IBM NALAIYATHIRAN
PROJECT REPORT

PLASMA DONOR APPLICATION

TEAM ID	PNT2022TMID39184
PROJECT NAME	Plasma Donor Application
TEAM MEMBERS	-Sathish RK -Anand K -Dhilkumar C -Vishnu V

TABLE OF CONTENTS

1. INTRODUCTION

Project Overview :

In Register or Login user starting the plasma donor application should login whether the user is donor or receiptant , if the user is new to application the user should Register and then he/she should login to the application. Conventionally, when a patient needs blood, he/she has to contact a blood bank or a compatible blood group of a donor in their circle, family, and friends. However, it is difficult to find suitable donor within a limited group of people in a given time. In addition, there is no guarantee that blood banks will have compatible blood group in stock. There is also steady increase in blood donation requests posts in social networking sites (like Facebook, twitter, Instagram, etc.) requesting for donation.

In the Plasma Donor Application the Dashboard for Donor and Receiptant.In the Register Dashboard user is a new donor , the user should enter all the details including blood group , contact details ,current location etc.. The data entered by the donor stored by the application .

The receiptant should request for the plasma and check the dashboard for the availablity of the plasma donor in their nearest location. if the donor is available the receiptant should enter the details of receiptant , if the details are matched with the donors the details of receiptant entered are send to the particular donors via email , then the donor will accept the request and contact the receiptant. The receiptant can contact the donors directly through contact details provided by the donors.

Purpose :

During the COVID 19 crisis, the requirement for plasma became high and the donor count was low. Saving the donor information and helping the need by notifying the current donors would be a helping hand. Regarding the problem faced, an application is to be built which would take the donor details, store them and inform them upon a request.The main objective of this project is to provide the recipient with a donor who is in good form with no health ailments to donate blood of the corresponding blood group.

Plasma Donor Application can resolve the issues by connecting patients promptly with a large pool of donors in the same region via an authorized clinic. When a patient needs a blood donation, the clinic (where the patient is admitted) can use the application to contact the blood donors in the vicinity or nearby city based on their location. The registered donors will get notification about the blood donation needed at a specific clinic where they can go and donate.

Plasma Donor Application provides donors with functionalities including “blood request feed”, “donation history”, “invite friend”, and “book an appointment” (with the clinic to donate blood), at the same time the requester (clinic) can send requests and use this application to maintain the different blood donation activities.

2. LITERATURE SURVEY

Existing problem :

In existing problem finding a donor requires much time by contecting blood banks and third party agencies. During the time of emergency the receiptant facing problem in finding a donor . Though some of online applications are available in internet they are less userfriendly and not secure and some of the application has fake data and the receipatant cannot find a donor at emergency time .

References :

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

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

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

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

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

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

Problem Statement Definition:

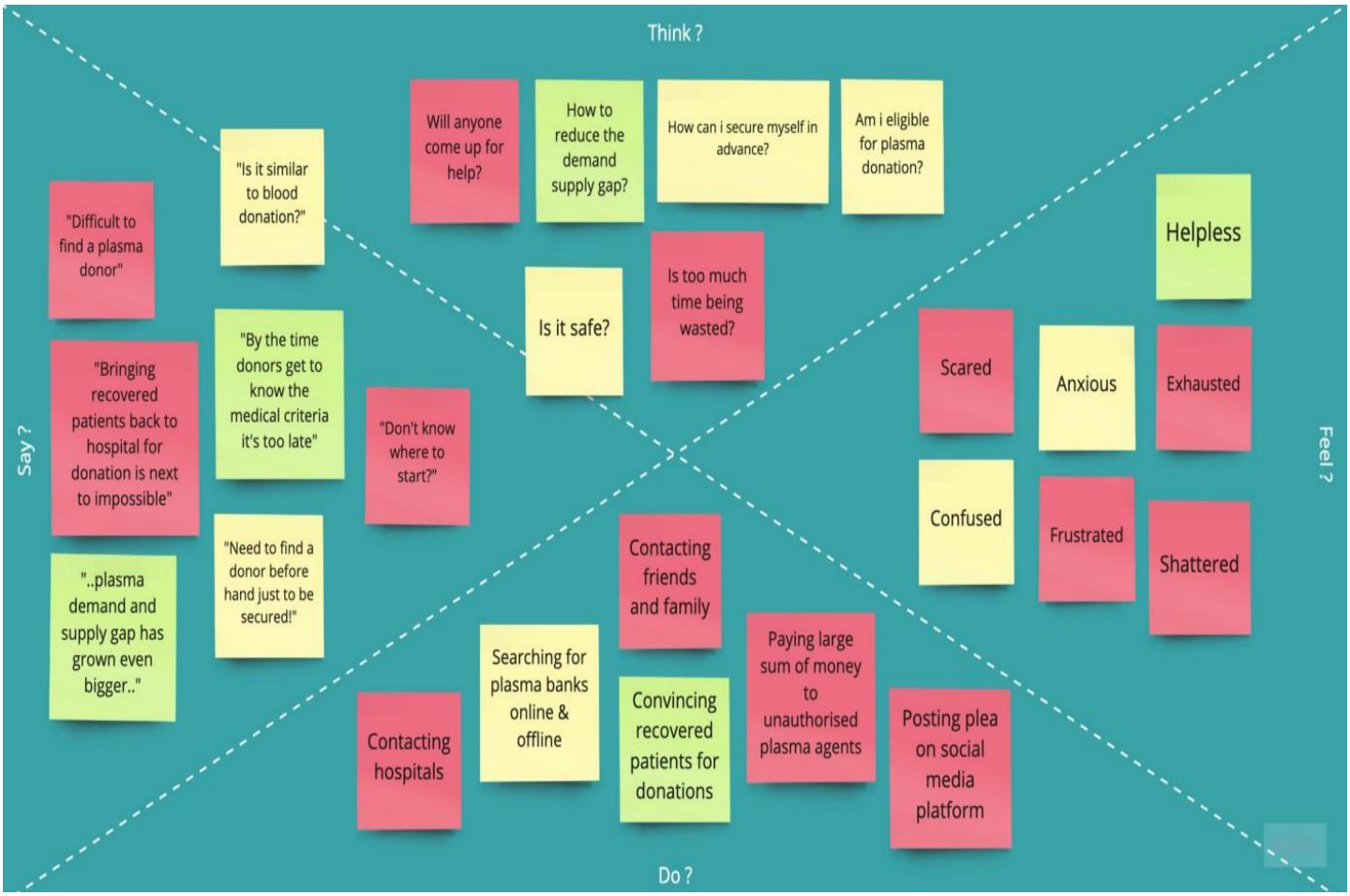
During COVID 19 crisis the requirement for plasma increased drastically as there were no vaccinations found in order to treat the infected patients.

In such situation it was very difficult to find the plasma donor, check whether the donor was infected previously and was recovered, and which donor is eligible to donate plasma was a challenging task.

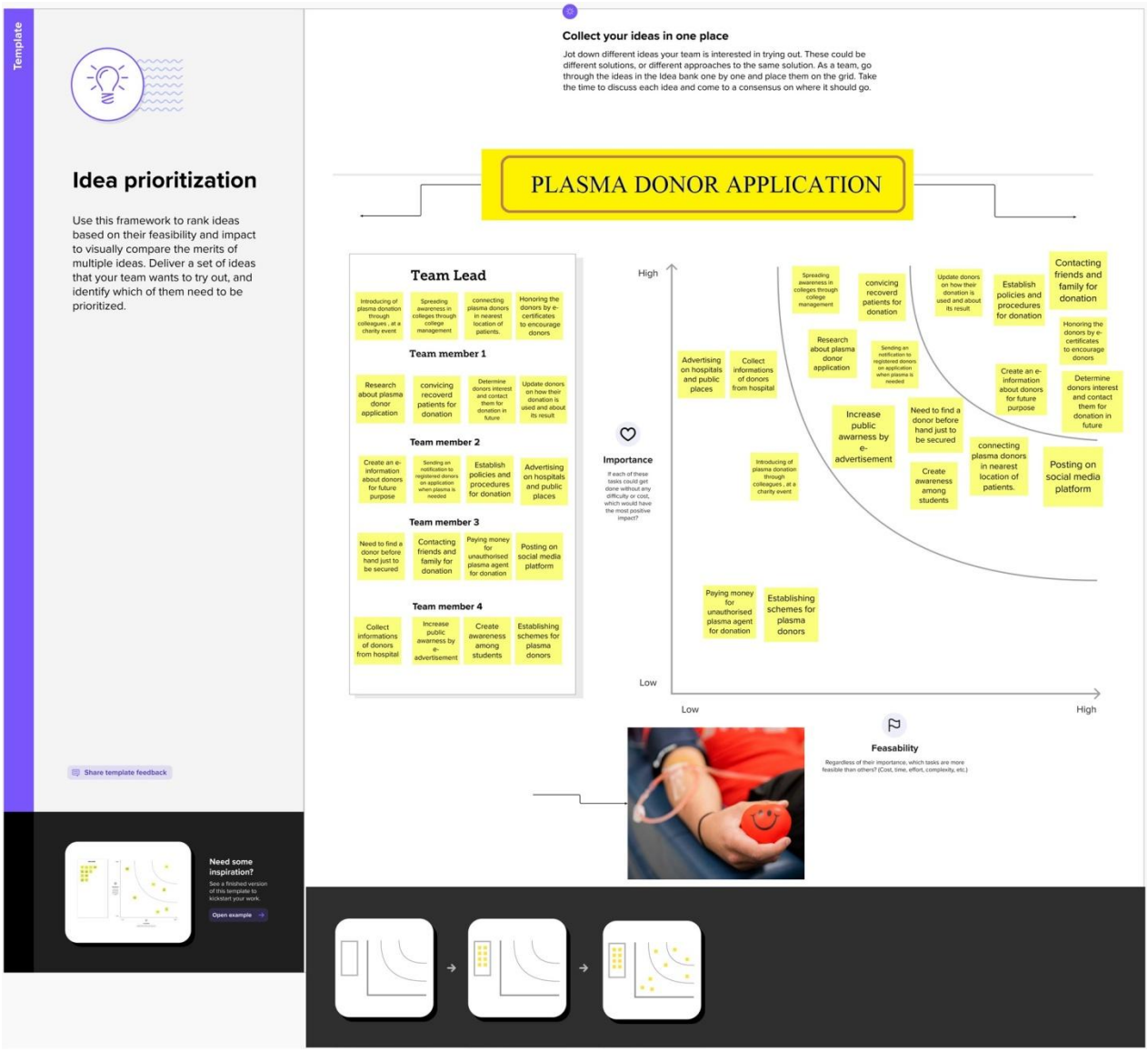
As the plasma therapy was one of the ways to treat the infected patients getting the donor details played a major role.

3. IDEATION & PROPOSED SOLUTION

Empathy Map :



Ideation & Brainstorming :



Proposed Solution

S.No.	Parameter	Description
1.	**Problem Statement (Problem to be solved)**	Saving the donor information and helping the needy by notifying the current donors list has become an important issue. We have to create an application that can connect the User and the Donor seamlessly.
2.	**Idea / Solution description**	The application we create will be able to connect the user and donor where the user can also become a donor if he wishes. When the user requests a specific blood plasma all the suitable donors of the particular blood type and location are notified.
3.	**Novelty / Uniqueness**	Enabling Users to filter and chose the donor of their choice with respect to location, blood type etc... and Connecting blood banks and blood camps to users giving them more
4.	**Social Impact / Customer Satisfaction**	User-friendly interface with an efficient, fast, and seamless connectivity between donor and acceptor. Creates a Plasma donation community that has both contributors and end users that are equally profited and create a sense of safety and assurance when referring to their needs for immediate blood plasma requirement and saves life.
5.	**Business Model (Revenue Model)**	The application is free to use and it comes under healthcare domain. It helps people who want to donate plasma to the people who need it. Data can be stored in IBM DB2 in cloud which reduces the

		overall cost incurred for developing the application. And we are not developing for revenue, we are doing this as social service.
6.	**Scalability of the Solution**	Global connectivity that creates a community all over the world ensuring all the emergency needs are acknowledged and are catered to at the required time. Establish a reliability factor of each user that ensures the delivery of service based on the user rating

Problem Solution fit

1. CUSTOMER SEGMENT(S)

- Peoples who are willing to donate plasma.
- Patients who need plasma

4. EMOTION BEFORE /AFTER

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

7.BEHAVIOUR

- Directly related: When the User requires a specific blood plasma type, they request for that specific blood plasma type and any donor that are available with the suitable type are notified.
- Indirectly associated: Contribute to the Blood banks available offline as well to update and cater to needs in places where internet connection is not possible or stable.

2. JOBS TO BE DONE/ PROBLEMS

The customer will be able to get the donor details and availability upon immediate request without any details.

Create awareness about plasma donation to save once life.

5. AVAILABLE SOLUTION

In the existing available solution, there is no intermediate to connect the plasma donors and patient/clients they will receive plasma through blood bank or hospitals.

In existing solution, they tried to find plasma donors without accessing internet.

This system has many disadvantages that we can't find the donors of same group and nearest location at the time of emergency.

8.CHANNELS OF BEHAVIOUR

- Users get their e-certificate after donating plasma and get details about how their donation helpful
- Registering themselves to donate plasma.

3. TRIGGER

Customers are exposed to existing services provided by our application assuring the timely and effective service catering to their needs during emergency so they will tie up with us .

6. CUSTOMER CONSTRAINTS

- Network connectivity
- Available Portable devices
- Donors availability at required time
- Donors reputability
- Location Constraints.

9. PROBLEM ROOT CAUSE

- Lack of information/awareness based on the need to donate plasma and due to this the scarcity created in the blood banks and other factors like Covid-19,lockdowns affect this drastically

10. YOUR SOLUTION

By using internet, we can connect the donors and patients/clients. The patients can search the plasma donor of same blood group, nearest location and etc.. . By creating the awareness among college students, public to register in the application. The patient can contact them whenever they need. In addition to it, the donors list and contact are collected from the various blood banks, hospitals and registered in the application so that the patients can contact them whenever they need.

4. REQUIREMENT ANALYSIS

Functional requirement

FR No.	**Functional Requirement (Epic)**	**Description**
FR-1	User Registration	Registration to plasma donor application is done by submitting the details of donors in the form in plasma donor application and through LinkedIn.
FR-2	User Confirmation	Users will be confirmed by sending them OTP through Gmail or phone number they provided.
FR-3	User Login	User can be login through submitting the login form by the details they provided in registration (i.e., username, email, password). The user can login when they provide the valid information.
FR-4	Plasma Donor	The user of our application is of two types. (i.e., The plasma donor and the plasma receiver). The donor should enter the full details such as blood group, location, contact information etc... The

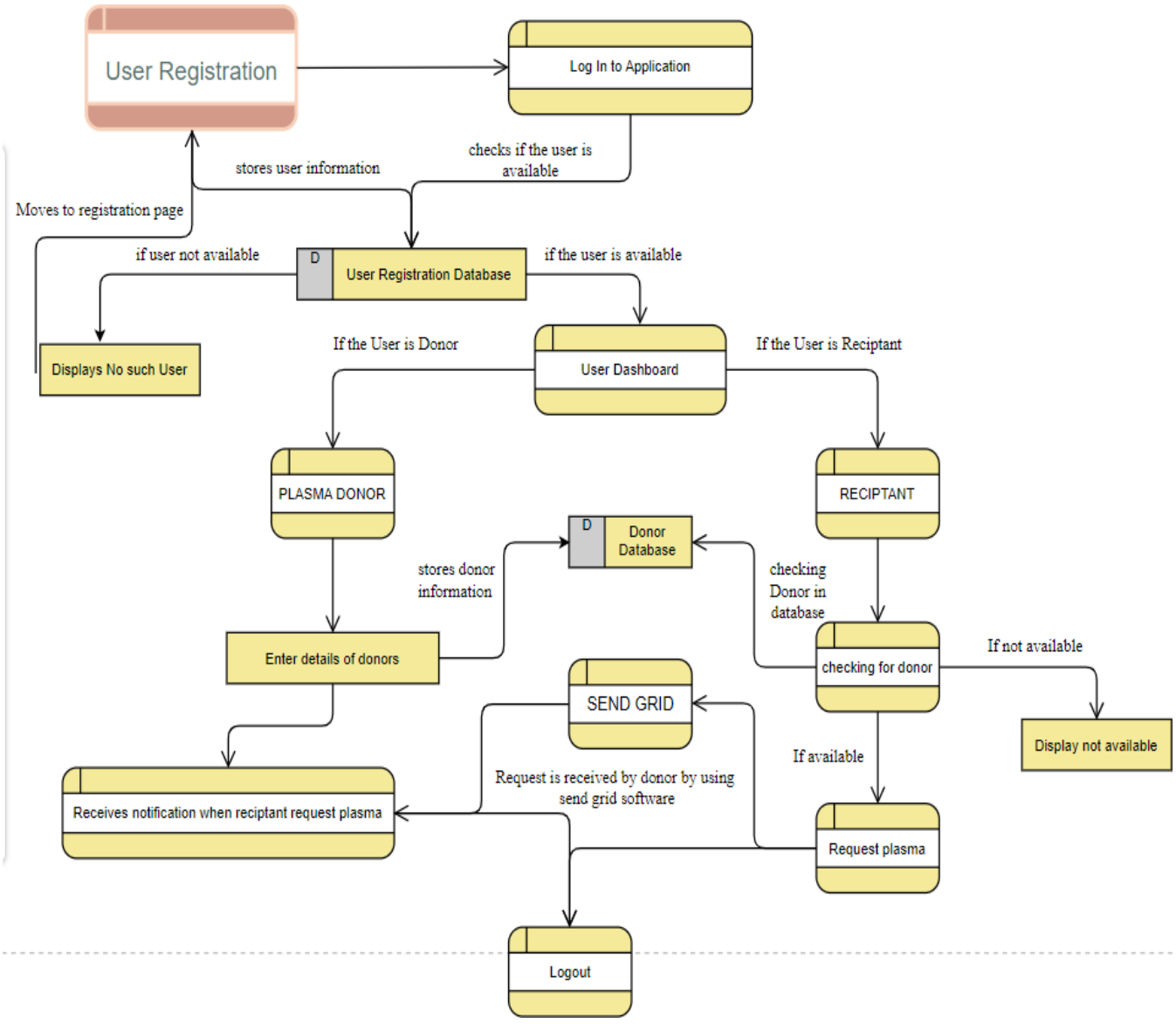
		donor receives a notification when a receives send a request to the donor.
FR-5	Plasma Receiver	The receiver of plasma can be classified into two types. (i.e., patient and client /blood banks). Both can send a request to the donor and can contact them directly through contact information they provided. The plasma receiver can also register as a donor to donate a plasma.
FR-6	User Logout	Both the Donor and Receiver can logout the application by clicking on logout in the application.

Non-Functional requirements

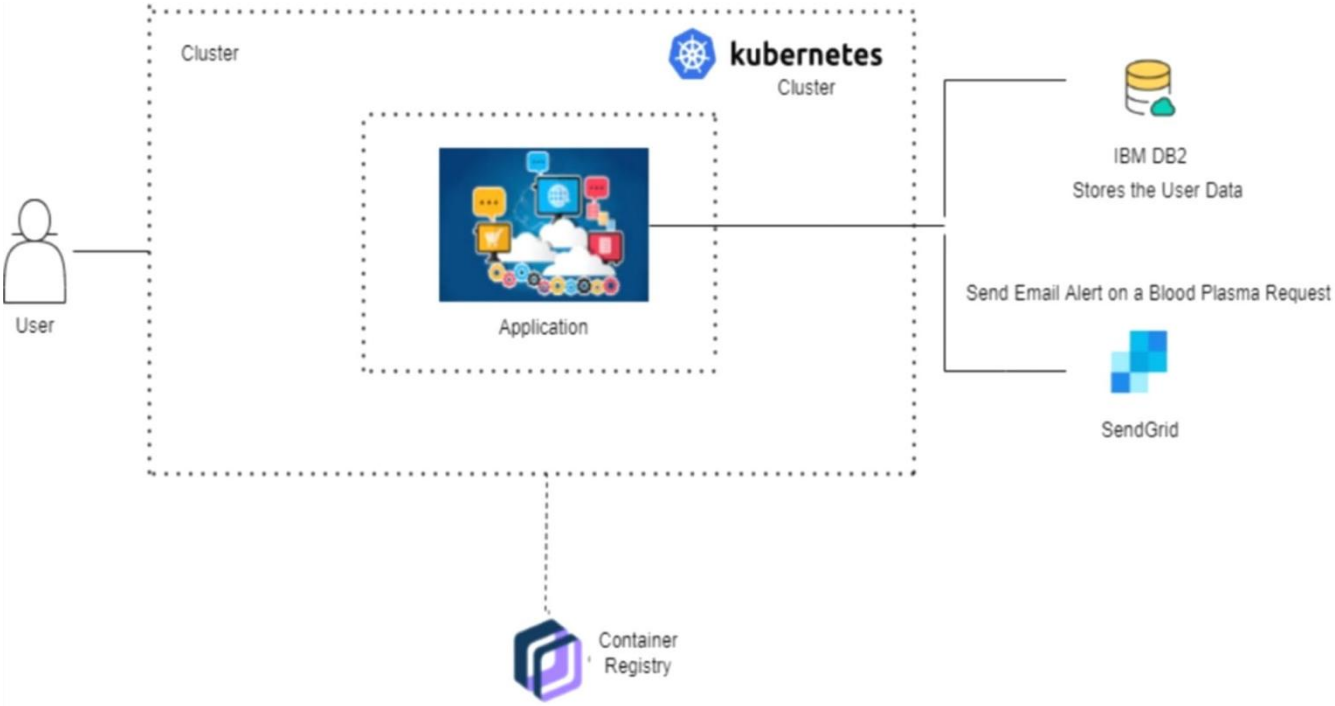
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application should be user-friendly and can useful to all patients/receivers
NFR-2	**Security**	The application should provide high security by enabling the user to login with username and password . The details of donors should protected safely and should be used only by the plasma receiver/patients.
NFR-3	**Reliability**	The application should provide high reliability by offering the plasma donors by precaution measures.
NFR-4	**Performance**	The system should be performed in all the situations.
NFR-5	**Availability**	The application should provide 24*7 services and available in all situation.
NFR-6	**Scalability**	The application should highly scalable and can store the details of various donors and the receiver can access the details of donors.

5. PROJECT DESIGN

Data Flow Diagrams



Solution & Technical Architecture



User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
User(Both plasma donor	User Registration	USN-1	The user can Register for the application by submitting	User can access their account /	High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
and recipient)			username,email,password and confirm password.	dashboard		
		USN-2	As a user, I will receive confirmation email once I have registered for the application	user can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through LinkedIn and other social media platform.	user can register & access the dashboard with LinkedIn and other social media Login	Low	Sprint-2

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	Login	USN-4	As a user, I can log into the application by entering email & password	User can access their account / dashboard	High	Sprint-1
	Dashboard					
Plasma Donor	Entering Donors Details	USN-5	The donor can enter the full details such as blood group, location, contact information etc...	Receive the information and store on donor database which can accessed by plasma	High	Sprint 1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
				receiver to contact donor		
	Receiving Notification	USN-6	The donor receives a notification when a receives send a request to the donor	Receiving Notification From recipient)	High	Sprint 1
Recipient (Plasma Receiver)	Checking for Donor	USN-7	Checking the availability of donor to contact them	Checking Donor	High	Sprint 1
	Sending request to donor.	USN-8	Recipient can send a request to the donor and can contact them directly through	Contact the donor	High	Sprint 1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
			contact information they provided.			
	Logout	USN-9	The user (both donor & receiver) can logout	Exit the application	High	Sprint 1

6. PROJECT PLANNING & SCHEDULING

Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story points	Priority	Team Members
Sprint 1	User Registration	USN-1	As a user, I can register for the application by entering my email, password, confirming my password and phone number..	10	High	Anand K Dhillkumar C Vishnu V
Sprint 1	User Login	USN-2	As a user, I can log into the application by entering username & password.	10	High	Anand K Sathish RK

Sprint-2	Virtual Certificate	USN-6	A user will get a virtual donor certificate after a verified successful plasma donation.	4	Medium	Sathish RK Dhillkumar CVishnu V
Sprint-2	Plasma Request	USN-7	A verified clinic is able to make a plasma request in the application	3	High	Anand K Sathish RK Vishnu V
Sprint-2	Verification ofDonor's details	USN-8	We the administrators will verify the details provided by the donors so only the genuine donors are able to use the application	2	Medium	Sathish RKVishnu V
Sprint-3	Accept thed	USN-9	A user and a registered donor will get a notification to accept the plasma request for their specific blood type.	3	High	Anand K Dhillkumar CVishnu V

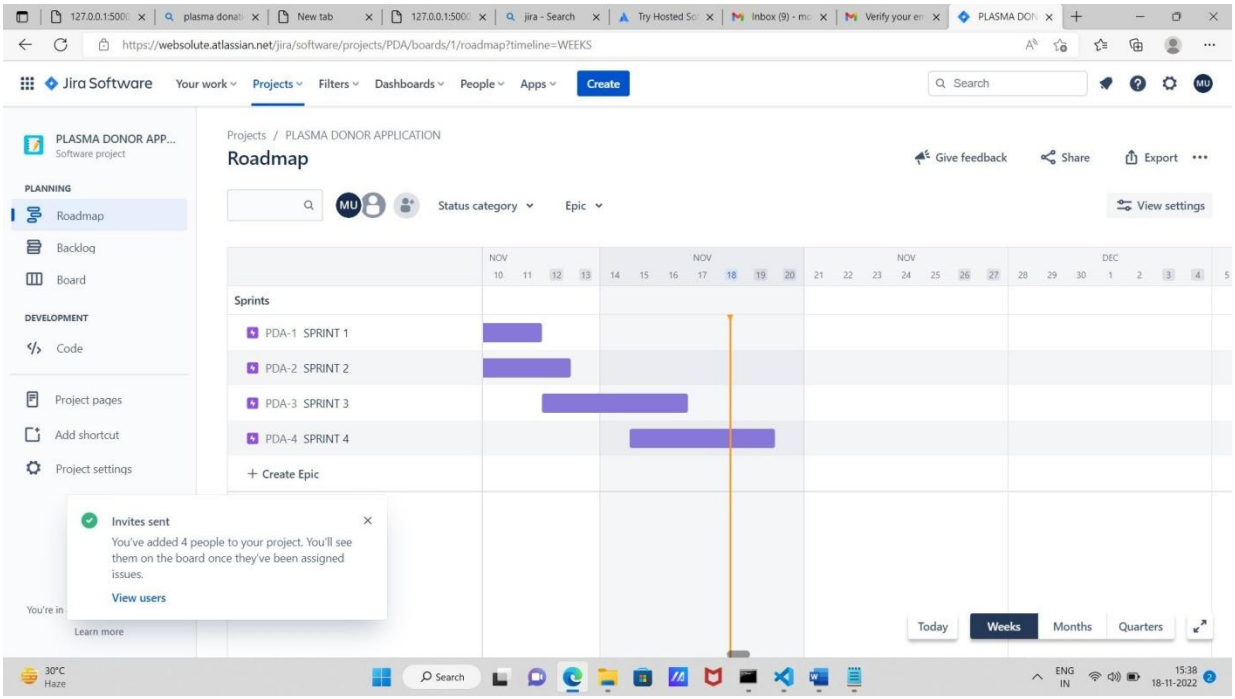
Sprint 3	View Donor Details	USN-8	The receiver can view the list of Donors of the blood type requested.	10	High	Moahamed Umar S Bava Rohith M Yugesh R
Sprint 4	Logout Process	USN-9	The User will be able to Logout of the application.	10	High	Moahamed Umar S Ajay Manikandan S Bava Rohith M
Sprint 4	Bot service in the website	USN-10	The user can use Bot Service to request for Blood Plasma and also switch between roles.	10	High	Moahamed Umar S Ajay Manikandan S Bava Rohith M
Sprint 3	Verified Donor	USN-7	As a donor, I can request for verified account by providing the required documents and details to the admin through the web application.	8	Medium	Yugesh R Vilva Praveen S
Sprint 4	Update the profile	USN-8	As a donor, I can update my profile at any time.	8	Medium	Ajay Manikandan S Vilva Praveen S Yugesh R

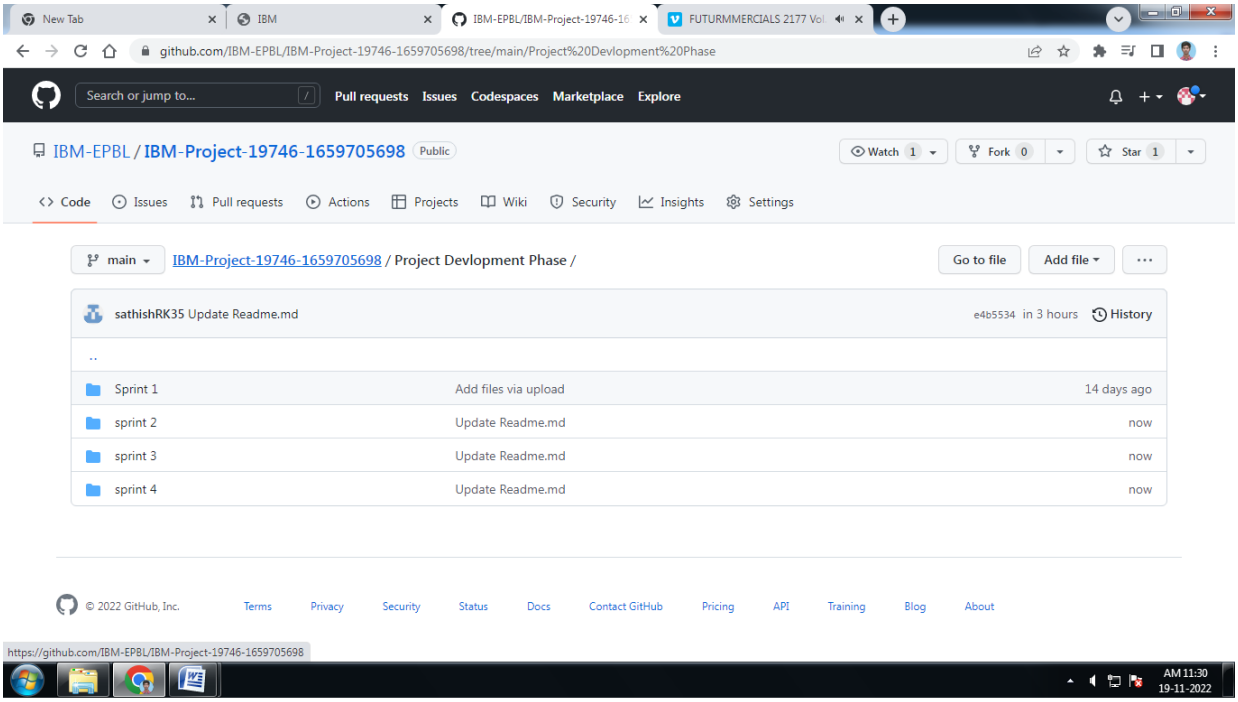
Sprint 4	**Feedback**	**USN-9**	**As a user, I can give the feedback to the Donor.**	**7**	**Low**	**Yugesh R** **Ajay Manikandan S** **Vilva Praveen S**
Sprint 4	**Reward**	**USN-9**	**The donor who donated plasma will receive E-certificate as a reward**	**7**	**Low**	**Yugesh R** **Vilva Praveen S** **Mohamed umar S**

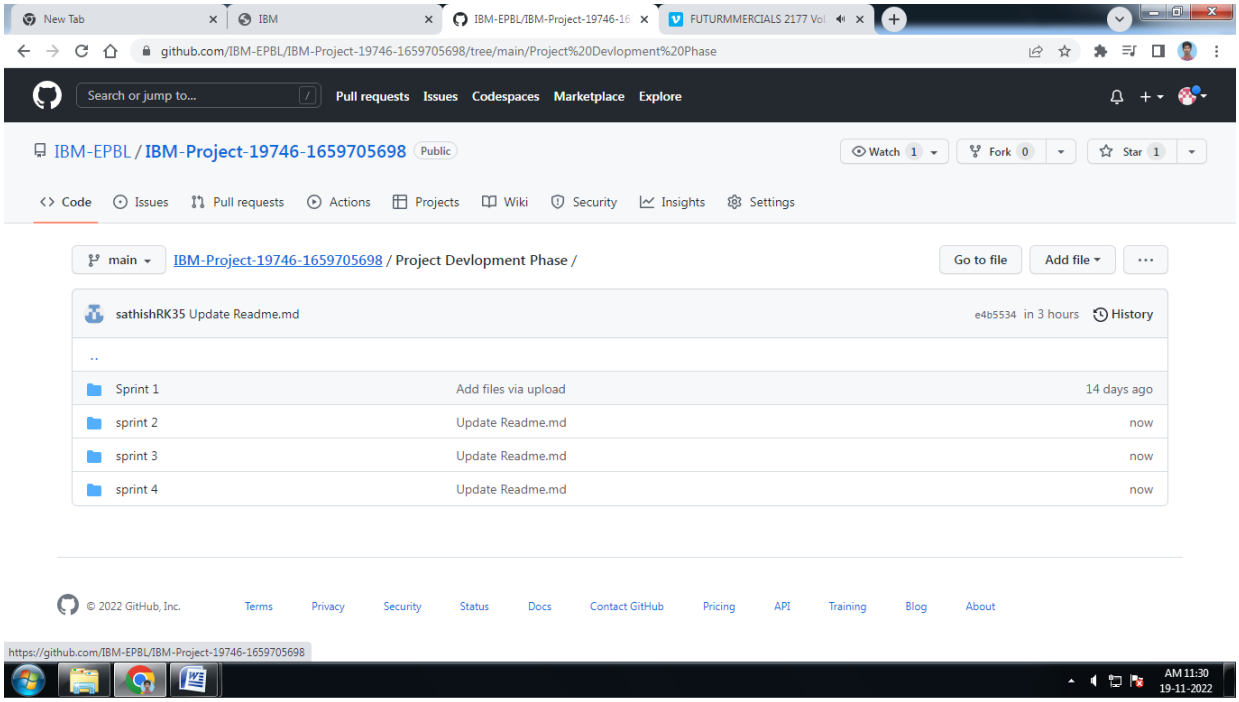
Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	30	8 Days	24 Oct 2022	31 Oct 2022	30	31 Oct 2022
Sprint-2	50	8 Days	3 Nov 2022	10 Nov 2022	50	10 Nov 2022
Sprint-3	30	8 Days	07 Nov 2022	14 Nov 2022	30	14 Nov 2022
Sprint-4	20	8 Days	12 Nov 2022	19 Nov 2022	20	19 Nov 2022

Reports from JIRA






New Tab
IBM
IBM-EPBL/IBM-Project-19746-16
FUTURMMERCIALS 2177 Vol
github.com/IBM-EPBL/IBM-Project-19746-1659705698/tree/main/Project%20Devlopment%20Phase
Search or jump to...
Pull requests
Issues
Codespaces
Marketplace
Explore
IBM-EPBL / IBM-Project-19746-1659705698
Public
Watch 1
Fork 0
Star 1
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
main
IBM-Project-19746-1659705698 / Project Devlopment Phase /
Go to file
Add file
sathishRK35 Update Readme.md
e4b5534 in 3 hours
History
..
Sprint 1
Add files via upload
14 days ago
sprint 2
Update Readme.md
now
sprint 3
Update Readme.md
now
sprint 4
Update Readme.md
now
© 2022 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About
https://github.com/IBM-EPBL/IBM-Project-19746-1659705698
AM 11:30
19-11-2022

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

Feature 1

PYTHON

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0.

Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support.

Python 3.0, released in 2008, was a major revision that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.

Python is freely available for everyone. It has a large community across the world that is dedicatedly working towards make new python modules and functions.

It can be easily integrated with languages like C, C++, and JAVA and FRONT-END languages such as HTML,CSS,javascript and also deals with Databases. Python runs code line by line like C,C++ Java. It makes easy to debug the code.

The code of the other programming language can use in the Python source code. We can use Python source code in another programming language as well. It can embed other language into our code.

Python supports object-oriented language and concepts of classes and objects come into existence.

It supports inheritance, polymorphism, and encapsulation, etc. The object-oriented procedure helps to programmer to write reusable code and develop applications in less code.

FLASK

- Flasks design is lightweight and modular. Therefore, it is easy to transform it into the web applications or framework when one needs very few extensions without weighing much.

- Flask is ORM-agnostic: i.e. user can plug in their favorite ORM like SqlAlchemy and The basic foundation of API is very nicely shaped and made coherent.

- Documentation of flask is very comprehensive, filled with lots of examples and are well structured. Users can even try out some sample applications to really get the real feel of Flask.

- It is very easy to deploy Flask in production as Flask comes with 100% WSGI 1.0 compliant

- Flask can handle HTTP request easily with help of its functionalities

- It is highly flexible. Its configuration is even more flexible than that of Django, which gives its users plenty of solutions for every product they need.

Feature 2

BOOTSTRAP:

- Bootstrap is a powerful, feature-packed frontend toolkit.

- Bootstrap is a Web design front-end framework Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and JavaScript-based design templates

- It is pretty easy, to begin with. Being easy to get started with is probably the first quality which makes Bootstrap very appealing.

- Another prominent feature of Bootstrap is its responsive utility classes. Using responsive utility classes, a particular piece of content can be made to appear or hide only on

- devices depending on the size of the screen being used. This feature is extremely helpful for designers who want to make a mobile and tablet-friendly version of their websites.

- Bootstrap is an HTML, CSS and JS Library that focuses on simplifying the development of informative web pages (as opposed to web apps).

- The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project.

- Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers.

- In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents

- Bootstrap is becoming the world’s favorite front-end component library. Using Bootstrap, we can easily build responsive, mobile-first projects on the web. You can quickly prototype your unique ideas.

- You can build an entire application using their Sass variables, powerful plugins, extremely responsive grid system, and a lot more.

Database Schema

IBM DB2

- DB2 is a database product from IBM.
- It is a Relational Database Management System (RDBMS). DB2 is designed to store, analyze and retrieve the data efficiently.
- DB2 product is extended with the support of Object-Oriented features and non-relational structures with XML.
- Provide a massively parallel processing (MPP) architecture Exploits Hive, HBase and Apache Spark concurrently for best-in-class analytic capabilities.
- Provides low latency support for ad-hoc and complex queries, high performance, and federation capabilities Understands dialects from other vendors and various products from Oracle, IBM® Db2® and IBM Netezza® Enables advanced row and column security

Kubernates

- **Kubernetes** is also known as **'k8s'. Kubernetes** is an extensible, portable, and open-source platform designed by **Google** in **2014**.

- It is mainly used to automate the deployment, scaling, and operations of the container-based applications across the cluster of nodes.

- Kubernetes helps to manage containerised applications in various types of physical, virtual, and cloud environments.

- Google Kubernetes is a highly flexible container tool to consistently deliver complex applications running on clusters of hundreds to thousands of individual servers

- Kubernetes is the Linux kernel which is used for distributed systems.

- It helps you to be abstract the underlying hardware of the nodes(servers) and offers a consistent interface for applications that consume the shared pool of resources.

8. TESTING

Test Cases

- It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectation and does not fail in an unacceptable manner.

- There are various types of test. Each test type addresses a specific testing requirement .

1.Purpose of Document

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

2. Defect Analysis

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

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

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

3.Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	8	0	0	8
Client Application	50	0	0	50
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	10	0	0	10
Final Report Output	6	0	0	6
Version Control	3	0	0	3

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

User Acceptance Testing

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Login popup when user clicked on Login button		1.Enter URL and click go 2.Click on Login button 3.Verify login popup displayed or not		Login page popup should display	Working as expected	Pass	
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements in Login popup		1.Enter URL and click go 2.Click on Login button 3.Verify login popup with below UI elements: a.username text box b.password text box c.Login button d.Dont have an account.signup		Application should show below UI elements: a.username text box b.password text box c.Login button. d.Dont have an account . Sign up.	Working as expected	Pass	Recover Password Feature not yet added
LoginPage_TC_OO3	Functional	Login page	Verify user is able to log into application with Valid credentials		1.Enter URL and click go 2.Click on Login button 3.Enter Valid username text box 4.Enter valid password in password text box 5.Click on login button	Username:monsumar123@gmail.com password: Mdumar@12#	User should navigate to user account dashboard	Working as expected	Pass	
LoginPage_TC_OO4	Functional	Login page	Verify user is not able to log into application with InValid credentials		1.Enter URL and click go 2.Click on Log in button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username:monsumar123@gmail.com password: Testing#234	Application should show 'Invalid username or password ' validation message.	Working as expected	Pass	
Register_TC_OO5	Functional	Register Page	Verify the UI elements in Signup popup		1.Enter URL and click go 2.Click on Signup button 3.Verify Singup popup with below UI elements: a.username b.email text box c.password text box d.repassword sign up Button	Username: mohamed umar password: Testing123 Email :abc@gmail.com Password:Mdumar23245#	Application should show below UI elements: a.Name b.email text box c.password text box d.repassword Sign up Button	Working as expected	Pass	
Register_TC_OO6	Functional	Register Page	Verify that New User is able to register and create his own account.		1.Enter URL and click go 2.Click on Login/Signup button 3.Fill the required fills mentioned below: a.Name b.email text box c.password text box d.repassword Sign up Button	Username: mohamed umar password: Testing123 Email :abc@gmail.com Password:Mdumar23245#	Application must redirect to proper webpage(signinpage) after verifying the details and ensuring that all the fields are filled	Working as expected	Pass	
Register_TC_OO7	Functional	Register Page	Verify that New User when registering with existing data		1.Enter URL and click go 2.Click on Login/Signup button 3.Fill the required fills mentioned below: a.Name b.email text box c.password text box d.repassword Sign up Button	Username: mohamed umar password: Testing123 Email :abc@gmail.com Password:Mdumar23245#	Application must redirect to the same page with prompts saying that username already taken.	Working as expected	Pass	
Main_TC_OO8	Functional	Dashboard	Verify that New User Can select the role he wants to be as.	Successfull Login/Register	1.After successfully login go to main page 2. Click on the button to select the role that you want 3.Select User or Donor accourding to your requirement	Donordashboard and Receivcerdashboard	Application must direct to concern dashboard	Working as expected	Pass	
Main_TC_OO9	UI	Donor Dashboard	Verify that User Can view Request list from receiver	Successfull Login/Register Select Role as donor	Click Donordashboard in main dashboard		Application must show request from donor and other options such as a.Add me donor b.Update profile c.verify donor d.rewards	Working as expected	Pass	

Main_TC_OO10	UI	Donor Dashboard	Verify that User Can add them as donor	login to donordashboard and clock on add me donor		Fill all the details	Application must display all UI buttons to register them as a donor	Working as expected	Pass	
Main_TC_OO11	UI	Donor dashboard	Verify that User Can update or verify profile	login to donordashboard and clock on update profile or verify donor		Fill all the details	Application must display all UI buttons to update or verify user details	Working as expected	Pass	
Main_TC_OO12	Functional	Donor dashboard	User can View their e-certificate	login to donordashboard and clock on rewards	1.After successfully login go to main page 2.select role as donor 3. click on on rewards		Application must display e-certificate	Working as expected	Pass	
Main_TC_OO13	Functional	Donor Dashboard	User can add them as donor ,update profilr and verify them as verified donor	Successfull Login/Register Select Role as donor		Fill all the details	Application must save the data in database	Working as expected	Pass	
Main_TC_OO14	Functional	receiver dashboard	user can view receiver dashboard	selecting role as a receiver	after selecting role as a receiver		application must view all the options in receiver dashboard such as switch role , request plasma , search donor , feedback	Working as expected	Pass	
Main_TC_OO15	UI	receiver dashboard	user can view request for plasma	selecting request plasma in receiver dashboard		Fill all the details	Application must save the data in database and view to donor	Working as expected	Pass	Request not send to donor email . Will be updated in future
Main_TC_OO16	Functional	receiver dashboard	user can switch roloe as donor				Application should redirect to donor dashboard	Working as expected	Pass	
Main_TC_OO17	Functional	receiver dashboard	user can send feedback		Select the user and send feedback	Entering feedback	application should saave data and store in database	Working as expected	Pass	
Main_TC_OO18	UI	receiver dashboard	user can view donor list				Application must show donor list	Working as expected	Pass	
Main_TC_OO19	Functional		After clicking on logout button available in navbar in all pages				Application should redirect to home page	Working as expected	Pass	

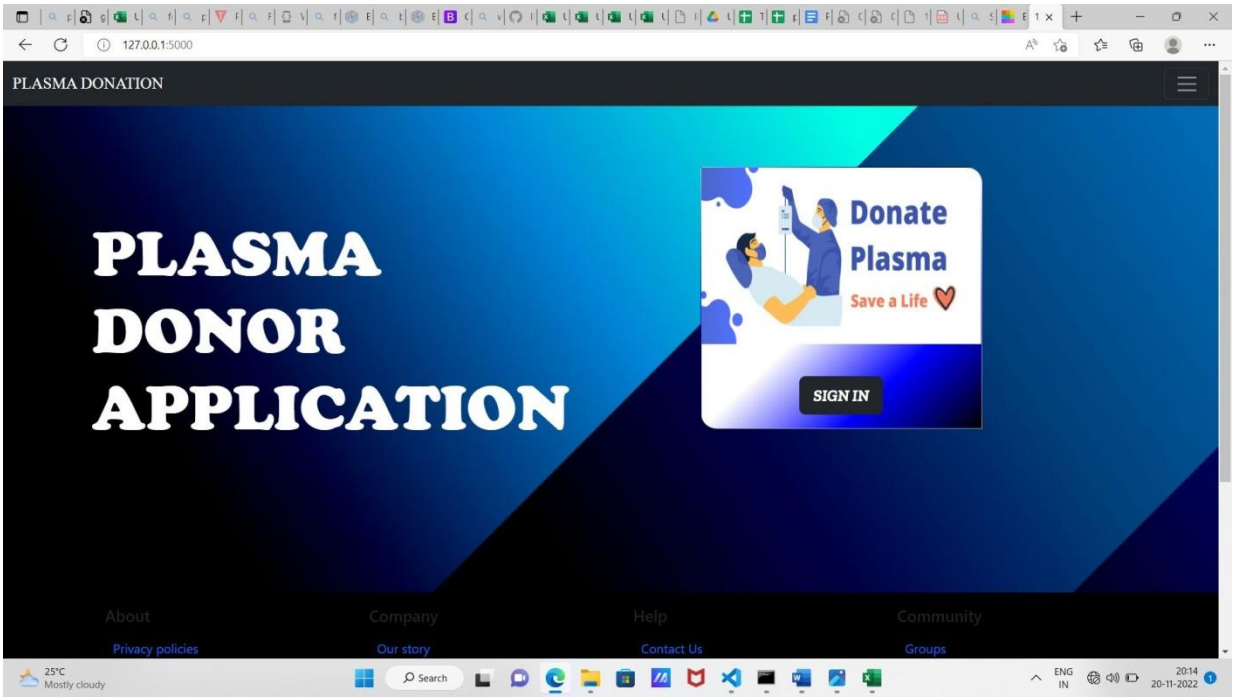
9. RESULTS

Performance Metrics

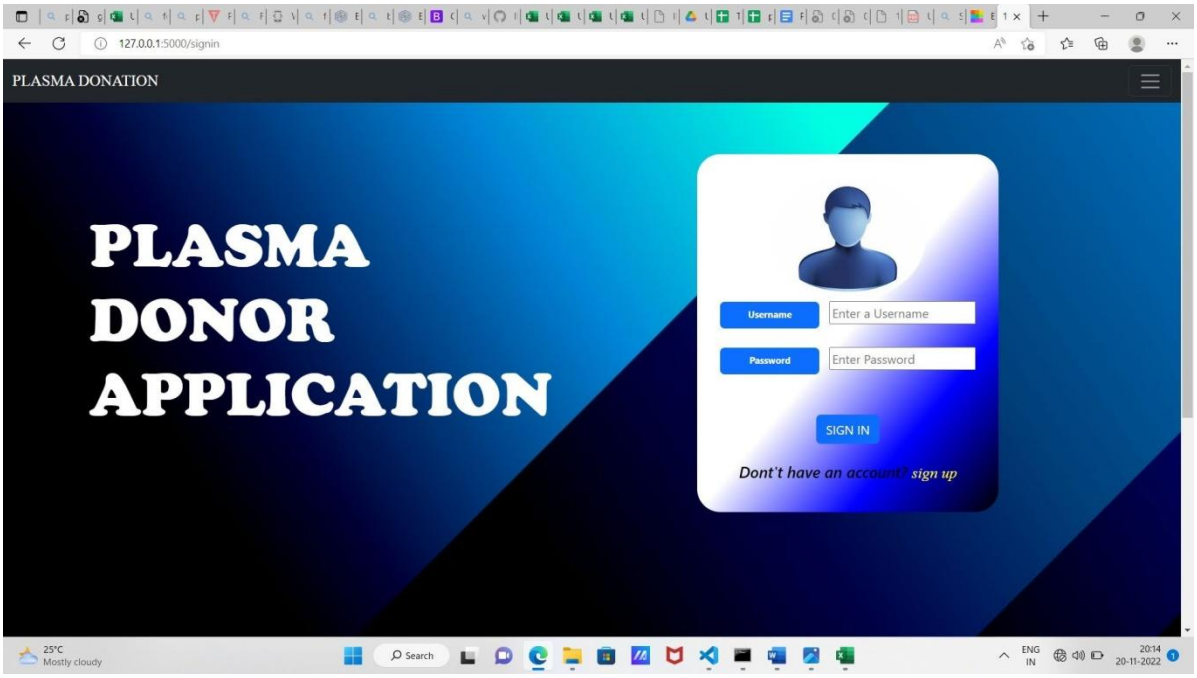
- Project metrics are used to track the progress and performance of a project.
- Monitoring parts of a project like productivity, scheduling, and scope make it easier for team leaders to see what's on track.
- As a project evolves, managers need access to changing
- deadlines or budgets to meet their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

OUTPUTS :

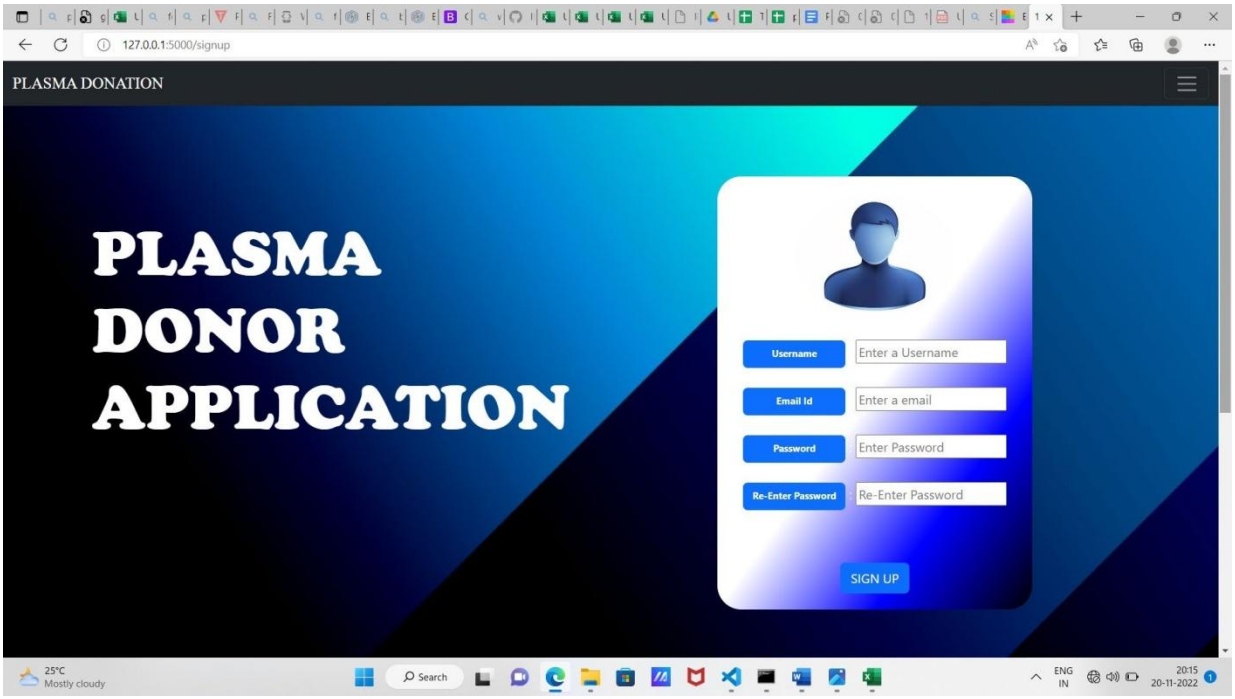
Home page



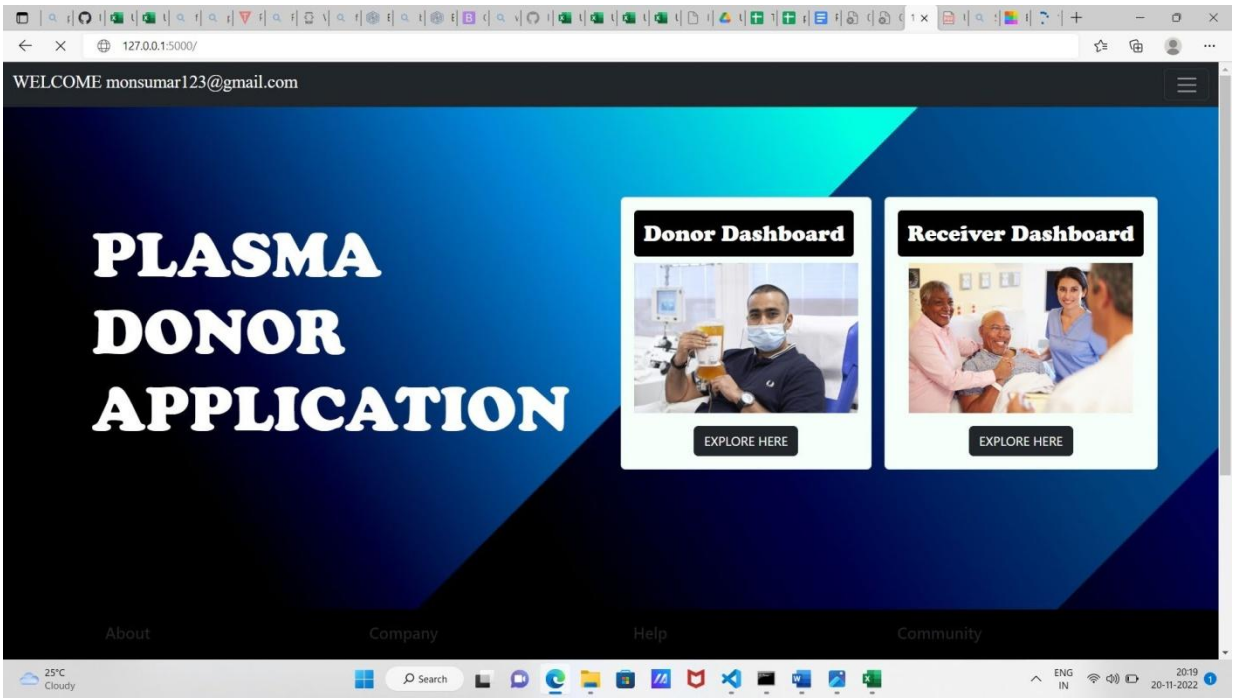
Signin page



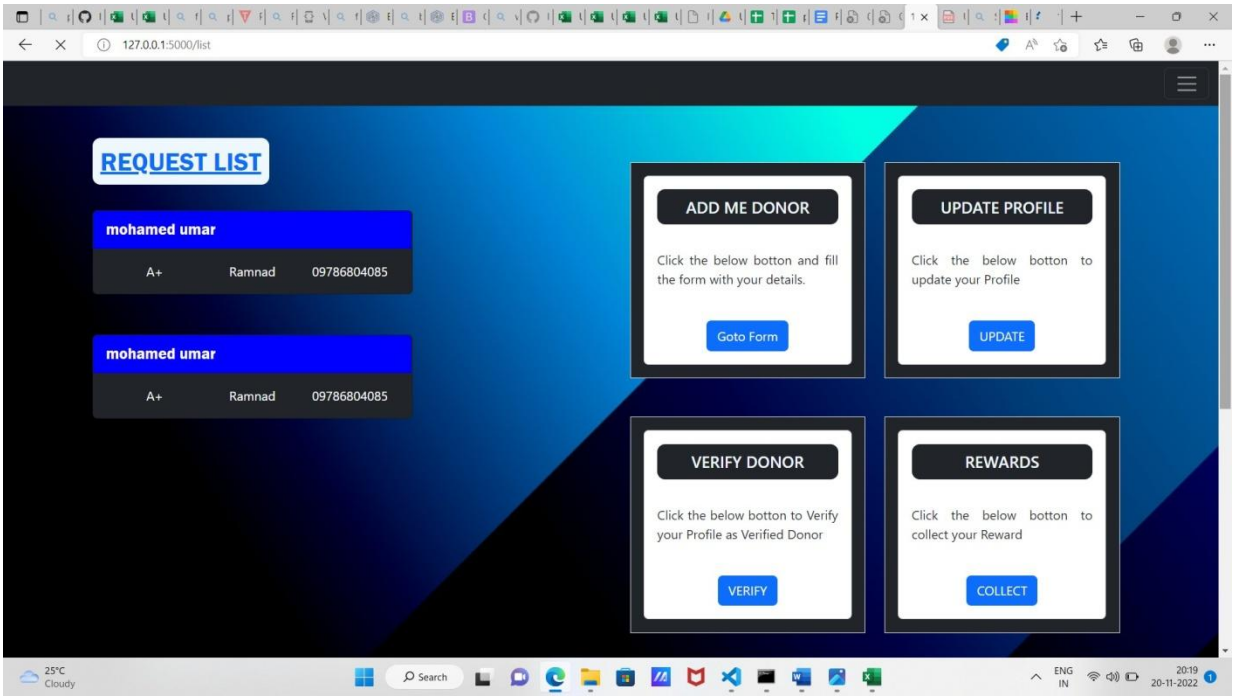
Signup page



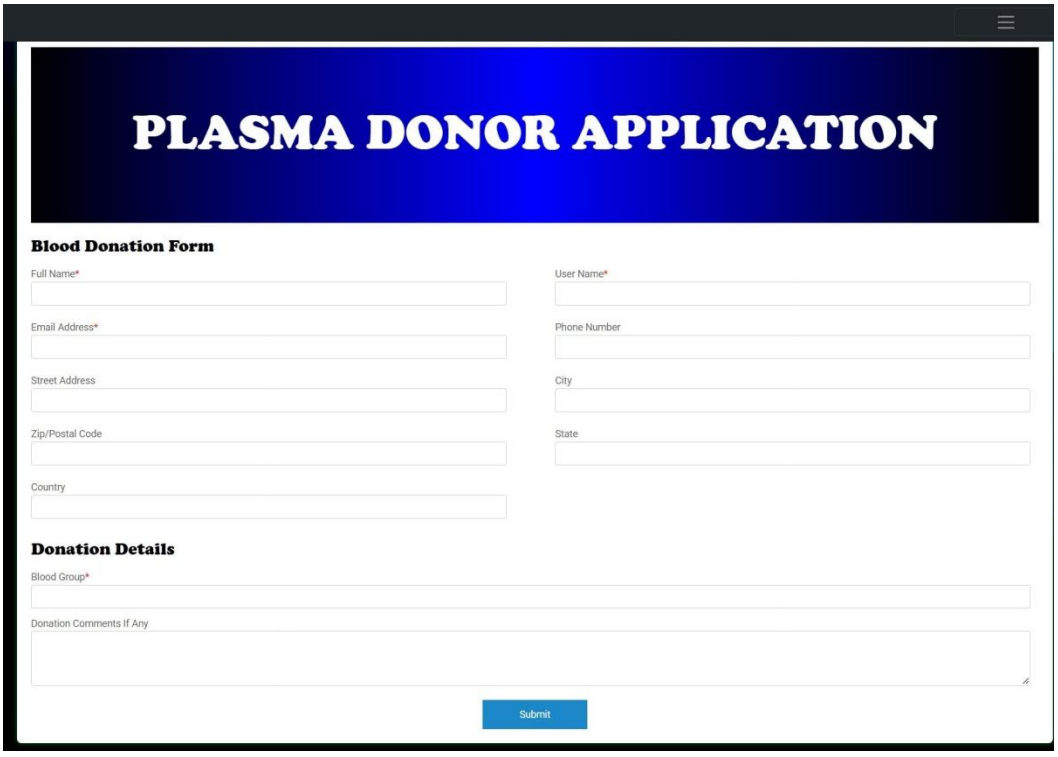
Dashboard



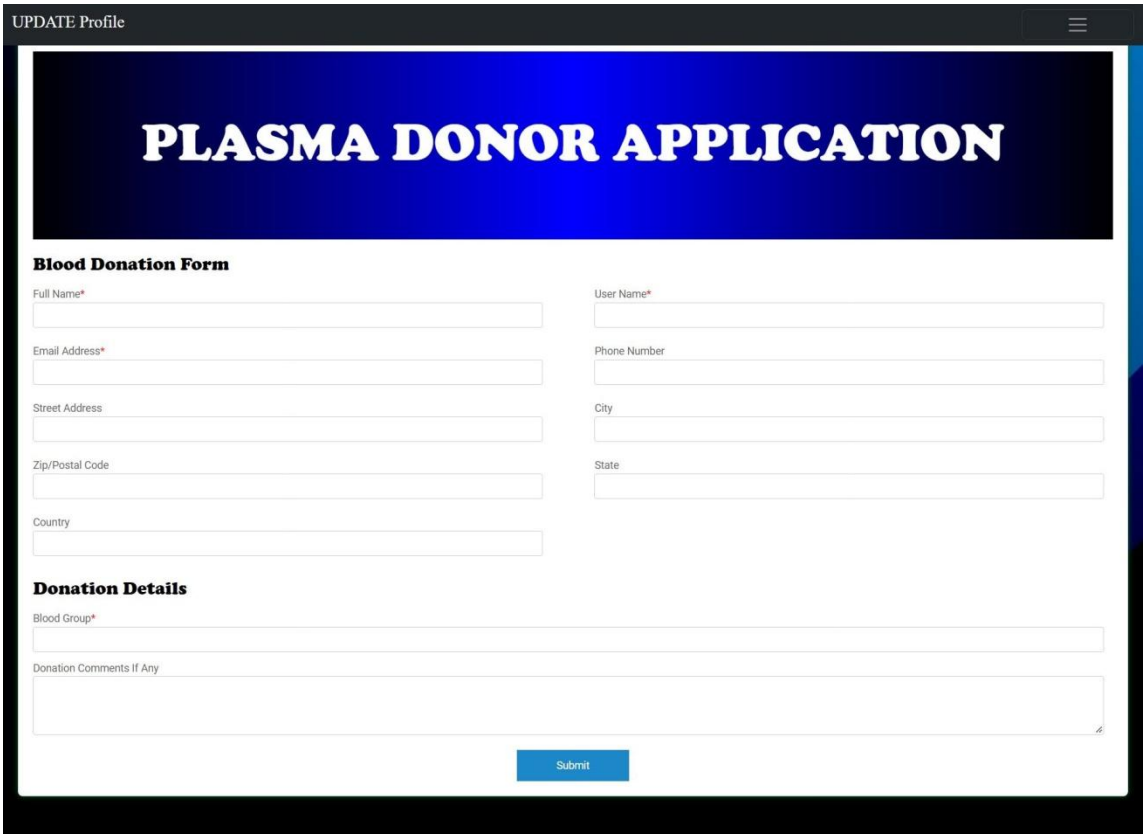
Donor dashboard



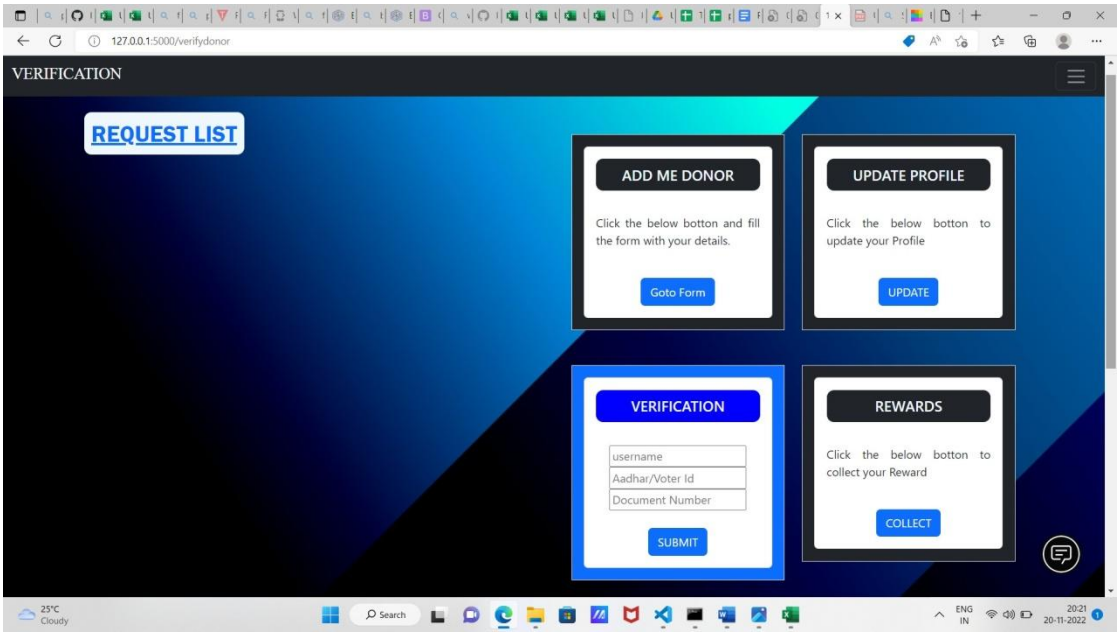
Add me donor



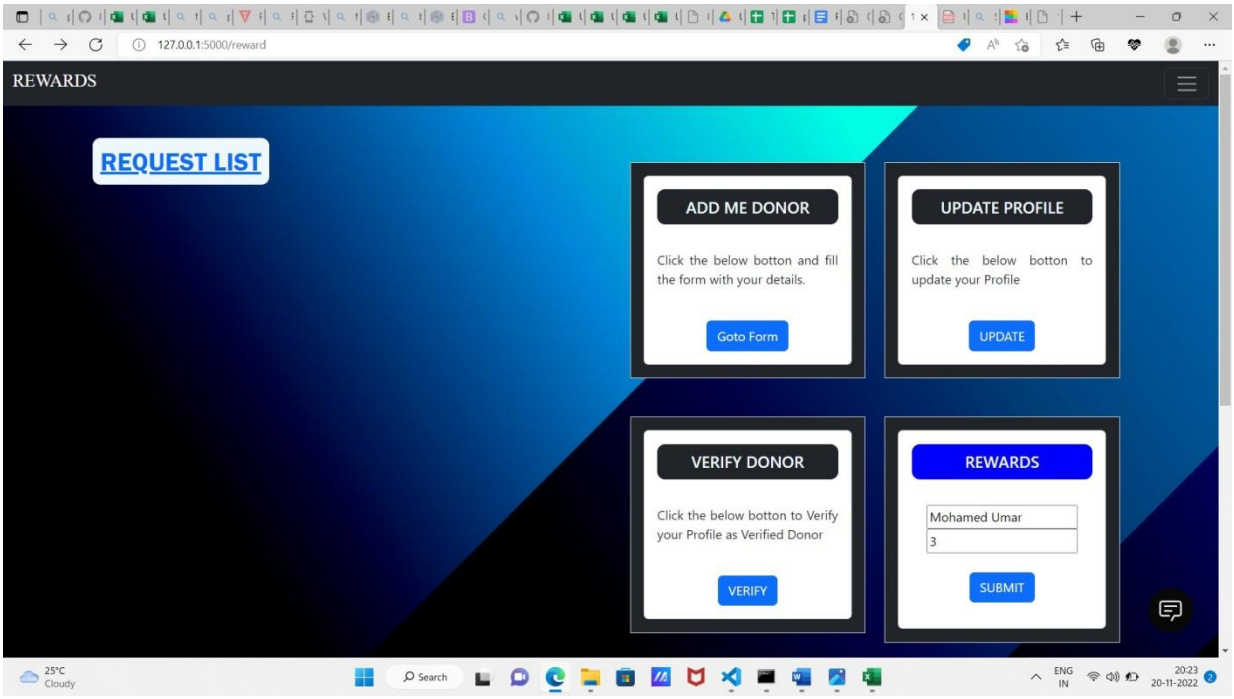
Update Profile



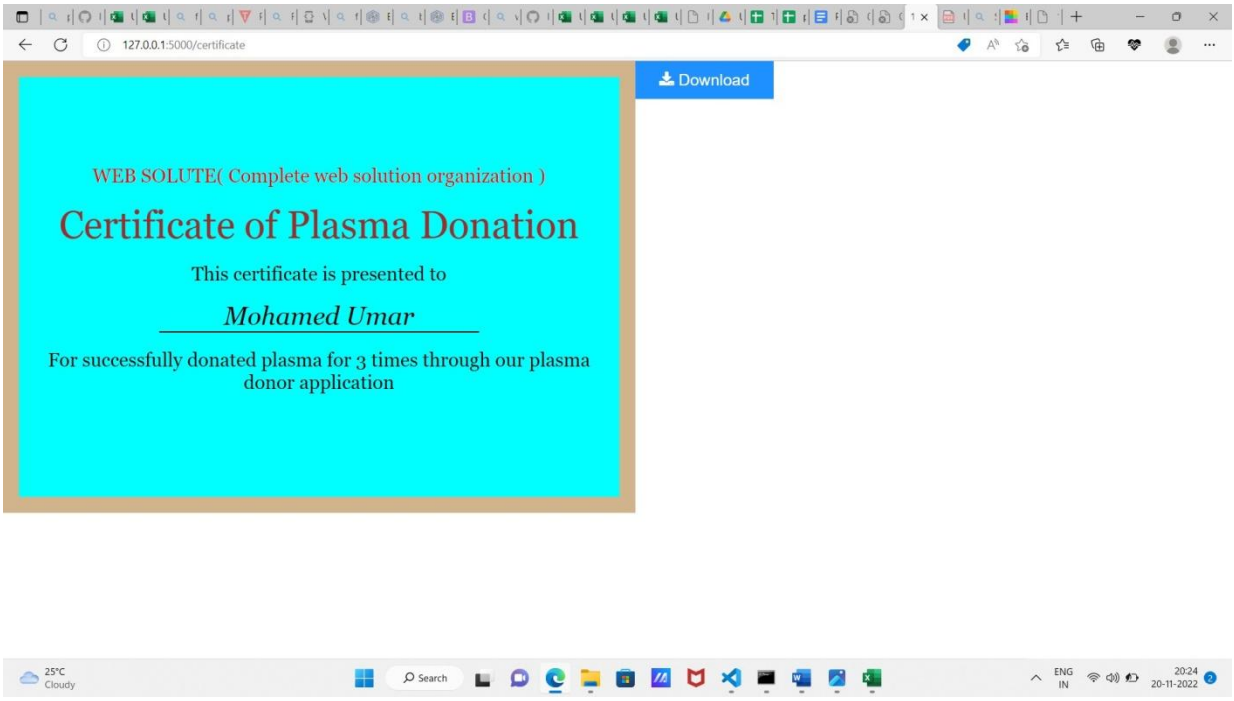
Verify donor



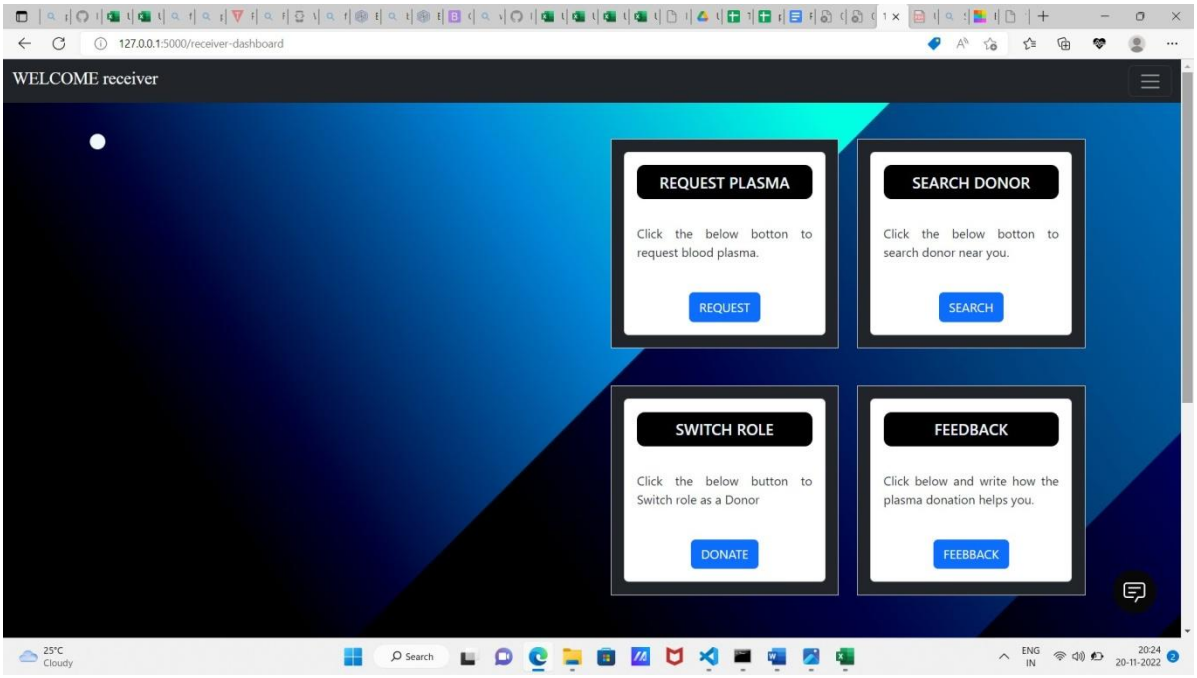
Reward



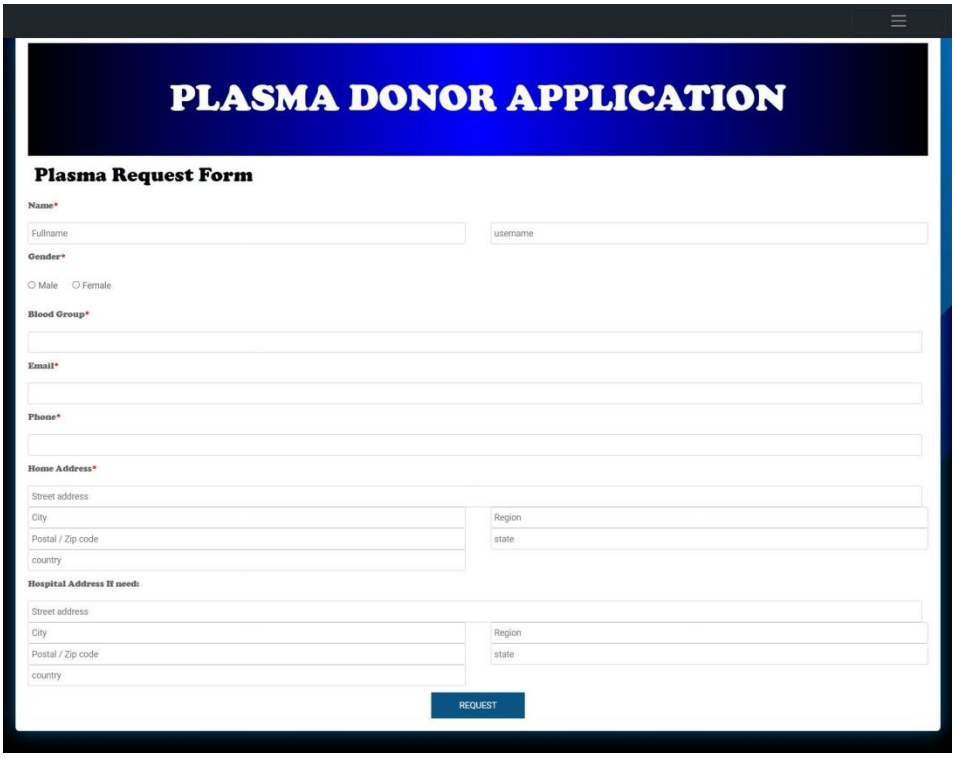
Certificate



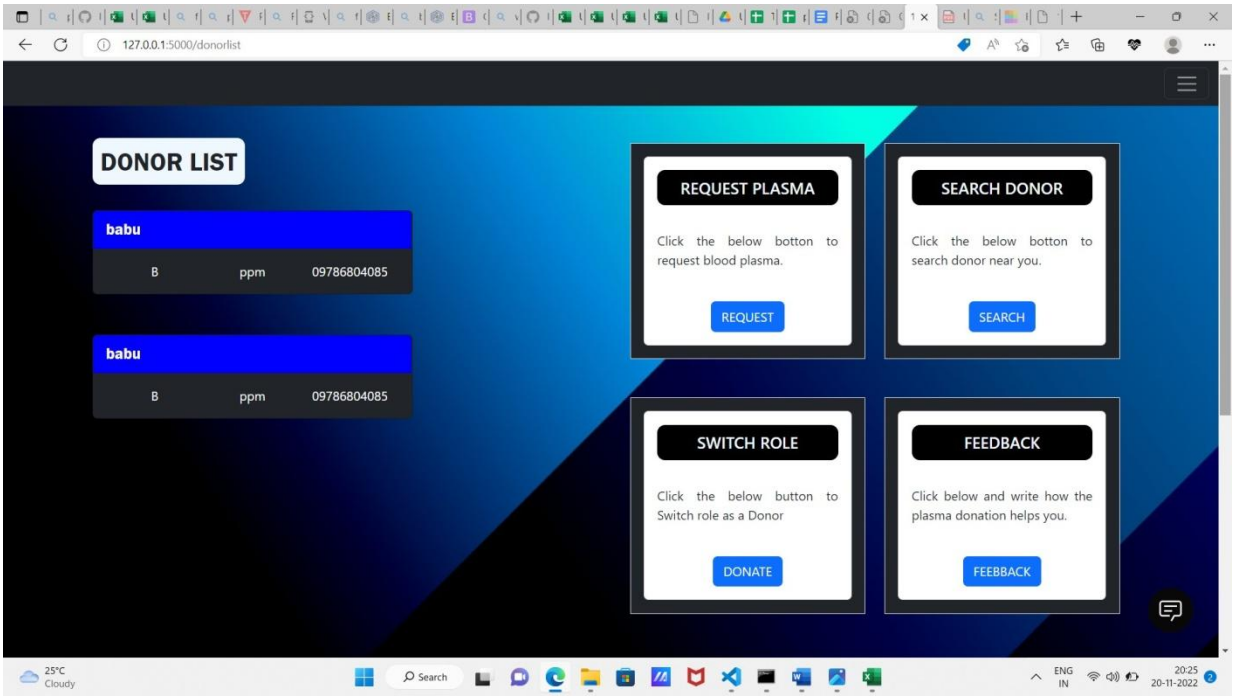
Receiver dashboard



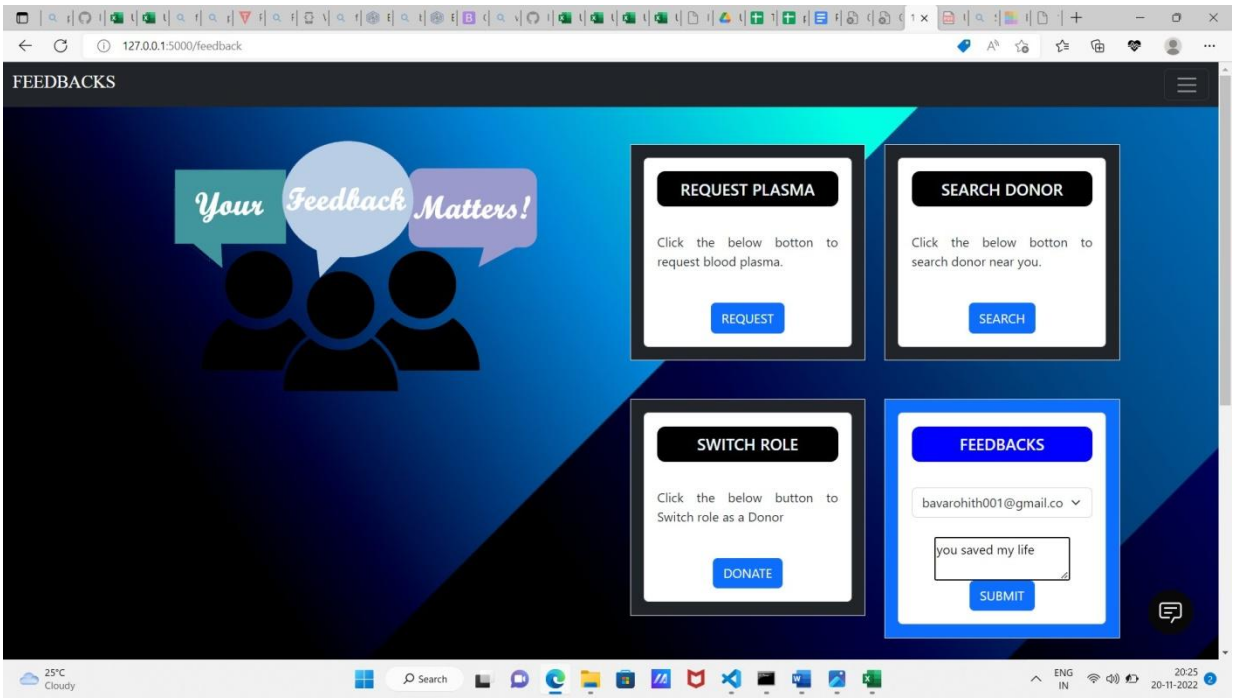
Request Plasma



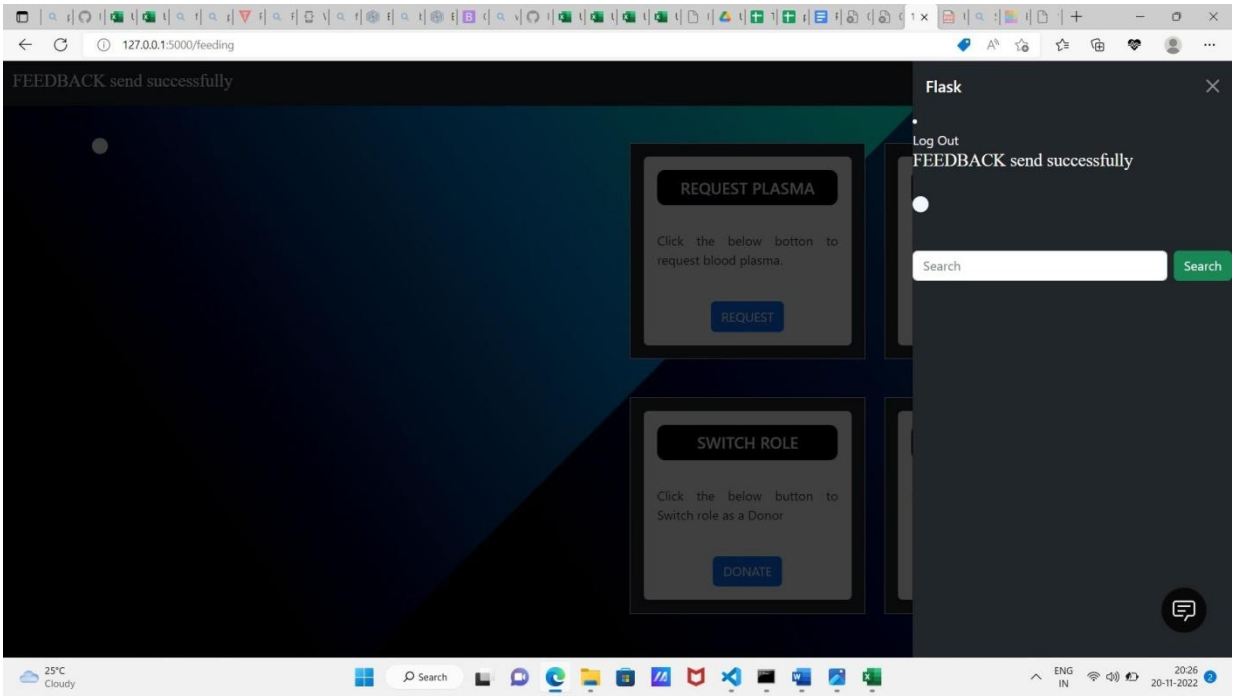
Search donor



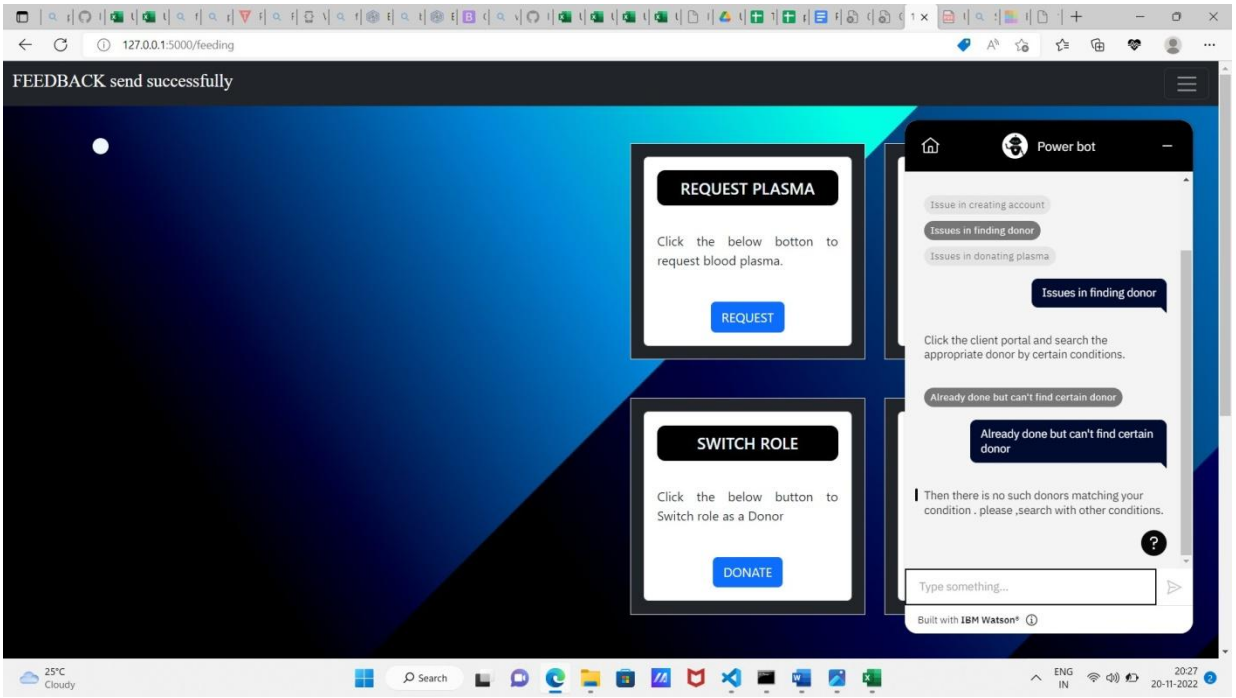
Feedback



Logout



Bot Service



10. ADVANTAGES & DISADVANTAGES

Advantages:

- This website is fast and offers great accuracy as compared to manual registered keeping.
- Less maintenance is required
- It is very easy to use and understand. It is easily workable and accessible for everyone.
- It would help you to provide plasma donors easily depending upon the availability of it.
- Searching for plasma donors made eassier through plasma donor application.
- Bringing all of the donors and receipients under one umbrella.

Disadvantages

- Internet is Mandatory to use the Application.
- Reports of donors are not verified Automatically.

11. CONCLUSION

- The efficient way of finding plasma door for the infected people is implemented using the plasma donor website that is hosted on IBM Cloud platform.

- To ensure the smooth functioning of the web site operation. I have hosted the website in IBM Db2 & Kubernates Cluster to make sure the operations are running successfully Cloud lambda function is used and to deploy the application IBM Db2 service is used.

- Other services such as IBM objectstorage for storing data in cloud and IBM Watson for creating Bot service in Website .

12. FUTURE SCOPE

- Sending Emails for donors when request for plasma.
- Upgrading the UI that is more user-friendly which will help many users to access the website and also ensures that many plasma donors can be added into the community.
- Advertising to add more donors.
- Integrating with other Applications.
- Social media advertisement and Integration.

13. APPENDIX

Source Code :

FLASK APPLICATION (app.py)

```
from turtle import st
from flask import Flask, render_template, request,redirect, url_for,
session,flash
from markupsafe import escape

app = Flask(__name__)

import ibm_db

hostname = "764264db-9824-4b7c-82df-
40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud"
uid = "qph68689"
pwd = "igr5nkXqgfIHqv7r"
driver = "{IBM DB2 ODBC DRIVER}"
db = "bludb"
port = "32536"
protocol = "TCPIP"
cert = "abc.crt"

dsn = (
```

```
    "DATABASE={0};"
    "HOSTNAME={1};"
    "PORT={2};"
    "UID={3};"
    "SECURITY=SSL;"
    "SSLServerCertificate={4};"
    "PWD={5};"
).format(db, hostname, port, uid, cert, pwd)
print(dsn)
try:
    conn = ibm_db.connect(dsn, "", "")
    print("Connected to data base")
except:
    print("Unable to connect", ibm_db.conn_errormsg())


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


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

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


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

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




@app.route('/request_1')
def request_1():
    return render_template('request.html')


@app.route('/dashboard')
def welcome():
    return render_template('dashboard.html')
```

```
@app.route('/verifydonor')
def verifydonor():
    return render_template('verify.html',msg4="VERIFICATION")

@app.route('/donor-dashboard')
def donor():
    return render_template('donordashboard.html',msg="WELCOME
Donor")

@app.route('/receiver-dashboard')
def receiver():
    return render_template('receiverdashboard.html',msg="WELCOME
receiver",msg1="PLASMA DONOR APPLICATION")

@app.route('/updateprofile')
def updateprofile():
    return render_template('updateprofile.html',msg="UPDATE Profile")

@app.route('/reward')
def reward():
    return render_template('reward.html',msg4="REWARDS")
```

```
@app.route('/log-in', methods=['POST', 'GET'])
def log_in():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        sql = "select * from user where username=? and password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.bind_param(stmt, 2, password)
        ibm_db.execute(stmt)
        dic = ibm_db.fetch_assoc(stmt)
        print(dic)
        if dic:
            msg = username
            return render_template('dashboard.html', msg=msg)
        else:
            msg = "Invalid Username or Password"
            return render_template('signin.html', msg=msg)

    elif request.method == 'GET':
        msg = "Invalid Username or Password"
        return render_template('signin.html', msg=msg)


@app.route('/sign-up', methods=['POST', 'GET'])
```

```
def sign_up():
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        repassword = request.form['repassword']
        sql = "SELECT * FROM user WHERE username =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        if account:
            return render_template('signup.html', msg="Username Already
Taken")


        if request.form['password'] == request.form['repassword']:


            sql = "insert into user(username,email,password,repassword)
values(?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, sql)
            ibm_db.bind_param(prep_stmt, 1, username)
            ibm_db.bind_param(prep_stmt, 2, email)
            ibm_db.bind_param(prep_stmt, 3, password)
```

```
            ibm_db.bind_param(prep_stmt, 4, repassword)
            ibm_db.execute(prep_stmt)
            return render_template('signin.html ', msg="Account Created
Successfully.")
        else:
            return render_template('signup.html ', msg="Please enter password
and repassword correctly")

    elif request.method == 'GET':
        return render_template('signup.html')



@app.route('/addrec', methods=['POST', 'GET'])
def addrec():
    if request.method == 'POST':
        name = request.form['name']
        username = request.form['username']
        email = request.form['email']
        phone = request.form['phone']
        street = request.form['street']
        city = request.form['city']
        pin = request.form['pin']
        state = request.form['state']
        country = request.form['country']
```

```
        bloodgroup = request.form['bloodgroup']
        command = request.form['command']
        sql = "SELECT * FROM donor WHERE username =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        if account:
            return render_template('donordashboard.html', msg2="You are
already registered as a donor")
        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, username)
            ibm_db.bind_param(prep_stmt, 3,email )
            ibm_db.bind_param(prep_stmt, 4, phone)
            ibm_db.bind_param(prep_stmt, 5,street )
            ibm_db.bind_param(prep_stmt, 6, city)
            ibm_db.bind_param(prep_stmt, 7, pin)
            ibm_db.bind_param(prep_stmt, 8,state)
            ibm_db.bind_param(prep_stmt, 9,country )
            ibm_db.bind_param(prep_stmt, 10,bloodgroup )
            ibm_db.bind_param(prep_stmt, 11,command )
```

```
            ibm_db.execute(prep_stmt)

        return render_template('donordashboard.html', msg2="Your Data
saved successfuly..")


@app.route('/update', methods=['POST', 'GET'])
def update():
    if request.method == 'POST':
        name = request.form['name']
        username = request.form['username']
        email = request.form['email']
        phone = request.form['phone']
        street = request.form['street']
        city = request.form['city']
        pin = request.form['pin']
        state = request.form['state']
        country = request.form['country']
        bloodgroup = request.form['bloodgroup']
        command = request.form['command']
        sql = "SELECT * FROM donor WHERE username =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
```

```
        if account:
            delete_sql="DELETE FROM donor WHERE username=?"
            stmt = ibm_db.prepare(conn, delete_sql)
            ibm_db.bind_param(stmt, 1, username)
            ibm_db.execute(stmt)

            insert_sql = "INSERT INTO donor VALUES (?,?,?,?,?,?,?,?,?,?,?);"

            prep_stmt1 = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt1, 1, name)
            ibm_db.bind_param(prep_stmt1, 2, username)
            ibm_db.bind_param(prep_stmt1, 3,email )
            ibm_db.bind_param(prep_stmt1, 4, phone)
            ibm_db.bind_param(prep_stmt1, 5,street )
            ibm_db.bind_param(prep_stmt1, 6, city)
            ibm_db.bind_param(prep_stmt1, 7, pin)
            ibm_db.bind_param(prep_stmt1, 8,state)
            ibm_db.bind_param(prep_stmt1, 9,country )
            ibm_db.bind_param(prep_stmt1, 10,bloodgroup )
            ibm_db.bind_param(prep_stmt1, 11,command )
            ibm_db.execute(prep_stmt1)
            return render_template('donordashboard.html', msg2="Profile
Updated successfully")
        else:
```

```
            return render_template('donordashboard.html', msg2="You are not
registered as a donor . Register Now!")




@app.route('/requestplasma', methods=['POST', 'GET'])
def requestplasma():
    if request.method == 'POST':
        name = request.form['name']
        username = request.form['username']
        gender=request.form['gender']
        email = request.form['email']
        phone = request.form['phone']
        bloodgroup=request.form['bloodgroup']
        street = request.form['street']
        city = request.form['city']
        region=request.form['region']
        pin =  request.form['pin']
        state = request.form['state']
        country = request.form['country']
        hoscity = request.form['hoscity']
        hosregion=request.form['hosregion']
        hospin = request.form['hospin']
```

```
        hosstate = request.form['hosstate']
        hoscountry = request.form['hoscountry']
        sql = "SELECT * FROM receivers WHERE username =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        if account:
            return render_template('receiverdashboard.html', msg2="You are already requested")
        else:
            insert_sql = "INSERT INTO receivers VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, username)
            ibm_db.bind_param(prep_stmt, 3, gender)
            ibm_db.bind_param(prep_stmt, 4, bloodgroup)
            ibm_db.bind_param(prep_stmt, 5,email )
            ibm_db.bind_param(prep_stmt, 6, phone)
            ibm_db.bind_param(prep_stmt, 7,street )
            ibm_db.bind_param(prep_stmt, 8, city)
            ibm_db.bind_param(prep_stmt, 9, region)
            ibm_db.bind_param(prep_stmt, 10, pin)
```

```
            ibm_db.bind_param(prep_stmt, 11,state)
            ibm_db.bind_param(prep_stmt, 12,country )
            ibm_db.bind_param(prep_stmt, 13, hoscity)
            ibm_db.bind_param(prep_stmt, 14, hosregion)
            ibm_db.bind_param(prep_stmt, 15, hospin)
            ibm_db.bind_param(prep_stmt, 16,hosstate)
            ibm_db.bind_param(prep_stmt, 17,hoscountry )
            ibm_db.execute(prep_stmt)
        return render_template('receiverdashboard.html', msg2="Plasma
Requested Successfully")


@app.route('/list')
def list():
    students = []
    sql = "SELECT * FROM receivers"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        # print ("The Name is : ", dictionary)
        students.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)

    if students:
        return render_template("donordashboard.html", students=students)
```

```
@app.route('/certificate', methods=['POST', 'GET'])
def certificate():
    if request.method == 'POST':
        name = request.form['name']
        count = request.form['count']

    return render_template("certificate.html", msg1=name,msg2=count)



@app.route('/donorlist')
def donorlist():
    students = []
    sql = "SELECT * FROM donor"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        # print ("The Name is : ", dictionary)
        students.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)

    if students:
        return render_template("receiverdashboard.html",
students=students,msg3="DONOR LIST")
```

```
@app.route('/feedback')
def feedback():
    students = []
    sql = "SELECT * FROM donor"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        # print ("The Name is : ", dictionary)
        students.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)

    if students:
        return render_template("feedback.html",
students=students,msg4="FEEDBACKS")

@app.route('/feeding', methods=['POST', 'GET'])
def feeding():
    if request.method == 'POST':
        feedtouser = request.form['feedtouser']
        feedback = request.form['feedback']
        sql = "INSERT INTO feedback(feedtouser,feedback) VALUES(?,?)"
        prep_stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(prep_stmt, 1, feedtouser)
        ibm_db.bind_param(prep_stmt, 2, feedback)
```

```
        ibm_db.execute(prep_stmt)
    return render_template("receiverdashboard.html",msg="FEEDBACK
send successfully")

@app.route('/verify', methods=['POST', 'GET'])
def verify():
    if request.method == 'POST':
        username = request.form['username']
        documenttype = request.form['documenttype']
        documentnumber = request.form['documentnumber']
        sql = "INSERT INTO
verify(username,documenttype,documentnumber) VALUES(?,?,?)"
        prep_stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(prep_stmt, 1, username)
        ibm_db.bind_param(prep_stmt, 2, documenttype)
        ibm_db.bind_param(prep_stmt, 3, documentnumber)
        ibm_db.execute(prep_stmt)
    return render_template('donordashboard.html',msg="Your details will be
verified soon . Thankyou!")
if__name__== '__main__':
    app.run(debug=True)
```

HTML TEMPLATES

Home.html

```
<html xmlns="http://www.w3.org/1999/html">
    <head>
{% block head %}
        {% endblock %}
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Indie+Flower&family=Zilla
+Slab:ital,wght@1,700&display=swap" rel="stylesheet">
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css
" rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1
WTRi" crossorigin="anonymous">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.mi
n.js" integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8
Qbsw3" crossorigin="anonymous"></script>
<script src="https://kit.fontawesome.com/e3edc23546.js"
crossorigin="anonymous"></script>
```

```
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js" integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3" crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js"
integrity="sha384-IDwe1+LCz02ROU9k972gdyvl+AESN10+x7tBKgc9I5HFtuNz0wWnPclzo6p9vxnk" crossorigin="anonymous"></script>

<link rel="stylesheet" href="https://plasmasonation.s3.jp-tok.cloud-object-storage.appdomain.cloud/style.css" type="text/css">

<link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.6.3/css/all.css"
integrity="sha384-UHRtZLI+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ8lWUE00s/" crossorigin="anonymous">


<script>
  window.watsonAssistantChatOptions = {
    integrationID: "34278875-92a4-4622-ad09-26a939641338", // The ID of this integration.
```

```
    region: "jp-tok", // The region your integration is hosted in.
    serviceInstanceID: "99e1e151-5742-4df4-8055-b5965108e707", // 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>
  </head>
<body background="https://plasmasonation.s3.jp-tok.cloud-object-
storage.appdomain.cloud/bg.jpg">

    <nav class="navbar navbar-dark bg-dark fixed-top">
        <div class="container-fluid">{% block heading %}

          <a class="navbar-brand"
href="https://en.wikipedia.org/wiki/Flask_(web_framework)" STYLE="font-
family:times new roman;">PLASMA DONATION</a>{% endblock %}
```

```
          <button class="navbar-toggler" type="button" data-bs-
toggle="offcanvas" data-bs-target="#offcanvasDarkNavbar" aria-
controls="offcanvasDarkNavbar">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="offcanvas offcanvas-end text-bg-dark" tabindex="-1"
id="offcanvasDarkNavbar" aria-labelledby="offcanvasDarkNavbarLabel">
            <div class="offcanvas-header">
              <h5 class="offcanvas-title"
id="offcanvasDarkNavbarLabel">Flask</h5>
              <button type="button" class="btn-close btn-close-white" data-bs-
dismiss="offcanvas" aria-label="Close"></button>
            </div>

{% block navbar %}
              <div class="offcanvas-body">
        <ul class="navbar-nav justify-content-end flex-grow-1 pe-3">

              <li class="nav-item">
                <a class="nav-link " aria-current="page" href="{{
url_for('signin') }}">Signin</a>
              </li>
              <li class="nav-item">
                <a class="nav-link" href="{{url_for('signup')}}">Signup</a>
              </li>
```

```
        </ul>
{% endblock %}
              <form class="d-flex mt-3" role="search">
                <input class="form-control me-2" type="search"
placeholder="Search" aria-label="Search">
                <button class="btn btn-success"
type="submit">Search</button>
              </form>
            </div>
          </div>
        </div>
      </nav><br<br>
      <br>

{% block body %}<br><br>

<center>

    <div class="container text-center">
        <div class="row">
          <div class="col">{% block div1 %}<br><br><br><br><br>

            <h1 class="h1">PLASMA DONOR APPLICATION </h1>
{% endblock %}
          </div>
```

```
            <div class="col">
                {% block content %}
        <div class="card" style="MARGIN-top:60PX; margin-
left:100px; border-radius:0px 20px 0px 20px;width:fit-
content; background-image: linear-gradient(to bottom right,
white,white,white ,blue,black);">
  <img src="https://plasmasonation.s3.jp-tok.cloud-object-
storage.appdomain.cloud/3.png" class="card-img-top" alt="login"
style="width:350px; height:220px; border-radius:0px 20px 0px 0px;">

  <div class="card-body" ><br>

    <a href="/signin" class="btn btn-dark btn-lg" style=" font-family: 'Zilla
Slab', serif; ">SIGN IN</a>
  </div>
</div>

{% endblock %}
{% endblock %}
</center>
<BR>
    <footer style="margin-top:150px;">
<div class="card text-center" style="background-color:black; text-
align:bottom; ">
```

```
        <div class="container text-center">
        <div class="row">
            <div class="col">
  <div class="card-body">
    <h5 class="card-title" style="text-align:left;">About </h5>
<table cellpadding="10" >
    <tr>
        <td> <a class="dn" href="#" class="">Privacy policies</a></td></tr>
         <tr>
    <td><a class="dn"  href="#" class="">Terms & conditions </a></td>
</tr>
     <tr>
    <td>    <a class="dn"  href="#" class="">Go somewhere</a></td>
</tr>
     <tr>
    <td> <a class="dn"   href="#" class="">Go somewhere</a></td>
</tr>
</table>
  </div></div>



                    <div class="col">
  <div class="card-body">
    <h5 class="card-title" style="text-align:left; ">Company</h5>
```

```
  <table cellpadding="10" >
   <tr>
       <td> <a class="dn" href="#" class="">Our story</a></td></tr>
        <tr>
   <td><a class="dn"  href="#" class="">Careers</a></td>
</tr>
    <tr>
   <td>    <a class="dn"  href="#" class="">Developer info</a></td>
</tr>
    <tr>
   <td> <a class="dn"   href="#" class="">Annual Report</a></td>
</tr>
</table></div>
 </div>


                    <div class="col">
 <div class="card-body">
   <h5 class="card-title" style="text-align:left">Help</h5>
      <table cellpadding="10" >
   <tr>
       <td> <a class="dn" href="#" class="">Contact Us</a></td></tr>
        <tr>
   <td><a class="dn"  href="#" class="">Help Center</a></td>
</tr>
    <tr>
```

```
    <td>    <a class="dn"  href="#" class="">FAQ</a></td>
</tr>
     <tr>
    <td> <a class="dn"   href="#" class="">Talk To Executives</a></td>
</tr>
</table></div>
  </div>


                        <div class="col">
  <div class="card-body">
    <h5 class="card-title" style="text-align:left;">Community</h5>
       <table cellpadding="10" >
    <tr>
        <td> <a class="dn" href="#" class="">Groups</a></td></tr>
         <tr>
    <td><a class="dn"  href="#" class="">Announcements</a></td>
</tr>
     <tr>
    <td>    <a class="dn"  href="#" class="">Events</a></td>
</tr>
     <tr>
    <td> <a class="dn"   href="#" class=""> Community Centers</a></td>
</tr>
</table></div>
  </div>
```

```
        </div></div><br>
  <div class="card-footer text-muted" style="padding:15px;">
    <div class="container text-center">
          <div class="card-header" style="text-align:left">
    Follow Us
  </div><br>
  <div class="row">
    <div class="col">
      <a href="www.facebook.com" STYLE="text-decoration:none;"><img
src="https://plasmasonation.s3.jp-tok.cloud-object-
storage.appdomain.cloud/facebook.png" width="30 cm"> FACEBOOK </a>
    </div>
    <div class="col">
      <a href="www.instagram.com" STYLE="text-decoration:none;"><img
src="https://plasmasonation.s3.jp-tok.cloud-object-
storage.appdomain.cloud/instagram.png" width="30 cm"> INSTAGRAM
</a>
    </div>
    <div class="col">
     <a href="www.youtube.com" STYLE="text-decoration:none;"><img
src="https://plasmasonation.s3.jp-tok.cloud-object-
storage.appdomain.cloud/youtube.png" width="30 cm"> YOUTUBE</a>
    </div>
       <div class="col">
```

```
    <a href="www.github.com/mdumar123.com" STYLE="text-
decoration:none;"><img src="https://plasmasonation.s3.jp-tok.cloud-object-
storage.appdomain.cloud/github.jpg" width="30cm"> GITHUB</a>
    </div>


  </div>

</div>

  </div>

</div>

    </footer>

</body>

</html>
```

Signin.html

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

 {% block navbar %}

<li class="nav-item ">
                    <a class="nav-link disabled"
href="{{url_for('signin')}}">Sign In</a>
                </li>

<li class="nav-item">
                    <a class="nav-link" href="{{url_for('signup')}}">Sign Up</a>
                </li>

 {% endblock %}

{% block content %}
<center>
    <form class="was-validated" action = "{{ url_for('log_in') }}" method =
"POST">

    <div style=" background-image: linear-gradient(to bottom right,
white,white,white ,blue,black); width:fit-content; border-radius:
30px;  margin:50px;">
```

```
<div style="padding:30px;">

      <div class="card-title">

   <img src="https://plasmasonation.s3.jp-tok.cloud-object-
storage.appdomain.cloud/5.jpg" class="card-img-top" alt="login"
style="width:200px; margin-top:0px;  border-radius:50%; ">
       <br>

        <div class="card-body">
            <div class="badge bg-primary text-wrap" style="width: 8rem;
padding-top: 10px;">

            <label for="username" class="form-label">Username </label>
            </div>
            <font color="white">:</font>

          <input type="text" class="col" placeholder="Enter a Username"
id="username" name="username" required >
      </div>
       <br>
      <div class="col">
        <div class="badge bg-primary text-wrap" style="width: 8rem;
padding-top: 10px;">
```

```
        <label for="password" class="form-label">Password </label>
        </div>
        <font color="white">:</font>
      <input type="text" class="col" id="password" placeholder="Enter
Password" name="password"
             pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}"
             title="Must contain at least one number and one uppercase and
lowercase letter, and at least 8 or more characters and match password"
             required>
      </div><h6 style="color:red; padding-top:20px;">{{msg}}</h6>
  <br>
  <div class="col-12">
    <button class="btn btn-primary" type="submit">SIGN IN </button>
  </div><br>
  <h5 style="font-style: italic;color:black; ">Don’t have an account? <a
href="{{url_for('signup')}}" style="text-decoration: none; font-style:
italic;color:YELLOW; font-family:times new roman; ">sign up</a></h5>
</div>
</div>
</center>
</form>
{% endblock %}
```

Signup.html

```
{% extends "home.html" %}
 {% block navbar %}
<li class="nav-item">
                    <a class="nav-link" href="{{url_for('signin')}}">Sign In</a>
                  </li>
<li class="nav-item">
                    <a class="nav-link disabled"
href="{{url_for('signup')}}">Sign Up</a>
                  </li>

 {% endblock %}

{% block content %}
<br><br><br>
<center>
    <form action = "{{ url_for('sign_up') }}" method = "POST">
    <div style=" background-image: linear-gradient(to bottom right,
white,white,white ,blue,black); width:fit-content; border-radius: 30px; ">
<div style="padding: 20px;">


        <img src="https://plasmasonation.s3.jp-tok.cloud-object-
storage.appdomain.cloud/5.jpg" class="card-img-top" alt="login"
style="width:200px; margin-top:0px;  border-radius:50%; "><br >
```

```
        <br>
        <div class="container text-center">
        <div class="col">
         <div class="badge bg-primary text-wrap" style="width: 8rem;
padding-top: 10px;">

         <label for="username" class="form-label">Username </label>
         </div>
         <font color="white">:</font>

       <input type="text" class="col" placeholder="Enter a Username"
id="username" name="username" required>

   </div><br>

         <div class=""mb-3"">
             <div class="badge bg-primary text-wrap" style="width: 8rem;
padding-top: 10px;">

             <label   for="exampleInputEmail1" class="form-label">Email Id
</label>
             </div>
             <font color="white">:</font>
```

```
          <input type="email" class="col" placeholder="Enter a email"
id="exampleInputEmail1" name="email" required>
          </div>
        <br>

       <div class="col">
         <div class="badge bg-primary text-wrap" style="width: 8rem;
padding-top: 10px;">

         <label for="password" class="form-label">Password </label>
         </div>
         <font color="white">:</font>
       <input type="password" class="col" placeholder="Enter Password"
id="password" name="password"
              pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}"
              title="Must contain at least one number and one uppercase and
lowercase letter, and at least 8 or more characters"
              required>
  </div>
  <br>
                  <div class="col">
         <div class="badge bg-primary text-wrap" style="width: 8rem;
padding-top: 10px;">
         <label for="repassword" class="form-label">Re-Enter Password
</label>
```

```
        </div>
        <font color="white">:</font>
      <input type="password" class="col" id="repassword" placeholder="Re-Enter Password" name="repassword"
             pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}"
             title="Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters and match password"
            required>
  </div>
  <br>
 </div><h6 style="color:red;  padding-top:10px; border-radius:30px;">{{msg}}</h6>
  <br>
  <div class="col-12">
    <button class="btn btn-primary" type="submit">SIGN UP </button>
  </div>
</div>
</div>
    </div>
</center>
</form>
{% endblock %}
```

Dashboard.html

```
{% extends "home.html" %}
 {% block navbar %}
<li class="nav-item ">
                  <a class="nav-link " href="{{url_for('logout')}}">LogOut</a>
                </li>
                {% block heading %}
                <h6 style="color:white; padding-bottom:0px; font-family:
'Times New Roman', Times, serif; font-size: 22px; ">
                WELCOME
                {{msg}} </h6>
                {% endblock %}
 {% endblock %}
{% block content %}
<br><br><br><br>
<div class="row" ">
  <div class="col-sm-6" >
    <div class="card" style=" background-color: mintcream; width: max-
content;">
      <div class="card-body">
        <h3 class="card-title" style="color: white; background-color: black;
width:fit-content;padding: 12px; border-radius: 5px; font-family: cooper
black;">Donor Dashboard</h3>
```

```
        <img src="https://plasmasonation.s3.jp-tok.cloud-object-
storage.appdomain.cloud/plasmadonor.jpg" width="280px;">
        <p class="card-text"></p>
        <a href="{{url_for('donor')}}" class="btn btn-dark">EXPLORE
HERE</a>
      </div>
    </div>
  </div>

  <div class="col-sm-6">
    <div class="card" style=" background-color: mintcream;;width: max-
content;">
      <div class="card-body">
        <h3 class="card-title" style="color: white; background-color: black;
width:fit-content;padding: 12px; border-radius: 5px; font-family: cooper
black;">Receiver Dashboard</h3>

        <img src="https://plasmasonation.s3.jp-tok.cloud-object-
storage.appdomain.cloud/plasmareceive.jpg" width="280px; ">

        <p class="card-text"></p>
        <a href="{{url_for('receiver')}}" class="btn btn-dark" style="text-
align: center;">EXPLORE HERE</a>
      </div>     </div>  </div></div>
{% endblock %}
```

Donordashboard.html

```
{% extends "home.html" %}
 {% block navbar %}
<li class="nav-item ">
                  <a class="nav-link " href="{{url_for('logout')}}">LogOut</a>
                </li>
                {% block heading %}
                <h6  style="color:white; padding-bottom:0px; font-family:
'Times New Roman', Times, serif; font-size: 22px; ">
                {{msg}}{{msg2}}{{msg4}} </h6>
                {% endblock %}
 {% endblock %}

{% block requestlist %}
{% block div1 %}
<br><a href="/list">

<h2 style="text-align: left; background-color:aliceblue; width: fit-content;
padding: 10px; font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial,
sans-serif; border-radius:10px;">REQUEST LIST </h2><br></a>
          {% for row in students %}

          <table style="margin-top:10px ; "">
           <tr>
```

```
        <div class="card text-bg-dark mb-3" style="max-width:400PX;">


         <div class="card-header" style="text-align:left ; font-size:20px; font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif; background-color:blue;"> {{row["NAME"]}}</div>
          <div class="card-body">
            <div class="container text-center">
            <div class="row">
            <div class="col">
             {{row["BLOODGROUP"]}} </div>
             <div class="col">
              {{row["CITY"]}} </div>
            <div class="col">
              {{row["PHONE"]}}

              </div></div></div>
          </div></div></div>
        </tr><br></table>
         {% endfor %}
   {% endblock %}
   {% endblock %}
{% block content %}
<br>
```

```
<div class="container overflow-hidden text-center" style="margin-top:30px ;
">
    <div class="row gy-5">
      <div class="col-6">
        <div class="p-3 border bg-dark overflow-hidden" ><div class="card">
            <div class="card-body">
              <h5 class="card-title bg-dark" style=" color:aliceblue;border-
radius :10px; padding:10px;">ADD ME DONOR</h5><br>
              <p class="card-text" style="text-align:justify ;">Click the below
botton and fill the form with your details.</p><br>
              <a href="/add-donor" class="btn btn-primary">Goto Form</a>
            </div>           </div>              </div>       </div>
      <div class="col-6">
        <div class="p-3 border bg-dark">
            <div class="card">
      <div class="card-body">
        <h5 class="card-title bg-dark" style=" color:aliceblue;border-radius
:10px; padding:10px;">UPDATE PROFILE</h5><br>
        <p class="card-text" style="text-align: JUSTIFY;">Click the below
botton to update your Profile</p><br>
        <a href="/updateprofile" class="btn btn-primary">UPDATE</a>
      </div>     </div>          </div>       </div>
      <div class="col-6">{% block verify %}
        <div class="p-3 border bg-dark">
            <div class="card">
```

```
        <div class="card-body">
          <h5 class="card-title bg-dark" style="color:aliceblue;border-radius
:10px; padding:10px;">VERIFY DONOR</h5><br>
          <p class="card-text" style="text-align: JUSTIFY;">Click the below
botton to Verify your Profile as Verified Donor</p><br>
          <a href="/verifydonor" class="btn btn-primary">VERIFY</a>
          {% endblock %}
        </div>     </div>          </div>       </div>
        <div class="col-6">
          <div class="p-3 border bg-dark">
              <div class="card">
        <div class="card-body">{% block reward%}
          <h5 class="card-title bg-dark" style=" color:aliceblue;border-radius
:10px; padding:10px;">REWARDS</h5><br>
          <p class="card-text" style="text-align: JUSTIFY;">Click the below
botton to collect your Reward</p><br>
          <a href="reward" class="btn btn-primary">COLLECT</a>{%
endblock %}
        </div>     </div>          </div>       </div>     </div>  </div>
 {% endblock %}
```

Adddonor.html

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

{% block navbar %}

<li class="nav-item ">
                    <a class="nav-link " href="{{url_for('logout')}}">LOG
OUT</a>
                </li>
                {% block heading %}
                <h6 style="color:white; padding-bottom:0px; font-family:
'Times New Roman', Times, serif; font-size: 22px; ">
                {{msg}}{{msg2}} </h6>
                {% endblock %}
{% endblock %}{% block head %}

<link
href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700"
rel="stylesheet">
    <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.5.0/css/all.css"
integrity="sha384-
B4dIYHKNBt8Bc12p+WXckhzcICo0wtJAoU8YZTY5qE0Id1GSseTk6S+L
3BlXeVIU" crossorigin="anonymous">
    <style>
```

```
html, body {
min-height: 100%;
}
body, div, form, input, select, textarea, label {
padding: 0;
margin: 0;
outline: none;
font-family: Roboto, Arial, sans-serif;
font-size: 14px;
color: #666;
line-height: 22px;
}
h1 {

margin:0;
font-family: cooper black;
font-size: 60px;
color: #fff;
z-index: 2;

padding: 15px;



}
legend {
```

```
    padding: 10px;
    font-family:COOPER BLACK;
    font-size: 22px;
    color:black;

    }
    textarea {
    width: calc(100% - 12px);
    padding: 5px;
    }
    .testbox {
    display: flex;
    justify-content: center;
    align-items: center;
    height: inherit;
    padding: 20px;
    }
    form  {
    width: 100%;
    padding: 20px;
    border-radius: 6px;
    background: #fff;
    box-shadow: 0 0 8px #006622;
    }
    .banner {
```

```
        position: relative;
        height: 250px;
        background-image:
url("/uploads/media/default/0001/02/cc6bc584f236c7234947015b89151ab6d
04c4cbf.jpeg");
        background-size: cover;
        display: flex;
        justify-content: center;
        align-items: center;
        text-align: center;
        }
        .banner::after {
        content: "";
        background:linear-gradient(to right,black,blue,black);
        position: absolute;
        width: 100%;
        height: 100%;
        }
        input, select, textarea {
        margin-bottom: 10px;
        border: 1px solid #ccc;
        border-radius: 3px;
        }
        input {
        width: calc(100% - 10px);
```

```
        padding: 5px;
        width: calc(100% - 12px);
        padding: 5px;
        }
        .item:hover p, .item:hover i, .question:hover p, .question label:hover,
input:hover::placeholder {
        color:  #006622;
        }
        .checkbox input[type=checkbox] {
        display:inline-block;
        height:15px;
        width:15px;
        margin-right:5px;
        vertical-align:text-top;
        }
        .item input:hover, .item select:hover, .item textarea:hover {
        border: 1px solid transparent;
        box-shadow: 0 0 3px 0  #006622;
        color: #006622;
        }
        .item {
        position: relative;
        margin: 10px 0;
        }
        .item span {
```

```
color: red;
}
.week {
display:flex;
justfiy-content:space-between;
}
.colums {
display:flex;
justify-content:space-between;
flex-direction:row;
flex-wrap:wrap;
}
.colums div {
width:48%;
}
input[type=radio], input[type=checkbox] {
display: none;
}
label.radio {
position: relative;
display: inline-block;
margin: 5px 20px 15px 0;
cursor: pointer;
}
.question span {
```

```
margin-left: 30px;
}
.question-answer label {
display: block;
}
label.radio:before {
content: "";
position: absolute;
left: 0;
width:  17px;
height: 17px;
border-radius: 50%;
border: 2px solid #ccc;
}
input[type=radio]:checked + label:before, label.radio:hover:before {
border: 2px solid  #006622;
}
label.radio:after {
content: "";
position: absolute;
top: 6px;
left: 5px;
width: 8px;
height: 4px;
border: 3px solid  #006622;
```

```
border-top: none;
border-right: none;
transform: rotate(-45deg);
opacity: 0;
}
input[type=radio]:checked + label:after {
opacity: 1;
}
.flax {
display:flex;
justify-content:space-around;
}
.btn-block {
margin-top: 10px;
text-align: center;
}
button  {
width: 150px;
padding: 10px;
border: none;
border-radius: 5px;
background: #1c87c9;
font-size: 16px;
color: #fff;
cursor: pointer;
```

```
    }
    button:hover {
    background: #0692e8;
    }
    @media (min-width: 568px) {
    .name-item, .city-item {
    display: flex;
    flex-wrap: wrap;
    justify-content: space-between;
    }
    .name-item input, .name-item div {
    width: calc(50% - 20px);
    }
    .name-item div input {
    width:97%;}
    .name-item div label {
    display:block;
    padding-bottom:5px;
    }
    }
  </style>

{% endblock %}
{% block body %}
```

```
<div class="testbox">

  {% block form %}
  <form action="{{url_for('addrec')}}" method="post">
  {% endblock %}

      <div class="banner">
        <h1 style="font-size: 70px;">PLASMA DONOR
APPLICATION</h1>
      </div>
      <br/>
      <fieldset>
        <legend>Blood Donation Form</legend>
        <div class="colums">
          <div class="item">
            <label for="name">Full Name<span>*</span></label>
            <input id="name" type="text" name="name" required>
          </div>
          <div class="item">
            <label for="username"> User Name<span>*</span></label>
            <input id="username" type="text" name="username" required />
          </div>
          <div class="item">
            <label for="email">Email Address<span>*</span></label>
            <input id="email" type="email"    name="email" required />
```

```
</div>
<div class="item">
  <label for="phone">Phone Number</label>
  <input id="phone" type="number"    name="phone" required/>
</div>
<div class="item">
  <label for="street">Street Address</label>
  <input id="street" type="text"    name="street" required />
</div>
<div class="item">
  <label for="city">City</label>
  <input id="city" type="text"    name="city" required />
</div>
<div class="item">
  <label for="pin">Zip/Postal Code</label>
  <input id="pin" type="text"    name="pin" required/>
</div>
<div class="item">
  <label for="state">State</label>
  <input id="state" type="text"    name="state" required />
</div>
<div class="item">
  <label for="country">Country</label>
  <input id="country" type="text"    name="country" required />
</div>
```

```
        </fieldset>
        <br/>
        <fieldset>
        <legend>Donation Details</legend>
        <div class="colums">
        </div>

        <div class="item">
        <label for="group">Blood Group<span>*</span></label>
        <input id="group" type="text"    name="bloodgroup" />
        </div>
        <div class="item">
        <label for="donation">Donation Comments If Any</label>
        <textarea id="donation" rows="3" name="command"></textarea>
        </div>
        </fieldset>
        {% block submit %}
        <div class="btn-block">
          <button type="submit" value="submit" >Submit</button>
          </div>
{% endblock %}
    </form>
    </div>
{% endblock %}
```

Updateprofile.html

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


{% block navbar %}


<li class="nav-item ">
                    <a class="nav-link " href="{{url_for('logout')}}">Log
Out</a>
                </li>
                {% block heading %}
                <h6 style="color:white; padding-bottom:0px; font-family:
'Times New Roman', Times, serif; font-size: 22px; ">
                {{msg}}{{msg2}}{{msg4}} </h6>
                {% endblock %}
 {% endblock %}
{% block form %}
<form action="{{url_for('update')}}" method="post">
{% endblock %}
{% block submit %}
<div class="btn-block">
    <button type="submit" value="submit" >Submit</button>
    </div>
{% endblock %}
```

Verify.html

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

{% block navbar %}

<li class="nav-item ">
                    <a class="nav-link " href="{{url_for('logout')}}">Log
Out</a>
                </li>


                {% block heading %}
                <h6  style="color:white; padding-bottom:0px; font-family:
'Times New Roman', Times, serif; font-size: 22px; ">
                {{msg}}{{msg2}}{{msg4}} </h6>
                {% endblock %}


 {% endblock %}
{% block requestlist %}
<BR>
    <img src="https://plasmasonation.s3.jp-tok.cloud-object-
storage.appdomain.cloud/feedback.png" width="500px;">
{% endblock %}
{% block verify %}
<form action="/verify" method="post" >
```

```
<div class="p-3 border bg-primary">
    <div class="card">
<div class="card-body">
<h5 class="card-title" style="background-color:blue; color:aliceblue;border-radius :10px; padding:10px;">{{msg4}}</h5><br>

<table >
    <tr >
<input type="text" name="username" placeholder="username"> </tr><br>
<tr>
    <input type="text" name="documenttype" placeholder="Aadhar/Voter Id"> </tr><br>
</tr>
<tr>

  <input type="number" name="documentnumber" placeholder="Document Number">
  </tr></table><br>

  <button type="submit" class="btn btn-primary" value="submit">SUBMIT</button>
</form>
{% endblock %}
```

Reward.html

```
{% extends "donordashboard.html" %}
{% block navbar %}
<li class="nav-item ">
                    <a class="nav-link " href="{{url_for('logout')}}">Log
Out</a>
                </li>
                {% block heading %}
                <h6 style="color:white; padding-bottom:0px; font-family:
'Times New Roman', Times, serif; font-size: 22px; ">
                {{msg}}{{msg2}}{{msg4}} </h6>
                {% endblock %}
 {% endblock %}
{% block reward%}
<form action="/certificate" method="post">
    <h5 class="card-title" style="background-color:blue;
color:aliceblue;border-radius :10px; padding:10px;">{{msg4}}</h5><br>
<input type="text" name="name" placeholder="Full name">
<input type="number" name="count" placeholder="No of Donation
count"><br><br>
<button type="submit" class="btn btn-primary"
value="submit">SUBMIT</button>
</form>
{% endblock %}
```

Certificate.html

```
<html>
    <head>
        <style type='text/css'>
            body, html {
                margin: 0;
                padding: 0;
            }
            body {
                color: black;
                display: table;
                font-family: Georgia, serif;
                font-size: 24px;
                text-align: center;

            }
            .container {
                border: 20px solid tan;
                width: 750px;
                height: 563px;
                display: table-cell;
                vertical-align: middle;
                background-color: aqua;
            }
```

```
.logo {
    color:red;
}

.marquee {
    color:brown;
    font-size: 48px;
    margin: 20px;
}
.assignment {
    margin: 20px;
}
.person {
    border-bottom: 2px solid black;
    font-size: 32px;
    font-style: italic;
    margin: 20px auto;
    width: 400px;
}
.reason {
    margin: 20px;
}
```

```
            .btn {
  background-color: DodgerBlue;
  border: none;
  color: white;
  padding: 12px 30px;
  cursor: pointer;
  font-size: 20px;
}

/* Darker background on mouse-over */
.btn:hover {
  background-color: RoyalBlue;
}
        </style>
    </head>
    <body>

                <div class="container" >
            <div class="logo">
                WEB SOLUTE(
                    Complete web solution organization
                )
            </div>

            <div class="marquee">
```

```
            Certificate of Plasma Donation
        </div>

        <div class="assignment">
            This certificate is presented to
        </div>

        <div class="person">

        {{msg1}}
        </div>

        <div class="reason">
           For successfully donated plasma for {{msg2}} times through our plasma donor application
        </div>
    </div>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<button class="btn"><i class="fa fa-download"></i> Download</button>
</body>
</html>
```

Receiverdashboard.html

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

 {% block navbar %}

<li class="nav-item ">
                    <a class="nav-link " href="{{url_for('logout')}}">Log
Out</a>
                </li>

                {% block heading %}
                <h6 style="color:white; padding-bottom:0px; font-family:
'Times New Roman', Times, serif; font-size: 22px; ">
                {{msg}}{{msg2}} </h6>
                {% endblock %}



                {% block div1 %}

                {% block donorlist %}
```

```
<br>
<h2 style="text-align: left; background-color:aliceblue; width: fit-content; padding: 10px; font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif; border-radius:10px;">{{msg3}} </h2><br>
{% for row in students %}

<table style="margin-top:10px ; "">
<tr>
<div class="card text-bg-dark mb-3" style="max-width:400PX;">

<div class="card-header" style="text-align:left ; font-size:20px; font-family:'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif; background-color:blue;"> {{row["NAME"]}}</div>
<div class="card-body">
<div class="container text-center">
<div class="row">
<div class="col">
{{row["BLOODGROUP"]}} </div>
<div class="col">
{{row["CITY"]}} </div>
<div class="col">
```

```
                        {{row["PHONE"]}}
                                                </div></div></div>
                    </div></div></div>
                  </tr><br></table>
                  {% endfor %}
                            {% endblock %}


 {% endblock %}
 {% endblock %}
{% block content %}
<div class="container overflow-hidden text-center" style="margin-top:30px
;">
  <div class="row gy-5">
    <div class="col-6">
      <div class="p-3 border bg-dark"><div class="card">
          <div class="card-body">
            <h5 class="card-title" style="background-color: black;
color:aliceblue;border-radius :10px; padding:10px;">REQUEST
PLASMA</h5><br>
            <p class="card-text" style="text-align:justify ;">Click the below
botton to request blood plasma.</p><br>
            <a href="/request_1" class="btn btn-primary">REQUEST</a>
          </div>
        </div>
```

```
      </div>
    </div>
    <div class="col-6">
      <div class="p-3 border bg-dark">
          <div class="card">
    <div class="card-body">
      <h5 class="card-title" style="background-color: black;
color:aliceblue;border-radius :10px; padding:10px;">SEARCH
DONOR</h5><br>
      <p class="card-text" style="text-align: JUSTIFY;">Click the below
botton to search donor near you.</p><br>
      <a href="/donorlist" class="btn btn-primary">SEARCH</a>
    </div>
  </div>
      </div>
    </div>
    <div class="col-6">
      <div class="p-3 border bg-dark">
          <div class="card">
    <div class="card-body">
      <h5 class="card-title" style="background-color: black;
color:aliceblue;border-radius :10px; padding:10px;"> SWITCH
ROLE</h5><br>
      <p class="card-text" style="text-align: JUSTIFY;">Click the below
button to Switch role as a Donor</p><br>
```

```
      <a href="{{url_for('donor')}}" class="btn btn-primary">DONATE</a>
    </div>
  </div>
      </div>
    </div>
    <div class="col-6">
      {% block feedback %}
      <div class="p-3 border bg-dark">
          <div class="card">
    <div class="card-body">
      <h5 class="card-title" style="background-color: black;
color:aliceblue;border-radius :10px; padding:10px;">FEEDBACK</h5><br>
      <p class="card-text" style="text-align: JUSTIFY;">Click below and
write how the plasma donation helps you.</p><br>
      <a href="/feedback" class="btn btn-primary">FEEBBACK</a>{%
endblock %}
    </div>
  </div>
      </div>
    </div>
  </div>
</div>
{% endblock %}
```

Request.html

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

 {% block navbar %}

<li class="nav-item ">
                    <a class="nav-link " href="{{url_for('logout')}}">LOG
OUT</a>
                </li>


                {% block heading %}
                <h6  style="color:white; padding-bottom:0px; font-family:
'Times New Roman', Times, serif; font-size: 22px; ">
                {{msg}}{{msg2}} </h6>
                {% endblock %}



 {% endblock %}
{% block head %}

<link
href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700"
rel="stylesheet">
```

```
    <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.5.0/css/all.css"
integrity="sha384-
B4dIYHKNBt8Bc12p+WXckhzcICo0wtJAoU8YZTY5qE0Id1GSseTk6S+L
3BlXeVIU" crossorigin="anonymous">
    <style>
      html, body {
      min-height: 100%;
      }
      body, div, form, input, select, p {
      padding: 0;
      margin: 0;
      outline: none;
      font-family: Roboto, Arial, sans-serif;
      font-size: 14px;
      color: #666;
      line-height: 22px;
      }
      h1 {

   margin:0;
   font-family: cooper black;
   font-size: 60px;
   color: #fff;
   z-index: 2;
```

```
padding: 15px;



}
    span.required {
    color: red;
    }
    .testbox {
    display: flex;
    justify-content: center;
    align-items: center;
    height: inherit;
    padding: 20px;
    }
    form  {
    width: 100%;
    padding: 20px;
    border-radius: 6px;
    background: #fff;
    box-shadow: 0 0 30px 0 #095484;
    }
    .banner {
    position: relative;
    height: 180px;
```

```
background-size: cover;
display: flex;
justify-content: center;
align-items: center;
text-align: center;
}
.banner::after {
content: "";
background:linear-gradient(to right,black,blue,black);
position: absolute;
width: 100%;
height: 100%;
}
p.top-info {
margin: 10px 0;
}
input, select {
margin-bottom: 10px;
border: 1px solid #ccc;
border-radius: 3px;
}
input {
width: calc(100% - 10px);
padding: 5px;
```

```
    }
    select {
    width: 100%;
    padding: 7px 0;
    background: transparent;
    }
    .item:hover p, .question:hover p, .question label:hover,
input:hover::placeholder {
    color: #095484;
    }
    .item input:hover, .item select:hover {
    border: 1px solid transparent;
    box-shadow: 0 0 5px 0 #095484;
    color: #095484;
    }
    .item {
    position: relative;
    margin: 10px 0;
    }
    .question input {
    width: auto;
    margin: 0;
    border-radius: 50%;
    }
    .question input, .question span {
```

```
vertical-align: middle;
}
.question label {
display: inline-block;
margin: 5px 20px 15px 0;
}
.btn-block {
margin-top: 10px;
text-align: center;
}
button  {
width: 150px;
padding: 10px;
border: none;
border-radius: 5px;
background: #095484;
font-size: 16px;
color: #fff;
cursor: pointer;
}
button:hover {
background: #0666a3;
}
@media (min-width: 568px) {
.name-item, .city-item {
```

```
        display: flex;
        flex-wrap: wrap;
        justify-content: space-between;
        }
        .name-item input, .city-item input {
        width: calc(50% - 20px);
        }
        .city-item select {
        width: calc(50% - 8px);
        }
        }
    </style>

{% endblock %}

{% block body %}
<div class="testbox">
    <form action="/requestplasma" method="post">
        <div class="banner">
            <h1 style="font-size:60px; ;">PLASMA DONOR
APPLICATION</h1>
        </div>
     <h2 style="font-family:COOPER BLACK;padding:10px;
color:black">Plasma Request Form</h2>
        <div class="item">
```

```
        <p  style="font-family: cooper black;">Name<span
class="required">*</span></p>
        <div class="name-item">
          <input type="text" name="name" placeholder="Fullname"
required/>
          <input type="text" name="username" placeholder="username"
required/>
        </div>
      </div>
      <div class="question">
        <p  style="font-family: cooper black;">Gender<span
class="required">*</span></p>
        <div class="question-answer">
          <label><input type="radio" value="none" name="gender" required/>
<span>Male</span></label>
          <label><input type="radio" value="none" name="gender" required/>
<span>Female</span></label>
        </div>
      </div>
      <div class="item">
        <p  style="font-family: cooper black;">Blood Group<span
class="required">*</span></p>
        <input type="text" name="bloodgroup" required/>
      </div>
      <div class="item">
```

```
        <p  style="font-family: cooper black;">Email<span
class="required">*</span></p>
        <input type="email" name="email" required/>
      </div>
      <div class="item">
        <p  style="font-family: cooper black;">Phone<span
class="required">*</span></p>
        <input type="text" name="phone"/>
      </div>

      <div class="item">
        <p  style="font-family: cooper black;">Home Address<span
class="required">*</span></p>
        <input type="text" name="street" placeholder="Street address"
required/>
        <div class="city-item">
          <input type="text" name="city" placeholder="City" required/>
          <input type="text" name="region" placeholder="Region" required/>
          <input type="text" name="pin" placeholder="Postal / Zip code"
required/>
          <input type="text" name="state" placeholder="state"
required/>
          <input type="text" name="country" placeholder="country"
required/>
        </div>
```

```
      </div>

      <div class="item">
        <p style="font-family: cooper black;">Hospital Address If need:</p>
        <input type="text" name="hosstreet" placeholder="Street address" />
        <div class="city-item">
          <input type="text" name="hoscity" placeholder="City" >
          <input type="text" name="hosregion" placeholder="Region"/>
          <input type="text" name="hospin" placeholder="Postal / Zip code"
/>
          <input type="text" name="hosstate" placeholder="state"
required/>
          <input type="text" name="hoscountry" placeholder="country" />
        </div>
      </div>


      <div class="btn-block">
        <button type="submit" value="submit">REQUEST</button>
      </div>
    </form>
  </div>
{% endblock %}
```

Feedback.html

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


{% block navbar %}

<li class="nav-item ">
                    <a class="nav-link " href="{{url_for('logout')}}">Log
Out</a>
                </li>


                {% block heading %}
                <h6 style="color:white; padding-bottom:0px; font-family:
'Times New Roman', Times, serif; font-size: 22px; ">
                {{msg}}{{msg2}}{{msg4}} </h6>
                {% endblock %}


 {% endblock %}
{% block donorlist %}
<BR>
    <img src="https://plasmasonation.s3.jp-tok.cloud-object-
storage.appdomain.cloud/feedback.png" width="500px;">
{% endblock %}
{% block feedback %}
```

```
<form action="feeding" method="post" >

<div class="p-3 border bg-primary">
    <div class="card">
<div class="card-body">
<h5 class="card-title" style="background-color:blue; color:aliceblue;border-radius :10px; padding:10px;">{{msg4}}</h5><br>

<select class="form-select"  aria-label="size 3 select example" aria-placeholder="Select User" name="feedtouser">


  <option selected >Admin</option>
  {% for row in students %}
    <option >{{row["USERNAME"]}}</option>
    {% endfor %}
  </select><br>
  <textarea name="feedback" placeholder="Type your feedback here">feedback </textarea>
  <button type="submit" class="btn btn-primary" value="submit">SUBMIT</button>
</form>
{% endblock %}
```

GitHub Link

IBM-EPBL/IBM-Project-19746-1659705698: Plasma Donor Application (github.com)

Project Demo Link

https://vimeo.com/773075891

NTP_Project_Demo.mp4 on Vimeo