

## 1. SYNOPSIS

### INTRODUCTION

The PG Room Management System is a web-based application designed to streamline the process of finding and managing paying guest (PG) rooms. This system aims to provide a convenient platform for PG room owners to showcase their properties and for users to find suitable accommodations. The system allows owners to add, edit, and manage their PG room details, while users can search and apply for rooms.

The application enables owners to receive and manage applications from users, simplifying the process of PG room management and making it more efficient and user-friendly. The system provides a secure and scalable solution for PG room management, catering to the growing demand for PG rooms and facilitating efficient communication between owners and users.

### Objectives:

- Develop a user-friendly web application for managing PG rooms.
- Provide a platform for PG room owners to showcase their properties.
- Enable users to search and apply for PG rooms efficiently.
- Streamline the application process for both owners and users.

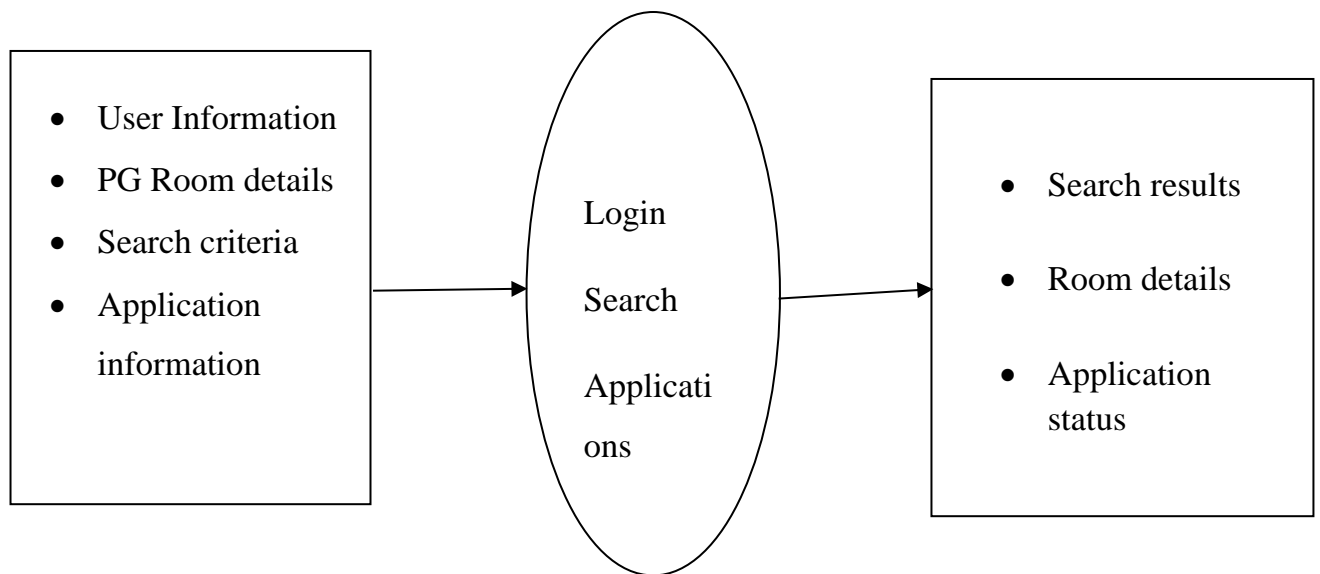
### INPUT OF THE PROJECT:

1. User registration information
2. PG room details
3. Search criteria
4. Application information

### OUTPUT OF THE PROJECT:

1. Search results
2. Room details
3. Application status

### PROCESS LOGIC



### Tools/Platforms, languages to be used:

#### Hardware requirements:

- Processor: Core i3 intel processor
- RAM : 2GB and above
- Hard Disk : 20GB
- Operating System: Windows 7 and above

## PG ROOMS MANAGEMENT

### **Software requirement:**

- Front End: HTML, CSS
- Middle ware: PHP
- Back End: MYSQL

### **Are you doing this Project for any industry/Client? If yes acceptance of the Project:**

- No

### **Duration of the Project:**

- 2 Months

### **Member of the Project:**

Nouseed Shaikh

### **Limitations of the Project:**

- Dependence on Internet connectivity.
- Dependence on user feedback and ratings.
- Limited integration with external services.

### **FUTURE AND SCOPE OF THE PROJECT:**

- Development of Android and IOS application.
- Integration with payment gateways.
- Advanced search filters.
- Rating and Review system.

## 2. INTRODUCTION

Implementation is the process of converting a new revised system design into operation. The objective is to put the new revised system, which has been tested into operation while holding costs, risks and personal irritation to the minimum. A critical aspect of the implementation process is to ensure that there will be no description in the function of the organization.

### **Introduction to technologies used in this project:**

Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy and it is a process of having the systems personnel check out and put new products into use and construct any files of data needed to use it.

### **Why you need XAMP, MySql and PHP?**

PHP is a powerful scripting language that can be run by itself in the command line of any computer with PHP installed. However PHP alone is not enough in order to build dynamic web sites. To use PHP on a web site, you need a server that can process PHP scripts. WAMP server allows developers to test PHP scripts locally. Additionally dynamic websites are dependent on stored information that can be accessed easily; this is the main difference between a dynamic site and a static HTML site. However PHP does not provide a simple, efficient way to store data. This is where a relational database management system like MySql comes into play.

### **PHP:**

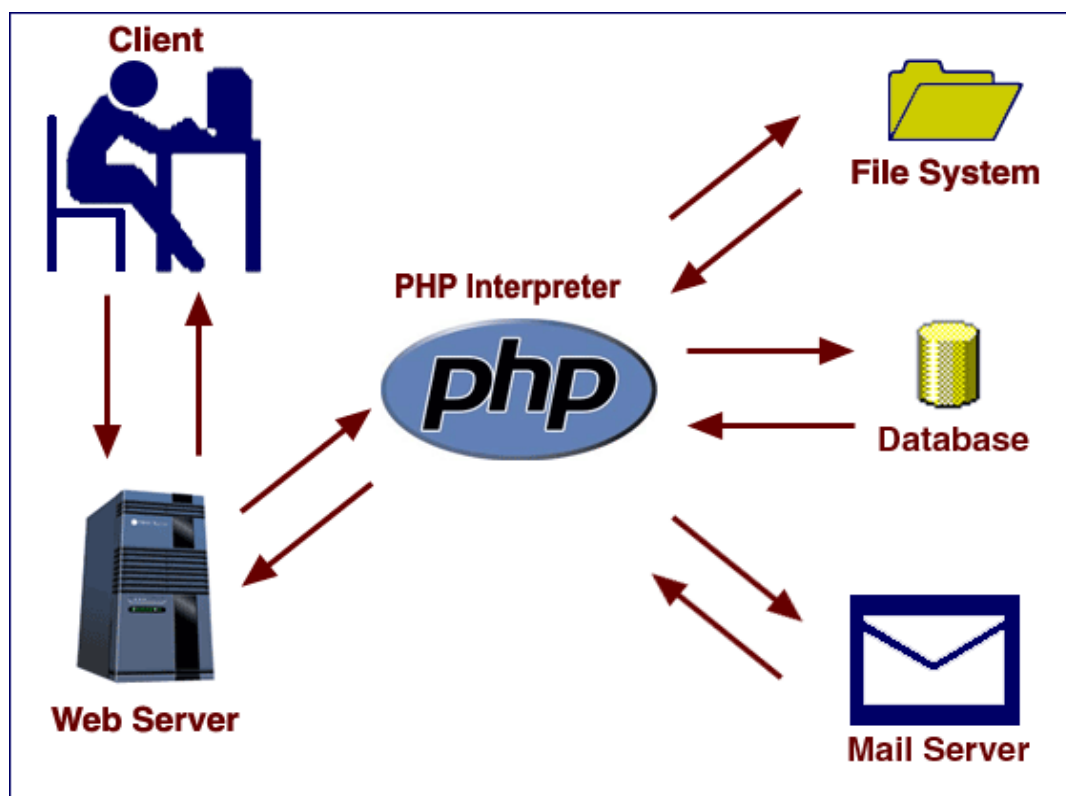
PHP is a scripting language originally designed for producing dynamic webpages. It has evolved to include a command line interface capability and can be used in standalone graphical application.

## PG ROOMS MANAGEMENT

While PHP was originally created by RasmusLerdorf in 1995, the main implementation of PHP is now produced by the PHP Groups and serves as the de facto standard for PHP as there is no formal specification.

PHP is a scripting language under the PHP License; however, it is incompatible with the GNU General Public License (GPL). Due to restrictions on the usage of the term PHP. It is widely used general-purpose scripting language that is especially suited for web development and can be embedded into HTML. It generally runs on a web server, taking PHP code as its input, I am creating web pages as out puts. It can be deployed on web servers and on almost every operating system and platform free of charge. PHP in installed on more the twenty million web sites and one million web servers.

### PHP Architecture:



### Usage:

PHP primarily acts as a filter, taking input from a file or stream containing text and/or PHP instructions and outputs another stream of data; most commonly the

## PG ROOMS MANAGEMENT

output will be HTML. It can automatically detect the language of the user. From PHP 4, the PHP parser compiles input to produce byte code for processing by the Zend Engine, giving improved performance over its interpreter predecessor. Originally designed to create dynamic web pages, PHP's principal focus is server-side scripting, and it is similar to other server-side scripting languages that provide dynamic content from a web server to a client, such as Microsoft's Active Server Pages, Sun Microsystems JavaServer Pages, and mod\_perl. PHP has also attracted the development of many frameworks that provide building blocks and a design structure to promote rapid application development (RAD). Some of these include CakePHP, Symfony, CodeIgniter, and Zend Framework, offering features similar to other web application frameworks.

The XAMP architecture has become popular in the web industry as a way of deploying web applications. PHP is commonly used as the PHP in this bundle alongside Linux, Apache and MySQL, although they may also refer to Python or Perl. As of April 2007, over 20 million Internet domains were hosted on servers with PHP installed, and PHP was recorded as the most popular Apache module. Significant websites are written in PHP including the user-facing portion of Facebook, Wikipedia (MediaWiki), Yahoo!, MyYearbook, Wordpress.

In addition to server-side scripting, PHP can be used to create stand-alone, compiled applications and libraries, it can be used for shell scripting.

### **Dreamweaver:**

A website authoring program originally developed by Macromedia. It does many things including allowing designer to move back and forth between visual and HTML modes

### **HTML:**

HTML means Hypertext Markup Language. HTML is a method of describing the format of document, which allows them to be viewed on computer screen. Web browsers display HTML documents, program which can navigate across networks and display a wide variety of types of information. HTML pages can be developed to be

## PG ROOMS MANAGEMENT

simple text or to be complex multimedia extra advantages containing, moving images, virtual reality, and java applets. The global publishing format of the Internet is HTML. It allows authors to use not only text but also format that text with headings, list and tables, and also includes still images videos, and sound within text. Readers can access pages information from any where in the world at the click of mouse button information can be downloaded to readers own PC or workstations HTML pages can also be used for entering a data and as a front end for commercial transaction.

Example to display message using HTML tags

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

<title>Untitled Document</title>

</head>

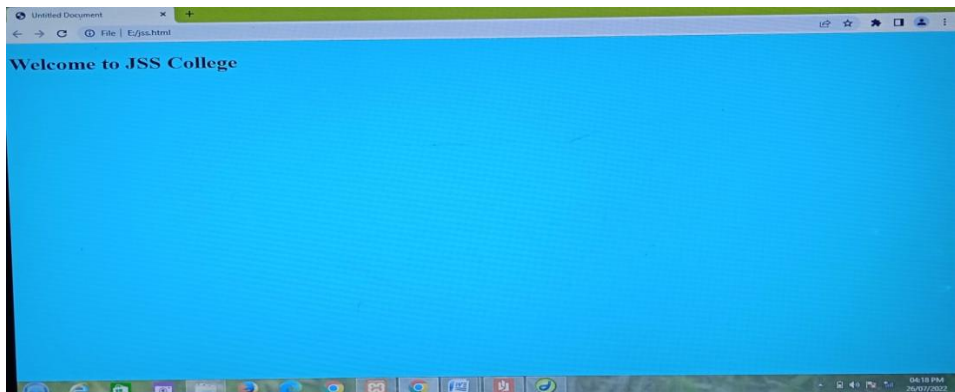
<body>

<h1>Welcome to JSS College</h1>

</body>

</html>
```

**Output:**



### MY SQL:

#### INTRODUCTION TO SQL SERVER

SQL Server is a Relational Database Management System (RDBMS) that runs exclusively under the Windows operating system. One benefit of using Windows exclusively is that you can send and receive E-mail messages based on SQL Server "events" and you can also let the operating system handle login security. The data base is an organized collection of data. A database management system (DBMS) such as Access, FileMaker Pro, Oracle or SQL Server provides you with the software tools you need to organize that data in a flexible manner. It includes facilities to add, modify or delete data from the database, ask questions (or queries) about the data stored in the database and produce reports summarizing selected contents.

MySql is a multithreaded, multi-user SQL database management system (DBMS). The basic program runs as a server providing multi-user access to a number of databases. Originally financed in a similar fashion to the JBoss model, MySql was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB now a subsidiary of Sun Micro system , which holds the copyright to most of the codebase.

#### Database Evolution:

SQL was invented back in the 1960's by E.F. Cod of IBM. In order to increase data integrity and reduce repetitive data. RDBMS systems didn't appear until the late 70's when Sybase and Oracle introduced systems. These systems existed on mainframes at the time.

ANSI-SQL came to be in the 1980's. This was important because it meant that disparate systems could communicate through an agreed set of standards. There are different levels of ANSI-SQL compliance. Almost every major RDBMS today is entry level compliant, including SQL Server 2000. Every RDBMS has its own flavour of SQL that complements ANSI-SQL with proprietary elements. SQL Server's flavour of SQL is known as Transact SQL (T-SQL).



## PG ROOMS MANAGEMENT

SQL Server was originally a Sybase product. Microsoft bought the product outright from Sybase and by version 7.0, the version prior to 2000, all the code had been rewritten by Microsoft's programming gurus

### Features of sql:

- It is simple English like language and uses simple commands such as SELECT, CREATE, DROP etc.
- It is not having condition loops, variables and most of the commands are single line commands.
- as PL/SQL (Procedural language of sql).
- One of the key features of sql server is the XML support. XML has Grown to be standard technology for organizations that share data on the web.
- Now with sql server 2000 XML documents can be retrieved directly from the database and it provides various ways to retrieve data in XML format.
- The entire SQL has been divided into 4 major categories.
  1. Data Manipulation Language.
  2. Data Definition Language.
  3. Transaction Control language.
  4. Data Control Language

### Security:

View are basically used as a part of security, means in many organizations, the end user will never be given original tables & all data entry will be done with the help of views only. But the data base administrator will be able to see everything because all the operations done by the different users will come to the same table.

### Queries:

A query is a question or a request. With MySql, we can query a database for specific information and have a record set returned. Create a connection to a database:

## PG ROOMS MANAGEMENT

Before you can access data in a database, you must create a connection to the database. In PHP, this is done with the `mysql_connect()` function.

Syntax:

```
mysql_connect (server name, username, password);
```

Server name: Optional Specifies the Server to connect .Default values is localhost: 3306

### **MySql Functions:**

What is a database? Quite simply, it's an organized collection of data. A database management system (DBMS) such as Access, FileMaker Pro, Oracle or SQL Server provides you with the software tools you need to organize that data in a flexible manner. It includes facilities to add, modify or delete data from the database, ask questions (or queries) about the data stored in the database and produce reports summarizing selected contents. MySql is a multithreaded, multi-user SQL database management system (DBMS). The basic program runs as a server providing multi-user access to a number of databases. Originally financed in a similar fashion to the JBoss model, MySql was owned and sponsored by a single for-profit firm, the Swedish company MySQLAB now a subsidiary of Sun Micro system , which holds the copyright to most of the codebase. The project's source code is available under terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySql is a database. The data in MySql is stored in database objects called tables. A table is a collection of related data entries and it consists of columns and rows. Databases are useful when storing information categorically.

### **Create a connection to a database:**

Before you can access data in a database, you must create a connection to the database. In PHP, this is done with the `mysql_connect()` function.

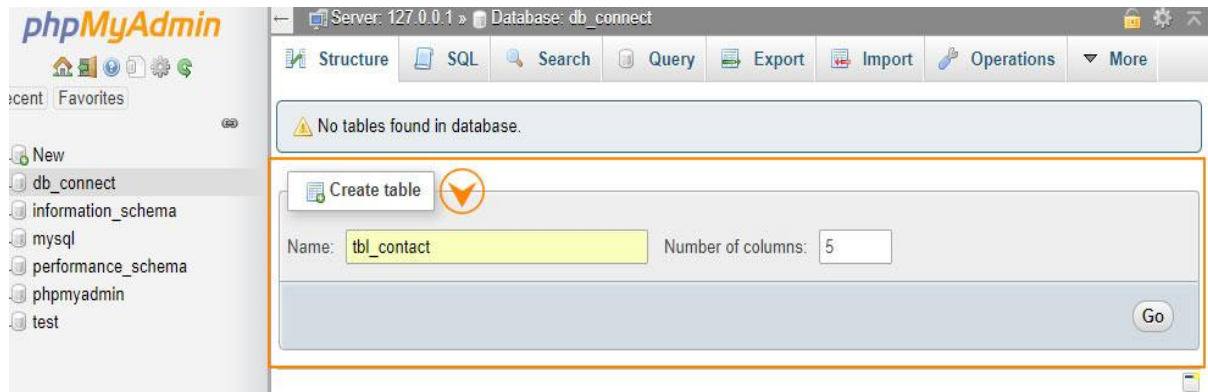
Syntax: `Mysql connect (server name, username, password);`

Server name: Optional Specifies the Server to connect .Default values is localhost: 3306

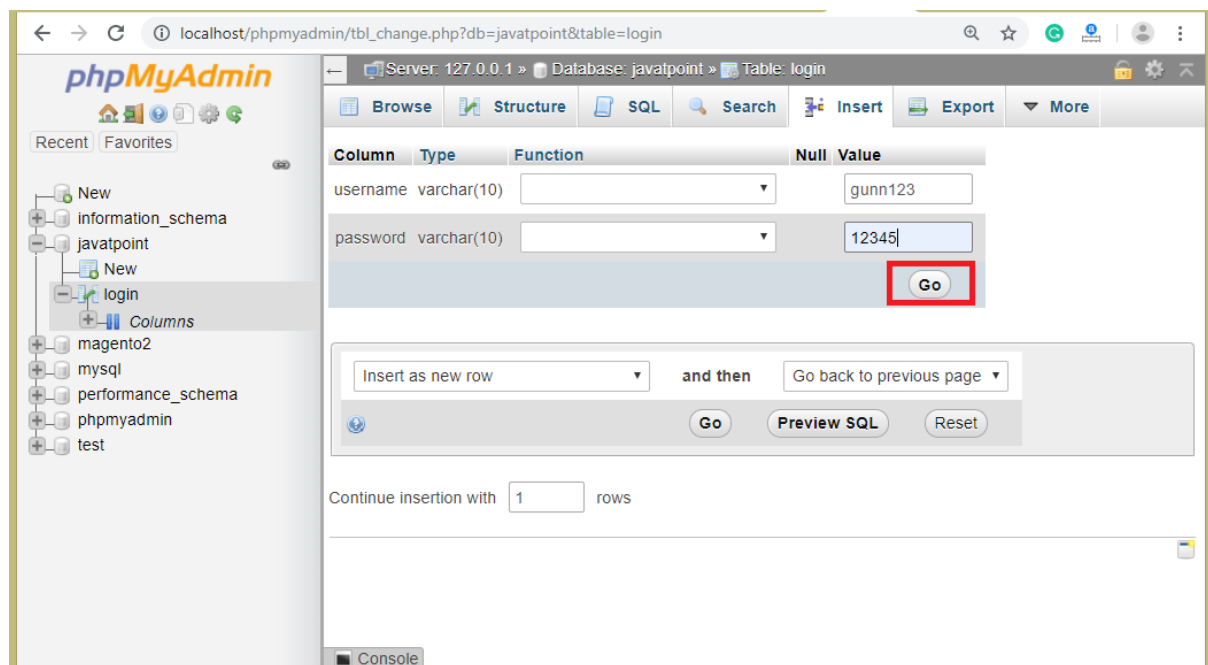
## PG ROOMS MANAGEMENT

### Steps to create a database in PHPMyAdmin:

1) The following figure shows your PHPMyAdmin interface, just enter your database name and click the 'Create' button to create your database.

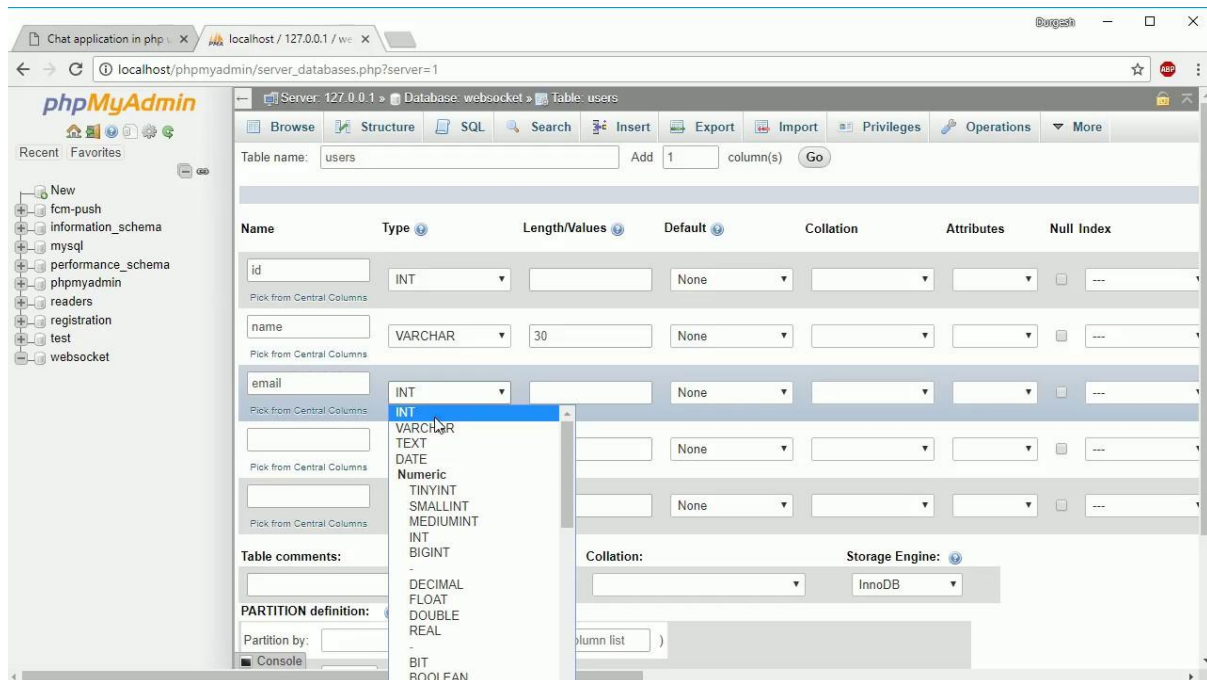


2) Now to create a new table enter your table name and the number of fields in the table, then click the 'Go' Button.

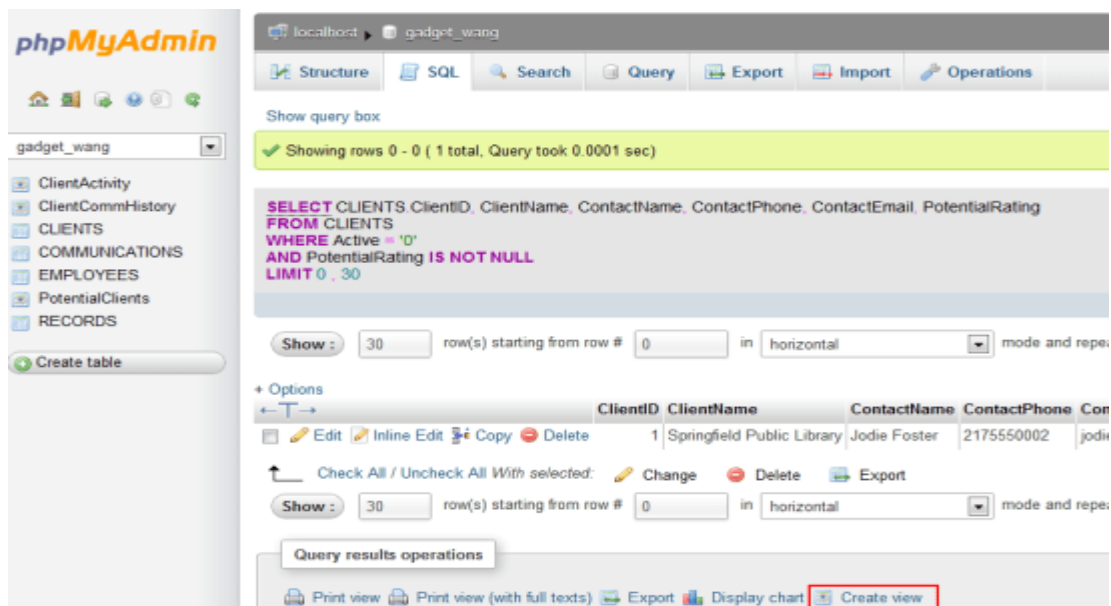


## PG ROOMS MANAGEMENT

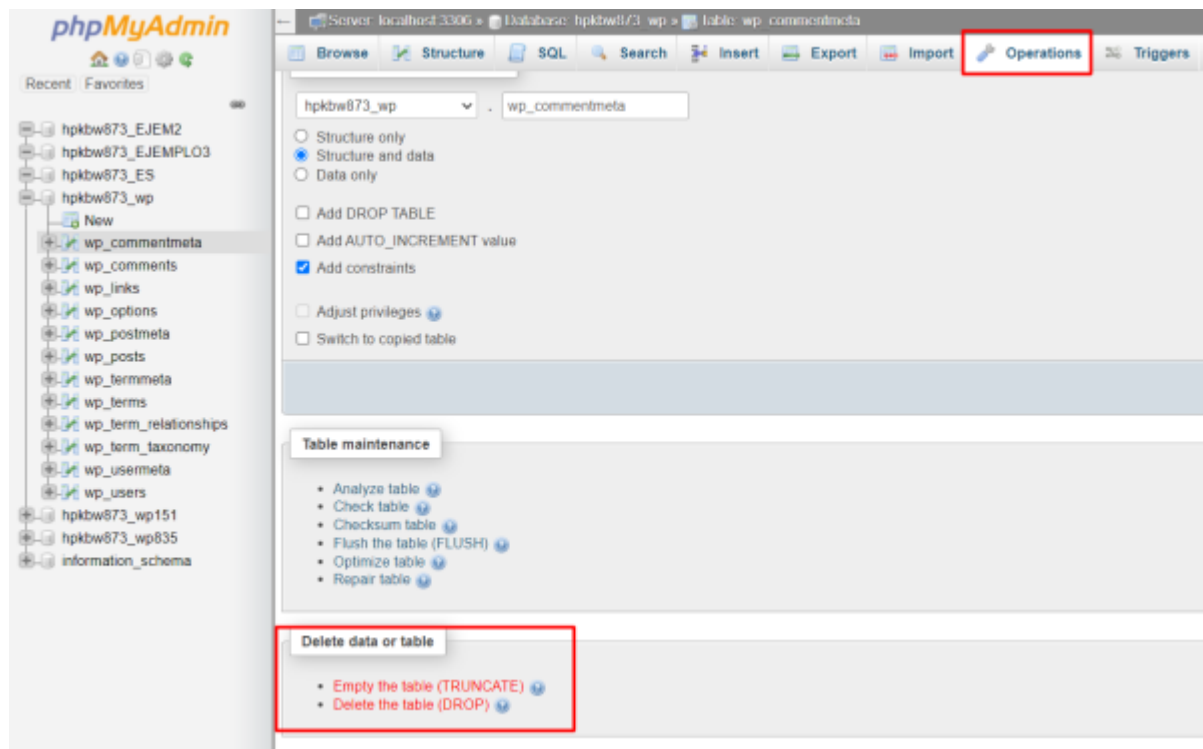
3) The next step is to create the fields, just enter values for each field name, type, length of the field, null option and mention whether it is a primary key or not. Then click the 'Save' button to complete your table creation.



4) The following figure is displayed upon successful creation of your table



The following figure to Drop Table in PHPMYAdmin:



### 3. SYSTEM ANALYSIS

#### **Existing System:**

The existing PG Rooms Management System relies heavily on manual processes, including room allocation using physical registers or basic spreadsheets, tenant information recorded on paper files or rudimentary digital formats, and payments tracked through cash books or simple accounting software. Maintenance requests are typically communicated verbally or through informal channels like messaging apps or emails, often leading to delayed responses and unresolved issues. This manual approach results in frequent errors, inefficiencies, and significant difficulties in tracking tenant information, payment history, and room occupancy rates.

The current system is not only time-consuming but also prone to discrepancies, lacks visibility into room availability, and makes it challenging to generate insightful reports or analytics. Furthermore, managing multiple PG locations becomes increasingly complex, and the lack of centralized data hinders decision-making. These challenges underscore the urgent need for automation and digitalization to streamline operations, enhance accuracy, and improve overall efficiency in managing PG rooms. By adopting a modern, integrated system, PG owners can expect to reduce administrative burdens, enhance tenant satisfaction, and drive business growth.

### **Proposed System:**

The proposed PG Rooms Management System aims to automate and streamline processes, enhancing efficiency, accuracy, and tenant satisfaction. It features a centralized database for tenant information and room allocation, as well as an online portal for tenants to submit maintenance requests, view room details, and access important documents. Real-time room availability tracking will improve occupancy management, enabling swift allocation and minimizing vacancies.

The system will provide customizable reporting and analytics, enabling data-driven decision-making and insights into occupancy rates, maintenance trends, and overall business performance. With a user-friendly interface, secure data storage, and regular updates, the proposed system will modernize PG room management, reducing errors, enhancing tenant experience, improving operational efficiency, and driving business growth through better resource utilization and informed decision-making.

## **4. SYSTEM DESIGN**

### **Introduction**

The purpose of the decision phase is to plan a solution of the problem specified by the requirements document. This phase is the first step in moving the problem domain to the solution domain. It involves the process, in which conceiving, planning and carrying out the plan generating the necessary report, In other words, the design phase act as a bridge between SRS and implementation phase. The design of the system is perhaps the most critical factor affecting the quality of the software, and as a major impact on the later phase, particularly the testing and maintenance.

### **Software Design**

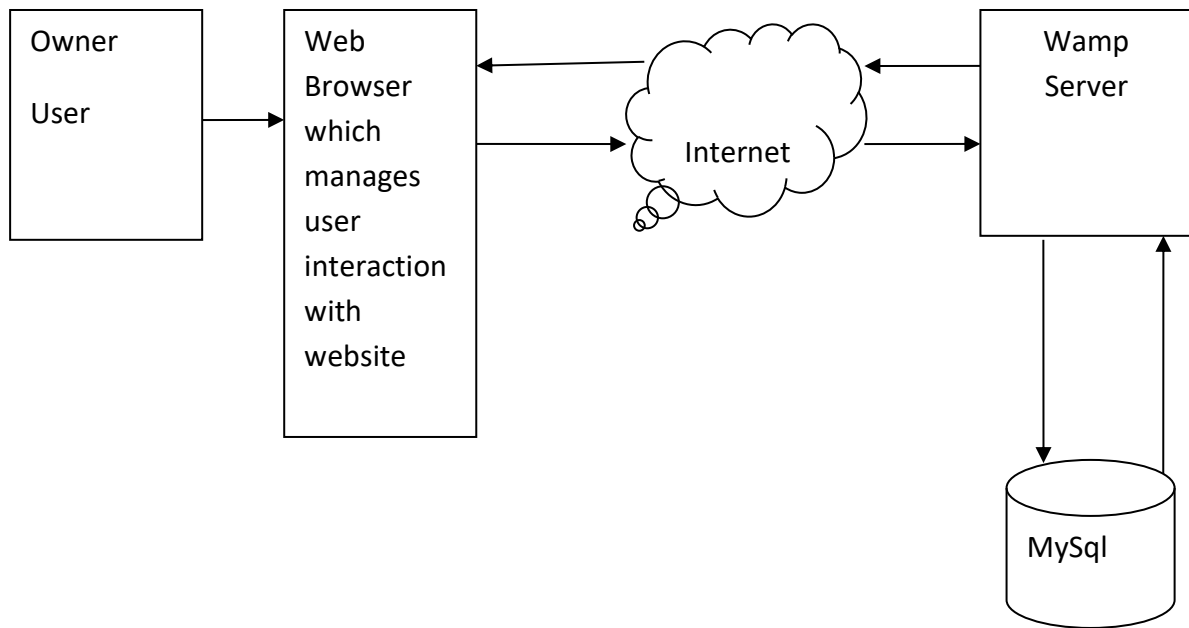
Design is the key phase of any project. It is the first step in moving from the problem domain to the solution domain. The input to the design phase is the specifications of the system to be designed. The output of the top-level design is the architectural design, or the system design for the software system to be built. A design should be very clear, verifiable, complete, traceable, efficient and simple.

### **Architecture Design**

The architecture design defines the relationship among major structural element of the program. Architecture diagram shows the relationship between different components of system. This diagram helps to understand the overall concept of system.



## PG ROOMS MANAGEMENT



### Logical design

The graphical representation of systems' data and how the process transforms the data is known as Data Flow Diagram. It shows the logical flow of the data.

The logical design describes the detailed specification for the system, describing its features, an effective communication and the user interface requirements. The logical system of proposed system should include the following.

1. External system structure.
2. Relationship between all the activities.
3. The physical construction and all the activities.
4. Global data.
5. Control flow.
6. Derived program structure.

### Designing Principles

Basic design principles that enable the software engineer to navigate the design process are:

- The design process should not suffer from “Tunnel vision”.
- The design should be traceable to analysis model.
- The design should not reinvent the wheel.
- The design should minimize the intellectual distance between the software and the problem, as it exists in the real world.
- The design should exhibit uniformity and integrity.
- The design should be structured to accommodate changes.
- The design not coding, the coding is not a design.
- The design should be assessed for the quality, as it is being created, not after the fact.
- The design should be reviewed to minimize the conceptual errors.

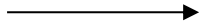
### DATA FLOW DIAGRAM

The data flow diagram(DFD) is one of the important modeling tools. It shows the user of the data pictorially. DFD represents the flow of the data between different transformations and processes in the systems. The data flow diagram shows logical flow of the data. It represents the functional dependencies within a system. It shows output values in a computation or derived from input values. It is a simple pictorial representation or model for system behavior. It specifies, “what is to be done but not how is to be done”. It describes the logical structure of the system. It relates data information to various processes of the system. It follows top-down approach.

## PG ROOMS MANAGEMENT

### Data Flow Diagram Notations

#### Data Flow:



It may be from file-to-file or file-to-process or process-to-process .It is generally in terms of attributes. There may be either an input data flow or output data flow.

#### Functional processing:



The process is nothing but the transformation of data . It starts with the subject and has the verb followed by the subject.

#### Data store:



It includes file, data base and repository. To parallel lines represent it one end closed rectangle.

#### Actor/source/sink:



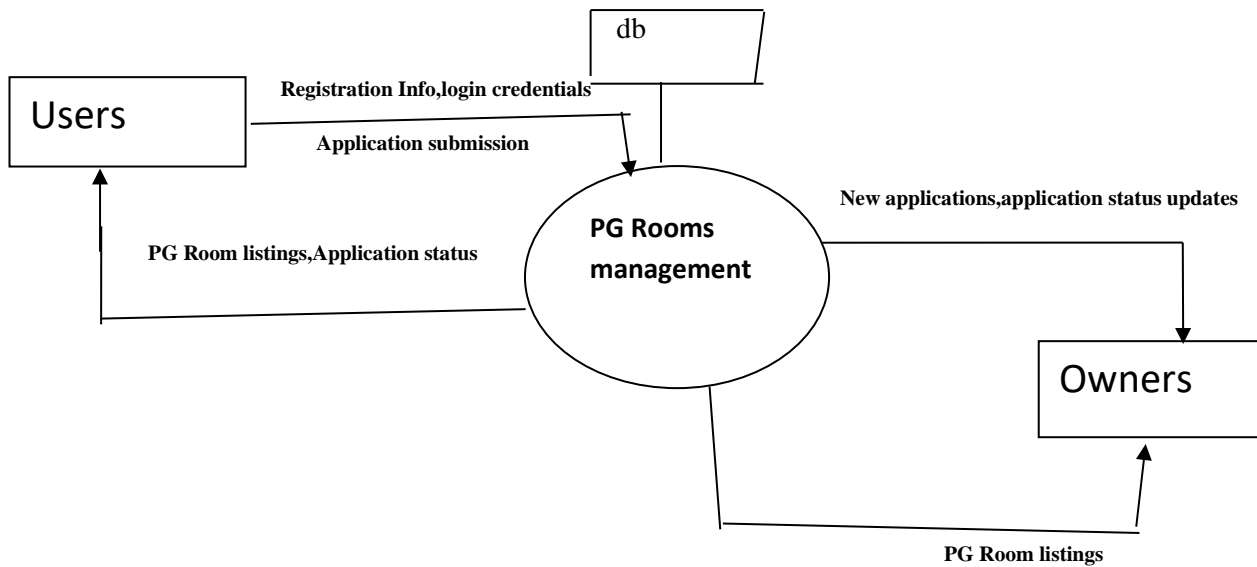
The files which are outside the system and used by the process or processes of the system. Generally Source/sink in the actor.

### Objectives:

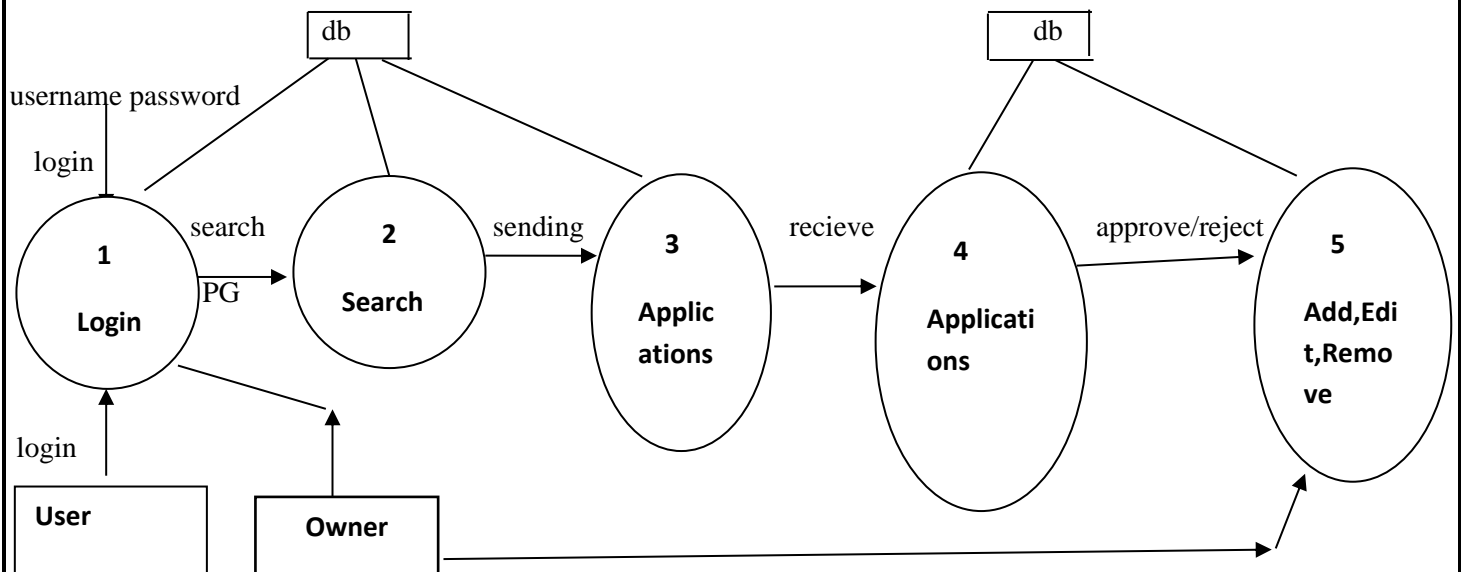
- To graphically document boundaries of a system.
- To provide hierarchy breakdown of the system.
- To show movement of information between a system and its environment.
- To document information flows within the system.
- To aid communication between users and developers.

# PG ROOMS MANAGEMENT

## Context level diagram(zero level DFD):

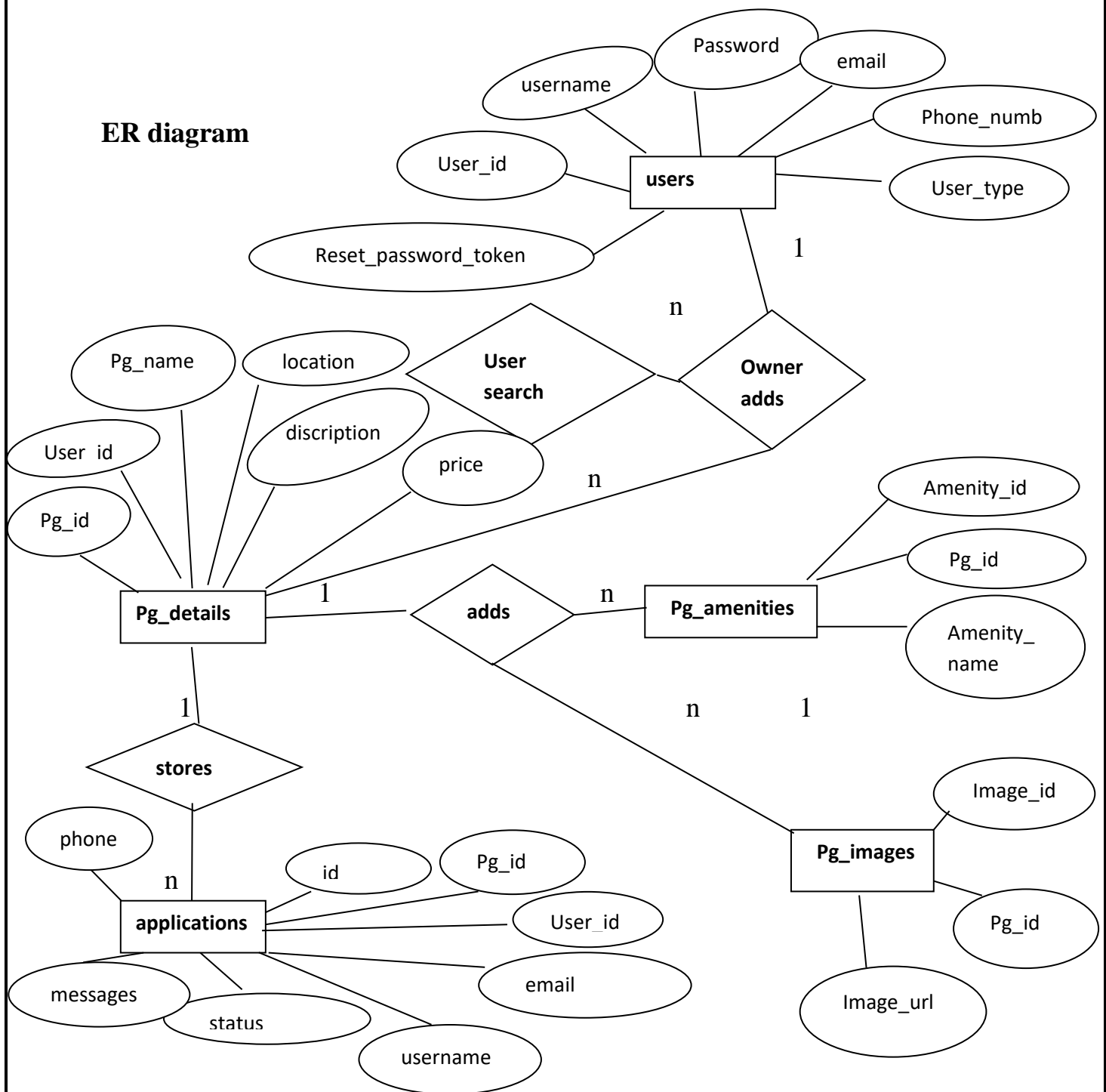


## Context level diagram(one level DFD):



## PG ROOMS MANAGEMENT

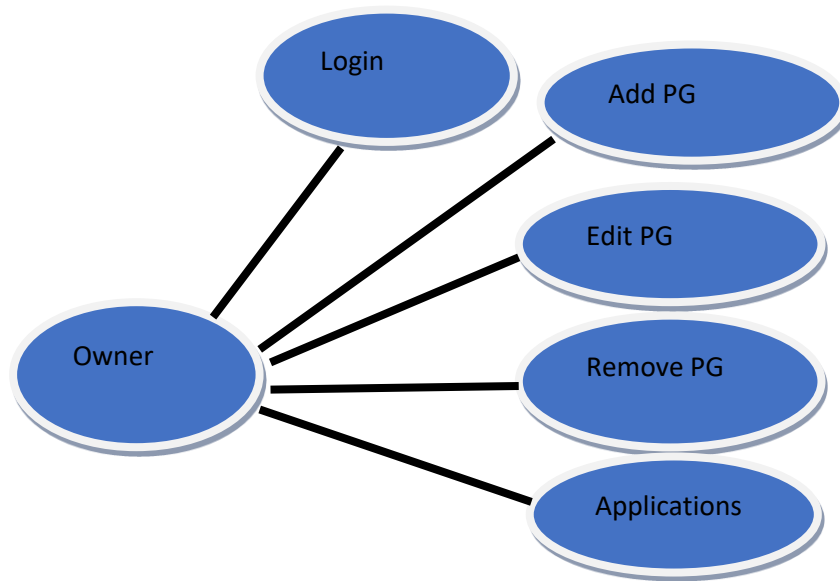
**ER diagram**



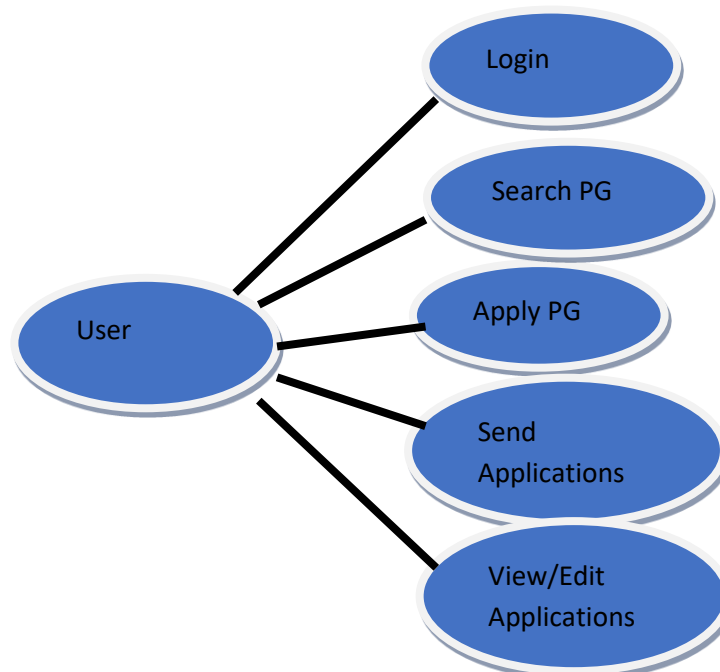
## PG ROOMS MANAGEMENT

### Use Case Diagram

#### Owners:



#### Users:



## 5. SYSTEM DEVELOPMENT

### System Development Phase:

System development is the process of converting the system design into a functional software application. It involves coding, testing, and integrating various modules to ensure the system meets the specified requirements. The PG Rooms Management System was developed using a structured, modular approach for better clarity, maintenance, and scalability.

#### 1. Development Approach

The development followed the Waterfall Model, where each phase (Requirement Analysis, Design, Implementation, Testing, and Deployment) was completed in sequence. This model was ideal due to the well-defined and fixed set of requirements for the system.

#### 2. Tools and Technologies Used

- **Frontend:** HTML, CSS, JavaScript (for forms and user interaction)
- **Backend:** PHP (to handle business logic and database operations)
- **Database:** MySQL (to store tenant records, room details, and maintenance requests)
- **Server:** WAMP/XAMPP (used for local development and testing)
- **Platform:** Windows OS (compatible with WAMP stack)

### **3. Development Steps**

#### **1. Database Creation:**

Tables for tenants, rooms, and maintenance requests were created using MySQL. Relationships were defined using primary and foreign keys.

#### **2. Frontend Design:**

Web pages were designed with HTML and styled using CSS. Forms were created for tenant entry, room entry, and maintenance requests.

#### **3. Backend Development:**

PHP scripts were written to handle logic such as room allocation, tenant management, and maintenance request tracking. Data validation was included to ensure input correctness.

#### **4. Module Integration:**

Each functional module (e.g., tenant management, room management, maintenance tracking) was tested.

### **4. Testing and Debugging**

- Functional testing ensured each feature worked as intended.
- Sample data was entered to verify room allocation and maintenance request updates.
- Bugs and errors were identified and corrected.



### **5. Outcome**

The final system is a web-based application that allows :

- Manage tenant data
- Record and track room allocation
- Handle maintenance requests
- Manage room details

It offers a simple interface and operates smoothly on standard local server environments.

## 6. SYSTEM IMPLEMENTATION

### TABLES USED IN THE DATABASE

#### Table structure for USERS table:

**Primary Key:** user\_id

Field	Data Type	Length	Allow Null	Constraint
user_id	Int	11	Not Null	Primary
username	Varchar	100	Not Null	
password	Varchar	200	Not Null	
email	Varchar	100	Not Null	
Phone_number	Varchar	20	Not Null	
User_type	Varchar	100	Not Null	
Password_reset_token	Varchar	100	Not Null	

#### Table structure for PG\_DETAILS table:

**Primary Key:** pg\_id

## PG ROOMS MANAGEMENT

Field	Data Type	Length	Allow Null	Constraint
pg_id	Int	100	Not Null	Primary Key
User_id	Int	50	Not Null	Foreign key
Pg_name	Varchar	100	Not Null	
pg_type	Varchar	100	Not Null	
location	Varchar	100	Not Null	
price	Varchar	100	Not Null	
room_type	Varchar	100	Not Null	
description	Varchar1	100	Not Null	

### Table structure for PG\_IMAGES table:

**Primary Key:** image\_id

Field	Data Type	Length	Allow Null	Constraint
image_id	Int	100	Not Null	Primary Key
pg_id	Int	100	Not Null	Foreign Key
Image_url	Varchar	200	Not Null	

## PG ROOMS MANAGEMENT

### Table structure for PG\_AMENITIES table:

**Primary Key:** amenity\_id

Field	Data Type	Length	Allow Null	Constraint
amenity_id	Int	100	Not Null	Primary Key
pg_id	Int	100	Not Null	Foreign key
Amenity_name	Varchar	200	Not Null	

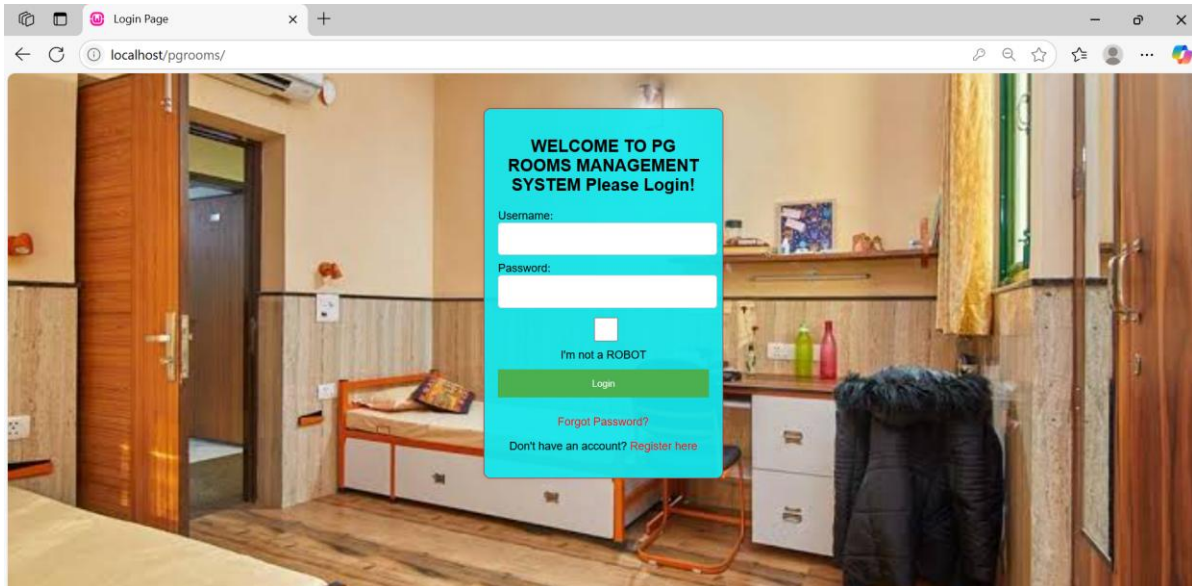
### Table structure for APPLICATIONS table:

**Primary Key:** id

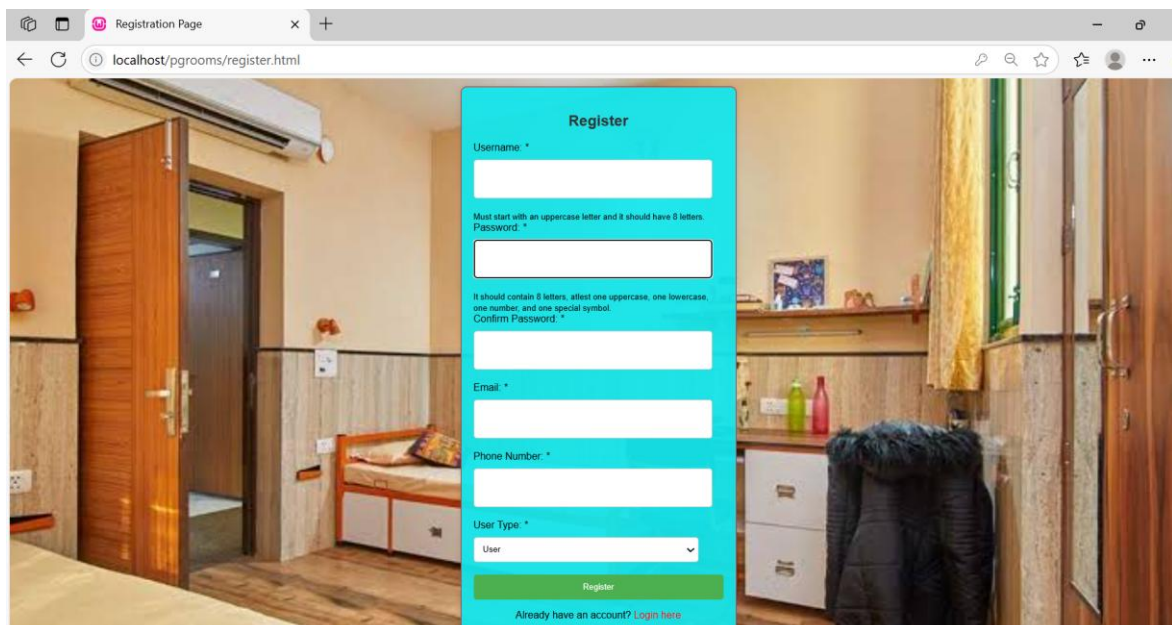
Field	Data Type	Length	Allow Null	Constraint
id	Int	11	Not Null	Primary Key
pg_id	Int	11	Not Null	Foreign Key
user_id	Int	11	Not Null	
username	Varchar	100	Not Null	
email	varchar	100	Not Null	
Phone_number	Varchar	100	Not Null	
message	Varchar	100	Not Null	
Status	Varchar	100	Not Null	

## 7. SCREEN SHOTS

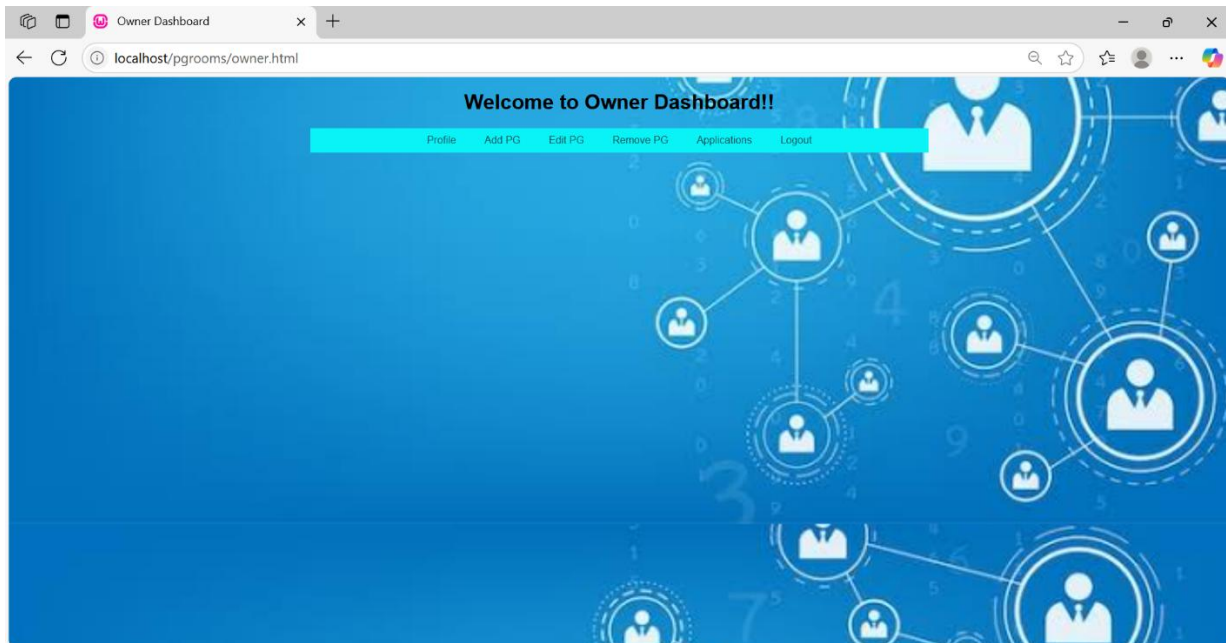
### A VIEW OF LOGIN PAGE



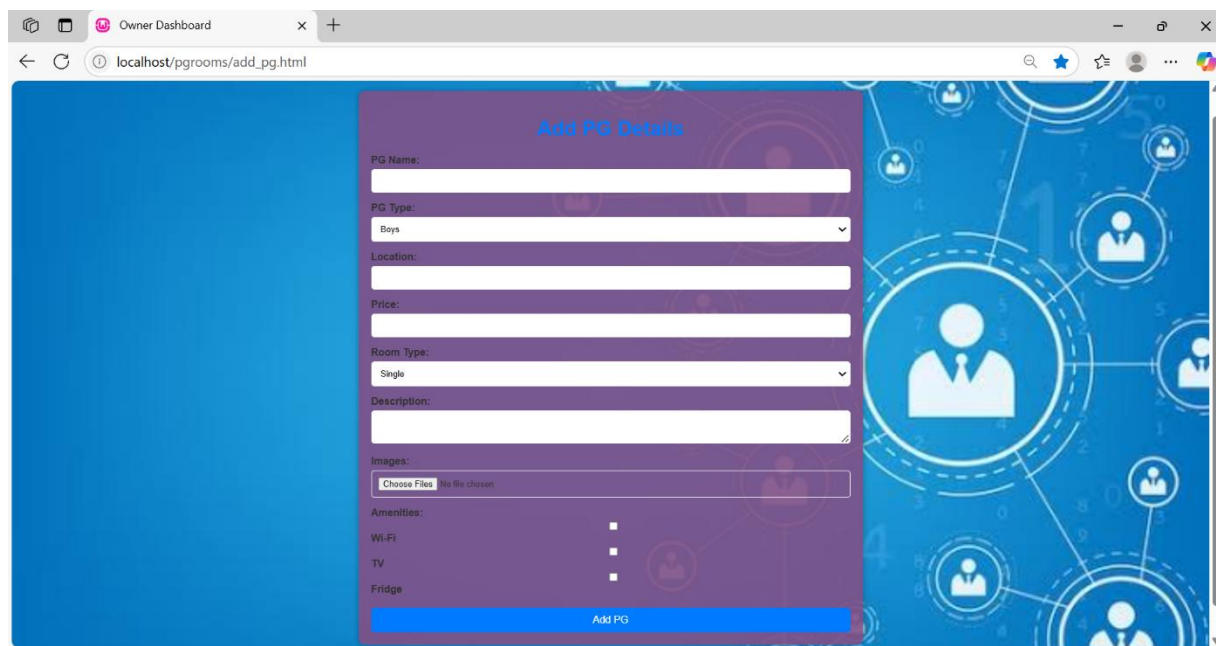
### A VIEW OF REGISTRATION PAGE



## OWNER DASHBOARD



## A VIEW OF ADDING THE PG



# PG ROOMS MANAGEMENT

## A VIEW OF EDITING THE PG

**Edit PG**

PG Name: Shridevi PG

PG Type: Boys

Location: Shankar mantappa opposite road Yallakki Shelter colony Dhanwad

Price: 1700.00

Room Type: shared

Description: Welcome to Shridevi PG, your home away from home! Our PG offers a comfortable and convenient living experience for students and working professionals. Enjoy:

Update PG

## A VIEW OF DELETING THE PG

**Shridevi PG**

Location: Shankar mantappa opposite road Yallakki Shelter colony Dhanwad

Price: 1700.00

Room Type: shared

Description: Welcome to Shridevi PG, your home away from home! Our PG offers a comfortable and convenient living experience for students and working professionals. Enjoy: - Spacious and well-ventilated rooms with attached bathrooms and Wi-Fi connectivity - Delicious and hygienic meals prepared by our experienced cooks - State-of-the-art amenities, including 24/7 power backup, CCTV surveillance, and secure parking - Prime location, close to major educational institutions, offices, and public transportation hubs - Friendly and supportive staff, always available to assist you with your needs Why Choose Us? - Affordable rates, without compromising on quality and comfort - Flexible stay options, including short-term and long-term rentals - Clean and maintained premises, ensuring a healthy living environment Book Your Stay! Contact us at 9007654210 or sk21@gmail.com to reserve your spot. We look forward to welcoming you to Shridevi PG!

Amenities: Wi-Fi

Images:

Delete PG

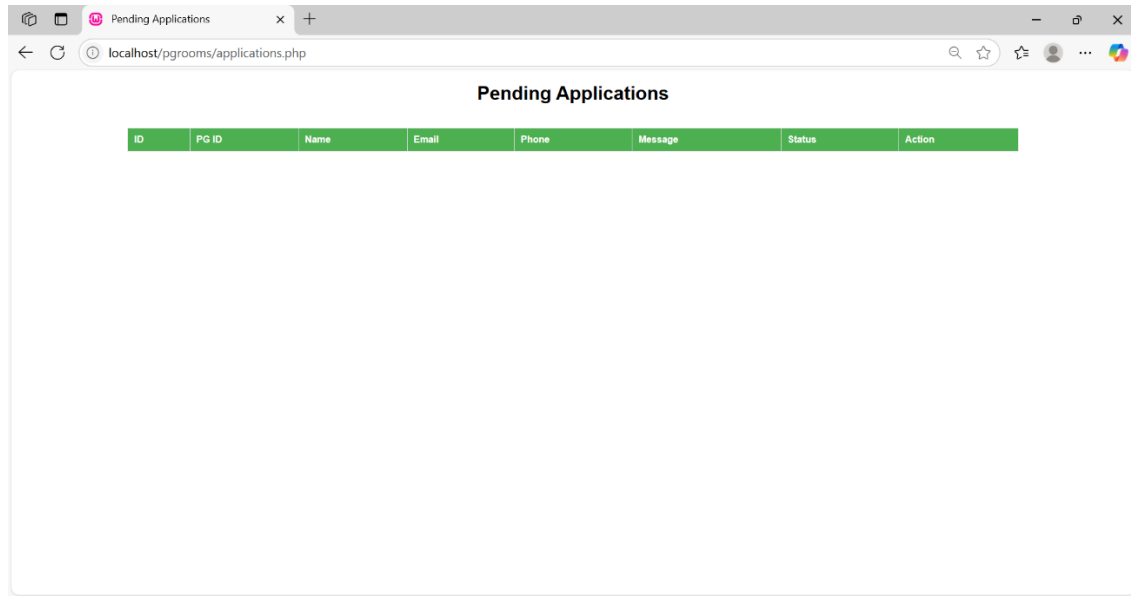
**Delete PG**

Are you sure you want to delete Shridevi PG?

Delete PG Cancel



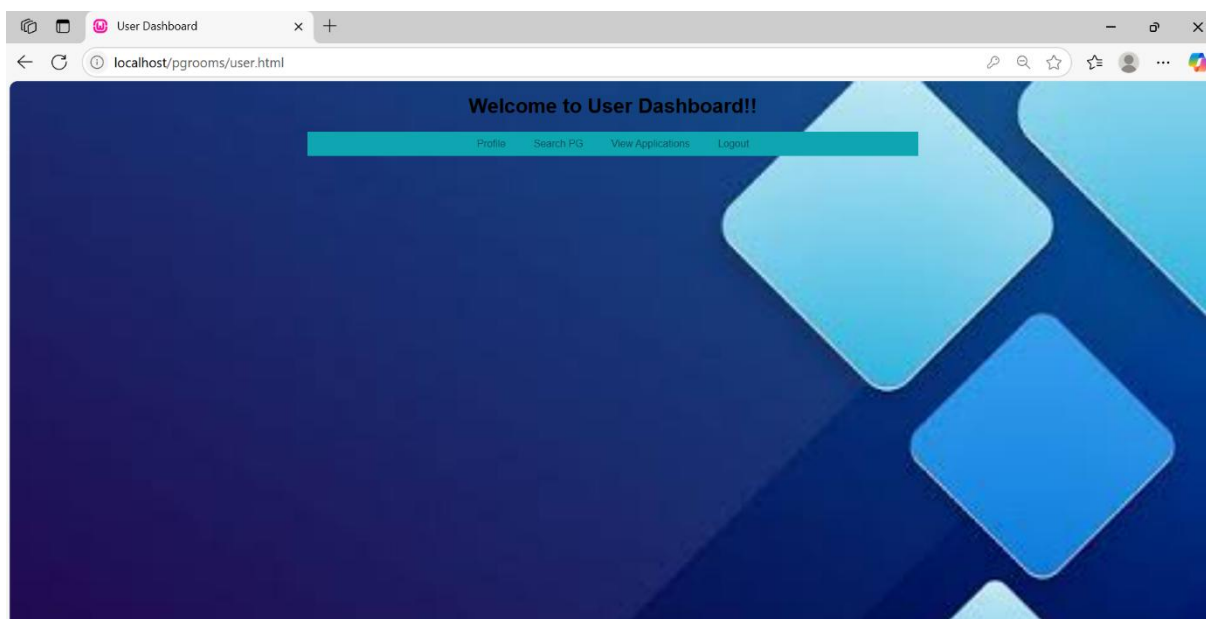
## A VIEW OF APPLICATIONS TO THE PG



Pending Applications

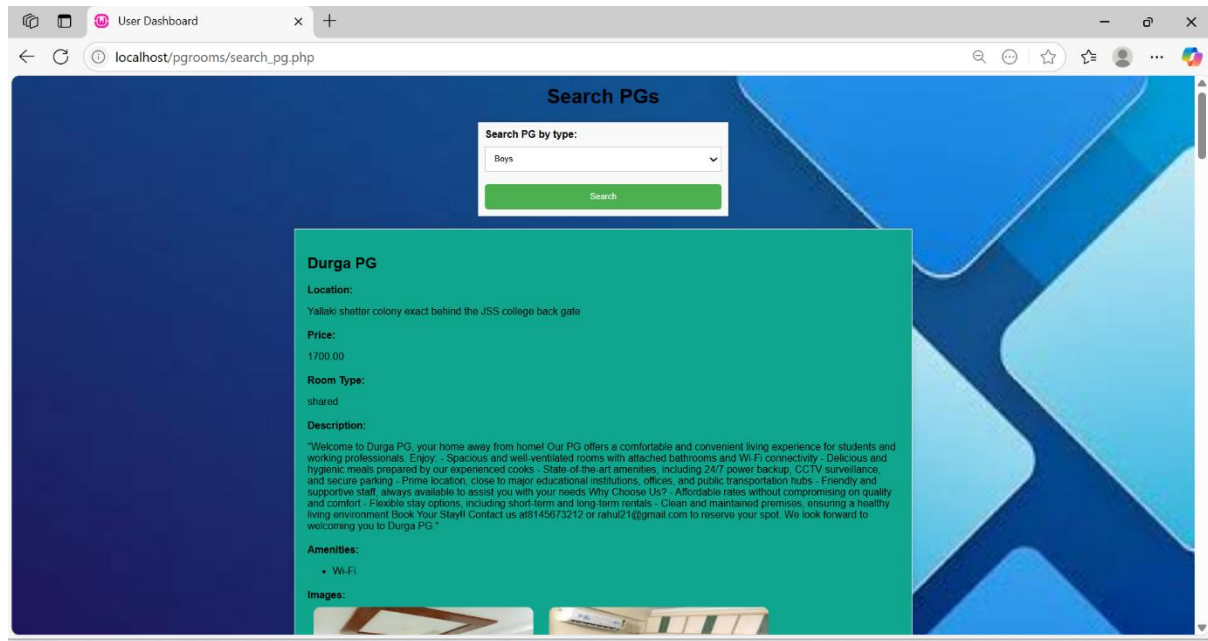
ID	PG ID	Name	Email	Phone	Message	Status	Action
----	-------	------	-------	-------	---------	--------	--------

## USER DASHBOARD

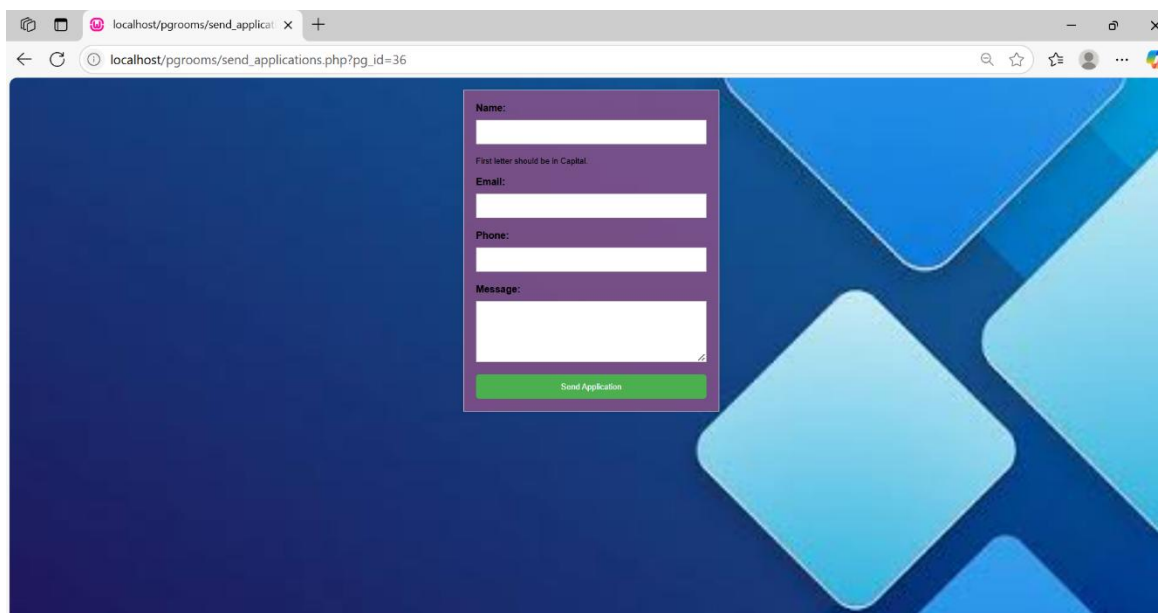




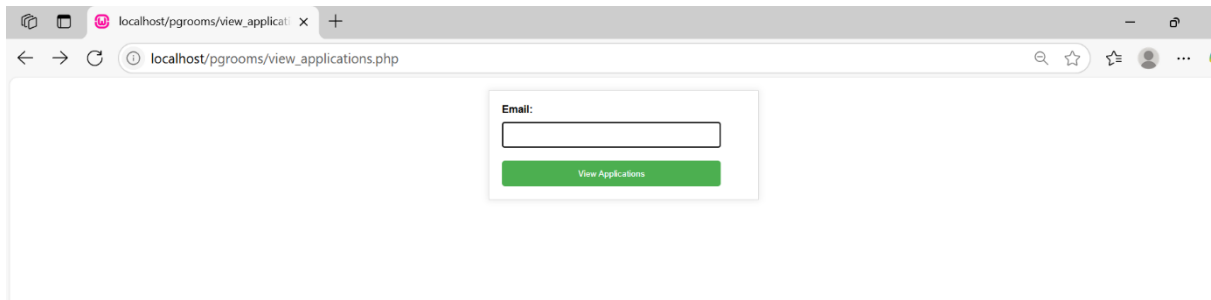
### A VIEW OF USER SEARCH FOR THE PG



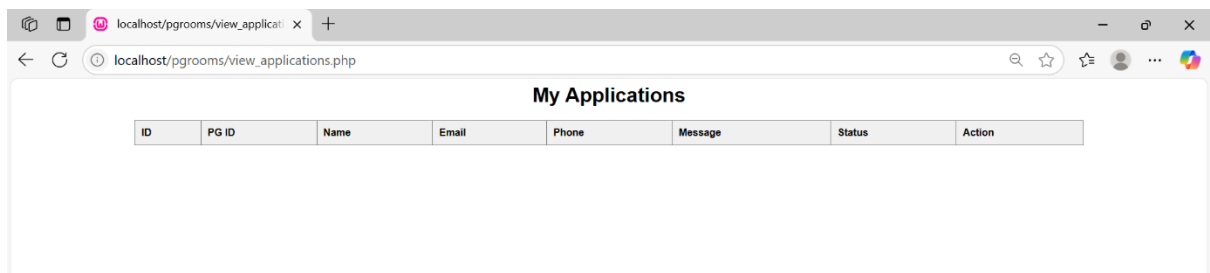
### A VIEW OF SEND APPLICATION TO THE PG



## A VIEW OF VIEW USER APPLICATION THE PG



A screenshot of a web browser window. The address bar shows the URL `localhost/pgrooms/view_applications.php`. The page content features a form with a label "Email:" above a text input field. Below the input field is a green button labeled "View Applications".



A screenshot of a web browser window showing a table titled "My Applications". The table has 8 columns: ID, PG ID, Name, Email, Phone, Message, Status, and Action. The table is currently empty.

ID	PG ID	Name	Email	Phone	Message	Status	Action
----	-------	------	-------	-------	---------	--------	--------

## 8. SYSTEM TESTING

### TESTING:

#### Introduction

Testing is a process of executing a program with the intent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding. System Testing is an important phase. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing. The code is tested at various levels in software testing. Unit, system and user acceptance testings are often performed.

#### Testing Objectives

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an undiscovered error.

#### Testing Principles

- All tests should be traceable to end user requirements.
- Tests should be planned long before testing begins.
- Testing should begin on a small scale and progress towards testing in large.
- Exhaustive testing is not possible.
- To be most effective testing should be conducted by a independent third party.

The primary objective for test case design is to derive a set of tests that has the highest likelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used. They are:

## PG ROOMS MANAGEMENT

- White box testing.
- Black box testing.

**White Box Testing:** White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

**Black Box Testing:** Black box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing mainly focuses on the information domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing functions, interface errors, errors in data structures, error in functional logic are the errors falling in this category.

**Testing strategies:** A strategy for software testing must accommodate low-level tests that are necessary to verify that all small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements.

There are two general strategies for testing software. They are as follows:

**Code Testing:** This examines the logic of the program. To follow this test, cases are developed such that every path of program is tested.

**Specification Testing:** Specification Testing examines the specification, starting what the program should do and how it should perform under various conditions. Then test cases are developed for each condition and combinations of conditions and to be submitted for processing.

## PG ROOMS MANAGEMENT

### Levels of Testing

The stages of Testing Process are:

**Unit Testing:** Individual components are tested to ensure that they operate correctly. Each component tested independently without other system components. Ex. Check for Username and Password with the table, after the next module is loaded.

**Integration Testing:** Integration testing is a systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfacing.

This testing is done using the bottom-up approach to integrate the software components of the software system in to functioning whole.

**System Testing:** System testing is actually a series of different tests whose primary purpose is fully to exercise the computer-based system. The system tests that where applied are recovery testing and performance testing. Finally, a review or audit is conducted which is a final evaluation that occurs only after operating the system long enough for user to have gained a familiarity with it. System testing was done by the inspection team to verify that all the functionality identified is the software requirement specification has been implemented. Defects that crept in the system has been found defect free and is working well. System testing is concerned with interfaces, design logic, control flow recovery, procedures throughput, capacity and timing characteristics of the entire system. For blank field, alphabets, number and special character validation.

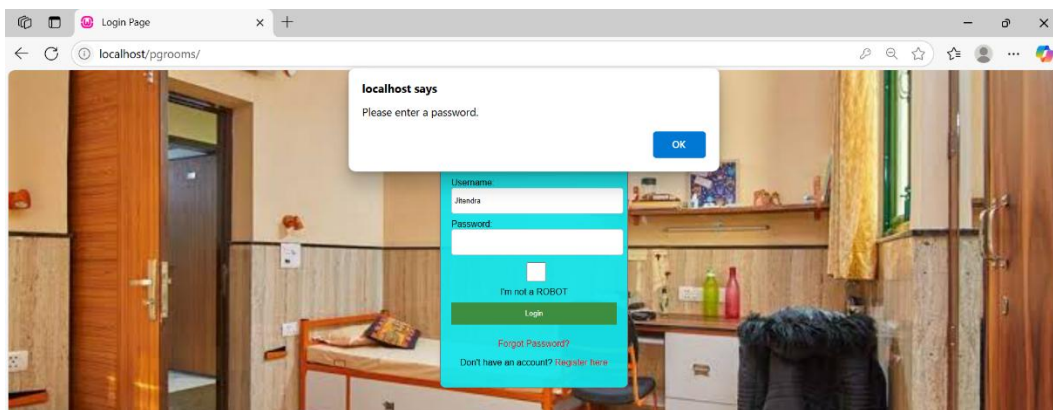
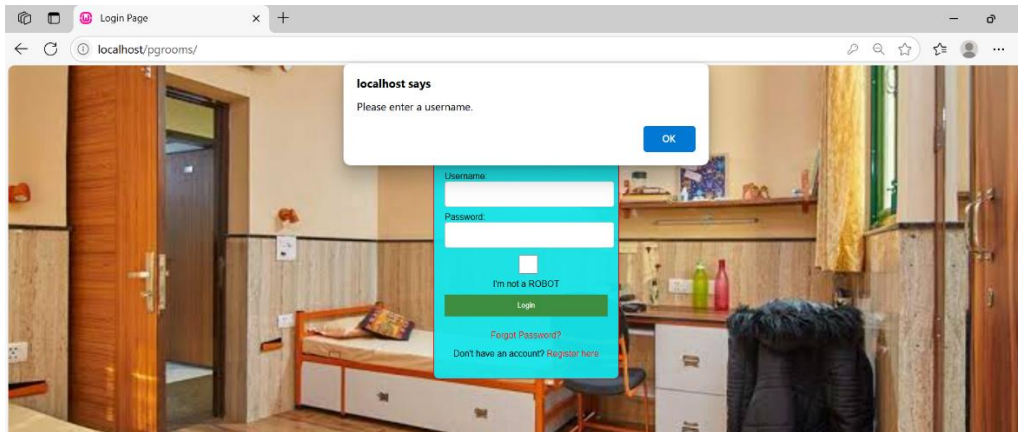
**Acceptance Testing:** User acceptance of the system is the key factor for the success of any system. This is done by user. The system is given to the user and they test it with live data. Acceptance testing involves the planning and execution of functional test. Performance tests, stress tests in order to demonstrate that the implemented system satisfies its requirements. Two sets of acceptance test can be run, those developed by the customer. The system has been tested for its performance at unit

## PG ROOMS MANAGEMENT

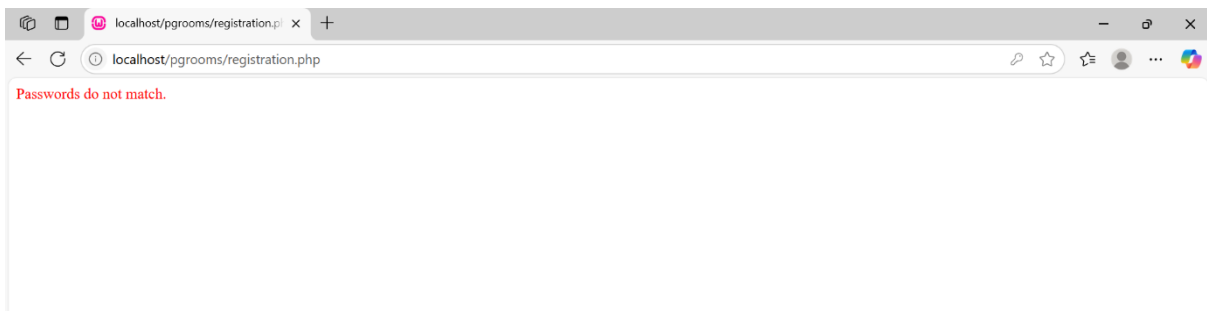
level by the individuals through performance testing that is designed to test the run time performance of the software. The performance of the fully integrated system is tested and was found good.

## Validating the tables

### Login Form

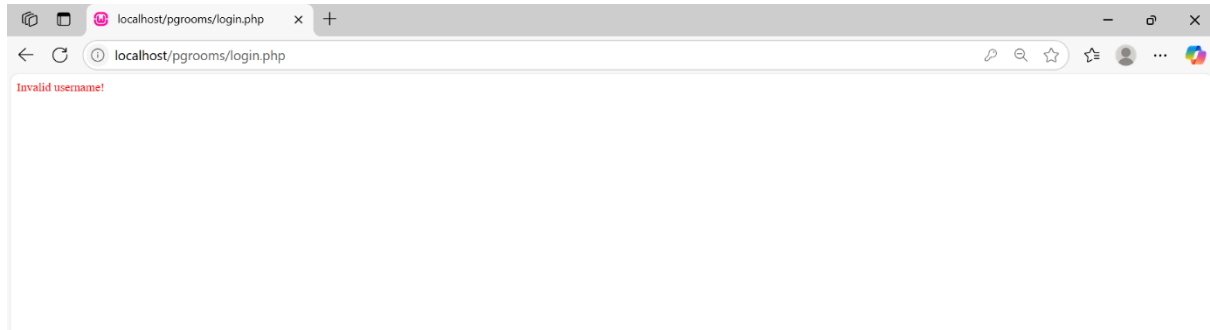


The above screenshots are showing the error message that we should enter valid username and password.

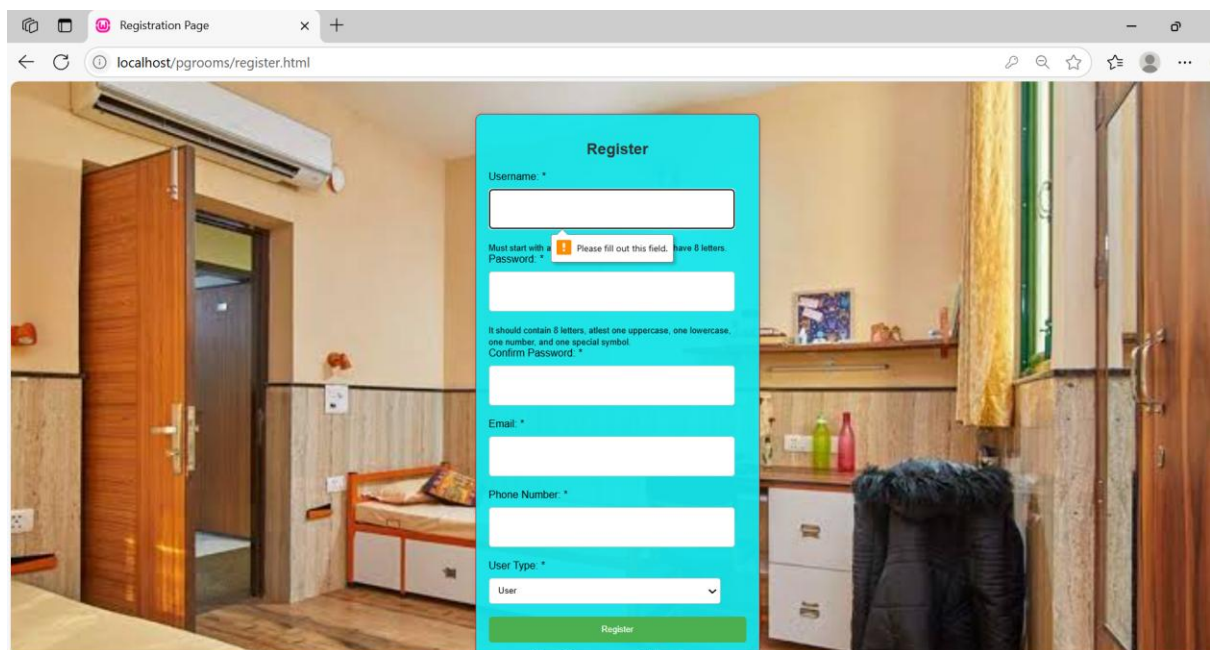


The above screenshot is showing the error message that passwords do not match.

## PG ROOMS MANAGEMENT



The above screenshot is showing the error message that invalid username .



The above screenshot is showing the message that we should fill the all fields.



## PG ROOMS MANAGEMENT

### Functional Testing:

#### Test case :

Test No	Test Case	Expected Result	Pass
1	Leave the Username and Password Fields blank	Message Stating that fill the feilds	✓
2	Invalid Username and Password	Message stating that invalid username and password	✓
3	Mismatch New password and confirm password	Message stating that new password and confirm password donot match	✓
4	Leave all the text boxes blank	Message staring that all these fields should not be empty	✓

## 9. LIMITATIONS OF THE PROJECT

- Dependence on Internet connectivity.
- Dependence on user feedback and ratings.
- Limited integration with external services.

## CONCLUSION

Software is said to have attained its objective only when it need all requirements of the user, further the user himself is the person to judge the success of the system. Every attempt has been made to ensure that the system is fully functional and works effectively and efficiently. The system has been tested with simple data to cover all possible options and checked for all outputs. Since the system is flexible and modular, further modification of this package can be easily incorporated.

### Importance of the system

- Less manual work.
- Increased efficiency.
- Decreases the rate of errors.
- It reduces the time consumption.
- Quick (instant) result.

## **10. FUTURE AND SCOPE OF THE PROJECT**

- Development of Android and IOS application.
- Integration with payment gateways.
- Advanced search filters.
- Rating and Review system

## 11. SOURCE CODE

### Registration Page Code:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Registration Page</title>

    <link rel="stylesheet" href="styles_registration.css">

</head>

<body>

    <div class="register-container">

        <h2>Register</h2>

        <form id="register-form" action="registration.php" method="POST">

            <label for="username ">Username: *</label>

            <input type="text" id="username " name="username" required>

            <small>Must start with an uppercase letter and it should have 8
letters.</small><br>

            <label for="password ">Password: *</label>

            <input type="password" id="password" name="password" required><br>

            <small> It should contain 8 letters, atleast one uppercase, one lowercase, one
number, and one special symbol.</small><br>

            <label for="confirm_password ">Confirm Password: *</label>

            <input type="text" id="confirm-password" name="confirm_password"
required>

            <label for="email">Email: *</label>
```

## PG ROOMS MANAGEMENT

```
<input type="email" id="email" name="email" required>
<label for="phone-number ">Phone Number: *</label>
<input type="phone_number" name="phone_number" required><br>
<label for="user-type ">User Type: *</label>
<select id="user-type " name="usertype" required>
    <option value="user ">User</option>
    <option value="owner ">Owner</option>
</select>
<button type="submit" name="submit">Register</button>
<p>Already have an account? <a href=" index.html ">Login here</a></p>
</form>
</div>
</body>
</html>
```

```
<?php
// db_connection.php

$servername = "localhost";
$username = "root";
$password="";
$database = "pgrooms";

// Create connection

$conn = new mysqli($servername,$username,"",$database);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

## PG ROOMS MANAGEMENT

```
}  
?>  
<?php  
require_once 'db_connection.php';  
error_reporting(1);  
if (isset($_POST['submit'])) {  
    $username = $_POST['username'];  
    $password = $_POST['password'];  
    $confirm_password = $_POST['confirm_password'];  
    $email = $_POST['email'];  
    $phone_number = $_POST['phone_number'];  
    $usertype = $_POST['usertype'];  
    $created_at = date("Y-m-d H:i:s");  
    $updated_at = date("Y-m-d H:i:s");  
    // Validation  
    $errors = array();  
    if ($password !== $confirm_password) {  
        $errors[] = "<p style='color:red';>Passwords do not match.</p>";  
    }  
    if (!preg_match("/^[a-zA-Z0-9._%+-]+@gmail\.com$/", $email)) {  
        $errors[] = "<p style='color:red';>Email should be a valid Gmail address.</p>";  
    }  
    if (!preg_match("/^[0-9]{10}$/", $phone_number)) {  
        $errors[] = "<p style='color:red';>Mobile number should have exactly 10  
digits.</p>";  
    }  
}
```

## PG ROOMS MANAGEMENT

```
if (empty($errors)) {  
    // Check if the username already exists  
  
    $sql = "SELECT * FROM users WHERE username = '$username'";  
  
    $result = $conn->query($sql);  
  
    if ($result->num_rows > 0) {  
        echo "<p style='color:red';>Username already exists. Please choose a different  
username.</p>";  
    } else {  
        // Check if the email already exists  
  
        $sql = "SELECT * FROM users WHERE email = '$email'";  
  
        $result = $conn->query($sql);  
  
        if ($result->num_rows > 0) {  
            echo "<p style='color:red';>Email already exists. Please use a different  
email.</p>";  
        } else {  
            // Check if the mobile number already exists  
  
            $sql = "SELECT * FROM users WHERE phone_number =  
'$phone_number'";  
  
            $result = $conn->query($sql);  
  
            if ($result->num_rows > 0) {  
                echo "<p style='color:red';>Mobile number already exists. Please use a  
different mobile number.</p>";  
            } else {  
                // Insert the new user into the database  
  
                $hashed_password = password_hash($password,  
PASSWORD_DEFAULT);  
  
                $sql = "INSERT INTO users (username, password, email,  
phone_number, usertype, created_at, updated_at) VALUES ('$username',
```

## PG ROOMS MANAGEMENT

```
'$hashed_password', '$email', '$phone_number', '$usertype', '$created_at',
'$updated_at')";

$result = $conn->query($sql);

if ($result) {
    echo "<p style='color:green';>Registration successful </p>";
} else {
    echo "<p style='color:red';>Error: </p>" . $conn->error;
}
}
}
}
} else {
    foreach ($errors as $error) {
        echo $error . "<br>";
    }
}
}

$conn->close();

?>
```

### Login Page Code:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Login Page</title>
```



## PG ROOMS MANAGEMENT

```
<link rel="stylesheet" href="styles.css">

</head>

<body>

  <div class="login-container">

    <h2>WELCOME TO PG ROOMS MANAGEMENT SYSTEM Please
Login!</h2>

    <form id="login-form" action="login.php" method="POST">

      <label for="username">Username:</label>

      <input type="text" id="username" name="username"><br>

      <label for="password">Password:</label>

      <input type="password" id="password" name="password"><br>

      <div class="robot-verification">

        <input type="checkbox" id="not-robot" name="not-robot">

        <label for="not-robot">I'm not a ROBOT</label>

      </div>

      <button type="submit" name="submit">Login</button>

      <p><a href="forgot.html">Forgot Password?</a></p>

      <p>Don't have an account? <a href="register.html">Register here</a></p>

    </form>

  </div>

</body>

</html>

<?php
ob_start();
session_start();
```

## PG ROOMS MANAGEMENT

```
error_reporting(E_ALL);
ini_set('display_errors', 1);
if (isset($_POST['submit'])) {
    $username = trim($_POST['username']);
    $password = trim($_POST['password']);
    // Check if username and password are entered
    if (empty($username)) {
        echo "<p style='color:red';>please enter username!!</p>";
        exit;
    }
    if (empty($password)) {
        echo "<p style='color:red';>Please enter password!!</p>";
        exit;
    }
    // Check if robot checkbox is checked
    if (!isset($_POST['not-robot'])) {
        echo "<p style='color: red ;>Please verify you are not a robot!</p>";
        exit;
    }
    $conn = mysqli_connect("localhost", "root", "", "pgrooms");
    if (!$conn) {
        die("Connection failed: " . mysqli_connect_error());
    }
    $query = "SELECT * FROM users WHERE username = '$username'";
    $result = mysqli_query($conn, $query);
    $row = mysqli_fetch_assoc($result);
```

## PG ROOMS MANAGEMENT

```
if ($row) {  
    if (password_verify($password, $row['password'])) {  
        $_SESSION['username'] = $username;  
        $_SESSION['user_id'] = $row['user_id'];  
        $_SESSION['usertype'] = $row['usertype'];  
  
        if (trim(strtolower($row['usertype'])) === 'user') {  
            header("Location: user.html");  
            exit();  
        } elseif (trim(strtolower($row['usertype'])) === 'owner') {  
            header("Location: owner.html");  
            exit();  
        } else {  
            echo "<p style='color:red;'>Unknown usertype.</p>";  
        }  
    } else {  
        echo "<p style='color:red;'>Invalid password!</p>";  
    }  
} else {  
    echo "<p style='color:red;'>Invalid username!</p>";  
}  
mysqli_close($conn);  
}  
?>
```

### Owner Dashboard Page:

```
<!DOCTYPE html>

<html>

<head>

    <title>Owner Dashboard</title>

</head>

<body>

<h1>Welcome to Owner Dashboard!!</h1>

    <header>

        <nav>

            <ul>

                <li><a href="profile1.php">Profile</a></li>

                <li><a href="add_pg.html">Add PG</a></li>

                <li><a href="edit_pg1.php">Edit PG</a></li>

                <li><a href="Remove1.php">Remove PG</a></li>

                <li><a href="applications.php">Applications</a></li>

                <li><a href="logout.php">Logout</a></li>

            </ul>

        </nav>

    </header>

</body>

</html>
```

### Add PG Page Code:

```
<!DOCTYPE html>

<html>
```

## PG ROOMS MANAGEMENT

```
<head>

  <title>Owner Dashboard</title>

  <link rel="stylesheet" href="styles1.css">

</head>

<body>

  <div class="container">

    <h1>Add PG Details</h1>

    <form action="add_pg.php" method="post" enctype="multipart/form-data">

      <div class="form-group">

        <label for="pg_name">PG Name:</label>

        <input type="text" id="pg_name" name="pg_name">

      </div>

      <div class="form-group">

        <label for="pg_type">PG Type:</label>

        <select name="pg_type">

          <option value="boys">Boys</option>

          <option value="girls">Girls</option>

        </select>

      </div>

      <div class="form-group">

        <label for="location">Location:</label>

        <input type="text" id="location" name="location">

      </div>

      <div class="form-group">

        <label for="price">Price:</label>

        <input type="number" id="price" name="price">

      </div>

    </form>

  </div>

</body>

</html>
```

## PG ROOMS MANAGEMENT

```
</div>

<div class="form-group">

  <label for="room_type">Room Type:</label>

  <select name="room_type">

    <option value="single">Single</option>

    <option value="shared">Shared</option>

  </select>

</div>

<div class="form-group">

  <label for="description">Description:</label>

  <textarea id="description" name="description"></textarea>

</div>

<div class="form-group">

  <label for="images">Images:</label>

  <input type="file" id="images" name="images[]" multiple>

</div>

<div class="form-group">

  <label for="amenities">Amenities:</label>

  <input type="checkbox" id="amenity1" name="amenities[]" value="Wi-
Fi">

  <label for="amenity1">Wi-Fi</label>

  <input type="checkbox" id="amenity2" name="amenities[]" value="TV">

  <label for="amenity2">TV</label>

  <input type="checkbox" id="amenity3" name="amenities[]"
value="Fridge">

  <label for="amenity3">Fridge</label>
```

## PG ROOMS MANAGEMENT

```
</div>

<input type="submit" class="submit-btn" value="Add PG">

</form>

</div>

</body>

</html>

<?php

// Start session

session_start();

// Connect to database

$conn = mysqli_connect("localhost", "root", "", "pgrooms");

// Check connection

if (!$conn) {

    die("Connection failed: " . mysqli_connect_error());

}

// Check if form is submitted

if ($_SERVER["REQUEST_METHOD"]=="POST") {

    // Check if user is logged in

    if (isset($_SESSION['user_id'])) {

        $user_id = $_SESSION['user_id'];

        // Get form data

        $pg_name = $_POST["pg_name"];

        $pg_type = $_POST['pg_type'];

        $location = $_POST["location"];

        $price = $_POST["price"];

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

## PG ROOMS MANAGEMENT

```
$description = $_POST["description"];

$amenities = $_POST["amenities"];

// Insert PG details into database

$sql = "INSERT INTO pg_details (pg_name, pg_type, location, price,
room_type, description, user_id) VALUES ('$pg_name', '$pg_type', '$location',
'$price', '$room_type', '$description', '$user_id')";

if (mysqli_query($conn, $sql)) {

    $pg_id = mysqli_insert_id($conn);

    // Insert amenities into database

    foreach ($amenities as $amenity) {

        $sql = "INSERT INTO pg_amenities (pg_id, amenity_name) VALUES
('$pg_id', '$amenity')";

        mysqli_query($conn, $sql);

    }

    // Handle image uploads

    if (isset($_FILES["images"])) {

        $images = $_FILES["images"];

        for ($i = 0; $i < count($images["tmp_name"]); $i++) {

            $image_name = $images["name"][$i];

            $image_url = "uploads/" . $image_name;

            move_uploaded_file($images["tmp_name"][$i], $image_url);

            $sql = "INSERT INTO pg_images (pg_id, image_url) VALUES
('$pg_id', '$image_url')";

            mysqli_query($conn, $sql);

        }

    }

    echo " <p style='color:green';>PG added successfully!</p>";
```



## PG ROOMS MANAGEMENT

```
        } else {  
            echo "<p style='color:red';>Error adding PG: </p>" . mysqli_error($conn);  
        }  
    } else {  
        echo "<p style='color:red';>You must be logged in to add a PG.</p>";  
    }  
}  
mysqli_close($conn);  
?>
```

### Edit PG Page code:

```
<?php  
// Start session  
session_start();  
  
// Connect to database  
$conn = mysqli_connect("localhost", "root", "", "pgrooms");  
  
// Check connection  
if (!$conn) {  
    die("Connection failed: " . mysqli_connect_error());  
}  
  
if (isset($_GET['pg_id'])) {  
    $pg_id = $_GET['pg_id'];  
    $user_id = $_SESSION['user_id']; // Assuming you store user ID in session  
  
    // Get PG details  
    $sql = "SELECT * FROM pg_details WHERE pg_id = '$pg_id' AND user_id = '$user_id'";  
    $result = mysqli_query($conn, $sql);
```

## PG ROOMS MANAGEMENT

```
if (mysqli_num_rows($result) > 0) {  
    $pg_details = mysqli_fetch_assoc($result);  
  
    // Update PG details  
  
    if ($_SERVER["REQUEST_METHOD"]=="POST") {  
        $pg_name = $_POST["pg_name"];  
        $pg_type = $_POST['pg_type'];  
        $location = $_POST["location"];  
        $price = $_POST["price"];  
        $room_type = $_POST['room_type'];  
        $description = $_POST["description"];  
  
        $sql = "UPDATE pg_details SET pg_name = '$pg_name', pg_type =  
'$pg_type', location = '$location', price = '$price', room_type = '$room_type',  
description = '$description' WHERE pg_id = '$pg_id' AND user_id = '$user_id'";  
  
        if (mysqli_query($conn, $sql)) {  
            header("Location:update.php");  
            exit;  
        } else {  
            echo "Error updating PG: " . mysqli_error($conn);  
        }  
    }  
} else {  
    echo "You do not have permission to edit this PG.";  
    exit;  
}  
} else {  
    echo "PG ID not found.";
```

## PG ROOMS MANAGEMENT

```
        exit;
    }
?>

<!DOCTYPE html>

<html>

<head>

    <title>Edit PG</title>

</head>

<body>

    <div class="form-container">

        <h2>Edit PG</h2>

        <form action="" method="post">

            <label>PG Name:</label>

            <input type="text" name="pg_name" value="<?php echo
$pg_details['pg_name']; ?>"><br><br>

            <label>PG Type:</label>

            <select name="pg_type">

                <option value="boys" <?php if($pg_details['pg_type'] == 'boys') echo
'selected'; ?>>Boys</option>

                <option value="girls" <?php if($pg_details['pg_type'] == 'girls') echo
'selected'; ?>>Girls</option>

            </select><br><br>

            <label>Location:</label>

            <input type="text" name="location" value="<?php echo
$pg_details['location']; ?>"><br><br>

            <label>Price:</label>
```

## PG ROOMS MANAGEMENT

```
<input type="text" name="price" value="<?php echo $pg_details['price'];
?>"><br><br>

<label>Room Type:</label>

<input type="text" name="room_type" value="<?php echo
$pg_details['room_type']; ?>"><br><br>

<label>Description:</label>

<textarea name="description"><?php echo $pg_details['description'];
?></textarea><br><br>

<input type="submit" value="Update PG">

</form>

</div>

</body>

</html>
```

### Remove PG Page code:

```
<?php

// Connect to database

$conn = mysqli_connect("localhost", "root", "", "pgrooms");

// Check connection

if (!$conn) {

    die("Connection failed: " . mysqli_connect_error());

}

if (isset($_GET['pg_id'])) {

    $pg_id = $_GET['pg_id'];

    // Delete PG

    if (isset($_POST['delete'])) {

        $sql = "DELETE FROM pg_details WHERE pg_id = '$pg_id'";

        if (mysqli_query($conn, $sql)) {
```

## PG ROOMS MANAGEMENT

```
        header("Location: remove1.php"); // Redirect to a page after deletion
    }
    exit;
} else {
    echo "Error deleting PG: " . mysqli_error($conn);
}
}

// Get PG details
$sql = "SELECT * FROM pg_details WHERE pg_id = '$pg_id'";
$result = mysqli_query($conn, $sql);
$pg_details = mysqli_fetch_assoc($result);
?>

<!DOCTYPE html>

<html>

<head>

    <title>Delete PG</title>

</head>

<body>

    <div class="delete-container">

        <h2>Delete PG</h2>

        <p>Are you sure you want to delete <?php echo $pg_details['pg_name'];
?>?</p>

        <form action="" method="post">

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

            <a href="remove1.php" class="cancel-btn">Cancel</a>

        </form>

    </div>
```

## PG ROOMS MANAGEMENT

</body>

</html>

<?php } ?>

### **Applications page code:**

<?php

// Start session

session\_start();

// Connect to database

\$conn = mysqli\_connect("localhost", "root", "", "pgrooms");

// Check connection

if (!\$conn) {

    die("Connection failed: " . mysqli\_connect\_error());

}

// Get applications from database for the logged-in user

\$user\_id = \$\_SESSION['user\_id'];

\$sql = "SELECT \* FROM applications WHERE status = 'pending' AND user\_id = '\$user\_id'";

\$result = mysqli\_query(\$conn, \$sql);

?>

<!DOCTYPE html>

<html>

<head>

    <title>Pending Applications</title>

</head>

<body>

    <h1 style="text-align: center;">Pending Applications</h1>

## PG ROOMS MANAGEMENT

```
<table>

  <tr>

    <th>ID</th>

    <th>PG ID</th>

    <th>Name</th>

    <th>Email</th>

    <th>Phone</th>

    <th>Message</th>

    <th>Status</th>

    <th>Action</th>

  </tr>

  <?php while ($row = mysqli_fetch_assoc($result)) { ?>

    <tr>

      <td><?php echo $row['id']; ?></td>

      <td><?php echo $row['pg_id']; ?></td>

      <td><?php echo $row['username']; ?></td>

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

      <td><?php echo $row['phone']; ?></td>

      <td><?php echo $row['message']; ?></td>

      <td><?php echo $row['status']; ?></td>

      <td>

        <a href="approve.php?id=<?php echo $row['id']; ?>" class="action-btn
approve-btn">Approve</a>

        <a href="reject.php?id=<?php echo $row['id']; ?>" class="action-btn
reject-btn">Reject</a>

      </td>

    </tr>

  </td>

  </tr>

</table>
```

## PG ROOMS MANAGEMENT

```
</tr>

<?php } ?>

</table>

</body>

</html>
```

### User Dashboard Page:

```
<html>

<body>

    <h1> Welcome to User Dashboard!!</h1>

    <header>

        <nav>

            <ul>

                <li><a href="profile.php"> Profile</a></li>

                <li><a href="search_pg.php">Search PG</a></li>

                <li><a href="view_applications.php">View Applications</a></li>

                <li><a href="logout1.php">Logout</a></li>

            </ul>

        </nav>

    </header>

</body>

</html>
```

### Search PG Page code:

```
<?php

// Connect to databas
```



## PG ROOMS MANAGEMENT

```
$conn = mysqli_connect("localhost", "root", "", "pgrooms");

// Check connection

if (!$conn) {

    die("Connection failed: " . mysqli_connect_error());

}

$sql = "SELECT * FROM pg_details";

if (isset($_POST['search'])) {

    $pg_type = $_POST['pg_type'];

    $sql .= " WHERE pg_type = '$pg_type'";

}

$result = mysqli_query($conn, $sql);

function getAmenities($pg_id, $conn) {

    $amenity_sql = "SELECT amenity_name FROM pg_amenities WHERE pg_id = '$pg_id'";

    $amenity_result = mysqli_query($conn, $amenity_sql);

    $amenities = array();

    while ($amenity = mysqli_fetch_assoc($amenity_result)) {

        $amenities[] = $amenity['amenity_name'];

    }

    return $amenities;

}

function getImages($pg_id, $conn) {

    $image_sql = "SELECT image_url FROM pg_images WHERE pg_id = '$pg_id'";

    $image_result = mysqli_query($conn, $image_sql);

    $images = array();

    while ($image = mysqli_fetch_assoc($image_result)) {
```

## PG ROOMS MANAGEMENT

```
$images[] = $image['image_url'];
}
return $images;
}
?>
<!DOCTYPE html>
<html>
<body>
    <h1 style="text-align: center;">Search PGs</h1>
    <form action="" method="post">
<label for="pg_type">Search PG by type:</label>
    <select name="pg_type" id="pg_type">
        <option value="boys">Boys</option>
        <option value="girls">Girls</option>
    </select>
    <input type="submit" name="search" value="Search">
</form>
<?php while ($pg = mysqli_fetch_assoc($result)) { ?>
    <div class="pg-container">
        <h2><?php echo $pg['pg_name']; ?></h2>
        <div class="pg-info">
            <label>Location:</label>
            <p><?php echo $pg['location']; ?></p>
        </div>
        <div class="pg-info">
            <label>Price:</label>
```

## PG ROOMS MANAGEMENT

```
<p><?php echo $pg['price']; ?></p>
</div>

<div class="pg-info">

    <label>Room Type:</label>

    <p><?php echo $pg['room_type']; ?></p>
</div>

<div class="pg-info">

    <label>Description:</label>

    <p><?php echo $pg['description']; ?></p>
</div>

<div class="pg-info">
<label>Amenities:</label>

    <ul>

        <?php foreach (getAmenities($pg['pg_id'], $conn) as $amenity) { ?>

            <li><?php echo $amenity; ?></li>

            <?php } ?>

        </ul>
</div>

<div class="pg-info">

    <label>Images:</label><br>

    <?php foreach (getImages($pg['pg_id'], $conn) as $image) { ?>

        <?php } ?>

    </div>

<div class="pg-info">
```

## PG ROOMS MANAGEMENT

```
<a href="send_applications.php?pg_id=<?php echo $pg['pg_id']; ?>"
class="apply-btn">Apply Now</a>
```

```
</div>
```

```
</div>
```

```
<?php } ?>
```

```
</body>
```

```
</html>
```

### Send Application Page code:

```
<?php
```

```
// Connect to database
```

```
$conn = mysqli_connect("localhost", "root", "", "pgrooms");
```

```
// Check connection
```

```
if (!$conn) {
```

```
    die("Connection failed: " . mysqli_connect_error());
```

```
}
```

```
// Get PG ID
```

```
$pg_id = $_GET['pg_id'];
```

```
// Get PG details
```

```
$sql = "SELECT user_id FROM pg_details WHERE pg_id = '$pg_id'";
```

```
$result = mysqli_query($conn, $sql);
```

```
$pg_details = mysqli_fetch_assoc($result);
```

```
$user_id = $pg_details['user_id'];
```

```
// Check if form is submitted
```

```
if (isset($_POST['submit'])) {
```

## PG ROOMS MANAGEMENT

```
$pg_id = $_POST['pg_id'];
$name = $_POST['username'];
$email = $_POST['email'];
$phone = $_POST['phone'];
$message = $_POST['message'];

// Validate name
if (!preg_match("/^[A-Z][a-zA-Z]*$/", $name)) {
    echo "<p style='color:red;'>Name should start with a capital letter.</p>";
}

// Validate email
elseif (!preg_match("/^[a-zA-Z0-9._%+-]+@gmail\.com$/", $email)) {
    echo "<p style='color:red;'>Invalid email. Only @gmail.com emails are
allowed.</p>";
}

// Validate phone
elseif (!preg_match("/^[0-9]{10}$/", $phone)) {
    echo "<p style='color:red;'>Invalid phone number. Phone number should have 10
digits.</p>";
} else {
    // Check if email or phone already exists

    $sql = "SELECT * FROM applications WHERE email = '$email' OR phone =
'$phone' AND pg_id = '$pg_id'";

    $result = mysqli_query($conn, $sql);

    if (mysqli_num_rows($result) > 0) {
        echo "<p style='color:red;'>Email or phone number already exists.</p>";
    } else {
        // Insert application into database
```

## PG ROOMS MANAGEMENT

```
$sql = "INSERT INTO applications (pg_id, user_id, username, email, phone, message, status) VALUES ('$pg_id', '$user_id', '$name', '$email', '$phone', '$message', 'pending')";
```

```
if (mysqli_query($conn, $sql)) {  
    echo "<p style='color:green;'>Application sent successfully</p>";  
    exit();  
} else {  
    echo "Error: " . mysqli_error($conn);  
}  
}  
}  
?>
```

```
<html>
```

```
<body>
```

```
<!-- Form to send application -->
```

```
<form action="" method="post">
```

```
<input type="hidden" name="pg_id" value="<?php echo $pg_id; ?>">
```

```
<label>Name:</label>
```

```
<input type="text" name="username" required>
```

```
<small>First letter should be in Capital.</small><br><br>
```

```
<label>Email:</label>
```

```
<input type="email" name="email" required><br>
```

```
<label>Phone:</label>
```

```
<input type="text" name="phone" required><br>
```

```
<label>Message:</label>
```

## PG ROOMS MANAGEMENT

```
<textarea name="message" required></textarea><br>
```

```
<input type="submit" name="submit" value="Send Application">
```

```
</form>
```

```
</body>
```

```
</html>
```

## 12. BIBLIOGRAPHY

### Books

- 1. “Software Engineering”, by Ian Somerville, Sixth Edition, Pearson Education Ltd 2007.
- 2. “Web Programming”, by ‘Chris Bates’ Wiley Dreamtech India, 2nd Edition.
- 3. IEEE SRS Format.
- 4. Database Management Systems, by Navathe.

### Websites:

- 1. <http://www.php.net.com> for Php.
- 2. <http://w3schools.com> for Php
- 3. <http://www.sitepoint.com> for php
- 4. <http://www.tutorialpoint.com> for mySQL



