ONLINE HOUSE
RENTAL WEBSITE

A PROJECT REPORT

Submitted to:

Yatika Hasija

SUBMITTED BY:

Narayan Sharma

Registration No: 12205991

in partial fulfillment for the award of the degree of

B.Tech

IN

BRANCH OF STUDY

Computer Science & Engineering

Lovely Professional University, Phagwara

NOVEMBER 2024

BONAFIDE CERTIFICATE

Certified that this project report **"Online House Rental Website"** is the bonafide work of **"Narayan Sharma "**, who carried out the project work under my supervision. This project is submitted in partial fulfillment for the award of the Bachelor of Technology degree in Computer Science and Engineering.

Table Of CONTENT

INTRODUCTION

1.Background

The rental housing market is a dynamic and growing sector that significantly impacts people's lives by providing accessible and affordable living arrangements. With the expansion of digital technology and the internet, the demand for online solutions in various domains has dramatically increased. Real estate and rental services, traditionally dominated by in-person transactions and paperwork, are experiencing a shift towards online platforms that enhance convenience and accessibility.

Traditionally, the process of renting a property involves a series of manual steps, from listing properties to meeting potential tenants and managing rental records. This approach has several limitations. First, it is time-consuming for both property owners and prospective tenants. Property managers must often rely on phone calls, physical meetings, or local advertisements to reach potential renters, which limits their reach and efficiency. For tenants, the process of locating rental options can be tedious and fragmented, often requiring multiple visits to various listings without a centralized point of information.

In response to these challenges, the Simple House Rental System project aims to provide a comprehensive, user-friendly platform that streamlines the rental process. This system enables property owners to list their properties, manage occupancy status, and handle complaints effectively, all within a centralized online platform. For tenants, the system offers easy access to property listings with detailed information, enabling them to search and filter properties based on specific criteria. This online approach not only saves time but also makes the rental process more transparent, accessible, and efficient.

Objectives

The primary goal of the Simple House Rental System is to provide an efficient and scalable solution for rental property management, encompassing both administrative and user-centric functionalities. The specific objectives of this project are as follows:

1. To develop an online platform for managing rental properties: This includes building a web-based application where property details can be systematically stored, managed, and updated in real time, providing both admins and users with easy access to the latest information.

2. To provide features for searching and listing properties with ease: The platform includes search and filter options, allowing tenants to view available properties by location, price, or type. This approach provides a streamlined way for renters to locate properties that meet their specific needs.

3. To ensure efficient customer and admin interfaces: Separate, intuitive interfaces for customers and administrators are integral to the system's usability. The customer panel allows users to browse, register, and interact with the listings, while the admin panel provides tools for managing property listings, tenant details, and complaints efficiently.

By achieving these objectives, the Simple House Rental System bridges the gap between property managers and tenants, reducing reliance on traditional, less efficient methods.

Significance of the Project

The Simple House Rental System project brings several practical benefits and adds value by addressing current limitations in the rental process. Here are some keyways this system contributes to enhancing rental management:

1. Streamlining Rental Processes:

- This system enables property managers to efficiently handle listing updates, tenant registrations, and complaint management without relying on physical paperwork or repeated phone interactions.

- The automation of routine tasks reduces the administrative load, allowing managers to focus on more strategic aspects, such as property maintenance or customer service improvements.

2. Increased Accessibility:

- With a user-friendly online platform, prospective tenants can easily view and compare rental properties from any location, enhancing the accessibility and convenience of finding suitable housing.

- This approach opens up opportunities for both local and remote renters to explore rental options, contributing to an expanded customer base for property managers.

3. Enhanced Transparency and Trust:

- By providing detailed property information, including occupancy status, contact information, and images, the system creates a transparent environment for renters. Tenants can make informed decisions based on comprehensive data, reducing the uncertainty often associated with traditional rental processes.

- Transparency builds trust between property managers and tenants, which is crucial for establishing long-term rental relationships.

4. Centralized Data Management:

- Property details, tenant information, and complaint records are stored in a structured database, facilitating easy access and retrieval of information.

- This centralization of data ensures that all parties, including administrators and users, have accurate, up-to-date information on demand, improving coordination and reducing the likelihood of miscommunication or lost records.

5. Scalability and Future Expansion:

- The online nature of this system allows for scalability, meaning more properties, features, and user types can be added as the platform grows. Potential expansions could include integrating payment options, adding location-based services, or incorporating analytics to help property managers make data-driven decisions.

The development of an online house rental system aligns with contemporary demands for digital solutions in the real estate sector. By integrating these objectives and benefits into a single platform, the Simple House Rental System makes the rental process more manageable, cost-effective, and user-friendly. Through this project, property managers and tenants alike gain a tool that not only meets current needs but is also adaptable for future enhancements, thus ensuring relevance in an increasingly digital landscape.

Overall, the Simple House Rental System presents a modern solution to the limitations of traditional rental management methods, providing a foundation for further development in the online rental services market.

TECHNOLOGY USED

The Simple House Rental System employs a diverse range of technologies to create a cohesive, interactive, and reliable platform for rental property management. Each technology is integral to achieving the system's objectives, from managing backend processes to delivering a smooth user experience on the frontend.

PHP

PHP (Hypertext Preprocessor) is an open-source, server-side scripting language commonly used for web development. In this project, PHP handles the backend processes, enabling the creation of dynamic web pages and managing interactions with the database. PHP is responsible for processing user requests, managing session data, handling form submissions, and performing authentication for secure user access.

In this system, PHP powers essential backend functionalities, such as user registration, login validation, property listing, and complaint management. For example, in the login process, PHP verifies user credentials against database records to grant access. This ensures that only authorized users can interact with secure sections of the application.

MySQL

MySQL is a powerful relational database management system that manages the structured data within the application. It stores essential information on properties, user accounts, and complaints, organized into tables with defined relationships. MySQL's efficiency and scalability make it ideal for handling relational data in real-time, supporting the system's objectives of data integrity and ease of access.

Within this project, MySQL maintains a centralized repository for all rental property details, user records, and complaint logs. This database structure allows the system to fetch and display information quickly. Additionally, the use of structured queries (SQL) ensures that data retrieval and management tasks are efficient, accurate, and optimized for performance.

HTML/CSS

HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) form the core of the web application's structure and presentation. HTML organizes the page layout, while CSS applies visual styling, ensuring a responsive and appealing design. These languages work together to create a consistent, user-friendly interface across various devices.

CSS in particular is used extensively in this project to enhance the visual elements and ensure usability. Styling elements define the look and feel of the application, from property listings to the user dashboard, making the platform visually engaging and easy to navigate.

JavaScript/jQuery

JavaScript is a versatile scripting language used on the client side to add interactivity and enhance the user experience. In this project, JavaScript enables dynamic content updates, form validation, and interactive elements that improve usability. By integrating jQuery, a popular JavaScript library, the system simplifies event handling, animation, and AJAX calls, making the user interface more responsive.

JavaScript and jQuery are responsible for real-time feedback during form submissions, such as error prompts when required fields are left empty. This ensures data accuracy before sending information to the server, improving both user experience and data integrity. These technologies also enhance interactivity, providing users with a smooth, responsive interface.

PHPMailer

PHPMailer is an open-source library that simplifies email communication in PHP applications. In the Simple House Rental System, PHPMailer is used to send email notifications to users, such as registration confirmations and inquiry acknowledgments. This email functionality enhances the system's engagement and communication with users, adding a professional touch to interactions.

By configuring PHPMailer, the system can send automated notifications, keeping users informed about their interactions and inquiries. This feature is particularly useful for ensuring prompt communication and maintaining a seamless experience for users, such as new tenants or property managers.

Summary of Technology Stack

Each technology in the Simple House Rental System plays a unique role in delivering a functional, user-friendly web application. PHP manages backend logic and database interaction, MySQL provides structured data storage, HTML/CSS define the structure and styling, JavaScript/jQuery enhance interactivity, and PHPMailer enables essential communication features. Together, these technologies create a cohesive platform that effectively meets the needs of both users and administrators in managing rental properties online.

System Functionality

The Simple House Rental System is designed to provide a comprehensive suite of features for managing rental properties. With separate panels for customers and administrators, the system offers streamlined processes for registration, property listing, and customer interactions. The main functionalities include customer and admin management, property searches, and complaint handling, all of which contribute to an efficient and user-friendly platform.

Customer Panel

The Customer Panel provides users (potential tenants) with access to property listings, a registration form, and search functionality, allowing them to find and view properties based on various filters. Here's an in-depth look at the key features available within the Customer Panel:

1. User Registration and Login:

- Customers must first register to access the platform. The registration process collects essential information, including username, email, and password, and stores it securely in the database.

- After registering, users can log in and manage their interactions, such as viewing saved properties and submitting complaints.

2. Viewing and Searching Properties:

- Customers can browse available properties with detailed listings, including property name, location, price, and availability status. The system displays a list of properties that users can filter based on specific needs.

- The search functionality is particularly helpful, allowing customers to narrow down listings by criteria such as property name, location, and price range.

- Filters empower customers to quickly locate properties that match their preferences, providing a convenient and time-efficient browsing experience.

3. Responsive Layout:

- The Customer Panel is designed with a responsive layout that adapts to various screen sizes, ensuring a user-friendly experience across devices.

- The panel's organized interface, with clean navigation and filter options, improves usability for customers, whether they're accessing it from a desktop or mobile device.

Admin Panel

The Admin Panel grants administrators the authority to manage all aspects of the platform, including property listings, user records, and customer complaints. This panel is essential for maintaining data accuracy and ensuring a smooth operation of the system. Below are the key features available to administrators:

1. Dashboard Overview:

- Upon logging into the Admin Panel, administrators are presented with a dashboard that provides a snapshot of the system's status, including the number of properties, active users, and recent complaints.

- The dashboard is designed for easy navigation, with quick links to various management sections such as properties, users, and complaints.

2. Property Management:

- Administrators can add, modify, and delete property listings directly through the Admin Panel. Each property entry includes owner details, pricing, location, and additional property-specific information, which are stored in the database.

- Admins can also update occupancy status, ensuring accurate information is displayed for prospective renters.

- The Admin Panel allows administrators to upload images for each property, giving renters a better visual understanding of the listings.

3. Customer Records and Complaint Management:

- Admins have access to a complete list of registered users. They can manage customer records, including viewing and modifying user details or deleting accounts if needed.

- The complaint management feature allows admins to view and respond to issues raised by customers. Each complaint is stored with details such as submission date, customer ID, and issue description.

- Admins can resolve or archive complaints once addressed, ensuring effective communication and problem resolution.

4. Access Control and Security:

- The Admin Panel is secured with access control measures to ensure that only authorized users can access sensitive information.

- Login credentials are verified against stored records, with further security measures to protect user data.

Search Functionality

The Search Functionality provides users with a way to locate specific properties through keyword-based filtering. This feature enhances the overall user experience by allowing customers to find properties that meet their needs quickly and accurately.

1. Filtering Options:

- Customers can filter properties by keywords such as name, location, price range, and availability. This functionality is key for narrowing down options in large databases.

- Users can combine multiple filters to produce precise search results, such as filtering properties within a specific city and within a certain price range.

2. Backend Logic for Search:

- The system processes search requests by querying the database based on the user's input, retrieving only the properties that match the specified criteria.

- This approach reduces the number of displayed properties to those most relevant to the user, making the browsing experience efficient and personalized.

3. Display of Search Results:

- The search results include essential property details, such as the owner's contact information, occupancy status, images, and rental price.

- Each result is presented in an organized format, allowing users to browse options, view property details, and contact the property manager if interested.

Property Listings

Property listings are at the heart of the Simple House Rental System, providing detailed information on available rentals. The platform includes several features to ensure comprehensive and organized property listings for both administrators and customers.

1. Property Information:

- Each property listing includes detailed information, such as property name, location, price, owner contact information, and availability status.

- Images are included with each listing to give users a visual representation of the property, enhancing the user experience by offering a clearer view of the property.

2. Fetching and Displaying Data:

- The system retrieves property information from the database, presenting it in a well-organized layout. Property details are displayed in a user-friendly format that emphasizes readability and ease of access.

- Properties can be sorted based on filters, ensuring that users find the listings most relevant to them.

3. Interactive Elements:

- Users can interact with the listings by saving properties they are interested in or by contacting property owners through contact information provided in the listing details.

Complaint Management

The Complaint Management feature is designed to foster communication between customers and administrators. It allows users to submit complaints about issues they may encounter, and admins can view, respond to, and resolve these complaints.

1. Complaint Submission for Customers:

- Registered users can lodge complaints through a dedicated form within the Customer Panel. They can describe their issues in detail, which is then submitted to the database for the admin's review.

- This feature ensures that customers have a platform to voice their concerns or report issues regarding property conditions, customer service, or system-related problems.

2. Admin's Ability to Address Complaints:

- Administrators can view complaints within the Admin Panel, which are organized by customer ID, complaint type, and submission date. Each complaint entry provides details of the issue raised, enabling the admin to understand and address it effectively.

- Once a complaint is resolved, the admin can update its status in the system, notifying the customer that their issue has been addressed.

3. Improved Communication and Transparency:

- The complaint management system fosters transparency and enhances customer satisfaction by offering a structured channel for addressing issues.

- This feature highlights the system's commitment to providing quality service, allowing administrators to maintain a responsive and customer-centric approach.

Database Design

The database design for the Simple House Rental System is structured to support efficient data management and retrieval across multiple entities, including users, properties, and complaints. The design focuses on creating a robust schema that ensures data integrity, enforces relationships between entities, and facilitates rapid access to key information for both customers and administrators.

Database Schema

The core database schema includes three primary tables: `users`, `properties`, and `complaints`. Each table is carefully designed with attributes that cater to specific requirements of the system, enabling streamlined interactions and data handling.

1. Users Table

- The `users` table stores details about both customers and administrators. Attributes include:
  - `id`: A unique identifier for each user.
  - `username`: The username chosen by the user for logging in.
  - `password`: A secure, hashed password for authentication.
  - `email`: The user's email address, used for contact and notifications.
  - `user_type`: Specifies the user's role (e.g., customer or admin).
- This table is linked to the `complaints` table to record issues submitted by each user, ensuring that complaints are associated with the correct user for tracking and resolution.

2. Properties Table

- The `properties` table contains information about rental listings. Attributes include:
  - `id`: A unique identifier for each property.

- `owner_id`: Links each property to a user who owns or manages it, creating a relationship with the `users` table.

- `location`: Specifies the property's geographical location, enhancing search functionality.

- `price`: Lists the rental price for each property.

- `description`: Provides a brief description of the property's features.

- `occupancy_status`: Indicates whether the property is currently rented or available.

- `images`: Stores the URLs or paths to images of the property for display in listings.

- The `properties` table is designed to support search and filtering by attributes like `location`, `price`, and `occupancy_status`, which aligns with the customer's need to find properties that meet specific criteria.

3. Complaints Table

- The `complaints` table records issues reported by customers. Attributes include:

- `id`: A unique identifier for each complaint.

- `user_id`: Links the complaint to the user who submitted it, referencing the `users` table.

- `property_id`: Associates the complaint with a specific property, creating a relationship with the `properties` table.

- `date`: The date the complaint was submitted, which helps track response times.

- `description`: Details the nature of the complaint.

- `status`: Tracks the resolution status of each complaint (e.g., pending, resolved).

- This table enables administrators to manage and respond to issues effectively, ensuring that each complaint is handled and resolved in a timely manner.

Entity-Relationship Diagram (ERD)

An Entity-Relationship Diagram (ERD) visually represents the relationships between the primary entities—`users`, `properties`, and `complaints`. This diagram demonstrates how each entity interacts within the system:

- Users to Properties: A one-to-many relationship, where one user (property owner) can list multiple properties. Each property references the `user_id` of its owner.

- Users to Complaints: Another one-to-many relationship, as each user may submit multiple complaints over time, and each complaint is associated with a single user.

- Properties to Complaints: A one-to-many relationship, with each property potentially having multiple complaints related to it. Each complaint entry includes a `property_id` to establish a link to the specific property in question.

This relational structure ensures that each entity is uniquely identifiable while allowing for efficient data retrieval across multiple tables. For example, an admin can quickly identify all complaints related to a particular property or view all properties listed by a specific owner.

Example Queries

The Simple House Rental System relies on SQL queries to manage and retrieve data from the database. Some essential queries are outlined below to illustrate common operations within the system.

1. Fetching Property Details

- To retrieve details about properties, including the owner's information, the system performs a join operation between the `users` and `properties` tables. This query supports displaying properties with complete owner details in the customer panel.

2. Registering a New User

- When a user registers, an `INSERT` query adds their details to the `users` table. This query ensures that each new customer or admin has a unique entry in the system, allowing for future interactions.

3. Adding a New Property

- Property listings are added to the database via an `INSERT` query in the `properties` table. This query includes details like `location`, `price`, `description`, and `owner_id`, linking the property to its owner.

4. Submitting a Complaint

- When a customer submits a complaint, the system records it in the `complaints` table. An `INSERT` query adds the complaint details, associating it with the specific user and property.

5. Filtering Properties by Location or Price

- To enable the search functionality, `SELECT` queries filter properties based on customer preferences. For example, filtering by `location` and `price` allows customers to find suitable properties without browsing all listings.

6. Updating Complaint Status

- Admins update the status of complaints as they are addressed. An `UPDATE` query modifies the `status` field in the `complaints` table, changing it from `pending` to `resolved` as necessary.

User Interface Design

The Simple House Rental System's user interface is designed with a focus on usability, responsiveness, and a clean, modern aesthetic. The layout and visual elements of the interface ensure an intuitive experience for both customers and administrators, making it easy to navigate and interact with the system's features. The UI combines well-organized page structures, effective use of Bootstrap for responsiveness, and custom CSS to create an engaging platform that meets the needs of different user groups.

UI Layout and Components

The interface for the Simple House Rental System is divided into several main pages, each with a distinct purpose and layout tailored to specific user roles.

1. Customer Panel:

- The customer panel serves as the primary access point for users who are searching for rental properties. The layout of this panel emphasizes ease of navigation, with options for browsing properties, filtering search results, and viewing individual listings.

- The main page includes a search bar and filter options that allow customers to find properties by entering keywords or selecting criteria like location and price range.

- Each property is displayed with a brief summary, including an image, name, location, and price. Clicking on a property provides more detailed information on a separate page, where users can view additional images, contact information, and occupancy status.

2. Admin Panel:

- The admin panel is structured to provide administrators with comprehensive management capabilities. Upon logging in, admins are directed to a dashboard that summarizes key information, such as the total number of properties, active complaints, and registered users.

- The panel includes side navigation links to access different sections, such as property listings, user records, and complaint management. Each section has a clear layout with tables or lists that provide an overview of all entries, making it easy for admins to access and update information as needed.

- In the property management section, admins can add new properties, edit existing ones, or change their status. The complaint section similarly allows admins to view, respond to, and update the status of customer complaints, ensuring efficient resolution.

3. Search Results Page:

- The search results page is accessible from both the customer and admin panels, with an interface that highlights properties matching the selected filters. Results are displayed in a grid or list view, depending on the screen size, to optimize readability and user experience.

- Each result includes key details, such as property name, location, and price, and features a quick preview image to give users an idea of the property's appearance. Users can click on any property to view more information or initiate contact with the owner.

4. Responsive Design with Bootstrap:

- Bootstrap's grid system and responsive utilities are extensively used across the platform to ensure that the layout adjusts seamlessly to various screen sizes. The use of Bootstrap ensures that customers and admins can access the system comfortably from desktops, tablets, and smartphones.

- Navigation bars, buttons, and forms are styled using Bootstrap classes, which contribute to a consistent look and feel while minimizing custom CSS needs. This also helps maintain a cohesive visual experience across all devices and screen orientations.

Color Scheme and Styling

The Simple House Rental System's color scheme is chosen to create a professional, inviting atmosphere that aligns with its purpose as a real estate platform. A balanced combination of neutral tones with occasional color accents gives the platform a clean and user-friendly look.

1. Color Scheme:

- The system primarily uses shades of gray, white, and light blue, giving the interface a minimalist feel. Backgrounds are often light, with subtle borders and highlights used to distinguish different sections and components.

- Key interactive elements, like buttons and links, use a slightly brighter color (e.g., blue or green) to make them stand out, ensuring that users can easily identify and interact with clickable components.

2. Typography:

- Fonts are chosen for readability, with larger font sizes used for headings and section titles to establish a visual hierarchy. Text styling, including font size and color, is optimized for contrast against the background, making content easily readable on screens of all sizes.

3. CSS Styling in `style.css` and `rent.css`:

- Custom CSS in `style.css` and `rent.css` is applied to give the platform a unique appearance while complementing Bootstrap's default styling. CSS rules are used to adjust padding, margins, borders, and other visual elements, enhancing the layout without overwhelming it.

- For instance, padding and margins are used to create a sense of space around text and images, preventing the interface from appearing crowded. Border-radius and shadow effects add subtle depth to elements like cards and buttons, contributing to a modern, polished look.

4. Example Styling Techniques:

- Card Layouts for Listings: Properties are displayed as cards, with each card containing an image, title, and brief information. Cards are styled with rounded corners and a slight shadow effect, giving them a floating appearance on the page.

- Button Styling: Buttons are styled with color gradients and hover effects that provide visual feedback when users interact with them. Hover animations are subtle, enhancing the user experience without causing distractions.

- Navigation Bar: The navigation bar is fixed at the top of the screen for easy access, with a light background and contrasting text. This design ensures consistent navigation across different sections, making it easy for users to explore the platform.

5. Responsive Adjustments:

- Media queries are used to fine-tune the layout for different screen sizes. For instance, the grid layout for property listings automatically switches from a multi-column display on larger screens to a single-column view on smaller devices, ensuring that each property remains fully visible and easy to interact with.

- Font sizes and spacing are adjusted to prevent text overflow on smaller screens, ensuring that all information is accessible without requiring excessive scrolling.

Testing & Validation

Testing Approach

The testing methodology used includes the following approaches to ensure the project meets the required standards of quality and functionality:

- Unit Testing: This testing focuses on verifying that each individual component or function performs as expected. Unit tests were conducted for various functions such as user login, property search, and complaint submission. Each function was tested in isolation to ensure it returns the correct output for different inputs.

- Functional Testing: This approach was used to test the features of the system in real-world scenarios. It aimed to verify that the application's functions, such as user registration, login, property search, and complaint submission, operate according to the requirements. Functional tests simulate the actions of an end user to check for consistency and reliability.

- Integration Testing: After individual components were unit tested, integration testing was carried out to ensure that the different components of the system work together smoothly. This involved testing the interaction between the frontend and backend, as well as the communication between the database and the application, ensuring seamless integration.

Tools Used

Several tools were employed to support the testing process:

- PHPUnit: This tool was used for unit testing PHP functions, ensuring that each function works as expected.

- MySQL Workbench: This was used to test MySQL queries, ensuring they retrieve and manipulate data correctly.

- Jest: For frontend testing, particularly React components, Jest was used to ensure that the user interface behaves as expected.

- Selenium: Selenium was used for functional testing, simulating user interactions to verify the system's behavior under real conditions.

Validation

User inputs were validated on both the client-side and server-side to ensure the security and integrity of the system:

- Client-side Validation: JavaScript was used to perform real-time validation of user inputs before they were submitted. This includes ensuring required fields are filled, verifying email formats, and preventing submission of invalid data. This improves the user experience by providing instant feedback.

- Server-side Validation: PHP was used to validate inputs before saving them to the database. This validation included sanitizing and validating the data to prevent issues like SQL injection, ensuring that user inputs such as emails, usernames, and property details were accurate and secure.

Sample Test Cases

Here are some key test cases for the core features:

1. Login Test Case

- Test: User enters valid credentials (username and password).

- Expected Result: The system authenticates the user and redirects them to the dashboard.

2. Property Search Test Case

- Test: User enters a search term (location or property type).

- Expected Result: The system displays property listings matching the search criteria.

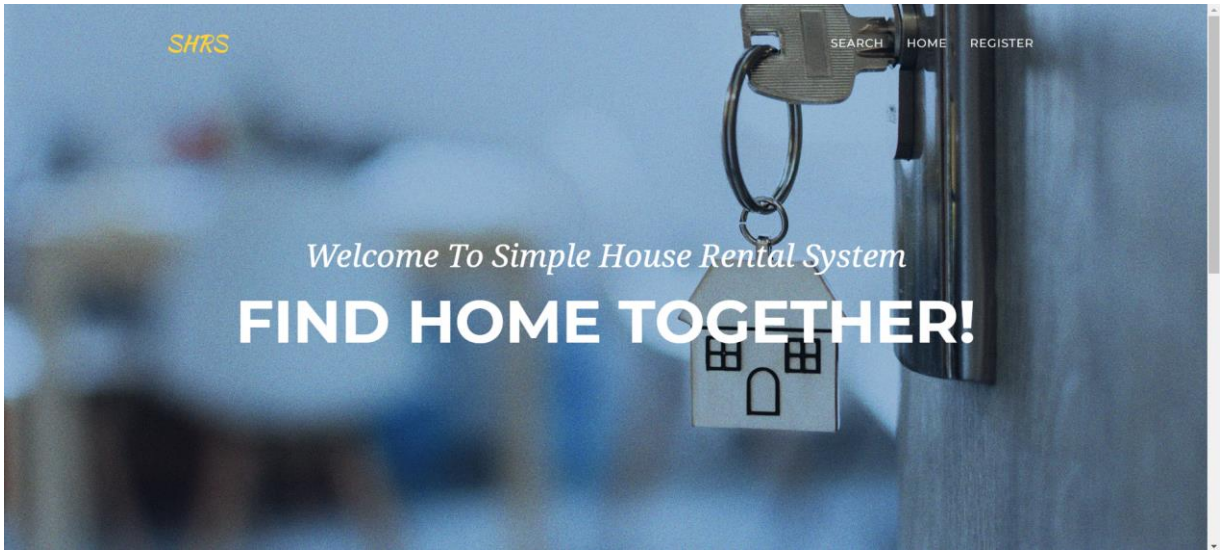
3. Complaint Submission Test Case

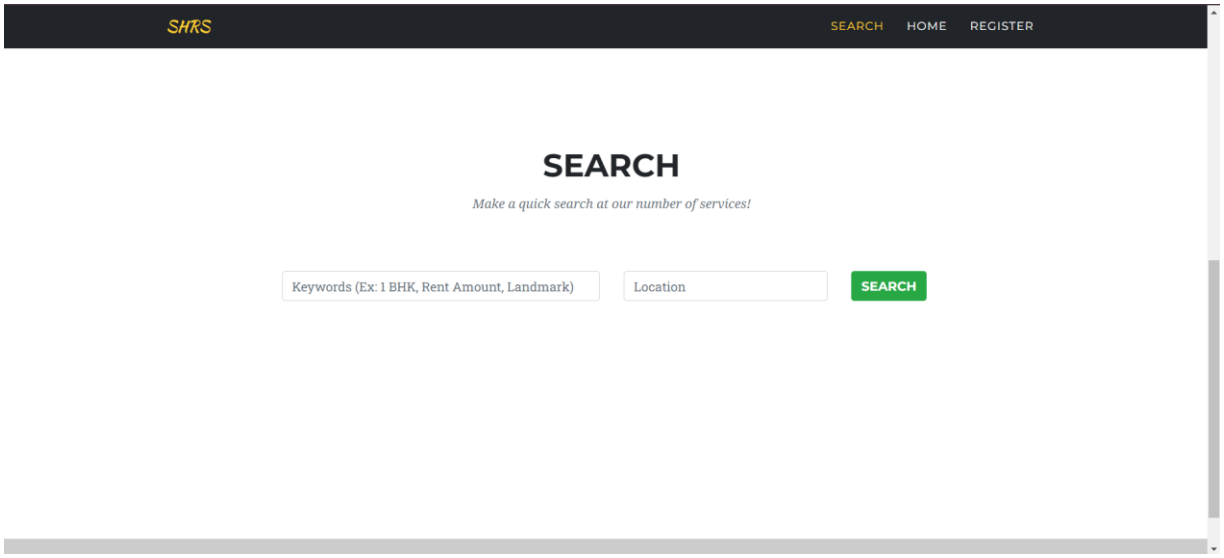
- Test: User submits a complaint with valid details.

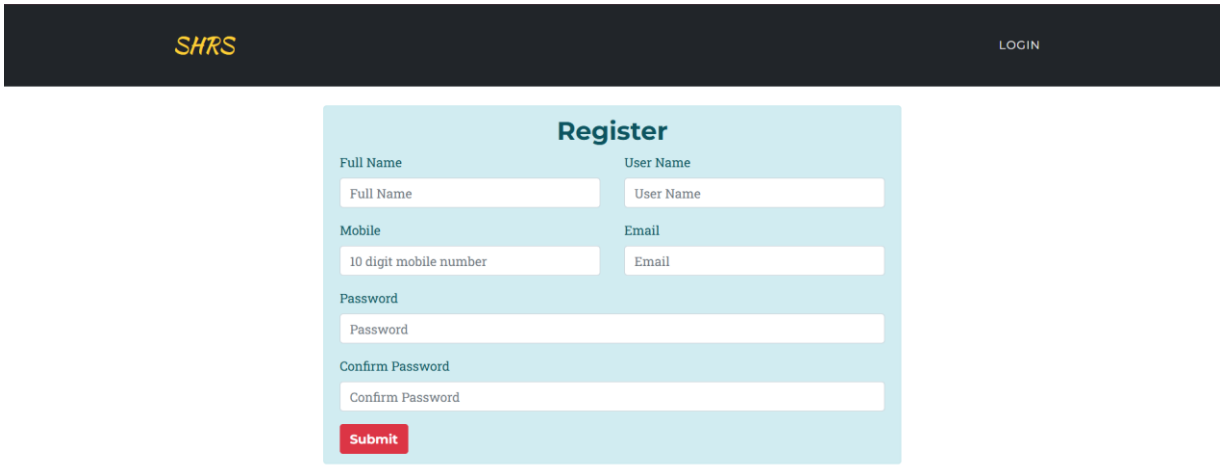
- Expected Result: The system records the complaint in the database and provides a confirmation message.

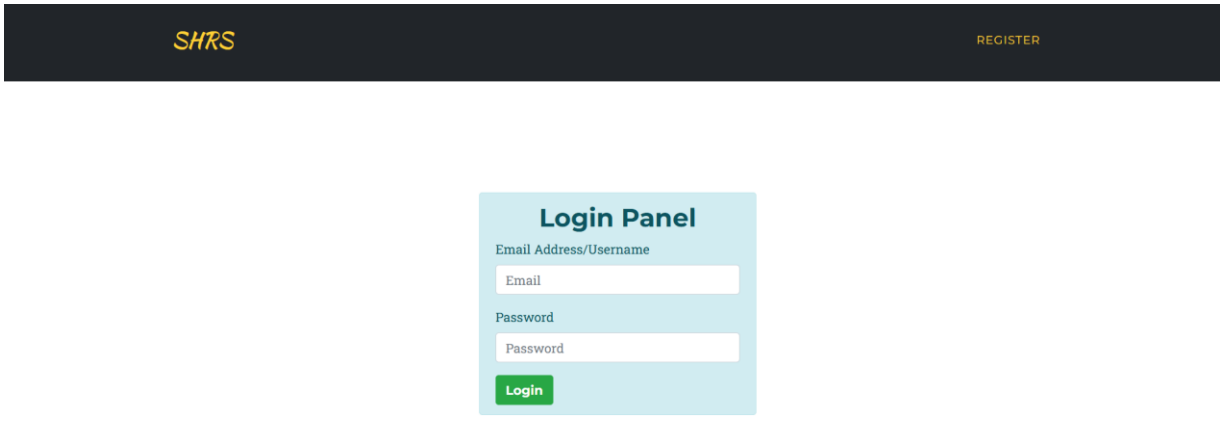
These test cases ensure that critical features like login, property search, and complaint submission are functioning correctly and provide a seamless user experience.

Some Pictures of House Rental System

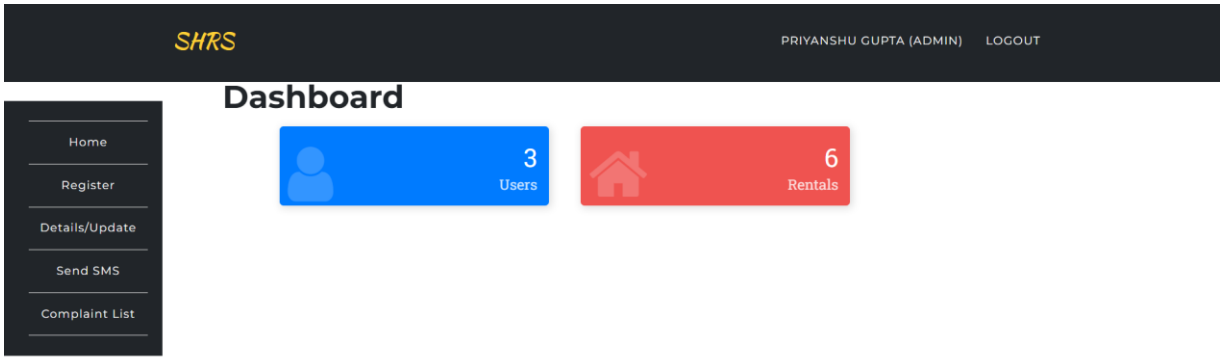


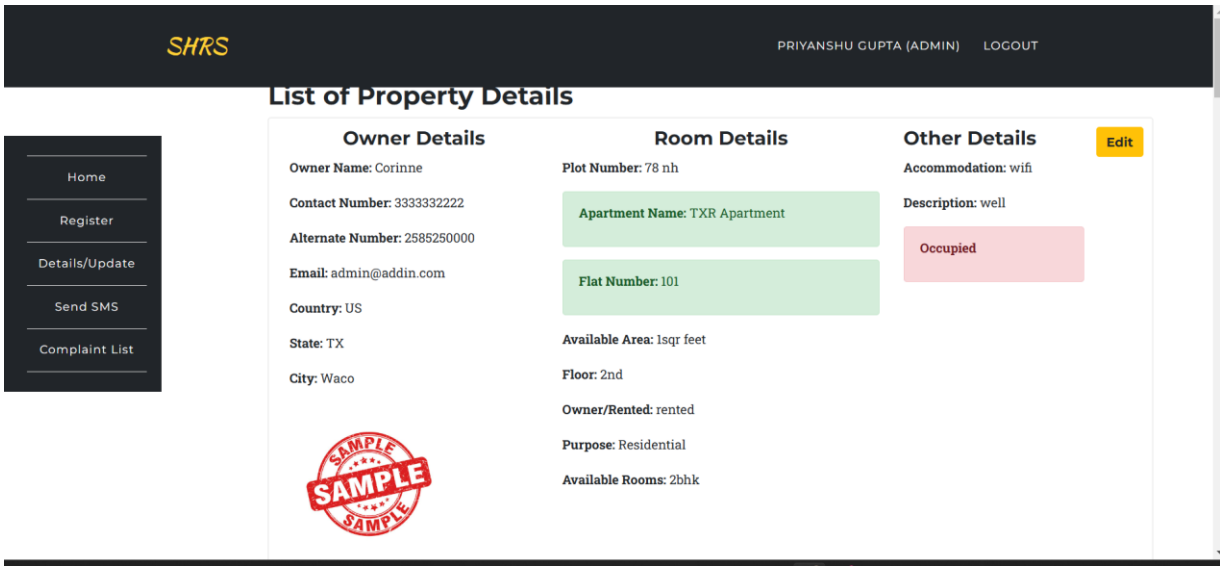






localhost/houserental/auth/register.php


SHRS
PRIYANSHU GUPTA (ADMIN)
LOGOUT
Dashboard
Home
Register
Details/Update
Send SMS
Complaint List
3
Users
6
Rentals


SHRS
PRIYANSHU GUPTA (ADMIN)
LOGOUT
List of Property Details
Home
Register
Details/Update
Send SMS
Complaint List
Owner Details
Owner Name: Corinne
Contact Number: 3333332222
Alternate Number: 2585250000
Email: admin@addin.com
Country: US
State: TX
City: Waco
SAMPLE
Room Details
Plot Number: 78 nh
Apartment Name: TXR Apartment
Flat Number: 101
Available Area: 1sqr feet
Floor: 2nd
Owner/Rented: rented
Purpose: Residential
Available Rooms: 2bhk
Other Details
Edit
Accommodation: wifi
Description: well
Occupied

Conclusion

In conclusion, the Simple House Rental System successfully meets its primary objective of streamlining rental property management. The system efficiently handles key tasks such as user registration, property listing, search functionality, and complaint submission, offering a user-friendly interface for both tenants and property owners. This project not only fulfills the immediate needs of rental management but also serves as a solid foundation for future expansion.

Throughout the development process, the project provided a valuable learning experience, particularly in working with PHP and MySQL. It allowed for the implementation of key concepts such as database management, form validation, and server-client interactions, as well as integration between different system components. These practical skills will prove beneficial in future web development endeavors.

The system is designed to be scalable, allowing for easy enhancements in the future. For example, integrating payment processing capabilities could further simplify the rental process for both tenants and property owners. Additionally, expanding the search functionality with more advanced filters or incorporating features such as user reviews and ratings would enhance the overall user experience.

Overall, the Simple House Rental System is a robust, functional solution that addresses the needs of property management while also providing a valuable platform for further innovation and improvement.