Ahmedabad University

BTech CSE Semester IV

<u>Course: CSE250 Database Management System</u>

Medico-helpers (Medical Shop Management System)

Group Members

AU1940280- Pankil Sheth

AU1940151- Purvam Sheth

AU1940116- Dhairya Purohit

Submitted To: **Prof. Shefali Naik**

Project Description.

- We have created a different login for different users. Here there are three main users: Medical shop owner, Doctor, Patient.
- So the medical shop owner after he/she login into the account can check the details of employee that are working in his medical shop, also user can check how much total amount of selling he/she has done, also can check the availability of the particular medicine, he/she can create a bill from this interface, he/she can check the stock of medicines available, also shop user can check which medicine should be expired by now, etc.
- Patients after they login can check the doctor's details by whom they got treated by now and also can add doctor by whom they are getting treatment currently.
- Doctor after they login can check patient details whom he/she has treated by now and also can add details of patients he is treating currently.
- Along with these three users, we have one more functionality called ‘Find a doctor’ which will give doctor details based on specialization entered, for e.g. pathologist, gynecologist, etc.
- In this project, we created a complete medical shop management system database in oracle DB (11g express edition), and with the help of NodeJS and React we tried to make a front end for displaying stored procedures and triggers.
- We have stored the data of many medical shops in which we included the shop ID, Name, City, Phone no, and email ID of respective shops.
- We have a billing table which contains records of a patient who bought which medicine from which medical shop on which date etc.
- We also have supplier and employee tables which are referred from the medical shop table. These employee and supplier tables contain the details of employees working at a particular medical shop and the supplier table contains details of suppliers of the particular medical shop respectively.
- Other than the medical shop perspective we have also widen our range of project and made relations between patient and doctor for e.g. which doctor has treated which patient and which patent has been treated by which doctor etc. This is a brief overview of the backend part along with the functionalities shown in our front end.

System Specification

Hardware:

- RAM: Minimum 1 GB physical memory, 4 GB recommended.
- Hard Disk: Minimum 500 MB free disk space for installation, 10 GB recommended.
- Processor: Intel Core i3 CPU @2.10GHz or above.

Software:

- Operating System: - Windows 7 or higher.
- Database Management System: - Oracle 11g or higher.
- Visual Studio Code.

User Interface:

- Front End: NodeJS, React.
- Back End: OracleDB 11g.

ERD tool:

- Draw.io

ER-DIAGRAM:

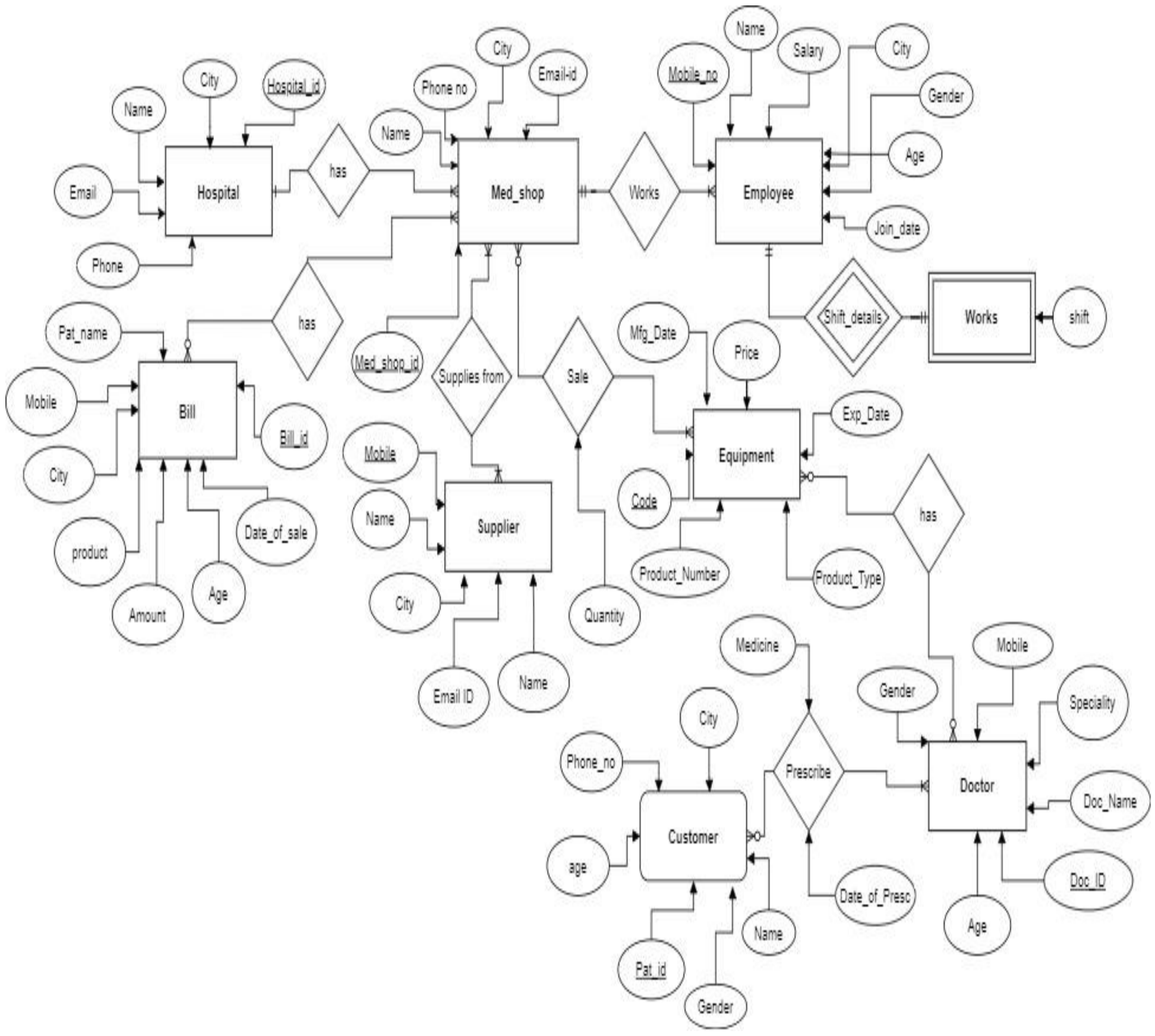

Hospital
Name
City
Hospital_id
Email
Phone
has
Med_shop
Phone no
City
Email-id
Name
Med_shop_id
Works
Employee
Mobile_no
Name
Salary
City
Gender
Age
Join_date
Shift_details
Works
shift
has
Bill
Pat_name
Mobile
City
product
Amount
Age
Date_of_sale
Bill_id
Supplies from
Supplier
Mobile
Name
City
Email ID
Name
Sale
Quantity
Equipment
Mfg_Date
Price
Exp_Date
Code
Product_Number
Product_Type
has
Doctor
Gender
Mobile
Speciality
Doc_Name
Doc_ID
Age
Prescribe
Medicine
Date_of_Presc
Customer
City
Phone_no
age
Pat_id
Gender
Name

Table Dictionary:

Med_Shop:

Column Name	Data Type	Constraints	Format	Description
Med_Shop_ID	Number	Primary Key	123	It is an unique id given to every medical shop.
Name	Varchar2 (20)	Not null	ABC	Name of the owner
City	Varchar2 (20)	Not Null	Delhi	City of the shop
Email_ID	Varchar2 (20)	Unique, Not Null	abc@gmail.com	Email ID of the shop
Phone	Number (10)	Unique, Not Null	1234567890	Contact no of the shop.

Employee:

Column_Name	Data Type	Constraints	Format	Description
Mobile_No	Number	Primary Key	1234567890	Mobile No of every employee working in a medical shop.
City	Varchar2(20)	Not Null	Mumbai	City of the employee
Join_Date	Date	Not Null	DD-MON-YY	Joining date of employee
Salary	Number (5)	Not Null	12345	Salary of employee
Age	Number (2)	Not Null	XX	Age of employee
Gender	Varchar2 (1)	Not Null	M/F	Gender of employee
Med_Shop_ID	Number	Foreign Key referencing Med_Shop (Med_Shop_ID)	123	It is a foreign key referring to med_shop_id which helps in integrating this table with med_shop table.

Works

Column_Name	Data Type	Constraints	Format	Description
Mobile_No	Number	Foreign Key referencing employee (Mobile_No)	1234567890	It is a foreign key referring to mobile_no which helps in integrating this table with the employee table.
Shift	Varchar2 (3)	Not Null	XXX	Shift of the employee

Doctor

Column_Name	Data Type	Constraints	Format	Description
Doc_ID	Number (3)	Primary Key	123	It is an unique id given to the doctor.
Doc_Name	Varchar2 (20)	Not Null	XYZ	Name of doctor
Age	Number (2)	Not Null	XX	Age of doctor
Speciality	Varchar2 (20)	Not Null	Cardiologist	Speciality of doctor
Gender	Varchar2 (6)	Not Null	M/F	Gender of doctor
Mobile	Number (10)	Unique, Not Null	1234567890	Contact No of Doctor

Equipment:

Column Name	Data Type	Constraints	Format	Description
Code	Number (4)	Primary Key	XXXX	Every product has an unique code.
Product_Name	Varchar2 (20)	Not Null	Crocin	Name of the product
Product_Type	Varchar2 (20)	Not Null	Tablet	Type of the product
MFG_Date	Date	Not Null	DD-Mon-YY	Manufacturing date of product

EXP_Date	Date	Not Null	DD-Mon-YY	Expiry date of product.
Price	Number (10,2)	Not Null	XX.YY	Price of the product
Doc_ID	Number (3)	Foreign Key referencing doctor (Doc_ID)	XXX	It is a foreign key referring to doc_id which helps in integrating this table with the doctor table.

Customer:

Column_Name	Data Type	Constraints	Format	Description
Pat_ID	Number (3)	Primary Key	XXX	It is an unique id given to the patients.
Name	Varchar2 (20)	Not Null	ABC	Name of the patient
Gender	Varchar2 (6)	Not Null	M/F	Gender of the patient
Age	Number (2)	Not Null	XX	Age of the patient
City	Varchar2 (20)	Not Null	Delhi	City of the patient
Phone	Number	Unique, Not Null	1234567890	Contact No of the patient
Doc_ID	Number (3)	Foreign Key referencing doctor (Doc_ID)	XXX	It is a foreign key referring to doc_id which helps in integrating this table with the doctor table.

Hospital:

Column_Name	Data Type	Constraints	Format	Description
Hospital_ID	Number (3)	Primary Key	XXX	It is an unique id given to hospitals.
Name	Varchar2 (20)	Not Null	ABC	Name of the hospital
Email	Varchar2 (40)	Unique, Not Null	abc@gmail.com	Email ID of the

				hospital
Phone	Number	Unique, Not Null	1234567890	Phone no of the hospital
City	Varchar2 (20)	Not Null	Chennai	City of the hospital
Med_Shop_ID	Number	Foreign Key referencing Med_Shop (Med_Shop_ID)	123	It is a foreign key referring to med_shop_id which helps in integrating this table with med_shop table.

Prescribe:

Column_Name	Data Type	Constraints	Format	Description
Doc_ID	Number (3)	Foreign Key referencing doctor (Doc_ID)	XXX	It is a foreign key referring to doc_id which helps in integrating this table with the doctor table.
Pat_ID	Number (3)	Foreign key referencing customer (Pat_ID)	XXX	It is a foreign key referring to pat_id which helps in integrating with this table with the customer table.
Medicine	Varchar2 (20)	Not Null	Crocin	Medicine prescribed
Date_of_Presc	Date	Not Null	DD-MON-YY	Date of prescription

Sale:

Column_Name	Data Type	Constraints	Format	Description
Med_Shop_ID	Number	PK (Combined)	123	It is a foreign key which is combined with code as a primary key which gives an idea of which medicines have been sold in which shop.

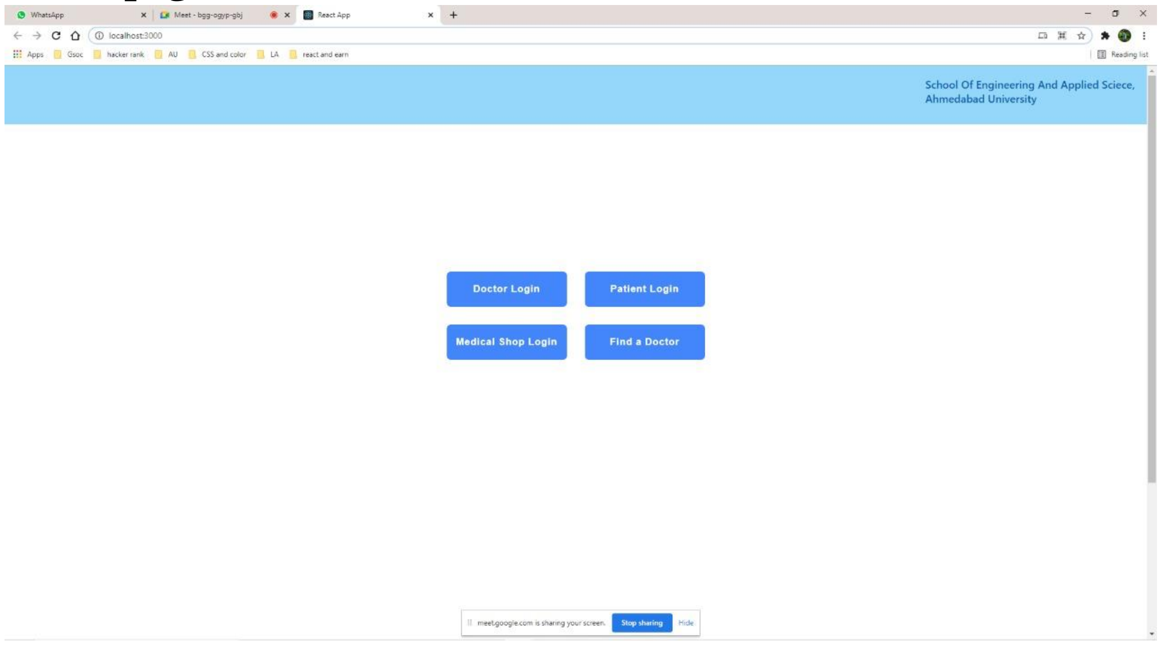
Code	Number (4)	PK (Combined)	XXXX	It is a foreign key which is combined with med_shop_id as a primary key which gives an idea of which medicines have been sold in which shop.
Quantity	Number	Not Null	1,2…	No of medicines sold

Bill:

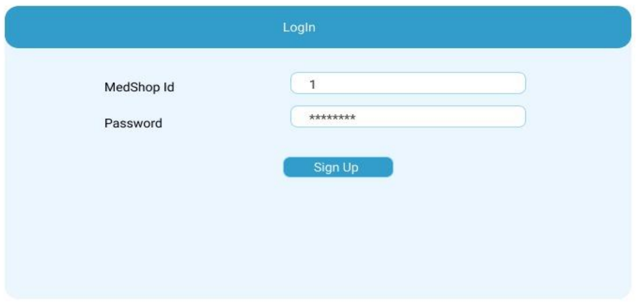
Column_Name	Data Type	Constraints	Format	Description
Bill_ID	Number (3)	Primary Key	XXX	It is an unique id given to bills.
Date_of_Sale	Date	Not Null	DD-MON-YY	Date on which equipment is sold.
Age	Number (3)	Not Null	XX	Age of patient
Pat_Name	Varchar2 (20)	Not Null	ABC	Name of Patient
Mobile	Number	Unique	1234567890	Contact no of patient
City	Varchar2 (20)	Not Null	Kolkata	City of Patient
Product	Varchar2 (20)	Not Null	Bandage	Name of products purchased.
Amount	Number (10,2)	Not Null	10.23	Total Amount Payable
Med_Shop_ID	Number	Foreign Key referencing Med_Shop (Med_Shop_ID)	123	It is a foreign key referring to med_shop_id which helps in integrating this table with med_shop table.

Procedures, Triggers and Snapshots

1) Home page



2) Med Shop login

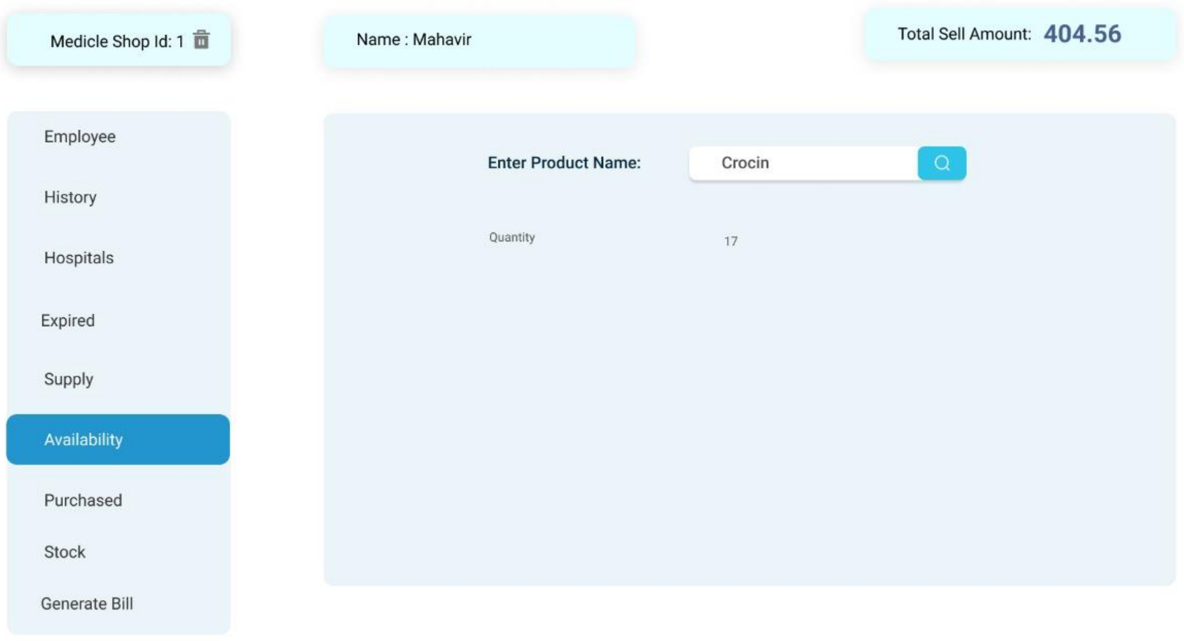


MedShop Id 1

Password ********

Procedures

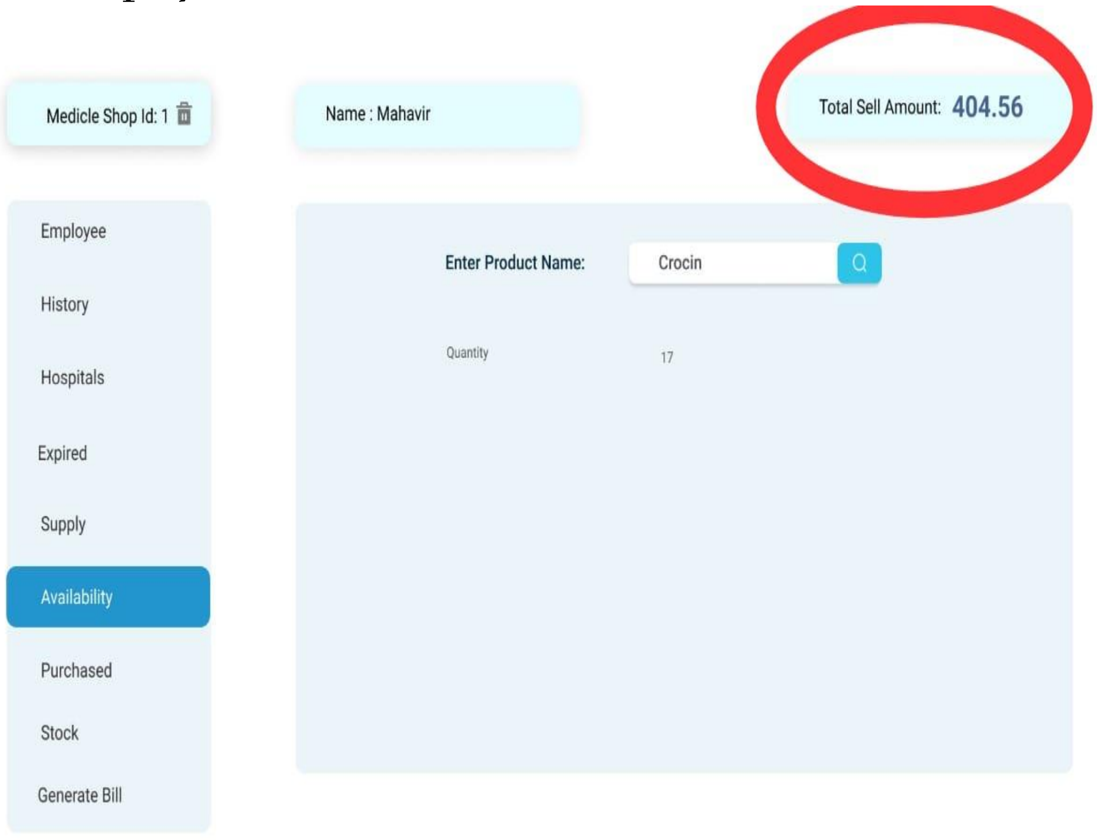
1) To check the availability of given product



```
create or replace procedure quantity (pro varchar2, medShop int) as
        cursor c_quantity is select sale.quantity as qty from sale where sale.med_shop_id =
medShop and sale.code in ( select equipment.code from equipment where
equipment.product_name = pro);

        r_quantity c_quantity%rowtype;
begin
        for r_quantity in c_quantity loop
                dbms_output.put_line ('Quantity present | '||r_quantity.qty);
        end loop;
end;

declare
begin
quantity('Crocin',1);
end;
```

/

2) To display total amount of sale.

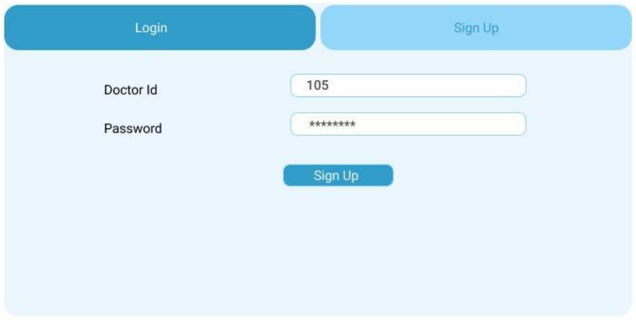


```
create or replace procedure sum_amount (medShopId int) as
        cursor c_sum_amount is select sum(amount) as sum from bill where
bill.med_shop_id = medShopId;
        r_sum_amount c_sum_amount%rowtype;

begin
        for r_sum_amount in c_sum_amount loop
                dbms_output.put_line('Sum of amount collected from selling :
'||r_sum_amount.sum);
        end loop;
end;

declare
begin
sum_amount(1);
end;
```

/

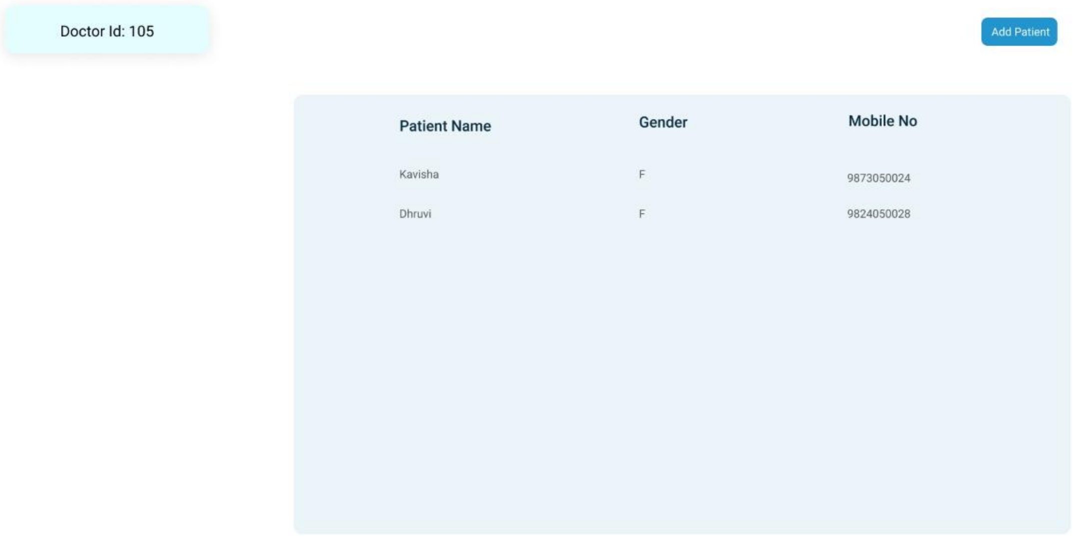
3) To display the patient details of a particular doctor

Login doctor:



Doctor Id	105
Password	********

Details

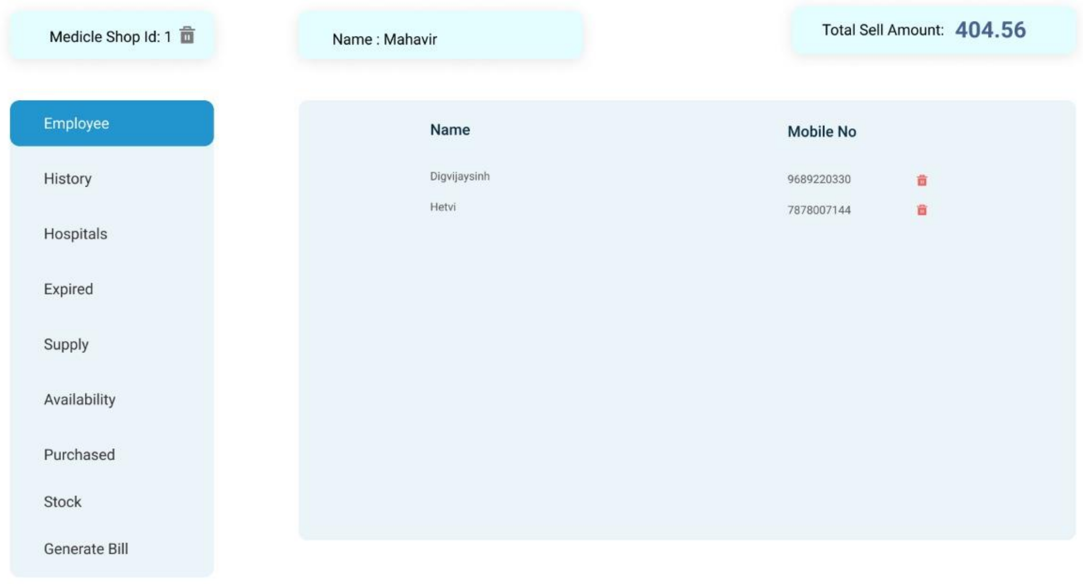


Doctor Id: 105

Patient Name	Gender	Mobile No
Kavisha	F	9873050024
Dhruvi	F	9824050028

```
 create or replace procedure patient_names (doctor_id int) as
        cursor c_patient_names is select customer.name, customer.gender,
customer.phone from customer where customer.pat_id in (select prescribe.pat_id from
prescribe where prescribe.doc_id = doctor_id);

        r_patient_names c_patient_names%rowtype;

begin
        for r_patient_names in c_patient_names loop
                dbms_output.put_line(r_patient_names.name||' |
'||r_patient_names.gender||' | '||r_patient_names.phone);
        end loop;
end;

declare
begin
patient_names(105);
end;
/
```

4)<u>To display the employee details of a particular medical shop</u>


Medicle Shop Id: 1

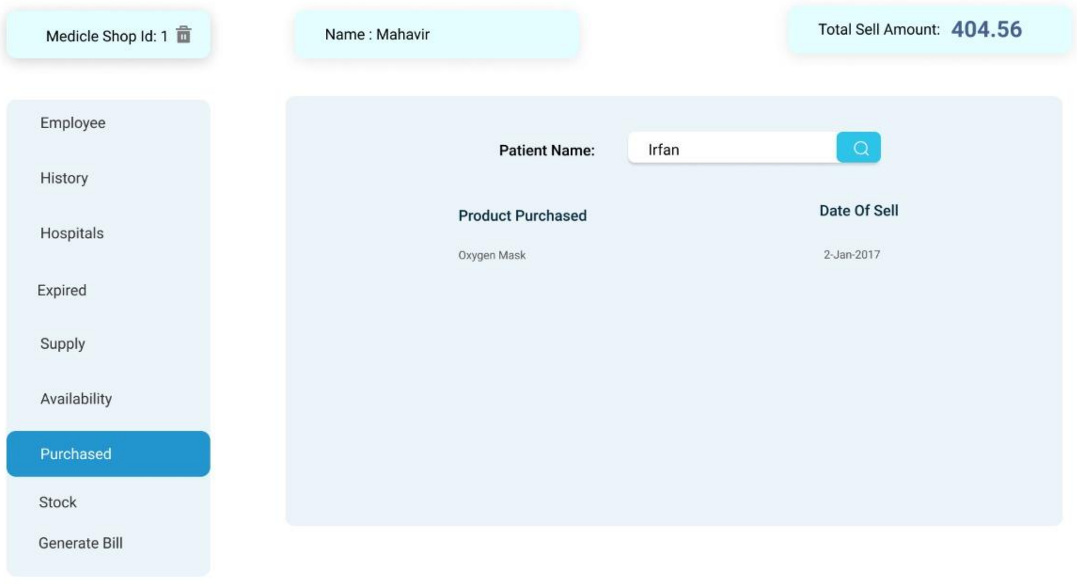
Name : Mahavir

Total Sell Amount: 404.56

Name	Mobile No
Digvijaysinh	9689220330
Hetvi	7878007144

```
create or replace procedure employee_details (medShopId int) as
        cursor c_employee_details is select employee.name,employee.mobile_no from
employee where employee.med_shop_id in (select med_shop.med_shop_id from
med_shop where med_shop.med_shop_id = medShopId);
        r_employee_details c_employee_details%rowtype;

begin
        for r_employee_details in c_employee_details loop
                dbms_output.put_line(r_employee_details.name||'
'||r_employee_details.mobile_no);
        end loop;
end;

declare
begin
employee_details(1);

end;
/
```

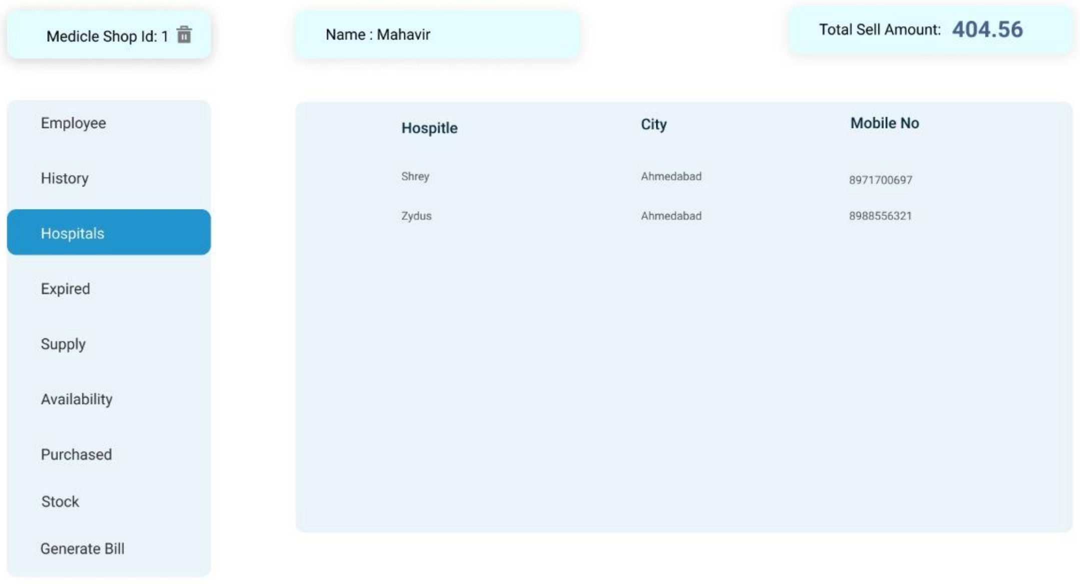
5) To display details of a customer



Product Purchased	Date Of Sell
Oxygen Mask	2-Jan-2017

```
create or replace procedure med_details (ptName varchar2, med int) as
        cursor c_med_details is select product,date_of_sale from bill where pat_name =
    ptName and med_shop_id = med;
        r_med_details c_med_details%rowtype;

    begin
        for r_med_details in c_med_details loop
                dbms_output.put_line(r_med_details.product||'
    '||r_med_details.date_of_sale);
        end loop;
    end;

    declare
    begin
    med_details('Irfan',1);
    end;
    /
```

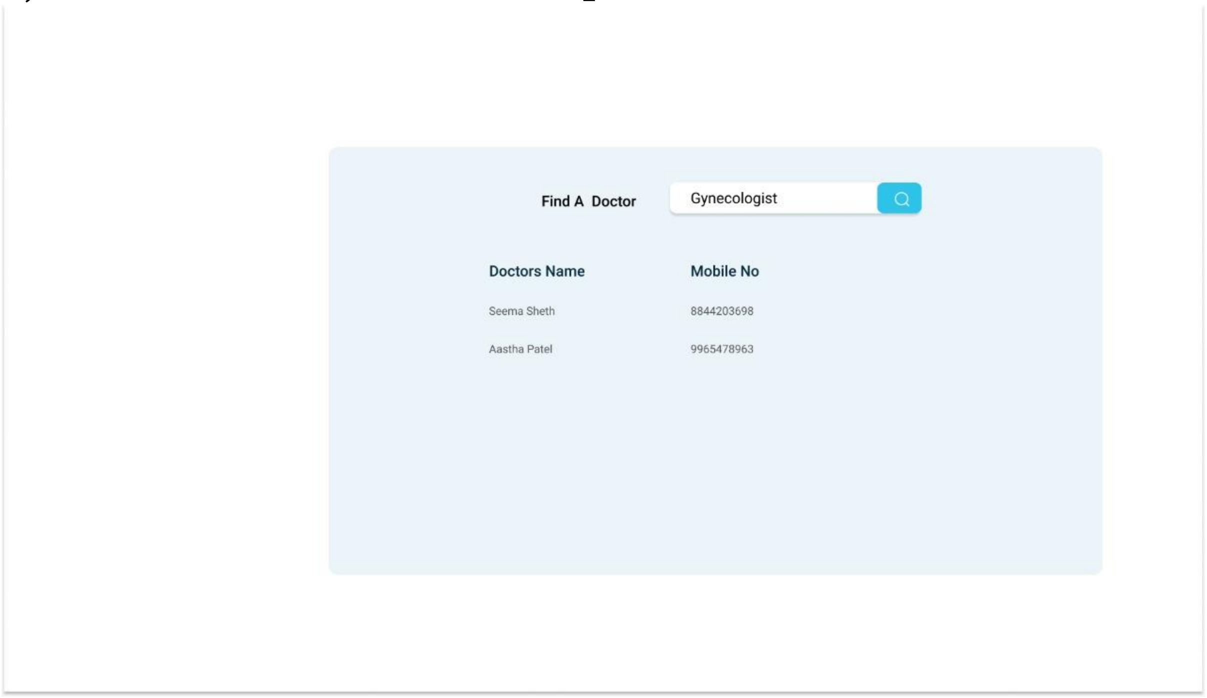
6) To display the details of hospitals which are being supplied medicines by a particular shop.



Hospitle	City	Mobile No
Shrey	Ahmedabad	8971700697
Zydus	Ahmedabad	8988556321

```
create or replace procedure hos_details (pharName varchar2) as
        cursor c_hos_details is select hospital.name,hospital.city,phone from hospital
where hospital.med_shop_id in (select med_shop.med_shop_id from med_shop where
name =pharName);
        r_hos_details c_hos_details%rowtype;

begin
        for r_hos_details in c_hos_details loop
                dbms_output.put_line(r_hos_details.name||' '||r_hos_details.city||' |
'||r_hos_details);
        end loop;
end;
/

declare
begin
hos_details('Mahavir');
end;
/
```

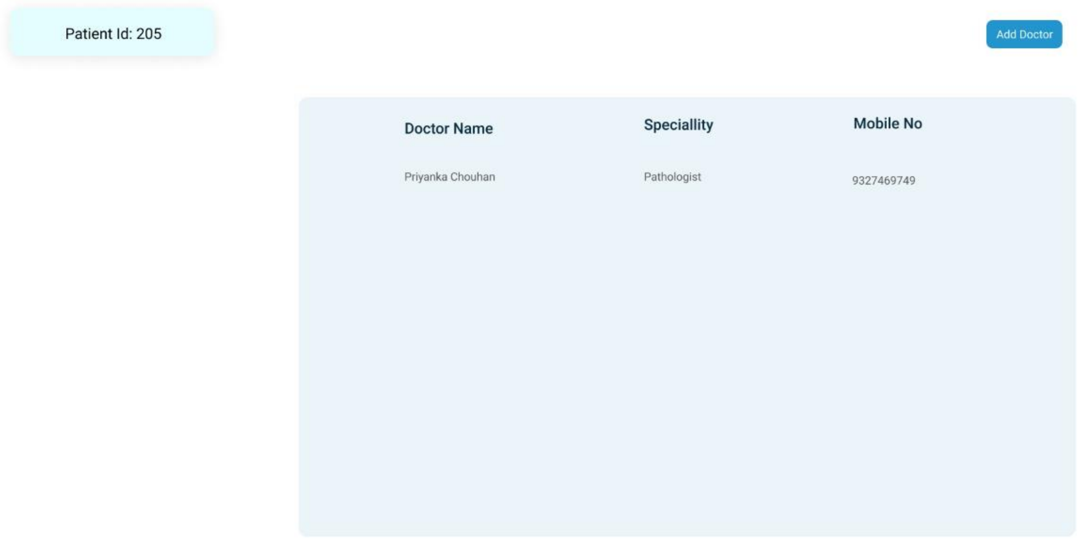
7) <u>To find the doctor based on specification</u>



Doctors Name	Mobile No
Seema Sheth	8844203698
Aastha Patel	9965478963

```
create or replace procedure spec (spec varchar2) as
        cursor c_spec is select doctor.doc_name, doctor.mobile from doctor where
speciality = spec;
        r_spec c_spec%rowtype;
begin
        for r_spec in c_spec loop
        dbms_output.put_line(r_spec.doc_name||' | '||r_spec.mobile);
        end loop;
end;
/

declare
begin
spec('Gynecologist');
end;
/
```

8) To display the doctors of particular patients.

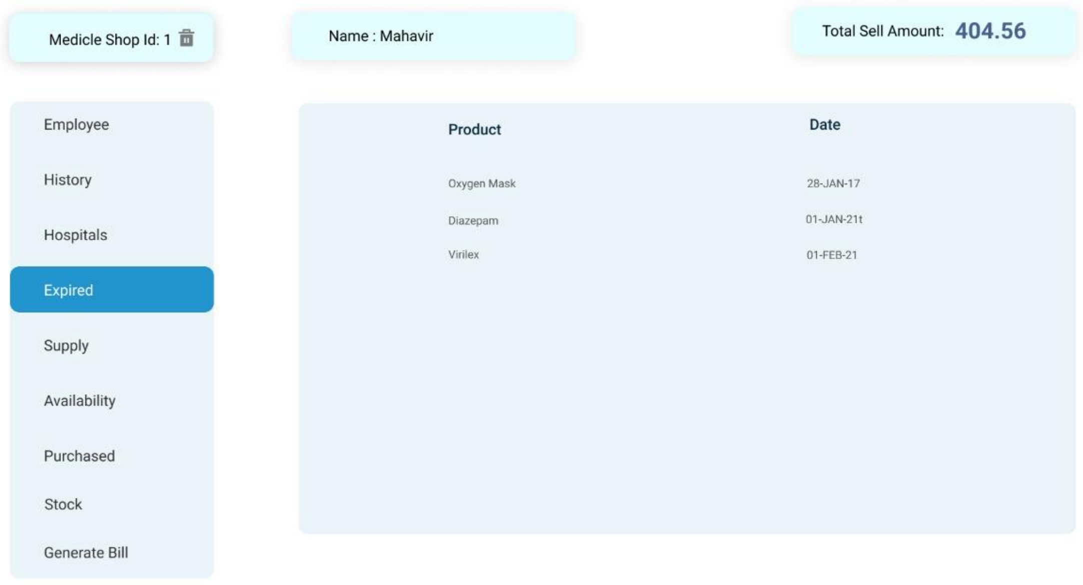


Patient Id: 205

Doctor Name	Speciallity	Mobile No
Priyanka Chouhan	Pathologist	9327469749

```
create or replace procedure doctor_names (patient_id int) as
	cursor c_doctor_names is select doctor.doc_name, SPECIALITY, doctor.mobile
from doctor where doctor.doc_id in (select prescribe.doc_id from prescribe where
prescribe.pat_id = patient_id);

	r_doctor_names c_doctor_names%rowtype;

begin
	for r_doctor_names in c_doctor_names loop
		dbms_output.put_line(r_doctor_names.doc_name||' |
'||r_doctor_names.SPECIALITY||' | '||r_doctor_names.mobile);
	end loop;
end;

declare
begin
doctor_names(205);
end;
/
```

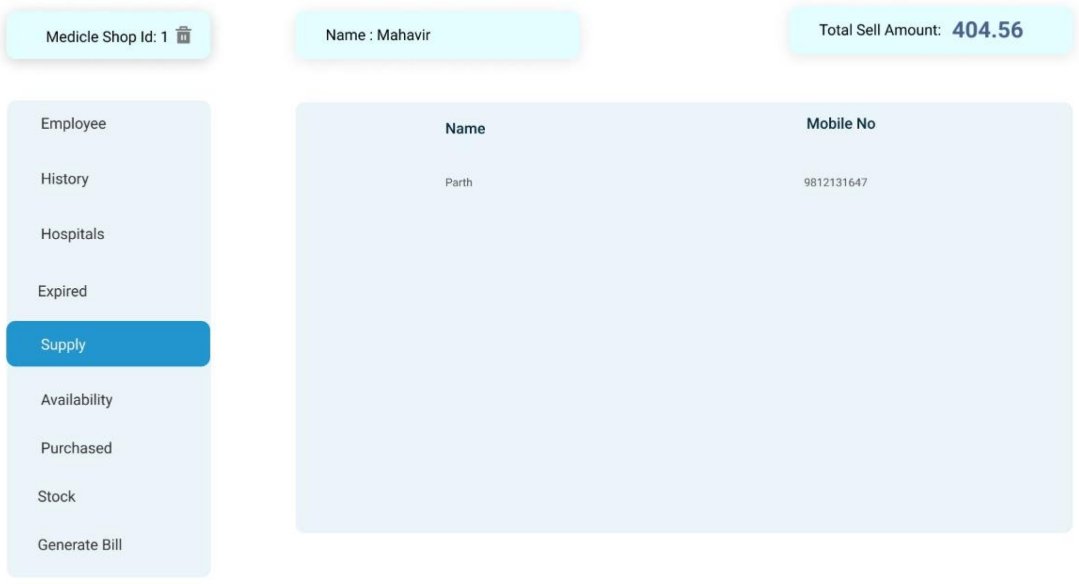
9) <u>To check the expired products of particular medical shops</u>.



Product	Date
Oxygen Mask	28-JAN-17
Diazepam	01-JAN-21t
Virilex	01-FEB-21

```
create or replace procedure expiry(phID int) as
        cursor c_exp is select product_name,exp_date from equipment where sysdate >
exp_date and equipment.code in (select sale.code from sale where med_shop_id = phID);
        r_exp c_exp%rowtype;
begin
        for r_exp in c_exp loop
        dbms_output.put_line(r_exp.product_name||' | '||r_exp.exp_date);
        end loop;
end;
/

declare
begin
expiry(1);
end;
/
```

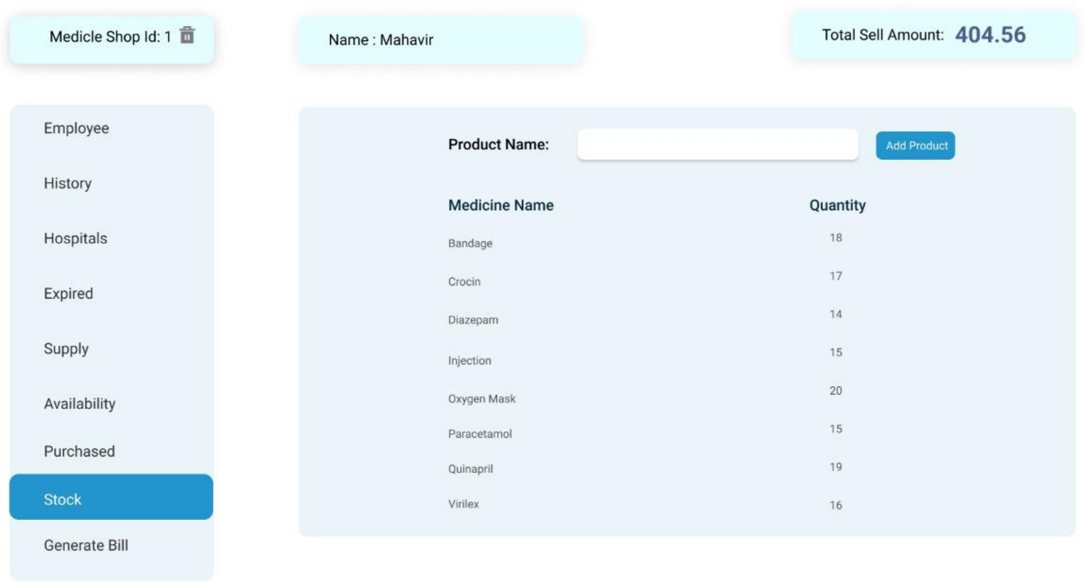
10) <u>To display supplier details that supplies medicines to a particular medical shop</u>.



Name	Mobile No
Parth	9812131647

```
create or replace procedure sup (phiD int) as
        cursor c_sup is select supplier.name,supplier.mobile_no from supplier where
supplier.med_shop_id = phiD;
        r_sup c_sup%rowtype;
begin
        for r_sup in c_sup loop
        dbms_output.put_line(r_sup.name||' | '||r_sup.mobile_no);
        end loop;
end;

declare
begin
sup(1);
end;
/
```

11) <u>To check the stock available in a medical shop</u>



Name : Mahavir

Total Sell Amount: 404.56

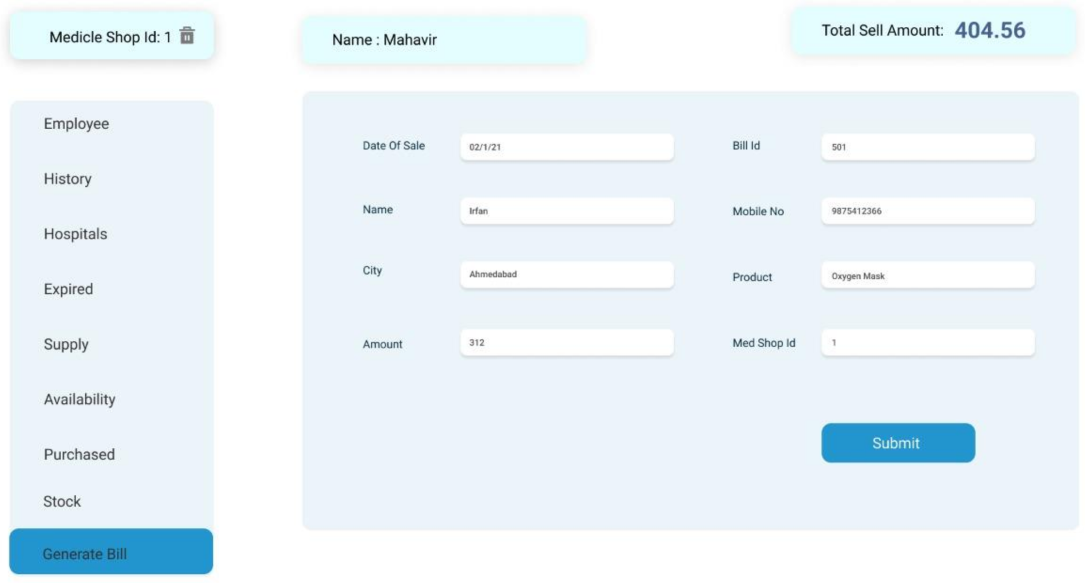
Medicine Name	Quantity
Bandage	18
Crocin	17
Diazepam	14
Injection	15
Oxygen Mask	20
Paracetamol	15
Quinapril	19
Virilex	16

```
create or replace procedure stock(phID int) as
        cursor c_stock is select product_name from equipment where equipment.code in (
select sale.code from sale where sale.med_shop_id = phid) order by product_name;
        cursor c_qty (nm equipment.product_name%type) is select quantity from sale
where sale.code in (select equipment.code from equipment where product_name = nm)
and sale.med_shop_id = phID;
```

```
        r_stock c_stock%rowtype;
        r_qty c_qty%rowtype;
begin
        for r_stock in c_stock loop
                for r_qty in c_qty(r_stock.product_name) loop
                        dbms_output.put_line(r_stock.product_name||' | ' ||r_qty.quantity);
                end loop;
        end loop;
end;
/

declare
begin
stock(1);
end;
/
```

Triggers

1) To give 10% discount on final amount which will store that in another table called discount table.



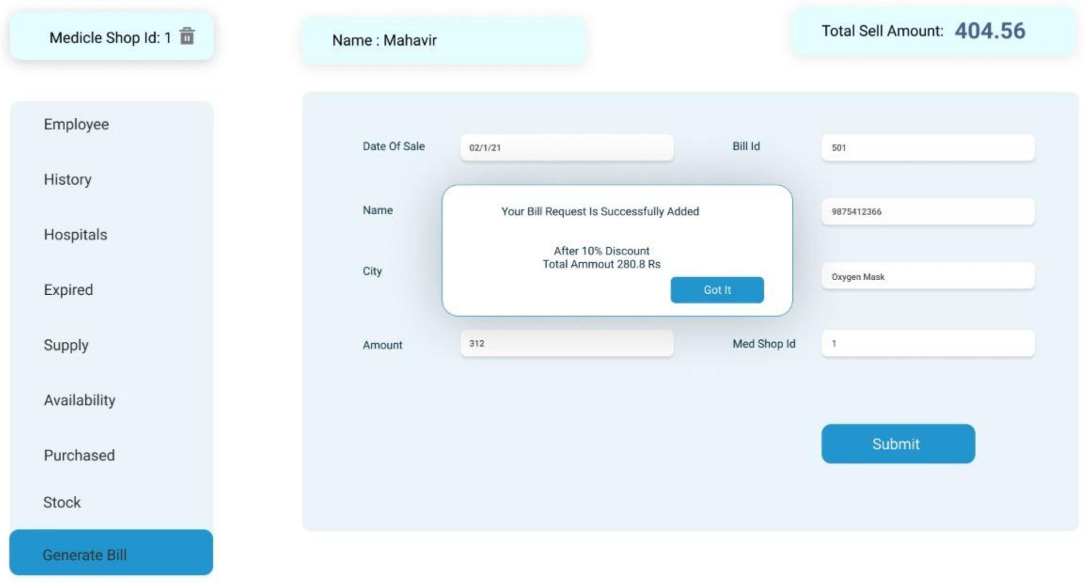
Medicle Shop Id: 1

Name : Mahavir

Total Sell Amount: 404.56

Date Of Sale	02/1/21	Bill Id	501
Name	Irfan	Mobile No	9875412366
City	Ahmedabad	Product	Oxygen Mask
Amount	312	Med Shop Id	1

After



Medicle Shop Id: 1

Name : Mahavir

Total Sell Amount: 404.56

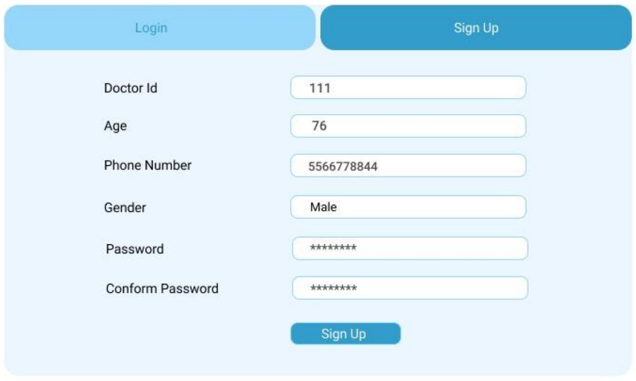
Date Of Sale	02/1/21	Bill Id	501
Name			9875412366
City			Oxygen Mask
Amount	312	Med Shop Id	1

Your Bill Request Is Successfully Added

After 10% Discount
Total Ammout 280.8 Rs

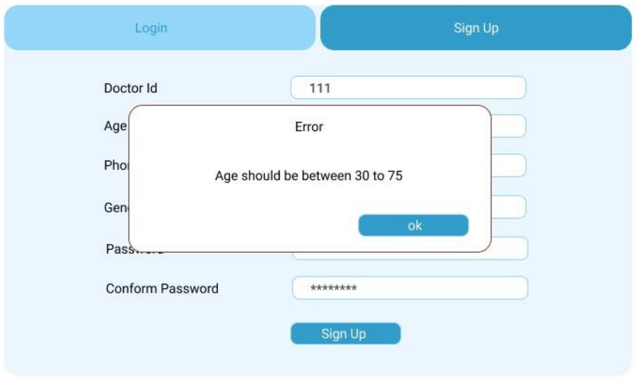
```
create table with_discount (
 bill_id int,
 original_price int,
 price_after_discount int,
 constraint pk_with_discount primary key(bill_id)
);



create or replace trigger discount after insert on bill
for each row
declare
 discounted_price int := (:new.amount*90)/100;
begin
	insert into with_discount values(:new.bill_id,:new.amount,discounted_price);
end;
/
```

2) To check the valid age of a doctor before signing up into the app.



Doctor Id	111
Age	76
Phone Number	5566778844
Gender	Male
Password	********
Conform Password	********

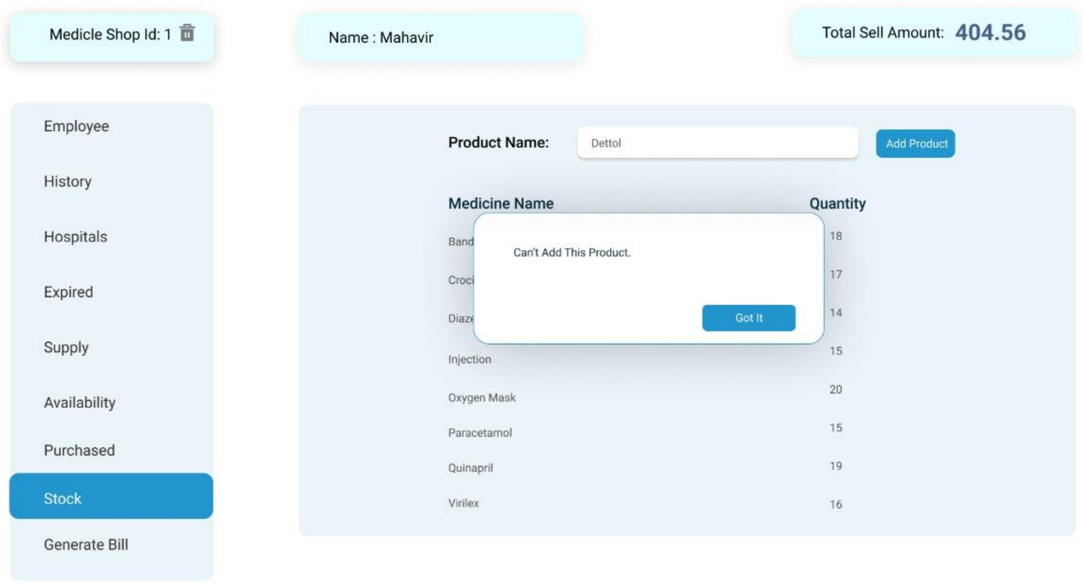
After



Age should be between 30 to 75

```
create or replace trigger age_chk
before insert or update on doctor
for each row
begin
        if(:new.age<30 and :new.age>75) then
                raise_application_error(-0021,'Age of doctor must be between 30 and 75');
        end if;
end;
/
```

3) To check whether the product is valid or not for the medical shop.



Can't Add This Product.

Medicine Name	Quantity
Band	18
Croci	17
Diaze	14
Injection	15
Oxygen Mask	20
Paracetamol	15
Quinapril	19
Virilex	16

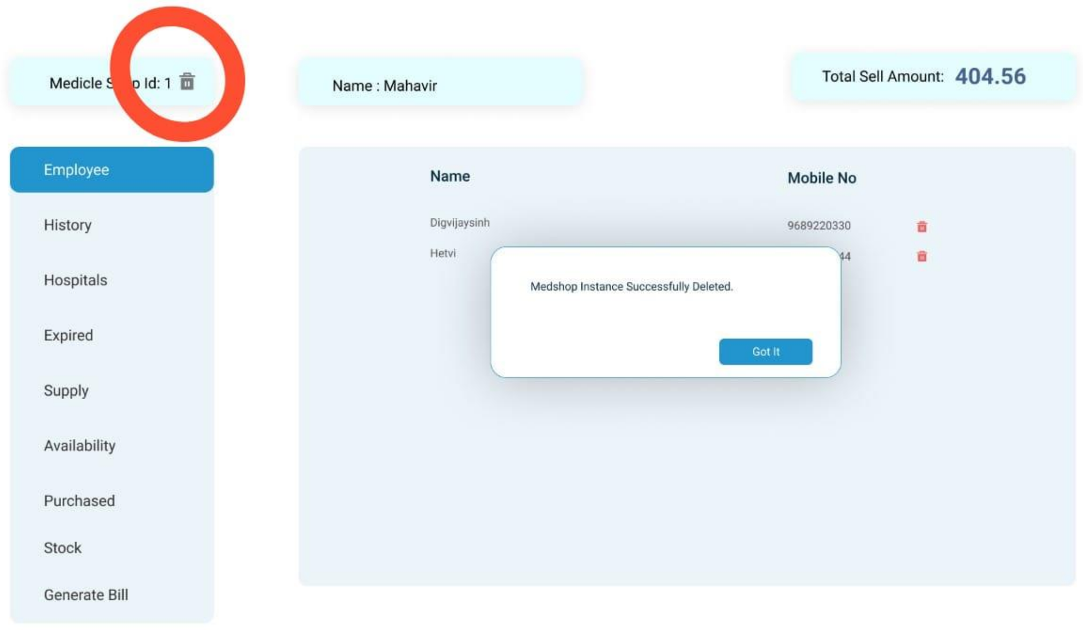
```
create or replace trigger inquiry before insert on sale
for each row
declare
 cursor c_inquiry(cid sale.med_shop_id%type) is select invoice.code as cd from invoice
where invoice.med_shop_id = cid;
 r_inquiry c_inquiry%rowtype;
 chk int;
begin
            if(:new.med_shop_id = 1) then
             for r_inquiry in c_inquiry(:new.med_shop_id) loop
                        if(:new.code = r_inquiry.cd) then
                                    chk := 1;
                        end if;
             end loop;
             if (chk = 0) then
              raise_application_error(-9001,'Cannnot sell those item which are not imported');
             end if;
            end if;
```

```
if(:new.med_shop_id = 2) then
 for r_inquiry in c_inquiry(:new.med_shop_id) loop
            if(:new.code = r_inquiry.cd) then
                        chk := 1;
            end if;
 end loop;
 if (chk = 0) then
  raise_application_error(-9001,'Cannnot sell those item which are not imported');
 end if;
end if;

if(:new.med_shop_id = 3) then
 for r_inquiry in c_inquiry(:new.med_shop_id) loop
            if(:new.code = r_inquiry.cd) then
                        chk := 1;
            end if;
 end loop;
 if (chk = 0) then
  raise_application_error(-9001,'Cannnot sell those item which are not imported');
 end if;
end if;

if(:new.med_shop_id = 3) then
 for r_inquiry in c_inquiry(:new.med_shop_id) loop
            if(:new.code = r_inquiry.cd) then
                        chk := 1;
            end if;
 end loop;
 if (chk = 0) then
  raise_application_error(-9001,'Cannnot sell those item which are not imported');
 end if;
end if;

if(:new.med_shop_id = 4) then
 for r_inquiry in c_inquiry(:new.med_shop_id) loop
            if(:new.code = r_inquiry.cd) then
                        chk := 1;
            end if;
 end loop;
 if (chk = 0) then
```

```
  raise_application_error(-9001,'Cannnot sell those item which are not imported');
 end if;
end if;

if(:new.med_shop_id = 5) then
 for r_inquiry in c_inquiry(:new.med_shop_id) loop
            if(:new.code = r_inquiry.cd) then
                        chk := 1;
            end if;
 end loop;
 if (chk = 0) then
  raise_application_error(-9001,'Cannnot sell those item which are not imported');
 end if;
end if;

if(:new.med_shop_id = 6) then
 for r_inquiry in c_inquiry(:new.med_shop_id) loop
            if(:new.code = r_inquiry.cd) then
                        chk := 1;
            end if;
 end loop;
 if (chk = 0) then
  raise_application_error(-9001,'Cannnot sell those item which are not imported');
 end if;
end if;

if(:new.med_shop_id = 7) then
 for r_inquiry in c_inquiry(:new.med_shop_id) loop
            if(:new.code = r_inquiry.cd) then
                        chk := 1;
            end if;
 end loop;
 if (chk = 0) then
  raise_application_error(-9001,'Cannnot sell those item which are not imported');
 end if;
end if;

if(:new.med_shop_id = 8) then
 for r_inquiry in c_inquiry(:new.med_shop_id) loop
            if(:new.code = r_inquiry.cd) then
```

```
                        chk := 1;
                end if;
         end loop;
         if (chk = 0) then
          raise_application_error(-9001,'Cannnot sell those item which are not imported');
         end if;
        end if;

        if(:new.med_shop_id = 9) then
         for r_inquiry in c_inquiry(:new.med_shop_id) loop
                if(:new.code = r_inquiry.cd) then
                        chk := 1;
                end if;
         end loop;
         if (chk = 0) then
          raise_application_error(-9001,'Cannnot sell those item which are not imported');
         end if;
        end if;
end;
/
```

4) To delete every detail of medical shop if the owner wants to leave


Medicle S p Id: 1

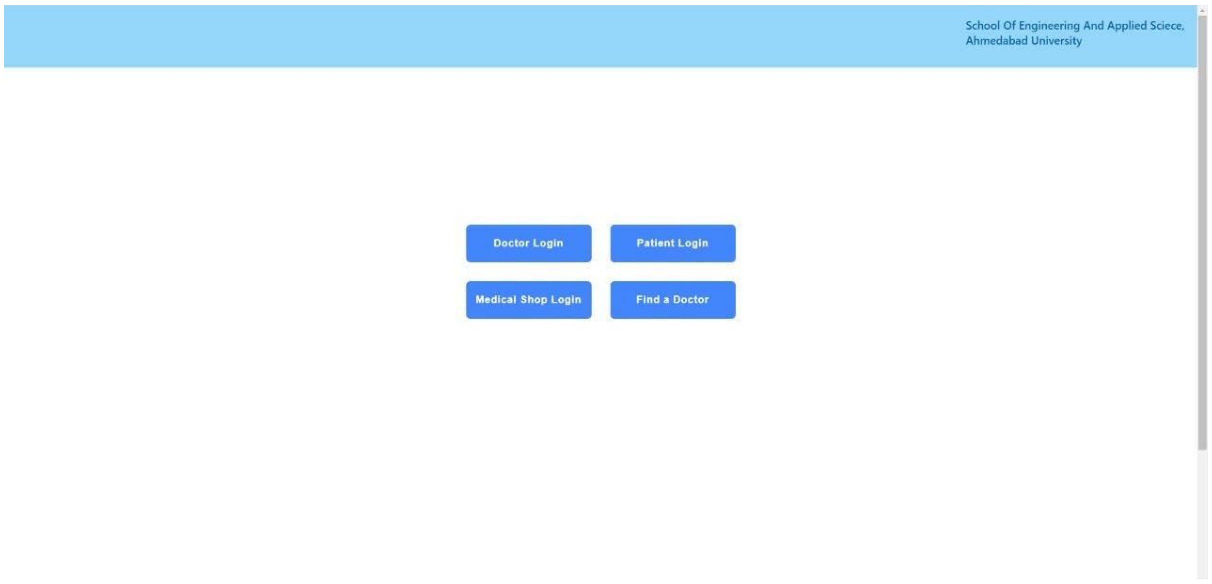
Name : Mahavir

Total Sell Amount: 404.56

Name	Mobile No
Digvijaysinh	9689220330
Hetvi	44

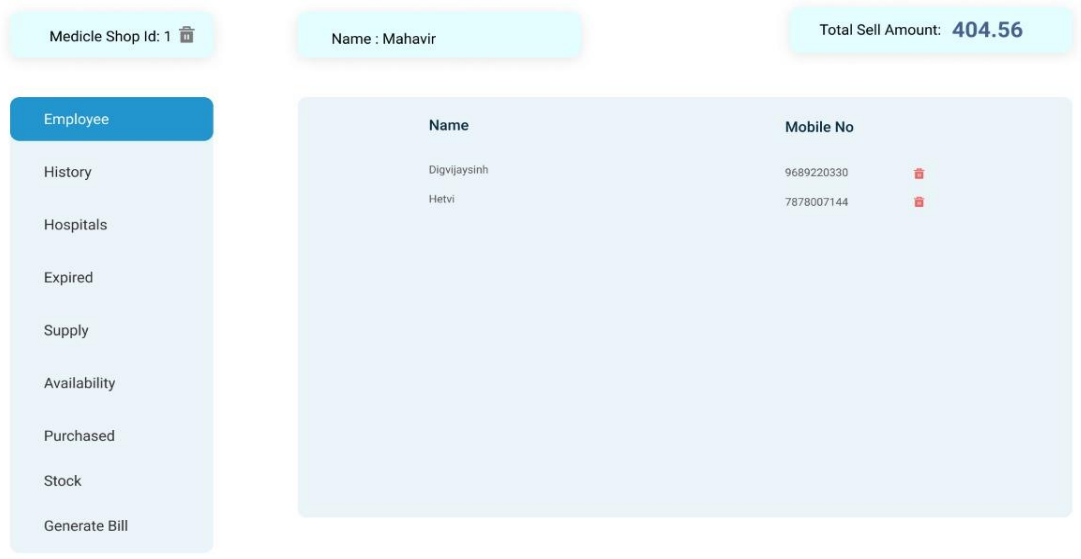
Medshop Instance Successfully Deleted.

After



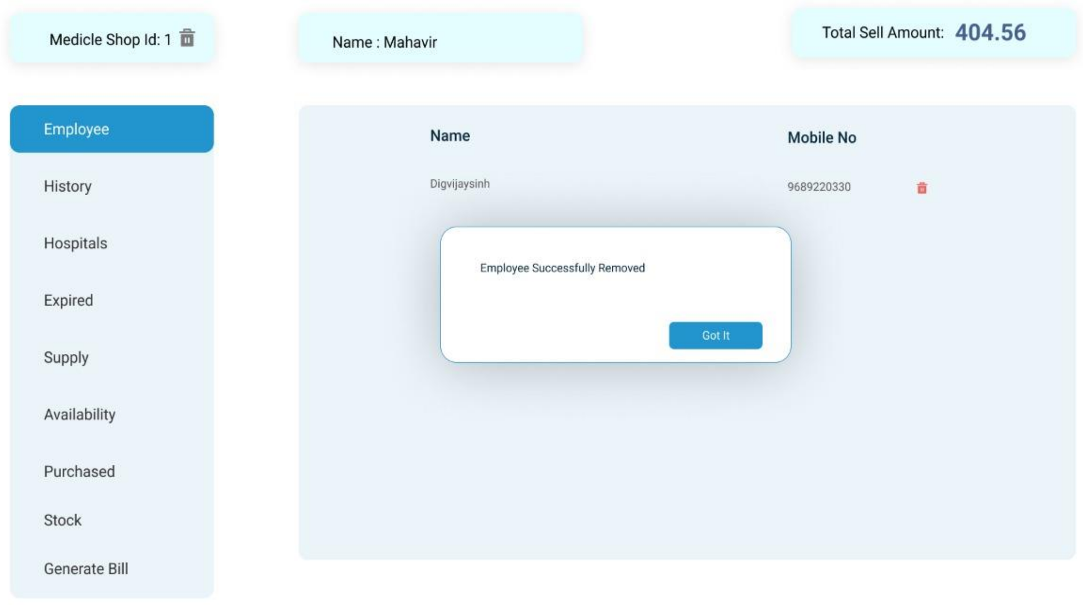
```
create or replace trigger med_shop_del before delete on med_shop
for each row
begin
        delete from employee where :old.med_shop_id = employee.med_shop_id;
        delete from supplier where :old.med_shop_id = supplier.med_shop_id;
        delete from sale where :old.med_shop_id = sale.med_shop_id;
        delete from invoice where :old.med_shop_id = invoice.med_shop_id;
        update hospital set hospital.med_shop_id = null where hospital.med_shop_id =
:old.med_shop_id;
end;
/
```

5) To delete employee records if he/she leaves the shop.



Name	Mobile No
Digvijaysinh	9689220330
Hetvi	7878007144

After



Name	Mobile No
Digvijaysinh	9689220330

Employee Successfully Removed

```
create or replace trigger tr_emp_shift before delete on employee
for each row
begin
delete from works where mobile_no=:old.mobile_no;
end;
/
```

THANK YOU