Ferdaus Shuva, Senior Lecturer, Department of CSE Daffodil International University. We further affirm that neither this project nor any portion of this project has been submitted to any other institution for the granting of a degree or certificate.

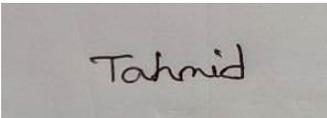
### Supervised by:



---

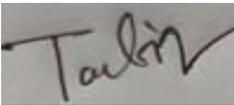
**Taslima Ferdaus  
Shuva**  
Senior Lecturer  
Department of CSE  
Daffodil International University

### Submitted by:



---

**Tahmid Ahmed**  
ID: 171-15-884  
Department of CSE  
Daffodil International University



---

**Taufiq Rahman**  
ID: 163-15-802  
Department of CSE  
Daffodil International University

## ACKNOWLEDGEMENT

First and foremost, we offer our heartfelt appreciation and gratitude to Almighty God for His divine gift, which has enabled us to successfully finish the final year project/internship.

We are really grateful to Taslima Ferdaus Shuva, Senior Lecturer, Department of CSE, Daffodil International University, Dhaka. To complete this project, our supervisor must have extensive knowledge and a deep interest in the subject of "Web Application Development." Her unending tolerance and intellectual advice helped us to complete this project.

Her persistent support, frequent and enthusiastic monitoring, constructive criticism, helpful advice, reading numerous substandard versions and fixing them at all stages allowed us to finish this project.

We would like to offer our heartfelt thanks to the Head, Department of CSE, for his generous assistance in completing our project, as well as to the other faculty members and employees of Daffodil International University's CSE department.

We would like to thank everyone of our Daffodil International University classmates who participated in this conversation while completing their course work.

Finally, we must express our gratitude for our parents' unwavering support and patience.

## **ABSTRACT**

We live in a technologically advanced era. Every day, more and more chores are becoming digitized. As a result, the use of web apps is expanding on a daily basis. Every firm, institution, and even restaurant now has its own online application for task management. However, the majority of the applications are paid, and small businesses and organizations may not be able to afford to utilize them due to financial constraints. One of them is house rental applications, the majority of which are available online and charge a fee.

These programs frequently overlook critical functionality that a website need. As a result, we propose a web tool that can assist with property rentals without incurring any costs. Apart from that, open-source projects are becoming more popular since developers may improve the application's reliability and strength. As a result, we created this house-renting application, which will have all of the functions that a house-renting application may require. As a result, both the owner and the renter profit from this application with minimal effort. A chatbot and an AI-based cost manager with a search engine will be implemented in the future. As a result, both stakeholders may obtain critical information via conversing or utilizing search engines.

## TABLE OF CONTENTS

<b>CONTENTS</b>	<b>PAGE</b>
Board of examiners	i
Declaration	ii
Acknowledgement	iii
Abstract	iv
<b>CHAPTER</b>	
<b>CHAPTER 1: INTRODUCTION</b>	<b>1</b>
1.1 Introduction	1
1.2 Motivation	2
1.3 Objectives	2
1.4 Expected outcome	3
1.5 Report layout	3
<b>CHAPTER 2: BACKGROUND</b>	<b>4</b>
2.1 Introduction	4
2.2 Related works	4
2.3 Comparative studies	7
2.4 Scope of the problem	8
2.5 Challenges	9
<b>CHAPTER 3: Technology &amp; REQUIREMENT SPECIFICATION</b>	<b>9</b>
3.1 Technology Used	10
3.2 Requirement collection and analysis	11
3.3 Use case modeling and description	12
3.3.1 The model	12
3.3.2 Description of case	13
3.4 Entity relationship diagram	17
3.5 UML class diagram	18
3.6 Design requirements	19

<b>CONTENTS</b>	<b>PAGE</b>
<b>CHAPTER 4: DESIGN SPECIFICATION</b>	<b>20</b>
4.1 Front-end design	20
4.2 Back-end design	29
4.3 Implementation requirements	31
<b>CHAPTER 5: IMPLEMENTATION AND TESTING</b>	<b>32</b>
5.1 Implementation of database	32
5.2 Implementation of front-end design	33
5.3 Testing implementation	38
5.3.1 Unit testing	38
5.3.2 Integration testing	38
5.3.3 Functional testing	38
5.3.4 System testing	38
5.3.5 Usability testing	39
5.3.6 Test result and report	
5.3.7 Result of unit testing	39
5.3.8 Result of integration testing	39
5.3.9 Result of functional testing	39
5.3.10 ult of system testing	Res 39
<b>CHAPTER 6: CONCLUSION AND FUTURE SCOPE</b>	<b>40</b>
6.1 Discussion and conclusion	40
6.2 Scope for further development	41
<b>REFERENCES</b>	<b>42</b>
<b>APPENDIX A</b>	<b>43</b>

## LIST OF FIGURES

<b>FIGURES</b>	<b>PAGE NO</b>
Figure 2.1: Reference site 1	5
Figure 2.2: Reference site 2	6
Figure 3.1: Use case diagram	12
Figure 3.2: E-R diagram	17
Figure 3.3: UML class diagram	18
Figure 4.1: User login	20
Figure 4.2: New user registration	21
Figure 4.3: User profile information	22
Figure 4.4: Creating post I	23
Figure 4.5: Creating post II	24
Figure 4.6: Post histories	25
Figure 4.7: Search page	26
Figure 4.8: Recent posts	27
Figure 4.9: Explore	28
Figure 4.10: Database relationship Table	29
Figure 4.11: Database Table	30

## LIST OF FIGURES

<b>FIGURES</b>	<b>PAGE NO</b>
Figure 5.1: Implementation of database	32
Figure 5.2: Initialize front end of search page	33
Figure 5.3: CSS and Bootstrap Login page	34
Figure 5.4: Implementation of CSS and bootstrap on profile page	35
Figure 5.5: CSS and bootstrap implementation Signup page	36
Figure 5.6: CSS and bootstrap on Login page	37

## LIST OF TABLES

<b>TABLES</b>	<b>PAGE NO</b>
Table 3.1: Use case of login	13
Table 3.2: Use case of Post creation	14
Table 3.3: Use case of Searching	15
Table 3.4: Use case of Account creation	15
Table 3.5: Use case of Admin	16
Table 3.6: Use case of Recent Post	16

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

We currently live in a world that is technologically sophisticated. To satisfy our daily needs, we rely on technology. As a result, the internet has become the most important technological factor. Because the cloud is now connected to all types of technology. As a result, many of our regular tasks are now performed over the internet. Many enterprises and organizations, especially tiny ones, today operate their operations using web-based technologies. As a consequence, we're working on something linked to online technology that can make people's lives simpler.

There are several house-renting websites that cater to a range of activities. There are also a variety of additional sites that can be located all around the world. So far, I've spoken about the rental system, which is frequently used to keep track of their actions. However, we can observe that the free renting approach used by a website is insufficient. As a result, we've taken steps to develop a system that may be utilized in a range of settings.

In our renting system application, we attempted to solve all of the main components of housing leasing. Our solution is more than just a rental system, which is exciting. This is a completely functional and integrated website that the rental system can manage. On the other side, we've integrated a fully working Google Map API, which allows users to swiftly and easily locate properties, making the system more user-friendly. A login mechanism is also available, allowing users to safeguard their accounts and ensuring that all users are secured. As a result, our major objective is to make house leasing a technology-driven, easy-to-manage operation. As a consequence, many of the houses will be able to rent their residences without issue in the future. They may, however, restructure all of the components to fit their needs because it is an open-source project.

## 1.2 Motivation

We explored for a better approach to rent a property in Bangladesh all over the internet. Then we learned that there are hardly no apps that can help you find a free rental property. There are a number of websites that provide online rental services, however they are all fee-based. We learned that a web application management system is critical for keeping the property rented as a result of our work on the site. As a result, all online resources have a fee associated with them.

We then decide to build an open source platform that anybody can use and that all developers can contribute to. As a result, we decided to design such a web application in order to make the life of a homeowner better and easier. We've tried to make the installation of this application as simple as possible so that even non-techies may use it. On the other hand, we provide open source apps that enable clients to do business on their own private network. I'm hoping that this will make it easier for all of the houses to do their assignments.

## 1.3 Objectives

So we have decided to make our web application about house renting. Then there are some objectives that we need to fulfill. Here are some of the targets of objectives that can be gained by our house renting system:

- To make it easier to keep track of who wants home and for an administrative management system.
- Creating a user-friendly interface.
- To make the process of renting a home easier for customers
- Dashboard elements that are well-designed and engaging.
- To convert the laborious procedure of renting a home to an automated and online system.
- Finding the resources and labor required to manage a rental site.
- User emails are used to validate the house renting system.
- Attempting to include all forms of tasks that a house-renting website can accomplish.

## 1.4 Expected Outcome

This house-renting technique will provide results. Here's what we'll talk about:

- **Improve Business Processes:** To be able to use internet technology to project the rental company's services to the rest of the world rather than just their local domain, resulting in a higher return on investment.
- **Online House Renting:** Rather of traveling around looking for a rental property, consumers may use this service to reserve available rooms, houses, and apartments online ahead of time.
- **Customer registration:** Instead of the ineffective existing systems in which only administrators have access to customer information, a registration portal would store customer information, monitor transactions, and use the data to provide better and improved services to customers, as well as user accounts where they could view their information.
- **Users may check and discover information about nearby available homes for rent or sale, as well as the current economic design of houses and flats, using the Rentals Notice and Blog feature.**

## 1.5 Report Layout

This is the report format that was used to create this report on a web application. Here's where you can stay:

1. All of the subjects are discussed in detail with relevant information.
2. All themes are separated into paragraphs to make it easier to grasp..
3. Figures and graphics are used to help readers understand the topic's substance.
4. Tables are displayed to compare data from one source to another.
5. The required typeface and size are used to organize the report's contents.
6. To format the information, certain margins and spaces are utilized.
7. Bullet points are used to more precisely explain the content value.
8. The required structure and menus are utilized to make it easier to access all of the material.

## **2.1 Introduction**

A management system can control the track of a critical component of a business or organization. On the other hand, It makes the life of renter's and those who are searching for a house easier and more comfortable. So everything that hasn't been placed into a management system is what we're searching for. Then we discovered that a slew of new house-renting websites are springing up every day. Their task, on the other hand, is badly executed. As a consequence, we notice that each site has a significant amount of work to be done. As a result, we believe they will require a rental system. However, we observed that several of the rental sites need payment in order to be used. There are, however, a number of limitations, and they aren't well-defined. As a result, we've made the decision to construct one. As a result, we devised the notion of house rentals. I hope this information is useful to all users.

## **2.2 Related Works**

There are various renting schemes that are equivalent, such as ours. In our system, we have attempted to accomplish something better and distinctive. Some rental systems are available on the internet, however the majority of them do not provide a free version. You must spend a certain amount of money to purchase and maintain them, and they have some limits in terms of functions. Some systems are well-designed but have limited functionality, while others have amazing capabilities but need monthly membership costs. That is why we have attempted to provide something unique to them. Below are some lists of house-renting services, along with their contact information.

Here is a list of various existing house-renting systems, along with a description of their online benefits and disadvantages.

Bangladesh বাংলা Site settings Favourite properties Saved searches Log in

bproperty Buy Rent Add Property Blog Guides New Projects Services Trends

PURPOSE: Buy LOCATION: PROPERTY TYPE: Residential PRICE (BDT): 0 to Any  
 BEDS: All AREA (SQFT): 0 to Any BATHS: All KEYWORD: ADVANCED FILTERS: Any

Residential Properties for Sale in Bangladesh

Dhaka (9,130) Chattogram (1,138) Narayanganj City (698) Cumilla (355)

VIEW ALL LOCATIONS

Completion Status: All Ready Under Construction Popular Active listings

Residential Properties For Rent

**BDT 8,274,800**  
 Sector 10, Uttara, Dhaka  
 Apartment  
 A Beautiful 1306 Sq Ft Apartment Is ...  
 3 3 1.306 sqft  
 CALL EMAIL

**BDT 7,886,200**  
 Sector 10, Uttara, Dhaka  
 Apartment

Windows Taskbar: Type here to search 12:43 PM 9/12/2021

Fig 2.1: Bproperties Home Page[1]

This system features a superb UI/UX design, no installation, no programming skills necessary to maintain it, event management, program administration, and an international payment option, all of which are highly beneficial. However, the Monthly Subscription procedure is featured here and is not open source, so you will have limits with their services. It also lacks a routing system.

This system also provides excellent features such as email confirmation, handling, member control, and cost management, among others; nevertheless, certain capabilities are restricted..

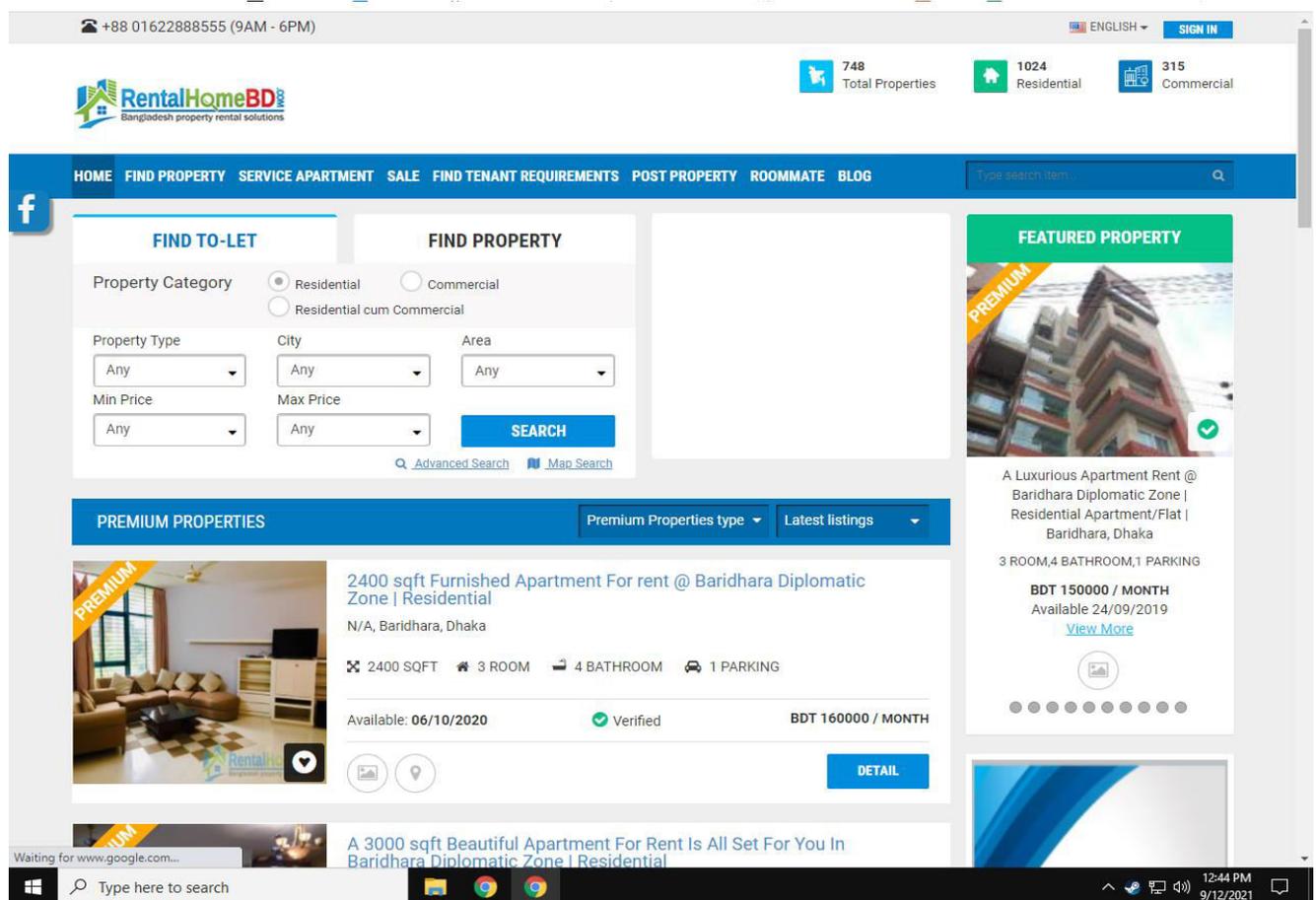


Fig 2.2: Rental Home BD's Home page

## **2.3 Comparative Studies**

As we can see, some rental systems are similar to ours, yet there are evident distinctions between them and us. First and foremost, because it is an open source project for house rental, our project includes certain unique features that match client needs, as well as the ability for clients to add their own personal features to the system. Because it is an open source project, anybody may use it without fear of infringement and modify or add functionality. Furthermore, unlike other services, they do not require any monthly payments. In the future, we'll be adding some really cool features.

Routing and phone number confirmation with an OTP are examples of systems. As a result, we can guarantee our system's customer pleasure. Also, because it is open source, they may add new features, which makes our product unique and valuable for customers who have any form of rental property and wish to manage it digitally.

## 2.4 Scope of the Problem

This project includes a variety of fields, from management theories to computer technology, and it required the completion of several research in order to satisfy the project's objectives. The following subjects are addressed:

- The proposed system includes an online rental house search and account management component.
- Customers would no longer have to go through the time-consuming procedure of renting a property.
- General clients will be able to make efficient use of the system.
- With the exception of small server failures, the term "web-platform" suggests that the system will be available 24 hours a day, seven days a week.
- Existing Systems: This comprises reviewing existing systems and finding defects with the purpose of developing a new system to handle the challenges that the local and global domains face when dealing with house rental problems.
- Will locate consumers a suitable house at the proper location.

## 2.5 Challenges

When there are many advantages of a system there is also many challenges with the system:

- Establish User-Friendly Usability: Creating a user-friendly system is difficult, yet it is the most crucial aspect of a system. Why would a user use a system if they don't understand how to use it? That is why we have attempted to make the house rental system as basic and as easy to use as possible for the user.
- Creating a link between all of the tables in a database: Although creating database tables is a difficult task, For our project, we tried our best to establish normalized database tables. That is why we have kept so much information for users.
- The procedure of deleting users is an important aspect of the house-renting system. We have a function to keep track of users; if the site's administrator wishes to remove a member, they can do it with ease using a simple approach.
- Front-end processing: This system includes an integrated website, and maintaining a website with a backend dashboard is a demanding task. For a better user experience, we've kept everything running smoothly.
- User file encryption: Every submitted file has a unique id, therefore if someone uploads a file with the same name, the fill will be visible and there will be no collision. It takes a lot of effort to develop.

## Chapter 3

### Survey of Requirement & Technology

#### 3.1 Technology Used:

- HTML: HTML code ensure that the text and visuals in your browser are formatted correctly. A browser would be unable to display text as elements or load images or other components if HTML was not present.
- PHP: PHP is a general-purpose programming language that is specifically tailored to server-side web development, which involves running PHP on a web server.
- CSS: CSS is a stylesheet language that is used to determine how web pages look. It describes how a markup language-created document should look and be formatted. It extends HTML's capabilities.
- MySQL is a computer database management system (RDBMS) that runs as a server and allows many users to access a variety of databases.
- VS Code: It is the IDE we used
- Apache Server: Before delivering webpages, this project uses the Apache server to interpret and execute PHP pages.

### **3.2 Requirement Collection and Analysis**

The first step is to figure out which technology is the most extensively utilized throughout the world. Then we became aware of all the web-based technologies available. Then we searched for subjects or items that people need but can't get their hands on. We learned that the majority of firms and organizations utilize commercial software when we started hunting for it. Then we realized that websites may be made for any cause at any time. Purchasing software for them, on the other hand, may be a considerably more expensive choice. And the majority of users are unable to locate the information they want within the software.

As a result, our product has been labeled as a house-renting website. As a result, we looked into a few things and chores that would need us to handle the majority of the property rental ourselves. Then we construct a list of numerous needs and determine the most typical house renting requirements. As a consequence, the menu that must be included in the project is identified. After that, we experimented with the technology that the majority of people were utilizing. As a consequence, this application can accommodate the hands of the vast majority of individuals. As a consequence, we choose all of our web-based technologies with care. Then we looked for a file structure that users might comprehend for location development. The file structure is discovered as a consequence. After that, we looked for the most extensively utilized backend technology. Then we came upon php. We opted to construct this form of web-based application as a consequence of our study and analysis.

### 3.3 Use Case Modeling and Description

#### 3.3.1 The Model:

Use case diagrams exhibit every one of the system's users and then explain how they can interact with it. Figure 3.2 shows the here is your house use case diagram with task relationships:

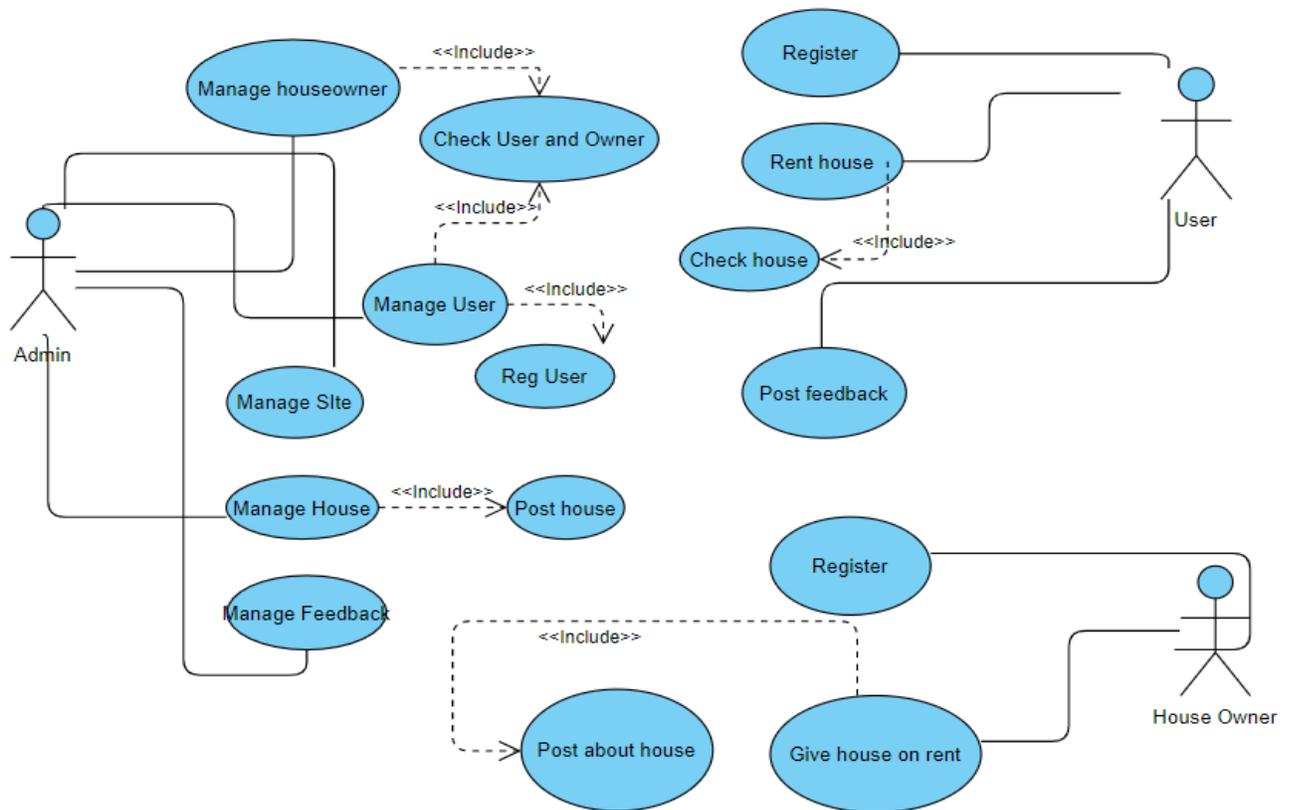


Figure 3.1: Use Case Diagram

### 3.3.2 Description of use case:

There are three actors in this use case illustration, as we can see from the description. They are the administrator, the owner of the residence, and the user. In this scenario, the consumer would be the primary actor. All acts that are carried out in accordance with the user's wishes. There's also an admin page where you can manage all of the users and the application's activities. Now, if we look at the user actor, we can see that he can login, search, check recent postings, leave feedback, and rent a property if it is suitable. As a result, users can carry out this activity in accordance with the use case diagram. Now, if we look at the admin area, we can see that the admin is in charge of all user activity. Let's take a look at all of the use case table models to learn more about this.

This figure 3.1 of use case will tell about how the login process of the system works.

**Table 3.1: Use Case of Login**

<b>Cases</b>	<b>Detail s</b>
Actors	Admin, user, house owner
Pre-Condition	nothing
Post-Conditions	After successful login actor will go to the Desire dashboard.
Basic Flows	Actors will use their username and password to log in. After that, a successful login will take them to their dashboard, where they may continue working..

This table 3.2 will tell about the post creation use case.

**Table 3.2: Use Case of Post creation**

<b>Cases</b>	<b>Details</b>
Actors	Admin, user
Pre-Condition	<ul style="list-style-type: none"> <li>- Please log in to the system.</li> <li>- Go to the user profile page.</li> </ul>
Post-Conditions	<ul style="list-style-type: none"> <li>-The database will be controlled by the administrator.</li> <li>-Users will create their own posts after registering.</li> <li>-Users have the ability to post advertisements..</li> </ul>
Basic Flows	<ul style="list-style-type: none"> <li>-Administrator has access to all postings and has the ability to edit, change, or remove them from the database.</li> <li>-Administrators can use a database to post.</li> <li>-Users can choose a location or let the device decide.</li> <li>-Users may also upload photos and leave comments on their postings.</li> <li>-Users can only post when they have completed all of the required fields.</li> </ul>

This table 3.3 will now show you the use case for a blog. Which will be used to publish and remark on something.

**Table 3.3: Use Case of searching**

<b>Cases</b>	<b>Details</b>
Actors	user
Pre-Condition	-Log into the system
Post-Conditions	-Administrators have the ability to create and delete posts. -Users have the ability to search.
Basic Flows	-The user will search for what he is looking for. He can see posts that match his search criteria.

This following table 3.4 will show about creating a account use case which will be used to create account:

**Table 3.4: Use Case Of Account creation**

<b>Cases</b>	<b>Details</b>
Actors	User,house owner
Pre-Condition	-Nothing
Post-Conditions	Actors can then access the application
Basic Flows	All users are given a email and a one-time password (OTP). They can then make a post or look for one.

This table 3.5 will show about the admin use case:

**Table 3.5: Use Case Of Admin**

<b>Cases</b>	<b>Details</b>
Actors	Admin
Pre-Condition	-Have access to database
Post-Conditions	-Admin can make a notice to users.
Basic Flows	-The application will be managed by an administrator. -He has the ability to create, edit, and remove posts. -He also has access to user and house owner information.

This table 3.6 will show the recent posts

**Table 3.6: Use Case Of recent posts**

<b>Case s</b>	<b>Detail s</b>
Actors	user
Pre-Condition	-Login to the system
Post-Conditions	-Can look at the house's information on the post
Basic Flows	-After logging in, the user gets to this page. -From there, the user may look at the postings and see the information.

### 3.4 Entity Relationship Diagram

The E-R diagram for this Renting system is shown in Figure 3.3.

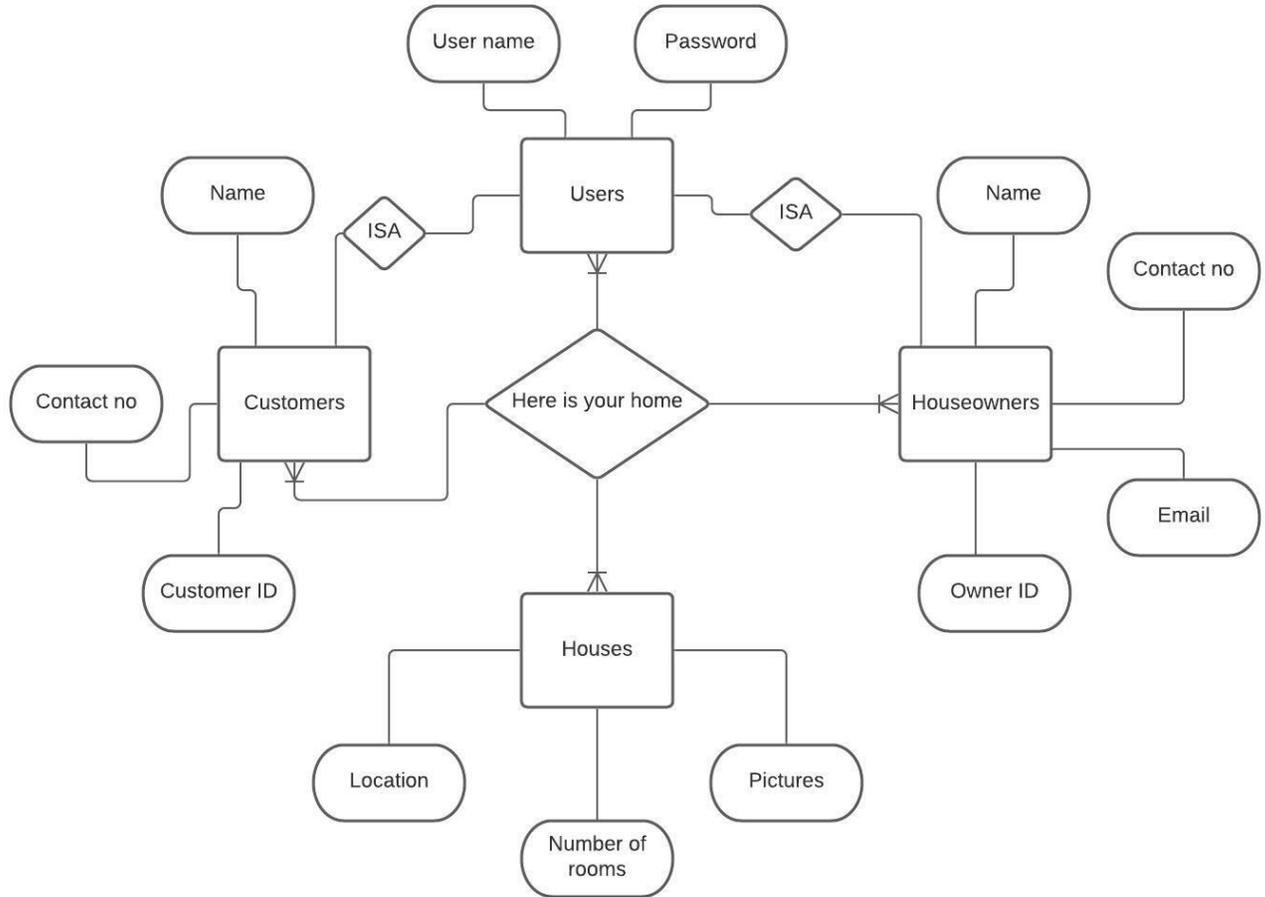


Figure 3.2: E-R Diagram

### 3.5 UML Class Diagram

The uml class diagram is shown in Figure 3.4:

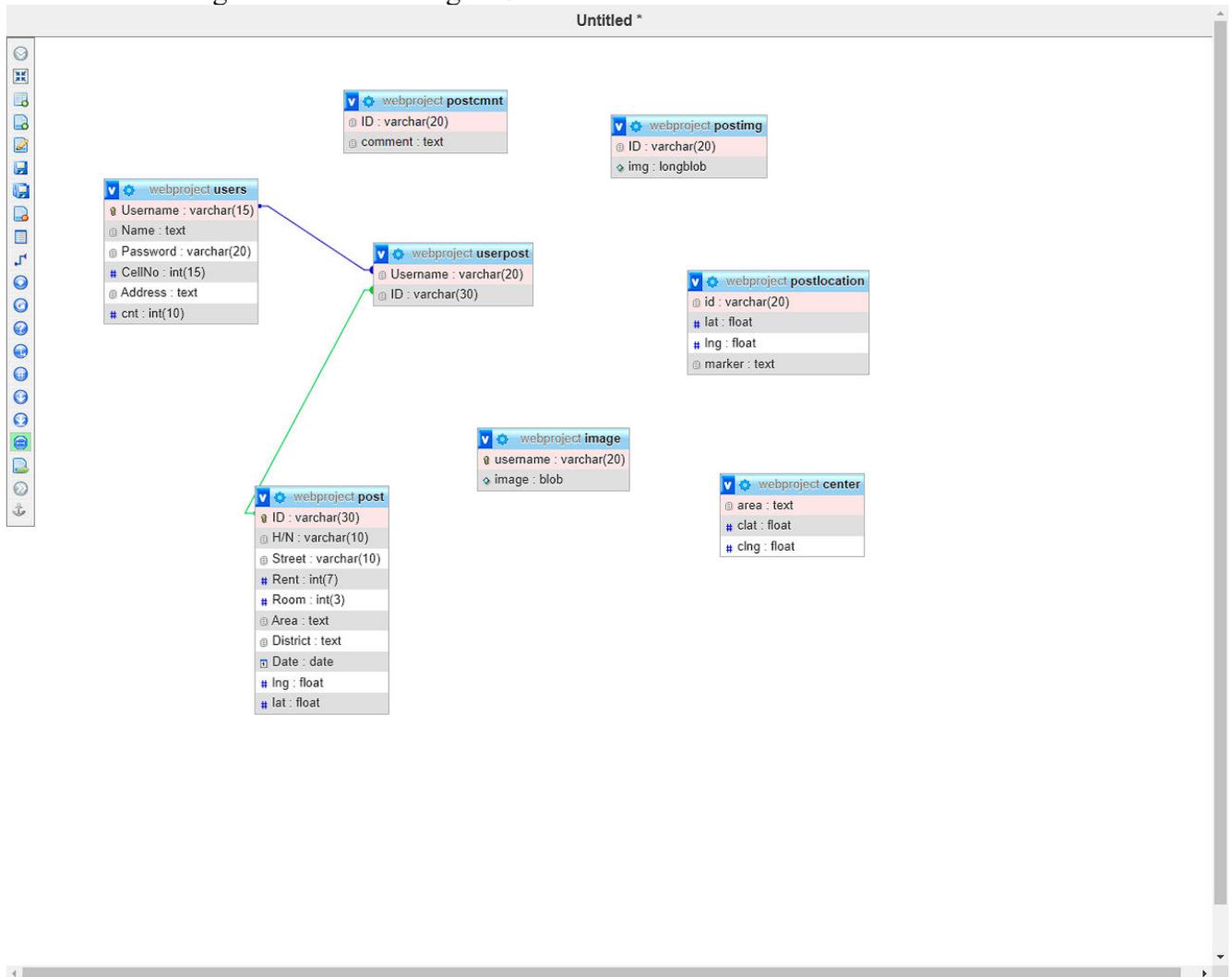


Figure 3.3: UML Class Diagram

### **3.6 Design Requirements**

Some designs are necessary to move on with this project in order to make the management system more user-friendly and dependable. There are two different sorts of designs for this, one for the frontend and one for the backend.

We must first draw out the layouts on paper before beginning the frontend design. Then, in order to construct tidy code, we must identify the appropriate columns and rows. The frontend coding must next incorporate bootstrap, bootstrap js, and some jquery to complete the design. The project must then be designed in compliance with the documentation requirements, and the coding must be completed.

In order to create the backend, we now need to understand database structure and relational databases. It's also necessary to grasp the links between the database's tables, as well as to consider redundancy. As a result, some form of normalization is essential to avoid this. We also tried to keep to the object-oriented coding approach when it came to backend work. Then check for the file structure of the source repository. As a consequence, we've completed all of the backend design needs. As a result, that was the project's design requirement.

## CHAPTER 4

### DESIGN SPECIFICATION

#### 4.1 Front-end Design

As it is a web-based House rental management-System, we have several user interfaces and other amazing capabilities which we've explained in earlier sections. Now we'll show you some of the front-end design of our system:

Figure 4.1 depicts the GUI user LOGIN system, which will take you to the Home page of our online application.

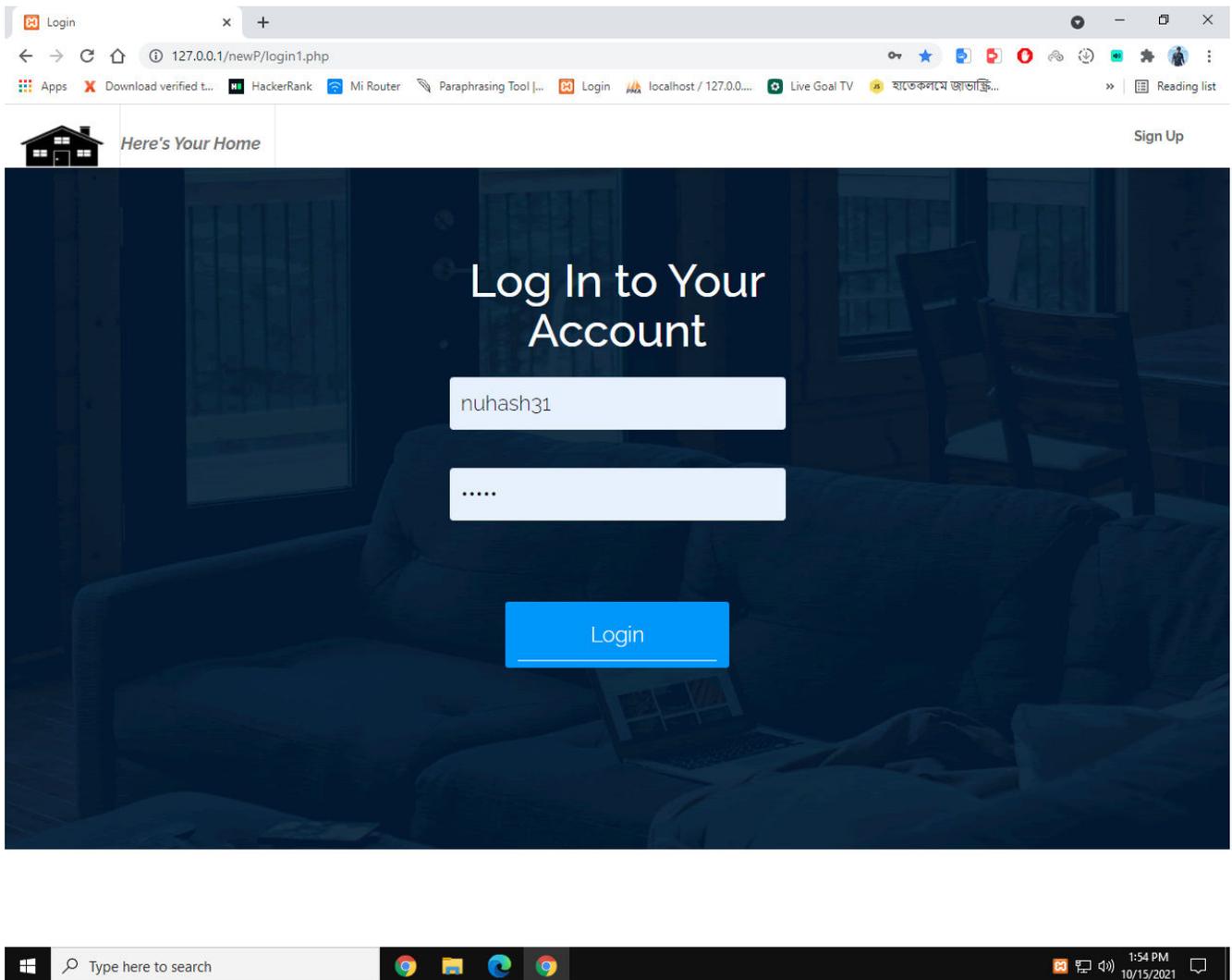


Figure 4.1: User Login Login

The graphical user interface (GUI) for new user registration is shown in Figure 4.2, and it contains all of the necessary information to register a new user.

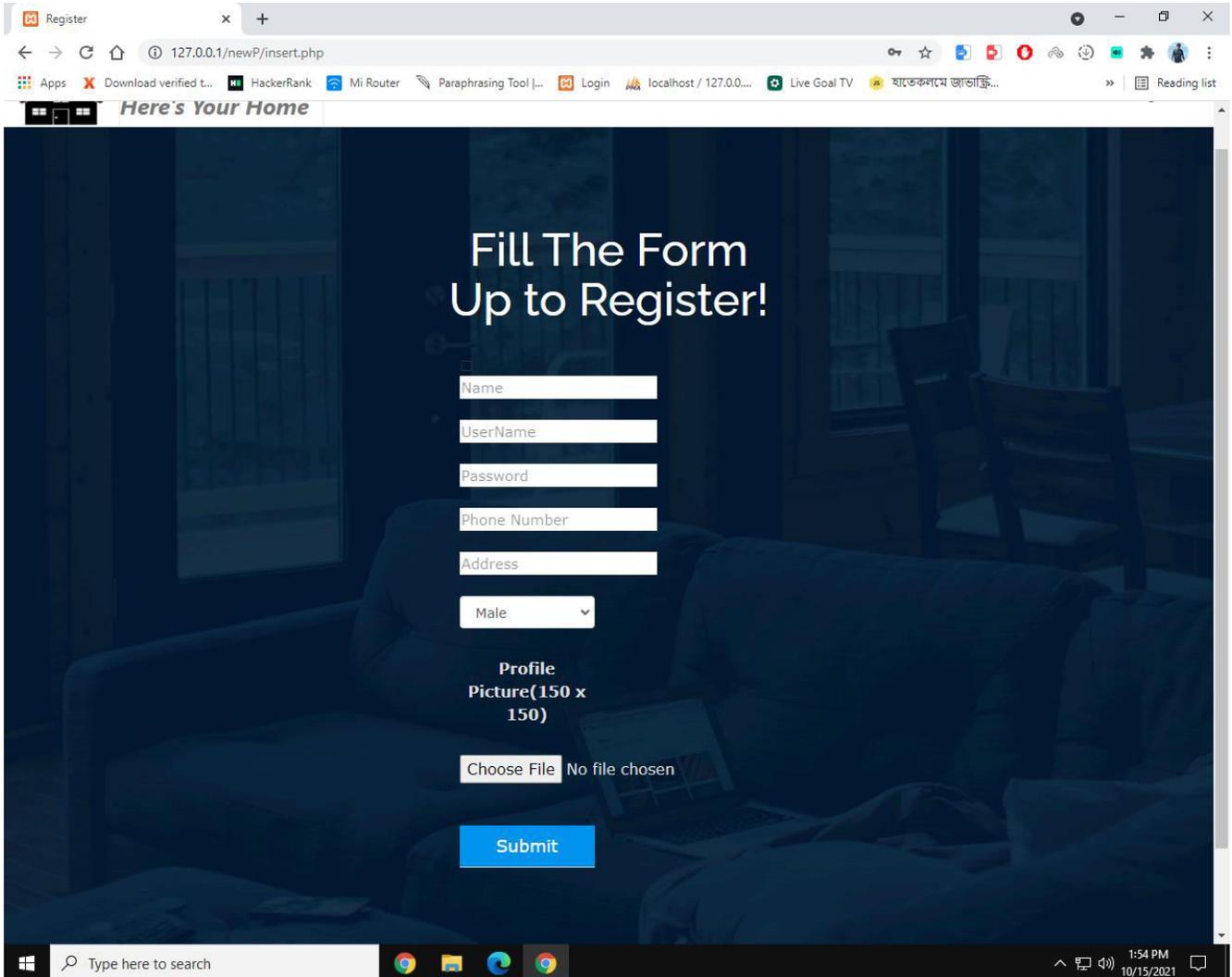


Figure 4.2: New user registration

The user's profile GUI is shown in the following figure 4.3. Users' information will be shown on this page.

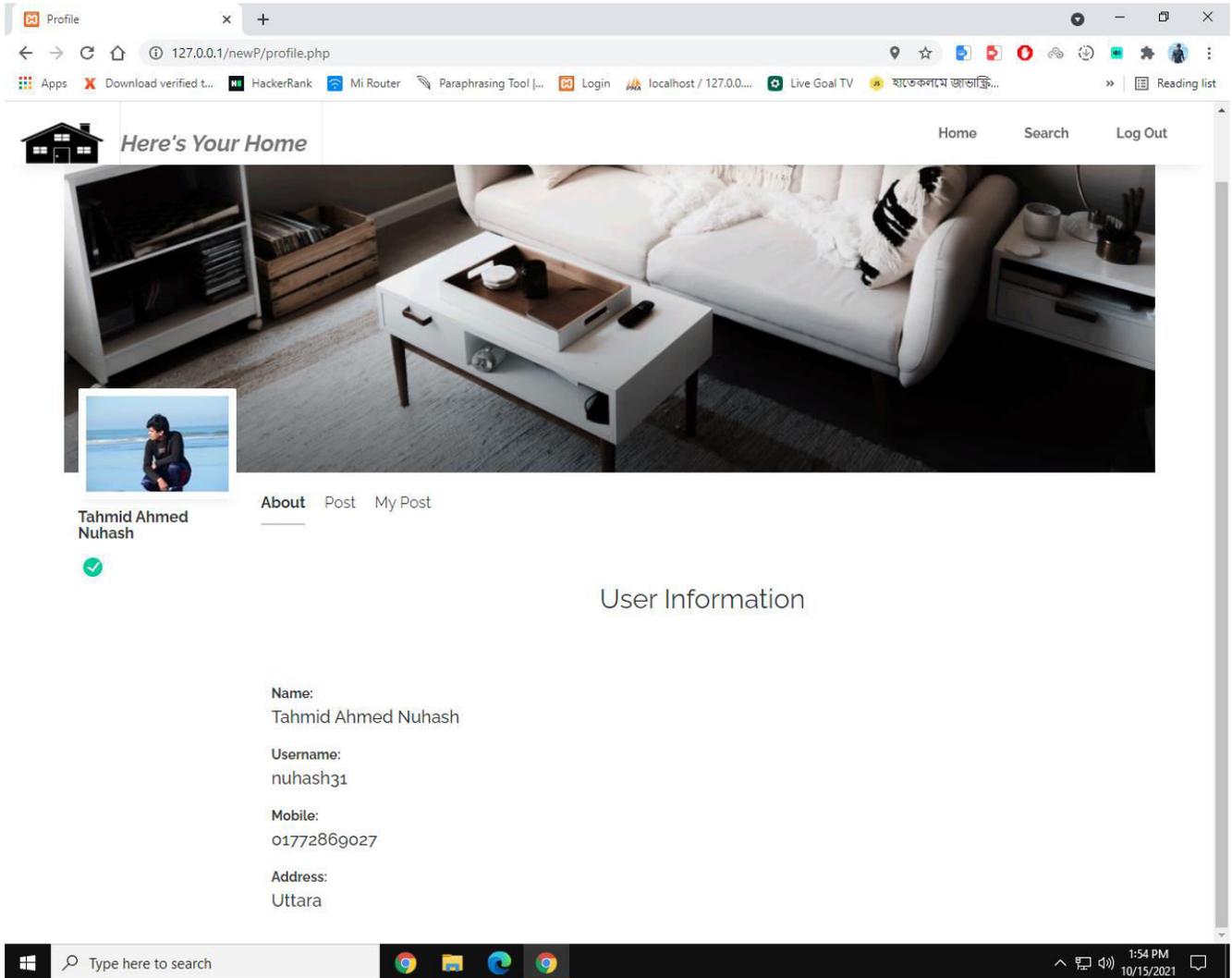


Figure 4.3: User profile information

The GUI for renting a property is shown in Figures 4.4 and 4.5 once it has been created. Users may place pins on the map to denote their whereabouts. Otherwise, the device on which the post is generated will be used as the house location on the map.

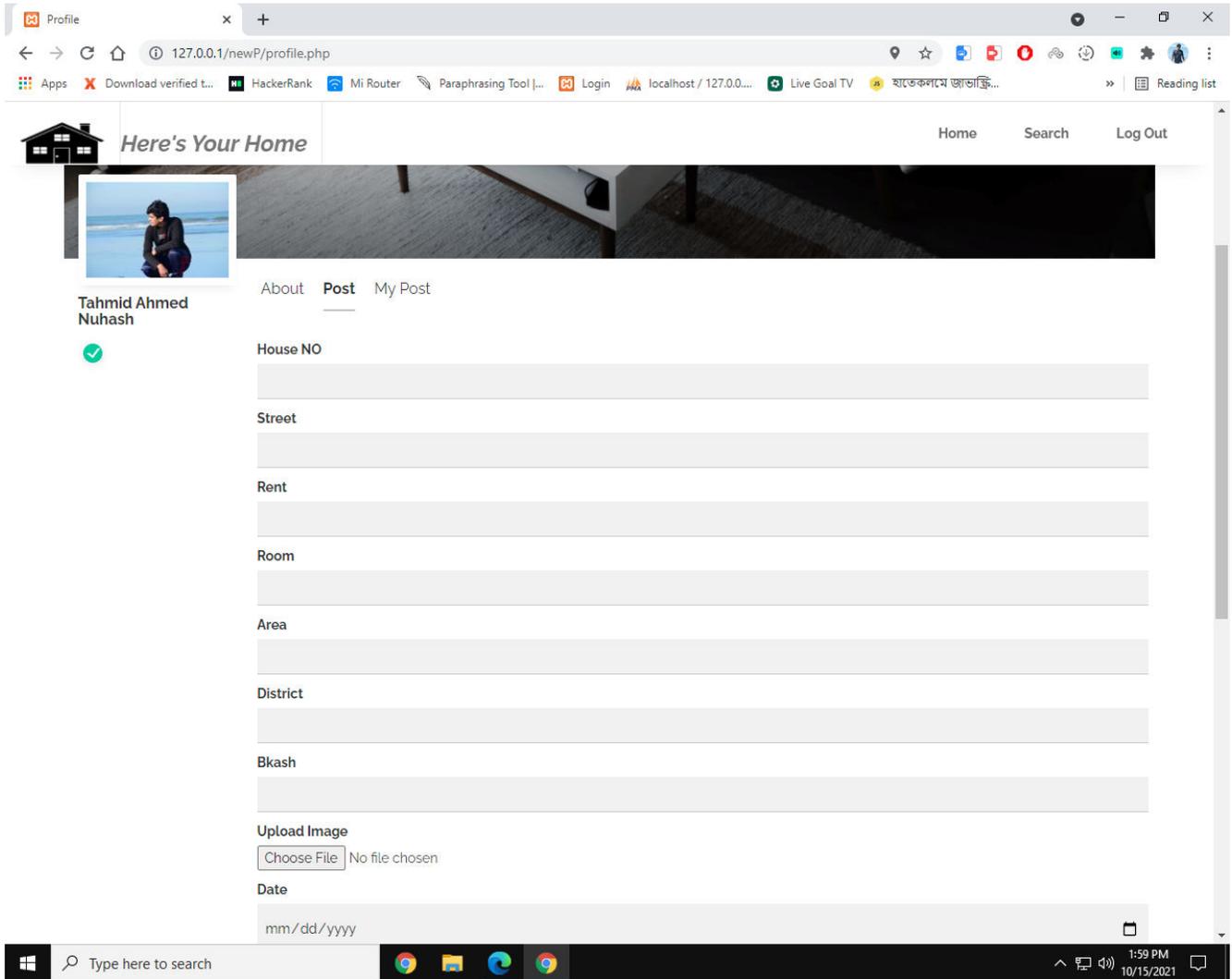


Figure 4.4: Creating a post

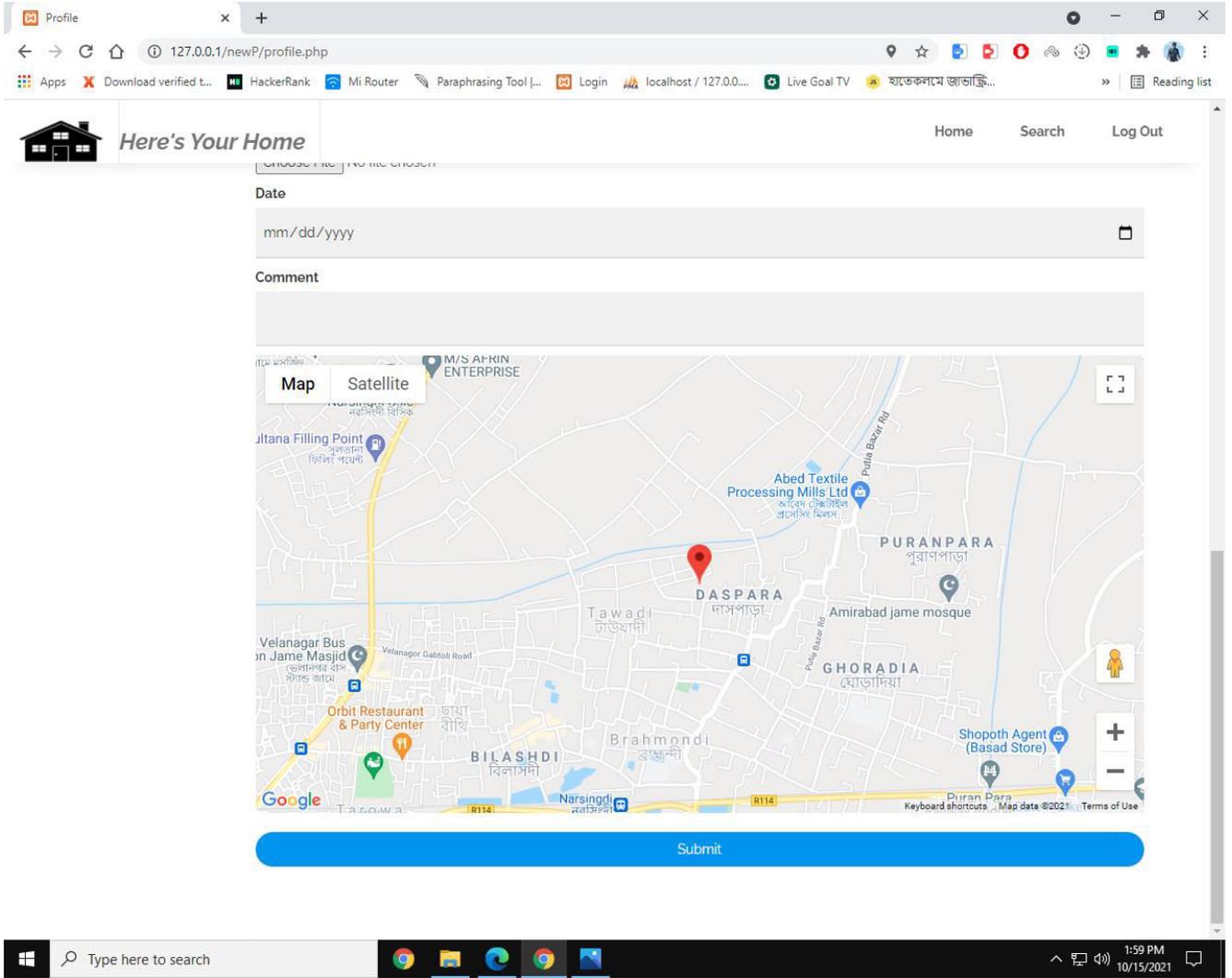


Figure 4.5: creating a post

The GUI prior history of generated posts is shown in Figure 4.6. The account owner can remove his posts from here.

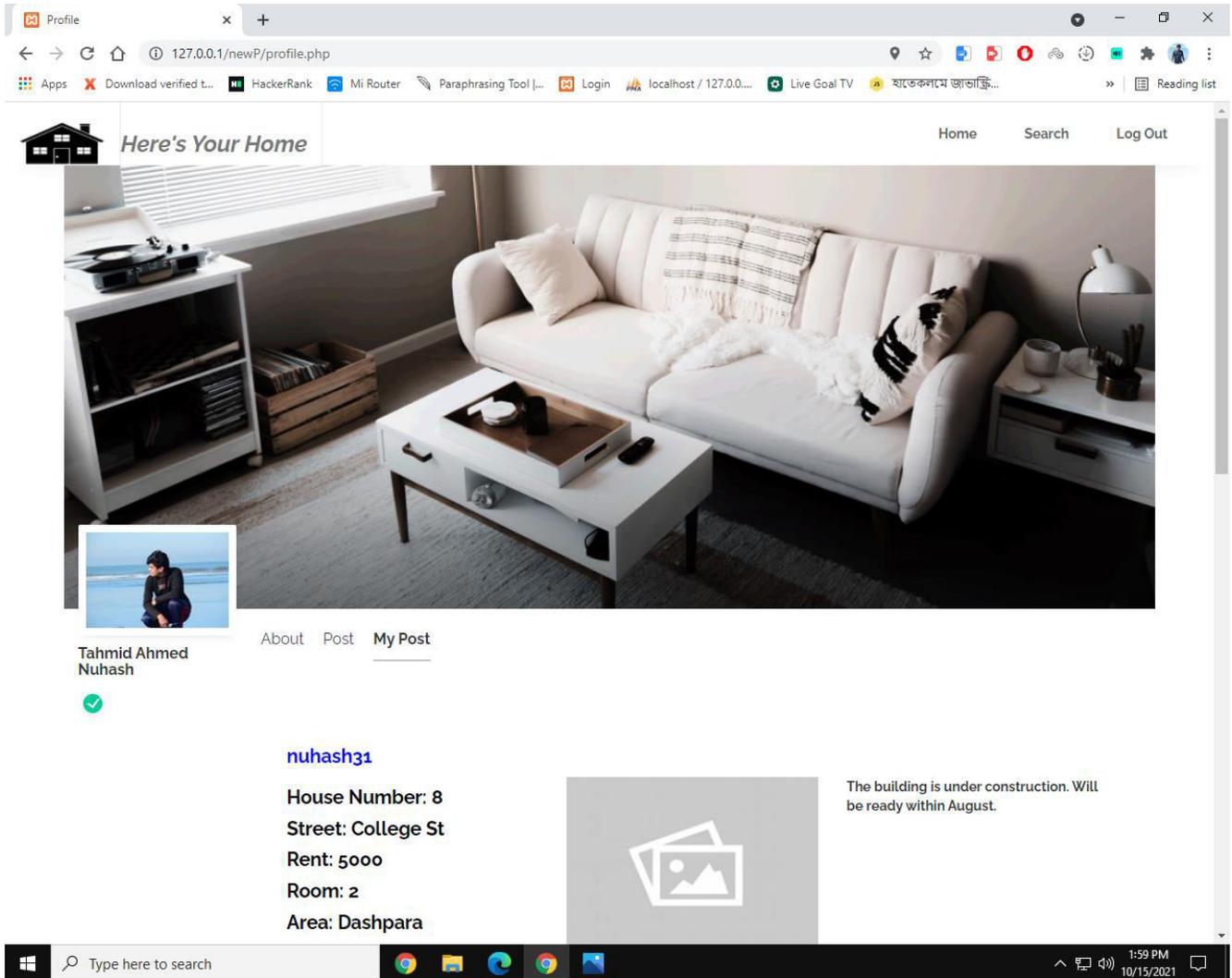


Figure 4.6: Post histories

The GUI for looking for a house based on the user's specifications is shown in Figure 4.7.

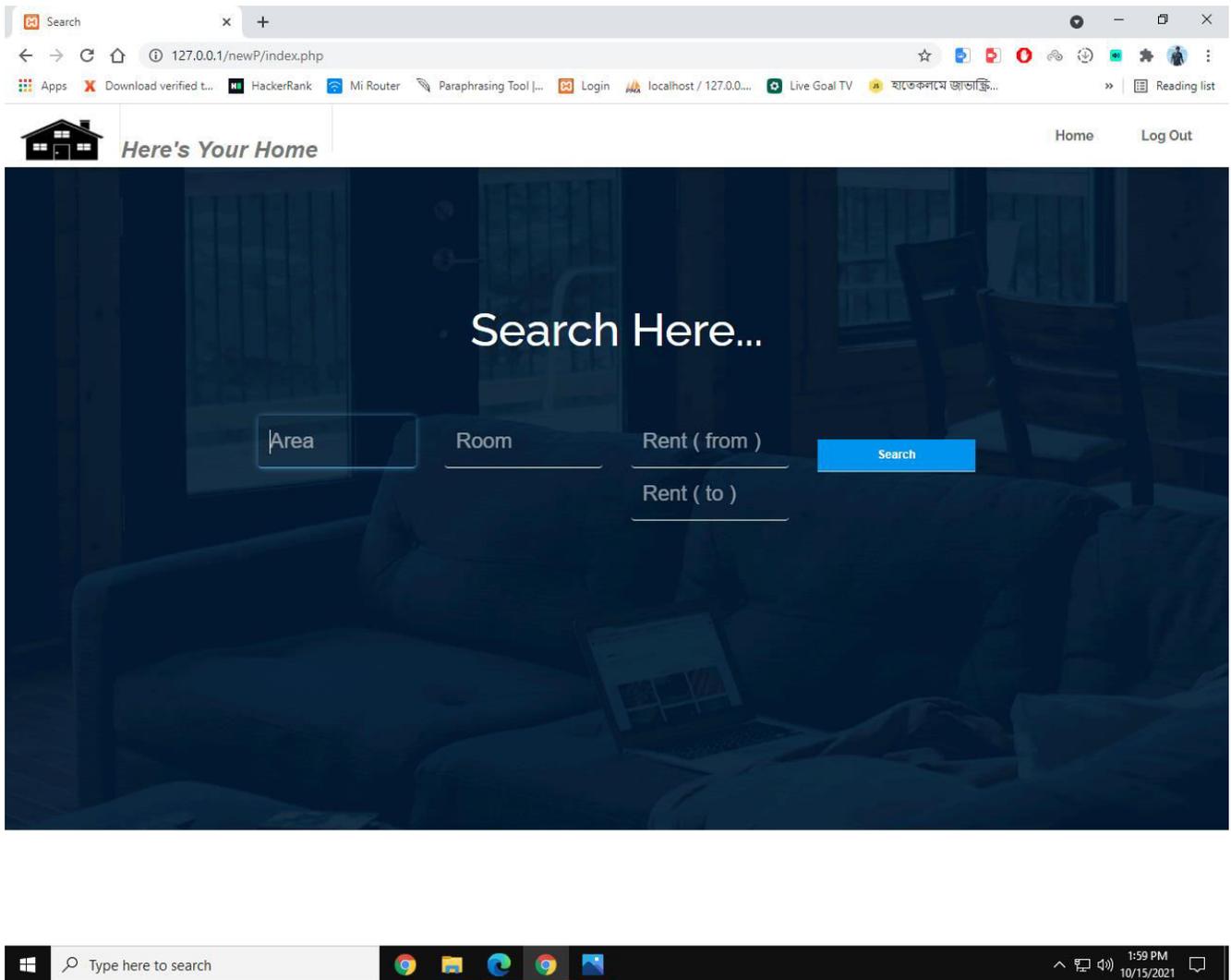


Figure 4.7: Search page

This image 4.8 depicts the GUI for freshly posted rental announcements with photos and

comments.

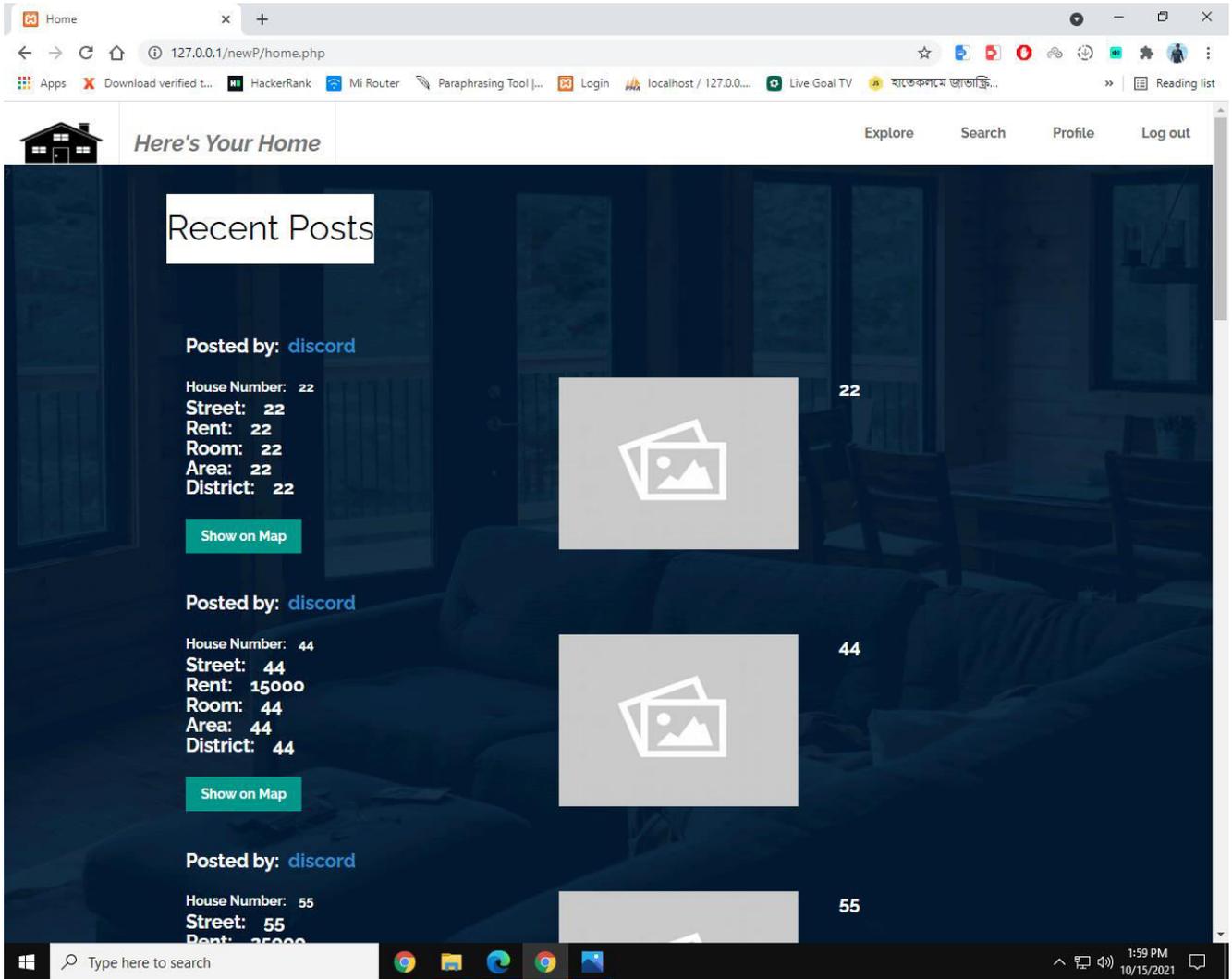


Figure 4.8: Recent posts

The GUI explore page will be shown in Figure 4.9. This website allows users to explore a certain location and discover a rental home that meets their needs.

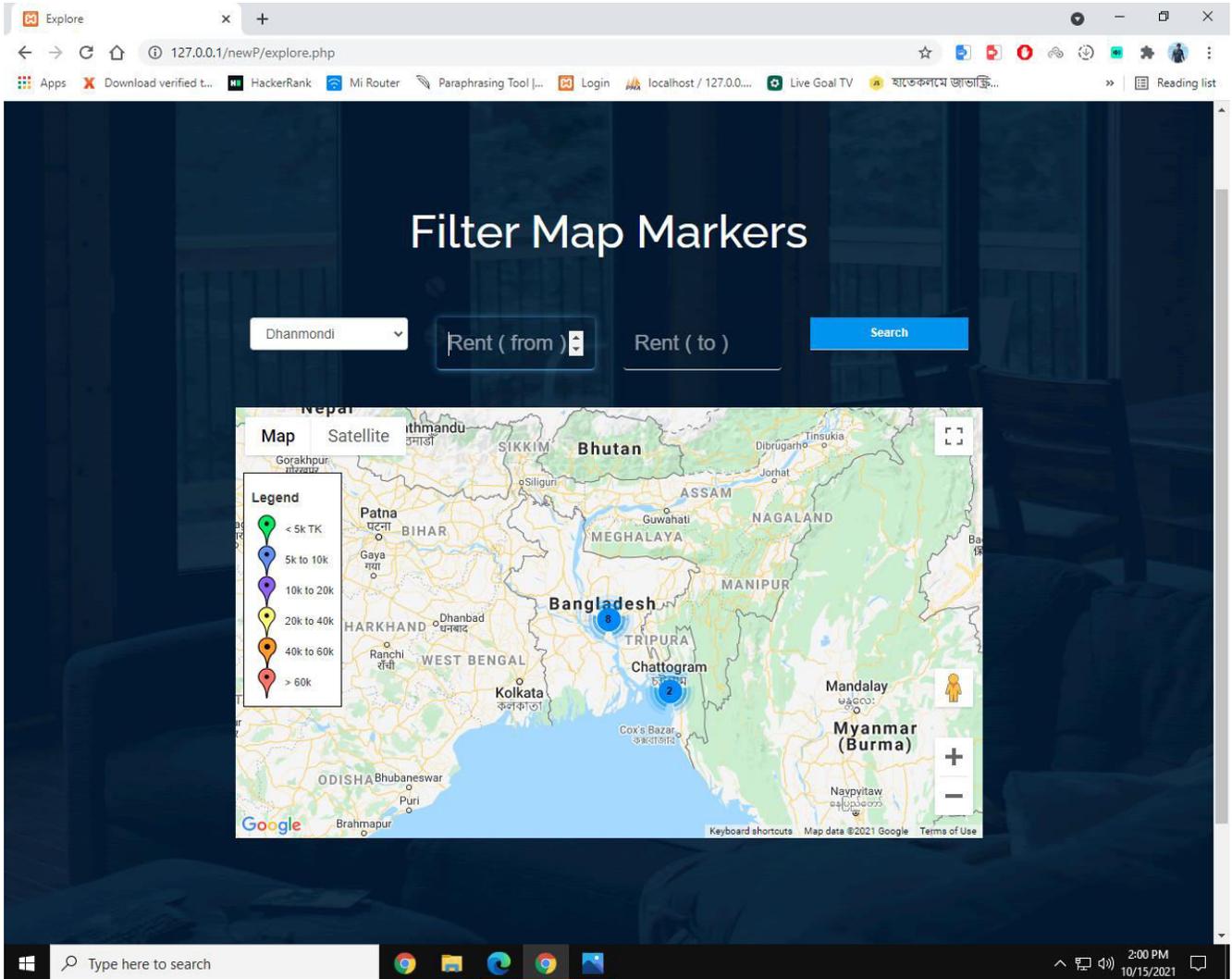


Figure 4.9: Explore

## 4.2 Back-end Design

Here's the Here's Your Home database relationship table.

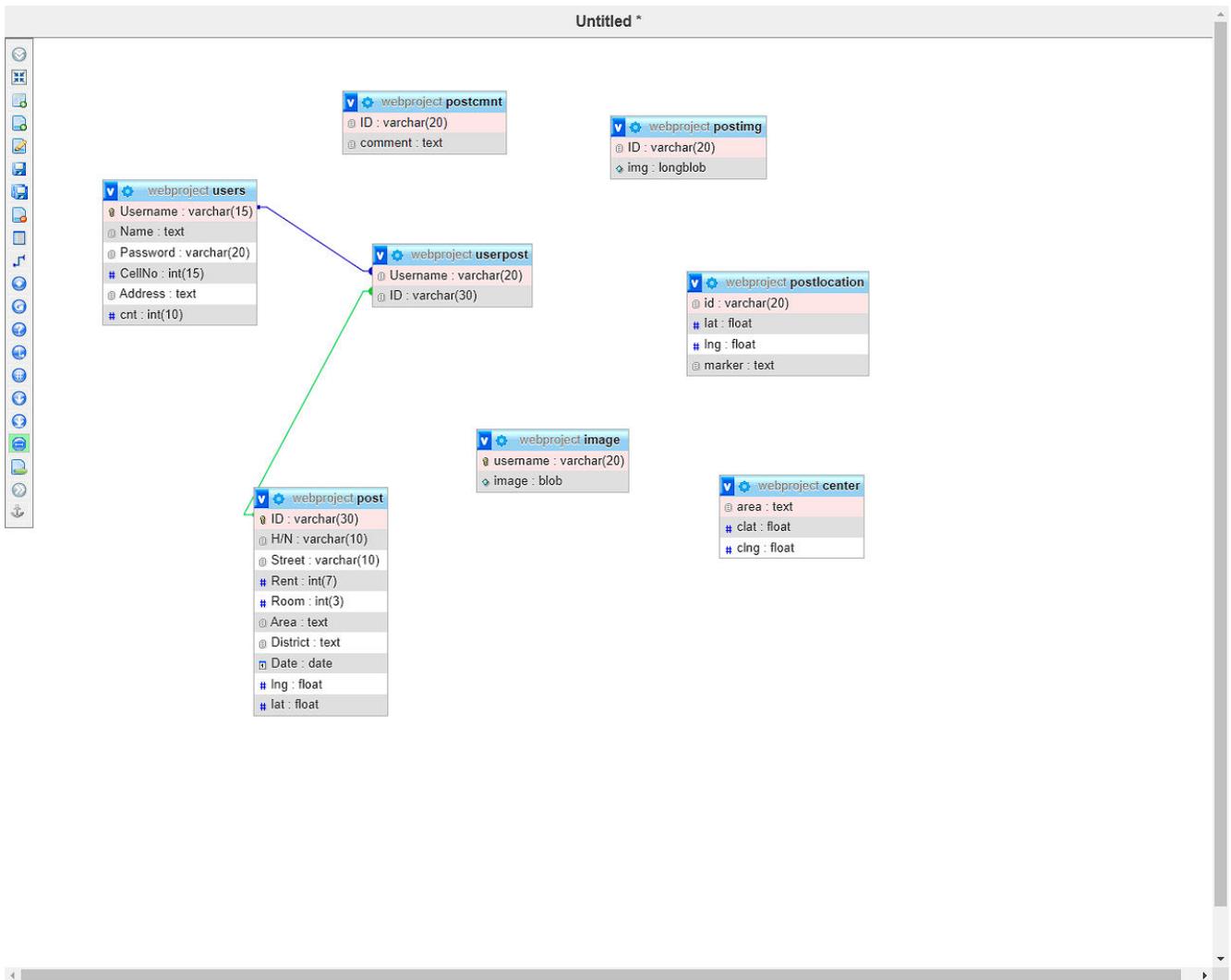


Figure 4.10: Database Relationship Table

The database table that has been established will be described in the following figure 4.14.

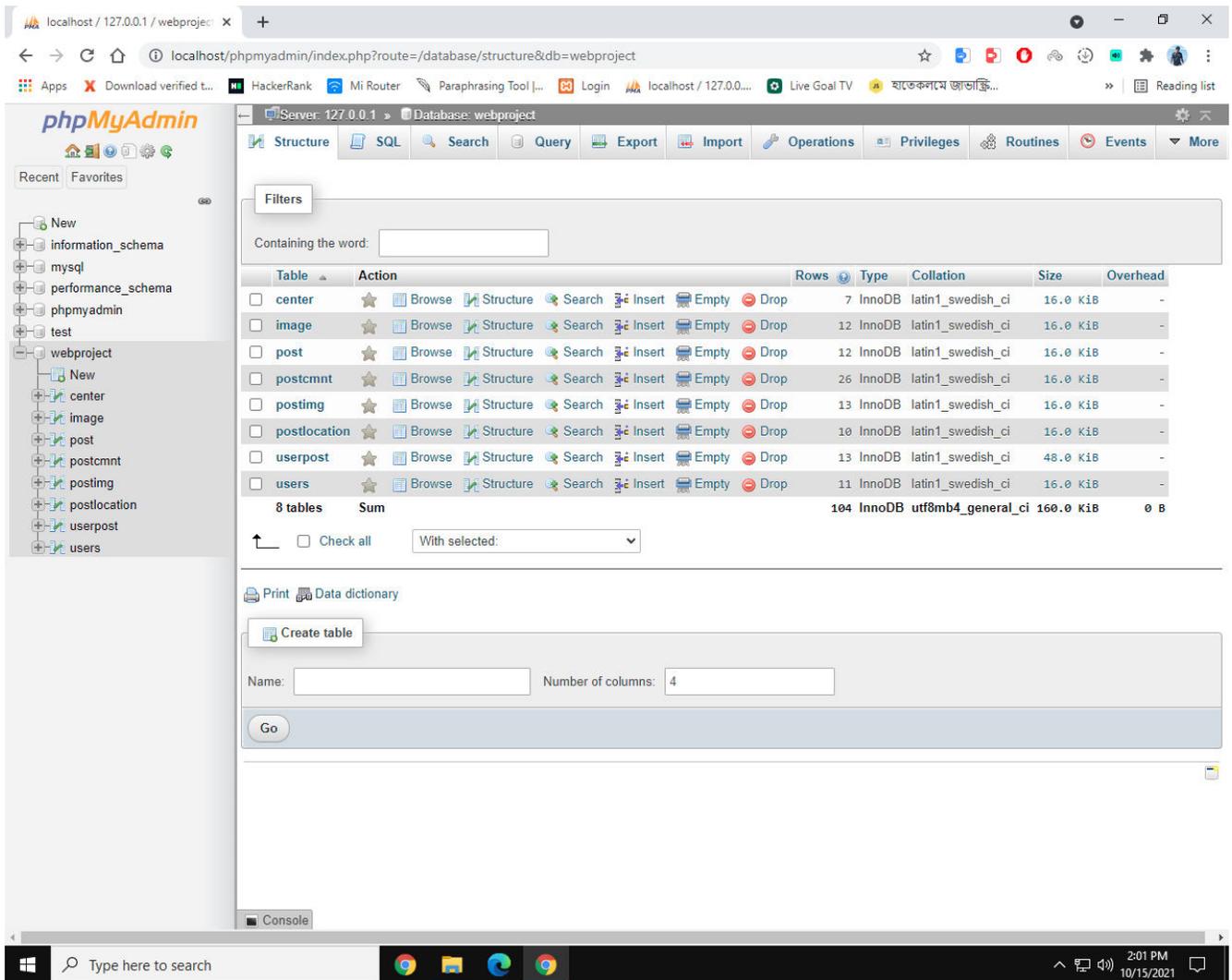


Figure 4.11: Database Table

### **4.3 Implementation Requirements**

The following items and technologies are required to complete this project. The following is a list of the items on the list:

- Internet Browser
- VS code
- MySQL server
- PHP
- HTML
- CSS
- JQuery
- Bootstrap

## CHAPTER 5

### IMPLEMENTATION AND TESTING

We'll discuss how we're planning on putting this project together. Because this is a web-based application, both the front end and the back end will be constructed. The final version will be depicted in several figures.

#### 5.1 Implementation of Database

This chapter will show the way we created our database. This section will go through way we connect to databases and how they interact with users. Also included is the process of developing relationships.

```
index.html X
H: > Xampp > htdocs > newP > <> index.html > ?
1  <?php
2  session_start();
3  include('connection.php');
4  error_reporting(0);
5  $userprofile = $_SESSION['user_name'];
6  $query = "SELECT * FROM USERS WHERE Username = '$userprofile'";
7  $data = mysqli_query($conn,$query);
8  $result = mysqli_fetch_assoc($data);
9  $name = $result['Name'];
10
11 $query1 = "SELECT * FROM image WHERE username = '$userprofile'";
12 $data1 = mysqli_query($conn,$query1);
13 $result1 = mysqli_fetch_assoc($data1);
14 $pic = $result1['image'];
15
16 if($userprofile == true)
17 {
18
19 }
20 else{
21     header('location: login1.php');
22 }
23
24 >?<
```

Figure 5.1: Implementation of Database

## 5.2 Implementation of Front-End Design

Due to the fact that our project is web-based. As a result, a lot of time was spent on the front end design. Because it is via this front end that users engage with the project. We utilized html, css, bootstrap, and jquery to create the front end of our web-based project. We sought to keep our user interface as simple as possible so that users could see all of the information quickly. The implementation procedure is described as follows:

Initial, have a look at how we put up our bootstrap and first page. The page's title has been provided, and then a scaling has been used to get the precise screen ratio for all devices. After that, we've added our bootstrap to the mix.

As a result, the following diagrams will explain how to set up the front end design of the pages:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Search</title>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <!--enable mobile device-->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!--fontawesome css-->
    <link rel="stylesheet" href="CSS1/font-awesome.min.css">
    <!--bootstrap css-->
    <link rel="stylesheet" href="CSS1/bootstrap.min.css">
    <!--animate css-->
    <link rel="stylesheet" href="CSS1/animate-wow.css">
    <!--main css-->
    <link rel="stylesheet" href="CSS1/style.css">
    <link rel="stylesheet" href="CSS1/bootstrap-select.min.css">
    <link rel="stylesheet" href="CSS1/slick.min.css">
    <!--responsive css-->
    <link rel="stylesheet" href="CSS1/responsive.css">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></script>
    <style>
      ul{
        background-color: #eee;
        cursor:pointer;
      }
      li{
        padding:2px;
      }
    </style>
  </head>
```

Figure 5.2: Initialize front end of search page

```
!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Chamb - Responsive E-commerce HTML5 Template</title>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <!--enable mobile device-->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!--fontawesome css-->
    <link rel="stylesheet" href="css/font-awesome.min.css">
    <!--bootstrap css-->
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <!--animate css-->
    <link rel="stylesheet" href="css/animate-wow.css">
    <!--main css-->
    <link rel="stylesheet" href="css/style.css">
    <link rel="stylesheet" href="css/bootstrap-select.min.css">
    <link rel="stylesheet" href="css/slick.min.css">
    <!--responsive css-->
    <link rel="stylesheet" href="css/responsive.css">
  </head>
```

Figure 5.3: CSS and Bootstrap Login page

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Chamb - Responsive E-commerce HTML5 Template</title>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <!--enable mobile device-->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!--fontawesome css-->
    <link rel="stylesheet" href="css/font-awesome.min.css">
    <!--bootstrap css-->
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <!--animate css-->
    <link rel="stylesheet" href="css/animate-wow.css">
    <!--main css-->
    <link rel="stylesheet" href="css/style.css">
    <link rel="stylesheet" href="css/bootstrap-select.min.css">
    <link rel="stylesheet" href="css/slick.min.css">
    <link rel="stylesheet" href="css/jquery-ui.css">
    <!--responsive css-->
    <link rel="stylesheet" href="css/responsive.css">
  </head>

```

Figure 5.4: Implementation of CSS and bootstrap on profile page

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Register Here</title>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <!--enable mobile device-->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!--fontawesome css-->
    <link rel="stylesheet" href="css/font-awesome.min.css">
    <!--bootstrap css-->
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <!--animate css-->
    <link rel="stylesheet" href="css/animate-wow.css">
    <!--main css-->
    <link rel="stylesheet" href="css/style.css">
    <link rel="stylesheet" href="css/bootstrap-select.min.css">
    <link rel="stylesheet" href="css/slick.min.css">
    <!--responsive css-->
    <link rel="stylesheet" href="css/responsive.css">
  </head>
```

Figure 5.5: CSS and bootstrap implementation SignUp page

```
<!DOCTYPE html>
<html>
  <head>
    <title>Webslesson Tutorial | Autocomplete textbox using jQuery, PHP and MySQL</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></script>
    <style>
      ul{
        background-color: #eee;
        cursor:pointer;
      }
      li{
        padding:12px;
      }
    </style>
  </head>
```

Fig 5.6: CSS and bootstrap on Login page

## **5.3 Testing Implementation**

It is a web-based software application. As a result, before publishing the application, we'll need to conduct certain tests. Because it will evaluate overall performance and offer information about the application's input and output metrics. A test, on the other hand, can tell you how many times it can be performed and how well it performs. Application testing may be done in a variety of ways. For this application, we've also done some testing. Here are a few instances of what I'm talking about.

### **5.3.1 Unit Testing**

When the code is finished, unit testing is performed to check for any errors made by the developer. So, in our case, we double-checked while coding to see if there were any early issues. It happens when we develop code and immediately test it.

### **5.3.2 Integration Testing**

We began with the design of the front end. We completed merging all of the backend code after that. After that, we checked to see whether any of the layouts were broken. I also double-checked to make sure that all of the user interfaces are operational.

### **5.3.3 Functional Testing**

All of the features that we first selected for our web app. We've met all of the project's first requirements. In addition, each function is looked at independently.

### **5.3.4 System Testing**

The purpose of system testing is to see if an application works in a different system environment. We tested our program on a range of browsers, operating systems, and PCs, despite the fact that it is purely web-based.

### **5.3.5 Usability Testing**

Usability testing assesses how successfully a user interacts with an interface and how enjoyable it is to use. To do so, we're putting a few folks to the test to evaluate how they like the UI and how easy it is to navigate.

### **5.3.6 Test Results and Reports**

Even yet, after all of the testing, there are some discoveries on that test. The test findings and reports are as follows:

### **5.3.7 Result of Unit Test**

Several issues arose during the coding process. Then they were all resolved straight quickly, and no new problems emerged.

### **5.3.8 Result of Integration Testing**

The frontend and backend code were successfully combined in this case. There were no faults or issues detected during this test.

### **5.3.9 Result of Functional Testing**

During a certain testing, our post creation appears to be coming up short at times. Then we saw that the code has a number of logical mistakes. The problem was then remedied. There was a caching issue on the browser on another front end site management. This problem was also fixed by deleting and replacing old files with fresh ones. As a result, we've finished our functional test.

### **5.3.10 Result of System Testing**

Our method has shown to be effective in a wide range of systems and situations.

## **CHAPTER 6**

### **CONCLUSION AND FUTURE SCOPE**

#### **6.1 Discussion and Conclusion**

As a result, everything above is an attempt to demonstrate all facets of our project. This report was also created to have the sense of real-world software development.

We feel that now that we've finished this endeavor, we'll need some outcomes. And we've got to make something of it. As a result, we believe we have a strong knowledge of it. We've decided to utilize this project to take on difficulties that are similar to real-world projects that are done at an industrial level, because web development is our primary aim. As a consequence, we have a deep grasp of how to build a layout that never breaks, as well as how to construct a fully responsive layout.

We, on the other hand, had a comprehensive grasp of how to apply programming expertise to the development sector, despite the fact that we were constructing a web application. The database design was a crucial part of the project. Based on database models and styles, we may have a thorough idea of how to design a database. We can get a lot more out of this project than we've talked about thus far.

## **6.2 Scope For Further Development**

Despite the fact that we had to complete the chores in an acceptable amount of time, we were clearly unable to accomplish our whole goal. So, what is the point that we need to make in order to broaden our horizons? We can include an Android-based application into this House renting system. Because some users choose to use an application instead of visiting a website.

We are, however, working on integrating a chatbot into the system. Users will be able to find what they need as a consequence. Our AI bot will react and save the information provided by the user for future enquiries. As a result, the bot may respond to the user in a timely and useful manner..

Another limitation of the system is that it now does not enable money payment methods; however, additional forms of payment methods will be added to this software in the future.

So there you have it: all you need to know about our project and its long-term objectives. This may be the best house-renting system ever if all of these processes are followed.

## REFERENCES

### WEB SITE REFERENCES:

- <https://www.bproperty.com> Date: (02/06/21)  
Bproperty is Bangladesh's leading property solutions provider.
- <http://www.resgroup.net/> Date: (15/06/21)  
Real Estate Solutions Group, is a business with broad real estate, urban planning and economic development experience.
- <http://www.a1realestate.solutions> Date: (10/07/21)  
Dundas Real Estate, founded in 1995, and Coastal Real Estate have merged to form A1 Real Estate Solutions.
- <https://www.wikipedia.org/> Date: (07/08/21)  
The Wikimedia Foundation, a non-profit organization that also runs a number of other projects, hosts Wikipedia.
- <https://www.w3schools.com> Date: (29/08/21)  
W3Schools is designed with learning and training in mind. To increase reading and learning, examples might be simplified.
- <https://rentalhomebd.com/> Date: (04/10/21)  
RentalHomeBD is a renowned letting-property portal website that was founded in 2012 and offers a full search option for property seekers.
- <https://stackoverflow.com/> Date: (15/08/21)  
Nearly everyone who programmers use Stack Overflow's public platform to learn, share their skills, and enhance their skills.

## BOOK REFERENCES:

- Software Engineering - R.S. Pressman Published: September 9, 2019

Roger S. Pressman, President of R.S. Pressman & Associates, is an American software engineer, author, and consultant. He is also the founder of EVANNEX and its director of Engineering.

- PHP for Dummies - Janet Valade Published: January 6, 2004

Janet Valade is the author of PHP & MySQL For Dummies. In addition, she has authored and revised chapters for several Linux books and for a Webmaster certification book.

- PHP Beginners Guide by Published: October 23, 2008

Melonfire's creator and CEO, Vikram Vaswani, specializes in software consulting as well as content development and syndication services. He is the author of PHP Programming Solutions, a book about PHP programming.

- JavaScript by John Pollock Published: December 24, 2003

John Pollock is a Canadian sports journalist who is most known for his work on Fight Network Radio, Live Audio Wrestling, and developing the POST Wrestling website.

## APPENDIX A

### ORIGINALITY REPORT

<b>24%</b>	<b>22%</b>	<b>1%</b>	<b>16%</b>
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

### PRIMARY SOURCES

<b>1</b>	<b>www.ukessays.com</b> Internet Source	<b>7%</b>
<b>2</b>	<b>Submitted to Daffodil International University</b> Student Paper	<b>6%</b>
<b>3</b>	<b>dspace.daffodilvarsity.edu.bd:8080</b> Internet Source	<b>5%</b>
<b>4</b>	<b>Submitted to University of Greenwich</b> Student Paper	<b>1%</b>
<b>5</b>	<b>www.coursehero.com</b> Internet Source	<b>&lt;1%</b>
<b>6</b>	<b>books.google.co.in</b> Internet Source	<b>&lt;1%</b>
<b>7</b>	<b>Submitted to Sri Lanka Institute of Information Technology</b> Student Paper	<b>&lt;1%</b>
<b>8</b>	<b>Submitted to Sheffield Hallam University</b> Student Paper	<b>&lt;1%</b>
<b>9</b>	<b>fr.scribd.com</b> Internet Source	<b>&lt;1%</b>