# COMMUNITY LINK APP

## RESIDENTIAL MANAGEMENT SYSTEM

SUBMITTED BY:
ANUPAMA .A

# TABLE OF CONTENT

# 1. INTRODUCTION

**CommunityLink** is a comprehensive full-stack web application developed to enhance the management and communication processes within residential communities. The platform serves as a centralized hub where residents, community managers, and administrators can seamlessly interact and carry out essential tasks. From managing announcements and events to handling maintenance requests, payments, and feedback, CommunityLink aims to simplify and digitalize day-to-day community operations.

The core objective of CommunityLink is to foster better engagement, transparency, and efficiency by integrating all key community functions into a single, user-friendly system. By offering real-time updates, role-based access, and easy information sharing, the platform empowers both residents and administrators to stay informed and connected.

This project holds significant value as it addresses the often outdated and fragmented nature of traditional community management methods. Through the use of modern full-stack development technologies, CommunityLink provides a scalable and reliable solution that can adapt to the needs of different residential setups.

# 2.PROBLEM STATEMENT

Residential communities often encounter a range of operational challenges that hinder effective management and communication. Key issues include difficulty in organizing and managing community events, delays or inconsistencies in disseminating important announcements, and a lack of streamlined processes for collecting and tracking association payments. These problems are typically rooted in outdated manual systems, scattered communication channels, and the absence of a centralized platform to coordinate community affairs.

As a result, residents may miss critical updates, feel disconnected from community activities, or face inconveniences when fulfilling their responsibilities such as paying dues or submitting requests. Similarly, community administrators struggle with maintaining records, ensuring timely communication, and managing engagement across various platforms, which can lead to inefficiencies, frustration, and a decline in overall community cohesion.

This project seeks to solve these challenges by developing a centralized, web-based platform that integrates all essential community management functions. By transitioning to a digital solution, the system aims to streamline operations, foster better engagement among residents, and significantly improve the overall efficiency of community administration.

# 3.PROPOSED SOLUTION

The **CommunityLink** application is designed to serve as a centralized digital platform that automates and streamlines the core aspects of residential community management. By replacing fragmented, manual processes with a unified web-based system, CommunityLink aims to enhance efficiency, transparency, and resident engagement across the board.

At the heart of the application are several key features that cater to the specific needs of both residents and community administrators:

**Event Booking:** Residents can easily,reserve common areas or facilities (like clubhouses, parks, or halls), and receive notifications about scheduled activities. Administrators can manage event space, monitor availability,.

**Announcement Management:** Important community-wide announcements such as maintenance schedules, safety alerts, or meeting notices can be quickly published and shared with all residents. Notifications ensure timely delivery.

**Association Payment Processing:** The platform simplifies the management of association dues and other community-related payments. Residents can

make secure online payments, Administrators can track residents paid list

# 4. USER STORIES

- **User Story 1:**

  - "As a resident, I want to view upcoming community events and book my spot, so that I can easily conduct the function."
  - "Acceptance Criteria: The system displays event details (date, time, location). Residents can book for events ."

- **User Story 2:**

  - "As an administrator, I want to post announcements,events and dues to residents, so that I can keep them informed about important information."
  - "Acceptance Criteria: Administrators can create and publish announcements,events and dues details. Residents receive notifications."

- **User Story 3:**

  - "As a resident, I want to pay my association fees online, so that I can manage my dues conveniently."
  - "Acceptance Criteria: Residents can view dues and make online payments. "

# 5. TECHNOLOGY STACK

"The project will utilize the MERN stack:

- Front-end: React.js
- Back-end: Node.js, Express
- Authentication: JWT (JSON Web Tokens)
- Database: MongoDB Atlas
- Version Control: Git, GitHub

# 6.DEVELOPMENT ENVIRONMENT

"The development environment will include:

- IDE: Visual Studio Code
- Version Control: Git, GitHub
- Database Management Tool: MongoDB Atlas
- Package Manager: npm
- API Testing: Postman

# 7.PROJECT STRUCTURE

The project is organized into the following modules:

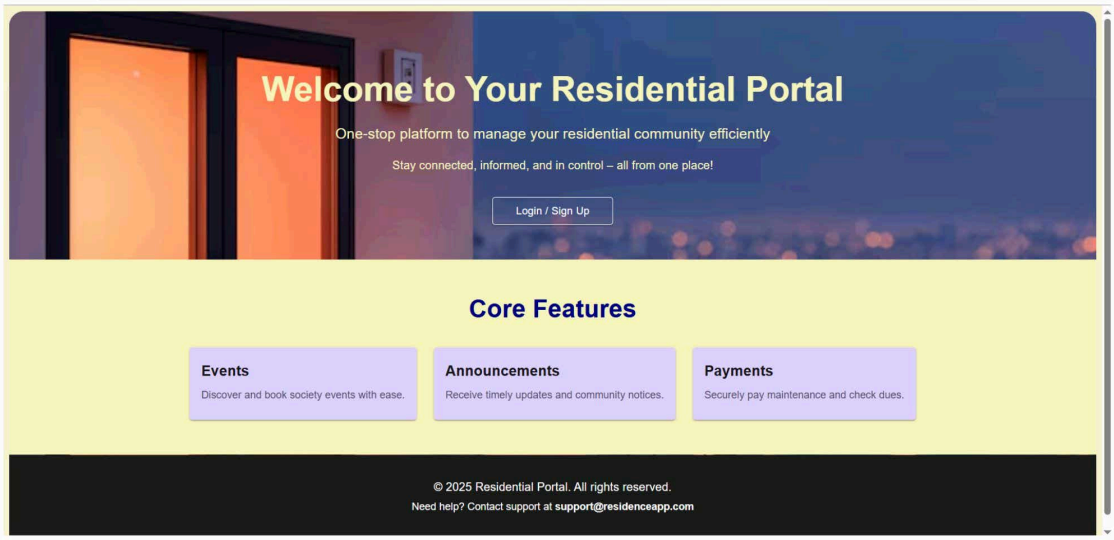
- **Front End:** Contains React components for user interface (e.g., login, Announcement,event, event booking,dues,logout).
- **Back End:** Contains Express routes,  and models for handling API requests.
- **Database:** MongoDB collections for users, bookings,announcement,event,dues data.
- **Admin Panel:** A separate module for admin functionalities like publication of announcement ,events.dues,dues paid list,votelist and user management.

# 8.CONCLUSION

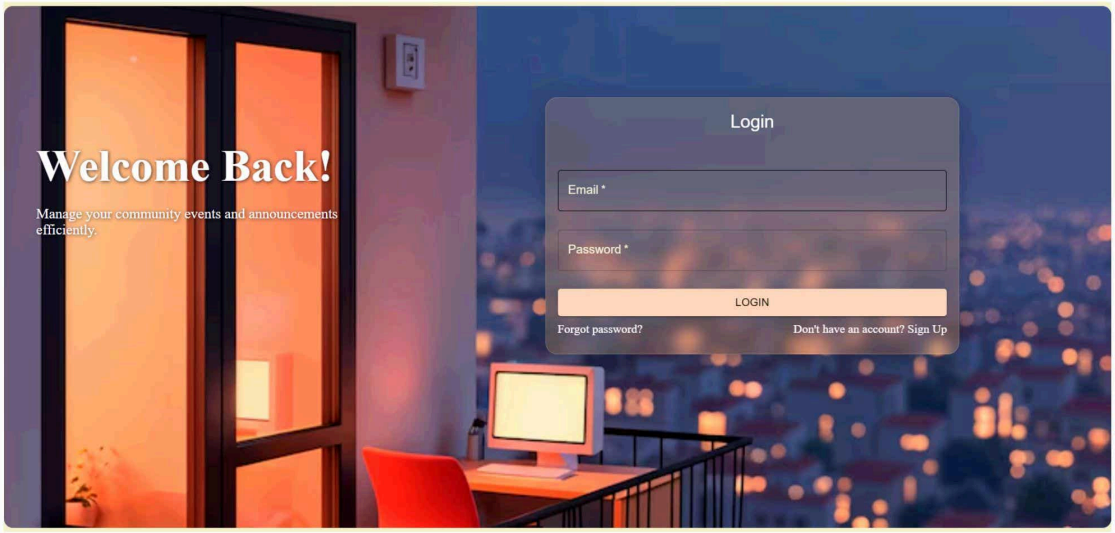
The Residential Management System streamlines community living by integrating essential features like announcements, event booking, and dues clearance into a unified platform. Residents stay informed through timely announcements, engage with the community via hassle-free event booking, and manage financial obligations with secure and transparent dues payment. This system enhances communication, participation, and accountability, creating a more connected and efficiently managed residential environment.

# 9.IMAGES

## 1.HOME PAGE

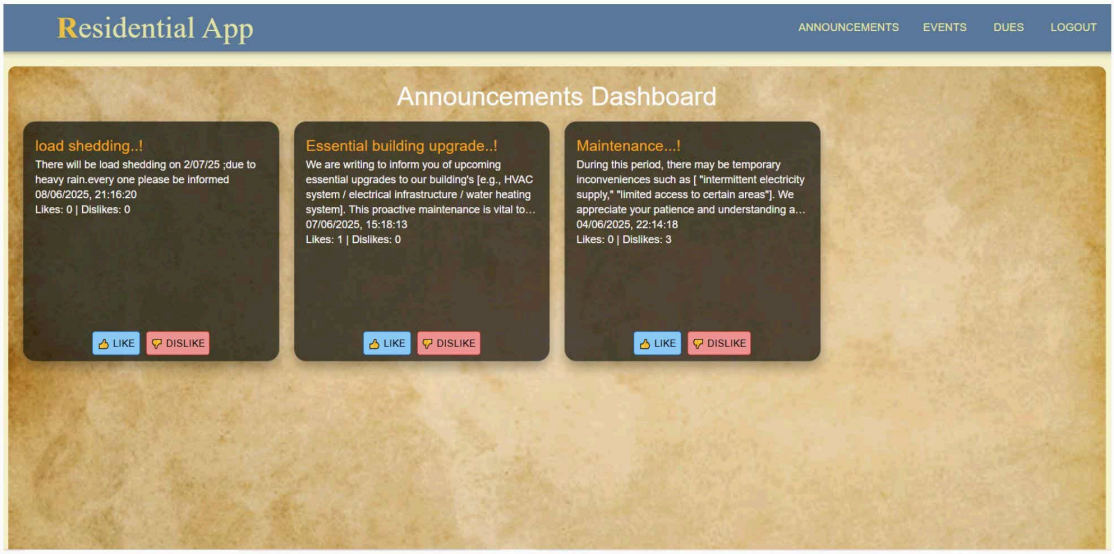



## 2.LOGIN PAGE

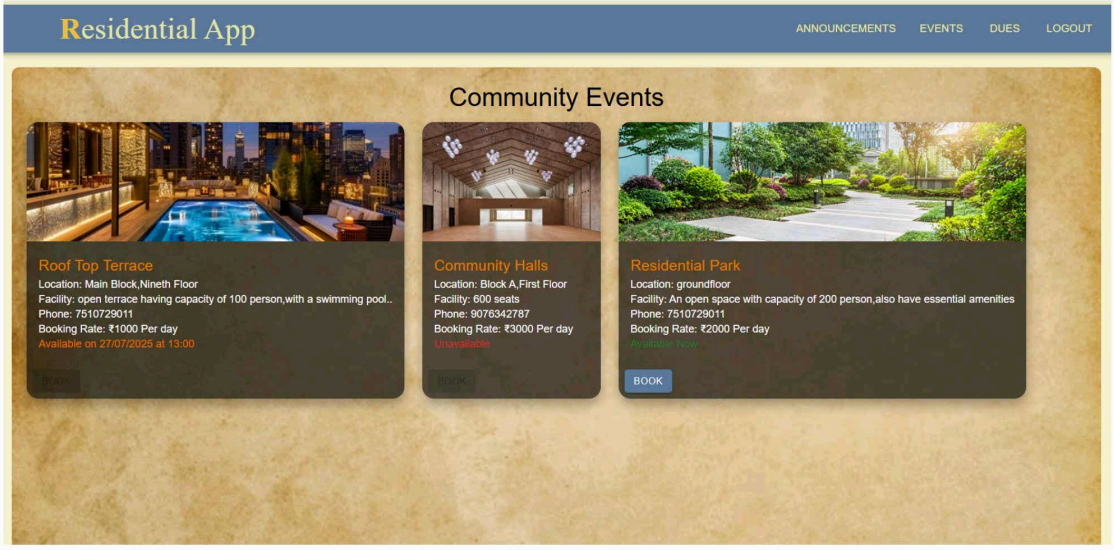


## 3.USER PAGES

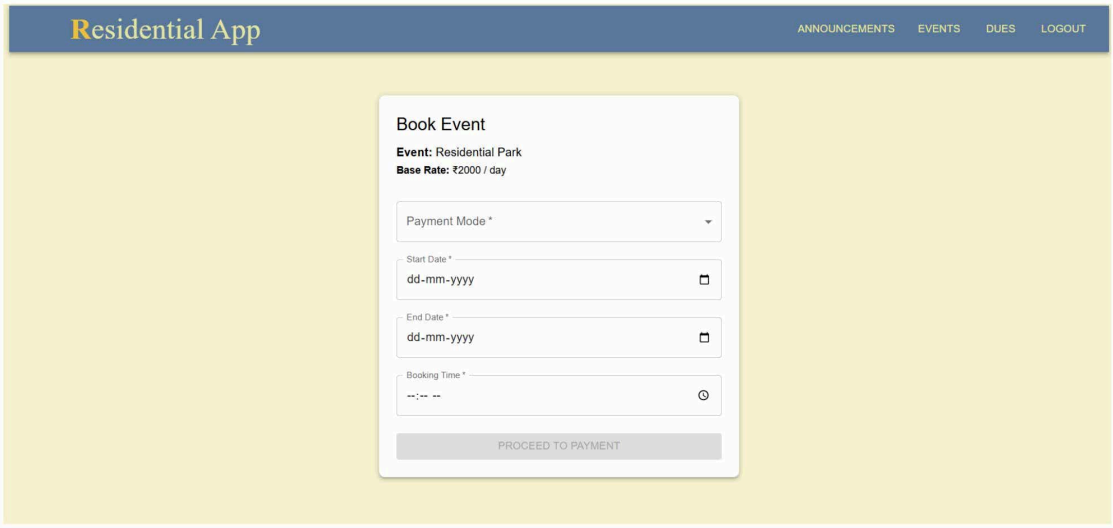
### a)Announcement Page



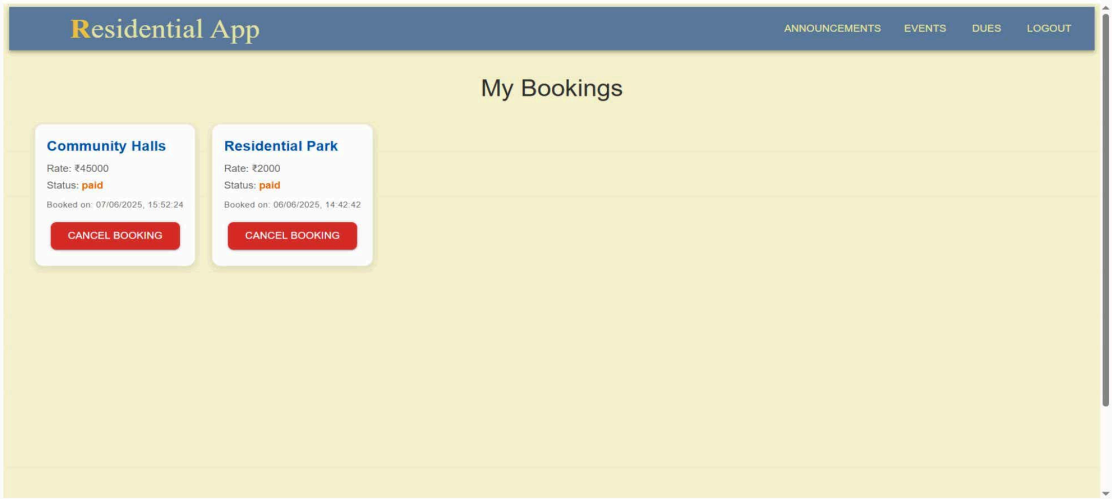

### b)event Page



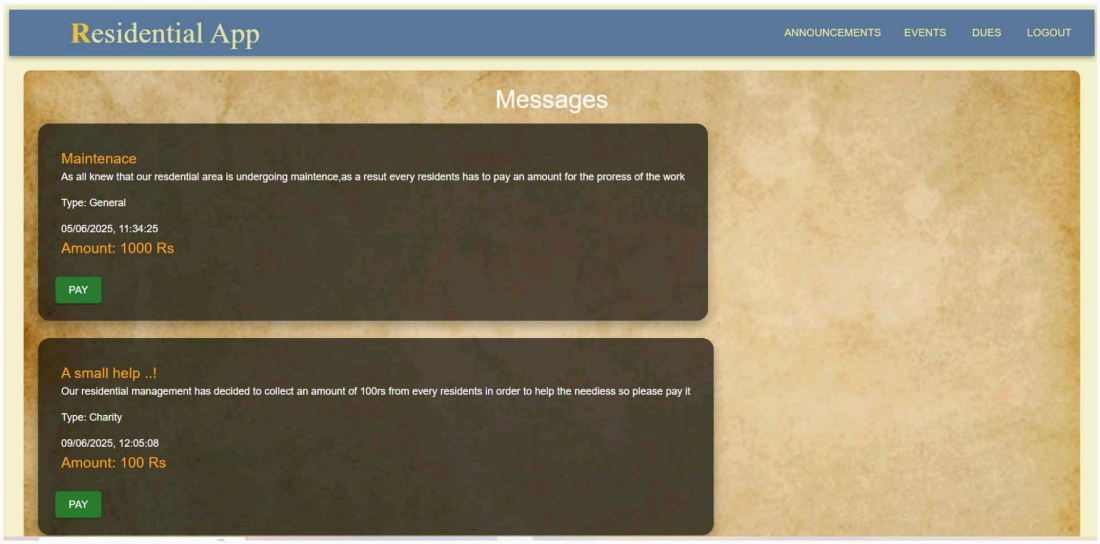
## c)Event Booking Page




## d)My Bookings Page

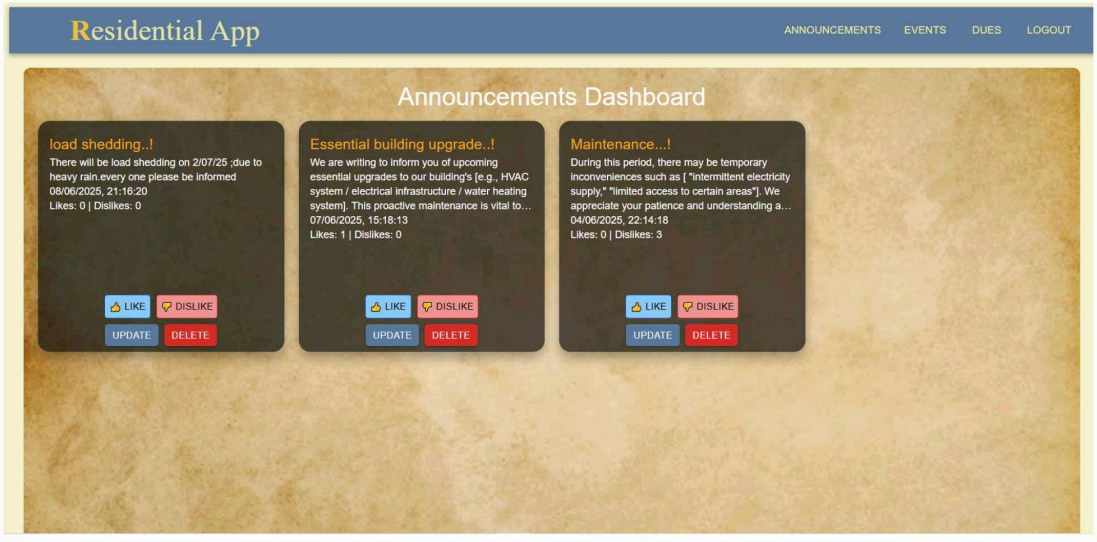


### e)Dues page

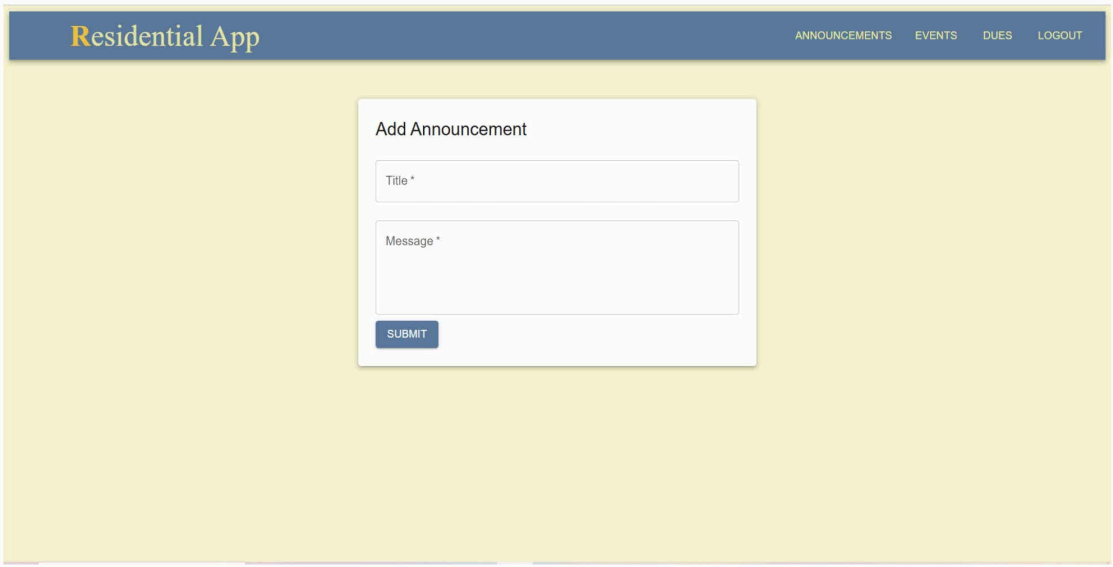



## 4.ADMIN PAGES

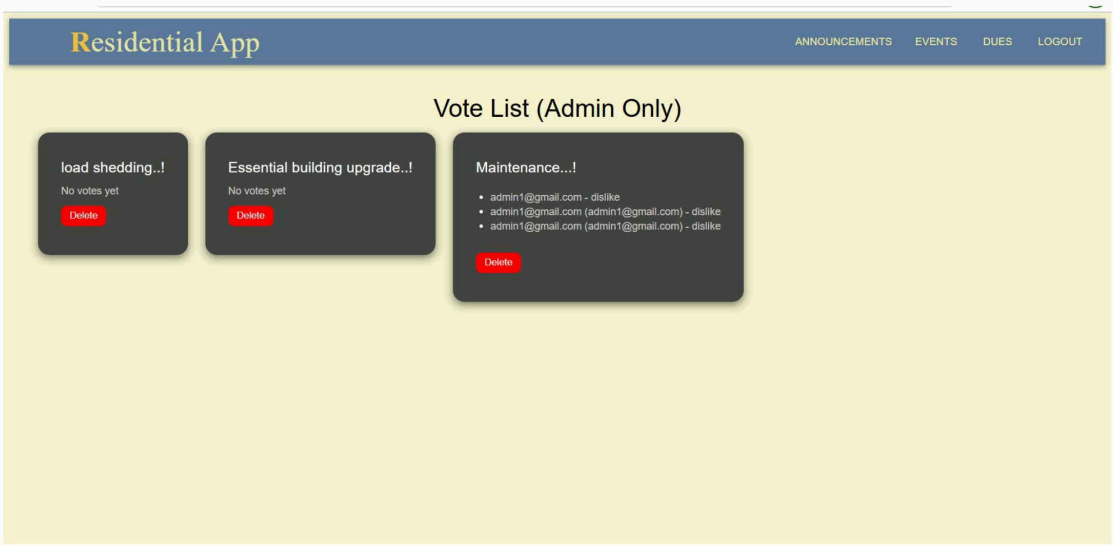
### a)Announcement page



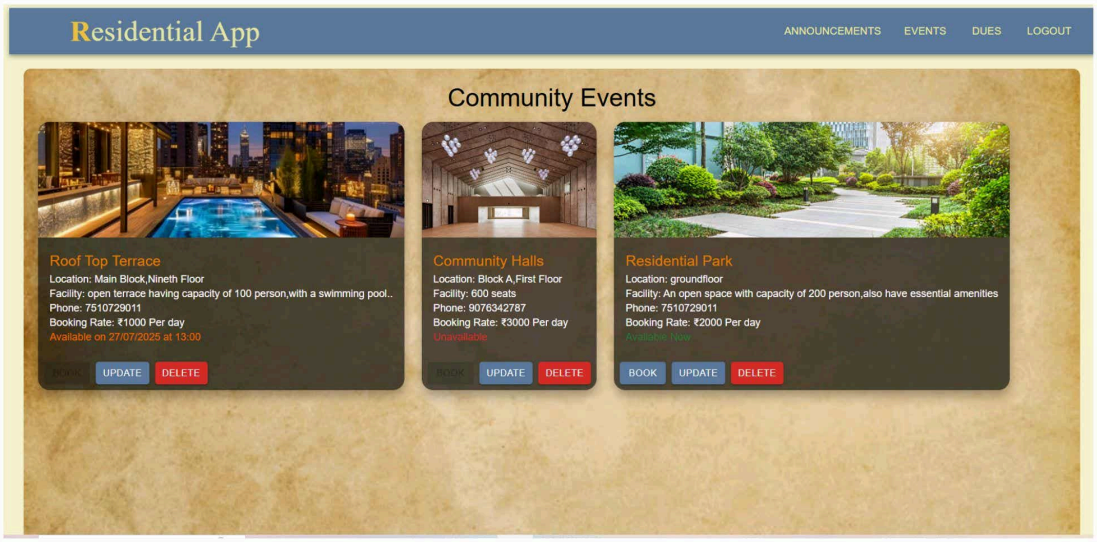
## b)Add Announcement Page



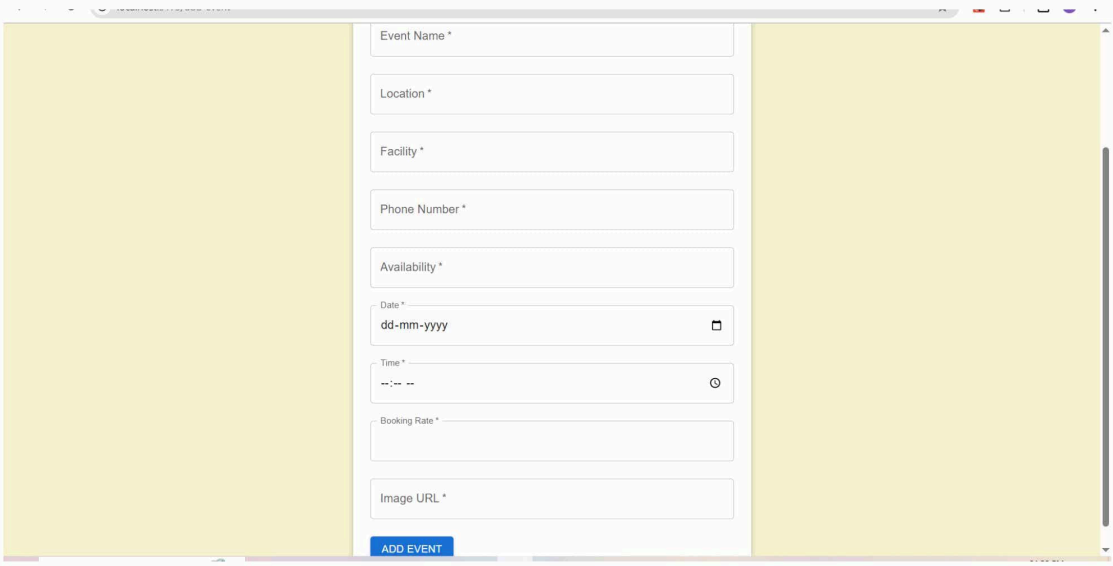

## d)Vote List Page



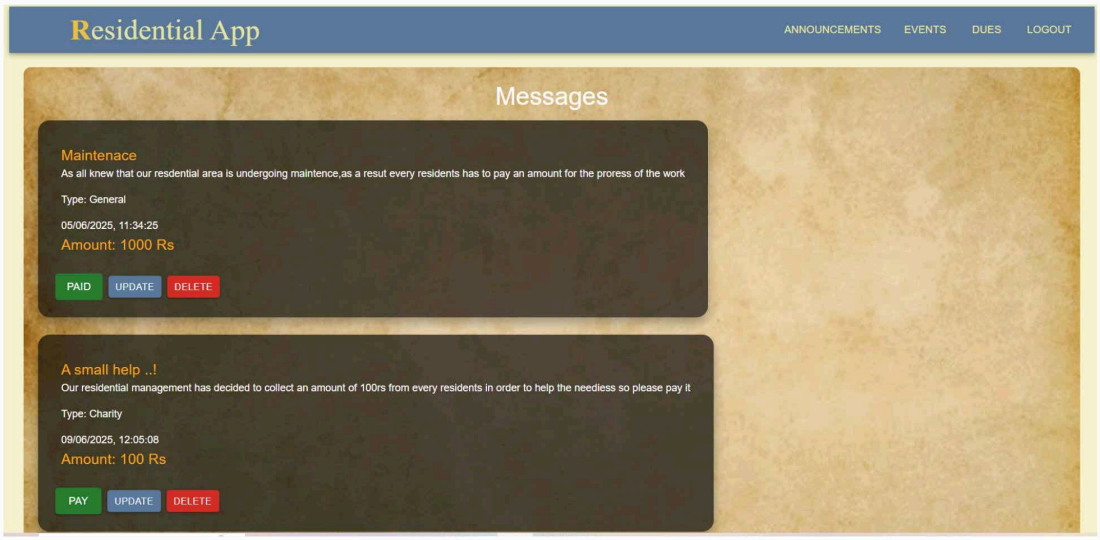
## e)Event Page



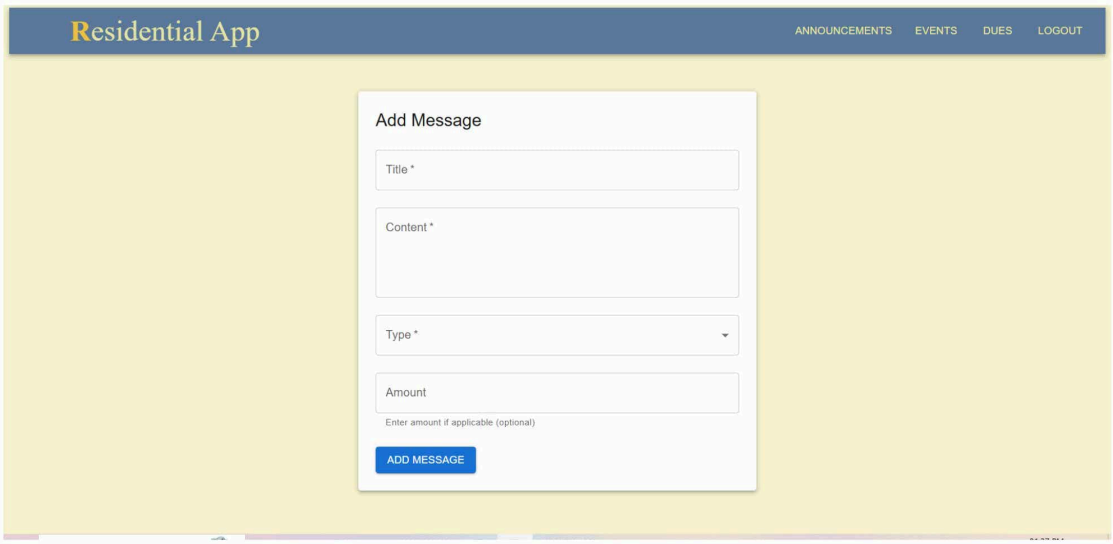

## f)Add event page



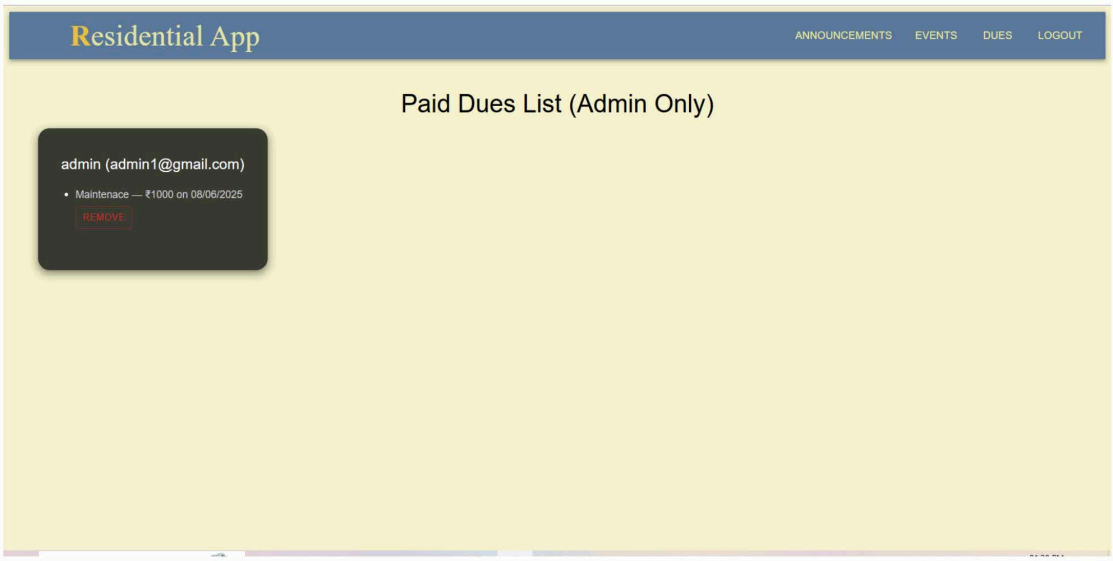
## g)Dues Page




## h)Add Dues Page



**i)Dues Paid List**




# 10.LOGIN CREDENTIALS

- Admin:
  - Username: admin1@gamil.com
  - Password: 984786
- User:
  - Username:user1@gmail.com
  - Password: 098765

Github repolink:https://github.com/Anupamaa1602/projectapp.git

AWS host link:http://13.126.215.181/
Video link:

https://drive.google.com/file/d/15fV4HGQctqNh38a7us55z6MV8IKxGL8r/view?usp=drive_link

## 11.MILESTONE AND REPORTING

| Milestone | Tasks | Reporting | Hrs |
|---|---|---|---|
| 1 - User Management | 1.1 - Design user schema, 1.2 - Implement registration/login,1.3 - Role-based access control | None | 30 |
| 2 - Announcement Module | 2.1 - Create announcement schema, 2.2 - Implement announcement creation, 2.3 - Display announcements, 2.4 - | Internal team meeting | 35 |
| 3 - Event Booking | 3.1 - Design event schema, 3.2 - Implement event creation, 3.3 - Event listing and booking, | Demo 1 | 45 |
| 4 - Payment Module | 4.1 - Integrate payment gateway (or mock) | None | 40 |
| 5 - Testing & Deployment | 5.1 - Unit testing, 5.2 - Integration testing, 5.3 - User acceptance testing, 5.4 - Deployment to staging, 5.5 - Final deployment | Demo 2 | 56 |