

PAYING GUEST MANAGEMENT SYSTEM

PROJECT REPORT

Submitted by

AMJAD V K

KMC20MCA-2003

to

the APJ Abdul Kalam Technological University in partial fulfillment of the requirements

for the award of the Degree

of

Master of Computer Applications



Department Of Management Studies & Computer Applications

KMCT College of Engineering

Kallanthode, NITC P.O, Kozhikode-673601

FEBRUARY 2022

DECLARATION

I hereby declare that the project report "**Paying Guest Management System**", submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of **Mr. Ajayakumar K K**. This submission represents ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that i have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree.

Place: Kallanthode

AMJAD V K

Date: 25/02/2022

DEPARTMENT OF MANAGEMENT STUDIES & COMPUTER APPLICATIONS

KMCT COLLEGE OF ENGINEERING

Kallanthode, NITC P.O, Kozhikode-673601



CERTIFICATE

This is to certify that the report entitled "**Paying Guest Management System**" submitted by **AMJAD VK (KMC20MCA-2003)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications is a bonafide record of the project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor

Project Coordinator

Head Of The Department

ACKNOWLEDGEMENT

I would like to take this opportunity to extend my sincere thanks to people who helped me to make this project possible. This project will be incomplete without mentioning all the people who helped me to make it real.

First and foremost I thank **Dr. K Kalaiselvan (Principal of KMCT College of Engineering)** who gave me all support to this project. I also thank **Mr. Ajayakumar K.K (Head of the Department,MCA)** for providing all the facilities and resources for my project. I would also like to express my gratitude towards **Mrs. Sabna T.S (Assistant Professor, MCA), Project Coordinator** and **Mr. Ajayakumar K K, Project Guide** for their continuous support, guidance and supervision without which the project wouldn't have been a reality. I would also take this opportunity to thank all my friends who took time out of their busy schedule to encourage, support and motivate me which has been the key reason for the successful completion of this project.

Above all I thank God, the almighty for his grace without which it would not have been possible to complete this work on time.

ABSTRACT

The need for accommodation is expected to grow significantly over the next decade. Users can find good and affordable housing of their choice, they can specify their requirement and the system will match the users requirements with the details in the database and after finding the most fitting match the required details are given to the user. Students and Employees don't need to visit and check all the providers, this system will help them to save time and find the most suitable housing of their needs.

This system helps the user to end up with the roommates who are like-minded as per the requirements the user selects, some might be in the mood for some fun or a good sleep after the tiring job or students who wants concentrate on their studies and works after the class. If the user face any difficulties they can directly notify the PG Owner and they can take quick actions and solving them as soon as possible, for example maintenance of the building, food etc. User may stay for a long or short period, the owner can easily calculate the fees easily when the user vacates the room and the system can update that there is a slot available. Owner doesn't need to manually update the vacancies. This system is a web-based application that can be easily accessed anywhere and can ease the booking and maintenance for both Owner and Users.

Contents

Chapter 1 INTRODUCTION	2
1.1 General Background	2
1.2 Objective	3
Chapter 2 SYSTEM ANALYSIS	4
2.1 Existing system	4
2.2 Proposed system	6
2.3 Module Description	8
2.3.1 Admin	8
2.3.2 Owner	8
2.3.3 User	8
2.4 Feasibility Study	10
2.4.1 Operational Feasibility	10
2.4.2 Technical Feasibility	10
2.4.3 Economic Feasibility	10
2.5 System environment	11
2.5.1 Developer Requirement	11
2.5.2 User	11
2.6 Actors and Their Roles	12
2.6.1 Admin	12
2.6.2 Owner	12
2.6.3 User	12

Chapter 3 METHODOLOGY	13
3.1 Introduction	13
3.2 Uml Diagrams	14
3.2.1 Use case Diagram	14
3.2.2 Activity Diagrams	15
3.3 User Story	18
3.4 Product Backlog	21
3.5 Project plan	24
3.6 Database Design	26
3.6.1 Owner	26
3.6.2 User	27
3.6.3 Login	28
3.6.4 Pg_details	29
3.6.5 Booking	30
3.6.6 User_complaint	31
3.6.7 Owner_complaint	31
3.6.8 Review	32
3.7 Forms	33
3.7.1 Login	33
3.7.2 User Registration	34
3.7.3 Owner Registration	35
3.7.4 Search PGs	36
3.7.5 View PG Details	37
3.7.6 View User Profile	38
3.7.7 View Current Stay Details	39
3.7.8 Add Complaint	40
3.7.9 View and Respond to Complaints	41
3.7.10 Admin Home	42
3.7.11 User Home	43
3.7.12 Owner Home	44

3.7.13	Accept or Reject Bookings	45
3.7.14	Admin Manage PGs	46
3.7.15	Admin Manage User's	47
3.7.16	Owner Add PG	48
3.7.17	Owner PG Report	49
3.7.18	Add Review	50
3.7.19	Owner Update PG details	51
Chapter 4	TESTING AND IMPLEMENTATION	52
4.1	Testing	52
4.1.1	Test Case-1	52
4.1.2	Test Case-2	53
4.1.3	Test Case-3	54
4.1.4	Test Case-4	55
4.2	Implementation	56
Chapter 5	RESULT AND DISCUSSION	57
Chapter 6	CONCLUSION	58
6.1	References	59
6.2	Appendix	60
6.2.1	Source Code	60
6.3	Screenshots	78
6.3.1	Homepage	78
6.3.2	Login page	79
6.3.3	Owner Registration	80
6.3.4	User Registration	81
6.3.5	Owner Homepage	82
6.3.6	User Homepage	83
6.3.7	Admin Homepage	84
6.3.8	Add PG	85
6.3.9	Search PGs	86
6.3.10	Book a PG	87

6.3.11	Manage Bookings	88
6.3.12	User Add Review	89
6.3.13	User Add Complaint	90
6.3.14	Vacate PG	91
6.3.15	Admin Report	92
6.3.16	Owner Report	93
6.4	Git History	94

Chapter 1

INTRODUCTION

The “Paying Guest Management System” has been developed to eliminate and reduce the difficulties faced in the manual accommodation management. And most importantly this system can be used to find the best possible environment for the tenant, also by using this system the owner can easily manage their PG’s very efficiently. Anyone who is looking for a place to stay for some time can easily search through this application and find the best possible facilities according to the tenants requirements.

1.1 General Background

Accommodation is an important part in our life for someone who is far away from home for work,studies,tours etc should be able to find the best suitable and affordable place to stay for their needs can locate that particular facility through this system. Accommodation can cost a large amount for anyone who is far from home and it is difficult to find the best housing for their needs. Owners of houses,buildings,apartments etc can rent their place for an income if they are not using it currently.This system is can enhance the working of the management of accommodation. This system is implement for ‘Heidhin’ Paying Guest facility to manage their operations efficiently. Also this system can be used for anyone who is willing to rent their place for an income can register in this system and the people who are looking for housing can easily find your facility and book a bed if its available.

1.2 Objective

The objective of this system is to enhance the functioning of a Paying Guest facility for both Users and Owners. They have a great advantage of using this system. A tenant who is in need to find a place to stay can easily find one by searching in this application. If all the user's requirement are met then the suitable results are displayed the user can book the PG of their choice and find out the best housing by searching through the application. An advantage of this system is that the occupant only needs to pay for the facilities that are required by he/she. Occupant won't need to pay extra money for the facilities they are not using. For example a Student won't need to stay in a fully furnished room they only need a good place to sleep and study, but in the case of a family they must need all the facilities that are required for a family to stay such as Attached Bathroom,Kitchen,TVs and Fridges for their daily needs.

This application is useful for both Users and Owners. It is designed for Users which can be students,families etc or anyone who want 's to stay at a particular place for a long or short period of time, this system can find out the best facility according to the requirements given by the user. The Owners can easily rent out their unused and extra rooms and building for an income. Owners of Paying guest facilities have different kinds of housing provided, it can be a single Single/Double room, shared rooms, fully furnished rooms etc. They can post all the details about their facility and users can search and filter out the best housing of their needs and book a certain facility through this system. The owners can also collect the rent as soon as possible when the date occurs. This system will make it easier for owners who need to manage a large no of tenants in their housings and they can also store all the information about the people in their facility and the information can be retrieved easily when needed. This system is able to display if the PG is currently full and the number of available spaces to accommodate. Users can book the PG if they are willing to stay here.Overall this system is very beneficial for both Owners and Users for providing and finding the PG facility in a certain location.

Chapter 2

SYSTEM ANALYSIS

2.1 Existing system

In the existing system the process of finding a housing is not at all easy, he/she need to visit the PG they are interested and verify that all the facilities are provided as the owner advertised. This is a very time consuming task because there may be more than one PG's that they are interested, it is not possible to visit each of them to know more about it. The searching process included a lot of hardship and wastage of time .If the tenant is from another state the language may be a barrier he/she cannot communicate easily and find out more about the PG. Currently if we want to book a PG we must go directly to the owner to book one. If the tenant needs to report a complaint to the owner they must wait for the owner to visit the PG or by calling. This can be very difficult for the owner to manage if the owner have more than one PGs.

Owner's of the PGs advertise their facility through billboards,notices,newspaper advertisements and also through sites like OLX and NAUKRI etc. Tenants can only know very little information about that facility because of the less information provided and difficulty in accessing information from a remote location. They manage their PG in the old-fashioned way by writing down all the business on a paper or by using a simple file-system. It is very difficult to view and understand all the data's of the tenants. Owner cannot easily know when there would be vacant rooms or when should they collect the rent. If there is large no of bookings owner cannot do all the process easily. First the owner

must prompt the tenant to fill out a form to collect all the information. And verify all the information's are genuine to continue the process of admission. Owners should manage their PGs very efficiently, for example if there is a complaint raised from the tenant the owner can only resolve it by interacting directly with the tenant. In the existing system this can be hefty job if the owner has more than one PGs. The aim of the owner is to accommodate the maximum number of people's in their PG, so the owner must accept first is the type of tenant that would stay in the facility for a longer period of time so the owner can achieve maximum profit and make their PG occupied at all the time. A PG is very different from a Hostel people staying in a PG are with different aims some want to stay at a place for a while, some wants to have a homely feeling and so on. If the owner has many PGs with different facilities, it is not easy to manage them. He must be fully aware of the no of vacant rooms of different types and accommodate the new tenants to the particular rooms according to their needs, without a proper system it is very difficult to manage these activities. Collection of rent from tenants can also be a difficult task the owner must check if the rents are due or not manually and collect rent regularly from everyone at the end of the month. But in a Paying Guest system rent may not be collected on a monthly basis because some people stay for only some week or for some days. Rents are different for different tenants some may require food, some need a kitchen to cook, some only require a place to stay and so on, collecting these rents are difficult if there does not exist a paying guest management system. The existing system work efficiently as not equally same as a facility management system there may be some difficulties following the existing one only when we have to manage a large no of guest's.

2.2 Proposed system

Proposed system is a Paying Guest Management system by using this both Owners and Tenants can easily manage and find the best suitable PG for the tenant's requirements. The main objective of this system is to find the best suitable environment for the tenant according to their requirements. They can live in the best possible environment in the next few days or months of their choice. The User/Tenant find it very easy to find an accommodation when they are far away from home can just search one by giving out the requirements and the location.

Managing the PG through this system can be very easily done. Owners can easily collect all the required information when needed very easily. Also the owner can easily calculate the rent when the tenant is vacating the PG. If he/she wants to report a problem about the PG it can be done easily by directly notifying the owner through the Complaint Portal. And the problems can be easily rectified by the owner as soon as possible.

Registration

Admin will register Owners only they can enter the details of the Owners. When an Owner register into this system the Admin will verify that all the information given by the Owner is genuine and then only the Owner get accepted in this system. A unique username and password are given to the owners after they get accepted to use this application. User or Tenants also need to register to use this application they must enter all the required details and they must fill out all the mandatory fields to use this application.

Booking

Booking is done by the user he/she can book a PG easily after finding a suitable one. They can specify the no of days they would stay in the PG and also all the facilities they want to use in the PG like Cooking facility, Air Conditioner, Entertainment Systems etc, the main advantage is that the tenant only need to pay for the facilities they are using. In the process of booking the user can see the total amount he/she would cost after selecting

the facilities they need.

View Reports

All of the users in this system can view their reports through the Dashboard. Admin can view the full details about the no of PGs registered by the Owners. Owner can view all the details of the tenants like their stay details, rent details, status etc. Users dashboard have information like their accommodation details like their stay details, rent and all other information that ease the User to manage their stay in the PG.

Add Complaints

Users can make a complaint to the Owner if he/she faces any problems in the facility. This reduces the time consumption that normally takes to solve a complaint. Owners can also make complaints to the Admins who are managers of this system, if they faces any problem or some functionality should be added for the improvement of the system.

Rent Collection

Owners can understand easily when they have to collect the Rents from the tenants the amount each person should need to pay also if there is any dues from the tenant. Using this system can make the job easy for the owners to collect rents.

Advantages of proposed system

- Reduces time consumption
- User Friendly
- Easy to access information
- Easy to maintain

2.3 Module Description

2.3.1 Admin

The Admin has the central control of managing the PG's in a particular area. Admin can login to the system after entering the username and password. They have control of the overall functionalities of the management of PGs. Admin can add and update the PG's also giving out the access to register a PG by the Owner. Admin can view all the Owners details and PG's provided.

- View and Manage PGs
- View and Manage Owners
- Monitor any Unwanted Behavior and filtering them.

2.3.2 Owner

Owners are the people who are willing to rent out their building are registered in this system with the permission from the Admin and can avail their facility to the User's. They upload all the details about their facility including the pictures and a user can easily view it while searching for a PG. Owners are notified and can take actions on receiving the complaints from the tenants easily. They can know when they should collect the rent from tenants and exactly know the correct amount the owner should collect from the tenant. Owner need to change the information's when they are necessary, like if there is any problem with a certain room the owner need to remove it from the PG this can be done with simple steps. Owner plays an important role in the smooth and efficient operation of every PGs.

- View and Manage rooms
- Solve complaints received from user.
- Collect rent from tenants while vacating the PG

2.3.3 User

User's are the ones who want to find a housing at a certain location. They must register first in this application then only they can view the PG's available. They can search for a particular housing provider by entering their requirements with the location and they can

view all the details of the PG's and can book one in a certain PG if there is a vacancy. They can also they can calculate their expenses by entering the check in and check out date. The user can always add an extra facility that may require after selecting them at the booking time. They can notify the owner if there are any issues faced. The owner can easily view the complaints received and know all the details like who raised the complaint and the owner can resolve the issue as soon as possible. And view the status of your complaints through the dashboard of the user that if it is solved or not.

- Book a PG
- Vacate and Extend the stay
- Add Facilities when necessary
- Make a Complaint
- Contact the Owner

2.4 Feasibility Study

2.4.1 Operational Feasibility

It is to understand whether the proposed solution will solve the problem or not. It is important to determine how the new web application improves the functioning of a PG Facility. Owner would be able to easily rent out their buildings easily through this application. Thus Users can easily find the suitable PG of their choice by registering and booking a certain facility through this system. This is the indication of operational feasibility of the proposed system.

2.4.2 Technical Feasibility

Assessing technical feasibility is to evaluate whether the newly developed system will perform adequately and whether the individual has ability to construct the proposed system or not. The proposed system is a web-based application. It allows the users to conduct an event from announcing event details, accepting registrations and final result publication. This system will be made using HTML,CSS,JavaScript,PHP and MySQL. The chosen platform, tools and software has the technical capacity required for the system. Hence the proposed solution is technically feasible.

2.4.3 Economic Feasibility

Economic Feasibility study is a vital part dealing with factors that can be quantified, measured and compared in monetary terms. The software's used are widely available and will be of less cost. Moreover anybody with a smart device can access the web application free of cost. Hence the proposed system is economically feasible.

2.5 System environment

2.5.1 Developer Requirement

2.5.1.1 Hardware requirement

- Processor : Intel Core i3
- RAM :4GB
- Storage : 500GB Hard disk

2.5.1.2 Software requirement

- Operating system : Windows 10
- Front end : HTML, CSS, PHP,JS
- Back end : MySQL
- IDE : Visual Studio Code
- Web browser : Internet Explorer/Google chrome/Firefox

2.5.2 User

- A Laptop or a Computer.

2.6 Actors and Their Roles

2.6.1 Admin

Admin in this project is responsible for managing this in a web-based application. Various features and services are:

- View and Manage Paying Guest facilities
- View Report
- Giving unique username and password to Owners
- Filtering and Monitoring Unwanted Behaviours

2.6.2 Owner

The Owners can access different services of application. Owners have to login first. After login they can utilize various features and services.

- View and Manage tenants
- Get information about rents
- View and Reply to complaints

2.6.3 User

User is the next actor who also plays a very important role in this system.

- Registering into the system
- Login to the System
- Search PG by a location
- View and Book a PG Facility
- Vacating a PG Facility
- Report Complaints to Owner

Chapter 3

METHODOLOGY

3.1 Introduction

This project follows Agile methodology. Agile software development comprises various approaches to software development under which requirements and solutions evolve through the collaborative effort of self organizing and cross-sectional teams and their customers/end users. It advocates adaptive planning, evolutionary development, early delivery and continuous improvement and it encourage rapid and flexible response to change.

3.2 Uml Diagrams

3.2.1 Use case Diagram

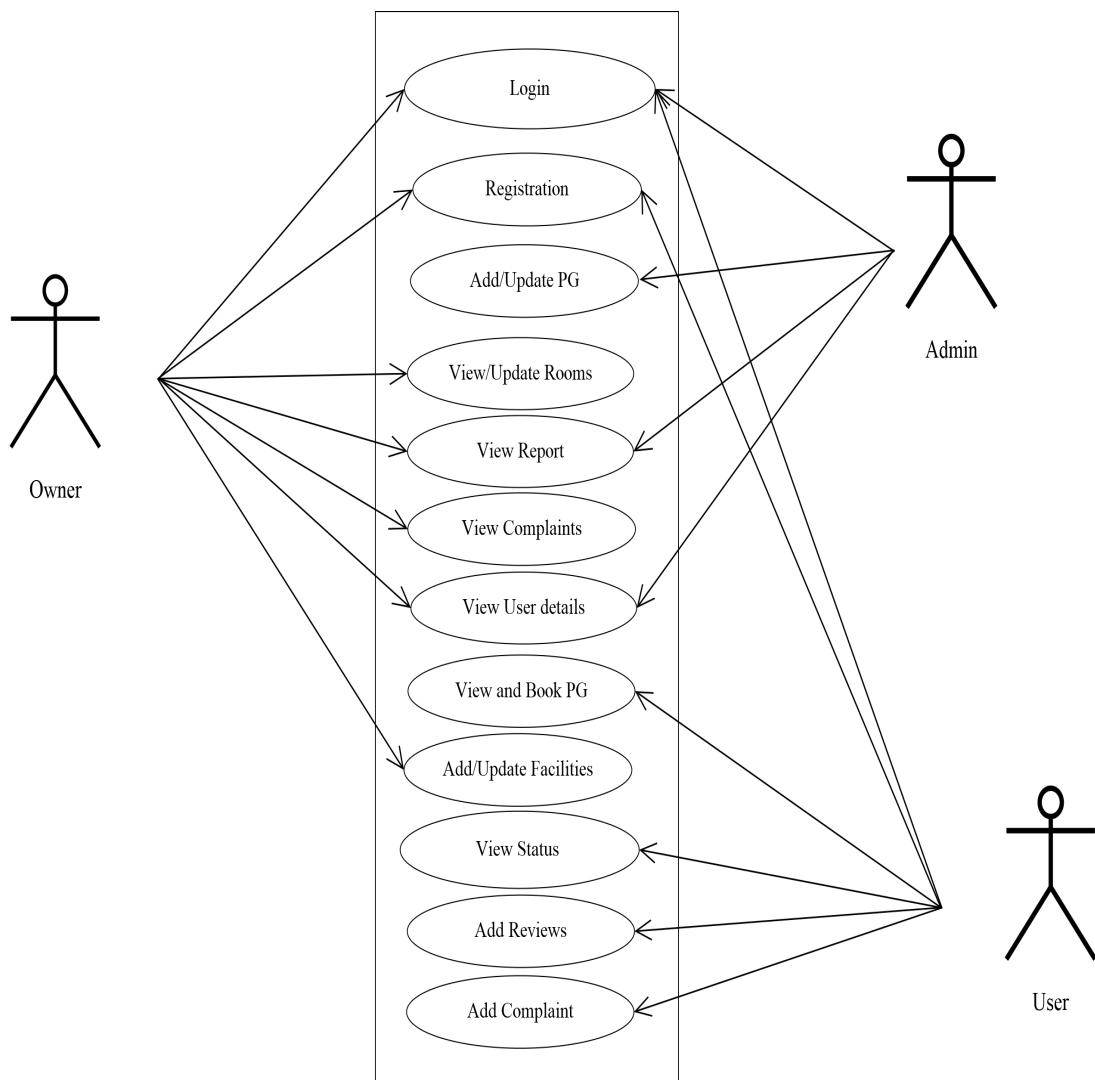


Figure 3.1: Use case Diagram

3.2.2 Activity Diagrams

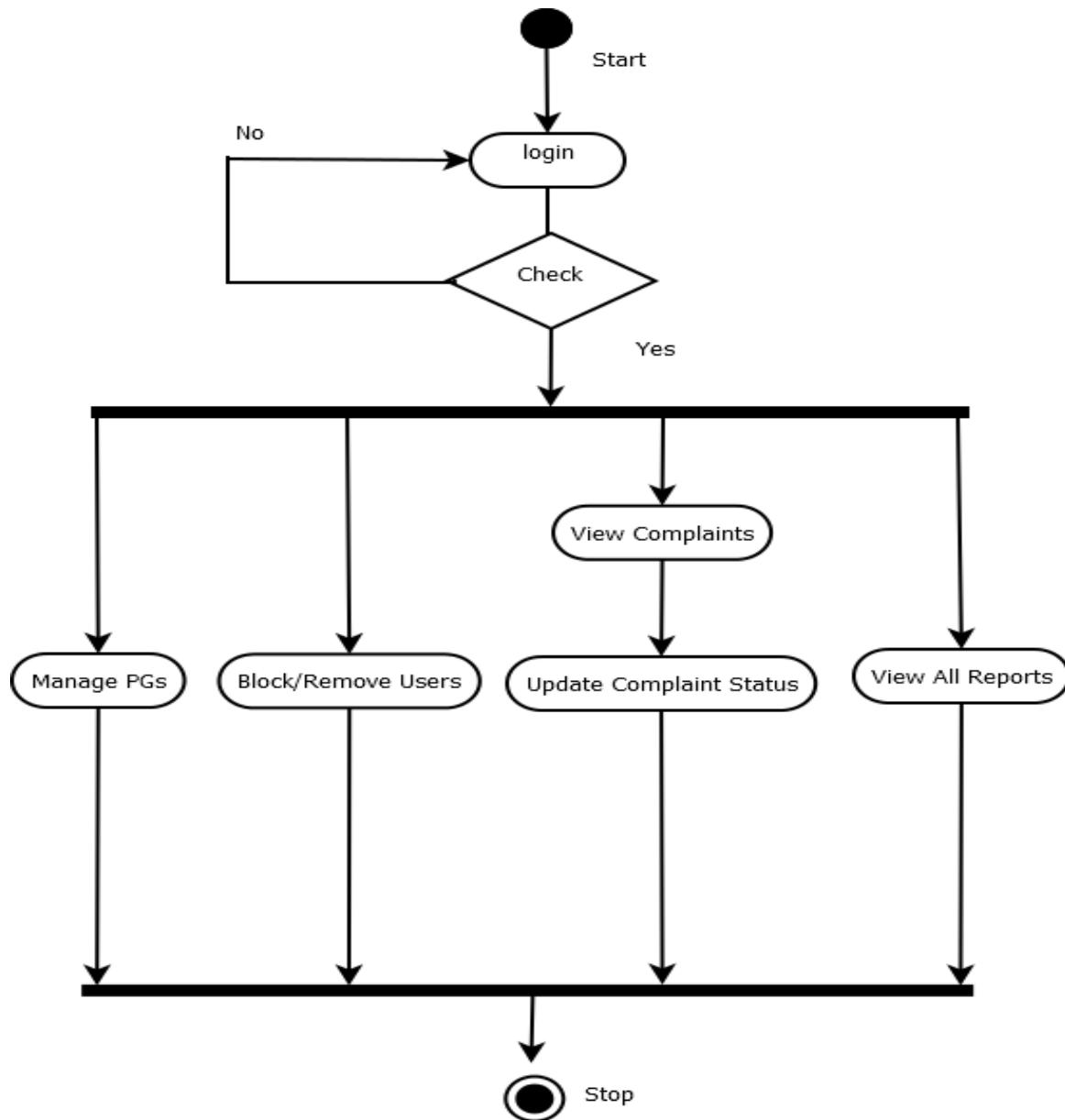


Figure 3.2: Admin activity

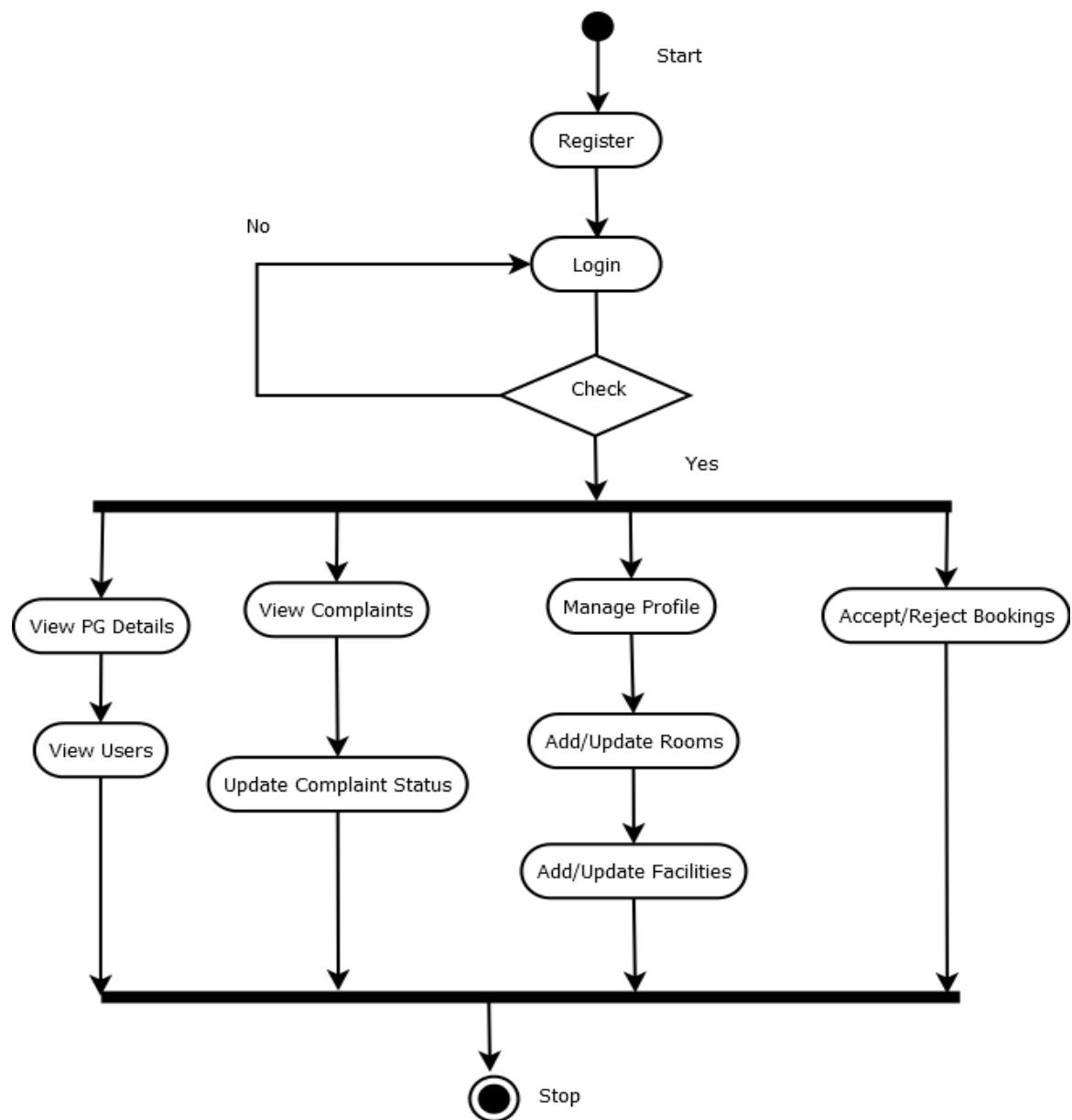


Figure 3.3: Owner activity

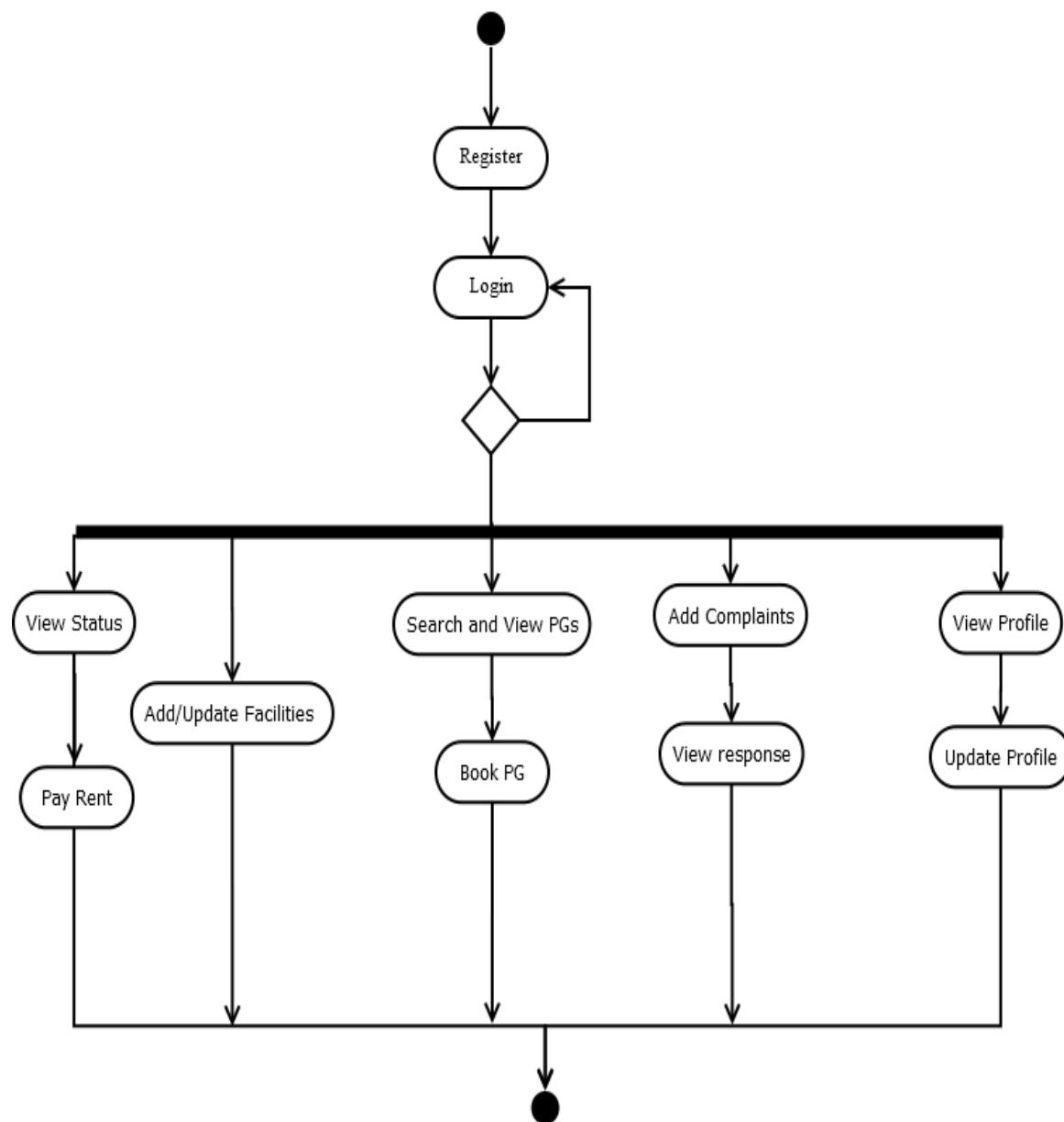


Figure 3.4: User activity

3.3 User Story

User story ID	As a (Type of Users)	I want to (Perform some task)	So that I can (Achieve some goal)
1	Admin,Owner,User	Homepage	Homepage of all users
2	Admin	Login to the system	Access the system
3	Owner,User	Register into the system	Login and Access the system
4	Owner,User	Login to the system	Access the system
5	Owner	Add or Update PGs	Add new PG or Update the details of their existing PG
6	Admin	Manage User's	Search and Delete users from the system
7	Admin	Manage PG's	Search and Delete the total PGs registered

User story ID	As a (Type of Users)	I want to (Perform some task)	So that I can (Achieve some goal)
8	User	View PGs	Search and View details of PGs
9	User	Book PG	User book a PG after selecting the suitable one
10	Owner	Accept/Reject Bookings	Owner Accept/Reject bookings made by the User
11	Owner	View PG details	View the Users staying in their PG
12	User,Owner	Add Complaints	Owner add complaints to the Admin.User add Complaints to the Owner
13	User	Add Reviews	Add review about the PG they are currently staying
14	User	Vacate PG	Vacate from the PG they are currently staying

User story ID	As a (Type of Users)	I want to (Perform some task)	So that I can (Achieve some goal)
15	Owner	View Rent details	View the rents of each users staying in their PG
16	User	View Users Stay Details	Show the details like Rent and No of days stayed
17	Owner	View and Update Complaint Status	View and Update the status of complaints posted by the User
18	Admin	View and Update Complaint Status	View and Update the status of complaints posted by the Owner
19	Owner	View Status of PG	View the Overall details of their PG like the no of Users currently staying,no of vacant rooms
20	Admin	View Status	View the total details like the no of registered User's and no of PGs registered

Table 3.1 User Story

3.4 Product Backlog

User Story ID	Priority (Low, High, Medium)	Size	Sprint	Status (Planned, Progressed, Completed)	Release Date	Release Goal
1	MEDIUM	10	1	COMPLETED	15/12/2021	Homepage of All Users
2	MEDIUM	9		COMPLETED	20/12/2021	Admin Login to the system
3	MEDIUM	8		COMPLETED	25/12/2021	Registration of User and Owner
4	MEDIUM	10		COMPLETED	28/12/2021	Owner and User Login to the system
5	MEDIUM	10	2	COMPLETED	3/1/2022	Add or Update PGs
6	MEDIUM	9		COMPLETED	5/1/2022	Manage the Users registered
7	HIGH	10		COMPLETED	9/1/2022	Manage the PGs registered
8	MEDIUM	10		COMPLETED	13/1/2022	Search and View PG Details

USER STORY ID	PRIORITY (LOW,HIGH, MEDIUM)	SIZE	SPRINT	STATUS (PLANNED, PROGRESSED, COMPLETED)	RELEASE DATE	RELEASE GOAL
9	HIGH	10	3	COMPLETED	15/1/2022	Book a PG by User
10	HIGH	10		COMPLETED	17/1/2022	Accept or Reject bookings mad by User
11	HIGH	10		COMPLETED	19/1/2022	Owner view the User staying in the PG
12	HIGH	10		COMPLETED	22/1/2022	User and Owner Add Complaints
13	HIGH	10		COMPLETED	24/1/2022	Add Review about the PG
14	HIGH	10		COMPLETED	27/1/2022	User Vacate their PG

User Story ID	Priority (Low,High, Medium)	Size	Sprint	Status (Planned, Progressed, Completed)	Release Date	Release Goal
15	HIGH	10	4	COMPLETED	31/1/2022	Owner view Rent Details
16	HIGH	10		COMPLETED	5/2/2022	User's view their stay details
17	HIGH	10		COMPLETED	10/2/2022	Owner View and Update complaint status
18	HIGH	10		COMPLETED	16/2/2022	Admin View and Update complaint status
19	HIGH	10		COMPLETED	20/2/2022	Owner View the Overall Status
20	HIGH	10		COMPLETED	22/2/2022	Admin View the Overall Status

Table 3.2 Product Backlog

3.5 Project plan

User story ID	Task name	Start date	End date	Days	Status Goal
Sprint 1					
1	Sprint1	14/12/2021	15/12/2021	1	COMPLETED
2	Sprint1	15/12/2021	20/12/2021	5	COMPLETED
3	Sprint1	20/12/2021	25/12/2021	5	COMPLETED
4	Sprint1	25/12/2021	28/12/2021	3	COMPLETED
Sprint 2					
5	Sprint1	28/12/2021	3/1/2022	6	COMPLETED
6	Sprint2	3/1/2021	5/1/2022	2	COMPLETED
7	Sprint2	5/1/2022	9/1/2022	4	COMPLETED
8	Sprint2	9/1/2022	13/1/2022	4	COMPLETED
Sprint 3					
9	Sprint2	13/1/2022	15/1/2022	2	COMPLETED
10	Sprint2	15/1/2022	17/1/2022	2	COMPLETED
11	Sprint3	17/1/2022	19/1/2022	2	COMPLETED
12	Sprint3	19/1/2022	22/1/2022	3	COMPLETED
13	Sprint3	22/1/2022	24/1/2022	2	COMPLETED
14	Sprint3	24/1/2022	27/1/2022	3	COMPLETED

User story ID	Task name	Start date	End date	Days	Status Goal
Sprint 4					
15	Sprint3	27/1/2022	31/1/2022	4	COMPLETED
16	Sprint4	31/1/2022	5/2/2022	5	COMPLETED
17	Sprint4	5/2/2022	10/2/2022	5	COMPLETED
18	Sprint4	10/2/2022	16/2/2022	6	COMPLETED
19	Sprint4	16/2/2022	20/2/2022	4	COMPLETED
20	Sprint4	20/2/2022	22/2/2022	2	COMPLETED

Table 3.3 Project plan

3.6 Database Design

3.6.1 Owner

This is the Owner registration table.

No.	Name	Type	Constraints	Description
1	owner_id	INT	PRIMARY KEY	Registration id of Owner
2	owner_name	VARCHAR(30)	NOT NULL	Name of Owner
3	id_proof	BLOB	NOT NULL	Username of Owner
4	address	VARCHAR(50)	NOT NULL	Address of Owner
5	email_id	VARCHAR(30)	NOT NULL	Email-id of Owner
6	phone_no	VARCHAR(10)	NOT NULL	Phone number of Owner

Table 3.4.1 Owner table

3.6.2 User

This is User registration table.

No.	Name	Type	Constraints	Description
1	user_id	INT	PRIMARY KEY	Registration id of User
2	username	VARCHAR(15)	NOT NULL	Name of User
3	email	VARCHAR(10)	NOT NULL	Email of User
4	address	VARCHAR(50)	NOT NULL	Address of User
5	phone_no	VARCHAR(10)	NOT NULL	Phone number of User
6	gender	VARCHAR(10)	NOT NULL	Genderr of User
7	id_proof	BLOB	NOT NULL	Id proof of User
8	profile_photo	BLOB	NOT NULL	Profile Photo of User

Table 3.4.2 User table

3.6.3 Login

This is the Login table for Owner, Admin and User.

No.	Name	Type	Constraints	Description
1	l_id	INT	PRIMARY KEY	Login id of user
2	username	VARCHAR(15)	NOT NULL	username of user
3	password	VARCHAR(15)	NOT NULL	password given
4	user_type	VARCHAR(10)	NOT NULL	type of user

Table 3.4.3 Login table

3.6.4 Pg_details

This is the PG details table

No.	Name	Type	Constraints	Description
1	pg_id	INT	PRIMARY KEY	PG id
2	owner_id	INT	FOREIGN KEY	Registration id of Owner
3	pg_name	VARCHAR(30)	NOT NULL	PG name
4	facility_type	VARCHAR(50)	NOT NULL	Facilities provided
5	location	VARCHAR(20)	NOT NULL	Location of PG
6	address	VARCHAR(30)	NOT NULL	Address of PG
7	pg_photo1	BLOB	NOT NULL	Photo of PG
8	pg_photo2	BLOB	NOT NULL	Photo of PG
9	pg_photo3	BLOB	NULL	Photo of PG
10	rent	INT	NOT NULL	Rent amount of PG

Table 3.4.4 PG details table

3.6.5 Booking

This is the Bookings table.

No.	Name	Type	Constraints	Description
1	booking_id	INT	PRIMARY KEY	Id of booking
2	pg_id	INT	FOREIGN KEY	PG id
3	user_id	INT	FOREIGN KEY	Id of user
4	purpose	VARCHAR(50)	NOT NULL	Purpose of User
5	duration	VARCHAR(15)	NOT NULL	Duration of stay
6	date	DATE	NOT NULL	Date of stay
7	status	VARCHAR(15)	NOT NULL	Status of Booking

Table 3.4.5 Booking table

3.6.6 User_complaint

This is the User Complaints table. It includes details of posted complaints by Users

No.	Name	Type	Constraints	Description
1	compliant_id	INT	PRIMARY KEY	Id of complaint
2	pg_id	INT	FOREIGN KEY	PG id
3	user_id	INT	FOREIGN KEY	Id of user
4	complaint	VARCHAR(30)	NOT NULL	Complaint in Words
5	status	VARCHAR(15)	NOT NULL	Current status of the complaint

Table 3.4.6 User Complaints table

3.6.7 Owner_complaint

This is the Owner Complaints table. It includes details of posted complaints by Owners

No.	Name	Type	Constraints	Description
1	compliant_id	INT	PRIMARY KEY	Id of complaint
2	user_id	INT	FOREIGN KEY	Id of user
3	complaint	VARCHAR(30)	NOT NULL	Complaint in Words
4	status	VARCHAR(15)	NOT NULL	Current status of the complaint

Table 3.4.7 Owner Complaints table

3.6.8 Review

This is the Review table. It includes details about reviews posted by User about PGs

No.	Name	Type	Constraints	Description
1	review_id	INT	PRIMARY KEY	Id of Review
2	pg_id	INT	FOREIGN KEY	PG id
3	user_id	INT	FOREIGN KEY	Id of user
4	review_words	VARCHAR(30)	NOT NULL	Review in Words

Table 3.4.8 Review table

3.7 Forms

3.7.1 Login

This is the login form of User, Admin and Owner.

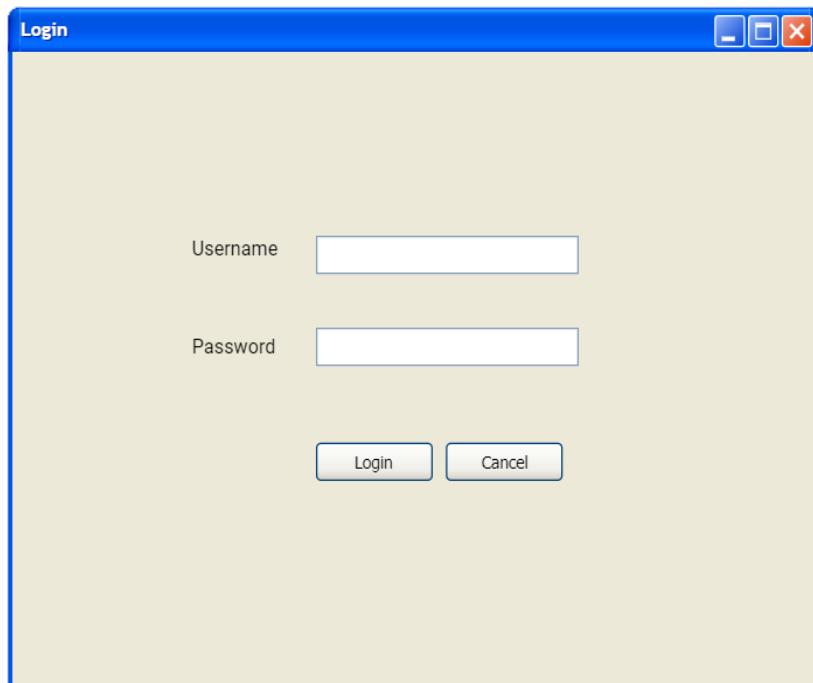
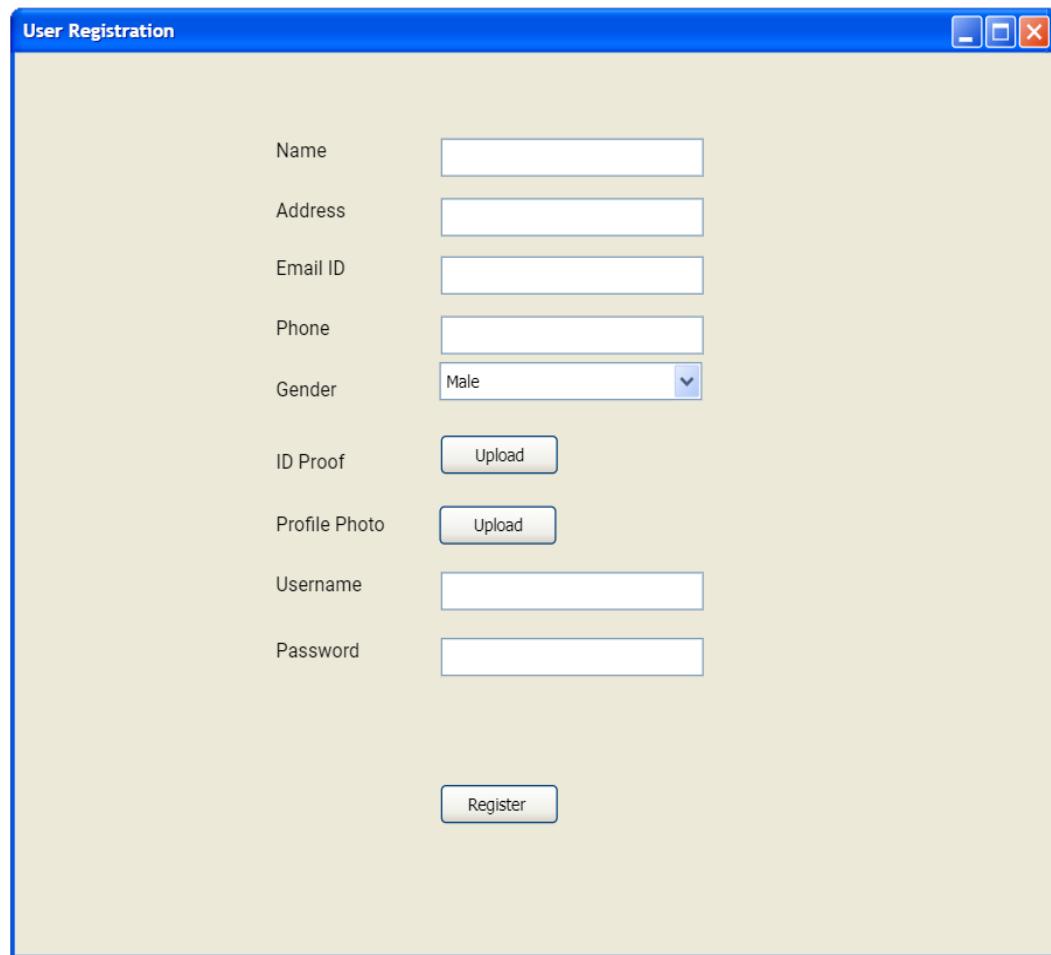


Figure 3.5: Login form

3.7.2 User Registration

This is the registration form of user.



The image shows a Windows-style application window titled "User Registration". The window has a blue title bar and standard window controls (minimize, maximize, close) in the top right corner. The main area is a light beige color and contains the following fields:

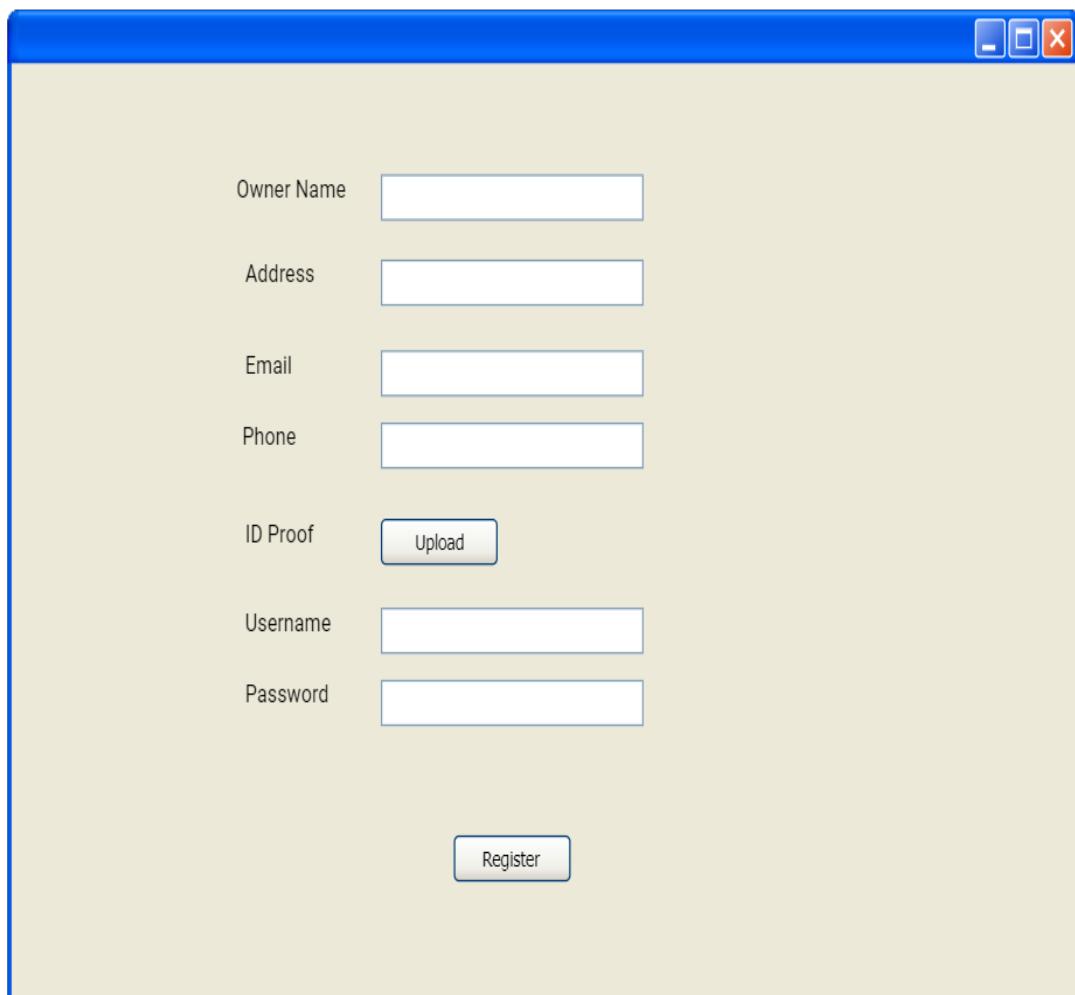
Name	<input type="text"/>
Address	<input type="text"/>
Email ID	<input type="text"/>
Phone	<input type="text"/>
Gender	Male <input type="button" value="▼"/>
ID Proof	<input type="button" value="Upload"/>
Profile Photo	<input type="button" value="Upload"/>
Username	<input type="text"/>
Password	<input type="text"/>

At the bottom center of the window is a blue "Register" button.

Figure 3.6: Registration form

3.7.3 Owner Registration

This is the registration form of Owner.



The image shows a Windows-style application window titled "Owner Registration". The window has a blue header bar with standard minimize, maximize, and close buttons. The main area is a light beige color and contains the following fields:

- Owner Name: An input field.
- Address: An input field.
- Email: An input field.
- Phone: An input field.
- ID Proof: A label followed by an "Upload" button.
- Username: An input field.
- Password: An input field.

At the bottom center of the window is a blue "Register" button.

Figure 3.7: Registration form

3.7.4 Search PGs

Search PGs.

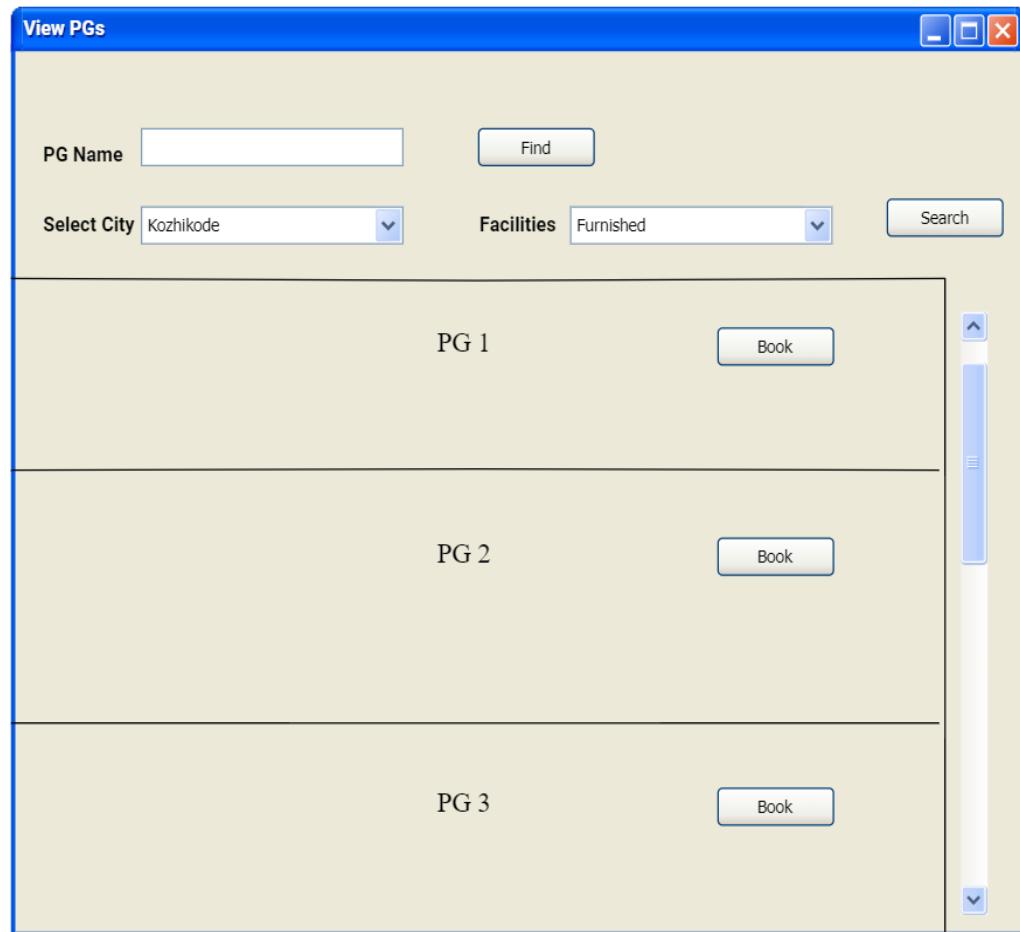


Figure 3.8: Search PG

3.7.5 View PG Details

View the full details about th PG.

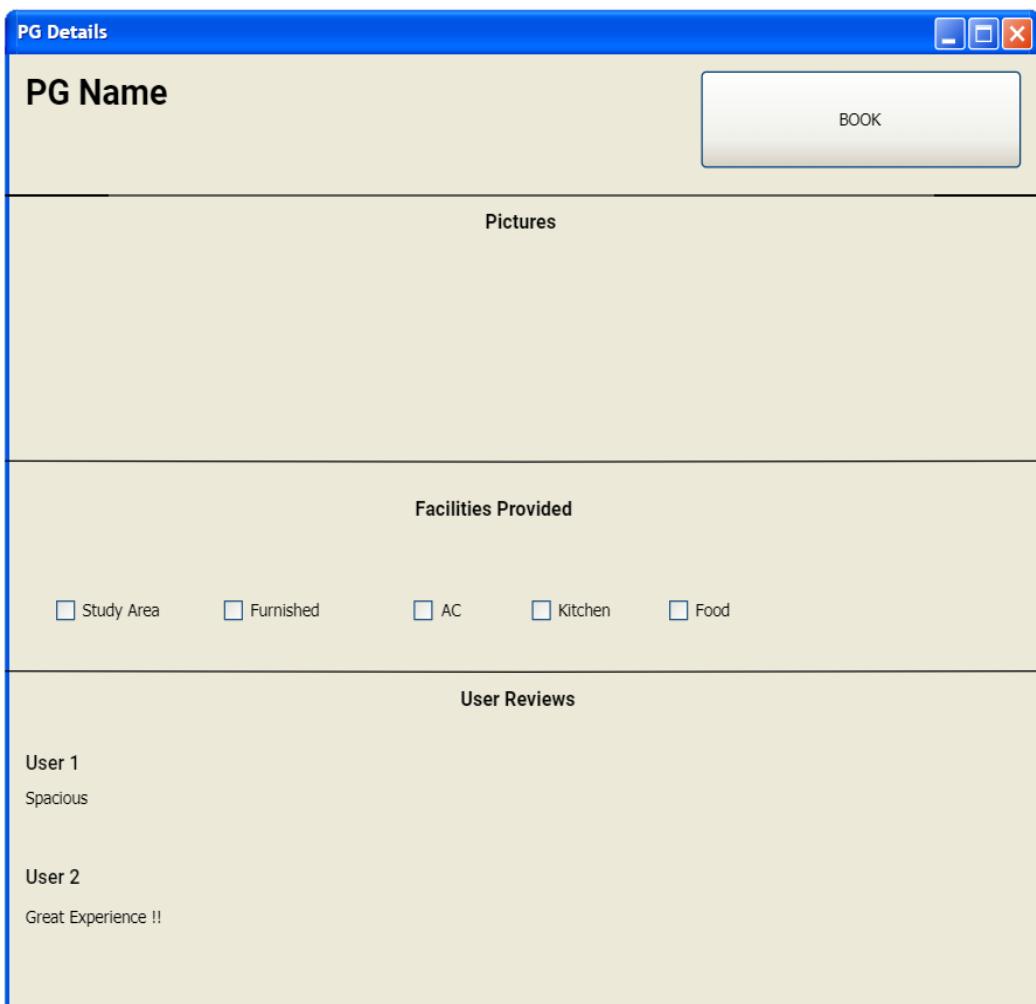


Figure 3.9: View PG Details

3.7.6 View User Profile

View and Update User details.

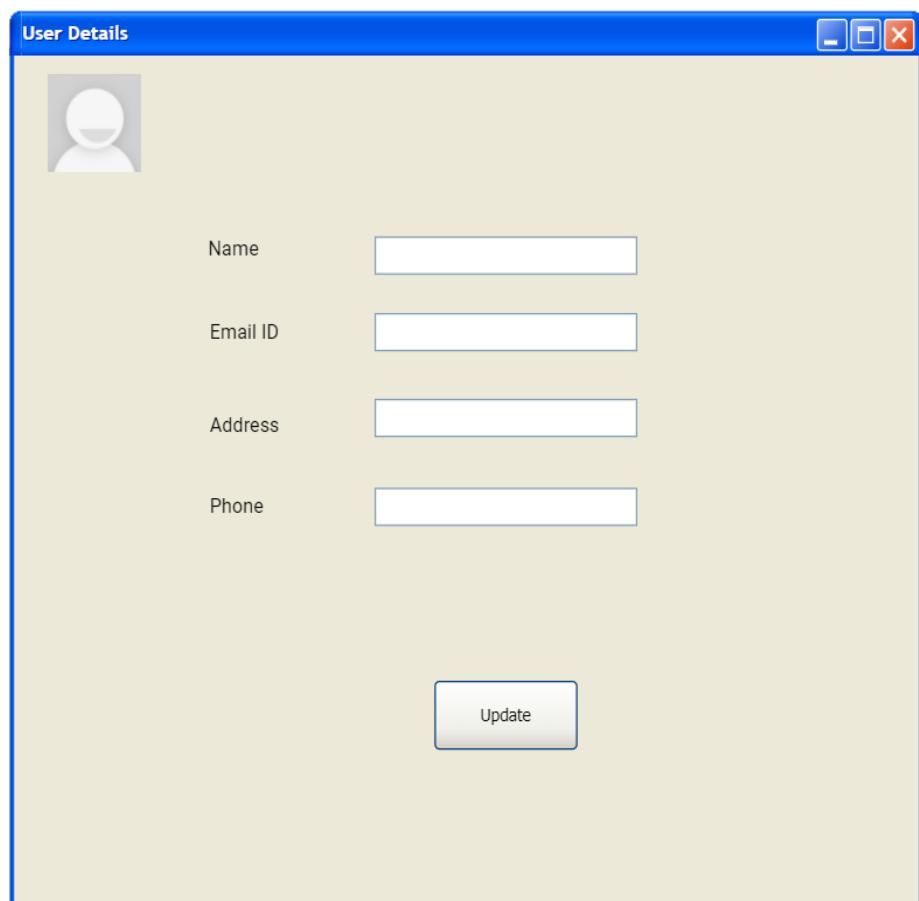


Figure 3.10: View and Edit User's Details

3.7.7 View Current Stay Details

View the overall details of stay in the current PG.

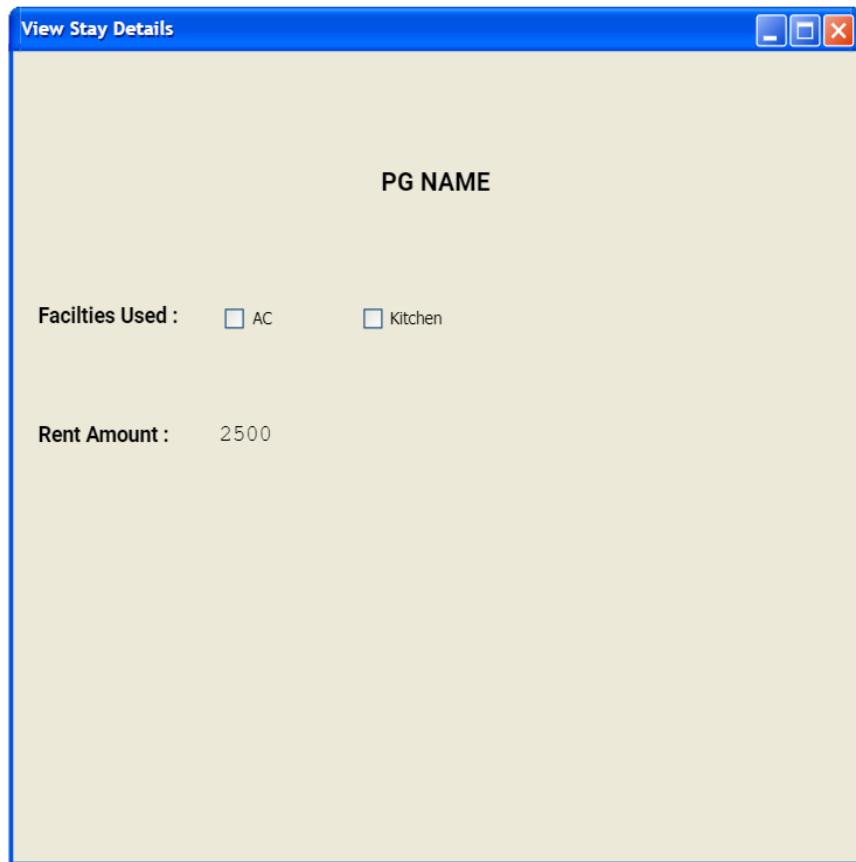


Figure 3.11: View Stay Details

3.7.8 Add Complaint

Add Complaint to Owner/Admin

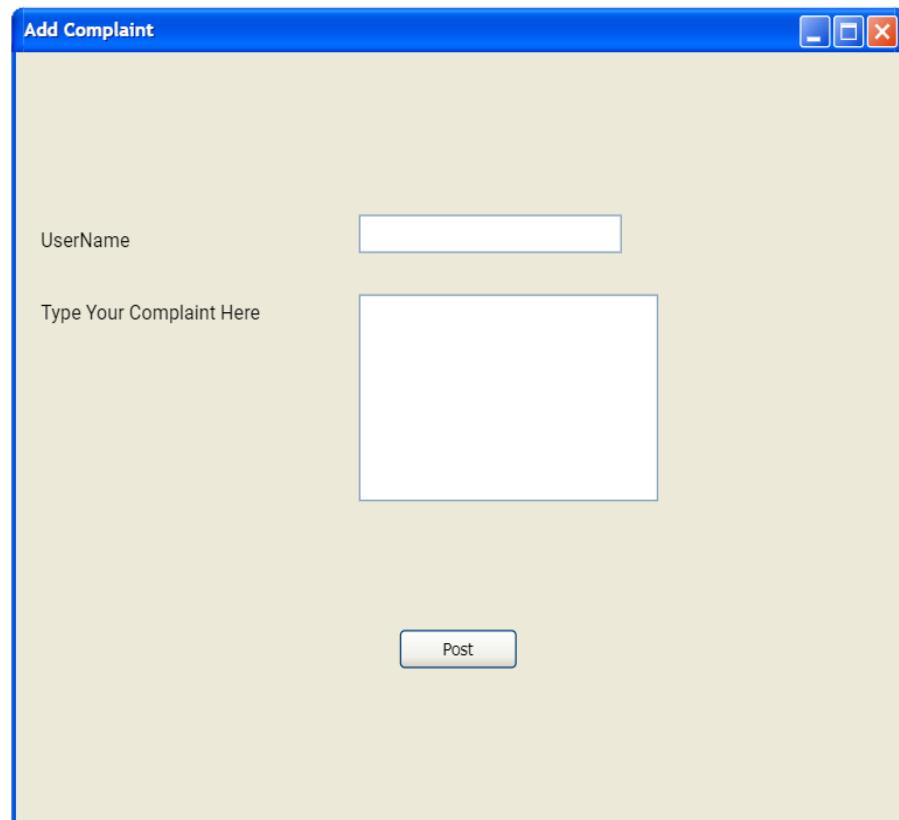


Figure 3.12: Add Complaints

3.7.9 View and Respond to Complaints

View the complaints posted and reply to them



Figure 3.13: Respond to Complaints

3.7.10 Admin Home

Admins Homepage

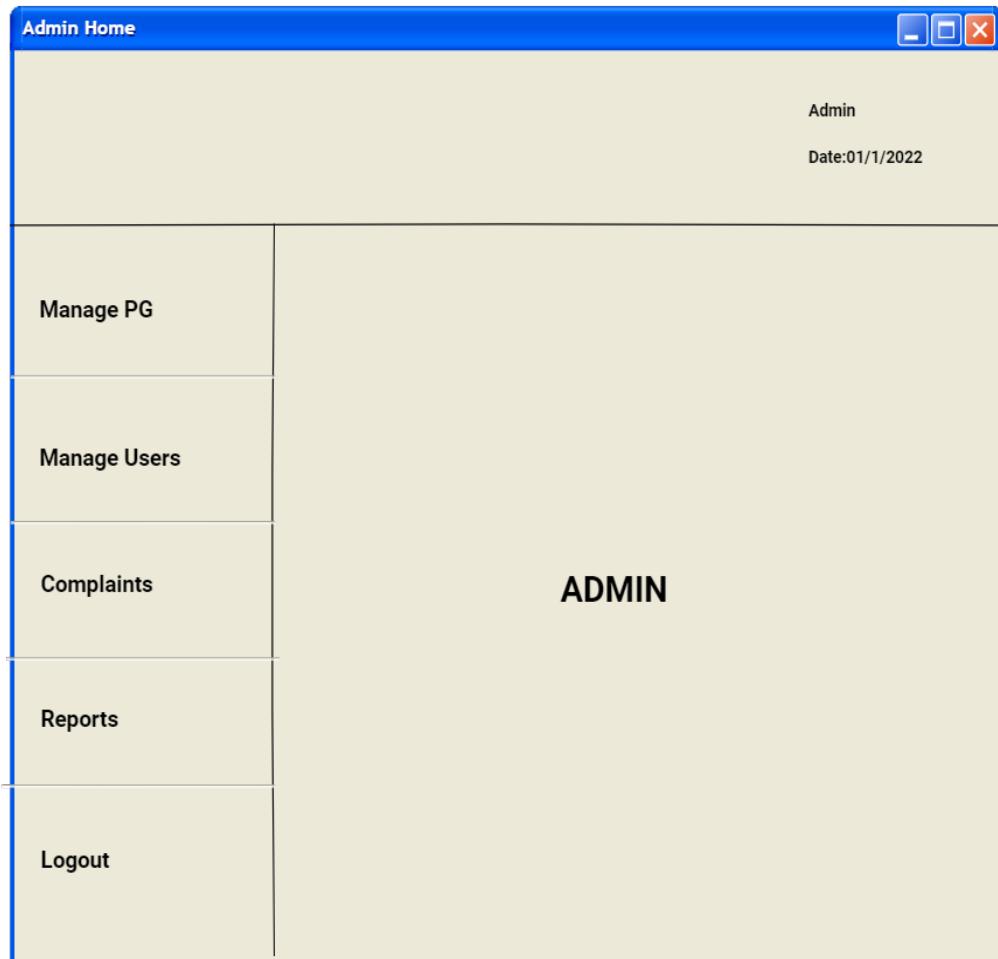


Figure 3.14: Admin Homepage

3.7.11 User Home

User's Homepage

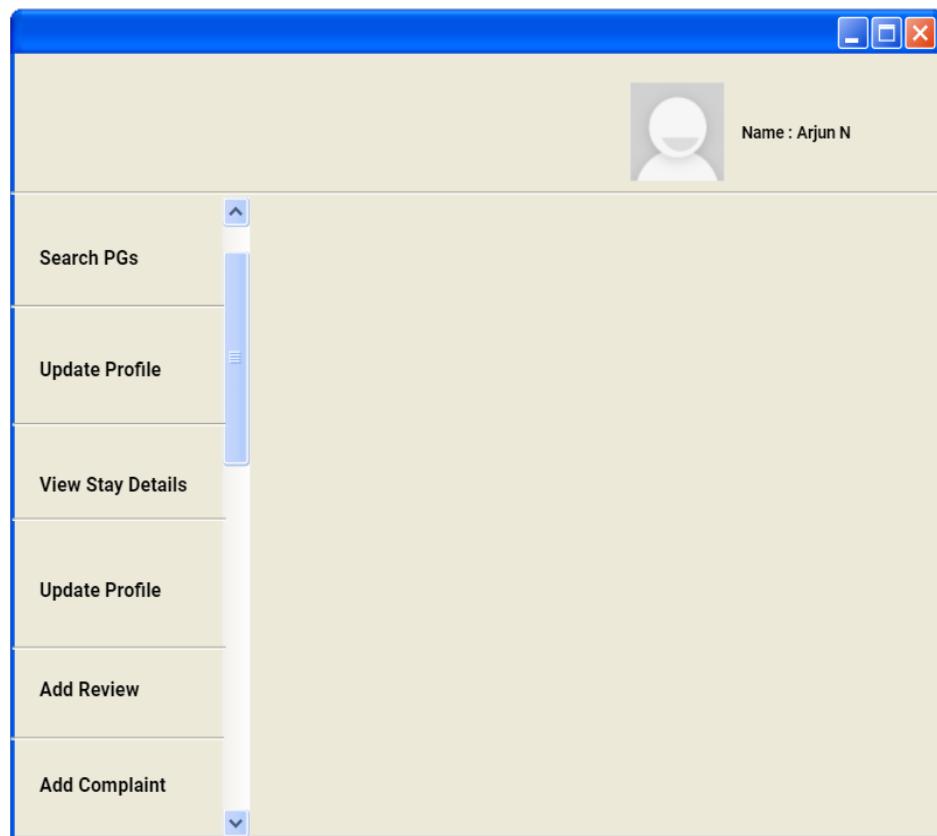


Figure 3.15: User Homepage

3.7.12 Owner Home

Owner's Homepage

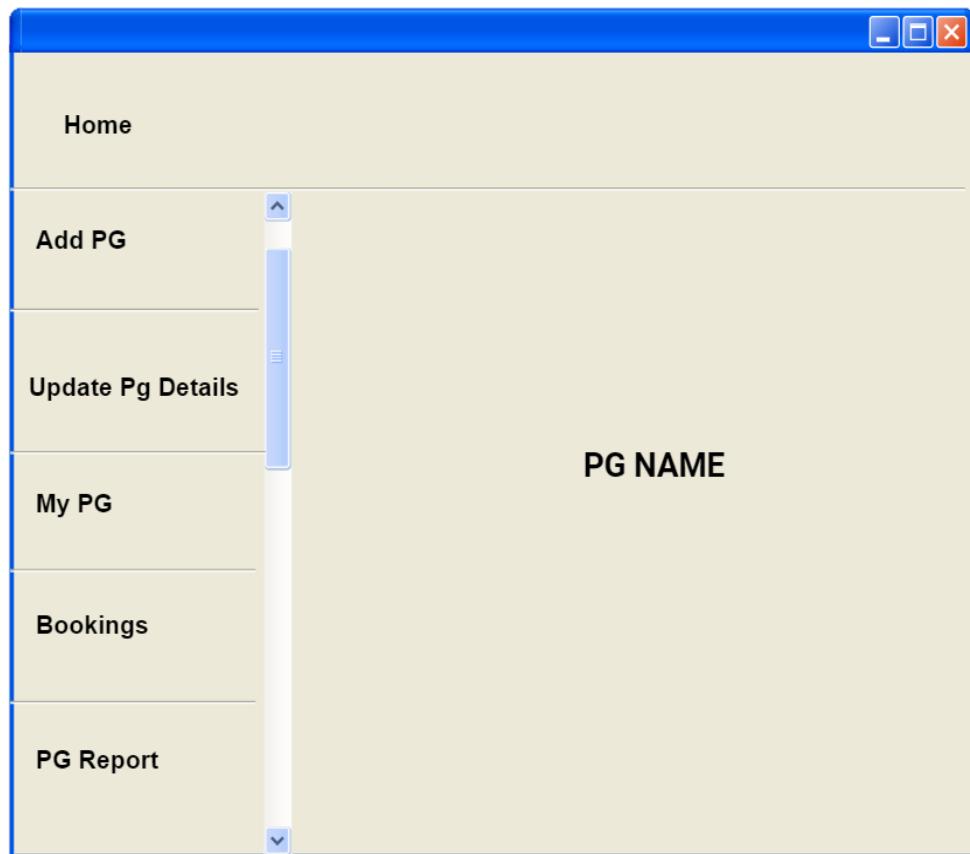


Figure 3.16: Owner Homepage

3.7.13 Accept or Reject Bookings

Accept or Reject Bookings made by Users

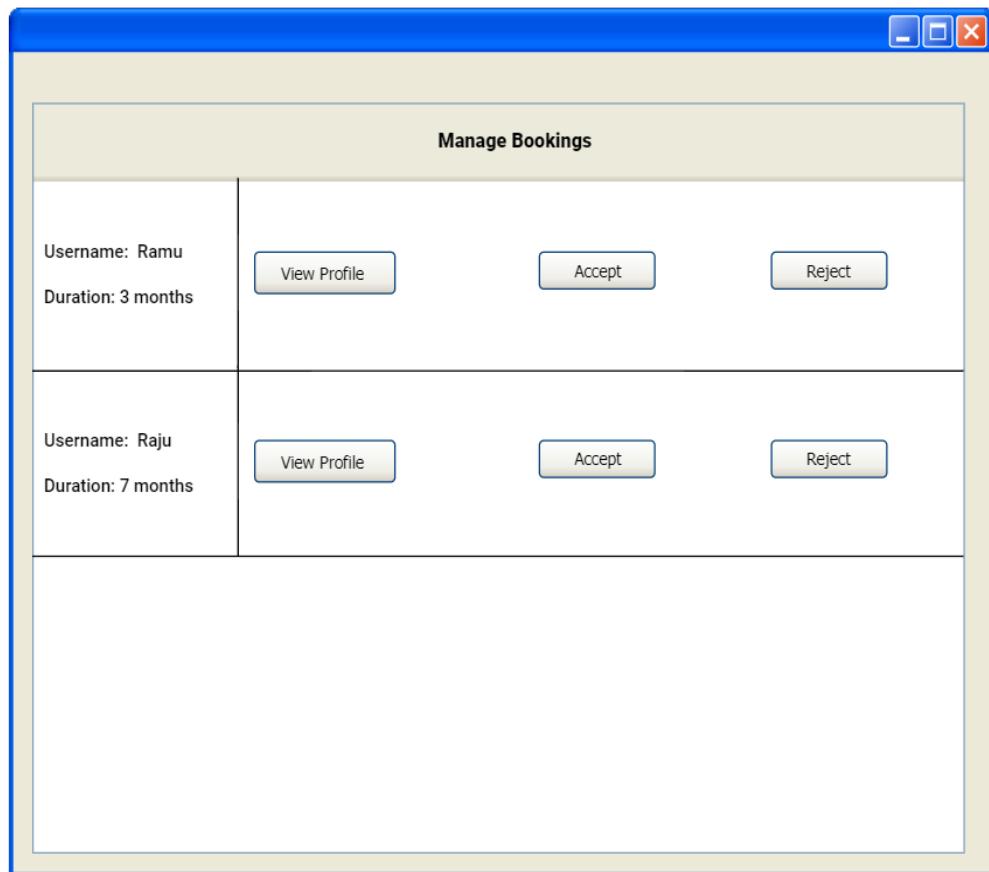


Figure 3.17: Accept/Reject Bookings

3.7.14 Admin Manage PGs

Admin Manages full PGs

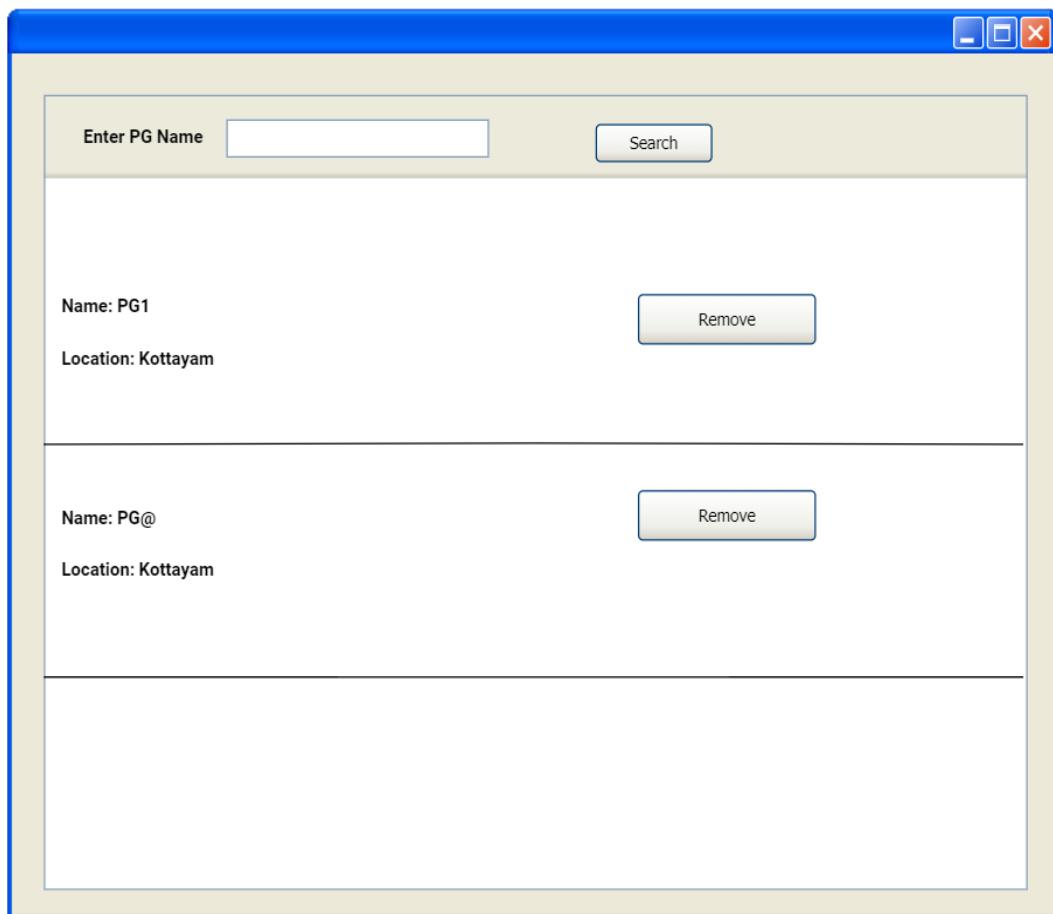


Figure 3.18: PG Managing by admin

3.7.15 Admin Manage User's

Admins Manages whole User's

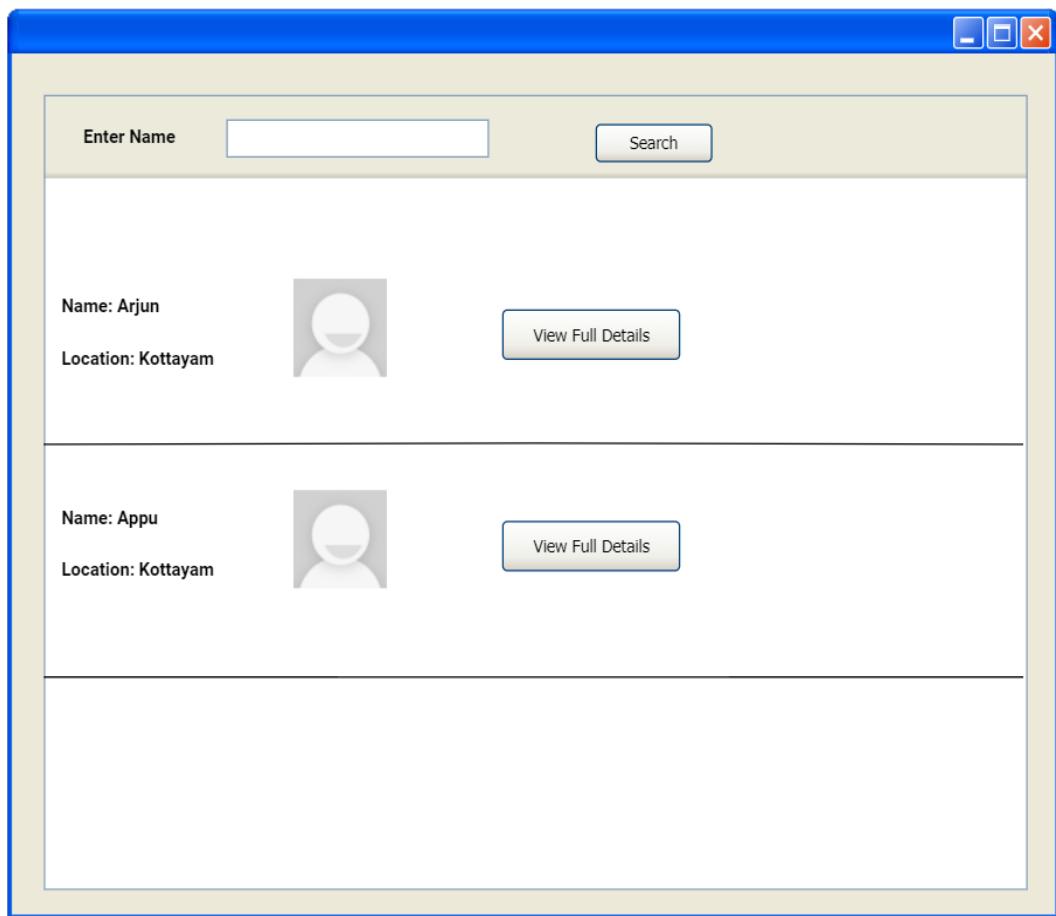


Figure 3.19: Admin Manage User

3.7.16 Owner Add PG

Owners can Register their PG

The screenshot shows a Windows-style application window titled "Owner Add PG". The window has a blue header bar with standard minimize, maximize, and close buttons. The main area contains the following fields:

- "PG Name" followed by a text input field.
- "Facilities Provided" with checkboxes for "Furnished", "Kitchen", "Study Area", "AC", and "Food".
- "Location" followed by a text input field.
- "Addresses" followed by a text input field.
- "Photo1" followed by an "Upload" button.
- "Photo1" followed by an "Upload" button.
- "Photo1" followed by an "Upload" button.
- "Rent" followed by a text input field.
- A large "Register" button at the bottom center.

Figure 3.20: Owners PG Registration

3.7.17 Owner PG Report

Owners can view report about their PG

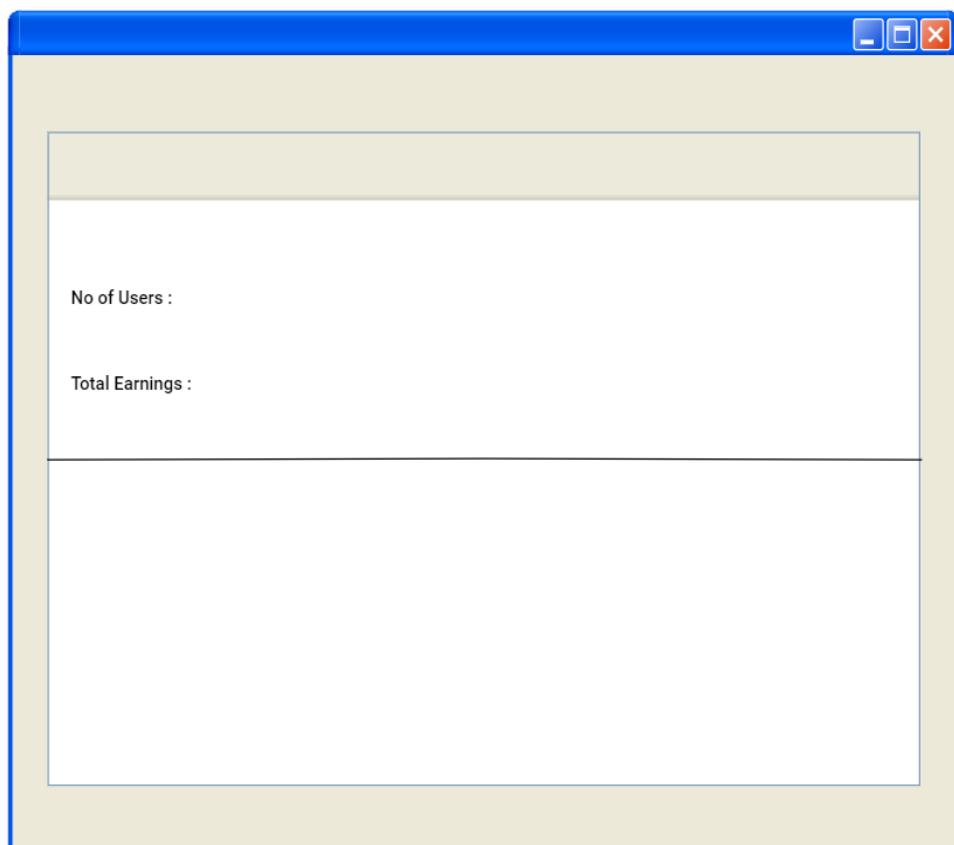


Figure 3.21: Owner PG Report

3.7.18 Add Review

Add Reviews by Users



Figure 3.22: Add Reviews

3.7.19 Owner Update PG details

Owners update their details about PG

The screenshot shows a window titled "Update PG Details". The form contains the following fields:

- PG Name: An input field.
- Facilities: A group of checkboxes for "Furnished", "Kitchen", "AC", "Study Area", and "Food".
- Location: An input field.
- Address: An input field.
- Photo1: An input field with an "Upload" button.
- Photo2: An input field with an "Upload" button.
- Photo3: An input field with an "Upload" button.
- Rent: An input field.
- Submit: A large blue "Submit" button at the bottom.

Figure 3.23: Update PG Details

Chapter 4

TESTING AND IMPLEMENTATION

4.1 Testing

4.1.1 Test Case-1

No.	Date	Action	Expected Result	Actual Result	Pass?
1	15-12-21	Homepage of All Users	All Users can view Homepage	All Users can view Homepage	Yes
2	20-12-21	Login of Admin	Admin can Login	Admin can Login	Yes
3	25-12-21	Registration of User,Owner	User and Owner can Register	User and Owner can Register	yes
4	28-12-21	Login for Owner and User	Owner and User can Login	Owner and User can Login	Yes

Table 4.1: Test case-1

4.1.2 Test Case-2

No.	Date	Action	Expected Result	Actual Result	Pass?
5	3-1-22	Manage PGs	Admin can manage PGs	Admin can manage PGs	Yes
6	5-1-22	View Users	Admin can view User's	Admin can view User's	Yes
7	9-1-22	Add and Update PG	Owner can add and update PG	Owner can add and update PG	Yes
8	13-1-22	View PG	User's can view PG Details	User's can view PG Details	Yes

Table 4.2: Test case-2

4.1.3 Test Case-3

No.	Date	Action	Expected Result	Actual Result	Pass?
9	19-01-22	Book PG	User can book a PG after selecting one	User can book a PG after selecting one	Yes
10	21-01-22	Accept or Reject Bookings	Owner can accept/reject bookings made by user's	Owner can accept/reject bookings made by user's	Yes
11	23-01-22	View PG details	View the Users staying in their PG	View the Users staying in their PG	Yes
12	26-01-22	Add Complaints	Owner add complaints to the Admin. User add Complaints to the Owner	Owner add complaints to the Admin. User add Complaints to the Owner	Yes
13	29-01-22	Add Reviews	User add review about the PG they are currently staying	User add review about the PG they are currently staying	Yes
14	29-01-22	Vacate PG	User Vacate from the PG they are currently staying	User Vacate from the PG they are currently staying	Yes

Table 4.3: Test case-3

4.1.4 Test Case-4

No.	Date	Action	Expected Result	Actual Result	Pass?
15	23-01-22	View Rent details	View the rents of each users staying in their PG	Deliveryboy able to accept the orders	Yes
16	26-01-22	View Users Stay Details	Show the details like Rent and No of days stayed	Show the details like Rent and No of days stayed	Yes
17	29-01-22	Owner View and Update Complaint Status	View and Update the status of complaints posted by the User	View and Update the status of complaints posted by the User	Yes
18	02-02-22	Admin View and Update Complaint Status	View and Update the status of complaints posted by the Owner	View and Update the status of complaints posted by the Owner	Yes
19	22-02-22	Owner View Status of PG	Owner View the Overall details of their PG	Owner View the Overall details of their PG	Yes
20	25-02-22	Admin View System Status	Admin View the total details of System	Admin View the total details of System	Yes

Table 4.4: Test case-4

4.2 Implementation

After testing, the next phase is Implementation. The proposed system is ready for the implementation. Implementation is the realization of an app or execution of a plan, ideas, model, design, specification, standard, algorithm or policy. System implementation phase is the phase, involves the process of converting a new system design into an operational one.

To implement this system, Admin need a personal computer to host an web application. For hosting an web application, first of all choose web hosting provider means web domain for this project. This web hosting provider provides web space (which means web servers) which store website files of project, as well as technologies and services that are needed for website to be viewed on the internet. Stored file can't exceed the size. Project's website can be host in "http://in.000webhost.com". This site is free to host and it has free and paid version. Free versions have some limitation to access and need to pay an dollar amount for paid version. The website uploaded to server can be accessed by anyone with a particular domain name.

Chapter 5

RESULT AND DISCUSSION

The project Paying Guest Management System was developed with proper planning and guidance. Agile methodology is used during the development of this project. Planning at each stage was done properly. Each sprint has been conducted as per protocol. Testing was performed at each stage of development. The project is meant to manage Paying Guest homes in Kozhikode through web application. Web application is managed by Admin and the Owners register their PGs through this and the User can book one of them after selecting the suitable one.

Chapter 6

CONCLUSION

The project "Paying Guest Management System " gives users an interface to find the best suitable PG of their choice with less effort,user can easily book one by searching the location that they would like to find a PG.Owners who would like to rent their homes for an income can easily register their home as a PG ,they can manage the bookings received from the Users by using this system.Owner will be able to select the best tenant with the provided facilities.Owners and Users can use the web application for Booking a PG and Renting their PGs easily through this system.

6.1 References

- [1] https://www.w3schools.com/howto/howto_css_login_form.asp
- [2] <https://github.com>
- [3] <https://www.w3schools.com/php>
- [4] <https://www.w3schools.com/css>
- [5] <https://www.mysqltutorial.org/>

6.2 Appendix

6.2.1 Source Code

- **PG Booking**

```
<?php
set connection
include('dbconnection.php');
session_start();
$uid=$_SESSION['UID'];
if(isset($_GET['id']))
{
    $idd=$_GET['id'];
    $_SESSION['pgid']=$idd;
    $sql="select * from `pddet` where pgid='$idd'";
    $result=$con->query($sql);
    $row=mysqli_fetch_assoc($result);
}

if(isset($_POST['submit']))
{
    //inserting datas to table
    $purpose=$_POST['purpose'];
    $duration=$_POST['duration'];
    $date=$_POST['date'];
    $uid=$_SESSION['UID'];
    $pgid=$_SESSION['pgid'];
    $sql="INSERT INTO `booking` ('pgid', 'uid', 'purpose',
    'duration', 'date', 'status') VALUES ('$pgid', '$uid',
    '$purpose', '$duration', '$date', 'Booked')";
}
```

```

$result=$con->query($sql);

?>

<script type="text/javascript">
alert("Booked Successfully");
Location.href="user_home.php";
</script>
<?php
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
<link rel="icon" href="images/icons/favicon.png"/>
<title>vacayhome</title>
<link href="css/bootstrap.min.css" rel="stylesheet">
<link href="font-awesome/css/font-awesome.min.css"
rel="stylesheet" type="text/css" />
<link href="css/style.css" rel="stylesheet">
<link href="fonts/antonio-exotic/stylesheet.css"
rel="stylesheet">
<link rel="stylesheet" href="css/lightbox.min.css">
<link href="css/responsive.css" rel="stylesheet">
<script src="js/jquery.min.js"
type="text/javascript"></script>
<script src="js/bootstrap.min.js"

```

```
type="text/javascript">></script>
<script src="js/lightbox-plus-jquery.min.js"
type="text/javascript">></script>
<script src="js/instafeed.min.js"
type="text/javascript">></script>
<script src="js/custom.js"
type="text/javascript">></script>
</head>
<body>
<div id="page">
<div class="top-header">
<div class="container">
<div class="row">
<div class="col-md-6">

</div>
<div class="col-md-6"><div class="social-grid">
<ul class="list-unstyled text-right">
<li><a><i class="fa fa-facebook"></i></a></li>
<li><a><i class="fa fa-twitter"></i></a></li>
<li><a><i class="fa fa-linkedin"></i></a></li>
<li><a><i class="fa fa-instagram"></i></a></li>
</ul>
</div></div>
</div>
</div>
</div>

<header class="header-container">
<div class="container">
```

```
<div class="top-row">
<div class="row">
    <div class="col-md-2 col-sm-6 col-xs-6">
        <div id="logo">
            <a href="index.html"><span>New</span>home</a>
        </div>
    </div>
    <div class="col-sm-6 visible-sm">
        <div class="text-right"><button type="button" class="book-now-btn">Book Now</button></div>
    </div>
    <div class="col-md-8 col-sm-12 col-xs-12 remove-padd">
        <nav class="navbar navbar-default">
            <div class="navbar-header page-scroll">
                <button data-target=".navbar-ex1-collapse" data-toggle="collapse" class="navbar-toggle" type="button">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
            </div>
            <?php include 'usermenu.php' ?>
        </nav>
    </div>
    </div>
    </div>
</div>
```

```

</header>

<div class="clearfix"></div>

<section class="contact-block">
<div class="container">
<div class="col-md-6 contact-form">
<h3>Book<span>Now</span></h3>
<form method="post" enctype="multipart/form-data">
<table class="table table-bordered" id="dataTable"
cellspacing="0">
<tr>
<td>Purpose</td>
<td><input type="text" class="form-control"
name="purpose" required=""></td>
</tr>
<tr>
<td>Duration of Your Stay</td>
<td><input type="text" class="form-control"
name="duration" placeholder="e.g. 3 Months, 1 Year"
required=""></td>
</tr>
<tr>
<td>Check-In Date</td>
<td><input type="date" class="form-control"
name="date" min='<?php echo date('Y-m-d'); ?>'
required=""></td>
</tr>
<tr>
<td></td>
<td><input type="submit" class="submit-btn"
name="submit"></td>

```

```

        </tr>
    </table>
</form>

</div>
<div class="clearfix"></div>
</div>
</section>
<section class="gallery-block gallery-front">
<div class="container">
<div class="row">
    <h1><b><?php echo $row['name'] ?></b></h1>
<div class="col-lg-3 col-md-3 col-sm-3 col-xs-12">
    <div class="gallery-image">
        
        <div class="overlay">
            <a class="info pop example-image-link
img-responsive"
href="<?php echo $row['photo1'] ?>"
data-lightbox="example-1"><i class="fa fa-search"
aria-hidden="true"></i></a>
        </div>
    </div>
</div>
</div>

<div class="col-lg-3 col-md-3 col-sm-3 col-xs-12">
    <div class="gallery-image">
        
        <div class="overlay">

```

```

<a class="info pop example-image-link img-responsive"
  href="<?php echo $row['photo2'] ?>" data-lightbox="example-1"><i class="fa fa-search" aria-hidden="true"></i></a>
</div>
<div>
</div>
<div class="col-lg-3 col-md-3 col-sm-3 col-xs-12">
<div class="gallery-image">

<div class="overlay">
<a class="info pop example-image-link img-responsive" href="<?php echo $row['photo3'] ?>" data-lightbox="example-1"><i class="fa fa-search" aria-hidden="true"></i></a>
</div>
</div>
</div>

<h3><b>Rent:</b><?php echo $row['rent'] ?>/month</h3><br>
<h3><b>Location:</b><?php echo $row['location'] ?></h3><br>
<h3><b>Address:</b><?php echo $row['address'] ?></h3><br>
<h3><b>Contact No:</b><?php echo $row['pgphone'] ?></h3>
</div>
</div>
<section>
<?php echo $row['photol'] ?>" class="img-responsive" alt="gallery1"></a>
</div>
<div class="grid-item percent25 gallery-image">

```

```

<a class="example-image-link
img-responsive" href="<?php echo $row['photo2'] ?>" data-lightbox="example-1"></a>
</div>

<div class="grid-item percent37 gallery-image">
<a class="example-image-link img-responsive"
href="<?php echo $row['photo3'] ?>" data-lightbox="example-1"></a>
</div>
</div>
</div>
</div>
</div>
</section>

<footer>
<div class="container">
<div class="row">
<div class="col-md-3 col-sm-6 col-xs-12 width-set-50">
<div class="footer-details">
</div>
</div>
<div class="col-md-3 col-sm-6 col-xs-12 width-50 width-set-50">
</div>
<div class="col-md-6 col-sm-6 col-xs-12">

```

```
</div>
</div>
<div>
</footer>
<a style="display: none;" href="javascript:void(0);"
class="scrollTop back-to-top" id="back-to-top">
<span><i aria-hidden="true"
class="fa fa-angle-up fa-lg"></i></span>
<span>Top</span>
</a>
</div>
</body>
</html>
```

- **Add PG**

```
<?php
//set connection
include('dbconnection.php');
session_start();
$uid=$_SESSION['UID'];
if(isset($_POST['submit']))
{
//inserting data to table
$filename= basename($_FILES["file1"]["name"]);
$ext = pathinfo($filename, PATHINFO_EXTENSION);
$fnn=date("YmdHis")."1.".$ext;
$target_dir = "uploads/";
$target_file = $target_dir.$fnn;
if (move_uploaded_file($_FILES["file1"]["tmp_name"],
$target_file)) {
;
}
else
{
$filename1= basename($_FILES["file2"]["name"]);
$ext1 = pathinfo($filename1, PATHINFO_EXTENSION);
$fnn1=date("YmdHis")."2.".$ext1;
$target_dir1 = "uploads/";
$target_file1 = $target_dir1.$fnn1;
if (move_uploaded_file($_FILES["file2"]
["tmp_name"], $target_file1)) {
}
else
```

```

{
}

$filename2= basename($_FILES["file3"]["name"]);
$ext2 = pathinfo($filename2, PATHINFO_EXTENSION);
$fnn2=date("YmdHis")."3.".$ext2;
$target_dir2 = "uploads/";
$target_file2 = $target_dir2.$fnn2;
if (move_uploaded_file($_FILES["file3"]["tmp_name"],
$target_file2)) {
}
else
{
}
$name=$_POST['name'];
$address=$_POST['address'];
$location=$_POST['location'];
$rent=$_POST['rent'];
$fac="";
if (isset($_POST['fac1']))
{
$fur=$_POST['fac1'];
$fac="$fac $fur";
}
if (isset($_POST['fac2']))
{
$kit=$_POST['fac2'];
$fac="$fac $kit";
}
if (isset($_POST['fac3']))
{
}

```

```

$sarea=$_POST['fac3'];
$fac="$fac $sarea";
}

if (isset($_POST['fac4']))
{
$ac=$_POST['fac4'];
$fac="$fac $ac";
}

if (isset($_POST['fac5']))
{
$food=$_POST['fac5'];
$fac="$fac $food";
}

$sql="INSERT INTO `pdet` ( `name`, `address`,
`location`, `photo1`, `photo2`, `photo3`,
`rent`, `owid`, `facilities`) VALUES ('$name',
'$address', '$location', '$target_file',
'$target_file1', '$target_file2',
'$rent', '$uid','$fac')";

$result=$con->query($sql);
}

?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible"
content="IE=edge">
<meta name="viewport"

```

```
content="width=device-width,
initial-scale=1.0">
<link rel="icon"
href="images/icons/favicon.png"/>
<title>vacayhome</title>
<link href="css/bootstrap.min.css"
rel="stylesheet">
<link href="font-awesome/css/font-a
wesome.min.css" rel="stylesheet" type="text/css" />
<link href="css/style.css"
rel="stylesheet">
<link href="fonts/antonio-exotic/stylesheet.css" rel="stylesheet">
<link rel="stylesheet" href="css/lightbox.min.css">
<link href="css/responsive.css"
rel="stylesheet">
<script src="js/jquery.min.js"
type="text/javascript"></script>
<script src="js/bootstrap.min.js"
type="text/javascript"></script>
<script src="js/lightbox-plus-jquery.min.js"
type="text/javascript"></script>
<script src="js/instafeed.min.js"
type="text/javascript"></script>
<script src="js/custom.js"
type="text/javascript"></script>
</head>
<body>
<div id="page">
<div class="top-header">
<div class="container">
```

```
<div class="row">
<div class="col-md-6">
</div>

<div class="col-md-6"><div class="social-grid">
<ul class="list-unstyled text-right">
<li><a><i class="fa fa-facebook"></i></a></li>
<li><a><i class="fa fa-twitter"></i></a></li>
<li><a><i class="fa fa-linkedin"></i></a></li>
<li><a><i class="fa fa-instagram"></i></a></li>
</ul>
</div></div>
</div>
</div>
</div>

<header class="header-container">
<div class="container">
<div class="top-row">
<div class="row">
<div class="col-md-2 col-sm-6 col-xs-6">
<div id="logo">
<a href="index.html"><span>New</span>home</a>
</div>
</div>
<nav class="navbar navbar-default">
<div class="navbar-header page-scroll">
<button data-target=".navbar-ex1-collapse"
data-toggle="collapse" class="navbar-toggle"
type="button">
<span class="sr-only">Toggle navigation</span>
```

```

<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
</div>

<?php include 'ownermenu.php' ?>
</nav>
</div>
</div>
</div>
</div>
</div>
</header>

<div class="clearfix"></div>
<section class="contact-block">
<div class="container">
<div class="col-md-6 contact-form">
<h3>Add<span>PG</span></h3>
<form method="post" enctype="multipart/form-data">
<table class="table table-bordered" id="dataTable"
cellspacing="0">
<tr>
<td>PG Name</td>
<td><input type="text" class="form-control"
name="name" required=""></td></tr>
<tr>
<td>Facility</td>
<td>
<input type="checkbox" id="fac1" name="fac1"
value="Furnished"><label for="fac1">Furnished</label>
<input type="checkbox" id="fac2" name="fac2"
value="Unfurnished"><label for="fac2">Unfurnished</label>

```

```

value="Kitchen">><label for="fac2">Kitchen</label>
<input type="checkbox" id="fac3" name="fac3"
value="Study Area">><label for="fac3">Study Area</label>
<input type="checkbox" id="fac4" name="fac4"
value="AC">><label for="fac4">AC</label>
<input type="checkbox" id="fac5" name="fac5"
value="Food">><label for="fac5">Food</label>
</td>
</tr>
<tr>
<td>Location</td>
<td><input type="text" class="form-control"
name="location" required=""></td>
</tr>
<td>Address</td>
<td><input type="text" class="form-control"
name="address" required=""></td>
</tr>
<tr>
<td>Photo</td>
<td><input type="file" class="form-control"
name="file1" ></td>
</tr>
<tr>
<td>Photo</td>
<td><input type="file" class="form-control"
name="file2" ></td>
</tr>
<tr>
<td>Photo</td>

```

```
<td><input type="file" class="form-control"
name="file3" ></td>
</tr>
<tr>
<td>Rent</td>
<td><input type="text" class="form-control"
name="rent" required=""></td>
</tr>
<tr>
<td></td>
<td><input type="submit" class="submit-btn"
name="submit"></td>
</tr>
</table>
</form>
</div>
<div class="clearfix"></div>
</div>
</section>
<footer>
<div class="container">
<div class="row">
<div class="col-md-3 col-sm-6 col-xs-12
width-set-50">
<div class="footer-details">
</div>
</div>
<div class="col-md-3 col-sm-6 col-xs-12
width-50 width-set-50">
</div>
```

```
<div class="col-md-6 col-sm-6 col-xs-12">
</div>
</div>
</div>
</footer>
<a style="display: none;" href="javascript:void(0);"
class="scrollTop back-to-top" id="back-to-top">
<span><i aria-hidden="true"
class="fa fa-angle-up fa-lg"></i></span>
<span>Top</span>
</a>
</div>
</body>
</html>
```

6.3 Screenshots

6.3.1 Homepage

This is the homepage of User,Owner and Admin in the system. Here they can Login to the system and Register into the system.

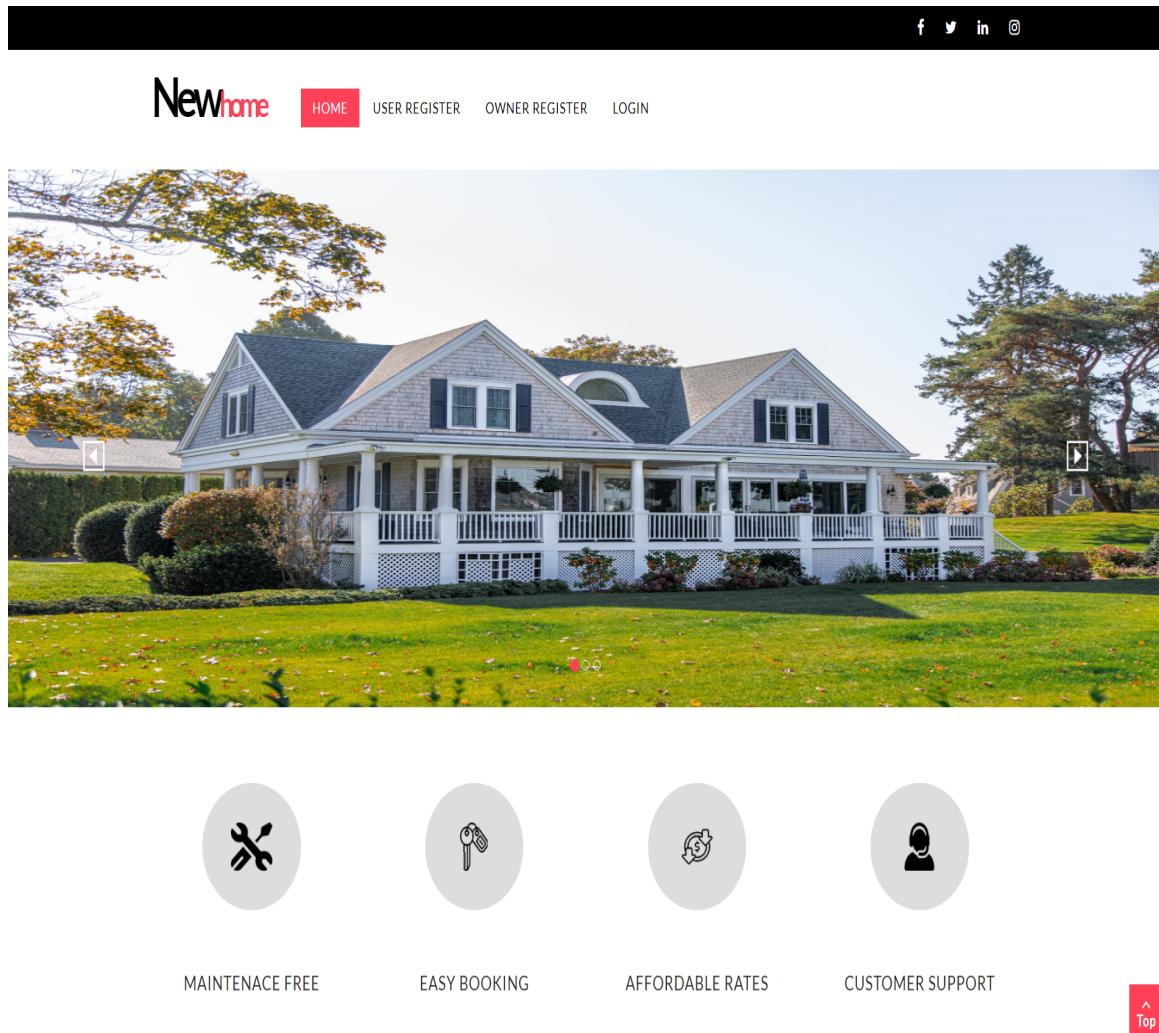


Figure 6.1: Homepage

6.3.2 Login page

Login page of all users.

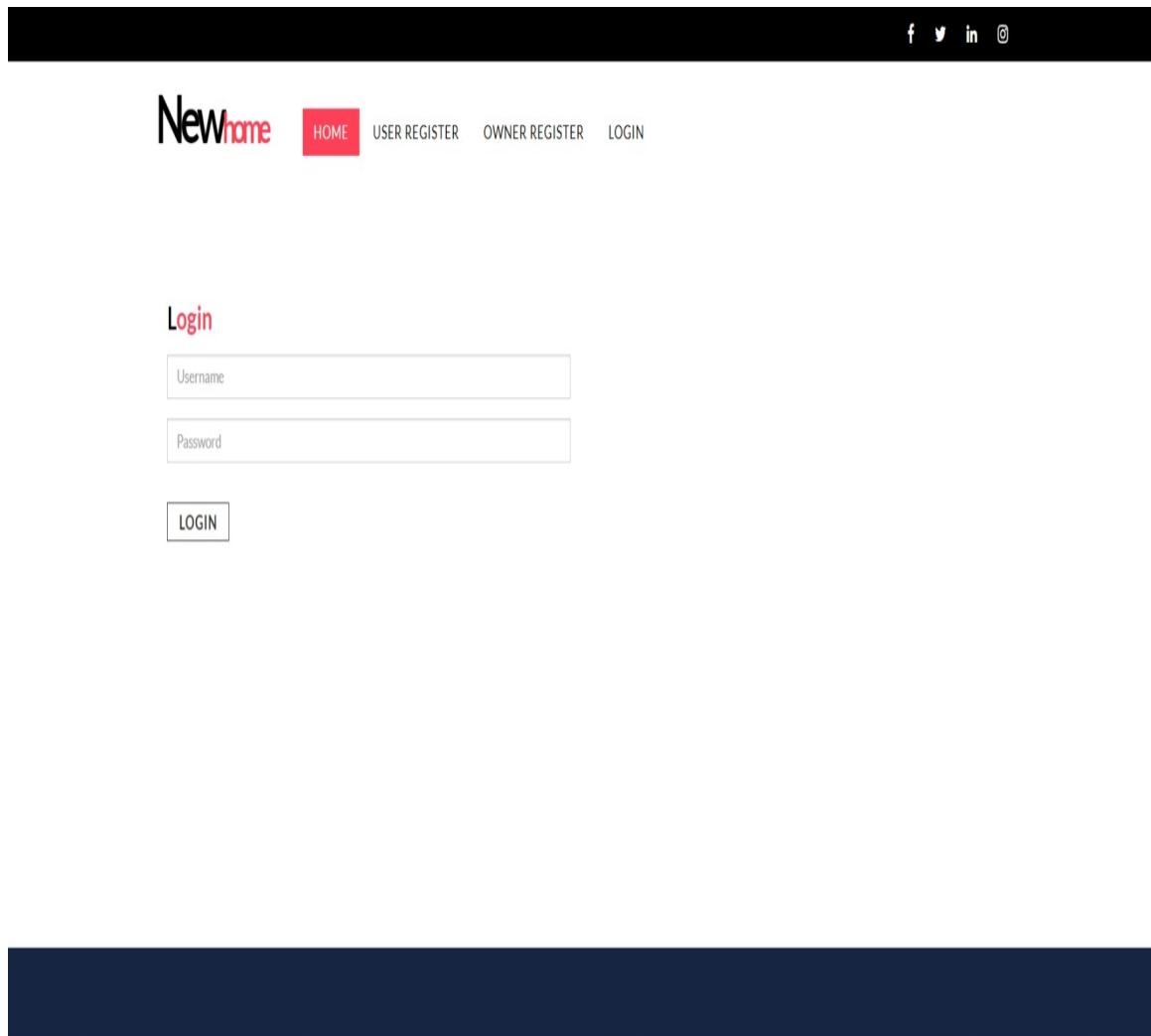
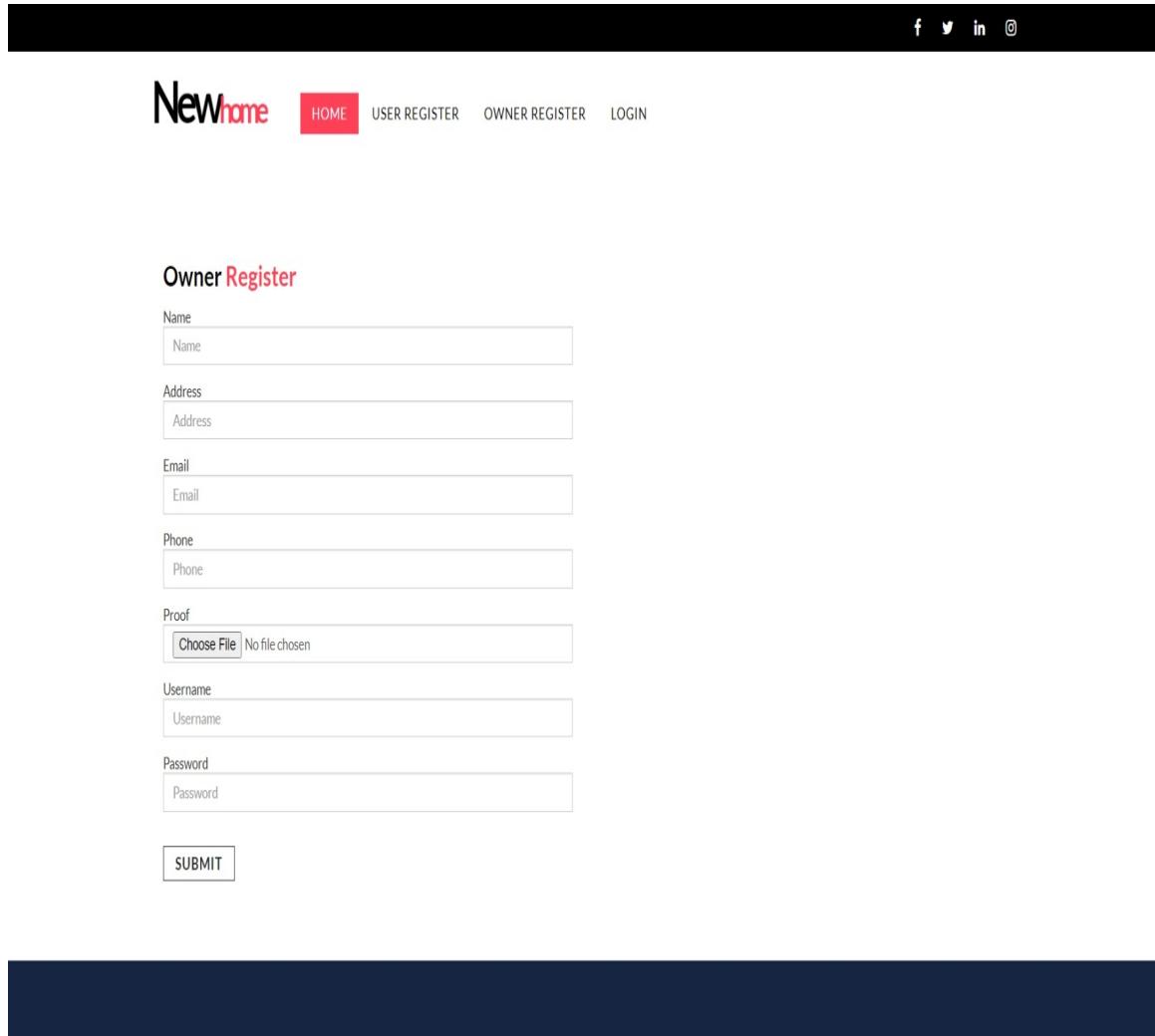


Figure 6.2: Login page

6.3.3 Owner Registration

Registration of Owner is done by the owner itself. Details include fullname, username, address, phone number, password.



The screenshot shows the 'Owner Register' page of the Newhome website. At the top, there is a navigation bar with links for HOME, USER REGISTER, OWNER REGISTER, and LOGIN. Below the navigation bar, there is a large black rectangular redaction box covering the majority of the page content. Above this redaction box, the title 'Owner Register' is visible. Below the title, there are several input fields for registration details:

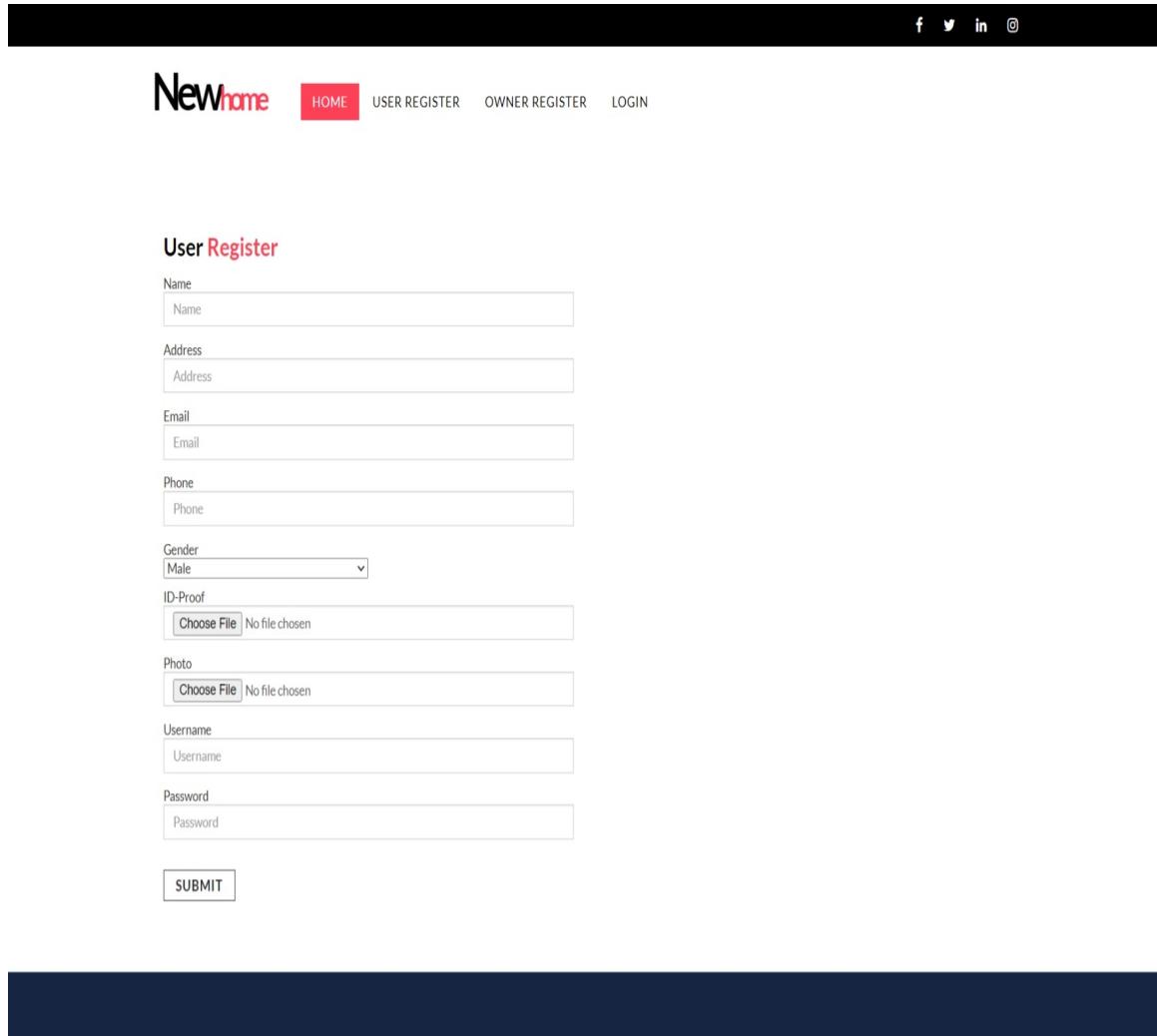
- Name: An input field labeled 'Name'.
- Address: An input field labeled 'Address'.
- Email: An input field labeled 'Email'.
- Phone: An input field labeled 'Phone'.
- Proof: A file upload input field labeled 'Choose File' with the message 'No file chosen'.
- Username: An input field labeled 'Username'.
- Password: An input field labeled 'Password'.

At the bottom left of the redacted area, there is a 'SUBMIT' button.

Figure 6.3: Registration of Owner

6.3.4 User Registration

Registration of User is done by the user itself. Details include fullname, username, address, phone number, password,idproof.



The screenshot shows the 'User Register' form on the Newhome website. The form fields are as follows:

- Name: A text input field labeled "Name".
- Address: A text input field labeled "Address".
- Email: A text input field labeled "Email".
- Phone: A text input field labeled "Phone".
- Gender: A dropdown menu set to "Male".
- ID-Proof: A file upload field with a placeholder "No file chosen".
- Photo: A file upload field with a placeholder "No file chosen".
- Username: A text input field labeled "Username".
- Password: A text input field labeled "Password".
- Submit: A button labeled "SUBMIT".

Figure 6.4: Registration of User

6.3.5 Owner Homepage

Homepage of Owner after logging in successfully.

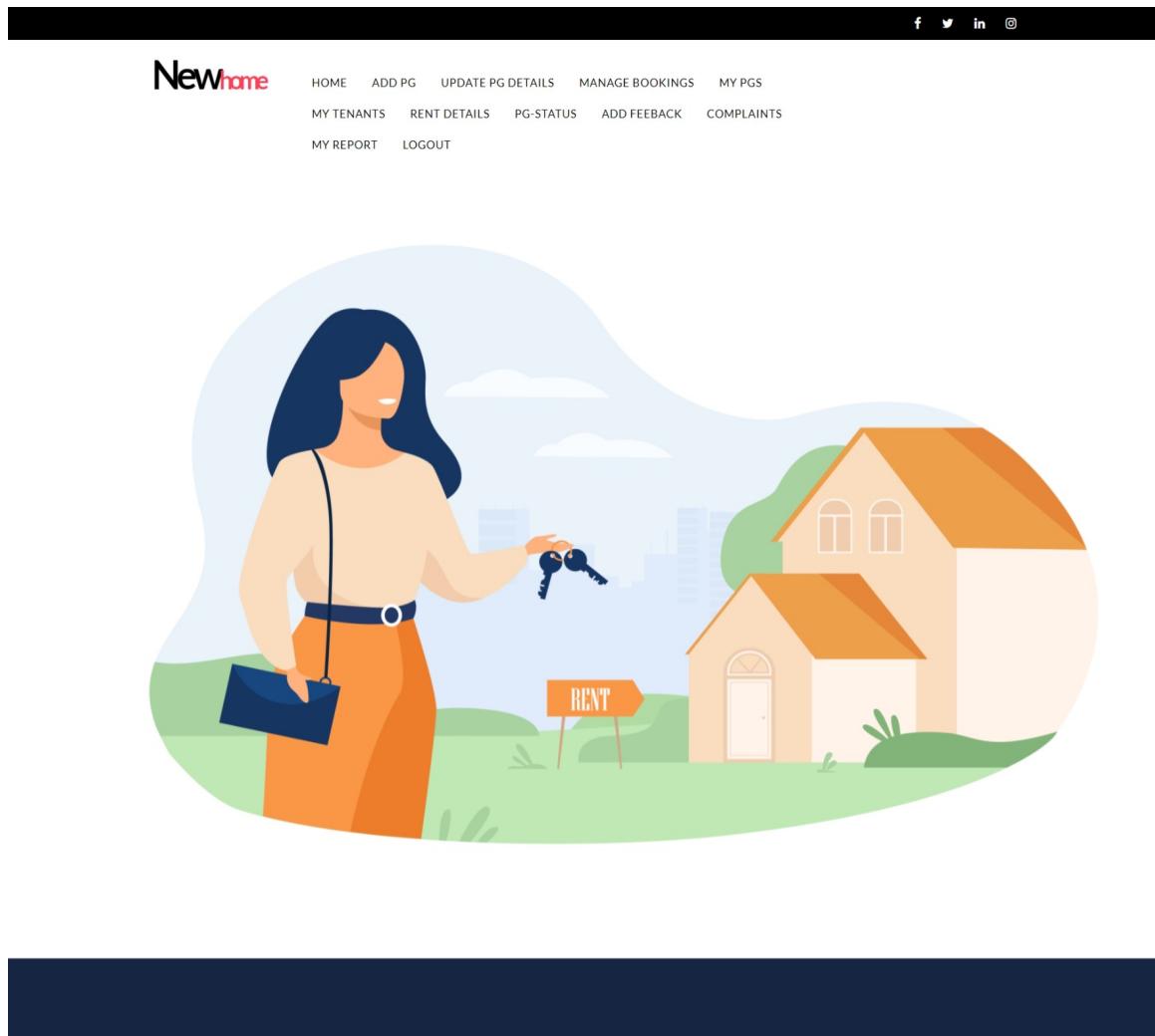


Figure 6.5: Owner Homepage

6.3.6 User Homepage

Homepage of User after logging in successfully.

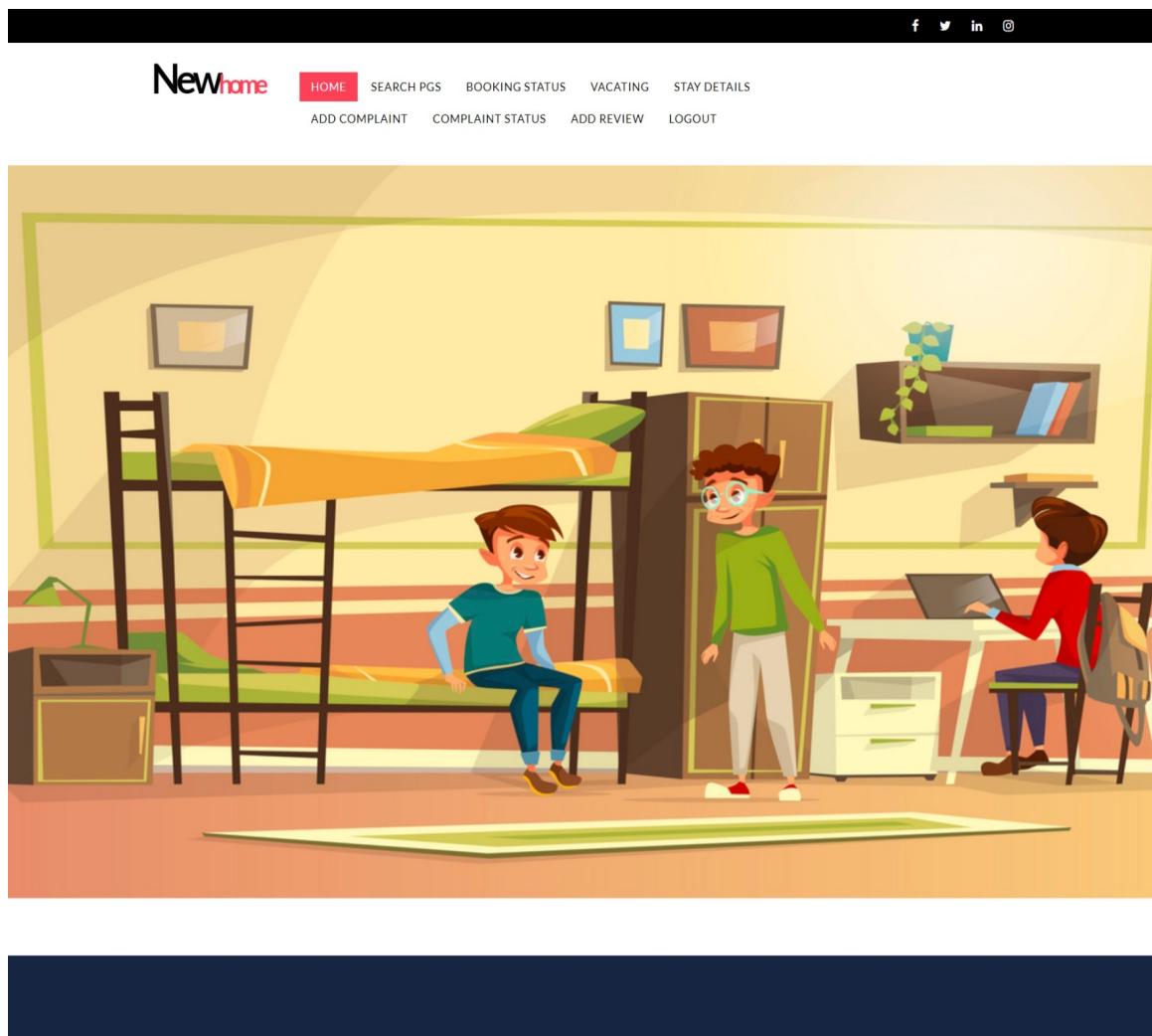


Figure 6.6: User Homepage

6.3.7 Admin Homepage

HomePage of Admin.

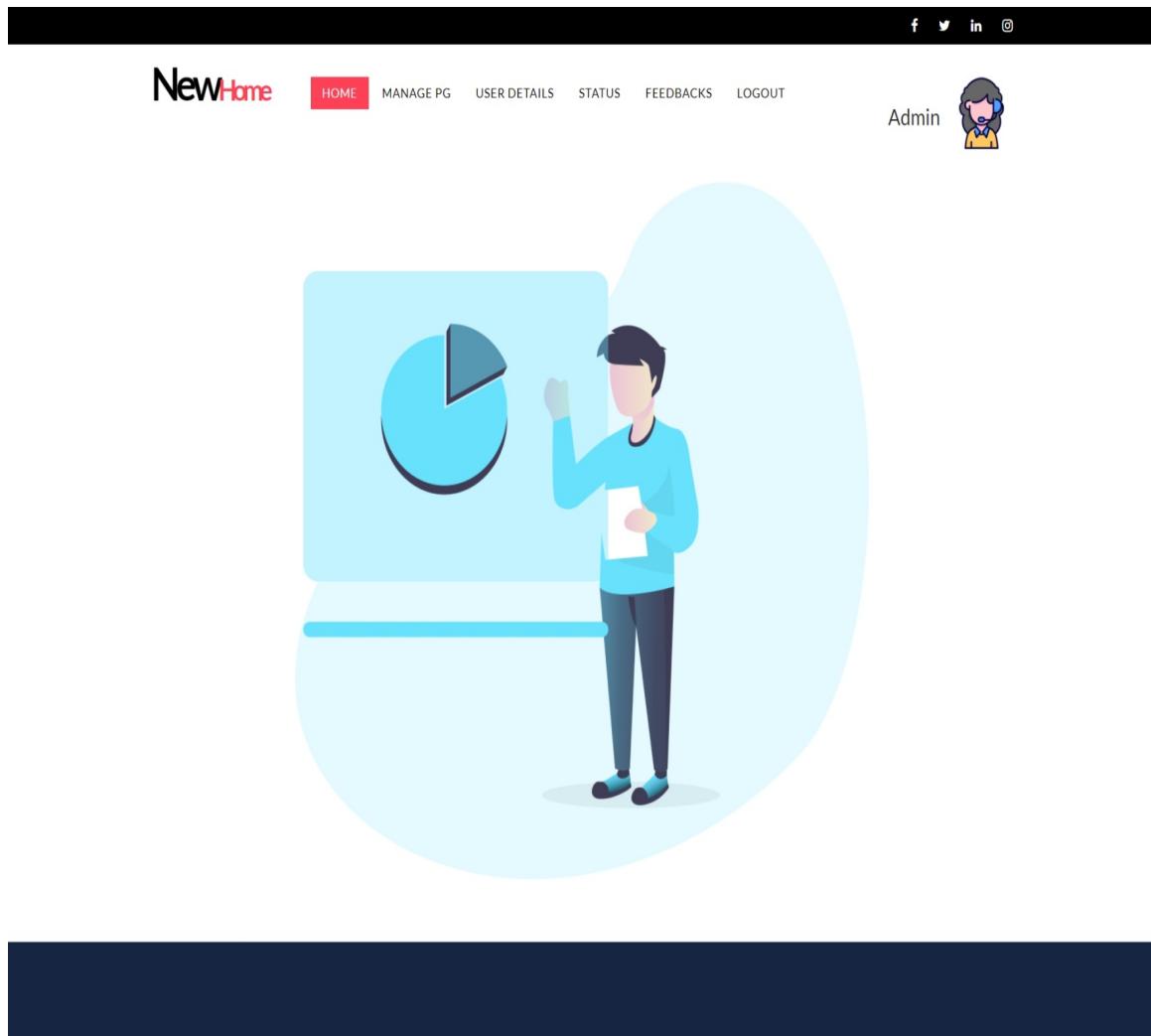


Figure 6.7: Admin Homepage

6.3.8 Add PG

Owner can Register a PG for make it avail to the Users.Details include PG Name,facilities provided,address,phone,rent .

The screenshot shows the 'Newhome' website interface. At the top, there is a navigation bar with links: HOME, ADD PG, UPDATE PG DETAILS, MANAGE BOOKINGS, MY PGS, MY TENANTS, RENT DETAILS, PG-STATUS, ADD FEEDBACK, COMPLAINTS, MY REPORT, and LOGOUT. Below the navigation bar, there is a social media sharing section with icons for Facebook, Twitter, LinkedIn, and Instagram. The main content area is titled 'AddPG' and contains a form for adding a new PG listing. The form fields are:

PG Name	<input type="text"/>
Facility	<input type="checkbox"/> Furnished <input type="checkbox"/> Kitchen <input type="checkbox"/> Study Area <input type="checkbox"/> AC <input type="checkbox"/> Food
Location	<input type="text"/>
Address	<input type="text"/>
Photo	<input type="button" value="Choose File"/> No file chosen
Photo	<input type="button" value="Choose File"/> No file chosen
Photo	<input type="button" value="Choose File"/> No file chosen
Rent	<input type="text"/>
<input type="button" value="SUBMIT"/>	

Figure 6.8: Add PG

6.3.9 Search PGs

User can search a PG by a Location that they would like to stay. And they can book one after selecting it.

The screenshot shows the Newhome website interface. At the top, there is a navigation bar with links: HOME (highlighted in red), SEARCH PGS, BOOKING STATUS, VACATING, STAY DETAILS, ADD COMPLAINT, COMPLAINT STATUS, ADD REVIEW, and LOGOUT. To the right of the navigation bar are social media icons for Facebook, Twitter, LinkedIn, and Instagram. Below the navigation bar, the page title is "Search By Location". A search input field contains the placeholder "e.g. Kozhikode" and a red "Search" button. The main content area displays three search results:

- COMFY STAY**
Address :Behind Theatre
Location :Eranjipalam
Rent :2500
Facilities :Furnished, Study, Area, AC

[BOOK PG](#)
- AASHRAYA CO-LIVING**
Address :Kallanthode
Location :Mukkam
Rent :2500
Facilities :Furnished, Study, Area

[BOOK PG](#)
- ANNAPOORNA STAYS**
Address :Behind Treasury
Location :Civil Station
Rent :3000
Facilities :Furnished, Study, Area

[BOOK PG](#)

Figure 6.9: Search PG

6.3.10 Book a PG

After selecting a PG the user can book one by giving the details like purpose of stay,duration of stay and the date they need to check in.

The screenshot shows the Newhome website interface. At the top, there is a navigation bar with links: HOME (highlighted in red), SEARCH PGS, BOOKING STATUS, VACATING, STAY DETAILS, ADD COMPLAINT, COMPLAINT STATUS, ADD REVIEW, and LOGOUT. Below the navigation bar, there is a social media sharing section with icons for Facebook, Twitter, LinkedIn, and Instagram. The main content area is titled "BookNow". It contains a form with three input fields: "Purpose" (empty), "Duration of Your Stay" (e.g. 3 Months,1 Year), and "Check-In Date" (dd-mm-yyyy). A "SUBMIT" button is located at the bottom right of the form. To the left of the form, there is a heading "Comfy Stay" above three small images of a room: a living room with red curtains, a bedroom with a double bed, and a view of the room's interior. To the right of the images, there is descriptive text: "Rent:2500/month", "Location:Eranjipalam", "Address:Behind Theatre", and "Contact No:7356831913". A red "Top" button is located at the bottom right of the page.

Figure 6.10: PG Booking

6.3.11 Manage Bookings

Owner accept or reject bookings made by the User.

The screenshot shows the Newhome web application interface. At the top, there is a black header bar with social media icons for Facebook, Twitter, LinkedIn, and Email. Below the header is the Newhome logo and a navigation menu with links: HOME, ADD PG, UPDATE PG DETAILS, MANAGE BOOKINGS (which is the active page), MY PGS, MY TENANTS, RENT DETAILS, PG-STATUS, ADD FEEDBACK, COMPLAINTS, MY REPORT, and LOGOUT.

The main content area has a title "Manage Bookings". Below it is a table with the following data:

Booking ID	Name	PG	Purpose	Date	Duration	Status	Actions
23	Akash K	Comfy Stays	Job Interview	2022-03-11	2 weeks	Booked	<button>Accept</button> <button>Reject</button>

Figure 6.11: Manage Bookings

6.3.12 User Add Review

User add review about the PG which they stayed.

The screenshot shows a web page with a dark header bar containing social media icons (f, v, in, @). Below the header is the Newhome logo and a navigation menu with links: HOME, SEARCH PGS, BOOKING STATUS, VACATING, STAY DETAILS, ADD COMPLAINT, COMPLAINT STATUS, ADD REVIEW, and LOGOUT. The main content area is titled "Add Review". It contains a form with three fields: "PG Name" (with a dropdown menu showing "Select"), "Write Your Review" (a text input field), and a "SUBMIT" button. The entire form is enclosed in a light gray border.

Figure 6.12: Add Review

6.3.13 User Add Complaint

User add a complaint to the owner.

The screenshot shows the Newhome website's header with social media links (f, t, in, g) and navigation menu items: HOME (highlighted in red), SEARCH PGS, BOOKING STATUS, VACATING, STAY DETAILS, ADD COMPLAINT, COMPLAINT STATUS, ADD REVIEW, and LOGOUT. Below the header is the 'Add Complaint' form. It consists of three rows: 1. A row with a 'Select PG' dropdown and a 'Select' button. 2. A row with a 'Type Your Complaint Here' text input field. 3. A row with a large empty area containing a 'SUBMIT' button. The entire form is enclosed in a light gray border.

Figure 6.13: Add Complaint

6.3.14 Vacate PG

User can vacate from PG when they no longer need to stay.

The screenshot shows the Newhome website interface. At the top, there is a dark header bar with social media icons (Facebook, Twitter, LinkedIn, and a magnifying glass). Below the header, the Newhome logo is displayed, followed by a navigation menu with links: HOME (highlighted in red), SEARCH PGS, BOOKING STATUS, VACATING, STAY DETAILS, ADD COMPLAINT, COMPLAINT STATUS, ADD REVIEW, and LOGOUT.

The main content area is titled "VacateForm". Below the title is a table with the following data:

Booking-ID	Name	PG	Purpose	Date	Duration	Status	Take Action
23	Akash K	Comfy Stays	Job Interview	2022-03-11	2 weeks	Accept	Check Out

Figure 6.14: Vacate PG

6.3.15 Admin Report

Admin can view the overall report about the system. The details include the Total Bookings Received, Total Users of the system, User Reviews.

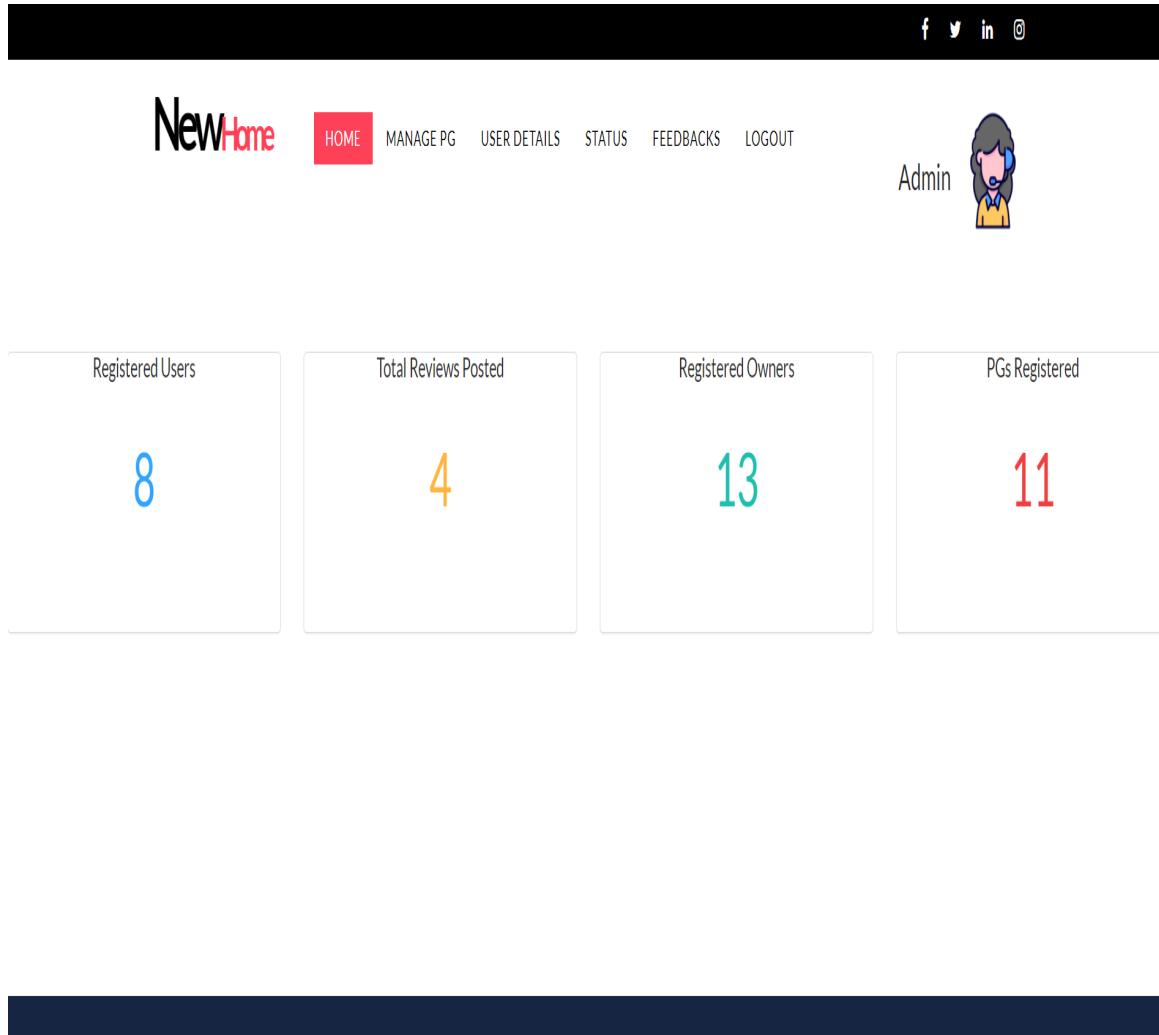


Figure 6.15: Admin Report

6.3.16 Owner Report

Owner can view the overall report about their PG. The details include the Total Bookings Received, Total Users, User Reviews and Monthly Income.

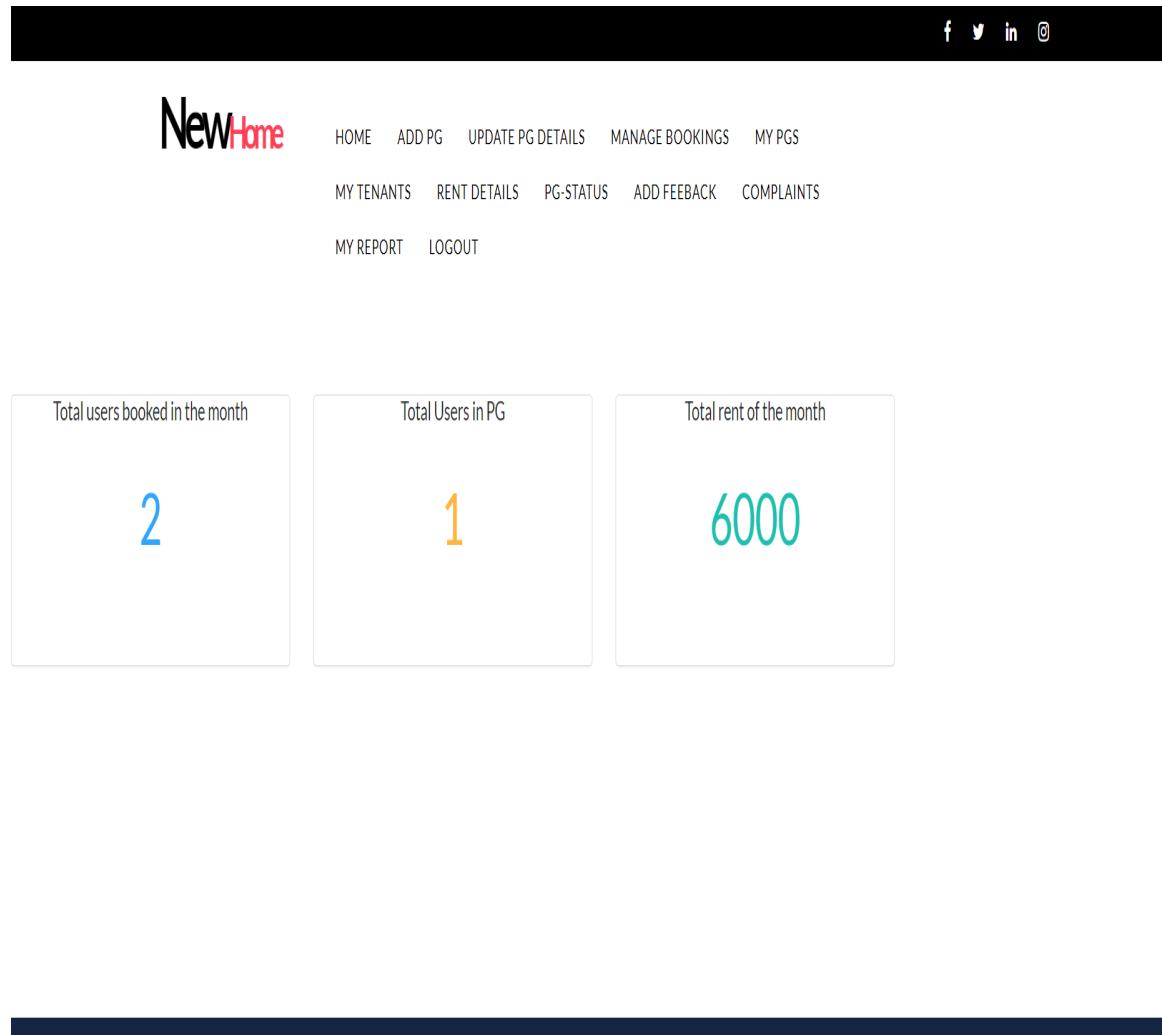


Figure 6.16: Owner Report

6.4 Git History

The screenshot shows a GitHub repository page for 'Amjad-VK / Mini-Project'. The repository is private. The main branch is 'main' (1 branch, 0 tags). The commit history lists 12 commits from 'Amjad-VK' (a4504b5) made 'now' (less than a minute ago). The commits include file uploads for 'css', 'database', 'font-awesome', 'fonts/antonio-exotic', 'js', 'Owner_Home.php', 'README.md', 'accept_booking.php', 'admin_delcomplaint.php', 'admin_delpg.php', 'admin_delusr.php', 'admin_home.php', 'admin_managepg.php', and 'admin_manageusr.php'. All commits were made within the last month.

File	Commit Message	Time Ago
css	Add files via upload	29 minutes ago
database	Add files via upload	1 hour ago
font-awesome	Add files via upload	11 minutes ago
fonts/antonio-exotic	Add files via upload	11 minutes ago
js	Add files via upload	29 minutes ago
Owner_Home.php	Add files via upload	1 minute ago
README.md	Update README.md	1 hour ago
accept_booking.php	Add files via upload	6 minutes ago
admin_delcomplaint.php	Add files via upload	6 minutes ago
admin_delpg.php	Add files via upload	last month
admin_delusr.php	Add files via upload	last month
admin_home.php	Add files via upload	last month
admin_managepg.php	Add files via upload	last month
admin_manageusr.php	Add files via upload	last month

Figure 6.17: Git 1

	admin_mycomplaints.php	Add files via upload	7 minutes ago
	admin_viewuser.php	Add files via upload	7 minutes ago
	adminmenu.php	Add files via upload	7 minutes ago
	check_in.php	Add files via upload	2 minutes ago
	check_out.php	Add files via upload	2 minutes ago
	complaint-create.php	Add files via upload	7 minutes ago
	dbconnection.php	Add files via upload	last month
	index.php	Add files via upload	last month
	login.php	Add files via upload	2 minutes ago
	owner_addfacilities.php	Add files via upload	2 minutes ago
	owner_adddpg.php	Add files via upload	2 minutes ago
	owner_complaint.php	Add files via upload	2 minutes ago
	owner_editpg.php	Add files via upload	2 minutes ago
	owner_register.php	Add files via upload	last month
	owner_viewbookings.php	Add files via upload	1 hour ago
	owner_viewcomplaints.php	Add files via upload	1 hour ago
	owner_viewoccupied.php	Add files via upload	1 hour ago
	owner_viewusers.php	Add files via upload	1 hour ago
	reject_booking.php	Add files via upload	1 hour ago
	report_admin.php	Add files via upload	1 hour ago
	report_owner.php	Add files via upload	1 hour ago

Figure 6.18: Git 2

📄	tableeg.php	Add files via upload
		last month
📄	user_booking.php	Add files via upload
		1 minute ago
📄	user_checkout.php	Add files via upload
		1 minute ago
📄	user_home.php	Add files via upload
		last month
📄	user_register.php	Add files via upload
		last month
📄	user_review.php	Add files via upload
		1 hour ago
📄	user_staydetails.php	Add files via upload
		1 hour ago
📄	user_vacatepg.php	Add files via upload
		1 hour ago
📄	user_vacatepg2.php	Add files via upload
		1 minute ago
📄	user_viewpg1.php	Add files via upload
		1 minute ago
📄	usermenu.php	Add files via upload
		1 minute ago
README.md		
∅		
<h2>Mini-Project</h2>		
<hr/> <p>Paying Guest Management System - Team Member :AMJAD V K</p>		

Figure 6.19: Git 3