

# PROJECT REPORT

## PLASMA DONOR APPLICATION

<b>Date</b>	21 November 2022
<b>Team ID</b>	PNT2022TMID11900
<b>Project Name</b>	Plasma Donor Application
<b>Document</b>	Project Report

### TEAM MEMBERS

<b>ROLE</b>	<b>NAME</b>	<b>ROLL NO</b>
TEAM LEADER	ROOPAN RAJ P	73771921177
TEAM MEMBER 1	MUTHUKUMAR S	73771921165
TEAM MEMBER 2	KARTHIKEYAN S	73771921143
TEAM MEMBER 3	THARUN R	73771921207

### 1.INTRODUCTION

- 1.1 Project Overview
- 1.2 Purpose

### 2. LITERATURE SURVEY

- 2.1 Existing problem
- 2.2 References
- 2.3 Problem Statement Definition

### 3. IDEATION & PROPOSED SOLUTION

- 3.1 Empathy Map Canvas
- 3.2 Ideation & Brainstorming
- 3.3 Proposed Solution
- 3.4 Problem Solution fit

#### **4. REQUIREMENT ANALYSIS**

- 4.1 Functional requirement
- 4.2 Non-Functional requirements

#### **5. PROJECT DESIGN**

- 5.1 Data Flow Diagrams
- 5.2 Solution & Technical Architecture
- 5.3 User Stories

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

- 6.1 Sprint Planning & Estimation
- 6.2 Sprint Delivery Schedule
- 6.3 Reports from JIRA

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

- 7.1 Request for Plasma
- 7.2 Make Plasma Donation
- 7.3 SendGrid API Integration
- 7.4 Database Schema

#### **8. TESTING**

- 8.1 Test Cases
- 8.2 User Acceptance Testing

#### **9. RESULTS**

- 9.1 Performance Metrics

#### **10. ADVANTAGES & DISADVANTAGES**

#### **11. CONCLUSION**

#### **12. FUTURE SCOPE**

#### **13. 13. APPENDIX**

Source Code

# **INTRODUCTION :**

## **1.1 PROJECT OVERVIEW**

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 COVIDpositive patients and help them recover faster. Therapy, which is considered reliable and safe. If a particular person has fully recovered from COVID19, they are 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.

## **1.2 PURPOSE**

The purpose of the system is to providing the platform for the plasma donors and the recipient and it acts as a connecting bridge between the plasma donors and the needy. The donor simply registers their profile with us by providing the necessary information to create a donor account. If a person in the need of plasma can easily able to make a plasma request for the required blood type just fill in up the request form. The donor who are willing to donate can able to accept the plasma request and make an appointment for donation once the donor eligibility has been verified. The proposed solution makes the plasma hunting process very simple just by connecting the donor and the needy around.

## 2. LITERATURE SURVEY

### 2.1 EXISTING PROBLEM

### 2.2 REFERENCES

### 2.3 PROBLEM STATEMENT

YEAR	TITLE	AUTHOR(s)	TECHNIQUE(s)	PROS	CONS
2022	Instant Plasma Donor Recipient connector web application	Kalpna Devi Guntoju, Tejaswini Jalli, Sreeja Uppala, Sanjay Malliseti	Web Technologies, API, Database	The Donor needs to upload their recovered COVID-19 Certificate and it required to verified by the blood bank. It is a user-friendly application. It will help people to find plasma easily.	This is system is closed for general plasma donation and mainly focused on COVID-19 patients for plasma donation
2021	BDoor App-Blood Donation Application using Android Studio	S Periyanaayagi, A Manikandan, M Muthukrishnan, and M Ramakrishnan	Android, Flutter UI, Dart, Firebase, Decision tree algorithm	The Donor details are verified before they allow to donate and have to authorised by institution. The Verification and validation are done in email base.	The android mobile user will not be able to insert or view details if the server goes down. Thus, there is disadvantage of single point failure.
2020	Lifesaver E-Blood Donation App Using Cloud	Rishab Chakrabarti, Asha Darade, Neha Jadhav, Prof. S. M. Chitalkar	E-health, GPS, Blood bank database, Cloud Computing	Reduction in the errors of blood bank using most eligible donor method. Direct Communication Between donor and the person in need of blood During the Emergency situation.	The user given details are maintained unverified.
2020	Developing a plasma donor application using Function-as-a-service in AWS	Aishwarya R Gowri	Serverless, aws, plasma theory, covid19, dynamoDB, cloud	The efficient way of finding plasma donor for the infected people. Aws lambda function is used and to deploy the application AWS EC2 service is used.	The user interface can be better than now.
2019	D'WORLD: Blood Donation App Using Android	A. Meiyappan, K. Loga Vignesh, R. Prasanna, T. Sakthivel	Android, Global Positioning System (GPS), Mobile Computing	When the giver gives the blood, it will naturally evacuate the contributor detail for next three months. It additionally confirms with the Department of Health and Welfare to guarantee the benefactor medical case history.	The user must have an device with android operating system with an active internet connection to interact with this application.

2018	Automated blood bank system using Raspberry PI	Ashlesha C. Adsul, V. K. Bhosale, R. M. Autee	Raspberry Pi, Embedded Blood Bank, GSM, Android	When there is urgent need for blood then If this model is adopted the caller is immediately connected to the donor	Tackling the fake users.
------	--	---	---	--	--------------------------

2018	Blood donation and life saver-blood donation app	M.R. Anish Hamlin, J. Albert Mayan	Android, GPS, Cloud Computing	One-Time Password (OTP) is used to verify the donor, once the donor accepts the request. Once the donor donates the blood it will automatically remove the donor detail for next three months.	This application searches for donors only in the nearest areas.
2018	Android Based Health Application in Cloud Computing for Blood Bank	Sayali Dhond, Pradnya Randhavan, Bhagyashali Munde, Rajnandini Patil,	Cloud Computing, Global Positioning System (GPS), Web Technologies, Android.	Accessibility and availability are the criteria on which an application is designed for its success in the IT market.	Requires the patient records to be accurate and accessible.
2016	mHealth: Blood donation application using android smartphone	Muhammad Fahim, Halil Ibrahim Cebe, Jawad Rasheed and Farzad Kiani	Android (operating system), medical computing, mobile computing	mHealth is one of the best possible concepts for the provision of healthcare services and improve quality of life.	We have to utilize the cloud computing service for keeping the application data available, anywhere and anytime.
2015	An Android Application for Volunteer Blood Donors	Sultan Turhan	Distance Calculation, Web Services, GPS, Databases	This application helps health care centres to provide the blood as quick as possible when their stocks are insufficient. The application sends periodically actual location information of available donors to main system and the blood requests to the donors.	If the stocks are insufficient, the only source of blood supply will be the people who come to the health centre and donate the blood on a voluntary basis

### 3. IDEATION & PROPOSED SOLUTION 3.1

#### Empathy Map Canvas

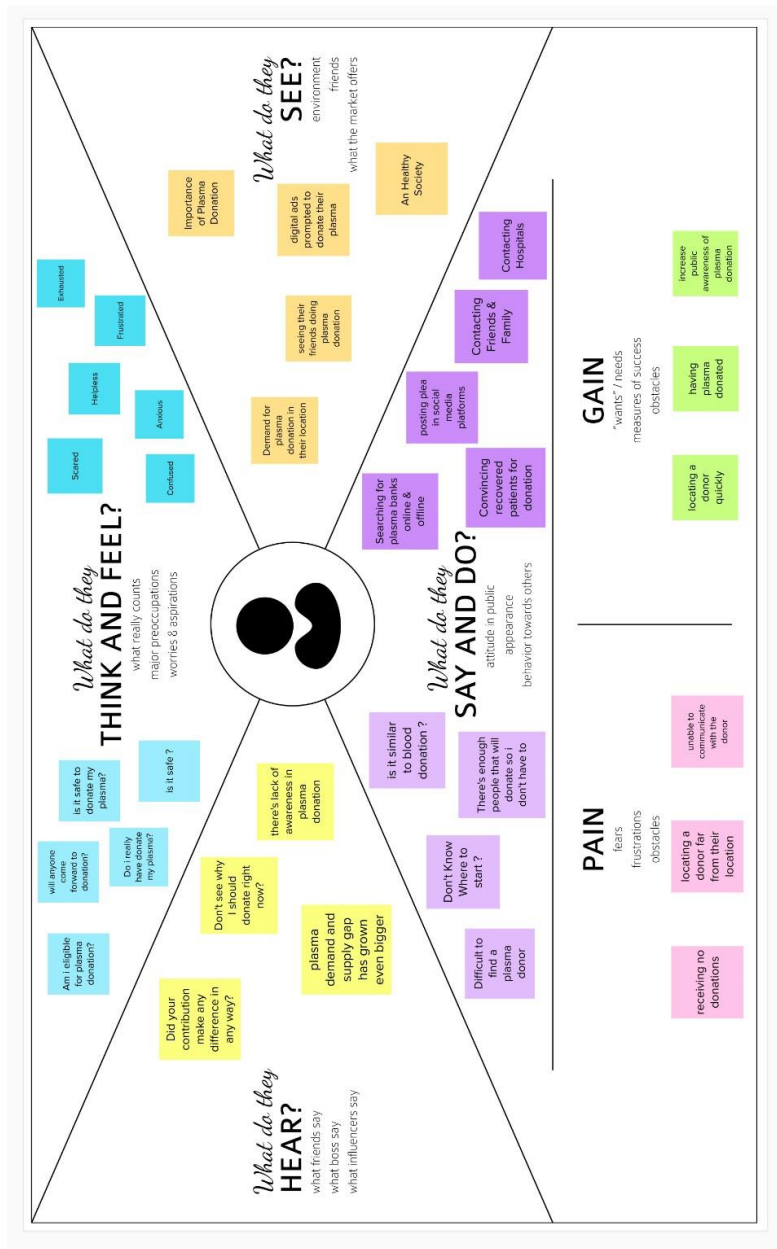


Fig 3.1.1 Empathy Map Canvas

[illegible]

### 3.3 PROPOSED SOLUTION

S. No	Parameter	Description
1	Problem Statement (Problem to be solved)	Plasma is a critical part of the treatment for many serious health problems. This is why there are blood drives asking people to donate blood plasma. 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.
2	Novelty / Uniqueness	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. In This way, the one who in need in plasma can able to make a request, then the application can able to read the information of donors that are stored in database and informing up the donors regarding that request
3	Novelty / Uniqueness	This application can able to perform certain functionality and possess certain feature which are unique. Those are listed below: <ul style="list-style-type: none"> <li>• Those who want to donate their plasma can do by simply register by uploading their covid-19 recovery certificate.</li> <li>• It can able to find donors who are located close to the needy by using GPS location tracking.</li> <li>• A chat-bot to answer frequently asked question about plasma donation.</li> </ul>
<u>4</u>	Social Impact / Customer Satisfaction	By using the application one can easily able to find the donor at emergency situations and the one who willing to donate their plasma can easily be connected with the needy. Since this process takes place continuously, we can build a healthy society of tomorrow.
<u>5</u>	Business Model (Revenue Model)	We can provide some additional medical services in order to generate some revenue. Medical services like blood test, medical record management, medical transportation service and some other health care service.
<u>6</u>	Scalability of the Solution	Since the whole application is developed based on micro-services architecture, the scalability of the application is made easy. The application can ably scale as the users grow and handle the traffic at any situations.



### 3.4 PROPOSED SOLUTION FIT

Problem-Solution fit canvas 2.0		Purpose / Vision To Build a Web-based application for Plasma Donation	
Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span> <ul style="list-style-type: none"> <li>Donors who wants to donate their plasma.</li> <li>Seekers or needy who are in the need of plasma.</li> </ul>	<b>6. CUSTOMER CONSTRAINTS</b> <span>CC</span> <ul style="list-style-type: none"> <li>Easy finding of donors</li> <li>Availability of plasma types</li> <li>Donors within their nearest location</li> </ul>	<b>5. AVAILABLE SOLUTIONS</b> <span>AS</span> <ul style="list-style-type: none"> <li>Asking their friends and family for donating their plasma</li> <li>Posting the situation in the social media</li> <li>Contacting nearest blood banks and NGO's</li> </ul>
	Focus on J&P, tap into BE, understand RC	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <span>J&amp;P</span> <ul style="list-style-type: none"> <li>Helps the needy or plasma seeker to find the donors available to their nearest location.</li> <li>Provide a platform to volunteer donors to help the needy.</li> <li>Lack of information about the donors.</li> <li>The details of donors to be maintained properly.</li> </ul>	<b>9. PROBLEM ROOT CAUSE</b> <span>RC</span> <ul style="list-style-type: none"> <li>During the COVID 19 crisis, the requirement of plasma became a high priority and the donor count has become low.</li> <li>Saving the donor information and helping the needy by notifying the current donors list, would be a helping hand.</li> </ul>
Identify strong TR & EM		<b>3. TRIGGERS</b> <span>TR</span> <ul style="list-style-type: none"> <li>Seeing the donors count become low.</li> <li>Emergency situation of plasma need.</li> </ul>	<b>10. YOUR SOLUTION</b> <span>SL</span> <p>In regard to the problem faced, a web-based application is to be built which would take the donor details, store them and inform them upon a request.</p>
	<b>4. EMOTIONS: BEFORE / AFTER</b> <span>EM</span> <p>Confused, Anxious, Exhausted, Helpless, Scared, Relaxed, Motivated, Blessed</p>	<b>8.2 OFFLINE</b> <ul style="list-style-type: none"> <li>Arranging the required medical infrastructure for the donation process.</li> <li>Donating the plasma.</li> </ul>	

Problem-Solution fit canvas is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 license  
Created by Daria Nepriakhina / Amaltama.com

**AMALTAMA**

Fig 3.4.1 Proposed Solution Fit

## 4.REQUIREMENT ANALYSIS

### 4.1 FUNCTIONAL REQUIREMENTS

NFR No.	Non-Functional Requirement	Description
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Creating Donor Profile	Volunteer Donor able to create their donor profile by providing their medical information and past donations in the form.
FR-4	Making Plasma request	Filling up the plasma request form, user can able to make a request for plasma.
FR-5	Virtual Donor Card	Active Donors will get a virtual donor card represents their donation activity.
FR-6	Statistical Dashboard	Every user will be provided with the statistical dashboard which will contains the information about the availability of donors.
FR-7	Help Chatbot	Users can ask their doubts about plasma donation to the help chatbot.

## 4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
NFR-1	Usability	User must able to feel easy to perform all the operation supported by the system and it should contain the user-friendly UI and UX
NFR-2	Security	The system must be designed in the way that it should prevent the unauthorized access and cyberattacks
NFR-3	Reliability	The system able to free from failures and should be perform consistently irrespective of the amount of load given.
NFR-4	Performance	The system must be able to performs in terms of responsiveness and stability under a particular workload.
NFR-5	Availability	The system should able to remains operational under normal circumstances in order to serve its intended purpose.
NFR-6	Scalability	The system must able to increase or decrease in performance and cost in response to changes in application.

## 5. PROJECT DESIGN

### 5.1 DATA FLOW DIAGRAM

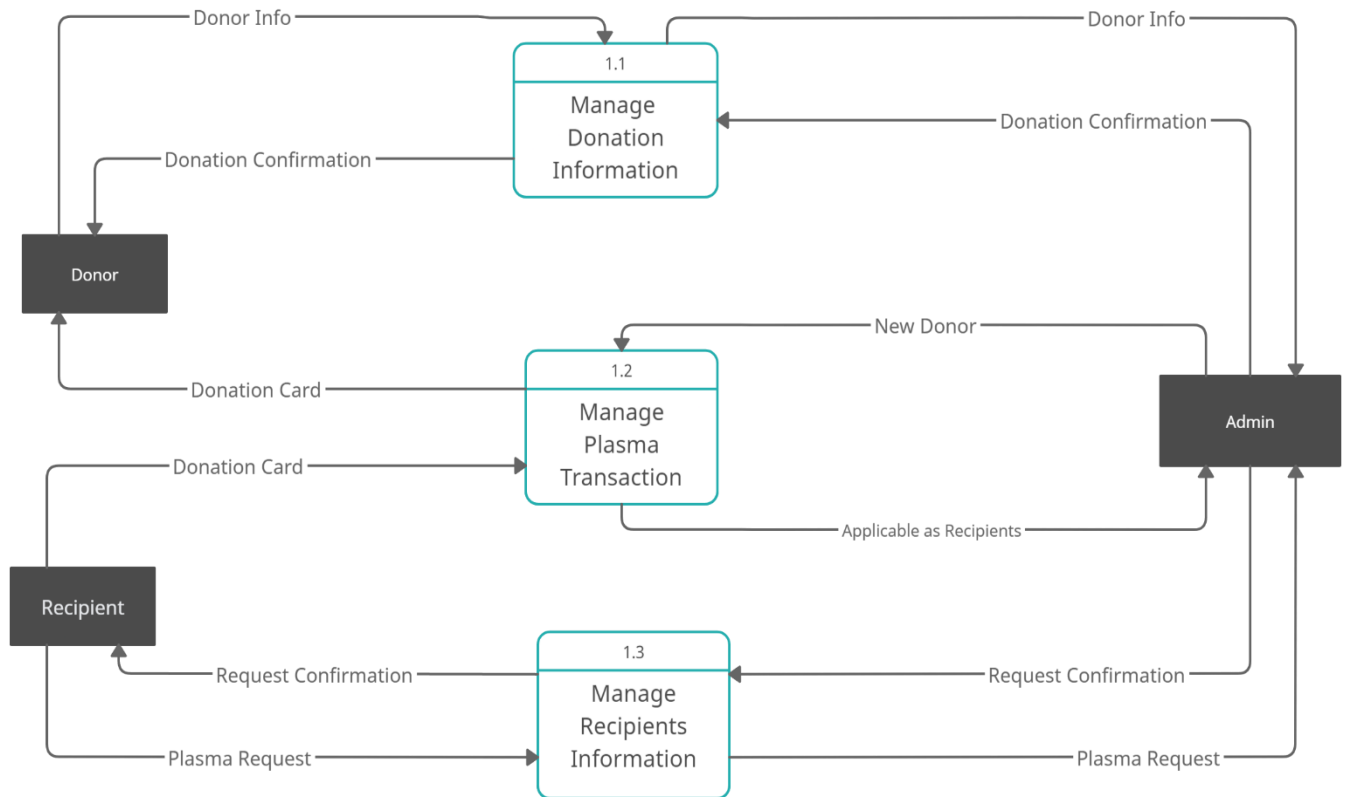


Fig.5.1. Data flow Diagram

## 5.2 SOLUTION AND TECHNICAL ARCHITECTURE

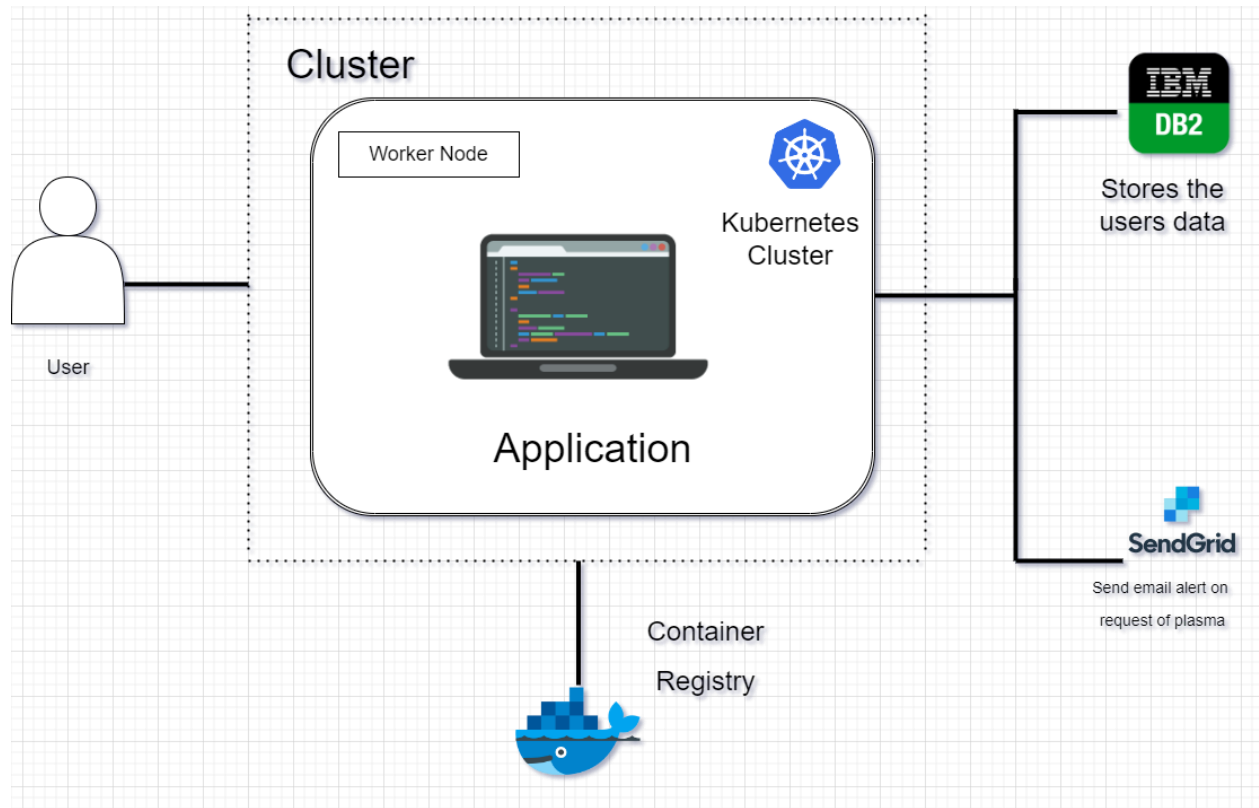


Fig.5.2. Solution and Technical Architecture

## 5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
-----------	-------------------------------	-------------------	-------------------	---------------------	----------	---------

Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email & password		High	Sprint-1
	Donor registration	USN-5	As a Donor, I can create a donor profile by entering my medical and previous donation information	I can create a donor profile and can able to receive plasma request	High	Sprint-2
	Recipient Request	USN-6	As a recipient, I can make a Plasma request by entering the required information in the form	I can able to make a plasma donation request	High	Sprint-2
	Donor Virtual Card	USN-7	As a donor, I can get my virtual donor card after my successful donations	I can receive a virtual donor card which will showcase my donation activity	Medium	Sprint-3
<b>User Type</b>	<b>Functional Requirement (Epic)</b>	<b>User Story Number</b>	<b>User Story / Task</b>	<b>Acceptance criteria</b>	<b>Priority</b>	<b>Release</b>
	Help Chatbot	USN-8	As a user, I can clarify all my doubts by asking the help chatbot in help section	I can able to clarify all my doubts about donation	Medium	Sprint-3

	FAQ section	USN-9	As a user, I can clear my doubts about using the application in the FAQ section	I can able to use the application freely after clearing my doubts	Low	Sprint-3
Administrator	Verification and Validation	USN-10	As an admin, I can verify and validate the information provided by the both donor and recipient	I can validate and verify the user information	High	Sprint-3
	Manage and maintain the transaction	USN-11	As an admin, I can manage all the transaction takes places in the application and maintain real-time database	I can manage and maintain all the transaction and keep database updated	High	Sprint-3

## 6. PROJECT PLANNING AND SCHEDULING

### 6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
--------	-------------------------------	-------------------	-------------------	--------------	----------	--------------

Sprint1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	Sanjay J, Dinesh S
Sprint1	Email verification	USN-2	As a user, I will receive confirmation email once I have registered for the application	3	High	Arul Niranjana V, Thiruvankadakumar L
Sprint1		USN-3	As a user, I can register for the application through Google	2	Medium	Sanjay J
Sprint1	Login	USN-4	As a user, I can log into the application by entering email & password	5	High	Arul Niranjana V
Sprint1	Donor Profile	USN-5	As a user, I can able to register me as a verified donor for plasma.	3	High	Dinesh S, Thiruvankadakumar L
Sprint2	Dashboard / Statistics	USN-6	As a user, I can able to see the statistics of plasma availability, donor count on my dashboard.	5	High	Sanjay J, Thiruvankadakumar L
Sprint2	Virtual Donor Card	USN-7	As a user, I can able to get a virtual donor card after a verified successful plasma donation.	3	Medium	Dinesh S, Arul Niranjana V
Sprint2	Plasma Request	USN-8	As a user, I can able to make a plasma request in the application	5	High	Sanjay J, Arul Niranjana V, Thiruvankadakumar L
Sprint2	Verification of Donor's details	USN-9	As an admin, I will verify the details provided by the donors so only the genuine donors can able to use the application	2	Medium	Sanjay J, Dinesh S
Sprint2		USN-10	As a user, I will upload my COVID-19 recovery certification in order to become a verified registered donor	5	High	Sanjay J, Arul Niranjana V
<b>Sprint</b>	<b>Functional Requirement (Epic)</b>	<b>User Story Number</b>	<b>User Story / Task</b>	<b>Story Points</b>	<b>Priority</b>	<b>Team Members</b>
Sprint3	Accept the donation request	USN-11	As a user and a registered donor, I will accept the plasma request for my specific type	5	High	Arul Niranjana V, Thiruvankadakumar L



Sprint3	Donation Appointment	USN-12	As a user and a volunteer donor, I can able to book an appointment for donation after accepting the plasma request	3	High	Sanjay J, Arul Niranjan V
Sprint3	Communication Channel	USN-13	As a user, I can able to communicate with the donor personally within the application	3	Medium	Dinesh S, Thiruvankadachumar L
Sprint3		USN-14	As a user and a registered donor, I can able to share my location with the recipient after accepting their plasma request	2	Medium	Dinesh S, Arul Niranjan V
Sprint3	Administrator	USN-15	As an admin, I will store the registered donor details after verification into the database and maintain it periodically	5	High	Sanjay J, Thiruvankadachumar L
Sprint4	Support	USN-16	As a user, I can able to ask my doubts and basic question related to plasma donation to the help chat-bot	2	Medium	Sanjay J, Dinesh S
Sprint4		USN-17	As a user, I will clarify the answers for the frequently asked question about the plasma donation in the FAQ section	5	High	Arul Niranjan V, Thiruvankadachumar L
Sprint4	About	USN-18	As a user and if I am new to plasma donation, I can read about the plasma and plasma donation in dedication about section	3	Medium	Dinesh S, Thiruvankadachumar L
Sprint4	Administrator	USN-19	As an admin, I will approve all the plasma transaction in the application after the proper verification	5	High	Sanjay J, Arul Niranjan V
Sprint4		USN-20	As an admin, I will update the plasma availability and donors count periodically	3	Medium	Thiruvankadachumar L, Dinesh S

## 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	18	6 Days	24 Oct 2022	29 Oct 2022	18	29 Oct 2022

Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	18	6 Days	07 Nov 2022	12 Nov 2022	18	12 Nov 2022
Sprint-4	18	6 Days	14 Nov 2022	19 Nov 2022	18	19 Nov 2022

<b>Sprint</b>	<b>Total Story Points</b>	<b>Duration</b>	<b>Sprint Start Date</b>	<b>Sprint End Date (Planned)</b>	<b>Story Points Completed (as on Planned End Date)</b>	<b>Sprint Release Date (Actual)</b>	<b>Average Velocity (AV) = Sprint duration / velocity</b>
Sprint-1	18	6 Days	24 Oct 2022	29 Oct 2022	18	29 Oct 2022	3
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022	3.33
Sprint-3	18	6 Days	07 Nov 2022	12 Nov 2022	18	12 Nov 2022	3
Sprint-4	18	6 Days	14 Nov 2022	19 Nov 2022	18	19 Nov 2022	3

## 6.3 REPORTS FROM JIRA

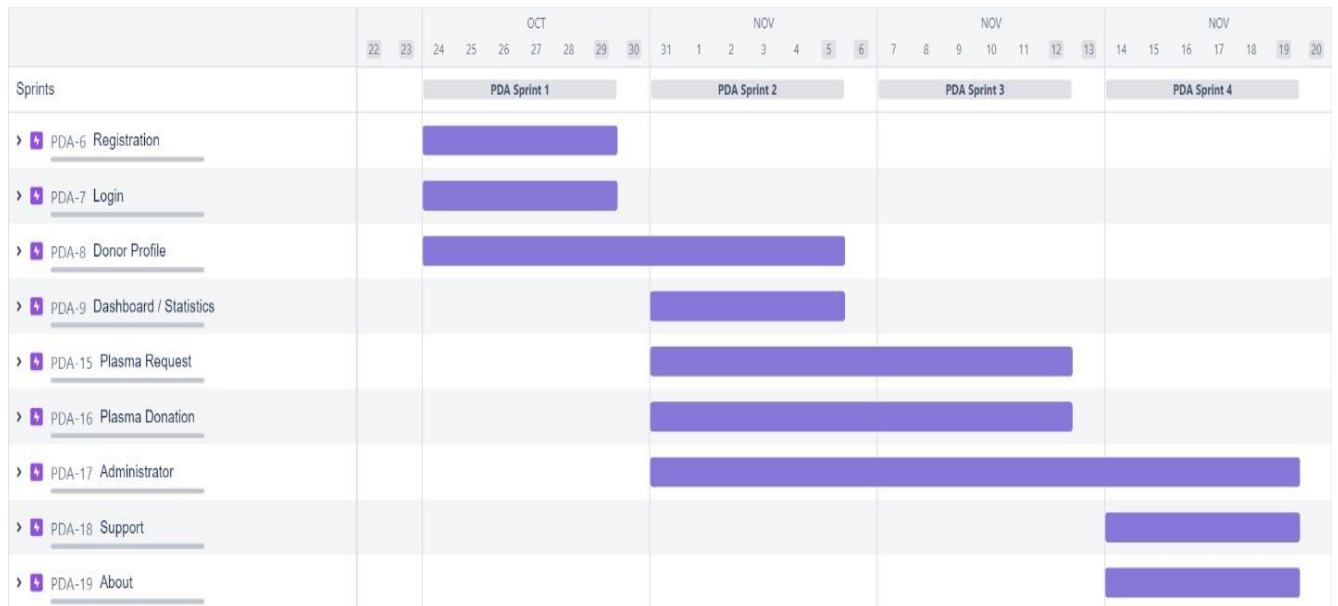


Fig.6.1.1

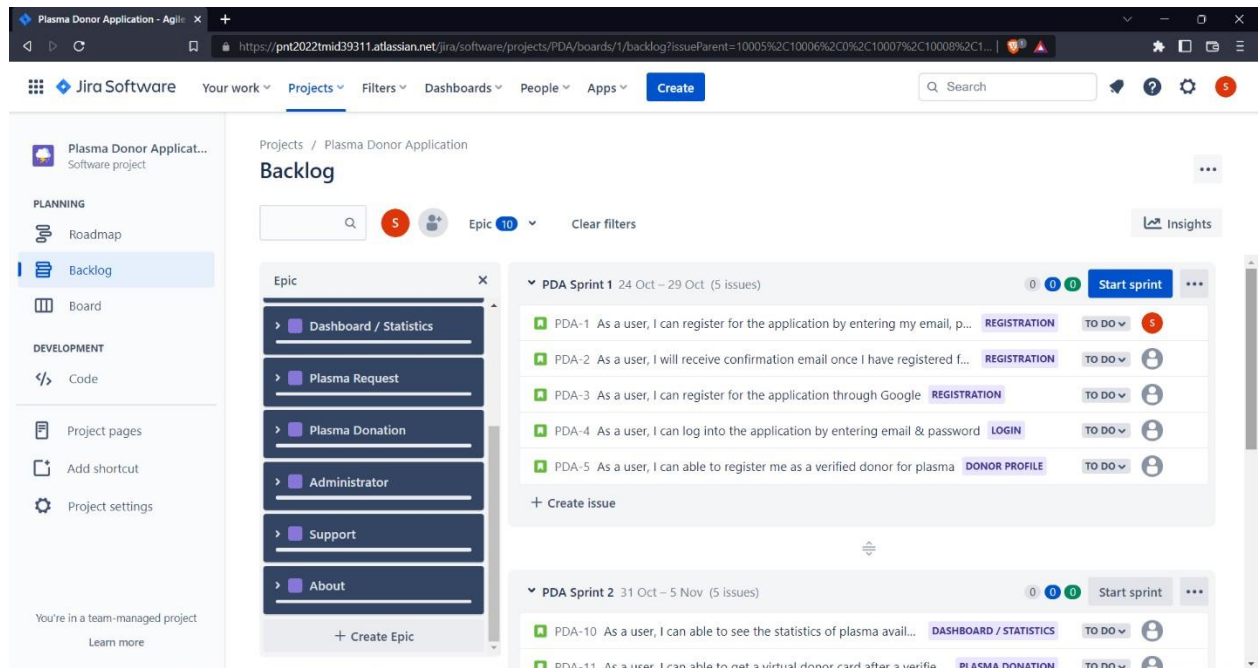


Fig.6.1.2

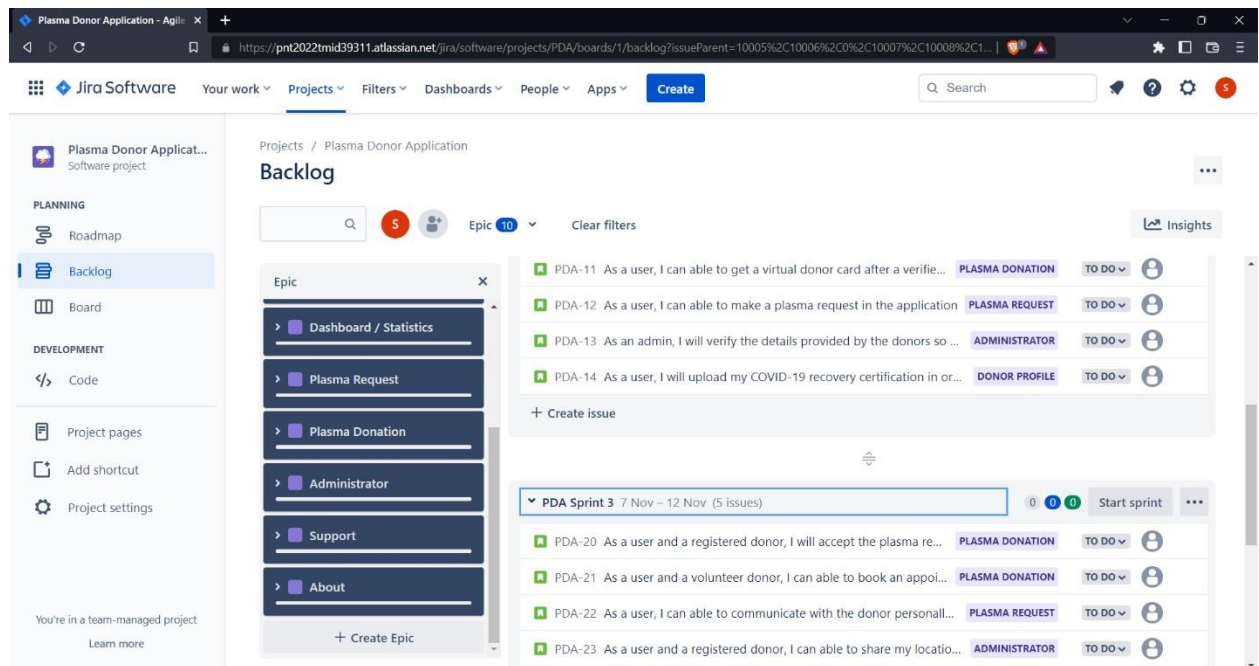


Fig.6.1.3

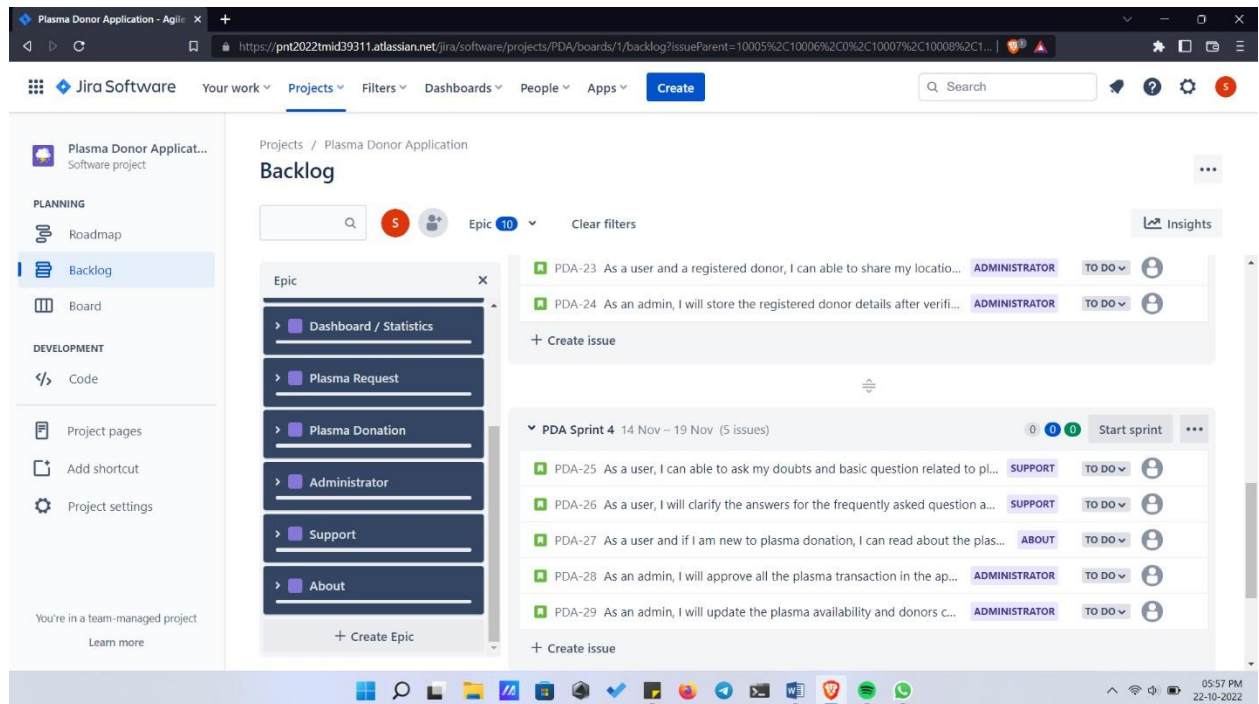


Fig.6.1.4

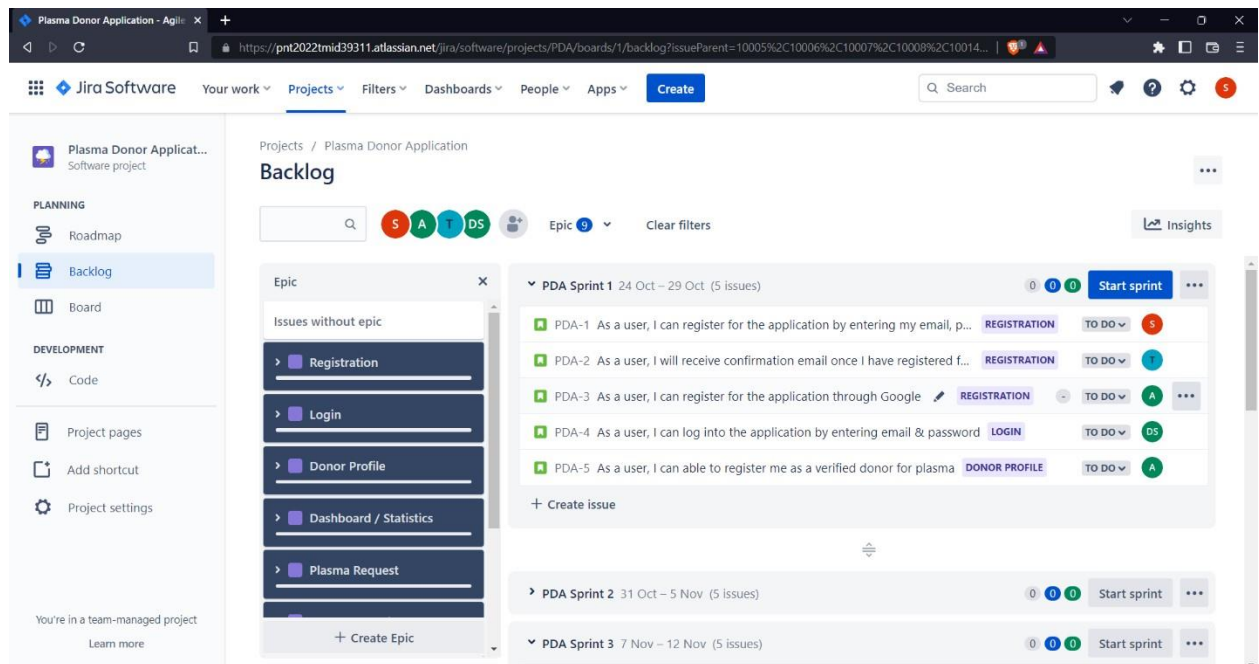


Fig.6.1.5

## **7. CODING AND SOLUTIONING**

### **7.1 Request for Plasma**

#### **Web Framework**

A Web Application Framework or a simply a Web Framework represents a collection of libraries and modules that enable web application developers to write applications without worrying about low-level details such as protocol, thread management, and so on.

#### **Flask**

Flask is a web application framework written in Python. It was developed by Armin Ronacher, who led a team of international Python enthusiasts called Pocco. Flask is based on the Werkzeug WSGI toolkit and the Jinja2 template engine. Both are Pocco projects.

#### **WSGI**

The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications.

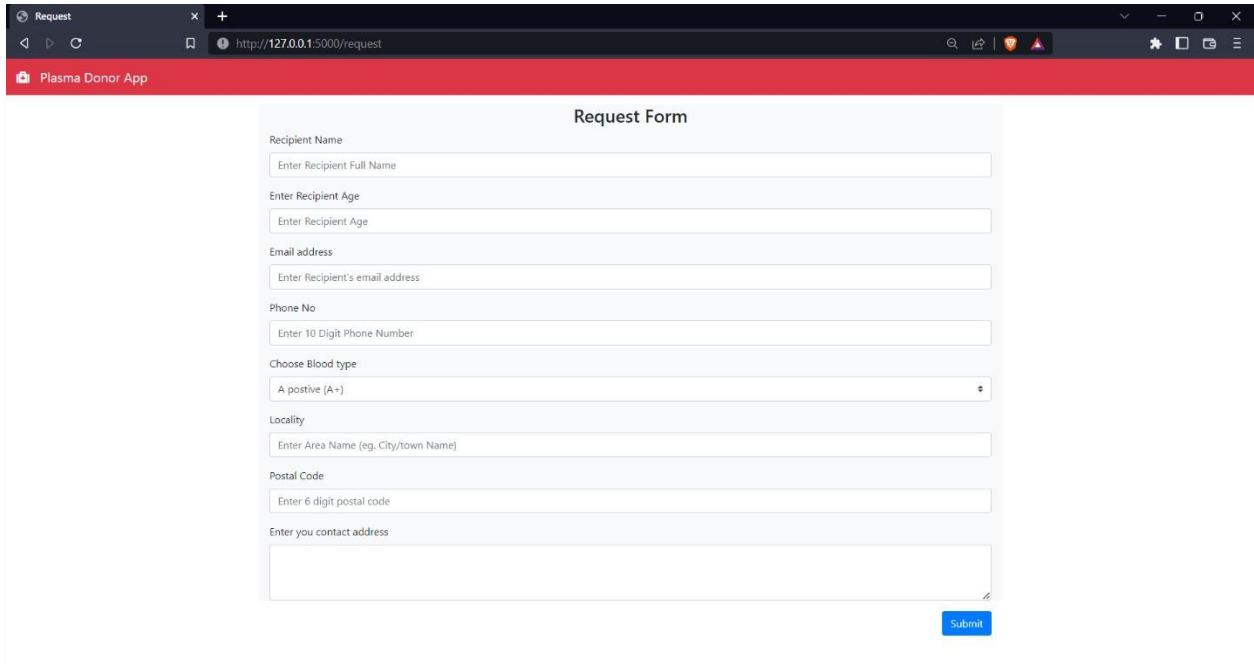
#### **Werkzeug**

Werkzeug is a WSGI toolkit that implements requests, response objects, and utility functions. This enables a web frame to be built on it. The Flask framework uses Werkzeug as one of its bases.

#### **Jinja2**

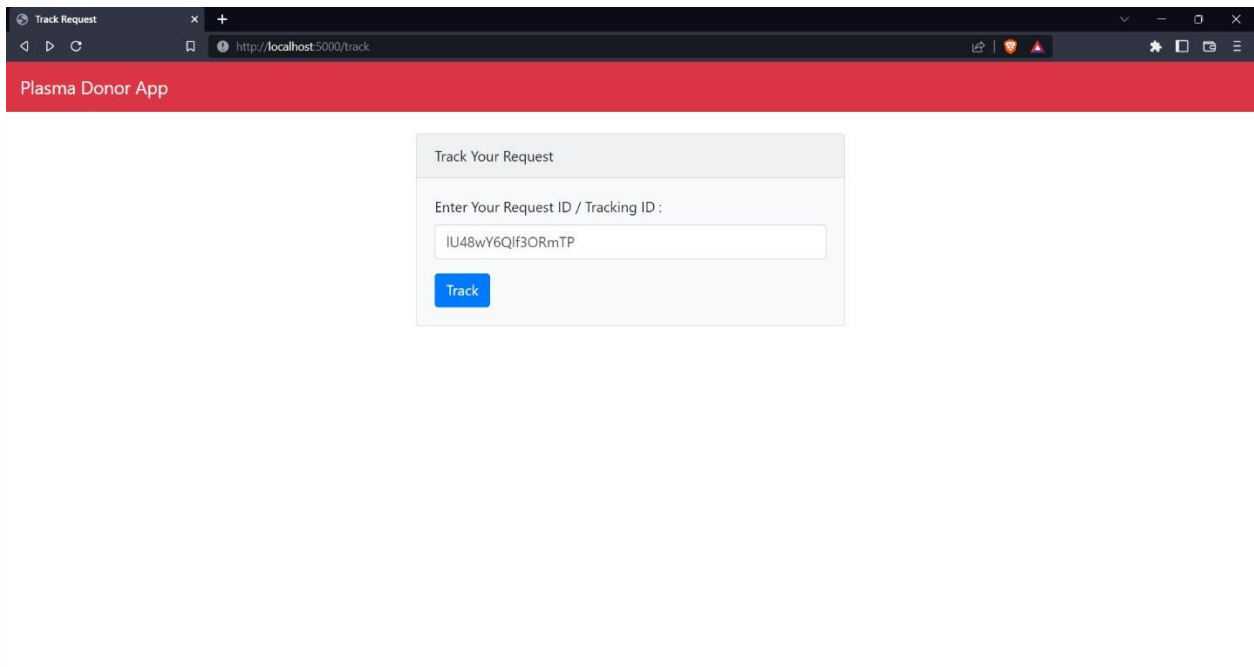
Jinja2 is a popular template engine for Python. A web template system combines a template with a specific data source to render a dynamic web page.

## ScreenShots



A screenshot of a web browser displaying the 'Request Form' of the 'Plasma Donor App'. The browser's address bar shows 'http://127.0.0.1:5000/request'. The form is titled 'Request Form' and contains several input fields: 'Recipient Name' (with placeholder 'Enter Recipient Full Name'), 'Enter Recipient Age' (with placeholder 'Enter Recipient Age'), 'Email address' (with placeholder 'Enter Recipient's email address'), 'Phone No' (with placeholder 'Enter 10 Digit Phone Number'), 'Choose Blood type' (a dropdown menu showing 'A positive (A+)'), 'Locality' (with placeholder 'Enter Area Name (eg. City/town Name)'), 'Postal Code' (with placeholder 'Enter 6 digit postal code'), and 'Enter you contact address' (a large text area). A blue 'Submit' button is located at the bottom right of the form.

Fig.7.1.1



A screenshot of a web browser displaying the 'Track Request' page of the 'Plasma Donor App'. The browser's address bar shows 'http://localhost:5000/track'. The page has a red header with the text 'Plasma Donor App'. The main content area features a box titled 'Track Your Request' which contains the text 'Enter Your Request ID / Tracking ID :', a text input field containing the value 'IU48wY6Qlf3ORmTP', and a blue 'Track' button.

Fig.7.1.2

## CODE:

### Request.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>Request</title>

  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.14.0/css/all.css">

  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
>

</head> <style>  main{    margin:
70px 0 0 0 ;

  }

</style>

<body>

  <header>

    <nav class="navbar navbar-dark navbar-expand-sm bg-danger fixed-top">

      <a href="/" class="navbar-brand">

        <i class="fas fa-medkit"></i> &nbsp;  

        Plasma Donor App

      </a>

    </nav>

  </header>

  <main>
```



```

<div class="container-sm bg-light ">

    <h3 class="text-center">Request Form</h3>

    <form action="{ {url_for('do_request')}} " method="post">

        <div class="form-group">

            <label for="recipient-name">Recipient Name</label>

            <input type="text" class="form-control" name="name" placeholder="Enter Recipient Full Name"
required>

        </div>

        <div class="form-group">

            <label for="r-age">Enter Recipient Age</label>

            <input type="number" class="form-control" name="age" id="age" placeholder="Enter Recipient Age"
required>

        </div>

        <div class="form-group">

            <label for="email">Email address</label>

            <input type="email" class="form-control" name="email" placeholder="Enter Recipient's email address"
required>

        </div>

        <div class="form-group">

            <label for="phone">Phone No</label>

            <input type="tel" class="form-control" name="phone" placeholder="Enter 10 Digit Phone Number"
required>

        </div>

        <div class="form-group">

            <label for="b-type">Choose Blood type</label>

```

```

    <select id="blood-group" name="blood-type" class="formcontrol browser-default custom-select"
required>

    <option value="A Positive">A postive (A+)</option>

    <option value="A Negative">A Negative (A-)</option>                <option value="B Positive">B
postive (B+)</option>

    <option value="B Negative">B Negative (B-)</option>                <option value="O Positive">O
postive (O+)</option>

    <option value="O Negative">O Negative (O-)</option>

    <option value="AB Positive">AB postive (AB+)</option>

    <option value="AB Negative">AB Negative (AB-)</option>

</select>

</div>

<div class="form-group">

    <label for="locality">Locality</label>

    <input type="text" required class="form-control" name="locality" id="location" placeholder="Enter
Area Name (eg. City/town Name)">

</div>

<div class="form-group">

    <label for="postal-code">Postal Code</label>

    <input type="zip" required class="form-control" name="postal-code" placeholder="Enter 6 digit postal
code">

</div>

<div class="form-group">

    <label for="contact-address">                Enter you contact
address

    </label>

    <textarea class="form-control" required name="contactaddrss" id="address" cols="4"
rows="3"></textarea>

```

```

        </div>

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

    </form>

</div>

</main>

</body>

</html>

```

## Track.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>Track Request</title>

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
>

</head> <style>

main{

margin-top: 80px;

}

</style>

<body>

    <nav class="navbar navbar-dark navbar-expand-sm bg-danger fixed-top">

        <a href="/" class="navbar-brand">

            Plasma Donor App

        </a>

```

```

</nav>

<main>

    {% if session['track_id'] == False %}

    <div class="container d-flex justify-content-center">

        <div class="card mb-2 bg-light" style="width: 30rem;">

            <div class="card-header">

                Track Your Request

            </div>

            <div class="card-body">

                <form action="{ {url_for('track_request')} }" method="post">

                    <div class="form-group">

                        <label for="tracking-id">Enter Your Request ID / Tracking ID :</label>

                        <input type="text" value="{ {req_id} }" placeholder="uZFMliQJtywmytv" class="form-control"
name="tracking-id">

                    </div>

                    <button type="submit" class="btn btnprimary">Track</button>

                </form>

            </div>

        </div>

    </div>

</main>

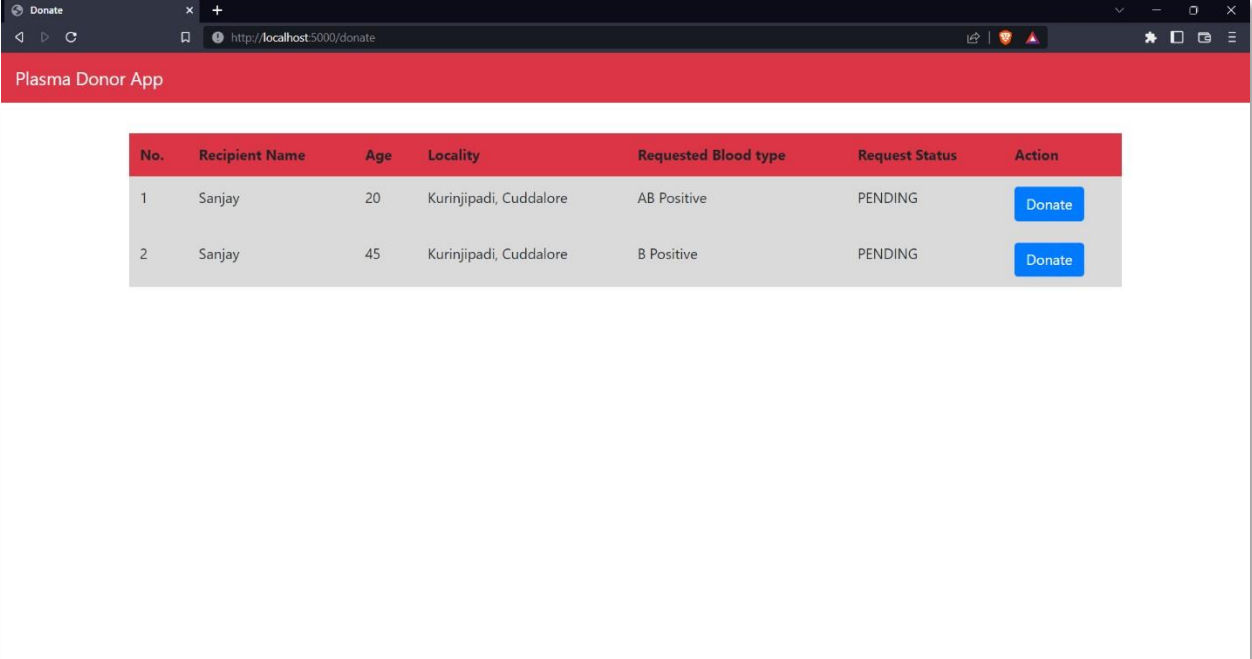
</body>

</html>

```

## 7.2 Make Plasma Donation

The donors have to click the donate button in the donate section of their account which will contains all the plasma request made by the users. Then, the donor will be asked to fill up the donation form to proceed with booking an donation appointment.



No.	Recipient Name	Age	Locality	Requested Blood type	Request Status	Action
1	Sanjay	20	Kurinjiipadi, Cuddalore	AB Positive	PENDING	<a href="#">Donate</a>
2	Sanjay	45	Kurinjiipadi, Cuddalore	B Positive	PENDING	<a href="#">Donate</a>

Fig.7.2.1

The screenshot shows a web browser window with the title 'Book Appointment'. The address bar shows the URL 'http://localhost:5000/BookAppointment/uzFMIIQJtywmbuOb'. The page has a red header with the text 'Plasma Donor App'. The main content is a 'Donation Form' with the following fields and elements:

- Donor's Name:** A text input field containing 'John Doe'.
- Donor's Age:** A text input field with a warning message: 'You must be atleast 18 years old to donate Plasma'.
- Mobile Number:** A text input field with a warning message: 'Enter 10 digit phone number'.
- Choose Blood type:** A dropdown menu showing 'A positive (A+)'. Below it is a checkbox labeled 'Does the requested blood type match your blood type ?'.
- What is your current Medical Status ?** A text input field containing 'Healthy'.
- Enter the Location:** A text input field with the placeholder 'enter the location of medical center near you'.
- Choose the date and time:** A date and time picker showing 'dd-mm-yyyy --:-- --'.
- Enter your contact address:** A large text area for the address.
- Footer:** A checkbox labeled 'I have read the [Guide](#) completely and i'm aware that i am eligible to make this donation'.
- Submit:** A blue button labeled 'Submit'.

Fig.7.2.2

## CODE:

### donateForm.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>Book Appointment</title>

  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
```

```

>
</head> <style>    main{
margin-top: 60px;
    }
</style>
<body>
    <header>
        <nav class="navbar navbar-dark navbar-expand-sm bg-danger fixed-top">
            <a href="/" class="navbar-brand">
                <i class="fas fa-medkit"></i> &nbsp;
                Plasma Donor App
            </a>
        </nav>
    </header>
    <main>
        <div class="container-sm bg-light">
            <h4 class="text-center">
                Donation Form
            </h4>
            <form action="{ {url_for('make_donation')}} " method="post">
                <div class="form-group">
                    <label for="RequestID" hidden>
                        RequestID :
                    </label>
                    <input type="text" class="form-control" name="req_id" required value="{ {req_id}} " hidden>
                </div>
                <div class="form-group">
                    <label for="donor-name">Donor's Name</label>

```

```

        <input type="text" name="donor-name" id="donor-name" class="form-control" required
placeholder="John Doe">

    </div>

    <div class="form-group">

        <label for="age">Donor's Age</label>

        <input type="number" min="18" max="65" name="donor-age"
placeholder="You must be atleast 18 years old to donate Plasma" class="form-
control" required>

    </div>

    <div class="form-group">

        <label for="mobile">Mobile Number</label>

        <input type="text" name="phone-number" class="form-control" placeholder="Enter 10 digit phone
number" required>

    </div>

    <div class="form-group">

        <label for="b-type">Choose Blood type</label>

        <select id="blood-group" name="blood-type" class="formcontrol browser-default custom-select"
required>

            <option value="A Positive">A postive (A+)</option>

            <option value="A Negative">A Negative (A-)</option> <option value="B
Positive">B postive (B+)</option>

            <option value="B Negative">B Negative (B-)</option>

            <option value="O Positive">O postive (O+)</option> <option value="O
Negative">O Negative (O-)</option>

            <option value="AB Positive">AB postive (AB+)</option>

            <option value="AB Negative">AB Negative (AB-)</option>

        </select>

    </div>

    <div class="form-group">

        <input type="checkbox" required name="btype-match">

```



```

        <label for="btype-cnf">Does the requested blood type match your blood type ?</label>

</div>

<div class="form-group">

    <label for="med-status">What is your current Medical Status ?</label>

    <input type="text" name="medical-status" class="formcontrol" placeholder="Healthy" required>

</div><div class="form-group"><label for="location">Enter the Location</label><input type="text"
name="location" class="form-control" placeholder="enter the location of medical center near you">

</div>

<div class="form-group">

    <label for="date">Choose the date and time </label>

    <input type="datetime-local" class="form-control" name="datetime" required>

</div><div class="form-group"><label for="address">Enter your contact address</label><textarea
class="form-control" name="contact-address" cols="6" rows="4"></textarea></div><div class="form-group">

    <input type="checkbox" required>

    <label for="eligibility-check">I have read the <a href="/guide">Guide</a> completely and i'm aware that i am
    eligible to make this donation </label>

</div> <button type="submit" class="btn btn-primary mb-4
floatright">Submit</button></form></div></main>

</body></html>

```

## 7.3 SendGrid API Integration

### 7.3.1.SendGrid

SendGrid is a cloud-based SMTP provider that allows you to send email without having to maintain email servers. SendGrid manages all of the technical details, from scaling the infrastructure to ISP outreach and reputation monitoring to whitelist services and real time analytics. SendGrid provides two ways to send email: through our SMTP relay or through our Web API. SendGrid provides client libraries in many languages. This is the preferred way to integrate with SendGrid. If you choose to use SendGrid without a client library, the Web API is recommended in most cases as it is faster, provides some benefit with encoding, and tends to be easier to use.

## 7.3.2. Implementation

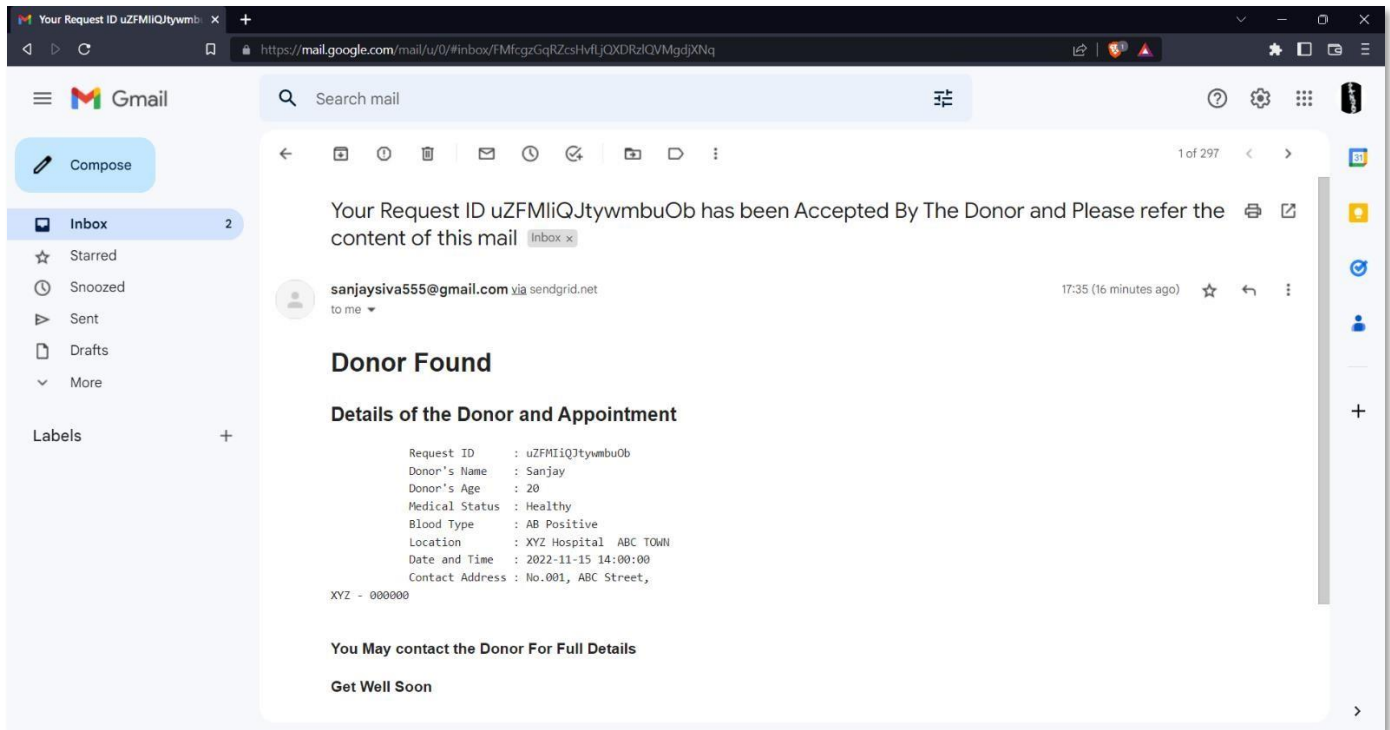


Fig.7.3.2

## 7.3.3.CODE

Mailer.py

```
from sendgrid import SendGridAPIClient from
sendgrid.helpers.mail
import Mail from dotenv import load_dotenv import os
load_dotenv()
def send_the_email(to_email,subject,html_content):
message = Mail(from_email='sanjaysiva555@gmail.com',
to_emails=to_email,subject=subject,
html_content=html_content)
try:
    sg = SendGridAPIClient(os.environ.get('SENDGRID_API_KEY'))
response = sg.send(message)
print(response.status_code)
print(response.body)
print(response.headers)
```

```
return except Exception as e:
print(e.message) return
```

# 7.4 Database Schema

For Database, IBM Cloud DB2 instance is used which is really efficient and more reliable.

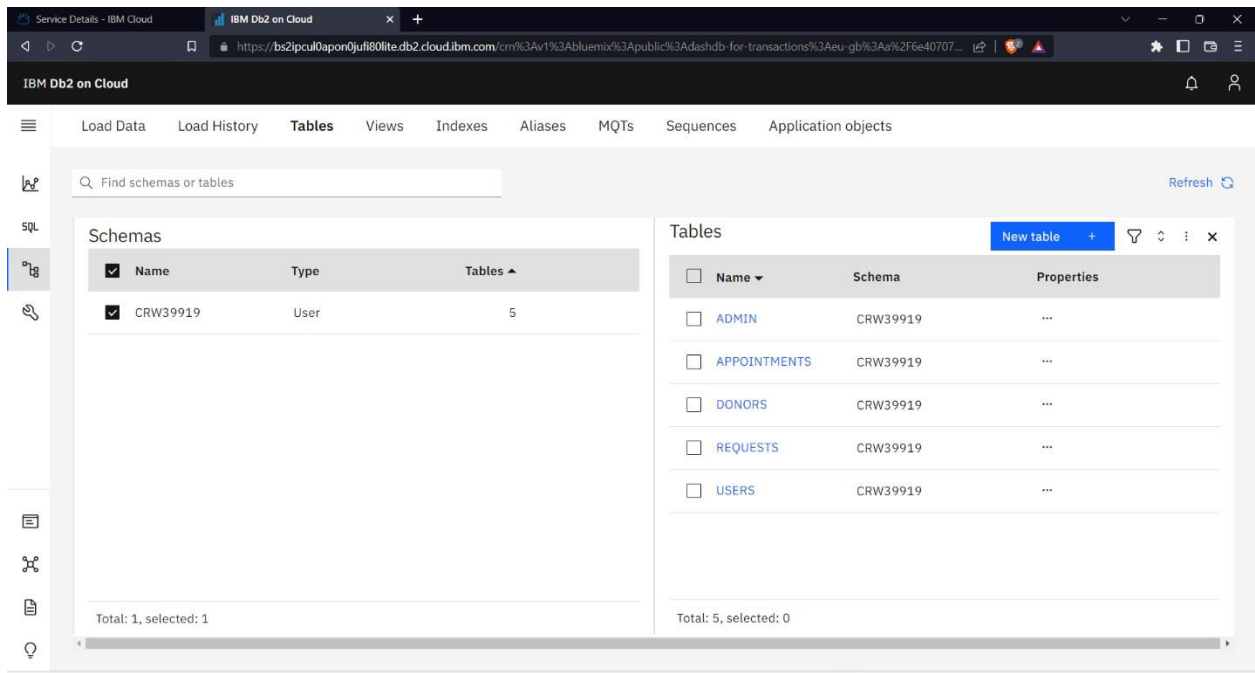


Fig.7.4.1

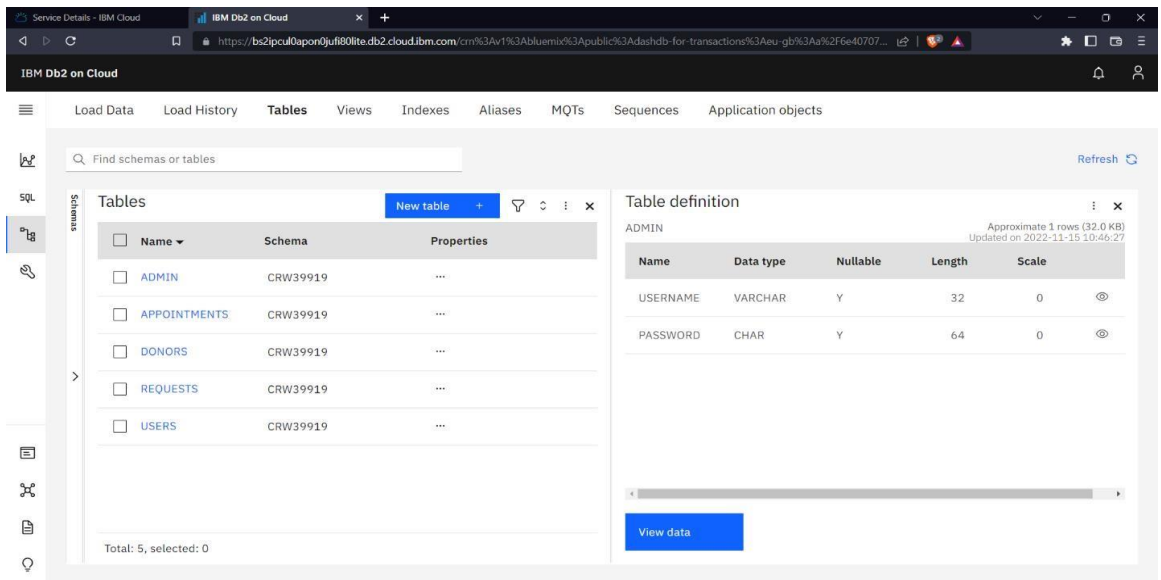


Fig.7.4.2

The screenshot shows the IBM Db2 on Cloud web interface. The 'Tables' tab is active, displaying a list of tables in the 'CRW39919' schema: ADMIN, APPOINTMENTS, DONORS, REQUESTS, and USERS. The 'APPOINTMENTS' table is selected, and its definition is shown on the right. The table has 8 columns: REQUEST\_ID, DONOR\_NAME, DONOR\_AGE, BLOOD\_TYPE, MEDICAL\_STATUS, LOCATION, DATE\_TIME, and PHONE\_NUMBER. The 'CONTACT\_ADDRESS' column is also listed but appears to be a separate field or a truncated entry.

Name	Data type	Nullable	Length	Scale
REQUEST_ID	CHAR	Y	16	0
DONOR_NAME	CHAR	Y	32	0
DONOR_AGE	VARCHAR	Y	32	0
BLOOD_TYPE	CHAR	Y	64	0
MEDICAL_STATUS	VARCHAR	Y	255	0
LOCATION	VARCHAR	Y	255	0
DATE_TIME	TIMESTAMP	Y	12	10
PHONE_NUMBER	VARCHAR	Y	64	0
CONTACT_ADDRESS	VARCHAR	Y	255	0

Fig.7.4.3

The screenshot shows the IBM Db2 on Cloud web interface. The 'Tables' tab is active, displaying a list of tables in the 'CRW39919' schema: ADMIN, APPOINTMENTS, DONORS, REQUESTS, and USERS. The 'DONORS' table is selected, and its definition is shown on the right. The table has 14 columns: FIRSTNAME, LASTNAME, USER\_EMAIL, ADDRESS1, ADDRESS2, STATE, PINCODE, DOB, GENDER, PHONE, COVID\_STATUS, BLOOD\_TYPE, and PASS\_WORD.

Name	Data type	Nullable	Length	Scale
FIRSTNAME	CHAR	Y	16	0
LASTNAME	CHAR	Y	16	0
USER_EMAIL	VARCHAR	Y	32	0
ADDRESS1	VARCHAR	Y	32	0
ADDRESS2	VARCHAR	Y	32	0
STATE	CHAR	Y	16	0
PINCODE	DECIMAL	Y	6	0
DOB	DATE	Y	4	0
GENDER	CHAR	Y	12	0
PHONE	VARCHAR	Y	32	0
COVID_STATUS	VARCHAR	Y	50	0
BLOOD_TYPE	VARCHAR	Y	64	0
PASS_WORD	CHAR	Y	64	0

Fig.7.4.4

The screenshot shows the IBM Db2 on Cloud web interface. The 'Tables' tab is selected, displaying a list of tables: ADMIN, APPOINTMENTS, DONORS, REQUESTS, and USERS, all under the schema CRW39919. The 'REQUESTS' table is selected, and its definition is shown on the right. The table has 2 rows (32.0 KB) and was updated on 2022-11-13 17:36:34.

Name	Data type	Nullable	Length	Scale
REQUEST_ID	CHAR	Y	16	0
REQUEST_STATUS	VARCHAR	Y	32	0
RECIPIENT_NAME	CHAR	Y	32	0
RECIPIENT_AGE	VARCHAR	Y	32	0
RECIPIENT_EMAIL	CHAR	Y	32	0
RECIPIENT_PHONE	VARCHAR	Y	32	0
REQUESTED_BLOOD_TYPE	CHAR	Y	50	0
LOCALITY	VARCHAR	Y	50	0
POSTAL_CODE	CHAR	Y	6	0
RECIPIENT_ADDRESS	LONG VARCHAR	Y	32700	0

Buttons: 'New table', 'View data'.

Fig.7.4.5

The screenshot shows the IBM Db2 on Cloud web interface. The 'Tables' tab is selected, displaying a list of tables: ADMIN, APPOINTMENTS, DONORS, REQUESTS, and USERS, all under the schema CRW39919. The 'USERS' table is selected, and its definition is shown on the right. The table has 2 rows (32.0 KB) and was updated on 2022-11-04 16:11:00.

Name	Data type	Nullable	Length	Scale
FULLNAME	CHAR	Y	16	0
USER_DOB	DATE	Y	4	0
PHONE_NO	VARCHAR	Y	32	0
EMAIL	CHAR	Y	32	0
PASSWORD	CHAR	Y	64	0

Buttons: 'New table', 'View data'.

Fig.7.4.6

## 8. TESTING

### 8.1 Test Cases 8.2 User Acceptance Testing

#### 1. *Purpose of Document*

The purpose of this document is to briefly explain the test coverage and open issues of the **Plasma Donor App** 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	5	4	2	3	15
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	1	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	73

#### 3. *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	35	0	4	31
Security	2	0	0	2
Outsource Shipping	5	0	0	5
Exception Reporting	6	0	0	6
Final Report Output	4	0	0	4
Version Control	1	0	0	1

## 9. RESULTS

### 9.1 Performance Metrics

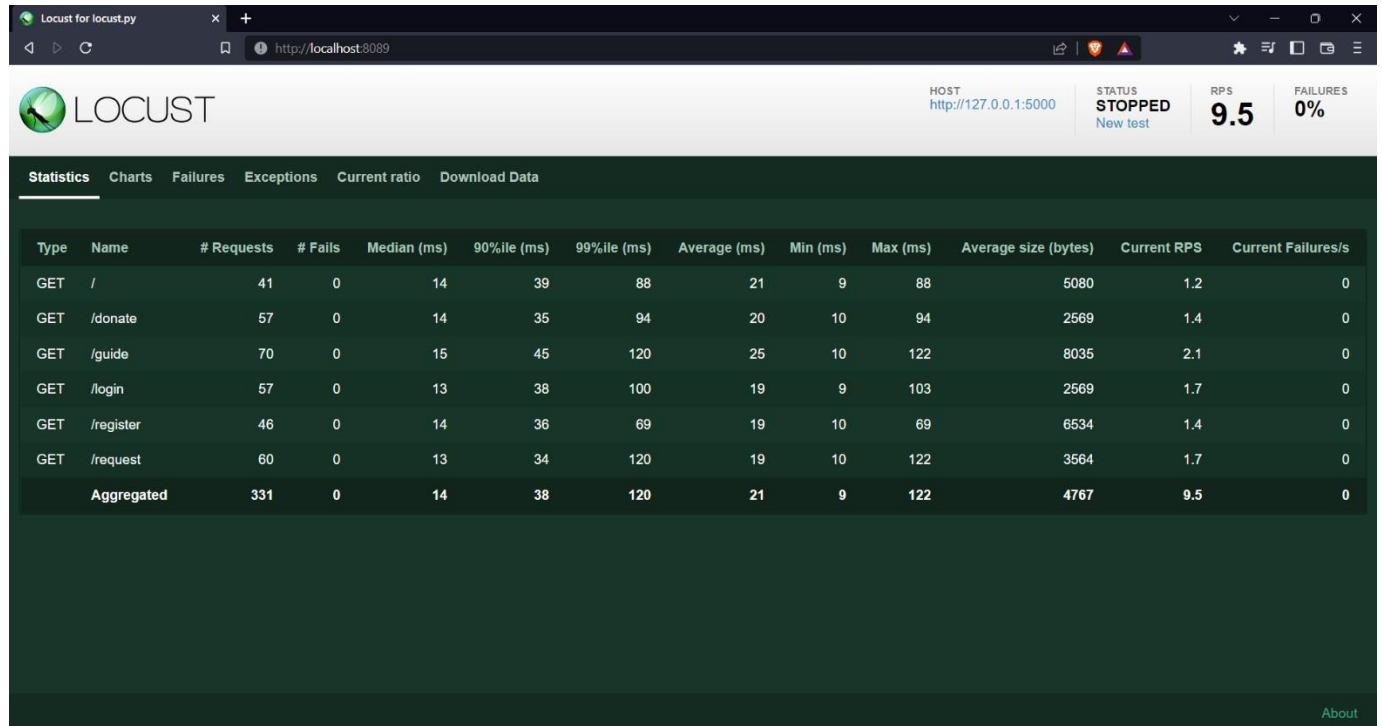


Fig.9.1.1

Type	Name	Request C	Failure C	Median Re	Average R	Min Respo	Max Respo	Average C	Requests/s	Failures/s	50%	66%	75%	80%	90%	95%	98%	99%	99.90%	99.99%	100%
GET	/	41	0	14	20.55202	9.3601	87.703	5080	1.171159	0	14	17	17	22	39	64	88	88	88	88	88
GET	/donate	57	0	14	19.90231	10.2592	93.6444	2569	1.628196	0	14	17	20	24	35	61	80	94	94	94	94
GET	/guide	70	0	15	25.06131	10.4198	121.592	8035	1.999539	0	15	20	22	28	45	120	120	120	120	120	120
GET	/login	57	0	13	19.49542	9.3547	102.9061	2569	1.628196	0	13	16	18	25	38	40	93	100	100	100	100
GET	/register	46	0	14	18.5614	9.7119	68.9916	6534	1.313983	0	14	16	19	22	36	43	69	69	69	69	69
GET	/request	60	0	13	18.72855	10.1287	121.9202	3564	1.713891	0	13	17	21	22	34	43	46	120	120	120	120
Aggregated		331	0	14	20.60463	9.3547	121.9202	4767.372	9.454964	0	14	17	21	23	38	46	98	120	120	120	120

Fig.9.2.1



## **10. ADVANTAGES AND DISADVANTAGES**

### **Advantages • User-Friendly Interface**

The users can able to find its very easy to use and they can also smooth experience while using the application.

- **Cross-Platform Compatibility**

Since the application is purely web-based, the user can able to access the application from any kind of device. Hence it provides cross-platform compatibility for the users.

- **Speed**

The application is completely light-weight and can able to response much faster and provides user with real-time experience.

- **Maintenance**

The application need only be installed on the server placing minimum requirements on the end user's workstation. Because of that the maintaining and updating the system is much simpler.

- **Scalability**

Since the application is developed using the micro-services architecture which provides vertical scaling the application can able to grow and shrink on its own based on the traffic

## **Disadvantages**

- **Internet Reliance**

The application requires an uninterrupted internet connection to access the web application.

- **Self-verification**

The application cannot have the capability of distinguish between the fake user and genuine user on its own. It demands the admin work to getting things done.

## **11. CONCLUSION**

Although the government is carrying out Covid vaccination campaigns on a large scale, the number of vaccines produced is not enough for all the population to get vaccinated at present. And with the corona positive cases rising every day, saving lives has become the prime matter of concern. As per the data provided by WHO more than 3 million people have died due to the coronavirus. However, apart from vaccination, there is another scientific method by which a covid infected person can be treated and the death risk can be reduced. A person who has recovered from Covid can donate his/her plasma to a person who is infected with the coronavirus. 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. Both parties can Accept or Reject the request. User has to Upload a Covid Negative report to be able to Donate Plasma. This system is used if anyone needs a Plasma Donor Blood and Plasma donation is a kind of citizen's social responsibility in which an individual

can willingly donate blood/plasma via our app. This Application has been created with the concept and has sought to make sure that the donor gives plasma to community. This model is made user friendly so anybody can view and maintain his/her account. This application will break the chain of business through blood/plasma and help the poor to find donor at free of cost.

## **12. FUTURE SCOPE**

Plasma Application can be developed to further improve user accessibility via integrating this application with various social networks application program interfaces (APIs). Consequently, users can login and sign up using various social networks. This would increase number of donors and enhances the process of blood donation.

User interface (UI) can be improved in future to accommodate global audience by supporting different languages across countries. Data scraping can be done from different social networks and can be shown in the Blood/Plasma Request Feeds. Appointments can be synchronized with Google and Outlook calendars for the ease of users.

Donor and Beneficiary Stories feature aims to create a sense of belonging to the community. Donors will be able to view and share personal experiences about their donation; Beneficiaries can share their experiences of receiving blood transfusion which contributed to their improved health and lives.

Live Check-in Process feature aims to provide a better experience with regards to the waiting time when the user is in the process of donation. We hypothesis that a more efficient experience will help the user look forward to his blood/plasma donation appointments.

## **14. APPENDIX**

Source Code (GitHub) link: <https://github.com/IBM-EPBL/IBM-Project-14960-1659592732>