

Real Time
Assignment
KSCo

A Project Report Submitted by

Chandeep G
(19CSR020)
Charanraj T
(19CSR021)
Cibikumar M V
(19csr022)

For the completion of course
Database Management Systems Laboratory

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMMUNICATION AND COMPUTER SCIENCES



KONGU ENGINEERING COLLEGE (Autonomous)

PERUNDURAI, ERODE – 638060

Abstract

This program is about maintaining a database for a law firm company. The database includes a table for Clients, Employees, Designation and Cases and extra tables are created for normalizing. The application is developed using python and MySql. Client and Employee table store their respective details. Each client is assigned to a particular employee who's name is stored in the client table. Case details are stored in case tables along with who they belong to.

Admin of the company can use it to insert, delete, view or update the records. Deleting a employee sets on delete cascade which sets the client currently assinged to the deleted employee to unassingned state. The status of a case, client's assigned lawyer and a employee's designation can all be changed.

CONVERSION OF ER TO SCHEMA

Step-1:

Converting relationship into relations.

Client:

Client ID	First Name	Last Name	Door No.	Street Name	Pincode
-----------	------------	-----------	----------	-------------	---------

Employee:

Emp ID	First Name	Last Name	Door No.	Street Name	Pincode
--------	------------	-----------	----------	-------------	---------

Case:

Case ID	Case Title	Date	Status	Fee
---------	------------	------	--------	-----

Position:

Designation	Salary
-------------	--------

Step-2:

Converting weak entities.

There are no weak entities.

Step-3:

Converting 1:1 relationship into relations.

There are no 1:1 relations.

Step-4:

Converting 1:N relationship into relations.

Adding foreign key to tables,

Client:

Client ID	Emp ID	First Name	Last Name	Door No.	Street Name	Pincode

Employee:

Emp ID	Designation	First Name	Last Name	Door No.	Street Name	Pincode

Case:

Case ID	Emp ID	Case Title	Date	Status	Fee

Step-5:

Converting N:M relationship into relations.

Involved In:

Client ID	Case ID

Step-6:

Converting Multivalued attribute into relations.

Employee Contact:

Emp ID	Mobile No.
--------	------------

Client Contact:

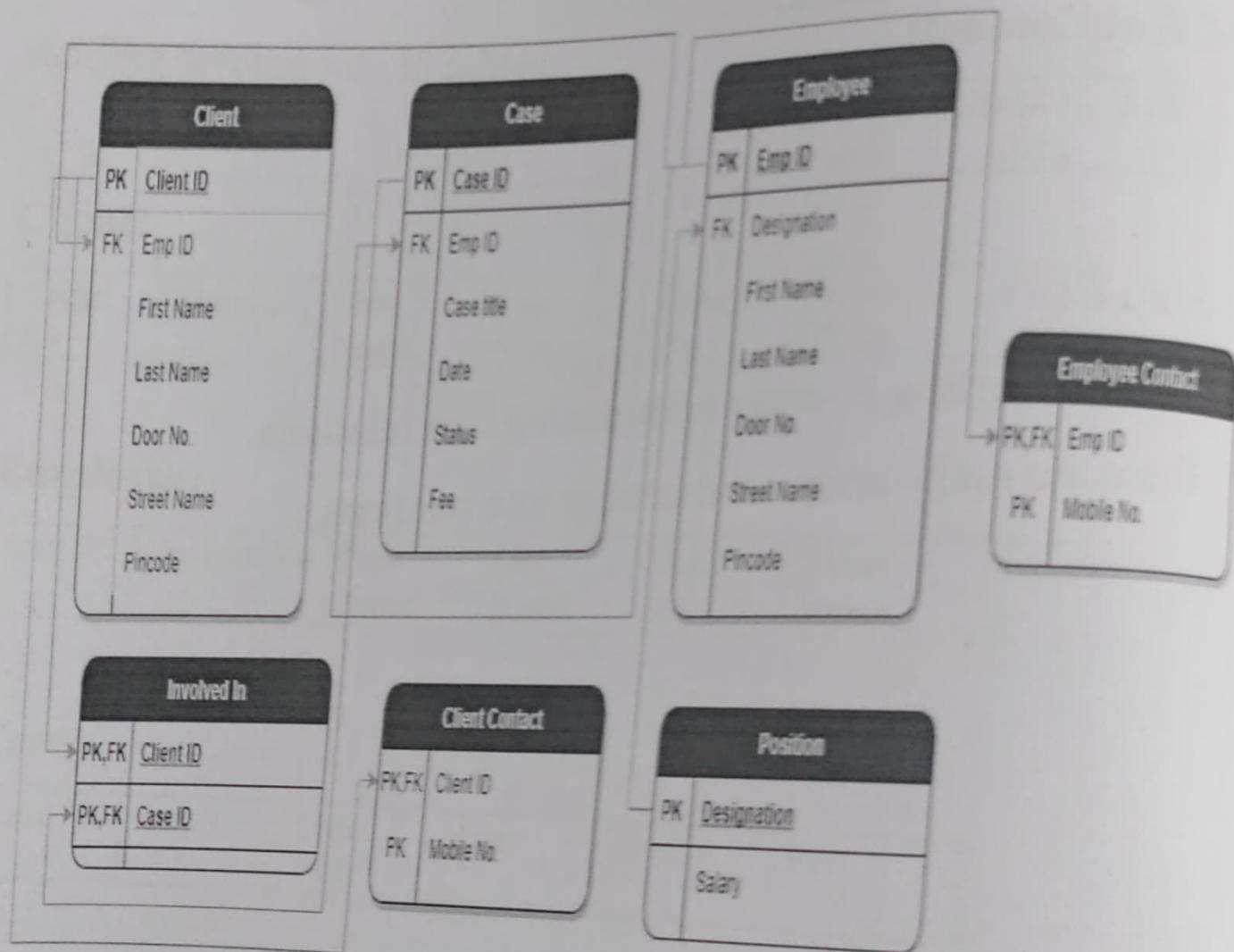
Client ID	Mobile NO.
-----------	------------

Step-7:

Converting N-array relationship into relations.

There are no N-array relationship into relations.

SCHEMA



OPERATIONS

- Choices:

```
1-Insert employee  
2-Insert client  
3-Insert case  
4-Update details  
5-Display options  
6-Delete employee  
7-Exit
```

- Inserting Employee:

```
1-Insert employee  
2-Insert client  
3-Insert case  
4-Update details  
5-Display options  
6-Delete employee  
7-Exit  
Enter choice:1  
  
Enter name  
First Name:charan  
Last Name:raj  
Enter address  
Door no:6  
Street name:kse street,y city  
Pin code:654789  
Enter designation:intern  
Enter primary phone no:8825850137  
Press 0 if no secondary no  
Enter secondary phone no:0  
employee inserted successfully!
```

• Deleting a Employee:

```
1-Insert employee  
2-Insert client  
3-Insert case  
4-Update details  
5-Display options  
6>Delete employee  
7-Exit  
Enter choice:6  
Enter employee id:4  
Employee deleted successfully!
```

• Viewing Tables:

Employee Table:

```
1.employee details  
2.client details  
3.case details  
Enter your choice:1  
Id      Name          Address           Designation    Contact1      Contact2  
1       cibi kumar    5,shd street,x city,695847    senior        8456213790    9632587410  
2       charan raj    6,kse street,y city,654789    intern        8825850137
```

Cases Table:

```
1.employee details  
2.client details  
3.case details  
Enter your choice:3  
Id      Client        Employee        Case name      Status     Fee  
1       ram kumar     cibi kumar     accident      open      5000
```

• Updating Case Status:

```
1.case status  
2.employee designation  
3.assign employee  
Enter your choice:1  
Enter case ID to update:1  
Enter status:closed  
Updation successfull!
```

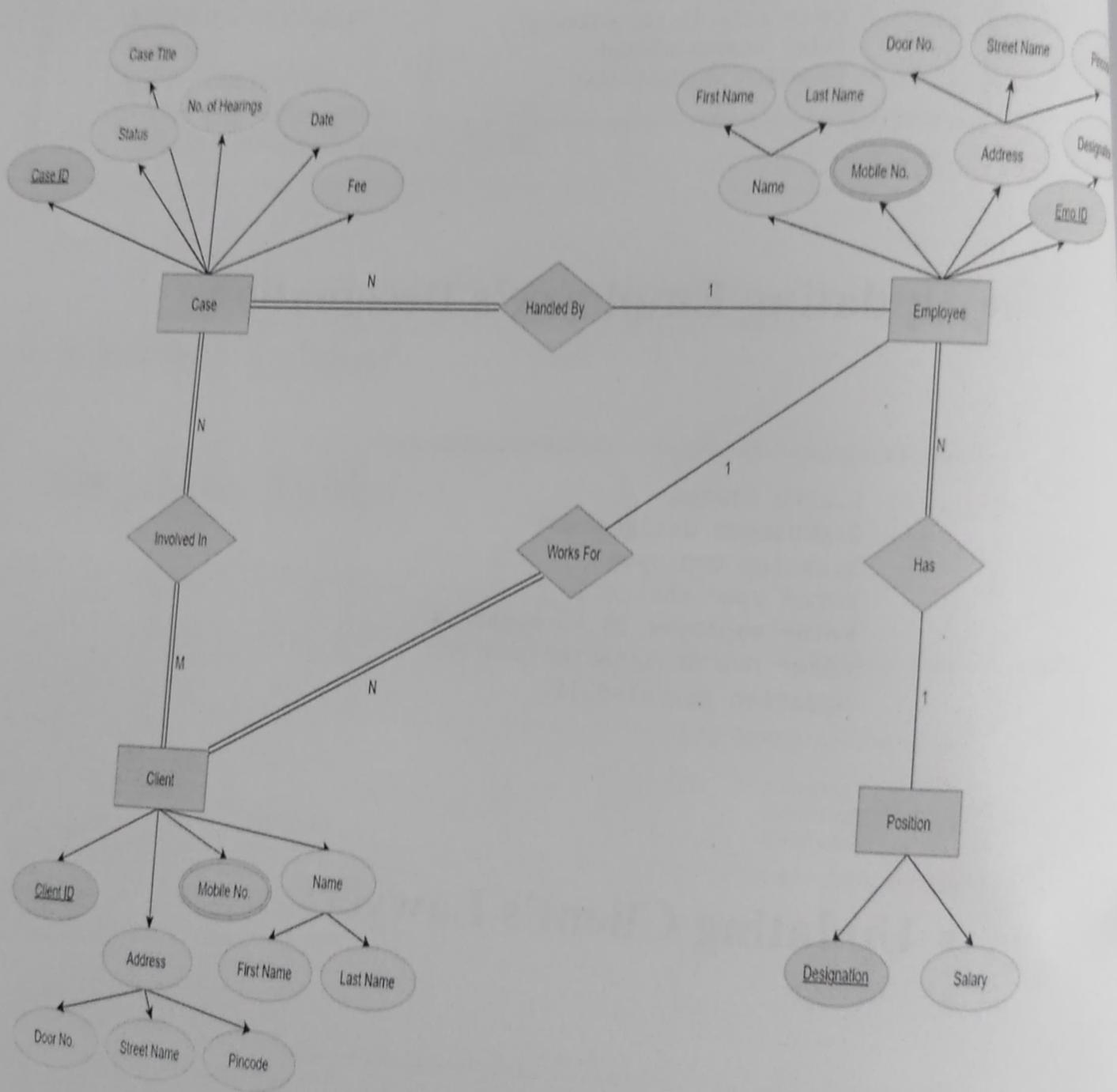
• Updating Employee's Designation:

```
1.case status  
2.employee designation  
3.assign employee  
Enter your choice:2  
Enter employee ID to update:4  
Enter new designation:senior  
Updation successfull!
```

• Updating Client's Lawyer:

```
1.case status  
2.employee designation  
3.assign employee  
Enter your choice:3  
Enter client ID to update:3  
Enter new employee id:2  
Updation successfull!
```

ER DIAGRAM



• Inserting Client:

```
1-Insert employee
2-Insert client
3-Insert case
4-Update details
5-Display options
6>Delete employee
7-Exit
Enter choice:2

Enter name
First Name:sathish
Last Name:j
Enter address
Door no:3
Street name:mew street,y city
Pin code:654789
Enter assigned employee id:3
Enter primary phone no:9867534210
Press 0 if no secondary no
Enter secondary phone no:7481592603
client inserted successfully!
```

• Inserting Case:

```
1-Insert employee
2-Insert client
3-Insert case
4-Update details
5-Display options
6>Delete employee
7-Exit
Enter choice:3

Enter client id:1
Enter case title:property debate
Enter date(yyyy-mm-dd):2021-05-20
Enter assigned employee id:2
Enter fees:50000
case inserted successfully!
```

Client Table:

1.employee details

2.client details

3.case details

Enter your choice:2

Id	Name
1	chandeep g
2	sathish j
3	venkat m

Address

4,ken street,x city,695847
3,mew street,y city,654789
9,dke street,z city,654987

Employee
cibi
charan
saran

Contact1	Contact2
8475916320	7481592603
9382714560	9571302486

9867534210

'B' - section

Tourism Management System

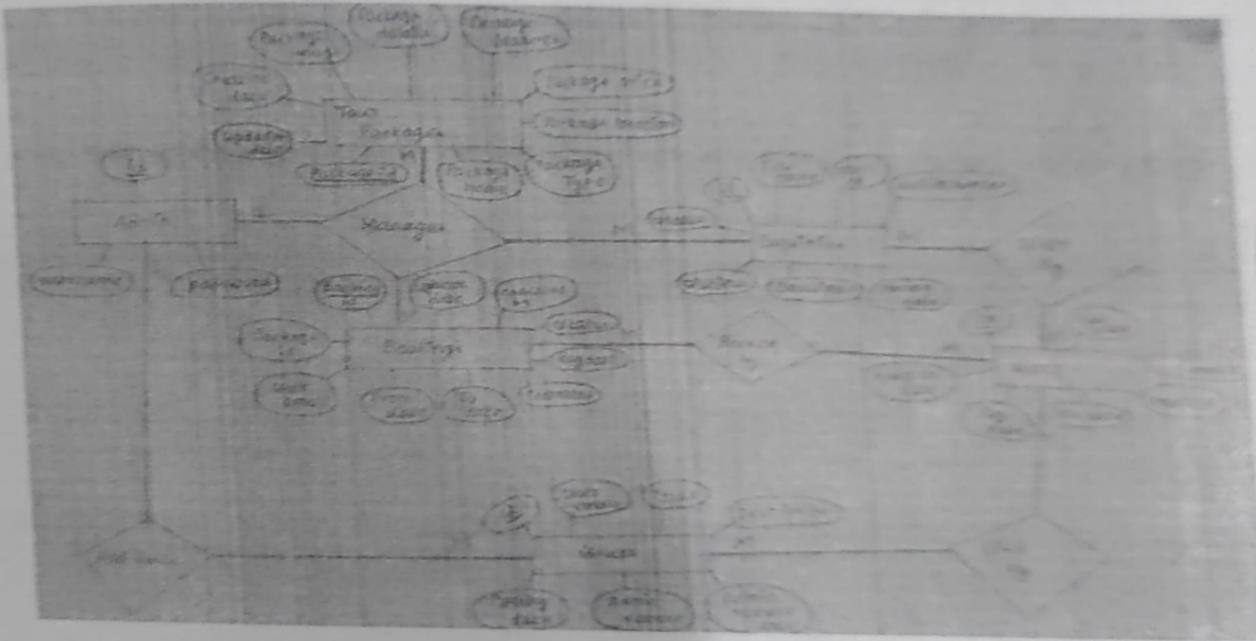
DBMS Project Presentation

BY Janani N -19CSR069

Krithika Tharani A S – 19CSR093



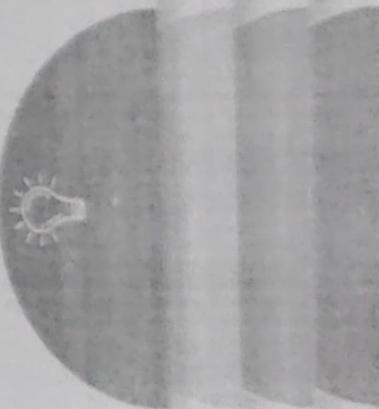
ER DIAGRAM



Tables Used

- *admin*
- *tblbooking*
- *tblenquiry*
- *tblissues*
- *tbltourpackages*
- *tblusers*

DDL QUERIES USED



Tblusers

```

CREATE TABLE `tblusers` (
    `id` int NOT NULL AUTO_INCREMENT,
    `FullName` varchar(100) DEFAULT NULL,
    `MobileNumber` char(10) DEFAULT NULL,
    `EmailId` varchar(70) DEFAULT NULL,
    `Password` varchar(100) DEFAULT NULL,
    `RegDate` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
    `UpdationDate` timestamp NULL DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
    PRIMARY KEY (`EmailId`),
);

```

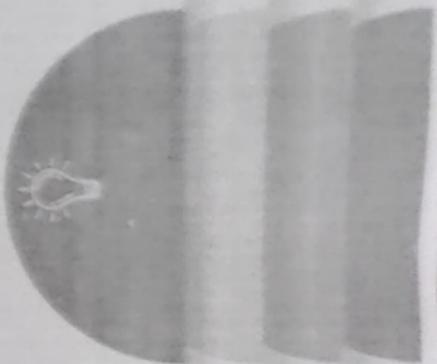


Id	FullName	MobileNumber	EmailId	Password	RegDate	UpdationDate
12	krithika	1234567890	123abc@gmail.com	e10adc3949ba59abbe56e057f20f883e	2021-03-25 20:02:49	NULL
13	krithika	1234567890	123@gmail.com	e10adc3949ba59abbe56e057f20f883e	2021-03-26 15:38:04	NULL
14	tharani	1234876509	abcd@gmail.com	e2fc714c4727ee9395f324cd2e7f331f	2021-05-18 19:14:46	2021-05-23 17:40:19
15	janani	9086578904	janu@gmail.com	3605ec021bac4a1db600450360a366a5	2021-05-22 16:53:13	NULL
17	NULL	NULL	NULL	d41d8cd98f00b204e9800998ecf8427e	2021-05-23 17:46:44	NULL

Admin

```
CREATE TABLE `admin` (
  `id` int NOT NULL AUTO_INCREMENT,
  `UserName` varchar(100) DEFAULT NULL,
  `Password` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id`)
);
```

id	UserName	Password
1	admin	21232f297a57a5a743894a0e4a801fc3



Tblbooking

```
CREATE TABLE `tblbooking` (
  `BookingId` int NOT NULL AUTO_INCREMENT,
  `PackageId` int DEFAULT NULL,
  `UserEmail` varchar(100) DEFAULT NULL,
  `FromDate` varchar(100) DEFAULT NULL,
  `ToDate` varchar(100) DEFAULT NULL,
  `Comment` varchar(100),
  `RegDate` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `status` int DEFAULT NULL,
  `CancelledBy` varchar(5) DEFAULT NULL,
  `UpdationDate` timestamp NULL DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`BookingId`),constraint fk foreign key (`PackageId`) references `tbltourpackages`(`PackageId`)
);
```

	BookingId	PackageId	UserEmail	FromDate	ToDate	Comment	RegDate	status	CancelledBy	UpdationDate
17	2	2	abcd@gmail.com	2021-05-23	2021-05-24	sample test	2021-05-23 18:06:37	2	a	2021-05-23 18:09:10
18	3	3	janu@gmail.com	2021-05-16	2021-05-18	test test	2021-05-23 18:07:49	0	NULL	NULL

tbltourpackages

```
CREATE TABLE `tbltourpackages` (
  `PackageId` int NOT NULL AUTO_INCREMENT,
  `PackageName` varchar(200) DEFAULT NULL,
  `PackageType` varchar(150) DEFAULT NULL,
  `PackageLocation` varchar(100) DEFAULT NULL,
  `PackagePrice` int DEFAULT NULL,
  `PackageFeatures` varchar(255) DEFAULT NULL,
  `PackageDetails` varchar(100),
  `PackageImage` varchar(100) DEFAULT NULL,
  `Creationdate` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `UpdateDate` timestamp NULL DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`PackageId`)
);
```

PackageId	PackageName	PackageType	PackageLocation	PackagePrice	PackageFeatures	PackageDetails	PackageImage	Creationdate	UpdateDate
1	Kerala Trip	General	Kulu Manali India	7000	N/AConditioning Balcony / Terrace Cabin / Staff..	"Lorem ipsum dolor sit amet, consectetur adipiscing.."	main1.jpg	2021-05-13 19:53:44	2021-05-23 17:36:47
2	Ladakh	Family	India	6400	"Lorem ipsum dolor sit amet, consectetur adipiscing.."	"Lorem ipsum dolor sit amet, consectetur adipiscing.."	Shrik_Inv_Ladakh.jpg	2021-05-13 08:54:28	2021-05-23 17:30:36
3	Mannarkudi Arunachal Temple	Family	Tamil Nadu, India	1000	Demo Demo Test	Sample Text for testing.	Meenakshi-Temple-II-134-qj.jpg	2021-05-13 21:30:58	2021-05-23 17:37:21
4	Kerala	Family and Couple	Kerala	2000	"voll esse illum dolore eu fugiat nulla pariatur .."	"voll esse illum dolore eu fugiat nulla pariatur .."	WhatsApp Image 2021-05-22 at 3:20:00 PM.jpg	2021-05-13 04:19:37	2021-05-23 17:37:33
5	Goorg Tour Packages	General	Goorg	2000	"voll esse illum dolore eu fugiat nulla pariatur .."	"voll esse illum dolore eu fugiat nulla pariatur .."	WhatsApp Image 2021-05-22 at 8:16:26 PM.jpg	2021-05-13 04:12:10	2021-05-23 17:38:02
6	Indonesia	Family	Indonesia	5000	Finesse w/ pickup and drop etc	"Test Test Test Test Test"	WhatsApp Image 2021-05-22 at 9:14:40 PM.jpg	2021-05-13 12:31:02	2021-05-23 17:38:26

Tblenquiry

```
CREATE TABLE `tblenquiry` (
  `id` int NOT NULL AUTO_INCREMENT,
  `FullName` varchar(100) DEFAULT NULL,
  `EmailId` varchar(100) DEFAULT NULL,
  `MobileNumber` char(10) DEFAULT NULL,
  `Subject` varchar(100) DEFAULT NULL,
  `Description` varchar(100),
  `PostingDate` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `Status` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  constraint fk1 forign key (`EmailId`) references `tblusers`(`EmailId`)
);
```

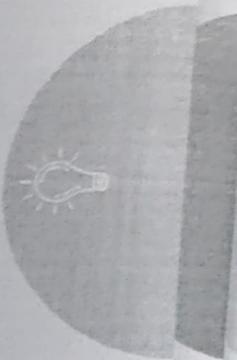
<i>id</i>	<i>FullName</i>	<i>EmailId</i>	<i>MobileNumber</i>	<i>Subject</i>	<i>Description</i>	<i>PostingDate</i>	<i>Status</i>
5	abc	abc@gmail.com	1234567890	availability	test test test test test test test test test ...	2021-05-23 14:08:24	1
6	jani	jani@gmail.com	1234567890	sample	test test 123	2021-05-23 17:44:12	1

Tblissues

```
CREATE TABLE `tblissues` (
  `id` int NOT NULL AUTO_INCREMENT,
  `UserEmail` varchar(100) DEFAULT NULL,
  `Issue` varchar(100) DEFAULT NULL,
  `Description` varchar(100),
  `PostingDate` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `AdminRemark` varchar(100),
  `AdminremarkDate` timestamp NULL DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`), constraint fk2 forign key (`UserEmail`) references tblusers(`Emailid`)
);
```

id	UserEmail	Issue	Description	PostingDate	AdminRemark	AdminremarkDate
12	abcd@gmail.com	Other	123	2021-05-23 14:12:10	sorry for the inconvenience	2021-05-23 14:13:14
13	abcd@gmail.com	Booking Issues	booking not accepted	2021-05-23 17:46:44	NULL	NULL

DML QUERIES USED



Insert Queries

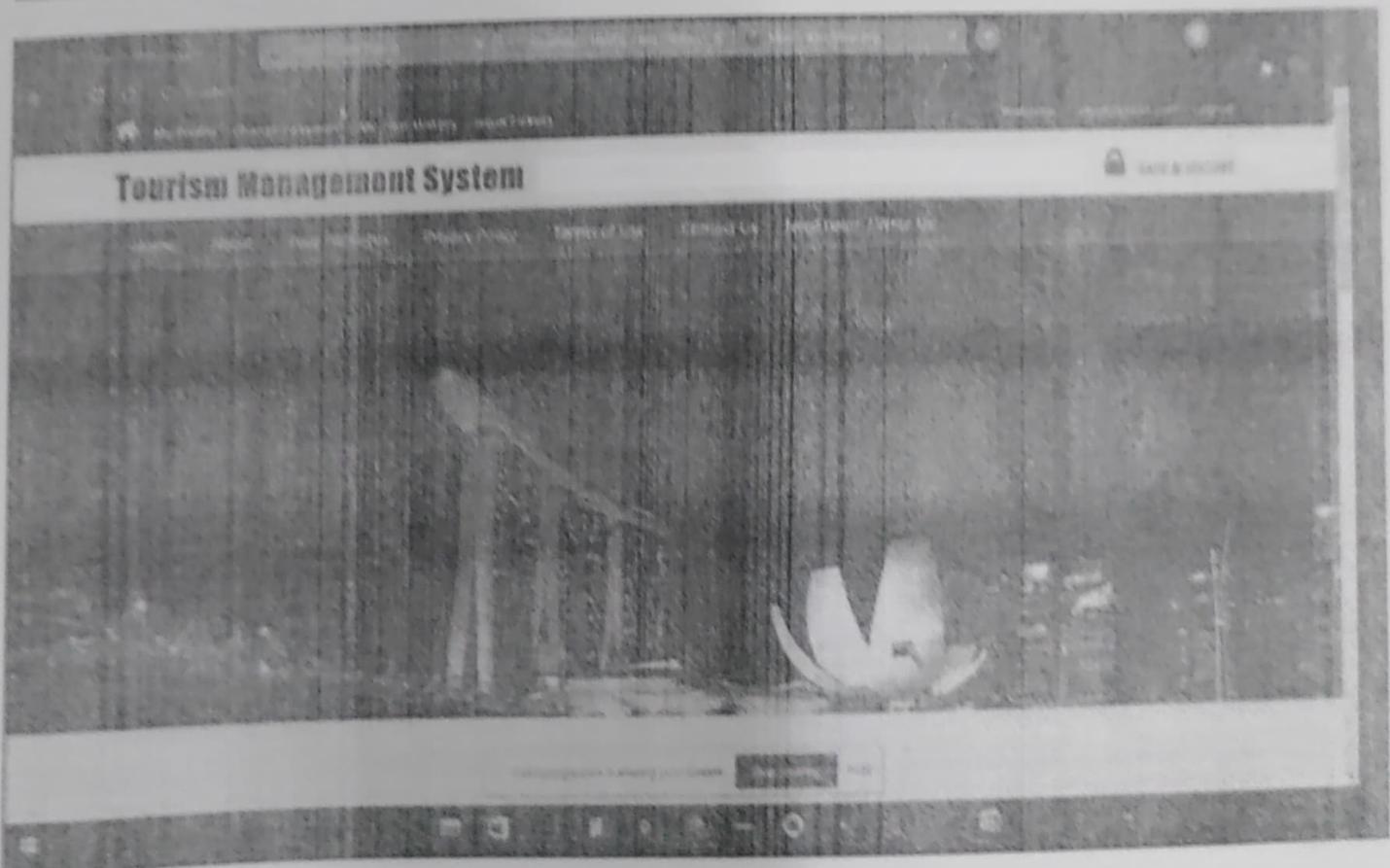
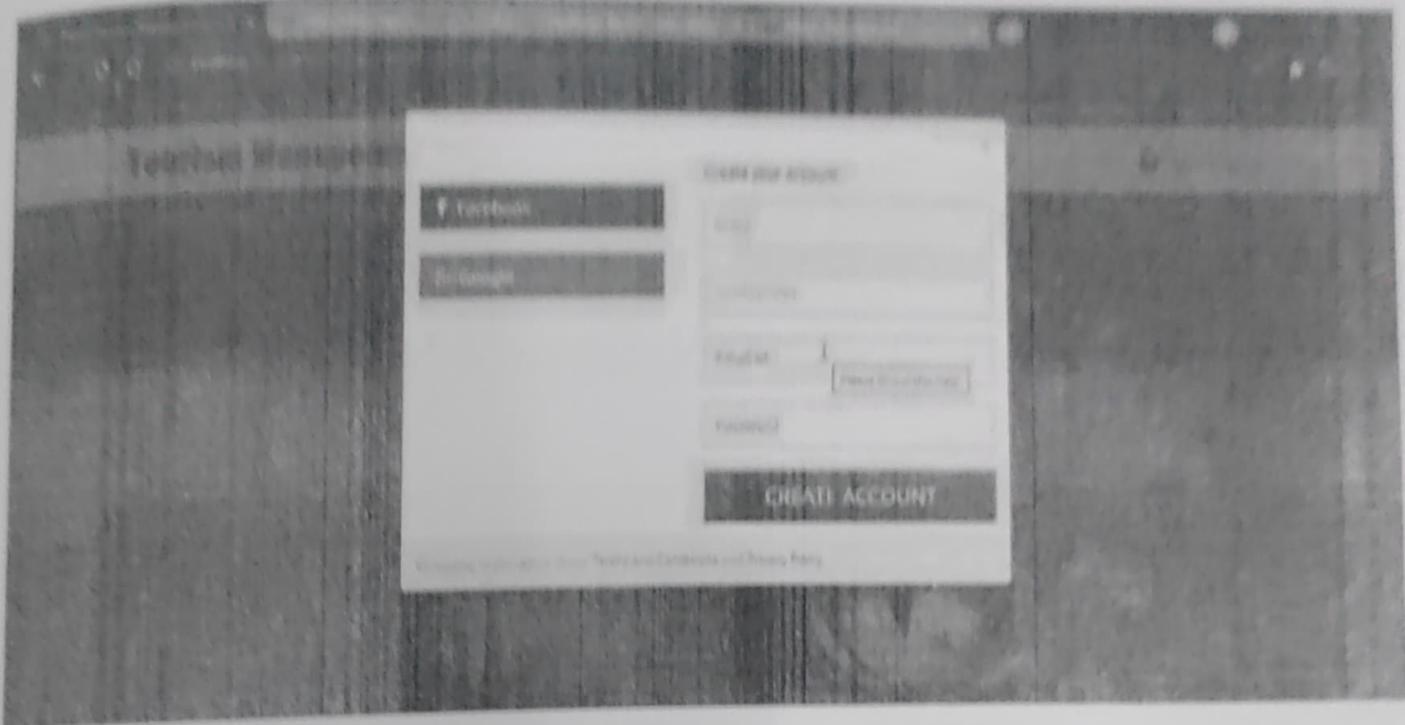
- ❖ INSERT INTO tblusers(FullName,MobileNumber,EmailId,Password) VALUES(:fname,:mnumber,:email,:password);
- ❖ INSERT INTO tblissues(UserEmail,Issue,Description) VALUES(:email,:issue,:description);
- ❖ INSERT INTO TblTourPackages(PackageName,PackageType,PackageLocation,PackagePrice,PackageFeatures,PackageDetails,PackageImage) VALUES(:pname,:ptype,:plocation,:pprice,:pfeatures,:pdetails,:pimage);
- ❖ INSERT INTO tblenquiry(FullName,EmailId,MobileNumber,Subject,Description) VALUES(:fname,:email,:mobile,:subject,:description);

Select Queries

- ❖ SELECT EmailId,Password FROM tblusers WHERE EmailId=:email and Password=:password;
- ❖ SELECT * from tbltourpackages;
- ❖ SELECT FromDate FROM tblbooking WHERE UserEmail=:email and BookingId=:bid;

Update Queries

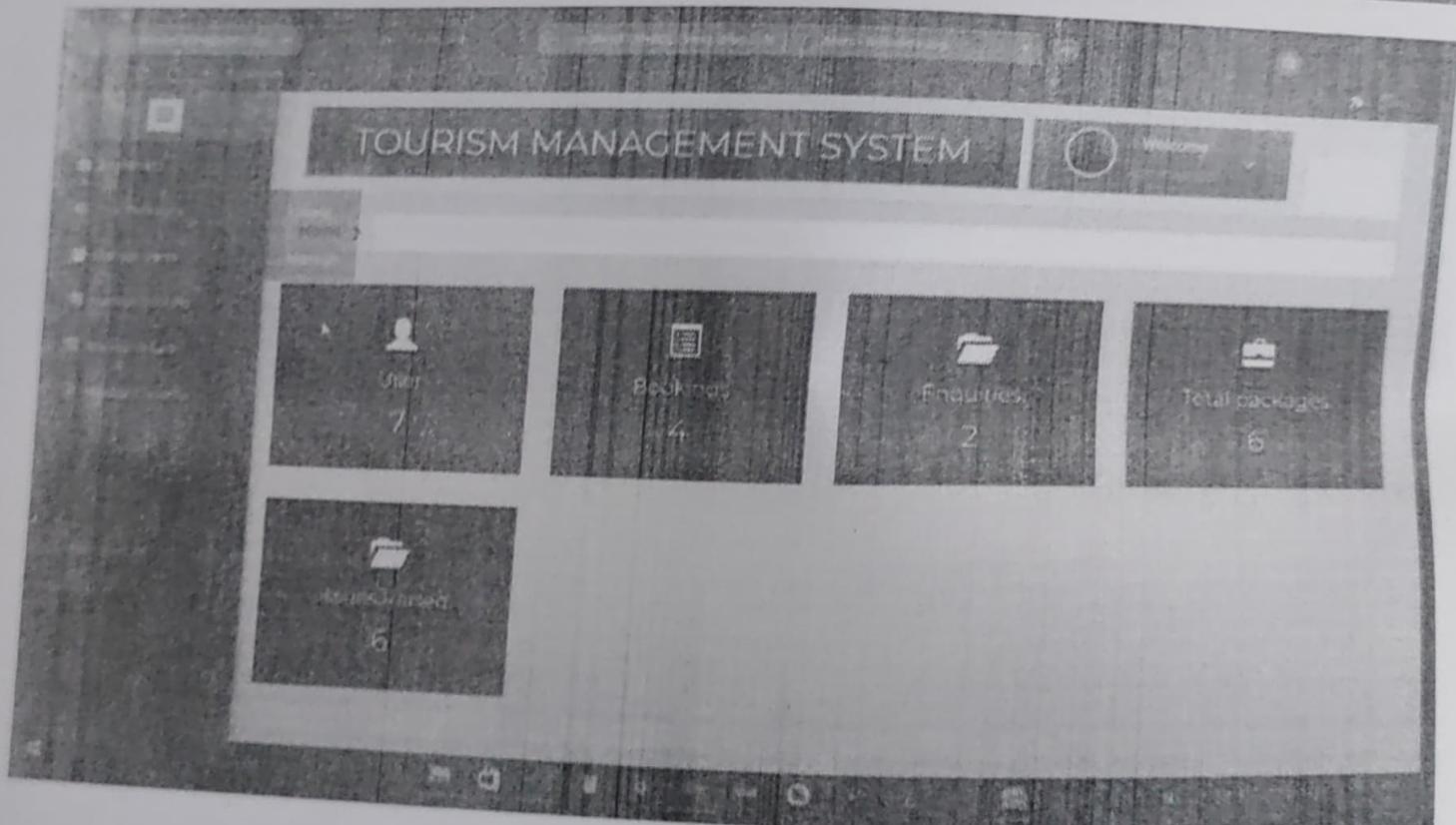
- ❖ UPDATE tblbooking SET status=:status,CancelledBy=:cancelby WHERE BookingId=:bid;
- ❖ update admin set Password=:newpassword where UserName=:username
- ❖ UPDATE tblenquiry SET Status=:status WHERE id=:eid;
- ❖ update TblTourPackages set PackageName=:pname,PackageType=:ptype, PackageLocation=:plocation,PackagePrice=:pprice,PackageFetures=:pfeatures,PackageDetails=:pdetails where PackageId=:pid;
- ❖ UPDATE tbliissues SET AdminRemark=:remark WHERE id=:iid;



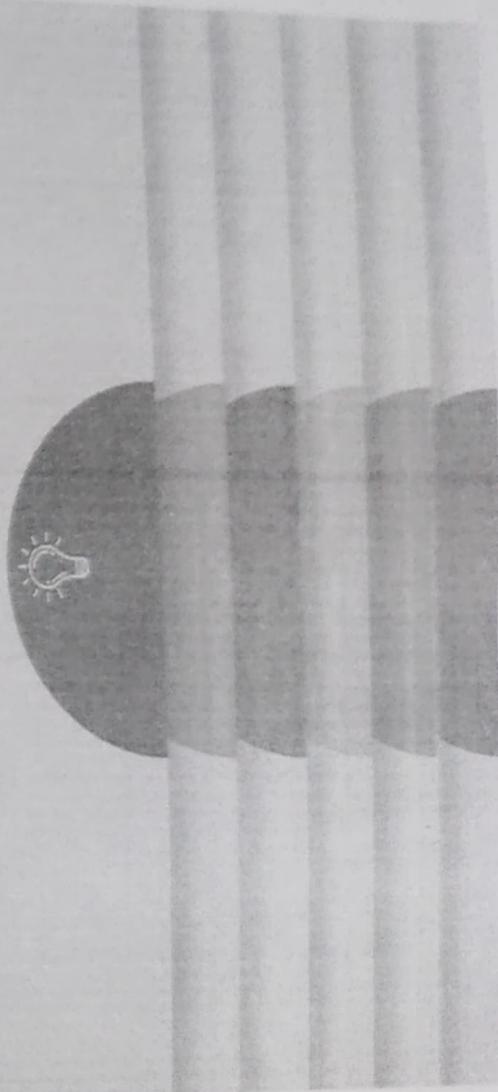
TOURISM > Manage Issues

Manage Issues

#	NAME	MOBILE NO.	EMAIL ID	ISSUES	DESCRIPTION	POSTING DATE	ACTION
10010	Rakesh	9898778999	rakesh@gmail.com	Issue	220	2021-09-23 10:32:15	View
10011	Rakesh	9898778999	rakesh@gmail.com	Booking Issues	Booking not accepted	2021-09-23 17:40:48	View
10012	Sunita	9898778999	sunita@gmail.com	Cancellation	Package not issued	2021-09-24 09:17:06	View
10013	Rakesh	9898778999	rakesh@gmail.com	Refund	Amount not claimed	2021-09-24 09:20:47	View

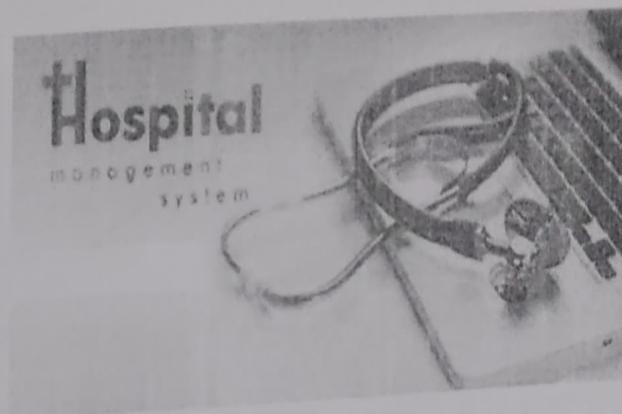


THANK YOU



Database Management Systems

Mini Project
Hospital Management System



Submitted by,
D. Oviya
19CSR130

AIM:

Create a mini project (Application Development using Oracle/ SQL SERVER / MYSQL) for Hospital Management Systems

PROGRAM:

```
import pymysql
class MySqlDb:
    def __init__(self, host, username, password, db_name):
        self.conn = pymysql.connect(
            host=host,
            user=username,
            passwd=password,
            database=db_name
        )
        self.my_cursor = self.conn.cursor()

    def createTable(self, sql):
        try:
            self.my_cursor.execute(sql)
            self.conn.commit()
            return "Table Created"
        except:
            try:
                dropHospital = 'Drop table hospital'
                dropDoctor = 'Drop table Doctor'
                self.my_cursor.execute(dropDoctor)
                self.my_cursor.execute(dropHospital)
                self.my_cursor.execute(sql)
                self.conn.commit()
                return "Table Created"
            except:
                dropHospital = 'Drop table hospital'
                self.my_cursor.execute(dropHospital)
```

```
        self.my_cursor.execute(sql)
        self.conn.commit()
        return "Table Created"

def findAll(self, sql):
    try:
        self.my_cursor.execute(sql)
        return self.my_cursor.fetchall()
    except:
        print("Exception occurred!!! Try again... ")

def insertOne(self, sql, params):
    try:
        self.my_cursor.execute(sql, params)
        self.conn.commit()
        return self.my_cursor.rowcount
    except:
        print("Exception occur!!! make sure that u had enter the unique id and the entered values satisfies
primary and foreign key constraint \nTRY AGAIN..... ")

def findByParams(self, sql, params):
    try:
        self.my_cursor.execute(sql, params)
        return self.my_cursor.fetchall()
    except:
        print("Exception occurred!!! Try again... ")

def updateOne(self, sql, params):
    try:
        self.my_cursor.execute(sql, params)
        self.conn.commit()
        print("Updated Successfully")
    except:
        print("Exception occurred!!! Try again... ")

def deletebyId(self, sql, id):
    try:
        self.my_cursor.execute(sql, id)
        self.conn.commit()
        print("Deleted Successfully")
    except:
        print("Exception occurred!! As u can't delete this info which is referred by another table.... ")

class InfoPrint:
```

```
def beauty_print_one(self, hosp):
    print(f"\t\t{hosp[0]} | {hosp[1]} | {hosp[2]} ")

def beauty_print_list(self, lists):
    print(f"\t\t ID| Hospital name | Bed Count")
    for hosp in lists:
        self.beauty_print_one(hosp)

def doc_beauty_print_one(self, doc):
    print(f"\t\t{doc[0]} | {doc[1]} | {doc[2]} | {doc[3]} | {doc[4]} | {doc[5]} ")

def doc_beauty_print_list(self, lists):
    print(f"\t\t ID| Doctor name | Hospital Name | Date of joining | Speciality | Salary")
    for doc in lists:
        self.doc_beauty_print_one(doc)

class HospitalManagement(MysqlDb):
    def __init__(self):
        super().__init__('localhost', 'root', 'oviYa', 'my_first_db')
        self.my_print = InfoPrint()

    def hosp_show_menu(self):
        print("""
            -- 1.Query all Hospital --
            -- 2.Add Hospital information --
            -- 3.Modify Hospital information --
            -- 4.Delete Hospital information --
            -- 5.Query Hospital information --
            -- 6.RETURN--
        """))

    def doc_show_menu(self):
        print("""
            -- 1.Query all Doctor --
            -- 2.Add Doctor information --
            -- 3.Modify Doctor information --
            -- 4.Delete Doctor information --
            -- 5.Query Doctor information --
            -- 6.RETURN--
        """)
```

```
def hosp_show_all(self):
    try:
        lists = self.findAll('select * from hospital')
        self.my_print.beauty_print_list(lists)
    except:
        print("Exception occurred!!! Try again.....")
    doctor inner join hospital on doctor.hospital_id=hospital.hospital_id
    self.my_print.doc_beauty_print_list(lists)

def add_hospital(self):
    try:
        id = int(input('Please enter hospital id:'))
        name = input('Please enter hospital name:')
        count = int(input('Please enter Bed count:'))
        insert_sql = 'insert into hospital(hospital_id, hospital_name, Bed_count) values(%s, %s, %s)'
        sql_params = (id, name, count)
        rowcount = self.insertOne(insert_sql, sql_params)
        if rowcount == 1:
            print("Add success")
    except:
        print("Exception occurred!!! invalid input")

def add_doctor(self):
    try:
        id = int(input('Please enter doctor id:'))
        name = input('Please enter doctor name:')
        lists = self.findAll('select hospital_id,hospital_name from hospital')
        print(f"\t\tHospital ID | Hospital Name ")
        for doc in lists:
            print(f"\t\t{doc[0]} | {doc[1]} ")
        hosp_id=int(input("Please select doctor's coresponding hospital's id:"))
        date=input("Please enter joining date:")
        speciality=input("Please enter doctor's speciality:")
        salary=int(input("Please enter doctor's Salary:"))
        insert_sql = 'insert into doctor(doctor_id, doctor_name, hospital_id,joining_date,speciality,Salary)'values(%s, %s, %s, %s, %s ,%s)'
        sql_params = (id, name, hosp_id,date,speciality,salary)
        rowcount = self.insertOne(insert_sql, sql_params)
        if rowcount == 1:
            print("Add success")
    except:
```

```

        self.start()

    except:
        print("Exception Occurred!! Invalid input")

if __name__ == '__main__':
    try:
        app=HospitalManagement()
        app.start()
    except:
        print("Exception occurred!! ")
    finally:
        print("*****Program Executed SUCCESSFULLY *****")

```

OUTPUT:

```

CREATE TABLE 1.hospital | 2.doctor | 3.created both,NEXT!
1
Table Created
CREATE TABLE 1.hospital | 2.doctor | 3.created both,NEXT!
2
Table Created
CREATE TABLE 1.hospital | 2.doctor | 3.created both,NEXT!
3
2 tables created successfully
Do u want to add initial data into these tables y/n:y
Please enter the sequence number of the table to which u want to modify: 1.HOSPITAL | 2. DOCTOR |
0.EXIT 1

-- 1.Query all Hospital --
-- 2.Add Hospital information --
-- 3.Modify Hospital information --
-- 4.Delete Hospital information --
-- 5.Query Hospital information --
-- 6.RETURN--

Please enter the sequence number of the operation to be performed: 1
ID| Hospital name | Bed Count
2 | psg hospital | 500
3 | kmch hospital | 600
4 | healthcare hospital | 700
5 | appollo hospital | 800
6 | kmch hospital | 600
7 | appollo hospital | 800

```

- 1.Query all Hospital --
- 2.Add Hospital information --
- 3.Modify Hospital information --
- 4.Delete Hospital information --
- 5.Query Hospital information --
- 6.RETURN--

Please enter the sequence number of the operation to be performed: 2

Please enter hospital id:1

Please enter hospital name:abi hospital

Please enter Bed count:100

Add success

- 1.Query all Hospital --
- 2.Add Hospital information --
- 3.Modify Hospital information --
- 4.Delete Hospital information --
- 5.Query Hospital information --
- 6.RETURN--

Please enter the sequence number of the operation to be performed: 1

ID|Hospital name | Bed Count

1	abi hospital 100
2	psg hospital 500
3	kmch hospital 600
4	healthcare hospital 700
5	appollo hospital 800
6	kmch hospital 600
7	appollo hospital 800

- 1.Query all Hospital --
- 2.Add Hospital information --
- 3.Modify Hospital information --
- 4.Delete Hospital information --
- 5.Query Hospital information --
- 6.RETURN--

Please enter the sequence number of the operation to be performed: 6

0.EXIT 2

Please enter the sequence number of the table to which u want to modify: 1.HOSPITAL / 2. DOCTOR

- 1.Query all Doctor --
- 2.Add Doctor information --

- 3.Modify Doctor information --
- 4.Delete Doctor information --
- 5.Query Doctor information --
- 6.RETURN--

Please enter the sequence number of the operation to be performed: 1

ID	Doctor name	Hospital Name	Date of joining	Speciality	Salary
7	dr.justin	psg hospital	2-4-2007	allergist	400000
8	dr.sam	kmch hospital	3-4-2005	cardiologist	500000
9	dr.lucy	healthcare hospital	6-7-2006	dermatologist	700000
10	dr.drake	apollo hospital	30-8-2019	hematologist	900000
11	dr.sam	kmch hospital	3-4-2005	cardiologist	500000
12	dr.drake	apollo hospital	30-8-2019	hematologist	900000

- 1.Query all Doctor --
- 2.Add Doctor information --
- 3.Modify Doctor information --
- 4.Delete Doctor information --
- 5.Query Doctor information --
- 6.RETURN--

Please enter the sequence number of the operation to be performed: 3

Please enter the name of the Doctor you want to modify:8

There is no such Doctor name.

- 1.Query all Doctor --
- 2.Add Doctor information --
- 3.Modify Doctor information --
- 4.Delete Doctor information --
- 5.Query Doctor information --
- 6.RETURN--

Please enter the sequence number of the operation to be performed: 3

Please enter the name of the Doctor you want to modify:dr.sam

ID| Hospital name | Bed Count

8 | dr.sam | 3

11 | dr.sam | 6

Multiple results are found. Please enter the hospital id to be modified: 8
Please select the information to modify, 1. Doctor Name | 2. Hospital ID | 3.Date of Join | 4.Speciality |

5.Salary: 5

Enter the salary you want to modify:100000

Updated Successfully

- 1.Query all Doctor --

-- 2.Add Doctor information --
-- 3.Modify Doctor information --
-- 4.Delete Doctor information --
-- 5.Query Doctor information --
-- 6.RETURN--

Please enter the sequence number of the operation to be performed: 1

ID	Doctor name	Hospital Name	Date of joining	
Speciality	Salary	7	dr.justin	psg hospital 2-4-2007
allergist	400000			
8	dr.sam	kmch hospital	3-4-2005	cardiologist 100000
9	dr.lucy	healthcare hospital	6-7-2006	
dermatologist	700000	10	dr.drake	appollo hospital
30-8-2019	hematologist	900000	11	dr.sam kmch
hospital	3-4-2005	cardiologist	500000	
12	dr.drake	appollo hospital	30-8-2019	hematologist 900000

-- 1.Query all Doctor --
-- 2.Add Doctor information --
-- 3.Modify Doctor information --
-- 4.Delete Doctor information --
-- 5.Query Doctor information --
-- 6.RETURN--

Please enter the sequence number of the operation to be performed: 5
Query by 1.NAME | 2.ID | 3.HOSPITAL NAME | 4.JONING DATE | 5.SPECIALITY | 6.SALARY: 3

Please enter the hospital name of the doctor you want to query:kmch hospital

ID	Doctor name	Hospital Name	Date of joining	
Speciality	Salary	8	dr.sam	kmch hospital 3-4-2005
cardiologist	100000			
11	dr.sam	kmch hospital	3-4-2005	cardiologist 500000

-- 1.Query all Doctor --
-- 2.Add Doctor information --
-- 3.Modify Doctor information --
-- 4.Delete Doctor information --
-- 5.Query Doctor information --
-- 6.RETURN--

Please enter the sequence number of the operation to be performed: 6
Please enter the sequence number of the table to which u want to modify: 1.HOSPITAL | 2.

DOCTOR | 0.EXIT

Topic : Online Food Ordering

Team Members :

19CSR197-G.SUBALAKSHMI.
19CSR180-A.SARANYA.
19CSR201-S.SUJA.

SignUp Page :

The screenshot shows a web browser window with the title "Online Food Store". The address bar displays "localhost:3050/user/register". The main content is a registration form titled "Register". It contains three input fields: "Name" with the value "Suja", "Phone no." with the value "9876543210", and "Password" with the value "....". Below the password field is a "Register" button. At the bottom of the screen, there is a taskbar with various icons and a system tray showing the date and time as "20-06-2021 18:59".

Name	Suja
Phone no.	9876543210
Password

Type here to search

Q

Q B E I



Milkshake

Rs. 70.0

Add to cart

Rs. 140.0

Add to cart

Rs. 250.0

Pizza

I usually have a sandwich for lunch.

Pasta

There is a smell of Pasta in this room.

Baked Beans

Search

Sales
Units
Suggets

Cart
Orders

Logout

Menu

ONLINE FOOD ORDER

→ G (D) [Logout](#) [User Profile](#) [Logout](#)

DashBoard :

New User Register Here

Login

....

Phone No

9876543210

Passed

Login

Logout

Logout

SignIn (Or) Login Page :

Search for Specific food :

A screenshot of a web browser window titled "Online Food ORDER". The search bar at the top contains the text "pasta". Below the search bar, there are two search results:

- Pasta**
There is a smell of Pasta in this room.
Background Bowls
Rs. 250.0
[Add to cart](#)
- Cheese Pasta**
I'd like a cheeseburger to take away, please.
Sri O'One
Rs. 170.0
[Add to cart](#)

The browser interface includes a navigation bar with icons for back, forward, and search, as well as a taskbar at the bottom showing system status.

Order in Cart with Total Price:

A screenshot of a web browser window titled "Online Food ORDER". The search bar at the top contains the text "pasta". Below the search bar, there is a message indicating the total cost of the items in the cart:

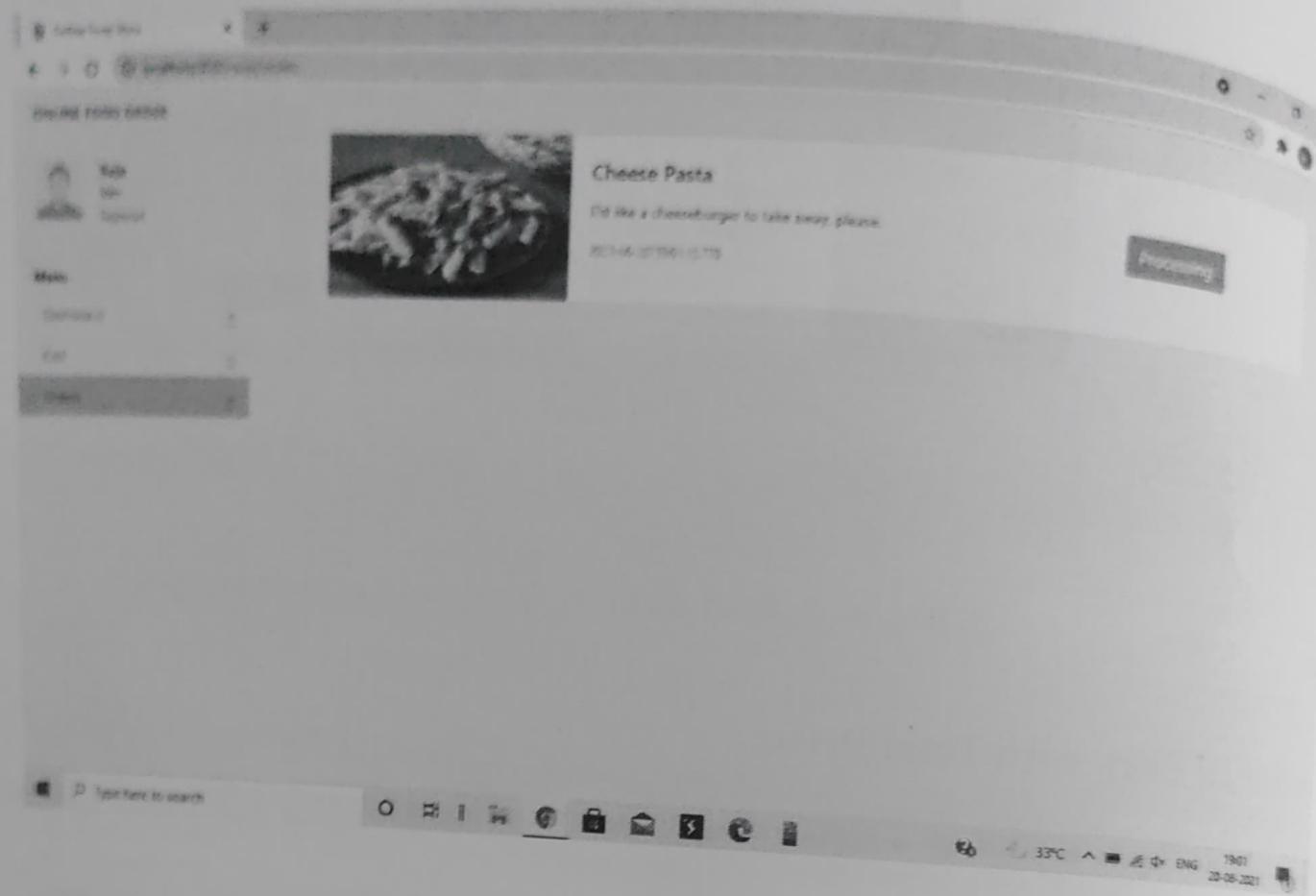
TotalCost: Rs 170.0

The cart contains one item:

- Cheese Pasta**
I'd like a cheeseburger to take away, please.
Rs. 170.0
[Remove from cart](#)

The browser interface includes a navigation bar with icons for back, forward, and search, as well as a taskbar at the bottom showing system status.

Order Status:



Cart Status After Ordered Food :

