

# **PLASMA DONOR APPLICATION**



**Team ID:PNT2022TMID48261**

**Faculty mentor:**

**M.Naseem banu**

**Team leadA.ANUSUDHA**

**Team member:K. Jeeva**

**Team member:M.Sopitha**

**Team member:K.Kaviya**

## **1. INTRODUCTION**

1.1Project Overview

1.2Purpose

## **2. LITERATURE SURVEY**

2.1Existing problem

2.2References

2.3Problem Statement Definition

## **3. IDEATION & PROPOSED SOLUTION**

3.1Empathy Map Canvas

3.2Ideation & Brainstorming

3.3Proposed Solution

3.4Problem Solution fit

## **4. REQUIREMENT ANALYSIS**

4.1Functional requirement

4.2Non-Functional requirements

## **5. PROJECT DESIGN**

5.1Data Flow Diagrams

5.2Solution & Technical Architecture

5.3User Stories

## **6. PROJECT PLANNING & SCHEDULING**

6.1Sprint Planning & Estimation

6.2Sprint Delivery Schedule

6.3Reports from JIRA

## **7. CODING & SOLUTIONING (Explain the features added in the project along with code)**

7.1Feature 1

7.2Feature 2

## 8. TESTING

8.1Test Cases

8.2User Acceptance Testing

## 9. RESULTS

9.1Performance Metrics

10. **ADVANTAGES & DISADVANTAGES**

11. **CONCLUSION**

12. **FUTURE SCOPE**

13. **APPENDIX**

Source Code

GitHub & Project demo link.

# **1.INTRODUCTION**

## **1.1 Project overview**

The main objective of this project is to provide the recipient with a donor who is in good form with no health ailments to donate blood of the corresponding blood group. This project provides quick access to donors for an immediate requirement of blood. In case of an emergency/surgery, blood procurement is always a major problem which consumes a lot of time. This helps serve the major time-lapse in which a life can be saved!

## **1.2 Purpose**

Blood plasma donations are used for slightly more specific purposes than a general blood donation. The most common uses of plasma donations include individuals who have experienced a severe trauma, burn or shock, adults or children with cancer, and people with liver or clotting factor disorders.

## **2.Litreature survey:**

### **2.1. Existing solution:**

- Cannot Upload and Download the latest updates.
- No use of Web Services and Remoteing.
- Risk of mismanagement and of data when the project is under development.
- Security.
- proper coordination between different Applications and Users.
- Fewer Users – Friendly.

### **2.2. References:**

- [1] X. Chen, "Commercial plasma donation and individual health in impoverished rural china," Health economics review, vol. 4,no. 1, p. 30, 2014.
- [2] "Blood safety and availability," <https://www.who.int/news-room/fact-sheets/detail/blood-safety-and-availability>,(Accessed on 09/05/2020).
- [3] T. Alana and B. Alsaed, "Use of social media in the blood donation process in saudi arabia," Journal of Blood Medicine,vol. 10, p. 417, 2019.
- [4] T. Changchun, K. Wang mo, U. Changchun, P. Gem, P. R.Dhungyel et al., "Need of medium for Finding blood donor in Bhutan," Asian Journal For Convergence In Technology(AJCT), 2018.
- [5] V. K. Datamation and H. El-Okla, "Blood r: blood donor and requester mobile application," M health, vol. 3, 2017.
- [6] M. S. Hosanna, N. Das, M. K. H. Patwary, and M. Al-Hasan, "Finding the nearest blood donors using Dijkstra al-algorithm," SISFORMA: Journal of Information Systems (e-Journal), vol. 5, no. 2, pp. 40–44, 2019.
- [7] H. D. D as, R. Ahmed, N. Smrity, and L. Islam, "Bdonor:A geo-localized blood donor management system using mobile crowdsourcing," in 2020 IEEE 9th International Conference onCommunication Systems and Network Technologies (CSNT).IEEE, 2020, pp. 313–317.
- [8] S.-Q. Wang and D.-M. Zhu, "Research on selecting initial points for k-means clustering," in 2008 International Conference on Machine Learning and Cybernetics, vol. 5. IEEE, 2008, pp.2673–2677.

- [9] S. Na, L. Xumin, and G. Yong, "Research on k-means clustering algorithm: An improved k-means clustering algorithm," in 2010 Third International Symposium on intelligent information technology and security informatics. IEEE, 2010, pp. 63–67.
- [10] K. Sapphire and P. Baby, "Agglomerate hierarchical clustering algorithm-a," International Journal of Scientific and Research Publications, vol. 83, p. 83, 2013.
- [11] D. Müllner, "Modern hierarchical, agglomeration clustering algorithms," arXiv pre print arXiv:1109.2378, 2011.
- [12] "Open cage geocoder." [Online]. Available: <https://opencagedata.com/>.

### **2.3.Problem statement definition:**

We are designing a system which contains all the information regarding blood donation camps ongoing in a particular area so that people who want to donate blood will get information regarding these camps.

### 3.IDEATION AND PROPOSED SOLUTION:

#### 3.1. Empathy map

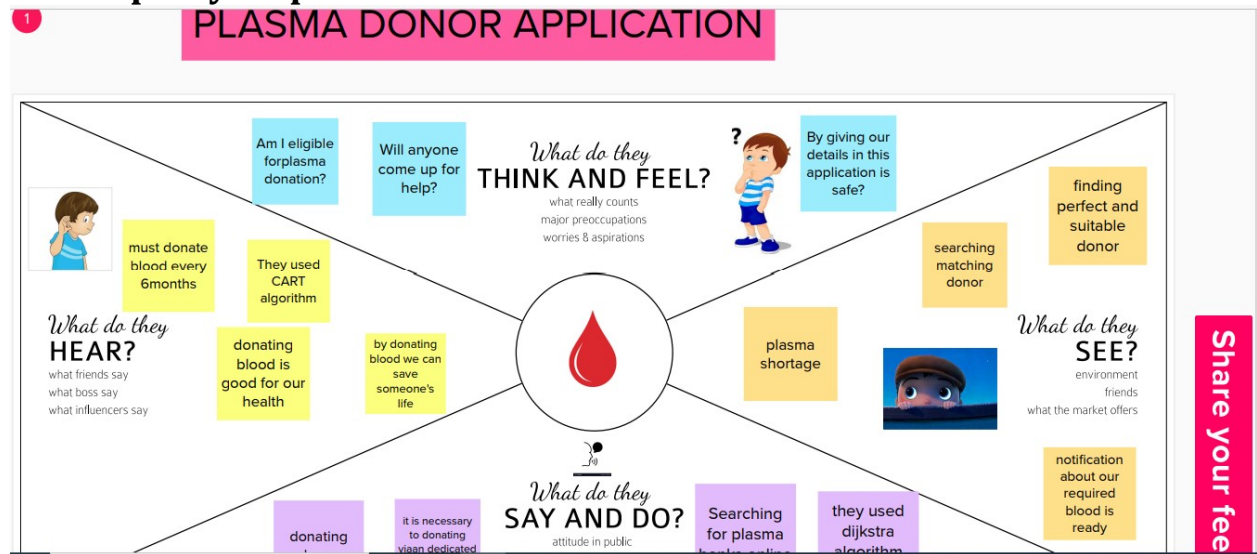


Figure 3.1

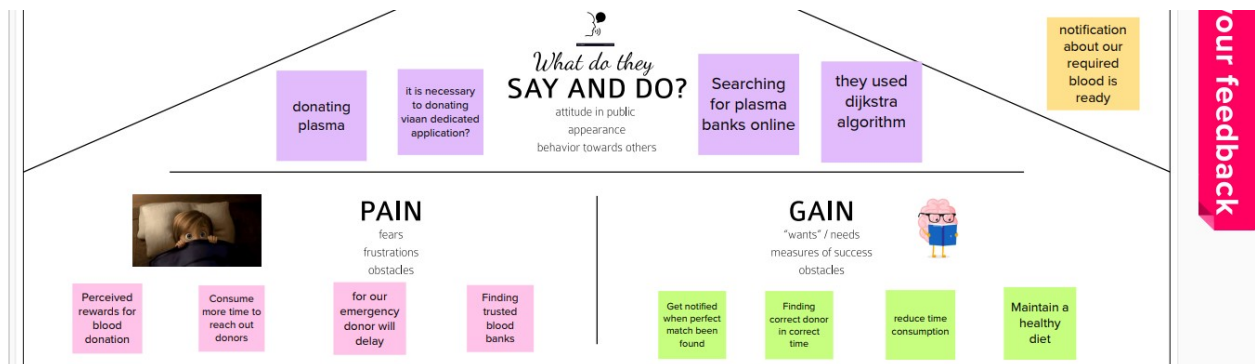


Figure 3.2

#### 3.2.Ideation & Brainstorming

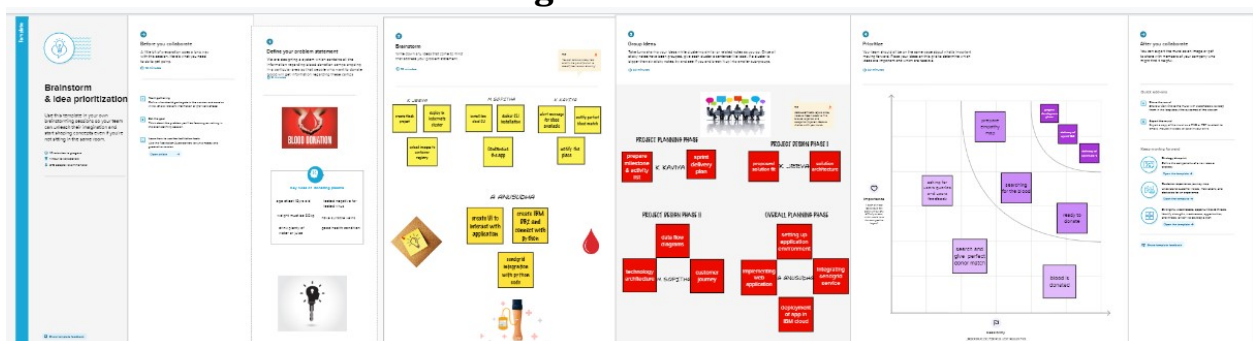


figure 3.2.1

### 3.3 Proposed solution

#### PROBLEM SOLUTION PROBLEM SOLUTION

**Category:** Cloud App Development

**Skills Required:**

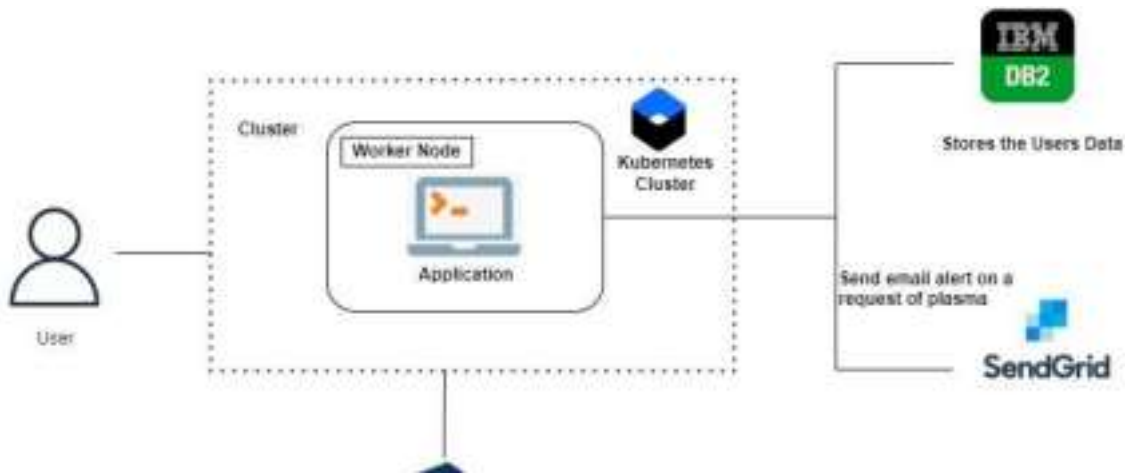
Storage,Python- IBM Cloud,HTML,Javascript,IBM Cloud Object  
Flask,Kubernetes,Docker,IBM DB2,IBM Container Registry

**Description:**

During the COVID 19 crisis, the requirement of plasma became a high priority and the donor count has become low. Saving the by notifying the donor information and helping the needy by notifying the current donors list, would be a helping hand. In regard to 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 the donor details, store them .

**Technical Architecture:**





### Project Workflow:

1. The user interacts with the application.
2. Registers by giving the details as a donor.
3. The database will have all the details and if a user posts a request then the concerned blood group donors will get notified about it.

## 3.4 PROPOSED SOLUTION FIT

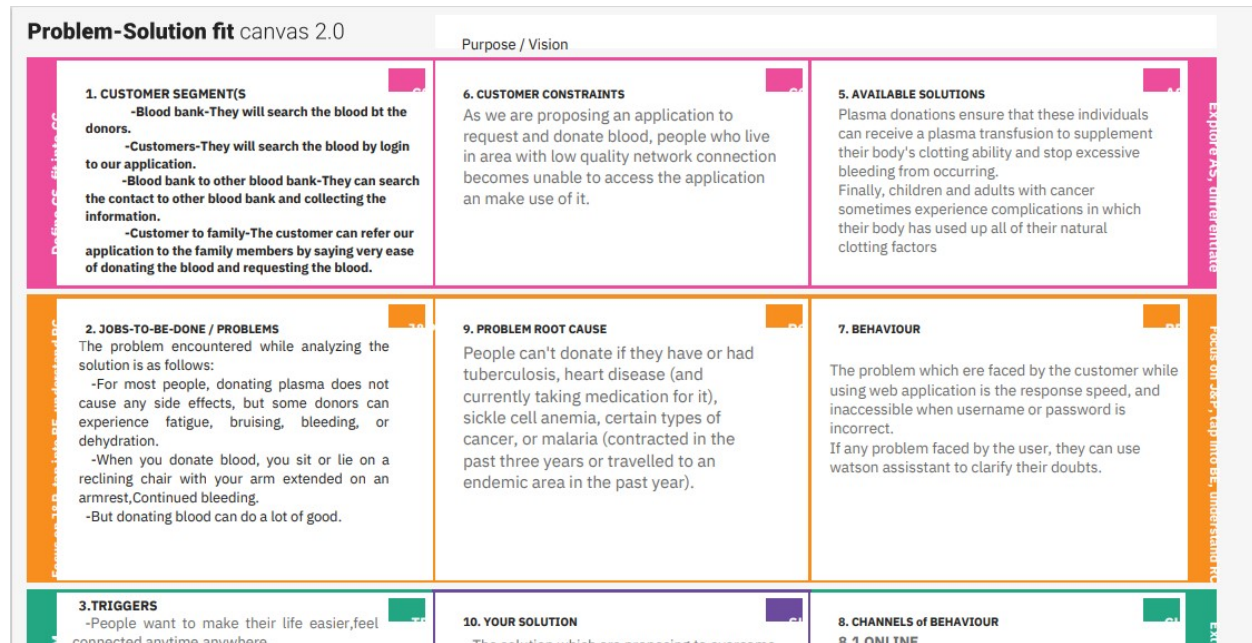


Figure3.4.1



Figure3.4.2

## 4. REQUIREMENT ANALYSIS

### 4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Html, javascript, flask	Through web application
FR-4	Login portal	Through the application
FR-5	Donor history	Through the database
FR-6	Donors list based on their locality	Through the database

Figure4.1.1

### 4.2 Non Functional requirement

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	User can donate and request blood.
NFR-2	Security	User details can be safe
NFR-3	Reliability	Can access from anywhere at any time.
NFR-4	Performance	Can request and donate blood at anytime through this application
NFR-5	Availability	In blood bank all blood groups are available
NFR-6	Scalability	More facilities can be added like more than ten database.

Figure4.2.1

## 5.PROJECT DESIGN

### 5.1Data Flow Diagrams

The content of plasma donor DFD level 1 must be single process node from the context diagram and is broken into sub process.

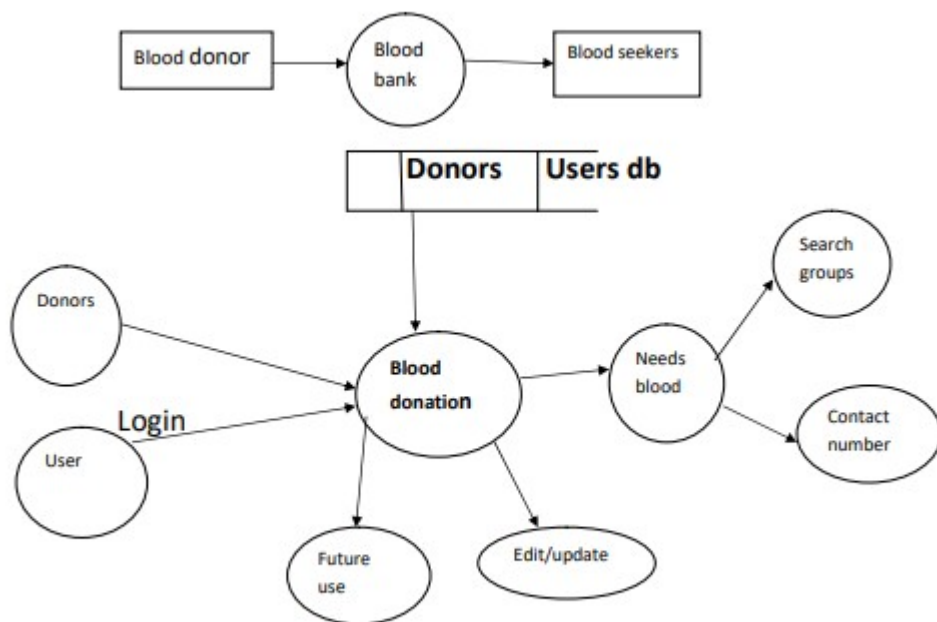


Figure5.1.1

DFD Level 0:

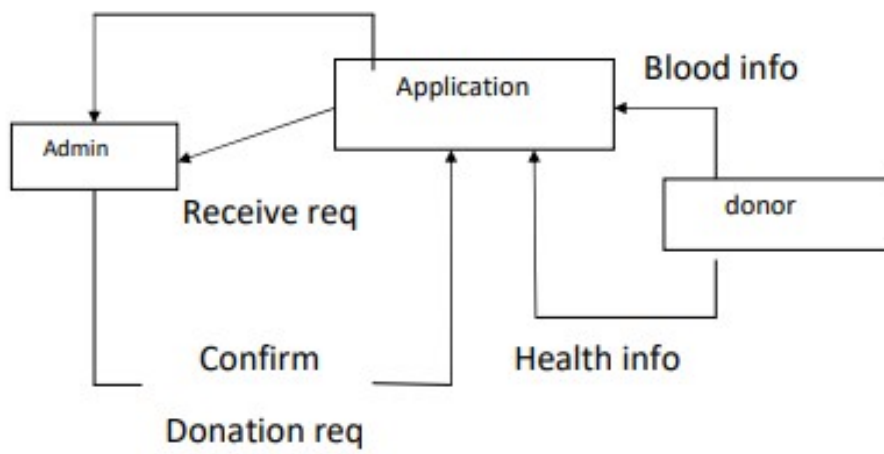


Figure5.1.2

DFD Level 1:

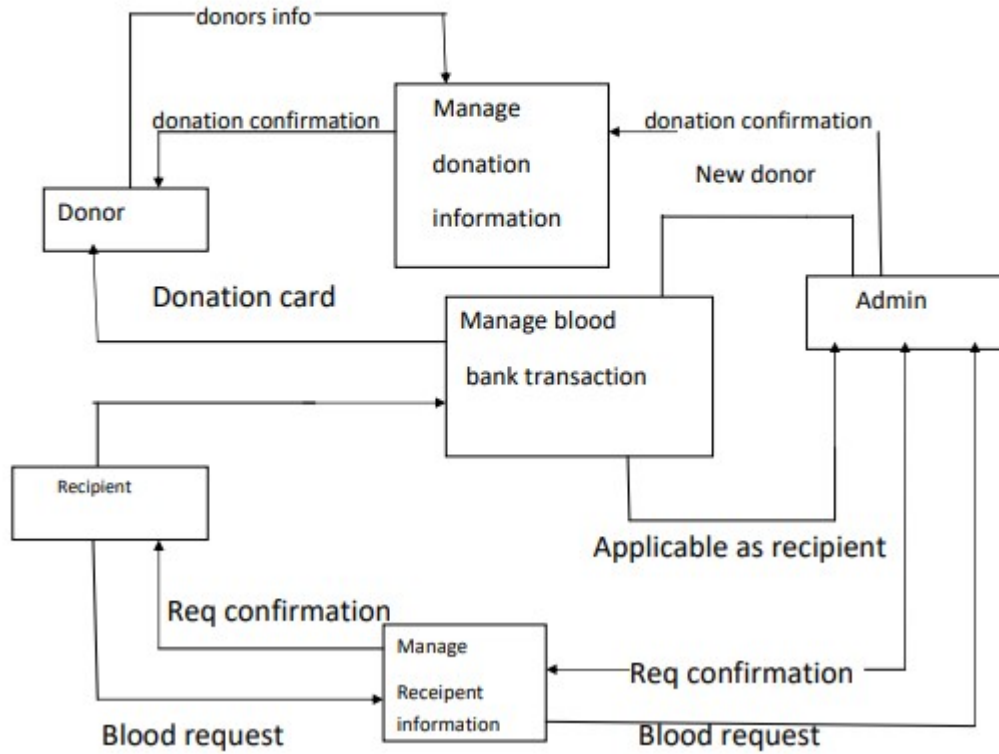


Figure5.1.3

DFD Level 2:

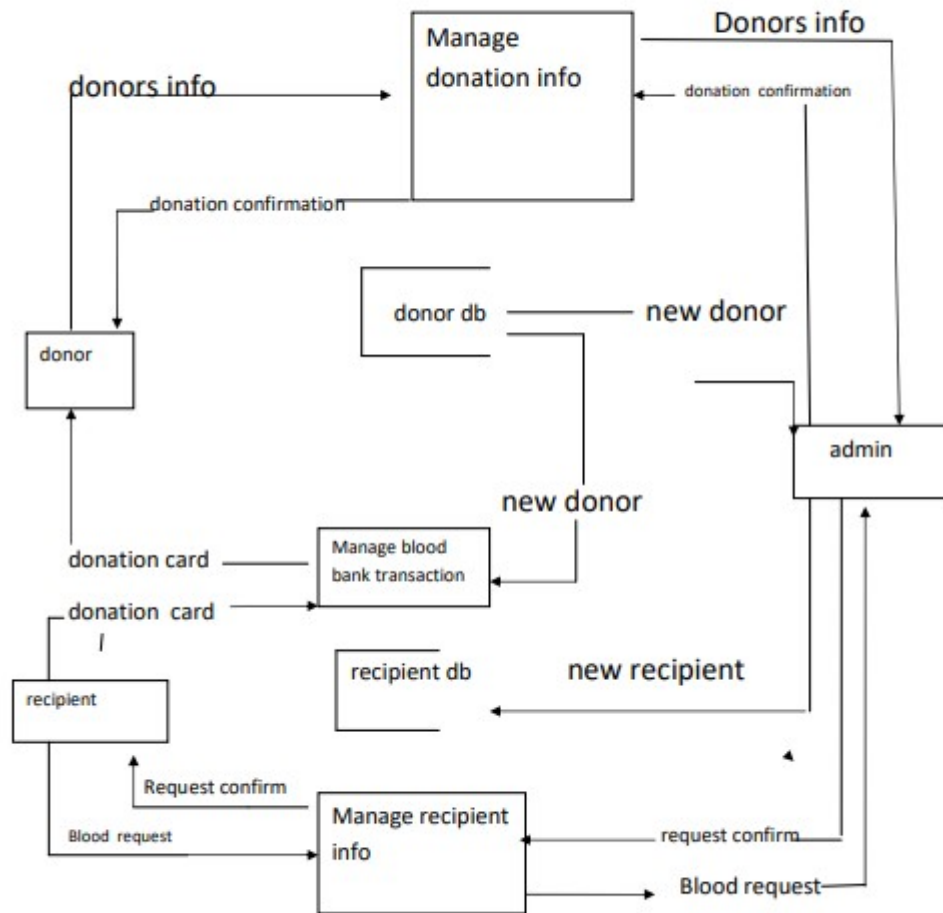


Figure5.1.4

## 5.2 Solution & Technical Architecture

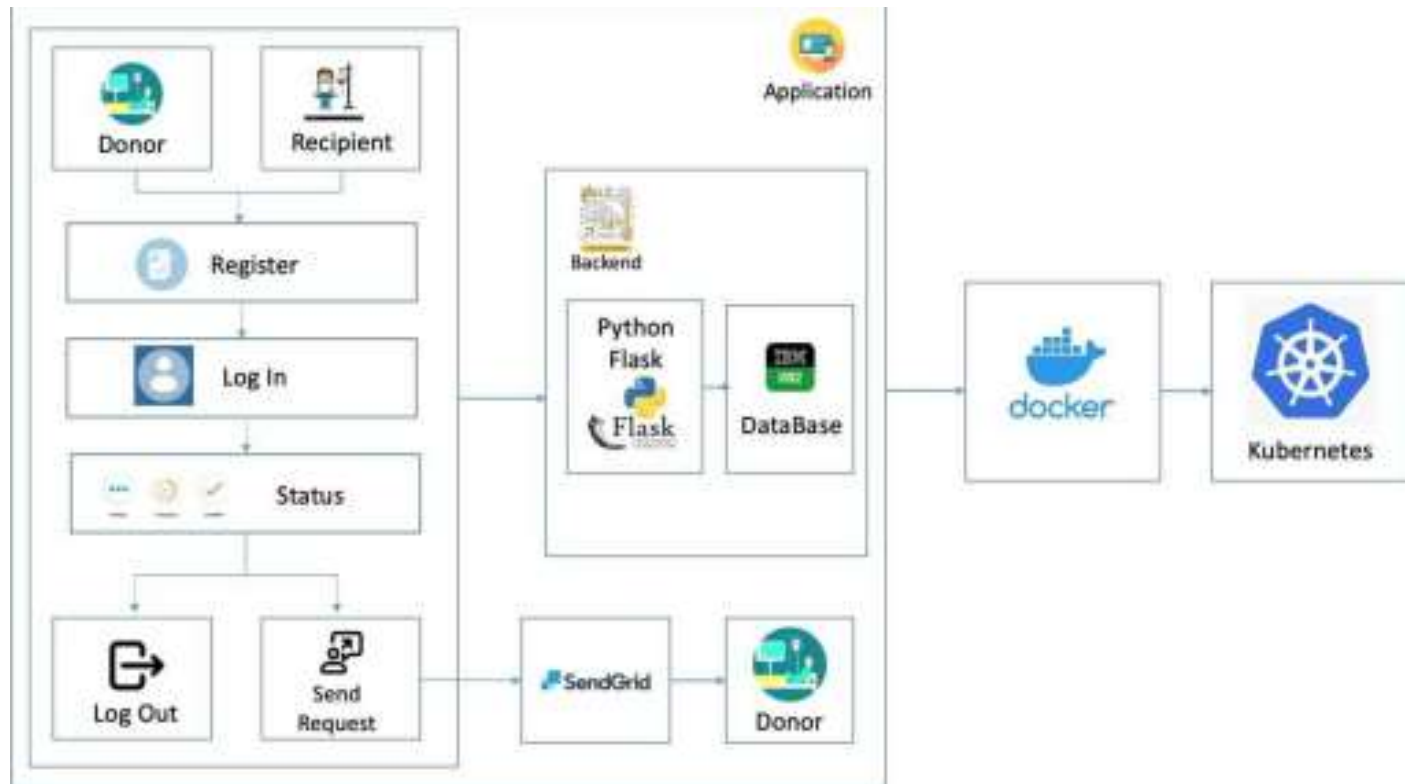
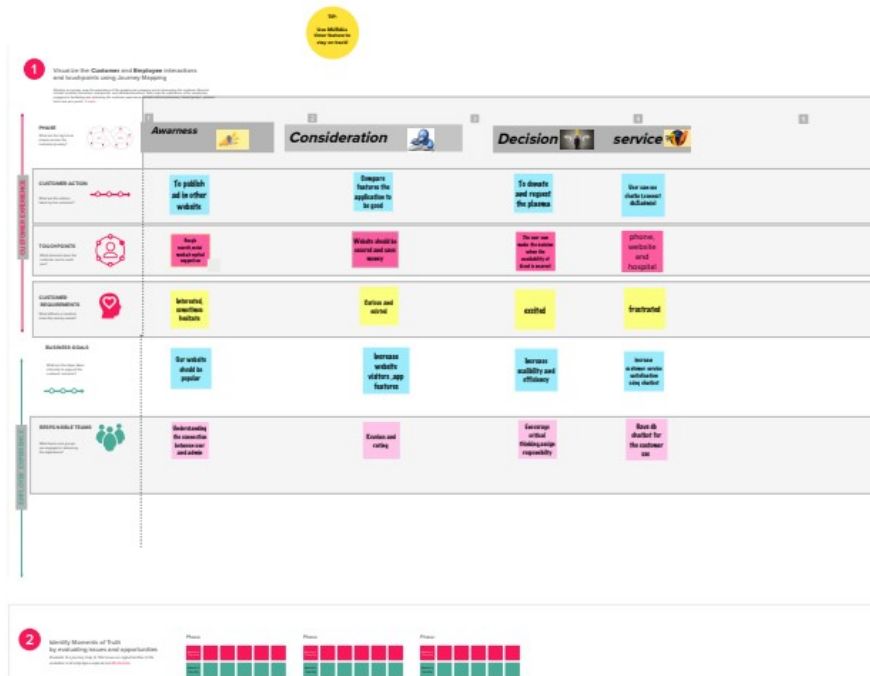


Figure5.2.1



## 5.3 User Stories



## 6 Sprint Planning & Estimation

### 6.1 PROJECT PLANNING & SCHEDULING

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering the details.	8	High	A.Anusudha K.Jeeva
Sprint-1	Login	USN-2	As a user, I can login for the application by entering my email, password, and confirming my password.	7	High	K.Kaviya M.Sopitha
Sprint-1	Email	USN-3	As a user, I can receive mail for registered for the application.	5	Low	A.Anusudha K.Jeeva
Sprint-2	Notification	USN-4	As a user I can receive the notification when the donor is ready to donate plasma.	5	Medium	K.Kaviya M.Sopitha
Sprint-2	Dashboard	USN-5	As a user and admin, can view, request plasma and donate plasma.	15	High	A.Anusudha K.Jeeva
Sprint-3	Donor details	USN-6	As a donor, I can enter the details to donate the plasma.	10	High	K.Kaviya M.Sopitha
Sprint-3	Request plasma	USN-7	As a user, I can request for the plasma.	10	High	A.Anusudha K.Jeeva
Sprint-4	Database	USN-8	Connect db-2 with the application.	10	High	K.Kaviya M.Sopitha
Sprint-4	End to end testing	USN-9	Implementing end to end testing	10	High	A.Anusudha K.Jeeva

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	11 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

#### Velocity:

we have a 6 days sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \text{sprint duration} / \text{velocity} = 6 / 20$$

$$= 3.33333$$

## 6.3. Reports from jira

The screenshot shows the Jira Software interface for the 'Plasma donor application' project. The left sidebar contains navigation options: PLANNING (Roadmap, Backlog, Board, Reports), DEVELOPMENT (Code, Releases), and Project pages. The main area displays the 'Backlog' for the project. It shows two sprints: 'PDA1 Sprint 1' (24 Oct – 29 Oct, 3 issues) and 'PDA1 Sprint 2' (7 Nov – 12 Nov, 2 issues). Each sprint contains a list of issues with their status (DONE) and assignee (AA, J, K, SM). The issues are: PDA1-25 (As a user, I can register for the application by entering the details.), PDA1-26 (As a user, I can login for the application by entering my Email, Password, a...), PDA1-27 (As a user, I can receive mail for registered for the application.), PDA1-28 (As a user, I can receive the notification when the donor is ready to donate ...), and PDA1-29 (As a user and Admin can view, request plasma and donate plasma.).

Figure 6.3.1

The screenshot shows the Jira Software interface for the 'Plasma donor application' project, displaying the 'Backlog' for Sprints 3 and 4. The left sidebar is the same as in Figure 6.3.1. The main area shows 'PDA1 Sprint 3' (7 Nov – 12 Nov, 2 issues) and 'PDA1 Sprint 4' (18 Nov – 19 Nov, 2 issues). The issues are: PDA1-32 (As a donor, I can enter the details to donate the plasma.), PDA1-33 (As a user, I request for the plasma.), PDA1-36 (Connect database2 to the application.), and PDA1-39 (Implementing end to end testing.).

Figure 6.3.2

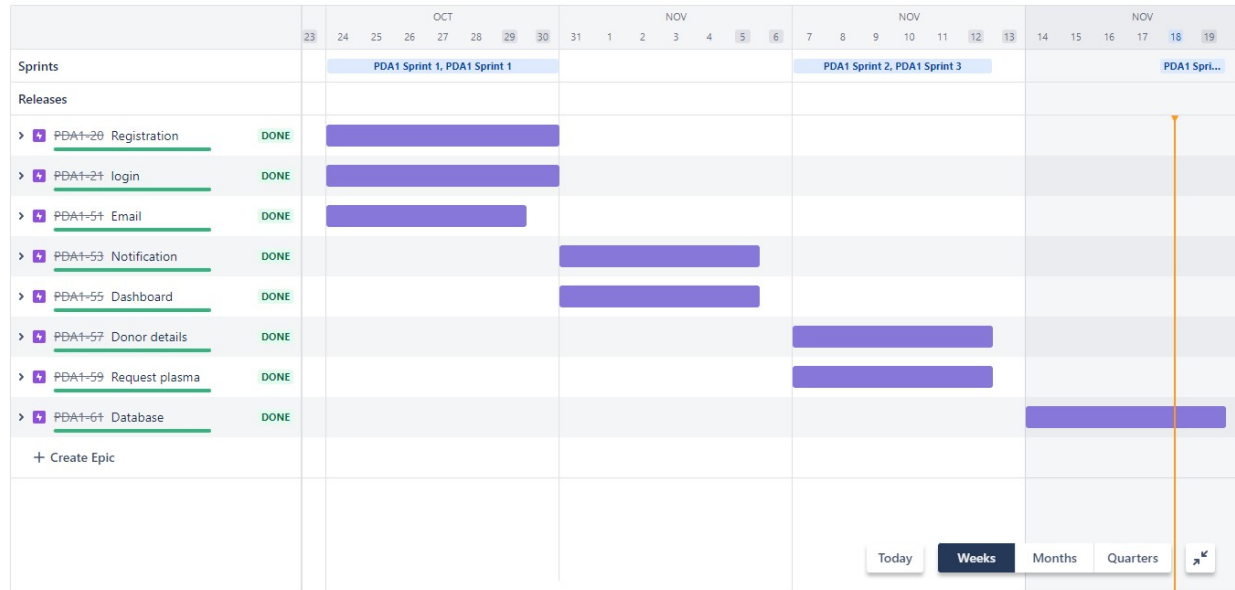


Figure 6.3.3

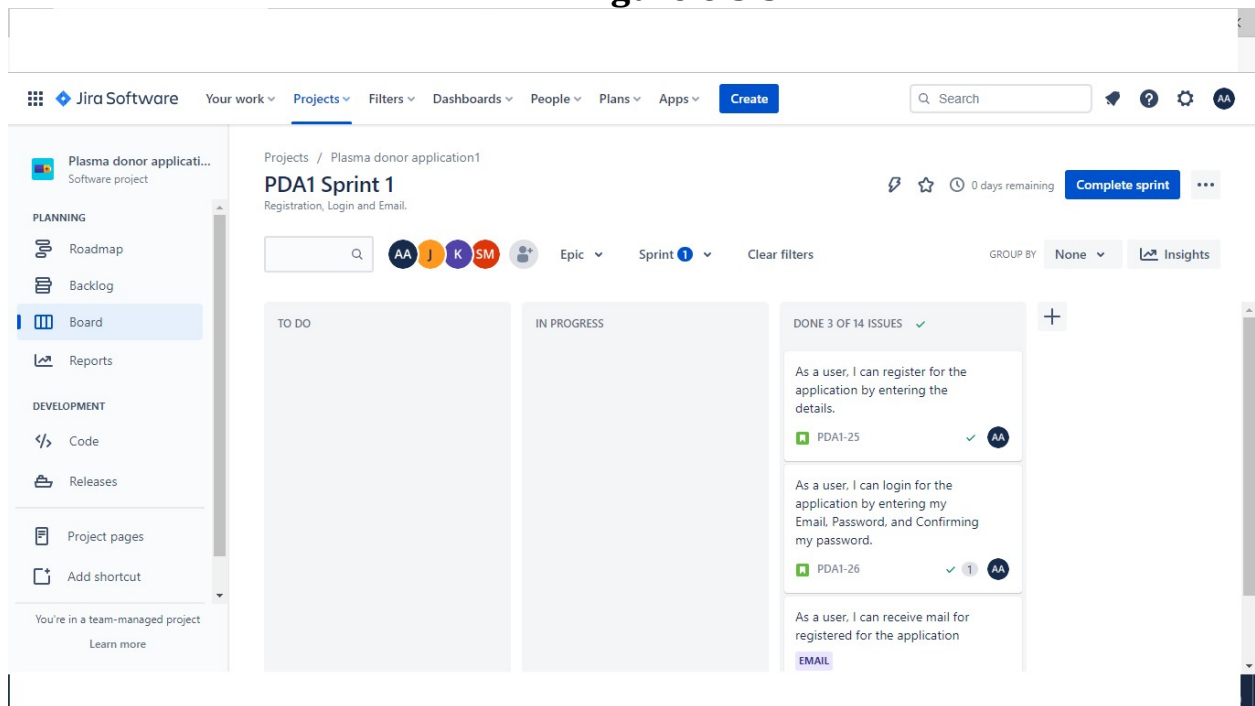


Figure 6.3.4

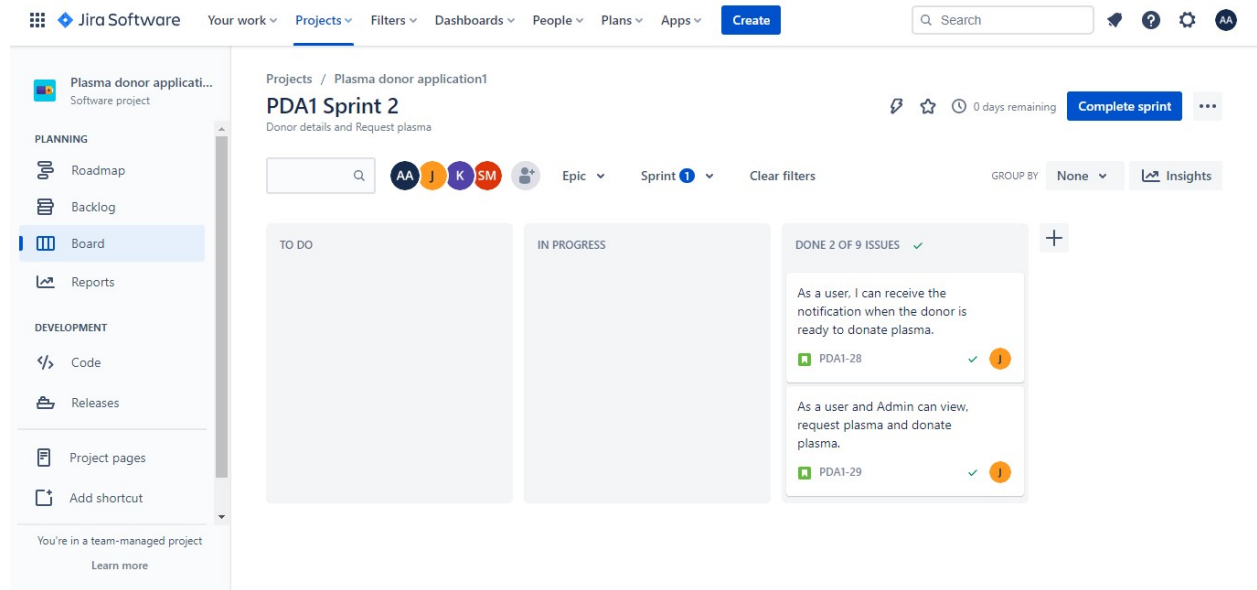


Figure 6.3.5

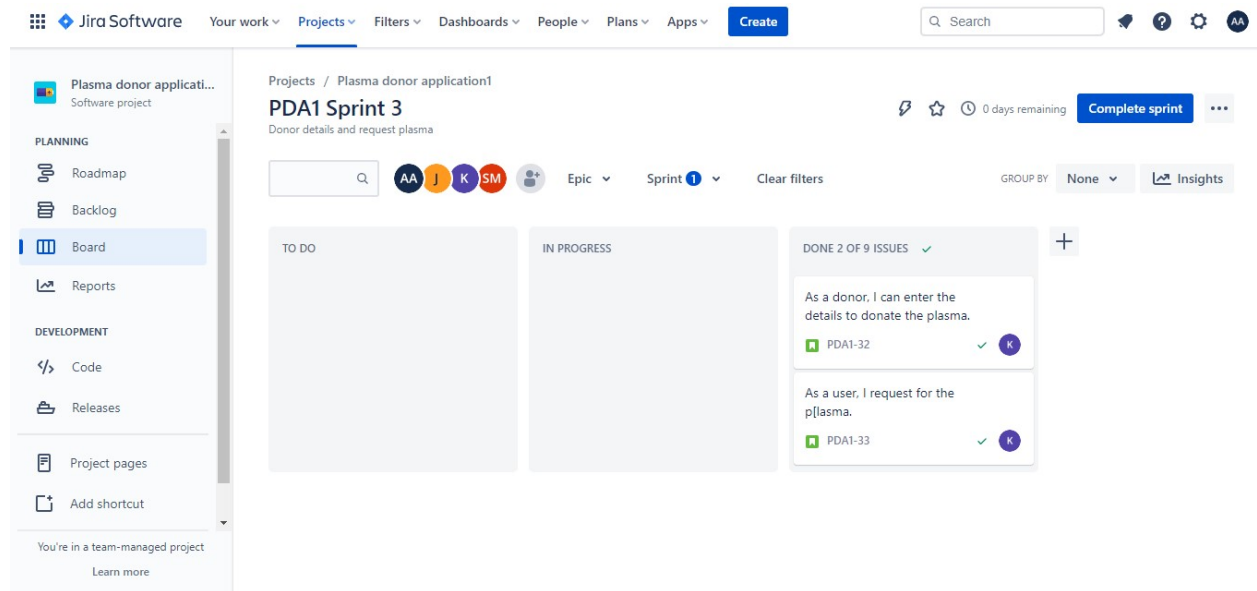


Figure 6.3.6

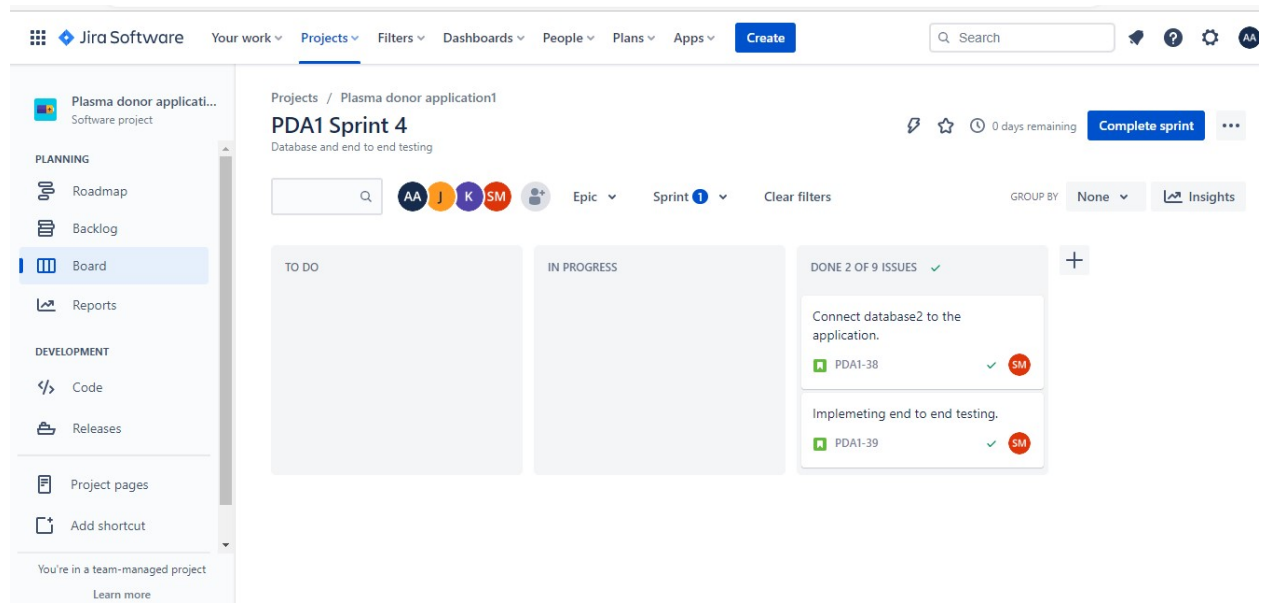


Figure 6.3.7

## 7.CODING AND SOLUTIONING

### 7.1.FEATURE -1:

```
#main.py
#importing libraries
from flask import Flask,render_template, request, redirect, url_for, session
import ibm_db
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30875;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=tqj08800;PWD=UKFB37LGZ3q0xHGy",',' ')
app = Flask(__name__)
print(conn)
print("DATABASE CONNECTED SUCESSFULLY")
```

### 7.2 FEATURE - 2:

```
DATABASE=bludb
HOSTNAME=98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud
PORT=30875
SECURITY=SSL
SSLServerCertificate=DigiCertGlobalRootCA.crt
UID=tqj08800
PWD=UKFB37LGZ3q0xHGy
```

### SENDGRID CODING:

```
import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail

message = Mail(
    from_email='plasmadonar21@gmail.com',
    to_emails='gusamyuktha@gmail.com',
    subject='Hello there! Welcome to PLASMA DONOR APPLICATION',
```

```
        html_content='<strong>PDA warmly welcomes YOU!!!</strong>')
try:
    sg =
SendGridAPIClient('SG.agdL93_hTXSf1Pi8EGC9xw.zlxSPuwGvwW0zz9CaFoG
1kqF Cq9fPLBROX-_ALVk_g')
    response = sg.send(message)
    print(response.status_code)
    print(response.body)
    print(response.headers)

except Exception as e:
    print(str(e))
```

### **7.3 DATABASE SCHEME :**

Database Used : IBM Cloud Database



## 8.TESTING

### 8.1 TESTING CASE:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product.

It provides a way to check the functional of your components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectation and does not fail in an unacceptable manner.

There are various types of test. Each test type addresses a specific testing requirements.

### 8.2 ACCEPTANCE TESTING

#### UAT Execution & Report Submission

##### 1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Plasma Donor Application project at the time of the release to User Acceptance Testing (UAT).

##### 2 .Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20

Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

## 1. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9

Final Report Output	4	0	0	4
Version Control	2	0	0	2

## 9. RESULTS

### 9.1 PERFORMANCE METRICS:

- Project metrics are used to track the progress and performance of a project.
- Monitoring parts of a project like productivity, scheduling, and scope make it easier for team leaders to see what's on track.
- As a project evolves, managers need access to changing deadlines or budgets to meet their client's expectations.

## **10.ADVANTAGES AND DISADVANTAGES:**

### **Advantages:**

- Donating blood can help people with many health conditions, such as those who:
- Have internal or external bleeding due to an injury
- Have sickle cell disease or another illness that affects the blood
- Are undergoing cancer treatment
- Are undergoing surgery, such as cardiovascular or orthopedic surgery
- Have an inherited blood disorder or are undergoing a transplant need treatments involving plasma or other blood products

### **Disadvantages**

- Donating blood is safe, as long as the center follows the standard guidelines.
- The U.S. and many other countries have strict regulations to ensure safety. The FDA and American Association of Blood Banks (AABB) monitor blood banks for this purpose.
- Safety precautions they take include:
- Screening donors for existing health conditions
- Using new needles for each donation
- Having professional staff on hand
- Providing monitoring and refreshments to ensure a safe recovery.

## **11 CONCLUSIONS:**

The efficient way of finding plasma donor for the infected people is implemented using the plasma donor website that is hosted on IBM Cloud platform.

To ensure the smooth functioning of the web site operation. I have hosted the website in IBM Db2 & Kubernetes Cluster to make sure the operations are running successfully Cloud lambda function is used and to deploy the application IBM Db2 service is used.

## **12 FUTURE ENHANCEMENTS:**

Upgrading the UI that is more user-friendly which will help many users to access the website and also ensures that many plasma donors can be added into the community.

Using elastic load balancer, it helps to handle multiple requests at the same time which will maintain the uptime of the website with negligible downtime.

## 13.APPENDIX:

### 13.1:SOURCE CODE:

#### dashboard.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-   scale=1.0">

  <title>Simple HTML HomePage</title>

  <linkrel="stylesheet"href="https://cdnjs.cloudflare.com/ajax/libs/font-
  awesome/5.15.3/css/all.min.css">

  <style>

    @import
url('https://fonts.googleapis.com/css2?family=Sriracha&display=swap');

    margin: 0;

    box-sizing: border-box;

  }

  /* CSS for header */

  .header {

  body {

    background-image: url('567.gif');

    background-repeat: no-repeat;

    background-attachment: fixed;
```

```
    background-size: cover;
}

display: flex;

justify-content: space-between;

align-items: center;

background-color: #f5f5f5;
}

.header .logo {

font-size: 25px;

font-family: 'Sriracha', cursive;

color: #000;

text-decoration: none;

margin-left: 30px;
}

.nav-items {

display: flex;

justify-content: space-around;

align-items: center;

background-color: #f5f5f5;

margin-right: 20px;
}

.nav-items a {

text-decoration: none;
```



```
color: #000;

padding: 35px 20px;
}

.intro {

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

width: 100%;

height: 520px;

background: linear-gradient(to bottom, rgba(0, 0, 0, 0.5) 0%, rgba(0, 0, 0, 0.5)
100%), url("photography1.gif");

background-size: cover;

background-position: center;

background-repeat: no-repeat;
}

.intro h1 {

font-family: sans-serif;

font-size: 60px;

color: #fff;

font-weight: bold;

text-transform: uppercase;

margin: 0;
}
```

```
.intro p {  
  
    font-size: 20px;  
  
    color: #d1d1d1;  
  
    text-transform: uppercase;  
  
    margin: 20px 0;  
  
}  
  
.intro button {  
  
    background-color: #5edaf0;  
  
    color: #000;  
  
    padding: 10px 25px;  
  
    border: none;  
  
    border-radius: 5px;  
  
    font-size: 20px;  
  
    font-weight: bold;  
  
    cursor: pointer;  
  
    box-shadow: 0px 0px 20px rgba(255, 255, 255, 0.4)  
  
}  
  
.achievements {  
  
    display: flex;  
  
    justify-content: space-around;  
  
    align-items: center;  
  
    padding: 40px 80px;  
  
}
```

```
.achievements .work {  
  
  display: flex;  
  
  flex-direction: column;  
  
  justify-content: center;  
  
  align-items: center;  
  
  padding: 0 40px;  
  
}  
  
.achievements .work i {  
  
  width: fit-content;  
  
  font-size: 50px;  
  
  color: #333333;  
  
  border-radius: 50%;  
  
  border: 2px solid #333333;  
  
  padding: 12px;  
  
}  
  
.achievements .work .work-heading {  
  
  font-size: 20px;  
  
  color: #333333;  
  
  text-transform: uppercase;  
  
  margin: 10px 0;  
  
}  
  
.achievements .work .work-text {  
  
  font-size: 15px;
```

```
    color: #585858;

    margin: 10px 0;
}

.about-me {

    display: flex;

    justify-content: center;

    align-items: center;

    padding: 40px 80px;

    border-top: 2px solid #eeeeee;
}

.about-me img {

    width: 500px;

    max-width: 100%;

    height: auto;

    border-radius: 10px;
}

.about-me-text h2 {

    font-size: 30px;

    color: #333333;

    text-transform: uppercase;

    margin: 0;
}

.about-me-text p {
```

```
font-size: 15px;

color: #585858;

margin: 10px 0;

}

/* CSS for footer */

.footer {

display: flex;

justify-content: space-between;

align-items: center;

background-color: #302f49;

padding: 40px 80px;

}

.footer .copy {

color: #fff;

}

.bottom-links {

display: flex;

justify-content: space-around;

align-items: center;

padding: 40px 0;

}

.bottom-links .links {

display: flex;
```

```
flex-direction: column;

justify-content: center;

align-items: center;

padding: 0 40px;
}
```

```
.bottom-links .links span {

font-size: 20px;

color: #fff;

text-transform: uppercase;

margin: 10px 0;
}
```

```
.bottom-links .links a {

text-decoration: none;

color: #a1a1a1;

padding: 10px 20px;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p><a href="https://api.jptok.assistant.watson.cloud.ibm.com/instances/405d2e0a-6b3a-484d-aac0-582f11bbcba9"></a></p>
```

```
<header class="header">
```

```
<a href="#" class="logo"></a>
```

```
<nav class="nav-items">
```

```
<a href="dashboard1.html">Home</a>

<a href="Plasma request.html">Plasma request</a>

<a href="donor details.html">Donate plasma</a>

<a href="Registration.html">Register</a>

<a href="login.html">Login</a>

<a href="about.html">About</a>
```

```
</nav>
```

```
</header>
```

```
<main>
```

```
<div class="intro">
```

```
<h1>plasma donar application</h1>
```

```
<p>A drop for you, an ocean for someone else.</p>
```

```
<button>Donate</button>
```

```
</div>
```

```
<div class="about-me">
```

```
<div class="about-me-text">
```

```
<h2>About plasma</h2>
```

```
<p>Plasma donation involves drawing blood, extracting the plasma, and
returning the rest of the blood to the donor through a process called
plasmapheresis. This type of donation takes slightly longer than whole blood, but as
a person is only donating plasma, it yields a larger volume of blood plasma, meaning
people can donate more frequently..</p>
```

```
</div>
```

```

```

```

    </div>

</main>

<footer class="footer">

    <div class="bottom-links">

        <div class="links">

            <span>More Info</span>

            <a href="#">Home</a>

            <a href="#">About</a>

            <a href="#">Contact</a>

            <a href="https://api.jp-
tok.assistant.watson.cloud.ibm.com/instances/405d2e0a-6b3a-484d-aac0-
582f11bbcb9">chatbot</a>

        </div>

    </div>

</footer>

</body>

</html>

```

### Home.html:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>PLASMA </title>

    <style>

```



```
body {  
  
  margin: 0;  
  
  box-sizing: border-box;  
  
}  
  
.container {  
  
  height: 100vh;  
  
  display: flex;  
  
  justify-content: center;  
  
  align-items: center;  
  
  flex-direction: column;  
  
  background: linear-gradient(to right, rgba(0, 0, 0, .5), rgba(0, 0, 0, .1)),  
url('098.gif');  
  
  background-size: cover;  
  
  background-position: center;  
  
  background-repeat: no-repeat;  
  
}  
  
h1 {  
  
  font-size: 10rem;  
  
  color: #fff;  
  
  text-transform: uppercase;  
  
  font-weight: 700;  
  
  margin-bottom: 1rem;  
  
}  
  
h2 {
```

```
    font-size: 2rem;

    color: #fff;

    text-transform: uppercase;

    font-weight: 700;

    margin-bottom: 1rem;
}

p {

    font-size: 1.5rem;

    color: #fff;

    font-weight: 700;

    margin-bottom: 1rem;
}

a {

    padding: 15px 20px;

    background: #52caee;

    font-size: 1rem;

    text-decoration: none;

    color: #333333;

    border-radius: .25rem;

    box-shadow: 0 0 20px rgba(255, 255, 255, 0.808)
}

</style>

</head>
```

```

<body>

  <div class="container">

    <h1>PLASMA</h1>

    <h2>DONATION</h2>

    <p>Plasma donation is different than donating whole blood. Plasma is the liquid
    part of the blood that contains the elements necessary for blood clotting. Donating it
    is a more complicated process than donating blood. The blood is drawn from your
    arm, the components are separated, and the plasma is put into a separate bag.</p>

    <form>

      <p><a href="dashboard1.html"><button type="submit">Back</button>

      <form action="http://dashboard1.html">

    </form>

  </div>

</body>

</html>

```

### Plasma request.html:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Apply</title>
  <script>
    window.watsonAssistantChatOptions = {
      integrationID: "3b214ae8-6eba-404e-85be-63772e4c2fef", // The ID of this
      integration. region: "eu-gb", // The region your integration is hosted in.
      serviceInstanceID: "3b4bf0dd-e329-4199-99b4-25f9ba1ee227", // The ID of your
      service instance.
      onLoad: function(instance) { instance.render(); }
    };
    setTimeout(function(){
      const t=document.createElement('script');
      chat.global.assistant.watson.appdomain.cloud/versions/" +
      (window.watsonAssistantChatOptions.clientVersion || 'latest')+
      "/WatsonAssistantChatEntry.js";
      t.src="https://web-

```

```
document.head.appendChild(t);

});

</script>

<style>

    body{

background: linear-gradient(to bottom, rgba(0, 0, 0, 0.5) 0%, rgba(0, 0, 0, 0.5)
100%), url("321.gif");

background-size: cover;

background-position: center;

background-repeat: no-repeat;

    }

    h3{

        color: rgb(6, 241, 238);

        font-size: 40px;

    }

    input{

        border: 2px;

        border-style: groove;

        border-radius: 7px;

        padding-top: 8px;

        padding-left: 8px;

        padding-bottom: 8px;

        padding-right: 8px;

        border-color: rgb(0, 255, 251);
```

```
height: 25px;

width:400px;

}

button{

border: 2px;

border-radius: 10px;

padding-top: 8px;

padding-left: 8px;

padding-bottom: 8px;

padding-right: 8px;

border-style: groove;

width: 90px;

opacity: 90%;

border-color: rgb(0, 247, 255);

cursor:pointer;

}

.right{

width:50%;

float: right;

}

.left{

width: 50%;

float: left;
```

```
    }

</style>

</head>

<body>

    <h3>PLASMA REQUEST</h3>

    <form method="post">

        <div class="right">

            <input type="text" name="blood_group" placeholder="Blood Group" required
            /><br><br>

            <input type="number" name="aadhar" placeholder="Aadhar" required
            /><br><br>

            <input type="text" name="state" placeholder="State" required /><br><br>

            <input type="text" name="city" placeholder="City" required /><br><br>

            <input type="password" name="password" placeholder="Password" required
            /><br><br>

        </div>

        <div class="left">

            <input type="text" name="name" placeholder="Name" required /><br><br>

            <input type="email" name="email" placeholder="Email" required /><br><br>

            <input type="number" name="mobile" placeholder="Mobile" required
            /><br><br>

            <input type="number" name="age" placeholder="Age" required/><br><br>

            <input type="text" name="gender" placeholder="Gender" required/><br><br>

        </div>

        <button type="submit">Apply</button>
```

```
</form>

<button type="success">

<p style="color: rgb(27, 161, 27);">success</p>

<p style="color: rgb(214, 25, 25)">error</p>

<a href="/logout">Logout</a>

</body>

</html>
```

### **Donor details.html:**

```
<!DOCTYPE html>

<html>

<head>

<style>

body {

background-image: url('567.gif');

background-repeat: no-repeat;

background-attachment: fixed;

background-size: cover;

}

</style>

</head>

<div class="jumbotron jumbotron-fluid">

<div class="container">

<h1 class="display-4">Welcome To Donar's Portal</h1>
```

<br>

<h2 class="display-6">Enter Donor's Details</h2>

<br>

<p class="lead">A EASY WAY TO DONATE YOUR BLOOD TO THE PEOPLE IN  
NEED</p>

</div>

</div>

<form>

<div class="form-group">

<label for="inputAddress">Name</label>

<input type="text" class="form-control" id="inputName" placeholder="Full  
Name">

</div>

<div class="form-row">

<div class="form-group col-md-6">

<label for="inputEmail4">Email</label>

<input type="email" class="form-control" id="inputEmail"  
placeholder="Email">

</div>

<div class="form-group col-md-6">

<label for="inputPassword4">Contact Number</label>

<input type="tel" class="form-control" id="inputPassword"  
placeholder="Phone">

</div>



```

</div>

<div class="form-row">

  <div class="form-group col-md-6">

    <label for="inputEmail4">Date of Birth</label>

    <input type="text" class="form-control" id="inputEmail4" placeholder="Date
of Birth">

  </div>

  <div class="form-group col-md-6">

    <label>Gender:</label>

    <div class="radio">

      <input type="radio" name="gender" class="radio1" value="Male"><span
class="radio1" required autocomplete="off">Male</span>

      <input type="radio" class="radio2" name="gender"
value="Female">

      <span class="radio2" required autocomplete="off">Female</span>

    </div>

  </div>

</div>

<div class="form-row">

  <div class="form-group col-md-6">

    <label for="inputCity">City</label>

    <input type="text" class="form-control" id="inputCity">

  </div>

  <div class="form-group col-md-4">

    <label for="inputState">State</label>

    <select id="inputState" class="form-control">

      <option value="" selected>Select State</option>

```

<option value="Andaman and Nicobar Islands">Andaman and Nicobar Islands</option>

<option value="Andhra Pradesh">Andhra Pradesh</option>

<option value="Arunachal Pradesh">Arunachal Pradesh</option>

<option value="Assam">Assam</option>

<option value="Bihar">Bihar</option>

<option value="Chandigarh">Chandigarh</option>

<option value="Chhattisgarh">Chhattisgarh</option>

<option value="Dadra and Nagar Haveli">Dadra and Nagar Haveli</option>

<option value="Daman and Diu">Daman and Diu</option>

<option value="Delhi">Delhi</option>

<option value="Goa">Goa</option>

<option value="Gujarat">Gujarat</option>

<option value="Haryana">Haryana</option>

<option value="Himachal Pradesh">Himachal Pradesh</option>

<option value="Jammu and Kashmir">Jammu and Kashmir</option>

<option value="Jharkhand">Jharkhand</option>

<option value="Karnataka">Karnataka</option>

<option value="Kerala">Kerala</option>

<option value="Lakshadweep">Lakshadweep</option>

<option value="Madhya Pradesh">Madhya Pradesh</option>

<option value="Maharashtra">Maharashtra</option>

<option value="Manipur">Manipur</option>

```
<option value="Meghalaya">Meghalaya</option>
<option value="Mizoram">Mizoram</option>
<option value="Nagaland">Nagaland</option>
<option value="Orissa">Orissa</option>
<option value="Pondicherry">Pondicherry</option>
<option value="Punjab">Punjab</option>
<option value="Rajasthan">Rajasthan</option>
<option value="Sikkim">Sikkim</option>
<option value="Tamil Nadu">Tamil Nadu</option>
<option value="Tripura">Tripura</option>
<option value="Uttaranchal">Uttaranchal</option>
<option value="Uttar Pradesh">Uttar Pradesh</option>
<option value="West Bengal">West Bengal</option>
```

```
</select>
```

```
</div>
```

```
<div class="form-group col-md-2">
```

```
<label for="inputZip">Zip</label>
```

```
<input type="text" class="form-control" id="inputZip">
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="inputAddress">Blood Group</label>
```

```
<select id="inputState" class="form-control">
```

```

    <option value="" selected>Select Blood Group</option>
    <option value="A+">A+</option>
    <option value="A-">A-</option>
    <option value="A1+">A1+</option>
    <option value="A1-">A1-</option>
    <option value="A1B+">A1B+</option>
    <option value="A1B-">A1B-</option>
    <option value="A2+">A2+</option>
    <option value="A2-">A2-</option>
    <option value="A2B+">A2B+</option>
    <option value="A2B-">A2B-</option>
    <option value="AB+">AB+</option>
    <option value="AB-">AB-</option>
    <option value="B+">B+</option>
    <option value="B-">B-</option>
    <option value="O+">O+</option>
    <option value="O-">O-</option>
</select>
</div>
<div class="form-group">
    <label for="inputAddress">Diseases (if Any/None)</label>
    <input      type="text"      class="form-control"      id="inputAddress"
placeholder="Diseases">
</div>

```

```
<div class="form-group">

  <div class="form-check">

    <input class="form-check-input" type="checkbox" id="gridCheck">

    <label class="form-check-label" for="gridCheck">

      Agree to terms and conditions

    </label>

    <div class="invalid-feedback">

      You must agree before submitting.

    </div>

  </div>

</div>

<button type="submit" class="btn btn-primary">Submit Details</button>

</form>
```

**about.html:**

```
<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

body {

  background-image: url('765.jpg');

  background-repeat: no-repeat;
```

```
background-attachment: fixed;

background-size: cover;

}

body {

font-family: Arial, Helvetica, sans-serif;

margin: 0;

}

html {

box-sizing: border-box;

}

*, *:before, *:after {

box-sizing: inherit;

}

.column {

float: left;

width: 33.3%;

margin-bottom: 16px;

padding: 0 8px;

}

.card {

box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);

margin: 8px;

}
```

```
.about-section {  
    padding: 50px;  
    text-align: center;  
    background-color: #474e5d;  
    color: white;  
}  
  
.container {  
    padding: 0 16px;  
}  
  
.container::after, .row::after {  
    content: "";  
    clear: both;  
    display: table;  
}  
  
.title {  
    color: grey;  
}  
  
.button {  
    border: none;  
    outline: 0;  
    display: inline-block;  
    padding: 8px;  
    color: white;
```

```
background-color: #000;

text-align: center;

cursor: pointer;

width: 100%;

}

.button:hover {

background-color: #555;

}

@media screen and (max-width: 650px) {

.column {

width: 100%;

display: block;

}

}

</style>

</head>

<body>

<div class="about-section">

<h1>About Us Page</h1>

<p>Plasma includes antibodies that help you fight illness. It's chock-full of proteins that drug manufacturers can utilize to build life-saving drugs, but it's in short supply. The demand for plasma is considerably greater now because the COVID-19 outbreak disrupted plasma collections across the country.</p>

<p>Resize the browser window to see that this page is responsive by the way.</p>

</div>
```



<h2 style="text-align:center">Our Team</h2>

<div class="row">

<div class="column">

<div class="card">



<div class="container">

<h2>Anusudha</h2>

<p class="title">team leader</p>

<p>Team leaders are responsible for training team members, setting strategy and monitoring progress towards goals.</p>

<p>anusudha160@gmail.com</p>

<p><button class="button">Contact:9894596603</button></p>

</div>

</div>

</div>

<div class="column">

<div class="card">



<div class="container">

<h2>Jeeva</h2>

<p class="title">team member</p>

<p>Valuable team members should be experts in their chosen disciplines – they shouldn't be expected to switch roles with one another when there isn't a major overlap in competencie.</p>

<p>jeevakaruppiyah21@gmail..com</p>

```
<p><button class="button">Contact:6374646402</button></p>

</div>

</div>

</div>

<div class="column">

  <div class="card">

    <div class="container">

      <h2>kaviya</h2>

      <p class="title">team member</p>

      <p>Valuable team members should be experts in their chosen disciplines – they
      shouldn't be expected to switch roles with one another when there isn't a major
      overlap in competence.</p>

      <p>kavyakutti2020@gmail.com</p>

      <p><button class="button">Contact:7806887541</button></p>

    </div>

  </div>

</div>

</div>

<div class="column">

  <div class="card">

    <div class="container">

      <h2>Sopitha</h2>
```

```
<p class="title">team member</p>

<p>Valuable team members should be experts in their chosen disciplines – they
shouldn't be expected to switch roles with one another when there isn't a major
overlap in competence.</p>

<p>muthuselvamabi19078@gmail.com</p>

<p><button class="button">Contact:7904754363</button></p>

</div>

</div>

</div>

</body>

</html>
```

### **Login.html:**

```
<!DOCTYPE html>

<html lang="en" >

<head>

<meta charset="UTF-8">

<title>Animated Background Login Screen</title>

<link rel='stylesheet' href='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.0/css/
bootstrap.min.css'>

<link rel="stylesheet" href="./style.css">

</head>

<body>

<!-- partial:index.partial.html -->
```

```

<div class="container">

    <div id="login-box">

        <div class="logo">

            <h1class="logo-caption">

                <span class="tweak">L</span>ogin</h1>

            </div><!-- /.logo -->

            <div class="controls">

                <input
                    type="text"
                    name="username"
                    placeholder="Username" class="form-control" />

                <input type="text" name="username" placeholder="Password"
                    class="form-control" />

                <button type="button" class="btn btn-default btn-block btn-
custom">Login</button>

            </div>

        </div>

    </div>

<div id="particles-js"></div>

<!-- partial -->

    <script
src='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.0/css/bootstrap.min.css'></
script>

    <scriptsrc='https://code.jquery.com/jquery1.11.1.min.js'></script>

    <script src="./script.js">

</script>

```

```
</body>
```

```
</html>
```

### **Index1.html:**

```
<!DOCTYPE html>
```

```
<!-- Code By Webdevtrick ( https://webdevtrick.com ) -->
```

```
<html lang="en" >
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Animated Login Page | Webdevtrick.com</title>
```

```
<linkhref="https://fonts.googleapis.com/css?family=Rubik&display=swap"
rel="stylesheet">
```

```
<link rel="stylesheet" href="style.css">
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<div class="left-section">
```

```
<div class="header">
```

```
<h1 class="animation a1">Welcome Back!</h1>
```

```
<h4 class="animation a2">Log in for entering your membership
dashboard.</h4>
```

```
</div>
```

```
<div class="form">
```

```
<input type="email" class="form-field animation a3" placeholder="Username">
```

```
<input type="password" class="form-field animation a4"
placeholder="Password">
```

```
<p class="animation a5"><a href="#">Forgot Password</a></p>

<button class="animation a6">LOGIN</button>

</div>

</div>

<div class="right-section"></div>

</div>

</body>

</html>
```

#### **Style.css:**

```
{

    Margin;

    padding: 0;

    box-sizing: border-box;

}

.full-page

{

    height: 100vh;

    width: 100%;

    background-image:url('C:\Users\user\Desktop\Project\image1.jpg');

    position: absolute;

}

.sub-page
```

```
{  
    width: 1266px;  
    height: 559px;  
    position: absolute;  
    background-image: url('C:\Users\user\Desktop\Project\giphy.jpeg');  
    background-size: cover;  
    background-position: center;  
    left: 50px;  
    top: 50px;  
}  
  
.navigation-bar  
{  
    display: flex;  
    align-items: center;  
    padding: 20px;  
    padding-left: 80px;  
    padding-right: 30px;  
    padding-top: 50px;  
}  
  
.logo  
{  
    position: fixed;  
    margin-top: 10px;
```

```
}

.logo a
{
    text-decoration: none;
    color: white;
    font-size: 30px;
}

nav
{
    flex: 1;
    position: fixed;
    right: 0;
}

nav ul
{
    display: inline-block;
    list-style: none;
}

nav ul li
{
    display: inline-block;
    margin-right: 90px;
    margin-top: 17px;
```



```
}

nav ul li a
{
    text-decoration: none;
    font-size: 20px;
    color: Red;
    font-family: sans-serif
}

nav ul li a: hover
{
    color: saddlebrown;
}

.row
{
    display: flex;
    align-items: center;
    flex-wrap: wrap;
    justify-content: space-around;
}

.col-1
{
    flex-basis: 50%;
    min-width: 300px;
```

```
}

.form-box
{
    width: 300px;
    height: 400px;
    position: relative;
    top: 50px;
    left: 120px;
    background: rgba(0,0,0,0.6);
}

.main-heading
{
    color: orangered;
    padding-bottom: 20px;
}

.form
{
    position: relative;
    margin: 0 auto 100px;
    padding: 45px;
    text-align: center;
}

.form input
```

```
{  
  
    font-family: sans-serif;  
  
    outline: none;  
  
    border: none;  
  
    border-bottom: 1px solid black;  
  
    width: 100%;  
  
    margin: 0 0 15px;  
  
    padding: 15px;  
  
    font-size: 14px;  
  
}
```

```
.form button  
  
{  
  
    font-family: sans-serif;  
  
    width: 100%;  
  
    color: white;  
  
    font-size: 14px;  
  
    cursor: pointer;  
  
    padding: 15px;  
  
    border: none;  
  
    background: coral;  
  
}
```

```
.form .message  
  
}
```

```

        font-size: 12px;

        margin: 15px 0 0;

        color: white;
    }

    .form .message a
{
        color: orangered;

        text-decoration: none;
    }

    .form .register-form{

        display: none;
    }

    .defination
{
        text-align: left;

        font-size: 30px;

        color: white;

        font-family: 'Kaushan Script', cursive;

        padding-left: 60px;

    }

```

### **app.py**

```

from flask import *

import sys

```

```
app=Flask(__name__)
```

```
app.secret_key="123"
```

```
class User:
```

```
    def __init__(self,username,password):
```

```
        self.username=username
```

```
        self.password=password
```

```
users=[]
```

```
users.append(User(username="anusudha",password="anu160@"))
```

```
users.append(User(username="jeeva",password="jeeva21@"))
```

```
users.append(User(username="sopitha",password="sobi@14"))
```

```
users.append(User(username="kaviya",password="kaviya03@"))
```

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

```
def dashboard():
```

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

```
        uname=request.form.get['uname']
```

```
        upass = request.form.get['upass']
```

```
        if users.username==uname and users.password==upass:
```

```

        return redirect(url_for('user'))

    else:

        flash("Username or Password Mismatch...!!!",'danger')

        return redirect(url_for('login'))

    return render_template("dashboard.html")

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

    if request.method == 'POST':

        uname=request.form['uname']

        upass = request.form['upass']

        for data in users:

            if data.username==uname and data.password==upass:

                return redirect("/dashboard")

        else:

            flash("Username or Password Mismatch...!!!",'danger')

            return redirect("/")


    return render_template("index1.html")


@app.route("/donate-plasma",methods=['GET','POST'])
def donateplasma():

```

```

if request.method=='POST':

    iname = request.form.get("inputName")

    # uemail = request.form.get("inputEmail")

    # ucontact = request.form.get("inputcontact")

    # udob = request.form.get("inputDoB")

    # ugender = request.form.get("gender")

    # ucity = request.form.get("inputCity")

    # ustate = request.form.get("inputState")

    # upin = request.form.get("inputZip")

    print(iname,file=sys.stdout)

    return render_template("donor_details.html")

@app.route("/plasma-req",methods=['GET','POST'])

def plasmarequest():

    if request.method=='POST':

        uname=request.form['uname']

        upass = request.form['upass']


    for data in users:

        if data.username==uname and data.password==upass:

            session['userid']=data.id

            g.record=1

            return redirect(url_for('/'))

    else:

```

```

        g.record=0

    if g.record!=1:

        flash("Username or Password Mismatch...!!!",'danger')

        return redirect(url_for('login'))

    return render_template("plasma_request.html")


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

    if request.method=='POST':

        uname=request.form['uname']

        upass = request.form['upass']


        for data in users:

            if data.username==uname and data.password==upass:

                session['userid']=data.id

                g.record=1

                return redirect(url_for('/'))

            else:

                g.record=0

        if g.record!=1:

            flash("Username or Password Mismatch...!!!",'danger')

            return redirect(url_for('login'))

    return render_template("home.html")

```



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

def about():

    if request.method=='POST':

        uname=request.form['uname']

        upass = request.form['upass']


    for data in users:

        if data.username==uname and data.password==upass:

            session['userid']=data.id

            g.record=1

            return redirect(url_for('/'))

        else:

            g.record=0

    if g.record!=1:

        flash("Username or Password Mismatch...!!!",'danger')

        return redirect(url_for('login'))

    return render_template("about.html")

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

def register():

    if request.method=='POST':

        uname=request.form['uname']

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

```
for data in users:

    if data.username==uname and data.password==upass:

        session['userid']=data.id

        g.record=1

        return redirect(url_for('/'))

    else:

        g.record=0

if g.record!=1:

    flash("Username or Password Mismatch...!!!",'danger')

    return redirect(url_for('login'))

return render_template("Registration.html")
```

```
@app.before_request
```

```
def before_request():
```

```
    if 'userid' in session:
```

```
        for data in users:
```

```
            if data.id==session['userid']:
```

```
                g.user=data
```

```
@app.route('/user')
```

```
def user():
```

```
    if not g.user:
```

```
        return redirect(url_for('login'))
```

```
    return render_template('index1.html')
```

```
@app.route('/logout')
```

```
def logout():
```

```
    session.clear()
```

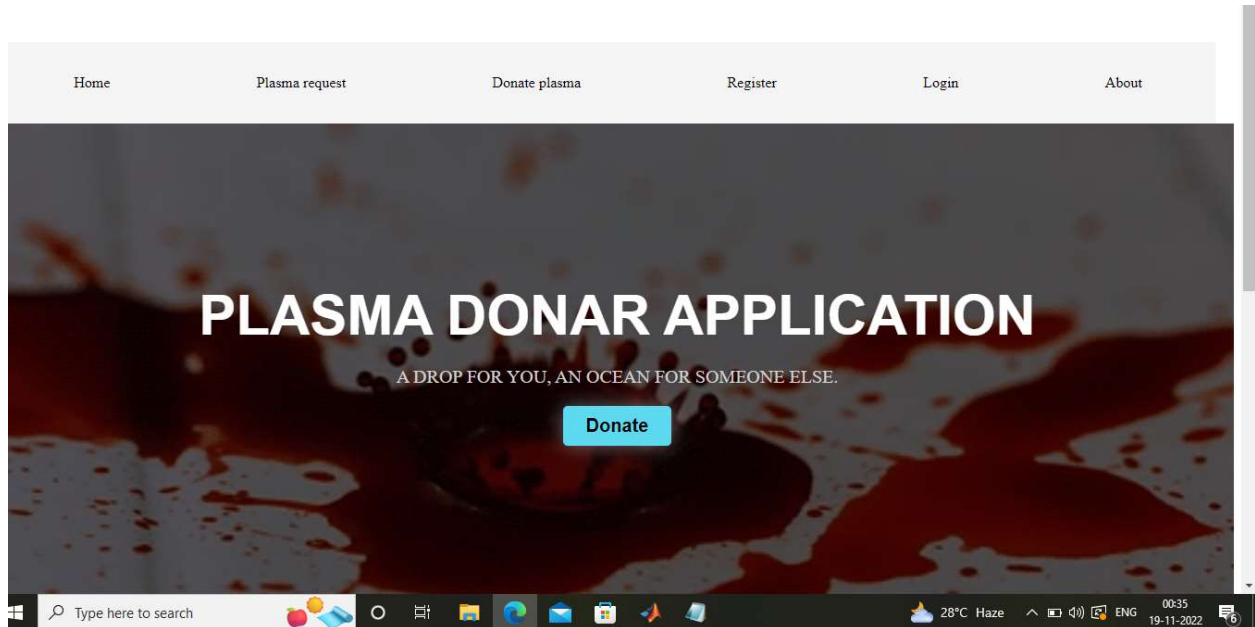
```
    return redirect(url_for('login'))
```

```
if __name__ == '__main__':
```

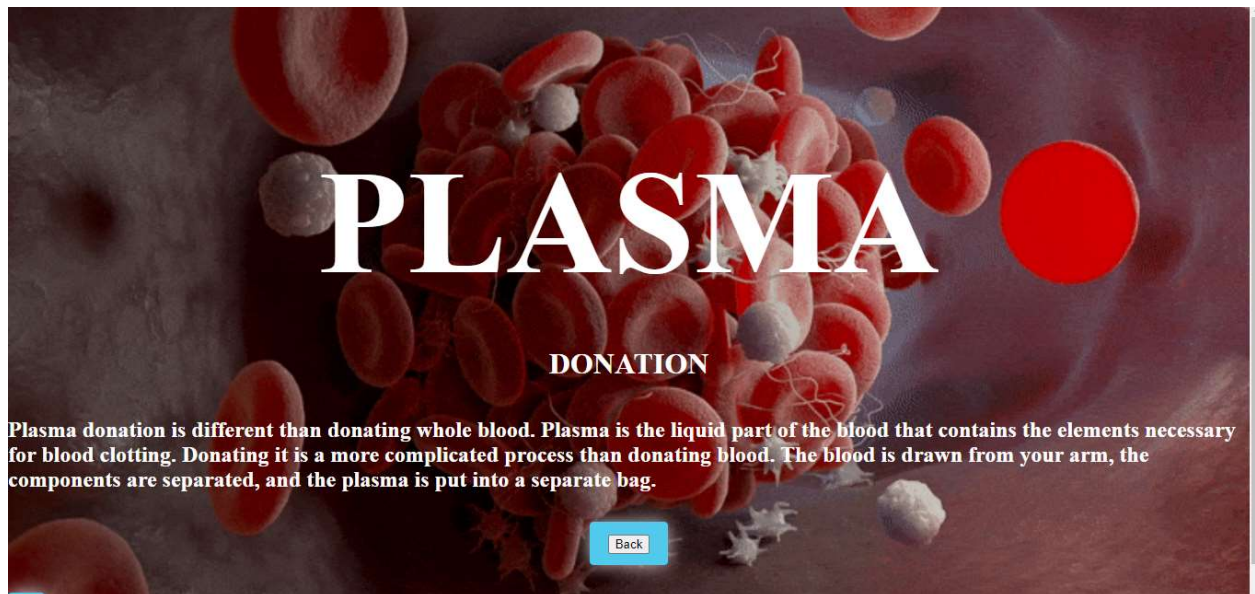
```
    app.run(debug=True)
```

**Output:**

**Dashboard:**



**Home page:**



## Donate plasma:

### Welcome To Donar's Portal

#### Enter Donor's Details

A EASY WAY TO DONATE YOUR BLOOD TO THE PEOPLE IN NEED

Name

Email

Contact Number

Date of Birth

Gender:-  
☐ Male ☐ Female

City

State

Zip

Blood Group

Diseases (if Any/None)

☐ Agree to terms and conditions  
You must agree before submitting.

## Blood request:

REGISTRATION

Hello, Friend!

Enter your personal details and start journey with us

**Login Details**

User Name:-

Full Name:-  Email Id:-

Password:-  Confirm Password:-

**Contact Details**

Mobile Number:-  Address:-

Pincode  City:-  State:-

**Personal Details**

Date Of Birth:-

Blood Group

Plasma Type

Weight In Kg [Approx]:-

Gender:-  
☐ Male ☐ Female

☐ I agree to have my contact details broadcasted to the registered donors of B24U.net

## About page:

## About Us Page

Plasma includes antibodies that help you fight illness. It's chock-full of proteins that drug manufacturers can utilize to build life-saving drugs, but it's in short supply. The demand for plasma is considerably greater now because the COVID-19 outbreak disrupted plasma collections across the country.

Resize the browser window to see that this page is responsive by the way.

## Our Team

### Anusudha

team leader

Team leaders are responsible for training team members, setting strategy and monitoring progress towards goals.

anusudha160@gmail.com

Contact:9894596603

### Jeeva

team member

Valuable team members should be experts in their chosen disciplines – they shouldn't be expected to switch roles with one another when there isn't a major overlap in competence.

jeevakarupiah21@gmail..com

Contact:6374646402

### kaviya

team member

Valuable team members should be experts in their chosen disciplines – they shouldn't be expected to switch roles with one another when there isn't a major overlap in competence.

kavyakutti2020@gmail.com

Contact:7806887541

**IBM DB2:**

IBM Db2 on Cloud

Load DataLoad HistoryTablesViewsIndexesAliasesMQTsSequencesApplication objects

Find schemas or tables

Refresh

Schemas

<input checked="" type="checkbox"/>	Name	Type	Tables
<input checked="" type="checkbox"/>	DWP42983	User	6

Total: 1, selected: 1

Tables

New table

☐

Name

Schema

Properties

☐

DATA

DWP42983

...

☐

DATABASE

DWP42983

...

☐

DB2

DWP42983

...

☐

LOGIN

DWP42983

...

☐

REGISTER

DWP42983

...

☐

REQUEST

DWP42983

...

Total: 6, selected: 0

IBM Db2 on Cloud

Load DataLoad HistoryTablesViewsIndexesAliasesMQTsSequencesApplication objects

DWP42983.LOGIN

Back

Export to CSV

NAME	PASSWORD
A.Anusudha	anu160@
K.Jeeva	jeeva21@
M.Sopitha	sobi@14
k.kaviya	kaviya03@



IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

DWP42983.REQUEST

Back

Export to CSV

NAME	EMAIL_ID	PASSWORD	MOBILE_NO	AGE	GENDER	BLOOD_GROUP	AADHAR	STATE	CITY
Anusudha	anusudha160@gmail.com	anu@16	9894596603	22	Female	B+	794512874391	tamil nadu	trichy
Arun	arunrolex45@gmail.com	arun@50	7809326722	21	Male	B+	456259951003	tamil nadu	dingugal
Ismail	mohamadabi14@gmail.com	abi@147	8883968206	19	Male	B+	761439660324	tamil nadu	salem
Jeeva	jeevakarupiah21@gmail.com	jeevaK@21	6374646402	21	Female	B+	790347843404	tamil nadu	thanjavur
Kathir	kathiravan157@gmail.com	kathi34@	8423290377	25	Male	AB-	790367893256	tamil nadu	kerla
Kaviya	kavyakutti2020@gmail.com	kavi@22	7806887541	21	Female	AB+	667342162943	tamil nadu	madurai
Ishiva	lvhimu8iii14@gmail.com	lvhi@14	9834588402	19	Female	O+	23467287017	tamil	coimbatore

## 13.2 .GITHUB

**IBM-Project-3384-1658557794**

**Project demo link:**

**[https://drive.google.com/drive/folders/1VRrJJhjynESUBqOdc7jV005n2bP\\_0oFP?usp=share\\_link](https://drive.google.com/drive/folders/1VRrJJhjynESUBqOdc7jV005n2bP_0oFP?usp=share_link)**