

# **DBMS MINI PROJECT REPORT**

**NAME: AKSSHAYA RAVIKUMAR**

**SRN: PES2UG20CS035**

**SECTION: A**

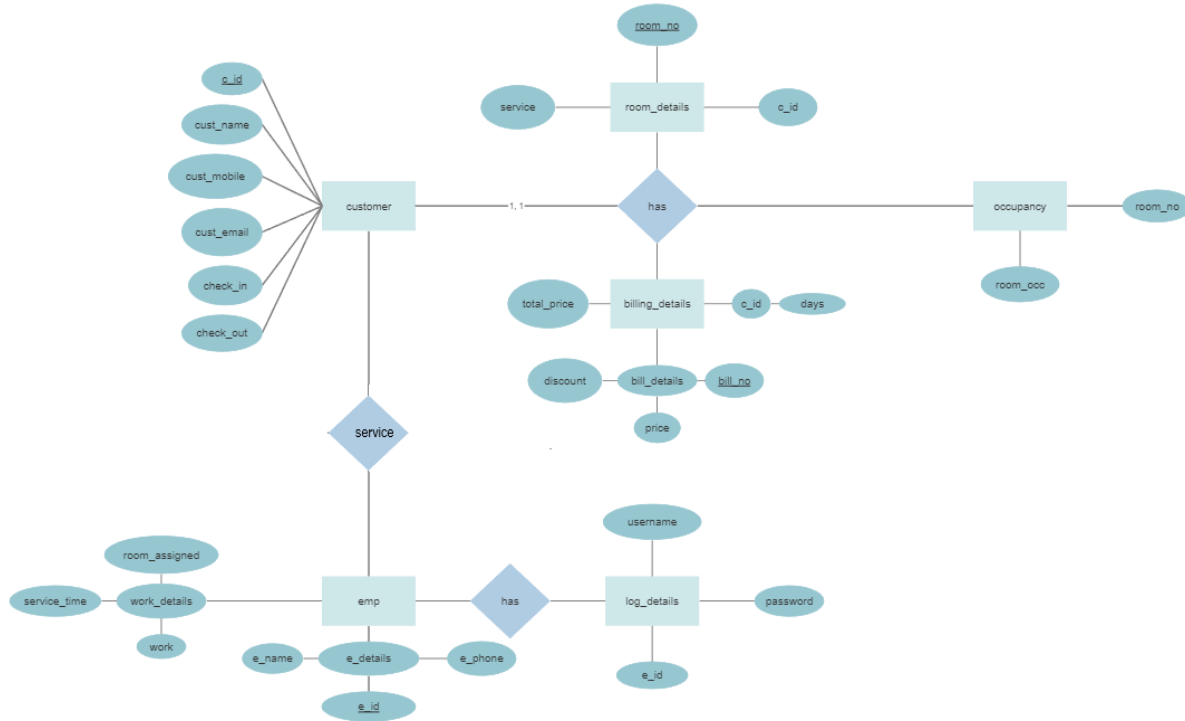
## **TOPIC: Hotel Management System**

### **1. A short description about the project and scope**

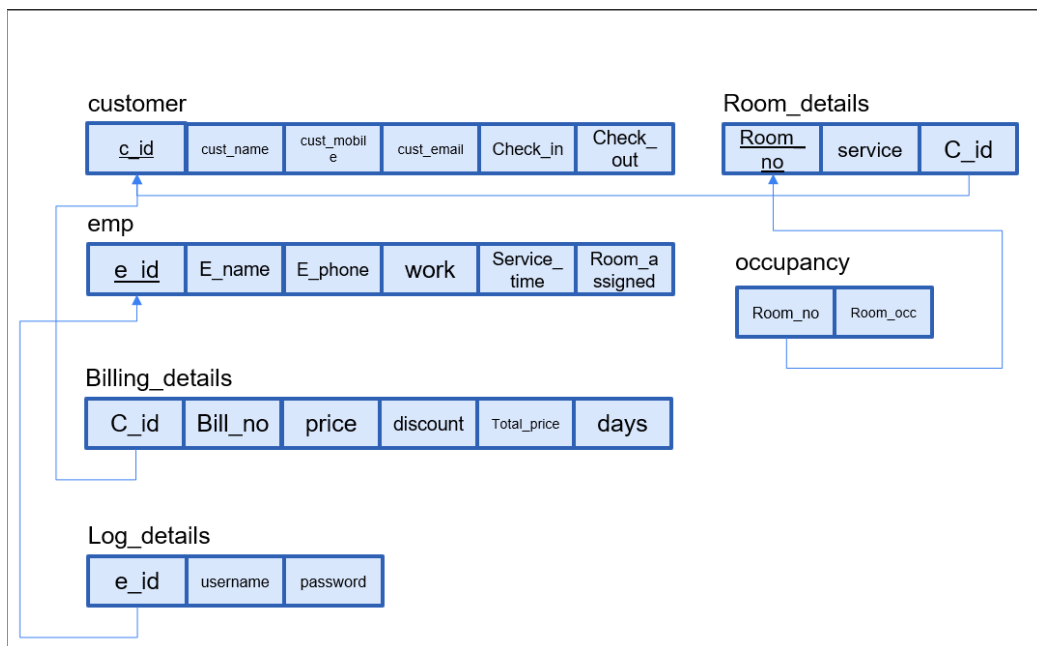
Hotel management system will be helpful in managing hotels and reduce the workload of employees.

The focus of this project is to make the Hotel Management system less complex than it typically is by using sectional design and a constructed system which is very much user oriented.

## 2. ER Diagram



## 3. Relational Scheme



## 4. DDL statements to build the database

Create database hotel;

Use hotel;

```
create table customer(  
    c_id int primary key not null,  
    cust_name varchar(50) not null,  
    cust_mobile int,  
    cust_email varchar(50),  
    check_in datetime not null,  
    check_out datetime not null  
);
```

```
MariaDB [project]> desc customer;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| c_id       | int(11)       | NO   | PRI | NULL    |       |  
| cust_name  | varchar(50)   | NO   |     | NULL    |       |  
| cust_mobile| int(11)       | YES  |     | NULL    |       |  
| cust_email | varchar(50)   | YES  |     | NULL    |       |  
| check_in   | datetime      | NO   |     | NULL    |       |  
| check_out  | datetime      | NO   |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
6 rows in set (0.033 sec)
```

```
create table room_details(  
    room_no int primary key not null,  
    c_id int not null,  
    r_service varchar(50)  
);
```

```
MariaDB [project]> desc room_detail;
```

Field	Type	Null	Key	Default	Extra
room_no	int(11)	NO	PRI	NULL	
c_id	int(11)	YES		NULL	
r_service	varchar(50)	YES		NULL	

3 rows in set (0.034 sec)

```
create table occupancy(  
    room_no int not null,  
    constraint rm foreign key (room_no) references room_details(room_no),  
    room_occ varchar(50) not null  
);
```

```
MariaDB [project]> desc occupancy;
```

Field	Type	Null	Key	Default	Extra
room_no	int(11)	NO	MUL	NULL	
room_occ	varchar(50)	NO		NULL	

2 rows in set (0.043 sec)

```
create table emp(  
    e_id int not null primary key,  
    e_name varchar(50) not null,  
    e_phone int,  
    room_assigned int,  
    service_time time,  
    work varchar(80)  
);
```

```
MariaDB [project]> desc emp;
```

Field	Type	Null	Key	Default	Extra
e_id	int(11)	NO	PRI	NULL	
e_name	varchar(50)	NO		NULL	
e_phone	int(11)	YES		NULL	
room_assigned	int(11)	YES		NULL	
service_time	time	YES		NULL	
work	varchar(80)	YES		NULL	

6 rows in set (0.034 sec)

```
create table billing(  
    bill_no int primary key,  
    c_id int,  
    days int,  
    price int not null,  
    discount int,  
    total_price int not null,  
    paid varchar(10),  
    constraint cid foreign key(c_id) references customer(c_id)  
);
```

```
MariaDB [project]> desc billing;
```

Field	Type	Null	Key	Default	Extra
bill_no	int(11)	NO	PRI	NULL	
c_id	int(11)	YES		NULL	
days	int(11)	YES		NULL	
price	int(11)	NO		NULL	
discount	int(11)	YES		NULL	
total_price	int(11)	NO		NULL	
paid	varchar(10)	YES		NULL	

7 rows in set (0.025 sec)

```
create table log_details(  
    username varchar(20) not null,  
    password varchar(20) not null,  
    e_id int,  
    constraint eid foreign key (e_id) references emp(e_id)
```

);

```
MariaDB [project]> desc log_details;
```

Field	Type	Null	Key	Default	Extra
username	varchar(20)	NO		NULL	
password	varchar(20)	NO		NULL	
e_id	int(11)	YES	MUL	NULL	

3 rows in set (0.033 sec)

## 5. Different methods used to populate data. - Show statements used under different methods

### Login:

Employee enters username , password, employee id to get access.

```
INSERT INTO log_details VALUES(emp1,1,1),(emp2,2,2),(emp3,3,3);
```

### Check-in new customer data:

New customer data into table **customer** is entered during check-in.

```
INSERT INTO customer VALUES  
(1,'a',12345,2022-11-24 00:00:00,2022-11-25 20:00:00),  
(2,'b',34567,2022-11-24 02:00:00,2022-11-25 20:00:00),  
(3,'c',53243,2022-11-24 10:00:00,2022-11-26 03:00:00),  
(4,'d',99888,2022-11-24 09:00:00,2022-11-27 20:00:00),  
(5,'e',45767,2022-11-24 06:00:00,2022-11-29 07:00:00);
```

### Check-out and billing:

Occupancy of customer and room is changed and billing is processed.

```
INSERT INTO billing VALUES(1,1,1,1000,10,900,'no');
```

### Room details:

Inserts room data to table room\_detail .

```
INSERT INTO room_detail VALUES(100,1,'room service'),  
(200,2,'food'),  
(300,3,'food');
```

```
INSERT INTO occupancy VALUES(100,'yes');
```

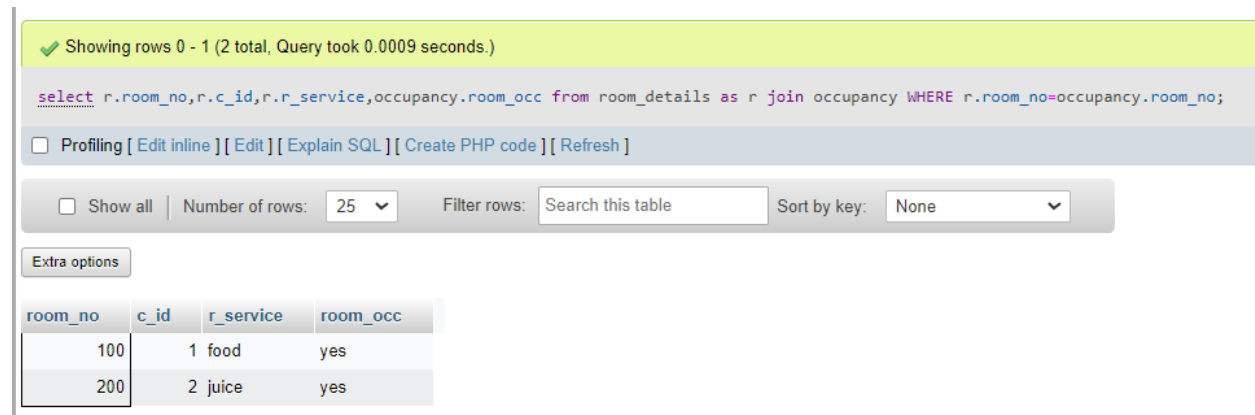
### Employee details:

Inserts employee data to table **emp**.

```
INSERT INTO emp VALUES  
(1,'emp1',12345,100,20:00:00,'room service'),  
(2,'emp2',34521,200,20:00:00,'food'),  
(3,'emp3',99787,300,20:00:00,'food');
```

## 6. JOIN queries

```
select r.room_no,r.c_id,r.r_service,occupancy.room_occ from room_details  
as r join occupancy WHERE r.room_no=occupancy.room_no;
```



Showing rows 0 - 1 (2 total, Query took 0.0009 seconds.)

```
select r.room_no,r.c_id,r.r_service,occupancy.room_occ from room_details as r join occupancy WHERE r.room_no=occupancy.room_no;
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

room_no	c_id	r_service	room_occ
100	1	food	yes
200	2	juice	yes

```
select * from emp inner join room_detail where  
emp.room_assigned=room_detail.room_no;
```

✓ Showing rows 0 - 2 (3 total, Query took 0.0008 seconds.)

```
select * from emp inner join room_detail where emp.room_assigned=room_detail.room_no;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows: 25 ▼ Filter rows:  Sort by key: None ▼

Extra options

e_id	e_name	e_phone	room_assigned	service_time	work	room_no	c_id	r_service
1	emp1	999999	100	14:34:11	room service	100	1	room service
2	emp2	988888	200	21:09:00	food	200	2	food
3	emp3	987788	300	21:09:00	food	300	3	food

☐ Show all | Number of rows: 25 ▼ Filter rows:  Sort by key: None ▼

Query results operations

## 7. Aggregate Functions

SELECT count(\*) from customer;

✓ Showing rows 0 - 0 (1 total, Query took 0.0006 seconds.)

```
SELECT count(*) from customer;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows: 25 ▼ Filter rows:

Extra options

count(*)
5

SELECT MAX(total\_price) from billing;



✓ Showing rows 0 - 0 (1 total, Query took 0.0007 seconds.)

```
SELECT MAX(total_price) from billing;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows: 25 ▼ Filter rows:

Extra options

MAX(total_price)
900

## 8. SET Operators

### INTERSECT:

SELECT \* FROM room\_detail WHERE room\_no IN (SELECT room\_assigned FROM emp);

✓ Showing rows 0 - 2 (3 total, Query took 0.0014 seconds.)

```
SELECT * FROM room_detail WHERE room_no IN (SELECT room_assigned FROM emp);
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows: 25 ▼ Filter rows:

Extra options

	room_no	c_id	r_service
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	100	1	room service
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	200	2	food
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	300	3	food

SELECT \* FROM customer WHERE c\_id IN (SELECT c\_id FROM room\_detail);

✓ Showing rows 0 - 3 (4 total, Query took 0.0008 seconds.)

```
SELECT * FROM customer WHERE c_id IN (SELECT c_id FROM room_detail);
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows: 25 ▼ Filter rows:  Sort by key: None ▼

Extra options

	c_id	cust_name	cust_mobile	cust_email	check_in	check_out
<input type="checkbox"/> Edit Copy Delete	1	AC	0		2022-11-24 00:00:00	2022-11-24 00:00:00
<input type="checkbox"/> Edit Copy Delete	2	bcd	234567891	bcd@gmail.com	2022-11-22 00:00:00	2022-11-24 00:00:00
<input type="checkbox"/> Edit Copy Delete	3	c	846673	c@gmail.com	2022-11-23 08:03:33	2022-11-26 12:33:33
<input type="checkbox"/> Edit Copy Delete	5	e	0	e@gmail.com	2022-11-23 00:00:00	2022-11-25 00:00:00

## 9. Functions and Procedure

### FUNCTIONS:

Calculate total price with number of days as parameter.

DELIMITER \$\$

CREATE FUNCTION **total**(days INT) RETURNS int

BEGIN

    return 1000\*(1-0.1)\*days;

END \$\$

DELIMITER ;

✓ Your SQL query has been executed successfully.

```
SET @p0='2'; SELECT `total`(@p0) AS `total`;
```

Execution results of routine `total`

total





1800

## Routines

☐ Check all

 Export

 Drop

	Name	Type	Returns				
<input type="checkbox"/>	total	FUNCTION	int(11)	 Edit	 Execute	 Export	 Drop

## PROCEDURES:

**Assign a room to customer who checks in.**

DELIMITER &&

CREATE PROCEDURE **assign\_room** (IN var1 INT)

BEGIN

    SELECT room\_no INTO @room from room\_detail where c\_id is NULL LIMIT 1;

    UPDATE room\_detail SET c\_id=var1 where room\_no=@room;

    UPDATE occupancy SET room\_occ='yes' WHERE room\_no=@room;

END &&

DELIMITER ;

✓ Your SQL query has been executed successfully.  
0 rows affected by the last statement inside the procedure.

```
SET @p0='3'; CALL `assign_room`(@p0);
```

Execution results of routine `assign\_room`

⚠ MySQL returned an empty result set (i.e. zero rows).

## Routines ⓘ

☐ Check all  Export  Drop

	Name	Type	Returns				
<input type="checkbox"/>	assign_room	PROCEDURE		 Edit	 Execute	 Export	 Drop
<input type="checkbox"/>	bill	PROCEDURE		 Edit	 Execute	 Export	 Drop
<input type="checkbox"/>	check_out	PROCEDURE		 Edit	 Execute	 Export	 Drop
<input type="checkbox"/>	emp_work	PROCEDURE		 Edit	 Execute	 Export	 Drop

**Update room\_detail to update work and room assigned to an employee.**

DELIMITER &&

CREATE PROCEDURE **emp\_work** (IN var1 INT)

BEGIN

    SELECT work,room\_assigned INTO @work,@room\_no from emp where e\_id=1 LIMIT 1;

    UPDATE room\_detail SET r\_service=@work where room\_no=@room;

END &&

DELIMITER ;

✓ Your SQL query has been executed successfully.  
0 rows affected by the last statement inside the procedure.

```
SET @p0='2'; CALL `emp_work`(@p0);
```

Execution results of routine `emp\_work`

⚠ MySQL returned an empty result set (i.e. zero rows).

### Calculate and Insert bill information into billing table.

DELIMITER &&

CREATE PROCEDURE **bill** (IN c\_id INT,IN days INT,IN total INT)

BEGIN

    SELECT max(bill\_no) INTO @bill\_no FROM billing;

    INSERT INTO billing VALUES(@bill\_no +1,c\_id,days,1000,10,total);

    SELECT total;

END &&

DELIMITER ;

**Execute routine**

**Routine parameters**

Name	Type	Function	Value
c_id	INT	<input type="text" value="2"/>	2
days	INT	<input type="text" value="3"/>	3
total	INT	<input type="text" value="900"/>	900

**Error Message:**  
 The following query has failed: "SET @p0='2'; SET @p1='3'; SET @p2='900'; CALL 'bill'(@p0, @p1, @p2);"  
 MySQL said: #1644 - pay bill

### Customer check out calculate bill.

DELIMITER &&

CREATE PROCEDURE **check\_out**(c\_id INT)

BEGIN

SELECT check\_in,check\_out INTO @check\_in,@check\_out FROM customer WHERE  
 c\_id=c\_id LIMIT 1;

SELECT datediff(@check\_out,@check\_in) INTO @days;

SELECT total(@days) INTO @total;

CALL bill(c\_id,@days,@total);

END &&

DELIMITER ;

**❗** The following query has failed: "SET @p0='2';  
CALL `check\_out`(@p0); "  
  
MySQL said: #1644 - pay bill

Execute routine `check\_out`☐

Routine parameters

Name	Type	Function	Value
c_id	INT	<input type="text" value=""/>	<input type="text" value="2"/>

## 10. Triggers and cursors

### TRIGGERS:

DROP TABLE IF EXISTS reminders;

```
CREATE TABLE reminders (  
  c_id int primary key,  
  message VARCHAR(255) NOT NULL  
);
```

**Update message to remainder table when customer mobile is not inserted to customer table.**

```

DELIMITER $$
CREATE TRIGGER after_cus_insert
AFTER INSERT
ON customer FOR EACH ROW
BEGIN
    IF NEW.cust_mobile IS NULL THEN
        INSERT INTO reminders(c_id, message)
        VALUES(new.c_id,CONCAT('Hi ', NEW.c_id, ', please enter your mobile number.));
    END IF;
END$$
DELIMITER ;

```

✓ Showing rows 0 - 1 (2 total, Query took 0.0008 seconds.)

```
SELECT * FROM `reminders`
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 ▼ Filter rows:

Extra options

	c_id	message
<input type="checkbox"/> Edit Copy Delete	3	Hi 3, please pay your bills.
<input type="checkbox"/> Edit Copy Delete	4	Hi 4, please enter your mobile number.

**Show error if paid is no in billing table after insert.**

```

DELIMITER $$
CREATE TRIGGER after_bill_insert
AFTER INSERT
ON billing FOR EACH ROW
BEGIN
    IF NEW.paid='no' THEN
        SET error_msg="Train compartment exceeds more than 4";
        signal SQLSTATE '45000'
        set MESSAGE_TEXT =error_msg;
    END IF;
END $$
DELIMITER ;

```



**Execute routine**

**Routine parameters**

Name	Type	Function	Value
c_id	INT	<input type="text"/>	<input type="text" value="2"/>
days	INT	<input type="text"/>	<input type="text" value="3"/>
total	INT	<input type="text"/>	<input type="text" value="900"/>

**Error Message:**  
 The following query has failed: "SET @p0='2'; SET @p1='3'; SET @p2='900'; CALL 'bill'(@p0, @p1, @p2);"  
 MySQL said: #1644 - pay bill

## CURSORS:

```
def view_all_customer():
    c.execute('SELECT * FROM customer')
    customers = c.fetchall()
    return customers

def view_all_emp():
    c.execute('SELECT * from emp')
    emp = c.fetchall()
    return emp
```

## 11. Higher Level Programming

CHECK IN AND QUERY BOX:

Customer Menu

Check in

Employee Menu

show employees

Enter customer Details:

Customer ID:

0.00

-

+

check in:

2022/11/24

Customer Name:

check out:

2022/11/24

Customer mobile:

0.00

-

+

Customer Email

Add Customer

Query:

sql statement:

desc customer;

submit

('c\_id', 'int(11)', 'NO', 'PRI', None, '')

('cust\_name', 'varchar(50)', 'NO', '', None, '')

('cust\_mobile', 'int(11)', 'YES', '', None, '')

('cust\_email', 'varchar(50)', 'YES', '', None, '')

('check\_in', 'datetime', 'NO', '', None, '')

('check\_out', 'datetime', 'NO', '', None, '')

## VIEW CUSTOMER:

Customer Menu

View customers

Employee Menu

show employees

Hotel Management System

View customers Details:

View all Customers

	c_id	cust_name	cust_mobile	cust_email	check_in	check_out
0	1	abc	12,345.0000	abc@gmail.com	2022-11-22T04:49:15	2022-11-23T09:19:15
1	2	bcd	234,567,891.0000	bcd@gmail.com	2022-11-22T00:00:00	2022-11-24T00:00:00
2	3	c	846,673.0000	c@gmail.com	2022-11-23T08:03:33	2022-11-26T12:33:33
3	4	d	<NA>	<NA>	2022-11-23T09:32:42	2022-11-29T14:02:42
4	5	e	0.0000	e@gmail.com	2022-11-23T00:00:00	2022-11-25T00:00:00

Employee Details:

View all Employees

## EDIT CUSTOMER:

Customer Menu

Edit customers

Employee Menu

show employees

Edited customers Details:

View all Customers

	c_id	cust_name	cust_mobile	cust_email	check_in	check_out
0	1	abc	12,345,0000	abc@gmail.com	2022-11-22T04:49:15	2022-11-23T09:19:15
1	2	bcd	234,567,891,0000	bcd@gmail.com	2022-11-22T00:00:00	2022-11-24T00:00:00
2	3	c	846,673,0000	c@gmail.com	2022-11-23T08:03:33	2022-11-26T12:33:33
3	4	d	<NA>	<NA>	2022-11-23T09:32:42	2022-11-29T14:02:42
4	5	e	0.0000	e@gmail.com	2022-11-23T00:00:00	2022-11-25T00:00:00

Customer to Edit

1.00

0 : [

0 : 1

1 : "abc"

2 : 12345

3 : "abc@gmail.com"

4 : "datetime.datetime(2022, 11, 22, 4, 49, 15)"

5 : "datetime.datetime(2022, 11, 23, 9, 19, 15)"

]

Customer Name:

AC

check in:

2022/11/24

Customer mobile:

0.00

check out:

2022/11/24

Customer Menu

Edit customers

Employee Menu

show employees

Customer Name:

AC

check in:

2022/11/24

Customer mobile:

0.00

check out:

2022/11/24

Customer Email:

Update Customer

Successfully updated

Updated data

	c_id	cust_name	cust_mobile	cust_email	check_in	check_out
0	1	AC	0.0000		2022-11-24T00:00:00	2022-11-24T00:00:00
1	2	bcd	234,567,891,0000	bcd@gmail.com	2022-11-22T00:00:00	2022-11-24T00:00:00
2	3	c	846,673,0000	c@gmail.com	2022-11-23T08:03:33	2022-11-26T12:33:33
3	4	d	<NA>	<NA>	2022-11-23T09:32:42	2022-11-29T14:02:42
4	5	e	0.0000	e@gmail.com	2022-11-23T00:00:00	2022-11-25T00:00:00

Employee Details:

View all Employees

DELETE CUSTOMER:

Customer Menu

delete customer

Employee Menu

show employees

## Hotel Management System

### Delete customer Details:

Current data

c\_id to delete

1.00

Do you want to delete ::1.0

Delete customer

Updated data

### Employee Details:

View all Employees

CHECK OUT:

Customer Menu

check out

## Hotel Management System

### check out:

customer id:

2.00

ROOM DETAILS:

Query:

sql statement:

submit

### Room Details:

Room details

	room_no	c_id	r_service	occupancy
0	100	1	food	yes
1	200	2	juice	yes

Show rooms available

ROOM AVAILABILITY:

Show rooms available

View all Rooms Available



	rooms
0	600

## BILLING DETAILS:

### Billing Details:

customer id:

1.00

