

Jaypee University of Information Technology
Waknaghat, Distt. Solan (H.P.)

Learning Resource Center

CLASS NUM:

BOOK NUM.:

ACCESSION NO.: SP09004/SP0913004

This book was issued is overdue due on the date stamped below. If the book is kept over due, a fine will be charged as per the library rules.

Due Date	Due Date	Due Date

Online Real Estate Property Management
Project Report submitted in partial fulfillment of the
requirement for the degree of
Bachelor of Technology.

in

Computer Science and Engineering

under the Supervision of

Mr. Arvind

By

Aman (091320)

Amit Kumar Pandey (091254)

to



Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that project report entitled “**ONLINE REAL ESTATE PROPERTY MANAGEMENT**”, submitted by **Amit Kumar Pandey (091254)** and **Aman (091320)** in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Mr. Arvind

Arvind
28/05/13

Date: *28/05/13*

Department of Computer Science Engineering and Information Technology,
Jaypee University of Information Technology,
Waknaghat.

Acknowledgement

I would like to acknowledge our sincere thanks towards our Project guide Mr. Arvind for his valuable guidance and suggestions, Mr. S.P.Gherera for his moral support and our computer science faculty for their teaching and undivided attention to us during our B.Tech course from which we have been able to successfully complete our 4th year project.

I would also like to acknowledge our thanks towards the facilities provided by the University which has helped us in our project.

I am grateful to all those who have directly or indirectly helped me in completion of the project.

Date: 28/05/ 2013

Aman(091320)

Amit Kumar Pandey(091254)

Table of Content

1.0 Introduction

1.1	Purpose Of Online real estate property management	1
1.2	Scope	2
1.3	Definitions, Acronyms and Abbreviations	3
1.4	Literature review	4
1.5	Advantages of Online real estate property management	5
1.6	User profile	5

2.0 Problem statement and requirements

2.1	Business requirements	7
2.2	User requirements	8
2.3	Operational requirements	8
2.4	System requirements	9
2.5	Specific requirements	9

3.0 Overall description

3.1	Project persperctive	11
3.2	Software interface	11
3.3	Modules	12
3.4	Software engineering design	17
3.4	System architechture	24
3.6	Project roadmap	25
3.7	Databases	26
3.8	Test plan	27

3.9	design goals and constraints	28
-----	------------------------------	----

4.0 Code and Snapshots

4.1	Code of database	30
4.2	Snapshots of database	35
4.3	User interface coding	40
4.4	Snapshots of user interface	78

List of Figures

Title	Page No.
FIGURE 1: USE CASE DIAGRAM	17
FIGURE 2: CONTEXT LEVEL DFD	19
FIGURE 3: DFD LEVEL 1	20
FIGURE 4: DFD LEVEL 2	21
FIGURE 5: DFD LEVEL 3	22
FIGURE 6: DFD LEVEL 4	23
FIGURE 7: SYSTEM ARCHITECTURE	24
FIGURE 8: PROJECT ROADMAP	25
FIGURE 9: SNAPSHOT - HOMEPAGE	78
FIGURE 10: SNAPSHOT - HOMEPAGE	78
FIGURE 11: SNAPSHOT –SEARCH PAGE	79
FIGURE 12: SNAPSHOT –SEARCH PAGE	79
FIGURE 13: SNAPSHOT –ADD PROPERTY PAGE	80
FIGURE 14: SNAPSHOT –ADD PROPERTY PAGE	80
FIGURE 15: SNAPSHOT - ADD PROPERTY PAGE	81
FIGURE 16: SNAPSHOT –MEMBER PROFILE OF ADMINISTRATOR	81
FIGURE 17: SNAPSHOT – MEMBER PROFILE OF ADMINISTRATOR	82
FIGURE 18: SNAPSHOT - MEMBER PROFILE OF USER	82
FIGURE 19-20: SNAPSHOT - REGISTRATION PAGE	83
FIGURE 21: SNAPSHOT – LOGIN PAGE	84
FIGURE 22: SNAPSHOT - LOGOUT PAGE	84
FIGURE 23: SNAPSHOT – USER DETAILS PAGE	85

List of Tables

Title	Page No.
TABLE 1: Events Table	18
TABLE 2: cat_prop	35
TABLE 3: categories	36
TABLE 4: fac_prop	36
TABLE 5: facilities	37
TABLE 6: fav_prop	37
TABLE 7: images	38
TABLE 8: property	38
TABLE 9: telephone	39
TABLE 10: users	39

Abstract

This website is a complete end to end solution to cover all aspects of real estate Property buying selling procedure for small and large organization and individuals.

It is basically a user friendly contact and property manager for real estate professionals and property buyers. The main role of the project is

- Real estate system serves the server operator
 1. When searching the properties to buy.
 2. When uploading the property details .
 3. When uploading information of users who are either buyer or seller.
- Real estate website serves customers to search about the property through the web.
- Real estate website focuses on main operation like selling, buying and searching but the financial operation is out of scope.

The basic objectives of the project are:-

- Each seller can know information about the customer before signing the deal to know if there is any unpaid money or if it using the property is bad.
- Each user is linked with a network and share information in a central data base that gives the user beneficial information like: -
 1. All properties that are available.
 2. Details on all the properties that are available.
 3. Information about the lessees and sellers.
- Each user can either buy or sell a property according to his/her requirement.
- The administrator has unlimited access to all of the website's operation and function.
- The administrator can edit, delete or modify any user's detail or property detail in the website from the central database.
- Each interested buyer can directly contact the seller to sign the deal on the property he/she is intending to buy.
- Users can modify details about the property uploaded by them in the website.
- The home page will have new and featured properties displayed for sale and re .

CHAPTER 1

INTRODUCTION

1.1 Purpose of online real estate property management

Technology plays a very important role in today's life. Everything from regular routine work to big industry operations are getting digitized. Computers today are making everyone's life much easier than before. Everything nowadays is online from buying clothes, travel ticket reservation to booking movie tickets etc. E-commerce is becoming a competitive industry and according to experts it is still in its infant stage and has a very good future to look forward to.

The real estate properties have always been a point of interest among the people and after technological revolution people have started trading properties over the internet. Due to the advantage of large data storage and easy access people can now easily browse through wide range of properties easily over the internet with minimal effort and then can invest or buy more conveniently. The internet has changed the real estate business drastically and will continue to revolutionize it with new concepts.

This project is another step in that direction. It will enhance the real estate market even more by providing a website which will take care of all users' needs and will help them in every possible way to buy and sell properties online. It will become a point of contact between the interested buyers and property sellers.

Each user will be able to register him/herself on the website and then will be able to either buy a property posted by another user or post a property on the website for the purpose of selling it. The user will be able to perform even other functions such as edit his account, edit his details, edit his posted properties, perform property search etc.

A user friendly interface will help gain the website an edge in the market and efficiency in operations will help in providing users' trust in the website and would thus result in a good user experience.

1.2 Scope of online real estate property management

- The project will deliver an altogether new version of online real estate property management with an added feature of home page displayed properties.
- Will reduce the work and improve efficiency as a user will be having everything on one website.
- The portal will provide a facility of notifications to the user. (Eg: A user wants to share some property details with his friend, then he view his contact info and send him the required details).
- Daily feeds will be provided to the user when he logs in. A person can directly start browsing the properties from there itself.
- FAQ's and other interactive features will make website more user friendly.
- Feedback option will be available.
- Loan provider details will also be available.
- Important support Email Ids will be displayed on the website.
- The website administrators contact details will also be available.

1.3 Definitions, Acronyms, and Abbreviations

Terms	Definitions
User	User of the system.
Guest	User of the system.
Admin	The administrator and the moderator of the system.
HTTP	Hypertext Transfer Protocol is a transaction oriented client/server protocol between web browser & a Web Server
HTML	Hyper text markup language
SDD	Software Design Document
ODMCS	Online Document Modification, Conversion and Sharing Portal
JSP	Java Server Pages
Eclipse	Eclipse is a toolkit which is designed for the creation of more complex projects, providing fully dynamic web application utilizing EJB's. This consist of EJB tools, CMP, data mapping tools & a universal test client that is designed to aid testing of EJB's
XML	Extensible Mark-up Language is used for the storage and carriage of data.
FTP	File Transfer Protocol
PHP	Hypertext Preprocessor
FAQ	Frequently Asked Questions

1.4 Literature Review

➔ Magicbricks:

Weblink: www.magicbricks.com

- To really improve the user experience they have some of the most common searches in there area accessible through a single click.
- They have a blog based site
- They post regularly in there blog.
- They have a great Search interface for their clients, so that they can find it easily on the internet.
- They use mysql as their backend database to store all their data.

➔ 99 acres:

Weblink: www.99acres.com

- Most property listings and great listing information are given in their website.
- You can go into a deal negotiation feeling comfortable and educated.
- Includes in-depth neighborhood statistics and unusual property types.
- It has very minute details of every real estate property listed in their website.
- they have an instant property search option in their homepage.

➔ Makanwala

Weblink: www.makanwala.com

- It has a clean, captivating traditional design.
- It is very simple and user friendly especially good for a first time buyer of real estate properties.
- It presents a housing locator to visitors in the top half of the home page.

→ Estate Agent and Property Management System eProperty by Atanu Malik

Weblink: <http://www.programmer2programmer.net/>

- It has simple features of buying and selling residential and commercial property.
- It also has a feature of instant search where a user can instantly search for his/her property by just giving a few details.

1.5 Advantages of online real estate property management

- Using online real estate management property system in managing real estate helps the user to digitize all the real estate data.
- It makes all the tasks of buying and selling easier compared to the traditional ways of real estate business.
- The details are stored safely in the server.
- The property details are easily available that is they can be accessed from anywhere from just a computer with an internet connection.
- The website has a user friendly experience and helps the users to easily search through properties.
- The users can easily post their properties using the post property link.
- Even user registration is very easy as it requires one simple step.
- The administrator can view, edit and delete every detail on the website.

1.6 User profiles

The following are the different user profile of the website:-

1. Visitor:- it has the following characteristic;-
 - It browse through the website
 - It should be able to perform quick search
 - He/she should be able to register him/herself.
2. User – It has the following characteristic :-
 - It has read only rights to most functions of the website.
 - It can only edit its own account information and its own posted properties.
 - It can view all the posted properties in the website by using the instant search or the advanced search option.

- The user can register himself by clicking the link given on the home page.
 - The user can also login to website every time he/she visits it.
3. System administrator – It has the following characteristic:-
- It can view all the user profiles that exist in the database.
 - It can change any user profile if it wants to.
 - The administrator can edit and even delete any user profile.
 - The administrator can even edit and delete any property posted on the website.
 - It has unlimited access to the user profile, the property details and the website's database.

CHAPTER 2

Problem statement and requirements

The main basic requirement of this real estate management property system is to provide a user friendly interface and manage all the property management tasks efficiently and effectively.

The following preliminary list are the main requirements:-

1.1 Business Requirements

The business goal for the application is to increase efficiency of registration process and powerful search facilities and capable of matching property with client's requirement while increasing the productivity and managing it effectively.

The main business requirements are as follows:-

- Improve the search facility and all the users of Property Management System should get all the information in a second.
- The application should support the capability to use multi user environment.
- Administrator should be able to view all types of profiles as and when required by the organization.
- System should store all the property details in the database which should be easily accessible.
- Improve client's ability to analyze property and easily find what they need.
- The system should be refreshing and updating the database from time to time and should be able to identify the best properties for the featured properties section.
- The system should also update properties in the new property section and also in the feature ads section so that users and the administrator can view the new and featured properties on the website.
- The system should display all the property details instantly when the user clicks on the properties.
- The images should not take time to display.

1.2 User requirements

User requirements are categorized by user type.

1. Visitor:-

- Able to browse through the website
- Register himself as the user
- Should be able to perform quick search
- Should be able to view featured properties.

2. User:-

- Able to search and view his or her own read only information.
- Able to register himself quickly.
- Should be able to login and view his profile easily.
- Should be able to view his posted profile on the website.
- He/She should experience a user friendly environment when browsing the website.
- He/she should be able to change and edit its own profile information.

3. System Administrator:-

- Create new user, assign role and privileges
- Database backup and maintenance.
- It should be able to view all the user profiles that exist in the database.
- It should be able any user profile if it wants to.
- The administrator should be able to edit and even delete any user profile.
- The administrator should be able to edit and delete any property posted on the website.
- It should have unlimited access to the user profile, the property details and the website's database.

1.3 Operational Requirements

The following requirements provide a high-level view of how the system will run:

- A full weekly backup is required to WORM drives.

- The server should be monitored from time to time.
- The website should have a quick access time.
- Application should run in Client-Server architecture.

1.4 System Requirements

These are additional constraints from a system perspective:

- The administrator must be able to monitor everything from the IT department.
- The administrator should have unlimited access to the websites operations and its database.

1.5 Specific requirements

Follwing are the functional and non-functional requirements:-

1. Functional requirements:-

- The home page to consist of features like: featured properties, new preoperties.
- Database to be maintained.
- Feedback to be supported to enhance the user experience.
- Guests can view the featured porperties.
- Prevent standard automated software from filling out a form.
- Any person can register as members.
- Registered members can perform their required functions.
- File can be downloaded in various formats.
- File can be downloaded in various formats.
- File can be uploaded in various formats.
- Registered users and Administrator will have their own profile, where they can perform their work.

2. Non Functional Requirements:-

- Secure access of confidential data (user's details).
- 24 X 7 availability
- Better component design to get better performance at peak time
- Flexible service based architecture will be highly desirable for future extension.

CHAPTER 3

OVERALL DESCRIPTION

3.1 The project's perspective

The website will provide the means for the users to use the services at their own convenience. The website will automate the whole system of real estate property buying and selling and sharing it easily, all online.

3.2 Software Interface

Server Requirements:

- Operating System - Windows XP, Vista, 7
- Database - Mysql

Technologies:

- PHP
- HTML

Client Requirements:

- Operating System - Windows XP, Vista, 7
- Browser - Internet Explorer, Mozilla Firefox, Google Chrome, Netscape Navigator.

Front end :

- HTML/ CSS

Platform :

- PHP

Backend:

- MySQL

Webservers:

- Wamp 2.2

Communication Interface

- Client on Internet will be using HTTP/HTTPS protocol.
- Client on Intranet will be using TCP/IP protocol.

3.3 Modules**-----Master modules**

1. Property registration.
2. User registration
3. Website login
4. Change password

----Administrator modules

1. Matching properties.
2. User detail.
3. Seller's detail.
4. Storing details.
5. User's post

----Search query modules

1. Instant search
2. Extensive search
3. Update keyword

----User modules

1. User registration
2. User login
3. User requirements
4. User records
5. User properties

- **Master modules**

1. Property registration:-

- A user can register his property.
- An option would be displayed to upload property images.
- An address bar would take the address and store it in the database.
- A property document option would store the property details.
- The property details would then be validated.
- The property details would then be stored in the database.

2. User registration

- This module will be available at the home page.
- The user has to register by filling up the user registration form.
- The user has to give the following details
 1. Name.
 2. Address.
 3. Age.
 4. Sex
 5. Occupation
 6. Annual income
 7. An E-mail id.
 8. A password.
- Then he would get a registration confirmation and then he can start buying, selling or dealing properties by logging in to the website.

3. Website login
→ This module will ask for username and password.

4. Change password
→ The user can change to his desired password.

- **Administrator modules:-**

1. Matching properties
→ It will match the properties with the clients.
→ It will be done by using the instant search.
→ After a successful match it will wait for the user response.
→ After the user gives his response he would then store the user response.

2. User details
→ The database will store the user details and it will be readily available for the operator and the user.
→ It would also store the user activity on the website.
→ It would also store the property search preference for new properties.

3. Seller's detail
→ The user can view the property owner's details.

4. Storing details
→ This would store all the details of the operations performed
→ A notification would also be generated to the operator.

5. User's post

→ It would store all the properties posted by the user.

→ And his/her details

→ And his/her activities.

*

- **Search module**

1. Property locator.

→ It is an instant search which would search for property details.

→ It will ask the details of the property to be searched.

→ Enter an appropriate keyword to search.

2. Extensive search

→ It would ask for more details for a specific type of property as follows:-

1. The Property location.
2. The building's characteristics.
3. The property size
4. The land rate for a building.
5. The purpose of the property that is if it is a commercial or a residential property.

→ The details filled would be more exact and matching very closely to the property desired.

3. Update keyword

→ The user with the help of this function can update a keyword to change his search.

→ The search will then again start from the beginning.

→ The update keyword could be used in normal search and also in extensive search.

- **User modules**

1. User registration

➔ The new user can register his details in form available in the home page.

2. User login

➔ The user can login from the homepage with his user name and password.

3. User requirements

➔ The user can give his requirements and then the search algorithm will search for the property for the user.

4. User records

➔ all the user details will be stored in database.

5. User properties

➔ The user's properties will be stored since his/her registration would be stored.

➔ Property suggestions would be made according to his/her activities.

3.4 Software engineering design

Use case diagram:-

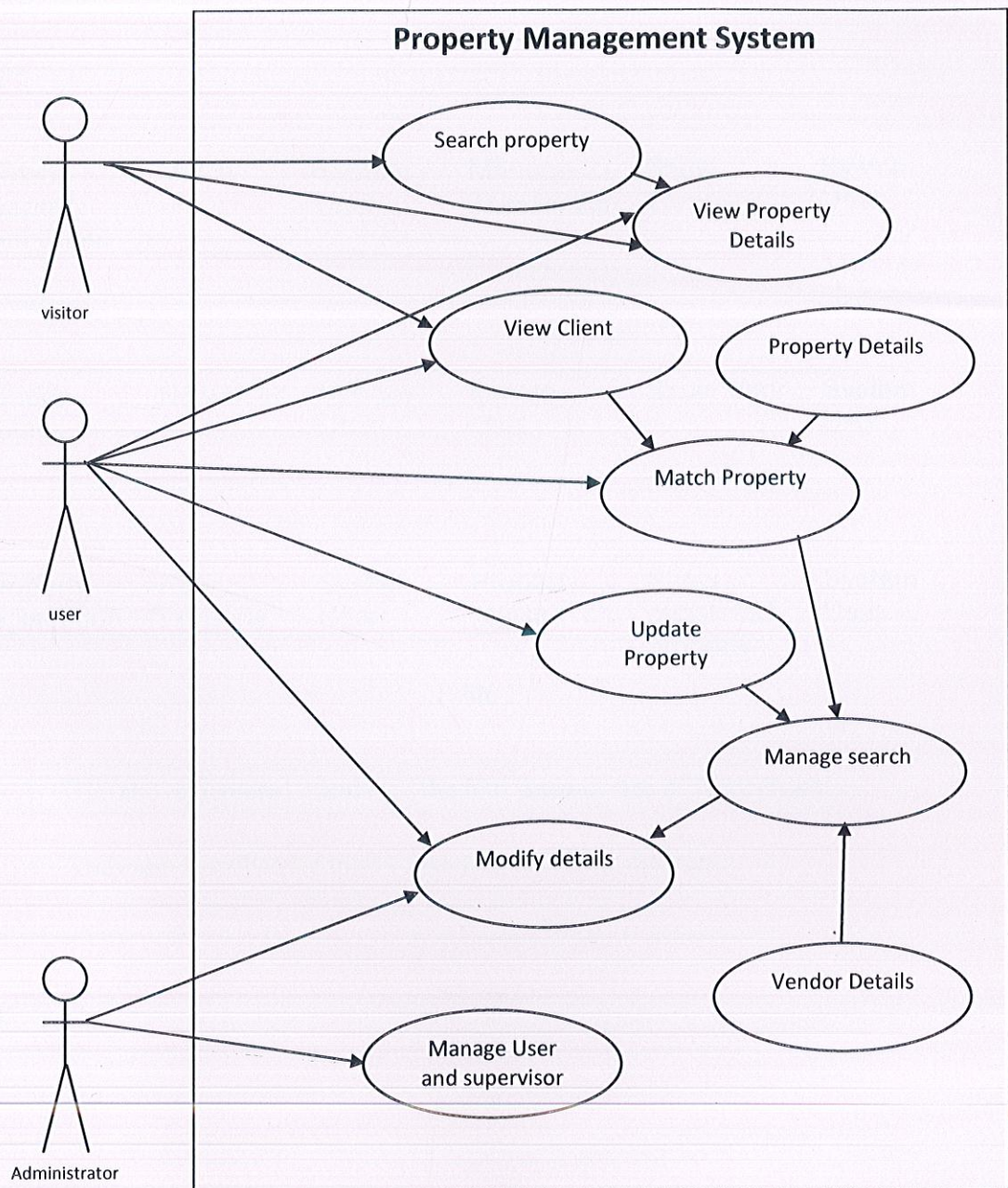


Fig - 1

Event table :-

Event	Trigger	Source	Activity/Use Case	Response	Destination
Client Registration	New User	System User	Create User	Show User	System User
Match Property with Client	Match	System User	Manage Transaction	Show Transaction	System User
vendor details	New User	System User	Create User	Show User	System User
Property registration	New Property	System User	Property registration	Show registered property	System User

Table 1

- The above depicted events are the four main events of the website.
- They are described by their respective data flow diagram.

Context level DFD:

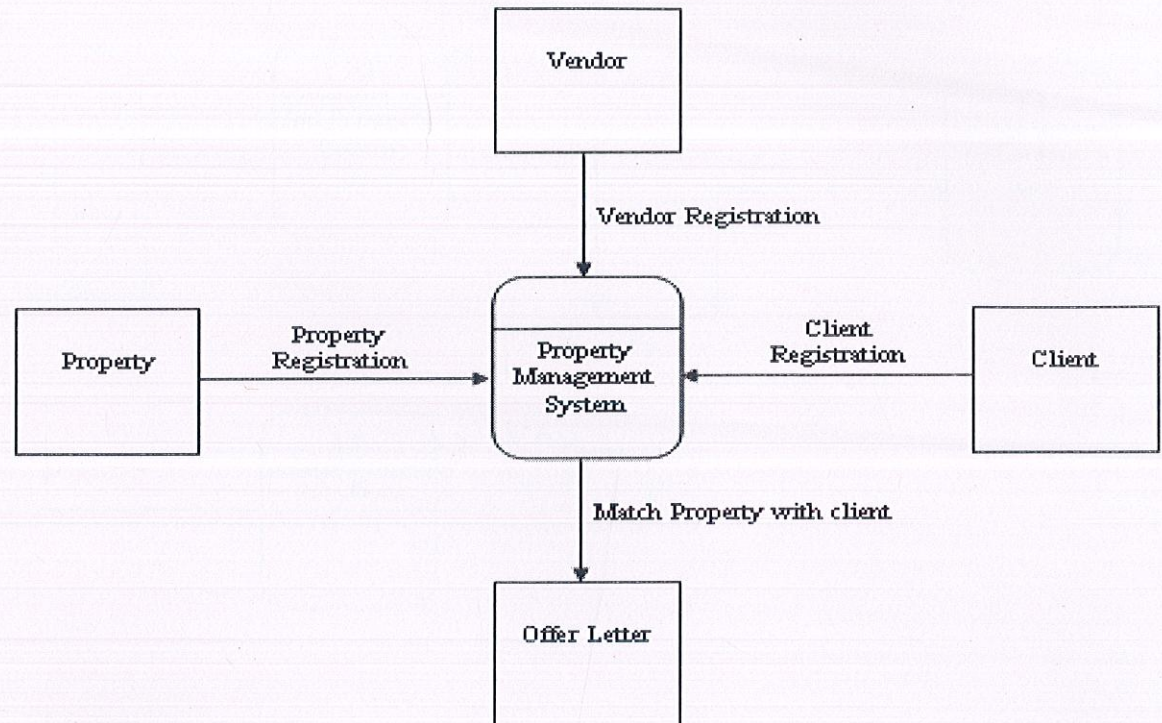


Fig - 2

The four objects which are client, vendor, property and offer letter would perform their functions which are registration done by client/vendor which then registers their property and if it is matched with the client demands offer letter is generated.

Level 1 DFD :
(to match property with client)

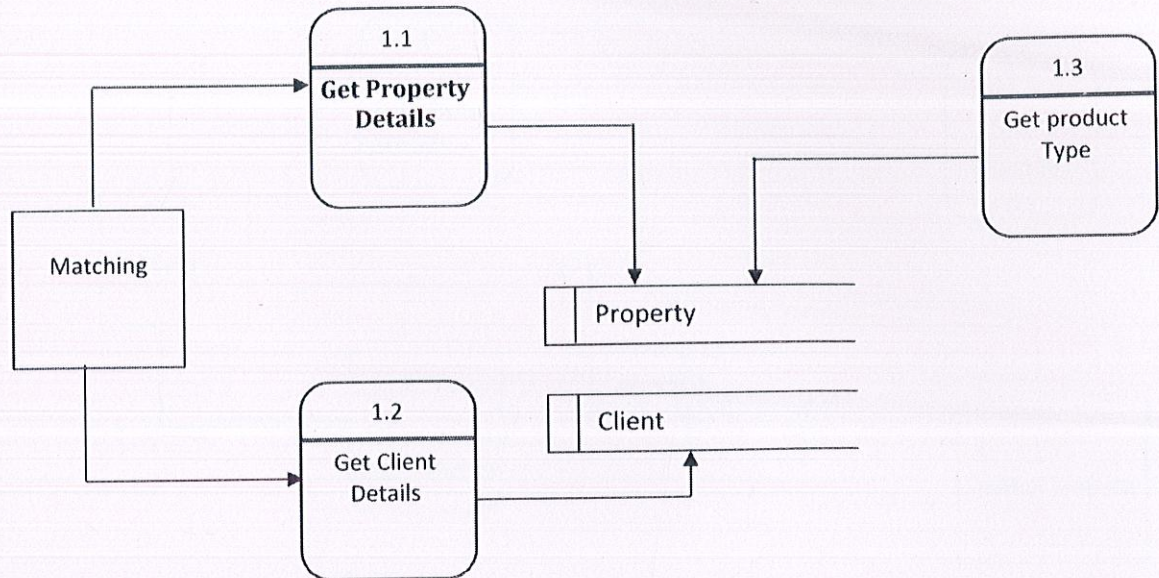


Fig-3

For matching property we will get the property and client details from the database and then the matching algorithm would match the property and after a successful match an offer letter would be sent to user.

Level 2 DFD:
(client registration)

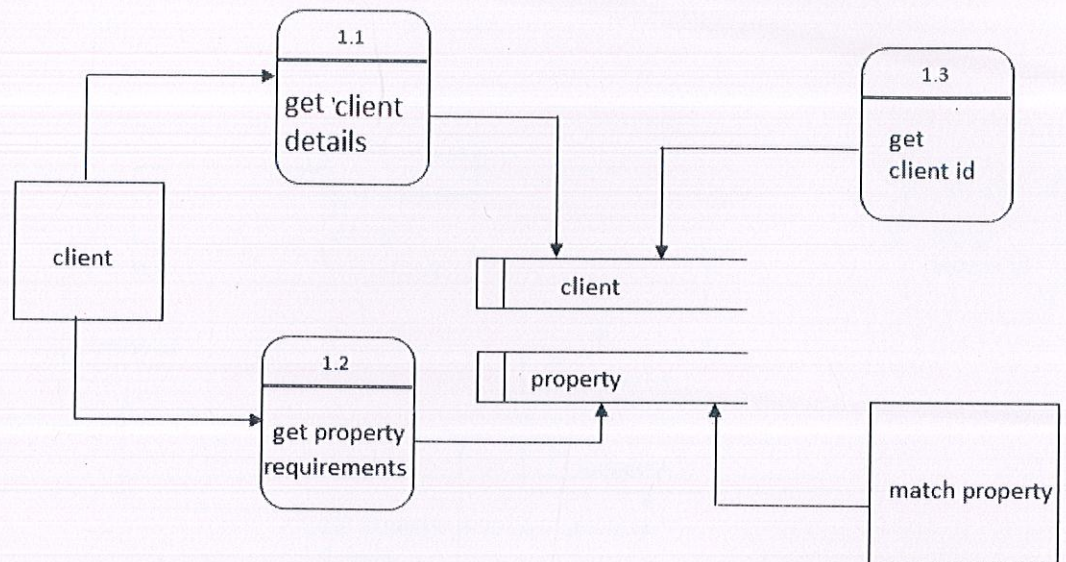


Fig-4

The client would enter his/her details and the type of property he is looking and after this the details would then be stored in the database and a client id would be generated and after successful generation of client id the match property function would then be executed.

Level 3 DFD :
(vendor registration)

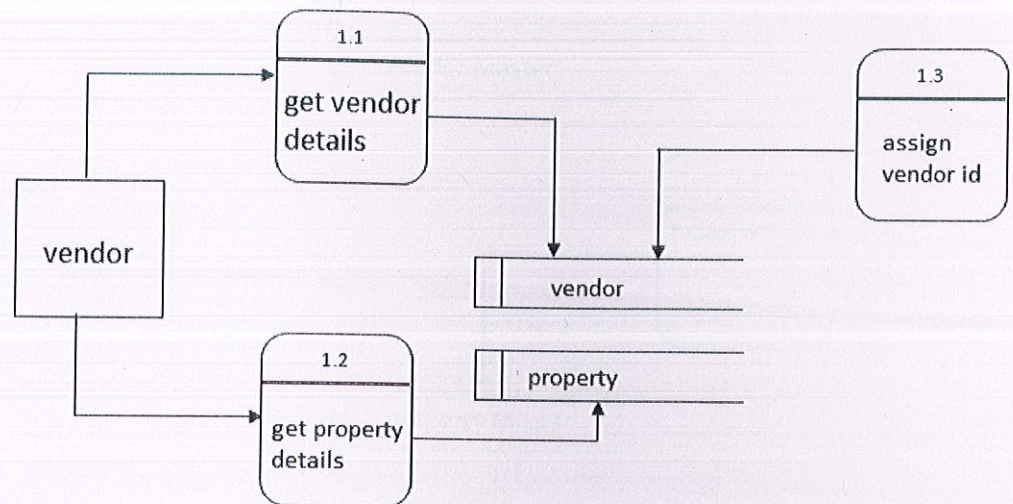


Fig-5

The vendor would register himself/herself by giving his/her details and the property he/she is selling. After the generation of the vendor id the details would then be stored in the database.

Level 4 DFD:
(property registration)

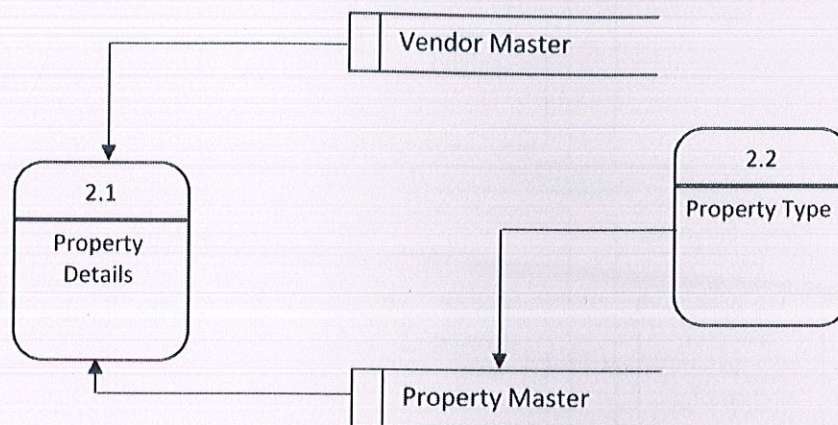


Fig- 6

The property details would be registered by the vendor and then stored in the database and it would be assigned a property type.

This property would then be available to view in the website.

3.5 System Architecture

High Level Solution Architecture

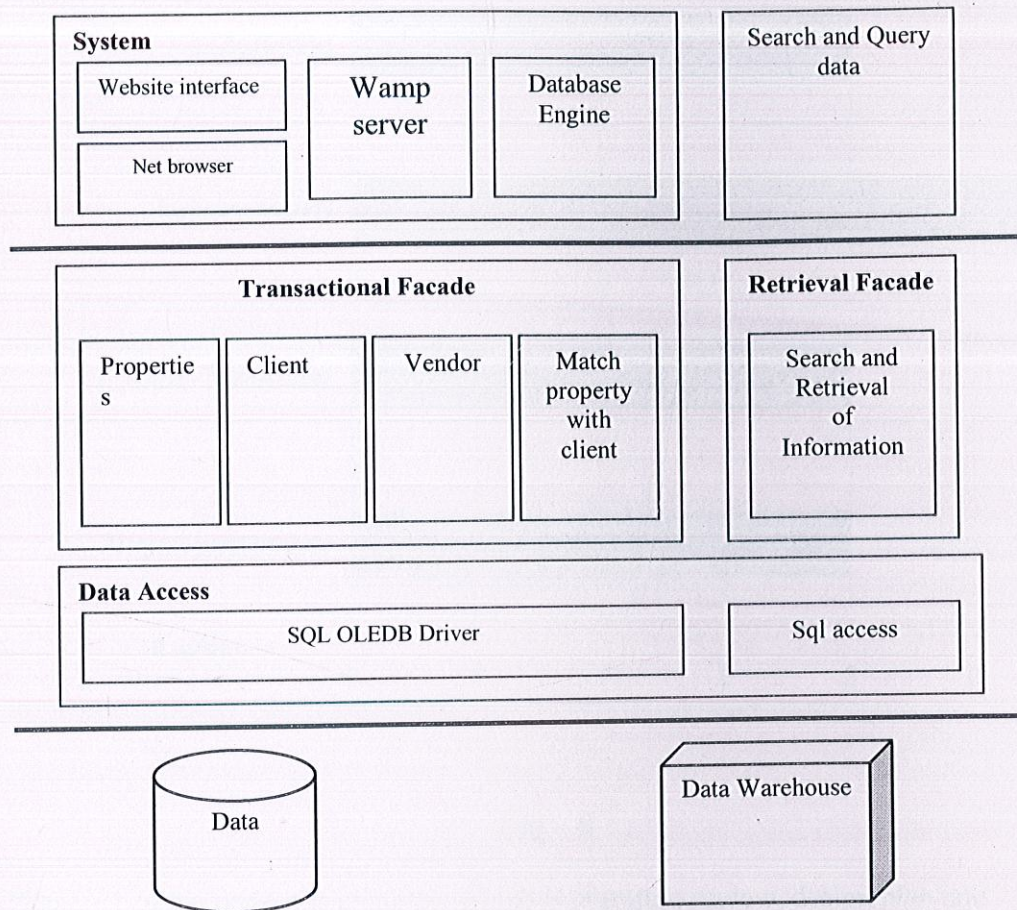


Fig - 7

High Level Solution Architecture of online real estate Property Management

3.6 Project Roadmap

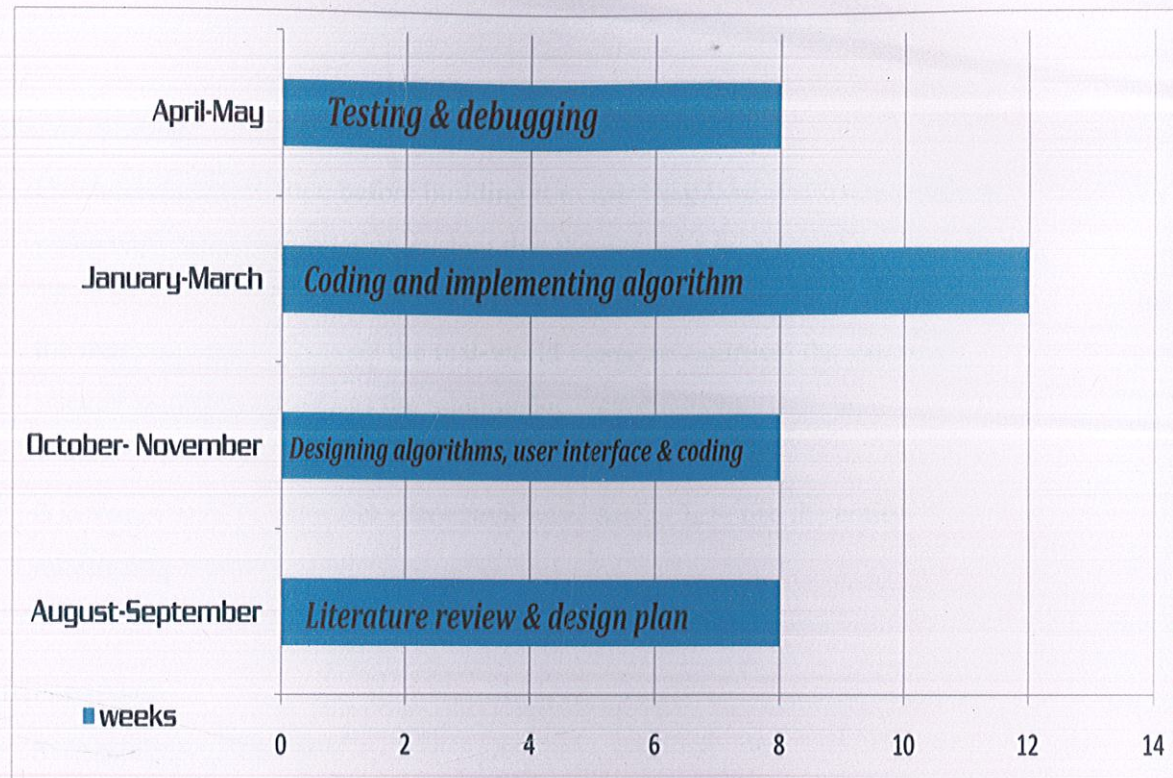


Fig - 8

Phase – 1 of the project is complete which is literature review, design plan and designing, coding and implementing algorithms and user interface.

Phase – 2 of the project was started on January and was finished by may.

3.7 Databases

A **database** is a structured collection of data. The data are typically organized to model relevant aspects of reality in a way that supports processes requiring the information.

Database design:

- Database design is done before building it to meet needs of end-users within a given application/information-system that the database is intended to support.
- The conceptual level is designed, which defines the over-all picture/view of the database, and reflects all the real-world elements (entities) the database intends to model, as well as the relationships among them.
- A common way to carry out conceptual level design is to use the entity-relationship model.

Requirements:

- Statement of the Problem
- Data Requirements
- Transaction Requirements
- Domain Objectives

Database Design Objectives:

- Represent the data and the relationships between data required from the database application.
- Provide a data model that supports any transactions required on the data.
- Specify a minimal design that is appropriately structured to achieve the stated performance requirements for the database application.

Relational Database Design Phases:

- Building the logical/conceptual data model.
- Building the physical database design.

3.8 TEST PLAN

The complete test strategy for the Real Estate Property Management System is to perform the following kinds of tests, in sequence:

1. **Component testing** of each component that makes up the Property Management System.
2. **Integration testing** of the Property Management System, to ensure the correct interworking of its components.
3. **Validation testing** of the Property Management System, to ensure that it works correctly in a pseudo-live environment.
4. **User acceptance testing** of the Property Management System, to ensure that its function is acceptable to its users.

Preconditions:-

The following items are required before testing can take place:

- A complete and coherent functional specification of the Property Management System expressed as use cases and usage scenarios
- Sufficient, suitable resources to carry out the testing
- A proper collection of test cases for each of the testing phase.

Roles and Responsibilities:-

The following roles are defined:

- A person responsible for planning and ensuring the smooth running of the test process.
- A person who is a tester who carries out the tests according to the test plan, and then reports the results
- A person who ensures that the tests are carried out successfully from a user perspective.

Threats to Testing:-

Potential threats to the testing process are as follows:

- **Insufficient resources available for testing.** Testing resources might lack to the real-world testing requirement of the website.
- **Unavailability of professional tester for testing.**
- **Lack of test cases to carry out the testing.**

Final testing and implementation:-

Rigorous testing is crucial to the implementation and maintenance of a reliable system that will create customer confidence.

Test results:

Summarized results of the system test, Include the completion status of each test case associated with the tests.

Test Analysis :

It summarizes the results of tests already conducted and identifies testing that remain to be conducted. Provide a brief summary of the project objectives, and identify the System Proponent and users.

3.9 Design goals and constraints

- **Performance**

→ No more than a 5-percent degradation in average query response is allowed while all concurrent user are using the system.

→ Processor utilization should not exceed 80 percent during all concurrent users are using the system.

- **Availability**

→ Because the system is accessed by more than one concurrent users so their should not be any single point of failure.

- **Reliability**

→ The system should be reliable to all the users.

- **Scalability**

- **Security**

→ Every resource in the system are defined by the role and privileged. System administrator assigned user role and privileged for their access rights.

→ The system should have anti-hacking programs.

→ The communication should be done with the help of encryption and decryption programs.

- **Interoperability**

→ The system should be interoperable that is able to use and exchange information efficiently between large heterogeneous networks.

- **Setup/Installation**

Setup and installation must not interrupt the system user's daily tasks and work flow.

CHAPTER 4

SCREENSHOTS AND CODING

4.1 Database code

```
drop DATABASE if exists testroom;

create DATABASE testroom;

use testroom;

create table users
(
    username char(20) not null,
    password char(16) not null,
    email char(50) not null,
    name varchar(100),
    surname varchar(100),
    user_type enum('A','U'),
    unique(email),
    primary key(username)
)
engine=InnoDB default character set utf8 collate
utf8_general_ci;

create table telephone
(
    tel_id int auto_increment,
    home char(10),
    mobile1 char(10) not null,
```



```

        mobile2 char(10),
        other char(10),
        user_id char(20) not null,
        primary key(tel_id),
        constraint phones_constr
        foreign key(user_id) references users(username)
        on delete cascade on update cascade
    )
    engine=InnoDB default character set utf8 collate
    utf8_general_ci auto_increment=1;
create table property
(
    prop_id int not null auto_increment,
    address char(100) not null,
    price int(15) not null,
    offer_type ENUM('S','L'),
    area int(10) not null,
    constr_date int(4),
    latitude float(10,7),
    longitude float(10,7),
    region char(100),
    Afloor int,
    views int(10) default '0',
    comments text,
    modified TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    propState ENUM('T','F'),

```



```

        user_id char(20) not null,
primary key(prop_id),

        foreign key(user_id) references users(username)

        on delete cascade on update cascade
)engine=InnoDB default character set utf8 collate utf8_general_ci auto_increment=1;

```

create table images

```

(
        image_id int not null auto_increment,
        prop_id int not null,
        filename varchar(255) not null,
        mime_type varchar(255) not null,
        image_size int(20),
        description text,
        primary key(image_id),
        foreign key(prop_id) references property(prop_id)

        on delete cascade on update cascade
)
engine=InnoDB default character set utf8 collate utf8_general_ci auto_increment=1;

```

create table fav_prop

```

(
        user_id char(20) not null,
        prop_id int not null,
        primary key(user_id,prop_id),
        constraint favProp_c1
        foreign key(user_id) references users(username)

```



```

        on delete cascade on update cascade,

        constraint favProp_c2

        foreign key(prop_id) references property(prop_id)

        on delete cascade on update cascade

    )

engine=innnoDB default character set utf8 collate utf8_general_ci;

create table facilities

(

    fac_id int not null auto_increment,

    facility varchar(100) not null,

    primary key(fac_id)

)

engine=innnoDB default character set utf8 collate utf8_general_ci auto_increment=1;

create table fac_prop

(

    prop_id int not null,

    fac_id int not null,

    primary key(fac_id,prop_id),

    constraint facProp_constr1

    foreign key(prop_id) references property(prop_id)

    on delete cascade on update cascade,

    constraint facProp_constr2

    foreign key(fac_id) references facilities(fac_id)

    on delete cascade on update cascade

```



```

)

engine=innnoDB default character set utf8 collate utf8_general_ci;

create table categories

(
    cat_id int not null auto_increment,
    category varchar(100) not null,
    primary key(cat_id)
)

engine=innnoDB default character set utf8 collate utf8_general_ci auto_increment=1;

create table cat_prop

(
    prop_id int not null,

cat_id int not null,

    constraint catProp_constr1
    foreign key(prop_id) references property(prop_id)
    on delete cascade on update cascade,
    constraint catProp_constr2
    foreign key(cat_id) references categories(cat_id)
    on delete cascade on update cascade ,

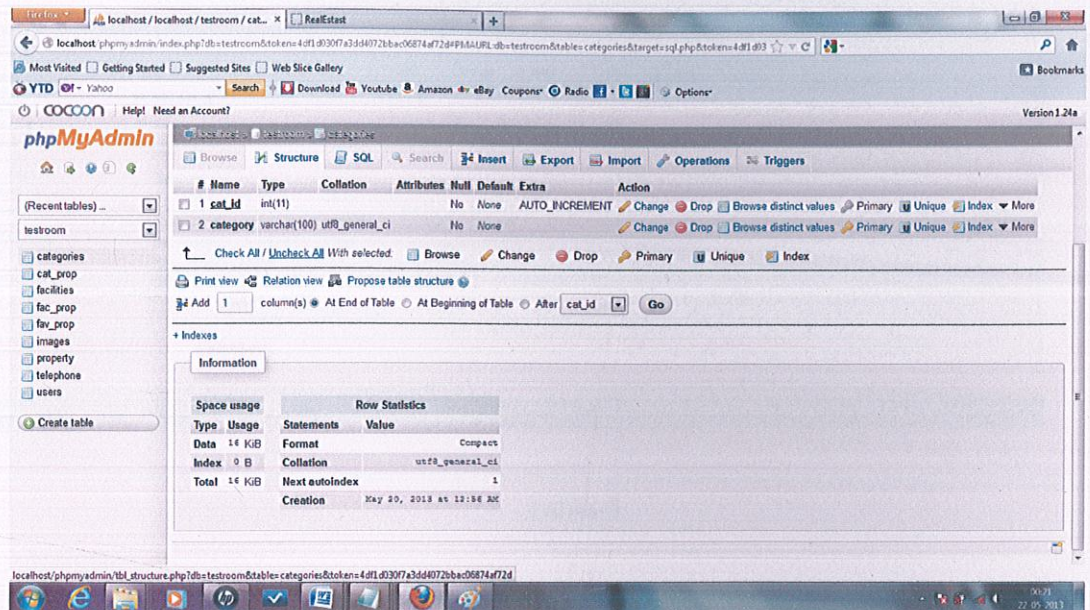
    primary key(cat_id,prop_id)
)

engine=innnoDB default character set utf8 collate utf8_general_ci;

```


4.2. Database Snapshots

Table – cat_prop



The screenshot displays the phpMyAdmin interface for a database named 'testroom'. The 'Structure' tab is active, showing the table 'cat_prop'. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	cat_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index More
2	category	varchar(100)	utf8_general_ci		No	None		Change Drop Browse distinct values Primary Unique Index More

The 'Information' tab is selected, showing the following details:

Space usage		Row Statistics	
Type	Usage	Statements	Value
Data	14 KiB	Format	Compact
Index	0 B	Collation	utf8_general_ci
Total	14 KiB	Next autoIndex	1
		Creation	May 20, 2018 at 12:58 AM

Table-2

Table – categories

localhost/phpmyadmin/structure.php?db=testroom&table=categories&tokens=4df1d30f7a3dd4072bba06874af72d4PMAURLdb=testroom&table=categories&tokens=4df1d30f7a3dd4072bba06874af72d4

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	cat_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index More
2	category	varchar(100)	utf8_general_ci		No	None		Change Drop Browse distinct values Primary Unique Index More

Space usage		Row Statistics	
Type	Usage	Statements	Value
Data	14 KiB	Format	Compact
Index	0 B	Collation	utf8_general_ci
Total	14 KiB	Next autoIndex	1
		Creation	May 20, 2013 at 12:18 AM

Table-3

Table – fac_prop

localhost/phpmyadmin/structure.php?db=testroom&table=fac_prop&tokens=4df1d30f7a3dd4072bba06874af72d4PMAURLdb=testroom&table=fac_prop&tokens=4df1d30f7a3dd4072bba06874af72d4

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	prop_id	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
2	fac_id	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext

Space usage		Row Statistics	
Type	Usage	Statements	Value
Data	14 KiB	Format	Compact
Index	14 KiB	Collation	utf8_general_ci
Total	32 KiB	Creation	May 20, 2013 at 12:18 AM

Table-4

Table – facilities

Structure of the 'facilities' table:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	fac_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index More
2	facility	varchar(100)	utf8_general_ci		No	None		Change Drop Browse distinct values Primary Unique Index More

Space usage:

Type	Usage
Data	14 KB
Index	9 B
Total	14 KB

Row Statistics:

Statements	Value
Format	Compact
Collation	utf8_general_ci
Next autoindex	1
Creation	May 20, 2013 at 12:56 AM

Table-5

Table – fav_prop

Structure of the 'fav_prop' table:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	user_id	char(20)	utf8_general_ci		No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext
2	prop_id	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index Spatial Fulltext

Space usage:

Type	Usage
Data	14 KB
Index	14 KB
Total	32 KB

Row Statistics:

Statements	Value
Format	Compact
Collation	utf8_general_ci
Creation	May 20, 2013 at 12:56 AM

Table-6

Table – images

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	image_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index More
2	prop_id	int(11)			No	None		Change Drop Browse distinct values Primary Unique Index More
3	filename	varchar(255)	utf8_general_ci		No	None		Change Drop Browse distinct values Primary Unique Index More
4	mime_type	varchar(255)	utf8_general_ci		No	None		Change Drop Browse distinct values Primary Unique Index More
5	image_size	int(20)			Yes	NULL		Change Drop Browse distinct values Primary Unique Index More
6	description	text	utf8_general_ci		Yes	NULL		Change Drop Browse distinct values Primary Unique Index More

Table-7

Table – property

	prop_id	address	price	offer_type	area	constr_date	latitude	longitude	region	floor	views	comments	modified	propState	user_id
Copy Delete	1	kaia\q\2\1\01001	100	L	8	1973	NULL	NULL	NULL	3	0	NULL	2013-05-21 22:32:35	T	user1001
Copy Delete	2	kaia\q\2\1\11001	416	L	190	1969	NULL	NULL	NULL	1	0	NULL	2013-05-21 22:32:35	T	user1001
Copy Delete	3	kaia\q\2\1\21001	505	L	113	1987	NULL	NULL	NULL	7	0	NULL	2013-05-21 22:32:36	T	user1001
Copy Delete	4	kaia\q\2\1\31001	515	L	174	1996	NULL	NULL	NULL	7	0	NULL	2013-05-21 22:32:36	T	user1001
Copy Delete	5	kaia\q\2\1\41001	1100	L	120	1976	NULL	NULL	NULL	7	0	NULL	2013-05-21 22:32:36	T	user1001
Copy Delete	6	kaia\q\2\1\51001	736	L	33	1980	NULL	NULL	NULL	5	0	NULL	2013-05-21 22:32:36	T	user1001
Copy Delete	7	kaia\q\2\1\61001	360	L	27	1980	NULL	NULL	NULL	4	0	NULL	2013-05-21 22:32:36	T	user1001
Copy Delete	8	kaia\q\2\1\71001	785	L	156	2004	NULL	NULL	NULL	2	0	NULL	2013-05-21 22:32:36	T	user1001
Copy Delete	9	kaia\q\2\1\81001	313	L	160	1997	NULL	NULL	NULL	4	0	NULL	2013-05-21 22:32:36	T	user1001
Copy Delete	10	kaia\q\2\1\91001	208	L	85	1985	NULL	NULL	NULL	0	0	NULL	2013-05-21 22:32:36	T	user1001
Copy Delete	11	kaia\q\2\1\101001	287	L	131	1993	NULL	NULL	NULL	2	0	NULL	2013-05-21 22:32:37	T	user1001
Copy Delete	12	kaia\q\2\1\111001	606	L	92	1991	NULL	NULL	NULL	10	0	NULL	2013-05-21 22:32:37	T	user1001
Copy Delete	13	kaia\q\2\1\121001	843	L	189	1969	NULL	NULL	NULL	4	0	NULL	2013-05-21 22:32:37	T	user1001
Copy Delete	14	kaia\q\2\1\131001	601	L	70	2009	NULL	NULL	NULL	3	0	NULL	2013-05-21 22:32:37	T	user1001
Copy Delete	15	kaia\q\2\1\141001	312	L	31	1965	NULL	NULL	NULL	0	0	NULL	2013-05-21 22:32:37	T	user1001
Copy Delete	16	kaia\q\2\1\151001	767	L	152	1984	NULL	NULL	NULL	10	0	NULL	2013-05-21 22:32:37	T	user1001
Copy Delete	17	kaia\q\2\1\161001	1043	L	133	1961	NULL	NULL	NULL	10	0	NULL	2013-05-21 22:32:37	T	user1001

Table-8

Table – telephone

The screenshot shows the phpMyAdmin interface for the 'testroom' database. The 'telephone' table is selected, and its structure is displayed. The table has columns: tel_id, home, mobile1, mobile2, other, and user_id. The data is as follows:

tel_id	home	mobile1	mobile2	other	user_id
1	NULL	0997654321	NULL	NULL	aman

Table-9

Table – users

The screenshot shows the phpMyAdmin interface for the 'testroom' database. The 'users' table is selected, and its structure and data are displayed. The table has columns: username, password, email, name, surname, and user_type. The data is as follows:

username	password	email	name	surname	user_type
aman	aman	aman@gmail.com	NULL	NULL	A
user1001	password1001	user1001@gmail.com	NULL	NULL	NULL
user1002	password1002	user1002@gmail.com	NULL	NULL	NULL
user1003	password1003	user1003@gmail.com	NULL	NULL	NULL
user1004	password1004	user1004@gmail.com	NULL	NULL	NULL
user1005	password1005	user1005@gmail.com	NULL	NULL	NULL
user1006	password1006	user1006@gmail.com	NULL	NULL	NULL
user1007	password1007	user1007@gmail.com	NULL	NULL	NULL
user1008	password1008	user1008@gmail.com	NULL	NULL	NULL
user1009	password1009	user1009@gmail.com	NULL	NULL	NULL
user1010	password1010	user1010@gmail.com	NULL	NULL	NULL
user1011	password1011	user1011@gmail.com	NULL	NULL	NULL
user1012	password1012	user1012@gmail.com	NULL	NULL	NULL
user1013	password1013	user1013@gmail.com	NULL	NULL	NULL
user1014	password1014	user1014@gmail.com	NULL	NULL	NULL
user1015	password1015	user1015@gmail.com	NULL	NULL	NULL
user1016	password1016	user1016@gmail.com	NULL	NULL	NULL

Table-10

4.3. User interface coding

-ajax.php

```
<?php
include('includes.php');
if(isset($_POST['oldPassword'])) $oldPas=$_POST['oldPassword'];
else $oldPas=NULL;
if(isset($_POST['newPassword1'])) $pas1=$_POST['newPassword1'];
else $pas1=NULL;
if(isset($_POST['newPassword2'])) $pas2=$_POST['newPassword2'];
else $pas2=NULL;
if(isset($_POST['newEmail'])) $newMail=$_POST['newEmail']; else
$newMail=NULL;
if(isset($_POST['newMob1'])) $newMob1=$_POST['newMob1']; else
$newMob1=NULL;
if(isset($_POST['newName'])) $newName=$_POST['newName']; else
$newName=NULL;
if(isset($_POST['newLast'])) $newLast=$_POST['newLast']; else
$newLast=NULL;
if(isset($_POST['newMob2'])) $newMob2=$_POST['newMob2']; else
$newMob2=NULL;
if(isset($_POST['newHome'])) $newHome=$_POST['newHome'];
else $newHome=NULL;
if(isset($_POST['newOthr'])) $newOthr=$_POST['newOthr']; else
$newOthr=NULL;
if(isset($_POST['privilege'])) $priv=$_POST['privilege']; else
$priv=NULL;
$user=strval($_GET['user']);
try{
    check_valid_user();
    if(!$oldPas&& !$pas1&& !$pas2&& !$newMail&& !$newMob1&& !$n
ewName&& !$newLast&& !$newMob2&& !$newHome&& !$newOthr&
& !$priv)
    {
        dispHeader("");
        showUserProfile($user);
        echo "<br /><br />";
        displayUserOptions($user,$type);
        dispFooter();
    }
}
else
```



```

{
    if($type=="User")
    {
        if($oldPas && $pas1 && $pas2)
        {
            if($pas1 != $pas2)
            {
                throw new Exception('The passwords
you entered are different!');
            }
            else
            {
                $new=db_check("users","username","password","$user","$oldPa
s");
                if($new==false) throw new
Exception('Wrong password');
                db_update("users","username","password","$user","$pas1");
                $message="password changed!";
            }
        }
        else if(($oldPas && (!$pas1 || !$pas2)) || ($pas1 &&
(!$oldPas || !$pas2)) || ($pas2 && (!$oldPas || !$pas1)))
        {
            throw new Exception('form not filled
correctly,try again');
        }
    }
    else
    {
        if($pas1 && $pas2)
        {
            if($pas1 != $pas2)
            {
                throw new Exception('password not
same');
            }
            else
            {
                db_update("users","username","password","$user","$pas1");
                $message="password of $user changed!";
            }
        }
        else if((!$pas1 && $pas2) || ($pas1 && !$pas2))
    }
}

```



```

        {
            throw new Exception('form not filled
correctly,try again');
        }
    }
    if($newMail)
    {
        $ret=valid_email($newMail);
        if($ret==false) throw new Exception('The email
address you entered is invalid');

        db_update("users","username","email","$user","$newMail");
        if($val_user!=$user) $message="email of $user
changed!";
        else $message="email changed!";
    }
    if($newMob1)
    {
        db_update("telephone","user_id","mobile1","$user",$newM
ob1);
        if($val_user!=$user) $message="mobile no. of $user
changed!";
        else $message="Your mobile has changed!";
    }
    if($newName)
    {
        db_update("users","username","name","$user","$newName");
        if($val_user!=$user) $message="name of $user
changed!";
        else $message="name changed!";
    }
    if($newLast)
    {
        db_update("users","username","surname","$user","$newLast");
        if($val_user!=$user) $message="surname of $user
changed!";
        else $message="Surname changed!";
    }
    if($newMob2)
    {
        db_update("telephone","user_id","mobile2","$user",$newMob2);
        if($val_user!=$user) $message="mobile no of $user
changed!";
    }

```



```

        else $message=""
            ["!";
        }
        if($newHome)
        {
            db_update("telephone","user_id","home","$user",$newHome);
            if($val_user!=$user) $message="Landline No
changed";
            else $message="telephone changed!";
        }
        if($newOthr)
        {
            db_update("telephone","user_id","other","$user",$newOthr);
            if($val_user!=$user) $message="The other phone of
$user changed!";
            else $message="Other Phone Changed";
        }
        if($priv=="Admin")
        {
            db_update("users","username","user_type","$user","A");
            $message="user $user become Admin !";
        }
        else if($priv=="User")
        {
            db_update("users","username","user_type","$user","U");
            $message=" $user Low User!";
        }
        if($type=="Admin") dispHeader("User Profile $user");
        else dispHeader("To Profile");
        showUserProfile($user);
        echo "<br />";
        echo $message;
        echo "<br /><br />";
        displayUserOptions($user,$type);
        dispFooter();
        $message="";
    }
}
catch(Exception $e)
{
    dispHeader("error with the profile $user:");
    echo $e->getMessage();
    dispURL('editUser.php?user='.$user, 'Edit Profile');
}

```



```

        dispFooter();
        exit;
    }
?>

```

- delUser.php

```

<?php
require_once("includes.php");
try{
    check_valid_user();
    $user=strval($_GET['user']);
    db_del_user($user);
    dispHeader('Deletion successful',2);
    echo "user $user deleted".<br />';
    dispURL('member.php', 'Members page');
    dispFooter();
}
catch(Exception $e)
{
    dispHeader('Error');
    echo $e->getMessage();
    dispURL('member.php', 'Members page');
    dispFooter();
    exit;
}
?>

```

- displayUsers.php

```

<?php
include('includes.php');
try
{
    check_valid_user();
    dispHeader("");
    dispCurrUsers();
    dispFooter();
}

```



```

catch(Exception $e)
{
    dispHeader("Error:");
    echo $e->getMessage();
    dispFooter();
    exit;
}
?>

```

- editCategories.php

```

<?php
include('includes.php');
if(isset($_POST['newCat'])) $newCat=$_POST['newCat']; else
$newCat=NULL;
if(isset($_POST['category'])) $delCat=$_POST['category']; else
$delCat=NULL;
try
{
    check_valid_user();
    if(filledOut($delCat))
    {
        dispHeader('Delete category');
        foreach($delCat as $cat)
        {
            echo "$cat <br />";
            db_del_cat($cat);
        }
        dispCategoriesSettings();
        dispFooter();
    }
    else if(filledOut($newCat))
    {
        $newCat=$_POST['newCat'];
        db_insert1('categories','category','$newCat');
        dispHeader("new category $newCat");
        dispCategoriesSettings();
        dispFooter();
    }
    else
    {
        dispHeader("");
    }
}

```



```

        dispCategoriesSettings();
        dispFooter();
    }
}
catch(Exception $e)
{
    dispHeader("Error:");
    echo $e->getMessage();
    dispFooter();
    exit;
}
?>

```

- editFacilities.php

```

<?php
include('includes.php');
if(isset($_POST['newFac'])) $newFac=$_POST['newFac']; else
$newFac=NULL;
if(isset($_POST['facility'])) $delFac=$_POST['facility']; else
$delFac=NULL;
try
{
    check_valid_user();
    if(filledOut($delFac))
    {
        dispHeader('Delete facility');
        foreach($delFac as $fac)
        {
            echo "$fac <br />";
            db_del_fac($fac);
        }
        dispFooter();
    }
    else if(filledOut($newFac))
    {
        $newFac=$_POST['newFac'];
        db_insert1('facilities','facility',"{$newFac}");
        dispHeader("new facility $newFac");
        dispFacilitiesSettings();
        dispFooter();
    }
}
else

```



```

        {
            dispHeader("");
            dispFacilitiesSettings();
            dispFooter();
        }
    }
    catch(Exception $e)
    {
        dispHeader("Error:");
        echo $e->getMessage();
        dispFooter();
        exit;
    }
?>

```

- editProperty.php

```

<?php
include('includes.php');
if(isset($_POST['newAddress']))
$newAddress=trim($_POST['newAddress']); else $newAddress=NULL;
if(isset($_POST['newType'])) $newType=$_POST['newType']; else
$newType=NULL;
if(isset($_POST['newArea'])) $newArea=$_POST['newArea']; else
$newArea=NULL;
if(isset($_POST['newRegion']))
$newRegion=trim($_POST['newRegion']); else $newRegion=NULL;
if(isset($_POST['newAfloor'])) $newAfloor=$_POST['newAfloor']; else
$newAfloor=NULL;
if(isset($_POST['newConstrDate']))
$newConstrDate=$_POST['newConstrDate']; else
$newConstrDate=NULL;
if(isset($_POST['newPrice'])) $newPrice=$_POST['newPrice']; else
$newPrice=NULL;
if(isset($_POST['typos'])) $typos=$_POST['typos']; else $typos=NULL;
if(isset($_POST['category'])) $category=$_POST['category']; else
$category=NULL;
if(isset($_POST['delete'])) $delete=$_POST['imgId']; else
$delete=NULL;
if(isset($_POST['add'])) $add=$_POST['add']; else $add=NULL;

```



```

if(isset($_POST['file_count'])) $file_count=$_POST['file_count']; else
$file_count=0;
if(isset($_POST['description'])) $description=$_POST['description']; else
$description="";
if(isset($_POST['comments'])) $comments=$_POST['comments']; else
$comments=NULL;
if(isset($_POST['facilities'])) $facilities=$_POST['facilities']; else
$facilities=NULL;
if(isset($_POST['deleteProperty']))
$deleteProperty=$_POST['deleteProperty']; else $deleteProperty=NULL;
$message="";
$propId=strval($_GET['propId']);
try{
    check_valid_user();
    if(isset($add))
    {
        if($add=="addfav")
        {
            db_insert('fav_prop','user_id','prop_id',"$_val_user",$propId);
            $message="<br />Mark as favorite <br />";
        }
        else
        {
            db_delete('fav_prop','user_id','prop_id',"$_val_user",$propId);
            $message="<br />Unmark <br />";
        }
    }
    if(isset($category))
    {
        $selectCat="select *from categories where
        category='$_category'";
        $cat_id=db_execute($selectCat,"selectCat");
        $cat=mysql_fetch_array($cat_id);
        db_update("cat_prop","prop_id","cat_id",$propId,$cat['cat_id']);
        $message="category renewed";
    }
    if(isset($typos))
    {
        if($typos=="pwlhsh") $typos="S";
        else $typos="L";
        db_update("property","prop_id","offer_type",$propId,"$typos");
        $message="offer type renewed!";
    }
}

```



```

if(isset($newAddress))
{
    $flag=false;
    if(isset($_POST['latitude']) && $_POST['latitude']!="0")
    {
        if(isset($_POST['longitude']) &&
        $_POST['longitude']!="0")
        {
            $latitude=$_POST['latitude'];
            $longitude=$_POST['longitude'];
            $flag=true;
        }
    }
    if($flag==false)
    {
        throw new Exception("There was a problem with
        google maps.");
    }
    db_update("property","prop_id","latitude",$propId,$latitude);
    db_update("property","prop_id","longitude",$propId,$longitude);
    db_update("property","prop_id","address",$propId,"$newAddress'
");
    $message="address changed!";
}
if(isset($newArea))
{
    db_update("property","prop_id","area",$propId,$newArea);
    $message="area changed!";
}
if(isset($newRegion))
{
    db_update("property","prop_id","region",$propId,"$newRegion");
    $message="region changed!";
}

if(isset($newAfloor))
{
    db_update("property","prop_id","Afloor",$propId,$newAfloor);
    $message="floors changed";
}
if(isset($newConstrDate))
{

```



```

        db_update("property","prop_id","constr_date",$propId,$newConst
rDate);
        $message="The year of property changed!";
    }
    if(isset($newPrice))
    {
        db_update("property","prop_id","price",$propId,$newPrice);
        $message="The value of the property changed!";
    }
    if(isset($facilities))
    {
        db_delete1("fac_prop","prop_id",$propId);
        $facilities=$_POST['facilities'];
        foreach($facilities as $fac)
        {
            $selectFac="select *from facilities where
facility='$fac'";
            $fac_id=db_execute($selectFac,'select3');
            $fac=mysql_fetch_array($fac_id);

            db_insert('fac_prop','prop_id','fac_id',$propId,$fac['fac_id']);
        }
    }
    if(isset($comments))
    {
        $comments=trim($comments);
        db_update("property","prop_id","comments",$propId,"$comments
");
        $message="The information of the property changed!";
    }
    for($i=0; $i<$file_count; $i++)
    {
        if(isset($_FILES["new_file"]["error"][$i]))
        {
            $phMessage="you uploaded a file !!!\n";
            if ($_FILES["new_file"]["error"][$i] > 0)
            {
                throw new Exception("Error: " .
$_FILES["new_file"]["error"][$i] . "<br />");
            }
            $phMessage.="Upload: " .
$_FILES["new_file"]["name"][$i] . "<br />";

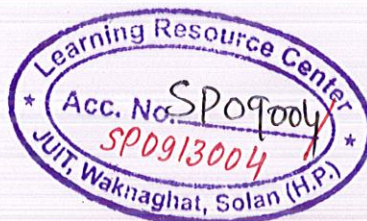
```



```

$phMessage.="Type: " .
$_FILES["new_file"]["type"][$i] . "<br />";
$phMessage.="Size: " .
($_FILES["new_file"]["size"][$i] / 1024) . " Kb<br
/>";
$phMessage.="Stored in: " .
$_FILES["new_file"]["tmp_name"][$i];
$uploadName=$_FILES["new_file"]["name"][$i];
$mimeType=$_FILES["new_file"]["type"][$i];
$size=(($_FILES["new_file"]["size"][$i] / 1024);
$stored=$_FILES["new_file"]["tmp_name"][$i];
$pwd=getcwd();
$photosD="photos/".$propId."/";
$middle="/";
$findRows="select * from images where
prop_id=$propId";
$result=db_execute($findRows,"exists");
$rows=mysql_num_rows($result);
$filename=$photosD.$propId."-".$rows."-
".$_FILES["new_file"]["name"][$i];
$destination=$pwd.$middle.$filename;
$phMessage.="<br /> Destination directory:
".$destination;
$phMessage.="<br /> Source directory: ".$stored;
$phMessage.="<br /> filename: ".$filename;
$phMessage.="<br /> file_count: ".$file_count;
$phMessage.="<br /> description: ".$description[$i];
$filename=addslashes($filename);
$phMessage.="<br /> Actual filename that is insert
into database: ".$filename;
$description_=trim($description[$i]);
if($description_=="description") $description_="";
$insert="insert into images
(prop_id,filename,mime_type,image_size,description) values
($propId,'$filename','$mimeType',$size,'$description_')";
if(!copy($stored,$destination)) throw new
Exception("Failed to copy the file...");
db_execute($insert,"insert image");
$message="image $uploadName inserted.";
}
}
if(isset($delete))
{

```




```

        $selectFile="select * from images where image_id=$delete";
        $result=db_execute($selectFile,"selectFile");
        $row = mysql_fetch_array($result);
        $pwd=getcwd();
        $filename=$pwd."/".$row['filename'];
        if(unlink($filename)==false) throw new Exception("Unable
        to delete file".$row['filename']);
        db_delete("images","image_id","prop_id",$delete,$propId);
    }
    if(isset($deleteProperty))
    {
        db_del_prop($propId);
        dispHeader("delete ad");
        echo "<br /><br /> ad deleted";
        dispFooter();
    }
    else
    {
        dispHeader("Edit ad",'2');
        showProperty($propId);
        echo "<br />";
        echo $message;
        echo "<br /><br />";
        dispPropOptions($propId);
        showPropPhotosDel($propId);
        dispFooter();
    }
}
catch(Exception $e)
{
    dispHeader("There was an error processing the property
    $propId:");
    echo $e->getMessage();
    dispURL('editProperty.php?propId='.$propId, 'Edit Profil');
    dispFooter();
    exit;
}
?>

```


- editUser.php

```
<?php
include('includes.php');
if(isset($_POST['oldPassword'])) $oldPas=$_POST['oldPassword']; else
$oldPas=NULL;
if(isset($_POST['newPassword1'])) $pas1=$_POST['newPassword1'];
else $pas1=NULL;
if(isset($_POST['newPassword2'])) $pas2=$_POST['newPassword2'];
else $pas2=NULL;
if(isset($_POST['newEmail'])) $newMail=$_POST['newEmail']; else
$newMail=NULL;
if(isset($_POST['newMob1'])) $newMob1=$_POST['newMob1']; else
$newMob1=NULL;
if(isset($_POST['newName'])) $newName=$_POST['newName']; else
$newName=NULL;
if(isset($_POST['newLast'])) $newLast=$_POST['newLast']; else
$newLast=NULL;
if(isset($_POST['newMob2'])) $newMob2=$_POST['newMob2']; else
$newMob2=NULL;
if(isset($_POST['newHome'])) $newHome=$_POST['newHome']; else
$newHome=NULL;
if(isset($_POST['newOthr'])) $newOthr=$_POST['newOthr']; else
$newOthr=NULL;
if(isset($_POST['privilege'])) $priv=$_POST['privilege']; else
$priv=NULL;
$user=strval($_GET['user']);
try{
    check_valid_user();
    if(!$oldPas&& !$pas1&& !$pas2&& !$newMail&& !$newMob1&&
    !$newName&& !$newLast&& !$newMob2&& !$newHome&& !$newOthr
    && !$priv)
    {
        dispHeader("");
        showUserProfile($user);
        echo "<br /><br />";
        displayUserOptions($user,$type);
        dispFooter();
    }
    else
    {
```



```

if($type=="User")
{
    if($oldPas && $pas1 && $pas2)
    {
        if($pas1 != $pas2)
        {
            throw new Exception('The passwords
you entered are different!');
        }
        else
        {
            $new=db_check("users","username","password","$user","$oldPa
s");
            if($new==false) throw new
Exception('Wrong password');
            db_update("users","username","password","$user","$pas1");
            $message="password changed!";
        }
    }
    else if(($oldPas && (!$pas1 || !$pas2)) || ($pas1 &&
(!$oldPas || !$pas2)) || ($pas2 && (!$oldPas || !$pas1)))
    {
        throw new Exception('form not filled
correctly,try again');
    }
}
else
{
    if($pas1 && $pas2)
    {
        if($pas1 != $pas2)
        {
            throw new Exception('password not
same');
        }
        else
        {
            db_update("users","username","password","$user","$pas1");
            $message="password of $user changed!";
        }
    }
    else if((!$pas1 && $pas2) || ($pas1 && !$pas2))
    {

```



```

                                throw new Exception('form not filled
correctly,try again');
                                }
                                }
                                if($newMail)
                                {
                                    $ret=valid_email($newMail);
                                    if($ret==false) throw new Exception('The email
address you entered is invalid');
                                    db_update("users","username","email","$user","$newMail");
                                    if($val_user!=$user) $message="email of $user
changed!";
                                    else $message="email changed!";
                                }
                                if($newMob1)
                                {
                                    db_update("telephone","user_id","mobile1","$user",$newMob1);
                                    if($val_user!=$user) $message="mobile no. of $user
changed!";
                                    else $message="Your mobile has changed!";
                                }
                                if($newName)
                                {
                                    db_update("users","username","name","$user","$newName");
                                    if($val_user!=$user) $message="name of $user
changed!";
                                    else $message="name changed!";
                                }
                                if($newLast)
                                {
                                    db_update("users","username","surname","$user","$newLast");
                                    if($val_user!=$user) $message="surname of $user
changed!";
                                    else $message="Surname changed!";
                                }
                                if($newMob2)
                                {
                                    db_update("telephone","user_id","mobile2","$user",$newMob2);
                                    if($val_user!=$user) $message="mobile no of $user
changed!";
                                    else $message=""
                                    ][!";
                                }

```



```

        if($newHome)
        {
            db_update("telephone","user_id","home",""$user""",$newHome);
            if($val_user!=$user) $message="Landline No
            changed";
            else $message="telephone changed!";
        }
        if($newOthr)
        {
            db_update("telephone","user_id","other",""$user""",$newOthr);
            if($val_user!=$user) $message="The other phone of
            $user changed!";
            else $message="Other Phone Changed";
        }
        if($priv=="Admin")
        {
            db_update("users","username","user_type",""$user""","A");
            $message="user $user become Admin !";
        }
        else if($priv=="User")
        {
            db_update("users","username","user_type",""$user""","U");
            $message=" $user Low User!";
        }
        if($type=="Admin") dispHeader("User Profile $user");
        else dispHeader("To Profile");
        showUserProfile($user);
        echo "<br />";
        echo $message;
        echo "<br /><br />";
        displayUserOptions($user,$type);
        dispFooter();
        $message="";
    }
}
catch(Exception $e)
{
    dispHeader("error with the profile $user:");
    echo $e->getMessage();
    dispURL('editUser.php?user='.$user, 'Edit Profile');
    dispFooter();
    exit;
}

```


?>

- favAdvs.php

```
<?php
include('includes.php');
if(isset($_POST['remProperty'])) $remProp=$_POST['remProperty'];
else $remProp=NULL;
if(isset($_GET['page'])) $page=$_GET['page']; else $page=1;
try
{
    check_valid_user();
    if(isset($remProp))
    {
        foreach($remProp as $prop)
        {
            db_delete('fav_prop','prop_id','user_id',$prop,"$val_user");
        }
    }
    $select="select * from fav_prop where user_id='$val_user'";
    $result=db_execute($select,"select");
    $message="select distinct
property.prop_id,address,price,offer_type,area,views,category from
property,categories,cat_prop where (";
    if(mysql_num_rows($result)>0)
    {
        while($row=mysql_fetch_assoc($result))
        {
            $message.=" property.prop_id=".$row['prop_id']."
or";
        }
        $message=substr($message,0,-2);
        $message.=")";
    }
    $message.=" and property.prop_id=cat_prop.prop_id and
categories.cat_id=cat_prop.cat_id";
    //echo $message;
    check_valid_user();
    dispHeader("");
    echo "<div class='header-bar-full'><h1 class='blue'>My Favourite
Ads</h1></div>";
    if(!(mysql_num_rows($result)>0)) echo "
<div class='content-box-1'>
```



```

<div class=\"content-box-1-top\"></div>
<div class=\"content-box-1-middle\">
<div class=\"content-box-1-content\">
<h3> You have not put any advertisement on your favorite! </ h3>
<p> to add a favorite ad simply click the Add button! </ P>
    <a href=member.php> Back to Profile </ a> <br/>
<a href=main.php> Back to Home </ a>
</div>
</div>
<div class=\"content-box-1-bottom\">&nbsp;</div>
</div>    ";
        else propertySearch($message,"Favourites",$page);
        dispFooter();
    }
    catch(Exception $e)
    {
        dispHeader("Error:");
        echo $e->getMessage();
        dispFooter();
        exit;
    }
?>

```

- homeAdvertise.php

```

<?php
include('includes.php');
try
{
    check_valid_user();
    dispHeader("");
    dispHomeAdvertise();
    dispFooter();
}
catch(Exception $e)
{
    dispHeader('Error');
    echo $e->getMessage();
    dispFooter();
    exit;
}

```


?>

- homeAdvertiseNew.php

```
<?php
include('includes.php');
try
{
    check_valid_user();
    dispHeader("");
    dispHomeAdvertise();
    dispFooter();
}
catch(Exception $e)
{
    dispHeader('Error');
    echo $e->getMessage();
    dispFooter();
    exit;
}
?>
```

- homeSearch.php

```
<?php
include('includes.php');
try
{
    $_SESSION['prevPage']="homeSearch.php";
    check_valid_user(1);
    dispHeader("");
    dispHomeSearch();
    dispFooter();
}
catch(Exception $e)
{
    dispHeader("Error:");
    echo $e->getMessage();
    dispFooter();
    exit;
}
```


?>

- homeSearchRes.php

```
<?php
require_once('includes.php');
if(isset($_POST['delProperty'])) $delProp=$_POST['delProperty']; else
$delProp=NULL;
if(isset($_POST['mainQuery'])) $mainQuery=$_POST['mainQuery'];
else $mainQuery=NULL;
if(isset($_GET['page'])) $page=$_GET['page']; else $page=1;
if(isset($_GET['results'])) $results=$_GET['results']; else $results=15;
try
{
    if(isset($mainQuery))
    {
        $message="select distinct
property.prop_id,address,price,offer_type,area,views,category from
property,categories,cat_prop,fac_prop,facilities";
        $message.=" where (property.address like '%$mainQuery%'
or categories.category like '%$mainQuery%' or facilities.facility like
'%$mainQuery%') ";
        $message.=" and property.prop_id=cat_prop.prop_id and
categories.cat_id=cat_prop.cat_id and facilities.fac_id=fac_prop.fac_id
and property.prop_id=fac_prop.prop_id";
        $message=$message.";";
    }
    else
    {
        if($_SESSION['prevPage']=="homeSearchRes.php")
        {
            $message=$_SESSION['query'];
            if(isset($delProp))
            {
                foreach($delProp as $prop)
                {
                    db_del_prop($prop);
                }
            }
        }
        else
        {

```



```

$_SESSION['prevPage']="homeSearchRes.php";
$message="select distinct
property.prop_id,address,price,offer_type,area,views,category from
property,categories,cat_prop";
if(isset($_POST['facilities']))
$message.=",fac_prop,facilities";
$message.=" where property.propState='T'";
if(isset($_POST['typos']))
{
    $message.=" and";
    $typos=$_POST['typos'];
    $message.=" (";
    foreach ($typos as $typ)
    {
        $message.=" property.offer_type='$typ'
or";

    }
    $message=substr($message,0,-2);
    $message.=")";
}
if(isset($_POST['category']))
{
    $message=$message." and";
    $category=$_POST['category'];
    $message.=" (";
    foreach ($category as $cat)
    {
        $message.=" categories.category='$cat'
or";

    }
    $message=substr($message,0,-2);
    $message.=")";
}
if(isset($_POST['low_price']))
{
    $value=$_POST['low_price'];
    if($value!="nolimit")
    {
        $message.=" and
property.price>=$value";
    }
}
if(isset($_POST['high_price']))

```



```

        {
            $value=$_POST['high_price'];
            if($value!="nolimit")
            {
                $message.=" and
property.price<=$value";
            }
        }
        if(isset($_POST['low_area']))
        {
            $value=$_POST['low_area'];
            if($value!="nolimit")
            {
                $message.=" and property.area>$value";
            }
        }
        if(isset($_POST['high_area']))
        {
            $value=$_POST['high_area'];
            if($value!="nolimit")
            {
                $message.=" and property.area<$value";
            }
        }
        if(isset($_POST['Afloor']))
        {
            $message=$message." and";
            $Afloor=$_POST['Afloor'];
            $message.=" (";
            foreach($Afloor as $fl)
            {
                $message.=" property.Afloor=$fl or";
            }
            $message=substr($message,0,-2);
            $message.=" )";
        }
        if(isset($_POST['etos_katask']))
        {
            $value=$_POST['etos_katask'];
            if($value!="nolimit")
            {
                $message.=" and
property.constr_date>$value";
            }
        }
    }

```



```

    }
}
if(isset($_POST['facilities']))
{
    $message=$message." and";
    $facility=$_POST['facilities'];
    $message.=" (";
    foreach ($facility as $fac)
    {
        $message.=" facilities.facility='$fac' or";
    }
    $message=substr($message,0,-2);
    $message.=")";
    $message.=" and
property.prop_id=fac_prop.prop_id and
facilities.fac_id=fac_prop.fac_id";
}
$message.=" and property.prop_id=cat_prop.prop_id
and categories.cat_id=cat_prop.cat_id ORDER BY property.prop_id
DESC";
    $message=$message."";
    $_SESSION['query']=$message;
}

}
if(isset($message)) $_SESSION['message']=$message;
else $message=$_SESSION["message"];
check_valid_user(1);
dispHeader("");
echo "<div class='header-bar-full'><h1 class='blue'>Search
Results</h1></div>";
$result=db_execute($message,"searchRes");
if(mysql_num_rows($result)==0) echo "No ads on the criteria you
selected!";
else propertySearch($message,"Delete",$page,$results);
dispFooter();
}
catch (Exception $e)
{
    dispHeader('Error:');
    echo $e->getMessage();
    dispFooter();
    exit;
}

```



```
}  
?>
```

- includes.php

```
<?php  
    session_start();  
    if(isset($_SESSION['user_type'])) $type=$_SESSION['user_type'];  
    if(isset($_SESSION['valid_user']))  
$val_user=$_SESSION['valid_user'];  
    if(isset($_SESSION['prevPage']))  
$prevPage=$_SESSION['prevPage']; else $prevPage=NULL;  
    if(isset($_SESSION['query'])) $query=$_SESSION['query']; else  
$prevPage=NULL;  
    require_once('html_functions.php');  
    require_once('main_functions.php');  
?>
```

- login.php

```
<?php  
include('includes.php');  
dispHeader();  
if(!isset($val_user))  
{  
    dispLoginBox();  
}  
else  
{  
echo "You are connected!";  
    dispURL("logout.php","logout");  
}  
dispFooter();  
?>
```

- logout.php

```
<?php  
  
include('includes.php');  
  
if(isset($_POST['logout'])) $logout=$_POST['logout']; else  
$logout=NULL;
```



```

if(isset($_POST['nlogout'])) $nlogout=$_POST['nlogout']; else
$nlogout=NULL;
$oldUser = $_SESSION['valid_user'];
if(isset($oldUser))
{
    if(isset($logout))
    {
        unset($_SESSION['valid_user']);
        session_destroy();
        sleep(2);
        header( 'Location: main.php' );
    }
    else if(isset($nlogout))
    {
        sleep(1);
        header( 'Location: main.php' );
    }
    else
    {
        dispHeader("");
        dispLogout();
        dispFooter();
    }
}
else
{
    dispHeader ('Logout');
    echo '<br /> <br /> You were not connected. <br />';
    echo "<br /> <br /> <a href=\"main.php\"> Back to home page </ a>";
    dispLogout();
}
?>

```

- main.php

```

<?php
include('includes.php');

try
{
    check_valid_user(1);
    dispHeader();

```



```

        dispMainPage();
        dispFooter();
    }
    catch(Exception $e)
    {
        dispHeader();
        echo $e->getMessage();
        dispFooter();
        exit;
    }
?>

```

- member.php

```

<?php
include('includes.php');
if(isset($_POST['username'])) $username = $_POST['username']; else
$username=NULL;
if(isset( $_POST['password'])) $passwd = $_POST['password']; else
$passwd=NULL;
if ($username && $passwd)
{
    try
    {
        $type_=login($username, $passwd);
        $_SESSION['valid_user'] = $username;
        $val_user=$_SESSION['valid_user'];
        if ($type_=="A") $type_="Admin";
        else $type_="User";
        $_SESSION['user_type'] = $type_;
        $type=$_SESSION['user_type'];
        header('Location: main.php');
    }
    catch(Exception $e)
    {
        dispHeader("");
        echo "<div class='header-bar-full'><h1 class='blue'>Home</h1></div>
            <div class='content-box-1'>
<div class='content-box-1-top'></div>
<div class='content-box-1-middle'>
<div class='content-box-1-content'> <div align='left'>
            <h3> Unable to load the page. </h3>

```



```

        <h4 class='blue-tip-text'> The username or password you entered
was incorrect. </h4> <br/> ";
echo $e->getMessage();
    dispURL('login.php', 'try again');
echo "<br /> <br /> you're not a member?; Register Now?";
dispURL("signup.php", "Registration");
    echo "</div></div></div>
<div class='content-box-1-bottom'>&nbsp;</div></div>";
    dispFooter();
    exit;
}
}
check_valid_user();
dispHeader("");
if (isset($_SESSION['user_type']))
{
    echo"<div class='header-bar-full'><h1 class='blue'>Home page
settings</h1></div>";
    if($type=="Admin")
    {
        echo "<div id='sub-header-left'>Welcome<span
class='yellow'>$val_user</span>!/</div>
        <div class='content-box-1'>
        <div class='content-box-1-top'></div>
        <div class='content-box-1-middle'>
        <div class='content-box-1-content'> <div align='left'>
            <h4> You are logged in as admin </ h4> ";
        dispURL ("editUser.php? user =". $val_user, "Edit my profile");
        echo "<br />";
        dispURL ("displayUsers.php", "Show user");
        echo "<br />";
        dispURL ("editCategories.php", "View/Edit Category");
        echo "<br />";
        dispURL ("editFacilities.php", "View/Edit benefit");
        echo "<br />";
        dispURL ("showNewAdvs.php", "View/Edit Newest ads");
        echo "<br />";
        dispURL ("favAdvs.php", "View of my favorite ads");
        echo "<br />";
        dispURL ("myProperties.php", "View my ads");
        echo "<br />";
        echo "<br />";
        echo "
    }
}

```



```

        <form method=\"post\" action=\"member.php\" >
        <input type=\"submit\" name=\"testData\"
value=\"Display options for the test data!\" \>
        </form>
        ";
        if(isset($_POST['testData']))
        {
            echo "
                <fieldset>
                <legend>entry of test data</legend>

                <form method=\"post\" action=\"member.php\"
>
                    1.<input type=\"submit\"
name=\"createSProperties\" value=\"Create Random 1000 Property
(Sales)!\" \>

                    </form>
                    ";
            echo "
                <form method=\"post\" action=\"member.php\"
>
                    2.<input type=\"submit\"
name=\"createLProperties\" value=\"Create Random 500 properties
(rental)!\" \>

                    </form>
                    </fieldset>
                    ";
        }
        if(isset($_POST['createSProperties'])) createSProperties();
        if(isset($_POST['createLProperties'])) createLProperties();

        echo "
            <form method=\"post\" action=\"member.php\" >
            <fieldset>
            <legend>Send email</legend>
            <input type=\"text\" name=\"email\" value=\"email\"
/>

            <input type=\"text\" name=\"subjEmail\"
value=\"sub\" /><br />
            <textarea name=\"textEmail\" rows=\"2\"
cols=\"41\">text</textarea>
            <input type=\"submit\" name=\"emailTest\"
value=\"Send email!\" \>

```



```

        </fieldset>
    </form>
    ";
    if(isset($_POST['emailTest']))
    {
        $headers = 'From: uttam@gmail.com' . "\r\n" .
        'Reply-To: uttam@hghg.com' . "\r\n" .
        'X-Mailer: PHP/' . phpversion();
        $ret=mail_utf8($_POST['email'],
        $_POST['subjEmail'], $_POST['textEmail'], $headers);
        if($ret==true) echo "The mail was sent!<br />";
        else echo "The mail is not sent!<br />";
    }
    }
    else
    {
        echo "<div id='sub-header-left'>welcome <span
        class='yellow'>$val_user</span>!</div>
        <div class='content-box-1'>
        <div class='content-box-1-top'></div>
        <div class='content-box-1-middle'>
        <div class='content-box-1-content'> <div align='left'>";
        dispURL("editUser.php?user=".$val_user,"Edit My
        Profile");
        echo "<br />";
        dispURL("favAdvs.php","View my favorite ads");
        echo "<br />";
        dispURL("myProperties.php","View my own ads");
        echo "<br />";
    }
    dispURL("logout.php","logout");
    echo "</div></div></div>
    <div class='content-box-1-bottom'>&nbsp;  </div></div>";
    }
    dispFooter();
    ?>

```

- myProperties.php

```

<?php
include('includes.php');
if(isset($_POST['remProperty'])) $remProp=$_POST['remProperty']; else
$remProp=NULL;

```



```

if(isset($_POST['delProperty'])) $delProp=$_POST['delProperty']; else
$delProp=NULL;
if(isset($_GET['page'])) $page=$_GET['page']; else $page=1;
if(isset($_GET['results'])) $results=$_GET['results']; else $results=15;

try
{
    if(isset($delProp))
    {
        foreach($delProp as $prop)
        {
            db_del_prop($prop);
        }
    }
    check_valid_user();
    $messageT="select distinct
property.prop_id,address,price,offer_type,area,views,category from
property,categories,cat_prop,fac_prop,facilities";
    $messageT.= " where property.user_id='$val_user' and
property.propState='T'";
    $messageT.=" and property.prop_id=cat_prop.prop_id and
categories.cat_id=cat_prop.cat_id";
    $messageT.="";
    $messageF="select distinct
property.prop_id,address,price,offer_type,area,views,category from
property,categories,cat_prop,fac_prop,facilities";
    $messageF.= " where property.user_id='$val_user' and
property.propState='F'";
    $messageF.=" and property.prop_id=cat_prop.prop_id and
categories.cat_id=cat_prop.cat_id";
    $messageF.="";
    $resultT=db_execute($messageT,"myProperty1");
    $resultF=db_execute($messageF,"myProperty2");
    check_valid_user();
    dispHeader("");
    echo "<div class='header-bar-full'><h1 class='blue'>My own
properties </h1></div>";
    if(!(mysql_num_rows($resultT)>0) &&
!(mysql_num_rows($resultF)>0))
    {
        echo "<br /> You have no property!<br />";
    }
    else

```



```

        {
            echo "<div id='sub-header'><span
class='yellow'>approved</span>ads</div>";
            if(!(mysql_num_rows($resultT)>0)) echo "<br /> You have
no approved advert!<br />";
            else
propertySearch($messageT,"UserDelete",$page,$results);
            echo "<div id='sub-header'><span class='yellow'> In
Waiting for Approval </span>ads</div>";
            if(!(mysql_num_rows($resultF)>0)) echo "<br /> You have
no ad to be approved!<br />";
            else
propertySearch($messageF,"UserDelete",$page,$results);
        }
        dispFooter();
    }
    catch(Exception $e)
    {
        dispHeader("Error:");
        echo $e->getMessage();
        dispFooter();
        exit;
    }
?>

```

- profUser.php

```

<?php

include('includes.php');
try
{
    check_valid_user();
    $user=strval($_GET['user']);
    dispHeader("");
    showUserProfile($user);
    dispFooter();
}
catch (Exception $e)
{
    dispHeader('error:');
    echo $e->getMessage();
    dispFooter();
}

```



```

        exit;
    }
    ?>

- showNewAdvs.php

<?php
include('includes.php');
if(isset($_POST['accProperty'])) $accProp=$_POST['accProperty']; else
$accProp=NULL;
if(isset($_GET['page'])) $page=$_GET['page']; else $page=1;

try
{
    check_valid_user();
    if(isset($accProp))
    {
        foreach($accProp as $prop)
        {
            echo "$prop <br />";
            db_update('property','prop_id','propState',$prop,'"T"');
        }
    }
    dispHeader("");
    echo "<div class='header-bar-full'><h1
class='blue'>Ads</h1></div>";
    $message="select distinct
property.prop_id,address,price,offer_type,area,views,category from
property,categories,cat_prop";
    $message.=" where property.propState='F' and";
    $message.=" property.prop_id=cat_prop.prop_id and
categories.cat_id=cat_prop.cat_id";
    $message=$message.";";
    $result=db_excecute($message,"fevAds");
    if(mysql_num_rows($result)==0) echo "<h3>There are no new
property! </h3>
<a href=member.php> Back to Profile </ a> <br/>
<a href=main.php> Back to Home </a> ";
    else propertySearch($message,"Accept",$page);
    dispFooter();
}
catch(Exception $e)
{

```



```

dispHeader("Error:");
echo $e->getMessage();
dispFooter();
exit;
}

```

```
?>
```

- signup.php

```

<?php
include('includes.php');
dispHeader("");
dispRegForm();
dispFooter();

```

```
?>
```

- signup_new.php

```

<?php
include('includes.php');
if(isset($_POST['username'])) $username=$_POST['username']; else
$username=NULL;
if(isset($_POST['pwd'])) $passwd=$_POST['pwd']; else
$passwd=NULL;
if(isset($_POST['pwd2'])) $passwd2=$_POST['pwd2']; else
$passwd2=NULL;
if(isset($_POST['mail'])) $email=$_POST['mail']; else $email=NULL;
if(isset($_POST['homephone'])) $home=$_POST['homephone']; else
$home=NULL;
if(isset($_POST['mob1phone'])) $mob1=$_POST['mob1phone']; else
$mob1=NULL;
if(isset($_POST['mob2phone'])) $mob2=$_POST['mob2phone']; else
$mob2=NULL;
if(isset($_POST['othrnumber'])) $othr=$_POST['othrnumber']; else
$othr=NULL;
if(isset($_POST['frstname'])) $frst=$_POST['frstname']; else
$frst=NULL;
if(isset($_POST['lstname'])) $lst=$_POST['lstname']; else $lst=NULL;

try
{

```



```

        if (!filledOut($username) || !filledOut($passwd) ||
!filledOut($passwd2) || !filledOut($email) || !filledOut($mob1))
        {
            throw new Exception('You have not filled out the form
correctly,try again');
        }

        if (!valid_email($email))
        {
            throw new Exception('This mail you entered is not valid.');
```

```

        }
        if ($passwd != $passwd2)
        {
            throw new Exception('password not matched.');
```

```

        }
        if (strlen($passwd)<4 || strlen($passwd) >16)
        {
            throw new Exception('The length of your password must be
between 4 and 16.');
```

```

        }
        if (strlen($mob1)!=10 || !(is_numeric($mob1)))
        {
            throw new Exception('cell should contain only numbers and
has exactly 10 digits.');
```

```

        }
        if (strlen($mob2)>0 && (strlen($mob2)!=10 ||
!(is_numeric($mob2))))
        {
            throw new Exception('mobile no should only contain
numbers and have exactly 10 digits.');
```

```

        }
        if (strlen($othr)>0 & (strlen($othr)!=10 || !(is_numeric($othr))))
        {
            throw new Exception('other phone should only contain
numbers and have exactly 10 digits.');
```

```

        }
        if (strlen($home)>0 & (strlen($home)!=10 ||
!(is_numeric($home))))
        {
            throw new Exception('The landline should contain only
numbers and have exactly 10 digits.');
```

```

        }
        if (strlen($home)>0 & (strlen($home)!=10 ||
!(is_numeric($home))))
        {
            throw new Exception('The landline should contain only
numbers and have exactly 10 digits.');
```

```

        }
        if (strlen($home)>0 & (strlen($home)!=10 ||
!(is_numeric($home))))
        {
            throw new Exception('The landline should contain only
numbers and have exactly 10 digits.');
```

```

        }
        if (strlen($home)>0 & (strlen($home)!=10 ||
!(is_numeric($home))))
        {
            throw new Exception('The landline should contain only
numbers and have exactly 10 digits.');
```

```

        }
        if (strlen($home)>0 & (strlen($home)!=10 ||
!(is_numeric($home))))
        {
            throw new Exception('The landline should contain only
numbers and have exactly 10 digits.');
```



```

    }
    register($username, $passwd, $email, $mob1);

    if($first)
        db_update("users","username","name","$username","$first");
    if($lst)
        db_update("users","username","surname","$username","$lst");
    if($home)
        db_update("telephone","user_id","home","$username","$home");
    if($mob2)
        db_update("telephone","user_id","mobile2","$username","$mob2
    ");
    if($other)
        db_update("telephone","user_id","other","$username","$other");
    $message = "Welcome".$username." at Real Estate Project.\n";

    $message = wordwrap($message, 70);
    $to = $email;
    $subject = 'Real Estate Project';
    $headers = 'From: uttamdhabas@gmail.com' . "\r\n" .
'Reply-To: uttamdhabas@gmail.com' . "\r\n" .
'X-Mailer: PHP/' . phpversion();
    $ret=mail_utf8($email, $subject, $message,$headers);

    dispHeader("");
    echo "<div class='header-bar-full'><h1 class='blue'>successful
Registration</h1></div>
    <div class='content-box-1'>
    <div class='content-box-1-top'></div>
    <div class='content-box-1-middle'>
    <div class='content-box-1-content'>
    <h3> Registration was successful! </ h3> Welcome to Real Estate
Project! <br/>
    You can now log in via the login page. <br/> <br/> ";
    dispURL('login.php', 'Login page');
    echo "</div></div>
    <div class='content-box-1-bottom'>&nbsp;</div></div>";
    dispFooter();
}
catch (Exception $e)
{
    dispHeader('Error:');

```



```

        echo $e->getMessage();
        dispFooter();
        exit;
    }

```

```
?>
```

- viewProperty.php

```
<?php
```

```

require_once("includes.php");
if(isset($_GET['propId'])) $id=strval($_GET['propId']); else $id=NULL;
if(isset($_POST['add'])) $add=$_POST['add']; else $add=NULL;
try
{
    $message="";
    if(isset($add)&&isset($val_user))
    {
        if($add=="add!")
        {
            db_insert('fav_prop','user_id','prop_id',"{$val_user}",$id);
            $message="<br />advertisement added to your
Favourite Ads! <br />";
        }
        else
        {
            db_delete('fav_prop','user_id','prop_id',"{$val_user}",$id);
            $message="<br />advertisement erased from your
favorite listings! <br />";
        }
    }
    else if(isset($add)&&!isset($val_user))
    {
        check_valid_user();
    }
    check_valid_user(1);
    dispHeader("");
    db_update('property','prop_id','views',$id,'views+1');

    showProperty($id);
}

```



```
        echo $message;
        dispFooter();
    }
    catch(Exception $e)
    {
        dispHeader('Error');
        echo $e->getMessage();
        dispFooter();
        exit;
    }
?>
```


4.4 Snap shots of user interface

■ home page

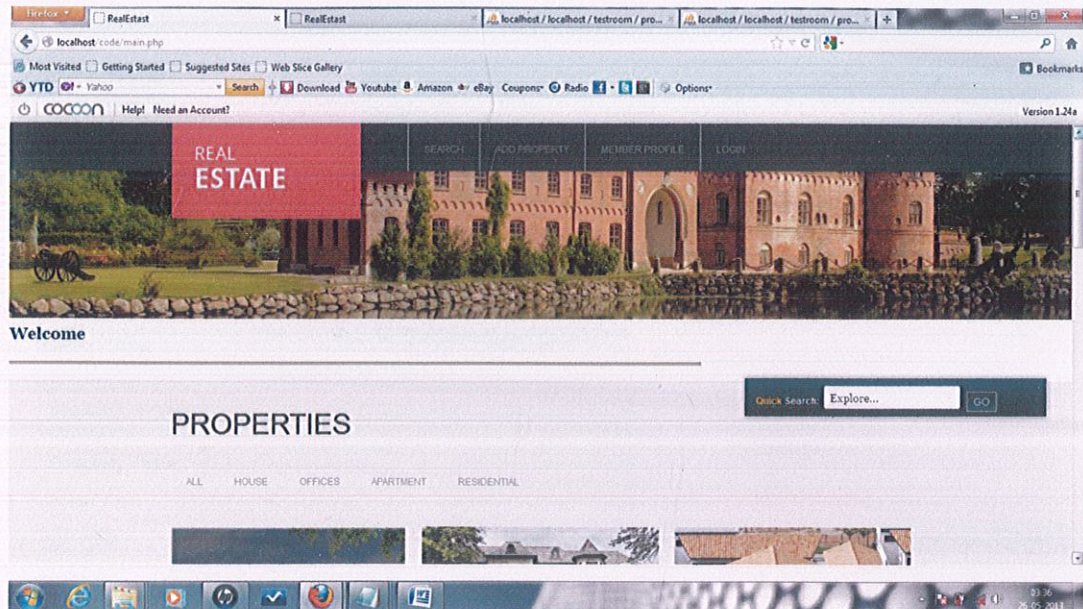


Fig-18

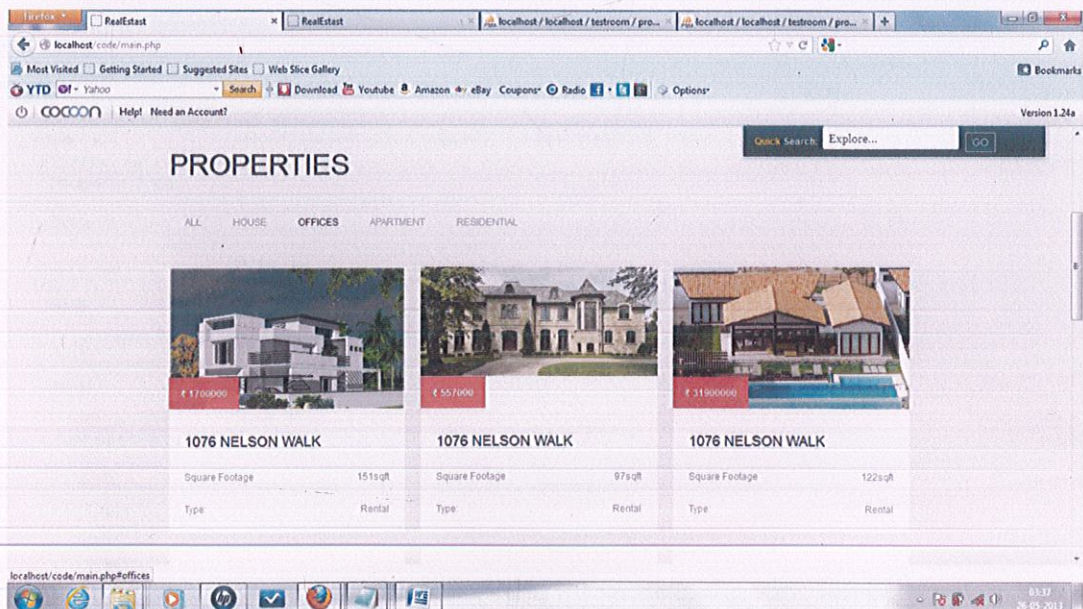


Fig-19

■ Search page

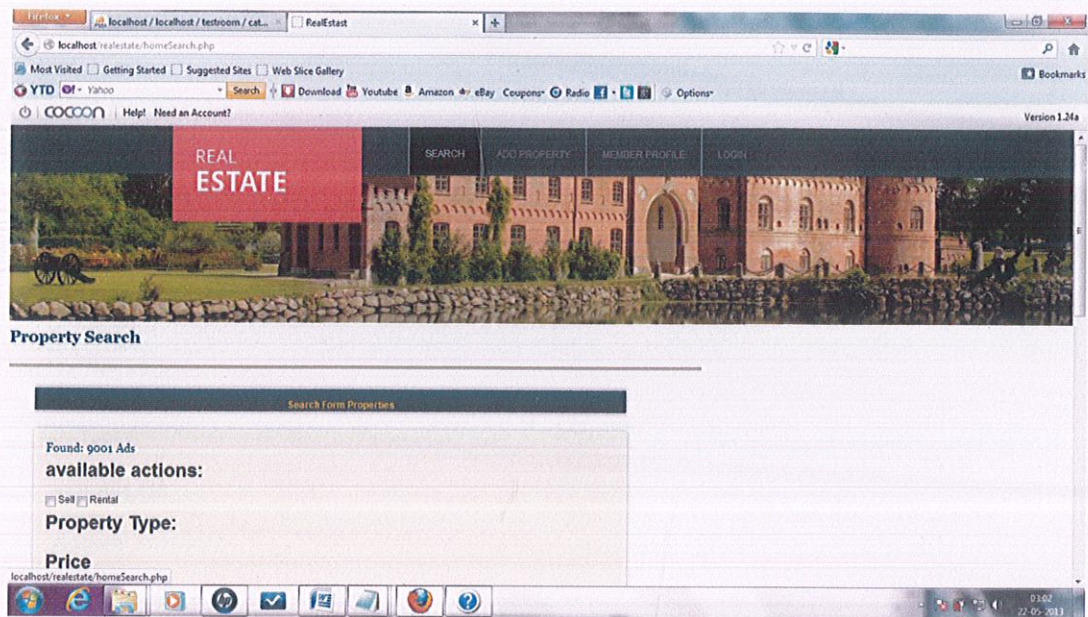


Fig-20

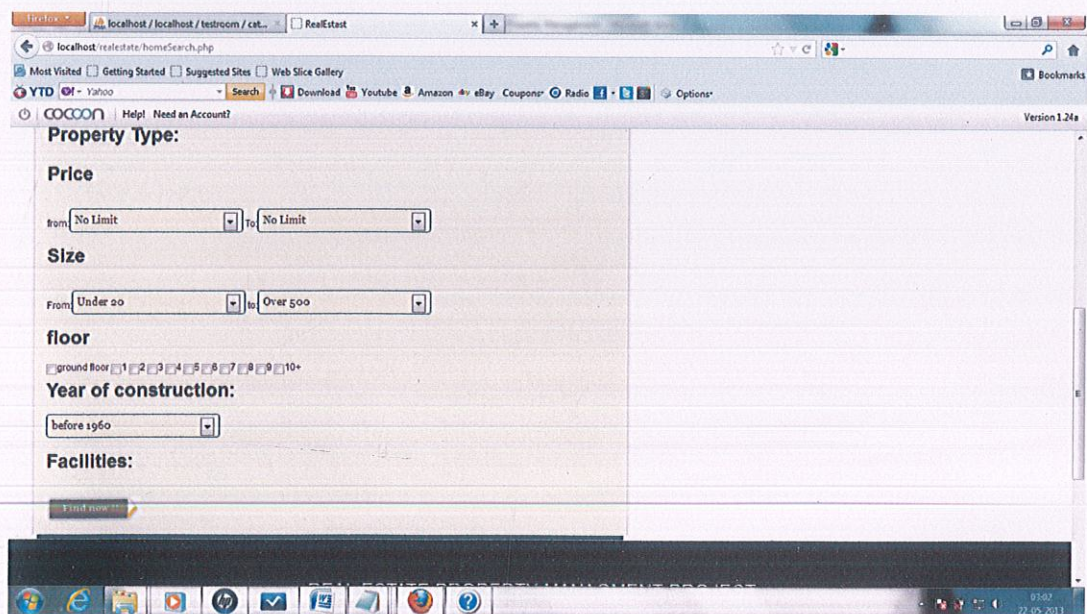


Fig-21

■ Add property page

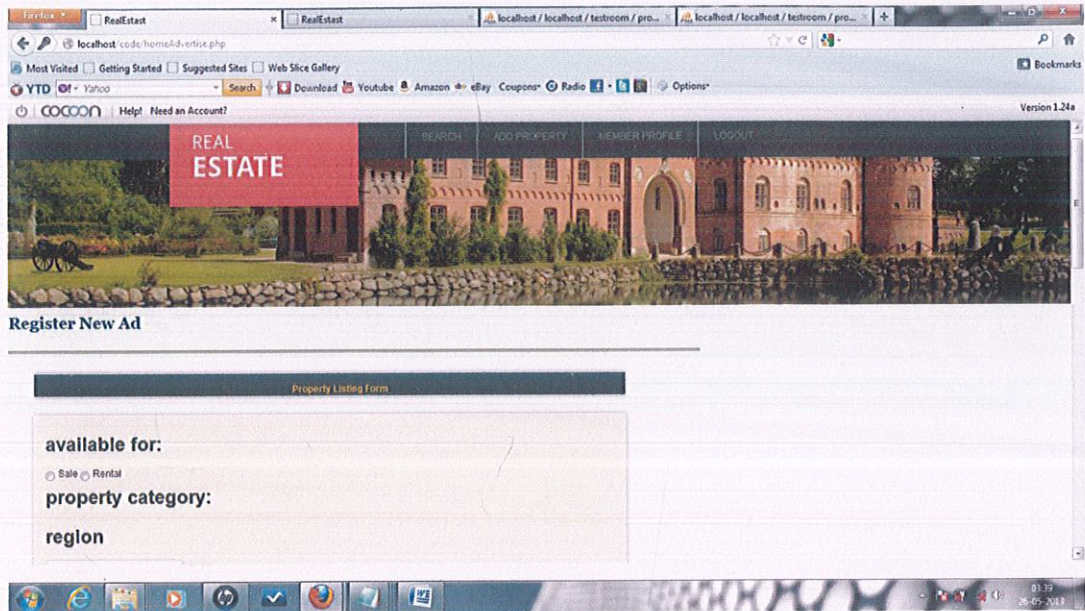


Fig-22

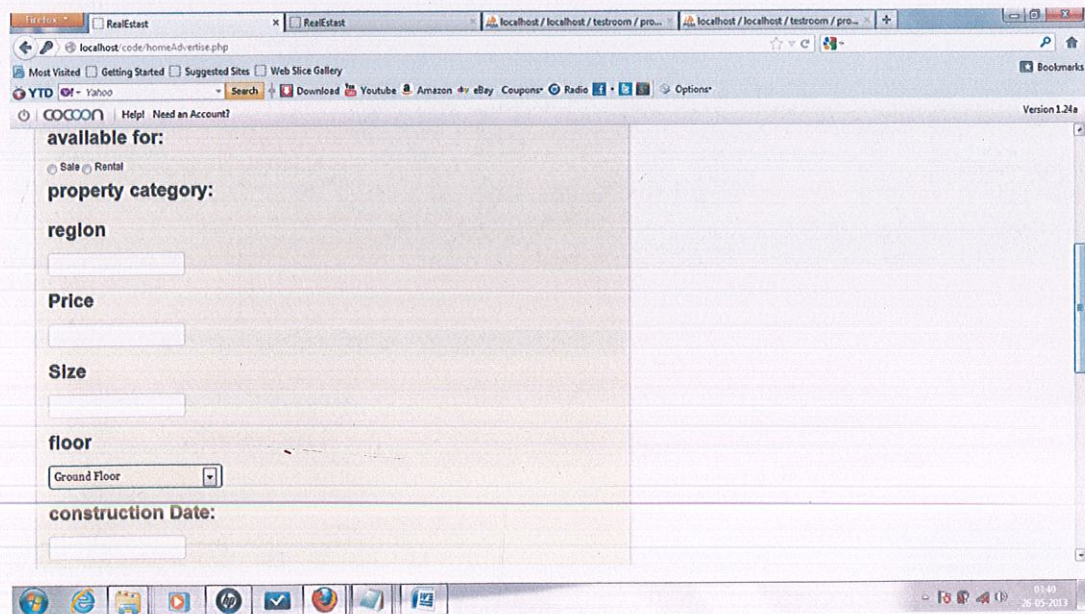


fig-23

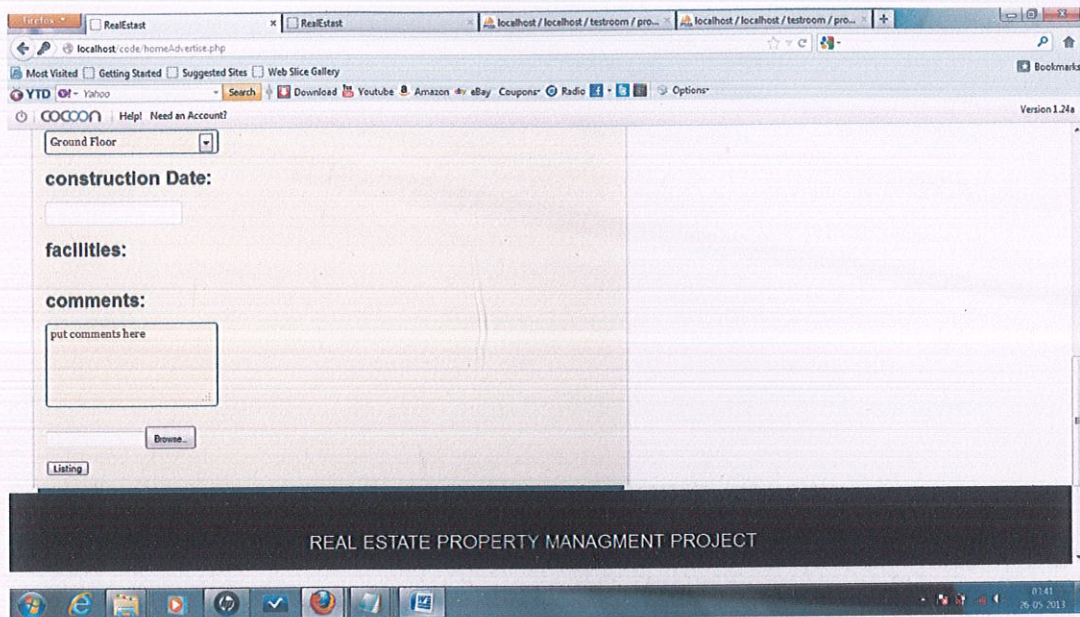


Fig-24

■ Member profile of administrator

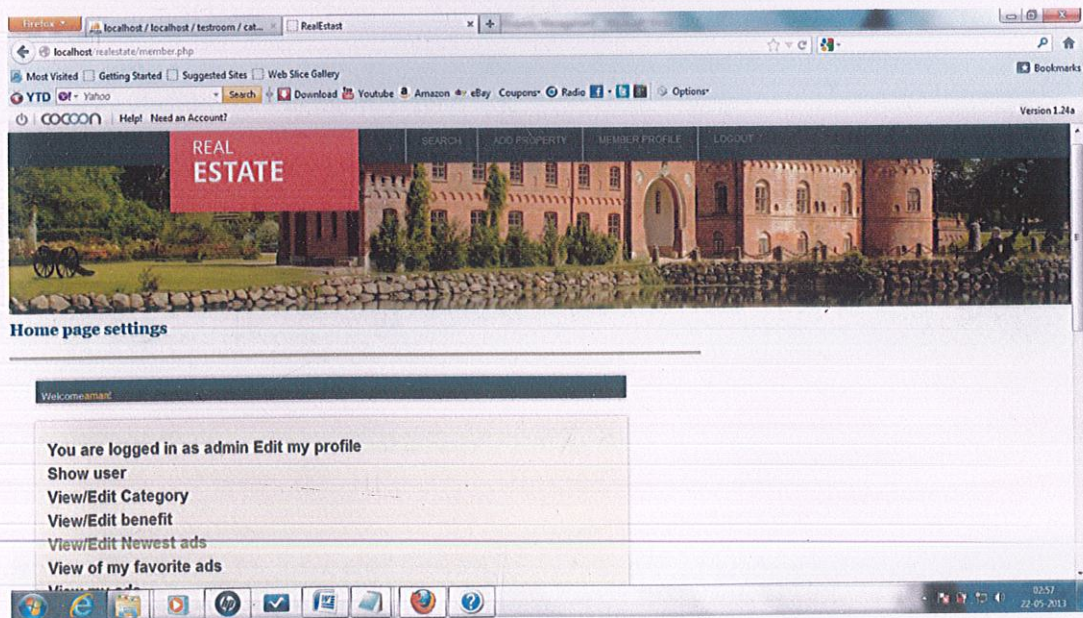


Fig-25

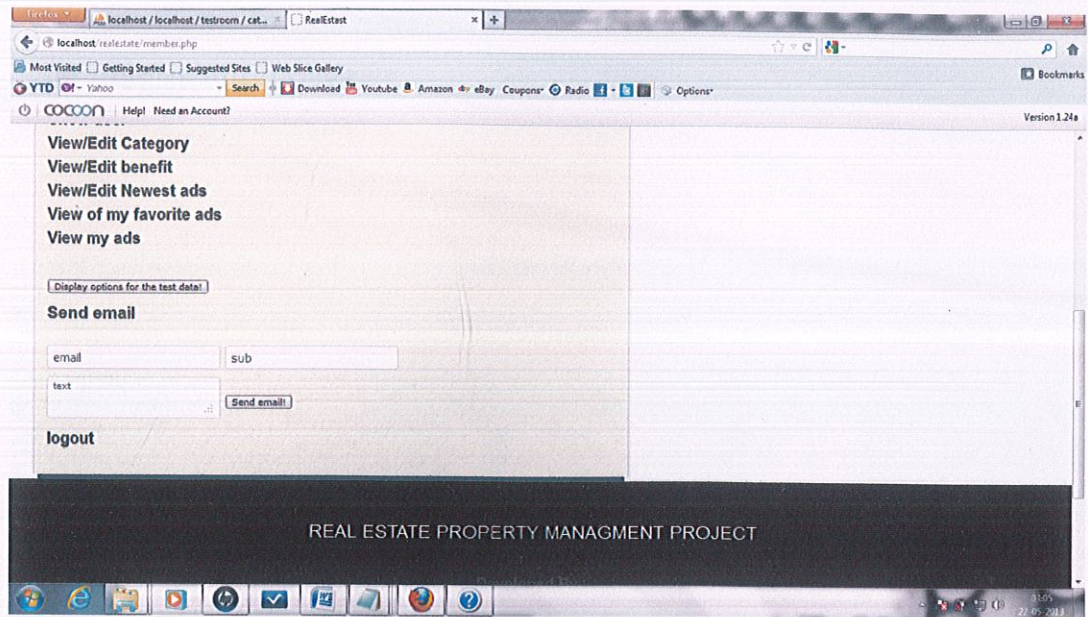


Fig-26

■ Member profile of user

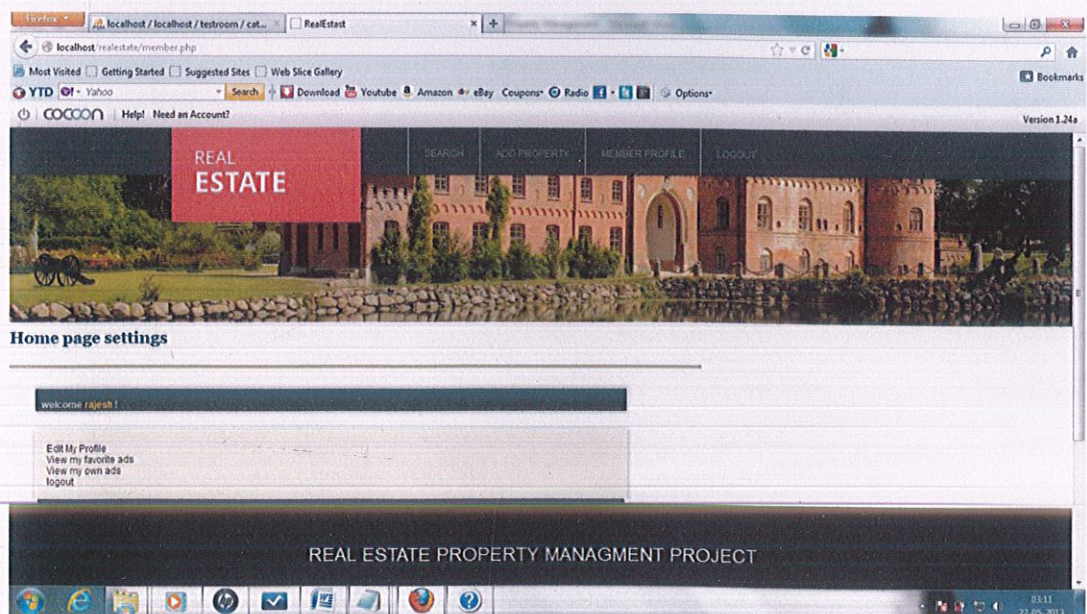


Fig-27

■ Registration page

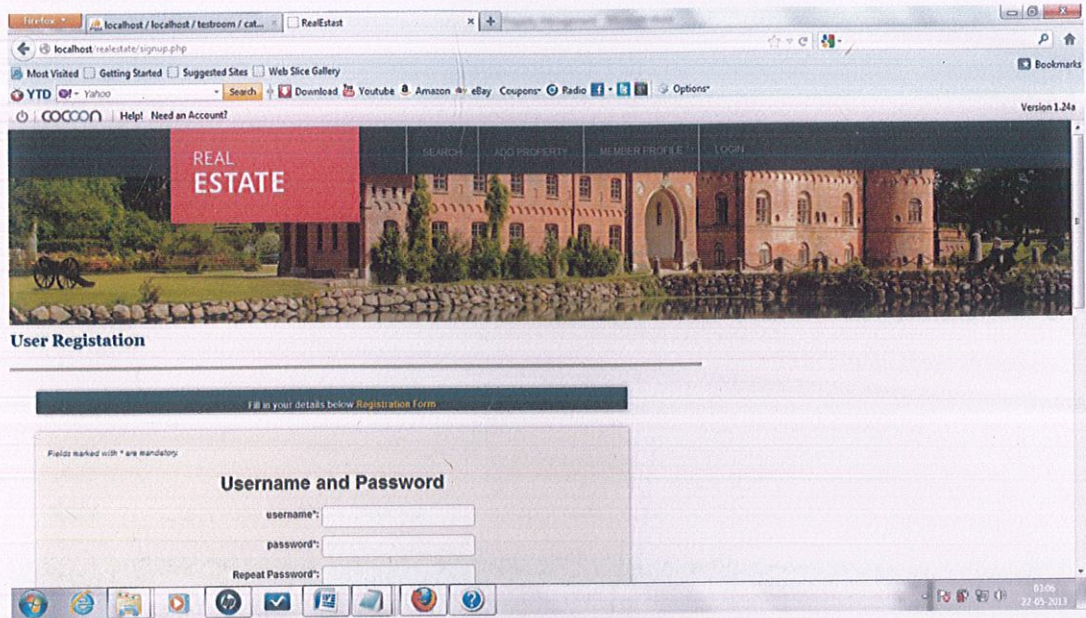


Fig-28

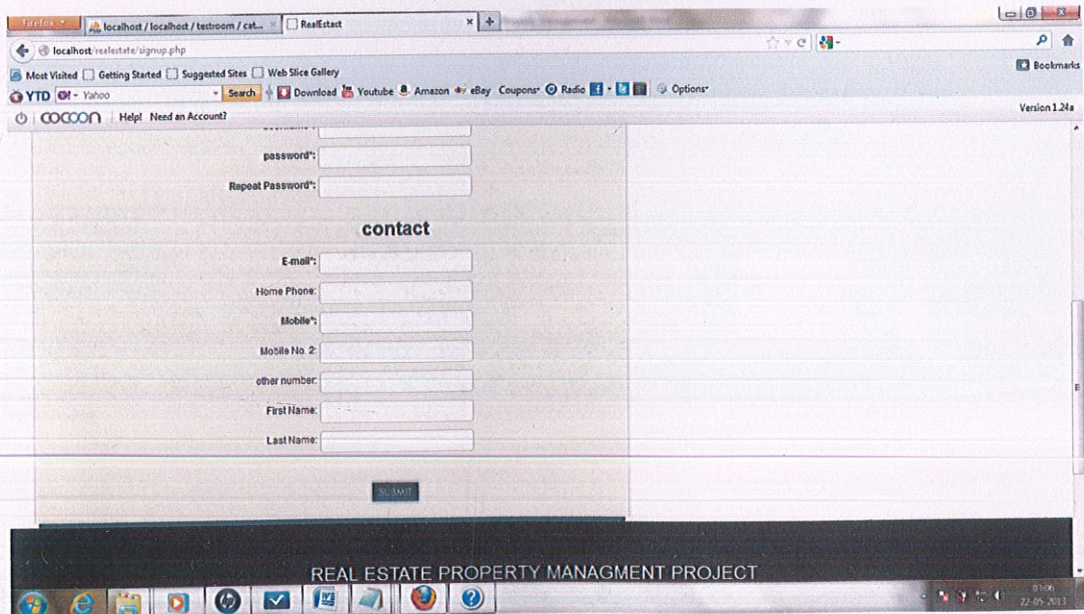


Fig-29

■ Login page

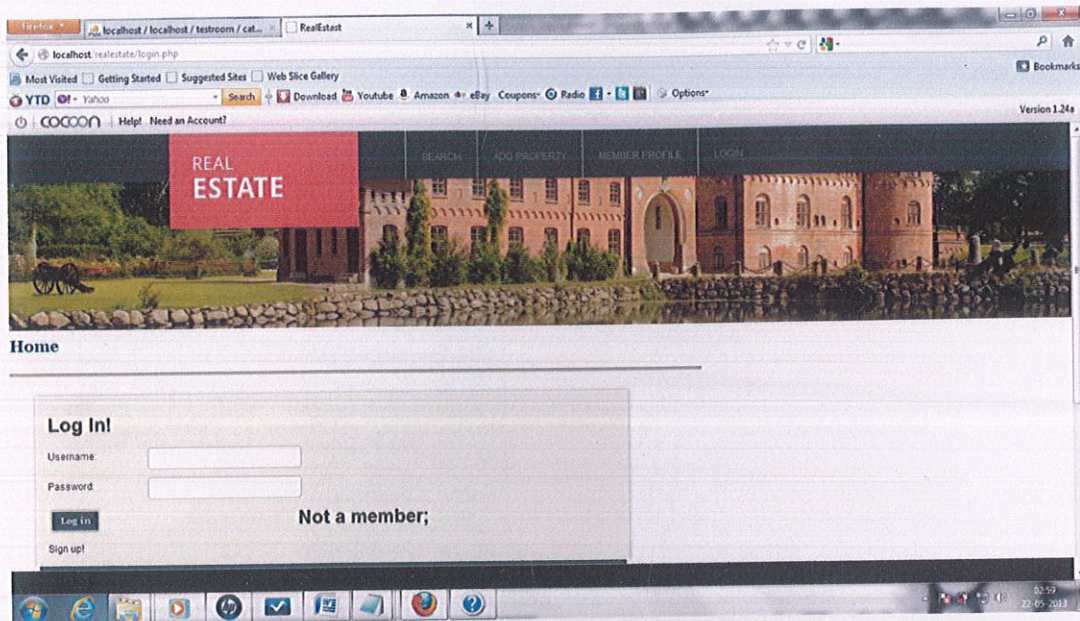


Fig-30

■ Logout page

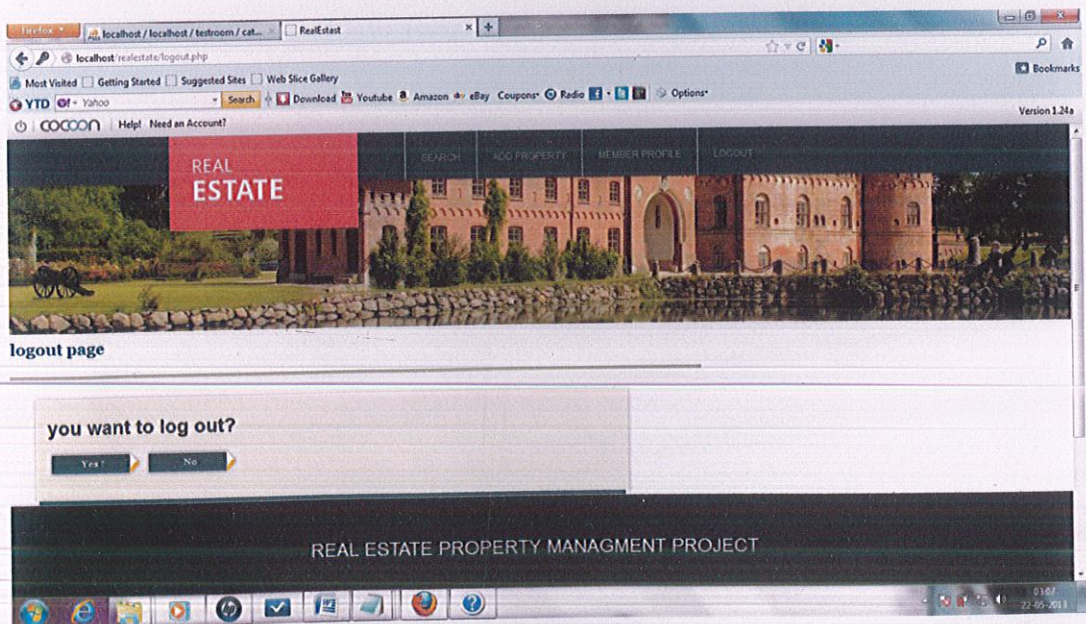


Fig-31

■ User Details Page

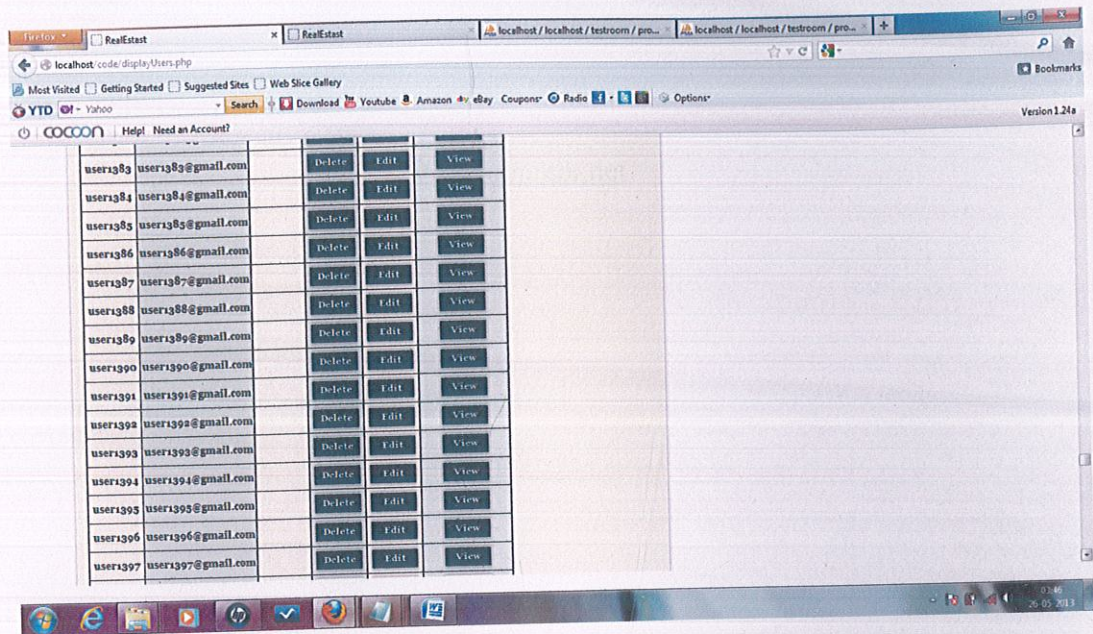


Fig-32

BIBLIOGRAPHY

Websites

- <http://www.google.com>
- <http://www.microsoft.com>
- <http://www.programmer2programmer.net>
- <http://www.codeproject.com>
- <http://www.msdn.com>
- <http://www.sqltuner.com>
- <http://www.wikipedia.com>

Books

- The complete java reference
- Computer Networks 4th Ed - Andrew S. Tanenbaum
- McGraw Hill - Software Engineering - A Practitioner's Approach - Pressman (5th Ed)(2001)
- (Wiley) The Art Of Software Testing 2nd Edition
- data And Databases Concepts In Practice - Joe Celkos
- John Wiley and Sons HTML Complete Course eBook-LiB

BIBLIOGRAPHY

Websites

- <http://www.google.com>
- <http://www.microsoft.com>
- <http://www.programmer2programmer.net>
- <http://www.codeproject.com>
- <http://www.msdn.com>
- <http://www.sqltuner.com>
- <http://www.wikipedia.com>

Books

- The complete java reference
- Computer Networks 4th Ed - Andrew S. Tanenbaum
- McGraw Hill - Software Engineering - A Practitioner's Approach - Pressman (5th Ed)(2001)
- (Wiley) The Art Of Software Testing 2nd Edition
- data And Databases Concepts In Practice - Joe Celkos
- John Wiley and Sons HTML Complete Course eBook-LiB