**Abstract**

Technology has grown significantly in the past few years. This has led to the automation of the world that we are living in. It has almost become a basic need in any organization, system. Landlords have also sought the use of technology to help them manage their properties. There are many challenges that are being faced by property owners that need to be addressed by an effective system. Most of the challenges are associated by paperwork. Some documents can be lost or damaged incase of such scenarios as fire outbreak, heavy rains etc. It is time consuming to go through the files in search of a document especially when the filing system is not done properly. Tenants have to physically visit the premises to strike a contract with the landlords. Calculations can be messy when done manually and are prone to errors. The landlord has to go through the documents manually to keep track the validity of contracts to ensure that tenants pay their dues in time. The proposed rental house management system will help curb the challenges illustrated above and ensure that services offered to the customers are up to satisfaction. Majority of the tasks will be automated giving the management an easy time while running their businesses.

**CHAPTER ONE**

**INTRODUCTION**

**Background Information**

Rental House Management System is a web-based system that helps the owners of a house rental system to manage their business. It automates the activities involved in the managerial of rental houses. The system has one module which is the admin’s. The admin has been granted all the privileges. The admin registers a new house and the type of the house into the system. When a tenant rents the house, the admin registers them into the system to their respective houses. The admin then advances to the payments page and uploads the payments that the tenants have made. He has been granted the rights to edit information on a house, house type, tenants and payments. He can also delete the records. Such a scenario is when a tenant vacates. The system analyses the information on the payments and generates a monthly report and a balances report which can be printed by the admin for record keeping. The system as well determines the arrears and over payments on each tenant. This makes it easy to keep track of rents. It also helps update the due rents.

**Problem statement**

The demand for rental houses has skyrocketed in the recent days. This has been positively spearheaded by such factors as; people getting jobs away from their homes, students going to different places for schools e.g., universities, lack of enough lands, businesses, high population among others.

Property owners have made a lucrative business out of it. However, managing their properties has become cumbersome given the paperwork system widely used. Handling tenants is one huge challenge that is incurred by landlords who use the paperwork system. This involves loss of documents containing information on the house details, tenant details, payment records among others. This can breed a bad relationship between the landlord and the tenants. Retrieving documents that are stored in a paperwork system is time consuming. Papers are prone to physical damages such as fire outbreak, water spillage among others. Some files might be lost in the process or even stolen thus losing some sensitive information. Transporting documents in paperwork form is cumbersome. Paperwork filing is quite expensive given the cost of files, papers, printing and stationeries involved. Editing can be quite a mess given that once a document has been filled, you have to start a new duplicate copy to make changes.

House Rental Management System is a system designed to curb the challenges associated with paperwork filing system. There is no loss of information since records are kept digitally safe and back ups are in place to ensure safety of documents. It saves on space that would have incurred by storing papers, this reduces the cost for stationeries as well. Transportation/transfer of documents is easy compared to the paperwork systems since large amounts of documents can be shared through mails, or storage devices such as flash disks and memory cards. Editing can easily be done without creating a mess.

**Significance of study**

The main purpose of the House Rental management system include;

1. Reduction of the cost incurred in the managing of the house rental business.
2. Manage the renting of rooms
3. Maintain a clear digital paper trail for compliance purposes.
4. Maintain key property details

**Objectives of the project**

Specific objectives of the project are;

1. Reviewing the existing House Rental Management System which is paper-based
2. Coming up with an automated House Rental Management System
3. Implementing the automated House Rental Management System
4. Validating the automated House Rental Management System

**Project justification**

House Rental Management System is a system that will facilitate the management of rentals including maintenance, legalities and personnel all through a single web-based system. It will replace the old-fashioned paper-based system which tends to be cumbersome, time consuming, costly and thus inefficient. House Rental System is a web-based system with advanced features and an easy-to-use User Interface that facilitates the operations and control of rental houses.

**Scope of study**

The project scope defines the description of the work that is required in delivering the house rental management system. The following are the scopes of work during the course of the project:

1. Study and understand the requirement of this project
2. Construct Software Requirement Specification document of the system
3. Construct Software Design Document of the system

**CHAPTER TWO**

**LITERATURE REVIEW**

**Government strategy and incentives in the housing** **sector**

Some of the dominant strategies for housing and service provision for the Kenya’s urban poor include slum upgrading and site and service schemes. However, the efficiency of these strategies has been limited by ambivalent government attitude to irregular settlement. These strategies have failed because of a reliance on inappropriate building by-laws and infrastructural standards and modern designs, construction technology and conventional building materials that all make housing unaffordable to the poor, even after subsidies.

Thus, government initiatives in assisting house owners in management have proven to be pathetically slow with many of the houses provided being economically and socially irrelevant, this further prompting the rise of informal settlement (Macoloo, 1994).

**The role of the private sector in house management**

Private sector housing management is defined as any process which is not connected at all with the actions of the state neither directly constructed by state nor financially sponsored by the state where production is not expected to have a social element (Golland, 1996).

(Ambrose and Barlow, 1987) have argued that three factors are important in influencing the level of new house building. These are direct capital investment by the state for public housing, state support for production and consumption and changes in the profitability of house builders in the private sector.

The private sector can play an important role in housing provision provided that the state offers sufficient and appropriate incentives to the sector (Mitullar, 2003).

The clear motivation that underlies the private sector is profit (or potential profitability) with profit maximizing options being in the context of housing, producing and selling more of the product; reducing the cost of production through lower raw material and wage costs and finally increasing the price of the product or service (Hancock, 1998).\

Profitability in housing is advocated to be based on three variables; House prices, land prices and building costs, where:

Profit=House prices-{Land prices + Building costs} (Golland, 1996).

**The role of object-oriented programming (oop)**

The concept of object-oriented modeling is becoming increasingly practical because of its ability to thoroughly represent complex relationships as well as to represent data and data processes in a consistent manner. This concept has been implemented in computer software engineering; ranging from system analysis, system design, operating system, computer programming and database management system (Cohn, 1996).

**The role of relational database management system (RDBMS) (levin, 1999)**

Database Management System (DBMS) has replaced the file system data management by having a pool of data that can be shared by multiple application programs and users concurrently. DBMS also provide logical and physical data independence, so that changing of data structure or application program will not affect one another.

**CHAPTER THREE**

**METHODOLOGY**

**Facts finding techniques**

It shows how data will be collected from the users of the system. The data collection techniques to be used include:

**Objectives**

* It will use this technique to collect information about how the current system operates and its processes. This involves systematically watching and recording the behavior and characteristics of operations and processes.
* It gives more detailed and context related information and can adapt to event as they occur however the method may be time consuming.

**Questionnaires**

I will prepare a number of questionnaires whereby I will submit them to business owners (Landlords) to get a deeper insight of how the system is going to work. I prefer this method because it gives more information from various individuals and offers greater flexibility as the opportunity to restructure questions. This technique is preferred because it will provide a closer contact between the users and the developer hence dispelling the probability of the completed system being rejected by user(s). This technique also:

* Permits clarification
* Has high response rate than interviews.
* Helps get full range and depth of information

**Secondary Data Collection**

This data I will collect from existing sources e.g books, internet, journals and magazines that was collected by other researchers and analysis was done. It is from that data that I will then compare with the primary data and make a decision and conclusion.

**System Development and Methodology**

System development methodology is a technique that is used to show how the proposed system will be developed. In this case, the methodology used will be a waterfall model.

**Waterfall Model**

It is comprised of the stages that the developer will use when developing the system. It is a sequential model hence, the name waterfall. The developer has to finish with one stage before going to the next one. It comprises of the feasibility study, analysis phase, design phase, coding phase, testing phase, implementation phase and finally the maintenance phase. It is a simple model and easy to use and understand. With waterfall development-based methodologies, the analysts and users proceed sequentially from one phase to the next. The deliverables from each phase are voluminous and are presented to the project sponsor for approval as the project moves from phase to phase. Once the phase is approved by the sponsor it ends and the next phase begins.

**Diagram of waterfall model**

*Figure: waterfall model*

**Feasibility study**

Here, I will carry out a study to gain an understanding of the customers (tenants) current system and problems experienced in this system through interviews, observations, and participations. I will use the obtained data to determine the viability of the system being proposed in terms of technical, economic and social feasibility.

**Requirement and analysis**

At this stage, I will gather information about what the customer needs and define the problems the system is expected to solve. I will also include customers’ business context, products functions and its compatibility. I will gather requirement such as software like the programming language to use, database model and hardware needed such as laptop, printers etc

**Design**

At this stage, I will make an overall design of the system architecture and physical design which includes User Interface and Database design. It is at this stage that I will identify any faults before moving onto the next stage. The output of this stage is the design specification which is used in the next stage of implementation.

**Coding/Implementation**

At this stage, I will begin coding as per the design specification(s). The output of this step is one or more product components built according to a pre-defined coding standard and debugged, tested and integrated to satisfy the system architecture requirement.

**Testing**

At this stage, I will ensure both individual and integrated whole are methodically verified to ensure they are error free and satisfy customer requirement. I will involve both unit testing of individual code module, system testing of the integrated product and acceptance testing conducted by or on behalf of customer. I will ensure bugs found are corrected before moving to the next stage. I will also prepare, review and publish product documentation at this stage.

**Installation**

It is done once the product has been tested and certified as fit for use. The system is prepared for use at customer site.

**Maintenance**

This stage occurs after installation. It involves modifications on the system to improve performance. Such changes are user initiated or as a result of bug being discovered which were initially not known. These modifications are recorded for documentation and system update.

**CHAPTER FOUR**

**SYSTEM ANALYSIS AND REQUIREMENT MODELLING**

**Introduction**

The system objectives outlined during the feasibility study served as the basis from which the work of system design was initiated. Much of the activities involved at this stage were of technical nature requiring a certain degree of experience in designing systems sound knowledge of computer related technology and through understanding of computers available in the market and the various facilities provided by the vendors. Nevertheless, a system could not be designed in isolation without the active involvement of the user. The user had a vital role to play at this stage too.

Data collected during feasibility study was utilized systematically during the system design. Designing a system is a creative process which calls for logical as well as lateral thinking

Logical approach involves systematic moves towards the end product keeping in mind the capabilities of the personnel and the equipment at each design making step.

**Requirements analysis**

Requirement analysis involved defining customer needs and objectives in the context of planned customer use, environments and identified system characteristics to determine requirements for system functions.

**User Requirements**

It entailed user involvement and statements of facts and assumptions that define the expectations of the system in terms of mission objectives, environment, constraints and measures of effectiveness and suitability. Basically, the users:

* A system that improves on the efficiency of information storage and retrieval.
* A system that is easy to learn and use
* A system that is fast in processing transactions
* A system that is flexible, safe and convenient

**Functional Requirements**

This is a necessary task, action or activity that was accomplished. The proposed system is able to:

* Allow administrator to add a house and tenant details
* Allow the administrator to delete houses and tenant’s details
* Allow the administrator to search data in the database
* Allow the administrator to edit data in the database

**Hardware Requirements**

* Processor 2.0 GHZ processor speed and above
* Memory 2GB RAM and above
* 64 bit computer

**Software Requirements**

* Operating System- windows 7 and above
* Microsoft Office Power point- Used during presentation
* Apache server, xampp
* Sublime Text.
* web browser e.g. mozilla.

**CHAPTER FIVE**

**SYSTEM DESIGN**

**Design phases**

The user’s requirements document was analyzed for better understanding of what was required of the system. Ways of implementing these requirements were analyzed. Physical modules of the system were designed and identifying of the operating environment in which they were to work on. The system was a visual basic system/application. The database was updated each time the administrator; add, deletes or deletes data on the system.

It is only the administrator who has access to the system to view or make changes when necessary. The system was designed to allow the administrator to view, edit, delete and add data to the database

Each time a customer comes, he/she is registered in the tenant table of the database with other relevant details about the tenant.

System design involved transforming the software requirements into an architecture that described its top-level structure and identified the software components and developed a detailed design for each software components. For each requirement, a set of one or more design elements was produced.

**Conceptual Design**

Conceptual design was the very first phase of design in which drawings or solid models were the dominant tools and products. The conceptual design phase provided a description of the proposed system in terms of a set of integrated ideas and concepts about what it was to do, behave and look like, that was understandable by the users in the manner intended.

SYSTEM

ADMIN

TENANT

**Database Design**

The general theme behind a database is to handle information as an integrated whole. A database is a collection of interrelated data stored with minimum redundancy to serve the users quickly and effectively. After designing input and output, the analyst must concentrate on database design or how data should be organized around user requirement. The general objective is to make information access easy, quick and flexible for other users.

**Tables used**

1. **Categories**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data type** | **Description** |
| Id | Int | Primary key |
| Name | varchar | - |

1. **Houses**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Description** |
| Id | Int | Primary key |
| House\_no | Varchar | - |
| Category\_id | Int | - |
| Descpription | Text | - |
| Price | double | - |

1. **Payments**

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Description** |
| Id | Int | Primary key |
| Tenant\_id | Int | - |
| Amount | Float | - |
| Invoice | Varchar | - |
| Date\_created | Datetime | - |

1. **Tenants**

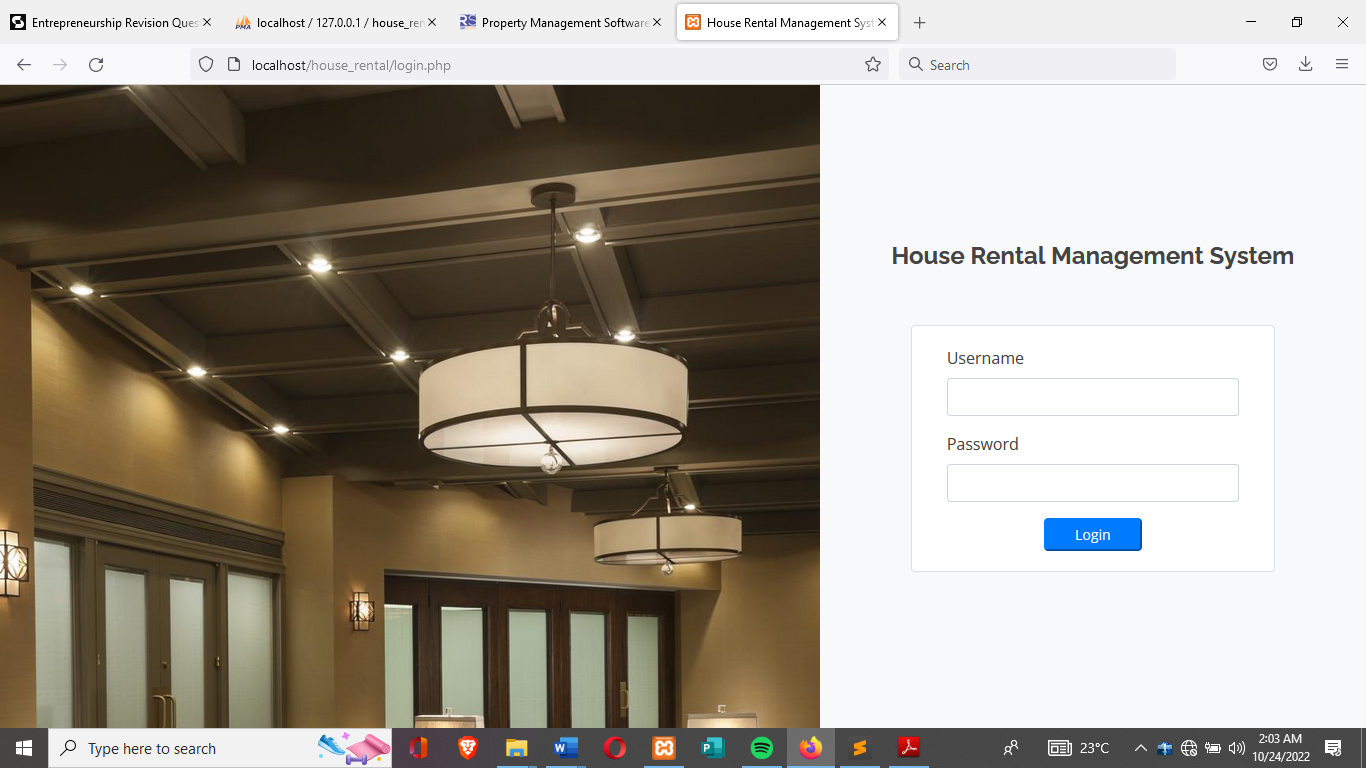
|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Description** |
| Id | Int | Primary key |
| Firstname | varchar | - |
| Middlename | Varchar | - |
| Lastname | Varchar | - |
| Email | Varchar | - |
| Contact | Varchar | - |
| House\_id | int | - |
| Status | Tinyint | - |
| Date\_in | date | - |

1. **Users**

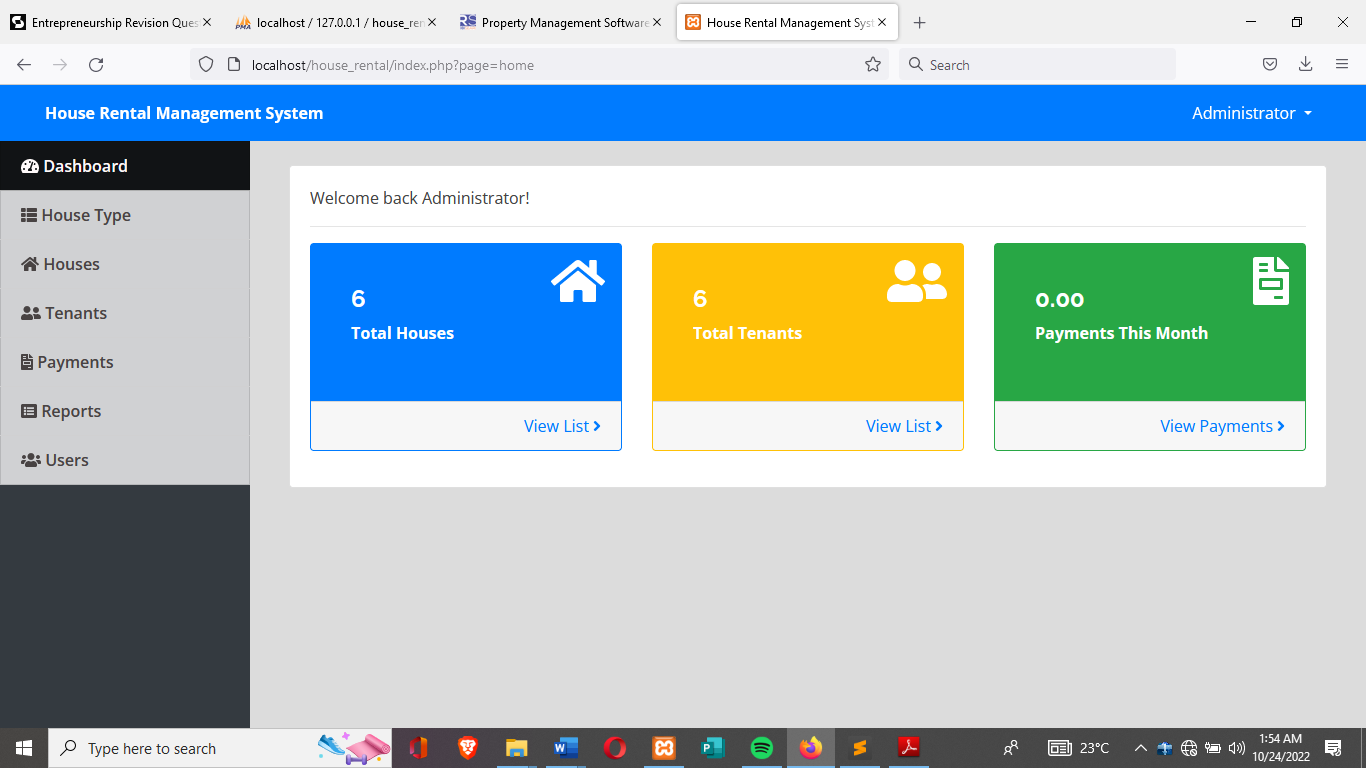
|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type** | **Description** |
| Id | Int | Primary key |
| Name | Text | - |
| Username | Varchar | - |
| Password | Text | - |
| type | tinyint | - |

**Software interface**

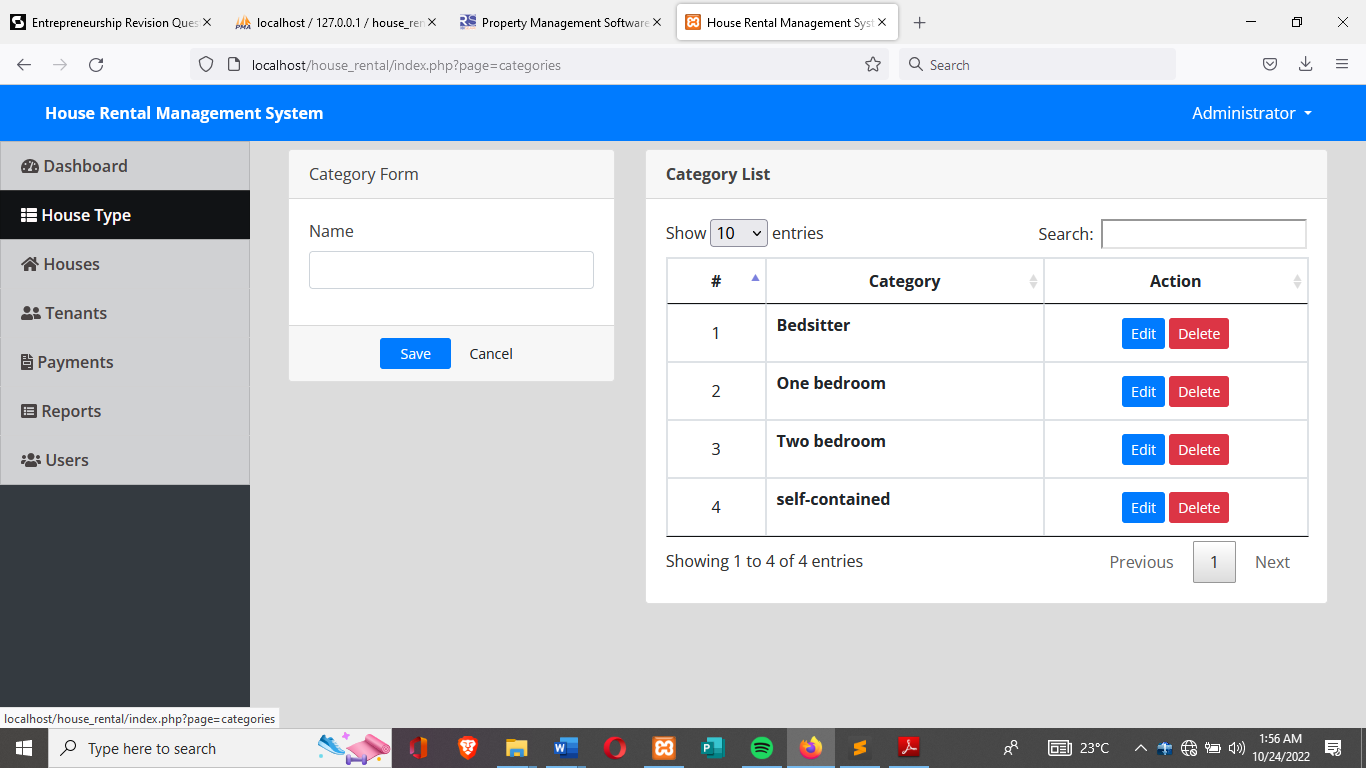
1. **Homepage/login menu**



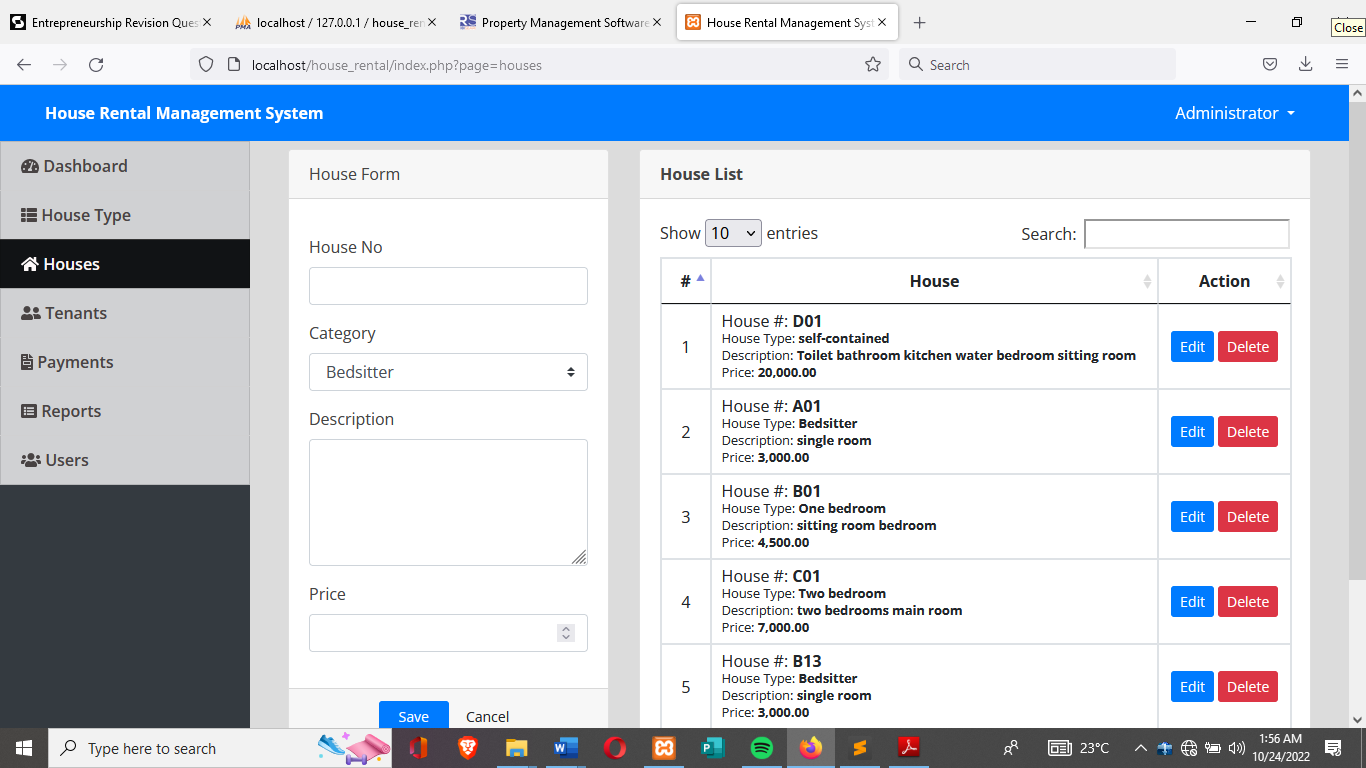
1. **Dashboard**



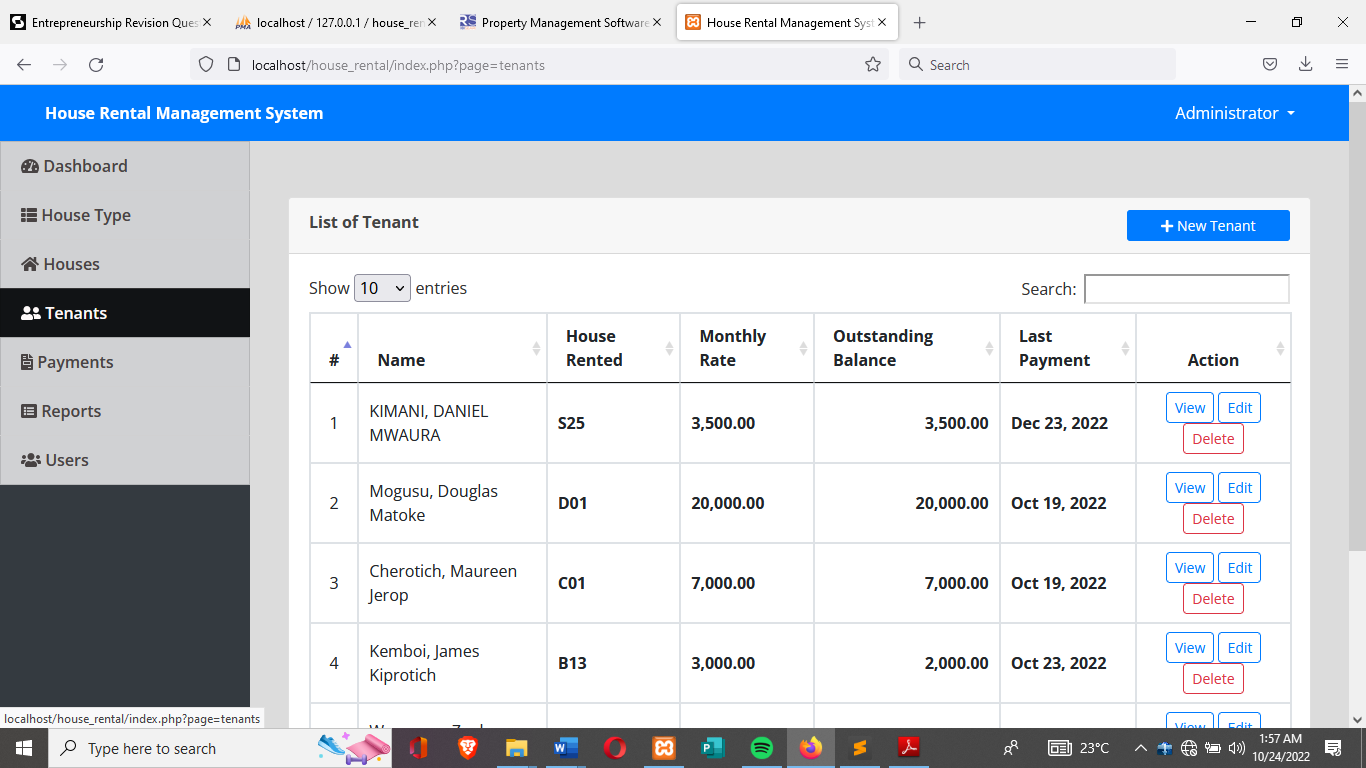
1. **House type**



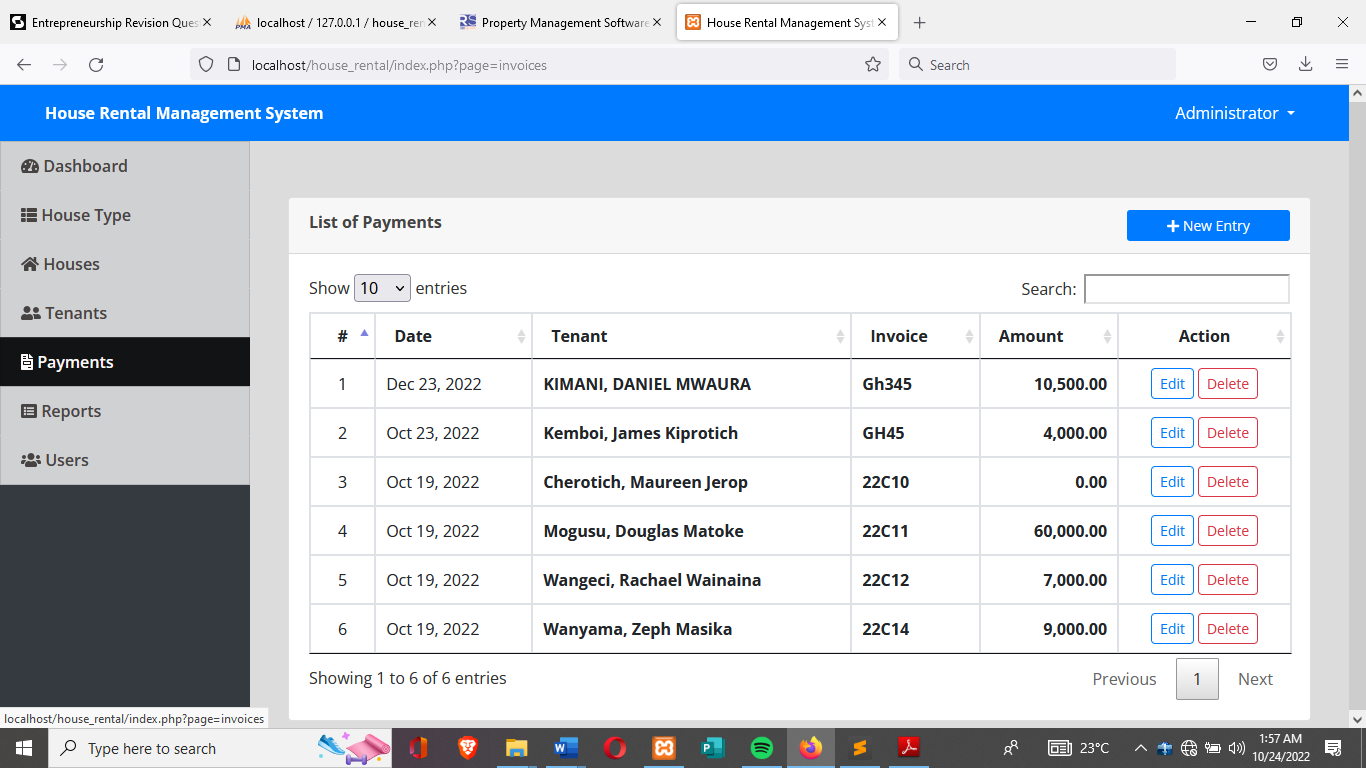
1. **Houses**



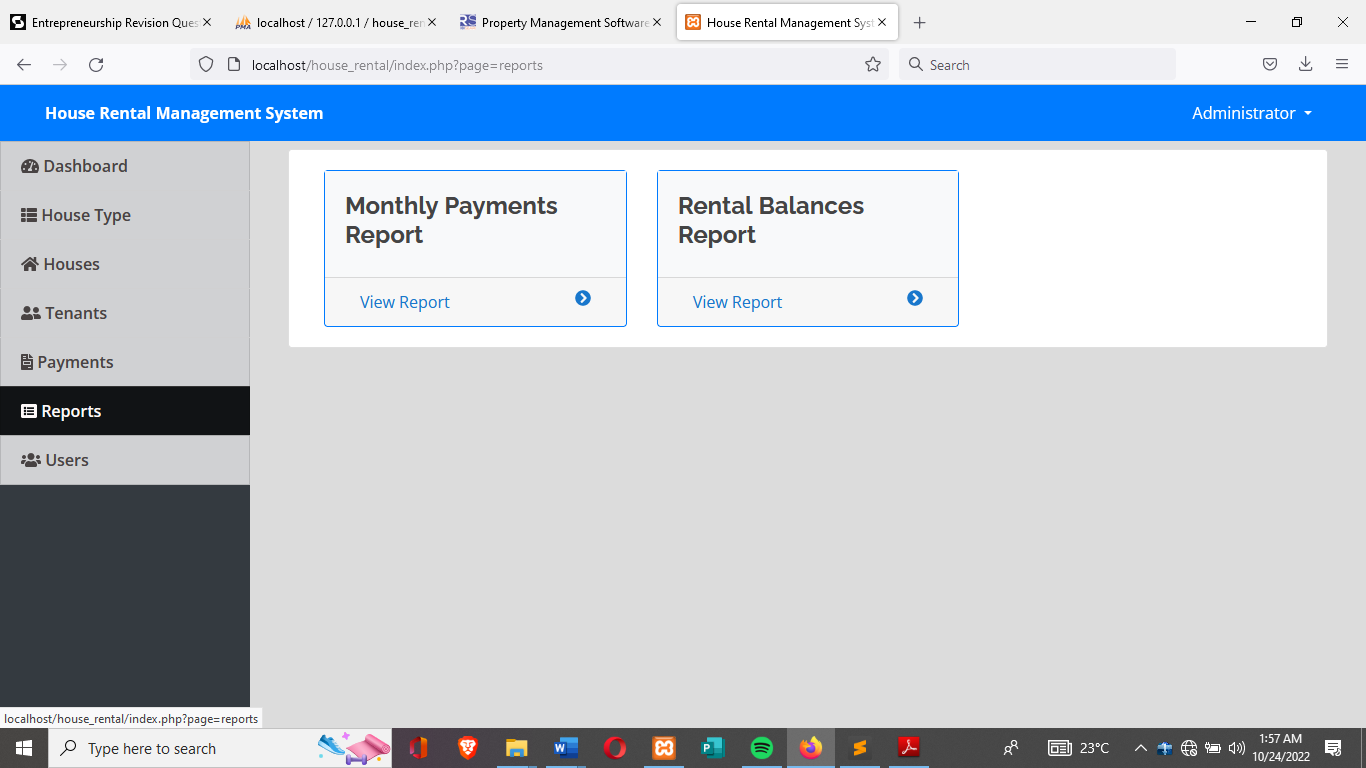
1. **Tenants**



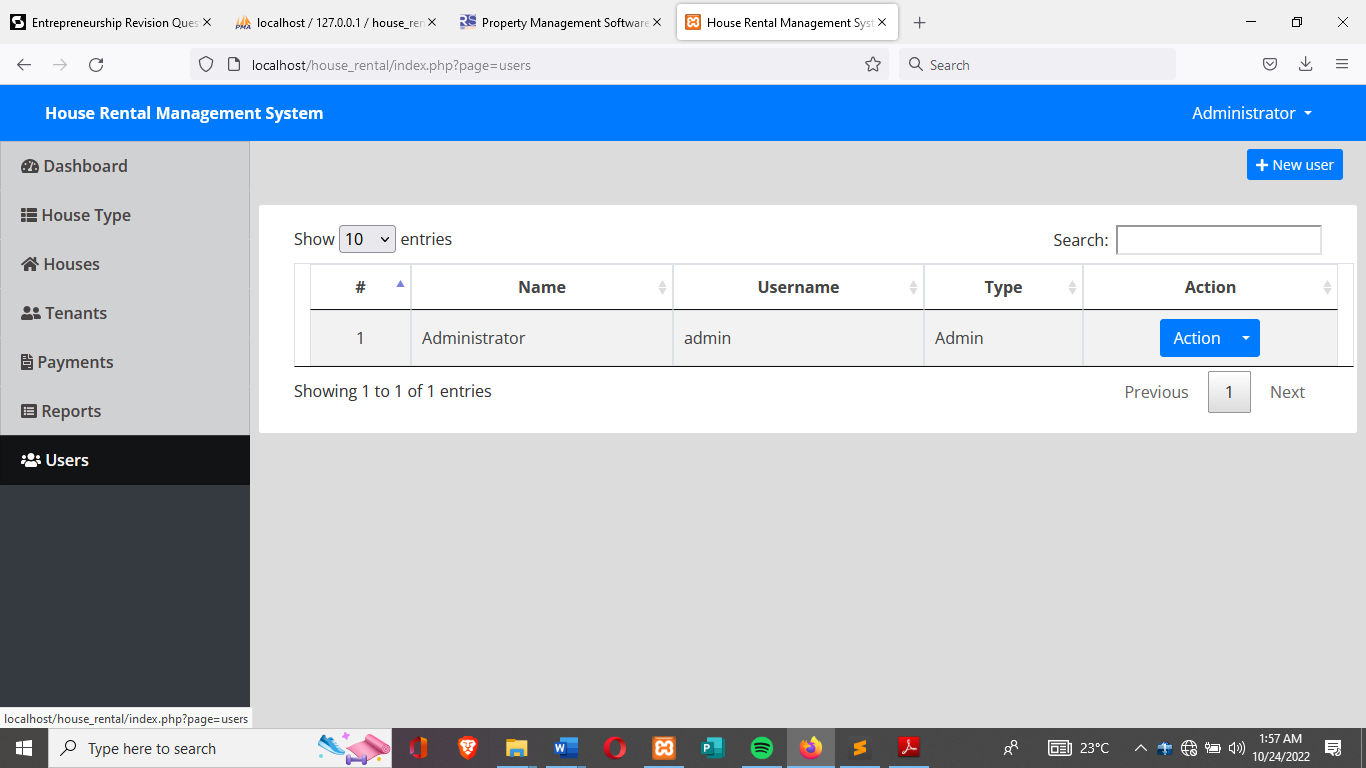
1. **Payments**



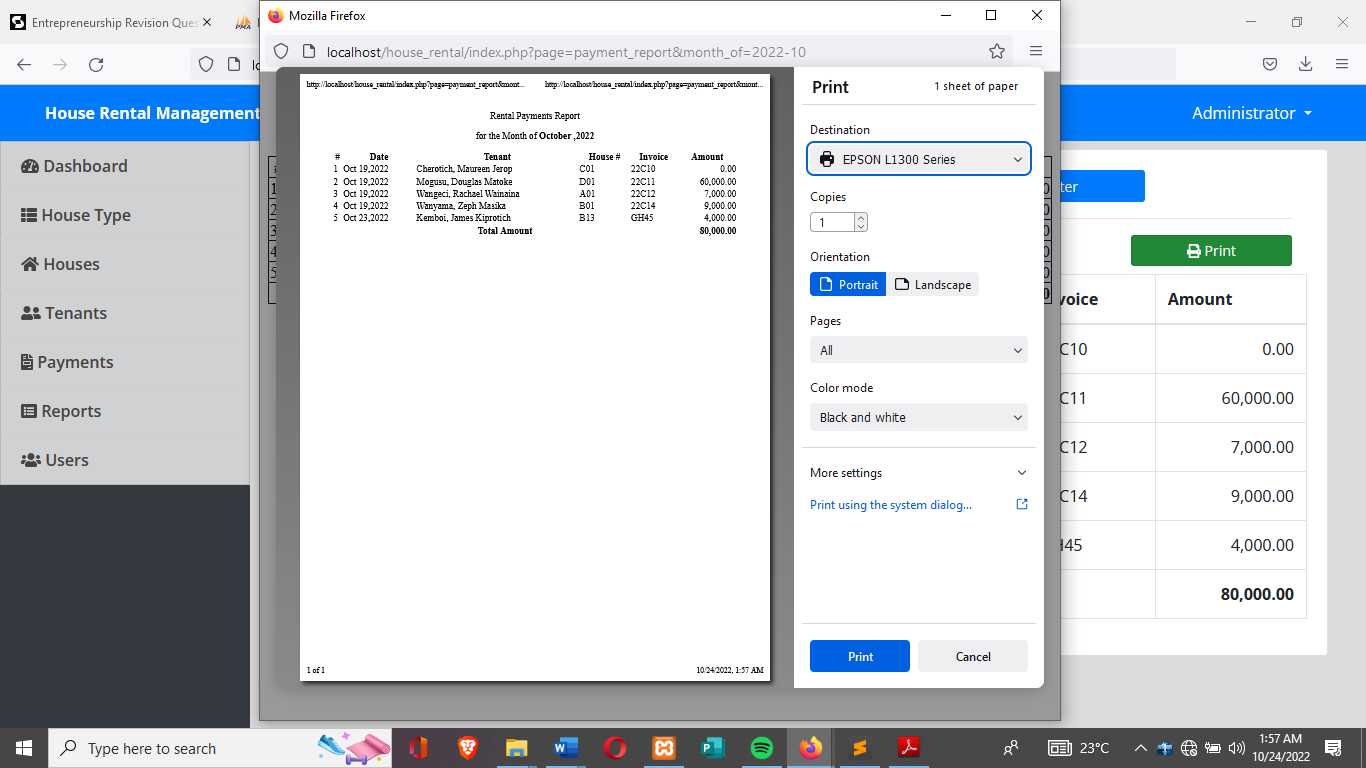
1. **Reports**



1. **Users**



1. **Print tenant payment report**



**CHAPTER SIX**

**SYSTEM IMPLIMENTATION**

Implementation is the stage in the project where the theoretical design is turned into a working system. The implementation phase constructs, installs and operates the new system. The most crucial stage in achieving a new successful system is that it will work efficiently and effectively.

There are several activities involved while implementing a new project. They are

* End user training
* End user Education
* Training on the application software
* Post implementation Review

**End user Training:**

The successful implementation of the new system will purely upon the involvement of the officers working in that department. The officers will be imparted the necessary training on the new technology

**End User Education:**

The education of the end user start after the implementation and testing is over. When the system is found to be more difficult to understand and complex, more effort is put to educate the end user to make them aware of the system, giving them lectures about the new system and providing them necessary documents and materials about how the system can do this.

**Training of application software:**

After providing the necessary basic training on the computer awareness, the users will have to be trained upon the new system such as the screen flows and screen design, type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the way to correct the data entered. It should then cover information needed by the specific user or group to use the system.

**Post Implementation review:**

The department is planning a method to know the states of the past implementation process. For that regular meeting will be arranged by the concerned officers about the implementation problem and success

**Software Testing**

Is it possible to invoke each menu function using logical assumptions that if all parts of the system are correct, the goal will be successfully achieved? Inadequate testing or non-testing will lead to errors that may appear few months later. That’s why it is very important to always test the new software. This creates two problems

The purpose of the system testing is to consider all the likely variations to which it will be suggested and push the systems to limits.

The testing process focuses on the logical intervals of the software ensuring that all statements have been tested and on functional interval is conducting tests to uncover errors and ensure that defined input will produce actual results that agree with the required results. Program level testing, modules level testing integrated and carried out. There are two major types of testing they are

* White Box Testing.
* Black Box Testing.

**White Box Testing.**

White box sometimes called “Glass box testing” is a test case design that uses the control structure of the procedural design to drive test case. Using white box testing methods, the following tests were made on the system

* All independent paths within a module have been exercised once. In our system, ensuring that case was selected and executed checked all case structures. The bugs that were prevailing in some part of the code where fixed
* All logical decisions were checked for the truth and falsity of the values.

**Black box Testing**

Black box testing focuses on the functional requirements of the software. This is black box testing enables the software engineering to derive a set of input conditions that will fully exercise all functional requirements for a program. Black box testing is not an alternative to white box testing rather it is complementary approach that is likely to uncover a different class of errors that white box methods like;

* Interface errors
* Performance in data structure
* Performance errors
* Initializing and termination errors

**CHAPTER SEVEN**

**RECOMMENDATIONS AND CONCLUSION**

**Recommendations**

My project is meant to satisfy the needs of rental house owners. Several user-friendly interfaces have also been adopted. This package shall prove to be a powerful in satisfying all the requirements of the users It is with utmost faith that I present this software to you hoping that it will solve the problems and encourage rental owners to continue appreciating technology because it is meant to change and ease all their work that seems to be very difficult. I don’t mean that my project is the best or that I have used the best technology available it just a simple and a humble venture that is easy to understand. However, I would encourage anyone who has the ability to advance it using advanced technologies so as to increase its capabilities.

**Conclusion**

In conclusion, the software can be used as an inventory system to provide a frame work that enables the mangers to make reasonable transactions made within a limited time frame. Each transaction made on the system go hand in hand with the data being updated in the database in my case it is MySQL which is the back end. Last but not least it is not the work that played the ways to success but ALMIGHTY GOD.

**APPENDIX**

**Login code**

<!DOCTYPE html>

<html lang="en">

<?php session\_start(); ?>

<head>

<meta charset="utf-8">

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

<title><?php echo isset($\_SESSION['system']['name']) ? $\_SESSION['system']['name'] : '' ?></title>

<?php

if(!isset($\_SESSION['login\_id']))

header('location:login.php');

include('./header.php');

// include('./auth.php');

?>

</head>

<style>

body{

background: #80808045;

}

.modal-dialog.large {

width: 80% !important;

max-width: unset;

}

.modal-dialog.mid-large {

width: 50% !important;

max-width: unset;

}

#viewer\_modal .btn-close {

position: absolute;

z-index: 999999;

/\*right: -4.5em;\*/

background: unset;

color: white;

border: unset;

font-size: 27px;

top: 0;

}

#viewer\_modal .modal-dialog {

width: 80%;

max-width: unset;

height: calc(90%);

max-height: unset;

}

#viewer\_modal .modal-content {

background: black;

border: unset;

height: calc(100%);

display: flex;

align-items: center;

justify-content: center;

}

#viewer\_modal img,#viewer\_modal video{

max-height: calc(100%);

max-width: calc(100%);

}

</style>

<body>

<?php include 'topbar.php' ?>

<?php include 'navbar.php' ?>

<div class="toast" id="alert\_toast" role="alert" aria-live="assertive" aria-atomic="true">

<div class="toast-body text-white">

</div>

</div>

<main id="view-panel" >

<?php $page = isset($\_GET['page']) ? $\_GET['page'] :'home'; ?>

<?php include $page.'.php' ?>

</main>

<div id="preloader"></div>

<a href="#" class="back-to-top"><i class="icofont-simple-up"></i></a>

<div class="modal fade" id="confirm\_modal" role='dialog'>

<div class="modal-dialog modal-md" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Confirmation</h5>

</div>

<div class="modal-body">

<div id="delete\_content"></div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-primary" id='confirm' onclick="">Continue</button>

<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

</div>

</div>

</div>

</div>

<div class="modal fade" id="uni\_modal" role='dialog'>

<div class="modal-dialog modal-md" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title"></h5>

</div>

<div class="modal-body">

</div>

<div class="modal-footer">

<button type="button" class="btn btn-primary" id='submit' onclick="$('#uni\_modal form').submit()">Save</button>

<button type="button" class="btn btn-secondary" data-dismiss="modal">Cancel</button>

</div>

</div>

</div>

</div>

<div class="modal fade" id="viewer\_modal" role='dialog'>

<div class="modal-dialog modal-md" role="document">

<div class="modal-content">

<button type="button" class="btn-close" data-dismiss="modal"><span class="fa fa-times"></span></button>

<img src="" alt="">

</div>

</div>

</div>

</body>

<script>

window.start\_load = function(){

$('body').prepend('<di id="preloader2"></di>')

}

window.end\_load = function(){

$('#preloader2').fadeOut('fast', function() {

$(this).remove();

})

}

window.viewer\_modal = function($src = ''){

start\_load()

var t = $src.split('.')

t = t[1]

if(t =='mp4'){

var view = $("<video src='"+$src+"' controls autoplay></video>")

}else{

var view = $("<img src='"+$src+"' />")

}

$('#viewer\_modal .modal-content video,#viewer\_modal .modal-content img').remove()

$('#viewer\_modal .modal-content').append(view)

$('#viewer\_modal').modal({

show:true,

backdrop:'static',

keyboard:false,

focus:true

})

end\_load()

}

window.uni\_modal = function($title = '' , $url='',$size=""){

start\_load()

$.ajax({

url:$url,

error:err=>{

console.log()

alert("An error occured")

},

success:function(resp){

if(resp){

$('#uni\_modal .modal-title').html($title)

$('#uni\_modal .modal-body').html(resp)

if($size != ''){

$('#uni\_modal .modal-dialog').addClass($size)

}else{

$('#uni\_modal .modal-dialog').removeAttr("class").addClass("modal-dialog modal-md")

}

$('#uni\_modal').modal({

show:true,

backdrop:'static',

keyboard:false,

focus:true

})

end\_load()

}

}

})

}

window.\_conf = function($msg='',$func='',$params = []){

$('#confirm\_modal #confirm').attr('onclick',$func+"("+$params.join(',')+")")

$('#confirm\_modal .modal-body').html($msg)

$('#confirm\_modal').modal('show')

}

window.alert\_toast= function($msg = 'TEST',$bg = 'success'){

$('#alert\_toast').removeClass('bg-success')

$('#alert\_toast').removeClass('bg-danger')

$('#alert\_toast').removeClass('bg-info')

$('#alert\_toast').removeClass('bg-warning')

if($bg == 'success')

$('#alert\_toast').addClass('bg-success')

if($bg == 'danger')

$('#alert\_toast').addClass('bg-danger')

if($bg == 'info')

$('#alert\_toast').addClass('bg-info')

if($bg == 'warning')

$('#alert\_toast').addClass('bg-warning')

$('#alert\_toast .toast-body').html($msg)

$('#alert\_toast').toast({delay:3000}).toast('show');

}

$(document).ready(function(){

$('#preloader').fadeOut('fast', function() {

$(this).remove();

})

})

$('.datetimepicker').datetimepicker({

format:'Y/m/d H:i',

startDate: '+3d'

})

$('.select2').select2({

placeholder:"Please select here",

width: "100%"

})

</script>

</html>

**Tenants**

<?php include('db\_connect.php');?>

<div class="container-fluid">

<div class="col-lg-12">

<div class="row mb-4 mt-4">

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

</div>

</div>

<div class="row">

<!-- FORM Panel -->

<!-- Table Panel -->

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

<div class="card">

<div class="card-header">

<b>List of Tenant</b>

<span class="float:right"><a class="btn btn-primary btn-block btn-sm col-sm-2 float-right" href="javascript:void(0)" id="new\_tenant">

<i class="fa fa-plus"></i> New Tenant

</a></span>

</div>

<div class="card-body">

<table class="table table-condensed table-bordered table-hover">

<thead>

<tr>

<th class="text-center">#</th>

<th class="">Name</th>

<th class="">House Rented</th>

<th class="">Monthly Rate</th>

<th class="">Outstanding Balance</th>

<th class="">Last Payment</th>

<th class="text-center">Action</th>

</tr>

</thead>

<tbody>

<?php

$i = 1;

$tenant = $conn->query("SELECT t.\*,concat(t.lastname,', ',t.firstname,' ',t.middlename) as name,h.house\_no,h.price FROM tenants t inner join houses h on h.id = t.house\_id where t.status = 1 order by h.house\_no desc ");

while($row=$tenant->fetch\_assoc()):

$months = abs(strtotime(date('Y-m-d')." 23:59:59") - strtotime($row['date\_in']." 23:59:59"));

$months = floor(($months) / (30\*60\*60\*24) + 1);

$payable = $row['price'] \* $months;

$paid = $conn->query("SELECT SUM(amount) as paid FROM payments where tenant\_id =".$row['id']);

$last\_payment = $conn->query("SELECT \* FROM payments where tenant\_id =".$row['id']." order by unix\_timestamp(date\_created) desc limit 1");

$paid = $paid->num\_rows > 0 ? $paid->fetch\_array()['paid'] : 0;

$last\_payment = $last\_payment->num\_rows > 0 ? date("M d, Y",strtotime($last\_payment->fetch\_array()['date\_created'])) : 'N/A';

$outstanding = $payable - $paid;

?>

<tr>

<td class="text-center"><?php echo $i++ ?></td>

<td>

<?php echo ucwords($row['name']) ?>

</td>

<td class="">

<p> <b><?php echo $row['house\_no'] ?></b></p>

</td>

<td class="">

<p> <b><?php echo number\_format($row['price'],2) ?></b></p>

</td>

<td class="text-right">

<p> <b><?php echo number\_format($outstanding,2) ?></b></p>

</td>

<td class="">

<p><b><?php echo $last\_payment ?></b></p>

</td>

<td class="text-center">

<button class="btn btn-sm btn-outline-primary view\_payment" type="button" data-id="<?php echo $row['id'] ?>" >View</button>

<button class="btn btn-sm btn-outline-primary edit\_tenant" type="button" data-id="<?php echo $row['id'] ?>" >Edit</button>

<button class="btn btn-sm btn-outline-danger delete\_tenant" type="button" data-id="<?php echo $row['id'] ?>">Delete</button>

</td>

</tr>

<?php endwhile; ?>

</tbody>

</table>

</div>

</div>

</div>

<!-- Table Panel -->

</div>

</div>

</div>

<style>

td{

vertical-align: middle !important;

}

td p{

margin: unset

}

img{

max-width:100px;

max-height: :150px;

}

</style>

<script>

$(document).ready(function(){

$('table').dataTable()

})

$('#new\_tenant').click(function(){

uni\_modal("New Tenant","manage\_tenant.php","mid-large")

})

$('.view\_payment').click(function(){

uni\_modal("Tenants Payments","view\_payment.php?id="+$(this).attr('data-id'),"large")

})

$('.edit\_tenant').click(function(){

uni\_modal("Manage Tenant Details","manage\_tenant.php?id="+$(this).attr('data-id'),"mid-large")

})

$('.delete\_tenant').click(function(){

\_conf("Are you sure to delete this Tenant?","delete\_tenant",[$(this).attr('data-id')])

})

function delete\_tenant($id){

start\_load()

$.ajax({

url:'ajax.php?action=delete\_tenant',

method:'POST',

data:{id:$id},

success:function(resp){

if(resp==1){

alert\_toast("Data successfully deleted",'success')

setTimeout(function(){

location.reload()

},1500)

}

}

})

}

</script>

**Payments**

<?php include('db\_connect.php');?>

<div class="container-fluid">

<div class="col-lg-12">

<div class="row mb-4 mt-4">

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

</div>

</div>

<div class="row">

<!-- FORM Panel -->

<!-- Table Panel -->

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

<div class="card">

<div class="card-header">

<b>List of Payments</b>

<span class="float:right"><a class="btn btn-primary btn-block btn-sm col-sm-2 float-right" href="javascript:void(0)" id="new\_payment">

<i class="fa fa-plus"></i> New Entry

</a></span>

</div>

<div class="card-body">

<table class="table table-condensed table-bordered table-hover">

<thead>

<tr>

<th class="text-center">#</th>

<th class="">Tenant</th>

<th class="">House #</th>

<th class="">Outstanding Balance</th>

<th class="">Last Payment</th>

<th class="text-center">Action</th>

</tr>

</thead>

<tbody>

<?php

$i = 1;

$tenants =$conn->query("SELECT t.\*,concat(t.lastname,', ',t.firstname,' ',t.middlename) as name,h.house\_no,h.price FROM tenants t inner join houses h on h.id = t.house\_id where t.status = 1 order by h.house\_no desc ");

while($row=$tenants->fetch\_assoc()):

$months = abs(strtotime(date('Y-m-d')." 23:59:59") - strtotime($row['date\_in']." 23:59:59"));

$months = floor(($months) / (30\*60\*60\*24));

$payable = $row['price'] \* $months;

$paid = $conn->query("SELECT SUM(amount) as paid FROM payments where tenant\_id =".$row['id']);

$last\_payment = $conn->query("SELECT \* FROM payments where tenant\_id =".$row['id']." order by unix\_timestamp(date\_created) desc limit 1");

$paid = $paid->num\_rows > 0 ? $paid->fetch\_array()['paid'] : 0;

$last\_payment = $last\_payment->num\_rows > 0 ? date("M d, Y",strtotime($last\_payment->fetch\_array()['date\_created'])) : 'N/A';

$outstanding = $payable - $paid;

?>

<tr>

<td class="text-center"><?php echo $i++ ?></td>

<td class="">

<p> <b><?php echo ucwords($row['name']) ?></b></p>

</td>

<td class="">

<p> <b><?php echo $row['house\_no'] ?></b></p>

</td>

<td class="text-right">

<p> <b><?php echo number\_format($outstanding,2) ?></b></p>

</td>

<td class="">

<p><b><?php echo $last\_payment ?></b></p>

</td>

<td class="text-center">

<button class="btn btn-sm btn-outline-primary view\_payment" type="button" data-id="<?php echo $row['id'] ?>" >View</button>

</td>

</tr>

<?php endwhile; ?>

</tbody>

</table>

</div>

</div>

</div>

<!-- Table Panel -->

</div>

</div>

</div>

<style>

td{

vertical-align: middle !important;

}

td p{

margin: unset

}

img{

max-width:100px;

max-height: :150px;

}

</style>

<script>

$(document).ready(function(){

$('table').dataTable()

})

$('#new\_payment').click(function(){

uni\_modal("New payment","manage\_payment.php","mid-large")

})

$('.edit\_payment').click(function(){

uni\_modal("Manage payment Details","manage\_payment.php?id="+$(this).attr('data-id'),"mid-large")

})

$('.view\_payment').click(function(){

uni\_modal("Tenants Payments","view\_payment.php?id="+$(this).attr('data-id'),"mid-large")

})

$('.delete\_payment').click(function(){

\_conf("Are you sure to delete this payment?","delete\_payment",[$(this).attr('data-id')])

})

function delete\_payment($id){

start\_load()

$.ajax({

url:'ajax.php?action=delete\_payment',

method:'POST',

data:{id:$id},

success:function(resp){

if(resp==1){

alert\_toast("Data successfully deleted",'success')

setTimeout(function(){

location.reload()

},1500)

}

}

})

}

</script>

**Houses**

<?php include('db\_connect.php');?>

<div class="container-fluid">

<div class="col-lg-12">

<div class="row">

<!-- FORM Panel -->

<div class="col-md-4">

<form action="" id="manage-house">

<div class="card">

<div class="card-header">

House Form

</div>

<div class="card-body">

<div class="form-group" id="msg"></div>

<input type="hidden" name="id">

<div class="form-group">

<label class="control-label">House No</label>

<input type="text" class="form-control" name="house\_no" required="">

</div>

<div class="form-group">

<label class="control-label">Category</label>

<select name="category\_id" id="" class="custom-select" required>

<?php

$categories = $conn->query("SELECT \* FROM categories order by name asc");

if($categories->num\_rows > 0):

while($row= $categories->fetch\_assoc()) :

?>

<option value="<?php echo $row['id'] ?>"><?php echo $row['name'] ?></option>

<?php endwhile; ?>

<?php else: ?>

<option selected="" value="" disabled="">Please check the category list.</option>

<?php endif; ?>

</select>

</div>

<div class="form-group">

<label for="" class="control-label">Description</label>

<textarea name="description" id="" cols="30" rows="4" class="form-control" required></textarea>

</div>

<div class="form-group">

<label class="control-label">Price</label>

<input type="number" class="form-control text-right" name="price" step="any" required="">

</div>

</div>

<div class="card-footer">

<div class="row">

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

<button class="btn btn-sm btn-primary col-sm-3 offset-md-3"> Save</button>

<button class="btn btn-sm btn-default col-sm-3" type="reset" > Cancel</button>

</div>

</div>

</div>

</div>

</form>

</div>

<!-- FORM Panel -->

<!-- Table Panel -->

<div class="col-md-8">

<div class="card">

<div class="card-header">

<b>House List</b>

</div>

<div class="card-body">

<table class="table table-bordered table-hover">

<thead>

<tr>

<th class="text-center">#</th>

<th class="text-center">House</th>

<th class="text-center">Action</th>

</tr>

</thead>

<tbody>

<?php

$i = 1;

$house = $conn->query("SELECT h.\*,c.name as cname FROM houses h inner join categories c on c.id = h.category\_id order by id asc");

while($row=$house->fetch\_assoc()):

?>

<tr>

<td class="text-center"><?php echo $i++ ?></td>

<td class="">

<p>House #: <b><?php echo $row['house\_no'] ?></b></p>

<p><small>House Type: <b><?php echo $row['cname'] ?></b></small></p>

<p><small>Description: <b><?php echo $row['description'] ?></b></small></p>

<p><small>Price: <b><?php echo number\_format($row['price'],2) ?></b></small></p>

</td>

<td class="text-center">

<button class="btn btn-sm btn-primary edit\_house" type="button" data-id="<?php echo $row['id'] ?>" data-house\_no="<?php echo $row['house\_no'] ?>" data-description="<?php echo $row['description'] ?>" data-category\_id="<?php echo $row['category\_id'] ?>" data-price="<?php echo $row['price'] ?>" >Edit</button>

<button class="btn btn-sm btn-danger delete\_house" type="button" data-id="<?php echo $row['id'] ?>">Delete</button>

</td>

</tr>

<?php endwhile; ?>

</tbody>

</table>

</div>

</div>

</div>

<!-- Table Panel -->

</div>

</div>

</div>

<style>

td{

vertical-align: middle !important;

}

td p {

margin: unset;

padding: unset;

line-height: 1em;

}

</style>

<script>

$('#manage-house').on('reset',function(e){

$('#msg').html('')

})

$('#manage-house').submit(function(e){

e.preventDefault()

start\_load()

$('#msg').html('')

$.ajax({

url:'ajax.php?action=save\_house',

data: new FormData($(this)[0]),

cache: false,

contentType: false,

processData: false,

method: 'POST',

type: 'POST',

success:function(resp){

if(resp==1){

alert\_toast("Data successfully saved",'success')

setTimeout(function(){

location.reload()

},1500)

}

else if(resp==2){

$('#msg').html('<div class="alert alert-danger">House number already exist.</div>')

end\_load()

}

}

})

})

$('.edit\_house').click(function(){

start\_load()

var cat = $('#manage-house')

cat.get(0).reset()

cat.find("[name='id']").val($(this).attr('data-id'))

cat.find("[name='house\_no']").val($(this).attr('data-house\_no'))

cat.find("[name='description']").val($(this).attr('data-description'))

cat.find("[name='price']").val($(this).attr('data-price'))

cat.find("[name='category\_id']").val($(this).attr('data-category\_id'))

end\_load()

})

$('.delete\_house').click(function(){

\_conf("Are you sure to delete this house?","delete\_house",[$(this).attr('data-id')])

})

function delete\_house($id){

start\_load()

$.ajax({

url:'ajax.php?action=delete\_house',

method:'POST',

data:{id:$id},

success:function(resp){

if(resp==1){

alert\_toast("Data successfully deleted",'success')

setTimeout(function(){

location.reload()

},1500)

}

}

})

}

$('table').dataTable()

</script>