



Inspiring Excellence

**CSE370 : Database Systems  
Project Report  
Project Title : RedConnect**

## **Table of Contents**

<b>Section No</b>	<b>Content</b>	<b>Page No</b>
1	Introduction	03
2	Project Features	02-04
3	ER/EER Diagram	05
4	Schema Diagram	05
5	Frontend Development	06-14
6	Backend Development	15-25
7	Source Code Repository	26
8	Conclusion	26
9	References	26

## **Introduction**

Bangladesh has an urgent need for all blood types every day, especially rare blood groups such as O Negative, B Negative, and A Negative, which are highly needed. As hospitals grow, there is an increased need for blood. In Bangladesh the demand for blood stands at 3 to 3.5 lakh bags per year. That amounts to around 822 to 959 bags of blood a day, used mainly for major surgeries, delivery cases, kidney ailments, and other conditions that require transfusions, including dengue, severe anaemia, blood cancer, thalassaemia, and haemophilia (Md et al., 2024). There are many organisations like BDRCS, Badhan, and Lab One Blood Donor Club, and many Facebook groups working on supplying blood, but from our experience, all the organisations' donation processes are quite time-consuming. Most of the cases of receivers who need blood are waiting more time for a response from donors. So I built this RedConnect where receivers don't need to wait to search for donors. He/she can easily access donor information, especially the most powerful tool, mobile number, as a communication medium.

## **Project Features**

For recipient:

### **1. Filtering donors according to city and blood group.**

- The recipient can search donor filters with city and blood groups.

### **2. Table of all registered donors information.**

- The recipient can see all registered donors' information (name, city, blood group, phone number).

### **3. Register in the RedConnect Database.**

- A recipient can also be a donor and register to the RedConnect database.

For donors:

### **4. Register in the RedConnect Database.**

- A donor must register in the RedConnect database to become a donor, and it is mandatory.

## **5. Sign in.**

- A donor can sign in in the system using his phone number and password.

## **6. Dedicated donor's dashboard.**

- A dedicated dashboard for donors where donors can see their donation eligibility status. If one donor is both eligible and available, then he or she can donate blood. If not eligible, he or she can't donate blood and also see a days countdown when he/she will be available for the next donation. Besides that, all donors can see their total donation in this dashboard.

## **7. Update profile information and change password.**

- After creating an account in RedConnect, a donor can update his/her information (phone number, city, blood group, and availability status) at any time. He/she can change the account's password.

## **8. Filtering donors according to city and blood group.**

- A donor can also be a recipient. So he or she can search donors, filtering with the city and blood group for accessing information faster.

## **9. Table of all registered donors information.**

- A donor can also see all other registered donors' information (name, city, blood group, phone number).

For admin:

## **10. Sign-in portal for admin.**

## **11. Dedicated admin dashboard.**

- Admin can see the total numerical number of each blood group registered donor.

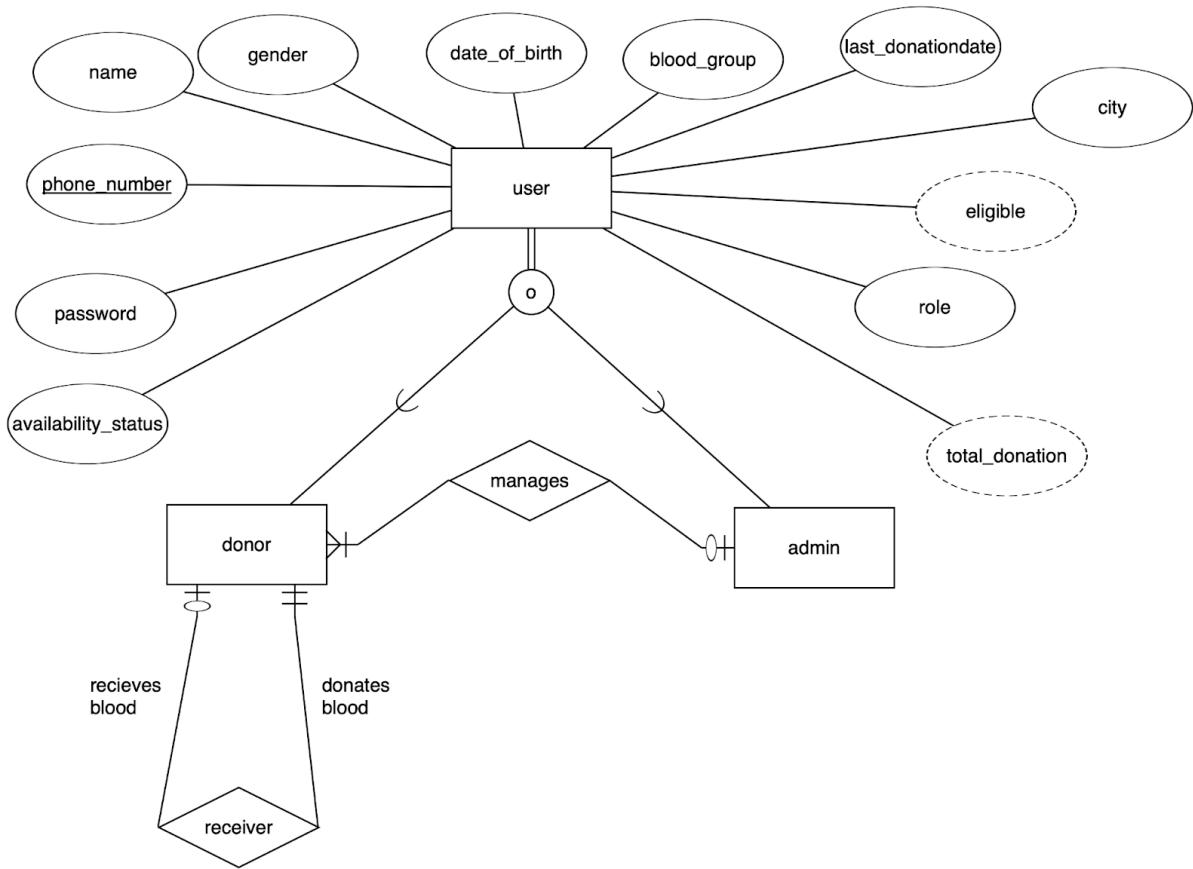
## **12. Update donors information.**

- Admin can update all information of all registered donors.

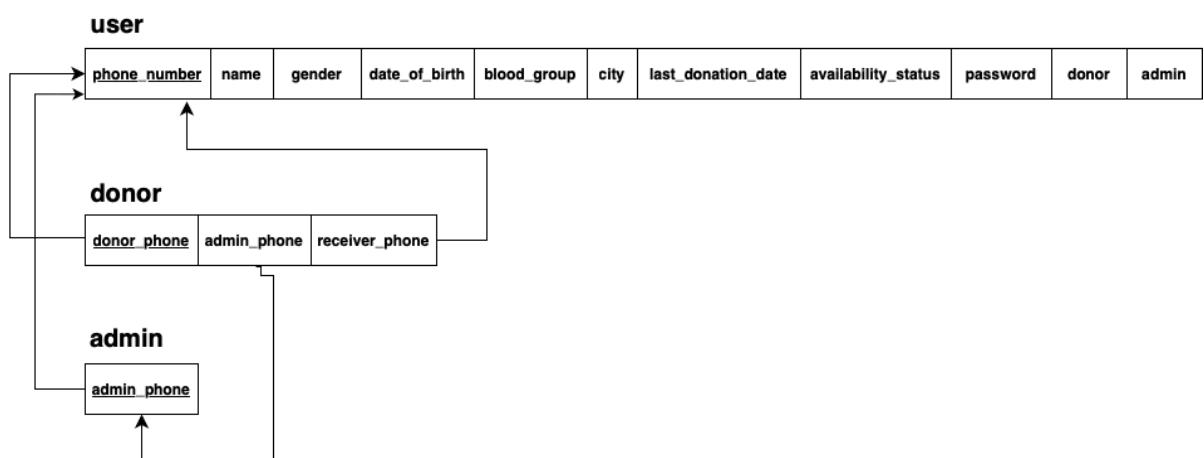
## **13. Delete donors.**

- Admin can delete any donor from the database.

## ER/EER Diagram



## Schema Diagram



## Frontend Development

Use HTML5, CSS, and the Bootstrap CSS framework for developing the frontend.

Before starting development, I set our project's basic colour code and web page background image and many conventions so everyone can follow this same thing when they are doing their part. Besides that, I built three top navigation bars using HTML and the CSS framework Bootstrap. One for the users who are not signed in and the other one for the logged-in user, basically for the donors. And the last one for the admin view.



Figure 01: Navigation bar for three different users (recipient, donor, admin)

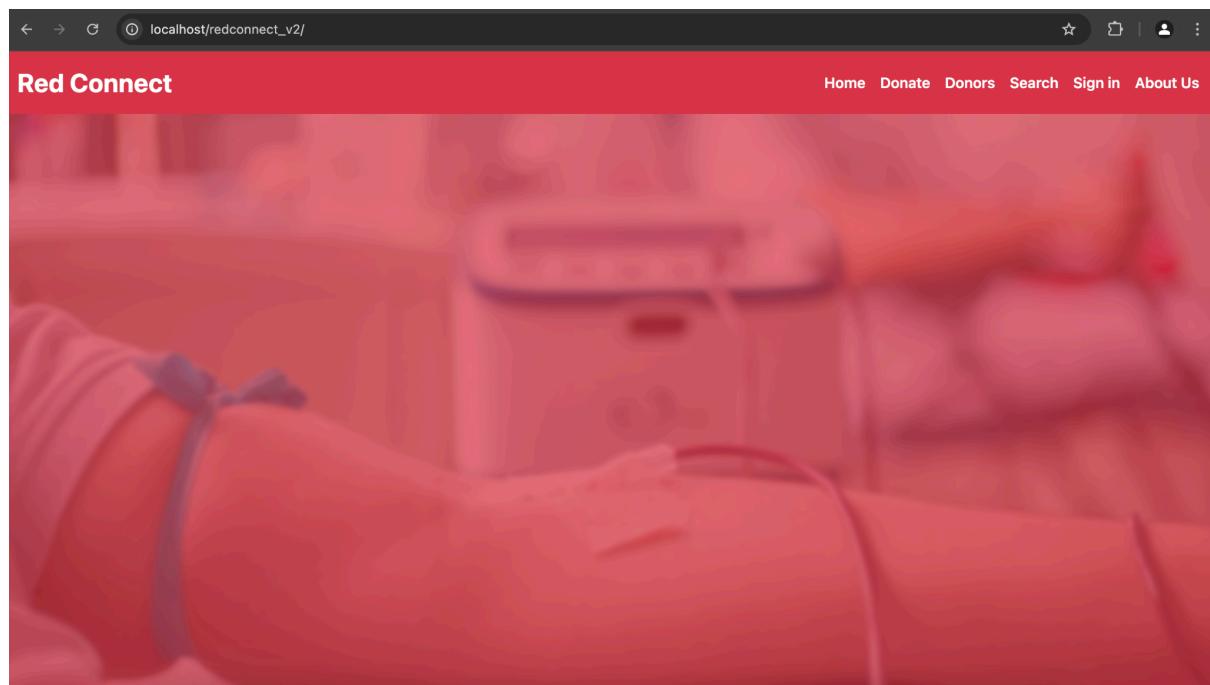


Figure 02: Background image

## 1. After login success, welcome message page for the donor.

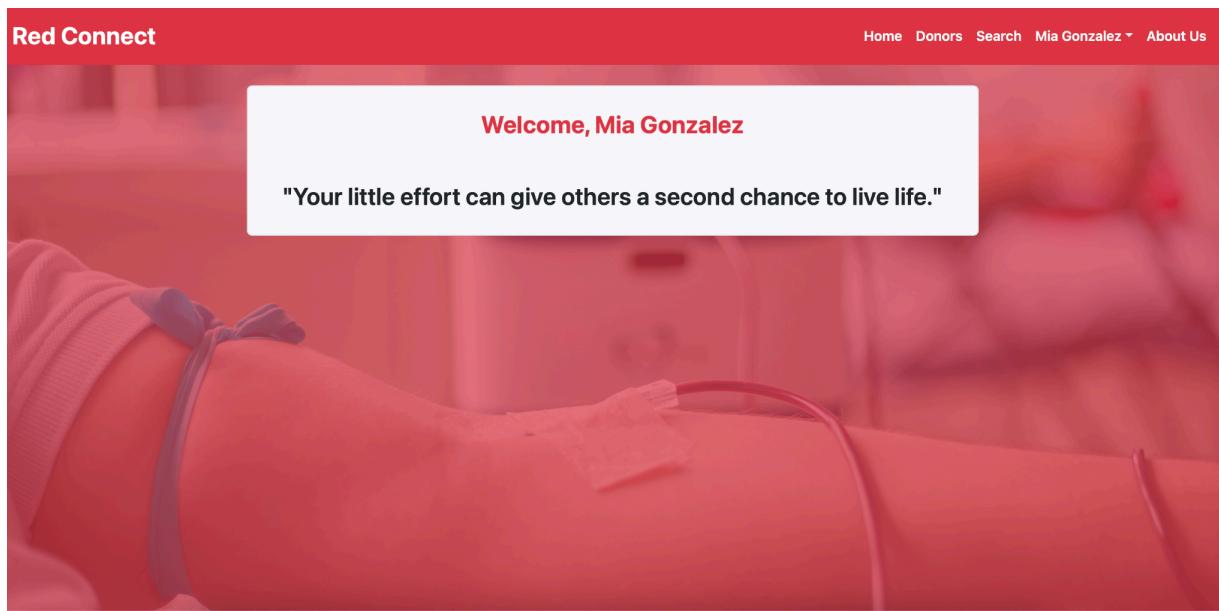


Figure 02: Welcome message for the donor

**2. Donors' Dashboard:** There are 3 views for three different cases. One for those who are eligible and now can donate the blood, one for the eligible but not available donors, and another one for those who are not eligible, where the donor can see the remaining days for the next donation. But there is one part common for all three cases where the donor can see the total donation counter box.

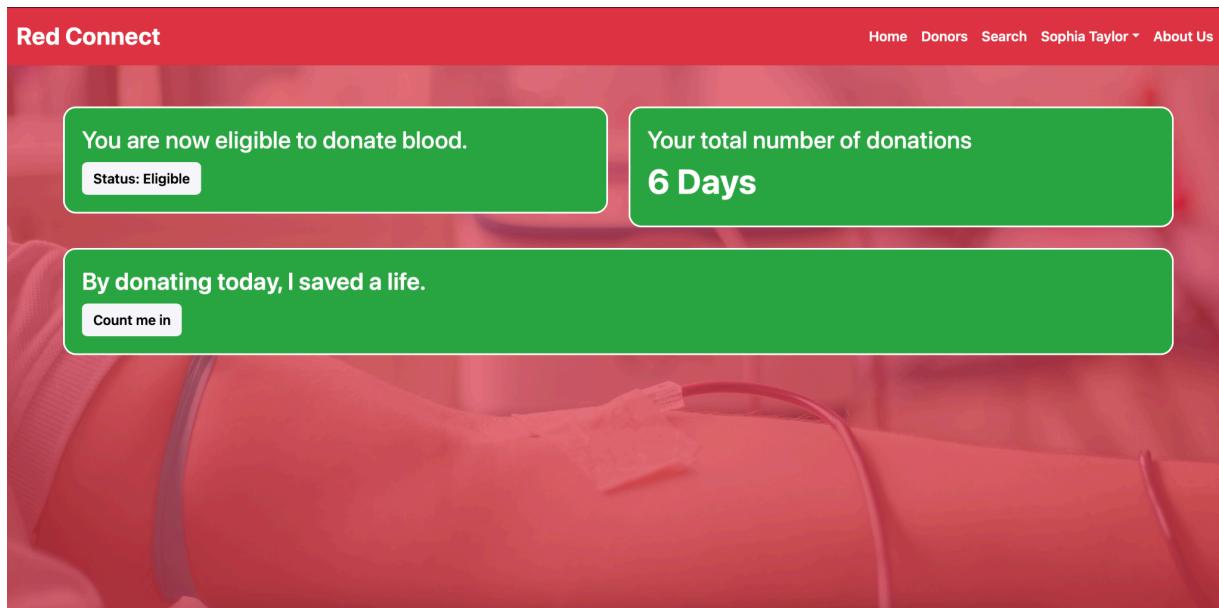


Figure 03: Dashboard view for eligible and available donors

When a donor donates blood, he/she has to click the button **Count me in** for updating the profile.

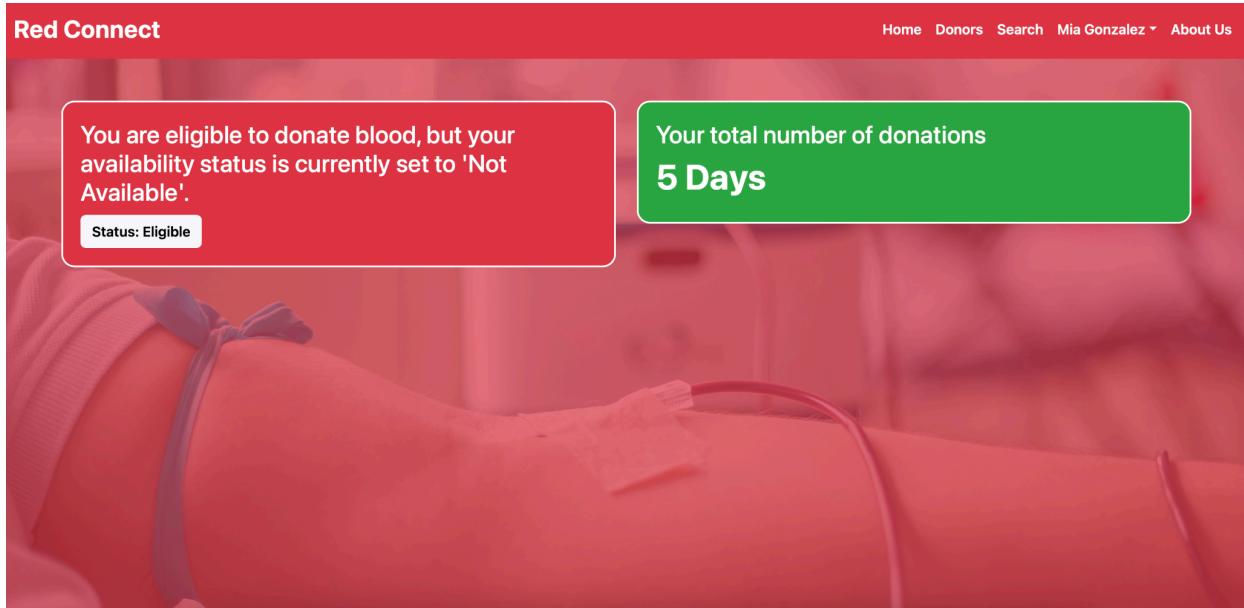


Figure 04: Dashboard view for eligible but not available donors

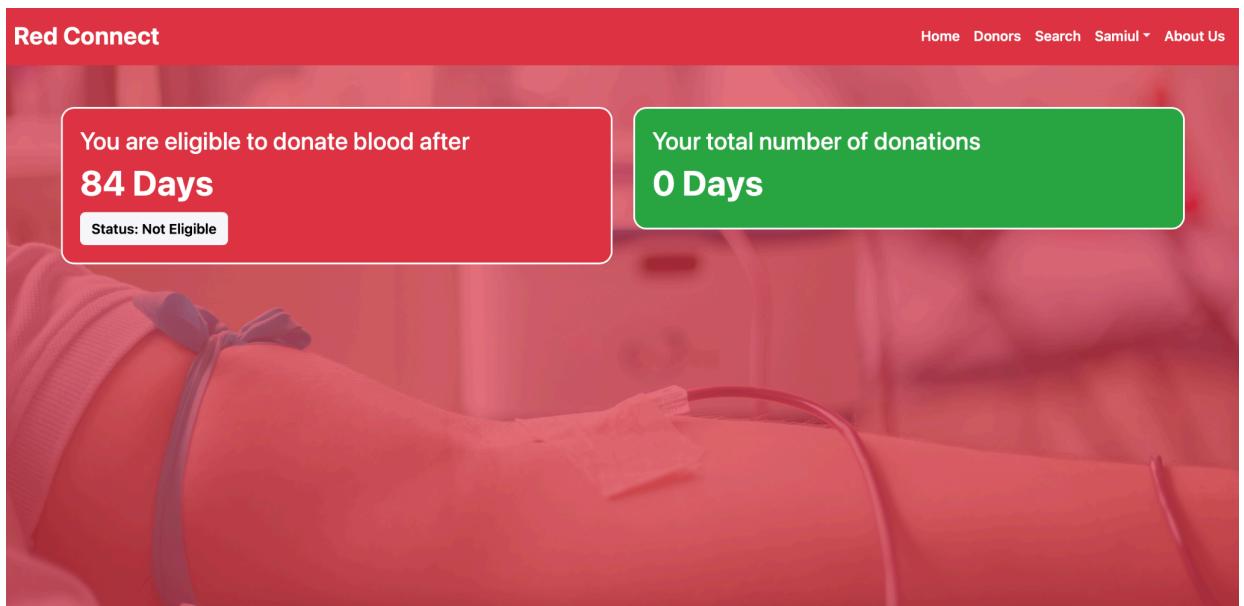


Figure 05: Dashboard view for not eligible donors

### 3. Admin Dashboard:

Admin sign-in form where admin can log in to the dashboard.

The image shows an 'Admin' sign-in form. It has two input fields: 'Username' containing 'Secret username' and 'Password' containing a series of asterisks. Below the fields is a red 'Sign in' button. At the bottom right of the form is a red link that reads 'Return to the home page -->'. The entire form is set against a light gray background.

Figure 06: Admin sign-in form

After logging in with the correct username and password, the admin can see the numerical value count for each blood group registered donor.

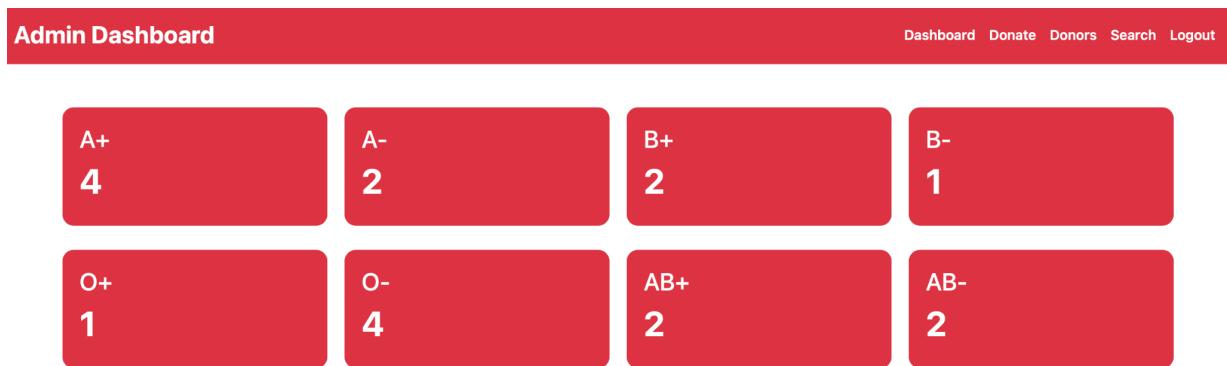


Figure 07: Admin dashboard

**4. Registered donors information:** Admin can see all registered donors information in tabular format, just like in the same MySQL database. Where the admin can update any donor's information and delete any donors from the main database. For a faster process to access data, here I use a search box for finding the donors using their unique phone number.

## All Registered Donors

Search using a phone number.

**Search**

Name	Gender	DOB	Blood Group	Phone Number	Password	City	Last Donation Date	Availability Status	Total Donation	Eligibility Status	Update	Delete
John Doe	Male	1990-05-15	O+	1234567890	password123	Dhaka	2024-01-15	Yes	5	Yes		
Jane Smith	Female	1985-08-20	A+	9876543211	securePass	Chittagong	2023-10-20	No	10	No		
Robert Brown	Male	1992-03-10	B-	4567891234	pass@word	Rajshahi	2024-05-10	Yes	2	Yes_No		
Emily Davis	Female	1988-07-12	O-	5678901245	wordPass	Khulna	2023-12-25	Yes	8	Yes		
David Wilson	Male	1994-11-05	AB+	6789012356	admin123	Sylhet	2023-12-01	No	3	No		

James White	Male	1995-01-25	O+	8901234578	password456	Rajbari	2023-08-30	No	4	No		
Olivia Harris	Female	1993-04-17	B+	9012345689	Olivia123	Comilla	2024-06-12	Yes	7	Yes_No		
William Clark	Male	1987-10-09	AB-	2345678907	pass@123	Feni	2024-03-05	No	9	Yes_No		
Ava Lewis	Female	1996-06-22	O-	3456789018	securePassword	Pabna	2023-07-14	Yes	12	Yes		
Ethan Walker	Male	1982-02-01	A+	4567890129	Ethan@123	Noakhali	2024-05-10	No	11	Yes		
Isabella Robinson	Female	1989-12-05	B-	5678901258	passWord321	Pirojpur	2024-03-28	Yes	6	Yes_No		
Liam Martinez	Male	1997-03-22	AB+	6789012379	Liam@123	Khulna	2023-09-30	Yes	13	Yes		
Mia Gonzalez	Female	1994-07-14	O+	7890123478	MiaPass	Moulvibazar	2024-06-12	No	5	Yes_No		
Noah Perez	Male	1991-11-30	A-	8901234590	Noah2024	Bagerhat	2023-06-10	Yes	7	No		
Charlotte Young	Female	1986-02-16	B+	9012345701	Charlotte@123	Rangamati	2024-01-05	Yes	3	Yes_No		

Figure 08: All registered donors information

For updating a donor's information, the admin has to click the edit icon. After that, the admin can see the donor's previous information in a light font and also update any information as the admin wants.

## Update Information

<b>Full Name</b> Liam Martinez  <input type="text" value="Enter your full name"/>	<b>Date of Birth</b> 1997-03-22  <input type="text" value="01/01/2025"/>
<b>Gender</b> Male  <input type="text" value="Choose..."/>	<b>City</b> Khulna  <input type="text" value="Choose..."/>
<b>Blood Group</b> AB+  <input type="text" value="Choose..."/>	<b>Availability Status</b> Yes  <input type="text" value="Choose..."/>
<b>Phone Number</b> 6789012379  <input type="text" value="Enter your phone number"/>	<b>Total Donation</b> 13  <input type="text" value="Total donation count"/>
<b>Last Donation Date</b> 2023-09-30  <input type="text" value="01/01/2025"/>	<b>Password</b> Liam@123  <input type="text" value="Create a password"/>
<input type="button" value="Update"/>	

Figure 09: Update the donor's information form

After clicking the cross icon, the admin can delete any donors. For cross checking, the admin has to confirm again to complete the delete process.

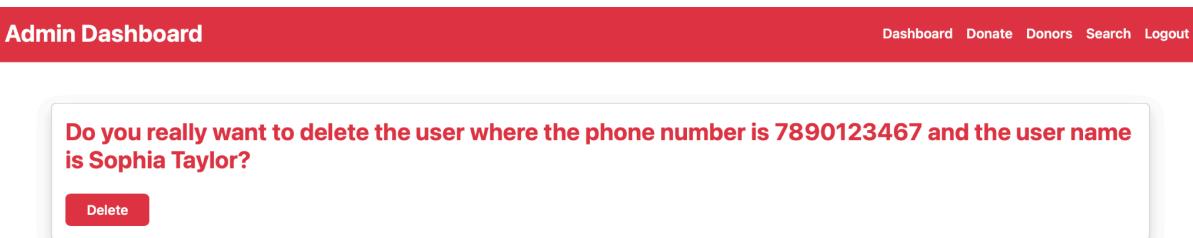


Figure 10: Delete donor from database

**5. Donor's registration form:** If anyone wants to donate through our system, he/she needs to register in the RedConnect.

"Be a hero, give the gift of life-donate blood today!"  
Complete the form to begin!

Full Name	Date of Birth
Enter your full name	01/01/2025
Gender	City
Choose...	Choose...
Blood Group	Availability Status
Choose...	Choose...
Phone Number	Password
Enter your phone number	Create a password
Last Donation Date	<input type="checkbox"/> I agree to the terms and conditions
01/01/2025	<input type="button" value="Submit"/>

Figure 12: Donor's registration form

**6. Filtering donors according to city and blood group:** Anyone can search donors by filtering city and blood groups for efficient searching. If no donor is found, the show shows no results found; otherwise, show all donors names, cities, blood groups, and phone numbers.

Search for Heroes by City and Blood Type

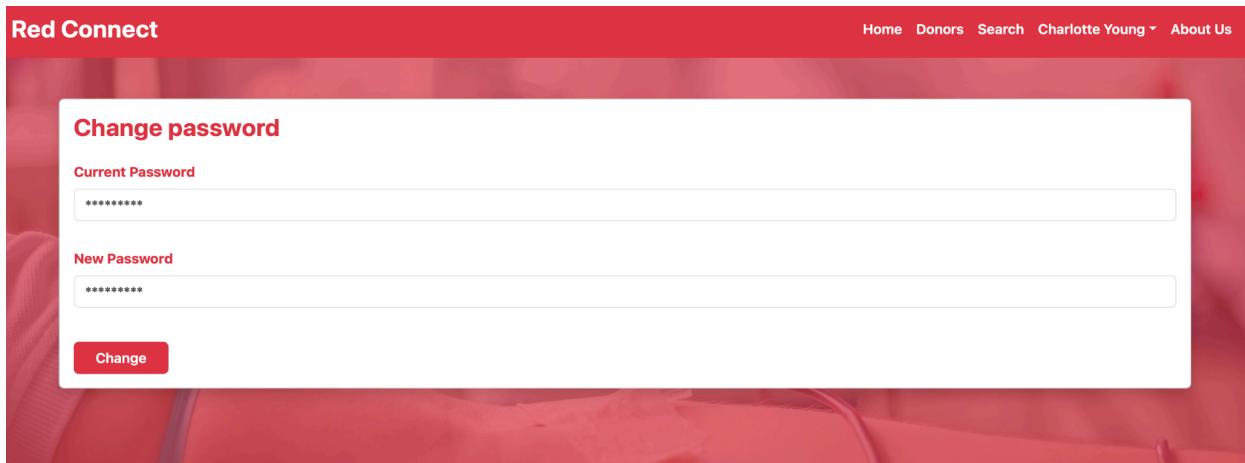
Name	City	Blood Group	Phone Number
Ella Mitchell	Satkhira	A+	5678901269

Search for Heroes by City and Blood Type

No records found.

Figure 13: Filtering donors search form

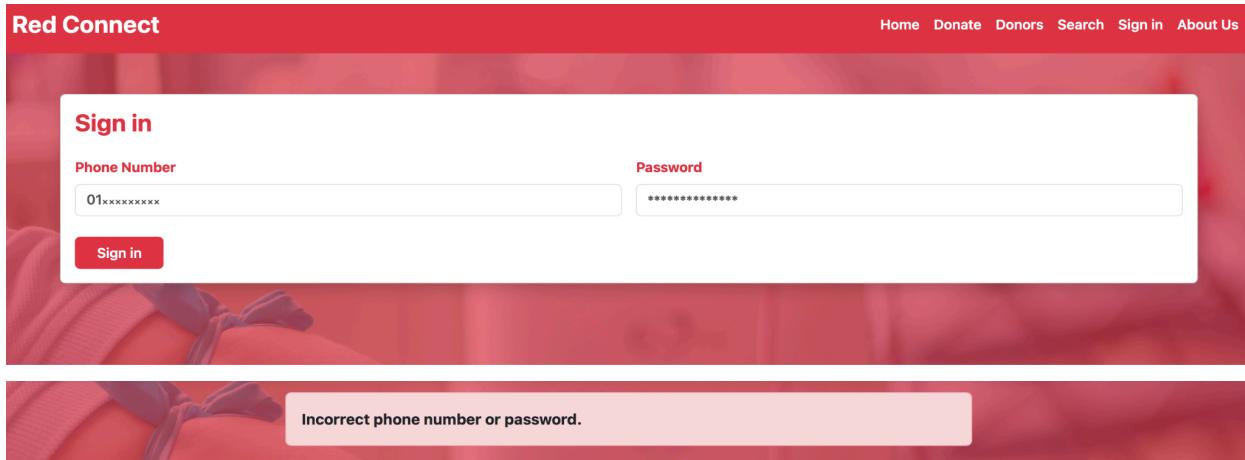
**7. Password changing form:** The donor can change the password by filling out this form. If any error is found, also show the error message.



The screenshot shows the 'Change password' form on the Red Connect website. At the top, there is a red header bar with the 'Red Connect' logo and a navigation menu containing 'Home', 'Donors', 'Search', 'Charlotte Young ▾', and 'About Us'. Below the header is a large, semi-transparent background image of a person wearing a red shirt with a bow tie. The main form area has a white background and a thin gray border. It contains two input fields: 'Current Password' and 'New Password', both represented by a series of asterisks. A red 'Change' button is located at the bottom left of the form.

Figure 14: Password changing form

**8. Donor's sign-in form:** The donor can sign in to RedConnect using a phone number and password. If any credentials do not meet the requirement, then show an error message.



The screenshot shows the 'Sign in' form on the Red Connect website. The layout is similar to Figure 14, with a red header bar and a background image of a person in a red shirt. The form itself has a white background and a thin gray border. It features two input fields: 'Phone Number' (containing '01xxxxxxx') and 'Password' (represented by asterisks). A red 'Sign in' button is positioned below the fields. At the bottom of the page, a pink horizontal bar displays the error message 'Incorrect phone number or password.'

Figure 15: Donor's sign-in form

**9. Donor's update information:** By filling out this form, the donor can update his/her information.

**Red Connect**

Home Donors Search Charlotte Young ▾ About Us

**Update Information**

**Phone Number**  
01xxxxxxxx

**Blood Group**  
Choose...

**City**  
Choose...

**Availability Status**  
Choose...

**Update**

Figure 16: Donor's update information form

**10. All registered donors information:** Here anyone can see all registered donors information in tabular format. Donor's name, gender, city, blood group, phone number.

**Red Connect**

Home Donors Search Charlotte Young ▾ About Us

**All Registered Donors**

Name	Gender	Blood Group	Phone Number	City
John Doe	Male	O+	1234567890	Dhaka
Emily Davis	Female	O-	5678901245	Khulna
Sophia Taylor	Female	A-	7890123467	Faridpur
Ava Lewis	Female	O-	3456789018	Pabna
Ethan Walker	Male	A+	4567890129	Noakhali
Liam Martinez	Male	AB+	6789012379	Khulna
Lucas Scott	Male	O-	2345678908	Feni
Amelia Adams	Female	AB-	3456789021	Madaripur
Ella Mitchell	Female	A+	5678901269	Satkhira
Harper Lee	Female	A+	9012345702	Bagerhat
Jackson Perez	Male	B+	2345678909	Bandarban
Evelyn Wilson	Female	B-	5678901270	Moulvibazar

Figure 17: All registered donors information table

## Backend Development

Use PHP for our backend.

### 1. Donor's Dashboard:

- A. Session Start: Start a session and include the database connection file (DBconnect.php).
- B. Retrieve Current User Phone Number: Get the logged-in user's phone number from the session variable.
- C. Fetch User Details: Query the donors table using the phone number to fetch the user's details (e.g., name, total donations, last donation date, and availability status).
- D. Calculate Days Since Last Donation:
  - If the last donation date is "0000-00-00", set date\_remain to 0.
  - Otherwise, calculate the difference in days between the current date and the last donation date.
  - If the difference is less than 90 days, calculate the remaining days until eligibility (date\_remain = 90 - total\_days).
  - If the difference is 90 days or more, set date\_remain to 0.
- E. Determine Eligibility and Update Database:
  - If date\_remain > 0: Mark the user as "not eligible" for blood donation and update the eligible field in the database to 'No'.
  - If date\_remain == 0 and availability status is 'Yes': Mark the user as "eligible" and update the eligible field to 'Yes'.
  - If date\_remain == 0 and availability status is 'No': Mark the user as "eligible but not available" and update the eligible field to 'Yes\_No'.
- F. Set Status: Assign a corresponding status (not\_eligible, eligible, eligible\_but\_not\_available) based on the above conditions.

```

<?php
session_start();
require_once('DBconnect.php');

$phone_number = $_SESSION['current_user_phone_number'];

$sql = "SELECT * FROM donors WHERE phone_number = '$phone_number'";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    $user = mysqli_fetch_assoc($result);
}

$name = $user['name'];
$total_donation = $user['total_donation'];
$last_donation_date = $user['last_donation_date'];
$availability_status = $user['availability_status'];

if ($last_donation_date == "0000-00-00") {
    $date_remain = 0;
} else {
    $today = new DateTime();
    $last_donation = new DateTime($last_donation_date);
    $interval = $today->diff($last_donation);

    $total_days = $interval->days;

    if ($total_days < 90) {
        $date_remain = 90 - $total_days;
    } else {
        $date_remain = 0;
    }
}

if ($date_remain > 0) {
    // $status = "Not Eligible for donating blood";
    $status = "not_eligible";
    $updateSql = "UPDATE donors SET eligible = 'No' WHERE phone_number = '$phone_number'";
    mysqli_query($conn, $updateSql);

} elseif ($availability_status == 'Yes') {
    // $status = "Eligible for donating blood";
    $status = "eligible";
    $updateSql = "UPDATE donors SET eligible = 'Yes' WHERE phone_number = '$phone_number'";
    mysqli_query($conn, $updateSql);

} elseif ($availability_status == 'No') {
    // $status = "You are eligible to donate blood, but your availability status is currently set to 'Not Available.'";
    $status = "eligible_but_not_available";
    $updateSql = "UPDATE donors SET eligible = 'Yes_No' WHERE phone_number = '$phone_number'";
    mysqli_query($conn, $updateSql);
}

?>

```

Figure 18: Donor's dashboard PHP script

## 2. Admin sign-in form and dashboard:

- A. Include Database and Start Session: Load the database connection file and start a session.
- B. Check Request Method: Proceed only if the request method is POST.
- C. Retrieve Input: Capture the username and password from the form submission.
- D. Validate Admin Credentials:
  - If the username is team\_redconnect and the password is redconnect@, redirect the user to admin\_dashboard.php and terminate the script.
  - Otherwise, set an error message indicating incorrect credentials.

```
<?php
require_once('DBconnect.php');
session_start();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $user_name = $_POST['username'];
    $password = $_POST['password'];

    if ($user_name == 'team_redconnect' && $password == 'redconnect@'){
        header("Location: admin_dashboard.php");
        exit();
    } else {
        $error = "Incorrect username or password.";
    }
}
?>
```

Figure 19: Admin sign-in PHP script

- A. Include Database and Start Session: Load the database connection file and start a session.
- B. SQL Queries: Prepare SQL queries to count eligible donors for each blood group (A+, A-, B+, B-, O+, O-, AB+, AB-) with eligible = 'Yes'.
- C. Execute Queries: Run each query using mysqli\_query() and store the result.
- D. Fetch Counts: Use mysqli\_fetch\_assoc() to extract the count for each blood group and store them in respective variables.

```

<?php
require_once('DBconnect.php');
session_start();

$a_pos_sql = "SELECT count(*) AS count FROM donors WHERE blood_group = 'A+' AND eligible = 'Yes'";
$a_neg_sql = "SELECT count(*) AS count FROM donors WHERE blood_group = 'A-' AND eligible = 'Yes'";
$b_pos_sql = "SELECT count(*) AS count FROM donors WHERE blood_group = 'B+' AND eligible = 'Yes'";
$b_neg_sql = "SELECT count(*) AS count FROM donors WHERE blood_group = 'B-' AND eligible = 'Yes'";
$o_pos_sql = "SELECT count(*) AS count FROM donors WHERE blood_group = 'O+' AND eligible = 'Yes'";
$o_neg_sql = "SELECT count(*) AS count FROM donors WHERE blood_group = 'O-' AND eligible = 'Yes'";
$ab_pos_sql = "SELECT count(*) AS count FROM donors WHERE blood_group = 'AB+' AND eligible = 'Yes'";
$ab_neg_sql = "SELECT count(*) AS count FROM donors WHERE blood_group = 'AB-' AND eligible = 'Yes';

$a_pos_query = mysqli_query($conn, $a_pos_sql);
$a_neg_query = mysqli_query($conn, $a_neg_sql);
$b_pos_query = mysqli_query($conn, $b_pos_sql);
$b_neg_query = mysqli_query($conn, $b_neg_sql);
$o_pos_query = mysqli_query($conn, $o_pos_sql);
$o_neg_query = mysqli_query($conn, $o_neg_sql);
$ab_pos_query = mysqli_query($conn, $ab_pos_sql);
$ab_neg_query = mysqli_query($conn, $ab_neg_sql);

$count_a_pos = mysqli_fetch_assoc($a_pos_query);
$count_a_neg = mysqli_fetch_assoc($a_neg_query);
$count_b_pos = mysqli_fetch_assoc($b_pos_query);
$count_b_neg = mysqli_fetch_assoc($b_neg_query);
$count_o_pos = mysqli_fetch_assoc($o_pos_query);
$count_o_neg = mysqli_fetch_assoc($o_neg_query);
$count_ab_pos = mysqli_fetch_assoc($ab_pos_query);
$count_ab_neg = mysqli_fetch_assoc($ab_neg_query);

?>

```

Figure 20: Admin dashboard PHP script

### 3. Registered donors information (edit and delete operation)

#### A. Database and Session:

- Connects to the database and starts a session.

#### B. Search or Fetch Data:

- Check if the request method is POST.
- Retrieves the search input search\_phone\_number.
- Queries the donors table for records matching the phone number.
- If records exist, fetch them into \$results. Otherwise, set \$error.

#### C. Display Data:

- Outputs donor details in a table with htmlspecialchars() for security.

#### D. Action Links:

- Includes links for updating and deleting records:
  - Update: Redirects to admin\_user\_update.php with the phone number as a parameter.
  - Delete: Redirects to admin\_user\_delete.php with the phone number as a parameter

Here I used the GET method.

```

<?php
require_once('DBconnect.php');
session_start();

if ($_SERVER['REQUEST_METHOD'] == 'POST'){
    $search_phone_number = $_POST['search_phone_number'];

    $sql = "SELECT * FROM donors WHERE phone_number = '$search_phone_number'";
    $query = mysqli_query($conn, $sql);

    if ($query && mysqli_num_rows($query) > 0) {
        $results = $query->fetch_all(MYSQLI_ASSOC);
    } else {
        $error = "Empty records.";
    }
} else{
    $sql = "SELECT * FROM donors";
    $query = mysqli_query($conn, $sql);

    if ($query && mysqli_num_rows($query) > 0) {
        $results = $query->fetch_all(MYSQLI_ASSOC);
    } else {
        $error = "Empty records.";
    }
}
?>

```

```

<?php
foreach ($results as $row){ ?>
    <tr>
        <td><?php echo htmlspecialchars($row['name']); ?></td>
        <td><?php echo htmlspecialchars($row['gender']); ?></td>
        <td><?php echo htmlspecialchars($row['date_of_birth']); ?></td>
        <td><?php echo htmlspecialchars($row['blood_group']); ?></td>
        <td><?php echo htmlspecialchars($row['phone_number']); ?></td>
        <td><?php echo htmlspecialchars($row['password']); ?></td>
        <td><?php echo htmlspecialchars($row['city']); ?></td>
        <td><?php echo htmlspecialchars($row['last_donation_date']); ?></td>
        <td><?php echo htmlspecialchars($row['availability_status']); ?></td>
        <td><?php echo htmlspecialchars($row['total_donation']); ?></td>
        <td><?php echo htmlspecialchars($row['eligible']); ?></td>
        <td>
            <?php echo '<a href="admin_user_update.php?phone_number=' . $row['phone_number'] . '"';
            </a>' ?>
        </td>
        <td>
            <?php echo '<a href="admin_user_delete.php?phone_number=' . $row['phone_number'] . '"';
            </a>' ?>
        </td>
    </tr>
<?php } ?>

```

Figure 21: Registered donors with update and delete actions PHP script

#### **4. Donor's registration form:**

- A. Database and Session:
  - Connects to the database and starts a session.
- B. Data Collection and Validation:
  - Retrieves form inputs (fullName, dob, bloodGroup, etc.).
  - Calculates donor age and days since the last donation.
- C. Eligibility Check:
  - If the last donation was < 90 days, calculate remaining days to eligibility.
  - Donors < 18 or already registered (by phone\_number) are marked ineligible with appropriate messages.
- D. Eligibility Assignment:
  - Sets eligible based on donation gap, age, and availability status.
- E. Data Insertion:
  - Inserts donor details into the database if:
    - Donor is  $\geq$  18 years.
    - Phone number is not already registered.
    - Meets eligibility criteria.
- F. Session Status Messages:
  - Tracks outcomes like success, already registered, or age-based ineligibility.

```

<?php
require_once('DBconnect.php');
session_start();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $fullName = $_POST['fullName'];
    $gender = $_POST['gender'];
    $dob = $_POST['dob'];
    $bloodGroup = $_POST['bloodGroup'];
    $phoneNumber = $_POST['phoneNumber'];
    $city = $_POST['city'];
    $lastDonationDate = $_POST['lastDonationDate'];
    $availabilityStatus = $_POST['availabilityStatus'];
    $password = $_POST['password'];
    $terms = $_POST['terms'];

    $todayDate = date("Y-m-d");
    $todayObj = new DateTime($todayDate);
    $birthDateObj = new DateTime($dob);
    $lastDonationDateObj = new DateTime($lastDonationDate);

    // eligibility_check
    if ($lastDonationDate == "") {
        $date_remain = 0;
    } else {
        $difference = $todayObj->diff($lastDonationDateObj);
        $total_days = $difference->days;

        if ($total_days < 90) {
            $date_remain = 90 - $total_days;
        } else {
            $date_remain = 0;
        }
    }

    $interval = $todayObj->diff($birthDateObj);
    $years = $interval->y;
}

```

```

$years = $interval->y;

$checkPhoneSql = "SELECT * FROM donors WHERE phone_number = '$phoneNumber'";
$checkPhoneSqlResult = mysqli_query($conn, $checkPhoneSql);
$affectedRow = mysqli_num_rows($checkPhoneSqlResult);

if ($years>=18 and $affectedRow==0){
    $_SESSION['status'] = "Your details have been successfully saved in our database.";

    if (0<$date_remain && $date_remain<90) {
        $eligible = "No";
    } elseif ($date_remain>=90 or $date_remain==0) {
        // etto aghe dichi
        if ($availabilityStatus == 'Yes'){
            $eligible = "Yes";
        }else{
            $eligible = "Yes_No";
        }
    }

    $eligible_after_days = $date_remain;
    $sql = "INSERT INTO donors VALUES ('$fullName', '$gender', '$dob', '$bloodGroup', '$phoneNumber', '$city',
        '$lastDonationDate', '$availabilityStatus', '$password', '$eligible')";
    $result = mysqli_query($conn, $sql);
}
elseif ($affectedRow>0) {
    $_SESSION['status'] = "Your phone number is already registered.";
}
elseif ($years<18) {
    $_SESSION['status'] = "You are ineligible to donate blood as you are under 18.";
}
}?

```

Activate Windows  
Go to Settings to activate

Figure 22: Donor's registration form PHP script

## 5. Filtering donors according to city and blood group:

### A. Database and Session:

- Connects to the database and starts a session.

### B. Handle POST Request:

- Retrieves form inputs: city and blood\_group.
- Queries the donors table for eligible donors matching the city and blood group.

### C. Results Handling:

- If matching records are found, store them in \$results.
- If no records are found, set an error message: "No records found."

```

<?php
require_once('DBconnect.php');
session_start();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $city = $_POST['city'];
    $blood_group = $_POST['blood_group'];

    $sql = "SELECT name, city, blood_group, phone_number FROM donors WHERE city = '$city' AND blood_group = '$blood_group' AND eligible = 'Yes'";
    $query = mysqli_query($conn, $sql);

    if ($query && mysqli_num_rows($query) > 0) {
        $results = $query->fetch_all(MYSQLI_ASSOC);
    } else {
        $error = "No records found.";
    }
}
?>

```

Figure 23: Search donors PHP script

## 6. Password change:

- A. Session and Database: Starts a session and connects to the database.
- B. Password Change: Validates the current password.
- C. Update Password: If valid, updates to the new password and redirects to user\_dashboard.php. If not, set an error status ('mismatch').

```

<?php
session_start();
require_once('DBconnect.php');

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $currentPassword = $_POST['current_password'];
    $newPassword = $_POST['new_password'];

    $phone_number = $_SESSION['current_user_phone_number'];

    $sql = "SELECT * FROM donors WHERE phone_number = '$phone_number' AND password = '$currentPassword'";
    $result = mysqli_query($conn, $sql);

    if (mysqli_num_rows($result) > 0){
        $updateSQL = "UPDATE donors SET password = '$newPassword' WHERE phone_number = '$phone_number'";
        mysqli_query($conn, $updateSQL);
        header("Location: user_dashboard.php");
    } else{
        $_SESSION['status'] = 'mismatch';
    }
}
?>

```

Figure 24: Change password PHP script

## 7. Donor's sign-in form:

- A. Session and Database: Starts a session and connects to the database.
- B. Login Process: Validates the phone number and password.
- C. Successful Login: If credentials match, sets session variables and redirects to home.php.
- D. Error Handling: If credentials don't match, set an error message.

```
<?php
require_once('DBconnect.php');
session_start();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $phone_number = $_POST['phone_number'];
    $password = $_POST['password'];

    $sql = "SELECT * FROM donors WHERE phone_number = '$phone_number' AND password = '$password'";
    $query = mysqli_query($conn, $sql);

    if ($query && mysqli_num_rows($query) > 0) {
        $_SESSION['state'] = true;
        $_SESSION['current_user_phone_number'] = $phone_number;
        header("Location: home.php");
        exit();
    } else {
        $error = "Incorrect phone number or password.";
    }
}
?>
```

Figure 25: Donor's sign-in PHP script

## 8. All registered donors information:

- A. Database Query: Selects eligible donors' details (name, gender, blood group, phone number, city).
- B. Results Handling: If records are found, store them in \$results. Otherwise, it sets an error message ("Empty records.").

```
<?php
require_once('DBconnect.php');
$sql = "SELECT name, gender, blood_group, phone_number, city FROM donors WHERE eligible = 'Yes'";
$query = mysqli_query($conn, $sql);

if ($query && mysqli_num_rows($query) > 0) {
    $results = $query->fetch_all(MYSQLI_ASSOC);
} else {
    $error = "Empty records.";
}
?>
```

Figure 26: All registered donors PHP script

## 9. Update donor's information:

- A. Session and Database: Starts a session and connects to the database.
- B. Form Data Handling: Retrieves the form inputs: phone\_number, bloodGroup, city, and availabilityStatus.
- C. Update Process:
  - If the current phone number matches, update the donor's details.
  - If the phone number is different, check if it already exists in the database.
- D. Phone Number Validation:
  - If the new phone number exists, set a session status ('exist').
  - If not, update the details and redirects to the user\_dashboard.php.
- E. Session Update: Updates the session with the new phone number after the update.

```
<?php
session_start();
require_once('DBconnect.php');

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $phoneNumber = $_POST['phone_number'];
    $bloodGroup = $_POST['bloodGroup'];
    $city = $_POST['city'];
    $availabilityStatus = $_POST['availabilityStatus'];
    $phone_number = $_SESSION['current_user_phone_number'];
    if ($phone_number == $phoneNumber){
        $sql = "UPDATE donors SET phone_number = '$phoneNumber', blood_group = '$bloodGroup', city = '$city',
            availability_status = '$availabilityStatus' WHERE phone_number = '$phone_number'";
        $result = mysqli_query($conn, $sql);
        $_SESSION['current_user_phone_number'] = $phoneNumber;
        header("Location: user_dashboard.php");
    } else{
        $checkPhoneSql = "SELECT * FROM donors WHERE phone_number = '$phoneNumber'";
        $checkPhoneSqlResult = mysqli_query($conn, $checkPhoneSql);
        if (mysqli_num_rows($checkPhoneSqlResult) > 0){
            $_SESSION['status'] = "exist";
        } else{
            $sql = "UPDATE donors SET phone_number = '$phoneNumber', blood_group = '$bloodGroup', city = '$city',
                availability_status = '$availabilityStatus' WHERE phone_number = '$phone_number'";
            $result = mysqli_query($conn, $sql);
            $_SESSION['current_user_phone_number'] = $phoneNumber;
            header("Location: user_dashboard.php");
        }
    }
}
?>
```

Figure 27: Update donor's information PHP script

## **Source Code Repository**

### **Drive link:**

<https://drive.google.com/drive/folders/1vXhmE30-yb4lBIxUvrpjRmEe3sau6Yb?usp=sharing>

### **GitHub repository:**

<https://github.com/turtle-shifa/RedConnect---Automated-Blood-Management-System>

## **Conclusion**

Successfully built this project that all of us were dreaming of. I Believe that if I Launch this project in the market, I can serve many people. But it is also true that there are many things that should be fixed to expand its efficiency. I also have some extra features in mind, and hopefully I will build this in the near future and help people in Bangladesh by ensuring faster and safer blood.

## **References**

Md, S., Rupa, R., Chowdhury, M. H., Hasina, M., & Rakibul, H. (2024). Blood donation systems in Bangladesh: Problems and remedies. *International Journal of Business and Management*, 14(8), 145-145. <https://doi.org/10.5539/ijbm.v14n8p145>