Mini Project Report on
“Property Management System”

Submitted by :

Name	**Roll No**	**ERP No**
Pallavi Udatewar	**PH 08**	**S1032180183**
Rishabh Shinde	**PH 09**	**S1032180322**
Raj Patel	**PH 17**	**S1032180614**
Sarth Raut	**PH 64**	**S1032191805**

Under the Guidance of

Mrs. Vasundhara Ghate

At

Dr. Vishwanath Karad MIT WORLD PEACE UNIVERSITY | PUNE

TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

School of Computer Engineering and Technology

I. Abstract

The project focuses on providing Property Management to real estate agencies and civilians. This helps customer to save time and get best business solutions.
The real estate business deals with the development of the property and the lease, rent, sale of establishments. It is one of the fastest growing enterprises in India. It has potentially never ending growth.

MOTIVATION:-
As a real estate agent one has to maintain a lot of data. He/ She is involved with clients who has to lease out, rent or sale the property and with the customer who intends to buy, rent or lease the property. Hence it involves lot of information exchange.

The advent of computers can ease out this hassle. With the organized data storage system it allows faster search time, interaction and deal methods. Indeed the DBMS application can be a boom to the field of property management.

OBJECTIVES:-

- The admin should have all type authority.
- The admin should maintain property type and identify it as residential or commercial.
- To manage the registration details, approval details and types of properties.
- To make the system useful for companies or builders to post and edit their offers and availability of the property.
- Manage the information of buyers.
- Editing, adding and updating of records which will result in proper management of data and types of properties

II. Table of Figures

III. Table of Contents

1. Introduction

a. This document describes the requirements of the “PROPERTY MANAGEMENT SYSTEM”. It sets out the functional and non-functional requirements and includes a description of the user interface and documentation and training requirements. The main objective of the project is to create a tracking system for properties like houses and lands for buying purpose. Our proposed system serves as an aid for both property searchers and property holders.

b. The PROPERTY MANAGEMENT SYSTEM is highly interactive and taken as an adaptive approach as compared to the existing system along with a set of advanced tracking features. The product will be web based and interactive.

c. The product to be produced is a Property Management System which will automate the different property tracking and advertising. The system is helpful for normal users who search for properties and property holders. It helps the viewers to view the property pictures and etc.

d. The Property Management System will have the following features:

e. The system will handle all the activities for a real-estate organization.

f. The system will provide the users to search properties advertise property, buy property, even rent property, etc.

g. It will provide the administrator to control overall management of the system.

h. This is system will help people find the best property for them based on the location and criteria of property they specify.

i. It also targets to help users to get the best deal possible

2. Problem Definition

a. In the old existing Property Management System all information of the property or client proceeded manually and it had to maintain the record of the activity involved in manual system. At the time of searching the property all the records had to be scanned and even after that the people can’t be sure they will find a proper property.
b. It was unreliable and efficient data entry was not possible. Same data is maintained in various file which is leading to redundancy of data. Retrieval of required information was difficult and time consuming. Security of data is very critical issue which was not addressed.
c. The new database system provide solution to all the problems which was faced by traditional file system. It will keep record of housing properties available on rent or for sale, and will work as connecting bridge between customer and property sellers. The system is highly flexible one and is very efficient to make easy interactions with the client.
d. Due to this system there is no need to visit various places in search of desired property. The client has to specify what they are looking for and the system accordingly provides the information. The information of various places in search of desired property is can be accessible at one place and you can book an appointment and have a look at the property that you wish to buy.
e. This system will provide a platform for people to sell and buy property. This will keep record of property either commercial and residential for sale or on rent with their rates and make available for the customers. It will also keep record of contact information of customer and send necessary notices and/or reminders to the customers.
f. This will help users to list their property for sale and rent and assist users to find properties, in the their desired areas with necessary amenities at a great price.
g. It keeps record of all deals that take place and can be easily accessible by the admin.
h. It also provides additional services of linking the users to a broker and financial consultant.
i. It is a great all in one service that can be easily accessible by users

3. Tools and Technologies Used

a. Tools:

- Python 3.8
- MySQL
- Anaconda IDE
- Spyder
- MySQL Workbench 8
- MySQL Command Client
- PyOt
- PySide
- Qt for Python
- Tkinter
- PHP
- CSS
- Java
- Eclipse
- JSON

b. Technologies:

- Windows OS
- Mac OS
- Linux/ Ubuntu

4. Database Design(Entity Relationship Diagram)

Entities:

a. Broker

b. Financial Consultant

c. Owner
   - i. Seller
   - ii. Landlord

d. Customer
   - i. Buyer
   - ii. Tenant

e. Property
   - i. Residential Property
   - ii. Commercial Property

f. Registration

Relationship Set:

a. Broker and Owner
   - i. Cardinality: One to Many
     Relationship: Assists (Binary Relationship, Degree: 2)
     A broker can assist many Owners sell their property.

b. Broker and Customer
   - i. Cardinality: One to Many
     Relationship: Assists (Binary Relationship, Degree: 2)
     A broker can assist many Customers to choose the right property at good price.

c. Financial Consultant and Customer
   - i. Cardinality: One to Many
     Relationship: Advises (Binary Relationship, Degree: 2)
     A financial customer can advise a customer on whether or not he should invest in the project and also advises type of loan to take if required.

d. Owner and Property
   - i. Cardinality: One to Many
     Relationship: Lists (Binary Relationship, Degree: 2)
     An owner can list whichever property he likes for either sale or rent as per his choice. Owner can list many properties.

e. Customer and Property
   - i. Cardinality: One to Many
     Relationship: Looks at (Binary Relationship)
     A customer can look at as many properties as he like. He can sell or rent more than one property as well

f. Registration, Owner, Property and Customer
   - i. Aggregation
     Relationship (Degree 4 relationship)

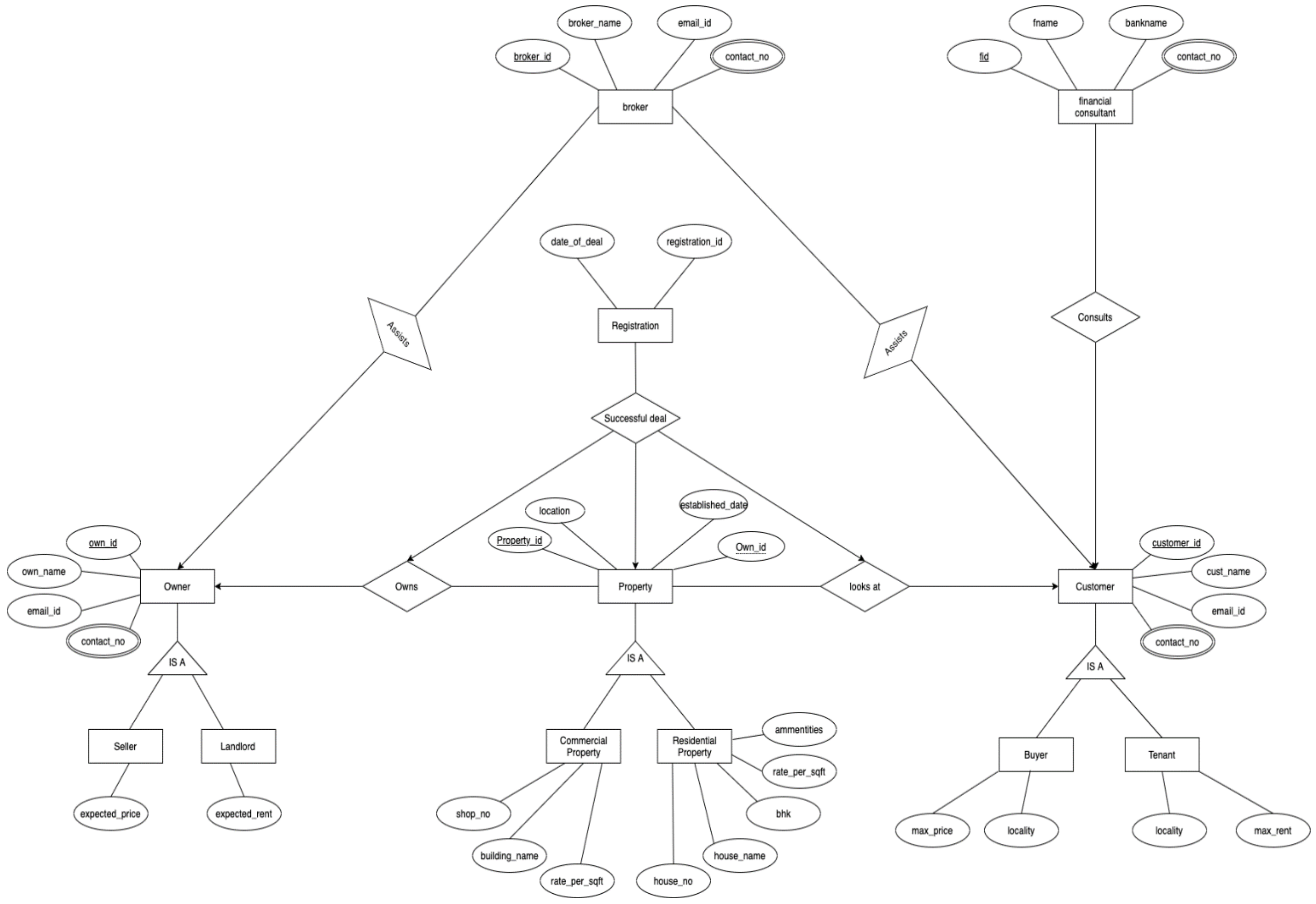


Figure 1 Entity Relationship Diagram

5. Database Schema

Owner(own_id, own_name, email, broker_id)

own_id	own_name	email	broker_id

Owner_phone (own_id , contact_no)

own_id	contact_no

Seller (own_id, expected_price, broker_id)

own_id	expected_price	broker_id

Renter (own_id, expected_rate, broker_id)

broker_id	expected_rate	broker_id

Property (property_id, location, established_date, own_id, customer_id)

property_id	location	established_date	own_id	customer_id

Commercial_Property (property_id, shop_no, building_name, rate_per_sqft, own_id, cutomer_id)

property_id	shop_no	building_name	rate_per_sqft	own_id	cutomer_id

Residential_Property (property_id, house_no, house_name, bhk, rate_per_sqft, ammenties, close_by_services, own_id, customer _id)

property_id	house_no	house_name	bhk	rate_per_sqft	ammenties

Close_by_services	own_id	customer _id

Customer (customer_id, cust_name, email_id, broker_id, fid)

customer_id	cust_name	email_id	broker_id	fid

Customer_phone (customer_id, contact_no)

customer_id	contact_no

Buyer(customer_id, max_price, locality, broker_id, fid)

customer_id	max_price	locality	broker_id	fid

Tenant(customer_id, max_rent, locality, broker_id, fid)

customer_id	max_rent	locality	broker_id	fid

Broker(broker_id, broker_name, email_id)

broker_id	broker_name	email_id

Broker_contact(broker_id, contact_no)

broker_id	contact_no

Financial_consultant(fid, fname, bankname)

fid	fname	bankname

Financial_contact(fid, contact_no)

fid	contact_no

Registration(registration_id,date_of_deal,own_id,property_id,customer_id)

registration_id	date_of_deal	own_id	property_id	customer_id

6. Relational Database Design using schema diagram

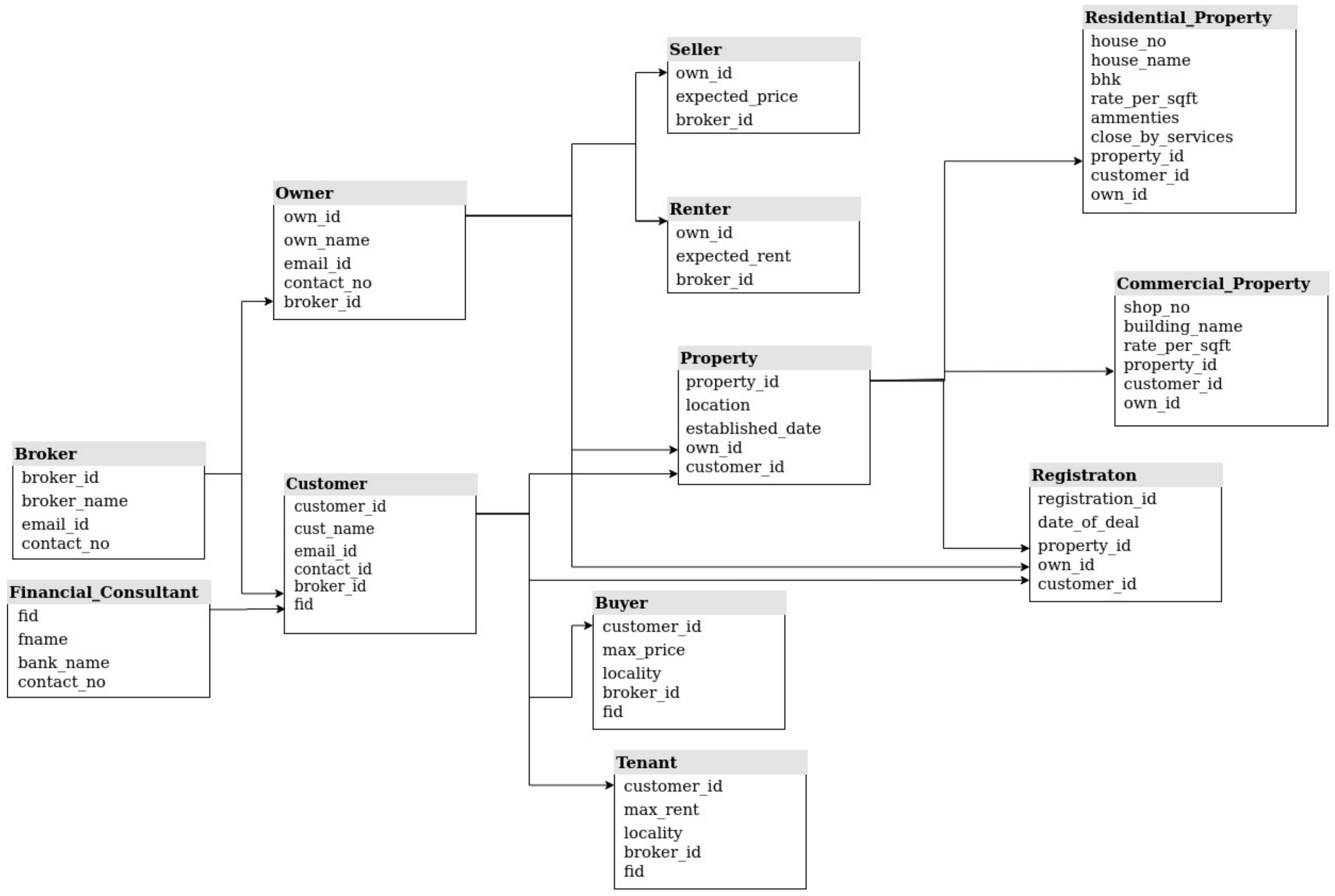


Figure 2 Schema Diagram

7. Database Normalization till 3 NF

There are no transitive or partial dependencies in any of the tables from the ER schema and they is highly atomic hence it satisfies 1NF, 2NF and 3NF state.
Therefore the schema is in completely normalised

Thus, most of the tables in the schema are normalized to 3NF

Eg:

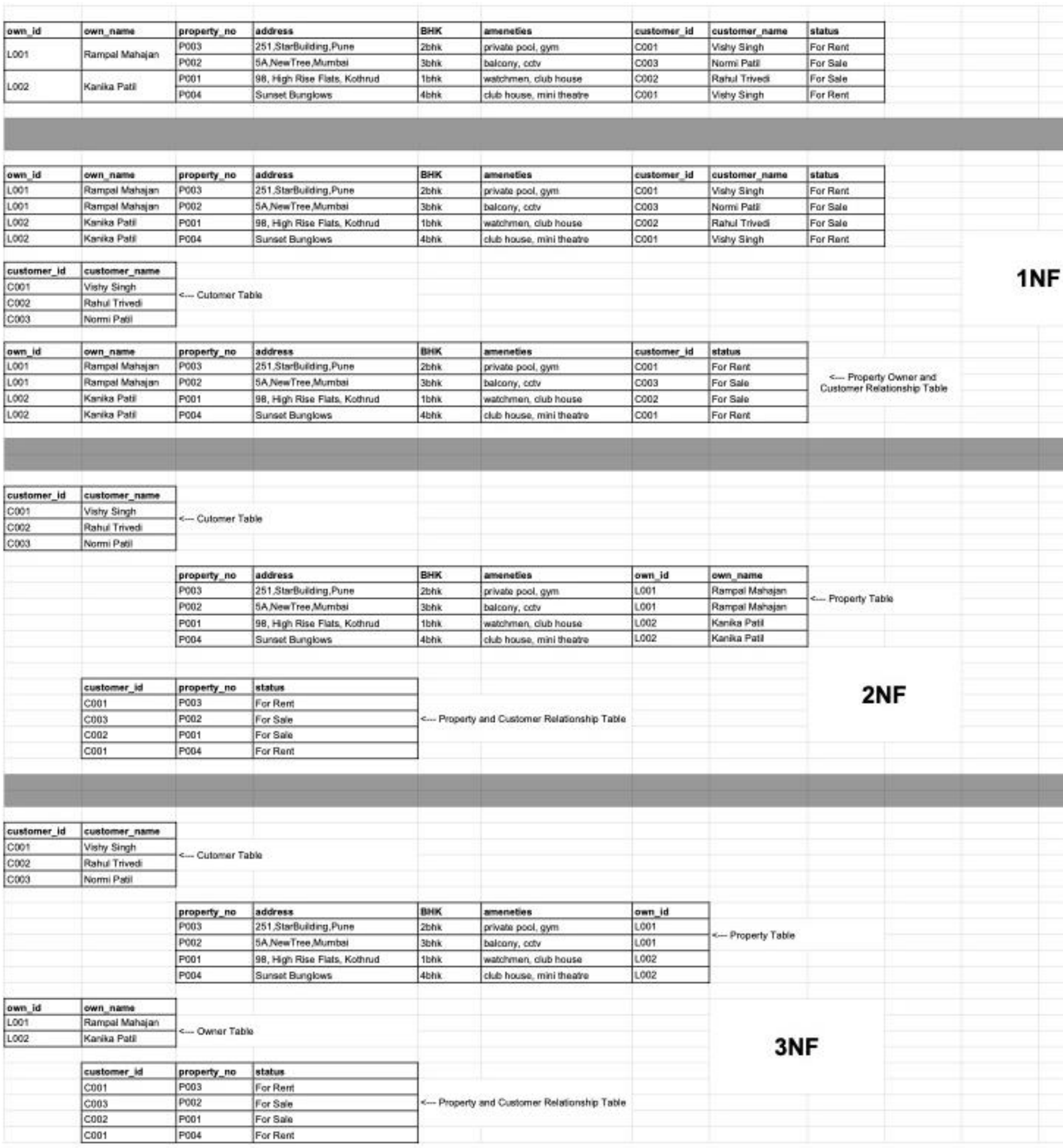
In the following table we see that:
own_id, cust_id -> own_name, property_no, address, BHK, amenities, customer_name, status
As we see the table is not atomic.
For owner L001 we have two 2 properties enlisted in the same row, therefore the table is not atomic.

We normalize it to 1NF by making separate rows for the entries
The table is now in the 1st Normal Form.

To check if the table has any partial dependencies, we find that there are 2 partial dependencies:
own_id -> own_name, property_no, address, BHK, amenities, status &
cust_id -> customer_name
By creating two separate tables and removing partial dependencies the table is now in 2nd Normal Form.

Next, is to check if there are any transitive dependencies.
As we can see the following transitive dependency exits:
own_id -> own_name, property_no
property_no -> address, BHK, amenities, status
We split these into separate tables and normalize it into 3rd Normal Form.

The steps and normalization procedure is shown in the figure below



own_id	own_name	property_no	address	BHK	ameneties	customer_id	customer_name	status
L001	Rampal Mahajan	P003	251,StarBuilding,Pune	2bhk	private pool, gym	C001	Vishy Singh	For Rent
		P002	5A,NewTree,Mumbai	3bhk	balcony, cctv	C003	Normi Patil	For Sale
L002	Kanika Patil	P001	98, High Rise Flats, Kothrud	1bhk	watchmen, club house	C002	Rahul Trivedi	For Sale
		P004	Sunset Bunglows	4bhk	club house, mini theatre	C001	Vishy Singh	For Rent

1NF

own_id	own_name	property_no	address	BHK	ameneties	customer_id	customer_name	status
L001	Rampal Mahajan	P003	251,StarBuilding,Pune	2bhk	private pool, gym	C001	Vishy Singh	For Rent
L001	Rampal Mahajan	P002	5A,NewTree,Mumbai	3bhk	balcony, cctv	C003	Normi Patil	For Sale
L002	Kanika Patil	P001	98, High Rise Flats, Kothrud	1bhk	watchmen, club house	C002	Rahul Trivedi	For Sale
L002	Kanika Patil	P004	Sunset Bunglows	4bhk	club house, mini theatre	C001	Vishy Singh	For Rent

customer_id	customer_name
C001	Vishy Singh
C002	Rahul Trivedi
C003	Normi Patil

<--- Cutomer Table

own_id	own_name	property_no	address	BHK	ameneties	customer_id	status
L001	Rampal Mahajan	P003	251,StarBuilding,Pune	2bhk	private pool, gym	C001	For Rent
L001	Rampal Mahajan	P002	5A,NewTree,Mumbai	3bhk	balcony, cctv	C003	For Sale
L002	Kanika Patil	P001	98, High Rise Flats, Kothrud	1bhk	watchmen, club house	C002	For Sale
L002	Kanika Patil	P004	Sunset Bunglows	4bhk	club house, mini theatre	C001	For Rent

<--- Property Owner and Customer Relationship Table

2NF

customer_id	customer_name
C001	Vishy Singh
C002	Rahul Trivedi
C003	Normi Patil

<--- Cutomer Table

property_no	address	BHK	ameneties	own_id	own_name
P003	251,StarBuilding,Pune	2bhk	private pool, gym	L001	Rampal Mahajan
P002	5A,NewTree,Mumbai	3bhk	balcony, cctv	L001	Rampal Mahajan
P001	98, High Rise Flats, Kothrud	1bhk	watchmen, club house	L002	Kanika Patil
P004	Sunset Bunglows	4bhk	club house, mini theatre	L002	Kanika Patil

<--- Property Table

customer_id	property_no	status
C001	P003	For Rent
C003	P002	For Sale
C002	P001	For Sale
C001	P004	For Rent

<--- Property and Customer Relationship Table

3NF

customer_id	customer_name
C001	Vishy Singh
C002	Rahul Trivedi
C003	Normi Patil

<--- Cutomer Table

property_no	address	BHK	ameneties	own_id
P003	251,StarBuilding,Pune	2bhk	private pool, gym	L001
P002	5A,NewTree,Mumbai	3bhk	balcony, cctv	L001
P001	98, High Rise Flats, Kothrud	1bhk	watchmen, club house	L002
P004	Sunset Bunglows	4bhk	club house, mini theatre	L002

<--- Property Table

own_id	own_name
L001	Rampal Mahajan
L002	Kanika Patil

<--- Owner Table

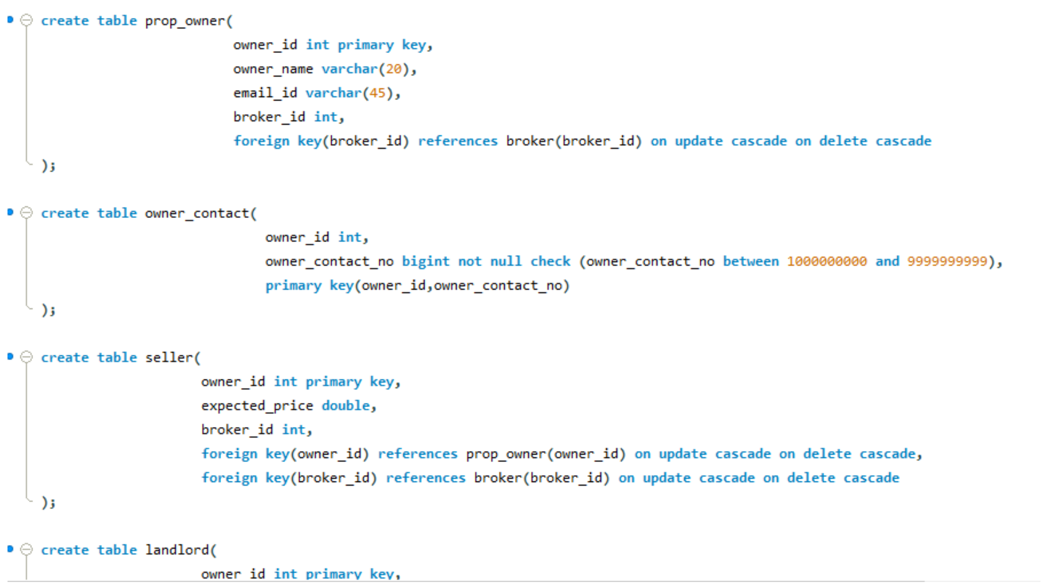
customer_id	property_no	status
C001	P003	For Rent
C003	P002	For Sale
C002	P001	For Sale
C001	P004	For Rent

<--- Property and Customer Relationship Table

Figure 3 Normalization to 3NF

8. DDL Commands

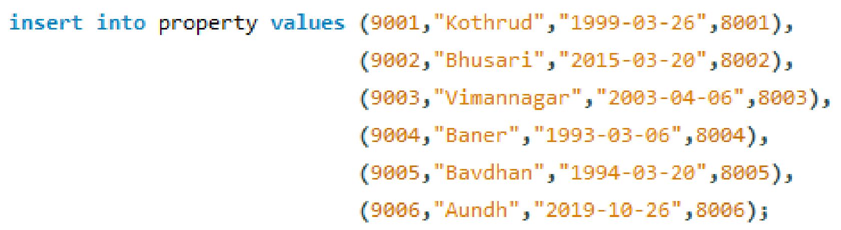
```
create table broker(
                                        broker_id int primary key,
             broker_name varchar(20) NOT NULL,
             email_id varchar(45) NOT NULL,
             brokerpassword text NOT NULL
);

create table prop_owner(
                                                owner_id int primary key,
                owner_name varchar(20),
                email_id varchar(45),
                broker_id int,
                foreign key(broker_id) references broker(broker_id) on update
cascade on delete cascade
);

create table property(
                                                property_id int primary key,
                location varchar(20),
                established_date date,
                owner_id int not null,
                foreign key(owner_id) references prop_owner(owner_id) on update
cascade on delete cascade
);
```



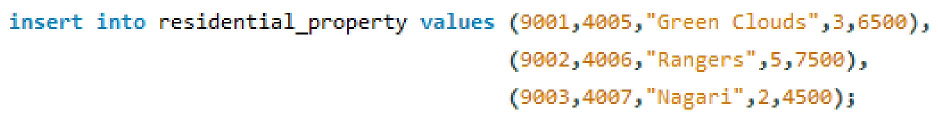
```
create table prop_owner(
                        owner_id int primary key,
                        owner_name varchar(20),
                        email_id varchar(45),
                        broker_id int,
                        foreign key(broker_id) references broker(broker_id) on update cascade on delete cascade
);

create table owner_contact(
                            owner_id int,
                            owner_contact_no bigint not null check (owner_contact_no between 1000000000 and 9999999999),
                            primary key(owner_id,owner_contact_no)
);

create table seller(
                    owner_id int primary key,
                    expected_price double,
                    broker_id int,
                    foreign key(owner_id) references prop_owner(owner_id) on update cascade on delete cascade,
                    foreign key(broker_id) references broker(broker_id) on update cascade on delete cascade
);

create table landlord(
                    owner_id int primary key,
```

9. DCL Commands

```
insert into prop_owner values (8001, "Pratap Kulkarni","Pratap@gmail.com",101),
                               (8002, "Manoj Kale","kale@gmail.com",103),
                              (8003, "Sudheer Shek","Sudher@gmail.com",102),
                              (8004, "Mehek Kumar","Kumarmehak@gmail.com",103),
                              (8005, "Yashami Joshi","Joshiyashmi@gmail.com",102),
                              (8006, "Ram Verma","vermaram@gmail.com",101);

insert into owner_contact values (8001,7854126394),
                                 (8001,9587412563),
                                 (8002,7458123694),
                                 (8003,7894561235),
                                 (8004,9568741236),
                                 (8005,9854216378),
                                 (8006,9857461321),
                                 (8006,8741259631);


insert into property values (9001,"Kothrud","1999-03-26",8001),
            (9002,"Bhusari","2015-03-20",8002),
            (9003,"Vimannagar","2003-04-06",8003),
            (9004,"Baner","1993-03-06",8004),
            (9005,"Bavdhan","1994-03-20",8005),
            (9006,"Aundh","2019-10-26",8006);
```



```
insert into property values (9001,"Kothrud","1999-03-26",8001),
                            (9002,"Bhusari","2015-03-20",8002),
                            (9003,"Vimannagar","2003-04-06",8003),
                            (9004,"Baner","1993-03-06",8004),
                            (9005,"Bavdhan","1994-03-20",8005),
                            (9006,"Aundh","2019-10-26",8006);
```



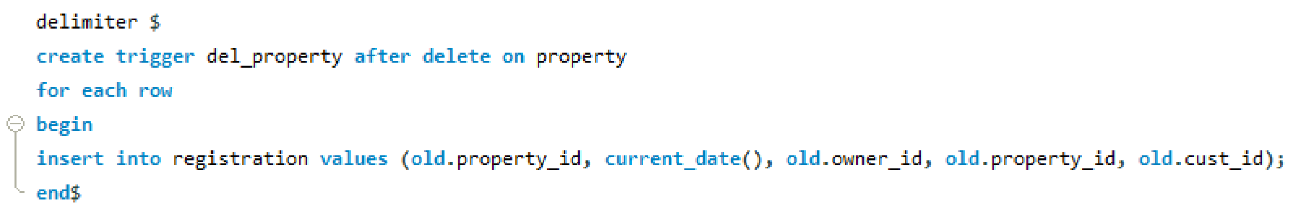
```
insert into residential_property values (9001,4005,"Green Clouds",3,6500),
                                        (9002,4006,"Rangers",5,7500),
                                        (9003,4007,"Nagari",2,4500);
```

```
create table ammeneties(
```

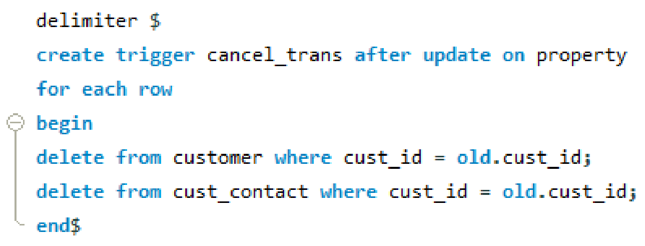
10. Triggers

```
delimiter $
create trigger del_property after delete on property
for each row
begin
insert into registration values (old.property_id, current_date(), old.owner_id,
old.property_id, old.cust_id);
end$
```

```
delimiter $
create trigger cancel_trans after update on property
for each row
begin
delete from customer where cust_id = old.cust_id;
delete from cust_contact where cust_id = old.cust_id;
end$
```



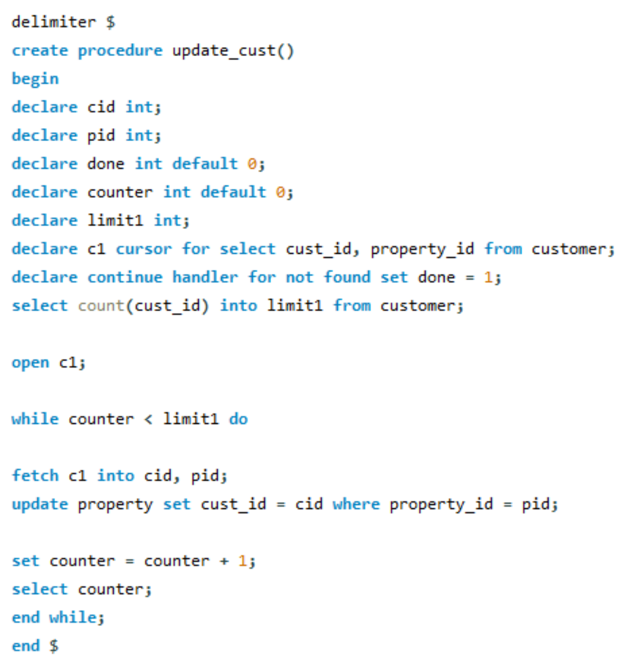
```
delimiter $
create trigger del_property after delete on property
for each row
begin
insert into registration values (old.property_id, current_date(), old.owner_id, old.property_id, old.cust_id);
end$
```



```
delimiter $
create trigger cancel_trans after update on property
for each row
begin
delete from customer where cust_id = old.cust_id;
delete from cust_contact where cust_id = old.cust_id;
end$
```

11. Procedure

```
delimiter $
create procedure update_cust()
begin
declare cid int;
declare pid int;
declare done int default 0;
declare counter int default 0;
declare limit1 int;
declare c1 cursor for select cust_id, property_id from customer;
declare continue handler for not found set done = 1;
select count(cust_id) into limit1 from customer;

open c1;

while counter < limit1 do

fetch c1 into cid, pid;
update property set cust_id = cid where property_id = pid;

set counter = counter + 1;
select counter;
end while;
end $
```



```
delimiter $
create procedure update_cust()
begin
declare cid int;
declare pid int;
declare done int default 0;
declare counter int default 0;
declare limit1 int;
declare c1 cursor for select cust_id, property_id from customer;
declare continue handler for not found set done = 1;
select count(cust_id) into limit1 from customer;

open c1;

while counter < limit1 do

fetch c1 into cid, pid;
update property set cust_id = cid where property_id = pid;

set counter = counter + 1;
select counter;
end while;
end $
```

12. Frontend GUI screenshots

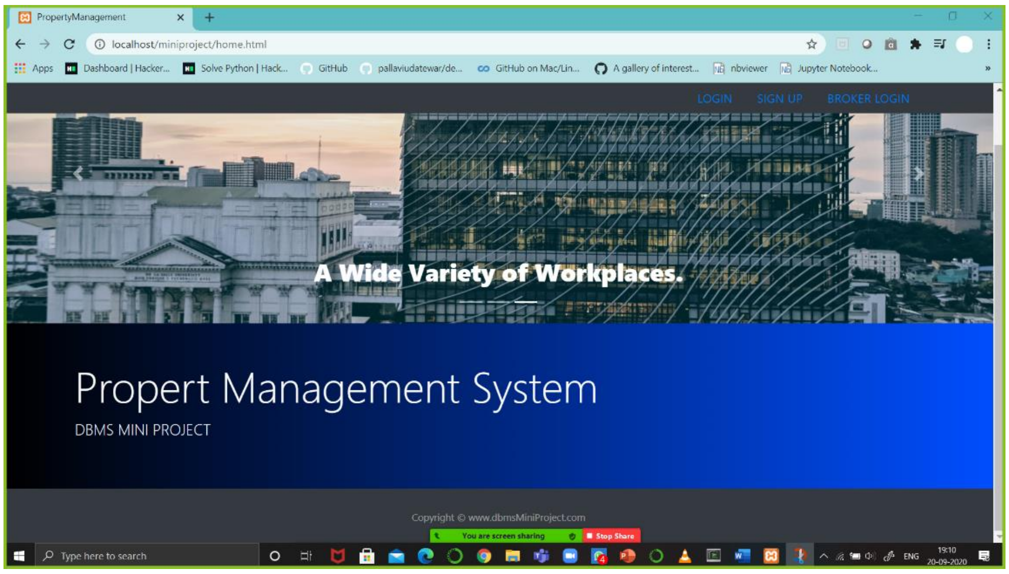

LOGIN
SIGN UP
BROKER LOGIN
A Wide Variety of Workplaces.
Propert Management System
DBMS MINI PROJECT
Copyright © www.dbmsMiniProject.com

Figure 4 Home Page

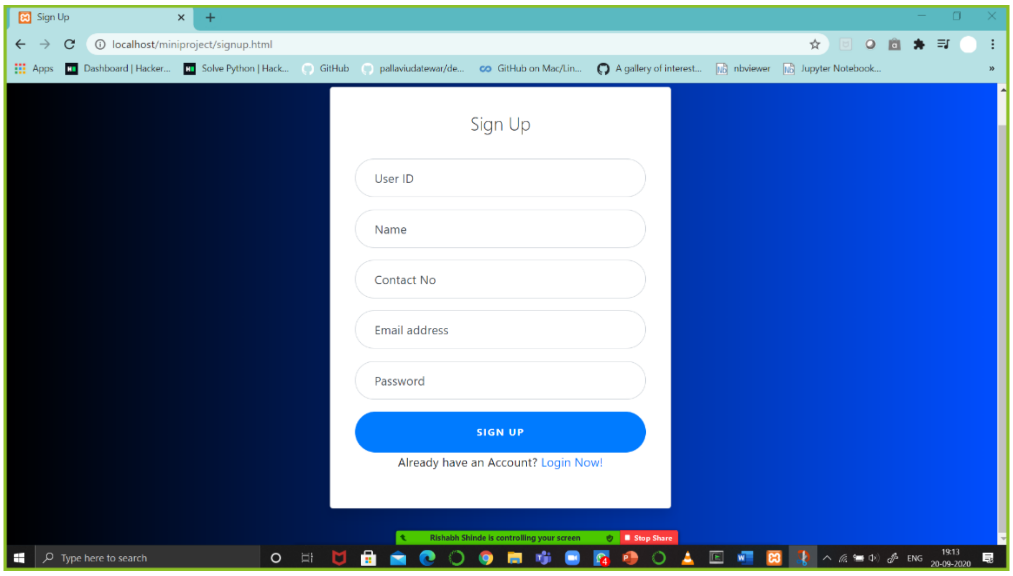
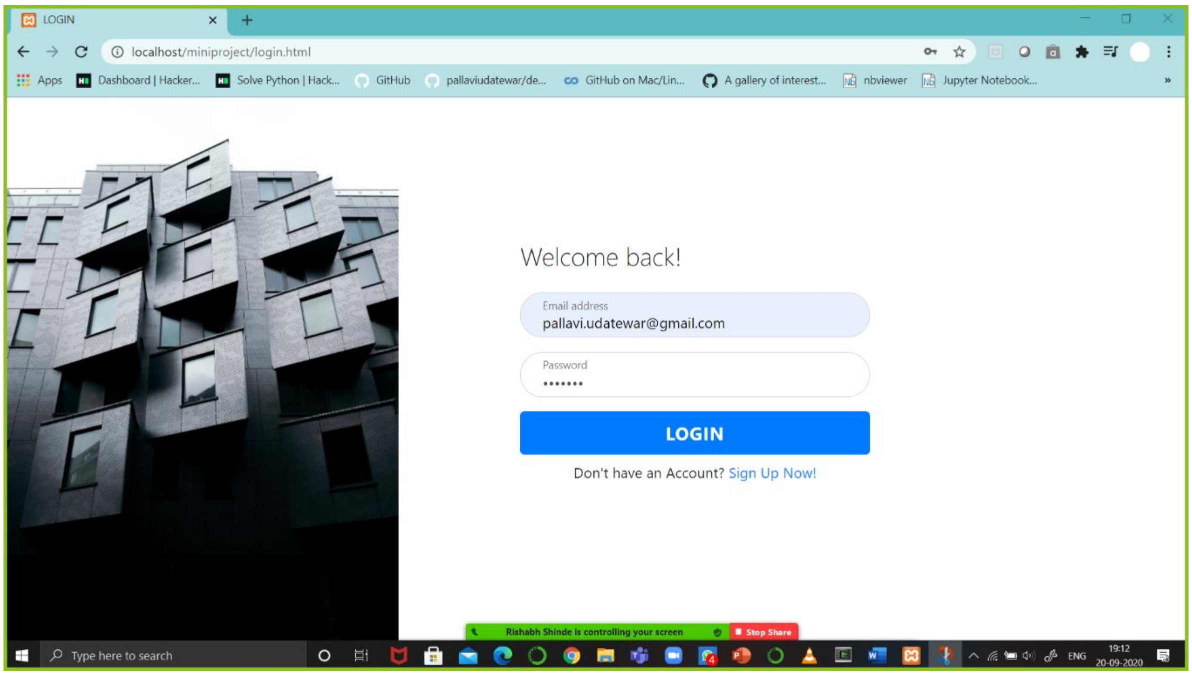


Figure 5 Sign Up



Welcome back!

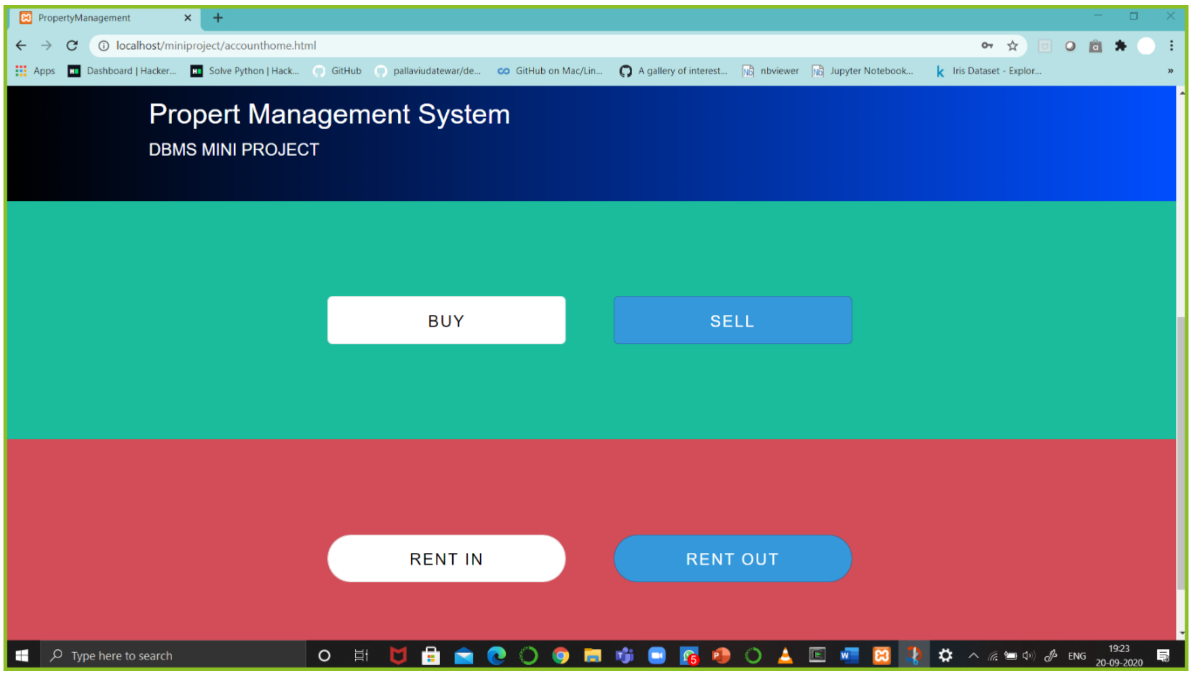
Email address
pallavi.udatewar@gmail.com

Password
•••••••

LOGIN

Don't have an Account? Sign Up Now!

Figure 6 Login



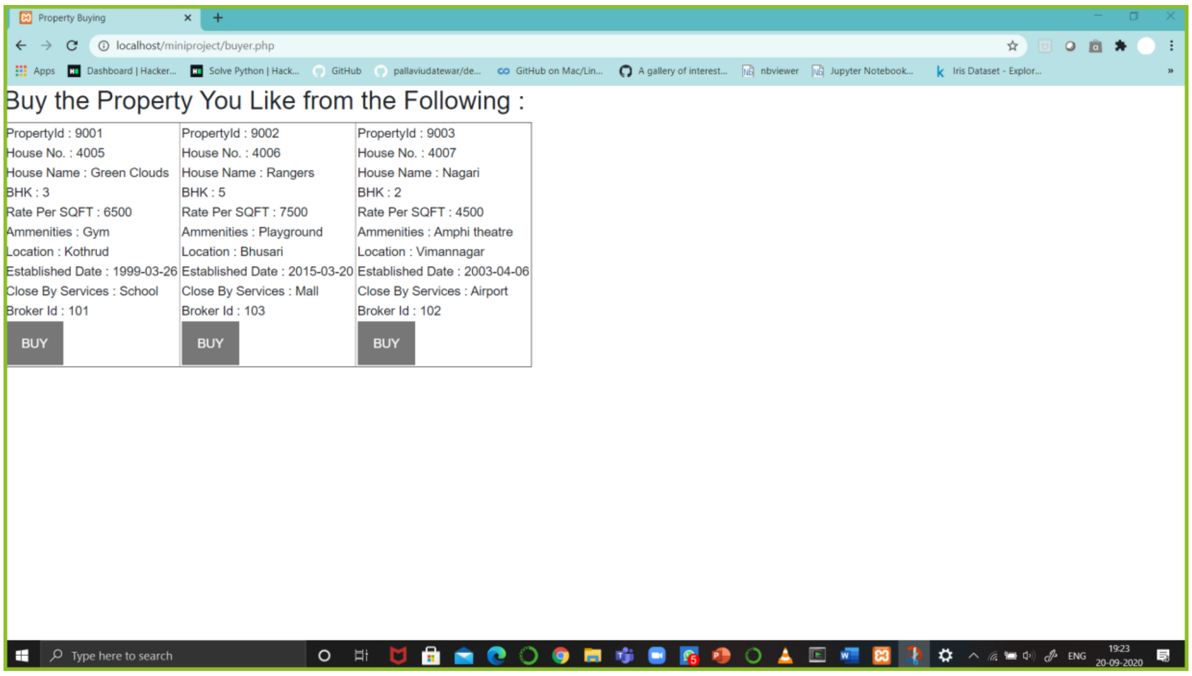
Propert Management System

DBMS MINI PROJECT

BUY SELL

RENT IN RENT OUT

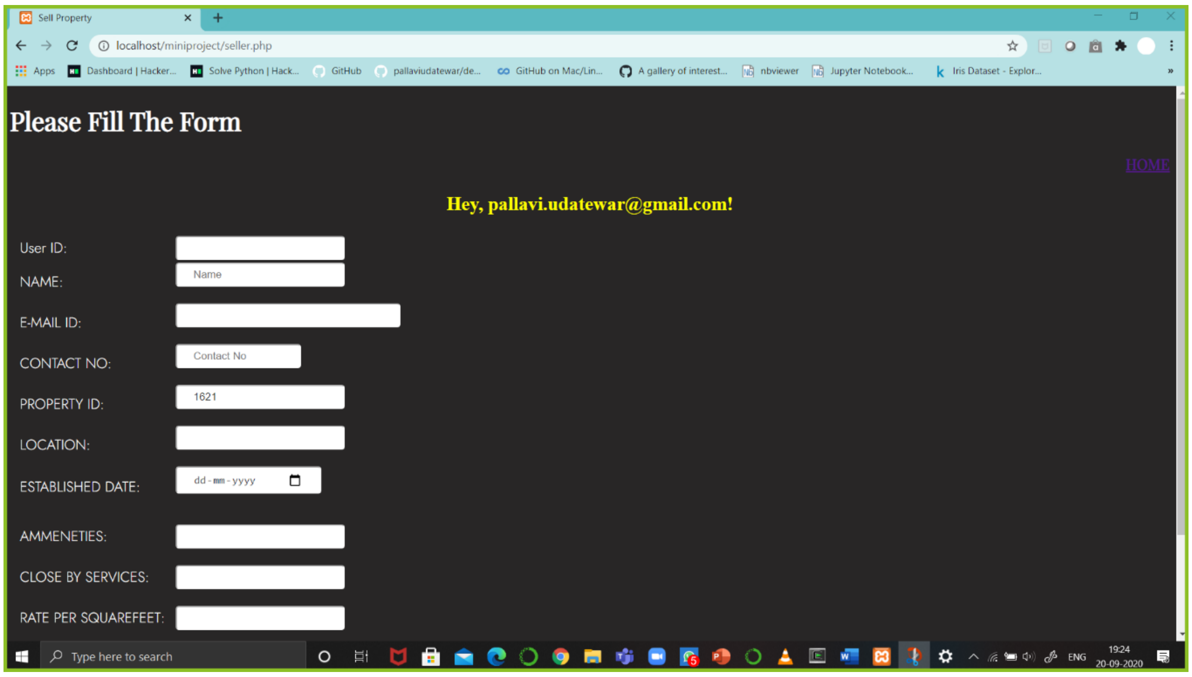
Figure 7 Home Page



Buy the Property You Like from the Following :

PropertyId : 9001	PropertyId : 9002	PropertyId : 9003
House No. : 4005	House No. : 4006	House No. : 4007
House Name : Green Clouds	House Name : Rangers	House Name : Nagari
BHK : 3	BHK : 5	BHK : 2
Rate Per SQFT : 6500	Rate Per SQFT : 7500	Rate Per SQFT : 4500
Ammenities : Gym	Ammenities : Playground	Ammenities : Amphi theatre
Location : Kothrud	Location : Bhusari	Location : Vimannagar
Established Date : 1999-03-26	Established Date : 2015-03-20	Established Date : 2003-04-06
Close By Services : School	Close By Services : Mall	Close By Services : Airport
Broker Id : 101	Broker Id : 103	Broker Id : 102
BUY	BUY	BUY

Figure 8 Buyer



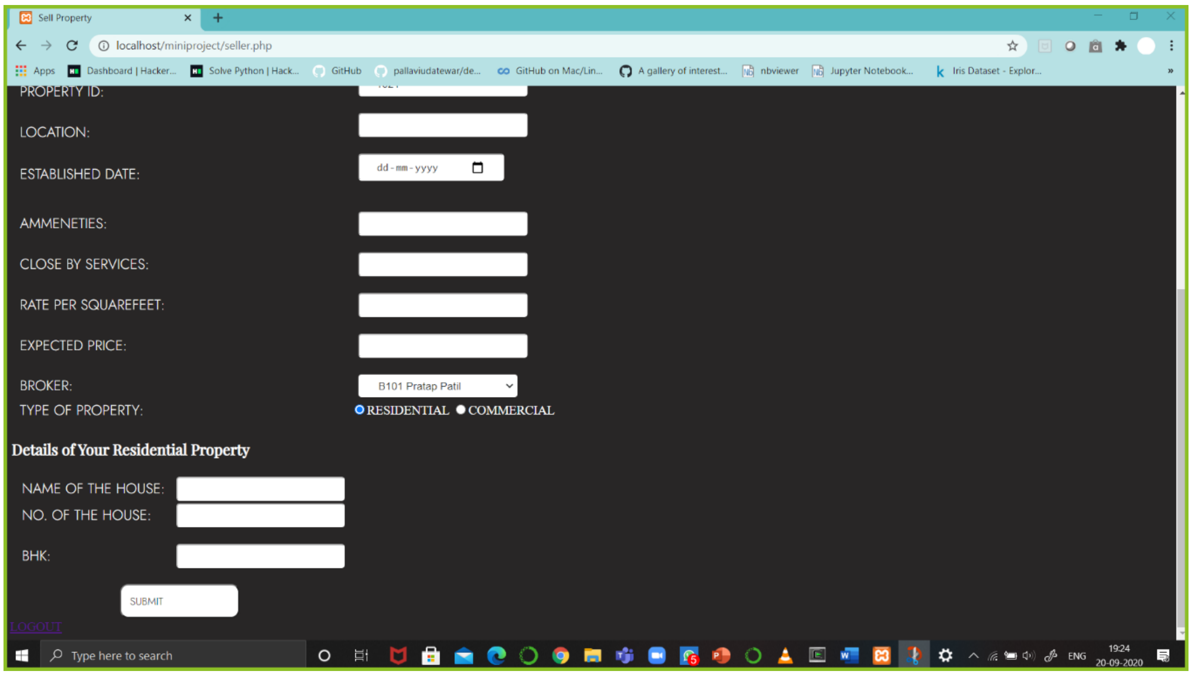
Please Fill The Form

HOME

Hey, pallavi.udatewar@gmail.com!

User ID:	
NAME:	Name
E-MAIL ID:	
CONTACT NO:	Contact No
PROPERTY ID:	1621
LOCATION:	
ESTABLISHED DATE:	dd-mm-yyyy
AMMENETIES:	
CLOSE BY SERVICES:	
RATE PER SQUAREFEET:	

Figure 9 Seller

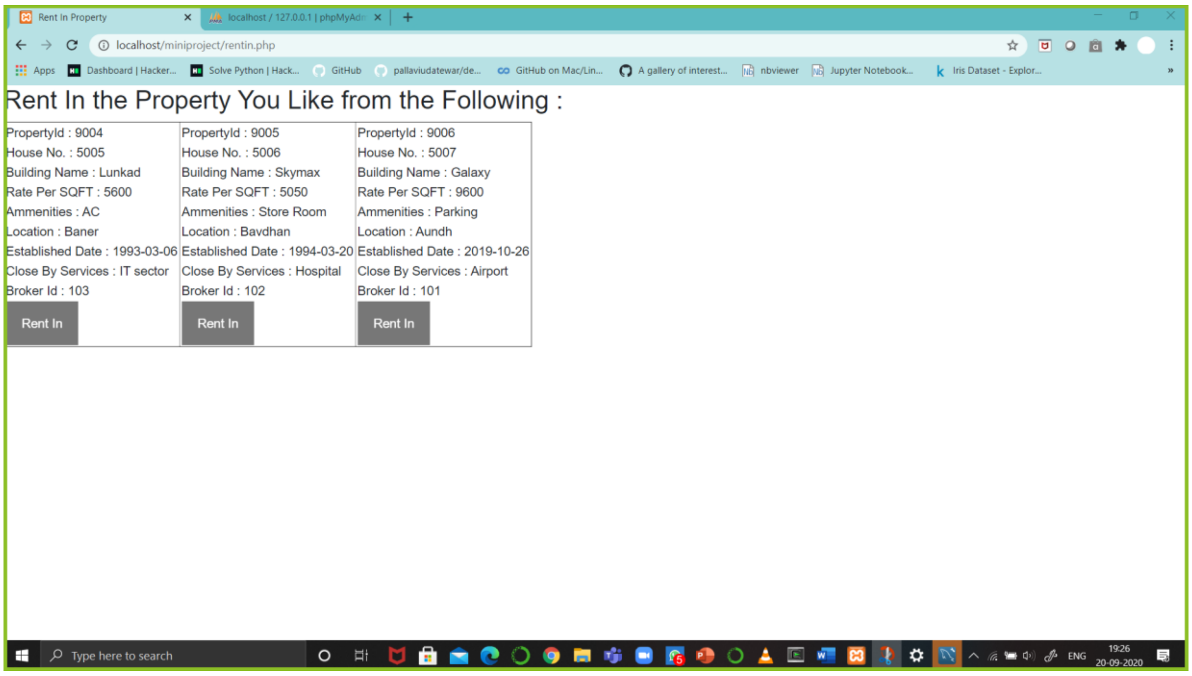


PROPERTY ID:	
LOCATION:	
ESTABLISHED DATE:	dd-mm-yyyy
AMMENETIES:	
CLOSE BY SERVICES:	
RATE PER SQUAREFEET:	
EXPECTED PRICE:	
BROKER:	B101 Pratap Patil
TYPE OF PROPERTY:	[✓ RESIDENTIAL] COMMERCIAL

Details of Your Residential Property

NAME OF THE HOUSE:	
NO. OF THE HOUSE:	
BHK:	

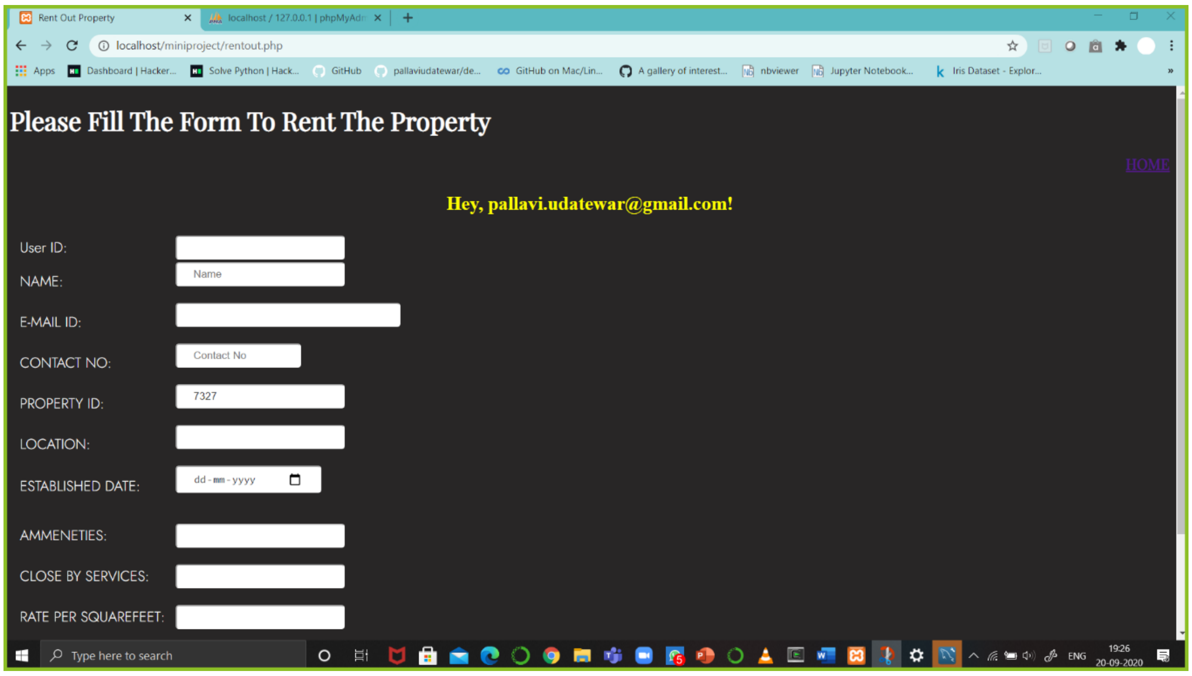
Figure 10 Seller


Rent In Property
localhost / 127.0.0.1 | phpMyAdmin
localhost/miniproject/rentin.php

Rent In the Property You Like from the Following :

PropertyId : 9004	PropertyId : 9005	PropertyId : 9006
House No. : 5005	House No. : 5006	House No. : 5007
Building Name : Lunkad	Building Name : Skymax	Building Name : Galaxy
Rate Per SQFT : 5600	Rate Per SQFT : 5050	Rate Per SQFT : 9600
Ammenities : AC	Ammenities : Store Room	Ammenities : Parking
Location : Baner	Location : Bavdhan	Location : Aundh
Established Date : 1993-03-06	Established Date : 1994-03-20	Established Date : 2019-10-26
Close By Services : IT sector	Close By Services : Hospital	Close By Services : Airport
Broker Id : 103	Broker Id : 102	Broker Id : 101
Rent In	Rent In	Rent In

Figure 11 Rent In



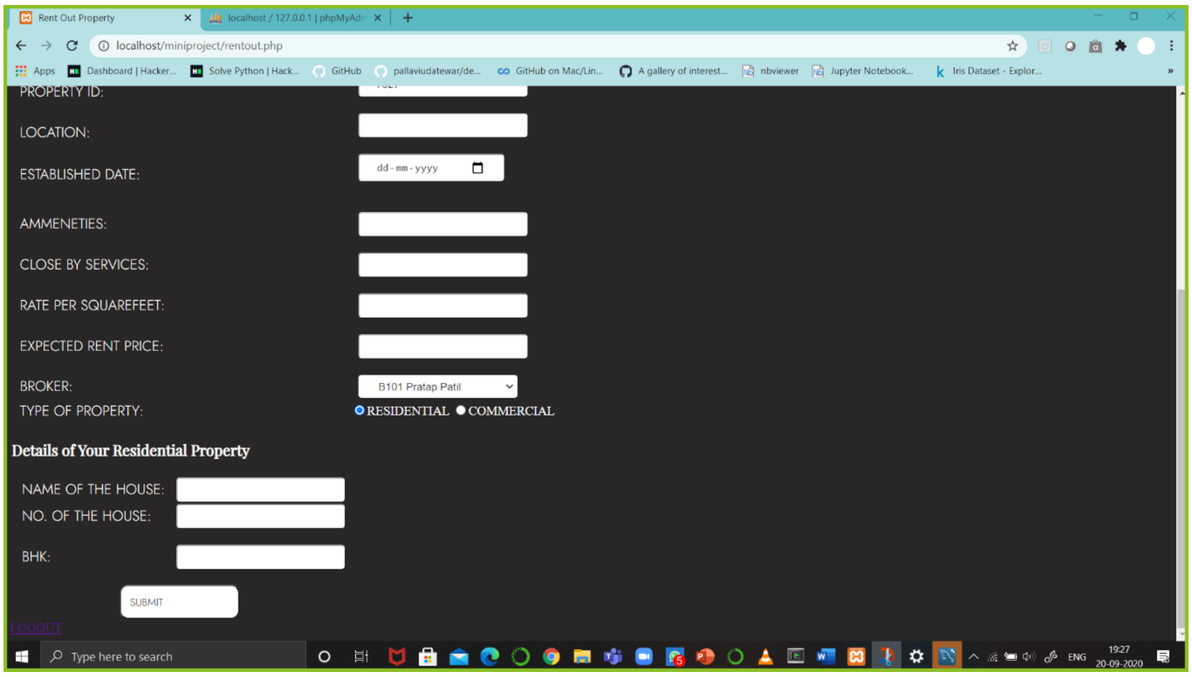
Please Fill The Form To Rent The Property

HOME

Hey, pallavi.udatewar@gmail.com!

User ID:	
NAME:	Name
E-MAIL ID:	
CONTACT NO:	Contact No
PROPERTY ID:	7327
LOCATION:	
ESTABLISHED DATE:	dd-mm-yyyy
AMMENETIES:	
CLOSE BY SERVICES:	
RATE PER SQUAREFEET:	

Figure 12 Rent In



PROPERTY ID:	
LOCATION:	
ESTABLISHED DATE:	dd-mm-yyyy
AMMENETIES:	
CLOSE BY SERVICES:	
RATE PER SQUAREFEET:	
EXPECTED RENT PRICE:	
BROKER:	B101 Pratap Patil
TYPE OF PROPERTY:	[✓ RESIDENTIAL] COMMERCIAL

Details of Your Residential Property

NAME OF THE HOUSE:	
NO. OF THE HOUSE:	
BHK:	

SUBMIT

LOGOUT

Figure 13 Rent In

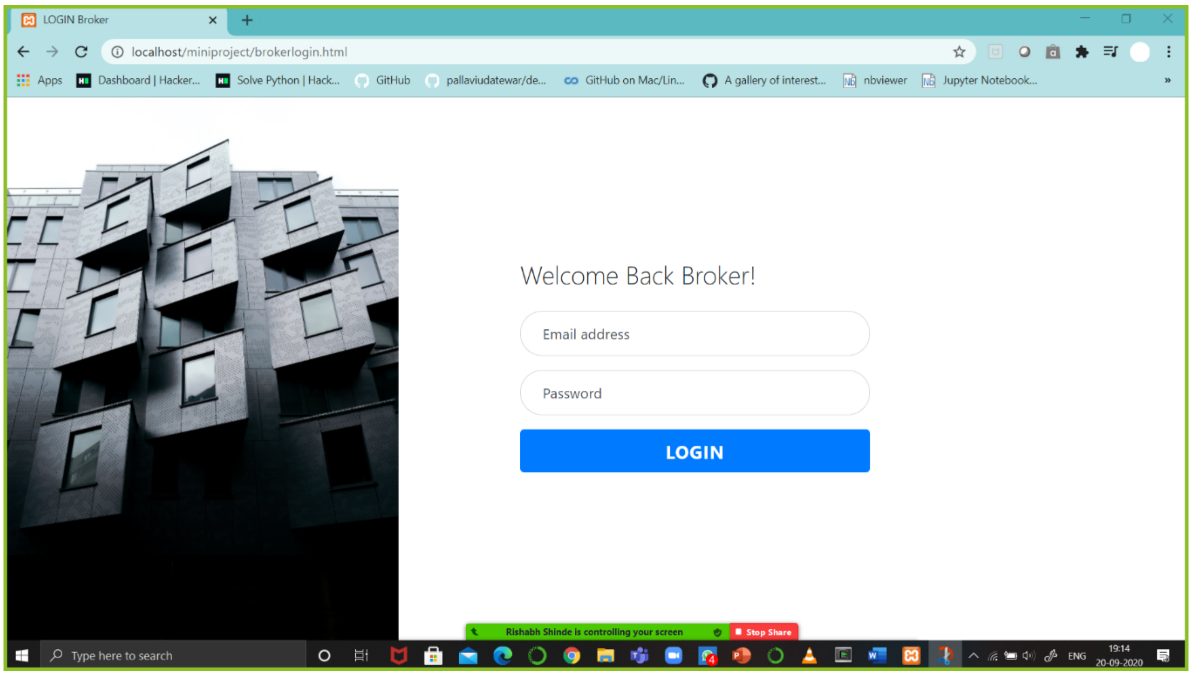
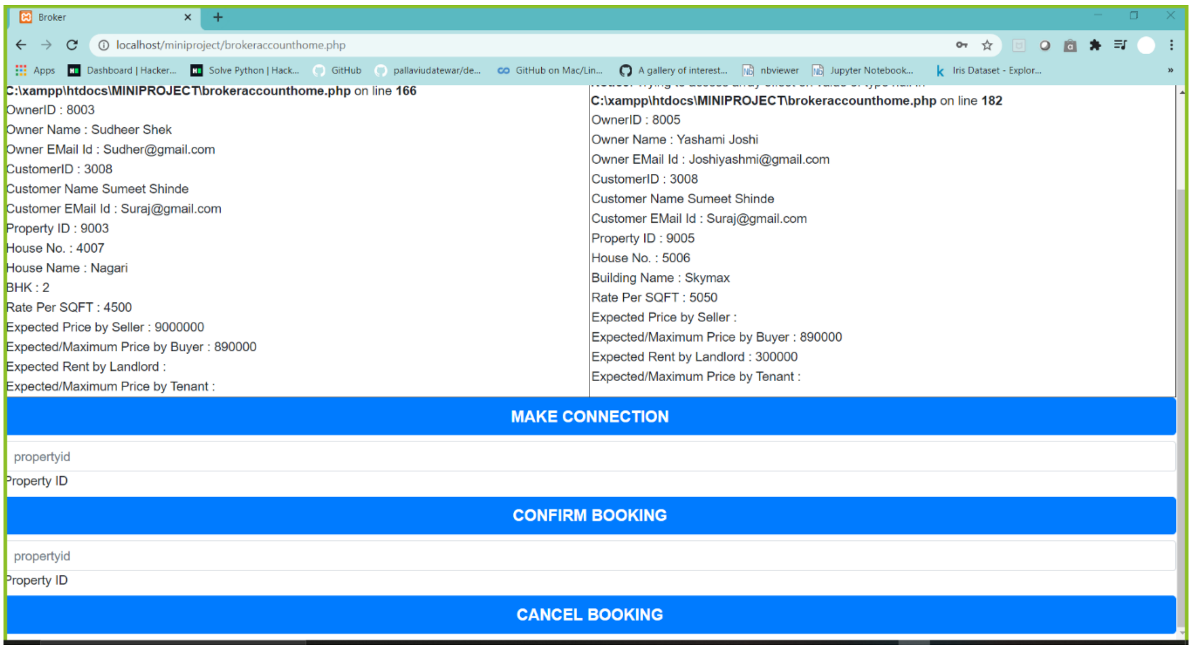


Figure 14 Broker Login



```
C:\xampp\htdocs\MINIPROJECT\brokeraccounthome.php on line 166
OwnerID : 8003
Owner Name : Sudheer Shek
Owner EMail Id : Sudher@gmail.com
CustomerID : 3008
Customer Name Sumeet Shinde
Customer EMail Id : Suraj@gmail.com
Property ID : 9003
House No. : 4007
House Name : Nagari
BHK : 2
Rate Per SQFT : 4500
Expected Price by Seller : 9000000
Expected/Maximum Price by Buyer : 890000
Expected Rent by Landlord :
Expected/Maximum Price by Tenant :
```

```
C:\xampp\htdocs\MINIPROJECT\brokeraccounthome.php on line 182
OwnerID : 8005
Owner Name : Yashami Joshi
Owner EMail Id : Joshiyashmi@gmail.com
CustomerID : 3008
Customer Name Sumeet Shinde
Customer EMail Id : Suraj@gmail.com
Property ID : 9005
House No. : 5006
Building Name : Skymax
Rate Per SQFT : 5050
Expected Price by Seller :
Expected/Maximum Price by Buyer : 890000
Expected Rent by Landlord : 300000
Expected/Maximum Price by Tenant :
```

Figure 15 Broker Home Page

13. Conclusion

Thus, we have implemented DBMS properties to create a property management system that helps people buy and sell their properties at a convenient price. We have successfully used MySQL at the backend and linked to the front end which we made in html using PHP.

14. References in IEEE format

https://www.w3schools.com/php/php_mysql_connect.asp
https://www.w3schools.com/howto/howto_css_signup_form.asp
https://dev.mysql.com/doc/
https://www.w3schools.com/howto/howto_css_login_form.asp
https://htmlcheatsheet.com/