MAHATMA GANDHI MEMORIAL COLLEGE UDUPI-576102

Accredited by NAAC with “A+” Grade(CGPA 3.36)

PROJECT REPORT ON

“BLOOD BANK MANAGEMENT SYSTEM”

DEVELOPED BY

Navaneetha J N	**Reg. No. :** U05MG21S0246
Sathwik Shetty	**Reg. No. :** U05MG21S0120

Under The Guidance Of

Prof. Deepali Kamath

Department of Computer Science

Submitted to the Mangalore University in partial fulfilment of the Award Bachelor in Computer Application

MANGLORE UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

MAHATMA GANDHI MEMORIAL COLLEGE

UDUPI-576102

2023-2024

MAHATMA GANDHI MEMORIAL COLLEGE
UDUPI – 576102

Accredited by NAAC with "A+" Grade (CGPA 3.36)

2022-2023

DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

This is to certify that the project entitled "**BLOOD BANK MANAGEMENT SYSTEM"** has been carried out by:

Navaneetha J N	**U05MG21S0246**
Sathwik Shetty	**U05MG21S0120**

Students of **B.C.A** under the supervision and guidance and submitted as partial fulfilment of the requirement for the award of **Bachelor in Computer Application** of **Manglore University** during the academic year 2023-2024.

They have certified out this project during the period April-2024 to June 2024 in our Lab.Further.it is certified to the best of our knowledge that the matter embodied in this work has not been submitted for the award of any other degree.

Ms.Deepali Kamath	**Prof.Laxminarayana Karath**	**Dr.M.VishwanathPai**
(Project guide)	(Principal)	(Head of Dept)

Submitted to the Manglore University Practical Examination held on ___________________

At Mahatma Gandhi Memorial College Udupi.

Examiners:

1._________________________

2._________________________

Index

ACKNOWLEDGEMENT

Behind every achievement, there is a sea of gratitude to those who have activated this project. The magnitude of this project demanded the co-operation, guidance and help of many people. We have been fortunate enough to have this in the entire task of completion of our project on “BLOOD BANK MANAGEMENT SYSTEM”.

We would like to thank our principal Prof. Laxminarayana Karanth for giving us opportunity to carry out our project.

We thank Dr. M. Vishwanath Pai, a source of inspiration and encouragement, head of the department of Computer Science, Mahatma Gandhi Memorial College, Udupi for having permitted us to carry out our project work.

We are extremely grateful to express our overwhelming gratitude to our guide Prof. Deepali Kamath, Lecturer of Computer Science Department, Mahatma Gandhi Memorial College Udupi for giving us valuable guidance to undertake this project.

Last but not the least, we are indebted to all teaching and non-teaching staff members, Mahatma Gandhi Memorial College Udupi for making this project successful.

Thanking you

Sathwik Shetty

Navaneetha J N

DECLARATION BY STUDENTS

We, the final year students of BCA, hereby declare that the project report entitled "Blood Bank Management System" is an website based project carried out by us under the guidance and supervision of Prof.Deepali Kamath, Lecturer of Department of Computer Science ,Mahatma Gandhi Memorial College, Udupi. We assure that this project work embodied the result of our original work and contents of the project have not been submitted to anybody else for the award of degree. This project is purely of academic interest. We have followed the guidelines provided by the university in writing the report.

Date: June 07,2024

Signature of the Students

Navaneetha J N

Sathwik Shetty

1. INTRODUCTION

The Blood Bank Management System (BBMS) is a comprehensive web-based application developed using PHP, designed to manage and streamline the operations of a blood bank. This system aims to facilitate the efficient collection, storage, and distribution of blood and blood products to meet the demands of hospitals, clinics, and patients. The project addresses critical aspects of blood bank management, including donor registration, blood inventory management, and the handling of blood requests and donations.

Blood is the saver of all existing lives in case of emergency needs. Blood donation is one of the noblest donations someone can ever make in his life. It is a great service that a person can offer to the society. The blood bank web application is used to maintain day-to-day transactions in a blood bank .The web application helps to register all the Donor, Blood collection details, Blood issued details etc. Typically the blood banks collect blood from voluntary Blood Donors. The project Blood Bank Application is developed so users can view the information about register Blood Donors such as name, address and other such personal information along with the details of Blood Group.

a. Project Title:

"BLOOD BANK MANAGEMENT SYSTEM".

b. Category:

Web Application.

c. Overview:

The Blood Bank Management System (BBMS) is a dynamic and interactive web application designed to automate and manage the various operations of a blood bank. Developed using PHP and integrated with MySQL for database management, the system aims to provide a streamlined, efficient, and user-friendly platform for handling blood donations, inventory management, and distribution processes.

2. IMPORTANCE

A Blood Bank Management System (BBMS) is an essential tool for modern healthcare, offering numerous advantages that significantly enhance the efficiency, accuracy, and overall effectiveness of blood bank operations.

By automating routine tasks such as donor registration, blood collection, inventory tracking, and request management, the BBMS reduces the workload on staff, allowing them to focus on more critical activities.The system provides real-time updates on inventory levels and donor information, ensuring that data is always current and easily accessible.

The system facilitates smooth donor registration and management, including appointment scheduling and communication, improving the overall donor experience and encouraging repeat donations. By efficiently managing blood requests from hospitals , the BBMS ensures that blood is available when and where it is needed, potentially saving lives in critical situations.

3.METHODOLOGY

1. Project Planning and Requirement Analysis:

Steps:

1. **Requirement Gathering**: Collect requirements from stakeholders, including blood bank staff, donors, and hospital representatives.
2. **Feasibility Study**: Assess the feasibility of the project in terms of technical, operational, and financial aspects.
3. **Requirement Specification**: Document the functional and non-functional requirements in a Software Requirement Specification (SRS) document.

2. System Design:

Steps:

1. **High-Level Design (HLD)**:
   - Define the system architecture, including the client-server model.
   - Identify the major modules and their interactions.
2. **Low-Level Design (LLD)**:
   - Design database schemas to include tables for donors, blood inventory, requests, and users.
   - Create detailed designs for each module, including data flow diagrams, entity-relationship diagrams, and use case diagrams.
3. **User Interface Design**: Develop wireframes and mockups for the system's user interface.

3. Technology Stack Selection:

Steps:

1. **Backend**: PHP for server-side scripting.
2. **Database**: MySQL for data storage and management.
3. **Frontend**: HTML, CSS, JavaScript, and Bootstrap for a responsive and user-friendly interface.

4. Implementation:

Steps:

1. **Setup Development Environment**: Configure the necessary development tools, including a local server (e.g., XAMPP or WAMP), code editors, and version control systems (e.g., Git).
2. **Module Development**:
   - **Donor Management**: Implement features for donor registration, profile management, and communication.

- **Inventory Management**: Develop functionalities for tracking blood units, including addition, update, and deletion of records.
- **Request Management**: Create features for handling blood requests from hospitals and clinics, including approval and tracking.
- **User Management**: Implement user authentication, role-based access control, and user profile management.
- **Reporting and Analytics**: Develop reporting tools to generate insights on donor activity, inventory status, and blood usage trends.

3. **Integration**: Integrate all modules to ensure they work seamlessly together.

5. Testing:

Steps:

1. **Unit Testing**: Test individual components and modules for correct functionality.
2. **Integration Testing**: Test the integration of modules to ensure they work together as intended.
3. **System Testing**: Conduct end-to-end testing of the complete system to validate overall performance.
4. **User Acceptance Testing (UAT)**: Involve end-users in testing to ensure the system meets their needs and expectations.
5. **Bug Fixing**: Identify and fix any issues discovered during testing.

6. Documentation:

Steps:

1. **User Documentation**: Create user manuals and help guides to assist end-users in using the system.
2. **Technical Documentation**: Document the system architecture, design, and code for future maintenance and development.

7. Maintenance and Support:

Steps:

1. **Bug Fixes and Updates**: Address any issues that arise post-deployment and release updates as needed.
2. **Feature Enhancements**: Add new features and enhancements based on user feedback and changing requirements.
3. **Regular Backups**: Implement a backup strategy to protect data against loss.

4.OBJECTIVES

- ➢ Automate and streamline the operations of blood banks to reduce the administrative burden and improve overall efficiency.
- ➢ Minimize errors in record-keeping and inventory management through automated data entry and validation.
- ➢ Facilitate donor registration and management, making it easier to schedule donations and maintain an updated database of donors.
- ➢ Maintain an accurate and up-to-date inventory of blood and blood products, ensuring the timely fulfillment of blood requests.

➢ Provide a web-based interface accessible to authorized personnel, enabling real-time updates and access to information from any location.

5.ANALYSIS AND INTERPRETATION

1.SYSTEM DESIGN :

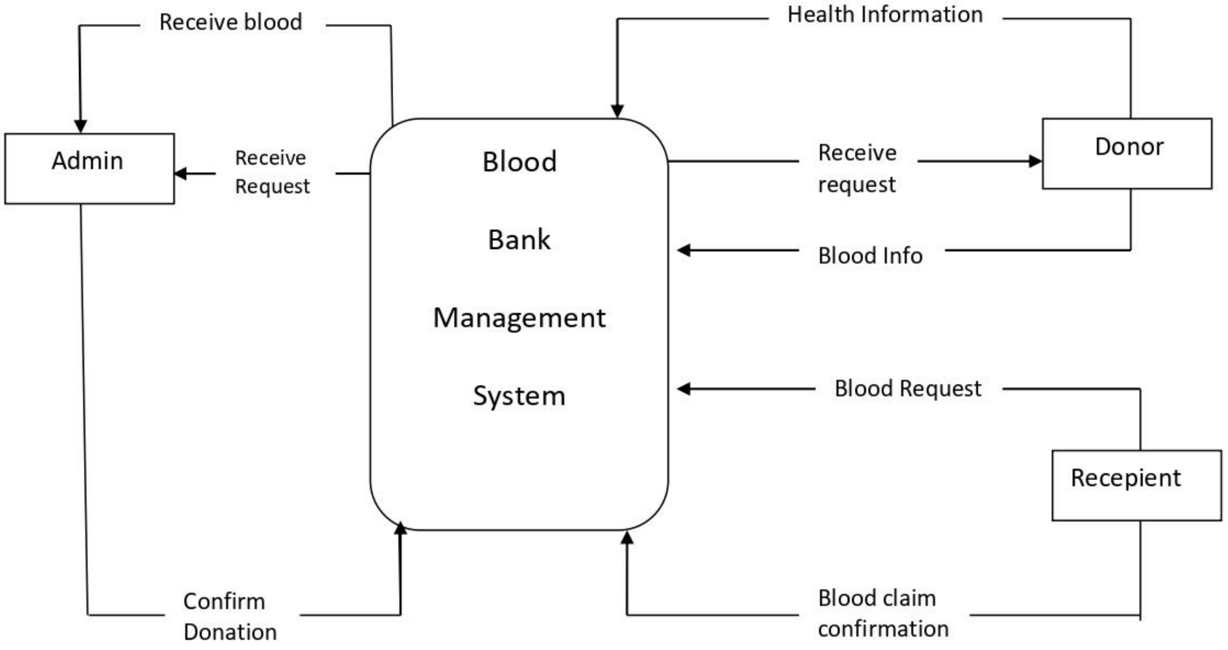
a. Introduction:

System design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organisation.

b. Description of Programs:

i. Context flow Diagram:

CFD is one in which the entire system is treated as a single process and all its inputs, outputs, sinks and sources are identified. A context diagram, sometimes called a level 0 data-flow diagram, is drawn in order to define and clarify the boundaries of the software system. It identifies the flows of the information between the system and external entities. The entire software system is shown as a single process. It is used to establish the context and boundaries of the system to be modelled and identify the relationship of the system with the external entities

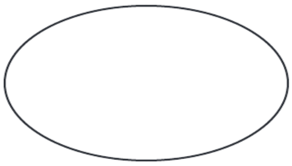
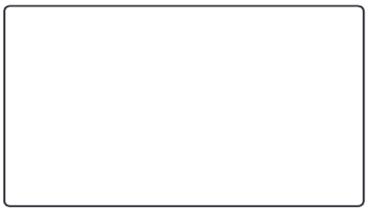
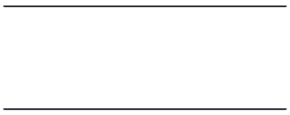
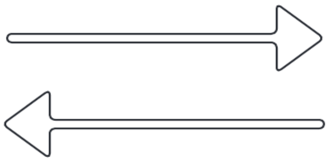


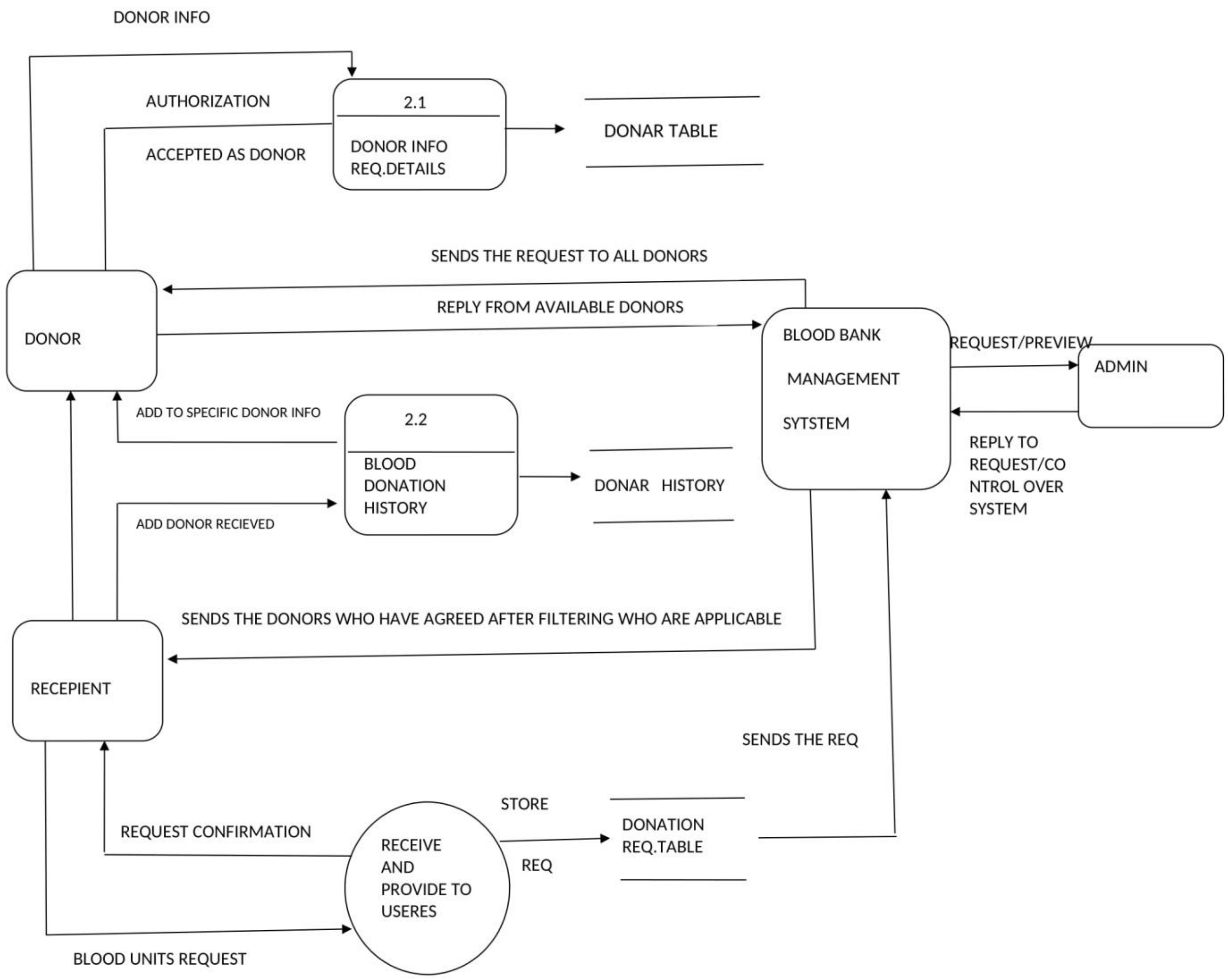
ii. <u>Data Flow Diagram:</u>

A data-flow diagram is a way of representing a flow through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data flow diagram has no control flow; there are no decision rules and no loops. Levels in DFD are numbered 0, 1, 2 and beyond.

The DFD server two purposes:

- To provide an indication of flow of data are transformed as they move through the system.
- To depict the function that transforms the data flow.

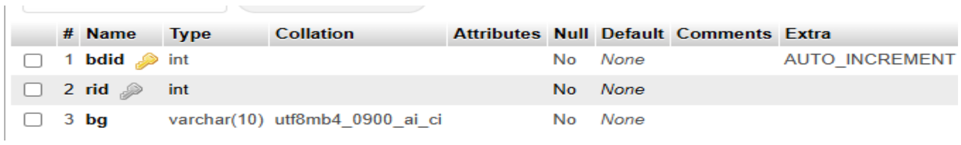
Notations	Description
	A circle represents a Process that performs some transformation of input data to yield output data.
	A rectangle is used to represent an external entity i.e a source and a sink. System that produce information by the software is a Source and a system that receives or stores information produced by the system is a Sink.
	The open box represents a repository of data stored that is used by the software.
	An arrow represents one or more data items or data objects and is used to connect processes to each other or to sources and sinks. The arrow head indicates direction of data flow.



2.DATABASE DESIGN:

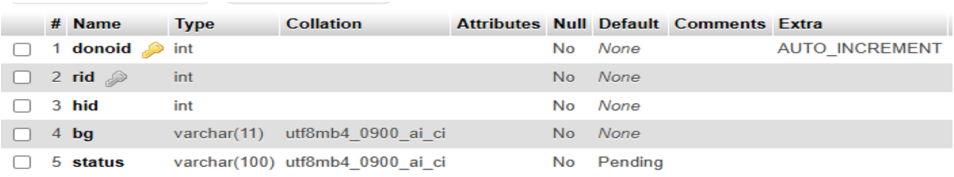
i. Schema Information:

a) Bloodinfo Table:



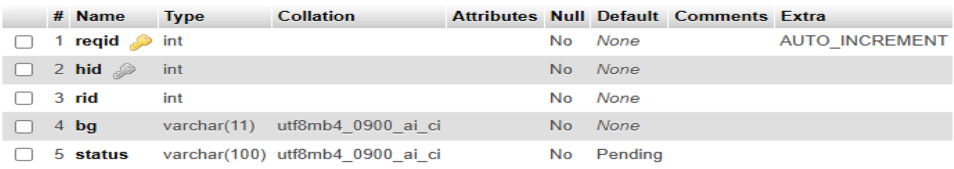
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
☐	1	bdid	int			No	None		AUTO_INCREMENT
☐	2	rid	int			No	None		
☐	3	bg	varchar(10)	utf8mb4_0900_ai_ci		No	None		

b) Blood Donate Table:



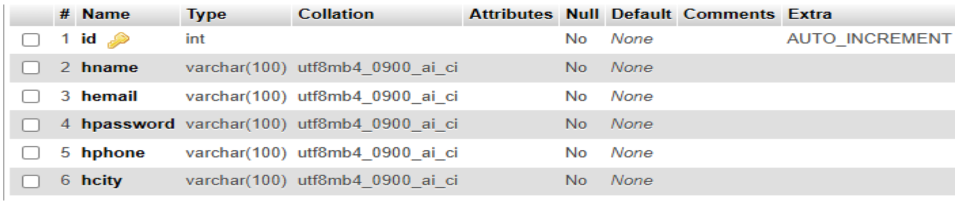
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
□	1	**donoid** 🔑	int			No	*None*		AUTO_INCREMENT
□	2	**rid** 🔑	int			No	*None*		
□	3	**hid**	int			No	*None*		
□	4	**bg**	varchar(11)	utf8mb4_0900_ai_ci		No	*None*		
□	5	**status**	varchar(100)	utf8mb4_0900_ai_ci		No	Pending		

c) Blood Request Table:



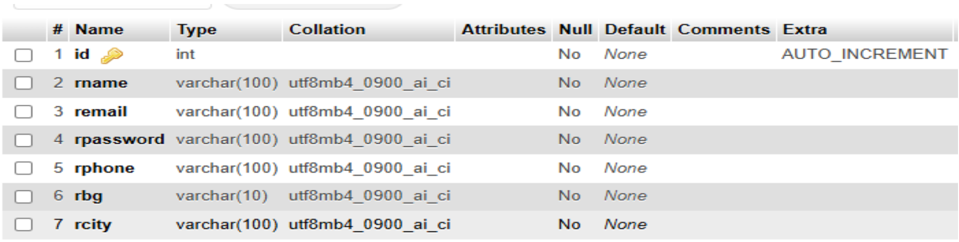
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
□	1	**reqid** 🔑	int			No	*None*		AUTO_INCREMENT
□	2	**hid** 🔑	int			No	*None*		
□	3	**rid**	int			No	*None*		
□	4	**bg**	varchar(11)	utf8mb4_0900_ai_ci		No	*None*		
□	5	**status**	varchar(100)	utf8mb4_0900_ai_ci		No	Pending		

d) Hospital Table:



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
□	1	**id** 🔑	int			No	*None*		AUTO_INCREMENT
□	2	**hname**	varchar(100)	utf8mb4_0900_ai_ci		No	*None*		
□	3	**hemail**	varchar(100)	utf8mb4_0900_ai_ci		No	*None*		
□	4	**hpassword**	varchar(100)	utf8mb4_0900_ai_ci		No	*None*		
□	5	**hphone**	varchar(100)	utf8mb4_0900_ai_ci		No	*None*		
□	6	**hcity**	varchar(100)	utf8mb4_0900_ai_ci		No	*None*		

e) Receiver Table:



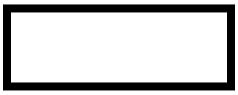
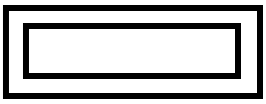
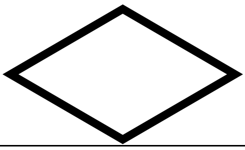
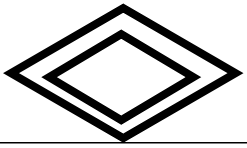
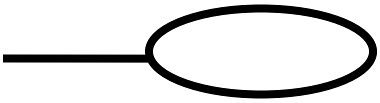
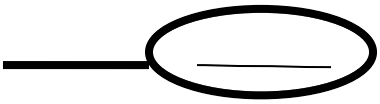
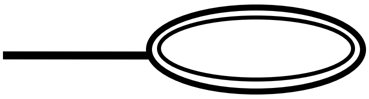
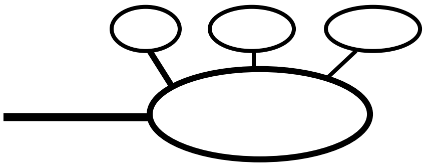
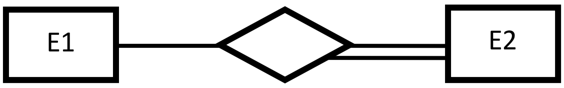
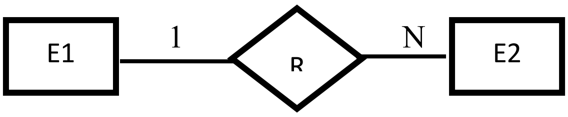
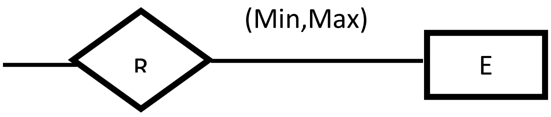
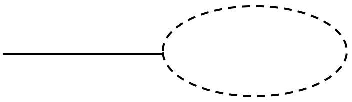
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
□	1	**id** 🔑	int			No	*None*		AUTO_INCREMENT
□	2	**rname**	varchar(100)	utf8mb4_0900_ai_ci		No	*None*		
□	3	**remail**	varchar(100)	utf8mb4_0900_ai_ci		No	*None*		
□	4	**rpassword**	varchar(100)	utf8mb4_0900_ai_ci		No	*None*		
□	5	**rphone**	varchar(100)	utf8mb4_0900_ai_ci		No	*None*		
□	6	**rbg**	varchar(10)	utf8mb4_0900_ai_ci		No	*None*		
□	7	**rcity**	varchar(100)	utf8mb4_0900_ai_ci		No	*None*		

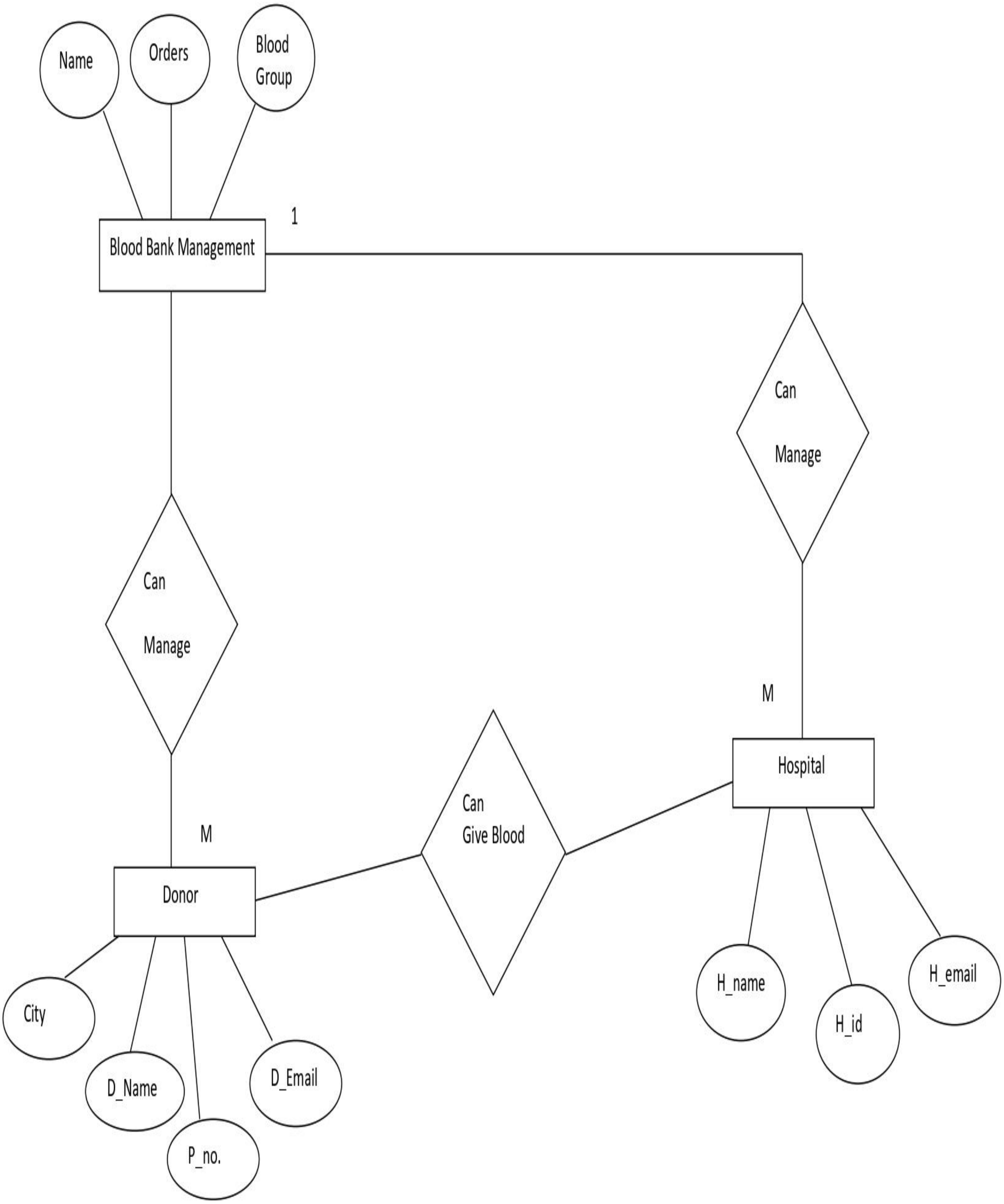
ii. ER Diagram:

Entity Relationship (ER) Diagram:

- Entity Relationship Model is a popular high-level conceptual model.

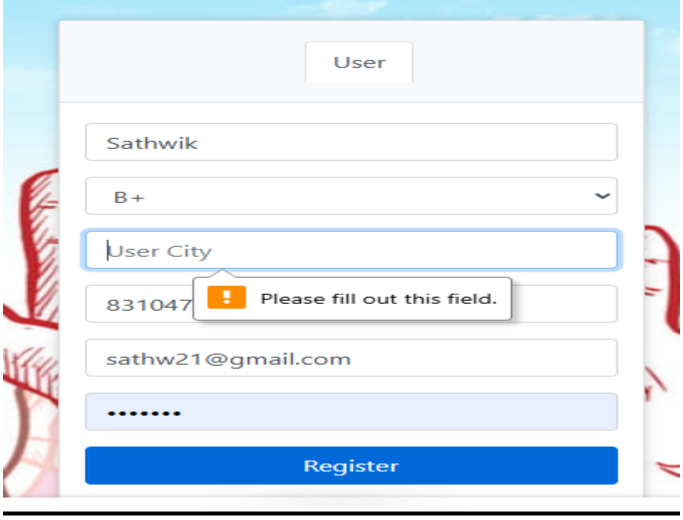
Basic of Entity Relationship Notation:-

Symbol	Mean
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Total participation of E2 in R
	Cardinality ratio 1:N for E1:E2 in R
	Structured constraint (Min, Max) on participation of E in R
	Derived Attribute

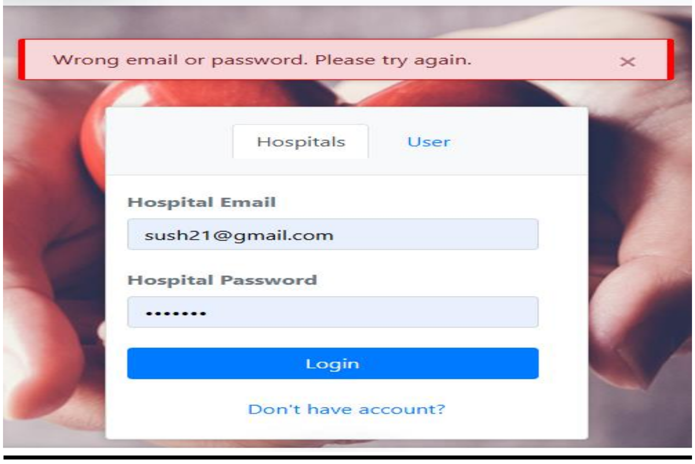

Name
Orders
Blood Group
Blood Bank Management
1
Can Manage
Can Manage
M
M
Hospital
Can Give Blood
Donor
City
D_Name
P_no.
D_Email
H_name
H_id
H_email

6.TESTING

Test case 1:Test for all the fields



Test case 2: : Test for Username and Password

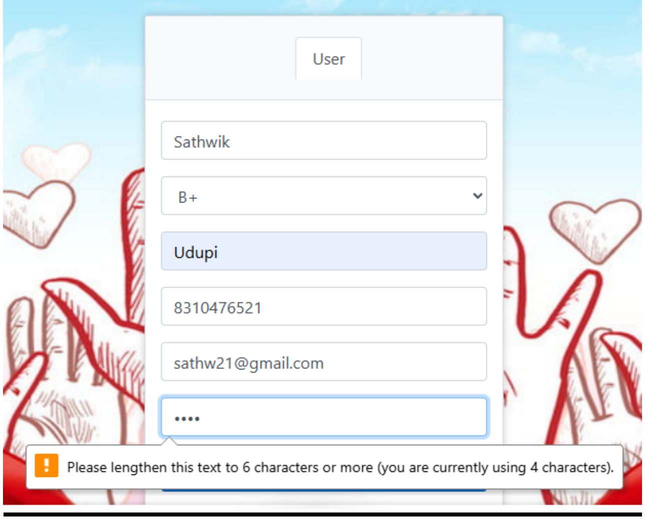


Wrong email or password. Please try again.

Hospital Email: sush21@gmail.com

Hospital Password: •••••••

Test case3:Checking Password Length



Sathwik

B+

Udupi

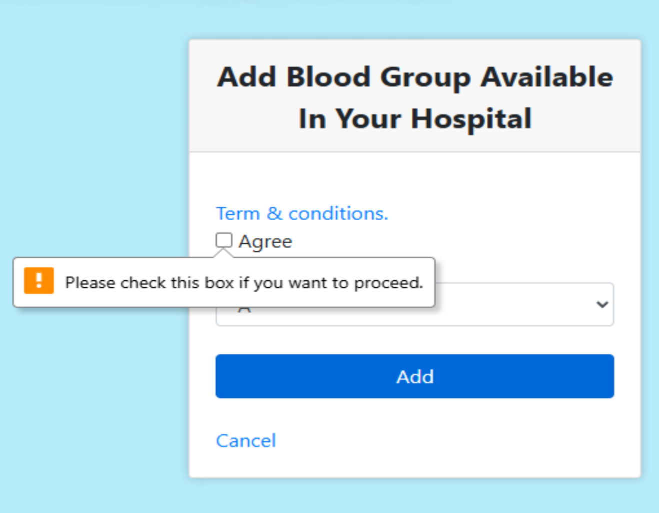
8310476521

sathw21@gmail.com

••••

Please lengthen this text to 6 characters or more (you are currently using 4 characters).

Test case4:



Add Blood Group Available In Your Hospital

Term & conditions.

□ Agree

Please check this box if you want to proceed.

7.CONCLUSION

The Blood Bank Management System is a critical tool that transforms the way blood banks operate. By enhancing efficiency, accuracy, security, and communication, the BBMS ensures that blood banks can provide timely and reliable services, ultimately contributing to better healthcare outcomes and saving lives. The importance of such a system cannot be overstated, as it plays a vital role in ensuring the availability and quality of blood products, which are essential for countless medical treatments and emergency interventions.

8.LEARNING OUTCOME OF THE PROJECT

Technical Skills Development

➢ **PHP Programming**:

- Enhanced proficiency in PHP, including object-oriented programming, session management, and database interactions.
- Improved understanding of PHP frameworks and libraries that can be leveraged for web development.SS

➢ **Database Management**:

- Gained expertise in MySQL for designing and managing relational databases.
- Learned to write efficient SQL queries for data retrieval, manipulation, and reporting.

➢ **Frontend Development**:

- Acquired skills in HTML, CSS, and JavaScript to create a responsive and user-friendly interface.
- Used Bootstrap to design aesthetically pleasing and mobile-friendly web pages.

- **Full Stack Development**:

- Understood the integration of frontend and backend components to create a cohesive web application.
- Gained experience in managing data flow between the client-side interface and the server-side logic.

- **Requirement Analysis**:

- Learned the importance of thorough requirement gathering and analysis to understand stakeholder needs and project scope.

- **Team Collaboration**:

- Enhanced ability to work effectively in a team, coordinating with other developers, designers, and stakeholders.
- Learned the importance of clear communication and documentation to ensure all team members are aligned and informed.

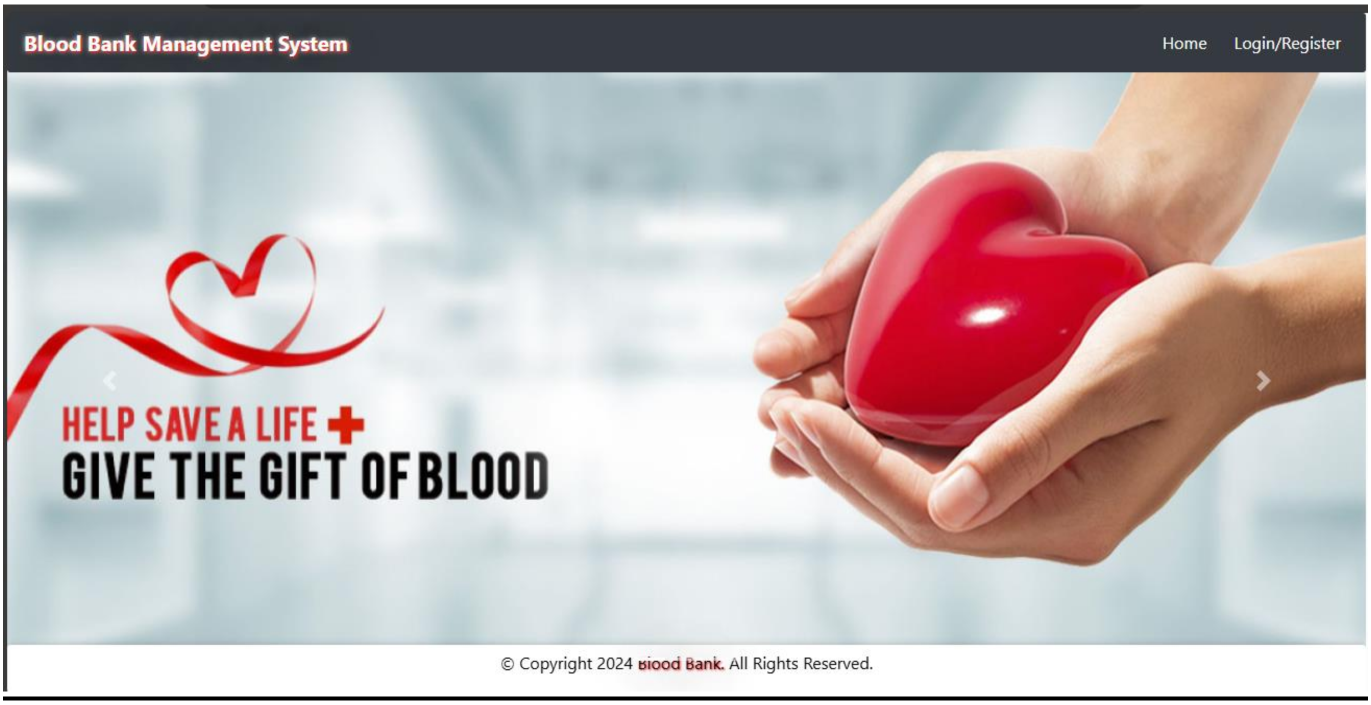
9.REFERENCES IN APA STYLE

Online Resources

- Bootstrap Documentation. (n.d.). Retrieved from https://getbootstrap.com/docs/4.0/getting-started/introduction/
- PHP: Hypertext Preprocessor. (n.d.). Retrieved from https://www.php.net/manual/en/intro-whatis.php
- W3Schools. (n.d.). SQL Tutorial. Retrieved from https://www.w3schools.com/sql/

10.PHOTOGRAPHES

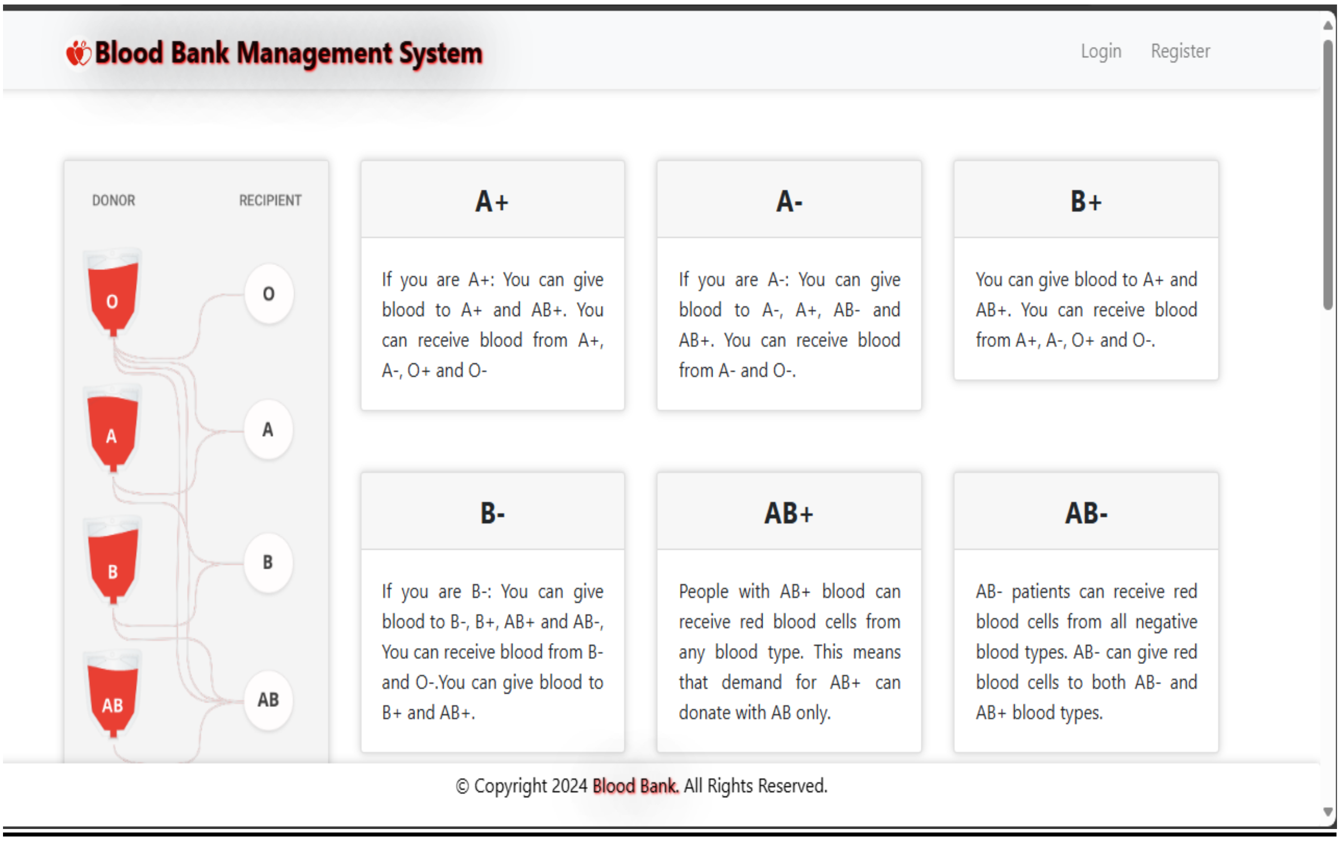
Home page:



Blood Bank Management System

© Copyright 2024 Blood Bank. All Rights Reserved.

Login/Register page:



Blood Bank Management System

A+

If you are A+: You can give blood to A+ and AB+. You can receive blood from A+, A-, O+ and O-

A-

If you are A-: You can give blood to A-, A+, AB- and AB+. You can receive blood from A- and O-.

B+

You can give blood to A+ and AB+. You can receive blood from A+, A-, O+ and O-.

B-

If you are B-: You can give blood to B-, B+, AB+ and AB-, You can receive blood from B- and O-.You can give blood to B+ and AB+.

AB+

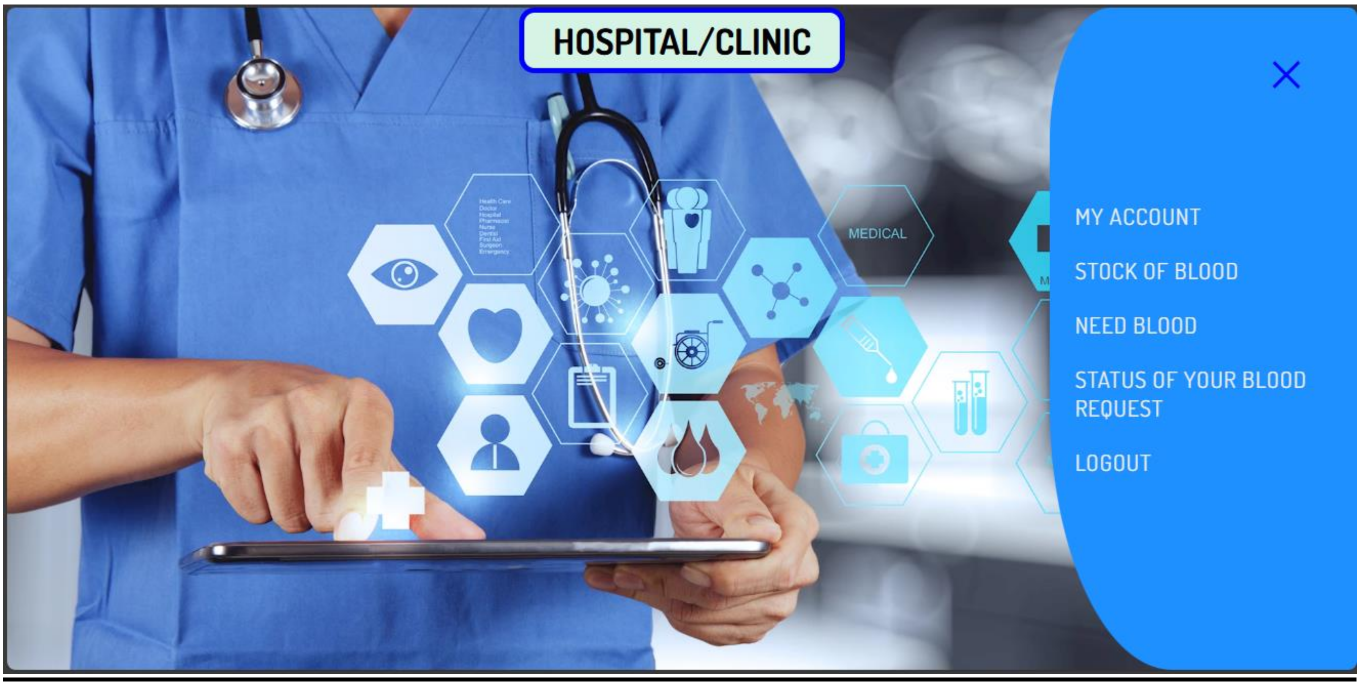
People with AB+ blood can receive red blood cells from any blood type. This means that demand for AB+ can donate with AB only.

AB-

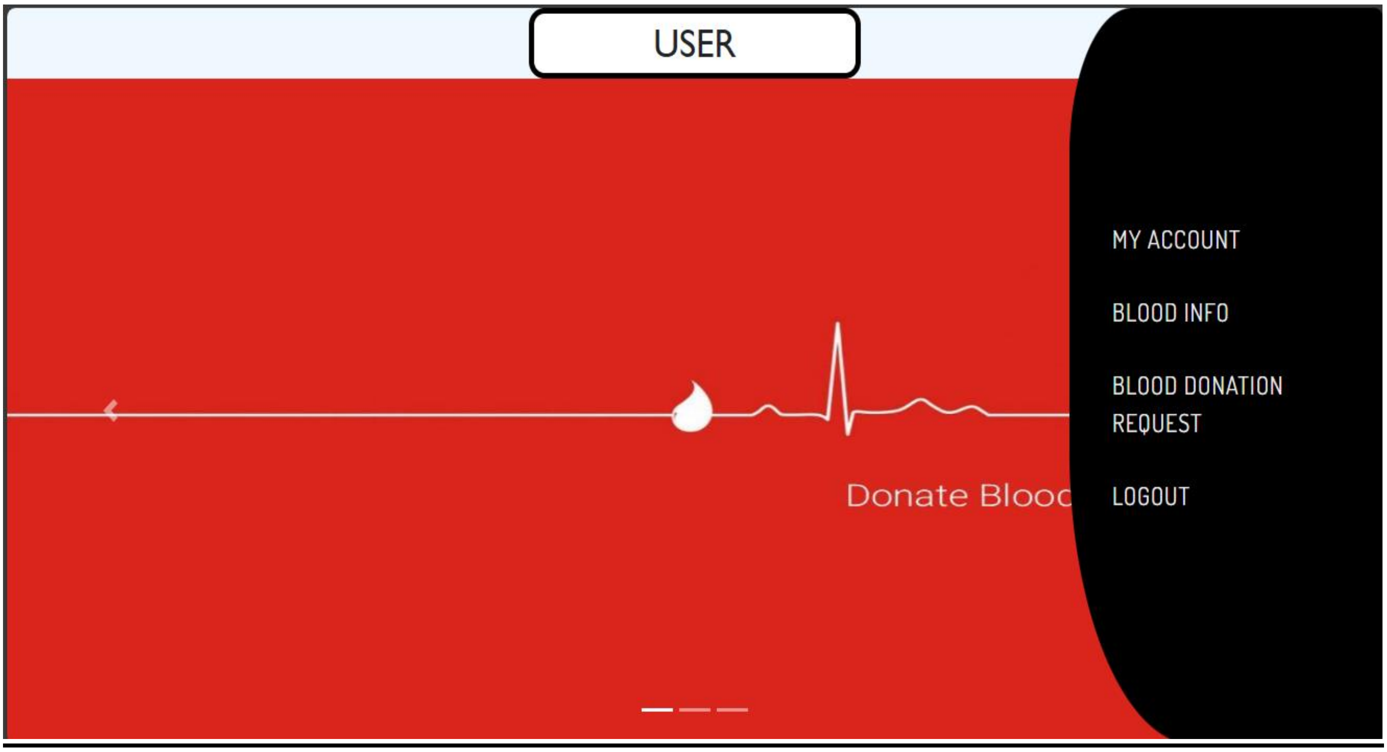
AB- patients can receive red blood cells from all negative blood types. AB- can give red blood cells to both AB- and AB+ blood types.

© Copyright 2024 Blood Bank. All Rights Reserved.

Hospital page:



Donar page:



11.CODING

Register.php

```
<?php

session_start();

if (isset($_SESSION['hid'])) {

  header("location:bloodrequest.php");

}elseif (isset($_SESSION['rid'])) {

  header("location:sentrequest.php");

}else{

?>

<!DOCTYPE html>

<html lang="en">

<head>

  <style>

body{

    background: url(jastimage/bb1.jpg) no-repeat center;

    background-size: cover;

    min-height: 0;

    height: 530px;

  }

.login-form{

    width: calc(100% - 20px);

    max-height: 650px;
    max-width: 450px;

    background-color: white;

}

</style>

</head>

<?php $title="Bloodbank | Register"; ?>

<?php require 'head.php'; ?>

<body>

  <?php include 'header.php'; ?>



    <div class="container cont">



    <?php require 'message.php'; ?>



      <div class="row justify-content-center">

        <div class="col-lg-4 col-md-5 col-sm-6 col-xs-7 mb-5">

          <div class="card rounded">

            <ul class="nav nav-tabs justify-content-center bg-light" style="padding: 20px">



              <li class="nav-item">

                <a class="nav-link" data-toggle="tab" href="#receivers">User</a>

              </li>

            </ul>
    <div class="tab-content">
        <div class="tab-pane container fade" id="receivers">
          <form action="file/receiverReg.php" method="post" enctype="multipart/form-data">
          <input type="text" name="rname" placeholder="User Name" class="form-control mb-
3" required>
          <select name="rbg" class="form-control mb-3" required>
                <option disabled="" selected="">Blood Group</option>
                <option value="A+">A+</option>
                <option value="A-">A-</option>
                <option value="B+">B+</option>
                <option value="B-">B-</option>
                <option value="AB+">AB+</option>
                <option value="AB-">AB-</option>
                <option value="O+">O+</option>
                <option value="O-">O-</option>
          </select>
          <input type="text" name="rcity" placeholder="User City" class="form-control mb-3"
required>
          <input type="tel" name="rphone" placeholder="User Phone Number" class="form-
control mb-3" required pattern="[0,6-9]{1}[0-9]{9,11}" title="Password must have start
from 0,6,7,8 or 9 and must have 10 to 12 digit">
          <input type="email" name="remail" placeholder="User Email" class="form-control
mb-3" required>
```

```
          <input type="password" name="rpassword" placeholder="User Password" class="form-control mb-3" required minlength="6">
          <input type="submit" name="rregister" value="Register" class="btn btn-primary btn-block mb-4">
        </form>


      </div>
   </div>
   <a href="login.php" class="text-center mb-4" title="Click here">Already have account?</a>
</div>
</div>
</div>
</div>
<?php require 'footer.php' ?>
</body>
</html>
<?php } ?>
```

Receiverreg.php

```
<?php
if(isset($_POST['rregister'])){
          require 'connection.php';
          $rname=$_POST['rname'];
```

```
        $remail=$_POST['remail'];

        $rpassword=$_POST['rpassword'];

        $rphone=$_POST['rphone'];

        $rcity=$_POST['rcity'];

        $rbg=$_POST['rbg'];

        $check_email = mysqli_query($conn, "SELECT remail FROM receivers where remail = '$remail' ");

        if(mysqli_num_rows($check_email) > 0){

   $error= 'Email Already exists. Please try another Email.';

   header( "location:../register.php?error=".$error );

}else{

        $sql = "INSERT INTO receivers (rname, remail, rpassword, rphone, rcity, rbg)

        VALUES ('$rname','$remail', '$rpassword', '$rphone', '$rcity', '$rbg')";

        if ($conn->query($sql) === TRUE) {

                $msg = "You have successfully registered. Please, login to continue.";

                header( "location:../login.php?msg=".$msg);

        } else {

                $error = "Error: " . $sql . "<br>" . $conn->error;

      header( "location:../register.php?error=".$error );

        }

        $conn->close();

}

}

?>
```

Login.php

```
<?php
session_start();
if (isset($_SESSION['hid'])) {
  header("location:bloodrequest.php");
}elseif (isset($_SESSION['rid'])) {
  header("location:sentrequest.php");
}else{
?>
<!DOCTYPE html>
<html>
<head>
  <style>
    body{
    background: url(image/RBC11.jpg) no-repeat center;
    background-size: cover;
    min-height: 0;
    height: 650px;
  }
.login-form{
    width: calc(100% - 20px);
    max-height: 650px;
    max-width: 450px;
    background-color: white;

}

</style>

</head>

<?php $title="Bloodbank | Login"; ?>

<?php require 'head.php'; ?>

<body>

  <?php require 'header.php'; ?>



    <div class="container cont">



      <?php require 'message.php'; ?>



      <div class="row justify-content-center">

        <div class="col-lg-4 col-md-5 col-sm-6 col-xs-7 mb-5">



          <div class="card rounded">

            <ul class="nav nav-tabs justify-content-center bg-light" style="padding: 20px;">

      <li class="nav-item">

        <a class="nav-link active" data-toggle="tab" href="#hospitals">Hospitals</a>

      </li>

     <li class="nav-item">

        <a class="nav-link" data-toggle="tab" href="#receivers">User</a>
  </li>

 </ul>


 <div class="tab-content">

   <div class="tab-pane container active" id="hospitals">

    <form action="file/hospitalLogin.php" class="login-form" method="post">

     <label class="text-muted font-weight-bold" class="text-muted font-weight-bold">Hospital Email</label>

     <input type="email" name="hemail" placeholder="Hospital Email" class="form-control mb-4">

     <label class="text-muted font-weight-bold" class="text-muted font-weight-bold">Hospital Password</label>

     <input type="password" name="hpassword" placeholder="Hospital Password" class="form-control mb-4">

     <input type="submit" name="hlogin" value="Login" class="btn btn-primary btn-block mb-4">

    </form>

   </div>



  <div class="tab-pane container fade" id="receivers">

    <form action="file/receiverLogin.php" class="login-form" method="post">

     <label class="text-muted font-weight-bold" class="text-muted font-weight-bold">User Email</label>
```

```
          <input type="email" name="remail" placeholder="User Email" class="form-control mb-4">
          <label class="text-muted font-weight-bold" class="text-muted font-weight-bold">User Password</label>
          <input type="password" name="rpassword" placeholder="User Password" class="form-control mb-4">
          <input type="submit" name="rlogin" value="Login" class="btn btn-primary btn-block mb-4">
        </form>
      </div>


    </div>
    <a href="register.php" class="text-center mb-4" title="Click here">Don't have account?</a>
</div>
</div>
</div>
</div>
<?php require 'footer.php' ?>
</body>
</html>
<?php } ?>
```

Hospitallogin.php

```
<?php
```

```
session_start();
   require 'connection.php';
   if(isset($_POST['hlogin'])){
   $hemail=$_POST['hemail'];
   $hpassword=$_POST['hpassword'];
   $sql="select * from hospitals where hemail='$hemail' and hpassword='$hpassword'";
   $result=mysqli_query($conn,$sql) or die(mysqli_error($conn));
   $rows_fetched=mysqli_num_rows($result);
   if($rows_fetched==0){
      $error= "Wrong email or password. Please try again.";
      header( "location:../login.php?error=".$error);
   }else{
      $row=mysqli_fetch_array($result);
      $_SESSION['hemail']=$row['hemail'];
      $_SESSION['hname']=$row['hname'];
      $_SESSION['hid']=$row['id'];
      $msg= $_SESSION['hname'].' have logged in.';
      header( "location:../hospitalpage.html?msg=".$msg);
   }
  }
?>
```

ReceiverLogin.php

```
<?php
session_start();
    require 'connection.php';
    if(isset($_POST['rlogin'])){
    $remail=$_POST['remail'];
    $rpassword=$_POST['rpassword'];
    $sql="select * from receivers where remail='$remail' and rpassword='$rpassword'";
    $result=mysqli_query($conn,$sql) or die(mysqli_error($conn));
    $rows_fetched=mysqli_num_rows($result);
    if($rows_fetched==0){
        $error= "Wrong email or password. Please try again.";
        header( "location:../login.php?error=".$error);
    }else{
        $row=mysqli_fetch_array($result);
        $_SESSION['remail']=$row['remail'];
        $_SESSION['rname']=$row['rname'];
        $_SESSION['rid']=$row['id'];
        $msg= $_SESSION['rname'].' have logged in.';
        header( "location:../Userpage.html?msg=".$msg);
    }
  }
?>
```

SentRequested.php

```
<?php
require 'file/connection.php';
session_start();
  if(!isset($_SESSION['hid']))
  {
  header('location:login.php');
  }
  else {
    $hid = $_SESSION['hid'];
    $sql = "SELECT blooddonate.*, receivers.* from blooddonate, receivers where hid='$hid'
&& blooddonate.rid=receivers.id";
    $result = mysqli_query($conn, $sql);
?>


<!DOCTYPE html>
<html>
<?php $title="Bloodbank | Sent Requests"; ?>
<?php require 'head.php'; ?>
<style>
    body{
    background: url(image/p4.jpg) no-repeat center;
    background-size: cover;
    min-height: 0;
    height: 650px;

  }

.login-form{

    width: calc(100% - 20px);

    max-height: 650px;

    max-width: 450px;

    background-color: white;

}

</style>

<body>

        <?php require 'header.php'; ?>

        <div class="container cont">


                <?php require 'message.php'; ?>


        <table class="table table-responsive table-striped rounded mb-5">

                <tr><th colspan="8" class="title">Sent requests</th></tr>

                <tr>

                        <th>#</th>

                        <th>Name</th>

                        <th>Email</th>

                        <th>City</th>

                        <th>Phone</th>
```

```
                    <th>Blood Group</th>

                    <th>Status</th>

                    <th>Action</th>

               </tr>


                 <div>

            <?php

            if ($result) {

               $row =mysqli_num_rows( $result);

               if ($row) { //echo "<b> Total ".$row." </b>";

            }else echo '<b style="color:white;background-color:red;padding:7px;border-
radius: 15px 50px;">You have not requested yet. </b>';

          }

          ?>

          </div>


               <?php while($row = mysqli_fetch_array($result)) { ?>


               <tr>

                    <td><?php echo ++$counter;?></td>

                    <td><?php echo $row['rname'];?></td>

                    <td><?php echo $row['remail'];?></td>

                    <td><?php echo $row['rcity'];?></td>

                    <td><?php echo $row['rphone'];?></td>
```

```
                              <td><?php echo $row['bg'];?></td>

                              <td><?php echo $row['status'];?></td>

                              <td><?php if($row['status'] == 'Accepted'){ ?>

                              <?php }

                              else{ ?>

                                       <a href="file/canceld.php?donoid=<?php echo $row['donoid'];?>"
class="btn btn-danger">Cancel</a>

                              <?php } ?>

                              </td>

                     </tr>

                     <?php } ?>

          </table>

</div>

   <?php require 'footer.php'; ?>

</body>

</html>

<?php } ?>
```

<u>Blooddinfo.php</u>

```
<?php

  require 'file/connection.php';

  session_start();

  if(!isset($_SESSION['rid']))

  {
```

```
  header('location:login.php');
  }
  else {
?>
<!DOCTYPE html>
<html>
<?php $title="Bloodbank | Add blood samples"; ?>
<?php require 'head.php'; ?>
<style>
    body{
    background: url(image/p2.png) no-repeat center;
    background-size: cover;
    min-height: 0;
    height: 100%;
  }
.login-form{
    width: calc(100% - 20px);
    max-height: 650px;
    max-width: 450px;
    background-color: white;
}
</style>
<body>
```

```
<?php require 'header.php'; ?>

 <div class="container cont">

  <?php require 'message.php'; ?>

  <div class="row justify-content-center">

    <div class="col-lg-4 col-md-5 col-sm-6 col-xs-7 mb-5">
     <div class="card">
      <div class="card-header title">Add blood group available in your known community</div>
    <div class="card-body">
    <form action="file/infoAddd.php" method="post">
     <a data-toggle="collapse" href="#collapseExample" role="button" aria-expanded="false" aria-controls="collapseExample" title="click to see">Term & conditions.</a><br>
     <div class="collapse" id="collapseExample">
     If you or your Friends/Family have the mentioned(below) blood then only add Blood group(No spam).So,that the hospital can contact you with your given details if they are in need of you or your friends/family blood.You should have a blood sample tested by your doctor’s, nurse, or trained phlebotomist , at a pathology collection centre, clinic or hospital. Blood samples are most commonly taken from the inside of the elbow where the veins are usually closer to the surface.Make sure you have been eating healthy diet(No Smoking/Drinking)atleast for a week before you have to decided to donate Blood.By
```

```
clicking tick mark you are promising that you are promising that you have read and accepted
the above instructions and also willing to donate blood volunteerly.<br><br>

        </div>

          <input type="checkbox" name="condition" value="agree" required> Agree<br><br>

          <select class="form-control" name="bg" required="">


                <option>A-</option>

                <option>A+</option>

                <option>B-</option>

                <option>B+</option>

                <option>AB-</option>

                <option>AB+</option>

                <option>O-</option>

                <option>O+</option>

          </select><br>

          <input type="submit" name="add" value="Add" class="btn btn-primary btn-
block"><br>

          <a href="Userpage.html" class="float-right" title="click here">Cancel</a>

        </form>

         </div>

       </div>

     </div>


<?php   if(isset($_SESSION['rid'])){
    $rid=$_SESSION['rid'];
    $sql = "SELECT * from blooddinfo where rid='$rid'";
    $result = mysqli_query($conn, $sql);
  }
  ?>
    <div class="col-lg-4 col-md-5 col-sm-6 col-xs-7 mb-5">
          <table class="table table-striped table-responsive">
            <th colspan="4" class="title">User</th>
            <tr>
              <th>#</th>

              <th>User</th>
              <th>Action</th>
            </tr>
            <div>
                <?php
                if ($result) {
                    $row =mysqli_num_rows( $result);
                    if ($row) { //echo "<b> Total ".$row." </b>";
                }else echo '<b style="color:white;background-color:red;padding:7px;border-radius: 15px 50px;">Nothing to show.</b>';
            }
            ?>
            </div>
            <?php while($row = mysqli_fetch_array($result)) { ?>

            <tr>

              <td><?php echo ++$counter; ?></td>



              <td><?php echo $row['bg'];?></td>

              <td><a href="file/deleted.php?bdid=<?php echo $row['bdid'];?>" class="btn btn-
danger">Delete</a></td>

            </tr>

            <?php } ?>

          </table>

      </div>



   </div>

</div>

<?php require 'footer.php' ?>

</body>

<?php } ?>
```

<u>Blooddonate.php</u>

```
<?php

require 'file/connection.php';

session_start();

  if(!isset($_SESSION['rid']))

  {
```

```
  header('location:login.php');
  }
  else {
   $rid = $_SESSION['rid'];
   $sql = "SELECT blooddonate.*, hospitals.* from blooddonate, hospitals where rid='$rid'
&& blooddonate.hid=hospitals.id";
   $result = mysqli_query($conn, $sql);
?>

<!DOCTYPE html>
<html>
<style>
   body{
   background: url(image/p4.jpg) no-repeat center;
   background-size: cover;
   min-height: 0;
   height: 650px;
  }
.login-form{
   width: calc(100% - 20px);
   max-height: 650px;
   max-width: 450px;
   background-color: white;
}
```

```
.footer {
        position: fixed;
        left: 0;
        bottom: 0;
        width: 100%;
        background-color: white;
        color: black;
        text-align: center;
}
</style>
<?php $title="Bloodbank | Blood Donate"; ?>
<?php require 'head.php'; ?>
<body>
        <?php require 'header.php'; ?>
        <div class="container cont">

                <?php require 'message.php'; ?>


        <table class="table table-responsive table-striped rounded mb-5">
                <tr><th colspan="9" class="title">Blood Donate</th></tr>
                <tr>
                        <th>#</th>
                        <th>Name</th>
```

```
                    <th>Email</th>

                    <th>City</th>

                    <th>Phone</th>

                    <th>Blood Group</th>

                    <th>Status</th>

                    <th colspan="2">Action</th>

               </tr>


                 <div>

             <?php

             if ($result) {

                 $row =mysqli_num_rows( $result);

                 if ($row) { //echo "<b> Total ".$row." </b>";

             }else echo '<b style="color:white;background-color:red;padding:7px;border-
radius: 15px 50px;">No one has requested yet. </b>';

           }

           ?>

           </div>


               <?php while($row = mysqli_fetch_array($result)) { ?>


               <tr>

                    <td><?php echo ++$counter;?></td>

                    <td><?php echo $row['hname'];?></td>
```

```
                        <td><?php echo $row['hemail'];?></td>

                        <td><?php echo $row['hcity'];?></td>

                        <td><?php echo $row['hphone'];?></td>

                        <td><?php echo $row['bg'];?></td>

<td><?php echo 'You have '.$row['status'];?></td>

                        <td><?php if($row['status'] == 'Accepted'){ ?> <a href="" class="btn btn-
success disabled">Accepted</a> <?php }

                        else{ ?>

                                <a href="file/acceptd.php?donoid=<?php echo $row['donoid'];?>"
class="btn btn-success">Accept</a>

                        <?php } ?>

                        </td>

                        <td><?php if($row['status'] == 'Rejected'){ ?> <a href="" class="btn btn-
danger disabled">Rejected</a> <?php }

                        else{ ?>

                                <a href="file/rejectd.php?donoid=<?php echo $row['donoid'];?>"
class="btn btn-danger">Reject</a>

                        <?php } ?>

                        </td>


                </tr>

                <?php } ?>

        </table>
```

```
</div>
<?php require 'footer.php'; ?>
</body>
</html>
<?php } ?>
```

Accept.php

```
<?php
include "connection.php";
   $reqid=$_GET['reqid'];
        $status = "Accepted";
        $sql = "update bloodrequest SET status = '$status' WHERE reqid = '$reqid'";
   if (mysqli_query($conn, $sql)) {
        $msg="You have accepted the request.";
        header("location:../bloodrequest.php?msg=".$msg );
   } else {
   $error= "Error changing status: " . mysqli_error($conn);
   header("location:../bloodrequest.php?error=".$error );
   }
   mysqli_close($conn);
?>
```

Reject.php

```
<?php
```

```
include "connection.php";
   $reqid=$_GET['reqid'];
        $status = "Rejected";
        $sql = "update bloodrequest SET status = '$status' WHERE reqid = '$reqid'";
   if (mysqli_query($conn, $sql)) {
        $msg="You have Rejected the request.";
        header("location:../bloodrequest.php?msg=".$msg );
   } else {
   $error= "Error changing status: " . mysqli_error($conn);
   header("location:../bloodrequest.php?error=".$error );
   }mysqli_close($conn);
?>
```