DBMS Project Report

Title : House Rental System

Team:

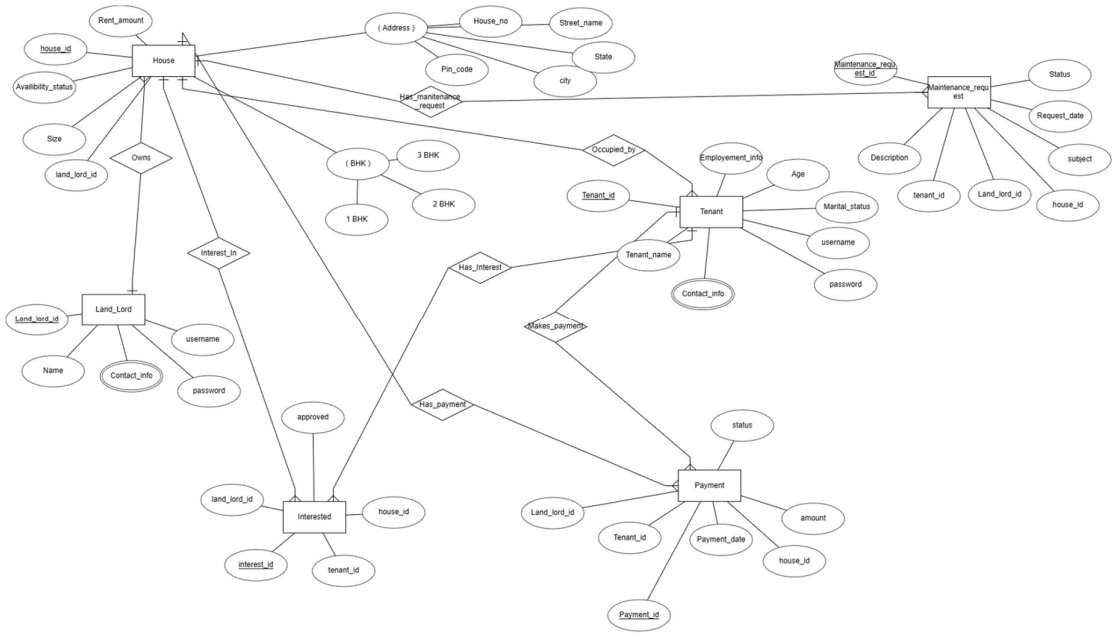
Pritam S Gurav (PES1UG21CS453)

Pratheek K N (PES1UG21CS443)

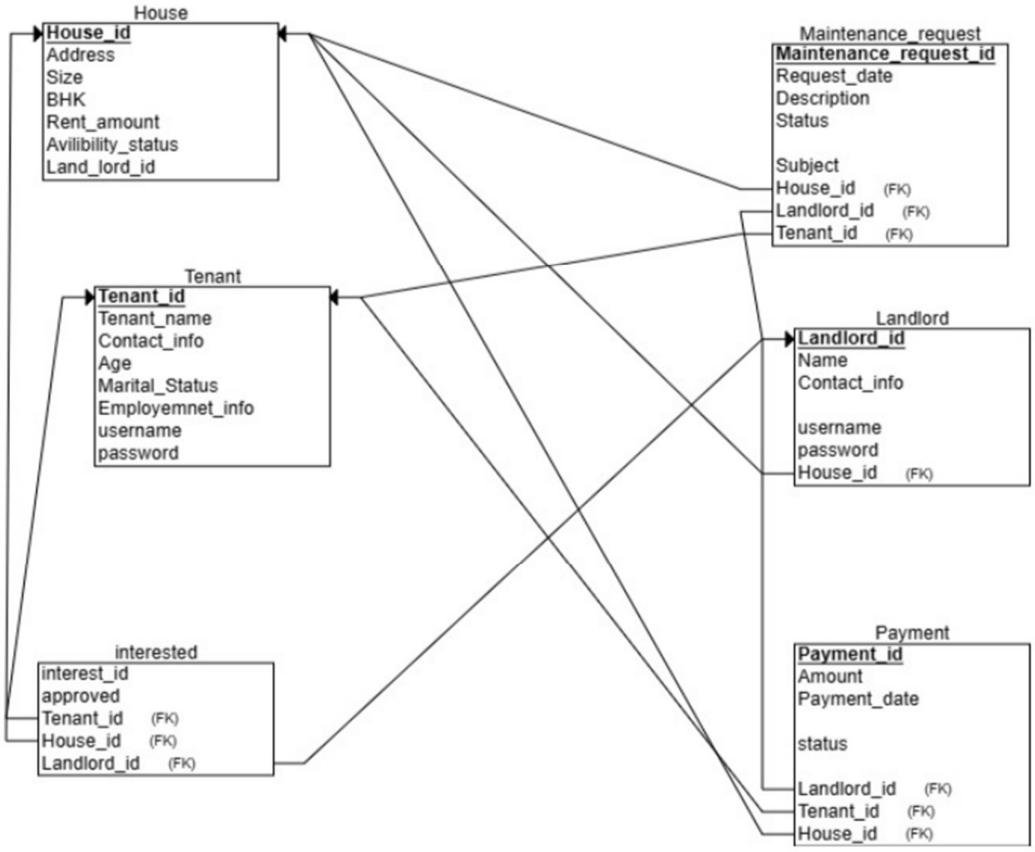
Abstract:

Our House rental system website, the tenant experience is enriched with user-friendly features that simplify the house hunting process. Tenants can easily search for available houses, streamlining their search with filters that match their preferences. The system facilitates seamless communication between tenants and landlords through maintenance request submissions, allowing tenants to address property-related concerns directly. Additionally, tenants can efficiently manage rent payments, receiving timely reminders and notifications for pending requests. On the landlord side, the platform empowers property owners with tools for effective property management. Landlords can effortlessly add and showcase rental properties, enhancing their visibility to potential tenants. The system provides a centralized hub for landlords to review maintenance requests, ensuring prompt and efficient property maintenance. The ability to generate payment requests and manage tenant requests for a specific property adds a layer of control and convenience for landlords, creating a holistic solution for property management. Together, these features create a balanced and comprehensive house rental ecosystem, catering to the distinct needs of both tenants and landlords

ER – Diagram



Relational Schema



Queries:

The queries that we have used in our Project are

Join query

DDL Sql Command:

1) Create Table

   Interested table

```
CREATE TABLE `interested` (
  `interest_id` int NOT NULL AUTO_INCREMENT,
  `tenant_id` int NOT NULL,
  `land_lord_id` int NOT NULL,
  `house_id` int NOT NULL,
  `approved` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`interest_id`),
  KEY `tenant_id` (`tenant_id`),
  KEY `land_lord_id` (`land_lord_id`),
  KEY `house_id` (`house_id`),
  CONSTRAINT `interested_ibfk_1` FOREIGN KEY (`tenant_id`) REFERENCES `tenant` (`tenant_id`),
  CONSTRAINT `interested_ibfk_2` FOREIGN KEY (`land_lord_id`) REFERENCES `land_lord` (`land_lord_id`),
  CONSTRAINT `interested_ibfk_3` FOREIGN KEY (`house_id`) REFERENCES `house` (`house_id`)
) ENGINE=InnoDB AUTO_INCREMENT=58 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

House

```
CREATE TABLE `house` (
  `house_id` int NOT NULL AUTO_INCREMENT,
  `size` int NOT NULL,
  `bhk` int NOT NULL,
  `rent_amount` int NOT NULL,
  `house_no` varchar(10) NOT NULL,
  `street_name` varchar(20) NOT NULL,
  `city` varchar(20) NOT NULL,
  `pincode` varchar(10) NOT NULL,
  `state` varchar(20) NOT NULL,
  `availibility_status` tinyint(1) NOT NULL,
  `land_lord_id` int NOT NULL,
  PRIMARY KEY (`house_id`),
  KEY `land_lord_id` (`land_lord_id`),
  CONSTRAINT `house_ibfk_1` FOREIGN KEY (`land_lord_id`) REFERENCES `land_lord` (`land_lord_id`)
```

Tenant

```
CREATE TABLE `tenant` (
  `tenant_id` int NOT NULL AUTO_INCREMENT,
  `username` varchar(50) NOT NULL,
  `password` varchar(255) NOT NULL,
  `age` int NOT NULL,
  `contact_info` varchar(20) NOT NULL,
  `employment_info` char(1) DEFAULT NULL,
  `tanant_name` varchar(255) DEFAULT NULL,
  `marital_status` char(1) DEFAULT NULL,
  `house_id` int DEFAULT NULL,
  PRIMARY KEY (`tenant_id`),
  KEY `house_id` (`house_id`),
  CONSTRAINT `tenant_ibfk_1` FOREIGN KEY (`house_id`) REFERENCES `house` (`house_id`)
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Land_lord

```
CREATE TABLE `land_lord` (
  `land_lord_id` int NOT NULL AUTO_INCREMENT,
  `username` varchar(50) NOT NULL,
  `password` varchar(50) NOT NULL,
  `name` varchar(50) NOT NULL,
  `contact_info` varchar(20) NOT NULL,
  PRIMARY KEY (`land_lord_id`)
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Maintenance_req table

```
CREATE TABLE `maintenance_req` (
  `main_req_id` int NOT NULL AUTO_INCREMENT,
  `house_id` int NOT NULL,
  `tenant_id` int NOT NULL,
  `request_date` varchar(10) NOT NULL,
  `subject` varchar(40) NOT NULL,
  `description` varchar(300) DEFAULT NULL,
  `land_lord_id` int NOT NULL,
  `status` tinyint(1) NOT NULL,
  PRIMARY KEY (`main_req_id`),
  KEY `house_id` (`house_id`),
  KEY `tenant_id` (`tenant_id`),
  KEY `land_lord_id` (`land_lord_id`),
  CONSTRAINT `maintenance_req_ibfk_1` FOREIGN KEY (`house_id`) REFERENCES `house` (`house_id`),
  CONSTRAINT `maintenance_req_ibfk_2` FOREIGN KEY (`tenant_id`) REFERENCES `tenant` (`tenant_id`),
  CONSTRAINT `maintenance_req_ibfk_3` FOREIGN KEY (`land_lord_id`) REFERENCES `land_lord` (`land_lord_id`)
```

Payment:

```
CREATE TABLE `payment` (
  `payment_id` int NOT NULL AUTO_INCREMENT,
  `amount` int NOT NULL,
  `status` tinyint(1) NOT NULL,
  `payment_date` varchar(10) NOT NULL,
  `tenant_id` int NOT NULL,
  `land_lord_id` int NOT NULL,
  `house_id` int NOT NULL,
  PRIMARY KEY (`payment_id`)
) ENGINE=InnoDB AUTO_INCREMENT=30 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
```

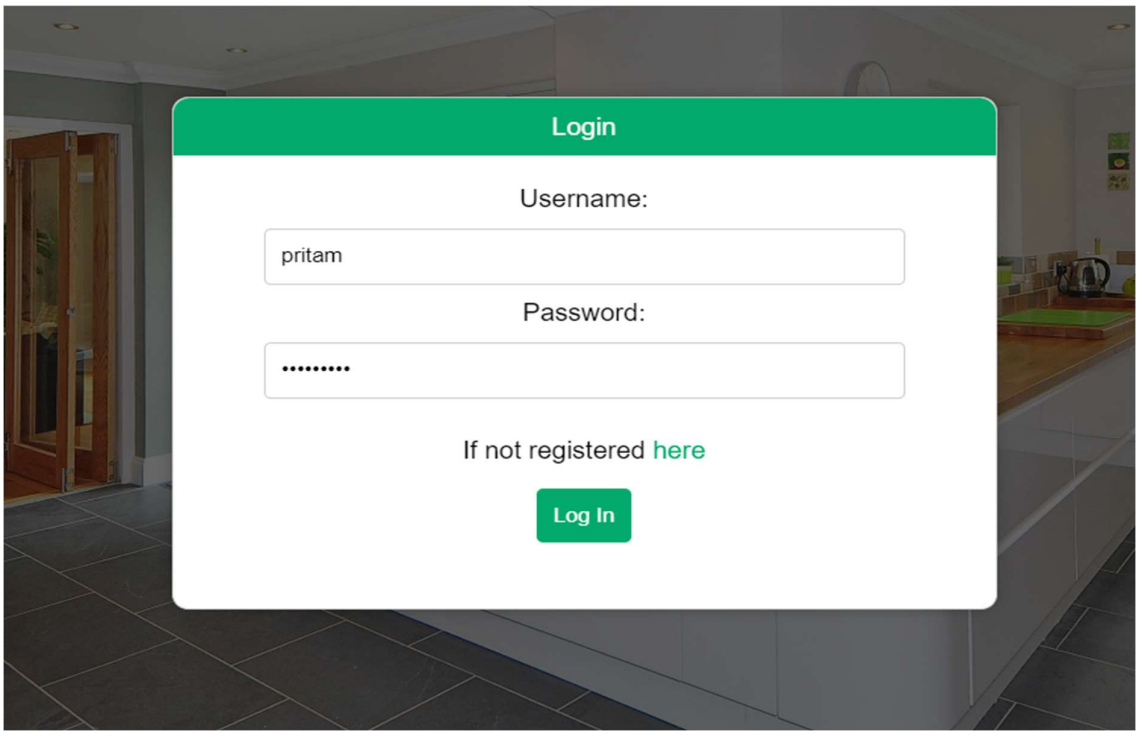
CRUD Operation

Select Command:

There are many select some of them are

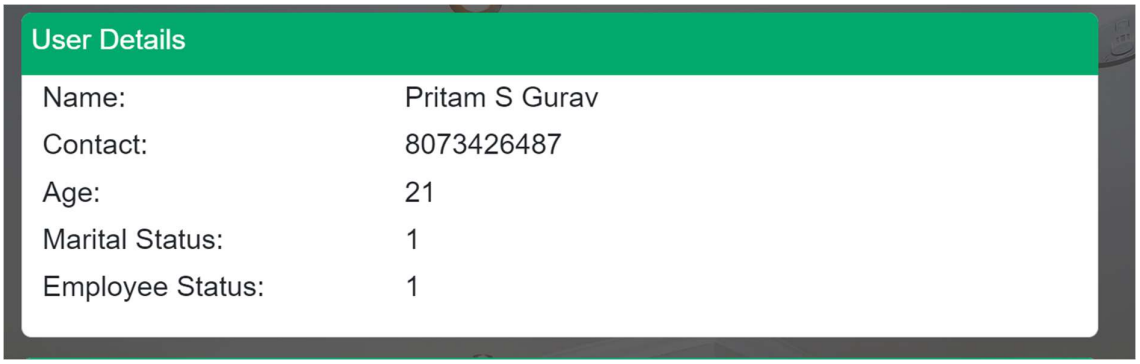
```
cur.execute('SELECT * FROM tenant WHERE username = %s AND password =
%s',(username,password))
```

This will check the password and username if the account is present then it will show the account is present or else it will login



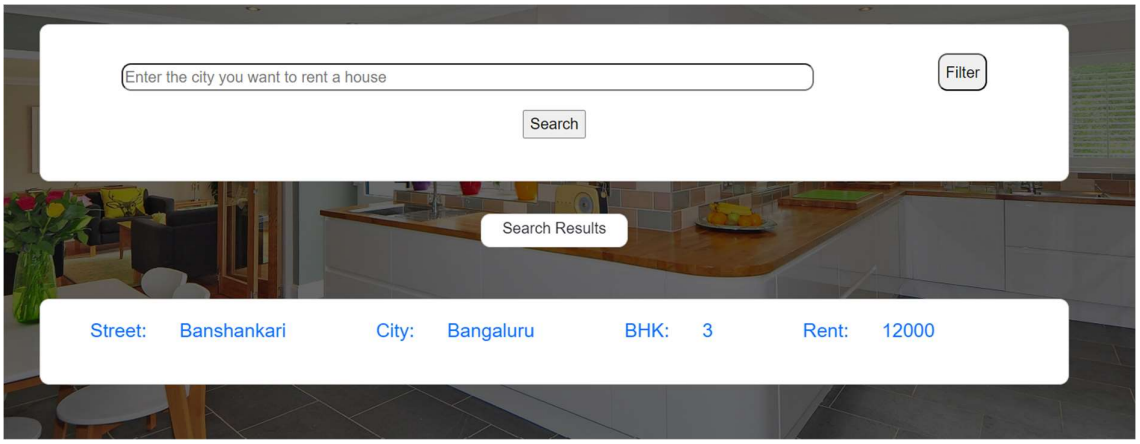
```
str = f"SELECT * FROM tenant WHERE username = '{(username)}'"
```

gets the user details



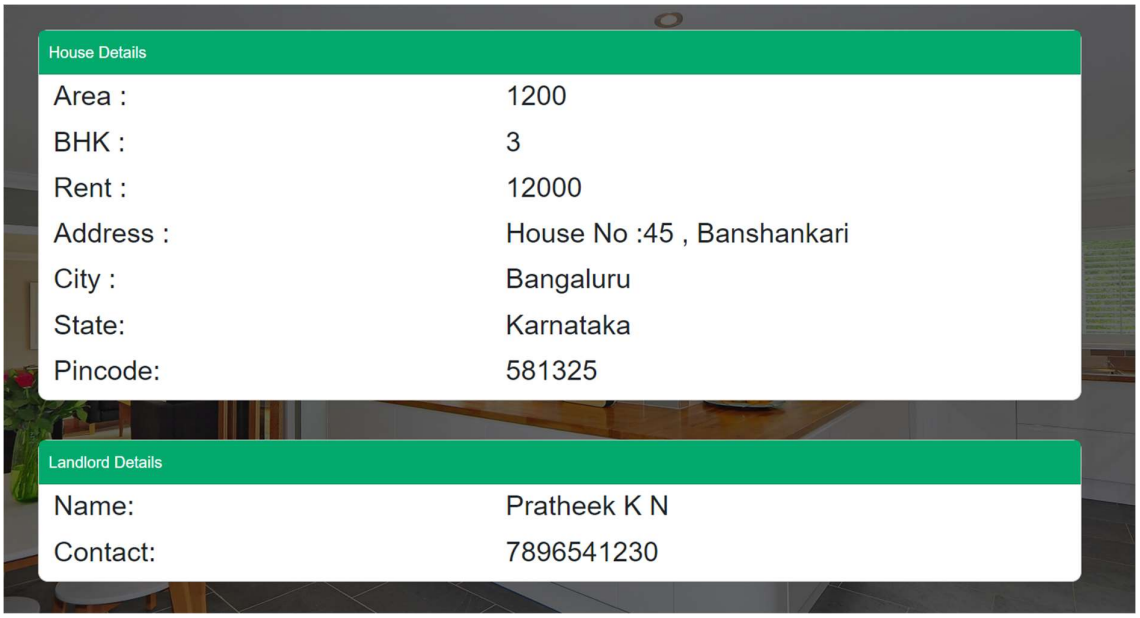
```
cur.execute(f"select * from house where city = '{search_city}' and bhk =
'{bhk}' and rent_amount >= '{1000}' and rent_amount <= '{price_range}' and
availibility_status = 1")
```

this is a search query that will search in the database according to the city , price and bhk



```
cur.execute(f"SELECT * from house where house_id = '{house_details_id}'")
account = cur.fetchone()
cur.execute(f"SELECT * from land_lord where land_lord_id = '{account[-1]}'")
landlord = cur.fetchone()
```

This query will get the house details and landlord details



These are some of this select queries

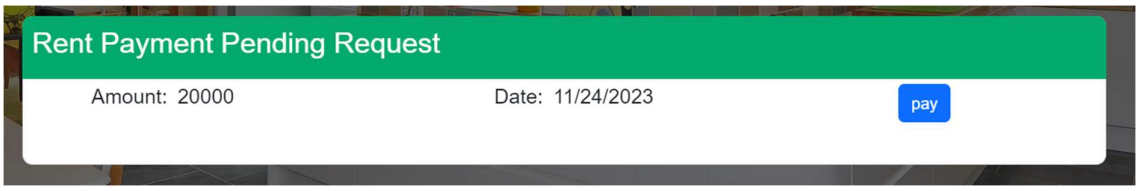
---

2) Update

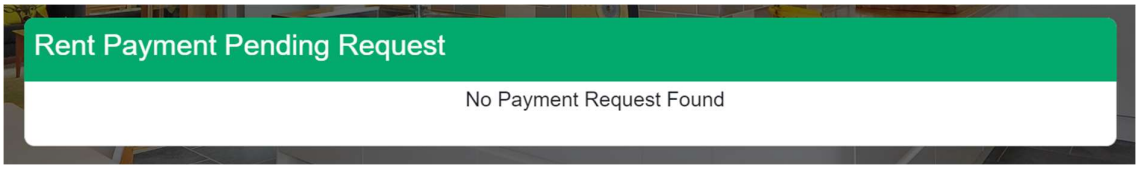
```
payment_id = request.form['payment']
cur.execute(f"UPDATE payment SET status = 0 where payment_id = '{payment_id}'")
db.commit()
```

If paid it will show none update command will set avalibility status to 0

before



After update

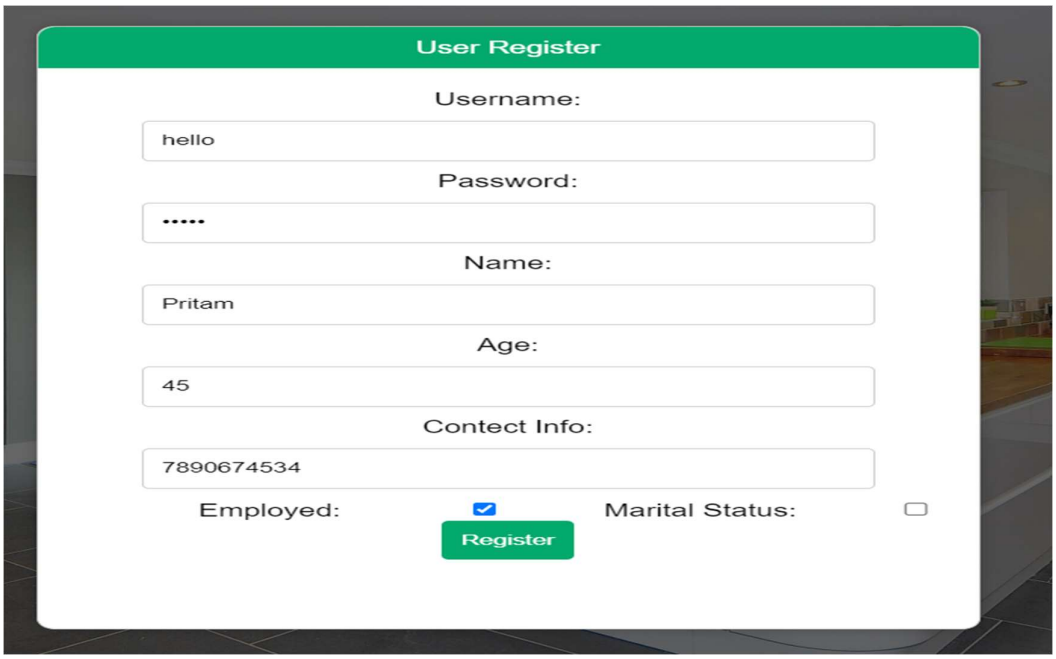


---

3) Insert

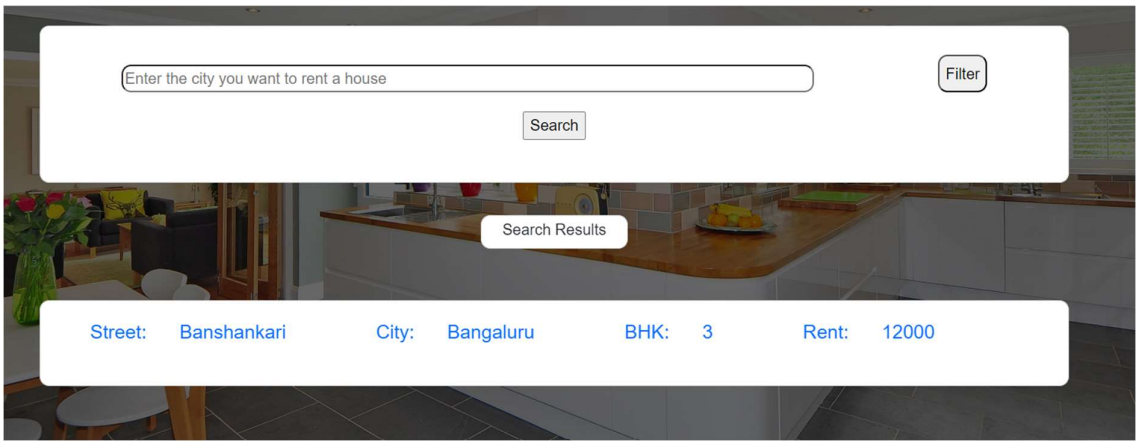
```
cur.execute("insert into land_lord (username,password,name,contact_info) values (%s,%s,%s,%s)",(ownername,password,name,contact))
db.commit()
```

This code is used for register the user register same for landlord



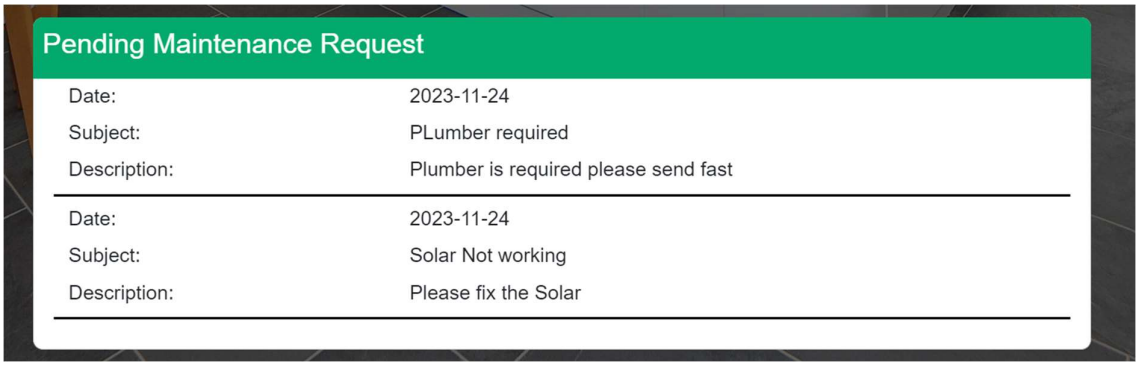
Functionality:

Search:

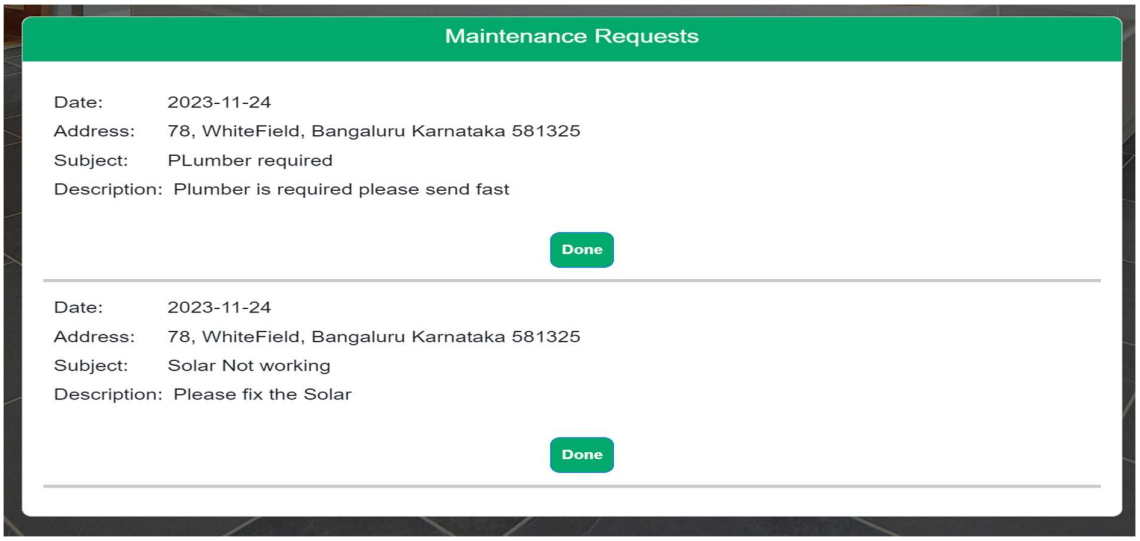


Maintenance_req

Pending maintenance req at user side



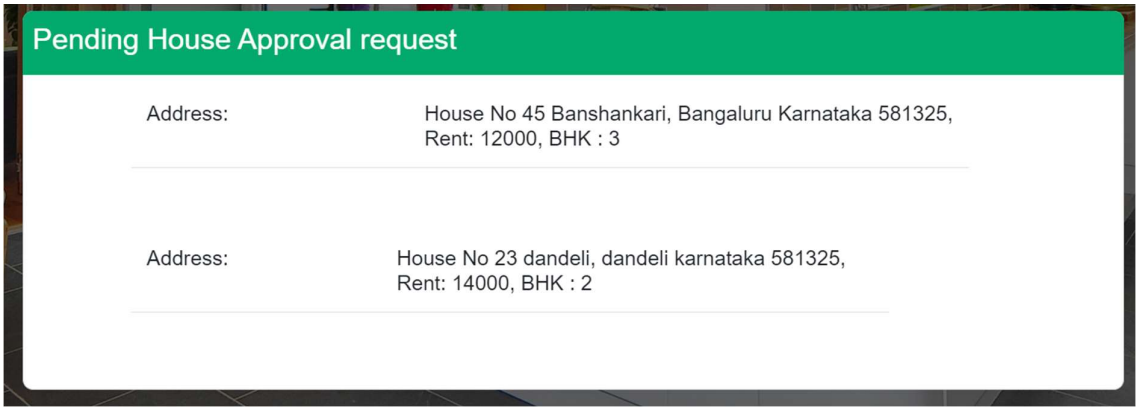
Pending maintenance req at land_lord side



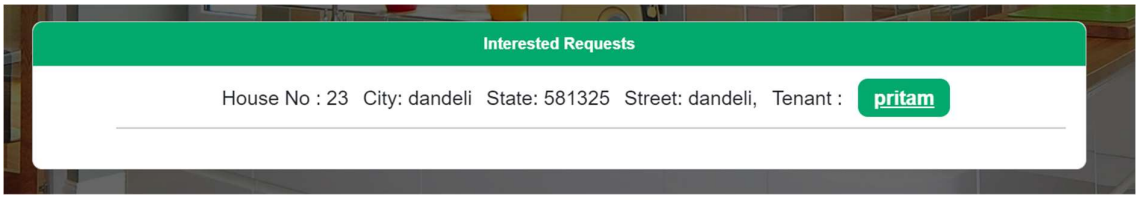
House Interested Request

If user request for a house the interested request will be shown on both side user and land_lord and land_lord has the right to accept and reject

At user side



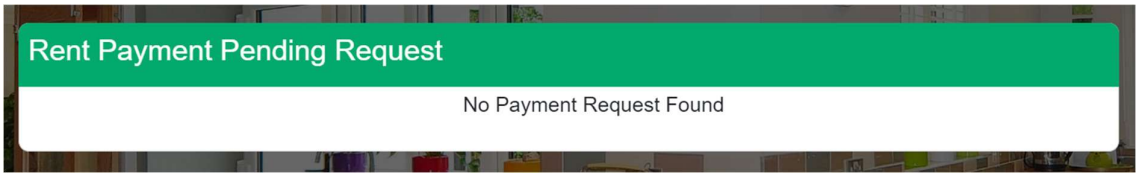
At land_lord side



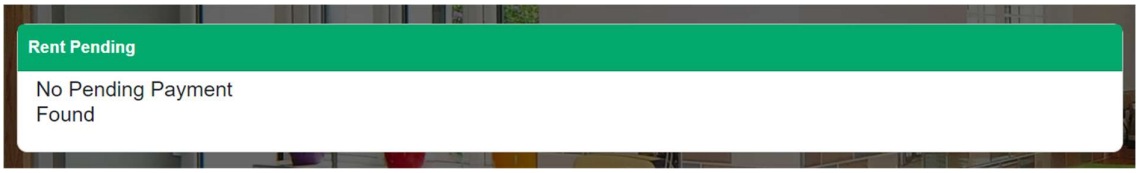
Rent Payment Pending Request:

All the pending payment request will be shown here

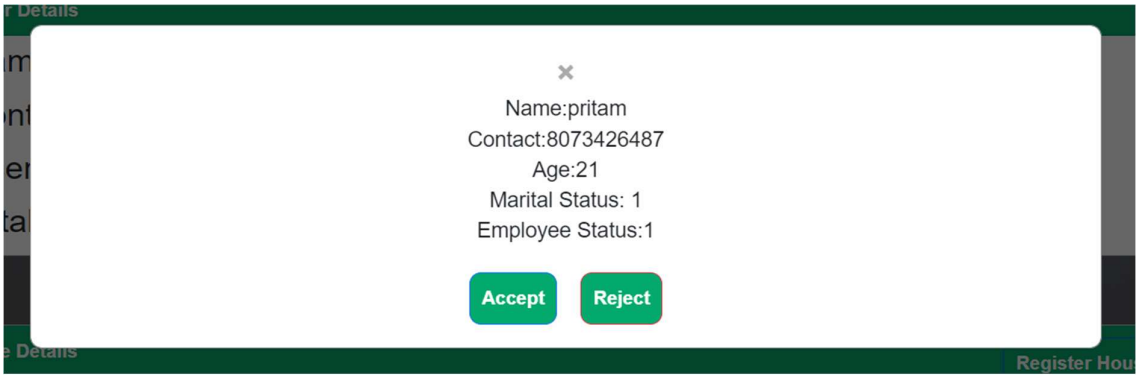
At user side



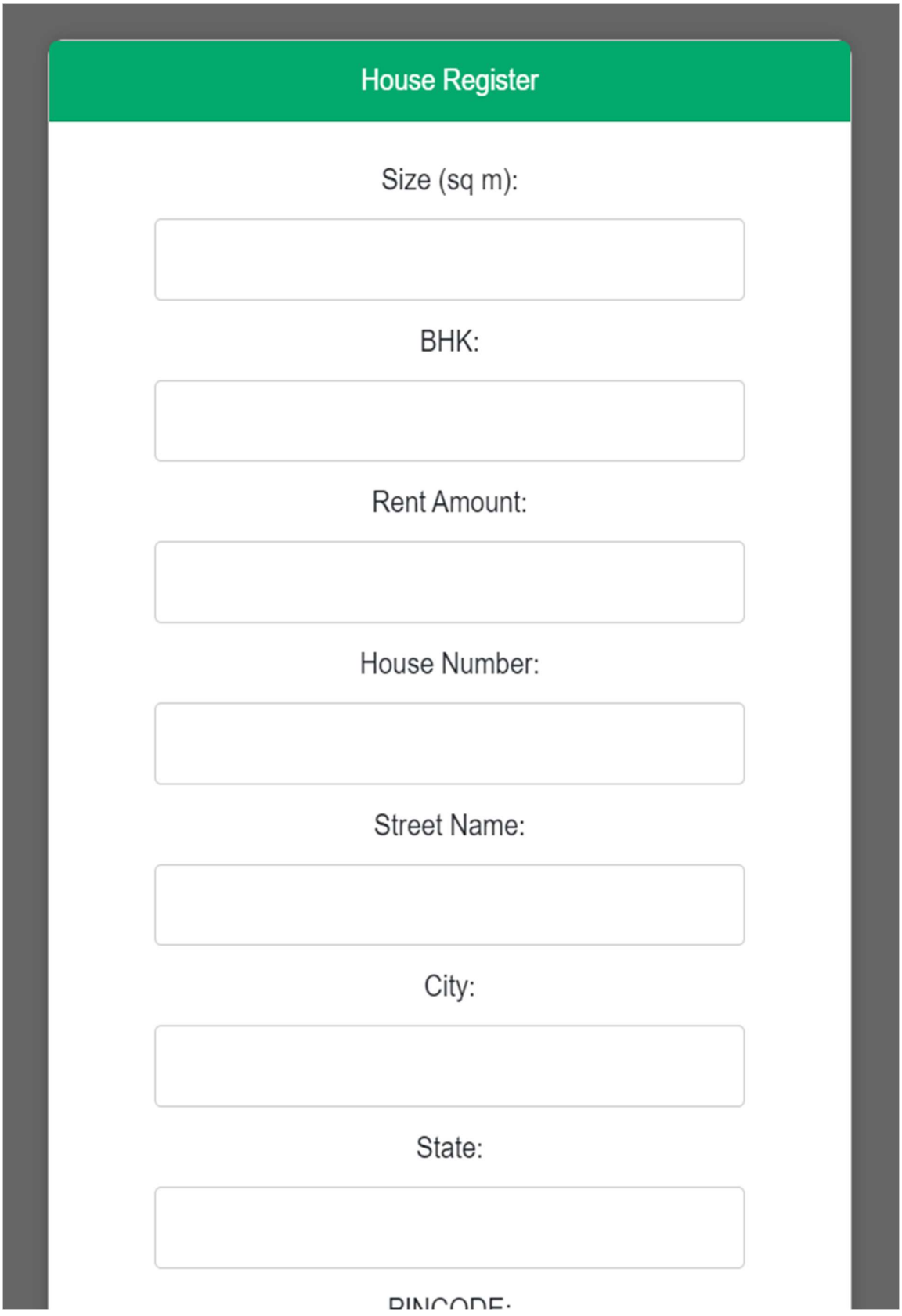
At land_lord req:



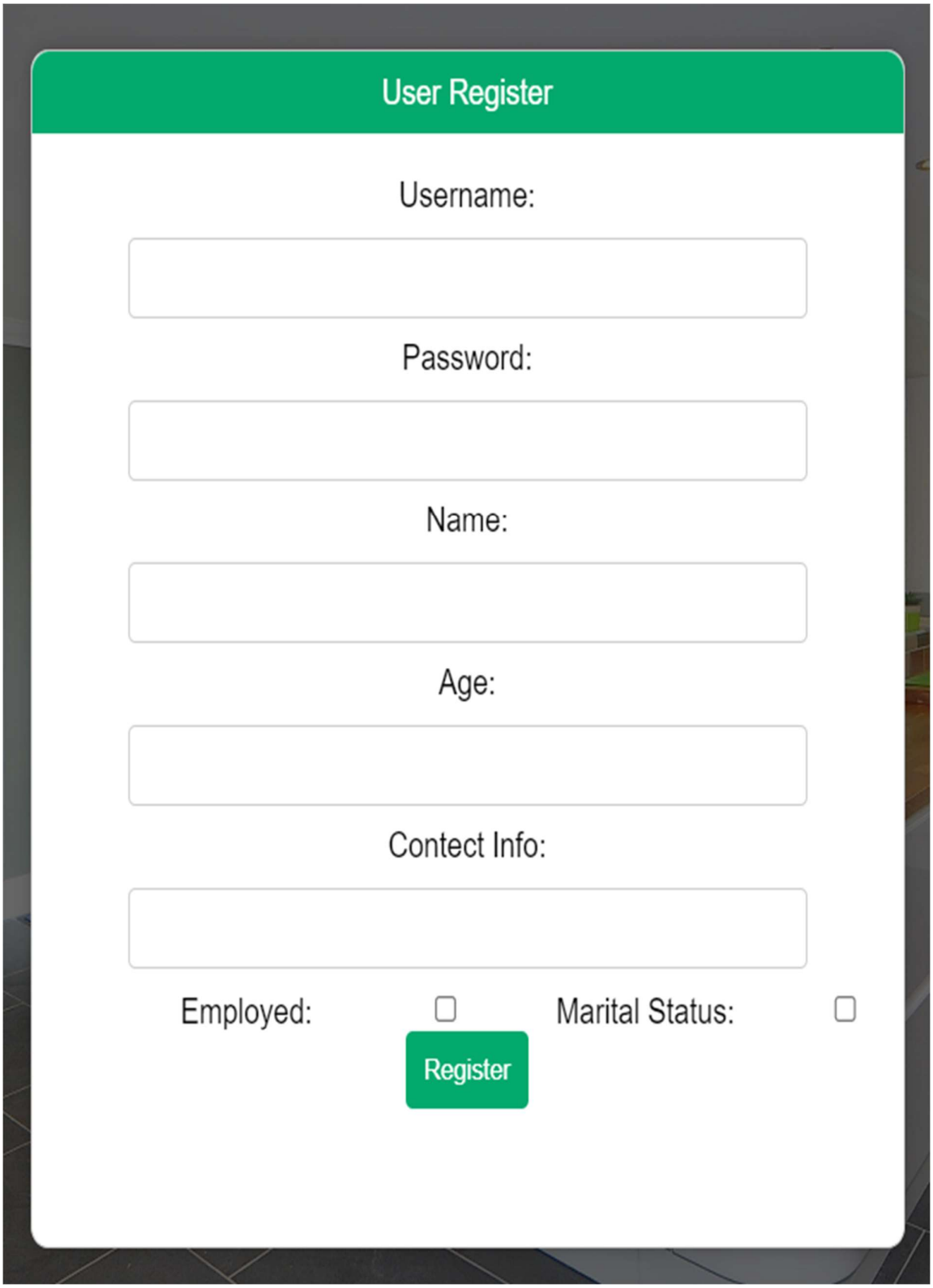
House Request Approval:(Only for Land_lord)



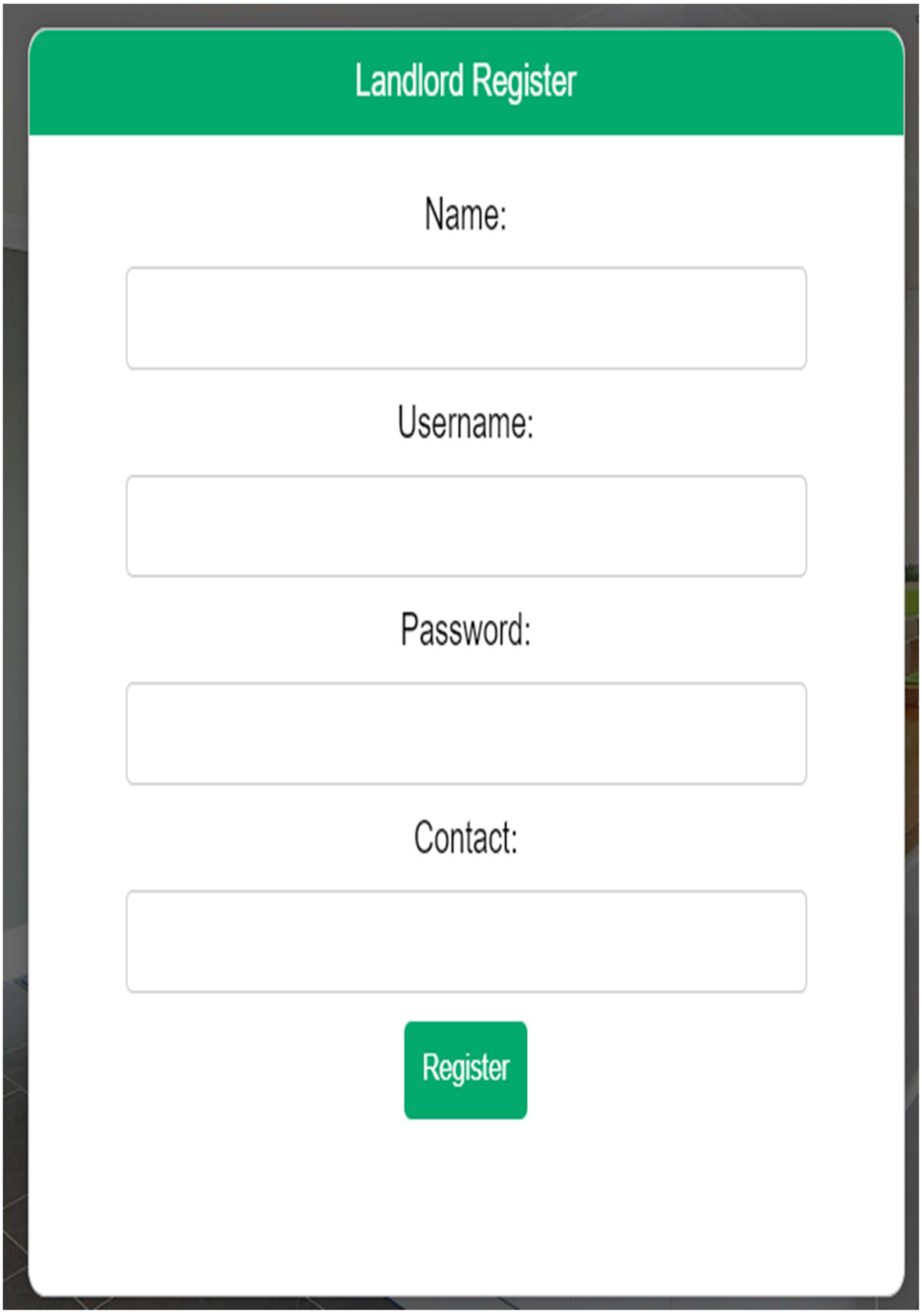
House Register Page(Only for Land_lord)



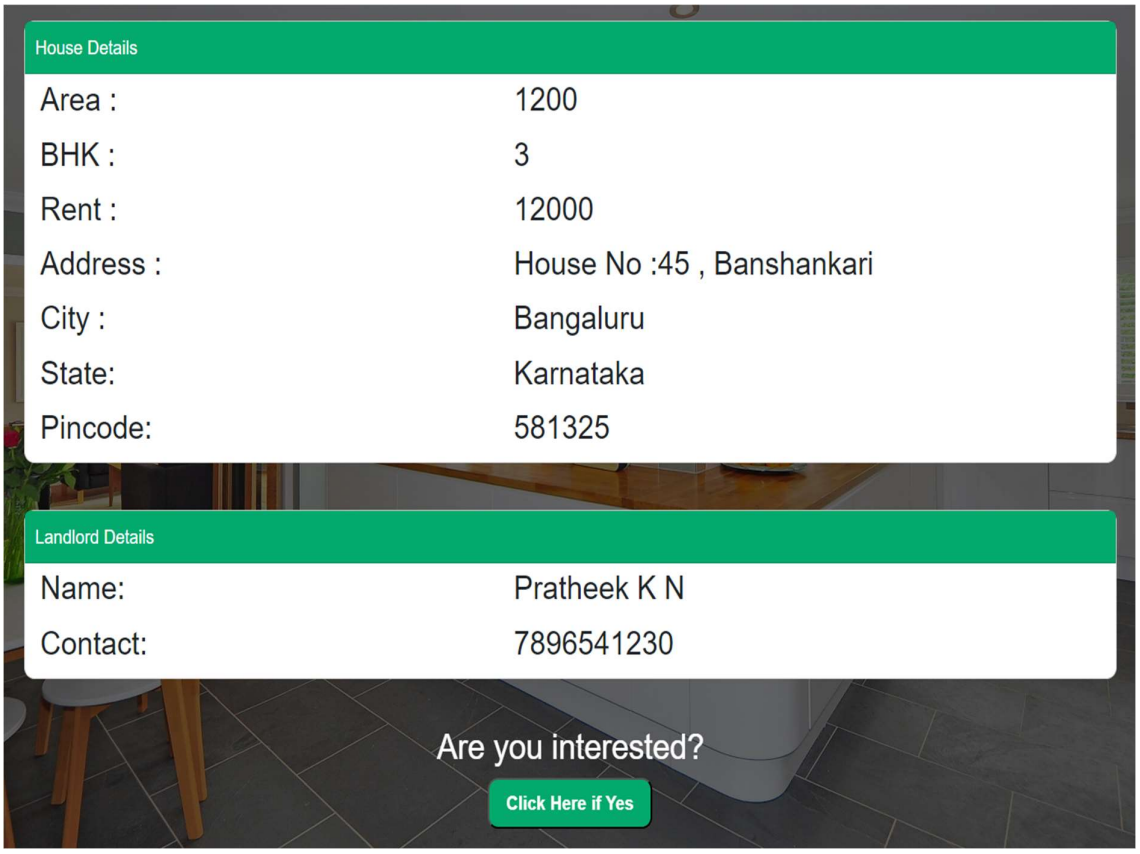
User Register (Tenant Register Page)



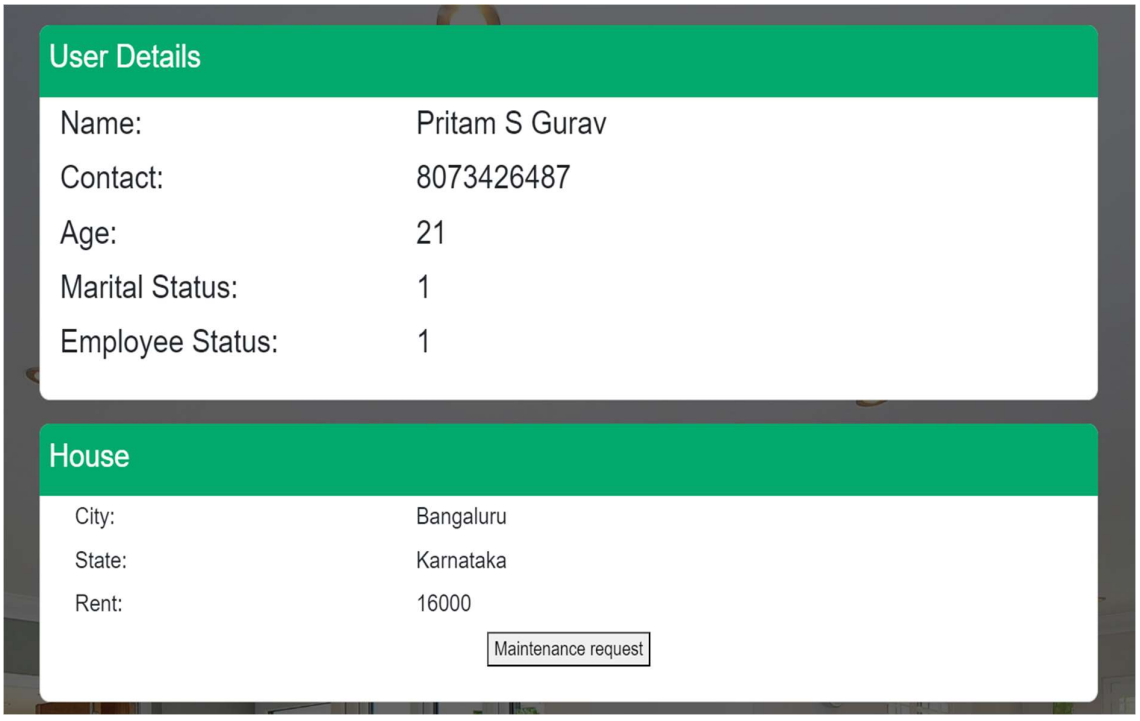
Owner Register:

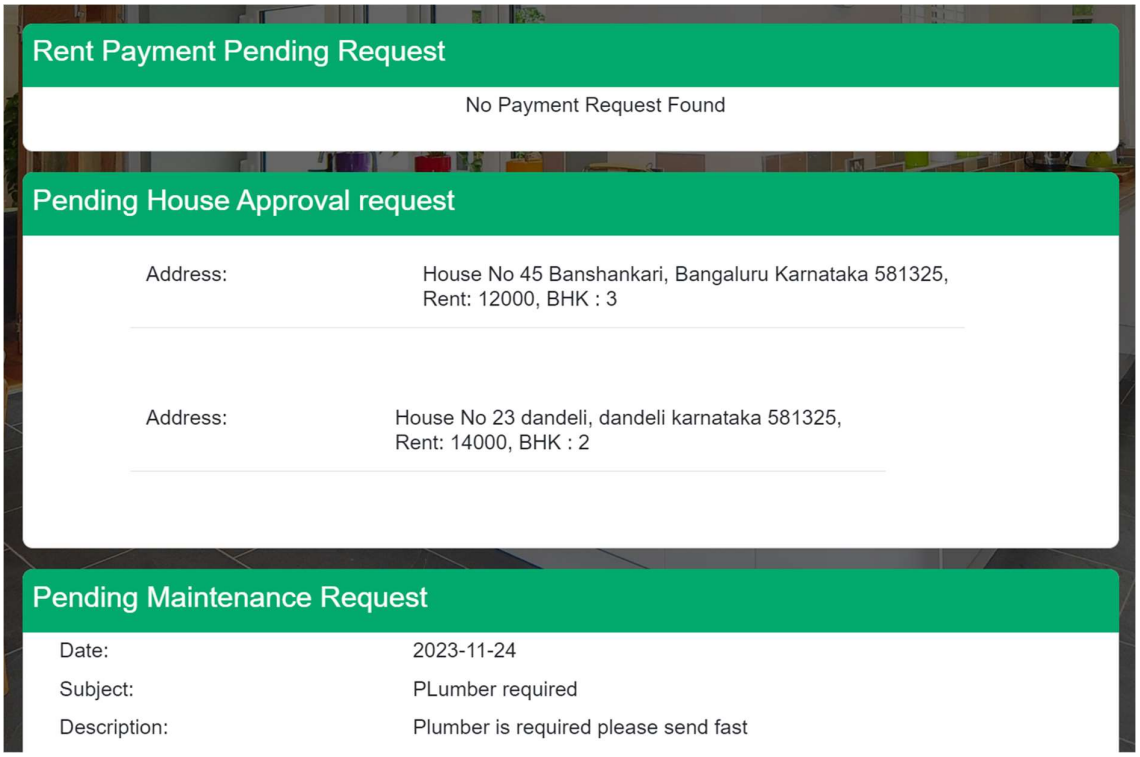


House_Details Page (Only for User ot tenant)

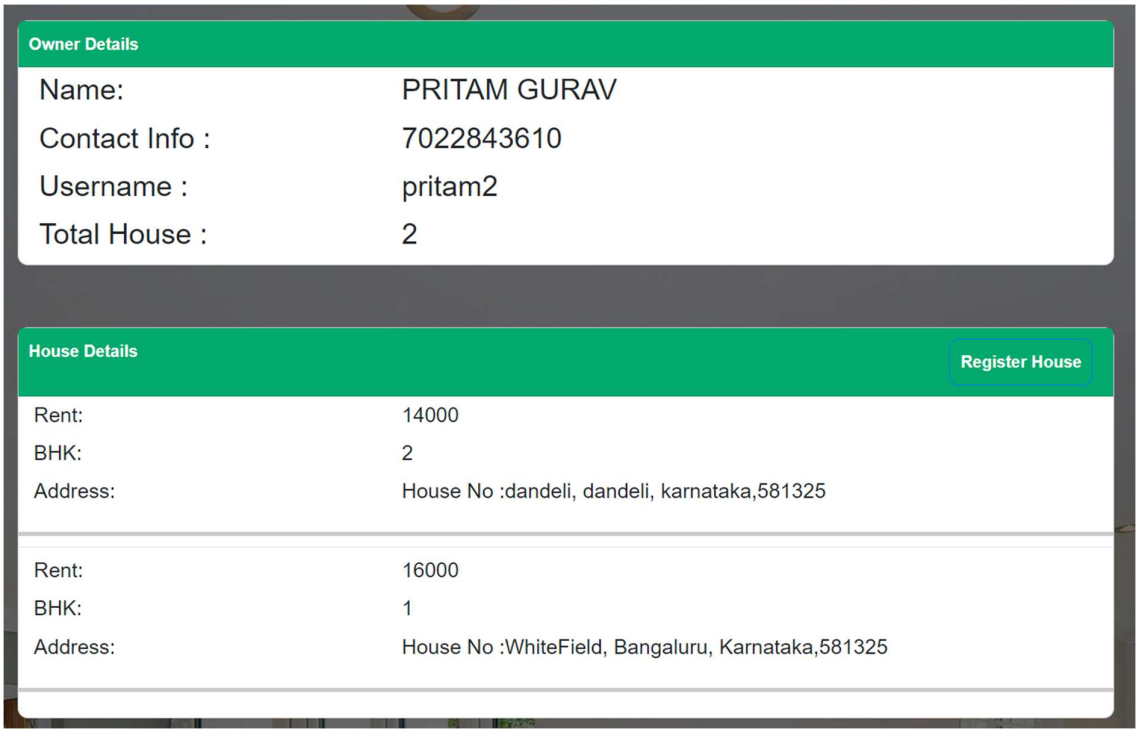


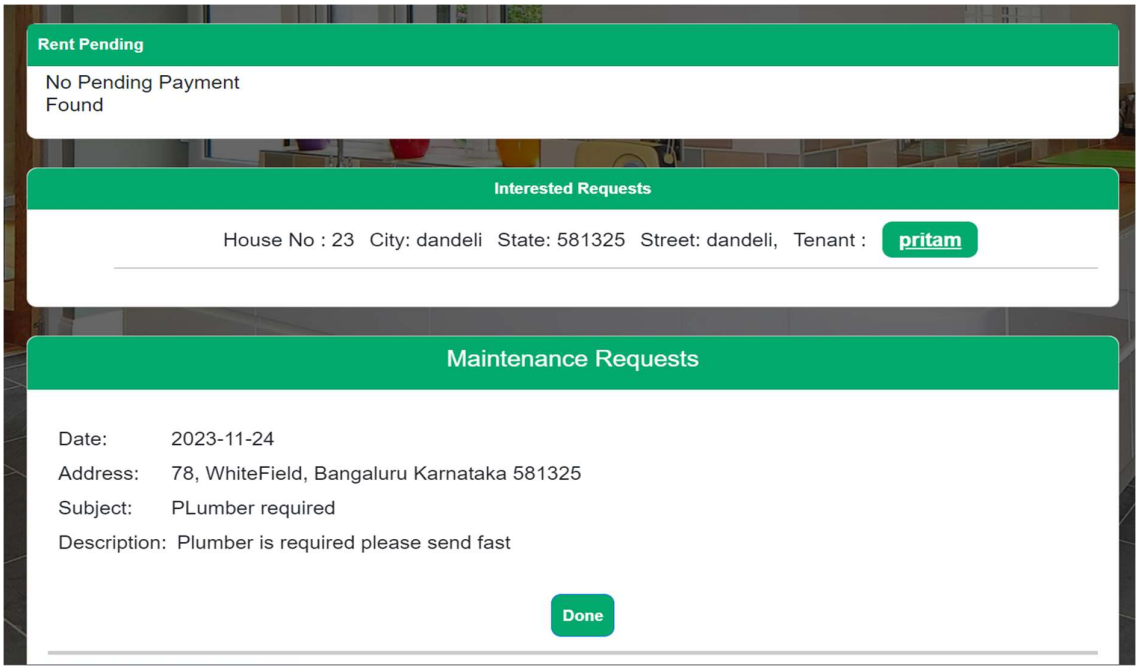
User page



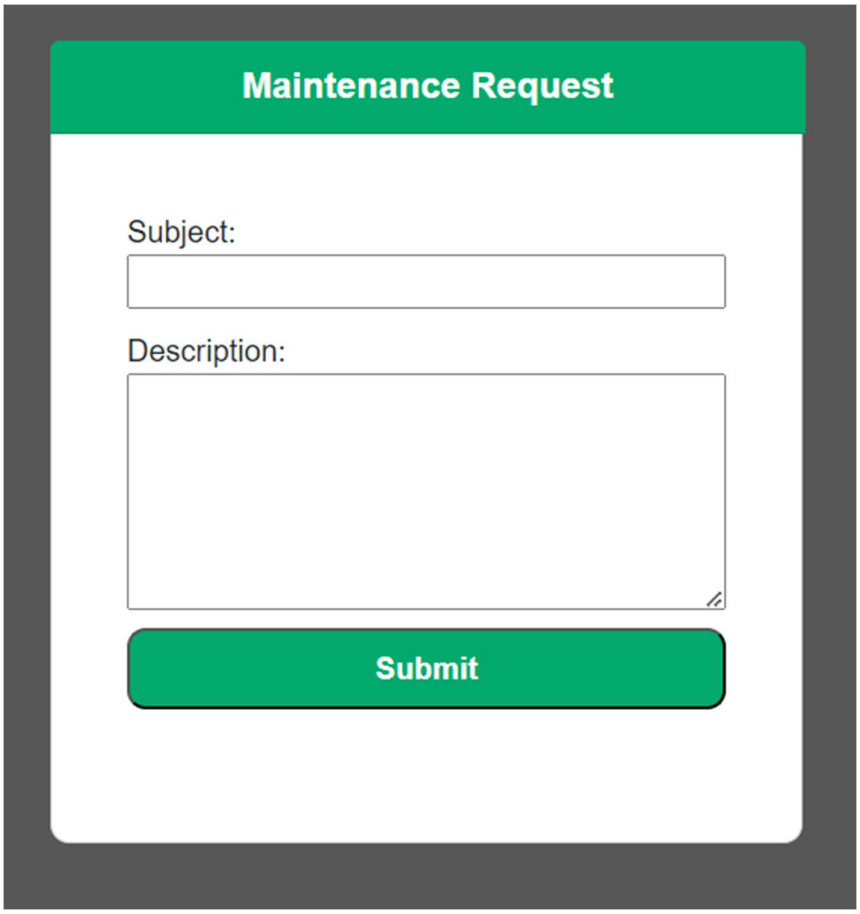

Rent Payment Pending Request
No Payment Request Found
Pending House Approval request
Address:
House No 45 Banshankari, Bangaluru Karnataka 581325,
Rent: 12000, BHK : 3
Address:
House No 23 dandeli, dandeli karnataka 581325,
Rent: 14000, BHK : 2
Pending Maintenance Request
Date:
2023-11-24
Subject:
PLumber required
Description:
Plumber is required please send fast

Owner Page:


Owner Details
Name:
PRITAM GURAV
Contact Info :
7022843610
Username :
pritam2
Total House :
2
House Details
Register House
Rent:
14000
BHK:
2
Address:
House No :dandeli, dandeli, karnataka,581325
Rent:
16000
BHK:
1
Address:
House No :WhiteField, Bangaluru, Karnataka,581325



Maintenance request form



Procedure and triggers:

Procedure:

Procedure Defination

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `delete_interested_with_house_id`(IN house_id_to_delete INT)
BEGIN

    DELETE FROM interested
    WHERE house_id = house_id_to_delete;
END
```

Calling the precedure

```
cur.execute(f"call delete_interested_with_house_id({accept_id[0]})")
db.commit()
```

Trigger:

For Invoking Trigger (after_tenant_update)

```
cur.execute(f"update tenant set house_id = {accept_id[0]} where tenant_id = '{accept_id[1]}'")
db.commit()
```

When there is an update in tenant the trigger will invoke

Trigger Defination

```
after_tenant_update | UPDATE | tenant | BEGIN
IF NEW.house_id IS NOT NULL THEN
  UPDATE house
  SET availibility_status = 0
  WHERE house_id = NEW.house_id;
END IF;
```

Trigger Result:

Before Trigger –

```
mysql> select * from house;
+----------+------+-----+-------------+----------+--------------+-----------+---------+-----------+---------------------+--------------+
| house_id | size | bhk | rent_amount | house_no | street_name  | city      | pincode | state     | availibility_status | land_lord_id |
+----------+------+-----+-------------+----------+--------------+-----------+---------+-----------+---------------------+--------------+
|       24 |  400 |   2 |       20000 | 23       | Banashakari  | Bangaluru | 581325  | karnataka |                   1 |           12 |
|       25 | 4586 |   3 |       30000 | 80       | Jayanagar    | Bangaluru | 581325  | karnataka |                   0 |           12 |
|       26 | 4500 |   1 |        8000 | 5        | Bhogur       | Hyderabad | 581340  | Karnataka |                   1 |           12 |
|       27 | 1000 |   2 |       15000 | 78       | Bangur Nagar | Dandeli   | 581369  | Karnataka |                   1 |           12 |
|       29 | 2500 |   2 |       14000 | 23       | dandeli      | dandeli   | 581325  | karnataka |                   1 |           14 |
|       30 | 2500 |   1 |       16000 | 78       | WhiteField   | Bangaluru | 581325  | Karnataka |                   0 |           14 |
|       31 | 1200 |   3 |       12000 | 45       | Banshankari  | Bangaluru | 581325  | Karnataka |                   1 |           12 |
+----------+------+-----+-------------+----------+--------------+-----------+---------+-----------+---------------------+--------------+
7 rows in set (0.00 sec)

mysql> select * from tenant;
+-----------+----------+-----------+-----+--------------+-----------------+----------------+----------------+----------+
| tenant_id | username | password  | age | contact_info | employment_info | tanant_name    | marital_status | house_id |
+-----------+----------+-----------+-----+--------------+-----------------+----------------+----------------+----------+
|         8 | pritam   | qwerty123 |  21 | 8073426487   | 1               | Pritam S Gurav | 1              |       30 |
|        12 | garuda   | qwerty123 |  45 | 8073426487   | 1               | Garuda Naik    | 0              |       25 |
|        13 | hello    | world     |  45 | 9876543210   | 1               | isfjnkn        | 0              |     NULL |
+-----------+----------+-----------+-----+--------------+-----------------+----------------+----------------+----------+
3 rows in set (0.61 sec)
```

The availibity of house with house_id 30 is 1 and tenant is tenant_id 8

After Trigger :

When there is update in tenant table then it will set avilibity of previous house to 1

```
mysql> select * from house;
+----------+------+-----+-------------+----------+--------------+-----------+---------+-----------+---------------------+--------------+
| house_id | size | bhk | rent_amount | house_no | street_name  | city      | pincode | state     | availibility_status | land_lord_id |
+----------+------+-----+-------------+----------+--------------+-----------+---------+-----------+---------------------+--------------+
|       24 |  400 |   2 |       20000 | 23       | Banashakari  | Bangaluru | 581325  | karnataka |                   0 |           12 |
|       25 | 4586 |   3 |       30000 | 80       | Jayanagar    | Bangaluru | 581325  | karnataka |                   0 |           12 |
|       26 | 4500 |   1 |        8000 | 5        | Bhogur       | Hyderabad | 581340  | Karnataka |                   1 |           12 |
|       27 | 1000 |   2 |       15000 | 78       | Bangur Nagar | Dandeli   | 581369  | Karnataka |                   1 |           12 |
|       29 | 2500 |   2 |       14000 | 23       | dandeli      | dandeli   | 581325  | karnataka |                   1 |           14 |
|       30 | 2500 |   1 |       16000 | 78       | WhiteField   | Bangaluru | 581325  | Karnataka |                   1 |           14 |
|       31 | 1200 |   3 |       12000 | 45       | Banshankari  | Bangaluru | 581325  | Karnataka |                   1 |           12 |
+----------+------+-----+-------------+----------+--------------+-----------+---------+-----------+---------------------+--------------+
7 rows in set (0.00 sec)
```