Floor Plan Management System

-by Dishant Gupta

Table of Contents

Table of Contents **2**
1. Introduction **3**
Overview: 3
2. Tech Stack **3**
Frontend: 3
Backend: 3
Database: 3
3. Features **3**
4. User Interface **4**
5. Database Structure **9**
6. Additional robust features **10**
1. Authentication: 10
2. Local Storage Mechanism for Offline Modifications: 10
3. Automatic Synchronization on Reconnection: 11
4. System Monitoring: 11
5. Error and Exception Handling: 12
9. Project Link **12**

1. Introduction

Overview:

The Intelligent Floor Plan Management System is a seamless workspace experience application designed for managing meeting rooms within a building. It allows administrators to manage room details, track seat capacity, and optimize meeting room usage. Users can easily book, modify, and free meeting rooms as per their requirements.

2. Tech Stack

Frontend:

React.js: User interface development.

Tailwind CSS: Styling and responsive design.

Backend:

Firebase: Real-time database for room and booking management firebase authentication for robust authentication.

Database:

Firebase Realtime Database for Rooms and MeetRecords

3. Features

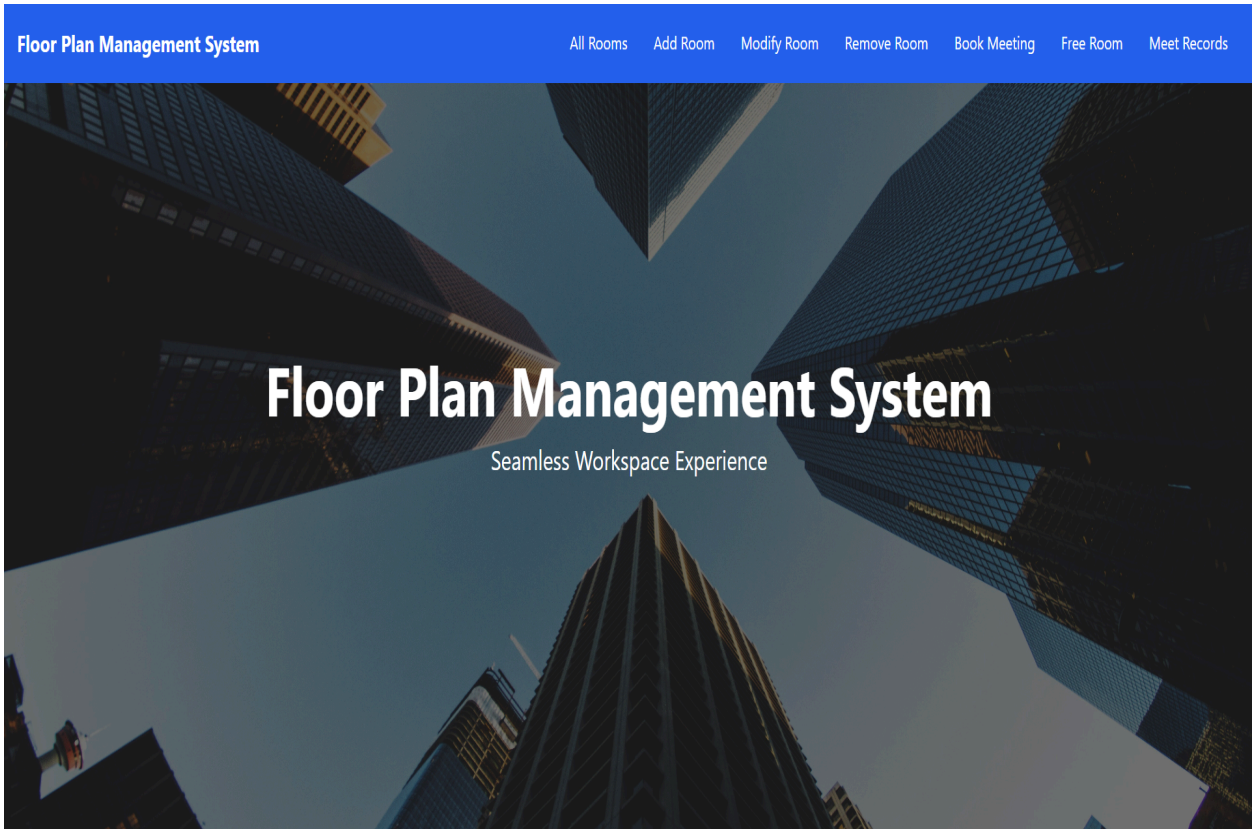
1. Add, modify, and delete room information (room number, floor number, capacity, etc.).
2. Manage the availability and occupancy of rooms.
3. Offline mechanism for admin access.
4. Book meeting rooms based on seat capacity and availability.
5. Suggestions for best-fit rooms based on proximity and capacity.
6. View and modify existing bookings.
7. Free occupied meeting rooms.

4. User Interface

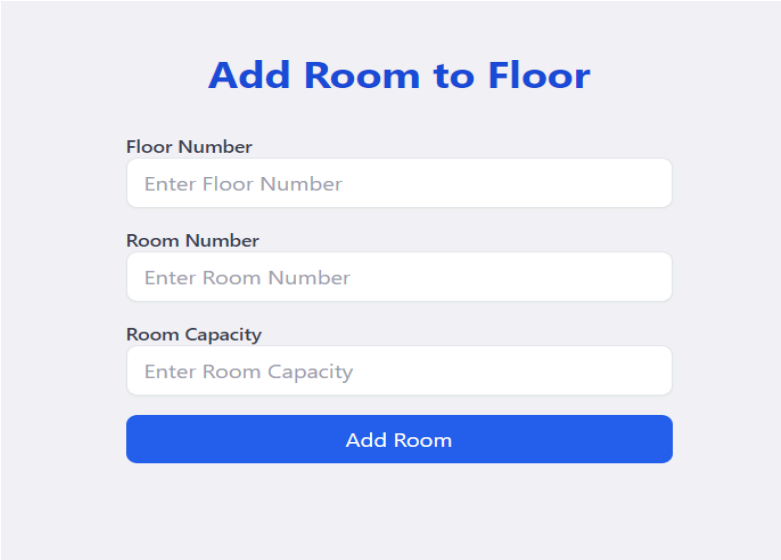
The system includes a responsive navigation bar where following features can be accessed:

- Home Page
- View All Room
- Add Room
- Modify Room
- Delete Room
- Book Meeting
- Free Occupied Rooms
- View Meeting Records

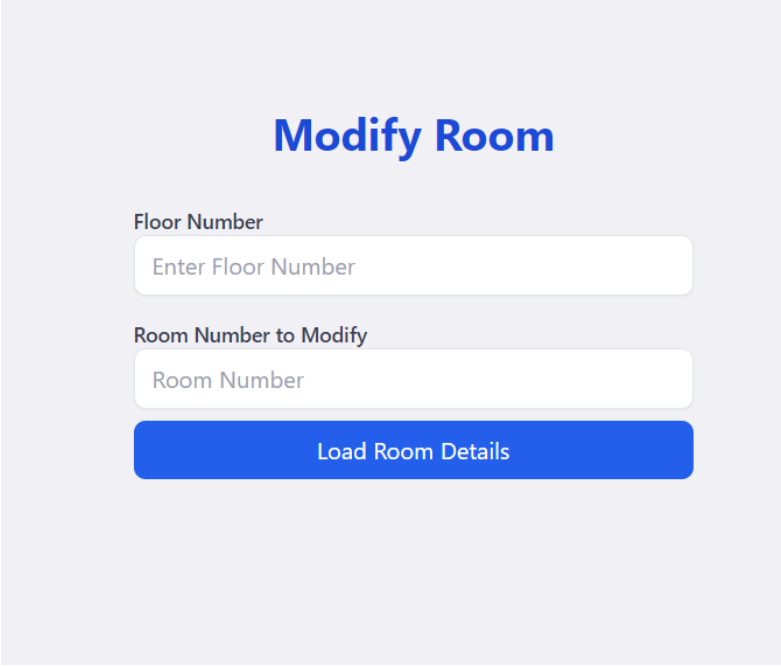
Home Page:



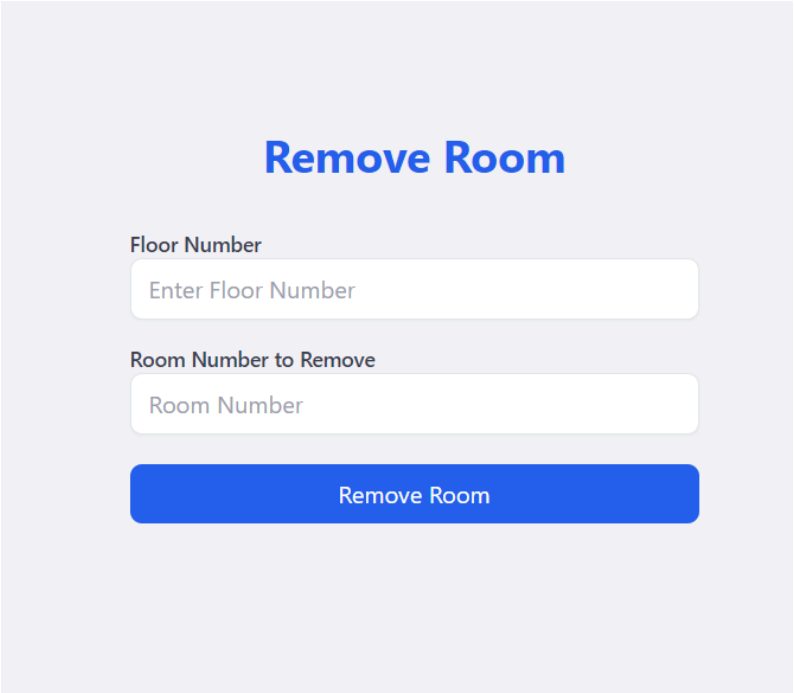
Add Room : Admin can add the details of new rooms to add rooms on a floor.



Modify Room: Admin can change the details of existing rooms, such as increasing/decreasing seat capacity, changing the floor number, or adjusting room details if it is not currently occupied for a meeting.



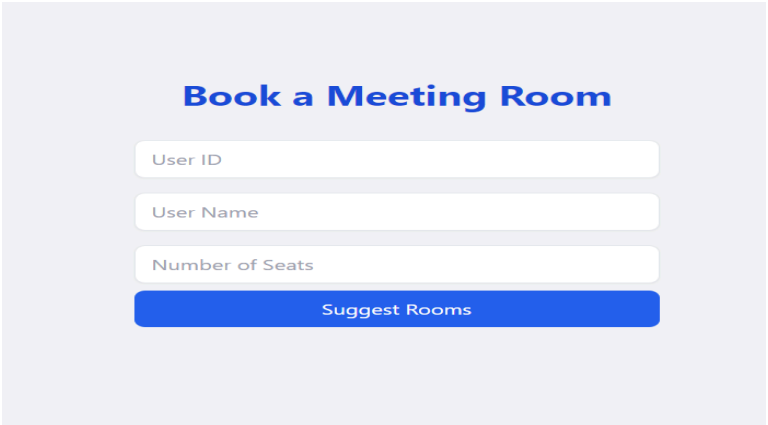
Remove Room: Admin can remove the details of existing rooms if it is not currently occupied for a meeting.



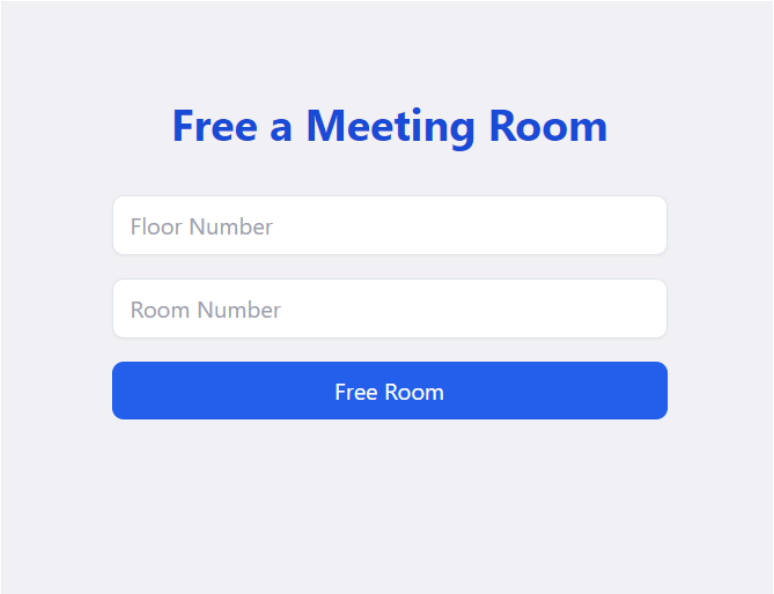
Book Meeting: Admin can book a room on user request by specifying the number of required seats. The system suggests the best-fit rooms based on:

1. Proximity (same or nearest floor which meet seat requirements and not occupied for other meetings).
2. Capacity (rooms that best fit the seat requirements and not occupied for other meetings no matter what is the location of the floor).

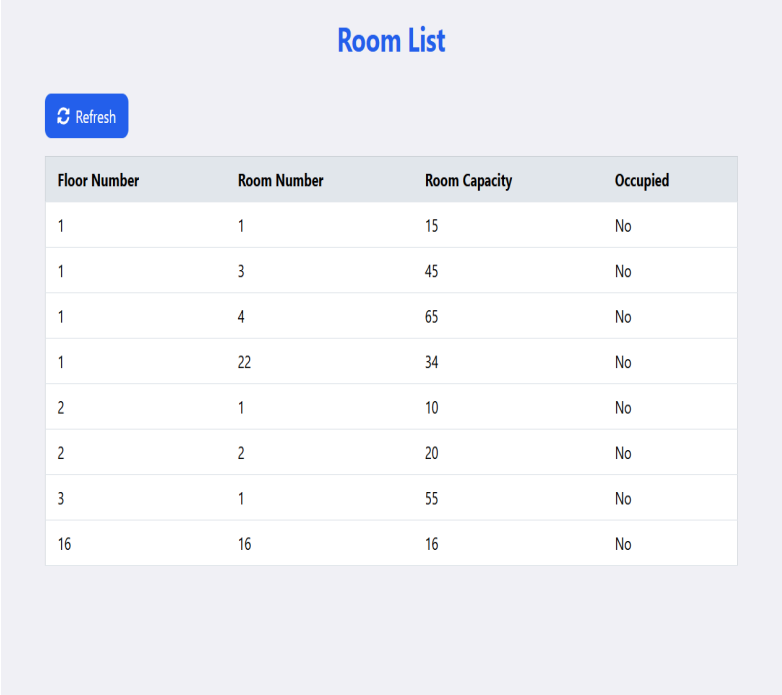
Once a room is booked, the occupancy status is updated and a record is saved in the MeetRecord database.



Free Room: Admin is allowed to free an occupied room by specifying the room and floor number.

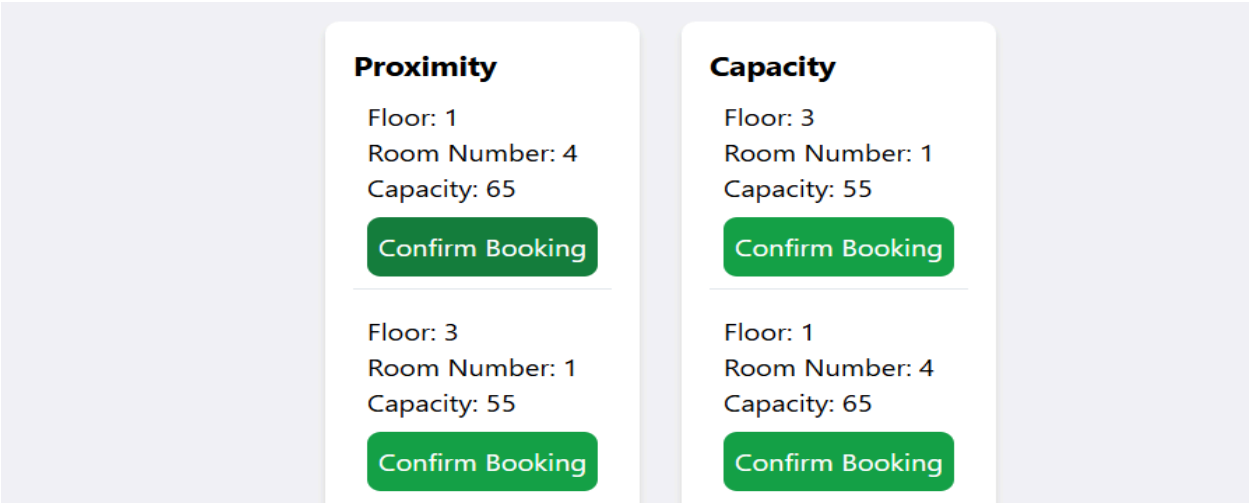


View All Rooms: View all previous meeting records.

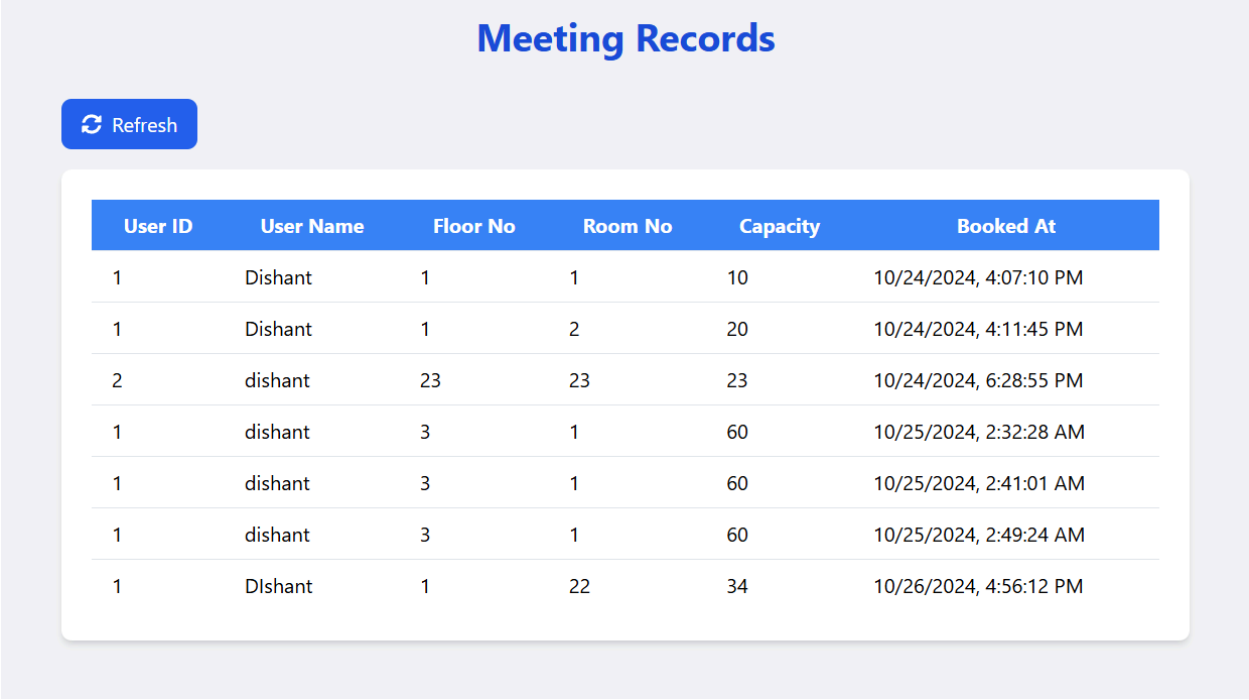


Floor Number	Room Number	Room Capacity	Occupied
1	1	15	No
1	3	45	No
1	4	65	No
1	22	34	No
2	1	10	No
2	2	20	No
3	1	55	No
16	16	16	No

View Suggestions:



View Meeting Records:

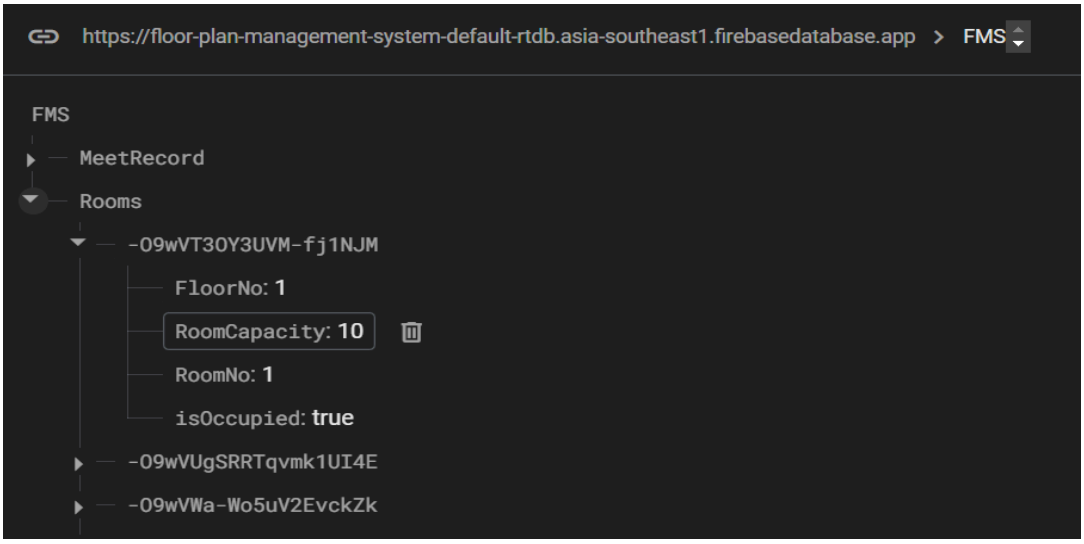


User ID	User Name	Floor No	Room No	Capacity	Booked At
1	Dishant	1	1	10	10/24/2024, 4:07:10 PM
1	Dishant	1	2	20	10/24/2024, 4:11:45 PM
2	dishant	23	23	23	10/24/2024, 6:28:55 PM
1	dishant	3	1	60	10/25/2024, 2:32:28 AM
1	dishant	3	1	60	10/25/2024, 2:41:01 AM
1	dishant	3	1	60	10/25/2024, 2:49:24 AM
1	DIshant	1	22	34	10/26/2024, 4:56:12 PM

5. Database Structure

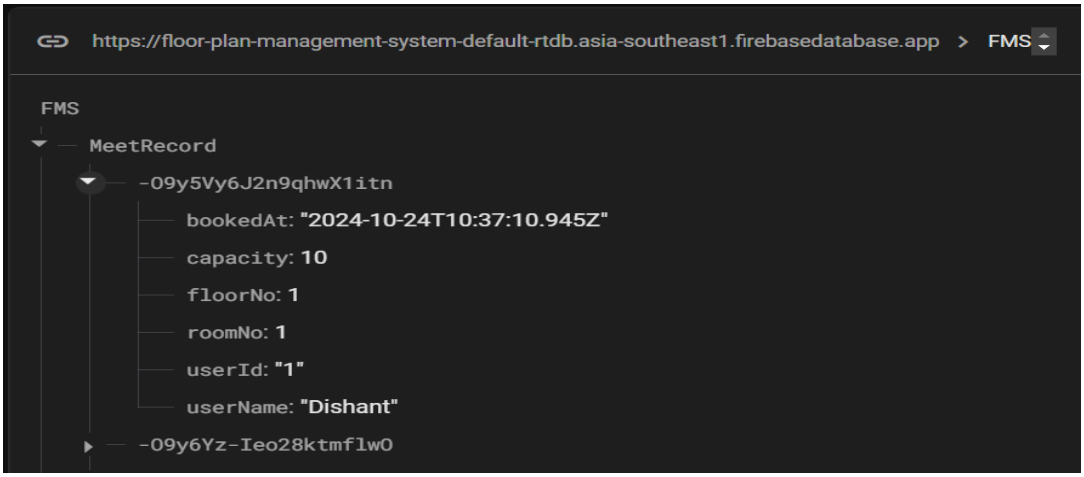
Rooms: Stores room data under FMS/Rooms.

1. RoomNo: Integer representing room number.
2. FloorNo: Integer representing the floor where the room is located.
3. RoomCapacity: Integer representing seat capacity.
4. isOccupied: Boolean representing room availability.



MeetRecord: Stores meeting bookings under FMS/MeetRecord.

1. userId: ID of the user who booked the meeting.
2. userName: Name of the user.
3. roomNo: Booked room number.
4. floorNo: Booked floor number.
5. bookedAt: Time of the booking.

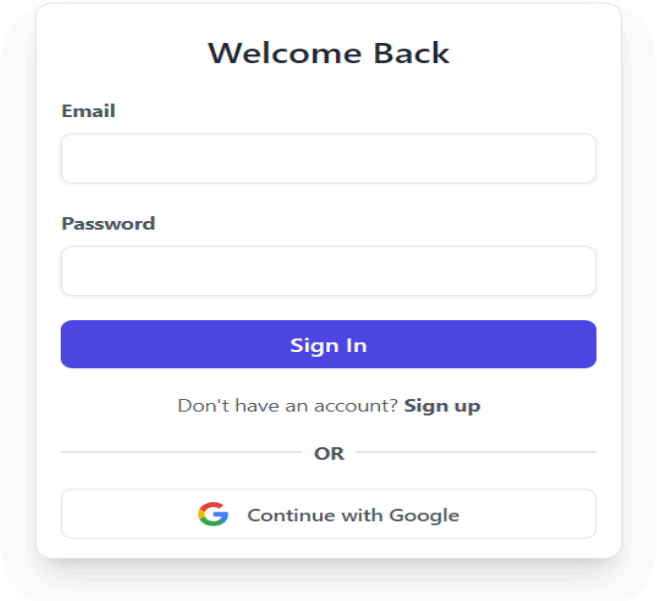


6. Additional robust features

1. Authentication:

The system includes authentication where:

- Admin must log in or register to access any functionality.
- Admin privileges allow additional room management options.
- Firebase Authentication is used for secure login and registration.



2. Local Storage Mechanism for Offline Modifications:

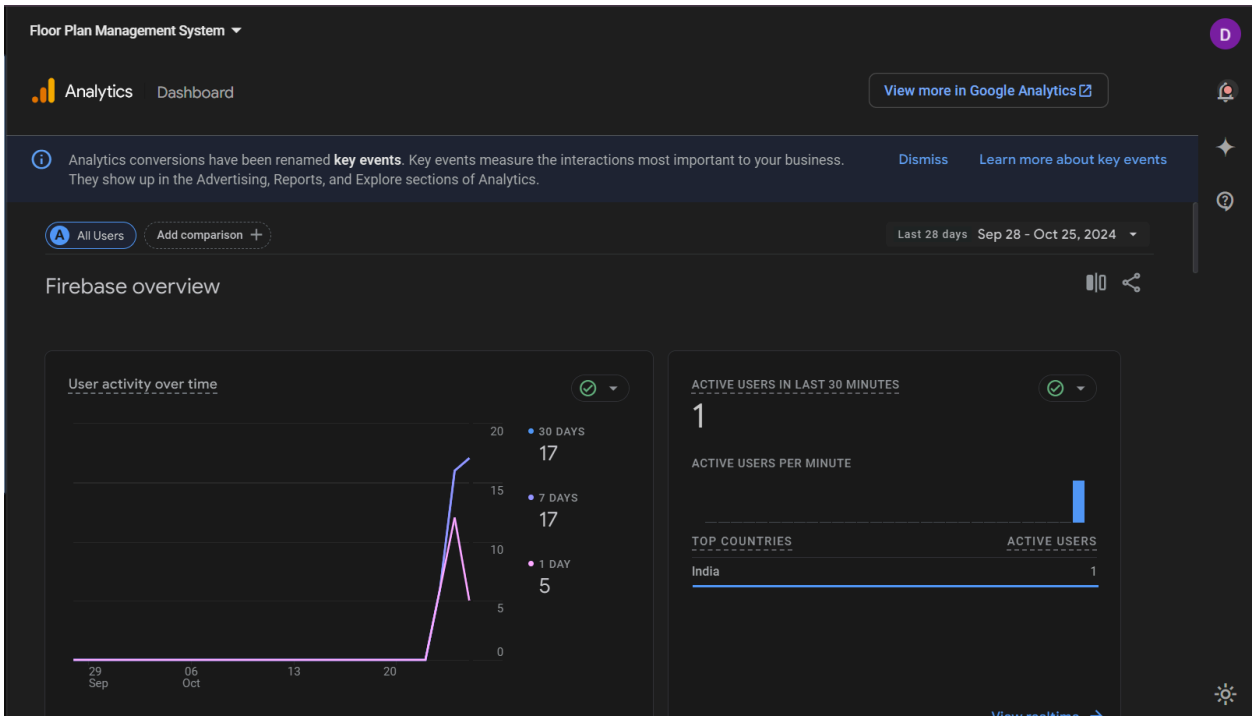
The system allows admins to make room modifications even when offline. Any changes made while offline are stored locally on the admin's device to ensure uninterrupted functionality. This enables efficient data handling, letting admins perform critical updates without needing immediate server access.

3. *Automatic Synchronization on Reconnection:*

Once the system detects an active internet connection, it automatically synchronizes offline changes with the server, updating the main database. This ensures all room modifications are saved to the cloud, keeping records accurate and up-to-date without requiring additional admin actions.

4. *System Monitoring:*

Firebase provides analytical tools for system monitoring.



5. ***Error and Exception Handling:***

   The system features a robust error-handling framework that provides detailed error messages for easier debugging.

7. Links Related to Project

[Live Link]
[Github]
[Demo Video]
[Document]