

Indian Institute of Information Technology, Allahabad



Database Management System Project Report On Community Management System

Submitted By Group 4:

Abhishek Kumar	(IIT2022122)
Induri Naga Aditya Reddy	(IIT2022123)
Ungarala Suhas	(IIT2022128)
Mukund Kumar	(IIT2022183)
Tarun Balaji Nidiganti	(IIT2022195)
Venkatesh Vadthya	(IIT2022200)

Table Of Contents

Serial Number	Topic	Page.No
1	Acknowledgement	3
2	Abstract	4
3	Contents	5
4	Problem Statement	6
5	Front-End	7
6	Back-End	7-8
7	ER Diagram	8-9
8	Table Description	9-16
9	Relational Schema	17
10	Constraints in Relational Schema	18-19
11	Functional Components	20
12	Code	21-29

Acknowledgement

We would like to thank our course instructor Dr. Soumyadev Maity Sir For giving us the proper guidance which proved to be very helpful for This course and to make this project and we would like to thank our Assigned Teaching assistants for providing valuable inputs throughout Span of this project.

Abstract

In the growing number of gated societies in today's world, we have observed a gap where the lack of a proper management system still exists. This can be achieved by using an easy application where we can have a common interacting space between the four users of our community which are Admin, Resident, Security and Workforce respectively. This app caters to resident needs like allow visitors with proper safety measures and view their information, booking community spaces and view their upcoming bookings, a dedicated complaint desk, a platform to view all the early maintenance and electricity bills at one place, view notices posted by the admin and lastly view information about neighbours and fellow residents for easier contact. Next at security we have given an add residents feature where a special code given to the resident while adding visitor is checked and the visitor is given entry ensuring proper safety and ease of use, next security can view the residents and residents phone numbers like a directory in case of an emergency. Next comes the workforce who can view the complaints posted by the residents, serve them and then update status on the complaints. The admin is the only login that has the rights to look over all the functions of all the other logins and he is the one who adds all the other users and assigns logins to assure security and looks over other admin activities like adding bills overseeing complaints etc.

CONTENTS

1. Introduction

1.1 Problem Statement

2. Project Component Description

2.1 Front End

2.2 Back End

3. Database Design

3.1 ER Diagram

3.2 Relational Schema

3.3 Primary Keys, Foreign Keys, Triggers

4. Functional Components

5. Code

Introduction

1.1 Problem Statement:

In today's proliferating gated communities, a notable deficiency persists in the form of inefficient management systems, leading to substantial communication barriers among key stakeholders: Administrators, Residents, Security Personnel, and the Workforce. This gap impedes effective coordination and administration within these communities. To address this pressing issue, our proposed solution entails the development of a comprehensive application tailored specifically to the needs of gated communities.

This application aims to streamline various aspects of community management by providing a centralized platform that facilitates seamless interaction and collaboration among residents and other stakeholders. Residents will be empowered to manage visitors, book community spaces, file complaints, access bills and notices, and engage in neighbourly communication. Security personnel will benefit from features like streamlined visitor entry and emergency contact access. The Workforce will efficiently handle resident complaints, ensuring timely resolution and updates.

Central to this solution is the role of the Administrator, who will have overarching control over all application functionalities, user management, security protocols, and administrative tasks. Through this application, we seek to bridge the existing management gap in gated communities, ultimately enhancing communication, coordination, and administration to foster a more cohesive and secure living environment.

Project Component Description

2.1 Front-End

It provides an user friendly and simple interface with access to all buttons from every page and logout on every page with simple standout black and white colour to improve readability.

Key Front-End Features:

- Proper display of tables in a new opened tab with consistent UI for better user experience.
- Proper input boxes and mostly restricted to drop down menus ensuring less human error while typing out words which may or may not be recognized by the respective database values.
- Clear notice displays on top for proficient view of notices.
- Implementing a dashboard like setup with all available options on a side bar for quick access.

Technologies Used: HTML, CSS, JS

2.2 Back-End

The back end of the system manages data storage, processing, and retrieval. Key components include:

- **Database:** Utilizes MySQL for storing user details, feedback data, and menu information.
- **Server-side Scripting:** Implemented in PHP to handle data retrieval, processing, and database interactions.
- **Triggers:** MySQL triggers are used to implement the security features of allowing visitors and moving their entries to active visitors and similarly from active to past. Also been used to move

the entries of active Complaints to past complaints once served and reported by workforce.

- **User Authentication:** Helps provide proper authentication and helped us implement different views/logins for different type of user in the application.

Database Design

3.1 ER Diagram

This ER diagram represents the model of Community Management System. The Entity Relationship (ER) diagram of the Community Management System show all the visual instruments of database tables and relationship between Resident, Flats, Admin, Complaint Desk, Active Visitors, Past Visitors, Passcodes, Security, Maintenance Bills, Electricity Bills, Past Complaints and Book Community Spaces. It used structured data and define relationship between structured data groups of Community Management System functionalities. The Relations are Adds, has, contains, complaints, create, pays, books, etc.

The Entities involved in the ER-diagram are –

- (1) Resident
- (2) Flats
- (3) Admin
- (4) Complaint Desk
- (5) Active Visitors

- (6) Past Visitors
- (7) Passcodes
- (8) Security
- (9) Maintenance Bills
- (10) Electricity Bills
- (11) Past Complaints
- (12) Book Community Spaces.
- (13) Loginids
- (14) Workforce

TABLE DESCRIPTION

(1) Resident

SR. NO.	ATTRIBUTE NAME	ATTRIBUTE TYPE
1	Name	VARCHAR
2	Gender	VARCHAR
3	flatno	INT
4	towerno	INT
5	phonenumner	VARCHAR
6	username	VARCHAR
7	password	VARCHAR

(2) Flats

SR. NO.	ATTRIBUTE NAME	ATTRIBUTE TYPE
1	flatno	INT
2	towerno	INT
3	floorno	INT
4	flatsize	VARCHAR

(3) Admin

SR. NO.	ATTRIBUTE NAME	ATTRIBUTE TYPE
1	username	VARCHAR
2	password	VARCHAR
3	name	VARCHAR

(4) Active Complaints

SR. NO.	ATTRIBUTE NAME	ATTRIBUTE TYPE
---------	----------------	----------------

1	flatno	INT
2	towerno	INT
3	Complaint_type	VARCHAR
4	Location	VARCHAR
5	Description	VARCHAR
6	Status	VARCHAR

(5) Active Visitors

SR. NO.	ATTRIBUTE NAME	ATTRIBUTE TYPE
1	Name	VARCHAR
2	Date_of_entry	DATE
3	Time_of_entry	TIMESTAMP
4	flatno	INT
5	towerno	INT
6	Passcode	VARCHAR

(6) Past Visitors

SR. NO.	ATTRIBUTE NAME	ATTRIBUTE TYPE
1	Name	VARCHAR
2	Date_of_exit	DATE
3	Time_of_exit	TIMESTAMP
4	Date_of_entry	DATE
5	Time_of_entry	TIMESTAMP
6	flatno	INT
7	towerno	INT

(7) Passcodes

SR. NO.	ATTRIBUTE NAME	ATTRIBUTE TYPE
1	Name	VARCHAR
2	Date_of_entry	DATE
3	Time_of_entry	TIMESTAMP
4	flatno	INT
5	towerno	INT
6	Passcode	VARCHAR

(8) Security

SR. NO.	ATTRIBUTE NAME	ATTRIBUTE TYPE
1	Assigned_place	VARCHAR
2	Security_ID	VARCHAR
3	Name	VARCHAR
4	Phone_number	VARCHAR

(9) Maintenance Bills

SR. NO.	ATTRIBUTE NAME	ATTRIBUTE TYPE
1	flatno	INT
2	towerno	INT
3	date	DATE
4	due_amount	INT

(10) Electricity Bills

SR. NO.	ATTRIBUTE NAME	ATTRIBUTE TYPE
1	flatno	INT
2	towerno	INT
3	Billing_date	DATE
4	due_amount	INT

(11) Past Complaints:

SR. NO.	ATTRIBUTE NAME	ATTRIBUTE TYPE
1	flatno	INT
2	towerno	INT
3	Complaint_type	VARCHAR
4	location	VARCHAR
5	decription	VARCHAR

(12) Book Community Spaces:

SR. NO.	ATTRIBUTE NAME	ATTRIBUTE TYPE
1	flatno	INT
2	towerno	INT
3	Community_space	VARCHAR
4	date	DATE
5	Start_time	TIMESTAMP
6	End_time	TIMESTAMP

(13) Loginids:

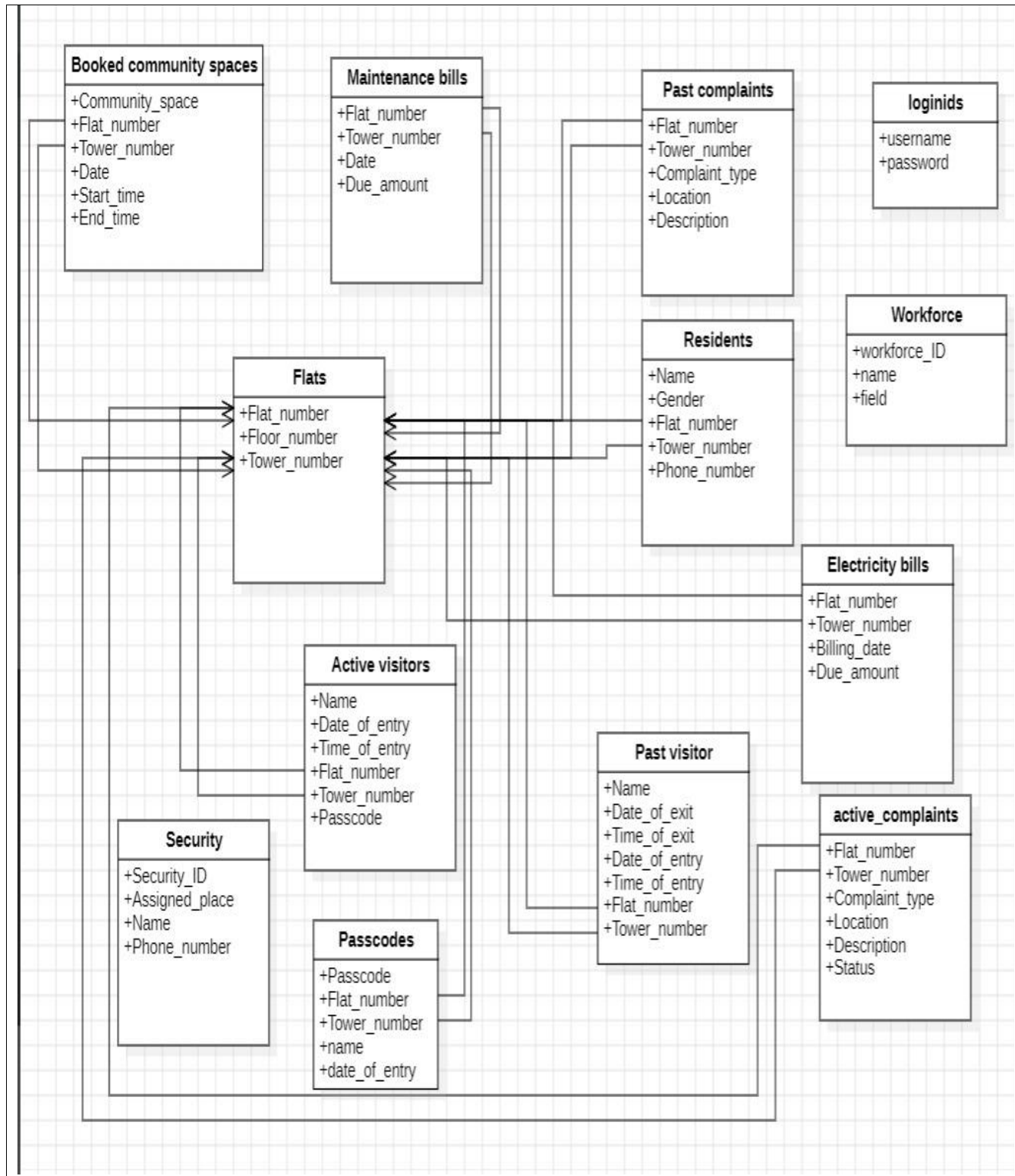
SR. NO.	ATTRIBUTE NAME	ATTRIBUTE TYPE
1	Username	VARCHAR
2	password	VARCHAR

(14) Workforce:

SR. NO.	ATTRIBUTE NAME	ATTRIBUTE TYPE
1	Workforce_ID	VARCHAR
2	name	VARCHAR
3	field	VARCHAR



3.2 Relational Schema:



Relational schematic diagram for Community Management System

3.3 Constraints in Relation Schema:

Give all the types of constraints with explanations that you have used for your project. For example:

- Key Constraints

Relation	Primary Key	Foreign Key
Flats	(flatno,towerno)	
Residents	(flatno,towerno,name)	(flatno,towerno)
Security	Security_id	
Passcodes	(passcode)	(flatno,towerno)
activevisitors	(passcode)	(flatno,towerno)
pastvisitors		(flatno,towerno)
maintenancebills	(flatno,towerno,month)	(flatno,towerno)
electricitybills	(flatno,towerno,month)	(flatno,towerno)
activecomplaints	(flatno,towerno,type,date)	(flatno,towerno)
pastcomplaints	(flatno,towerno,type,date)	(flatno,towerno)
bookedcommunityspaces	(space,date,time)	(flatno,towerno)
workforce	(id)	
loginids	(username)	

EXPLANATION:

1. **flats:** This table is used to store all the flat information and in which tower are they located.
2. **residents:** This table contains a list of all the residents residing in the flats from the flats table.
3. **security:** This table contains a list of all the security present in the community along with their allotted spaces.
4. **workforce:** This table contains a list of all the workforce present in the community along with their professions.
5. **passcodes:** This contains a list of all the expected visitors with their name and date of arrival.
6. **Activevisitors :** This contains a list of all visitors who have given passcode and are still in the community .
7. **Maintenancebills :** This contains a list of all the maintenancebills of the residents.
8. **Electricitybills :** This contains a list of all the electricitybills of the residents.
9. **Activecomplaints :** This contains a list of all the complaints given by the resident which are yet to be resolved.
10. **Pastcomplaints :** This contains a list of all the complaints given by the resident which are resolved by the workforce.
11. **Bookedcommunityspaces :** This contains a list of all the bookings made by the residents overtime.
12. **Loginids :** This contains all the login credentials of every user and the login access they have to be given.

Functional Components:

1. Session Management:

- The script starts with **session_start()** to initiate a session. Sessions are used to persist user data across multiple pages until the session is terminated.

2. Database Connection:

- **\$conn** is a variable storing the connection to the MySQL database using **mysqli_connect()** function. It connects to the database server with the specified credentials (**localhost, root, "", project**).

3. Data Retrieval Queries:

- Several SQL queries (**\$query1, \$query2, \$query3, \$query4**) retrieve items from the table based on the type of query.

4. HTML Content Generation:

- HTML content is generated using PHP echo statements mixed with HTML code. It dynamically populates items based on the query results.

5. Form Submission Handling:

- The script contains multiple HTML forms, each with a submit button. These forms submit the selected fields to the respective PHP file mentioned in action for further processing.

Code:

```
--
-- Table structure for table `activevisitors`
--

DROP TABLE IF EXISTS `activevisitors`;
CREATE TABLE `activevisitors` (
  `flatno` int NOT NULL,
  `towerno` int NOT NULL,
  `visname` varchar(30) NOT NULL,
  `visdate` date NOT NULL,
  `otp` int NOT NULL,
  PRIMARY KEY (`otp`),
  KEY `fkactive` (`flatno`,`towerno`),
  CONSTRAINT `fkactive` FOREIGN KEY (`flatno`,`towerno`) REFERENCES `flats`
  (`flatno`,`towerno`)
) ;
--
-- Dumping data for table `activevisitors`
--

INSERT INTO `activevisitors` VALUES (123,1,'akdh','2024-04-23',848920);

-- trigger on active visitors
CREATE TRIGGER `pastvisitors_entry` AFTER DELETE ON `activevisitors` FOR EACH
ROW BEGIN
  INSERT INTO pastvisitors (flatno, towerno, visname, visdate, otp)
  VALUES (OLD.flatno, OLD.towerno, OLD.visname, OLD.visdate, OLD.otp);
END;

--
-- Table structure for table `bookings`
--

CREATE TABLE `bookings` (
  `datey` date NOT NULL,
  `timeslot` varchar(20) NOT NULL,
  `area` varchar(30) NOT NULL,
  `flatno` int NOT NULL,
```

```

    `townerno` int NOT NULL,
    PRIMARY KEY (`flatno`,`townerno`,`timeslot`),
    CONSTRAINT `fkbook` FOREIGN KEY (`flatno`,`townerno`) REFERENCES `flats`
(`flatno`,`townerno`)
);

--
-- Dumping data for table `bookings`
--

INSERT INTO `bookings` VALUES ('2024-04-19','4-5','volleyball',123,1),('2024-
04-24','5-6','cricketnet',123,1),('2024-04-26','6-
7','volleyball',123,1),('2024-04-02','4-5','basketball',123,2),('2024-04-
24','5-6','basketball',128,1),('2024-04-02','5-6','basketball',185,1),('2024-
04-25','4-5','basketball',195,1);

--
-- Table structure for table `complaints`
--

CREATE TABLE `complaints` (
  `place` varchar(30) NOT NULL,
  `timeslot` varchar(20) NOT NULL,
  `problem` varchar(100) NOT NULL,
  `townerno` int NOT NULL,
  `flatno` int NOT NULL,
  PRIMARY KEY (`flatno`,`townerno`,`problem`,`timeslot`),
  CONSTRAINT `fkcomp` FOREIGN KEY (`flatno`,`townerno`) REFERENCES `flats`
(`flatno`,`townerno`)
);

--
-- Dumping data for table `complaints`
--

INSERT INTO `complaints` VALUES ('5th floor','8-9','Plumber',1,123);

CREATE TRIGGER `complaints_trigger` AFTER DELETE ON `complaints` FOR EACH ROW
BEGIN
  INSERT INTO pastcomplaints (flatno, towerno, place, timeslot, problem,
timeofresolution)
  VALUES (OLD.flatno, OLD.townerno, OLD.place, OLD.timeslot, OLD.problem,
NOW());
END;

--

```

```

-- Table structure for table `electricitybills`
--

CREATE TABLE `electricitybills` (
  `flatno` int NOT NULL,
  `towerno` int NOT NULL,
  `mon` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  `dueamt` int NOT NULL,
  PRIMARY KEY (`flatno`,`towerno`,`mon`),
  CONSTRAINT `fkelec` FOREIGN KEY (`flatno`,`towerno`) REFERENCES `flats`
(`flatno`,`towerno`)
);

--
-- Dumping data for table `electricitybills`
--

INSERT INTO `electricitybills` VALUES
(123,1,'july',250),(123,1,'may',2000),(200,5,'april',2000);

--
-- Table structure for table `flats`
--

CREATE TABLE `flats` (
  `flatno` int NOT NULL,
  `towerno` int NOT NULL,
  PRIMARY KEY (`flatno`,`towerno`)
);

--
-- Dumping data for table `flats`
--

INSERT INTO `flats` VALUES
(123,1),(123,2),(128,1),(156,2),(185,1),(195,1),(200,1),(200,5);

--
-- Table structure for table `loginids`
--

CREATE TABLE `loginids` (
  `email` varchar(30) NOT NULL,
  `passcode` varchar(30) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT
NULL,
  `designation` varchar(30) NOT NULL,

```

```

    PRIMARY KEY (`email`)
);

--
-- Dumping data for table `loginids`
--

INSERT INTO `loginids` VALUES
('abhishek','abhi','resident'),('adi.com','adi','admin'),('balu','123','resident'),
('first','first','resident'),('harshin.com','har','security'),('mukund','mukund','workforce'),
('suh.com','suh','security'),('tar.com','tar','resident'),('ven.com','ven','workforce'),
('venkatesh','venky','resident');

--
-- Table structure for table `maintenancebills`
--

CREATE TABLE `maintenancebills` (
  `flatno` int NOT NULL,
  `towerno` int NOT NULL,
  `mon` varchar(20) NOT NULL,
  `dueamt` int NOT NULL,
  PRIMARY KEY (`flatno`,`mon`,`towerno`),
  KEY `fkmain` (`flatno`,`towerno`),
  CONSTRAINT `fkmain` FOREIGN KEY (`flatno`,`towerno`) REFERENCES `flats`
(`flatno`,`towerno`)
);

--
-- Dumping data for table `maintenancebills`
--

INSERT INTO `maintenancebills` VALUES
(123,1,'january',505),(123,1,'may',5000),(200,5,'january',2500);

--
-- Table structure for table `mem_presidents`
--

CREATE TABLE `mem_presidents` (
  `pres_name` varchar(30) NOT NULL,
  `towerno` int NOT NULL,
  `phone` varchar(20) NOT NULL,
  PRIMARY KEY (`towerno`)
);

```



```

--
-- Dumping data for table `mem_presidents`
--

INSERT INTO `mem_presidents` VALUES ('suhas',1,'999999999');

--
-- Table structure for table `mem_residents`
--

CREATE TABLE `mem_residents` (
  `flatno` int NOT NULL,
  `towerno` int NOT NULL,
  `resi_name` varchar(30) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT
NULL,
  `gender` varchar(10) NOT NULL,
  `phone` varchar(15) NOT NULL,
  PRIMARY KEY (`towerno`,`resi_name`,`flatno`),
  KEY `fkmemr` (`flatno`,`towerno`),
  CONSTRAINT `fkmemr` FOREIGN KEY (`flatno`,`towerno`) REFERENCES `flats`
(`flatno`,`towerno`)
);

--
-- Dumping data for table `mem_residents`
--

INSERT INTO `mem_residents` VALUES
(123,1,'first','Male','884764638'),(128,1,'suhas','Male','99999999999'),(200,1,
'venkat','Male','1234567899'),(156,2,'abhishek','Male','7937893247'),(200,5,'v
enkatesh','Male','9898989898');

--
-- Table structure for table `mem_security`
--

CREATE TABLE `mem_security` (
  `security_id` varchar(30) NOT NULL,
  `sec_name` varchar(30) NOT NULL,
  `phone` varchar(15) NOT NULL,
  `assigned_place` varchar(50) NOT NULL,
  PRIMARY KEY (`security_id`)
);

--
-- Dumping data for table `mem_security`

```

```

--

INSERT INTO `mem_security` VALUES ('ABC','Bob','6666666666','gate');

--
-- Table structure for table `mem_workforce`
--

CREATE TABLE `mem_workforce` (
  `worker_name` varchar(30) NOT NULL,
  `phone` varchar(15) NOT NULL,
  `field` varchar(15) NOT NULL,
  PRIMARY KEY (`phone`)
);

--
-- Dumping data for table `mem_workforce`
--

INSERT INTO `mem_workforce` VALUES ('mukund','9898989898','Plumber');

--
-- Table structure for table `notices`
--

CREATE TABLE `notices` (
  `notice` varchar(258) NOT NULL,
  `startdate` date NOT NULL,
  `enddate` date NOT NULL
);

--
-- Dumping data for table `notices`
--

INSERT INTO `notices` VALUES ('welcome to iiita','2024-04-26','2024-04-27'),('welcome to clg','2024-04-26','2024-04-27'),('welcome','2024-04-25','2024-04-27'),('yufif','2024-04-25','2024-04-30');

--
-- Table structure for table `passcodes`
--

CREATE TABLE `passcodes` (
  `visiname` varchar(30) NOT NULL,
  `visidate` date NOT NULL,

```

```

`flatno` int NOT NULL,
`towerno` int NOT NULL,
`otp` int NOT NULL,
PRIMARY KEY (`otp`),
KEY `fkpass` (`flatno`,`towerno`),
CONSTRAINT `fkpass` FOREIGN KEY (`flatno`,`towerno`) REFERENCES `flats`
(`flatno`,`towerno`)
);

--
-- Dumping data for table `passcodes`
--

INSERT INTO `passcodes` VALUES ('match','2024-04-19',195,1,746894);

CREATE TRIGGER `passcode_trigger` AFTER DELETE ON `passcodes` FOR EACH ROW
BEGIN
    INSERT INTO activevisitors (flatno, towerno, visname, visdate, otp)
    VALUES (OLD.flatno, OLD.towerno, OLD.visiname, OLD.visidate, OLD.otp);
END ;

--
-- Table structure for table `pastcomplaints`
--

CREATE TABLE `pastcomplaints` (
  `place` varchar(30) NOT NULL,
  `timeslot` varchar(30) NOT NULL,
  `problem` varchar(30) NOT NULL,
  `towerno` int NOT NULL,
  `flatno` int NOT NULL,
  `timeofresolution` timestamp(6) NOT NULL,
  KEY `fkpastc` (`flatno`,`towerno`),
  CONSTRAINT `fkpastc` FOREIGN KEY (`flatno`,`towerno`) REFERENCES `flats`
(`flatno`,`towerno`)
);

--
-- Dumping data for table `pastcomplaints`
--

INSERT INTO `pastcomplaints` VALUES ('gate','11:00 -
12:00','Network',1,123,'2024-04-23 19:49:04.000000'),('gate','17:00 -
18:00','Plumber',1,195,'2024-04-25 00:32:04.000000'),('cc3','10:00 -
11:00','Electrician',1,123,'2024-04-25 11:41:00.000000'),('cc3','12:00 -
13:00','Plumber',1,123,'2024-04-25 11:42:17.000000');

```

```
--
-- Table structure for table `pastvisitors`
--

CREATE TABLE `pastvisitors` (
  `flatno` int NOT NULL,
  `towerno` int NOT NULL,
  `visname` varchar(30) NOT NULL,
  `visdate` date NOT NULL,
  `otp` int NOT NULL,
  PRIMARY KEY (`otp`,`visdate`),
  KEY `fkpastv` (`flatno`,`towerno`),
  CONSTRAINT `fkpastv` FOREIGN KEY (`flatno`,`towerno`) REFERENCES `flats`
  (`flatno`,`towerno`)
);

--
-- Dumping data for table `pastvisitors`
--

INSERT INTO `pastvisitors` VALUES (195,1,'srh_always','2024-04-24',149887),(123,1,'adi','2024-04-20',332274),(123,1,'first','2024-04-26',392324),(123,2,'sdf','2024-04-16',584476),(195,1,'srh','2024-04-24',904205);

--
-- Table structure for table `user_res`
--

CREATE TABLE `user_res` (
  `email` varchar(30) NOT NULL,
  `passcode` varchar(20) NOT NULL,
  `towerno` int NOT NULL,
  `flatno` int NOT NULL,
  PRIMARY KEY (`email`),
  KEY `fkuser` (`flatno`,`towerno`),
  CONSTRAINT `fkuser` FOREIGN KEY (`flatno`,`towerno`) REFERENCES `flats`
  (`flatno`,`towerno`),
  CONSTRAINT `fkuser_res` FOREIGN KEY (`email`) REFERENCES `loginids`
  (`email`)
);

--
-- Dumping data for table `user_res`
--
```

```
;
INSERT INTO `user_res` VALUES ('first','first',1,123),('tar.com','tar',1,195);
-- Dump completed on 2024-04-29  4:43:27
```