IBM NALAIYATHIRAN PROJECT REPORT

PLASMA DONOR APPLICATION

Team Id	**PNT2022TMID02342**
Project Name	**Plasma Donor Application**
Team Members	**- Mithul Sudharsan R (2116190701273)** **- Vishva Prasad P (2116190701254)** **- Vignesh P (2116190701241)** **- Vishal V (2116190701251)**

Table Of Contents

1.INTRODUCTION

1.1 Project Overview

A vital component of the treatment for many significant health issues is plasma. As a result, people are asked to donate blood plasma at blood drives. Our project's major objective is to make it simpler for COVID-19 patients to find a plasma donor and donate plasma once they have recovered. The system is designed for two different user types: those looking to donate plasma and those in need of plasma. Additionally, the user can check the number of active cases, neighbouring immunisation facilities, and hospitals' addresses.

The primary goal of creating the website is to make it simpler for COVID-19 patients to find a plasma donor quickly and conveniently. However, the demand for plasma-derived products has been rising steadily for a while, and blood collection organisations must change to fulfil this need. This essay seeks to cover the primary incentives and disincentives for donating whole blood and to contrast them with those that we currently know about donating plasma. Current study reveals parallels between these behaviours, as well as distinctions that point to the necessity for more plasma donation research.

1.2 Purpose

The need for plasma became urgent during the COVID 19 crisis, and the number of donors has declined.

It would be helpful to save the donor information and assist the less fortunate by informing the list of current donors. In order to solve the issue, a programme will be created that will collect donor information, store it, and provide information upon request.

2. LITERATURE SURVEY

2.1 EXIXTING PROBLEM

- Only mobile based system is available web-based system is available
- Less Security
- No proper coordination between different applications and users
- Cannot upload and download the latest updates at right time Fewer users-friendly

2.2 REFERENCE

Several experiments have been carried out over the years by different groups of researchers. Here are some of the following groups:

[1] Denuis O'Neil (1999). "Blood component" Archived from the original on June 5, 2013.

[2] ways to keep your plasma healthy, Original Archived November 1, 2013, Accessed November 11, 2011.

[3] Ripathis S, Kumar V, Prabhakar A, Joshi S, Agarwal A (2015). "Microscale Passive Plasma Separation: A Review of Design Principles and Microdevices," J. Micromech Micro 25 (8): 083001;

[4] P. C. P. C. a. V. I. M. Yan, "Building a chatbot with server less computing," IBM watson research center, 2016.

[5] S. E. a. B. J. J. Short, """Cloud Event Programming Paradigms: Applications and Analysis,"," 9th IEEE International Conference on Cloud Computing (CLOUD), pp. pp. 4 00-406, 2017.

2.3 Problem Statement Definition

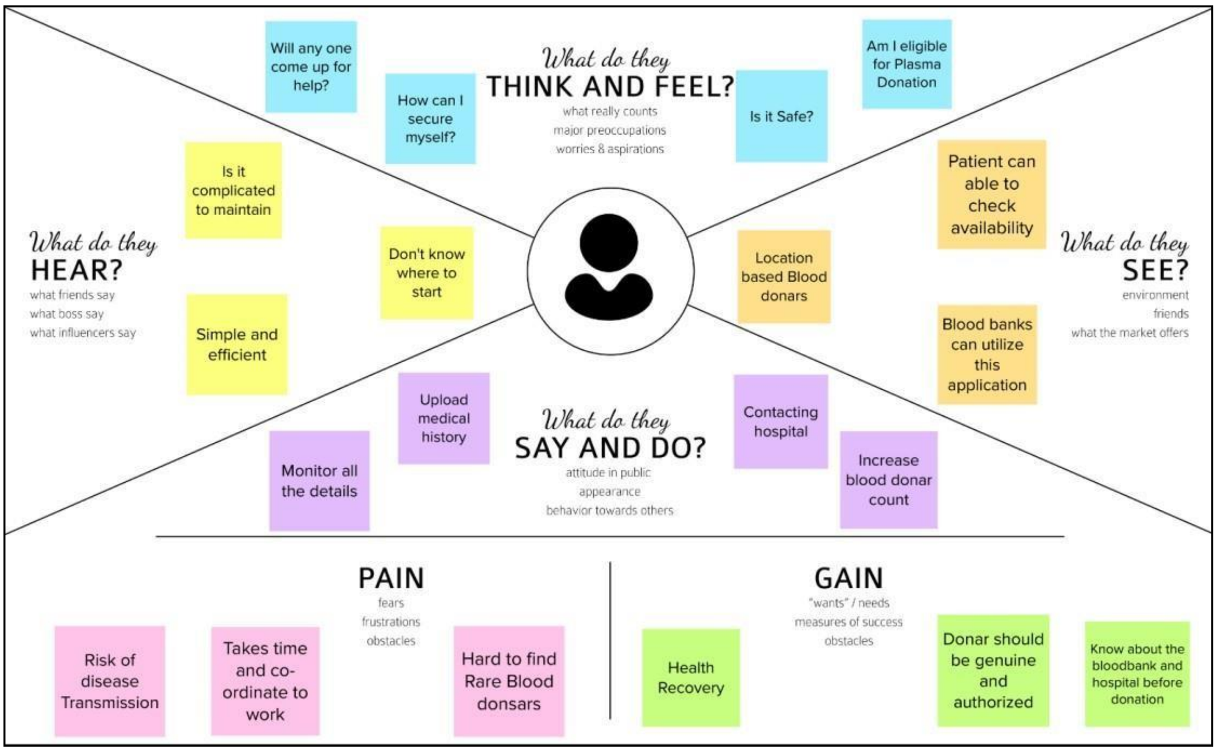
The need for plasma surged significantly during the COVID 19 crisis since there were no vaccines available to treat the infected patients.

Finding a plasma donor in such a situation was extremely difficult, and determining which donors are eligible to donate plasma as well as whether they have previously been infected and have recovered was a difficult task.

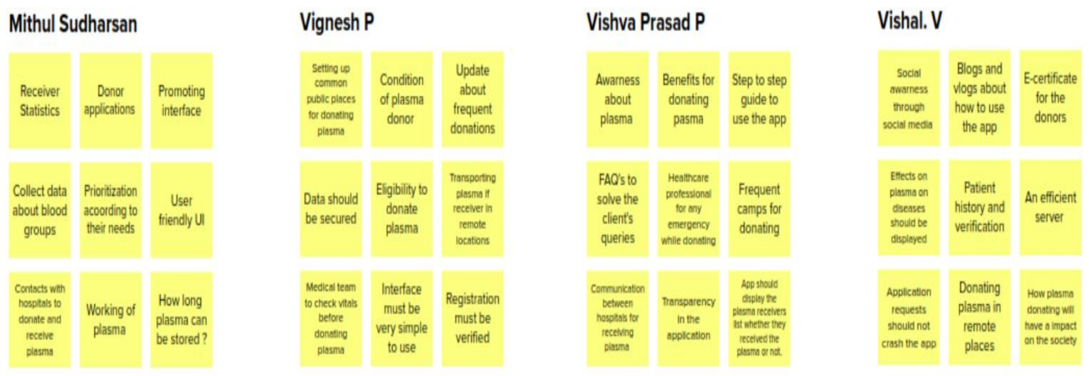
As the plasma therapy was one of the ways to treat the infected patients getting the donor details played a major role.

3. IDEATION AND PROPOSED SYSTEM

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming


Mithul Sudharsan
Receiver Statistics
Donor applications
Promoting interface
Collect data about blood groups
Prioritization according to their needs
User friendly UI
Contacts with hospitals to donate and receive plasma
Working of plasma
How long plasma can be stored ?
Vignesh P
Setting up common public places for donating plasma
Condition of plasma donor
Update about frequent donations
Data should be secured
Eligibility to donate plasma
Transporting plasma if receiver in remote locations
Medical team to check vitals before donating plasma
Interface must be very simple to use
Registration must be verified
Vishva Prasad P
Awarness about plasma
Benefits for donating pasma
Step to step guide to use the app
FAQ's to solve the client's queries
Healthcare professional for any emergency while donating
Frequent camps for donating
Communication between hospitals for receiving plasma
Transparency in the application
App should display the plasma receivers list whether they received the plasma or not.
Vishal. V
Social awarness through social media
Blogs and vlogs about how to use the app
E-certificate for the donors
Effects on plasma on diseases should be displayed
Patient history and verification
An efficient server
Application requests should not crash the app
Donating plasma in remote places
How plasma donating will have a impact on the society

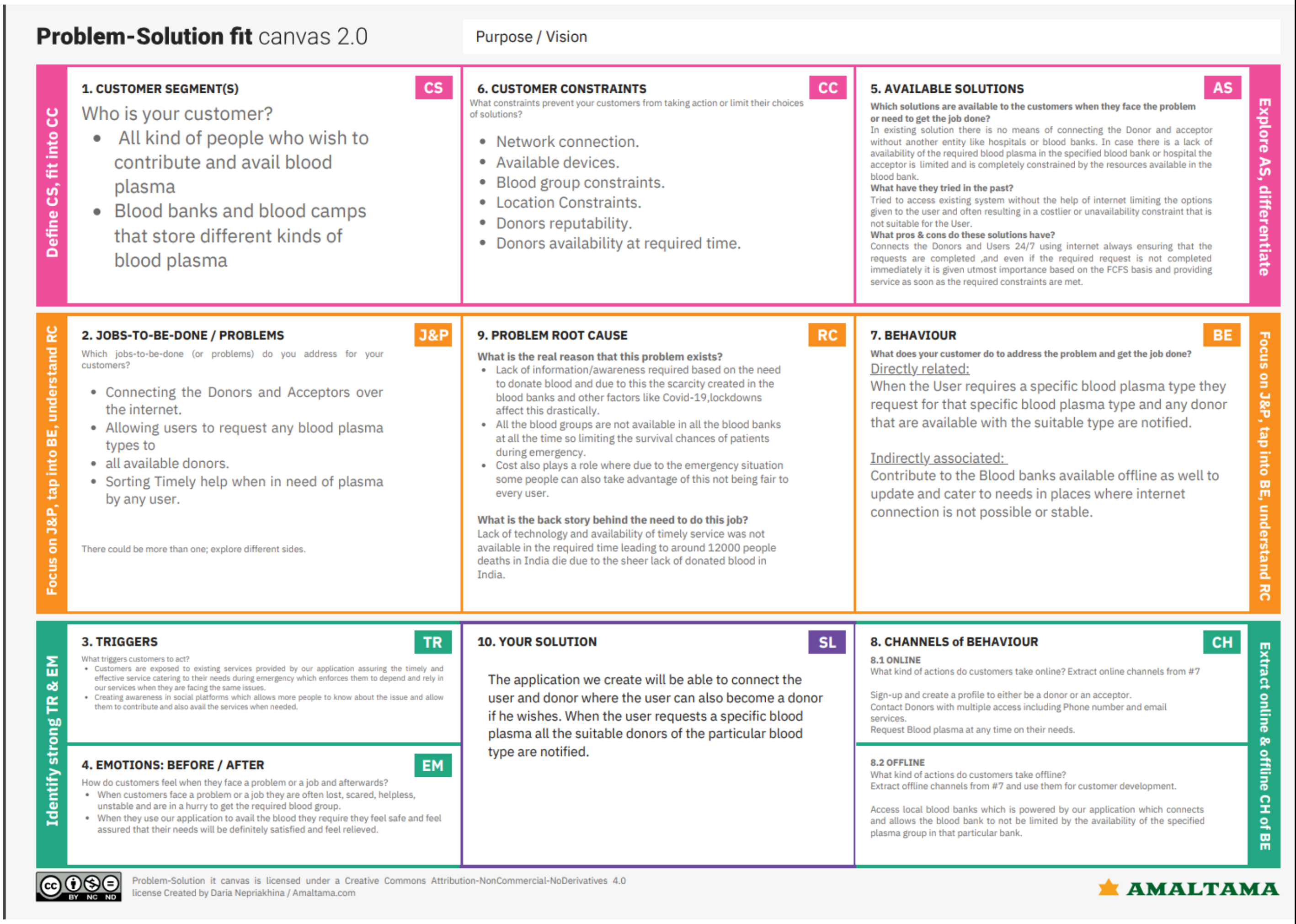
3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The need for plasma surged significantly during the COVID 19 crisis since there were no vaccines available to treat the infected patients. Finding a plasma donor in such a case was extremely difficult, and determining which donors are eligible to donate plasma as well as whether they had previously been infected and have recovered was a difficult effort.

2.	Idea / Solution description	The application we develop will be able to link users and donors, with the user having the option to opt-in as a donor as well. 2. All eligible donors of the requested blood type are alerted when a user asks a particular blood plasma.
3.	Novelty / Uniqueness	1. A user-friendly interface that connects the donor and acceptor in a quick, seamless, and efficient manner. 2. Establishes a Plasma donation community where contributors and recipients benefit equally, fostering a sense of security and assurance when discussing their needs for urgent blood plasma requirements.
4.	Social Impact / Customer Satisfaction	You can gain from using the software properly in a variety of ways, and it also makes management very simple and error-free. Donor tracking, prompt and accurate reports when needed, and centralised data storage with security are all made possible with the aid of the software. Additionally, the software will contribute to customer satisfaction.
5.	Business Model (Revenue Model)	1. The creation of a global community through global connectedness ensures that all emergency requirements are recognised and met at the appropriate moment. 2. Create a reliability factor for each user to guarantee service delivery based on user ratings.

6.	Scalability of the Solution	This programme allows users to find plasma donors while lounging at home rather than searching the entire world. When an emergency occurs, make a plasma request to everyone. The contributor is ready to The donation is disclosed to the donor receiver. The donor may be contacted by the recipient. As a result, donors can use an app to check their eligibility, which also makes it easier to find a compatible donor.

3.4 Problem Statement Fit



Problem-Solution fit canvas 2.0

Purpose / Vision

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

- All kind of people who wish to contribute and avail blood plasma
- Blood banks and blood camps that store different kinds of blood plasma

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions?

- Network connection.
- Available devices.
- Blood group constraints.
- Location Constraints.
- Donors reputability.
- Donors availability at required time.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done?
In existing solution there is no means of connecting the Donor and acceptor without another entity like hospitals or blood banks. In case there is a lack of availability of the required blood plasma in the specified blood bank or hospital the acceptor is limited and is completely constrained by the resources available in the blood bank.
What have they tried in the past?
Tried to access existing system without the help of internet limiting the options given to the user and often resulting in a costlier or unavailability constraint that is not suitable for the User.
What pros & cons do these solutions have?
Connects the Donors and Users 24/7 using internet always ensuring that the requests are completed ,and even if the required request is not completed immediately it is given utmost importance based on the FCFS basis and providing service as soon as the required constraints are met.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

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

Which jobs-to-be-done (or problems) do you address for your customers?

- Connecting the Donors and Acceptors over the internet.
- Allowing users to request any blood plasma types to
- all available donors.
- Sorting Timely help when in need of plasma by any user.

There could be more than one; explore different sides.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?

- Lack of information/awareness required based on the need to donate blood and due to this the scarcity created in the blood banks and other factors like Covid-19,lockdowns affect this drastically.
- All the blood groups are not available in all the blood banks at all the time so limiting the survival chances of patients during emergency.
- Cost also plays a role where due to the emergency situation some people can also take advantage of this not being fair to every user.

What is the back story behind the need to do this job?
Lack of technology and availability of timely service was not available in the required time leading to around 12000 people deaths in India die due to the sheer lack of donated blood in India.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?

Directly related:
When the User requires a specific blood plasma type they request for that specific blood plasma type and any donor that are available with the suitable type are notified.

Indirectly associated:
Contribute to the Blood banks available offline as well to update and cater to needs in places where internet connection is not possible or stable.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act?

- Customers are exposed to existing services provided by our application assuring the timely and effective service catering to their needs during emergency which enforces them to depend and rely in our services when they are facing the same issues.
- Creating awareness in social platforms which allows more people to know about the issue and allow them to contribute and also avail the services when needed.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?

- When customers face a problem or a job they are often lost, scared, helpless, unstable and are in a hurry to get the required blood group.
- When they use our application to avail the blood they require they feel safe and feel assured that their needs will be definitely satisfied and feel relieved.

10. YOUR SOLUTION SL

The application we create will be able to connect the user and donor where the user can also become a donor if he wishes. When the user requests a specific blood plasma all the suitable donors of the particular blood type are notified.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

Sign-up and create a profile to either be a donor or an acceptor.
Contact Donors with multiple access including Phone number and email services.
Request Blood plasma at any time on their needs.

8.2 OFFLINE
What kind of actions do customers take offline?
Extract offline channels from #7 and use them for customer development.

Access local blood banks which is powered by our application which connects and allows the blood bank to not be limited by the availability of the specified plasma group in that particular bank.

Extract online & offline CH of BE

AMALTAMA

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**Access Website**	Software developers should be able to access web applications on a computer using a browser or something similar.
FR-2	**Software operator Registration**	The online application should allow the software operator to register. User name, gender, blood/plasma group, location, and contact information are required from the donor software operator.
FR-3	**Login/logout/update details**	The login information will be stored on thedatabase for future use.
FR-4	**Search for donor**	A list of search results can be seen. Each item on the list corresponds to a particular donor, complete with donor information.
FR-5	**User plasma request**	By completing the request form on the page, users can submit a request to donate plasma. They will receive an email once the request has been submitted.
FR-6	**View distribution details**	The plasma bank should have access to the distribution

4.2 Non-functional Requirements:

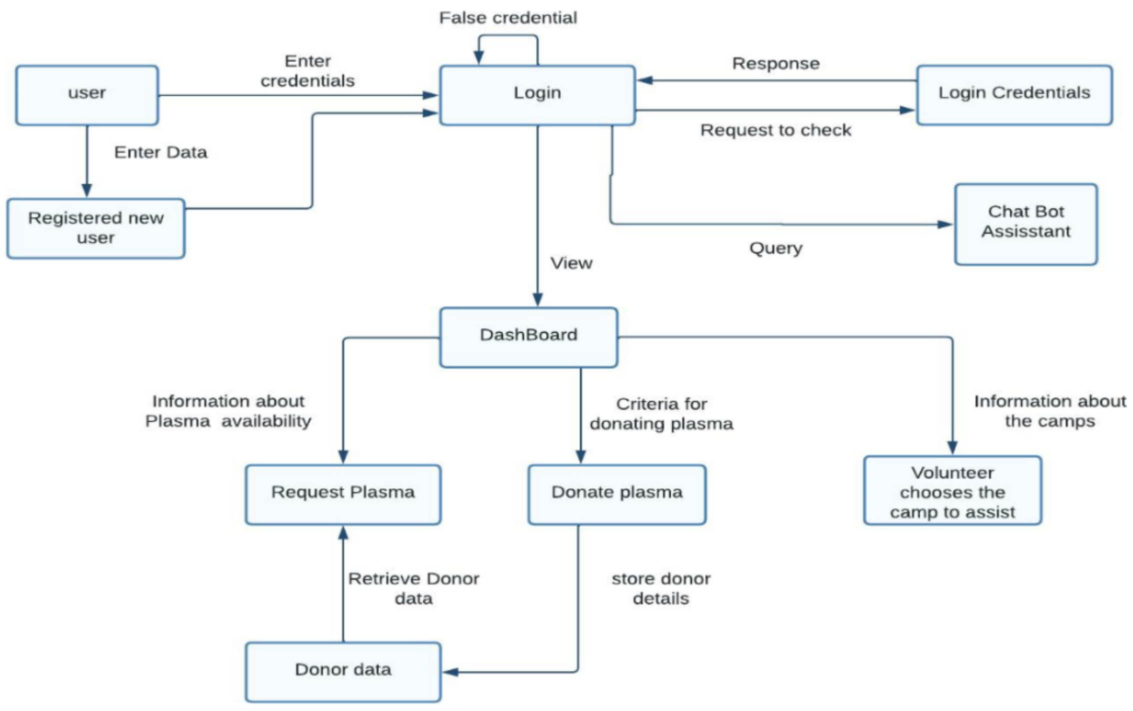
Following are the non-functional requirements of the proposed solution.

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	The user interface of the plasma donor application needs to be attractive and functional.
NFR-2	**Security**	Proper user names and passwords must be used to secure the plasma donor application.
NFR-3	**Reliability**	The plasma donor application should work properly,even when faults occur.
NFR-4	**Performance**	The plasma donor application must perform well
		in different scenarios.
NFR-5	**Availability**	The plasma donor application must available 24 hours a day with no bandwidth issues.

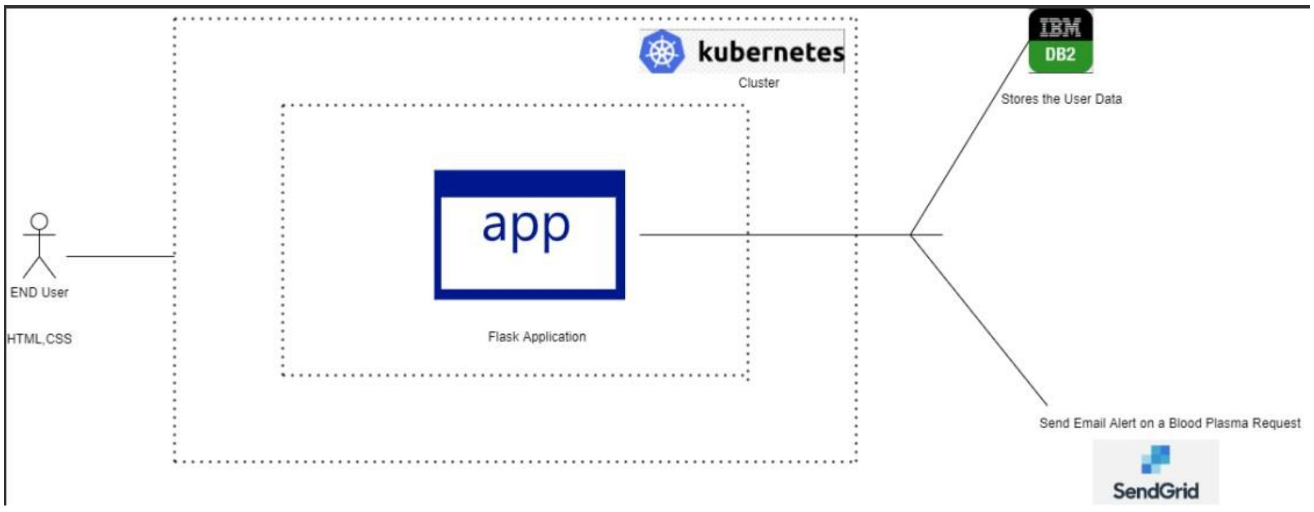
NFR-6	**Scalability**	The performance and cost of the plasma donor application should be able to change in response to modifications in the requirements for application and system processing.

5.PROJECT DESIGN

5.1 Data Flow Diagram:



5.2 Solution and Technical Architecture :



5.3 User Stories:

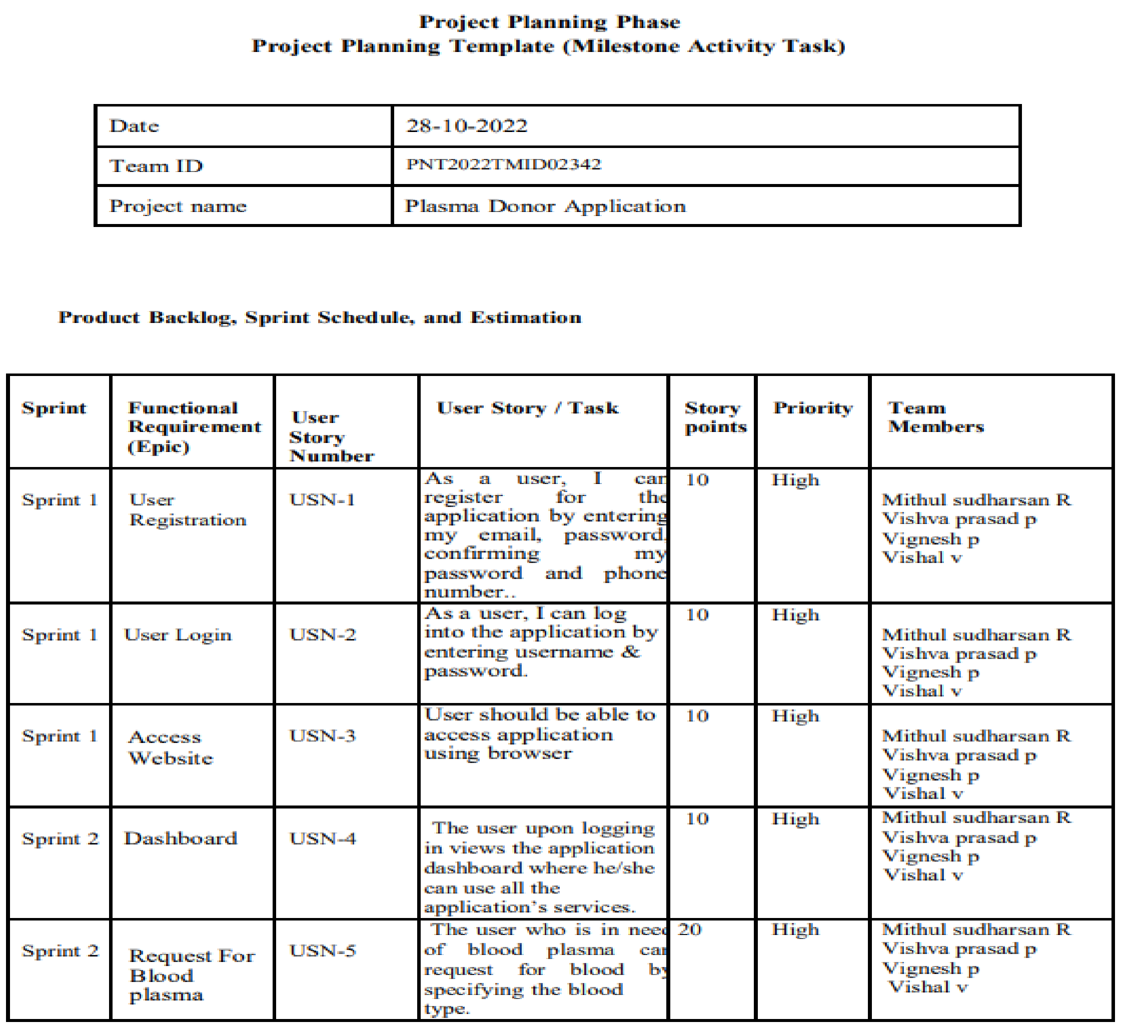
User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	I can sign up for the application as a user by providing my email address, a password, and a password confirmation.	I can access my account / dashboard	High	Sprint-1
		USN-2	After registering for the application, I as a user will receive a confirmation email.	I can get a confirmation email and confirm it.	High	Sprint-1
		USN-3	I can sign up for the application as a user using Gmail.	Through Gmail, I may receive confirmation emails.	Medium	Sprint-1
	Login	USN-4	I can access the application as a user by providing my email address and password.	I can access my user profile and look at the information in the dashboard.	High	Sprint-1
	Dashboard	USN-5	I can send the appropriate requests to donate and obtain plasma as a user.	I can respond to all questions about our application.	High	Sprint-1
Customer (Web user)	Login	USN-6	I can sign up and use the application as a user and view my profile by providing my email address and password.	I can access my user profile and look at the information in the dashboard.	High	Sprint-1
	Dashboard	USN-7	I can send the appropriate requests to donate and obtain plasma as a user..	I can get the right notifications by email.	High	Sprint-1
Customer Care Executive	Application	USN-8	I can try to respond to users' inquiries and issues as a customer service representative.	I can see their issues and respond to their inquiries.	Medium	Sprint-2
Administrator	Application	USN-9	As a website administrator, I can assist with user-facing features like a website's look, navigation, and media usage.	I have the ability to modify the navigation and design to make them more user-friendly.	Medium	Sprint-3
		USN-10	I can work with the technical aspect of websites as an administrator..	I can assist with tasks like server programming, setting up web hosts, ensuring users have access, and debugging problems.	Medium	Sprint-1
Chatbot	Dashboard	USN-11	Additionally, the customer service representative's chatbot might attempt to respond to users' queries and concerns.	I can respond to all questions about our application.	Medium	Sprint-3

6.PROJECT PLANNING AND SCHEDULING

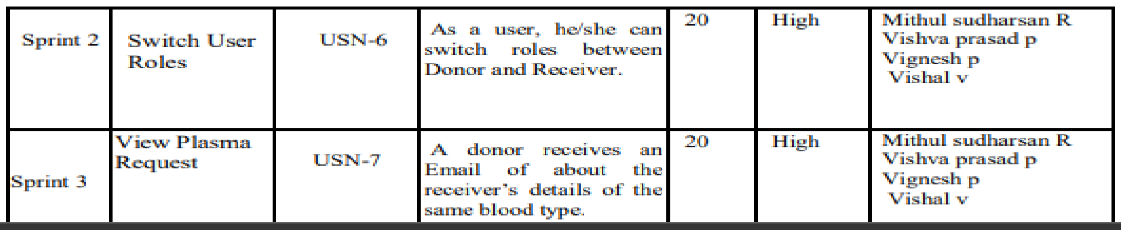
6.1 Sprint Planning & Estimation



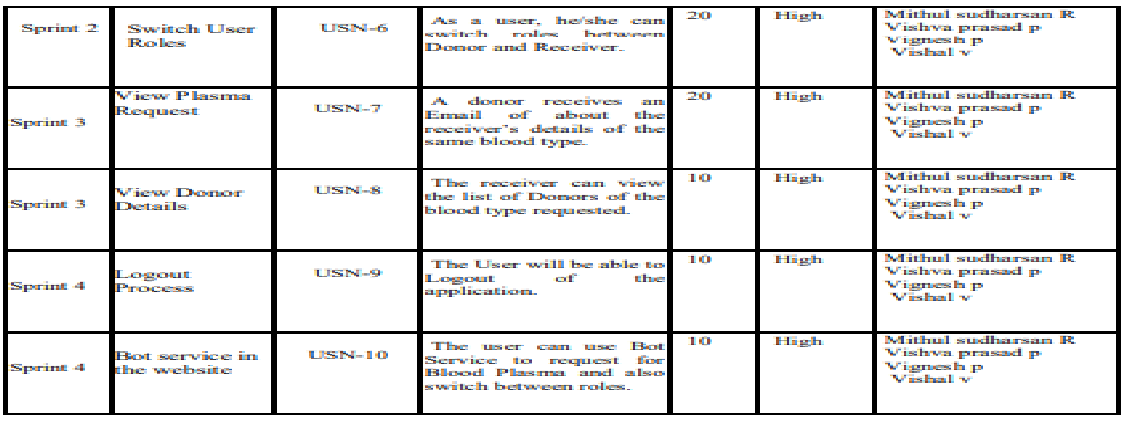
Project Planning Phase
Project Planning Template (Milestone Activity Task)

Date	28-10-2022
Team ID	PNT2022TMID02342
Project name	Plasma Donor Application

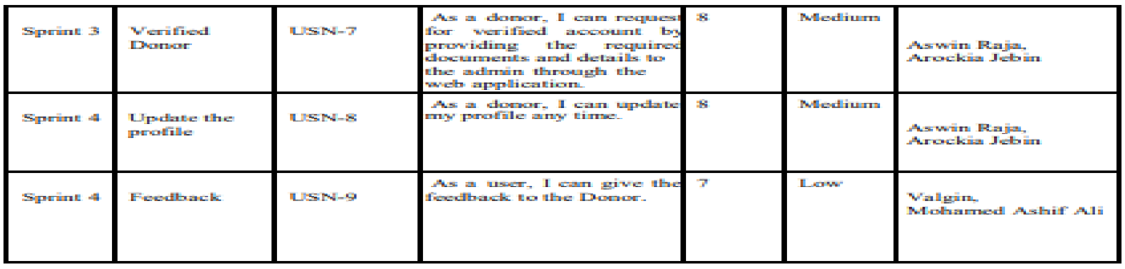
Product Backlog, Sprint Schedule, and Estimation



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story points	Priority	Team Members
Sprint 1	User Registration	USN-1	As a user, I can register for the application by entering my email, password, confirming my password and phone number..	10	High	Mithul sudharsan R Vishva prasad p Vignesh p Vishal v
Sprint 1	User Login	USN-2	As a user, I can log into the application by entering username & password.	10	High	Mithul sudharsan R Vishva prasad p Vignesh p Vishal v
Sprint 1	Access Website	USN-3	User should be able to access application using browser	10	High	Mithul sudharsan R Vishva prasad p Vignesh p Vishal v
Sprint 2	Dashboard	USN-4	The user upon logging in views the application dashboard where he/she can use all the application's services.	10	High	Mithul sudharsan R Vishva prasad p Vignesh p Vishal v
Sprint 2	Request For Blood plasma	USN-5	The user who is in need of blood plasma can request for blood by specifying the blood type.	20	High	Mithul sudharsan R Vishva prasad p Vignesh p Vishal v
Sprint 2	Switch User Roles	USN-6	As a user, he/she can switch roles between Donor and Receiver.	20	High	Mithul sudharsan R Vishva prasad p Vignesh p Vishal v
Sprint 3	View Plasma Request	USN-7	A donor receives an Email of about the receiver's details of the same blood type.	20	High	Mithul sudharsan R Vishva prasad p Vignesh p Vishal v



Sprint 2	Switch User Roles	USN-6	As a user, he/she can switch roles between Donor and Receiver.	20	High	Mithul sudharsan R Vishva prasad p Vignesh p Vishal v
Sprint 3	View Plasma Request	USN-7	A donor receives an Email of about the receiver's details of the same blood type.	20	High	Mithul sudharsan R Vishva prasad p Vignesh p Vishal v
Sprint 3	View Donor Details	USN-8	The receiver can view the list of Donors of the blood type requested.	10	High	Mithul sudharsan R Vishva prasad p Vignesh p Vishal v
Sprint 4	Logout Process	USN-9	The User will be able to Logout of the application.	10	High	Mithul sudharsan R Vishva prasad p Vignesh p Vishal v
Sprint 4	Bot service in the website	USN-10	The user can use Bot Service to request for Blood Plasma and also switch between roles.	10	High	Mithul sudharsan R Vishva prasad p Vignesh p Vishal v

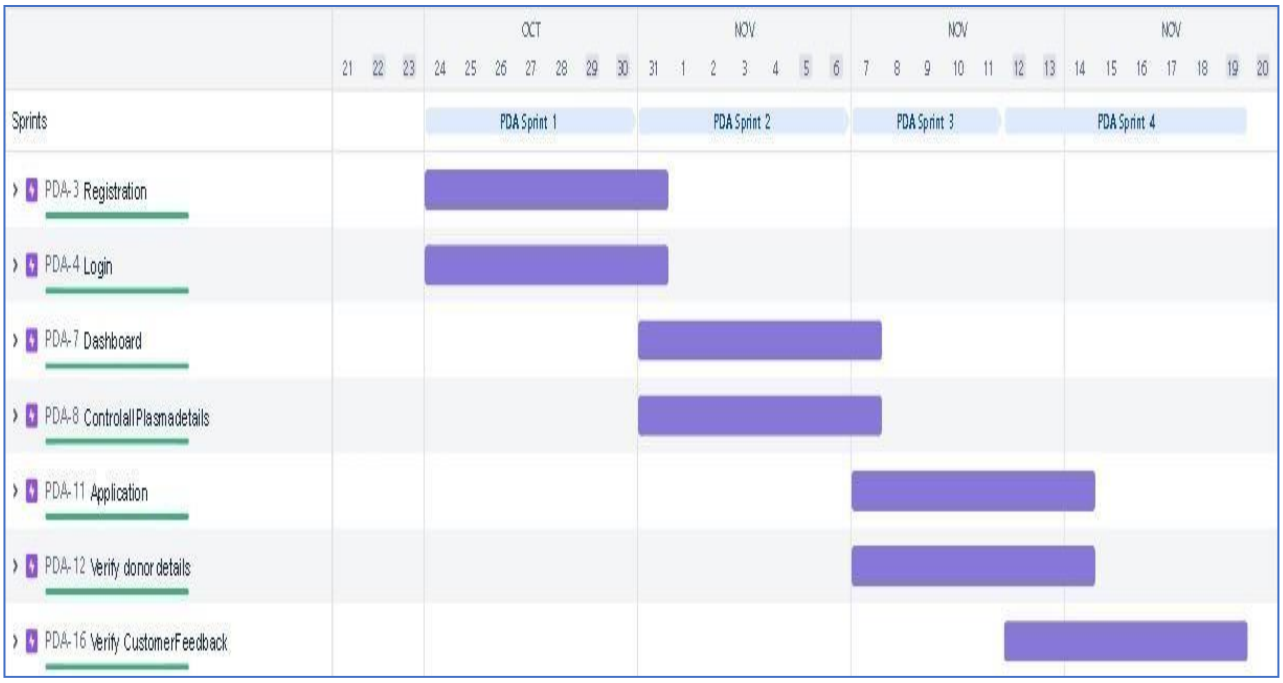


Sprint 3	Verified Donor	USN-7	As a donor, I can request for verified account by providing the required documents and details to the admin through the web application.	8	Medium	Aswin Raja, Arockia Jebin
Sprint 4	Update the profile	USN-8	As a donor, I can update my profile any time.	8	Medium	Aswin Raja, Arockia Jebin
Sprint 4	Feedback	USN-9	As a user, I can give the feedback to the Donor.	7	Low	Valgin, Mohamed Ashif Ali

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	30	8 days	22-10-2022	29-10-2022	30	29-10-2022
Sprint 2	50	8 days	29-11-2022	05-11-2022	50	05-11-2022
Sprint 3	30	8 days	05-11-2022	12-11-2022	30	12-11-2022
Sprint 4	20	8 days	12-11-2022	19-11-2022	20	19-11-2022

6.3 Reports from JIRA



7. CODING & SOLUTIONING

7.1 Feature 1:

Python

- Python is a popular, interpreted, object-oriented, high-level, dynamically semantic, and general-purpose programming language. Whether they are aware of it or not, consumers use many Python-powered devices on a regular basis.

- On February 20, 1991, Guido van Rossum released the first version of Python.

- Several languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, the Unix shell, and other scripting languages are the ancestors of Python.

- Python possesses a copyright. Python source code is now accessible under the GNU General Public License, just like Perl (GPL)

- It is simple to learn - learning Python takes less time than learning many other languages, making it possible to get started on programming right away.

- It is simple to use for creating new software, and Python frequently makes it possible to write code more quickly.

- Python is free, open, and multiplatform; not all languages can make that claim. It is simple to get, install, and deploy.

- Programming abilities are necessary if you wish to rise to increasingly complex and lucrative software development and engineering roles and equip you for careers in practically any industry.

- Currently, a core development team at the institute is responsible for maintaining Python, though Guido van Rossum continues to play a key role in guiding its development.

7.2 Feature 2:

Flask

- Python-based Flask is a microweb framework. Due to the fact that it doesn't require any specific tools or libraries, it is categorised as a microframework.

- It lacks any components where pre-existing third-party libraries already provide common functions, such as a database abstraction layer, form validation, or other components. However, Flask allows for extensions that may be used to add application functionalities just like they were built into the core of Flask.

- There are extensions for object-relational mappers, form validation, upload handling, different open authentication technologies, and a number of tools associated with common frameworks.

- Pinterest and LinkedIn are two programmes that utilise the Flask framework.

7.3 Database Scheme

IBM Db2

- A database product from IBM is called DB2.
- A relational database management system is what it is. DB2 is made to efficiently store, analyze, and retrieve the data.
- XML-based non-relational structures and support for Object-Oriented features are added to the DB2 product.
- Offer a massively parallel processing (MPP) architecture that simultaneously uses Apache Spark, HBase, and Hive for the best possible analytic performance.
- provides federation capabilities, high performance, and low latency support for ad-hoc and complex queries. Advanced row and column security is possible thanks to the understanding of dialects from other vendors and a variety of Oracle, IBM Db2, and IBM Netezza products.

Kubernates

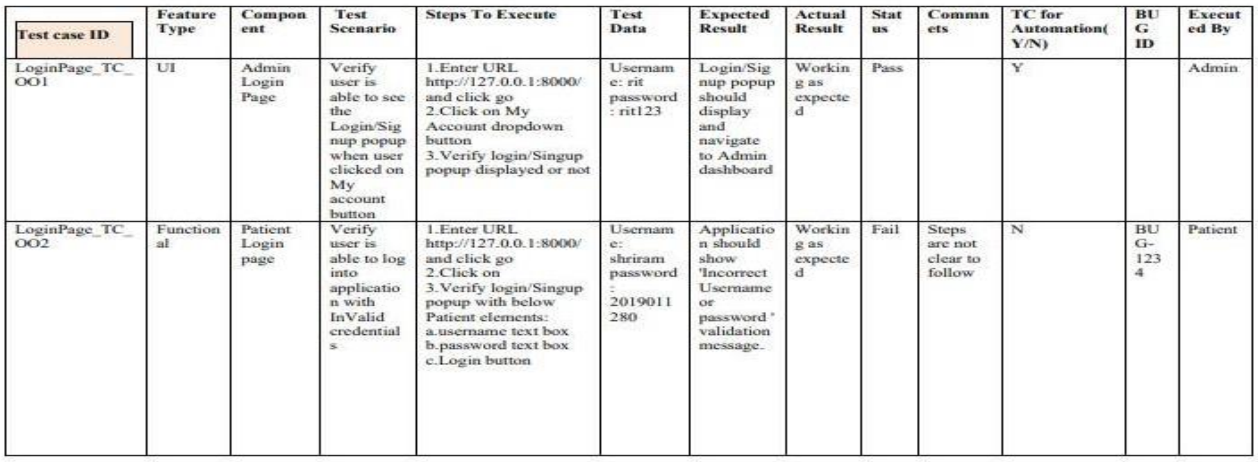
- Kubernetes is also referred to as "k8s."
- A flexible, portable, and open-source platform called Kubernetes was created by Google in 2014.
- Its primary usage is to automatically deploy, scale, and run container-based applications across a cluster of nodes.
- In a variety of physical, virtual, and cloud contexts, Kubernetes aids in managing containerized applications.
- To reliably deliver complex applications running on clusters of hundreds to thousands of individual servers, Google Kubernetes is a highly flexible container tool.
- The Linux kernel used for distributed systems is known as Kubernetes.

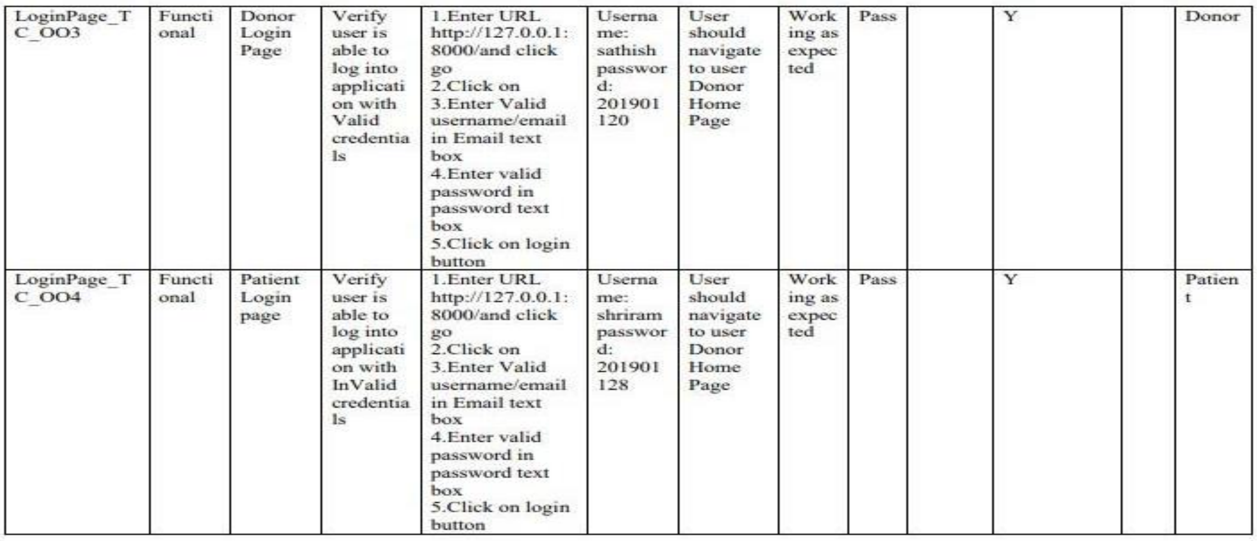
8.TESTING

8.1 Test case

- It is the process of testing software with the goal of ensuring that it satisfies user expectations and meets requirements without failing in an unacceptable way.

- Different test types exist. Every test type responds to a distinct testing requirement

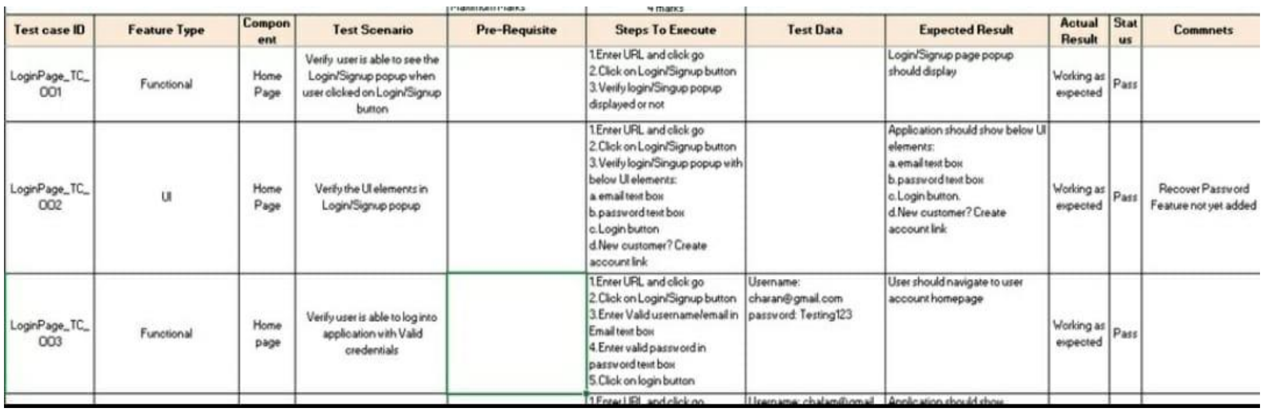




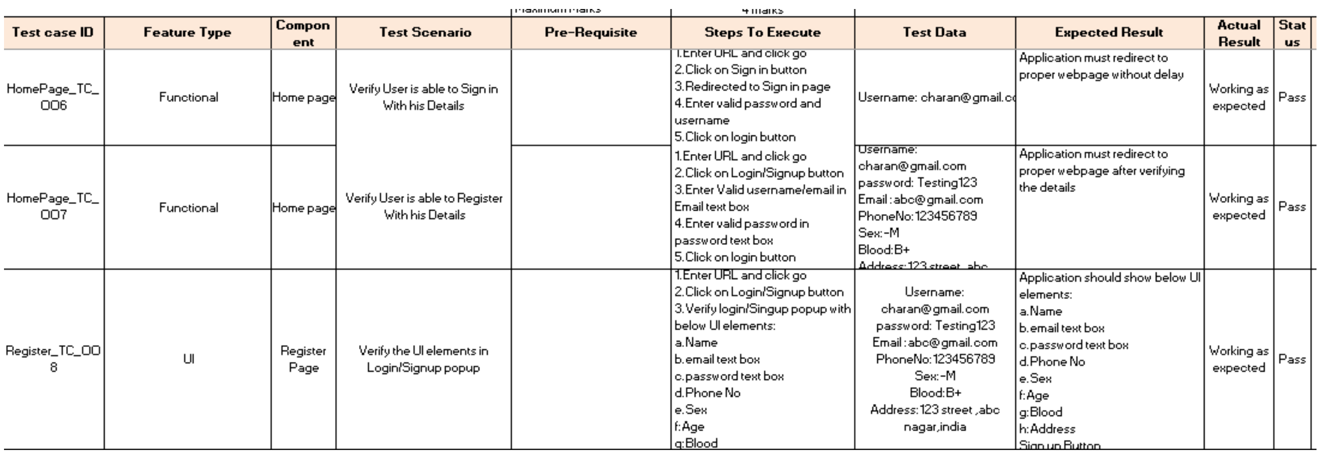
Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_OO1	UI	Admin Login Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	1.Enter URL http://127.0.0.1:8000/ and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	Username: rit password : rit123	Login/Signup popup should display and navigate to Admin dashboard	Working as expected	Pass		Y		Admin
LoginPage_TC_OO2	Functional	Patient Login page	Verify user is able to log into application with InValid credentials	1.Enter URL http://127.0.0.1:8000/ and click go 2.Click on 3.Verify login/Singup popup with below Patient elements: a.username text box b.password text box c.Login button	Username: shriram password : 2019011280	Application should show 'Incorrect Username or password ' validation message.	Working as expected	Fail	Steps are not clear to follow	N	BUG-1234	Patient
LoginPage_TC_OO3	Functional	Donor Login Page	Verify user is able to log into application with Valid credentials	1.Enter URL http://127.0.0.1:8000/and click go 2.Click on 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: sathish password: 201901120	User should navigate to user Donor Home Page	Working as expected	Pass		Y		Donor
LoginPage_TC_OO4	Functional	Patient Login page	Verify user is able to log into application with InValid credentials	1.Enter URL http://127.0.0.1:8000/and click go 2.Click on 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: shriram password: 201901128	User should navigate to user Donor Home Page	Working as expected	Pass		Y		Patient

.

8.2 User Acceptance Testing



Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets
LoginPage_TC_001	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on Login/Signup button		1.Enter URL and click go 2.Click on Login/Signup button 3.Verify login/Singup popup displayed or not		Login/Signup page popup should display	Working as expected	Pass	
LoginPage_TC_002	UI	Home Page	Verify the UI elements in Login/Signup popup		1.Enter URL and click go 2.Click on Login/Signup button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link		Application should show below UI elements: a.email text box b.password text box c.Login button. d.New customer? Create account link	Working as expected	Pass	Recover Password Feature not yet added
LoginPage_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials		1.Enter URL and click go 2.Click on Login/Signup button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: charan@gmail.com password: Testing123	User should navigate to user account homepage	Working as expected	Pass	
					1.Enter URL and click go	Username: chalam@gmail	Application should show			



Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status
HomePage_TC_006	Functional	Home page	Verify User is able to Sign in With his Details		1.Enter URL and click go 2.Click on Sign in button 3.Redirected to Sign in page 4.Enter valid password and username 5.Click on login button	Username: charan@gmail.co	Application must redirect to proper webpage without delay	Working as expected	Pass
HomePage_TC_007	Functional	Home page	Verify User is able to Register With his Details		1.Enter URL and click go 2.Click on Login/Signup button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: charan@gmail.com password: Testing123 Email : abc@gmail.com PhoneNo: 123456789 Sex:-M Blood:B+ Address:123 street ,abc	Application must redirect to proper webpage after verifying the details	Working as expected	Pass
Register_TC_008	UI	Register Page	Verify the UI elements in Login/Signup popup		1.Enter URL and click go 2.Click on Login/Signup button 3.Verify login/Singup popup with below UI elements: a.Name b.email text box c.password text box d.Phone No e.Sex f:Age g:Blood	Username: charan@gmail.com password: Testing123 Email : abc@gmail.com PhoneNo: 123456789 Sex:-M Blood:B+ Address: 123 street ,abc nagar,india	Application should show below UI elements: a.Name b.email text box c.password text box d.Phone No e.Sex f:Age g:Blood h:Address Sign up Button	Working as expected	Pass

1. Purpose of Document

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

2. Defect Analysis

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

Resolution	**Severity 1**	**Severity 2**	**Severity 3**	**Severity 4**	**Sub total**
By Design	8	4	2	3	17
Duplicate	1	0	2	1	4
External	2	3	0	1	6

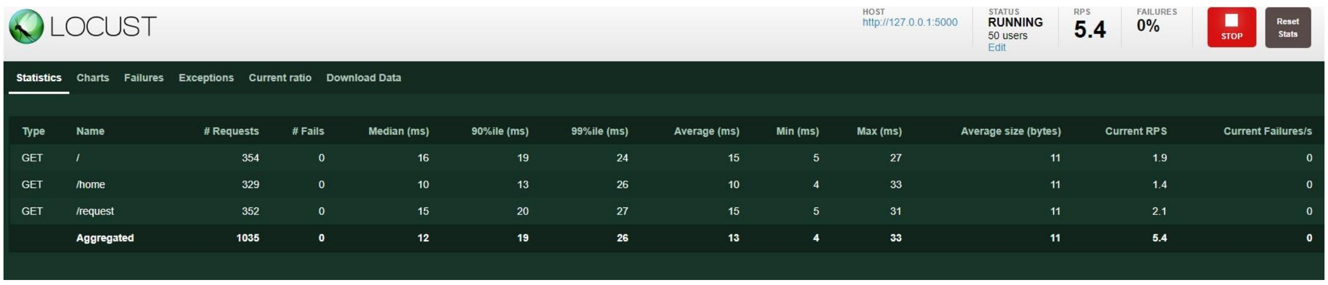
Fixed	10	2	5	18	35
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	3	2	1	6
Totals	21	12	13	25	71

3. Test Case Analysis

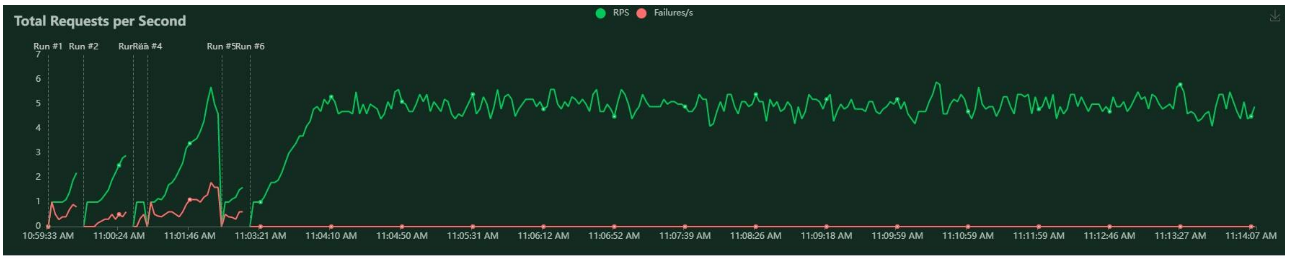
This report shows the number of test cases that have passed, failed, and untested

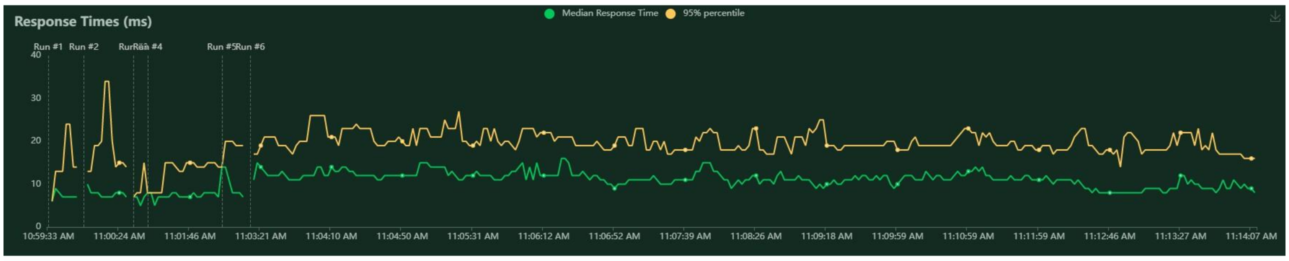
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	8	0	0	8
Client Application	50	0	0	50
Security	2	0	0	2
Outsource Shipping	3	0	0	3

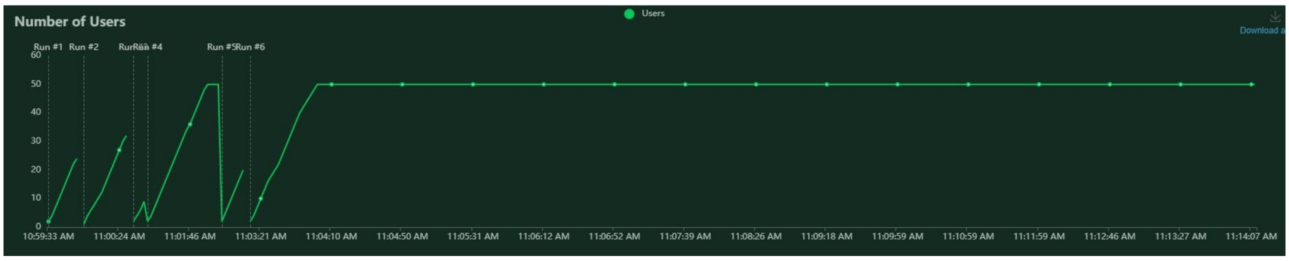
Exception Reporting	10	0	0	10
Final Report Output	6	0	0	6
Version Control	3	0	0	3



Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/	354	0	16	19	24	15	5	27	11	1.9	0
GET	/home	329	0	10	13	26	10	4	33	11	1.4	0
GET	/request	352	0	15	20	27	15	5	31	11	2.1	0
	Aggregated	**1035**	**0**	**12**	**19**	**26**	**13**	**4**	**33**	**11**	**5.4**	**0**







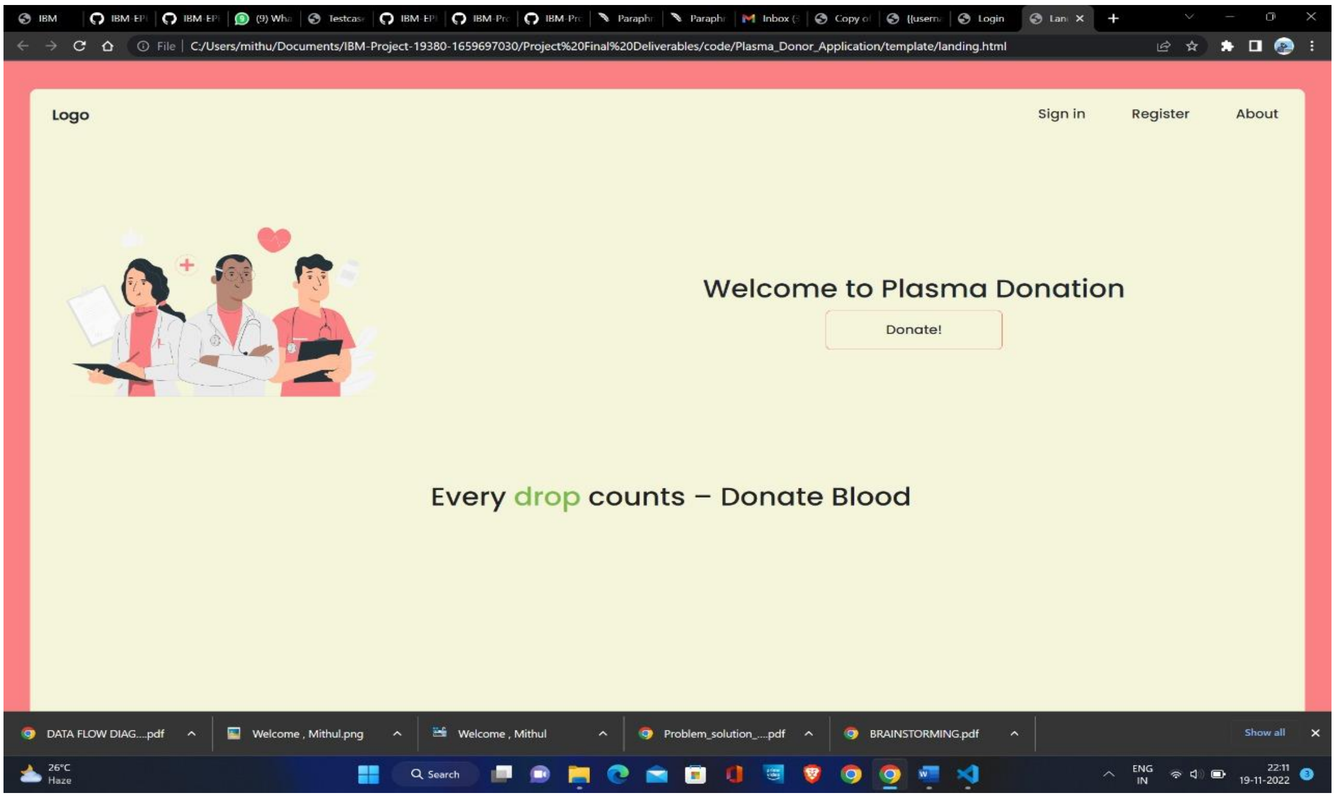
9.RESULTS

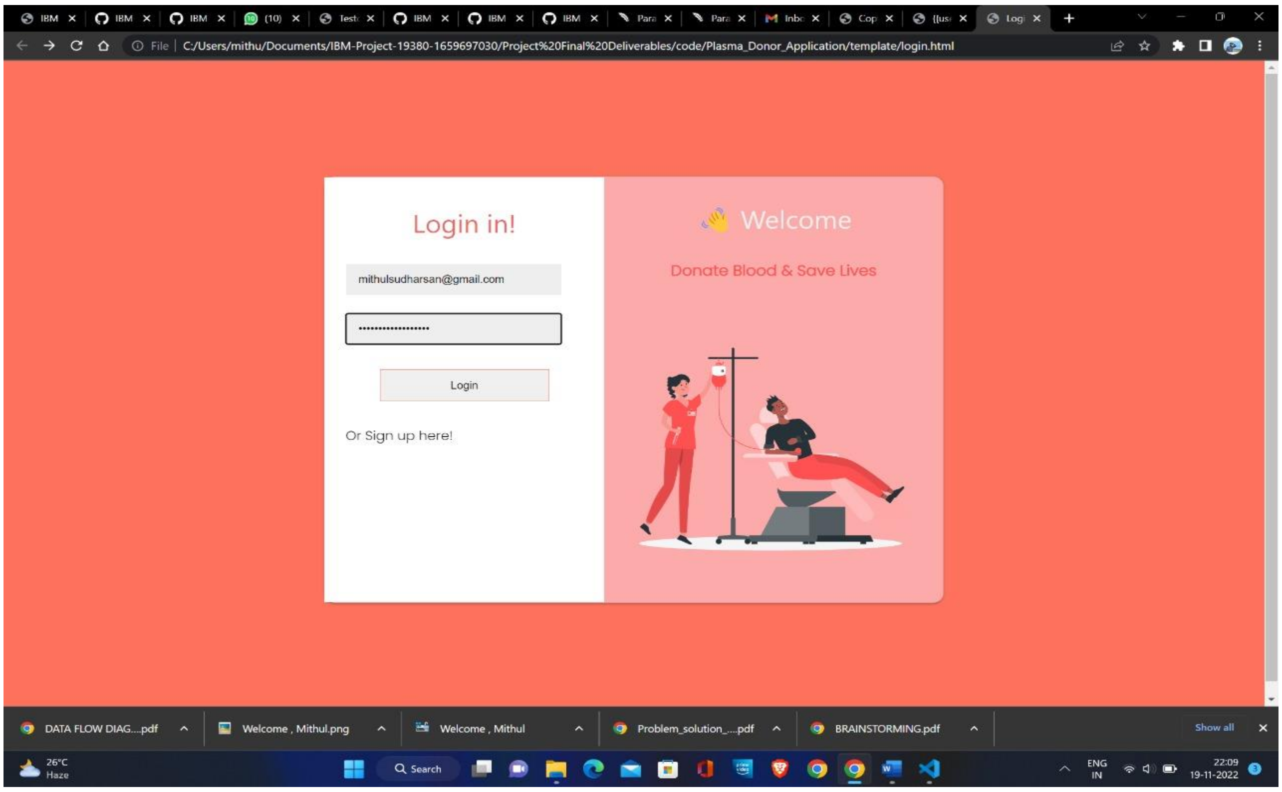
9.1 Performance Metrics

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

OUTPUT SCREENS

Landing page


C:/Users/mithu/Documents/IBM-Project-19380-1659697030/Project%20Final%20Deliverables/code/Plasma_Donor_Application/template/landing.html
Logo
Sign in
Register
About
Welcome to Plasma Donation
Donate!
Every drop counts – Donate Blood

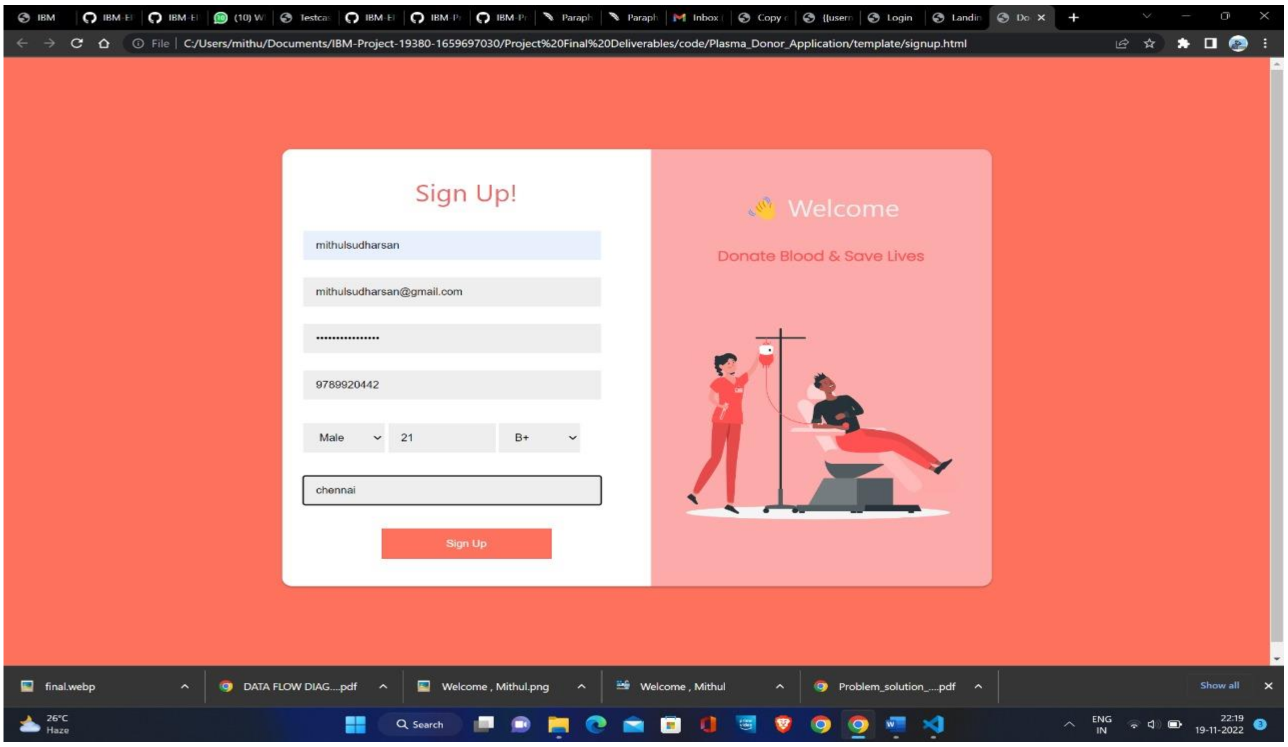

C:/Users/mithu/Documents/IBM-Project-19380-1659697030/Project%20Final%20Deliverables/code/Plasma_Donor_Application/template/login.html

Login in!

mithulsudharsan@gmail.com

••••••••••••••••••

Register Page


C:/Users/mithu/Documents/IBM-Project-19380-1659697030/Project%20Final%20Deliverables/code/Plasma_Donor_Application/template/signup.html

Sign Up!

mithulsudharsan

mithulsudharsan@gmail.com

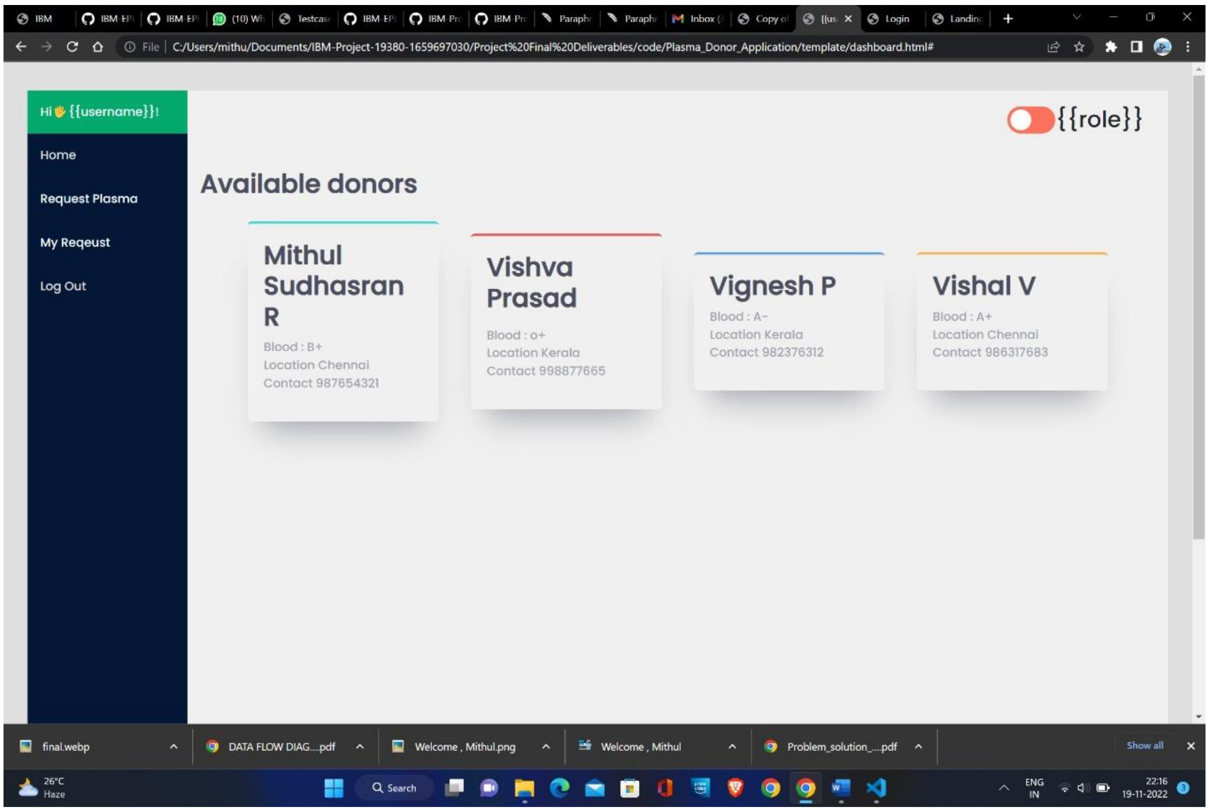
••••••••••••••••

9789920442

Male | 21 | B+

chennai

Dashboard



Available donors

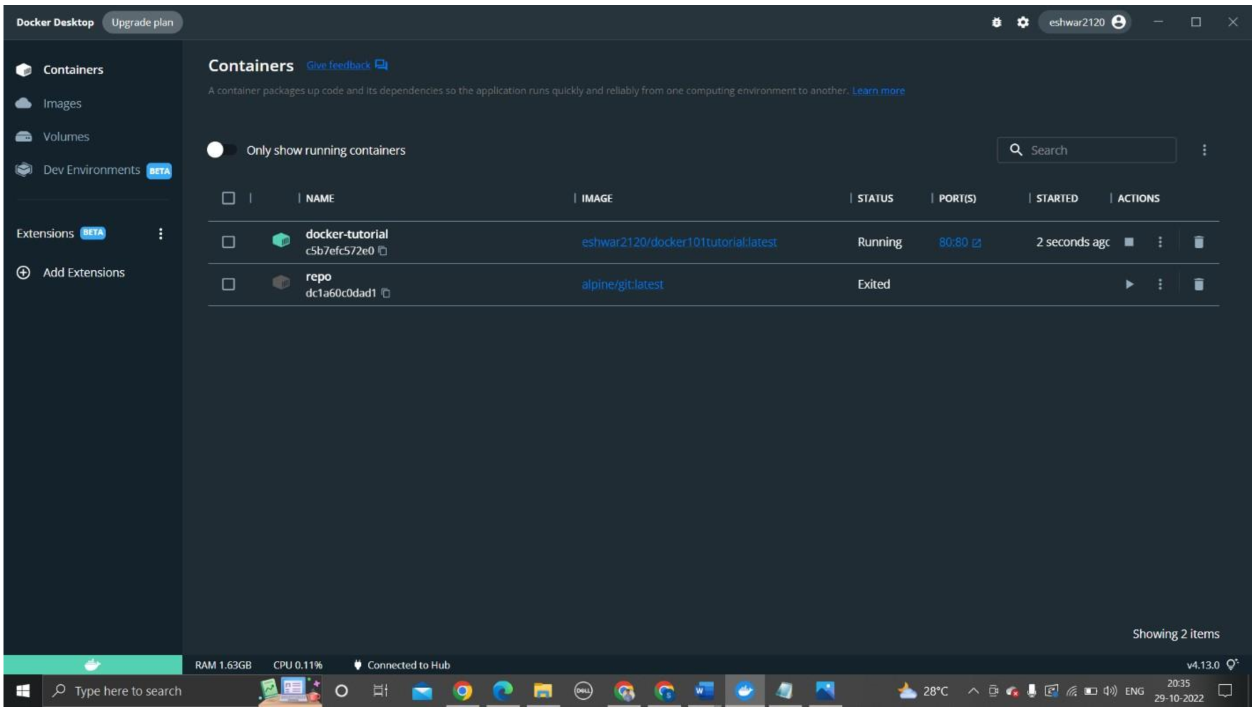
Mithul Sudhasran R
Blood : B+
Location Chennai
Contact 987654321

Vishva Prasad
Blood : o+
Location Kerala
Contact 998877665

Vignesh P
Blood : A-
Location Kerala
Contact 982376312

Vishal V
Blood : A+
Location Chennai
Contact 986317683

Dockerize the app

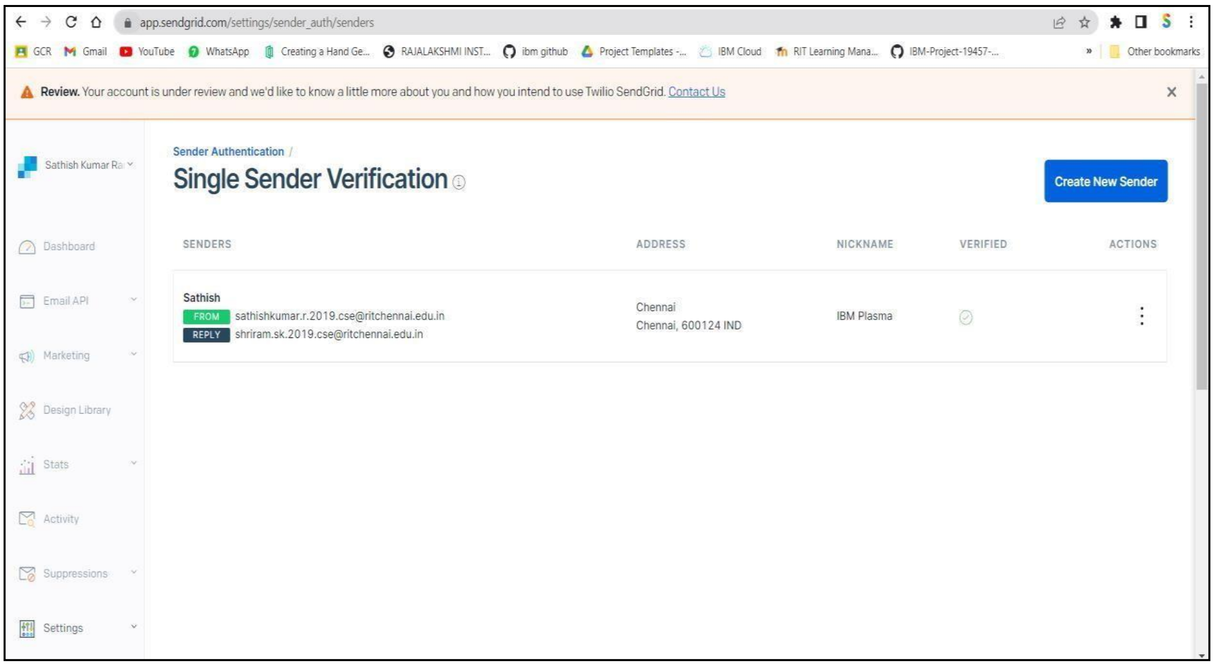


Containers

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

NAME	IMAGE	STATUS	PORT(S)	STARTED
docker-tutorial c5b7efc572e0	eshwar2120/docker101tutorial:latest	Running	80:80	2 seconds agc
repo dc1a60c0dad1	alpine/git:latest	Exited		

SEND GRID



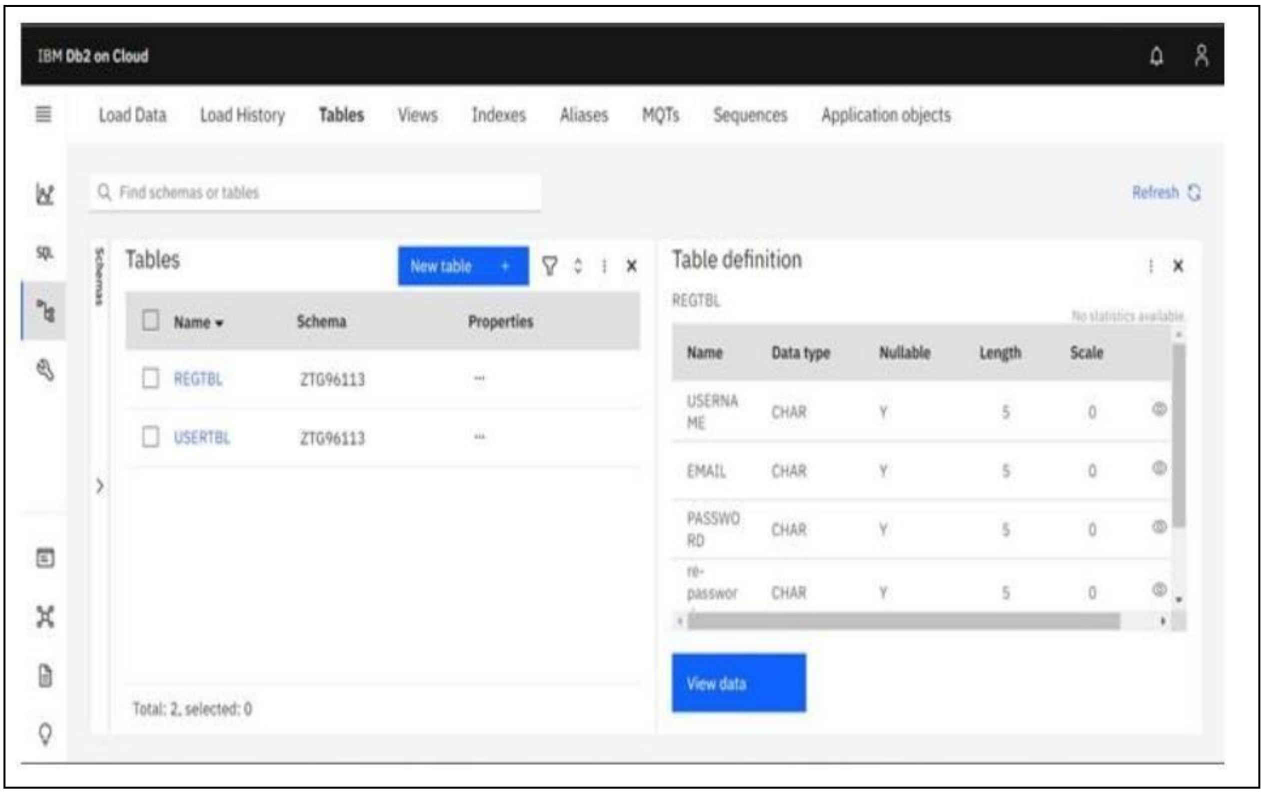
app.sendgrid.com/settings/sender_auth/senders
Review. Your account is under review and we'd like to know a little more about you and how you intend to use Twilio SendGrid. Contact Us
Dashboard
Email API
Marketing
Design Library
Stats
Activity
Suppressions
Settings
Create New Sender

Sender Authentication /

Single Sender Verification

SENDERS	ADDRESS	NICKNAME	VERIFIED	ACTIONS
Sathish FROM sathishkumar.r.2019.cse@ritchennai.edu.in REPLY shriram.sk.2019.cse@ritchennai.edu.in	Chennai Chennai, 600124 IND	IBM Plasma	✓	⋮

IBM DB2



Tables

Name	Schema	Properties
REGTBL	ZTG96113	···
USERTBL	ZTG96113	···

Total: 2, selected: 0

Table definition

REGTBL

No statistics available

Name	Data type	Nullable	Length	Scale
USERNAME	CHAR	Y	5	0
EMAIL	CHAR	Y	5	0
PASSWORD	CHAR	Y	5	0
re-passwor	CHAR	Y	5	0

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- Speed: When compared to manual register keeping, this website is quick and delivers excellent accuracy.
- Less maintenance is needed, and it is incredibly user-friendly and simple to grasp. It is simple to use and open to everyone.
- Fast Results: Depending on its availability, it would assist you in finding plasma donors quickly.

DISADVANTAGES:

- The website would need to be connected to the internet in order to function.
- Auto-verification is not possible because only legitimate users can be confirmed.

11. CONCLUSION

- Using the plasma donor website, which is housed on the IBM Cloud platform, afflicted individuals can discover plasma doors in an effective manner.

- To guarantee the efficient running of the website. To ensure that the activities are going smoothly, I have hosted the website in an IBM Db2 and Kubernates Cluster. In order to deploy the application, IBM Db2 service is employed together with cloud lambda function.

12. FUTURE SCOPE

- Making the user interface (UI) more user-friendly, which will make it easier for many users to access the website and ensure that there are plenty of new plasma donors who can join the community.

- Using an elastic load balancer makes it possible to manage multiple requests at once, maintaining website uptime with minimal downtime.

13.APPENDIXES

13.1 SAMPLE SOURCE CODE: DONOR

MAIN.py

```
from flask import Flask, redirect, url_for, render_template, request, make_response, jsonify, request

import ibm_db

from flask import request

import json


conn = ibm_db.connect(

    "DATABASE=bludb;HOSTNAME=764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=32536;SECURITY=SSL;SSLServerCertificate=abc.crt;UID=gnq12618;PWD=0glS4tFaR2ciK8fB",

    '', '')

print(conn)

print("connection successful...")

app = Flask(__name__)

import os

from sendgrid import SendGridAPIClient

from sendgrid.helpers.mail import Mail



@app.route('/')

def home():

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

```
@app.route('/home')
def dash():
    return render_template("dashboard.html")



@app.route('/login', methods=['POST', 'GET'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        sql = "select * from user where username=? and password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.bind_param(stmt, 2, password)
        ibm_db.execute(stmt)
        dic = ibm_db.fetch_assoc(stmt)
        print(dic)
        role = str()
        requests = []
        if dic:
            role = dic['ROLE']
            # sql = "select * from user where blood_group=?"
            # stmt = ibm_db.prepare(conn, sql)
            # ibm_db.bind_param(stmt, 1, username)
            # ibm_db.execute(stmt)
            # dic = ibm_db.fetch_assoc(stmt)
            # while dic != False:
            #     single_request = {
            #         'name': dic['NAME'],
            #         'age': dic['AGE'],
            #         'sex': dic['SEX'],
            #         'blood_type': dic['BLOOD_TYPE']
            #     }
            #     print(single_request)
            #     requests.append(single_request)
            #     dic = ibm_db.fetch_assoc(stmt)
            return render_template('dashboard.html', username=username, role=role)


        else:
            return redirect(url_for('login'))
        return redirect(url_for('home'))
    elif request.method == 'GET':
        return render_template('login.html')



@app.route('/signup', methods=['POST', 'GET'])
def signup():
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        roll_no = request.form['roll_no']
        sex = request.form['sex']
        age = request.form['age']
        address =  request.form['address']
        blood_group = request.form['blood_group']
        sql = "insert into user values(?,?,?,?,?,?,?,?,?)"
        prep_stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(prep_stmt, 1, username)
        ibm_db.bind_param(prep_stmt, 2, email)
        ibm_db.bind_param(prep_stmt, 3, password)
        ibm_db.bind_param(prep_stmt, 4, roll_no)
        ibm_db.bind_param(prep_stmt, 5, sex)
        ibm_db.bind_param(prep_stmt, 6, age)
        ibm_db.bind_param(prep_stmt, 7, "USER")
        ibm_db.bind_param(prep_stmt, 8, address)
        ibm_db.bind_param(prep_stmt, 9, blood_group)
        ibm_db.execute(prep_stmt)
        # db post operation
        return redirect(url_for('login'))
    elif request.method == 'GET':
        return render_template('signup.html')


@app.route('/toggle', methods=['POST'])
def toggle_user():
    data = request.get_json(force=True)

    username = data['username']
    role = data['role']
    print(username)
    print(role)
    sql = "update user set role=? where username=?"
    prep_stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(prep_stmt, 1, role)
    ibm_db.bind_param(prep_stmt, 2, username)
    ibm_db.execute(prep_stmt)
    return jsonify(
        status="success",
        role=role
    )


@app.route('/requestPalsma', methods=['POST'])
def requestBloodPlasma():
    # fetch mail address of the donors
    data = request.get_json(force=True)
    username = data['username']
    name = data['name']
    age =  data['age']
    sex = data['sex']
    blood_type = data['bloodtype']
    phone_number = data['phone_num']
    sql = "select email from user where blood_group=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, blood_type)
    ibm_db.execute(stmt)
    dic = ibm_db.fetch_assoc(stmt)
    email_list = []
    while dic != False:
        email_list.append(dic['EMAIL'])
        print(dic['EMAIL'])
        dic = ibm_db.fetch_assoc(stmt)
    # send mail

    message = Mail(
        from_email='eshwaran.s.2019.cse@rajalakshmi.edu.in',
        to_emails=email_list,
        subject='Sending with Twilio SendGrid is Fun',
        html_content='<h1>Need Of Blood</h1><table><tr><th>Name</th><th>' +
name + '</th></tr><tr><th>Age</th><th>' + age + '</th></tr><tr><th>Sex</th><th>'
+ sex + '</th></tr><tr><th>Blood Group</th><th>' + blood_type +
'</th></tr><tr><th>Phone Number</th><th>' + phone_number + '</th></tr></table>'
    )
    try:
        sg = SendGridAPIClient("SG.3iBLSgAYTEuVbfSHu9dCPA.-
nrnikWJvaRlNLMONA04_CuKAyPeV69c46vPAh3vUX0")
        response = sg.send(message)
        print(response.status_code)
        print(response.body)
        print(response.headers)
    except Exception as e:
        print(e.message)
    # insert data into requests table
    sql = "insert into bloodrequests(username,name,age,sex,blood_type) values
(?,?,?,?,?)"
    prep_stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(prep_stmt, 1, username)
    ibm_db.bind_param(prep_stmt, 2, name)
    ibm_db.bind_param(prep_stmt, 3, age)
    ibm_db.bind_param(prep_stmt, 4, sex)
    ibm_db.bind_param(prep_stmt, 5, blood_type)
    ibm_db.execute(prep_stmt)

    return jsonify(
        name=name,
        age=age,
        sex=sex,
        bloodtype=blood_type,
        status="yes"
    )


@app.route('/getrequests', methods=['POST'])
def getBloodRequests():
    data = request.get_json(force=True)
    username = data['username']
    sql = "select * from bloodrequests where username=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, username)
    ibm_db.execute(stmt)
    dic = ibm_db.fetch_assoc(stmt)
    requests = []
    print(type(dic))
    while dic != False:
        single_request = {
            'name': dic['NAME'],
            'age': dic['AGE'],
            'sex': dic['SEX'],
            'blood_type': dic['BLOOD_TYPE']
        }
        print(single_request)
        requests.append(single_request)
        dic = ibm_db.fetch_assoc(stmt)
    return jsonify(
        username=username,
        requests=requests
    )


if__name__== '__main__':
    app.run(host="0.0.0.0", debug=True)
```

13.2 GITHUB

https://github.com/IBM-EPBL/IBM-Project-19380-1659697030

13.3 VIDEO LINK

https://drive.google.com/file/d/1_vlc3GoltUzG8I9DyYKp-oHgXAIesKS0/view?usp=share_link