

UCS1412 – Database Lab

Mini Project

Pharmacy management system

Done by

Abhishek K (185001005)

Gokhulnath T (185001051)

Hemanth Kumar K M (185001059)

1) Abstract

Pharmacy management system is a management system that is designed to improve accuracy and to enhance safety and efficiency in the pharmaceutical store. This program can be used in any pharmaceutical shops having a database to maintain. It is a computer-based system which helps the pharmacist to improve inventory management, cost, medical safety etc. The system allows the user to enter a manufacturing and expiry date for a particular product or drug during opening stock and sales transaction. The software can print invoices, bills, receipts etc. It can also maintain the record of supplies sent in by the supplier. The system services and goals are established by consultation with system user. It also involves manual entry upon arrival of new batches of drugs and upon drug movement out of the pharmacy for a certain period. Pharmacy management system deals with the maintenance of drugs and consumables in the pharmacy unit. The main aim of the project is to manage the database of a pharmaceutical shop. This project provides insight into the design and implementation of a Pharmacy Management System. It is done by creating a database of the available medicines and equipment in the shop. The primary aim of pharmacy management system is to improve accuracy and enhance safety and efficiency in the pharmaceutical store and to develop software for its effective management. Pharmacy management system is useful to maintain correct database by providing an option to update the drugs in stock and is used to manage most pharmacy related activities.

2) ENTITY-RELATIONSHIP DIAGRAM

ER Diagram for this project consists of 9 regular entities and 3 relational/associative entities. The regular entities are:

- 1: PHARMACY
- 2: CUSTOMER
- 3: DOCTOR
- 4: MANUFACTURER
- 5: MEDIQUIPMENT
- 6: EMPLOYEE
- 7: HOSPITAL
- 8: BILL
- 9: SUPPLIER

The associative entities are:

- 1: PRESCRIBE
- 2: CONTRACT
- 3: WORKS

The attributes of each of these entities are:

1: PHARMACY {PHID, NAME, CITY, FAX, PHONE} where PHID is primary key.

Pharmacy Management System

Dept of ISE, BNMIT 2017-18 Page | 4

2: CUSTOMER {PID, NAME, SEX, CITY, PHONE, AGE, DID} where PID is primary key and DID is foreign key.

3: DOCTOR {DID, DNAME, SPECIALITY, AGE, MOBILE, GENDER} where DID is primary key.

4: MANUFACTURER {CID, NAME, EMAIL, MOBILE, CITY, PHARID} where CID is primary key and PHARID is foreign key.

5: MEDIQUIPMENT {CODE, TRADE_NAME, PRODUCT_TYPE, MFG_DATE, EXP_DATE, PRICE, CID} where CODE is primary key and CID is foreign key.

6: EMPLOYEE {NAME, CITY, DOJ, MOBILE, SALARY, AGE, SEX, PHARID} where MOBILE is primary key and PHARID is foreign key.

7: HOSPITAL {HID, NAME, EMAIL, PHONE, CITY, PHARID} where HID is primary key and PHARID is foreign key.

8: BILL {BID, DOB, AGE, PNAME, MOBILE, CITY, PRODUCT, AMOUNT, PHARID} where BID is primary key.

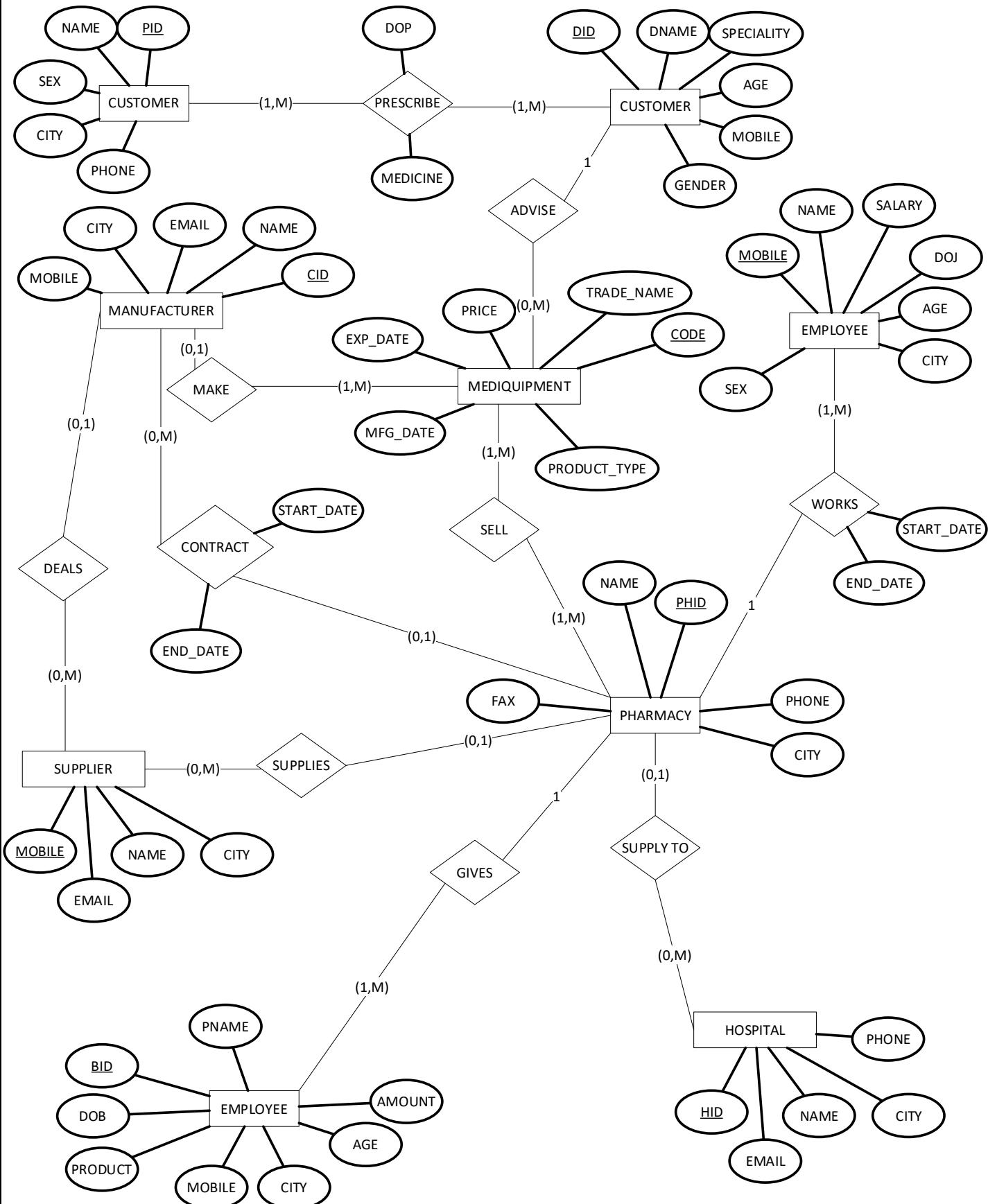
9: SUPPLIER {NAME, CITY, MOBILE, EMAIL, CID, PHARID} where MOBILE is primary key and CID, PHARID is foreign key.

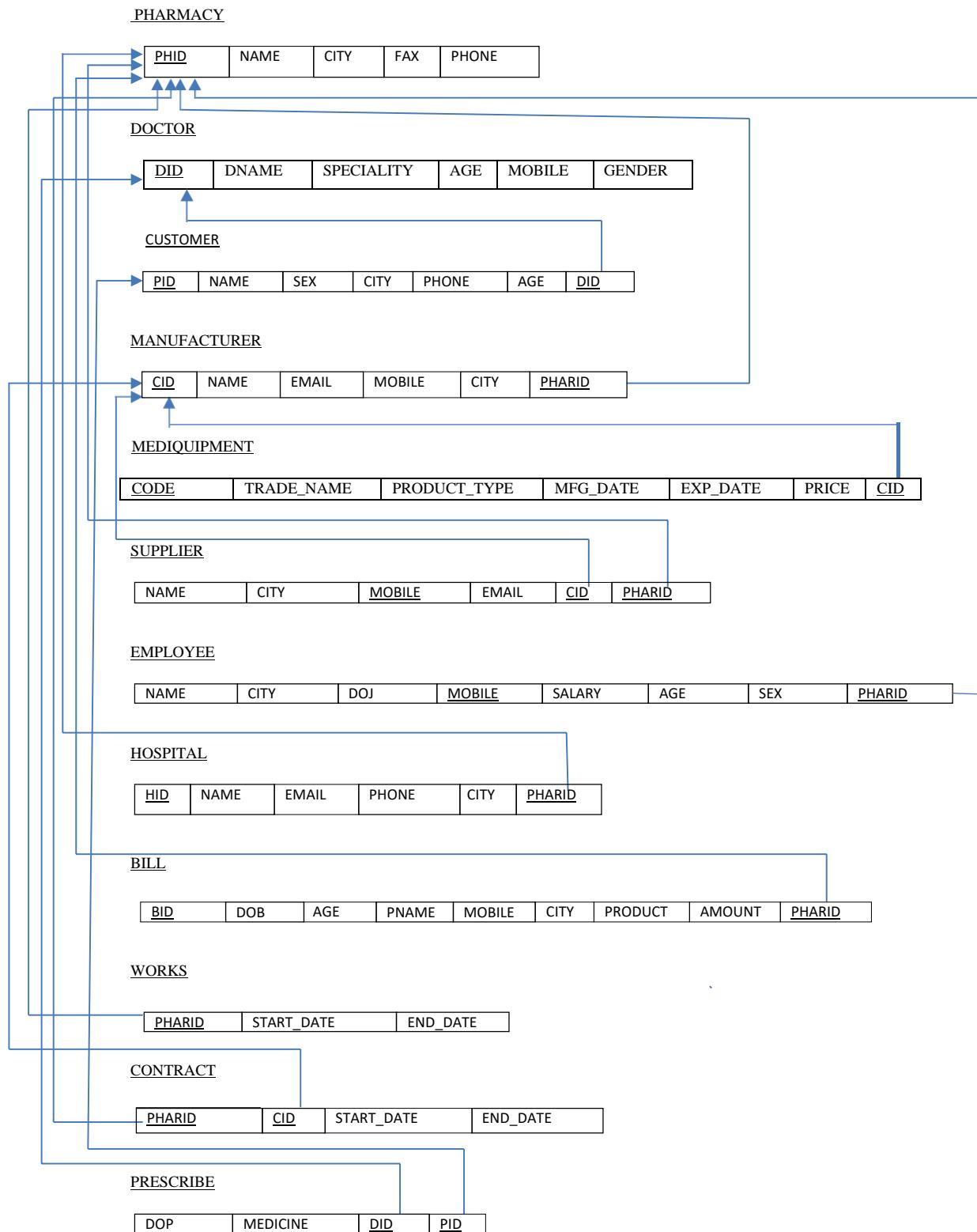
10: PRESCRIBE {DATE, MEDICINE, DID, PID} where DID, PID is foreign key.

11: CONTRACT {PHARID, CID, START_DATE, END_DATE} where PHARID, CID is foreign key.

12: WORKS {PHARID, START_DATE, END_DATE} where PHARID is foreign key.

ER Diagram of Pharmacy Management System





Relational Database Schema for Pharmacy Management System

NORMALISATION

① PHARMACY:

<u>PHID</u>	NAME	CITY	FAX	PHONE
	↑	↑	↑	↑

FD(s):

$$\text{PHID} \rightarrow \{\text{NAME}, \text{CITY}, \text{FAX}, \text{PHONE}\}$$

INF: This states that an attribute must include only atomic values. Else, the table should be decomposed.

⇒ since all the attributes of PHARMACY have atomic values, the relation is already in INF.

2NF: This rule states that a relation is in 2NF if all the non-prime attribute of the relation are fully functional dependent on the primary key.

Else, the relation is decomposed.

⇒ since all the non-primary attributes of the relation are fully functional dependent on the primary key, the relation is in 2NF.

3NF: This rule states that a relation R is in 3NF form if it is in 2NF and no non-prime attribute is transitively dependent on the primary key.

Else, the table is decomposed.

\Rightarrow Since no non-prime attribute is transitively dependent on the primary key and is in 2NF, the relation is in 3NF.

② DOCTOR:

DID	DNAME	SPECIALITY	AGE	MOBIE	GENDER

FD(s):

$$DID \rightarrow \{DNAME, SPECIALITY, AGE, MOBILE, GENDER\}$$

INF: Relation (DOCTOR) is already in INF as all the attributes of the relation have atomic values.
 \Rightarrow Hence no need to decompose the table.

2NF: The Relation (DOCTOR) is in 2NF as all the non-prime attributes are fully dependent on the primary key. and also it is in INF.

3NF: The relation (DOCTOR) is already in 3NF as no non-prime attribute is dependent on the candidate key and the relation is in 2NF.
 \Rightarrow Hence no need to decompose the table.

③ CUSTOMER :

PID	NAME	SEX	CITY	PHONE	AGE	PID

$\text{PID} \rightarrow [\text{NAME}, \text{SEX}, \text{CITY}, \text{PHONE}, \text{AGE}]$

INF : The Relation (customer) is already in INF as all the attributes have atomic values.

\Rightarrow Hence no need to decompose.

2NF : The Relation (customer) is in 2NF as all the non-prime attributes of the relation are fully functionally dependent on the primary key and is in INF

\Rightarrow Hence no need to decompose.

3NF : The relation (customer) is in 3NF as no non-prime attributes are transitively dependent on the candidate keys, as is in 2NF.

\Rightarrow Hence no need to decompose.

④ HOSPITAL :

HID	NAME	EMAIL	PHONE	CITY	PHID

FD(s) : HID \rightarrow { NAME, EMAIL, PHONE, CITY }

INF : The relation R (HOSPITAL) is in INF as all the attributes of the relation have atomic values.
 \Rightarrow Hence no need to decompose.

2NF : The relation R (HOSPITAL) is in 2NF as all the non-prime attributes of R are fully dependent on the primary keys and it is in \geq INF.
 \Rightarrow Hence no need to decompose.

3NF : The relation R (HOSPITAL) is in 3NF as no non-prime attributes of R are transitively dependent on the candidate keys, and it is in 2NF.
 \Rightarrow Hence no need to decompose.

⑤ EMPLOYEE :

NAME	CITY	DOJ	MOBILE	SALARY	AGE	SEX	PHTID
↑	↑	↑	↑	↑	↑	↑	↑

FD(s) :
MOBILE \rightarrow { NAME, CITY, DOJ, SALARY, AGE, SEX }

INF : The relation R (EMPLOYEE) is in INF as all the attributes in R have atomic values
 \Rightarrow Hence no need to decompose.

2NF: The relation R (EMPLOYEE) is in 2NF as all the non-prime attributes in R are fully dependent on the primary keys, and as it is in 1NF.

⇒ Hence no need to decompose.

3NF: The relation R (EMPLOYEE) is in 3NF as no non-prime attribute is transitively dependent on the candidate keys, and as it is in 2NF.

⇒ Hence no need to decompose.

⑥ MANUFACTURER :

CID	NAME	EMAIL	MOBILE	CITY	PHID

FD (b) :

$$CID \rightarrow \{NAME, EMAIL, MOBILE, CITY\}$$

1NF: The relation R (MANUFACTURER) is in 1NF as all the attributes in R have atomic values.

⇒ Hence no need to decompose.

2NF: The relation R (MANUFACTURER) is in 2NF as all the non-prime attributes are fully dependent on the primary keys and as the relation is in 1NF.
⇒ Hence no need to decompose.

3NF: The relation R (MANUFACTURER) is in 3NF as no non-prime attributes are transitively dependent on the candidate keys and as the relation is in 2NF.
⇒ Hence no need to decompose.

④ MEDIQUIPMENT:

CODE	TRADE-NAME	PRODUCT-NAME	MFGI-DATE	EXP-DATE	PRICE	CID
	↑	↑	↑	↑	↑	↑

FD(b): $\text{CODE} \rightarrow \{\text{TRADE-NAME}, \text{PRODUCT-NAME}, \text{MFGI-DATE}, \text{EXP-DATE}, \text{PRICE}\}$

1NF: The relation R (MEDIQUIPMENT) is in 1NF as all the attributes in the R are atomic values.
⇒ Hence no need to decompose.

2NF: The relation R (MEDIQUIPMENT) is in 2NF as all the non-prime attributes of R are fully dependent on the primary keys, and as the relation is in 1NF.
⇒ No need to decompose further.

3NF: The Relation R (MED EQUIPMENT) is in 3NF as no non-prime attributes of R are transitively dependent on the candidate keys and the relation (R) is in 2NF.

⇒ no need to decompose further.

⑧ SUPPLIER

NAME	CITY	MOBILE	SALARY	EMAIL	CID	PHID
↑	↑	↓				↑

FD(G): MOBILE → {NAME, CITY, EMAIL}

INF: The relation R (SUPPLIER) is in INF as all the attributes in R are having atomic values
⇒ Hence no need to decompose.

2NF: The relation R (SUPPLIER) is in 2NF as all the non-prime attributes of R are fully dependent on the primary keys and the relation R is in 2NF.
⇒ Hence no need to decompose

3NF: The relation R (SUPPLIER) is in 3NF as no non-prime attribute is dependent (transitively) on candidate keys and the relation R is in 2NF
 \Rightarrow Hence no further decomposition needed.

⑨ BILL:

BID	DOB	AGE	PNAME	MOBILE	CITY	PRODUCT	AMOUNT	PHID

FD(b): $BID \rightarrow \{DOB, AGE, PNAME, MOBILE, CITY, PRODUCT, AMOUNT\}$

1NF: The relation R (BILL) is already in 1NF as all the attributes are having atomic values.
 \Rightarrow Hence no need to decompose further.

2NF: The relation R (BILL) is in 2NF as all the non-prime attributes are fully dependent on primary keys and the relation R is in 1NF.
 \Rightarrow Hence no need to decompose further.

3NF: The relation R (BILL) is in 3NF as no non-prime attributes are transitively dependent on candidate keys and the relation R is in 2NF
 \Rightarrow Hence no further decomposition required.

(10)

WORK :

PHID	START-DATE	END-DATE
I	↑	R

FD(S) : PHID $\rightarrow \{ \text{START-DATE}, \text{END-DATE} \}$

INF : The relation R (WORK) is in INF as all the attributes of relation R are atomic values

\Rightarrow Hence no need to decompose.

2NF : The relation R (WORK) is in 2NF as all the non-prime attributes of Relation R are fully independent on the primary keys and the relation R is in $\geq 1NF$.

\Rightarrow Hence no need to decompose further.

3NF : The Relation R (WORK) is in 3NF as no non-prime attributes are transitively dependent on the candidate keys and the relation R is in 2NF.

\Rightarrow Hence no need to decompose further.

11. CONTRACT :

PHID	CID	START-DATE	END-DATE

FD(σ) : PHID $\rightarrow \{CID\}$

INF : The Relation R (CONTRACT) is in INF as all the attributes have atomic values.
 \Rightarrow Hence no need to decompose.

2NF : The relation R (CONTRACT) is not in 2NF as a few non-prime ~~key~~ attributes are not fully independent (partial) dependent.
 \Rightarrow Hence we need to decompose.

CONTRACT(1) :

PHID	CID	START-DATE

CONTRACT(2) :

PHID	CID	END-DATE

\Rightarrow CONTRACT(1) and CONTRACT(2) are in 2NF and INF.

3NF : The Relation R₁ (CONTRACT(1)) and R₂ (CONTRACT(2)) is in 3NF as no non-prime attribute are transitively dependent on the candidate keys and they both are in 2NF

\Rightarrow Hence no need to decompose further.

(12) PREScribe :

DID	PID	DATE	MEDICINE

FD(s): DID \rightarrow [PID]

INF : The Relation R (PREScribe) is in INF as all the attributes of R are having atomic values.

\Rightarrow Hence no need to decompose further.

2NF : The relation R (PREScribe) is not in 2NF as not all the non-prime attributes are fully dependent (PARTIAL) on the primary key.

\Rightarrow Hence we need to decompose the relations.

PREScribe (1) :

DID	PID	DATE

PREScribe (2) :

DID	PID	MEDICINE

Now both the relations R₁ (PREScribe) and R₂ (PREScribe) are in 2NF as the non-prime attributes are fully dependent on the prime keys and the relations R₁ and R₂ Both are in INF.

3NF: Decomposed relations R_1 (PRESCRIBE(1)) and R_2 (PRESCRIBE(2)) are both in 3NF as no non-prime attribute is transitively dependent on the candidate keys and both the relations R_1 and R_2 are in 2NF

\Rightarrow Hence no need to decompose further.

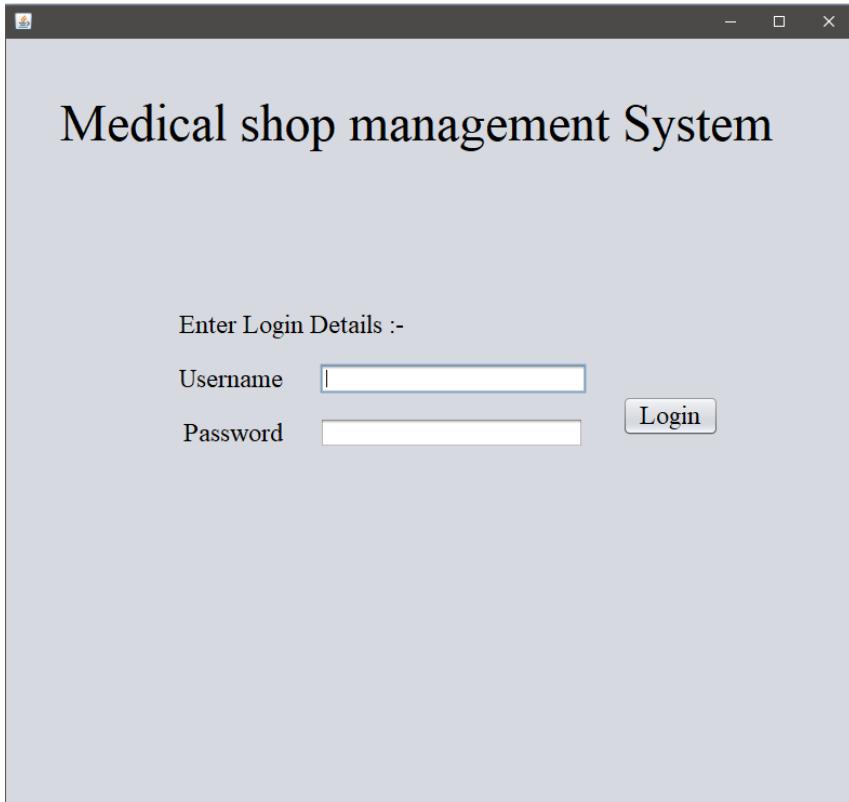
KEY ATTRIBUTES

<u>ENTITY</u>	<u>PRIMARY KEY</u>	<u>FOREIGN KEY</u>
PHARMACY	PHID	
DOCTOR	DID	
CUSTOMER	PID	DID
MANUFACTURER	CID	PHARID
MEDIQUIPMENT	CODE	CID
SUPPLIER	MOBILE	CID, PHARID
EMPLOYEE	MOBILE	PHARID
HOSPITAL	HID	PHARID
BILL	BID	PHARID
WORKS	PHARID	PHARID
CONTRACT	PHARID, CID	PHARID, CID
PRESCRIBE	PID, DID	DID,PID

SNAPSHOTS

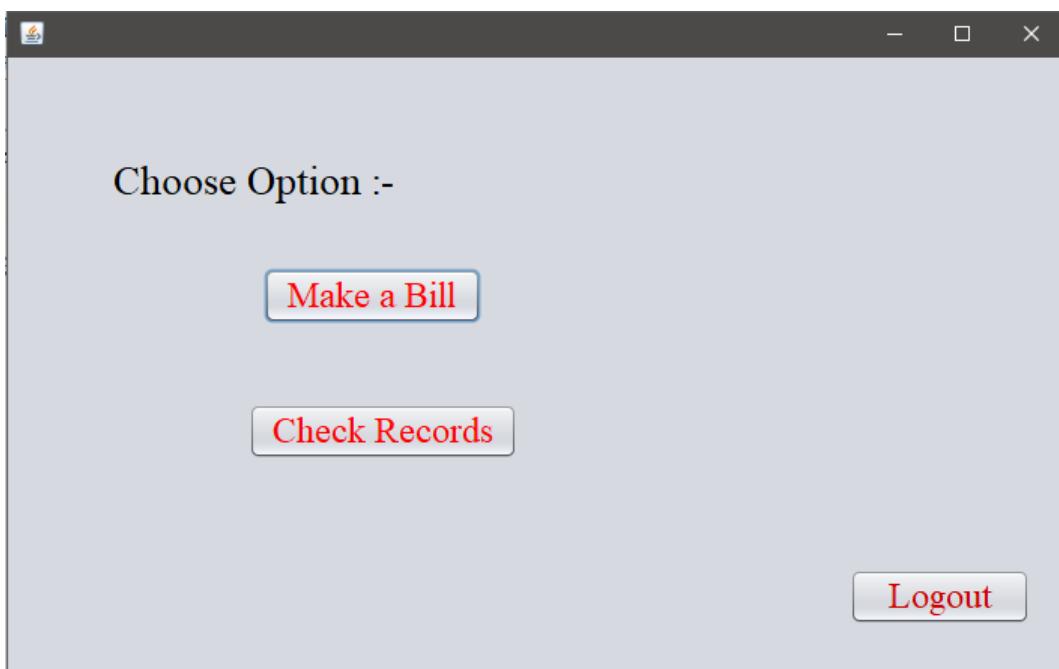
1)Login Page :-

This page asks the user to enter the username and password to login into the system.



2)Option Page :-

This page lets the user select from the different options available in the form.



3) Billing Page :-

This page allows the user to generate a bill and save the bill, once all the data that are asked in the form are inserted.

The screenshot shows a Windows application window titled "Billing Form". Inside, there are several input fields:

Date	03-may-2020		
Bill Id	100	Age	22
Name	gokhu		
Mobile	9566220635		
City	chennai		
Product	tablets		
Amount	600		

At the bottom are three buttons: "Back", "Save", and "Generate Invoice".

4) Bill Invoice Page :-

This page displays all the data that the user has entered in the billing page.

The screenshot shows a Windows application window titled "BILL INVOICE". It lists the same data as the Billing Form:

Date	03-may-2020
Bill ID	100
Age	22
Name	gokhu
Mobile	9566220635
City	chennai
Product	tablets
Amount	600

At the bottom is a single "Home" button.

5)Category Page :-

This page displays the different categories that are available in the system and are connected to the database.



6)Pharmacy Page :-

This page lets the user to insert data into the database from the front end, delete the data that the user wants to delete and view the database that is stored in the database.

A screenshot of a Windows application window titled "Pharmacy Details". The window is divided into three main sections: "Insert Details", "Delete Details", and "View All Details".

- Insert Details:** Contains fields for "Pharmacy ID" (text box), "Name" (text box), "City" (text box), "FAX" (text box), "Phone" (text box), and an "Insert" button.
- Delete Details:** Contains a "Delete" button and a text input field labeled "Enter Pharmacy Id" with the value "1".
- View All Details:** Contains a "View" button and a "Back" button. Below these are two tables. The first table has columns: Pharmacy_ID, Pharmacy_Name, City, Fax, and Phone. The second table lists five rows of data:

Pharmacy_ID	Pharmacy_Name	City	Fax	Phone
1	gokhul_shop_1	chennai	-9765221	9001002003
2	gokhul_shop_2	mumbai	-9837990	9001002004
3	gokhul_shop_3	bangalore	-9811880	9001002001
4	abhi_shop_1	delhi	-9862028	9010126387
5	abhi_shop_2	chennai	-9817933	9661268268

After deletion,

The screenshot shows the 'Pharmacy Details' application window. In the 'Delete Details' section, the 'Enter Pharmacy Id' field contains '1'. A 'Delete' button is visible next to it. To the right, a message box titled 'Message' displays 'Delete Successfull' with an information icon and an 'OK' button.

Pharmacy Details

Insert Details

Pharmacy ID	<input type="text"/>	City	<input type="text"/>
Name	<input type="text"/>		
FAX	<input type="text"/>	Phone	<input type="text"/>

Delete Details

Enter Pharmacy Id Delete

View All Details

View Back

Pharmacy_ID	Pharmacy_Name	City	Fax	Phone
2	gokhul_shop_2	mumbai	-9837990	9001002004
3	gokhul_shop_3	bangalore	-9811880	9001002001
4	abhi_shop_1	delhi	-9862028	9010126387
5	abhi_shop_2	chennai	-9817933	9661268268
6	abhi_shop_3	bangalore	-9866294	9662312312

Inserting the details,

The screenshot shows the 'Pharmacy Details' application window. In the 'Insert Details' section, the 'Pharmacy ID' field contains '1', 'Name' is 'gokhul shop 1', 'FAX' is '-1234567', and 'Phone' is '9566220635'. An 'Insert' button is visible. To the right, a message box titled 'Message' displays 'Saved' with an information icon and an 'OK' button.

Pharmacy Details

Insert Details

Pharmacy ID	<input type="text"/> 1	City	<input type="text"/> chennai
Name	<input type="text"/> gokhul shop 1		
FAX	<input type="text"/> -1234567	Phone	<input type="text"/> 9566220635

Delete Details

Enter Pharmacy Id Delete

View All Details

View Back

Pharmacy_ID	Pharmacy_Name	City	Fax	Phone
2	gokhul_shop_2	mumbai	-9837990	9001002004
3	gokhul_shop_3	bangalore	-9811880	9001002001
4	abhi_shop_1	delhi	-9862028	9010126387
5	abhi_shop_2	chennai	-9817933	9661268268
6	abhi_shop_3	bangalore	-9866294	9662312312

7) Doctor Page :-

This page lets the user to insert data into the database from the front end, delete the data that the user wants to delete and view the database that is stored in the database.

Doctor Details

Insert Details

Doctor Id	<input type="text"/>	Age	<input type="text"/>
Name	<input type="text"/>		
Speciality	<input type="text"/>	Gender	<input type="text"/>
Phone	<input type="text"/>		

Delete Details

Enter Doctor Id

View All Details

Doctor_ID	Doctor_Name	Speciality	Age	Mobile	Gender
10	gokhul	cardiologists	45	9831312512	m
11	abhishek	diabetologists	50	9826263264	m
12	hemanth	gynaecologists	30	9006001002	m
13	ajay	hygienists	32	9875400500	m
14	gowtham	lenroloists	28	9861621312	m

After deletion,

Doctor Details

Insert Details

Doctor Id	<input type="text"/>	Age	<input type="text"/>
Name	<input type="text"/>		
Speciality	<input type="text"/>	Gender	<input type="text"/>
Phone	<input type="text"/>		

Delete Details

Enter Doctor Id

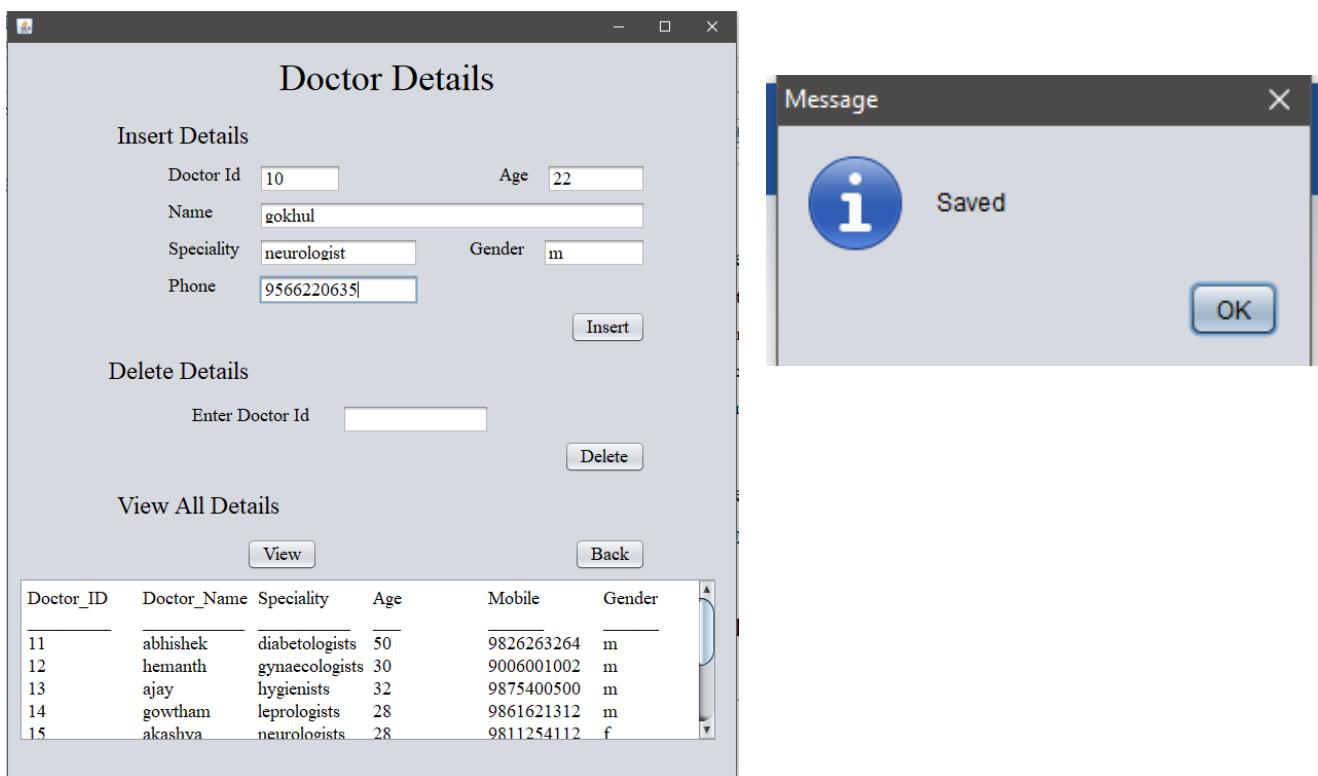
View All Details

Doctor_ID	Doctor_Name	Speciality	Age	Mobile	Gender
10	gokhul	cardiologists	45	9831312512	m
11	abhishek	diabetologists	50	9826263264	m
12	hemanth	gynaecologists	30	9006001002	m
13	ajay	hygienists	32	9875400500	m
14	gowtham	lenroloists	28	9861621312	m

Message

Delete Successfull

Inserting the details,



8)Customer Page :-

This page lets the user to insert data into the database from the front end, delete the data that the user wants to delete and view the database that is stored in the database.



After deletion,

Customer Details

Insert Details

Customer Id	<input type="text"/>	Age	<input type="text"/>
Name	<input type="text"/>		
City	<input type="text"/>	Sex	<input type="text"/>
Phone	<input type="text"/>		

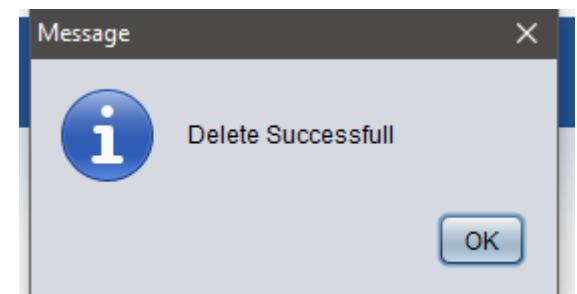
Delete Details

Enter Customer Id **Delete**

View All Details

View **Back**

Customer_ID	Customer_Name	Sex	City
100	anshu	m	chean
101	iqbal	m	rampu
102	priya	f	chenn
103	kaushik	m	patna



Inserting the details,

Customer Details

Insert Details

Customer Id	<input type="text" value="100"/>	Age	<input type="text" value="20"/>
Name	<input type="text" value="gokhul"/>		
City	<input type="text" value="chennai"/>	Sex	<input type="text" value="M"/>
Phone	<input type="text" value="9566220635"/>		

Insert

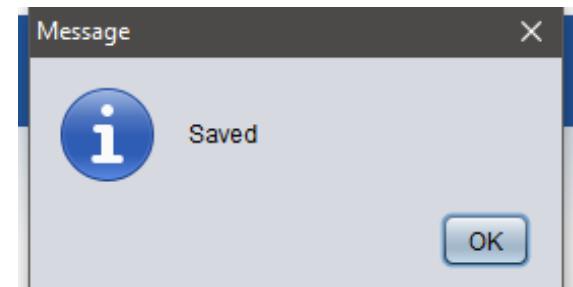
Delete Details

Enter Customer Id **Delete**

View All Details

View **Back**

Customer_ID	Customer_Name	Sex	City
101	iqbal	m	rampu
102	priya	f	chenn
103	kaushik	m	patna
104	nien	f	bawoo



9) Manufacturer Page :-

This page lets the user to insert data into the database from the front end, delete the data that the user wants to delete and view the database that is stored in the database.

Manufacturer Details

Insert Details

Manufacturer Id	<input type="text"/>
Name	<input type="text"/>
Email	<input type="text"/>
Mobile	<input type="text"/> City <input type="text"/>

Delete Details

Enter Manufacturer Id

View All Details

Manufacturer_ID	Name	Email	Mobile	City
1001	medicyl	med2@gmail.com	9212123134	Mysore
1002	vitade	vit3@gmail.com	9812121368	Chennai
1003	annuva	ann4@gmail.com	9001002115	Delhi
1004	mandavus	man5@gmail.com	9122113114	Hyderabad
1005	inovine	ino5@gmail.com	9881828112	Bangalore

After deletion,

Manufacturer Details

Insert Details

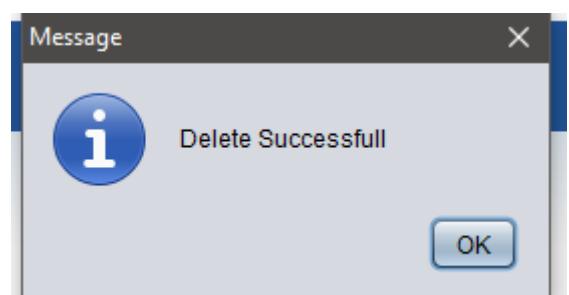
Manufacturer Id	<input type="text"/>
Name	<input type="text"/>
Email	<input type="text"/>
Mobile	<input type="text"/> City <input type="text"/>

Delete Details

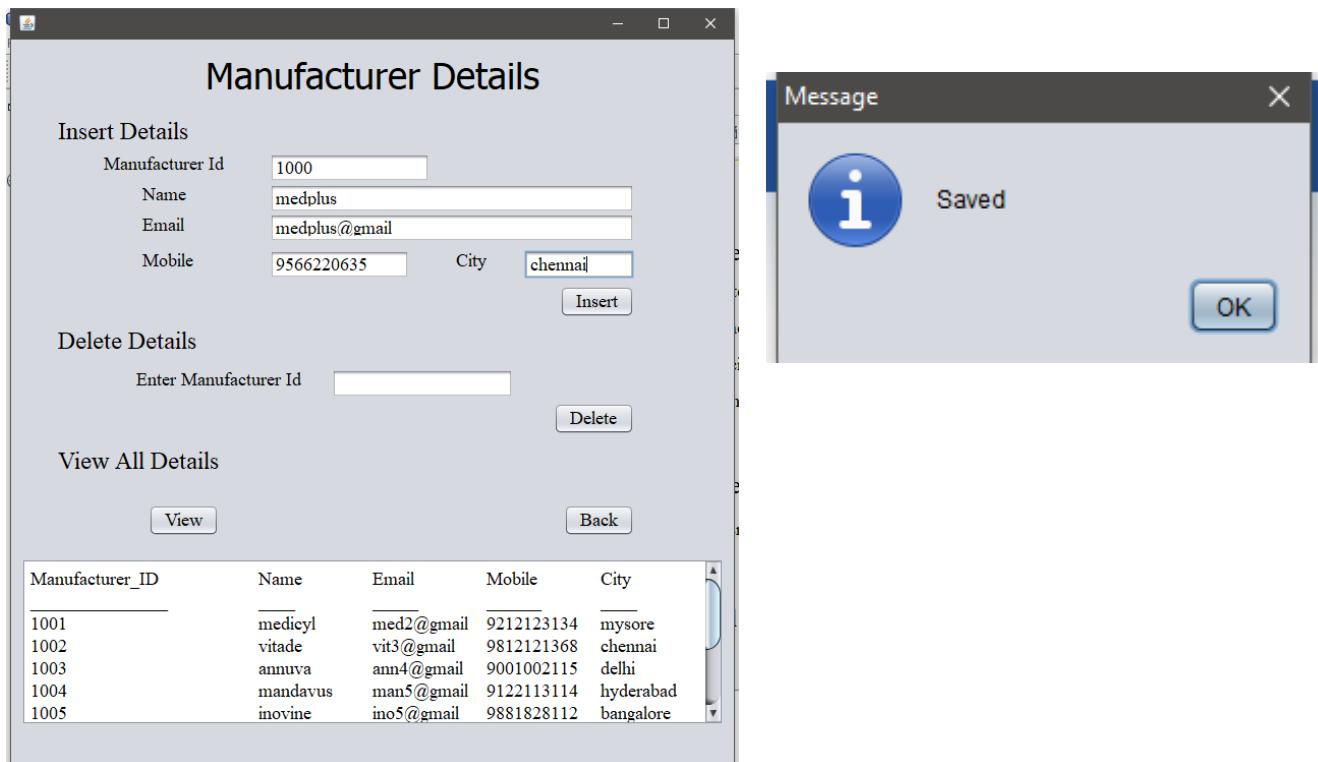
Enter Manufacturer Id

View All Details

Manufacturer_ID	Name	Email	Mobile	City
1000	hoivo	hoivo1@gmail.com	9112113114	Bangalore
1001	medicyl	med2@gmail.com	9212123134	Mysore
1002	vitade	vit3@gmail.com	9812121368	Chennai
1003	annuva	ann4@gmail.com	9001002115	Delhi
1004	mandavus	man5@gmail.com	9122113114	Hyderabad

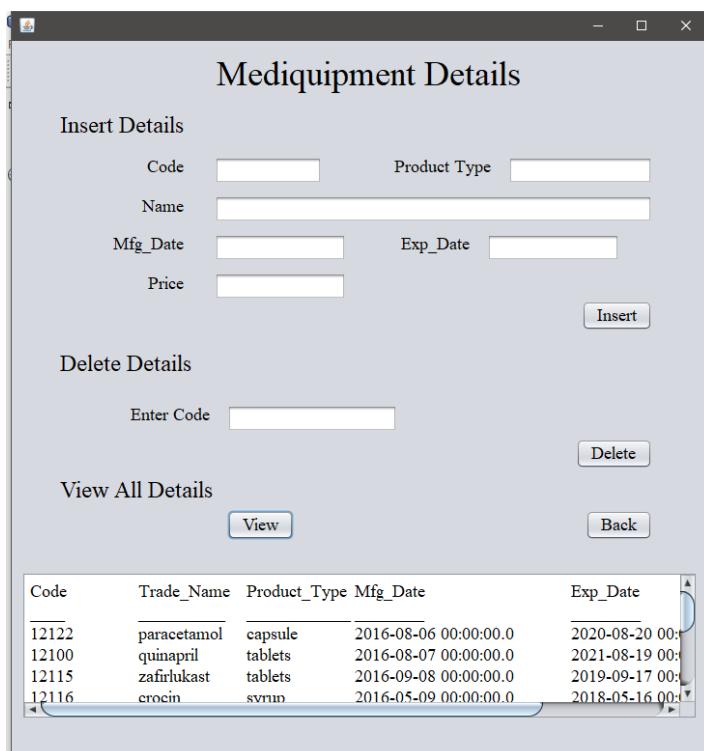


Inserting the details,



10)Mediquipment Page :-

This page lets the user to insert data into the database from the front end, delete the data that the user wants to delete and view the database that is stored in the database.



After deletion,

Mediquipment Details

Insert Details

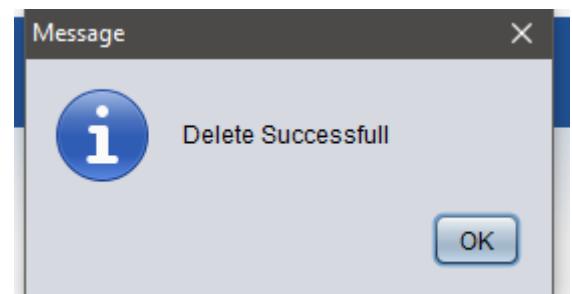
Code	<input type="text"/>	Product Type	<input type="text"/>
Name	<input type="text"/>		
Mfg_Date	<input type="text"/>	Exp_Date	<input type="text"/>
Price	<input type="text"/>		

Delete Details

Enter Code

View All Details

Code	Trade_Name	Product_Type	Mfg_Date	Exp_Date
12122	paracetamol	capsule	2016-08-06 00:00:00.0	2020-08-20 00:00:00.0
12100	quinapril	tablets	2016-08-07 00:00:00.0	2021-08-19 00:00:00.0
12115	zaflurukast	tablets	2016-09-08 00:00:00.0	2019-09-17 00:00:00.0
12116	crocin	syrup	2016-05-09 00:00:00.0	2018-05-16 00:00:00.0



Inserting the details,

Mediquipment Details

Insert Details

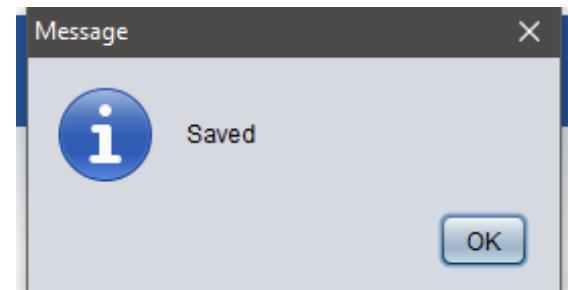
Code	<input type="text" value="12122"/>	Product Type	<input type="text" value="tablets"/>
Name	<input type="text" value="paracetamol"/>		
Mfg_Date	<input type="text" value="2016-08-07"/>	Exp_Date	<input type="text" value="2021-08-19"/>
Price	<input type="text" value="20"/>		

Delete Details

Enter Code

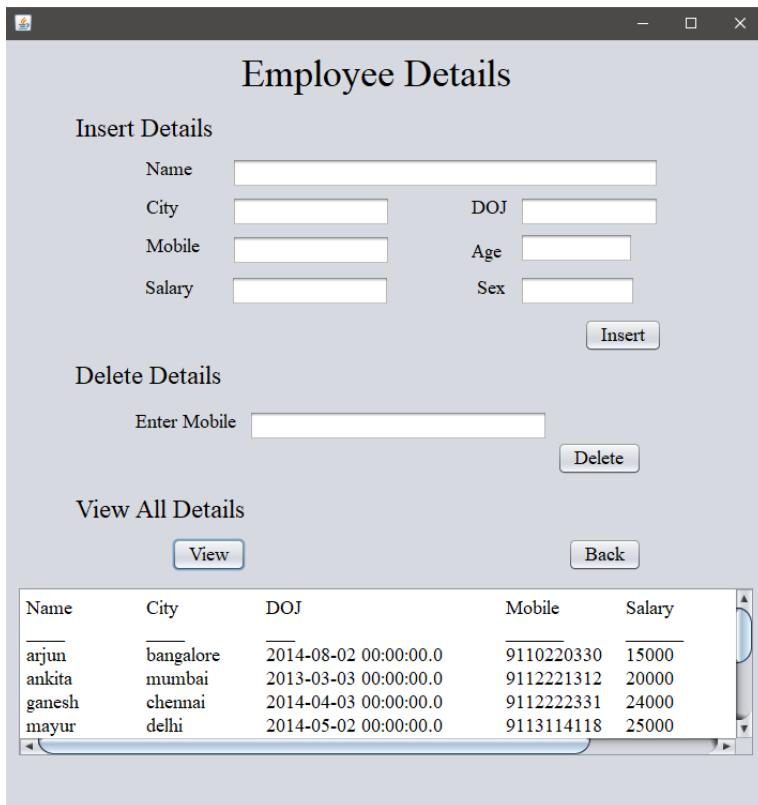
View All Details

Code	Trade_Name	Product_Type	Mfg_Date	Exp_Date
12100	quinapril	tablets	2016-08-07 00:00:00.0	2021-08-19 00:00:00.0
12115	zaflurukast	tablets	2016-09-08 00:00:00.0	2019-09-17 00:00:00.0
12116	crocin	syrup	2016-05-09 00:00:00.0	2018-05-16 00:00:00.0
12117	injection	injection	2016-06-18 00:00:00.0	2023-04-23 00:00:00.0

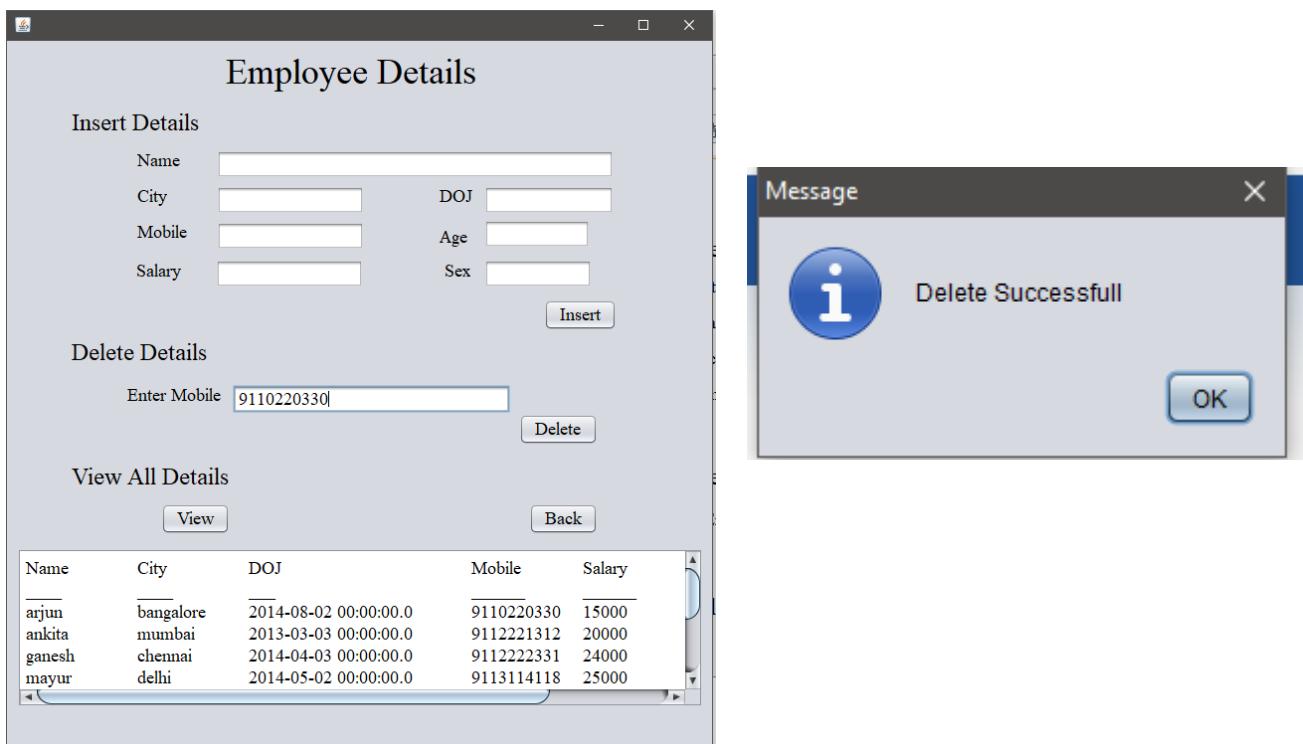


11)Employee Page :-

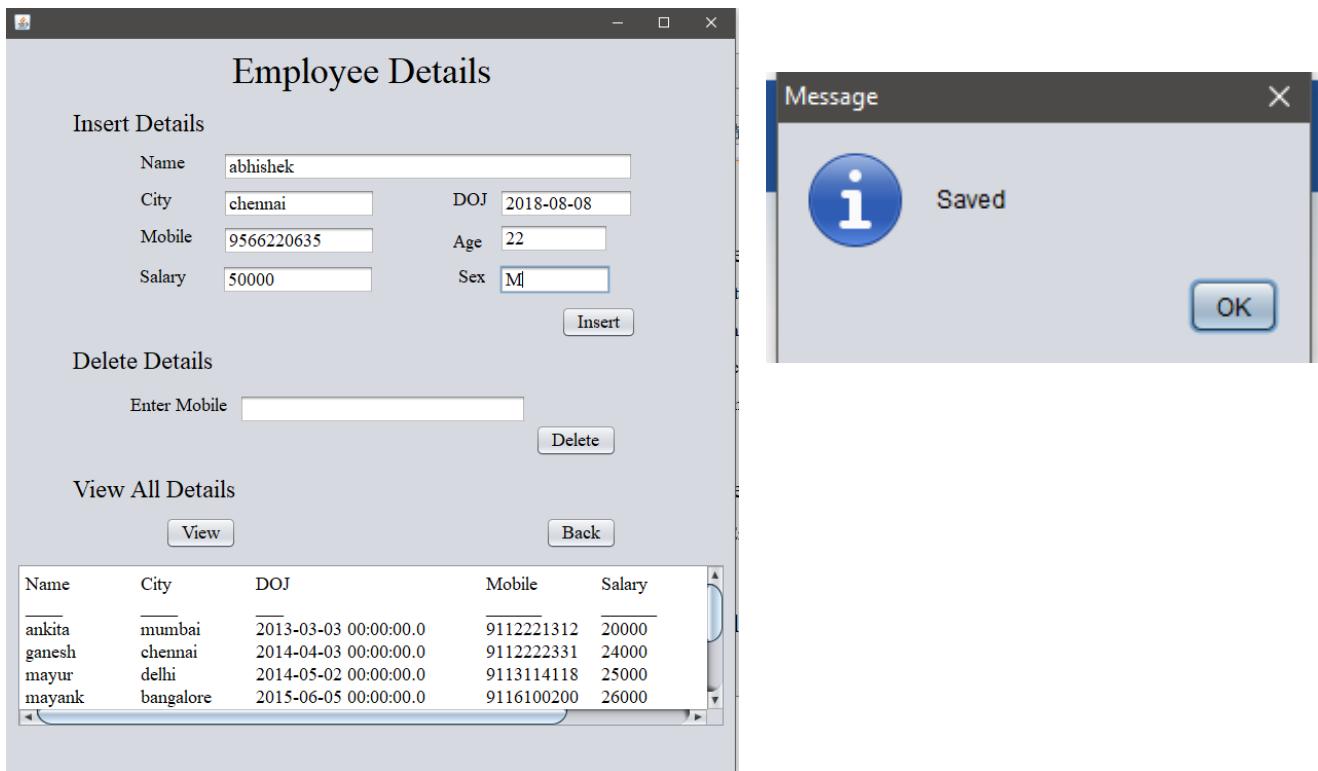
This page lets the user to insert data into the database from the front end, delete the data that the user wants to delete and view the database that is stored in the database.



After deletion,

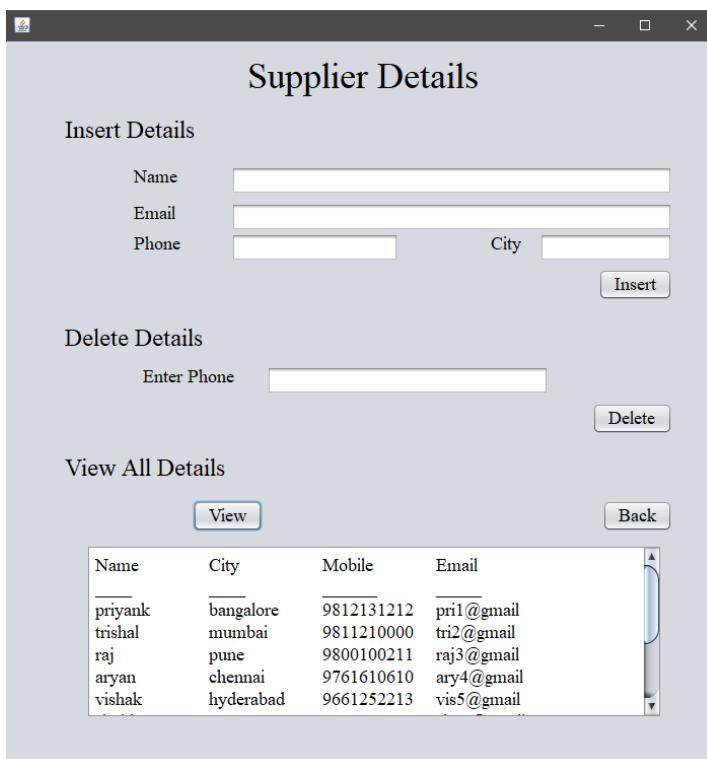


Inserting the details,



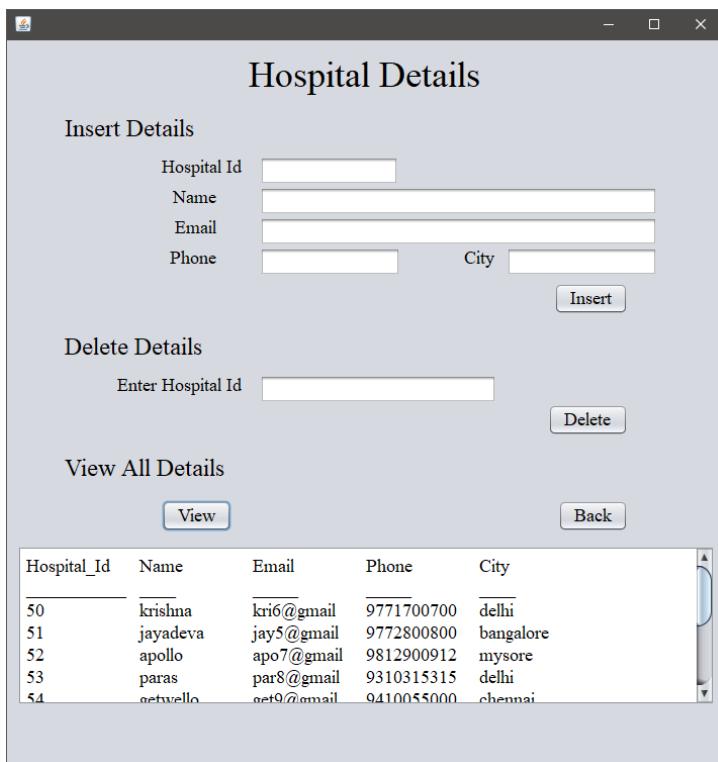
12) Supplier Page :-

This page lets the user to insert data into the database from the front end, delete the data that the user wants to delete and view the database that is stored in the database.



13) Hospital Page :-

This page lets the user to insert data into the database from the front end, delete the data that the user wants to delete and view the database that is stored in the database.



SQL QUERY

- 1) This query displays the name and salary of all employee's whose salary is greater than the average salary of all employee's working in Bangalore.

```
SQL> select name,salary
  2      from employee
  3      group by name,salary
  4      having salary > (select avg(salary)
  5                          from employee
  6                          where city='bangalore')
  7      order by 2 asc;

NAME                      SALARY
-----
ganesh                    24000
mayur                     25000
mayank                    26000
satyam                    27000
```

- 2) This query updates the medicine's price in medquipment table to 300, if the manufacturer is from Pune.

CODE	TRADE_NAME	PRODUCT_TYPE	MFG_DATE	EXP_DATE
PRICE	CID			
12205	crocin 499	syrup	15-DEC-17	15-DEC-18
12112	paramax 308.16	tablets	29-APR-16	20-FEB-20

11 rows selected.

```
SQL> update medquipment set price=300
2   where cid in (select cid
3     from manufacturer
4       where city='pune');

1 row updated.
```

(Before execution)

CODE	TRADE_NAME	PRODUCT_TYPE	MFG_DATE	EXP_DATE
PRICE	CID			
12205	crocin 499	syrup	15-DEC-17	15-DEC-18
12112	paramax 300	tablets	29-APR-16	20-FEB-20

11 rows selected.

```
SQL>
```

(After execution)

- 3) This query displays the name of pharmacy, pharmacy id and the count of the manufacturer id, if a pharmacy is taking medicine's from more than one manufacturer.

```
SQL> select p.name,pharid,count(distinct cid) as count1
2   from pharmacy p, manufacturer
3   where phid = pharid
4   group by p.name, pharid
5   having count(*)>1;

NAME          PHARID      COUNT1
-----        -----
abbott           1            2
allergan         2            3

SQL> |
```

- 4) This query displays the pharmacy name, employee name and employee salary, who are working in a pharmacy except the female employee.

```
SQL> select p.name, e.name, e.salary
  2      from employee e, pharmacy p
  3      where phid=pharid
  4      minus
  5      select p.name,e.name, e.salary
  6      from employee e, pharmacy p
  7      where e.sex='f';

NAME          NAME          SALARY
-----        -----        -----
abbott        arjun         15000
janssen       ganesh        24000
novartis     daniel         19000
roche         mayank         26000
schwarz      satyam         27000
```

PROCEDURE :-

This procedure update's the medicine price by 100 in the medquipment table, where the trade name is paramax.

```
CODE TRADE_NAME      PRODUCT_TYPE      MFG_DATE   EXP_DATE
-----  -----
PRICE    CID
-----  -----
12205 crocin           syrup        15-DEC-17 15-DEC-18
499

12112 paramax          tablets      29-APR-16 20-FEB-20
300      1009

11 rows selected.

SQL> drop procedure update_medi;
Procedure dropped.

SQL> create procedure update_medi as
  2  begin
  3    update medquipment set price = price+100 where trade_name='paramax';
  4  end;
  5  /

Procedure created.

SQL> begin
  2  update_medi;
  3  end;
  4  /

PL/SQL procedure successfully completed.
```

(Before procedure is executed)

CODE	TRADE_NAME	PRODUCT_TYPE	MFG_DATE	EXP_DATE
PRICE	CID			
-----	-----			
12205	crocin	syrup	15-DEC-17	15-DEC-18
499				
12112	paramax	tablets	29-APR-16	20-FEB-20
400	1009			

(After procedure is executed)

Triggers:

- 1) This trigger will change the price of a medicine to 499, if the price is greater than 500.

```
SQL> create trigger price_check
  2  before insert or update on medquipment
  3  for each row
  4  when (new.price>500)
  5  begin
  6  :new.price:= 499;
  7  end;
  8 /

Trigger created.

SQL> insert into medquipment values(12205,'vinyl','syrup','13-sep-16','13-mar-18',550,1003);
```

(Trigger is executed and a insertion is performed)

CODE	TRADE_NAME	PRODUCT_TYPE	MFG_DATE	EXP_DATE
PRICE	CID			
12205	vinyl	syrup	13-SEP-16	13-MAR-18
499	1003			
12112	paramax	tablets	29-APR-16	20-FEB-20
400	1009			

(Inserted price is less than 500)

- 2) This trigger will display a message if the age of a doctor is greater than 80.

```
SQL> create trigger age_check1
  2  before insert or update on doctor
  3  for each row
  4  when (new.age>80)
  5  begin
  6  dbms_output.put_line('DOCTOR AGE CANNOT BE MORE THAN 80!!!');
  7  end;
  8 /

Trigger created.

SQL> set serveroutput on;
SQL> insert into doctor values(50,'abhishek','cardiologist',85,9566220635,'m');
DOCTOR AGE CANNOT BE MORE THAN 80!!!
```

Database creation commands

Connection commands:

```
connect system/gokhul;  
create user medicalshop22 identified by gokhul;  
grant connect, resource to medicalshop22;  
connect medicalshop22/gokhul;  
@ C:\Users\gokhu\Desktop\pharmacy.sql
```

Database commands:

```
create table pharmacy  
(  
    phid integer primary key,  
    name varchar2(20),  
    city varchar2(20),  
    fax varchar2(20),  
    phone number(10));
```

```
create table doctor  
(  
    did number(2) primary key,  
    dname varchar2(20),  
    speciality varchar2(20),  
    age number(2) not null,  
    mobile integer,  
    gender varchar2(6));
```

```
create table customer  
(  
    pid number(3) primary key,  
    name varchar2(20),
```

```
sex varchar2(6),
city varchar2(20),
phone integer,
age number(2),
did references doctor(did) on delete set null);
```

```
create table manufacturer
(
cid number(4) primary key,
name varchar2(20),
email varchar2(20),
mobile integer,
city varchar2(20),
pharid references pharmacy(phid) on delete set null);
```

```
create table medquipment
(
code number(5) primary key,
Trade_name varchar2(20),
Product_Type varchar2(20),
Mfg_date date,
Exp_date date,
price number(10,2),
cid references manufacturer(cid) on delete set null);
```

```
create table supplier
(
name varchar2(20),
city varchar2(20),
mobile integer primary key,
```

```
email varchar2(20),  
cid references manufacturer(cid) on delete set null,  
pharid references pharmacy(phid) on delete set null);
```

```
create table employee  
(  
    name varchar2(20),  
    city varchar2(20),  
    DOJ date,  
    mobile integer primary key,  
    salary number(10),  
    age number(2),  
    sex varchar2(1),  
    pharid references pharmacy(phid) on delete set null);
```

```
create table hospital  
(  
    hid number(2) primary key,  
    name varchar2(20),  
    email varchar2(40),  
    phone integer,  
    city varchar2(20),  
    pharid references pharmacy(phid) on delete set null);
```

```
create table bill  
(  
    bid number(3) primary key,  
    DOB date,  
    age number(3),  
    pname varchar2(20),
```

```
mobile integer,  
city varchar2(20),  
product varchar2(20),  
amount number(10,2),  
pharid references pharmacy(phid) on delete set null);
```

```
create table works  
(  
pharid references pharmacy(phid) on delete cascade,  
start_date date,  
end_date date,  
primary key(pharid));
```

```
create table contract  
(  
pharid references pharmacy(phid) on delete cascade,  
cid references manufacturer(cid) on delete cascade,  
start_date date,  
end_date date,  
primary key(pharid,cid));
```

```
create table prescribe  
(  
DOP date,  
medicine varchar2(20),  
did references doctor(did) on delete cascade,  
pid references customer(pid) on delete cascade,  
primary key(pid,did));
```

```
select name,salary  
      from employee  
      group by name,salary  
      having salary > (select avg(salary)  
                        from employee  
                        where city='bangalore')  
      order by 2 asc;
```

```
update medquipment set price=300  
      where cid in (select cid  
                     from manufacturer  
                     where city='pune');
```

```
select p.name,pharid,count(distinct cid) as count1  
      from pharmacy p, manufacturer  
      where phid = pharid  
      group by p.name, pharid  
      having count(*)>1;
```

```
select p.name, e.name, e.salary  
      from employee e, pharmacy p  
      where phid=pharid  
      minus  
      select p.name,e.name, e.salary  
      from employee e, pharmacy p  
      where e.sex='f';
```

```
create procedure update_medi as
begin
update medquipment set price = price+100 where trade_name='paramax';
end;
/
```

```
create trigger price_check
before insert or update on medquipment
for each row
when (new.price>500)
begin
:new.price:= 499;
end;
/
```

```
create trigger age_check1
before insert or update on doctor
for each row
when (new.age>80)
begin
dbms_output.put_line('DOCTOR AGE CANNOT BE
MORE THAN 80!!!!');
end;
/
```

```
commit;
```

Application Code

1) Billing Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    try{Class.forName("oracle.jdbc.OracleDriver");  
  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-C3RN1T1:1521:XE","medicalshop22","gokhul");  
  
        String sql="Insert into bill(bid,dob,age,pname,mobile,city,product,amount) values(?,?,?,?,?,?,?,?)";  
  
        PreparedStatement pst=con1.prepareStatement(sql);  
  
        pst.setString(1,b.getText());  
  
        pst.setString(2,a.getText());  
  
        pst.setString(3,c.getText());  
  
        pst.setString(4,d.getText());  
  
        pst.setString(5,e.getText());  
  
        pst.setString(6,f.getText());  
  
        pst.setString(7,g.getText());  
  
        pst.setString(8,h.getText());  
  
        pst.execute();  
  
  
        JOptionPane.showMessageDialog(null,"Saved");  
        pst.close();  
        con1.close();  
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

2) Pharmacy Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    try{Class.forName("oracle.jdbc.OracleDriver");  
  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22","gokhul");  
  
        String sql="Insert into pharmacy(phid,name,city,fax,phone) values(?,?,?,?,?)";  
  
        PreparedStatement pst=con1.prepareStatement(sql);  
  
        pst.setString(1,jTextField1.getText());  
  
        pst.setString(2,jTextField3.getText());  
  
        pst.setString(3,jTextField2.getText());  
  
        pst.setString(4,jTextField4.getText());  
  
        pst.setString(5,jTextField5.getText());  
  
        pst.execute();  
  
        jTextField1.setText("");  
  
        jTextField2.setText("");  
  
        jTextField4.setText("");  
  
        jTextField3.setText("");  
  
        jTextField5.setText("");  
  
        JOptionPane.showMessageDialog(null,"Saved");  
  
        pst.close();  
  
        con1.close();  
  
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

```

private void viewActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    try{Class.forName("oracle.jdbc.OracleDriver");

        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
C3RN1T1:1521:XE","medicalshop22"," gokhul");

        Statement stmt=con1.createStatement();

        ResultSet rs=stmt.executeQuery("select * from pharmacy");

        String show="";
        show="Pharmacy_ID\t"+ "Pharmacy_Name\t"+ "City\t"+ "Fax\t"+ "Phone\n";
        show+="_____ \t" + "_____ \t" + "____ \t" + "___ \t" + "____ \n";
        while(rs.next())

        {
            int phid=rs.getInt("phid");
            String name=rs.getString("name");
            String city=rs.getString("city");
            String fax = rs.getString("fax");
            String phone=rs.getString("phone");
            show+=phid+"\t"+name+"\t"+city+"\t"+fax+"\t"+phone+"\n";
            jTextArea1.setText(show);
        }
        rs.close();
        stmt.close();
        con1.close();
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}

}

```

```
private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22"," gokhul");  
  
        String phid = jTextField6.getText();  
  
        String sql = "delete from pharmacy where phid="+phid;  
  
        PreparedStatement stmt2=con1.prepareStatement(sql);  
  
        stmt2.execute();  
  
        jTextField6.setText("");  
  
        JOptionPane.showMessageDialog(null,"Delete Successfull");  
    }  
  
    catch(Exception e){JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

3) Doctor Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    try{Class.forName("oracle.jdbc.OracleDriver");  
  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22","gokhul");  
  
        String sql="Insert into doctor(did,dname,speciality,age,mobile,gender) values(?,?,?,?,?,?)";  
  
        PreparedStatement pst=con1.prepareStatement(sql);  
  
        pst.setString(1,jTextField1.getText());  
  
        pst.setString(2,jTextField3.getText());  
  
        pst.setString(3,jTextField4.getText());  
  
        pst.setString(4,jTextField2.getText());  
  
        pst.setString(5,jTextField6.getText());  
  
        pst.setString(6,jTextField5.getText());  
  
        pst.execute();  
  
        jTextField1.setText("");  
  
        jTextField2.setText("");  
  
        jTextField4.setText("");  
  
        jTextField3.setText("");  
  
        jTextField5.setText("");  
  
        jTextField6.setText("");  
  
        JOptionPane.showMessageDialog(null,"Saved");  
  
        pst.close();  
  
        con1.close();  
  
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

```

private void viewActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");

        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
C3RN1T1:1521:XE","medicalshop22","gokhul");

        Statement stmt=con1.createStatement();

        ResultSet rs=stmt.executeQuery("select * from doctor");

        String show="";
        show="Doctor_ID\t"+ "Doctor_Name\t"+ "Speciality\t"+ "Age\t"+ "Mobile\t"+ "Gender\n";
        show+="_____ \t" + "_____ \t" + "_____ \t" + "___\t" + "____ \t" + "____ \n";
        while(rs.next())
        {
            int did=rs.getInt("did");
            String dname=rs.getString("dname");
            String speciality=rs.getString("speciality");
            String age = rs.getString("age");
            String mobile=rs.getString("mobile");
            String gender=rs.getString("gender");
            show+=did+"\t"+dname+"\t"+speciality+"\t"+age+"\t"+mobile+"\t"+gender+"\n";
            jTextArea1.setText(show);
        }
        rs.close();
        stmt.close();
        con1.close();
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}
}

```

```
private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22"," gokhul");  
  
        String did = jTextField7.getText();  
  
        String sql = "delete from doctor where did="+did;  
  
        PreparedStatement stmt2=con1.prepareStatement(sql);  
  
        stmt2.execute();  
  
        jTextField7.setText("");  
  
        JOptionPane.showMessageDialog(null,"Delete Successfull");  
    }  
  
    catch(Exception e){JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

4) Customer Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    try{Class.forName("oracle.jdbc.OracleDriver");  
  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22","gokhul");  
  
        String sql="Insert into customer(pid,name,sex,city,phone,age) values(?,?,?,?,?,?)";  
  
        PreparedStatement pst=con1.prepareStatement(sql);  
  
        pst.setString(1,jTextField1.getText());  
  
        pst.setString(2,jTextField3.getText());  
  
        pst.setString(3,jTextField5.getText());  
  
        pst.setString(4,jTextField4.getText());  
  
        pst.setString(5,jTextField6.getText());  
  
        pst.setString(6,jTextField2.getText());  
  
        pst.execute();  
  
        jTextField1.setText("");  
  
        jTextField2.setText("");  
  
        jTextField4.setText("");  
  
        jTextField3.setText("");  
  
        jTextField5.setText("");  
  
        jTextField6.setText("");  
  
        JOptionPane.showMessageDialog(null,"Saved");  
  
        pst.close();  
  
        con1.close();  
  
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

```

private void viewActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    try{Class.forName("oracle.jdbc.OracleDriver");

        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
C3RN1T1:1521:XE","medicalshop22","gokhul");

        Statement stmt=con1.createStatement();

        ResultSet rs=stmt.executeQuery("select * from customer");

        String show="";

        show="Customer_ID\t"+Customer_Name\t"+Sex\t"+City\t"+Phone\t"+Age\n";
        show+="_____ \t" + _____ \t" + __ \t" + __ \t" + __ \t" + __ \n";
        while(rs.next())

        {

            int pid=rs.getInt("pid");

            String name=rs.getString("name");

            String sex=rs.getString("sex");

            String city = rs.getString("city");

            String phone=rs.getString("phone");

            String age=rs.getString("age");

            show+=pid+"\t"+name+"\t"+sex+"\t"+city+"\t"+phone+"\t"+age+"\n";

            jTextArea1.setText(show);

        }

        rs.close();

        stmt.close();

        con1.close();

    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}

}

```

```
private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22","gokhul");  
  
        String pid = jTextField7.getText();  
  
        String sql = "delete from customer where pid="+pid;  
  
        PreparedStatement stmt2=con1.prepareStatement(sql);  
  
        stmt2.execute();  
  
        jTextField7.setText("");  
  
        JOptionPane.showMessageDialog(null,"Delete Successfull");  
    }  
  
    catch(Exception e){JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

5) Manufacturer Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    try{Class.forName("oracle.jdbc.OracleDriver");  
  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22","gokhul");  
  
        String sql="Insert into manufacturer(cid,name,email,mobile,city) values(?,?,?,?,?)";  
  
        PreparedStatement pst=con1.prepareStatement(sql);  
  
        pst.setString(1,jTextField1.getText());  
  
        pst.setString(2,jTextField2.getText());  
  
        pst.setString(3,jTextField3.getText());  
  
        pst.setString(4,jTextField4.getText());  
  
        pst.setString(5,jTextField5.getText());  
  
        pst.execute();  
  
        jTextField1.setText("");  
  
        jTextField2.setText("");  
  
        jTextField4.setText("");  
  
        jTextField3.setText("");  
  
        jTextField5.setText("");  
  
        JOptionPane.showMessageDialog(null,"Saved");  
  
        pst.close();  
  
        con1.close();  
  
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

```

private void viewActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    try{Class.forName("oracle.jdbc.OracleDriver");

        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
C3RN1T1:1521:XE","medicalshop22","gokhul");

        Statement stmt=con1.createStatement();

        ResultSet rs=stmt.executeQuery("select * from manufacturer");

        String show="";

        show="Manufacturer_ID\t"+ "Name\t"+ "Email\t"+ "Mobile\t"+ "City\n";
        show+="_____ \t" + "____ \t" + "____ \t" + "____ \n";
        while(rs.next())

        {
            int cid=rs.getInt("cid");
            String name=rs.getString("name");
            String email=rs.getString("email");
            String mobile = rs.getString("mobile");
            String city=rs.getString("city");
            show+=cid+"\t"+name+"\t"+email+"\t"+mobile+"\t"+city+"\n";
            jTextArea1.setText(show);
        }
        rs.close();
        stmt.close();
        con1.close();
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}

}

```

```
private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22"," gokhul");  
  
        String cid = jTextField6.getText();  
  
        String sql = "delete from manufacturer where cid="+cid;  
  
        PreparedStatement stmt2=con1.prepareStatement(sql);  
  
        stmt2.execute();  
  
        jTextField6.setText("");  
  
        JOptionPane.showMessageDialog(null,"Delete Successfull");  
    }  
  
    catch(Exception e){JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

6) Mediquipment Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    try{Class.forName("oracle.jdbc.OracleDriver");  
  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22","gokhul");  
  
        String sql="Insert into mediquipment(code,trade_name,product_type,mfg_date,exp_date,price)  
values(?,?,?,?,?,?)";  
  
        PreparedStatement pst=con1.prepareStatement(sql);  
  
        pst.setString(1,jTextField1.getText());  
  
        pst.setString(2,jTextField3.getText());  
  
        pst.setString(3,jTextField2.getText());  
  
        pst.setString(4,jTextField4.getText());  
  
        pst.setString(5,jTextField5.getText());  
  
        pst.setString(6,jTextField6.getText());  
  
        pst.execute();  
  
        jTextField1.setText("");  
  
        jTextField2.setText("");  
  
        jTextField4.setText("");  
  
        jTextField3.setText("");  
  
        jTextField5.setText("");  
  
        jTextField6.setText("");  
  
        JOptionPane.showMessageDialog(null,"Saved");  
  
        pst.close();  
  
        con1.close();  
  
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

```

private void viewActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-C3RN1T1:1521:XE","medicalshop22","gokhul");
        Statement stmt=con1.createStatement();
        ResultSet rs=stmt.executeQuery("select * from medquipment");
        String show="";
        show="Code\t"+"Trade_Name\t"+"Product_Type\t"+"Mfg_Date\t\t"+"Exp_Date\t\t"+"Price\n";
        show+="____\t"+_____"\t"+_____"\t"+_____"\t\t"+_____"\t\t"+_____"\t\t"+____"\n";
        while(rs.next())
        {
            int code=rs.getInt("code");
            String trade_name=rs.getString("trade_name");
            String product_type=rs.getString("product_type");
            String mfg_date = rs.getString("mfg_date");
            String exp_date=rs.getString("exp_date");
            int price=rs.getInt("price");
            show+=code+"\t"+trade_name+"\t"+product_type+"\t"+mfg_date+"\t"+exp_date+"\t"+price+"\n";
            jTextArea1.setText(show);
        }
        rs.close();
        stmt.close();
        con1.close();
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}
}

```

```
private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22"," gokhul");  
  
        String code = jTextField7.getText();  
  
        String sql = "delete from medquipment where code="+code;  
  
        PreparedStatement stmt2=con1.prepareStatement(sql);  
  
        stmt2.execute();  
  
        jTextField6.setText("");  
  
        JOptionPane.showMessageDialog(null,"Delete Successfull");  
    }  
  
    catch(Exception e){JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

7) Supplier Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    try{Class.forName("oracle.jdbc.OracleDriver");  
  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22","gokhul");  
  
        String sql="Insert into supplier(name,city,mobile,email) values(?, ?, ?, ?)";  
  
        PreparedStatement pst=con1.prepareStatement(sql);  
  
        pst.setString(1,jTextField1.getText());  
  
        pst.setString(2,jTextField4.getText());  
  
        pst.setString(3,jTextField3.getText());  
  
        pst.setString(4,jTextField2.getText());  
  
        pst.execute();  
  
        jTextField1.setText("");  
  
        jTextField2.setText("");  
  
        jTextField4.setText("");  
  
        jTextField3.setText("");  
  
        JOptionPane.showMessageDialog(null,"Saved");  
  
        pst.close();  
  
        con1.close(); }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);} } }
```

```

private void viewActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    try{Class.forName("oracle.jdbc.OracleDriver");

        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
C3RN1T1:1521:XE","medicalshop22","gokhul");

        Statement stmt=con1.createStatement();

        ResultSet rs=stmt.executeQuery("select * from supplier");

        String show="";
        show="Name\t"+ "City\t"+ "Mobile\t"+ "Email\n";
        show+="____\t"+____\t"+_____ \t"+____\n";
        while(rs.next())
        {
            String name = rs.getString("name");
            String city=rs.getString("city");
            String mobile=rs.getString("mobile");
            String email = rs.getString("email");
            show+=name+"\t"+city+"\t"+mobile+"\t"+email+"\n";
            jTextArea1.setText(show);
        }
        rs.close();
        stmt.close();
        con1.close();
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}
}

```

```
private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22"," gokhul");  
  
        String mobile = jTextField5.getText();  
  
        String sql = "delete from supplier where mobile="+mobile;  
  
        PreparedStatement stmt2=con1.prepareStatement(sql);  
  
        stmt2.execute();  
  
        jTextField5.setText("");  
  
        JOptionPane.showMessageDialog(null,"Delete Successfull");  
    }  
  
    catch(Exception e){JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

8) Employee Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    try{Class.forName("oracle.jdbc.OracleDriver");  
  
        Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-C3RN1T1:1521:XE","medicalshop22","gokhul");  
  
        String sql="Insert into employee(name,city,doj,mobile,salary,age,sex) values(?,?,?,?,?,?)";  
  
        PreparedStatement pst=con1.prepareStatement(sql);  
  
        pst.setString(1,jTextField1.getText());  
        pst.setString(2,jTextField2.getText());  
        pst.setString(3,jTextField3.getText());  
        pst.setString(4,jTextField4.getText());  
        pst.setString(5,jTextField6.getText());  
        pst.setString(6,jTextField5.getText());  
        pst.setString(7,jTextField7.getText());  
  
        pst.execute();  
  
        jTextField1.setText("");  
        jTextField2.setText("");  
        jTextField4.setText("");  
        jTextField3.setText("");  
        jTextField5.setText("");  
        jTextField6.setText("");  
        jTextField7.setText("");  
  
        JOptionPane.showMessageDialog(null,"Saved");  
        pst.close();  
        con1.close();  
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

```

private void viewActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");

        Connection
con1=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","medicalshop22"," gokhul");

        Statement stmt=con1.createStatement();

        ResultSet rs=stmt.executeQuery("select * from employee");

        String show="";
        show="Name\t"+ "City\t" + "DOJ\t" + "Mobile\t" + "Salary\t" + "Age\t" + "Sex\n";
        show+="____\t" + "____\t" + "____\t" + "____\t" + "____\t" + "____\t" + "____\n";
        while(rs.next())
        {
            String name=rs.getString("name");
            String city=rs.getString("city");
            String doj=rs.getString("doj");
            String mobile = rs.getString("mobile");
            int salary=rs.getInt("salary");
            int age=rs.getInt("age");
            String sex=rs.getString("sex");
            show+=name+"\t"+city+"\t"+doj+"\t"+mobile+"\t"+salary+"\t"+age+"\t"+sex+"\n";
            jTextArea1.setText(show);
        }
        rs.close();
        stmt.close();
        con1.close();
    }catch(Exception e) { JOptionPane.showMessageDialog(null,"Error="+e);}
}

```

```
private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
        Connection  
con1=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","medicalshop22"," gokhul");  
        String mobile = jTextField8.getText();  
        String sql = "delete from employee where mobile="+mobile;  
        PreparedStatement stmt2=con1.prepareStatement(sql);  
        stmt2.execute();  
        jTextField8.setText("");  
        JOptionPane.showMessageDialog(null,"Delete Successfull");  
    }  
    catch(Exception e){JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

9) Hospital Page:

```
private void InsertActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    try{Class.forName("oracle.jdbc.OracleDriver");  
  
Connection con1=DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-  
C3RN1T1:1521:XE","medicalshop22","gokhul");  
  
String sql="Insert into hospital(hid,name,email,phone,city) values(?,?,?,?,?)";  
  
PreparedStatement pst=con1.prepareStatement(sql);  
  
pst.setString(1,jTextField1.getText());  
  
pst.setString(2,jTextField2.getText());  
  
pst.setString(3,jTextField3.getText());  
  
pst.setString(4,jTextField4.getText());  
  
pst.setString(5,jTextField5.getText());  
  
pst.execute();  
  
jTextField1.setText("");  
  
jTextField2.setText("");  
  
jTextField4.setText("");  
  
jTextField3.setText("");  
  
jTextField5.setText("");  
  
JOptionPane.showMessageDialog(null,"Saved");  
  
pst.close();  
  
con1.close();  
  
}catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```

```

private void viewActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:
    try{Class.forName("oracle.jdbc.OracleDriver");

        Connection
con1=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","medicalshop22"," gokhul");

        Statement stmt=con1.createStatement();

        ResultSet rs=stmt.executeQuery("select * from hospital");

        String show="";
        show="Hospital_Id\t"+ "Name\t"+ "Email\t"+ "Phone\t"+ "City\n";
        show+="_____ \t"+____\t"+____\t"+____\t"+____\n";
        while(rs.next())
        {
            int hid=rs.getInt("hid");
            String name=rs.getString("name");
            String email=rs.getString("email");
            String phone = rs.getString("phone");
            String city=rs.getString("city");
            show+=hid+"\t"+name+"\t"+email+"\t"+phone+"\t"+city+"\n";
            jTextArea1.setText(show);
        }
        rs.close();
        stmt.close();
        con1.close();
    }catch(Exception e) {JOptionPane.showMessageDialog(null,"Error="+e);}
}

```

```
private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try{Class.forName("oracle.jdbc.OracleDriver");  
        Connection  
con1=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","medicalshop22"," gokhul");  
        String hid = jTextField6.getText();  
        String sql = "delete from hospital where hid="+hid;  
        PreparedStatement stmt2=con1.prepareStatement(sql);  
        stmt2.execute();  
        jTextField6.setText("");  
        JOptionPane.showMessageDialog(null,"Delete Successfull");  
    }  
    catch(Exception e){JOptionPane.showMessageDialog(null,"Error="+e);}  
}
```