Khushal Khan Khattak University Karak

MY Estate Agency

This project report submitted is partial fulfilment of the requirements for the degree of

BS Computer Science

Submitted by:

1. Mr. Yasir Shahzad
2. Mr. Mussadiq Farid

Supervised by:

Mr. SHAD MUHAMMAD

Department of Computer Science and Bio Informatics

Khushal Khan Khattak University, Karak (KKKUK)

Session 2016-2020

With The Name of Almighty ALLAH, The Most Merciful

PROJECT APPROVAL

This is to certify that we have to read and checked the project report by Yasir Shahzad and Mussadiq Farid.

This project is hereby approved and recommended as partial fulfilment for the award of BS Computer Science from Khushal Khan Khattak University Karak (KKKUK).

External Examiner

Signature:

Name:

Designation:

Institute:

Internal Examiner

Signature:

Name: Shad Muhammad

Designation: Lecturer Computer Science

Institute: Khushal Khan Khattak University Karak

Head of Department

Signature:

Name: Dr. Muhammad Inam Ul Haq

Department: Department of CS&BI

Institute: Khushal Khan Khattak University Karak.

DEDICATION

---------Respected Parents, Teachers & Friends----------

With love and gratitude

May Allah bless them with a very happy, successful and a healthy life?

(AMIN)

Acknowledgement

In the name of Almighty **Allah,** the most Gracious and the Merciful. No words to describe a respect for our parents, because without their love and support it would have been impossible to attain this target. We acknowledge the support and inspiration of our brothers and sisters throughout our studies. We also acknowledge the guidance and supervision of our teachers who gave us knowledge. We are especially grateful and thankful to our supervisor, **Mr. Shad Muhammad** for his guidance and supervision to complete this work. We are very thankful to him for giving time and expertise. His suggestions, time-to-time help and backing and careful attention were valuable/treasured for us.

Abstract

After studying the existing manual system, it was found that the existing system was totally paper based resulting in huge time and paper consumption. MY Estate Agency is developed in order to replace the existing manual system. The MY Estate Agency is intended to fulfill the requirements of the Properties for selling and purchasing. It provides online access to the users and admin through secured login system. The MY Estate Agency is web based application connecting all of the users and allows online access to all properties.

The Admin will be responsible and having privileges to add, update and delete the users, and properties. The application will help in selling and purchasing of properties, and will also improve the overall performance.

Project in Brief

Project Title:	MY Estate Agency
Developed by:	YASIR SHAHZAD MUSSADIQ FARID
Supervised by:	Mr. Shad Muhammad
Tools Used:	PHP, HTML, CSS, JavaScript, JQuery and My SQL
System Used:	Desktop System or Laptop
Operation System:	Microsoft Windows 7,8 & 10 etc.

Table of Contents

List of Figures

CHAPTER

1

INTRODUCTION

Chapter-1: Introduction

1.1. Introduction

The MY Estate Agency is web based application connecting all of the users and allows online access to all properties. It provides online access to the users and admin through secured login system. The MY Estate Agency is intended to fulfill the requirements of the Properties for selling and purchasing. The application will help in selling and purchasing of properties, and will also improve the overall performance.

In order to accommodate and satisfy for some of the problems faced due to manual processing, we proposed a project to develop an automated solution for the Estate Agency system two of the sections of this proposed system i.e.

- Administrator section.
- User section.

Our project is a database based system as this is the most appropriate form of automation for MY Estate Agency and storing consistent data. We have provided proper security mechanism in order to forbid unauthorized access to the system. Only authorized user can have logged in to the system. An admin of a section is the most privileged user who can perform most of the operations.

The system stores the necessary information about each Users and the properties. Every user will have his own profile. Each user will be able to create a profile by entering all the correct information relevant to him. In his profile a user can see all his record, his information, and also every property details.

Different software and tools were used during the completion of project, which include: JAVASCRIPT, JQuery, MYSQL, Sublime/notepad++ and PHP.

1.2 Purpose

A MY Estate Agency is web application, used to perform all the necessary actions which are required for selling of property. This application deals with most of the issues that may arise in a selling of property. This type of applications might be helpful to reduce the time and human strain. We hope the implementation of this system will find an efficient solution for the selling of property.

1.3 Literature Review

For the development of the Web applications we study PHP language and MySQL from different books. Internet was a great help as many good sites were available to take help from. We were familiar with PHP and MySQL as we studied it in our course.

One good aspect of our project was to give due consideration to understand the problem. We spent more than half of our project's time in gathering requirement so that to fully understand the system. We study CSS, Bootstrap to make interactive design of our project.

Before the development process we study software engineering, its phases and project management through different book and related internet searches. It is for the purpose to manage our project precisely and control the flow of data in easy way. The literature we followed for our project is as under:

1.4 Objectives of the project

To solve the problems, it is suggested that a computer-based system will be the best choice and by replacing the existing system with computer-based system, this system will be more efficient, fast, reliable and user friendly.

- Cast Effective
- Vast Reach
- Minimizing Risk
- Maximizing Efficiency
- Promoting a Service Online
- Selling a Service
- Providing Customer support
- Providing Corporate Information
- Establishing brand awareness and Corporate identity

1.5 Gantt Chart for Project Development

Following grant chart shows the sequence of phases and progress of project graphically.

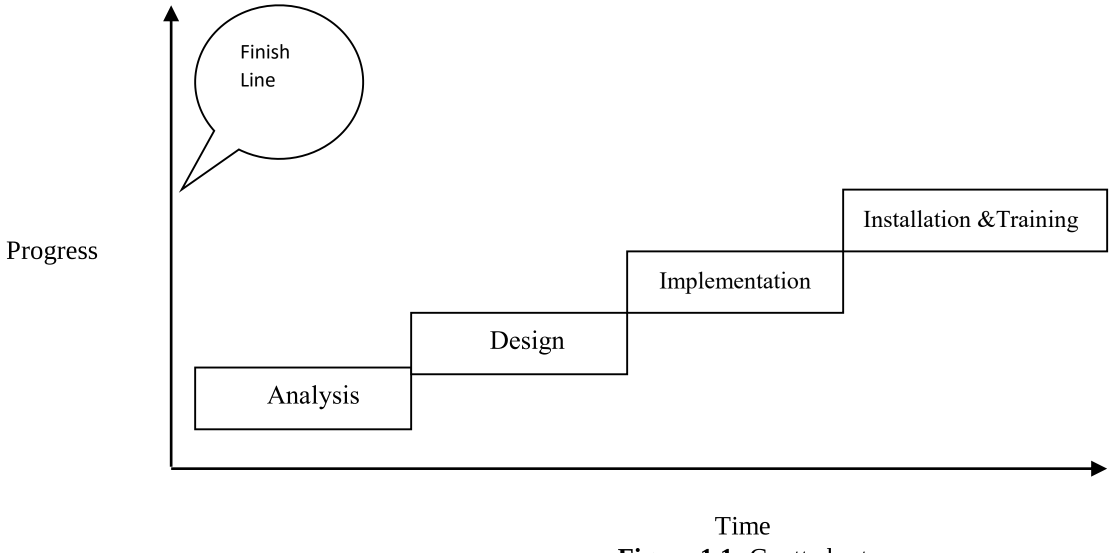


Figure 1.1: Gantt chart

1.6 Existing System

The study of the existing system is necessary because in this way, we become familiar with the existing system and we come to know how it works and what modifications do we want to make in the existing system. Keeping this in mind, we used different tools and techniques of information gathering. The techniques which we used for our analysis are as follows:

- Review of the literature, procedure and forms
- On site observation

- Interviewing
- Questionnaires

1.7 Software engineering

Let's look at definition given by Ian Somerville. "Software Engineering is not just concerned with the technical processes of software development but also with the activities such as software project management and with the development of tools, methods and theories to support software production"

Software engineering is set of processes and tools to develop software. Software engineering is combination of all the tools, techniques and processes that used in software productions. Therefore, software engineering encompasses all those things that are used in software production like:

- Programming Language
- Programming Language Design
- Software Design Techniques
- Tools
- Testing
- Maintenance
- Development etc.

So all those thing that are related to software are also related to software engineering. We have also studied about well-engineered software from software related books i.e. what will be the characteristics of acceptable software? It has the following characteristics.

- It is reliable
- It has good user-interface (User friendly)
- It has acceptable performance (efficient)
- It is of good quality
- It is cost-effective
- Provide the required functionality
- Maintainable

1.8 Software engineering phases

There are four fundamental phases in most. These phases are:

- Analysis
- Design
- Implementation
- Testing.

These phases address what is to be built, how it will be built, building it, and making it high quality.

1.9 Analysis

The Analysis phase defines the requirements of the system, independent of how these requirements will be accomplished. This phase defines the problem that the customer is trying to solve. The deliverable result at the end of this phase is a requirement document. Ideally, this document states in a clear and precise fashion what is to be built. This analysis represents the “what” phase. The requirement document tries to capture the requirements from the customer's perspective by defining goals and interactions at a level removed from the implementation details.

The requirement document may be expressed in a formal language based on mathematical logic. Traditionally, the requirement document is written in English or another written language. The requirement document does not specify the architectural or implementation details, but specifies information at the higher level of description. The problem statement, the customer's expectations, and the criteria for success are examples of high-level descriptions.

The requirement document should state in a clear and precise fashion what is to be built. The conclusive mechanism to author such a document, either formally or informally, has yet to be developed, although reasonable success has been achieved with existing methods including **CASE** tools and tools based on mathematical logic. The analysis team develops the requirement document, which talks about things and actions on things. This document should also include states, events, typical scenarios of usage, and typical scenarios of usage.

1.10 Design

In the design phase the architecture is established. This phase starts with the requirement document delivered by the requirement phase and maps the requirements into architecture. The architecture defines the components, their interfaces and behaviors. The deliverable design document is the architecture. The design document describes a plan to implement the requirements. This phase represents the ``**how**'' phase.

The architectural team can now expand upon the information established in the requirement document. Using the typical and a typical scenarios provided from the requirement document, performance trade-offs can be accomplished as well as complexity of implementation trade-offs.

The design activity is often divided into two separate *phase-system design* and *detailed design*. System design, which is sometimes also called top-level design, aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of system design all the major data structures, file formats, output formats, as well as the major modules in the system and their specifications are decided. During detailed design the internal logic of each of the modules specified in system design is decided. During this phase further details of the data structures and algorithmic design of each of the modules is specified. The logic of a module is usually specified in a high-level design description language, which is independent of the target language in which the software will eventually be implemented.

During the design phase two separate documents are produced, one for the system design and other for the detailed design. Together these documents completely specify design of the system.

1.11 Implementation

In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability guideline. The architecture document should give guidance. Sometimes, this guidance is found in the requirement document.

The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging. The end deliverable is the product itself.

1.12 Testing

In many software engineering methodologies, the testing phase is a separate phase which is performed by a different team after the implementation is completed. There is merit in this approach; it is hard to see one's own mistakes, and a fresh eye can discover obvious errors much faster than the person who has read and re-read the material many times. Unfortunately, delegating testing to another team leads to a slack attitude regarding quality by the implementation team.

The testing technique is from the perspective of the system provider. Because it is nearly impossible to duplicate every possible customer's environment and because systems are released with yet-to-be-discovered errors, the customer plays an important, though reluctant, role in testing.

1.13 Use cases

A use case is a description of how users will perform tasks on your Web site. A use case includes two main parts:

- the steps a user will take to accomplish a particular task on your site
- the way the Web site should respond to a user's actions

A use case begins with a user's goal and ends when that goal is fulfilled. A use case describes a sequence of interactions between a user and a Web site, without specifying the user interface. Each use case captures:

- The actor (who is using the System?)
- The interaction (what does the user want to do?)
- The goal (what is the user's goal?)

Generally, you write the steps in a use case in an easy-to-understand narrative. This engages members of the design team and encourages them to be actively involved in defining the requirements.

1. Identify who is going to be using the Web site.
2. Pick one of those actors.
3. Define what that actor wants to do on the site. Each thing the actor does on the site becomes a use case.
4. For each use case, decide on the normal course of events when that actor is using the site.
5. Describe the basic course in the description for the use case. Describe it in terms of what the actor does and what the system does in response that the actor should be aware of.
6. When the basic course is described, consider alternate courses of events and add those to "extend" the use case.

Repeat the steps 2 through 6 for all other actor.

CHAPTER

2

SOFTWARE REQIUEREMNT SPECIFICATION

2.1 Introduction

There are two main Modules in our system i.e. (Admin and user). These requirements were collected in various meetings that we did with the related stakeholders of the above-mentioned sections.

This document contains purpose, general description, constraint, functional requirements, non-functional requirements security requirements and change request for requirements. This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. All these parts are explained in brief below.

2.2 Purpose

This SRS describes the software functional and nonfunctional requirements for the release of MY Estate Agency. This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here are high priority and committed for the release of the specified system. MY Estate Agency is made for the users for selling and purchasing of property using web. This document is prepared for the guidance of the development of the project.

It went through the several stages during the course of project. These stages have been described below.

2.2.1 Draft

It was the first version. It was compiled after requirements have were gathered, classified and prioritized.

2.2.2 Proposed

The draft document was then proposed as a potential requirements specification for the project. The proposed documents were reviewed by admin. They comment on these requirements and identify missing requirements several times. This document was re-proposed several times before moving to next stage.

2.2.3 Validation

Once the admin was agreed to the requirement in the document it was considered validated.

2.2.4 Approved

The validation was accepted by admin or end users, as an appropriate statement of requirement for the project. We then used these requirements as a guide to develop our project.

2.3 Project Scope and Product Features

MY Estate Agency is useful for the users. In this System Admin can add/edit/update the users and properties as well. This system has separate registration forms for users, users can view its information’s, properties and the messages of the other users. The system contains the properties for the users who wants to purchase it, or want to sell his property. Admin can add/edit/delete the user details such as login name, password etc.

2.4 Product Perspective

The MY Estate Agency is a new system to provides the fast and reliable functioning for selling and purchasing of property. The system is expected to evolve over several releases, ultimately connecting to the Local Area Network (LAN) for managing and maintaining the MY Estate Agency.

The proposed system can be accessed by two types of user's i.e.

i. Admin
ii. Users

We discuss these two types of users one by one.

2.4.1 Admin

The admin are the end users of the project. They have access to the database so they can save, edit, update and delete data. Admin will insert all the personal information of a newly registered users. The software will also able the Admin to delete and update the user record. In short Admin is able to access all related information relevant to a user. Admin will insert all the property information to save it in Database. The software will also able the Admin to delete and update the property. In short Admin is able to access all related information relevant to a property. Admin will be given a security password i.e. the admin after entering the password will be able to access the database otherwise they will be counted as a general. The admin can also change their own section password to improve their security.

2.4.2 Users

The other type are users. They are actually the visitors of the system through web interface. Every user will be given his own profile which contains his personal information, property related information and messages. These users can change their personal data i.e. the system will give them access to database.

General constraints

The management of this project has prescribed that

- All software development on this project will be done according to the standard for software engineering.
- The language used in the deliverables is English. This means the source code (including comments), documentation, and user interface will be delivered to the client in English.

Because the development team working on this project will not be available after the project is completed, the client should be able to maintain the product without help of this our development team. Therefore, the complete source code and documentation generated during this project is given to the client at the end of the project.

2.5 Functional Requirements

label	Requirements
1	The system will save the personal information of each user inserted by Admin / User The personal information's are i. Name ii. Father Name iii. Email iv. Password v. Mobile Number vi. CNIC Number vii. Address viii. Image
2	The system will save the property Details inserted by Admin The property Details are i. Area ii. Price iii. location iv. Property Type v. Status vi. Images vii. description viii. Date
3	The system shall Generate appropriate Information for user
4	The system will allow user to view information's.
5	The system will allow user to edit his profile
6	The system will also facilitate the novice user.
7	The system will allow the users to see the personal information's.
8	The system will allow admin to view messages
9	The system shall allow admin to Manage users i.e. enter new user, delete user, update user.
10	The system will allow admin to add new property, which further allow him to delete, update the property.
11	The system should be powerful enough to let the admin manipulate the messages and credentials.
12	The system will Manage the security issues through proper user login for all the users and admin.

13	System should allow only the mighty admin to add new admin to keep the system secure.
14	Admin can register users and provide them the login details with activation level approved
15	Also the user can register by their own, and the admin can DE active it. So that he can't login to his account.

2.6 Security Requirements

- The software allows each specific Module to his own section by entering the specific password.
- The software should not allow any other user to log into the Admin Module sections.
- User will be required to log in to their own profiles. They cannot have access to Admin section.
- The software will provide separate password and login details for each section.
- The system will allow the Admin to change their password of their relevant section.

2.7 Non Functional requirements

Non Functional requirements are those requirements that describe how the system is supposed to be. In this case following are the non-functional requirements

2.7.1 Usability

The system will be simple and will have easy user interface. Users will be able to understand and use the system easily. For example, user want to sign up than he can just click on sign up button and a friendly form will appear and user will input the following requirements for registration.

2.7.2 Efficiency

The system performance will be fast and will not take more than ten seconds to response. The processing will be fast and efficient. The user can upload delete data on the system so efficiently.

2.7.3 Availability

The system will be available for the Users every time and they will be able to access from everywhere easily.

2.7.4 Adaptability

The system will be adaptable and making changes in it will be easy the system will adapt the changes done by users or admin.

2.7.5 Training requirement

The system is so user friendly that the users can use it without any training and developers help.

Some of the Nonfunctional requirements are:

- Admin and Users will be able to access their account any time.
- Unauthorized access to the system and its data will be not allowed
- The system will not operate in the absence of electricity.
- The website can be accessed on other Operating System (OS) too.
- MY Estate Agency logo will be used on every page.

CHAPTER

3

SYSTEM ANALYSIS

3.1 System Analysis

Analysis can be defined as breaking up of any whole so as to find out their nature, function etc. It defines design as to make preliminary sketches of; to sketch a pattern or outline for plan. To plan and carry out especially by artistic arrangement or in a skill full wall. System analysis and design can be characterized as a set of techniques and processes, a community of interests, a culture and an intellectual orientation

The various tasks in the system analysis include the following:

- Understanding application.
- Planning
- Scheduling
- Developing candidate solution.
- Performing trade studies.
- Performing cost benefit analysis.
- Recommending alternative solutions.
- Selling of the system.
- Supervising, installing and maintaining the system

3.2 PROPOSED SYSTEM

A MY Estate Agency is web application, used to perform all the necessary actions which are required for selling and purchasing of property. This application deals with most of the issues that may arise in a selling and purchasing of property. This type of applications might be helpful to reduce the time and human strain. We hope the implementation of this system will find an efficient solution for the selling and purchasing of property.

3.2.1 Advantages of Proposed System

- It is trouble-free to use
- It is a relatively fast
- Is highly reliable, approximate result from user
- Best user Interface
- Efficient reports
- Planning

3.3 FEASIBILITY STUDY

Feasibility analysis begins once the goals are defined. It starts by generating broad possible solutions, which are possible to give an indication of what the new system should look like. This is where creativity and imagination are used. Analysts must think up new ways of doing things-generate new ideas. There is no need to go into the detailed system operation yet. The solution should provide enough information to make reasonable estimates about project cost and give users an indication of how the new system will fit into the organization. It is important not to exert considerable effort at this stage only to find out that the project is not worthwhile or that there is a need significantly change the original goal. Feasibility of a new system means ensuring that the new system, which we are going to implement, is efficient and affordable. There are various types of feasibility to be determined. They are;

3.3.1 Economically Feasibility

Development of this application is highly economically feasible. The only thing to be done is making an environment with an effective supervision. It is cost effective in the sense that has eliminated the movement to Agents or other things completely. The system is also time effective because the user will only post his property detail or look for the property on this application, which are made as per the user requirement.

3.3.2 Technical feasibility

The technical requirement for the system is economic and it does not use any other additional Hardware and software. Technical evaluation must also assess whether the existing systems can be upgraded to use the new technology and whether the organization has the expertise to use it. Install all upgrades framework into the .Net package supported windows based application. this application depends on intranet service, database. Enter their records/information’s and generate report.

3.3.3 Operational Feasibility

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system. Technical performance includes issues such as determining whether the system can provide the right information for the personnel user details, and whether the system can be organized so that it always delivers this information at the right place and on time using intranet services. Acceptance revolves around the current system and its personnel.

CHAPTER

4

SYSTEM DESIGN

4.1 System Design

The designing is the basic building block of any software. The design depends upon the study and understanding of the present system and the vision and approach of the software designer for the proposed system. The design can be divided into three parts:

- Database Design
- Input Design
- Output Design

In the designing phase of MY Estate Agency, the activities included many important designing aspects, such as follows.

4.2 Developing a logical data model

This is probably the most important part of the designing phase. No matter, how small the project, is an entity-relationship diagram should be created using a tool such as the Entity-relationship diagrammed in Designer/2000 or Logic Work's Erwin or Workbench. These tools can help in creating the logical model which generate the database and document of data being stored in the database.

4.3 Implementing the functionality in stages

It is important to have various "releases" of the product so users can have a preview of it, and as well as make suggestions about whether it has the features that they were expecting or not. The designing phase should consider this sequential development and identify the core objects.

4.4 Using a configuration management tool

A configuration management tool should be used to perform various version controls and a backup/recovery strategy should be decided on:

Identifying users, choosing a consistent user interface and Determining a testing plan.

4.5 Logical Database design:

Logical database design is the process of transforming (or mapping) a conceptual schema of the application domain into a schema for the data model underlying a particular DBMS, such as the relational or object-oriented data model.

This mapping can be understood as the result of trying to achieve two distinct sets of goals:

(i) Representation goal: preserving the ability to capture and distinguish all valid states of the conceptual schema;

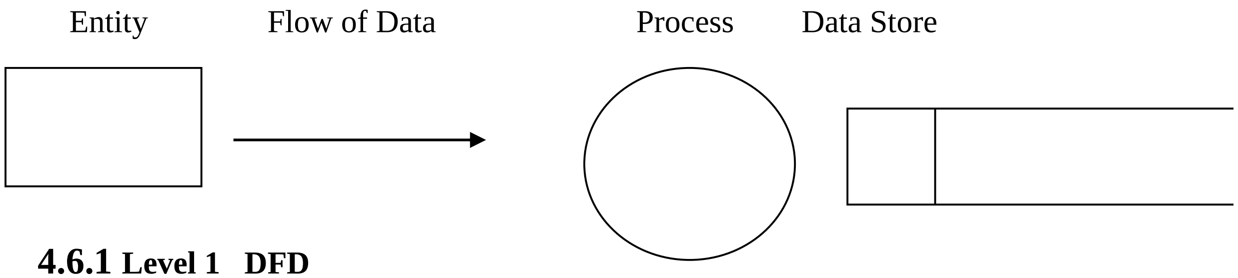
(ii) Data management goals: addressing issues related to the ease and cost of querying the logical schema, as well as costs of storage and constraint maintenance. This entry focuses mostly on the mapping of (Extended) Entity-Relationship (EER) diagrams to relational databases.

4.6 Data Flow Diagram

Data Flow Diagrams (DFDs) graphically characterize data process and flows in a business system. Here data flow diagrams have been used to depict the broadest possible overview of system inputs, process and outputs. Also a series of layered data flow diagrams have been used to represent and analyse detailed procedures within this large system. Therefore, by using combinations of only four symbols, we have created a pictorial depiction of process that will eventually provide solid system concepts and documentation.

The four basic symbols used in the data flow diagram are as under:

Basic Symbols:



4.6.1 Level 1 DFD

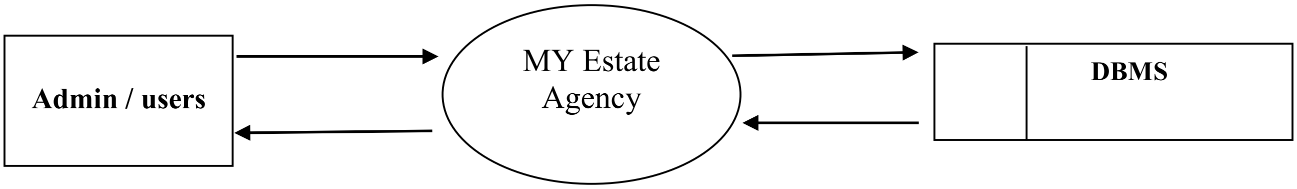


Figure 4.6.1 Level 1 DFD

4.6.2 Level 2 DFD

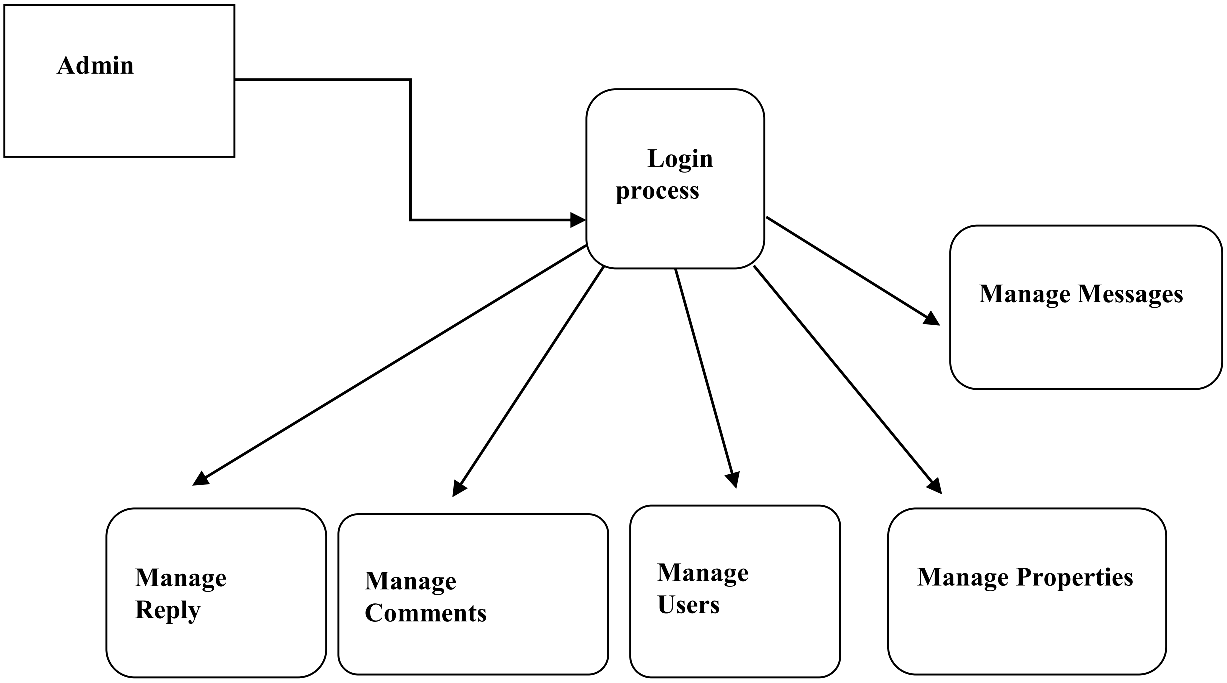


Figure 4.6.2 Level 2 DFD

4.6.3 Level 3 DFD

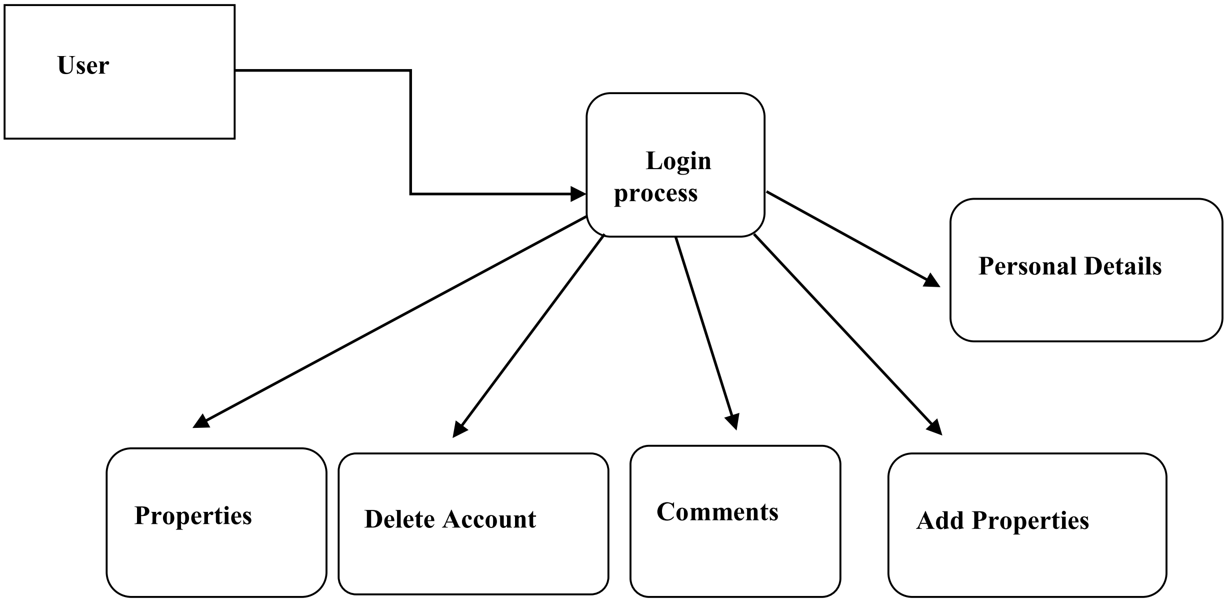


Figure 4.6.3 Level 2 DFD

4.6.4 User Login DFD

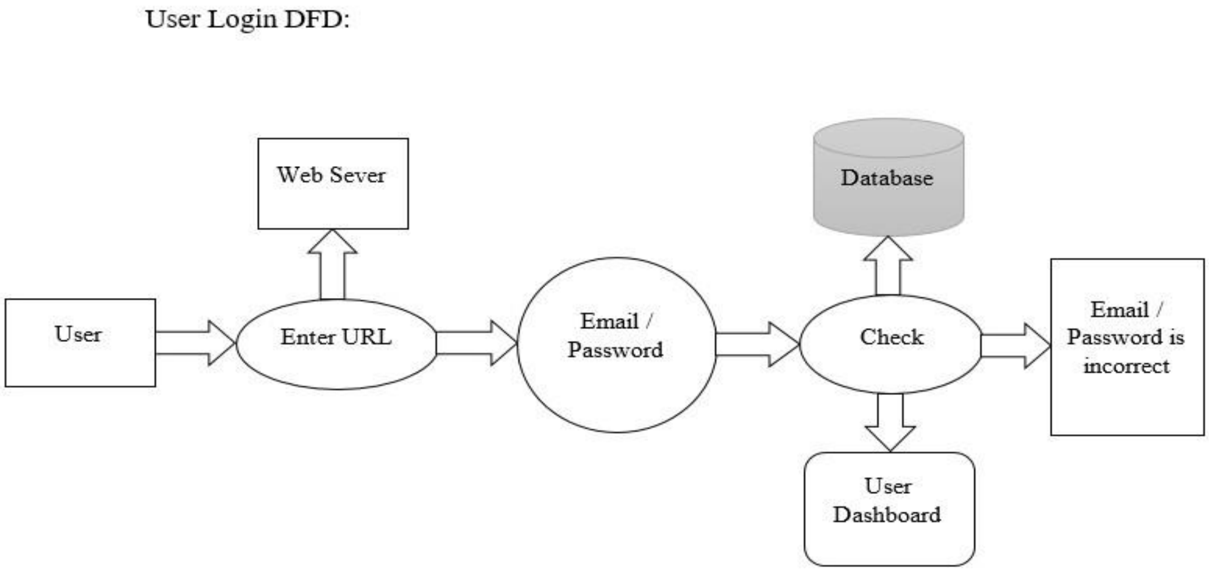


Figure 4.6.4 User Login DFD

User select login in option where they are asked to enter Email and password, and if he/ she is valid User then a User dashboard will be displayed.

4.6.5 Admin Login DFD

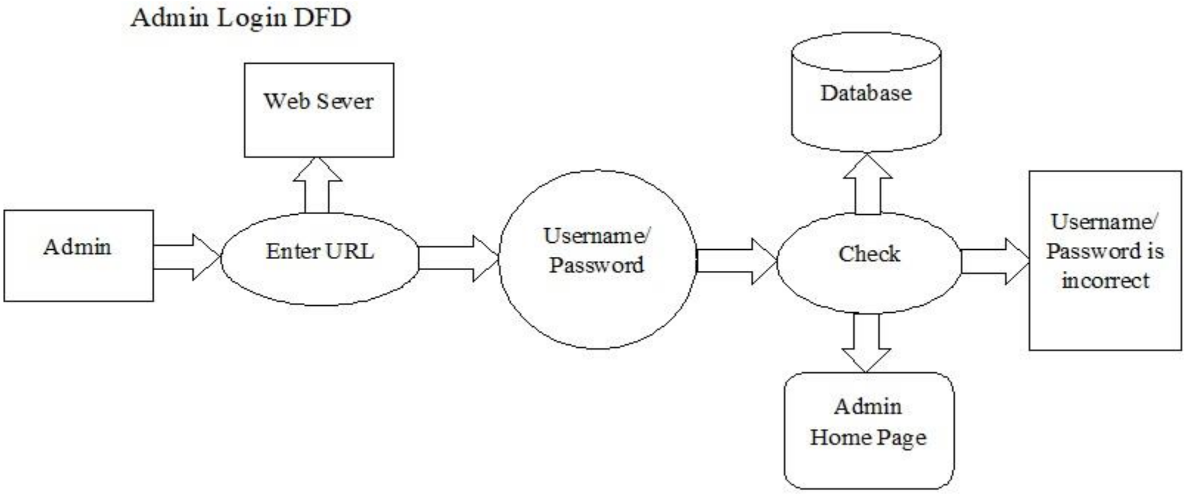


Figure 4.6.5 Admin Login DFD

Admin select login in option where they are asked to enter Username and password, and if he/ she is valid User then a Admin main page (Admin dashboard) will be displayed.

4.6.6 Admin Add User DFD:

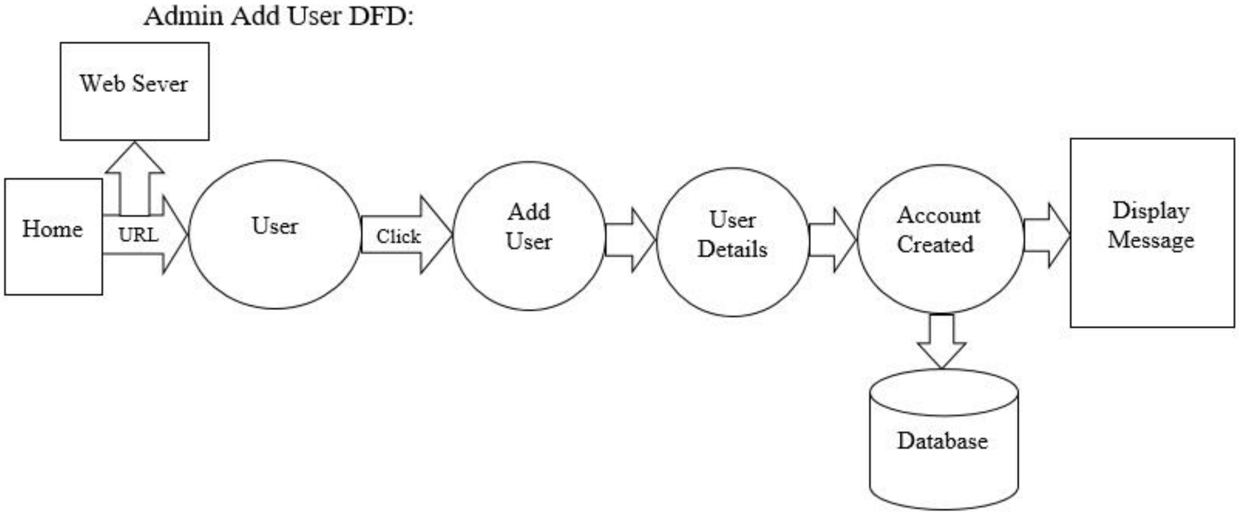


Figure 4.6.6 Admin Add User DFD

When the admin selects User, and click on add button, a new page will be opened. So the admin enters the details of that User. All field are needed to be filled and click add button, so message is displayed that User is registered.

4.6.7 Admin Add Property DFD

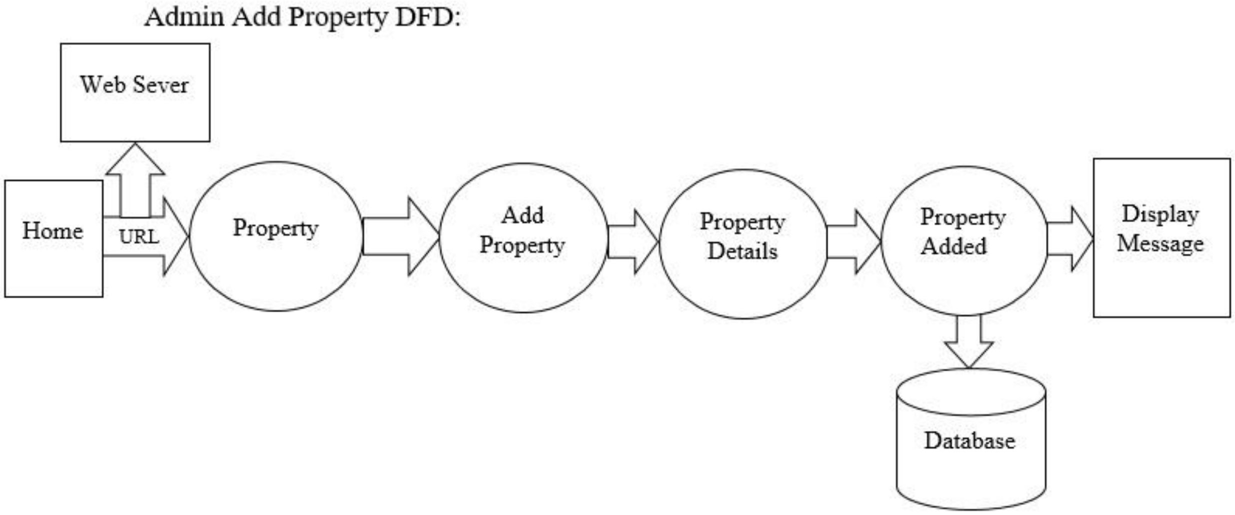


Figure 4.6.7 Add Property DFD

When the admin clicks on Property, and click on add button, a new page opened. So the admin enters the details of that property. All field are needed to be filled and click add button, so message will be displayed.

4.7 Entity Relationship Diagram

The object relationship pair is the cornerstone of the data model. These pairs can be represented graphically using the entity relationship diagram (ERD).

Peter Chen originally proposed the entity relationship diagram for the design of the relational database system. The primary purpose of the entity relationship diagram is to represent the data objects in the form of relationships. Data objects, attributes, relationships and various type indicators are the primary components of entity relationship diagram. It has following features.

i) Data objects are represented by a labelled rectangle.

ii) Relationships are indicated with a labelled line connecting objects. In some variations of entity relationship diagram relationships are represented by special diamond symbol, mostly for junctions.

4.7.1 ERD Diagram of MY Estate Agency:

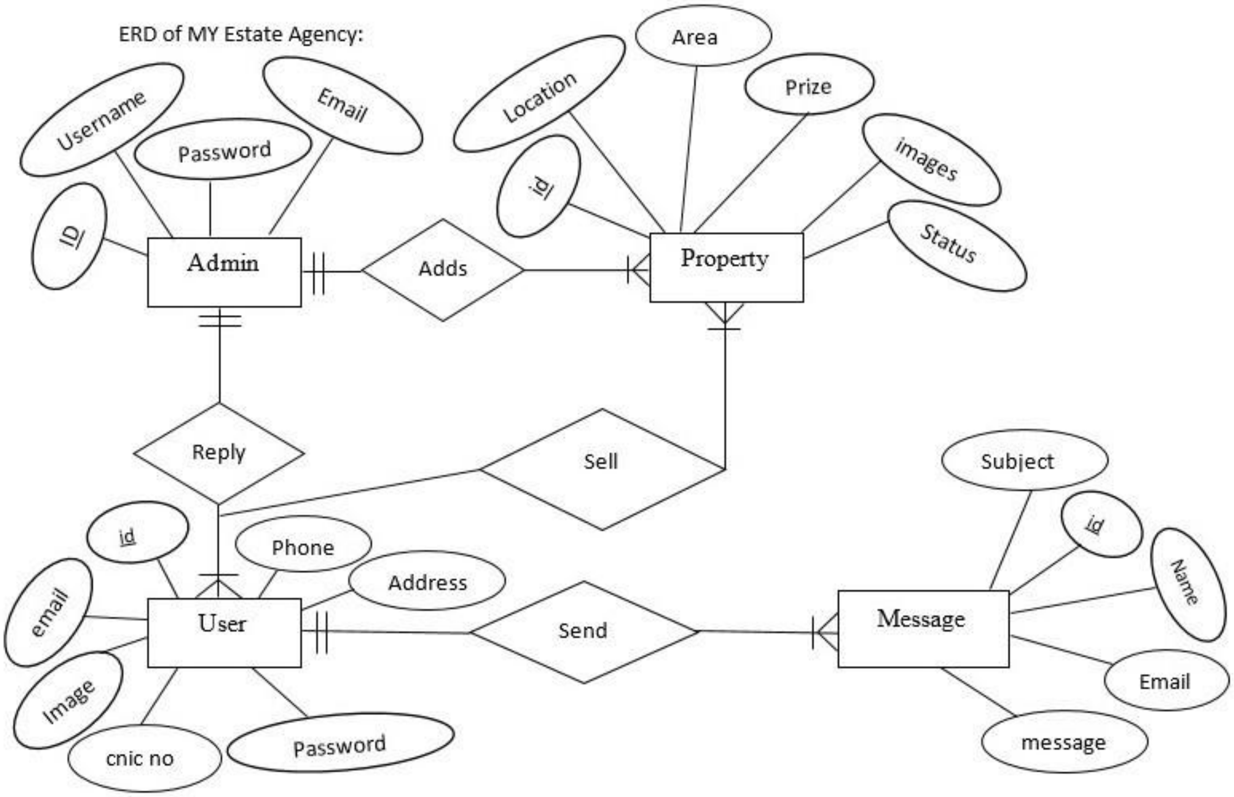


Figure 4.7.1 ERD MY Estate Agency

4.8 Physical Design

The purpose of physical database design is to translate the logical description of data into technical specification for storing and retrieving data. Its goal is to create such a design, which will store data and will provide adequate performance and insure database integrity, security, consistency and recoverability.

Designing physical files and databases, certain information is required that should have been collected and produced during prior development phases. The information needed for physical file and database design include the following requirements:

- Normalized relations, including volume estimates.
- Definition of each attribute.
- Description of data that where and when it will be used, entered, retrieved, deleted and updated?
- Expectations or requirements for response time, data security, backup, recovery, retention and integrity.
- Description of technologies (Database Management System) used for the development and implementation of the database.

- The Automation of universities Management consists of the following physical files / relations.

4.9 Data Dictionary

The purpose of this section is to explain all the data elements used in different data base files. The dictionary of the files used in the program will make the reader to understand what data name means and where it is used. All their field names, their data types, sizes and their Full Names are described below.

4.10 Input Design

Input design is the second category, which prescribes the manner in which data is to be entered or fed into the computer and then to be processed to produce the desired outputs.

If the data entered is accurate and in correct format then desired outputs may be obtained otherwise there would be the problem of black hole, black hole means that the system has inputs but no outputs have been returned by the system. It means that the inputs entered to the system are not accurate or valid which gives no output. Input design is very important so that no wrong data is to be entered to the system and so no false or unexpected outputs will be faced.

In the input design following things must be followed carefully to achieve integrity of the system.

- What are the inputs?
- Types of inputs.
- Length of input data item.
- Source of input data item.

Input means data items entered to the system for processing. After reviewing and discussing thoroughly we collected the whole data items that has to be used in different database files.

Keyboard will be used as main input device however a mouse can also be used because the software which is going to be developed is in visual language so mouse can also be the most recent input device.

It is very important to define or declare the maximum length of data items in first attempt because one data item may be used in different places so its length must be specified before it is used.

4.11 Output Design

By output design we mean the design produced by the system and the form in which it is to be produced. That is whether printed, displayed or spoken. Output design is an important factor as the system success and failure largely depends on the output design therefore more attention is given to this output design. It is through these outputs that the system becomes useful to the user. Presently computer can produce two types of outputs i.e. Output on Screen & Output on Paper.

CHAPTER

5

CODING AND IMPLEMENTATION

5.1 Implementation:

Implementation is done to the system for its working. Implementing the system means to make the system run. If the system is running correctly, then it means that the implementation is right and system is running well.

The project is implemented using the PHP programming language with MY SQL as the database used at the backend. Notepad++/Sublime Text Editor is used for implementation of the site. As we have selected the incremental model we use a modular approach. We started with a single module and moved on to the next after completing and testing the first. Every module was broken down into several module and deal one at a time.

Web application is more important for security. The hacker's life has become harder in recent days. Various intrusion detection and defense mechanisms developed by network security companies, it is no longer easy to breach security perimeters and gain unauthorized access to an organization's network.

As our project is web base project. There are many tools available to develop Web application. However, we prefer to develop out project in PHP language and MYSQL. The reason for selecting these tools to develop our project is explained in detailed below.

5.2 MYSQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by MySQL. MySQL is a commercial company, founded by the MySQL developers, that builds its business by providing services around the MySQL database management system.

5.2.1 MySQL is a Database Management System (DBMS)

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as stand-alone utilities or as parts of other applications.

5.3 PHP

PHP is one of the most popular server side scripting languages running today. It is used for creating dynamic webpages those interact with the user offering customized information.

PHP is open source and similarly to most open source software, it is free for both personal and commercial use. PHP is available for many operating systems although its native operating system is Linux. PHP has many advantages. Some of these are as follows.

- As PHP is a server side technology, the user does not need any special browser or plug-ins to see the PHP in action.

- PHP is simple, it is easy to understand and learn, especially for those with backgrounds in programming such as C, JavaScript and HTML. The language is similar to C so that anyone with a background in C programming will feel comfortable using.
- PHP runs on just about every platform including most UNIX, Macs and Windows versions.
- Another key advantage of PHP is its connective abilities. PHP uses a modular system of extensions to interface with a variety of libraries such as graphics, XML, encryption, etc. In addition, programmers can extend PHP by writing their own extensions and compiling them into the executable or they can create their own executable and load it using PHP's dynamic loading mechanism.
- The main PHP source repository is loaded with modules and interfaces that users have written and contributed. There you can find modules for flash movies, PDF files, calendars and more.
- Benefit of applications written in PHP is that they are fast and if written properly, they could be pretty secure. There are also tons of ready PHP scripts and functions, which you can customize to your liking and use in your PHP applications.

5.4 Screen Shoots:

Following are the Screen Shoots of different Modules.

5.4.1 Admin Login

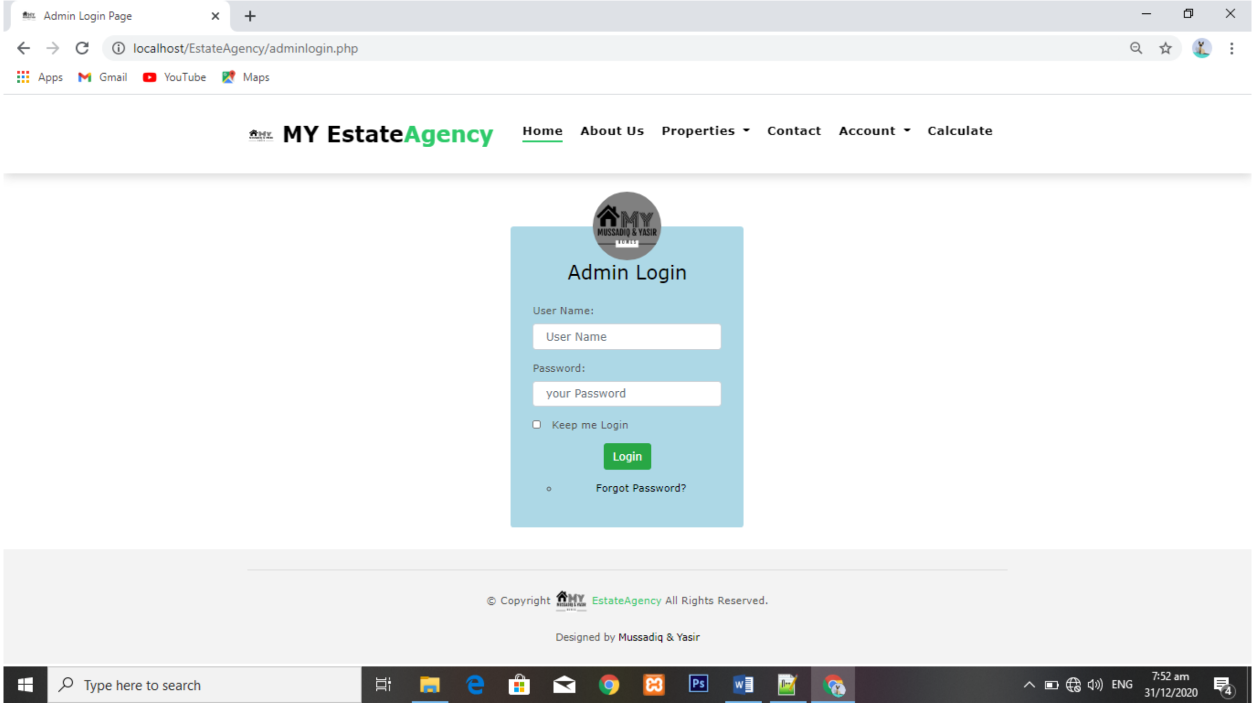


Figure 5.4.1 Admin Login

Source Code of Admin Login:

```
<?php
        $Page_title='Admin Login Page';
        include_once 'include/head.php';
        include("include/dbconn.php");
    $msg="; $username=";
    if(isset($_POST['submit'])){
      $username=$_REQUEST['username'];
      $password=$_REQUEST['password'];
      $query="SELECT * FROM `admin` WHERE `username`='$username' AND
`password`='$password'";
      $result=mysqli_query($con,$query);
       //print_r($result);
      $row=mysqli_num_rows($result);
      if ($row < 1){
        $msg="<div class='error'>Incoorect UserName/password. </div>";
      }
      else{
        $data=mysqli_fetch_assoc($result);
        $admin_id=$data['id'];
        session_start();
        $_SESSION['admin_id']=$admin_id;
        header("location:admin/dashboard.php");
      }
    }
?>
<style type="text/css">
        body{
                margin: 0px;
                padding: 0px;
                font-family:Verdana,Geneva,sen-serif;
                font-size: 15px;
        }
  .loginpic {
        width: 100px;
        height: 100px;
        border-radius: 50%;
        position: absolute;
        top: -50px;
        left: calc(50% - 50px);  }
        .error{
                color: red;  }
</style>
</head>
<body>
```

```
    <?php
        include_once 'include/searchbar.php';
        include_once 'include/header.php';
    ?>
  <main id="main">
    <div class="container" style="margin-top: 170px;">
        <div class="login-form col-md-4 offset-md-4 ">
            <div class="jumbotron" style="background-color: lightblue; margin-top:
30%; padding-top: 50px; padding-bottom: 30px;">
                <img src="images/logo.png" style="background-color: gray;"
class="loginpic">
                <h3 align="center"> Admin Login </h3> <br/>
                <form method="POST">
                    <div class="form-group">
                        <label>  User Name: </label>
                        <input type="text" name="username" value="<?php
echo $username; ?>" placeholder=" User Name" class="form-control" required />
                    </div>
                    <div class="form-group">
                        <label> Password: </label>
                        <input type="password" name="password"
placeholder=" your Password" class="form-control" required />
                        <span id="error_password" class="text-danger"
style="font-size: 14px; font-weight:bold;"> <?php echo $msg; ?></span>
                    </div>
                    <div class="form-group">
                        <input type="checkbox" name="check" /> &nbsp;
Keep me Login
                    </div>
                    <div class="form-group">
                    <center> <input type="submit" name="submit"
value="Login" class="btn btn-md btn-success"> </center>
                </div>
            <ul type="circle">
                <li><center><a href="#"> Forgot Password? </a> </center> </li>
            </ul>
                </form>
            </div>
   </div>
  </div>
 <main>
  <footer>
   <div class="container">
    <div class="row">
     <div class="col-md-12">
      <div class="copyright-footer">
       <p class="copyright color-text-a">
        &copy; Copyright
        <span class="color-b"><img src="images/logo.png" width="50" height="50" alt="LOGO"> EstateAgency</span> All Rights Reserved.  </p>
      </div>
      <div class="credits">
       Designed by <a href="#">  Mussadiq & Yasir</a>
      </div>
     </div>
    </div>
   </div>
  </footer><!-- End  Footer -->
  <script src="assets/vendor/jquery/jquery.min.js"></script>
  <script src="assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
  <script src="assets/vendor/jquery.easing/jquery.easing.min.js"></script>
  <script src="assets/vendor/php-email-form/validate.js"></script>
  <script src="assets/vendor/owl.carousel/owl.carousel.min.js"></script>
  <script src="assets/vendor/scrollreveal/scrollreveal.min.js"></script>
  <script src="assets/js/main.js"></script>
 </body>
 </html>
```

5.4.2 User Login

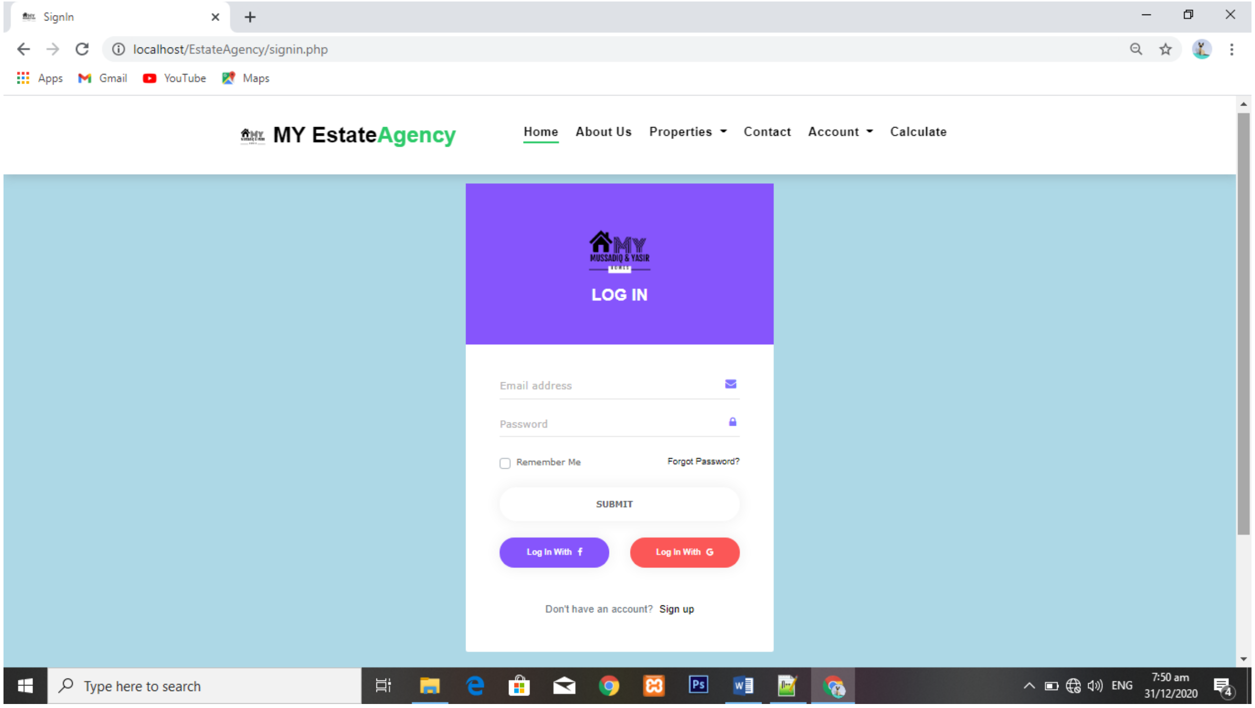
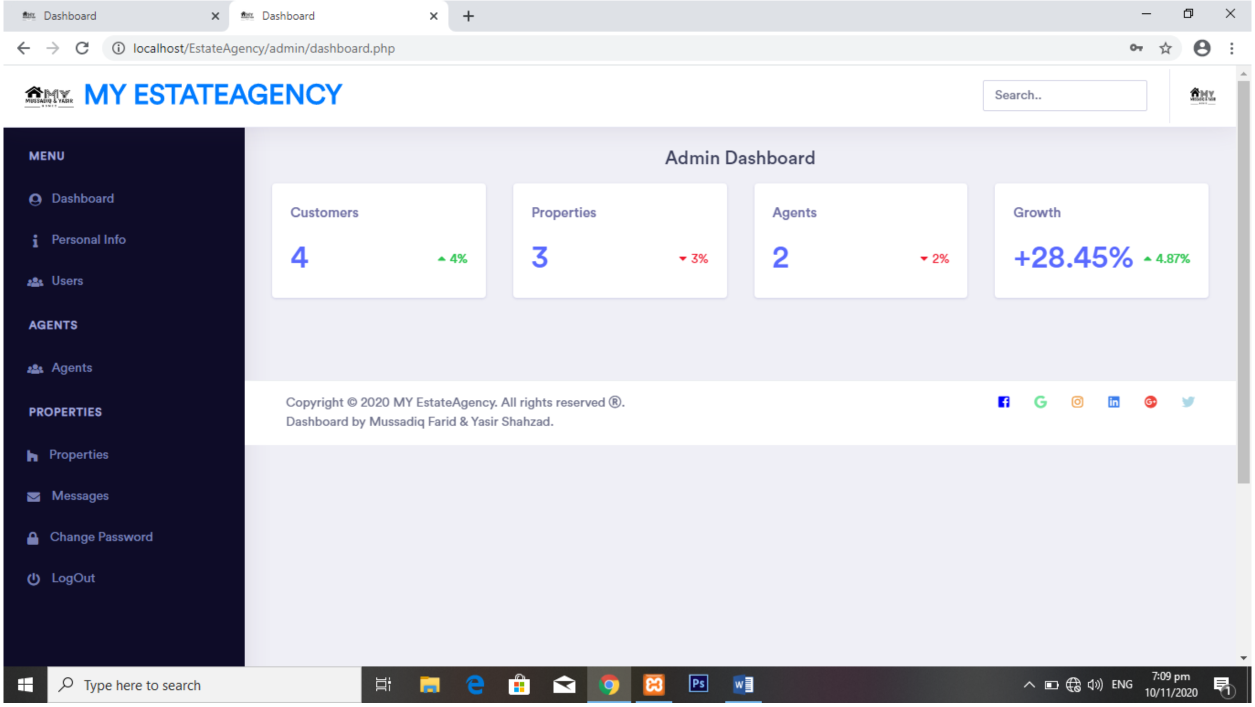


Figure 5.4.2 User Login

5.4.3 Admin Dashboard



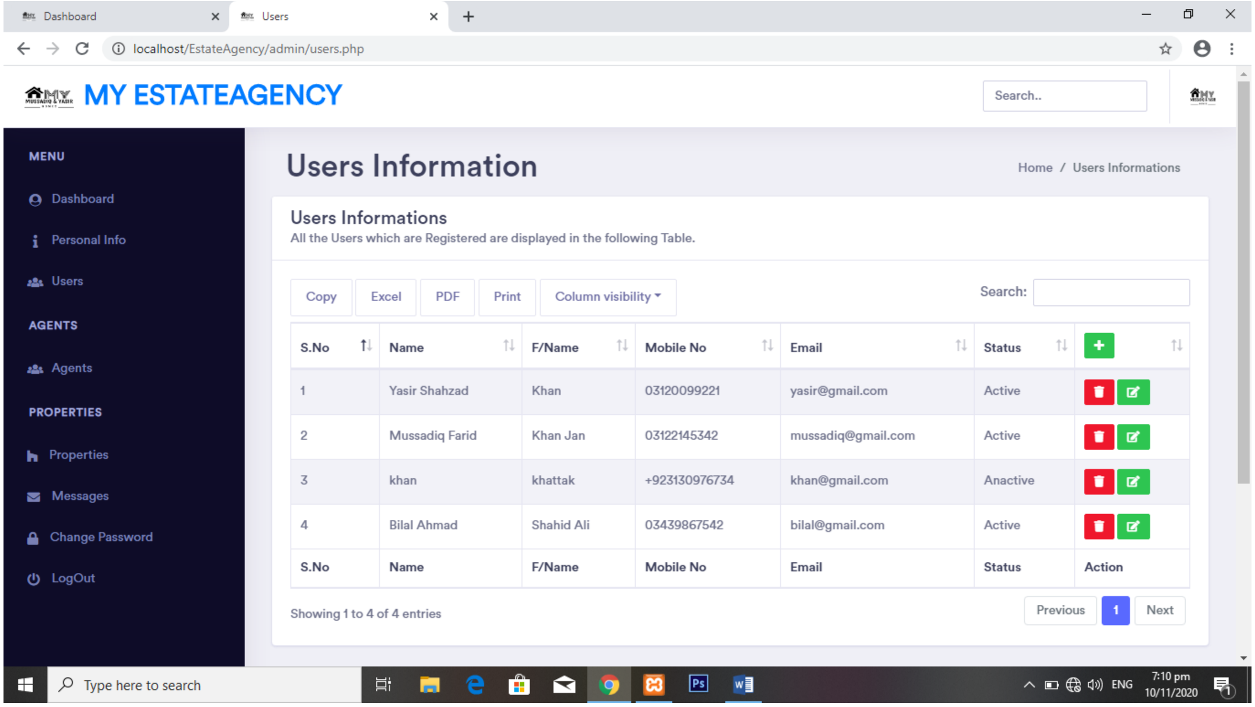
Admin Dashboard

Customers	Properties	Agents	Growth
4 ▲ 4%	3 ▼ 3%	2 ▼ 2%	+28.45% ▲ 4.87%

Copyright © 2020 MY EstateAgency. All rights reserved ®.
Dashboard by Mussadiq Farid & Yasir Shahzad.

Figure 5.4.3 Admin Dashboard

5.4.4 View User



Users Information

Users Informations

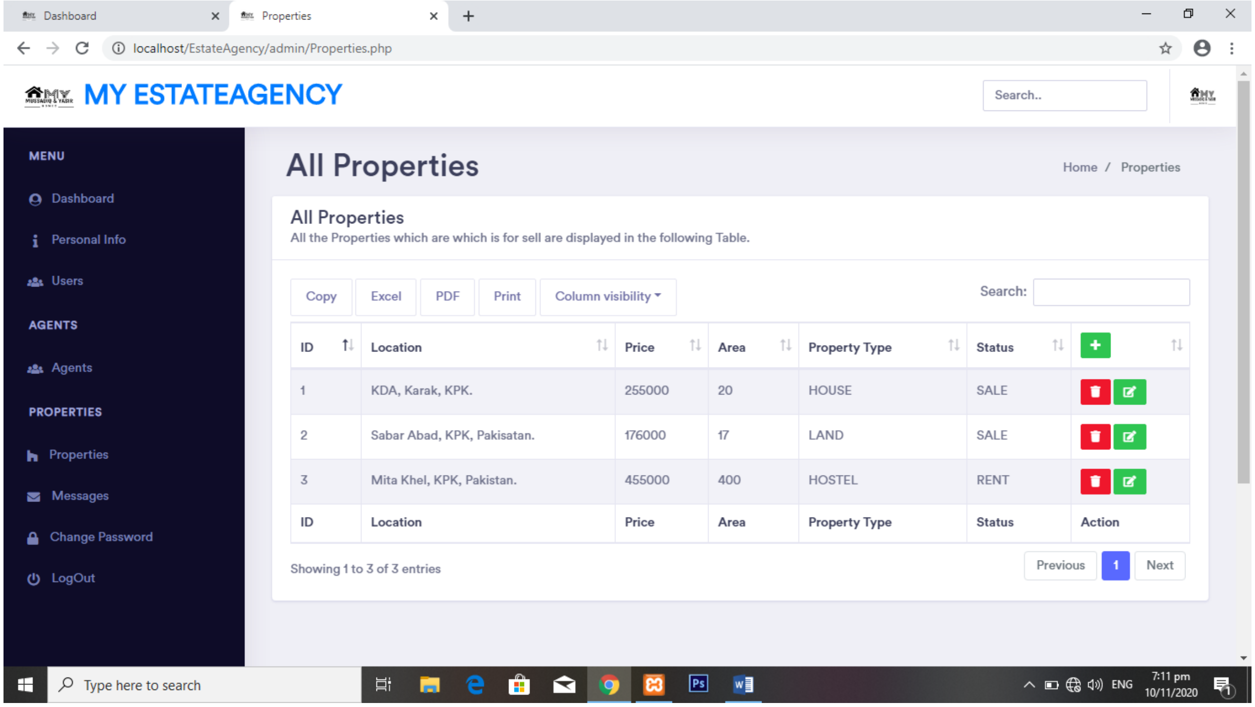
All the Users which are Registered are displayed in the following Table.

S.No	Name	F/Name	Mobile No	Email	Status	+
1	Yasir Shahzad	Khan	03120099221	yasir@gmail.com	Active	🟥 🟩
2	Mussadiq Farid	Khan Jan	03122145342	mussadiq@gmail.com	Active	🟥 🟩
3	khan	khattak	+923130976734	khan@gmail.com	Anactive	🟥 🟩
4	Bilal Ahmad	Shahid Ali	03439867542	bilal@gmail.com	Active	🟥 🟩
S.No	Name	F/Name	Mobile No	Email	Status	Action

Showing 1 to 4 of 4 entries

Figure 5.4.4 View User

5.4.5 Properties



All Properties

All Properties

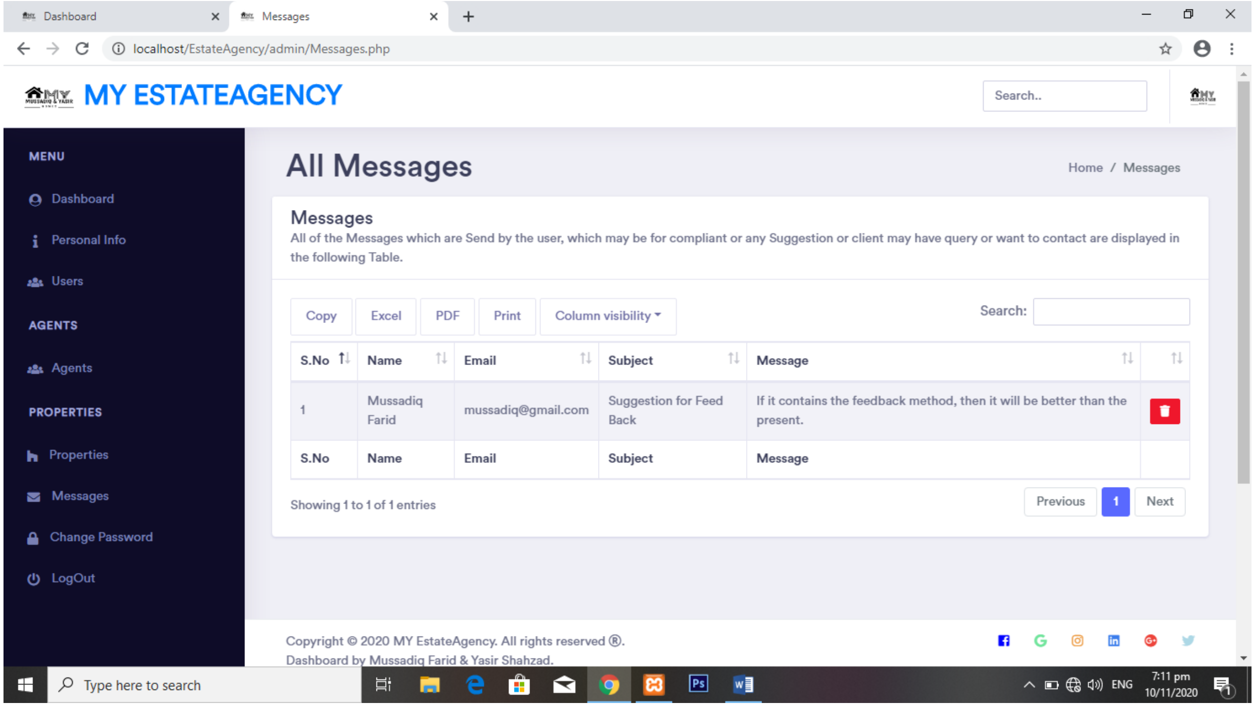
All the Properties which are which is for sell are displayed in the following Table.

ID	Location	Price	Area	Property Type	Status	+
1	KDA, Karak, KPK.	255000	20	HOUSE	SALE	
2	Sabar Abad, KPK, Pakisatan.	176000	17	LAND	SALE	
3	Mita Khel, KPK, Pakistan.	455000	400	HOSTEL	RENT	
ID	Location	Price	Area	Property Type	Status	Action

Showing 1 to 3 of 3 entries

Figure 5.4.5 Properties

5.4.6 Messages



All Messages

Messages

All of the Messages which are Send by the user, which may be for compliant or any Suggestion or client may have query or want to contact are displayed in the following Table.

S.No	Name	Email	Subject	Message	
1	Mussadiq Farid	mussadiq@gmail.com	Suggestion for Feed Back	If it contains the feedback method, then it will be better than the present.	
S.No	Name	Email	Subject	Message	

Showing 1 to 1 of 1 entries

Copyright © 2020 MY EstateAgency. All rights reserved ®.
Dashboard by Mussadiq Farid & Yasir Shahzad.

Figure 5.4.6 Messages

5.4.7 Reset Password

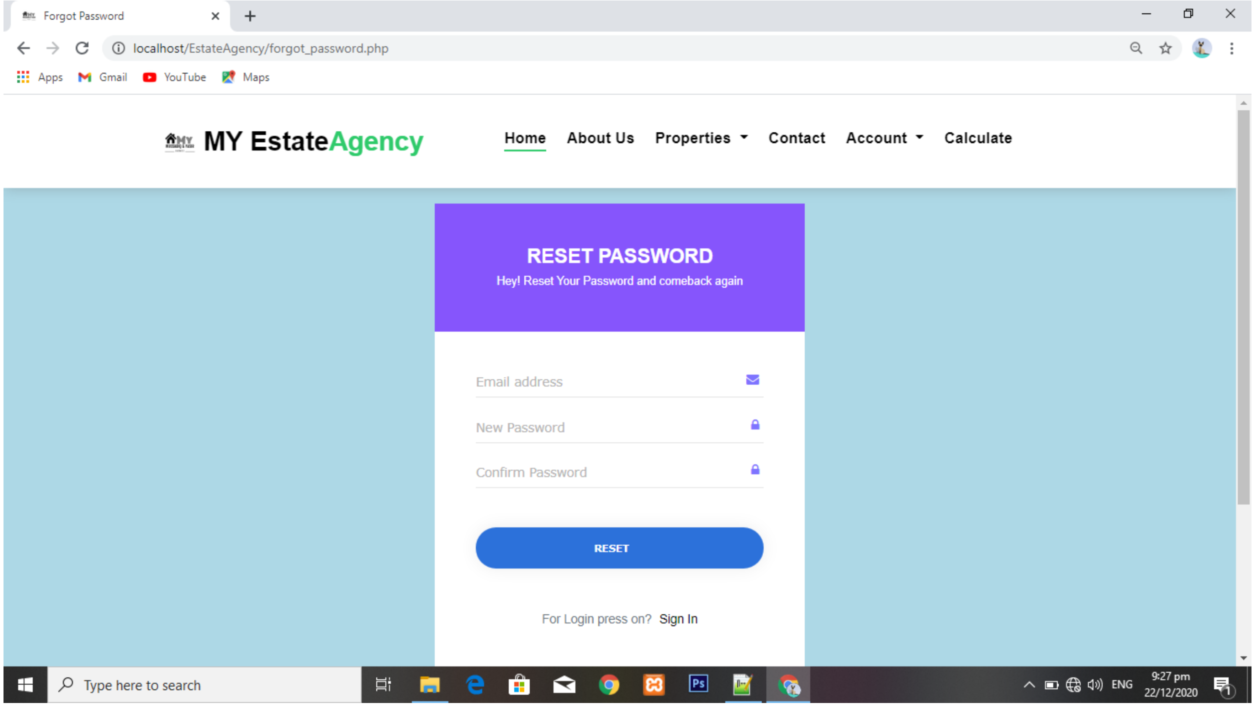


Figure 5.4.7 Reset Password

CHAPTER 6

TESTING

Software testing strategy provides a road map for the software developer, the quality assurance and the user. Testing is done in actual to assure the quality of the product developed and it compares the actual functionality of the product developed with the initial requirements of the project. Testing can basically be defined as: once the source code has been generated then the software must be tested to uncover and correct as many errors or bugs in the program.

The biggest advantage of testing is that with its help it is possible to highlight the maximum number of bugs or errors in the program. No matter how well the developers develop the software or how well it is analyzed; there is always a chance of some bugs. Testing will help to point errors and later the programmer may remove them. The importance of testing is double when web application testing is in question. A web application should be tested thoroughly tested before it is launched on the World Wide Web and is opened for the visitors.

6.1 Evaluation

The development of software system involves a series of production activities where opportunities for injection of human fallibilities are enormous. Error may begin to occur. At the very inception of the process where the objectives imperfectly specified, as well as in later design and developments stages. The increasing visibility of software as a system element and the related" costs" associated with software failure are motivating forces for well for planned, though testing. It is not unusual for software development organization to expend between 30 to 40 percent of total project of testing.

6.2 Testing Objective

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high probability of finding an as-yet undiscovered error.
- A successful testing is one that uncovers an as-yet undiscovered error. The above objectives are simply a dramatic change in viewpoint. They move counter to the commonly help view that successful test is one systematically uncover different classed of errors and do so with minimum amount of time and effort.

If testing conducted successfully (according to the object stated above), it will uncover error in the software. As a secondary benefit, testing demonstrates that software functions appear to be working according to specification and that performance requirement appears to have been met. There is one thing that testing show the absence of errors, it can only show that software errors are present.

6.3 System Testing

While developing software, there is tendency to push the end product "out the door "as soon as possible. Most of the time, software projects are not estimated properly and tend to run behind schedule. When it comes to meeting the deadline, people seem to ignore a very important phase in the whole process testing. But our project has been thoroughly tested by us. System testing is an essential step for the development of a reliable and error-free system. Testing is a process of executing a program with the explicit intension of finding errors but this does not mean to embarrass the programmer or fail the product but the positive intention to remove as many problems room the system.

A test case is a set of data items that the system processes as normal input. Good testing involves much more than just running the program a few times to see whether it works. A successful test is the one that finds error.

6.4 The place of Testing in SDLC

Testing has an important place is SDLC (Software Development Life Cycle). The SQA (Software Quality Assurance) team should be involved in the early phases of his project. This will make them the aware of the requirements and techniques that are going followed enabling them to develop an appropriate test strategy.

- The marketing group will bring the customer requirement.
- The engineering group will access the feasibility.
- The engineering and SQA groups will work together during the design phase and generate the product specification.
- The developers will start implementing while the SQA team is developing the test plans and so on.

While the developers are ready with the code, the SQA team will be ready for testing. The testing phase is no longer bottleneck for product delivery because the SQA team has been involved since the start of the project.

6.5 Testing Strategies

A well –organized testing strategy should include following types of testing:

6.5.1 Functional Testing

This is also referred to as white box testing. It is some at high level and takes individually tested features and places them together in small applications and test cases to test whether they work together in small application.

6.5.2 Black-Box Testing

A strategy in which a software component is treated like an opaque box. This tests designers' focus on determining how well the component conforms to the published requirements for the component, instead of worrying about the implementation details.

Black box testing focuses on the functional requirement of the software. This testing strategy enables us to derive sets of input conditions that wilfully exercise all functional requirements for a program. Black box testing is not an alternative to white box testing. Rather, it is a complementary approach that is likely to uncover a different class of errors than white box testing method.

Black box testing used to find following errors:

- Incorrect or missing functions.
- Interface errors.
- Initializing and termination errors.
- Performance errors.

6.5.3 White- Box testing

A strategy in which software component is treated as a transparent box. Test designer can peek into the box and gain knowledge about the implementation. They can use this knowledge to build test cases cover different parts of the code and also follow different execution paths.

White box testing is a test case design method that uses the control structure of the procedure design to derive test cases. White box testing enables to derive test case that:

- Guarantee that all independent paths within a module have exercised at least once.
- Execute the all loops at their boundaries and within their operational bounds.
- Exercise all logical conditions on their true and false paths.

6.5.3.1 Benefit of White Box Testing

Logic errors and incorrect assumptions are inversely proportional to the probability that a program path will be execute. Errors tend to creep into our work when we design and implement functions,

- Condition or controls that are out of mainstream. Every day processing tends to be well understood, while “special cases” tends to fall into the cracks.
- We in fact believe that a logical path is not likely to be executed when; in fact, it may be executed on regular bases. The logical flow of a program is sometime counterintuitive. Meaning that our unconscious assumption about flow of control and data may lead us to make design error that is uncovered only once path testing commences.
- Typographical errors are random. When a program is translated into programming language source code. It is likely that some typing error will occur. Many will be uncovered by syntax checking mechanism, but will go undetected until testing begins. It is a likely that a typo will exist on an obscure logical path as on a mainstream path.

6.5.4 Unit Testing

The testing of individual units of the application in isolation for example, a single class. Unit testing focuses verification efforts on the smallest unit of software design, the module. Important control paths are tested to uncover errors within the boundary of the module. All he slides of child Educator are tested separately to find error easily with the range of single slide. Every slide is tested to make sure that slides are working within the specifications.

In unit testing we will test each individual component of a system.

Serial No	Test case Description	Expected result	Result
1	Login using correct username and password	The system will give access to the user	Pass
2	Login using wrong username and password	The user will not access to the website	Pass
3	Add User with filling all requirements	The User will be registered	Pass

4	Add User with leaving any form empty	The User will be not be registered and error will generate to fill the form	Pass
5	Add Property with filling all requirements	The Property will be added	Pass
6	Add Property with leaving any form empty	The Property will be not be added and error will generate to fill the form	Pass
7	User see the information's when user is logged in	The User will see information gives access	Pass
8	User see the information's when user is NOT logged in	The User will NOT see information and access denied	Pass
9	Manage Users(edit and delete Users) when logged in as admin	Successfully	Pass
10	Manage Property (edit and delete) when logged in as admin	Successfully	Pass
11	Manage Messages(edit and delete) when logged in as admin	Successfully	Pass

6.6 Testing Specification Plan

For quality control to be effective, one must test time same things he same way every time one test. When you change your tests, your results become inconsistent. You need a test plan. A test plan is simply a high-level summary of the areas (functionality, elements, regions, etc.) one will test, how frequently one will them, and where in the development or publication process one will test them. A test plan also needs an estimate of the duration of testing, and statement of any required resources.

The major phases of software need test plans, because the focus and emphasis of testing will change over time. Testing new software in development is very different from testing software that has been running for some time. Furthermore, any changes to the software code, incremental or major, require regression test plans.

Clearly, one needs to decide what will be test, understand the software-the software should have a concrete explanation of the "vision" behind its creation and the hoped-for "path "for its success. If the software has no such explicit statement of direction, then the codifying of such a statement should be first goal. To help define what one should test, ask yourself if these types of questions.

- Why did I make this software? What is the software's purpose?
- What are the business goals, if any, behind this software?
- What has to work for this software to be effective?
- What has to work for his to even function as software?

- Who are the end-users for this software?
- Can they use this software with ease?
- What is core functionality offered by this software?
- Can all users at least access this core functionality?

Use the answers to these questions to decide what needs to be tested, and then develop your test cases.

6.7 Software Bug

A piece of software code that is not working as expected. No matter what metrics you use for testing your software, it is important to realize that robustness you expect from the software is dependent on his development phase of the software. In other words, the robustness expected from a final product is more than that from a beta release, which in turn is more than an alpha release.

6.8 Project Testing Report

The testing of “MY Estate Agency” is under gone through all stages of black box testing and white box testing. In the evaluation phase the system is reviewed to see whether the objectives of the system are accomplished or not. A major factor during system evaluation is to evaluate the system with the perspective of the user because he/she is eventually being the one who use it. The testing of “MY Estate Agency” is as follow.

- Trace-ability Test matrix

6.8.1 Trace-ability Matrix for This Project

Test Case ID	Test	Result
Test 1	Verify that user can see all Pages of the Software	Test has passed successfully.
Test 2	Verify that user can See all modules	Test has passed successfully.
Test 3	Verify that user get error messages on wrong entry	Test has passed successfully.
Test 4	Verify that user gets error message on entering wrong password	Test has passed successfully.
Test 5	Verify that all links are linked with the desired forms	Test has passed successfully.