A PROJECT ON

Online Society Management System

SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENT

FOR THE COURSE OF DIPLOMA IN ADVANCED COMPUTING FROM CDAC

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY

Hinjawadi

SUBMITTED BY:

Hitesh Kolhe,

Aniruddha Suryawanshi,

Soham Mirajgaonkar,

Muskan Shaikh

UNDER THE GUIDENCE OF:

Mrs. Pooja Jaiswal

Faculty Member

Sunbeam Institute of Information Technology, Pune

ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mr. Yogesh Kolhe (Course Coordinator, SIIT ,Pune) .

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

Hitesh Kolhe,

Aniruddha Suryawanshi,

Soham Mirajgaonkar,

Muskan Shaikh

0324 PG-DAC

SIIT Pune

A PROJECT ON

"Online Society Management System"

SUBMITTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE COURSE OF
DIPLOMA IN ADVANCED COMPUTING FROM CDAC

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY
Hinjawadi

SUBMITTED BY:

Hitesh Kolhe,
Aniruddha Suryawanshi,
Soham Mirajgaonkar,
Muskan Shaikh

UNDER THE GUIDENCE OF:
Mrs. Pooja Jaiswal
Faculty Member
Sunbeam Institute of Information Technology, PUNE.

CERTIFICATE

This is to certify that the project work under the title 'Online Society Management System' is done by Hitesh Kolhe, Aniruddha Suryawanshi, Soham Mirajgaonkar, Muskan Shaikh in partial fulfillment of the requirement for award of Diploma in Advanced Computing Course.

Mr. Yogesh Kolhe

Project Guide

Course Co-Coordinator

Date: 11-02-2025

Table of Contents:

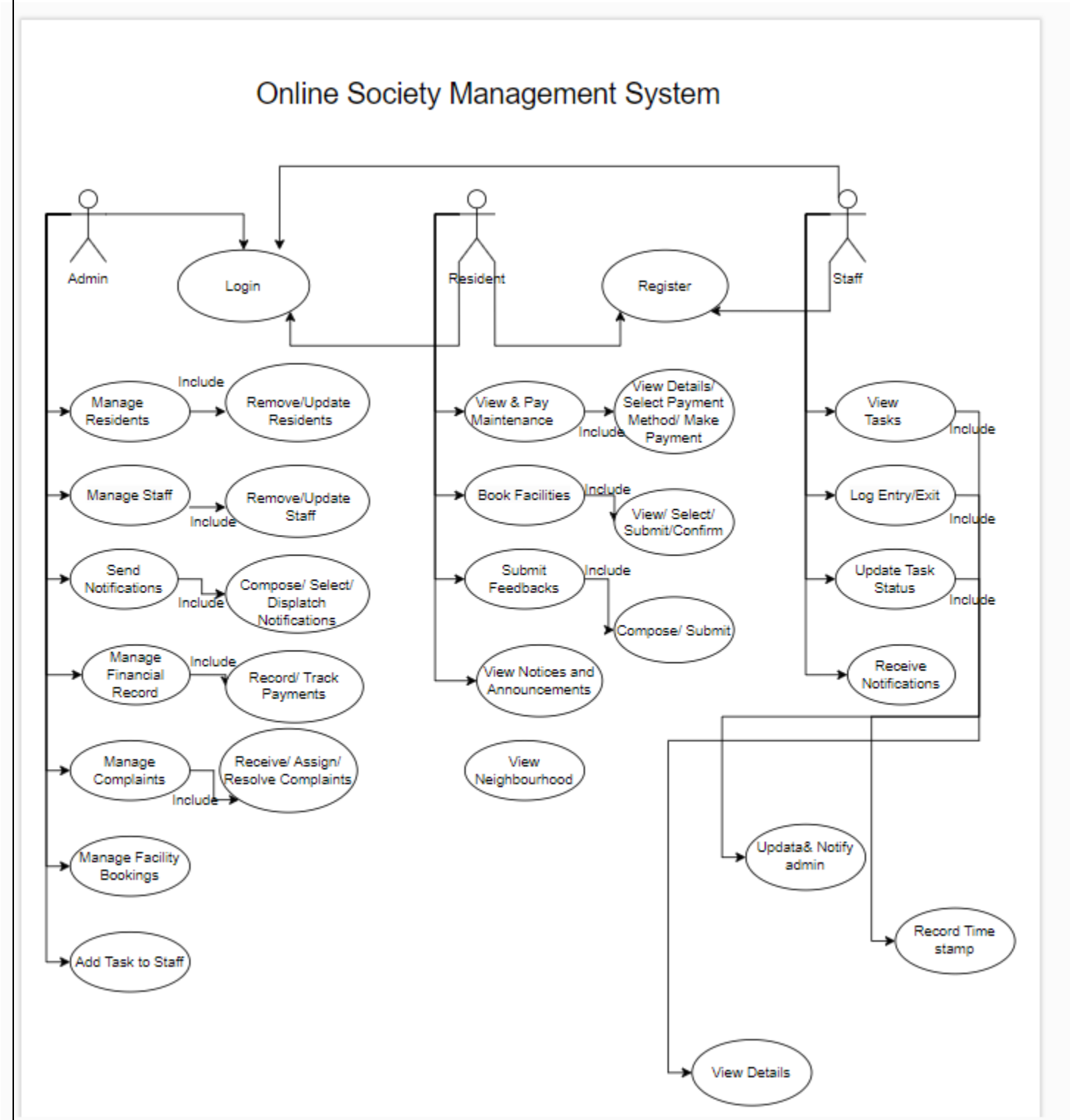
INTRODUCTION TO PROJECT

The **Online Society Management System** is a web-based application designed to streamline and automate the management of residential societies or apartment complexes This system provides an efficient platform for managing resident details security staff, Maintenance Payments,Facility Bookongs.

The primary objective of this project is to enhance transparency, improve operational efficiency, and facilitate better communication between society members and the management committee. By digitizing various processes, the system minimizes manual work and ensures a well-organized approach to society administration.

1. REQUIREMENTS

1.1 FUNCTIONAL REQUIREMENTS



2.1 Resident Flow

2.1.1 Home Page

- **Objective:** Display a list of Resident.
- **Features:**
  - View a list of available Resident with search options.

2.1.2 Resident Selection

- **Objective:** Select a Resident to view their details.
- **Features:**
  - Click on a Resident to see their details.

2.1.3 Resident Details

- **Objective:** Allow Admin to view and manage Resident details.
- **Features:**
  - View a list of details for the selected Resident.

Profile Page

- **Objective:** Manage Resident profile and view details.
- **Features:**
  - **If Not Logged In:**
    - Display a login prompt.
  - **If Logged In:**
    - View and update profile details.
    - Submit Complaints.
    - Logout.
  - Update Resident and remove Resident from the List.

2.2 Staff Flow

2.2.1 Home Page

- **Objective:** Allow staff to manage their accounts.
- **Features:**
  - Login and Register to access staff functionalities.

2.3 Staff Home Page

- **Objective:** Manage Staff details and view Tasks.
- **Features:**
  - View tasks .
  - View tasks assigned by admin and update status
  - Update status tasks.
  - Logout.

2.4 Admin Flow

2.4.1 Home Page

- **Objective:** Admin access to system functionalities.
- **Features:**
  - Login to access admin functionalities.

2.4.2 Admin Home Page

- **Objective:** Oversee and manage system operations.
- **Features:**
  - View Resident and staff details.
  - View lists of Resident,Staff.
  - View Tasks, reviews/feedback.

2. Non-Functional Requirements

2.1 Interface

- User interfaces must be intuitive and user-friendly. Detailed designs are provided in Appendix B.

2.2 Performance

- **Number of Concurrent Users:** The system should handle at least 1000 transactions/inquiries per second.
- **System Resilience:** The application should be resilient to temporary server failures.

2.3 Constraints

- The system should maintain performance standards of handling 1000 transactions/inquiries per second.

2.4 Other Requirements

2.4.1 Hardware Interfaces

Requirements: Intel Core i5 or higher (or AMD equivalent), 8 GB RAM, 512 GB SSD or larger.

-

2.4.2 Software Interfaces

- **Operating Systems:** MS Windows 13, Ubuntu 22.04.
- **Database:** MySqL.
- **Server:** Embedded Tomcat.
- **Browsers:** Compatible with modern web browsers.

3. System Design

3.1 Architecture

- **Front-End:** Developed using React.js
- **Back-End:** Built with Spring Boot for server-side logic.
- **Database:** MySqL for storing user data, orders, and other system information.
- **Server:** Embedded Tomcat for hosting the application.

4. DESIGN

4.1 Database Design

The following table structures depict the database design.

Table 1: Users

Field	Column Name	Type	length	Null
Unique	id	Long	-	0
Not null	Created_on	Date	-	0
Not null	Updated_on	Date	-	0
Unique	email	varchar	40	0
Not null	fullname	varchar	40	0
Not null	Mobile no	Varchar1	10	0
unique	password	varchar	255	0
Not null	Role(Admin,Security,Resident,Cleaner)	enum	-	0
Not null	status	boolean	-	0

Table 2: Tasks

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes; 0=No)
Primary key	id	BIGINT	-	0
Not null	created_on	DATE	-	1
Not null	Updated_on	Date	-	1
Not null	Assigned_date	DATE	-	1
Not null	description	VARCHAR	1000	0
Not null	status	VARCHAR	255	11
Mul	Assigned_to	bigint		

Table 3: Facility_bookings

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes; 0=No)
Primary	id	BIGINT	-	0
null	created_on	DATE	-	1
null	updated_on	DATE	6	1
Not null	facility_name	VARCHAR	255	1
Not null	from_date_time	DATE	255	1
unique	status	VARCHAR	255	1
Not null	to_date_time	DATE	-	1
Not null	resident_id	BIGINT	-	0

Table 4: Flats

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes; 0=No)
primary	id	BIGINT	-	0
Not null	Created_on	VARCHAR	255	1
Not null	updated_on	VARCHAR	255	1
Not null	Flat_number	VARCHAR	10	1
Not null	Resident_id	VARCHAR	20	1

Table 5: Notifications

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes; 0=No)
Primary key	id	BIGINT	-	0
Not null	created_on	DATE	-	1
null	updated_on	DATE	6	1
null	message	VARCHAR	1000	1
Not null	sent_date_time	ENUM	-	1
null	sent_to_all_residents	BIT	-	0

Table 6: Payments

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes; 0=No)
primary	id	BIGINT	-	0
null	created_on	DATE	-	1
null	updated_on	DATETIME	6	1
0	paymentdate	DATE	-	1
0	status	VARCHAR	255	1
0	totalamount	double	-	1
0	resident_id	bigint	30	1

E-R Diagram,Dataflow diagram and Class Diagram:

Go to Appendix A

5. CODING STANDARDS IMPLEMENTED

Naming and Capitalization

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances

Identifier	Case	Examples	Additional Notes
Class	Pascal	Users, UserController	Class names should be based on "objects" or "real things" and should generally be **nouns**. No '_' signs allowed. Do not use type prefixes like 'C' for class.

Method	Camel	Login, Register,addTasks	Methods should use **verbs** or verb phrases.
Parameter	Camel	FullName, mobileno,email, password	Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in most scenarios.
Interface	Pascal with "I" prefix	UserRepository, StaffRepository ResidentRepository	Do not use the '_' sign
Annotation	Pascal	SpringBootApplication	Use @ at start of annotation
DTOs	Camel	FacilityBookingDTO, DisplayNotificationDto, ComplaintDto	Use to transfer data between the processes
Exception Class	Pascal with "Exception" suffix	ResourceNotFoundException	

Comments

- Comment each type, each non-public type member, and region

declaration.

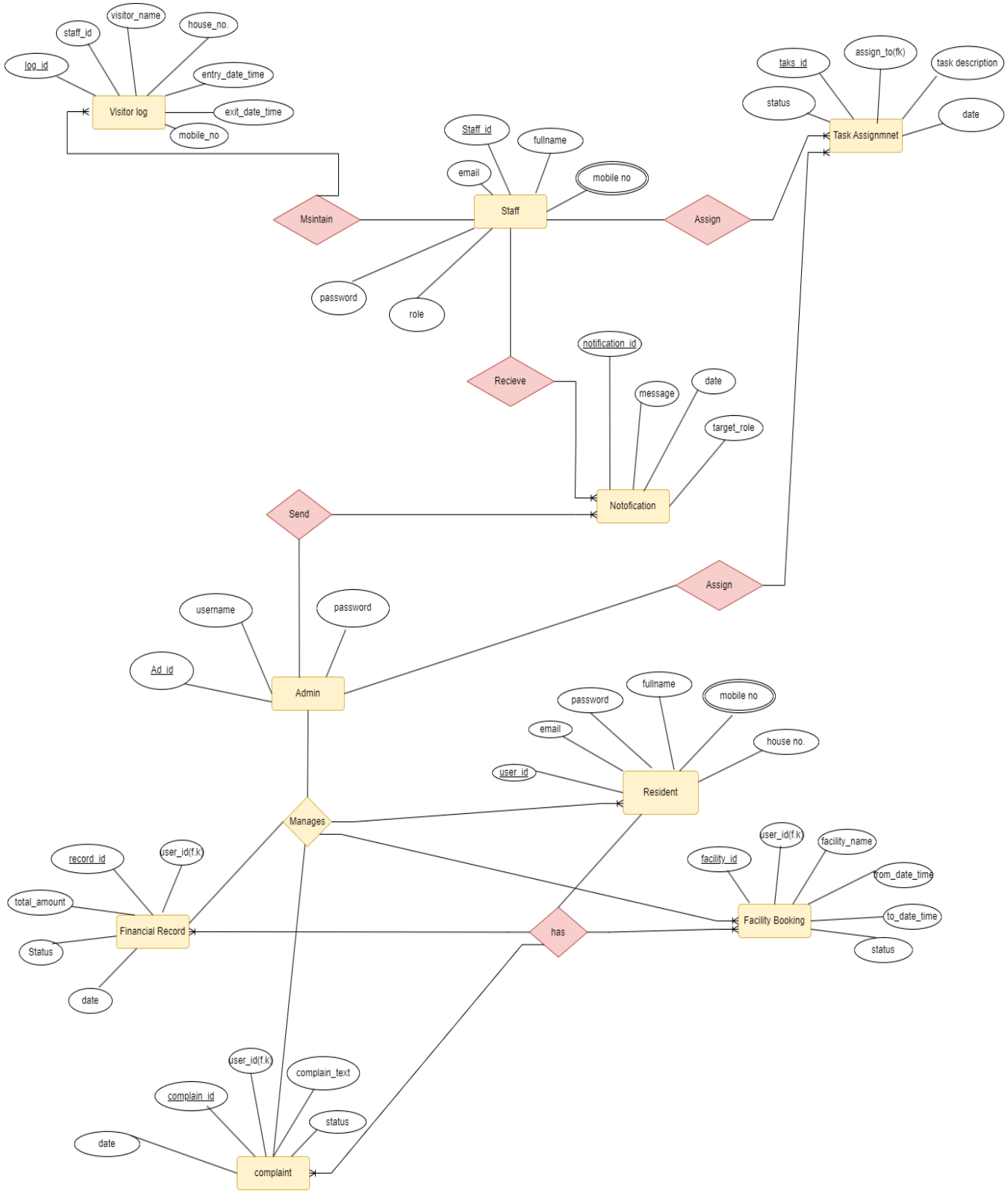
- Use end-line comments only on variable declaration lines.
- End-line comments are
- comments that follow code on a single line.
- Separate comments from comment delimiters (apostrophe) or // with one space.
- Begin the comment text with an uppercase letter.
- End the comment with a period.
- Explain the code; do not repeat it.

6. PROJECT MANAGEMENT RELATED STATISTICS

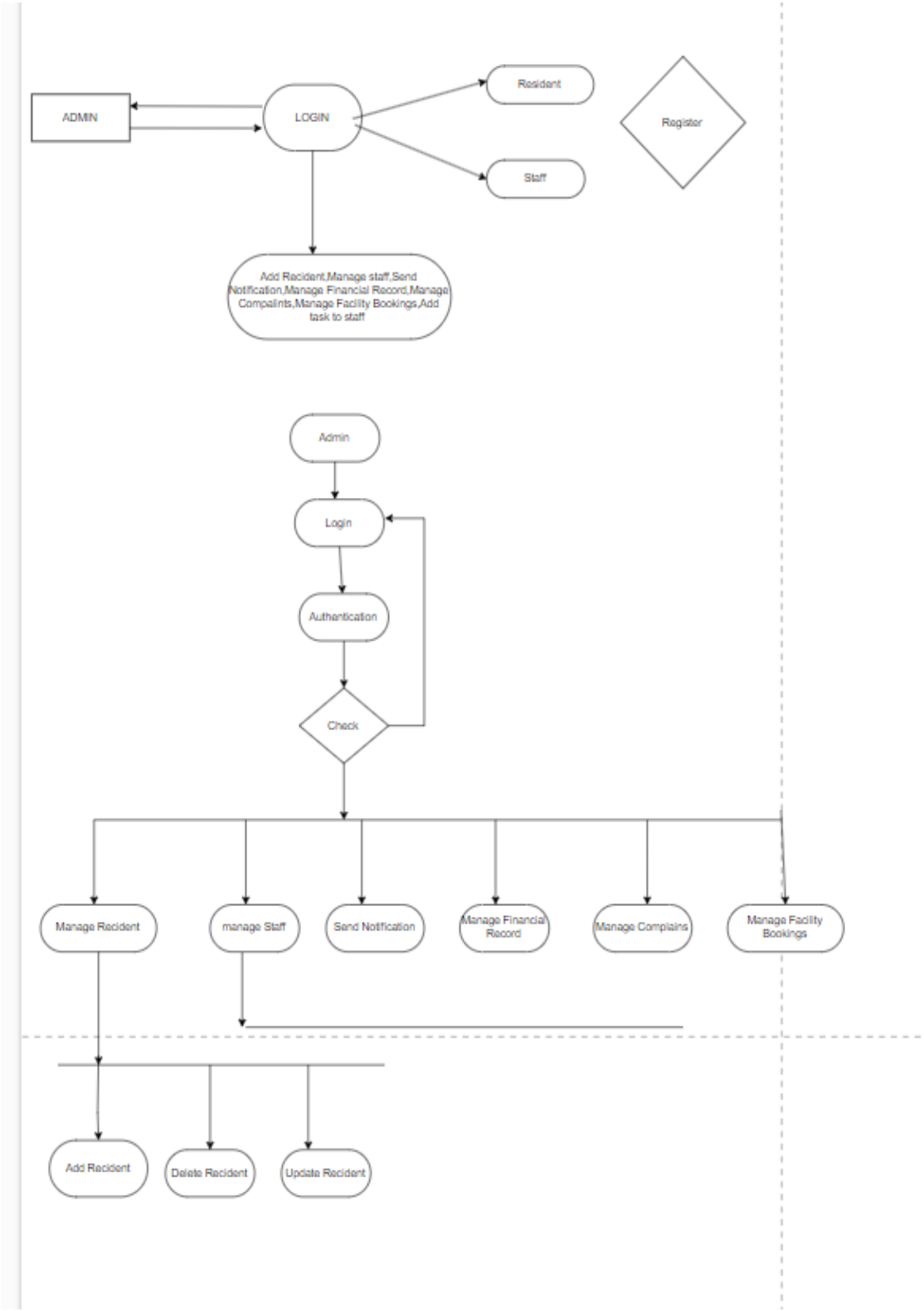
DATE	WORK PERFORMED	SLC PHASE	Additional Notes
Oct 11, 2024	Project Allotment and User Requirements Gathering	Feasibility Study	Our team met the client Mr. Nitin Kudale (CEO, SIIT Pune) to know his requirements.
Oct 17, 2024	Initial SRS Document Validation and Team Structure Decided	Requirement Analysis (Elicitation)	The initial SRS was presented to the client to understand his requirements better.
Oct 30,	Designing the use-	Requirement	Database Design completed.

DATE	WORK PERFORMED	SLC PHASE	Additional Notes
Nov 2024	cases, Class Diagram, Collaboration Diagram, E-R Diagram, and User Interfaces	Analysis & Design Phase	
Nov 25, 2024	Business Logic Component Design Started	Design Phase	----------
Dec 16, 2024	Coding Phase Started	Coding Phase	70% of Class Library implemented.
Dec 17, 2024	Implementation of Web Application and Window Application Started	Coding Phase	Class Library Development going on.
Dec 18, 2024	Off	Off	Off
Dec 19, 2024	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	Class Library Modified as per the need.
Jan 10, 2025	Implementation of Web Application and Window Application Continued	Coding Phase and Unit Testing	----------
Jan 21, 2025	After Ensuring Proper Functioning the Required Validations were Implemented	Coding Phase and Unit Testing	Module Integration was done by the Project Manager
Jan 30, 2024	The Project was Tested by the respective Team Leaders and the Projec Manager	Testing Phase (Module t Testing)	----------
August 13, 2025	The Project was Submitted to Other Project Leader of Oth Project Group For Testing	Testing Phase er (Acceptance Testing)	The Project of Other Team was Taken up by the Team for Testing
Feb 14, 2025	The Errors Found were Removed	Debugging	The Project was complete for submission
Feb 11, 2025	Final Submission of Project	----------	----------

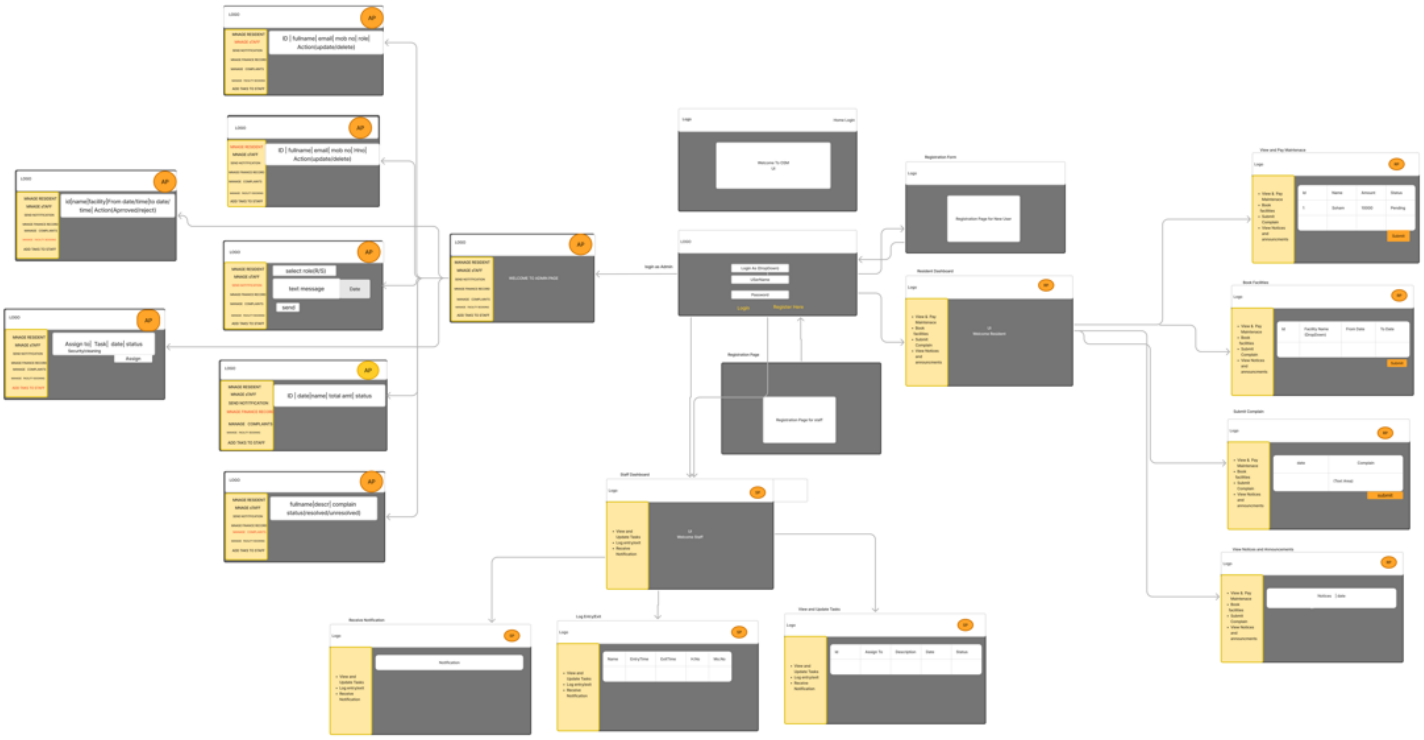
Appendix A
Entity Relationship Diagram



Data Flow Diagram



Class Diagram

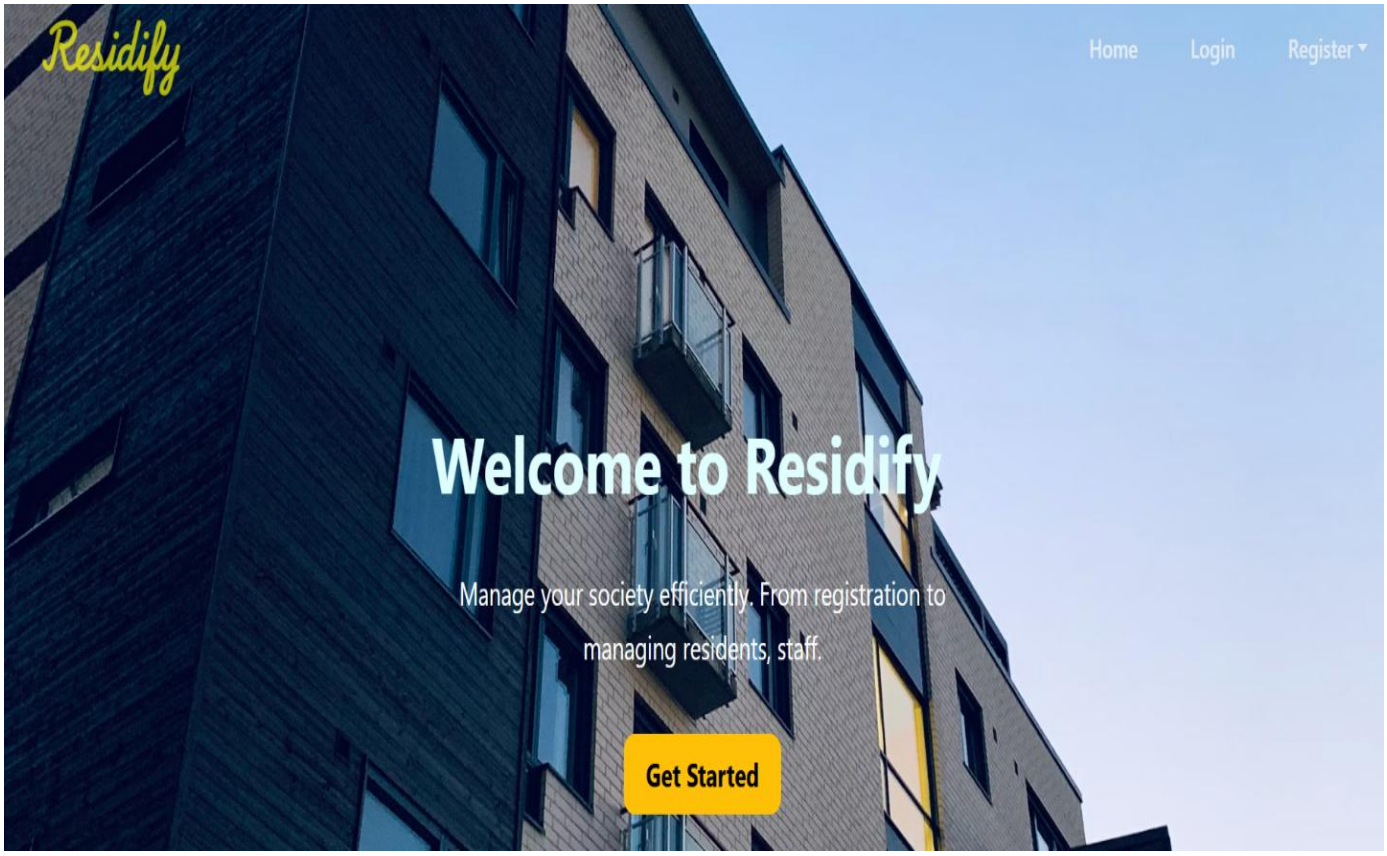


Appendix B

Home Dashboard:

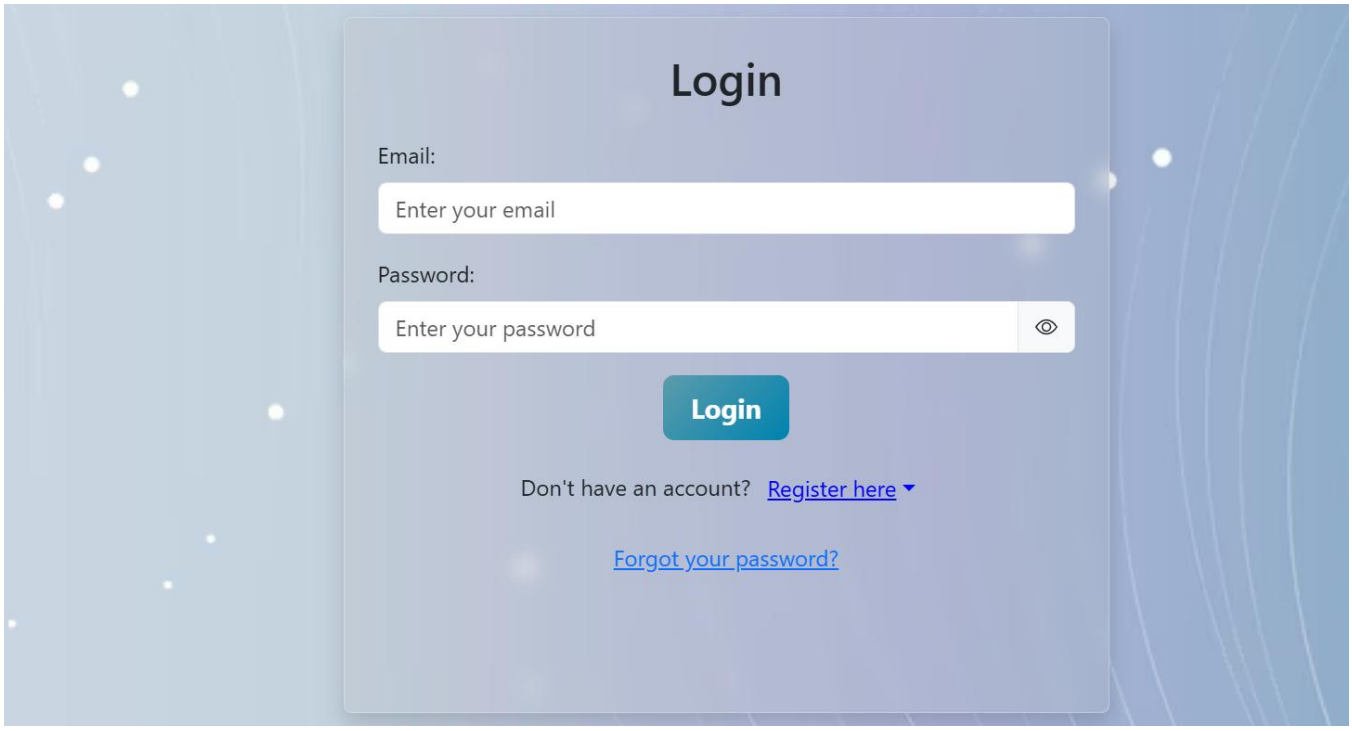
Url:

http://localhost:3000

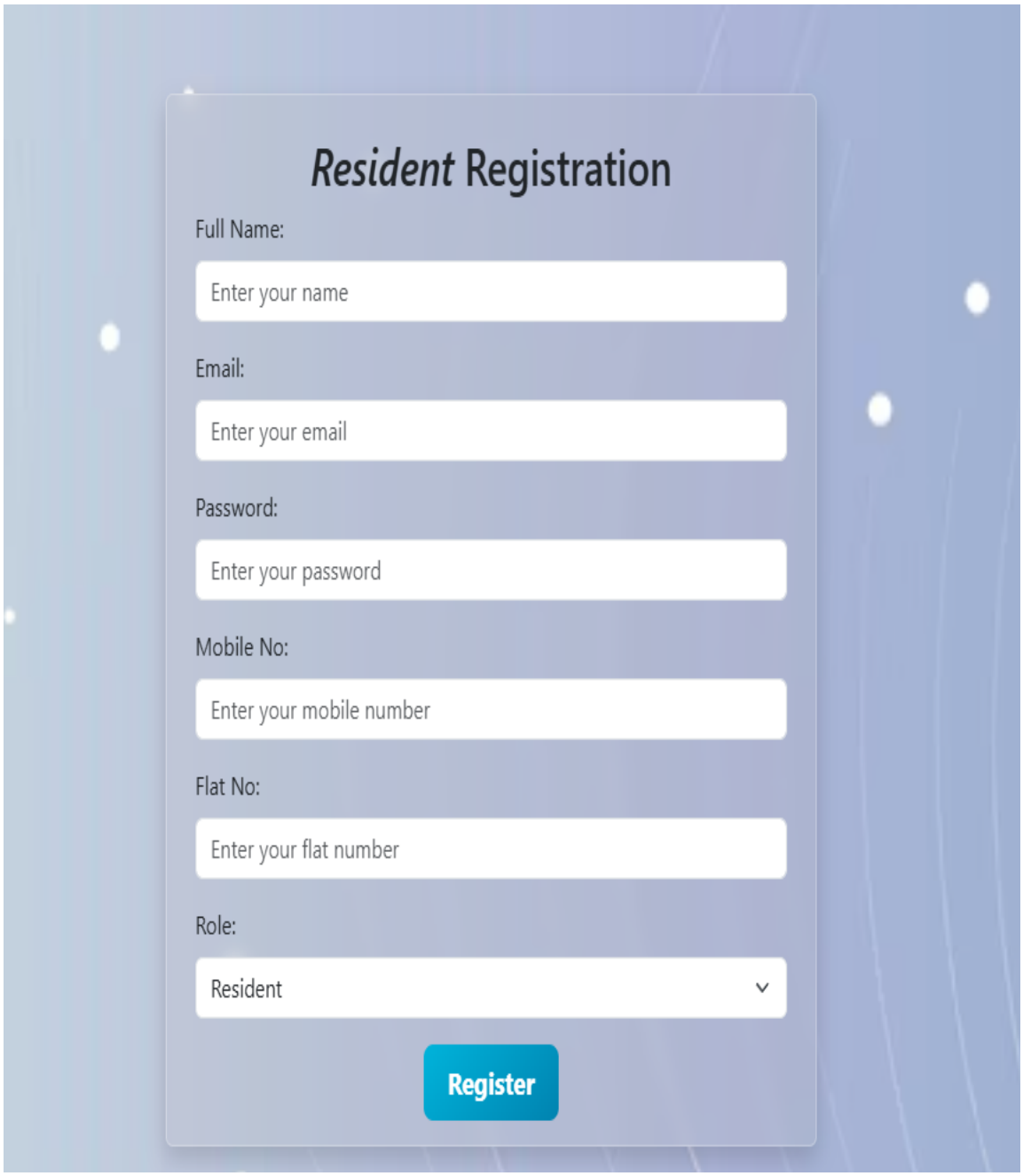


Login Page:

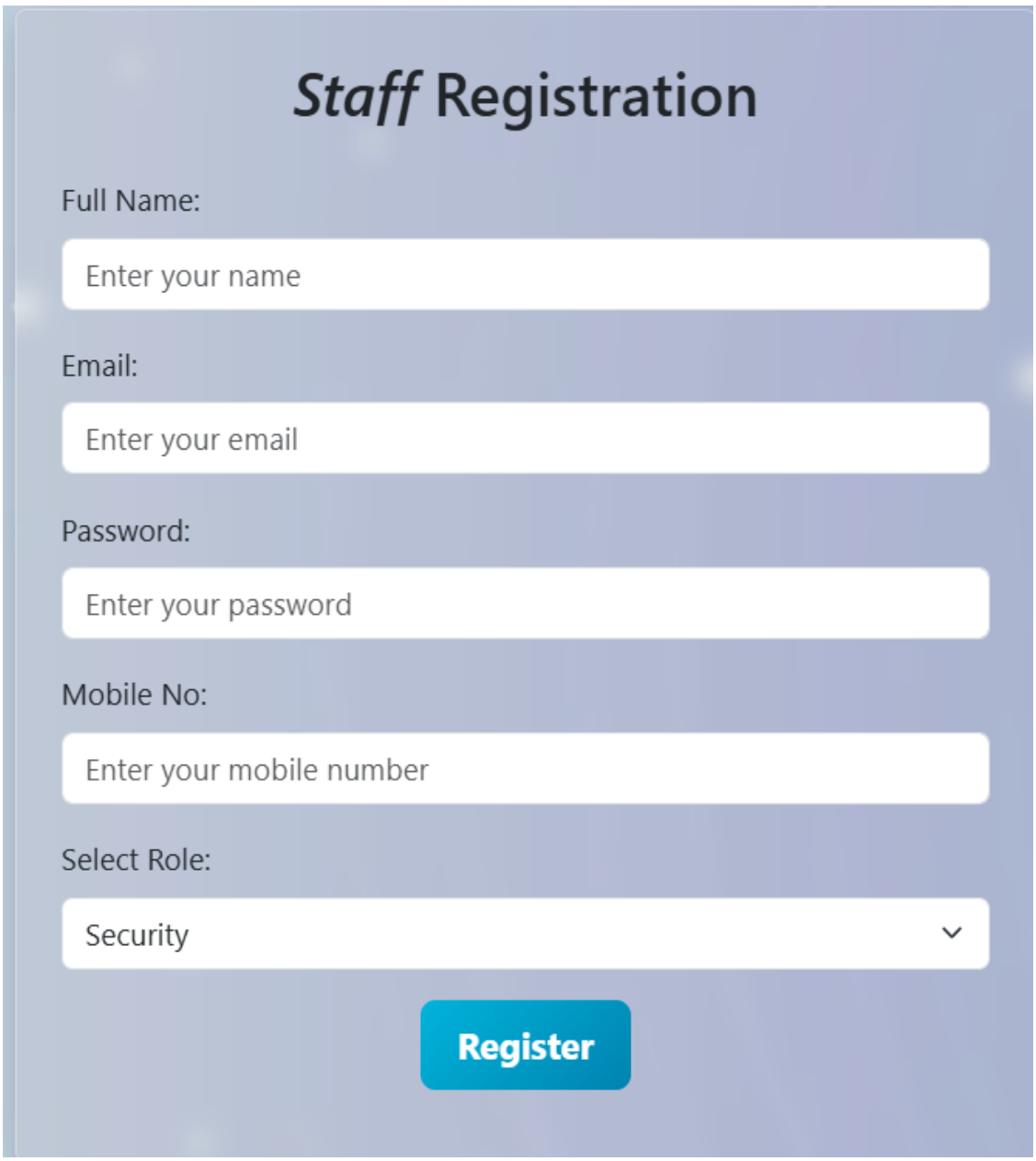
Url: http://localhost:3000/login



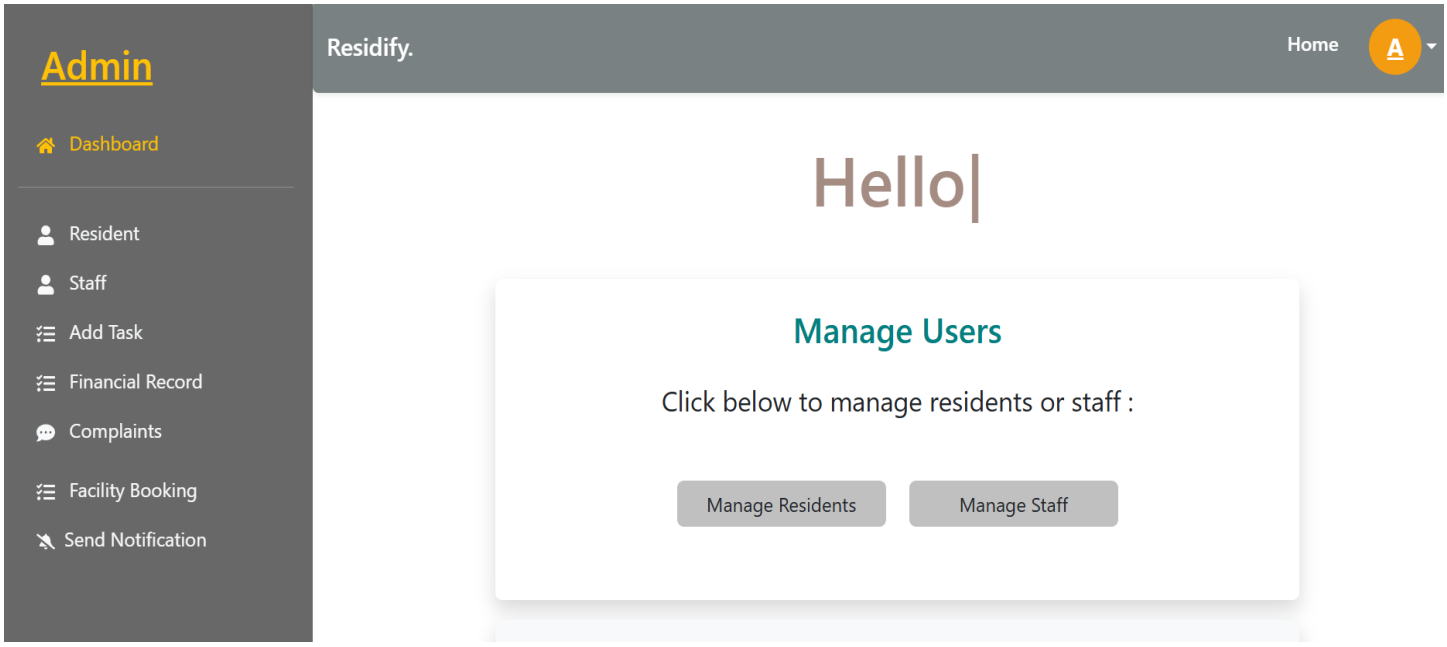
Register Page
url: http://localhost:3000/register/resident



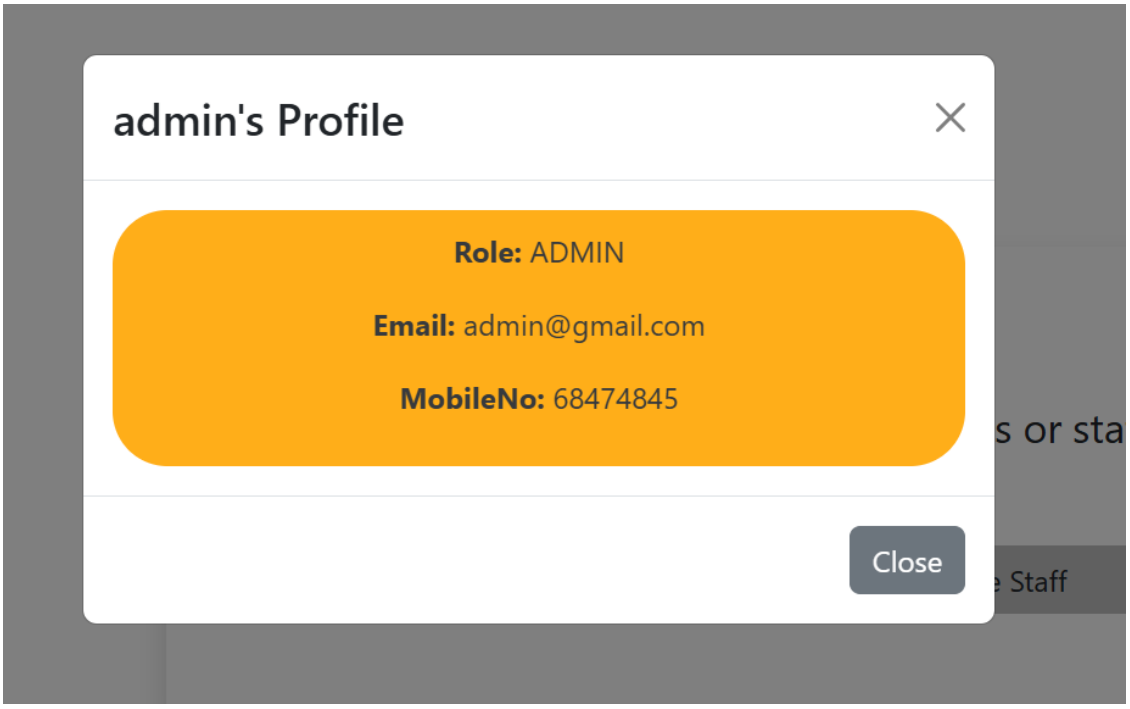
Register Page
url: http://localhost:3000/register/staff



Admin Dashboard
url: http://localhost:3000/admin

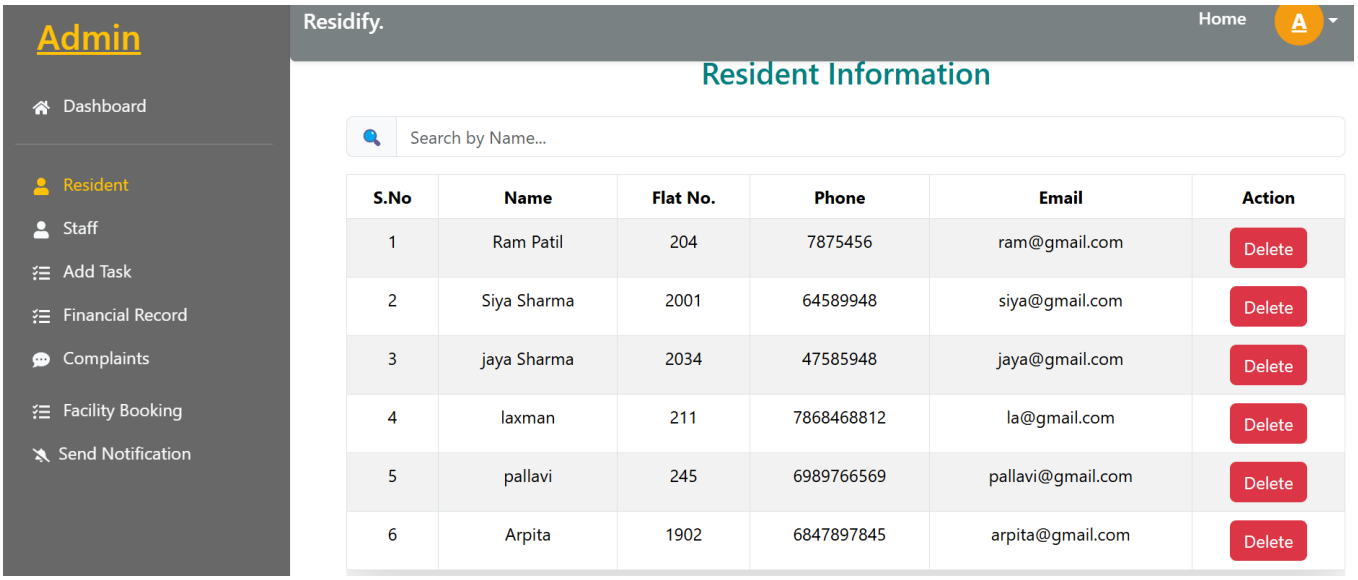


Admin Profile Page
url:http://localhost:8080/admin



Resident List

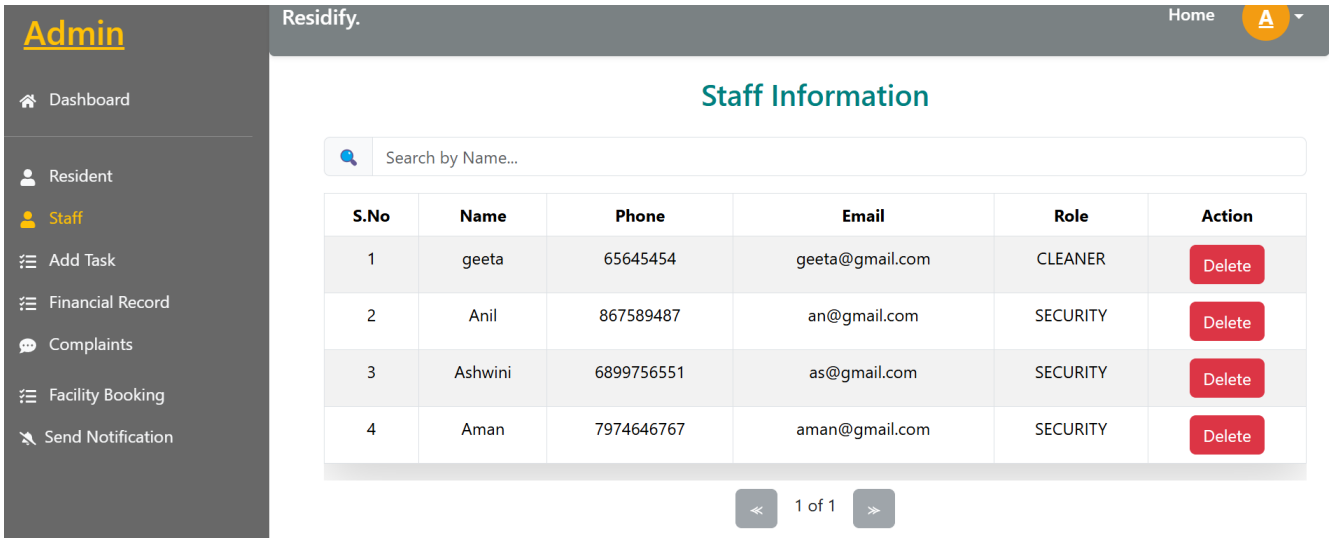
url: http://localhost:3000/admin/residents



S.No	Name	Flat No.	Phone	Email	Action
1	Ram Patil	204	7875456	ram@gmail.com	Delete
2	Siya Sharma	2001	64589948	siya@gmail.com	Delete
3	jaya Sharma	2034	47585948	jaya@gmail.com	Delete
4	laxman	211	7868468812	la@gmail.com	Delete
5	pallavi	245	6989766569	pallavi@gmail.com	Delete
6	Arpita	1902	6847897845	arpita@gmail.com	Delete

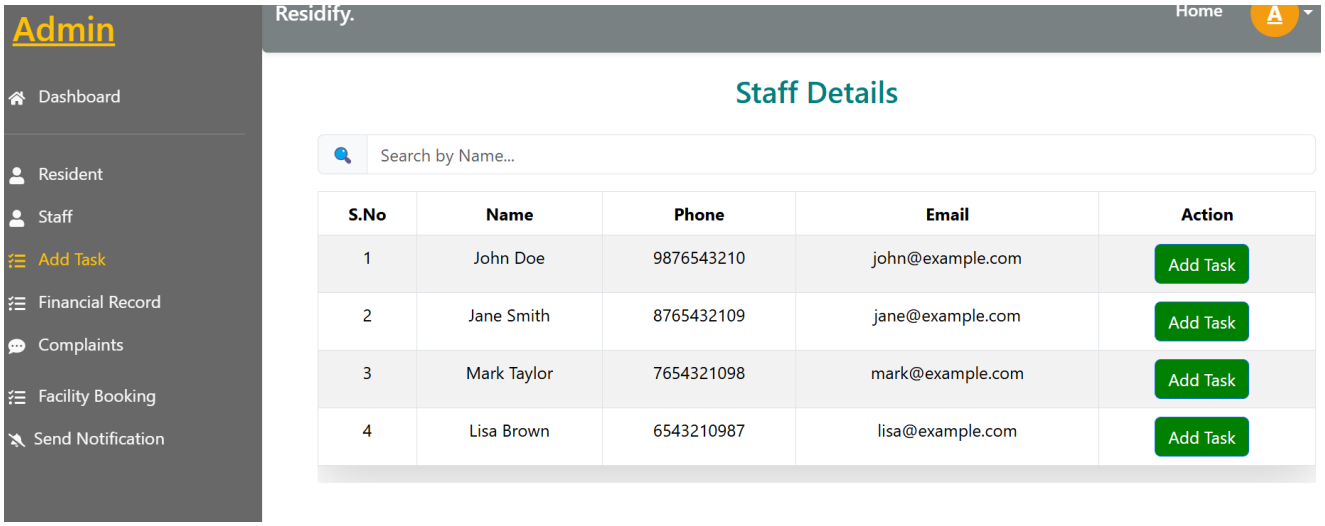
Staff Details

url: http://localhost:3000/admin/staffs

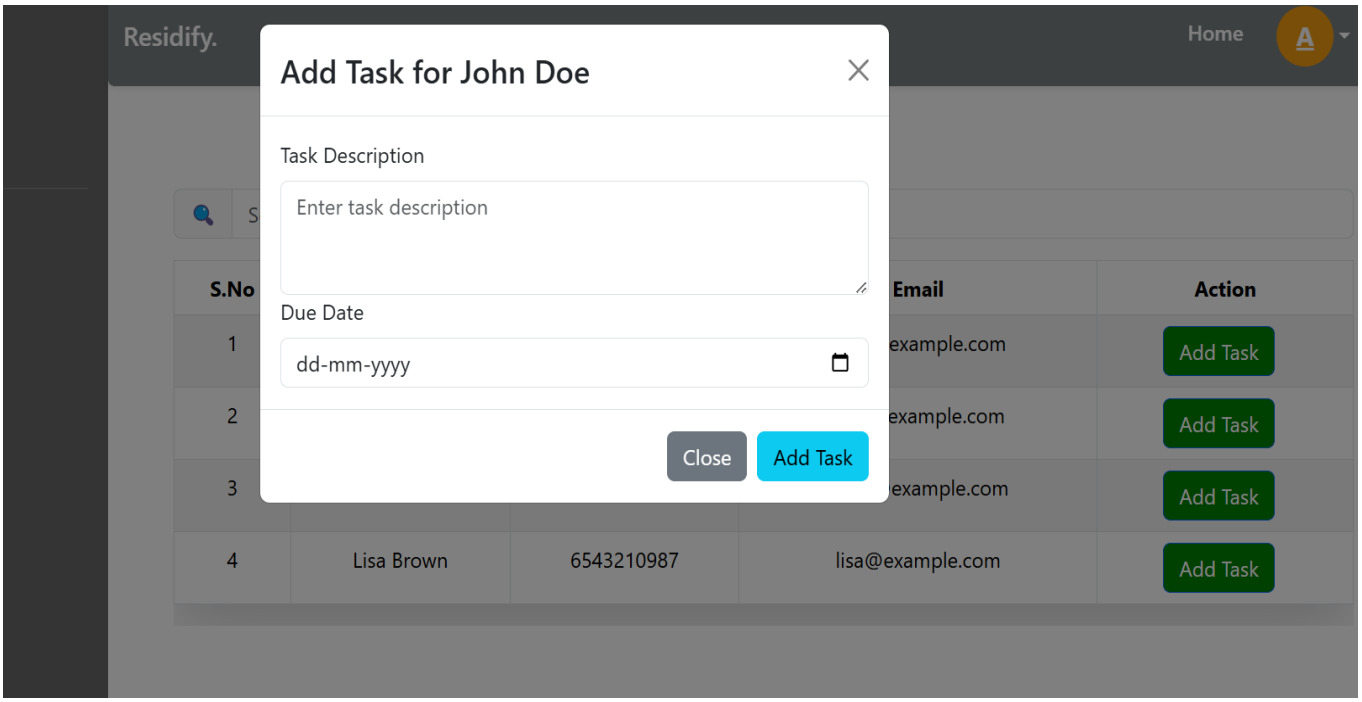


S.No	Name	Phone	Email	Role	Action
1	geeta	65645454	geeta@gmail.com	CLEANER	Delete
2	Anil	867589487	an@gmail.com	SECURITY	Delete
3	Ashwini	6899756551	as@gmail.com	SECURITY	Delete
4	Aman	7974646767	aman@gmail.com	SECURITY	Delete

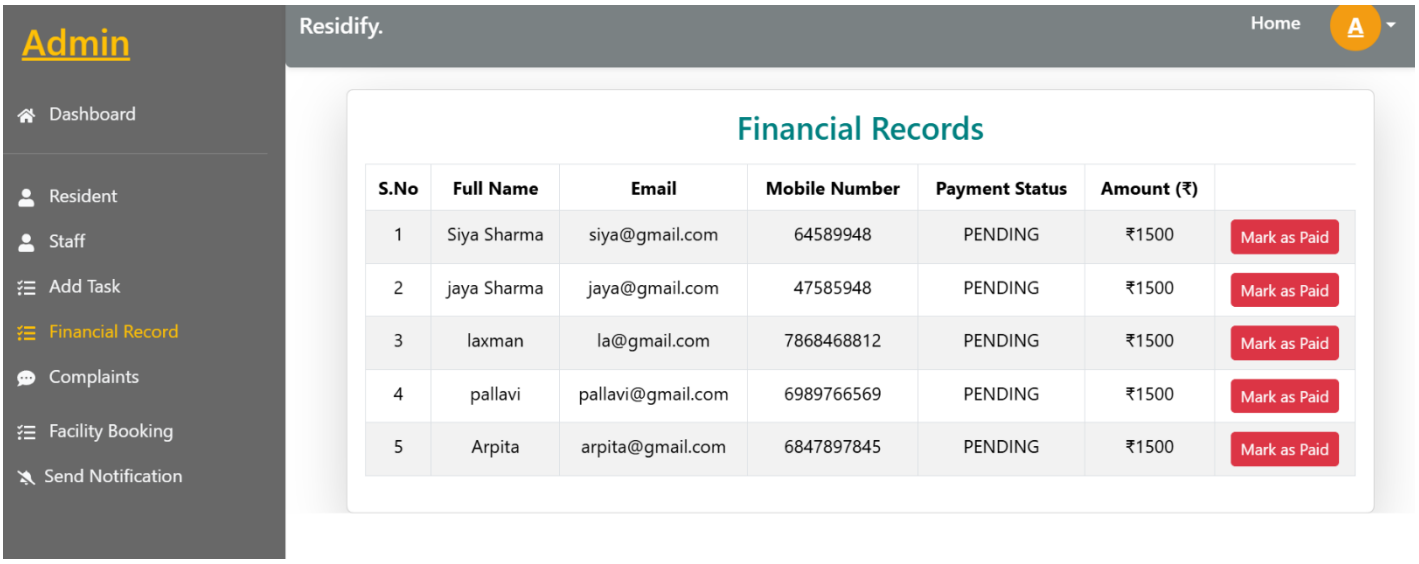
Staff Dashboard
url: http://localhost:3000/staff



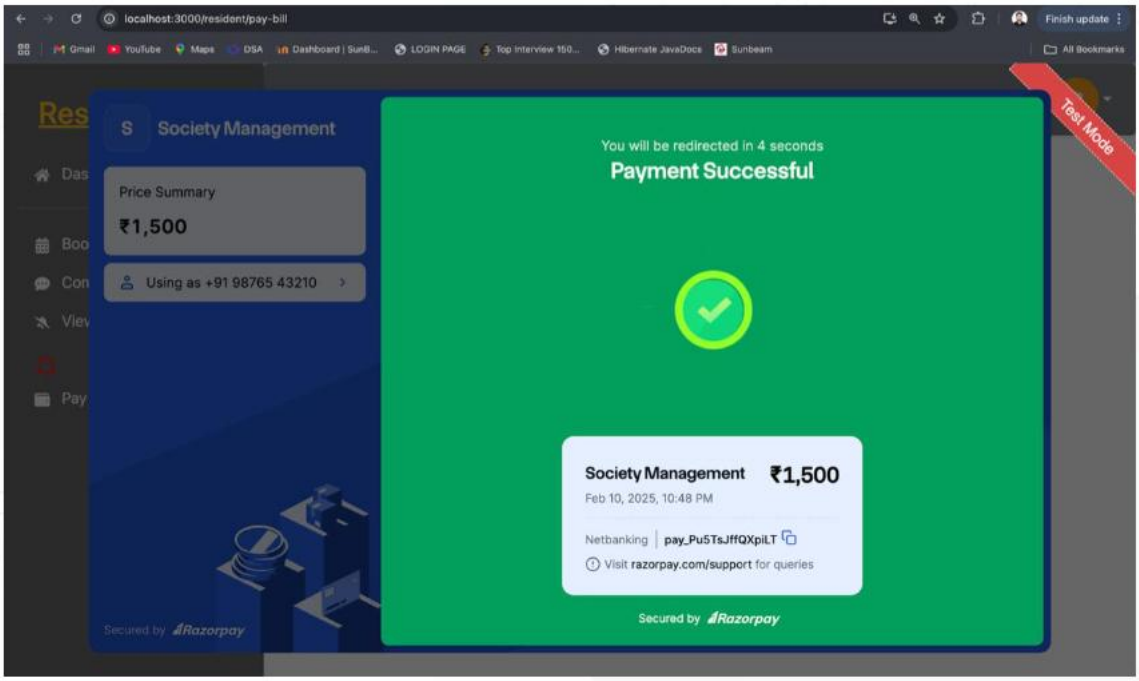
Add Task To Staff
url: http://localhost:3000/admin/add-task



Financial Record
url: http://localhost:3000/admin/financial-record



Payment
url: http://localhost:3000/resident/pay-bill



7. REFERENCES

1. **Spring Boot Documentation**
   URL: https://spring.io/projects/spring-boot

2. **React.js Documentation**
   URL: https://reactjs.org/docs/getting-started.html

3. **Java Programming Language**
   URL: https://www.oracle.com/java/

4. **MySQL Workbench Documentation**
   URL: https://dev.mysql.com/doc/workbench/en/

5. **Spring Boot with React**
   URL: https://www.bezkoder.com/react-spring-boot-crud/

6. **Java Persistence API (JPA) Documentation**
   URL: https://www.eclipse.org/eclipselink/documentation/2.7/

7. **Swagger Documentation for Spring Boot**
   URL: https://springdoc.org/

8. **MDN Web Docs**
   URL: https://developer.mozilla.org/