

# Design and Implementation of Blood Donation Hub

by

**Jebinur Akter**

ID: CSE1901016053

**Mehrin Zihan Khan**

ID: CSE1901016050

Supervised by

**Bulbul Ahamed**

Submitted in partial fulfillment of the requirements for the degree of  
Bachelor of Science in Computer Science and Engineering



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
SONARGAON UNIVERSITY (SU)**

January 2023

# Design and Implementation of Blood Donation Hub

by

**Jebinur Akter**

ID: CSE1901016053

**Mehrin Zihan Khan**

ID: CSE1901016050

Supervised by

**Bulbul Ahamed**

Submitted in partial fulfillment of the requirements for the degree of  
Bachelor of Science in Computer Science and Engineering



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
SONARGAON UNIVERSITY (SU)**

January 2023

# APPROVAL

The project titled “**Design and Implementation of Blood Donation Hub**” submitted by Jebinur Akter (CSE1901016053) and Mehrin Zihan Khan(CSE1901016050) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

## Board of Examiners

-----  
**Bulbul Ahamed**

Associate Professor & Head  
Department of Computer Science and Engineering  
Sonargaon University (SU)

**Supervisor**

-----  
(Examiner Name & Signature)

Department of Computer Science and Engineering  
Sonargaon University (SU)

**Examiner 1**

-----  
(Examiner Name & Signature)

Department of Computer Science and Engineering  
Sonargaon University (SU)

**Examiner 2**

-----  
(Examiner Name & Signature)

Department of Computer Science and Engineering  
Sonargaon University (SU)

**Examiner 3**

# DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Bulbul Ahamed, Associate Professor & Head,** Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

Signature

-----  
**(Bulbul Ahamed)**  
**Supervisor**

-----  
Jebinur Akter  
ID: CSE1901016053

-----  
Mehrin Zihan Khan  
ID: CSE1901016050

# **ABSTRACT**

Blood donation is a vital part of worldwide healthcare. It relates to blood transfusion as a life-sustaining and life-saving procedure as well as a form of therapeutic phlebotomy as a primary medical intervention. Over one hundred million units of blood are donated each year throughout the world. The modern world almost wholly depends on advanced technology. Healthcare is now also a part of online based world. Blood Donation Hub is basically a web-based application that has been chosen to donate and receive blood. This application is based on an online platform. It is user-friendly, simple, fast, and cost-effective which saves time and lives. This application has many features and the main feature is that receiver can find donors online for blood management. Another important feature is that receiver can send a request for blood and if the blood group matches then the donor can accept the blood request.

# ACKNOWLEDGMENT

At the very beginning, we would like to express my deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the schedule time.

We are auspicious that we had the kind association as well as supervision of **Bulbul Ahamed**, Associate Professor & Head, Department of Computer Science and Engineering, Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our project.

We would like to convey our special gratitude to **Prof. Dr. Md Alamgir Hossain**, Dean, Faculty of Science and Engineering for his kind concern and precious suggestions.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to my parents for their support, encouragement, and endless love.

# LIST OF ABBREVIATIONS

BDH	Blood Donation Hub
CSS	Cascading Style Sheets
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
PHP	Hypertext Preprocessor
SQL	Structured Query Language
UI	User Interface
XAMPP	X-operating system, Apache, Mysql, Php, Perl

# TABLE OF CONTENTS

<u>Title</u>	<u>Page No.</u>
<b>DECLARATION</b> .....	iii
<b>ABSTRACT</b> .....	iv
<b>ACKNOWLEDGEMENT</b> .....	v
<b>LIST OF ABBREVIATIONS</b> .....	vi
<b>CHAPTER 1</b>	1 – 2
<b>INTRODUCTION TO BLOOD DONATION HUB</b>	
1.1 Introduction.....	1
1.2 Motivation.....	1
1.3 Problem Statement.....	1
1.4 What is “Blood Donation Hub”?.....	2
1.5 Objectives .....	2
1.6 Chapter Review.....	2
<b>CHAPTER 2</b>	3– 4
<b>BACKGROUND OF THE PROJECT</b>	
2.1 Introduction.....	3
2.2 Background Study .....	3
2.3 Design Requirements.....	4
2.4 Challenges.....	4
<b>CHAPTER 3</b>	5 – 7
<b>TOOLS AND TECHNOLOGY</b>	
3.1 Introduction.....	5
3.2 Tools and Technology.....	5 – 7



<b>CHAPTER 4</b>	8 – 16
<b>LOGICAL SYSTEM SPECIFICATION</b>	
4.1 Introduction.....	8
4.2 Modules Overview.....	8
4.3 Entities.....	9
4.4 Attributes.....	9
4.5 Relationships.....	9
4.6 E-R Diagram.....	10
4.7 Use case Diagram.....	11
4.8 Schema Diagram.....	12
4.9 Data Flow Diagram.....	13 – 14
4.10 Database Table.....	15 – 16
<b>CHAPTER 5</b>	17 – 30
<b>USER INTERFACE AND IMPLEMENTATION</b>	
5.1 Introduction .....	17
5.2 Home Page .....	17
5.3 Registration Form Page.....	18
5.4 User Login And Admin Login Page.....	19
5.5 Receiver Profile.....	20
5.6 Blood Request Page.....	21
5.7 No Interested Donors (from Receiver Profile).....	22
5.8 Blood Managed(from Receiver Profile).....	23
5.9 Donor Profile (Status Unavailable).....	24
5.10 Donor Profile(Status Available).....	25
5.11 Donor Profile (Request Cancel).....	26
5.12 Admin Dashboard.....	27
5.13 Admin Dashboard (All Donors List) .....	28
5.14 Admin Dashboard (All Receivers List) .....	29
5.15 Admin Dashboard (All Blood Requests List).....	30

<b>CHAPTER 6</b>	31
<b>CONCLUSION, LIMITATIONS AND FUTURE WORKS</b>	
6.1 Conclusion .....	31
6.2 Limitations.....	31
6.3 Future Works.....	31
<b>REFERENCES</b> .....	32
<b>APPENDIX</b> .....	33 - 45

## **LIST OF TABLES**

<u><b>Table No.</b></u>	<u><b>Title</b></u>	<u><b>Page No.</b></u>
Table 3.1:	Donor Table.....	15
Table 3.2:	Interested Donor Table.....	15
Table 3.3:	Receiver Table .....	16
Table 3.4:	Blood Request Table.....	16
Table 3.5:	Blood Group Table.....	16

# LIST OF FIGURES

<u>Figure No.</u>	<u>Title</u>	<u>Page No.</u>
Fig 3.1:	E-R Diagram.....	10
Fig 3.2:	Use case Diagram.....	11
Fig 3.3:	Schema Diagram.....	12
Fig 3.4:	Data-flow Diagram (Level-0) .....	13
Fig 3.5:	Data-flow Diagram (Level-1) .....	13
Fig 3.6:	Data-flow Diagram (Level:2) .....	13
Fig 3.7:	Data-flow Diagram (Level:3) .....	14
Fig 3.8:	Data-flow Diagram (Level:4) .....	14
Fig 4.1:	Home Page.....	17
Fig 4.2:	Registration Form page.....	18
Fig 4.3:	User Login & Admin Login Page .....	19
Fig 4.4:	Receiver Profile.....	20
Fig 4.5:	Blood Request Page.....	21
Fig 4.6:	Receiver Profile.....	22
Fig 4.7:	Receiver Profile (Blood Managed).....	23
Fig 4.8:	Donor Profile (Status Unavailable) .....	24
Fig 4.9:	Donor Profile (Status Available) .....	25
Fig 4.10:	Donor Profile (Request Cancel) .....	26
Fig 4.11:	Admin Dashboard .....	27
Fig 4.12:	Admin Dashboard (All Donors List).....	28
Fig 4.13:	Admin Dashboard (All Receivers List).....	29
Fig 4.14:	Admin Dashboard (All Blood Requests List) .....	30

# CHAPTER 1

## INTRODUCTION TO BLOOD DONATION HUB

---

### 1.1 Introduction

Nowadays everything is switching from a desktop environment to client server architecture. Most of the application software is designed for distribution systems. That is why internet application is growing rapidly. To capture the global market through this internet all the software companies are adapting this internet technology. [1] This project has been developed for those users who are willing to get the Blood information easily by a website. By using them Blood Donation Hub can access all information over desktop, phone or tablet where internet is available. In this system receivers & donors can store and update their information easily. This system is very useful for humans because patients can die if they don't get blood in time and this system is useful for humans to prevent that death. The receiver sends the blood request and the blood request accepted by the donor.

### 1.2 Motivation

Resource sharing is the main idea of these client-server technologies. Centralized distribution system can be counted as a major turning point for this internet technology. Global authenticated access causes the most inspiration for this new technology. Net to support network or web based applications only that even most applications are coming with network support. The html and web based application follow up the rhythm of these technologies. It is used in the proposed system, to fulfill the basic requirement to the application or internet application. However, asp.net is also used in the proposed system because it is a free resource and we can develop web based applications very easily with it. JavaScript and MYSQL are free development tools which add further benefits. MYSQL and JavaScript resources, tutorial, administrative tools are also available. This software can easily get from the net and can develop powerful applications.

### 1.3 Problem Statement

The percentage of people donating blood is increasing day by day due to awareness to donate blood for those needed. The blood received have to be managed thoroughly so that there will be no negative effect to the blood receiver once they received blood. From the observations and interview conducted that have been made during the user requirements phase, it was found out that there is no interaction medium between a hospital and the public to announce their blood donation schedule. The blood donation event schedule is normally advertised to the public so that they are aware of the blood donation campaign period. At the blood house unit, the staffs and nurses only are informed about the blood donation schedule for each month on the whiteboard at the blood house. So they are using manual way in informing the schedule. The problem arises when the space provided is not enough. The medium used to inform the staff about the schedule of the month is using whiteboard and it is written by using whiteboard marker. Therefore, the writing tends to become unclear.

## **1.4 What is “Blood Donation Hub”?**

Blood Donation Hub is basically a blood management system. Blood donation hub is a website where The receiver can find a suitable donor for him/her. It saves the patient's life and saves time. This website is a nice and simple process to manage blood.

## **1.5 Objectives**

The Blood Donation Hub offers a way to connect donors and receivers. The objectives of this hub are given below:

1. To find possible donors easily
2. To connect between donors and receivers
3. To find blood on requests
4. To request for blood easily

## **1.6 Chapter Review**

**In chapter 1:** We talked about the basic things in our project report like what is Blood Donation Hub, scope of this project, objective of this project, objective of this document, Development tools etc. We also give an overall preview of the document in this chapter.

**In chapter 2:** We describe the statistics and background study of the project.

**In chapter 3:** This chapter describes system design and implementation. Here we discuss about modules overview like receiver section, donor section and admin section and also give some table structures. We also give, entities, attributes, relationship, E-R diagram, Use case diagram and database table.

**In chapter 4:** The system design parts are designed by unified model language. We fully documented this part to understand easily for the user and it is described in this chapter. We discuss UI and implementation. Here we gave some structures and screenshots of our project in the form design that helps users to use this hub easily. We gave a demo of admin login page, index page, donor & receiver registration, login page, donor & receiver profile page, admin dashboard, request page.

**In chapter 5:** In this chapter we discuss about the physical design and implementation of the project. We can also see examples of some major codes of the project.

# **CHAPTER 2**

## **BACKGROUND OF THE PROJECT**

---

### **2.1 Introduction**

Finding blood donors is a challenging issue in almost every country. Our blood donation system helps people who need emergency blood from the nearest donor. In the era of social media, some of the struggles have decreased, but still, when a patient needs blood, he/she has to contact a blood bank or a compatible blood group of a donor in their circle. However, it is difficult to find suitable donors within a limited group of people at a given time. Our target is to make an application where the blood receivers can connect with their nearby donors. We provide here a registration system for receiver's & donor's. First, the donor and receiver will fill the registration form. There are certain conditions to be fulfilled for filling the registration form, such as name, valid email Address, valid phone number, blood group, gender (minimum 18 years), password (minimum 8 characters) then it will be successfully created by donors and receivers. After registration, the donor and receiver will not be able to login to the page until the admin approves. Admin can delete the registration if he/she wants. Here the receiver always sends blood requests. So to send a blood request, the receiver must first login to the page. Then, in order to create a blood request, the receiver has to fulfill certain conditions. The conditions are name, phone number, the date of blood draw, approximate time, urgent or pre-collection. Then the receiver will create a blood request by selecting the location of the hospital that needs blood. Then the receiver can see the notification of the blood request.

### **2.2 Background Study**

Blood donation hub is an online based website. The main purpose of BDH is to provide necessary blood to all patients on an urgent basis. Finding blood in an emergency situation is very challenging. Many die from lack of blood. Considering all those situations we got the inspiration to make this website. There are many websites on the internet about blood donation. But ours is a little different. We have tried how users can get donors very easily as the location of the donor can be known through this website. Expectant mothers may need blood. Therefore, many people want to manage blood donors in advance. But many are in such a situation that the blood group of someone they know does not match. And even if the blood bank is told in advance, there is no guarantee that the blood group will be available at the time of need. It may also happen that the blood of that blood group has been given to an accident patient in an emergency. But you can keep the donor ready on our website through the pre-collection option. In this, if the pregnant mother needs blood during labor, it will be available at the right time without any additional hassle.

## **2.3 Design Requirements**

When we design software or a system, we consider some requirements to make the project efficient. We have tried to build a user-friendly web application. Our system user interface is very simple and everyone can use it without facing any problem. For better knowing, we have drawn a business process model, use case diagram, activity diagram and logical data model. We have used MySQL for designing our database. For front end design we have used HTML, CSS. We have used Laravel framework and PHP for back end design.

## **2.4 Challenges**

Every work has some challenges. So, we have to face a small number of challenges too.

- As it's a Web-based system, so people need to register/login online through any web connected devices such as computer, laptop, smart phone, tablets otherwise the system will not be helpful
- If a donor doesn't check the notification for blood request which is posted by receiver, then the main motto of this system will fail.



# CHAPTER 3

## TOOLS AND TECHNOLOGY

---

### 3.1 Introduction

In this chapter we will discuss the last phase of the project which is considered the application and the parallel phase of the design phase, which is about transforming what has been understood in the analysis phase and what has been outfitted in the design phase into reality.

### 3.2 Tools & Technology

#### Development Tools:

- **HTML:** HTML stands for Hyper Text Markup Language. HTML is the standard markup language for creating Web pages. HTML describes the structure of a Web page. HTML consists of a series of elements. HTML elements tell the browser how to display the content. HTML allows users to create and structure sections, headings, links, paragraphs, and more, on a website using various tags and elements. Almost everything you want to create on a web page can be done using a specific HTML code.
- **CSS:** It stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files. CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language.
- **jQuery:** jQuery is an open-sourced JavaScript library that simplifies creation and navigation of web applications. It is a lightweight, "write less, do more", JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on your website. jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.
- **JavaScript:** JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM). JavaScript is used by programmers across the world to create dynamic and interactive web content like applications and browsers. JavaScript is so popular that it's the most used programming language in the world, used as a client-side programming language by 97.0% of all websites.

- **Bootstrap:** Bootstrap utilizes Sass for a modular and customizable architecture. Import only the components you need, enable global options like gradients and shadows, and write your own CSS with our variables, maps and functions. Bootstrap is a free, open source front-end development framework for the creation of websites and web apps. Designed to enable responsive development of mobile-first websites, Bootstrap provides a collection of syntax for template designs.
- **PHP:** PHP is an acronym for "PHP: Hypertext Preprocessor" PHP is a widely-used, open source scripting language. PHP scripts are executed on the server. PHP is free to download and use. PHP (short for Hypertext Preprocessor) is the most widely used open source and general purpose server side scripting language used mainly in web development to create dynamic websites and applications. A PHP framework is a platform to create PHP web applications. PHP frameworks provide code libraries for commonly used functions, cutting down on the amount of original code you need to write.
- **Laravel:** Laravel attempts to take the pain out of development by easing common tasks used in the majority of web projects, such as authentication, routing, sessions, and caching. Laravel aims to make the development process a pleasing one for the developer without sacrificing application functionality. The best advantages of Laravel include scalability, rapid page loading, Modular View controller architecture, and security. Building web applications in Laravel framework will enhance the performance and provides user-friendly access.
- **MySQL:** MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL). A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or a place to hold the vast amounts of information in a corporate network. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server.
- **XAMPP:** XAMPP is a free and Open-source cross-platform Web Server Solution Stack built by Apache Friends, which consists mostly of the Apache HTTP Server, MySQL Database, and interpreters for PHP and Perl scripts. The goal of XAMPP is to build an easy to install distribution for developers to get into the world of Apache. To make it convenient for developers, XAMPP is configured with all features turned on.

## Front End Design

The front end is normally designed in two parts. The web design and front end web development. We use HTML, CSS and JavaScript for our front end design. These include things like fonts, drop-down menus, contact forms etc. In this section we include some home page front end design.



## Back End Design:

The backend is normally designed in two parts

- API
- A database

Users input the data that the application stores in a database that was created on a server. We basically used PHP.



## Supporting Tools:



**Code Editor:** Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications.



**Local Server:** XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache

# CHAPTER 4

## LOGICAL SYSTEM SPECIFICATION

---

### 3.1 Introduction

Blood donation hubs are a vital part of worldwide healthcare. It relates to blood transfusion as a life-sustaining and life-saving procedure as well as a form of therapeutic phlebotomy as a primary medical intervention. Over one hundred million units of blood are donated each year throughout the world. [5] The system allows one to easily access the donors' information and make blood donating arrangements easy. It's a helpful website for patients and donors.

### 3.2 Modules Overview

#### Admin Section:

1. Donors and receivers can register
2. Donors and receivers can't login until their registration
3. Admin can delete donors and receivers. if he/she provides any wrong information
4. Admin can make changes on the admin dashboard
5. Admin can approve or delete all blood requests
6. Admin can send selected/verified donors' information to receivers
7. Admin can view all users' lists (total donors, total receivers and blood requests)
8. Admin can see on his/her dashboard, if the receivers can manage or can't manage blood

#### Donor Section:

1. Donors can register
2. Donors can't login until their registration
3. Donors can edit their personal information on their profile
4. Donors can change their status (available or unavailable)
5. Donors can't see any blood requests if their status remains unavailable
6. Donors can see receiver's information while accepting blood requests
7. Donors can add their previous blood donating date

#### Receiver Section:

1. Receivers can register
2. Receivers can't login until their registration
3. Receivers can edit their personal information on their profile
4. Receivers can send blood requests
5. Receivers can switch to donors
6. Receiver can edit or delete his blood request if he wants
7. When the donor accepts the receiver's blood request (after admin approves and sends), Then he/she can see the details of the donor and accepts if he wants
8. If the receiver manages the blood, then his blood request will be deleted (if he goes to his profile and clicks on the "managed" option)

### 3.3 Entities

An entity is an object that exists. It doesn't have to do anything; it just has to exist. In database administration, an entity can be a single thing, person, place, or object. Data can be stored about such entities. A design tool that allows database administrators to view the relationships between several entities is called the entity relationship diagram (ERD). If you aren't going to capture data about something, there's no point in creating an entity in a database. [3]

### 3.4 Attributes

An attribute is a property, trait, or characteristic of an entity, relationship, or another attribute. For example, the attribute Inventory Item Name is an attribute of the entity Inventory Item. An entity can have as many attributes as necessary. Meanwhile, attributes can also have their own specific attributes.

For example, the attribute customer address can have the attributes number, street, city, and state. These are called composite attributes. Note that some top level ER diagrams do not show attributes for the sake of simplicity. In those that do, however, attributes are represented by oval shapes.

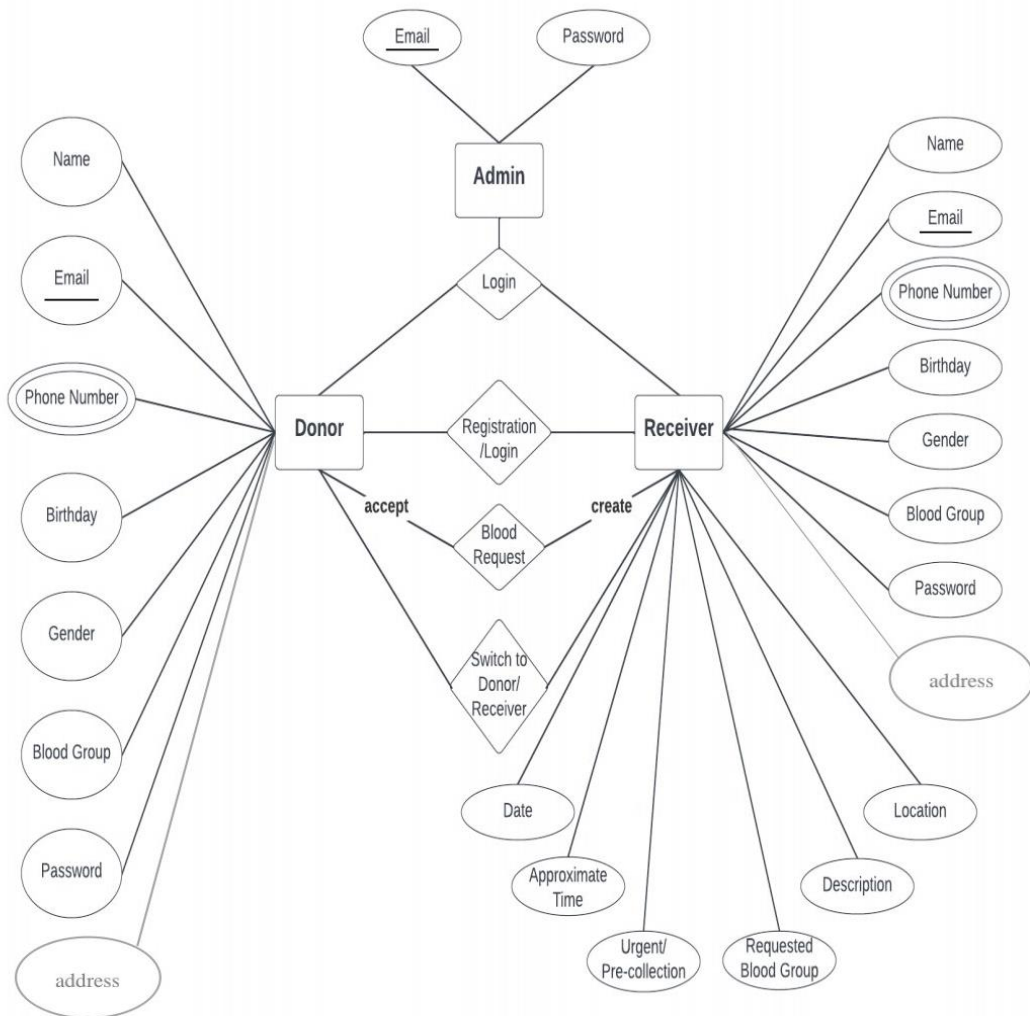
- **Multivalued Attribute:** If an attribute can have more than one value it is called a multivalued attribute. It is important to note that this is different to an attribute having its own attributes. For example, a teacher entity can have multiple subject values.
- **Derived Attribute:** An attribute based on another attribute. This is found rarely in ER diagrams. For example, for a circle, the area can be derived from the radius.

### 3.5 Relationships

Relationships are how the entities are related to each other. So that records can be associated with each other in queries or view definitions. A relationship is similar to a parent and a child within a family. The parent entity has a key value (primary key) that uniquely identifies a record in one table with other tables or entities that are considered children. The child table or entity has a foreign key, a column that holds the value of the parent entity. So the two entities joined the ducting query process.

### 3.6 E-R Diagram

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases. An entity relationship model, also called an entity-relationship (ER) diagram, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data-an object or concept about which data is stored.



**Fig 3.1: E-R Diagram**

### 3.7 Use Case Diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

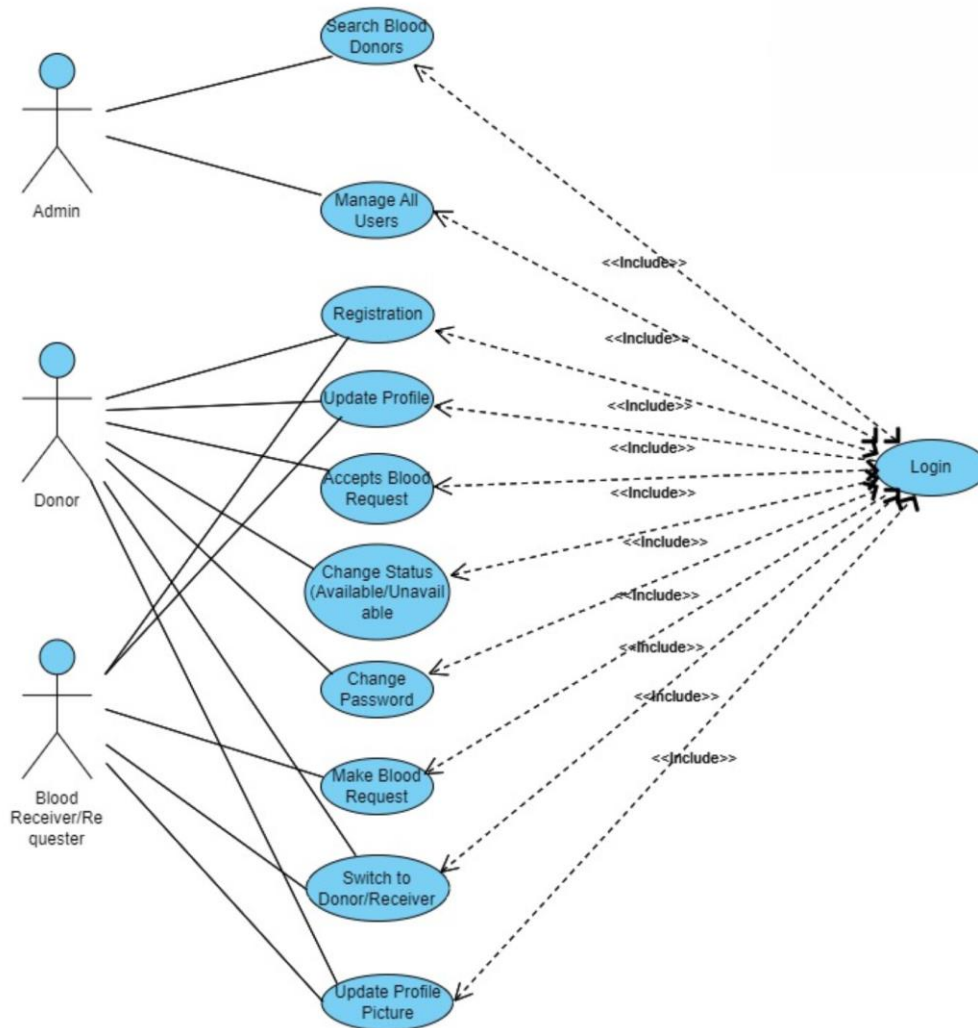
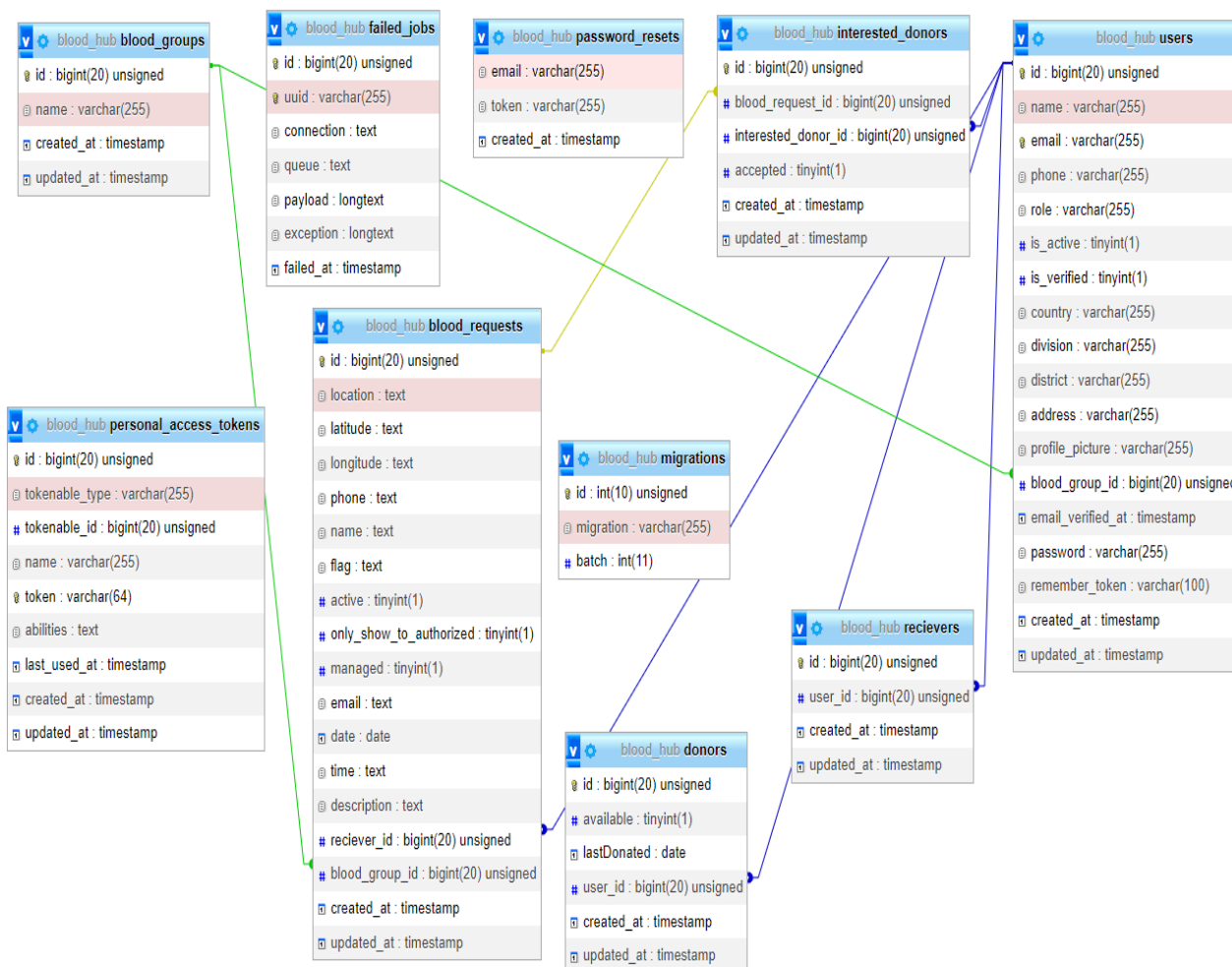


Fig 3.2: Use case Diagram

### 3.8 Schema Diagram

The design of the database is called a schema. This tells us about the structural view of the database. It gives us an overall description of the database. A database schema defines how the data is organized using the schema diagram. A schema diagram is a diagram which contains entities and the attributes that will define that schema. A schema diagram only shows us the database design. It does not show the actual data of the database. Schema can be a single table or it can have more than one table which is related. The schema represents the relationship between these tables.

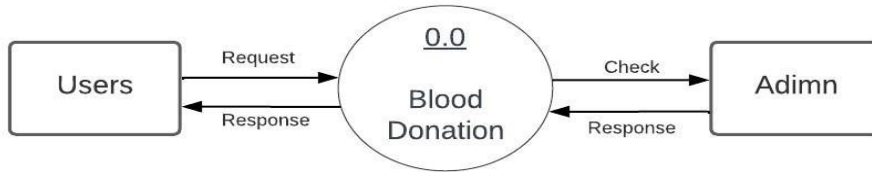


**Fig 3.3: Schema Diagram**

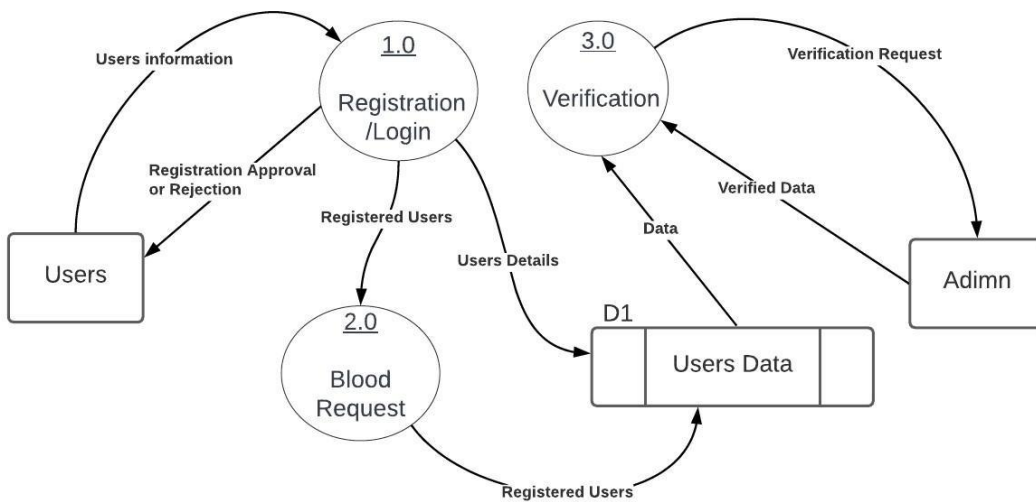


### 3.9 Data-Flow Diagram

A data-flow diagram is a way of representing a flow of data through a process or a 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.

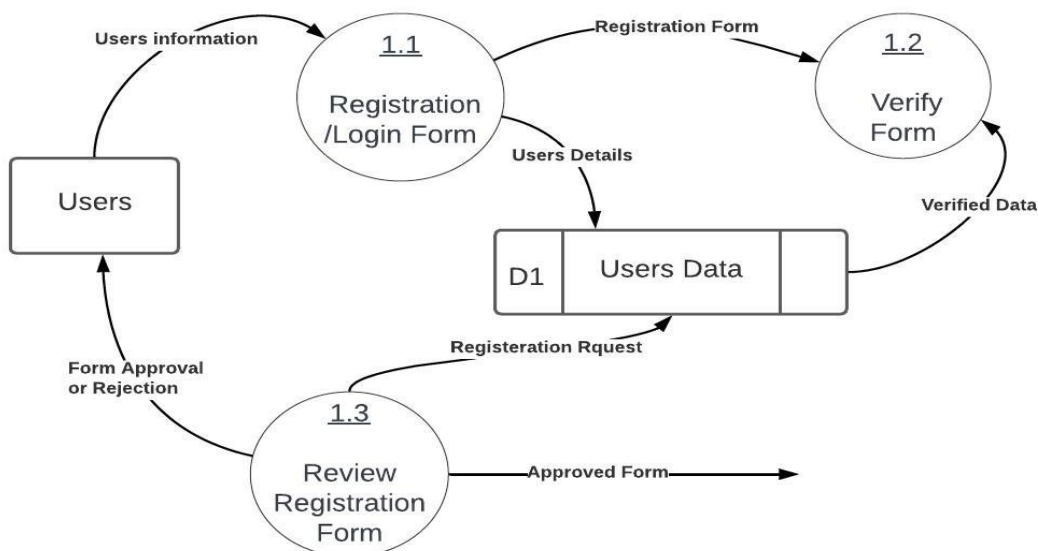


**Fig 3.4: Data-flow Diagram (Level-0)**



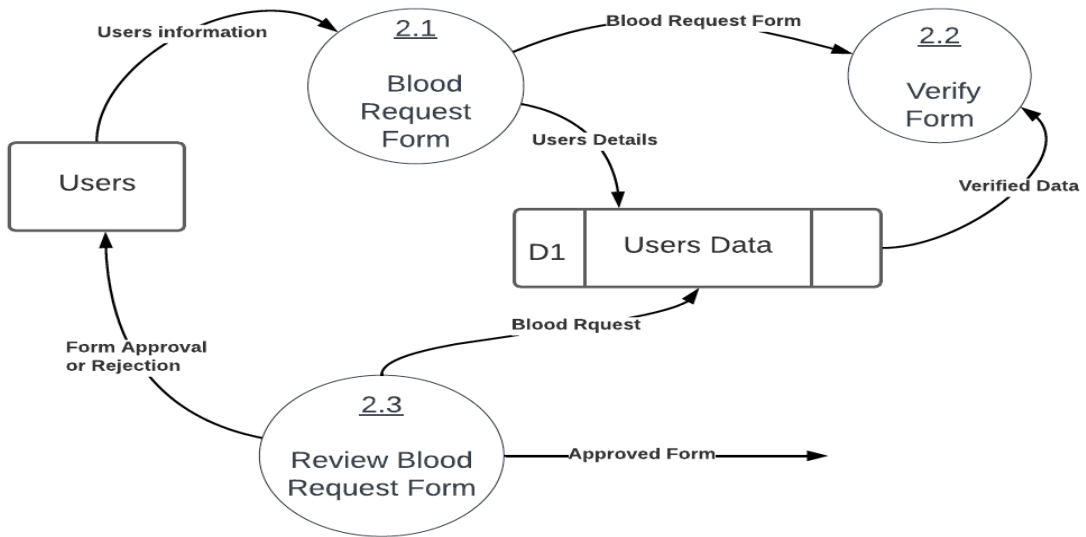
**Fig 3.5: Data-flow Diagram (Level-1)**

**Process-1:**



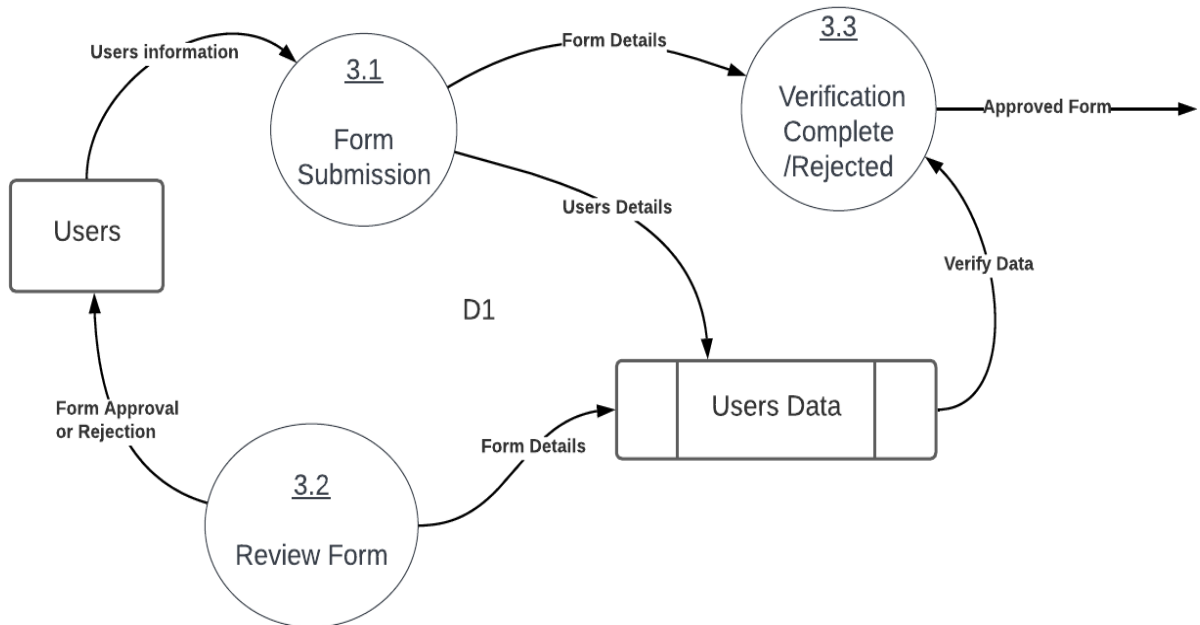
**Fig 3.6: Data-flow Diagram (Level:2)**

**Process-2:**



**Fig 3.7: Data-flow Diagram (Level:3)**

**Process-3:**



**Fig 3.8: Data-flow Diagram (Level:4)**

## 3.10 Database Table

### Donor Table

	id	name	email	phone	role	is_active	is_verified	country	division	district	address	profile_picture	blood_group_id	email_verified_at
<input type="checkbox"/>	1	Admin	admin@bdh.com	01712345678	admin	1	1	NULL	NULL	NULL	NULL	NULL	1	NULL
<input type="checkbox"/>	4	munia	munia@gmail.com	01868476952	donor	1	0	NULL	NULL	NULL	NULL	NULL	7	NULL
<input type="checkbox"/>	5	shamali	shamali@gmail.com	01786786858	reciever	1	0	NULL	NULL	NULL	NULL	NULL	5	NULL
<input type="checkbox"/>	6	mahad	mahad@gmail.com	01868476952	donor	1	0	NULL	NULL	NULL	NULL	NULL	5	NULL
<input type="checkbox"/>	7	kakoli	kakoli@gmail.com	01789456534	donor	1	0	NULL	NULL	NULL	NULL	NULL	7	NULL
<input type="checkbox"/>	8	Saqlin	saqlin@gmail.com	01868476952	donor	1	0	NULL	NULL	NULL	NULL	NULL	1	NULL
<input type="checkbox"/>	9	Fatema	fatema@gmail.com	01786786858	reciever	1	0	NULL	NULL	NULL	NULL	NULL	2	NULL
<input type="checkbox"/>	10	Munia	muniaakter@gmail.com	01868476952	donor	1	0	NULL	NULL	NULL	NULL	NULL	2	NULL
<input type="checkbox"/>	11	Lara	lara@gmail.com	01758573598	reciever	1	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL
<input type="checkbox"/>	12	Habib	habib@bdh.com	01758573598	reciever	1	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL
<input type="checkbox"/>	13	Nahian	nahian@gmail.com	01758573598	donor	1	0	NULL	NULL	NULL	NULL	NULL	8	NULL
<input type="checkbox"/>	14	Raiyan	raiyan@gmail.com	01789585764	donor	1	0	NULL	NULL	NULL	NULL	NULL	8	NULL

Table 3.1: Donor Table

### Interested Donor Table

	id	blood_request_id	interested_donor_id	accepted	created_at	updated_at
<input type="checkbox"/>	1	1	6	1	2022-09-10 08:33:11	2022-09-10 15:58:46
<input type="checkbox"/>	2	2	8	1	2022-09-10 15:20:30	2022-09-10 15:56:08
<input type="checkbox"/>	3	3	4	1	2022-09-13 02:15:18	2022-09-13 02:17:46
<input type="checkbox"/>	4	4	14	1	2022-09-15 05:17:42	2022-09-15 05:21:22
<input type="checkbox"/>	5	5	7	1	2022-09-18 07:21:39	2022-09-18 07:23:04
<input type="checkbox"/>	6	3	7	1	2022-09-18 07:21:46	2022-09-18 07:22:56

Table 3.2: Interested Donor Table

## Receiver Table

		id	user_id	created_at	updated_at
<input type="checkbox"/>	Edit  Copy  Delete	3	5	2022-09-10 07:53:12	2022-09-10 07:53:12
<input type="checkbox"/>	Edit  Copy  Delete	4	7	2022-09-10 15:05:24	2022-09-10 15:05:24
<input type="checkbox"/>	Edit  Copy  Delete	5	9	2022-09-10 15:08:30	2022-09-10 15:08:30
<input type="checkbox"/>	Edit  Copy  Delete	6	11	2022-09-13 02:02:46	2022-09-13 02:02:46
<input type="checkbox"/>	Edit  Copy  Delete	7	12	2022-09-15 04:19:44	2022-09-15 04:19:44

Table 3.3: Receiver Table

## Blood Request Table

		id	location	latitude	longitude	phone	name	flag	active	only_show_to_authorized	managed	email	date	time
<input type="checkbox"/>	Edit  Copy  Delete	1	72-23 37th Ave, Queens, NY 11372, USA	40.74896201368282	-73.89300649999997	01786786858	sharnali urget		1		1	1 shamali@gmail.com	2022-09-10	04:00 AM
<input type="checkbox"/>	Edit  Copy  Delete	2	Secretariat Rd, Dhaka 1000, Bangladesh	23.725942356979502	90.39744285000003	01789456534	kakoli urget		1		0	0 kakoli@gmail.com	2022-09-12	01:00 AM
<input type="checkbox"/>	Edit  Copy  Delete	3	2 Bara Maghbazar, Outer Circular Rd, Dhaka 1217, B...	23.74848265698702	90.40554940000003	01758573598	Lara	pre_collection	1		0	1 lara@gmail.com	2022-09-17	02:00 AM
<input type="checkbox"/>	Edit  Copy  Delete	4	2 Bara Maghbazar, Outer Circular Rd, Dhaka 1217, B...	23.74848265698702	90.40554940000003	01758573598	Habib	urget	1		0	1 habib@bdh.com	2022-09-15	02:30 AM
<input type="checkbox"/>	Edit  Copy  Delete	5	2 Bara Maghbazar, Outer Circular Rd, Dhaka 1217, B...	23.74848265698702	90.40554940000003	01786786858	sharnali urget		1		0	1 shamali@gmail.com	2022-09-21	02:00 AM

Table 3.4: Blood Request Table

## Blood Group Table

		id	name	created_at	updated_at
<input type="checkbox"/>	Edit  Copy  Delete	1	A+	2022-09-09 16:35:43	2022-09-09 16:35:43
<input type="checkbox"/>	Edit  Copy  Delete	2	A-	2022-09-09 16:35:43	2022-09-09 16:35:43
<input type="checkbox"/>	Edit  Copy  Delete	3	B+	2022-09-09 16:35:43	2022-09-09 16:35:43
<input type="checkbox"/>	Edit  Copy  Delete	4	B-	2022-09-09 16:35:43	2022-09-09 16:35:43
<input type="checkbox"/>	Edit  Copy  Delete	5	AB+	2022-09-09 16:35:43	2022-09-09 16:35:43
<input type="checkbox"/>	Edit  Copy  Delete	6	AB-	2022-09-09 16:35:43	2022-09-09 16:35:43
<input type="checkbox"/>	Edit  Copy  Delete	7	O+	2022-09-09 16:35:43	2022-09-09 16:35:43
<input type="checkbox"/>	Edit  Copy  Delete	8	O-	2022-09-09 16:35:43	2022-09-09 16:35:43

Table 3.5: Blood Group Table

# CHAPTER 5

## USER INTERFACE AND IMPLEMENTATION

### 4.1 Introduction

In this chapter, we will create and develop, review and run our project and also test our application. The outcome of testing will be provided to verify the application ability and quality.

### 4.2 Home Page

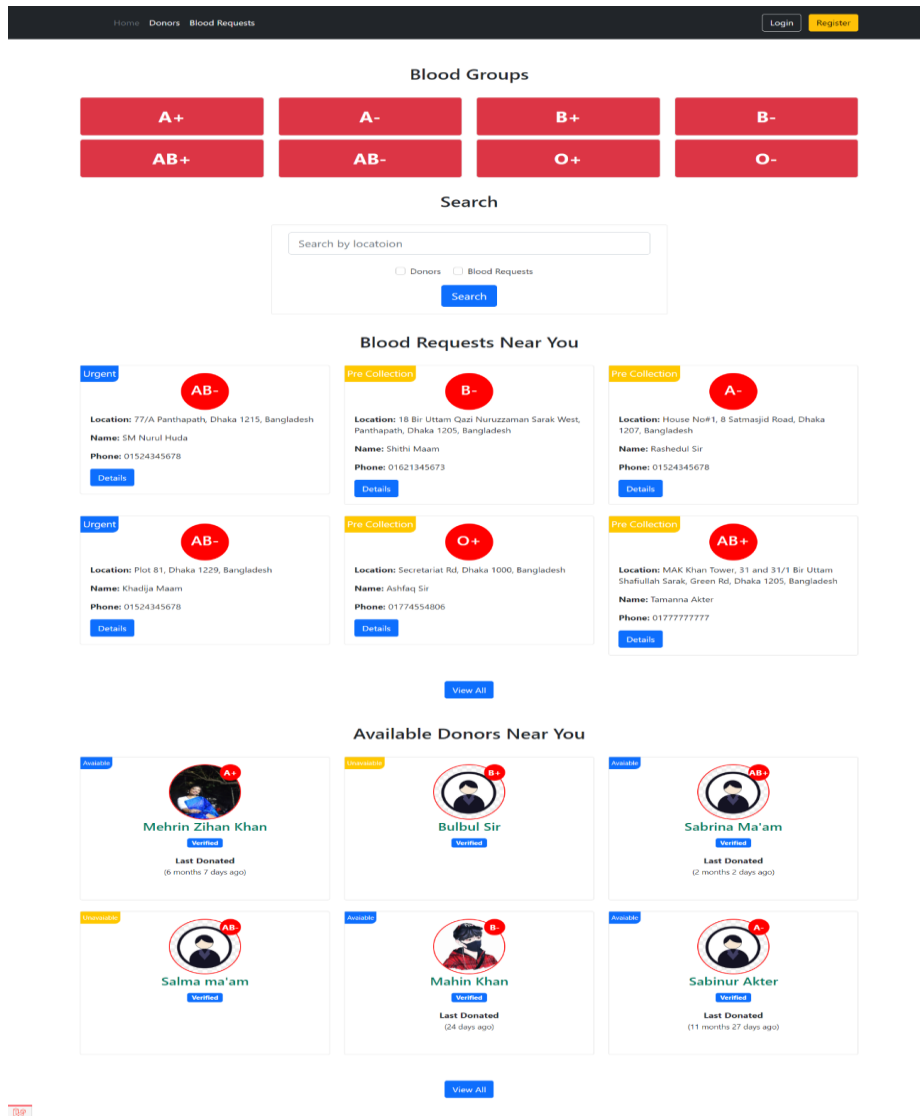


Fig 4.1: Home Page

Login and registration can be done from the home page. Receivers can search donors by location. Donors and blood requests of all blood groups can also be found here.

### 4.3 Registration Form page

The screenshot shows a web page with a dark header. On the left, there are links for 'Home', 'Donors', and 'Blood Requests'. On the right, there are 'Login' and 'Register' buttons. The main content is a 'Registration Form' with the following fields and options:

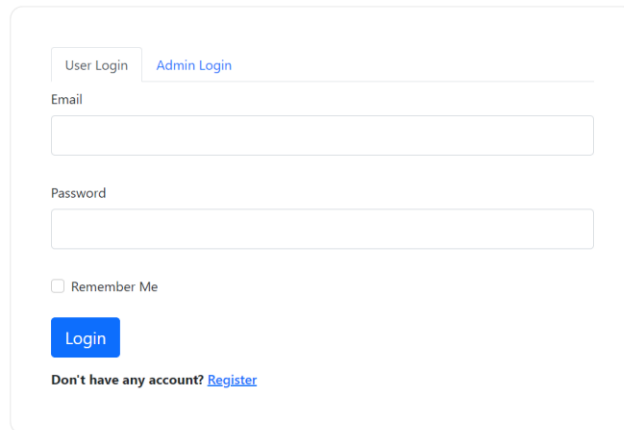
- Name: Text input field
- Email: Text input field
- Phone Number: Text input field
- Address: Text input field
- Birthday: Text input field with placeholder 'mm/dd/yyyy' and a calendar icon
- Gender: Radio buttons for 'Female' (selected), 'Male', and 'Other'
- Register As: Radio buttons for 'Receiver' (selected) and 'Donor'
- Select Blood Group: Dropdown menu with 'Select Blood Group' and a downward arrow
- Password: Text input field
- Confirm Password: Text input field

At the bottom of the form, there is a blue 'Register' button and a link that says 'Have an account? [Login](#)'.

**Fig 4.2: Registration Form page**

Donors and receivers register here by giving name, email, phone number, address, birthday, gender, blood group and password.

## 4.4 User Login & Admin Login Page



The form is contained within a light gray rounded rectangle. At the top, there are two tabs: 'User Login' (active) and 'Admin Login'. Below the tabs are two input fields: 'Email' and 'Password'. Under the 'Password' field is a checkbox labeled 'Remember Me'. A blue 'Login' button is positioned below the checkbox. At the bottom of the form, there is a link: 'Don't have any account? [Register](#)'.


**Fig 4.3: User Login & Admin Login Page**

From here admin and user login by giving email and password. Admin will be one person so his/her email ID, password will be fixed so no one else can be an admin.

## 4.5 Receiver Profile



Home Donors Blood Requests

Home / User Profile

  
**Rashedul Sir**  
(receiver)  
[Blood Request](#)

Full Name	Rashedul Sir
Email	rashed@gmail.com
Phone	01524345678
Blood Group	A-
Address	Razabazar, Dhaka

### Blood Requests


#	Blood Group	Location	Date	Time	Flag	Can See By	Managed	Interested Donors	Action
1	A-	House No#1, 8 Satmasjid Road, Dhaka 1207, Bangladesh	2023-01-27	07:00 PM	<span>Pre Collection</span>	<span>All Donors</span>	<span>Not Managed</span>	Details	 

**Fig 4.4: Receiver Profile**

It is a receiver profile, where the receiver sends blood requests.




## 4.6 Blood Request page

Home Donors Blood Requests 

### Create Blood Request

Name

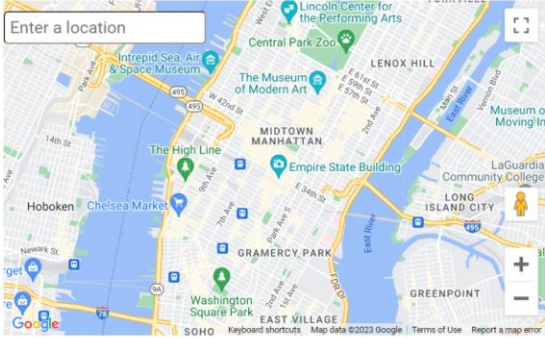
Contact Number  Contact Email

Date   Approx. Time

Flag  Urgent  Pre Collection Select Blood Group

Description

Location



Only Show Request To authorized Donors

**Fig 4.5: Blood Request Page**

Receiver can send blood request by informing date, approximate time, urgent/pre-collection, requested blood group, description and location from here.

## 4.7 No Interested Donors (from Receiver Profile)

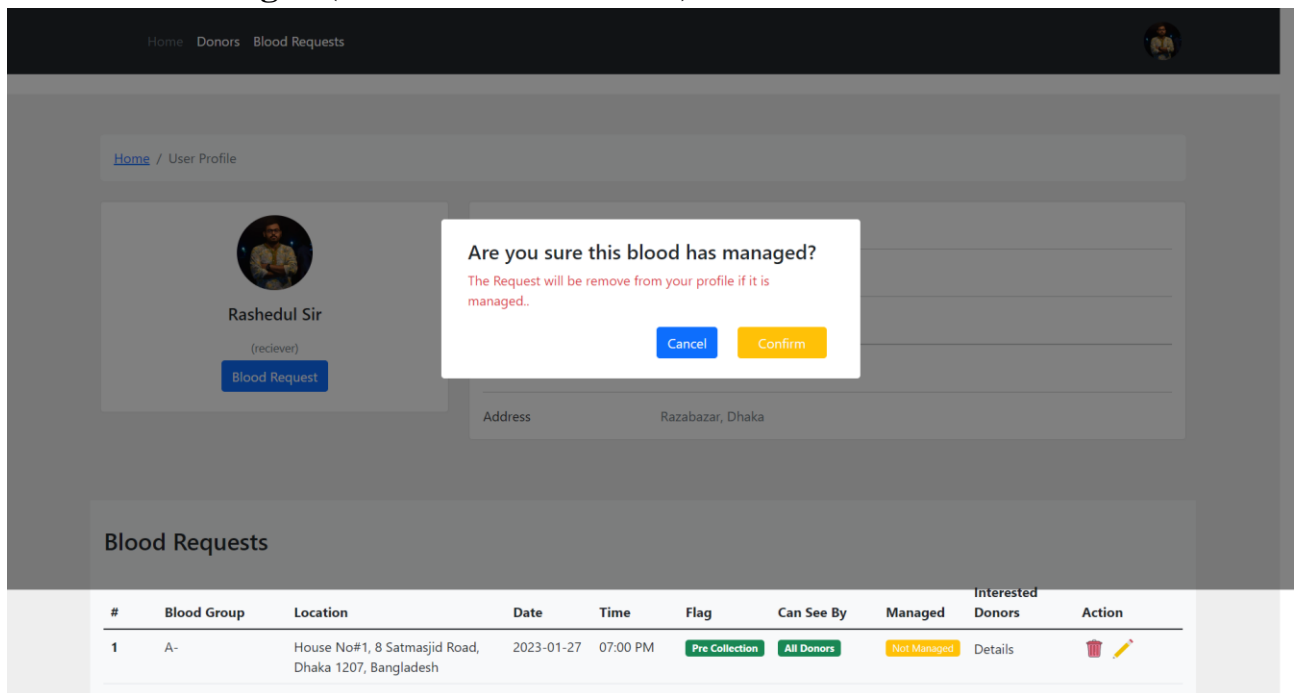
The screenshot shows a user profile for 'Rashedul Sir' (receiver) with a 'Blood Request' button. A modal window titled 'Interested Donors' is displayed, stating 'No Interested Donors Found!' with a 'Close' button. Below the profile, a table lists blood requests. The table has columns: #, Blood Group, Location, Date, Time, Flag, Can See By, Managed, Interested Donors, and Action. The first row shows a request for blood group A- at House No#1, 8 Satmasjid Road, Dhaka 1207, Bangladesh, dated 2023-01-27 at 07:00 PM, with flags 'Pre Collection' and 'All Donors', and a 'Not Managed' status. The 'Interested Donors' column shows 'Details' and the 'Action' column has delete and edit icons.

#	Blood Group	Location	Date	Time	Flag	Can See By	Managed	Interested Donors	Action
1	A-	House No#1, 8 Satmasjid Road, Dhaka 1207, Bangladesh	2023-01-27	07:00 PM	Pre Collection	All Donors	Not Managed	Details	



**Fig 4.6: Receiver Profile**

Here, the receiver sends blood requests successfully but does not find interested donors.

## 4.8 Blood Managed (from Receiver Profile)



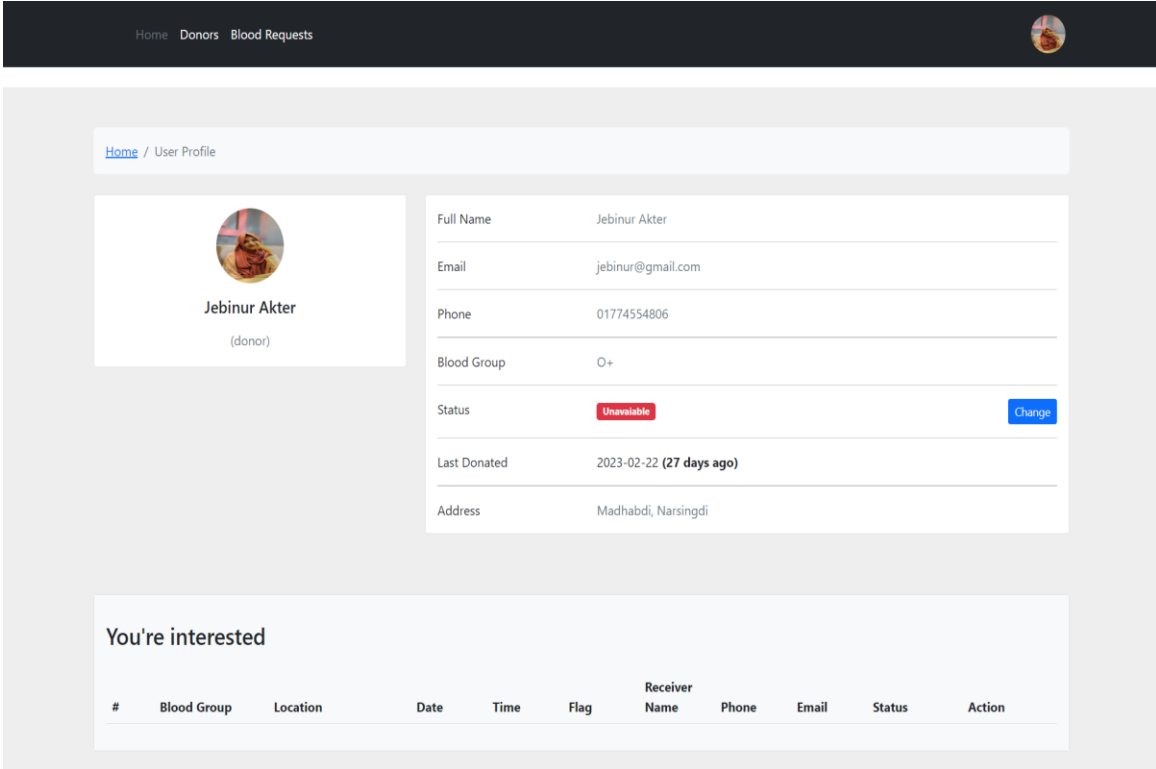
The screenshot shows a user profile for 'Rashedul Sir' (receiver) with a 'Blood Request' button. A modal dialog is displayed with the text: 'Are you sure this blood has managed? The Request will be remove from your profile if it is managed.' with 'Cancel' and 'Confirm' buttons. Below the profile, the address is 'Razabazar, Dhaka'. At the bottom, a 'Blood Requests' table is visible.

#	Blood Group	Location	Date	Time	Flag	Can See By	Managed	Interested Donors	Action
1	A-	House No#1, 8 Satmasjid Road, Dhaka 1207, Bangladesh	2023-01-27	07:00 PM	Pre Collection	All Donors	Not Managed	Details	 

**Fig 4.7: Receiver Profile (Blood Managed)**

When the receiver's blood is managed, he/she can confirm if he/she wants, and if he/she confirms, the blood request will be removed from his/her profile.


## 4.9 Donor Profile (Status Unavailable)



The screenshot shows a user profile for 'Jebinur Akter' with a status of 'Unavailable'. The profile includes personal details such as full name, email, phone, blood group, and last donation date. A 'Change' button is visible next to the status. Below the profile is a table titled 'You're interested' with columns for #, Blood Group, Location, Date, Time, Flag, Receiver Name, Phone, Email, Status, and Action.

Home Donors Blood Requests

Home / User Profile

  
**Jebinur Akter**  
(donor)

Full Name	Jebinur Akter
Email	jebinur@gmail.com
Phone	01774554806
Blood Group	O+
Status	<span>Unavailable</span> <a href="#">Change</a>
Last Donated	2023-02-22 (27 days ago)
Address	Madhabdi, Narsingdi

You're interested

#	Blood Group	Location	Date	Time	Flag	Receiver Name	Phone	Email	Status	Action
---	-------------	----------	------	------	------	---------------	-------	-------	--------	--------

**Fig 4.8: Donor Profile (Status Unavailable)**

It is a donor profile. Here the donor will not get any request as long as the status of the donor is unavailable.

## 4.10 Donor Profile (Status Available)

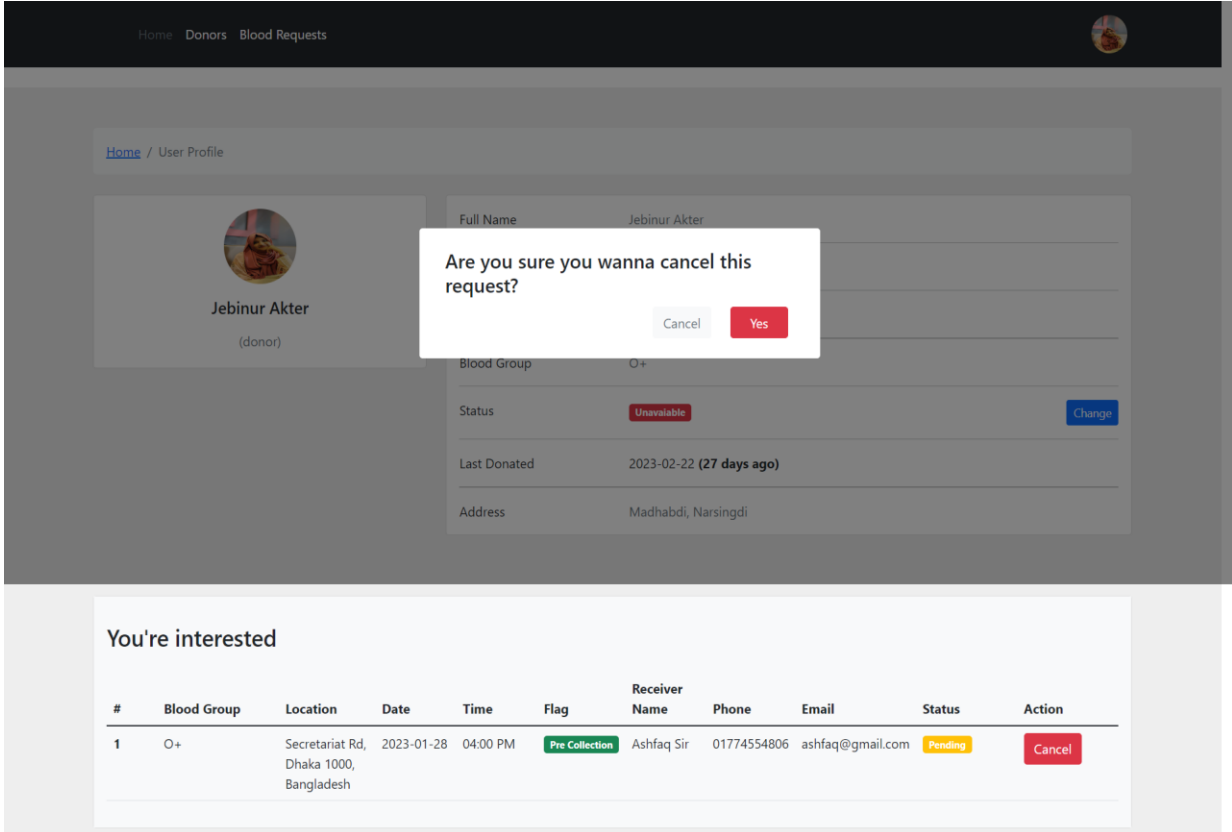
The screenshot displays a user profile for a donor named Jebinur Akter. The profile includes a circular profile picture, the name 'Jebinur Akter', and the role '(donor)'. To the right, a list of personal details is shown: Full Name (Jebinur Akter), Email (jebinur@gmail.com), Phone (01774554806), Blood Group (O+), Status (Available), Last Donated (2023-02-22, 27 days ago), and Address (Madhabdi, Narsingdi). Below this, a section titled 'You're interested' contains a table of blood requests.

#	Blood Group	Location	Date	Time	Flag	Receiver Name	Phone	Email	Status	Action
1	O+	Secretariat Rd, Dhaka 1000, Bangladesh	2023-01-28	04:00 PM	Pre Collection	Ashfaq Sir	01774554806	ashfaq@gmail.com	Pending	Cancel

**Fig 4.9: Donor Profile (Status Available)**

When the donor's status is available, he will receive blood requests and he/she can also accept the request.

## 4.11 Donor Profile (Request Cancel)



The screenshot shows a user profile for 'Jebinur Akter' (donor). A modal dialog is displayed in the center, asking 'Are you sure you wanna cancel this request?' with 'Cancel' and 'Yes' buttons. The background profile card shows the following details:

- Full Name: Jebinur Akter
- Blood Group: O+
- Status: Unavailable (with a 'Change' button)
- Last Donated: 2023-02-22 (27 days ago)
- Address: Madhabdi, Narsingdi

Below the profile card, there is a section titled 'You're interested' containing a table of requests:

#	Blood Group	Location	Date	Time	Flag	Receiver Name	Phone	Email	Status	Action
1	O+	Secretariat Rd, Dhaka 1000, Bangladesh	2023-01-28	04:00 PM	Pre Collection	Ashfaq Sir	01774554806	ashfaq@gmail.com	Pending	Cancel

**Fig 4.10: Donor Profile (Request cancel)**

The donor can cancel the blood request in an emergency.

## 4.12 Admin Dashboard

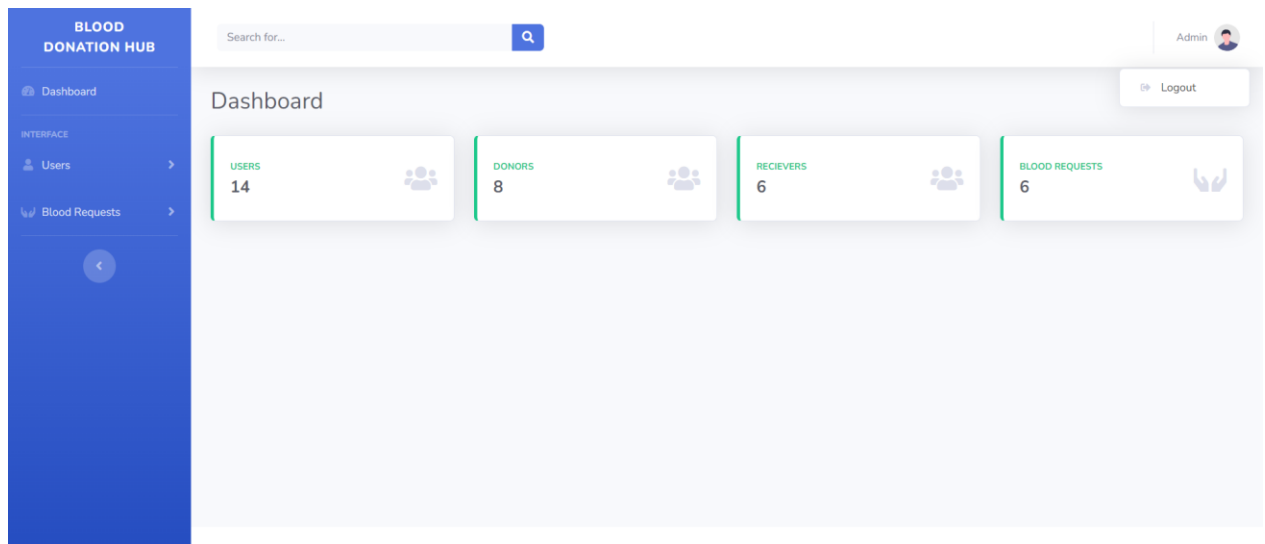


Fig 4.11: Admin Dashboard

## 4.13 Admin Dashboard (All Donors List)

The screenshot displays the 'All Donors' section of the 'BLOOD DONATION HUB' admin dashboard. The interface includes a search bar at the top, a sidebar with navigation options (Dashboard, Users, Blood Requests), and a table listing donor details. The table has columns for ID, Name, Blood Group, Role, Email, Phone, Address, Status, Registered At, and Action. Each row represents a donor, and the 'Action' column contains 'Delete' and 'Deactivate' buttons. The status of all donors is 'Active'.

#	Name	Blood Group	Role	Email	Phone	Address	Status	Registered At.	Action
1	munia	O+	donor	munia@gmail.com	01868476952		Active	2022-09-10 07:43:08	Delete Deactivate
2	kakoli	O+	donor	kakoli@gmail.com	01789456534		Active	2022-09-10 15:05:23	Delete Deactivate
3	Saqlin	A+	donor	saqlin@gmail.com	01868476952		Active	2022-09-10 15:07:04	Delete Deactivate
4	Munia	A-	donor	muniaakter@gmail.com	01868476952		Active	2022-09-10 15:11:04	Delete Deactivate
5	Nahian	O-	donor	nahian@gmail.com	01758573598		Active	2022-09-15 04:57:35	Delete Deactivate
6	Raiyan	O-	donor	raiyan@gmail.com	01789585764		Active	2022-09-15 05:08:51	Delete Deactivate

Showing 1 to 6 of 6 entries

Previous 1 Next

**Fig 4.12: Admin Dashboard (All Donors List)**



## 4.14 Admin Dashboard (All Receivers List)

The screenshot displays the 'All Receivers List' on the Admin Dashboard. The dashboard has a blue sidebar with 'BLOOD DONATION HUB' and navigation options like 'Dashboard', 'Users', and 'Blood Requests'. The main content area features a search bar, a 'Show 10 entries' dropdown, and a search input. Below is a table with 5 entries, each with columns for #, Name, Blood Group, Role, Email, Phone, Address, Status, Registered At, and Action. The 'Action' column contains 'Delete' and 'Deactivate' buttons. At the bottom, it shows 'Showing 1 to 5 of 5 entries' and a pagination control with 'Previous', '1', and 'Next' buttons.

#	Name	Blood Group	Role	Email	Phone	Address	Status	Registered At	Action
1	sharnali	AB+	reciever	sharnali@gmail.com	01786786858		Active	2022-09-10 07:53:12	Delete Deactivate
2	mahad	AB+	reciever	mahad@gmail.com	01868476952		Active	2022-09-10 08:25:13	Delete Deactivate
3	Fatema	A-	reciever	fatema@gmail.com	01786786858		Active	2022-09-10 15:08:30	Delete Deactivate
4	Lara		reciever	lara@gmail.com	01758573598		Active	2022-09-13 02:02:46	Delete Deactivate
5	Habib		reciever	habib@bdh.com	01758573598		Active	2022-09-15 04:19:44	Delete Deactivate

**Fig 4.13: Admin Dashboard (All Receivers list)**

## 4.15 Admin Dashboard (Blood Request List)

The screenshot displays the Admin Dashboard for the Blood Donation Hub. The main content area is titled 'All Users' and features a table listing 10 users. The table columns are: #, Name, Blood Group, Role, Email, Phone, Address, Status, Registered At, and Action. Each user entry includes a 'Delete' button (red) and a 'Deactivate' button (yellow). The status for all users is 'Active' (green). The dashboard also includes a sidebar with navigation options: Dashboard, Users, and Blood Requests. A search bar is located at the top right, and a user profile icon is visible in the top right corner.

#	Name	Blood Group	Role	Email	Phone	Address	Status	Registered At	Action
1	Jebinur Akter	O+	donor	jebinur@gmail.com	01774554806	Madhabdi, Narsingdi	Active	2023-01-26 17:10:24	Delete Deactivate
2	Mehrin Zihan Khan	A+	donor	mehrin@gmail.com	01780026073	Nowabganj, Dhaka	Active	2023-01-26 17:16:08	Delete Deactivate
3	Tamanna Akter	O-	donor	tamanna@gmail.com	01777777777	Narayanganj, Dhaka	Active	2023-01-26 17:18:03	Delete Deactivate
4	Bulbul Sir	B+	donor	bulbul@gmail.com	01621345673	Banasree, Rampura, Dhaka	Active	2023-01-26 17:20:18	Delete Deactivate
5	Arif Sir	B-	reciever	arif@gmail.com	01524345678	Noakhali	Active	2023-01-26 17:22:17	Delete Deactivate
6	Sabrina Ma'am	AB+	donor	sabrina@gmail.com	01777777778	Rongpur	Active	2023-01-26 17:24:36	Delete Deactivate
7	Salma ma'am	AB-	donor	salma@gmail.com	01777777788	thakurgaon	Active	2023-01-26 17:27:09	Delete Deactivate
8	Mahin Khan	B-	donor	mahin@gmail.com	01774554806	Nowabganj, Dhaka	Active	2023-01-26 18:05:37	Delete Deactivate
9	Sabinur Akter	A-	donor	sabinur@gmail.com	01621345673	Mirpur, Dhaka	Active	2023-01-26 18:17:18	Delete Deactivate
10	Rashedul Sir	A-	reciever	rashed@gmail.com	01524345678	Razabazar, Dhaka	Active	2023-01-26 18:39:52	Delete Deactivate

Showing 1 to 10 of 14 entries

Previous 1 2 Next

**Fig 4.14: Admin Dashboard (Blood Request List)**

# CHAPTER 6

## CONCLUSION, LIMITATIONS AND FUTURE WORKS

---

### 6.1 Conclusion

The system provides a web-based application that is acutely useful for emergency services. It will come very useful in urgent times by providing donors information searched by location and blood type. The system consists of a well-maintained database to keep all the registered records. It also provides news and information about the ongoing coronavirus pandemic. In the end, it provided us the knowledge regarding the latest technology required to build a web-based application. During the building of this project, it provided us an awareness of how blood donation can save lives. This inspired us to donate blood at regular times and also motivate and persuade our fellow citizens to donate blood. A database has been set up to store historical data related to donation and reception of blood and also to store data from camps so as to take future decisions based on concrete analytical results.

### 6.2 Limitations

Limitations of our project are minimal and manageable, they are:

- There is no automatic page generator.
- There is no automatic menu builder.
- There is no option of live chatting.
- There is no blood test report giving option.

### 6.3 Future Works

Online system is always changeable. It develops day by day, getting better and easier for everyone. We believe we can make this system more advanced in the future. The changes we are going to make in this project are given bellow:

- To add automatic menu builder.
- To add an automatic page generator.
- To add the option of live chatting.
- To add a blood test report giving option.

## REFERENCES

- [1] Waller, Daniel, et al., "Donating blood for research: a potential method for enhancing customer satisfaction of permanently deferred blood donors," *Blood Transfusion*, vol. 15, no. 16, p. 3, 2017.
- [2] Guglielmetti Mugion, Roberta, et al., "Promoting the propensity for blood donation through the understanding of its determinants," *BMC Health Services Research*, vol. 2, no. 3, pp. 2-3, 2021.
- [3] A. M. S. P. C. Stephen Harrell, "Promoting blood donation through social media: Evidence from Brazil, India and the USA," *Social Science & Medicine*, vol. 315, no. 213, pp. 4-5, December 2022.
- [4] D. M. C. C. Andy Ngo, "Blood Banking and Transfusion Medicine Challenges During the COVID-19 Pandemic," *National Library of Medicine*, vol. 2, no. 4, 2020.
- [5] M. L. Science, "My Personal Blood Bank Notebook: A fun blood Bank Journal filled with promotes ranging from silly to fun facts for any medical laboratory scientist or student," *International Kindle Paperwhite*, vol. 12, no. 4, p. 120, 2019.

# APPENDIX

## Login Controller Codes

```
<?php
namespace App\Http\Controllers\Auth;
use App\Http\Controllers\Controller;
use App\Providers\RouteServiceProvider;
use Illuminate\Foundation\Auth\AuthenticatesUsers;
use Illuminate\Support\Facades\Auth;
class LoginController extends Controller
{
    /*
    |-----
    | Login Controller
    |-----
    */
    use AuthenticatesUsers;
    /**
     * Where to redirect users after login.
     * @var string
     */
    // protected $redirectTo = RouteServiceProvider::HOME;
    public function redirectTo()
    {
        $role = Auth::user()->role;
        switch ($role) {
            case 'admin':
                return '/admin/dashboard';
                break;
            default:
                return '/profile';
                break;
        }
    }
}
/**
 * Create a new controller instance.
 * @return void
 */
public function __construct()
{
    $this->middleware('guest')->except('logout');
}
}
```

## Registration Controller Codes

```
<?php
namespace App\Http\Controllers\Auth;
use App\Http\Controllers\Controller;
use App\Models\Donor;
use App\Models\Reciever;
use App\Providers\RouteServiceProvider;
use App\Models\User;
use Illuminate\Foundation\Auth\RegistersUsers;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;

class RegisterController extends Controller
{
    /**
     |-----
     | Register Controller
     |-----
     */

    use RegistersUsers;

    /**
     * Where to redirect users after registration.
     *
     * @var string
     */
    protected $redirectTo = RouteServiceProvider::HOME;

    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('guest');
    }

    /**
     * Get a validator for an incoming registration request.
     *
     * @param array $data
     */
}
```

```

* @return \Illuminate\Contracts\Validation\Validator
*/

public static $rules_save = [
    'class_subjects' => ['required', 'regex:/[0-9]([0-9]|-(?!-))+/'],
];
protected function validator(array $data)
{

    return Validator::make($data, [
        'name' => 'required|max:255',
        'email' => 'required|unique:users|max:255',
        'address' => 'required',
        'phone' => ['required', 'regex:/^(?:\+88|88)?(01[3-9]\d{8})$/'],
        'birthday' =>
        $data['role'] == 'donor' ? 'required|date|before:-18 years' : '',
        'password' => 'required|min:8|confirmed',
        'role' => 'required',
        'blood_group_id' =>
        $data['role'] == 'donor' ? 'required' : '',
        'password' => 'required|min:8|confirmed',
    ]);
}

/**
 * Create a new user instance after a valid registration.
 *
 * @param array $data
 * @return \App\Models\User
 */
protected function create(array $data)
{

    // valid phone number

    $user = User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'phone' => $data['phone'],
        'address' => $data['address'],
        'role' => $data['role'],
        'blood_group_id' => (int)$data['blood_group_id'] ?: null,
        'birthday' => $data['birthday'],
        'password' => Hash::make($data['password']),
    ]);
}

```

```
]);  
  
if ($user) {  
    if ($user->role == 'donor') {  
        Donor::create([  
            'user_id' => $user->id,  
        ]);  
    } else {  
        Reciever::create([  
            'user_id' => $user->id,  
        ]);  
    }  
}  
  
return $user;  
}  
}
```



## Blood Request Controller Codes

```
<?php
namespace App\Http\Controllers\reciever;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\BloodGroup;
use App\Models\BloodRequest;
use App\Models\InterestedDonor;
use Illuminate\Support\Facades\Auth;

class BloodRequestController extends Controller
{

    public function index()
    {
        $bloodGroups = BloodGroup::all();
        $bloodRequests = BloodRequest::where('active', 1)->latest()-
>join('blood_groups', 'blood_groups.id', '=', 'blood_requests.blood_group_id')-
>select('blood_requests.*', 'blood_groups.name as blood_group')-
>simplePaginate(10);

        return view('blood_request_page', [
            'bloodGroups' => $bloodGroups,
            'bloodRequests' => $bloodRequests
        ]);
    }

    public function bloodRequest(Request $request)
    {
        if ($request->isMethod('GET')) {
            $user = Auth::user();
            $bloodGroups = BloodGroup::all();
            return view('frontend.user.reciever.blood_request', ['user' => $user,
'bloodGroups' => $bloodGroups]);
        } else if ($request->isMethod('POST')) {
            // dd($request->all());
            $request->validate([
                'blood_group_id' => 'required',
                'location' => 'required',
                'phone' => 'required',
                'name' => 'required',
                'email' => 'required|email',
                'date' => 'required',
            ]);
        }
    }
}
```

```

        'time' => 'required',
        'description' => 'required',
    ]);

    // // dd((int)$request->blood_group_id);
    // dd($request->all());

    // // dd('valid');

    $bloodRequest = new BloodRequest();
    $bloodRequest->blood_group_id = (int)$request->blood_group_id;
    $bloodRequest->location = $request->location;
    $bloodRequest->latitude = $request->latitude;
    $bloodRequest->longitude = $request->longitude;
    $bloodRequest->phone = $request->phone;
    $bloodRequest->name = $request->name;
    $bloodRequest->only_show_to_authorized = $request->
>only_show_to_authorized === 'on' ? 1 : 0;
    $bloodRequest->email = $request->email;
    $bloodRequest->flag = $request->flag;
    $bloodRequest->date = $request->date;
    $bloodRequest->time = $request->time;
    $bloodRequest->description = $request->description;
    $bloodRequest->reciever_id = Auth::user()->id;
    $bloodRequest->save();

    return redirect()->route('user_profile')->with('success', 'Blood
Request Created Successfully');
    }
}

public function bloodRequestEdit(Request $request)
{
    if ($request->isMethod('GET')) {
        $user = Auth::user();
        $bloodGroups = BloodGroup::all();
        $bloodRequest = BloodRequest::where('id', $request->id)->first();
        return view('frontend.user.reciever.edit_blood_request', [
            'user' => $user, 'bloodGroups' => $bloodGroups, 'bloodRequest' =>
$bloodRequest
        ]);
    } else if ($request->isMethod('POST')) {
        $request->validate([
            'blood_group_id' => 'required',

```

```

        'location' => 'required',
        'phone' => 'required',
        'name' => 'required',
        'email' => 'required|email',
        'date' => 'required',
        'time' => 'required',
        'description' => 'required',
    ]);

    $bloodRequest = BloodRequest::where('id', $request->id)->first();
    $bloodRequest->blood_group_id = (int)$request->blood_group_id;
    $bloodRequest->location = $request->location;
    $bloodRequest->latitude = $request->latitude;
    $bloodRequest->longitude = $request->longitude;
    $bloodRequest->phone = $request->phone;
    $bloodRequest->name = $request->name;
    $bloodRequest->only_show_to_authorized = $request
    >only_show_to_authorized === 'on' ? 1 : 0;
    $bloodRequest->email = $request->email;
    $bloodRequest->flag = $request->flag;
    $bloodRequest->date = $request->date;
    $bloodRequest->time = $request->time;
    $bloodRequest->description = $request->description;
    $bloodRequest->reciever_id = Auth::user()->id;
    $bloodRequest->update();

    // return redirect()->back()->with('success', 'Blood Request Upated
    Successfully');

    return redirect()->route('user_profile')->with('success', 'Blood
    Request Upated Successfully');
}
}

public function delete(Request $request)
{
    $bloodRequest = BloodRequest::find($request->id);
    $bloodRequest->delete();
    return response()->json([
        'success' => 'Blood Request Deleted Successfully',
        'status' => 200,
    ]);
}
}

```

```

public function bloodManage($id)
{
    $bloodRequest = BloodRequest::find($id);
    $bloodRequest->managed = 1;
    $bloodRequest->save();
    return redirect()->back()->with('success', 'congratulation! You
successfully managed your blood!');
}

public function getBloodRequestById(Request $request)
{
    $bloodRequest = BloodRequest::find($request->id);
    $bloodGroup = BloodGroup::find($bloodRequest->blood_group_id);

    $bloodRequest->blood_group_name = $bloodGroup->name;

    return response()->json([
        'success' => 'Blood Request By Id Retrived Successfully',
        'data' => $bloodRequest,
        'status' => 200,
    ]);
}

public function interestedDonorRequest(Request $request)
{
    $user = Auth::user();

    if (!$user) {
        return response()->json([
            'success' => false,
            'message' => 'Unauthorized. Please Login First.',
            'status' => 401,
        ]);
    }

    if ($user->role === 'donor') {
        $bloodRequest = BloodRequest::find($request->id);

        $alreadyInetrested = InterestedDonor::where('interested_donor_id',
$user->id)->where('blood_request_id', $bloodRequest->id)->first();
        // dd($alreadyInetrested);
        if ($alreadyInetrested) {
            return response()->json([

```

```

        'success' => false,
        'message' => 'You Already Interested in This Blood Request',
        'status' => 405,
    ]);
}

if ($bloodRequest->reciever_id == $user->id) {
    return response()->json([
        'success' => false,
        'message' => 'You made this request, so you can not request
interest in it',
        'status' => 405,
    ]);
}

if ($user->blood_group_id === $bloodRequest->blood_group_id) {

    $interestedDonor = new InterestedDonor();
    $interestedDonor->interested_donor_id = $user->id;
    $interestedDonor->blood_request_id = $bloodRequest->id;
    $interestedDonor->save();

    return response()->json([
        'success' => true,
        'message' => 'Your Interested In This Blood Request has been
sent to the Reciever',
        'status' => 200,
    ]);
} else {
    return response()->json([
        'success' => false,
        'message' => 'Your blood group does not match with this blood
request',
        'status' => 405,
    ]);
}
} else {
    return response()->json([
        'success' => false,
        'message' => 'You are not a donor. Please change your role as
donor and try again',
        'status' => 405,
    ]);
}
}
}

```

```

public function requestAccetByDonor($id)
{
    $user = Auth::user();

    if ($user && $user->role == 'donor') {
        $bloodRequest = BloodRequest::find($id);

        $alreadyInetrested = InterestedDonor::where('interested_donor_id',
$user->id)->where('blood_request_id', $bloodRequest->id)->first();
        // dd($alreadyInetrested);
        if ($alreadyInetrested) {
            return redirect()->back()->with('error', 'You have already
accepted');
        }

        if ($bloodRequest->reciever_id == $user->id) {
            return redirect()->back()->with('error', 'You made this request,
so you can not request interest in it');
        }

        if ($user->blood_group_id === $bloodRequest->blood_group_id) {
            $interestedDonor = new InterestedDonor();
            $interestedDonor->interested_donor_id = $user->id;
            $interestedDonor->blood_request_id = $bloodRequest->id;
            $interestedDonor->save();
            return redirect()->back()->with('success', 'Your Interested In
This Blood Request has been sent.');
```

```

        } else {
            return redirect()->back()->with('error', 'Your blood group does
not match with this blood request');
```

```

        }
    } else {
        return redirect()->back()->with('error', 'You are not a donor. Please
change your role as donor and try again');
```

```

    }
}

public function requestCancelByDonor($id)
{
    $user = Auth::user();

    if ($user && $user->role == 'donor') {
        $bloodRequest = BloodRequest::find($id);

```

```

        $alreadyInetrested = InterestedDonor::where('interested_donor_id',
$user->id)->where('blood_request_id', $bloodRequest->id)->first();

        // dd($alreadyInetrested);
        if ($alreadyInetrested && !$alreadyInetrested->accepted) {
            $alreadyInetrested->delete();
            return redirect()->back()->with('success', 'You have cancelled
the request');
        } else {
            return redirect()->back()->with('error', 'The Request already
accepted. You are not allowed to cancel this request');
        }
    } else {
        return redirect()->back()->with('error', 'You are not a donor. Please
change your role as donor and try again');
    }
}

```

```

public function interestedDonors(Request $request)
{

    $user = Auth::user();
    $interestedDonor = [];

    if ($user->role == 'reciever') {
        $interestedDonor = InterestedDonor::where('blood_request_id',
$request->id)->join(
            'users',
            'interested_donors.interested_donor_id',
            '=',
            'users.id'
        )->select(
            'interested_donors.*',
            'users.name as donor_name',
            'users.phone as donor_phone',
            'users.email as donor_email'
        )->get();

        return response()->json([
            'success' => true,
            'data' => $interestedDonor,

```

```

        'status' => 200,
    ]);
}

$interestedDonor = InterestedDonor::where('blood_request_id', $request-
>id)->join(
    'users',
    'interested_donors.interested_donor_id',
    '=',
    'users.id'
)->select(
    'interested_donors.*',
    'users.name as donor_name',
    'users.phone as donor_phone',
    'users.email as donor_email'
)->get();

// dd($interestedDonor);

return response()->json([
    'success' => true,
    'data' => $interestedDonor,
    'status' => 200,
]);
}

public function acceptInterestedDonors(Request $request)
{
    $interestedDonor = InterestedDonor::where('blood_request_id', $request-
>bloodRequestId)->where('interested_donor_id', $request->donorId)->first();

    if ($interestedDonor) {
        if ($interestedDonor->accepted == 1) {
            $interestedDonor->accepted = 0;
            $interestedDonor->update();

            return response()->json([
                'success' => true,
                'message' => 'canceled',
                'status' => 200,
            ]);
        } else {
            $interestedDonor->accepted = 1;
            $interestedDonor->update();
        }
    }
}

```



```
        return response()->json([
            'success' => true,
            'message' => 'accepted',
            'status' => 200,
        ]);
    }
} else {
    return response()->json([
        'success' => false,
        'message' => 'Something went wrong',
        'status' => 405,
    ]);
}
}
}
```