



**INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE  
DEVELOPMENT AKURDI, PUNE**

Documentation On

**“PAYMENT REMINDER APP”  
REMINDER FOR DUE PAYMENTS**

PG-DAC SEPT 2022

**Submitted By:**

**Group No: 29**

**Roll No.**

**229024**

**229046**

**Name:**

**Deeksha Sahu**

**Nikhil Solankhi**

**Mrs. Megha Mane**  
**Project Guide**

**Mr. Rohit Puranik**  
**Centre Coordinator**

## ABSTRACT

This project is a web-based payment reminder system for the property owners. The project's objective is to provide the automated payment remainder emails to the tenant and owner of the properties on the due date with the help of a web platform.

A payment reminder app for rental systems is a specialized software application designed to assist landlords, property managers, and tenants to keep track of rental payment schedules. This app allows users to input rental payment due dates, set reminders, and receive notifications to ensure timely payments. Users can view their payment history and rental schedule as list format, making it easy to track and manage their rental payments. The app may also include features such as maintenance tracking, and tenant communication. By using a payment reminder app for rental systems, landlords and property managers can easily monitor payment status and reduce the risk of missed payments, while tenants can avoid late fees and maintain a good rental payment record. Overall, this app helps to streamline the rental payment process and improve the overall efficiency of managing rental properties.

## ACKNOWLEDGEMENT

This project was a great learning experience for us and we are submitting this work to Institute for Advance Computing and Software Development (C-DAC IACSD, Akurdi). I extend my sincere and heartfelt thanks to our esteemed guide, Mrs. Megha Mane for providing me with the right guidance and advice at the crucial juncture sand for showing me the right way. I extend my sincere thanks to our respected Centre Co-Ordinator Mr. Rohit Puranik, who gave us all the required support and kind coordination to provide all the necessities to complete the project and allowing us to use the facilities available throughout the course. I would like to express our sincere gratitude towards Mrs. Madhura Anturkar and Mrs. Kishori Khadilkar, our faculty for J2EE and React, who was always there for us. Their guidance and support helped us overcome various obstacles. Without their tremendous support, guidance, and efforts, this project would not have been possible. I would like to thank the other faculty members also, at this occasion. Last but not the least, I would like to thank my friends and family for the support and encouragement they have given me during the course of our work.

## Table of Content

INTRODUCTION .....	1
Features.....	2
1.1 PROJECT OBJECTIVE.....	2
1.2 PROJECT OVERVIEW .....	3
1.3 PROJECT SCOPE .....	3
1.4 STUDY OF THE SYSTEM.....	4
1.4.1 MODULES:.....	4
Figure 1: Admin Activity Diagram .....	5
SYSTEM ANALYSIS.....	10
2.1 EXISTING SYSTEM .....	10
2.2 PROPOSED SYSTEM :.....	11
2.3 SYSTEM REQUIREMENT SPECIFICATION:.....	11
2.3.1 GENERAL DESCRIPTION .....	11
2.3.2 Objective (Purpose):.....	11
2.3.3 SYSTEM REQUIREMENTS .....	12
SYSTEM DESIGN .....	14
3.1 INPUT AND OUTPUT DESIGN.....	14
3.1.1 INPUT DESIGN:.....	14
3.1.2 OUTPUT DESIGN: .....	15
DATABASE DESIGN.....	15
3.2 DATABASE.....	15
3.3 SYSTEM TOOLS.....	15
3.3.1 FRONT END: () .....	15
3.3.2 BACKEND: .....	16
Figure 3: Level 0 DFD for Admin .....	17
Figure 4: Level 1 DFD for Admin .....	17
Figure 5: Level 2 DFD for Admin .....	18
Figure 6: Level 0 DFD for Owner.....	19
Figure 7: Level 1 DFD for Admin .....	19
Figure 8: Level 2 DFD for Owner.....	20
Figure 9: E-R Diagram .....	21

Figure 10: E-R Diagram Auto Generated .....	22
Use Case Diagrams .....	23
Figure 11: Admin Use Case Diagram .....	23
Figure 12: Owner Use Case Diagram .....	24
CLASS DIAGRAM .....	25
Figure 13: Class Diagram .....	25
TABLE STRUCTURE:.....	26
Figure 14: Tables.....	26
Figure 15: Owner Table .....	26
Figure 16: Property Table .....	27
Figure 17: Client Table .....	27
Figure 18: Payment Table.....	28
Figure 19: Payment Reminder Table .....	28
Figure 20: Owner Client Relation Table.....	28
PROJECT DIAGRAMS.....	29
CONCLUSION .....	36
Future Scope.....	37
REFERENCES.....	38

## INTRODUCTION

A payment reminder app for a rental application is a software tool designed to help landlords and property managers keep track of rent payments from their tenants. The app can be installed on a smartphone or computer and allows the user to set up reminders for rent payments, send notifications to tenants, and track payment history.

With a payment reminder app, landlords can easily monitor rent payments and avoid the hassle of manually tracking payments. The app can also help tenants avoid late fees and keep track of their payment history. Some features of a payment reminder app for rental applications may include:

1. Customizable reminders: The app allows the user to set up reminders for rent payments that are tailored to their specific needs.
2. Payment tracking: The app tracks rent payments and provides a history of payments, making it easy to keep track of who has paid and who hasn't.
3. Notification system: The app sends notifications to tenants when rent is due or when a payment is overdue.
4. Payment options: The app may allow tenants to make payments directly through the app, making the payment process more convenient.

In summary, a payment reminder app for rental applications is a useful tool for landlords and property managers to stay on top of rent payments and ensure timely payments from their tenants.

Features:-

PRA (Property Rental Application) is a software tool designed for property managers and landlords to manage their rental properties. Here are some of the features that a PRA rental application may offer:

- Property management: PRA rental application allows landlords to manage their properties from a single dashboard. They can add, edit or delete property details, upload pictures, and manage rental agreements.
- Tenant management: The app allows landlords to manage their tenants, including adding new tenants, viewing tenant details, managing rental payments, and tracking rent payment history.
- Online rental applications: The app can provide an online application process for tenants to apply for rental properties, including background checks and screening.
- Maintenance tracking: The app can help landlords manage maintenance requests and track maintenance history for each property.
- Communication: The app provides a communication tool for landlords to communicate with tenants through email.

Overall, a PRA rental application can help landlords and property managers efficiently manage their rental properties, streamline rental applications and payments, and improve communication with tenants.

### 1.1 PROJECT OBJECTIVE

Payment reminder app is useful for those who often forget to pay their rent and for those who find it hard to collect the payment from their clients/tenants. This app reminds them to pay as in case of the tenant and to collect in case of the owner of the properties by sending notifications on email as well as mobile phone. This application makes it very easy to manage the finances and dues.

### 1.2 PROJECT OVERVIEW

The central concept of the application is to allow the user to maintain payment system virtually using the internet.

and allow Owners add their properties and respective clients on the property and manage payments. The information pertaining to the owners, clients, properties and payments are stores on an RDBMS at the server side (store).

Payment reminder application is a type of time management computer application that is designed to alert the user and their client of important events in our case the due payment that have been input in our program. This programs provide a list view of events (view ,add ,update), as well as a reminding technique. In this application the reminders are provided through email to the owner of the properties and their clients.

### 1.3 PROJECT SCOPE

This system allows the users to track the record for the payments from their respective property clients.

This System allows Admin for the analysis of vacant properties.

Intended Audience:

The app is intended for individuals who have a hard time keeping track of when their payments are due.

### 1.4 STUDY OF THE SYSTEM

1.4.1 MODULES: The system after careful analysis has been identified to be presented with the following modules and roles. The modules involved are:

- Administrator
- Owner
- Property
- Client
- Payment
- Confirmation Token

#### **1.4.1.1 Administrator**

The administrator is the super user of this application. Only admin have access into the admin page. The administrator has all the information about the users and about all products.

This module is divided into different sub modules.

1. Manage Owners
2. Manage Properties
3. Manage Clients
4. Manage Payments

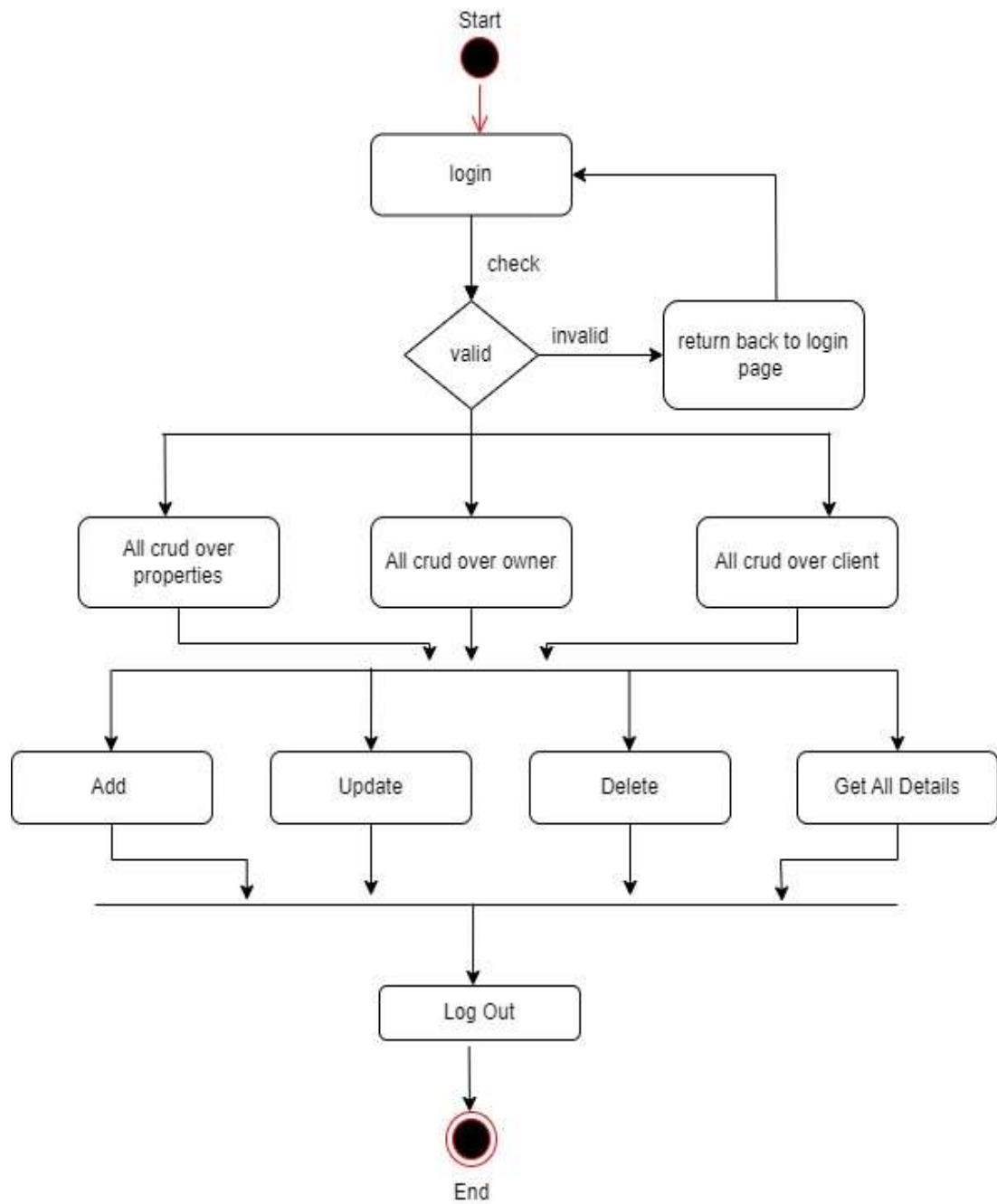


Figure 1: Admin Activity Diagram

**> Add Owner**

The application contains different Owners of properties having different clients.. Admin can add new owner into the existing system with all its details.

**> Delete Owner**

Administrator can delete the Owner based on the number of clients of that particular property.

**> Search Owner**

Admin will have a list view of all the existing owners. He can also search for a particular owner by owner id.

**> Update Owner**

Admin can also add owners.

**> Remove Property**

Admin have the privilege to remove the property.

**> Update Property**

Admin can update Property of the owner. Admin can see the details of Property and enhance the details.

**> Search Property**

Admin will have a list view of all the existing properties. He can also search for a particular property by property id.

**> Search Client**

Only admin is having the privilege to add a Owner. He can search the client to manage the property details.

### ➤ **Search Payment**

Admin will have a list view of all the existing payments. He can also search for a particular payment by payment id.

#### **1.4.1.2 Owner**

The Owner is the user of this application. Owner have access into the owner page. The owner has all the information about the properties, client and payments. This module is divided into different sub modules.

1. Manage Properties
2. Manage Clients
3. Manage Payments

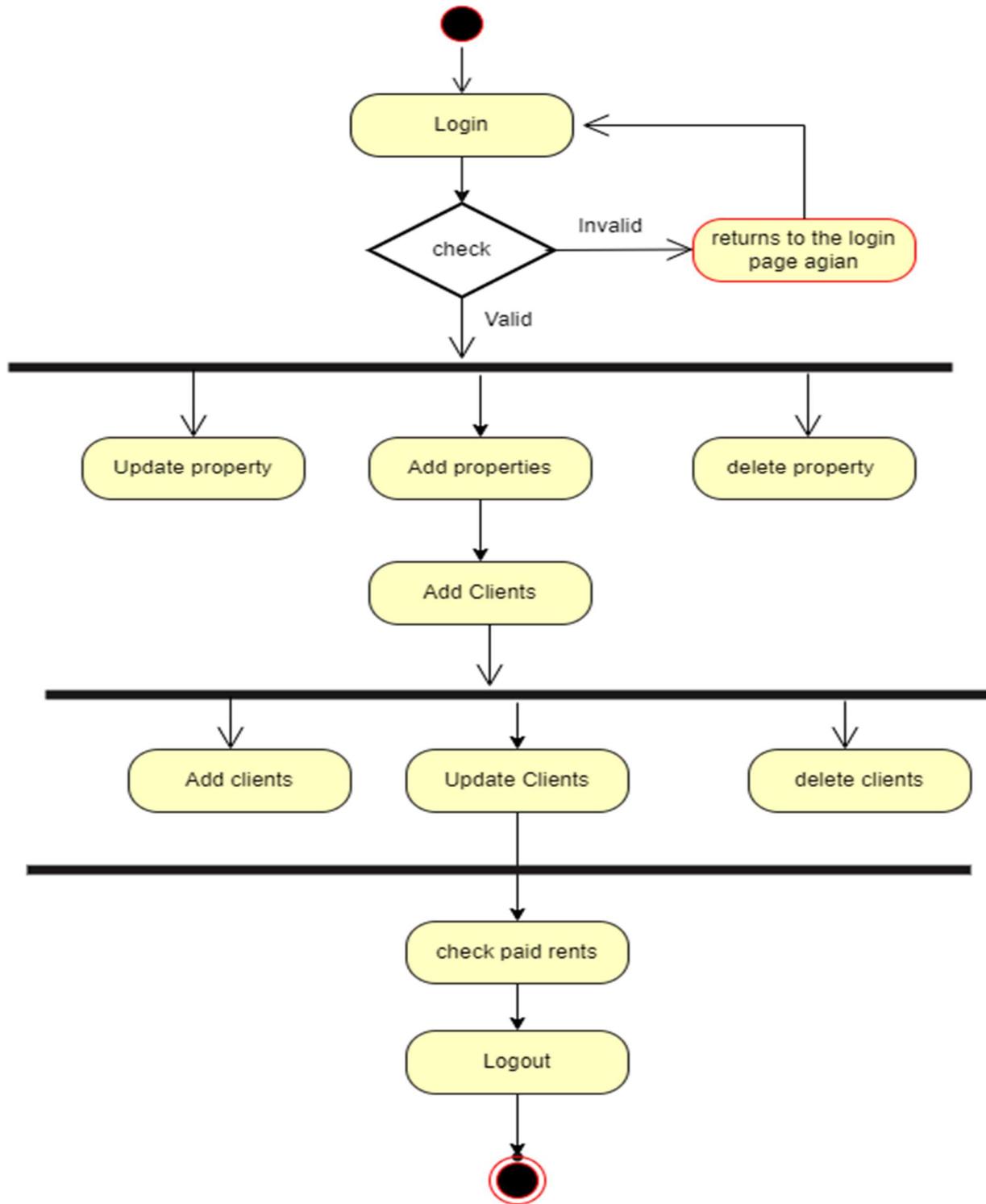


Figure 2: Owner Activity Diagram

**> Add Property**

Owner can add Property. Owner can see the details of Property and the payment status.

**> Remove Property**

Owner has the privilege to remove the property.

**> Update Property**

Owner can update Property. Owner can see the details of Property and enhance the details.

**> Search Property**

Owner will have a list view of all the existing properties. He can also search for a particular property by property id.

**> Add Client**

Owner can add Client. Owner can see the details of Client and the payment status.

**> Remove Client**

Owner has the privilege to remove the Client.

**> Update Client**

Owner can update Client. Owner can see the details of Client and enhance the details.

**> Search Client**

Owner is having the privilege to add Client. He can search the client to manage the property details.

**> View Payment**

Owner has the privilege to view the payment details.

**> Update Payment**

Owner can update payment status.

## SYSTEM ANALYSIS

System analysis is the process of gathering and interpreting facts, diagnosing problems, and using the information to recommend improvements on the system. System analysis is a problem-solving activity that requires intensive communication between the system users and system developers.

System analysis or study is an important phase of any system development process. The system is viewed as a whole, the inputs are identified, and the system is subjected to close study to identify the problem areas. The solutions are given as a proposal. The proposal is reviewed on user request and suitable changes are made. This loop ends as soon as the user is satisfied with the proposal.

### 2.1 EXISTING SYSTEM

The current system for rental arrangements is to visit the property manually and from the available clients collect the rents and manage the records manually which is:

- ✓ less user-friendly.
- ✓ User must go to properties and collect payments.
- ✓ It is difficult to manage the communication with the client.
- ✓ Details of the client is difficult to manage.
- ✓ It is a time-consuming process.
- ✓ Not in reach of distant property clients

## 2.2 PROPOSED SYSTEM :

Provides the automated solution for the above problems. Owner can maintain the properties be it a distant one or the nearby one by just touching one button can maintain the clients records very easily and gather the information about the payment done by the clients effectively and maintain all the above described records at just one place and owner does not have to worry about the due dates as well. This app will provide auto generated emails to both the clients and the owner as per the due dates.

## 2.3 SYSTEM REQUIREMENT SPECIFICATION:

### 2.3.1 GENERAL DESCRIPTION

**Title:**

System Requirement Specification for Payment reminder app.

**Team:**

Direct User (Owner of the properties), Admin.

### 2.3.2 Objective (Purpose):

The objective of this app is to provide timely due date reminders to the client of the Users using the application via email alongside with the payment QR code and provide the information to the user's customers also.

**Scope:**

This system allows the users to track the record for the payments from their respective property clients.

This System allows Admin for the analysis of vacant properties.

**Intended Audience:**

The app is intended for individuals who have a hard time keeping track of when their payments are due.

**Definitions:**

- **Portal:** Personalized Online Web Application.
- **Dashboard:** Each individual dashboard for the relevant information as per the profile example- User, Admin will have different information on their dashboard.
- **SRS** Software Requirement Specification.

- **GUI** Graphical User Interface.
- **Owner** The person who will participate in system.

### 2.3.3 SYSTEM REQUIREMENTS

#### **Functional Requirements:**

- Owner can add multiple clients for different properties.
- Owner can set reminders for specific payment due dates.
- Owner can view a list of upcoming payments.
- Owner can mark payments as paid.
- Owner can receive notifications for upcoming payments.
- Owner can set custom reminders for specific payments.
- Owner's client will get reminder for the due payment.
- Login Access will be given to User and Admin.

#### • **Owner Login:**

Owner will be able to login the application with the valid credentials and perform the tasks and logout.

#### • **Admin Login:**

Admin will be able to login through the same portal as admin and have complete authority to modify any details in the system as per requirement.

This application will enable the user (the owner of the properties) to keep a track on the payments of their customers and the user will also be able to get notifications 3 days beforehand the due date alongside of the QR code once the payment is done by the customer the Owner will also get the confirmation email.

#### **Nonfunctional Requirement:**

are those that relate to the overall performance of the application, reliability, security, and usability of a system or process, rather than specific functionalities or capabilities. In the context of a payment reminder process, some examples of non-functional requirements might include:

- **Confidentiality:** The process should be designed to protect the confidentiality of User's information.
- **Accuracy:** The process should be designed to ensure that the reminders are based on accurate and up-to-date information.

- **Efficiency:** The process should be designed to be efficient and minimize the time and resources required to manage payment reminder.
- **Transparency:** The process should be designed to be at most transparent for the users.
- **Scalability:** The process should be able to accommodate a large number of users without becoming unwieldy or inefficient.
- **User-friendliness:** The process should be easy for users and admin to use and understand.
- **Data security:** The process should be designed to protect against unauthorized access to user data.

## SYSTEM DESIGN

System design is the solution for the creation of a new system. This phase focuses on the detailed implementation of the feasible system. Its emphasis on translating design specifications to performance specification. System design has two phases of development.

- Logical Design
- Physical Design

During logical design phase the analyst describes inputs (sources), outputs(destinations), databases (data stores) and procedures (data flows) all in a format that meets the user requirements. The analyst also specifies the needs of the user at a level that virtually determines the information flow in and out of the system and the data resources. Here the logical design is done through data flow diagrams and database design. The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which specify exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data and produce the required report on a hard copy or display it on the screen.

### 3.1 INPUT AND OUTPUT DESIGN

#### 3.1.1 INPUT DESIGN:

Input design is the link that ties the information system into the world of its users. The input design involves determining the inputs, validating the data, minimizing the data entry and provides a multi-user facility. Inaccurate inputs are the most common cause of errors in data processing. Errors entered by the data entry operators can be controlled by input design. The user-originated inputs are converted to a computer-based format in the input design. Input data are collected and organized into groups of similar data. Once identified, the appropriate input media are selected for processing. All the input data are validated and if any data violates any conditions, the user is warned by a message. If the data satisfies all the conditions, it is transferred to the appropriate tables in the database. In this project the student details are to be entered at the time of registration. A page is designed for this

purpose which is user friendly and easy to use. The design is done such that users get appropriate messages when exceptions occur.

### 3.1.2 OUTPUT DESIGN:

Computer output is the most important and direct source of information to the user. Output design is a very important phase since the output needs to be in an efficient manner. Efficient and intelligible output design improves the system relationship with the user and helps in decision making. Allowing the user to view the sample screen is important because the user is the ultimate judge of the quality of output. The output module of this system is the selected notifications.

## DATABASE DESIGN

### 3.2 DATABASE

Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are • Primary key - the field that is unique for all the record occurrences • Foreign key - the field used to set relation between tables  
Normalization is a technique to avoid redundancy in the tables.

### 3.3 SYSTEM TOOLS

The various system tools that have been used in developing both the front end and the back end of the project are being discussed in this chapter.

#### 3.3.1 FRONT END

React is a library which is developed by Facebook are utilized to implement the frontend. React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications

usually requires the use of additional libraries for routing, as well as certain client-side functionality.

- Axios version: 1.3.4
- Bootstrap version: 5.2.3
- React version: 18.2.0
- React-bootstrap version: 2.7.2

### **3.3.2 BACKEND:**

The back end is implemented using MySQL which is used to design databases.

#### **MySQL:**

MySQL is the world's second most widely used open-source relational database management system (RDBMS). The SQL phrase stands for Structured Query Language. An application software called Navicert was used to design the tables in MySQL.

- MySQL version : 8.0.32

#### **Spring-Boot:**

This is used to connect MYSQL and fetch data from database and store the data in database. The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an addition to the Enterprise JavaBeans (EJB) model. The Spring Framework is Open-source Framework.

- Spring Boot version : 2.7.9

### Level Zero



Figure 3: Level 0 DFD for Admin

### Level One

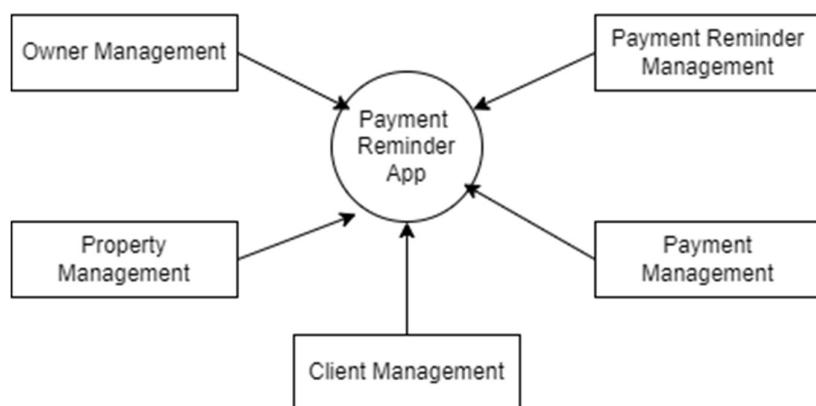


Figure 4: Level 1 DFD for Admin

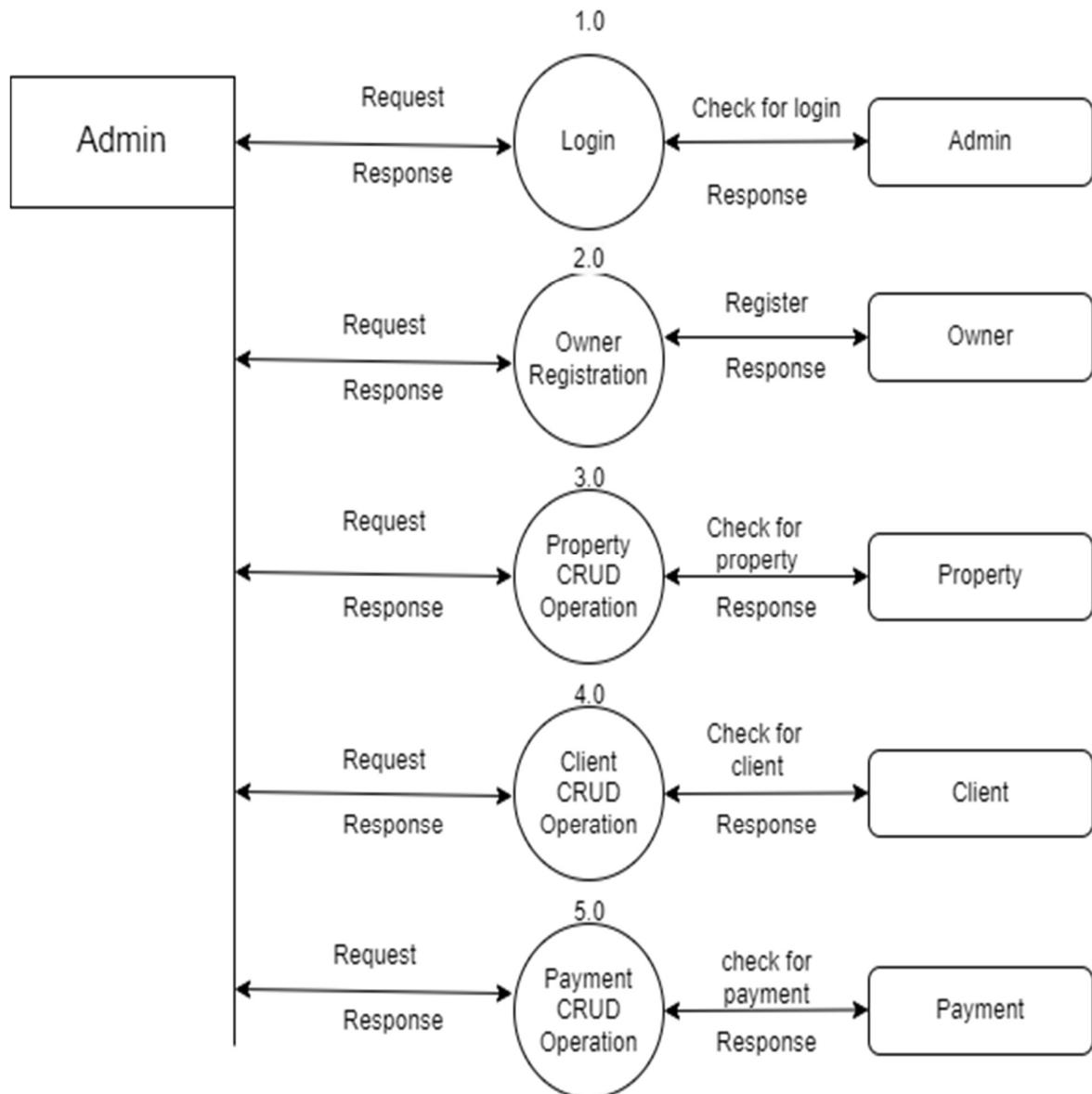


Figure 5: Level 2 DFD for Admin



Figure 6: Level 0 DFD for Owner

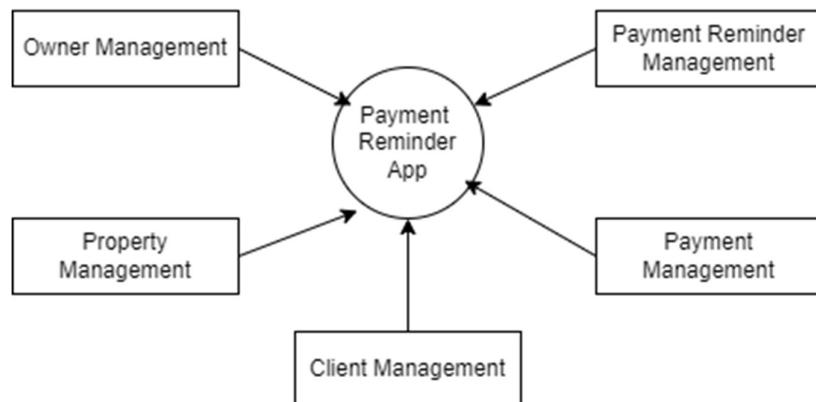


Figure 7: Level 1 DFD for Admin

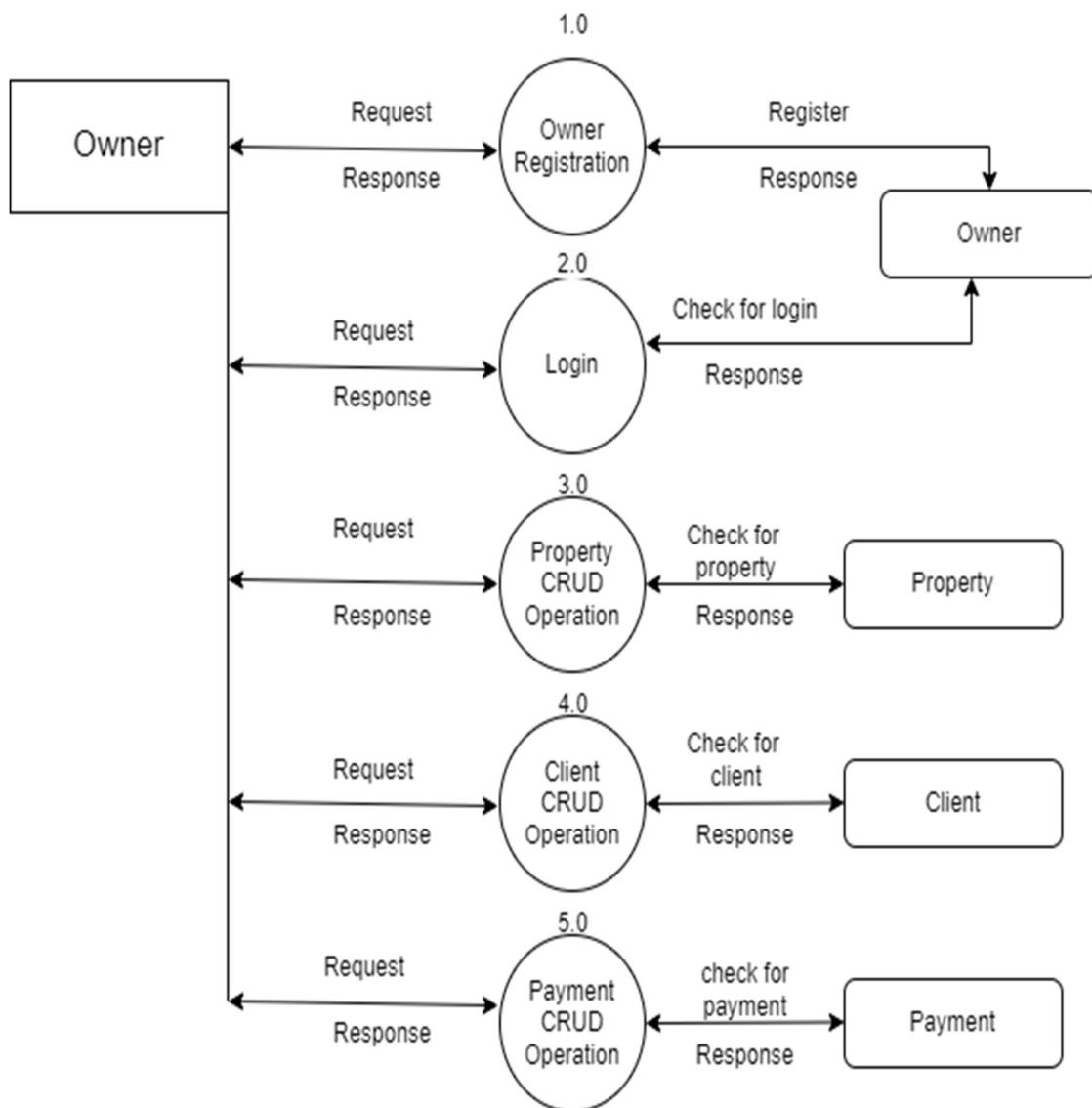


Figure 8: Level 2 DFD for Owner

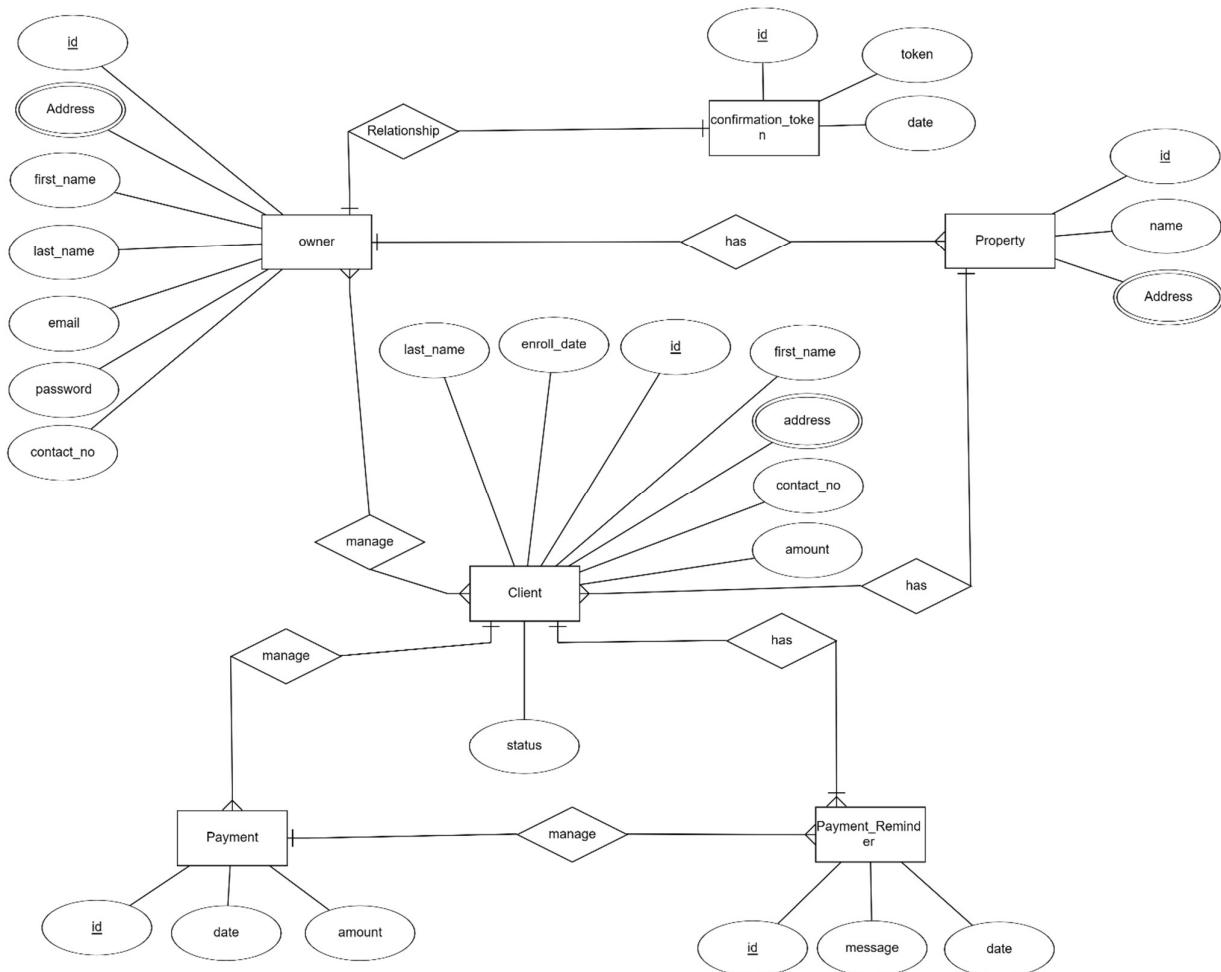


Figure 9: E-R Diagram

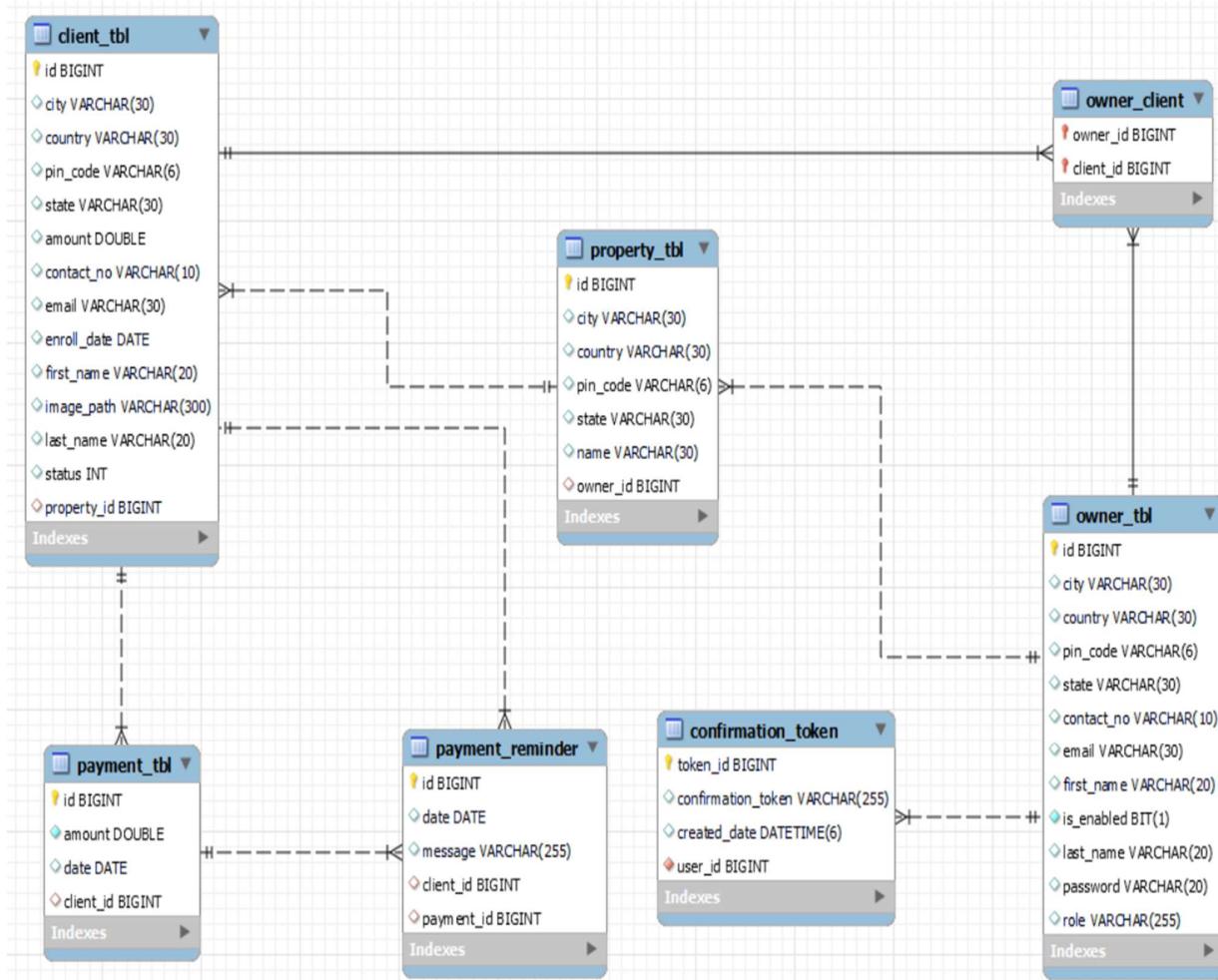


Figure 10: E-R Diagram Auto Generated

## Use Case Diagrams

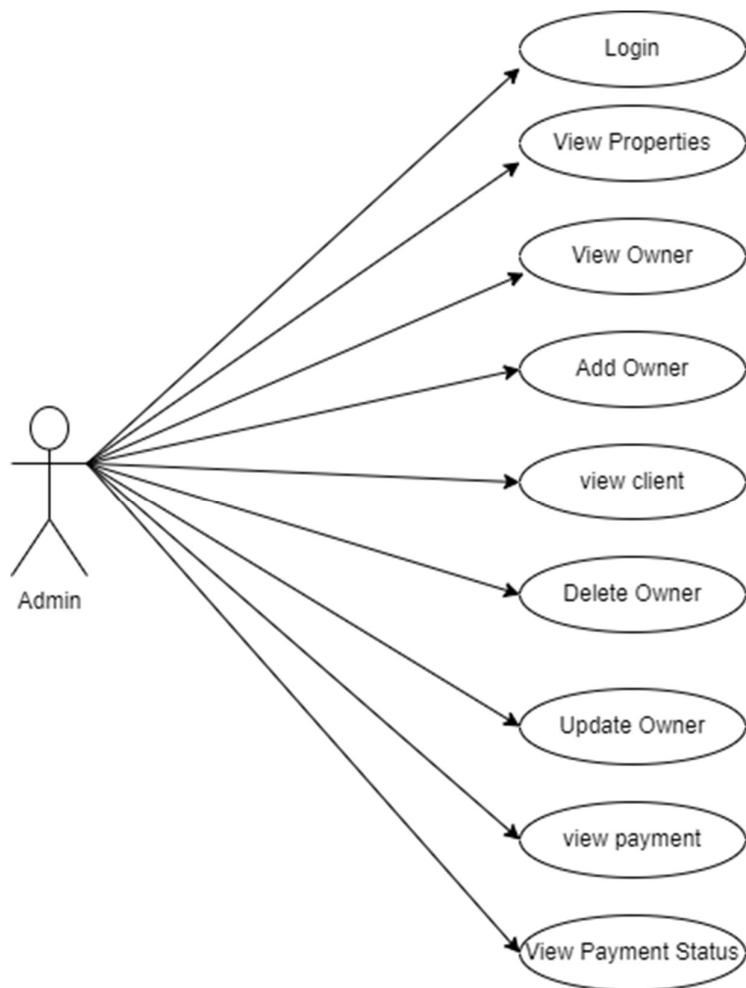


Figure 11: Admin Use Case Diagram

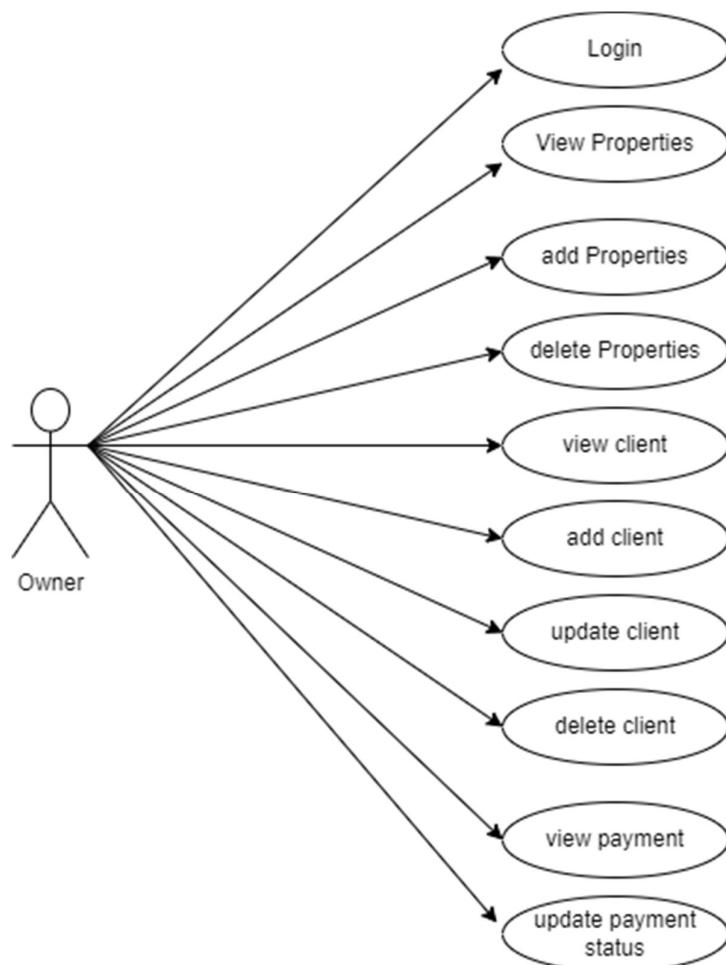


Figure 12: Owner Use Case Diagram

## CLASS DIAGRAM

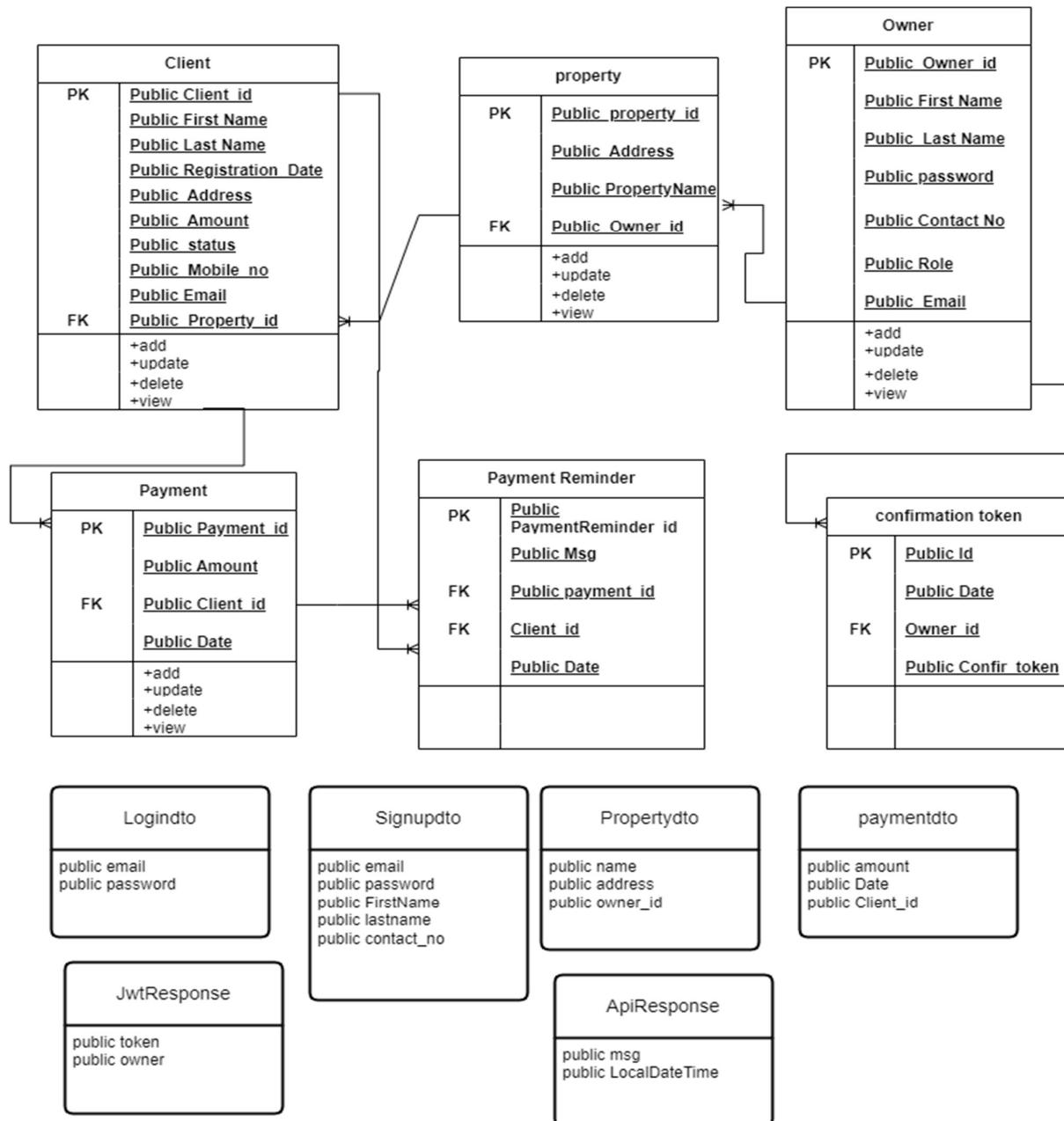


Figure 13: Class Diagram

## TABLE STRUCTURE:

Tables_in_pra
client_tbl
owner_client
owner_tbl
payement_tbl
payment_reminder
properties

Figure 14: Tables

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
city	varchar(30)	YES		NULL	
country	varchar(30)	YES		NULL	
pincode	varchar(6)	YES		NULL	
state	varchar(30)	YES		NULL	
email	varchar(30)	NO		NULL	
first_name	varchar(20)	NO		NULL	
last_name	varchar(20)	NO		NULL	
mobile_no	varchar(10)	NO		NULL	
pwd	varchar(20)	YES		NULL	
role	int	YES		NULL	

Figure 15: Owner Table

mysql> desc properties;						
Field	Type	Null	Key	Default	Extra	
id	bigint	NO	PRI	NULL	auto_increment	
city	varchar(30)	YES		NULL		
country	varchar(30)	YES		NULL		
pincode	varchar(6)	YES		NULL		
state	varchar(30)	YES		NULL		
prop_name	varchar(20)	NO		NULL		
owner_id	bigint	YES	MUL	NULL		

Figure 16: Property Table

mysql> desc client;						
Field	Type	Null	Key	Default	Extra	
id	bigint	NO	PRI	NULL	auto_increment	
city	varchar(30)	YES		NULL		
country	varchar(30)	YES		NULL		
pincode	varchar(6)	YES		NULL		
state	varchar(30)	YES		NULL		
email	varchar(30)	NO		NULL		
enroll_date	date	NO		NULL		
first_name	varchar(20)	NO		NULL		
last_name	varchar(20)	NO		NULL		
mobile_number	varchar(20)	NO		NULL		
rent_amount	double	NO		NULL		
rent_status	int	NO		NULL		
property_id	bigint	YES	MUL	NULL		

Figure 17: Client Table

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
amount	double	NO		NULL	
date	date	YES		NULL	
client_id	bigint	YES	MUL	NULL	

Figure 18: Payment Table

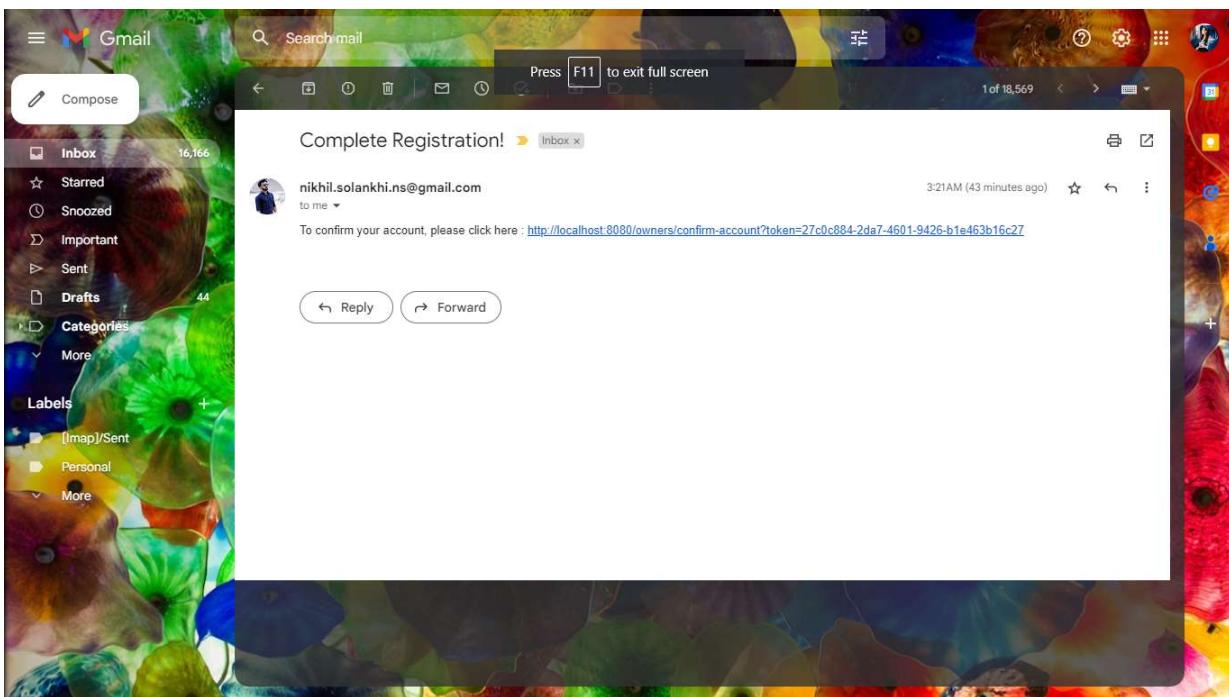
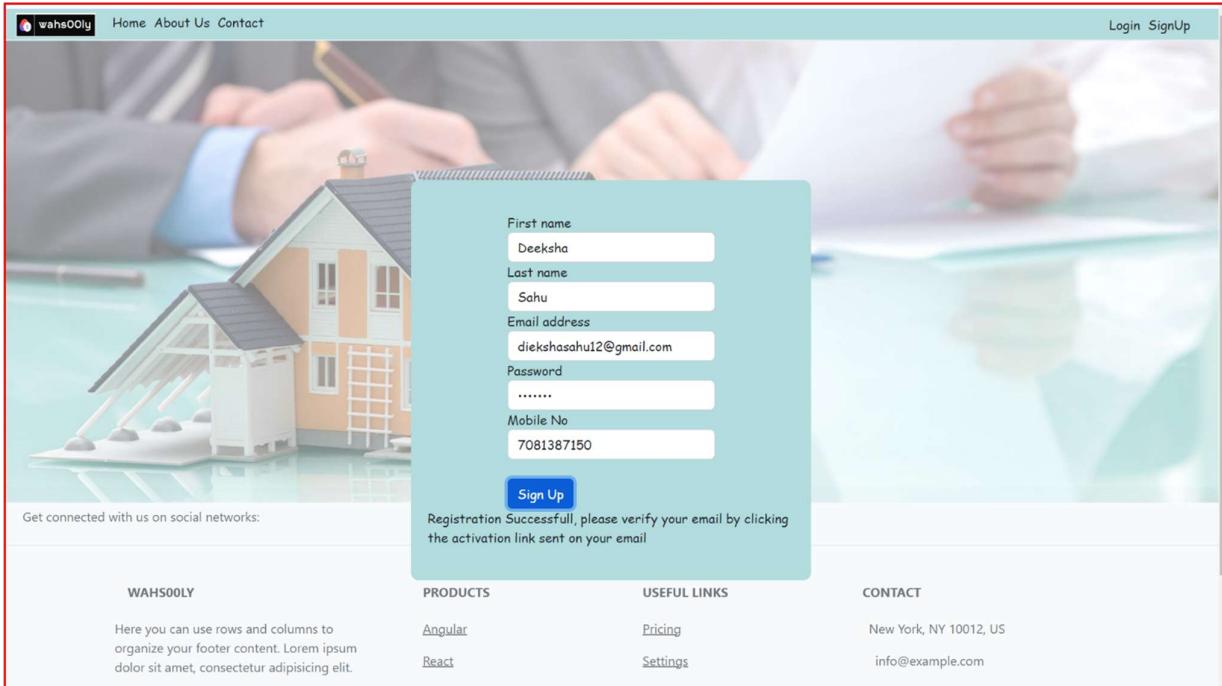
Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
date	date	YES		NULL	
message	varchar(255)	YES		NULL	
client_id	bigint	YES	MUL	NULL	
payment_id	bigint	YES	MUL	NULL	

Figure 19: Payment Reminder Table

Field	Type	Null	Key	Default	Extra
owner_id	bigint	NO	PRI	NULL	
client_id	bigint	NO	PRI	NULL	

Figure 20: Owner Client Relation Table

## PROJECT DIAGRAMS



Your email is verified please visit the login page <http://localhost:8080/owners/signIn>

Press F11 to exit full screen

The screenshot shows the Wahs00ly login page. At the top, there is a navigation bar with links for Home, About Us, Contact, Login, and SignUp. A large background image of a person signing documents is visible. In the center, there is a modal window for logging in. It contains fields for Email address (with the value diekshasahu12@gmail.com) and Password (with the value .....). Below these fields is a blue 'Login' button. To the left of the modal, there is a small image of a house model.

Get connected with us on social networks:

WAHS00LY

Here you can use rows and columns to organize your footer content. Lorem ipsum dolor sit amet, consectetur adipisicing elit.

PRODUCTS

[Angular](#) [React](#)

USEFUL LINKS

[Pricing](#) [Settings](#)

CONTACT

New York, NY 10012, US  
info@example.com  
91-22-1567-89

The screenshot shows the properties page for Mr. Deeksha. At the top, there is a navigation bar with links for Home, About Us, Contact, and a user profile section with links for Profile and Logout. A large background image of a person signing documents is visible. In the center, there is a modal window titled "Properties of Mr. Deeksha". It contains a "Add Properties" button and a table with columns for Property Name, City, State, Country, Pin Code, Delete, Edit, and View. Below the table is a small image of a house model.

Get connected with us on social networks:

WAHS00LY

Here you can use rows and columns to organize your footer content. Lorem ipsum dolor sit amet, consectetur adipisicing elit.

PRODUCTS

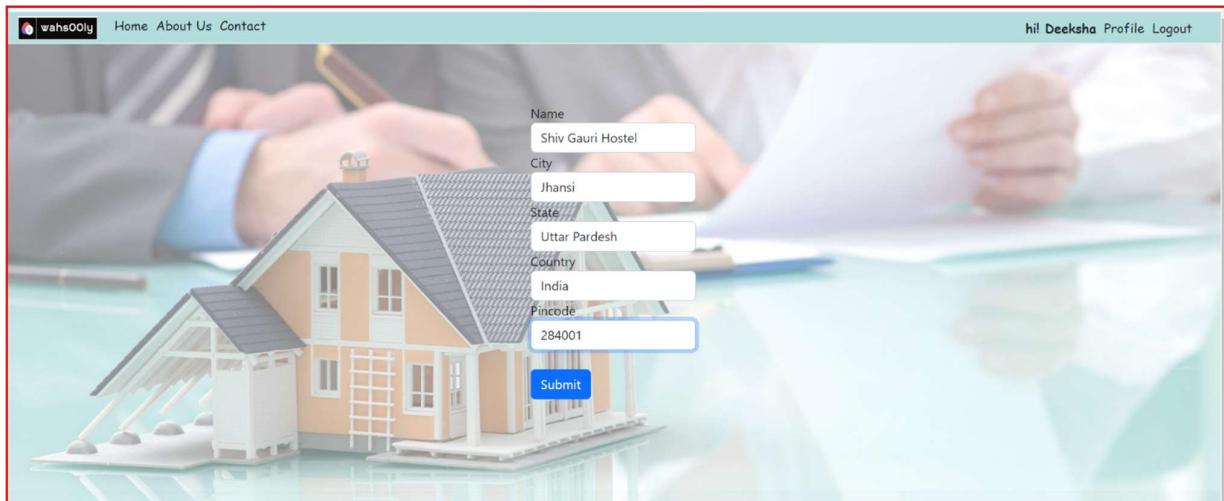
[Angular](#) [React](#)

USEFUL LINKS

[Pricing](#) [Settings](#)

CONTACT

New York, NY 10012, US  
info@example.com  
91-22-1567-89



A screenshot of a web application for property management. In the center, there is a form for adding a new property. The fields are as follows:

- Name: Shiv Gauri Hostel
- City: Jhansi
- State: Uttar Pradesh
- Country: India
- Pincode: 284001

Below the form is a "Submit" button. The background features a 3D rendering of a house.

Get connected with us on social networks:

**WAHS00LY**

Here you can use rows and columns to organize your footer content. Lorem ipsum dolor sit amet, consectetur adipisicing elit.

**PRODUCTS**

- [Angular](#)
- [React](#)

**USEFUL LINKS**

- [Pricing](#)
- [Settings](#)

**CONTACT**

New York, NY 10012, US  
info@example.com



A screenshot of a web application showing a list of properties for Mr. Deeksha. The table displays the following data:

Property Name	City	State	Country	Pin Code	Delete	Edit	View Clients
Shiv Gauri Hostel	Jhansi	Uttar Pradesh	India	284001	<a href="#">Delete</a>	<a href="#">Edit</a>	<a href="#">View Clients</a>
Rudra	Bikaner	Rajasthan	India	335001	<a href="#">Delete</a>	<a href="#">Edit</a>	<a href="#">View Clients</a>
Ganga Girls PG	Ghaziabad	Delhi	India	987456	<a href="#">Delete</a>	<a href="#">Edit</a>	<a href="#">View Clients</a>
Sidhi	Pune	Maharashtra	India	411018	<a href="#">Delete</a>	<a href="#">Edit</a>	<a href="#">View Clients</a>
Dutta Girls PG	Lucknow	Uttar Pradesh	India	941456	<a href="#">Delete</a>	<a href="#">Edit</a>	<a href="#">View Clients</a>

Get connected with us on social networks:

**WAHS00LY**

Here you can use rows and columns to organize your footer content. Lorem ipsum dolor sit amet, consectetur adipisicing elit.

**PRODUCTS**

- [Angular](#)
- [React](#)

**USEFUL LINKS**

- [Pricing](#)
- [Settings](#)

**CONTACT**

New York, NY 10012, US  
info@example.com

The screenshot shows a web application interface for managing client data. At the top, there's a navigation bar with links for Home, About Us, Contact, Profile, and Logout. Below the header, a banner features a 3D model of a house and the text "Clients at Shiv Gauri Hostel property". A blue button labeled "Add Client" is visible. A table header row includes columns for First name, Last name, Mobile No, Email, Rent Amount, Registration date, City, State, Country, Pincode, Delete, Edit, and View. The main content area displays a table with several rows of client information.

Get connected with us on social networks:

WAHS00LY

Here you can use rows and columns to organize your footer content. Lorem ipsum dolor sit amet, consectetur adipisicing elit.

PRODUCTS

[Angular](#)

[React](#)

USEFUL LINKS

[Pricing](#)

[Settings](#)

CONTACT

New York, NY 10012, US

info@example.com

The screenshot shows a modal dialog box for adding a new client. The form fields include:

- First name: Nikhil
- Last name: Solankhi
- Mobile No: 8947959408
- Email: nikhil.solankhi.ns@gmail.com
- Registration date: 09-03-2023
- Amount: 5500
- City: Pune
- State: Maharashtra
- Country: India
- Pincode: 411018

A "Submit" button is located at the bottom of the form.

Get connected with us on social networks:

WAHS00LY

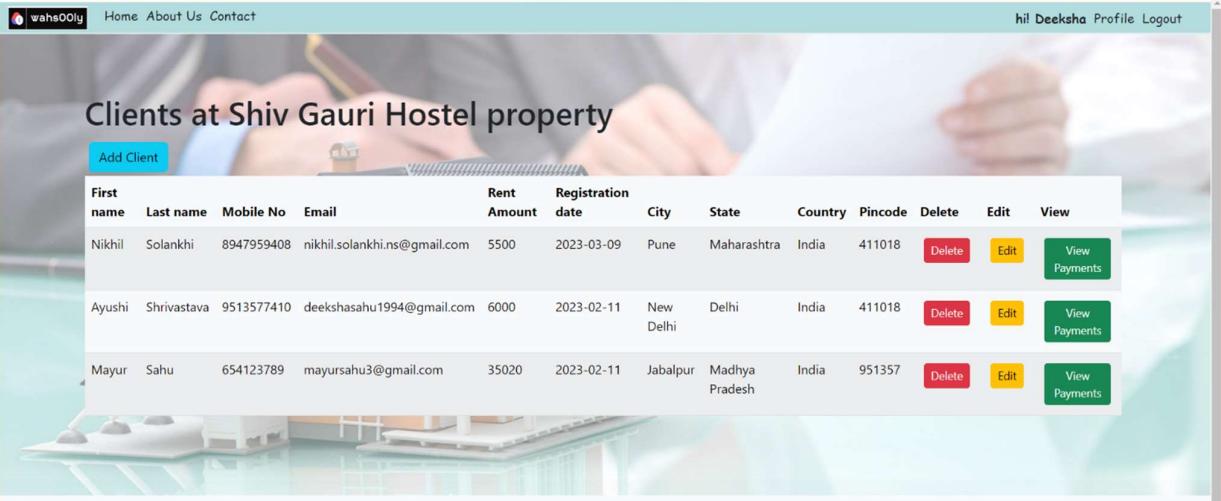
Here you can use rows and columns to organize your footer content. Lorem ipsum dolor sit amet, consectetur adipisicing elit.

NY 10012, US

ample.com

567 88

567 89



**Clients at Shiv Gauri Hostel property**

First name	Last name	Mobile No	Email	Rent Amount	Registration date	City	State	Country	Pincode	Delete	Edit	View Payments
Nikhil	Solankhi	8947959408	nikhil.solankhi.ns@gmail.com	5500	2023-03-09	Pune	Maharashtra	India	411018	<button>Delete</button>	<button>Edit</button>	<button>View Payments</button>
Ayushi	Shrivastava	9513577410	deekshasahu1994@gmail.com	6000	2023-02-11	New Delhi	Delhi	India	411018	<button>Delete</button>	<button>Edit</button>	<button>View Payments</button>
Mayur	Sahu	654123789	mayursahu3@gmail.com	35020	2023-02-11	Jabalpur	Madhya Pradesh	India	951357	<button>Delete</button>	<button>Edit</button>	<button>View Payments</button>

Get connected with us on social networks:

WAHS00LY
PRODUCTS
USEFUL LINKS
CONTACT

Angular
Pricing
New York, NY 10012, US

React
Settings
info@example.com



**Payments of Mr. Mayur**

Add Payment	Amount	Paid on	Due Date

Get connected with us on social networks:

WAHS00LY
PRODUCTS
USEFUL LINKS
CONTACT

Angular
Pricing
New York, NY 10012, US

React
Settings
info@example.com

The screenshot shows a payment interface overlaid on a background image of a house model and a person signing documents. The form includes fields for Amount (35020) and Last Date (11-02-2023), with a Submit button. The top navigation bar has links for Home, About Us, Contact, and a user profile. The footer contains social media links and a footer menu.

Amount  
35020

Last Date  
11-02-2023

Submit

Get connected with us on social networks:

WAHS00LY

Here you can use rows and columns to organize your footer content. Lorem ipsum dolor sit amet, consectetur adipisicing elit.

PRODUCTS

[Angular](#)

[React](#)

USEFUL LINKS

[Pricing](#)

[Settings](#)

CONTACT

New York, NY 10012, US

info@example.com

The screenshot shows a table titled "Payments of Mr. Mayur" overlaid on a background image of a house model and a person signing documents. The table lists two payments: one for 30000 on 2023-01-11 due by 2023-02-11, and another for 35020 on 2023-02-11 due by 2023-03-11. The top navigation bar has links for Home, About Us, Contact, and a user profile. The footer contains social media links and a footer menu.

Amount	Paid on	Due Date
30000	2023-01-11	2023-02-11
35020	2023-02-11	2023-03-11

Get connected with us on social networks:

WAHS00LY

Here you can use rows and columns to organize your footer content. Lorem ipsum dolor sit amet, consectetur adipisicing elit.

PRODUCTS

[Angular](#)

[React](#)

USEFUL LINKS

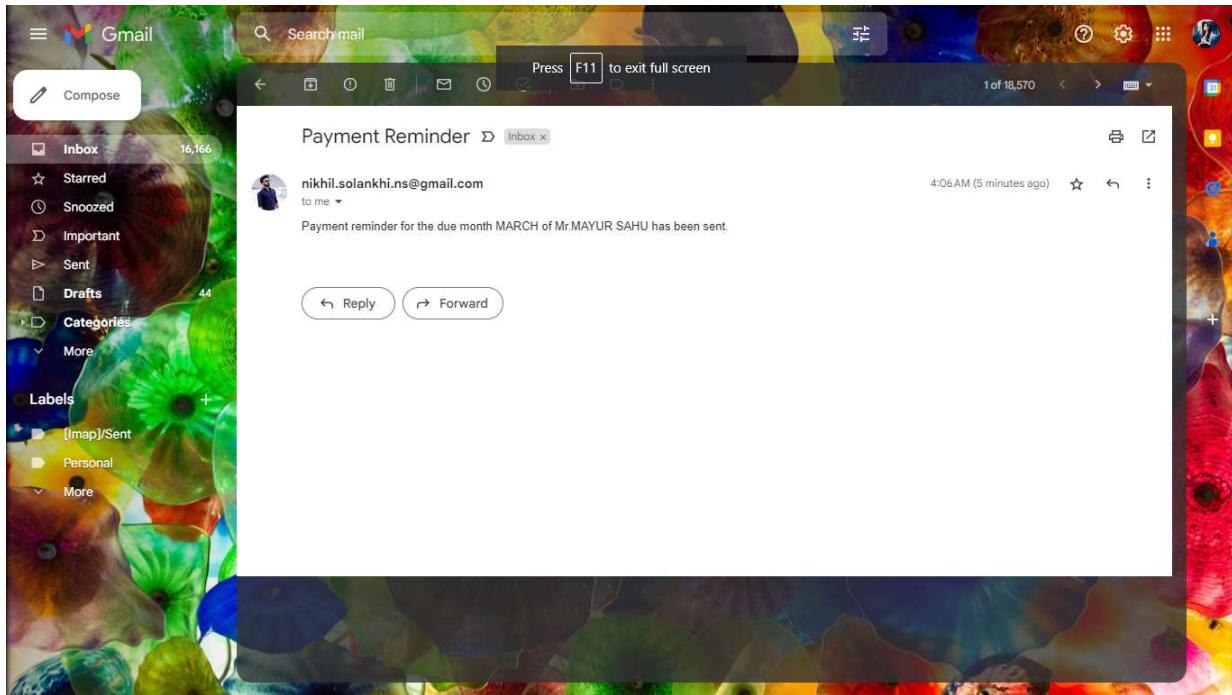
[Pricing](#)

[Settings](#)

CONTACT

New York, NY 10012, US

info@example.com



## CONCLUSION

Payment reminder apps for rental management can be particularly useful for property managers who manage multiple properties and tenants, as keeping track of rent payments manually can be time-consuming and prone to errors. By using a payment reminder app, property managers can automate the rent collection process and free up time to focus on other important tasks.

In addition to sending rent reminders, some payment reminder apps for rental management may also allow tenants to make rent payments directly through the app, making the payment process more convenient for both tenants and landlords. Some apps may also offer features such as online rent collection, lease management, and maintenance requests.

When choosing a payment reminder app for rental management, it's important to consider factors such as cost, ease of use, security, and customer support. Some apps may require a monthly subscription fee or charge a transaction fee for online rent collection, so it's important to weigh the costs and benefits before making a decision.

Overall, a payment reminder app for rental management can be a valuable tool for landlords and property managers, helping to streamline the rent collection process, reduce late payments, and improve overall efficiency and organization in rental management.

## Future Scope

1. Mobile app: Creating a mobile app version of the PRA rental application would allow landlords and property managers to manage their properties and communicate with tenants while on the go.
2. Analytics and reporting: Adding analytics and reporting functionality to the PRA rental application could provide landlords and property managers with valuable insights into their rental properties' performance, including occupancy rates, revenue, and expenses.
3. Virtual reality tours: Incorporating virtual reality tours into the PRA rental application could provide prospective tenants with a more immersive rental property viewing experience.
4. Language translation: Providing language translation services within the PRA rental application could make it easier for landlords and property managers to communicate with tenants who speak different languages.
5. Streamlining payment processes: A payment reminder app can automate the payment process, which can save time and effort for both landlords and tenants.
6. Improved cash flow: By sending timely payment reminders, landlords can improve their cash flow and avoid any late payment fees or penalties.
7. Reduced late payments: An app that sends reminders to tenants can help reduce the number of late payments, which can be beneficial for landlords.
8. Enhanced tenant experience: By providing tenants with a user-friendly payment system, landlords can improve their overall rental experience and increase tenant satisfaction.
9. Increased efficiency: An app can help landlords manage multiple properties more efficiently, reducing the administrative workload.
10. Integration with other property management tools: An app can be integrated with other property management tools, such as accounting software, to provide a comprehensive management solution.

## REFERENCES

- JavaScript Enlightenment, Cody Lindley-First Edition, based on JavaScript 1.5, ECMA-262, Edition
- Mc Graw Hill's, Java: The complete reference 7thEdition, HerbertScheldt
- Complete CSS Guide, Maxine Sherrin and John Allsopp-O'ReillyMedia; September 2012

## ONLINE REFERENCE

- [www.w3school.com](http://www.w3school.com)
- [www.javatpoint.com](http://www.javatpoint.com)
- [www.github.com](http://www.github.com)
- [www.geeksforgeeks.com](http://www.geeksforgeeks.com)
- <https://docs.oracle.com/en/java/>
- <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>
- <https://getbootstrap.com/>
- <https://jwt.io/>
- <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>
- <https://reactjs.org/>