UNIVERSITY OF HOUSTON

A PROJECT REPORT ON

HOUSE SALES MANAGEMNET SYSTEM PROJECT

By

GROUP – IV

Tejas Murali (Business rules, ERD, Data Entries, Data Dictionary, Database, Management)

Abdul Aziz Mohammed (Database, Data Entries, Report)

Akshitha Reddy Thokala (Front End, Database)

Jaswanthi Boyapati(Front end modifications, Database, Data Entries, Report)

Jayanth Varma Konduru(Report)

Prudhvi Kolla (Front end, Database Connection, Database)

Sai Praneeth Varma Kalidindi(Business Rules, Data Analysis and querying)

Vineeth Yadama-(Report, Data Dictionary)

Saikiran Anugam-(Business Rules, Data dictionary, ERD, Database)

FNU Syed Sohaib Ali(Data base, Front end modifications, Data Entries, Report, Data Entries)

Under the guidance of

PROF. LUCY NWOSU

INTODUCTION

Our project, the Web-Based House Sales Management System (HSMS): a cutting-edge solution poised to redefine the real estate landscape. HSMS is engineered to transform the way properties are bought and sold. Anchored by a robust database architecture, HSMS ensures seamless storage and retrieval of crucial property information, including listings, pricing data, and user preferences. With a steadfast commitment to data integrity and security, HSMS establishes a solid framework for streamlined property management. By prioritizing connectivity and user-friendly interfaces, our system empowers both buyers and sellers with intuitive tools for property search, selection, and transaction management. HSMS is not just a platform; it's a catalyst for informed decision-making, operational efficiency, and unparalleled user experience in the realm of residential property transactions. Welcome to the future of real estate management.

KEY FEATURES AND FUNCTIONALITIES

- ***Effortless Property Transaction:*** The House Sales Management System (HSMS) redefines the real estate landscape with a suite of innovative features designed to streamline property transactions and enhance user experience. At its core, HSMS facilitates seamless property transactions, enabling users to effortlessly initiate and manage multiple transactions with clarity and accountability. Each transaction is intricately associated with specific users, ensuring transparency and ease throughout the buying and selling process.

- ***Dynamic Property Listings:*** Managing property listings is made intuitive and dynamic with HSMS. Users can easily add properties to multiple listings, while maintaining clarity and distinction for each property. This flexibility empowers users to showcase properties effectively and efficiently, maximizing exposure and potential buyers' interest.

- ***Flexible Property Search and Selection:*** HSMS offers users flexible tools for property search and selection, enhancing the overall user experience. The system's advanced search functionalities enable users to filter properties based on their preferences, ensuring a tailored and efficient search process. Whether users are looking for specific features or locations, HSMS provides the tools to find the perfect property match.

- ***Personalized User Cart:*** Personalization is key in the property selection process, and HSMS delivers with its personalized user cart feature. Users can add multiple properties to their cart, with each item clearly associated with a single user. This allows for easy comparison and management of selected properties throughout the buying journey, enhancing user convenience and satisfaction.

- ***Comprehensive User Management:*** HSMS prioritizes user management, allowing administrators to associate multiple properties and transactions with a single user.

ENTITIES

1. Admin
2. Messages
3. Property
4. Requests
5. Saved
6. Users
7. Payment
8. Appointments

BUSINESS RULES

1.Admins

One-to-Many

Admins to Property Listings: Each admin can manage multiple property listings. (One admin, many property listings)

Admins to Appointments: Each admin may handle multiple appointments. (One admin, many appointments)

2.Messages

One-to-One:

Messages to Users: Each message is associated with one user. (One message, one user)

3.Property:

One-to-Many:

Property to Users: Each property listing is managed by one user. (One property, one user)

Property to Appointments: Each property can have multiple appointments. (One property, many appointments)

Many-to-One:

Property to Requests: Many requests can be associated with one property. (Many requests, one property)

4.Requests:

Many-to-One:

Requests to Property: Many requests can be made for one property. (Many requests, one property)

Requests to Users: Many requests can be sent by one user. (Many requests, one user)

5.Saved:

Many-to-One:

Saved to Users: Many saved properties can be associated with one user. (Many saved properties, one user)

Saved to Property: Many users can save one property. (Many users, one property)

6.Users:

One-to-Many:

Users to Messages: One user can send/receive multiple messages. (One user, many messages)

Users to Requests: One user can make multiple property requests. (One user, many requests)

Users to Saved: One user can save multiple properties. (One user, many saved properties)

Users to Appointments: One user can schedule multiple appointments. (One user, many appointments)

Many-to-One:

Users to Admins: Many users may interact with one admin. (Many users, one admin)

Users to Property: Many users may own or manage one property. (Many users, one property)

7.Payments:

Many-to-One:

Payments to Users: Many payments can be made by one user. (Many payments, one user)

One to One:

Payments to Property: One payment can be made for one property. (One payments, one property)

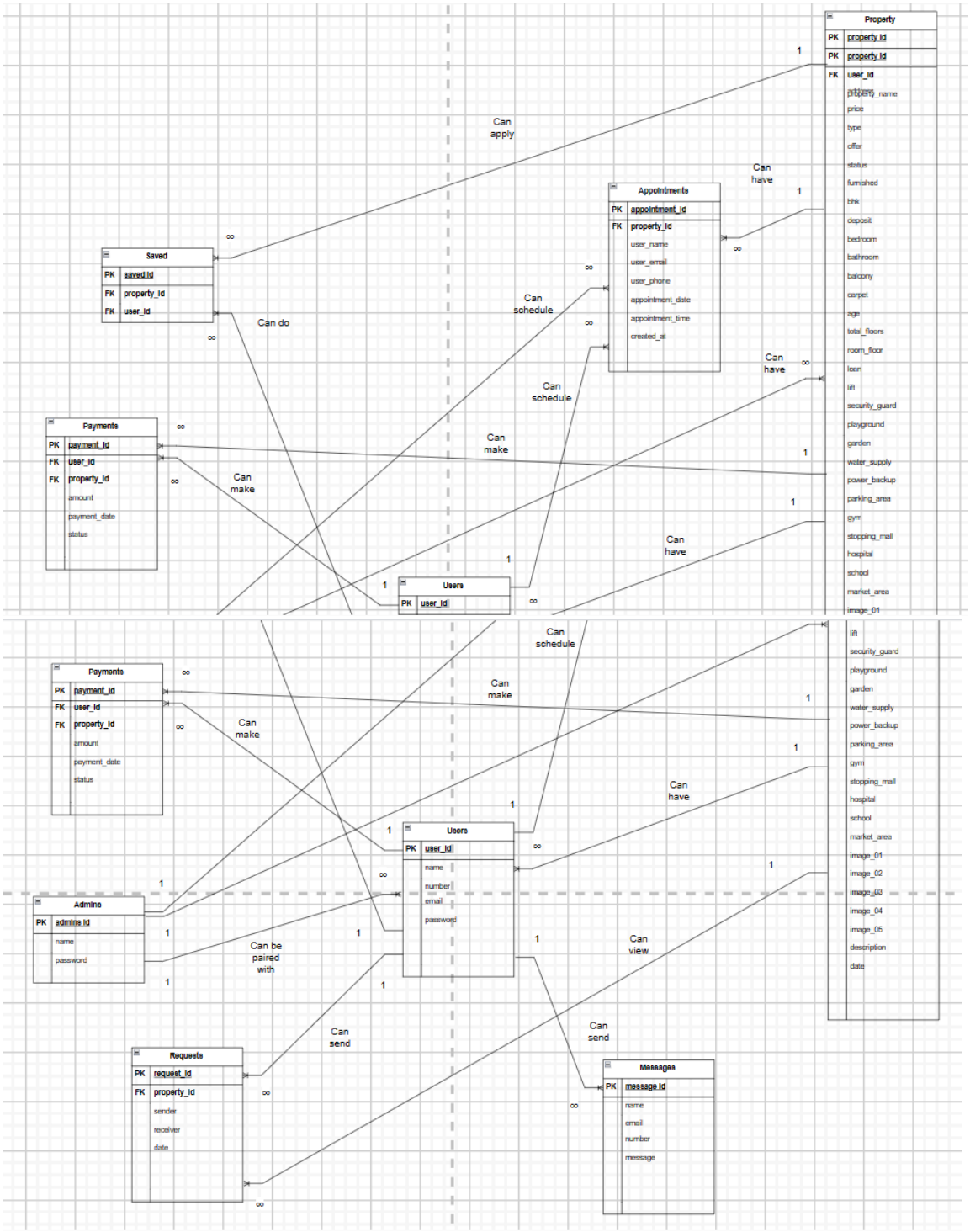
8. Appointment Reminders:

Many-to-One:

Appointments to Users: Many appointments can be scheduled by one user. (Many appointments, one user)

These sections outline the relationships and business rules for your system entities, providing a comprehensive view of how they interact and function within your application.

ENTITY RELATIONSHIP DIAGRAM



Meta Data (Data Dictionary)

Table Name	Attribute Name	Attribute Content	Attribute Data Type	Attribute Required?	PK or FK	FK Reference Table
ADMINS	Admin_id	Admin id	VARCHAR(20)	Y	PK	
	Name	Name	VARCHAR(20)	Y		
	Password	Password	VARCHAR(50)	Y		
MESSAGES	Messages_id	Message id	VARCHAR(20)	Y	PK	
	Name	Name	VARCHAR(50)	Y		
	Email	Email id	VARCHAR(50)	Y		
	Phone_Number	Phone Number	VARCHAR(20)	Y		
	Message	Message	VARCHAR(20)	Y		
PROPERTY	Property_id	Property Id	VARCHAR(20)	Y	PK	
	User_id	User Id	VARCHAR(20)	Y	FK	USERS
	Property_Name	Property Name	VARCHAR(50)	Y		
	Address	Address	VARCHAR(100)	Y		
	Price	Price	VARCHAR(10)	Y		
	Type	Type	VARCHAR(10)	Y		
	Offer	Offer	VARCHAR(10)	Y		
	Status	Status	VARCHAR(50)	Y		
	Furnished	Furnished	VARCHAR(50)	Y		
	bhk	bedroom Hall Kitchen	VARCHAR(10)	Y		
	deposite	deposite	VARCHAR(10)	Y		
	bedroom	bedroom	VARCHAR(10)	Y		
	bathroom	bathroom	VARCHAR(10)	Y		
	balcony	balcony	VARCHAR(10)	Y		
	carpet	carpet	VARCHAR(10)	Y		
	age	age	VARCHAR(2)	Y		
	total_floors	Total Floors	VARCHAR(2)	Y		
	room_floor	Room Floor	VARCHAR(2)	Y		
	loan	loan	VARCHAR(50)	Y		
	lift	lift	VARCHAR(3)	Y		
	security_guard	Security Guard	VARCHAR(3)	Y		
	play_ground	Play Ground	VARCHAR(3)	Y		
	garden	garden	VARCHAR(3)	Y		
	water_supply	Water_Supply	VARCHAR(3)	Y		
	power_backup	Power Backup	VARCHAR(3)	Y		
	parking_area	Parking Area	VARCHAR(3)	Y		
	gym	gym	VARCHAR(3)	Y		
	shopping_mall	Shopping Mall	VARCHAR(3)	Y		
	hospital	hospital	VARCHAR(3)	Y		
	school	school	VARCHAR(3)	Y		
	market_area	Market Area	VARCHAR(3)	Y		
	image_01	Image 01	VARCHAR(50)	Y		

	image_02	Image 02	VARCHAR(50)	Y		
	image_03	Image 03	VARCHAR(50)	Y		
	image_04	Image 04	VARCHAR(50)	Y		
	image_05	Image 05	VARCHAR(50)	Y		
	description	description	VARCHAR(1000)	Y		
	date	date	DATE	Y		
REQUESTS	Request_id	Request Id	VARCHAR(20)	Y	PK	
	property_id	property Id	VARCHAR(20)	Y	FK	PROPERTY
	sender	sender	VARCHAR(20)	Y		
	receiver	receiver	VARCHAR(20)	Y		
	date	date	DATE	Y		
SAVED	Saved_id	Saved Id	VARCHAR(20)	Y	PK	
	property_id	property_id	VARCHAR(20)	Y	FK	PROPERTY
	user_id	user_id	VARCHAR(20)	Y	FK	USER
USERS	Users_id	Users Id	VARCHAR(20)	Y	PK	
	name	name	VARCHAR(50)	Y		
	number	number	VARCHAR(10)	Y		
	email	email	VARCHAR(50)	Y		
	password	password	VARCHAR(50)	Y		
PAYMENT	payment_id	Payment Id	INT(11)	Y	PK	
	user_id	User Id	VARCHAR(20)	Y	FK	USER
	property_id	Property Id	VARCHAR(20)	Y	FK	PROPERTY
	amount	Amount	DECIMAL(10,2)	Y		
	payment_date	Payment Date	TIMESTAMP	Y		
	status	Status	VARCHAR(20)	N		
APPOINTMENTS	appointment_id	Appointment Id	INT(11)	Y	PK	
	property_id	Property Id	INT(11)	Y	FK	PROPERTY
	user_name	Username	VARCHAR(255)	Y		
	user_email	User email	VARCHAR(255)	Y		
	user_phone	User phone	VARCHAR(20)	Y		
	appointment_date	Appointment Date	DATE	Y		
	appointment_time	Appointment Time	TIME	Y		
	created_at	Time Created	TIMESTAMP	Y		

QUERIES

1. Table structure for table 'admins'

```
CREATE TABLE admins (
  id varchar(20) NOT NULL PRIMARY KEY,
  name varchar(20) NOT NULL,
  password varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

2. Table structure for table 'messages'

```
CREATE TABLE messages (
  id varchar(20) NOT NULL PRIMARY KEY,
  name varchar(50) NOT NULL,
  email varchar(50) NOT NULL,
  number varchar(10) NOT NULL,
  message varchar(1000) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

3. Table structure for table 'property'

```
CREATE TABLE property (
  id varchar(20) NOT NULL PRIMARY KEY,
  user_id varchar(20) NOT NULL,
  property_name varchar(50) NOT NULL,
  address varchar(100) NOT NULL,
  price varchar(10) NOT NULL,
  type varchar(10) NOT NULL,
  offer varchar(10) NOT NULL,
  status varchar(50) NOT NULL,
  furnished varchar(50) NOT NULL,
  bhk varchar(10) NOT NULL,
  deposite varchar(10) NOT NULL,
  bedroom varchar(10) NOT NULL,
  bathroom varchar(10) NOT NULL,
  balcony varchar(10) NOT NULL,
  carpet varchar(10) NOT NULL,
  age varchar(2) NOT NULL,
  total_floors varchar(2) NOT NULL,
  room_floor varchar(2) NOT NULL,
  loan varchar(50) NOT NULL,
  lift varchar(3) NOT NULL DEFAULT 'no',
```

```
  security_guard varchar(3) NOT NULL DEFAULT 'no',
  play_ground varchar(3) NOT NULL DEFAULT 'no',
  garden varchar(3) NOT NULL DEFAULT 'no',
  water_supply varchar(3) NOT NULL DEFAULT 'no',
  power_backup varchar(3) NOT NULL DEFAULT 'no',
  parking_area varchar(3) NOT NULL DEFAULT 'no',
  gym varchar(3) NOT NULL DEFAULT 'no',
  shopping_mall varchar(3) NOT NULL DEFAULT 'no',
  hospital varchar(3) NOT NULL DEFAULT 'no',
  school varchar(3) NOT NULL DEFAULT 'no',
  market_area varchar(3) NOT NULL DEFAULT 'no',
  image_01 varchar(50) NOT NULL,
  image_02 varchar(50) NOT NULL,
  image_03 varchar(50) NOT NULL,
  image_04 varchar(50) NOT NULL,
  image_05 varchar(50) NOT NULL,
  description varchar(1000) NOT NULL,
  date date NOT NULL DEFAULT current_timestamp(),
  FOREIGN KEY (user_id) REFERENCES users (id) ON DELETE CASCADE ON UPDATE
CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

4. Table structure for table 'requests'

```
CREATE TABLE requests (
  id varchar(20) NOT NULL PRIMARY KEY,
  property_id varchar(20) NOT NULL,
  sender varchar(20) NOT NULL,
  receiver varchar(20) NOT NULL,
  date date NOT NULL DEFAULT current_timestamp(),
  FOREIGN KEY (property_id) REFERENCES property (id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
```

5. Table structure for table 'saved'

```
CREATE TABLE saved (
  id varchar(20) NOT NULL PRIMARY KEY,
  property_id varchar(20) NOT NULL,
  user_id varchar(20) NOT NULL,
  FOREIGN KEY (property_id) REFERENCES property (id),
  FOREIGN KEY (user_id) REFERENCES users (id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

6. Table structure for table 'users'

```
CREATE TABLE users (
  id varchar(20) NOT NULL PRIMARY KEY,
  name varchar(50) NOT NULL,
  number varchar(10) NOT NULL,
  email varchar(50) NOT NULL,
  password varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

7. Table structure for table 'payments'

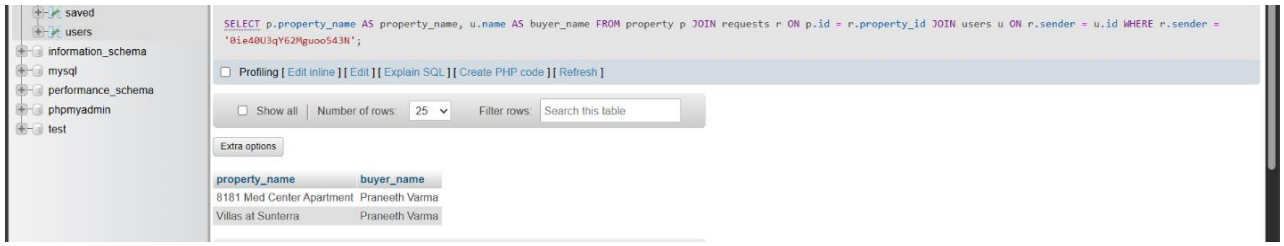
```
CREATE TABLE `payments` (
`id` varchar(20) NOT NULL,
`user_id` varchar(20) NOT NULL,
`property_id` varchar(20) NOT NULL,
`amount` decimal(10, 2) NOT NULL,
`payment_date` TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY (`id`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

8. Table structure for appointments

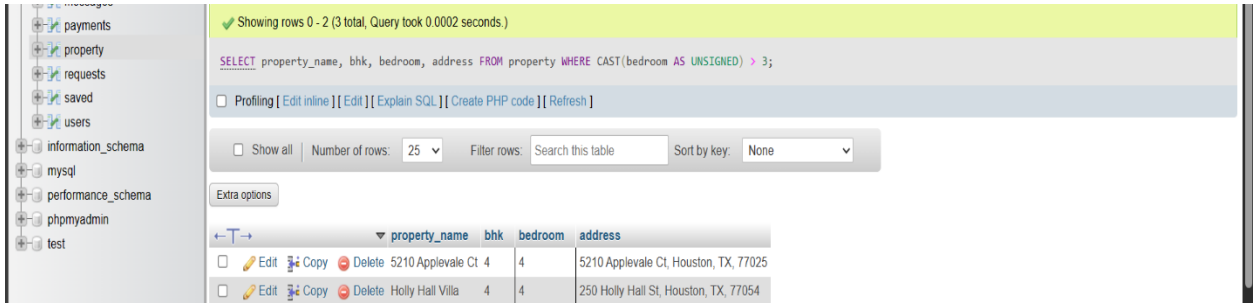
```
CREATE TABLE appointments (
id INT AUTO_INCREMENT PRIMARY KEY,
property_id INT NOT NULL,
user_name VARCHAR(255) NOT NULL,
user_email VARCHAR(255) NOT NULL,
user_phone VARCHAR(20) NOT NULL,
appointment_date DATE NOT NULL,
appointment_time TIME NOT NULL,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

DATA ANALYSIS

1.Retrieve all houses purchased by a specific buyer.

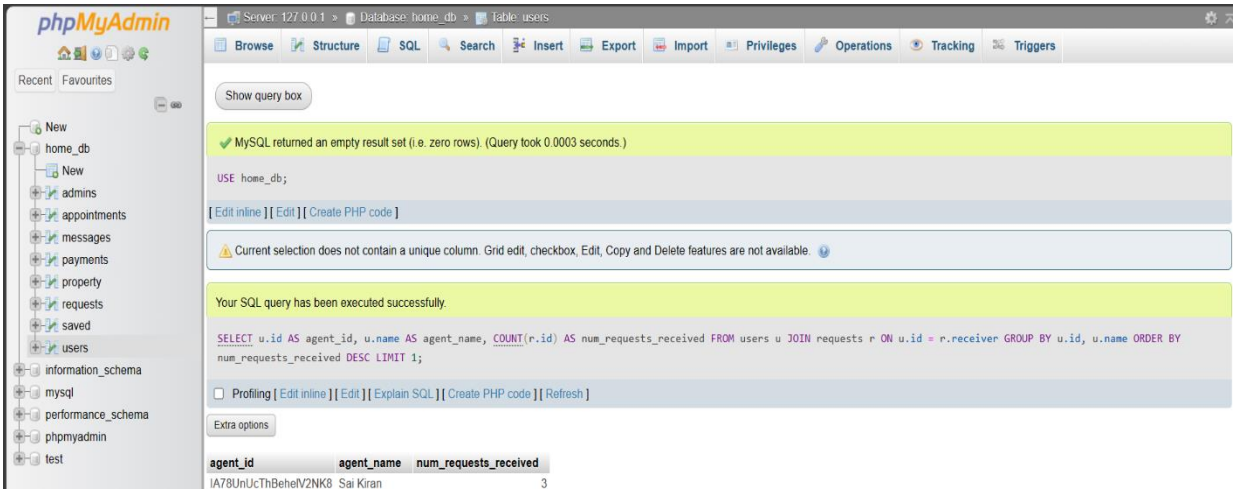


2.Retrieve all houses with more than 3 bedrooms.



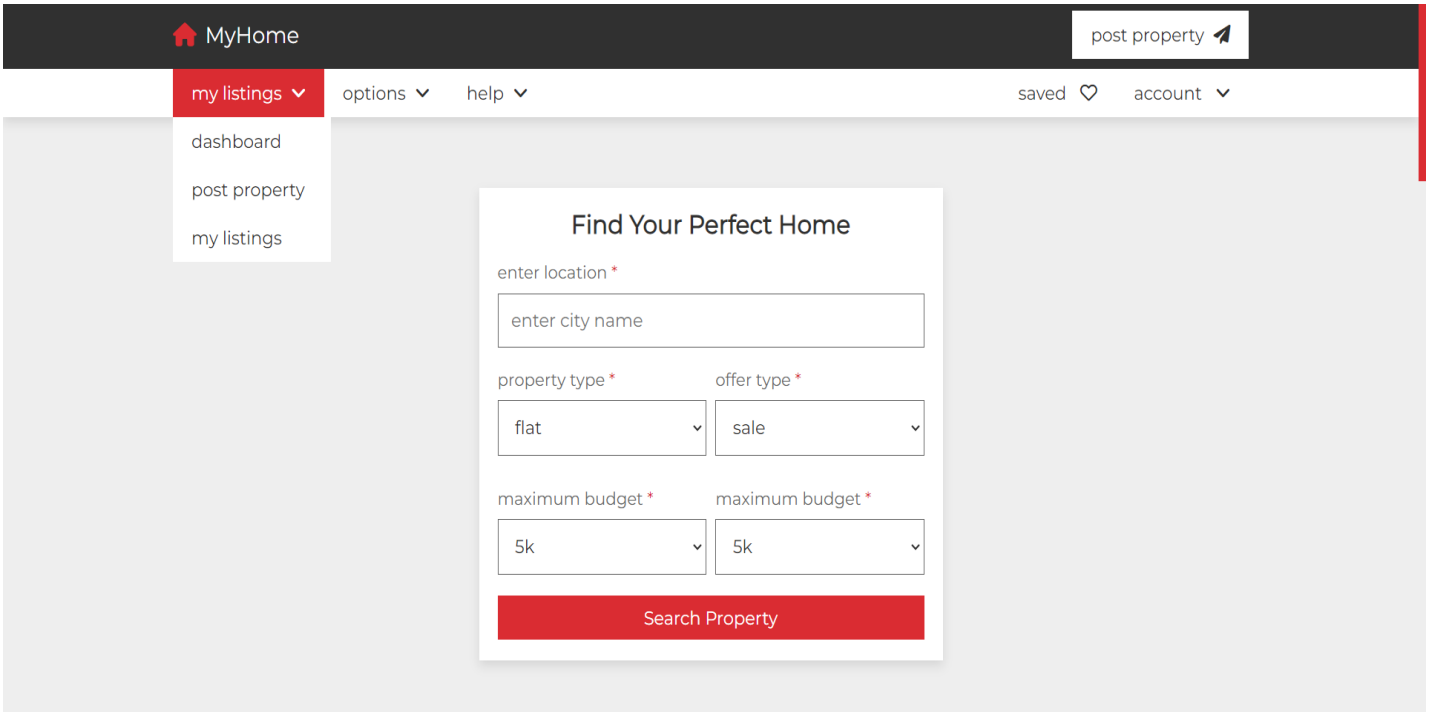
3.Retrieve the total number of houses in a specific neighborhood.

4.Retrieve the top real estate agent with the highest number of sales.

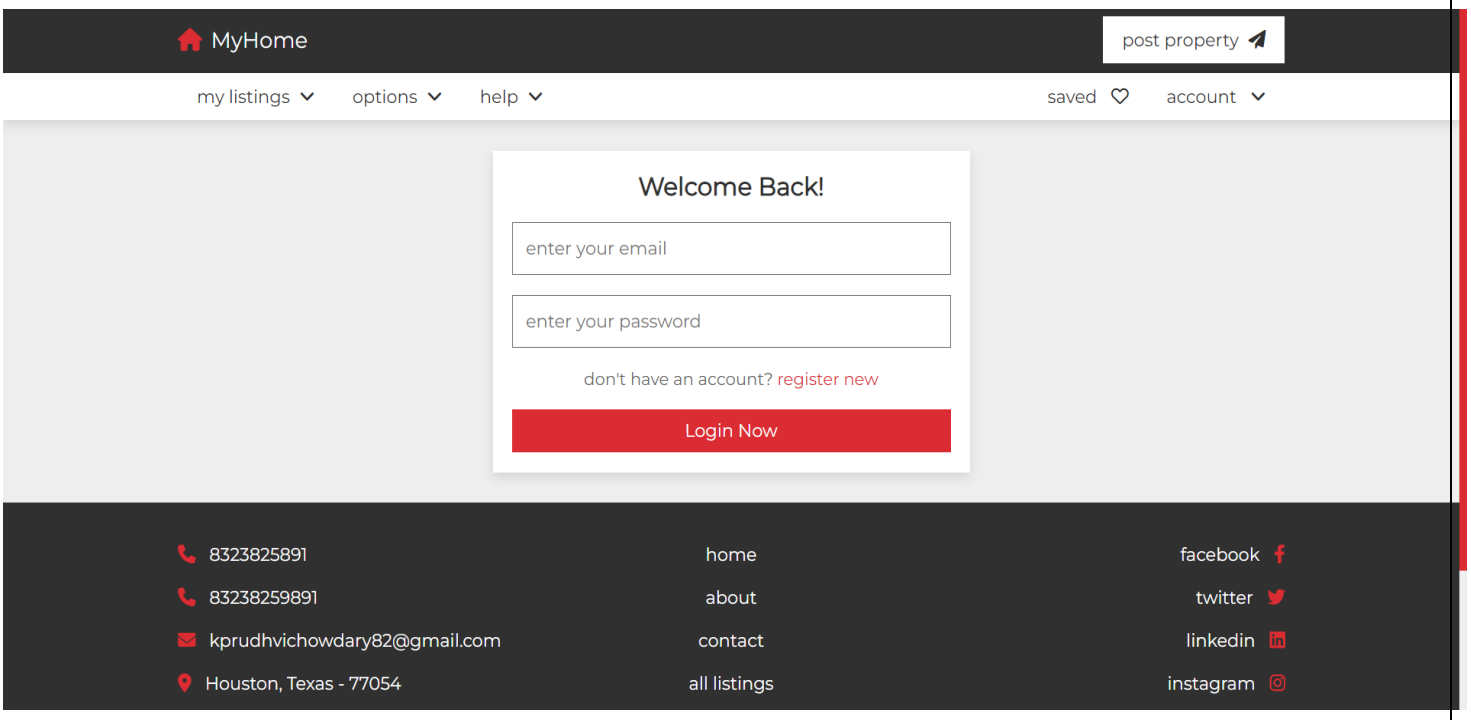


WEBSITE LAYOUT

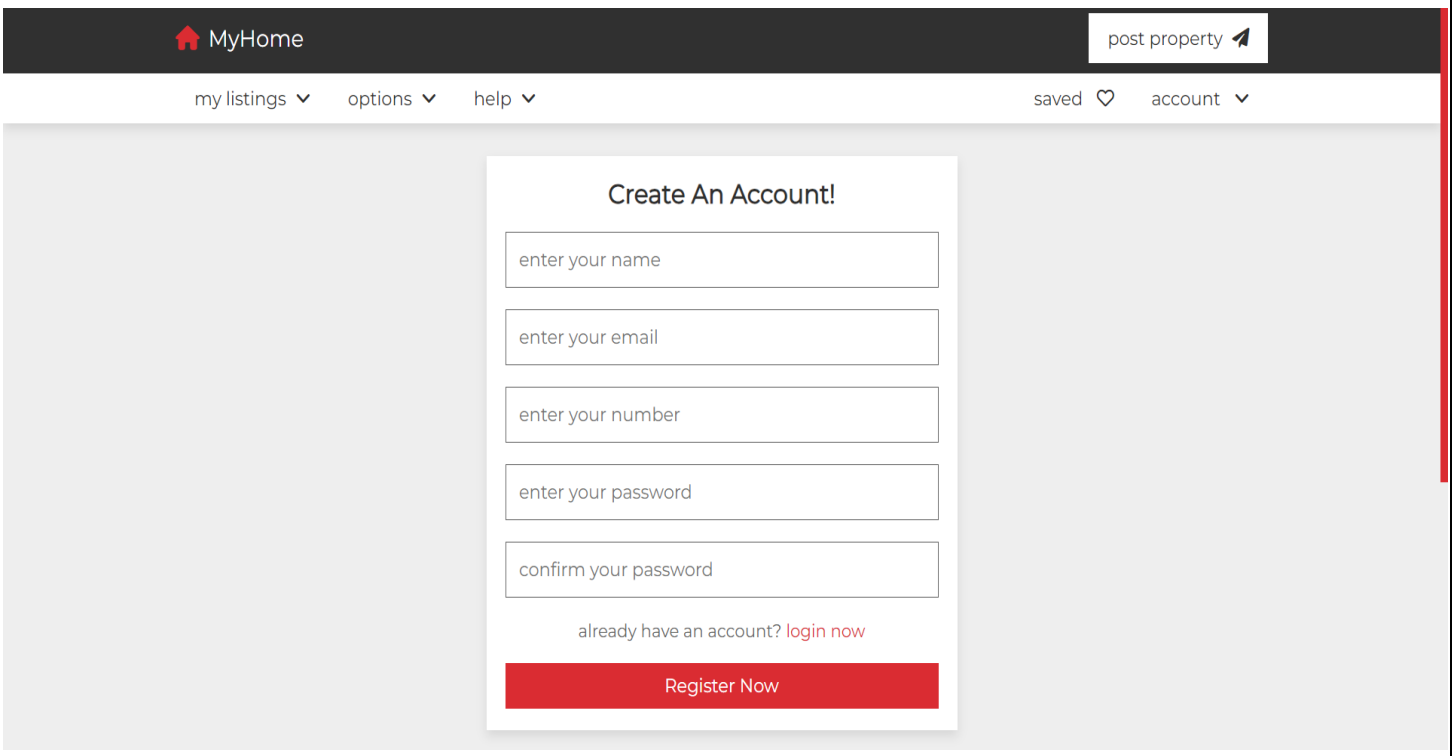
Home Page



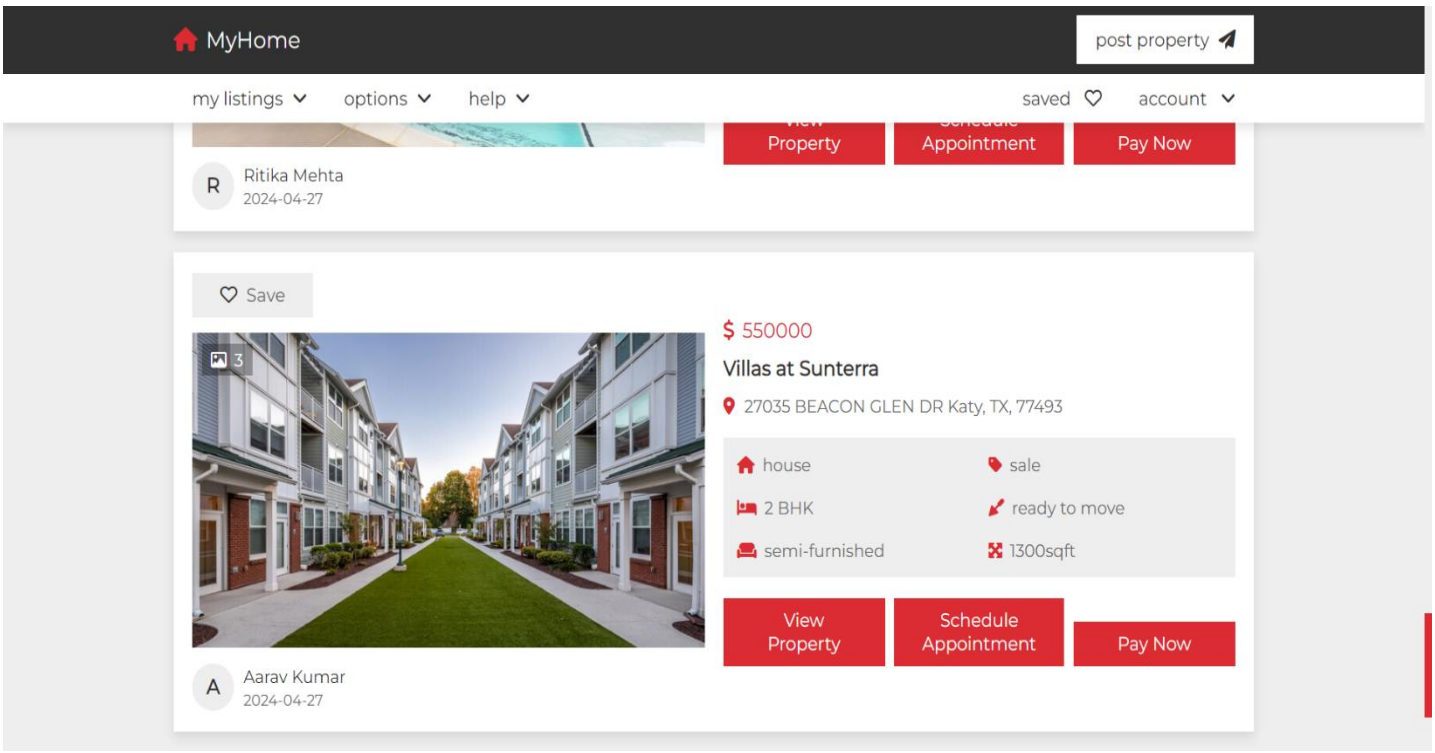
Login page(for user authentication)



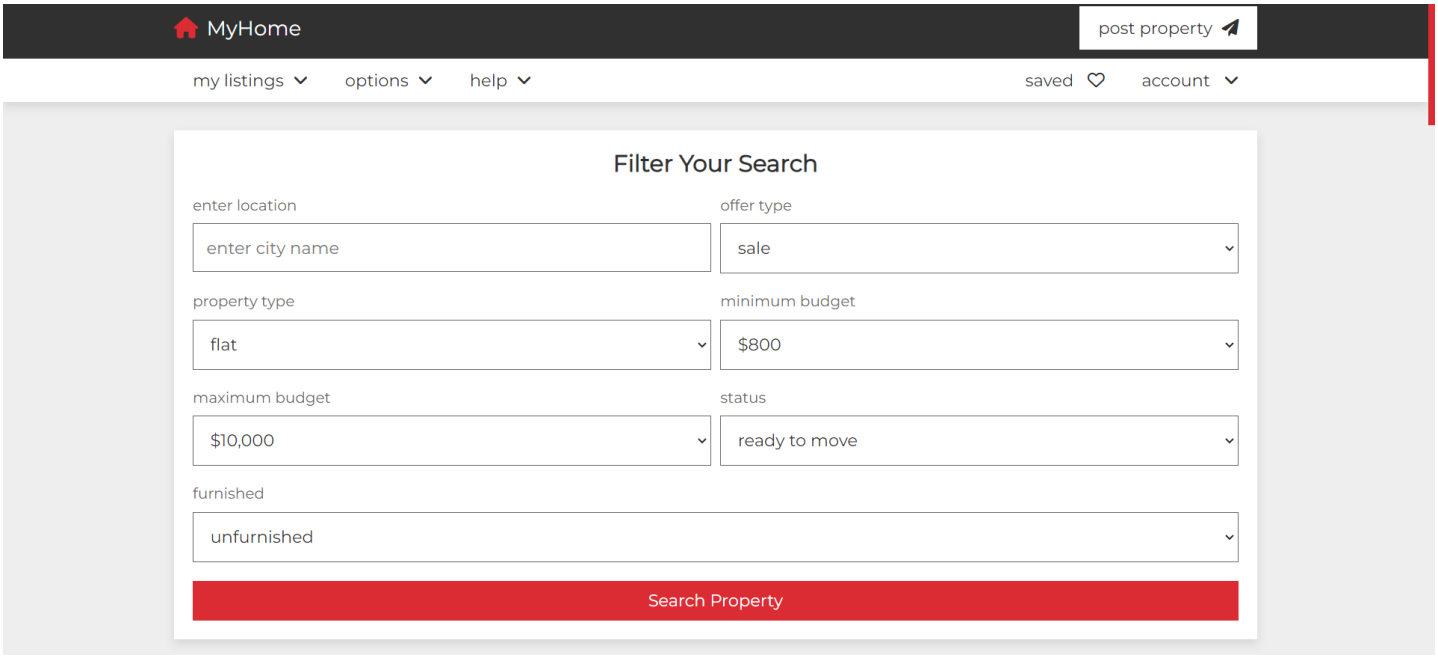
Register page(for user registration)



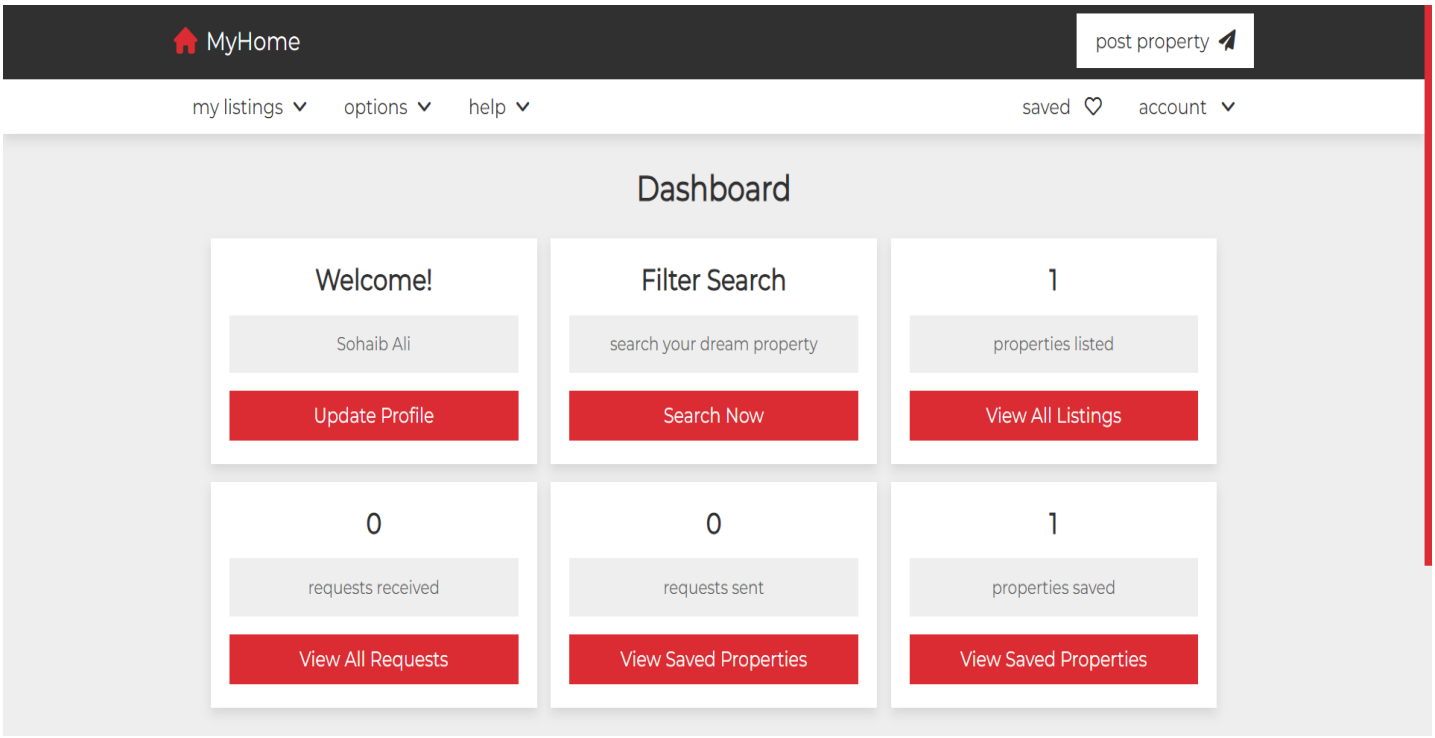
Listing page(for property listing management,buttons on this page lead us to view property description,payments and scheduling appointment)



Filter page(For property search and filtering)



Dashboard Page(for reporting and analytics)



CONCLUSION

In conclusion, the development of the House Sales Management System (HSMS) represents a significant endeavor aimed at modernizing and enhancing the real estate industry's operational efficiency and user experience. Through the careful consideration and implementation of various business rules and system functionalities, HSMS is poised to revolutionize the way properties are bought, sold, and managed.

By enabling seamless interactions between users, properties, appointments, orders, payments, and administrative functions, HSMS fosters transparency, accountability, and convenience throughout the entire property transaction lifecycle. The system's robust database structure, coupled with stringent data integrity and security measures, ensures the confidentiality and reliability of the information stored within the system.

HSMS prioritizes user experience, employing intuitive interfaces, comprehensive search and filtering capabilities, and personalized features to streamline property search, selection, and management processes. The system's scalability, performance optimizations, and rigorous testing procedures further underscore its reliability and suitability for handling large volumes of data and user interactions.

Ultimately, HSMS is not just a software solution but a catalyst for innovation and transformation within the real estate industry.

GITHUB LINK:

https://github.com/tejasmurali1998/House-Sales-Management-Project

https://www.canva.com/design/DAGDrYytZuE/hjubBSMnfxOC6sQy6Z8_bw/edit