**HOUSE RENTAL MANAGEMENT SYSTEM**

**A Project Report submitted to Srimad Andavan Arts & Science College (Autonomous),**

**Affiliated to BHARATHIDASAN UNIVERSITY in partial** **fulfillment of the requirement for the award of the Degree of**

# BACHELOR OF SCIENCE IN

***COMPUTER SCIENCE***

*Submitted By*

**N.DHEENA**

Register No:U20CS0019

**M.V.DHAYANETHI**

Register No:U20CS0018

Under the Guidance of

**DR.M.SHBHASHINI.,MCA,M.Phil,Ph.D**

**DEPARTMENT OF COMPUTER SCIENCE**



**Srimad Andavan Arts & Science College (Autonomous)**

(Affiliated to Bharathidasan University) Nationally Re -accredited by NAAC An ISO 9001:2015 Certified Institution No 7, Nelson Road, Tiruvanaikoil, Tiruchirappalli - 620 005.

## **APRIL 2023**



Srimad Andavan Arts & Science College ( Autonomous)

**(Affiliated to Bharathidasan University)**

**Nationally Re –accredited by NAAC**

**An ISO 9001:2015 Certified Institution No 7, Nelson**

**Road, Tiruvanaikoil, Tiruchirapalli - 620 005.**

**DR.M.SHBHASHINI.,MCA,M.Phil,Ph.D**

Assistant Professor in Computer Science

**CERTIFICATE**

This is to certify that the Project Report appellation entitled **“HOUSE RENTAL MANAGEMENT SYSTEM”** submitted to Srimad Andavan Arts & Science College (Autonomous), Affiliated to Bharathidasan University, Tiruchirappalli in partial fulfillment of the requirement for the award of the degree of Bachelor of Science in Computer Science for the academic Year 2022-2023, is a bonafide record of the project work done by **DHEENA.N** under my guidance and supervision. The dissertation has not previously formed the basis for the award of any degree or fellowship to the candidate.

Comments:

SIGNATURE OF THE GUIDE SIGNATURE OF THE HOD

SIGNATURE OF THE EXAMINER

Date:

**DECLARATION**

I declare that the project work entitled **“HOUSE RENTAL MANAGEMENT SYSTEM”** submitted to, Srimad Andavan Arts and Science College (Autonomous), affiliated to Bharathidasan University, Tiruchirappalli in partial fulfillment of the requirement for the award of the degree of Bachelor of Science in Computer Science for the academic year 2022-2023, is the result of the original work carried out by me under the guidance and supervision of **DR.M.SHBHASHINI.,MCA,M.PHIL,PH.D**, Department of Computer Science, Srimad Andavan Arts and Science College (Autonomous), Thiruchirappalli.

Date :

Place :

**Signature of the Candidate**

N.DHEENA

**VIVA VOCE**

The Viva-Voce Examination of this Project Work is held on \_\_\_\_\_\_\_\_\_at Srimad Andavan Arts and Science College (Autonomous), Tiruchirappalli-5

**Internal Examiner External Examiner**

**ACKNOWLEDGEMENT**

First, I am very thankful to the Almighty for showering the Blessings throughout my life. I would like to express my heartfelt thanks to my beloved parents who have sacrificed a lot for my future. This support was the foundation stone for my efforts.

I acknowledge my sincere thanks to **Shri Ammangi V.BALAJI**, Secretary & Correspondent, **Dr.M.PITCHAIMANI.,** Principal, **,Dr.S.R.SATHYANARAYANAN,**Vice Principal,**Dr.S.MADHU,** Dean of Sciences, **Dr.V.UPENDRAN.,** Head-PG and Research Department of Computer Science, **Dr.N.KARTHIKEYAN,** Co-ordinator, PG and Research Department of Computer Science, for giving me the opportunity to do my Under Graduate Degree course in this renowned institution and providing facilities to carry out the project work.

I am grateful to **DR.M.SHBHASHINI.,MCA,M.Phil,Ph.D,** PG and Research Department of Computer Science, the internal guide for timely suggestions and constant encouragement and support that led to the accomplishment of the project.

I extend my thanks to all the teaching, non-teaching staff members and Computer Center Staffs who helped me either directly or indirectly to complete this project.

I wish to express my heartfelt thanks to my friends for their useful tips that helped me to complete this project successfully

**Abstract**

The main aim of the project is to provide utility to maintain day to day operations of house and apartments. This project helps them to store all the in formations about house and apartment. In this application used to maintain the house blocks details, user details and rent details. To manage all the transactions we proposed here with a software solution which will take care all the necessary transactions. On implementation of this software it will help them in many ways. In this project the guest can login the system using user name and password after registration process. The he/she add the house or apartment details like apartment name, id, location, price etc. The user can login the system using user name and password after registration process then he/she view the house or apartment details. If the user select or book an apartment or house send the request to the admin. The admin can share the user details to guest profile. In this project the user can contact or communicate the guest through the admin only.

1 **CONTENT**

|  |  |  |
| --- | --- | --- |
| **s.no** | **Title** | **Page no** |
| 1 | **Introduction** |  |
| 2 | **Requirement specification**  2.1 Hardware Requirement  2.2 Software Requirement  2.3 Existing System  2.4 Proposed System  2.5 Software Description  2.6 Project Description |  |
| 3 | **Design**  3.1 Data Flow Diagram  3.2 Use Case Diagram  3.3 Table Design |  |
| 4 | **Implementation**  4.1 Source Code  4.2 Screenshots |  |
| 5 | **Testing** |  |
| 6 | **Conclusion** |  |
| 7 | **Bibliography** |  |

.

1. **INTRODUCTION**

**1.1. OVERVIEW OF THE PROJECT**

Rental house management has become important factor in modern society hence the need to have a rental house management system. Housing has a central importance to quality of life with considerable economic, social, cultural and personal significance. Though a country’s national prosperity is usually measured in economic terms, increasing wealth is of diminished value unless all can share its benefits and if the growing wealth is not used to redress growing social deficiencies, one of which is housing. Housing plays a huge role in revitalizing economic growth in any country, with shelter being among key indicators of development. The universal declaration of human rights gives one of the basic human rights as the right to a decent standard of living, central to which is the access to adequate housing. Housing as a basic human right demands that urban dwellers should have access to a decent housing, defined as one that provides a foundation for rather than being a barrier to good physical and mental health, personal development and fulfillment of life objectives. The focus of this research paper is basically managing housing for low income, medium and high incomes households or what is commonly known as affordable housing. “Affordable” is a term used to describe individuals “capability to pay for certain products or services because their income is enough to do so. Although the term “affordable housing” is often applied to rental housing; that is within the financial means of those in the lower income ranges of a geographical area, the concept is applicable to both middle and high income individuals. Developing rental houses comes with many advantages especially to the Landlords who are able to increase their profits through rent paid by the tenants. Increased number of tenants and Landlords makes management difficult especially for the landlords who are losing huge sum of money through tenants who evade rent. The above statement gives a clear declaration as to why rental house management system need to be develope

**2. REQUIREMENT ANALYSIS**

**2.1. HARDWARE REQUIREMENTS**

* Processor : Intel processor 3.0 GHz
* RAM : 4 GB
* Hard disk : 1 TB
* Compact Disk : 650 MB
* Keyboard : Standard keyboard
* Mouse : Logitech mouse
* Monitor : 15 inch color monitor

**2.2. SOFTWARE REQUIREMENTS**

* Front End : PHP
* Back End : MYSQL
* Server : WAMP
* Operating System : Windows OS
* System type : 32-bit or 64-bit Operating System
* IDE : DREAMWEAVER
* DLL : Depends upon the title

**2.3. EXISTING SYSTEM**

City does n’ t have online home rental management system for who wants to rent home get a lot of trouble to find home even the home is available. There is no properly allocate home and the system is not easily arrange according to their user interest. And also the home rental management system almost is done through the manual system. The administrative system doesn’t have the facility to make home rental management system through online and the most time the work done through illegal intermediate person without awareness of the administrative and this make more complex and more cost to find home for the customer. This leads to customer in to more trouble, cost, dishonest and time wastage.

**2.3.1 DISADVANTAGES OF EXISTING SYSTEM**

* Manual work.
* Spent lot of time to search house.
* Manual search the house.
* It is not proper to maintain all records.
* Not convenient to user.

**2.4. PROPOSED SYSTEM:**

House rent management system in proposed system is used to search the room in particular place in room, office, paying guest, office also. It is user friendly web on the application. House management system is used to search the available location and available on the space. It is easy way to communicate house owner. The customer did not take no risk to search house.

**2.4.1. ADVANTAGES OF PROPOSED SYSTEM**

* The System Which Will allow the user to quickly and easily search house details.
* The register user can upload his property for sale or rent out.
* The System is design and developed in such way that it tries to overcome all the prescribe problem.
* The system being an online system will give accurate information regarding the property which helps to view all the stuff information directly from anywhere.

**2.5. SOFTWARE DESCRIPTION**

**2.5.1 FRONT-END**

**PHP**

PHP: Hypertext Preprocessor (or simply PHP) is a general-purpose programming language originally designed for web development. It was originally created by Rasmus Lerdorf in 1994 the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

PHP code may be executed with a command line interface (CLI), embedded into HTML code, or used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common Gateway Interface (CGI) executable. The web server outputs the results of the interpreted and executed PHP code, which may be any type of data, such as generated HTML code or binary image data. PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, with the original implementation acting as the de facto standard which other implementations aimed to follow. Since 2014, work has gone on to create a formal PHP specification.

**PHP Objects**

Basic [object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming) functionality was added in PHP 3 and improved in PHP 4. This allowed for PHP to gain further abstraction, making creative tasks easier for programmers using the language. Object handling was completely rewritten for PHP 5, expanding the feature set and enhancing performance. In previous versions of PHP, objects were handled like [value types](https://en.wikipedia.org/wiki/Value_type). The drawback of this method was that code had to make heavy use of PHP's "reference" variables if it wanted to modify an object it was passed rather than creating a copy of it. In the new approach, objects are referenced by [handle](https://en.wikipedia.org/wiki/Handle_(computing)), and not by value.

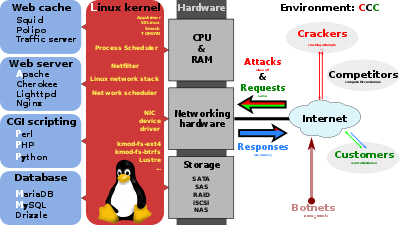
**Implementations**

The only complete PHP implementation is the original, known simply as PHP. It is the most widely used and is powered by the [Zend Engine](https://en.wikipedia.org/wiki/Zend_Engine). To disambiguate it from other implementations, it is sometimes unofficially called "Zend PHP". The Zend Engine [compiles](https://en.wikipedia.org/wiki/Compiler) PHP [source code](https://en.wikipedia.org/wiki/Source_code) on-the-fly into an internal format that it can execute, thus it works as an [interpreter](https://en.wikipedia.org/wiki/Interpreter_(computing)). It is also the "reference implementation" of PHP, as PHP has no formal specification, and so the semantics of Zend PHP define the semantics of PHP. Due to the complex and nuanced semantics of PHP, defined by how Zend works, it is difficult for competing implementations to offer complete compatibility.

**Licensing**

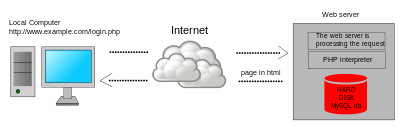
PHP is [free software](https://en.wikipedia.org/wiki/Free_software) released under the [PHP License](https://en.wikipedia.org/wiki/PHP_License), which stipulates that: Products derived from this software may not be called "PHP", nor may "PHP" appear in their name, without prior written permission from group@php.net. You may indicate that your software works in conjunction with PHP by saying "[Foo](https://en.wikipedia.org/wiki/Foo) for PHP" instead of calling it "PHP Foo" or "phpfoo". This restriction on use of "PHP" makes the PHP License incompatible with the [General Public License](https://en.wikipedia.org/wiki/General_Public_License) (GPL), while the Zend License is incompatible due to an advertising clause similar to that of the original [BSD license](https://en.wikipedia.org/wiki/BSD_license).

**Use**



A broad overview of the LAMP software bundle, displayed here together with [Squid](https://en.wikipedia.org/wiki/Squid_(software))

PHP is a general-purpose scripting language that is especially suited to [server-side](https://en.wikipedia.org/wiki/Server-side_scripting) [web development](https://en.wikipedia.org/wiki/Web_development), in which case PHP generally runs on a [web server](https://en.wikipedia.org/wiki/Web_server). Any PHP code in a requested file is [executed](https://en.wikipedia.org/wiki/Execution_(computing)) by the PHP runtime, usually to create [dynamic web page](https://en.wikipedia.org/wiki/Dynamic_web_page) content or dynamic images used on websites or elsewhere. It can also be used for [command-line](https://en.wikipedia.org/wiki/Command-line) scripting and [client-side](https://en.wikipedia.org/wiki/Client-side) [graphical user interface](https://en.wikipedia.org/wiki/Graphical_user_interface) (GUI) applications. PHP can be deployed on most web servers, many [operating systems](https://en.wikipedia.org/wiki/Operating_system) and [platforms](https://en.wikipedia.org/wiki/Computing_platform), and can be used with many [relational database management systems](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). Most [web hosting](https://en.wikipedia.org/wiki/Web_hosting) providers support PHP for use by their clients. It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.



Dynamic web page: example of [server-side scripting](https://en.wikipedia.org/wiki/Server-side_scripting) (PHP and MySQL)

PHP acts primarily as a [filter](https://en.wikipedia.org/wiki/Filter_(software)) taking input from a file or stream containing text and/or PHP instructions and outputting another stream of data. Most commonly the output will be HTML, although it could be [JSON](https://en.wikipedia.org/wiki/JSON), [XML](https://en.wikipedia.org/wiki/XML) or [binary data](https://en.wikipedia.org/wiki/Binary_data) such as image or audio formats. Since PHP 4, the PHP [parser](https://en.wikipedia.org/wiki/Parser) [compiles](https://en.wikipedia.org/wiki/Compiler) input to produce [bytecode](https://en.wikipedia.org/wiki/Bytecode) for processing by the [Zend Engine](https://en.wikipedia.org/wiki/Zend_Engine), giving improved performance over its [interpreter](https://en.wikipedia.org/wiki/Interpreter_(computing)) predecessor.

Originally designed to create dynamic [web pages](https://en.wikipedia.org/wiki/Web_page), PHP now focuses mainly on [server-side scripting](https://en.wikipedia.org/wiki/Server-side_scripting), and it is similar to other server-side scripting languages that provide dynamic content from a web server to a [client](https://en.wikipedia.org/wiki/Client_(computing)), such as [Microsoft](https://en.wikipedia.org/wiki/Microsoft)'s [ASP.NET](https://en.wikipedia.org/wiki/ASP.NET), [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems)' [Java Server Pages](https://en.wikipedia.org/wiki/JavaServer_Pages), and [mod\_perl](https://en.wikipedia.org/wiki/Mod_perl). PHP has also attracted the development of many [software frameworks](https://en.wikipedia.org/wiki/Software_framework) that provide building blocks and a design structure to promote [rapid application development](https://en.wikipedia.org/wiki/Rapid_application_development) (RAD). Some of these include [PRADO](https://en.wikipedia.org/wiki/PRADO_(framework)), [CakePHP](https://en.wikipedia.org/wiki/CakePHP), [Symfony](https://en.wikipedia.org/wiki/Symfony), [CodeIgniter](https://en.wikipedia.org/wiki/CodeIgniter), [Laravel](https://en.wikipedia.org/wiki/Laravel), [Yii Framework](https://en.wikipedia.org/wiki/Yii_Framework), [Phalcon](https://en.wikipedia.org/wiki/Phalcon_(framework)) and [Zend Framework](https://en.wikipedia.org/wiki/Zend_Framework), offering features similar to other [web frameworks](https://en.wikipedia.org/wiki/Web_framework).

The [LAMP architecture](https://en.wikipedia.org/wiki/LAMP_(software_bundle)) has become popular in the web industry as a way of deploying web applications. PHP is commonly used as the P in this bundle alongside [Linux](https://en.wikipedia.org/wiki/Linux), [Apache](https://en.wikipedia.org/wiki/Apache_HTTP_Server) and [MySQL](https://en.wikipedia.org/wiki/MySQL), although the P may also refer to [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), [Perl](https://en.wikipedia.org/wiki/Perl), or some mix of the three. Similar packages, [WAMP](https://en.wikipedia.org/wiki/WAMP_(software_bundle)) and [MAMP](https://en.wikipedia.org/wiki/MAMP), are also available for [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) and [macOS](https://en.wikipedia.org/wiki/MacOS), with the first letter standing for the respective operating system. Although both PHP and Apache are provided as part of the macOS base install, users of these packages seek a simpler installation mechanism that can be more easily kept up to date.

As of April 2007, over 20 million Internet domains had web services hosted on servers with PHP installed and mod\_php was recorded as the most popular [Apache HTTP Server](https://en.wikipedia.org/wiki/Apache_HTTP_Server) module. As of August 2019, PHP was used as the server-side programming language on 79.1% of websites, down from 83.5% previously, where the language could be determined. [Web content management systems](https://en.wikipedia.org/wiki/Web_content_management_system) written in PHP include [MediaWiki](https://en.wikipedia.org/wiki/MediaWiki), [Joomla](https://en.wikipedia.org/wiki/Joomla), [eZ Publish](https://en.wikipedia.org/wiki/EZ_Publish), [eZ Platform](https://en.wikipedia.org/wiki/EZ_Platform), [SilverStripe](https://en.wikipedia.org/wiki/SilverStripe), [WordPress](https://en.wikipedia.org/wiki/WordPress), [Drupal](https://en.wikipedia.org/wiki/Drupal), and [Moodle](https://en.wikipedia.org/wiki/Moodle). Websites written in PHP, in [back-end](https://en.wikipedia.org/wiki/Front_and_back_ends) and/or user-facing portion, include [Facebook](https://en.wikipedia.org/wiki/Facebook), [Digg](https://en.wikipedia.org/wiki/Digg), [Tumblr](https://en.wikipedia.org/wiki/Tumblr), [Dailymotion](https://en.wikipedia.org/wiki/Dailymotion), and [Slack](https://en.wikipedia.org/wiki/Slack_(software)).

For specific and more advanced usage scenarios, PHP offers a well-defined and documented way for writing custom extensions in [C](https://en.wikipedia.org/wiki/C_(programming_language)) or [C++](https://en.wikipedia.org/wiki/C%2B%2B). Besides extending the language itself in form of additional [libraries](https://en.wikipedia.org/wiki/Library_(computing)), extensions are providing a way for improving execution speed where it is critical and there is room for improvements by using a true [compiled language](https://en.wikipedia.org/wiki/Compiled_language). PHP also offers well defined ways for embedding itself into other software projects. That way PHP can be easily used as an internal [scripting language](https://en.wikipedia.org/wiki/Scripting_language) for another project, also providing tight interfacing with the project's specific internal [data structures](https://en.wikipedia.org/wiki/Data_structure). PHP received mixed reviews due to lacking support for [multithreading](https://en.wikipedia.org/wiki/Multithreading_(software)) at the core language level, though using threads is made possible by the "pthreads" [PECL](https://en.wikipedia.org/wiki/PHP_Extension_Community_Library) extension. As of January 2013, PHP was used in more than 240 million [websites](https://en.wikipedia.org/wiki/Website) (39% of those sampled) and was installed on 2.1 million [web servers](https://en.wikipedia.org/wiki/Web_server). A command line interface, php-cli, and two [ActiveX](https://en.wikipedia.org/wiki/ActiveX) [Windows Script Host](https://en.wikipedia.org/wiki/Windows_Script_Host) scripting engines for PHP have been produced. As of 2019, PHP 5 is most used on the web; which was last updated with security updates in January 2019, with PHP 5.6.40.

**Security**

In 2017, 3% of all vulnerabilities listed by the National Vulnerability Database were linked to PHP; historically, about 30% of all vulnerabilities listed since 1996 in this database are linked to PHP. Technical security flaws of the language itself or of its core libraries are not frequent (22 in 2009, about 1% of the total although PHP applies to about 20% of programs listed).Recognizing that programmers make mistakes, some languages include taint checking to automatically detect the lack of [input validation](https://en.wikipedia.org/wiki/Data_validation) which induces many issues. Such a feature is being developed for PHP, but its inclusion into a release has been rejected several times in the past. There are advanced protection patches such as [Suhosin](https://en.wikipedia.org/wiki/Suhosin) and [Hardening](https://en.wikipedia.org/wiki/Hardening_(computing))-Patch, especially designed for web hosting environments. Historically, old versions of PHP had some configuration parameters and default values for such runtime settings that made some PHP applications prone to security issues. Among these, [magic\_quotes\_gpc](https://en.wikipedia.org/wiki/Magic_quotes) and register\_globals configuration directives were the best known; the latter made any URL parameters become PHP variables, opening a path for serious security vulnerabilities by allowing an attacker to set the value of any uninitialized global variable and interfere with the execution of a PHP script. Support for "magic quotes" and "register globals" settings has been deprecated as of PHP 5.3.0, and removed as of PHP 5.4.0.

**WAMP**

WAMP is an acronym that stands for Windows, Apache, MySQL, and PHP. It’s a software stack which means installing WAMP installs Apache, MySQL, and PHP on your operating system (Windows in the case of WAMP). Even though you can install them separately, they are usually bundled up, and for a good reason too.

What’s good to know is that WAMP derives from LAMP (the L stands for Linux). The only difference between these two is that WAMP is used for Windows, while LAMP – for Linux based operating systems.

Let’s quickly go over what each letter represents:

“W” stands for Windows, there’s also LAMP (for Linux) and MAMP (for Mac).

“A” stands for Apache. Apache is the server software that is responsible for serving web pages. When you request a page to be seen by you, Apache grants your request over HTTP and shows you the site.

“M” stands for MySQL. MySQL’s job is to be the database management system for your server. It stores all of the relevant information like your site’s content, user profiles, etc.

“P” stands for PHP. It’s the programming language that was used to write WordPress. It acts like glue for this whole software stack. PHP is running in conjunction with Apache and communicating with MySQL.

Instead of installing and testing WordPress on your hosting account, you can do it on your personal computer (localhost).

WAMP acts like a virtual server on your computer. It allows you to test all WordPress features without any consequences since it’s localized on your machine and is not connected to the web.

First of all, this means that you don’t need to wait until files are uploaded to your site, and secondly – this makes creating backups much easier.

WAMP speeds up the work process for both developers and theme designers alike. What is more, you also get the benefit of playing around with your site to your heart’s content.

However, to actually make the website go live, you need to get some form of hosting service and a Domain. See our beginner-friendly article about web hosting for more information.

**Functionalities**

* WampServer`s functionalities are very complete and easy to use so we won`t explain here how to use them. With a left click on WampServer`s icon, you will be able to:
* Manage your Apache and MySQL services
* Switch online/offline (give access to everyone or only localhost)
* Install and switch Apache, MySQL and PHP releases
* Manage your servers settings
* Access your logs
* Access your settings files
* Create Alias
* Change WampServer`s menu language
* Access this Page

**2.5.2. BACK-END**

**What is MySQL? – An Introduction To Database Management Systems**

Database Management is the most important part when you have humungous data around you. MySQL is one of the most famous Relational Database to store & handle your data. In this **What is MySQL**blog, you will be going through the following topics:

**What are Data & Database?**

Suppose a company needs to store the names of hundreds of employees working in the company in such a way that all the employees can be individually identified. Then, the company collects the **data** of all those employees. Now, when I say data, I mean that the company collects distinct pieces of information about an object. So, that object could be a real-world entity such as people, or any object such as a mouse, laptop etc.

**Database Management System & Types of DBMS**

A Database Management System (DBMS) is a software application that interacts with the user, applications and the database itself to capture and analyze data. The data stored in the database can be modified, retrieved and deleted, and can be of any type like strings, numbers, images etc.

**Types of DBMS**

There are mainly 4 types of DBMS, which are Hierarchical, Relational, Network, and Object-Oriented DBMS.

* **Hierarchical DBMS:**As the name suggests, this type of DBMS has a style of predecessor-successor type of relationship. So, it has a structure similar to that of a tree, wherein the nodes represent records and the branches of the tree represent fields.
* **Relational DBMS (RDBMS):** This type of DBMS, uses a structure that allows the users to identify and access data in relation to another piece of data in the database.
* **Network DBMS:**This type of DBMS supports many to many relations wherein multiple member records can be linked.
* **Object-oriented DBMS:**This type of DBMS uses small individual software called objects. Each object contains a piece of data, and the instructions for the actions to be done with the data.

**Angular JS**

AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. AngularJS's data binding and dependency injection eliminate much of the code you would otherwise have to write. And it all happens within the browser, making it an ideal partner with any server technology.

AngularJS is what HTML would have been, had it been designed for applications. HTML is a great declarative language for static documents. It does not contain much in the way of creating applications, and as a result building web application is an exercise in what do I have to do to trick the browser into doing what I want?

The impedance mismatch between dynamic applications and static documents is often solved with:

* **a library** - a collection of functions which are useful when writing web apps. Your code is in charge and it calls into the library when it sees fit. E.g., jQuery.
* **frameworks** - a particular implementation of a web application, where your code fills in the details. The framework is in charge and it calls into your code when it needs something app specific. E.g., durandal, ember, etc.

AngularJS takes another approach. It attempts to minimize the impedance mismatch between document centric HTML and what an application needs by creating new HTML constructs. AngularJS teaches the browser new syntax through a construct we call directives. Examples include:

* Data binding, as in {{}}.
* DOM control structures for repeating, showing and hiding DOM fragments.
* Support for forms and form validation.
* Attaching new behavior to DOM elements, such as DOM event handling.
* Grouping of HTML into reusable components.

**A complete client-side solution**

AngularJS is not a single piece in the overall puzzle of building the client-side of a web application. It handles all of the DOM and AJAX glue code you once wrote by hand and puts it in a well-defined structure. This makes AngularJS opinionated about how a CRUD (Create, Read, Update, Delete) application should be built. But while it is opinionated, it also tries to make sure that its opinion is just a starting point you can easily change. AngularJS comes with the following out-of-the-box:

* Everything you need to build a CRUD app in a cohesive set: Data-binding, basic templating directives, form validation, routing, deep-linking, reusable components and dependency injection.
* Testability story: Unit-testing, end-to-end testing, mocks and test harnesses.
* Seed application with directory layout and test scripts as a starting point.

**AngularJS's sweet spot**

AngularJS simplifies application development by presenting a higher level of abstraction to the developer. Like any abstraction, it comes at a cost of flexibility. In other words, not every app is a good fit for AngularJS. AngularJS was built with the CRUD application in mind. Luckily CRUD applications represent the majority of web applications. To understand what AngularJS is good at, though, it helps to understand when an app is not a good fit for AngularJS.

Games and GUI editors are examples of applications with intensive and tricky DOM manipulation. These kinds of apps are different from CRUD apps, and as a result are probably not a good fit for AngularJS. In these cases, it may be better to use a library with a lower level of abstraction, such as jQuery.

**Dreamweaver**

**Adobe Dreamweaver** is a [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) [web development](https://en.wikipedia.org/wiki/Web_development) tool from [Adobe Inc.](https://en.wikipedia.org/wiki/Adobe_Inc.). It was created by [Macromedia](https://en.wikipedia.org/wiki/Macromedia) in 1997and developed by them until Macromedia was acquired by Adobe Systems in 2005.

Adobe Dreamweaver is available for the [macOS](https://en.wikipedia.org/wiki/MacOS) and [Windows](https://en.wikipedia.org/wiki/Windows) [operating systems](https://en.wikipedia.org/wiki/Operating_system). Following Adobe's acquisition of the Macromedia product suite, releases of Dreamweaver subsequent to version 8.0 have been more compliant with [W3C](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) standards. Recent versions have improved support for [Web](https://en.wikipedia.org/wiki/World_Wide_Web) technologies such as [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets), [JavaScript](https://en.wikipedia.org/wiki/JavaScript), and various [server-side scripting](https://en.wikipedia.org/wiki/Server-side_scripting) [languages](https://en.wikipedia.org/wiki/Programming_language) and [frameworks](https://en.wikipedia.org/wiki/Software_framework) including [ASP](https://en.wikipedia.org/wiki/Active_Server_Pages) (ASP JavaScript, ASP VBScript, ASP.NET C#, ASP.NET VB), [ColdFusion](https://en.wikipedia.org/wiki/ColdFusion), [Scriptlet](https://en.wikipedia.org/wiki/Scriptlet), and [PHP](https://en.wikipedia.org/wiki/PHP).

Adobe Dreamweaver CC is a web design and development application that uses both a visual design surface known as Live View and a code editor with standard features such as [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting), [code completion](https://en.wikipedia.org/wiki/Code_completion), and code collapsing as well as more advanced features such as real-time [syntax checking](https://en.wikipedia.org/wiki/Syntax_analysis) and code introspection for generating code hints to assist the user in writing code. Combined with an array of site management tools, Dreamweaver allows for its users to design, code and manage websites, as well as mobile content. Dreamweaver is an [Integrated Development Environment](https://en.wikipedia.org/wiki/Integrated_Development_Environment) (IDE) tool. You can live preview of changes for the frontend. Dreamweaver is positioned as a versatile web design and development tool that enables visualization of web content while coding. Dreamweaver, like [other HTML editors](https://en.wikipedia.org/wiki/Comparison_of_WYSIWYG_HTML_editors), edits [files](https://en.wikipedia.org/wiki/Computer_file) locally then uploads them to the remote web server using [FTP](https://en.wikipedia.org/wiki/File_Transfer_Protocol), [SFTP](https://en.wikipedia.org/wiki/SSH_file_transfer_protocol), or [WebDAV](https://en.wikipedia.org/wiki/WebDAV). Dreamweaver CS4 now supports the [Subversion (SVN)](https://en.wikipedia.org/wiki/Subversion_(software)) version control system

**2.6 PROJECT DESCRIPTION**

**2.6.1. NAME OF THE MODULES**

**Owner**

* Login
* Registration
* Upload House Status

**Customer**

* Login
* Registration
* View house rent details
* Confirmation
* Feedback

**2.6.2. MODULE DESCRIPTION**

**Owner Login**:

Owner login the house rent management system. It is web based application. User can able to access anywhere and anytime. Create username and password to submit the house rent system by owner. Further update house rent details in website.

**Owner registration:**

The registration module is collect the house details to register in house rent management system. the owner registered name, phone number, email, house location, house rent details and etc. the details are visible for customer to take house rent.

**Update house details**

The owner update house rent, add room facilities, update available bathrooms, add garden details and etc. the above details are updated in house rent management system.

**Customer login**

The customer login the house rent management system using username and password. the customer checked all information about house.

**Customer registration**

The online house rent management system in Registration modules is used to collect the user personal information. The customer register name, address, phone number, email, house details and house rent details. The registration module details are stored in house management database. The Registration modules can be implemented for customer validation.

**View house rent details**

The customer can view room details, pictures, location, rent etc. if the customer liked house location and also booking in online. the customer paid house rent using online or offline transaction.

**Confirmation**

This module is operated by a customer who needs to confirmation with the rent house. A customer booking all details and confirm the house. The customer is confirmation in house rent management system.

**Feedback**

The user can provide their feedback about the entire process whether it may be positive or negative and the feedback can be viewed by the owner.

**3.DESIGN**

**3.1. DATA FLOW DIAGRAM**

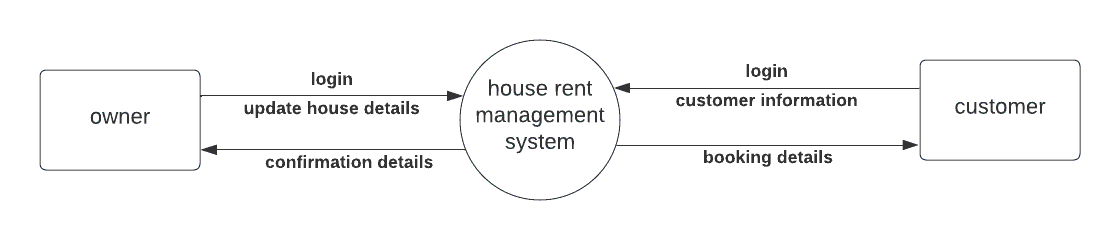
A two-dimensional diagram that explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a Cable Operator Management on output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

**Data flow symbols:**

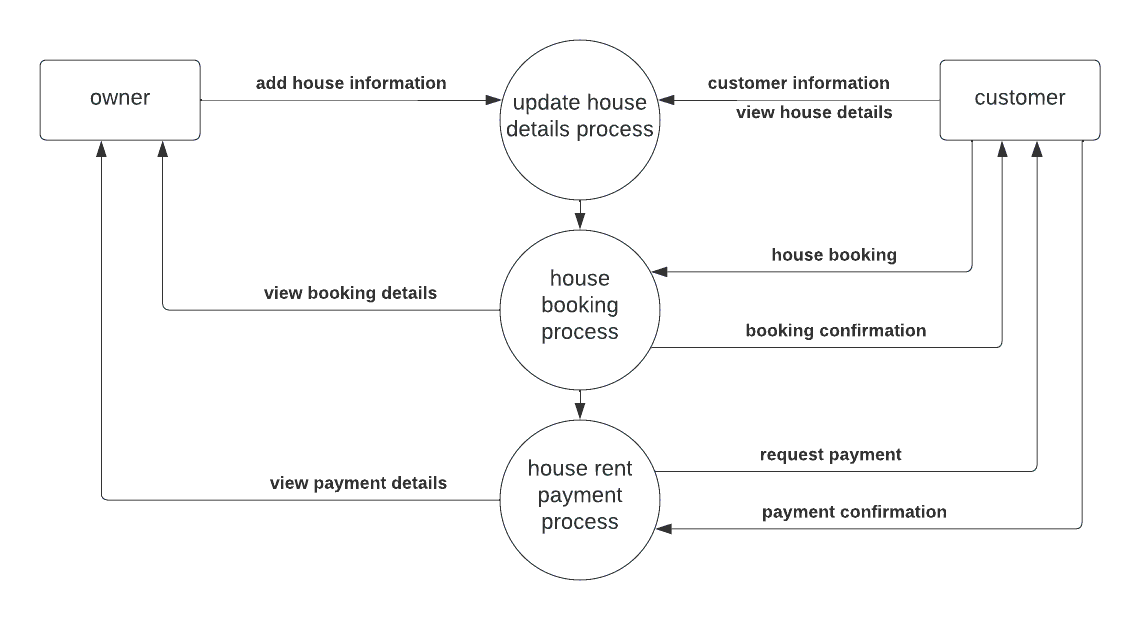
|  |  |
| --- | --- |
| **Symbol** | **Description** |
|  | An **entity**. A source of data or a destination for data. |
|  | A **process** or task that is performed by the system. |
|  | A **data store**, a place where data is held between processes. |
|  | A **data flow**. |

**Data flow diagram**

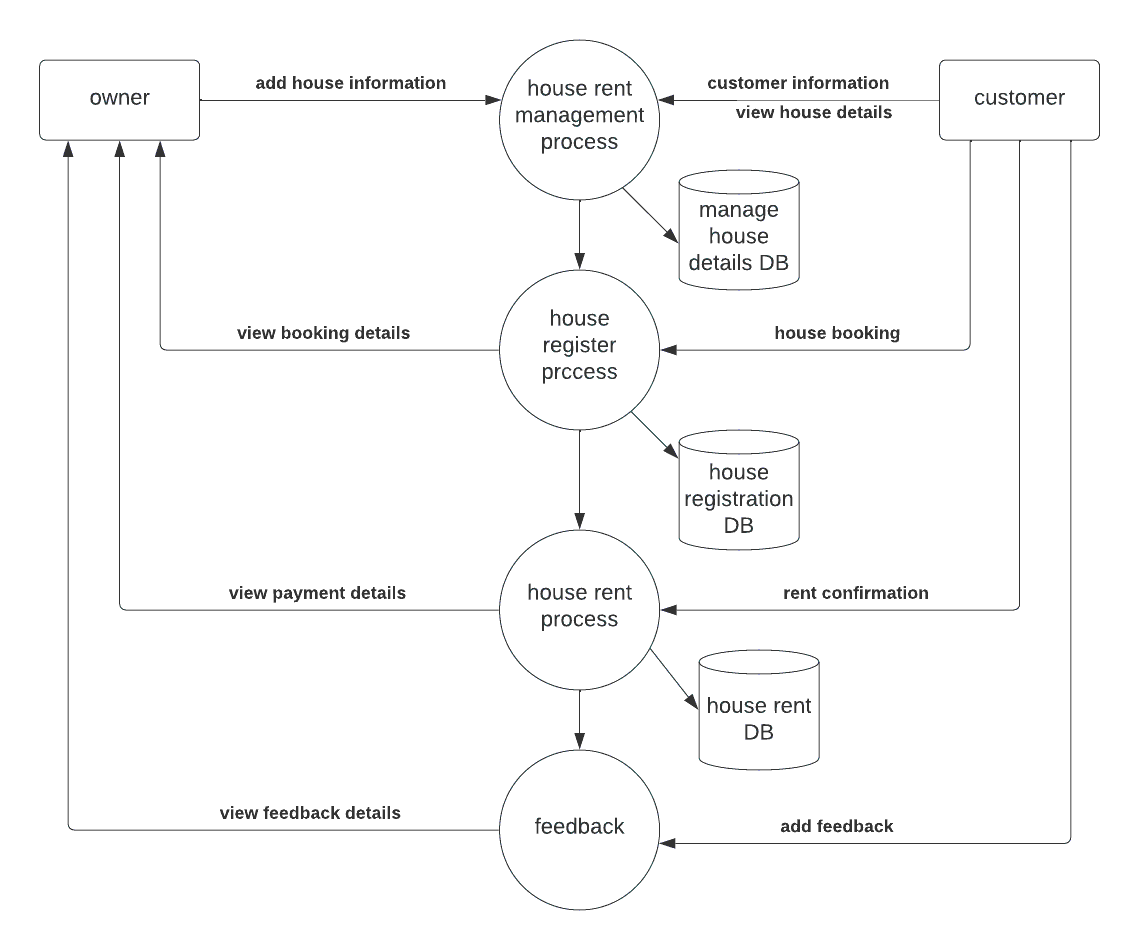
**level 0**



**level 1**



**3.2.USE CASE DIAGRAM:**



**3.3.DATABASE DESIGN**

**Table structure for table tbl\_admin**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Default** |
| ***Id*** | int(11) | Yes | NULL |
| Email | varchar(252) | Yes | NULL |
| username | varchar(252) | Yes | NULL |
| password | varchar(50) | Yes | NULL |

**Table structure for table tbl\_house**

**Table structure for table tbl\_message**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Default** |
| ***Id*** | int(50) | Yes | NULL |
| from\_id | int(50) | Yes | NULL |
| to\_id | int(50) | Yes | NULL |
| Message | text | Yes | NULL |
| read\_message | int(11) | Yes | NULL |
| Time | timestamp | Yes | CURRENT\_TIMESTAMP |

**Table structure for table tbl\_rentrequest**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Default** |
| ***Id*** | int(50) | Yes | NULL |
| house\_id | int(50) | Yes | NULL |
| tenant\_id | int(50) | Yes | NULL |
| owner\_id | int(50) | Yes | NULL |

**Table structure for table tbl\_user**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Default** |
| ***Id*** | int(11) | Yes | NULL |
| User | enum('owner', 'tenant') | Yes | NULL |
| Fname | varchar(32) | Yes | NULL |
| Lname | varchar(32) | Yes | NULL |
| Email | varchar(52) | Yes | NULL |
| password | varchar(152) | Yes | NULL |
| Nid | int(20) | Yes | NULL |
| Address | Text | Yes | NULL |
| Pic | varchar(32) | Yes | NULL |
| phone\_number | varchar(20) | Yes | NULL |
| Account | int(10) | Yes | NULL |
| description | Text | Yes | NULL |

**4.IMPLEMENTATION**

**4.1. SOURCE CODE**

**Index**

<?php

include 'inc/header.php';

Session::checkAdmin();

include 'inc/sidebar.php';

?>

<?php

if (isset($\_POST['logout'])) {

Session::sessionDestroyadmin();

}

?>

<div class="main-panel">

<?php

include 'inc/topnav.php';

include 'inc/main\_content.php';

include 'inc/footer.php';

?>

</div>

<?php

include 'inc/jsarea.php';

?>

**Admin login**

<?php

include\_once 'admincontroller/Login.php';

include\_once '../lib/Session.php';

Session::init();

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Document</title>

<link rel="stylesheet" href="../assets/css/bootstrap.min.css">

</head>

<style>

input{

margin-bottom: 15px;

}

form{

margin: 0px auto;

width: 460px;

background-color: aliceblue;

padding: 10px;

margin-top: 100px;

box-shadow: 0px 0px 5px 1px grey;

}

.house\_image{

width: 320px;

height: 280px;

margin: 0px auto;

}

.house\_image img{

width: 100%;

height: 100%;

}

body{

background-color: lightgray;

}

.login\_main{

padding-top: 20px;

}

</style>

<script>

if(window.history.replaceState){

window.history.replaceState( null, null, window.location.href );

}

</script>

<body>

<div class="main">

<div class="login\_area">

<div class="login\_main container">

<?php

$login = new Login();

if(($\_SERVER["REQUEST\_METHOD"] == "POST") && isset($\_POST['login'])) {

$msg = $login->login($\_POST['username'],$\_POST['password']);

}

if((isset($msg)) && ($msg=='nouser')){

echo "<p class='alert alert-danger'>No User Found!</p>";

$msg = null;

}

else if((isset($msg)) && ($msg=='empty')){

echo "<p class='alert alert-danger'>Fill all the Field please!</p>";

$msg = null;

}

?>

<form action="<?php echo $\_SERVER['PHP\_SELF']; ?>" method="post">

<div class="house\_image">

<img src="../assets/images/loginhouse.png" alt="">

</div>

<div class="main\_input">

<input type="text" name="username" class="form-control" value="" placeholder="Username">

<input type="text" name="password" class="form-control" value="" placeholder="Password">

<p class="text-center"><input type="submit" class="btn btn-success" name="login" class="from-control" value="Login"></p>

</div>

</form>

</div>

</div>

</div>

<script src="../assets/js/jquery.min.js" charset="utf-8"></script>

<script src="../assets/js/bootstrap.min.js" charset="utf-8"></script>

</body>

</html>

**About us**

<?php

include 'inc/header.php';

Session::checkAdmin();

include 'inc/sidebar.php';

?>

<div class="main-panel">

<?php

include 'inc/topnav.php';

?>

<div class="content">

<div class="container-fluid">

<div class="card" style="padding: 20px 5px;">

<div class="card-body" style="padding:10px;">

<form class="" action="<?php echo $\_SERVER['PHP\_SELF']; ?>" method="post">

<textarea style="background-color:#fff;border:1px solid black; margin-bottom:10px;height:500px;" class="form-control" name="about"></textarea>

<input type="submit" class="btn btn-primary" name="about" value="Update About">

</form>

</div>

</div>

</div>

</div>

<?php

include 'inc/footer.php';

?>

</div>

<?php

include 'inc/jsarea.php';

?>

**Edit details**

<?php

include 'inc/header.php';

Session::checkAdmin();

include 'inc/sidebar.php';

?>

<div class="main-panel">

<?php

include 'inc/topnav.php';

include 'Admincontroller/Tenant.php';

?>

<?php

$owner = new Tenant();

$alluser = $owner->alltenant();

?>

<div class="content">

<div class="container-fluid">

<div class="card" style="padding: 20px 5px;">

<div class="card-body">

<table id="table\_id" class="display">

<thead>

<tr>

<th>Serial</th>

<th>Name</th>

<th>Gmail</th>

<th>address</th>

<th>View</th>

</tr>

</thead>

<tbody>

<?php

if($alluser){

$i = 0;

foreach($alluser as $user){

$i++;

?>

<tr>

<td><?php echo $i; ?></td>

<td><?php echo $user['fname'].' '.$user['lname']; ?></td>

<td><?php echo $user['email']; ?></td>

<td style="padding-left:75px;"><?php echo $user['address']; ?></td>

<td> <a class="btn btn-info" href="tenantdetails.php?id=<?php echo $user['id']; ?>" >view</a> </td>

</tr>

<?php } }else{ echo "<p>There is no user!</p>"; } ?>

</tbody>

</table>

</div>

</div>

</div>

</div>

<?php

include 'inc/footer.php';

?>

</div>

<?php

include 'inc/jsarea.php';

?>

**Owner register**

<?php

include 'inc/header.php';

Session::checkAdmin();

include 'inc/sidebar.php';

?>

<div class="main-panel">

<?php

include 'inc/topnav.php';

include\_once 'admincontroller/Owner.php';

?>

<?php

$owner = new Owner();

$alluser = $owner->allowner();

?>

<div class="content">

<div class="container-fluid">

<div class="card" style="padding: 20px 5px;">

<div class="card-body">

<table id="table\_id" class="display">

<thead>

<tr>

<th>Serial</th>

<th>Name</th>

<th>Gmail</th>

<!-- <th>Number of House</th> -->

<th>View</th>

</tr>

</thead>

<tbody>

<?php

if($alluser){

$i = 0;

foreach($alluser as $user){

$i++;

?>

<tr>

<td><?php echo $i; ?></td>

<td><?php echo $user['fname'].' '.$user['lname']; ?></td>

<td><?php echo $user['email']; ?></td>

<!-- <td style="padding-left:75px;"></td> -->

<td> <a class="btn btn-info" href="ownerdetails.php?id=<?php echo $user['id']; ?>" >view</a> </td>

</tr>

<?php } }else{ echo "<p>There is no user!</p>"; } ?>

</tbody>

</table>

</div>

</div>

</div>

</div>

<?php

include 'inc/footer.php';

?>

</div>

<?php

include 'inc/jsarea.php';

?>

**Customer request**

<?php

include 'inc/header.php';

Session::checkAdmin();

include 'inc/sidebar.php';

?>

<div class="main-panel">

<?php

include 'inc/topnav.php';

include 'Admincontroller/Tenant.php';

?>

<?php

$tenant = new Tenant();

$alluser = $tenant->getMessage();

if (isset($\_POST['id'])) {

$tenant->delmessage($\_POST['id']);

}

?>

<div class="content">

<div class="container-fluid">

<div class="card" style="padding: 20px 5px;">

<div class="card-body">

<table id="table\_id" class="display">

<thead>

<tr>

<th>Serial</th>

<th>From</th>

<th>To</th>

<th>Message</th>

<th>Action</th>

</tr>

</thead>

<tbody>

<?php

if($alluser){

$i = 0;

foreach($alluser as $user){

$i++;

?>

<tr>

<td><?php echo $i; ?></td>

<td><?php echo $user['from\_id']; ?></td>

<td><?php echo $user['to\_id']; ?></td>

<td style="padding-left:75px;"><?php echo $user['message']; ?></td>

<td>

<form class="" action="allmessage.php" method="post">

<input type="hidden" name="id" value="<?php echo $user['id']; ?>">

<input class="btn btn-danger" type="submit" name="delmsg" value="Delete">

</form>

</td>

</tr>

<?php } }else{ echo "<p>There is no user!</p>"; } ?>

</tbody>

</table>

</div>

</div>

</div>

</div>

<?php

include 'inc/footer.php';

?>

</div>

<?php

include 'inc/jsarea.php';

?>

**House rental details**

<?php

include 'inc/header.php';

Session::checkAdmin();

include 'inc/sidebar.php';

?>

<div class="main-panel">

<?php

include 'inc/topnav.php';

include 'Admincontroller/Owner.php';

?>

<?php

$owner = new Owner();

if(isset($\_GET['id'])){

$singleuser = $owner->singleowner($\_GET['id']);

}

if($\_SERVER['REQUEST\_METHOD'] == 'POST' && isset($\_POST['delete'])){

$del = $owner->delowner($\_GET['id']);

}

?>

<div class="content">

<div class="container-fluid">

<div class="card owner\_details" style="padding: 20px 5px;">

<div class="card-body">

<div class="profile\_pic">

<img src="../assets/images/owner.png" alt="">

</div>

<table>

<tr>

<td>Name:</td>

<td><?php echo $singleuser['fname']. ' '.$singleuser['lname']; ?></td>

</tr>

<tr>

<td>Number of House:</td>

<td><?php echo $singleuser['house\_number']; ?></td>

</tr>

<tr>

<td>email:</td>

<td><?php echo $singleuser['email']; ?></td>

</tr>

<tr>

<td>address:</td>

<td><?php echo $singleuser['address']; ?></td>

</tr>

<tr>

<td>phone:</td>

<td><?php echo $singleuser['phone\_number']; ?></td>

</tr>

<tr>

<td>National ID:</td>

<td><?php echo $singleuser['nid']; ?></td>

</tr>

</table>

<div class="action\_btn">

<form style="display:inline-block;" class="" action="ownerdetails.php?id=<?php echo $user['id']; ?>" method="post">

<input type="submit" name="delete" class="btn btn-danger" value="delete">

</form>

<a href="allowner.php" class="btn btn-default">Back</a>

</div>

</div>

</div>

</div>

</div>

<?php

include 'inc/footer.php';

?>

</div>

<?php

include 'inc/jsarea.php';

?>

**Tenant details**

<?php

include 'inc/header.php';

Session::checkAdmin();

include 'inc/sidebar.php';

?>

<div class="main-panel">

<?php

include 'inc/topnav.php';

include 'Admincontroller/Tenant.php';

?>

<?php

$tenant = new Tenant();

if(isset($\_GET['id'])){

$singleuser = $tenant->singletenant($\_GET['id']);

}

if($\_SERVER['REQUEST\_METHOD'] == 'POST' && isset($\_POST['delete'])){

$del = $tenant->deltenant($\_GET['id']);

}

?>

<div class="content">

<div class="container-fluid">

<div class="card owner\_details" style="padding: 20px 5px;">

<div class="card-body">

<div class="profile\_pic">

<img src="../assets/images/owner.png" alt="">

</div>

<table>

<tr>

<td>Name:</td>

<td><?php echo $singleuser['fname']. ' '.$singleuser['lname']; ?></td>

</tr>

<tr>

<td>email:</td>

<td><?php echo $singleuser['email']; ?></td>

</tr>

<tr>

<td>address:</td>

<td><?php echo $singleuser['address']; ?></td>

</tr>

<tr>

<td>phone:</td>

<td><?php echo $singleuser['phone\_number']; ?></td>

</tr>

<tr>

<td>National ID:</td>

<td><?php echo $singleuser['nid']; ?></td>

</tr>

</table>

<div class="action\_btn">

<form style="display:inline-block;" class="" action="ownerdetails.php?id=<?php echo $user['id']; ?>" method="post">

<input type="submit" name="delete" class="btn btn-danger" value="delete">

</form>

<a href="allowner.php" class="btn btn-default">Back</a>

</div>

</div>

</div>

</div>

</div>

<?php

include 'inc/footer.php';

?>

</div>

<?php

include 'inc/jsarea.php';

?>

**User registration**

<?php

include 'inc/header.php';

include 'inc/navbar.php';

include 'Controller/Register.php';

?>

<div class="register\_area">

<div class="register\_main container">

<?php

$reg = new Register();

if(($\_SERVER["REQUEST\_METHOD"] === "POST") && isset($\_POST['register'])) {

$msg = $reg->register($\_POST);

}

if((isset($msg)) && ($msg=='success')){

echo "<p class='alert alert-success'>You registered Successfully!</p>";

$msg = null;

}

else if((isset($msg)) && ($msg=='fail')){

echo "<p class='alert alert-danger'>Unfortunately Not Registered!</p>";

$msg = null;

}

else if((isset($msg)) && ($msg=='empty')){

echo "<p class='alert alert-danger'>Fill all the Field please!</p>";

$msg = null;

}

else if((isset($msg)) && ($msg=='smallpass')){

echo "<p class='alert alert-danger'>Password length must be 6 or more characters!</p>";

$msg = null;

}

else if((isset($msg)) && ($msg=='emailprob')){

echo "<p class='alert alert-danger'>Email Format is Invalid!</p>";

$msg = null;

}

?>

<form enctype="multipart/form-data" action="<?php echo $\_SERVER['PHP\_SELF']; ?>" method="post">

<div class="house\_pic">

<img src="assets/images/house.jpg" alt="">

</div>

<div class="reg\_title">

<p class="text-center">Register</p>

</div>

<select class="form-control" id="reg\_as" name="user" style="margin-top:10px;background-color:lavender;" required="true">

<option value="" hidden>Register As</option>

<option value="1">Owner</option>

<option value="2">Tenant</option>

</select>

<input type="text" class="form-control" name="fname" placeholder="First Name">

<input type="text" class="form-control" name="lname" placeholder="Last Name">

<input type="text" class="form-control" name="email" placeholder="Email">

<input type="password" name="password" class="form-control" placeholder="Password">

<p class="text-right clear"> <input type="submit" class="btn btn-primary loginbtn" name="register" value="Register"> </p>

<p class="text-center"> Already a member? <a href="user\_login.php"> Sign In! </a></p>

</form>

</div>

</div>

<?php

include 'inc/footer.php';

?>

**User login**

<?php

include 'inc/header.php';

if(Session::get('login')==true && (Session::get('user') == 'owner' || Session::get('user') == 'tenant')) {

Header('Location:index.php');

}

include 'inc/navbar.php';

include 'Controller/Login.php';

?>

<div class="register\_area">

<div style="padding:40px 0px;min-height:600px;" class="register\_main container">

<?php

$login = new Login();

if(($\_SERVER["REQUEST\_METHOD"] === "POST") && isset($\_POST['login'])) {

$msg = $login->login($\_POST['email'],$\_POST['password']);

}

if((isset($msg)) && ($msg=='nouser')){

echo "<p class='alert alert-danger'>No User Found!</p>";

$msg = null;

}

else if((isset($msg)) && ($msg=='empty')){

echo "<p class='alert alert-danger'>Fill all the Field please!</p>";

$msg = null;

}

else if((isset($msg)) && ($msg=='emailprob')){

echo "<p class='alert alert-danger'>Email Format is Invalid!</p>";

$msg = null;

}

?>

<form class="" action="<?php echo $\_SERVER['PHP\_SELF'];?>" method="post">

<div class="house\_pic">

<img src="assets/images/house.jpg" alt="">

</div>

<div class="login\_title">

<p class="text-center">Login</p>

</div>

<input type="email" class="form-control" name="email" placeholder="Email">

<input type="password" name="password" class="form-control" placeholder="Password">

<p class="text-right clear"> <input type="submit" class="btn btn-primary loginbtn" name="login" value="Login"> </p>

<p class="text-center"> Not a member? <a href="user\_register.php"> Sign Up ! </a></p>

</form>

</div>

</div>

<?php

include 'inc/footer.php';

?>

**View available house**

<?php

include 'inc/header.php';

include 'inc/navbar.php';

include\_once 'Controller/Homecontroller.php';

?>

<div class="available\_page\_area">

<?php

$home = new Homecontroller();

$data = $home->gethomeDetails();

if(!$data){

echo "<p>No data found</p>";

}

?>

<?php

if ($\_SERVER['REQUEST\_METHOD'] == 'POST' && isset($\_POST['search\_house'])) {

$arr = explode('-',$\_POST['rental\_value']);

$range1 = substr($arr[0],1);

$range2 = substr($arr[1],2);

$data = $home->searchHome($range1,$range2,$\_POST);

}

if ($\_SERVER['REQUEST\_METHOD'] == 'GET' && isset($\_GET['search\_cover'])) {

$arr = explode('-',$\_GET['rental\_value']);

$range1 = substr($arr[0],1);

$range2 = substr($arr[1],2);

$data = $home->searchHome($range1,$range2,$\_GET);

}

?>

<div class="available\_page\_main container">

<div class="search\_house">

<div class="search\_house\_inner card">

<div class="well search\_card card-body">

<form class="search\_house\_form" action="<?php echo $\_SERVER['PHP\_SELF']; ?>" method="post">

<input type="text" name="address" class="form-control" value="<?php if(isset($\_POST['address'])){

echo $\_POST['address'];

} ?>" placeholder="Address">

<select class="form-control" style="background-color:lavender;" name="house\_type">

<option value="" selected disabled>Rent Type</option>

<option value="Family"

<?php if(isset($\_POST['house\_type']) && $\_POST['house\_type']=='Family'){

echo "selected";

} ?>

>Family</option>

<option value="Bachelor"

<?php if(isset($\_POST['house\_type']) && $\_POST['house\_type']=='Bachelor'){

echo "selected";

} ?>

>Bachelor</option>

<option value="Sublet"

<?php if(isset($\_POST['house\_type']) && $\_POST['house\_type']=='Sublet'){

echo "selected";

} ?>

>Sub-Let</option>

<option value="Mess/Hostel"

<?php if(isset($\_POST['house\_type']) && $\_POST['house\_type']=='Mess/Hostel'){

echo "selected";

} ?>

>Hostel/Mess</option>

</select>

<div id="range">

<label for="input\_range">Price range:</label>

<input type="text" id="input\_range" name="rental\_value" readonly style="border:0; color:#f6931f; font-weight:bold;">

<div id="main\_range" class="myrange" title="Tap left or right button to set more precise value."></div>

</div>

<input type="submit" name="search\_house" class="btn btn-info" value="Search house">

</form>

</div>

</div>

</div>

<div class="all\_houses row">

<?php

foreach ($data as $value) {

?>

<div class="single\_houses card">

<div class="single\_house\_inner card-body">

<div class="house\_title">

<p style="font-weight:600;"> <i class="fas fa-map-marker-alt"></i> <?php echo $value['address']; ?> </p>

<p class="rent"> <i class="fas fa-money-check-alt"></i> <?php echo $value['rental\_value']; ?> </p>

</div>

<div class="house\_img">

<img src="assets/images/house/house29.png" alt="House">

</div>

<a href="housedetails.php?house\_id=<?php echo $value['id']; ?>">Details</a>

</div>

</div>

<?php } ?>

</div>

</div>

</div>

<?php

include 'inc/footer.php';

?>

<script>

$(function(){

$( "#main\_range" ).slider({

range: true,

min: 100,

max: 100000,

values: [<?php if(isset($range1) && isset($range2)){

echo $range1.','.$range2;}

else{?> 100,100000 <?php } ?>],

slide: function( event, ui ) {

$( "#input\_range" ).val( "$" + ui.values[ 0 ] + " - $" + ui.values[ 1 ] );

}

});

$( "#input\_range" ).val( "$" + $( "#main\_range" ).slider( "values", 0 ) +

" - $" + $( "#main\_range" ).slider( "values", 1 ) );

});

</script>

**Add house rental**

<?php

include 'inc/header.php';

include 'inc/navbar.php';

include\_once 'Controller/Homecontroller.php';

?>

<div class="available\_page\_area">

<?php

$home = new Homecontroller();

$data = $home->gethomeDetails();

if(!$data){

echo "<p>No data found</p>";

}

?>

<?php

if ($\_SERVER['REQUEST\_METHOD'] == 'POST' && isset($\_POST['search\_house'])) {

$arr = explode('-',$\_POST['rental\_value']);

$range1 = substr($arr[0],1);

$range2 = substr($arr[1],2);

$data = $home->searchHome($range1,$range2,$\_POST);

}

if ($\_SERVER['REQUEST\_METHOD'] == 'GET' && isset($\_GET['search\_cover'])) {

$arr = explode('-',$\_GET['rental\_value']);

$range1 = substr($arr[0],1);

$range2 = substr($arr[1],2);

$data = $home->searchHome($range1,$range2,$\_GET);

}

?>

<div class="available\_page\_main container">

<div class="search\_house">

<div class="search\_house\_inner card">

<div class="well search\_card card-body">

<form class="search\_house\_form" action="<?php echo $\_SERVER['PHP\_SELF']; ?>" method="post">

<input type="text" name="address" class="form-control" value="<?php if(isset($\_POST['address'])){

echo $\_POST['address'];

} ?>" placeholder="Address">

<select class="form-control" style="background-color:lavender;" name="house\_type">

<option value="" selected disabled>Rent Type</option>

<option value="Family"

<?php if(isset($\_POST['house\_type']) && $\_POST['house\_type']=='Family'){

echo "selected";

} ?>

>Family</option>

<option value="Bachelor"

<?php if(isset($\_POST['house\_type']) && $\_POST['house\_type']=='Bachelor'){

echo "selected";

} ?>

>Bachelor</option>

<option value="Sublet"

<?php if(isset($\_POST['house\_type']) && $\_POST['house\_type']=='Sublet'){

echo "selected";

} ?>

>Sub-Let</option>

<option value="Mess/Hostel"

<?php if(isset($\_POST['house\_type']) && $\_POST['house\_type']=='Mess/Hostel'){

echo "selected";

} ?>

>Hostel/Mess</option>

</select>

<div id="range">

<label for="input\_range">Price range:</label>

<input type="text" id="input\_range" name="rental\_value" readonly style="border:0; color:#f6931f; font-weight:bold;">

<div id="main\_range" class="myrange" title="Tap left or right button to set more precise value."></div>

</div>

<input type="submit" name="search\_house" class="btn btn-info" value="Search house">

</form>

</div>

</div>

</div>

<div class="all\_houses row">

<?php

foreach ($data as $value) {

?>

<div class="single\_houses card">

<div class="single\_house\_inner card-body">

<div class="house\_title">

<p style="font-weight:600;"> <i class="fas fa-map-marker-alt"></i> <?php echo $value['address']; ?> </p>

<p class="rent"> <i class="fas fa-money-check-alt"></i> <?php echo $value['rental\_value']; ?> </p>

</div>

<div class="house\_img">

<img src="assets/images/house/house29.png" alt="House">

</div>

<a href="housedetails.php?house\_id=<?php echo $value['id']; ?>">Details</a>

</div>

</div>

<?php } ?>

</div>

</div>

</div>

<?php

include 'inc/footer.php';

?>

<script>

$(function(){

$( "#main\_range" ).slider({

range: true,

min: 100,

max: 100000,

values: [<?php if(isset($range1) && isset($range2)){

echo $range1.','.$range2;}

else{?> 100,100000 <?php } ?>],

slide: function( event, ui ) {

$( "#input\_range" ).val( "$" + ui.values[ 0 ] + " - $" + ui.values[ 1 ] );

}

});

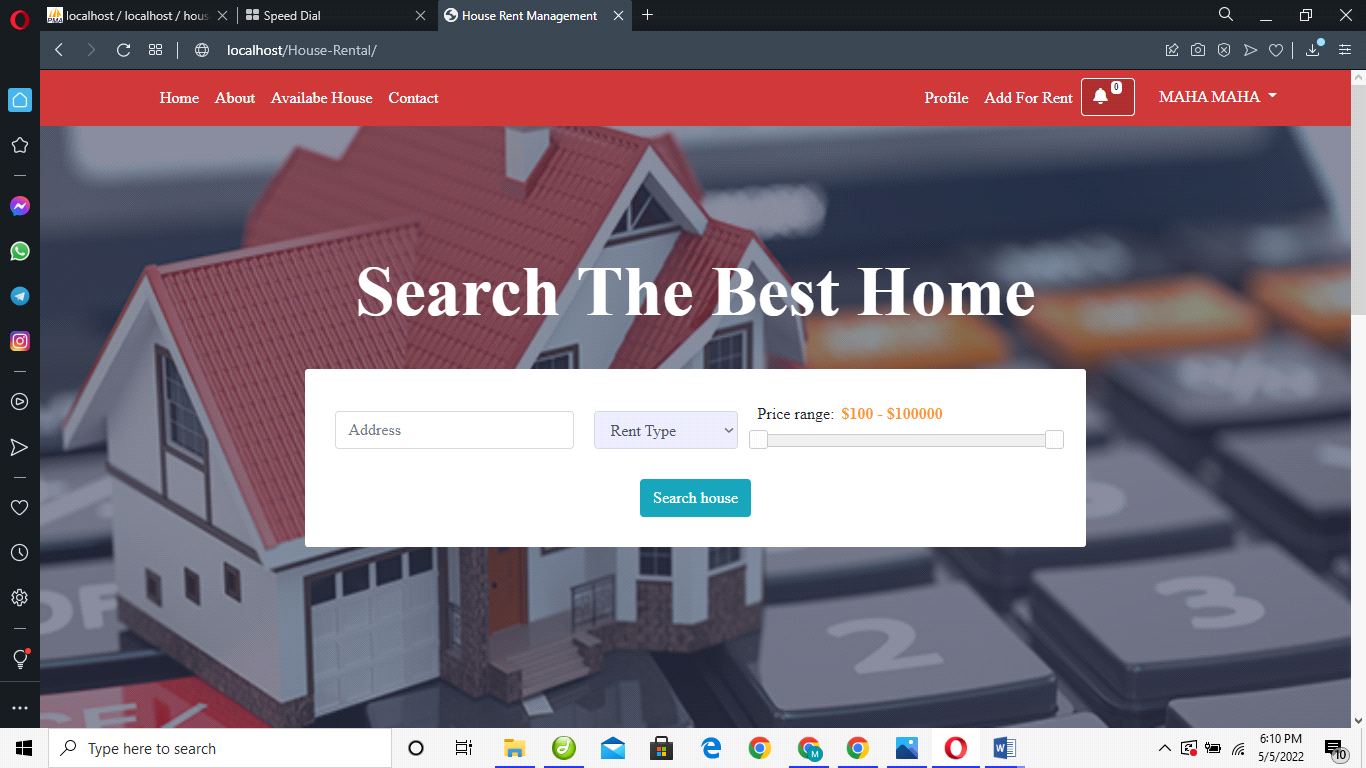
$( "#input\_range" ).val( "$" + $( "#main\_range" ).slider( "values", 0 ) +

" - $" + $( "#main\_range" ).slider( "values", 1 ) );

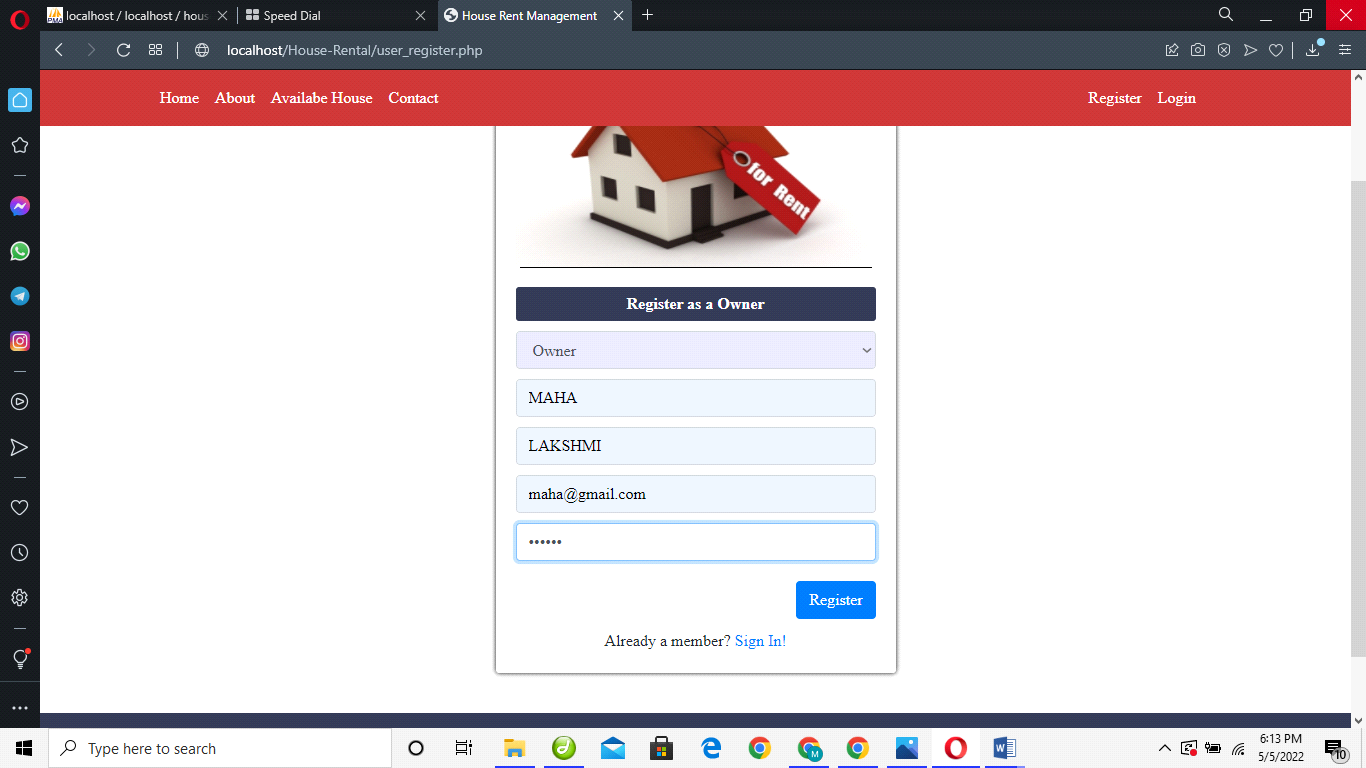
});

</script>

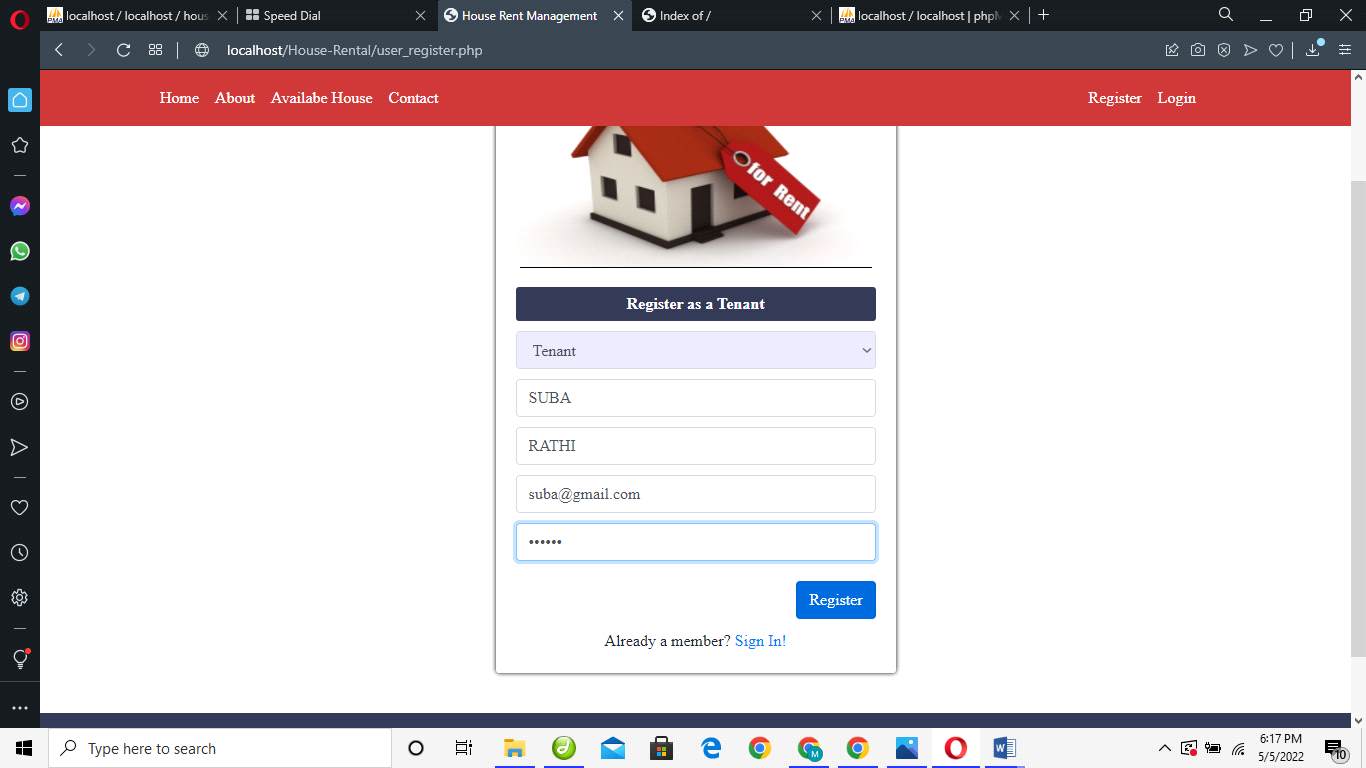
**4.2. SCREEN SHOTS**

**Home page**

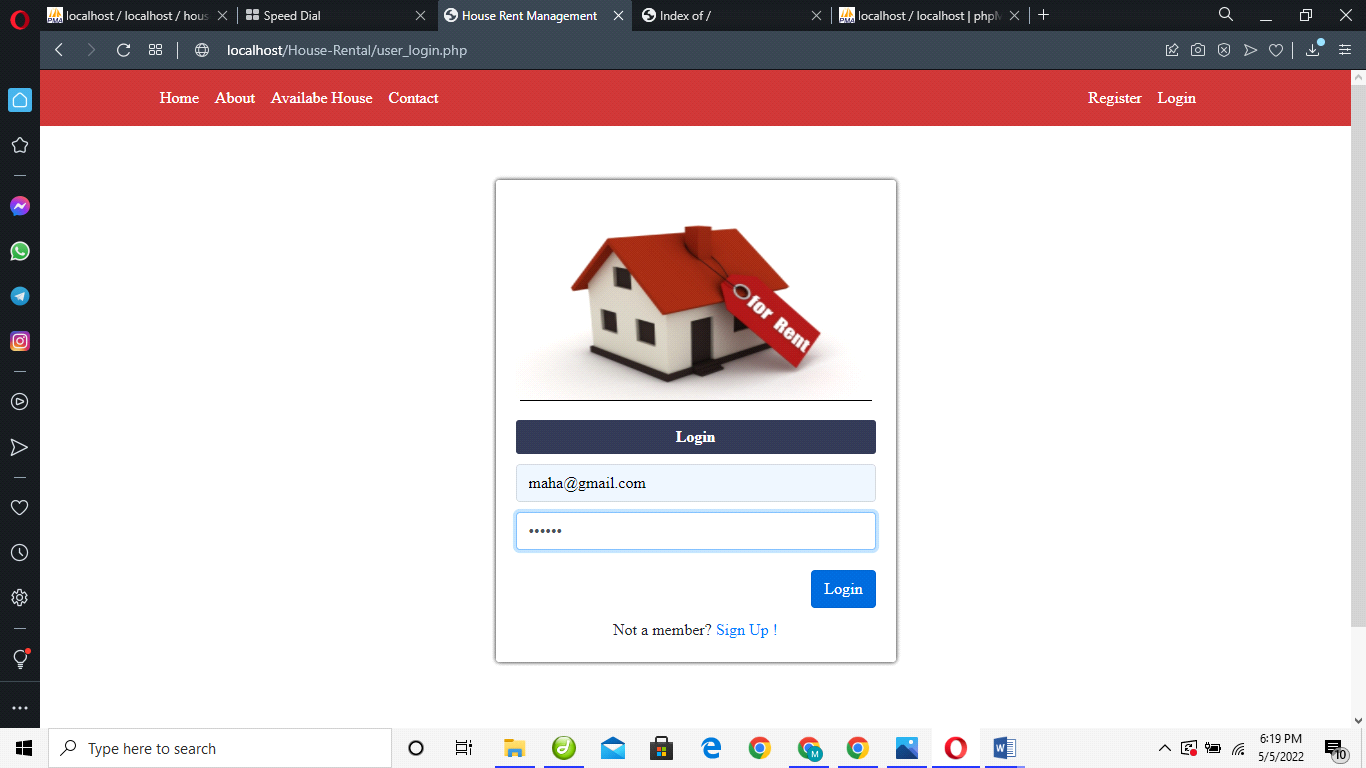
**Owner registration**



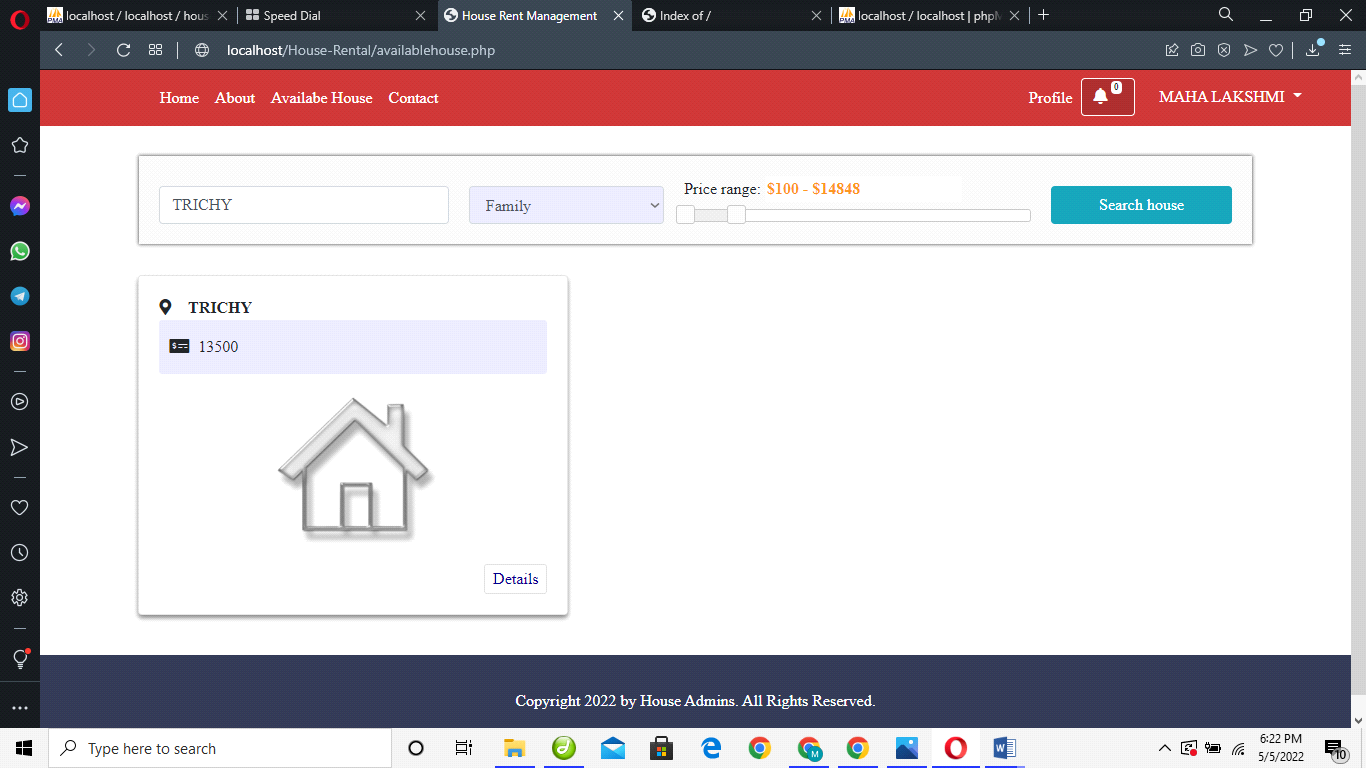
**Tenant registration**



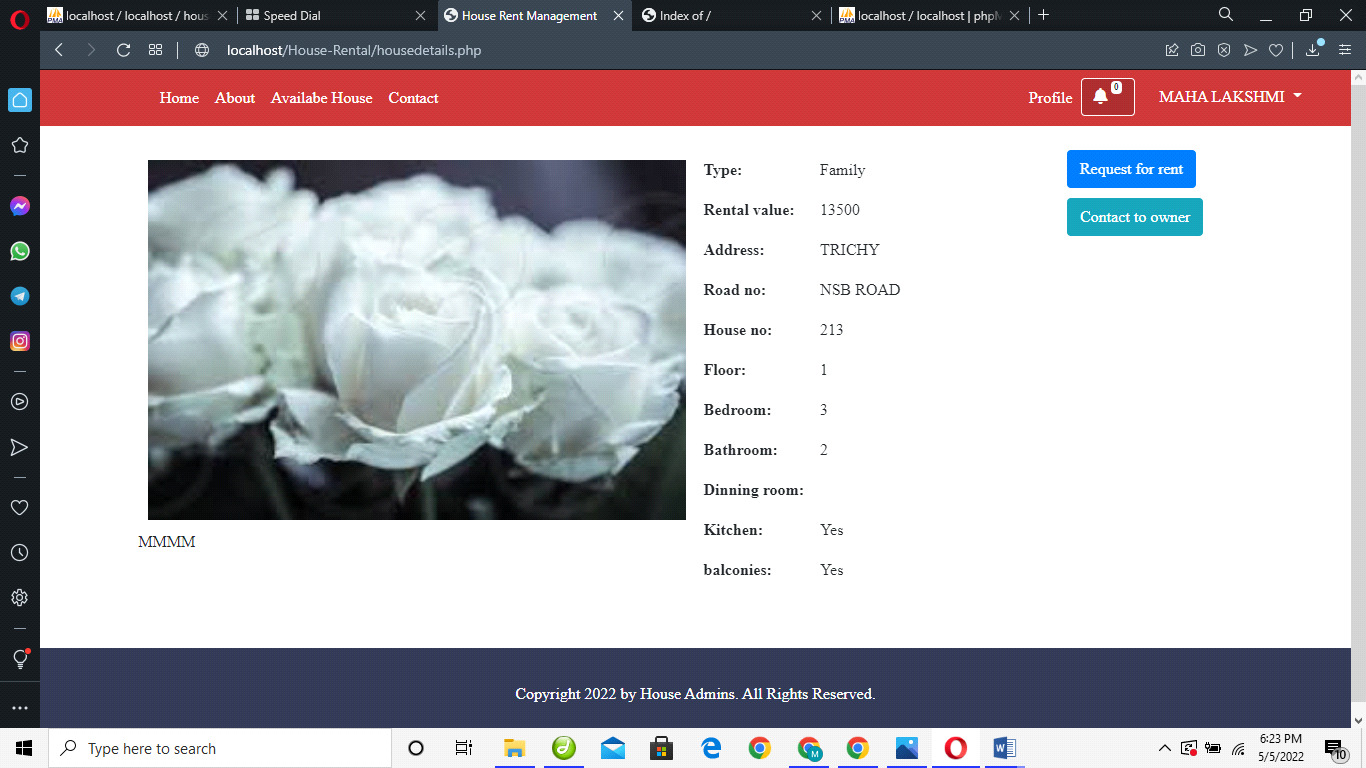
**House owner login**



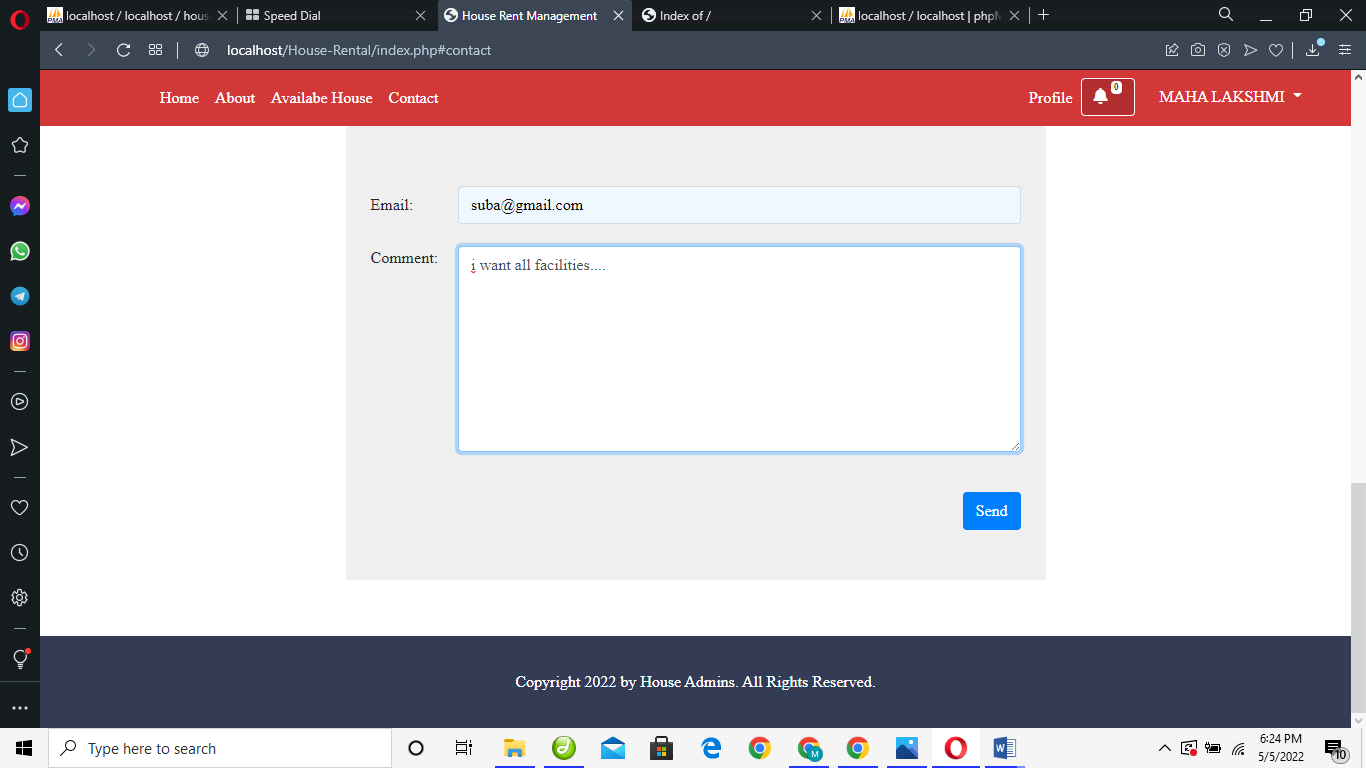
**View house details**



**Request for owner**



**Contact us**



**5. SYSTEM TESTING**

**5.1. TYPES OF TESTING**

**Test Case**

File level deduplication will save a relatively large memory space. In general, file level deduplication view multiple copies of same file. It stores first file and then it links other references to the first file. Only one copy will be stored. In testing, even though file names are same, the system can able to detect deduplication. If we upload the same file by using different names, it will view only the content and not names. Thus redundant data is avoided.

In registration phase, the user may not registered before and type their information. So if the user is new user, the alert message will display that the user is not registered before.

Fig 7.2 System Testing.

**Unit Testing**

It is the testing of an individual unit or group of related units. It is done by programmer to test that the implementation is producing expected output against given input and it falls under white box testing. Unit testing is done in order to check registration whether the user properly registered into the cloud. It is done in order to check whether a file is properly uploaded into the cloud. And an encryption and decryption is checked with unit testing if it is converted properly. Then deduplication is checked with unit testing.

**Integration Testing**

All the modules should be integrated into a single module and it should be checked that it is still working still by integration testing.

**System Testing**

It is done to ensure that by putting the software in different environments and check that it still works. System Testing is done by uploading same file in this cloud checking whether any duplicate file exists.

**Software Testing**

It is the process of evaluating a software item to detect differences between given input and expected output. Also to assess the feature of a software item. Testing assess the quality of the product. It is a process that should be done during the development process. In other words software testing is a verification and validation process.

There are two types of software testing.

1. Black box testing

2. White box testing

**Verification**

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

**Validation**

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

**Black Box Testing**

Black box testing is a testing which ignores internal mechanism of system and focuses on output generated against any input and execution of system. It is done for validation. It is done to check encryption and decryption after uploading a file into the cloud.

**White Box Testing**

It is done for verification and it is a testing that takes into account the internal mechanism of the system. It is done by checking content verification. It will verify that whether same content exists in the cloud.

6.**CONCLUSION**

This rental housing Web Application is a typical web application programming language. We provide the user select “City”, “Cost range”, “BHK”(Bedroom Hall Kitchen) and we provide two option buttons for the user to select whether he/she wants to buy or rent that property. Owner uploads the house and rent details then admin view and verifies all the details. All details its true admin approve the house then user can view all house details after choose and book a house for rent all module its work properly.

**7.BIBLIOGRAPHY**

Following books and eBooks are used to complete this project reports.

* [Advanced PHP Remoting](http://www.thinktecture.com/Resources/Books/default.html) 2nd Edition (Ingo Rammer and Mario Szpuszta, Apress, March 2005)
* [PHP to PHP Migration Handbook](http://www.thinktecture.com/Resources/Books/default.html) (Christian Nagel et al, Wrox, January 2003)
* [OF](http://www.thinktecture.com/Resources/Books/default.html) PHP (Christian Nagel et al, Wrox, September 2001)
* [Data-Centric PHP Programming](http://www.thinktecture.com/Resources/Books/default.html) (Christian Nagel et al, Wrox, December 2001)
* [Professional PHP Network Programming](http://www.thinktecture.com/Resources/Books/default.html) 2nd Edition (Christian Nagel et al, Wrox, September 2004)
* [Professional PHP](http://www.thinktecture.com/Resources/Books/default.html) (Christian Nagel et al, Wrox, March 2002)
* [Professional PHP Web Services](http://www.thinktecture.com/Resources/Books/default.html) (Christian Nagel et al, Wrox, December 2001)