Fall 2022

Management System for an Apartment

Srihari Mondem

MANAGEMENT SYSTEM FOR AN APARTMENT

By

Srihari Mondem

B. Tech, St Martins Engineering college, 2017

GRADUATE CAPSTONE SEMINAR PROJECT

Submitted in partial fulfilment of the requirements

For the Degree of Master of Science,

With a Major in Computer Science

Governors State University
University Park, IL 60484

2022

ABSTRACT

We all know that the population is growing daily, and the existing resources are not satisfying the population's needs. If we consider a city where in finding a home for rent, knowing the amenities of the house, rent of the house, location of the house is difficult. In the current system, finding a house for rent is difficult for a tenant, but managing the home by the owner is also a complicated issue. Keeping in mind about the issues of the tenant and owner, a new application is developed which is known as Management system for Apartment.

This application will be having the users like tenant, owner, and the administrator. If a tenant wants to find a house or apartment on rent, then he needs to go to the application and he should register with it. After registering with the application, he should log into it and at the same time he can search for the vacant houses or apartments in the given location. Application will be taking the input and provides the necessary details to the users. If we click on that information, then we can view all the details of the house or apartment. This information will help the tenant to view the home without going there, they can know the rent of the house, they can contact the owner directly. If both parties are agreed for the house, then they will be entering the house, owner will be having the right to enter the tenant details. A user name will be created for the tenant and then he can contact to the owner related to the issues of the homes, bills, payments etc. All this information will be stored in the database and whenever the owner, and the tenant is having any issue and they can make use of these details to clarify the issue. Existing applications will not be suitable for the current requirements, so we want to design and develop new applications. We are expecting to complete this application before 12th of December, 2022.

This application is developed by using Java technologies and the database we use is MySQL. To design the GUI of the application, we will be using HTML5, CSS3, Bootstrap and the coding will be written by using jQuery and PHP.

Table of Content

1. *Project Description*

As per the project requirements, application was divided into different modules. These modules are

a. Owner module.

b. Tenant module.

c. Admin module.

Owner Module: Owner will register with the application, they can add the details of their house like address of the house, amenities present in the house, house rent, status of the house etc. owner will have some features like edit, add, delete, view, communicate with the tenant etc.

Tenant Module: Tenant should register with the application, he can view the different vacant apartments, houses in the selected locality. After communicating with the owners, they can become like a tenant in the house. Later he can communicate with the owner, he can keep track of all the payments, issues, and other related information related to the house etc. He can view the reports whenever he wants. [2].

Admin Module: Administrator is the privileged user when compared to the other user. He will be viewing all the operations of both tenant and owner. He can restrict the operations of both the tenant and the owner. He can even block the owner and tenant. He can able to communicate with both the tenant and owner. If we want any of the details of the owners and tenant, he can view all the information in the form of reports whenever needed and from wherever he wants.[4].

In developing this application, we will use the core PHP and CodeIgniter framework for implementing it. Apart from this we will be using the java technologies for implementing the requirements of the application.

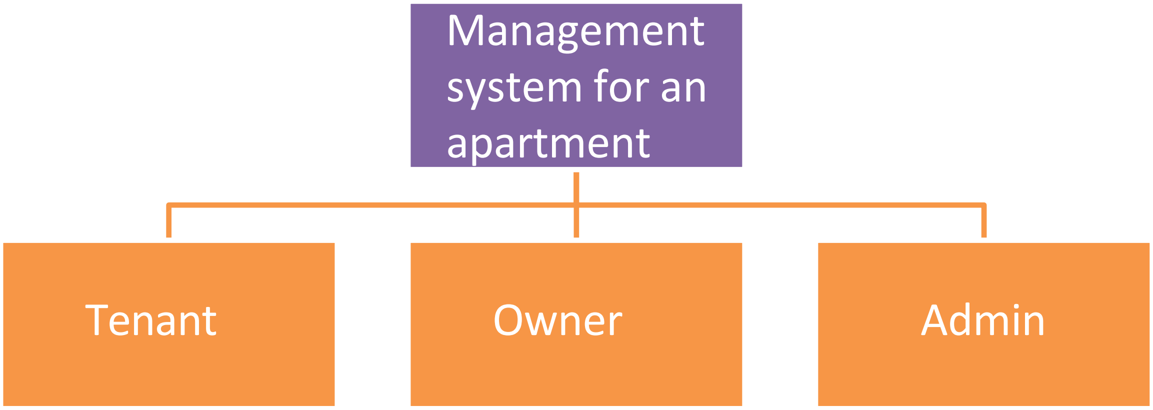


Figure 1: Modules of Management system for an Apartment

Above Figure 1 explains how many modules are there in the given application. All the modules internal details are clearly explained in the previous page. By using the above figure, we can even know how many numbers of users there are in this application. This helps us to understand all the requirements of the application.

1.1. *Competitive Information*

Application is developed based on the given requirements, there are no similar applications with the same requirements. By this we can understand there is no competitive or similar application with these requirements.

1.2. Relationship to Other Applications/Projects

As it is an individual application, it is not related to any other application. We cannot integrate this application in any others project.

1.3. Assumptions and Dependencies

- To view the information from the tenant, the owner should enter it. The owner should register with the application; he needs to enter the details. The tenants can view the details of the houses entered by the owner.
- It is assumed that admin will be having the high privileges when compared to the owner and the tenant.
- Tenant will not have any right to delete the data.

1.4. Future Enhancements

In future, we want to establish a communication between the tenants of the apartment. They can even view the information related to the other tenants of the apartment, they can send friend request, communicate with each other etc.

1.5. Definitions and Acronyms

SQL – Structure query language
HTML – Hypertext markup language
CSS – Cascading Style sheet.

2. Project Technical Description

This section is provided with the project technical description. This project is designed by using the different user interface tools like HTML, CSS etc. we have used the bootstrap mechanism and MySQL database is used to store and retrieve the data. To achieve the operations, we used the PHP scripting language and jQuery for implementing the project requirements. We have used the Xampp webserver and different other design notations to represent the requirements.

2.1. Application Architecture

Here we are using the 3-tier architecture in which user layer, application layer and the database layer will be present. In the user layer, all the users of the application will be present who will be interacting with the application. They will be using the application and then they will be doing all the operations. Here, the owner, tenant and the admin are the users, and they will be interacting with the application layer by staying at the user layer. [1]

2.2. Application Information flows

Application will be used by the 3 people like owner of the house, tenant of the house and the admin. Following use case diagram shows what are the operations that will be done by each user. This diagram provides the information about the services or operations that can be done by the user.

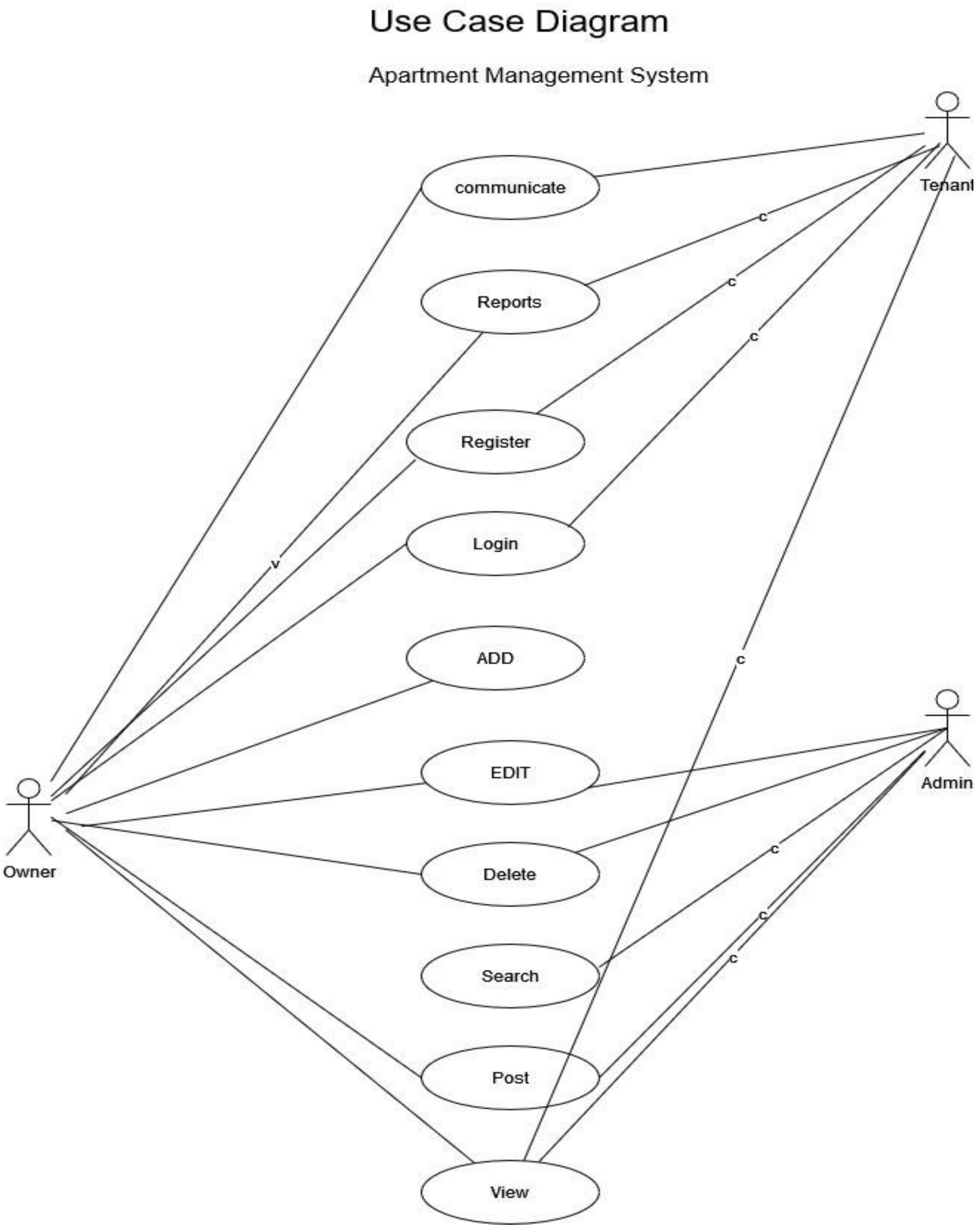


Figure 2: Use case diagram for Management system for Apartment

2.3. Interactions with other Projects (if Any)

As it is individual project, there are no interactions from this application to any other application.

2.4. Interactions with other Applications

NO

2.5. Capabilities

As per the type of the user, capabilities are provided here. They are

Owner:

a. owner will able to create the profile.

b. He can able to add the house.

c. He can view the details.

d. He can edit the details of the house.

e. He can delete the details.

f. He can view the interested tenants.

g. He can edit the personal details.

Tenant:

a. Registration page

b. Login page

c. Search the houses.

d. View the house details.

e. Communicate with the owner through mail.

f. He can edit his personal details.

Admin:

a. Login operation.

b. He can view list of owners.

c. Search operation.

d. Delete operation.

e. Deactivate operation.

f. Adding the owners directly.

g. View tenants.

h. Delete tenants

i. Deactivate tenants

2.6. Risk Assessment and Management

Risk assessment and management will be implemented from the initial stage of the project. At the initial stage, we did1` the risk assessment stage in the requirements phase and then came to know several types of risks. In the initial stages, there are missing requirements; if that continues, the project will not be useful. After identifying the missing requirements, all the required requirements are added and then they are updated in the requirements document. If the risk analysis is not implemented and missing requirements or any other risks are not identified in the initial stages, the same will be continued in the later stages and there will be possibility of increasing risk and the impact of the risk will be more.

Risk assessment involves different steps. All these steps are applied in this project to manage several types of risks. These steps include risk identification, risk prioritization, risk mitigation.

3. Project Requirements

3.1. Identification of Requirements

After careful analysis of the apartment operations, I have divided the requirements based on the users

<GSU-APM_2022_ Owner profile creation -000100>

This requirement will allow the owner to have the registration process with the application and he will be creating the profile by entering the required information.

<GSU-APM_2022_Owner_Adding the house -000101>

This requirement helps the owner to enter the house details into the application. These details later can be viewed by the tenant.

<GSU-APM_2022_Owner _View -000103>

The owner will have the right to view his details and the information sent or communicated by the tenant.

<GSU-APM_2022_Owner _Edit -000104>

Owner will be having the right to edit his personal details, house details etc. This provision will be given to the owner.

<GSU-APM_2022_Owner _Delete -000105>

Owner will be having the right to delete his profile from the application.

<GSU-APM_2022_Owner _View tenants -000106>

Owner will be having the right to view the interested tenants. This helps the owner to view how many people are interested in taking their house for rent.

Tenant Requirements:

<GSU-APM_2022_Tenant_Registration_000107>

To use this application, tenants must register with it. After registration he will become the user of the application. To complete the registration process, he needs to enter the required details.

<GSU-APM_2022_Tenant_Login_000108>

After registration, with the help of the login credentials, tenant will enter the application. After entering the application, the user will have the dashboard with the required operations. These operations can be done by the tenant.

<GSU-APM_2022_Tenant_Search_000109>

After logging into the application by the tenant, tenant should have a feature to search the houses. This feature should find the house as per the requirements of the tenant.

<GSU-APM_2022_Tenant_View_000110>

This feature allows the tenant to view their personal details, the house details, the owner details, etc.

<GSU-APM_2022_Tenant_mail_000111>

If the tenant is interested in the house and want to know about the details personally, then he can establish the communication using the mail from the application itself. This mail feature is provided in the application.

<GSU-APM_2022_Tenant_Edit_000112>

If there is a change in his personal details, he should be given the editing features so that he will be updating them.

Admin operations:

<GSU-APM_2022_Admin_Login_000113>

Admin will be directly logging into the application; he will not have any registration process; hard coding will be done in the application with the username and password.

<GSU-APM_2022_Admin_view_000114>

Admin will be having additional privilege when compared to the owner and the tenant. He will be having the right to view all the owners. He will be viewing individually single owner details.

<GSU-APM_2022_Admin_search_000115>

Admin can able to search the tenant and owner information. He can search the houses also.

<GSU-APM_2022_Admin_Delete_000116>

Admin will be deleting the owner or tenant with this feature.

<GSU-APM_2022_Admin_Deactivate_000117>

Admin will able to deactivate any of the user in performing the operations.

3.2. Operations, Administration, Maintenance and Provisioning (OAM&P)

Apartment is having set of rules and standards that need to be followed by the owners and tenants. As per the rules and standards, users' operations have been decided. All these operations are implemented in this application, and they will be performing the operations with the help of this application.

When there is any kind of crash with the application, all the data that is present in the database can be recovered. This kind of implementation will bring confidence in the Management system for Apartment.

3.3. Security and Fraud Prevention

Application will be used by several types of people and every user will have their own dashboard. One user operation will not be visible to the other users. This will give the privacy to all the users. Another security provided to the users is providing username and password. This kind of implementation prevents unauthorized usage of application. When the user provides the credentials, it will be validated in all the ways. If the details entered are correct, only they will be allowed to perform the operations. In this way security is implemented in the application.

3.4. Release and Transition Plan

S. No	Activity	Start date	Final date	Status
1	Abstract submission	09-01-22	09-15-22	completed
2	Requirements Gathering	09-16-22	10-06-22	Completed
3	Requirements Analysis	10-07-22	10-25-22	Completed
4	Design	10-26-22	11-05-2022	Completed
5	Coding	11-26-2022	11-18-2022	Completed
6	Testing	11-19-2022	11-22-2022	Completed
7	Documentation	11-23-2022	11-28-2022	Completed
8	Presentation	01-12-2022	01-12-2022	Completed

Table 1: Release and transition plan

4. Project Design Description

This section is useful to represent the requirements in the diagrammatic way. This helps the people to understand the flow of the application. Several types of design notations are used to represent the implementation in separate ways. We have used different UML notations, database design and the User Interface design to implement all the requirements of 3 users. [4]

A. Use case design:

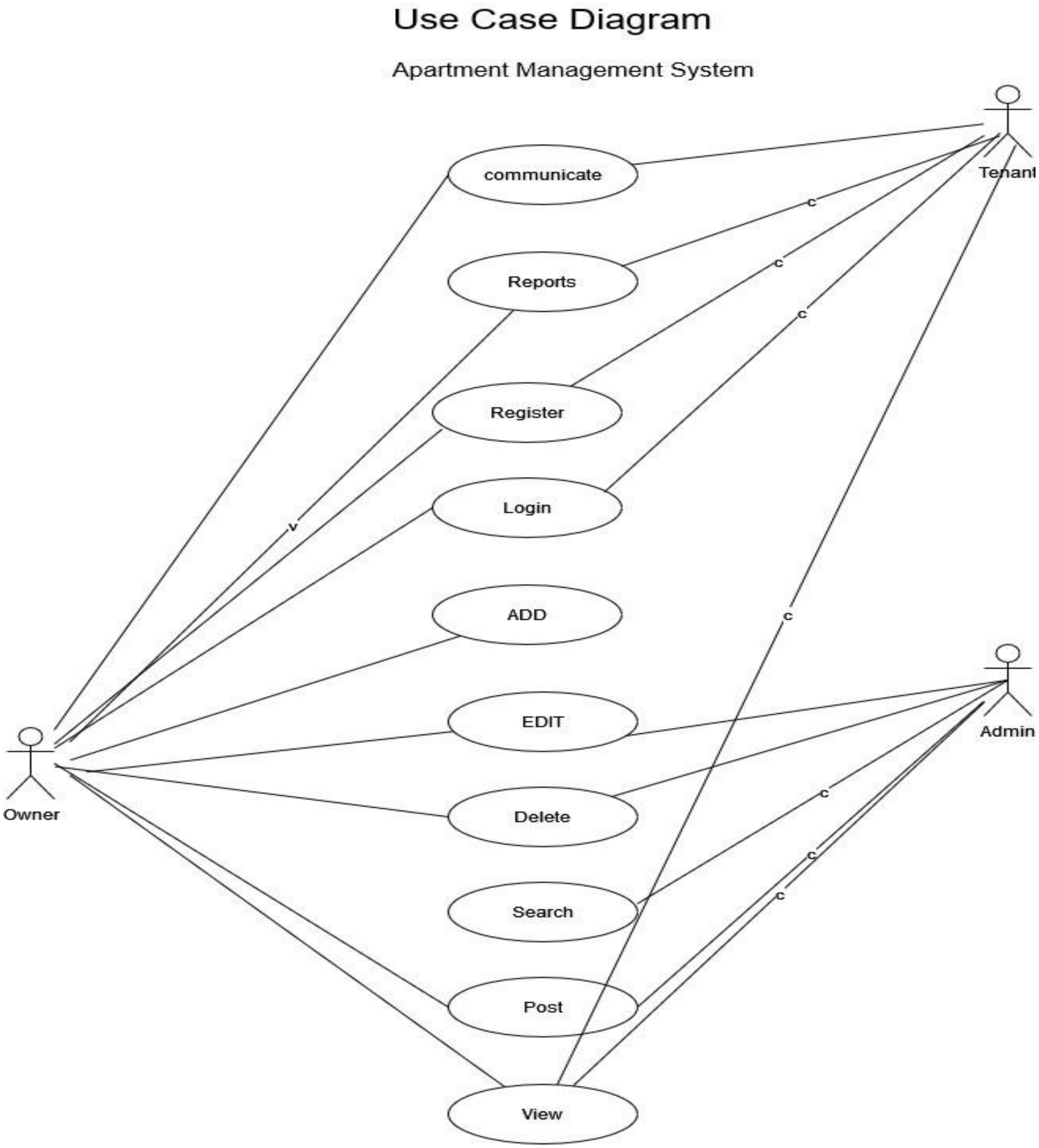


Figure 1: Use case design for Management system for Apartment.

Database design:

Db name: apartment_db

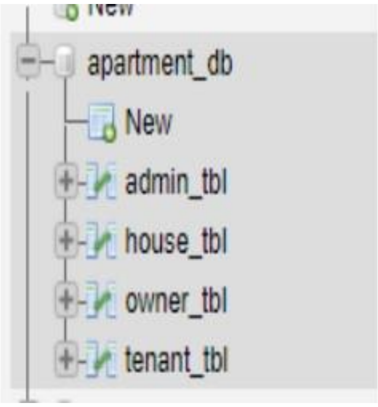


Figure 2: List of tables

DB tables:

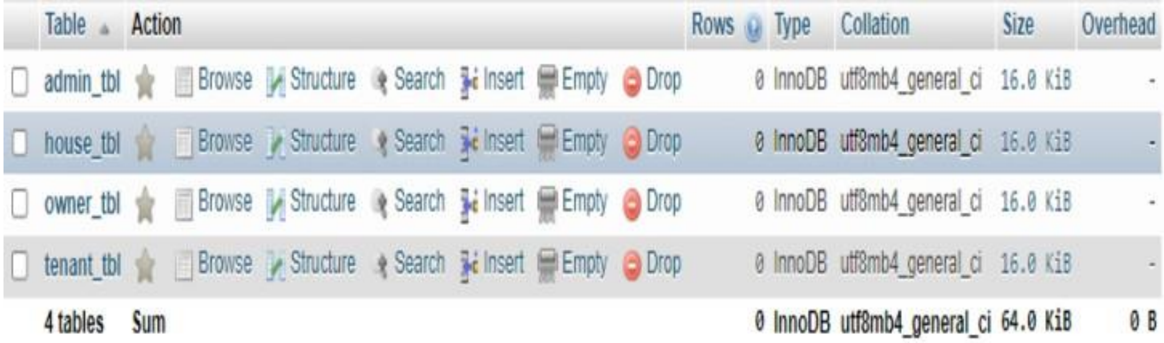
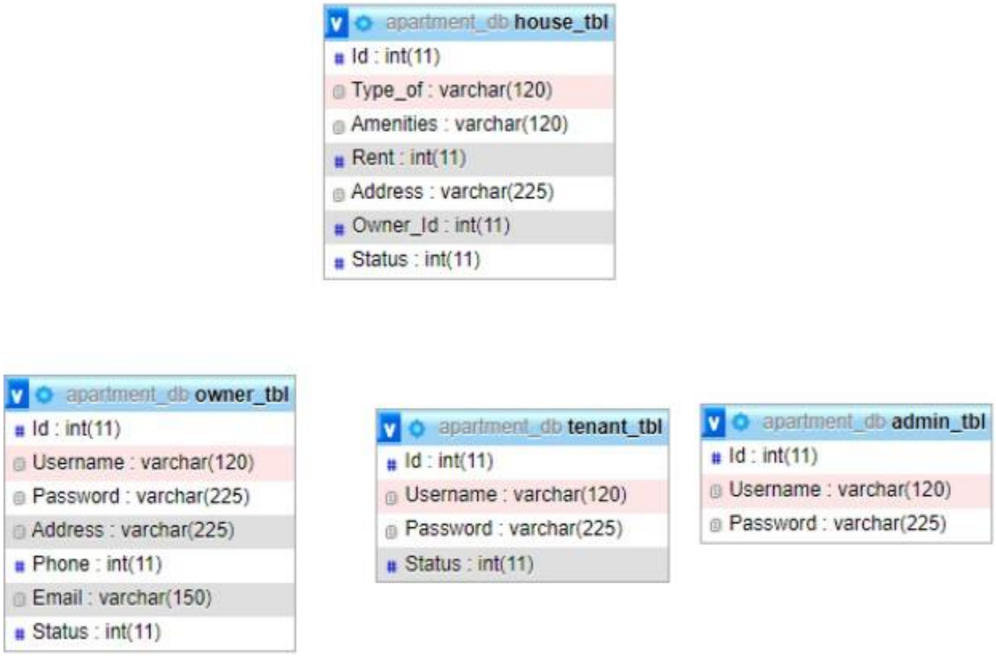


	Table	Action	Rows	Type	Collation	Size	Overhead
□	admin_tbl	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	house_tbl	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	owner_tbl	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	tenant_tbl	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
	4 tables	Sum	0	InnoDB	utf8mb4_general_ci	64.0 KiB	0 B

Figure 3: Table structures

Er diagram:



apartment_db house_tbl

- Id : int(11)
- Type_of : varchar(120)
- Amenities : varchar(120)
- Rent : int(11)
- Address : varchar(225)
- Owner_Id : int(11)
- Status : int(11)

apartment_db owner_tbl

- Id : int(11)
- Username : varchar(120)
- Password : varchar(225)
- Address : varchar(225)
- Phone : int(11)
- Email : varchar(150)
- Status : int(11)

apartment_db tenant_tbl

- Id : int(11)
- Username : varchar(120)
- Password : varchar(225)
- Status : int(11)

apartment_db admin_tbl

- Id : int(11)
- Username : varchar(120)
- Password : varchar(225)

Figure 4: E-R Diagram

5. Internal/external Interface Impacts and Specification

To perform the operations of the users, it should interact with the database. To view the information, the application will be retrieving it. Database connectivity is established while writing the coding and to interact with the database a user-friendly interface is designed. With the help of these interfaces, users will be interacting with the database. [3].

6. Design Units Impacts

Requirements are gathered based on the type of the user. All these requirements are transformed into representation by several types of design. With this design one will understand the flow of the information, flow of the operations etc. As this is the web application, everything will be covered in one unit.

6.1. Functional Area A/Design Unit A

6.1.1. Functional Overview

This application is developed for the owners and tenants of the apartment. Sometimes owners will not find the tenant and even if the tenant is there, establishing the communication with the tenant became a problem to the owner. Even for the tenants, it is exceedingly difficult to search for a vacant apartment. This application provides an interaction between the owner and the tenant and exchanges the information between them.

6.1.2. Impacts

With this project's help, there will be a significant impact on the owner and tenant.

Owner: Finding a tenant for the owner will become easy with this application. Owner will be benefitted and he will be communicated with the interested tenants directly.

Tenant: Searching a house or an apartment is not easy. Physically searching is a tedious task, to overcome this a third-party application is needed in satisfying the needs of the tenant. The tenant will see all the information related to the house, amenities, rent etc. If he is satisfied with them, he will contact the owner to rent the house.

6.1.3. Requirements

All the requirements which have been stated in the previous section are designed with the user interfaces. Following are the user interfaces which allows the user to perform the operations.

Management system for Apartment:
Owner panel:
Owner Registration page:

Registration page:

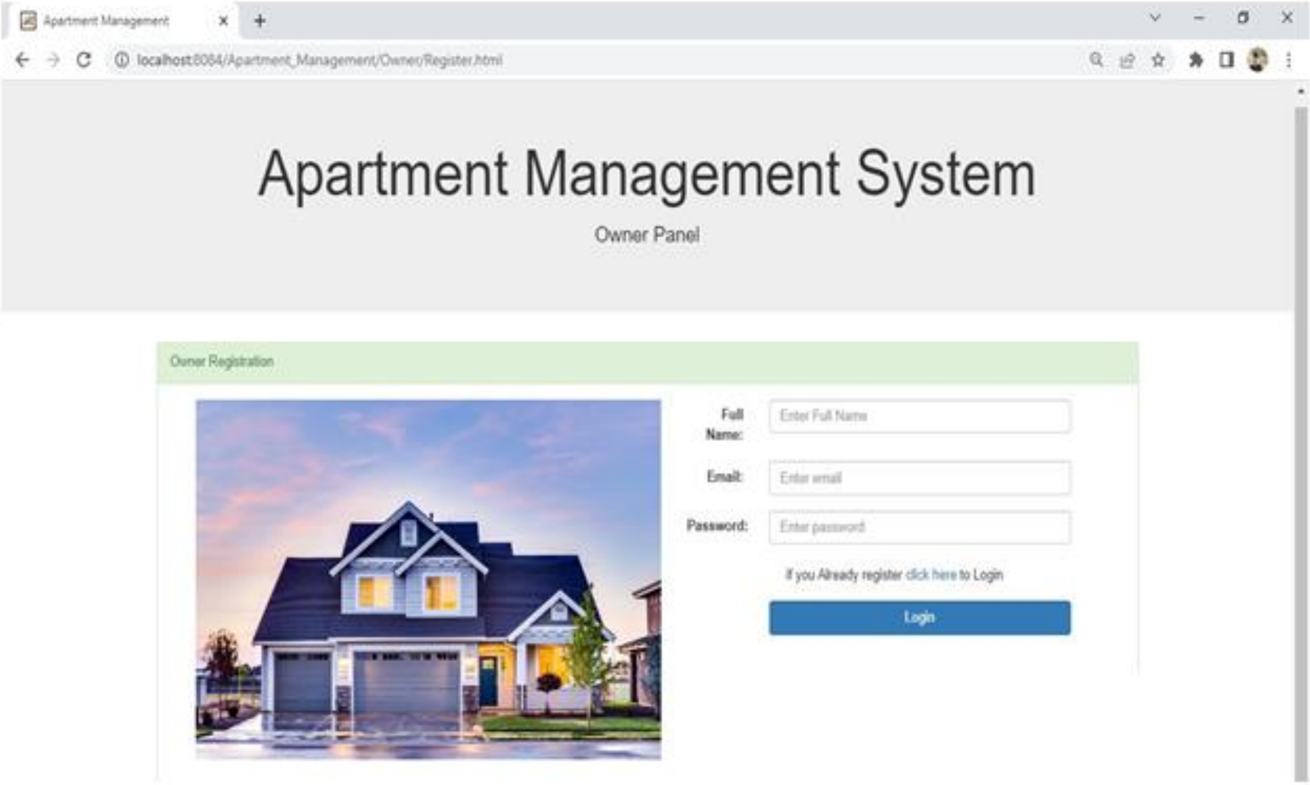
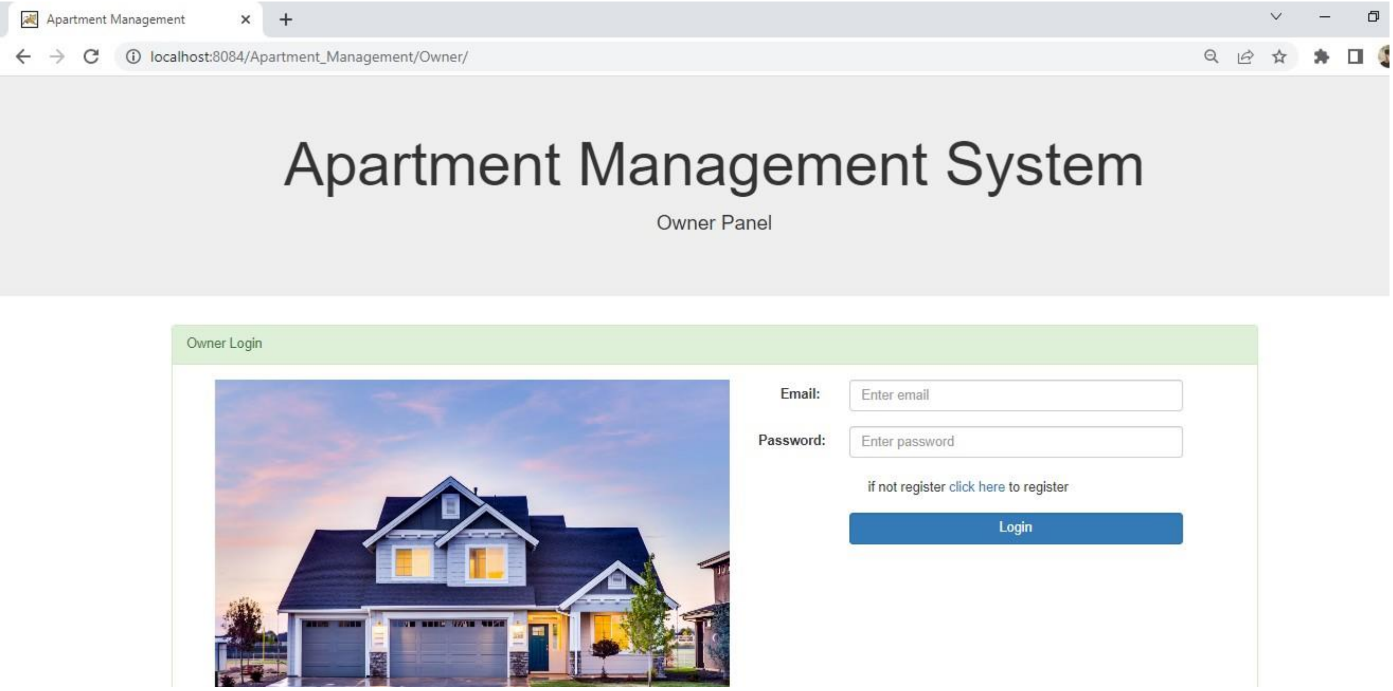

Apartment Management
localhost:8084/Apartment_Management/Owner/Register.html
Apartment Management System
Owner Panel
Owner Registration
Full Name:
Enter Full Name
Email:
Enter email
Password:
Enter password
If you Already register click here to Login
Login

Figure 5: Registration page

Login Page:



Apartment Management

localhost:8084/Apartment_Management/Owner/

Apartment Management System

Owner Panel

Owner Login

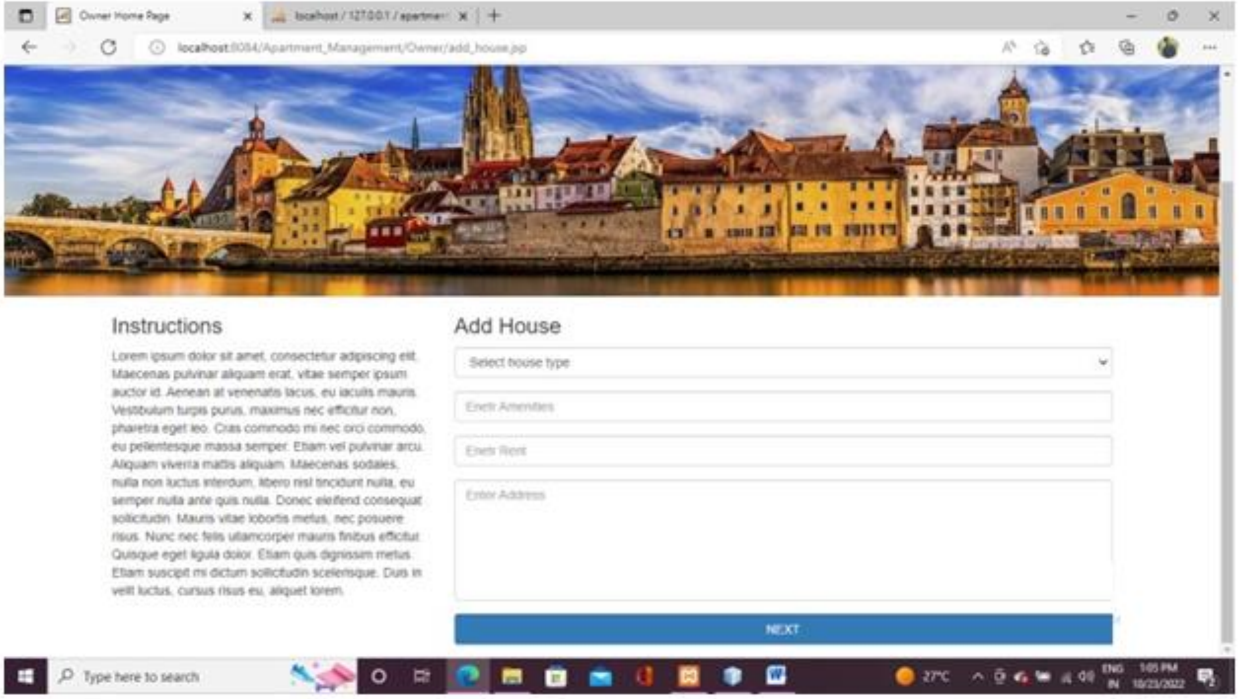
Email: Enter email

Password: Enter password

if not register click here to register

Figure 6: Login page

Add houses:


Owner Home Page
localhost / 127.0.0.1 / apartmen
localhost:8084/Apartment_Management/Owner/add_house.jsp

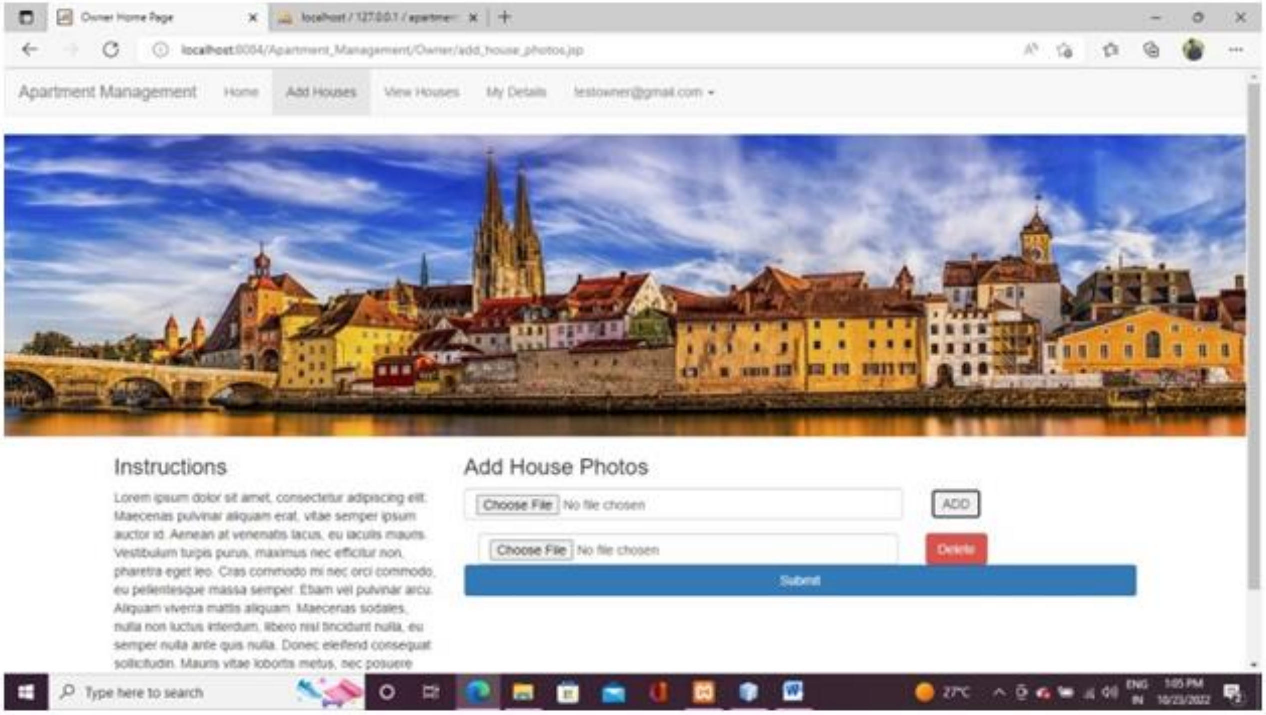
Instructions

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas pulvinar aliquam erat, vitae semper ipsum auctor id. Aenean at venenatis lacus, eu iaculis mauris. Vestibulum turpis purus, maximus nec efficitur non, pharetra eget leo. Cras commodo mi nec orci commodo, eu pellentesque massa semper. Etiam vel pulvinar arcu. Aliquam viverra mattis aliquam. Maecenas sodales, nulla non luctus interdum, libero nisl tincidunt nulla, eu semper nulla ante quis nulla. Donec eleifend consequat sollicitudin. Mauris vitae lobortis metus, nec posuere risus. Nunc nec felis ullamcorper mauris finibus efficitur. Quisque eget ligula dolor. Etiam quis dignissim metus. Etiam suscipit mi dictum sollicitudin scelerisque. Duis in velit luctus, cursus risus eu, aliquet lorem.

Add House

Figure 7: Add houses

Adding house images:


Owner Home Page
localhost / 127.0.0.1 / apartmen
localhost:8084/Apartment_Management/Owner/add_house_photos.jsp
Apartment Management
Home
Add Houses
View Houses
My Details
testowner@gmail.com ▾

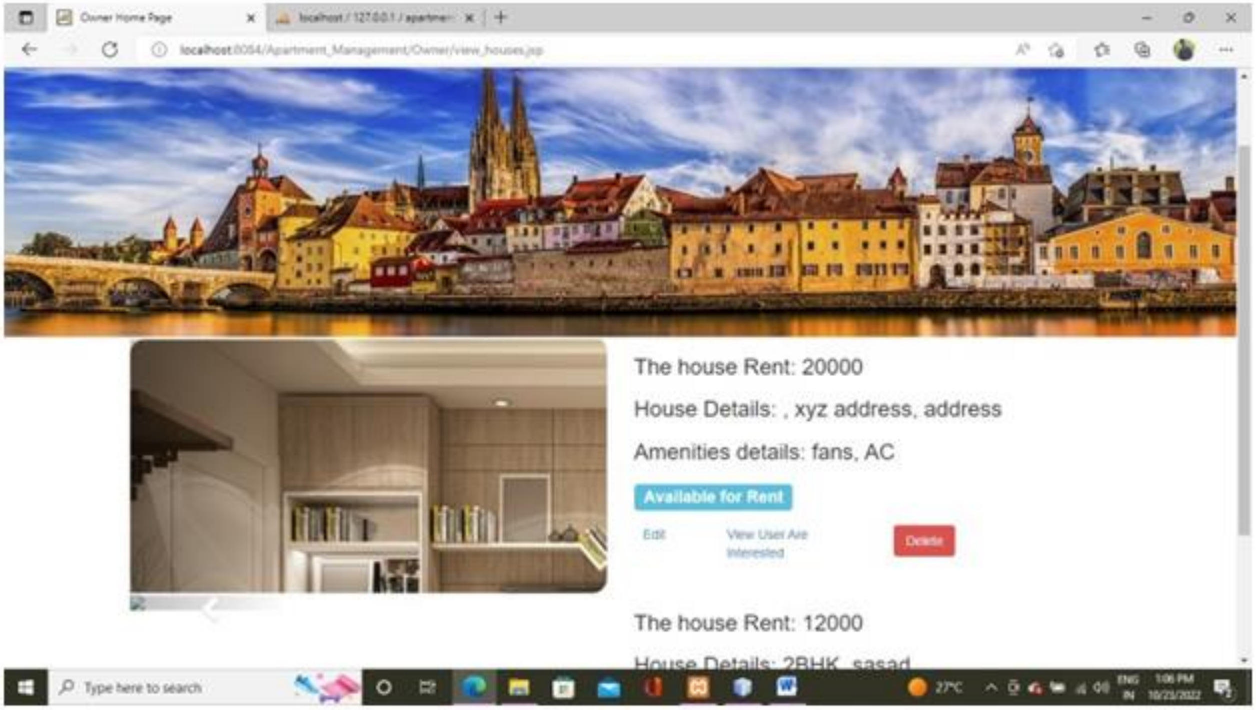
Instructions

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas pulvinar aliquam erat, vitae semper ipsum auctor id. Aenean at venenatis lacus, eu iaculis mauris. Vestibulum turpis purus, maximus nec efficitur non, pharetra eget leo. Cras commodo mi nec orci commodo, eu pellentesque massa semper. Etiam vel pulvinar arcu. Aliquam viverra mattis aliquam. Maecenas sodales, nulla non luctus interdum, libero nisl tincidunt nulla, eu semper nulla ante quis nulla. Donec eleifend consequat sollicitudin. Mauris vitae lobortis metus, nec posuere

Add House Photos

Figure 8: Adding house images

View house details:



The house Rent: 20000

House Details: , xyz address, address

Amenities details: fans, AC

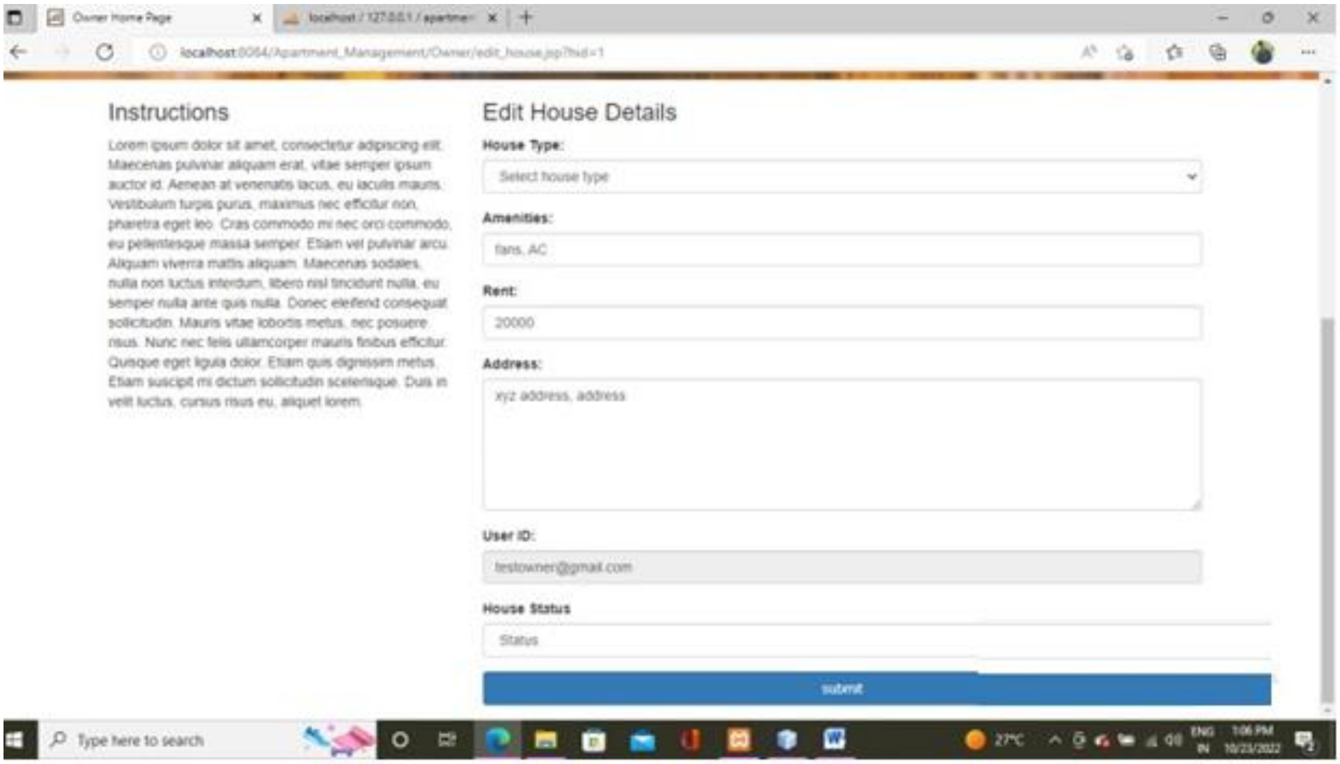
Owner Home Page
localhost / 127.0.0.1 / apartmen
localhost:8084/Apartment_Management/Owner/view_houses.jsp
Available for Rent
Edit
View User Are Interested
Delete

The house Rent: 12000

House Details: 2BHK, sasad

Figure 9: View house details

Edit house details:


Owner Home Page
localhost / 127.0.0.1 / apartme
localhost:8084/Apartment_Management/Owner/edit_house.jsp?hid=1
Type here to search
27°C
ENG
IN
1:06 PM
10/23/2022

Instructions

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas pulvinar aliquam erat, vitae semper ipsum auctor id. Aenean at venenatis lacus, eu iaculis mauris. Vestibulum turpis purus, maximus nec efficitur non, pharetra eget leo. Cras commodo mi nec orci commodo, eu pellentesque massa semper. Etiam vel pulvinar arcu. Aliquam viverra mattis aliquam. Maecenas sodales, nulla non luctus interdum, libero nisl tincidunt nulla, eu semper nulla ante quis nulla. Donec eleifend consequat sollicitudin. Mauris vitae lobortis metus, nec posuere risus. Nunc nec felis ullamcorper mauris finibus efficitur. Quisque eget ligula dolor. Etiam quis dignissim metus. Etiam suscipit mi dictum sollicitudin scelerisque. Duis in velit luctus, cursus risus eu, aliquet lorem.

Edit House Details

House Type:

Select house type

Amenities:

fans, AC

Rent:

20000

Address:

xyz address, address

User ID:

testowner@gmail.com

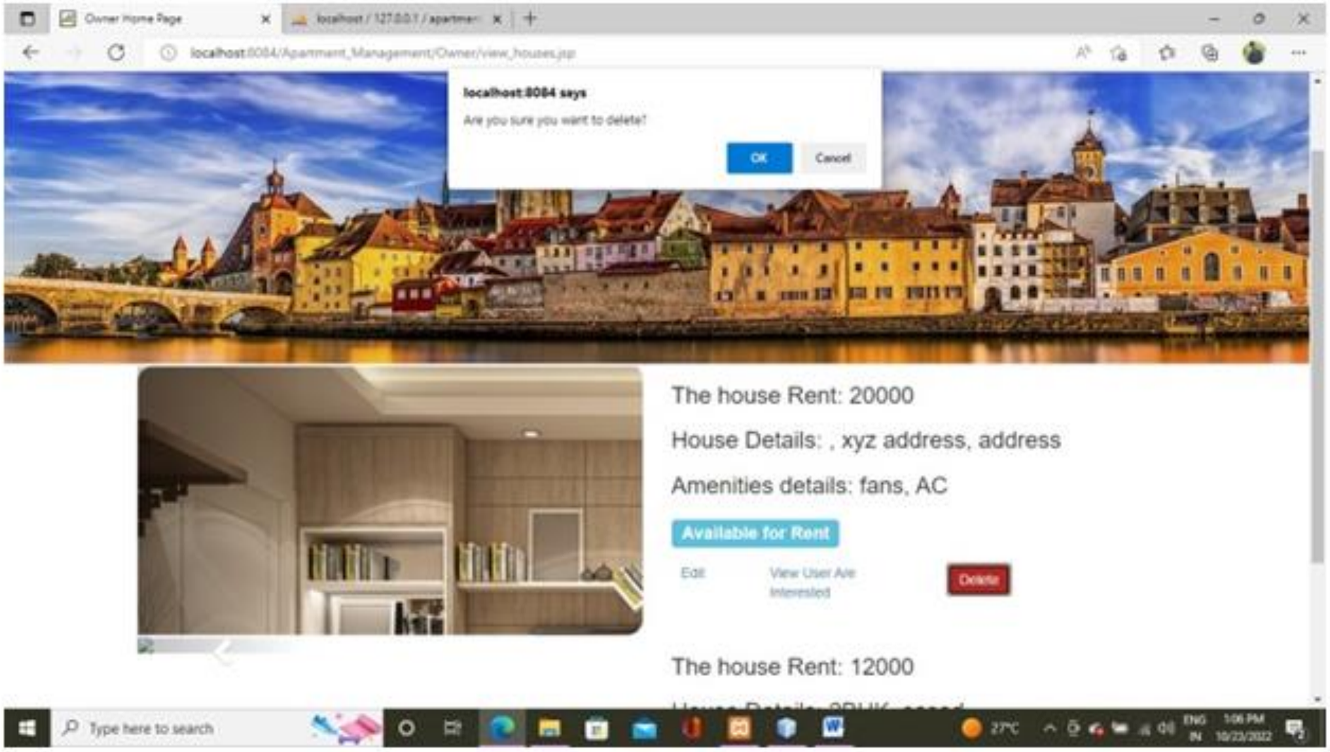
House Status

Status

submit

Figure 10: Edit house details

Delete house details:



localhost:8084 says

Are you sure you want to delete?

The house Rent: 20000

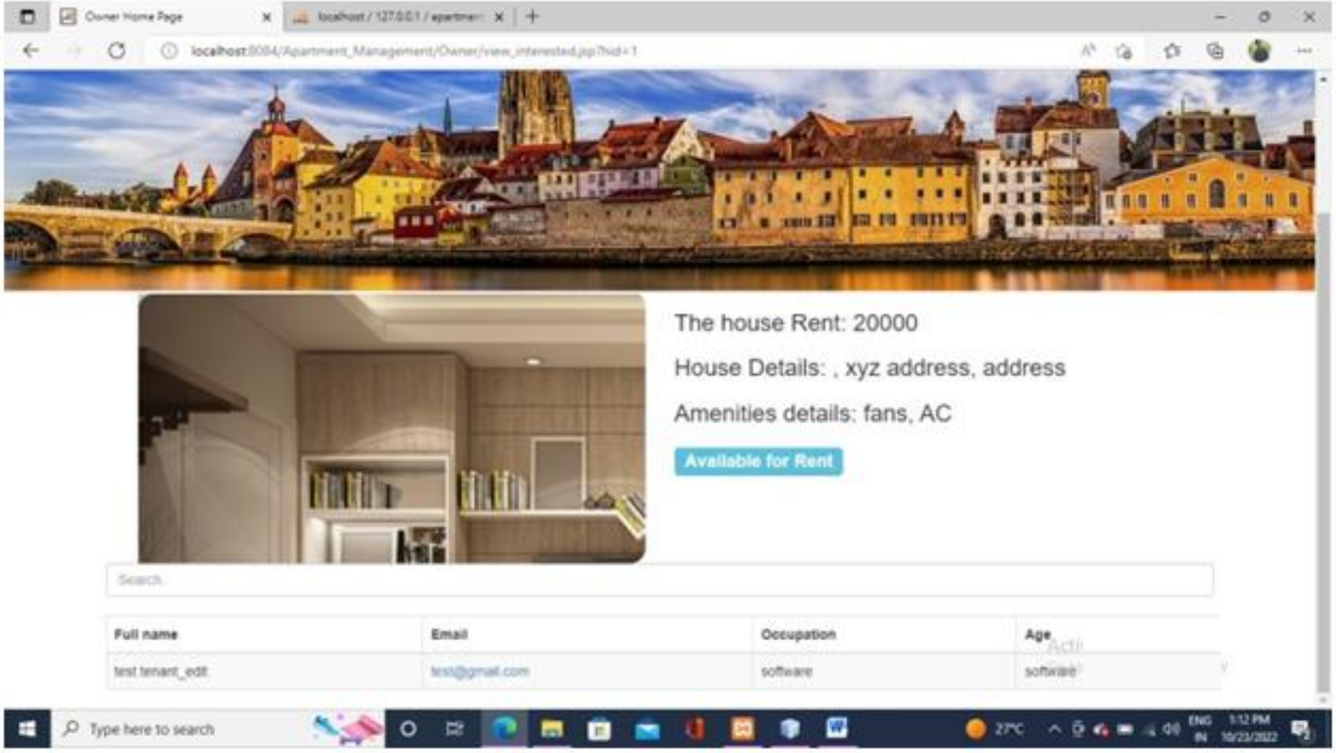
House Details: , xyz address, address

Amenities details: fans, AC

The house Rent: 12000

Figure 11: Delete house details

View interested tenants:



Owner Home Page
localhost / 127.0.0.1 / apartmen
localhost:8084/Apartment_Management/Owner/view_interested.jsp?hid=1

The house Rent: 20000

House Details: , xyz address, address

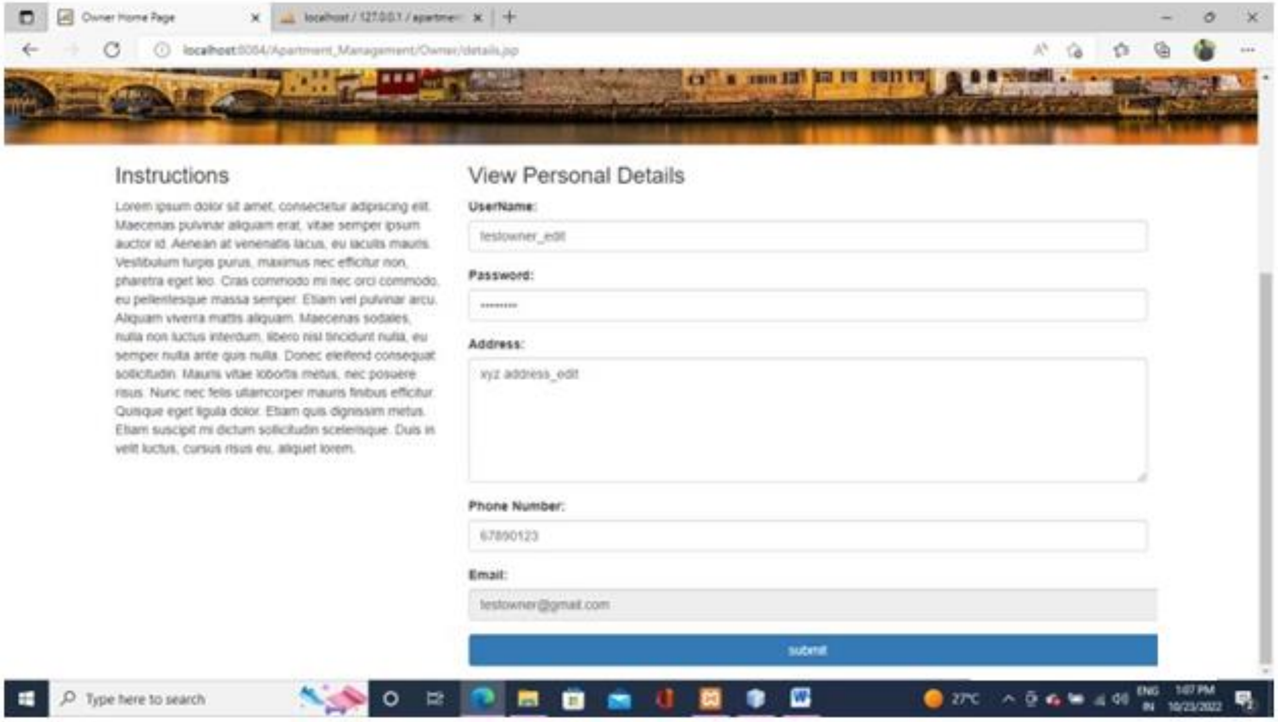
Amenities details: fans, AC

Available for Rent
Search
Type here to search
27°C
ENG
IN
1:12 PM
10/23/2022

Full name	Email	Occupation	Age
test tenant_edit	test@gmail.com	software	software

Figure 12: View interested tenants

View and edit personal details:


Owner Home Page
localhost / 127.0.0.1 / apartme
localhost:8084/Apartment_Management/Owner/details.jsp
Type here to search
27°C
ENG
IN
1:07 PM
10/23/2022

Instructions

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas pulvinar aliquam erat, vitae semper ipsum auctor id. Aenean at venenatis lacus, eu iaculis mauris. Vestibulum turpis purus, maximus nec efficitur non, pharetra eget leo. Cras commodo mi nec orci commodo, eu pellentesque massa semper. Etiam vel pulvinar arcu. Aliquam viverra mattis aliquam. Maecenas sodales, nulla non luctus interdum, libero nisl tincidunt nulla, eu semper nulla ante quis nulla. Donec eleifend consequat sollicitudin. Mauris vitae lobortis metus, nec posuere risus. Nunc nec felis ullamcorper mauris finibus efficitur. Quisque eget ligula dolor. Etiam quis dignissim metus. Etiam suscipit mi dictum sollicitudin scelerisque. Duis in velit luctus, cursus risus eu, aliquet lorem.

View Personal Details

UserName:
testowner_edit

Password:
••••••••

Address:
xyz address_edit

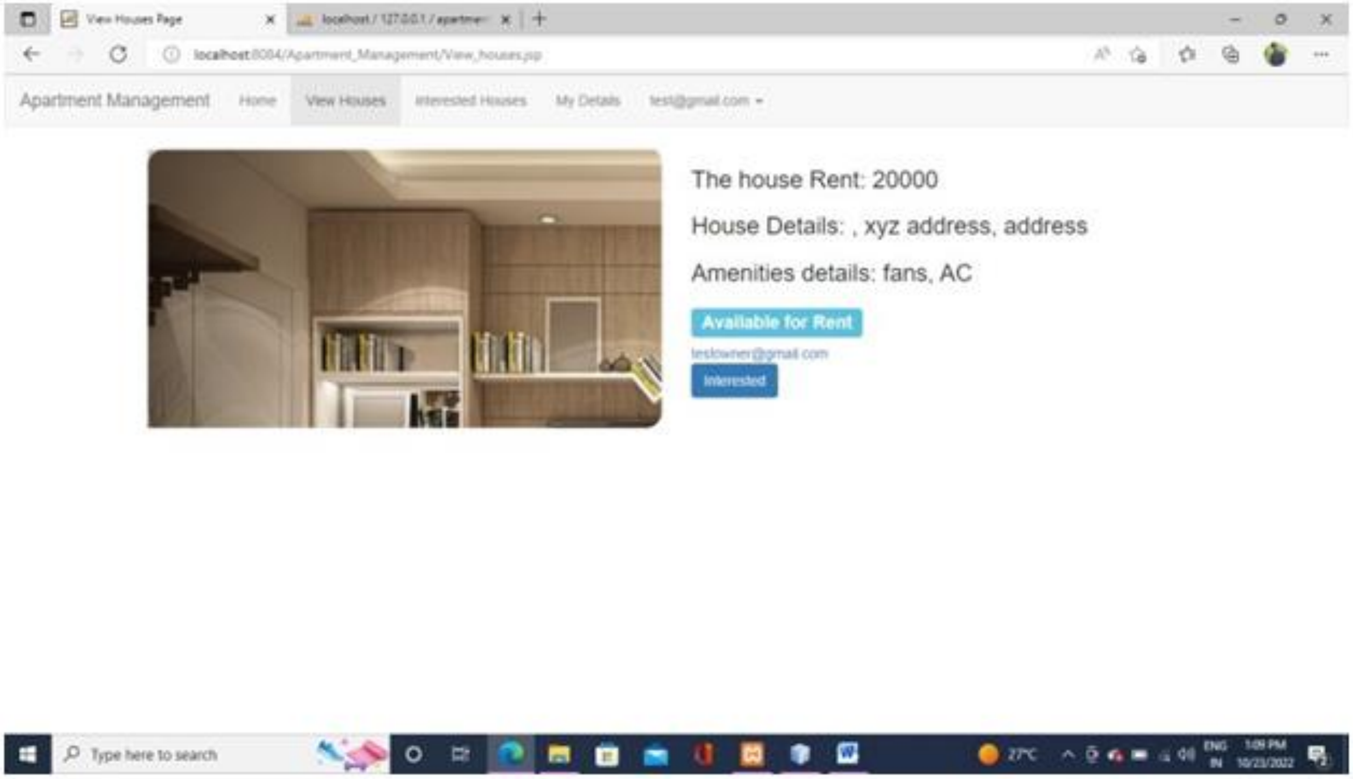
Phone Number:
67890123

Email:
testowner@gmail.com

submit

Figure 13: View and edit personal details

Tenant panel:



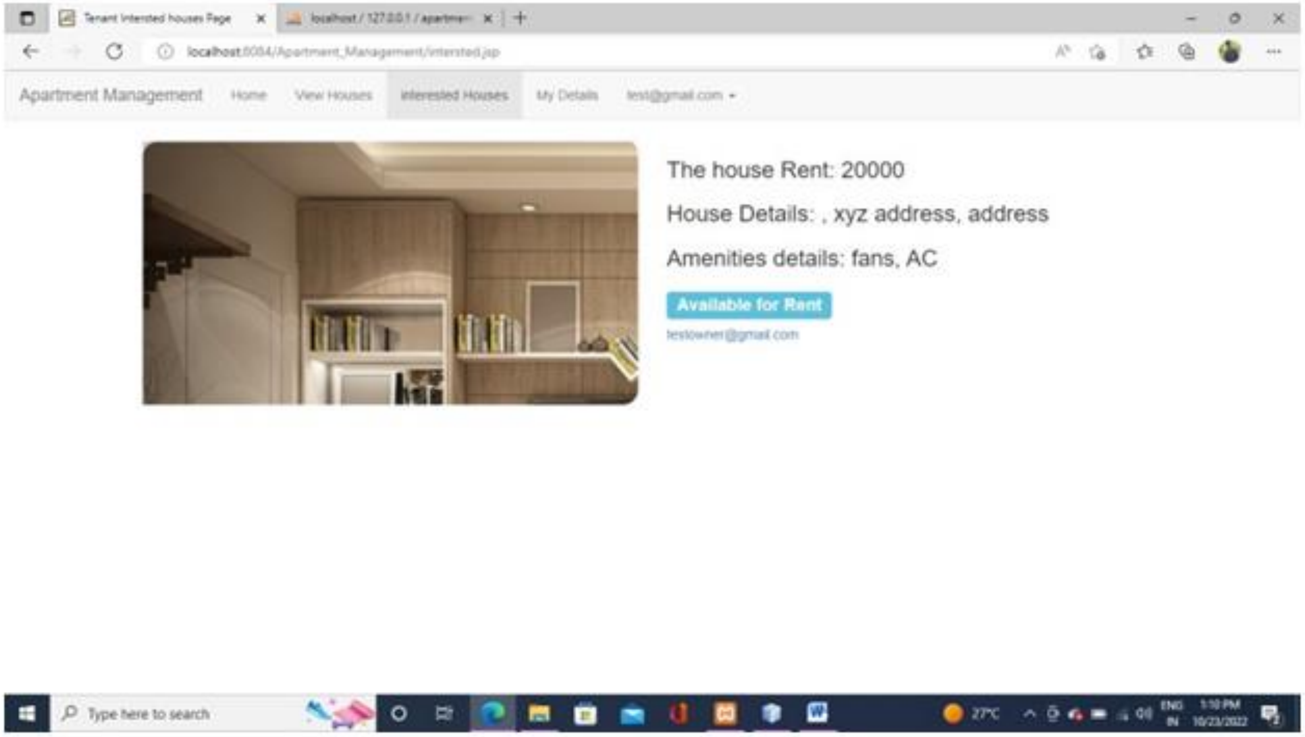
The house Rent: 20000

House Details: , xyz address, address

Amenities details: fans, AC

Figure 14: View houses available

We can email directly on the website
Interested page:



The house Rent: 20000

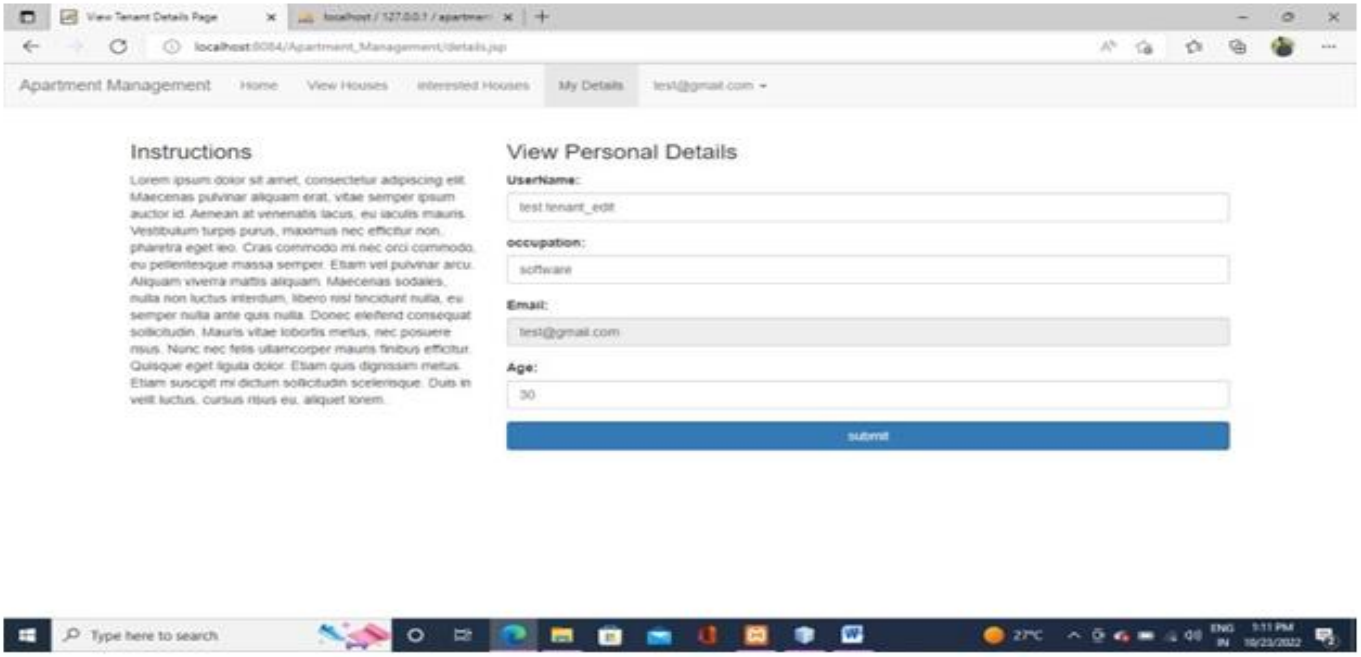
House Details: , xyz address, address

Amenities details: fans, AC

testowner@gmail.com

Figure 15: View Interested page

View and edit personal details:



Instructions

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas pulvinar aliquam erat, vitae semper ipsum auctor id. Aenean at venenatis lacus, eu iaculis mauris. Vestibulum turpis purus, maximus nec efficitur non, pharetra eget leo. Cras commodo mi nec orci commodo, eu pellentesque massa semper. Etiam vel pulvinar arcu. Aliquam viverra mattis aliquam. Maecenas sodales, nulla non luctus interdum, libero nisl tincidunt nulla, eu semper nulla ante quis nulla. Donec eleifend consequat sollicitudin. Mauris vitae lobortis metus, nec posuere risus. Nunc nec felis ullamcorper mauris finibus efficitur. Quisque eget ligula dolor. Etiam quis dignissim metus. Etiam suscipit mi dictum sollicitudin scelerisque. Duis in velit luctus, cursus risus eu, aliquet lorem.

View Personal Details

UserName:

test tenant_edit

occupation:

software

Email:

test@gmail.com

Age:

30

submit

Figure 16: View and edit personal details

7. *Open Issues*

As the application requirements are familiar, identifying the requirements was easy. But when implementing the operations, sometimes clarity was missing. To get the clarity, we have contacted different owners and tenants and gathered in the information regarding their needs. With this activity all the needed requirements are gathered and implemented.

8. *Acknowledgements*

We thank our project mentor Dr. Yunchuan Liu for constant encouragement in completing this project on time. We thank everyone who supported us in implementing this project. This project plays a vital role in our academic career.

9. *References*

[1]. Ali, S. H., & Azmi, H. (2017). Development Of Online Management system for Apartment. *Journal of Computing Technologies and Creative Content (JTec)*, *2*(1), 59-62. *EESSMod@ MoDELS* (pp. 3-12).

[2]. Higuma, T., Inoue, M., Nanjo, K., Suzuki, S., & Kobayashi, T. (1994). System architecture and interface for a Management system for Apartment. *IEEE Transactions on Consumer Electronics*, *40*(3), 411-417.

[3]. Jang, J. M., Oh, S. W., Kim, J. H., & Kim, Y. S. (2007). Development of a Web-based A/S Management System for Apartment Housing Projects. *Korean Journal of Construction Engineering and Management*, *8*(3), 66-75.

[4]. Kelley, E. N. (2009). *Practical* Management system for Apartment. Institute of Real Estate Ma.
Reggio, G., Leotta, M., Ricca, F., & Clerissi, D. (2013, October). What are the used UML diagrams? A Preliminary Survey.

10. *Appendices*

https://www.w3schools.com/sql/

https://www.w3schools.com/java/java_getstarted.asp

https://www.w3schools.com/bootstrap4/default.asp