1.INTRODUCTION

1.1 PROJECT PURPOSE

This website is a Online real estate business website through which a user can access its information and manage all the adding, updating, deleting the assets and some of its tasks. The Admin user can change the update the information regarding property selling and buying and cancellation. The system is very useful for the companies who developed apartments, hotels, villa, residential properties and commercial properties. Companies or individual agents can also advertise their property.

1.2 SCOPE OF THE PROJECT

The real of World Wide Web have spread across millions of household, so naturally, Internet has become by far the best platform for real estate marketing today. Now days when everything is online, how is it possible that real estate left web application behind. There are lots of real estate companies who advertise their property online so idea behind developing this application is that their property can also sell, or buy rental property using this. These applications are not widely popular but in future, they have large scope of growth. This website is a online real estate management through which individual agents or buyer can maintain their property document keeping and managing property registration and also access its information and manage all the adding, updating, deleting the as and some of its tasks. The Admin user can inform their agents for regarding to property and update the information regarding property and cancellation of property or changing buyer choice. The system is very useful for the companies or builders that can post and edit their properties and their personal info and admin can monitor records of all of them. The system is also useful which also keeps track of Account details of buyers and Investors and also RES Industry.

1.3. MODULES IN THE SOFTWARE

Admin Module:

1. Dashboard: In this section, admin can see all detail in brief like total property type, total country, total state, total city, total agent, total owner, total buyer(user) ant total property listed.
2. Property Type: In this section, admin can manage property type (add/update).
3. Country: In this section, admin can manage country (add/update).
4. State: In this section, admin can manage state (add/update).
5. City: In this section, admin can manage city (add/update).
6. Owner: In this section, admin can view the detail of owners.
7. Agents: In this section, admin can view the detail of agents.
8. User: In this section, admin can view the detail of user.
9. List of properties: In this section, admin can view details of property listed.
10. Reviews: In this section admin, can view reviews and also approved, disapproved and delete the reviews which is given by users.
11. Pages: In this section, admin can manage about us and contact us pages.
12. Search Property: In this section admin, can search the listed property by its property id, name and mobile number.

Admin can also update his profile, change the password and recover the password.

User Module

1. Home Page: Owner can view the home page of real estate management system.
2. About: Owner can view about us page.
3. Properties: Owner can view own and other owner properties they can view property type wise, Status wise and city wise.
4. Contact us: Owner can view contact us page.
5. My Account: In this there is three section:

->User Profile

In this section owner do the following activity

1. Edit Profile: Owner can edit his/her own profile.
2. Change Password: Owner can change his/her own password.
3. Add Property: Owner can add his/her own property.
4. My Properties: Owner can see his/her own listed properties.
5. Received Enquiries: Owner can view receive enquiries against his/her own listed properties and also answer the enquiries.

- vi. Answer Enquiries: Owner can views answer enquiries.
- vii. Logout: Owner can logout from own account.

-> Change Password: Owner can change his/her own password.

-> Logout: Owner can logout from own account.

Agents Module

1. Home Page: Agents can view the home page of real estate management system.
2. About: Agents can view about us page.
3. Properties: Agents can view own and other owner properties they can view property type wise, Status wise and city wise.
4. Contact us: Agents can view contact us page.
5. My Account: In this there is three section:

->Agent Profile

In this section Agents do the following activity

a) Edit Profile: Agents can edit his/her own profile.
b) Change Password: Agents can change his/her own password.
c) Add Property: Agents can add his/her own property.
d) My Properties: Agents can see his/her own listed properties.
e) Received Enquiries: Agents can view receive enquiries against his/her own listed properties and also answer the enquiries.
f) Answer Enquiries: Agents can views answer enquiries.
g) Logout: Agents can logout from own account.

->Change Password: Agents can change his/her own password.

->Logout: Agents can logout from own account.

2. SYSTEM STUDY

2.1 EXISTING SYSTEM

The existing system which is followed at present consists of the poor data quality and sloppy data collection due to manual paper-based system and difficulty in maintaining hard files.

Disadvantages of existing system

- Manual entries of the records become lengthy, complex, and difficult to manage.
- Increase in the cost.
- There is always a security risk.

In order to overcome these complexities, the present system is introduced to overcome the faulty techniques.

2.2 FEASIBILITY STUDY

A feasibility study is an analysis of how successfully a project can be completed, accounting for factors that affect it such as economic, technological, legal and scheduling factors. Project managers use feasibility studies to determine potential positive and negative outcomes of a project before investing a considerable amount of time and money into it. A feasibility study tests the viability of an idea, a project or even a new business. The goal of a feasibility study is to place emphasis on potential problems that could occur if a project is pursued and determines if, after all significant factors are considered, the project should be pursued. Feasibility studies also allow a business to address where and how it will operate, potential obstacles, competition and the Funding needed to get the business up and running.

This project "REAL ESTATE MANAGEMENT SYSTEM ” has undergone the following Feasibility study:

- Economic Feasibility
- Technical Feasibility
- Behavioral Feasibility
- Schedule Feasibility

Every project is feasible for given unlimited resources and infinitive time. Feasibility study is an evaluation of the proposed system regarding its workability, impact on the organization, ability to meet the user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development. Feasibility and risk analysis are related in many ways. The feasibility analysis in this project has been discussed below based on the above-mentioned components of feasibility.

TECHNICAL FEASIBILITY:

Technical feasibility centers on the technology used. It means the computerized system is technically feasible i.e., it doesn’t have any technical fault and work properly in the given environment. Our system is technically feasible it is providing us the required output.

ECONOMIC FEASIBILITY:

Economic analysis is the most frequently used method for evaluating the effectiveness of the computerized system. We analyze the computerized system is feasible as than the manual system because it saves the money, time and manpower. It is also feasible according to cost benefit analysis.

•BEHAVIORAL FEASIBILITY:

Behavioral feasibility is the analysis of behavior of the computerized system. In this we analysis that the computerized system is working properly or not. If working then it is communicating properly with the environment or not. All the matters are analyzed and a good computerized system is prepared.

• SCHEDULE FEASIBILITY:

Time evaluation is the most important consideration in the development of project. The time schedule required for the developed of this project is very important since more development time effect machine time, cost and cause delay in the development of other systems.

2.3 PROPOSED SYSTEM:

This website is an online real estate management through which individual agents or buyer can maintain their property document keeping and managing property registration and also access its information and manage all the adding, updating, deleting the ads and some of its tasks. The Admin user can inform their agents for regarding to property and update the information regarding property and cancellation of property or changing buyer choice.

Objectives

- ➢ The system should have a login. A login box should appear when the system is invoked.
- ➢ The Admin should have all the type of authority.
- ➢ The Admin should maintain property. Admin identify property type as it is residential or commercial property.
- ➢ The Admin user can inform their agents for regarding to property and update the information regarding property and cancellation of property or changing buyer choice.
- ➢ The user should book the property for sell or rent with detail of property.
- ➢ The system is very useful for the companies or builders that can post and edit their properties and their personal info and admin can monitor records of all of them.
- ➢ The system is also useful which also keeps track of Account details of buyers and Investors and also RES Industry.

3.SYSTEM DESIGN

Unified Modelling Language Diagrams(UML):

➢ The unified modelling language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.

➢ A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

User Model View

➢ This view represents the system from the user's perspective.

➢ The analysis representation describes a usage scenario from the end-users perspective.

Structural model view

➢ In this model the data and functionality are arrived from inside the system.

➢ This model view models the static structures.

Behavioural Model View

It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

Implementation Model View

In this the structural and behavioural as parts of the system are represented as they are to be built.

Environmental Model View

- In this the structural and behavioural aspects of the environment in which the system to be implemented are represented.
- UML is specifically constructed through two different domains they are
- UML Analysis modelling, which focuses on the user model and structural model views of the system?
- UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views

LOGICAL DESIGN:

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modeling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems, designs are included. Logical design includes entity-relationship diagrams (ER diagrams).

PHYSICAL DESIGN:

The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified /authenticated, how it is processed, and how it is displayed.

In physical design, the following requirements about the system are decided.

• Input requirement,

• Output requirements,

• Storage requirements,

• Processing requirements,

• System control and backup or recovery.

Put another way, the physical portion of system design can generally be broken down into three sub-tasks:

• User Interface Design

• Data Design

• Process Design

User Interface Design is concerned with how users add information to the system and with how the system presents information back to them. It is concerned with how the data is represented and stored within the system. Finally, **Process Design** is concerned with how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system.

At the end of the system design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase. Physical design, in this context, does not refer to the tangible physical design of an information system.

To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc. It involves a detailed design of a user and a product database structure processor and control processor. The H/S personal specification is developed for the proposed system.

3.1 E-R DIAGRAM

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases

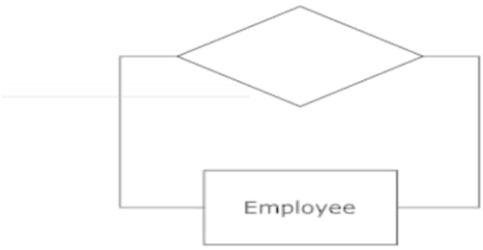
. **Structure of an Entity Relationship Diagram with Common ERD Notations**

An entity relationship diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

• **Entities**, which are represented by rectangles. An entity is an object or concept about which you want to store information.

- **Weak entity** is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.

- **Actions**, which are represented by diamond shapes, show how two entities share information in the database. In some cases, entities can be self-linked. For example, employees can supervise other employees.



- **Relationship**: The degree of a relationship is the number of entity types that participate in the relationship.

- **Attributes**, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an

employee's social security number might be the employee's key attribute.

- **Multi-valued attribute** can have more than one value. For example, an employee entity can have multiple skill values.

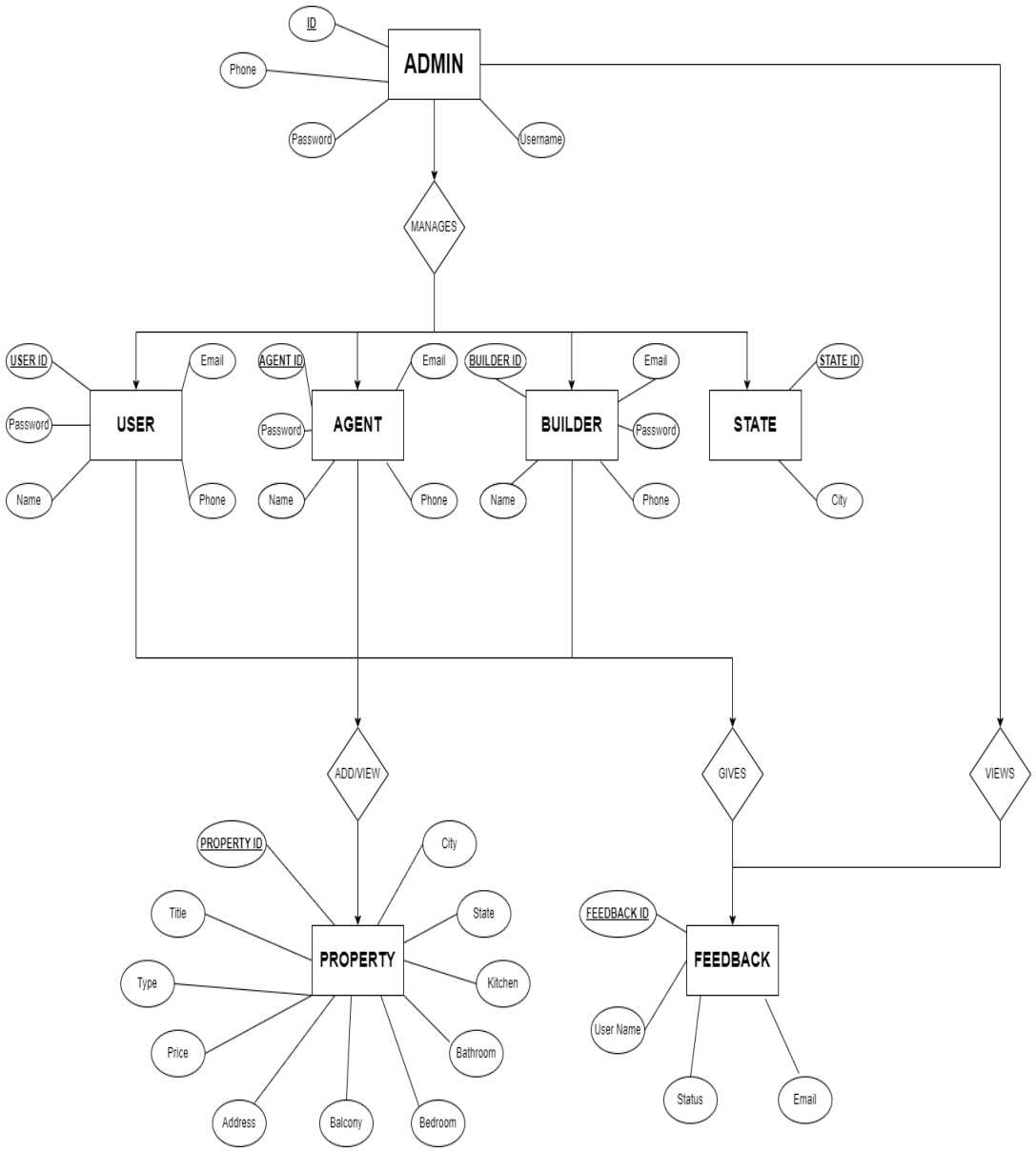
- **Derived attribute** is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.

- **Connecting lines**, solid lines that connect attributes to show the relationships of entities in the diagram.

- **Cardinality** specifies how many instances of an entity relate to one instance of another entity. Ordinarily is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinarily describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and cordiality specifies the absolute minimum number of relationships.

1. One to One
2. One to Many
3. Many to One
4. Many to Many

E-R DIAGRAM FOR FESTIVEPRO - SELLING FESTIVAL'S PRODUCTS ON WEB:



ADMIN – Admin is the one who manages the portal, Admin can also update his profile, change the password and recover the password.

Property Type: In this section, admin can manage property type (add/update).

Owner: In this section, admin can view the detail of owners.

Agents: In this section, admin can view the detail of agents. User: In this section, admin can view the detail of user.

3.2 DATA FLOW DIAGRAM (level 0 and level 1)

The Data Flow Diagrams (DFDs) are used for structure analysis and design. DFDs show the flow of data from external entities into the system. DFDs also show how the data moves and are transformed from one process to another, as well as its logical storage. The following symbols are used within DFDs.

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

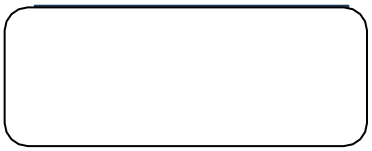
A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel

PHYSICAL VS LOGICAL DFD

A logical DFD captures the data flows that are necessary for a system to operate. It describes the processes that are undertaken, the data required and produced by each process, and the stores needed to hold the data. On the other hand, a physical DFD shows how the system is actually implemented, either at the moment (Current Physical DFD), or how the designer intendsit to be in the future (Required Physical DFD).

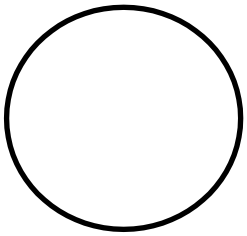
Thus, a Physical DFD may be used to describe the set of data items that appear on each piece of paper that move around an office, and the fact that a particular set of pieces of paper are stored together in a filing cabinet. It is quite possible that a Physical DFD will include references to data that are duplicated, or redundant, and that the data stores, if implemented as a set of database tables, would constitute an un-normalized (or de-normalized) relational database. In contrast, a Logical DFD attempts to capture the data flow aspects of a system in a form that has neither redundancy nor duplication.

DATA FLOW SYMBOLS AND THEIR MEANINGS: -



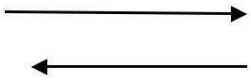
An entity: A source of data or a destination for data.

Source/Sink: Represented by rectangles in the diagram. Sources and Sinks are external entities which are sources or destinations of data, respectively.



Process: Represented by circles in the diagram. Processes are responsible for manipulating the data. They take data as input and output an altered version of the data.

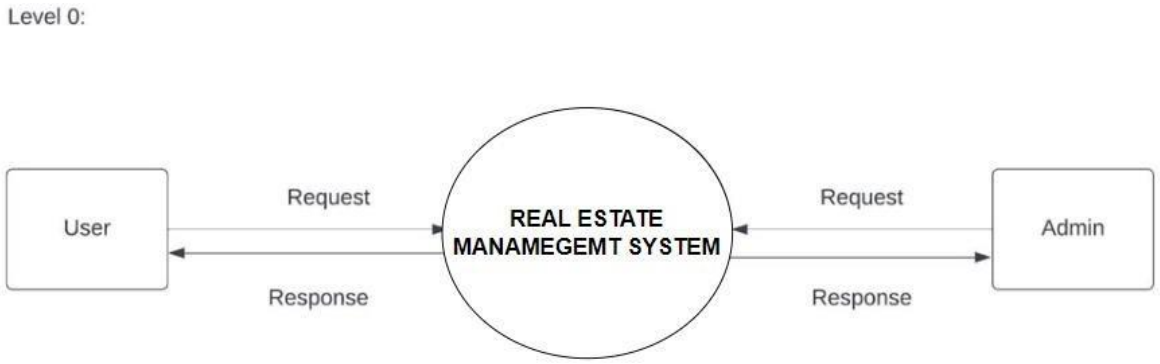
Data Store: Represented by a segmented rectangle with an open end on the right. Data Stores are both electronic and physical locations of data. Examples include databases, directories, files, and even filing cabinets and stacks of paper.



Data Flow: Represented by a unidirectional arrow. Data Flows show how data is moved through the System. Data Flows are labeled with a description of the data that is being passed through it.

A level-0 DFD is the most basic form of DFD. It aims to show how the entire system worksat a glance. There is only one process in the system and all the data flows either into or out of this process. Level-0 DFD's demonstrates the interactions between the process and external entities. They do not contain Data Stores.

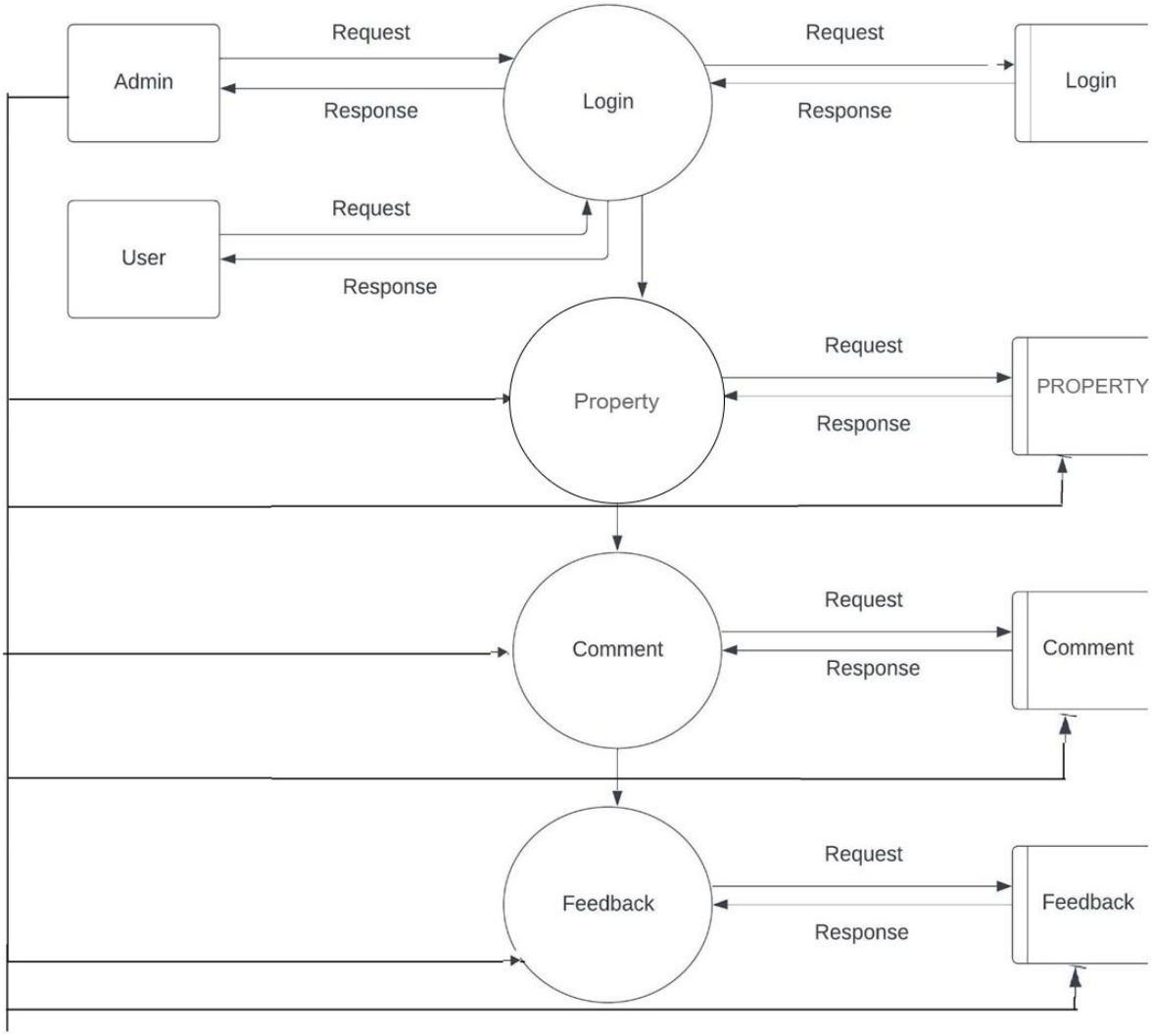
. Level 0 DFD/Context Diagram



LEVEL 1 DFD:

Level 1 DFD's aim is to give an overview of the full system. They look at the system in more detail. Major processes are broken down into sub-processes. Level 1 DFD's also identifies data stores that are used by the major processes. When constructing a Level 1 DFD we must startby examining the Context Level DFD. We must break up the single process into its sub- processes. We must then pick out the data stores from the text we are given and include them in our DFD. Like the Context Level DFD's, all entities, data stores and processes must be labeled. We must also state any assumptions made from the text.

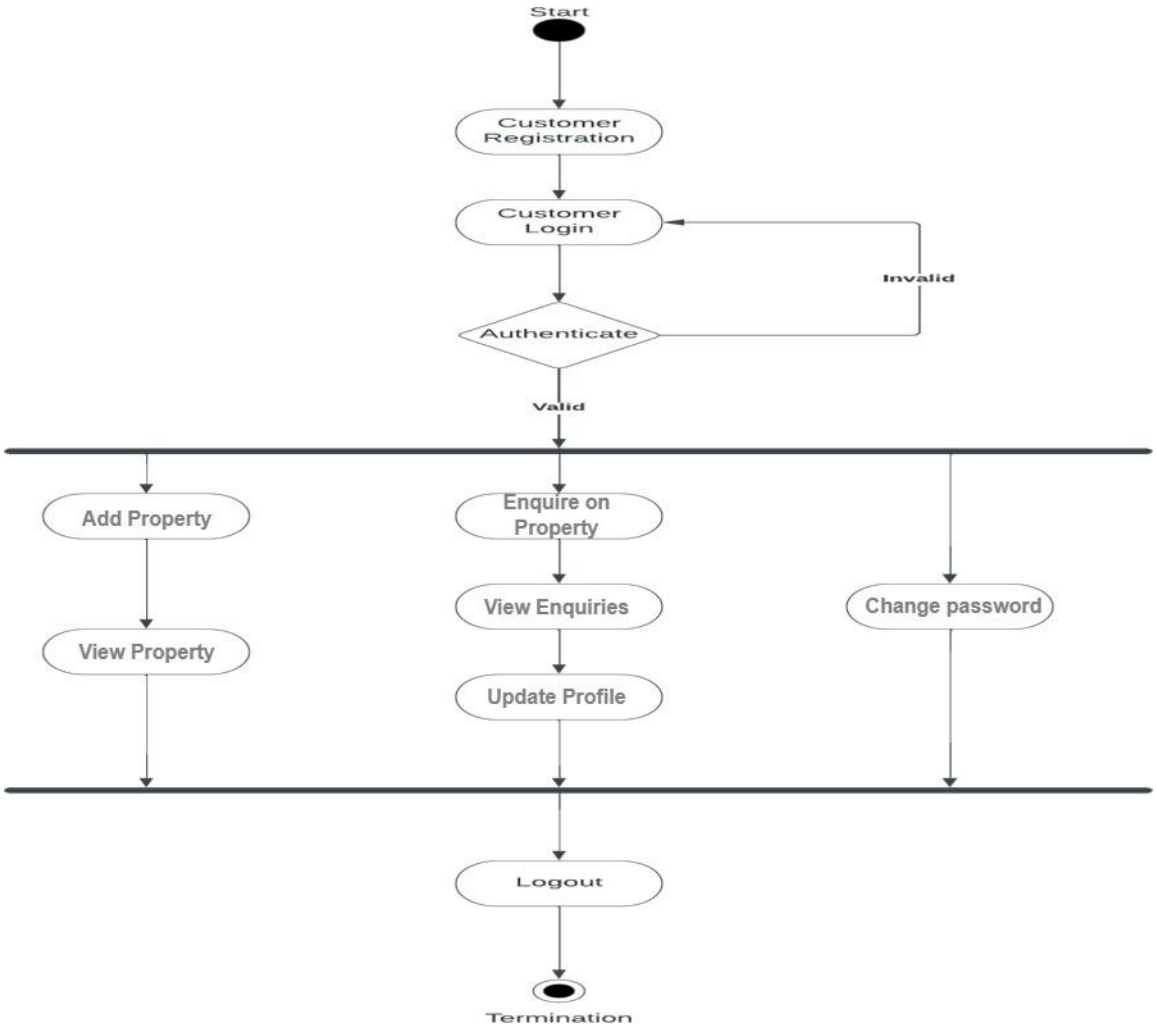
Level 1:


Admin
Request
Response
Login
Request
Response
Login
User
Request
Response
Property
Request
Response
PROPERTY
Comment
Request
Response
Comment
Feedback
Request
Response
Feedback

3.3 ACTIVITY DIAGRAM

An activity diagram visually represents the series of actions or flow of control in a system similar to a flow chart or data flow diagram. Activity diagram are often used in business processing modeling. They can also describe steps in a used case diagram. The activity diagram for Admin module and User module of College Gadget Booking is given below.

Activity Diagram:



3.4 GANTT CHART:

A Gantt chart is a type of bar chart, devised by Henry Gantt in the 1910s, that illustrates aproject schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project. Modern Gantt charts also show the dependency (i.e., precedence network) relationships between activities.

• HISTORICAL DEVELOPMENT:

The first known tool of this type was developed in 1896 by Karol Adamiecki, who called it a Harmon gram. Adamiecki did not publish his chart until 1931, however, and only in Polish, which limited both its adoption and recognition of his authorship. The chart is named after Henry Gantt (1861–1919), who designed his chart around the years 1910–1915. One of the first major applications of Gantt charts was by the United States during World War I, at the instigation of General William Crozier in the 1980s, personal computers allowed widespread creation of complex and elaborate Gantt charts. The first desktop applications were intended mainly for project managers and project schedulers. With the advent of the Internet and increased collaboration over networks at the end of the 1990s, Gantt charts became a common feature of web-based applications, including collaborative groupware.

• GANTT CHART BENEFITS:

Clarity:

One of the biggest benefits of a Gantt chart is the tool's ability to boil down multiple tasks and timelines into a single document. Stakeholders throughout an organization can easily understand where teams are in a process while grasping the ways in which independent elements come together toward project completion.

Communication:

Teams can use Gantt charts to replace meetings and enhance other status updates. Simply clarifying chart positions offers an easy, visual method to help team members understand task progress

• MOTIVATION:

Some teams or team members become more effective when faced with a form of external motivation. Gantt charts offer teams the ability to focus work at the front of a task timeline, or at the tail end of a chart segment. Both types of team members can find Gantt charts meaningful as they plug their own work habits into the overall project schedule.

• COORDINATION:

For project managers and resource schedulers, the benefits of a Gantt chart include the ability to sequence events and reduce the potential for overburdening team members. Some project managers even use combinations of charts to break down projects into more manageable sets of tasks.

• CREATIVITY:

Sometimes, a lack of time or resources forces project managers and teams to find creative solutions. Seeing how individual tasks intertwine on Gantt charts often encourages new partnerships and collaborations that might not have evolved under traditional task assignment systems.

• TIME MANAGEMENT:

Most managers regard scheduling as one of the major benefits of Gantt charts in a creative environment. Helping teams understand the overall impact of project delays can foster stronger collaboration while encouraging better task organization.

• FLEXIBILITY:

Whether you use Excel to generate Gantt charts or you load tasks into a more precise chart generator, the ability to issue new charts as your project evolves lets you react to unexpected changes in project scope or timeline. While revising your project schedule too frequently can eliminate some of the other benefits of Gantt charts, offering a realistic view of a project can help team members recover from setbacks or adjust to other changes.

• MANAGEABILITY:

For project managers handling complex assignments, like software publishing or event planning, the benefits of Gantt charts include externalizing assignments. By visualizing all of thepieces of a project puzzle, managers can make more focused, effective decisions about resources and timetables.

• EFFICIENCY:

Another one of the benefits of Gantt charts is the ability for teams members to leverage each other's deadlines for maximum efficiency. For instance, while one team member waits on the outcome of three other tasks before starting a crucial piece of the assignment, he or she can perform other project tasks. Visualizing resource usage during projects allows managers to make better use of people, places, and things.

• ACCOUNTABILITY:

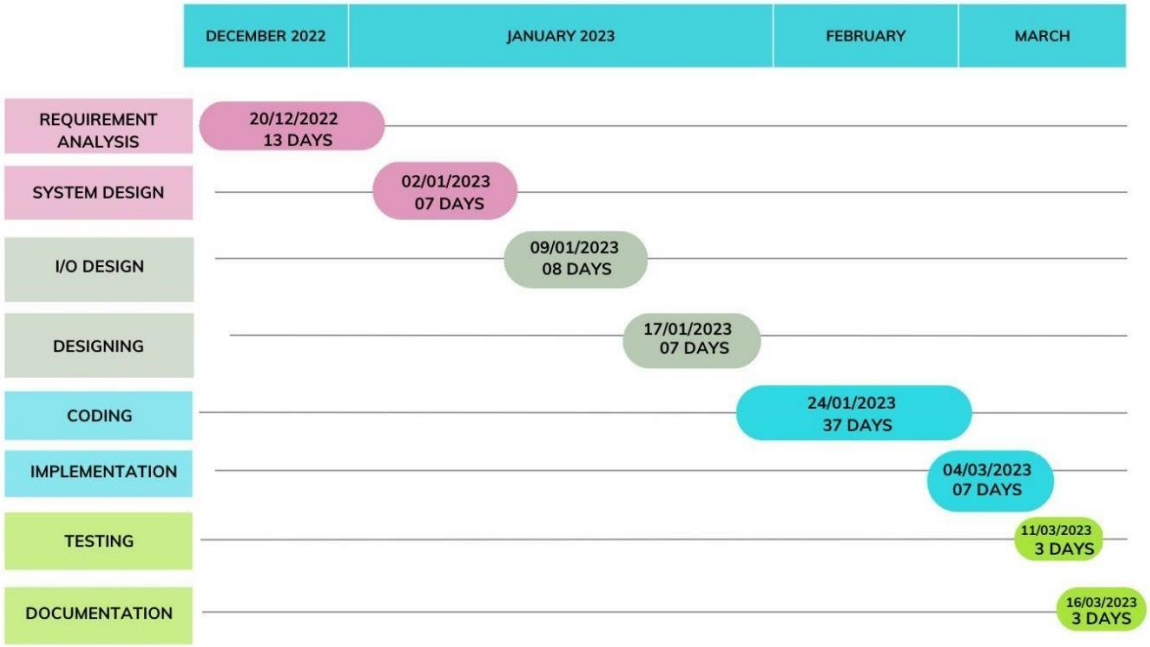
When project teams face major organizational change, documenting effort and outcomes becomes crucial to career success. Using Gantt charts during critical projects allows both project managers and participants to track team progress, highlighting both big wins and major failures during professional review periods; team members who frequently exceed expectations can leverage this documentation into larger raises or bonuses.

• GANTT CHART IMPORTANCE:

The project's summary and terminal elements, which combine to form the project's internal structure, are shown on the Gantt chart. Many charts will also depict the precedence rankings and dependencies of various tasks within the project. The charts can illustrate the start and finish project terminal elements in project management. It can also show summary elements and terminal dependencies. The smallest task tracked as part of the project effort is known as a terminal element. Gantt chart represents the tasks in most modern project scheduling packages. However other management applications use simpler communication tools such as message boards, to-do lists and simple scheduling etc., therefore, they do not use Gantt charts as heavily.

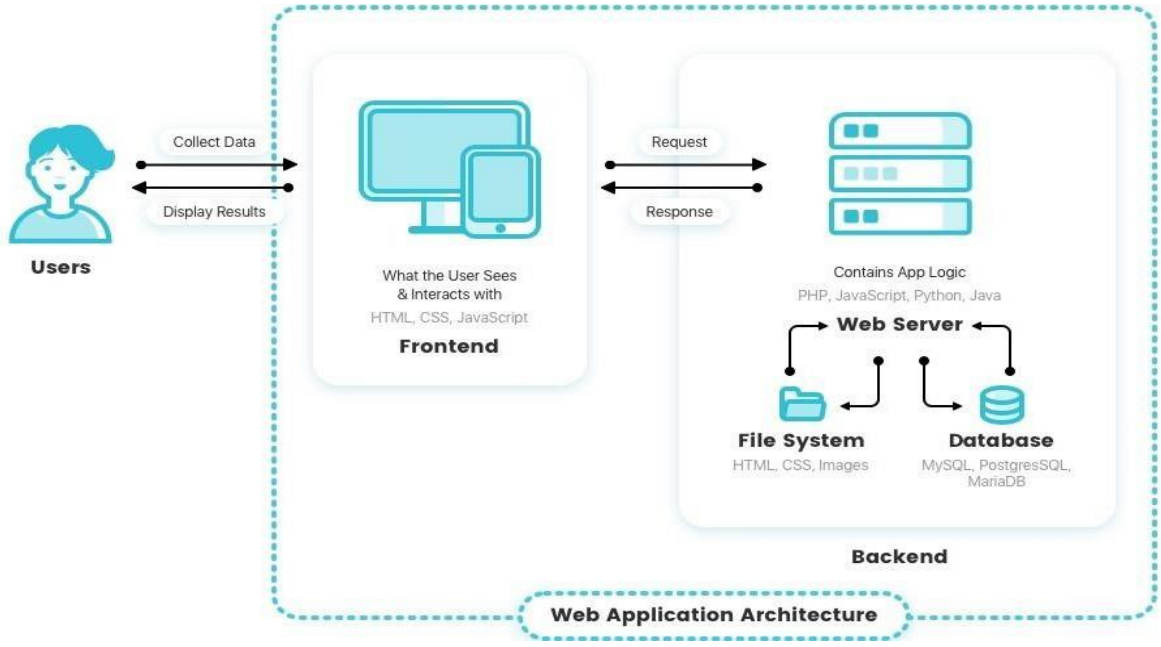
The way to create this chart begins by determining and listing the necessary activities. Next, sketch out how you expect the chart to look. List which items depend on others and what activities take place when. For each activity, list how many man-hours it will require, and who is responsible. Lastly, determine the throughput time.

This technique's primary advantage is its good graphical overview that is easy to understand for nearly all project participants and stakeholders. Its primary disadvantage is its limited applicability for many projects, since projects are often more complex than can be effectively communicated with this chart.



3.5. ARCHITECTURAL DESIGN

An architectural model (in software) is a rich and rigorous diagram, created using available standards, in which the primary concern is to illustrate a specific set of tradeoffs inherent in the structure and design of a system or ecosystem.



4.DATABASE DESIGN

Login:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(10)			No	*None*		AUTO_INCREMENT
2	FullName	varchar(120)	latin1_swedish_ci		Yes	*NULL*		
3	Email	varchar(120)	latin1_swedish_ci		Yes	*NULL*		
4	MobileNumber	bigint(10)			Yes	*NULL*		
5	Password	varchar(120)	latin1_swedish_ci		Yes	*NULL*		
6	UserType	int(5)			Yes	*NULL*		
7	PostingDate	timestamp			Yes	current_timestamp()		
8	Aboutme	mediumtext	latin1_swedish_ci		No	*None*		
9	UpdationDate	timestamp			No	0000-00-00 00:00:00		ON UPDATE CURRENT_TIMESTAMP()

Payment:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(10)			No	*None*		AUTO_INCREMENT
2	UserID	char(20)	latin1_swedish_ci		Yes	*NULL*		
3	PropertyTitle	mediumtext	latin1_swedish_ci		Yes	*NULL*		
4	PropertDescription	mediumtext	latin1_swedish_ci		Yes	*NULL*		
5	Type	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
6	Status	varchar(100)	latin1_swedish_ci		Yes	*NULL*		
7	Location	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
8	Bedrooms	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
9	Bathrooms	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
10	Floors	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
11	Garages	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
12	Area	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
13	Size	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
14	RentorsalePrice	varchar(120)	latin1_swedish_ci		Yes	*NULL*		
15	BeforePricelabel	varchar(120)	latin1_swedish_ci		Yes	*NULL*		
16	AfterPricelabel	varchar(120)	latin1_swedish_ci		Yes	*NULL*		
17	PropertyID	varchar(120)	latin1_swedish_ci		Yes	*NULL*		
18	CenterCooling	int(4)			Yes	*NULL*		
19	Balcony	int(4)			Yes	*NULL*		
20	PetFriendly	int(4)			Yes	*NULL*		
21	Barbeque	int(4)			Yes	*NULL*		
22	FireAlarm	int(4)			Yes	*NULL*		
23	ModernKitchen	int(4)			Yes	*NULL*		
24	Storage	int(4)			Yes	*NULL*		
25	Dryer	int(4)			Yes	*NULL*		
26	Heating	int(4)			Yes	*NULL*		
27	Pool	int(4)			Yes	*NULL*		
28	Laundry	int(4)			Yes	*NULL*		
29	Sauna	int(4)			Yes	*NULL*		
30	Gym	int(4)			Yes	*NULL*		
31	Elevator	int(4)			Yes	*NULL*		
32	DishWasher	int(4)			Yes	*NULL*		
33	EmergencyExit	int(4)			Yes	*NULL*		
34	FeaturedImage	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
35	GalleryImage1	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
36	GalleryImage2	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
37	GalleryImage3	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
38	GalleryImage4	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
39	GalleryImage5	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
40	Address	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
41	Country	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
42	City	varchar(220)	latin1_swedish_ci		Yes	*NULL*		
43	State	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
44	ZipCode	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
45	Neighborhood	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
46	ListingDate	timestamp			Yes	current_timestamp()		

Feedback:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	UserId	int(11)			No	*None*		
3	PropertyId	int(11)			No	*None*		
4	UserRemark	mediumtext	latin1_swedish_ci		No	*None*		
5	PostingDate	timestamp			Yes	current_timestamp()		
6	Is_Publish	int(1)			Yes	*NULL*		

Enquiry:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	int(11)			No	*None*		AUTO_INCREMENT
2	PropertyID	int(10)			No	*None*		
3	FullName	varchar(200)	latin1_swedish_ci		No	*None*		
4	Email	varchar(200)	latin1_swedish_ci		No	*None*		
5	MobileNumber	bigint(10)			No	*None*		
6	Message	mediumtext	latin1_swedish_ci		No	*None*		
7	EnquiryNumber	varchar(200)	latin1_swedish_ci		No	*None*		
8	EnquiryDate	timestamp			No	current_timestamp()		
9	Status	varchar(10)	latin1_swedish_ci		Yes	*NULL*		
10	Remark	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
11	RemarkDate	timestamp			Yes	*NULL*		ON UPDATE CURRENT_TIMESTAMP()

5. SYSTEM CONFIGURATION

5.1 HARDWARE REQUIREMENTS

RAM	4.00GB
Hard disk	500GB
Processor	i3
Processing speed	2.00GHz

5.2 SOFTWARE REQUIREMENTS:

Front end	HTML, CSS, JAVASCRIPT
Back end	PHP, MYSQL
Tools	VS Code, LucidChart, Visual Studio,powerbi
Operating System	Windows 10
Documentation	Microsoft Word 2010

6. DETAILS OF THE SOFTWARE

A development process consists of various phases, each phase ending with a defined output. The phases are performed in an order specified by the process model being followed. The main reason for having a phased process is that it breaks the problem of developing software into successfully performing a set of phases, each handling a different concern of software development.

This ensures that the cost of development is lower than what it would have been if the whole problem were tackled together. A phased development process is central to the software engineering approach for solving the software crisis

6.1 OVERVIEW OF FRONT END

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It can be used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework,

.NET Compact Framework and Microsoft Silver light. Visual Basic 9.0 was released together with the Microsoft .NET Framework 3.5 on 19 November 2007. ASP.NET is a set of Web development tools offered by Microsoft. Programs like Visual Studio .NET and Visual Web Developer allow Web developers to create dynamic websites using a visual interface. ASP.NET also supports Visual Basic .NET, Jscript. .NET and open-source languages like Python and Perl. .NET provide a visual interface for developers to create their applications, which makes .NET a reasonable choice for designing Web-based interfaces as well. SQL Server is a relational database developed and sold by Microsoft. Originally bought from Sybase, Microsoft has released versions 6, 6.5, 7, 2000 and 2005. The Express Edition of SQL Server 2005 is used in the C++ and VB.

6.2 OVERVIEW OF BACK-END

Introduction to SQL Server 2008 The Structured Query Language (SQL) comprises one of the fundamental building blocks of modern database architecture. SQL defines the methods used to create and manipulate relational databases on all major platforms.

SQL comes in many flavors. Oracle databases utilize their proprietary PL / SQL. Microsoft SQL Server makes use of Transact – SQL. However, all of these variations are based upon the industry standard ANSI SQL.

SQL commands can be divided into two main sub languages. The Data Definition Language (DDL) contains the commands used to create and destroy databases and database objects. After the database structureis defined with DDL, database administrators and users can utilize the Data Manipulation Language to insert, retrieve and modify the data contained within it.

SQL Server 2008 is a powerful tool for turning information into opportunity. Industry leading support for XML, enhanced tools for system management and tuning, and exceptional scalability and reliability make SQL Server 2000 the best choice for the agile enterprise.

6.3 ABOUT THE PLATFORM

Windows is a series of Operating Systems developed by Microsoft. Each version of Windows includes aGraphical User Interface, with a desktop that allows users to view files and folders in Windows. For the past two decades, Windows has been the most widely used operating system for personal computers PCs.

Microsoft Windows is designed for both home computing and professional purposes. Past versions of Windows home editions include Windows 3.0 (1990), Windows 3.1 (1992), Windows 95 (1995), Windows 98 (1998), Windows Me (2000), Windows XP (2001), and Windows Vista (2006). The current version, Windows7, was released in 2009.

The first business-oriented version of Windows, called Windows NT 3.1, was in 1993. This was followed by Windows 3.5, 4.0, and Windows 2000. When Microsoft released

Windows XP in 2001, the company simply created different editions of the operating system for personal and business purposes.Windows Vista and Windows 7 have followed the same release strategy.

Windows is designed to run on standard x86 hardware, such as Intel and AMD processors. Therefore, it can be installed on multiple brands of hardware, such as Dell, HP, and Sony computers, as well as home-built PCs. Windows 7 also includes several touch screen features, that allow the operating system to run on certain tablets and computers with touch screen displays. Microsoft's mobile operating system, Windows Phone, is designed specifically for smart phones and runs on several brands of phones, including HTC, Nokia, and Samsung.

PHP

PHP is a general-purpose scripting language geared toward web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1993 and released in 1995. The PHP reference implementation is now produced by The PHP Group. PHP was originally an abbreviation of Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside the web context, such as standalone graphical applications and robotic drone control. PHP code can also be directly executed from the command line.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on a variety of operating systems and platforms.

7. CODING

About us:

```
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');
if (strlen($_SESSION['remsaid']==0)) {
  header('location:logout.php');
  } else{
    if(isset($_POST['submit']))
  {


$cmsaid=$_SESSION['remsaid'];
 $pagetitle=$_POST['pagetitle'];
$pagedes=$_POST['pagedes'];


 $query=mysqli_query($con,"update tblpage set
PageTitle='$pagetitle',PageDescription='$pagedes' where  PageType='aboutus'");


    if ($query) {
    $msg="About Us has been updated.";
  }
  else
    {
      $msg="Something Went Wrong. Please try again";
```

```
    }
}
?>
<!doctype html>
<html lang="en">
<head>
    <title>My Dream House || About Us</title>
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="assets/vendor/bootstrap/css/bootstrap.min.css">
    <link href="assets/vendor/fonts/circular-std/style.css" rel="stylesheet">
    <link rel="stylesheet" href="assets/libs/css/style.css">
    <link rel="stylesheet" href="assets/vendor/fonts/fontawesome/css/fontawesome-all.css">
    <script src="http://js.nicedit.com/nicEdit-latest.js" type="text/javascript"></script>
<script type="text/javascript">bkLib.onDomLoaded(nicEditors.allTextAreas);</script>
</head>

<body>
    <!-- ============================================================== -->
    <!-- main wrapper -->
    <!-- ============================================================== -->
    <div class="dashboard-main-wrapper">

        <?php include_once('includes/header.php');?>

      <?php include_once('includes/sidebar.php');?>
```

```
      <div class="dashboard-wrapper">

        <div class="container-fluid  dashboard-content">

          <!--
================================================================== -->

          <!-- pageheader -->

          <!--
================================================================== -->

          <div class="row">

            <div class="col-xl-12 col-lg-12 col-md-12 col-sm-12 col-12">

              <div class="page-header">

                <h2 class="pageheader-title">About Us </h2>


                <div class="page-breadcrumb">

                  <nav aria-label="breadcrumb">

                    <ol class="breadcrumb">

                      <li class="breadcrumb-item"><a href="dashboard.php"
class="breadcrumb-link">Dashboard</a></li>


                      <li class="breadcrumb-item active" aria-current="page">About Us

                    </ol>

                  </nav>

                </div>

              </div>

            </div>

          </div>


          <div class="row">
```

```
                <!--
================================================================== -->

                <!-- valifation types -->

                <!--
================================================================== -->

                <div class="col-xl-12 col-lg-12 col-md-12 col-sm-12 col-12">

                  <div class="card">

                    <h5 class="card-header">About Us</h5>

                    <div class="card-body">

                      <form  method="post">

                        <p style="font-size:16px; color:red" align="center"> <?php if($msg){

  echo $msg;

 }  ?> </p>

<?php


$ret=mysqli_query($con,"select * from  tblpage where PageType='aboutus'");

$cnt=1;

while ($row=mysqli_fetch_array($ret)) {


?>


                          <div class="form-group row">

                            <label class="col-12 col-sm-3 col-form-label text-sm-right">Page
Title</label>

                            <div class="col-12 col-sm-8 col-lg-6">

                              <input type="text" name="pagetitle" id="pagetitle"
required="true" value="<?php  echo $row['PageTitle'];?>" class="form-control">
```

```
                        </div>
                      </div>
                      <div class="form-group row">
                        <label class="col-12 col-sm-3 col-form-label text-sm-right">Page
Description</label>
                        <div class="col-12 col-sm-8 col-lg-6">
                          <textarea type="text" name="pagedes" id="pagedes"
required="true"class="form-control"><?php  echo $row['PageDescription'];?></textarea>
                        </div>
                      </div>
                    <?php } ?>
                      <div class="form-group row text-right">
                        <div class="col col-sm-10 col-lg-9 offset-sm-1 offset-lg-0">
                          <p style="text-align: center;"> <button type="submit" class="btn
btn-space btn-primary" name="submit">Submit</button></p>

                        </div>
                      </div>
                    </form>
                  </div>
                </div>
              </div>
              <!--
================================================================== -->
              <!-- end valifation types -->
              <!--
================================================================== -->
```

```
            </div>


        </div>
        <!-- ================================================================== -
->
        <!-- footer -->
        <!-- ================================================================== -
->
         <?php include_once('includes/footer.php');?>
        <!-- ================================================================== -
->
        <!-- end footer -->
        <!-- ================================================================== -
->
      </div>
   </div>
   <!-- ================================================================== -->
   <!-- end main wrapper -->
   <!-- ================================================================== -->
   <!-- Optional JavaScript -->
   <script src="assets/vendor/jquery/jquery-3.3.1.min.js"></script>
   <script src="assets/vendor/bootstrap/js/bootstrap.bundle.js"></script>
   <script src="assets/vendor/slimscroll/jquery.slimscroll.js"></script>
   <script src="assets/vendor/parsley/parsley.js"></script>
   <script src="assets/libs/js/main-js.js"></script>
   <script>
   $('#form').parsley();
```

```
  </script>
  <script>
  // Example starter JavaScript for disabling form submissions if there are invalid fields
  (function() {
    'use strict';
    window.addEventListener('load', function() {
      // Fetch all the forms we want to apply custom Bootstrap validation styles to
      var forms = document.getElementsByClassName('needs-validation');
      // Loop over them and prevent submission
      var validation = Array.prototype.filter.call(forms, function(form) {
        form.addEventListener('submit', function(event) {
          if (form.checkValidity() === false) {
            event.preventDefault();
            event.stopPropagation();
          }
          form.classList.add('was-validated');
        }, false);
      });
    }, false);
  })();
  </script>
</body>

</html>
<?php }  ?>
```

Change of password:

```
<?php
session_start();
include('includes/dbconnection.php');
error_reporting(0);
if (strlen($_SESSION['remsaid']==0)) {
  header('location:logout.php');
  } else{
if(isset($_POST['submit']))
{
$adminid=$_SESSION['remsaid'];
$cpassword=md5($_POST['currentpassword']);
$newpassword=md5($_POST['newpassword']);
$query=mysqli_query($con,"select ID from tbladmin where ID='$adminid' and
Password='$cpassword'");
$row=mysqli_fetch_array($query);
if($row>0){
$ret=mysqli_query($con,"update tbladmin set Password='$newpassword' where
ID='$adminid'");
$msg= "Your password successully changed";
} else {
$msg="Your current password is wrong";
}
}
?>
<!doctype html>
```

```
<html lang="en">
<head>
   <title>My Dream House || Change Password</title>
   <!-- Bootstrap CSS -->
   <link rel="stylesheet" href="assets/vendor/bootstrap/css/bootstrap.min.css">
   <link href="assets/vendor/fonts/circular-std/style.css" rel="stylesheet">
   <link rel="stylesheet" href="assets/libs/css/style.css">
   <link rel="stylesheet" href="assets/vendor/fonts/fontawesome/css/fontawesome-all.css">
   <script type="text/javascript">
function checkpass()
{
if(document.changepassword.newpassword.value!=document.changepassword.confirmpasswor
d.value)
{
alert('New Password and Confirm Password field does not match');
document.changepassword.confirmpassword.focus();
return false;
}
return true;
}
</script>
</head>
<body>
   <!-- ============================================================== -->
   <!-- main wrapper -->
   <!-- ============================================================== -->
```

```
   <div class="dashboard-main-wrapper">
      <?php include_once('includes/header.php');?
     <?php include_once('includes/sidebar.php');?>
      <div class="dashboard-wrapper">
         <div class="container-fluid  dashboard-content">
            <!--
============================================================== -->
            <!-- pageheader -->
            <!--
============================================================== -->
            <div class="row">
               <div class="col-xl-12 col-lg-12 col-md-12 col-sm-12 col-12">
                  <div class="page-header">
                     <h2 class="pageheader-title">Change Password </h2>
                     <div class="page-breadcrumb">
                        <nav aria-label="breadcrumb">
                           <ol class="breadcrumb">
                              <li class="breadcrumb-item"><a href="dashboard.php"
class="breadcrumb-link">Dashboard</a></li>
                              <li class="breadcrumb-item active" aria-current="page">Change
Password</li>
                           </ol>
                        </nav>
                     </div>
                  </div>
               </div>
            </div>
```

```
                <div class="row">
                  <!--
================================================================== -->
                  <!-- valifation types -->
                  <!--
================================================================== -->
                  <div class="col-xl-12 col-lg-12 col-md-12 col-sm-12 col-12">
                    <div class="card">
                      <h5 class="card-header">Change Password</h5>
                      <div class="card-body">
                        <form  method="post" name="changepassword" onsubmit="return
checkpass();">
                          <p style="font-size:16px; color:red" align="center"> <?php if($msg){
   echo $msg;
 }  ?> </p>
 <?php
$adminid=$_SESSION['remsaid'];
$ret=mysqli_query($con,"select * from tbladmin where ID='$adminid'");
$cnt=1;
while ($row=mysqli_fetch_array($ret)) {

?>
  <div class="form-group row">
                            <label class="col-12 col-sm-3 col-form-label text-sm-right">Current
Password</label>
                            <div class="col-12 col-sm-8 col-lg-6">
```

```
                        <input type="password" name="currentpassword"
id="currentpassword" class="form-control" value="" required='true'>
                    </div>
                </div>
                <div class="form-group row">
                    <label class="col-12 col-sm-3 col-form-label text-sm-right">New
Password</label>
                    <div class="col-12 col-sm-8 col-lg-6">
                        <input type="password" name="newpassword"
id="newpassword" class="form-control" value="" required='true'>
                    </div>
                </div>
                <div class="form-group row">
                    <label class="col-12 col-sm-3 col-form-label text-sm-right">Confirm
Password</label>
                    <div class="col-12 col-sm-8 col-lg-6">
                        <input type="password" name="confirmpassword"
id="confirmpassword" class="form-control" value="" required='true'>
                    </div>
                </div>
            <?php } ?>
                <div class="form-group row text-right">
                    <div class="col col-sm-10 col-lg-9 offset-sm-1 offset-lg-0">
                        <p style="text-align: center;"> <button type="submit" class="btn
btn-space btn-primary" name="submit">Change</button></p>

                    </div>
                </div>
```

```
                        </form>
                      </div>
                    </div>
                  </div>
                  <!--
========================================================================== -->
                  <!-- end valifation types -->
                  <!--
========================================================================== -->
                </div>

        </div>
        <!-- ========================================================================== -
->
        <!-- footer -->
        <!-- ========================================================================== -
->
         <?php include_once('includes/footer.php');?>
        <!-- ========================================================================== -
->
        <!-- end footer -->
        <!-- ========================================================================== -
->
      </div>
   </div>
   <!-- ========================================================================== -->
   <!-- end main wrapper -->
   <!-- ========================================================================== -->
```

```
<!-- Optional JavaScript -->
<script src="assets/vendor/jquery/jquery-3.3.1.min.js"></script>
<script src="assets/vendor/bootstrap/js/bootstrap.bundle.js"></script>
<script src="assets/vendor/slimscroll/jquery.slimscroll.js"></script>
<script src="assets/vendor/parsley/parsley.js"></script>
<script src="assets/libs/js/main-js.js"></script>
<script>
$('#form').parsley();
</script>
<script>
// Example starter JavaScript for disabling form submissions if there are invalid fields
(function() {
  'use strict';
  window.addEventListener('load', function() {
    // Fetch all the forms we want to apply custom Bootstrap validation styles to
    var forms = document.getElementsByClassName('needs-validation');
    // Loop over them and prevent submission
    var validation = Array.prototype.filter.call(forms, function(form) {
      form.addEventListener('submit', function(event) {
        if (form.checkValidity() === false) {
          event.preventDefault();
          event.stopPropagation();
        }
        form.classList.add('was-validated');
      }, false);
    });
```

```
      }, false);
   })();
   </script>
</body>


</html>
<?php }  ?>
```

Login page:

```
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');


if(isset($_POST['login']))
  {
   $adminuser=$_POST['username'];
   $password=md5($_POST['password']);
   $query=mysqli_query($con,"select ID from tbladmin where  UserName='$adminuser' && Password='$password' ");
   $ret=mysqli_fetch_array($query);
   if($ret>0){
     $_SESSION['remsaid']=$ret['ID'];
    header('location:dashboard.php');
   }
   else{
   $msg="Invalid Details.";
```

```
  }
 }
 ?>
<!doctype html>
<html lang="en">

<head>

   <title>My Dream House || Login</title>
   <!-- Bootstrap CSS -->
   <link rel="stylesheet" href="assets/vendor/bootstrap/css/bootstrap.min.css">
   <link href="../assets/vendor/fonts/circular-std/style.css" rel="stylesheet">
   <link rel="stylesheet" href="assets/libs/css/style.css">
   <link rel="stylesheet" href="assets/vendor/fonts/fontawesome/css/fontawesome-all.css">
   <style>
   html,
   body {
      height: 100%;
   }

   body {
      display: -ms-flexbox;
      display: flex;
      -ms-flex-align: center;
      align-items: center;
      padding-top: 40px;
```

```
    padding-bottom: 40px;
  }
  </style>
</head>

<body>
  <!-- ============================================================== -->
  <!-- login page  -->
  <!-- ============================================================== -->
  <div class="splash-container">
    <div class="card ">
      <div class="card-header text-center"><h2 style="color: blue">REMS</h2><span class="splash-description">Please enter your user information.</span></div>
      <div class="card-body">
       <form action="" method="post" name="login">
            <p style="font-size:16px; color:red" align="center"> <?php if($msg){
  echo $msg;
 }  ?> </p>
          <div class="form-group">
            <input class="form-control form-control-lg" id="username" type="text" placeholder="Username" required="true" name="username" autocomplete="off">
          </div>
          <div class="form-group">
            <input class="form-control form-control-lg" id="password" type="password" placeholder="Password" name="password" required="true">
          </div>
```

```
                <button type="submit" class="btn btn-primary btn-lg btn-block"
name="login">Sign in</button>
            </form>
        </div>
        <div class="card-footer bg-white p-0  ">

            <div class="card-footer-item card-footer-item-bordered">
                <a href="forgot-password.php" class="footer-link">Forgot Password</a>

            </div>
            <div class="card-footer-item card-footer-item-bordered">
                <a href="../index.php" class="footer-link">Back to Home</a>

            </div>
        </div>
    </div>
  </div>

  <!-- ============================================================== -->
  <!-- end login page  -->
  <!-- ============================================================== -->
  <!-- Optional JavaScript -->
  <script src="assets/vendor/jquery/jquery-3.3.1.min.js"></script>
  <script src="assets/vendor/bootstrap/js/bootstrap.bundle.js"></script>
</body>
</html>
```

ADD Property:

```
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');
if (strlen($_SESSION['remsuid']==0 || $_SESSION['ut']==3)) {
echo "<script>alert('Please login for add property.');</script>";
echo "<script>window.location.href ='logout.php'</script>";
 }
 else{

if(isset($_POST['submit']))
 {
$uid=$_SESSION['remsuid'];
$protitle=$_POST['propertytitle'];
$prodec=$_POST['propertydescription'];
$type=$_POST['selecttype'];
$status=$_POST['status'];
$location=$_POST['location'];
$bedrooms=$_POST['bedrooms'];
$bathrooms=$_POST['bathrooms'];
$floors=$_POST['floors'];
$garages=$_POST['garages'];
$area=$_POST['area'];
$size=$_POST['size'];
$srprice=$_POST['salerentprice'];
```

```
$beforepricelabel=$_POST['beforepricelabel'];

$afterpricelabel=$_POST['afterpricelabel'];

$ccolling=$_POST['centercolling'];

$balcony=$_POST['balcony'];

$petfrndly=$_POST['petfrndly'];

$barbeque=$_POST['barbeque'];

$firealarm=$_POST['firealarm'];

$modkitchen=$_POST['modkitchen'];

$storage=$_POST['storage'];

$dryer=$_POST['dryer'];

$heating=$_POST['heating'];

$pool=$_POST['pool'];

$laundry=$_POST['laundry'];

$sauna=$_POST['sauna'];

$gym=$_POST['gym'];

$elevator=$_POST['elevator'];

$dishwasher=$_POST['dishwasher'];

$eexit=$_POST['eexit'];


$proaddress=$_POST['address'];

$procountry=$_POST['country'];

$procity=$_POST['city'];

$prostate=$_POST['state'];

$prozipcode=$_POST['zipcode'];

$neighborhood=$_POST['neighborhood'];
```

```
$proid=mt_rand(100000000, 999999999);

//fetured Image

$pic=$_FILES["featuredimage"]["name"];

$extension = substr($pic,strlen($pic)-4,strlen($pic));

//Property  Image 1

$pic1=$_FILES["galleryimage1"]["name"];

$extension1 = substr($pic1,strlen($pic1)-4,strlen($pic1));

//Property  Image 2

$pic2=$_FILES["galleryimage2"]["name"];

$extension2 = substr($pic2,strlen($pic2)-4,strlen($pic2));

//Property  Image 3

$pic3=$_FILES["galleryimage3"]["name"];

$extension3 = substr($pic3,strlen($pic3)-4,strlen($pic3));

//Property  Image 4

$pic4=$_FILES["galleryimage4"]["name"];

$extension4 = substr($pic4,strlen($pic4)-4,strlen($pic4));

//Property  Image 5

$pic5=$_FILES["galleryimage5"]["name"];

$extension5 = substr($pic5,strlen($pic5)-4,strlen($pic5));

// allowed extensions

$allowed_extensions = array(".jpg","jpeg",".png",".gif");

// Validation for allowed extensions .in_array() function searches an array for a specific value.

if(!in_array($extension,$allowed_extensions))

{

echo "<script>alert('Featured image has Invalid format. Only jpg / jpeg/ png /gif format
allowed');</script>";
```

```
}
if(!in_array($extension1,$allowed_extensions))
{
echo "<script>alert('Property gallery image1 has Invalid format. Only jpg / jpeg/ png /gif format allowed');</script>";
}
if(!in_array($extension2,$allowed_extensions))
{
echo "<script>alert('Property gallery image2 has Invalid format. Only jpg / jpeg/ png /gif format allowed');</script>";
}
if(!in_array($extension3,$allowed_extensions))
{
echo "<script>alert('Property gallery image3 has Invalid format. Only jpg / jpeg/ png /gif format allowed');</script>";
}
if(!in_array($extension4,$allowed_extensions))
{
echo "<script>alert('Property gallery image4 has Invalid format. Only jpg / jpeg/ png /gif format allowed');</script>";
}
if(!in_array($extension5,$allowed_extensions))
{
echo "<script>alert('Property gallery image5 has Invalid format. Only jpg / jpeg/ png /gif format allowed');</script>";
}
else
```

```
{
//rename property images
$propic=md5($pic).time().$extension;
$propic1=md5($pic1).time().$extension1;
$propic2=md5($pic2).time().$extension2;
$propic3=md5($pic3).time().$extension3;
$propic4=md5($pic4).time().$extension4;
$propic5=md5($pic5).time().$extension5;
    move_uploaded_file($_FILES["featuredimage"]["tmp_name"],"propertyimages/".$propic);

move_uploaded_file($_FILES["galleryimage1"]["tmp_name"],"propertyimages/".$propic1);

move_uploaded_file($_FILES["galleryimage2"]["tmp_name"],"propertyimages/".$propic2);

move_uploaded_file($_FILES["galleryimage3"]["tmp_name"],"propertyimages/".$propic3);

move_uploaded_file($_FILES["galleryimage4"]["tmp_name"],"propertyimages/".$propic4);

move_uploaded_file($_FILES["galleryimage5"]["tmp_name"],"propertyimages/".$propic5);

   $query=mysqli_query($con,"insert into
tblproperty(UserID,PropertyTitle,PropertDescription,Type,Status,Location,Bedrooms,Bathroo
ms,Floors,Garages,Area,Size,RentorsalePrice,BeforePricelabel,AfterPricelabel,PropertyID,Cent
erCooling,Balcony,PetFriendly,Barbeque,FireAlarm,ModernKitchen,Storage,Dryer,Heating,Po
ol,Laundry,Sauna,Gym,Elevator,DishWasher,EmergencyExit,FeaturedImage,GalleryImage1,G
alleryImage2,GalleryImage3,GalleryImage4,GalleryImage5,Address,Country,City,State,ZipCo
de,Neighborhood)value('$uid','$protitle','$prodec','$type','$status','$location','$bedrooms','$bat
hrooms','$floors','$garages','$area','$size','$srprice','$beforepricelabel','$afterpricelabel','$proi
d','$ccolling','$balcony','$petfrndly','$barbeque','$firealarm','$modkitchen','$storage','$dryer',
'$heating','$pool','$laundry','$sauna','$gym','$elevator','$dishwasher','$eexit','$propic','$propi
```

```
c1','$propic2','$propic3','$propic4','$propic5','$proaddress','$procountry','$procity','$prostate'
,'$prozipcode','$neighborhood')");

  if ($query) {

   echo '<script>alert("Property detail has been added.")</script>';

echo "<script>window.location.href ='add-property.php'</script>";

 }

 else

  {

      echo '<script>alert("Something Went Wrong. Please try again")</script>';

  }




}

}




?>


 <!DOCTYPE html>

<html dir="ltr" lang="en-US">


<head>


  <!-- Fonts

  ========================================== -->
```

```
    <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i,8
00,800i%7CPoppins:100,100i,200,200i,300,300i,400,400i,500,500i,600,600i,700,700i,800,800i,900,
900i" rel="stylesheet">
    <!-- Stylesheets
    ================================================== -->
    <link href="assets/css/external.css" rel="stylesheet">
    <link href="assets/css/bootstrap.min.css" rel="stylesheet">
    <link href="assets/css/style.css" rel="stylesheet">

    <title>My Dream House|| Add Property</title>
</head>
<script>
function getsate(val) {
  $.ajax({
type:"POST",
url:"get-sate.php",
data:'$countryid='+val,
success:function(data){
$("#state").html(data);
}

  });
}
</script>

<script>
```

```
function getcity(val1) {
  $.ajax({
type:"POST",
url:"get-sate.php",
data:'$stateid='+val1,
success:function(data){
$("#city").html(data);
}

  });
}
</script>



<body>
   <!-- Document Wrapper
          ============================================= -->
   <div id="wrapper" class="wrapper clearfix">
      <?php include_once('includes/header.php');?>

      <!-- Page Title #1
============================================= -->
      <section id="page-title" class="page-title bg-overlay bg-overlay-dark2">
         <div class="bg-section">
            <img src="assets/images/page-titles/1.jpg" alt="Background" />
         </div>
```

```
    <div class="container">
      <div class="row">
        <div class="col-xs-12 col-sm-12 col-md-6 col-md-offset-3">
          <div class="title title-1 text-center">
            <div class="title--content">
              <div class="title--heading">
                <h1>Add Property</h1>
              </div>
              <ol class="breadcrumb">
                <li><a href="#">Home</a></li>
                <li class="active">Add Property</li>
              </ol>
            </div>
            <div class="clearfix"></div>
          </div>
          <!-- .title end -->
        </div>
        <!-- .col-md-12 end -->
      </div>
      <!-- .row end -->
    </div>

</section>
<!-- #page-title end -->

<!-- #Add Property
```

```
========================================== -->


    <section id="add-property" class="add-property">


      <div class="container">


        <div class="row">


          <div class="col-xs-12 col-sm-12 col-md-12">


            <form class="mb-0" method="post"  enctype="multipart/form-data">
              <p style="font-size:16px; color:red" align="center"> <?php if($msg){
 echo $msg;
 }  ?> </p>
              <div class="form-box">
                <div class="row">
                  <div class="col-xs-12 col-sm-12 col-md-12">
                    <h4 class="form--title">Property Description</h4>
                  </div>
                  <!-- .col-md-12 end -->
                  <div class="col-xs-12 col-sm-12 col-md-12">
                    <div class="form-group">
                      <label for="property-title">Property Title*</label>
                      <input type="text" class="form-control" name="propertytitle"
id="propertytitle" required>
                    </div>
```

```
                    </div>
                    <!-- .col-md-12 end -->
                    <div class="col-xs-12 col-sm-12 col-md-12">
                      <div class="form-group">
                        <label for="property-description">Property Description*</label>
                        <textarea class="form-control" name="propertydescription"
id="propertydescription" rows="2"></textarea>
                      </div>
                    </div>
                    <!-- .col-md-12 end -->
                    <div class="col-xs-12 col-sm-4 col-md-4">
                      <div class="form-group">
                        <label for="select-type">Type</label>
                        <div class="select--box">
                          <i class="fa fa-angle-down"></i>
                          <select id="selecttype" name="selecttype" required="true">
                        <option value="">Select Property Type</option>
        <?php $query1=mysqli_query($con,"select * from tblpropertytype");
        while($row1=mysqli_fetch_array($query1))
        {
        ?>
          <option value="<?php echo $row1['PropertType'];?>"><?php echo
$row1['PropertType'];?></option>
          <?php } ?>
                    </select>
                      </div>
```

```
                    </div>
                </div>
                <!-- .col-md-4 end -->
                <div class="col-xs-12 col-sm-4 col-md-4">
                    <div class="form-group">
                        <label for="select-status">Status</label>
                        <div class="select--box">
                            <i class="fa fa-angle-down"></i>
                            <select id="status" name="status">
                        <option>Sale</option>
                        <option>Rent</option>
                    </select>
                        </div>
                    </div>
                </div>
                <!-- .col-md-4 end -->
                <div class="col-xs-12 col-sm-4 col-md-4">
                    <div class="form-group">
                        <label for="location">Location</label>
                        <input type="text" class="form-control" name="location"
id="location">
                    </div>
                </div>
                <!-- .col-md-4 end -->
                <div class="col-xs-12 col-sm-4 col-md-4">
                    <div class="form-group">
```

```
                        <label for="Bedrooms">Bedrooms</label>
                        <input type="text" class="form-control" name="bedrooms"
id="bedrooms">
                    </div>
                </div>
                <!-- .col-md-4 end -->
                <div class="col-xs-12 col-sm-4 col-md-4">
                    <div class="form-group">
                        <label for="Bathrooms">Bathrooms</label>
                        <input type="text" class="form-control" name="bathrooms"
id="bathrooms">
                    </div>
                </div>
                <!-- .col-md-4 end -->
                <div class="col-xs-12 col-sm-4 col-md-4">
                    <div class="form-group">
                        <label for="Floors">Floors</label>
                        <input type="text" class="form-control" name="floors"
id="floors">
                    </div>
                </div>
                <!-- .col-md-4 end -->
                <div class="col-xs-12 col-sm-4 col-md-4">
                    <div class="form-group">
                        <label for="Garages">Garages</label>
                        <input type="text" class="form-control" name="garages"
id="garages">
```

```
                    </div>
                </div>
                <!-- .col-md-4 end -->
                <div class="col-xs-12 col-sm-4 col-md-4">
                    <div class="form-group">
                        <label for="Area">Area</label>
                        <input type="text" class="form-control" name="area" id="area"
placeholder="sq ft">
                    </div>
                </div>
                <!-- .col-md-4 end -->
                <div class="col-xs-12 col-sm-4 col-md-4">
                    <div class="form-group">
                        <label for="Size">Size</label>
                        <input type="text" class="form-control" name="size" id="size"
placeholder="sq ft">
                    </div>
                </div>
                <!-- .col-md-4 end -->
                <div class="col-xs-12 col-sm-4 col-md-4">
                    <div class="form-group">
                        <label for="Sale-Rent-Price">Sale or Rent Price*</label>
                        <input type="text" class="form-control" name="salerentprice"
id="salerentprice" required>
                    </div>
                </div>
                <!-- .col-md-4 end -->
```

```
                        <div class="col-xs-12 col-sm-4 col-md-4">
                          <div class="form-group">
                            <label for="Before-Price-Label">Before Price Label</label>
                            <input type="text" class="form-control" name="beforepricelabel"
id="beforepricelabel" placeholder="ex: start from">
                          </div>
                        </div>
                        <!-- .col-md-4 end -->
                        <div class="col-xs-12 col-sm-4 col-md-4">
                          <div class="form-group">
                            <label for="After-Price-Label">After Price Label</label>
                            <input type="text" class="form-control" name="afterpricelabel"
id="afterpricelabel" placeholder="ex: monthly">
                          </div>
                        </div>

                      </div>
                      <!-- .row end -->
                    </div>
                    <!-- .form-box end -->
                    <div class="form-box">
                      <div class="row">
                        <div class="col-xs-12 col-sm-12 col-md-12">
                          <h4 class="form--title">Property Features</h4>
                        </div>
                        <!-- .col-md-12 end -->
```

```
                    <div class="col-xs-12 col-sm-6 col-md-3">
                      <div class="input-checkbox">
                        <label class="label-checkbox">
                      <span>Center Cooling</span>
                      <input type="checkbox" name="centercolling" id="centercolling"
value="1">
                      <span class="check-indicator"></span>
                    </label>
                      </div>
                    </div>
                    <!-- .col-md-3 end -->
                    <div class="col-xs-12 col-sm-6 col-md-3">
                      <div class="input-checkbox">
                        <label class="label-checkbox">
                      <span>Balcony</span>
                      <input type="checkbox" name="balcony" id="balcony" value="1">
                      <span class="check-indicator"></span>
                    </label>
                      </div>
                    </div>
                    <!-- .col-md-3 end -->
                    <div class="col-xs-12 col-sm-6 col-md-3">
                      <div class="input-checkbox">
                        <label class="label-checkbox">
                      <span>Pet Friendly</span>
                      <input type="checkbox" name="petfrndly" id="petfrndly" value="1">
```

```
                    <span class="check-indicator"></span>
                  </label>
                    </div>
                  </div>
                  <!-- .col-md-3 end -->
                  <div class="col-xs-12 col-sm-6 col-md-3">
                    <div class="input-checkbox">
                      <label class="label-checkbox">
                    <span>Barbeque</span>
                    <input type="checkbox" name="barbeque" id="barbeque"
value="1">
                    <span class="check-indicator"></span>
                  </label>
                    </div>
                  </div>
                  <!-- .col-md-3 end -->
                  <div class="col-xs-12 col-sm-6 col-md-3">
                    <div class="input-checkbox">
                      <label class="label-checkbox">
                    <span>Fire Alarm</span>
                    <input type="checkbox" name="firealarm" id="firealarm" value="1">
                    <span class="check-indicator"></span>
                  </label>
                    </div>
                  </div>
                  <!-- .col-md-3 end -->
```

```
                    <div class="col-xs-12 col-sm-6 col-md-3">
                      <div class="input-checkbox">
                        <label class="label-checkbox">
                      <span>Modern Kitchen</span>
                      <input type="checkbox" name="modkitchen" id="modkitchen"
value="1">
                      <span class="check-indicator"></span>
                    </label>
                      </div>
                    </div>
                    <!-- .col-md-3 end -->
                    <div class="col-xs-12 col-sm-6 col-md-3">
                      <div class="input-checkbox">
                        <label class="label-checkbox">
                      <span>Storage</span>
                      <input type="checkbox" name="storage" id="storage" value="1">
                      <span class="check-indicator"></span>
                    </label>
                      </div>
                    </div>
                    <!-- .col-md-3 end -->
                    <div class="col-xs-12 col-sm-6 col-md-3">
                      <div class="input-checkbox">
                        <label class="label-checkbox">
                      <span>Dryer</span>
                      <input type="checkbox" name="dryer" id="dryer" value="1">
```

```
    <span class="check-indicator"></span>
</label>
    </div>
</div>
<!-- .col-md-3 end -->
<div class="col-xs-12 col-sm-6 col-md-3">
    <div class="input-checkbox">
        <label class="label-checkbox">
    <span>Heating</span>
    <input type="checkbox" name="heating" id="heating" value="1">
    <span class="check-indicator"></span>
</label>
    </div>
</div>
<!-- .col-md-3 end -->
<div class="col-xs-12 col-sm-6 col-md-3">
    <div class="input-checkbox">
        <label class="label-checkbox">
    <span>Pool</span>
    <input type="checkbox" name="pool" id="pool" value="1">
    <span class="check-indicator"></span>
</label>
    </div>
</div>
<!-- .col-md-3 end -->
<div class="col-xs-12 col-sm-6 col-md-3">
```

```
    <div class="input-checkbox">
      <label class="label-checkbox">
    <span>Laundry</span>
    <input type="checkbox" name="laundry" id="laundry" value="1">
    <span class="check-indicator"></span>
</label>
    </div>
</div>
<!-- .col-md-3 end -->
<div class="col-xs-12 col-sm-6 col-md-3">
    <div class="input-checkbox">
      <label class="label-checkbox">
    <span>Sauna</span>
    <input type="checkbox" name="sauna" id="sauna" value="1">
    <span class="check-indicator"></span>
</label>
    </div>
</div>
<!-- .col-md-3 end -->
<div class="col-xs-12 col-sm-6 col-md-3">
    <div class="input-checkbox">
      <label class="label-checkbox">
    <span>Gym</span>
    <input type="checkbox" name="gym" id="gym" value="1">
    <span class="check-indicator"></span>
</label>
```

```
                    </div>
                </div>
                <!-- .col-md-3 end -->
                <div class="col-xs-12 col-sm-6 col-md-3">
                    <div class="input-checkbox">
                        <label class="label-checkbox">
                    <span>Elevator</span>
                    <input type="checkbox" name="elevator" id="elevator" value="1">
                    <span class="check-indicator"></span>
                </label>
                    </div>
                </div>
                <!-- .col-md-3 end -->
                <div class="col-xs-12 col-sm-6 col-md-3">
                    <div class="input-checkbox">
                        <label class="label-checkbox">
                    <span>Dish Washer</span>
                    <input type="checkbox" name="dishwasher" id="dishwasher"
value="1">
                    <span class="check-indicator"></span>
                </label>
                    </div>
                </div>
                <!-- .col-md-3 end -->
                <div class="col-xs-12 col-sm-6 col-md-3">
                    <div class="input-checkbox">
```

```
                        <label class="label-checkbox">
                        <span>Emergency Exit</span>
                        <input type="checkbox" name="eexit" id="eexit" value="1">
                        <span class="check-indicator"></span>
                      </label>
                        </div>
                      </div>
                      <!-- .col-md-3 end -->
                    </div>
                    <!-- .row end -->
                  </div>
                  <!-- .form-box end -->

                  <div class="form-box">
                    <div class="row">
                      <div class="col-xs-12 col-sm-12 col-md-12">
                        <h4 class="form--title">Property Gallery</h4>
                      </div>
                      <!-- .col-md-12 end -->
                      <div class="col-xs-4 col-sm-4 col-md-4">
                        <div class="form-group">
                          <label for="address">Featured Image</label>
                          <input type="file" class="form-control" name="featuredimage"
required>
                        </div>
                      </div>
```

```
                    <div class="col-xs-4 col-sm-4 col-md-4">
                      <div class="form-group">
                        <label for="address">Gallery Image1</label>
                        <input type="file" class="form-control" name="galleryimage1"
required>
                      </div>
                    </div>
                    <div class="col-xs-4 col-sm-4 col-md-4">
                      <div class="form-group">
                        <label for="address">Gallery Image2</label>
                        <input type="file" class="form-control" name="galleryimage2"
required>
                      </div>
                    </div>
                    <div class="col-xs-4 col-sm-4 col-md-4">
                      <div class="form-group">
                        <label for="address">Gallery Image3</label>
                        <input type="file" class="form-control" name="galleryimage3"
required>
                      </div>
                    </div>
                    <div class="col-xs-4 col-sm-4 col-md-4">
                      <div class="form-group">
                        <label for="address">Gallery Image4</label>
                        <input type="file" class="form-control" name="galleryimage4"
required>
                      </div>
```

```
                    </div>
                    <div class="col-xs-4 col-sm-4 col-md-4">
                      <div class="form-group">
                        <label for="address">Gallery Image5</label>
                        <input type="file" class="form-control" name="galleryimage5"
required>
                      </div>
                    </div>
                    <!-- .col-md-12 end -->

                  </div>
                  <!-- .row end -->
                </div>
                <!-- .form-box end -->

                <div class="form-box">
                  <div class="row">
                    <div class="col-xs-12 col-sm-12 col-md-12">
                      <h4 class="form--title">Property Location</h4>
                    </div>
                    <!-- .col-md-12 end -->
                    <div class="col-xs-12 col-sm-4 col-md-4">
                      <div class="form-group">
                        <label for="address">Address*</label>
                        <input type="text" class="form-control" name="address"
id="address" placeholder="Enter your property address" required>
```

```
                    </div>
                  </div>
                  <!-- .col-md-4 end -->
                  <div class="col-xs-12 col-sm-4 col-md-4">
                    <div class="form-group">
                      <label for="select-country">Country</label>
                      <div class="select--box">
                        <i class="fa fa-angle-down"></i>
                   <select type="text" name="country" id="country" required="true"
onChange="getsate(this.value)" class="form-control">
                        <option value="">Select Country</option>
          <?php $query=mysqli_query($con,"select * from tblcountry");
          while($row=mysqli_fetch_array($query))
          {
          ?>
            <option value="<?php echo $row['ID'];?>"><?php echo
$row['CountryName'];?></option>
            <?php } ?>
                    </select>
                    </div>
                  </div>
                  </div>

        <div class="col-xs-12 col-sm-4 col-md-4">
                    <div class="form-group">
```

```
                        <label for="state">State</label>
                        <div class="select--box">
                          <i class="fa fa-angle-down"></i>
                        <select type="text" class="form-control" name="state" id="state"
onChange="getcity(this.value)" >
                        </select>
                    </div>
                </div>
                </div>


                <!-- .col-md-4 end -->
                <div class="col-xs-12 col-sm-4 col-md-4">
                    <div class="form-group">
                        <label for="city">City</label>
                        <div class="select--box">
                          <i class="fa fa-angle-down"></i>
                        <select class="form-control" name="city" id="city">
                        </select>
                    </div>
                    </div>
                </div>
                <!-- .col-md-4 end -->

                <!-- .col-md-4 end -->
                <div class="col-xs-12 col-sm-4 col-md-4">
```

```
                        <div class="form-group">
                            <label for="Zip/Postal-code">Zip/Postal Code</label>
                            <input type="text" class="form-control" name="zipcode"
id="zipcode">
                        </div>
                    </div>
                    <!-- .col-md-4 end -->
                    <div class="col-xs-12 col-sm-4 col-md-4">
                        <div class="form-group">
                            <label for="neighborhood">Neighborhood</label>
                            <input type="text" class="form-control" name="neighborhood"
id="neighborhood">
                        </div>
                    </div>
                    <!-- .col-md-4 end -->


                    <!-- .col-md-12 end -->
                </div>
                <!-- .row end -->
            </div>
            <!-- .form-box end -->
            <input type="submit" value="Submit" name="submit" class="btn btn--
primary">
        </form>
    </div>
    <!-- .col-md-12 end -->
</div>
```

```
            <!-- .row end -->
          </div>
        </section>
              </div>
              <!-- /.navbar-collapse -->
            </div>
            <!-- /.container-fluid -->
          </nav>

      </header>
      <!-- Footer #1
================================================== -->
      <?php include_once('includes/footer.php');?>
    </div>
    <!-- #wrapper end -->
    <!-- Footer Scripts
================================================== -->
    <script src="assets/js/jquery-2.2.4.min.js"></script>
    <script src="assets/js/plugins.js"></script>
    <script src="assets/js/functions.js"></script>
</body>

</html>
 <?php } ?>
```

8. TESTING

Testing:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Testing Types:

Unit testing: Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration Testing: Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test: Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals Functional testing is centred on the following items: Valid Input: Identified classes of valid input must be accepted.

Invalid Input: Identified classes of invalid input must be rejected. Functions: Identified functions must be exercised.

Output: Identified classes of application outputs must be exercised.

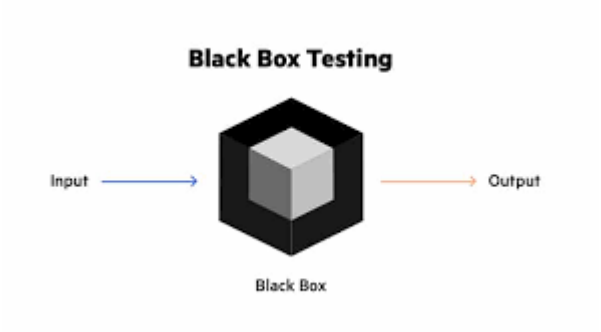
Systems/Procedures: Interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test: System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing: White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing:

Black box testing is testing software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of texts, must be written from a definitive source document, such as specification or requirement document. It is a testing in which the software under test is treated, as a black box. The test provides inputs and responds to outputs without considering how the software works.



This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

- Incorrect or missing functions.
- Interface errors.
- Errors in data structures or external database access.
- Behavior or performance errors.
- Initialization and termination errors.

System Implementation:

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Security and Maintenance:

All the details can be viewed by admin, who is also responsible for managing other users. Much of the human intervention will be eliminated which will enhance security, integrity and reduce error rates.

Test Cases:

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed

Integration Testing: Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g., components in a software system or-one step up-software applications at the company level-interact without error.

Acceptance Testing: User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

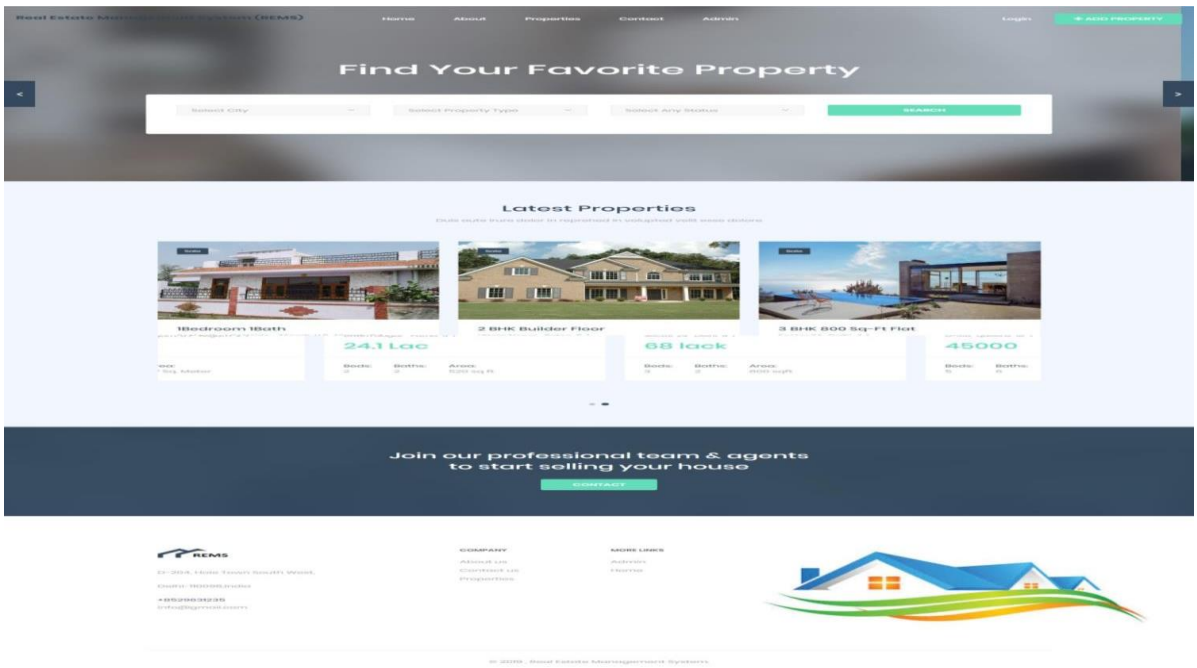
Test Results: All the test cases mentioned above passed successfully. No defects encountered

9.FORM LAYOUT

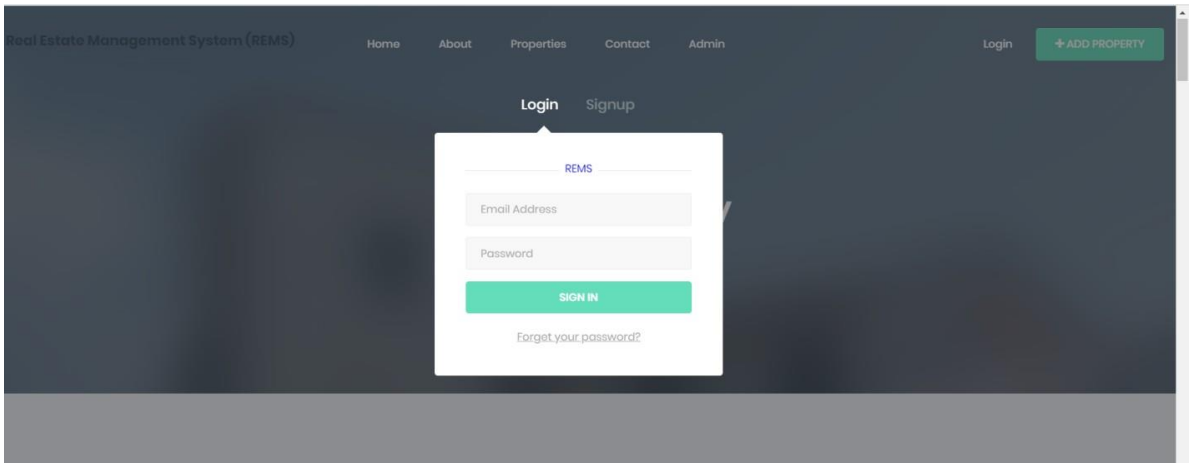
The project is compiled and executed on chrome. Some screen shots are present here to show the working of the application.

Screenshots: The below figure refers to the initial bootup page of the application

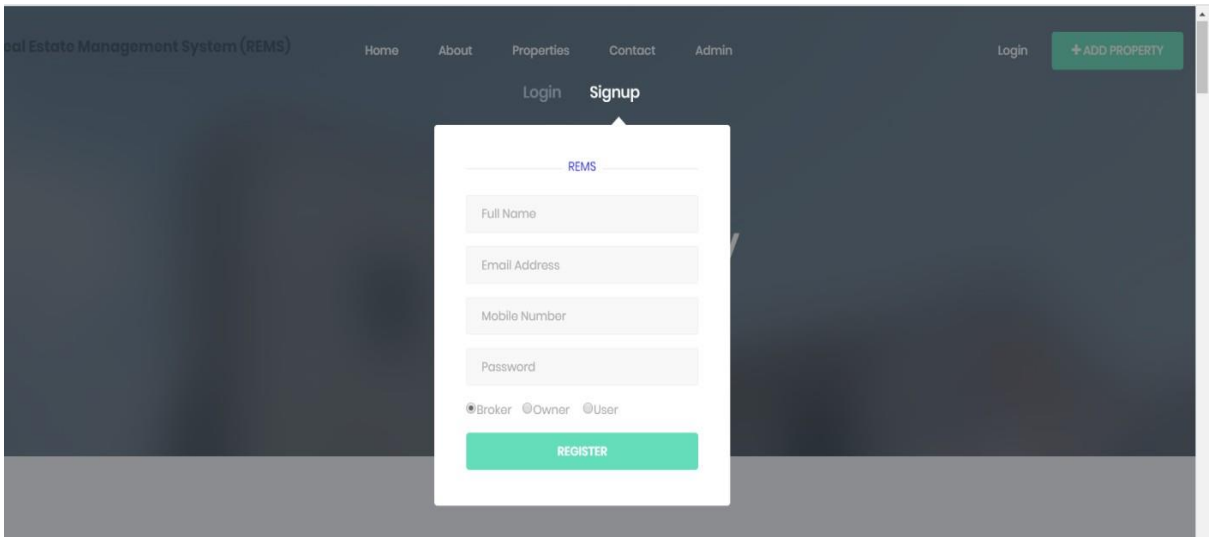
Home page:



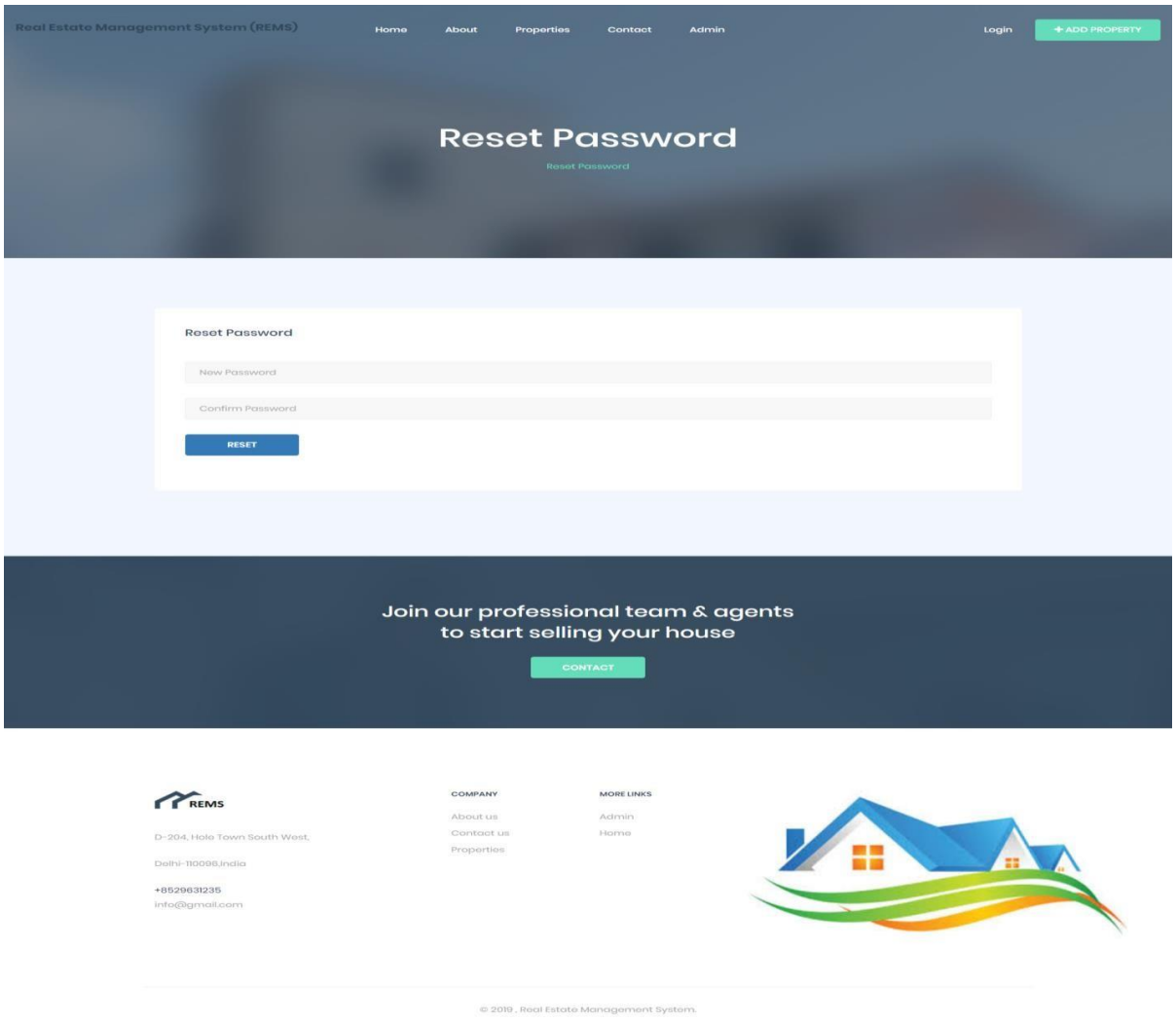
Login page:



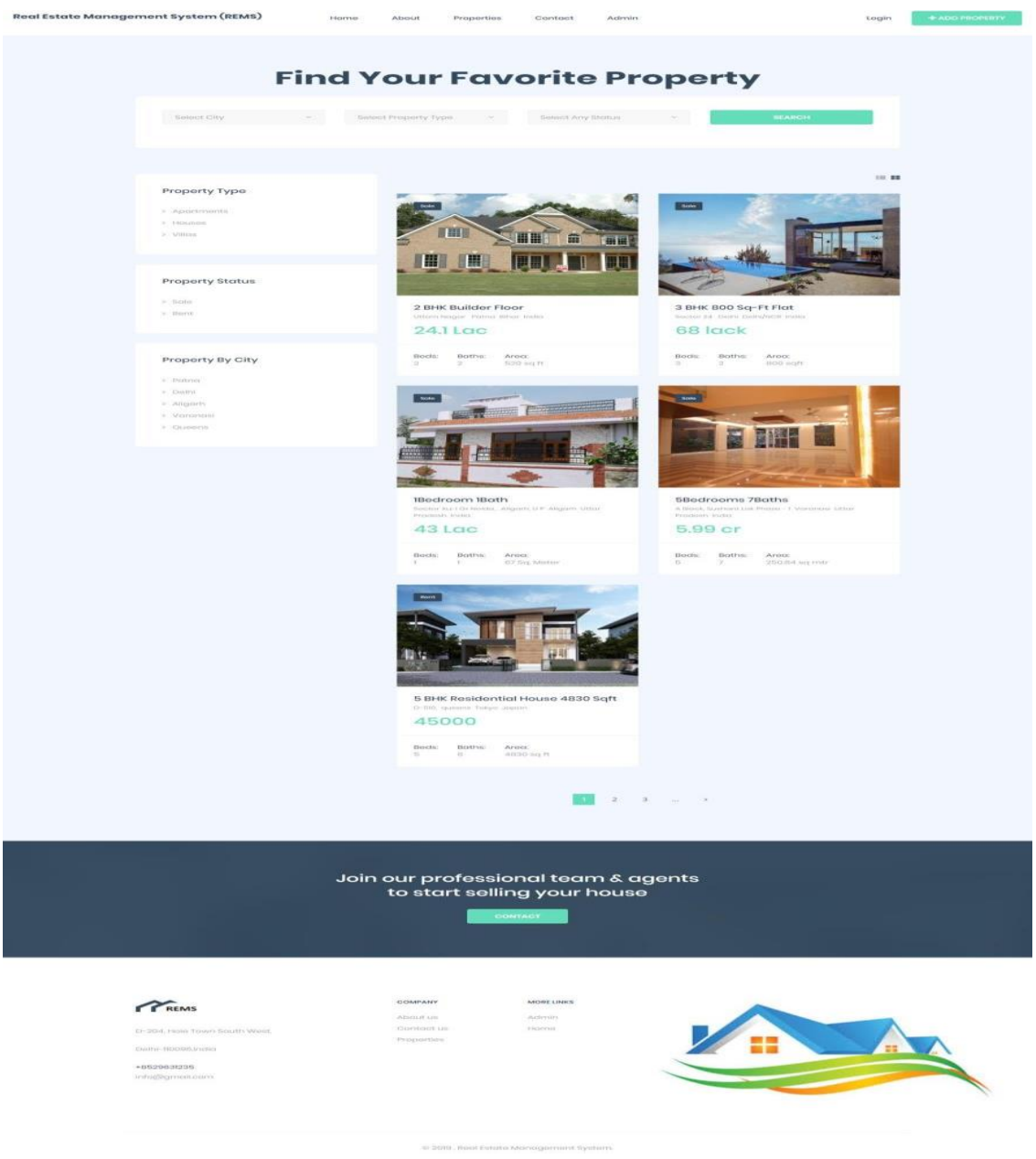
Registration page:



Change password:



Property:


Real Estate Management System (REMS)
Home
About
Properties
Contact
Admin
Login
+ ADD PROPERTY
Find Your Favorite Property
Select City
Select Property Type
Select Any Status
SEARCH
Property Type
Apartments
Houses
Villas
Property Status
Sale
Rent
Property By City
Patna
Delhi
Aligarh
Varanasi
Queens
2 BHK Builder Floor
24.1 Lac
3 BHK 800 Sq-Ft Flat
68 lack
1Bedroom 1Bath
43 Lac
5Bedrooms 7Baths
5.99 cr
5 BHK Residential House 4830 Sqft
45000
Join our professional team & agents
to start selling your house
CONTACT
REMS
COMPANY
About Us
Contact Us
Properties
MORE LINKS
Admin
Home

Mortgage Calculator:

Calculator

Sale Price	200000
Down Payment	80000
Actual Payment	120000
Term	3
Interest Rate	12
Interest	14400
Actual Payment + Interest	134400
EMI	3733.3333333333

10.Conclusion and Future Enhancement

The system is very useful for the companies or builders that can post and edit their properties and their personal info and admin can monitor records of all of them. The system is also useful which also keeps track of Account details of buyers and Investors and also RES Industry.

This project can be further enhanced as follows:

a. Minimizing the hardware and software requirement so that it supports maximum user base. b. Modifying the project with better approach, with more graphics.

c. The Users can easily communicate with the service providers, in case of any problem or enquiries.

d. Security could be heightened.

11.BIBLIOGRAPHY

For the development of our project there has been a few reference and sites we visited and are listed below: For the development of our project there has been a few reference and sites we visited and are listed below:

Books:

1. The Complete Reference Visual Basic.Net 2002 Edition by JefferyR.Shipario published by TateMcGraw-Hill.

2. Visual Basic .Net Programming 2002 Edition by PeterAikten published by Dream Tech NewDelhi.

3. Programming Microsoft ASP.NET 3.5" edition by Dino Esposito.

4. Visual Basic.net Programming Black Book 2005 Edition by StevenHolzer published by Para glyphPress USA.

WEB REFERENCES:

1. http://www.vbdotnetheaven.com/UploadFile/Report201242006041121AM/Report2

2. http://www.devarticles.com/c/a/VB.Net/Regular-Expressions-in.NET/1/

3. http://www.tutorialspoint.com/sql/

4. http://www.saisecurity.in/manual-guarding.html\

5. http://www.devarticles.com/c/a/VB.Net/Regular-Expressions-in.NET/1/