

IT - ITes SSC
NASSCOM

NALAYATHIRAN PROJECT UNDER NAAN MUDHAVAN SCHEME
BY THE INITIATIVE OF THE
GOVERNMENT OF TAMILNADU

SUBMITTED BY

COLLEGE NAME : JEPPIAAR ENGINEERING COLLEGE

ID : PNT2022TMID26969

PROJECT DOMAIN : CLOUD APPLICATION DEVELOPMENT

PROJECT NAME : PLASMA DONOR APPLICATION

TEAM LEADER : PAVITHRA R (310819104058)

TEAM MEMBERS : SHALINI R (310819104078)

SWATHI K (310819104088)

SUWETHA B (310819104087)

Under The Guidance Of

SPOC : Mr. GOUDHAMAN MARIMUTHU

Industrial Mentor : Mrs/Ms. NAVYA

Faculty Mentor : Mrs. VIDHYA A

Faculty Evaluator : Mrs. JEEVITHA D

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. APPENDIX Source Code

GitHub & Project Demo Link

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 COVID- positive 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

With rapid increase in the usage of social networks sites across the world, there is also a steady increase in blood donation requests as being noticed in the number of posts on these sites such as Facebook and twitter seeking blood donors. Finding blood donor is a challenging issue in almost every country. There are some blood donor finder applications in the market such as Blood app by Red Cross and Blood Donor Finder application by Neologix. However, more reliable applications that meet the needs of users are prompted.

The simplest way to do this was to ask donors what they expect from the app. Good relations with the client allowed

us to reach the potential users, among whom we spread the information that we want to conduct workshops with blood donors on their needs regarding the potential app. Those previous apps haven't fulfilled the utmost need.

2.2 REFERENCES

- Eder A, et al. Selection criteria to protect the blood donor in North America and Europe: past (dogma), present (evidence), and future (hemovigilance). *Transfusion Medicine Reviews*. 2009;23(3):205–220. [PubMed]
- Moreno J. “Creeping precautionism” and the blood supply. *Transfusion*. 2003;43:840–842. [PubMed]
- Farrugia A. The mantra of blood safety: time for a new tune? *Vox Sanguinis*. 2004;86:1–7. [PubMed]
- Fifty-Eighth World Health Assembly. Resolution WHA58.13: Blood safety: proposal to establish World Blood Donor Day. Geneva: World Health Organization; 2005. [17 August 2012].
- The Melbourne Declaration on 100% voluntary non-remunerated donation of blood and blood components. Geneva: World Health Organization; 2009. [17 August 2012]. http://www.who.int/worldblooddonorday/Melbourne_Declaration_VNRBD_2009.pdf.
- WHO/CDC/IFRC. Blood donor counselling: Implementation guidelines. Geneva: World Health Organization; 2012. [17 August 2012]. http://www.who.int/bloodsafety/voluntary_donation/blood_donor_selection_counselling/en/ [PubMed]
- Screening donated blood for transfusion-transmissible infections. Geneva: World Health Organization; 2010. [17 August 2012].
- Global Database on Blood Safety. Summary report 2011. Geneva: World Health Organization; 2011. [22 August 2012].
- Boulton F. Evidence-based criteria for the care and selection of blood donors, with some comments on the relationship to blood supply and emphasis on the management of donation-induced iron depletion. *Transfusion Medicine*. 2008;18:13–27. [PubMed]

2.3 PROBLEM STATEMENT DEFINITION

YEAR	TITLE	AUTHOR(s)	TECHNIQUE(s)	PROS	CONS
2021	BDoor App- Blood Donation Application using Android Studio	S Periyannayagi, A Manikandan, M Muthukrishnan, and M Ramakrishnan	Android, Flutter UI, Dart, Firebase, Decision tree algorithm	The Donor details are verified before they allow donation and have to be authorized by the institution. The Verification and validations are done in the email base.	The android mobile user will not be able to insert or view details if the server goes down. Thus, there is a disadvantage of single point failure.

2020	Lifesaver EBlood 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.
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 is required to be verified by the blood bank. It is a user-friendly application. It will help people to find plasma easily.	This system is closed for general plasma donation and mainly focused on COVID-19 patients for plasma donation
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 a 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

2. IDEATION & PROPOSED SOLUTION

2.1 EMPATHY MAP LINK

https://drive.google.com/file/d/1pOIIgzvfQZSi1QTH6oKLDzQ9cEcFtYiL/view?usp=share_link

2.2 IDEATION AND BRAINSTORMING

BRAINSTORMING

https://drive.google.com/file/d/1Px76j5jsAp3r3-8Hh_dCzkdAqbqH7_9a/view?usp=share_link

PROBLEM SOLUTION FIT

https://drive.google.com/file/d/1R1t8iAxrVJcQmKZXgXE6vyqDbpijRolC/view?usp=share_link

2.3 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	<p>This application can able to perform certain functionality and possess certain feature which are unique. Those are listed below:</p> <ul style="list-style-type: none"> • Those who want to donate their plasma can do by simply register by uploading their covid-19 recovery certificate. • It can able to find donors who are located close to the needy by using GPS location tracking. • A chat-bot to answer frequently asked question about plasma donation.
<u>4</u>	Social Impact / Customer Satisfaction	<p>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.</p>
<u>5</u>	Business Model (Revenue Model)	<p>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.</p>
<u>6</u>	Scalability of the Solution	<p>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.</p>

3. REQUIREMENT ANALYSIS

3.1 FUNCTIONAL REQUIREMENTS

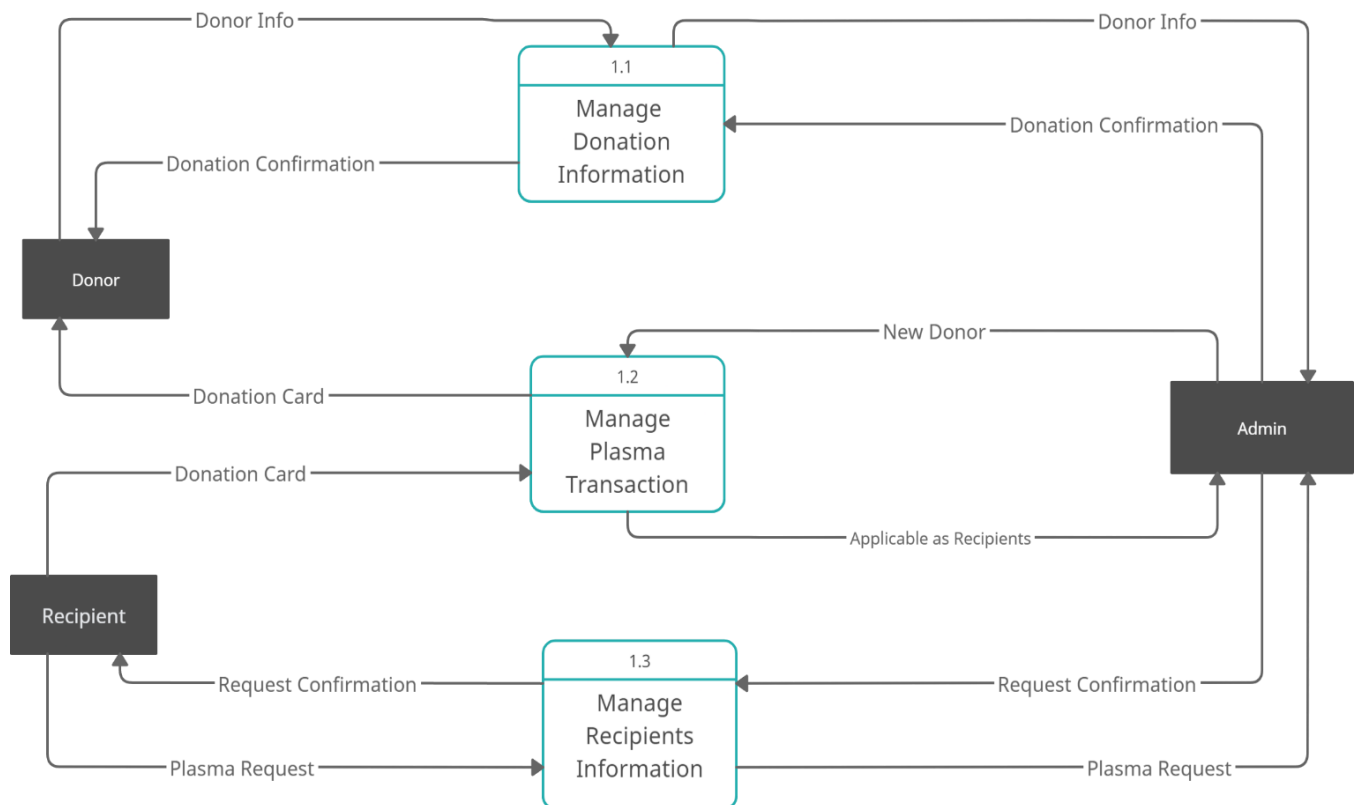
NFR No.	Non-Functional Requirement	Description
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Creating Donor Profile	Volunteer Donor able to create their donor profile by providing their medical information 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.

3.2 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.

4. PROJECT DESIGN

4.1 DATA FLOW DIAGRAM



4.1 SOLUTION AND TECHNICAL ARCHITECTURE

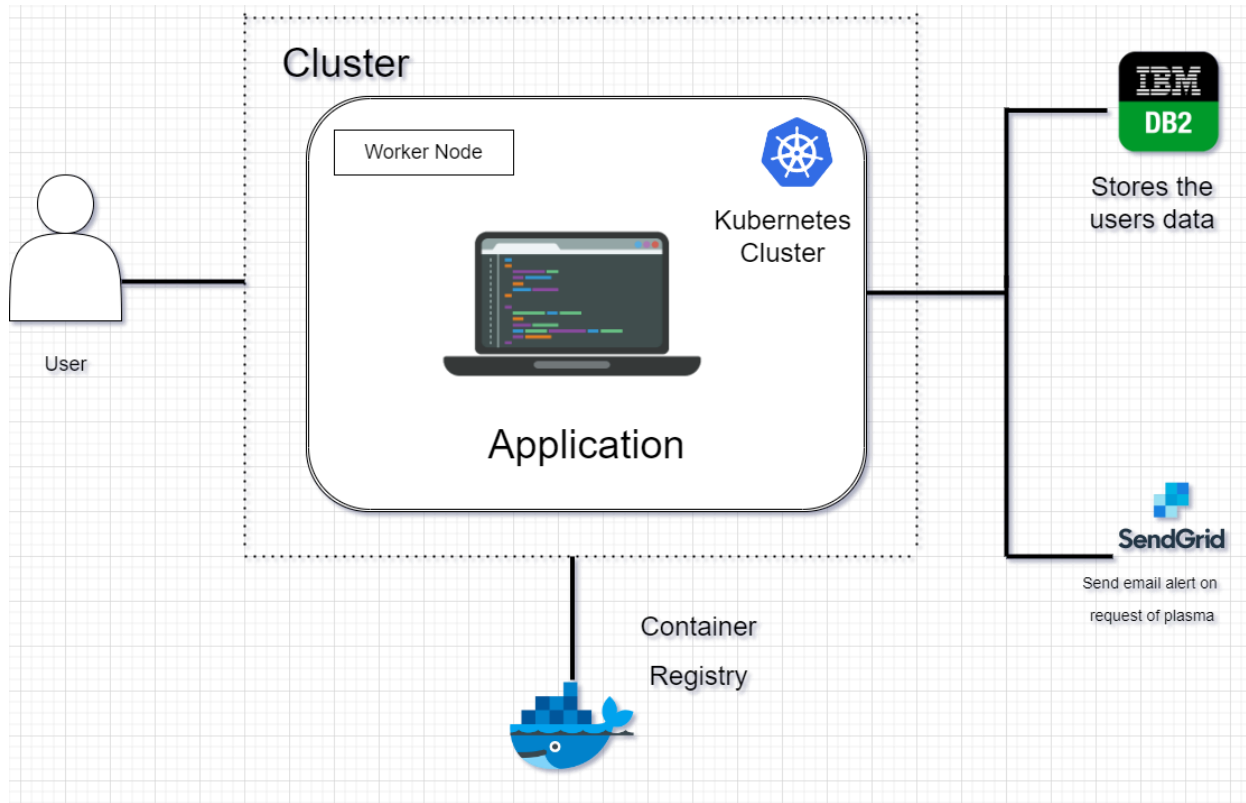


Fig.5.2. Solution and Technical Architecture

4.2 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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	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
Admini strator	Verificatio n 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

5. PROJECT PLANNING AND SCHEDULING

5.1 SPRINT PLANNING AND ESTIMATION

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 my email, password, and confirming my password.	5	High	Pavithra R Shalini R
Sprint- 1	Email verification	USN-2	As a user, I will receive confirmation email once I have registered for the application	3	High	Swathi K , Suwetha B

Sprint- 1		USN-3	As a user, I can register for the application through Google	2	Medium	Pavithra R
Sprint- 1	Login	USN-4	As a user, I can log into the application by entering email & password	5	High	Shalini R
Sprint- 1	Donor Profile	USN-5	As a user, I can able to register me as a verified donor for plasma.	3	High	Swathi K, Suwetha B
Sprint- 2	Dashboard / Statistics	USN-6	As a user, I can able to see the statistics of plasma availability, donor count on my dashboard.	5	High	Pavithra R, Suwetha B
Sprint- 2	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	Swathi K Shalini R
Sprint- 2	Plasma Request	USN-8	As a user, I can able to make a plasma request in the application	5	High	Pavithra R, Shalini R, Suwetha B
Sprint- 2	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	Pavithra R, Shalini R
Sprint- 2		USN-10	As a user, I will upload my COVID-19 recovery certification in order to become a verified registered donor	5	High	Pavithra R, Suwetha B, Swathi K

Sprint- 3	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	Shalini R, Suwetha B
Sprint- 3	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	Pavithra R, Swathi K
Sprint- 3	Communication Channel	USN-13	As a user, I can able to communicate with the donor personally within the application	3	Medium	Swathi K, Shalini R

Sprint- 3		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	Swathi K, Shalini R
Sprint- 3	Administrator	USN-15	As an admin, I will store the registered donor details after verification into the database and maintain it periodically	5	High	Pavithra R , Suwetha B
Sprint- 4	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	Pavithra R , Swathi K
Sprint- 4		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	Swathi K, Shalini R , Suwetha B
Sprint- 4	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	Suwetha B
Sprint- 4	Administrator	USN-19	As an admin, I will approve all the plasma transaction in the application after the proper verification	5	High	Pavithra R , Shalini R, Swathi K
Sprint- 4		USN-20	As an admin, I will update the plasma availability and donors count periodically	3	Medium	Swathi K , Pavithra R

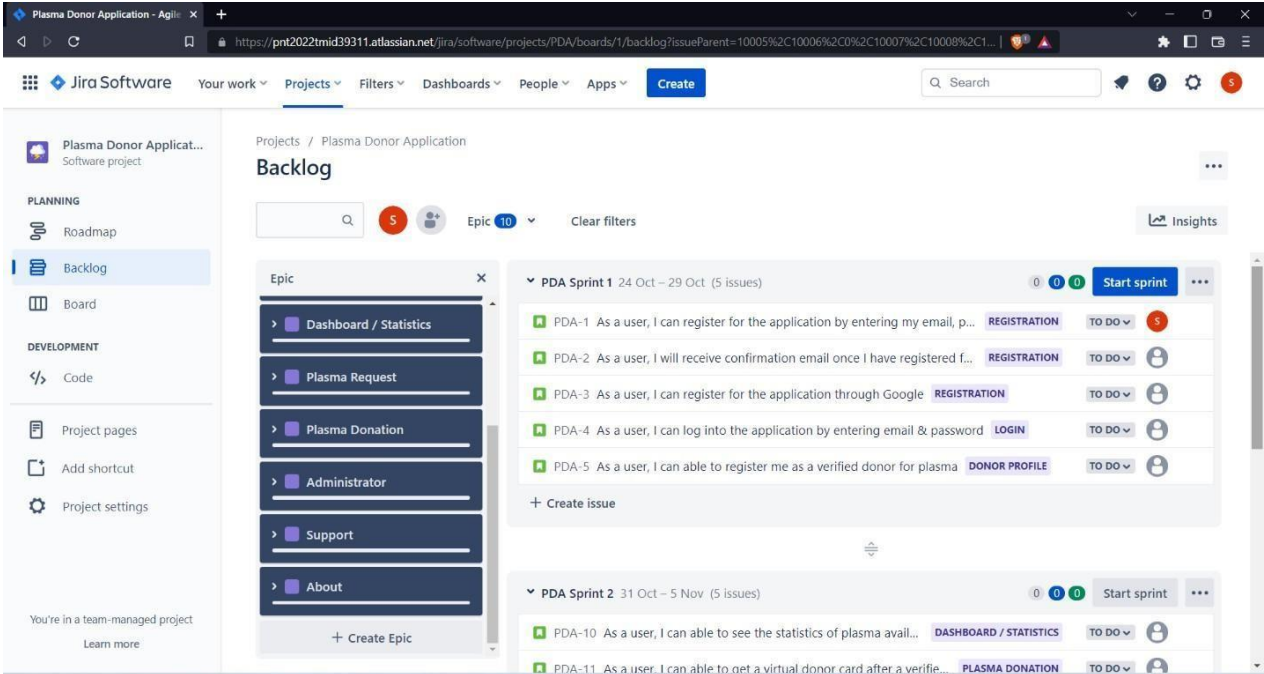
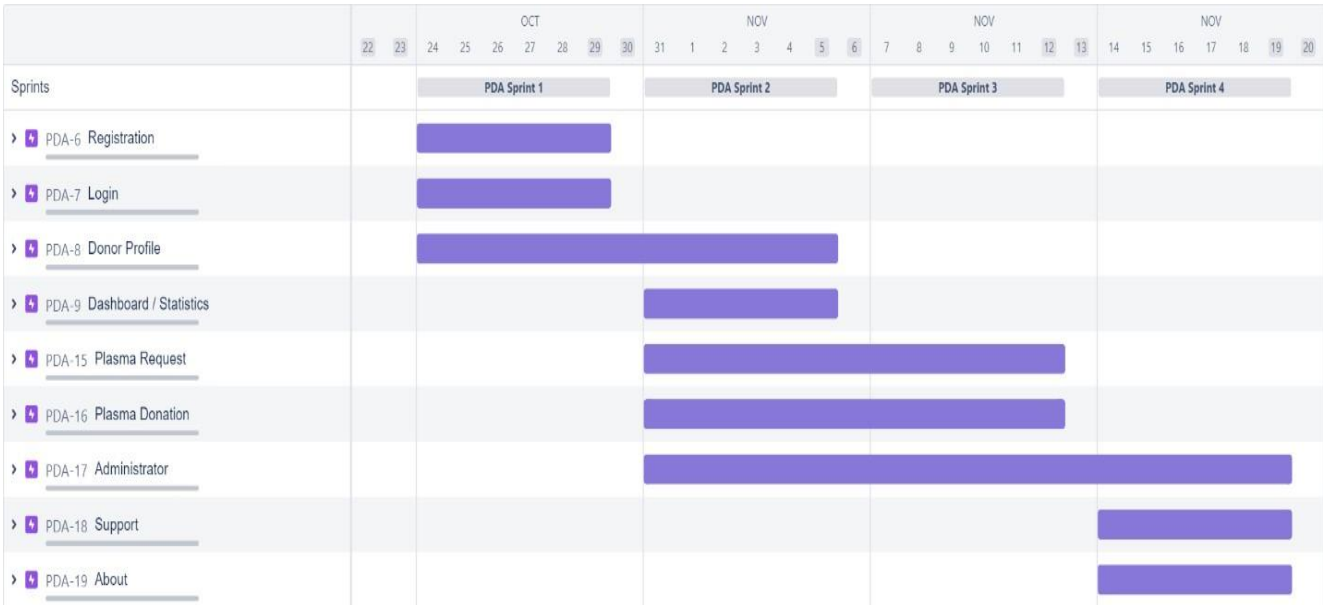
5.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)	Average Velocity (AV)= $\frac{\text{Sprint duration}}{\text{velocity}}$
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

5.3

REPORTS FROM JIRA



Plasma Donor Application - Agile

https://pnt2022tmid39311.atlassian.net/jira/software/projects/PDA/boards/1/backlog?issueParent=10005%2C10006%2C0%2C10007%2C10008%2C1...

Jira Software

Your work

Projects

Filters

Dashboards

People

Apps

Create

Search

Plasma Donor Applicat...

Software project

PLANNING

Roadmap

Backlog

Board

DEVELOPMENT

Code

Project pages

Add shortcut

Project settings

You're in a team-managed project

Learn more

Projects / Plasma Donor Application

Backlog

Clear filters

Insights

Epic 10

PDA-11 As a user, I can able to get a virtual donor card after a verifie... PLASMA DONATION TO DO

PDA-12 As a user, I can able to make a plasma request in the application PLASMA REQUEST TO DO

PDA-13 As an admin, I will verify the details provided by the donors so ... ADMINISTRATOR TO DO

PDA-14 As a user, I will upload my COVID-19 recovery certification in or... DONOR PROFILE TO DO

Create issue

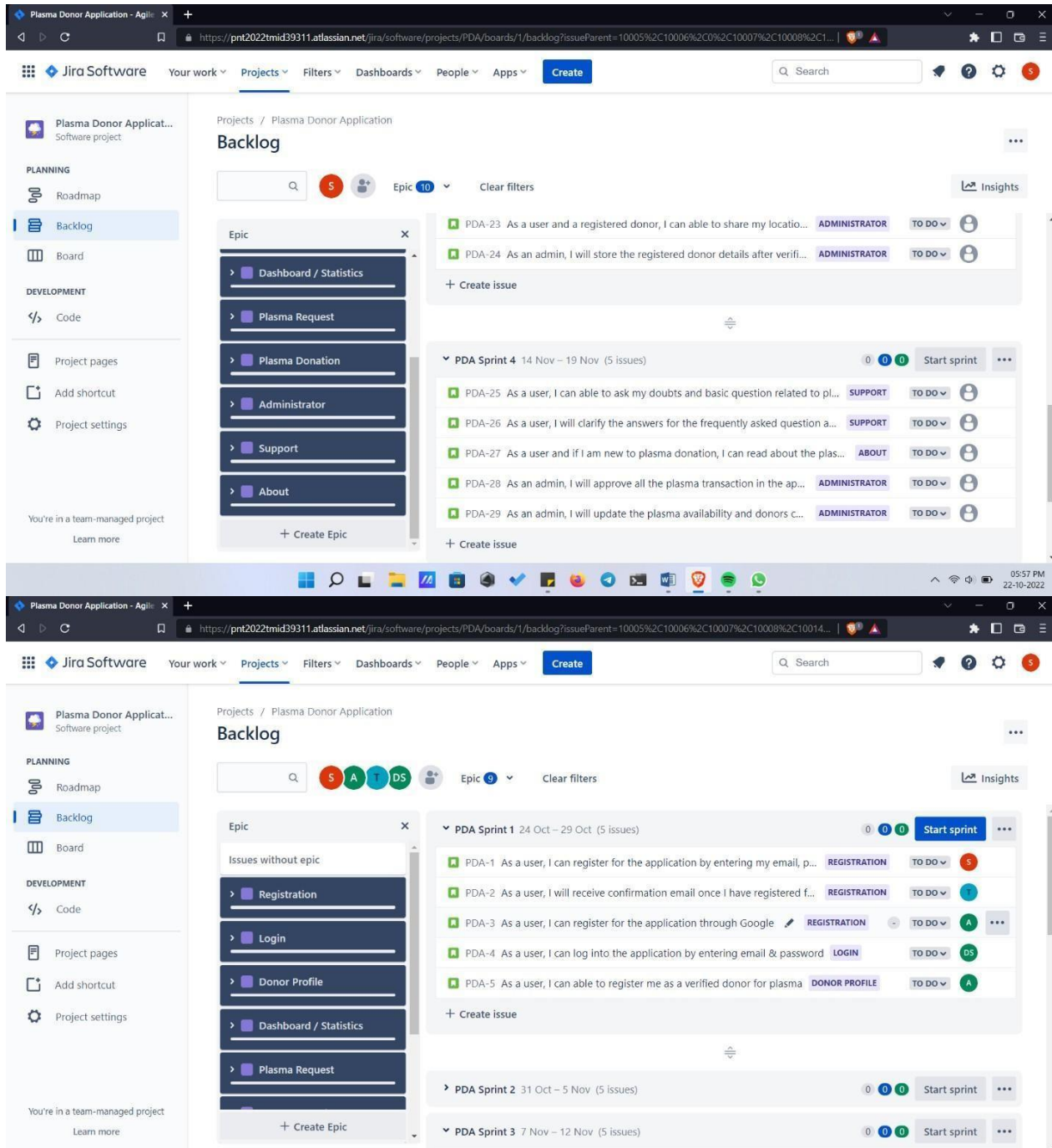
PDA Sprint 3 7 Nov - 12 Nov (5 issues)

PDA-20 As a user and a registered donor, I will accept the plasma re... PLASMA DONATION TO DO

PDA-21 As a user and a volunteer donor, I can able to book an appoi... PLASMA DONATION TO DO

PDA-22 As a user, I can able to communicate with the donor personall... PLASMA REQUEST TO DO

PDA-23 As a user and a registered donor, I can able to share my locatio... ADMINISTRATOR TO DO



6. CODING AND SOLUTIONING

6.1 Request for Plasma Web

Framework

A Web Application Framework or simply a Web Framework represents a collection of libraries and modules that enable web application developers to write applications without worrying about lowlevel 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

Request

127.0.0.1:5000/request

Gmail YouTube Maps tech mahindra hr in...

Plasma Donor App

Request Form

Recipient Name

Enter Recipient Age

Email address

Phone No

Choose Blood type

Locality

Postal Code

Book Appointment

Not secure | 192.168.1.6:5000/BookAppointment/r7zNbaxT5s0V78zF

Gmail YouTube Maps tech mahindra hr in...

Plasma Donor App

Donation Form

Donor's Name

Donor's Age

Mobile Number

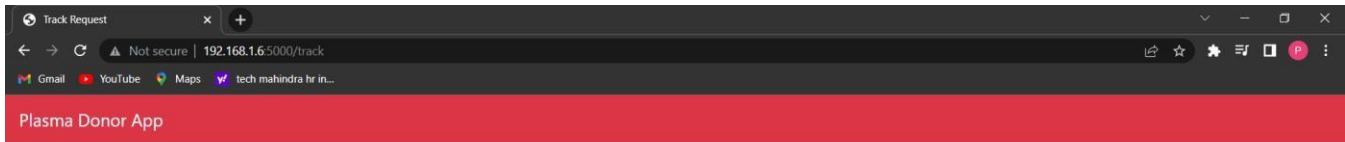
Choose Blood type

☐ Does the requested blood type match your blood type ?

What is your current Medical Status ?

Enter the Location

Choose the date and time



Track Your Request

Enter Your Request ID / Tracking ID :

Track

CODE:

Account.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>Account</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-top: 80px;
  }
</style>
<body>
  <header>
    <div class="header navbar-wrapper">
      <nav class="navbar navbar-dark navbar-expand-sm bg-danger fixed-top">
        <div class="container">
          <a href="/" class="navbar-brand">
            <i class="fas fa-medkit"></i> &nbsp;
            Plasma Donor App
          </a>
        </div>
      </nav>
    </div>
  </header>
</body>
</html>
```

```
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarCollapse">
<span class="navbar-toggler-icon"></span>
</button>
<div id="navbarCollapse" class="collapse navbar-collapse">
<ul class="navbar-nav ml-auto">
  <li class="nav-item">
    <a href="/home" class="nav-link ">
      Home
    </a>
  </li>
  <li class="nav-item ">
    <a href="/donate" class="nav-link btn btn-secondary ">
      Donate
    </a>
  </li>
  <li class="nav-item">
    <a href="/request" class="nav-link ">
      Request
    </a>
  </li>
  <li class="nav-item">
    <a href="/about" class="mr-3 nav-link ">
      Help
    </a>
  </li>
  <li class="nav-item">
    <a href="/account" class="nav-link active">
      Account
    </a>
  </li>
  <li class="nav-item">
    <a href="/logout" class="ml-4 mr-2 btn btn-dark">
      Logout
    </a>
  </li>
</ul>
</div>
</div>
</header>
<main>
  <div class="container">
    <div class="card bg-primary">
      <div class="card-header">
        Account Details
      </div>
      <div class="card-body bg-light">
        <div class="card mb-2">
```

```

<h6 class="card-header">FULL NAME</h6>
<div class="card-body">
  <p class="card-text">
    {% if session['account-type'] == 'Donor' %}
    {{ res['FIRSTNAME']+' '+res['LASTNAME'] }}
    {% else %}
    {{ res['FULLNAME'] }}
    {% endif %}

  </p>
</div>
</div>
<div class="card mb-2">
  <h6 class="card-header">DATE OF BIRTH (YYYY-MM-DD):</h6>
  <div class="card-body">
    <p class="card-text">
      {% if session['account-type'] == 'Donor' %}
      {{ res['DOB'] }}
      {% else %}
      {{ res['USER_DOB'] }}
      {% endif %}
    </p>
  </div>
</div>
<div class="card mb-2">
  <h6 class="card-header">PHONE NUMBER:</h6>
  <div class="card-body">
    <p class="card-text">+91
      {% if session['account-type'] == 'Donor' %}
      {{ res['PHONE'] }}
      {% else %}
      {{ res['PHONE_NO'] }}
      {% endif %}
    </p>
  </div>
</div>
<div class="card mb-2">
  <h6 class="card-header">EMAIL:</h6>
  <div class="card-body">
    <p class="card-text">
      {% if session['account-type'] == 'Donor' %}
      {{ res['USER_EMAIL'] }}
      {% else %}
      {{ res['EMAIL'] }}
      {% endif %}
    </p>
  </div>
</div>

```



```

    {% if session['account-type'] == 'Donor' % }
    <div class="card mb-2">
        <h6 class="card-header">COVID STATUS:</h6>
        <div class="card-body">
            <p class="card-text">
                {{ res['COVID_STATUS'] }}
            </p>
        </div>
    </div>
    <div class="card mb-2">
        <h6 class="card-header">BLOOD TYPE:</h6>
        <div class="card-body">
            <p class="card-text">
                {{ res['BLOOD_TYPE'] }}
            </p>
        </div>
    </div>
    <div class="card mb-2">
        <h6 class="card-header">PINCODE</h6>
        <div class="card-body">
            <p class="card-text">
                {{ res['PINCODE'] }}
            </p>
        </div>
    </div>
    <div class="card mb-2">
        <h6 class="card-header">STATE</h6>
        <div class="card-body">
            <p class="card-text">
                {{ res['STATE'] }}
            </p>
        </div>
    </div>
    {% endif % }
    <div class="card mb-2">
        <h6 class="card-header">Is this account Donor or Not Donor?</h6>
    <div class="card-body">
        <p class="card-text">
            {% if session['account-type'] == 'Donor': % }
            <span class="badge badge-success">Donor</span>
        {% else % }
            <span class="badge badge-danger">Not Donor</span></p>
        {% endif % }
    </div>
    </div>
    <!-- Button trigger modal -->
    <button type="button" class="btn btn-danger float-right" data-toggle="modal" data-target="#exampleModal">
Delete Account

```



```

    }
</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="uZFMliQJtywmbytv" class="form-control"
name="tracking-id">
                        </div>
                        <button type="submit" class="btn btn-primary">Track</button>
                    </form>
                </div>
            </div>
        </div>
        {% endif % }

        <div class="container">
            {% if session['track_id'] == True % }
            <h5 class="text-center">Track the Status of Any Request</h5>

            <div class="row">
                <div class="col bg-light mr-2">
                    <h6 class="mt-1">Request ID</h6>
                </div>
                <div class="col bg-secondary">
                    <p class="text-white mt-1">
                        {{ res['REQUEST_ID'] }}
                    </p>
                </div>
            </div>
            <div class="row">
                <div class="col bg-light mr-2">
                    <h6 class="mt-1">Request STATUS</h6>
                </div>
            </div>
        </div>
    </main>
</body>
</html>

```

```

<div class="col bg-secondary">
  {% if res['REQUEST_STATUS'] == 'PENDING': %}
  <p class="text-white badge badge-danger mt-1">
    {{res['REQUEST_STATUS']}}
  </p>
  {% else %}
  <p class="text-white badge badge-primary mt-1">
    {{res['REQUEST_STATUS']}}
  </p>
  {% endif %}
</div>
</div>
<div class="row">
  <div class="col bg-light mr-2">
    <h6 class="mt-1">Recipient Name</h6>
  </div>
  <div class="col bg-secondary">
    <p class="text-white mt-1">
      {{res['NAME']}}
    </p>
  </div>
</div>
<div class="row">
  <div class="col bg-light mr-2">
    <h6 class="mt-1">Recipient Age</h6>
  </div>
  <div class="col bg-secondary">
    <p class="text-white mt-1">
      {{res['AGE']}}
    </p>
  </div>
</div>
<div class="row">
  <div class="col bg-light mr-2">
    <h6 class="mt-1">Recipient Email</h6>
  </div>
  <div class="col bg-secondary">
    <p class="text-white mt-1">
      {{res['USER_EMAIL']}}
    </p>
  </div>
</div>
<div class="row">
  <div class="col bg-light mr-2">
    <h6 class="mt-1">Recipient Phone</h6>
  </div>
  <div class="col bg-secondary">
    <p class="text-white mt-1">

```

```

        +91 {{res['PHONE']}}
    </p>
</div>
</div>
<div class="row">
    <div class="col bg-light mr-2">
        <h6 class="mt-1">Requested Blood Type</h6>
    </div>
    <div class="col bg-secondary">
        <p class="text-white mt-1">
            {{res['REQUESTED_BLOOD_TYPE']}}
        </p>
    </div>
</div>
<div class="row">
    <div class="col bg-light mr-2">
        <h6 class="mt-1">Locality</h6>
    </div>
    <div class="col bg-secondary">
        <p class="text-white mt-1">
            {{res['LOCALITY']}}
        </p>
    </div>
</div>
<div class="row">
    <div class="col bg-light mr-2">
        <h6 class="mt-1">Postal Code</h6>
    </div>
    <div class="col bg-secondary">
        <p class="text-white mt-1">
            {{res['POSTAL_CODE']}}
        </p>
    </div>
</div>
<div class="row">
    <div class="col bg-light mr-2">
        <h6 class="mt-1">Contact Address</h6>
    </div>
    <div class="col bg-secondary">
        <p class="text-white mt-1">
            {{res['RECIPIENT_ADDRESS']}}
        </p>
    </div>
</div>
<!-- <a href="/cancel"><button type="submit" class="btn btn-danger float-right mt-5">Cancel
Request</button></a> -->
{% if res['REQUEST_STATUS'] != "PENDING" : %}
<div class="container">

```

```

<div class="card">
  <div class="card-body bg-info mt-10">
    <h3 class="card-text">Please Check your Mail for Any Appointment by the Donor</h3>
  </div>
</div>
</div>
{% endif % }
{% endif % }

</div>

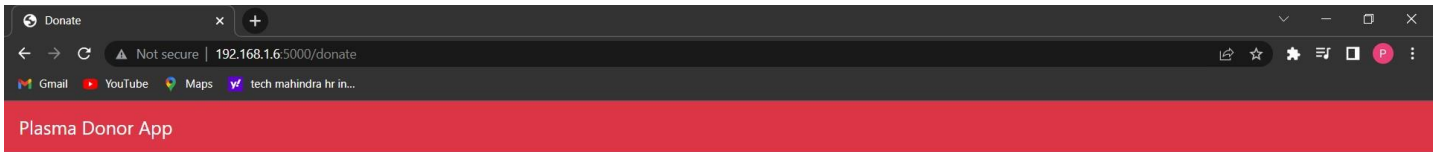
</main>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-
9/reFTGAW83EW2RDu2S0VKalzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN" crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV" crossorigin="anonymous"></script>
</body>
</html>

```

6.2 Make Plasma Donation

The donors have to click the donate button in the donate section of their account which will contains all the plasma requests made by the users.

Then, the donor will be asked to fill up the donation form to proceed with booking an donation appointment.



6.3 SendGrid API Integration

6.3.1. 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.

6.3.2 IMPLEMENTATION

mail.google.com/mail/u/0/?tab=rm&ogbl=inbox/FMfcgzGrbHmWznzQcwngTPZzLRRikB

1 of 4,693

1:35 PM (8 minutes ago)

Suwethabv@gmail.com via sendgrid.net to Suwethabv

<h1>Donor Found</h1>
<h2>Details of the Donor and Appointment</h2>
<body>
<pre>
Request ID : O7QCv6rHcTGLOGpJ
Donor's Name : Suwetha
Donor's Age : 20
Medical Status : Healthy
Blood Type : O Negative
Location : balaji medical college and hospital, chromepet
Date and Time : 2022-11-30 13:37:00
Contact Address : chennai
</pre>
<h3>You May contact the Donor For Full Details</h3>
<h3>Get Well Soon</h3>
</body>

13:43 20-11-2022

6.3.2. CODE

sendmail.py

```
import configparser

import ssl
ssl._create_default_https_context = ssl._create_unverified_context
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail

config=configparser.ConfigParser()

config.read("config.ini")

def sendMailUsingSendGrid(API, from_email, to_email, subject,html_content):
if API!=None and from_email!=None and len(to_email)>0:      message=
Mail(from_email, to_email, subject,html_content)      try:
    sg = SendGridAPIClient(API)
response = sg.send(message)
print(response.status_code)
print(response.body)
print(response.headers)      except
Exception as e:
print(e.message)
```

6.4 Database Schema

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

IBM Db2 on Cloud

bs2ipcul0apon0juf80lite.db2.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aeu-gb%3Aa%2Fec9630d6bab4d23b7e39fa2e2b028cb%3A8b1eed72-0df...

IBM Db2 on Cloud

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

Find schemas or tables Refresh

Schemas

<input checked="" type="checkbox"/>	Name	Type	Tables
<input checked="" type="checkbox"/>	BTS01747	User	4

Total: 1, selected: 1

Tables

<input type="checkbox"/>	Name	Schema	Properties
<input type="checkbox"/>	APPOINTMENTS	BTS01747	...
<input type="checkbox"/>	DONORS	BTS01747	...
<input type="checkbox"/>	REQUESTS	BTS01747	...
<input type="checkbox"/>	USERS	BTS01747	...

Total: 4, selected: 0

IBM Db2 on Cloud

bs2ipcul0apon0juf80lite.db2.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aeu-gb%3Aa%2Fec9630d6bab4d23b7e39fa2e2b028cb%3A8b1eed72-0df...

IBM Db2 on Cloud

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

Find schemas or tables Refresh

Schemas

Tables

<input type="checkbox"/>	Name	Schema	Properties
<input type="checkbox"/>	APPOINTMENTS	BTS01747	...
<input type="checkbox"/>	DONORS	BTS01747	...
<input type="checkbox"/>	REQUESTS	BTS01747	...
<input type="checkbox"/>	USERS	BTS01747	...

Total: 4, selected: 0

Table definition

APPOINTMENTS

Approximate 16 rows (32.0 KB)
Updated on 2022-11-21 08:26:38

Name	Data type	Nullable	Length	Scale
REQUEST_ID	VARCHAR	N	16	0
DONOR_NAME	VARCHAR	N	200	0
DONOR_AGE	INTEGER	N		0
BLOOD_TYPE	VARCHAR	N	200	0
MEDICAL_STATUS	VARCHAR	N	200	0
LOCATION	VARCHAR	N	200	0

View data

IBM Db2 on Cloud

bs2ipcul0apon0jufi80lite.db2.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aeu-gb%3Aa%2Fec9630d6bab4d23b7e39fa2e2b028cb%3A8b1eed72-0df...

IBM Db2 on Cloud

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

Find schemas or tables Refresh

Schemas

Tables

New table +

Table definition

DONORS

Approximate 7 rows (32.0 KB)
Updated on 2022-11-20 15:46:37

Name	Data type	Nullable	Length	Scale
FIRST_NAME	VARCHAR	N	88	0
LAST_NAME	VARCHAR	N	88	0
USER_EMAIL	VARCHAR	N	88	0
ADDRSS1	VARCHAR	N	88	0
ADDRSS2	VARCHAR	N	88	0
STATE	VARCHAR	N	88	0

View data

Total: 4, selected: 0

IBM Db2 on Cloud

bs2ipcul0apon0jufi80lite.db2.cloud.ibm.com/crm%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aeu-gb%3Aa%2Fec9630d6bab4d23b7e39fa2e2b028cb%3A8b1eed72-0df...

IBM Db2 on Cloud

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

Find schemas or tables Refresh

Schemas

Tables

New table +

Table definition

REQUESTS

Approximate 16 rows (32.0 KB)
Updated on 2022-11-20 17:51:40

Name	Data type	Nullable	Length	Scale
REQUEST_ID	VARCHAR	N	1000	0
REQUEST_STATUS	VARCHAR	N	1000	0
NAME	VARCHAR	N	1000	0
AGE	INTEGER	N		0
USER_EMAIL	VARCHAR	N	1000	0
PHONE	INTEGER	N		0

View data

Total: 4, selected: 0

7. TESTING

7.1 Test Cases

7.2 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

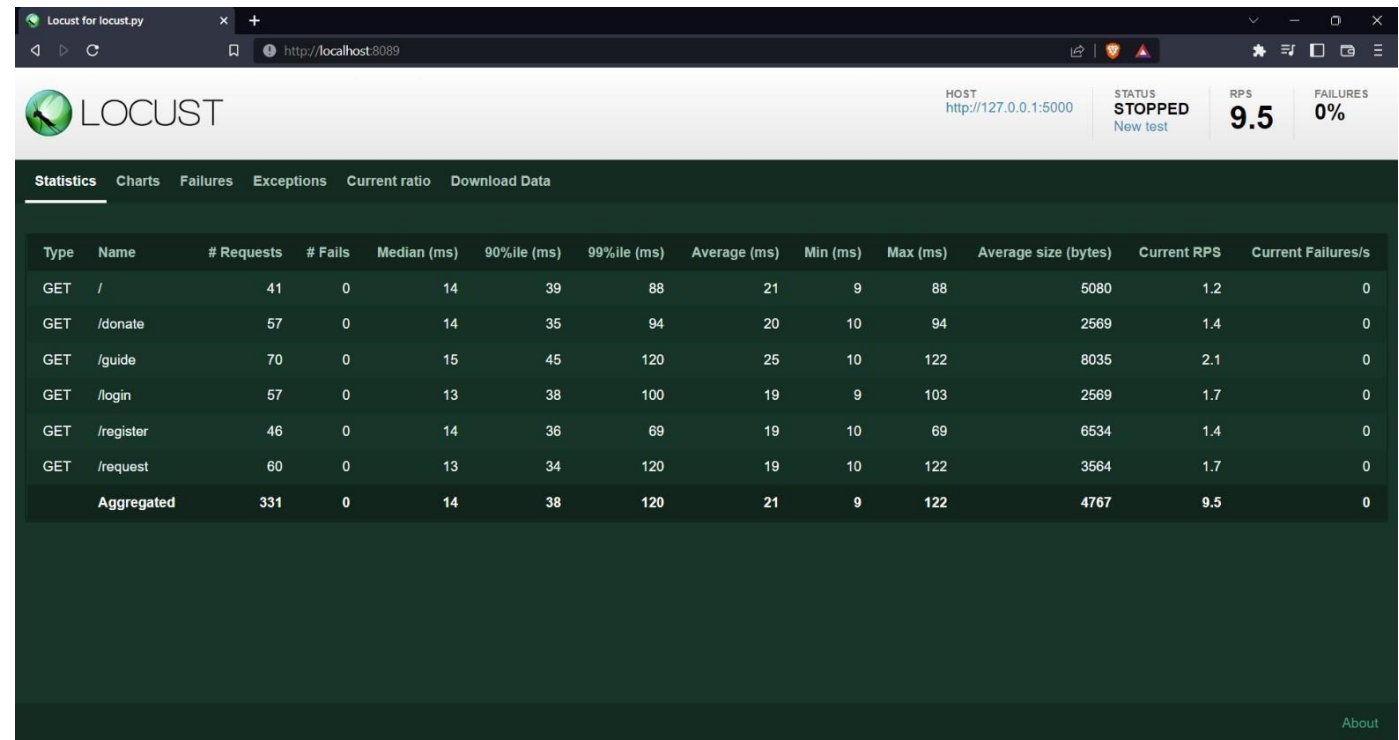
3. Test Case Analysis

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

P	Section	Total			F	ass
		Cases	Not Tested	ail		
	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

8. RESULTS

8.1 Performance Metrics



Type	Name	Request C	Failure Co	Median Re	Average R	Min Respo	Max Respc	Average C	Requests /	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
	Aggregate	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

9.

ADVANTAGES AND DISADVANTAGES

Advantages

Cross-Platform Compatibility

Since the application is purely web-based, the user is 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 is able to respond much faster and provides users with real-time experience.

User-Friendly Interface

The users are able to find it very easy to use and they can also have a smooth experience while using the application.

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

Maintenance

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

Disadvantages

Internet Reliance

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

10.

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.

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 the 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 donors at free of cost.

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 the number of donors and enhance the process of blood donation.

User interface (UI) can be improved in future to accommodate a global audience by supporting different languages across countries. Data scraping can be done from different social networks and

11.

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 hypothesize that a more efficient experience will help the user look forward to his blood/plasma donation appointments.

12. APPENDIX

- **Source Code link:** [Source code link](#)
- **GitHub Link:** [GitHub Link](#)
- **Project Demo link:** [Demo link](#)

