



NITTE
EDUCATION TRUST

**NMAM INSTITUTE
OF TECHNOLOGY**

A MINI PROJECT REPORT ON
Submitted in partial fulfilment requirements for the credit
of
the Course on Database Management System (CS2102-1)
BY

CHAITHRA S NAYAK NNM22CC011

THRISHA J SHETTY NNM22CC064

Under the Guidance of
Dr. Chithra K
Assistant Professor Grade III
DEPARTMENT
OF
COMPUTER AND COMMUNICATION ENGINEERING
2023-2024

Department of Computer and Communication Engineering

CERTIFICATE

This is to certify that **CHAITHRA S NAYAK (NNM22CC011)** and **THRISHA J SHETTY (NNM22CC064)** have successfully completed the mini project work on '**Apartment Management System**' and submitted in partial fulfillment of the requirements of the Course on **Database Management System(CS2102-1)** prescribed by the **NMAMIT, Nitte** during the academic year **2023-2024**.

Course Instructor
Dr. Chithra K
Assistant Professor

H.O. D
Dr. Radhakrishna
Associate Professor

Name of the Examiners

Signature with Date

- 1.
- 2.

TABLE OF CONTENTS

CONTENTS	PAGE NO.
1. INTRODUCTION	4-5
1.1 Introduction to DBMS	4
1.2 Mini Project Description	4-5
2. REQUIREMENT SPECIFICATION	6
2.1 Software Requirements	6
3. DESIGN	7
3.1 ER Diagram	7
4. IMPLEMENTATION DETAILS	8-15
4.3 SQL Code with proper comment lines	8-15
5. RESULTS	16-19
5.1 Sample Output (Screenshots)	16-19
6. CONCLUSION	20
7. REFERENCES	21

CHAPTER 1

INTRODUCTION

1.1 Introduction to DBMS:

A Database Management System (DBMS) is a software solution designed to facilitate efficient and secure interaction with databases. It enables users to store, retrieve, update, and manage data effectively. Serving as a mediator between the database and users or applications, a DBMS ensures smooth data operations and enhances overall data management processes.

1.2 Mini Project Description

The Apartment Management System (AMS) is a comprehensive solution designed to streamline operations within residential complexes by alleviating the challenges associated with manual data entry. Leveraging modern technologies and developed as part of a Database Management System Course, this software aims to optimize apartment management processes for administrators, owners and tenants.

Technological Framework: Utilizes React.js for the frontend to provide a dynamic and responsive user interface, while Node.js powers the backend for robust logic and data processing capabilities.

Administrator Functionalities: Includes secure login authentication, access to tenant and owner details for efficient management, creation of new owners and allocation of parking slots, monitoring and resolution of complaints, and oversight of total owners and tenants.

Owner Features: Provides access to tenant details for their respective owned rooms, ability to create and manage tenants, monitoring complaints related to their owned rooms, and viewing room details and total complaints.

Tenant Capabilities: Offers access to personal information such as tenant ID, name, age, date of birth, and room number, visibility into allocated parking slots, and seamless complaint submission.

Residential complexes often face challenges in manual data entry and management. The AMS project addresses these challenges by integrating a user-

friendly interface and efficiently storing data in a MySQL database. Functionalities for admins, owners, and tenants ensure efficient management and interaction within the residential complex.

The primary goal of the project is to design and implement a relational database schema for the apartment management system. This includes incorporating tables for authentication, blocks, block administrators, employees, identity, owners, rentals, rooms, and tenants. The system ensures data integrity through foreign key constraints and maintains relationships between these entities.

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 Software Requirements

Frontend:

React JS: A JavaScript library for building user interfaces, known for its efficiency and flexibility.

Tailwind CSS: A utility-first CSS framework for quickly building custom designs.

Backend:

Node.js: A JavaScript runtime environment enabling execution of JavaScript code outside the browser, facilitating server-side application development and real-time applications.

Express.js: A JavaScript framework that simplifies the development of web applications and APIs in Node.js by offering a minimalist and flexible approach, along with powerful features and middleware.

Database:

MySQL: An open-source relational database management system, widely used for storing and managing data in various types of applications.

Other Tools Used:

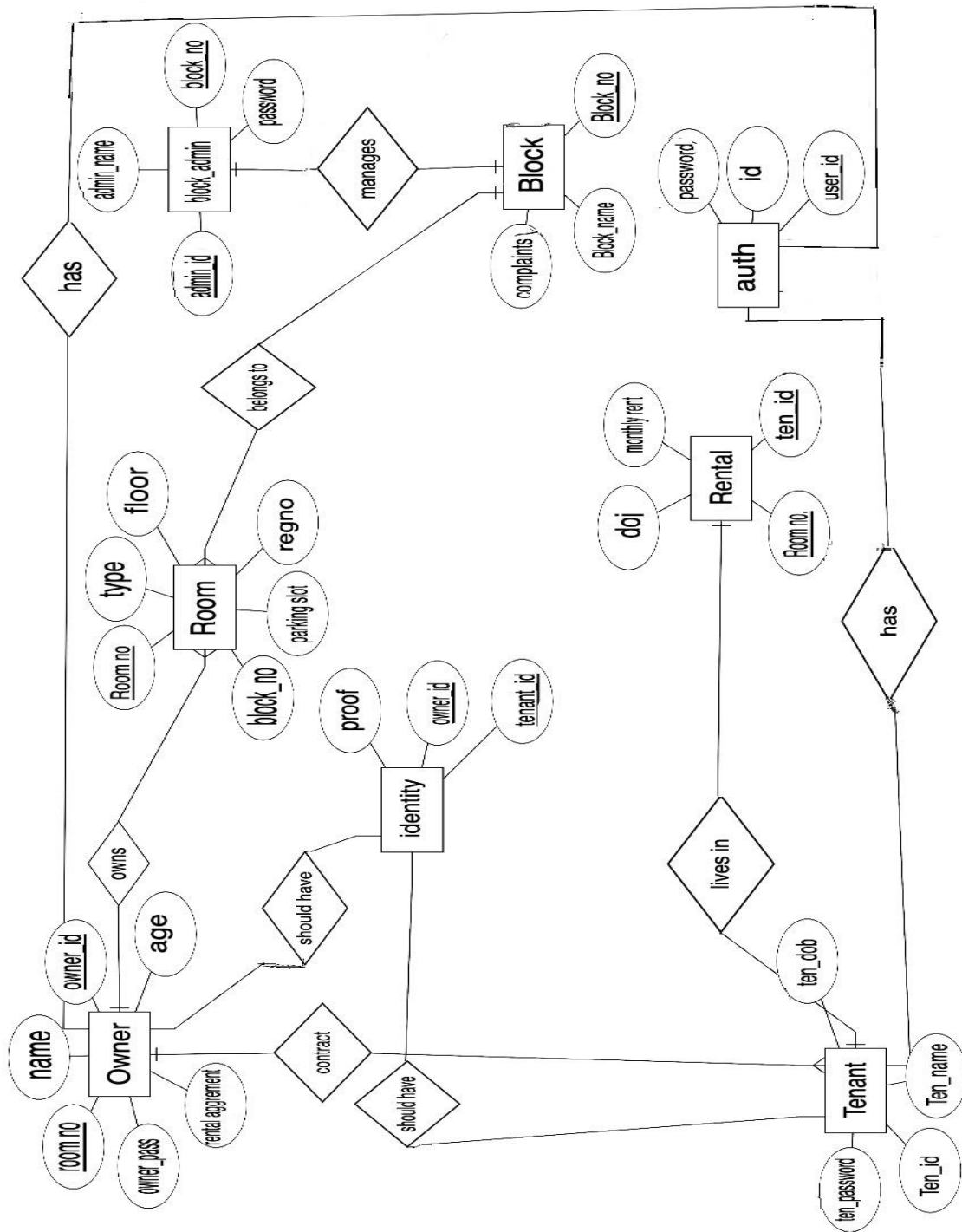
Visual Studio Code: A lightweight, powerful code editor with built-in support for JavaScript, Node.js, and many other programming languages.

MySQL Workbench 8.0 CE: A unified visual tool for database architects, developers, and DBAs, providing data modeling, SQL development, and comprehensive administration tools for MySQL databases.

CHAPTER 3

DESIGN

3.1 E-R diagram



CHAPTER 4

IMPLEMENTATION DETAILS

4.1 SQL code with proper comment lines

```
-- Set initial configurations
/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

-- Table structure for authentication
DROP TABLE IF EXISTS `auth`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `auth` (
  `user_id` varchar(10) NOT NULL,
  `password` varchar(20) NOT NULL DEFAULT '12345678',
  `id` int NOT NULL,
  PRIMARY KEY (`user_id`),
  UNIQUE KEY `id` (`id`)
```



```

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
-- Insert data into authentication table
LOCK TABLES `auth` WRITE;
/*!40000 ALTER TABLE `auth` DISABLE KEYS */;
INSERT INTO `auth` VALUES ('a-123','12345678',101),('a-124','12345678',102),('a-
909','12345678',103),('o-123','12345678',501),('o-124','12345678',502),('o-
456','12345678',503),('o-909','12345678',504),('t-123','12345678',601),('t-
124','12345678',602),('t-145','12345678',603),('t-190','12345678',604),('t-
345','12345678',605);
/*!40000 ALTER TABLE `auth` ENABLE KEYS */;
UNLOCK TABLES;

-- Table structure for block details
DROP TABLE IF EXISTS `block`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `block` (
  `block_no` int NOT NULL,
  `block_name` varchar(10) DEFAULT NULL,
  `complaints` varchar(100) DEFAULT NULL,
  `room_no` int DEFAULT NULL,
  PRIMARY KEY (`block_no`),
  KEY `fk_r` (`room_no`),
  CONSTRAINT `fk_r` FOREIGN KEY (`room_no`) REFERENCES `room`
(`room_no`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

-- Insert data into block table
LOCK TABLES `block` WRITE;
/*!40000 ALTER TABLE `block` DISABLE KEYS */;

```

```
INSERT INTO `block` VALUES (1,'kaveri','Water
problem',11),(2,'Narmadha','Plumbing work',12),(3,'yamuna','Tenant
issue',13),(4,'jamuna',NULL,21);
```

```
/*!40000 ALTER TABLE `block` ENABLE KEYS */;
```

```
UNLOCK TABLES;
```

```
-- Table structure for block admin
```

```
DROP TABLE IF EXISTS `block_admin`;
```

```
/*!40101 SET @saved_cs_client = @@character_set_client */;
```

```
/*!50503 SET character_set_client = utf8mb4 */;
```

```
CREATE TABLE `block_admin` (
  `admin_id` int NOT NULL,
  `admin_name` varchar(20) DEFAULT NULL,
  `block_no` int DEFAULT NULL,
  PRIMARY KEY (`admin_id`),
  KEY `block_no` (`block_no`),
  CONSTRAINT `block_admin_ibfk_1` FOREIGN KEY (`block_no`) REFERENCES
`block` (`block_no`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
-- Insert data into block admin table
```

```
LOCK TABLES `block_admin` WRITE;
```

```
/*!40000 ALTER TABLE `block_admin` DISABLE KEYS */;
```

```
INSERT INTO `block_admin` VALUES (101,'shiva',1),(102,'rajaa',NULL);
```

```
/*!40000 ALTER TABLE `block_admin` ENABLE KEYS */;
```

```
UNLOCK TABLES;
```

```
-- Table structure for identity
```

```
DROP TABLE IF EXISTS `identity`;
```

```
/*!40101 SET @saved_cs_client = @@character_set_client */;
```

```
/*!50503 SET character_set_client = utf8mb4 */;
```

```
CREATE TABLE `identity` (
  `proof` varchar(15) DEFAULT NULL,
```

```

`owner_id` int DEFAULT NULL,
`tenant_id` int DEFAULT NULL,
UNIQUE KEY `proof` (`proof`),
KEY `owner_id` (`owner_id`),
KEY `fk_t` (`tenant_id`),
CONSTRAINT `fk_t` FOREIGN KEY (`tenant_id`) REFERENCES `tenant`
(`tenant_id`),
CONSTRAINT `identity_ibfk_1` FOREIGN KEY (`owner_id`) REFERENCES
`owner` (`owner_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

```

-- Insert data into identity table

```

LOCK TABLES `identity` WRITE;
/*!40000 ALTER TABLE `identity` DISABLE KEYS */;
INSERT INTO `identity` VALUES
('1234567890',501,NULL),('987654321',502,NULL),('2764724628',503,NULL),('9876
543456',504,NULL),('6789876987',NULL,601),('4567898769',NULL,602),('98765678
88',NULL,603),('2345676543',NULL,604),('7657876788',NULL,605);
/*!40000 ALTER TABLE `identity` ENABLE KEYS */;
UNLOCK TABLES;

```

-- Table structure for owner

```

DROP TABLE IF EXISTS `owner`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `owner` (
  `owner_id` int NOT NULL,
  `name` varchar(20) DEFAULT NULL,
  `age` int DEFAULT NULL,
  `aggrement_status` varchar(20) NOT NULL,
  `room_no` int DEFAULT NULL,
  `dob` varchar(15) DEFAULT NULL,
  PRIMARY KEY (`owner_id`),

```

```

KEY `FK_rrno` (`room_no`),
CONSTRAINT `FK_rrno` FOREIGN KEY (`room_no`) REFERENCES `room`
(`room_no`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

```

```

-- Insert data into owner table
LOCK TABLES `owner` WRITE;
/*!40000 ALTER TABLE `owner` DISABLE KEYS */;
INSERT INTO `owner` VALUES (501,'yuvarraj S',19,'yes',11,'2004-06-
06'),(502,'Tharun',19,'yes',13,'2005-12-07'),(503,'Surya DK',20,'no',21,'2004-05-
06'),(504,'Shivanesh',21,'no',31,'2003-08-04');
/*!40000 ALTER TABLE `owner` ENABLE KEYS */;
UNLOCK TABLES;

```

```

-- Table structure for rental details
DROP TABLE IF EXISTS `rental`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `rental` (
  `doj` varchar(20) DEFAULT NULL,
  `monthly_rent` int DEFAULT NULL,
  `room_no` int DEFAULT NULL,
  `tenant_id` int DEFAULT NULL,
  KEY `tenant_id` (`tenant_id`),
  KEY `FK_rno` (`room_no`),
  CONSTRAINT `FK_rno` FOREIGN KEY (`room_no`) REFERENCES `room`
(`room_no`),
  CONSTRAINT `rental_ibfk_1` FOREIGN KEY (`tenant_id`) REFERENCES `tenant`
(`tenant_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

```

```

-- Insert data into rental table

```

```

LOCK TABLES `rental` WRITE;
/*!40000 ALTER TABLE `rental` DISABLE KEYS */;
INSERT INTO `rental` VALUES ('2020-01-02',25000,11,601),('2021-01-
03',25000,12,602),('2021-04-03',35000,13,603),('2021-03-06',15000,21,604),('2021-
05-07',15000,31,605);
/*!40000 ALTER TABLE `rental` ENABLE KEYS */;
UNLOCK TABLES;

```

-- Table structure for room details

```

DROP TABLE IF EXISTS `room`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `room` (
  `room_no` int NOT NULL,
  `type` varchar(10) DEFAULT NULL,
  `floor` int DEFAULT NULL,
  `parking_slot` varchar(10) DEFAULT NULL,
  `reg_no` int DEFAULT NULL,
  `block_no` int DEFAULT NULL,
  PRIMARY KEY (`room_no`),
  UNIQUE KEY `parking_slot` (`parking_slot`),
  UNIQUE KEY `reg_no` (`reg_no`),
  KEY `block_no` (`block_no`),
  CONSTRAINT `room_ibfk_1` FOREIGN KEY (`block_no`) REFERENCES `block`
(`block_no`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

```

-- Insert data into room table

```

LOCK TABLES `room` WRITE;
/*!40000 ALTER TABLE `room` DISABLE KEYS */;
INSERT INTO `room` VALUES (11,'3bhk',2,'B-123',1234,1),(12,'2bhk',2,'B-
124',12345,2),(13,'3bhk',2,'B-125',123,1),(21,'1bhk',3,'B-234',456,4),(31,'4bhk',4,'B-
789',2345,4),(67,'1bhk',7,'B-890',654,3);

```

```

/*!40000 ALTER TABLE `room` ENABLE KEYS */;
UNLOCK TABLES;

-- Table structure for tenant details
DROP TABLE IF EXISTS `tenant`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `tenant` (
  `tenant_id` int NOT NULL,
  `name` varchar(30) DEFAULT NULL,
  `dob` varchar(10) DEFAULT NULL,
  `stat` varchar(10) DEFAULT NULL,
  `room_no` int DEFAULT NULL,
  `age` int DEFAULT NULL,
  PRIMARY KEY (`tenant_id`),
  KEY `fk_rn` (`room_no`),
  CONSTRAINT `fk_rn` FOREIGN KEY (`room_no`) REFERENCES `room`
  (`room_no`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

-- Insert data into tenant table
LOCK TABLES `tenant` WRITE;
/*!40000 ALTER TABLE `tenant` DISABLE KEYS */;
INSERT INTO `tenant` VALUES (601,'nithin','2004-11-
04','no',11,19),(602,'rohith','2002-18-23','not paid',12,23),(603,'mothi','2002-06-
12','not paid',13,41),(604,'abu danish','2002-09-23','not paid',21,35),(605,'Hari','2002-
09-30','not paid',31,56);
/*!40000 ALTER TABLE `tenant` ENABLE KEYS */;
UNLOCK TABLES;

-- Reset SQL configurations
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;

```

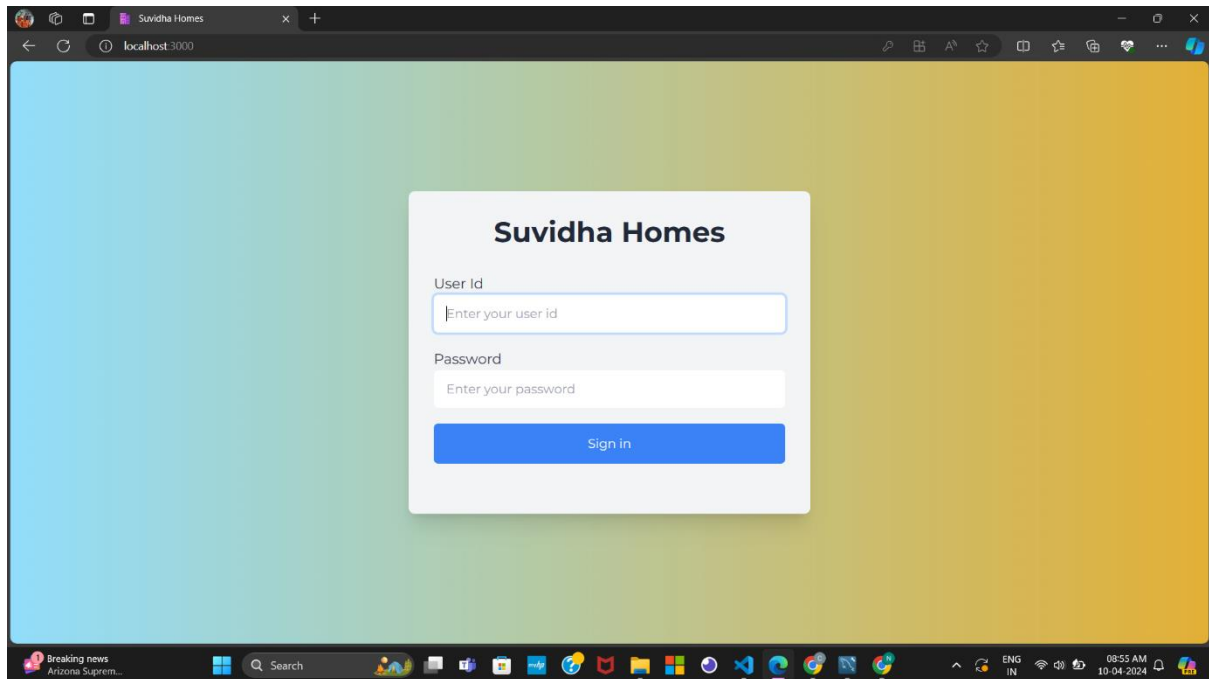
```
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;  
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;  
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET  
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;  
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

CHAPTER 5

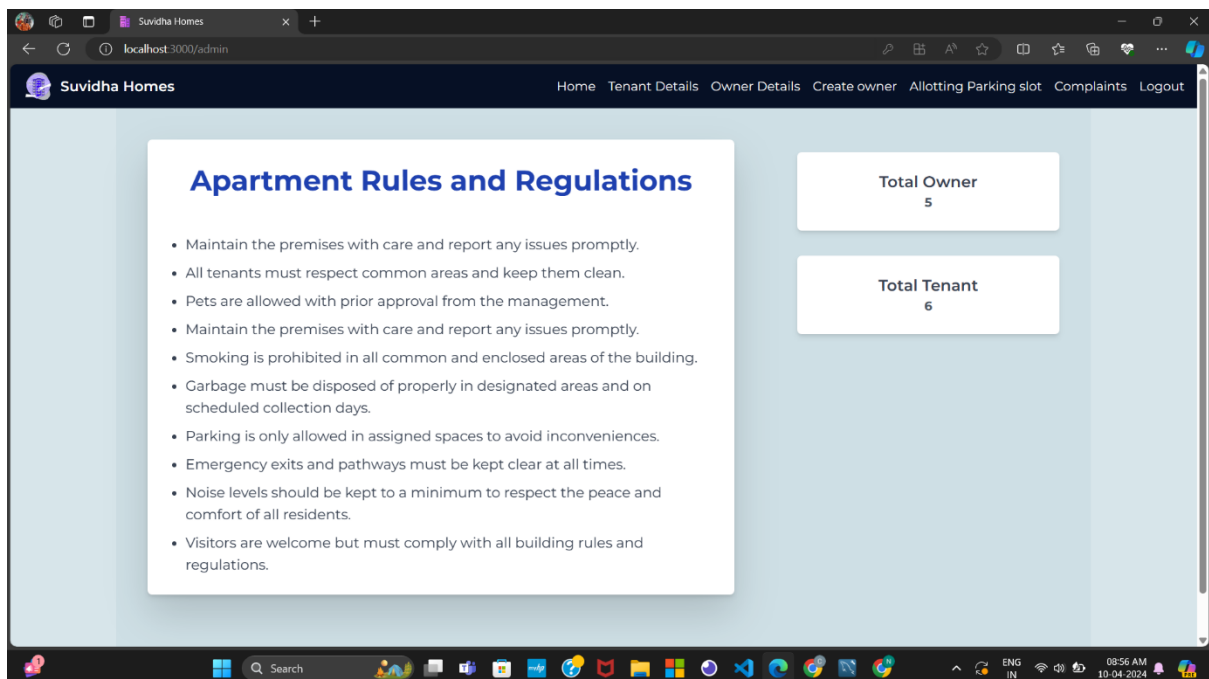
RESULTS

5.1 Sample Output (Screenshots)

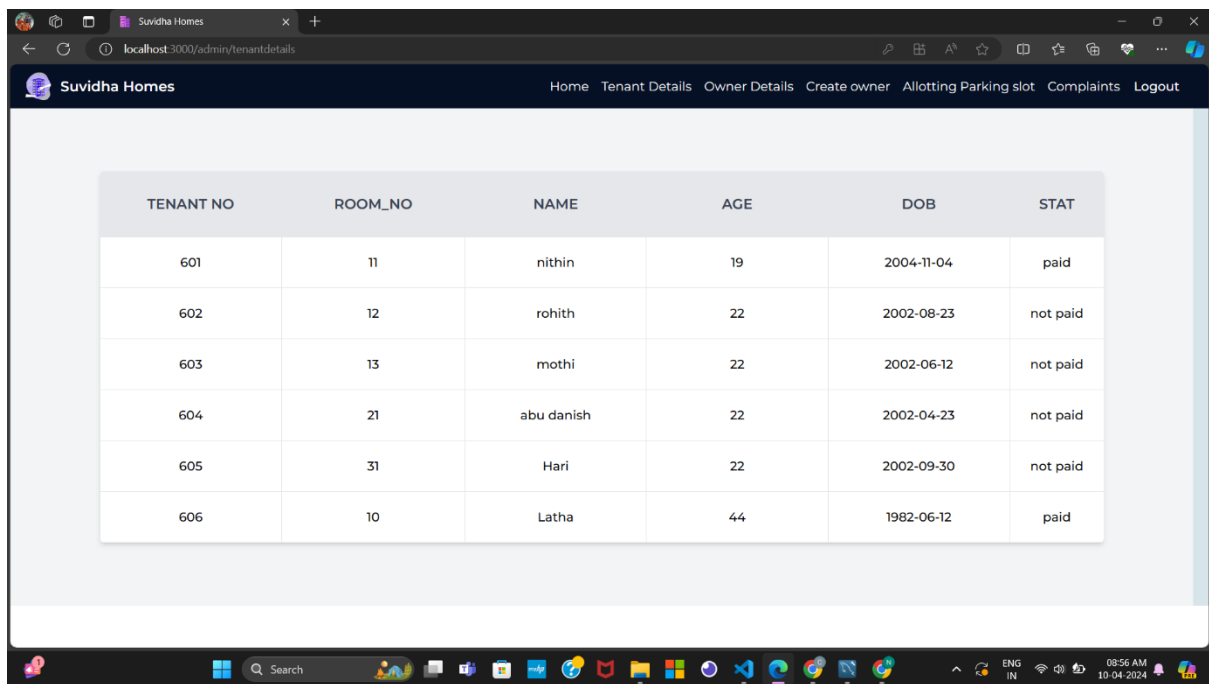
Login page:



Dashboard:



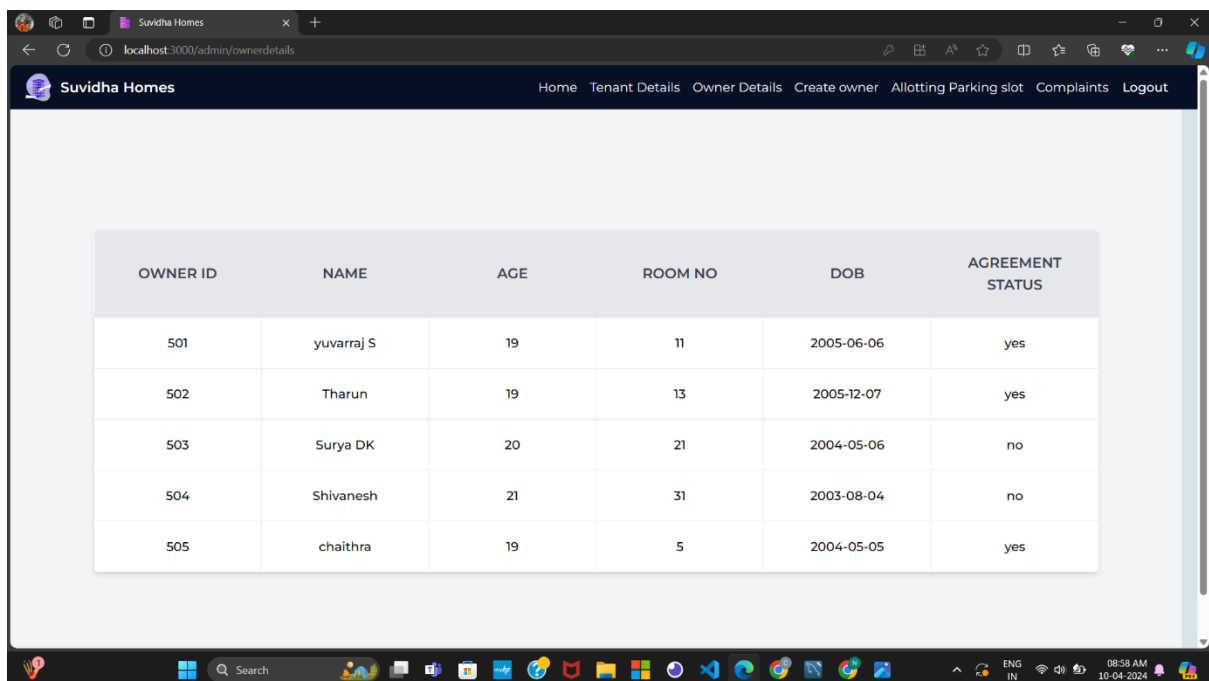
Admin viewing Tenant Details:



The screenshot shows a web browser window with the URL `localhost:3000/admin/tenantdetails`. The page has a dark blue header with the logo "Suvidha Homes" and navigation links: Home, Tenant Details, Owner Details, Create owner, Allotting Parking slot, Complaints, and Logout. The main content area displays a table with tenant information.

TENANT NO	ROOM_NO	NAME	AGE	DOB	STAT
601	11	nithin	19	2004-11-04	paid
602	12	rohith	22	2002-08-23	not paid
603	13	mothi	22	2002-06-12	not paid
604	21	abu danish	22	2002-04-23	not paid
605	31	Hari	22	2002-09-30	not paid
606	10	Latha	44	1982-06-12	paid

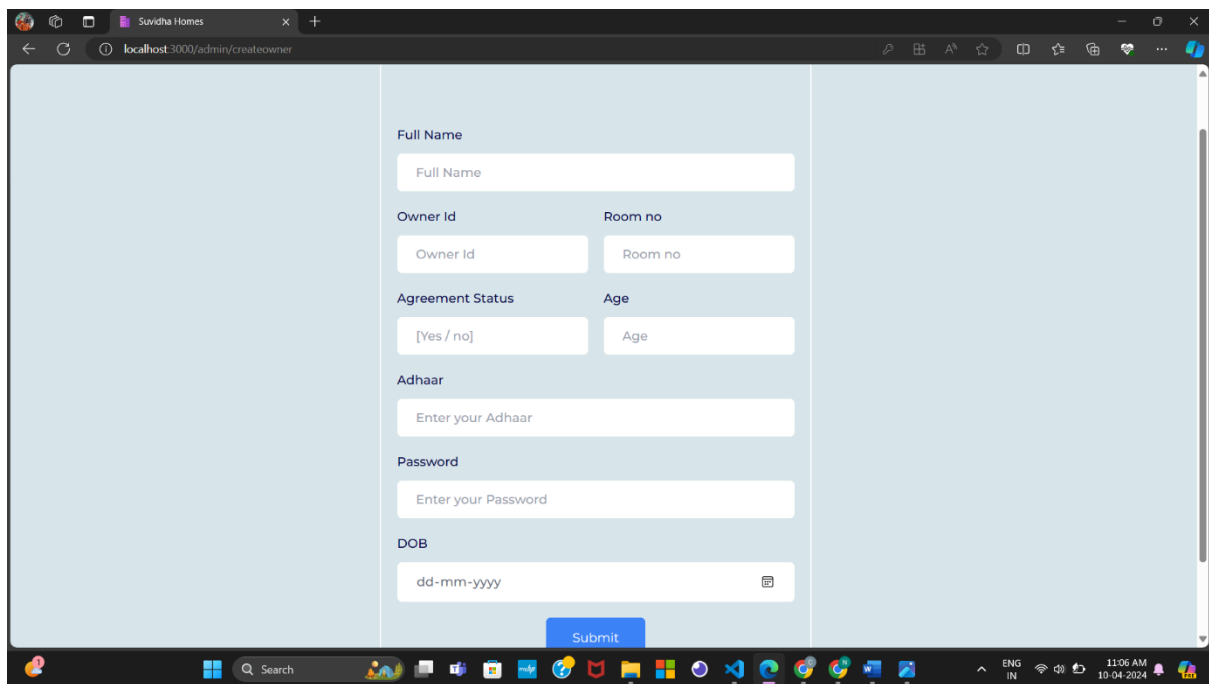
Admin viewing Owner Details:



The screenshot shows a web browser window with the URL `localhost:3000/admin/ownerdetails`. The page has a dark blue header with the logo "Suvidha Homes" and navigation links: Home, Tenant Details, Owner Details, Create owner, Allotting Parking slot, Complaints, and Logout. The main content area displays a table with owner information.

OWNER ID	NAME	AGE	ROOM NO	DOB	AGREEMENT STATUS
501	yuvraj S	19	11	2005-06-06	yes
502	Tharun	19	13	2005-12-07	yes
503	Surya DK	20	21	2004-05-06	no
504	Shivanesh	21	31	2003-08-04	no
505	chaithra	19	5	2004-05-05	yes

Admin can create new Owner:

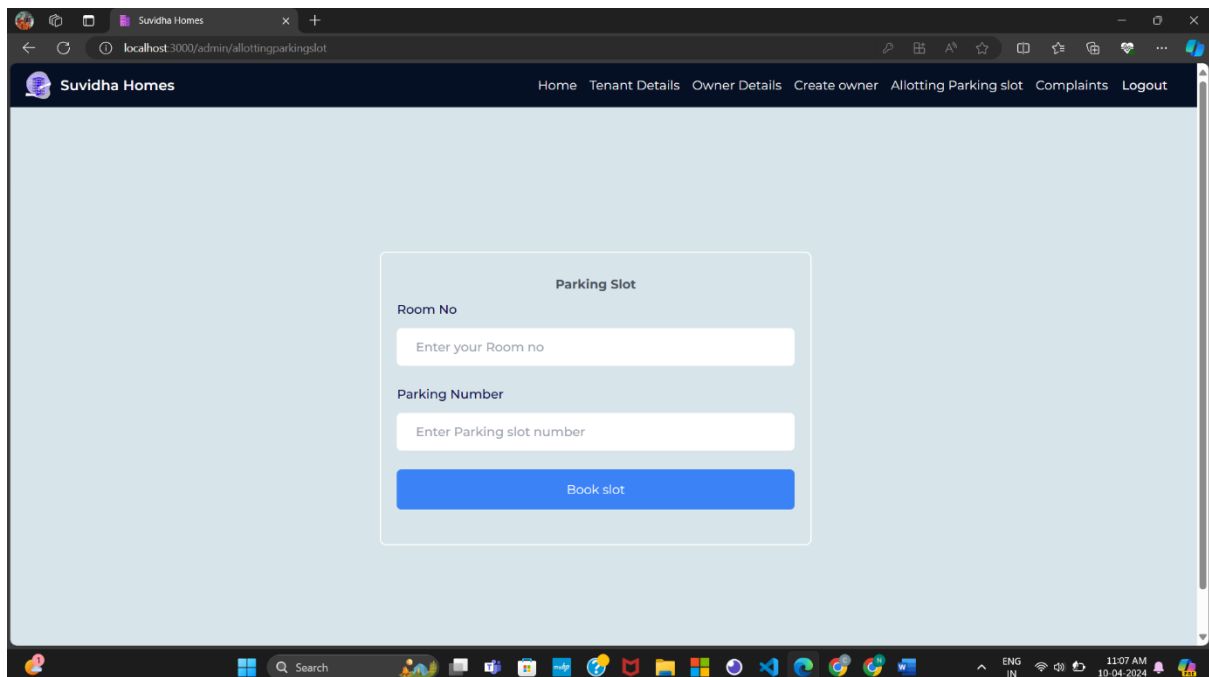


The screenshot shows a web browser window with the URL `localhost:3000/admin/createowner`. The page contains a form for creating a new owner. The form fields are:

- Full Name:
- Owner Id:
- Room no:
- Agreement Status:
- Age:
- Adhaar:
- Password:
- DOB:

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

Admin can allot Parking slot:



The screenshot shows a web browser window with the URL `localhost:3000/admin/allottingparkingslot`. The page has a navigation bar with the following links: Home, Tenant Details, Owner Details, Create owner, Allotting Parking slot, Complaints, and Logout. The main content area contains a form titled "Parking Slot" with the following fields:

- Room No:
- Parking Number:

A blue "Book slot" button is located at the bottom of the form.

Owner can create new Tenant:

The screenshot shows a web browser window with the URL `localhost:3000/owner/createtenant`. The page contains a form for creating a new tenant. The form fields are as follows:

- Full Name:** A text input field with the placeholder text "Full Name".
- Tenant No.:** A text input field with the placeholder text "Tenant No.".
- Room no:** A text input field with the placeholder text "Room no".
- DOB:** A text input field with the placeholder text "dd-mm-yyyy" and a calendar icon.
- Age:** A text input field with the placeholder text "Age".
- Adhaar:** A text input field with the placeholder text "Adhaar".
- Password:** A text input field with the placeholder text "Enter your Password".

Below the password field is a blue "Submit" button. The browser's taskbar at the bottom shows the Windows logo, a search bar, and various application icons. The system clock indicates 11:03 AM on 10-04-2024.

Tenant can add new Complaint:

The screenshot shows a web browser window with the URL `localhost:3000/tenant/raisingcomplaints`. The page has a navigation bar with the following links: Home, Raising Complaints, Alloted Parking slot, Pay maintenance, and Logout. The main content area features a white card titled "Add Complaint". The form fields on the card are:

- Room no:** A text input field with the placeholder text "Room no".
- Block no:** A text input field with the placeholder text "Block no".
- Tenant id:** A text input field with the placeholder text "Tenant id".
- Description:** A text area with the placeholder text "Write here..".

Below the description field is a blue "Add Complaint" button. The browser's taskbar at the bottom shows the Windows logo, a search bar, and various application icons. The system clock indicates 11:07 AM on 10-04-2024.

CHAPTER 6

CONCLUSION & FUTURE ENHANCEMENTS

The Apartment Management System (AMS) revolutionizes residential complex management by seamlessly integrating user-friendly interfaces with efficient MySQL database storage. AMS empowers administrators, owners, and tenants alike with tailored functionalities, ensuring smooth interaction and management within the complex. By addressing the challenges associated with manual data entry and management, AMS streamlines processes and enhances overall efficiency, thereby optimizing the residential living experience.

Future Enhancements:

- **Event Management:** Introduce a module for scheduling and tracking community events.
- **Employee Directory:** Implement a system to manage employee records and schedules.
- **Feedback Mechanism:** Incorporate a platform for tenants to provide feedback on various aspects of their living experience.
- **Visitor Tracking:** Develop a system to track and manage guest visits for enhanced security.
- **Maintenance Requests:** Enhance the complaint management system to specifically track maintenance issues reported by tenants.

CHAPTER 7

REFERENCES

- OpenAI - <http://www.openai.com/>
- GitHub - <http://www.github.com/>
- Stack OverFlow - <https://stackoverflow.com/>
- YouTube Videos:
 - MySQL Tutorial: Search for MySQL tutorials on YouTube for comprehensive learning resources.
 - React Tutorial: Explore various React tutorials available on YouTube for learning front-end development.
 - Node.js Tutorial: Find numerous Node.js tutorials on YouTube covering backend development techniques and best practices.