****

**IBM CAREER EDUCATION**

**MAIN PROJECT**

**DOMAIN NAME: JAVA**

**Blood Bank Management System**

**Submitted By,**

* Tanvi Trivedi (18162171031)
* Meshva Gupta (181621710)
* Jhanvi Zala(18162171035)

II Year – ( CS ) ‘B’ Section

Ganpat University, Ahmedabad.

**Submitted To,**

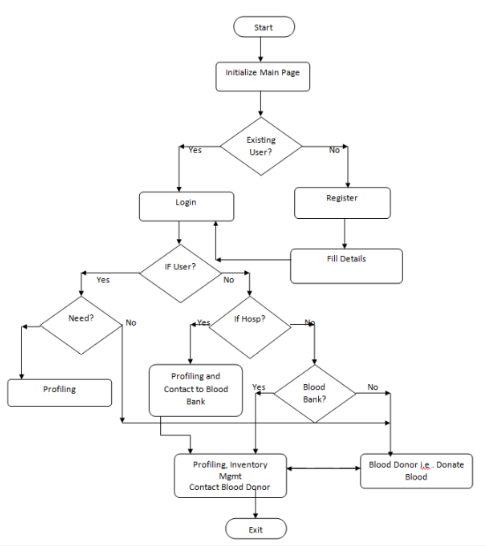
A. Saai Sanjeev Achaarya

IBM Software Technical Trainer

****

**FLOWCHART**

* System Architecture



* Flowchart of our program

START

Choice(ch)

ch=3?

ch=2?

ch=1?

ch=4?

Input donor

details

Show Blood

Stock

details

Input petient

details and

Show it

Input order details and show it

Show and Add it to stock

Extends to Blood cell

Return patient

ch again as input

ch=1,2,3

or 4?

STOP

NO

NO

NO

**SOFTWARE SPECIFICATIONS**

* OPERATING SYSTEM : Linux / Windows / IOS
* ENVIRONMENT : IBM RAD Software

**HARDWARE SPECIFICATIONS**

|  |  |  |
| --- | --- | --- |
|  PROCESSOR | : | PENTIUM IV 2.8MHz |
|  RAM | : | 256 MB SD RAM |
|  MONITOR | : | 15” COLOR |
|  HARD DISK | : | 40 GB |
|  FLOPPY DRIVE | : | 1.44 MB |
|  |  |  |

**LANGUAGES USED:**

|  |  |  |
| --- | --- | --- |
|  JAVA | : | For main coding |
|  HTML | : | For Webpage |
|  CSS | : | For Webpage |
|  JAVASCRIPT | : | For Webpage |
|  JQUERY | : | For Webpage |
|  |  |  |

**DESCRIPTION**

**INTRODUCTION:**

The BLOOD BANK MANAGEMENT SYSTEM is great project. This project is designed for successful completion of project on blood bank management system. Blood Bank Management System (BBMS) is a Java Project that is designed to store, process, retrieve and analyze information concerned with the administrative and inventory management within a blood bank. This project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way. Project Aim is to provide transparency in this field, make the process of obtaining blood from a blood bank hassle free and corruption free and make the system of blood bank management effective.

**DESCRIPTION:**

Purpose:

* The proposed system (Blood Bank Management System) is designed to help the Blood Bank administrator to meet the demand of Blood by sending and/or serving the request for Blood as and when required. The proposed system gives the procedural approach of how to bridge the gap between Recipient, Donor, and Blood Banks. This Application will provide a common ground for all the three parties (i.e. Recipient, Donor, and Blood Banks) and will ensure the fulfillment of demand for Blood requested by Recipient and/or Blood Bank

Goals:

* To ease the process of blood donation and reception.
* To improve the existing system.
* To develop a scalable system.
* To be highly available.

Scope:

* Ensure that all the functionalities of a manual blood bank are covered.
* To include all the blood banks at least within a city.
* Make sure the program is simple and easy to use.

**PROS:**

* **Donors**: person who wants to donate the blood voluntarily at the blood donation camp.
* **Seekers**: person who wants the blood from the blood bank due to various reasons like accidents, surgeries, delivery and many more.
* **Blood bank**: staff people which are working in the blood bank which includes staff member, operator, blood bank in charge, head of pathological department.

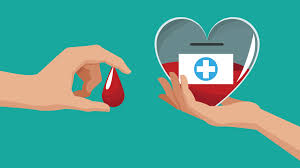
**CONS:**

* Wastage of blood due to expiry.

**BLOOD BANK MANAGEMENT SYSTEM**

**PICTURES**



**AIM:**

* The **aim** of this socially relevant project is to manage the donation and storing in blood bank so people can contact easily.

**PROGRAM:**

* **Java code**

//Importing all the relevant packages and their classes

import java.io.\*;

import java.util.\*;

class blood\_stock

{

//Method to show blood stock

static int qua1=0,qua2=0,qua3=0,qua4=0,qua5=0,qua6=0,qua7=0,qua8=0;

void blood\_stockk()

{

System.out.println("\nBlood Stocks");

System.out.println("Blood Group : A+ \nStock :" + qua1 + " ml");

System.out.println("Blood Group : A- \nStock :" + qua2 + " ml");

System.out.println("Blood Group : B+ \nStock :" + qua3 + " ml");

System.out.println("Blood Group : B- \nStock :" + qua4 + " ml");

System.out.println("Blood Group : O+ \nStock :" + qua5 + " ml");

System.out.println("Blood Group : O- \nStock :" + qua6 + " ml");

System.out.println("Blood Group : AB+ \nStock :" + qua7 + " ml");

System.out.println("Blood Group : AB- \nStock :" + qua8 + " ml");

}

}

class blood\_cell extends blood\_stock

{

void blood\_stockk()

{

System.out.println("\nBlood Stocks");

System.out.println("Blood Group : A+ \nStock :" + (qua1\*5000) + " units");

System.out.println("Blood Group : A- \nStock :" + (qua2\*5000) + " units");

System.out.println("Blood Group : B+ \nStock :" + (qua3\*5000) + " units");

System.out.println("Blood Group : B- \nStock :" + (qua4\*5000) + " units");

System.out.println("Blood Group : O+ \nStock :" + (qua5\*5000) + " units");

System.out.println("Blood Group : O- \nStock :" + (qua6\*5000) + " units");

System.out.println("Blood Group : AB+ \nStock :" + (qua7\*5000) + " units");

System.out.println("Blood Group : AB- \nStock :" + (qua8\*5000) + " units");

}

}

class patient

{

//Declaring variables.

String p\_name,pblg,padd,rbg4,psex;

int quan,i,pid;

//Method to get patient details

void new\_patient()

{

Scanner input = new Scanner(System.in);

System.out.print("Patient id : ");

pid = input.nextInt();

System.out.print("Patient name : ");

p\_name = input.next();

System.out.print("Address : ");

padd = input.next();

do{

System.out.print("Gender : ");

psex = input.next();

if(!(psex.equals("F") ||psex.equals("M")||psex.equals("f")||psex.equals("m")))

{

System.out.println("Wrong input please try again");

}

else

{

break;

}

}while(true);

do{

System.out.print("Blood Group of patient : ");

pblg = input.next();

if(!(pblg.equals("A+")||pblg.equals("B+")||pblg.equals("AB+")||pblg.equals("O+")||pblg.equals("A-")||pblg.equals("B-")||pblg.equals("AB-")||pblg.equals("O-")||pblg.equals("a+")||pblg.equals("b+")||pblg.equals("ab+")||pblg.equals("o+")||pblg.equals("a-")||pblg.equals("b-")||pblg.equals("ab-")||pblg.equals("o-")))

{

System.out.println("Input you entered is not a blood group. Please try again");

}

else

{

break;

}

}while(true);}

//Method to show patient's details

void patient\_info()

{

System.out.println("ID" + "\t" + "Name" + "\t" + "Gender" + "\t" + "Address" + "\t" + "Blg\_grp");

System.out.println(pid + "\t" + p\_name + "\t" + psex + "\t" + padd + "\t" + pblg);

}

}

class donor extends blood\_stock

{

//Declaring variables

String don\_name,en\_date,bg5,hc\_name,dsex;

int id\_don,qua;

//Method to get donor's details

void new\_donor()

{

Scanner input = new Scanner(System.in);

System.out.print("Donor id : ");

id\_don = input.nextInt();

System.out.print("Donor name : ");

don\_name = input.next();

System.out.print("Date of Donation : ");

en\_date = input.next();

do{

System.out.print("Gender : ");

dsex = input.next();

if(!(dsex.equals("F") ||dsex.equals("M")||dsex.equals("f")||dsex.equals("m")))

{

System.out.println("Wrong input please try again");

}

else

{

break;

}

}while(true);

do{

System.out.print("Blood Group of donor : ");

bg5 = input.next();

if(!(bg5.equals("A+")||bg5.equals("B+")||bg5.equals("AB+")||bg5.equals("O+")||bg5.equals("A-")||bg5.equals("B-")||bg5.equals("AB-")||bg5.equals("O-")||bg5.equals("a+")||bg5.equals("b+")||bg5.equals("ab+")||bg5.equals("o+")||bg5.equals("a-")||bg5.equals("b-")||bg5.equals("ab-")||bg5.equals("o-")))

{

System.out.println("Input you entered is not a blood group. Please try again");

}

else

{

break;

}

}while(true);

System.out.print("Quantity of Blood donated : ");

qua = input.nextInt();

if(bg5.equals("A+") || bg5.equals("a+"))

qua1 += qua;

else if(bg5.equals("A-") || bg5.equals("a-"))

qua2 += qua;

else if(bg5.equals("B+") || bg5.equals("b+"))

qua3 += qua;

else if(bg5.equals("B-") || bg5.equals("b-"))

qua4 += qua;

else if(bg5.equals("O+") || bg5.equals("o+"))

qua5 += qua;

else if(bg5.equals("O-") || bg5.equals("o-"))

qua6 += qua;

else if(bg5.equals("AB+") || bg5.equals("ab+"))

qua7 += qua;

else if(bg5.equals("AB-") || bg5.equals("ab-"))

qua8 += qua;

System.out.print("Hospital or camp name : ");

hc\_name = input.next();

}

//Method to show donor's details

void donor\_info()

{

System.out.println("ID" + "\t" + "Name" + "\t" + "Gender" + "\t" + "Entry\_date" + "\t" + "Bld\_grp" + "\t" + " Quantity" +"\t" + "Hospital\_name");

System.out.println(id\_don + "\t" + don\_name + "\t" + dsex + "\t" + en\_date + "\t\t" + bg5 + "\t\t" + qua5 +"\t" + hc\_name);

}

}

class order extends blood\_stock

{

//Method to show order details

void orderr()

{

String p\_name,pblg,date3,avl;

int quan,eml3;

Scanner input = new Scanner(System.in);

System.out.print("Patient name : ");

p\_name = input.next();

do{

System.out.print("Ordered blood group : ");

pblg = input.next();

if(!(pblg.equals("A+")||pblg.equals("B+")||pblg.equals("AB+")||pblg.equals("O+")||pblg.equals("A-")||pblg.equals("B-")||pblg.equals("AB-")||pblg.equals("O-")||pblg.equals("a+")||pblg.equals("b+")||pblg.equals("ab+")||pblg.equals("o+")||pblg.equals("a-")||pblg.equals("b-")||pblg.equals("ab-")||pblg.equals("o-")))

{

System.out.println("Input you entered is not a blood group. Please try again");

}

else

{

System.out.print("Quantity of Blood Group : ");

quan = input.nextInt();

if(pblg.equals("A+") || pblg.equals("a+"))

{

if(quan>qua1)

{

System.out.println("Sorry the quantity exceeds the stock.");

}

else

{

qua1 -= quan;

break;

}

}

else if(pblg.equals("A-") || pblg.equals("a-"))

{

if(quan>qua2)

{

System.out.println("Sorry the quantity exceeds the stock.");

}

else

{

qua2 -= quan;

break;

}

}

else if(pblg.equals("B+") || pblg.equals("b+"))

{

if(quan>qua3)

{

System.out.println("Sorry the quantity exceeds the stock.");

}

else

{

qua3 -= quan;

break;

}

}

else if(pblg.equals("B-") || pblg.equals("b-"))

{

if(quan>qua4)

{

System.out.println("Sorry the quantity exceeds the stock.");

}

else

{

qua4 -= quan;

break;

}

}

else if(pblg.equals("O+") || pblg.equals("o+"))

{

if(quan>qua5)

{

System.out.println("Sorry the quantity exceeds the stock.");

}

else

{

qua5 -= quan;

break;

}

}

else if(pblg.equals("O-") || pblg.equals("o-"))

{

if(quan>qua6)

{

System.out.println("Sorry the quantity exceeds the stock.");

}

else

{

qua6 -= quan;

break;

}

}

else if(pblg.equals("AB+") || pblg.equals("ab+"))

{

if(quan>qua7)

{

System.out.println("Sorry the quantity exceeds the stock.");

}

else

{

qua7 -= quan;

break;

}

}

else if(pblg.equals("AB-") || pblg.equals("ab-"))

{

if(quan>qua8)

{

System.out.println("Sorry the quantity exceeds the stock.");

}

else

{

qua8 -= quan;

break;

}

}

// break;

}

}while(true);

System.out.print("Emergency level(1 to 5) : ");

eml3 = input.nextInt();

}

}

class total

{

public static void main(String[] args)

{

Scanner input=new Scanner(System.in);

patient[] p=new patient[25];

donor d=new donor();

blood\_stock bd=new blood\_stock();

blood\_cell bc=new blood\_cell();

order o=new order();

int choice,c1,s1,count1=0,j=0,status=1;

s:

while(status==1)

{

System.out.println("\n MAIN MENU");

System.out.println("--------------------------------------------------------------------------------");

System.out.println("1.DONOR 2. BLOOD STOCK 3.PATIENT 4.ORDER 5.BLOOD CELL 6.EXIT");

System.out.println("--------------------------------------------------------------------------------");

System.out.print(" Enter choice : ");

choice = input.nextInt();

switch (choice)

{

case 3:

{

System.out.println("--------------------------------------------------------------------------------");

System.out.println(" \*\*PATIENT SECTION\*\*");

System.out.println("--------------------------------------------------------------------------------");

s1=1;

while(s1==1)

{

System.out.println("\n MAIN MENU");

System.out.println("--------------------------------------------------------------------------------");

System.out.println("1.Add a new data 2.Display 3.EXIT");

System.out.println("--------------------------------------------------------------------------------");

c1=input.nextInt();

//switch(c1)

if(c1==1)

{

//case 1:

p[count1] = new patient();

p[count1].new\_patient();

count1++;

//break;

}

//case 2:

else if(c1==2)

{

for(j=0;j<count1;j++)

{

System.out.println("The patient's details : ");

System.out.println();

p[j].patient\_info();

}

//break;

}

else if(c1==3)

{

System.out.println("Return back press (1/0) for more");

s1=input.nextInt();

}

else

{

System.out.print("default");

}

}

break;

}

case 1:

System.out.println("--------------------------------------------------------------------------------");

System.out.println(" \*\*DONOR SECTION\*\*");

System.out.println("--------------------------------------------------------------------------------");

d.new\_donor();

System.out.println("The donor's details : ");

d.donor\_info();

break;

case 2:

System.out.println("--------------------------------------------------------------------------------");

System.out.println(" \*\*BLOOD\_STOCK SECTION\*\*");

System.out.println("--------------------------------------------------------------------------------");

bd.blood\_stockk();

break;

case 4:

System.out.println("--------------------------------------------------------------------------------");

System.out.println(" \*\*ORDER SECTION\*\*");

System.out.println("--------------------------------------------------------------------------------");

o.orderr();

break;

case 5:

System.out.println("--------------------------------------------------------------------------------");

System.out.println(" \*\*BLOOD\_CELL SECTION\*\*");

System.out.println("--------------------------------------------------------------------------------");

bc.blood\_stockk();

break;

case 6: break s;

}

}

}

}

* + **Html code (including .html, .css, .js files)**

1. **Home page**

****

1. **Sign\_up Page**

****

1. **Admin Page**

****

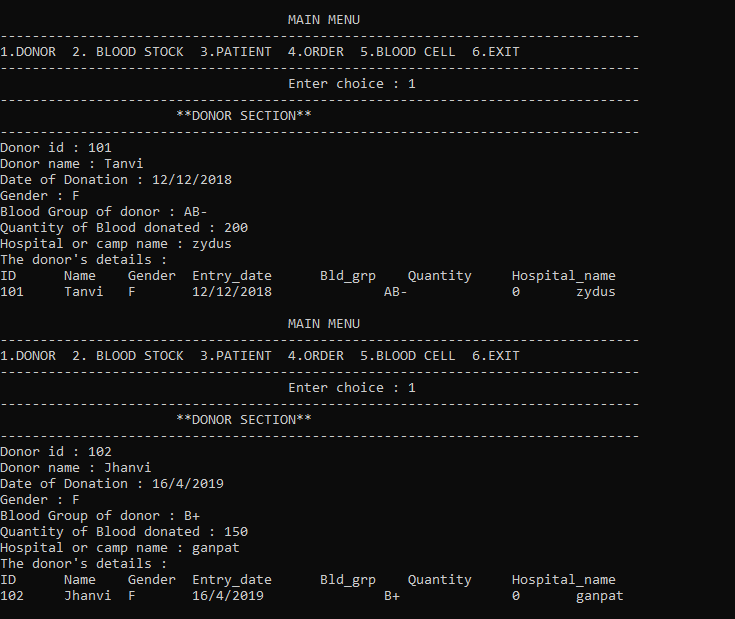
**EXPLANATION ABOUT PROJECT:**

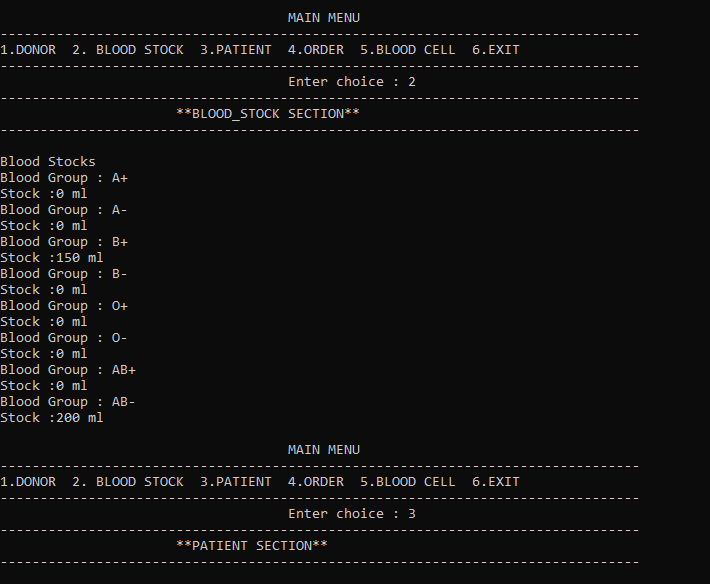
* The project of bloodbank management system is based on JAVA domain here, first of all, We declare main header Files and then we declare classes like blood\_stock, blood\_cells , patient, donor, order,total where the main method is in class total.
* In first class blood\_Stock returns the stock based on the donor’s donation and patient’s order for that to be done, here we declare variables and exteds it to the blood\_cell to retun the units of blood (i.e., 1ml = 5000units of blood cell).
* Then in patient class the variables are declared taken input for new entry of patient. There is a *validation that in patient gender, you can only add F or M else it will return “Wrong input please try again”* and then break.Again using do while loop, validation is for only 8 blood group you can enter.*Random entry for blood group is not possible otherwise it will return “Input you entered is not a blood group. Please try again" (i.e., validation for bloodgroup is used).*
* Class donor, which also extends blood\_stock, so if we are entering in donor, then the which blood group he/she donates, will directly goes to blood\_Stock in to that blood\_group and also goes to blood\_cell and multiply by 5000 units( for returning the unit of blood cells).*There is also validation in gender and blood group*.

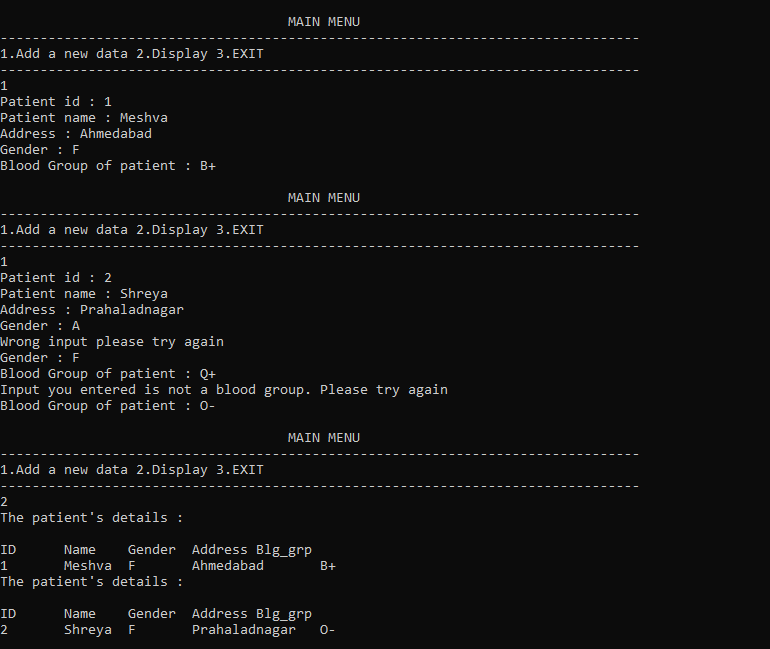
* Class order extends blood\_stock , *if ordered blood group quantity is less than the blood\_stock then it will execute the other conditions otherwise if the quantity is greater than the required blood group it will print “Sorry the quantity exceeds the stock”.* Then it will ask you emergency level of order out of 5.
* Main method is declared in class total using while loop and in that we have used switch case for main menu and for other sub menus.
* There is a option of exit and it ask you that you want exit only this or directly go to main menu.
* This was the briefing of our java code.

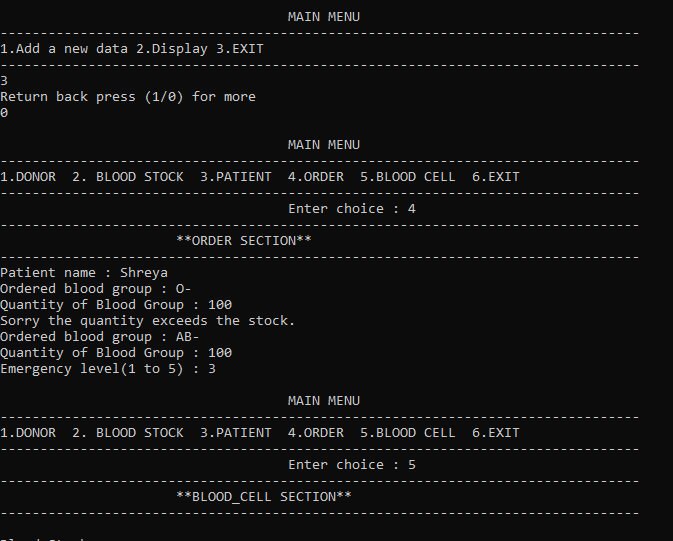
**OUTPUT SCREENSHOTS:**

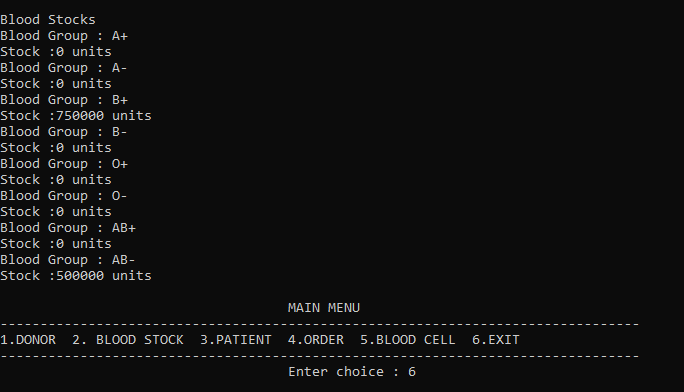
* **Java code output**









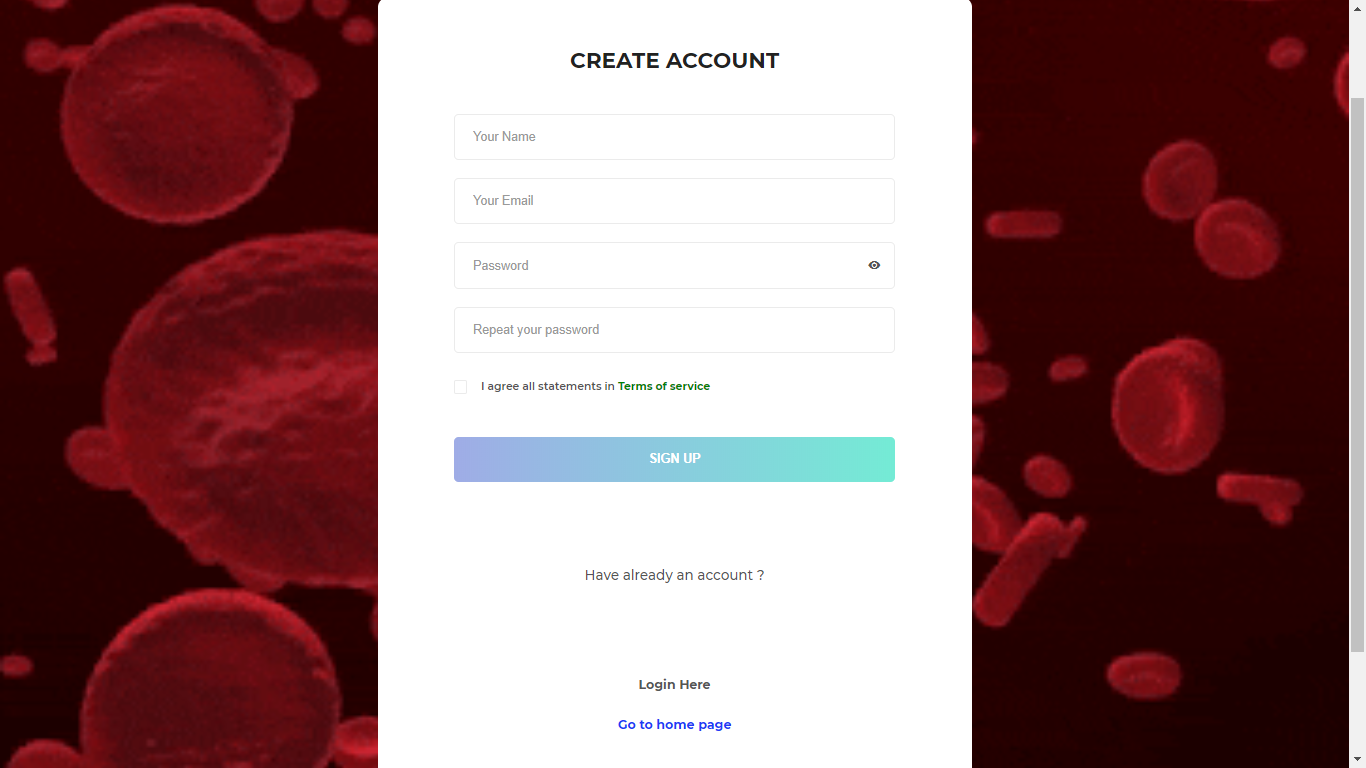


Here it exits the execution and returns back to the terminal.

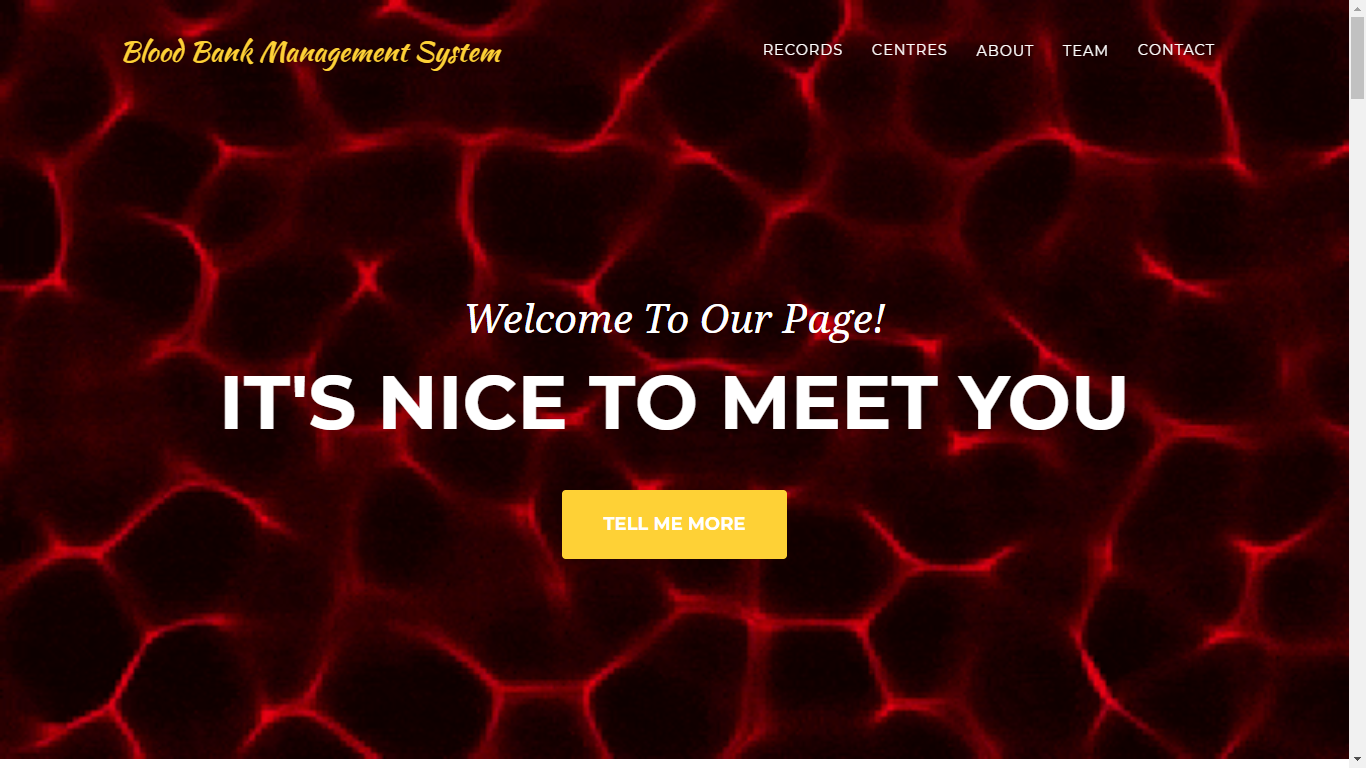
* **Home interface**

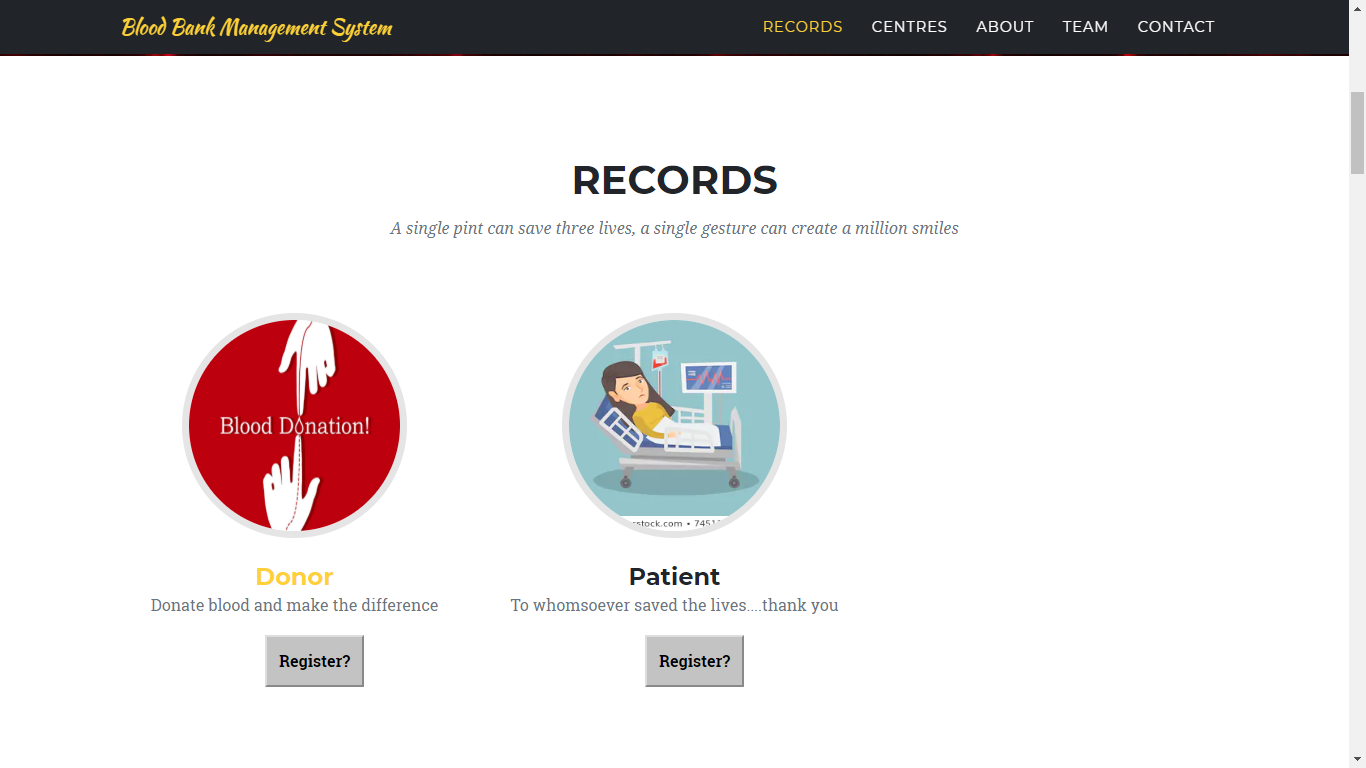


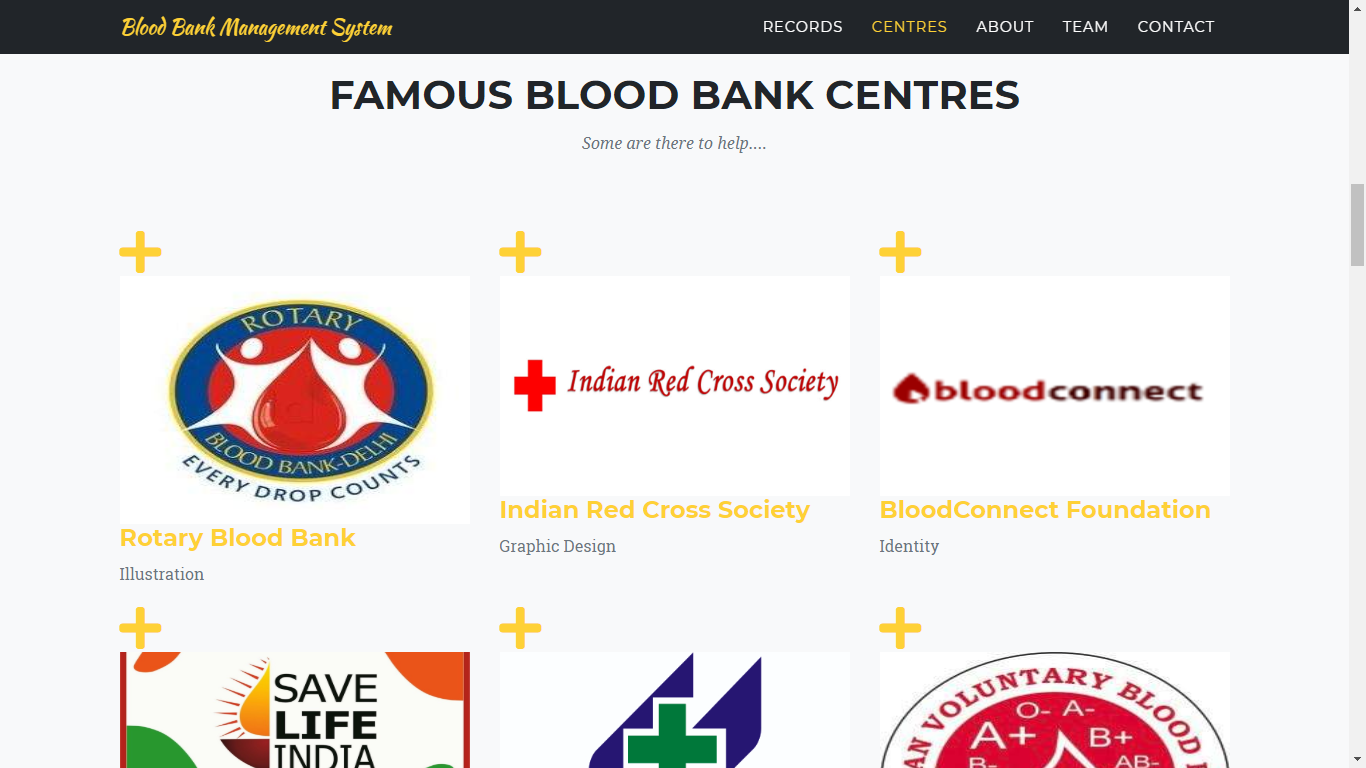
* **Sign\_up interface**

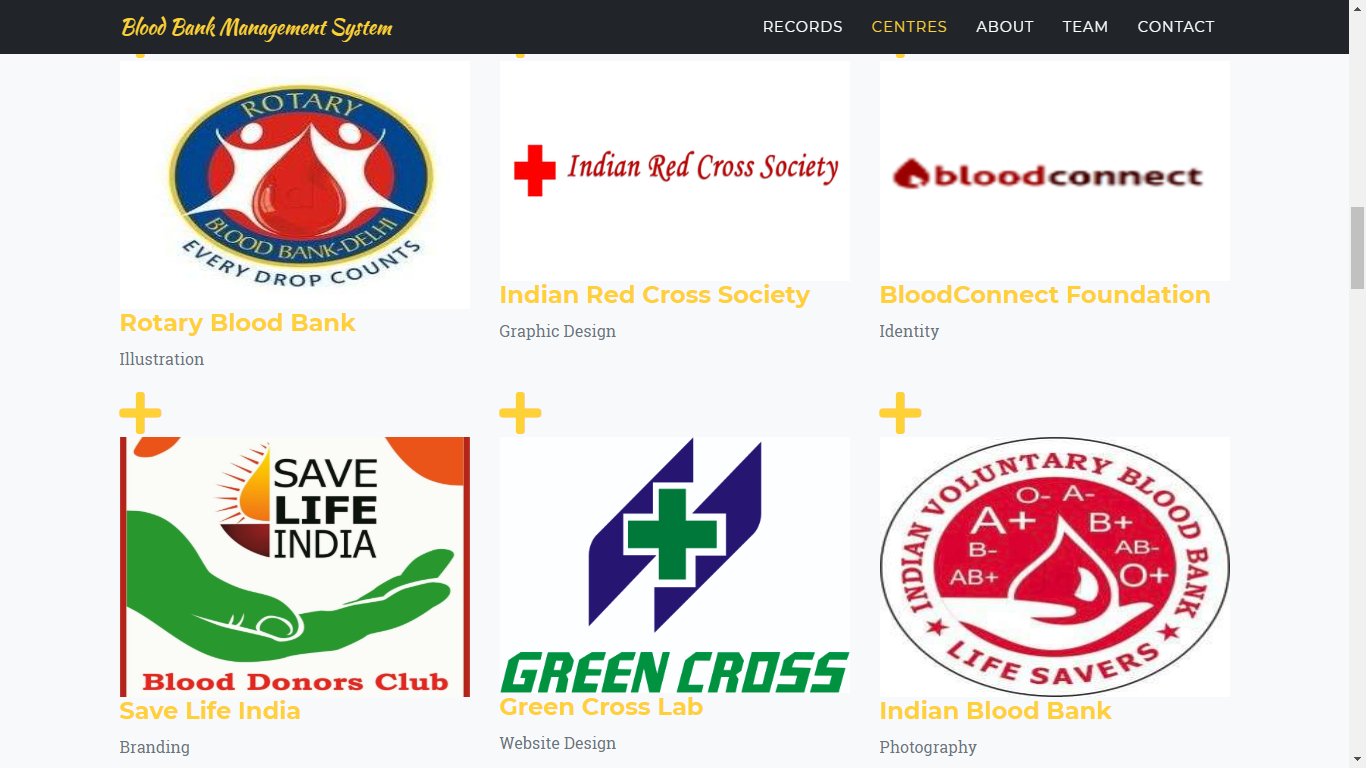


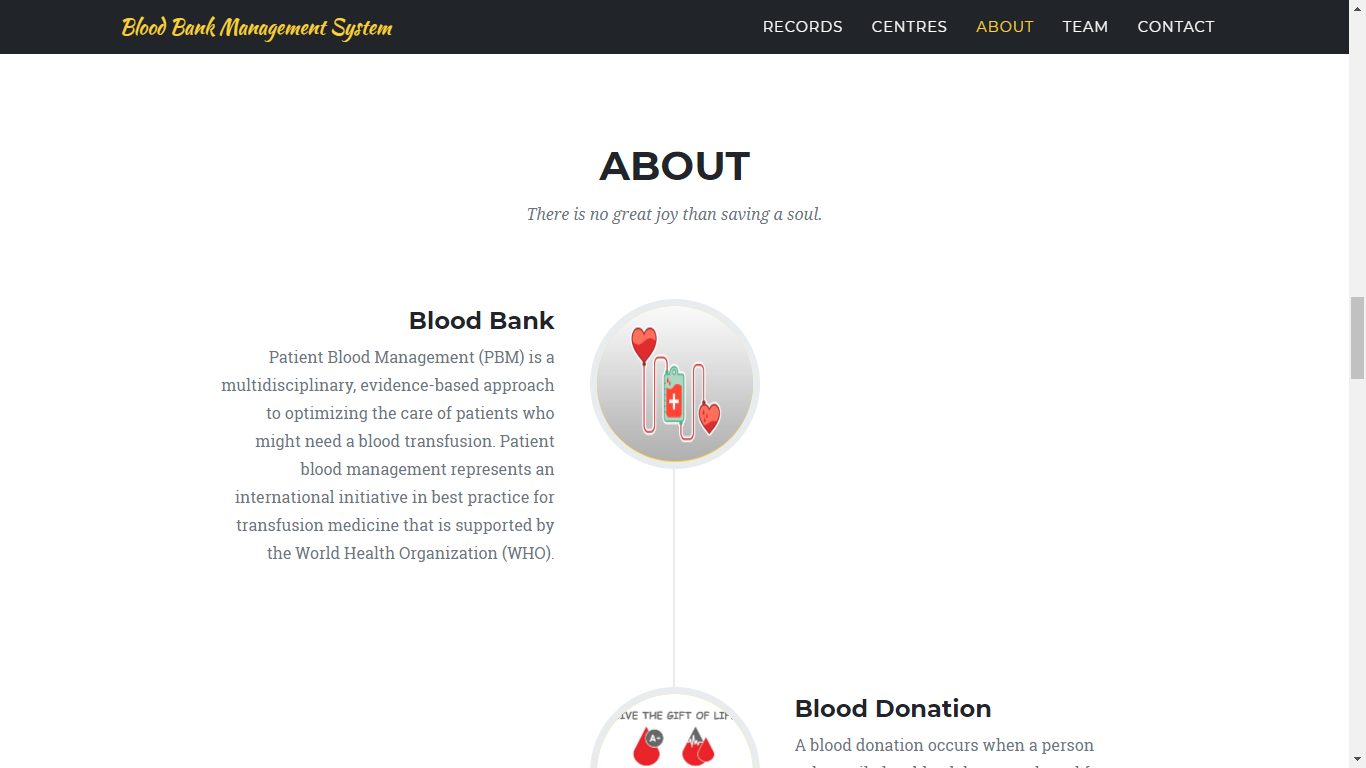
* **Admin interface**

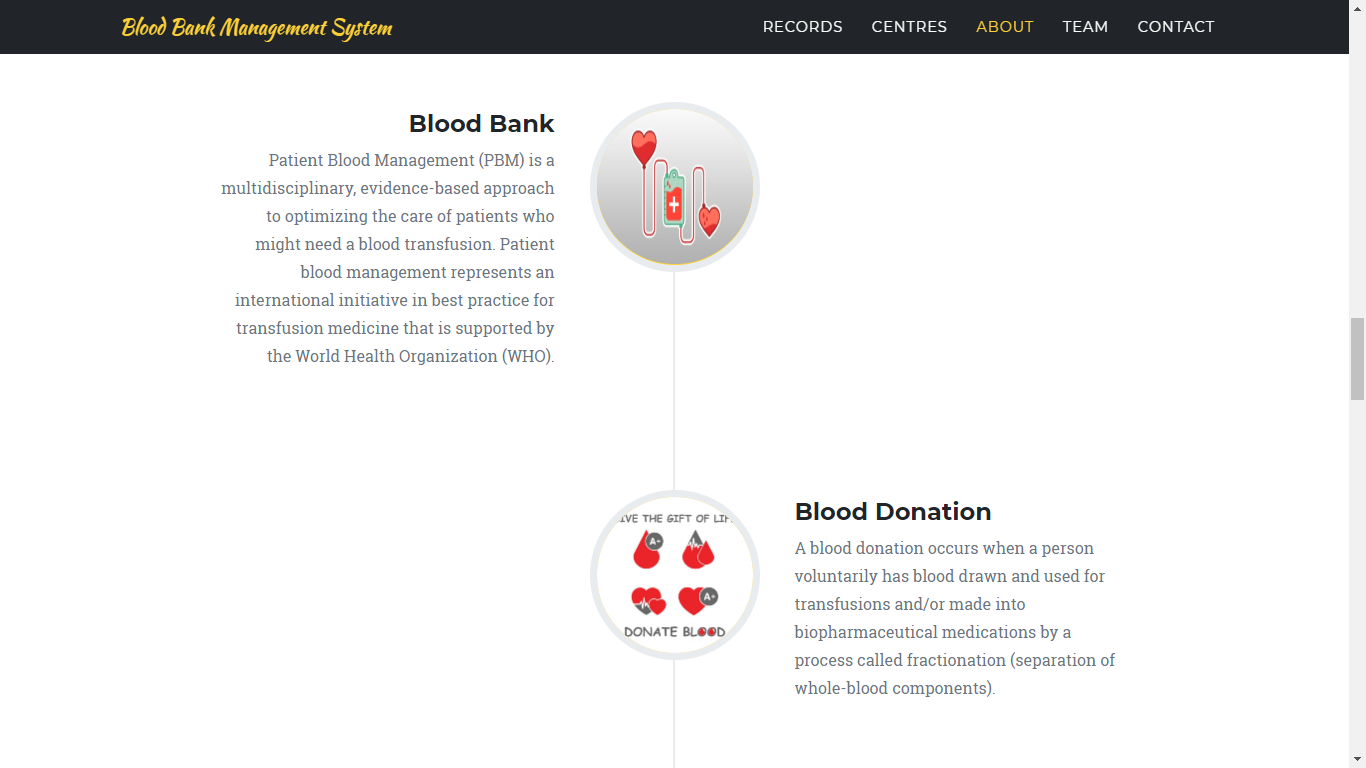


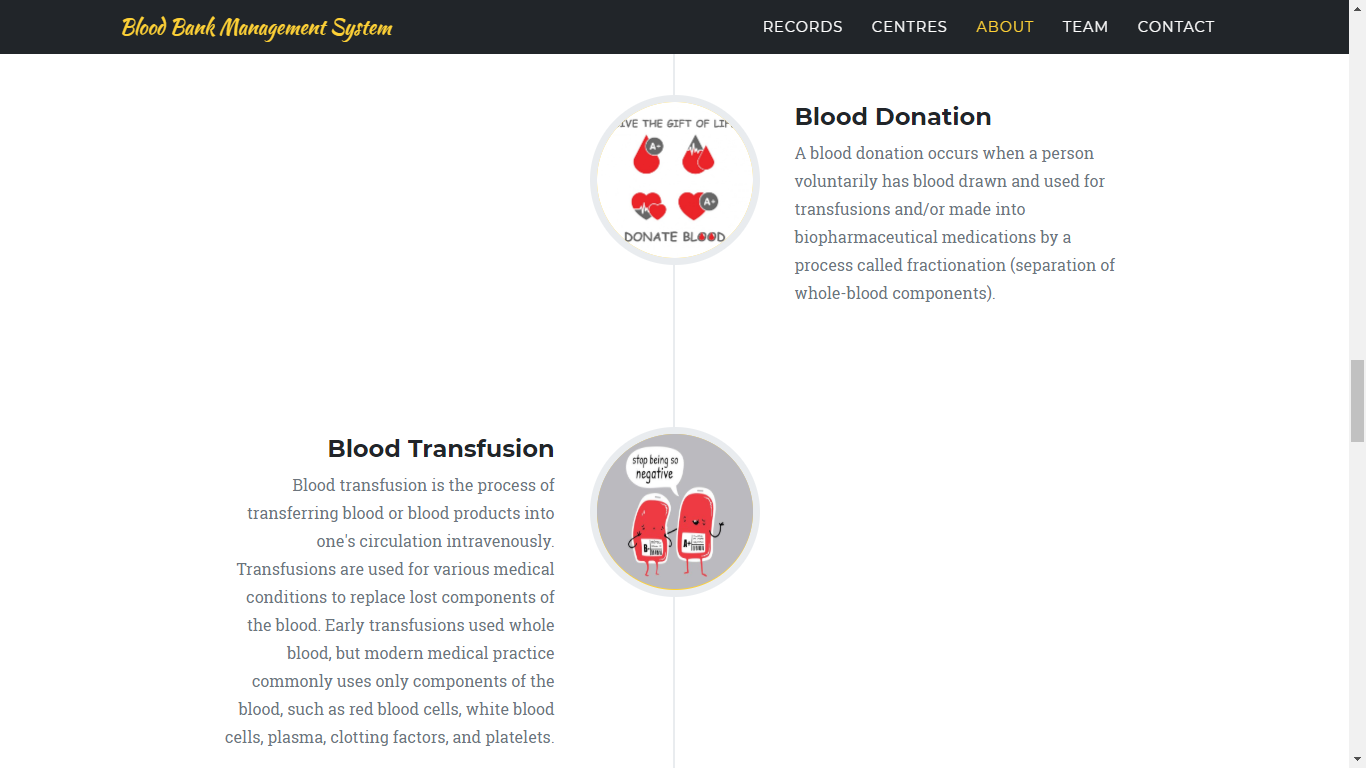


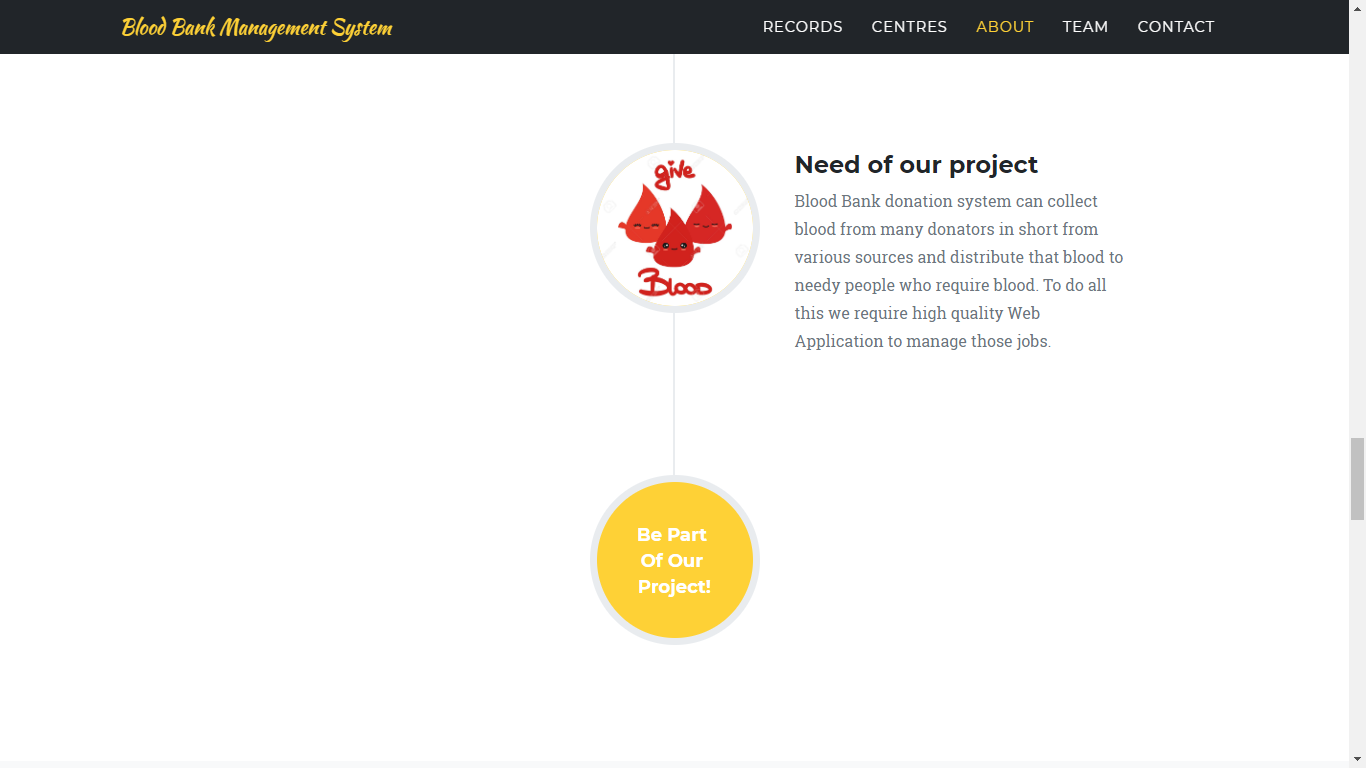


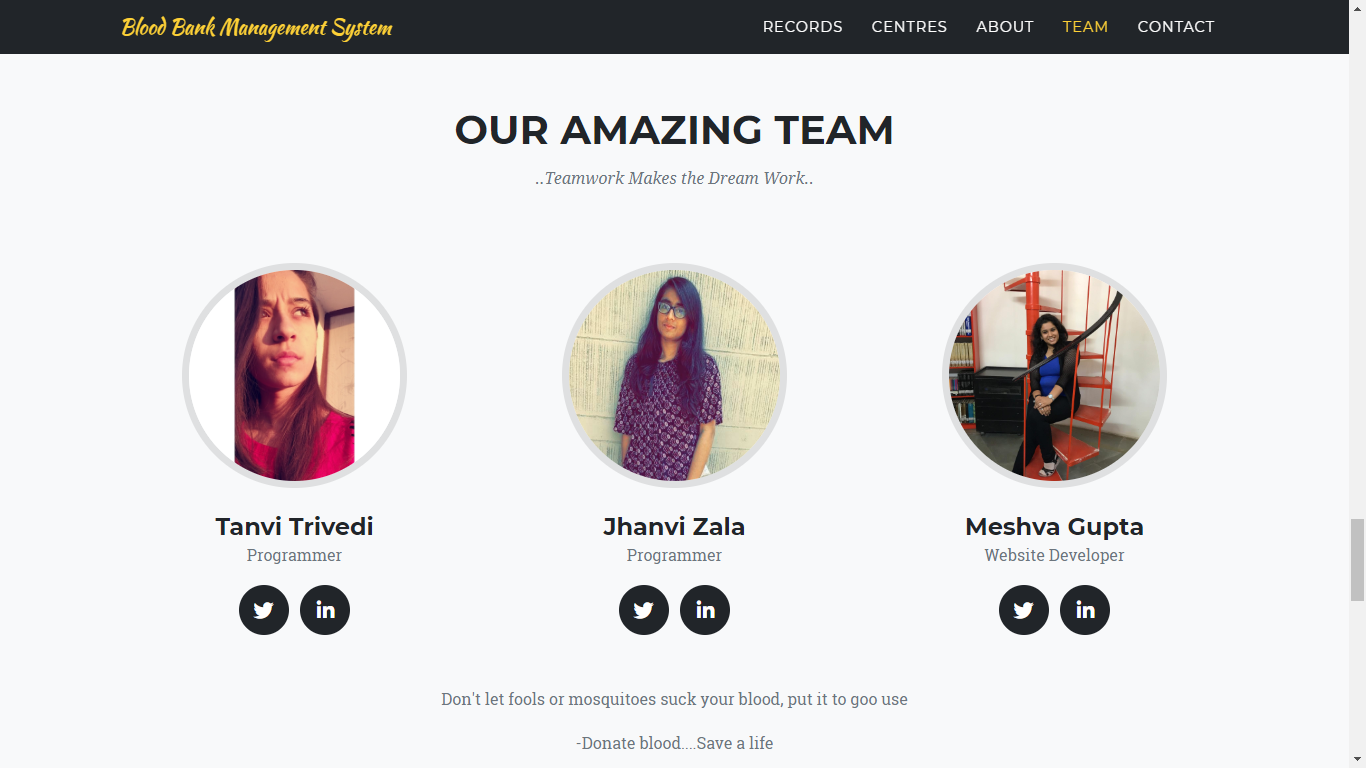
















**REFERENCES: (3 References Should Be**

**There)**

* [https://www.academia.edu › Blood\_Bank\_Management\_System\_A\_Project](https://www.academia.edu/30539620/Blood_Bank_Management_System_A_Project_Presentation_on_Internal_Guide)
* [https://www.freeprojectz.com › project-report](https://www.freeprojectz.com/project-report/1737)

**CONCLUSION:**

* Hereby, we conclude that In the world of information technology where whole world becomes global village, where end user can get the information just sitting at home on one click, infact government has taken a step in order to transform the system. Blood Bank Management information system helps to make the system paperless. It is small contribution of the researcher in order to serve the mankind. This is to make sure that the management of the blood stock   
  became effective, systematic and meeting user requirements.