TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

PROJECT OVERVIEW

The main goal of our project is to design a user-friendly web application that is like a scientific vehicle from which we can help reduce mortality or help those affected by COVID19 by donating plasma from patients who have recovered without approved antiretroviral therapy planning for a deadly COVID19 infection, plasma therapy is an experimental approach to treat those COVID-positive patients and help them recover faster .Therapy, which is considered reliable and safe. If a particular person has fully recovered from COVID19, they are eligible to donate their plasma.

During the COVID 19 crisis, the requirement of plasma became high and the donor count being low. Saving the donor information and helping the need by notifying the current donors would be a helping hand. In regard to the problem faced, an application is to be built which would take the donor details store it and inform them upon a request.

The main aim of this project is to save lives of people by providing blood. Our project Online Blood Donation Center Website. This website reduces the time to a greater extent that is searching for the required blood. This application provides the required information in less time and also helps in quicker decision making.

1.2 PURPOSE

The plasma data are maintained in the database. New plasma details are entered in to the project to manage plasma details. Plasma donor details are entered and maintained in the database .Basic purpose of the system is to Search plasma that occurs during the operation as well as performing calculation and updating database as and when necessary. The system is can also provide information of donor about current state .This project is the college level project and is implementing under the guidance of college professors. The purpose of the online system is to create convenient and easy-to-use online system for users, trying to get or donate plasma. The system is based on a relational database. The specification builds on the experience of users of IT technology in blood transfusion that is currently available and informs both Connecting for Health (CFH) and commercial companies producing both hardware and software.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

Entering the details about the blood groups, members, name, date of birth etc. And tracking the database is complicated when the details are maintained. This makes the maintenance of schedule erroneous. The major problem in plasma Donation systems was that, they don’t follow the actual needs of users. There was shortage and sometimes unavailability of rare blood

groups due to less modules i.e. patient and donors .In this way we realize that the new system is required and will certainly improve the performance of the exiting system over the exiting paper based system. Design the system to develop the alternative computer based system. To understand the user characteristic .Design a system for a particular types of user.

2.2 REFERENCES

[1] Dennis O’Neil(1999). “Blood Components”. Palomar College. Archived from the original on June 5,2013.

[2] Tuskegee University(May 29, 2013).”Chapter 9 Blood”.tuskegee.edu. Archi from the original on December 28, 2013.

[3] ”Ways to Keep Your Blood Plasma Healthy”. Archived from the original on November 1, 2013.Retrieved November 10, 2011.

[4] Jump up to Maton , Anthea; Jean Hopkins; Charles Wiliam McLaughlin; Susan Johnson; Maryanna Quon Warner LaHart; David LaHart; Jill D. Wright (1993), Human Biology and Health, Englewood Cliffs, New Jersey,USA.

[5] The Physics Factbook - Density of Blood.[6]Basic Biology(2015).”Blood cells”.

[6] Elkassabany NM, Meny GM, Doria RR, Marcucci C (2008). “Green Plasma Revisited”. Anesthesiology 108(4);

[7] “19th WHO Model List of Essential Medicines(April 2015)”(PDF). WHO April 2015. Retrieved May 10,2015.

[8] Tripathi S, Kumar V,Prabhakar A, Joshi S, Agarwal A(2015).”Passive blood plasma separation at the microscale; a review of design principles and microdevices”. J.Micromech, Microeng 25(8); 083001.

[9] Guo, Weijin; Hansson, Jonas; van der wijngaart, Wouter(2020).”Synthetic Paper Separates Plasmafrom Whole Blood with Low Protein Loss”.Analytical Chemistry.92(9): 6194-6199.

[10] Starr, Douglas P.(2000), Blood: An Epic History of Medicine and Commerce. New York:Quill.

2.3 PROBLEM STATEMENT DEFINITION

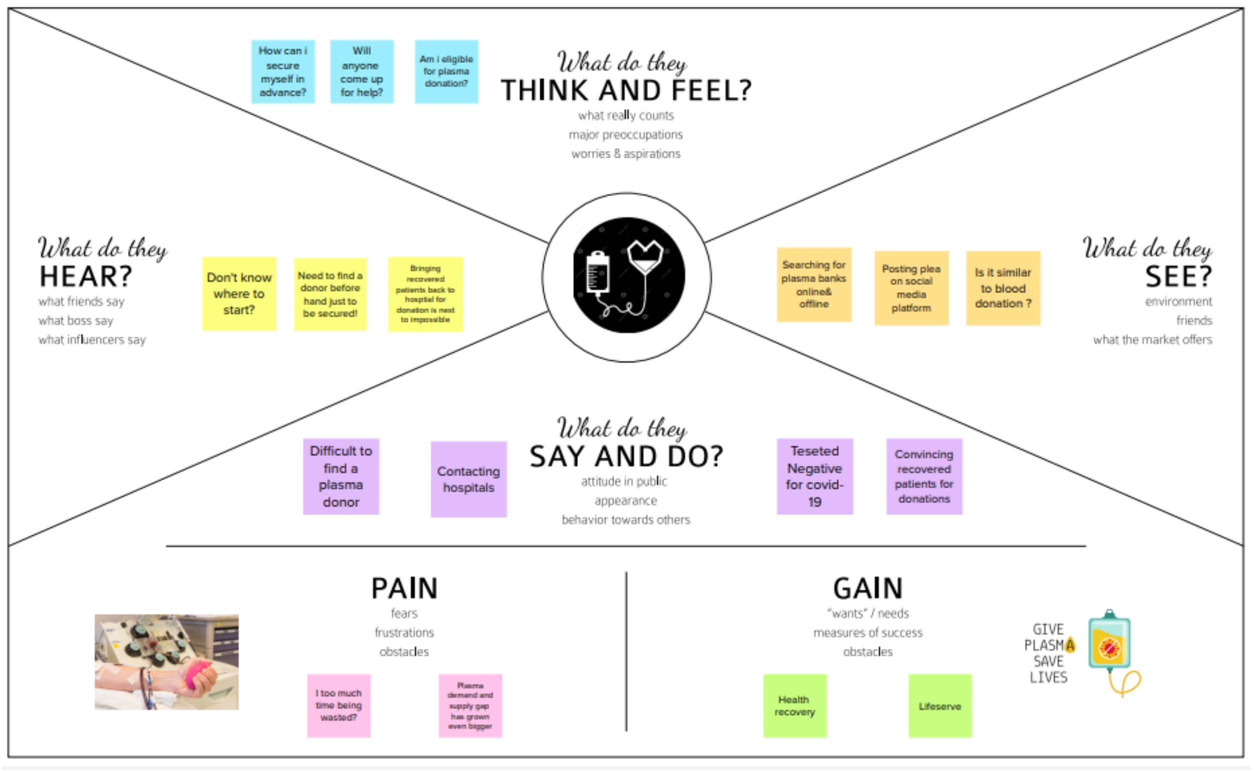
In recent days, it is noticed the increase in plasma request posts on social media. Interestingly there are many people across the world interested in donating plasma when there is a need, but those donors don’t have an access to know about the plasma donation requests in their local area. This is because that there is no platform to connect local plasma donors with patients. Plasma Donor Application solves the problem and creates a communication channel between donor and patients. It is a useful tool to find compatible plasma donors who can receive plasma request posts in their local area. Clinics can use this web application to maintain the plasma donation activity.Collected data through this application can be used to analyse donations to requests rates in a local area to increase the awareness of people by conducting donations camps.

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.
It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



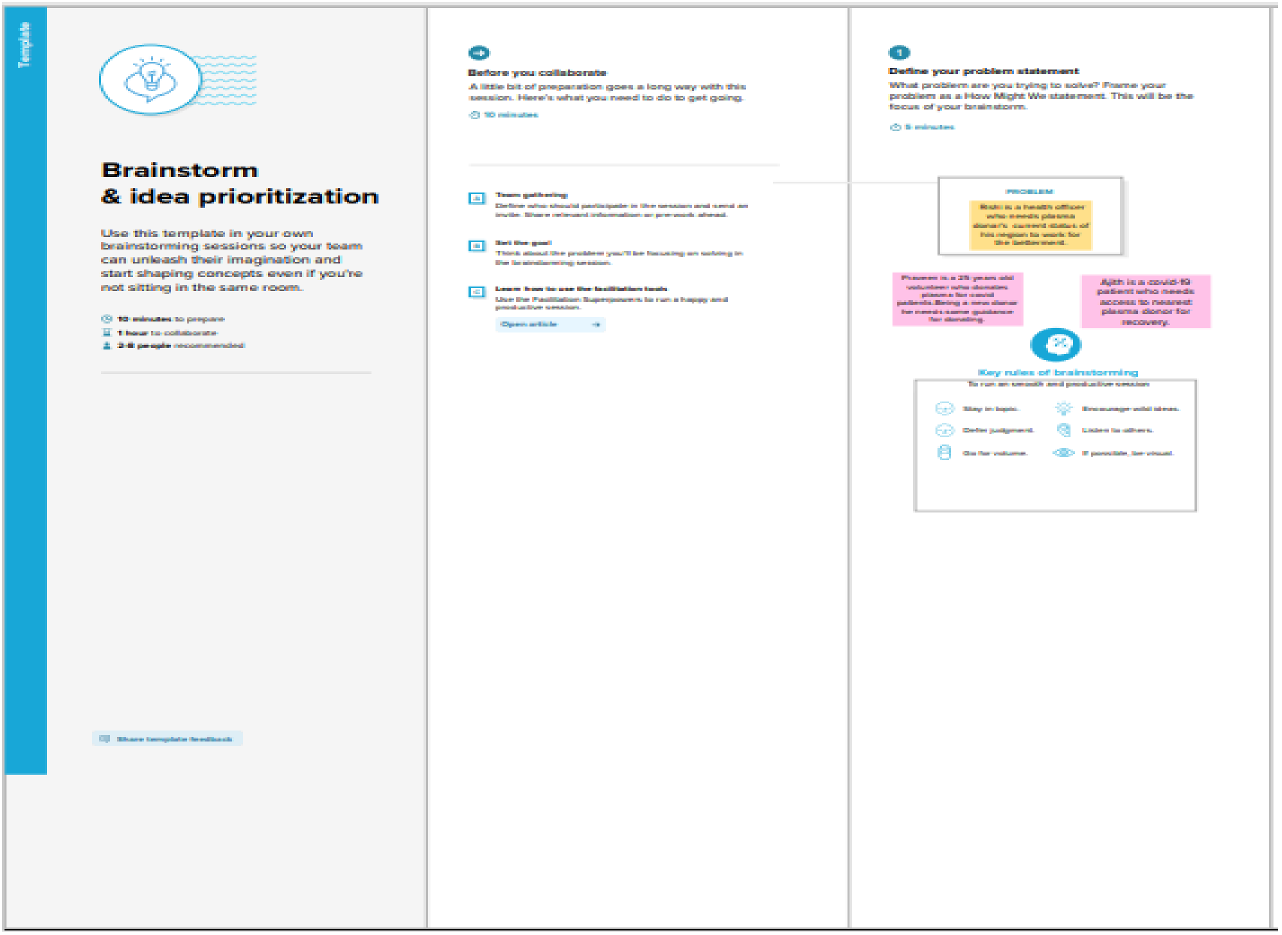
What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
How can i secure myself in advance?
Will anyone come up for help?
Am i eligible for plasma donation?
What do they HEAR?
what friends say
what boss say
what influencers say
Don't know where to start?
Need to find a donor before hand just to be secured!
Bringing recovered patients back to hosptial for donation is next to impossible
What do they SEE?
environment
friends
what the market offers
Searching for plasma banks online& offline
Posting plea on social media platform
Is it similar to blood donation ?
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Difficult to find a plasma donor
Contacting hospitals
Teseted Negative for covid-19
Convincing recovered patients for donations
PAIN
fears
frustrations
obstacles
I too much time being wasted?
Plasma demand and supply gap has grown even bigger
GAIN
"wants" / needs
measures of success
obstacles
Health recovery
Lifeserve
GIVE PLASMA SAVE LIVES

Figure 3.1 Empathy Map Canvas

3.2 IDEATION & BRAINSTORMING

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.


Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

Before you collaborate
A little bit of preparation goes a long way with this session. Here's what you need to do to get going.
10 minutes

Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.
Open article →

1
Define your problem statement
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.
5 minutes

PROBLEM
Rishi is a health officer who needs plasma donor's current status of his region to work for the betterment.

Praveen is a 25 years old volunteer who donates plasma for covid patients.Being a new donor he needs some guidance for donating.

Ajith is a covid-19 patient who needs access to nearest plasma donor for recovery.

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

Figure 3.2 Ideation And Brainstorming

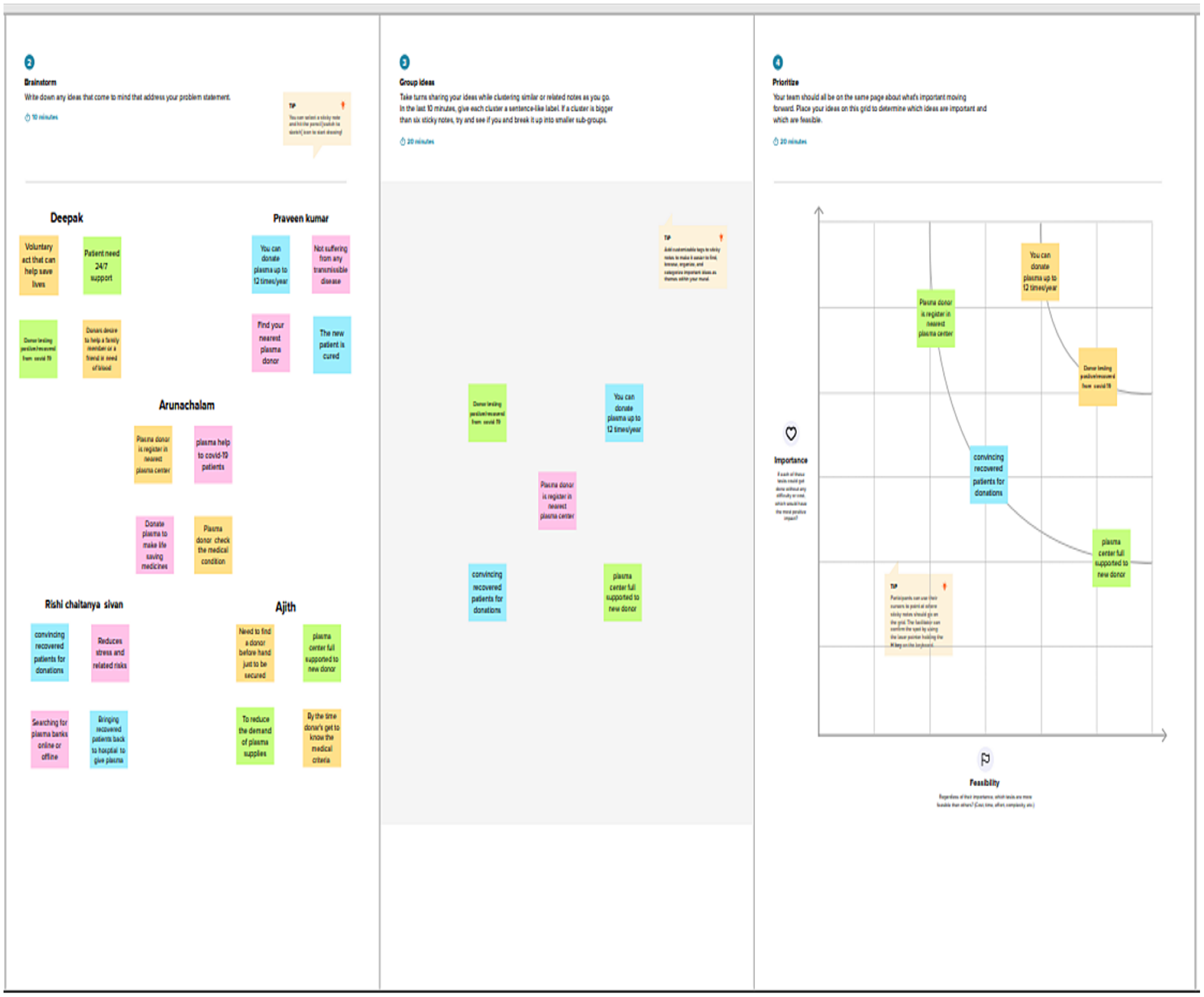

2
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
TIP
Deepak
Voluntary act that can help save lives
Patient need 24/7 support
Donor testing positive/recovered from covid 19
Donors desire to help a family member or a friend in need of blood
Praveen kumar
You can donate plasma up to 12 times/year
Not suffering from any transmissible disease
Find your nearest plasma donor
The new patient is cured
Arunachalam
Plasma donor is register in nearest plasma center
plasma help to covid-19 patients
Donate plasma to make life saving medicines
Plasma donor check the medical condition
Rishi chaitanya sivan
convincing recovered patients for donations
Reduces stress and related risks
Searching for plasma banks online or offline
Bringing recovered patients back to hospital to give plasma
Ajith
Need to find a donor before hand just to be secured
plasma center full supported to new donor
To reduce the demand of plasma supplies
By the time donor's get to know the medical criteria
3
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
TIP
Donor testing positive/recovered from covid 19
You can donate plasma up to 12 times/year
Plasma donor is register in nearest plasma center
convincing recovered patients for donations
plasma center full supported to new donor
4
Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes
Importance
Plasma donor is register in nearest plasma center
You can donate plasma up to 12 times/year
Donor testing positive/recovered from covid 19
convincing recovered patients for donations
plasma center full supported to new donor
TIP
Feasibility

Figure 3.2.1 Ideation And Brainstorming

3.3 PROPOSED SOLUTION

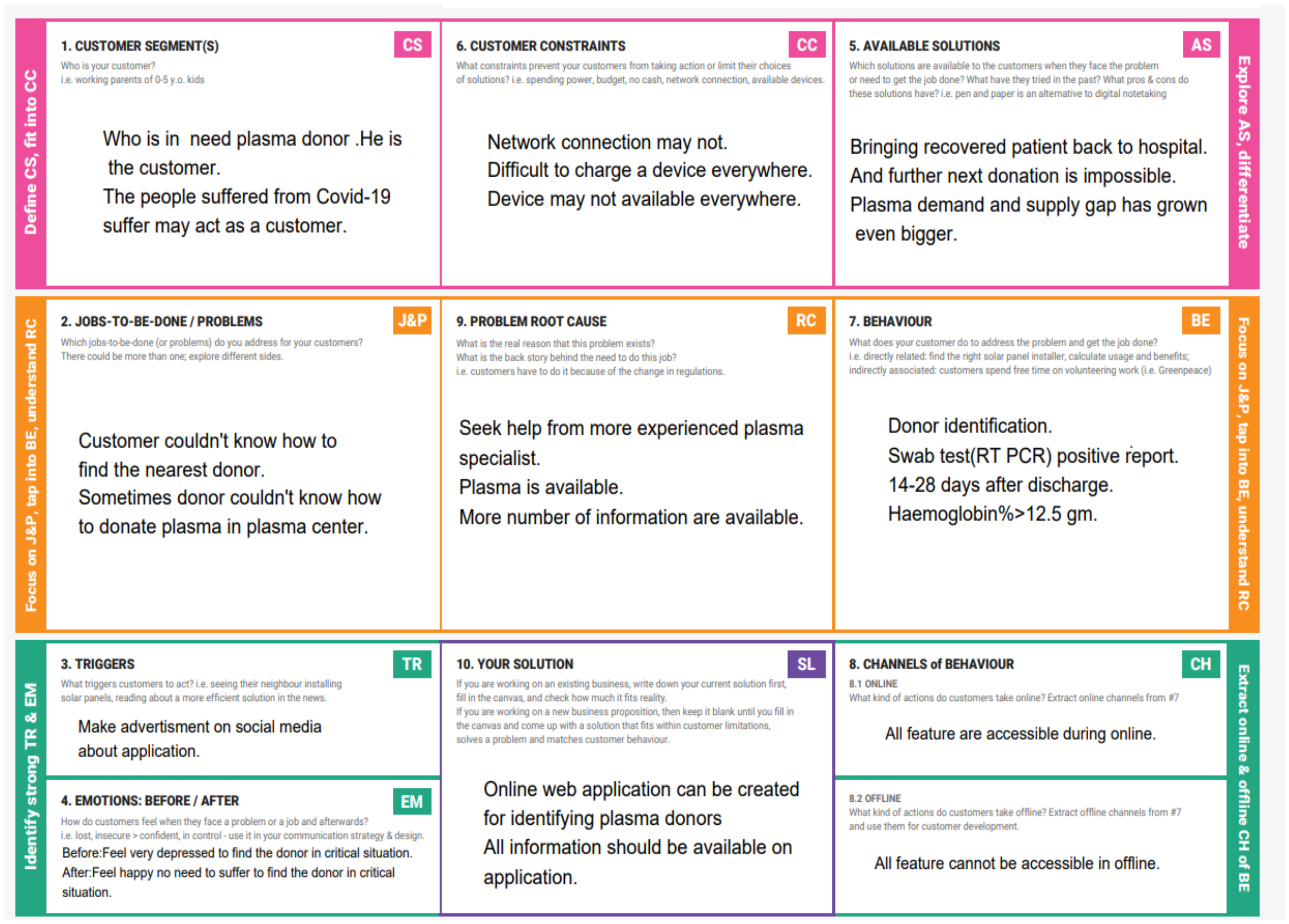
Having hooked your audience into the problem, now you want to paint a picture of what the world will be like when you solve the problem. Your proposed solution should relate the current situation to a desired result and describe the benefits that will accrue when the desired result is achieved. So, begin your proposed solution by briefly describing this desired result.

S.No.	**Parameter**	**Description**
1	Problem Statement (Problem to besolved)	Blood banks are required to maintain account of blood bags in the inventory. This increases with each blood donation recorded in our systemand decreases as they are checked out upon hospital requests. Our system will need to keep the information up todate to ensure correctness of the inventory.
2	Idea / Solution description	To develop a system that provides functions to support donors to view and manage their information conveniently. To maintain records of blood donors, blood donation information and bloodstocks in a centralized databasesystem.
3	Novelty / Uniqueness	Donors who wish to donate plasmacan donate by uploading their COVID19 recovery certificate on the donor'spage. If the donor isnew, they must registerbefore log in.Ifthe donor is an existing user they needto login.Username and e-mail provided at the timeofregistration.
4	Social Impact/ Customer Satisfaction	Interestingly thereare many people acrossthe world interested in donating blood when there is a need, but those donorsdon’t have an access to know about the blood donation requests in their localarea. This is because thatthere is no platform to connect local blood donorswith patients.

5	Business Model(Revenue Model)	Focused on specific phases/aspects of blood collection. Not based on the benefits monetization.Focused on actorsother than blood donars.
6	Scalabilityof the Solution	Blood bank around the world often struggle to collect sufficient quantities of blood, plasma,and other components to meet theneeds of the large and growing numberof patients who rely on transfusions and blood-derived therapies for survival. Several blood collection organization have explored the use of both behavioral nudgesand economics incentives, withvarying degrees of success.

3.4 PROBLEM SOLUTION FIT

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer’s problem.



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Who is in need plasma donor .He is the customer.
The people suffered from Covid-19 suffer may act as a customer.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Network connection may not.
Difficult to charge a device everywhere.
Device may not available everywhere.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Bringing recovered patient back to hospital.
And further next donation is impossible.
Plasma demand and supply gap has grown even bigger.

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?
There could be more than one; explore different sides.

Customer couldn't know how to find the nearest donor.
Sometimes donor couldn't know how to donate plasma in plasma center.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

Seek help from more experienced plasma specialist.
Plasma is available.
More number of information are available.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Donor identification.
Swab test(RT PCR) positive report.
14-28 days after discharge.
Haemoglobin%>12.5 gm.

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

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Make advertisment on social media about application.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Before:Feel very depressed to find the donor in critical situation.
After:Feel happy no need to suffer to find the donor in critical situation.

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

Online web application can be created for identifying plasma donors
All information should be available on application.

8. CHANNELS of BEHAVIOUR CH

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

All feature are accessible during online.

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

All feature cannot be accessible in offline.

Extract online & offline CH of BE

Figure3.4 Problem Solution Fit

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

FR NO.	***Functional Requirement (Epic)***	***Sub Requirement (Story / Sub-Task)***
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User queries	User interact with IBM Watson assistant
FR-4	User Notification	Donor accept request and send notification mail in user Id

4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	***Non-Functional Requirement***	***Description***
NFR-1	Usability	User with no understanding of application must be able to interact with chatbot.
NFR-2	Security	Access permission for the user information may only be changed by user ,be implemented as an extra security feature,authorized storage data.
NFR-3	Reliability	The database update process must roll back all related updates when any update fails
NFR-4	Performance	Reduce font page load time must be no more 2 seconds.
NFR-5	Availability	New module deployment must not impact font end page in upgrade time.
NFR-6	Scalability	Application traffic limit must be scable enough to support 10000 user at time

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

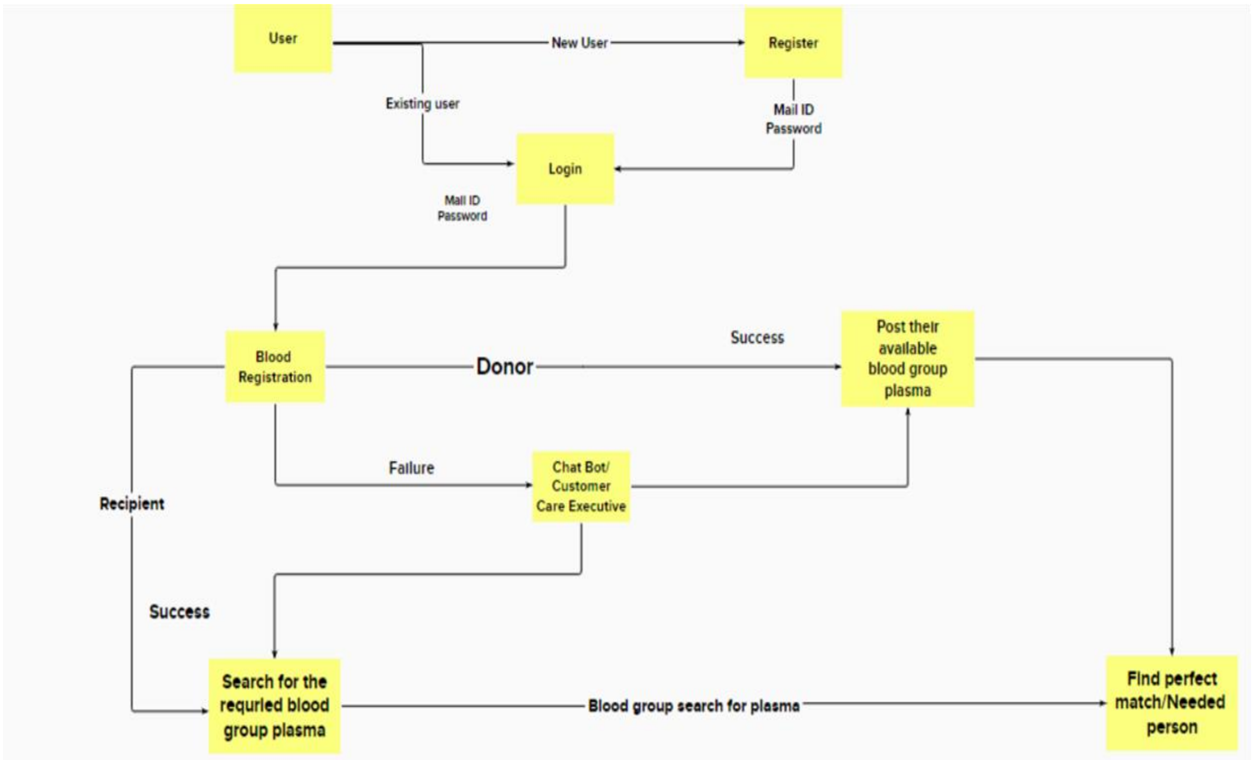

User
New User
Register
Existing user
Mail ID
Password
Login
Mail ID
Password
Blood Registration
Donor
Success
Post their available blood group plasma
Failure
Chat Bot/ Customer Care Executive
Recipient
Success
Search for the requried blood group plasma
Blood group search for plasma
Find perfect match/Needed person

Figure 5.1 Data flow diagram

5.2 SOLUTION ARCHITECTURE

Solution architecture is a complex process with many sub-processes that bridgesthe gap between business problems and technology solutions. Its goals are to:

Find the best tech solution to solve existing business problems.

- Describe the structure, characteristics, behaviour, and other aspects of thesoftware to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed,and delivered.

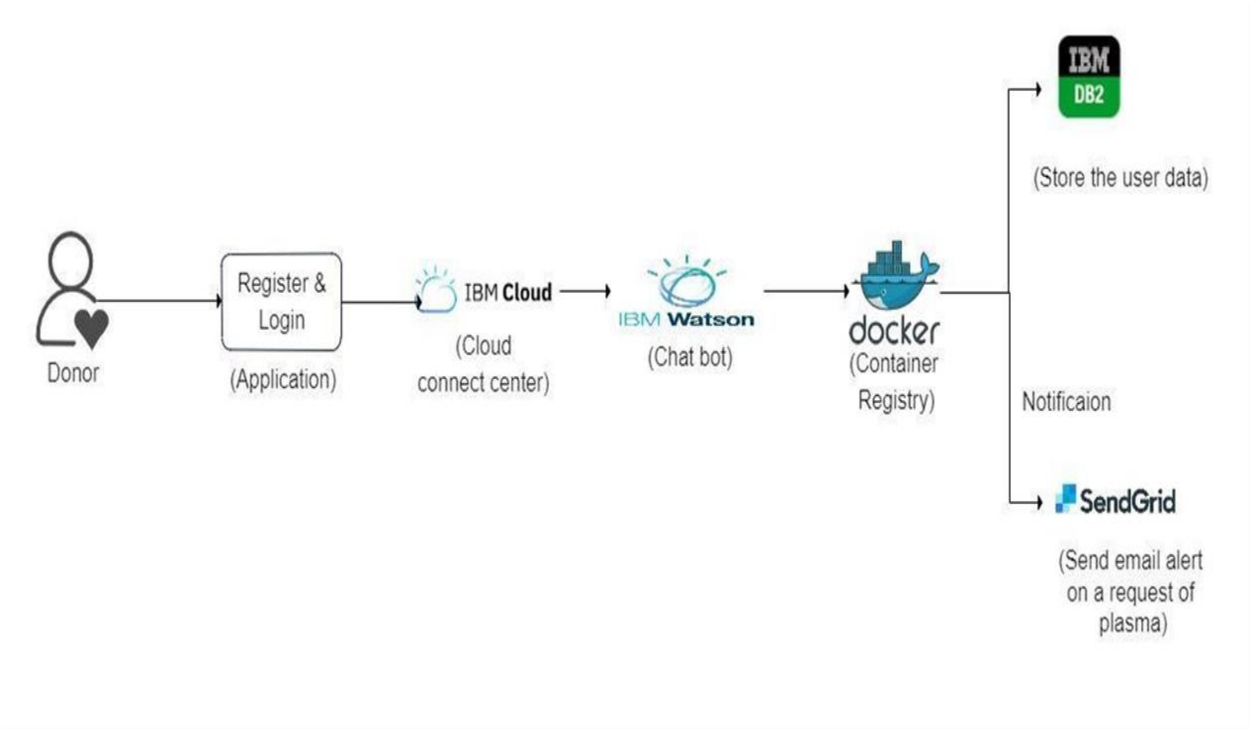


figure 5.2 Solution Architecture

5.2.1 TECHNOLOGYARCHITECTURE

Technical Architecture (TA) is a form of IT architecture that is used to design computer systems. It involves the development of a technical blueprint about the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.

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

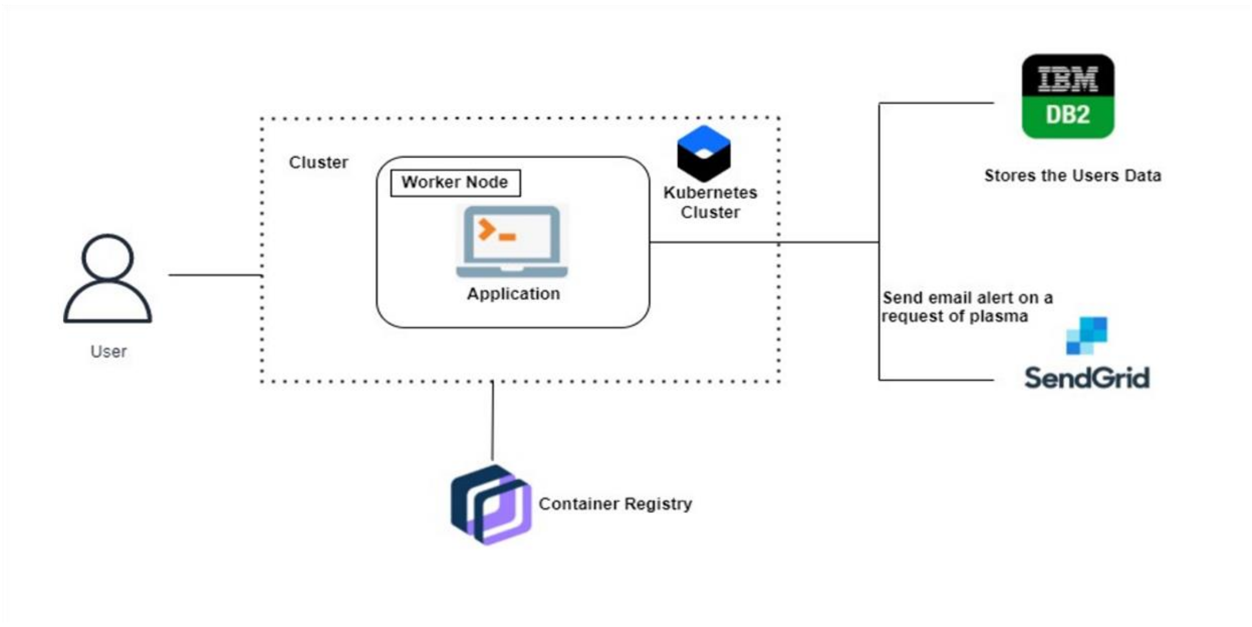
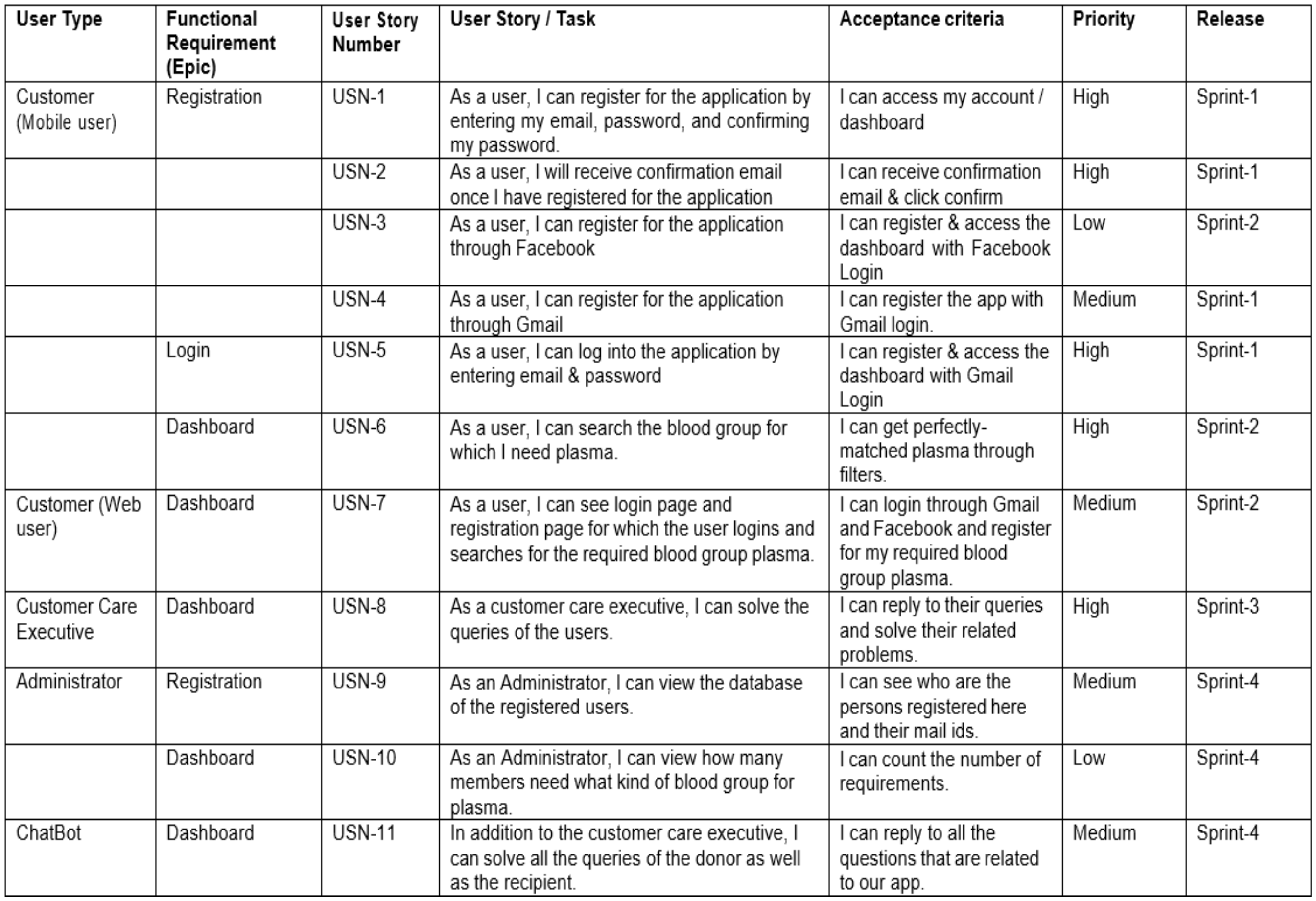


Figure 5.2.1 Technology architecture diagram

5.3 USER STORIES

A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer.



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register the app with Gmail login.	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can register & access the dashboard with Gmail Login	High	Sprint-1
	Dashboard	USN-6	As a user, I can search the blood group for which I need plasma.	I can get perfectly-matched plasma through filters.	High	Sprint-2
Customer (Web user)	Dashboard	USN-7	As a user, I can see login page and registration page for which the user logins and searches for the required blood group plasma.	I can login through Gmail and Facebook and register for my required blood group plasma.	Medium	Sprint-2
Customer Care Executive	Dashboard	USN-8	As a customer care executive, I can solve the queries of the users.	I can reply to their queries and solve their related problems.	High	Sprint-3
Administrator	Registration	USN-9	As an Administrator, I can view the database of the registered users.	I can see who are the persons registered here and their mail ids.	Medium	Sprint-4
	Dashboard	USN-10	As an Administrator, I can view how many members need what kind of blood group for plasma.	I can count the number of requirements.	Low	Sprint-4
ChatBot	Dashboard	USN-11	In addition to the customer care executive, I can solve all the queries of the donor as well as the recipient.	I can reply to all the questions that are related to our app.	Medium	Sprint-4

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	**Registration and Login**	USN-1	Create UI to interact with pages. To create the userand admin login functionality	20	High	Praveen Kumar M Deepak G
Sprint-2	**Cloud and Database**	USN-2	Connecting flask app with database[IBMDB2] Implementation of IBM chatbot	20	High	Rishi Chaitanya sivan E Ajith P
Sprint-3	**Deployment in Devops phase**	USN-3	Creating images with docker, Deploying Kubernetes and add the mailing service.	20	High	Deepak G Praveen Kumar M
Sprint-4	**Testing and Deployment to user**	USN-4	To make sure that the software is handy to users.	20	High	Ajith P Arunachalam K

6.2 SPRINT DELIVERY SCHEDULE

Project Tracker:

Sprint	**TotalStory Points**	**Duration**	**SprintStart Date**	**SprintEnd Date (Planned)**	**Story PointsCompleted(as on Planned End Date)**	**SprintReleaseDate (Actual)**
Sprint -1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint -2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov2022
Sprint -3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint -4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let'scalculate the team's average velocity (AV) per iterationunit (story points per day)

Sprint duration = 6 days Velocity of team=20points

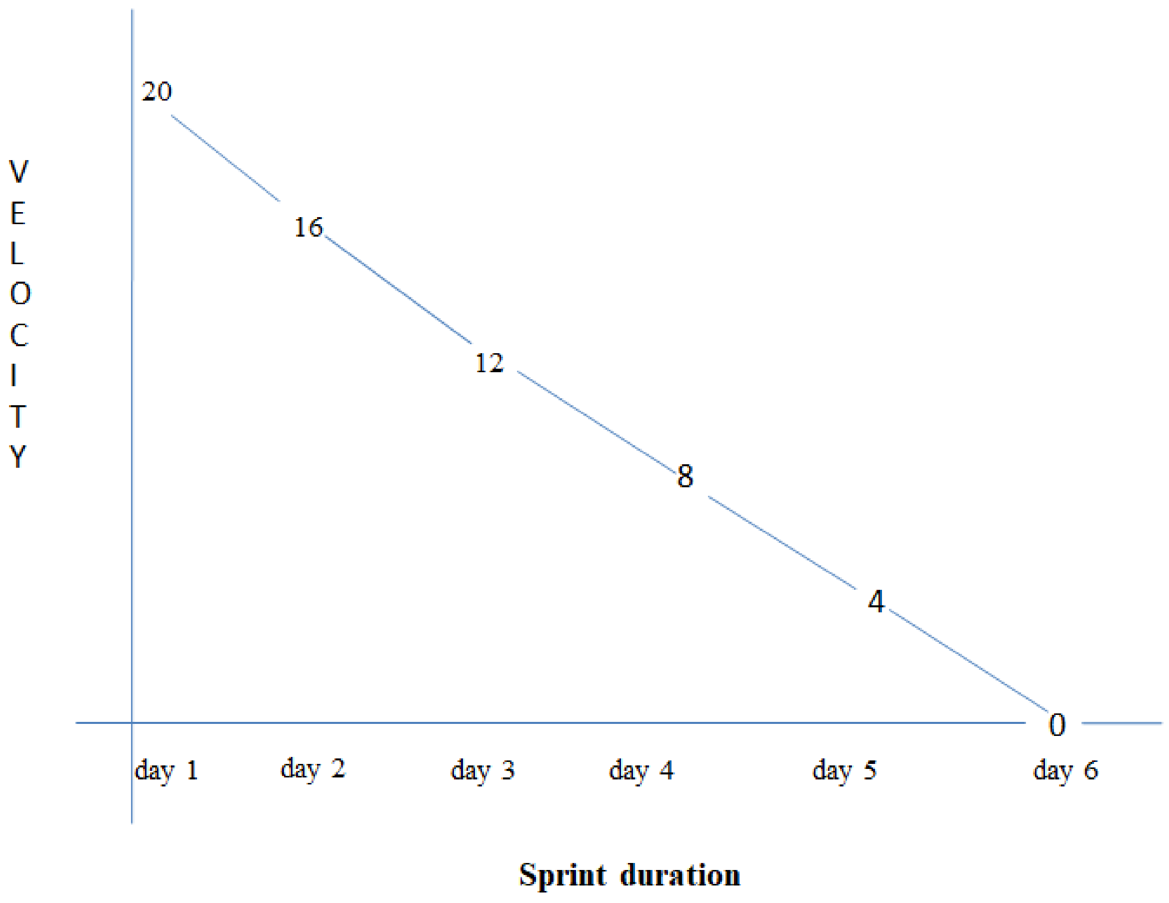
Average velocity(AV) = Velocity/Sprint duration

AV = 20/6 = 3.3

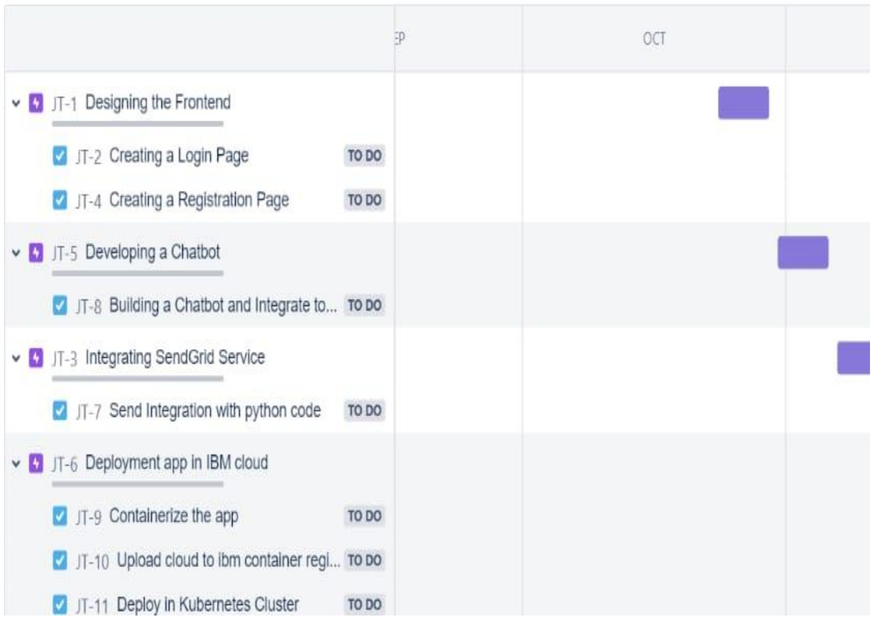
Average Velocity = 3.34

BURNDOWN CHART:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such asScrum. However, burn down charts can be applied to any project containing measurable progress over time.



6.3 REPORTS FROM JIRA



JT-1 Designing the Frontend
JT-2 Creating a Login Page TO DO
JT-4 Creating a Registration Page TO DO
JT-5 Developing a Chatbot
JT-8 Building a Chatbot and Integrate to... TO DO
JT-3 Integrating SendGrid Service
JT-7 Send Integration with python code TO DO
JT-6 Deployment app in IBM cloud
JT-9 Containerize the app TO DO
JT-10 Upload cloud to ibm container regi... TO DO
JT-11 Deploy in Kubernetes Cluster TO DO

Figure 6.3 Jira

CHAPTER 7

CODING AND SOLUTIONING

7.1 FEATURE 1

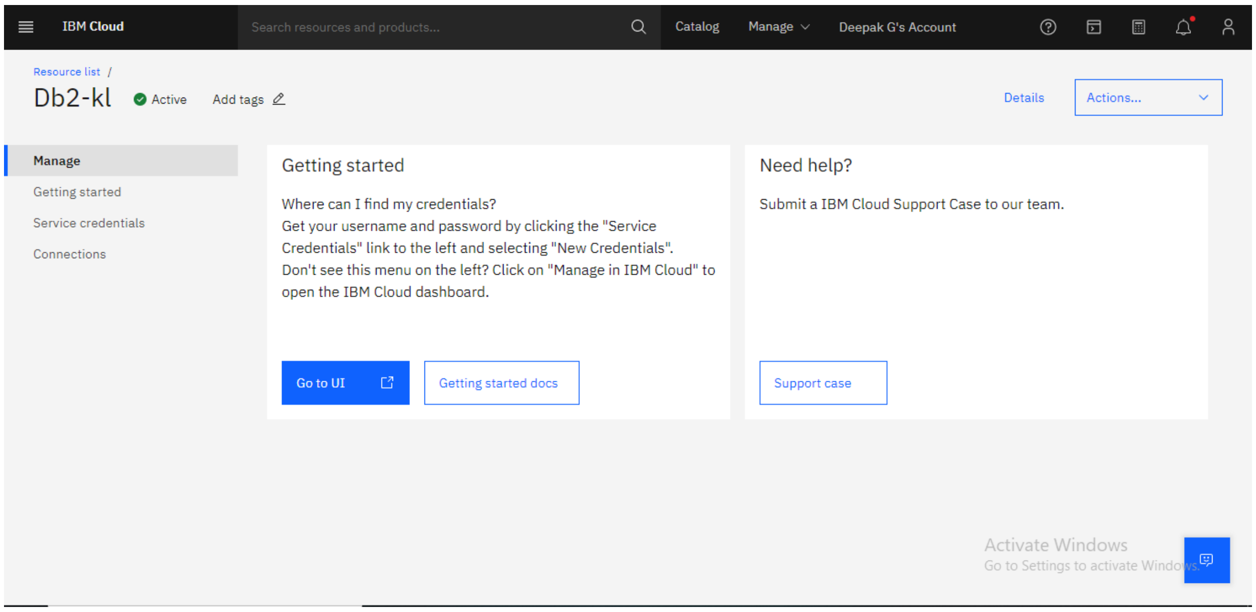
The top two features rated as “useful” or “very useful” by the highest percentage of surveyed donors were the ability to ask questions when needed (70.8%), and the ability to locate the nearest blood center on the map and calculate the time needed to reach it (67.8%). Also, more than half of the respondents (52.9%) expressed interest in being notified about special recruitment events, such as blood drives and promotions and blood product shortages (52.2%). The ability to quickly confirm an appointment was found to be useful for 49.8% of respondents. Furthermore, similar percentages of respondents rated the ability to share donations on social media for donation promotion, and the option of requesting someone from the blood center to call the donor back (43.7% and 43.1%, respectively). The option of leaving a message for the blood center, including comments, suggestions, pictures or video, was considered useful by the lowest percentages of respondents (33.9%)

7.2 FEATURE 2

When the surveyed donors were asked about potential areas of concerns associated with using the app, 67.3% of donors claimed that they are “concerned” or “very concerned” about using their personal information for other purposes. A similar percentage of respondents (65.8%) agreed that they are concern about getting too many alerts or messages. More than half of the respondents (54.1%) were concerned about being unable to get their questions answered immediately. Half of the respondents (50.2%) were concern about the app being too difficult to use, and 29.9% of respondents indicated concerns about the lack of personal touch, with it being the lowest percentage reported regarding areas of concern

7.3 DATABASE SCHEMA

Create database on IBM_DB



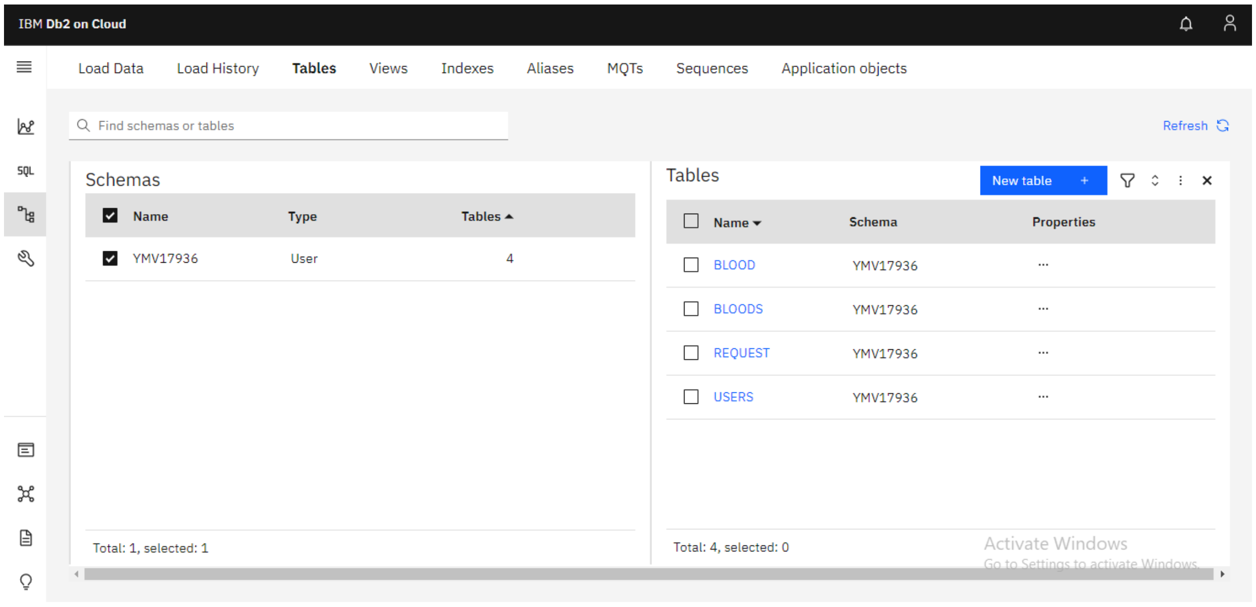
Getting started

Where can I find my credentials?
Get your username and password by clicking the "Service Credentials" link to the left and selecting "New Credentials".
Don't see this menu on the left? Click on "Manage in IBM Cloud" to open the IBM Cloud dashboard.

Need help?

Submit a IBM Cloud Support Case to our team.

Table Creation



Schemas

Name	Type	Tables
YMV17936	User	4

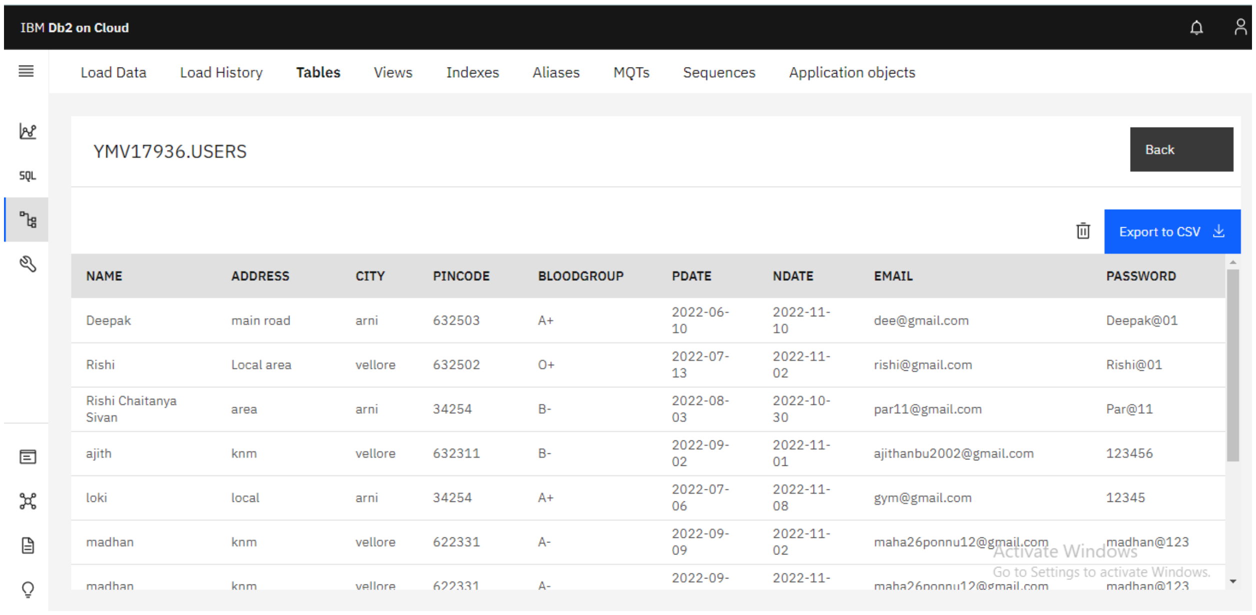
Total: 1, selected: 1

Tables

Name	Schema	Properties
BLOOD	YMV17936	...
BLOODS	YMV17936	...
REQUEST	YMV17936	...
USERS	YMV17936	...

Total: 4, selected: 0

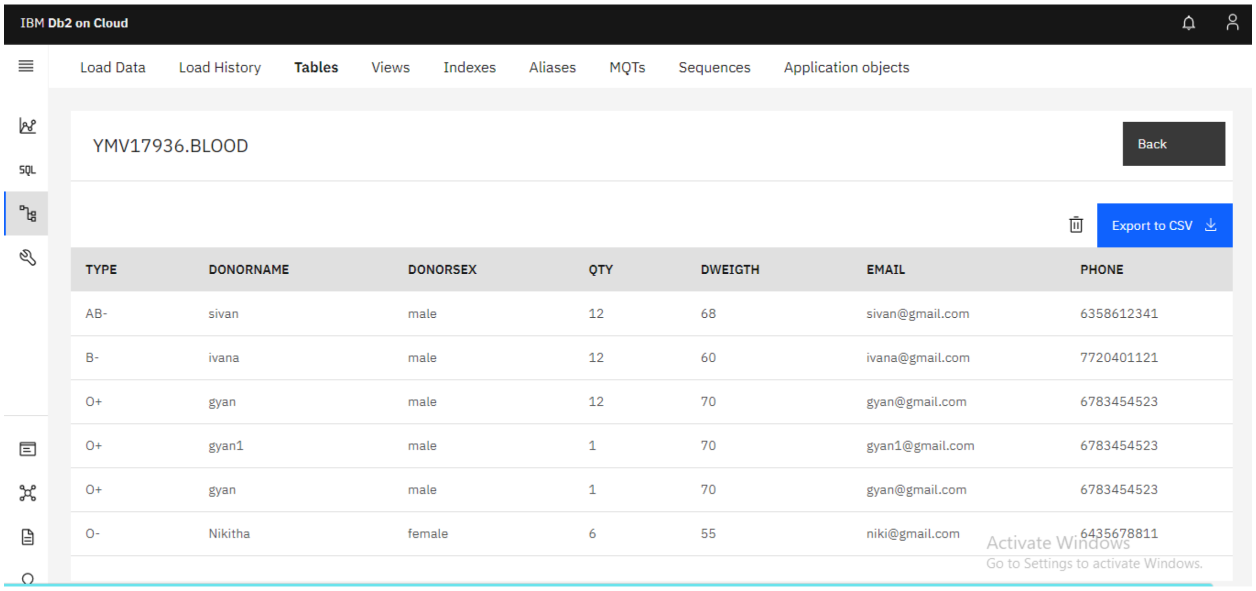
Users Table



YMV17936.USERS

NAME	ADDRESS	CITY	PINCODE	BLOODGROUP	PDATE	NDATE	EMAIL	PASSWORD
Deepak	main road	arni	632503	A+	2022-06-10	2022-11-10	dee@gmail.com	Deepak@01
Rishi	Local area	vellore	632502	O+	2022-07-13	2022-11-02	rishi@gmail.com	Rishi@01
Rishi Chaitanya Sivan	area	arni	34254	B-	2022-08-03	2022-10-30	par11@gmail.com	Par@11
ajith	knm	vellore	632311	B-	2022-09-02	2022-11-01	ajithanbu2002@gmail.com	123456
loki	local	arni	34254	A+	2022-07-06	2022-11-08	gym@gmail.com	12345
madhan	knm	vellore	622331	A-	2022-09-09	2022-11-02	maha26ponnu12@gmail.com	madhan@123
madhan	knm	vellore	622331	A-	2022-09-	2022-11-	maha26ponnu12@gmail.com	madhan@123

Blood Table



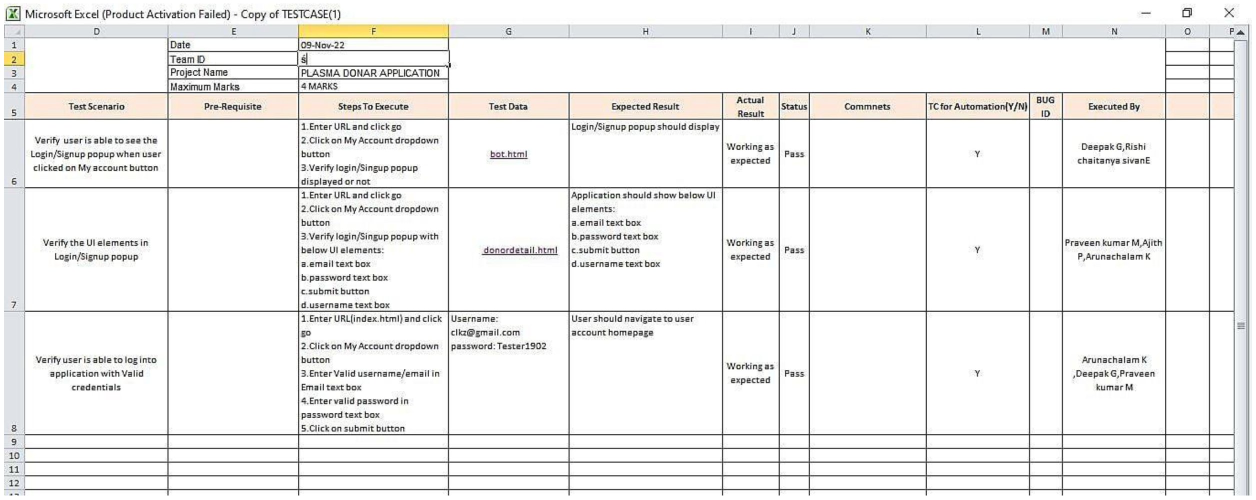
YMV17936.BLOOD

TYPE	DONORNAME	DONORSEX	QTY	DWEIGTH	EMAIL	PHONE
AB-	sivan	male	12	68	sivan@gmail.com	6358612341
B-	ivana	male	12	60	ivana@gmail.com	7720401121
O+	gyan	male	12	70	gyan@gmail.com	6783454523
O+	gyan1	male	1	70	gyan1@gmail.com	6783454523
O+	gyan	male	1	70	gyan@gmail.com	6783454523
O-	Nikitha	female	6	55	niki@gmail.com	6435678811

CHAPTER 8

TESTING

8.1 TEST CASE

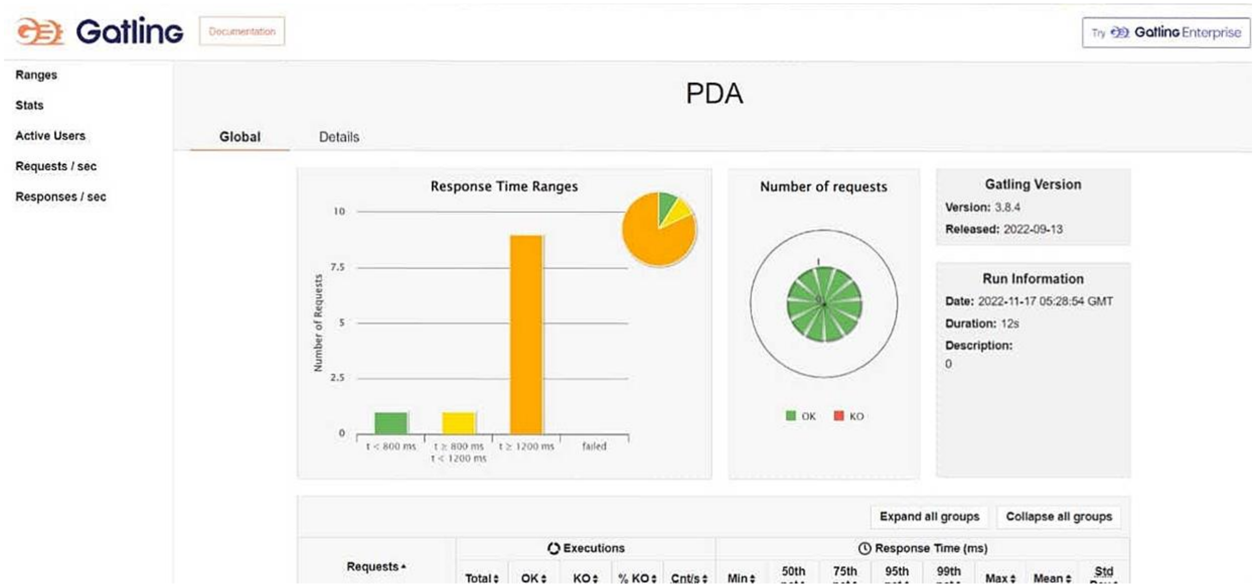


Microsoft Excel (Product Activation Failed) - Copy of TESTCASE(1)

Date	09-Nov-22
Team ID	ś
Project Name	PLASMA DONAR APPLICATION
Maximum Marks	4 MARKS

Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
Verify user is able to see the Login/Signup popup when user clicked on My account button		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	bot.html	Login/Signup popup should display	Working as expected	Pass		Y		Deepak G,Rishi chaitanya sivanE
Verify the UI elements in Login/Signup popup		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.submit button d.username text box	donordetail.html	Application should show below UI elements: a.email text box b.password text box c.submit button d.username text box	Working as expected	Pass		Y		Praveen kumar M,Ajith P,Arunachalam K
Verify user is able to log into application with Valid credentials		1.Enter URL(index.html) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on submit button	Username: clkz@gmail.com password: Tester1902	User should navigate to user account homepage	Working as expected	Pass		Y		Arunachalam K ,Deepak G,Praveen kumar M

8.2 PERFORMANCE CASES



Gatling Version

Version: 3.8.4

Released: 2022-09-13

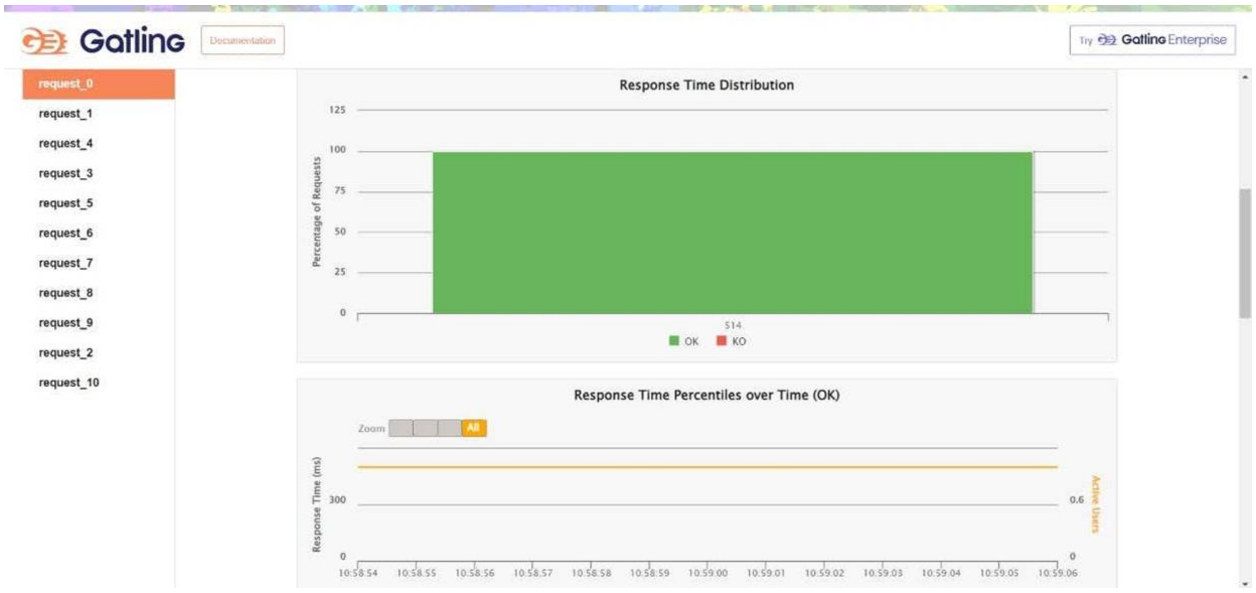
Run Information

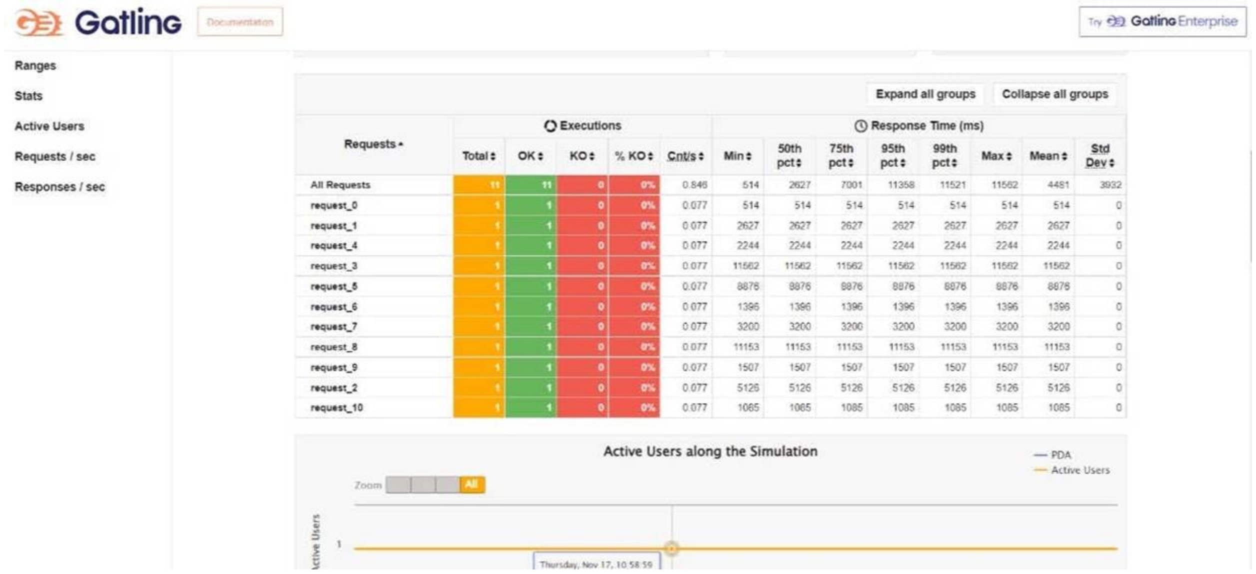
Date: 2022-11-17 05:28:54 GMT

Duration: 12s

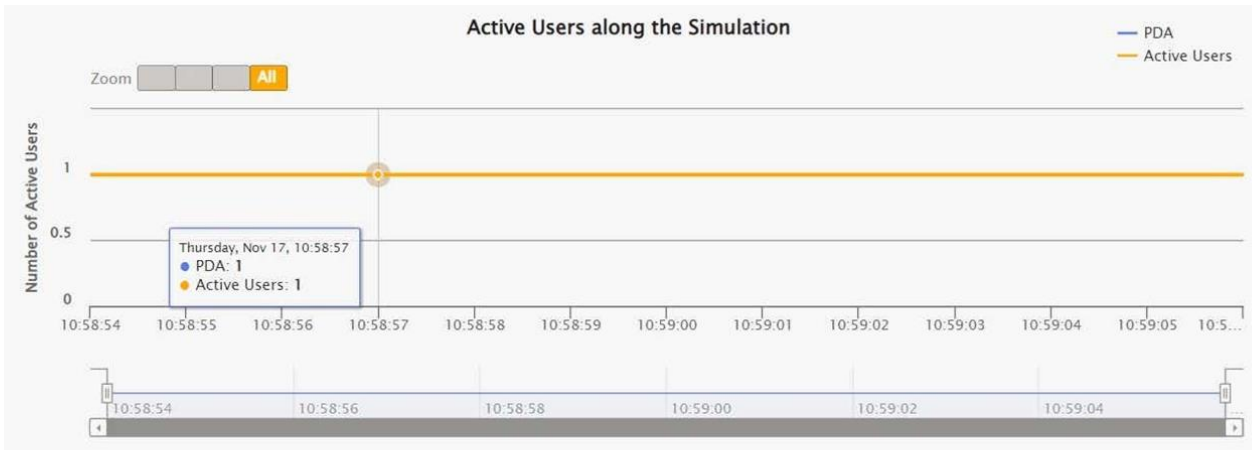
Description:

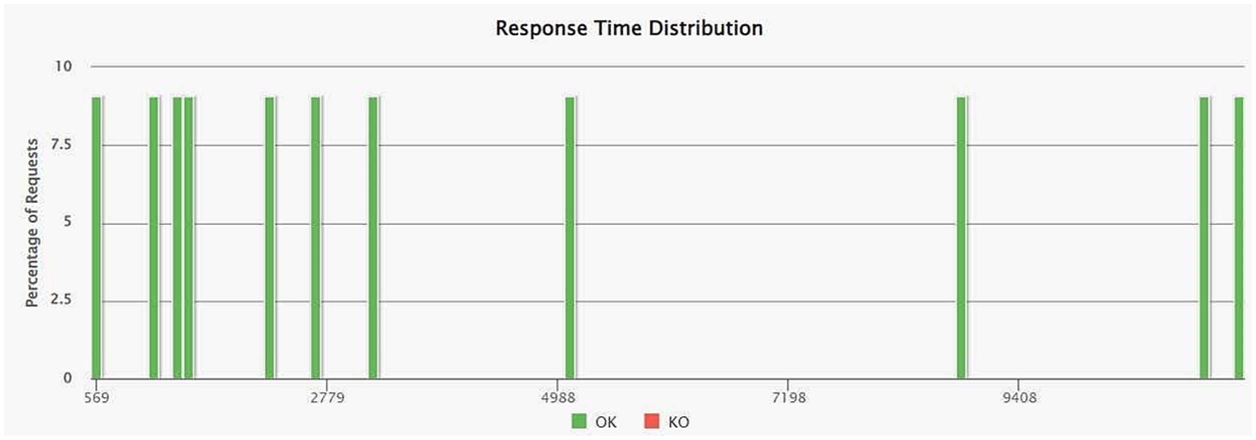
0

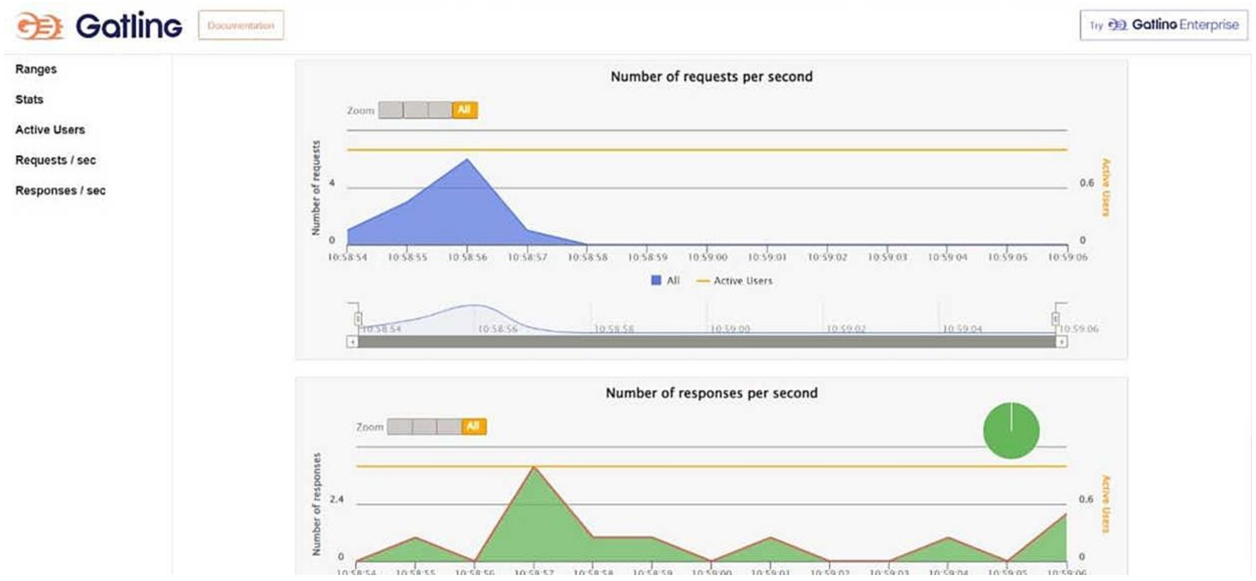


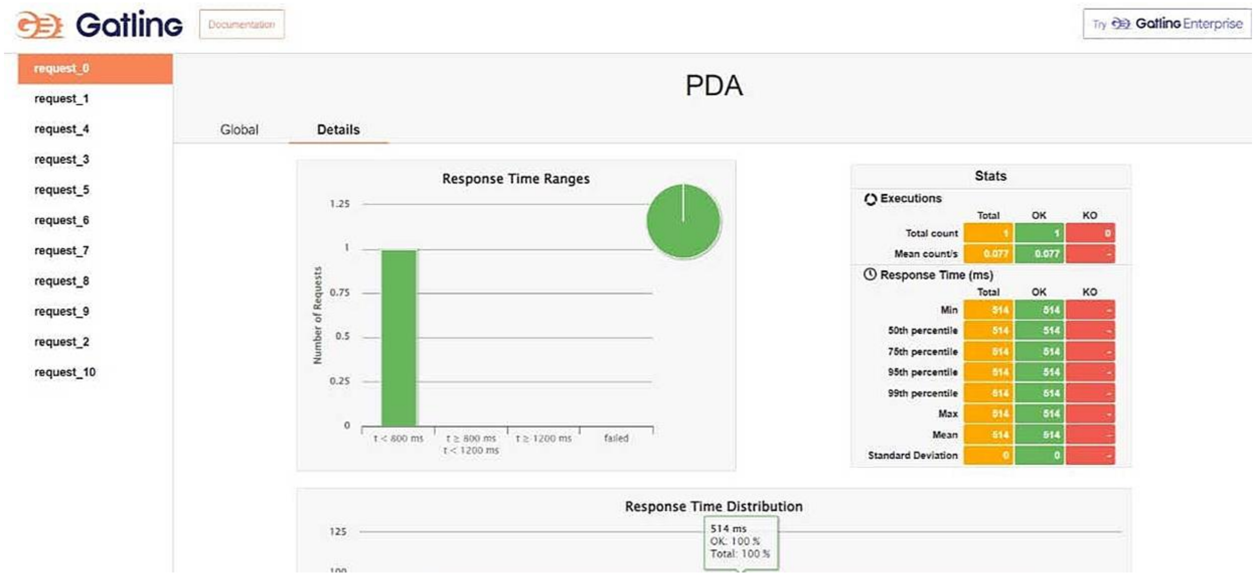


Requests	Executions					Response Time (ms)							
	Total	OK	KO	% KO	Cnt/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev
All Requests	11	11	0	0%	0.846	514	2627	7001	11358	11521	11562	4481	3932
request_0	1	1	0	0%	0.077	514	514	514	514	514	514	514	0
request_1	1	1	0	0%	0.077	2627	2627	2627	2627	2627	2627	2627	0
request_4	1	1	0	0%	0.077	2244	2244	2244	2244	2244	2244	2244	0
request_3	1	1	0	0%	0.077	11562	11562	11562	11562	11562	11562	11562	0
request_5	1	1	0	0%	0.077	8876	8876	8876	8876	8876	8876	8876	0
request_6	1	1	0	0%	0.077	1396	1396	1396	1396	1396	1396	1396	0
request_7	1	1	0	0%	0.077	3200	3200	3200	3200	3200	3200	3200	0
request_8	1	1	0	0%	0.077	11153	11153	11153	11153	11153	11153	11153	0
request_9	1	1	0	0%	0.077	1507	1507	1507	1507	1507	1507	1507	0
request_2	1	1	0	0%	0.077	5126	5126	5126	5126	5126	5126	5126	0
request_10	1	1	0	0%	0.077	1085	1085	1085	1085	1085	1085	1085	0


Active Users along the Simulation
PDA
Active Users
Zoom
All
Number of Active Users
1
0.5
0
Thursday, Nov 17, 10:58:57
PDA: 1
Active Users: 1
10:58:54
10:58:55
10:58:56
10:58:57
10:58:58
10:58:59
10:59:00
10:59:01
10:59:02
10:59:03
10:59:04
10:59:05
10:5...
10:58:54
10:58:56
10:58:58
10:59:00
10:59:02
10:59:04


Response Time Distribution
10
7.5
5
2.5
0
Percentage of Requests
569
2779
4988
7198
9408
OK
KO





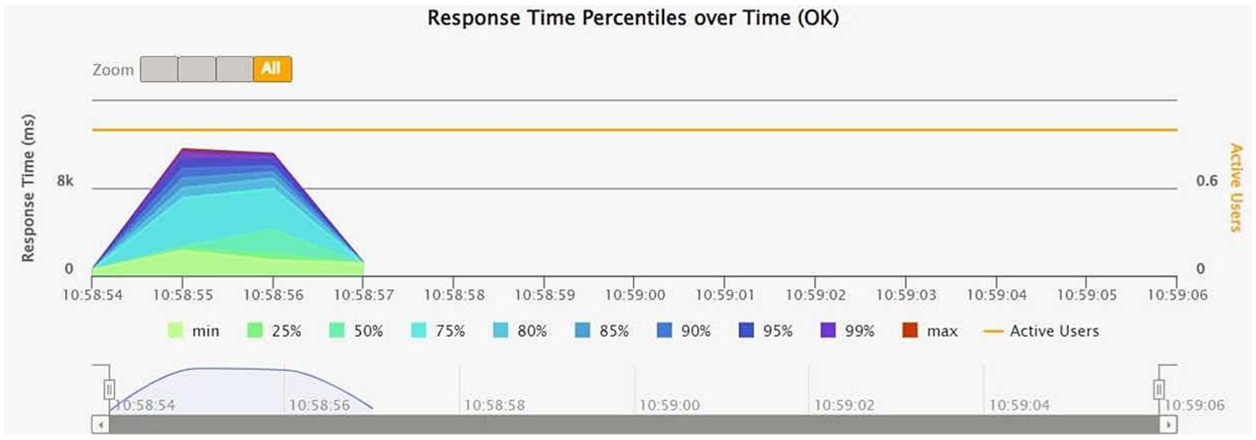
Stats

Executions

	Total	OK	KO
Total count	1	1	0
Mean count/s	0.077	0.077	-

Response Time (ms)

	Total	OK	KO
Min	514	514	-
50th percentile	514	514	-
75th percentile	514	514	-
95th percentile	514	514	-
99th percentile	514	514	-
Max	514	514	-
Mean	514	514	-
Standard Deviation	0	0	-



8.2 USER ACCEPTANCE TESTING

Purpose of Document

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

Defect Analysis

This report shows the number of resolvedor closed bugs at each severity level,and howthey were resolved

Resolution	**Severity1**	**Severity2**	**Severity3**	**Severity4**	**Subtotal**
By Design	12	4	2	2	20
Duplicate	1	1	2	0	4
External	2	4	0	1	7
Fixed	10	2	2	20	34
Not Reproduced	0	0	2	0	2
Skipped	0	0	2	1	3
Won't Fix	0	5	2	1	8
Totals	25	16	12	25	78

Test Case Analysis

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

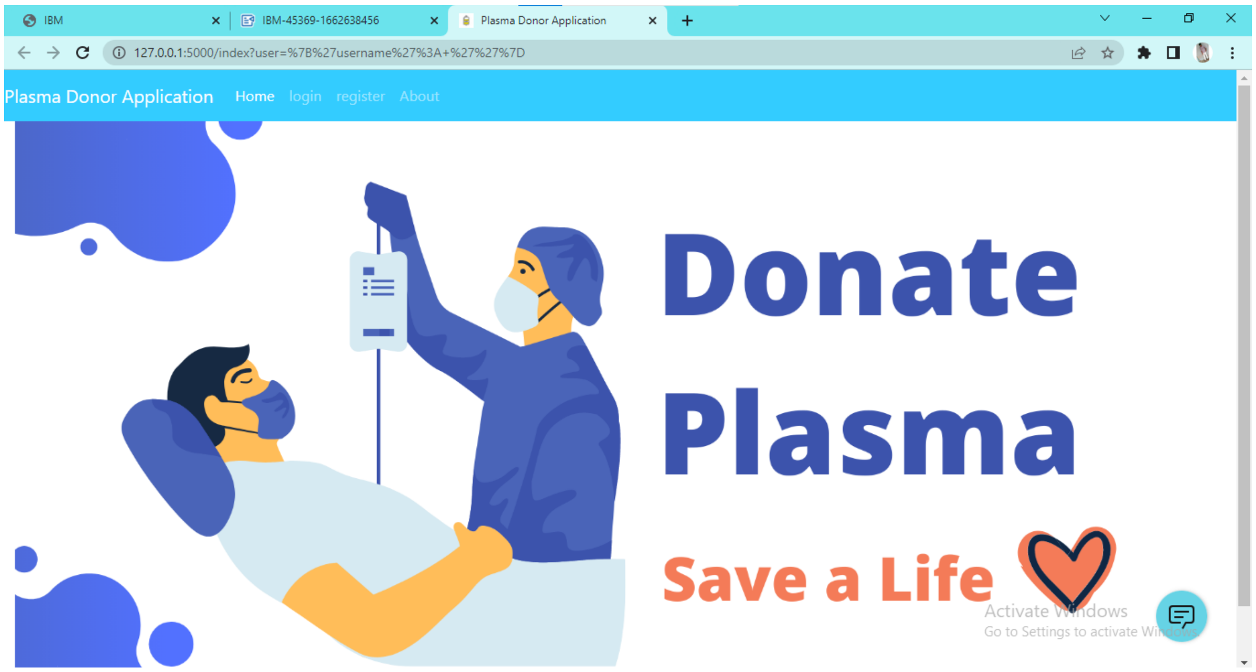
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	1	4
Client Application	47	0	2	45
Security	3	0	0	3
Outsource Shipping	2	0	0	2
Exception Reporting	11	0	2	9
Final Report Output	5	0	0	5
Version Control	2	0	1	2

CHAPTER 9

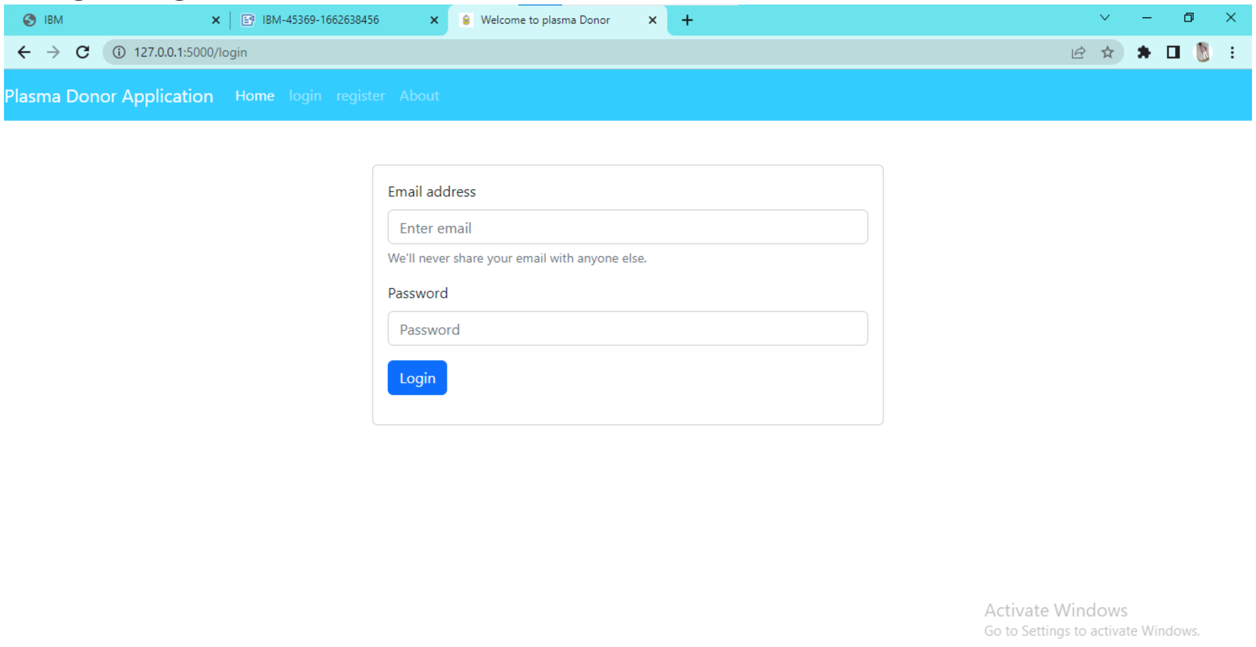
RESULTS

9.1 PERFORMANCE METRICS

Home Page



Login Page



Email address

Enter email

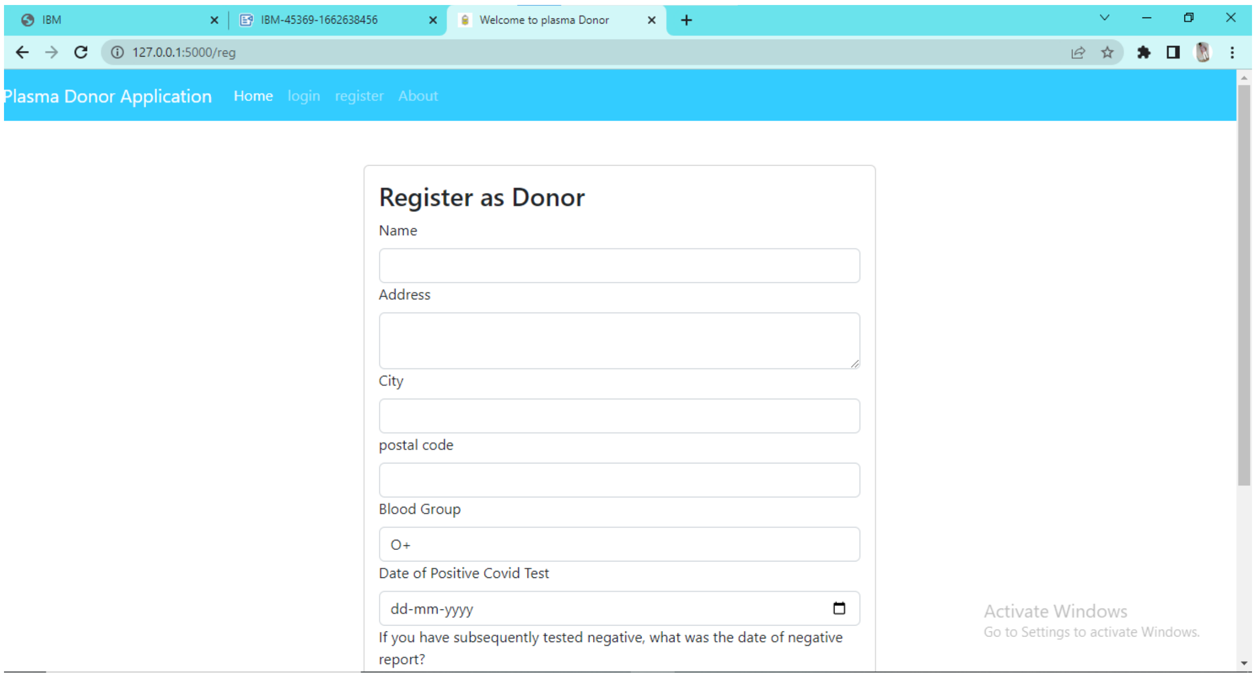
We'll never share your email with anyone else.

Password

Password

Login

Register Page



IBM | IBM-45369-1662638456 | Welcome to plasma Donor

127.0.0.1:5000/reg

Plasma Donor Application Home login register About

Register as Donor

Name

Address

City

postal code

Blood Group

O+

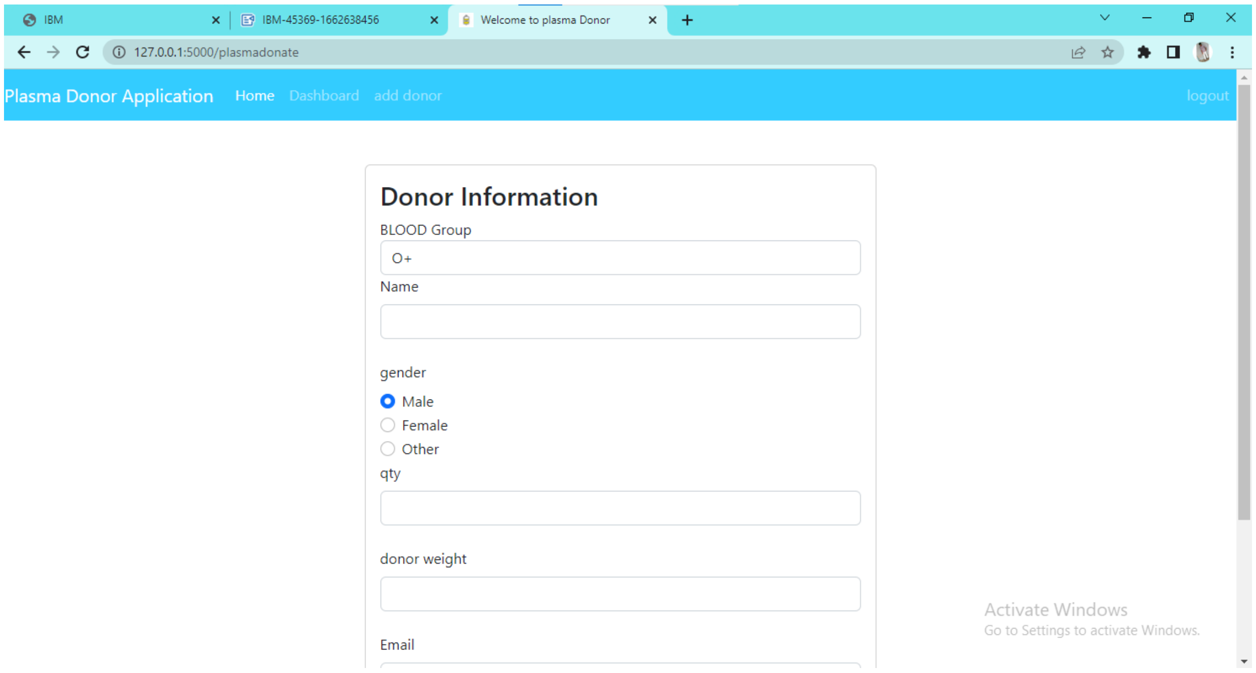
Date of Positive Covid Test

dd-mm-yyyy

If you have subsequently tested negative, what was the date of negative report?

Activate Windows
Go to Settings to activate Windows.

Donor Page



IBM | IBM-45369-1662638456 | Welcome to plasma Donor

127.0.0.1:5000/plasmadonate

Plasma Donor Application Home Dashboard add donor logout

Donor Information

BLOOD Group

O+

Name

gender

[✓ Male]
Female
Other

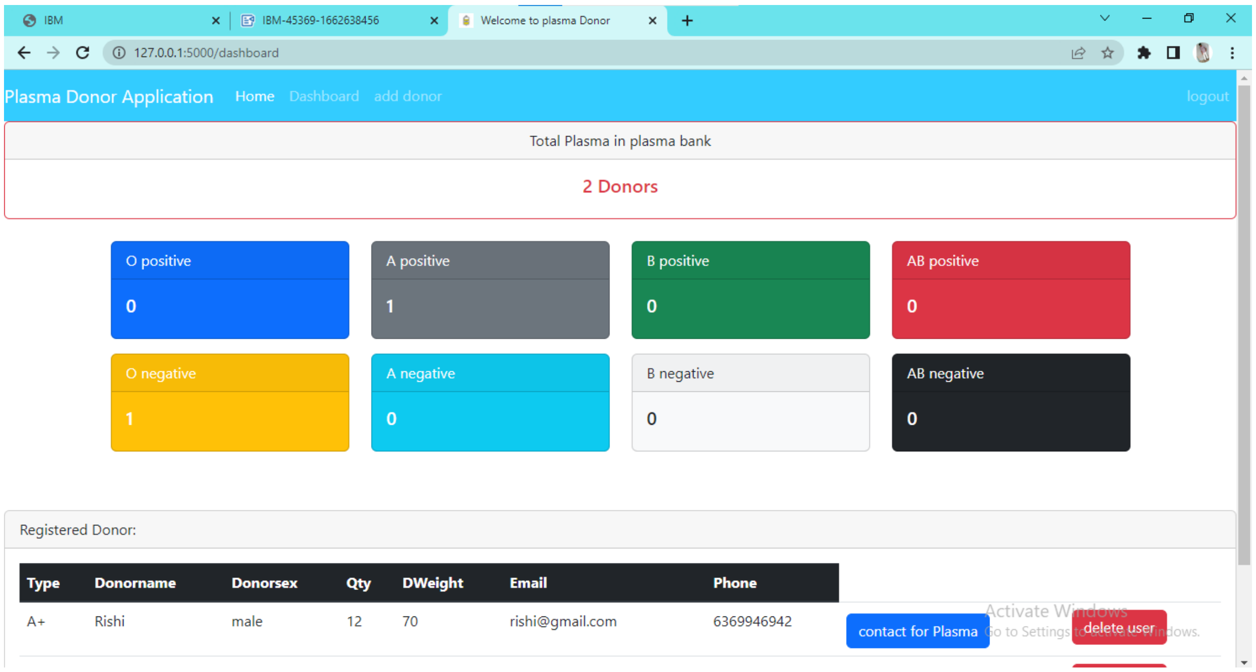
qty

donor weight

Email

Activate Windows
Go to Settings to activate Windows.

Dashboard Page



Total Plasma in plasma bank

2 Donors

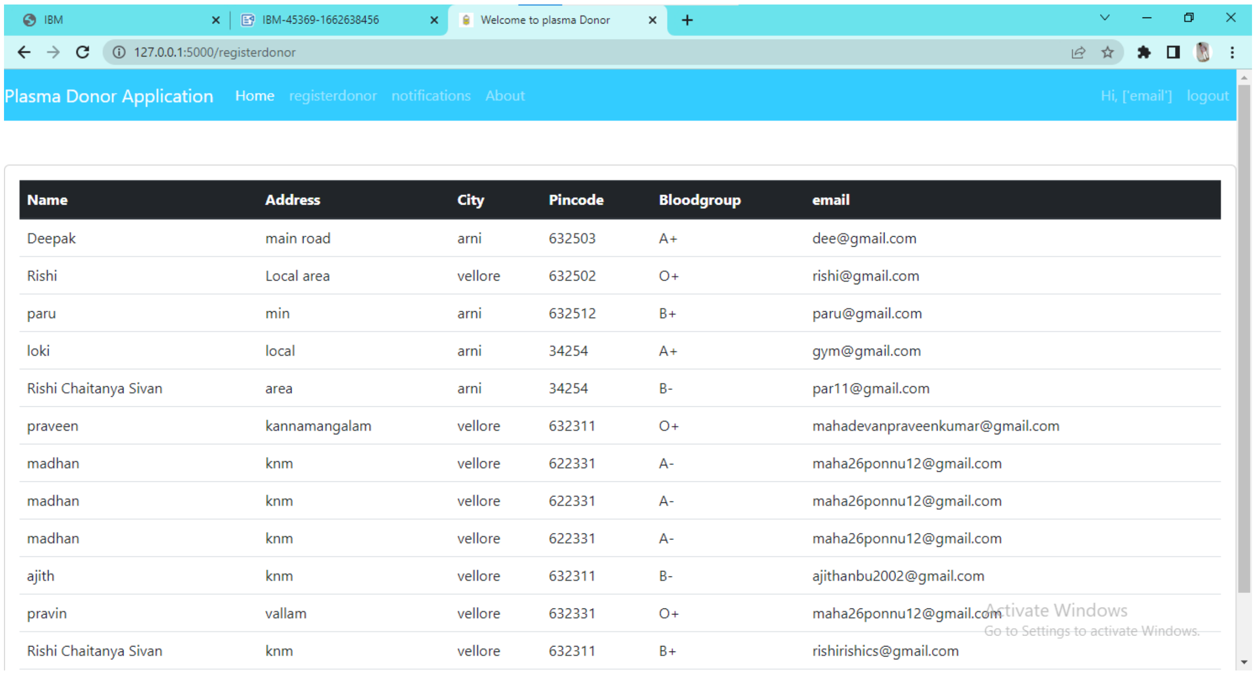
O positive	A positive	B positive	AB positive
0	1	0	0

O negative	A negative	B negative	AB negative
1	0	0	0

Registered Donor:

Type	Donorname	Donorsex	Qty	DWeight	Email	Phone
A+	Rishi	male	12	70	rishi@gmail.com	6369946942

Register Donor list



Name	Address	City	Pincode	Bloodgroup	email
Deepak	main road	arni	632503	A+	dee@gmail.com
Rishi	Local area	vellore	632502	O+	rishi@gmail.com
paru	min	arni	632512	B+	paru@gmail.com
loki	local	arni	34254	A+	gym@gmail.com
Rishi Chaitanya Sivan	area	arni	34254	B-	par11@gmail.com
praveen	kannamangalam	vellore	632311	O+	mahadevanpraveenkumar@gmail.com
madhan	knm	vellore	622331	A-	maha26ponnu12@gmail.com
madhan	knm	vellore	622331	A-	maha26ponnu12@gmail.com
madhan	knm	vellore	622331	A-	maha26ponnu12@gmail.com
ajith	knm	vellore	632311	B-	ajithanbu2002@gmail.com
pravin	vallam	vellore	632331	O+	maha26ponnu12@gmail.com
Rishi Chaitanya Sivan	knm	vellore	632311	B+	rishirishics@gmail.com

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- ***Speed*** : This website is fast and offers great accuracy as compared to manual registered keeping.
- ***Maintenance*** : Less maintenance is required
- ***User Friendly*** : It is very easy to use and understand. It is easily workable and accessible for everyone.
- ***Fast Results*** : It would help you to provide plasma donors easily depending upon the availability of it.
- It will help people to find plasma easily.
- It is a user-friendly application.

DISADVANTAGES

- ***Internet*** : It would require an internet connection for the working of the website.
- ***Auto- Verification*** : It cannot automatically verify the genuine users.

CHAPTER 11

CONCLUSION

Plasma is a liquid portion of blood; it is a mixture of water, proteins and salts. Antibodies are proteins made by the body in response to an infection. People fully rescued from COVID19 are encouraged to donate plasma, which can help to increase the lifespan of other patients because their plasma contains antigens which helps the affected person to recover faster. These immunoglobulin give your immune system away to fight the virus when you are sick, so your plasma can be used to help others fight off illness. Individuals must fully resolve symptoms for at least 14 days prior are eligible to donate

CHAPTER 12

FUTURE SCOPE

- The scope clearly defines the boundaries of the proposed system.

- The functional areas of this application that lies under the scope of the proposed system are the management of the availability of donors, hospitals, and blood banks to the user or member at any time.

- Upgrading the UI that is more user-friendly will help many users to access this app and also ensures that many plasma donors can be added to the

  community.
- Increasing a few features helps to handle multiple requests at the same time which will maintain the uptime of the website with negligible downtime.

CHAPTER 13

APPENDIX

SOURCE CODE

Application File

app.py

```
from flask import render_template
import requests
from flask import Flask
from flask import request,redirect,url_for,session,flash
from flask_wtf import Form
import ibm_db
conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=9938aec0-8105-433e-
8bf9-
0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32459;S
ECURITY=SSL;SSLservercertiicate=DigiCertGlobalRootCA.crt;UID=ymv17936;
PWD=4miHcNLYiftKlnkL",",")
from dotenv import load_dotenv
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail
import os

from wtforms import TextField
app = Flask(__name__)

load_dotenv()
FROM_EMAIL=os.getenv('FROM_EMAIL')
SENDG_API_KEY=os.getenv('SENDGRID_API_KEY')
IMB_DB_URL=os.getenv('IMB_DB_URL')
SECRET_KEY=os.getenv('SECRET_KEY')

app.secret_key=SECRET_KEY
```

```
@app.route('/')
def hel():

     if session.get('username')==True:
        messages = session['username']

     else:
        messages = ""
        user = {'username': messages}
     return redirect(url_for('index',user=user))



@app.route('/reg')
def add():
    return render_template('register.html')

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

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

   if request.method == 'POST':
         name = request.form['name']
         address = request.form['address']
         city = request.form['city']
         pincode = request.form['pincode']
         bloodgroup = request.form['bloodgroup']
         pdate = request.form['pdate']
         ndate = request.form['ndate']
         email = request.form['email']
         password = request.form['password']
         sql = "INSERT INTO USERS VALUES (?,?,?,?,?,?,?,?,?)"
         prep_stmt = ibm_db.prepare(conn, sql)
         ibm_db.bind_param(prep_stmt, 1, name)
         ibm_db.bind_param(prep_stmt, 2, address)
         ibm_db.bind_param(prep_stmt, 3, city)
         ibm_db.bind_param(prep_stmt, 4, pincode)
         ibm_db.bind_param(prep_stmt, 5, bloodgroup)
         ibm_db.bind_param(prep_stmt, 6, pdate)
         ibm_db.bind_param(prep_stmt, 7, ndate)
         ibm_db.bind_param(prep_stmt, 8, email)
         ibm_db.bind_param(prep_stmt, 9, password)
         ibm_db.execute(prep_stmt)
         message = Mail(
         from_email=FROM_EMAIL,
         to_emails=email,
         subject='Plasma Donor',
         html_content='<p>Hello, Your Registration was successfull. <br><br>
Thank you for choosing us.</p>')
         sg = SendGridAPIClient(
         api_key=SENDGRID_API_KEY)
         try:
           response=sg.send(message)
         except Exception:
               pass
         print("Mail Sent and response code is ",response.status_code)
         print("Inserted Successfully")
         return redirect(url_for('index'))



@app.route('/index',methods = ['POST', 'GET'])
def index():
    if request.method == 'POST':
     if session.get('username') is not None:
         messages = session['username']

     else:
         messages = ""
     user = {'username': messages}
     print(messages)
     val = request.form['search']
     print(val)
     type = request.form['type']
     print(type)
     if type=='blood':
      sql="SELECT * FROM users where bloodgroup =?",(val,)
      prep_stmt = ibm_db.prepare(conn, sql)
      ibm_db.bind_param(prep_stmt, 1, val)
      ibm_db.execute(prep_stmt)
      search = ibm_db.fetch_assoc(prep_stmt)

      sql="select * from users"
      prep_stmt = ibm_db.prepare(conn, sql)
      ibm_db.execute(prep_stmt)
      rows = ibm_db.fetch_assoc(prep_stmt)

      return render_template('index.html', title='Home',
user=user,rows=rows,search=search)

     if type=='donorname':
      sql="select * from users where name=?",(val,)
      prep_stmt = ibm_db.prepare(conn, sql)
      ibm_db.bind_param(prep_stmt, 1, val)
      ibm_db.execute(prep_stmt)
      search = ibm_db.fetch_assoc(prep_stmt)

      sql="select * from users"
```

```
         prep_stmt = ibm_db.prepare(conn, sql)
         ibm_db.execute(prep_stmt)
         rows = ibm_db.fetch_assoc(prep_stmt)

         return render_template('index.html', title='Home',
user=user,rows=rows,search=search)

      if session.get('username') is not None:
        messages = session['username']

      else:
        messages = ""
        user = {'username': messages}
        print(messages)
      if request.method=='GET':
         sql="select * from users"
         prep_stmt = ibm_db.prepare(conn, sql)
         ibm_db.execute(prep_stmt)
         rows = ibm_db.fetch_assoc(prep_stmt)

      return render_template('index.html', title='Home',  rows=rows)


@app.route('/login',methods = ['POST', 'GET'])
def login():
    if request.method == 'GET':
        return render_template('/login.html')
    if request.method == 'POST':
            email = request.form["email"]
            password = request.form["password"]

            if email == 'admin@plasmabank.com' and password == 'admin':
                a = 'yes'
                session['username'] = email
                #session['logged_in'] = True
                session['admin'] = True
                return redirect(url_for('index'))
                #print((password,email))
            sql = "SELECT * FROM USERS WHERE email= ? "
            prep_stmt = ibm_db.prepare(conn, sql)
            ibm_db.bind_param(prep_stmt, 1, email)
            ibm_db.execute(prep_stmt)
            email = ibm_db.fetch_assoc(prep_stmt)



            a = ["email"]
            session['username'] = a
            session['logged_in'] = True
            print(a)
            u = {'username': a}
            p = ['password']
            print(p)
            if email == a and password == p:
                return redirect(url_for('index'))
            else:
                return render_template('/index.html')



@app.route('/logout')
def logout():

   session.pop('username', None)
   session.pop('logged_in',None)
   try:
       session.pop('admin',None)
   except KeyError as e:
       print("I got a KeyError - reason " +str(e))
   return redirect(url_for('index'))



@app.route("/dashboard")
def dashboard():
    sql = "SELECT COUNT(*), (SELECT COUNT(*) FROM blood WHERE
type= 'O+'), (SELECT COUNT(*) FROM blood WHERE type='A+'), (SELECT
COUNT(*) FROM blood WHERE type='B+'), (SELECT COUNT(*) FROM
blood WHERE type='AB+'), (SELECT COUNT(*) FROM blood WHERE
type='O-'), (SELECT COUNT(*) FROM blood WHERE type='A-'), (SELECT
COUNT(*) FROM blood WHERE type='B-'), (SELECT COUNT(*) FROM blood
WHERE type='AB-') FROM blood"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print(account)



    users = []
    sql = "SELECT * FROM BLOOD"
    prep_stmt = ibm_db.exec_immediate(conn, sql)
    rows = ibm_db.fetch_assoc(prep_stmt)

    while rows != False:
        users.append(rows)
        rows = ibm_db.fetch_assoc(prep_stmt)

    if users:
        return render_template("dashboard.html",b=account ,users = users)



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

```
def bl():
    return render_template('/adddonor.html')



@app.route('/addb',methods =['POST','GET'])
def addb():
        msg = ""
        if request.method == 'POST':
           type = request.form['bloodgroup']
           donorname = request.form['donorname']
           donorsex = request.form['gender']
           qty = request.form['qty']
           dweight = request.form['dweight']
           email = request.form['email']
           phone = request.form['phone']
           sql = "INSERT INTO BLOOD VALUES (?,?,?,?,?,?,?)"
           prep_stmt = ibm_db.prepare(conn, sql)
           ibm_db.bind_param(prep_stmt, 1, type)
           ibm_db.bind_param(prep_stmt, 2, donorname)
           ibm_db.bind_param(prep_stmt, 3, donorsex)
           ibm_db.bind_param(prep_stmt, 4, qty)
           ibm_db.bind_param(prep_stmt, 5, dweight)
           ibm_db.bind_param(prep_stmt, 6, email)
           ibm_db.bind_param(prep_stmt, 7, phone)
           ibm_db.execute(prep_stmt)
           msg = "Record successfully added"
           print("Inserted Successfully")

        return redirect(url_for('dashboard'))




@app.route('/registerdonor')
def registerdonor():
    users = []
    sql = "SELECT * FROM USERS"
    prep_stmt = ibm_db.exec_immediate(conn, sql)
    rows = ibm_db.fetch_assoc(prep_stmt)

    while rows != False:
        users.append(rows)
        rows = ibm_db.fetch_assoc(prep_stmt)

    if users:
        return render_template("registerdonor.html" ,users = users)




@app.route('/contactforplasma/<emailid>')
def contactforplasma(emailid):
     if request.method=="GET":

        sql="SELECT * FROM REQUEST"
        prep_stmt = ibm_db.prepare(conn, sql)
        ibm_db.execute(prep_stmt)

        fromemail = session['username']
        name = request.form['name']
        address = request.form['address']

       # print(fromemail,emailid)
        sql=("INSERT INTO request (toemail,formemail,toname,toaddr) VALUES
(?,?,?,?)",(emailid,fromemail,name,address) )
        prep_stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param ( prep_stmt, 1,(emailid,fromemail,name,address))
        ibm_db.execute(prep_stmt)
        flash('request sent')
        return redirect(url_for('index'))

    if request.method == 'POST':

        fromemail = session['username']
        name = request.form['name']
        address = request.form['address']


        sql=("INSERT INTO request (toemail,formemail,toname,toaddr) VALUES
(?,?,?,?)",(emailid,fromemail,name,address) )
        prep_stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param ( prep_stmt, 1,(emailid,fromemail,name,address))
        ibm_db.execute(prep_stmt)

        flash('request sent')
        return redirect(url_for('index'))


@app.route('/notifications',methods=('GET','POST'))
def notifications():

    return render_template('notifications.html')



@app.route('/deleteuser/<useremail>',methods=['GET', 'POST'])
def deleteuser(useremail):
    if request.method == "GET":

        sql=f"DELETE FROM BLOOD WHERE EMAIL='{useremail}'"
        try:
            ibm_db.exec_immediate(conn,sql)
        except Exception:
            pass
        return redirect(url_for('dashboard'))



if __name__ == '__main__':
    app.run(debug=True)

Base.html
<html>
    <head>
        {% if title %}
        <title> Plasma Donor Application</title>
        {% else %}
        <title>Welcome to plasma Donor </title>
        {% endif %}
        <link rel="shortcut icon" href="{{url_for('static', filename='plasma.png')}}"/>
        <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdk
nLPMO" crossorigin="anonymous">
        <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo
" crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
integrity="sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPI
Pm49" crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
integrity="sha384-
```

```
ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stw
EULTy" crossorigin="anonymous"></script>

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi
" crossorigin="anonymous">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw
3" crossorigin="anonymous"></script>

<script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script src="{{url_for('static', filename='numscroller-1.0.js')}}"></script>
<style>div.someclass {
background-size: cover;
}
}</style>

   </head>
   <body>




     {% if session['admin'] == True %}

     <nav class="navbar navbar-expand-lg navbar-dark" style="background-
color:#33ccff;">



       <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-
```

```
controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <a class="navbar-brand" href="/"> Plasma Donor Application</a>
      <ul class="navbar-nav mr-auto">
        <li class="nav-item active">
          <a class="nav-link" href="/">Home <span class="sr-
only">(current)</span></a>
        </li>
      <li class="nav-item">
        <a class="nav-link" href="/dashboard">Dashboard</a>
      </li>

      <li class="nav-item">
        <a class="nav-link" href="/plasmadonate">add donor</a>
      </li>
      </ul>
      <ul class="navbar-nav ml-auto">
        <li class="nav-item">
          <a class="nav-link" href="{{url_for('logout')}}">logout</a>
        </li>
        </ul>
    </div>

  </nav>

      {% elif session['logged_in'] == True %}

      <nav class="navbar navbar-expand-lg navbar-dark" style="background-
color:#33ccff;">
```

```
        <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <a class="navbar-brand" href="/"> Plasma Donor Application</a>
      <ul class="navbar-nav mr-auto">
        <li class="nav-item active">
          <a class="nav-link" href="/">Home <span class="sr-
only">(current)</span></a>
        </li>

       <li class="nav-item">
        <a class="nav-link" href="{{url_for('registerdonor')}}">registerdonor</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="{{url_for('notifications')}}">notifications</a>
      </li>
      <li class="nav-brand">
        <a class="nav-link" href="/about">About</a>
      </li>
      </ul>



      <ul class="navbar-nav ml-auto">

    <li class="nav-item">
      <a class="nav-link" href="#"> Hi, {{ session['username']}}</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="{{url_for('logout')}}">logout</a>
    </li>
    </ul>

  </div>

</nav>



      {%else%}

      <nav class="navbar navbar-expand-lg navbar-dark" style="background-
color:#33ccff;">



        <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <a class="navbar-brand" href="/"> Plasma Donor Application</a>
      <ul class="navbar-nav mr-auto">
        <li class="nav-item active">
          <a class="nav-link" href="/">Home <span class="sr-
only">(current)</span></a>
        </li>
      <li class="nav-brand">
        <a class="nav-link" href="/login">login</a>
      </li>
      <li class="nav-brand">
```

```
        <a class="nav-link" href="{{url_for('add')}}">register</a>
      </li>
     <li class="nav-brand">
       <a class="nav-link" href="/about">About</a>
      </li>

    </ul>
  </div>

</nav>



    {%endif%}





          {% block content %}

           {% endblock %}

      </body>


<script type="text/javascript">
    function send_notification_clicked(email)
    {
    var element = document.getElementById("contactform");
    element.setAttribute("action",  '/contactforplasma/'+email);
    }
    </script>
</html>
```

index.html

```
{% extends "base.html" %} {% block content %}

<div class="container-fluid">
<img style="width:100%" src="{{url_for('static', filename='pd.png')}}"
class="card-body">
</div>
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "2de27163-e2ea-49e2-9c86-de9b90e62919", // The ID of
this integration.
    region: "us-south", // The region your integration is hosted in.
    serviceInstanceID: "d239becd-5476-49ab-87de-48ad7ac31118", // The ID
of your service instance.
    onLoad: function(instance) { instance.render(); }
  };
  setTimeout(function(){
    const t=document.createElement('script');
    t.src="https://web-
chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
    document.head.appendChild(t);
  });
</script>



 {% endblock %}
```

login.html

```
{% extends "base.html" %}
{% block content%}

<br>
<br>
<center>
<div class="card text-left" style="width: 35rem;">
  <div class="card-body">

    <form action = "{{url_for('login')}}" method = "POST">
<div class="form-group">
  <label for="exampleInputEmail1">Email address</label>
  <input type="email"  name ="email" class="form-control"
id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter
email" required>
  <small id="emailHelp" class="form-text text-muted">We'll never share your
email with anyone else.</small>
</div>
<div class="form-group">
  <label for="exampleInputPassword1">Password</label>
  <input type="password"  name ="password" class="form-control"
id="exampleInputPassword1" placeholder="Password" required>
</div>
<button type="submit" class="btn btn-primary">Login</button>
</form>

  </div>
</div>
</center>




{% endblock %}
```

register.html

```
{% extends "base.html" %}

{% block content %}
<br>
<br>
<center>
<div class="card text-left" style="width: 35rem;">
  <div class="card-body">
      <div class="form-group">
     <form action = "{{url_for('addrec')}}" method = "POST">
       <h3>Register as Donor</h3>
       <label for="name">Name</label>
       <input type = "text" name = "name" class="form-control" required/>

        <label for="addr">Address</label>
       <textarea name = "address" class="form-control" required></textarea>

        <label for="city">City</label>
       <input type = "text" name = "city" class="form-control" required/>

        <label for="pin">postal code</label>
       <input type = "text" name = "pincode" class="form-control" required/>

       <label for="Bloodgroup">Blood Group</label>
       <select name="bloodgroup"  class="form-control"
id="exampleFormControlSelect1">
         <option value="O+" selected>O+</option>
         <option value="O-">O-</option>
         <option value="A+">A+</option>
         <option value="A-">A-</option>
         <option value="B+">B+</option>
         <option value="B-">B-</option>
```

```
          <option value="AB+">AB+</option>
          <option value="AB-">AB-</option>
        </select>
        <label for="postivedate">Date of Positive Covid Test </label>
        <input type = "date" name ="pdate" class="form-control" required/>

        <label for="negativedate">If you have subsequently tested negative, what
was the date of negative report?</label>
        <input type = "date" name ="ndate" class="form-control" required/>

        <label for="exampleInputEmail1">Email address</label>
        <input type = "text" name ="email" class="form-control" required/>

        <label for="exampleInputPassword1">Password</label>
        <input type = "password" name ="password" class="form-control"
required/>
        <br>
        <button type="submit" class="btn btn-primary">Register</button>
      </form>
      </div>
      </div>
      </center>
      {% with messages = get_flashed_messages() %}
      {%if messages%}
         {%for mess in messages%}
           {{mess}}
         {%endfor%}
      {%endif%}
      {% endwith %}


{% endblock %}
```

dashboard.html

```
<!doctype html>
{% extends "base.html" %}
{% block content %}

{% with messages = get_flashed_messages() %}
{%if messages%}
   {%for mess in messages%}
  <div class="alert alert-warning alert-dismissible fade show" role="alert">
<strong>{{mess}}</strong>
<button type="button" class="close" data-dismiss="alert" aria-label="Close">
  <span aria-hidden="true">&times;</span>
</button>
</div>

   {%endfor%}
{%endif%}
{% endwith %}

    {% if session['logged_in'] == True %}

    {%else%}

    {%endif%}


<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.
```

```
css" rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WT
Ri" crossorigin="anonymous">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle
.min.js" integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbs
w3" crossorigin="anonymous"></script>
 </head>
 <body>

  <div class="card border-danger text-center">
   <div class="card-header">Total Plasma in plasma bank</div>
   <div class="card-body text-danger">
    <h5 class="card-title"><span class='numscroller' data-min='1'  data-
delay='5' data-increment='10'>{{b['1']}}</span> Donors</h5>
   </div>
  </div>
  <br>



    <div class="container">
 <div class="row">
  <div class="col"><div class="card text-white bg-primary mb-3" style="max-
width: 18rem;">
   <div class="card-header">O positive</div>
  <div class="card-body">
   <h5 class="card-title"><span class='numscroller' data-min='1' data-
delay='5' data-increment='10'>{{b['2']}}</span> </h5>

  </div>
 </div></div>
```

```
  <div class="col"><div class="card text-white bg-secondary mb-3"
style="max-width: 18rem;">
    <div class="card-header">A positive</div>
  <div class="card-body">
    <h5 class="card-title"><span class="count"><span class='numscroller'
data-delay='5' data-increment='10'>{{b['3']}}</span></span> </h5>

  </div>
  </div></div>
  <div class="col"><div class="card text-white bg-success mb-3" style="max-
width: 18rem;">
    <div class="card-header">B positive</div>
  <div class="card-body">
    <h5 class="card-title"><span class='numscroller' data-min='1'  data-
delay='5' data-increment='10'>{{b['4']}}</span> </h5>

  </div>
  </div></div>
  <div class="col"><div class="card text-white bg-danger mb-3" style="max-
width: 18rem;">
    <div class="card-header"> AB positive</div>
  <div class="card-body">
    <h5 class="card-title"><span class='numscroller' data-min='1'  data-
delay='5' data-increment='10'>{{b['5']}}</span> </h5>
    <p class="card-text"></p>
  </div>
  </div></div>

  <div class="w-100"></div>

  <div class="col"><div class="card text-white bg-warning mb-3" style="max-
width: 18rem;">
    <div class="card-header">O negative</div>
  <div class="card-body">
```

```
     <h5 class="card-title"><span class='numscroller' data-min='1'  data-
delay='5' data-increment='10'>{{b['6']}}</span> </h5>

   </div>
   </div></div>
   <div class="col"><div class="card text-white bg-info mb-3" style="max-
width: 18rem;">
     <div class="card-header">A negative</div>
   <div class="card-body">
     <h5 class="card-title"><span class="count"><span class='numscroller'
data-min='1'  data-delay='5' data-
increment='10'>{{b['7']}}</span></span></h5>

   </div>
   </div></div>
   <div class="col"><div class="card bg-light mb-3" style="max-width:
18rem;">
     <div class="card-header">B negative</div>
   <div class="card-body">
     <h5 class="card-title"><span class="count"><span class='numscroller'
data-min='1'  data-delay='5' data-increment='10'>{{b['8']}}</span> </h5>

   </div>
   </div></div>
   <div class="col"><div class="card text-white bg-dark mb-3" style="max-
width: 18rem;">
     <div class="card-header">AB negative</div>
   <div class="card-body">
     <h5 class="card-title"><span class="count"><span class='numscroller'
data-min='1'  data-delay='5' data-increment='10'>{{b['9']}}</span></h5>

   </div>
   </div></div>
  </div>
```

```
</div>


    <br><br>
    <div class="card">
      <div class="card-header">
       Registered Donor:
      </div>
      <div class="card-body">
        <table class="table">
          <thead class="thead-dark">
            <tr>
              <th scope="col">Type</th>
              <th scope="col">Donorname</th>
              <th scope="col">Donorsex</th>
              <th scope="col">Qty</th>
              <th scope="col">DWeight</th>
              <th scope="col">Email</th>
              <th scope="col">Phone</th>



            </tr>
          </thead>
          <tbody>
            <tr>
              {% for user in users %}

              <td>{{user['TYPE']}}</td>
              <td>{{user['DONORNAME']}}</td>
              <td>{{user['DONORSEX']}}</td>
              <td>{{user['QTY']}}</td>
              <td>{{user['DWEIGTH']}}</td>
              <td>{{user['EMAIL']}}</td>
          <td>{{user['PHONE']}}</td>

         <td><button type="button" class="btn btn-primary mt-1" data-
toggle="modal" data-target="#exampleModalCenter">
          contact for Plasma
         </button>
         <div class="modal fade" id="exampleModalCenter" tabindex="-1"
role="dialog" aria-labelledby="exampleModalCenterTitle" aria-
hidden="true">
          <div class="modal-dialog modal-dialog-centered" role="document">
           <div class="modal-content">
            <div class="modal-header">
             <h5 class="modal-title" id="exampleModalCenterTitle">contact for
plasma</h5>
             <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
              <span aria-hidden="true">&times;</span>
             </button>
            </div>
            <div class="modal-body">
             <form method="POST"
action="{{url_for('contactforplasma',emailid=user['email'])}}">
              <label for="name">Name</label>
              <input type = "text" name = "name"
value="admin@bloodbank.com" class="form-control" required/>
             <label for="address"> confirm your Address</label>
            <input type="text" name="address" class="form-control"
value="admin's address" required></textarea>
            <button type="button" class="btn btn-secondary mt-1" data-
dismiss="modal">Close</button>
             <button type="submit" class="btn btn-primary mt-1">send
request</button>
            </div>
           </div>
```

```
              </div>
            </div>
          </td>
                          <td><a href =
"{{url_for('deleteuser',useremail=user['EMAIL'])}}" class="btn btn-
danger">delete user</a></td>
                        </tr>

            {% endfor %}













            </tr>
          </tbody>
        </table>
      </div>
    </div>

    <br><br>

  </body>
</html>

{% endblock %}
```

adddonor.html

```
 <!doctype html>
{% extends "base.html" %}
{% block content %}
<center>

  {% with messages = get_flashed_messages() %}
  {%if messages%}
      {%for mess in messages%}
    <div class="alert alert-warning alert-dismissible fade show" role="alert">
  <strong>{{mess}}</strong>
  <button type="button" class="close" data-dismiss="alert" aria-label="Close">
    <span aria-hidden="true">&times;</span>
  </button>
</div>

      {%endfor%}
  {%endif%}
  {% endwith %}

<div class="card text-left mt-5" style="width: 35rem;">
  <div class="card-body">
      <form action = "{{url_for('addb')}}" method = "POST">
        <h3>Donor Information</h3>
        BLOOD Group<br>

        <select name="bloodgroup" class="form-control">
           <option value="O+" selected>O+</option>
          <option value="O-">O-</option>
          <option value="A+">A+</option>
          <option value="A-">A-</option>
          <option value="B+">B+</option>
          <option value="B-">B-</option>
          <option value="AB+">AB+</option>
```

```
           <option value="AB-">AB-</option>
          </select>

          <label for="name">Name</label>
          <input type = "text" name = "donorname" class="form-control"
required/><br>

          <label for="gender">gender</label>
          <div class="form-check">
  <input class="form-check-input" type="radio" name="gender"
id="exampleRadios1" value="male" checked>
  <label class="form-check-label" for="exampleRadios1">
    Male
  </label>
</div>
<div class="form-check">
  <input class="form-check-input" type="radio" name="gender"
id="exampleRadios2" value="female">
  <label class="form-check-label" for="exampleRadios2">
    Female
  </label>
</div>
<div class="form-check">
  <input class="form-check-input" type="radio" name="gender"
id="exampleRadios2" value="other"
  <label class="form-check-label" for="exampleRadios2">
    Other
  </label>
</div>

<label for="qty">qty</label>

          <input type = "int" name = "qty" class="form-control"  required/><br>
<label for="dweight">donor weight</label>
```

```
        <input type = "text" name ="dweight" class="form-control"
required/><br>

  <label for="email">Email</label>
        <input type = "text" name ="email" class="form-control"  required/><br>

      <label for="phone">Phone</label>
        <input type = "text" name ="phone" class="form-control"
required/><br>
        <input type = "submit" value = "submit" class="btn btn-primary" /><br>

      </form>
    </div>
  </div>
</center>
    {% endblock %}
```

registerdonor.html

```
<!doctype html>
{% extends "base.html" %}
{% block content %}

{% with messages = get_flashed_messages() %}
{%if messages%}
    {%for mess in messages%}
  <div class="alert alert-warning alert-dismissible fade show" role="alert">
<strong>{{mess}}</strong>
<button type="button" class="close" data-dismiss="alert" aria-label="Close">
  <span aria-hidden="true">&times;</span>
</button>
</div>
```

```
  {%endfor%}
{%endif%}
{% endwith %}

   {% if session['logged_in'] == True %}

   {%else%}

   {%endif%}


<html lang="en">
 <head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Bootstrap demo</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.
css" rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WT
Ri" crossorigin="anonymous">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle
.min.js" integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbs
w3" crossorigin="anonymous"></script>
 </head>
 <body>

<br><br>
```

```
<div class="card">

      <div class="card-body">
        <table class="table">
          <thead class="thead-dark">
            <tr>
              <th scope="col">Name</th>
              <th scope="col">Address</th>
              <th scope="col">City</th>
              <th scope="col">Pincode</th>
              <th scope="col">Bloodgroup</th>
              <th scope="col">email</th>




            </tr>
          </thead>
         <tbody>
            <tr>
              {% for user in users %}

              <td>{{user['NAME']}}</td>
              <td>{{user['ADDRESS']}}</td>
              <td>{{user['CITY']}}</td>
              <td>{{user['PINCODE']}}</td>
              <td>{{user['BLOODGROUP']}}</td>
              <td>{{user['EMAIL']}}</td>

            </tr>
            {% endfor %}

            </tr>
          </tbody>
        </table>
```

```
     </div>
    </div>

    <br><br>


  </body>
</html>

{% endblock %}
```

GITHUB & PROJECT DEMO LINK

GITHUB LINK

https://github.com/IBM-EPBL/IBM-Project-45369-1660729652

DEMO VIDEO LINK

https://drive.google.com/file/d/1LC3qbe3HaC8dPZbSDEs62fCB6HYJ7Mbf/view?usp=drivesd

[DEMO VIDEO LINK]