IBM PROJECT REPORT

PLASMA DONOR APPLICATION

TEAM ID: PNT2022TMID30529

Submitted by

ARUNA.P	**Reg. No: (613019104006)**
ARUNA. P	**Reg. No: (613019104007)**
DEVASHREE. R	**Reg. No: (613019104015)**
KAVYA. R	**Reg. No: (613019104037)**

TABLE OF CONTENTS

S.NO:	CONTENT	PAGE NO:
1.	**Introduction**	**4**
	1.1 Project Overview	4
	1.2 Purpose	4
2.	**Literature Survey**	**5**
	2.1 Exiting Problems	5
	2.2 References	10
	2.3 Problem Statement Definition	11
3.	**IDEATION AND PROPOSED SOLUTION**	**12**
	3.1 Empathy Map Canvas	12
	3.2 Ideation and Brainstorming	13
	3.3 Proposed Solution	14
	3.4 Problem solution Fit	14
4.	**REQUIREMENT ANALAYSIS**	**16**
	4.1 Functional Requirements	**16**
	4.2 Non-functional Requirements	16
5.	**PROJECT DESIGN**	**17**
	5.1 Data Flow Diagrams	17
	5.2 Solution &Technical Architecture	21
	5.3 User Stories	22
6.	**PROJECT PLANNING & SCHEDULING**	**23**
	6.1 Sprint Planning and Estimation	23
	6.2 Sprint delivery Schedule	24

	6.3 Reports From JIRA	25
7.	**CODING & SOLUTIONING**	**26**
	7.1 Feature 1	26
	7.2 Feature 2	44
8.	**TESTING**	**48**
	8.1 Test Cases	48
	8.2 User Acceptance Testing	49
9.	**RESULTS**	**50**
	9.1 Performance Metrics	50
10.	**ADVANTAGES & DISADVANTAGES**	**51**
11.	**CONCLUSION**	**52**
12.	**FUTURE SCOPE**	**53**
13.	**APPENDIX**	**54**
	Source Code	54
	Screenshots	67
	Github & Project Demo Link	75

PLASMA DONOR APPLICATION

ABSTRACT

During the COVID 19 crisis, the requirement of plasma became a high priority and the donor count has become low. Saving the donor information and helping the needy by notifying the current donors list, would be a helping hand. In regard to the problem faced, an application is to be built which would take the donor details, store them and inform them upon a request. This plasma therapy is considered to be safe & promising. This system proposed here aims at connecting the donors & the patients by an online application. By using this application, the users can either raise a request for plasma donation or requirement. This system is used if anyone needs a Plasma Donor.

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Recently concern grows about the plasma donation for COVID-19 during the pandemic situation. This convalescent plasma was used to recover patients who are critically ill as it helps to grow antibodies on their body. Recent researches show that many people are willing to help someone in need through money, blood and plasma donation etc. but they find it difficult to identify and approach the needy people who are not aware of technological innovations, including the use of social media. Plasma is used to various infectious diseases and it is one of the oldest methods known as plasma therapy. Plasma therapy is a Process where blood is donated by recovered patients in order to establish anti bodies that fights the infection. This system comprises of Admin, user and donor where both can request for Plasma. The proposed method helps the users to check the availability of donors. A donor has to register to the website providing their details. The registered users can get the information about the donor count of each blood group. The database will have all the details such as name, email, phone number, infected status. Whenever a user requests for a particular blood group then the concerned blood group donors will receive the notification regarding the requirement. For instance, during COVID 19 crisis the requirement for plasma increased drastically as there were no vaccination found in order to treat the infected patients, with plasma therapy the recovery rates where high but the donor count was very low and in such situations it was very important to get the information about the plasma donors. Saving the donor information and notifying about the current donors would be a helping hand as it can save time and help the users to track down the necessary information about the donors.

1.2 PURPOSE

The main aim of developing this system is to provide blood to the people who are in need of plasma. The numbers of persons who are in need of plasma are increasing in large number day by day. Using this system user can search blood group available in the city and he can also get contact number of the donor who has the same blood group he/she needs for plasma. In order to help people who are in need of plasma, this plasma donor application can be used effectively for getting the details of available plasma and user can also get contact number of the plasma donors having the same blood group and within the same city.

2. LITERATURE REVIEW

2.1 EXISTING PROBLEM

2.1.1 TITLE : Instant Plasma Donor Recipient Connector web application

AUTHOR: Kalpana Devi Guntoju*1, Tejaswini Jalli*2

The world is suffering from the COVID 19 crisis and no vaccine has been found yet.. But there is another scientific way in which we can help reduce mortality or help people affected by COVID19 by donating plasma from recovered patients. In the absence of an approved antiviral treatment plan for a fatal COVID19 infection, plasma therapy is an experimental approach to treat COVID19-positive patients and help them faster recovery. Therapy is considered competent. In the recommendation system, the donor who wants to donate plasma can donate by uploading their COVID19 certificate and the blood bank can see the donors who have uploaded the certificate and they can make a request to the donor and the hospital can register/login and search for the necessary things. plasma from a blood bank and they can request a blood bank and obtain plasma from the blood bank. The main goal of our project is to design a user-friendly web application that is like a scientific vehicle from which we can help reduce mortality or help those affected by COVID19 by donating plasma from patients who have recovered without approved antiretroviral therapy planning for a deadly COVID19 infection, plasma therapy is an experimental approach to treat those COVID-positive patients and help them recover faster. Therapy, which is considered reliable and safe. If a particular person has fully recovered from COVID19, theyare eligible to donate their plasma. As we all know, the traditional methods of finding plasma, one has to find out for oneself by looking at hospital records and contacting donors have been recovered, sometimes may not be available at home and move to other places. In this type of scenario, the health of those who are sick becomes disastrous. Therefore, it is not considered a rapid process to find plasma.

2.1.2 TITLE : A Web Application to Manage All Blood Donation and Transfusion Processes

AUTHOR: Rehab S. Ali,1 Tamer F. Hafez,2 Ali Badawey Ali

Many lives could be lost due to the difficulty in obtaining a proper blood bag, Therefore, this work aims to help citizens fulfill their needs for a safe and reliable blood group by searching for and locating a specific blood group. In this paper, we illustrate the problem of the blood bags shortage which is represented in the uncontrolled blood banks and parallel markets, lack of awareness and confidence, disappearance of the rare blood groups, and the difficulty in finding a specific blood group. Hence, we proposed the Blood Bag web-based application that is connected to a centralized database to gather and organize the data from all blood banks and blood donation campaigns. The proposed application organizes and controls the whole critical processes related to blood donation, testing and storage of blood bags, and delivering it to the patient. One blood bag can save a life during surgeries or road accidents, etc. Usually patients or their families look for a specific blood group they indeed need in the blood banks but they normally cannot find it due to the shortage of blood bags. This is because of the fear of donating blood and the misconception that donating blood is harmful and transmits diseases. This is one of the obstacles to provide the blood bags. The availability of the blood bags is critical because of the high proportion of patients with renal failure, some cases of birth, surgeries processes and incidents that need to get the blood as soon as possible to save these cases' lives. The blood bank is the pool of different blood groups where keeping a stockpile of blood to be distributed in case. The matched blood groups for a safe transfusion . Accidents (or any medical emergency and compensation of blood missing from the body), and keeping the blood in the freezer temperature.

2.1.3 **TITLE :** Developing a plasma donor application using Function-as-a-service in AWS

AUTHOR: Aishwarya R Gowri

Plasma is a liquid portion of the blood, over 55% of human blood is plasma. Plasma is used to treat various infectious diseases and it is one of the oldest methods known as plasma therapy. Plasma therapy is a process where blood is donated by recovered patients in order to establish an antibody that fights the infection. In this project plasma donor application is being developed by using AWS services. The services used are AWS Lambda, API gateway, Dynamo DB, AWS Elastic Compute Cloud with the help of these AWS services, it eliminates the need of configuring the servers and reduces the infrastructural costs associated with it and helps to achieve serverless computing. For instance, during COVID 19 crisis the requirement for plasma increased drastically as there were no vaccination found in order to treat the infected patients, with plasma therapy the recovery rates where high but the donor count was very low and in such situations it was very important to get the information about the plasma donors. Saving the donor information and notifying about the current donors would be a helping hand as it can save time and help the users to track down the necessary information about the donors. The proposed method helps the users to check the availability of donors. A donor has to register to the website providing their details. The registered users can get the information about the donor count of each blood group. The database will have all the details such as name, email, phone number, infected status. Whenever a user requests for a particular blood group then the concerned blood group donors will receive the notification regarding the requirement. A Json code is written to store the information, to fetch the requested information in lambda.

2.1.4 TITLE : Nearest Blood & Plasma Donor Finding: A Machine Learning Approach

AUTHOR: Nayan Das, Asif Iqbal.

The necessity of blood has become a significant concern in the present context all over the world. Due to a shortage of blood, people couldn't save themselves or their friends and family members. A bag of blood can save a precious life. Statistics show that a tremendous amount of blood is needed yearly because of major operations, road accidents, blood disorders, including Anemia, Hemophilia, and acute viral infections like Dengue, etc. Approximately 85 million people require single or multiple blood transfusions for treatment. Voluntary blood donors per 1,000 population of some countries are quite promising, such as Switzerland (113/1,000), Japan (70/1,000), while others have an unsatisfying result like India has 4/1,000, and Bangladesh has 5/1000. Recently a lifethreatening virus, COVID-19, spreading throughout the globe, which is more vulnerable for older people and those with pre-existing medical conditions. For them, plasma is needed to recover their illness. Our Purpose is to build a platform with clustering algorithms which will jointly help to provide the quickest solution to find blood or plasma donor. Closest blood or plasma donors of the same group in a particular area can be explored within less time and more efficiently. Keywords—Blood donation, Plasma donation, Kmeans clustering, Labeled Agglomerative clustering. Different methods have been used to solve this problem. This time, we have tried another way, a clustering approach, to solve the problem by grouping every user into small groups. This unsupervised machine learning approach is much faster and effective. In section II, we will discuss related work done previously to solve this problem. In section III, clustering algorithms relating to our project will explicitly be discussed. In section IV, our proposed method will be presented. In section V, we will analyze our experiment result.

2.1.5 **TITLE :** Securing Information on a Web Application System to Facilitate Online Blood Donation Booking

AUTHOR: Hrishitva Patel

Blood donation has saved many lives in the past. According to the American Red Cross statistics, a patient needs a blood transfusion every two seconds. Many benefits arise from blood donation to both the donor and the blood recipients. With blood donation, cancer patients, people involved in accidents, or those battling diseases that require blood donation have access to enough blood to sustain their survival. There is a need to digitize the blood donation booking to facilitate blood donation across the United States, and ensure patients in need of blood, receive their donation from eligible donors on time. This report demonstrates the security measures implemented to secure patient and blood donor data on a blood donation booking web application. Blood is donated for different reasons in hospitals and other blood banks. It is essential to help blood recipients survive surgeries, cancer treatment, and chronic illnesses, among other illnesses. The World Health Organization describes blood as the most precious gift a person can give to a person in need of it. Blood donated comprises four components: platelets, plasma, white blood cells, and red blood cells. Cancer patients require a blood transfusion to enhance platelets back into the body after radiation therapy. With a developed web application to book a blood donation session, it is easier for hospitals to know which blood component they need most and thus inform the system administrator to prompt for more blood donors.

2.2 REFERENCES

1. Kalpana Devi Guntoju*1, Tejaswini Jalli*2, Instant Plasma Donor Recipient Connector web application, 2022.
2. Rehab S. Ali,1 Tamer F. Hafez,2 Ali Badawey Ali, A Web Application to Manage All Blood Donation and Transfusion Processes, 2017.
3. Aishwarya R Gowri, Developing a plasma donor application using Function-as-a-service in AWS, 2020.
4. Nearest Blood & Plasma Donor Finding: A Machine Learning Approach, 2021
5. Hrishitva Patel, Securing Information on a Web Application System to Facilitate Online Blood Donation Booking, 2022.

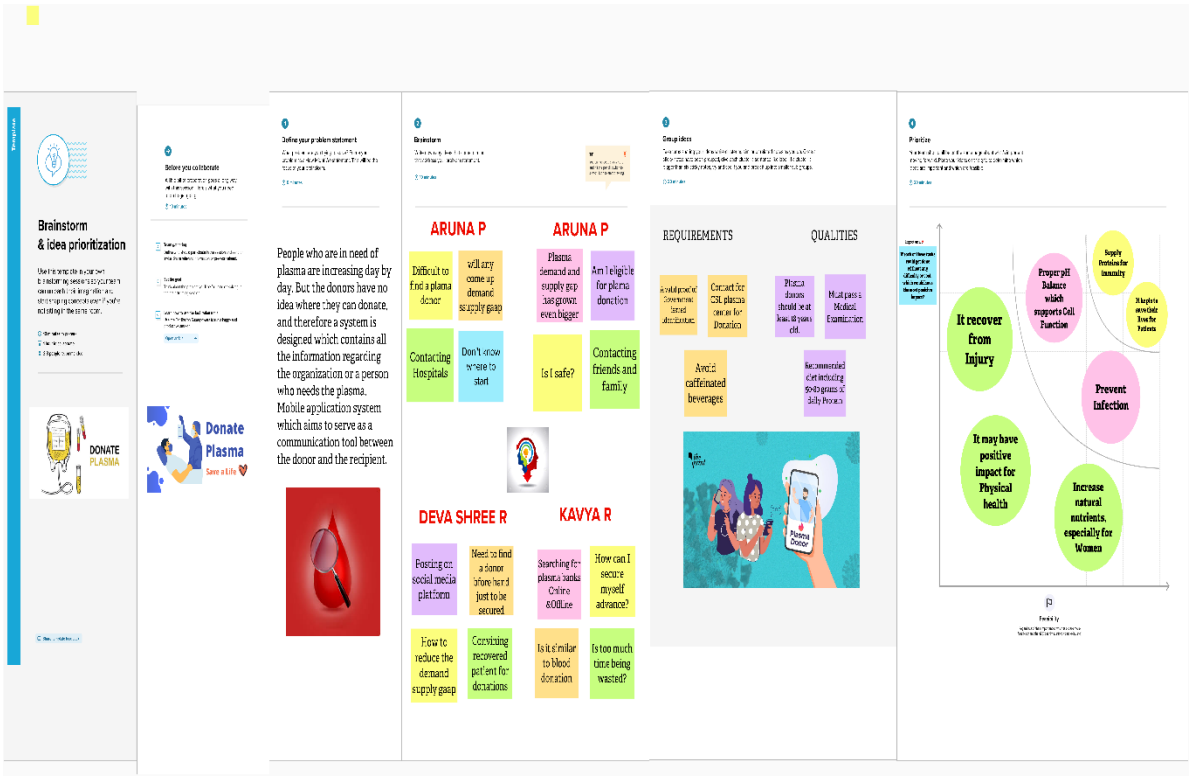
2.3 PROBLEM STATEMENT DEFINITION

In critical or emergency situations where accident occurs or during on-going treatments and surgeries etc there is urgent need for specific blood group. It requires lot of time to make the blood available and it is inconvenient during emergency situation, some rare blood groups are time consuming and difficult to arrange which are O- , AB- etc. In our country there is less awareness of blood donation, near about 20% of Indian population donates blood. In existing system the blood bank management system exhibited at a lot of ineffectiveness and inefficiency that had fetched impact taken by management. The system which was manual that is based on paper card to collect blood donor data, keep record of blood donors and disseminate results to blood donors, had weakness that needed IT based solutions.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

3.2 IDEATION & BRAINSTORMING



3.3 PROPOSED SOLUTION

The new idea will improve the existing system and it will move from conventional desktop system to mobile system. This paper introduces new features of improved system over existing system in many aspects. The proposed plasma donor application helps the people who are in need of plasma by giving them all details of plasma availability or regarding the donors with the same blood group. This is a web application allows you to access the whole information about plasma donor application, readily scalable and adaptable to meet the complex need of plasma Who are Key Facilitator for the Healthcare Sector, it also supports all the functionalities of plasma donor application.

3.4 PROBLEM SOLUTION FIT

In the emergency condition, sometimes it becomes very much difficult to look for the exact match of blood group of donor and acceptor. It may lead to delay in transaction of plasma within the specified amount of time. This application is providing each entity the facility to approach nearby blood donors so that it will become much easier to search rare blood groups in the hour of need.

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

- **Admin**

Admin can manage both donors and users. Admin has the only responsibility maintain and stored the record.

- **Users**

From this module user can create their account, when user create his account the user get a user id and password which identifies him uniquely. From this module user can search donor for blood.

- **Donors Registration**

In this module, people who are interested in donating blood get registered in this site and give his overall details related to donor. User details contain name, address, city, gender, blood group, location, contact number etc,.

- **Donor Search**

The people who are in need of blood can search in our site for getting the details of donors having the same blood group and within the same city.

- **Notification**

In this module, notification sends to donors for emergency. SMS send to registered donors phone number.

4.2 NON-FUNCTIONAL REQUIREMENTS

Usability

The system shall allow the users to access the system with pc using web application. The system uses a web application as an interface. The system is user friendly which makes the system easy

Availability

The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

Scalability

Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes in application and system processing demands.

Security

A security requirement is a statement of needed security functionality that ensures one of many different security properties of software is being satisfied.

Performance

The information is refreshed depending upon whether some updates have occurred or not in the application. The system shall respond to the member in not less than two seconds from the time of the request submittal. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 5 seconds to appear on the screen.

Reliability

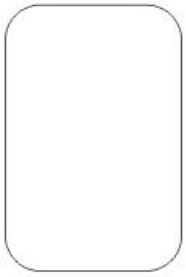
The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week. 24 hours a day.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

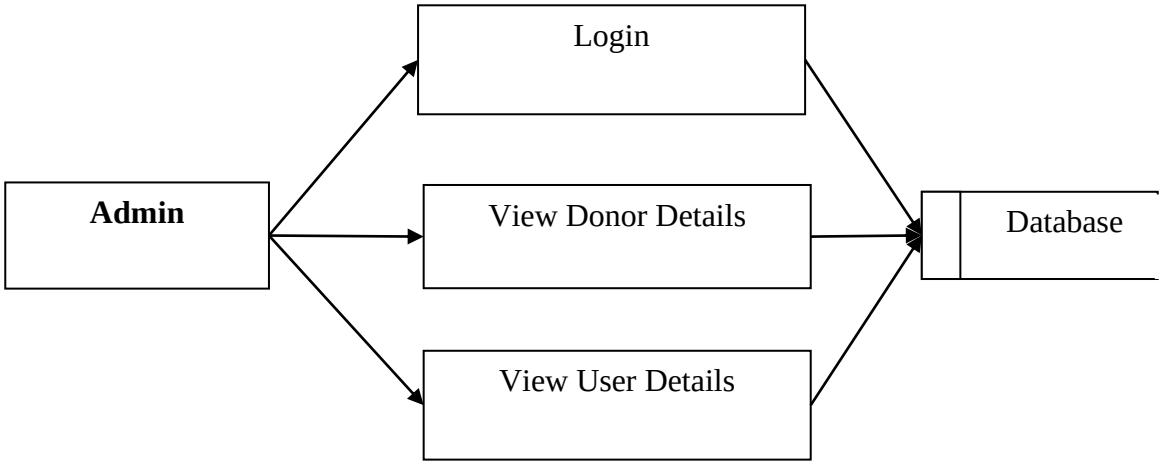
A data-flow diagram is a visual representation of how data moves through a system or a process (usually an information system). The DFD additionally gives details about each entity's inputs and outputs as well as the process itself. A data-flow diagram lacks control flow, loops, and decision-making processes. Using a flowchart, certain operations depending on the data may be depicted.

Data flow Symbols:

Symbol	Description
	An **entity**. A source of data or a destination for data.
	A **process** or task that is performed by the system.
	A **data store**, a place where data is held between processes.
	A **data flow**.

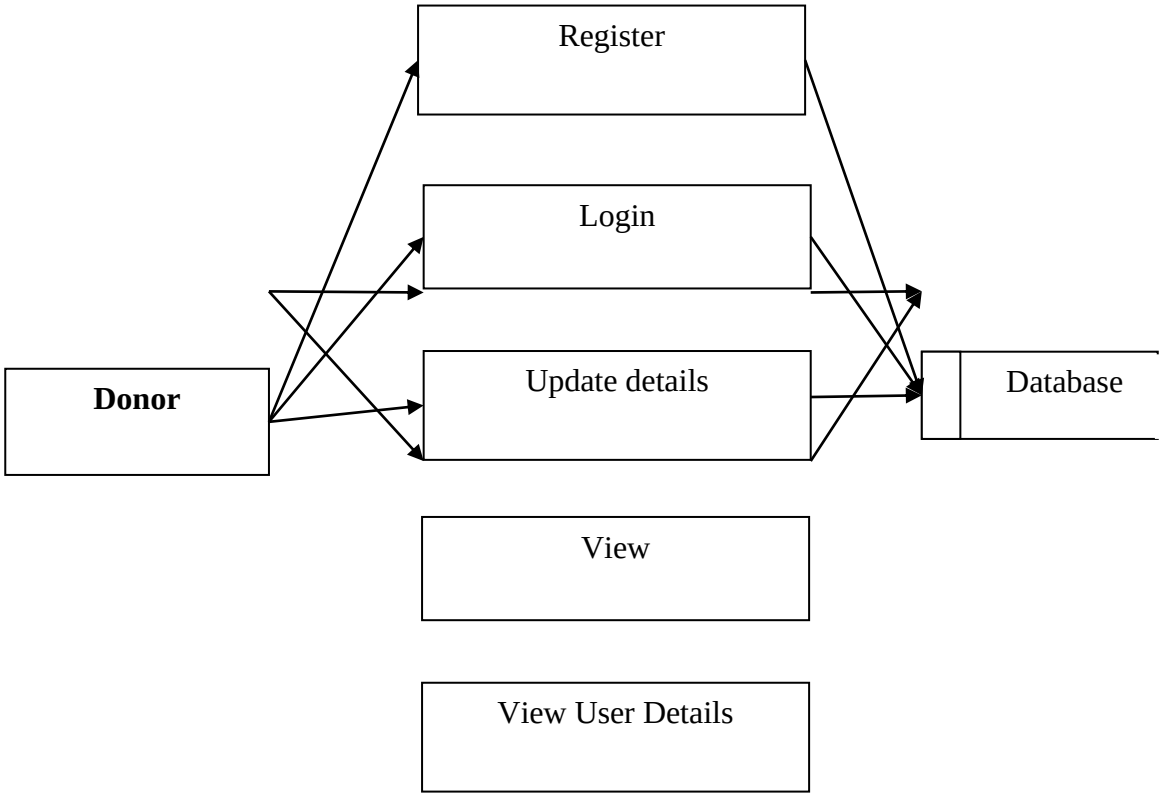
LEVEL 0

The Level 0 DFD shows how the system is divided into 'sub-systems' (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.



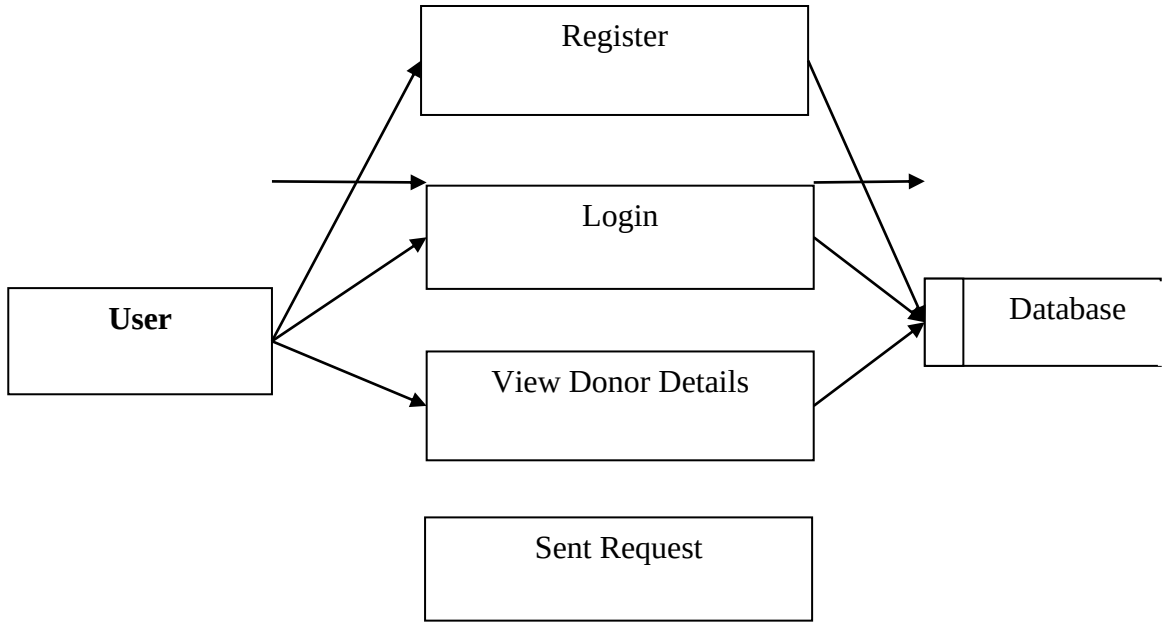
LEVEL 1

The next stage is to create the Level 1 Data Flow Diagram. This highlights the main functions carried out by the system. As a rule, to describe the system was using between two and seven functions - two being a simple system and seven being a complicated system. This enables us to keep the model manageable on screen or paper.

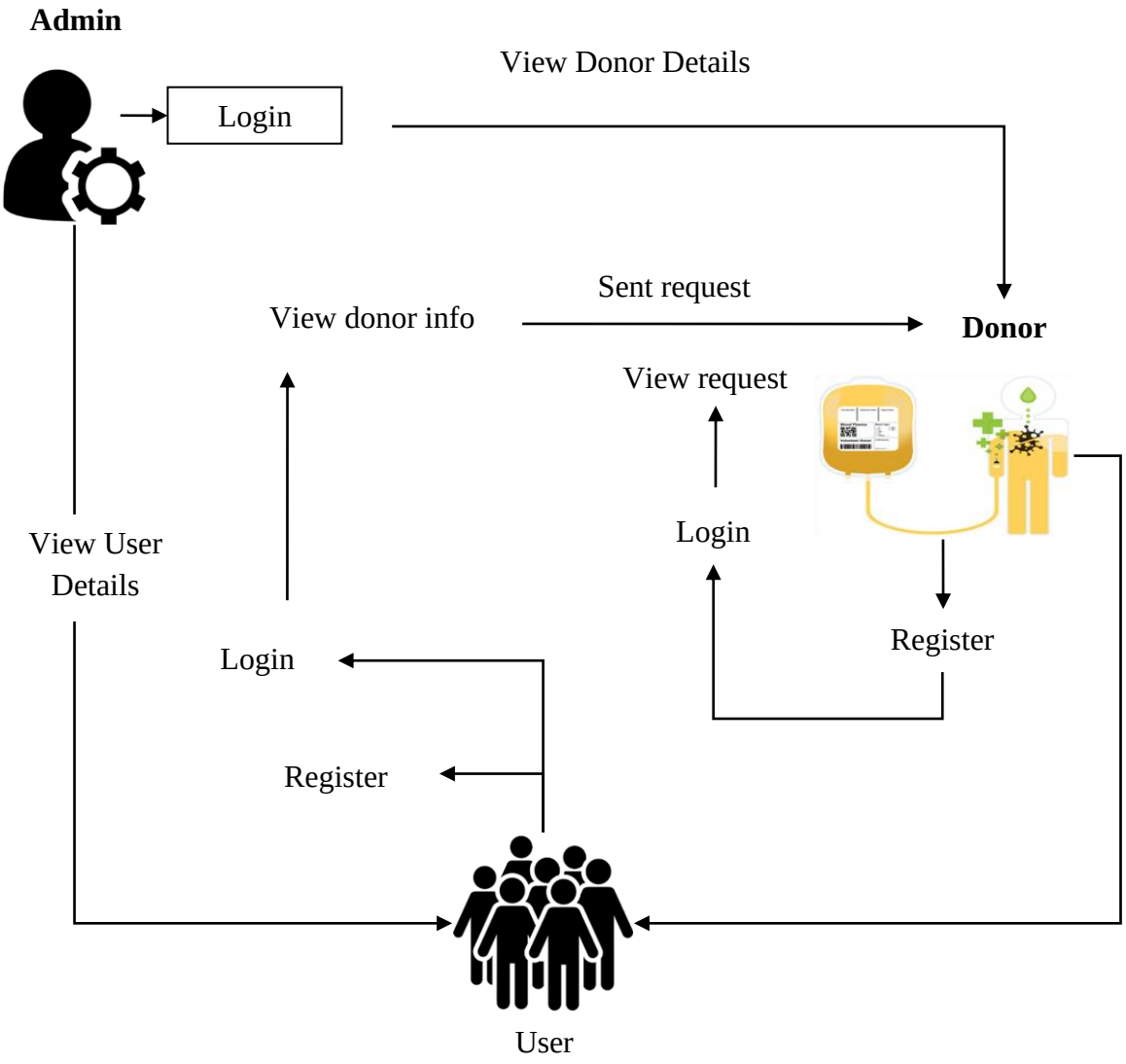


LEVEL 2

A Data Flow Diagram (DFD) tracks processes and their data paths within the business or system boundary under investigation. A DFD defines each domain boundary and illustrates the logical movement and transformation of data within the defined boundary. The diagram shows 'what' input data enters the domain, 'what' logical processes the domain applies to that data, and 'what' output data leaves the domain. Essentially, a DFD is a tool for process modeling and one of the oldest.



5.2 SOLUTION & TECHNICAL ARCHITECTURE



5.3 USER STORIES

The website works great, it is smooth and visually appealing. I am excited as a first-time donor to take advantage of the features offers and also, I can share the website for those who are in need of donor. From this website both the donor and recipient can find the place.

As a receptor, this website helps me lot. I got my donor on time and now I got cured completely. Not only me many got benefits by this website.

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

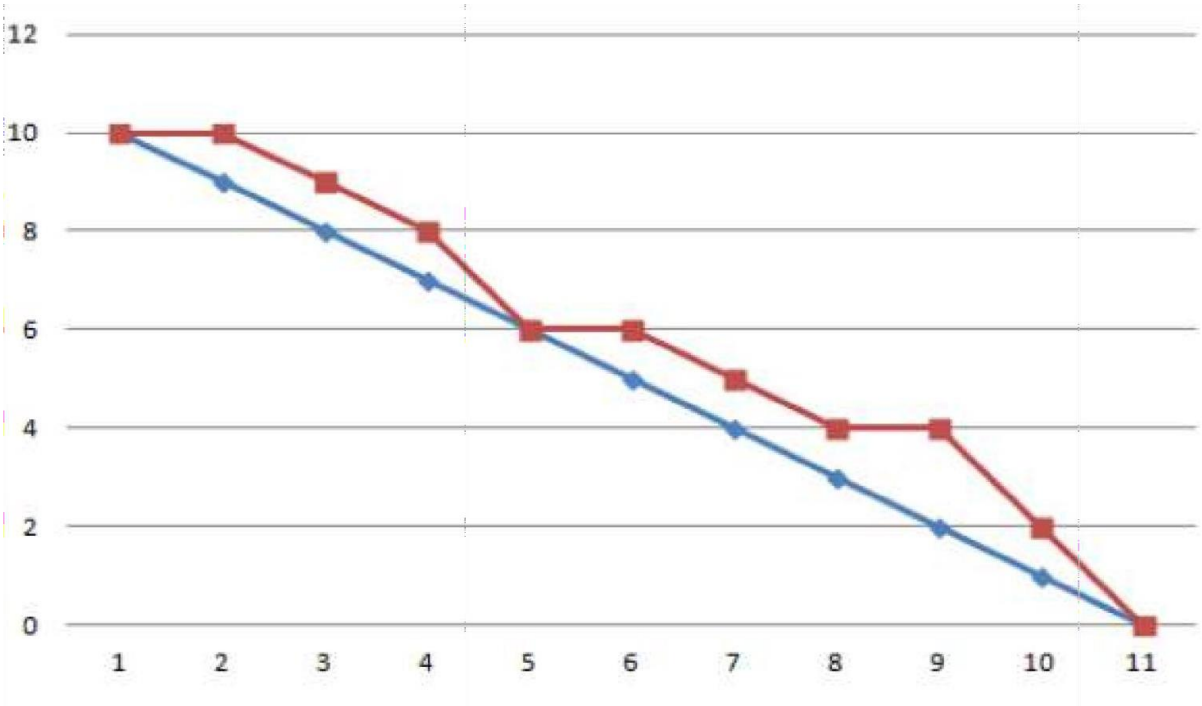
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	13
Sprint-2	Confirmation	USN-2	As a user, I will receive confirmation email once Ihave registered for the application	13
Sprint-1	Registration through Google account	USN-3	As a user, I can register for the applicationthrough google account	8
Sprint-3	Search for donor	USN-4	I can view list represents a specific donor withdonor details	13
Sprint-1	Login	USN-5	As a user, I can log into the application byentering email & password	13
Sprint-2	Dashboard	USN-6	As a user, I can log in into the application and viewthe dashboard for plasma information's.	8
Sprint-2	Notification	USN-7	As a user,I can get notifications after register forplasma donation/needy.	13
Sprint-3	Database	USN-8	Admin can access, view, modify, update alldetails of the plasma donor application	20
Sprint-4	Software testing And deployment	USN-9	As user want to access the application without anydrawbacks we need to test the software before release.	13

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	13	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	13	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	13	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	8	19 Nov 2022

Velocity:
Sprint 1(AV) =2.16
Sprint 2(AV) =2.16
Sprint 3(AV) =2.16
Sprint 4(AV) =1.33

6.3 REPORTS FROM JIRA



7. CODING & SOLUTIONING

7.1 FEATURE 1

```
<html>
<head>
<title>Plasma Donor</title>
<!-- for-mobile-apps -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="keywords" content="Clinical Lab Responsive web template,
Bootstrap Web Templates, Flat Web Templates, Android Compatible web
template,
Smartphone Compatible web template, free webdesigns for Nokia, Samsung,
LG, SonyEricsson, Motorola web design" />
<script type="application/x-javascript"> addEventListener("load", function() {
setTimeout(hideURLbar, 0); }, false);
                function hideURLbar(){ window.scrollTo(0,1); } </script>
<!-- //for-mobile-apps -->
<link href="css/bootstrap.css" rel="stylesheet" type="text/css" media="all" />
        <style>
                /*!
 * Bootstrap v3.3.4 (http://getbootstrap.com)
 * Copyright 2011-2015 Twitter, Inc.
 * Licensed under MIT
(https://github.com/twbs/bootstrap/blob/master/LICENSE)
 */

/*! normalize.css v3.0.2 | MIT License | git.io/normalize */
html {
  font-family: sans-serif;
  -webkit-text-size-adjust: 100%;
     -ms-text-size-adjust: 100%;
}
body {
```

```
  margin: 0;
}
article,
aside,
details,
figcaption,
figure,
footer,
header,
hgroup,
main,
menu,
nav,
section,
summary {
  display: block;
}
audio,
canvas,
progress,
video {
  display: inline-block;
  vertical-align: baseline;
}
audio:not([controls]) {
  display: none;
  height: 0;
}
[hidden],
template {
  display: none;
}
a {
  background-color: transparent;
```

```
}
a:active,
a:hover {
  outline: 0;
}
abbr[title] {
  border-bottom: 1px dotted;
}
b,
strong {
  font-weight: bold;
}
dfn {
  font-style: italic;
}
h1 {
  margin: .67em 0;
  font-size: 2em;
}
mark {
  color: #000;
  background: #ff0;
}
small {
  font-size: 80%;
}
sub,
sup {
  position: relative;
  font-size: 75%;
  line-height: 0;
  vertical-align: baseline;
}
sup {
```

```
  top: -.5em;
}
sub {
  bottom: -.25em;
}
img {
  border: 0;
}
svg:not(:root) {
  overflow: hidden;
}
figure {
  margin: 1em 40px;
}
hr {
  height: 0;
  -webkit-box-sizing: content-box;
     -moz-box-sizing: content-box;
          box-sizing: content-box;
}
pre {
  overflow: auto;
}
code,
kbd,
pre,
samp {
  font-family: monospace, monospace;
  font-size: 1em;
}
button,
input,
optgroup,
select,
```

```
textarea {
  margin: 0;
  font: inherit;
  color: inherit;
}
button {
  overflow: visible;
}
button,
select {
  text-transform: none;
}
button,
html input[type="button"],
input[type="reset"],
input[type="submit"] {
  -webkit-appearance: button;
  cursor: pointer;
}
button[disabled],
html input[disabled] {
  cursor: default;
}
button::-moz-focus-inner,
input::-moz-focus-inner {
  padding: 0;
  border: 0;
}
input {
  line-height: normal;
}
input[type="checkbox"],
input[type="radio"] {
  -webkit-box-sizing: border-box;
```

```
    -moz-box-sizing: border-box;
         box-sizing: border-box;
  padding: 0;
}
input[type="number"]::-webkit-inner-spin-button,
input[type="number"]::-webkit-outer-spin-button {
  height: auto;
}
input[type="search"] {
  -webkit-box-sizing: content-box;
    -moz-box-sizing: content-box;
         box-sizing: content-box;
  -webkit-appearance: textfield;
}
input[type="search"]::-webkit-search-cancel-button,
input[type="search"]::-webkit-search-decoration {
  -webkit-appearance: none;
}
fieldset {
  padding: .35em .625em .75em;
  margin: 0 2px;
  border: 1px solid #c0c0c0;
}
legend {
  padding: 0;
  border: 0;
}
textarea {
  overflow: auto;
}
optgroup {
  font-weight: bold;
}
table {
```

```
 border-spacing: 0;
 border-collapse: collapse;
}
td,
th {
 padding: 0;
}
/*! Source: https://github.com/h5bp/html5-
boilerplate/blob/master/src/css/main.css */
@media print {
 *,
 *:before,
 *:after {
  color: #000 !important;
  text-shadow: none !important;
  background: transparent !important;
  -webkit-box-shadow: none !important;
          box-shadow: none !important;
 }
 a,
 a:visited {
  text-decoration: underline;
 }
 a[href]:after {
  content: " (" attr(href) ")";
 }
 abbr[title]:after {
  content: " (" attr(title) ")";
 }
 a[href^="#"]:after,
 a[href^="javascript:"]:after {
  content: "";
 }
 pre,
```

```
blockquote {
  border: 1px solid #999;

  page-break-inside: avoid;
}
thead {
  display: table-header-group;
}
tr,
img {
  page-break-inside: avoid;
}
img {
  max-width: 100% !important;
}
p,
h2,
h3 {
  orphans: 3;
  widows: 3;
}
h2,
h3 {
  page-break-after: avoid;
}
select {
  background: #fff !important;
}
.navbar {
  display: none;
}
.btn > .caret,
.dropup > .btn > .caret {
  border-top-color: #000 !important;
```

```
    }
    .label {
      border: 1px solid #000;
    }
    .table {
      border-collapse: collapse !important;
    }
    .table td,
    .table th {
      background-color: #fff !important;
    }
    .table-bordered th,
    .table-bordered td {
      border: 1px solid #ddd !important;
    }
}
    @media screen and (-webkit-min-device-pixel-ratio: 0) {
      input[type="date"],
      input[type="time"],
      input[type="datetime-local"],
      input[type="month"] {
        line-height: 34px;
      }
      input[type="date"].input-sm,
      input[type="time"].input-sm,
      input[type="datetime-local"].input-sm,
      input[type="month"].input-sm,
      .input-group-sm input[type="date"],
      .input-group-sm input[type="time"],
      .input-group-sm input[type="datetime-local"],
      .input-group-sm input[type="month"] {
        line-height: 30px;
      }
      input[type="date"].input-lg,
```

```
  input[type="time"].input-lg,
  input[type="datetime-local"].input-lg,
  input[type="month"].input-lg,
  .input-group-lg input[type="date"],
  .input-group-lg input[type="time"],
  .input-group-lg input[type="datetime-local"],
  .input-group-lg input[type="month"] {
    line-height: 46px;
  }
}
.form-group {
  margin-bottom: 15px;
}
.radio,
.checkbox {
  position: relative;
  display: block;
  margin-top: 10px;
  margin-bottom: 10px;
}
.radio label,
.checkbox label {
  min-height: 20px;
  padding-left: 20px;
  margin-bottom: 0;
  font-weight: normal;
  cursor: pointer;
}
.radio input[type="radio"],
.radio-inline input[type="radio"],
.checkbox input[type="checkbox"],
.checkbox-inline input[type="checkbox"] {
  position: absolute;
  margin-top: 4px \9;
```

```
  margin-left: -20px;
}
.radio + .radio,
.checkbox + .checkbox {
  margin-top: -5px;
}
.radio-inline,
.checkbox-inline {
  position: relative;
  display: inline-block;
  padding-left: 20px;
  margin-bottom: 0;
  font-weight: normal;
  vertical-align: middle;
  cursor: pointer;
}
.radio-inline + .radio-inline,
.checkbox-inline + .checkbox-inline {
  margin-top: 0;
  margin-left: 10px;
}
input[type="radio"][disabled],
input[type="checkbox"][disabled],
input[type="radio"].disabled,
input[type="checkbox"].disabled,
fieldset[disabled] input[type="radio"],
fieldset[disabled] input[type="checkbox"] {
  cursor: not-allowed;
}
.radio-inline.disabled,
.checkbox-inline.disabled,
fieldset[disabled] .radio-inline,
fieldset[disabled] .checkbox-inline {
  cursor: not-allowed;
```

```
}
.radio.disabled label,
.checkbox.disabled label,
fieldset[disabled] .radio label,
fieldset[disabled] .checkbox label {
  cursor: not-allowed;
}
.form-control-static {
  min-height: 34px;
  padding-top: 7px;
  padding-bottom: 7px;
  margin-bottom: 0;
}
.form-control-static.input-lg,
.form-control-static.input-sm {
  padding-right: 0;
  padding-left: 0;
}
.input-sm {
  height: 30px;
  padding: 5px 10px;
  font-size: 12px;
  line-height: 1.5;
  border-radius: 3px;
}
select.input-sm {
  height: 30px;
  line-height: 30px;
}
textarea.input-sm,
select[multiple].input-sm {
  height: auto;
}
.form-group-sm .form-control {
```

```
  height: 30px;
  padding: 5px 10px;
  font-size: 12px;
  line-height: 1.5;
  border-radius: 3px;
}
select.form-group-sm .form-control {
  height: 30px;
  line-height: 30px;
}
textarea.form-group-sm .form-control,
select[multiple].form-group-sm .form-control {
  height: auto;
}
.form-group-sm .form-control-static {
  height: 30px;
  min-height: 32px;
  padding: 5px 10px;
  font-size: 12px;
  line-height: 1.5;
}
.input-lg {
  height: 46px;
  padding: 10px 16px;
  font-size: 18px;
  line-height: 1.3333333;
  border-radius: 6px;
}
select.input-lg {
  height: 46px;
  line-height: 46px;
}
textarea.input-lg,
select[multiple].input-lg {
```

```
  height: auto;
}
        <div class="about all_pad">
            <div class="container">
                <h3 class="title">Search Donor</h3>

                <form id="form1" name="form1" method="post"
        action="/dsearch">

                        <div>
                          <table width="100%" border="0">
                  <tr>
                    <td width="25%"> </td>
                    <td width="21%"> </td>
                    <td width="29%"> </td>
                    <td width="25%"> </td>
                  </tr>
                  <tr>
                    <td> </td>
                    <td> </td>
                    <td> </td>
                    <td> </td>
                  </tr>
                  <tr>
                    <td> </td>
                    <td>Select Blood Group</td>

                    <td><select name="blood" id="blood">
                      <option value="A+">A+</option>
                      <option value="A-">A-</option>
                        <option value="B+">B+</option>
                        <option value="B-">B-</option>
                <option value="O+">O+</option>
                        <option value="O-">O-</option>
```

```
<option value="AB+">AB+</option>
        <option value="AB-">AB-</option>

    </select>                </td></td>
    <td> </td>
  </tr>
  <tr>
    <td> </td>
    <td>Info</td>

    <td><textarea name="info" id="info" required></textarea></td>
    <td> </td>
  </tr>
     <tr>
    <td> </td>
    <td> </td>
    <td> </td>
    <td> </td>
  </tr>
  <tr>
    <td> </td>
    <td> </td>
    <td><input name="submit" type="submit" value="Search" />
        <input name="submit" type="submit" value="SendMail" /></td>
    <td> </td>
  </tr>



  <tr>
    <td> </td>
    <td> </td>
    <td> </td>
    <td> </td>
  </tr>
```

```
            <tr>
              <td> </td>
              <td> </td>
              <td> </td>
              <td> </td>
            </tr>
          </table>

                </div>

                    <div>

                    </div>

<div>
                      <table width="80%" border="1" align="center">
          <tr>


            <td bgcolor="#3399FF"><div align="center"
class="style5"><strong>id</strong></div></td>
              <td bgcolor="#3399FF"><div align="center"
class="style5"><strong>Name</strong></div></td>
            <td bgcolor="#3399FF"><div align="center"
class="style5"><strong>Gender</strong></div></td>
            <td bgcolor="#3399FF"><div align="center"
class="style5"><strong>Age</strong></div></td>
            <td bgcolor="#3399FF"><div align="center"
class="style5"><strong>Email</strong></div></td>

            <td bgcolor="#3399FF"><div align="center"
class="style5"><strong>Phone</strong></div></td>
                                <td bgcolor="#3399FF"><div align="center"
class="style5"><strong>Address</strong></div></td>
```

```
                                <td bgcolor="#3399FF"><div align="center"
class="style5"><strong>BloodGroup</strong></div></td>

                                <td bgcolor="#3399FF"><div align="center"
class="style5"><strong>HealthIssues</strong></div></td>
  <td bgcolor="#3399FF"><div align="center"
class="style5"><strong>Action</strong></div></td>
          </tr>
                        {% for item in data %}
          <tr>
            <td><div align="center">{{item[0]}}</div></td>
           <td><div align="center">{{item[1]}}</div></td>

           <td><div align="center">{{item[2]}}</div></td>
           <td><div align="center">{{item[3]}}</div></td>
           <td><div align="center">{{item[4]}}</div></td>

           <td><div align="center">{{item[5]}}</div></td>
                                <td><div align="center">{{item[6]}}</div></td>
                               <td><div align="center">{{item[7]}}</div></td>
                                <td><div align="center">{{item[8]}}</div></td>
             <td><div align="center"><a
href="SendRequest?cid={{item[0]}}">SendRequest</a></div></td>

          </tr>
                        {% endfor %}

        </table>
</div>
</form>

        </div>
</div>
```

```
<!-- contact -->
<div class="contact_w3agile">
	<div class="container">

		<p class="agileinfo">&copy; All rights reserved | Design by <a
href="#">Plasma Donor</a></p>
	</div>
</div>
<!-- smooth scrolling -->
	<script type="text/javascript">
		$(document).ready(function() {
		/*
			var defaults = {
			containerID: 'toTop', // fading element id
			containerHoverID: 'toTopHover', // fading element hover id
			scrollSpeed: 1200,
			easingType: 'linear'
			};
		*/
		$().UItoTop({ easingType: 'easeOutQuart' });
		});
	</script>
	<a href="#" id="toTop" style="display: block;"> <span
id="toTopHover" style="opacity: 1;"> </span></a>
<!-- //smooth scrolling -->
<script type="text/javascript" src="static/js/bootstrap-3.1.1.min.js"></script>
</body>
</html>
```

7.2 FEATURE 2

```
<html>
<head>
<title>plasma donor </title>
<!-- for-mobile-apps -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="keywords" content="Clinical Lab Responsive web template,
Bootstrap Web Templates, Flat Web Templates, Android Compatible web
template,
Smartphone Compatible web template, free webdesigns for Nokia, Samsung,
LG, SonyEricsson, Motorola web design" />
<script type="application/x-javascript"> addEventListener("load", function() {
setTimeout(hideURLbar, 0); }, false);
                function hideURLbar(){ window.scrollTo(0,1); } </script>
<!-- //for-mobile-apps -->
<link href="css/bootstrap.css" rel="stylesheet" type="text/css" media="all" />
</script>
<script src="static/js/responsiveslides.min.js"></script>
</head>
<body>
<!-- header -->
<div class="header_w3l">
        <div class="container">
                <nav class="navbar navbar-default">
                        <div class="navbar-header">

                                <h1><a  href="#">Plasma  <span> Donor
</span></a></h1>
                        </div>
                                <!-- top-nav -->
                        <div class="collapse navbar-collapse" id="bs-example-
navbar-collapse-1">
                                <ul class="nav navbar-nav navbar-right">
```

```
                                        <li><a
href="/AdminHome">Home</a></li>
                                    <li><a href="/AdminDonorInfo"
>DonorInfo</a></li>

                                    <li><a href="/">Logout</a></li>
                                </ul>
                                <div class="clearfix"> </div>
                            </div>
                        </nav>
                </div>
</div>
<!-- header -->
<!-- banner -->
<div class="banner_w3ls page_head">

</div>
<!-- //banner -->
<div class="about all_pad">
        <div class="container">
                <h3 class="title">User Information!</h3>

                 <form id="form1" name="form1" method="post"
action="/userinfo">
          <table width="80%" border="1" align="center">
           <tr>


            <td bgcolor="#3399FF"><div align="center"
class="style5"><strong>Name</strong></div></td>
              <td bgcolor="#3399FF"><div align="center"
class="style5"><strong>Gender</strong></div></td>
```

```
            <td bgcolor="#3399FF"><div align="center"
class="style5"><strong>Age</strong></div></td>
            <td bgcolor="#3399FF"><div align="center"
class="style5"><strong>EmailId</strong></div></td>
            <td bgcolor="#3399FF"><div align="center"
class="style5"><strong>Phone</strong></div></td>
            <td bgcolor="#3399FF"><div align="center"
class="style5"><strong>UserName</strong></div></td>


          </tr>
                        {% for item in data %}
          <tr>
            <td><div align="center">{{item[0]}}</div></td>
            <td><div align="center">{{item[1]}}</div></td>

            <td><div align="center">{{item[2]}}</div></td>
            <td><div align="center">{{item[3]}}</div></td>
            <td><div align="center">{{item[4]}}</div></td>
            <td><div align="center">{{item[6]}}</div></td>


          </tr>
                        {% endfor %}
          <tr>


            <td> </td>
            <td> </td>
            <td> </td>
            <td> </td>
            <td> </td>
              <td> </td>
          </tr>
```

```
            </table>
        </form>

            </div>
        </div>

        <!-- contact -->
        <div class="contact_w3agile">
            <div class="container">

                <p class="agileinfo">&copy; All rights reserved | Design by <a
        href="#">plasma donor </a></p>
            </div>
        </div>
        <!-- smooth scrolling -->
            <script type="text/javascript">
                $(document).ready(function() {
                /*
                    var defaults = {
                    containerID: 'toTop', // fading element id
                    containerHoverID: 'toTopHover', // fading element hover id
                    scrollSpeed: 1200,
                    easingType: 'linear'
                    };
                */
                $().UItoTop({ easingType: 'easeOutQuart' });
                });
            </script>
            <a href="#" id="toTop" style="display: block;"> <span
        id="toTopHover" style="opacity: 1;"> </span></a>
        <!-- //smooth scrolling -->
        <script type="text/javascript" src="static/js/bootstrap-3.1.1.min.js"></script>
        </body>
</html>
```

8. TESTING

8.1 TEST CASES

A test case has components that describe input, action and an expected response, in order to determine if a feature of an application is working correctly. A test case is a set of instructions on “HOW” to validate a particular test objective/target, which when followed will tell us if the expected behavior of the system is satisfied or not.
Characteristics of a good test case:

- Accurate: Exacts the purpose.
- Economical: No unnecessary steps or words.
- Traceable: Capable of being traced to requirements.
- Repeatable: Can be used to perform the test over and over.
- Reusable: Can be reused if necessary.

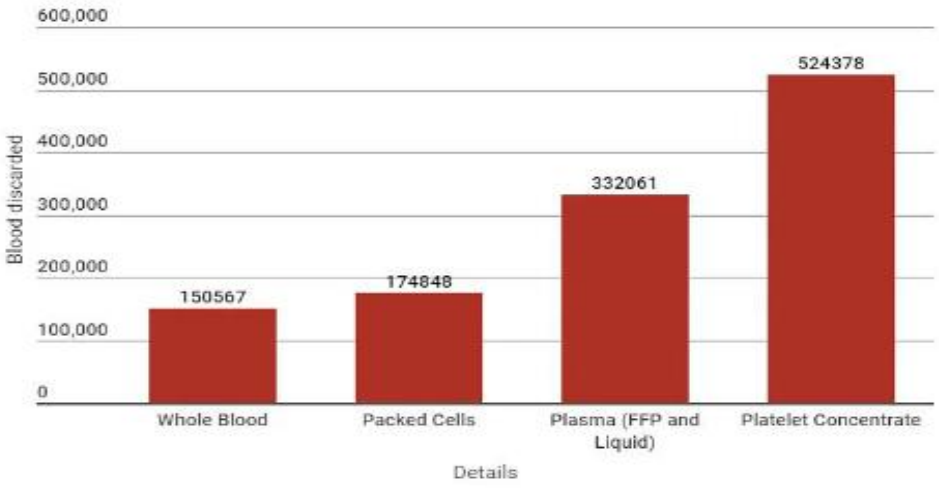
S.NO	Scenario	Input	Excepted output	Actual output
1	Admin Login Form	User name and password	Login	Login success.
2	Donor Registration Form	Donor basic details	Registration	Donor registration details stored in database.
3	User Registration Form	User basic details	Registration	User registration details stored in database.
4	User Login Form	User name and password	Login	Login success.

8.2 USER ACCEPTANCE TESTING

This is a type of testing done by users, customers, or other authorised entities to determine application/software needs and business processes. Acceptance testing is the most important phase of testing as this decides whether the client approves the application/software or not. It may involve functionality, usability, performance, and U.I of the application. It is also known as user acceptance testing (UAT), operational acceptance testing (OAT), and end-user testing.

9. RESULTS

9.1 PERFORMANCE METRICS



10. ADVANTAGES & DISADVANTAGES

ADVANTAGES

- It is a user-friendly application.
- The people in need of plasma can search for the donors by giving their blood group and city name.
- It saves time as he can search donors online without going anywhere.
- Using this system user can get plasma in time and can save and here our system work, whenever a person needs plasma user get information of the person who has the same blood group needs.

DISADVANTAGES

- It cannot auto verify user genuineness.
- It is time consuming
- It leads to error prone results
- It consumes lot of manpower to better results
- It lacks of data security
- Retrieval of data takes lot of time

11. CONCLUSION

This project is designed for successful completion of project on Plasma Donor Application system. The basic building aim is to provide plasma donation service to the city recently. Plasma Donor Application System is a Web based application that is designed to store, process, retrieve and analyze information concerned with the administrative and inventory management within a plasma. This project aims at maintaining all the information pertaining to plasma donors, different blood groups available in each plasma bank and helps them manage in a better way plasma donation system can collect plasma from many donators in short from various sources and distribute that plasma to needy people who require plasma. To do all this we require high quality Web Application to manage those jobs. Plasma application provides a reliable platform to connect local plasma donors with patients.

12. FUTURE SCOPE

This system is developed such a way that additional enhancement can be done without much difficulty. The renovation of the project would increase the flexibility of the system. In future, we can develop this project in android platform. We will add extra features like donor location tracking system (GPS), Feedback form, and enable call option etc.

13. APPENDIX

SOURCE CODE

```
from flask import Flask, render_template, flash, request, session,send_file
from flask import render_template, redirect, url_for, request
#from wtforms import Form, TextField, TextAreaField, validators, StringField, SubmitField
from werkzeug.utils import secure_filename
import datetime

from flask_mail import Mail, Message

import mysql.connector
import sys
app = Flask(__name__)
app.config['DEBUG']
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'

@app.route("/")
def homepage():

    return render_template('index.html')

@app.route("/AdminLogin")
def AdminLogin():

    return render_template('AdminLogin.html')



@app.route("/DonorLogin")
def DonorLogin():
    return render_template('DonorLogin.html')

@app.route("/NewDonor")
def NewDonor():
```

```
    return render_template('NewDonor.html')
@app.route("/UserLogin")
def UserLogin():
    return render_template('UserLogin.html')

@app.route("/PersonalInfo")
def PersonalInfo():
    return render_template('DonorPersonal.html')



@app.route("/NewUser")
def NewUser():
    return render_template('NewUser.html')

@app.route("/AdminHome")
def AdminHome():
    conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
    cur = conn.cursor()
    cur.execute("SELECT * FROM regtb ")
    data = cur.fetchall()
    return render_template('AdminHome.html',data=data)

@app.route("/AdminDonorInfo")
def AdminDonorInfo():

    conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
    cur = conn.cursor()
    cur.execute("SELECT * FROM personltb  ")
    data = cur.fetchall()



    return render_template('AdminDonorInfo.html', data=data)
```

```
@app.route("/UserHome")
def UserHome():
    user = session['uname']

    conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
    # cursor = conn.cursor()
    cur = conn.cursor()
    cur.execute("SELECT * FROM regtb where username='" + user + "'")
    data = cur.fetchall()
    return render_template('UserHome.html',data=data)



@app.route("/DonorHome")
def DonorHome():
    cuname = session['cname']

    conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
    # cursor = conn.cursor()
    cur = conn.cursor()
    cur.execute("SELECT * FROM companytb where username='" + cuname + "'")
    data = cur.fetchall()
    return render_template('DonorHome.html', data=data)

@app.route("/adminlogin", methods=['GET', 'POST'])
def adminlogin():
    error = None
    if request.method == 'POST':
       if request.form['uname'] == 'admin' or request.form['password'] == 'admin':

          conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
```

```
            # cursor = conn.cursor()
            cur = conn.cursor()
            cur.execute("SELECT * FROM regtb ")
            data = cur.fetchall()
            return render_template('AdminHome.html' , data=data)

        else:
         return render_template('index.html', error=error)



@app.route("/donorlogin", methods=['GET', 'POST'])
def donorlogin():
    error = None
    if request.method == 'POST':
        username = request.form['uname']
        password = request.form['password']
        session['dname'] = request.form['uname']

        conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
        cursor = conn.cursor()
        cursor.execute("SELECT * from donortb where username='" + username + "' and
Password='" + password + "'")
        data = cursor.fetchone()
        if data is None:
            alert = 'Username or Password is wrong'
            return render_template('goback.html', data=alert)
        else:
            print(data[0])
            session['uid'] = data[0]
            conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
            # cursor = conn.cursor()
            cur = conn.cursor()
```

```
        cur.execute("SELECT * FROM donortb where username='" + username + "' and
Password='" + password + "'")
        data = cur.fetchall()

        return render_template('DonorHome.html', data=data)

@app.route("/userlogin", methods=['GET', 'POST'])
def userlogin():

    if request.method == 'POST':
        username = request.form['uname']
        password = request.form['password']
        session['uname'] = request.form['uname']

        conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
        cursor = conn.cursor()
        cursor.execute("SELECT * from regtb where username='" + username + "' and
Password='" + password + "'")
        data = cursor.fetchone()
        if data is None:

            alert = 'Username or Password is wrong'
            return render_template('goback.html', data=alert)
        else:
            print(data[0])
            session['uid'] = data[0]
            conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
            # cursor = conn.cursor()
            cur = conn.cursor()
            cur.execute("SELECT * FROM regtb where username='" + username + "' and
Password='" + password + "'")
            data = cur.fetchall()
```

```
        return render_template('UserHome.html', data=data )

@app.route("/newuser", methods=['GET', 'POST'])
def newuser():
    if request.method == 'POST':

        name1 = request.form['name']
        gender1 = request.form['gender']
        Age = request.form['age']
        email = request.form['email']
        pnumber = request.form['phone']
        address = request.form['address']

        uname = request.form['uname']
        password = request.form['psw']



        conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
        cursor = conn.cursor()
        cursor.execute(
           "INSERT INTO regtb VALUES ('" + name1 + "','" + gender1 + "','" + Age + "','" +
email + "','" + pnumber + "','" + address + "','" + uname + "','" + password + "')")
        conn.commit()
        conn.close()
        # return 'file register successfully'

    return render_template('UserLogin.html')

@app.route("/personal", methods=['GET', 'POST'])
def personal():
    if request.method == 'POST':

        name1 = request.form['name']
```

```
        gender1 = request.form['gender']
        Age = request.form['age']
        email = request.form['email']
        pnumber = request.form['phone']
        address = request.form['address']

        blood = request.form['blood']
        health = request.form['health']
        dname = session['dname']



        conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
        cursor = conn.cursor()
        cursor.execute(
            "INSERT INTO personltb VALUES ('" + name1 + "','" + gender1 + "','" + Age +
"','" + email + "','" + pnumber + "','" + address + "','" + blood + "','" + health + "','"+
dname+"')")
        conn.commit()
        conn.close()

        alert = 'Record Saved'

        return render_template('goback.html', data=alert)


@app.route("/appr")
def appr():


    cid =  request.args.get('cid')
    dname = session['dname']
```

```
    conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
    cursor = conn.cursor()
    cursor.execute(
        "delete from  personltb where id='" + str(cid) + "' ")
    conn.commit()
    conn.close()


    conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
    cur = conn.cursor()
    cur.execute("SELECT * FROM personltb where Username='"+ dname +"' ")
    data = cur.fetchall()

    return render_template('DonorPersonalInfo.html', data=data)

@app.route("/DonorPersonalInfo")
def DonorPersonalInfo():

    dname = session['dname']

    conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
    cur = conn.cursor()
    cur.execute("SELECT * FROM personltb where Username='" + dname + "' ")
    data = cur.fetchall()

    return render_template('DonorPersonalInfo.html', data=data)



@app.route("/newdonor", methods=['GET', 'POST'])
def newdonor():
```

```
    if request.method == 'POST':

        name1 = request.form['name']

        phone = request.form['phone']

        email = request.form['email']

        uname = request.form['uname']
        password = request.form['psw']



        conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
        cursor = conn.cursor()
        cursor.execute(
            "INSERT INTO donortb VALUES ('" + name1 + "','" + phone + "','" + email + "','"
+ uname + "','" + password + "')")
        conn.commit()
        conn.close()

    return render_template('DonorLogin.html')

@app.route("/Search")
def Search():

    return render_template('Search.html')

@app.route("/dsearch", methods=['GET', 'POST'])
def dsearch():
    if request.form["submit"] == "Search":
        blood = request.form['blood']
```

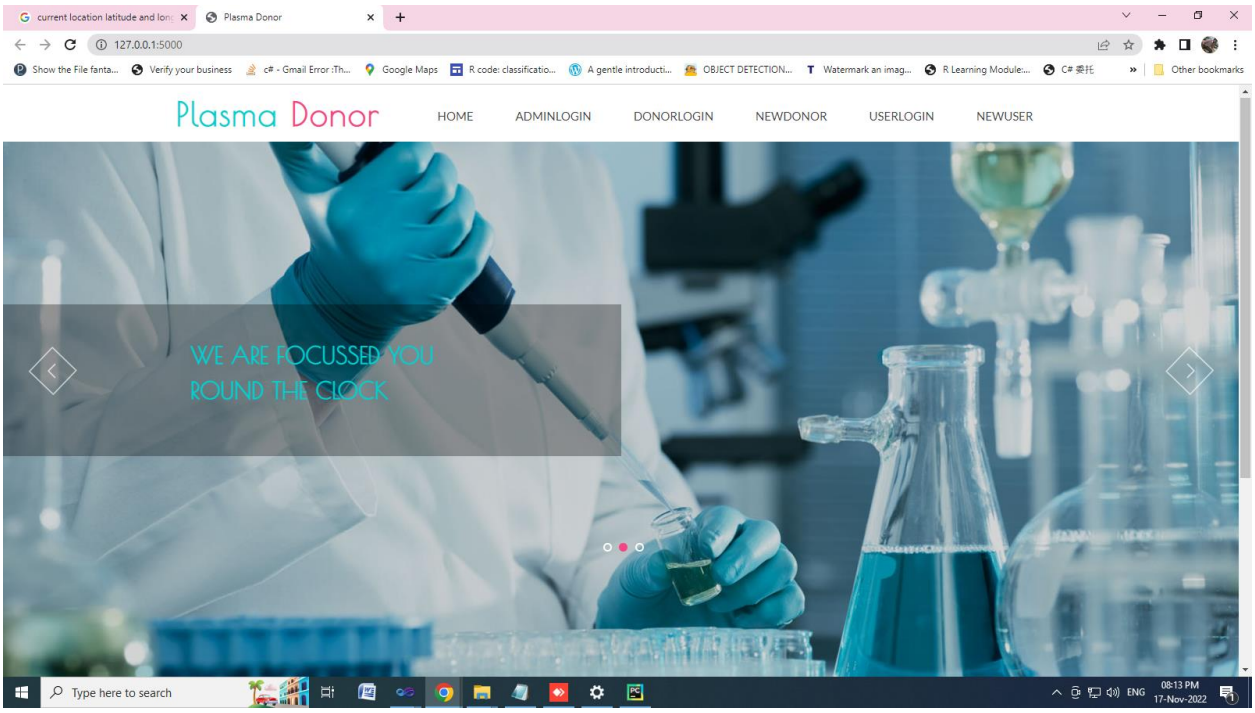
```
        conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
        cur = conn.cursor()
        cur.execute("SELECT * FROM personltb  where blood ='" + blood + "'")
        data = cur.fetchall()
        return render_template('Search.html', data=data)

    elif request.form["submit"] == "SendMail":
        blood = request.form['blood']
        info = request.form['info']

        conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
        cursor = conn.cursor()
        cursor.execute("SELECT  *  FROM  personltb where  Blood like '%" + blood + "%'")
        data = cursor.fetchall()
        for item in data:
            sendmsg(item[4], info)
            print(item[4])
        alert = 'Send Notication'

        return render_template('goback.html', data=alert)

@app.route("/SendRequest")
def SendRequest():

    session['cid'] =  request.args.get('cid')

    return render_template('Notification.html')


@app.route("/noti", methods=['GET', 'POST'])
def noti():
```

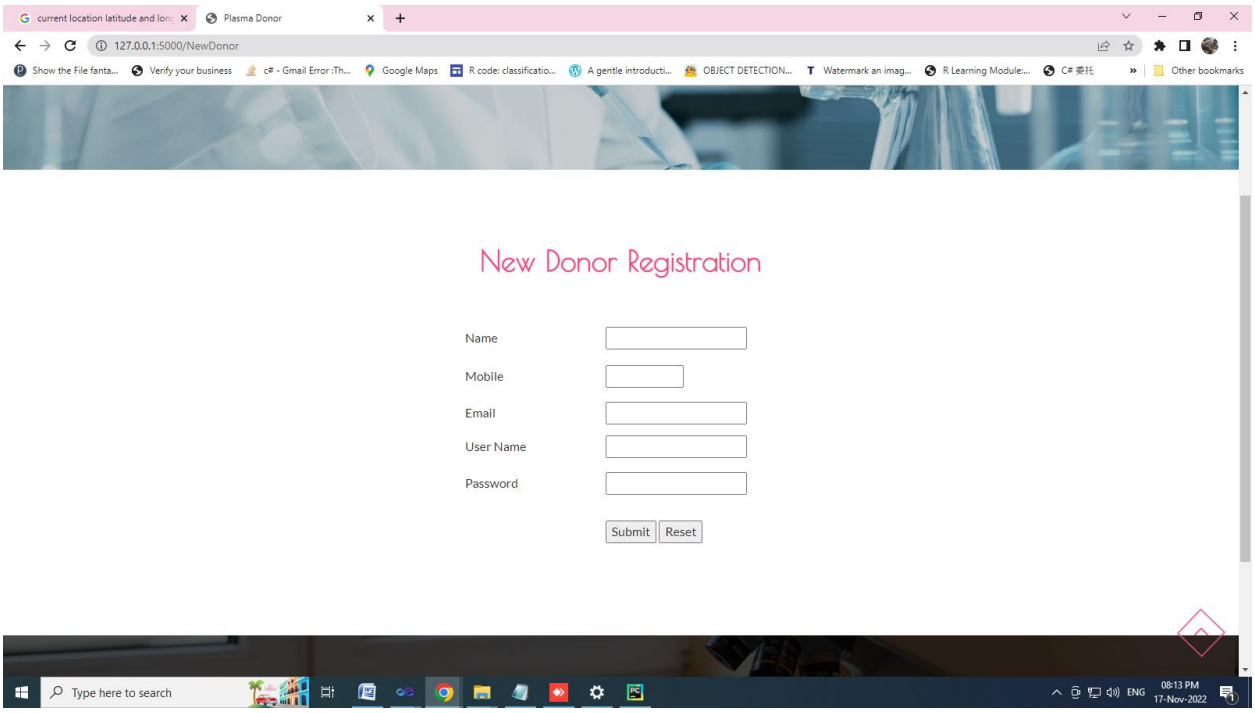
```
    info = request.form['info']

    did = session['cid']

    print(did)

    conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
    cursor = conn.cursor()
    cursor.execute("SELECT  *  FROM  personltb where  id='" + did + "'")
    data = cursor.fetchone()
    if data:
        bloo =data[7]
        print(bloo)
    else:
        return 'Incorrect username / password !'

    conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1Plasmadb')
    cursor = conn.cursor()
    cursor.execute("SELECT  *  FROM  personltb where  Blood like '%" + bloo + "%'")
    data = cursor.fetchall()
    for item in data:
        sendmsg(item[4], info)
        print(item[4])

    alert = 'Send Notication'

    return render_template('goback.html', data=alert)

def sendmsg(Mailid,message):
    import smtplib
    from email.mime.multipart import MIMEMultipart
```

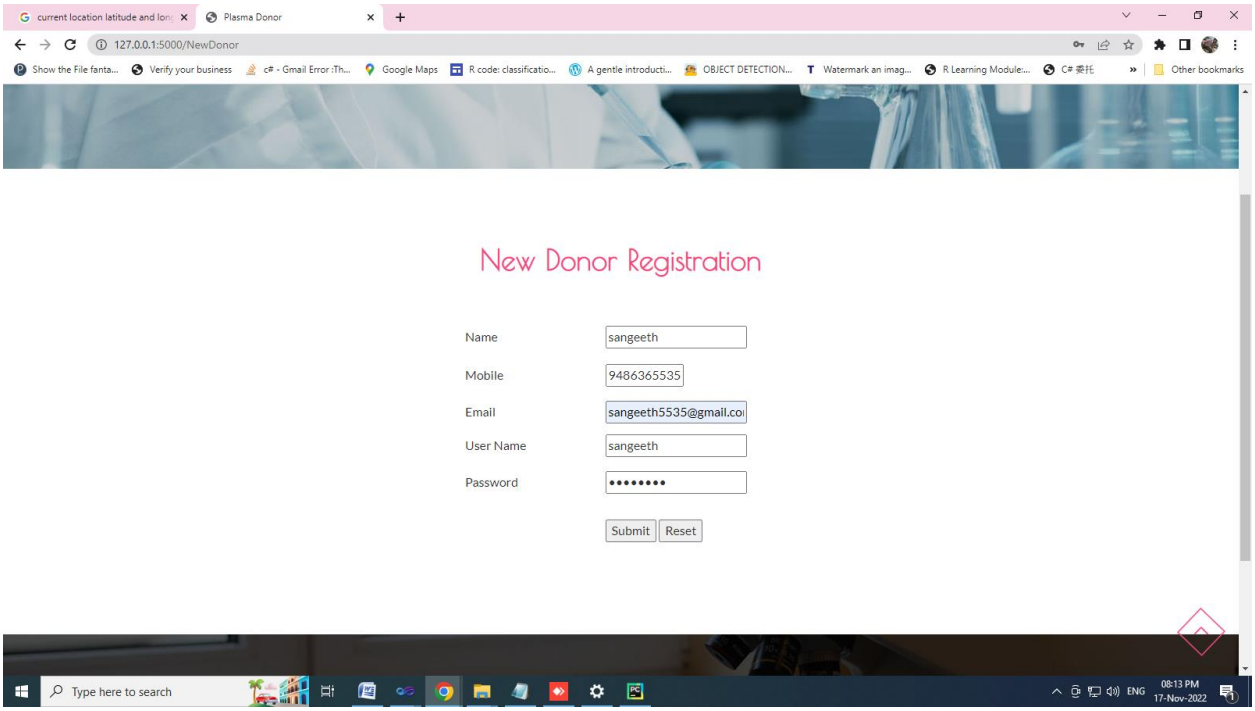
```
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders

fromaddr = "sampletest685@gmail.com"
toaddr = Mailid

# instance of MIMEMultipart
msg = MIMEMultipart()

# storing the senders email address
msg['From'] = fromaddr

# storing the receivers email address
msg['To'] = toaddr

# storing the subject
msg['Subject'] = "Alert"

# string to store the body of the mail
body = message

# attach the body with the msg instance
msg.attach(MIMEText(body, 'plain'))

# creates SMTP session
s = smtplib.SMTP('smtp.gmail.com', 587)

# start TLS for security
s.starttls()

# Authentication
s.login(fromaddr, "hneucvnontsuwgpj")
```

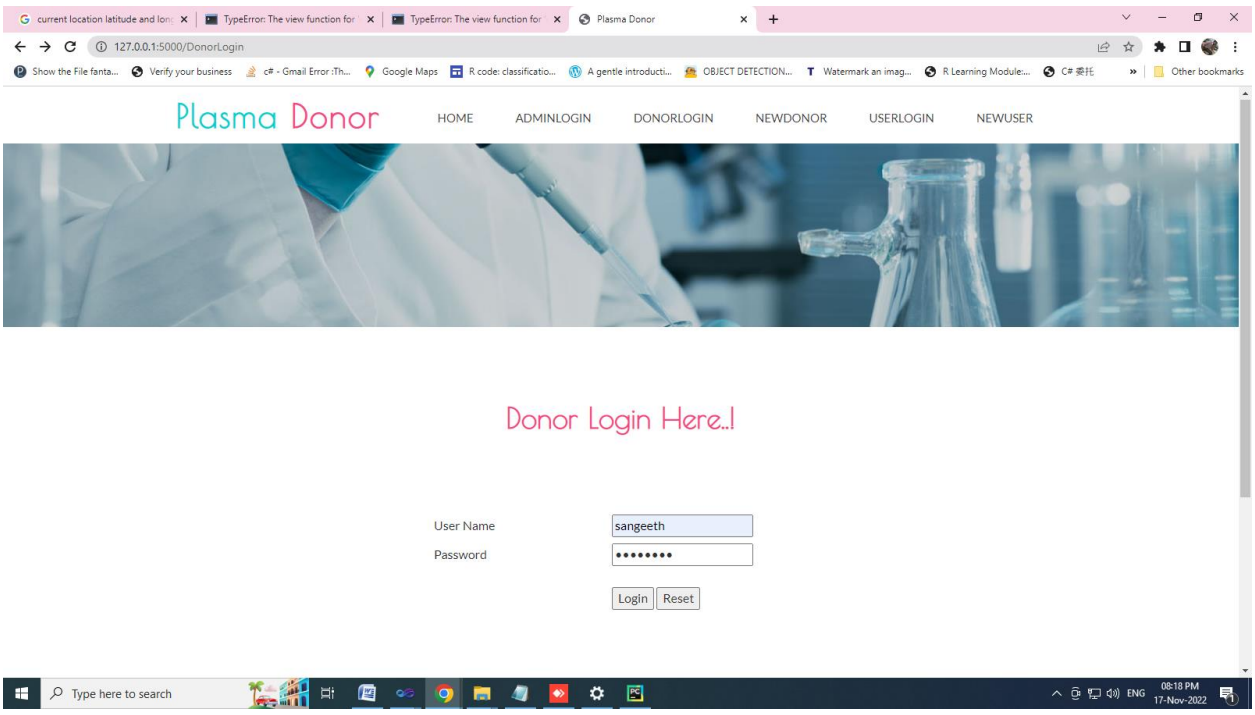
```
    # Converts the Multipart msg into a string
    text = msg.as_string()

    # sending the mail
    s.sendmail(fromaddr, toaddr, text)

    # terminating the session
    s.quit()


if __name__ == '__main__':
    app.run(debug=True, use_reloader=True)
```

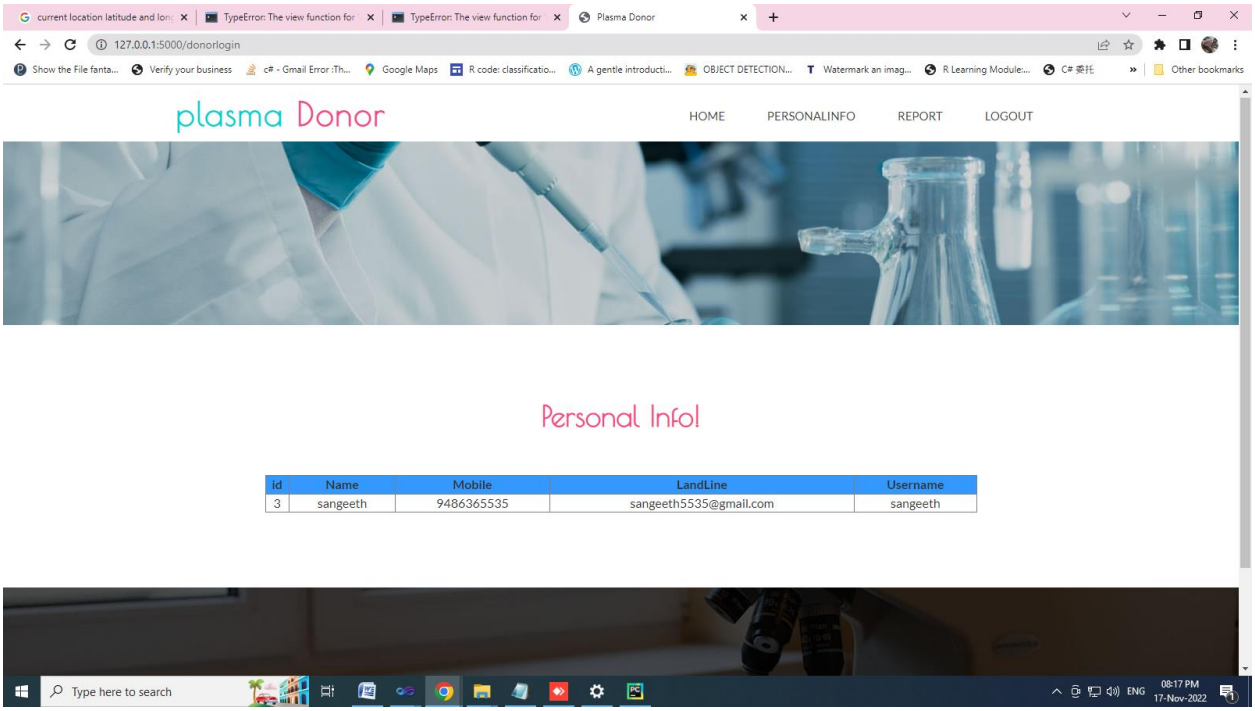
SCREENSHOTS

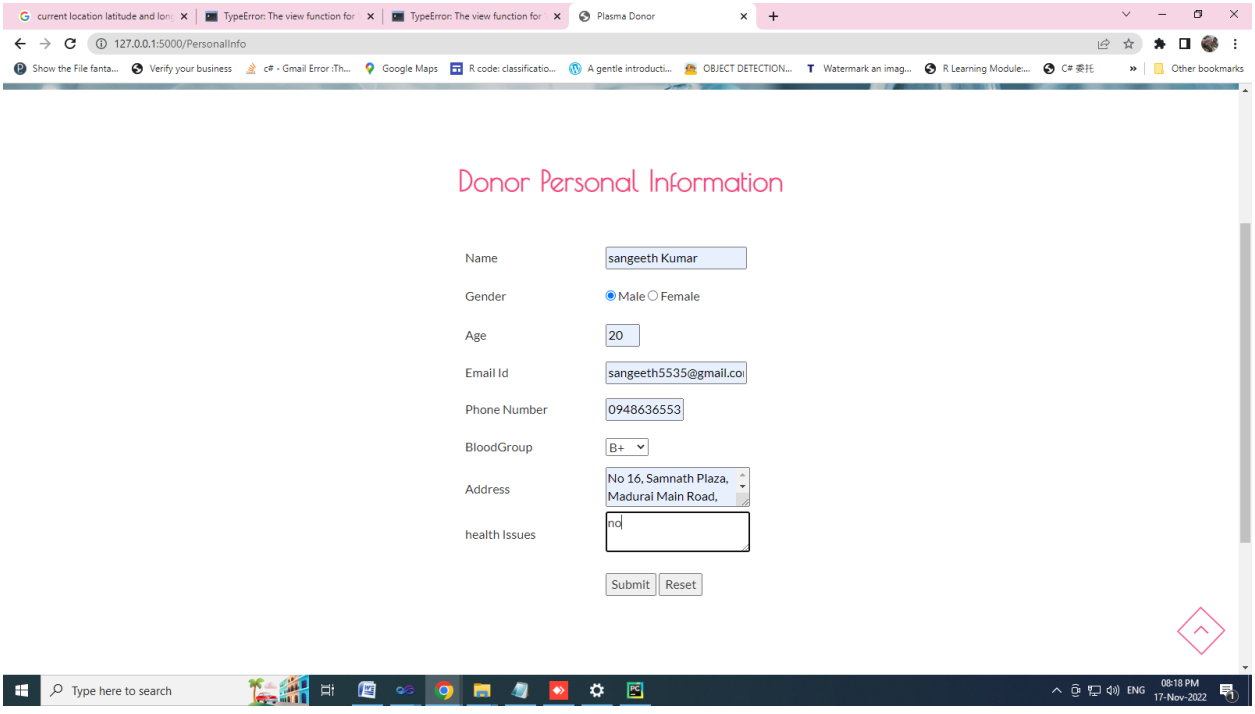


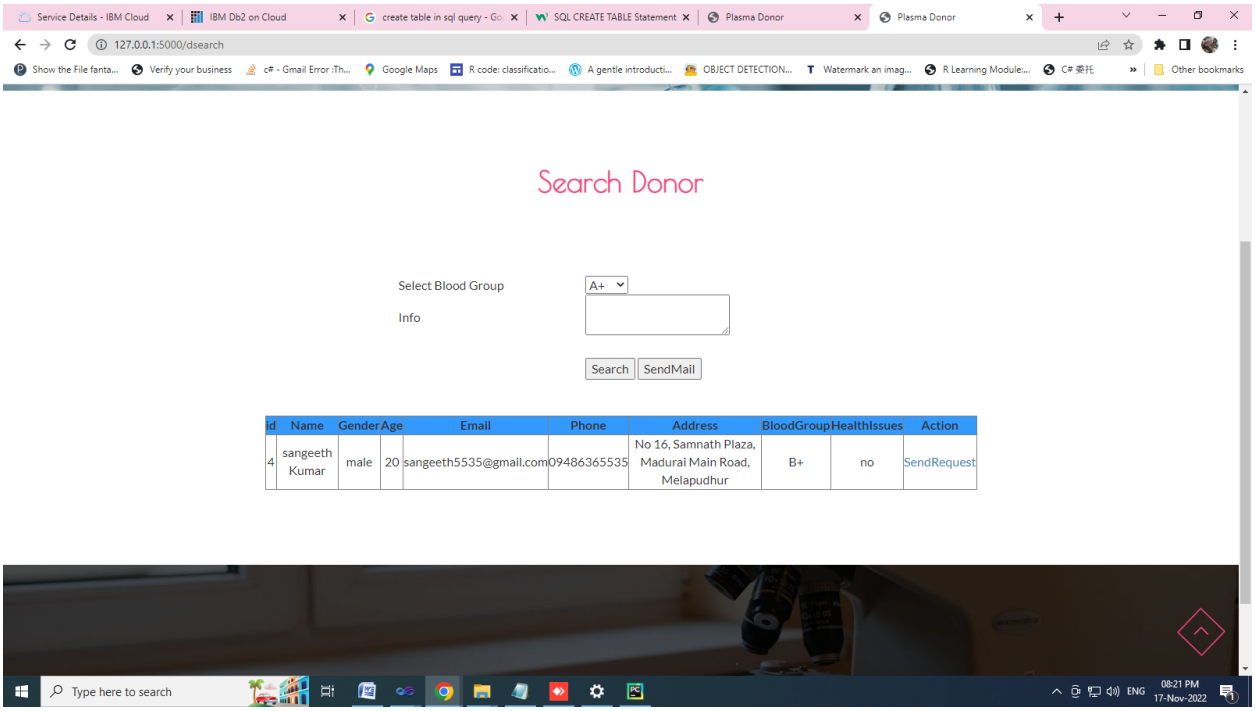


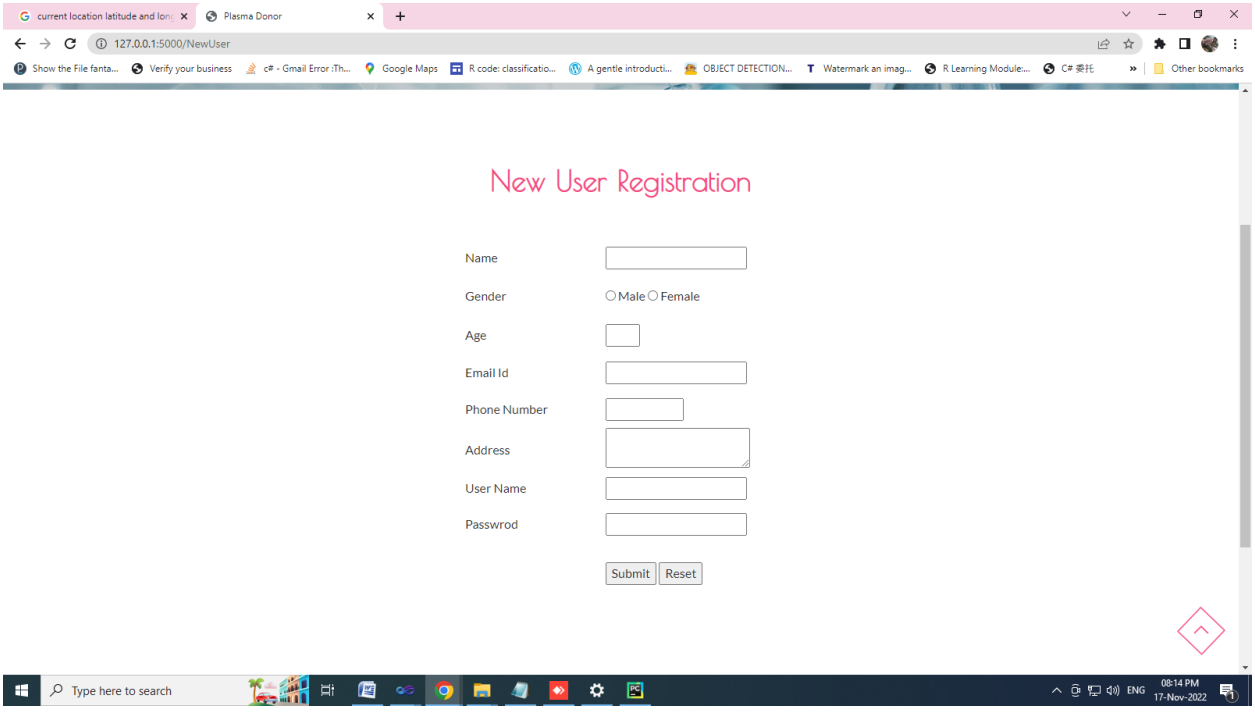

Plasma Donor
127.0.0.1:5000/NewDonor
New Donor Registration
Name
sangeeth
Mobile
9486365535
Email
sangeeth5535@gmail.co
User Name
sangeeth
Password
Submit
Reset
Type here to search
ENG
08:13 PM
17-Nov-2022

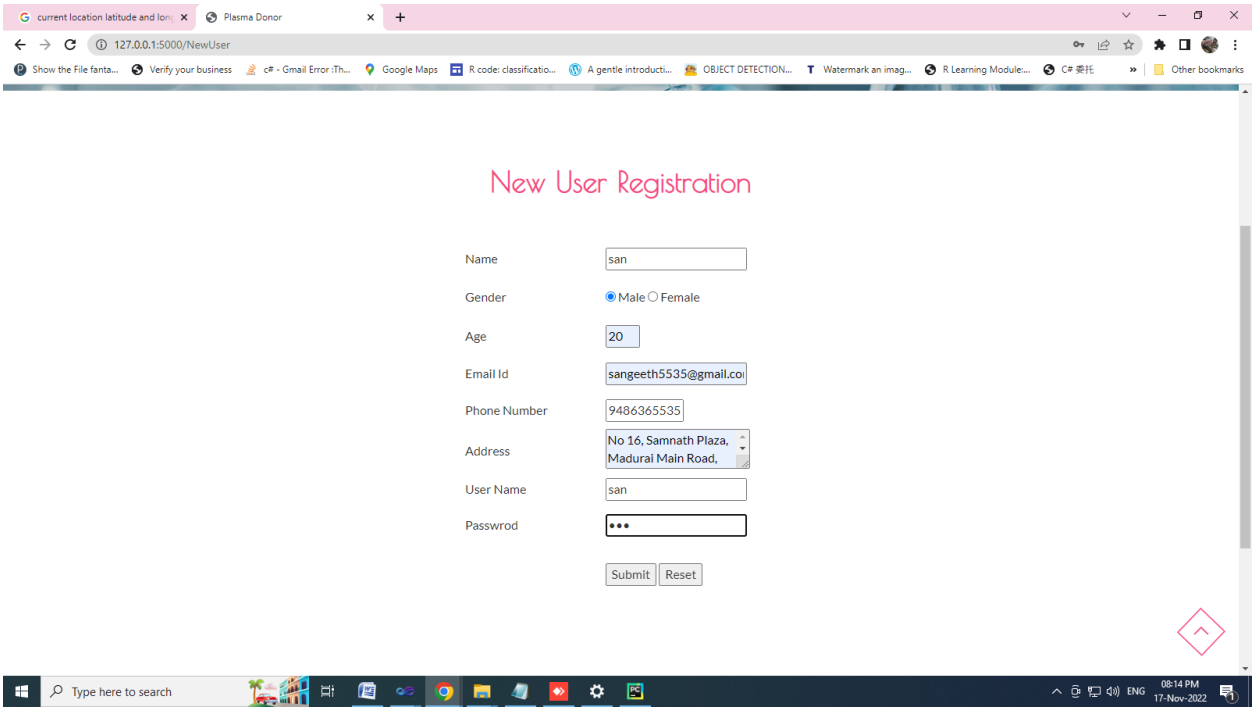

Plasma Donor
127.0.0.1:5000/DonorLogin
Plasma Donor
HOME
ADMINLOGIN
DONORLOGIN
NEWDONOR
USERLOGIN
NEWUSER
Donor Login Here..!
User Name
sangeeth
Password
Login
Reset
Type here to search
ENG
08:18 PM
17-Nov-2022

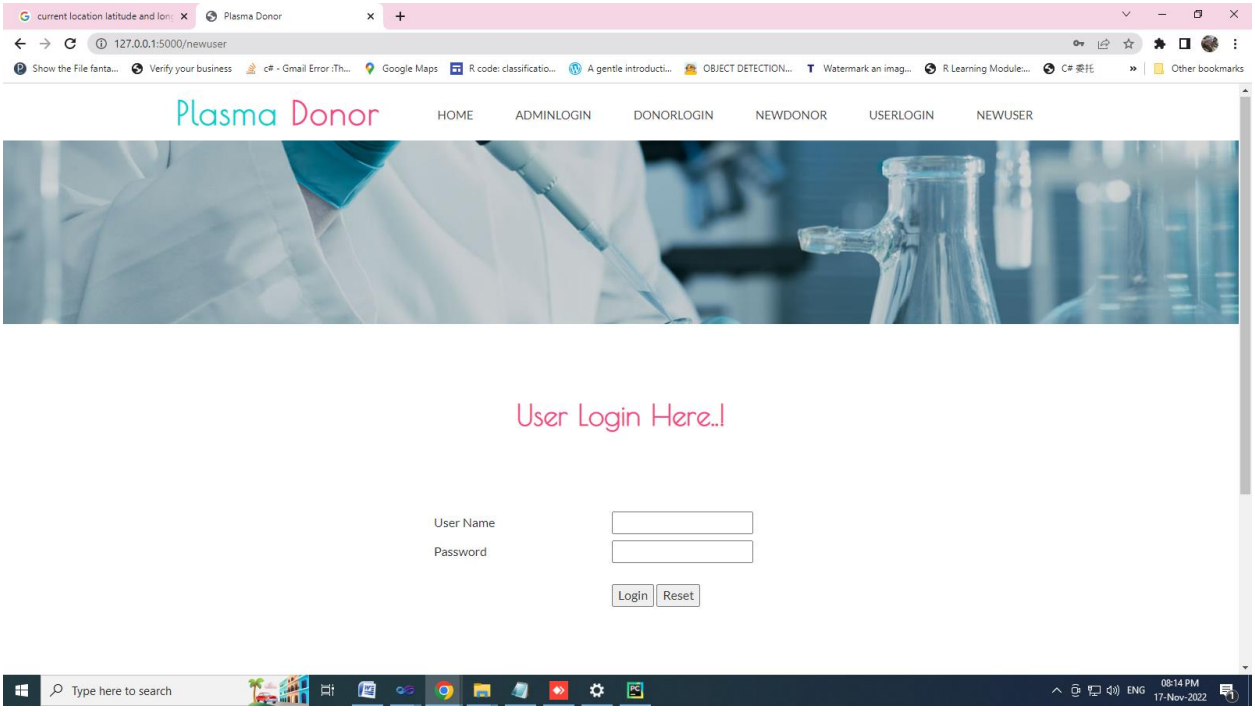

plasma Donor
HOME
PERSONALINFO
REPORT
LOGOUT
Personal Infol
id
Name
Mobile
LandLine
Username
3
sangeeth
9486365535
sangeeth5535@gmail.com
sangeeth

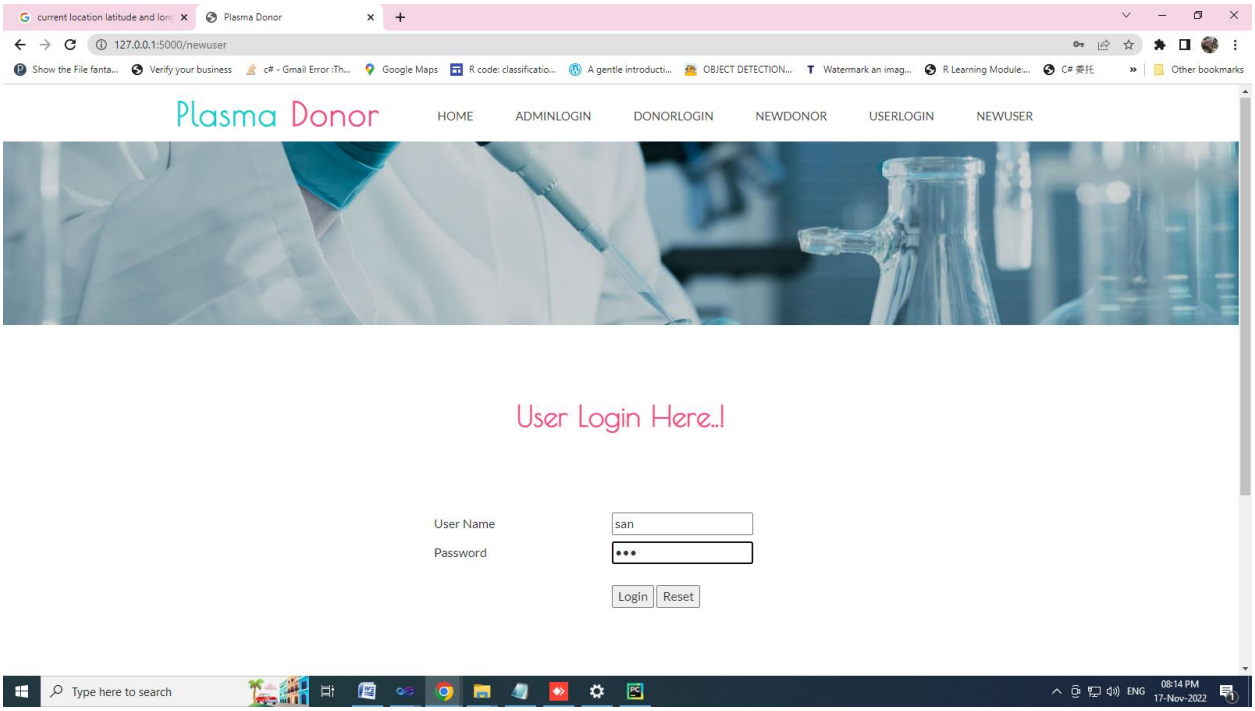

127.0.0.1:5000/PersonalInfo
Donor Personal Information
Name
sangeeth Kumar
Gender
Male
Female
Age
20
Email Id
sangeeth5535@gmail.co
Phone Number
0948636553
BloodGroup
B+
Address
No 16, Samnath Plaza,
Madurai Main Road,
health Issues
no
Submit
Reset

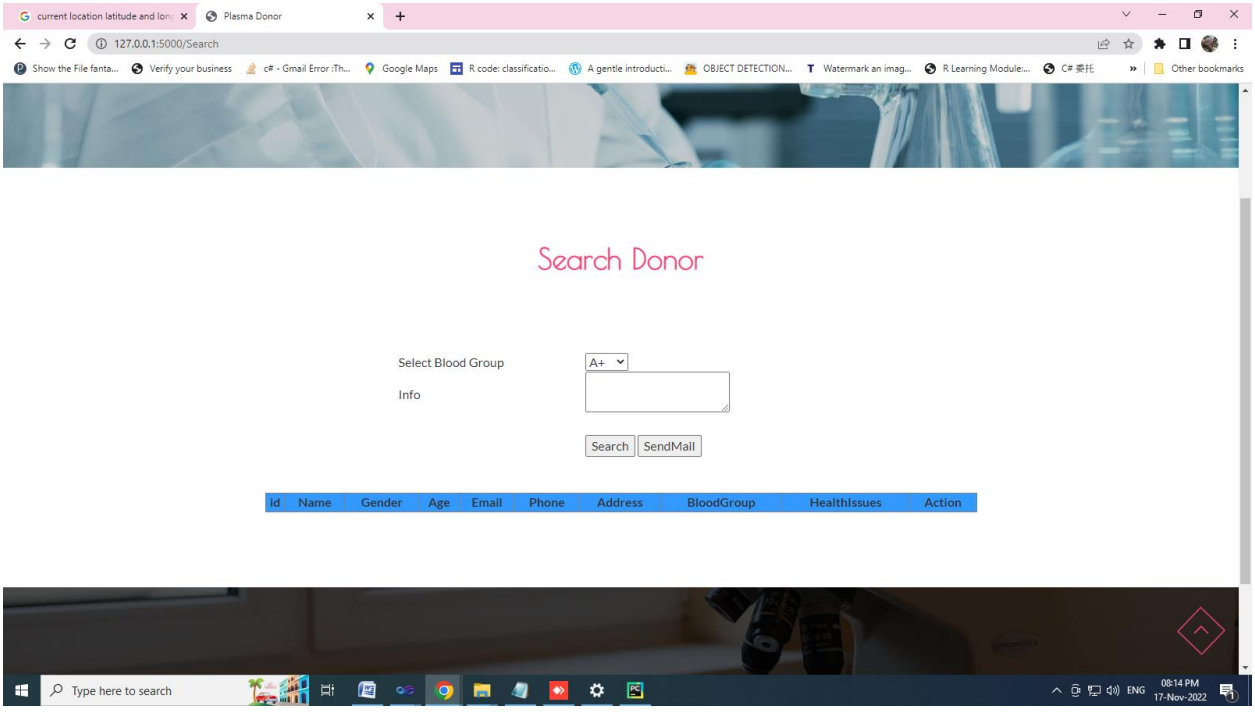

Search Donor
Select Blood Group
A+
Info
Search
SendMail
Id Name Gender Age Email Phone Address BloodGroup HealthIssues Action
4 sangeeth Kumar male 20 sangeeth5535@gmail.com 09486365535 No 16, Samnath Plaza, Madurai Main Road, Melapudhur B+ no SendRequest
Type here to search
ENG
08:21 PM
17-Nov-2022

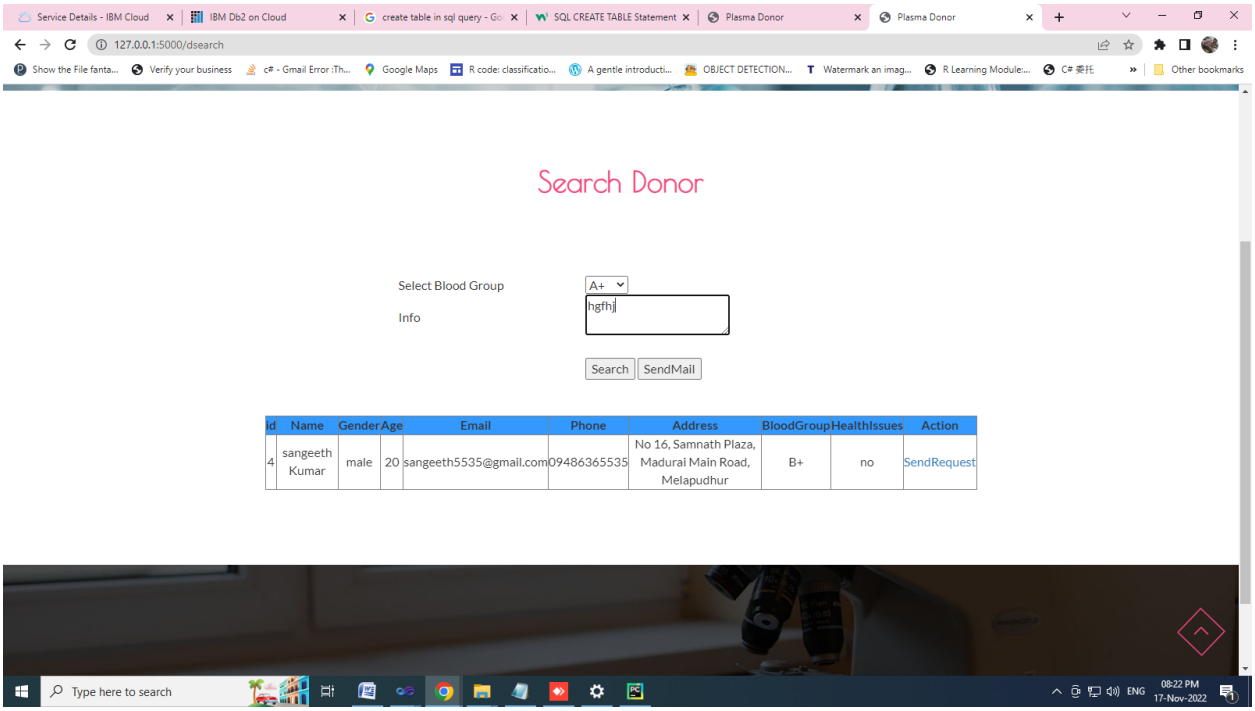

New User Registration
Name
Gender
Male Female
Age
Email Id
Phone Number
Address
User Name
Passwrod
Submit
Reset
Type here to search
ENG
08:14 PM
17-Nov-2022

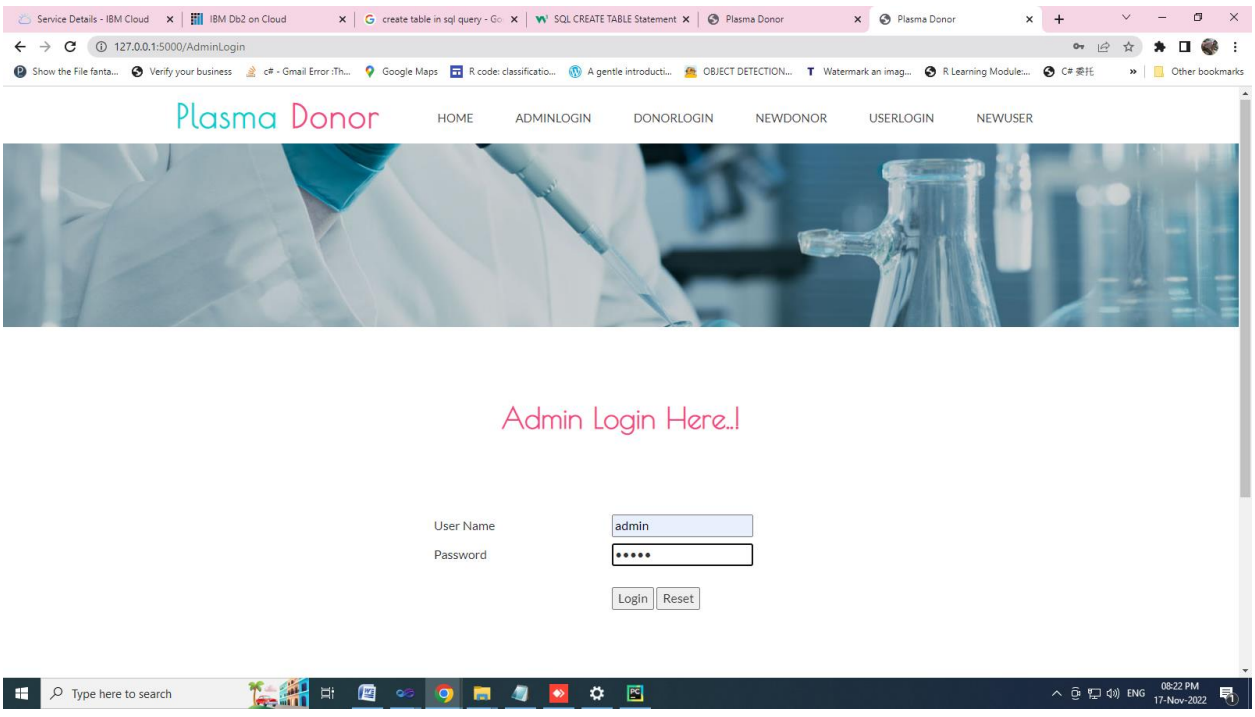

current location latitude and lon
Plasma Donor
127.0.0.1:5000/NewUser
New User Registration
Name
san
Gender
Male
Female
Age
20
Email Id
sangeeth5535@gmail.co
Phone Number
9486365535
Address
No 16, Samnath Plaza,
Madurai Main Road,
User Name
san
Passwrod
Submit
Reset
Type here to search
ENG
08:14 PM
17-Nov-2022

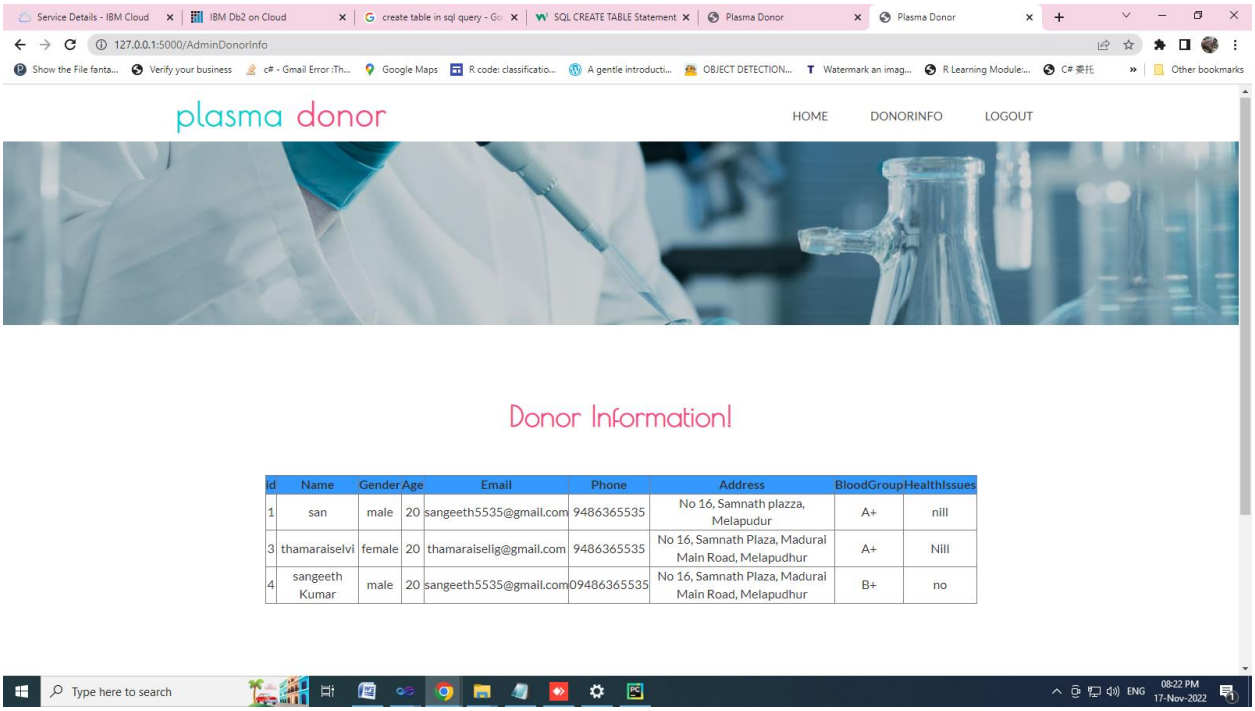

current location latitude and lon
Plasma Donor
127.0.0.1:5000/newuser
Plasma Donor
HOME
ADMINLOGIN
DONORLOGIN
NEWDONOR
USERLOGIN
NEWUSER
User Login Here..!
User Name
Password
Login
Reset
Type here to search
ENG
08:14 PM
17-Nov-2022


Plasma Donor
HOME
ADMINLOGIN
DONORLOGIN
NEWDONOR
USERLOGIN
NEWUSER
User Login Here..!
User Name
san
Password
Login
Reset


Search Donor
Select Blood Group
A+
Info
Search
SendMail
id
Name
Gender
Age
Email
Phone
Address
BloodGroup
HealthIssues
Action


Search Donor
Select Blood Group
A+
Info
hgfhj
Search
SendMail
Id Name Gender Age Email Phone Address BloodGroup HealthIssues Action
4 sangeeth Kumar male 20 sangeeth5535@gmail.com 09486365535 No 16, Samnath Plaza, Madurai Main Road, Melapudhur B+ no SendRequest


Plasma Donor
HOME
ADMINLOGIN
DONORLOGIN
NEWDONOR
USERLOGIN
NEWUSER
Admin Login Here..!
User Name
admin
Password
Login
Reset



Id	Name	Gender	Age	Email	Phone	Address	BloodGroup	HealthIssues
1	san	male	20	sangeeth5535@gmail.com	9486365535	No 16, Samnath plazza, Melapudur	A+	nill
3	thamaraiselvi	female	20	thamaraiselig@gmail.com	9486365535	No 16, Samnath Plaza, Madurai Main Road, Melapudhur	A+	Nill
4	sangeeth Kumar	male	20	sangeeth5535@gmail.com	09486365535	No 16, Samnath Plaza, Madurai Main Road, Melapudhur	B+	no

GITHUB & PROJECT DEMO LINK

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-15117-1659594302

PROJECT DEMO LINK:

https://youtu.be/CDzeYb-KpqU