BLOOD BANK MANAGEMENT SYSTEM

Prepared by:

Munir Abbasi 20K-0244

Raza Ali 20K-1061

Mustafa Zahid 20K-1045

FAST-NUCES

Submitted to : Miss Erum Shaheen

Table of Contents

1.0. Introduction 2
2.0. Technology Used.....4
3.0. System Design.....4
4.0. Normalization.....7
5.0. Database Concepts 8
5.1. Triggers 8
5.2. Procedures 12
5.3. Functions 17
5.4. Views.....19
5.5. Queries.....23
6.0. Connectivity Screenshots.....24
7.0. Project Screenshots 26

1.0. Introduction

The objective of this project is to create an online blood donation information system. The web app is made to manage the blood bank's daily operations and perform information searches as needed. There are several situations where the demand for blood is urgent. The Online Blood Bank initiative attempts to keep all the data on blood donors at this crucial moment. The blood donation agent is responsible for compiling e-Information about the donor and any organizations connected to blood donation. Anyone interested in donating blood can register using this application.

The blood bank management system is going to be an application on java spring boot. The purpose of this project was to develop a blood management information system to assist in the management of blood donor records and sells records. The Blood Bank Management System is a software that collects, organizes, retrieves, and examines information related to managing blood banks. Additionally, it is in charge of managing the blood bank's inventory.

The goal of the blood bank management project is to offer blood donation services to the community easily. Our Blood Bank Management System is a browser-based system that is meant to store, process, retrieve and analyze the information connected with the administrative and inventory management inside a blood bank.

This project aims at keeping all the information regarding blood donors, and various blood groups accessible in each blood bank and enable them to manage in a better way. The goal is to increase openness in this industry, make blood donation hassle- and corruption-free, and improve the efficiency of the blood bank management system. Some current systems are manual and it is very time-consuming. This system will help the blood bank to manage information on available blood packets of the corresponding blood group.

1.1 Core functions

Customers and administrators will have access to features, which are described below

1.1.1 Manage Recipients

Access to donor management is available to the admin. Admin can modify the blood record, search for donor or acceptor information, and examine the personal information.

1.1.2 Manage Donator

The administrator has access to the blood donor. He can update the donor's information.

1.1.3 Donor Registration

Customers will fill out the documents for the donor registration such as their full name, gender, date of birth, blood type, contact information (phone, email, home address, etc.), and so on.

1.1.4 Contact Information.

Users can view the blood bank address, phone number, and email address under "Contact Us." They can fill up a form.

1.1.5 Blood Bank FAQs

Under "About Us," you may view the blood bank management system's frequently asked questions.

1.1.6 Manage Blood packets

Manage and list no. of blood packets available.

1.1.7 Manage accepter information

This functionality enables administrator to view the accepter details.

1.1.8 Manage blood transfer

Manage the transfer of blood from donor to stock.

1.1.9 Generate donation

Produce a report for the blood donation by donors.

1.1.10 Donor eligibility

List of a donor who is eligible for donation on a particular date with a contact number.

1.1.11 Record retrievals

Retrieve data of blood packet accepters.

1.1.12 Donation by users

Provide a financial donation option to the blood bank.

1.1.13 Add/Delete/Update donor, hospital,bloodbank

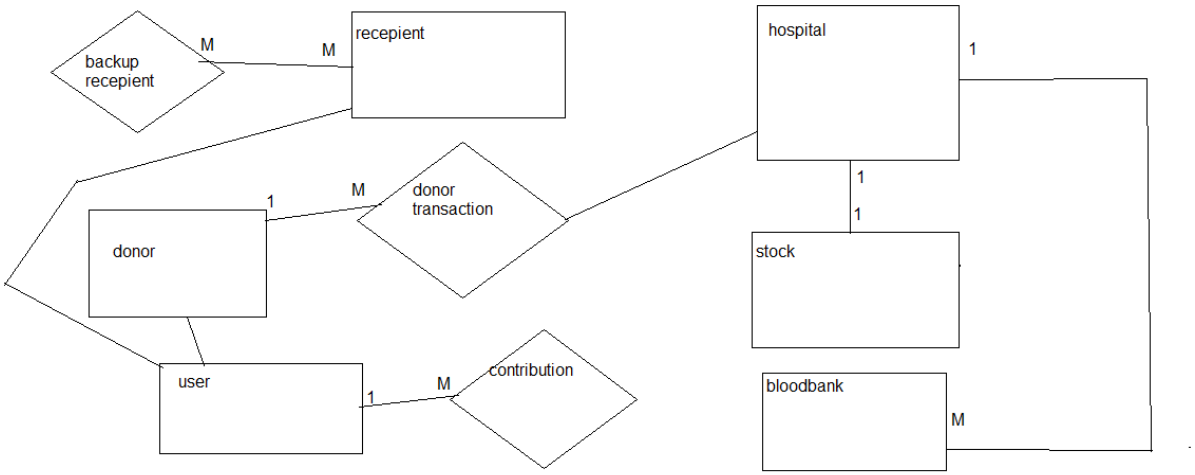
Add, delete, update donors, bloodbank and hostpial data.

2.0. Technology Used

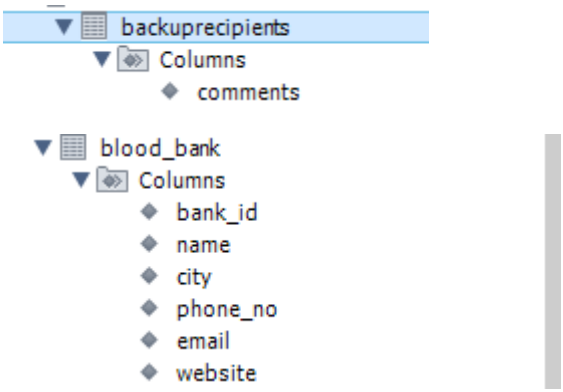
The technologies used are listed below:

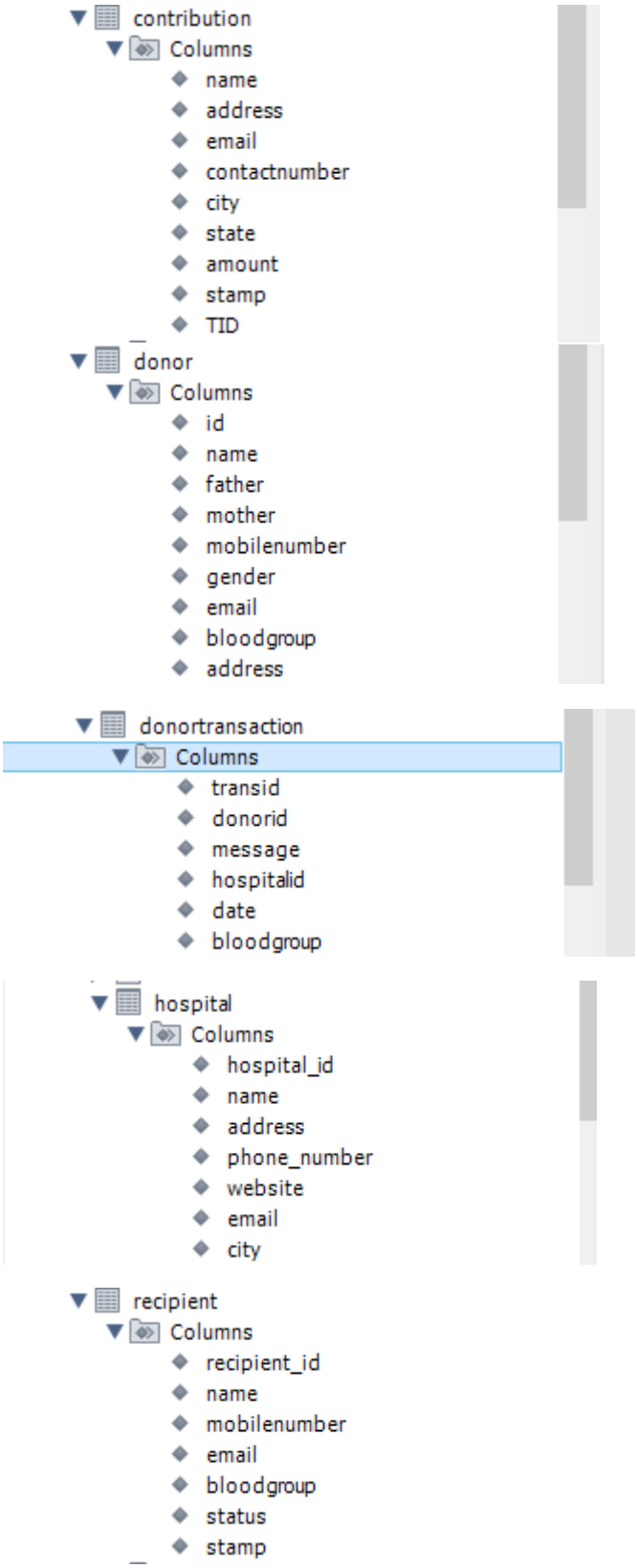
- Apache Tomcat version 9.0.68.
- MySQL Workbench 8.0.
- MySQL Shell.
- MySQL Command line client.
- Eclipse IDE for developers, 2022-06 version.
- Layershift Hosting.

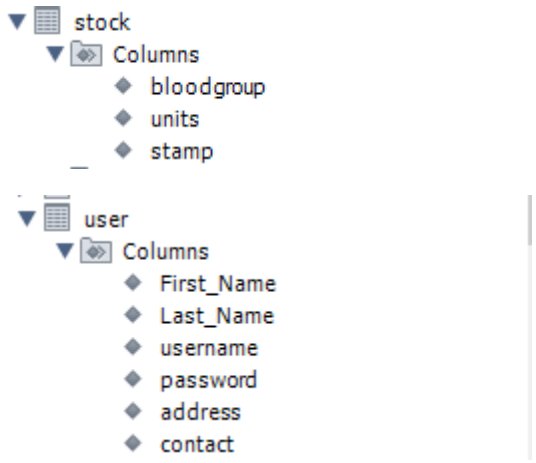
3.0. System Design



Since the attributes were too many to draw. The Entities and Relationships have been defined with their respective tables and functions.




contribution
Columns
name
address
email
contactnumber
city
state
amount
stamp
TID
donor
Columns
id
name
father
mother
mobilenumber
gender
email
bloodgroup
address
donortransaction
Columns
transid
donorid
message
hospitalid
date
bloodgroup
hospital
Columns
hospital_id
name
address
phone_number
website
email
city
recipient
Columns
recipient_id
name
mobilenumber
email
bloodgroup
status
stamp



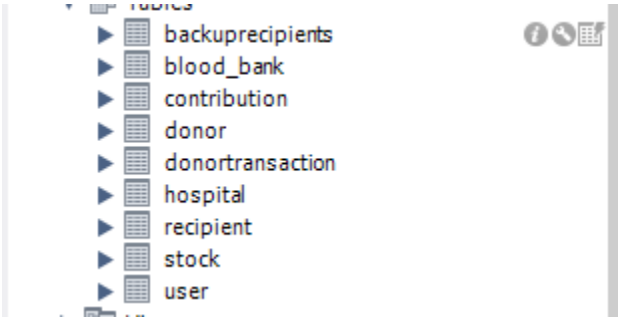
4.0. Normalization

The tables have been normalized. Normalization was achieved using the 1NF, 2NF, and the 3NF.

1NF=>We have atomic data.

2NF=> We have full functional dependency.

3NF=> We have achieved this by removing transitive dependency.



5.0. Database Concepts

5.1. Triggers

```
#--TRIGGERS--

delimiter //

CREATE TRIGGER chk_weight BEFORE INSERT ON donor

    FOR EACH ROW

    BEGIN

      IF NEW.weight < 50 THEN

                    SIGNAL SQLSTATE '50001' SET MESSAGE_TEXT =
'Person must weight above 50kg.';

       END IF;

    END;//

 delimiter ;



#Trigger to stop inputting negative numbers in contact number of user

delimiter //

 CREATE TRIGGER verify_number_user BEFORE INSERT ON user

       FOR EACH ROW

       BEGIN

            IF NEW.contact < 0 THEN

     SIGNAL SQLSTATE '50002' SET MESSAGE_TEXT='Contact number is
never negative';

     END IF;

       END;
```

#Trigger to stop inputting negative numbers in contact number of donor

```
delimiter //
 CREATE TRIGGER verify_number_donor BEFORE INSERT ON donor
        FOR EACH ROW
        BEGIN
              IF NEW.mobilenumber < 0 THEN
      SIGNAL SQLSTATE '50002' SET MESSAGE_TEXT='Contact number is never negative';
      END IF;
        END;
```

#Trigger to stop inputting negative numbers in contact number of donor Before update

```
delimiter //
 CREATE TRIGGER verify_number_donor_onUpdate BEFORE UPDATE ON donor
        FOR EACH ROW
        BEGIN
              IF NEW.mobilenumber < 0 THEN
      SIGNAL SQLSTATE '50002' SET MESSAGE_TEXT='Contact number is never negative';
      END IF;
        END;
```

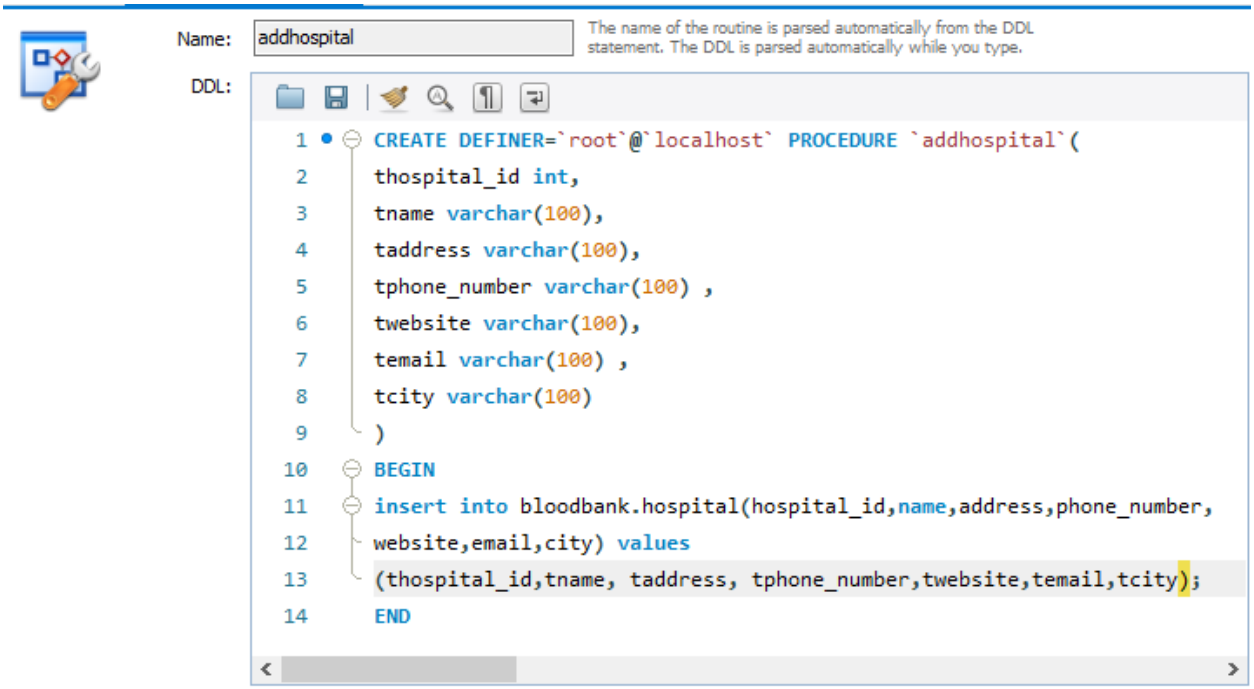
#Trigger to add 10 units stock table after a user donates blood

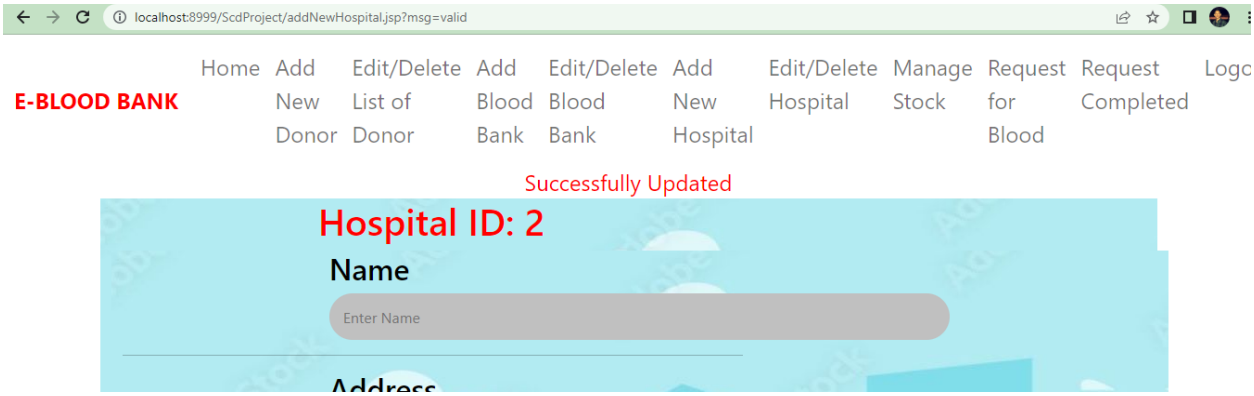
```
DELIMITER //
CREATE TRIGGER stocktriggeradd AFTER INSERT ON donortransaction
FOR EACH ROW
begin

DECLARE bloodgroup varchar(20);

SELECT   bg INTO bloodgroup
FROM      donortransaction
ORDER BY  transid DESC
LIMIT     1;

UPDATE stock
SET units=units+10
WHERE bg=bloodgroup;
END//

#Trigger to subtract 10 units of blood after a recipients blood request is accepted
DELIMITER //
CREATE TRIGGER stocktrigger AFTER UPDATE ON recipient
FOR EACH ROW
begin

DECLARE bloodgroup varchar(20);
```

```
SELECT   bg INTO bloodgroup
FROM      recipient
ORDER BY  recipient_id DESC
LIMIT     1;

UPDATE stock
SET units=units-10
WHERE bg=bloodgroup;
END//



#Trigger to Archive deleted bloodrequests
delimiter //
CREATE TRIGGER backingRecipients AFTER DELETE
        ON recipient FOR EACH ROW
   BEGIN
        INSERT INTO backuprecipients VALUES (concat('Recipient Name
Deleted:',old.name,',RecipientID Deleted:',old.recipient_id));
   END //


   DROP TRIGGER backingRecipients;
```

5.2. Procedures

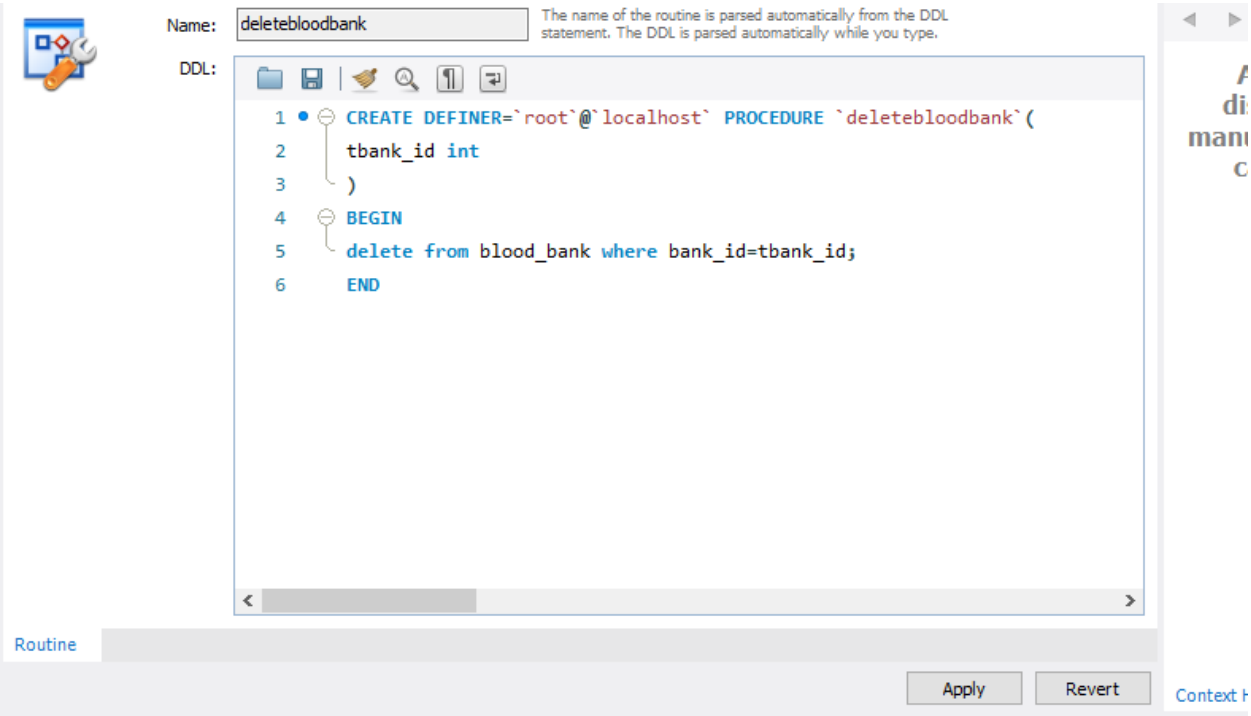
Add hospital





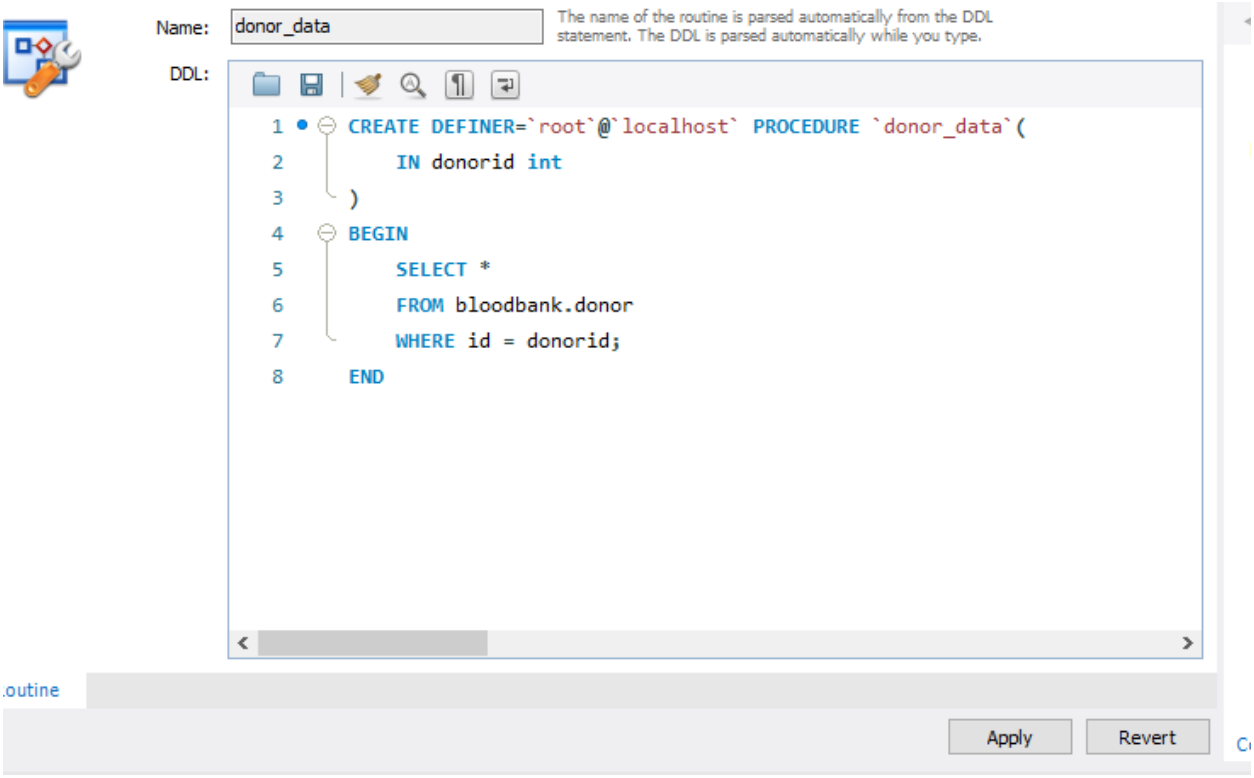
```jsp
<%@page import="Project.ConnectionProvider"%>
<%@page import="java.sql.*"%>
<%
String id = request.getParameter("id");
String name = request.getParameter("name");
String address = request.getParameter("address");
String phone_number = request.getParameter("phone_number");
String website = request.getParameter("website");
String email = request.getParameter("email");
String city = request.getParameter("city");

try{
    Connection con=ConnectionProvider.getCon();
    //PreparedStatement ps = con.prepareStatement("insert into hospital values(?,?,?,?,?,?,?)");
    PreparedStatement ps = con.prepareStatement("call addhospital(?,?,?,?,?,?,?)");
    ps.setString(1, id);
    ps.setString(2, name);
    ps.setString(3, address);
    ps.setString(4, phone_number);
    ps.setString(5, website);
    ps.setString(6, email);
    ps.setString(7, city);
    ps.executeUpdate();
    response.sendRedirect("addNewHospital.jsp?msg=valid");
```

Deletebloodbank

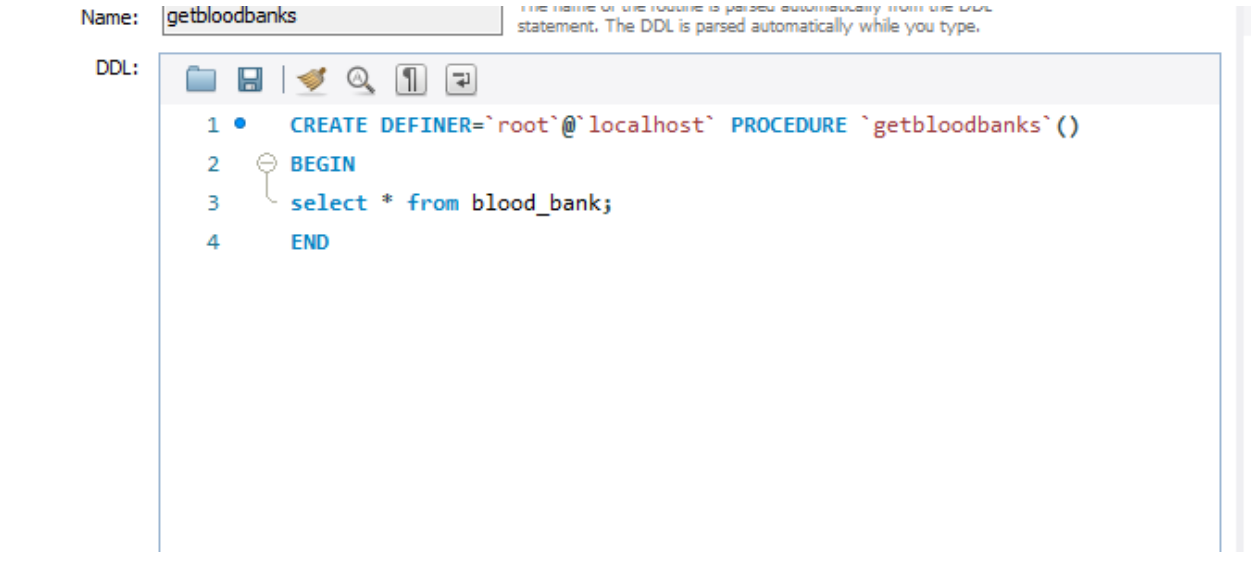


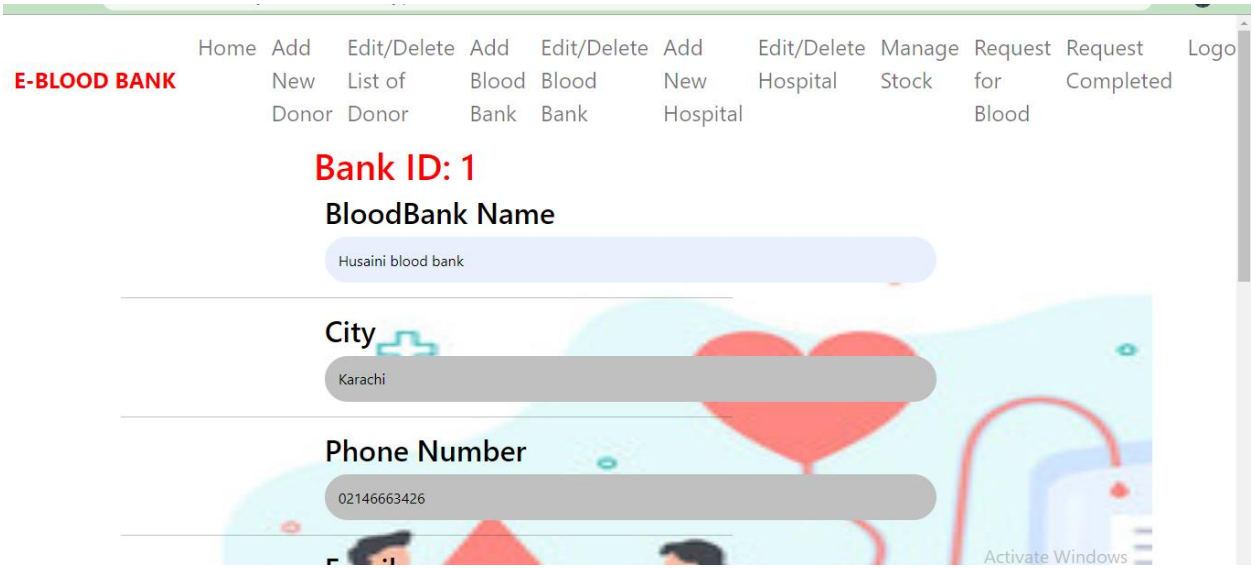

```jsp
String id=request.getParameter("id");
try{
    Connection con=ConnectionProvider.getCon();
    Statement st = con.createStatement();
    //st.executeUpdate("delete from blood_bank where bank_id='"+id+"'");
    st.executeUpdate("call deletebloodbank("+id+")");
```

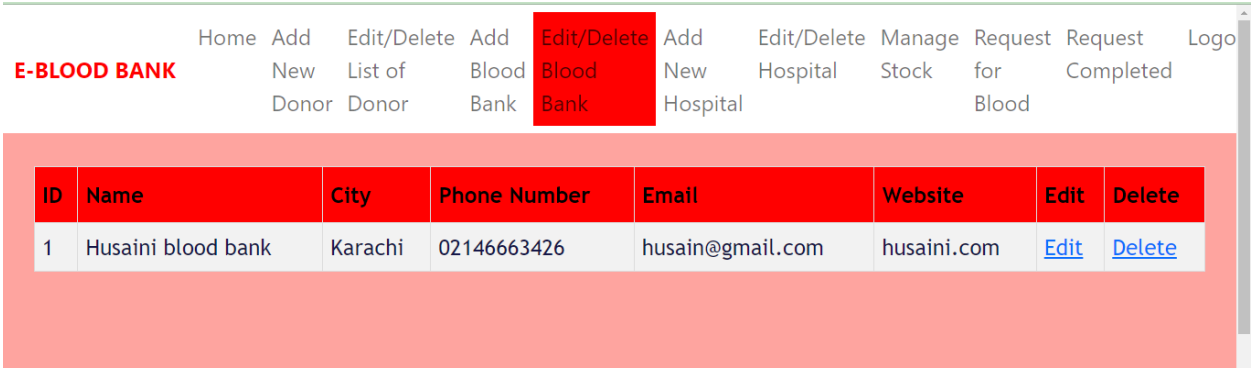
Donor_data



Getbloodbanks




E-BLOOD BANK
Home
Add New Donor
Edit/Delete List of Donor
Add Blood Bank
Edit/Delete Blood Bank
Add New Hospital
Edit/Delete Hospital
Manage Stock
Request for Blood
Request Completed
Logo
Bank ID: 1
BloodBank Name
Husaini blood bank
City
Karachi
Phone Number
02146663426
Activate Windows


E-BLOOD BANK
Home
Add New Donor
Edit/Delete List of Donor
Add Blood Bank
Edit/Delete Blood Bank
Add New Hospital
Edit/Delete Hospital
Manage Stock
Request for Blood
Request Completed
Logo
ID
Name
City
Phone Number
Email
Website
Edit
Delete
1
Husaini blood bank
Karachi
02146663426
husain@gmail.com
husaini.com
Edit
Delete

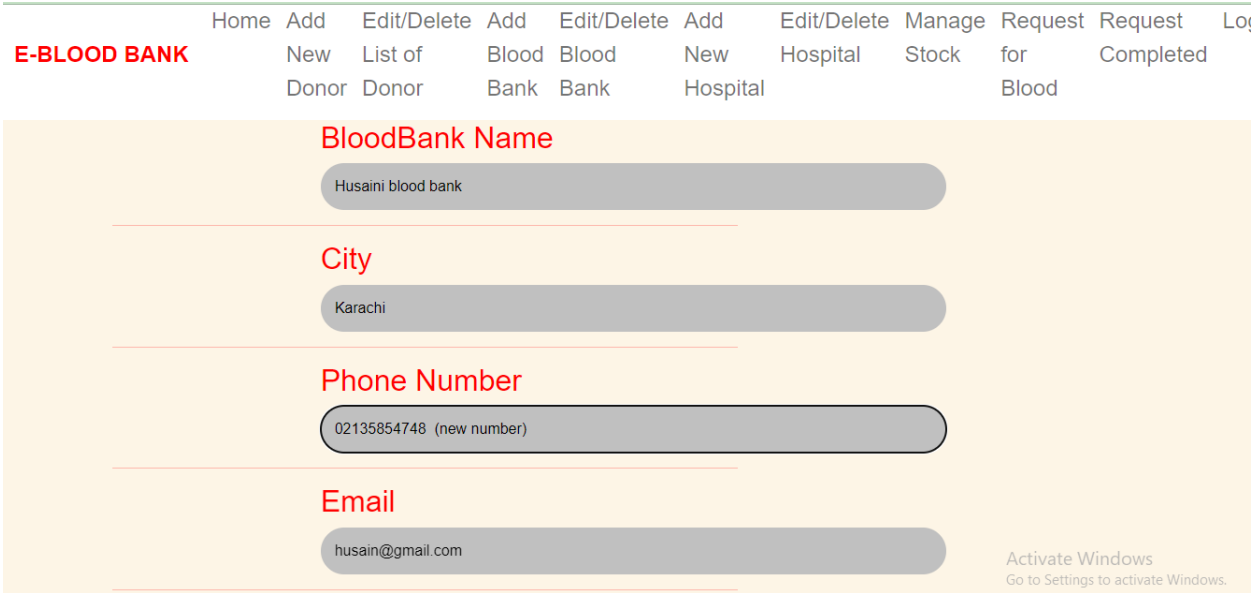
updatebloodbank

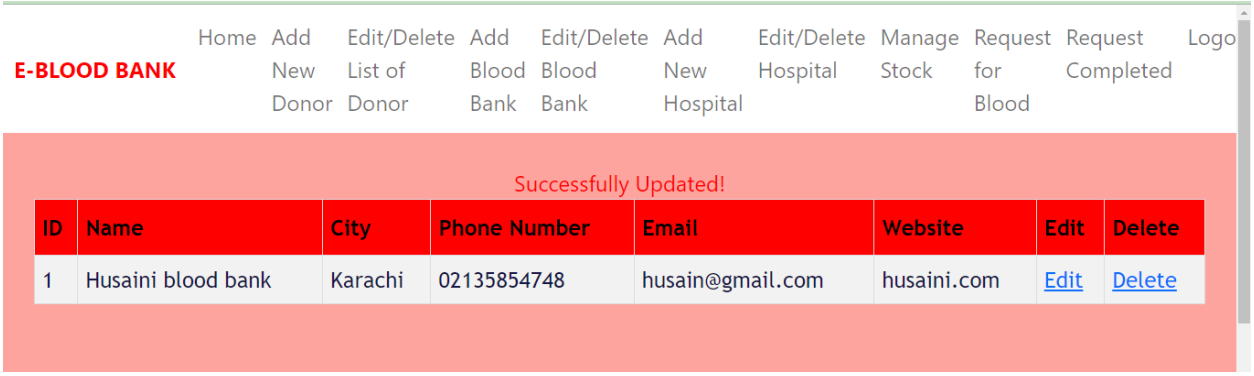
Name: updatebloodbank

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `updatebloodbank`(
tname varchar(100),
tcity varchar(100) ,
tphone_no varchar(100),
temail varchar(200) ,
twebsite varchar(100),
tbank_id int
)
BEGIN
update blood_bank set name=tname,city=tcity,phone_no=tphone_no,
email=temail,website=twebsite where bank_id=tbank_id;
END
```





5.3. Functions

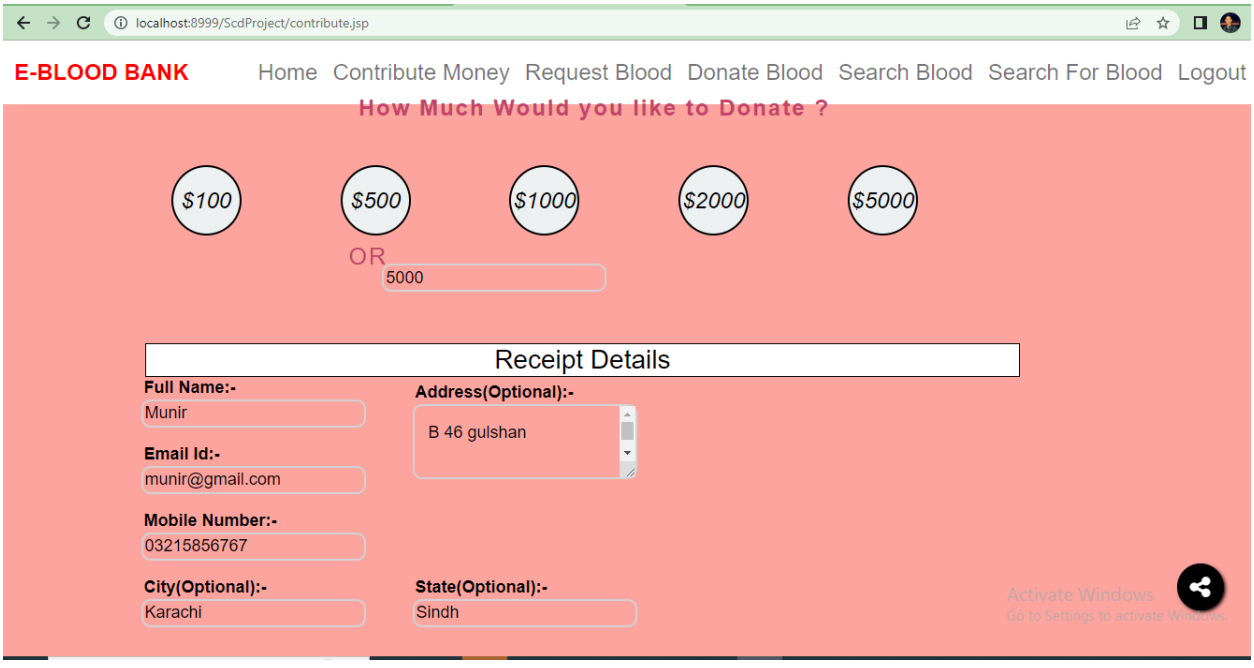
Tax deduction Function

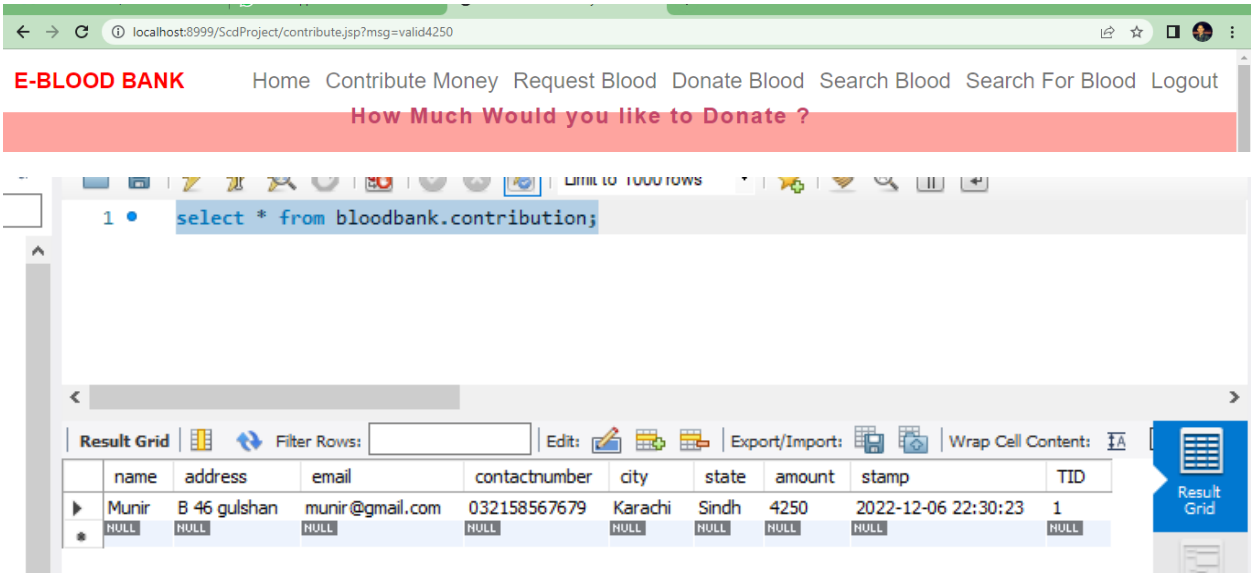
Name: TacDeducted

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
CREATE DEFINER=`root`@`localhost` FUNCTION `TacDeducted`(
    amt INT
) RETURNS int
    DETERMINISTIC
BEGIN
    DECLARE afttax INT;
    SET afttax = amt*0.85;
    RETURN afttax;
END
```



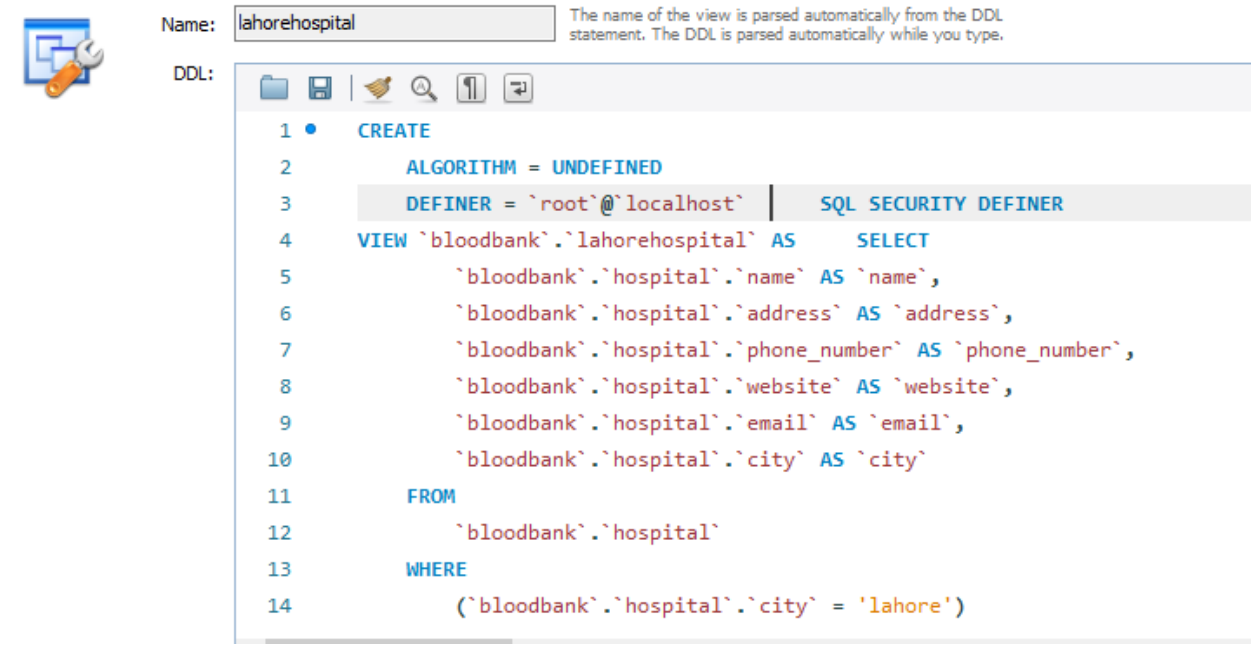


Amount inserted is 4250 as function gave us a new value. 15% tax was deducted.

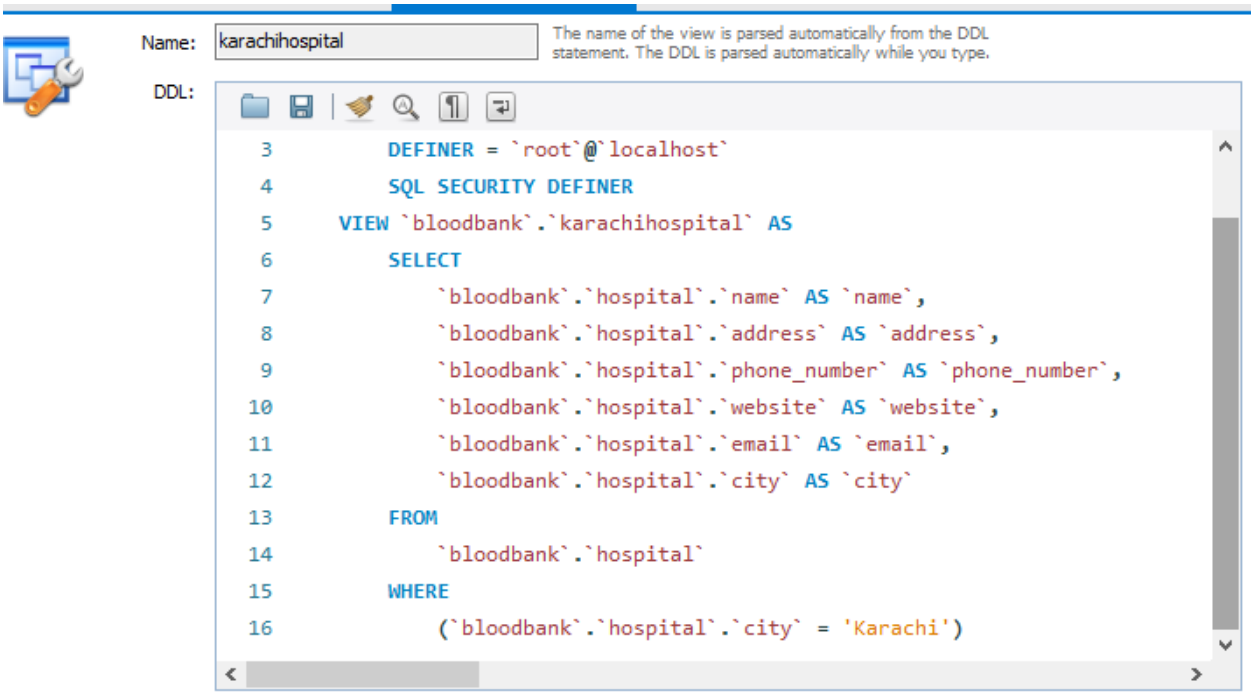
```
<%@page import="java.sql.Timestamp"%>
<%@page import ="java.time.*" %>
<%@page import="java.sql.*"%>
<%
Date date= new Date();
long time = date.getTime();
Timestamp ts = new Timestamp(time);
String d= ts+"";
```

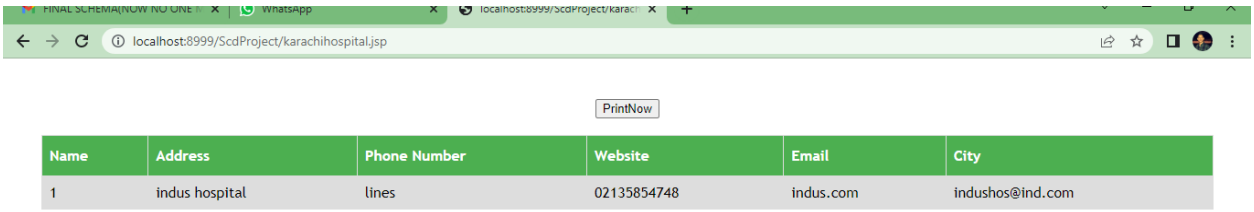
5.4. Views

Lahore Hospital View

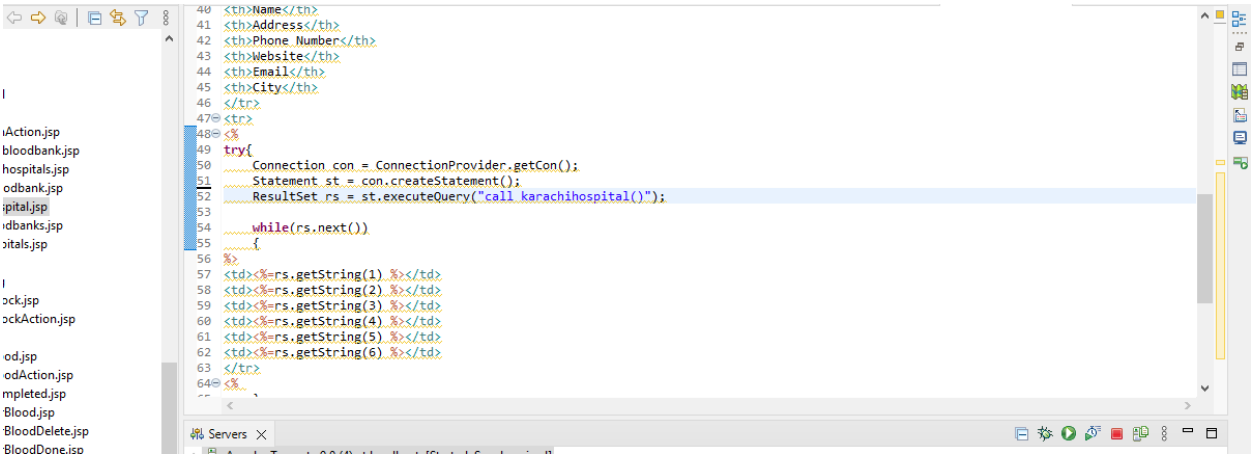


Karachi Hospital View

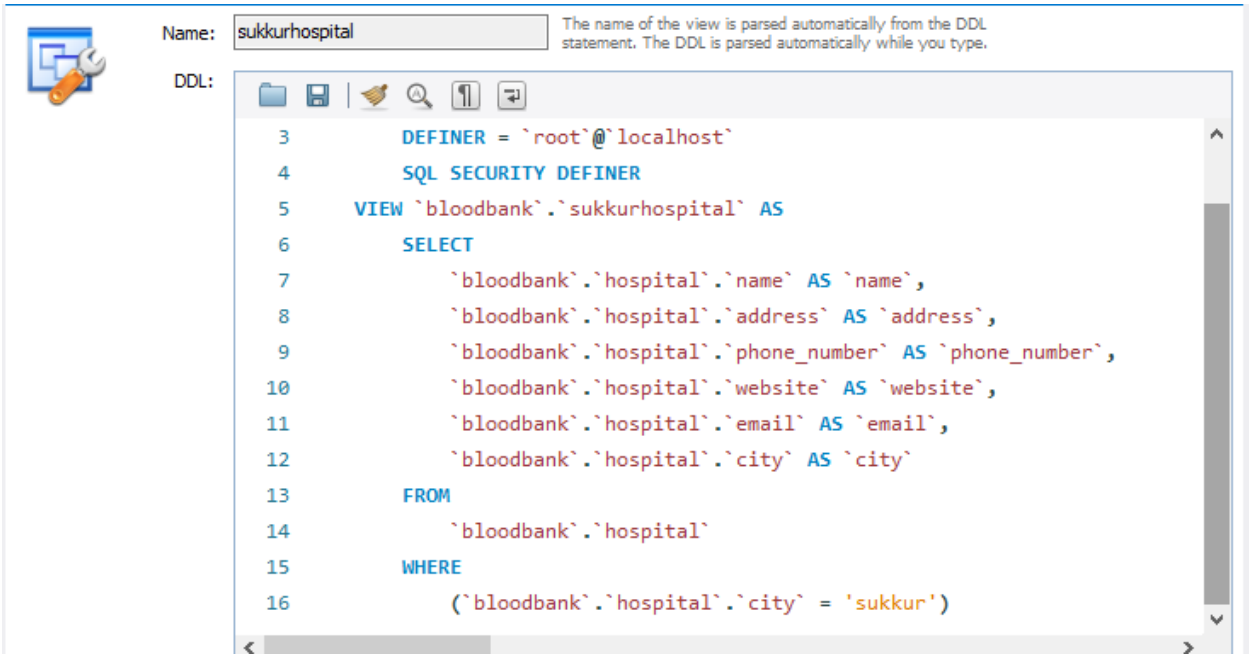




Name	Address	Phone Number	Website	Email	City
1	indus hospital	lines	02135854748	indus.com	indushos@ind.com

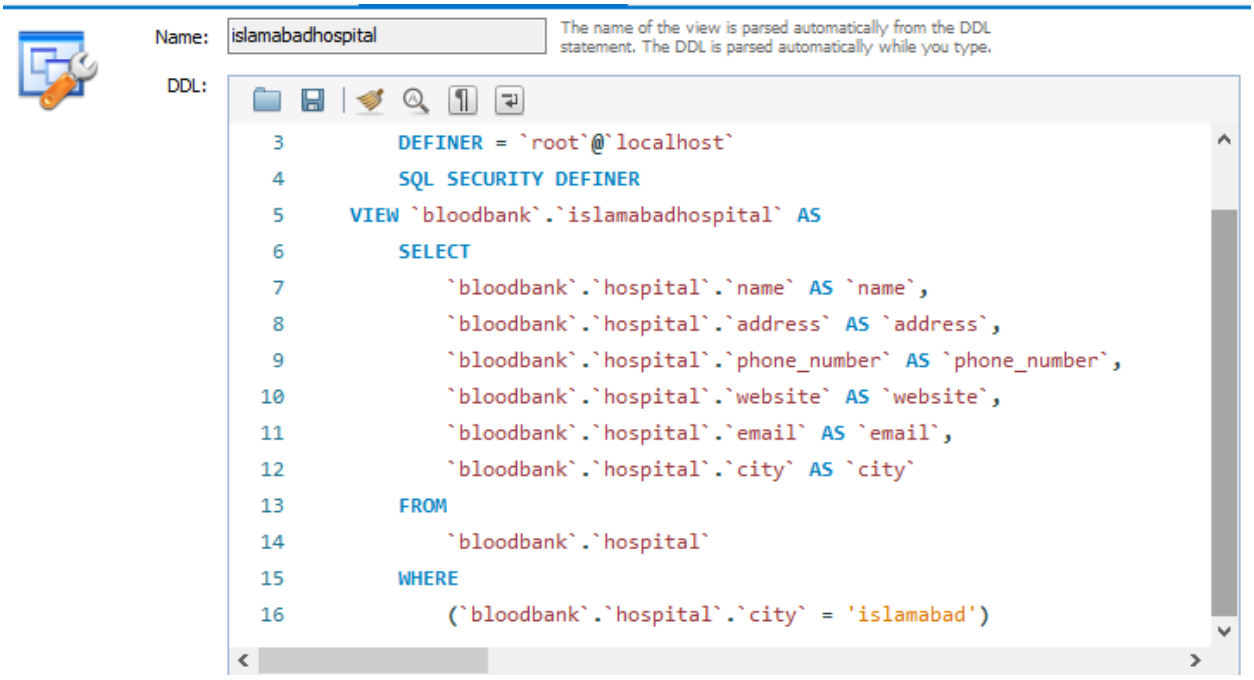


Sukkur Hospital View

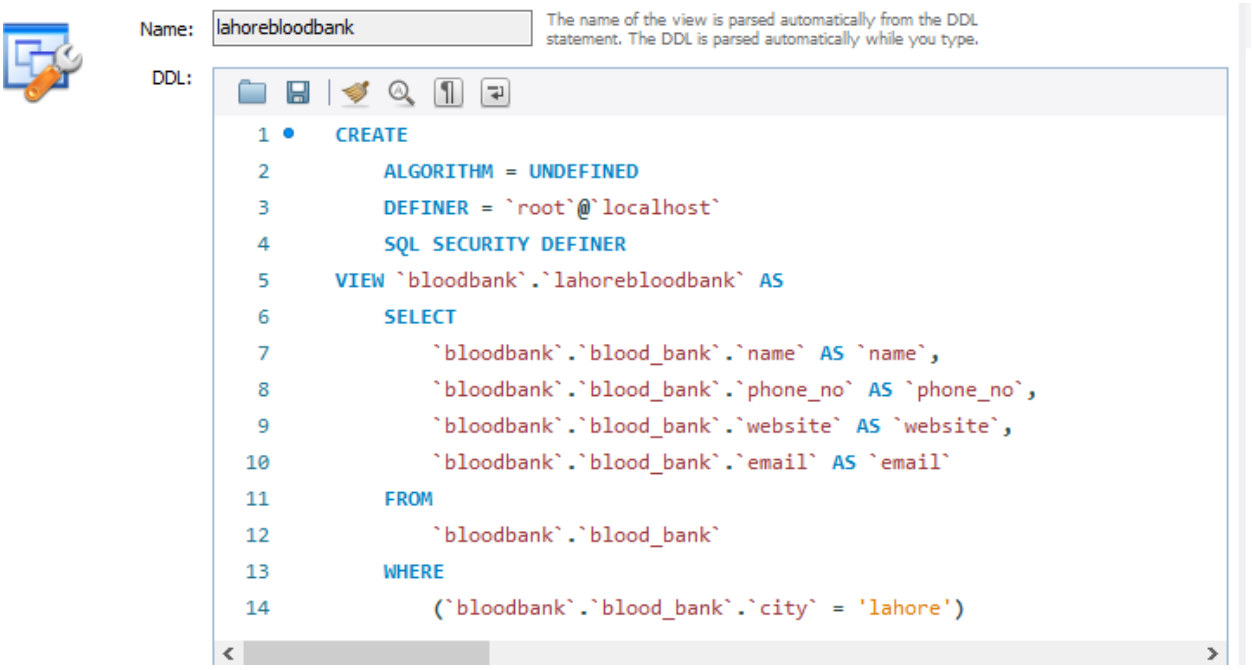


Submitted to Miss Erum Shaheen, Dr. Zulfiqar Ali

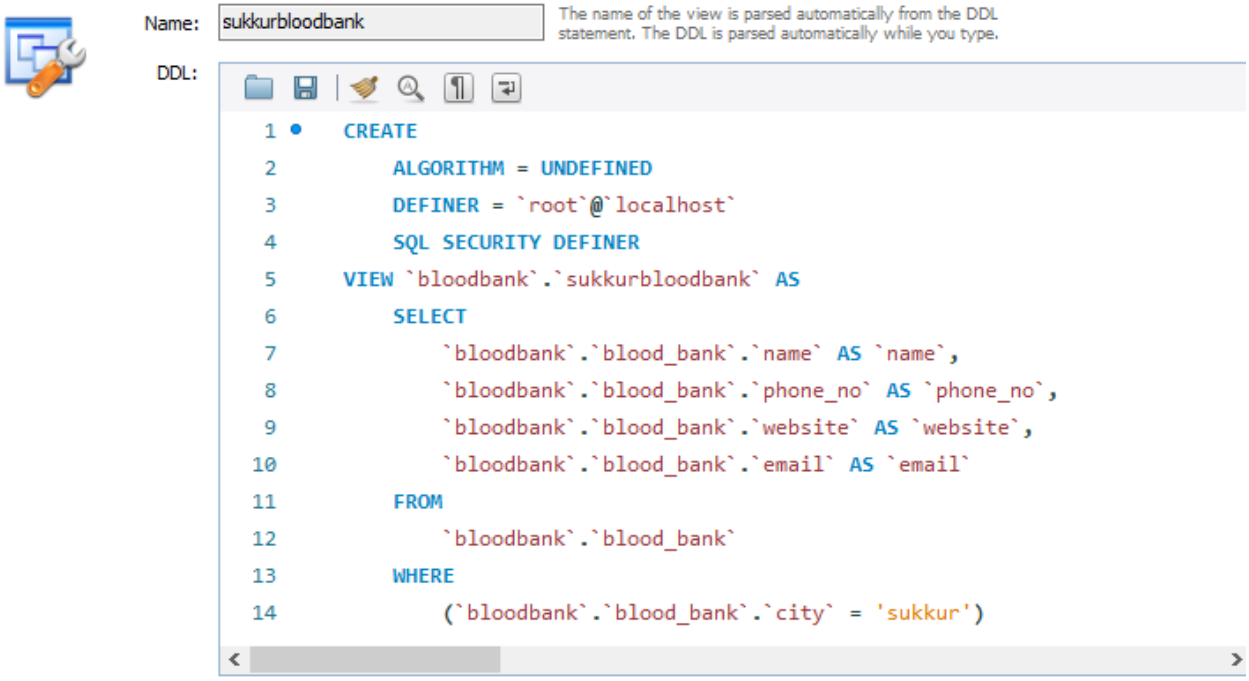
Islamabad Hospital View



Lahore Bloodbank View

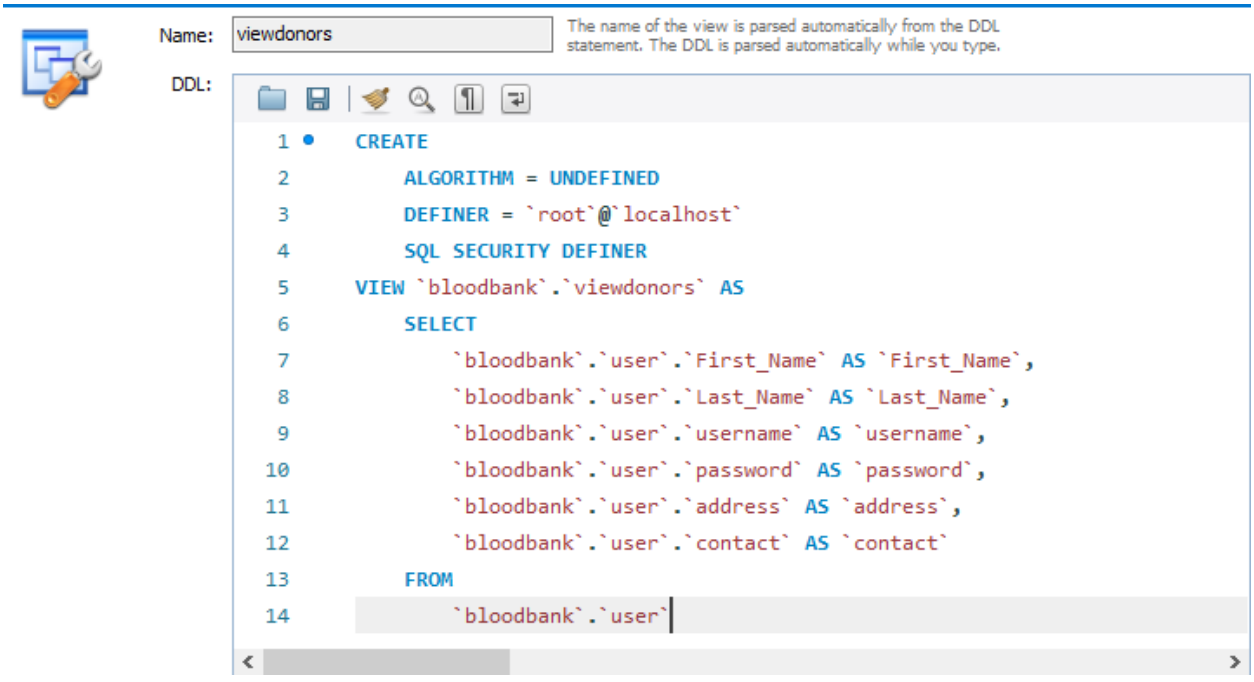


Sukkur bloodbank view




```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `bloodbank`.`sukkurbloodbank` AS
    SELECT
        `bloodbank`.`blood_bank`.`name` AS `name`,
        `bloodbank`.`blood_bank`.`phone_no` AS `phone_no`,
        `bloodbank`.`blood_bank`.`website` AS `website`,
        `bloodbank`.`blood_bank`.`email` AS `email`
    FROM
        `bloodbank`.`blood_bank`
    WHERE
        (`bloodbank`.`blood_bank`.`city` = 'sukkur')
```

Admin can view donors




```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `bloodbank`.`viewdonors` AS
    SELECT
        `bloodbank`.`user`.`First_Name` AS `First_Name`,
        `bloodbank`.`user`.`Last_Name` AS `Last_Name`,
        `bloodbank`.`user`.`username` AS `username`,
        `bloodbank`.`user`.`password` AS `password`,
        `bloodbank`.`user`.`address` AS `address`,
        `bloodbank`.`user`.`contact` AS `contact`
    FROM
        `bloodbank`.`user`
```

5.5. Queries

```
try{
Connection con = ConnectionProvider.getCon();
Statement st = con.createStatement();
ResultSet rs = st.executeQuery("select * from donor where city='"+city+"'and bloodgroup='"+bg+"'");
while(rs.next())
{
```

```
 try{
 Connection con=ConnectionProvider.getCon();
 PreparedStatement ps = con.prepareStatement("insert into user values(?,?,?,?,?,?)");
 ps.setString(1, first_name);
 ps.setString(2, last_name);
 ps.setString(3, username);
 ps.setString(4, password);
 ps.setString(5, address);
 ps.setString(6, contact);

 ps.executeUpdate();
 response.sendRedirect("user.jsp?msg=valid");
```

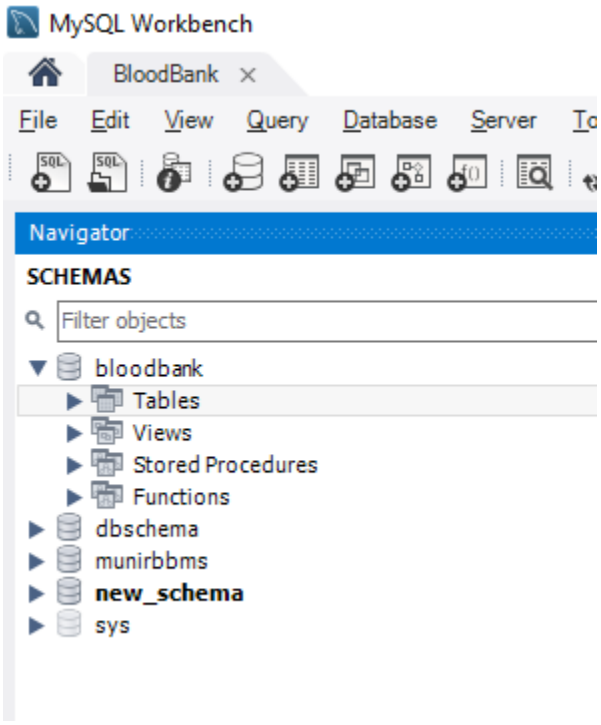
```
<%
 String username=request.getParameter("username");
 session.putValue("username",username);
 String password=request.getParameter("password");
 Connection con = ConnectionProvider.getCon();
 Statement st= con.createStatement();
 ResultSet rs=st.executeQuery("select * from user where username='"+username+"' and password='"+password+"'");
 try{
     rs.next();
             if(rs.getString("password").equals(password)&&rs.getString("username").equals(username))
             {
                 response.sendRedirect("userHome.jsp");
             }
         else{
             response.sendRedirect("userLogin.jsp?msg=invalid");
             }
 }
 catch (Exception e) {
```

Donate Blood Query

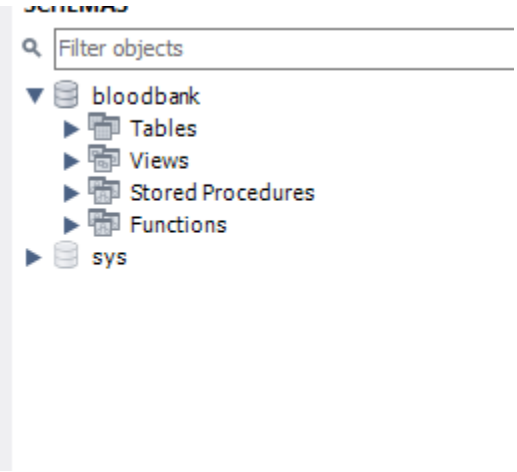
```
159 <%
160 int id=1;
161 try
162 {
163     Connection con = ConnectionProvider.getCon();
164     Statement st = con.createStatement();
165     ResultSet rs = st.executeQuery("select max(transid) from donortransaction");
166     while(rs.next())
167     {
168         id=rs.getInt(1);
169         id=id+1;
170     }%>
171     <div class="container">
172     <h1 style="color:red;">Transaction ID: <%out.print(id); %></h1>
173     <%
174     }
175 catch(Exception e)
176 {}
177 %>
```

```
17
18 try{
19     Connection con=ConnectionProvider.getCon();
20     Statement st= con.createStatement();
21     PreparedStatement ps = con.prepareStatement("insert into donortransaction(transid,donorid,message,hospitalid, date, bloodgroup) v
22     //PreparedStatement ps = con.prepareStatement("call addbloodbank(?,?,?,?,?,?)");
23     ResultSet rs=st.executeQuery("select bloodgroup from donor where id='"+did+"'");
24     rs.next();
25     String bg=rs.getString("bloodgroup");
26     ps.setString(1, id);
27     ps.setString(2, did);
28     ps.setString(3, confirm);
29     ps.setString(4, hid);
30     ps.setString(5,d);
31     ps.setString(6,bg);
32     ps.executeUpdate();
33     response.sendRedirect("donateblood.jsp?msg=valid");
34 }
35 catch(Exception e)
36 {
37     response.sendRedirect("donateblood.jsp?msg=invalid");
38 }
39 %>
```

6.0. Connectivity Screenshots

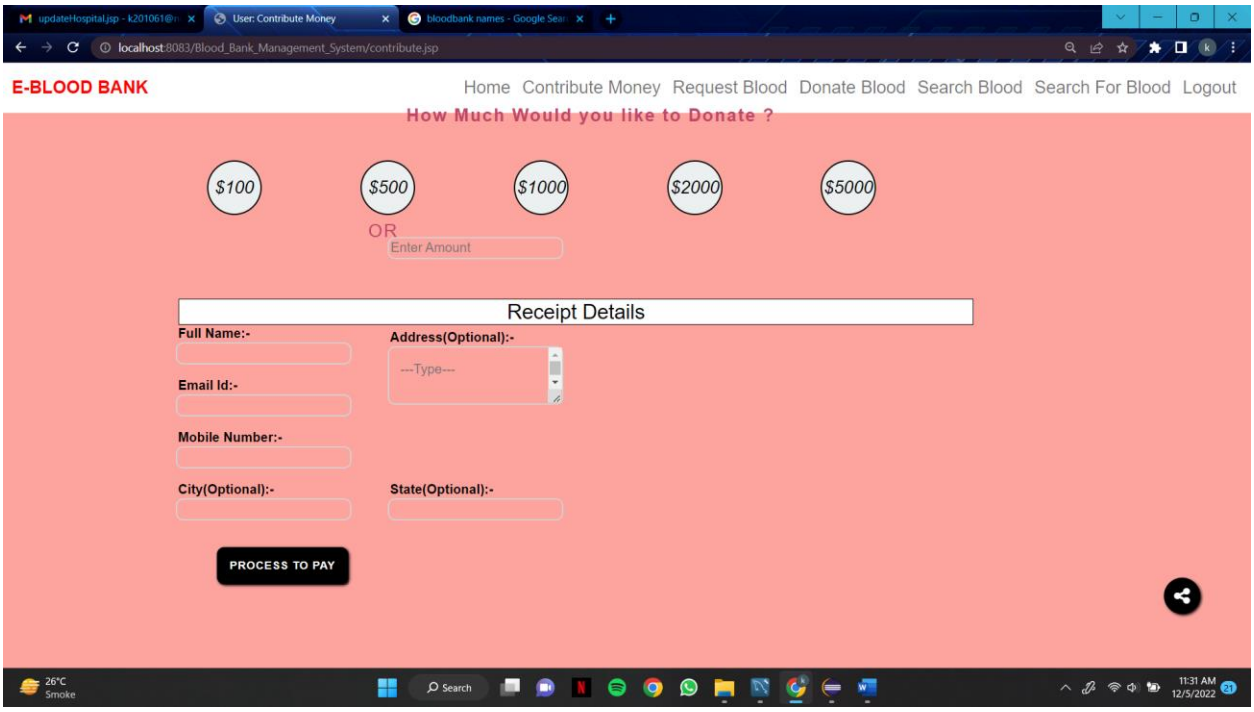


```java
package Project;
import java.sql.*;

public class ConnectionProvider {

    public static Connection getCon()
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/bloodbank","root","fast");
            return con;
        }
        catch(Exception e)
        {
            System.out.print(e);
            return null;
        }
    }

}
```

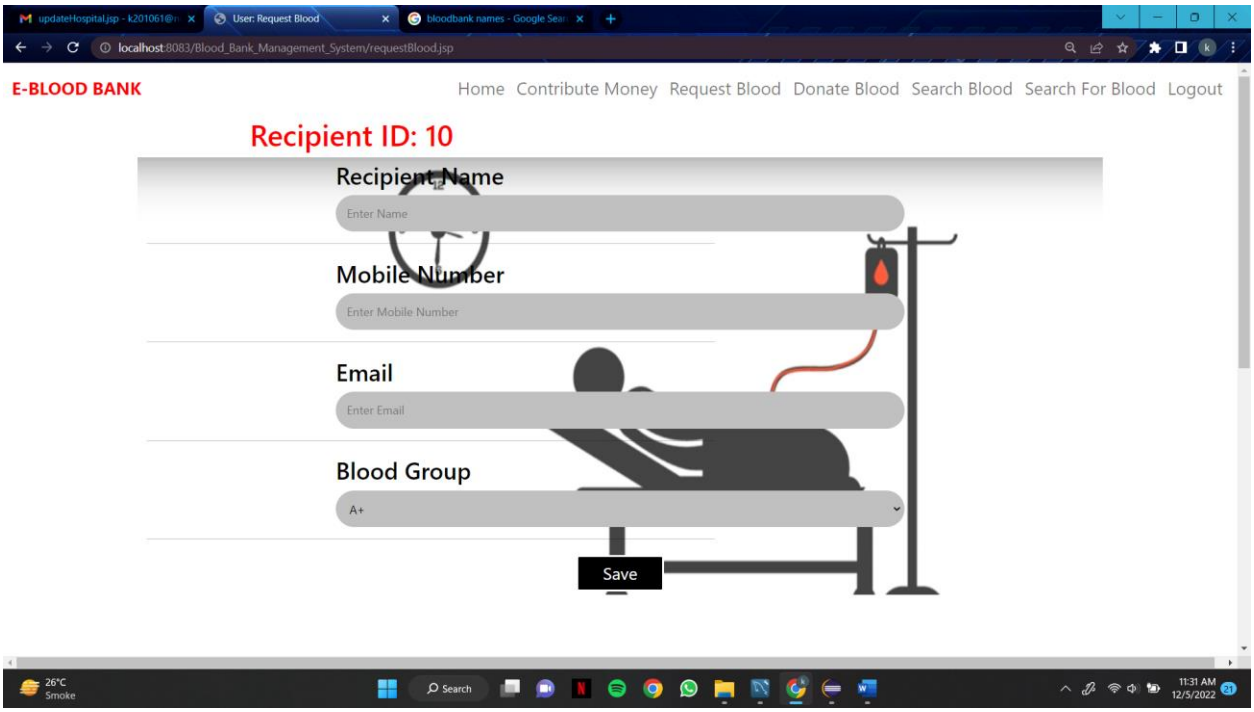


7.0. Project Screenshots

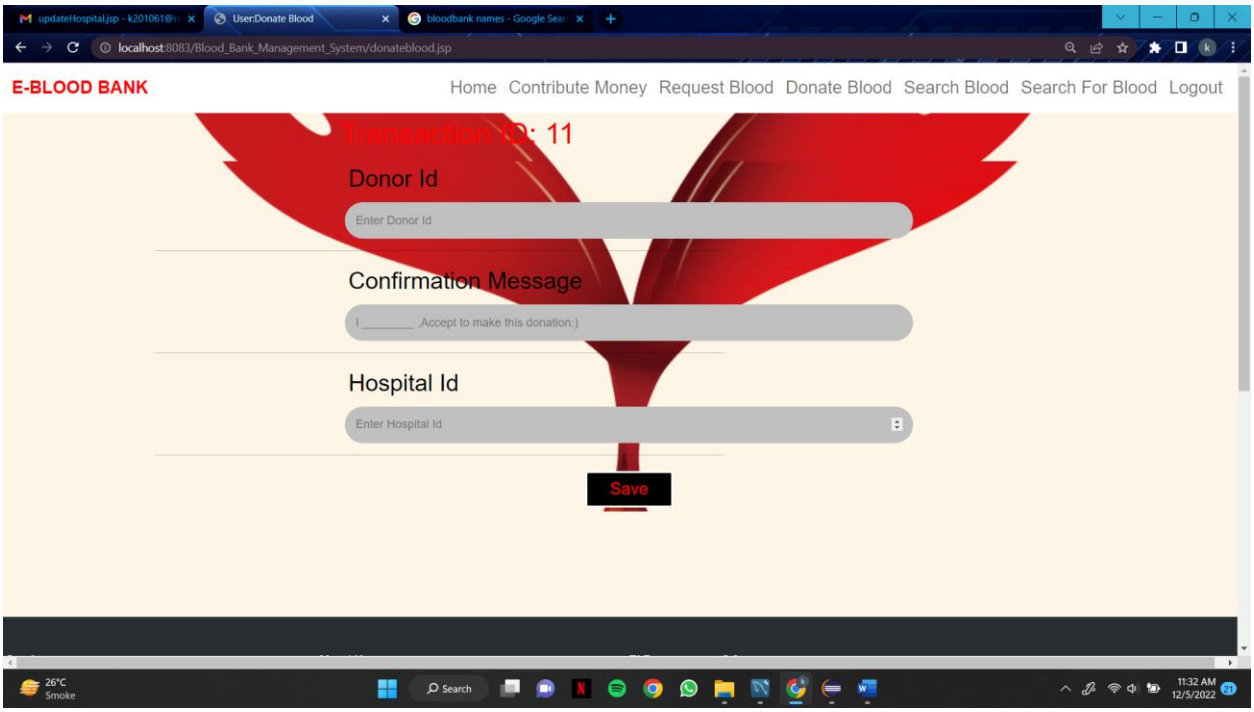
7.1. User money donation



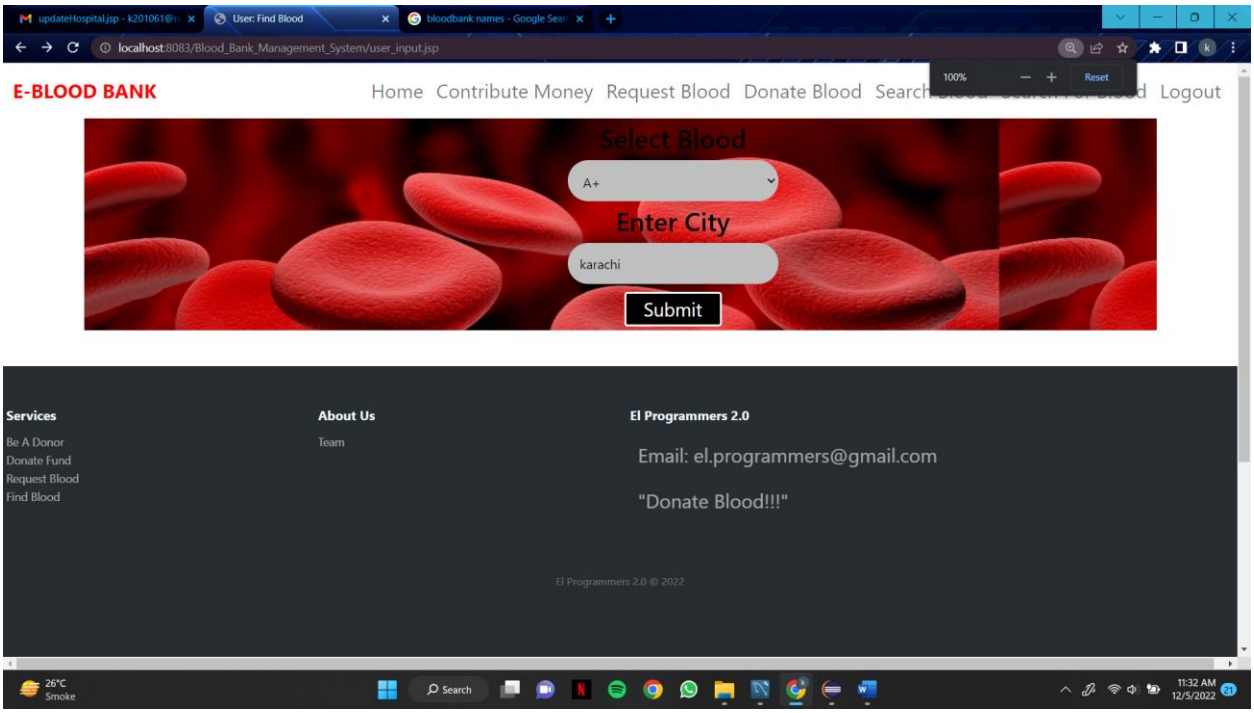
7.2. User receiver id



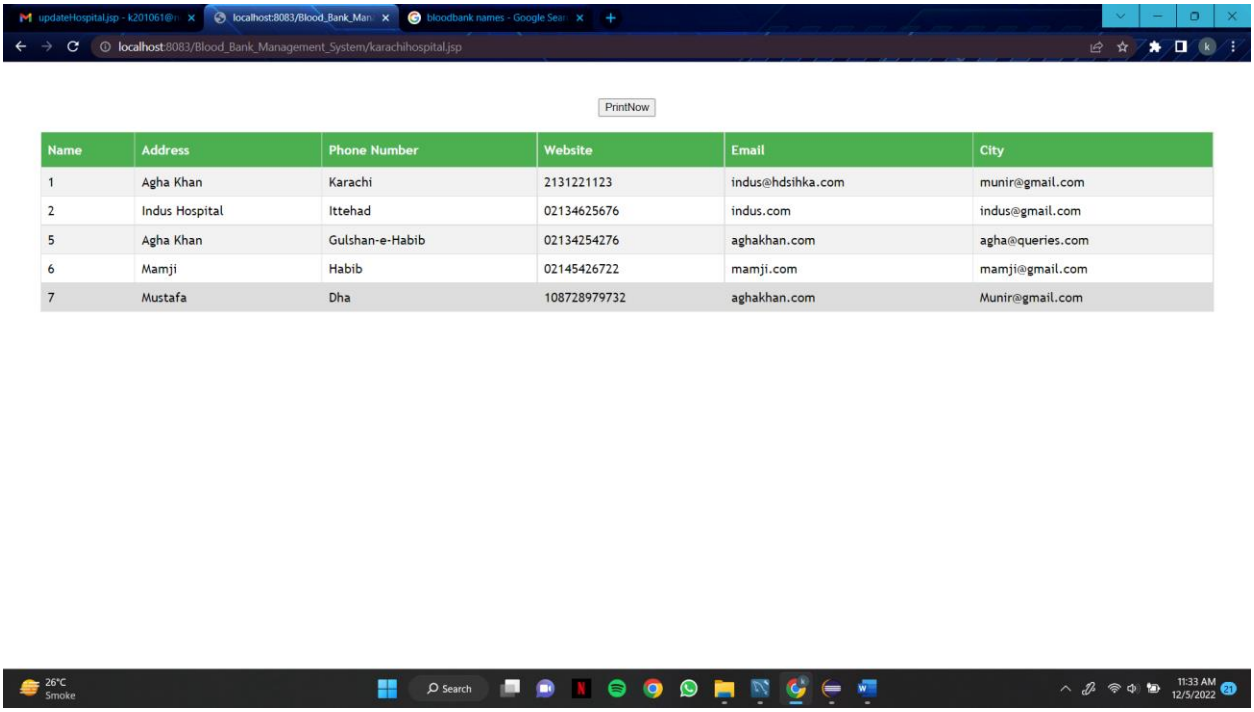
7.3. Donor donation blood



7.4. Search for blood

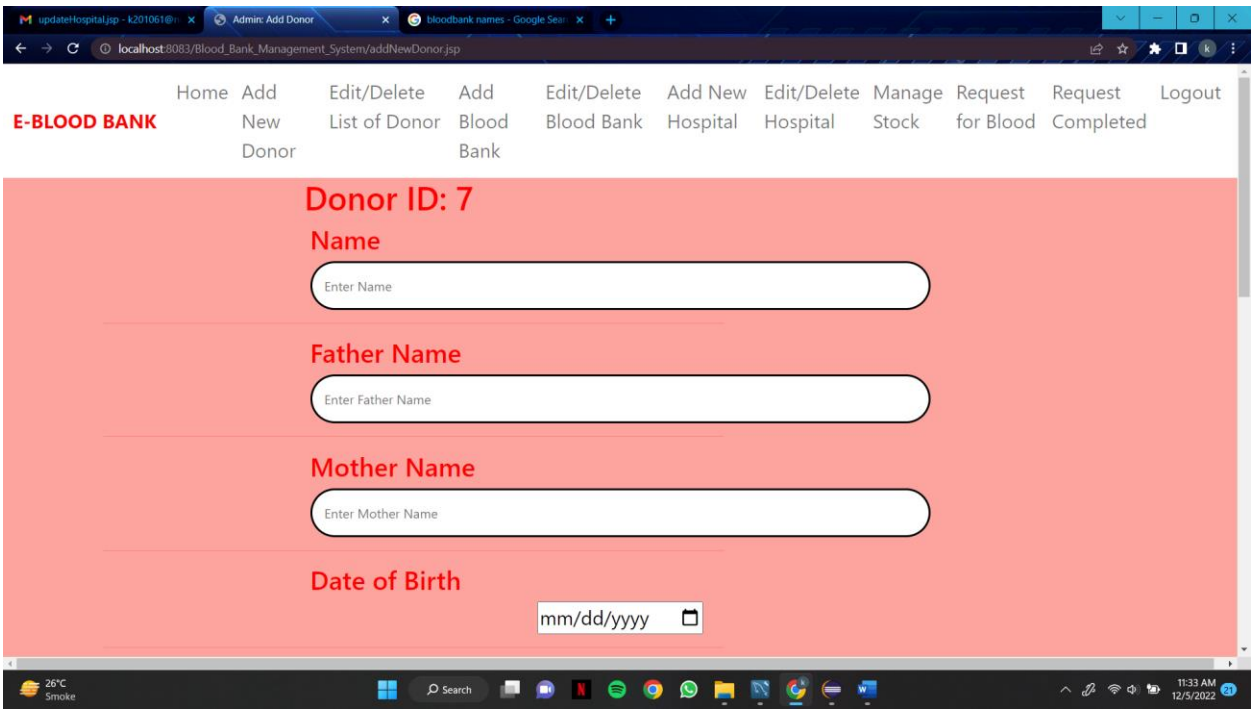


7.5. Show hospital data



Name	Address	Phone Number	Website	Email	City
1	Agha Khan	Karachi	2131221123	indus@hdsihka.com	munir@gmail.com
2	Indus Hospital	Ittehad	02134625676	indus.com	indus@gmail.com
5	Agha Khan	Gulshan-e-Habib	02134254276	aghakhan.com	agha@queries.com
6	Mamji	Habib	02145426722	mamji.com	mamji@gmail.com
7	Mustafa	Dha	108728979732	aghakhan.com	Munir@gmail.com

7.6. Add donor



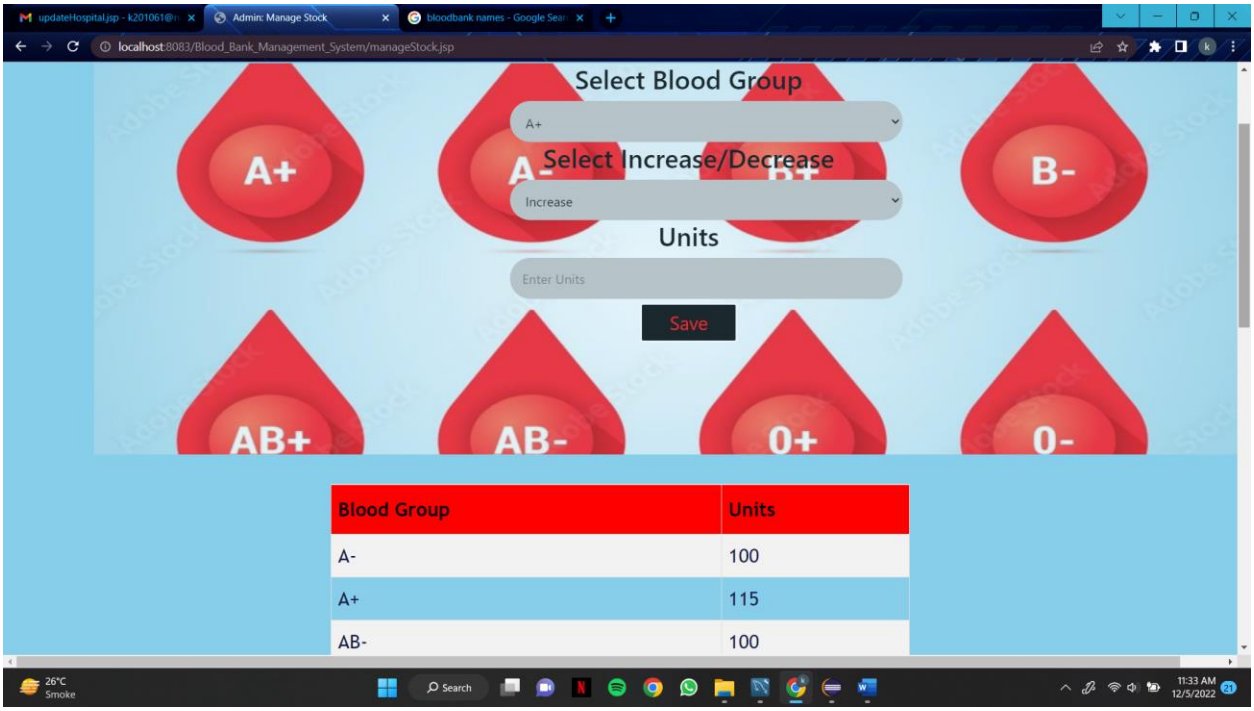
7.7. Edit delete hospital,donor,bloodbank

ID	Name	Father Name	Mother Name	Date Of Birth	Weight	Mobile Number	Gender	Email	Blood Group	City	Edit	Delete
1	Mustafa	Zahid	Aasia	2022-12-28	54.0	9302847	Male	thewatch786@hotmail.com	A+	Karachi	Edit	Delete
2	Musti	Zahid Jamil	Aasia	2022-12-28	55.0	0321321321	Male	shabby@nu.edu.pk	A+	karahi	Edit	Delete
3	Dehya	Khurraim	Aisha	2022-12-24	50.0	0321321321	Male	dehya@nu.edu.pk	A+	Lahore	Edit	Delete
4	Raza Ali	Abidi	Koyal	2022-08-11	65.0	0321321321	Male	thewatch786@hotmail.com	A+	Lahore	Edit	Delete
6	Pinky	Fast	nida	2022-	89.0	03022334330	Female	munist@gmail.com	AB+	Lahore	Edit	Delete

ID	Name	City	Phone Number	Email	Website	Edit	Delete
2	Tapowan Blood Bank Samitee	Srinagar	0378279	thewatch786@hotmail.com	hsja.com	Edit	Delete
3	chugtai lab	q street	123231213	chugtai@gmail.com	chugtailabs@cl.com	Edit	Delete
5	Taposha	Islamabad	0213426213	taposha@gmail.com	Taposhi.com	Edit	Delete
6	Indus Blood Bank	Karachi	0842189112	indusbloodbank@edu.pk	indus@hdsihka.com	Edit	Delete
7	PinkyBB	Karachi	0213542143	pinky@nu.edu.pk	pinky.com	Edit	Delete
8	rafiq	Karachi	213213213213	shabby@nu.edu.pk	Taposhi.com	Edit	Delete
9	Rotary Blood Bank.	Sukkur	02135172728	rotary@gamil.com	rotary.com	Edit	Delete

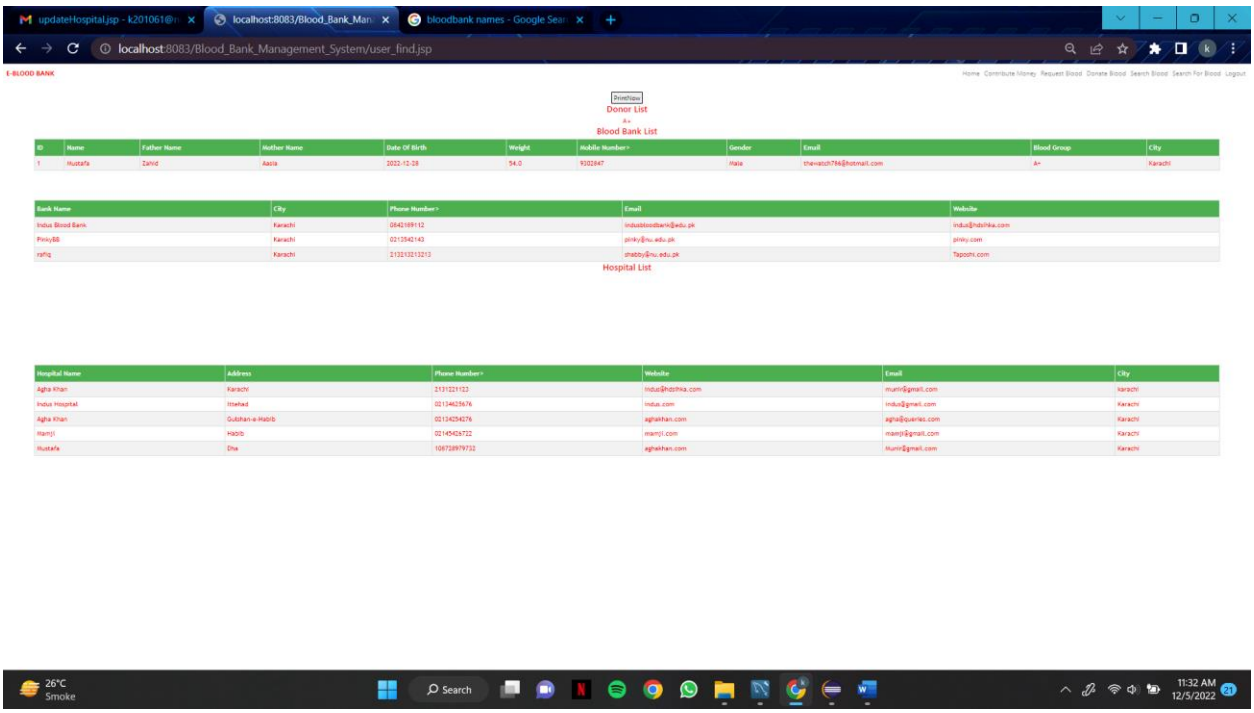
ID	Name	Address	Phone Number	Website	Email	City	Edit	Delete
1	Agha Khan	Karachi	2131221123	indus@hdsihka.com	munir@gmail.com	karachi	Edit	Delete
2	Indus Hospital	Ittehad	02134625676	indus.com	indus@gmail.com	Karachi	Edit	Delete
4	Mamji Hospital	Gulshan-e-Jamaal	0213256326	mamji.com	mamjiqueries@gmail.com	Lahore	Edit	Delete
5	Agha Khan	Gulshan-e-Habib	02134254276	aghakhan.com	agha@queries.com	Karachi	Edit	Delete
6	Mamji	Habib	02145426722	mamji.com	mamji@gmail.com	Karachi	Edit	Delete
7	Mustafa	Dha	108728979732	aghakhan.com	Munir@gmail.com	Karachi	Edit	Delete

7.8. Manage stock

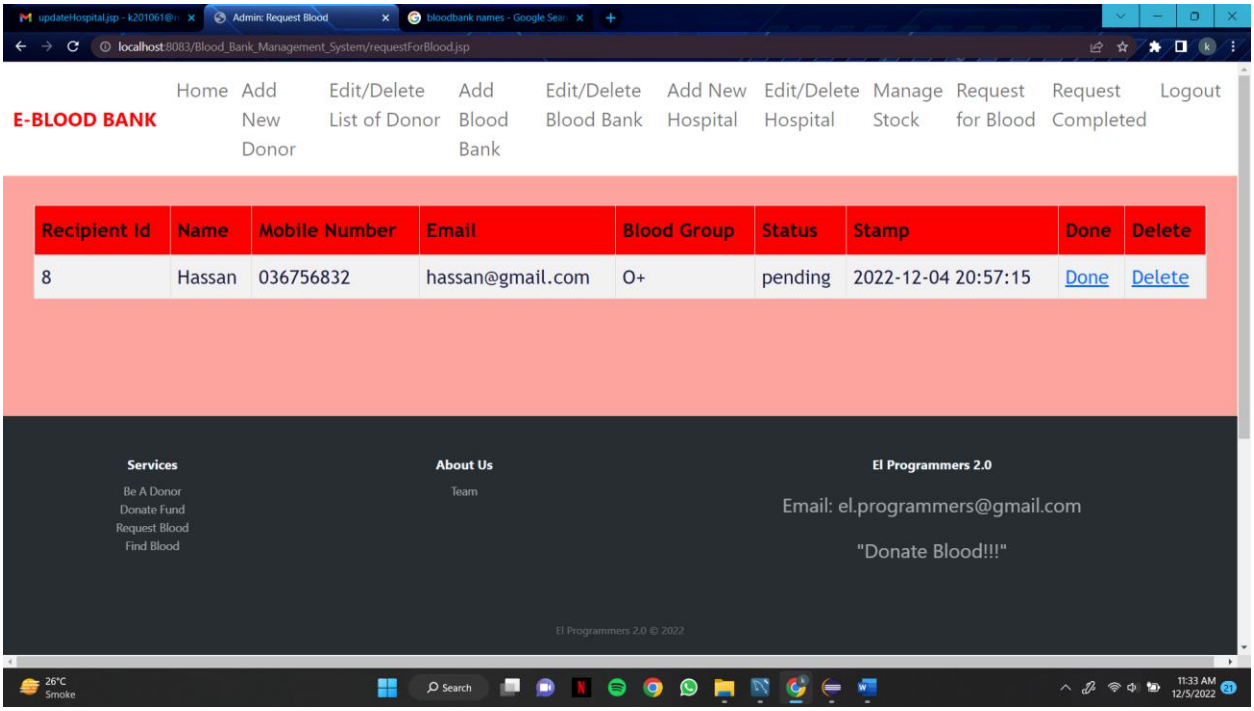


Blood Group	Units
A-	100
A+	115
AB-	100

7.9. Search for blood



7.10. Requests status



7.11. Check requests completed