

APARTMENT MANAGEMENT SYSTEM

MINI PROJECT

Submitted by

GUNNU JAI RAJ [RA2011003010171]
RAKSHANNA KANTHAN [RA2011003010145]
ANUPAM PALAI [RA2011003010206]

Under the Guidance of

Mr.S.RAMAMORTHY

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING



SCHOOL OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND
TECHNOLOGY KATTANKULATHUR - 603203
MAY 2023



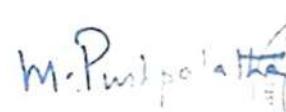
**SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203**

BONAFIDE CERTIFICATE

Certified that this mini project titled “APARTMENT MANAGEMENT SYSTEM” is the bonafide work done by **GUNNU JAI RAJ [RA2011003010171], RAKSHANNA KANTHAN [RA2011003010145], ANUPAM PALAI [RA2011003010206]** of III Year/VI Sem B.tech(CSE) who carried out the mini project work under my supervision for the course 18CSC303J- Database Management systems in SRM Institute of Science and Technology during the academic year 2022-2023 (Even sem).

SIGNATURE

Mr. S. RAMAMORTHY
DBMS – Course Faculty
Department of CTECH



SIGNATURE
Dr. M. Pushpalatha
Head of the Department
Department of computing
technologies.
SRMIST

ABSTRACT

Apartment management entails a wide variety of duties and obligations aimed at making sure the apartment block is kept up well and appealing to both present and prospective tenants. Rent collection and monitoring tenant compliance with lease obligations are two of a flat manager's main responsibilities. This includes keeping track of lease expiration dates, checking the credit and background of prospective renters, and managing eviction processes if necessary.

Maintenance and repair are important components of flat management as well. Buildings and grounds must be kept up-to-date and secure for renters, according to flat managers. This could entail coordinating maintenance with outside contractors, attending to tenant maintenance requests, and performing routine inspections to spot possible concerns before they develop into major difficulties.

Another crucial aspect of flat management is security. Apartment managers are required to take precautions to guarantee the security of the complex, including setting up security cameras, employing security professionals and putting in place access control systems.

Another vital competency for flat managers is the ability to communicate effectively with tenants. In addition to providing updates and information on the complex's operations and policies, they must be readily available to respond quickly and efficiently to tenant problems and complaints.

Overall, a mix of administrative, financial, legal, and interpersonal abilities is needed for successful flat management. The secret to keeping a lucrative and long-lasting residential property is the capacity to strike a balance between the requirements and interests of tenants and the complex's financial and operational goals.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
	ABSTRACT	
	TABLE OF CONTENTS	
	LIST OF FIGURES	
	LIST OF TABLES	
	ABBREVIATIONS	
1	INTRODUCTION	
1.1	Introduction	
1.2	Problem statement	
1.3	Objectives	
1.4	Scope and applications	
1.5	General and Unique Services in the database application	
1.6	Software Requirements Specification	
2	LITERATURE SURVEY	
2.1	Existing system	
2.2	Comparison of Existing vs Proposed System	
3	SYSTEM ARCHITECTURE AND DESIGN	
3.1	Architecture Diagram	
3.1.1	Front-end (UI)design	
3.1.2	Back end (Database) design	
3.2	ER Diagram and Use Case Diagram	
4	Modules and Functionalities	
4.1	List of Modules and Functionalities	
4.1.1	List of Database Relations	
4.2	Connectivity Used for database access	
5	CODING AND TESTING	
6	RESULTS AND DISCUSSIONS	
7	CONCLUSION AND FUTURE ENHANCEMENT	
	REFERENCES	

LIST OF FIGURES

Table No.	Table Name	Page No.
1.	Proposed Architecture Diagram	
2.	Registration page	
3.	Login page	
4.	Add houses	
5.	Viewing house details	
6.	Viewing Interested tenants	
7.	Delete house details	
8.	Report of Tenants	
9.	Table of Tenants	
10.	Details of Maintenance	
11.	Report of Custom Roles	
12.	Insertion of custom detail	
13.	Table of upload details	
14.	ER Diagram	
15.	Use Case Diagram	
16.	Database tables	
17.	Main page	
18.	Mysql code	
19.	HTML code	
20.	CSS code	

LIST OF TABLES

Table No.	Table Name	Page No.
1.	Table of login details	
2.	Table of complaint details	

ABBREVIATIONS

AES	Advanced Encryption Standard
CSS	Cascading Style Sheet
PHP	Hypertext Preprocessor
DB	Data Base
SQL	Structured Query Language
SVM	Support Vector Machine
UI	User Interface

1.1 INTRODUCTION

Apartment management refers to the process of overseeing and maintaining the operations of a residential complex, including managing the physical property, ensuring the safety and security of residents, handling financial matters, and addressing residents' concerns and complaints. Effective apartment management is crucial for ensuring that the complex is well-maintained, tenants are satisfied with their living arrangements, and the property is profitable for its owners. Apartment management requires a wide range of skills and expertise, including knowledge of property maintenance, marketing, financial management, customer service, and legal compliance.

In this modern age, apartment management has become increasingly sophisticated with the advent of technology, allowing for more efficient and streamlined management processes. With the help of digital tools and platforms, managers can easily keep track of maintenance requests, financial transactions, and resident communications. Additionally, smart home technologies are increasingly being integrated into residential complexes, allowing for remote monitoring and control of various systems. Overall, effective apartment management is essential for creating a comfortable, safe, and enjoyable living environment for residents, and ensuring the profitability and success of the residential complex.

As mentioned earlier, technology is becoming an increasingly important tool in apartment management. Property management software can help managers keep track of important data, such as tenant information, lease agreements, and maintenance requests. Smart home technologies can also provide benefits for both managers and residents, such as remote monitoring of security systems and temperature control.

Overall, apartment management is a complex and challenging field that requires a combination of skills, knowledge, and experience. With the right approach, apartment managers can create a thriving community that meets the needs of its residents and provides a profitable return for its owners.

1.2 PROBLEM STATEMENT

For the problem statement for apartment management, it is important to understand that managing an apartment complex can be a complex and demanding task. Property managers are responsible for ensuring that the property is well-maintained, that tenants are satisfied with their living conditions, and that the financial aspects of managing the property are handled effectively.

These challenges can be numerous and diverse, including maintaining and repairing the property, managing tenants, communication, security, and financial management. Ensuring that the apartment complex is in good condition, handling repair requests from tenants, collecting rent payments, managing move-ins and move-outs, and ensuring that tenants comply with lease agreements are just some of the difficulties property managers encounter. Effective communication among tenants, maintenance staff, and property managers is essential, but it can be challenging to maintain. The safety and security of tenants and the property is another important concern, requiring property managers to ensure that all residents comply with safety regulations.

Finally, managing the finances of an apartment complex can be a significant challenge, involving rent collection, accurate financial records, bill payments, and expense management. By identifying these challenges, property managers can develop strategies and solutions to improve their management practices and enhance the tenant experience.

Overall, the problem statement for apartment management involves identifying and addressing the various challenges that property managers face in managing an apartment complex effectively. By developing strategies and solutions to address these challenges, property managers can enhance the tenant experience, maintain the property, and ensure that the finances of the complex are handled effectively.

1.3. OBJECTIVES

The aims and outcomes that property managers hope to accomplish in successfully managing an apartment complex are referred to as the objectives for apartment management. These goals can change depending on the particular requirements and priorities of the property, but some typical ones include:

1. Maintenance and Repair: Keeping the apartment complex well-maintained and fixed-up is the major goal of apartment management. This includes keeping the complex in good condition, responding quickly to repair requests, and planning routine maintenance jobs to head off issues before they happen.
2. Tenant Satisfaction: Ensuring that renters are happy with their living arrangements is a crucial goal for flat management. This includes handling move-ins and move-outs efficiently and swiftly responding to tenant complaints.
3. Communication: The goal is to guarantee that requests and complaints are swiftly and effectively handled and that renters, maintenance personnel, and property managers are all on the same page.
4. Safety and Security: Ensuring the renters' and the property's safety and security is a crucial goal. This entails keeping exits and entrances secure, keeping an eye on common areas, and acting quickly in case of an emergency or safety risk.
5. Financial Management: Taking care of an apartment complex's finances is also a top priority. To keep the property's finances in good shape, it is important to make sure that rent is collected on time, that accurate financial records are kept, and that bills and costs are managed well.

By achieving these objectives, property managers can enhance the tenant experience, maintain the property effectively, and ensure that the financial aspects of managing the complex are handled appropriately. Effective apartment management can also lead to higher tenant retention rates, positive reviews and referrals, and more desirable property for prospective tenants.

1.4. SCOPE OF PROJECT

The scope of the project for apartment management refers to the specific boundaries and limitations of the project, including what areas of apartment management will be covered and what will not. The scope of the project can vary depending on the specific needs and priorities of the property, but it typically includes the following areas:

- 1. Design and development of apartment management:** The design and development involve creating a software application that can effectively organize and manage various aspects of apartment management. This includes functionalities such as resident information management, rent collection, maintenance management, accounting, and reporting. The software should be user-friendly, efficient, and scalable to meet the needs of a growing number of apartments.
- 2. Database design and implementation:** The apartment management system requires a robust and efficient database to store and manage resident information, financial data, maintenance records, and other relevant information. The database design and implementation involve creating a schema that can store all the necessary information and allow easy retrieval and updating of data.
- 3. User interface design and development:** The user interface design and development of an apartment management system should provide a user-friendly interface for both the apartment staff and residents to interact with the system. The interface should be intuitive and easy to use, allowing users to navigate and perform tasks with ease.
- 4. Integration with third-party platforms and services:** It may need to integrate with various third-party platforms and services, such as online payment gateways, maintenance service providers, and accounting software. APIs and web services can be used to facilitate this integration.
- 5. Testing and quality assurance:** The apartment management system should undergo rigorous testing to ensure that it meets the requirements and is functioning correctly. Testing should involve unit tests, integration tests, system tests, and user acceptance testing to identify and resolve any issues. Quality assurance should also be performed to ensure that the system is efficient, reliable, and secure. Regular maintenance and updates should be done to ensure that the system remains functional and up-to-date.

1.5 LIST OF GENERAL AND UNIQUE SERVICES IN THE DATABASE

General Services:

Rent Collection: The database can assist with managing the rent collection process, which includes recording payments, producing rent bills, and keeping track of unpaid balances.

Management of Tenant Information: The database can store and manage tenant data, including name, contact information, lease terms, dates of move-in and move-out, and payment history.

Maintenance Management: Using the tool, you can schedule and keep track of requests for maintenance and work orders, as well as track repairs and maintenance activities and assign work to maintenance workers.

Accounting and financial management: The database can assist in managing the financial facets of managing rental properties, such as keeping track of expenses, producing financial reports, and predicting revenue.

Unique Services:

Booking of Amenities: The programme enables residents to reserve and book amenities like a pool, gym, or community room.

Tracking arriving and outgoing shipments and informing tenants of deliveries are made possible by the database.

Parking Management: The programme can allocate parking spots to renters, monitor parking spaces, and issue parking permits.

Management of Lease Renewals: The database has the ability to generate new lease agreements and inform tenants about upcoming lease renewals.

Tenant Feedback Management: Using the programme, tenants can offer comments and recommendations, which property managers can then use to enhance facilities and services.

1.6 SOFTWARE REQUIREMENTS SPECIFICATION

A Software Requirements Specification (SRS) for an apartment management system is a document that outlines the functional and non-functional requirements of the software system. The SRS provides a detailed description of the system's features, user requirements, and design constraints. Below are the key components of an SRS for an apartment management system:

Introduction: The introduction should provide an overview of the apartment management system and its objectives. It should also describe the purpose of the SRS document and the target audience.

Scope: The scope section should define the boundaries of the system and describe the features and functionalities that the system will provide.

Functional Requirements: This section should list all the functional requirements of the system, such as vehicle registration, driver's license, road tax, and other related services. Each requirement should be clearly defined, and its priority level should be specified.

Non-functional Requirements: This section should list all the non-functional requirements of the system, such as performance, scalability, reliability, and security. Each requirement should be clearly defined, and its priority level should be specified.

Use Cases: Use cases are scenarios that describe how users will interact with the system to perform specific tasks. This section should provide a detailed description of each use case, including the user roles, system behaviour, and input/output requirements.

2. LITERATURE SURVEY

2.1 Existing System:

- Offers online booking portal for apartments.
- Provides features such as Online Rent Payments, Maintenance Requests, and Financial Management.
- Allows tenants to search for selected owners and selected property..
- It would not allow the admin to view some properties in the portal.

2.2 Comparison of Existing vs Proposed System:

- In terms of features and functionality, the proposed system offers more options for tracking and personalized activities.
- Offers more functions regarding security with OTP authentication etc.,
- Offers more options for tenants like Package tracking, This can help property managers track and manage packages that are delivered to the property, reducing the likelihood of lost or stolen packages.
- Marketing tools such as property listing and advertising features can be included in apartment management software to help property managers attract potential tenants and fill vacancies more efficiently.
- Overall, the proposed system could provide a more engaging and personalized music experience for customers, but it would need to be carefully designed and tested to address potential technical and ethical challenges.

3. SYSTEM ARCHITECTURE AND DESIGN

3.1 Proposed Architecture Diagram:

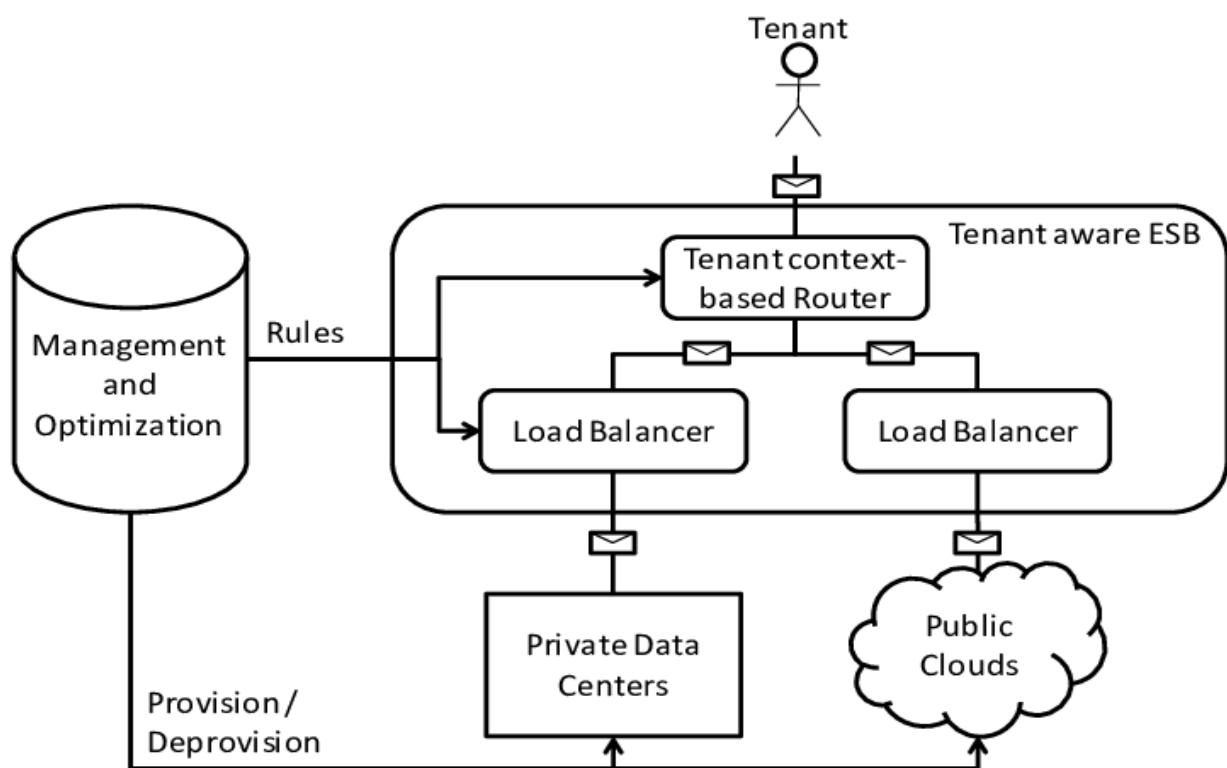


Figure 1: Proposed Architecture Diagram

3.1.1 Front end (UI) design:

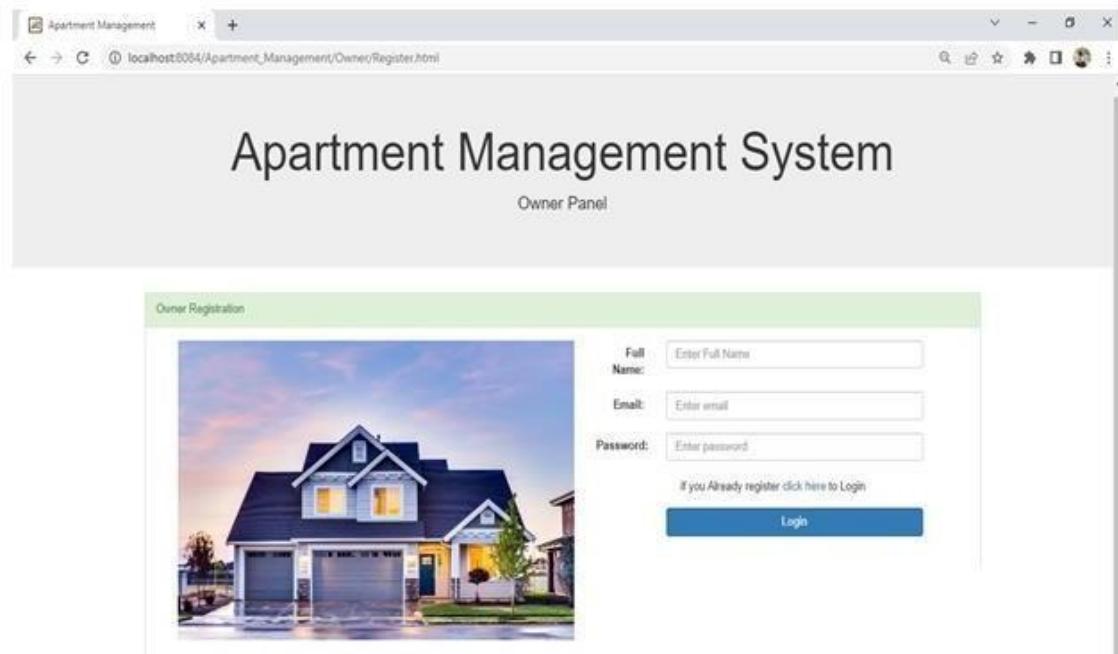


Figure 2: Registration page

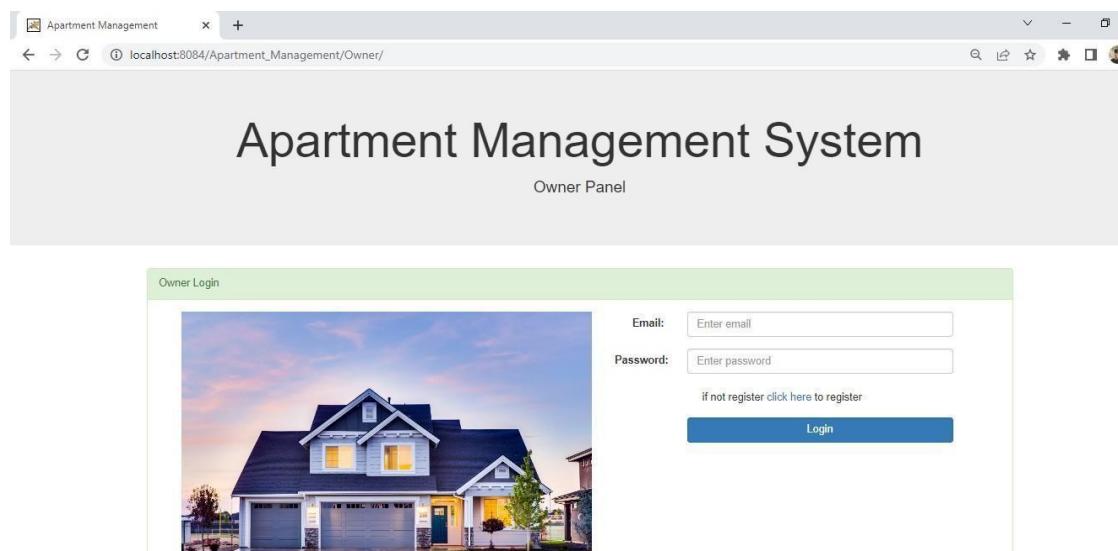


Figure 3: Login page

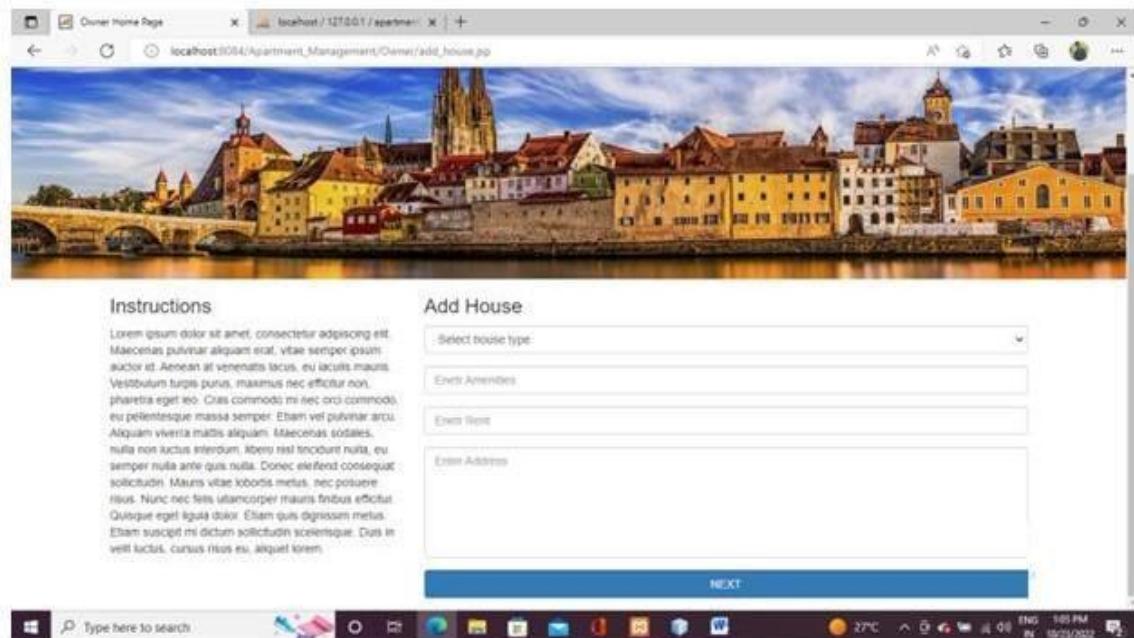


Figure 4: Add houses

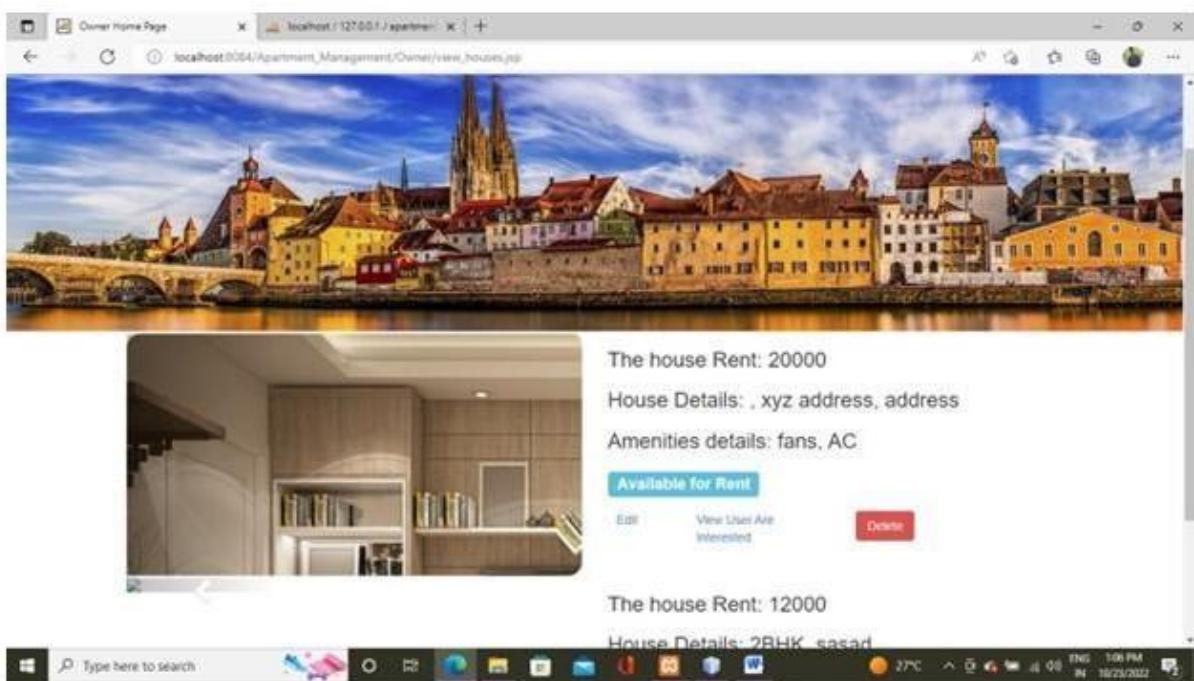


Figure 5: Viewing house details

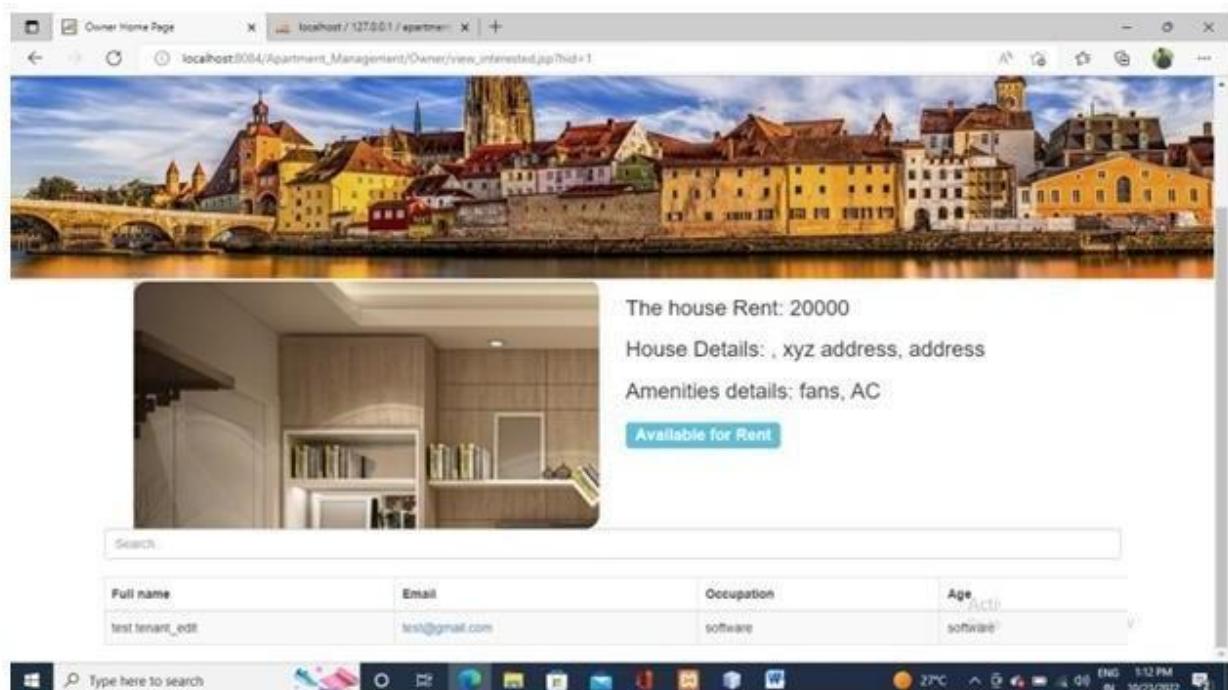


Figure 6: Viewing Interested tenants

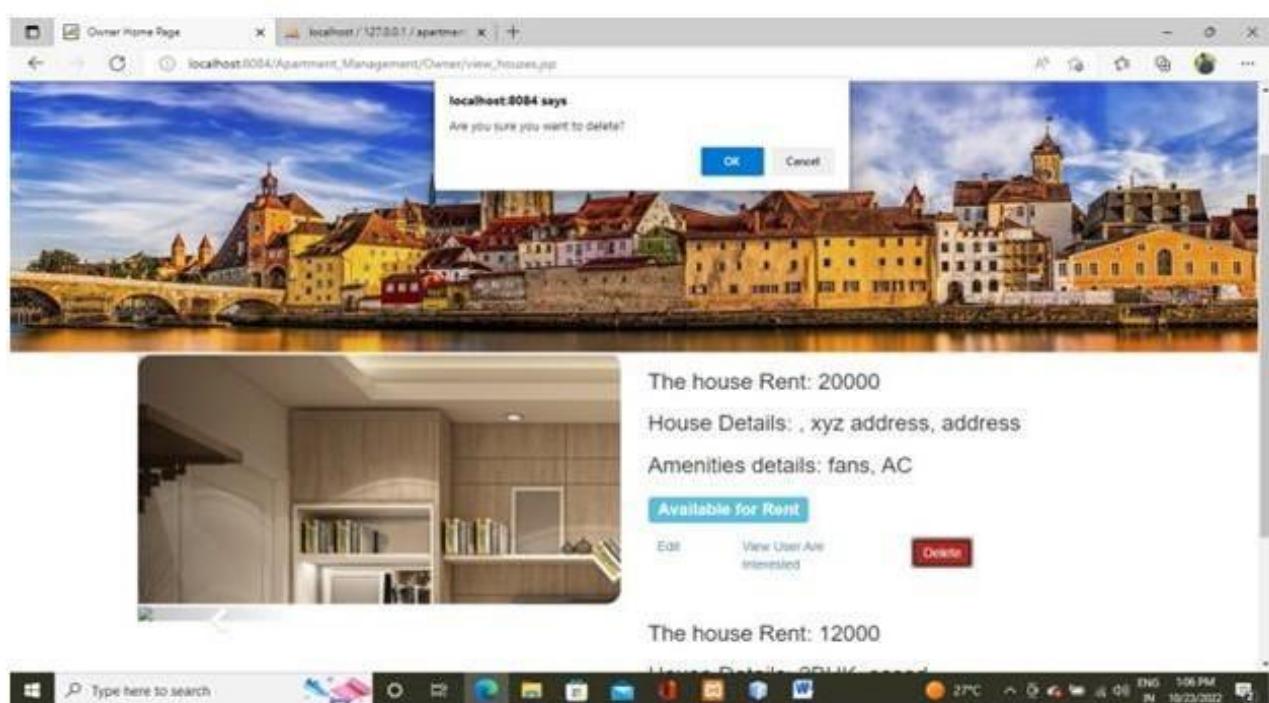


Figure 7: Delete house details

We have used the following front-end languages:

- **Hypertext Mark-up Language (HTML)**

Hypertext Mark-up Language (HTML) is the standard mark-up language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML is used to define the structure of web pages, including text, images, videos, and links. It uses a set of markup tags, also known as elements, to define the structure and content of the page. HTML tags are enclosed in angle brackets and are typically used in pairs, with the opening tag indicating the start of an element and the closing tag indicating the end.

- **Cascading Style Sheets (CSS)**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. CSS is designed primarily to enable the separation of presentation and content, including aspects such as layout, colours, and fonts. CSS includes a wide range of features, such as the layout and positioning of elements, typography, and colour management. It also provides support for responsive design, allowing web pages to adapt to different screen sizes and devices.

3.1.2 Back end (Database) design

The screenshot shows the phpMyAdmin interface for the 'Apartment Mgmt' database. The left sidebar lists various databases and tables. The current table is 'Custom Roles'. The table structure shows two columns: 'Name' and 'Type'. The first row has '1 K. Rakshana' in 'Name' and 'int(10)' in 'Type'. The second row has '2 Kalpana' in 'Name' and 'int(10)' in 'Type'. Below the table, there are sections for 'Indexes' (warning: 'No index defined!'), 'Create an index on 1 columns', and 'Partitions' (warning: 'No partitioning defined!'). The bottom status bar shows the date and time as 04/05/2023 14:52.

Figure 8: Report of Tenants

The screenshot shows the phpMyAdmin interface for the 'Apartment Mgmt' database. The left sidebar lists various databases and tables. The current table is 'Tenant reports'. The table structure shows four columns: 'Name', 'Type', 'Collation', and 'Attributes'. The rows are: 1 GUNNU JAI RAJ, 2 RAKSHANA KANTHAN, 3 ABNUPAM PALAI, and 4 TARUN. Below the table, there is a section for 'Indexes' showing an index named 'TARUN' on the column 'TARUN' with type 'BTREE'. There is also a warning: 'No partitioning defined!'. The bottom status bar shows the date and time as 04/05/2023 14:37.

Figure 9: Table of Tenants

The screenshot shows the phpMyAdmin interface for a MySQL database named 'Apartment Mgmt'. The left sidebar shows various databases and tables. The main area displays the 'Maintenance' table structure. The table has three columns: 'Name' (Pavi, Rama, Shruthi), 'Type' (int(10)), and 'Attributes' (UNSIGNED ZEROFILL, No, None). The 'Comments' column contains 'Portal', 'Financial', and 'Food' respectively. The 'Extra' column shows icons for Change, Drop, and More. Below the table, there are buttons for Print, Move columns, Normalize, Add, and Go. The 'Indexes' section shows a warning: 'No index defined!'. The 'Partitions' section also shows a warning: 'No partitioning defined!'. The bottom status bar shows system information like weather (33°C, Partly sunny) and date/time (04/05/2023, 14:51).

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Pavi	int(10)		UNSIGNED ZEROFILL	No	None	Portal		
2	Rama	int(10)		UNSIGNED ZEROFILL	No	None	Financial		
3	Shruthi	int(10)		UNSIGNED ZEROFILL	No	None	Food		

Figure 10: Details of Maintenance

The screenshot shows the phpMyAdmin interface for a MySQL database named 'Apartment Mgmt'. The left sidebar shows various databases and tables. The main area displays the 'Custom Roles' table structure. The table has two columns: 'Name' (K. Rakshanna, Kalpana) and 'Attributes' (UNSIGNED ZEROFILL, No, None). The 'Comments' column contains 'Manager' and 'Maid' respectively. The 'Extra' column shows icons for Change, Drop, and More. Below the table, there are buttons for Print, Move columns, Normalize, Add, and Go. The 'Indexes' section shows a warning: 'No index defined!'. The 'Partitions' section also shows a warning: 'No partitioning defined!'. The bottom status bar shows system information like weather (33°C, Partly sunny) and date/time (04/05/2023, 14:52).

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	K. Rakshanna	int(10)		UNSIGNED ZEROFILL	No	None	Manager		
2	Kalpana	int(10)		UNSIGNED ZEROFILL	No	None	Maid		

Figure 11: Report of Custom roles

The screenshot shows the 'Structure' tab of the phpMyAdmin interface. A new table named 'Tenant reports' is being created with the following structure:

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I	Comments
GUNNU JAI RAJ	TEXT	25	None	armSCII8_bin	on update CURRI	<input checked="" type="checkbox"/>	---		Tenant 1
RAKSHANNA KANTI	INT	25	None	armSCII8_bin	on update CURRI	<input type="checkbox"/>	---		Tenant 2
ABINUPAM PALAI	INT	25	None	armSCII8_general	UNSIGNED ZERI	<input type="checkbox"/>	---		Tenant 3
	INT		None						

Table comments: Tenant reports Collation: Storage Engine: InnoDB

PARTITION definition: Partition by: Expression or column list Partitions: Preview SQL Save

Warning: The column width of integer types is ignored in your MySQL version unless defining a TINYINT(1) column.

Information: phpMyAdmin Demo Server: Currently running Git revision RELEASE_5_2_1-2481-gdb5341af7 from the master branch.

Figure 12: Insertion of custom details

The screenshot shows the 'Structure' tab of the phpMyAdmin interface for the 'Uploads' table. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Admin_A	int(10)	UNSIGNED ZEROFILL	No	None	Default			Change Drop More
2	Padma_SK Apartments	int(10)	UNSIGNED ZEROFILL	No	None	Chennai			Change Drop More
3	Tarun_Mastero Apartments	int(10)	UNSIGNED ZEROFILL	No	None	Tambaram			Change Drop More
4	Rama_Zaffer Complex	int(10)	UNSIGNED ZEROFILL	No	None	Mumbai			Change Drop More
5	Vissu_Stanza Living	int(10)	UNSIGNED ZEROFILL	No	None	Vizag			Change Drop More
6	Anupam_Estancia Plots	int(10)	UNSIGNED ZEROFILL	No	None	Srikakulam			Change Drop More

Table structure Relation view

Add 1 column(s) after Anupam_Estancia Plots Go

Indexes

No index defined!

Create an index on 1 columns Go

Partitions

Figure 13: Table of upload details

We have used the following Back-end languages:

- **PHP MY ADMIN**

PhpMyAdmin is used for connecting the database to the Apache server. phpMyAdmin is a free and open-source administration tool for MySQL and MariaDB. As a portable web application written primarily in PHP, it is currently one of the most popular MySQL administration tools, especially for web hosting services. PhpMyAdmin offers a range of features, including support for multiple languages, import and export of data in various formats, visual representation of database relationships, and the ability to create and execute SQL queries directly from the interface. It also includes a set of security features, such as the ability to restrict access to specific databases and tables, and an audit log that records all actions performed by users.

- **MY SQL**

We have used, in this project, MySQL which is an open-source relational database management system. MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr, and YouTube. MySQL uses a structured query language (SQL) for managing data, which allows users to create, modify, and retrieve data in a structured and efficient manner. It also supports transactions and provides various security features, such as encryption and authentication, to ensure the security of data stored in the database.

3.2 ER Diagram

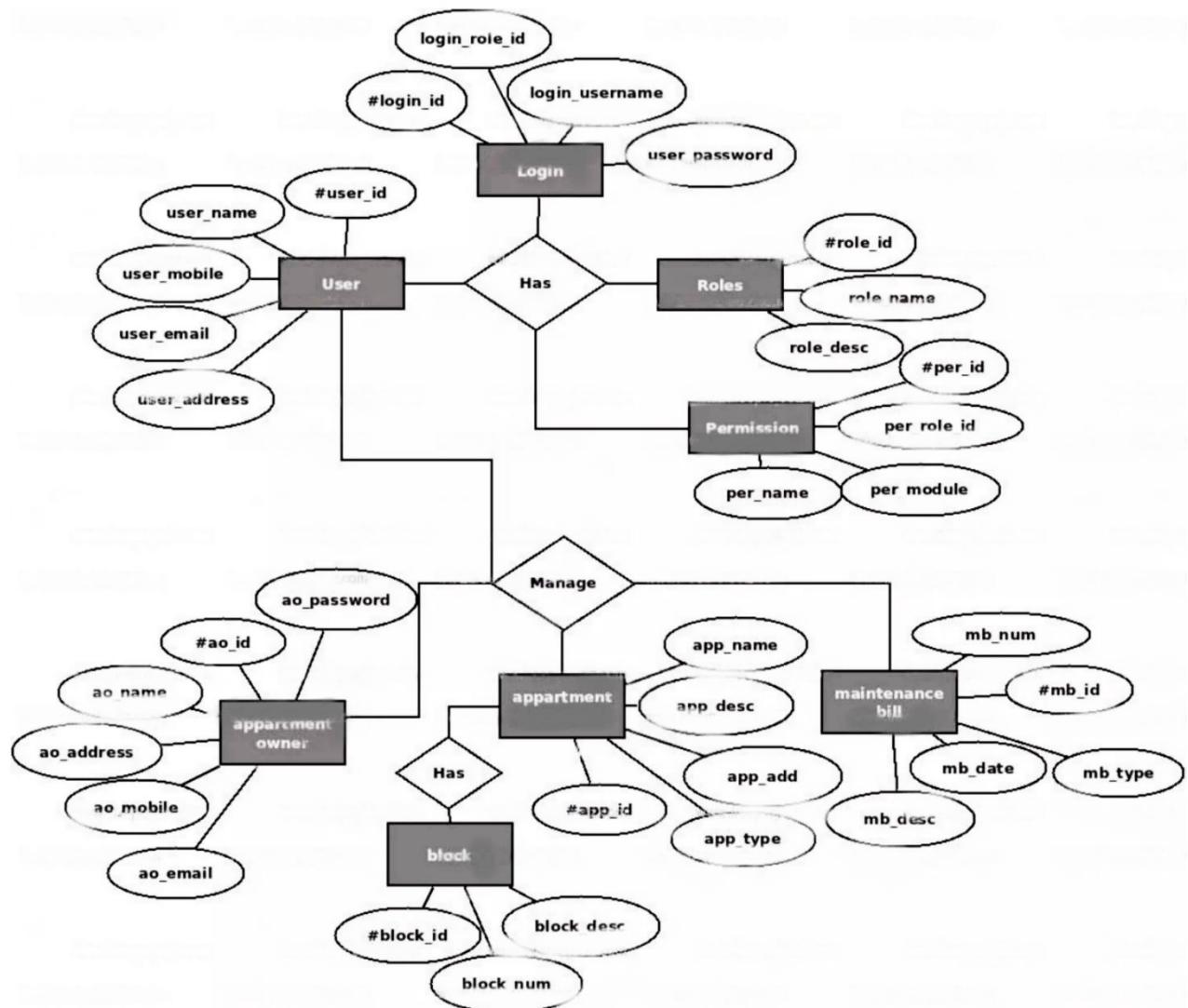


Figure 14: ER Diagram

3.2 Use Case Diagram

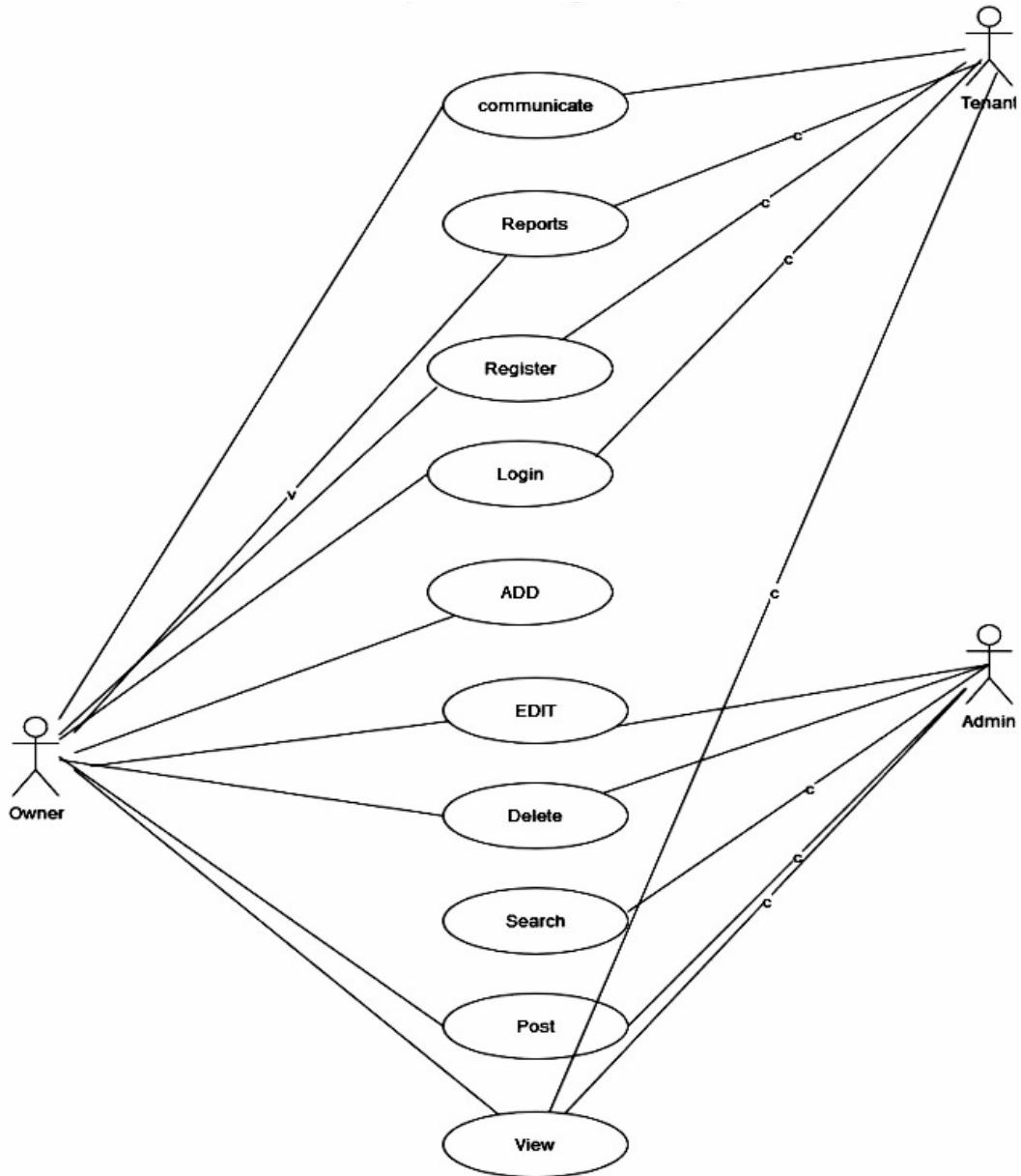


Figure 15: Use Case Diagram

4.1 List of Modules and Functionalities:

Here are the modules of the music management system:-

- **Home Page:** - As soon as the user enters, the first page will be the home page in which there will be three options: user login, user registration and admin login.
- **Sign Up Page:** - The user has to register through the sign-up page by entering his/her details such as Full name, e-mail, phone number etc. and the user has to create his/her own password on this page.
- **Login page:** - This is an authentication page which it checks for the credentials of the users.
- **Admin Login Page:** - On this page admin has to log in with his username and password to further alter the data on the website.
- **Admin Main Page:** - Admin can choose between database and logout to which he can add or delete the details of tenants, or add or delete the complaints.
- **Add Tenant:** - Here, the user can add a tenant after registration..
- **Delete Tenant:** - Here, the user can delete the tenants in the portal at any time and remove them after vacating.
- **Complaints:** - Here, tenants can post complaints regarding maintenance and other activities.
- **Uploads:** - Here, the uploads of tenants will be displayed/arranged.

4.1.1 List of Database Relations:

Table 1 Login: This table contains the details of tenant registration for the plot.

COLUMN NAME	DATA TYPE & SIZE	CONSTRAINTS	DESCRIPTION
Fname	Varchar(30)	Not Null	First name of a user
Lname	Varchar(30)	Not Null	Last name of a user
email	Varchar(10)	Primary key	Email of user
password	Varchar(20)	Not Null	Password set by a user
DOB	Date	Not Null	Date of birth of the user
country	Varchar(10)	Not Null	Country of user
gender	Number	Not Null	Gender of user
User type	Varchar(20)	Not Null	User/admin
Date joined	Date	Not Null	Date of the joining.
Plot cost	Number	Not null	Cost of plot

Table 1: Table of login details

Table 2 Complaints: This table contains the details of maintenance complaints.

COLUMN NAME	DATA TYPE & SIZE	CONSTRAINTS	DESCRIPTION
UserName	Varchar (20)	Not Null	Name of the user
plotID	Int (11)	Primary key	ID of tenant
RoomNo	Varchar (20)	Not Null	Room no. of tenant
Date	Date	Not Null	Dob of complaint
Complaint	Varchar (20)	Not Null	Complaint of tenant

Table 2: Table of complaint details

Normalisation:

- Process for evaluating and correcting table structures to minimise data redundancies
- Reduces data anomalies
- Works through a series of stages called normal forms:
 - i. First normal form (1NF)
 - ii. Second normal form (2NF)
 - iii. Third normal form (3NF)

1st NF:

Database is in first normal form if it satisfies the following conditions

1. Contains only atomic values.
2. There are no repeating groups.

2nd NF:

A relation schema R is in second normal form if every non-primary attribute A is fully functionally Dependant on the Primary key of R

1. It is in 1NF
2. All non-key attributes are fully functionally dependent on the primary key.

3rd NF:

1. It is in 2NF
2. There are no transitive functional dependencies

The above tables satisfy all three normal forms.

4.2 Connectivity used for database access

Table	Action	Rows	Type	Collation	Size	Overhead
Admin Access	<input type="button" value="Browse"/> <input type="button" value="Structure"/> <input type="button" value="Search"/> <input type="button" value="Insert"/> <input type="button" value="Empty"/> <input type="button" value="Drop"/>	0	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
Community	<input type="button" value="Browse"/> <input type="button" value="Structure"/> <input type="button" value="Search"/> <input type="button" value="Insert"/> <input type="button" value="Empty"/> <input type="button" value="Drop"/>	0	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
Complaints	<input type="button" value="Browse"/> <input type="button" value="Structure"/> <input type="button" value="Search"/> <input type="button" value="Insert"/> <input type="button" value="Empty"/> <input type="button" value="Drop"/>	0	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
Custom Roles	<input type="button" value="Browse"/> <input type="button" value="Structure"/> <input type="button" value="Search"/> <input type="button" value="Insert"/> <input type="button" value="Empty"/> <input type="button" value="Drop"/>	0	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
Login Portal	<input type="button" value="Browse"/> <input type="button" value="Structure"/> <input type="button" value="Search"/> <input type="button" value="Insert"/> <input type="button" value="Empty"/> <input type="button" value="Drop"/>	0	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
Maintenance	<input type="button" value="Browse"/> <input type="button" value="Structure"/> <input type="button" value="Search"/> <input type="button" value="Insert"/> <input type="button" value="Empty"/> <input type="button" value="Drop"/>	0	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
Registration	<input type="button" value="Browse"/> <input type="button" value="Structure"/> <input type="button" value="Search"/> <input type="button" value="Insert"/> <input type="button" value="Empty"/> <input type="button" value="Drop"/>	0	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
Tenant reports	<input type="button" value="Browse"/> <input type="button" value="Structure"/> <input type="button" value="Search"/> <input type="button" value="Insert"/> <input type="button" value="Empty"/> <input type="button" value="Drop"/>	0	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-

Figure 16: Database tables

phpMyAdmin is the most popular application for MySQL database management. We can create, update, drop, alter, delete, import, and export MySQL database tables by using this software. phpMyAdmin also supports a wide range of operations like managing databases, relations, tables, columns, indexes, permissions, users, etc., on MySQL and MariaDB. These operations can be performed via the user interface, while we still have the ability to execute any SQL statement.

phpMyAdmin is translated into 72 languages and also supports both RTL and LTR languages so that a wide range of people can easily use this software. We can run MySQL queries, repair, optimise, check tables, and also execute other database management commands. phpMyAdmin can also be used to perform administrative tasks such as database creation, and query execution.

phpMyAdmin is a web-based application used to manage MySQL databases. The type of connectivity used for phpMyAdmin is through the MySQL protocol, which uses the TCP/IP network protocol to establish a connection between the PHP-based web application and the MySQL database server.

When you access phpMyAdmin, it establishes a connection to the MySQL server using the MySQL protocol and allows you to perform various database management tasks, such as creating and modifying tables, running queries, importing and exporting data, and more.

If we don't have access to a web hosting server and want to learn how to use this application to manage the MySQL database locally, we can install PhpMyAdmin on our PC using third-party products, for example, XAMPP, which is the most popular PHP development environment for Windows, Linux and OS X, a completely free, Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be easy to install and use. We also have other means of installation, which we can see in the PhpMyAdmin documentation.

Another third-party tool used to install PhpMyAdmin is the WampServer, which is a Windows-only web development environment that allows us to create web applications with Apache2, PHP and a MySQL database and also installs PhpMyAdmin to manage the MySQL database. In my example, I have installed this tool to access PhpMyAdmin.

5. CODING & TESTING



The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure with files like `main.py`, `Movie.php`, `musicdb.sql`, `Login.html`, `user.html`, `index.html`, and `home.html`.
- Code Editor:** The main editor window contains a Python script (`main.py`) which performs the following tasks:
 - Establishes a MySQL connection to a local host with root privileges.
 - Prepares and executes a SELECT query to find unique movie names.
 - Prepares and executes an INSERT query to add new movie records to the database.
 - Handles cases where a movie already exists or fails to insert.
- Bottom Status Bar:** Displays the Python language icon, terminal tabs, and file navigation icons.
- Bottom Taskbar:** Shows the current workspace path (`C:\xampp\htdocs\Music-Catalogue-Management-System-master`), file status (151 changes), and system information (Windows 10, ENG, 03:10).

Fig 17. Main page

Fig 18. Mysql code

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a "DBMS PROJECT" folder containing "main.py".
- Editor:** Displays the content of "Login.html".
- Output:** Shows a "ModuleNotFoundError: No module named 'mysql'" error.
- Bottom Status Bar:** Shows the file path "C:\Users\kotha\OneDrive\Documents\DBMS project", line 74, column 31, and other system information like temperature (30°C) and date (28-04-2023).

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Login</title>
<link href="https://fonts.googleapis.com/css?family=Roboto|Varela+Round" rel="stylesheet">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
<style type="text/css">
body {
    font-family: 'Varela Round', sans-serif;
}
.modal-login {
    color: #636363;
    width: 350px;
}
.modal-login .modal-content {
    padding: 20px;
    border-radius: 5px;
    border: none;
}
.modal-login .modal-header {
    border-bottom: none;
    position: relative;
    justify-content: center;
}
.modal-login h4 {
    text-align: center;
    font-size: 26px;
    margin: 30px 0 15px;
}
```

Fig 19. HTML code

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a "DBMS PROJECT" folder containing "main.py".
- Editor:** Displays the content of "style.css".
- Output:** Shows a "ModuleNotFoundError: No module named 'mysql'" error.
- Bottom Status Bar:** Shows the file path "C:\Users\kotha\OneDrive\Documents\DBMS project", line 1, column 1, and other system information like temperature (30°C) and date (28-04-2023).

```
h1 {
    font-size: 6em;
    margin: 0;
    animation: pulse 2s ease-in-out infinite;
}

@keyframes pulse {
% {
    transform: scale(1);
}
50% {
    transform: scale(1.2);
}
100% {
    transform: scale(1);
}

p {
    font-size: 1.5em;
    margin: 0;
    margin-top: 20px;
}

a {
    display: inline-block;
    padding: 10px 20px;
    background-color: #4293ef;
    color: #fff;
    text-decoration: none;
    margin-top: 20px;
    font-size: 1.2em;
    border-radius: 50px;
```

Fig 20. CSS code

6. RESULTS & DISCUSSIONS

The use of apartment management systems has become increasingly popular in recent years, as more property managers and landlords seek to streamline their operations and improve the living experience for tenants. Here are some potential results and discussions related to the implementation of apartment management systems:

1. Improved Efficiency: One of the main benefits of using an apartment management system is improved efficiency in managing various aspects of apartment living, such as rent collection, maintenance requests, and communication with tenants. By automating many of these processes, property managers and landlords can save time and reduce the risk of errors.
2. Better Tenant Satisfaction: By providing tenants with online portals for managing their rent payments, submitting maintenance requests, and accessing important documents, apartment management systems can improve tenant satisfaction and reduce the likelihood of tenant turnover.
3. Reduced Maintenance Costs: By enabling property managers to track and respond to maintenance requests in a timely manner, apartment management systems can help reduce maintenance costs over time, as small issues are addressed before they become larger, more expensive problems.
4. Improved Financial Management: Apartment management systems provide features for tracking expenses, generating financial reports, and managing accounts payable and receivable, which can help property managers and landlords stay on top of their finances and make more informed business decisions.
5. Potential Challenges: While apartment management systems can provide numerous benefits, there may also be some challenges associated with their implementation, such as the need for training and support for property managers and tenants, the cost of implementation and ongoing maintenance, and the potential for technical issues.

Overall, the implementation of an apartment management system can lead to improved efficiency, better tenant satisfaction, reduced maintenance costs, and improved financial management, but it is important to consider potential challenges and weigh the costs and benefits before making a decision.

7. CONCLUSION AND FUTURE ENHANCEMENT

In conclusion, managing an apartment complex is a challenging procedure that calls for a variety of knowledge and abilities. Strong communication skills, an awareness of resident requirements and expectations, and resource management expertise are necessary for effective management. Any residential complex's success and sustainability depend on efficient flat management. A well-managed complex provides a safe and comfortable living environment, meets the requirements and expectations of its residents, and is well-maintained.

Additionally, it fosters beneficial interactions between the management staff and the residents, creating a more peaceful living environment. Residential complexes can offer a high-quality living environment that is advantageous to both management and tenants by putting appropriate management practices first.

As apartment management systems continue to evolve and improve, there are several potential future enhancements that could be implemented to further streamline operations and improve the living experience for tenants. Overall, the possibilities for future enhancement of apartment management systems are endless, and the continued evolution of these systems could have a significant impact on the way properties are managed and the living experience of tenants.

REFERENCES

- 1) Roger S.Pressman, Software Engineering. A Practitioner Approach, McGraw Hill, 2005
- 2) Jim Smith Agile Project Management Creating Innovative Products, Pearson 2008
- 3) Walker Royce, Software Project Management, Pearson Education, 1999
- 4) Ian Sommerville, Software Engineering, 8th ed., Pearson Education, 2010
- 5) Rajib Mall, Fundamentals of Software Engineering 4th ed., PHI Learning Private Limited 2014
- 6) Ashfaque Ahmed, Software Project Management A Process-driven Approach, Boca Raton, Florida: CRC Press, 2012
- 7) <http://dspace.christcollegeijk.edu.in:8080/jspui/bitstream/123456789/1094/7/CCASBCA006.pdf>
- 8) <https://creately.com/diagram/example/i1tgqmb4/apartment-rental-system-classic>
- 9) <https://codecanyon.net/search/apartment%20management%20system>