PLASMA DONOR APPLICATION

TEAM ID:PNT2022TMID05875

COLLEGE: SRI RAMAKRISHNA ENGINEERING COLLEGE

TEAM MEMBERS:

SUBHIKSHAA SHREE V N	1905144
SUBHIKSHA N	1905143
SIVARANJANI N	1905134
RAJSREE T	1905112

1.INTRODUCTION

1.1 PROJECT OVERVIEW

Plasma donation, also known as apheresis, can help save lives. It is a relatively safe procedure, but there can be minor side effects. Plasma is the liquid part of the blood. It contains proteins and antibodies that are crucial for clotting and immunity. Around 55% of the blood is plasma. Plasma donation involves drawing blood, extracting the plasma, and returning what is left of the blood to the person, all through a single needle that remains in the arm throughout the process. Plasma is in high demand, as it helps treat cancer and other health issues. In May 2020, the Food and Drug Administration (FDA) asked people who had recovered from COVID19 to donate plasma. Experts believe that the plasma may contain antibodies for SARS-CoV2, the virus behind the disease. Receiving plasma with these antibodies could help a person fight off the infection. People with AB blood have a universal type of plasma, which means that a person with any blood type can receive this plasma safely. This is different from having the universal blood type, which is O negative. The American Red Cross urge people with AB blood to donate plasma. A person can do every 28 days, or up to 13 times a year. Research shows that plasma donation is safe, and the National Institutes of Health (NIH) emphasize that there is no risk of getting the wrong blood back. Also, the FDA and other health authorities regulate the equipment and procedure of plasma donation. However, a person who donates plasma may experience minor adverse effects, and as with any other procedure involving a puncture, certain risks are involved. So, it is highly necessary and equally important to create an application to maintain the donors list and details to contact and track them during emergency situations. When talking about an application, it needs to be easy to handle and user friendly. To be more precise, this application should have all the facilities from registering to donating, and login to request satisfying. And one of the interesting facts is that this application runs on Cloud. This might be a useful application during critical times.

1.2 PURPOSE

When the world is struck by deadly diseases, there is a high risk of mass death of populations across the world. These diseases give no enough time for the surgeons to find medicine and so there is a need to find a quick remedy to reduce mass death of people to such illness. One of the best methods, which is highly effective is the donation of blood plasma of cured individuals to sick persons. This can possibly cure the illness of the infected person. Plasma donation was one of the best methods which was adopted to cure people during the recent global pandemic, COVID-19. The recovery rates were high during these times when death was ultimately increasing as no medicine was found across the globe. Plasma Donation also helps to increase immunity. Another issue was that no cured patient came forward to donate blood plasma, so the infected ones were highly worried as they can’t find anyone to help them. So, we are in need of an application that stores donor details, tracks and informs them upon request from a patient.

2.LITERATURE SURVEY

2.1 REALTIME SOFTWARE FOR EXISTING PROBLEMS

FLASK: There is a recent transformation into the development of multi-platform languages and frameworks. Flask is a small framework by most standards, small enough to be called a “microframework.” It is small enough that once you become familiar with it, you will likely be able to read and understand all of its source code. But being small does not mean that it does less than other frameworks. Flask was designed as an extensible framework from the ground up; it provides a solid core with the basic services, while extensions provide the rest. Because you can pick and choose the extension packages that you want, you end up with a lean stack that has no bloat and does exactly what you need. Flask has two main dependencies. The routing, debugging, and Web Server Gateway Interface (WSGI) subsystems come from Werkzeug, while template support is provided by Jinja2. Werkzeug and Jinja2 are authored by the core developer of Flask. There is no native support in Flask for accessing databases, validating web forms, authenticating users, or other high-level tasks. These and many other key services most web applications need are available through extensions that integrate with the core packages. As a developer, you have the power to cherry-pick the extensions that work best for your project or even write your own if you feel inclined to. This is in contrast with a larger framework, where most choices have been made for you and are hard or sometimes impossible to change.

DOCKER & KUBERNETES: Docker is an open-source engine that automates the deployment of applications into containers. It was written by the team at Docker, Inc (formerly dot Cloud Inc, an early player in the Platform-as-a-Service (PAAS) market), and released by them under the Apache 2.0 license. Docker adds an application deployment engine on top of a virtualized container execution environment. It is designed to provide a lightweight and fast environment in which to run your code as well as an efficient workflow to get that code from your laptop to your test environment and then into production. Docker is incredibly simple. Indeed, you can get started with Docker on a minimal host running nothing but a compatible Linux kernel and a Docker binary.

Kubernetes, or k8s for short, is an open-source container orchestrator. Originally developed by the engineers at Google, Kubernetes solves many problems involved with running a microservice architecture in production. Kubernetes automatically takes care of scaling, self-healing, load-balancing, rolling updates, and other tasks that used to be done manually by DevOps engineers. Since Kubernetes was open-sourced and managed by Cloud Native Computing Foundation in 2014, the development community has embraced its benefits to orchestrate container-based systems.

These Framework and Software help the application to gain a run-time structure and assist with all internal features of the app.

At times of pandemic, people should come forward to donate their blood plasma voluntarily to arrest the spread of disease and provide the cure that they have experienced. Our app will feature awareness videos and articles and will provide rewards for first time donors and regular donors. All These Humanitarian deeds are to be carried on regularly to maintain social well hood and supportive society. There are many incidents to be pointed out citing plasma donation during the past in Covid-19 Pandemic. One such incident where India's first Covid-19 plasma donor shares her story, urges other patients to do the same. Members of an Indian Islamic organisation are volunteering to donate blood for plasma therapy after their congregation sparked dozens of Covid-19 clusters across the country. All these articles are mentioned to encourage the donation of plasma in our application

2.2 REFERENCES

1) Philip J, Sarkar RS, Pathak A. Adverse events associated with apheresis procedures: Incidence and relative frequency. Asian J Transfus Sci. 2013 Jan;7(1):37-41. doi: 10.4103/0973-6247.106730. PMID: 23559763; PMCID: PMC3613659.

2) Flask Web Development by Miguel Grinberg Copyright © 2014 Miguel Grinberg. All rights reserved. Printed in the United States of America. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

3) Version v1.2.0 (fba92ef) of The Docker Book by James Turnbul © Copyright 2014 - James Turnbull. 4) An Introduction to Kubernetes by LEVEREGE , First Edition © Leverege LLC.

5) White Paper by IDC Sponsored by: IBM Andrew Smith, February 2021.

6) Essentials of Application Development on IBM Cloud December 2017 Third Edition (December 2017) by Ahmed Azraq Hala A. Aziz Uzma Siddiqui.

7) From Ahmedabad, Smriti Thakkar is the first recovered Covid-19 patient in India who volunteered to donate her plasma. India Today Web Desk New Delhi, April 25, 2020 UPDATED: April 25, 2020 19:15 IST

8) Tablighi Jamaat gives blood for plasma therapy By Zubair Ahmed BBC Hindi, Delhi Published 28 April 2020.

2.2 PROBLEM STATEMENT DEFINITION

A customer problem statement outlines problems that your customers face. It helps you figure out how your product or service will solve this problem for them. The statement helps you understand the experience you want to offer your customers. It can also help you understand a new audience when creating a new product or service. A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

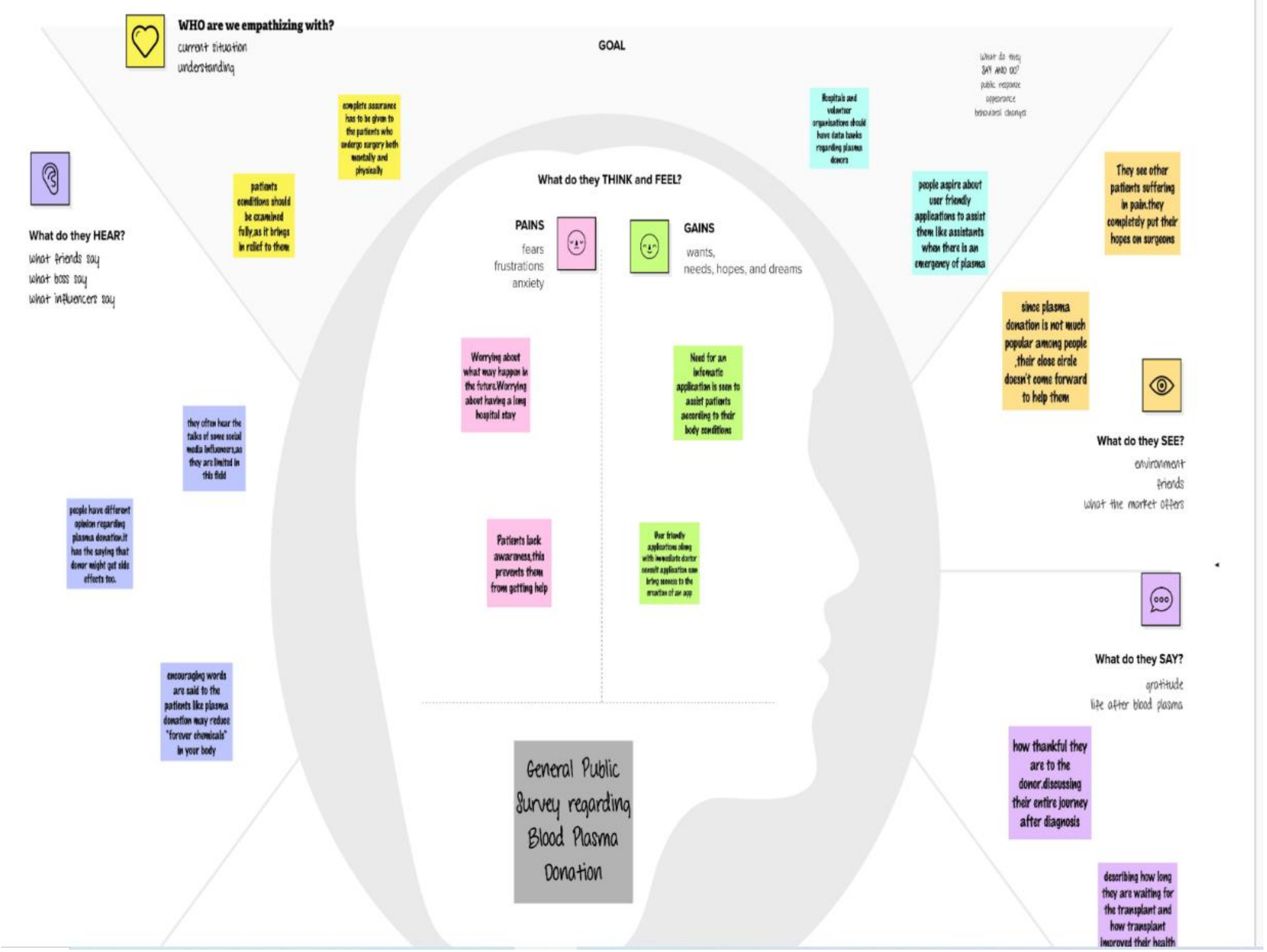
The most common problem faced by a person who is in need of blood plasma is reported below after a detailed analysis,

"As a patient in need of blood plasma, I am trying to contact a donor to help me with blood plasma, but I cannot find one easily as it takes long time and hardly possible, because the donors are usually far away from me and not ready to help me, which makes me feel hopeless and annoyed"

This is the most common problem statement derived after long surveys from the general public.

3.IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



complete assurance has to be given to the patients who undergo surgery both mentally and physically

patients conditions should be examined fully,as it brings in relief to them

Hospitals and volunteer organisations should have data banks regarding plasma donors

people aspire about user friendly applications to assist them like assistants when there is an emergency of plasma

They see other patients suffering in pain.they completely put their hopes on surgeons

since plasma donation is not much popular among people ,their close circle doesn't come forward to help them

Worrying about what may happen in the future.Worrying about having a long hospital stay

Need for an informatic application is seen to assist patients according to their body conditions

Patients lack awareness,this prevents them from getting help

User friendly applications along with immediate doctor consult application can bring success to the creation of an app

they often hear the talks of some social media influencers,as they are limited in this field

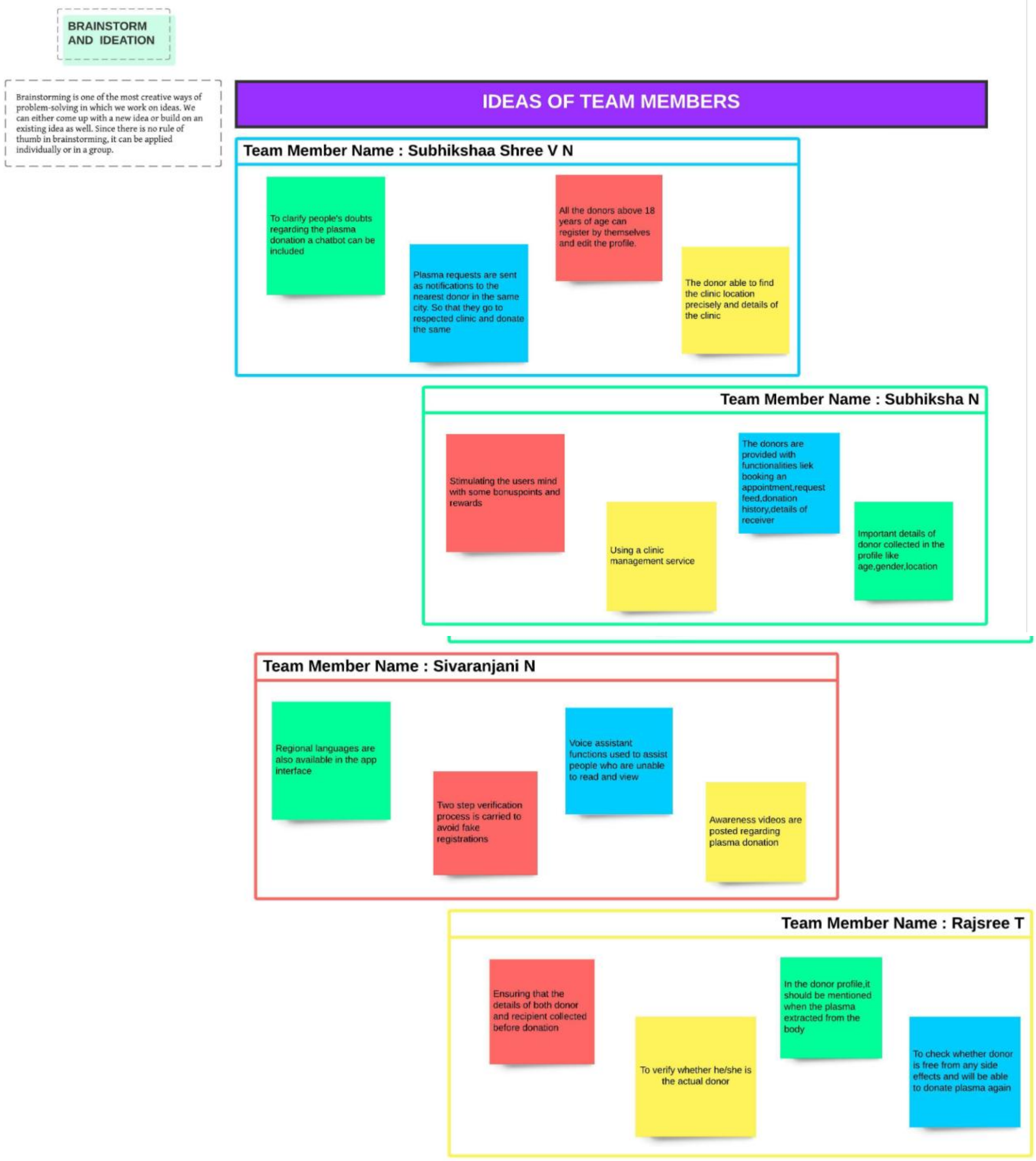
people have different opinion regarding plasma donation.it has the saying that donor might get side effects too.

encouraging words are said to the patients like plasma donation may reduce "forever chemicals" in your body

how thankful they are to the donor.discussing their entire journey after diagnosis

describing how long they are waiting for the transplant and how transplant improved their health

3.2 IDEATION & BRAINSTORMING



BRAINSTORM AND IDEATION

Brainstorming is one of the most creative ways of problem-solving in which we work on ideas. We can either come up with a new idea or build on an existing idea as well. Since there is no rule of thumb in brainstorming, it can be applied individually or in a group.

IDEAS OF TEAM MEMBERS

Team Member Name : Subhikshaa Shree V N

To clarify people's doubts regarding the plasma donation a chatbot can be included

Plasma requests are sent as notifications to the nearest donor in the same city. So that they go to respected clinic and donate the same

All the donors above 18 years of age can register by themselves and edit the profile.

The donor able to find the clinic location precisely and details of the clinic

Team Member Name : Subhiksha N

Stimulating the users mind with some bonuspoints and rewards

Using a clinic management service

The donors are provided with functionalities liek booking an appointment,request feed,donation history,details of receiver

Important details of donor collected in the profile like age,gender,location

Team Member Name : Sivaranjani N

Regional languages are also available in the app interface

Two step verification process is carried to avoid fake registrations

Voice assistant functions used to assist people who are unable to read and view

Awareness videos are posted regarding plasma donation

Team Member Name : Rajsree T

Ensuring that the details of both donor and recipient collected before donation

To verify whether he/she is the actual donor

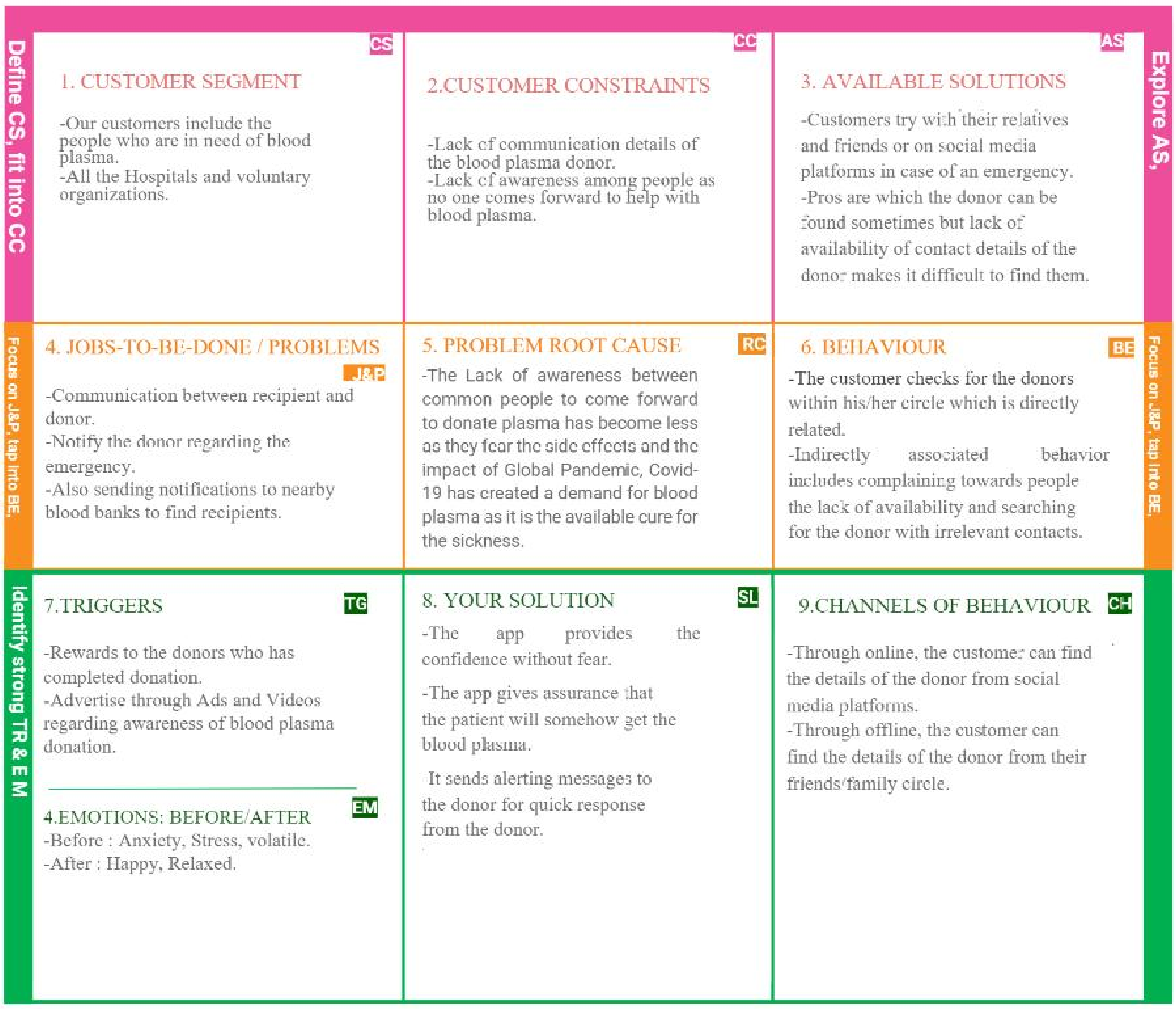
In the donor profile,it should be mentioned when the plasma extracted from the body

To check whether donor is free from any side effects and will be able to donate plasma again

3.3 PROPOSED SOLUTION

S.No.	**Parameter**	**Description**
1.	Problem Statement	People who need plasma are gaining weight every day. Plasma is necessary to help our body recover from injury, distribute nutrients, remove waste, and prevent infection as it moves through our circulatory system. They don't want to donate plasma but have no idea where to donate. We are designing a platform that contains all information about plasma donation.
2.	Solution description	Ours is a mobile application which aims to serve as a communication tool among plasma donation organizers and plasma donors. To emerge as a member of our system, donors want to create their profile through offering their statistics like name, blood group, e-mail address,phone number, password and actual region from 'Google Map', that are incorporated with this utility. This mobile app usually maintain updating the region of the donor.
3.	Uniqueness	Users can send feedback if they encountered difficulties during the donation process. This app will automatically continue to show nearby plasma donors. The donor saves the donor card digitally.
4.	Social Impact	This app will revolutionize the medical system by allowing people to donate plasma and serve humanity. You can also help people learn about the benefits of plasma donation, so your small contribution can help save someone's life.
5.	Business Model	There are many private sectors and NGOs that organize plasma donation camps. Even in partnership with companies like Biolife and other pharmaceutical companies, plasma is used to manufacture treatments for conditions like immune deficiency and bleeding disorders to increase sales.
6.	Scalability of the Solution	This application has the ability to manage more donors and provide good user experience to users. Manage traffic, respond accurately, and respond to the growing number of requests.

3.4 PROBLEM SOLUTION FIT



Define CS, fit into CC

Explore AS,

1. CUSTOMER SEGMENT CS

-Our customers include the people who are in need of blood plasma.
-All the Hospitals and voluntary organizations.

2.CUSTOMER CONSTRAINTS CC

-Lack of communication details of the blood plasma donor.
-Lack of awareness among people as no one comes forward to help with blood plasma.

3. AVAILABLE SOLUTIONS AS

-Customers try with their relatives and friends or on social media platforms in case of an emergency.
-Pros are which the donor can be found sometimes but lack of availability of contact details of the donor makes it difficult to find them.

Focus on J&P, tap into BE,

4. JOBS-TO-BE-DONE / PROBLEMS J&P

-Communication between recipient and donor.
-Notify the donor regarding the emergency.
-Also sending notifications to nearby blood banks to find recipients.

5. PROBLEM ROOT CAUSE RC

-The Lack of awareness between common people to come forward to donate plasma has become less as they fear the side effects and the impact of Global Pandemic, Covid-19 has created a demand for blood plasma as it is the available cure for the sickness.

6. BEHAVIOUR BE

-The customer checks for the donors within his/her circle which is directly related.
-Indirectly associated behavior includes complaining towards people the lack of availability and searching for the donor with irrelevant contacts.

Focus on J&P, tap into BE,

Identify strong TR & E M

7.TRIGGERS TG

-Rewards to the donors who has completed donation.
-Advertise through Ads and Videos regarding awareness of blood plasma donation.

4.EMOTIONS: BEFORE/AFTER EM

-Before : Anxiety, Stress, volatile.
-After : Happy, Relaxed.

8. YOUR SOLUTION SL

-The app provides the confidence without fear.

-The app gives assurance that the patient will somehow get the blood plasma.

-It sends alerting messages to the donor for quick response from the donor.

9.CHANNELS OF BEHAVIOUR CH

-Through online, the customer can find the details of the donor from social media platforms.
-Through offline, the customer can find the details of the donor from their friends/family circle.

4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement	Sub Requirement
FR-1	User Registration	Registration through Email and Social media accounts
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login through registered email id
FR-4	User Examination	Medical Examination before donating
FR-5	Recipient Request	The recipient makes request for blood type for plasma
FR-6	Donor Request Alert	The Donor gets alerted through email
FR-7	Closed Request Verification	Donor gets an e-certificate and rewards once donation is completed
FR-8	Videos and Donation camps	Users can look up the benefits of plasma donation and information related
FR-9	Chat Assistant	Helps to solve queries related to donation within the app

4.2 NON-FUNCTIONAL REQUIREMENT

Non-functional Requirements:

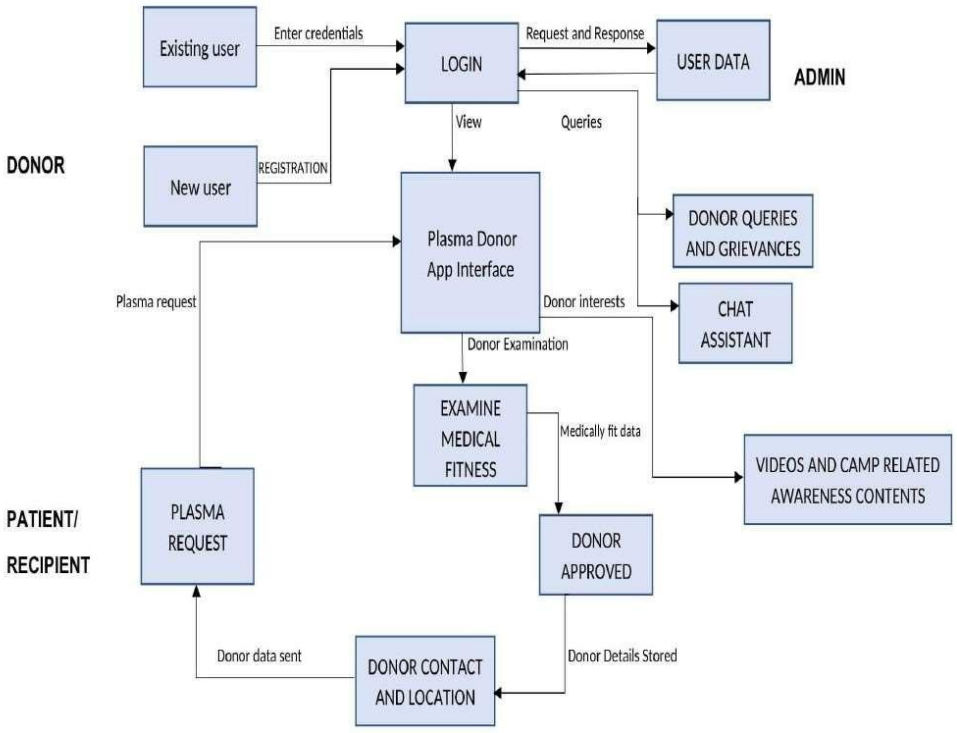
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This app is easy to use, easy to learn and navigate. Tasks such as booking a donation appointment could be completed in few steps and no instructions and training are required and this app is usable by people of all age group.
NFR-2	**Security**	This is a secure web application plus a secure database system that provides a safe environment for patients, doctors and transplant centres to create online profile for patients seeking living donors of plasma. Fake login and bots are carefully removed.
NFR-3	**Reliability**	All information that the user enters into the app is voluntary and the user can cease the usage at any time and delete their profile. If the user has shared any information through social network portals, it can also be removed. This app creates a friendly bond with the donors.
NFR-4	**Performance**	There is no lag during usage and the user can experience a glitch free usage. The user also gets route and tips on how to travel conveniently to the donation point.
NFR-5	**Availability**	This App will be available on Google Play store and App Store and also in web.
NFR-6	**Scalability**	This App has ability to handle multiple donors at a time and provides users with good user experience and reacts fast according to growing number of requests.

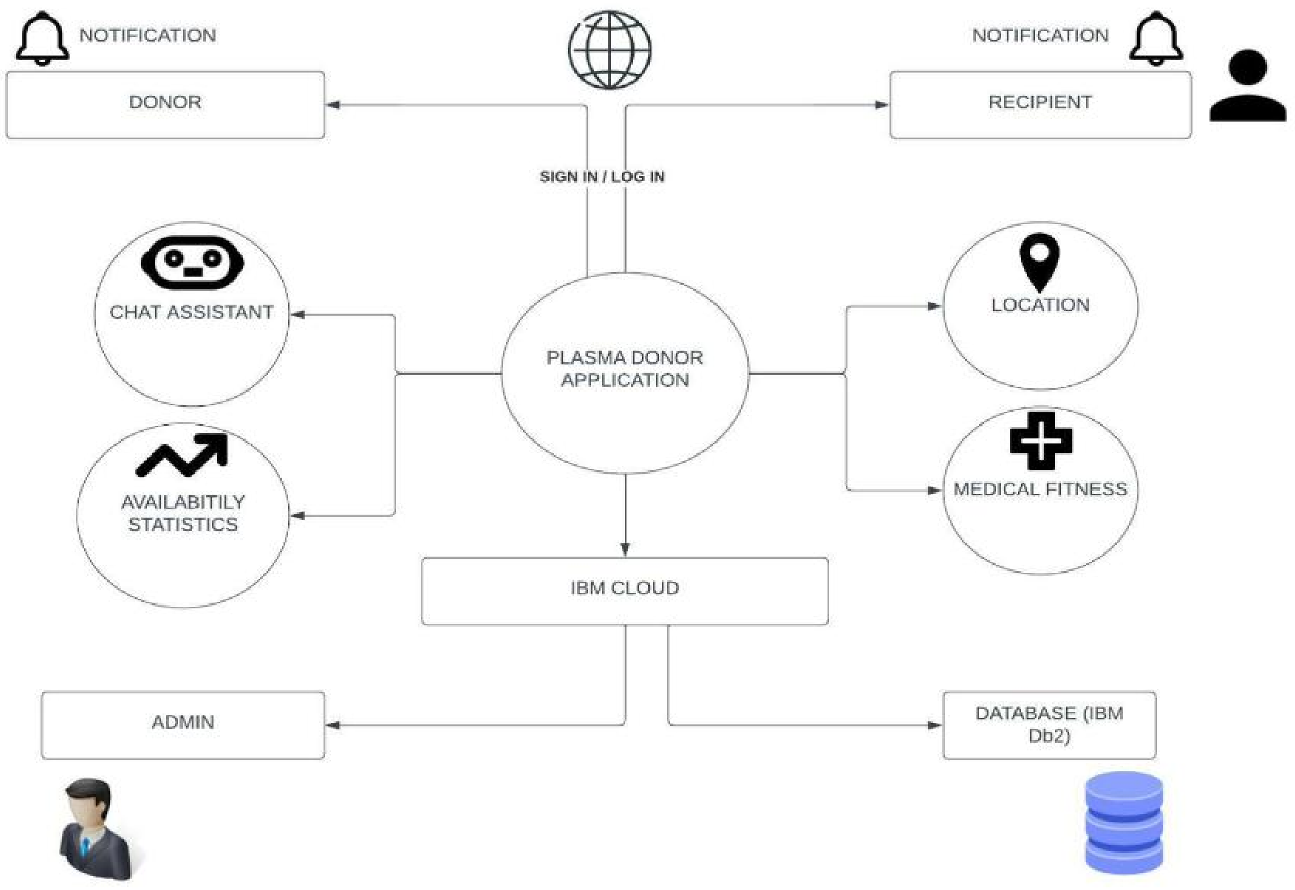
5.PROJECT DESIGN

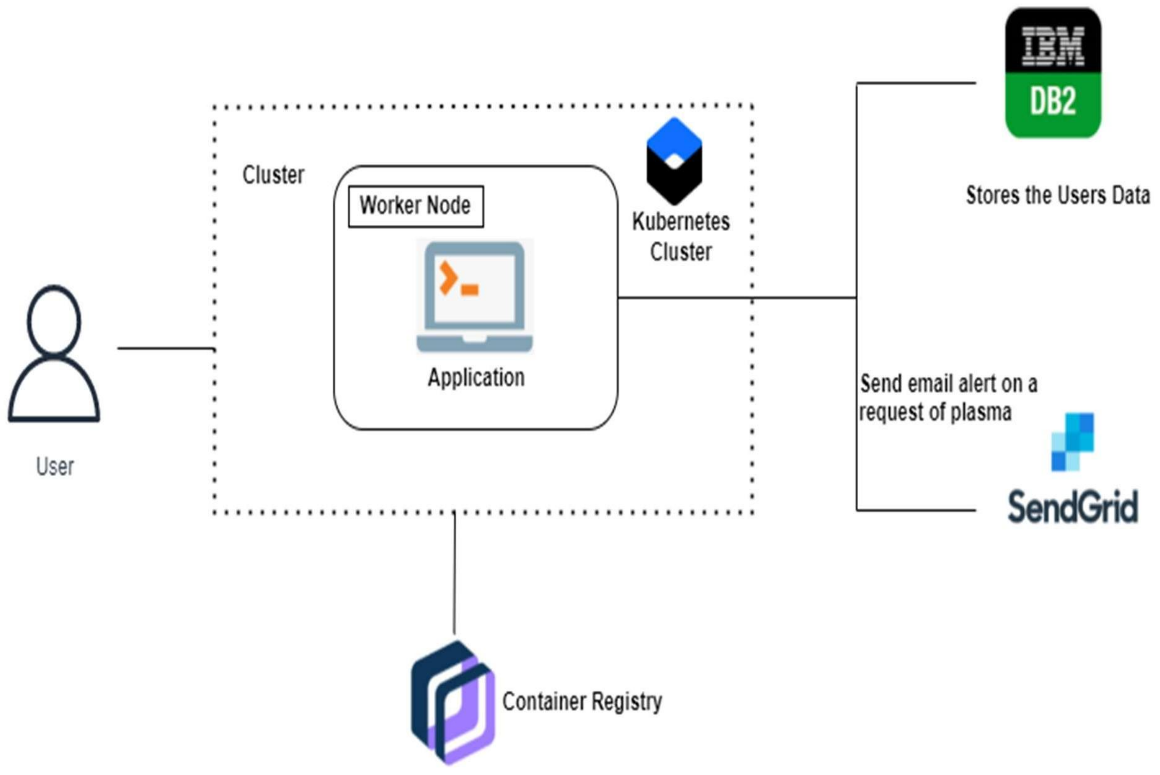
5.1 DATA FLOW DIAGRAM

Data Flow Diagrams:

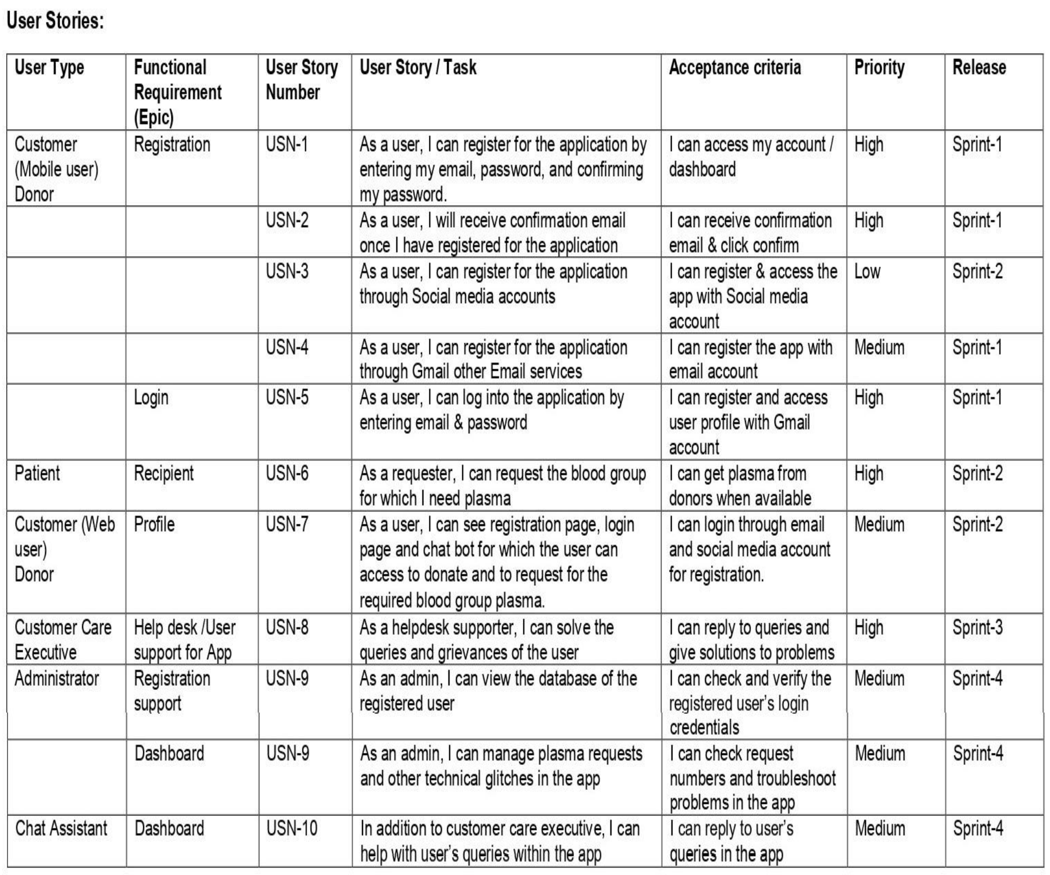


5.2 SOLUTION AND TECHNICAL ARCHITECTURE





5.3 USER STORIES



User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user) Donor	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 Social media accounts	I can register & access the app with Social media account	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail other Email services	I can register the app with email account	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can register and access user profile with Gmail account	High	Sprint-1
Patient	Recipient	USN-6	As a requester, I can request the blood group for which I need plasma	I can get plasma from donors when available	High	Sprint-2
Customer (Web user) Donor	Profile	USN-7	As a user, I can see registration page, login page and chat bot for which the user can access to donate and to request for the required blood group plasma.	I can login through email and social media account for registration.	Medium	Sprint-2
Customer Care Executive	Help desk /User support for App	USN-8	As a helpdesk supporter, I can solve the queries and grievances of the user	I can reply to queries and give solutions to problems	High	Sprint-3
Administrator	Registration support	USN-9	As an admin, I can view the database of the registered user	I can check and verify the registered user's login credentials	Medium	Sprint-4
	Dashboard	USN-9	As an admin, I can manage plasma requests and other technical glitches in the app	I can check request numbers and troubleshoot problems in the app	Medium	Sprint-4
Chat Assistant	Dashboard	USN-10	In addition to customer care executive, I can help with user's queries within the app	I can reply to user's queries in the app	Medium	Sprint-4

6.PROJECT PLANNING AND SCHEDULE

6.1 SPRINT PLANNING & ESTIMATION

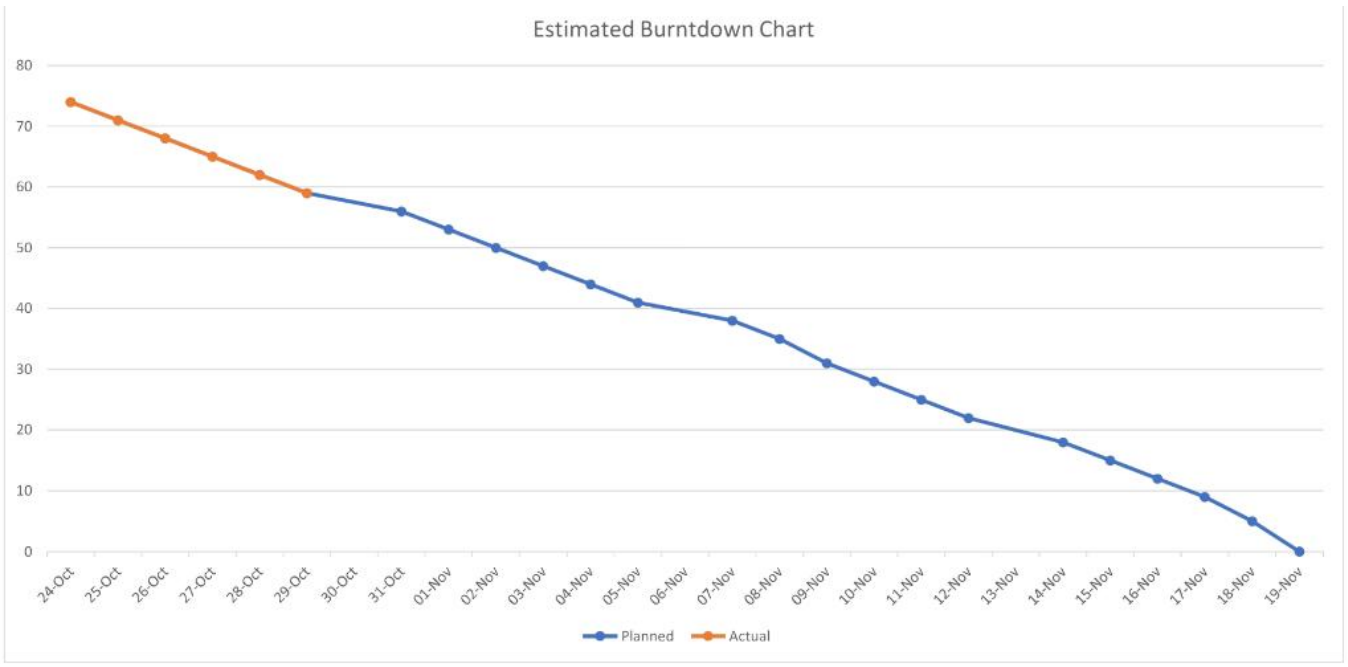
Product Backlog, Sprint Schedule, and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	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.	High	Rajsree T Subhiksha N
Sprint-1	Email Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	High	Sivaranjani N Subhiksha N
Sprint-1	Registration	USN-3	As a user, I can register for the application through Gmail and other Email services	Medium	Subhikshaa Shree V N
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password	High	Rajsree T Sivaranjani N
Sprint-1	Profile	USN-5	As a user, I am able to register myself as a registered plasma donor and view my profile page.	High	Subhiksha N Rajsree T
Sprint-2	Social Media	USN-6	As a user, I can link and register to the application through social media accounts	Low	Subhikshaa Shree V N
Sprint-2	Virtual Donor Badge	USN-7	As a user, I can receive a virtual donor badge once I am successfully registered.	Medium	Subhiksha N Subhikshaa Shree V N
Sprint-2	Plasma Request	USN-8	As a user, I can place a plasma request or donate plasma. I will include the Hospital details with the request.	High	Subhiksha N Rajsree T
Sprint-2	Verifying Request	USN-9	As a user, I will wait until my request is verified through Administrators of the app. (We Admins will verify the request after confirming with the concerned Hospital)	High	Sivaranjani N Subhikshaa Shree V N
Sprint-2	Verifying Donor	USN-10	As a user, I will wait until my donorship is verified through administrators of the app. (We Admins will verify the donor from a list of registered donors and share his details to the requester.)	High	Rajsree T Subhiksha N
Sprint-3	Donation Alarm	USN-11	The Registered Donor is notified with an alarm and a message regarding the request.	High	
Sprint-3	Accept the Request	USN-12	As a Donor, I will accept the plasma request based on my interest and volunteer for the donation.	Medium	Rajsree T Subhiksha N
Sprint-3	Communication Channel	USN-13	The Communication details of the donor will be sent to the Requester and vice versa. The Requester can personally communicate with the Donor. (Details of the donor will be provided according to the level of urgency)	High	Subhiksha N Subhikshaa Shree V N
Sprint-3	Donor Details	USN-14	The details of the volunteered donor are stored in the database.	Medium	Rajsree T Subhiksha N
Sprint-4	Support	USN-15	As a user, I can chat with a chatbot regarding my queries and doubts.	Medium	Subhiksha N Subhikshaa Shree V N
Sprint-4	Grievances and FAQ	USN-16	As a user, I can post my worries and grievances in the comment section. I can also find Frequently asked Questions with answers in the FAQ section.	Medium	Rajsree T Sivaranjani N
Sprint-4	Certificate and Rewards	USN-17	As a donor, I will receive an e-certificate after donations. Virtual rewards are also provided to the donor.	Low	Rajsree T Sivaranjani N
Sprint-4	About	USN-18	As a user, I will find about the importance of plasma donation in this section of the application.	Medium	Subhikshaa Shree V N Rajsree T
Sprint-4	Administrator		We admins will approve all the plasma transaction in the application after proper verification.	High	Subhiksha N Subhikshaa Shree V N

6.2 SPRINT DELIVERY SCHEDULE

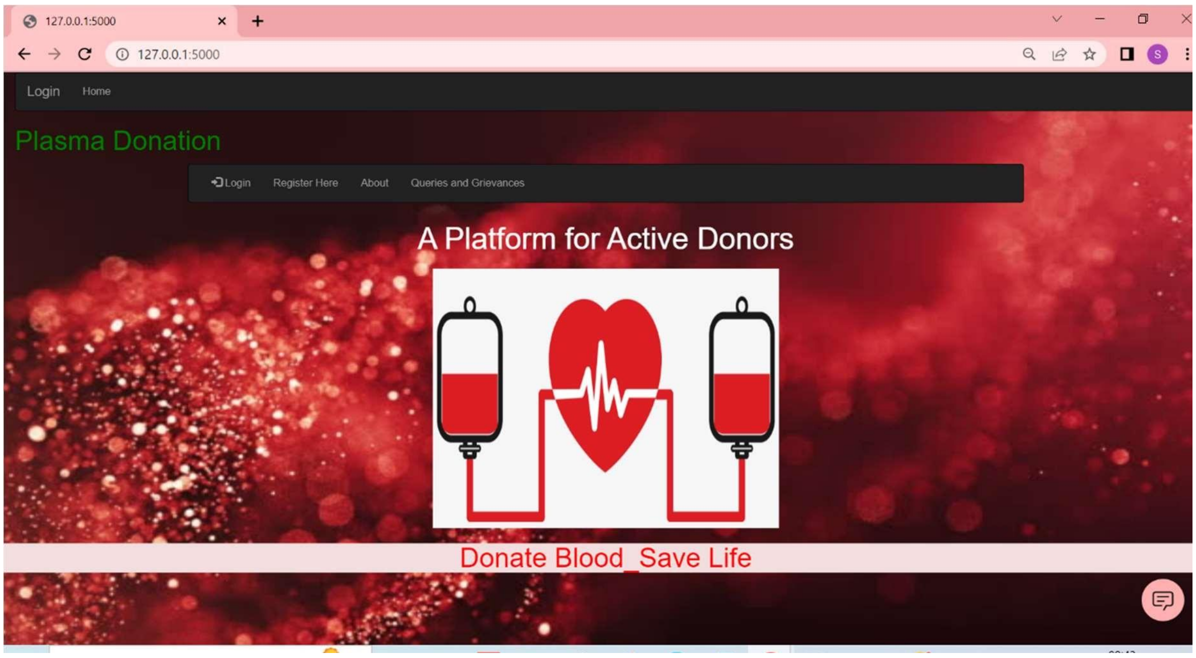
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	18	6 Days	25 Oct 2022	30 Oct 2022	18	30 Oct 2022
Sprint-2	18	6 Days	05 Nov 2022	07 Nov 2022	18	07 Nov 2022
Sprint-3	18	6 Days	10 Nov 2022	13 Nov 2022	18	13 Nov 2022
Sprint-4	20	6 Days	15 Nov 2022	18 Nov 2022	20	18 Nov 2022

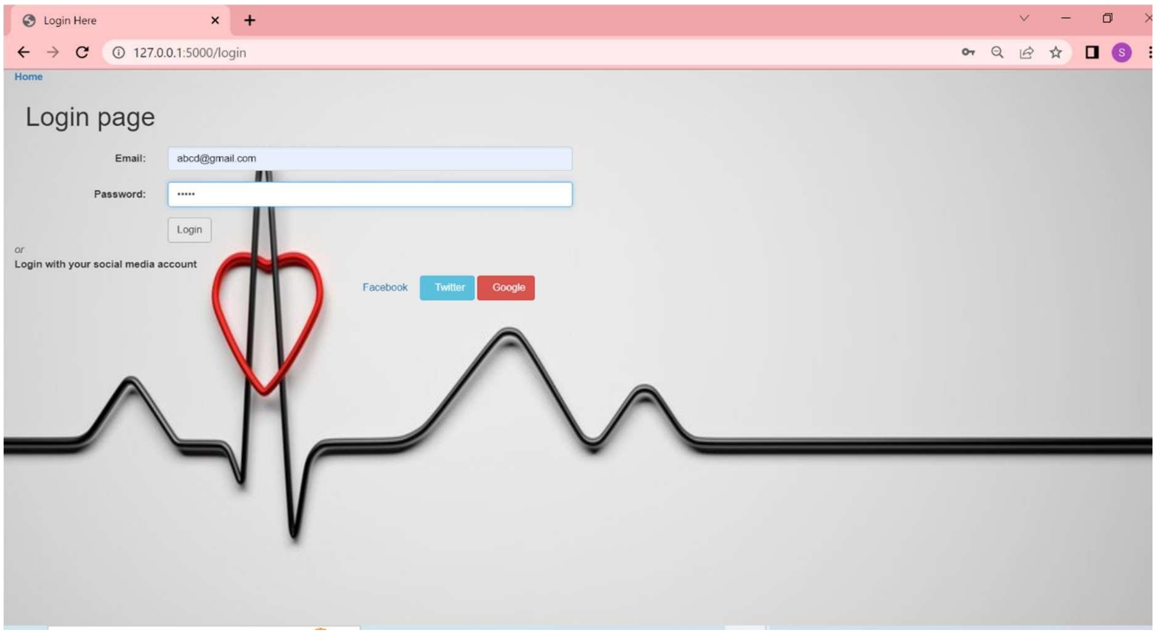
6.3 BURNDOWN CHART



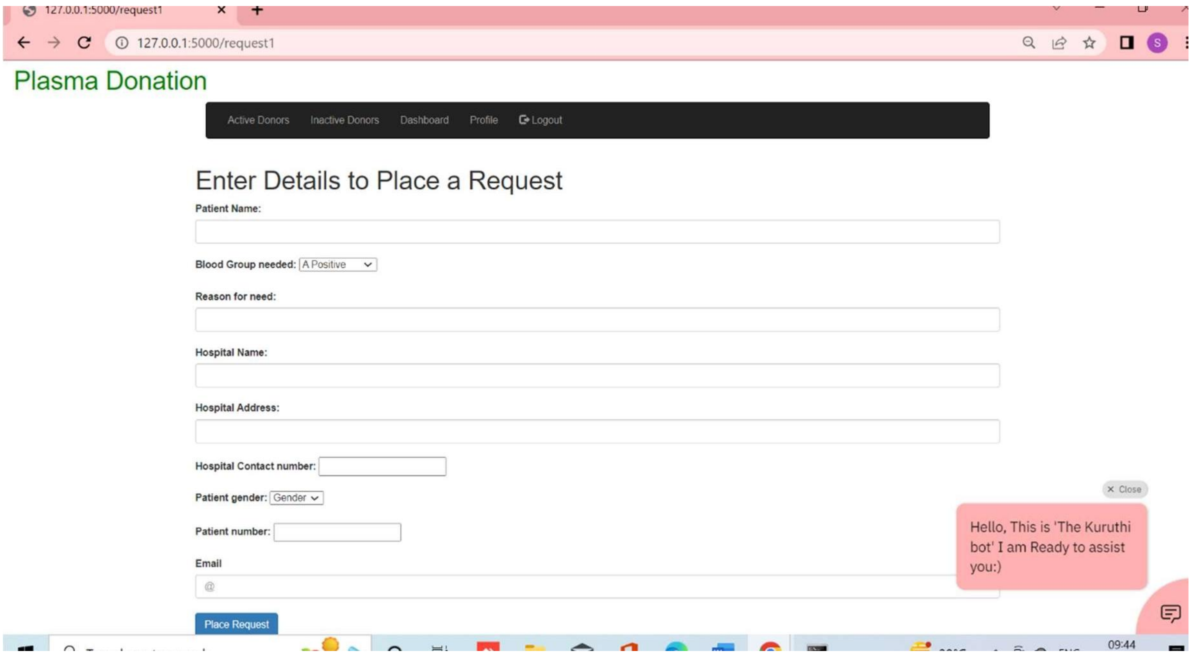
7. CODING AND SOLUTIONING

7.1 FEATURE 1





7.2 FEATURE 2



127.0.0.1:5000/request1

Plasma Donation

Active Donors | Inactive Donors | Dashboard | Profile | Logout

Enter Details to Place a Request

Patient Name:

Blood Group needed: A Positive

Reason for need:

Hospital Name:

Hospital Address:

Hospital Contact number:

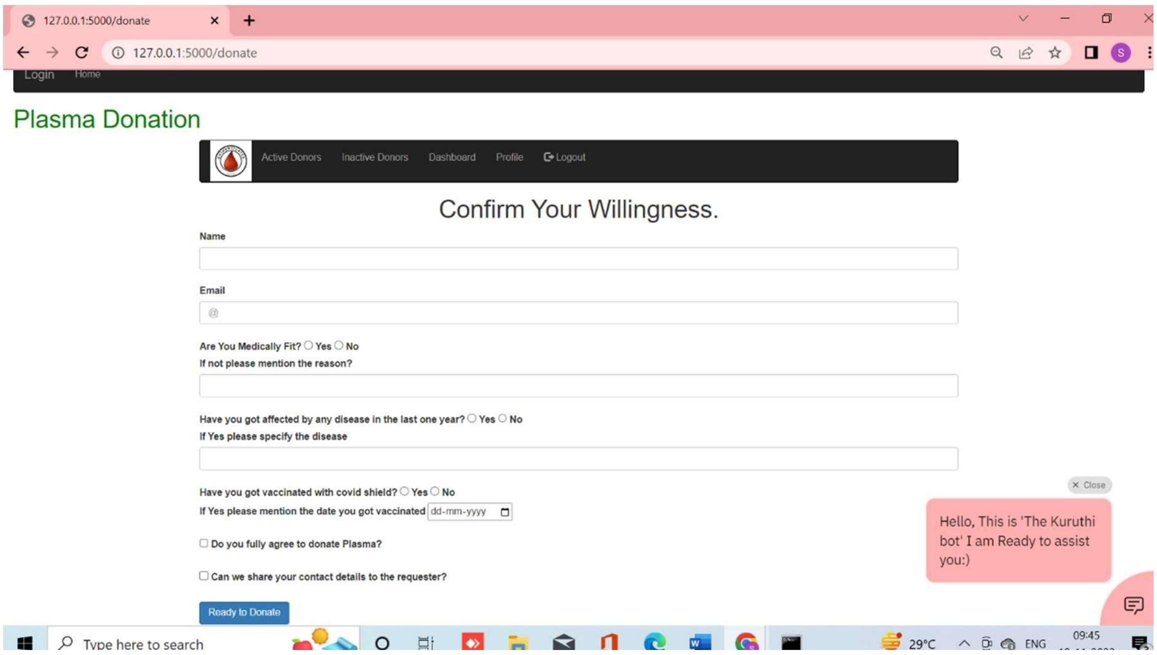
Patient gender: Gender

Patient number:

Email

Place Request

Hello, This is 'The Kuruthi bot' I am Ready to assist you:)



127.0.0.1:5000/donate

Login Home

Plasma Donation

Active Donors | Inactive Donors | Dashboard | Profile | Logout

Confirm Your Willingness.

Name

Email

Are You Medically Fit? ◯ Yes ◯ No

If not please mention the reason?

Have you got affected by any disease in the last one year? ◯ Yes ◯ No

If Yes please specify the disease

Have you got vaccinated with covid shield? ◯ Yes ◯ No

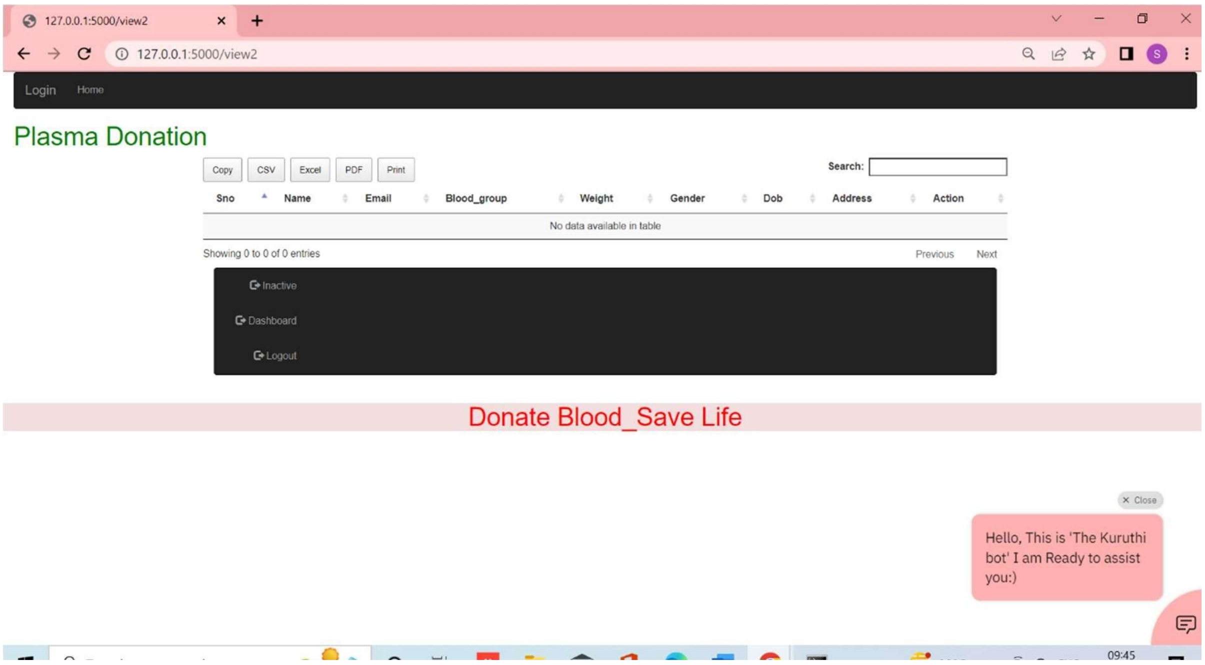
If Yes please mention the date you got vaccinated dd-mm-yyyy

☐ Do you fully agree to donate Plasma?

☐ Can we share your contact details to the requester?

Ready to Donate

Hello, This is 'The Kuruthi bot' I am Ready to assist you:)



127.0.0.1:5000/view2

Login Home

Plasma Donation

Copy CSV Excel PDF Print

Search:

Sno	Name	Email	Blood_group	Weight	Gender	Dob	Address	Action
No data available in table								

Showing 0 to 0 of 0 entries Previous Next

Inactive

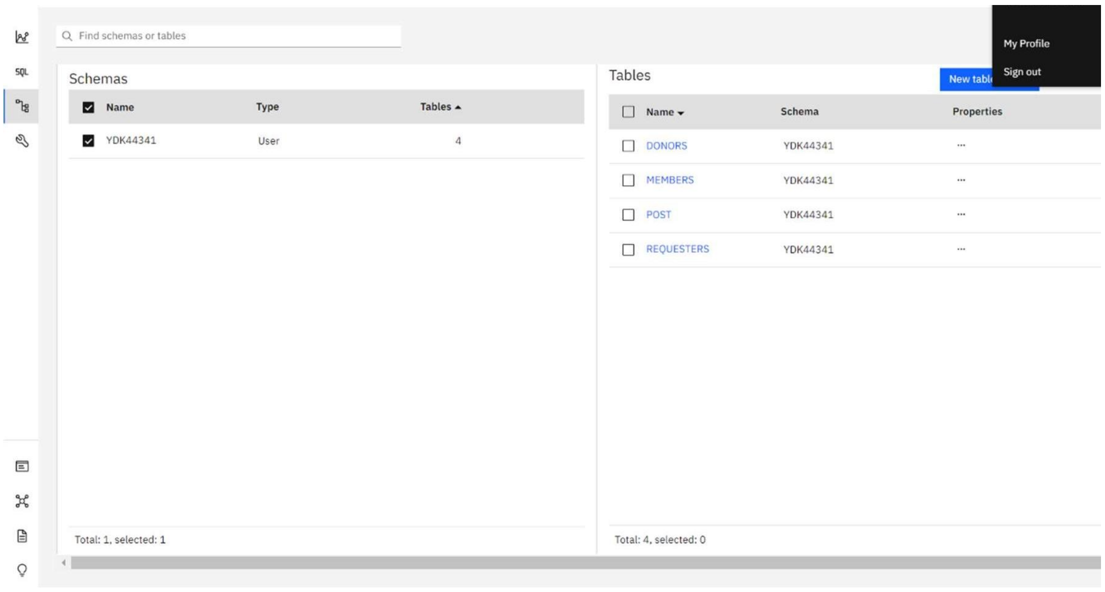
Dashboard

Logout

Donate Blood_Save Life

Hello, This is 'The Kuruthi bot' I am Ready to assist you:)

7.3 DATABASE SCHEMA



Find schemas or tables

My Profile

Sign out

Schemas

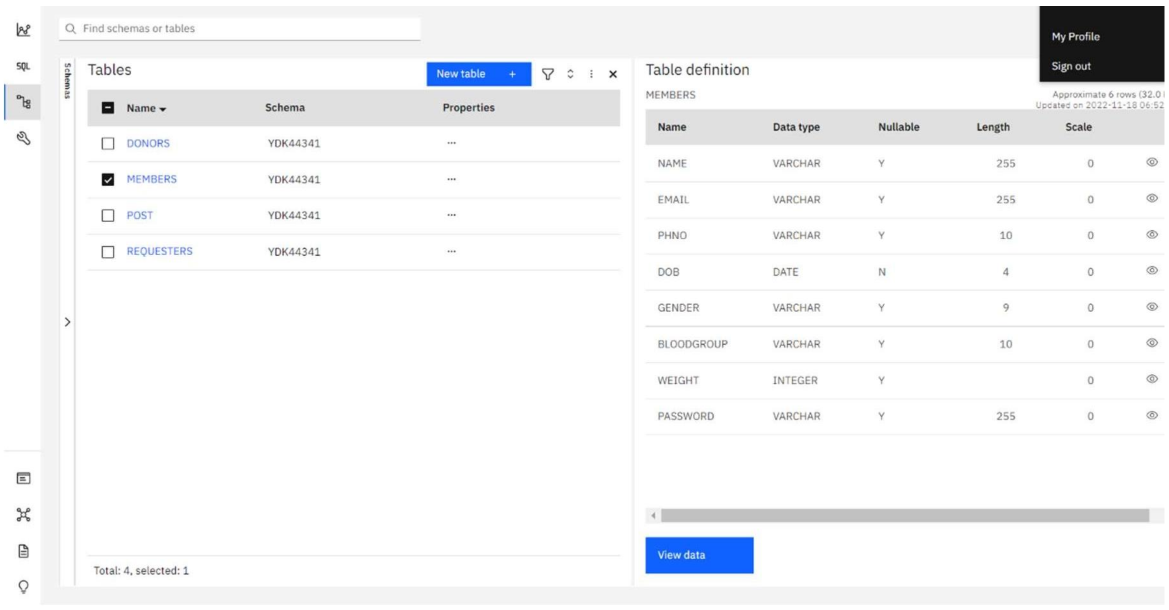
Name	Type	Tables
YDK44341	User	4

Total: 1, selected: 1

Tables

Name	Schema	Properties
DONORS	YDK44341	...
MEMBERS	YDK44341	...
POST	YDK44341	...
REQUESTERS	YDK44341	...

Total: 4, selected: 0



Tables

	Name	Schema	Properties
☐	DONORS	YDK44341	···
✓	MEMBERS	YDK44341	···
☐	POST	YDK44341	···
☐	REQUESTERS	YDK44341	···

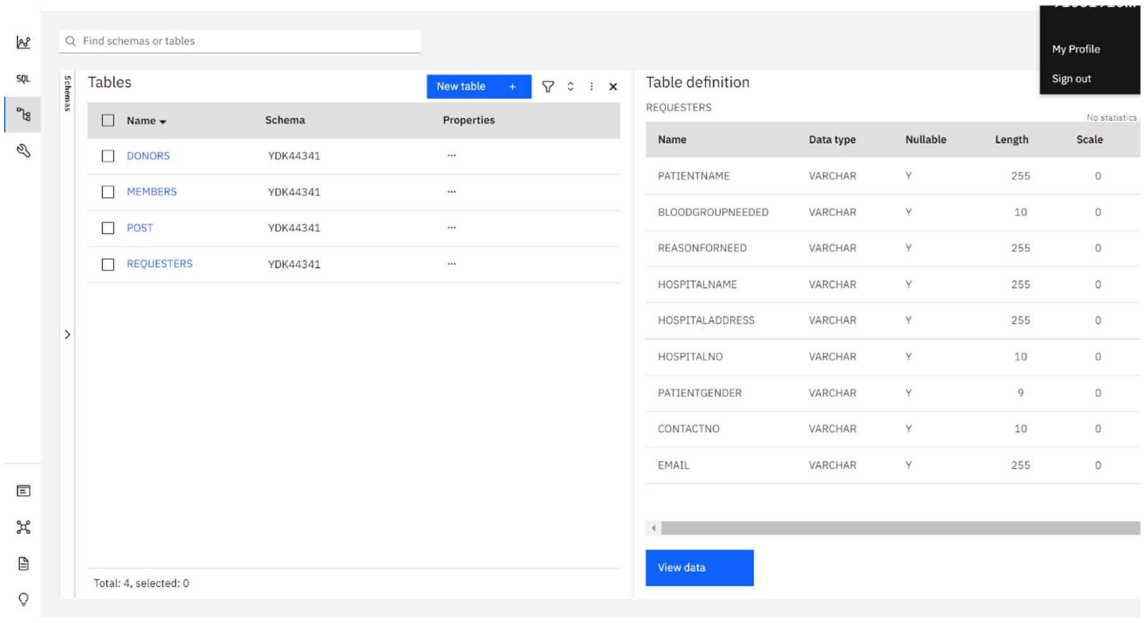
Total: 4, selected: 1

Table definition

MEMBERS

Approximate 6 rows (32.0
Updated on 2022-11-18 06:52

Name	Data type	Nullable	Length	Scale
NAME	VARCHAR	Y	255	0
EMAIL	VARCHAR	Y	255	0
PHNO	VARCHAR	Y	10	0
DOB	DATE	N	4	0
GENDER	VARCHAR	Y	9	0
BLOODGROUP	VARCHAR	Y	10	0
WEIGHT	INTEGER	Y		0
PASSWORD	VARCHAR	Y	255	0



Tables

	Name	Schema	Properties
☐	DONORS	YDK44341	···
☐	MEMBERS	YDK44341	···
☐	POST	YDK44341	···
☐	REQUESTERS	YDK44341	···

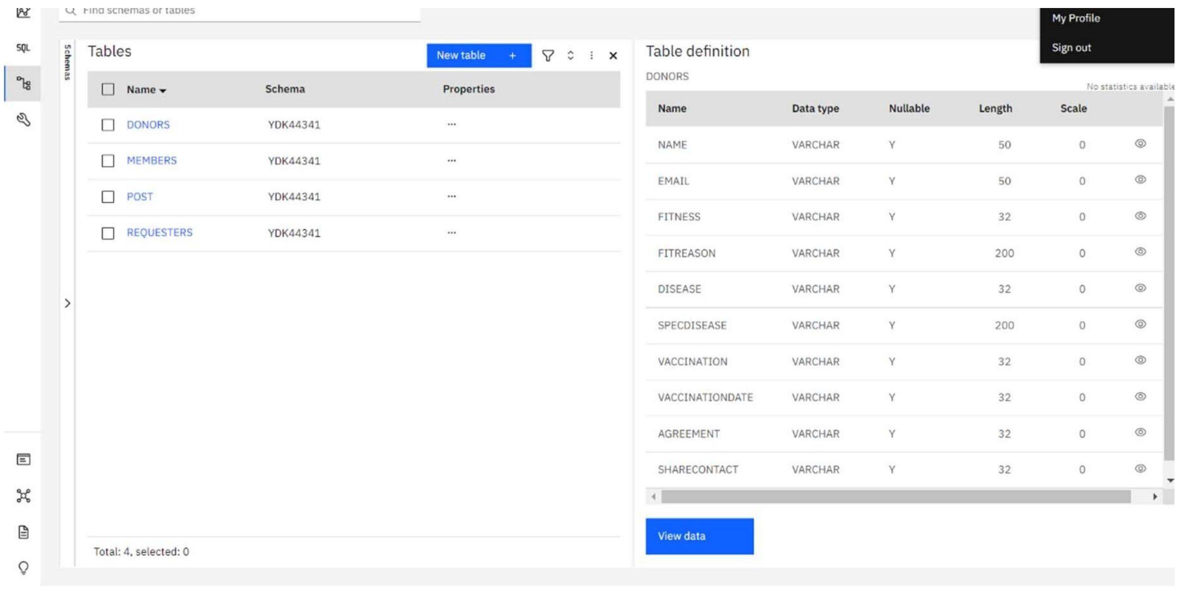
Total: 4, selected: 0

Table definition

REQUESTERS

No statistics

Name	Data type	Nullable	Length	Scale
PATIENTNAME	VARCHAR	Y	255	0
BLOODGROUPNEEDED	VARCHAR	Y	10	0
REASONFORNEED	VARCHAR	Y	255	0
HOSPITALNAME	VARCHAR	Y	255	0
HOSPITALADDRESS	VARCHAR	Y	255	0
HOSPITALNO	VARCHAR	Y	10	0
PATIENTGENDER	VARCHAR	Y	9	0
CONTACTNO	VARCHAR	Y	10	0
EMAIL	VARCHAR	Y	255	0



Tables

	Name	Schema	Properties
☐	DONORS	YDK44341	···
☐	MEMBERS	YDK44341	···
☐	POST	YDK44341	···
☐	REQUESTERS	YDK44341	···

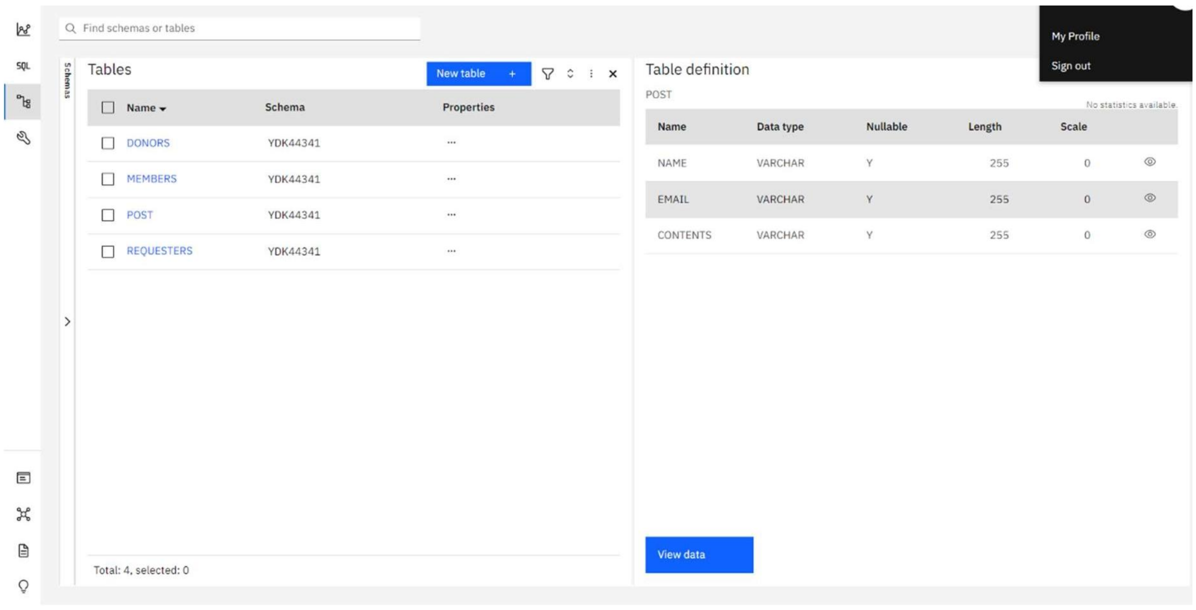
Total: 4, selected: 0

Table definition

DONORS

No statistics available

Name	Data type	Nullable	Length	Scale
NAME	VARCHAR	Y	50	0
EMAIL	VARCHAR	Y	50	0
FITNESS	VARCHAR	Y	32	0
FITREASON	VARCHAR	Y	200	0
DISEASE	VARCHAR	Y	32	0
SPECDISEASE	VARCHAR	Y	200	0
VACCINATION	VARCHAR	Y	32	0
VACCINATIONDATE	VARCHAR	Y	32	0
AGREEMENT	VARCHAR	Y	32	0
SHARECONTACT	VARCHAR	Y	32	0



Tables

	Name	Schema	Properties
☐	DONORS	YDK44341	···
☐	MEMBERS	YDK44341	···
☐	POST	YDK44341	···
☐	REQUESTERS	YDK44341	···

Total: 4, selected: 0

Table definition

POST

No statistics available.

Name	Data type	Nullable	Length	Scale
NAME	VARCHAR	Y	255	0
EMAIL	VARCHAR	Y	255	0
CONTENTS	VARCHAR	Y	255	0

8.TESTING

8.1 TEST CASES

8.1.1 FUNCTIONAL TESTING

Functional test can be defined as testing two or more modules together with the intent of finding defects, demonstrating that defects are not present, verifying that the module performs its intended functions as stated in the specification and establishing confidence that a program does what it is supposed to do.

8.1.2 WHITE BOX TESTING:

Testing based on an analysis of internal workings and structure of a piece of software. This testing can be done sing the percentage value of load and energy. The tester should know what exactly is done in the internal program. Includes techniques such as Branch Testing and Path Testing. Also known as Structural Testing and Glass Box Testing.

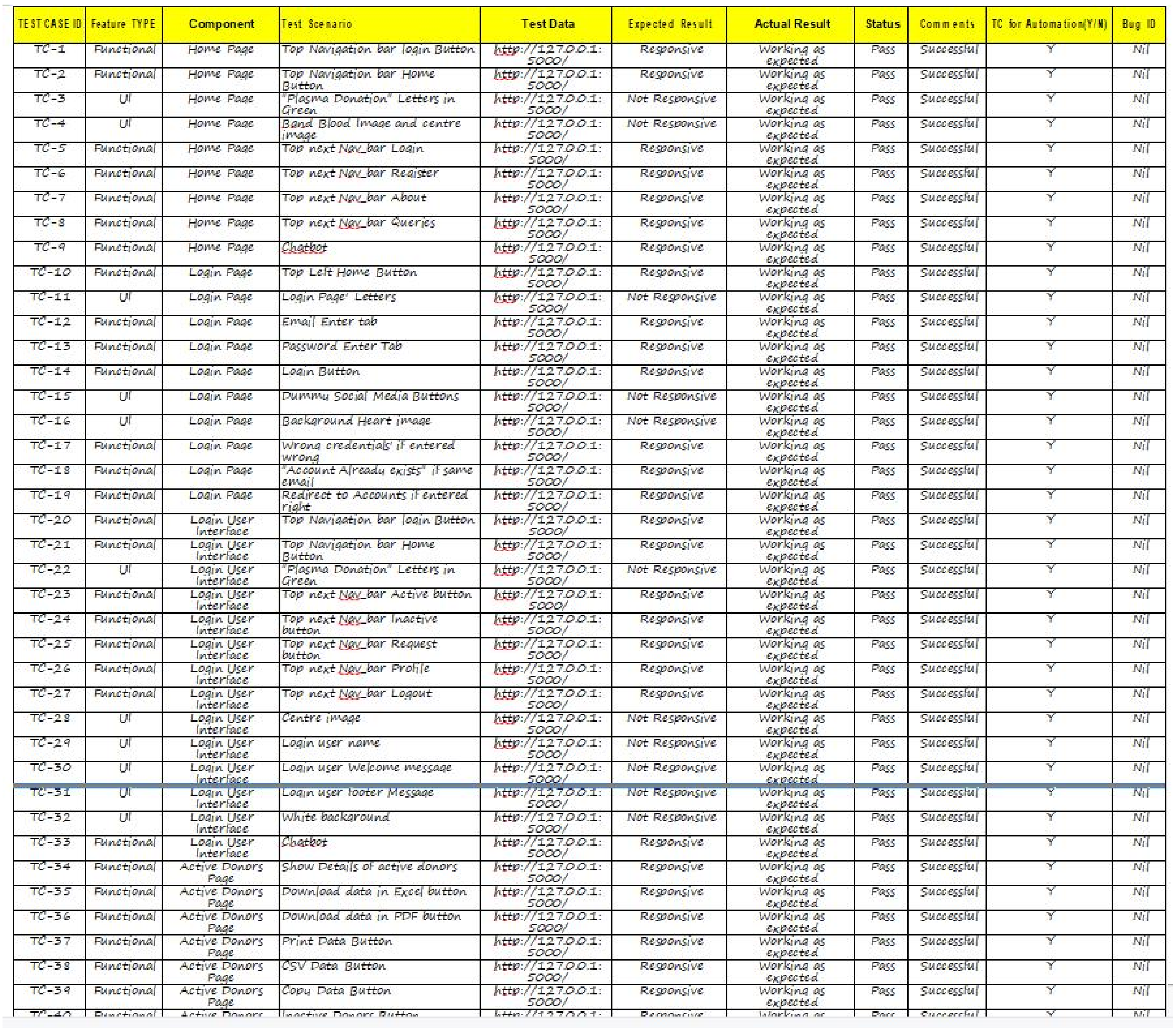
8.1.3 BLACK BOX TESTING:

Testing without knowledge of the internal workings of the item being tested. Tests are usually functional. This testing can be done by the user who has no knowledge of how the shortest path is found.

8.2 USER ACCEPTANCE TESTING

Acceptance testing can be defined in many ways, but a simple definition is the succeeds when the software functions in a manner that can be reasonable expected by the customer. After the acceptance test has been conducted, one of the two possible conditions exists. This is to fine whether the inputs are accepted by the database or other validations. For example accept only numbers in the numeric field, date format data in the date field. Also the null check for the not null fields. If any error occurs then show the error messages. The function of performance characteristics to specification and is accepted. A deviation from specification is uncovered and a deficiency list is created. User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

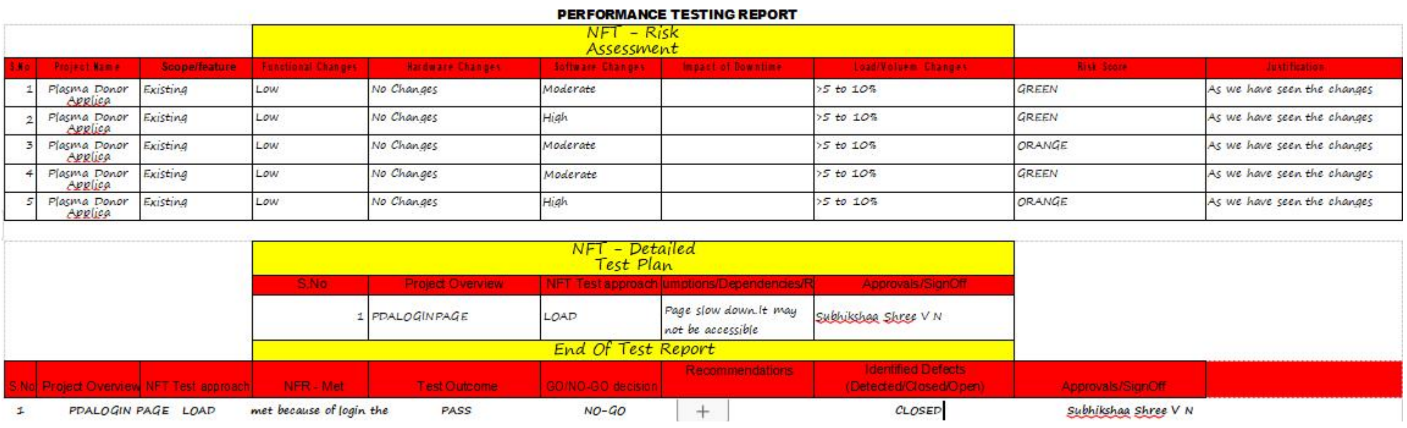
8.3 TEST CASES



TEST CASE ID	Feature TYPE	Component	Test Scenario	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	Bug ID
TC-1	Functional	Home Page	Top Navigation bar login Button	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-2	Functional	Home Page	Top Navigation bar Home Button	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-3	UI	Home Page	"Plasma Donation" Letters in Green	http://127.0.0.1: 5000/	Not Responsive	Working as expected	Pass	Successful	Y	Nil
TC-4	UI	Home Page	Band Blood Image and centre image	http://127.0.0.1: 5000/	Not Responsive	Working as expected	Pass	Successful	Y	Nil
TC-5	Functional	Home Page	Top next Nav_bar Login	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-6	Functional	Home Page	Top next Nav_bar Register	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-7	Functional	Home Page	Top next Nav_bar About	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-8	Functional	Home Page	Top next Nav_bar Queries	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-9	Functional	Home Page	Chatbot	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-10	Functional	Login Page	Top Left Home Button	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-11	UI	Login Page	Login Page' Letters	http://127.0.0.1: 5000/	Not Responsive	Working as expected	Pass	Successful	Y	Nil
TC-12	Functional	Login Page	Email Enter tab	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-13	Functional	Login Page	Password Enter Tab	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-14	Functional	Login Page	Login Button	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-15	UI	Login Page	Dummy Social Media Buttons	http://127.0.0.1: 5000/	Not Responsive	Working as expected	Pass	Successful	Y	Nil
TC-16	UI	Login Page	Background Heart image	http://127.0.0.1: 5000/	Not Responsive	Working as expected	Pass	Successful	Y	Nil
TC-17	Functional	Login Page	Wrong credentials' if entered wrong	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-18	Functional	Login Page	"Account Already exists" if same email	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-19	Functional	Login Page	Redirect to Accounts if entered right	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-20	Functional	Login User Interface	Top Navigation bar login Button	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-21	Functional	Login User Interface	Top Navigation bar Home Button	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-22	UI	Login User Interface	"Plasma Donation" Letters in Green	http://127.0.0.1: 5000/	Not Responsive	Working as expected	Pass	Successful	Y	Nil
TC-23	Functional	Login User Interface	Top next Nav_bar Active button	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-24	Functional	Login User Interface	Top next Nav_bar Inactive button	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-25	Functional	Login User Interface	Top next Nav_bar Request button	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-26	Functional	Login User Interface	Top next Nav_bar Profile	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-27	Functional	Login User Interface	Top next Nav_bar Logout	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-28	UI	Login User Interface	Centre image	http://127.0.0.1: 5000/	Not Responsive	Working as expected	Pass	Successful	Y	Nil
TC-29	UI	Login User Interface	Login user name	http://127.0.0.1: 5000/	Not Responsive	Working as expected	Pass	Successful	Y	Nil
TC-30	UI	Login User Interface	Login user Welcome message	http://127.0.0.1: 5000/	Not Responsive	Working as expected	Pass	Successful	Y	Nil
TC-31	UI	Login User Interface	Login user footer Message	http://127.0.0.1: 5000/	Not Responsive	Working as expected	Pass	Successful	Y	Nil
TC-32	UI	Login User Interface	White background	http://127.0.0.1: 5000/	Not Responsive	Working as expected	Pass	Successful	Y	Nil
TC-33	Functional	Login User Interface	Chatbot	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-34	Functional	Active Donors Page	Show Details of active donors	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-35	Functional	Active Donors Page	Download data in Excel button	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-36	Functional	Active Donors Page	Download data in PDF button	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-37	Functional	Active Donors Page	Print Data Button	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-38	Functional	Active Donors Page	CSV Data Button	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-39	Functional	Active Donors Page	Copy Data Button	http://127.0.0.1: 5000/	Responsive	Working as expected	Pass	Successful	Y	Nil
TC-40	Functional	Active Donors	Inactive Donors Button	http://127.0.0.1:	Responsive	Working as	Pass	Successful	Y	Nil

9.RESULTS

9.1 PERFORMANCE METRICES



PERFORMANCE TESTING REPORT

NFT - Risk Assessment

S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Voluem Changes	Risk Score	Justification
1	Plasma Donor Applica	Existing	Low	No Changes	Moderate		>5 to 10%	GREEN	As we have seen the changes
2	Plasma Donor Applica	Existing	Low	No Changes	High		>5 to 10%	GREEN	As we have seen the changes
3	Plasma Donor Applica	Existing	Low	No Changes	Moderate		>5 to 10%	ORANGE	As we have seen the changes
4	Plasma Donor Applica	Existing	Low	No Changes	Moderate		>5 to 10%	GREEN	As we have seen the changes
5	Plasma Donor Applica	Existing	Low	No Changes	High		>5 to 10%	ORANGE	As we have seen the changes

NFT - Detailed Test Plan

S.No	Project Overview	NFT Test approach	umptions/Dependencies/R	Approvals/SignOff
1	PDALOGINPAGE	LOAD	Page slow down.It may not be accessible	Subhikshaa Shree V N

End Of Test Report

S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff
1	PDALOGIN PAGE	LOAD	met because of login the	PASS	NO-GO		CLOSED	Subhikshaa Shree V N

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- Very much helpful at times of emergency as this app helps us to find donors easily.
- User friendly interface of the app makes it easier to interact
- Helps very much in voluntary activities.
- Clear details of donors and acceptors can be found once request/donation is placed
- Does not consume much space as it runs in the cloud

DISADVANTAGES:

- Since it is a beta verision some user troubles couldn't be handled
- Verification of details from Admin side could make delay.

11. CONCLUSION

“Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.” Cloud makes hardware resources readily available and quick to configure, which shortens the time required for developers to show a working version of their products. Also, cloud allows the reuse of the same resources for multiple successive projects, which is more cost-efficient. As this is a cloud based app it is easy to handle and use. We will hope to look after the updates of this Plasma Donor Application in the future!

12. APPENDIX

SOURCE CODE:

```
from flask import Flask, render_template,request,redirect,url_for,session

import ibm_db

import sendgrid

import os

from sendgrid.helpers.mail import *

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=764264db-9824-4b7c-82df-
40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=32536;SECURITY=SSL;S
SLServerCertificate=DigiCertGlobalRootCA.crt;UID=ydk44341;PWD=eENReIXIS7xOOBBa",",")


app = Flask(_name_)

app.secret_key = b'_5#y2L"F4Q8z\n\xec]/'


@app.route('/')

def home():

    return render_template('home.html')


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

def register():

    session['msg']=""

    if request.method == 'POST':

        name = request.form['name']

        email = request.form['email']

        phno = request.form['phno']

        dob = request.form['dob']

        gender = request.form['gender']

        bloodgroup = request.form['bloodgroup']

        weight = request.form['weight']

        password = request.form['newpassword']


        sql = "SELECT * FROM Members WHERE email =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)


        if account:
            session['msg']= 'Account already exists'
            return redirect(url_for("login"))
        else:
            insert_sql = "INSERT INTO Members VALUES (?,?,?,?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, email)
            ibm_db.bind_param(prep_stmt, 3, phno)
            ibm_db.bind_param(prep_stmt, 4, dob)
            ibm_db.bind_param(prep_stmt, 5, gender)
            ibm_db.bind_param(prep_stmt, 6, bloodgroup)
            ibm_db.bind_param(prep_stmt, 7, weight)
            ibm_db.bind_param(prep_stmt, 8, password)
            ibm_db.execute(prep_stmt)
            session['msg']= 'Account created Successfully '
            sg = sendgrid.SendGridAPIClient(api_key=os.environ.get('SENDGRID_API_KEY'))
            from_email = Email("910019106033@student.autmdu.in")
            to_email = To(email)
            subject = "Plasma Donor App"
            content = Content("text/plain", "You are successfully registered to Plasma Donor App")
            mail = Mail(from_email, to_email, subject, content)
            response = sg.client.mail.send.post(request_body=mail.get())
            print(response.status_code)
            print(response.body)
            print(response.headers)
            return redirect(url_for("login"))


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

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

    if request.method == 'POST':
        email = request.form['email']
        password = request.form['newpassword']



        sql = "SELECT * FROM Members WHERE Email =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email) ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        session['user'] = account
        accounts=account

        if (account):
            if (password == accounts['PASSWORD'] ):
                return render_template('accounts.html',name=account['NAME'])
            else :
                return render_template('login.html',msg='wrong Password')
        else :
            return render_template('login.html',msg='wrong credentials')



    else:
        return  render_template('login.html')

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

@app.route('/view2')
def view2():
    return render_template('view2.html')


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


@app.route('/request1',methods=['GET', 'POST'])
def request1():
    session['msg']=""
    if request.method == 'POST':
        patientname = request.form['patientname']
        bloodgroupneeded = request.form['bloodgroupneeded']
        reasonforneed = request.form['reasonforneed']
        hospitalname = request.form['hospitalname']
        hospitaladdress = request.form['hospitaladdress']
        hospitalno = request.form['hospitalno']
        patientgender = request.form['patientgender']
        contactno = request.form['contactno']
        email = request.form['email']
        insert_sql = "INSERT INTO Requesters VALUES (?,?,?,?,?,?,?,?,?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1, patientname)
        ibm_db.bind_param(prep_stmt, 2, bloodgroupneeded)
        ibm_db.bind_param(prep_stmt, 3, reasonforneed)
        ibm_db.bind_param(prep_stmt, 4, hospitalname)
        ibm_db.bind_param(prep_stmt, 5, hospitaladdress)
        ibm_db.bind_param(prep_stmt, 6, hospitalno)
        ibm_db.bind_param(prep_stmt, 7, patientgender)
        ibm_db.bind_param(prep_stmt, 8, contactno)
        ibm_db.bind_param(prep_stmt, 9, email)
        ibm_db.execute(prep_stmt)
        session['msg']= 'Request Placed Successfully'
        sg = sendgrid.SendGridAPIClient(api_key=os.environ.get('SENDGRID_API_KEY'))
        from_email = Email("910019106033@student.autmdu.in")
        to_email = To(email)
        subject = "Plasma Donor App-Request"
        content = Content("text/plain", "Your Request is under review")
        mail = Mail(from_email, to_email, subject, content)
        response = sg.client.mail.send.post(request_body=mail.get())
        print(response.status_code)
        print(response.body)
        print(response.headers)

    return render_template('request.html')

@app.route('/donate',methods=['GET','POST'])
def donate():
    session['msg']=""
    if request.method == 'POST':
        name = request.form['name']
        email = request.form['email']
        fitness = request.form['fitness']
        fitreason = request.form['fitreason']
        disease = request.form['disease']
        specdisease = request.form['specdisease']
        vaccination = request.form['vaccination']
        vaccinationdate = request.form['vaccinationdate']
        agreement = request.form['agreement']
        sharecontact = request.form['sharecontact']



        sql = "SELECT * FROM Donors WHERE email =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        if account:
            session['msg']= 'Already registered for Donation'
            return redirect(url_for("donate"))
        else:
            insert_sql = "INSERT INTO Donors VALUES (?,?,?,?,?,?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, email)
            ibm_db.bind_param(prep_stmt, 3, fitness)
            ibm_db.bind_param(prep_stmt, 4, fitreason)
            ibm_db.bind_param(prep_stmt, 5, disease)
            ibm_db.bind_param(prep_stmt, 6, specdisease)
            ibm_db.bind_param(prep_stmt, 7, vaccination)
            ibm_db.bind_param(prep_stmt, 8, vaccinationdate)
            ibm_db.bind_param(prep_stmt, 9, agreement)
            ibm_db.bind_param(prep_stmt, 10, sharecontact)
            ibm_db.execute(prep_stmt)
            return redirect(url_for("readytodonate", msg='Your Donorship is under review'))

    return render_template('donate.html')

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

@app.route('/profile')
def profile():
    return render_template('profile.html',user=session['user'])

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

@app.route('/queries',methods=['GET', 'POST'])
def queries():
    session['msg']=""
    if request.method == 'POST':
        name = request.form['name']
        email = request.form['email']
        contents = request.form['contents']
        insert_sql = "INSERT INTO Post VALUES (?,?,?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1, name)
        ibm_db.bind_param(prep_stmt, 2, email)
        ibm_db.bind_param(prep_stmt, 3, contents)
        ibm_db.execute(prep_stmt)
        session['msg']= 'Request Placed Successfully'
        return redirect(url_for("queries"))
    return render_template('queries.html')

if __name___== "_main_":
            app.run(host ='0.0.0.0', port = 5000, debug = True)
```