8-2023

WEB BASED MANAGEMENT SYSTEM FOR HOUSING SOCIETY

Likhitha Reddy Eddala

WEB BASED MANAGEMENT SYSTEM FOR HOUSING SOCIETY

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

In

Computer Science

---

by

Likhitha Reddy Eddala

August 2023

WEB BASED MANAGEMENT SYSTEM FOR HOUSING SOCIETY

A Project

Presented to the

Faculty of

California State University,

San Bernardino

---

by

Likhitha Reddy Eddala

August 2023

Approved by:

Dr. Yan Zhang, Advisor, School of Computer Science

Dr. Jennifer Jin, Committee Member

Dr. Qingquan Sun, Committee Member

ABSTRACT

Web Based Management System for Housing Society plays a major role in our day-to-day life. We develop a global web dependent application using AngularJS, Node JS and MySQL, with Xampp as the server to make an effective management system. This system is designed to provide a user-friendly and efficient platform for managing all the details of daily notices, monthly meetings, events, payments, maids etc., This system mainly consists of three modules, they are: Admin, User and Security. Each module here serves specific features and functionalities present within society. Admin module provides the features for managing user, houses, security, maids, notices, events, payment and voting information where we can add and view the details required as per the features. User module has features like events, notices, payments and voting where we can view the information provided by admin and make an action according to that. Security module has features like visitors, maids, notices, events and provide information as per the specific feature.

ACKNOWLEDGEMENTS

I would like to thank Dr. Yan Zhang for serving as my advisor for Web Based Management System for Housing Society project, for her continuous support and encouragement motivated me to overcome challenges and strive for excellence in successful completion of my project.

I am also thankful to Dr. Jennifer Jin for believing in me and providing her valuable contributions as a committee member and helping me in completing my project successfully.

I would also like to extend my appreciation to Dr. Qingquan Sun for serving as a committee member and providing immense support in the completion of my project successfully.

I am deeply grateful to all three committee members for their time, expertise and valuable contributions to this project. Their guidance and support have been invaluable in shaping the outcome of this work.

Lastly, I would like to thank my friends and family for their support and encouragement throughout the project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

Background

Now-a-days it’s very hard to maintain the data manually and store the data, so we are developing an application to maintain the complete information regarding the housing system at one place to easily access by everyone belonging to the society. In today’s technical world everyone is flexible using the internet and accessing such applications and making lifestyle easier day by day. By using Web Based Management System for Housing Society we can even provide security to the society members and maintain records of the visitors, maids and everyone by maintaining the data in the database (MySQL). By this we can easily reduce the fraud rate. The main objective of this project is to design an efficient and simple management system for housing society which can be used by people as well as for business purposes. This application is implemented using Angular JS and Node JS.

Significance

An efficient and simple management system can be used for maintaining housing society more effectively. System service request, monitoring online, etc., services are provided by this application. The management system design which helps in conflicts reduction arisen in housing society. This application is useful for sectors with complex residentials as well as real estate, as it provides modern

services and comprehension. This application is designed by focusing on the requirements of businessman and common people. Residents will get more unique experience in living and improved access for operations performed by management. The application design helps in the processes such as chat bot, charges board notifications, installment in payment etc.,

Purpose

Main objective of this project is to design an efficient and simple management system for housing society which can be used by people as well as for business purpose. This project helps in an individual society for people, security and admin. This application can be used by users as well as with some additional features for business purposes. This application works in the form of Input/Output data forms. Users, Security as well as admin can access the application very easily by logging into an application with the registered username and password.

Motivation

Daily living in urban regions has significant implications for housing society administration. In most circumstances society management communicates in a conventional manner. This does have constraints and drawbacks. Everyday announcements, monthly gatherings, occasional gatherings like events, various contacts for daily requirements, security warnings, high priority communication and lot more could fail to be delivered correctly in the present circumstances

because most activities are managed by manual means. To tackle the challenges caused by this time-consuming manual approach process. An automatic solution that reduces people's efforts must be devised.

The major goal of the society administration framework is to integrate several programs into only one smartphone application for handling alerts, updates related to well-being and social events. It also assists in event management and provides information about them in the application's interface. It alerts users to critical concerns and complaints regarding current problems.

CHAPTER TWO

LITERATURE REVIEW

“Housing Society Management System Using PHP” by V. Harsh et al (2022)

[16] In order to manage a cooperative society’s daily operations and improve the way things are now in society, the Housing Society Management System was created. The goal of the proposed study was to develop online tools for resolving complaints in guarded and gated communities. This system is developed to consider the challenges or problems in managing the daily activities and solving complaints in gated communities. These communities have all the security measures to ensure the safety and security of the residents. However, the current method to raise complaints is through messaging has proved to be time-consuming and not that efficient for the management. To make this process more efficient we recommend developing web-based apps. These apps provide a more efficient and user-friendly approach for handling complaints. The effectiveness of the application is being used to handle real complaints and the progress of these complaints is monitored. This system aims to enhance communication and streamline operations.

“Society management application on android” by R Bhagwat et al (2018)

[2] Housing Society administration is an important aspect of daily living in urban settings. Housing Society management covers a variety of daily

necessities including water supply, electricity, security and many more items that directly or indirectly affect residential living. Most of the time, society management uses conventional forms of communication. Yes, there are certain drawbacks and restrictions to this. Everyday announcements, monthly gatherings, cultural events, contacts for a variety of everyday requirements, security warnings, urgent communications, and a host of other things that might not be appropriately communicated in the existing environment. Since, the majority are handled manually.

Due to this, transparency is lacking. An automated system must be designed to lessen the requirement for human employment and solve the issues brought on by this time- consuming manual system. By minimizing efforts and advancing dependable communication, we provide a wiser and more effective method of handling crucial issues in order to overcome the shortcomings of the current system. Complaints, Meetings, Notices, Suggestions, Rules, and Miscellaneous contracts are just a few of the several social services that will be available in one place for users to view and utilize as needed.

“Housing Society Management “by S. Kudale et al (2018)

[7] The residential existence heavily relies on the administration of the housing society. The administration of housing societies was responsible for the daily requirements, including water supply, energy, security and maintenance. Although there were numerous conventional procedures and a lot of paperwork, this system exists to aid and make lives easier. The suggested solution will

computerize all of society's daily activities using an Android application. This automated system includes a calendar, compliance, domestic help, and other parts in addition to storing information on daily announcements, monthly meetings, and cultural events.

"Study of implementation of society Management System" by S Gavhane et al (2015)

[8] The research of Android application installation for housing society administration was covered in this essay. Housing society administration is an important aspect of daily living in urban settings. The management of the housing society includes all the daily necessities, including water, electricity, security, and many more elements that directly or indirectly affect residential life. Most of the time, society management uses conventional forms of communication. There are undoubtedly some drawbacks and restrictions to this.

"Housing Society Management Web Application with recommendation system" by S Raut et al (2017)

[17] created the suggested application by enabling people to connect with one another by joining a social, cultural, or athletic organization, this online application improved society's social fabric Using the app to submit concerns and ideas instead of going in person to the chairman was another benefit for the locals. The application was updated as soon as the issue with the complaint or suggestion was handled. The inhabitants benefit from this in that they are always

informed about happenings pertaining to society and can take part in numerous activities.

“Cloud based Housing Society Management System” by Tirth Shah (2018)

[3] In order to make the existing state of society more straight forward and effective, this article emphasizes a framework that allows users to administer housing societies utilizing a web-based, cloud-based system. Currently, every city is home to many societies. Property Tracking, online maintenance payments, notification systems, cost tracking, and the creation of yearly reports are just a few of the amenities offered by society management systems. Members of this system can interact with one another depending on their shared interests. Cloud-based software makes it simple to modify it to meet societal needs. Without the developer's involvement, this system can handle many users. The most crucial feature of this system is that it is user-friendly and entirely automated. It is also cloud-based. In the system, there are several features available. For example, there is an online voting scheme for various roles in society, such as secretary, chairman, and treasurer. The system offers a database of every member of society, allowing users to quickly find contact information for anybody they choose to speak with. The system aims to make communication simple.

"Improvement of the Management System of Housing and Communal Services in Uzbekistan" by Nurimbetov et al. (2019)

[4] The characteristics of housing and communal services, a significant component of the territorial infrastructure, are examined in this article along with their effects on the standard of living for the general population and the level of civilization in the community. Additionally, a study of the reforms, initiatives, and triumphs in Uzbekistan's housing and community services during the country's years of independence was conducted.

"Society connect App: E-Manager for Residential Society" by I. Niphade et al. (2023)

[9] A platform designed to take control of residential societies simpler was the Residential Society Management System Android app. When using the previous method, the secretary had to take the time to manually gather maintenance, post messages on the board, and handle complaints and problems. The process was lengthy, complicated, prone to errors and delays, and generally ineffective. The Society Connect app provides a number of features to manage many aspects of society, such as preservation collecting, facility reservations, sharing and deleting of society messages, and managing site visitors and employees. By addressing the issues with outdated guide systems, the software streamlines and simplifies societal governance. The software allows users to produce invoices, check their payment history, and pay

maintenance fees online, all of which reduce the likelihood of errors and guarantee on-time payments. The software also allows users to reserve amenities like a clubhouse, gym, and pool, eliminating the need for physical reservations and making the process more convenient and environmentally friendly. Every resident is informed of forthcoming events and developments thanks to the app. The highest level of sophistication is simplicity. Daily announcements, monthly conferences, cultural events, ad hoc connections for day-to-day needs, security indications, and high-priority communication may not be effectively delivered in the modern environment since most issues are handled manually. All the problems will be resolved and numerous elements for preserving societal openness will be offered through the Society Connect App. In this study, efforts were made to reduce the amount of paperwork so that owners could easily obtain information about other apartment owners. With the use of this internet tool, you may save back on paperwork, phone costs, and human labor.

" SURVEY PAPER ON HOUSING SOCIETY MANAGEMENT SYSTEM" by A. Arjun (2021)

[18] In order to improve the existing state of society, the Housing Society Management System was created. When a pandemic, like the current COVID-19, is present, leaving the house for activities like checking the noticeboard or giving a crucial message to other flat mates is not advised. By offering a wide range of facilities like daily notifications,

monthly meeting schedules, updates on upcoming cultural events, a breakdown of fund distribution to maintain society transparency, and a special health care facility, the Housing Society Management System alleviates all social problems. This function, which sends out an SOS signal to all the members of society if any one of them becomes ill or is gravely hurt and in need of assistance, will be highly helpful in the current scenario brought on by the notorious COVID -19 epidemic. Another element of this system is a chatbot that will assist users in navigating the app and using all its capabilities. Every resident may simply access all the social events and occurrences in the community thanks to the simple method provided by this program. Another function is help from physicians and nurses in case of an emergency.

"Society Management System using Web Technologies" by Chaudhari et al. (2021)

[19] The living circumstances were significantly influenced by the housing community's management. The housing community was responsible for providing for our everyday requirements, including water, power, security, and care. The purpose of this program was to make lives easier, yet it involves a lot of paperwork and old procedures. The application suggested here was a web-based one that will be digitized for common community usage. This application was an autonomous program that will keep track of daily alerts, monthly meetings, and

cultural events. It has sections for consensus, home assistance, calendar, and other things.

"Housing Society Management System Using IoT" by A. Sawalkar et al. (2021)

[20] All physical items will be connected to the Internet thanks to the IoT, a technological revolution. The routine activities will take on a new dimension thanks to emerging technologies. With this technology, a practical application was implemented simply and inexpensively. The goal of the housing society management system, which precisely echoes this term, is to simplify and improve the way things are now in the community. It simplifies life for society's citizens by automating some of the characteristics that occur inside it.

Challenges

In the current circumstances, the people taking care of the housing society administration, general notice board as the tool which is usually maintained by accountable society members. Numerous civilizations are also starting to adopt automatic chat infrastructure, which is beneficial to some level but fails to fulfill an objective. Following are a few of the primary drawbacks associated with these approaches [8]

The conventional method adopted in the housing societies could not be efficient as expected and it was prone to numerous challenges in it, which are as follows:

- Absence of dependability and authenticity- A board for posting information is capable of being controlled in person and may be written by hand or published. In everyday practice, it is readily changed or wrongly placed, which could give rise to a misunderstanding. It may be impacted by any fake data.
- Info being unreachable - This system is one of several types of systems incorporating Society members as well as notifying parties. Frequently, a notifying admin misses to make changes to the said board, or members of the society neglect to adhere to the instructions posted on it. Consequently, interaction is hampered.
- Absence of acknowledgement- Because this is a manual procedure, it is impossible to foresee whether the notification will reach every individual who is concerned. And any type of acknowledgment is lacking, which causes expenditure when trying to postpone or notify any sort of modification.
- Time incurring routine- to oversee and care for this said board, somebody must accept complete accountability. The individual must occasionally compromise with one's personal time plan.

Consequently, the automatic chat solutions are beneficial as a general chat window, they do not provide a specialized setting for devoted purposes, resulting in needless data exchange overwhelming the core topic.

CHAPTER THREE

SYSTEM REQUIREMENTS

Hardware Requirements

Application : Web Application

Memory (RAM) : 4GB

Storage : 64 GB

Hard Disk : 160 GB

Keyboard : Standard Windows Keyboard

processor : Intel i3 Processor

Monitor : SVGA

Software Requirements

Operating System : Windows

IDE : Visual Studio Code

Programming Languages: Angular JS, NodeJS, TypeScript

Database : MySQL

Tools : VS Code, Xampp, phpMyAdmin, MySQL, Stripe

Server : Apache Tomcat

CHAPTER FOUR

TOOLS AND TECHNOLOGIES

Web Based Management System for Housing Society project is deployed into a Localhost using Apache Tomcat server within the Xampp application. Here we are using Angular JS for Frontend and Node JS for the backend and MySQL for the database. Where MySQL databased is also running using the Xampp. Where Apache Tomcat and MySQL are the components of Xampp application. For data visualization we are using phpMyAdmin. We are using the Stripe payment gateway to make online payments. Here we are using the Visual Studio Code IDE for writing and running the code.

Angular JS

Angular.js is nothing but a scripting language. I have used the latest version of angular in my project which is one of the popular frontend frameworks. By using angular we can produce a single page application which is usually called SPA application. Angular provides type safety to the code where the code is written in the typescript format. We just change the data from one page to another, we can see this in our applications file structure. When we initially create our project many boilerplate files are created which means repeated files.

Angular.json has all the information about configuration settings. It has reference to the main file which tells where the application should start. Where in main.ts it will the direct the flow of execution to app.module

where it starts the app. This is the module, created with the @NgModule decorator, which has declarations of all the components we are creating within the app module so that angular is aware of them. app.module.ts we connect the entry point before building to app.component.ts

Bootstrap

After building the project in angular application, it was converted into one html file and two to three .js files where the html file is the index.html and the code is placed in the index.html file as we configure app.component.ts in index.html as shown below.

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>My Hello World App!</title>
<base href="/">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
<app-root></app-root>
</body>
</html>
```

After building it forms like this

```
<!doctype html>
 <html lang="en">
   <head>
     <meta charset="utf-8">
     <title>My Hello World App!</title>
     <base href="/">
     <meta name="viewport" content="width=device-width, initial-
scale=1">
     <link rel="icon" type="image/x-icon" href="favicon.ico">
   </head>
   <body>
     <app-root></app-root>
<script type="text/javascript" src="runtime.js"></script>
<script type="text/javascript" src="es2015-polyfills.js"nomodule></script>
<script type="text/javascript" src="polyfills.js"></script>
<script type="text/javascript" src="styles.js"></script><script
type="text/javascript" src="vendor.js"></script>
<script type="text/javascript" src="main.js"></script>
  </body>
 </html>
```

NodeJS

Node.js is nothing but a Scripting language. Here, I have used Node v14.17.3 version. JavaScript is a programming language that allows developers to make interactive websites. Normally, JavaScript runs in web browsers to make things happen on a webpage, like showing pop-up messages, validating forms, and creating user login pages etc., However, Node.js takes JavaScript to the next level by allowing it to run on servers. When we visit a website, the browser sends a request to the server, which processes the request and sends back the necessary data to display the webpage. Node.js allows developers to write the server-side code (Backend code) in JavaScript. In simpler terms, it lets them create the logic that handles the requests by using SQL queries for retrieving the data from the database and sends responses to users.

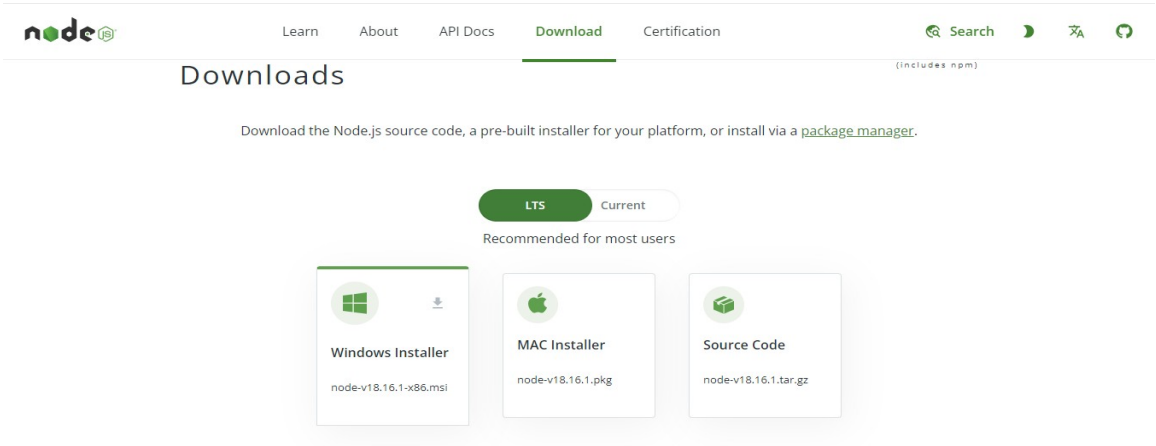

Figure 1. Node JS

MySQL

SQL is a language used for storing, manipulating the data and deleting the data that is stored. The data stored in the database in the form of tables with rows and columns with specific information. In the below table each row represents a student, and each column represents a characteristic of that student.

phpMyAdmin

phpMyAdmin is a database management system used for visualization. It is a free and open-source web-based application which provides graphical user Interface (GUI) for MySQL database.

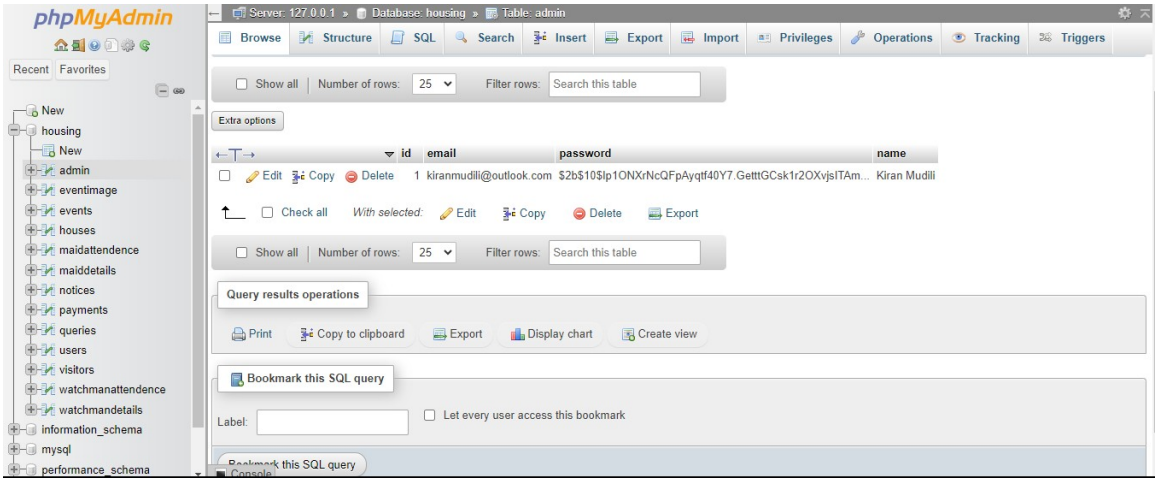

Figure 2. phpMyAdmin

XAMPP

Xampp is a popular open-source software package that provides a complete web server for local environment to build web applications. Xampp

stands for cross-platform which allows developers to build a web application without the need of an internet connection or remote server.

Apache HTTP server and MySQL are the components of Xampp. It also includes phpMyAdmin as a web-based database management tool which provides user-friendly interface for managing the MySQL database.

We can create and test web applications locally using Xampp. It is suitable for development and QA environments but not suitable for the production environment.

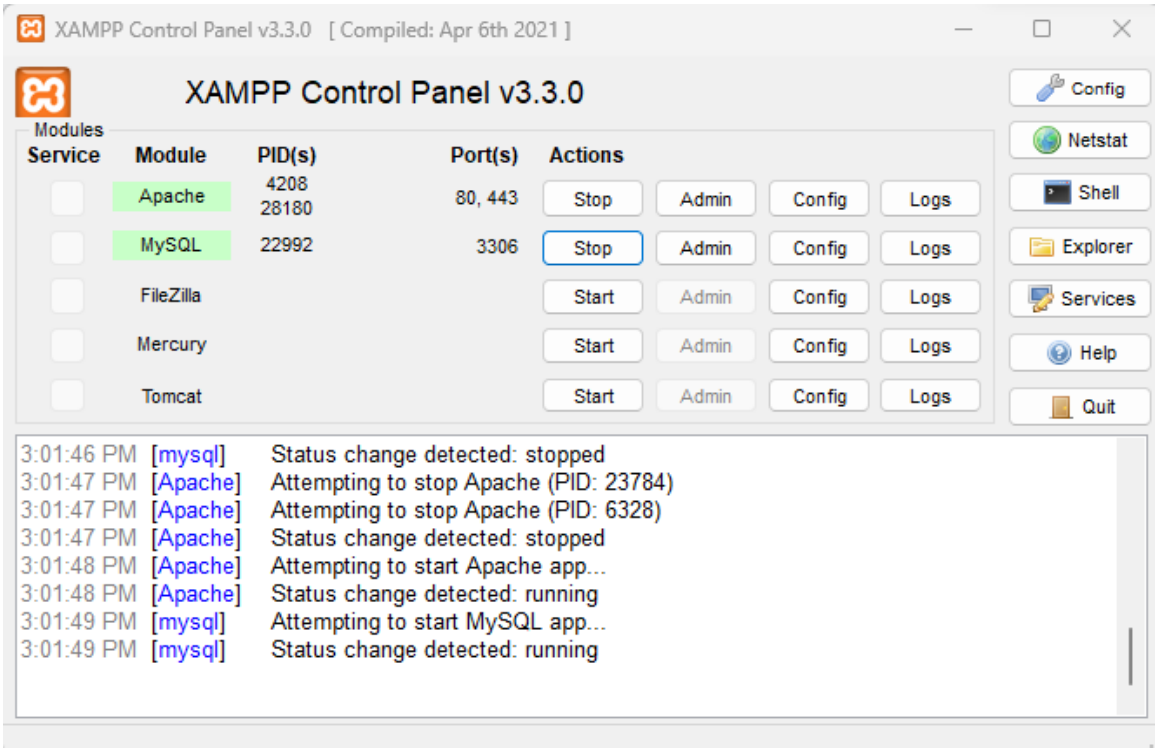

Figure 3. Xampp

CHAPTER FIVE

SYSTEM DESIGN

UML Representation

Our System designs could be expressed better with the various UML representations as shown in the below sections.

[21] UML could be abbreviated as Unified Modelling Language. It is a regulated common-application language used for modelling in the sector of object-based software engineering. The regulation is maintained as well as generated by the Group of Object administration. The UMLs are aimed to be a usual language for generating models of object-based computer software. It consists of two main elements like a Meta-model and a notation.

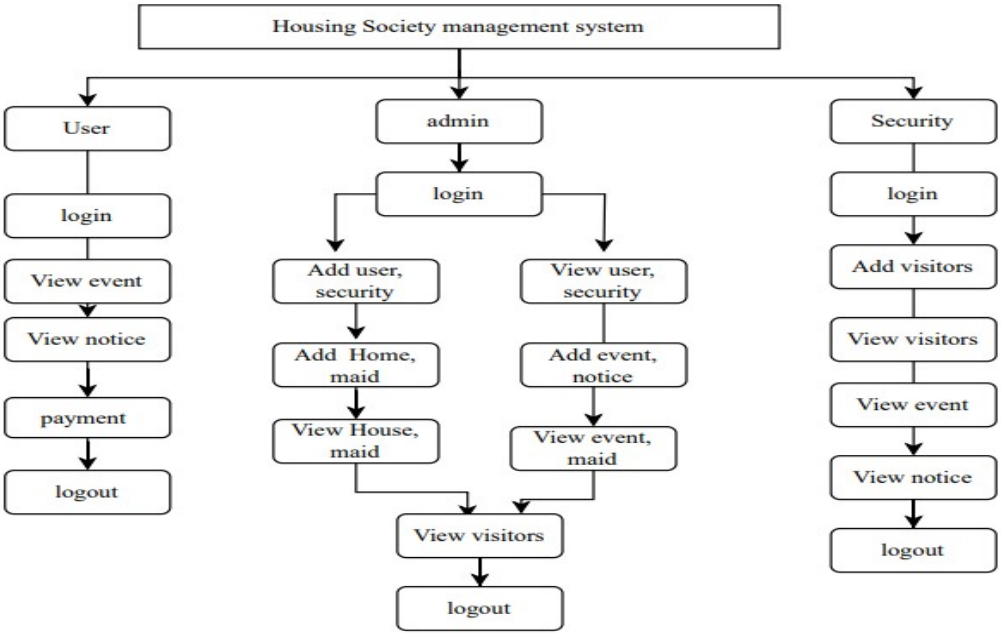

Figure 4 Architecture Diagram

Use Case Representation

[21] A use case representation is a form of behavior representation that is elucidated by and generated from a Use-case investigation. It exists to show a pictorial schema of the capability given by a system. The below 5.1 figure shows the use case representation.

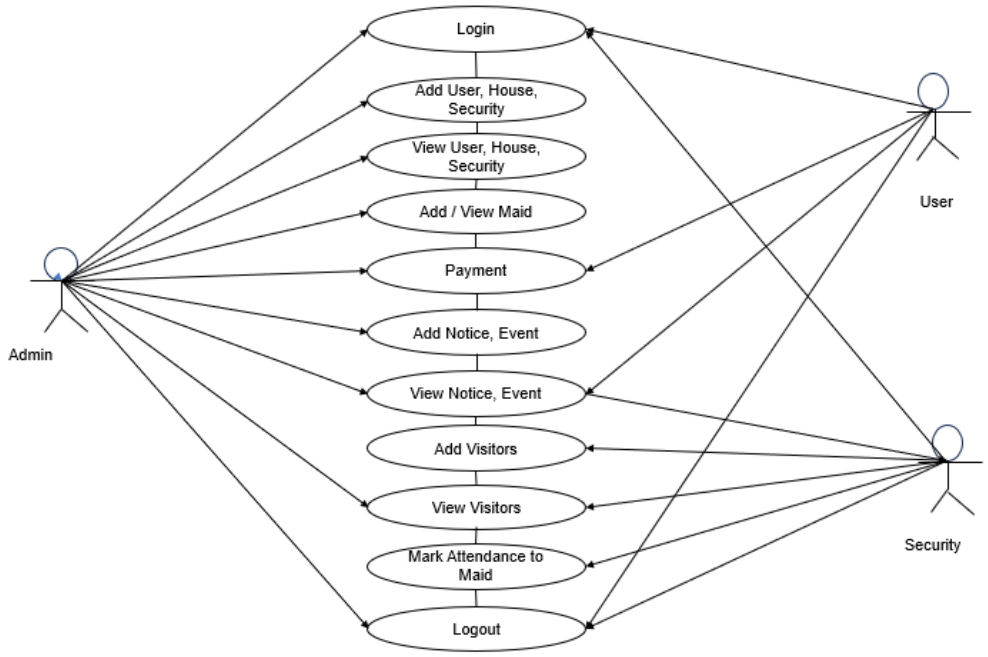

Figure 5. Use Case Representation.

Sequence Representation

[21] A sequence representation is another type of communication representation presenting the way in which operations process operate with each other and the way of their arrangement. The below figure 5.2 indicates the

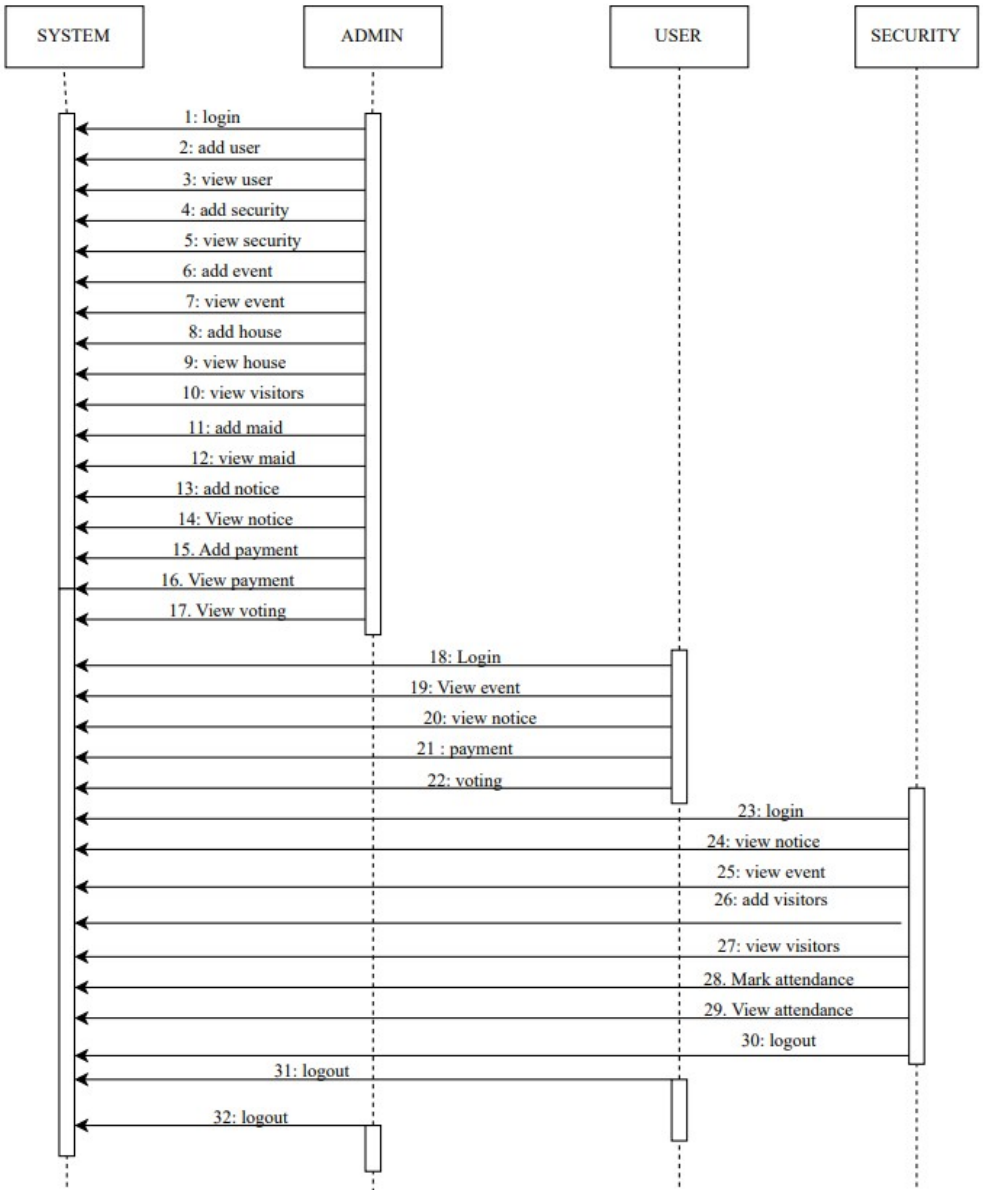

Fig 6. Sequence Representation.

Class Representation

[21] A class representation is a kind of static arrangement representation, which reports the arrangement of a system by presenting the classes, attributes, processes, and the associations between the classes in the system. This

representation shows the class that comprise of data.

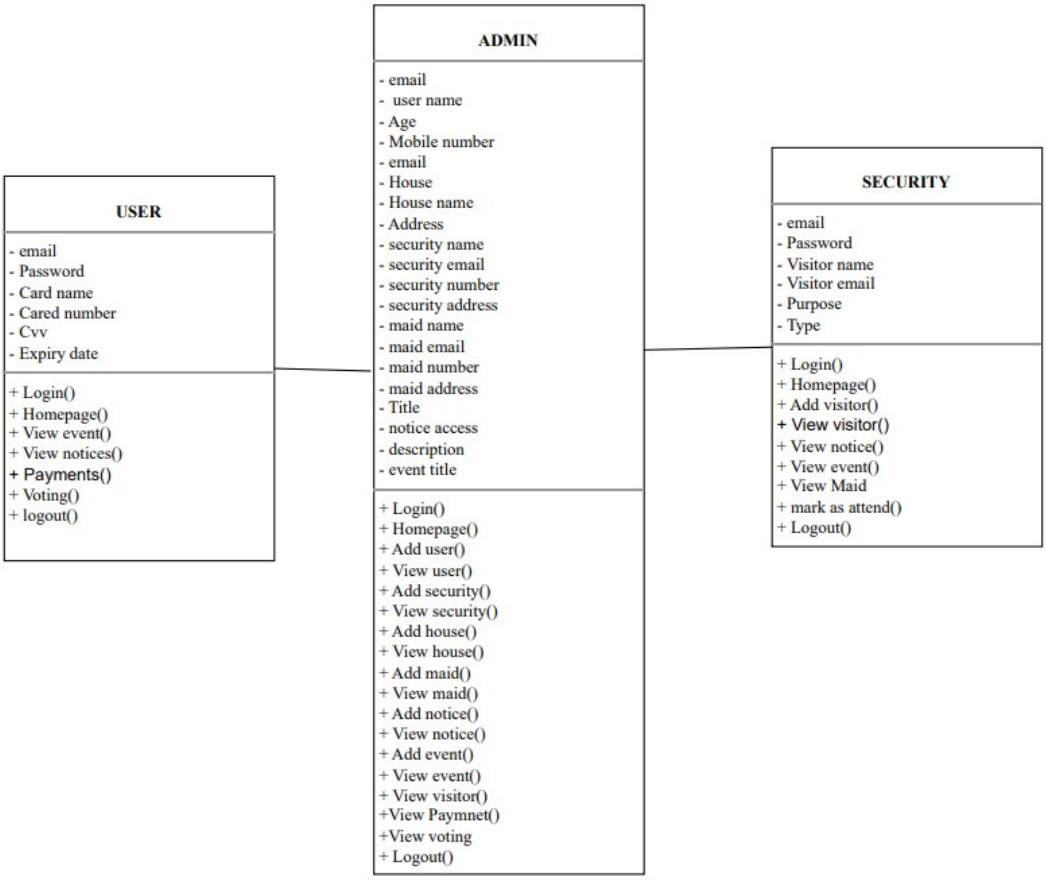

Figure 7. Class Representation.

Activity Representation

[21] Activity representations are pictorial indications of functionalities of step-by-step actions and activities by supporting the concurrency, iteration, and choice. In the UML, representations of activity could be utilized to show the commercial and processing stepwise functionalities of elements contained in a system. This representation exhibits the entire control flow.

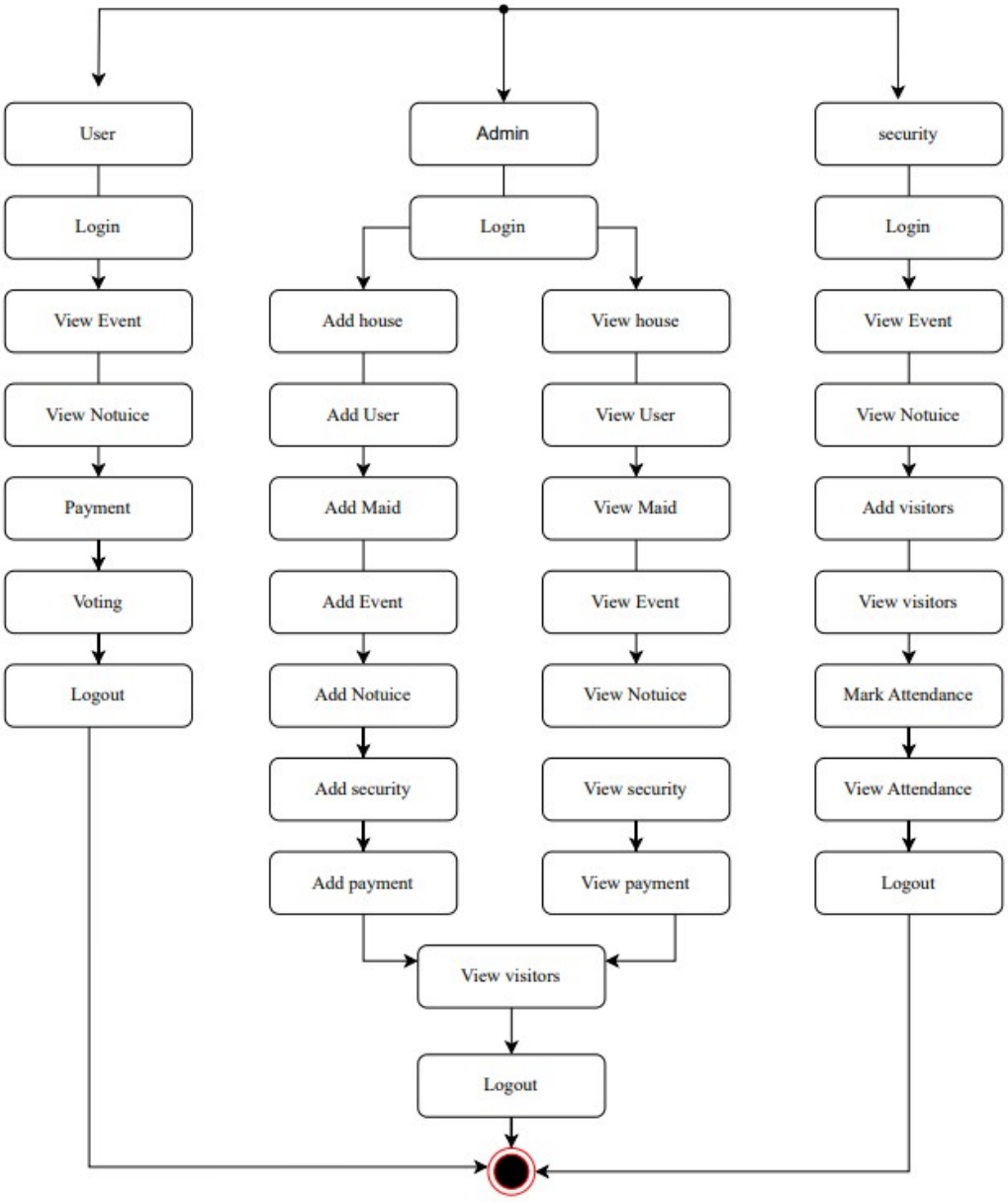

Figure 8. Activity Representation.

ER Representation

[21] An ER model reports the arrangement of a database by a representation called as ER representation. It is a blueprint or design of a database, which could

then be executed as a database. The major elements of this representation include relationship set and entity set.

This type of representation depicts the association between every entity set. An entity set contains a set of identical entities, which has attributes in it. From the sense of an information administration system, an entity is an attribute of a table or table itself in a database. Thus, an ER representation depicts the entire logical arrangement of the database through the indication of association between the attributes and its source tables.

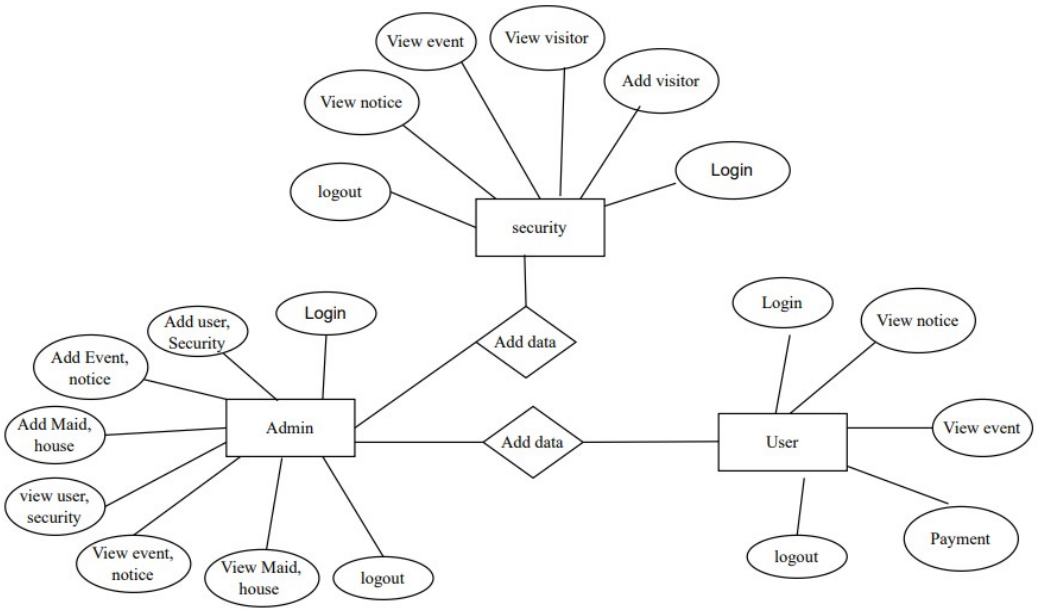

Figure 9. ER Representation.

CHAPTER SIX

SYSTEM ANALYSIS

Proposed System

This application is designed for reducing conflicts present in the multiple management system of housing society. Society members can easily access the ongoings as well as social happenings in society. In this application, three modules are considered, they are: Admin, User and Security. Application server will save profiles in the database, save events, add attendance and save flat bookings etc., Admin modules consists of components like User (add users, view users), Houses (add houses, view houses), Security (add security, view security), Maids (add maids, view maids), Notices (add notices, view notices), Events (add events, view events), Visitors (view visitors), Payments (add payments, view payments) and Voting. User modules consists of components like Events, Notices, Payments and Voting. Security module has components like Visitors (add visitor), Maids (view, mark as attend), Notices and Events.

Web Application

Admin Login

Here, we have created a login module of admin, connected with the database or whose information is already stored in our database. Here, the admin will define User (add users, view users), Houses (add houses, view houses), Security (add security, view security), Maids (add maids, view maids),

Notices (add notices, view notices), Events (add events, view events), Visitors (view visitors), Payments (add payments, view payments) and Voting details. All the essential Notices regarding society will be sent to every society member automatically.

User Login

Here, we have created a user login module for society members to login with their username and password if registered, if not they have to register first, for that admin must add the user. If one of the society members is not registered, then user must register first with the admin for getting username and password. To register the user must fill in the details like name, email id, mobile number, age, house number etc., Once the user login to the application user can access the features like User modules consists of components like Events, Notices, Payments and Voting.

Security Login

Here, even security has to login to the application. It is same as user login, we have created a security login module for security members to login with their username and password if registered, if not they have to register first, for that admin must add the security. If one of the security members is not registered, then security must register first with the admin for getting username and password.

CHAPTER SEVEN

IMPLEMENTATION

Primary Implementation

First, we must set up the development environment to build the project. To set up we must install Node.js version v14.17.3 from the official website. Then open the command prompt and run “npm install -g @angular/cli“ for installation of Angular CLI globally. Then install Xampp which has Apache, MySQL and PHP components in it. After completing the installation set up of the software according to our requirements, we must create a new Angular project.

To create a new Angular project, we must open the terminal or command prompt and select the location for the project. Then run “ng new housing-society-management” command and launch the development server with “ng serve” command.

To design the user interface, we must mention modules and components using Angular CLI commands for the project. Then create the templates, stylesheets, components using HTML, CSS and Angular component logic to build the interface.

Create the RESTful APIs for backend set up using Node.js and Express.js. Various operations such as authentications, managing services, managing residents etc., and define their routes in the specified file. Then connect the backend with the database (MySQL) using the library MySQL and then implement the database model and the CRUD operations. Make sure that

our backend APIs are handling the authentication and authorization appropriately.

Figure 10. Database Connection.

To connect backend and the frontend we must use “HttpClient” module to send request to the backend APIs from the Angular frontend. Then to encapsulate the API actions and data manipulations we must create Angular services.

Finally, to set up the database we must start the Apache and MySQL servers in Xampp application. For visualization of the data in the database we use phpMyAdmin. To access phpMyAdmin use web browser and enter http://localhost/phpmyadmin. Then create a new database for the web-based management system for housing society and define the tables and relationships based on the requirements.

When the Project development is completed, we must test whether the application is running successfully in the local server as per the requirements or not.

Exploring Database

The specific data in the Web Based Management System for Housing society use SQL query language. Open the phpMyAdmin browse the web page using http://localhost:4200/phpMyAdmin. After opening the phpMyAdmin we must click on the database tab and enter the database name and then click on create. Then developers create tables for storing data according to the data structure.

```
CREATE TABLE `admin` (
  `id` int(11) NOT NULL,
  `email` varchar(150) NOT NULL,
  `password` text NOT NULL,
  `name` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Figure 11. Table Creation.

To insert values into the data tables we can use the insert button in the phpMyAdmin interface to manually enter the data one by one or we can write SQL queries to insert values into the data tables.

```
INSERT INTO `admin` (`id`, `email`, `password`, `name`) VALUES
(1, 'kiranmudili@outlook.com', '$2b$10$V5kIaZUofSD4K1rDZvouqORU6H2WboyFE.IvPA9G2qMD8zfDZwihu', 'Kiran Mudili');
```

Figure 12. Inserting Values.

Here you can see the values in the data table. I have included the actual password, but it stores the encrypted password in the data table.

Table 1. Admin Login

S. No	E mail	Password	Name
1	Kiranmudili@outlook.com	Welcome@123	Kiran

Table 2. Notice

SL.NO	Title	Notice access	Description
1	Sample	reb@gmail.com	8574589652
2	Tyson	tyson@gmail.com	9685745632
3	Robin	robin@gmail.com	8965214785

Table 3. Data Table Design

Column name	Data Type and Length
SL.NO	int(20)
Title	varchar(150)
Notice access	varchar(150)
Description	varchar(150)
Email	varchar(150)
Password	varchar(150)
Name	varchar(150)

CHAPTER EIGHT

TESTING

Introduction

Testing is a process of evaluating a software application to identify the effectiveness of an application and checking that the system meets the specified requirements. There are various types of testing that can be performed throughout the software development lifecycle. Some of them are mentioned below and performed in our application.

Unit Testing

Testing individual modules or components in isolation to ensure they work correctly.

Integration Testing

Verifying the interaction between different modules and components when integrated.

Functional Testing

Validating the functional requirements of the system by testing its functionalities.

System Testing

Testing the overall performance of the system to ensure that it meets the requirements.

Application Testing

To test the performance of our application,

Admin Login

Open the admin module in the web browser by navigating “localhost:4200/admin-login”. You can see the login page for the admin module as shown below.

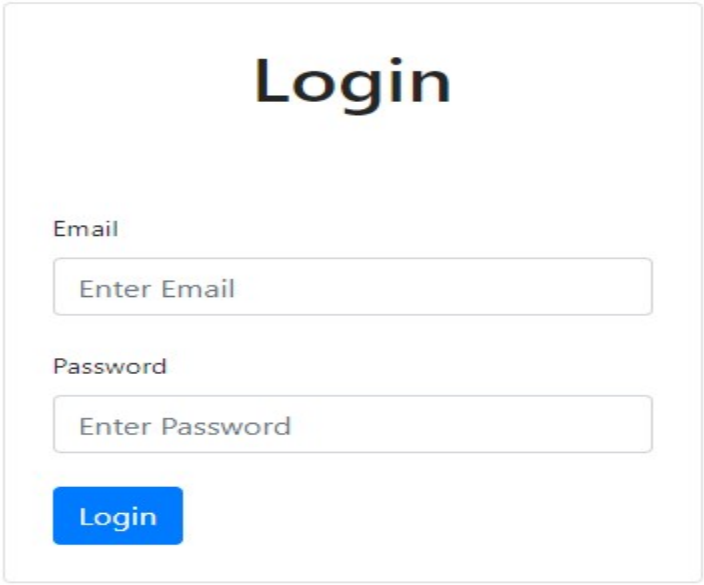

Figure 13. Admin Login Page.

Admin login credentials are created by default at the time of project development and given to admin directly. Admin can login to the admin module using the provided credentials.

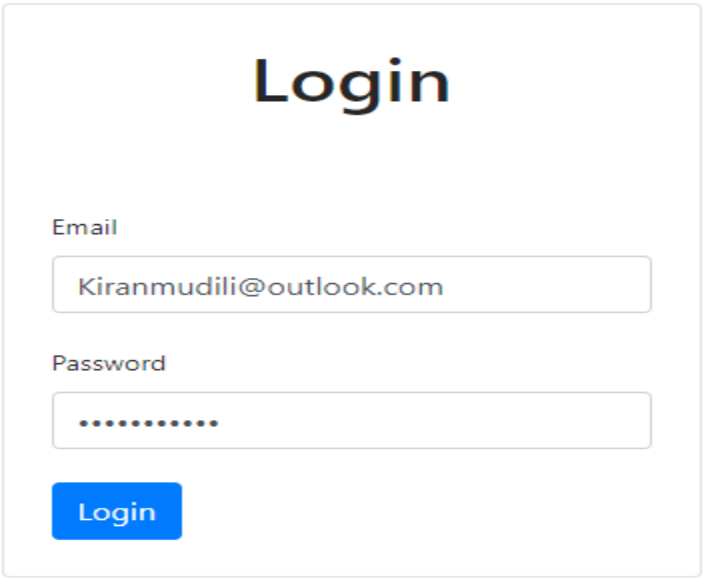

Figure 14. Admin Login

After logging into admin, admin can access the following features or components as shown below. admin can see the Welcome Kiran Mudili (Welcome Username) on the home page.

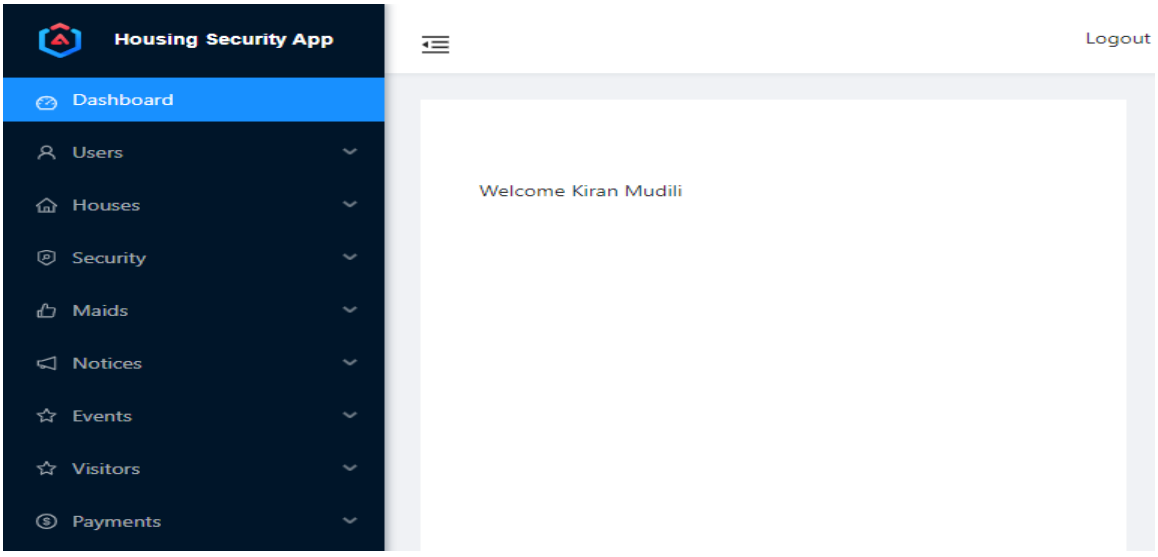

Figure 15. Admin Home Page.

Admin can access the Users (Add User, View Users), where admin can add new user and view users of the society.

To add user, where he needs to give user information like name, age, email, password, mobile number, and select the house.

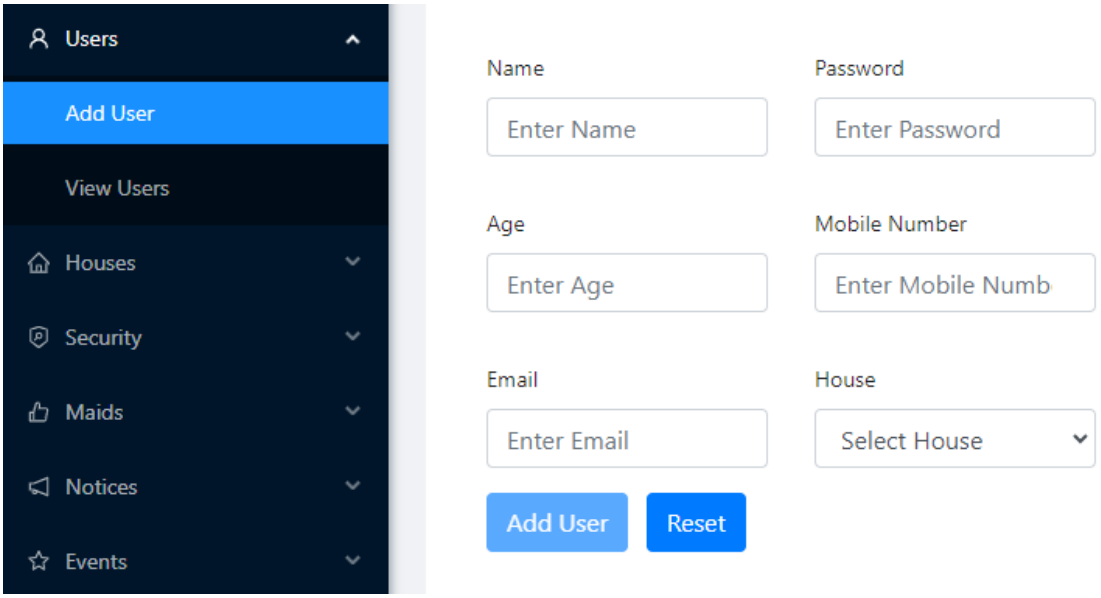

Figure 16. Add User.

In the view users, admin can view the user information like name, email, mobile number, age, house name and address.

Name	Email	Mobile Number	Age	House Name	Address
Kiran Mudili	kiranmudili@outlook.com	6281100477	22	Valentia	D.No:21, h1bApp
Raj	raj@outlook.com	6281100477	22	Ventia-2	sample
sample	mekiranmudili@gmail.com	6281100477	22	sample	address

Figure 17. View User.

Admin can access Houses (Add House, View House), where admin can add the new houses of the society and view houses of the society.

To add house, admin must give the information like house name, password and address.

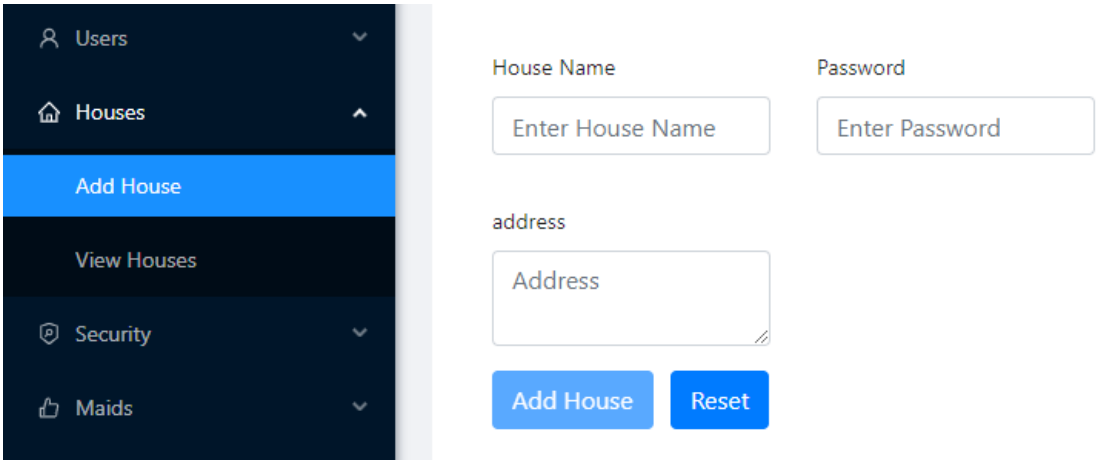

Figure 18. Add House.

In the view house, admin can view the information of house such as house name and house address.

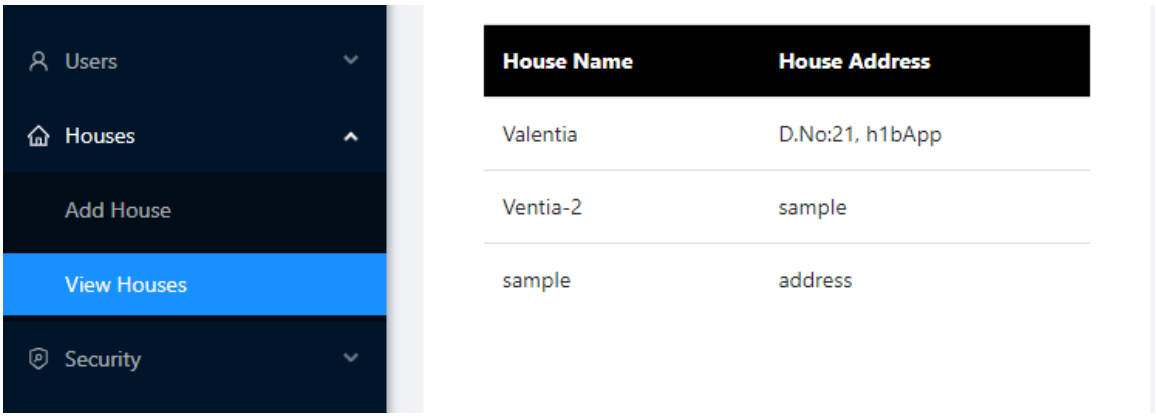

Figure 19. View Houses.

Admin can access Security (Add Security, View Security), where admin can add new security to the society and view the security of the society.

To add security, admin must fill in the details of security like name, email, password, mobile number and address.

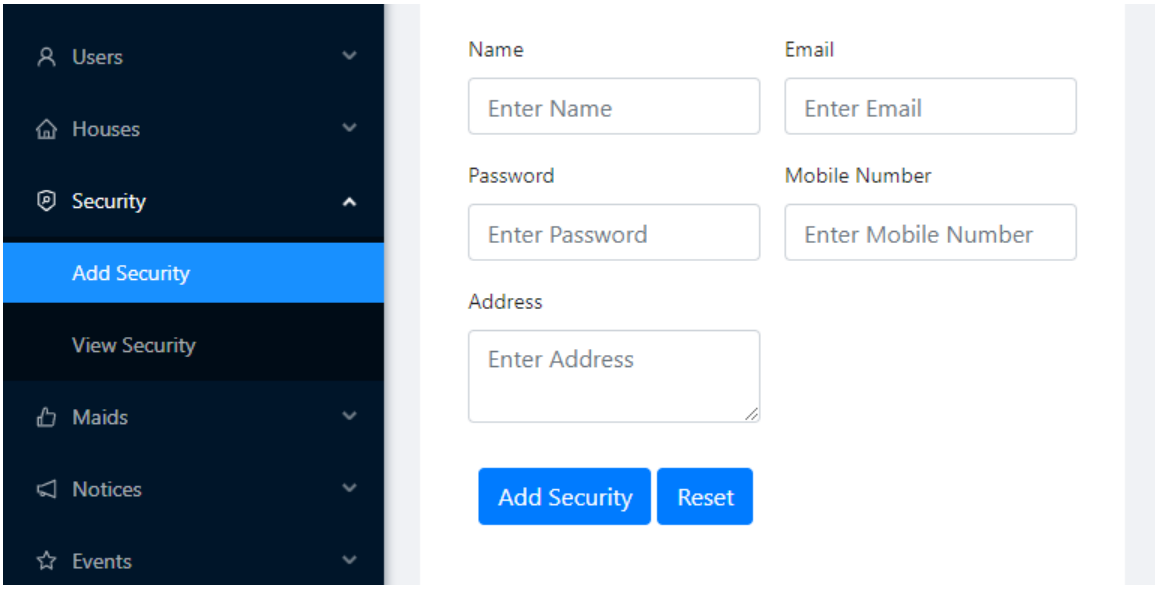

Figure 20. Add Security.

In the view security, admin can find security information like name, email, mobile number and address.

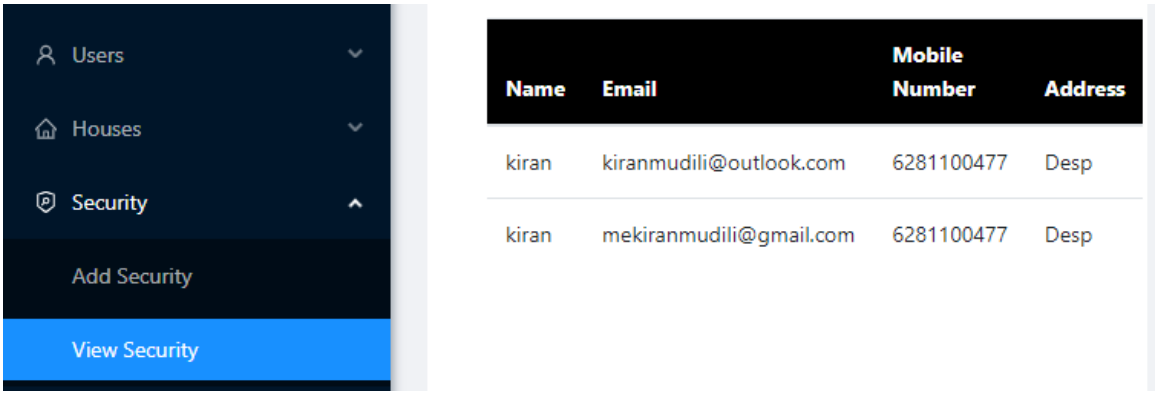

Figure 21. View Security.

Admin can access Maids (Add Maid, View Maids), where admin can add new maids to the society and view the maids of the society.

To add maids, admin must fill the details of maid's name, email, mobile number and address.

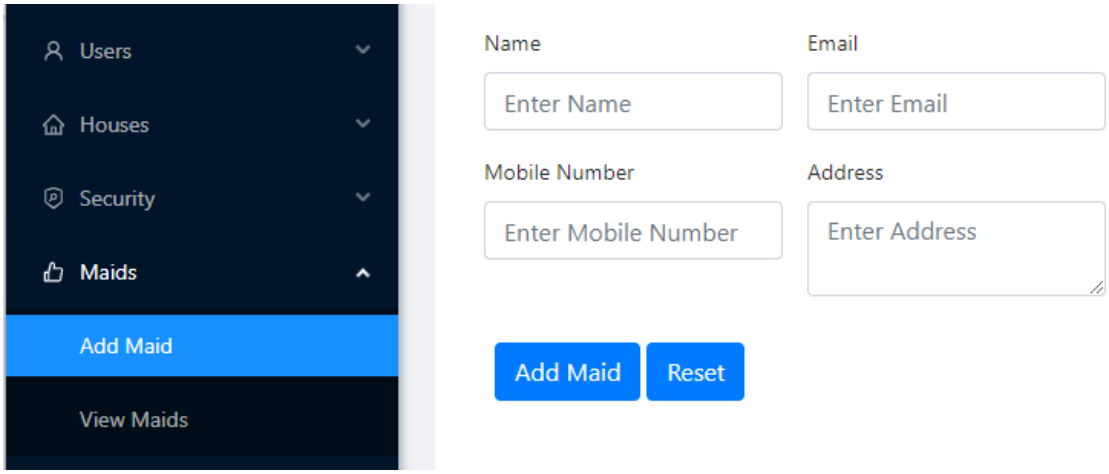

Figure 22. Add Maid.

In the view maids, admin can find maids information like name, email, mobile number and address.

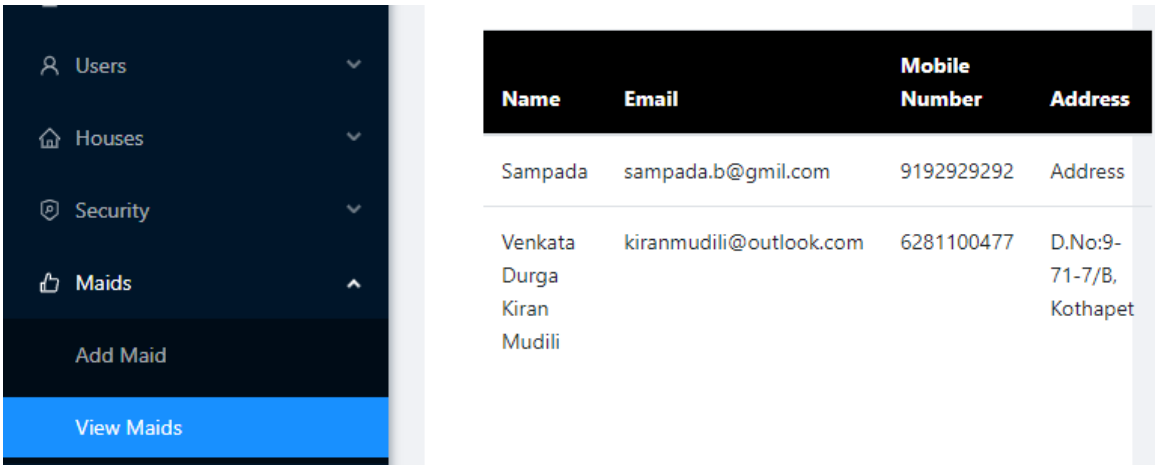

Figure 23. View Maid.

Admin can access Notices (Add Notice, View Notices), where admin can add new notices to the society members and view the past notices of the society.

To add notices, admin must fill in the details like title of the notice, notice access and description.

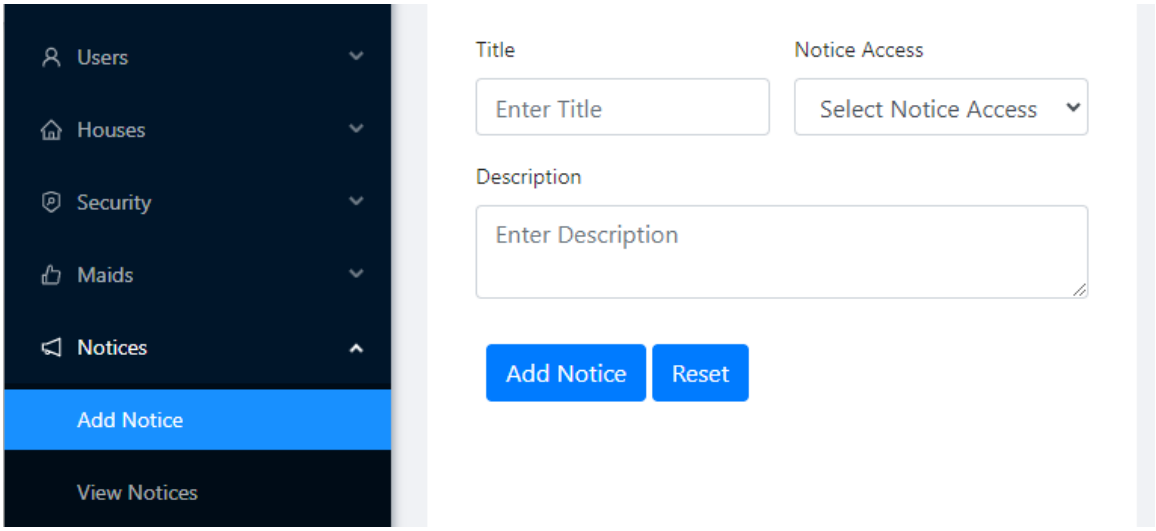

Figure 24. Add Notice.

In the view notices, admin can view the details regarding notice like title of the notice in the field of name, description, who can access the notice in the field notice access and the notice received by in the accessor field.

Figure 25. View Notice.

Admin can access Events (Add Events, View Events), where admin can add the events and view the events of the society.

To add the events admin must fill in the details such as title, event date and description.

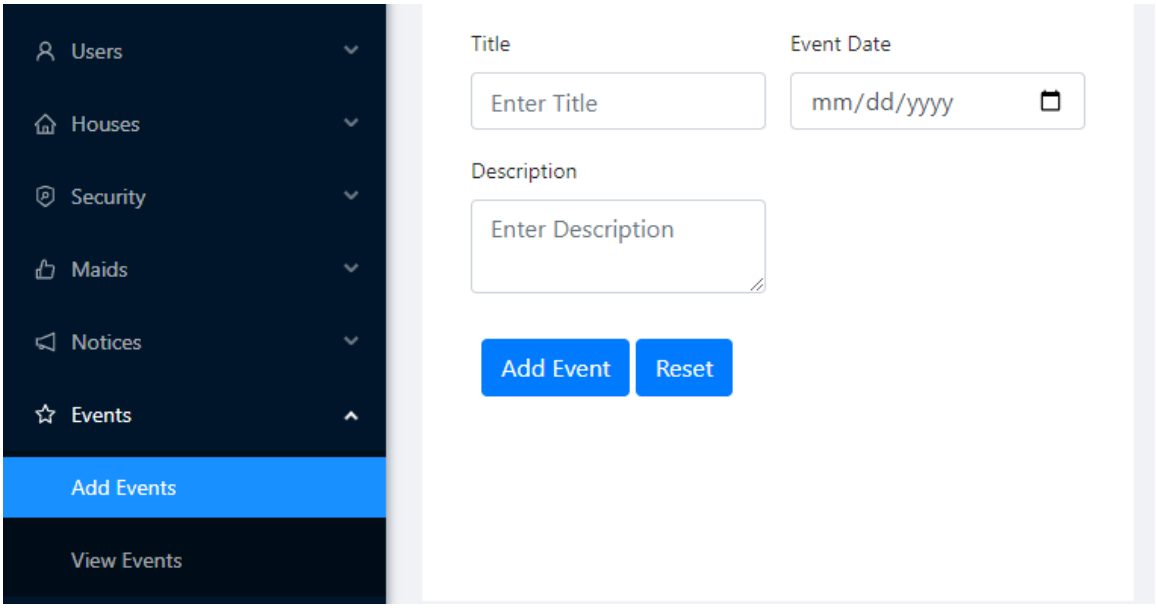

Figure 26. Add Events.

In view events, admin can find title of the event, event date and description of the event.

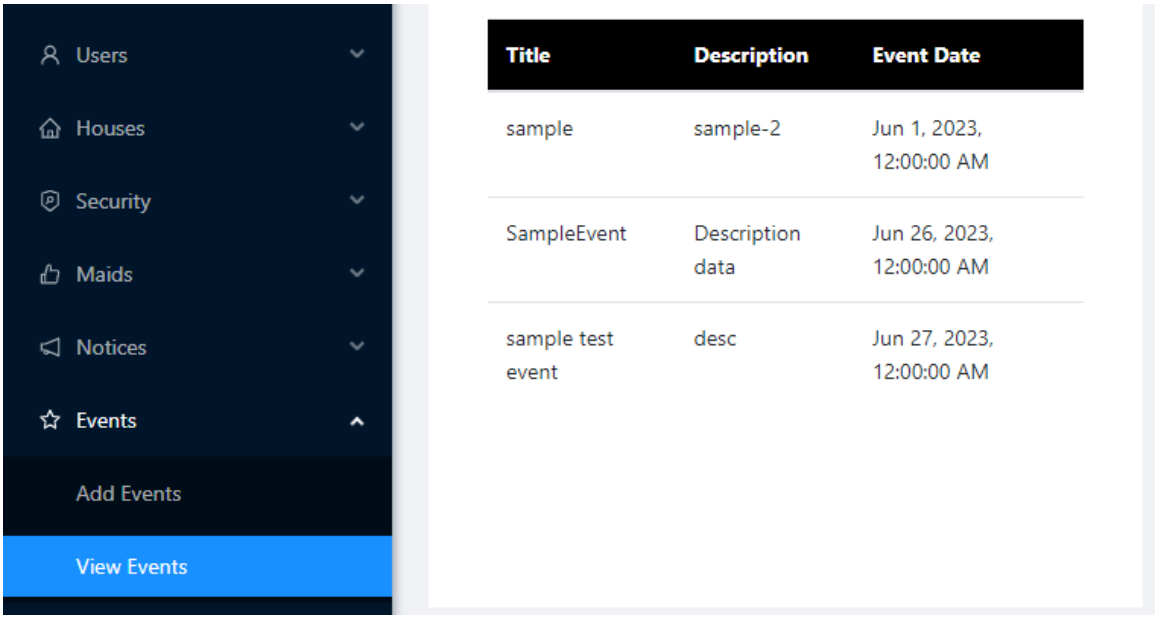

Figure 27. View Events.

Admin can access Visitors (View Visitors), where admin can view the visitors of the society which is added by the security in the security login.

In the view visitor's admin can see the details of visitors like visitor name, purpose of the visit, type of visitor and the entry time.

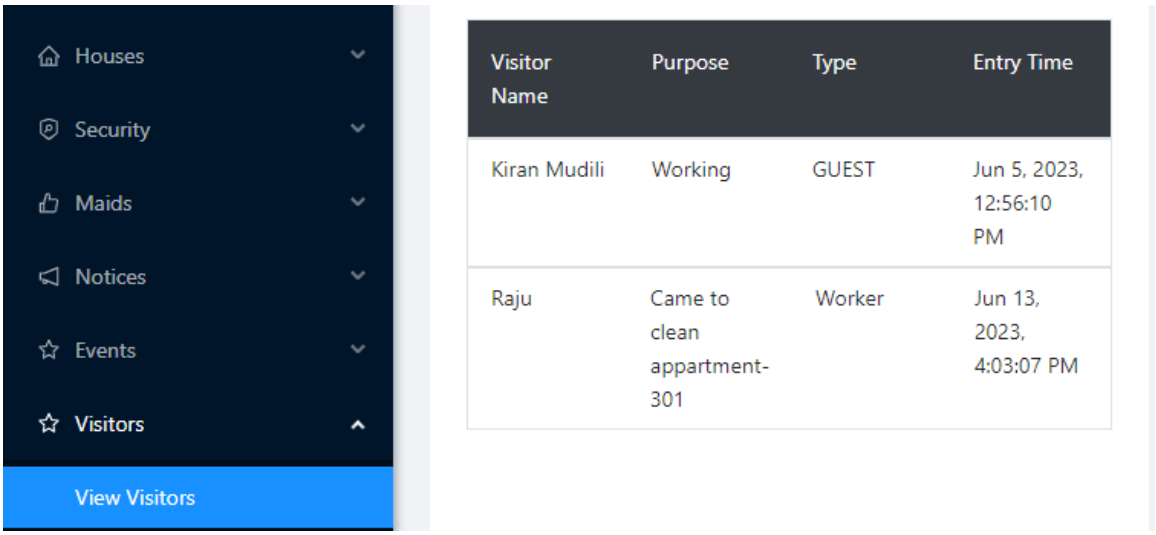

Figure 28. View Visitors.

Admin can access Payments (View payments, add payments), where he can view the payments of the society members and add payments to the society.

In the view payments, admin can view the details of payments like payment title, house name, description, amount, status and due date.

Payment Title	House Name	Description	Amount	Status	Due Date
Maintainence	Valentia	sample	$ 1000	Completed	Jun 1, 2023, 12:00:00 AM
Rent	Valentia	sample	$ 2000	Completed	Jun 1, 2023, 12:00:00 AM
sample	Valentia	ww	$ 2000	Completed	Jun 30, 2023, 12:00:00 AM
Housing Rent	sample	Here is your house rent.	$ 1000	Pending	Jul 8, 2023, 12:00:00 AM
maintanence	Ventia-2	..	$ 1000	Pending	Jun 30, 2023, 12:00:00 AM
maintenence	Valentia		$ 1500	Pending	Jun 30, 2023, 12:00:00 AM
maintenence	Valentia		$ 1000	Pending	Jun 30, 2023, 12:00:00 AM

Figure 29. View Payments.

To add payments the admin must provide details like title, amount, deadline, house and description.

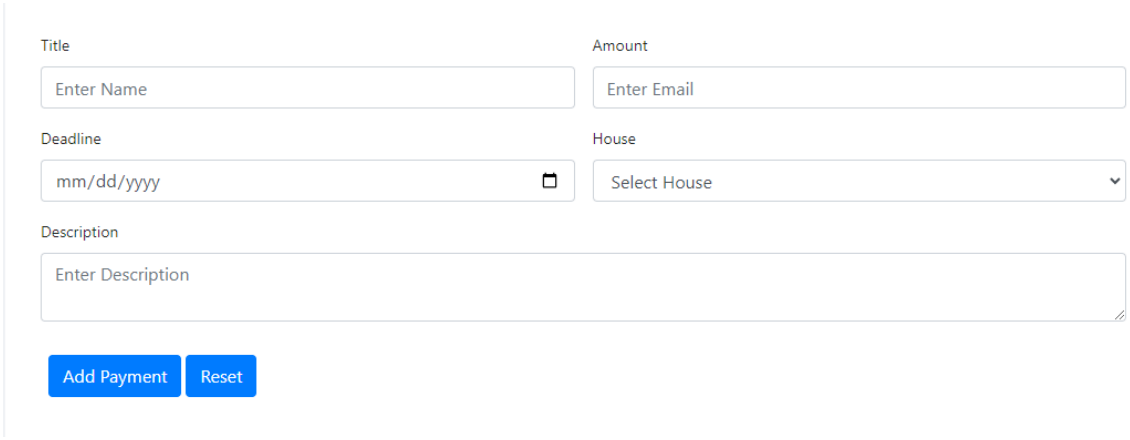

Figure 30. Add Payments.

Once the details are filled click on the add payment button, the payment is added successfully.

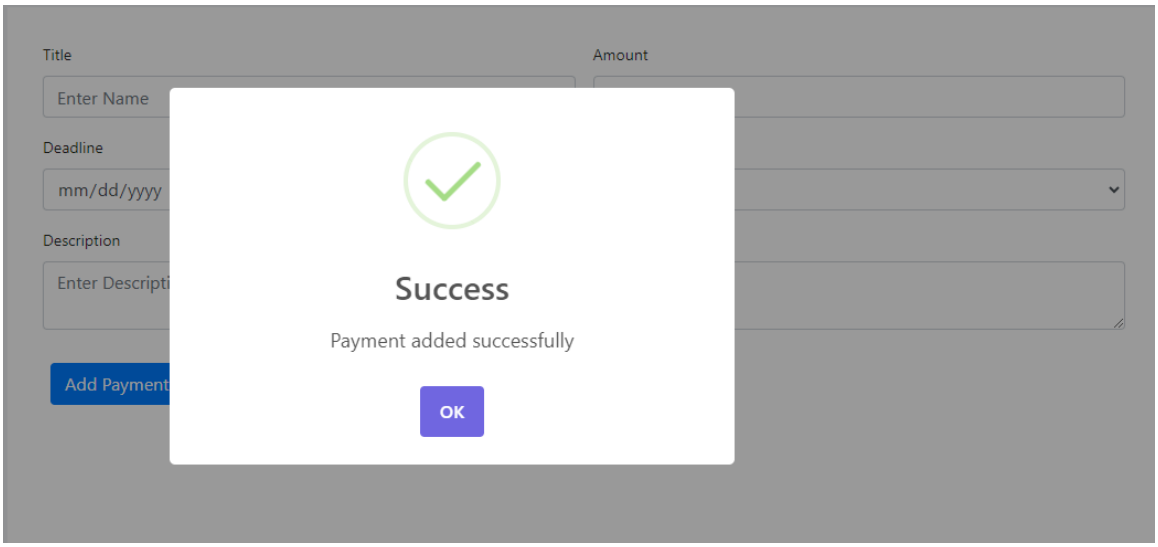

Figure 31. Payment Added Successful.

If the admin has logged out of the page, he must click on the logout button, and he will be successfully logged out.

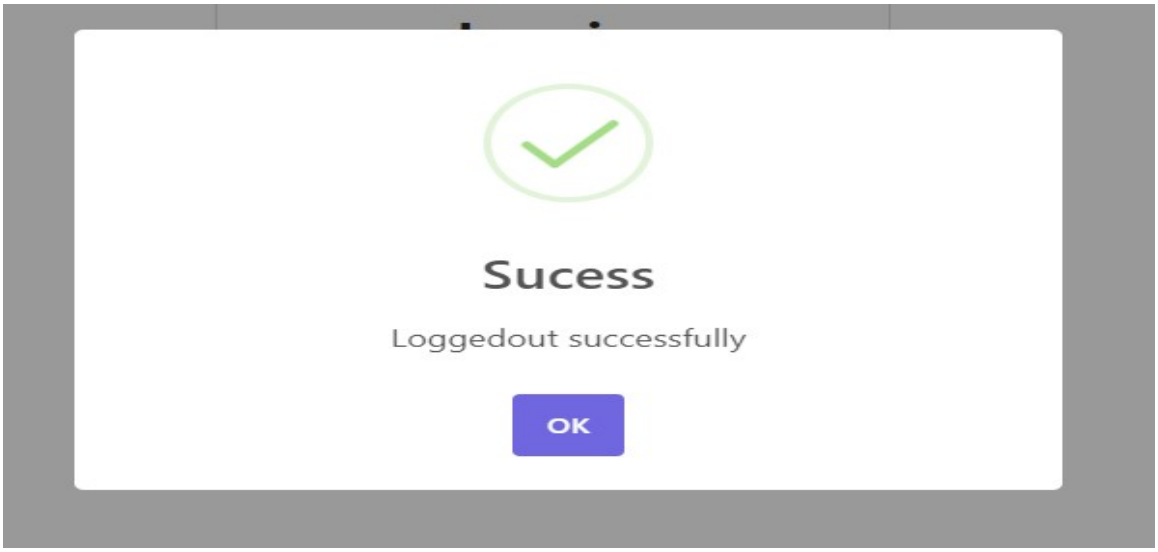

Figure 32. Logout.

User Login

To user module in the web browser by navigating “localhost:4200/user-login”. You can see the login page for the user module as shown below.

To login, the user must provide the registered credentials like email and password. If the user is not registered, he must first register with the admin and get the credentials.

Figure 33. User Login.

If the invalid credentials are entered, then the error pops up as shown below.

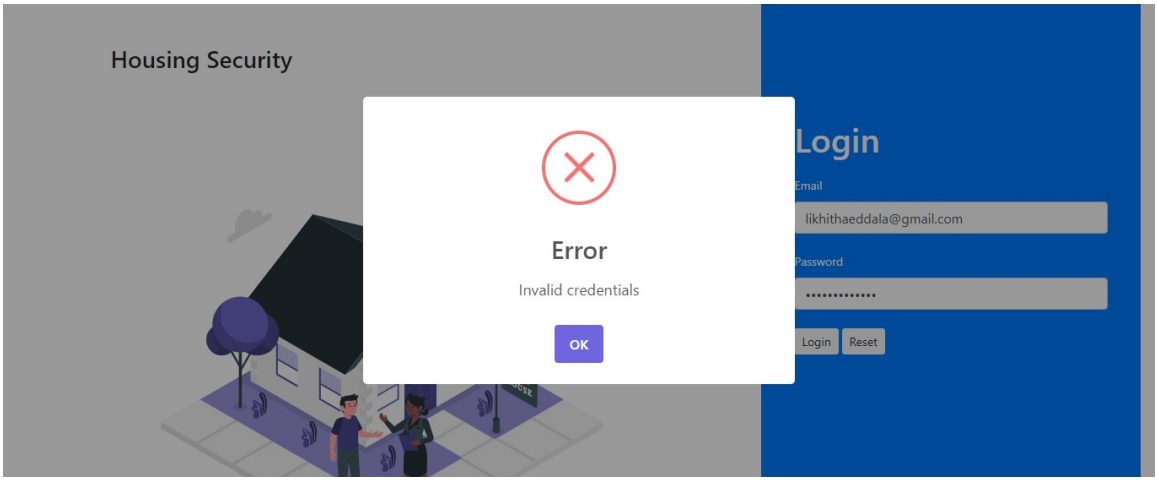

Figure 34. Invalid Credentials.

After logging into the user, the user can see the Welcome Kiran Mudili (Welcome Username) on the home page.

Figure 35. User Home page

User can access the events, If the admin has added this user to events then those events will be shown under events as shown below.

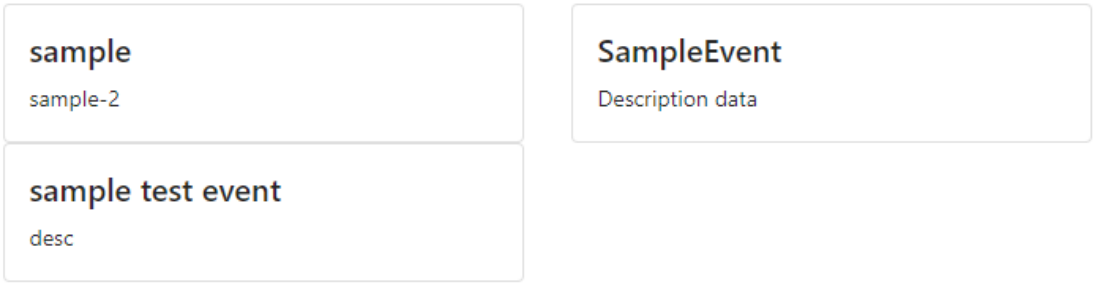

Figure 36. User Events.

User can access notices, if this user is added to notices by the admin, then those notices will be shown here as below.

Figure 37. User Notice.

User can access payments, if he is added with any payments by the admin then that will be shown here as below and must perform action according to it.

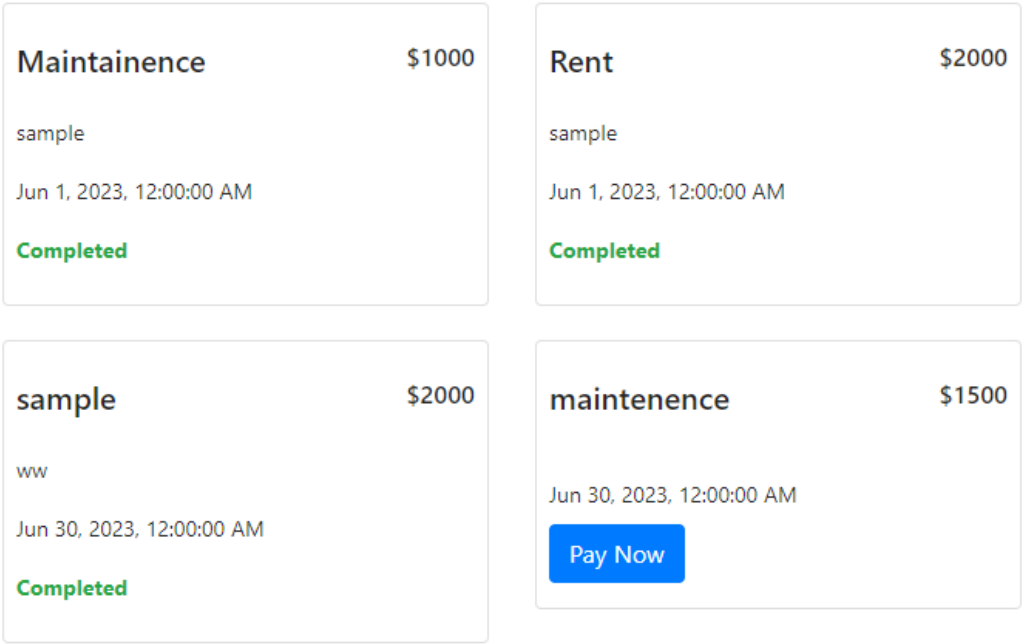

Figure 38. Payments.

Once the payment is completed the payment done successfully pop-up will be triggered as below.

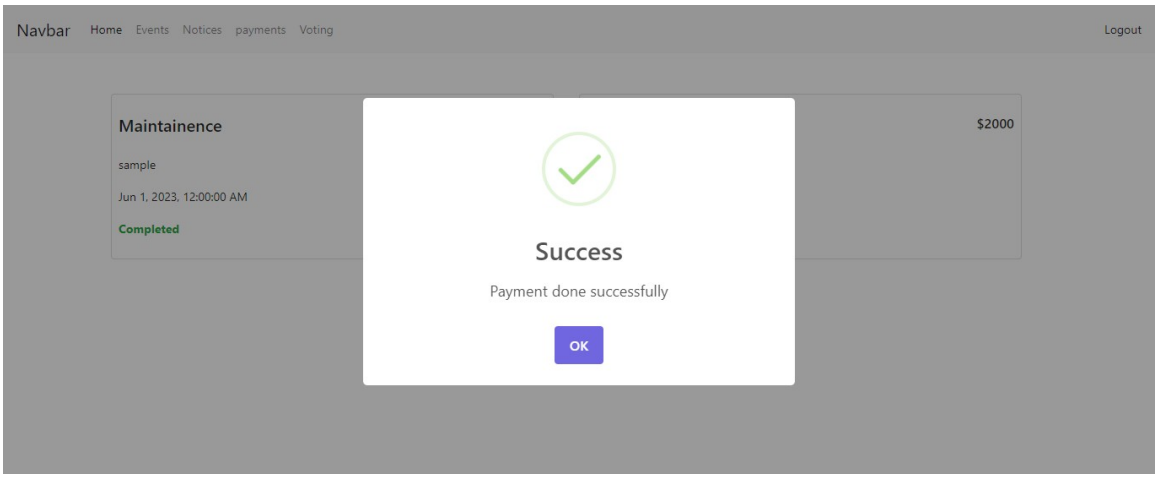

Figure 39. Payment Success.

User can access the voting form where they can poll their decision if he is added by the admin as below.

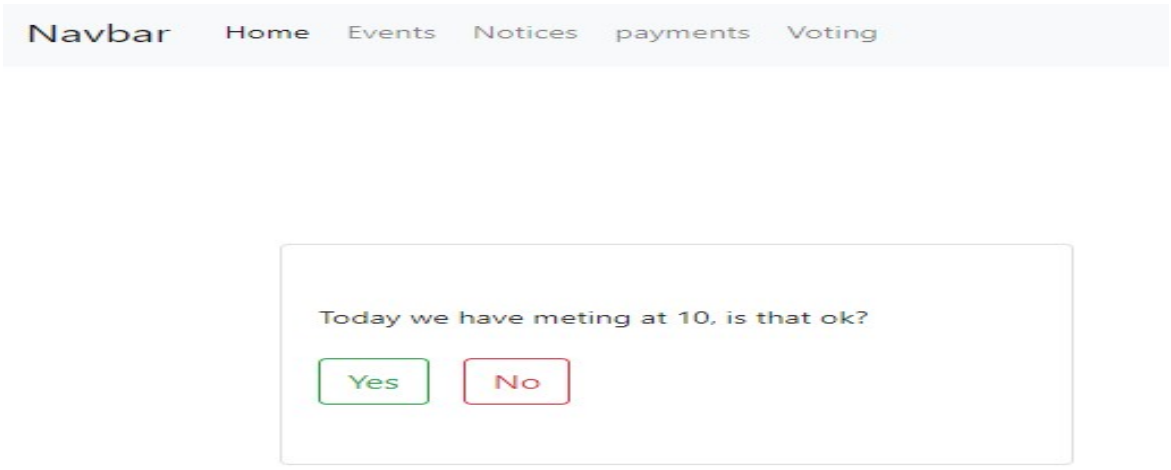

Figure 40. Voting Form.

Once the decision is polled, added vote successfully pop-up triggers as shown below.

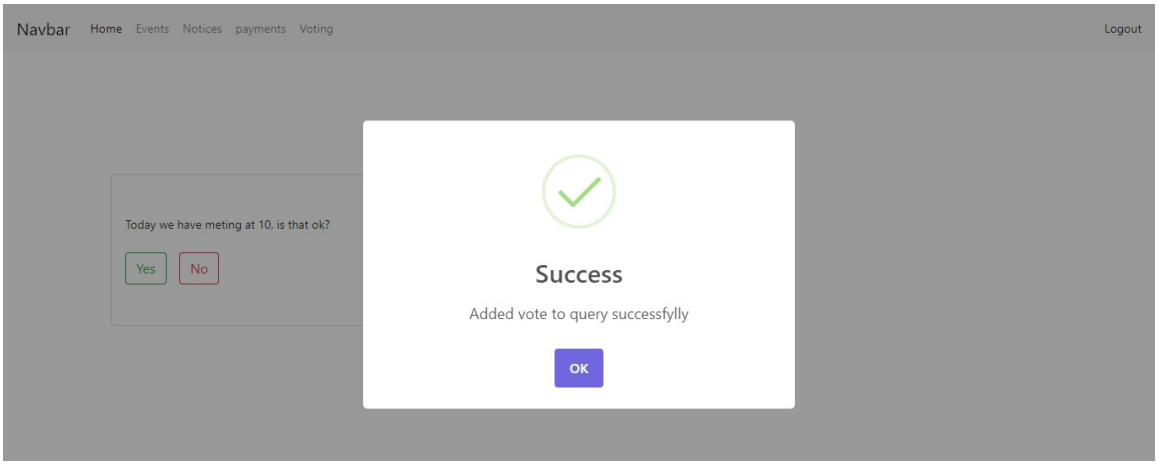

Figure 41. Voting Successful.

Security Login

Open the Security module in the web browser by navigating “localhost:4200/security-login”. You can see the login page for the security module as shown below.

To login, the security must provide the registered credentials like email and password. If the security is not registered, he must first register with the admin and get the credentials.

Figure 42. Security Login.

After logging into the security, the security can view the visitors and add visitors as shown below.

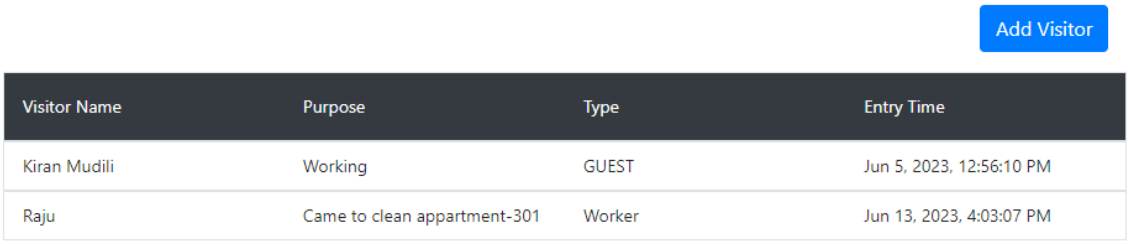

Figure 43. Visitors.

To add visitors' security must fill in the details like visitor name , type and purpose of the visit.

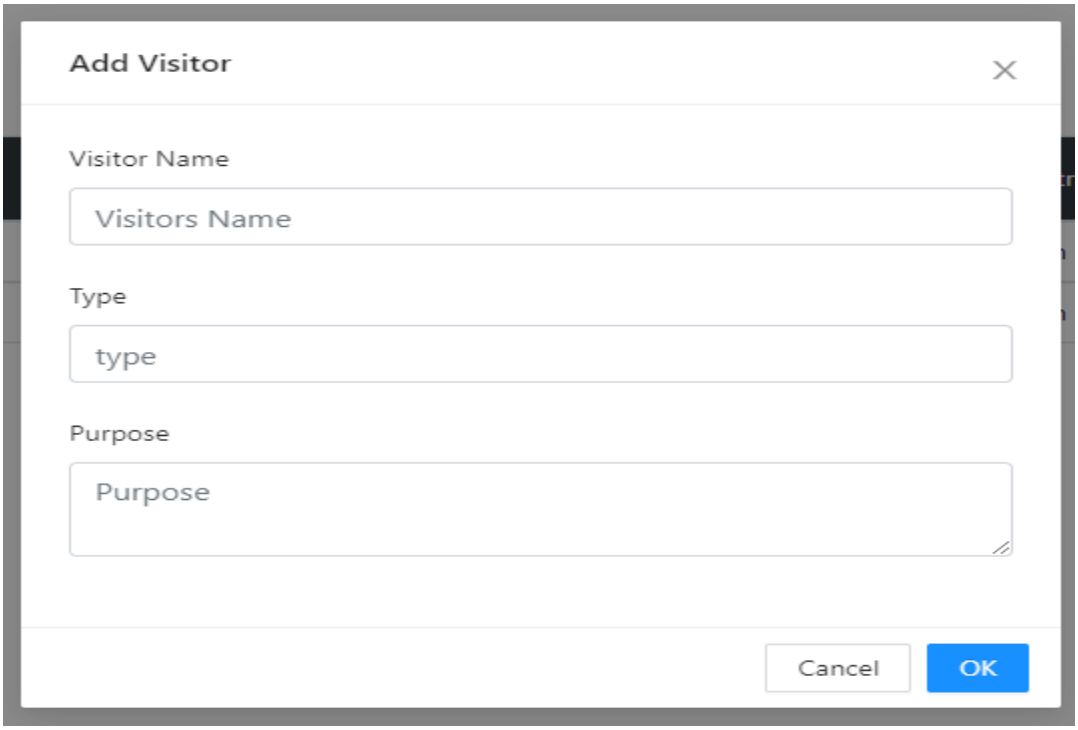

Figure 44. Add Visitors.

Once the visitor is added successfully the pop-up message will trigger as shown below.

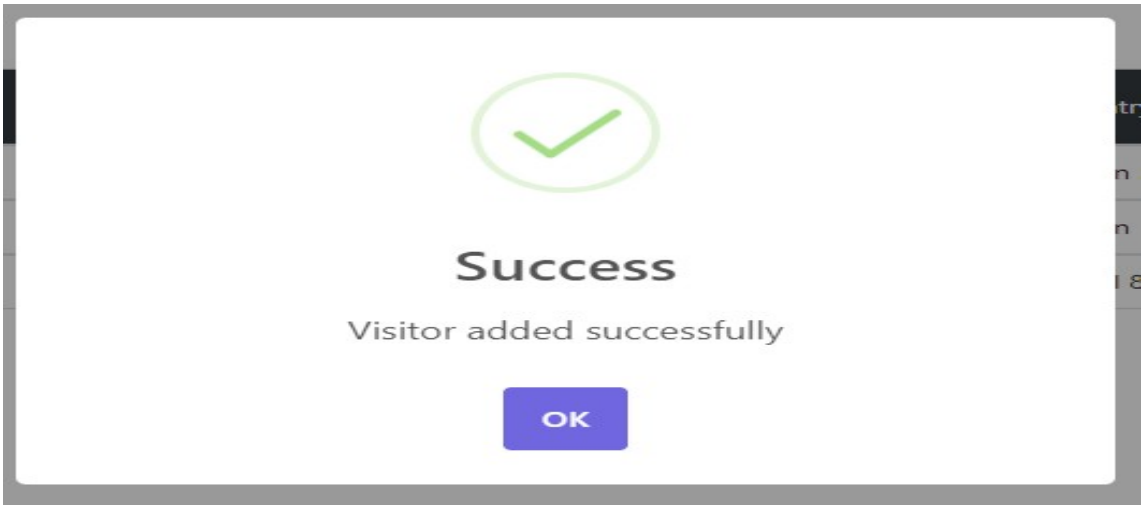

Figure 45. Visitor Added successful .

Security can access maid where he can view the security and mark attendance to the security as shown below.

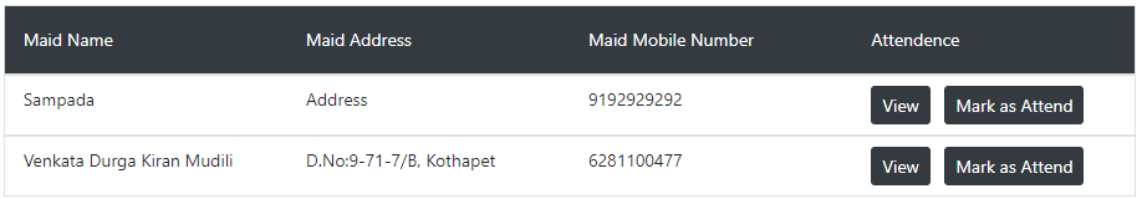

Figure 46. Mark Attendance of Maid.

Once the attendance is marked successfully, we can view the attendance of the maid as shown below.

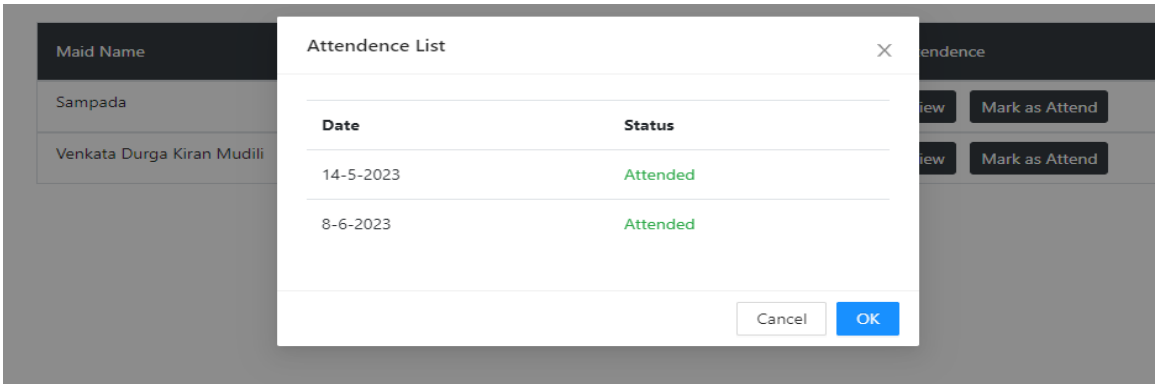

Figure 47. View Attendance of Maid.

Security can access notices, if the admin has sent notices to this security, then he can view the notices as below.

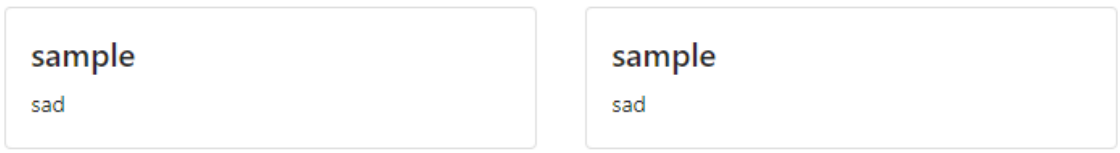

Figure 48. Security Notices.

Security can access event, if the admin has added the events, then he can view those events here as below.

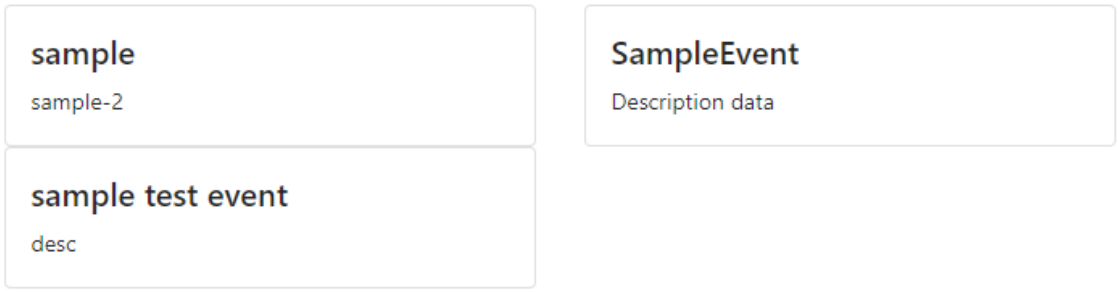

Figure 49. Security Events.

CHAPTER NINE

CONCLUSION

The Web-Based Management System for Housing Society has brought significant improvements to the management and operations of housing societies. The system has the capability of addressing the challenges faced by the people in the community. This system also provides a user-friendly environment and built as an efficient platform to streamline various tasks and enhance communication within the housing society.

The housing society management system has improved a lot in efficiency, transparency, and resident satisfaction. Regular testing, feedback and analyzing the problems and making improvements have ensured a robust and reliable solution. As the housing society landscape evolves, the Web-Based Management System can serve for further enhancements and future developments to improve the operational efficiency of the system.

By leveraging the power of technology, the system has better governance, communication within housing societies, by providing secure and safe living environment for residents.

CHAPTER TEN

FUTURE ENHANCEMENTS

By making the interface more intuitive and user-friendly, residents and housing management can have a better user experience.

The system can be implemented with a responsive design and ensure that it is compatible with different devices. Notification systems can be implemented or developed to inform residents about important updates and announcements by sending text messages so that the residents get updates. visual elements such as charts and graphs can be implemented so that data can appear to be more visually appealing manner. conduct security audits and updates frequently to ensure the system remains secure and up to date without any issues.

Implement a two-factor authentication for user login to prevent unauthorized access so that the user data will be safe and secured. Strengthen the system's security measures to protect sensitive resident information.

Regular user surveys, feedback collection, and collaboration with the management committee will help to identify and improve future development of the system.

APPENDIX A

CODE

TypeScript: main.ts (Frontend)

```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.error(err));
```

CSS: Style.css

```
.table thead {
    background-color: black !important;
    color: white !important;
    font-weight: bold;
}
```

TypeScript: test.ts

```
// This file is required by karma.conf.js and loads recursively all the .spec and
framework files

import 'zone.js/testing';
import { getTestBed } from '@angular/core/testing';
import {
  BrowserDynamicTestingModule,
  platformBrowserDynamicTesting
} from '@angular/platform-browser-dynamic/testing';

declare const require: {
  context(path: string, deep?: boolean, filter?: RegExp): {
  keys(): string[];
  <T>(id: string): T;
  };
};

// First, initialize the Angular testing environment.
getTestBed().initTestEnvironment(
```

```
  BrowserDynamicTestingModule,
  platformBrowserDynamicTesting(),
  { teardown: { destroyAfterEach: true }},
);

// Then we find all the tests.
const context = require.context('./', true, /\.spec\.ts$/);
// And load the modules.
context.keys().map(context);
```

JavaScript: Admin.controller.js (Backend)

```
const AdminModel = require("../Models/Admin.model");
const bcryptHeler = require("../Helpers/bcrypt.helper");
const jwtHelper = require("../Helpers/jwt.helper")
const AdminController = {};
AdminController.loginAdmin = async(req, res) => {
    try {
        let email = req.body.email;
        let password = req.body.password;
        AdminModel.getEmailByAdmin(email).then(async(response) => {
        let adminDetails = JSON.parse(JSON.stringify(response[0]));
        let check = await bcryptHeler.comparePassword(password,
adminDetails.password);
        console.log(check)
        if (check) {
        delete adminDetails.password
        token = await jwtHelper.generateToken(adminDetails);
        res.send({ token: token, adminDetails: adminDetails });
        } else {
        res.send({ status: "error", message: "Unable to login password" })
        }
        }).catch((e) => {
        console.log(e);
        res.send({ status: "error", message: "Unable to login" })
        })
    } catch (e) {
        console.log(e);
        res.send({
        status: "error",
        message: "Unable to login with admin credentials"
        })
    }
}
```

```
AdminController.addAdmin = async(req, res) => {
    try {
        let adminDetails = req.body;
        adminDetails["password"] = await
bcryptHeler.encryptPassword(adminDetails["password"])
        AdminModel.addAdmin(adminDetails).then((response) => {
        res.send({ status: "success", message: "Admin added successfully" })
        }).catch(e => {
        console.log(e)
        res.send({ status: "error", message: "Unable to add admin" })
        })
        } catch (e) {
        res.send({
        status: "error",
        message: "Unable to insert admin"
        })
    }
}

AdminController.getAdmins = (req, res) => {
    try {

      } catch (e) {
      console.log(e);
      res.send({ status: "error", message: "Unable to get admin details" })
    }
}

module.exports = AdminController;
```

REFERENCES

[1] D. J. A. p. e. p. m. Paris, "Action publique et projet métropolitain," pp. 1-380, 2006.

[2] R. Bhagwat, A. Bharadwaj, V. Harsode, A. Chawake, and D. J. I. R. J. E. T. Bhanage, "Society management application on android," 2018.

[3] T. J. I. J. o. C. A. Shah, "Cloud based Housing Society Management System," vol. 179, no. 47, pp. 35-41, 2018.

[4] R. I. Nurimbetov, T. A. Khasanov, A. S. J. T. Sultanov, and A. Science, "IMPROVEMENT OF THE MANAGEMENT SYSTEM OF HOUSING AND COMMUNAL SERVICES IN UZBEKISTAN," no. 3, pp. 66-71, 2019.

[5] S. Nahrath, J.-D. Gerber, P. Knoepfel, and C. J. N. Bréthaut, Science, Société, "Le rôle des institutions de gestion communautaire de ressources dans les politiques environnementales et d'aménagement du territoire en Suisse," vol. 20, no. 1, pp. 39-51, 2012.

[6] O. Barreteau, "Working Together: Collective Action, the Commons, and Multiple Methods in Practice, by Poteete, Amy R., Janssen, Marco A. and Ostrom, Elinor," 2011.

[7] S. K. C. A. H. S. S. K. S. Kadu, "Housing Society Management," (Accessed on June 29, 2023), 2023.

[8] S. Gavhane, R. Vatharkar, S. Sonar, and P. J. I. J. o. C. A. Patil, "Study of Implementation of Society Management System," vol. 132,

no. 1, pp. 34-36, 2015.

[9] V. K. Ishwari Niphade, Abhipray Jawanjal, Nikhil Kadam, and Ranjana Dahake, "Societyconnect App: E-Manager For Residential Society," International Journal of Creative Research Thoughts (IJCRT), vol. 11, no. 5, 2023.

[10] J. P. Pritesh A. Patil, Kirti Bhatane, Sakshi Raskar, and Saurabh Ruikar, "AI BASED SOCIETY MANAGEMENT APP," International Journal of Creative Research Thoughts (IJCRT), vol. 10, no. 5, 2022.

[11] P. Pakhale, S. Shirke, S. J. I. J. f. R. i. A. S. Dhake, and E. Technology, "Online Housing Society Management System," 2016.

[12] R. Vatharkar, P. Patil, S. Sonar, and S. J. I. J. S. T. Gavhane, "Implementation of society management system: societales," 2016.

[13] P. P. Jui Hande, Neha Hasan, and Dhanashree Date, "Study of Housing Society Management System," International Journal of Computer Trends and Technology (IJCTT), vol. 41, no. 2, 2016.

[14] D. Vora and R. J. I. J. A. R. C. E. T. Shah, "Emergency management system using android application," vol. 6, no. 2, pp. 2278-1323, 2017.

[15] R. Jadhav, J. Patel, D. Jain, S. J. I. J. o. C. S. Phadhtare, and I. Technologies, "Emergency management system using android application," vol. 5, no. 3, pp. 2803-2805, 2014.

[16] V. Harsh, T. Shubham, D. Ritesh, and R. Mansing, "Housing Society Management System Using PHP," in Computer Networks and

Inventive Communication Technologies: Proceedings of Fifth ICCNCT 2022: Springer, 2022, pp. 269-281.

[17] S. Raut, P. Pawar, M. Shaikh, N. Bhat, and P. Kalavadekar, "Housing Society Management Web Application with recommendation system," ed: January, 2017.

[18] A. S. Adukathil Arjun, Parui Priydarshan, Chakraborty Shriyansh, and Sangeetha Selvan, "SURVEY PAPER ON HOUSING SOCIETY MANAGEMENT SYSTEM," International Research Journal of Engineering and Technology (IRJET), vol. 8, no. 3, 2021.

[19] A. Chaudhari, A. Kuriakose, P. Gaikwad, A. Kadam, S. J. I. J. f. R. i. A. S. Sambherao, and E. Technology, "Society Management System using Web Technologies," vol. 9, no. 2, pp. 548-553, 2021.

[20] Y. P. Abhishek sawalkar, Niraj Patel, and Harsh Bhor, "Housing Society Management System Using IoT," International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET), vol. 10, no. 5, 2021.

[21] Lucidchart, Visual Paradigm, UML diagram sites.
https://www.lucidchart.com
https://www.visual-paradigm.com