



National Institute of Technology, Silchar

***CS 312: Database & Management System
(DBMS Project)***

Apartment Management System

Prepared by: -

Mwkhangsa Daimari (2112140)

Ankita Dutta (2112141)

Banothu Manjula (2112142)

Koushik Reddy Kandula (2112143)

Kumar Ashish Ranjan (2112144)

Title: *Apartment Management system*

Problem Statement:

An Apartment Management System (AMS) is a critical tool designed to alleviate the challenges of manual data entry in residential complexes. This comprehensive software solution, developed as part of a Database Management System Course, incorporates a user interface and MySQL database support. The primary objective is to streamline apartment management processes for administrators, owners, tenants, and employees.

The management of residential complexes is an intricate task that demands efficient processes to overcome challenges associated with manual data entry. As part of a Database Management System Course, our project endeavors to create a robust Apartment Management System (AMS) that seamlessly integrates a user interface and MySQL database support. The primary goal is to optimize apartment management processes for administrators, owners, tenants, and employees.

Background:

Apartments, regardless of their scale, face challenges in manual data entry and management. This project addresses these challenges by integrating a user-friendly interface and storing data efficiently in a MySQL database. The system includes functionalities for admins, owners, tenants, and employees, ensuring efficient management and interaction within the residential complex.

Project Scope:

The project's primary goal is to design and implement a relational database schema for apartment management system, incorporating tables for auth, block, block_admin, employee, identity, owner, rental, room and tenant. The system will ensure data integrity through foreign key constraints and maintain relationships between these entities.

Approach of Solving:

The approach to solving the problem of creating a apartment management system involves several structured steps:

1. *Requirements Gathering*: The initial phase of our project involves a comprehensive process of requirements gathering, aimed at grasping the specific needs of the residential complex for the development of an efficient Apartment Management System(AMS). The focus is on identifying and defining essential functionalities required for seamless apartment management.
2. *Design and Planning*: Creating a detailed plan by visualizing the system architecture, user interface, database schema, and system components. Deciding on the technology stack, security measures and integration.
3. *Database Design*: Developing an efficient database structure considering entities like auth, block, block_admin, employee, identity, owner, rental, room and tenant and relationships between them. Ensuring data integrity and efficient storage.
4. *Development*: Implementing the system using appropriate programming languages and frameworks for the frontend, backend, and database components. Writing code, following best practices, and maintaining modularity for scalability.

5. Iterative Improvement: Gathering feedback post-implementation to identify areas for enhancement. Continuously updating and improving the system to adapt to changing business needs and technology advancements.

This structured approach ensures a systematic development process, enabling the creation of a robust and tailored apartment management system that fulfills the apartment's operational requirements and provides an efficient user experience.

Technology Stack:

1. *Database:* Utilizes a relational database (e.g., **MySQL** and **SQL** language) to store and manage data.
2. *Backend:* Implements server-side logic using frameworks like **Node.js**, **Express.js**.
3. *Frontend:* Utilizes web technologies like frameworks (**React.Js**) and (**HTML**, **CSS**, **JS**, **Bootstrap**) for a user-friendly interface.
4. *Security Measures:* Implements encryption, secure authentication, and authorization protocols to protect sensitive data.

About Database:

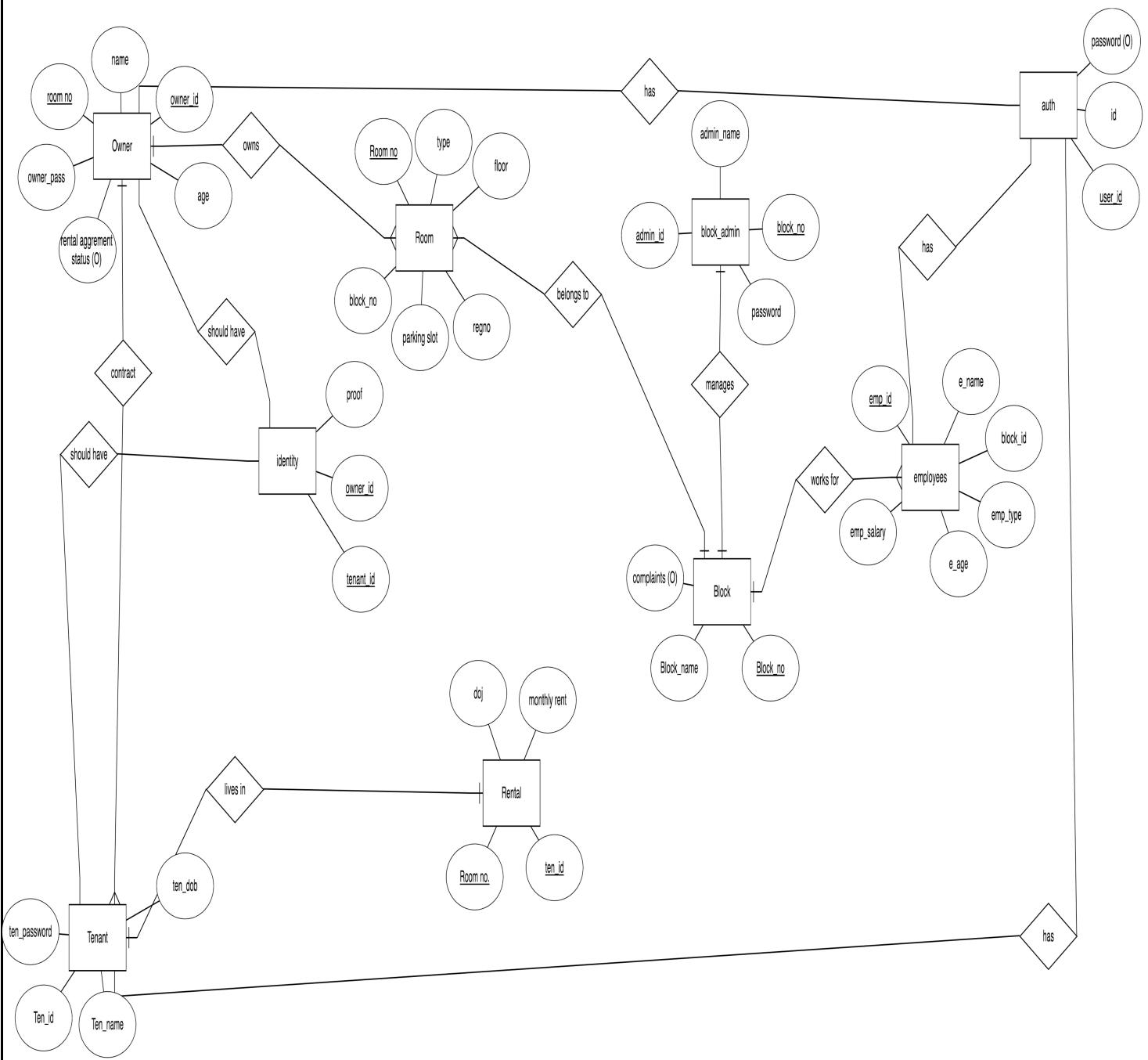
Entities-Attributes:

- Owner - name, owner_id, age, owner_pass, room no, rental agreement status
- Room - room_no, type, floor, region, parking slot, block_no
- Block_admin - admin_name, block_no, admin_id, password
- Auth - password, id, user_id
- Employees - e_name, block_id, emp_type, e_age, emp_salary, emp_id
- Block - block_no, block_name, complaints
- Rental - monthly rent, doj, room_no, ten_id
- Identity - proof, owner_id, tenant_id
- Tenant - ten_password, ten_id, ten_name, ten_dob

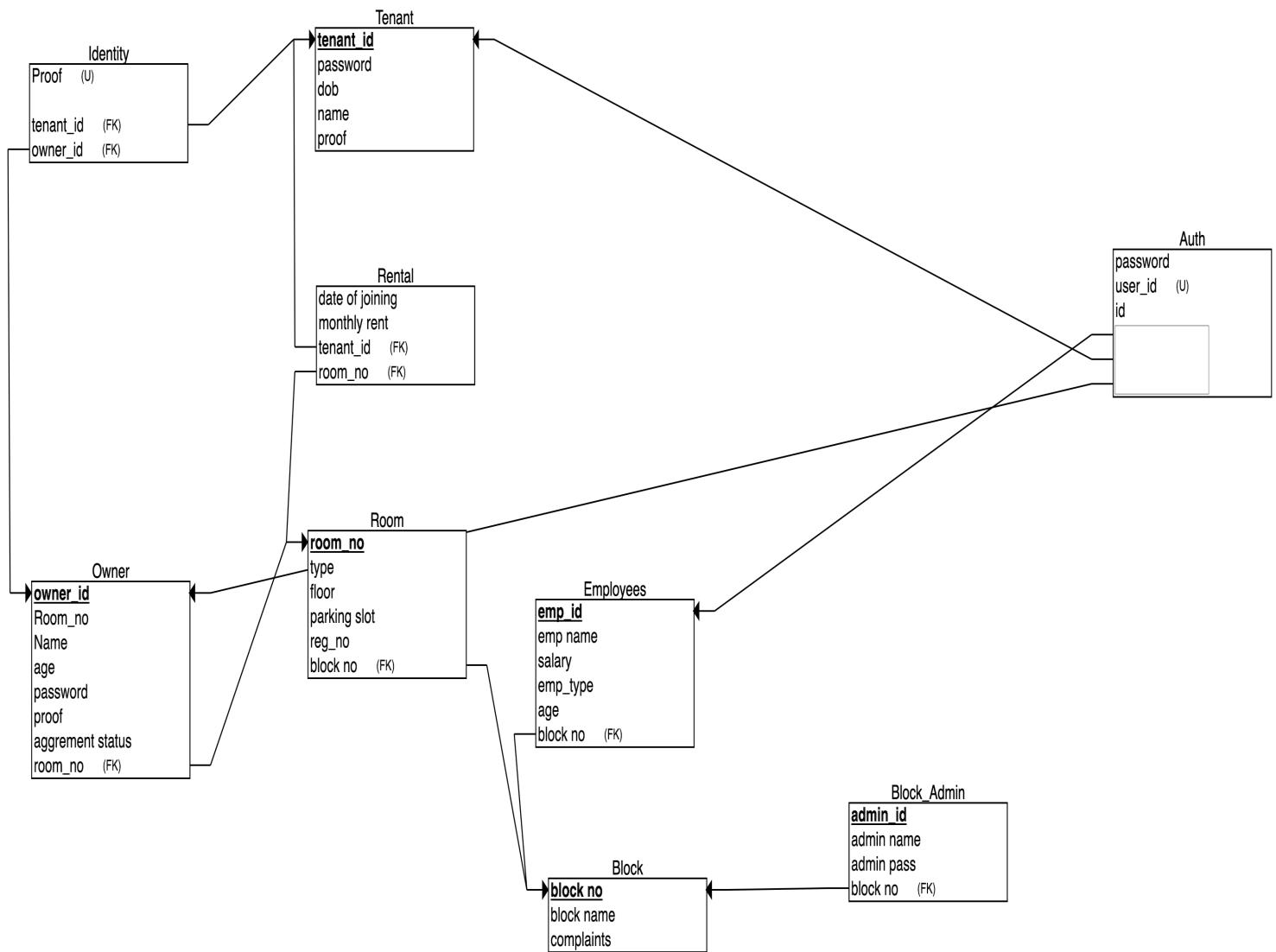
Relationships:

- Owner - Room
- Owner - Identity
- Owner - Auth
- Owner - Tenant
- Tenant - Identity
- Tenant - Rental
- Tenant - Auth
- Room - Block
- Block_admin - Block
- Block - Employees
- Employees - Auth

ER DIAGRAM



Schema Diagram



Tables and Schemas:

- auth:

| Field | Type | Null | Key | Default | Extra |
|----------|-------------|------|-----|----------|-------|
| user_id | varchar(10) | NO | PRI | NULL | |
| password | varchar(20) | NO | | 12345678 | |
| id | int | NO | UNI | NULL | |

- block:

| Field | Type | Null | Key | Default | Extra |
|------------|--------------|------|-----|---------|-------|
| block_no | int | NO | PRI | NULL | |
| block_name | varchar(10) | YES | | NULL | |
| complaints | varchar(100) | YES | | NULL | |
| room_no | int | YES | MUL | NULL | |

- block_admin:

| Field | Type | Null | Key | Default | Extra |
|------------|-------------|------|-----|---------|-------|
| admin_id | int | NO | PRI | NULL | |
| admin_name | varchar(20) | YES | | NULL | |
| block_no | int | YES | MUL | NULL | |

- employee:

| Field | Type | Null | Key | Default | Extra |
|----------|-------------|------|-----|---------|-------|
| emp_id | int | NO | PRI | NULL | |
| emp_name | varchar(30) | YES | | NULL | |
| salary | int | YES | | NULL | |
| emp_type | varchar(20) | YES | | NULL | |
| age | int | YES | | NULL | |
| block_no | int | YES | MUL | NULL | |

- Identity:

| Field | Type | Null | Key | Default | Extra |
|-----------|-------------|------|-----|---------|-------|
| proof | varchar(15) | YES | UNI | NULL | |
| owner_id | int | YES | MUL | NULL | |
| tenant_id | int | YES | MUL | NULL | |

- owner:

| Field | Type | Null | Key | Default | Extra |
|------------------|-------------|------|-----|---------|-------|
| owner_id | int | NO | PRI | NULL | |
| name | varchar(20) | YES | | NULL | |
| age | int | YES | | NULL | |
| aggrement_status | varchar(20) | NO | | NULL | |
| room_no | int | YES | MUL | NULL | |
| dob | varchar(15) | YES | | NULL | |

- rental:

| Field | Type | Null | Key | Default | Extra |
|--------------|-------------|------|-----|---------|-------|
| doj | varchar(20) | YES | | NULL | |
| monthly_rent | int | YES | | NULL | |
| room_no | int | YES | MUL | NULL | |
| tenant_id | int | YES | MUL | NULL | |

- room:

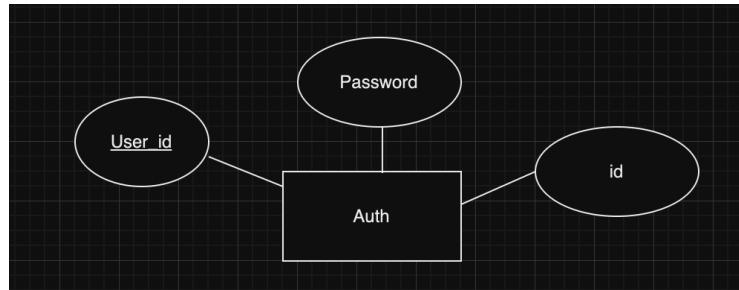
| Field | Type | Null | Key | Default | Extra |
|--------------|-------------|------|-----|---------|-------|
| room_no | int | NO | PRI | NULL | |
| type | varchar(10) | YES | | NULL | |
| floor | int | YES | | NULL | |
| parking_slot | varchar(10) | YES | UNI | NULL | |
| reg_no | int | YES | UNI | NULL | |
| block_no | int | YES | MUL | NULL | |

- Tenant:

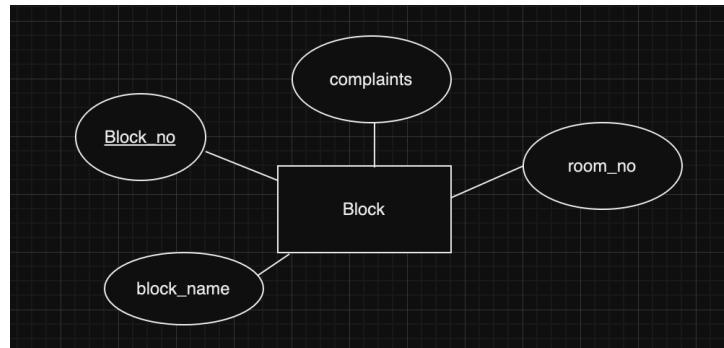
| Field | Type | Null | Key | Default | Extra |
|-----------|-------------|------|-----|---------|-------|
| tenant_id | int | NO | PRI | NULL | |
| name | varchar(30) | YES | | NULL | |
| dob | varchar(10) | YES | | NULL | |
| stat | varchar(10) | YES | | NULL | |
| room_no | int | YES | MUL | NULL | |
| age | int | YES | | NULL | |

Entity Sets:

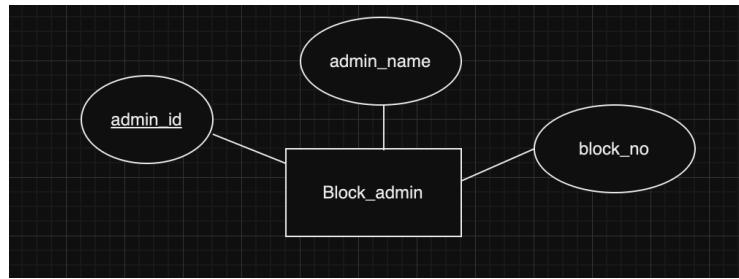
- Auth:



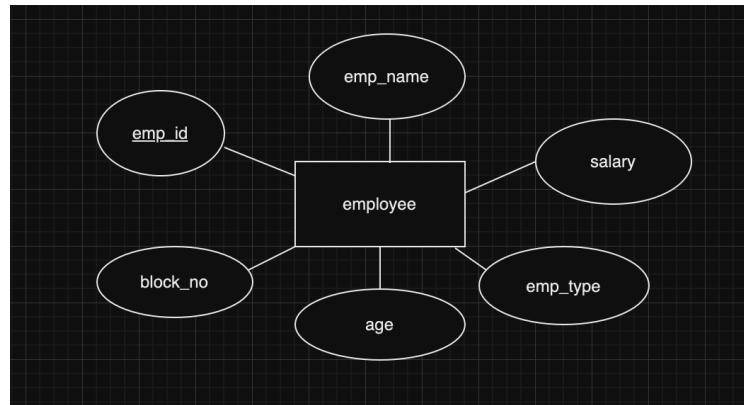
- Block:



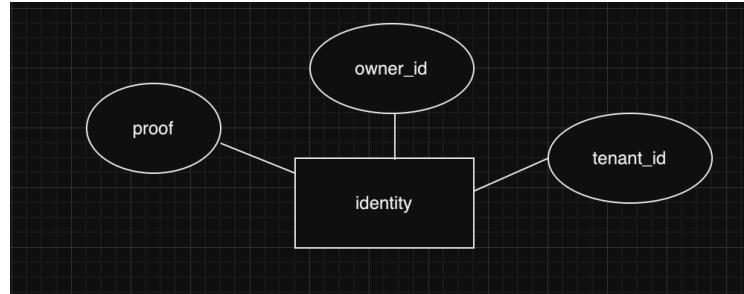
- Block_admin:



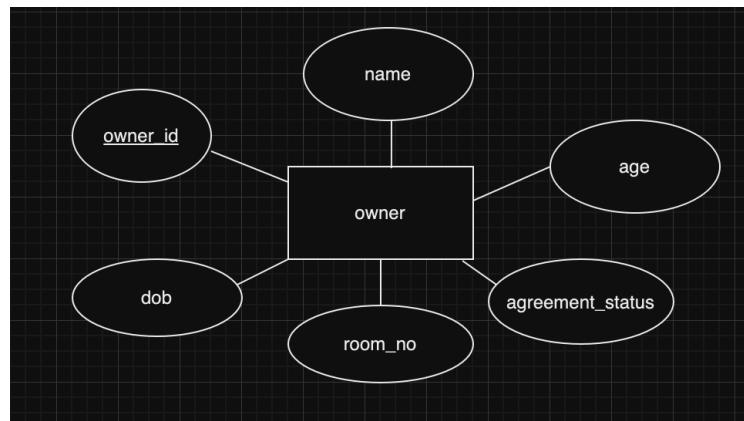
- Employee:



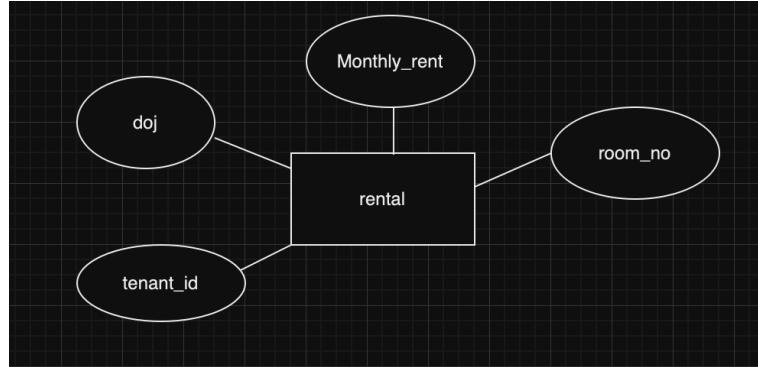
- Identity:



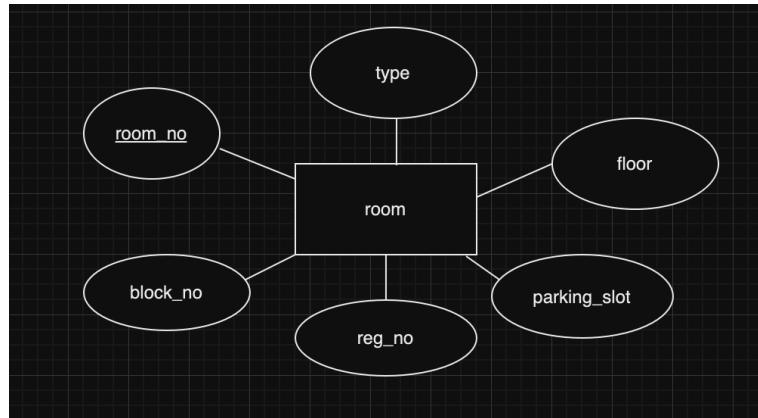
- Owner:



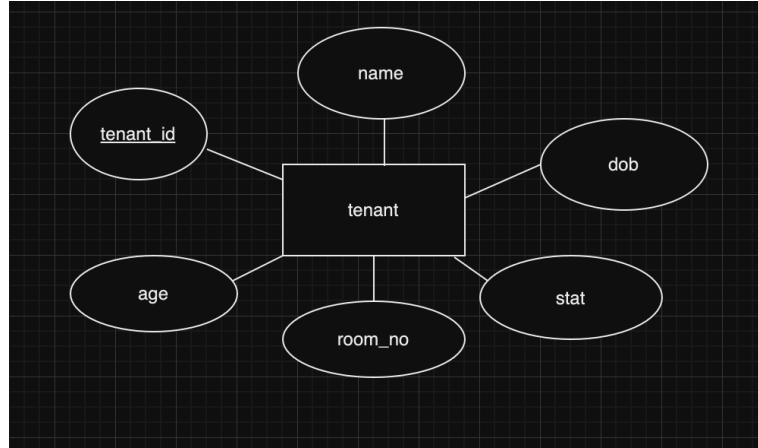
- Rental:



- Room:

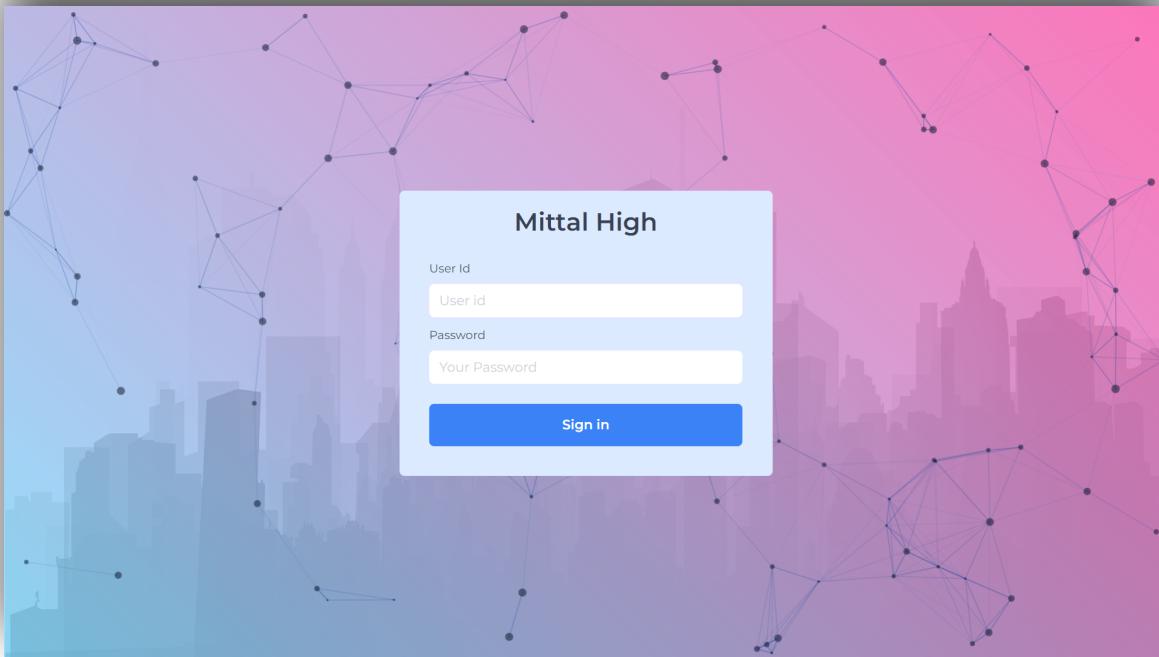


- Tenant:



User Interface:

- Login:



- Admin:

A screenshot of the Mittal High admin dashboard. The top navigation bar includes the logo, the building name, and a "Logout" link. On the left, a dark sidebar lists administrative functions: Home, Tenant Details, Owner Details, Create owner, Allotting Parking slot, and Complaints. The main content area features three summary boxes: "11 Total Owner", "6 Total Tenant", and "3 Total Employee". Below these is a section titled "Apartment Rules and Regulation" containing a list of rules. The rules are:

- Tenant shall keep premises in good condition.
- Tenant shall not interfere with other tenant's premises.
- Tenant shall pay rent promptly on the due date.
- Tenant shall not make any alterations to the premises without written permission of the landlord.
- Tenant shall keep proper liability, fire and/or other damage insurance on the contents of the premises leased.
- Tenants shall not receive a refund of the damage deposit until landlord is certain that the premises are free of damages upon the surrender of the premises.
- No tenant shall interfere in any manner with any portion either of the heating or lighting or other apparatus in or about the building.
- Automobiles must be kept within yellow lines of the parking lot areas.
- Sanitary napkins shall not be deposited in toilets but shall be wrapped and deposited with other waste matter and refuse.
- Tenant shall be responsible for closing of windows in his or her apartment during storms.

- Owner:

The screenshot shows the 'Mital High' owner dashboard. The left sidebar includes links for Home, Tenant details, Complaint, Create Tenant, and Room Details. Two summary boxes are at the top: 'No Of Employees' (3) and 'Total Complaints' (3). A large central box displays the 'Apartment Rules and Regulation' with a list of 16 rules.

Apartment Rules and Regulation

- Tenant shall keep premises in good condition.
- Tenant shall not interfere with other tenant's premises.
- Tenant shall pay rent promptly on the due date.
- Tenant shall not make any alterations to the premises without written permission of the landlord.
- Tenant shall keep proper liability, fire and/or other damage insurance on the contents of the premises leased.
- Tenants shall not receive a refund of the damage deposit until landlord is certain that the premises are free of damages upon the surrender of the premises.
- No tenant shall interfere in any manner with any portion either of the heating or lighting or other apparatus in or about the building.
- Automobiles must be kept within yellow lines of the parking lot areas.
- Sanitary napkins shall not be deposited in toilets but shall be wrapped and deposited with other waste matter and refuse.
- Tenant shall be responsible for closing of windows in his or her apartment during storms.

- Employee:

The screenshot shows the 'Mital High' employee dashboard. The left sidebar includes links for Home and Complaints. Two summary boxes are at the top: 'Total Complaints' (3) and 'Salary' (Rs. 20000). A large central box displays the 'Apartment Rules and Regulation' with a list of 16 rules.

Apartment Rules and Regulation

- Tenant shall keep premises in good condition.
- Tenant shall not interfere with other tenant's premises.
- Tenant shall pay rent promptly on the due date.
- Tenant shall not make any alterations to the premises without written permission of the landlord.
- Tenant shall keep proper liability, fire and/or other damage insurance on the contents of the premises leased.
- Tenants shall not receive a refund of the damage deposit until landlord is certain that the premises are free of damages upon the surrender of the premises.
- No tenant shall interfere in any manner with any portion either of the heating or lighting or other apparatus in or about the building.
- Automobiles must be kept within yellow lines of the parking lot areas.
- Sanitary napkins shall not be deposited in toilets but shall be wrapped and deposited with other waste matter and refuse.
- Tenant shall be responsible for closing of windows in his or her apartment during storms.

- Tenant:

The screenshot shows a web-based tenant portal for 'Mittal High'. The top navigation bar includes a logo, the site name 'Mittal High', and a 'Logout' link. On the left, a sidebar lists menu items: 'Home', 'Raising Complaints', 'Allotted Parking slot', and 'Pay maintenance'. The main content area displays four data cards: '601 Tenant Id', 'nithin Tenant Name', '19 Tenant Age', and '01-apr-02 Dob'. Below these cards is a section titled 'Apartment Rules and Regulation' containing a list of rules.

Apartment Rules and Regulation

- Tenant shall keep premises in good condition.
- Tenant shall not interfere with other tenant's premises.
- Tenant shall pay rent promptly on the due date.
- Tenant shall not make any alterations to the premises without written permission of the landlord.
- Tenant shall keep proper liability, fire and/or other damage insurance on the contents of the premises leased.
- Tenants shall not receive a refund of the damage deposit until landlord is certain that the premises are free of damages upon the surrender of the premises.
- No tenant shall interfere in any manner with any portion either of the heating or lighting or other apparatus in or about the building.
- Automobiles must be kept within yellow lines of the parking lot areas.
- Sanitary napkins shall not be deposited in toilets but shall be wrapped and deposited with other waste matter and refuse.
- Tenant shall be responsible for closing of windows in his or her apartment during storms.

References:

- W3schools - <https://www.w3schools.com>
- Geekforgeeks - <https://www.geeksforgeeks.org>
- Tutorialspoint - <https://www.tutorialspoint.com/index.htm>
- Stack OverFlow - <https://stackoverflow.com>