

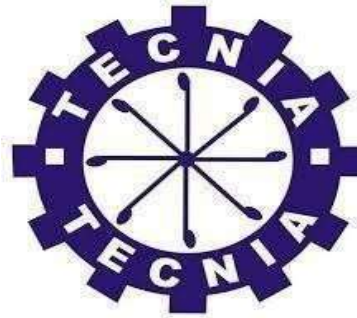
**DEPARTMENT OF INFORMATION, COMMUNICATION
&
TECHNOLOGY**

MINI PROJECT

OF

BCA-102T

DATABASE MANAGEMENT SYSTEM



Submitted to: -
Mr. Gaurav Singh Rawat
(Subject In- charge)

Submitted By: -
Name: Preeti Raj
Enrolment No.: 06617002024

Program: BCA
Semester: 2nd

Shift: Morning B

Session: 2024 - 2025

Batch: 2024 – 2028

ABSTRACT

This project is a backend-focused College Management System developed using MySQL. It is designed to efficiently manage core administrative tasks such as **room allocation, student information, complaint handling, and fee tracking**. The system is structured with relational database principles and ensures data consistency through the use of primary and foreign keys. While it lacks a front-end interface, it provides a reliable and scalable data structure that can support future application development or integration.

INTRODUCTION

Educational institutions require an organized system to manage student-related data and resources. Traditional manual methods often lead to inefficiencies and errors. To overcome these issues, this project introduces a backend database solution that can manage essential college operations. Using MySQL, the system ensures a structured approach to storing and retrieving data related to rooms, students, complaints, and fees. The database is intended to serve as the foundation for a full-fledged student management application.

OVERVIEW

The project consists of the following main components:

- Student Table: Stores details such as student name, course, department, contact number, address, and allocated room.
- Room Table: Manages information related to room number, floor, capacity, room type (AC/NON-AC), availability, and charges.
- Fees Table: Tracks total fees, paid amount, and calculates the due amount using a generated column.
- Complaints Table: Records complaints raised by students with status tracking.

All tables are connected using foreign key constraints to maintain relational integrity. The database allows insertion, deletion, updating, and querying of data through SQL statements, providing an efficient and secure backend structure.

CODE

Create database collegemanagement;

Use collegemanagement;

```
Create table student(  
    studId int primary key,  
    name varchar(50) not null,  
    course varchar(10) not null,  
    department varchar(25) not null,  
    phoneNo bigint(20) unique,  
    address varchar(80),  
    roomId int ,  
    foreign key (roomId) references room (roomId)  
);
```

```
Insert into student (studId ,name ,course ,department ,phoneNo ,address) values  
( 101 , 'Arav Mehta' , 'bca' , 'computer science' , 9632587412 , 'block a, room 204' ),  
( 102 , 'Priya Sharma' , 'bba' , 'business admin' , 9874563210 , 'block b, room 105' ),  
( 103 , 'Rohan Singh' , 'b.tech' , 'mechanical' , 1123654789 , 'block c, room 309' ),  
( 104 , 'Sneha Kapoor' , 'b.sc' , 'physics' , 7845691231 , 'block a, room 110' ),  
( 105 , 'Vikas Nair' , 'bca' , 'computer science' , 5879456321 , 'block b, room 202' );
```

```
Create table room(  
    roomId int primary key,  
    floor int not null,  
    capacity int not null,  
    roomType varchar(8),  
    availability varchar(20),  
    charge int,  
    bedNo int  
);
```

```
Insert into room values  
(1,1,2, 'NON AC', 'available', 3000, 2),  
(2,2,3, 'AC', 'full', 4500, 3),  
(3,3,1, 'NON AC', 'available', 2500, 1),  
(4, 1, 2, 'AC', 'full', 4500, 2);
```

```
CREATE TABLE complaints (  
    complaintId int(11) primary key,  
    studId int(11),  
    relatedTo enum( 'bullying', 'hostel', 'water', 'electricity', 'teachers' , 'other'),
```

```
description varchar(200),
status varchar(15) DEFAULT pending,
foreign key (studId) references student(studId)
);
```

```
insert into complaints values
(1,102, 'hostel', 'ac not working', 'pending'),
(2,103, 'water', 'water leakage', 'resolved' ),
(3,105, 'hostel' , 'no light in bathroom' , 'pending');
```

```
CREATE TABLE fee (
    id int(11)primary key,
    studId int(11) ,
    totalFee int(11) NOT NULL,
    paid int(11) ,
    status varchar(5) ,
    foreign key (studId) references student(studId)
);
```

```
Insert into fee values
(1,101,3000,3000, 'paid'),
(2,102,4500,2500, 'due'),
(3,103,2500,2500, 'paid'),
(4,104,3000,3000, 'paid'),
(5,105,4500,4000, 'due');
```

```
INSERT INTO student (roomid) VALUES
(1),
(1),
(2),
(2),
(3);
```

OUTPUT

```
mysql> show tables;
```

Tables_in_collegemanagement	
complaints	
fee	
room	
student	

```
4 rows in set (0.01 sec)
```

```
mysql> desc student;
```

Field	Type	Null	Key	Default	Extra
studId	int(11)	NO	PRI	NULL	
name	varchar(50)	NO		NULL	
course	varchar(10)	NO		NULL	
department	varchar(25)	NO		NULL	
phoneNo	bigint(20)	YES	UNI	NULL	
address	varchar(80)	YES		NULL	
roomId	int(11)	YES	MUL	NULL	

```
7 rows in set (0.02 sec)
```

```
mysql> desc room;
```

Field	Type	Null	Key	Default	Extra
roomId	int(11)	NO	PRI	NULL	
floor	int(11)	NO		NULL	
capacity	int(11)	NO		NULL	
roomType	varchar(8)	YES		NULL	
availability	varchar(20)	YES		NULL	
charge	int(11)	YES		NULL	
bedNo	int(11)	YES		NULL	

```
7 rows in set (0.03 sec)
```

```
mysql> desc complaints;
```

Field	Type	Null	Key	Default	Extra
complaintId	int(11)	NO	PRI	NULL	
studId	int(11)	YES	MUL	NULL	
relatedTo	enum('bullying', 'hostel', 'water', 'electricity', 'teachers', 'other')	YES		NULL	
description	varchar(200)	YES		NULL	
status	varchar(15)	YES		pending	

```
5 rows in set (0.01 sec)
```

```
mysql> desc fee;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	
studId	int(11)	YES	MUL	NULL	
totalFee	int(11)	NO		NULL	
paid	int(11)	YES		NULL	
status	varchar(5)	YES		NULL	

```
5 rows in set (0.02 sec)
```

```
mysql> select * from student;
```

studId	name	course	department	phoneNo	address	roomId
101	Arav Mehta	bca	computer science	9632587412	block a, room 204	1
102	Priya Sharma	bba	business admin	9874563210	block b, room 105	2
103	Rohan Singh	b.tech	mechanical	1123654789	block c, room 309	3
104	Sneha Kapoor	b.sc	physics	7845691231	block a, room 110	1
105	Vikas Nair	bca	computer science	5879456321	block b, room 202	2

```
5 rows in set (0.02 sec)
```

```
mysql> select * from room;
```

roomId	floor	capacity	roomType	availability	charge	bedNo
1	1	2	NON AC	available	3000	2
2	2	3	AC	Full	4500	3
3	3	1	NON AC	available	2500	1
4	1	2	AC	Full	4000	2

```
4 rows in set (0.00 sec)
```

```
mysql> select * from fee;
```

id	studId	totalFee	paid	status
1	101	3000	3000	paid
2	102	4500	2500	due
3	103	2500	2500	paid
4	104	3000	3000	paid
5	105	4500	4000	due

```
5 rows in set (0.01 sec)
```

```
mysql> select * from complaints;
```

complaintId	studId	reletedTo	description	status
1	102	hostel	ac not working	pending
2	103	water	water leakage	resolved
3	105	hostel	no light in bathroom	pending

```
3 rows in set (0.00 sec)
```

LEARNING OUTCOMES

Through this project, the following key skills and concepts were gained:

- Understanding of **relational database design** using MySQL.
- Practical knowledge of **SQL operations**: CREATE, INSERT, UPDATE, DELETE, and ALTER.
- Use of **primary and foreign keys** to enforce data relationships.
- Implementation of **check constraints, default values** etc.
- Experience in building a scalable and structured backend system.

CONCLUSION

The backend College Management System successfully fulfills the core requirements of a database-driven administrative system. It efficiently manages student records, room assignments, complaints, and fee details. Though it lacks a front-end interface, its robust and scalable design lays the groundwork for future expansion into a complete application. This project has provided valuable experience in relational database development and backend logic design, forming a strong base for more advanced software development tasks.

REFERENCES

- <https://www.w3schools.com>
- <https://www.geeksforgeeks.org>