**PROJECT REPORT**

**ON**

**APARTMENT MANAGEMENT SYSTEM**

**BY**

**FAJAR QADAR ROLL NO. 170**

**MUHAMMAD TALEEB ROLL NO. 218**

**GHAYOUR AHMAD ROLL NO. 287**

**SESSION: 2019-2023**

*A project report submitted to The University of Agriculture, Peshawar in Partial Fulfillment of the requirement for the degree of*

**BACHELOR OF SCIENCE IN COMPUTER SCIENCE**



**SUPERVISED BY**

**Mr. KAMRAN ULLAH**

**INSTITUTE OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**

**FACULTY OF MANAGEMENT AND COMPUTER SCIENCE**

**THE UNIVERSITY OF AGRICULTURE**

**PESHAWAR-PAKISTAN**

**JUNE, 2023**

**PROJECT REPORT**

**ON**

**APARTMENT MANAGEMENT SYSTEM**

**BY**

**FAJAR QADAR ROLL NO. 170**

**MUHAMMAD TALEEB ROLL NO. 218**

**GHAYOUR AHMAD ROLL NO. 287**

**SESSION: 2019-2023**

*A project report submitted to The University of Agriculture, Peshawar in Partial Fulfillment of the requirement for the degree of*

**BACHELOR OF SCIENCE IN COMPUTER SCIENCE**

**Approved By:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Advisor

Mr. Kamran Ullah

Lecturer ICS/IT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Member

Lecturer ICS/IT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Member

Lecturer ICS/IT

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Asfandyar Khan HoD, ICS/IT

**INSTITUTE OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**

**FACULTY OF MANAGEMENT AND COMPUTER SCIENCE**

**THE UNIVERSITY OF AGRICULTURE**

**PESHAWAR-PAKISTAN**

**JUNE, 2023**

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

[ACKNOWLEDGEMENT i](#_Toc137501217)

[LIST OF FIGURES ii](#_Toc137501218)

[CHAPTER 1 1](#_Toc137501219)

[INTRODUCTION 1](#_Toc137501220)

[1.1 Project Goal and Aims 1](#_Toc137501221)

[1.2 Objective 3](#_Toc137501222)

[CHAPTER 2 4](#_Toc137501223)

[EXISTING SYSTEMS 4](#_Toc137501224)

[2.1 Drawbacks of existing systems 5](#_Toc137501225)

[2.2 Proposed System 6](#_Toc137501226)

[2.3 Features of Apartment Management System 8](#_Toc137501227)

[CHAPTER 3 10](#_Toc137501228)

[SOFTWARE REQUIREMENTS 10](#_Toc137501229)

[3.1 Development Tools 10](#_Toc137501230)

[3.1.1 PHP 10](#_Toc137501231)

[3.1.2 MySQL 10](#_Toc137501232)

[3.1.3 HTML 11](#_Toc137501233)

[3.1.4 CSS 11](#_Toc137501234)

[3.1.5 Bootstrap 12](#_Toc137501235)

[3.1.6 jQuery 12](#_Toc137501236)

[3.2 Entity Relation Diagram 13](#_Toc137501237)

[3.3 Login Page 14](#_Toc137501238)

[CHAPTER 4 15](#_Toc137501239)

[METHODOLOGY 15](#_Toc137501240)

[4.1 Login page 15](#_Toc137501241)

[4.2 Super Admin 16](#_Toc137501242)

[4.3 Admin 16](#_Toc137501243)

[4.4 Owner 16](#_Toc137501244)

[4.5 Employee 17](#_Toc137501245)

[4.6 Tenant 17](#_Toc137501246)

[4.7 Super admin dashboard 18](#_Toc137501247)

[4.8 Super Admin dashboard 40](#_Toc137501248)

[4.9 Admin dashboard 40](#_Toc137501249)

[4.10 Owner Dashboard 41](#_Toc137501250)

[4.11 Employee dashboard 42](#_Toc137501251)

[4.12 Tenant dashboard 42](#_Toc137501252)

[CHAPTER 5 44](#_Toc137501253)

[CONCLUSION 44](#_Toc137501254)

# ACKNOWLEDGEMENT

I would like to express my deepest gratitude and appreciation to all those who have contributed to the successful completion of this project.

First and foremost, I extend my heartfelt thanks to my project supervisor for their guidance, support, and valuable insights throughout the entire duration of the project. Their expertise and encouragement have been instrumental in shaping the project's direction and ensuring its successful execution.

I am immensely grateful to the members of my project team who dedicated their time and effort to the project, collaborating and contributing their unique skills and perspectives. Their unwavering commitment and hard work have been essential in achieving our project goals.

I would also like to acknowledge the support and cooperation received from all the participants involved in this study. Their willingness to provide information, share their experiences, and offer their expertise has been invaluable in enriching the project's findings.

My sincere appreciation extends to my friends and family for their unwavering support and understanding throughout this project. Their encouragement and belief in me have been a constant source of motivation.

Finally, I would like to express my gratitude to the institution and all the resources provided, which have played a crucial role in the successful completion of this project.

Without the collective effort and support from all these individuals and entities, this project would not have been possible. I am truly grateful for their contributions and am honored to have had the opportunity to work with such remarkable individuals.

Thank you all once again for being an integral part of this journey and for making this project a reality.

# LIST OF FIGURES

**FIGURE NO. TITLE PAGE NO.**

[4.1 Login page 15](#_Toc137501056)

[4.2 Super Admin Dashboard 18](#_Toc137501057)

[4.3 Add Admin 18](#_Toc137501058)

[4.4 Admin List 19](#_Toc137501059)

[4.5 Add Floor 19](#_Toc137501060)

[4.6 Floor List 20](#_Toc137501061)

[4.7 Add Unit 21](#_Toc137501062)

[4.8 Unit List 22](#_Toc137501063)

[4.9 Add Owner 23](#_Toc137501064)

[4.10 Owner List 24](#_Toc137501065)

[4.11 Add Owner Utility 25](#_Toc137501066)

[4.12 Rent List 26](#_Toc137501068)

[4.13 Add Tenant 27](#_Toc137501069)

[4.14 Tenant List 28](#_Toc137501070)

[4.15 Add Rent 29](#_Toc137501071)

[4.16 Rent List 30](#_Toc137501073)

[4.17 Add Tenant Payment 31](#_Toc137501074)

[4.18 Tenant Invoice 32](#_Toc137501075)

[4.19 Add Employee 33](#_Toc137501076)

[4.20 Employee List 33](#_Toc137501077)

[4.21 Add Employee Salary 34](#_Toc137501078)

[4.22 Employee Salary List 34](#_Toc137501079)

[4.23 Maintenance Cost 35](#_Toc137501080)

[4.24 Maintenance Cost List 35](#_Toc137501081)

[4.25 Add Bill Deposit 36](#_Toc137501082)

[4.26 Bill Deposit List 36](#_Toc137501083)

[4.27 Add Visitor 37](#_Toc137501084)

[4.28 Visitor List 37](#_Toc137501085)

[4.29 Complaint List 38](#_Toc137501086)

[4.30 Notice Board 38](#_Toc137501087)

[4.31 Notice List 39](#_Toc137501088)

[4.32 View Notice Board 39](#_Toc137501089)

[4.33 Super Admin Dashboard 40](#_Toc137501090)

[4.34 Admin Dashboard 40](#_Toc137501091)

[4.35 Owner Dashboard 41](#_Toc137501092)

[4.36 Employee Dashboard 42](#_Toc137501093)

[4.37 Tenant Dashboard 43](#_Toc137501094)

# CHAPTER 1

# INTRODUCTION

The **Apartment Management System** to provide a comprehensive solution for residents of apartments to conveniently pay bills, make complaints, and update their information through an online platform. This system aims to enhance effectiveness, efficiency, and interactivity, addressing the limitations and challenges of the existing manual systems.

Currently, apartment residents face various difficulties in managing their bills and communicating with management. Traditional methods require residents to physically visit the management office to make payments, submit complaints, or update their personal details. This manual approach can be time-consuming, inconvenient, and prone to errors.

To overcome these limitations, the Apartment Management System project utilizes PHP as the primary programming language to develop a web-based platform that streamlines these processes. By providing an online interface, residents can conveniently access the system from anywhere and perform a range of tasks related to bill payments, complaint submission, and personal information updates.

## 1.1 Project Goal and Aims

The primary goal of the Apartment Management System (AMS) project is to create an efficient and user-friendly platform that simplifies various management tasks related to apartment living. The specific project goals of the AMS can include:

1. Streamline Bill Management: The AMS aims to automate the billing process for apartment residents. It provides a convenient way for residents to view and pay their bills online, eliminating the need for manual payment methods. The goal is to simplify the bill management process, reduce errors, and ensure timely payments.

2. Enhance Complaint Management: The AMS aims to improve the process of registering and resolving complaints for residents. It provides a centralized platform for residents to submit complaints, specify their urgency, and provide relevant details. The goal is to streamline the complaint management process, facilitate prompt response from management, and ensure efficient complaint resolution.

3. Facilitate Tenant Information Management: The AMS aims to simplify the management of tenant information. It allows residents to update their personal details, contact information, and other relevant information easily. The goal is to ensure accurate and up-to-date records, which aids effective communication between residents and management.

4. Improve Communication Channels: The AMS aims to enhance communication between residents and apartment management through a user-friendly notice board feature. The notice board serves as a platform for the management to post important announcements, notices, and updates, while residents can easily access and view this information.

5. Increase Efficiency and Convenience: The AMS aims to streamline various administrative tasks related to apartment management. By providing an online platform, it reduces manual efforts, minimizes paperwork, and improves overall efficiency. The goal is to create a system that saves time and effort for both residents and management, leading to a more convenient experience.

6. Ensure Data Security and Privacy: The AMS aims to prioritize data security and privacy. It incorporates appropriate security measures to protect sensitive information of residents, such as financial data and personal details. The goal is to ensure that resident data is stored and transmitted securely, building trust and confidence among users.

Overall, the project goal of the Apartment Management System is to develop a comprehensive solution that simplifies apartment management tasks, enhances communication, and improves the overall living experience for residents. By achieving these goals, the AMS aims to contribute to the efficient and effective management of apartment complexes.

## 1.2 Objective

The objectives of the Apartment Management System (AMS) can Facilitate Management Decision-making: The AMS can aim to provide management with valuable insights and data analytics to support decision-making processes. The objective is to generate reports, analytics, and trends related to billing, complaints and overall, the objectives of the Apartment Management System revolve around improving efficiency, enhancing communication, and simplifying various administrative processes related to apartment management. By achieving these objectives, the AMS aims to create a more organized, convenient, and transparent environment for both residents and management.

# CHAPTER 2

# EXISTING SYSTEMS

When implementing an Apartment Management System (AMS), it's important to consider the existing systems and processes in place. Understanding the strengths and limitations of these systems can help in designing an effective AMS. Here are some common existing systems in apartment management:

1. Manual Paper-Based Systems: Many apartment complexes still rely on manual paper-based systems for various tasks. These systems involve physical paperwork for bill generation, bill payments, complaint registration, and tenant information management. They can be time-consuming, error-prone, and inefficient, often requiring residents to visit the management office in person.

2. Spreadsheet-Based Systems: Some apartment complexes use spreadsheets to manage tenant information, bill records, and maintenance schedules. While spreadsheets provide a digital alternative to paper-based systems, they can still be limited in terms of functionality, scalability, and real-time data access. Collaboration and data integrity can also be challenging to maintain.

3. Email and Phone Communication: Apartment management often relies on email and phone communication to interact with residents. Residents may send emails or make phone calls to raise complaints, inquire about bills, or update their information. However, managing and tracking these communications can be cumbersome, and there may be delays in response times.

4. Physical Notice Boards: Many apartment complexes utilize physical notice boards to post announcements, event schedules, and other important information. While notice boards serve the purpose of disseminating information, they are limited in terms of reach, visibility, and timely updates. Residents may miss important notices, especially if they do not regularly visit common areas.

5. Independent Payment Gateways: Some apartment complexes have independent payment gateways or systems in place to handle rent and utility bill payments. These systems often require residents to log in to a separate platform to make payments. Integration with these existing payment gateways can be beneficial to provide a seamless payment experience within the AMS.

When designing the AMS, it is essential to consider how these existing systems can be integrated, replaced, or enhanced to improve efficiency, communication, and resident experience. The AMS should aim to automate manual processes, provide a centralized platform for communication and information sharing, and integrate with existing systems where applicable.

## 2.1 Drawbacks of existing systems

Existing systems in apartment management may have several drawbacks that can hinder efficient operations and user experience. Here are some common drawbacks associated with existing systems:

1. Manual Processes: Many existing systems rely heavily on manual processes, such as paper-based documentation, manual data entry, and physical record-keeping. This can lead to errors, inefficiencies, and increased administrative burden.

2. Lack of Integration: Existing systems may operate in silos, with limited or no integration between different functionalities or modules. This can result in data duplication, inconsistent information, and the need for manual data transfer between systems.

3. Limited Accessibility: Some systems may lack user-friendly interfaces or mobile accessibility, making it difficult for users to access and use the system on various devices. This can create barriers to adoption and hinder remote access for property managers and residents.

4. Inefficient Communication: Communication between property managers, tenants, and maintenance staff may rely on manual methods such as phone calls, emails, or physical notices. This can lead to delays, miscommunication, and difficulties in tracking and resolving issues.

5. Complex User Interfaces: Existing systems may have complex and outdated user interfaces, requiring extensive training or technical knowledge to operate effectively. This can result in a steep learning curve and decreased user satisfaction.

6. Limited Reporting and Analytics: Some systems lack robust reporting and analytics capabilities, making it challenging to generate meaningful insights and track key performance indicators. This can hinder data-driven decision-making and proactive management.

7. Security Risks: Outdated or poorly maintained systems may have security vulnerabilities, putting sensitive data at risk of unauthorized access or data breaches. This can lead to privacy concerns and potential legal and financial repercussions.

8. Scalability Issues: Existing systems may struggle to accommodate the growth and changing needs of apartment complexes. Scalability limitations can result in performance issues, slower response times, and difficulties in managing larger portfolios or multiple properties.

Addressing these drawbacks often requires considering a new or upgraded system that addresses the specific pain points and requirements of the apartment management process. A modern, integrated, and user-friendly system can improve efficiency, enhance communication, and provide better insights for decision-making.

## 2.2 Proposed System

The proposed system aims to address the drawbacks of existing apartment management systems by introducing an efficient and user-friendly solution. Here are the key features and benefits of the proposed system:

1. Integrated Platform: The proposed system will provide a centralized and integrated platform that combines various functionalities of apartment management, including tenant management, rent collection, maintenance tracking, communication, and reporting. This integration eliminates the need for multiple systems and ensures seamless data flow between different modules.

2. Automation of Processes: The proposed system will automate manual processes, reducing human errors, and improving efficiency. Tasks such as rent calculation, invoice generation, maintenance requests, and communication can be automated, saving time and effort for property managers and tenants.

3. User-Friendly Interface: The system will have a user-friendly interface that is easy to navigate and understand. Intuitive features and clear workflows will ensure a smooth user experience, reducing the learning curve and enhancing user adoption.

4. Mobile Accessibility: The proposed system will be accessible through mobile devices, enabling property managers and tenants to access information, submit requests, and communicate on the go. Mobile apps or responsive web interfaces will provide convenience and flexibility.

5. Real-time Updates and Notifications: The system will provide real-time updates and notifications to keep property managers, tenants, and maintenance staff informed about important events, such as rent due dates, maintenance schedules, and announcements. This ensures timely actions and enhances communication.

6. Robust Reporting and Analytics: The proposed system will offer comprehensive reporting and analytics capabilities. Property managers will be able to generate reports on rent payments, occupancy rates, maintenance history, and financial performance. Data-driven insights will support informed decision-making and strategic planning.

7. Enhanced Communication Channels: The system will provide effective communication channels for property managers, tenants, and maintenance staff. Features such as in-app messaging, email notifications, and a centralized notice board will streamline communication, ensuring prompt responses and reducing miscommunication.

8. Security and Data Privacy: The proposed system will prioritize security measures to protect sensitive data and ensure compliance with privacy regulations. Robust authentication, data encryption, and regular system audits will be implemented to maintain the security of the system.

9. Scalability and Customization: The system will be designed to accommodate the scalability needs of growing apartment complexes. It will be flexible and customizable, allowing property managers to tailor the system to their specific requirements and easily adapt to changing needs.

By implementing the proposed system, apartment management processes will become more streamlined, efficient, and transparent. Property managers will have better control over operations, tenants will experience improved convenience and communication, and overall management tasks will be simplified, resulting in enhanced resident satisfaction and optimized property management.

## 2.3 Features of Apartment Management System

* Five steps dashboard SUPER ADMIN, ADMIN, TENANT, EMPLOYEE, OWNER.
* Super admin can create multiple admin and assign Tenant for each Apartment.
* Super admin can create manage all the admin panel and can update delete all the files.
* Super admin can add the owner and give the apartment to the owner through lease amount.
* Admin can manage full apartment.
* Admin can manage all the apartment but cannot add the new admin the admin can be added just by the super admin
* Employee can check salary, manage complain and notice board.
* Tenant can check rent statement, manage complain and notice board.
* Owner can see just the apartment related with them. And see the rent of that units related with it
* Fully responsive.
* Suitable for all platforms.
* Extremely flexible interface.
* Admin can show notice to employee, tenant and owners.
* Admin can assign employee to the complaint.
* Admin rent due and paid invoice.
* Admin create invoice.
* Admin quick available room checking.
* Everyone can see their profile
* Everyone can change their password but must need the old password
* Whole apartment management system at your fingertips.
* Smooth operation, user friendly interface, full of functionality and compatible features.
* Expandable, customizable and fully supported by us!
* Full of features.

# CHAPTER 3

# SOFTWARE REQUIREMENTS

Web Server (Apache)

Php advance version

Xampp server

MySQL database

Any web browser

## 3.1 Development Tools

In the development of the apartment management system, various tools and technologies were utilized to build different aspects of the project. This chapter outlines the key development tools employed and provides an overview of their features and functionalities.

## 3.1.1 PHP

PHP (Hypertext Preprocessor) is a popular server-side scripting language widely used for web development. It was chosen as the primary backend programming language for the apartment management system. Some features of PHP include:

- Easy integration with HTML: PHP can be embedded within HTML code, allowing dynamic content generation and seamless integration with frontend elements.

- Extensive database support: PHP provides robust database connectivity and supports multiple database management systems, including MySQL, which was used in this project.

## 3.1.2 MySQL

MySQL is an open-source relational database management system (RDBMS) used to store and manage the project's data. It provides a robust and scalable database solution for handling various aspects of apartment management. Key features of MySQL include:

- Relational database management: MySQL offers efficient storage and retrieval of structured data through tables, relationships, and SQL queries.

- ACID compliance: MySQL ensures data integrity and transactional consistency by adhering to ACID (Atomicity, Consistency, Isolation, Durability) properties.

- Scalability and performance: MySQL are known for its scalability, allowing the management of large volumes of data with high performance and minimal latency.

- Security features: MySQL provides authentication, access control, and encryption mechanisms to protect sensitive data.

## 3.1.3 HTML

HTML (Hypertext Markup Language) is the standard markup language used for creating the structure and content of web pages. HTML is essential for building the frontend elements of the apartment management system. Some features of HTML include:

- Structure and semantics: HTML provides a structured format for organizing web content using elements like headings, paragraphs, lists, tables, and forms.

- Hyperlinking: HTML enables the creation of hyperlinks to navigate between pages and establish connections within the system.

- Support for multimedia: HTML supports the inclusion of images, videos, audio files, and other media elements within web pages.

## 3.1.4 CSS

CSS (Cascading Style Sheets) is a stylesheet language used for describing the presentation and visual styling of HTML elements. CSS enhances the appearance and layout of the apartment management system's frontend. Some features of CSS include:

- Selective styling: CSS allows developers to apply specific styles to individual HTML elements or groups of elements.

- Layout control: CSS provides mechanisms for defining the positioning, sizing, and arrangement of elements on web pages.

- Responsive design: CSS facilitates the creation of responsive and mobile-friendly interfaces that adapt to different screen sizes and devices.

## 3.1.5 Bootstrap

Bootstrap is a popular front-end framework that simplifies web development by providing pre-built responsive components and CSS styles. It was utilized in the project to ensure a consistent and visually appealing user interface. Bootstrap features include:

- Responsive grid system: Bootstrap's grid system enables the creation of responsive layouts that automatically adapt to different screen sizes.

- UI components: Bootstrap offers a wide range of pre-styled components such as buttons, forms, navigation menus, modals, and tooltips, which can be easily customized and integrated into the project.

- JavaScript plugins: Bootstrap includes a collection of JavaScript plugins that enhance interactivity and add functionality to the frontend.

## 3.1.6 jQuery

jQuery is a fast and lightweight JavaScript library that simplifies HTML document traversal, event handling, and animation. It was used in the project for client-side validation and dynamic interactions. Key features of jQuery include:

- DOM manipulation: jQuery

## 3.2 Entity Relation Diagram

## 3.3 Login Page

The login page of the Apartment Management System (AMS). It handles the submission of the login form and performs the login process based on the selected designation (super admin, admin, owner, tenant, or employee).

When the form is submitted, the code retrieves the username, password, and designation from the form fields. It then validates and sanitizes the input to prevent security vulnerabilities.

Based on the selected designation, the code queries the respective database table to check if the entered username and password match any records. If a match is found, the user's information is stored in session variables for future use.

If the login is successful, the user is redirected to the index page. Otherwise, if the designation is not selected correctly or the login credentials do not match, an appropriate error message is displayed.

The HTML part of the code defines the structure of the login form using Bootstrap classes. It includes input fields for username, password, and designation, along with options for "Remember Me" and "Forgotten Password?".

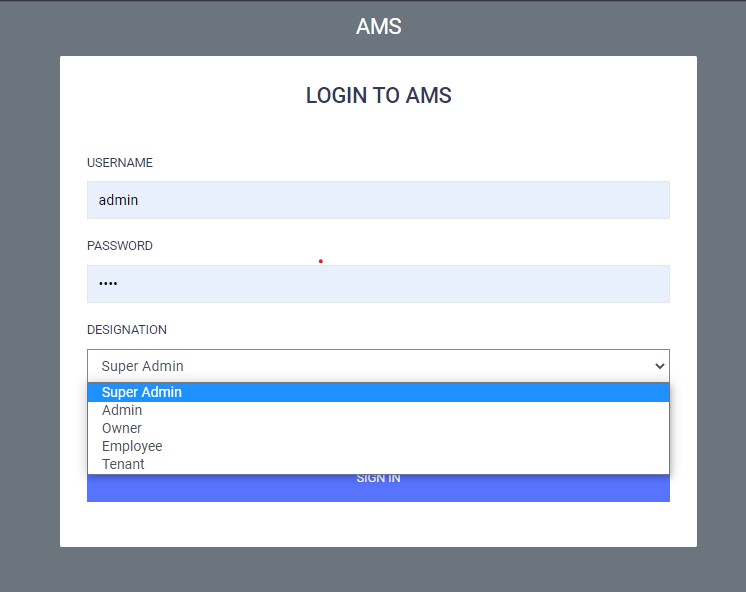
The code also includes JavaScript libraries for form validation using jQuery and the jQuery Validation Plugin. It references an external JavaScript file for custom validation rules.

In summary, the code handles the login process of AMS, validates user input, checks credentials against the database, and stores user information in session variables upon successful login.

# CHAPTER 4

# METHODOLOGY

## 4.1 Login page



### Fig. 4.1 Login page

## 4.2 Super Admin

This is the login page of the Apartment Management System (AMS), which provides five designations for users to select from. When the user chooses the "Super Admin" designation and enters the correct username and password for the super admin, the page redirects them from the login page to the index page. The index page serves as the main dashboard for the super admin, offering complete control over all system functionalities.

## 4.3 Admin

When the user selects the "Admin" designation on the login page and enters the correct username and password, they are granted access to the admin dashboard. From the admin dashboard, they can manage all system functionalities. However, the main difference between the super admin and admin roles is that the super admin has the ability to add multiple admins, whereas the admin cannot add more admins. Apart from this distinction, both the admin and super admin have the same privileges.

## 4.4 Owner

When the user selects the "Owner" designation on the login page and enter the correct username and password, they are granted access to the Owner dashboard. Here in the owner dashboard the owner can apply the complaint and check the notice board and also see the rent amount of the units which are related to the owner but cannot see the information of the other units When the user selects the "Owner" designation on the login page and enters the correct username and password, they are granted access to the Owner dashboard. In the owner dashboard, owners can apply complaints, check the notice board, and view the rent amount of the units that are specifically associated with them. However, they do not have access to the information of other units or properties. The dashboard is designed to provide owners with relevant information and functionalities related to their own units or properties only.

## 4.5 Employee

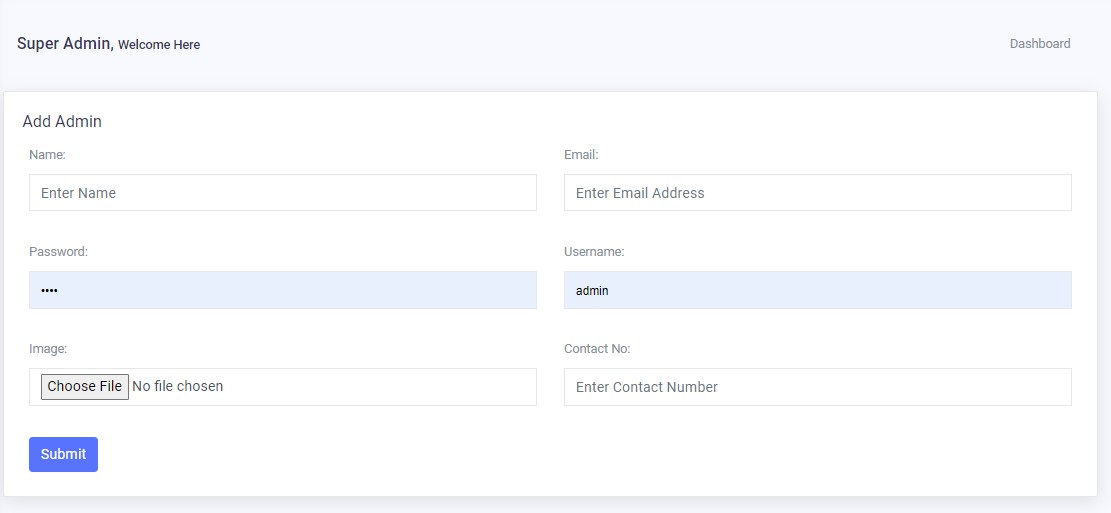
When the user selects the "Employee" designation on the login page and enters the correct username and password, they are granted access to the Employee dashboard. In the Employee dashboard, Employee can check complaints, check the notice board, and view their salary amount.

## 4.6 Tenant

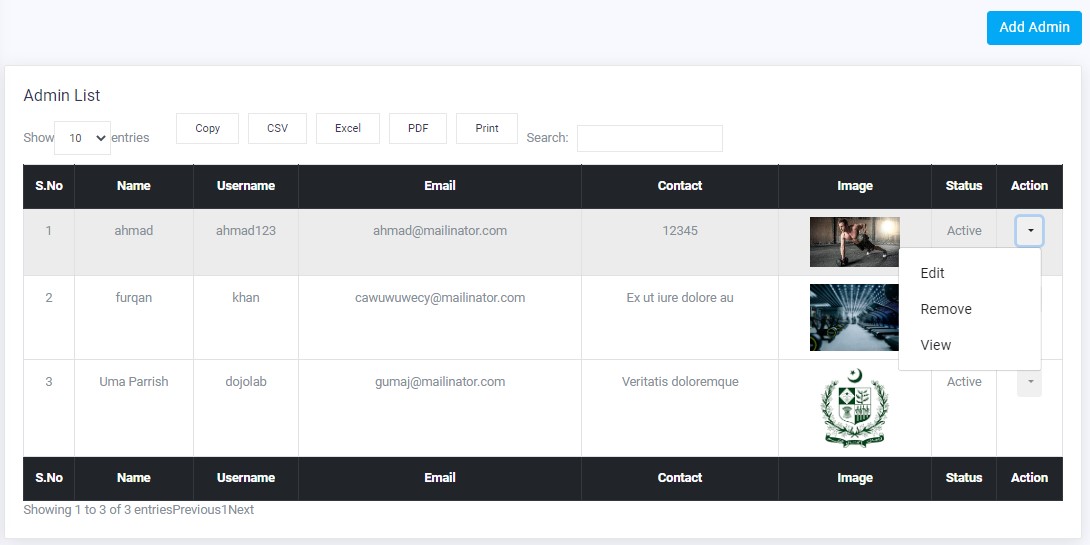
When the user selects the "Tenant" designation on the login page and enters the correct username and password, they are granted access to the Tenant dashboard. In the Tenant dashboard, tenants can submit complaints, check the notice board, and view the rent amount of the units that are specifically assigned to them. The dashboard is designed to provide tenants with relevant information and functionalities related to their own assigned units.

## 4.7 Super admin dashboard

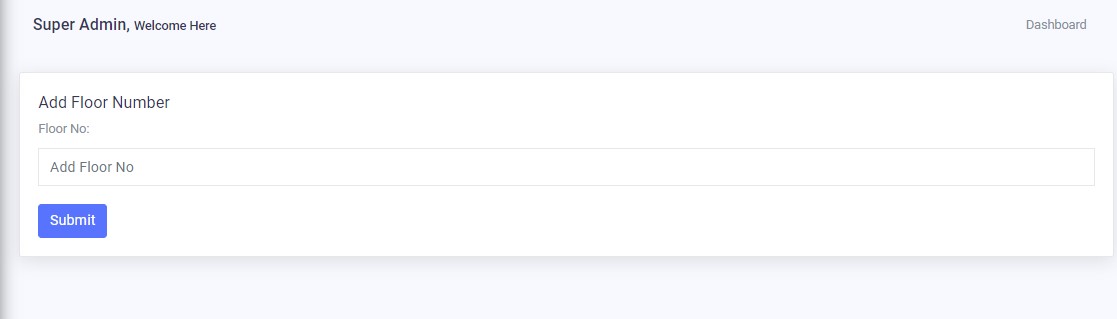
### Fig. 4.2 Super Admin Dashboard

This is the "Add Admin" page. From this form, the super admin can add a new admin. The form allows the super admin to enter the necessary details and submit them to create a new admin account.

### Fig. 4.3 Add Admin

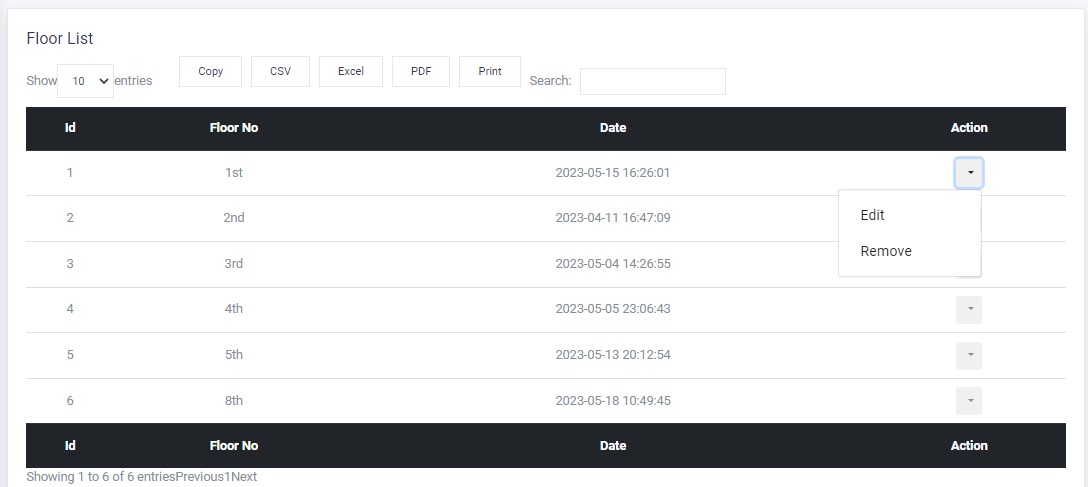
This is the "Admin List" page. Here, the super admin can view a list of admins along with their data. The super admin has the ability to perform various actions on each admin entry. By clicking on the "Edit" action, the super admin can update the admin's data. Clicking on the "Remove" action allows the super admin to delete the admin's data. Lastly, clicking on the "View" action enables the super admin to see the detailed information of the selected admin entry.

### Fig. 4.4 Admin List

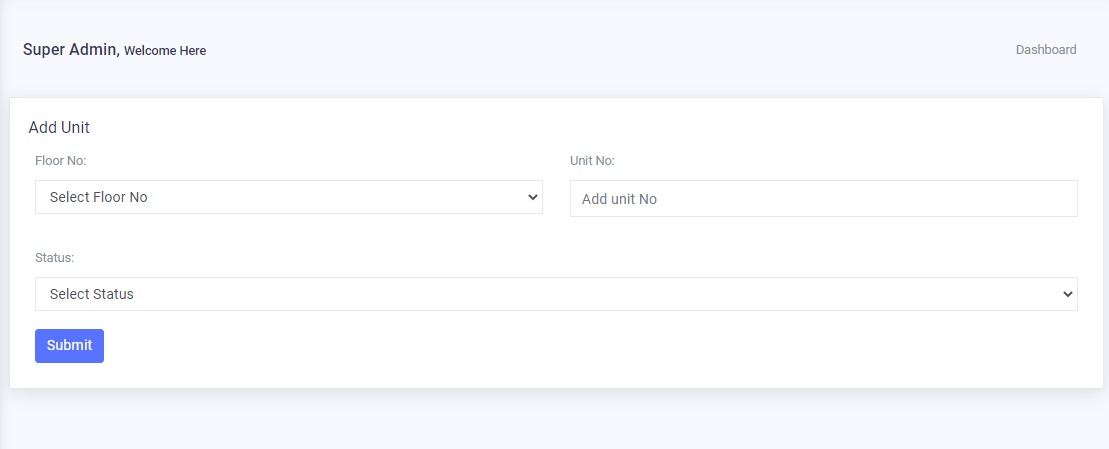
This is the "Add Floor" form. From here, we can add a new floor for the units. However, there is a validation in place to ensure that the same floor cannot be added again. In other words, each floor number can only be added once, and duplicate data cannot be inserted.

### Fig. 4.5 Add Floor

This is the "Floor List" page where both the admin and super admin can manage the floors. They can view the list of added floors, edit the floor details, and also delete the floor if needed.

However, if the floor is connected to child tables or units, the data cannot be deleted directly. First, the floor needs to be unattached from the child tables or units, and then the floor can be deleted.

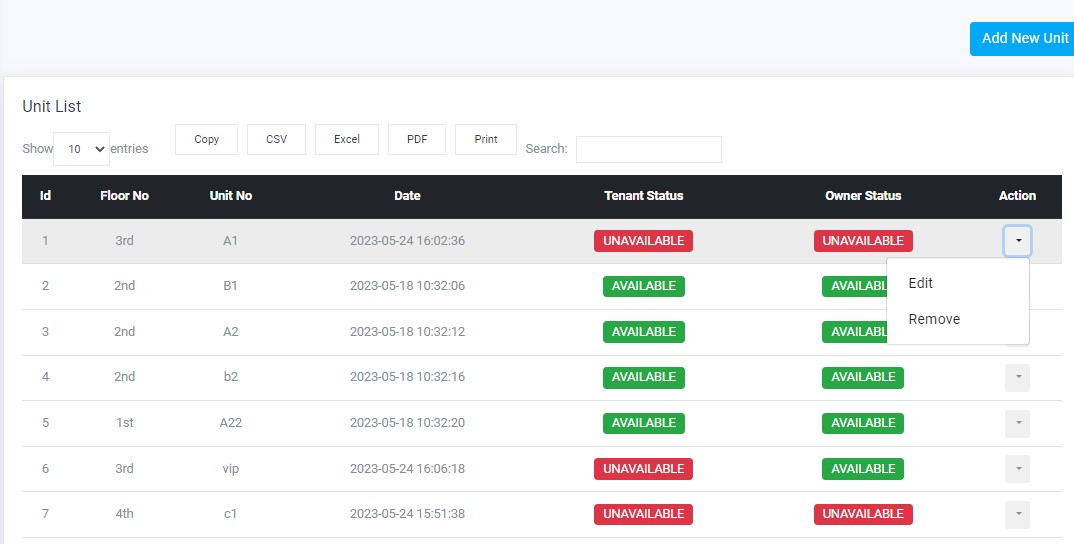
### Fig. 4.6 Floor List

This is the "Add Unit" form. Both the admin and super admin can use this form to add units for tenants and owners. In this form, we have implemented validation to prevent duplicate entries. The system checks if the same data (unit) is already inserted, and it prohibits inserting duplicate entries.

### Fig. 4.7 Add Unit

This is the "Unit List" page where the admin and super admin can manage the list of units. Each unit has two statuses: "Tenant" status and "Owner" status. When the super admin or admin assigns a unit to a tenant, the status changes from "Available" to "Unavailable". Similarly, when a unit is assigned to an owner, the status is updated from "Available" to "Unavailable".

On the unit list page, there are two buttons for each unit: "Edit" and "Remove". When the admin or super admin clicks on the "Edit" button, they can edit the unit's data. If they click on the "Remove" button, they can delete the unit's data from the system.

However, if the units are connected to child tables such as Owner Utility and Tenant, the data cannot be deleted directly. First, the units need to be unattached from the child tables, specifically the owner utility and tenant tables. Once the units are unattached from these tables, the unit can be deleted.

### Fig. 4.8 Unit List

This is the "Add Owner" form. From this form, the super admin and admin can add a new owner to the system. They can enter the owner's details, such as username, password, name, email, contact information, and any other relevant information. After filling out the form, the admin or super admin can submit it to add the new owner to the system.

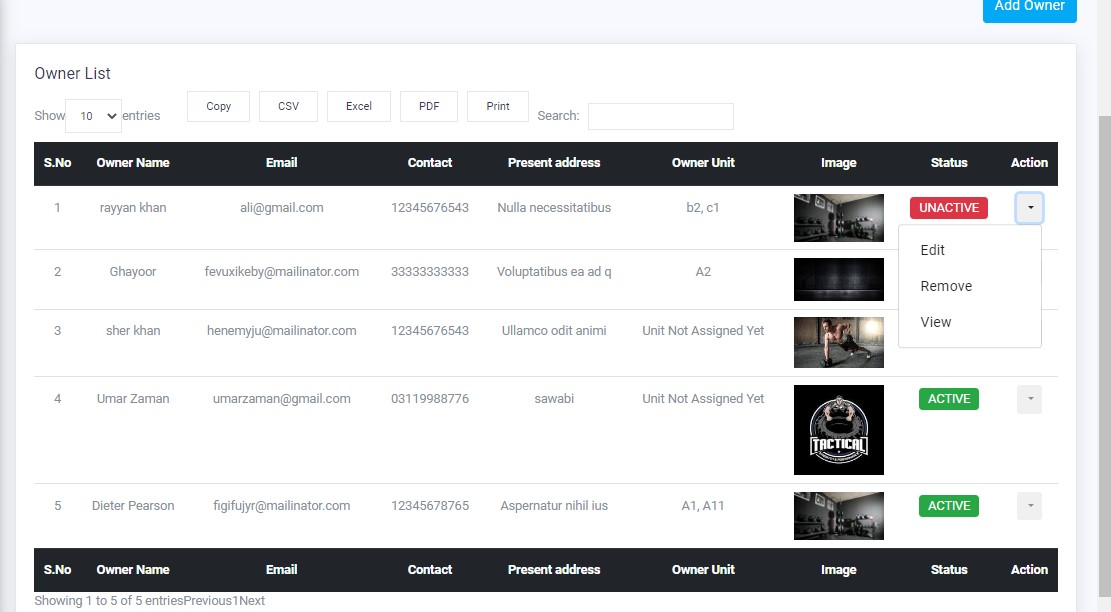
### Fig. 4.9 Add Owner

This is the "Owner List" page. Here, we display a list of owners along with their details. The page also shows the units that are associated with each owner.

There are several options available for each owner entry.

- "Edit": This button allows the admin to update the owner's details. By clicking on the "Edit" button, the admin can modify the owner's information such as username, password, name, email, contact details, and other relevant data.

- "Remove": This button is used to delete the owner's data from the system. By clicking on the "Remove" button, the admin can permanently remove the owner and all associated information from the database.

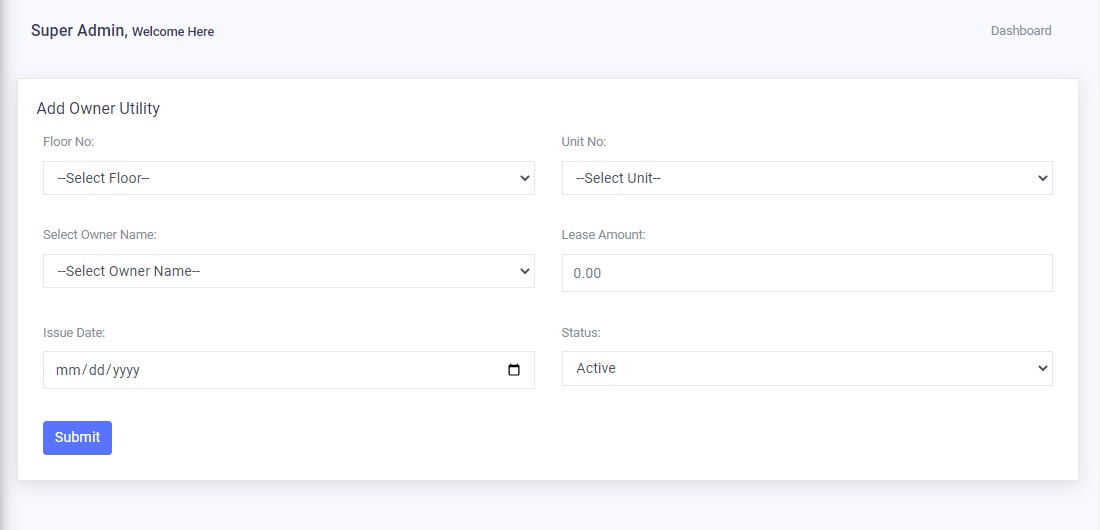
- "View": This button enables the admin to view the full details of the owner. By clicking on the "View" button, the admin can access a comprehensive view of the owner's information, including their username, password, name, email, contact details, and any other relevant data.

### Fig. 4.10 Owner List

This is the "Add Owner Utility" form. It allows the admin or super admin to add utility details for a specific owner. The form includes fields to input information such as utility type, utility name, utility cost, and any other relevant details.

By filling out this form and submitting it, the admin or super admin can add utility details for the owner. This information could include electricity bills, water bills, maintenance charges, or any other utility-related expenses associated with the owner's property.

The form includes dependent dropdowns for selecting the floor and units. When the admin or super admin selects a floor from the dropdown, the units dropdown will dynamically update to display only the units that are related to that specific floor. This ensures that the admin or super admin can only select units from the appropriate floor.

Once the form is submitted, the utility details will be stored in the system and associated with the respective owner. This enables the admin or super admin to keep track of the utility expenses and generate reports or invoices for the owner based on the utility data.

### Fig. 4.11 Add Owner Utility

This is the "Owner Utility List" page. Here, the super admin and admin can manage the details of owner utilities. The page displays a list of owner utility records, including information such as utility type, utility name, utility cost, and other relevant details.

The page provides an "Edit" button for each owner utility record. By clicking on the "Edit" button, the super admin and admin can update the details of the owner utility, such as modifying the utility type, changing the utility cost, or updating any other relevant information.

There is also a "Remove" button available for each owner utility record. Clicking on the "Remove" button allows the admins to delete the owner utility data from the system if needed.

Additionally, there is a "View" button provided for each owner utility record. By clicking on the "View" button, the admins can access the full details of the owner utility, enabling them to review all the information associated with that specific utility record.

Furthermore, there is an "Invoice" button available. Clicking on the "Invoice" button allows the admins to generate an invoice for the owner based on the utility details. This feature helps in creating bills or invoices for the owner's utility expenses, facilitating efficient financial management.

Overall, this page provides the necessary functionality for the super admin and admin to manage owner utility details, including editing, deleting, viewing, and generating invoices.

### 

### Fig. 4.12 Rent List

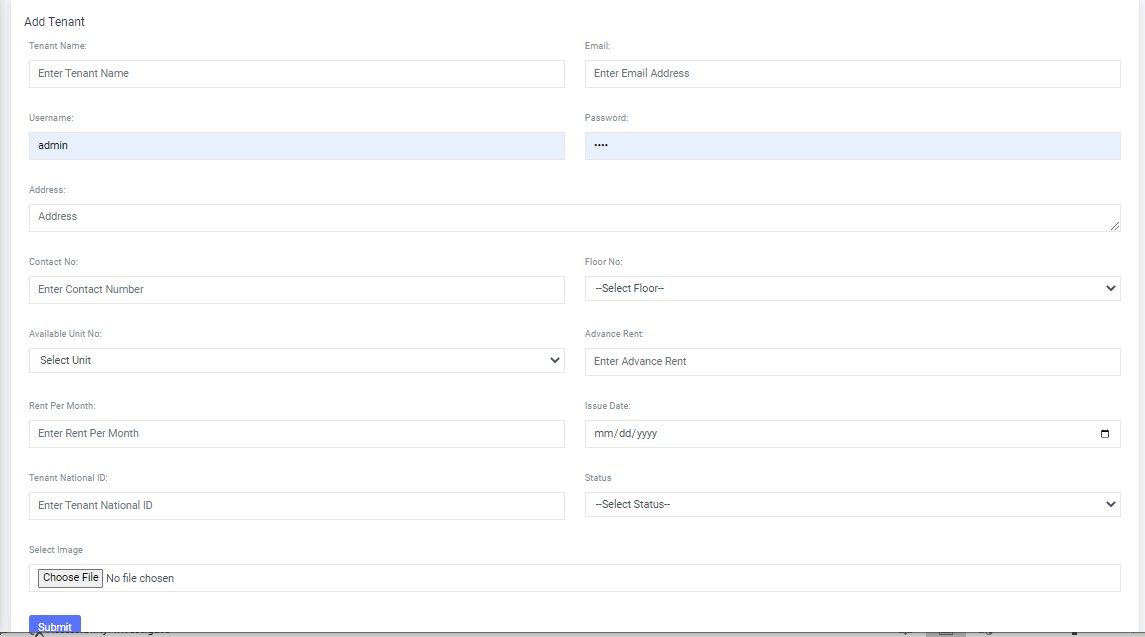
This is the "Add Tenant" form. Admins can use this form to add new tenants and assign units to them. The form includes validations to prevent the submission of incomplete or empty fields.

To ensure data integrity, the form enforces a rule that prohibits assigning the same unit to multiple tenants. Each unit can only be assigned to a single tenant at a time.

Moreover, when adding a new tenant, the form checks for duplicate entries based on the tenant's username. If a tenant with the same username already exists, the system prompts the admin to choose a different username to ensure uniqueness.

The form also includes dependent dropdowns for selecting the floor and units. When the admin or super admin selects a floor from the dropdown, the units dropdown dynamically updates to display only the units related to that specific floor. This ensures that the admin or super admin can only select units from the appropriate floor, reducing potential errors.

By incorporating these validation rules and dependent dropdowns, the form promotes accurate data entry, maintains data integrity, and prevents conflicts or duplication of tenant information.



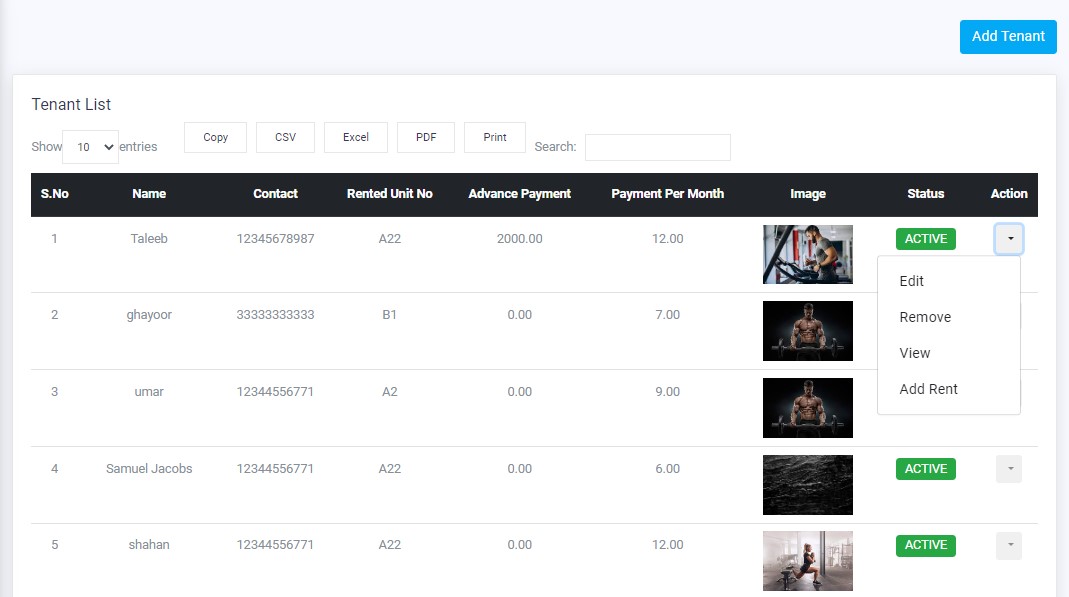
### Fig. 4.13 Add Tenant

This is the "Tenant List" page where both the super admin and admin can manage the tenant list and their details. The page displays a list of tenants along with their relevant information.

For each tenant, there are several action buttons available. The "Edit" button allows admins to update the tenant's information if necessary. The "Remove" button enables admins to delete the tenant's data from the system if required.

The "View" button provides access to view the complete details of a specific tenant, including their personal information, assigned units, and any other relevant data.

Additionally, there is an "Add Rent" button that allows admins to assign and manage rent-related information for each tenant. This includes details such as water bills, gas bills, and rent bills associated with the tenant.

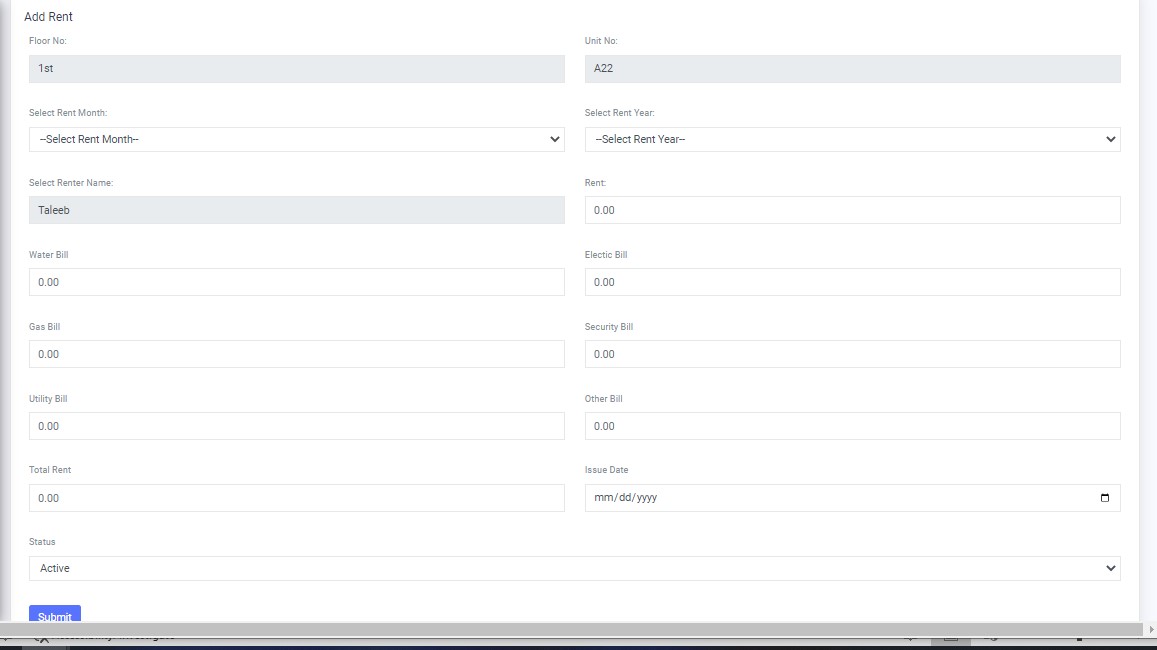
However, when it comes to deleting a tenant, it is not possible to delete them directly because the tenant table is the parent table for the add\_rent table. Therefore, to delete a tenant, the admin must first detach or delete the data associated with the tenant in the add\_rent table. Once the data in the child table is removed, the tenant can be deleted.

### Fig. 4.14 Tenant List

By utilizing these features and functionalities, admins can effectively manage the tenant list, handle tenant-related operations, and ensure accurate billing and record-keeping for each tenant. This is the "Add Rent" form. It is utilized to assign rent and other bills to a specific tenant. The form allows the super admin and admin to input and manage the rent details for a particular tenant.

Within the form, the admins can fill in the necessary information related to rent and bills. This may include fields such as tenant name, unit number, rental amount, due date, and details of various bills like water, gas, electricity, etc. The form ensures accurate record-keeping and efficient management of tenant payments.

By submitting the form, the assigned rent and bill information will be associated with the respective tenant. This allows for proper tracking and calculation of payments owed by the tenant.

Overall, the "Add Rent" form facilitates the process of assigning and managing rent and bills for individual tenants, providing admins with a convenient and organized way to handle financial aspects related to tenant occupancy.

### Fig. 4.15 Add Rent

This is the "Rent List" page. Here, the super admin and admin have the ability to manage the list of rents associated with tenants. The page provides various options for handling the rent details.

The "Edit" button allows the admins to update the rent list details. They can modify information such as rental amount, due dates, and any other relevant details related to the rent.

The "Remove" button is used to delete rent data if necessary. This option enables admins to remove rent entries that are no longer relevant or required.

The "Pay Bill" button is utilized by tenants to make payments towards their bills. When a tenant wishes to pay their bill, they can use this button to add the bill details and complete the payment process.

The "Invoice" button is used to generate invoices for tenants. This feature enables admins to create detailed invoices that outline the rent amount, bill charges, and any other associated fees. The invoice provides a clear breakdown of the payment due and facilitates effective communication between the tenant and admin regarding financial obligations.

Overall, the "Rent List" page provides admins with a comprehensive interface to manage rent-related information, handle updates and deletions, accept bill payments, and generate invoices for tenants.

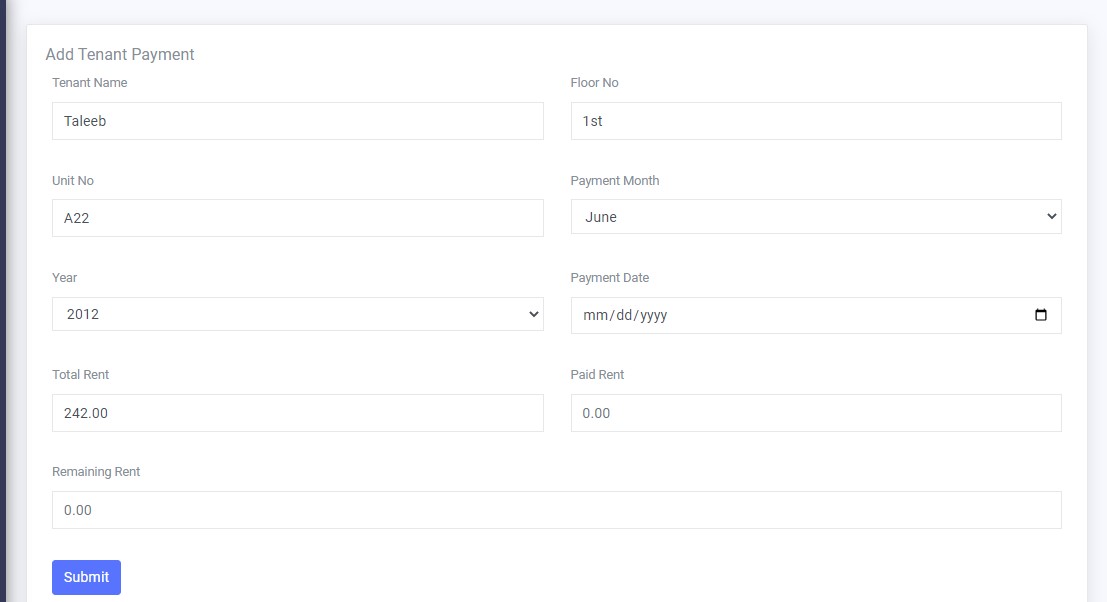
### 

### Fig. 4.16 Rent List

This form is specifically designed for adding tenant bill payments. It allows admins to record and manage the payment details associated with tenants' bills. By filling out this form, admins can effectively track and organize the payment information for each tenant.

The form likely includes fields for entering relevant details such as the tenant's name or ID, the bill type (e.g., water bill, gas bill, rent bill), the payment amount, the payment date, and any additional notes or comments.

Once the required information is entered into the form, admins can submit it to update the tenant's payment records and maintain accurate financial records for each tenant.

This form streamlines the process of managing tenant bill payments, ensuring that the payment information is properly recorded and easily accessible for administrative purposes.

### Fig. 4.17 Add Tenant Payment

The tenant invoice is a document that provides a comprehensive overview of all the bills and outstanding dues associated with a specific tenant. It serves as a summary of the financial obligations and payments made by the tenant during a specific period.

The tenant invoice typically includes the following information:

1. Tenant Details: The name, address, contact information, and tenant ID of the tenant receiving the invoice.

2. Invoice Number and Date: A unique identifier assigned to the invoice and the date the invoice was generated.

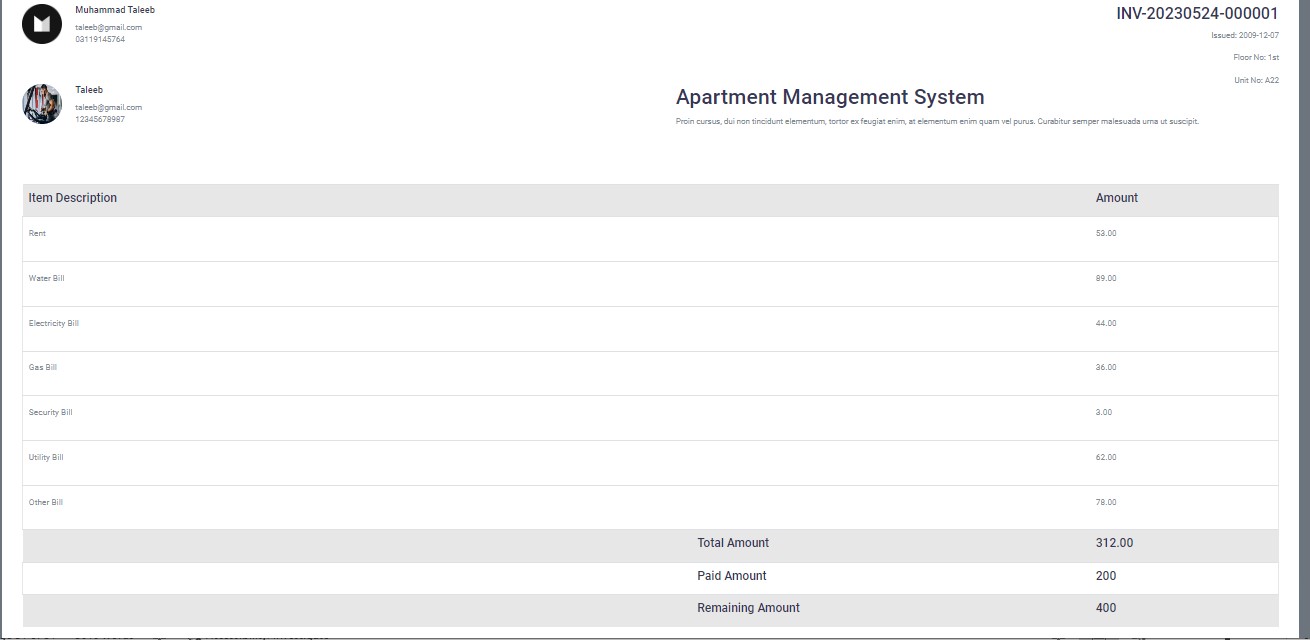
3. Billing Period: The period for which the invoice is issued, indicating the start and end dates.

4. Bill Details: A breakdown of all the bills and charges incurred by the tenant during the specified period. This may include rent, utilities (water, electricity, gas), maintenance fees, and any other applicable charges.

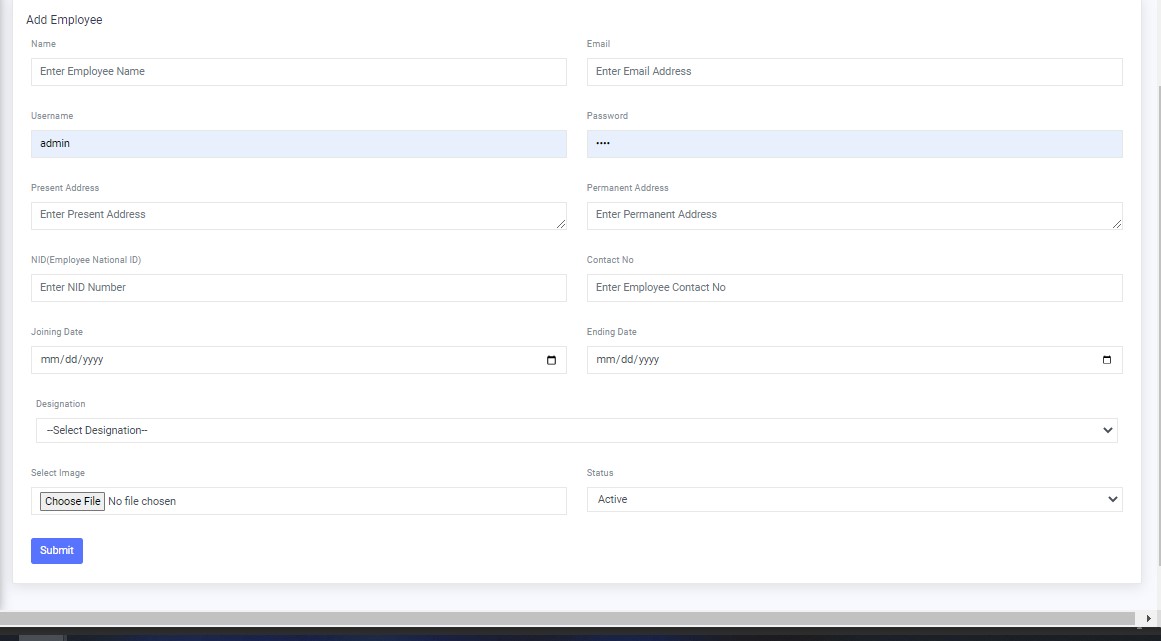
5. Payment History: A record of the payments made by the tenant, including the payment dates, amounts, and any outstanding balances.

6. Total Amount Due: The total amount that the tenant owes, considering both the outstanding balances and the current charges.

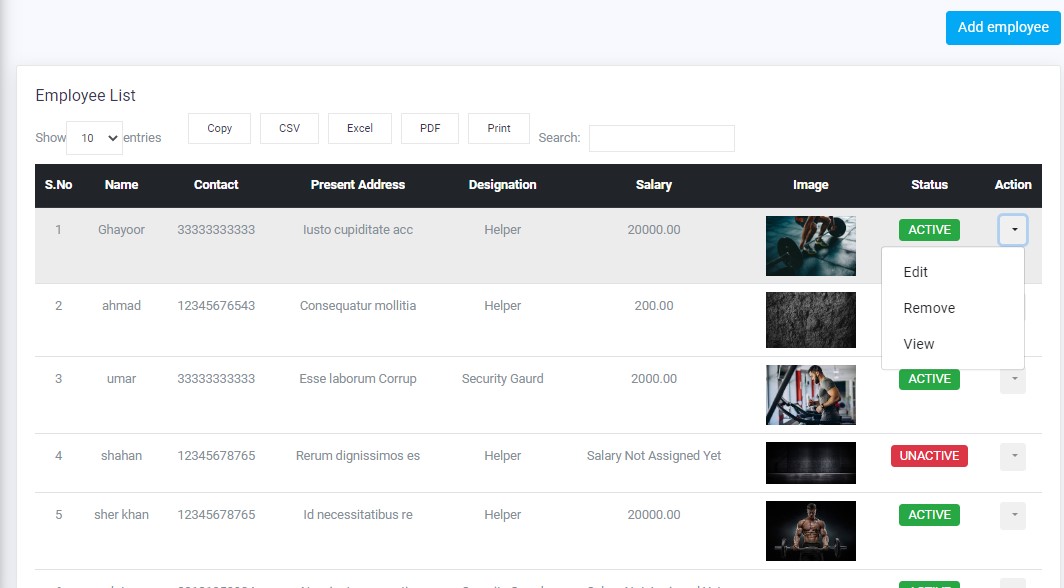
7. Due Date: The date by which the tenant is expected to make the payment.

The tenant invoice provides a clear overview of the financial status and obligations of the tenant, helping them understand their outstanding dues and facilitating timely payments. It also serves as a reference for both the tenant and the landlord or property management to track and reconcile the financial transactions.

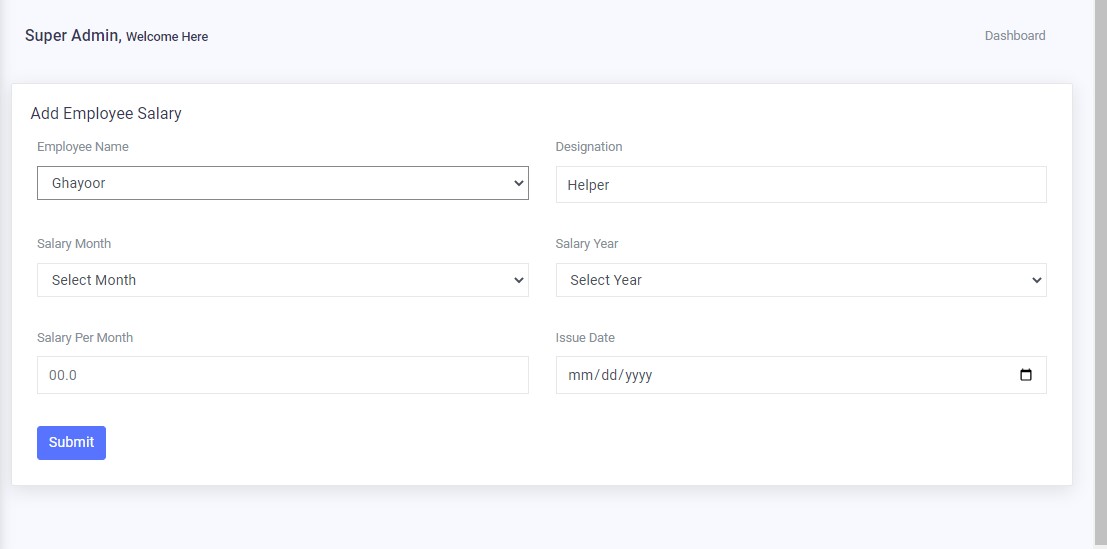
### Fig. 4.18 Tenant Invoice

This is the employee form. The admin and super admin can use this form to add new employees and provide their details.

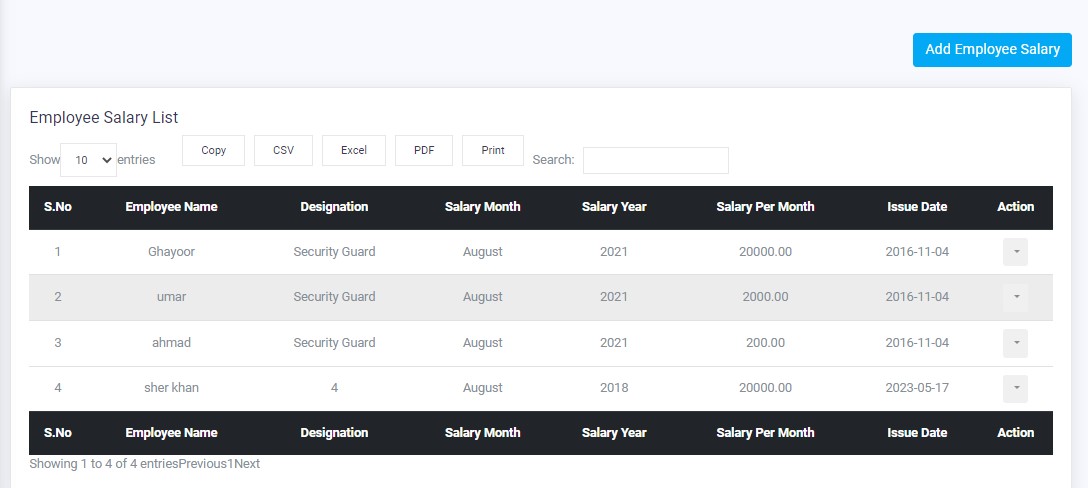
### Fig. 4.19 Add Employee

This is the employee management page where the admin and super admin can effectively handle employee data. They have the ability to update, delete, and view employee information from the provided list, ensuring efficient management of employee records.

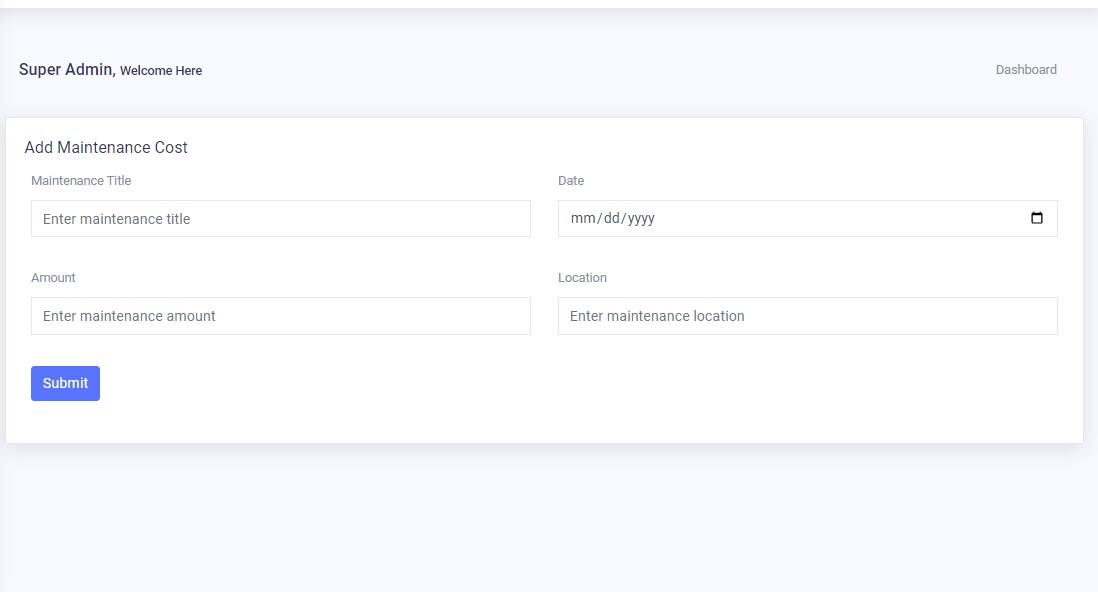
### Fig. 4.20 Employee List

 This is the "Add Employee Salary" form. From this form, the admin and super admin can assign salaries to employees based on their respective designations.

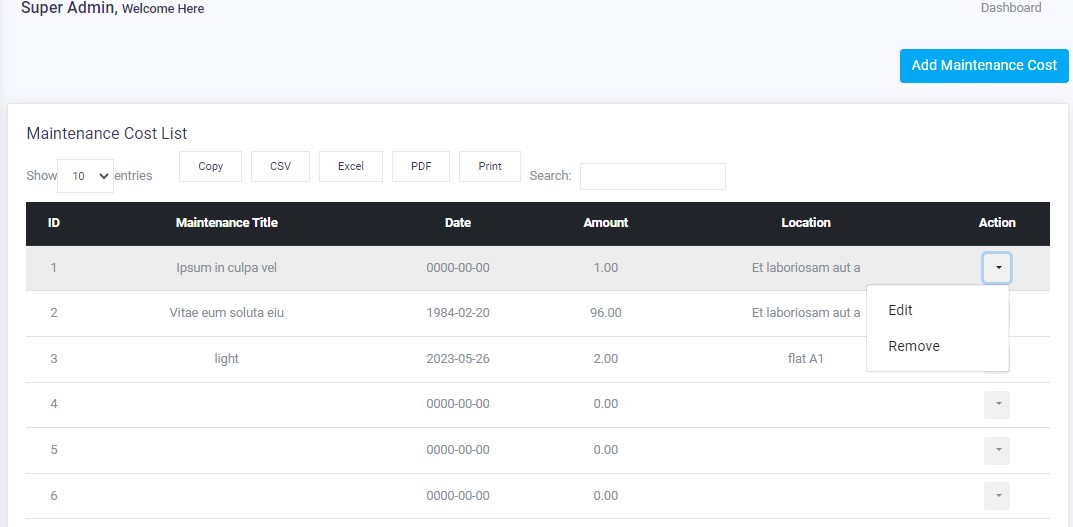
### Fig. 4.21 Add Employee Salary

 This is the "Employee Salary List" page. Here, the admin can view the salaries of employees, update their salary details, and delete employee records if needed.

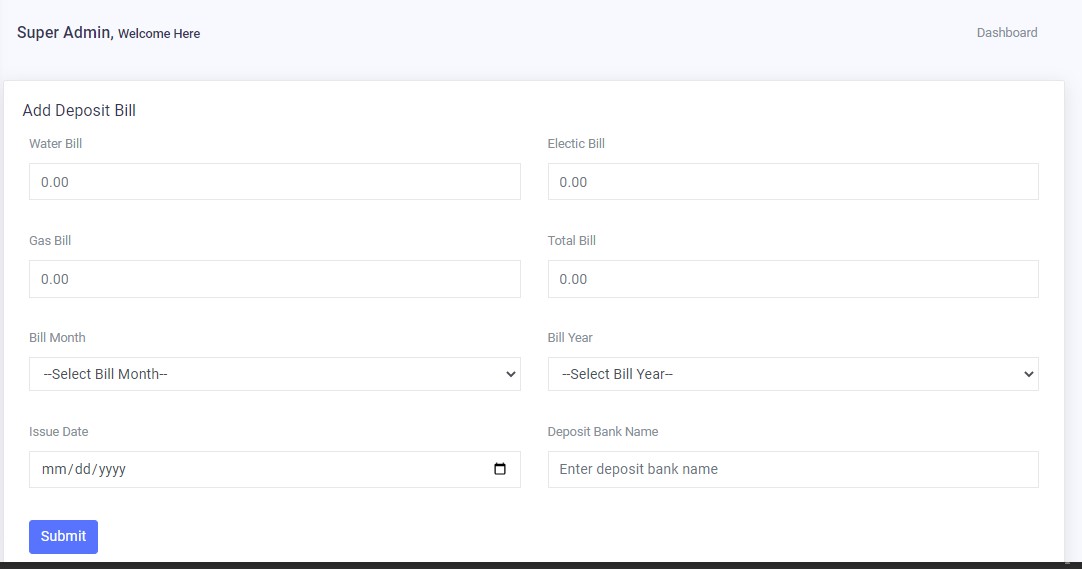
### Fig. 4.22 Employee Salary List

This is the "Maintenance Cost" form. Here, we can enter and add the details of maintenance costs.

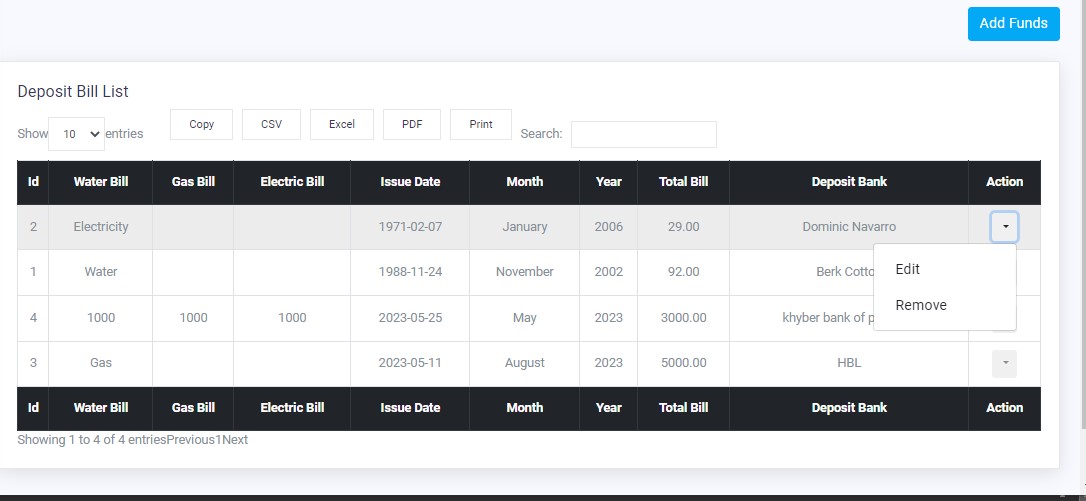
### Fig. 4.23 Maintenance Cost

This is the "Maintenance Cost List" page. Here, the admin and super admin can manage the details of maintenance costs. They can update and delete the records as needed.

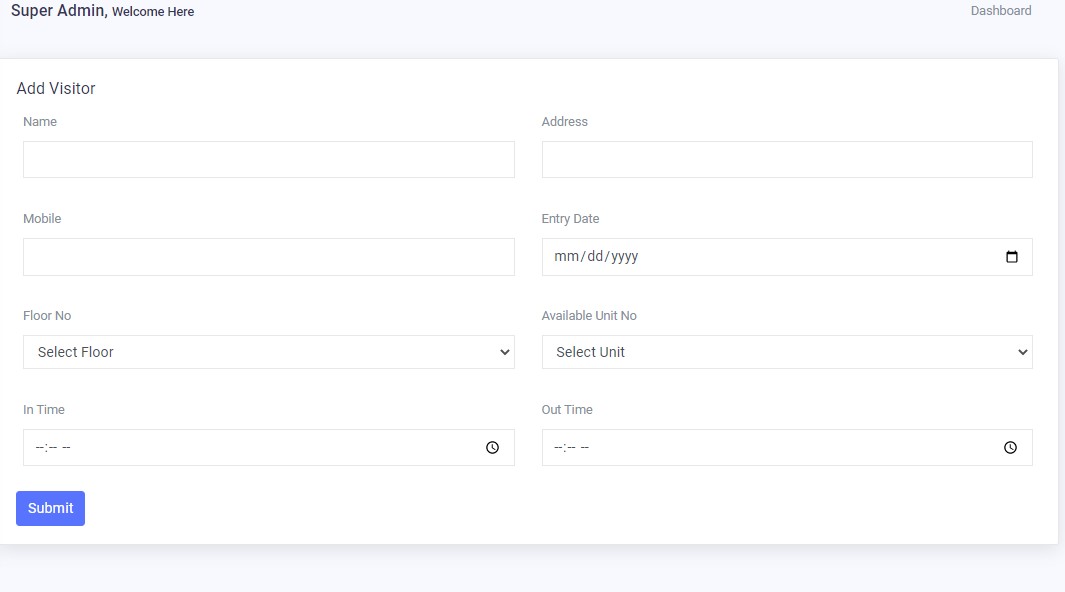
### Fig. 4.24 Maintenance Cost List

This is the "Bill Deposit" form. Here, the admin and super admin can add the total bills of the building.

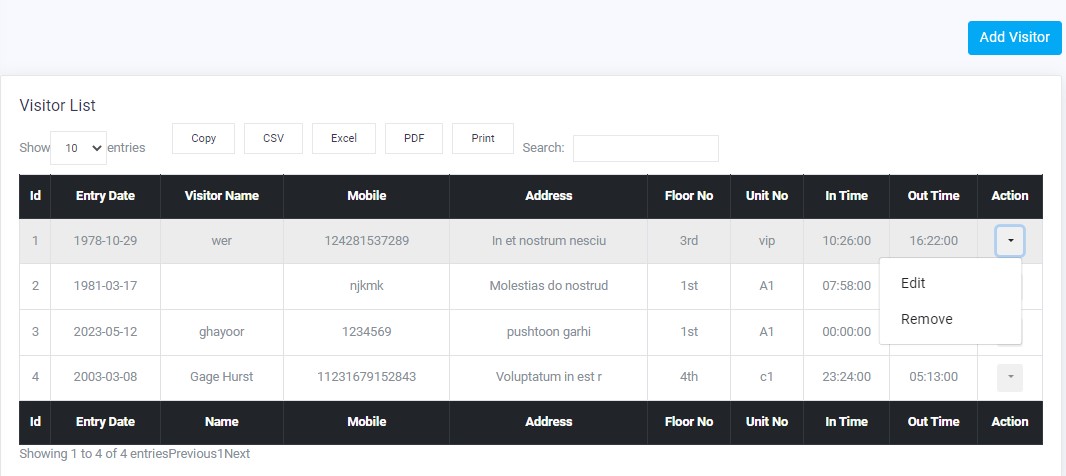
### Fig. 4.25 Add Bill Deposit

This is the deposit bill list from this list the admin and super admin can manage the deposit bill details and also can update and delete the record.

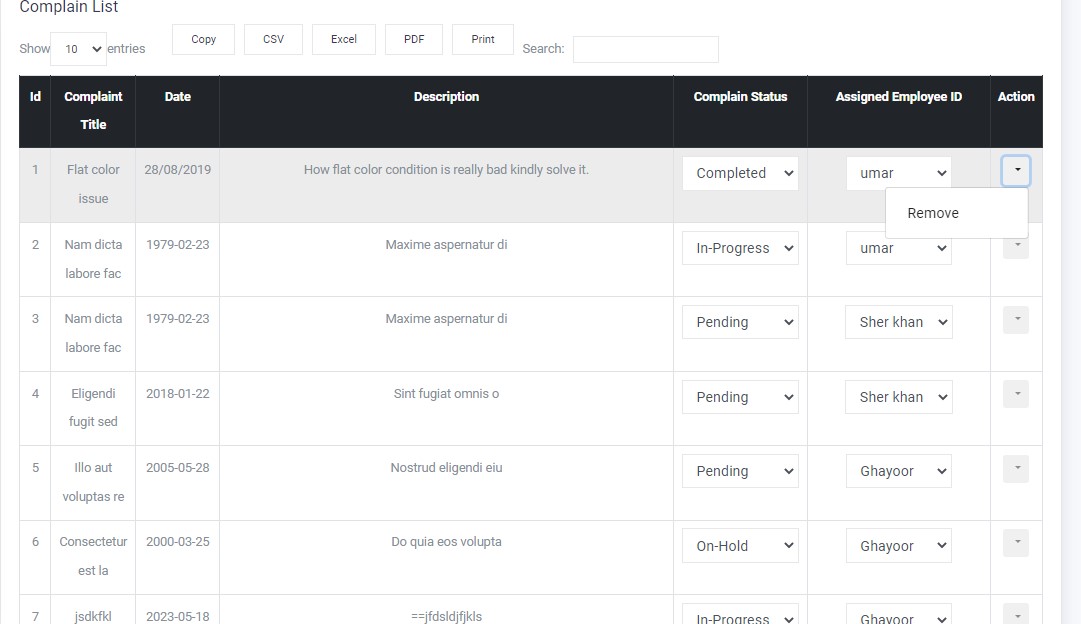
### Fig. 4.26 Bill Deposit List

This is the "Add Visitor" form. Here, we can enter the details of the visitors who come to the building as guests.

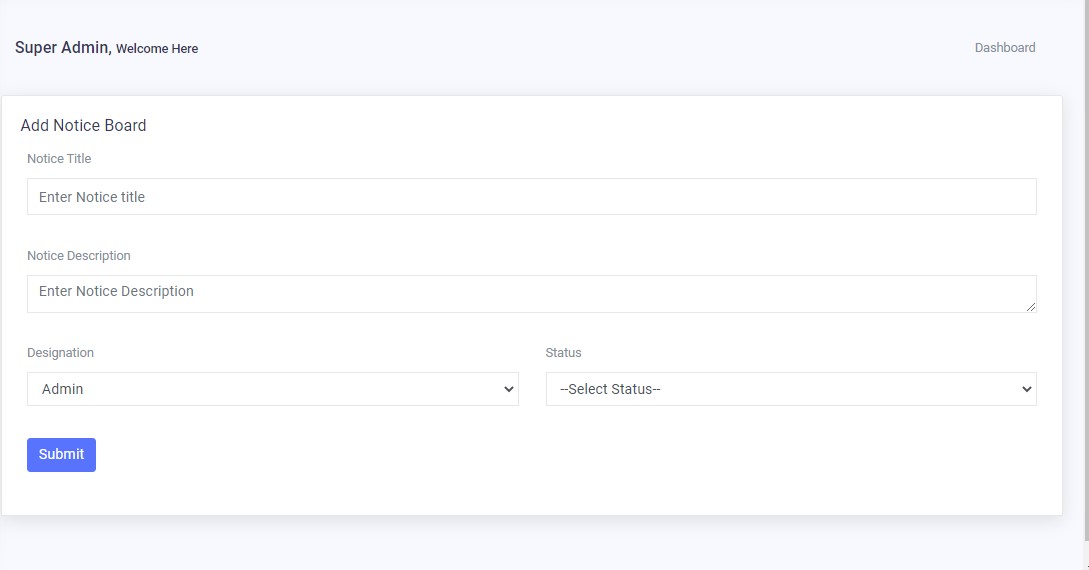
### Fig. 4.27 Add Visitor

This is the "Visitor List" page. Here, the admin and super admin can manage the details of the visitors. They can perform actions such as updating visitor information and deleting visitor records.

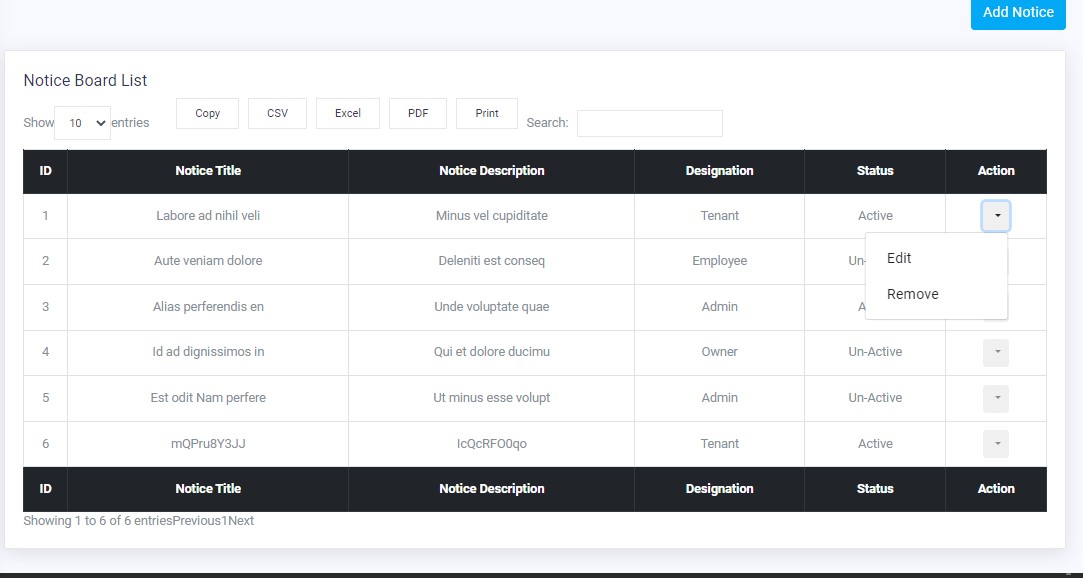
### Fig. 4.28 Visitor List

This is the "Complaint List" page. When a tenant or owner submits a complaint, the admin and super admin can assign it to a specific employee for resolution. This page allows them to manage and track the complaints, assign them to employees, and mark them as completed once resolved.

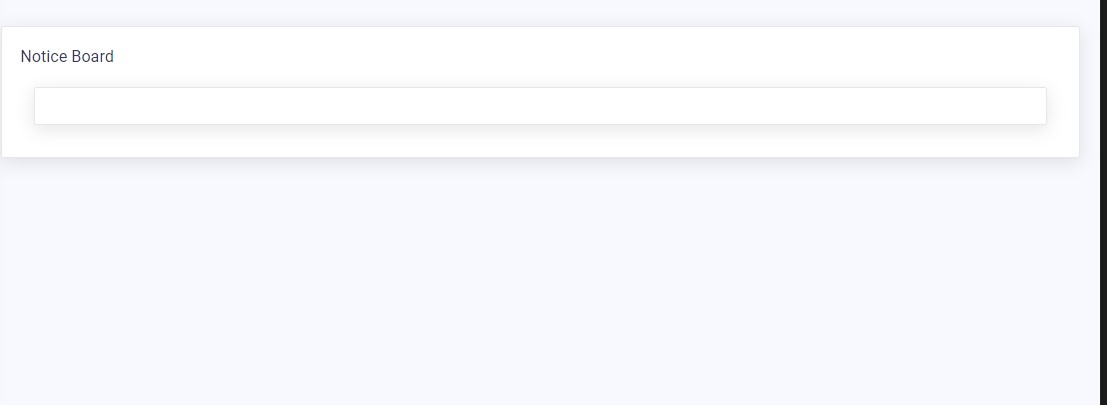
### Fig. 4.29 Complaint List

This is the "Notice Board" page. Here, the admin and super admin can add notices from tenants, owners, and employees. It serves as a platform to communicate important information, announcements, or updates to all relevant parties within the system.

### Fig. 4.30 Notice Board

This is the "Notice List" page. Here, the admin and super admin can manage the details of the notice board. They have the ability to update and delete notices as needed. This page allows them to effectively maintain and organize the notices displayed on the notice board.

### Fig. 4.31 Notice List

Here, the tenant, owner, and employee can view their respective notices. This feature allows them to access and read the notices specifically addressed to them, ensuring that they stay informed about any relevant information or announcements.

### Fig. 4.32 View Notice Board

## 4.8 Super Admin dashboard

### Fig. 4.33 Super Admin Dashboard

## 4.9 Admin dashboard

### Fig. 4.34 Admin Dashboard

## 4.10 Owner Dashboard

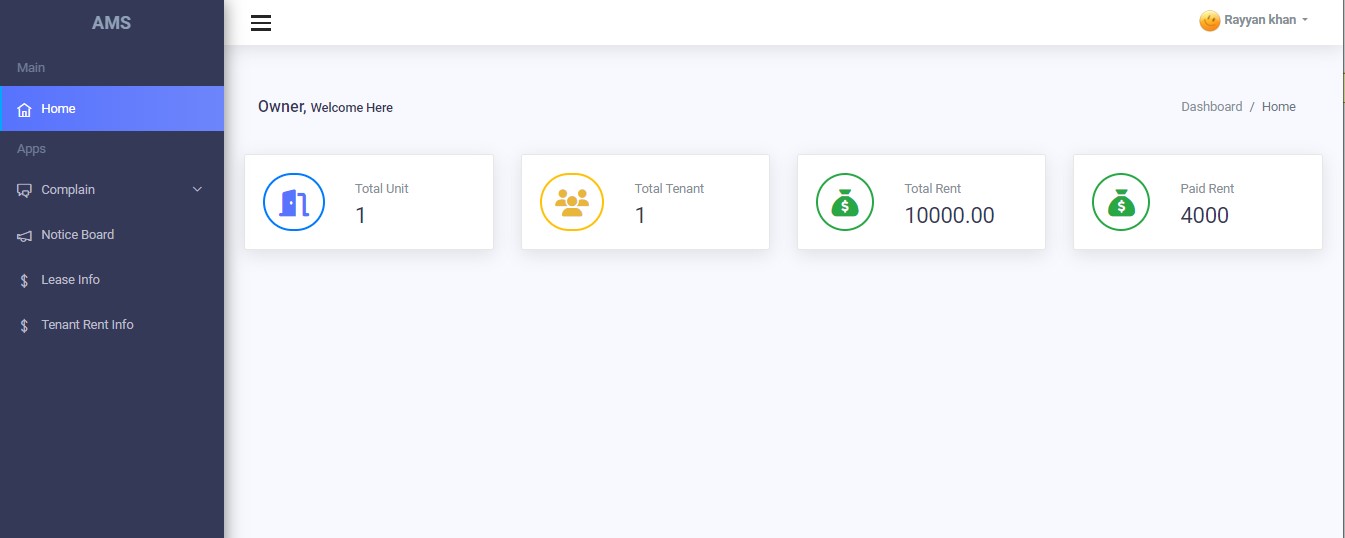
Here, owners have the ability to make complaints, view notices from the admin and super admin, check the lease amount of their units, and also view the list of tenants living in their units.

Owners can submit complaints through a dedicated platform where they can provide details about the issue they are facing. The admin and super admin will receive these complaints and take appropriate action to address them.

Owners can also access a notice board where admin and super admin can post important announcements or notices regarding the property or any related matters. This ensures effective communication between the owners and the management.

In addition, owners can check the lease amount associated with their units. This allows them to keep track of the rent or lease payments and ensures transparency in financial transactions.

Furthermore, owners have the ability to view the list of tenants who are currently residing in their units. This provides them with information about the occupants of their property and helps in maintaining accountability and security.

Overall, these features empower owners to actively engage with the management, stay informed about important updates, and have visibility into the financial and occupancy aspects of their units.

### Fig. 4.35 Owner Dashboard

## 4.11 Employee dashboard

### Fig. 4.36 Employee Dashboard

## 4.12 Tenant dashboard

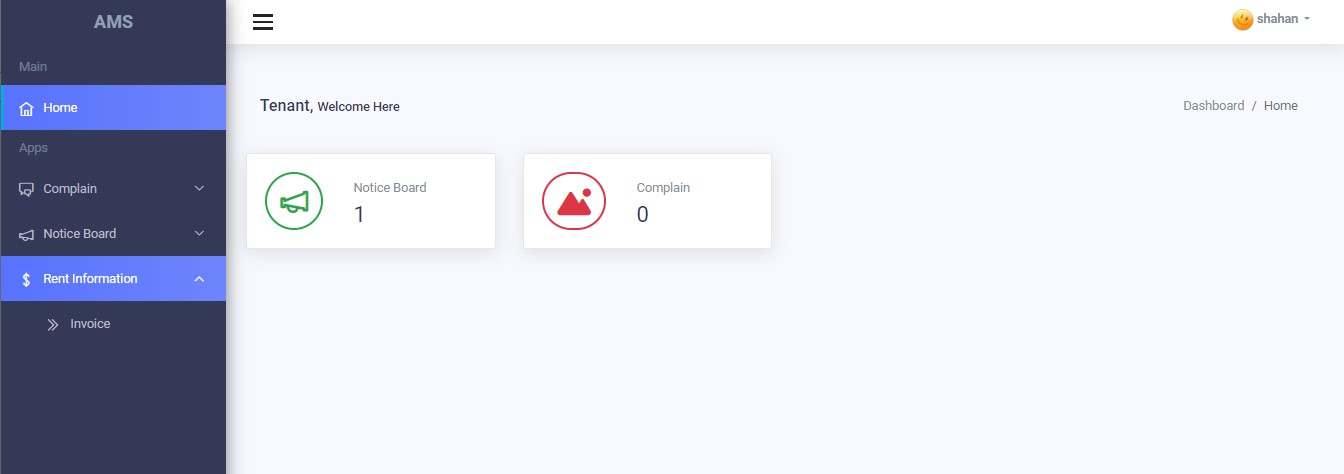
This is the tenant dashboard, where tenants have various features and functionalities available to them.

Tenants can submit complaints directly through the dashboard. They can provide details about the issue they are facing, such as maintenance requests or concerns about the property. The complaints will be received by the admin and super admin, who will take the necessary actions to address them.

Additionally, tenants can access the notice board, which displays important announcements or notices from the admin and super admin. This ensures that tenants stay informed about any updates or changes related to the property or their tenancy.

Furthermore, tenants can view their rent and dues information on the dashboard. This includes details about their rental payments, utility bills, and any outstanding dues they may have. This allows tenants to keep track of their financial obligations and ensure timely payments.

Moreover, tenants have the ability to create invoices for themselves. They can generate invoices for their rent payments, utility bills, or any other expenses associated with their tenancy. This feature helps tenants maintain a record of their financial transactions and facilitates clear communication between tenants and the management.

Overall, the tenant dashboard provides a convenient and centralized platform for tenants to communicate their concerns, access important information, manage their finances, and generate invoices, enhancing their overall experience and engagement with the property management.

### Fig. 4.37 Tenant Dashboard

# CHAPTER 5

# CONCLUSION

In conclusion, the Apartment Management System (AMS) is a comprehensive software solution designed to streamline and automate various tasks related to apartment management. The AMS provides different user roles such as super admin, admin, tenant, owner, and employee, each with specific privileges and functionalities.

The AMS offers features such as login authentication, allowing users to securely access their designated dashboards. The super admin has the highest level of control and can manage other admins, add floors, units, owners, tenants, and employee details. Admins have similar privileges but cannot add more admins.

The system allows for the addition, management, and validation of floors, units, owners, tenants, employee details, and visitor information. Users can assign units to tenants and owners, update and delete records, and view comprehensive details for each entity.

Additionally, the AMS facilitates bill management, enabling the assignment and tracking of rents, utilities, and other bills for tenants and owners. The system generates invoices and allows for the management of bill payments.

Complaint management and notice boards are also integral components of the system. Users can submit complaints, which can be assigned to specific employees for resolution. Notices can be posted by admins, tenants, owners, and employees, and are accessible for viewing by the respective recipients.

Overall, the AMS enhances the efficiency and organization of apartment management operations, providing a centralized platform for administration, communication, and data management. It streamlines processes, reduces manual effort, and improves communication and transparency among stakeholders.

## Summary, of the Apartment Management System:

In summary, the Apartment Management System (AMS) is a comprehensive software solution designed to streamline and automate various tasks related to apartment management. It offers different user roles, including super admin, admin, tenant, owner, and employee, each with specific privileges and functionalities.

The AMS provides secure login authentication for users to access their designated dashboards. The super admin has the highest level of control and can manage other admins, as well as add and manage floors, units, owners, tenants, and employee details. Admins have similar privileges but cannot add more admins.

The system allows for the addition, management, and validation of floors, units, owners, tenants, employee details, and visitor information. Users can assign units to tenants and owners, update and delete records, and view comprehensive details for each entity.

AMS also includes bill management features, enabling the assignment and tracking of rents, utilities, and other bills for tenants and owners. The system generates invoices and allows for the management of bill payments.

Complaint management and notice boards are integrated into the system, allowing users to submit complaints and assign them to specific employees for resolution. Notices can be posted by admins, tenants, owners, and employees, and are accessible for viewing by the respective recipients.

Overall, the AMS improves the efficiency and organization of apartment management operations, providing a centralized platform for administration, communication, and data management. It streamlines processes, reduces manual effort, and enhances communication and transparency among stakeholders.