

## CHAPTER 1

### INTRODUCTION

This project is about creating the database about Assets Management System.

Assets apartment management has become important factor in modern society hence the need to have a Assets house management system. The Assets Management System is Searching in Based on the Apartment Paying Guest, Office, House in metropolitan cities. The Assets Management System is Based on the Owners and the Customers. The Owner is updated on the Apartment, Office details, House, Paying Guest details. The Customer is details about the Room space, Room rent and the Address Details also. The Assets Management System is best Suitable the owners because time save and the only contact and the eligible person and there is no need to explain the room details on the speak. The Assets Management System is best application in the city place The customer contact and the easily search and the suitable place of Apartment, Office, PG, House and based the Money, Limit Person is based on the suitable house. The Assets Management System is save the time also. The Assets Management System is used to easily identify the suitable place in Save time, cost also. The Assets Management System is best way to search the house, Apartment office, Paying Guest. Hence this system is best applicable for the above reasons making House Assets an easy process through an online system acknowledgement.

#### 1.1 Background:

Asset management refers to the process of developing, operating, maintaining, and selling assets in a cost-effective manner. Most commonly used in finance, the term is used in reference to individuals or firms that manage assets on behalf of individuals or other entities.

Every company needs to keep track of its assets. That way, its stakeholders will know which assets are available to be employed to provide optimal returns. The assets owned by any business fall into two main categories: fixed and current assets. Fixed or non-current assets refer to assets acquired for long-term use, while current assets are those that can be converted into cash within a short amount of time.

## 1.2 Motivation

This Project is dedicated to:

- Model the existing assets management system
- Provide a Comprehensive set of features to enhance their operational limits
- Evaluate their performance in different Scenarios

Suggest modifications for greater efficiency

## 1.3 Problem Statement and Objectives

The problem statement entitled *“To implement a web application for Assets management system using PhP and MySQL”* is defined with the following objectives

- To produce a web-based system that allow customer to register and reserve houses online And for the company to effectively manage their House Assets business.
- To ease customer’s task whenever they need to rent a house.
- To Transform the manual process of renting house to an online and computerized system
- To validate the house Assets system using user feedback and testimonies
- To produce the documentation such as Software Requirement specification, Software Design Description and Software Development References

## 1.4 Hardware and Software Requirements

This section describes the software and hardware requirements of the system

Web Server	Apache Web Server
DBMS	MySQL
Web Programming language	HTML, PhP, JavaScript
Database Language	SQL
Web Browser	Google Chrome, Microsoft Edge
Software Package	XAMPP
Hard Disk Space	Minimum 1GB with 512 MB of RAM
Operating System	Windows, Ubuntu

## **1.5 Organization of the report**

In chapter 1, we have dealt with motivation, problem statement, objectives, software and hardware requirements used in Assets Management System. In chapter 2, discusses the database schema consisting of ER Diagram and the list of database tables used. In chapter 3 and 4, demonstrates the working of the project. In chapter 5, gives the conclusion.

## CHAPTER 2

# DATABASE DESIGN

This chapter includes details of database conceptual modelling, ER diagrams, and database design concepts.

### 2.1 Entity-Relationship (ER) diagram

An entity-relationship diagram (ERD) is an abstract and conceptual representation of data. ER modelling is a database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

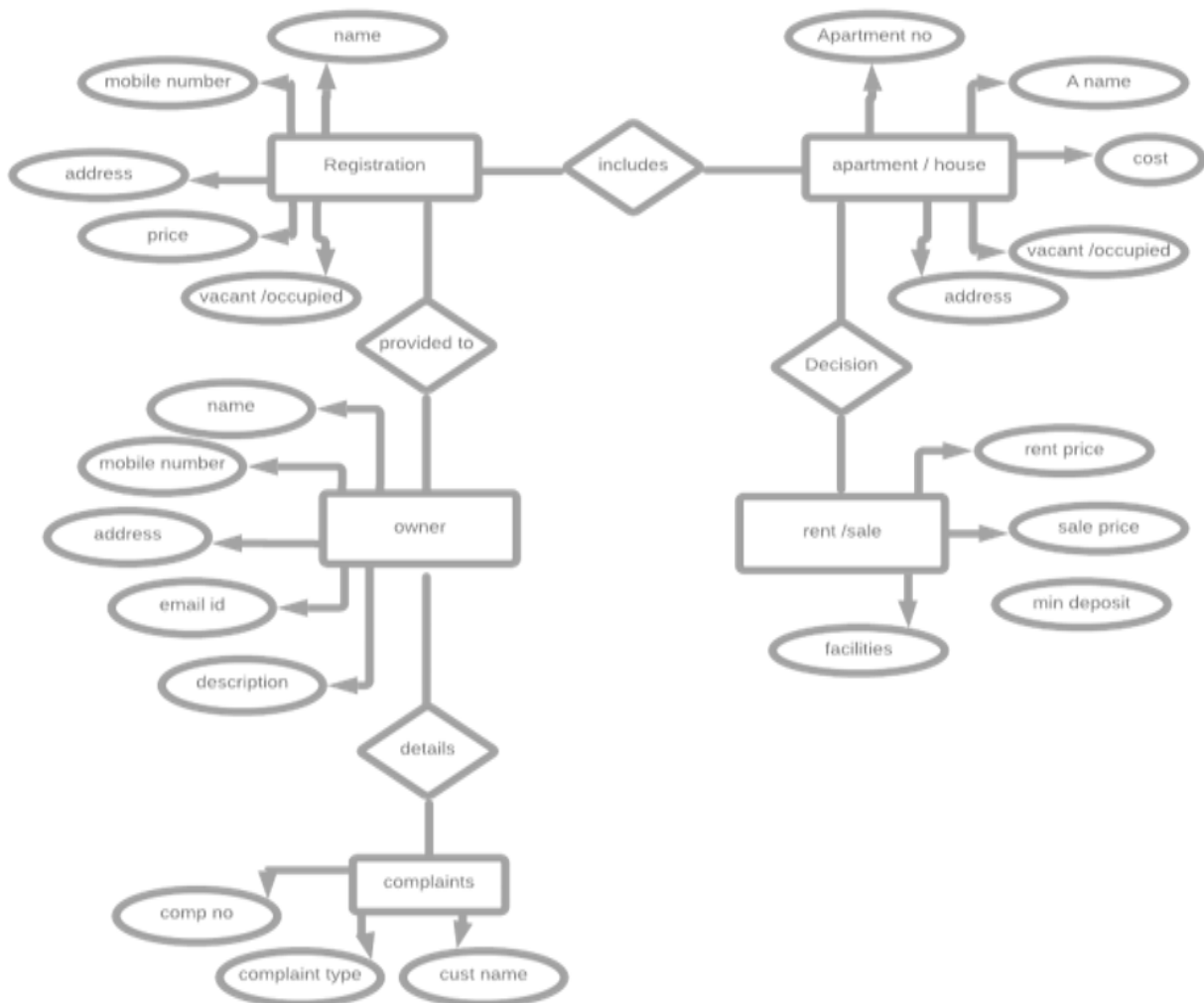


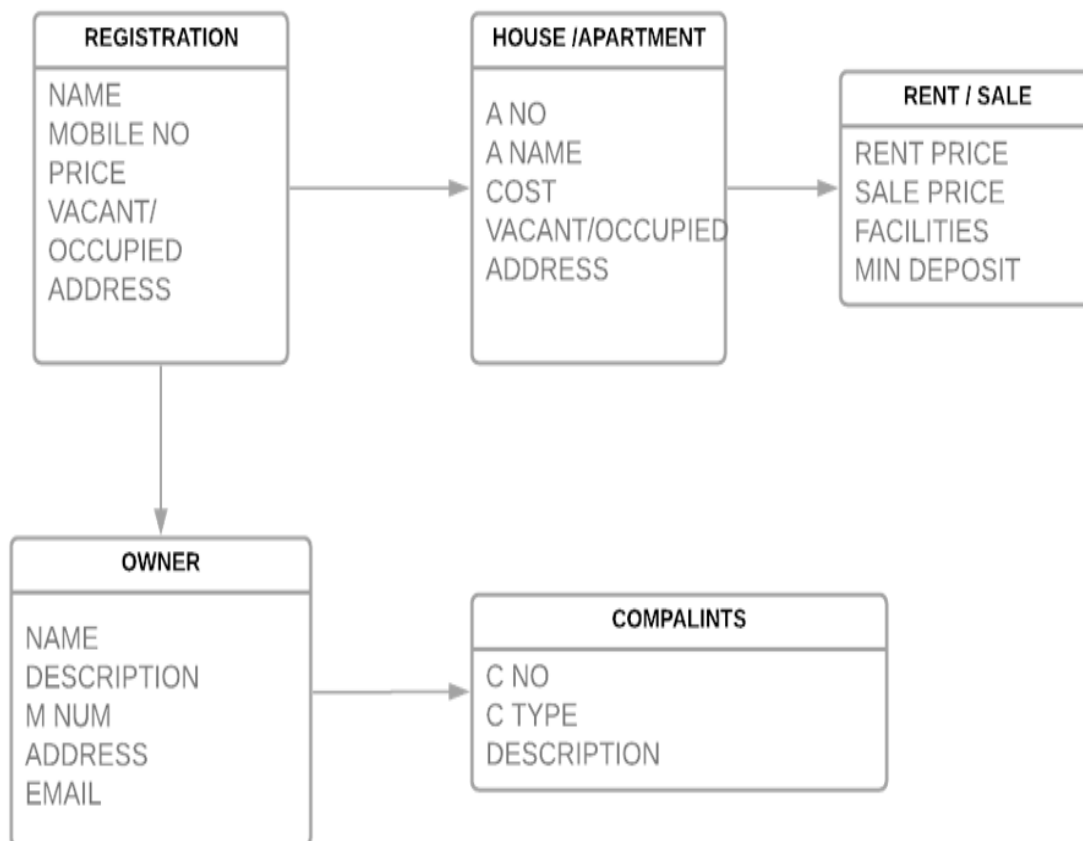
Figure 2.1 ER diagram

## 2.2 Logical Schema

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful.

Logical Database Schema – This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.



## 2.3 Database Tables and Metadata

Table name:	Registration
Description:	It gives details about registration of customer
Attributes:	id name mobilenumber address price vacant_occupied
Primary key:	id

**Table 5.2.1: Registration**

Table name:	Owner
Description:	It stores the details of owner of apartment
Attributes:	id name mobilenumber address email description
Primary key:	id

**Table 5.2.2: Owner**

Table name:	Complaints
Description:	It stores the details of complaints of customer
Attributes:	id complaint_type customer_name
Primary key:	id

**Table 5.2.3: Complaints**

Table name:	Apartment
Description:	It stores the details of apartment
Attributes:	Id name cost address vacant_occupied
Primary key:	id

**Table 5.2.4: Apartment**

Table name:	Rent_sale
Description:	It stores the details of rent and sale of apartment
Attributes:	id rent_price sell_price minimum_deposit facilities
Primary key:	id

**Table 5.2.7: Rent\_sale**

## 2.3 Summary

In the present chapter, the overall database modelling and design concepts are discussed.

## CHAPTER 3

# IMPLEMENTATION

This chapter includes implementation details of application with tools and techniques.

### 3.1 Introduction to DBMS

A database is an organized collection of data. A relational database, more restrictively, is a collection of schemas, tables, queries, reports, views, and other elements. Database designers typically organize the data to model aspects of reality in a way that supports processes requiring information, such as (for example) modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

A database-management system (DBMS) is a computer-software application that interacts with end-users, other applications, and the database itself to capture and analyse data. A general-purpose DBMS allows the definition, creation, querying, update, and administration of databases. Well-known DBMSs include MySQL, PostgreSQL, Enterprise DB, MongoDB, MariaDB, Microsoft SQL Server, Oracle, Sybase, SAP HANA, MySQL, SQLite and IBM.

A database is not generally portable across different DBMSs, but different DBMSs can interoperate by using standards such as SQL and ODBC or JDBC to allow a single application to work with more than one DBMS. Sometimes a DBMS is loosely referred to as a "database".

#### 3.1.1 Advantages of a DBMS

Using a DBMS to store and manage data comes with advantages, but also overhead. One of the biggest advantages of using a DBMS is that it lets end users and application programmers' access and use the same data while managing data integrity. Data is better protected and maintained when it can be shared using a DBMS instead of creating new iterations of the same data stored in new files for every new application. The DBMS provides a central store of data that can be accessed by multiple users in a controlled manner.

**Central storage and management of data within the DBMS provides:**

- Data abstraction and independence
- Data security
- A locking mechanism for concurrent access
- An efficient handler to balance the needs of multiple applications using the same data



- The ability to swiftly recover from crashes and errors, including restorability
- Robust data integrity capabilities
- Logging and auditing of activity
- Simple access using a standard application programming interface (API)
- Uniform administration procedures for data.

Another advantage of a DBMS is that it can be used to impose a logical, structured organization on the data. A DBMS delivers economy of scale for processing large amounts of data because it is optimized for such operations. DBMS can also provide many views of a single database schema. A view defines what data the user sees and how that user sees the data. The DBMS provides a level of abstraction between the conceptual schema that defines the logical structure of the database and the physical schema that describes the files, indexes and other physical mechanisms used by the database. When a DBMS is used, systems can be modified much more easily when business requirements change. New categories of data can be added to the database without disrupting the existing system and applications can be insulated from how data is structured and stored. Of course, a DBMS must perform additional work to provide these advantages, thereby bringing with it the overhead. A DBMS will use more memory and CPU than a simple file storage system.

### **3.2 Introduction to SQL**

Structure Query Language (SQL) is a programming language to request data from a database, to add, update, or remove data within a database, or to manipulate the metadata of the database. SQL was the first commercial language introduced for E.F Codd's Relational model. Today almost all RDBMS (MySQL, Oracle, Informix, Sybase, MS Access) uses SQL as the standard database language. SQL is used to perform all type of data operations in RDBMS.

**Commonly used statements are grouped into the following categories:**

#### **Data Query Language (DQL)**

- SELECT - Used to retrieve certain records from one or more tables.

#### **Data Manipulation Language (DML)**

- INSERT - Used to create a record.
- UPDATE - Used to change certain records.
- DELETE - Used to delete certain records.

**Data Definition Language (DDL)**

- CREATE - Used to create a new table, a view of a table, or other object in database.
- ALTER - Used to modify an existing database object, such as a table.
- DROP - Used to delete an entire table, a view of a table or other object in the database.

**Data Control Language (DCL)**

- GRANT - Used to give a privilege to someone.
- REVOKE - Used to take back privileges granted to someone.

**3.3 Introduction to MYSQL**

MySQL is a leading open-source database management system. It is a multi-user, multithreaded database management system. MySQL is especially popular on the web. It is one of the parts of the very popular LAMP platform. Linux, Apache, MySQL and PHP. MySQL database is available on most important OS platforms. It runs on BSD Uix, Linux, Windows or Mac. Wikipedia, YouTube, Facebook use MySQL. These sites manage millions of queries each day. MySQL comes in two versions: MySQL server system and MySQL embedded system. The MySQL server software and the client libraries are dual licensed: GPL version 2 and proprietary license. The development of MySQL began in 1994 by a Swedish company MySQL AB. Sun Microsystems acquired MySQL AB in 2008. Sun was bought by Oracle in 2010. MySQL, PostgreSQL, Firebird, SQLite, Derby, and HSQLDB are the most well-known open-source database systems. MySQL is developed in C/C++. Except for C/C++, APIs exist for PHP, Python, Java, C#, Eiffel, Ruby.

**3.4 PhpMyAdmin**

phpMyAdmin is a web-based interface to a MySQL server. The interface is written in the PHP language, which is frequently used for web-based interactive programs. phpMyAdmin translates what you enter into the web browser, sends queries to the MySQL database, and translates the returned information back to web format. As we will see, there are other ways to access the MySQL database, including directly from Mathematics.

### 3.5 Triggers

A trigger is a named database object that is associated with a table, and that activates when a particular event occurs for the table. Some uses for triggers are to perform checks of values to be inserted into a table or to perform calculations on values involved in an update. A trigger is defined to activate when a statement inserts, updates, or deletes rows in the associated table. These row operations are trigger events. For example, rows can be inserted by INSERT or LOAD DATA statements, and an insert trigger activates for each inserted row.

### 3.6 Stored procedure

A procedure (often called a stored procedure) is a subroutine like a subprogram in a regular computing language, stored in database. A procedure has a name, a parameter list, and SQL statement(s). All most all relational database system supports stored procedure, MySQL 5 introduce stored procedure. In the following sections we have discussed MySQL procedure in details and used MySQL 5.6 under Windows 7. MySQL 5.6 supports "routines" and there are two kinds of routines stored procedures which you call, or functions whose return values you use in other SQL statements the same way that you use pre-installed MySQL functions like pi( ).

### Stored procedure and trigger used in the above application

```
CREATE DEFINER=`root`@`localhost`  
PROCEDURE `GetUsers` (IN `aid` INT)  
SELECT * FROM users WHERE id=aid$$
```

```
CREATE TRIGGER `insertiLogs`  
AFTER INSERT ON `users`  
FOR EACH ROW INSERT INTO logs VALUES(null, NEW.Fullname, NEW.mobile, NOW())
```

# TESTING

## Introduction to Testing

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

## Stages in the Implementation of Testing

### ➤ Unit Testing

During this first round of testing, the program is submitted to assessments that focus on specific Units or components of the software to determine whether each one is fully functional. The main aim of this endeavour is to determine whether the application functions as designed. In this phase, a unit can refer to a function, individual program or even a procedure, and a White-box Testing method is usually used to get the job done. One of the biggest benefits of this testing phase is that it can be run every time a piece of code is changed, allowing issues to be resolved as quickly as possible. It's quite common for software developers to perform unit tests before delivering software to testers for formal testing

### ➤ Integration Testing

Integration testing allows individuals the opportunity to combine all of the units within a program and test them as a group. This testing level is designed to find interface defects between the modules/functions. This is particularly beneficial because it determines how efficiently the units are running together. Keep in mind that no matter how efficiently each unit is running, if they aren't properly integrated, it will affect the functionality of the software program. In order to run these types of tests, individuals can make use of various testing methods, but the specific method that will be used to get the job done will depend greatly on the way in which the units are defined.

### ➤ **System Testing**

System testing is the first level in which the complete application is tested as a whole. The goal at this level is to evaluate whether the system has complied with all of the outlined requirements and to see that it meets Quality Standards. System testing is undertaken by independent testers who haven't played a role in developing the program. This testing is performed in an environment that closely mirrors production. System Testing is very important because it verifies that the application meets the technical, functional, and business requirements that were set by the customer.

### ➤ **Acceptance Testing**

The final level, Acceptance testing (or User Acceptance Testing), is conducted to determine whether the system is ready for release. During the Software development life cycle, requirements changes can sometimes be misinterpreted in a fashion that does not meet the intended needs of the users. During this final phase, the user will test the system to find out whether the application meets their business" needs. Once this process has been completed and the software has passed, the program will then be delivered to production.

## **Results**

Several errors were detected and rectified and the whole project is working as it should with proper output and high efficiency. The several tests performed are as follows:

<b>Test case id</b>	<b>Test case</b>	<b>Input data</b>	<b>Steps to execute the test case</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Pass/Fail</b>
1	Login screen	Wrong username or password.	After entering the data click the login button.	A proper message indicating the error should appear and the user should be redirected to login screen.	A message was displayed saying incorrect username or password.	Pass

2	Insertion	If field not entered	After entering the data, click the insert button	A proper message indicating the error should appear and the user should be redirected	A message pointing to the respected field showing that this field is required.	Pass
3	Deletion	-	After clicking the delete button.	Particular row is deleted.	Row is deleted.	Pass
4	Edit	If field not entered	After clicking the edit button.	A proper message indicating the error should appear and the user should be redirected to update screen.	A message pointing to the respected field showing that this field is required.	Pass

## CHAPTER 4:

### SNAPSHOTS

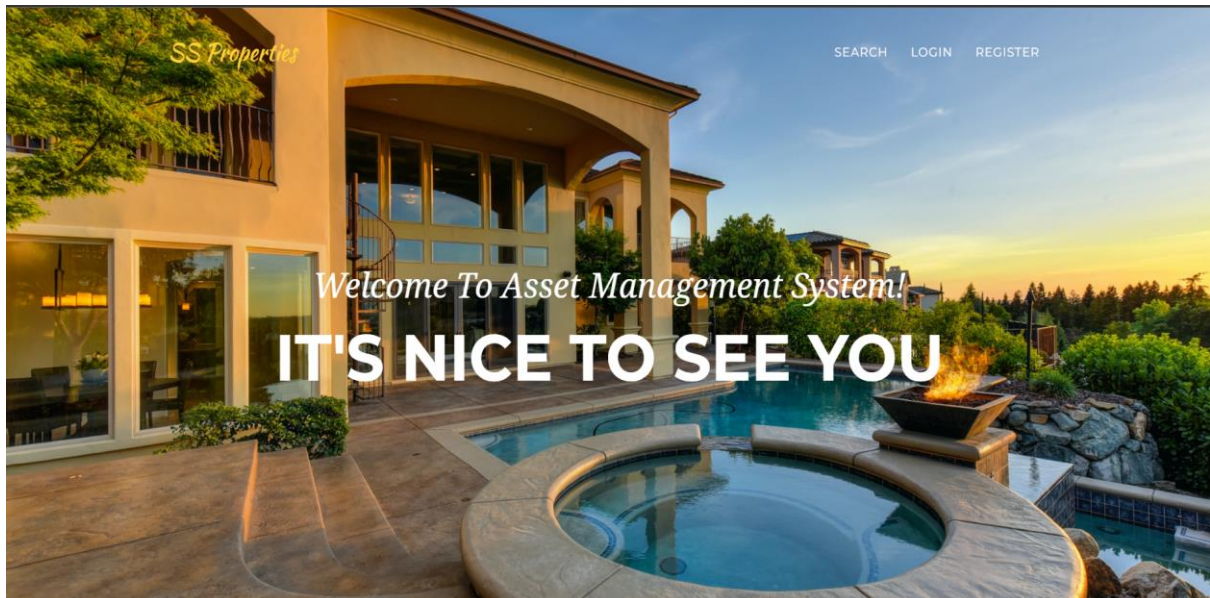


Figure 4.1: On clicking front screen directed to login screen.

---

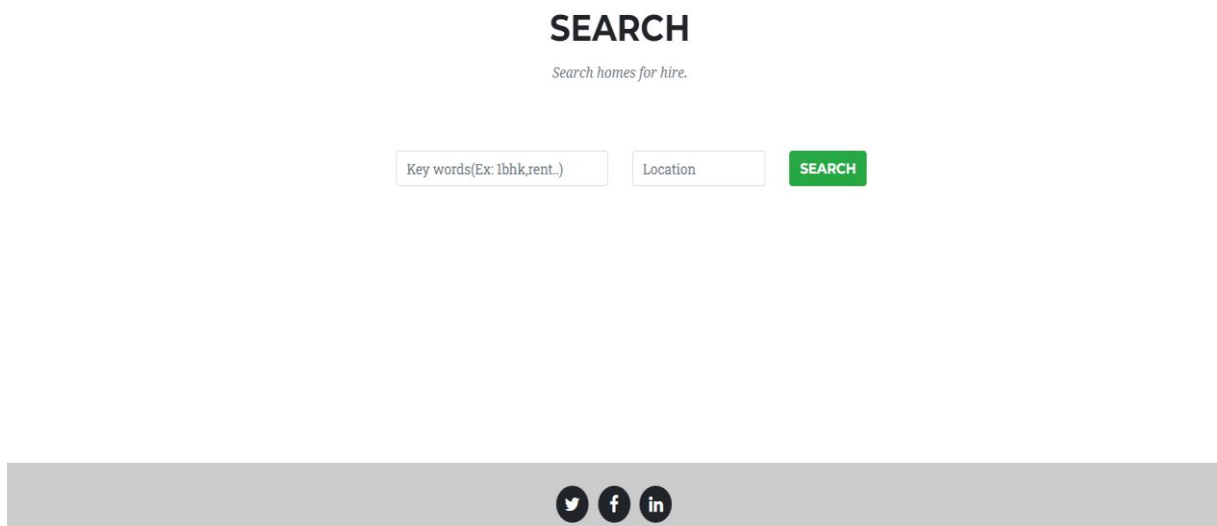


Figure 4.2: clicking on search, The customer can search a house.

*Logo/Home* LOGIN

### Register

Full Name	User Name
<input type="text" value="CHANDANA"/>	<input type="text" value="CHANDU"/>
Mobile	Email
<input type="text" value="9876543210"/>	<input type="text" value="CHANDU2000@GMAIL.COM"/>
Password	
<input type="password" value="*****"/>	
Confirm Password	
<input type="password" value="*****"/>	
<input type="button" value="Submit"/>	

**Figure 4.3: If you want to sell your property you should have a user id if not you can register.**

*Logo/Home* REGISTER

### Login

Email Address/User Name
<input type="text" value="CHANDU"/>
Password
<input type="password" value="***"/>
<input type="button" value="Submit"/>

**Figure 4.4: After the registration, you can login through id & password.**



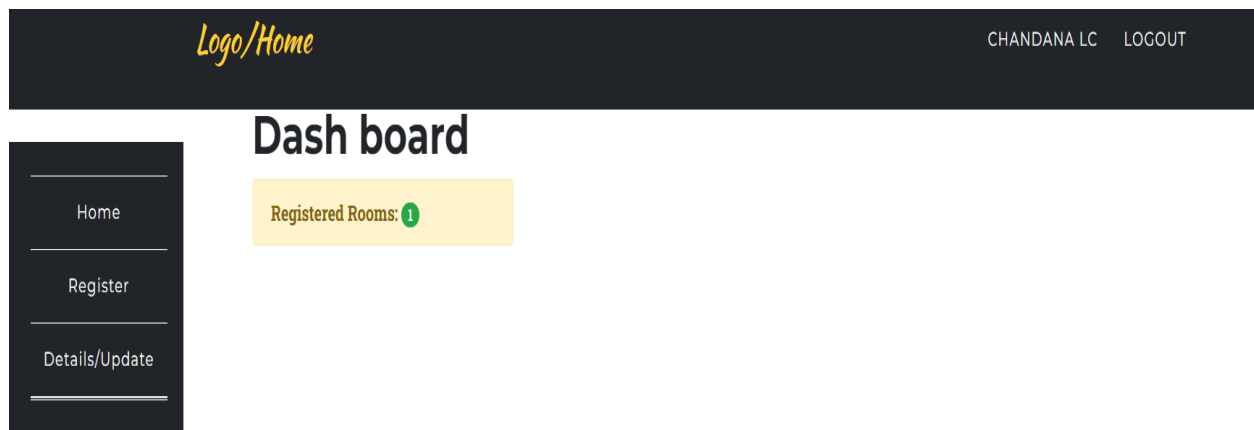


Figure 4.5: It will show your profile & how many properties are for sale.

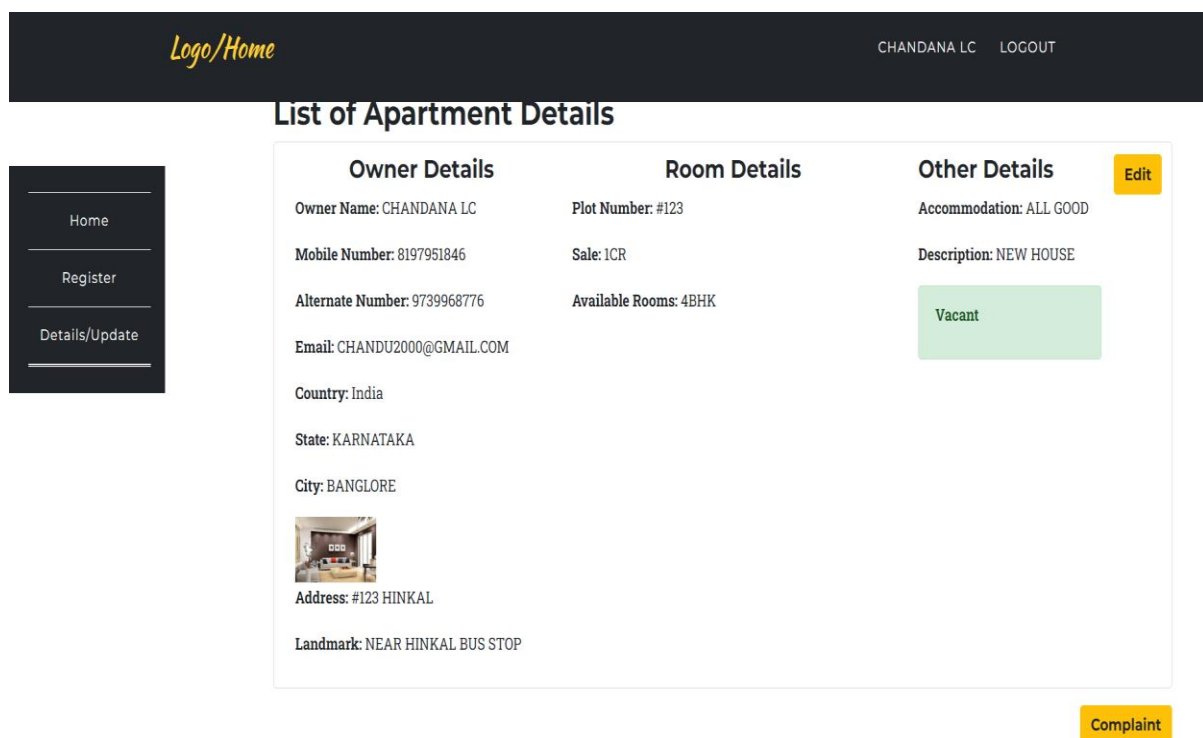


Figure 4.6: By clicking on registered rooms you will interface with this page, This page shows about the property details that you have kept for sale.

The screenshot shows the 'Property Registration' form. At the top, there is a dark header with 'Logo/Home' on the left and 'CHANDANA LC' and 'LOGOUT' on the right. Below the header, there are two tabs: 'Individual Home Registration' and 'Property Registration' (which is active). On the left side of the form, there is a vertical sidebar with buttons: 'Home', 'Register', and 'Details/Update'. The main form area is titled 'Registration' and contains several input fields arranged in a grid. The fields are: Full Name, Mobile (10 digit mobile number), Alternat Mobile (10 digit mobile number), Email, Plot Number/Home Number (with a placeholder 'Please fill out this field: Number'), Available Rooms (1BHK/2BHK/3BHK/1RK), Country, State, City, Rent (Rent), Sale (Sale), Deposit (Deposit), Facilities (Facilities), Description (Description), Landmark (landmark), Address (Address), Vacant/Occupied (Vacant), and Image (Choose File, No file chosen). A 'Submit' button is at the bottom left of the form.

Registration		
Full Name	Mobile	Alternat Mobile
<input type="text" value="Full Name"/>	<input type="text" value="10 digit mobile number"/>	<input type="text" value="10 digit mobile number"/>
Email	Plot Number/Home Number	Available Rooms
<input type="text" value="Email"/>	<input type="text" value="Please fill out this field: Number"/>	<input type="text" value="1BHK/2BHK/3BHK/1RK"/>
Country	State	City
<input type="text" value="Country"/>	<input type="text" value="State"/>	<input type="text" value="City"/>
Rent	Sale	Deposit
<input type="text" value="Rent"/>	<input type="text" value="Sale"/>	<input type="text" value="Deposit"/>
Description	Landmark	Facilities
<input type="text" value="Description"/>	<input type="text" value="landmark"/>	<input type="text" value="Facilities"/>
Vacant/Occupied	Image	Address
<input type="text" value="Vacant"/>	<input type="button" value="Choose File"/> No file chosen	<input type="text" value="Address"/>
<input type="button" value="Submit"/>		

**Figure 4.7:** By clicking on register This page will occur, where you can registration a new property for sale.

The screenshot shows the 'Dash board' for an admin user. At the top, there is a dark header with 'Logo/Home' on the left and '(ADMIN)' and 'LOGOUT' on the right. Below the header, there is a vertical sidebar with buttons: 'Home', 'Register', 'Details/Update', 'Send SMS', and 'Complaint List'. The main dashboard area has a title 'Dash board' and two yellow boxes displaying statistics: 'Registered Users: 8' and 'Rooms for Rent: 6'.

Dash board	
Registered Users: 8	Rooms for Rent: 6

**Figure 4.8:** This page is for admin only admin can access this, here you can see the users details and how many rooms are for rent.

*Logo/Home*(ADMIN) LOGOUT

Home

Register

Details/Update

Send SMS

Complaint List

## List Of Usres

#	Full Name	Email	Username	Role
0	N N Naveen	nnn@gmail.com	nnn	user
1	sayeed	Sayed12@fmail.co	sayed11	user
2	suhasini mam	abc123@gmail.com	mam	user
3	CHANDANA LC	CHANDU2000@GMAIL.COM	CHANDU	user

**Figure 4.9:** hear you can see the list of user's details.

## **CHAPTER 5**

### **CONCLUSION**

Benefits from asset management strategy appliance are obvious. Also having a good asset management system is one of the most important aspects. In this paper we tried to make an overview of desired features and technical preconditions. Also, we tried to value our desirability of such functions and value some of the existing software packages against our methodology. General conclusion is that some of the systems managed to gain high scores in some of the sections excelling in some of the features but failed in the rest of the metrics. Having in mind the need of such systems gives us a motivation for its development.

## BIBLIOGRAPHY

### Books:

- [1] Fundamental of DATABASE SYSTEMS – Ramez Elmasri, Shamkanth B Navathe, 7th Edition, Pearson Education, 2017.
- [2] The Complete Reference JAVA – Herbert Schildt, 7th Edition, Mc Graw Hill Education.
- [3] Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill.
- [4] Randy Connolly, Ricardo Hoar, "Fundamentals of Web Development", 1<sup>st</sup> Edition, Pearson Education India.
- [5] Robin Nixon, "Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5", 4<sup>th</sup> Edition, O'Reilly Publications, 2015.
- [6] Luke Welling, Laura Thomson, "PHP and MySQL Web Development", 5<sup>th</sup> Edition, Pearson Education, 2016.
- [7] David Sawyer McFarland, "JavaScript & jQuery: The Missing Manual", 1<sup>st</sup> Edition, O'Reilly/Shroff Publishers & Distributors Pvt Ltd, 2014.

### Websites & links:

- <http://www.wikipedia.org>
- <http://www.projectsclub.com>
- <http://www.google.com>
- <http://www.stackoverflow.com>
- <http://www.tutorialpoint.com>
- <http://localhost/phpmyadmin/>