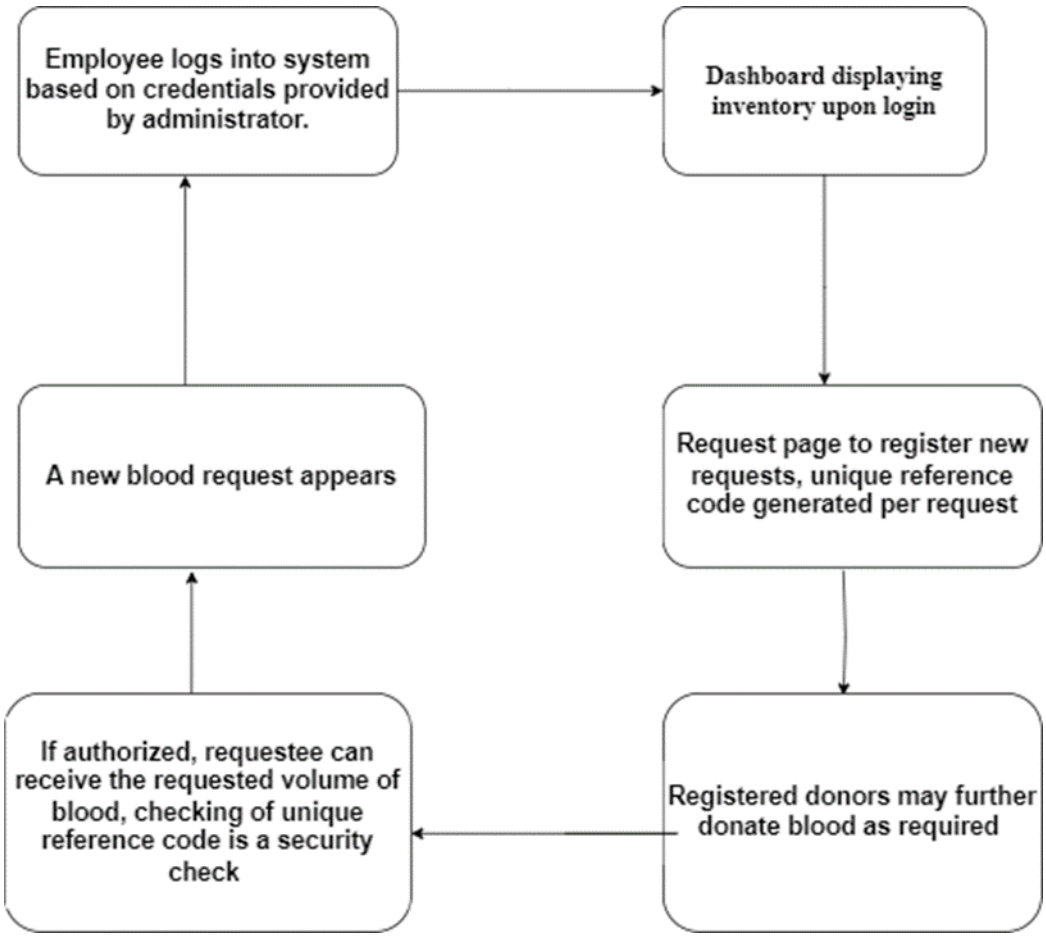
BLOOD DONOR MANAGEMENT SYSTEM

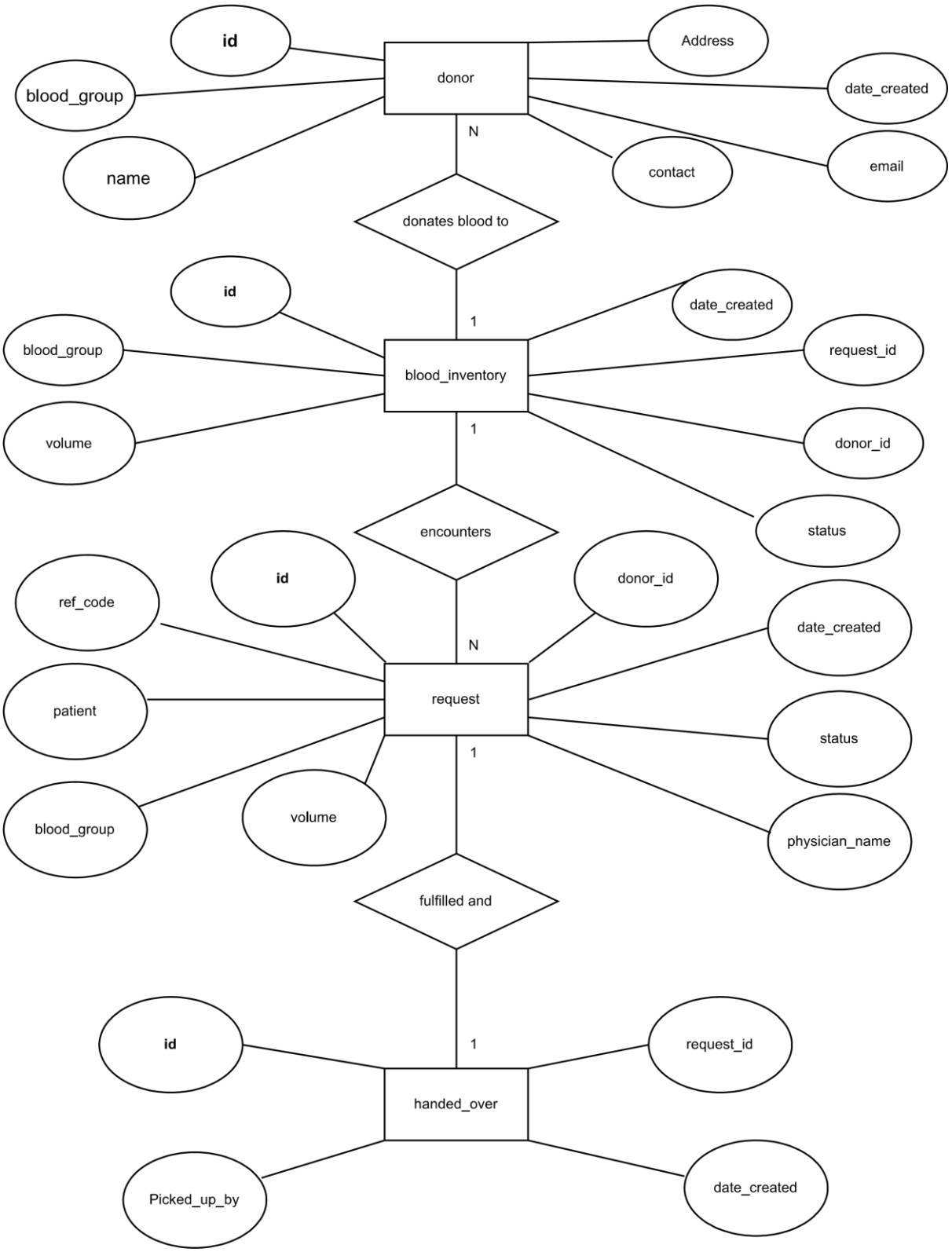
1. Candidate System

A Blood Donor Management System is utilised in individual hospitals for the purpose of managing inventory of Blood in their establishment.

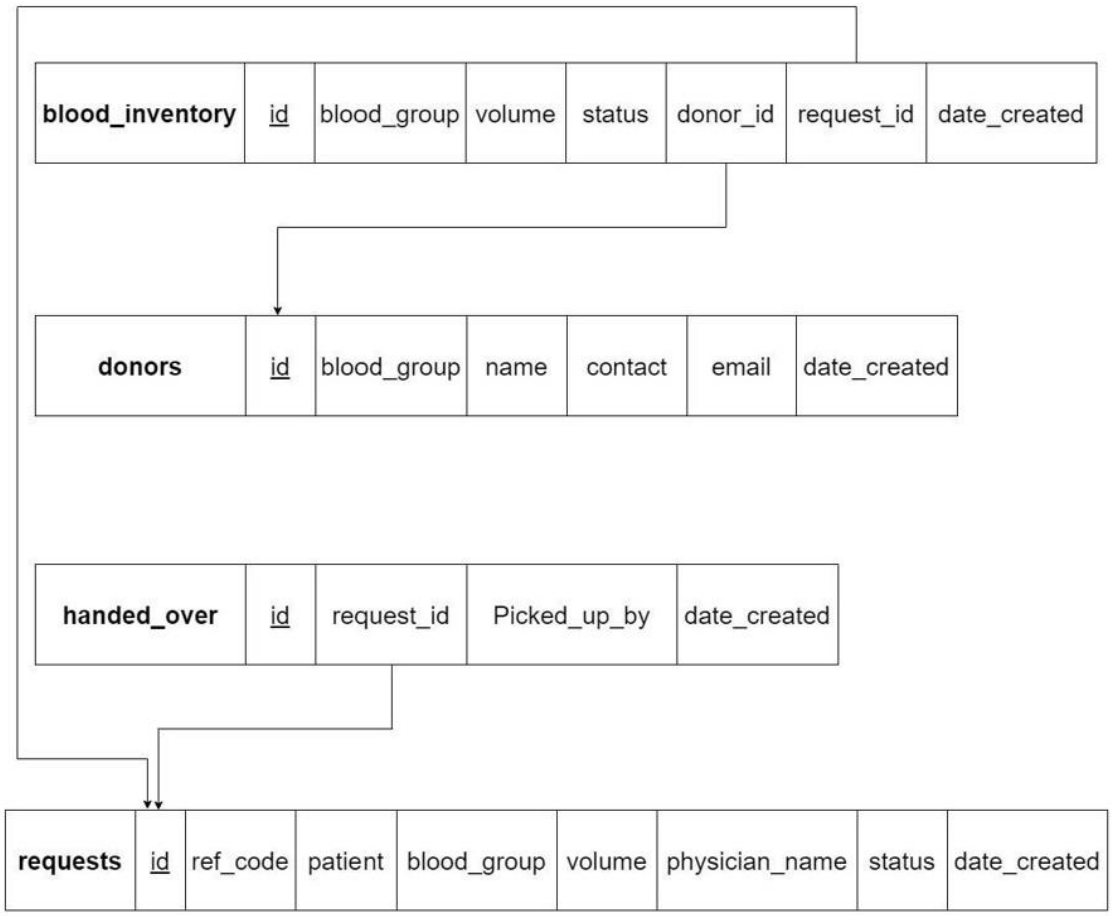
A purely employee operated system wherein various operations involving registrations of donors, recording donations, managing requests, supervising the receiving of authorized blood donations, means of adding users to the system by administrator based on privileges associated to employee job title.



2. ER Diagram:


id
Address
donor
blood_group
date_created
N
name
contact
email
donates blood to
id
date_created
1
blood_group
blood_inventory
request_id
volume
1
donor_id
encounters
status
id
donor_id
ref_code
date_created
N
patient
request
status
1
blood_group
volume
physician_name
fulfilled and
id
1
request_id
handed_over
Picked_up_by
date_created

3. Schema


blood_inventory
id
blood_group
volume
status
donor_id
request_id
date_created
donors
id
blood_group
name
contact
email
date_created
handed_over
id
request_id
Picked_up_by
date_created
requests
id
ref_code
patient
blood_group
volume
physician_name
status
date_created

4. Normalized Tables:

There are 5 tables present, namely:

1. Blood inventory (contains the information related to the available blood in the inventory)
2. Donors (contains the information related to the registered donors in the system)
3. Requests (contains the information related to registered requests put forth by patients)
4. Handed -over requests (Contains the information related to requests which have been approved)
5. Users (Contains the information related to the list of users that can access the system and their login credentials)

NOTE: The above tables are normalized up to BCNF form.

5. Explanation of each form:

1. Login:

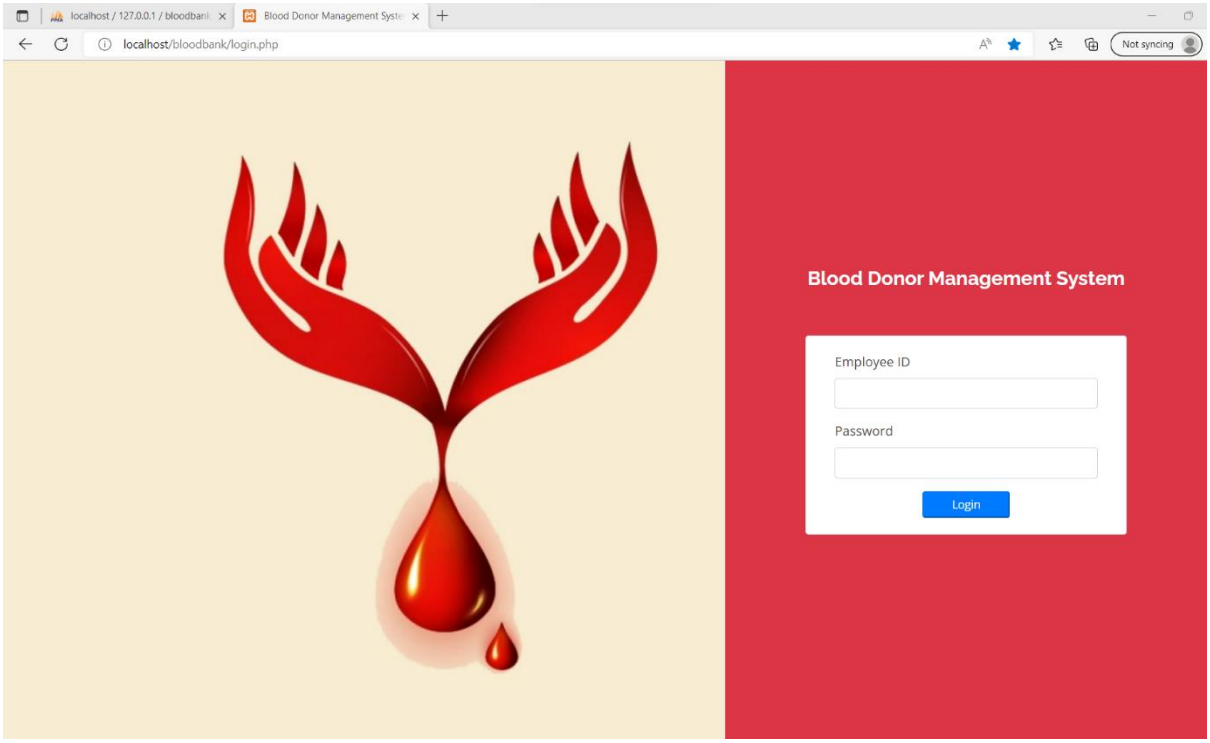


Figure 1

The login form(Figure 1) contains two main credentials:

1. Employee ID
2. Password

Employee should click login to enter the system.

Purpose: To ensure that an authorized user can access the system.

2. New Donor Registration Form:

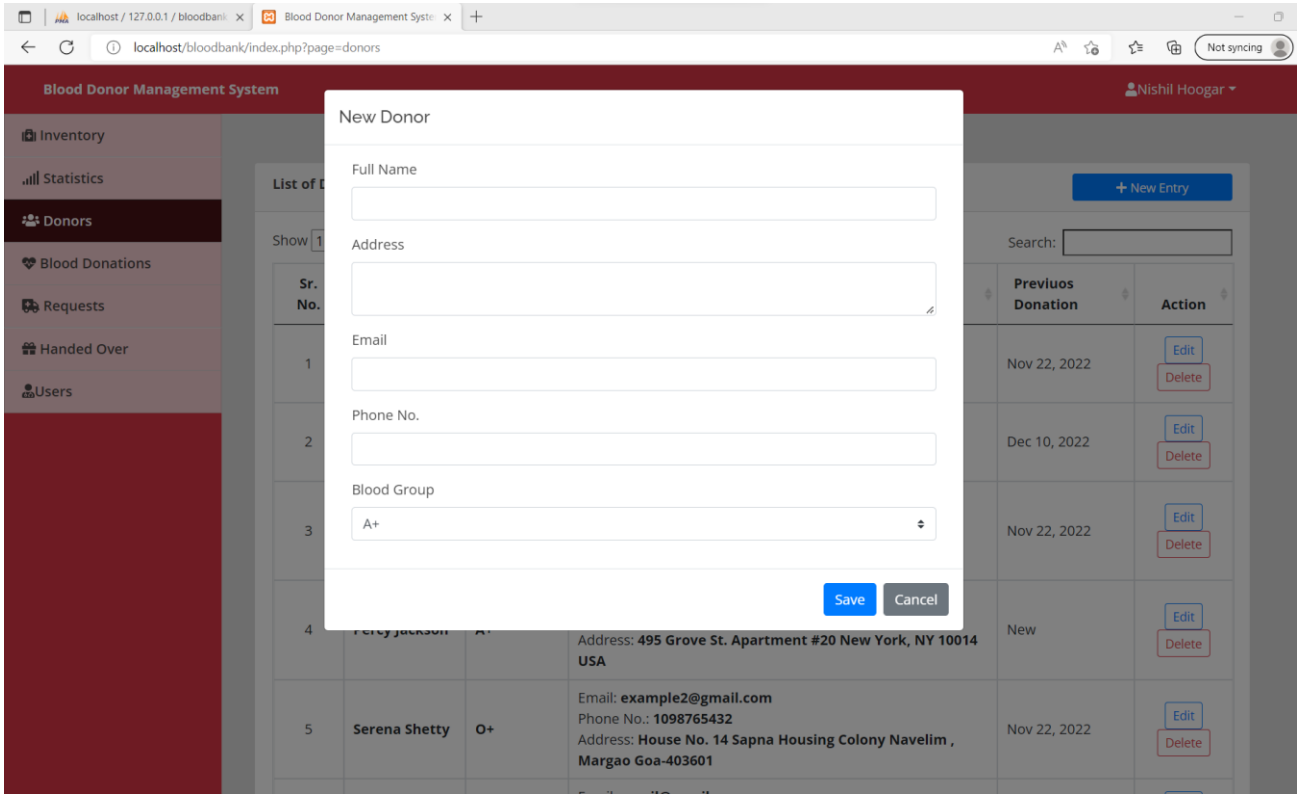


Figure 2

The New Donor Registration form(Figure 2) contains the following fields:

1. Full Name-Field to take Name as an input.
2. Address-Field to take Address as an input.
3. Email-Field to take Email Address as an input.
4. Phone No.-Field to take Phone No. as an input
5. Blood Group- A drop down selection with all blood groups as options.

Purpose: Collect information of a New Donor who wishes to register.

3.New Donation Form:

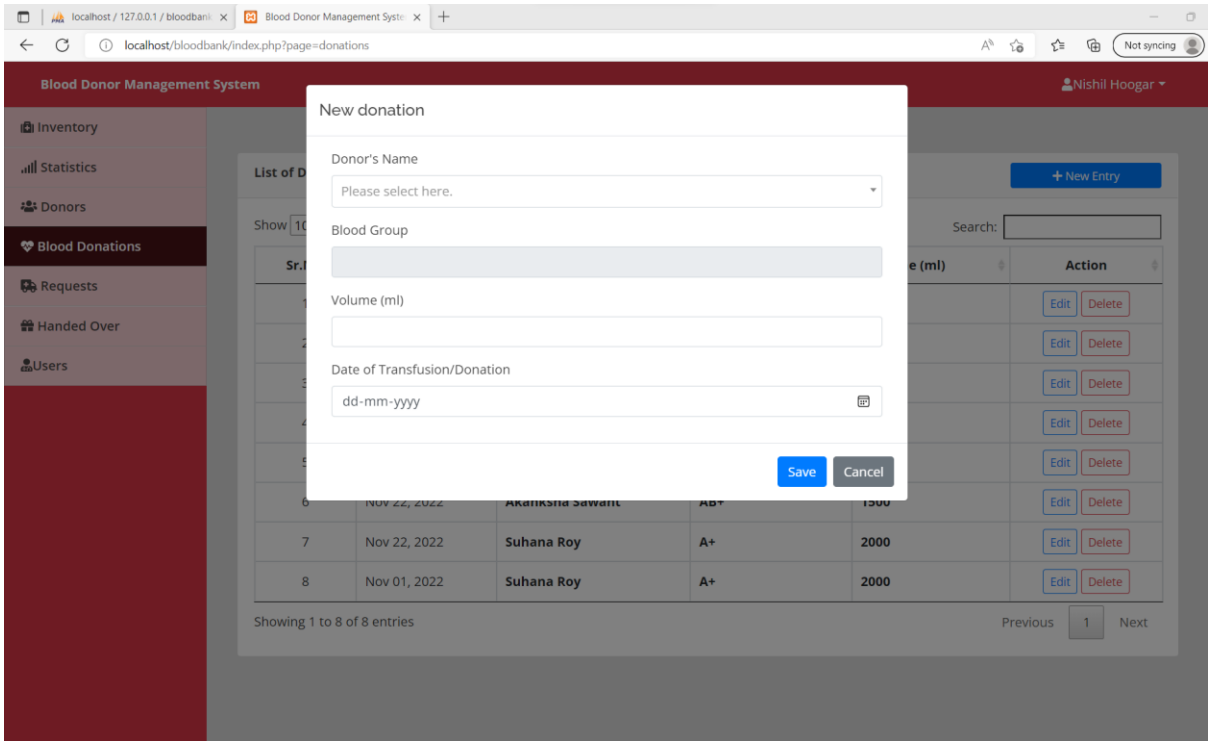


Figure 3

The New Donation Form (Figure 3) contains the following fields:

1. Donor's Name- A drop down selection with all registered blood donor's names as options.
2. Blood Group- A static field reflecting the blood group of the particular donor.
3. Volume- Records the volume of blood being donated.

4. Date of Transfusion/Donation-Records the current date on which the donation is occurring.
   Purpose: Collect information of a New Donation.

4.New Request Form:

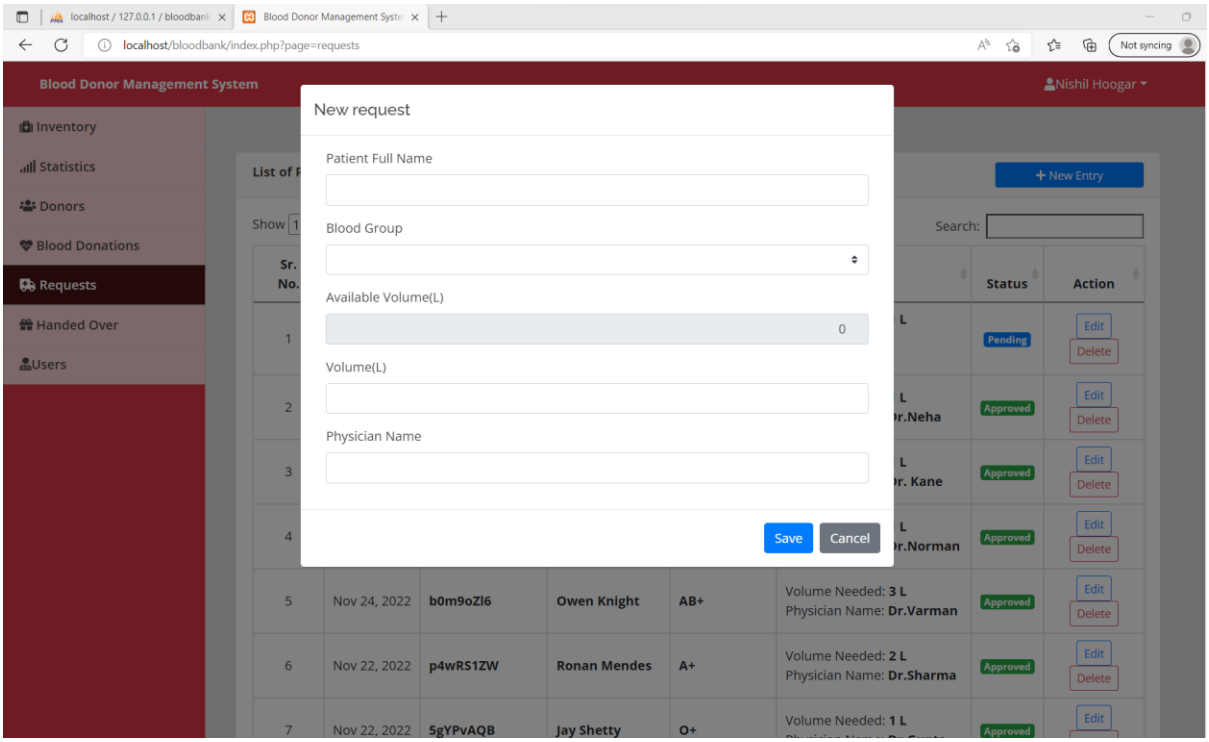


Figure 4

The New Request Form(Figure 4) contains the following fields:

1. Patent Full Name- Field to take Name as an input.
2. Blood Group- A drop down selection with all blood groups as options.
3. Available Volume- A static field which displays the volume of blood of the particular type that is available.

4. Volume- Records the volume of blood that is requested.
5. Physician Name- Field to take Name of the Physician as an input.

Purpose: Collect information of a new request.

5. New Handover Form:

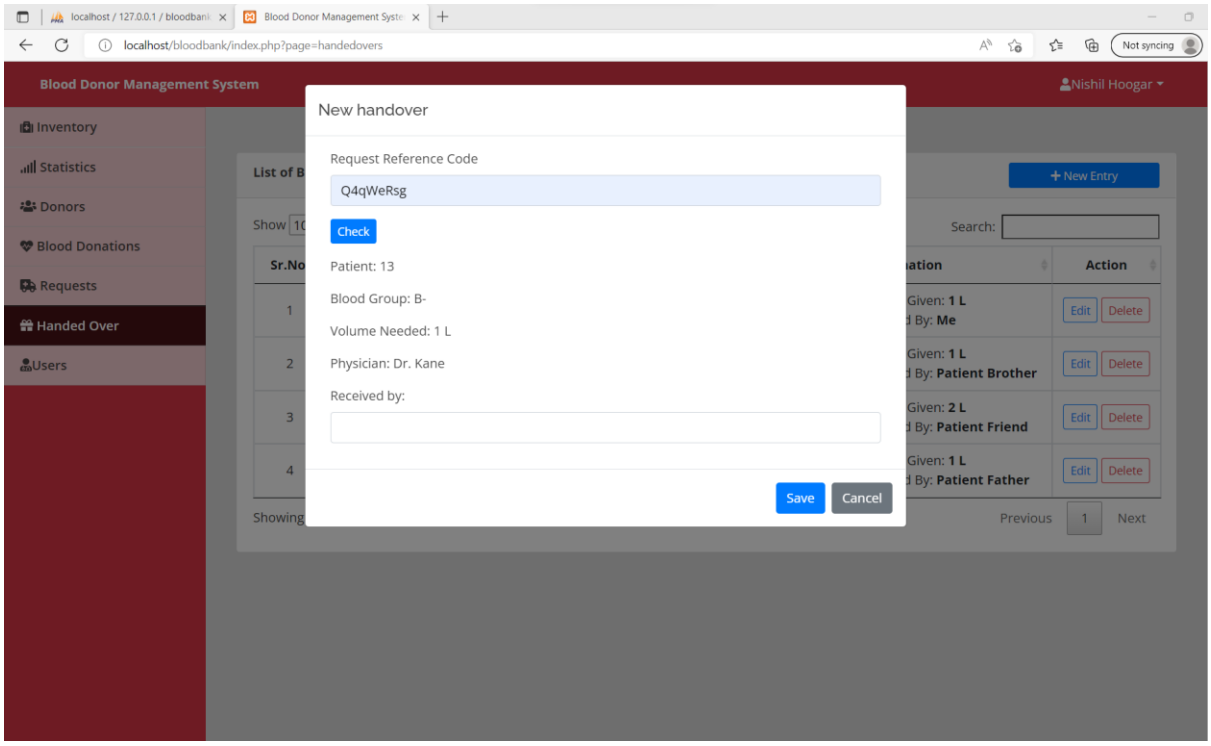


Figure 5

The New Handover Form (Figure 5) contains the following fields:

1. Request Refence Code- A unique reference code that has been generated upon submission of request has to be provided as a input in this field.
2. A check button- To check the status of the particular request (Pending/Approved) identified by the unique

reference code. Further displays information about the patient.

3. Received By: A field to enter the name of the Physician that has authorized the request.
   Purpose: Collect information of the handover of a particular request.

6. New User Form:

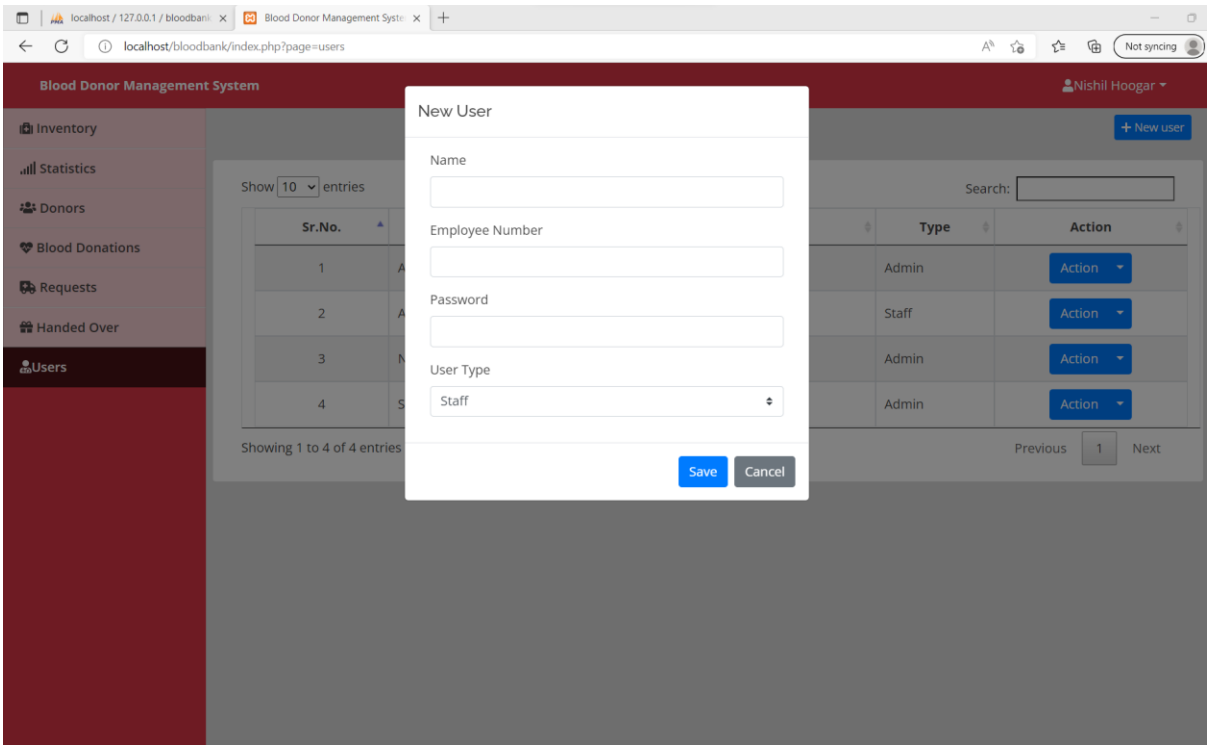


Figure 6

The New User Form (Figure 6) contains the following fields:

1. Name- Takes the full name of the user that is being added to the system.
2. Employee Number- Takes the Employee Number of the user that is being added to the system.

3. Password- Takes the password that shall be assigned to the particular employee that is being added to the system.
4. User Type- A drop down selection which contains the two types of user profiles for the proposed system , namely : Admin and Staff.

NOTE: A user with a Staff Profile does is not provided with the facility to add new users to the system.

Purpose: Collect information of a new user that is being added to the system.

7.Manage Account Form:

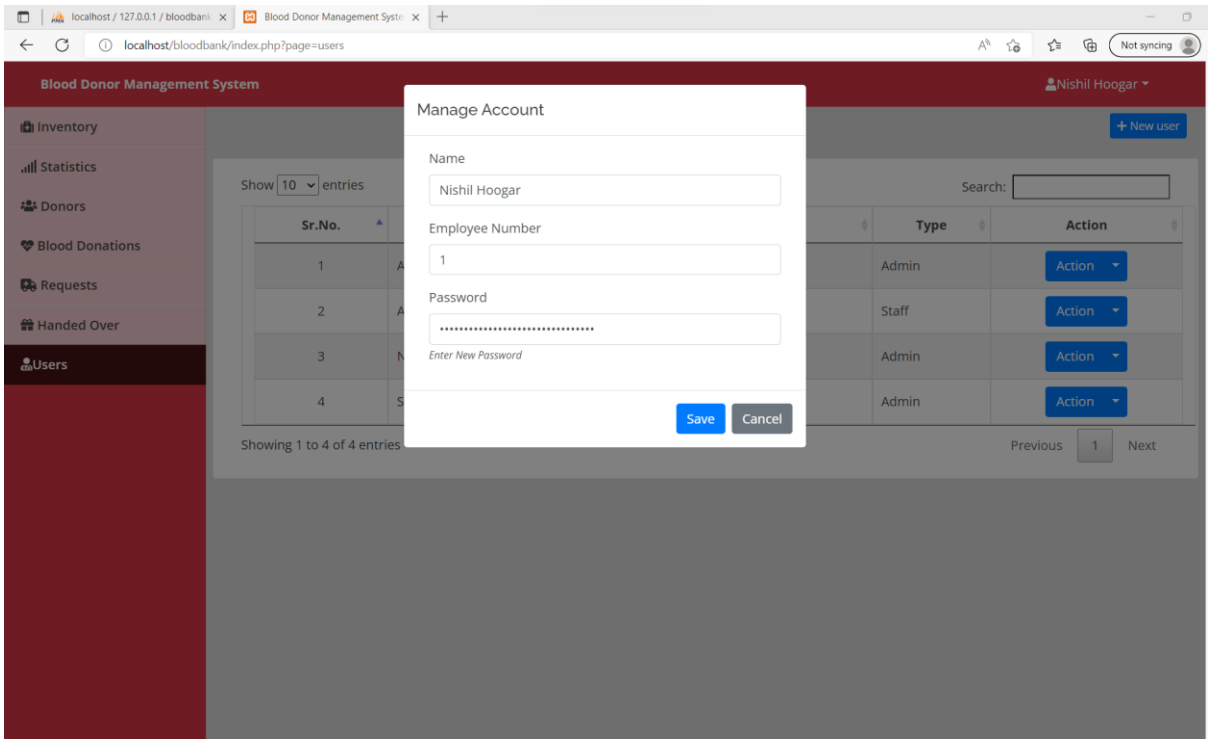


Figure 7

The Manage Account Form (Figure 7) contains the following fields:

1. Name- Name of the user that is currently logged in and wishes to change his password.
2. Employee Number – The employee number of the particular user.
3. Password- The new password that the user wishes to update.

Purpose- Update the password of a particular user who wishes to update his/her password.

DDL STATEMENTS:

CREATING TABLES WITH PRIMARY KEYS

1.

```
create table blood_inventory
        (
         id int(30) not null,
         blood_group varchar(10) not null,
         volume float not null,
         status tinyint(1) not null,
         donor_id int(30) not null,
         request_id int(30) not null,
         date_created datetime not null,
         primary key(id)
         );
```

2.

```
create table donors
        (
        id int(30) not null,
        blood_group varchar(10) not null,
        name text not null,
```

```
    address text not null,
    contact varchar(20) not null,
    email varchar(50) not null,
    date_created datetime not null,
    primary key (id)
    );
```

3.

```
create table handed_over
    (
    id int(30) not null,
    request_id int(30) not null,
    picked_up_by text not null,
    date_created datetime not null,
    primary key(id)
    );
```

4.

```
create table requests
    (
    id int(30) not null,
    ref_code varchar(20) not null,
    patient text not null,
    blood_group varchar(10) not null,
    volume float not null,
    physician_name text not null,
    status tinyint(1) not null,
    date_created datetime not null,
    primary key(id)
    );
```

5.

```
create table users
    (
    id int(30) not null,
    name text not null,
    username varchar(200) not null,
    password text not null,
    type tinyint(1) not null,
    primary key (id)
```

```
);
```

TRIGGERS

1.

```
{
create table donors_archives
(
        id int(30) PRIMARY KEY not null,
        blood_group varchar(10) not null,
        name text not null,
        address text not null,
        contact varchar(20) not null,
        email varchar(50) not null
);

DELIMITER $$

CREATE TRIGGER before_donors_delete
BEFORE DELETE
ON donors FOR EACH ROW
BEGIN
        INSERT INTO doners_archives(id,blood_group,name,address,
contact,email)
        VALUES(OLD.id,OLD.blood_group,OLD.name,OLD.address,OLD.contact,
OLD.email);
END $$

DELIMITER ;
}
```

2.

```
{
        create table handed_over_archives
        (
                id int(30) PRIMARY KEY not null,
                request_id int(30) not null,
                picked_up_by text not null
```

```
);
DELIMITER $$

CREATE TRIGGER before_handed_over_delete
BEFORE DELETE
ON handed_over FOR EACH ROW
BEGIN
    INSERT INTO
handed_over_archives(id,request_id,picked_up_by)
    VALUES(OLD.id,OLD.request_id,OLD.picked_up_by);
END $$

DELIMITER ;


}

3.
{
    create table requests_archives
    (
        id int(30) PRIMARY KEY not null,
        ref_code varchar(20) not null,
        patient text not null,
        blood_group varchar(10) not null,
        volume float not null,
        physician_name text not null,
        status tinyint(1) not null
    );

    DELIMITER $$

    CREATE TRIGGER before_request_delete
    BEFORE DELETE
    ON requests FOR EACH ROW
    BEGIN
        INSERT INTO
    requets_archives(id,ref_code,patient,blood_group,volume,physicia
    n_name,status)
        VALUES(OLD.id,OLD.ref_code,OLD.patient,OLD.blood_group,OLD.
    volume,OLD.physician_name,OLD.status);
    END $$
```

```
	DELIMITER ;


}


4.
{
create table system_settings_archives
(
	id int(30) PRIMARY KEY not null,
	name text NOT null,
	email varchar(200) not null,
	contact varchar(20) not null,
	cover_img text not null,
	about_contact text not null
);

DELIMITER $$

CREATE TRIGGER before_system_settings_delete
BEFORE DELETE
ON system_settings FOR EACH ROW
BEGIN
	INSERT INTO
system_settings_archives(id,name,email,contact,cover_img,about_conta
ct)
	VALUES(OLD.id,OLD.name,OLD.email,OLD.contact,OLD.cover_img,OLD.
about_contact);
END $$

DELIMITER ;


}

5.
{

create table users_archives
(
	id int(30) PRIMARY KEY not null,
	name text not null,
```

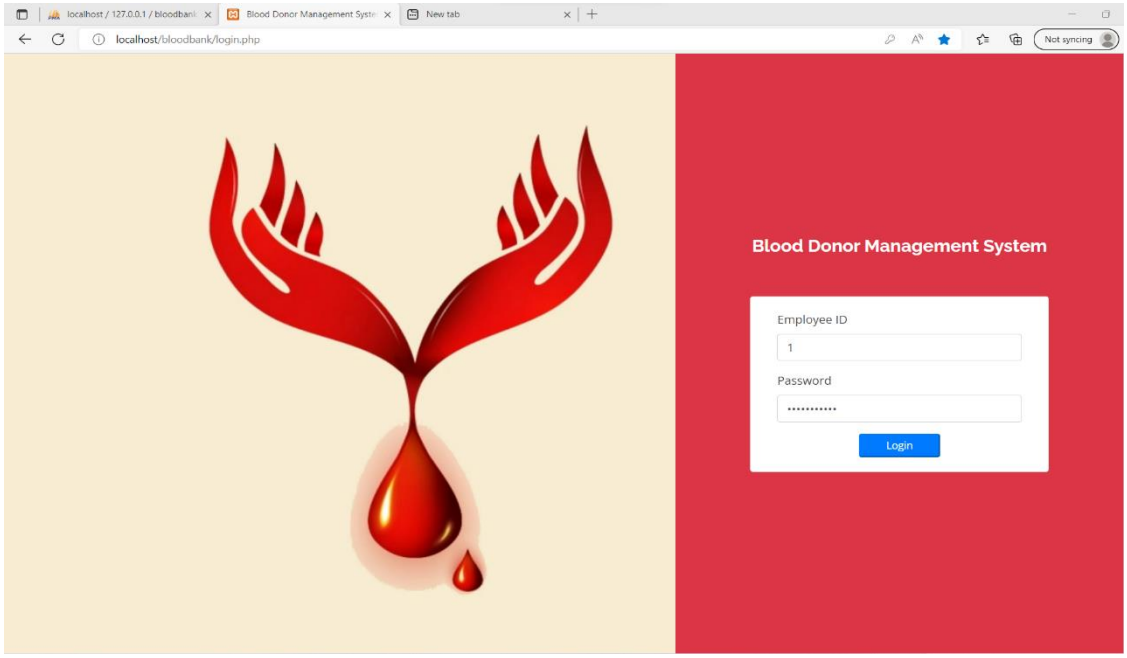
```
        username varchar(200) not null,
        password text not null,
        type tinyint(1) not null
);

DELIMITER $$

CREATE TRIGGER before_users_delete
BEFORE DELETE
ON users FOR EACH ROW
BEGIN
        INSERT INTO users_archives(id,name,username,password,type)
        VALUES(OLD.id,OLD.name,OLD.username,OLD.password,OLD.type);
END $$

DELIMITER ;
}
```

Views

```
CREATE VIEW max_request_per_bloodgroup AS
SELECT p.blood_group, MAX(p.volume) as volume
FROM requests p INNER JOIN
   (SELECT blood_group, MAX(volume) AS max_volumr
    FROM requests
    GROUP BY blood_group
   ) p2
  ON p.blood_group = p2.blood_group AND p.volume = p2.max_volume GROUP BY p.blood_group;
```

SYSTEM:

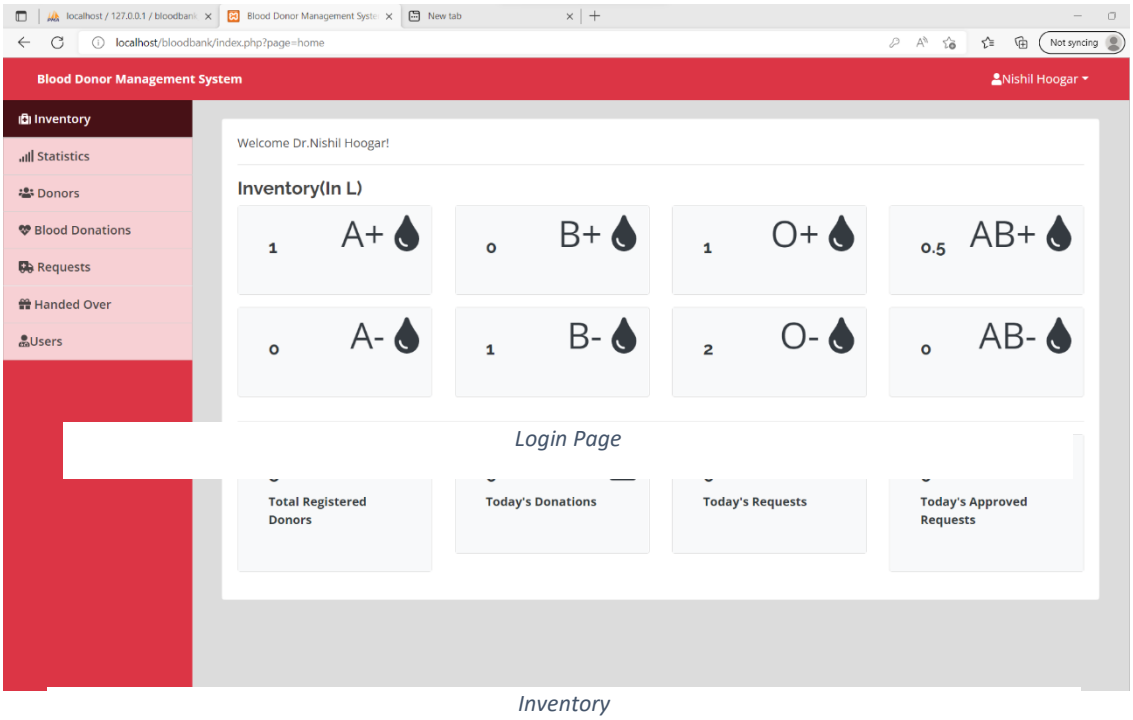
1. LOGIN PAGE:



User can log in with the appropriate credentials.

INVENTORY PAGE:

Inventory displays the availability of all blood groups.

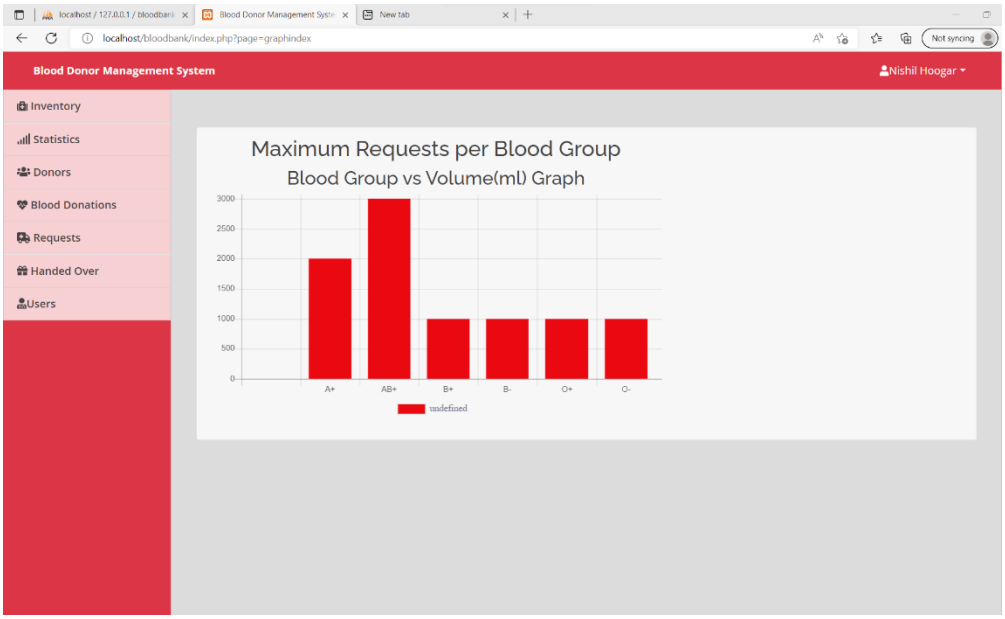


Login Page

Inventory

2. Statistics Page

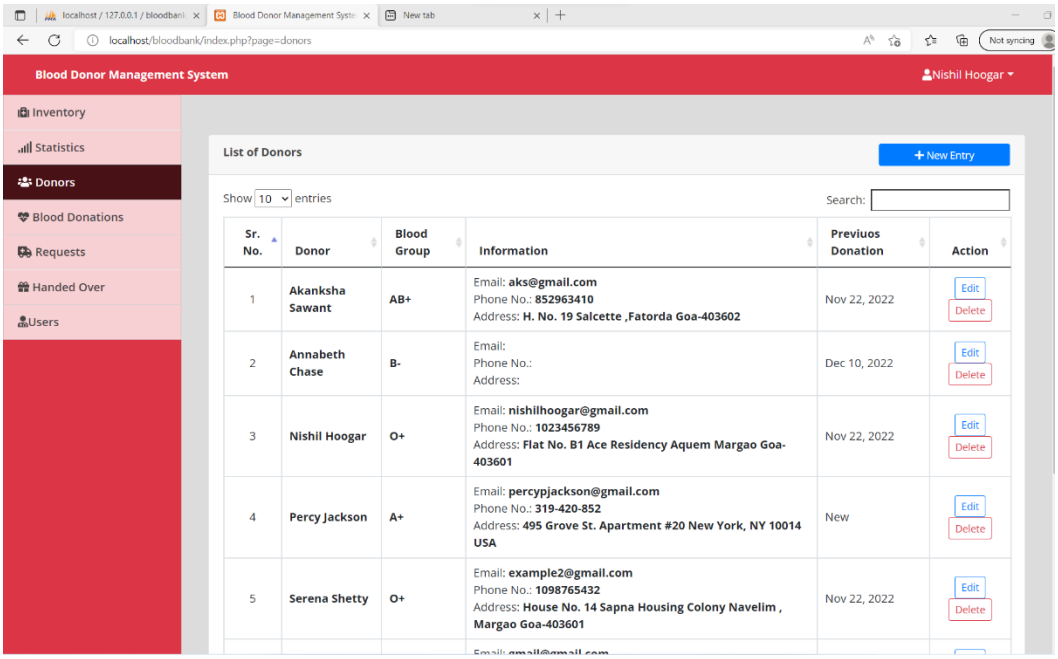
   Displays statistics in the form of bar graph.



Analytics

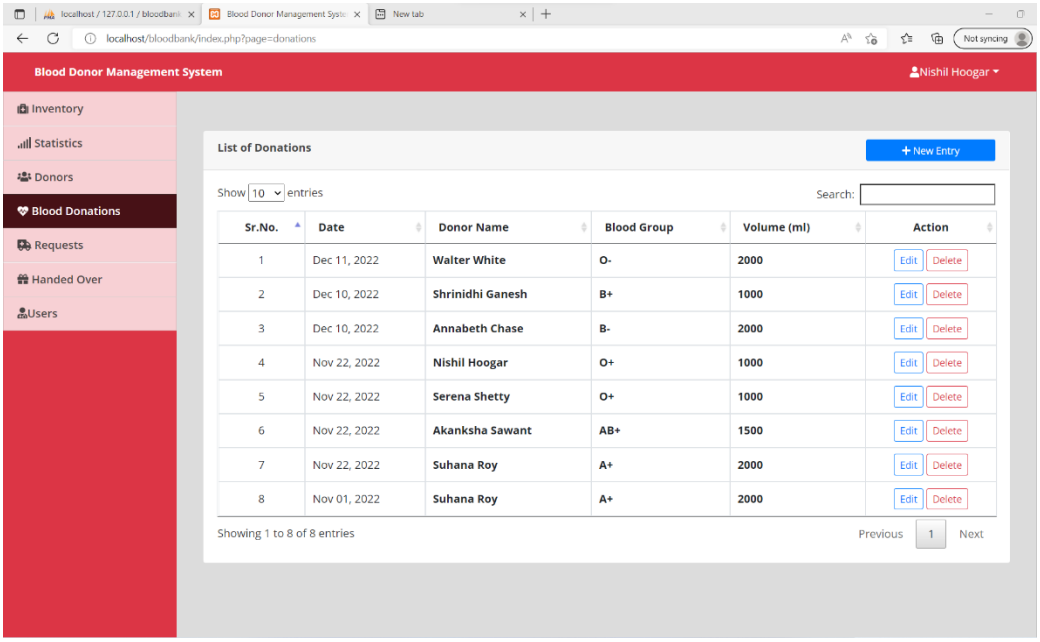
3. List of Donors

   Displays list of registered donors.



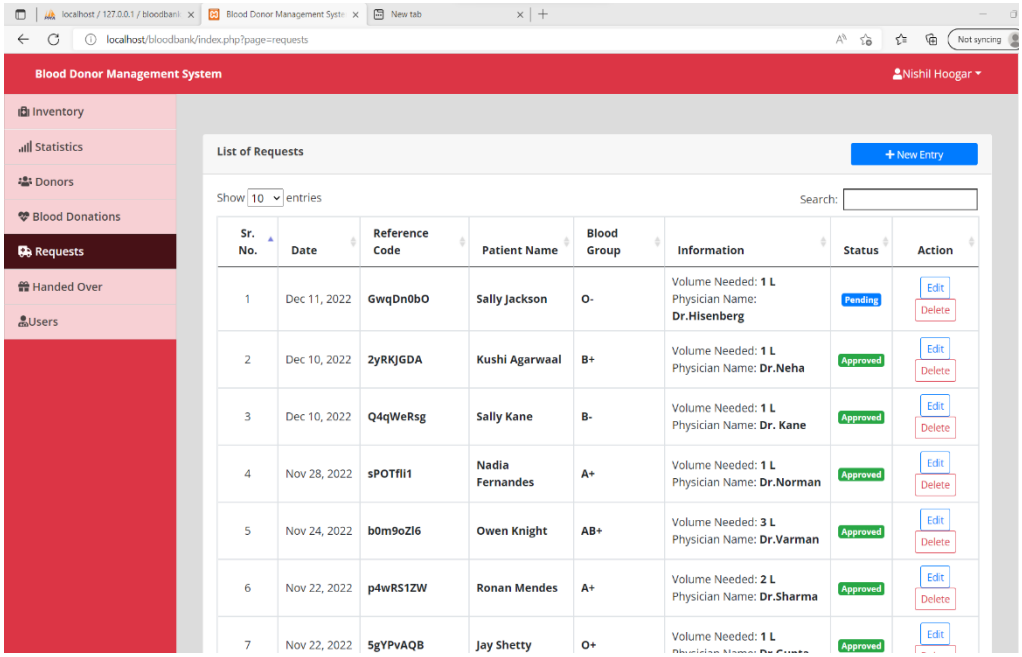
List of Donors

4. List of Donations:
   Displays a list of donations which have been implemented



List of Donations

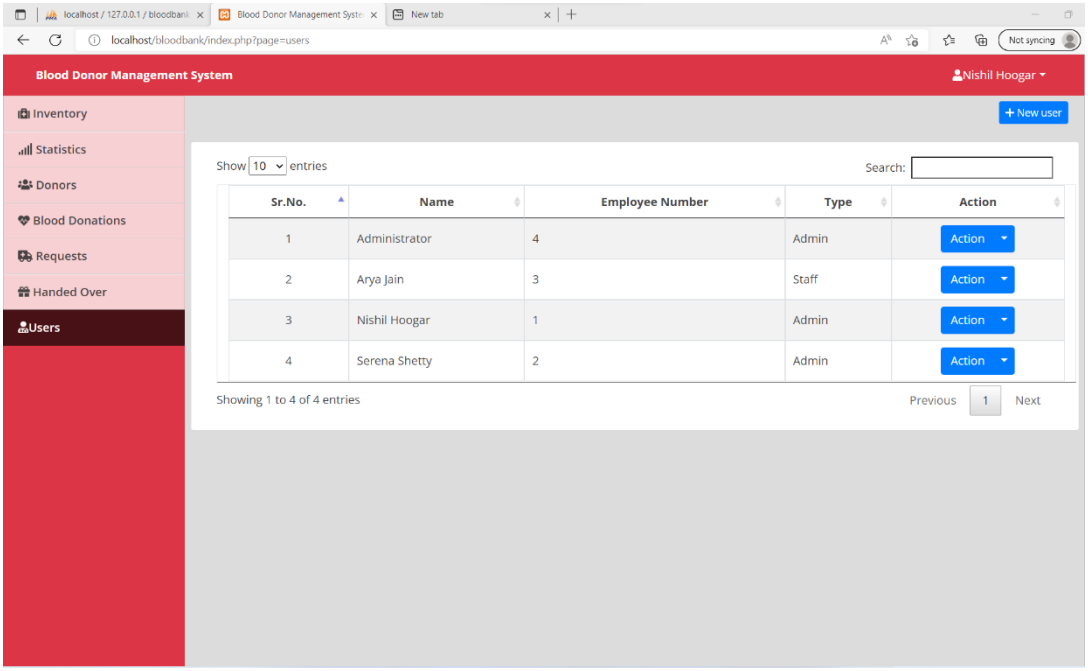
5. List of Requests
   Displays list of requests in the system



List of Requests

6. List of users

   Displays list of users that can access the system



List of users