DBMS - MINI PROJECT

"ONLINE HOUSE RENTAL SYSTEM"

RENT RIGHT NOW

Submitted By:

Name: HEMANTH KALYAN N HEGADE

SRN:PES1UG20CS163

V Semester Section _C

ABSTRACT

- House management has become important factor in modern society hence the need to have a house rental management system

- RENT RIGHT NOW is a House rental management website where house owners, agents and tenants can exchange information effectively and inexpensively.

- Provides user-friendly interface, satisfying the needs of the consumers.

- Employs a new strategy that facilitates easy management of rental houses.

Scope:

The project scope defines the description of the work that is required in delivering the rental house management system.

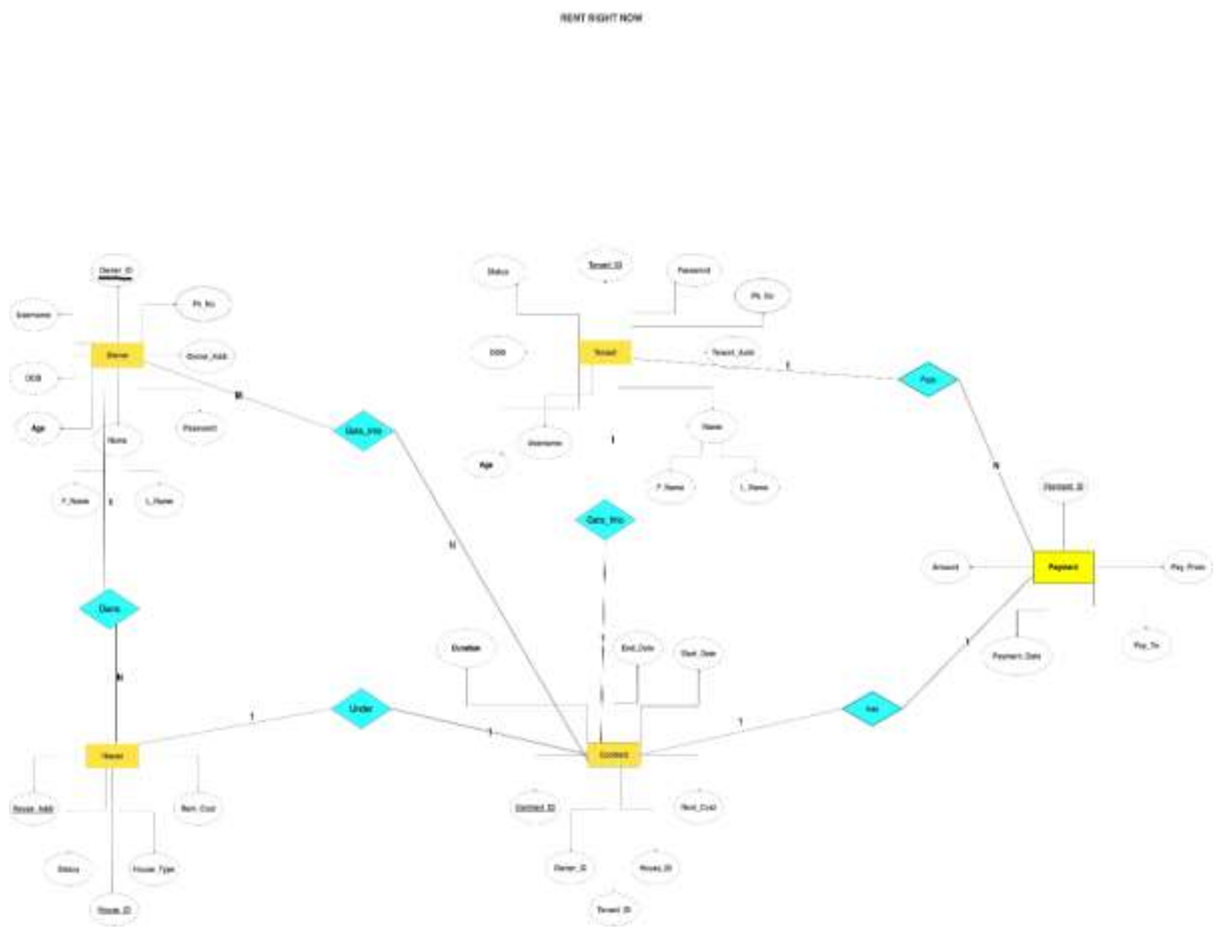
The following are the scopes of work during the course of the project:

- Study and understand the requirement of this project
- ER diagram and relational schema
- A minimalistic UI/front-end
- Creating and populating the database
- Using the SQL queries and show the reflections in the front-end developed

Modules Used:

- Owner: Displays the details of the Houses he/she owns for rent.
- Tenant: Searched the best suited house for the living and makes contract with the owner once he/she buys/rents the house.
- Agent: The owner chooses to act as an agent for better contracting communication between the owner and tenant and gain extra benefits.

ER Diagram

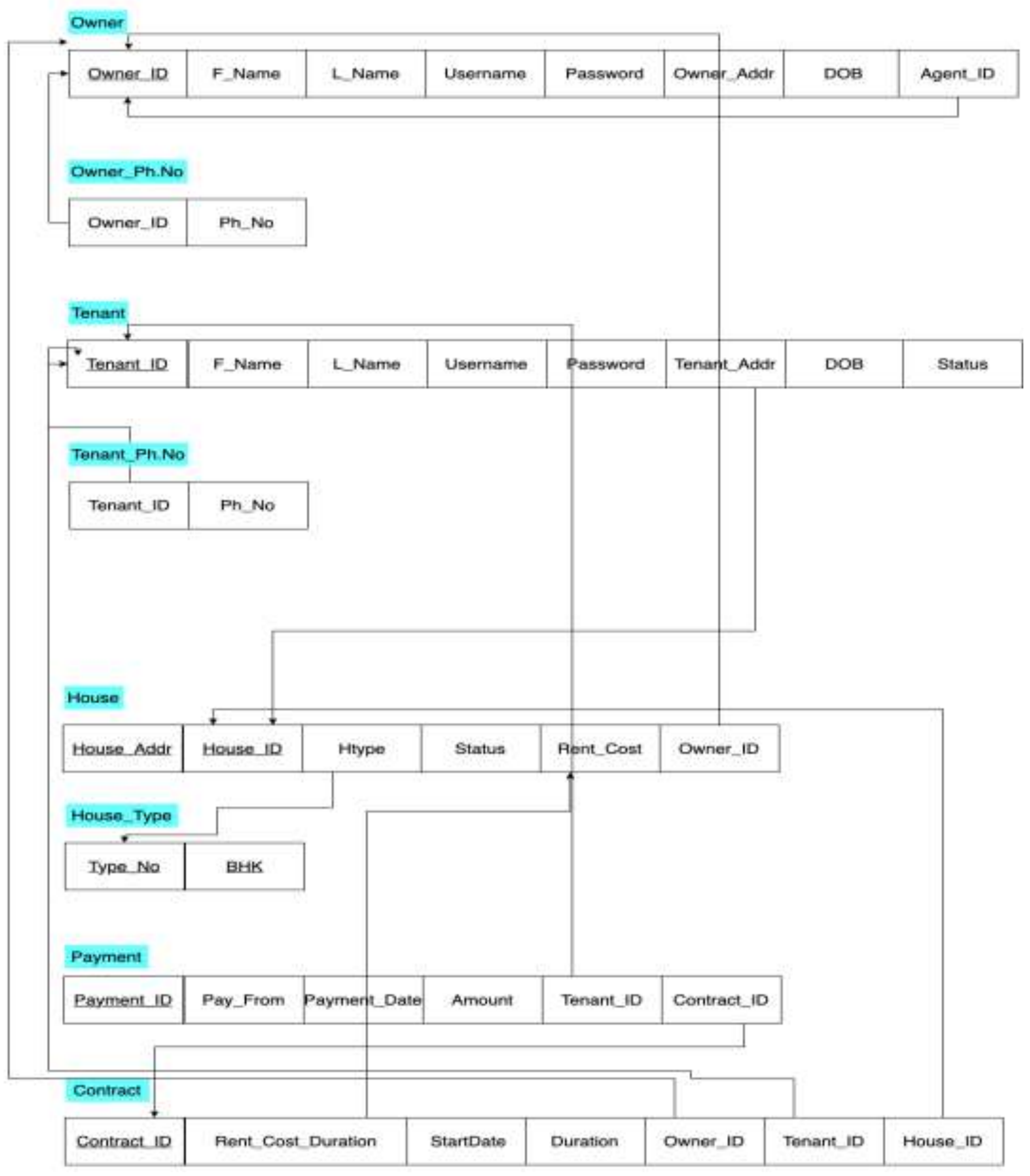

RENT RIGHT NOW
Owner_ID
Username
Ph_No
Owner
Owner_Add
DOB
Age
Name
Password
F_Name
L_Name
Gets_Info
Owns
House
House_Addr
Rent_Cost
Status
House_Type
House_ID
Under
Status
Tenant_ID
Password
Ph_No
DOB
Tenant
Tenant_Addr
Pays
Username
Age
Name
F_Name
L_Name
Gets_Info
Duration
End_Date
Start_Date
Contract
Contract_ID
Rent_Cost
Owner_ID
House_ID
Tenant_ID
Has
Payment_ID
Amount
Payment
Pay_From
Pay_To
Payment_Date
M
N
1
N
1
1
1
1
N
1

Relational Schema

Hemanth Kalyan N Hegade

PES1UG20CS163

Rent Right Now



DDL statements - Building the database

1.CREATING TABLE OWNER

```
CREATE TABLE IF NOT EXISTS Owner(
    Owner_ID int(6) auto_increment,
    primary key(Owner_ID),
    F_Name varchar(10) not null,
    L_Name varchar(10) not null,
    Username varchar(10) not null unique,
    Password varchar(10) not null,
    Owner_Addr varchar(50) unique,
    DOB date

);
```

2.CREATING TABLE OWNER_PHONE_NO

```
CREATE TABLE IF NOT EXISTS Owner_Ph(
Owner_ID int(6),
constraint fk_owner_id foreign key(Owner_ID) REFERENCES Owner(Owner_ID),
Ph_No char(10) not null
);
```

3.CREATING TABLE TENANT

```
CREATE TABLE IF NOT EXISTS Tenant(
    Tenant_ID int(6) auto_increment,
    primary key(Tenant_ID),
    F_Name varchar(10) not null,
    L_Name varchar(10) not null,
    Username varchar(10) not null unique,
    Password varchar(10) not null,
    Tenant_Addr varchar(50) default NULL,
    -- constraint fk_tenant_addr foreign key(Tenant_Addr) REFERENCES
House(House_Addr),
    DOB date,
    Status boolean default false
);
```

4.CREATING TABLE TENANT PHONE_NO

```sql
CREATE TABLE IF NOT EXISTS Tenant_Ph(
    Tenant_ID int(6),
    constraint fk_tenant_id foreign key(Tenant_ID) REFERENCES 
Tenant(Tenant_ID),
    Ph_No char(10) not null
);
```

5.CREATING TABLE HOUSE TYPE

```sql
CREATE TABLE IF NOT EXISTS House_Type(
    Type_No int(2),
    BHK varchar(6),
    primary key(Type_No,BHK)
);
```

6.CREATING TABLE HOUSE

```sql
CREATE TABLE IF NOT EXISTS House(
    House_ID int auto_increment,
    House_Addr varchar(50),
    primary key(House_ID,House_Addr),
    Htype int(2),
    constraint fk_house_type foreign key(Htype) REFERENCES 
House_type(Type_No),
    Rent_Cost int(10) not null,
    Owner_ID int(6),
    constraint fk_owner_id foreign key(Owner_ID) REFERENCES Owner(Owner_ID)

);
```

7.CREATING TABLE CONTRACT

```sql
CREATE TABLE IF NOT EXISTS Contract(
    Contract_ID int(6) auto_increment,
    primary key(Contract_ID),
    Start_Date date,
    Duration int(2) not null default 1,
    Rent_Cost_Duration int(10),
    -- constraint fk_rent_cost foreign key(Rent_Cost_Duration) references 
House(Rent_Cost),
    Owner_ID int(6),
    constraint fk_owner_id foreign key(Owner_ID) REFERENCES Owner(Owner_ID),
    Tenant_ID int(6),
    constraint fk_tenant_id2 foreign key(Tenant_ID) REFERENCES 
Tenant(Tenant_ID),
    House_ID int(6),
    constraint fk_house_id foreign key(House_ID) REFERENCES House(House_ID)
);
```

8.CREATING TABLE PAYMENT

```
CREATE TABLE IF NOT EXISTS Payment(
    Payment_ID int auto_increment,
    primary key(Payment_ID),
    Pay_From varchar(20) not null,
    Payment_Date date,
    Amount int(10) not null,
    Tenant_ID int(6),
    constraint fk_tenant_id3 foreign key(Tenant_ID) REFERENCES
Tenant(Tenant_ID),
    Contract_ID int(6),
    constraint fk_contract_id foreign key(Contract_ID) REFERENCES
Contract(Contract_ID)
);
```

9.ALTERING TABLE OWNER

```
alter table Owner auto_increment=100000;
```

10.ALTERING TABLE OWNER_PHONE_NO

```
alter table Owner_Ph add constraint check_ph1 check(char_length(Ph_No)=10);
```

11.ALTERING TABLE TENANT

```
alter table Tenant auto_increment=300000;
```

12.ALTERING TABLE TENANT_PHONE_NO

```
alter table Tenant_Ph add constraint check_ph2 check(char_length(Ph_No)=10);
```

13.ALTERING TABLE HOUSE

```
alter table House auto_increment=200000;
```

14.ALTERING TABLE CONTRACT

```
alter table Contract auto_increment=400000;
```

15.ALTERING TABLE PAYMENT

```
alter table Payment auto_increment=500000;
```

Populating the Database

1.INSERTING VALUES INTO OWNER

```
insert INTO Owner(F_Name,L_Name,username,password,Owner_Addr,DOB)
VALUES("Vrushank","G","vrush41","12345","2nd cross,millerpet,bellary",'2002-
08-12');
insert INTO Owner(F_Name,L_Name,username,password,Owner_Addr,DOB)
VALUES("Hemanth","N","hemanth28","23456","4th
cross,gauribidanur,chikkaballapur",'2002-01-28');
insert INTO Owner(F_Name,L_Name,username,password,Owner_Addr,DOB)
VALUES("Dhanush","M D","mdebro","34567","2nd cross,puttur,mangaluru",'2002-06-
01');
insert INTO Owner(F_Name,L_Name,username,password,Owner_Addr,DOB)
VALUES("Srinivas","Y","vasu03","45678","3rd cross,rr nagar,bengaluru",'2002-
03-27');
insert INTO Owner(F_Name,L_Name,username,password,Owner_Addr,DOB)
VALUES("Om","Prasad","om123","56789","2nd cross,jaynagar,bengaluru",'2002-05-
10');
```

2.INSERTING VALUES INTO OWNER_PHONE_NO

```
insert INTO Owner_Ph(Owner_ID,Ph_No) VALUES(100000,(9876929479));
insert INTO Owner_Ph(Owner_ID,Ph_No) VALUES(100001,(6968456239));
insert INTO Owner_Ph(Owner_ID,Ph_No) VALUES(100002,(9087565642));
insert INTO Owner_Ph(Owner_ID,Ph_No) VALUES(100003,(9980674554));
insert INTO Owner_Ph(Owner_ID,Ph_No) VALUES(100004,(9780678543));
```

3.INSERTING VALUES INTO TENANT

```
insert INTO Tenant(F_Name,L_Name,Username,Password,Tenant_Addr,DOB)
VALUES("Sathvik","A","stvk64","09875","guntur",'2002-04-03');
```

```
insert INTO Tenant(F_Name,L_Name,Username,Password,Tenant_Addr,DOB)
VALUES("Teja","Kanala","tsreddy43","98735","kurnool",'2002-07-21');
insert INTO Tenant(F_Name,L_Name,Username,Password,Tenant_Addr,DOB)
VALUES("Soumith","B","soumpi23","56780","bellary",'2002-01-23');
insert INTO Tenant(F_Name,L_Name,Username,Password,Tenant_Addr,DOB)
VALUES("Prathap","P","ptp45","12988","hindupur",'2002-07-09');
insert INTO Tenant(F_Name,L_Name,Username,Password,Tenant_Addr,DOB)
VALUES("Nayan","K","nyn987","09876","rajaji nagar,bengaluru",'2002-04-03');
```

4.INSERTING VALUES INTO TENANT_PHONE_NO

```
insert INTO Tenant_Ph(Tenant_ID,Ph_No) VALUES(300000,(9067825372));
insert INTO Tenant_Ph(Tenant_ID,Ph_No) VALUES(300001,(6273682936));
insert INTO Tenant_Ph(Tenant_ID,Ph_No) VALUES(300002,(9808577578));
insert INTO Tenant_Ph(Tenant_ID,Ph_No) VALUES(300003,(9768457902));
insert INTO Tenant_Ph(Tenant_ID,Ph_No) VALUES(300020,(6363787893));
```

5.INSERTING VALUES INTO HOUSE_TYPE

```
insert INTO House_Type(Type_No,BHK) VALUES(1,"1 BHK");
insert INTO House_Type(Type_No,BHK) VALUES(2,"2 BHK");
insert INTO House_Type(Type_No,BHK) VALUES(3,"3 BHK");
insert INTO House_Type(Type_No,BHK) VALUES(4,"4 BHK");
insert INTO House_Type(Type_No,BHK) VALUES(5,"5 BHK");
```

6.INSERTING VALUES INTO HOUSE

```
INSERT INTO House(House_Addr,Htype,Rent_Cost,Owner_ID)
VALUES("Bellary",3,10000,100000);
INSERT INTO House(House_Addr,Htype,Rent_Cost,Owner_ID)
VALUES("gauribidanur",2,8000,100001);
INSERT INTO House(House_Addr,Htype,Rent_Cost,Owner_ID)
VALUES("puttur",1,7000,100002);
INSERT INTO House(House_Addr,Htype,Rent_Cost,Owner_ID) VALUES("rr
nagar",5,15000,100003);
INSERT INTO House(House_Addr,Htype,Rent_Cost,Owner_ID)
VALUES("jaynagar",4,20000,100004);
```

Tool Used

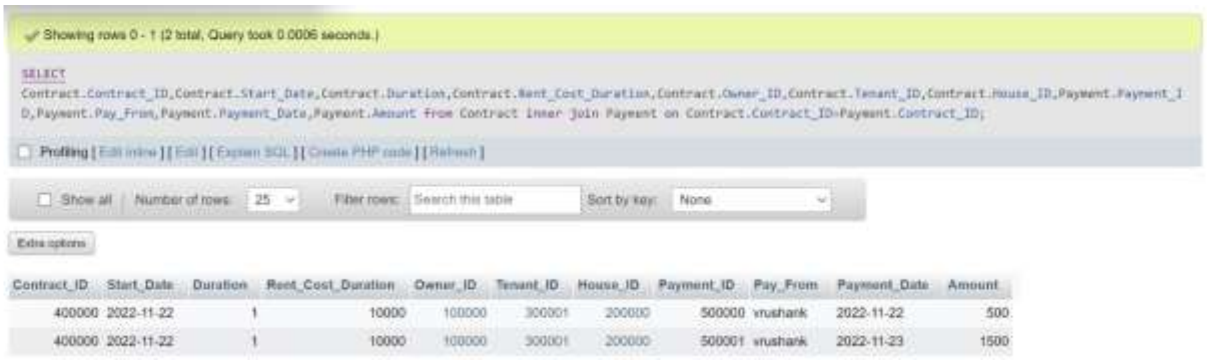
BACKEND: MariaDB

FRONTEND: STREAMLIT

Queries

Join queries

1.JOINING TABLE CONTRACT WITH PAYMENT TO SHOW THE PAYMENT DETAILS OF RESPECTIVE CONTRACT_ID

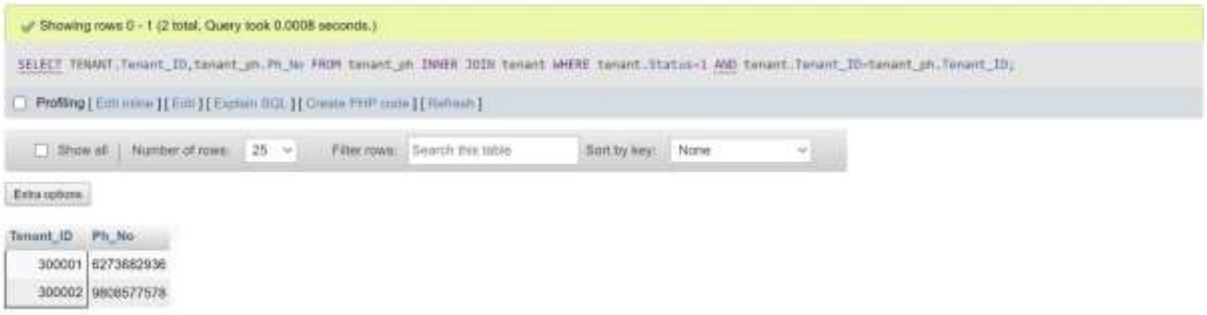


Showing rows 0 - 1 (2 total, Query took 0.0006 seconds.)

```
SELECT
Contract.Contract_ID,Contract.Start_Date,Contract.Duration,Contract.Rent_Cost_Duration,Contract.Owner_ID,Contract.Tenant_ID,Contract.House_ID,Payment.Payment_I
D,Payment.Pay_From,Payment.Payment_Date,Payment.Amount from Contract inner join Payment on Contract.Contract_ID=Payment.Contract_ID;
```

Contract_ID	Start_Date	Duration	Rent_Cost_Duration	Owner_ID	Tenant_ID	House_ID	Payment_ID	Pay_From	Payment_Date	Amount
400000	2022-11-22	1	10000	100000	300001	200000	500000	vrushank	2022-11-22	500
400000	2022-11-22	1	10000	100000	300001	200000	500001	vrushank	2022-11-23	1500

2.FINDING TENANTS WITH THEIR CONTACTS WHO ARE UNDER CONTRACT

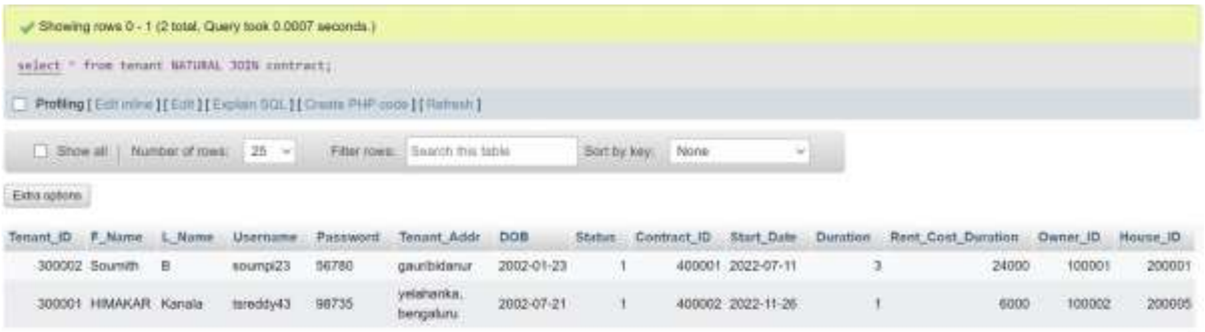


Showing rows 0 - 1 (2 total, Query took 0.0008 seconds.)

```
SELECT TENANT.Tenant_ID,tenant_ph.Ph_No FROM tenant_ph INNER JOIN tenant WHERE tenant.Status=1 AND tenant.Tenant_ID=tenant_ph.Tenant_ID;
```

Tenant_ID	Ph_No
300001	6273682936
300002	9808577578

3.NATURAL JOINING TENANT AND CONTRACT



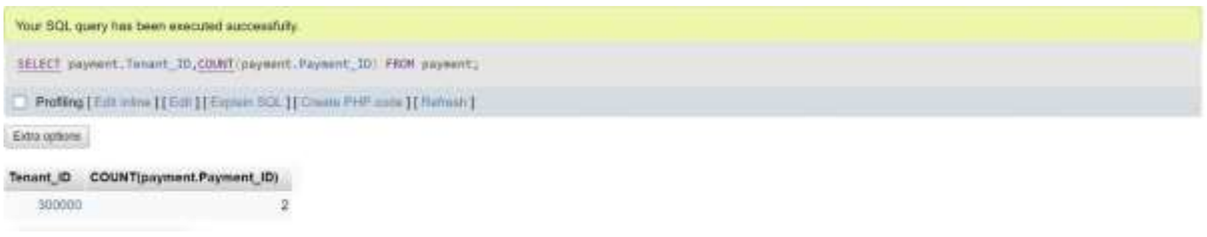
Showing rows 0 - 1 (2 total, Query took 0.0007 seconds.)

```
select * from tenant NATURAL JOIN contract;
```

Tenant_ID	F_Name	L_Name	Username	Password	Tenant_Addr	DOB	Status	Contract_ID	Start_Date	Duration	Rent_Cost_Duration	Owner_ID	House_ID
300002	Soumith	B	soumpi23	56780	gauribidanur	2002-01-23	1	400001	2022-07-11	3	24000	100001	200001
300001	HIMAKAR	Kanala	tsreddy43	98735	yelahanka, bengaluru	2002-07-21	1	400002	2022-11-26	1	6000	100002	200005

Aggregate Functions

1.COUNTING THE NUMBER OF PAYMENTS DONE BY THE TENANT

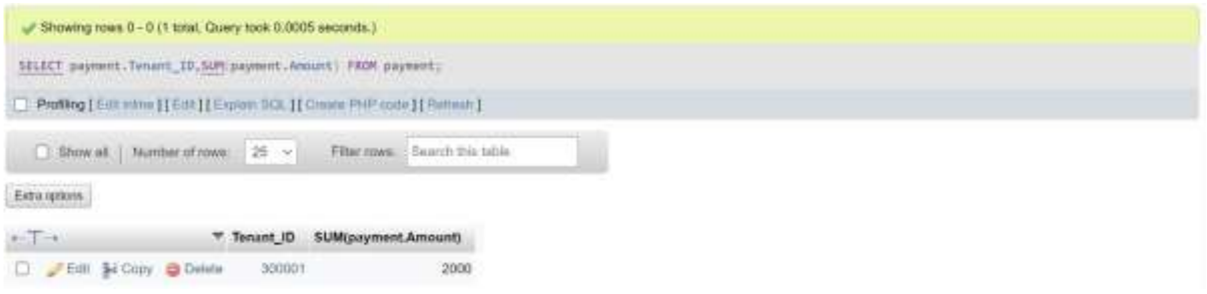

Your SQL query has been executed successfully.

```
SELECT payment.Tenant_ID,COUNT(payment.Payment_ID) FROM payment;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Extra options

Tenant_ID	COUNT(payment.Payment_ID)
300000	2

2.CALCULATING THE TOTAL AMOUNT PAYED BY THE TENANT


Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

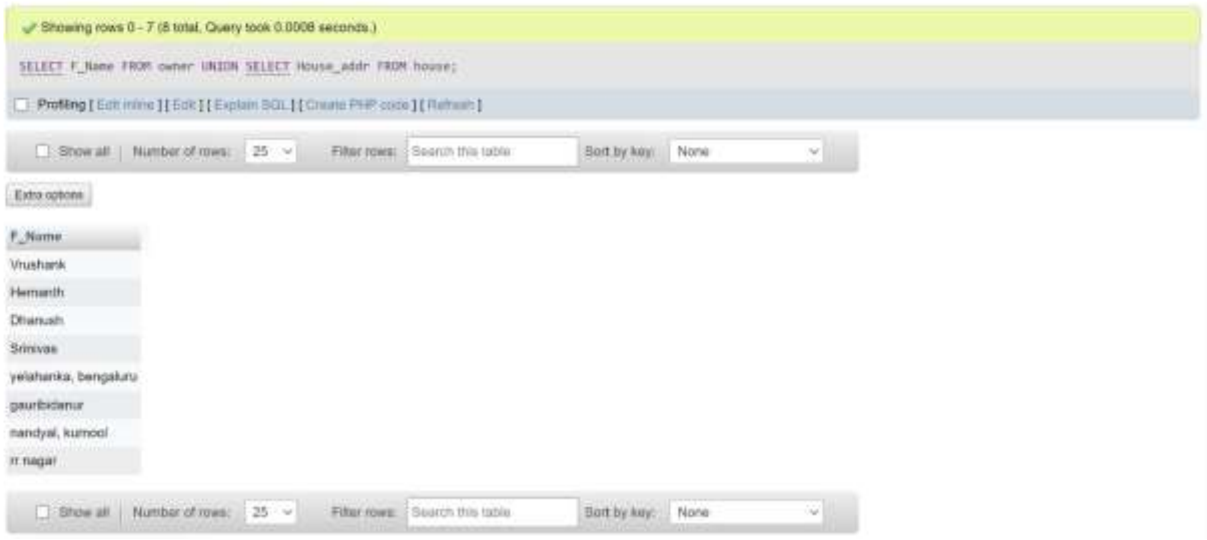
```
SELECT payment.Tenant_ID,SUM(payment.Amount) FROM payment;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Extra options
Edit
Copy
Delete

Tenant_ID	SUM(payment.Amount)
300001	2000

Set Operations

1.DISPLAYING HOUSE ADDRESS WITH THE NAMES OF THE OWNER


Showing rows 0 - 7 (8 total, Query took 0.0008 seconds.)

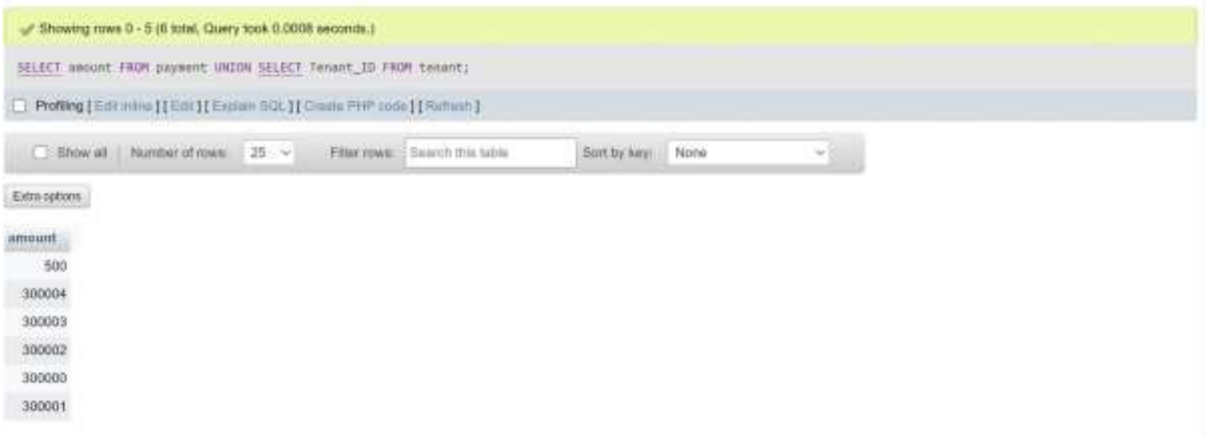
```
SELECT F_Name FROM owner UNION SELECT House_addr FROM house;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
Extra options

F_Name
Vrushank
Hemanth
Dhanush
Srinivas
yelahanka, bengaluru
gauribidanur
nandyal, kurnool
rr nagar

Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None

2.DISPLAYING THE AMOUNT PAID BY THE TENANT

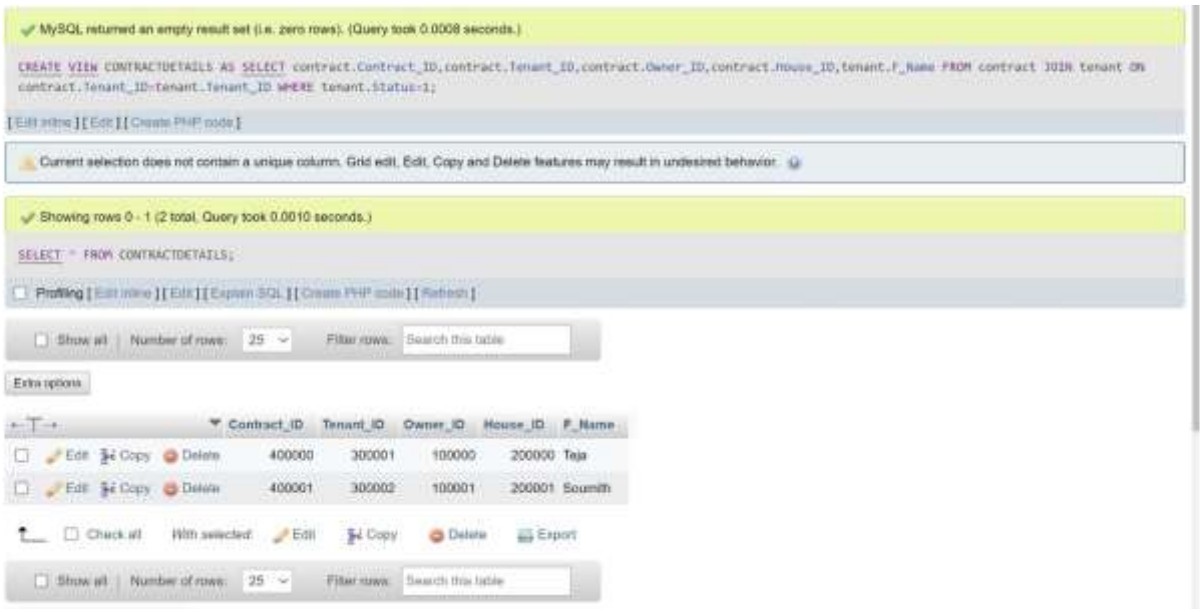


```
Showing rows 0 - 5 (6 total, Query took 0.0008 seconds.)

SELECT amount FROM payment UNION SELECT Tenant_ID FROM tenant;
```

amount
500
300004
300003
300002
300000
300001

View

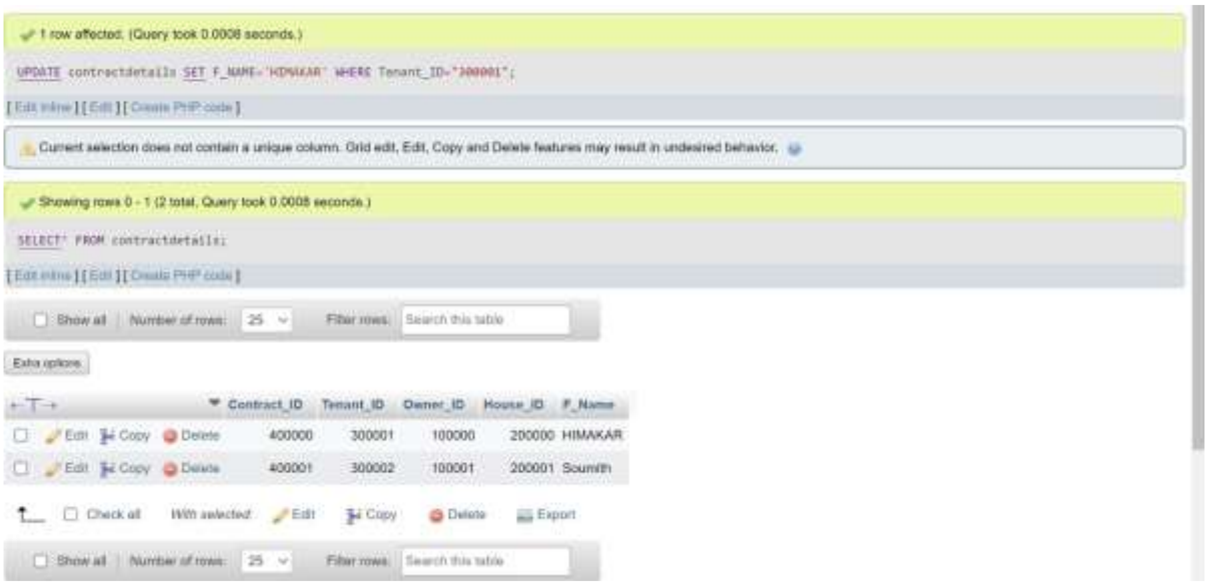
1.CREATING VIEW TO SHOW THE DETAILS OF TENANT WHO ARE UNDER THE CONTRACT



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0008 seconds.)

CREATE VIEW CONTRACTDETAILS AS SELECT contract.Contract_ID,contract.Tenant_ID,contract.Owner_ID,contract.House_ID,tenant.F_Name FROM contract JOIN tenant ON
contract.Tenant_ID=tenant.Tenant_ID WHERE tenant.Status=1;

Current selection does not contain a unique column. Grid edit, Edit, Copy and Delete features may result in undesired behavior.

Showing rows 0 - 1 (2 total, Query took 0.0010 seconds.)

SELECT * FROM CONTRACTDETAILS;
```

Contract_ID	Tenant_ID	Owner_ID	House_ID	F_Name
400000	300001	100000	200000	Teja
400001	300002	100001	200001	Soumith

2.UPDATING THE NAME OF THE TENANT (TEJA TO HIMAKAR)



```
✓ 1 row affected. (Query took 0.0008 seconds.)

UPDATE contractdetails SET F_NAME='HIMAKAR' WHERE Tenant_ID="300001";

[ Edit inline ] [ Edit ] [ Create PHP code ]

Current selection does not contain a unique column. Grid edit, Edit, Copy and Delete features may result in undesired behavior.

✓ Showing rows 0 - 1 (2 total, Query took 0.0008 seconds.)

SELECT * FROM contractdetails;

[ Edit inline ] [ Edit ] [ Create PHP code ]
```

	Contract_ID	Tenant_ID	Owner_ID	House_ID	F_Name
Edit Copy Delete	400000	300001	100000	200000	HIMAKAR
Edit Copy Delete	400001	300002	100001	200001	Soumith

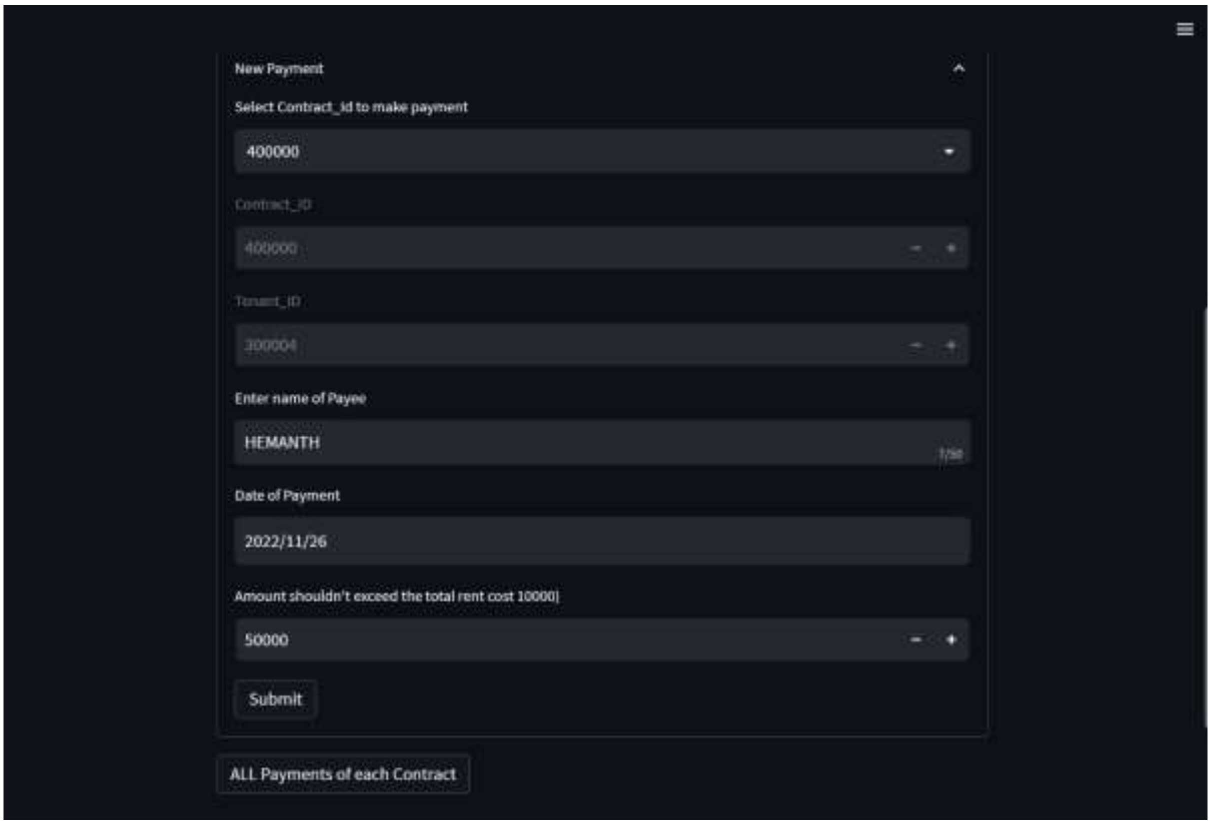
Show all
Number of rows: 25
Filter rows: Search this table
Extra options
Check all
With selected: Edit Copy Delete Export

Triggers (Functions or Procedures)

Create a Function or a Procedure. State the objective of the function / Procedure. Run and display the results.

1.TRIGGER TO SHOW THE ERROR WHEN THE TENANT PAYS THE AMOUNT MORE THAN THE TOTAL SUM TO BE PAYED.

```
CREATE TRIGGER amount_exceed AFTER INSERT ON  Payment FOR EACH ROW BEGIN IF
(select sum(Amount)+new.Amount from Payment group by Contract_ID={}) > {} THEN
SIGNAL SQLSTATE '45000' SET message_text="Amount can't exceed total Rent
Cost";END IF;END""".format(Contract_ID,totalrent))
```



Select Contract_id to make payment

400000

Contract_ID

400000

Tenant_ID

300004

Enter name of Payee

HEMANTH

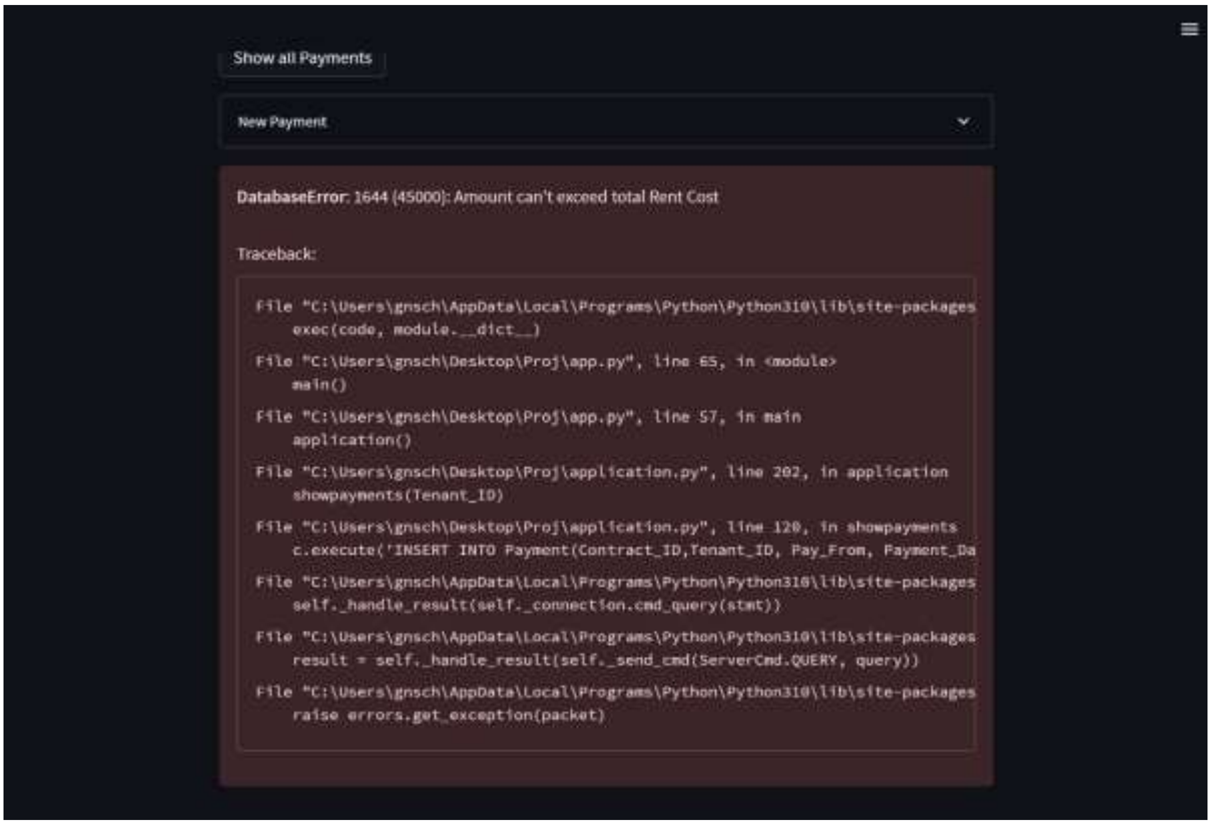
7/50

Date of Payment

2022/11/26

Amount shouldn't exceed the total rent cost 10000|

50000



DatabaseError: 1644 (45000): Amount can't exceed total Rent Cost

Traceback:

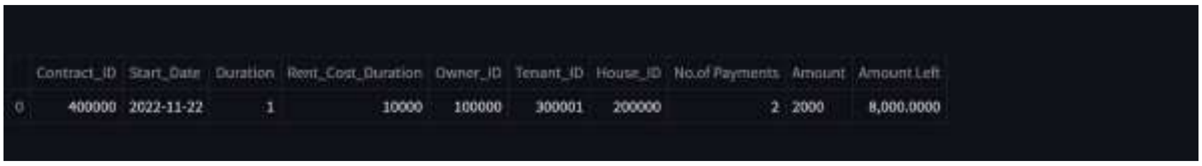
```
File "C:\Users\gnsch\AppData\Local\Programs\Python\Python310\lib\site-packages
    exec(code, module.__dict__)
File "C:\Users\gnsch\Desktop\Proj\app.py", line 65, in <module>
    main()
File "C:\Users\gnsch\Desktop\Proj\app.py", line 57, in main
    application()
File "C:\Users\gnsch\Desktop\Proj\application.py", line 202, in application
    showpayments(Tenant_ID)
File "C:\Users\gnsch\Desktop\Proj\application.py", line 120, in showpayments
    c.execute('INSERT INTO Payment(Contract_ID,Tenant_ID, Pay_From, Payment_Da
File "C:\Users\gnsch\AppData\Local\Programs\Python\Python310\lib\site-packages
    self._handle_result(self._connection.cmd_query(stmt))
File "C:\Users\gnsch\AppData\Local\Programs\Python\Python310\lib\site-packages
    result = self._handle_result(self._send_cmd(ServerCmd.QUERY, query))
File "C:\Users\gnsch\AppData\Local\Programs\Python\Python310\lib\site-packages
    raise errors.get_exception(packet)
```

2.STORED PROCEDURE TO CALCULATE THE END DATE OF THE CONTRACT

```
delimiter $

create Procedure end_date_calc(IN start_date date,IN duration int, OUT
end_date date)
begin
    select date_add(`start_date`, INTERVAL `duration` MONTH) into end_date;
end $

delimiter ;
```

3.FUNCTION TO CALCULATE THE REMAINING AMOUNT TO BE PAID BY THE TENANT

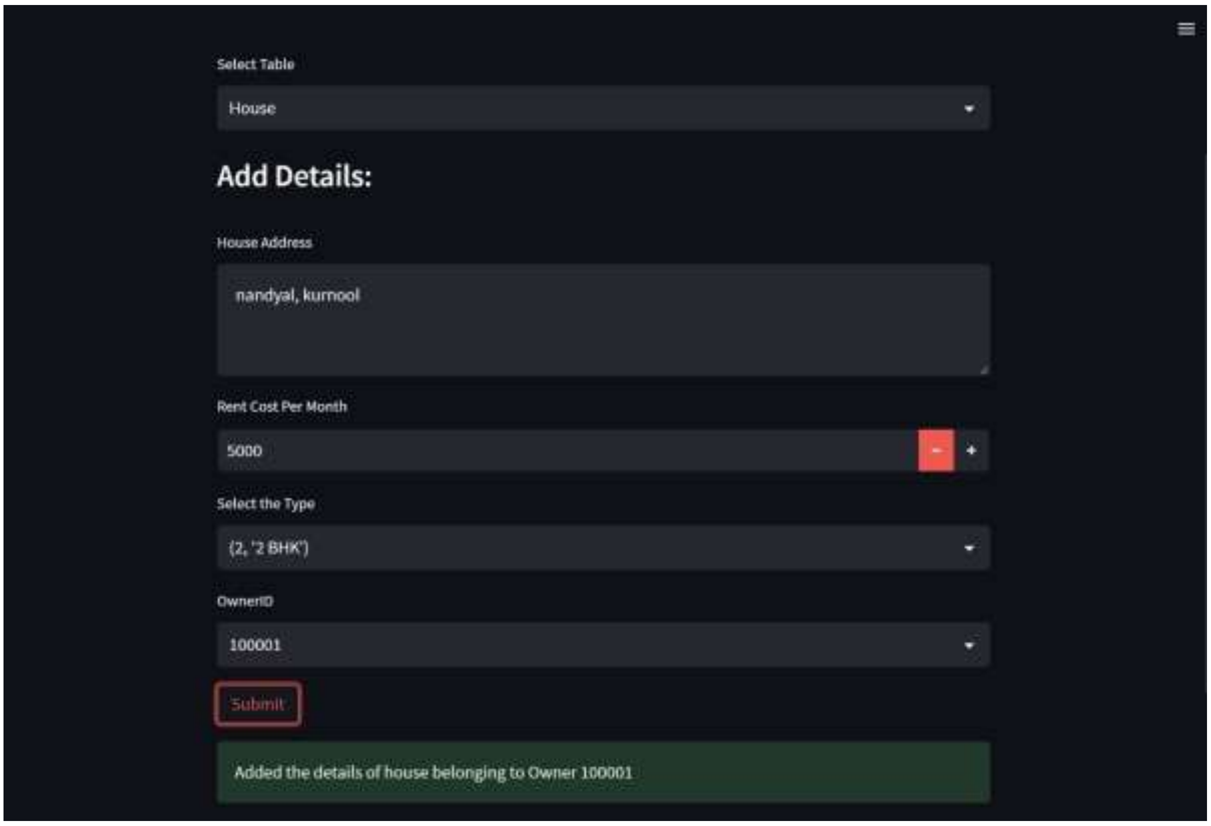
```
CREATE FUNCTION rent_remain(Amount double,Total double)
  RETURNS double
   DETERMINISTIC
    BEGIN
     DECLARE remain double;
        return Total-Amount;
    END $
delimiter ;
```



	Contract_ID	Start_Date	Duration	Rent_Cost_Duration	Owner_ID	Tenant_ID	House_ID	No.of Payments	Amount	Amount Left
0	400000	2022-11-22	1	10000	100000	300001	200000	2	2000	8,000.0000

Developing a Frontend

1. Addition, Modification and Deletion of records from any chosen table


Select Table

House

Add Details:

House Address

nandyal, kurnool

Rent Cost Per Month

5000

Select the Type

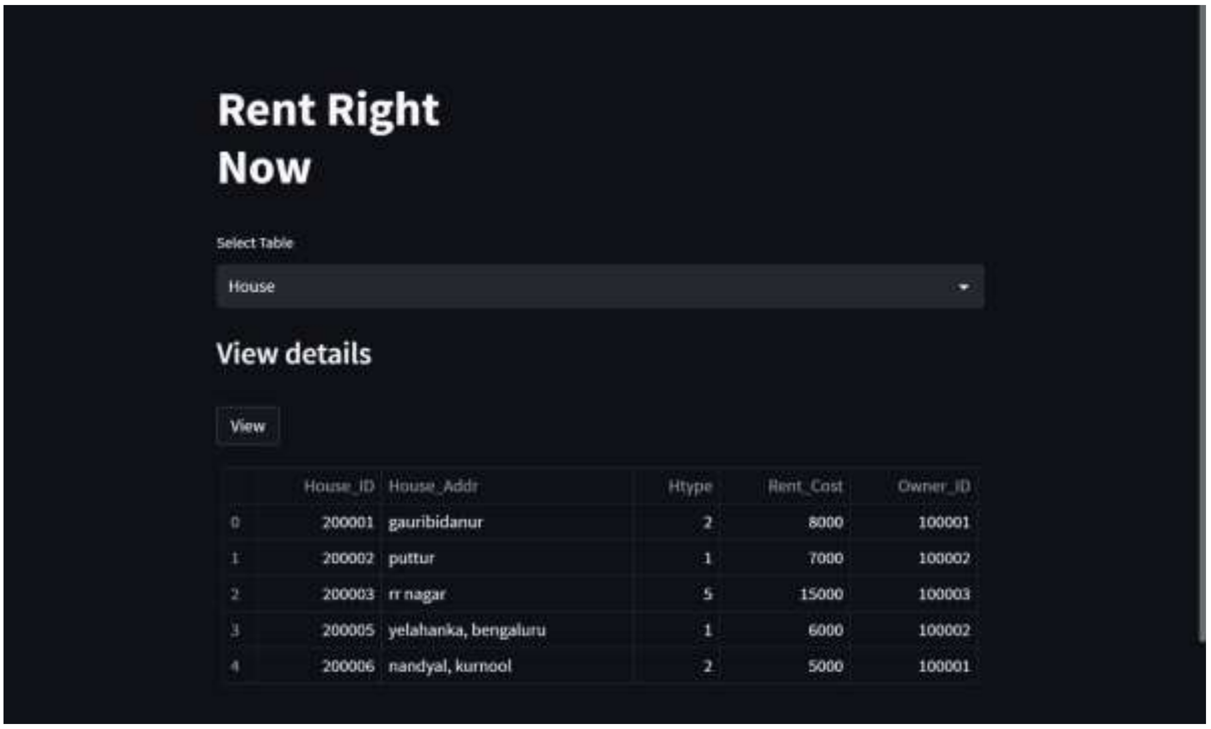
(2, '2 BHK')

OwnerID

100001

Submit

Added the details of house belonging to Owner 100001


Rent Right Now

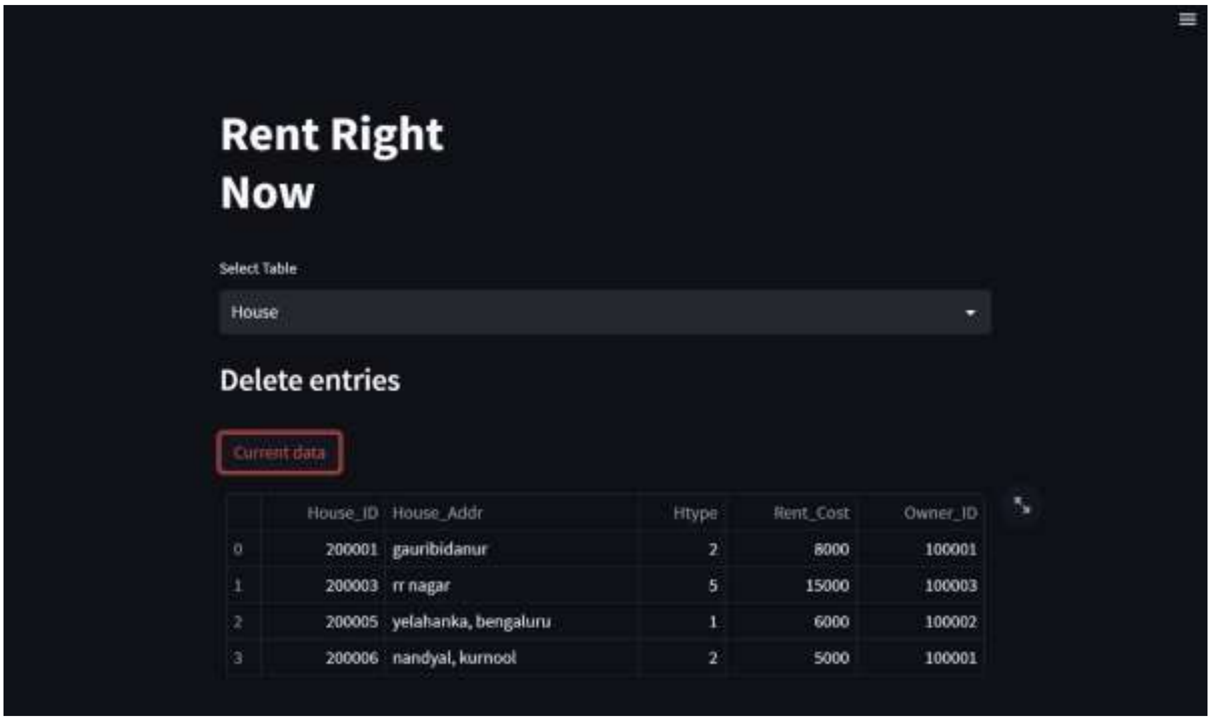
Select Table

House

View details

View

	House_ID	House_Addr	Htype	Rent_Cost	Owner_ID
0	200001	gauribidanur	2	8000	100001
1	200002	puttur	1	7000	100002
2	200003	rr nagar	5	15000	100003
3	200005	yelahanka, bengaluru	1	6000	100002
4	200006	nandyal, kurnool	2	5000	100001



Rent Right Now

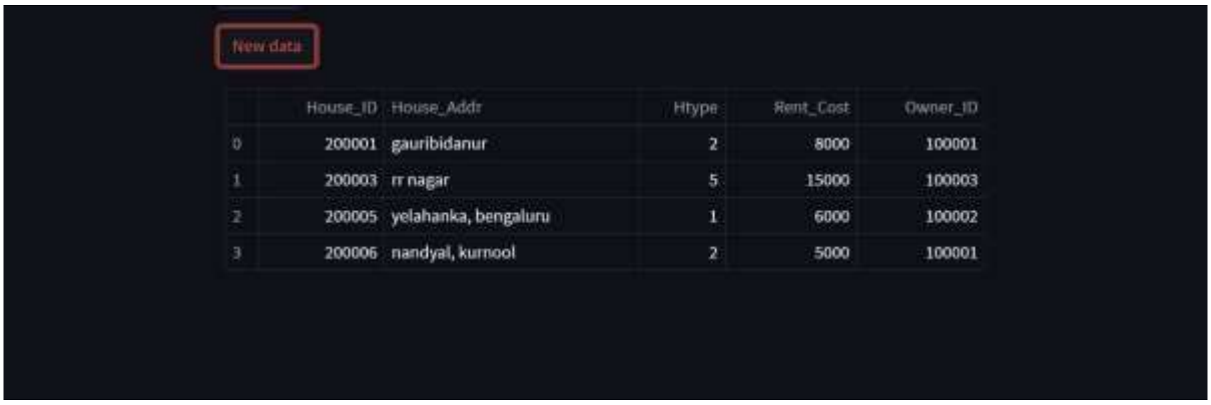
Select Table

House

Delete entries

Current data

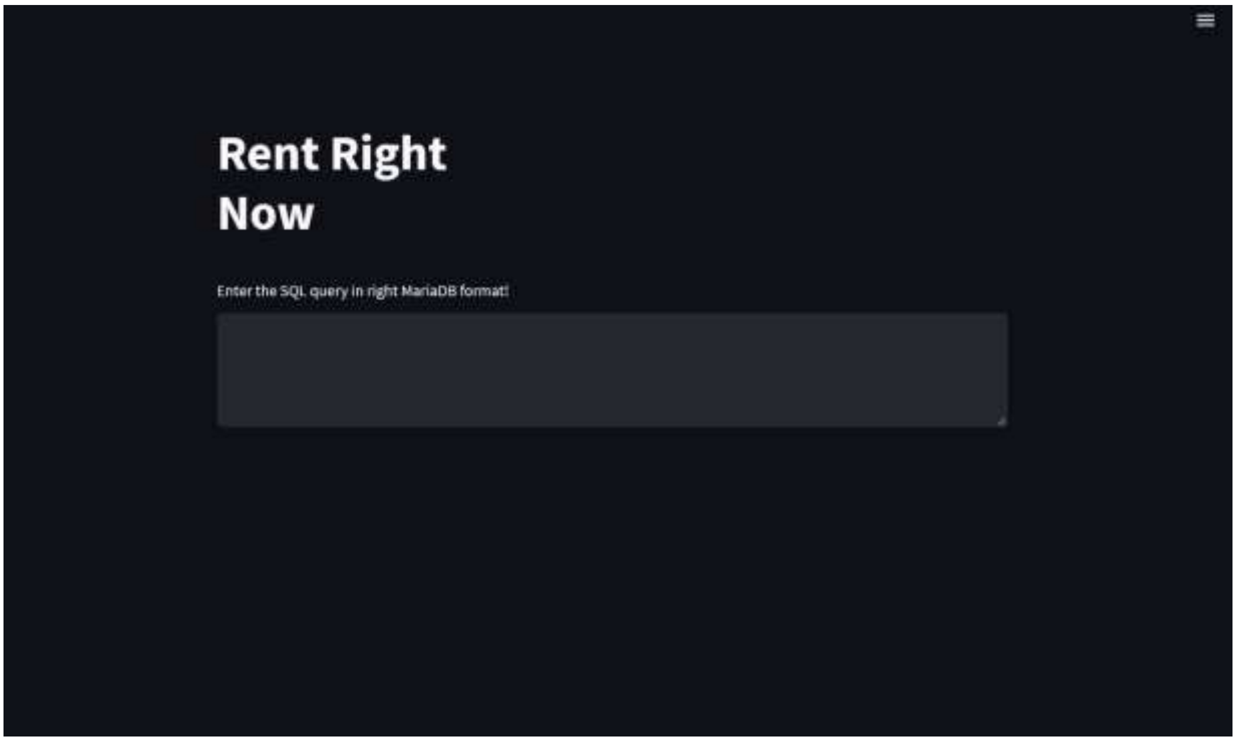
	House_ID	House_Addr	Htype	Rent_Cost	Owner_ID
0	200001	gauribidanur	2	8000	100001
1	200003	rr nagar	5	15000	100003
2	200005	yelahanka, bengaluru	1	6000	100002
3	200006	nandyal, kurnool	2	5000	100001



New data

	House_ID	House_Addr	Htype	Rent_Cost	Owner_ID
0	200001	gauribidanur	2	8000	100001
1	200003	rr nagar	5	15000	100003
2	200005	yelahanka, bengaluru	1	6000	100002
3	200006	nandyal, kurnool	2	5000	100001

2. Window to accept and run any SQL statement and display the result



Rent Right Now

Enter the SQL query in right MariaDB format!