Project Report Format

INTRODUCTION

INTRODUCTION

1.1 PROJECT OVERVIEW

Plasma is the clear, straw-colored liquid portion of blood that remains after red blood cells, white blood cells, platelets, and other cellular components are removed. Plasma is composed of 90% water, plasma is a transporting medium for cells and a variety of substances vital to the human body. Plasma carries out a variety of functions in the body, including clotting blood, fighting diseases, and other critical functions. Plasma donation requires a commitment both in the amount of time for each donation and the frequency of donation. Typically it takes between one and three hours to donate source plasma, and plasma can be donated twice within a seven-day period. Whole plasma donation takes less time under 30 minutes and donors donate less frequently no more than once in eight weeks. The programs may fit into a donor's life differently at various times in the donor's life and are equally important in helping to fulfill a vital medical need. Source plasma is plasma that is collected from healthy, voluntary donors through a process called plasmapheresis and is used exclusively for further manufacturing into final therapies (fractionation). Source plasma donors may be compensated for their time and effort. Recovered plasma is collected through whole blood donation in which plasma is separated from its cellular components. Recovered plasma may be used for fractionation. The plasma protein therapeutics industry supports volunteerism donation in all of its forms. Source plasma donation and blood donation are critically important activities that contribute to saving lives. Source plasma and recovered plasma are used to produce therapies that treat people with rare, chronic diseases and disorders such as primary immunodeficiency, hemophilia, and genetic lung disease, as well as in the treatment of trauma, burns, and shock. Whole plasma donations most often are used locally in hospitals for transfusions required during surgery or other medical treatment.

1.2 PURPOSE:

Many people are familiar with the benefits and the process of blood donation; however, few people understand the importance of plasma donation. Plasma is the pale, yellow portion of the blood. Nearly 50% of blood is made up of plasma , which itself contains water, proteins and salts. Plasma plays the critical role of maintaining a healthy blood pressure, blood volume and a proper pH balance. Without plasma, our body would not be supplied with many of the proteins that are necessary to support blood clotting and our immune system responses. In addition, plasma carries many of the electrolytes that our muscles need to function properly and support our activities of daily living .Our application saves life. It helps humans during emergency.

This app reduces the time lag between searching for and contacting different blood donors across the country in just few mins. Donor will be prompted to enter an individual's details, like name, phone number, and blood type. After that the contact details will be saved in the database. At the time of emergency requirement, one does not need to register but he can quickly check for contacts matching a particular or related blood group and reach out to them via phone call with the help of our app. This app provides list of donors in the city/area and also provides a google map function which shows all the donors and one can track them easily. Since almost everyone carries a mobile phone with them, it ensures instant location tracking and communication.

LITERATURE SURVEY

2.1 EXISTING PROBLEM

There are many people who are willing to donate plasma and who need plasma. But there is not any easy way to access to find plasma donation centers nearby. So, the problem is not the lack of donors, but finding the matching donor at the right time. If someone needs plasma, they seek plasma first from neighbors, then from hospitals, and then from the nearest plasma bank. If they can't process plasma in these ways, it's very difficult for them to contact another for a short-term plasma draw. This is a problem that I want to solve through this application. Instead of just providing plasma to people in need with an outdated list of regular plasma donors who may or may not be available to help, This application reaches the right donor at the time of emergency.

2.2 REFERENCES

LITERATURE REVIEW

Arunkumar Chinnaswamy, Gurusankar Gopalakrishnan, Shabala Natarajan(2015). A study on Automation of Blood donor classification and Notification Techniques. This paper presents the increasing demand of blood donor in the field of healthcare related to automation processes. The present scenario tells us that blood donation services are manual and the demand for the blood is stably on the rise. Meanwhile, the number of voluntary donors is decreasing over the last few years. To improve this blood donor, automation and notification methods came to connect communication through all over the world. In this paper, we compare the various implementation and previous research done on this techniques.

Sumazly Sulaiman, Abdul Aziz K.Abdul Hamid, Nurul Ain Najihah Yusri (2016). Development of a blood bank management system. This paper tells us about the development of blood bank system. There are 3 systems for blood bank management system. They are Blood Bank India, Lions Blood Bank & Research Foundation(LBBRF) and BBMS standalone version. The Blood Bank India is a website that provides the facility for the donor to register by him/himself as a blood donor. This website is only for Indian citizen can register to the system. It provides a feature where a person or hospital can request the blood stock from BBI. LBBRF is a private organisation that provides a place to donate blood. They will conduct an event and here the donor or public people can donate the blood. They will also inform when is their next event to the donor, public people or in their website. The standalone system uses the Microsoft as the database of the system. It contains user account management, view stock list, donor registration and customer registration.

Radha R. Mahalle, S. S. Thorat(2018). Smart Blood Bank Based on IOT. In this paper describes, blood is very important in the medical field. The main purpose of the blood bank is to provide the blood to the patients with minimal blood transfusion error. As the blood bank management system consists of number of manual steps, so it becomes difficult to the blood bank to provide a large level of accuracy, reliability and automation in blood storage as well as transfusion process. This IOT based system will improve the response time of the blood bank by connecting all the blood banks to cloud storage. The use of IOT system will provide benefits for blood bank.

Muddu krishna. G, Nagaraju. S(2016). Design and implementation of short message services (SMS) based blood bank. This paper describes about shor

message services based blood bank system. It consists of two types as data processing and packet account. The data processing type responds the user request and the packet count checks the availability of the blood samples. After that the user can communicate with the system via SMS whenever in-person required blood then that person has to send a request to the system via SMS. Thenthe system will respond to these request and send SMS including the address of blood bank which having availability of the blood stock. If the blood stock is notavailable then the donor’s contact number will be sent to the patient.

Anish Hamlin M R, Albert Mayan J(2016). Blood Donation and Life Saver- Blood Donation App. This paper develops an application for finding the blood donation for making a request for the blood. If any blood seeker would login to the given application using GIS the patient will get detail about the nearby blood donor. Also, any blood donor can add themselves for donating the blood then he/she will receive the notification related to the blood donation camp. In this app all the blood banks are connected to the cloud storage. The cloud storage provides the real time information related to the available blood stock in every blood bank. If the blood is out of stock then the system will provide the contact details of the blood donors of different blood groups.

Shweta Pai, Zubair Hasan, Madhusmita Jena(2020). Green Colored Plasma Discovered in a Male Blood Donor. This paper tells about the green coloured plasma found in a male donor. Plasma is the largest part of our blood, the rest is taken up by the formed elements. Normally, the plasma is yellowish in colour thisis not always , there have been range from yellow to orange to even brown plasma. This is due to the presence of factors such as bilirubin, hemoglobin, carotenoids and iron transferrin. Recently, we have been reported a highlyunusual sample of green coloured plasma in our blood bank. This paper clearly explains that how they found the green coloured plasma and how they evaluated.

Kalpana Devi Guntoju, Tejaswini Jalli, Sreeja Uppala(2022). Instant Plasma Donor Recipient Connector Web Application. This paper presents about the instant plasma donor. From the end of 2019, the world is suffering from the

COVID 19 and till now no vaccine has been found for this pandemic situation. There is another way in which we can help people affected by COVID 19 by donating plasma from recovered patients. The COVID 19 positive tested patients can recover by the treatment of plasma therapy and it help them faster recovery. In the guidance of the system, the donor who wants to donate plasma can donate by uploading their COVID 19 certificate. Then the blood bank can see the donors who have uploaded the COVID 19 certificate and they can make a request to the donor and the hospital can register/login the website.

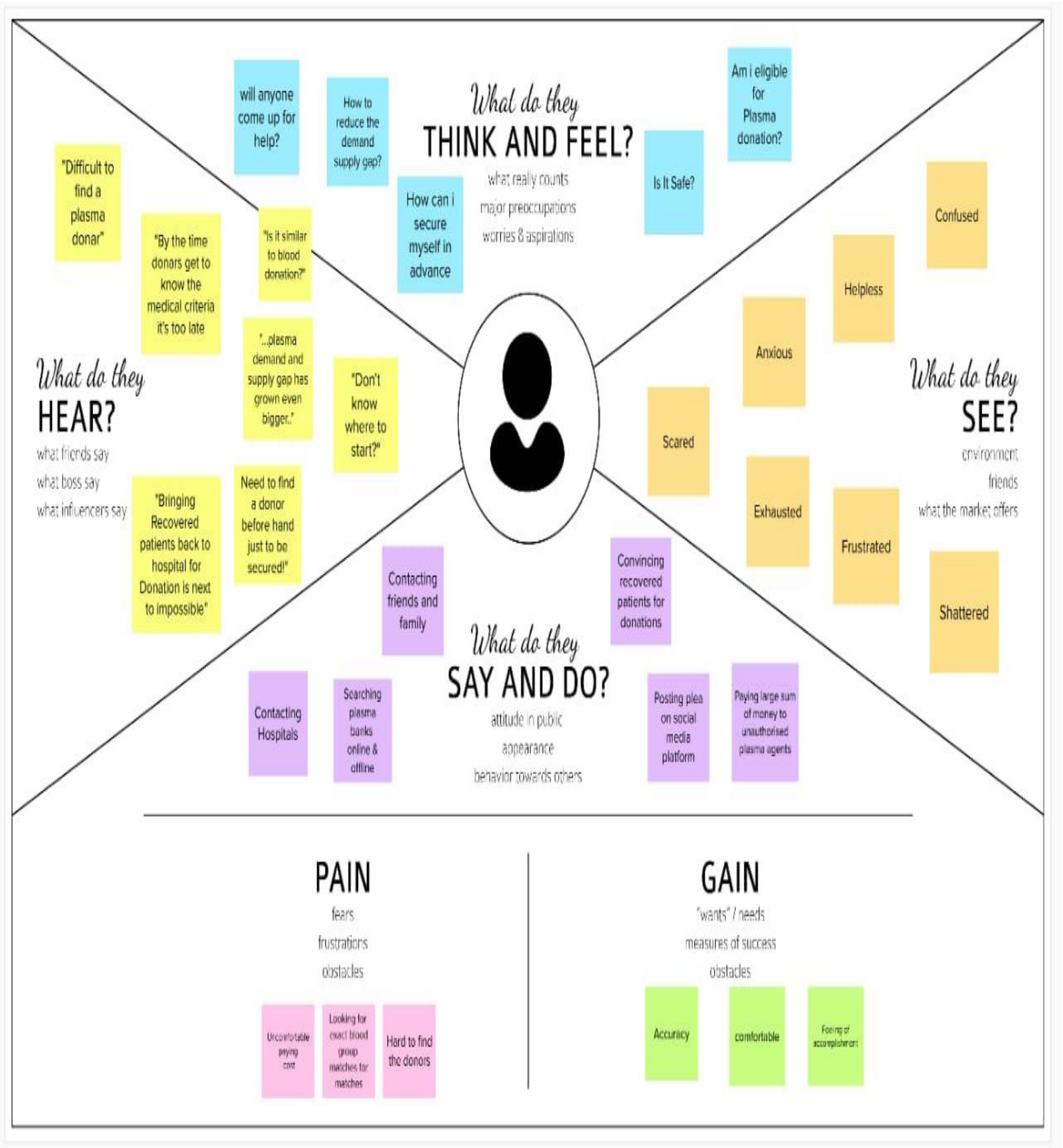
2.3 PROBLEM STATEMENT DEFINITION

It is an experimental online plasma donor management system. Donating plasmais a voluntary act that saves many lives. Donating plasma is similar to donating blood. A plasma therapy approach that helps many donors and recipients to recover from the issue. It is a safe and promising approach to connecting through an online application. By using this mobile application they can easily approach the donor. This system is used if anyone is in need of plasma. In the present scenario, it is having a lot of manual work through our application it is time- consuming. The system comprises of admin and user where both can request a plasma. The person who is active in this app is called an active member of the app. An individual user will enroll his/her details such as name, phone number, age, weight, date of birth, blood group, address, etc. He/She can share the need of a respective plasma donor so that it is easy to find the user. Both users can accept or reject the request. The donor must be healthy and he/she should haven'tdonated in the last 28 days. Once they have donated their details will be removedfor the next 28 days13 times in a year. They can donate The donor details will be stored in cloud applications. At the time of emergency, we can check for the blood donor using GPS. The number of plasma donors available in India is less than compared in other countries. Once the person needed for the blood they enterthe blood group in an app, The alert message will be sent to the nearby donor. It will search for all the donors, if the donor accepts the request the OTP will be sent to verify the donor. This app creates a communication channel through authenticated users whenever a patient needs a blood donation.

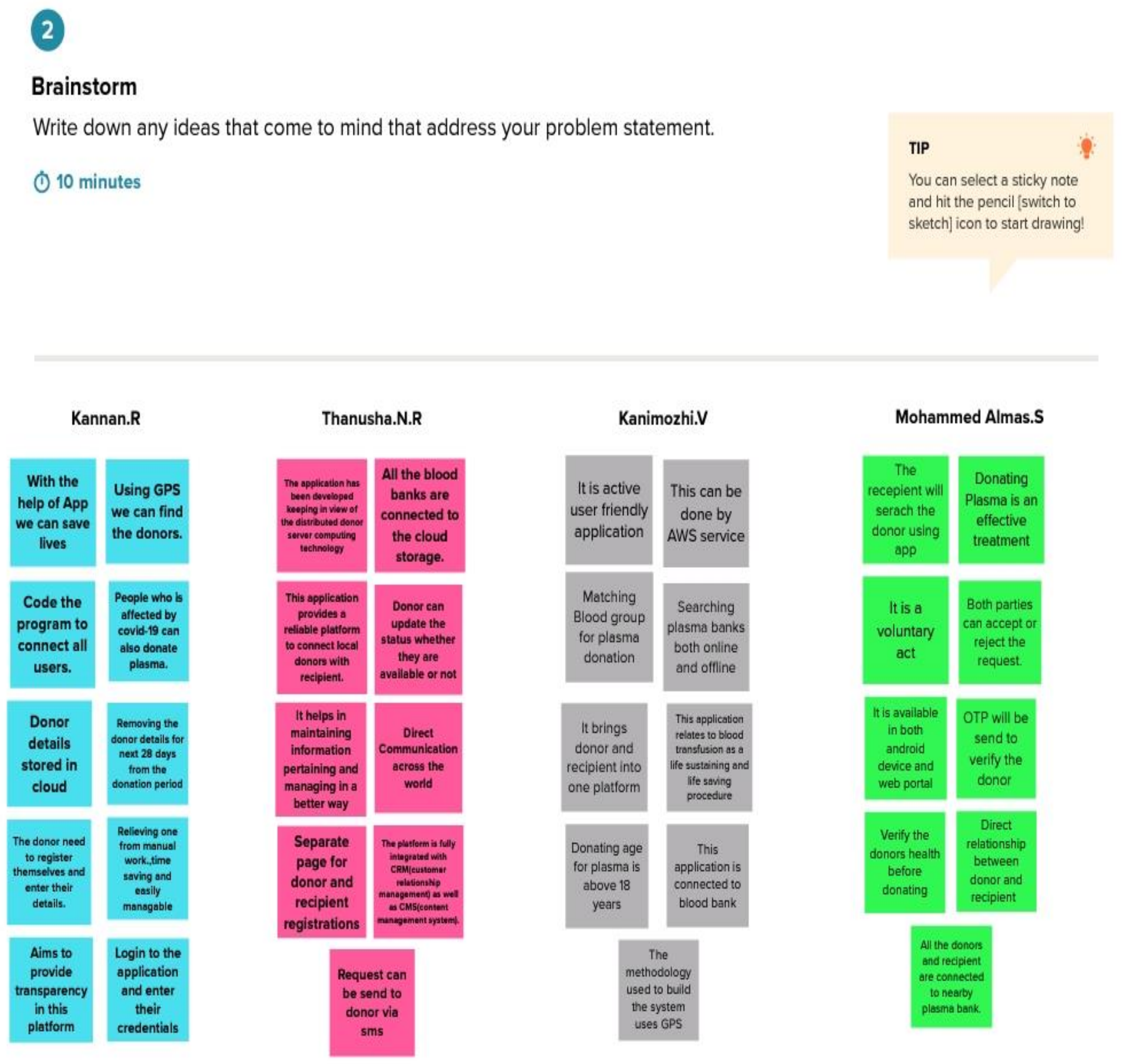
IDEATION AND PROPOSED SOLUTION

3.IDEATION AND PROPOSED SOLUTON

3.1 EMPATHY MAP CANVAS


What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
will anyone come up for help?
How to reduce the demand supply gap?
How can i secure myself in advance
Is It Safe?
Am i eligible for Plasma donation?
What do they HEAR?
what friends say
what boss say
what influencers say
"Difficult to find a plasma donar"
"By the time donars get to know the medical criteria it's too late
"Is it similar to blood donation?"
"...plasma demand and supply gap has grown even bigger.."
"Don't know where to start?"
"Bringing Recovered patients back to hospital for Donation is next to impossible"
Need to find a donor before hand just to be secured!"
What do they SEE?
environment
friends
what the market offers
Confused
Helpless
Anxious
Scared
Exhausted
Frustrated
Shattered
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Contacting friends and family
Convincing recovered patients for donations
Contacting Hospitals
Searching plasma banks online & offline
Posting plea on social media platform
Paying large sum of money to unauthorised plasma agents
PAIN
fears
frustrations
obstacles
Uncomfortable paying cost
Looking for exact blood group matches for matches
Hard to find the donors
GAIN
"wants" / needs
measures of success
obstacles
Accuracy
comfortable
Feeling of accomplishment

3.2 IDEATION & BRAINSTORMING



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Kannan.R

With the help of App we can save lives

Using GPS we can find the donors.

Code the program to connect all users.

People who is affected by covid-19 can also donate plasma.

Donor details stored in cloud

Removing the donor details for next 28 days from the donation period

The donor need to register themselves and enter their details.

Relieving one from manual work.,time saving and easily managable

Aims to provide transparency in this platform

Login to the application and enter their credentials

Thanusha.N.R

The application has been developed keeping in view of the distributed donor server computing technology

All the blood banks are connected to the cloud storage.

This application provides a reliable platform to connect local donors with recipient.

Donor can update the status whether they are available or not

It helps in maintaining information pertaining and managing in a better way

Direct Communication across the world

Separate page for donor and recipient registrations

The platform is fully integrated with CRM(customer relationship management) as well as CMS(content management system).

Request can be send to donor via sms

Kanimozhi.V

It is active user friendly application

This can be done by AWS service

Matching Blood group for plasma donation

Searching plasma banks both online and offline

It brings donor and recipient into one platform

This application relates to blood transfusion as a life sustaining and life saving procedure

Donating age for plasma is above 18 years

This application is connected to blood bank

The methodology used to build the system uses GPS

Mohammed Almas.S

The recepient will serach the donor using app

Donating Plasma is an effective treatment

It is a voluntary act

Both parties can accept or reject the request.

It is available in both android device and web portal

OTP will be send to verify the donor

Verify the donors health before donating

Direct relationship between donor and recipient

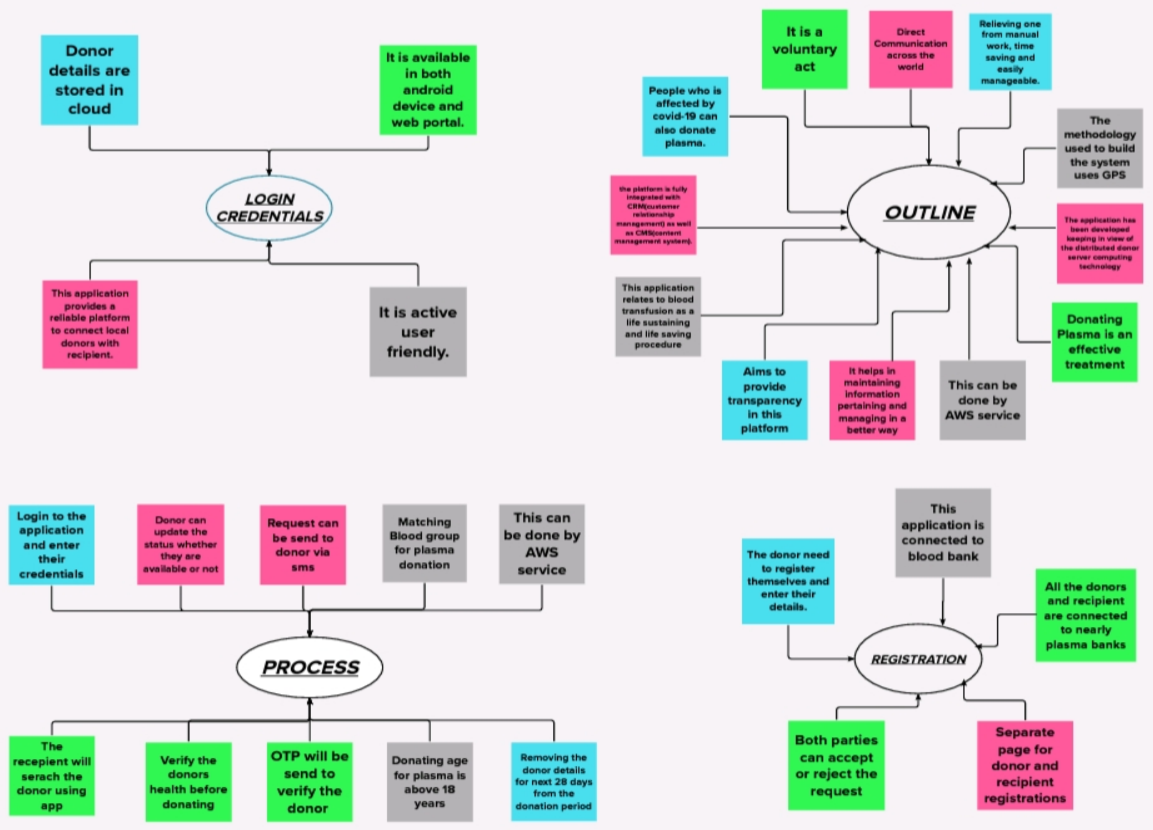
All the donors and recipient are connected to nearby plasma bank.

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



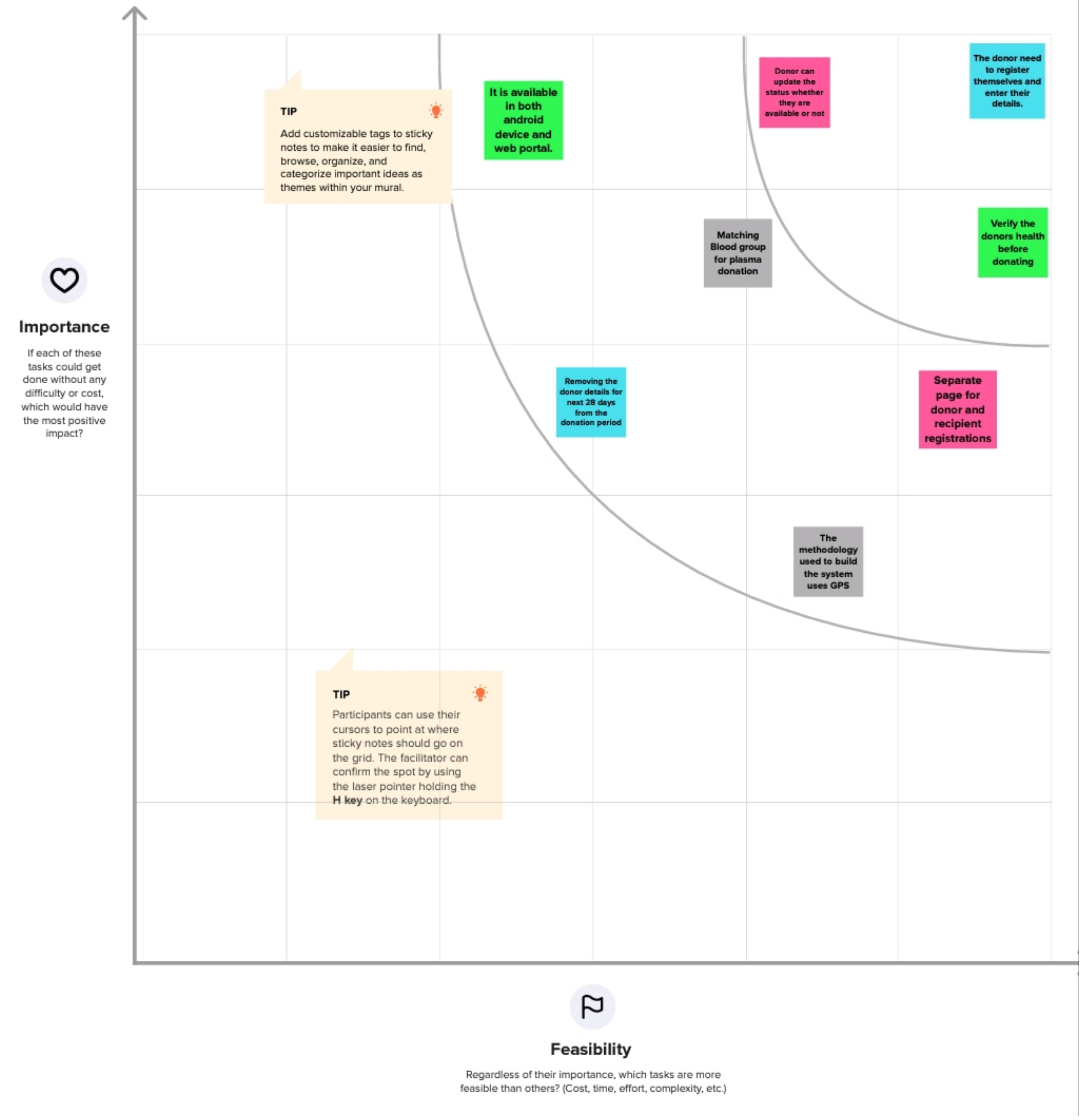
Donor details are stored in cloud
It is available in both android device and web portal.
LOGIN CREDENTIALS
This application provides a reliable platform to connect local donors with recipient.
It is active user friendly.
It is a voluntary act
Direct Communication across the world
Relieving one from manual work, time saving and easily manageable.
People who is affected by covid-19 can also donate plasma.
The methodology used to build the system uses GPS
the platform is fully integrated with CRM(customer relationship management) as well as CMS(content management system).
OUTLINE
The application has been developed keeping in view of the distributed donor server computing technology
This application relates to blood transfusion as a life sustaining and life saving procedure
Donating Plasma is an effective treatment
Aims to provide transparency in this platform
It helps in maintaining information pertaining and managing in a better way
This can be done by AWS service
Login to the application and enter their credentials
Donor can update the status whether they are available or not
Request can be send to donor via sms
Matching Blood group for plasma donation
This can be done by AWS service
PROCESS
The recepient will serach the donor using app
Verify the donors health before donating
OTP will be send to verify the donor
Donating age for plasma is above 18 years
Removing the donor details for next 28 days from the donation period
This application is connected to blood bank
The donor need to register themselves and enter their details.
All the donors and recipient are connected to nearly plasma banks
REGISTRATION
Both parties can accept or reject the request
Separate page for donor and recipient registrations

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.

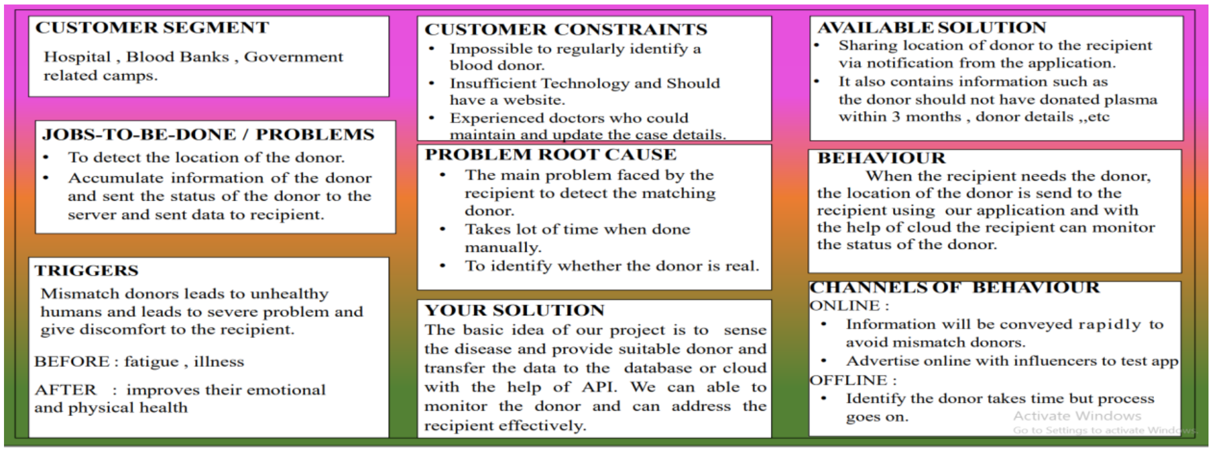
20 minutes


TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.
It is available in both android device and web portal.
Donor can update the status whether they are available or not
The donor need to register themselves and enter their details.
Matching Blood group for plasma donation
Verify the donors health before donating
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Removing the donor details for next 28 days from the donation period
Separate page for donor and recipient registrations
The methodology used to build the system uses GPS
TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Build cloud-based Plasma donor applications to save a life.
2.	Idea / Solution description	Our solution is to build a Cloud Based Application that can track the donors when the recipient needs plasma. It also has the list of donors who are ready to donate and removes the donor once donated. It also notifies the right donor when there is a need. It is an effective application during an emergency.
3.	Novelty / Uniqueness	It can track the donors in real time and monitor them with accurate details like name, age, and whether they have donated within the stipulated time, etc.. and can notify via email or message when the recipient finds the donor.
4.	Social Impact / Customer Satisfaction	Recipients need not worry about the right donor. All they must do is feed the data into the application to search for the right donor.
5.	Business Model (Revenue Model)	Based on the status of the donors, the recipient needs to offer to the user. The more critical the case is, the bigger the revenue.
6.	Scalability of the Solution	It can track and maintain any number of donors and recipients without any errors and give them accurate results.

3.4 PROBLEM SOLUTION FIT



CUSTOMER SEGMENT

Hospital , Blood Banks , Government related camps.

JOBS-TO-BE-DONE / PROBLEMS

- To detect the location of the donor.
- Accumulate information of the donor and sent the status of the donor to the server and sent data to recipient.

TRIGGERS

Mismatch donors leads to unhealthy humans and leads to severe problem and give discomfort to the recipient.

BEFORE : fatigue , illness

AFTER : improves their emotional and physical health

CUSTOMER CONSTRAINTS

- Impossible to regularly identify a blood donor.
- Insufficient Technology and Should have a website.
- Experienced doctors who could maintain and update the case details.

PROBLEM ROOT CAUSE

- The main problem faced by the recipient to detect the matching donor.
- Takes lot of time when done manually.
- To identify whether the donor is real.

YOUR SOLUTION

The basic idea of our project is to sense the disease and provide suitable donor and transfer the data to the database or cloud with the help of API. We can able to monitor the donor and can address the recipient effectively.

AVAILABLE SOLUTION

- Sharing location of donor to the recipient via notification from the application.
- It also contains information such as the donor should not have donated plasma within 3 months , donor details ,,etc

BEHAVIOUR

When the recipient needs the donor, the location of the donor is send to the recipient using our application and with the help of cloud the recipient can monitor the status of the donor.

CHANNELS OF BEHAVIOUR

ONLINE :

- Information will be conveyed rapidly to avoid mismatch donors.
- Advertise online with influencers to test app

OFFLINE :

- Identify the donor takes time but process goes on.

REQUIREMENT ANALYSIS

4 REQUIREMENT ANALYSIS

4.1FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form, Gmail, Blood Bank App.
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP, SMS
FR-3	Access	The form can be access through website, google forms.
FR-4	Gmail Account	Plasma donors can send request to the recipient through mail also. As soon as the request is submitted, they will get an confirmation e-mail to the registered one and then click confirm.
FR-5	Login requirements	Donor/Recipient can log into the application by entering email & password.
FR-6	Internet Facility	We can give information through website, blood bank app, social media, etc.

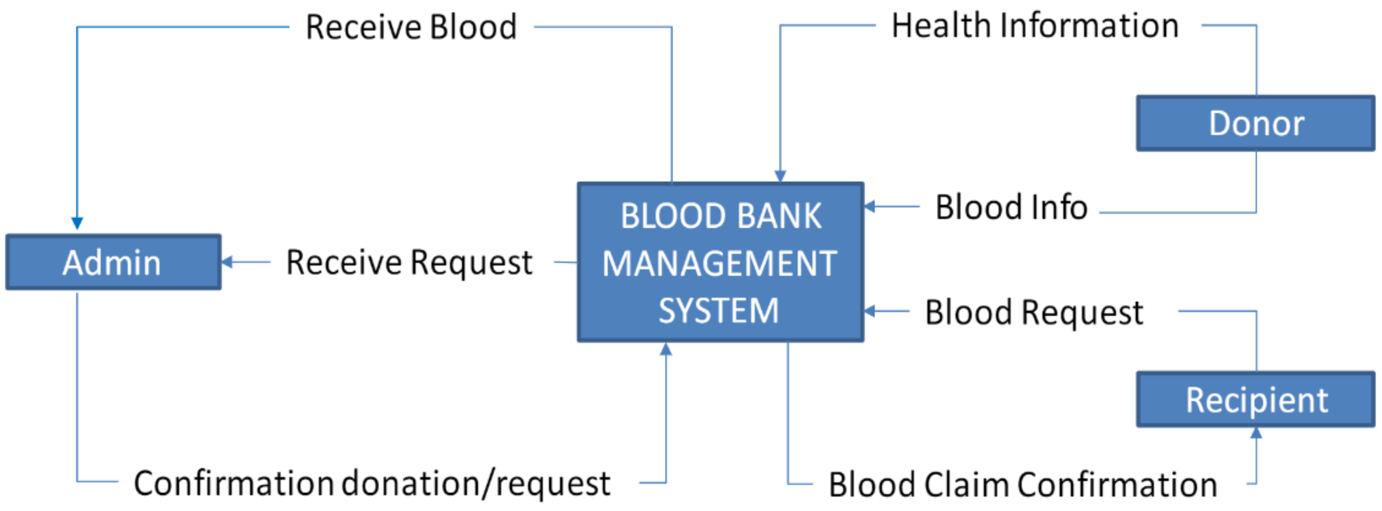
4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It is very useful when the patient is in need of blood.
NFR-2	**Security**	It should be secured with the proper matching blood group.
NFR-3	**Reliability**	It can recover the corona patients by blood transfusion of already corona affected and recovered people.
NFR-4	**Availability**	It requires an active internet connection for checking the status of blood donors.
NFR-5	**Scalability**	To ensure the maintenance of donor's healthy and allow them to give blood when needed, and ensuring that whether they have given the blood before.

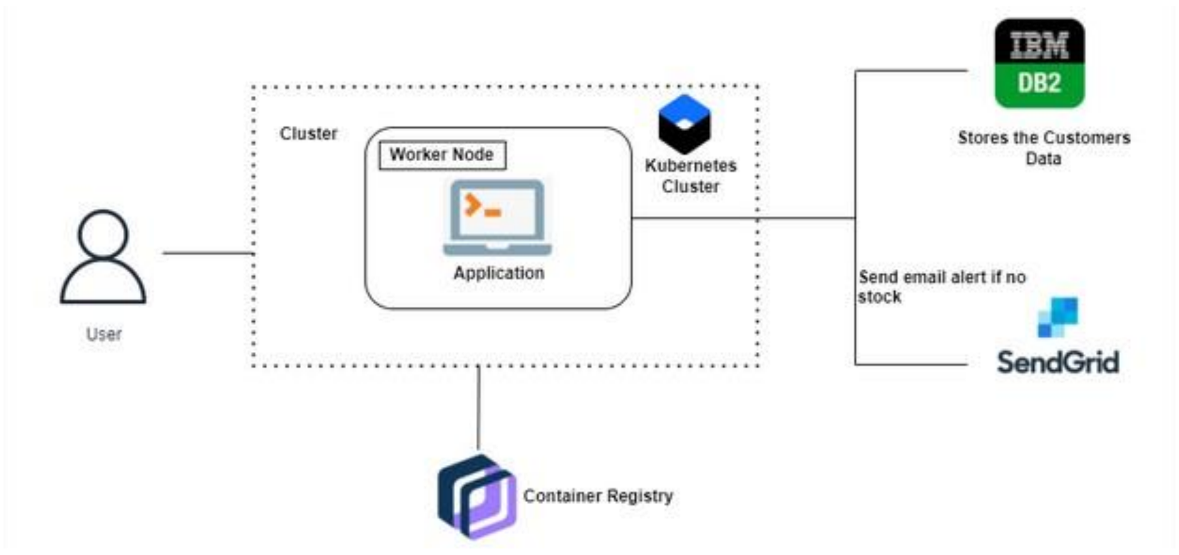
PROJECT DESIGN

5. PROJECT DESIGN

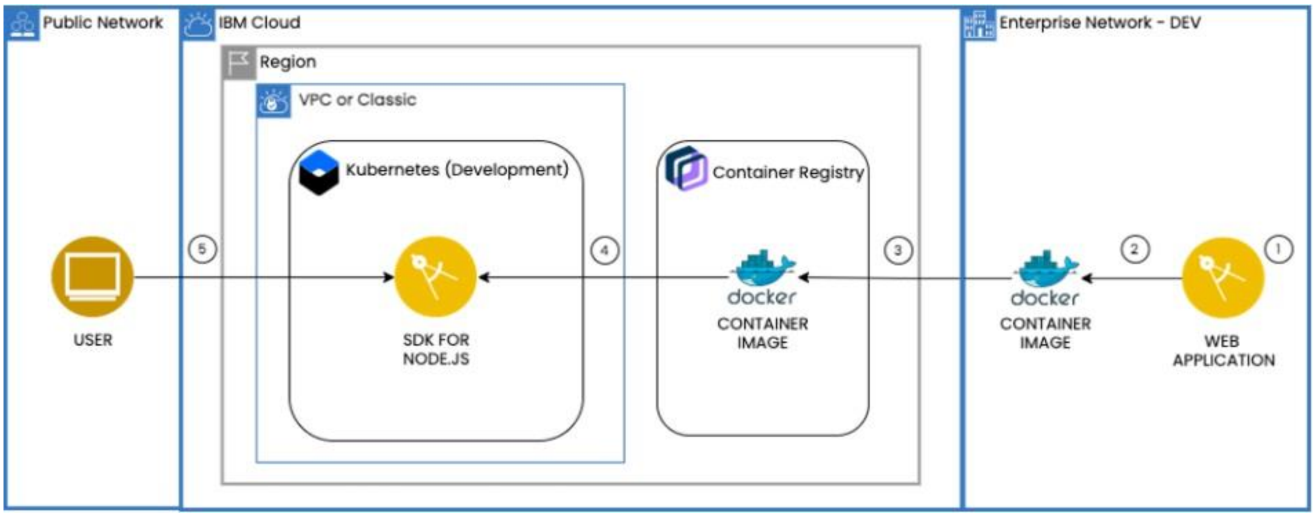
5.1 DATA FLOW DIAGRAMS



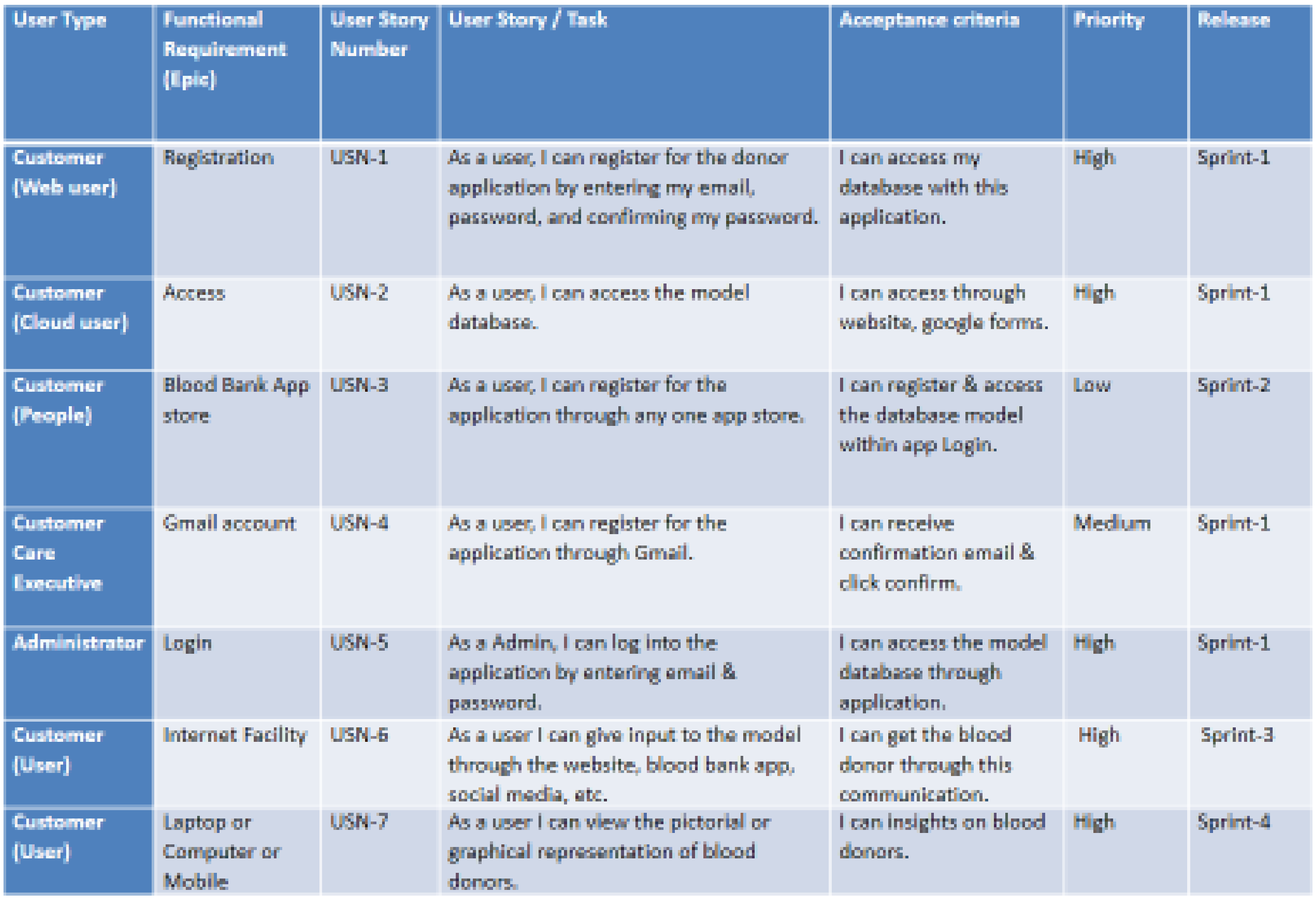
5.2 SOLUTION & TECHNICAL ARCHITECTURE



Cloud Services:



5.3 USER STORIES



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	USN-1	As a user, I can register for the donor application by entering my email, password, and confirming my password.	I can access my database with this application.	High	Sprint-1
Customer (Cloud user)	Access	USN-2	As a user, I can access the model database.	I can access through website, google forms.	High	Sprint-1
Customer (People)	Blood Bank App store	USN-3	As a user, I can register for the application through any one app store.	I can register & access the database model within app Login.	Low	Sprint-2
Customer Care Executive	Gmail account	USN-4	As a user, I can register for the application through Gmail.	I can receive confirmation email & click confirm.	Medium	Sprint-1
Administrator	Login	USN-5	As a Admin, I can log into the application by entering email & password.	I can access the model database through application.	High	Sprint-1
Customer (User)	Internet Facility	USN-6	As a user I can give input to the model through the website, blood bank app, social media, etc.	I can get the blood donor through this communication.	High	Sprint-3
Customer (User)	Laptop or Computer or Mobile	USN-7	As a user I can view the pictorial or graphical representation of blood donors.	I can insights on blood donors.	High	Sprint-4

PROJECT PLANNING AND SCHEDULING

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	USN-1	As a user, I can register for the applicationby entering my email, and password, and confirming my password.	6	High	Kannan R Thanusha N R
Sprint-1	Login	USN-2	As a user, I can log into the application by entering my email & password.	6	High	Kannan R Kanimozhi V
Sprint-1	Admin Register	USN-3	An admin can register through the admin registry.	4	Medium	Thanusha N R Mohammed Almas.S
Sprint-1	Admin Register via Script	USN-4	Creating an Admin Account using a python script. For security reasons we should implement a separate python script.	4	High	Kannan R Thanusha N R
Sprint-2	Implementing AuthenticationSystem	USN-5	Creating an authentication and secured system for both admin andusers using flask application	6	High	Kannan R Mohammed Almas.S
Sprint-2	Creating tables	USN-6	Creating Db2 account and creating tables in Db2 in IBM cloud db2	4	Medium	Thanusha N RKanimozhi V
Sprint-2	Creating SSL certificate and integrating with html code	USN-7	Creating the SSL certificate to connect db2via html code.	6	High	Kannan R Mohammed Almas.S
Sprint-2	Creating dashboard	USN-8	Admin and Donor can interact using thisapplication.	4	Medium	Thanusha N RKanimozhi V

Sprint	Functional Requirement (Epic)	User Story Number	User StoryNumber	Story Points	Priority	Team Members
Sprint-3	Plasma request and donor acknowledge feature	USN-9	Admin can request for plasma which willbe shown in the user portal.	6	High	Kannan R Kanimozhi V
Sprint-3	Creating dashboard for admin	USN-10	In Admin dashboard, admin can view allthe plasma request that has been requested by the recipient/user.	6	High	Kannan R Mohammed Almas.S
Sprint-3	Integrating the Watson chatbot	USN-11	Users can use the chatbot for basic clarification.	4	Medium	Kanimozhi V Mohammed Almas.S
Sprint-3	Integration with SendGrid .	USN-12	The source/verification mail for both user(donar and recipient).	4	Medium	Thanusha N R Kanimozhi V
Sprint-4	Docker installation	USN-13	Installing Docker CLI.	4	Low	Thanusha N R Kanimozhi V
Sprint-4	Creating docker image	USN-14	Setting up the docker environment and creating the docker image file.	6	High	Kannan R Mohammed Almas.S
Sprint-4	Kubernetes	USN-15	creating pods in Kubernetes and upload itin IBM cloud.	6	Medium	Thanusha N R Kanimozhi V
Sprint-4	End-to-End Testing	USN-16	Implementing End-to-End testing.	6	High	Kannan R Mohammed Almas.S

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date(Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	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 Nov 2022
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’s calculate the team’s average velocity (AV) per iteration unit (story points

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Velocity:

AV = Sprint Duration / Velocity = 20/6 = 3.3333…

Sprint 1(AV)= 3.34

Sprint 2(AV)=3.34

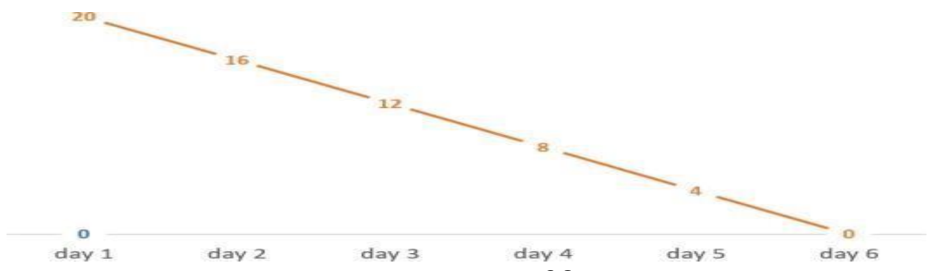
Sprint 3(AV)=3.34

Sprint 4(AV)=3.34

per day).

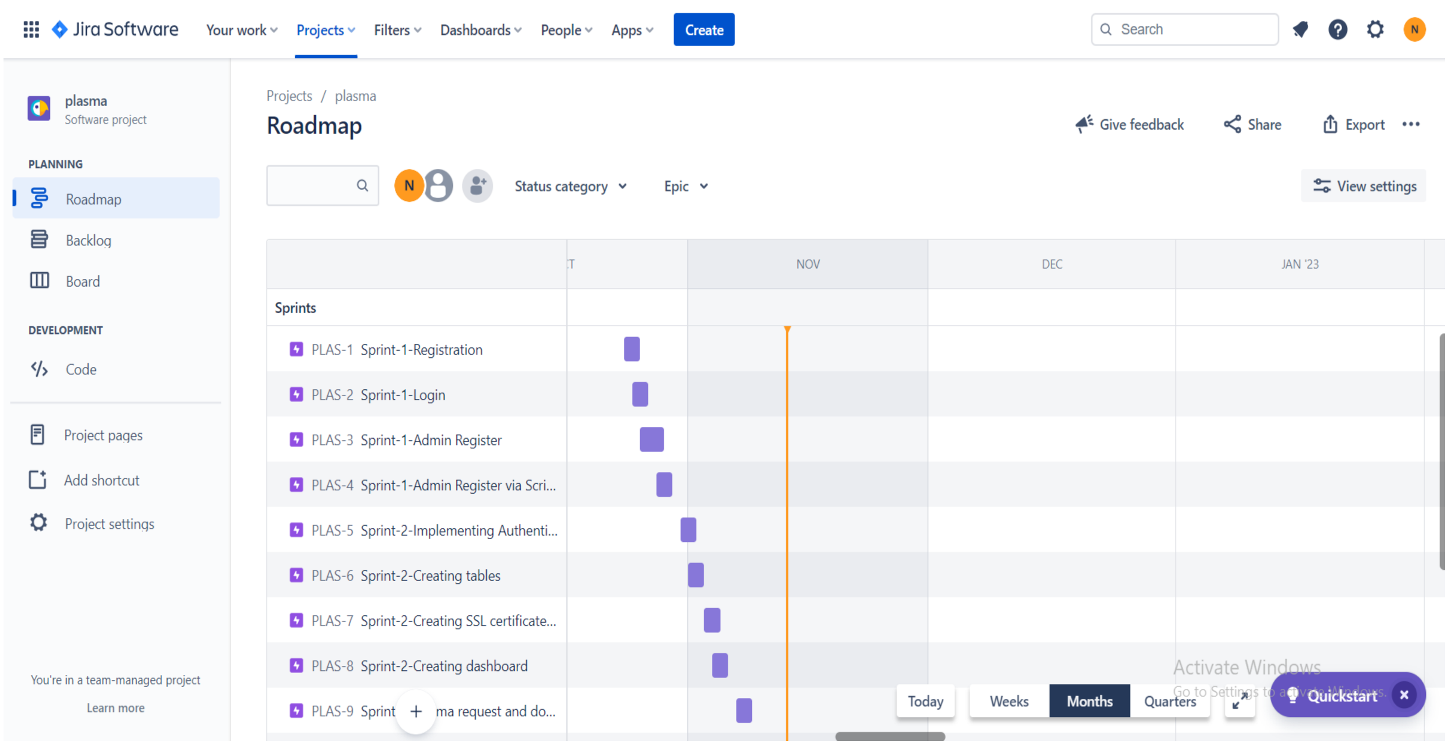
Burndown Chart:

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



REPORTS FROM JIRA

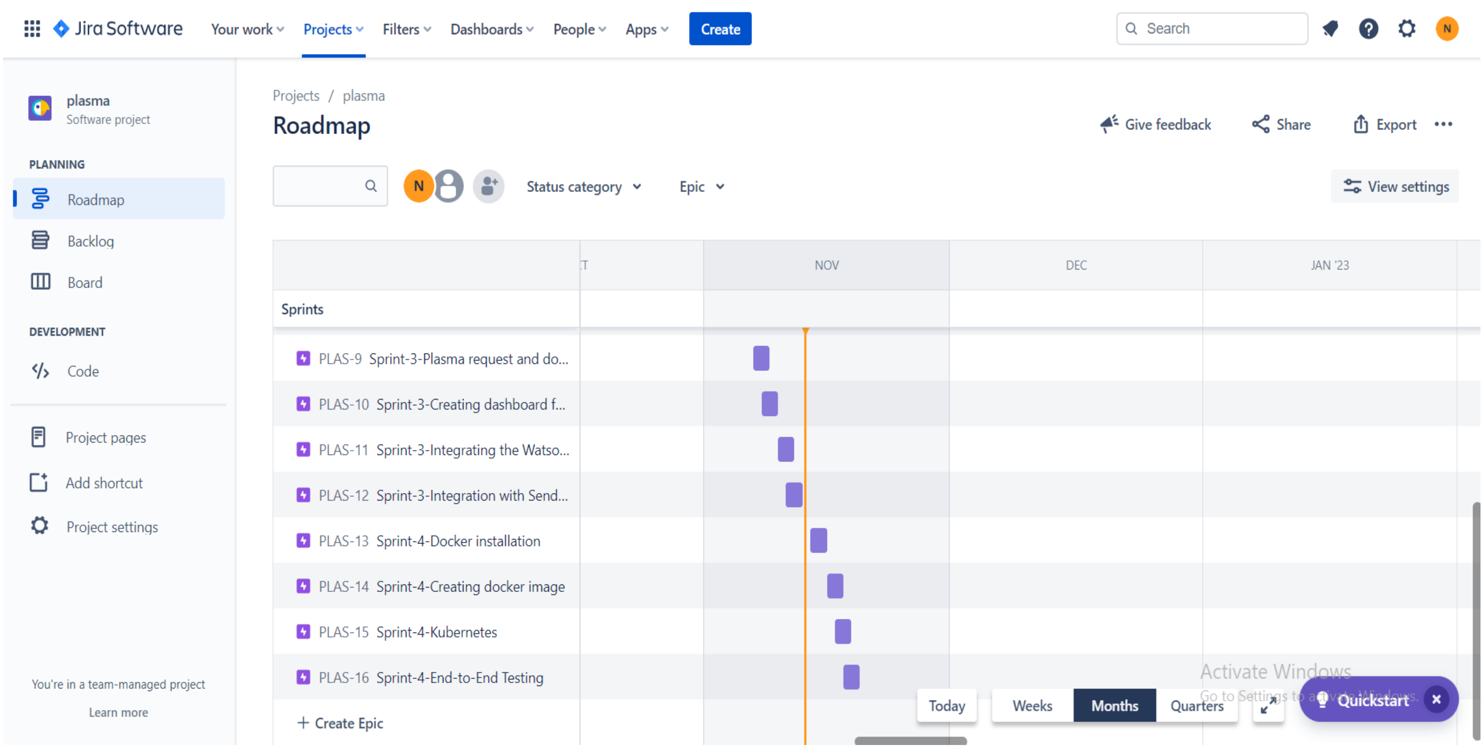
ROADMAP



Roadmap

Sprints

- PLAS-1 Sprint-1-Registration
- PLAS-2 Sprint-1-Login
- PLAS-3 Sprint-1-Admin Register
- PLAS-4 Sprint-1-Admin Register via Scri...
- PLAS-5 Sprint-2-Implementing Authenti...
- PLAS-6 Sprint-2-Creating tables
- PLAS-7 Sprint-2-Creating SSL certificate...
- PLAS-8 Sprint-2-Creating dashboard
- PLAS-9 Sprint ma request and do...



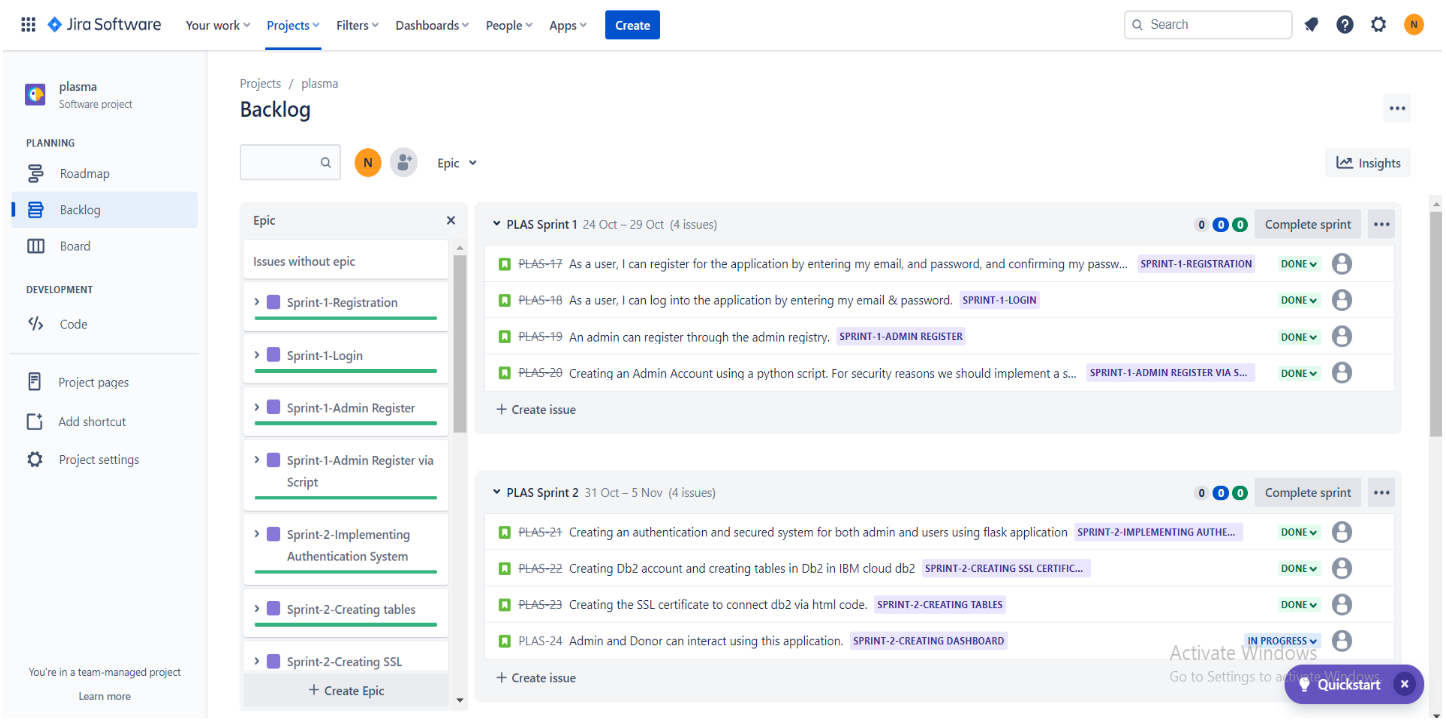
Roadmap

Sprints

- PLAS-9 Sprint-3-Plasma request and do...
- PLAS-10 Sprint-3-Creating dashboard f...
- PLAS-11 Sprint-3-Integrating the Watso...
- PLAS-12 Sprint-3-Integration with Send...
- PLAS-13 Sprint-4-Docker installation
- PLAS-14 Sprint-4-Creating docker image
- PLAS-15 Sprint-4-Kubernetes
- PLAS-16 Sprint-4-End-to-End Testing

+ Create Epic

BACKLOG



Jira Software | Your work | Projects | Filters | Dashboards | People | Apps | Create | Search

plasma
Software project

PLANNING
Roadmap
Backlog
Board

DEVELOPMENT
Code

Project pages
Add shortcut
Project settings

You're in a team-managed project
Learn more

Projects / plasma

Backlog

N | Epic | Insights

Epic
Issues without epic
Sprint-1-Registration
Sprint-1-Login
Sprint-1-Admin Register
Sprint-1-Admin Register via Script
Sprint-2-Implementing Authentication System
Sprint-2-Creating tables
Sprint-2-Creating SSL
+ Create Epic

PLAS Sprint 1 24 Oct – 29 Oct (4 issues) 0 0 0 Complete sprint

~~PLAS-17~~	As a user, I can register for the application by entering my email, and password, and confirming my passw...	SPRINT-1-REGISTRATION	DONE
~~PLAS-18~~	As a user, I can log into the application by entering my email & password.	SPRINT-1-LOGIN	DONE
~~PLAS-19~~	An admin can register through the admin registry.	SPRINT-1-ADMIN REGISTER	DONE
~~PLAS-20~~	Creating an Admin Account using a python script. For security reasons we should implement a s...	SPRINT-1-ADMIN REGISTER VIA S...	DONE

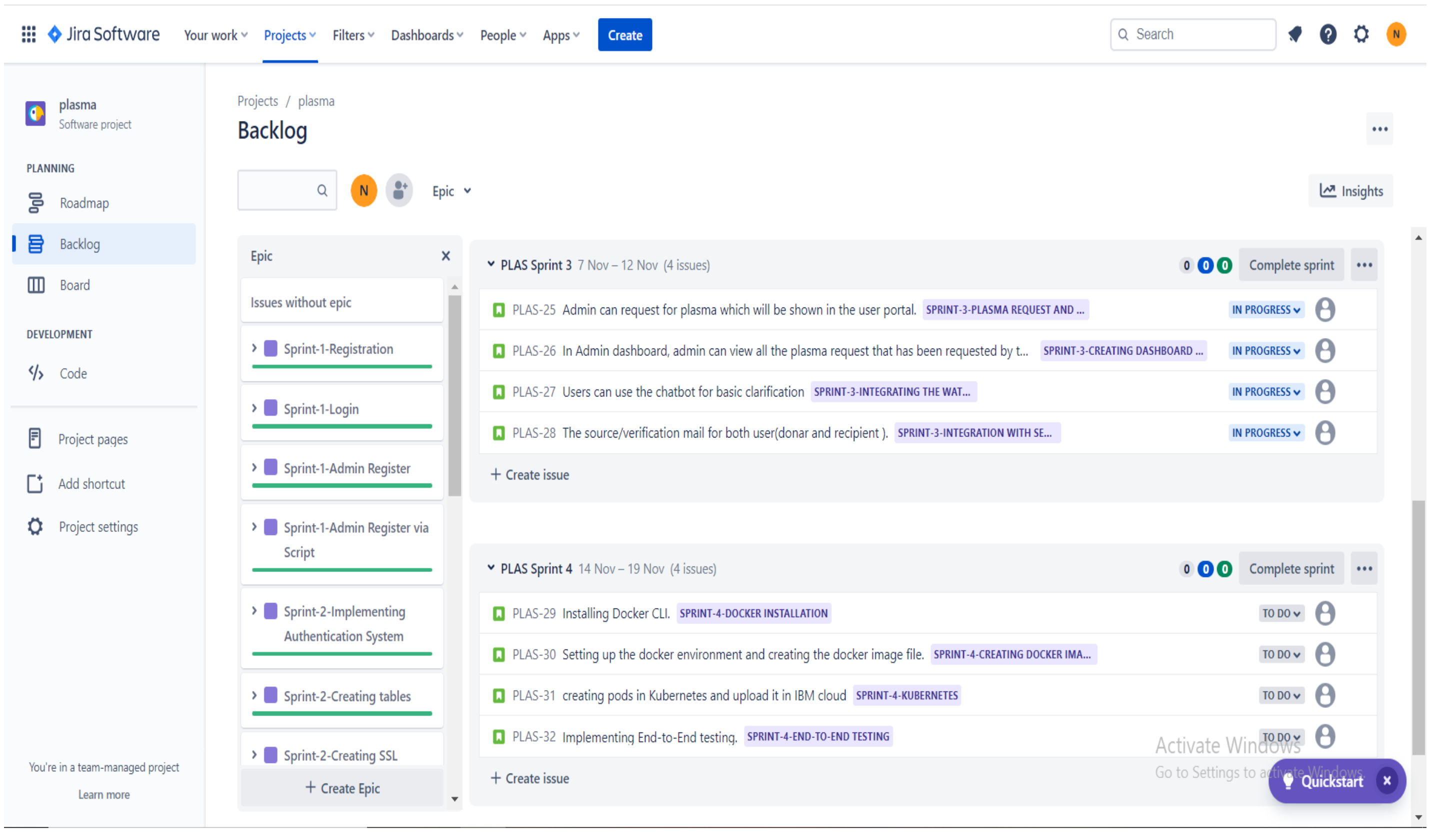
+ Create issue

PLAS Sprint 2 31 Oct – 5 Nov (4 issues) 0 0 0 Complete sprint

~~PLAS-21~~	Creating an authentication and secured system for both admin and users using flask application	SPRINT-2-IMPLEMENTING AUTHE...	DONE
~~PLAS-22~~	Creating Db2 account and creating tables in Db2 in IBM cloud db2	SPRINT-2-CREATING SSL CERTIFIC...	DONE
~~PLAS-23~~	Creating the SSL certificate to connect db2 via html code.	SPRINT-2-CREATING TABLES	DONE
PLAS-24	Admin and Donor can interact using this application.	SPRINT-2-CREATING DASHBOARD	IN PROGRESS

+ Create issue

Activate Windows
Go to Settings to activate Windows.
Quickstart



Jira Software | Your work | Projects | Filters | Dashboards | People | Apps | Create | Search

plasma
Software project

PLANNING
Roadmap
Backlog
Board

DEVELOPMENT
Code

Project pages
Add shortcut
Project settings

You're in a team-managed project
Learn more

Projects / plasma

Backlog

N | Epic | Insights

Epic
Issues without epic
Sprint-1-Registration
Sprint-1-Login
Sprint-1-Admin Register
Sprint-1-Admin Register via Script
Sprint-2-Implementing Authentication System
Sprint-2-Creating tables
Sprint-2-Creating SSL
+ Create Epic

PLAS Sprint 3 7 Nov – 12 Nov (4 issues) 0 0 0 Complete sprint

PLAS-25	Admin can request for plasma which will be shown in the user portal.	SPRINT-3-PLASMA REQUEST AND ...	IN PROGRESS
PLAS-26	In Admin dashboard, admin can view all the plasma request that has been requested by t...	SPRINT-3-CREATING DASHBOARD ...	IN PROGRESS
PLAS-27	Users can use the chatbot for basic clarification	SPRINT-3-INTEGRATING THE WAT...	IN PROGRESS
PLAS-28	The source/verification mail for both user(donar and recipient).	SPRINT-3-INTEGRATION WITH SE...	IN PROGRESS

+ Create issue

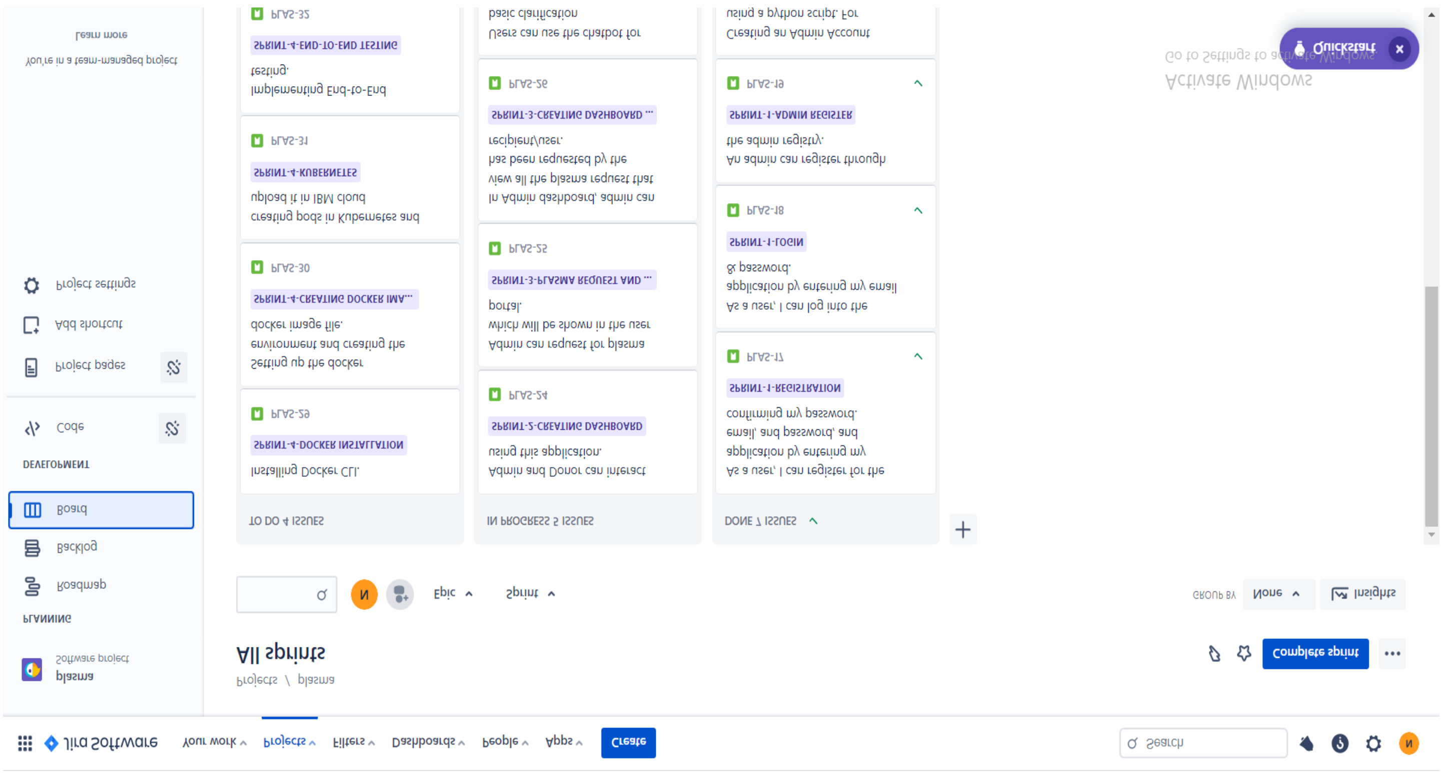
PLAS Sprint 4 14 Nov – 19 Nov (4 issues) 0 0 0 Complete sprint

PLAS-29	Installing Docker CLI.	SPRINT-4-DOCKER INSTALLATION	TO DO
PLAS-30	Setting up the docker environment and creating the docker image file.	SPRINT-4-CREATING DOCKER IMA...	TO DO
PLAS-31	creating pods in Kubernetes and upload it in IBM cloud	SPRINT-4-KUBERNETES	TO DO
PLAS-32	Implementing End-to-End testing.	SPRINT-4-END-TO-END TESTING	TO DO

+ Create issue

Activate Windows
Go to Settings to activate Windows.
Quickstart

BOARD



Jira Software Your work Projects Filters Dashboards People Apps Create Search

plasma
Software project

PLANNING
Roadmap
Backlog
Board

DEVELOPMENT
Code
Project pages
Add shortcut
Project settings

You're in a team-managed project
Learn more

Projects / plasma

All sprints

Complete sprint

Epic Sprint

GROUP BY None Insights

TO DO 4 ISSUES

Installing Docker CLI.
SPRINT-4-DOCKER INSTALLATION
PLAS-29

Setting up the docker environment and creating the docker image file.
SPRINT-4-CREATING DOCKER IMA...
PLAS-30

creating pods in Kubernetes and upload it in IBM cloud
SPRINT-4-KUBERNETES
PLAS-31

Implementing End-to-End testing.
SPRINT-4-END-TO-END TESTING
PLAS-32

IN PROGRESS 5 ISSUES

Admin and Donor can interact using this application.
SPRINT-2-CREATING DASHBOARD
PLAS-24

Admin can request for plasma which will be shown in the user portal.
SPRINT-3-PLASMA REQUEST AND ...
PLAS-25

In Admin dashboard, admin can view all the plasma request that has been requested by the recipient/user.
SPRINT-3-CREATING DASHBOARD ...
PLAS-26

Users can use the chatbot for basic clarification

DONE 7 ISSUES

As a user, I can register for the application by entering my email, password, and confirming my password.
SPRINT-1-REGISTRATION
PLAS-17

As a user, I can log into the application by entering my email & password.
SPRINT-1-LOGIN
PLAS-18

An admin can register through the admin registry.
SPRINT-1-ADMIN REGISTER
PLAS-19

Creating an Admin Account using a python script. For

Quickstart

Activate Windows
Go to Settings to activate Windows

:

CODING AND SOLUTIONING

7. CODING & SOLUTIONING

7.1 FEATURE 1

https://github.com/IBM-EPBL/IBM-Project-7825-1658900440/tree/main/Project%20Development%20Phase/Sprint%201

- It consists of three modules login page, navigation page, and register page.
- It is the main webpage of our model Story-
- It shows the need for plasma donation.

https://github.com/IBM-EPBL/IBM-Project-7825-1658900440/tree/main/Project%20Development%20Phase/Sprint%202

- Here we discussed about donor register module and needer register module
- In this module, users can register themselves as a donor. If a certain age limit is satisfied their registration process for plasma donors will be accepted.
- In this needer module , at the time of emergency the recipient can enter their details to search donor in need.

7.2 FEATURE 2

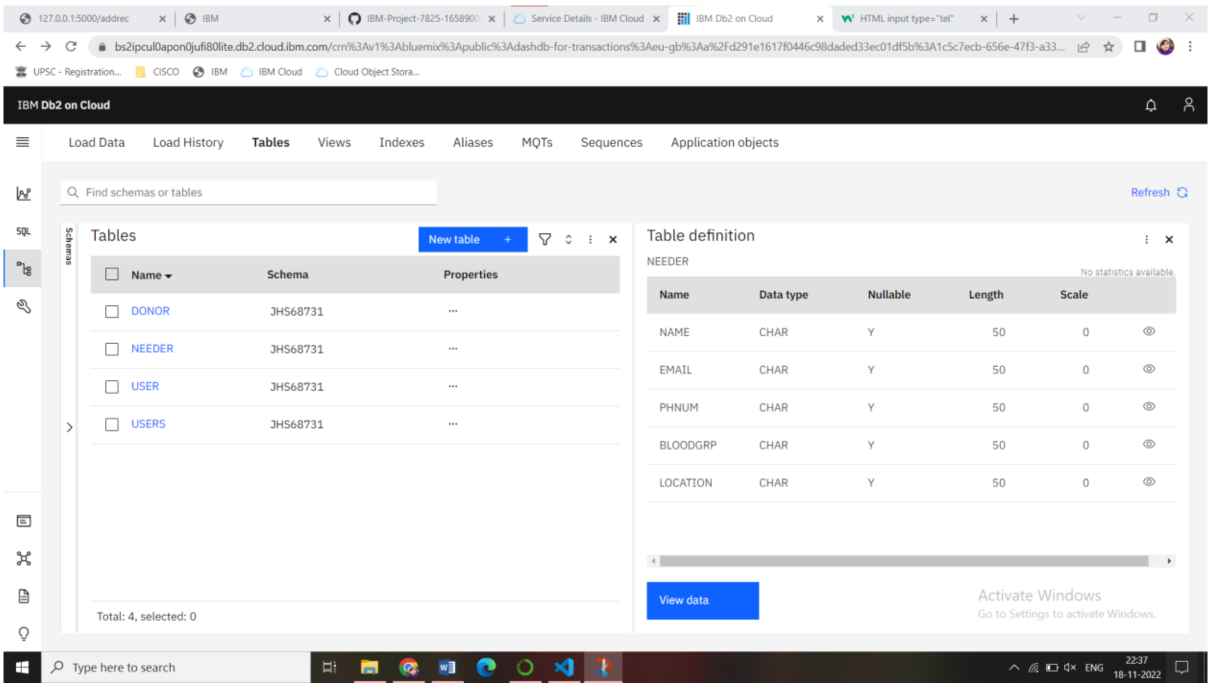
https://github.com/IBM-EPBL/IBM-Project-7825-1658900440/tree/main/Project%20Development%20Phase/Sprint%203

- Here we discussed index module
- In this module, the home page of our application to make users aware of this application.

https://github.com/IBM-EPBL/IBM-Project-7825-1658900440/tree/main/Project%20Development%20Phase/Sprint%204

- In this process we make a database connectivity for register , login and update of donor information.

7.3 DATABASE SCHEMA



Tables

Name	Schema	Properties
DONOR	JHS68731	...
NEEDER	JHS68731	...
USER	JHS68731	...
USERS	JHS68731	...

Total: 4, selected: 0

Table definition

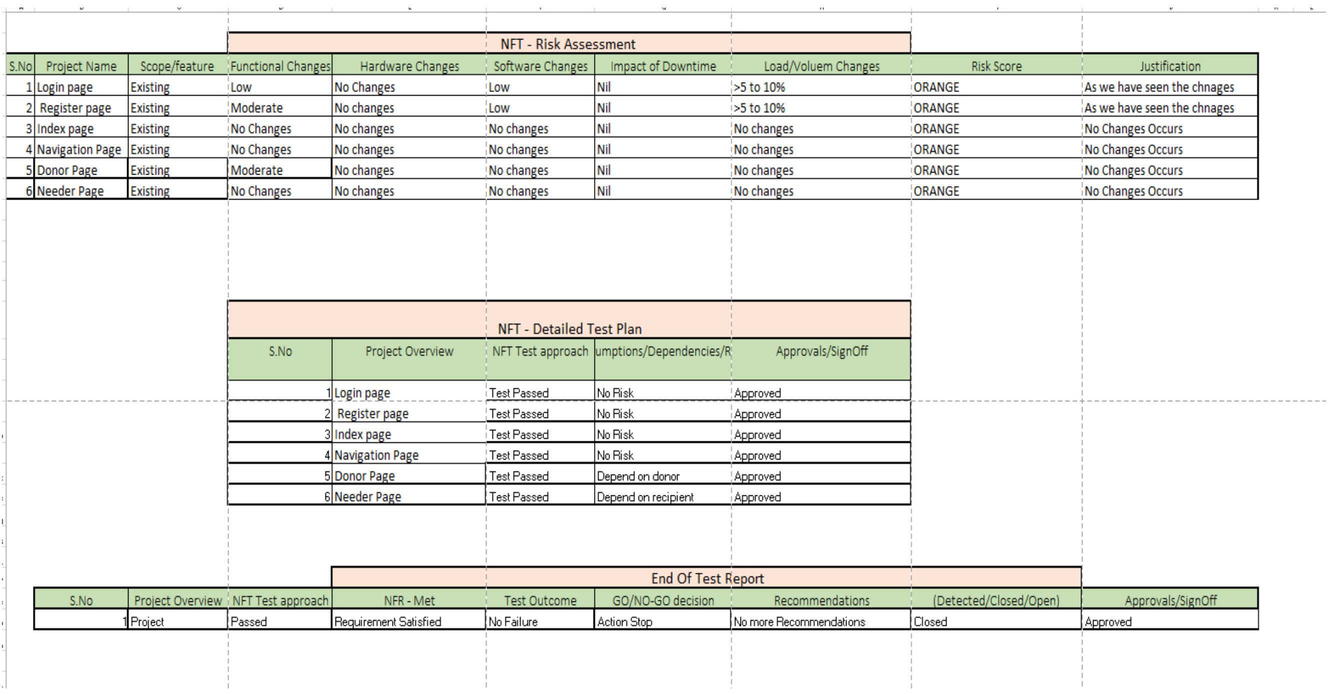
NEEDER

Name	Data type	Nullable	Length	Scale
NAME	CHAR	Y	50	0
EMAIL	CHAR	Y	50	0
PHNUM	CHAR	Y	50	0
BLOODGRP	CHAR	Y	50	0
LOCATION	CHAR	Y	50	0

TESTING

TESTING

8.1 PERFORMACE TESTING



NFT - Risk Assessment

S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Voluem Changes	Risk Score	Justification
1	Login page	Existing	Low	No Changes	Low	Nil	>5 to 10%	ORANGE	As we have seen the chnages
2	Register page	Existing	Moderate	No changes	Low	Nil	>5 to 10%	ORANGE	As we have seen the chnages
3	Index page	Existing	No Changes	No changes	No changes	Nil	No changes	ORANGE	No Changes Occurs
4	Navigation Page	Existing	No Changes	No changes	No changes	Nil	No changes	ORANGE	No Changes Occurs
5	Donor Page	Existing	Moderate	No changes	No changes	Nil	No changes	ORANGE	No Changes Occurs
6	Needer Page	Existing	No Changes	No changes	No changes	Nil	No changes	ORANGE	No Changes Occurs

NFT - Detailed Test Plan

S.No	Project Overview	NFT Test approach	umptions/Dependencies/R	Approvals/SignOff
1	Login page	Test Passed	No Risk	Approved
2	Register page	Test Passed	No Risk	Approved
3	Index page	Test Passed	No Risk	Approved
4	Navigation Page	Test Passed	No Risk	Approved
5	Donor Page	Test Passed	Depend on donor	Approved
6	Needer Page	Test Passed	Depend on recipient	Approved

End Of Test Report

S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	(Detected/Closed/Open)	Approvals/SignOff
1	Project	Passed	Requirement Satisfied	No Failure	Action Stop	No more Recommendations	Closed	Approved

8.2 USER ACCEPTANCE TESTING

1. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	19
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	5	1	1	10	17
Not Reproduced	0	0	0	0	0
Skipped	1	1	0	1	3
Won't Fix	0	2	2	0	4
Totals	19	11	8	15	62

2. 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	6	0	0	6
Client Application	12	0	0	12
Security	1	0	0	1
Outsource Shipping	0	0	0	0
Exception Reporting	9	0	0	9
Final Report Output	9	0	0	9
Version Control	1	0	0	1

8.3 TEST CASES

Date	03-Nov-22
Team ID	PNT2022TMID08777
Project Name	Plasma Donor Application
Maximum Marks	4 marks

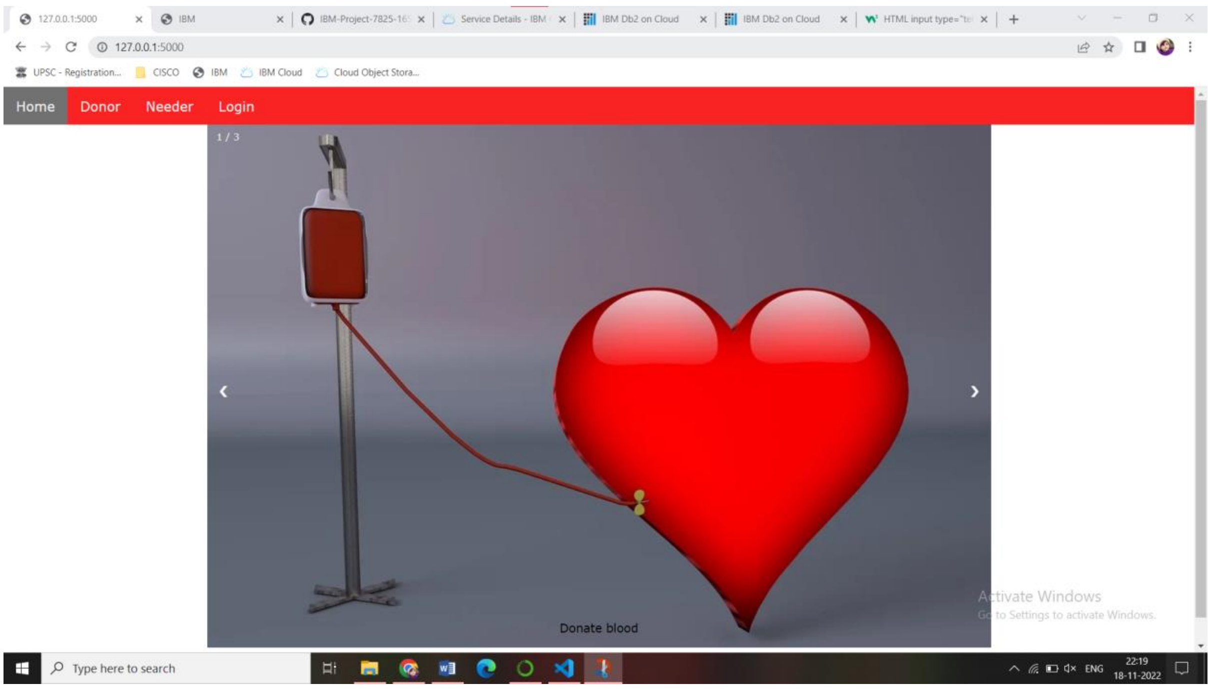
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_O01	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	-	Login/Signup popup should display	Working as expected	Pass				Kannan R
LoginPage_TC_O02	UI	Home Page	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.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 with orange colour d.New customer? Create account link e.Last password? Recovery password link	Working as expected	Pass	Steps are not clear to follow			Thanusha N R
LoginPage_TC_O03	Functional	Login page	Verify user is able to log into application with Valid credentials		1.Enter the url 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 login button	Username: red@gmail.com password: Testing123	User should navigate to user account homepage	Working as expected	Pass				Kanimozhi V
LoginPage_TC_O04	Functional	Donor page	Verify user is able to log into application with InValid credentials		1.Enter URL and click go 2.Click on My Account dropdown button 3.Click on donor button 4.Enter valid password in password text box 5.Fill the details	Username: orange@gmail password: 1234567	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass				Mohammed Almas
LoginPage_TC_O04	Functional	needer page	Verify user is able to log into application with InValid credentials		1.Enter URL and click go 2.Click on My Account dropdown button 3.Click on recipient button 4.Enter valid password in password text box	Username: violet@gmail.com password: registe456	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass				Kannan R

RESULTS

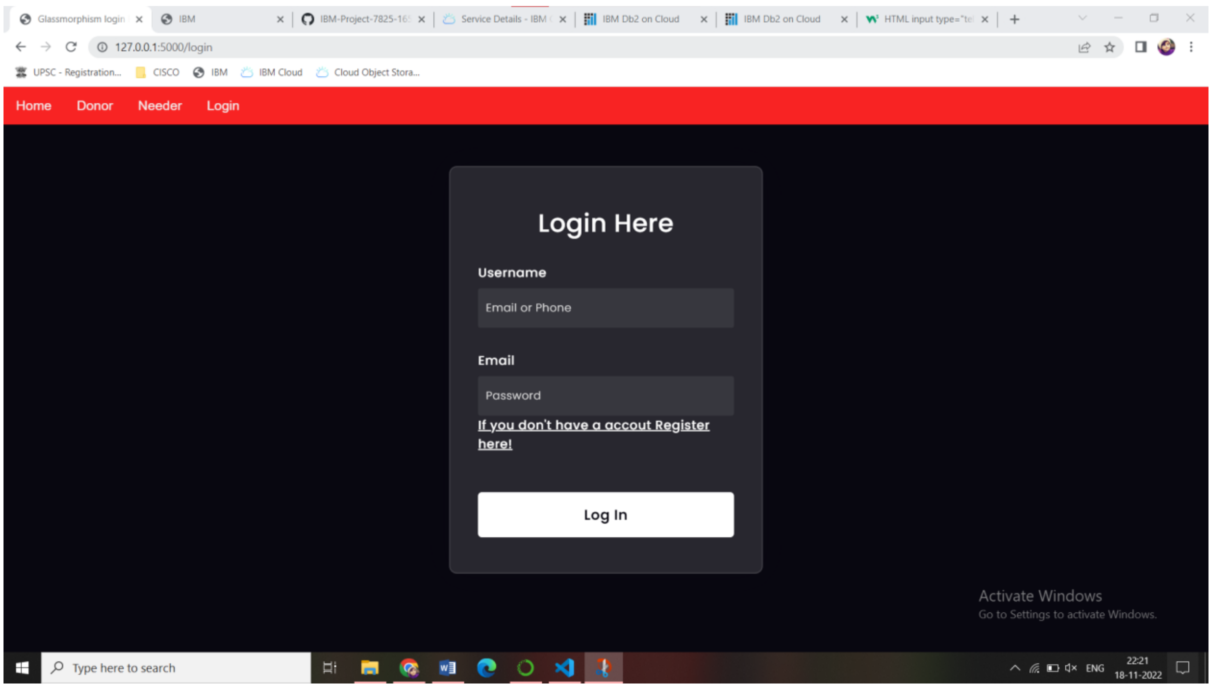
9. RESULTS

9.1 PERFORMANCE MATRIX

Home Page



Login Page



Login Here

Username

Email or Phone

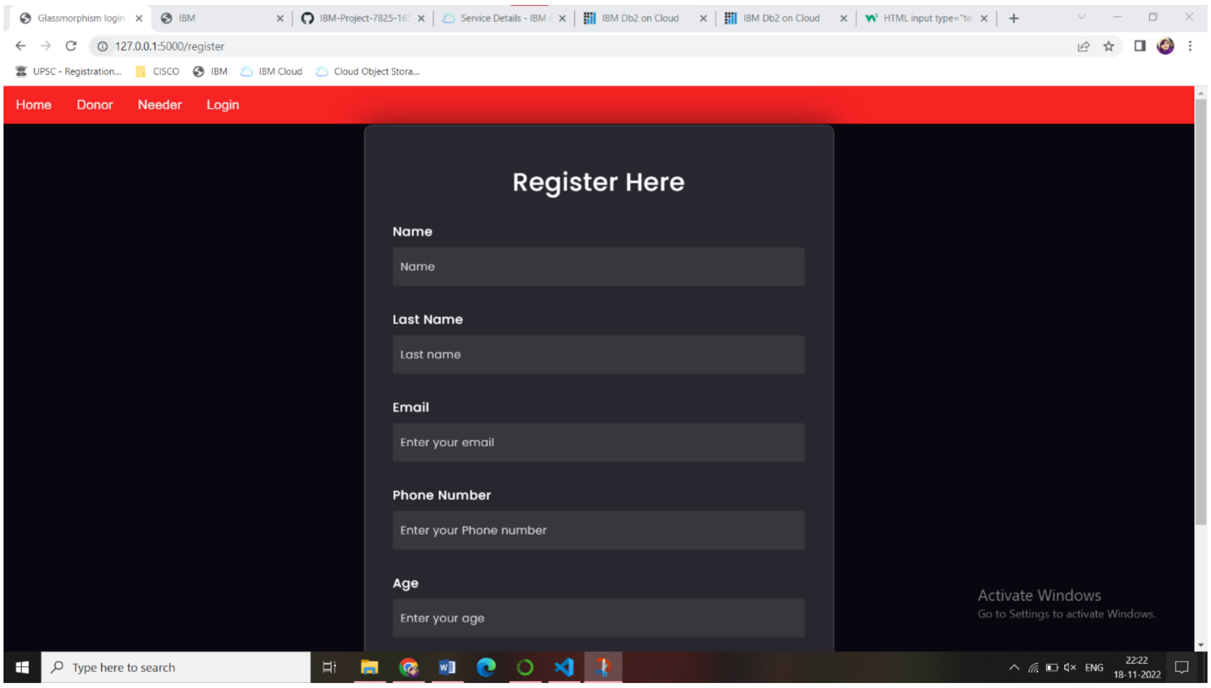
Email

Password

If you don't have a accout Register here!

Log In

If u don’t have an account register page



127.0.0.1:5000/register

Home Donor Needer Login

Register Here

Name

Name

Last Name

Last name

Email

Enter your email

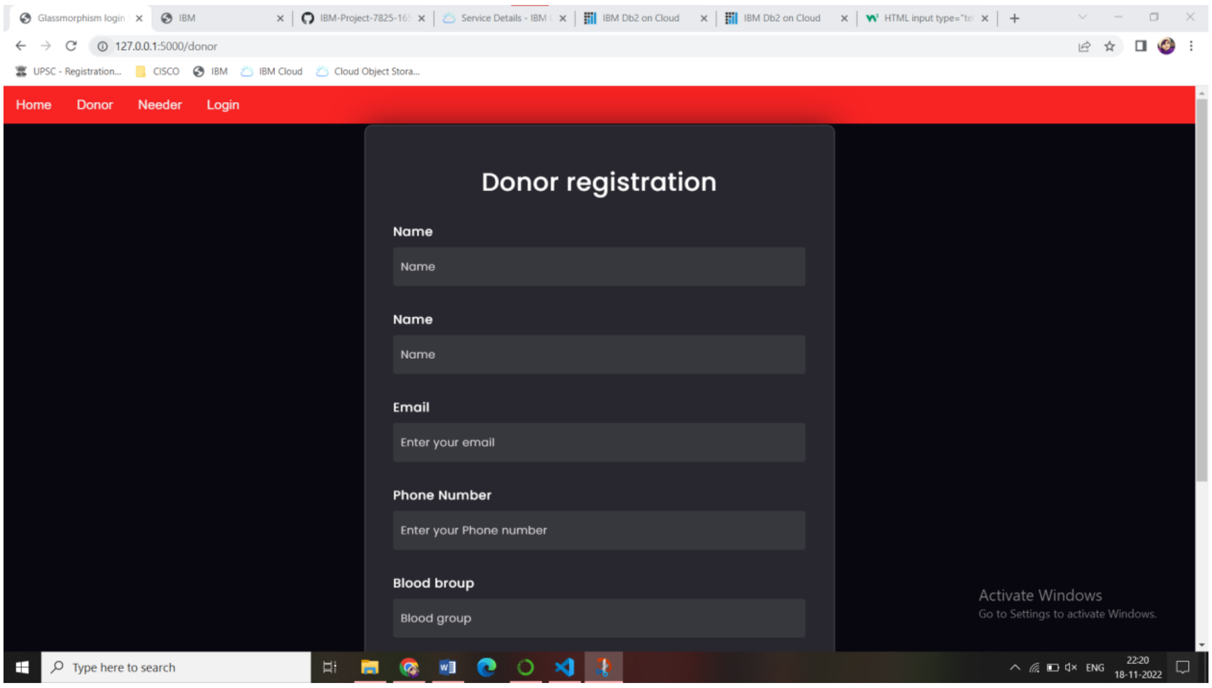
Phone Number

Enter your Phone number

Age

Enter your age

Donor Registration



127.0.0.1:5000/donor

Home Donor Needer Login

Donor registration

Name

Name

Name

Name

Email

Enter your email

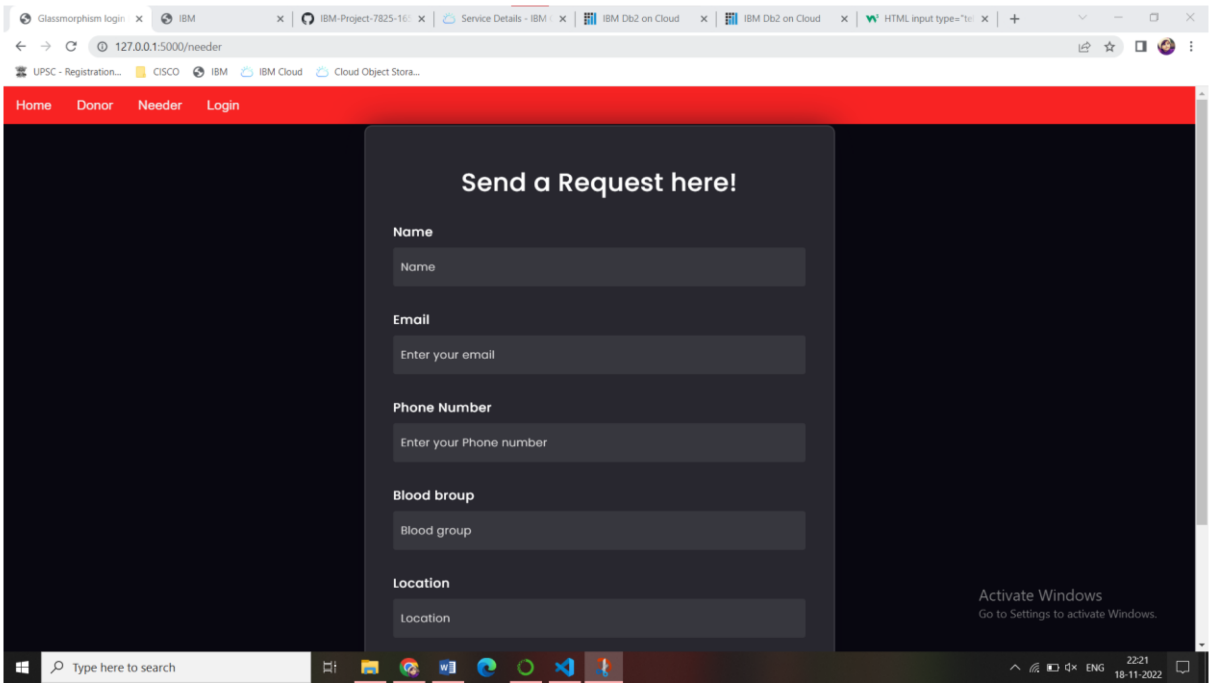
Phone Number

Enter your Phone number

Blood broup

Blood group

Needer Registration



Home Donor Needer Login

Send a Request here!

Name

Name

Email

Enter your email

Phone Number

Enter your Phone number

Blood broup

Blood group

Location

Location

ADVANTAGES AND DISADVANTAGES

10 ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- It is a useful tool to find compatible blood donors who can receive blood request posts in their local area. Clinics can use this web application to maintain blood donation activity.
- The prime motive of the app is to solve the perpetual shortfall of plasma donors.
- It's very easy to track the donors at different locations.
- It connects plasma donors and recipients through a single and scalable platform
- It works best at the time of emergency.
- This app does screening donors for existing health conditions.
- This project has a login page that allows only the registered user to login and thereby preventing unauthorized access.
- This system can be used to view all the donor details and accordingly select the right donor.
- The android mobile user will be able to make quick decisions selecting a donor. The usage of this application will greatly reduce time in selecting the right donor.

DISADVANTAGES

- People who donate blood may also experience side effects, such as minor bruising or feeling lightheaded.
- The android mobile user will not be able to insert or view details if the server goes down. Thus there is the disadvantage of single-point failure.

11 CONCLUSIONS

Donating blood helps save lives and has positive benefits for donors, such as improving their emotional and physical health. People who donate blood may also experience side effects, such as minor bruising or feeling lightheaded. This project aims to create a mobile app for android mobiles. the main aim of this project is to develop a computer system that will link all donors and blood banks. The system helps to control a blood transfusion service and create a database to hold data on donors and blood banks in each area or city. furthermore, people will be able to register as donors and thus receive a call from their local clients who needs blood to donate blood in cases of need. The app will help develop public awareness amongst its visitors of the hospitals' need for blood in order to supply the appropriate donors.

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.

APPENDIX

13 . APPENDIX

SOURCE CODE

Application file

app.py

```
from flask import Flask,render_template, request, redirect,
    url_for, session
import ibm_db
app = Flask(__name__)
conn                                                        =
    ibm_db.connect("DATABASE=bludb;HOSTNAME=21fecf
    d8-47b7-4937-840d-
    d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.
    cloud;PORT=31864;SECURITY=SSL;SSLServiceCertifica
    te=DigiCertGlobalRootCA.crt;UID=jhs68731;PWD=IA0K2
    2DLUUq1ncnj",'','')


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

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

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



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

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


@app.route('/addrec',methods = ['POST', 'GET'])
def addrec():
  if request.method == 'POST':
    name = request.form['name']
    lname = request.form['lname']
    email = request.form['email']
    phnum = request.form['phnum']
    age=request.form['age']
    bloodgrp = request.form['bloodgrp']


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

    if account:
      return render_template('index.html', msg="You are
    already a member, please login using your details")
    else:
      insert_sql = "INSERT INTO user VALUES (?,?,?,?,?,?)"
      prep_stmt = ibm_db.prepare(conn, insert_sql)
      ibm_db.bind_param(prep_stmt, 1, name)
      ibm_db.bind_param(prep_stmt, 2, lname)
      ibm_db.bind_param(prep_stmt, 3, email)
      ibm_db.bind_param(prep_stmt, 4, phnum)
      ibm_db.bind_param(prep_stmt, 5, age)
```

```
     ibm_db.bind_param(prep_stmt, 6, bloodgrp)

     ibm_db.execute(prep_stmt)

   return render_template('index.html', msg="Student Data
   saved successfuly.")


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

def loginpage():
   user = request.form['user']
   passw = request.form['passw']
   sql = "SELECT * FROM user WHERE name =? AND
   email=?"
   stmt = ibm_db.prepare(conn, sql)
   ibm_db.bind_param(stmt,1,user)
   ibm_db.bind_param(stmt,2,passw)
   ibm_db.execute(stmt)
   account = ibm_db.fetch_assoc(stmt)
   if account:
         return render_template('index.html')
   else:
      return render_template('login.html', pred="Login
   unsuccessful. Incorrect username / password !")

@app.route('/donorpage',methods = ['POST', 'GET'])
def donorpage():
  if request.method == 'POST':
   name = request.form['name']
   lname = request.form['lname']
   email = request.form['email']
   phnum = request.form['phnum']
   bloodgrp = request.form['bloodgrp']
   location=request.form['location']
   donated=request.form['donated']
```

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

    if account:
     return render_template('index.html', msg="You are
    already a member, please login using your details")
    else:
     insert_sql = "INSERT INTO donor VALUES
    (?,?,?,?,?,?,?)"
     prep_stmt = ibm_db.prepare(conn, insert_sql)
     ibm_db.bind_param(prep_stmt, 1, name)
     ibm_db.bind_param(prep_stmt, 2, lname)
     ibm_db.bind_param(prep_stmt, 3, email)
     ibm_db.bind_param(prep_stmt, 4, phnum)
     ibm_db.bind_param(prep_stmt, 5, bloodgrp)
     ibm_db.bind_param(prep_stmt,6, location)
     ibm_db.bind_param(prep_stmt, 7, donated)

     ibm_db.execute(prep_stmt)

    return render_template('index.html', msg="Student Data
    saved successfuly.")
@app.route('/neederpage',methods = ['POST', 'GET'])
def neederpage():
  if request.method == 'POST':
    name = request.form['name']
    email = request.form['email']
    phnum = request.form['phnum']
    bloodgrp = request.form['bloodgrp']
    location=request.form['location']

    insert_sql = "INSERT INTO needer VALUES (?,?,?,?,?)"
```

```
prep_stmt = ibm_db.prepare(conn, insert_sql)
ibm_db.bind_param(prep_stmt, 1, name)
ibm_db.bind_param(prep_stmt, 2, email)
ibm_db.bind_param(prep_stmt, 3, phnum)
ibm_db.bind_param(prep_stmt, 4, bloodgrp)
ibm_db.bind_param(prep_stmt,5, location)


ibm_db.execute(prep_stmt)

return render_template('index.html', msg="Student Data
 saved successfuly.")
```

Templates
HTML pages

donor.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Design by foolishdeveloper.com -->
    <title>Glassmorphism login Form Tutorial in html
     css</title>

    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link rel="stylesheet"
     href="https://cdnjs.cloudflare.com/ajax/libs/font-
     awesome/5.15.4/css/all.min.css">
    <link
     href="https://fonts.googleapis.com/css2?family=Poppins:wg
     ht@300;500;600&display=swap" rel="stylesheet">
    <!--Stylesheet-->
    <style media="screen">
      *,
*:before,
*:after{
    padding: 0;
    margin: 0;
    box-sizing: border-box;
}
body{
    background-color: #080710;
}
.background{

    width: 230px;
    height: 520px;
    position: absolute;
    transform: translate(-50%,-50%);
    left: 50%;
    top: 50%;
}
.background .shape{
    height: 200px;
    width: 200px;
    position: absolute;
    border-radius: 50%;
}
.vertical-center {
  margin: 0;
  position: absolute;
  top: 50%;
  -ms-transform: translateY(-50%);
  transform: translateY(-50%);
}


form{
    margin-top:8%;
    height: 120%;

    width: 600px;
    background-color: rgba(255,255,255,0.13);
    position: absolute;
    transform: translate(-50%,-50%);
    top: 50%;
    left: 50%;
    border-radius: 10px;
    backdrop-filter: blur(10px);
    border: 2px solid rgba(255,255,255,0.1);
    box-shadow: 0 0 40px rgba(8,7,16,0.6);
    padding: 50px 35px;
}
```

```
form *{
   font-family: 'Poppins',sans-serif;
   color: #ffffff;
   letter-spacing: 0.5px;
   outline: none;
   border: none;
}
form h3{
   font-size: 32px;
   font-weight: 500;
   line-height: 42px;
   text-align: center;
}

label{
   display: block;
   margin-top: 30px;
   font-size: 16px;
   font-weight: 500;
}
input{
   display: block;
   height: 50px;
   width: 100%;
   background-color: rgba(255,255,255,0.07);
   border-radius: 3px;
   padding: 0 10px;
   margin-top: 8px;
   font-size: 14px;
   font-weight: 300;
}
::placeholder{
   color: #e5e5e5;
}
button{
   margin-top: 50px;
   width: 100%;
   background-color: #ffffff;
   color: #080710;
   padding: 15px 0;
   font-size: 18px;
   font-weight: 600;
   border-radius: 5px;
   cursor: pointer;
}
.social{
  margin-top: 30px;
  display: flex;
}
.social div{
  background: red;
  width: 150px;
  border-radius: 3px;
  padding: 5px 10px 10px 5px;
  background-color: rgba(255,255,255,0.27);
  color: #eaf0fb;
  text-align: center;
}
.social div:hover{
  background-color: rgba(255,255,255,0.47);
}
.social .fb{
  margin-left: 25px;
}
.social i{
  margin-right: 4px;
}

   </style>
</head>
```

```
{% include 'navbar.html' %}
<body>
   <div class="background">
      <div class="shape"></div>
      <div class="shape"></div>
   </div>
   <form action="{{url_for('donorpage')}}" method="POST">
      <h3>Donor registration</h3>

      <label for="uname">Name</label>
      <input type="text" placeholder="Name" name="name">

      <label for="uname">Name</label>
      <input type="text" placeholder="Name" name="lname">


      <label for="Email">Email</label>
      <input type="text" placeholder="Enter your email" name="email">

      <label for="phnum">Phone Number</label>
      <input type="number" placeholder="Enter your Phone number" name="phnum">


      <label for="blood group">Blood broup</label>
      <input type="text" placeholder="Blood group" name="bloodgrp">


      <label for="location">Location</label>
      <input type="text" placeholder="Location" name="location">
```

```
    <label>Are you donated already??</label>
    <input type="text" placeholder="Type yes or no!"
  name="donated">




    <button style="height: 5%; width: 20%; align-items:
  center;">Register</button>


  </form>
</body>
</html>
```

index.html

```
{%include 'navbar.html' %}
<!DOCTYPE html>
<html>
<head>
  <meta   name="viewport"   content="width=device-width,
   initial-scale=1">
  <style>
    * {box-sizing: border-box}
    body {font-family: Verdana, sans-serif; margin:0}
    .mySlides {display: none}
    img {vertical-align: middle;}
    /* Slideshow container */
    .slideshow-container {
      max-width: 1000px;
      position: relative;
      margin: auto;
    }
    /* Next & previous buttons */
    .prev, .next {
      cursor: pointer;
      position: absolute;
      top: 50%;
      width: auto;
      padding: 16px;
      margin-top: -22px;
      color: white;
      font-weight: bold;
      font-size: 18px;
      transition: 0.6s ease;
      border-radius: 0 3px 3px 0;
      user-select: none;
    }
```

```
 /* Position the "next button" to the right */
 .next {
   right: 0;
   border-radius: 3px 0 0 3px;
 }
 /* On hover, add a black background color with a little bit
seethrough */
 .prev:hover, .next:hover {
   background-color: rgba(0,0,0,0.8);
 }
 /* Caption text */
 .text {
   color: black;
   font-size: 15px;
   padding: 8px 12px;
   position: absolute;
   bottom: 8px;
   width: 100%;
   text-align: center;
 }
 /* Number text (1/3 etc) */
 .numbertext {
   color: #f2f2f2;
   font-size: 12px;
   padding: 8px 12px;
   position: absolute;
   top: 0;
 }
 /* The dots/bullets/indicators */
 .dot {
   cursor: pointer;
   height: 15px;
   width: 15px;
   margin: 0 2px;
   background-color: #bbb;
```

```
        border-radius: 50%;
        display: inline-block;
        transition: background-color 0.6s ease;
      }
      .active, .dot:hover {
        background-color: #717171;
      }
    </style>
</head>
<body>
    <div class="slideshow-container">
      <div class="mySlides fade">
        <div class="numbertext">1 / 3</div>
        <img
     src="https://cdn.pixabay.com/photo/2019/04/29/16/56/blood
     -donation-4166552_960_720.jpg" style="width:100%">
        <div class="text">Donate blood</div>
        </div>
        <div class="mySlides fade">
          <div class="numbertext">2 / 3</div>
          <img
     src="https://cdn.pixabay.com/photo/2017/10/11/21/07/blood
     -2842450_960_720.jpg  " style="width:100%;">
          <div class="text"></div>
        </div>
        <div class="mySlides fade">
          <div class="numbertext">3 / 3</div>
          <img
     src="https://cdn.pixabay.com/photo/2020/02/19/06/23/earth
     -4861456_960_720.jpg" style="width:100%">
          <div class="text">Save World!!!</div>
        </div>
        <a class="prev" onclick="plusSlides(-1)">❮</a>
        <a class="next" onclick="plusSlides(1)">❯</a>
    </div>
```

```
  <br>
  <div style="text-align:center">
    <span class="dot" onclick="currentSlide(1)"></span>
    <span class="dot" onclick="currentSlide(2)"></span>
    <span class="dot" onclick="currentSlide(3)"></span>
  </div>
  <script>
    var slideIndex = 1;
    showSlides(slideIndex);
    function plusSlides(n) {
      showSlides(slideIndex += n);
    }
    function currentSlide(n) {
      showSlides(slideIndex = n);
    }
    function showSlides(n) {
      var i;
      var                    slides                    =
   document.getElementsByClassName("mySlides");
      var dots = document.getElementsByClassName("dot");
      if (n > slides.length) {slideIndex = 1}
      if (n < 1) {slideIndex = slides.length}
      for (i = 0; i < slides.length; i++) {
        slides[i].style.display = "none";
      }
      for (i = 0; i < dots.length; i++) {
        dots[i].className = dots[i].className.replace(" active",
   "");
      }
      slides[slideIndex-1].style.display = "block";
      dots[slideIndex-1].className += " active";
    }
  </script>
</body>
</html>
```

login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Design by foolishdeveloper.com -->
    <title>Glassmorphism login Form Tutorial in html
    css</title>

    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/font-
    awesome/5.15.4/css/all.min.css">
    <link
    href="https://fonts.googleapis.com/css2?family=Poppins:wg
    ht@300;500;600&display=swap" rel="stylesheet">
    <!--Stylesheet-->
    <style media="screen">
      *,
*:before,
*:after{
    padding: 0;
    margin: 0;
    box-sizing: border-box;
}
body{
    background-color: #080710;
}
.background{
    width: 430px;
    height: 520px;
    position: absolute;
    transform: translate(-50%,-50%);
    left: 50%;
    top: 50%;
}
.background .shape{
    height: 200px;
    width: 200px;
    position: absolute;
    border-radius: 50%;
}

form{
    height: 520px;
    width: 400px;
    background-color: rgba(255,255,255,0.13);
    position: absolute;
    transform: translate(-50%,-50%);
    top: 50%;
    left: 50%;
    border-radius: 10px;
    backdrop-filter: blur(10px);
    border: 2px solid rgba(255,255,255,0.1);
    box-shadow: 0 0 40px rgba(8,7,16,0.6);
    padding: 50px 35px;
}
form *{
    font-family: 'Poppins',sans-serif;
    color: #ffffff;
    letter-spacing: 0.5px;
    outline: none;
    border: none;
}
form h3{
    font-size: 32px;
    font-weight: 500;
    line-height: 42px;
    text-align: center;
}
```

```
label{
   display: block;
   margin-top: 30px;
   font-size: 16px;
   font-weight: 500;
}
input{
   display: block;
   height: 50px;
   width: 100%;
   background-color: rgba(255,255,255,0.07);
   border-radius: 3px;
   padding: 0 10px;
   margin-top: 8px;
   font-size: 14px;
   font-weight: 300;
}
::placeholder{
   color: #e5e5e5;
}
button{
   margin-top: 50px;
   width: 100%;
   background-color: #ffffff;
   color: #080710;
   padding: 15px 0;
   font-size: 18px;
   font-weight: 600;
   border-radius: 5px;
   cursor: pointer;
}
.social{
  margin-top: 30px;
  display: flex;
}
.social div{
  background: red;
  width: 150px;
  border-radius: 3px;
  padding: 5px 10px 10px 5px;
  background-color: rgba(255,255,255,0.27);
  color: #eaf0fb;
  text-align: center;
}
.social div:hover{
  background-color: rgba(255,255,255,0.47);
}
.social .fb{
  margin-left: 25px;
}
.social i{
  margin-right: 4px;
}

    </style>
</head>
{%include 'navbar.html' %}
<body>
    <div class="background">
        <div class="shape"></div>
        <div class="shape"></div>
    </div>
    <form action="{{url_for('loginpage')}}" method="POST">
        <h3>Login Here</h3>

        <label for="username">Username</label>
        <input type="text" placeholder="Email or Phone"
    name="user">
```

```
        <label for="password">Email</label>
        <input type="password" placeholder="Password"
    name="passw">
        <a href="{{url_for('register')}}"><p>If you don't have a
    accout Register here!</p></a>

        <button>Log In</button>

    </form>
</body>
</html>
```

navbar.html

```
<!DOCTYPE html>
<html>
<head>
<meta   name="viewport"   content="width=device-width,
   initial-scale=1">
<style>
body {
  margin: 0;
  font-family: Arial, Helvetica, sans-serif;
}

.topnav {
  overflow: hidden;
  background-color: rgb(248, 35, 35);
}

.topnav a {
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
}

.topnav a:hover {
  background-color: #ddd;
  color: black;
}
```

```
</style>
</head>
<body>

<div class="topnav">
  <a class="active" href="{{url_for('home')}}">Home</a>
  <a href="{{url_for('donor')}}">Donor</a>
  <a href="{{url_for('needer')}}">Needer</a>
  <a href="{{url_for('login')}}">Login</a>
</div>


</body>
</html>
```

needer.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Design by foolishdeveloper.com -->
    <title>Glassmorphism login Form Tutorial in html
     css</title>

    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link rel="stylesheet"
     href="https://cdnjs.cloudflare.com/ajax/libs/font-
     awesome/5.15.4/css/all.min.css">
    <link
     href="https://fonts.googleapis.com/css2?family=Poppins:wg
     ht@300;500;600&display=swap" rel="stylesheet">
    <!--Stylesheet-->
    <style media="screen">
      *,
*:before,
*:after{
    padding: 0;
    margin: 0;
    box-sizing: border-box;
}
body{
    background-color: #080710;
}
.background{

    width: 230px;
    height: 520px;
    position: absolute;
    transform: translate(-50%,-50%);
    left: 50%;
    top: 50%;
}
.background .shape{
    height: 200px;
    width: 200px;
    position: absolute;
    border-radius: 50%;
}
.vertical-center {
  margin: 0;
  position: absolute;
  top: 50%;
  -ms-transform: translateY(-50%);
  transform: translateY(-50%);
}

form{
    margin-top:8%;
    height: 120%;

    width: 600px;
    background-color: rgba(255,255,255,0.13);
    position: absolute;
    transform: translate(-50%,-50%);
    top: 50%;
    left: 50%;
    border-radius: 10px;
    backdrop-filter: blur(10px);
    border: 2px solid rgba(255,255,255,0.1);
    box-shadow: 0 0 40px rgba(8,7,16,0.6);
    padding: 50px 35px;
}
form *{
    font-family: 'Poppins',sans-serif;
    color: #ffffff;
    letter-spacing: 0.5px;
    outline: none;
    border: none;
}
form h3{
    font-size: 32px;
    font-weight: 500;
    line-height: 42px;
    text-align: center;
}

label{
    display: block;
    margin-top: 30px;
    font-size: 16px;
    font-weight: 500;
}
input{
    display: block;
    height: 50px;
    width: 100%;
    background-color: rgba(255,255,255,0.07);
    border-radius: 3px;
    padding: 0 10px;
    margin-top: 8px;
    font-size: 14px;
    font-weight: 300;
}
::placeholder{
    color: #e5e5e5;
}
button{
    margin-top: 50px;
    width: 100%;
    background-color: #ffffff;
    color: #080710;
    padding: 15px 0;
    font-size: 18px;
    font-weight: 600;
    border-radius: 5px;
    cursor: pointer;
}
.social{
  margin-top: 30px;
  display: flex;
}
.social div{
  background: red;
  width: 150px;
  border-radius: 3px;
  padding: 5px 10px 10px 5px;
  background-color: rgba(255,255,255,0.27);
  color: #eaf0fb;
  text-align: center;
}
.social div:hover{
  background-color: rgba(255,255,255,0.47);
}
.social .fb{
  margin-left: 25px;
}
.social i{
  margin-right: 4px;
}

    </style>
</head>
{% include 'navbar.html' %}
```

```
<body>
  <div class="background">
    <div class="shape"></div>
    <div class="shape"></div>
  </div>
  <form action="{{url_for('neederpage')}}"
  method="POST">
    <h3>Send a Request here!</h3>

    <label for="uname">Name</label>
    <input type="text" placeholder="Name" name="name">

    <label for="Email">Email</label>
    <input type="text" placeholder="Enter your email"
  name="email">

    <label for="phnum">Phone Number</label>
    <input type="number" placeholder="Enter your Phone
  number" name="phnum">

    <label for="blood group">Blood broup</label>
    <input type="text" placeholder="Blood group"
  name="bloodgrp">

    <label for="location">Location</label>
    <input type="text" placeholder="Location"
  name="location">


    <button style="height: 5%; width: 20%; align-items:
  center;">Register</button>


  </form>
</body>
</html>
```

register.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Design by foolishdeveloper.com -->
    <title>Glassmorphism login Form Tutorial in html
     css</title>

    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link                                        rel="stylesheet"
     href="https://cdnjs.cloudflare.com/ajax/libs/font-
     awesome/5.15.4/css/all.min.css">
    <link
     href="https://fonts.googleapis.com/css2?family=Poppins:wg
     ht@300;500;600&display=swap" rel="stylesheet">
    <!--Stylesheet-->
    <style media="screen">
      *,
*:before,
*:after{
    padding: 0;
    margin: 0;
    box-sizing: border-box;
}
body{
    background-color: #080710;
}
.background{

    width: 230px;
    height: 520px;
    position: absolute;
    transform: translate(-50%,-50%);
    left: 50%;
    top: 50%;
}
.background .shape{
    height: 200px;
    width: 200px;
    position: absolute;
    border-radius: 50%;
}
.vertical-center {
  margin: 0;
  position: absolute;
  top: 50%;
  -ms-transform: translateY(-50%);
  transform: translateY(-50%);
}


form{
    margin-top:8%;
    height: 120%;

    width: 600px;
    background-color: rgba(255,255,255,0.13);
    position: absolute;
    transform: translate(-50%,-50%);
    top: 50%;
    left: 50%;
    border-radius: 10px;
    backdrop-filter: blur(10px);
    border: 2px solid rgba(255,255,255,0.1);
    box-shadow: 0 0 40px rgba(8,7,16,0.6);
    padding: 50px 35px;
}
```

```
form *{
   font-family: 'Poppins',sans-serif;
   color: #ffffff;
   letter-spacing: 0.5px;
   outline: none;
   border: none;
}
form h3{
   font-size: 32px;
   font-weight: 500;
   line-height: 42px;
   text-align: center;
}

label{
   display: block;
   margin-top: 30px;
   font-size: 16px;
   font-weight: 500;
}
input{
   display: block;
   height: 50px;
   width: 100%;
   background-color: rgba(255,255,255,0.07);
   border-radius: 3px;
   padding: 0 10px;
   margin-top: 8px;
   font-size: 14px;
   font-weight: 300;
}
::placeholder{
   color: #e5e5e5;
}
button{
   margin-top: 50px;
   width: 100%;
   background-color: #ffffff;
   color: #080710;
   padding: 15px 0;
   font-size: 18px;
   font-weight: 600;
   border-radius: 5px;
   cursor: pointer;
}
.social{
  margin-top: 30px;
  display: flex;
}
.social div{
  background: red;
  width: 150px;
  border-radius: 3px;
  padding: 5px 10px 10px 5px;
  background-color: rgba(255,255,255,0.27);
  color: #eaf0fb;
  text-align: center;
}
.social div:hover{
  background-color: rgba(255,255,255,0.47);
}
.social .fb{
  margin-left: 25px;
}
.social i{
  margin-right: 4px;
}

   </style>
</head>
```

```
{% include 'navbar.html' %}
<body>
   <div class="background">
      <div class="shape"></div>
      <div class="shape"></div>
   </div>
   <form action="{{url_for('addrec')}}" method="POST">
      <h3>Register Here</h3>

      <label for="uname">Name</label>
      <input type="text" placeholder="Name" name="name">

      <label for="username">Last Name</label>
      <input type="text" placeholder="Last name"
   name="lname">
    <label for="Email">Email</label>
      <input type="text" placeholder="Enter your email"
   name="email">
  <label for="phnum">Phone Number</label>

      <input type="number" placeholder="Enter your Phone
   number" name="phnum">

      <label for="age">Age</label>
      <input type="number" placeholder="Enter your age"
   name="age">
       <label for="blood group">Blood broup</label>

      <input type="text" placeholder="Blood group"
   name="bloodgrp">

      <button style="height: 5%; width: 20%; align-items:
   center;">Register</button>

   </form>
</body>
</html>
```

GITHUB AND PROJECT DEMO LINK :

GitHub Link
https://github.com/IBM-EPBL/IBM-Project-7825-1658900440

Demo Video link :
https://youtu.be/U7l6ggSCq6E