PROJECT REPORT DOCUMENTATION

FINAL REPORT

Project Name	Project - Plasma Donor App
Team ID	PNT2022TMID23196
Team Leader	S.Sneha
Team Member 1	M.Susanthika
Team Member 2	G.S.Urmikha
Team Member 3	R.Sruthi

1. INTRODUCTION

1.1 Project Overview:

The main goal of our project is to design a user-friendly web application that is like a scientific vehicle from which we can help reduce mortality or help those affected by COVID-19 by donating plasma from patients who have recovered without approved antiretroviral therapy planning for deadly COVID-19 infection, plasma therapy is an experimental approach to treat those COVID-positive patients and help them recover faster. Therapy, which is considered reliable and safe. If a particular person has fully recovered from COVID-19, they are eligible to donate their plasma. As we all know, the traditional methods of finding plasma, one has to find out for oneself by looking at hospital records and contacting donors have been recovered, sometimes may not be available at home and move to other places. In this type of scenario, the health of those who are sick becomes disastrous. Therefore, it is not considered a rapid process to find plasma. The main purpose of the proposed system, the donor who wants to donate plasma can simply register through the web application and can donate the plasma to the blood bank, the blood bank can apply for the donor and once the donor has accepted the request, the blood bank can add the units they need and the hospital can also send the request to the blood bank that urgently needs the plasma for the patient and can take the plasma from the blood bank.

1.2 Purpose:

The Plasma Donation Application would help Donors, as well as patients in need of plasma. It would allow you to search for Plasma Donors within your city and having a specific Blood Group. People who have fully recovered from COVID-19 have antibodies in their plasma that can attack the virus. This convalescent plasma is being evaluated as a treatment for patients with serious or immediately life-threatening COVID-19 infections, or those judged by a healthcare provider to

be at high risk of progression to severe or life-threatening disease. This application can be considered as a contribution of its developers towards the medical unit of the country as well as towards humanity.

2. LITERATURE SURVEY

2.1 Existing Problem:

When a patient needs plasma, he/she has to contact a compatible donor on their circle, but it is difficult to find a suitable donor in a group for a particular time of period. Currently people in need of plasma post pleas on social media to attract potential donors, but pleas on social media take longer to reach a wider audience. As a result, recipients are unable to find the donors within the required time.

2.2 References:

1.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; Plasma separation is of great importance in the fields of diagnosis and healthcare. Due to the lagging transition to micro scale, these recent trends are a rapid shift towards shrinking complex macro processes.

2. Kalpana DeviGuntoju, Tejaswini Jalli, Sreeja Uppala, Sanjay Mallisettiinstant plasma donor recipient connector web application 2022. JOURNAL: InternationalResearch Journal of modernization in engineering technology and Science

3. M Sai Tarun, Ravi Kishan, Shaik AzaadSuraz Basha, Shaik RajAhammad, Chandrasekhar, Neha BaggaBlood BankManagement System2021. Journal of Emerging Technologies and Innovative Research.

4.Nayan Das, MDAsif Iqbal Nearest Blood Plasma Donor Finding: A Machine Learning Approach 2020 23rd International Conference on Computer and Information Technology.

5. Ms.PradnyaJagtap, Ms.MonikaMandale, Ms.PrachiMhaske, Ms.SonaliVidhate, Mr. S.S. Patil Implementation of blood donation application using android smartphone 2018 Open access International journal of science & engineering.

S.No.	Title	Author	Abstract

| 1. | Nearest Blood & Plasma Donor Finding: A MachineLearning Approach | Nayan Das, MD. Asif Iqbal | The necessity of blood has become a significant concern in the present context all over the world. Due to a shortage of blood, people couldn't save themselves or their friends and family members. A bag of blood can save a precious life. Statistics show that a tremendous amount of blood is needed yearly because of major operations, road accidents, blood disorders, including Anemia, Hemophilia, and acute viral infections like Dengue, etc. Approximately 85 million people require single or multiple blood transfusions for treatment. Voluntary blood donors per 1,000 population of some countries are quite promising, such as Switzerland (113/1,000), Japan (70/1,000), while others have an unsatisfying result like India has 4/1,000, and Bangladesh has 5/1000. Recently a life-threatening virus, COVID-19, spreading throughout the globe, which is more vulnerable for older people and those with pre-existing medical conditions. For them, plasma is needed to recover their illness. Our Purpose is to build a platform with clustering algorithms which will jointly help to provide the quickest solution to find blood or plasma donor. Closest blood or plasma donors of the same group in a particular area can be explored within less time and more efficiently. |
| 2. | Blood Bank System using Database Security | Dr. Danie Kingsley, Asst. Prof. Sr, SBST | Despite the immense technological advancement, blood bank systems are either manual or valuable data is easily retrievable. Consequently, one of the major issues in blood bank systems, as talked about in many research papers and articles, |

			is the lack of data security. People always doubt whether their personal information and medical records are safely stored and secured. Therefore, our project aims to develop an online blood donation system applying the concepts of database security and encryption. The following is what our project aims to achieve: Any person who is willing to donate blood will have to register first, even if the user is a new donor, or the user can directly login if he/she has an account already. Whenever they want to donate blood, a form will have to be filled. In the user account, the user will be able to view all the details and records of all earlier donations as well as information about upcoming blood donation events. There will be a link provided to find blood donors in the region of the users' choice. All this is related to the blood bank system. Apart from this, we will be using concepts of database encryption to make sure that the users' information is kept secure and confidential. This will help us keep their donation records protected from any threats from individuals with potentially malicious intentions, or any unforeseen hazards to the security of the data.
3.	Generation and Applications of Plasma	Faizan Mehmood, Tariq Kamal and Umair Ashra	Plasma being the fourth and most abundant form of matter extensively exists in the universe in the inter-galactic regions. It provides an electrically neutral medium of unbound negative and positive charged particles, which has been produced by subjecting air and various other gaseous mixtures to strengthen the electromagnetic field and by

			heating compressed air or inert gasses for creating negative and positive charged particles known as ions. Nowadays, many researchers are paying attention to the formation of artificial Plasma and its potential benefits for mankind. The literature is sparsely populated with the applications of Plasma. This paper presents specific methods of generation and applications of Plasma, which benefits humankind in various fields, such as in electrical, mechanical, chemical and medical fields. These applications include hydrogen production from alcohol, copper bonding, semiconductor processing, surface treatment, Plasma polymerization, coating, Plasma display panels, antenna beam forming, nanotechnology, Plasma Torch, Plasma pencils, low-current non-thermal Plasmatron, treatment of prostate cancer, Plasma source ion implantation, cutting by Plasma, Plasma etching, pollution control, neutralization of liquid radioactive waste, etc. Resultantly, the worth of Plasma technology in the medical industry is increasing exponentially, closing the gap between its benefits and cost of equipment used for generating and controlling it.
4.	Plasma Technology Research and Its Applications	Muhammad Nur	Researchers on the application of plasma technology in the areas of environment, health, food, agriculture have been conducted in the Laboratory of Atomic and Nuclear Physics Division of Plasma Technology in the Faculty of Science and Mathematics, University of Diponegoro. This paper reported research results on plasma technology and its

			applications in these fields that have been carried out in the recent years. Plasma for environmental applications can reduce gas emissions released by motorcycle and vehicle exhausts. This technology can reduce significantly emissions of SOx, COx, and NOx. Non-polluted plasma muffler prototype adaptation has been done in four wheels and more vehicles. Pilot scale improvement has done by integrating a reduction system into vehicle muffler from it previous position outside the muffler. High voltage that used to develop plasma condition comes from 12 V 34 A accumulator which connected with electronic equipment and able to develop voltage up to 20kV. Exhaust gases reduction Ability has done by varying engine rotation. Plasma muffler appearance in vehicle doesn't change outside dimension of its original muffler and it reactor placement in muffler has a function to change resonator chamber function and make this muffler still fulfill muffler standardization with more performance in reducing exhaust gases (COx, NOx. HC). Optimal reduction level made at 2200 rpm for COx is 88,52%. for CO is 88,93%, for HC is 97,34% and for NOx at 4800 rpm is 76,19%.
5.	Applications of Plasma-Liquid Systems	Fatemeh Rezaei L, Patrick Vanraes AntonNikiforov, Rino Morent and Nathalie De Geyter	Plasma-liquid systems have attracted increasing attention in recent years, owing to their high potential in material processing and nanoscience, environmental remediation, sterilization, biomedicine, and food applications. Due to the multidisciplinary character of this

			scientific field and due to its broad range of established and promising applications, an updated overview is required, addressing the various applications of plasma-liquid systems till now. In the present review, after a brief historical introduction on this important research field, the authors aimed to bring together a wide range of applications of plasma-liquid systems, including nanomaterial processing, water analytical chemistry, water purification, plasma sterilization, plasma medicine, food preservation and agricultural processing, power transformers for high voltage switching, and polymer solution treatment. Although the general understanding of plasma-liquid interactions and their applications has grown significantly in recent decades, it is aimed here to give an updated overview on the possible applications of plasma-liquid systems. This review can be used as a guide for researchers from different fields to gain insight in the history and state-of-the-art of plasma-liquid interactions and to obtain an overview on the acquired knowledge in this field up to now

2.3 Problem Statement Definition:

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

Who does the problem affect?

People who are affected by COVID and are in need of a Plasma Donor.

What is the issue?

When a patient needs plasma,he/she has to contact a compatible donor on their circle,family and friends but it is difficult to find suitable donor within a limited group of people in a given time.

What is the impact of the issue?

During the COVID 19 crisis, the requirement of plasma became high and the donor count being low. It is very difficult to find the respective blood group donors when someone is in need.

What would happen if we didn't solve the problem?

The gap between the Donor and Recipient would widen. People who are eager to donate plasma cannot find the right recipient. Currently, people in need of Plasma post Pleas on Social Media to attract potential donors. But Plea's on social media take longer to reach a wider audience. As a result recipients are unable to find donors within the required time.

What would happen when it is fixed?

The application makes it feasible for the COVID-19 patients to get a plasma donor easily and makes it possible to find a plasma donor without much difficulty.

Why is it important that we fix the problem?
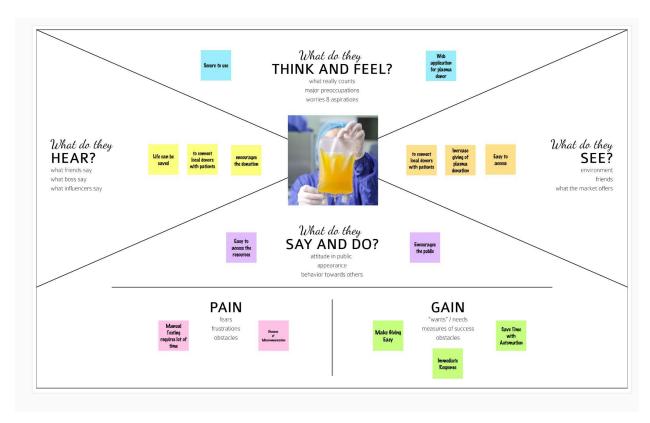
In severe cases if the recipient is unable to find a donor, then his/her condition could worsen and may potentially result in death.

Problem	I am	I'm trying to	But	Because	Which makes me feel

Statement (PS)	(Customer)				
PS-1	a plasma recipient	get a plasma from donor through online	i am unable to find donor	the covid 19 reduce the supply and increase the need of plasma.	worry and depressed about their health
PS-2	a plasma donor	donate a plasma through Web app	il can't donate a plasma immediately	I don't know whether the recipient is in normal or critical condition	Anxious

3. IDEATION & PROPOSED SOLUTION
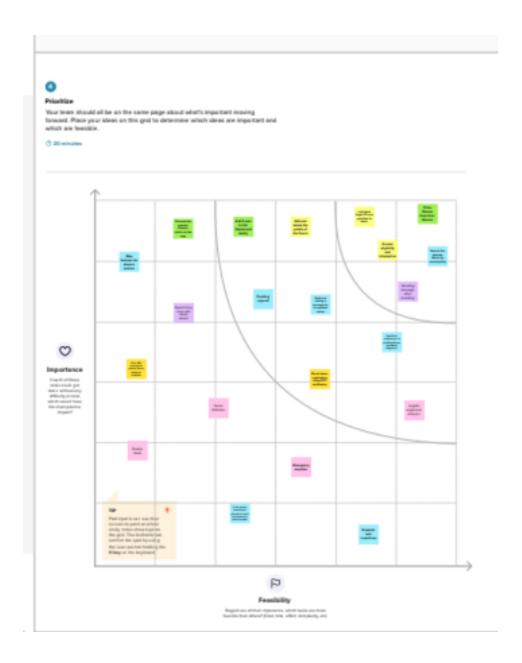
3.1 Empathy Map Canvas

3.2 Ideation and Brainstorming

PLASMA DONOR APPLICATION

PROBLEM STATEMENT :

During the COVID 19 crisis, the requirement of plasma became a high priority and the donor count has become low. Saving the donor information and helping the needy by notifying the current donors list, would be a helping hand. In regard to the problem faced,

an application is to be built which would take the donor details, store them and inform them upon a request.

IDEA / SOLUTION DESCRIPTION :

Plasma donor is an application which will make things easier and efficient at crucial times and to solve our problem statement.

Some of the features which can be incorporated are :
- The user and the donor both register all relevant information.
- An email message will be issued after registration is complete.
- The user has the option of sending a request for a blood group in need or donating plasma in this.
- Statistics for various blood types are shown and updated regularly.
- It contains details regarding plasma donation camps, including information about the location of the events.
- The users can choose to obtain a home sample collection as well.
- E- certificates can be provided.

NOVELTY / UNIQUENESS :

A visual representation that is simple for users to understand will be used to display the statistics for the blood group availability data for plasma donation. The user can send a request for plasma if they are unsure about its availability in their immediate vicinity. Whether plasma is in short supply or is more readily available, users will receive an email notification within a short period of time. If individuals sign up for our application for plasma donors and decide they want to donate plasma, they can schedule an appointment. They will obtain their e-certification for donating plasma once they have completed their session according to schedule.These are the innovative elements included in this.

SOCIAL IMPACT / CUSTOMER SATISFACTION :

We reside in a modern world where digital health services are available and can be accessed online instantly. Despite the apparent abundance of resources, there are still cases where hospitals or blood banks run out of essential resources, such as specific blood type shortages. For instance, In the United States, a blood donation request is generated approximately every two seconds, and blood transfusion events save 4.5

million lives annually. One of the major issues health
facilities run into is the shortage of certain blood types. An additional problem is
facilities need access to patient data as quickly as possible before beginning patient
blood transfer.

Considering such issues, the availability and knowledge base of collected blood types with patient history is critical. To solve these issues, modern software applications are employed with Machine Learning, Cloud Computing, and Internet of Things (IoT) tools which enable features such as significant information retrieval, continuous data tracking with analytics, and Cloud-powered search engines.
This application, along with all the services it provides, also helps to eradicate certain spam messages and mails circulating around regarding fake or already satisfied blood emergency situations . A single platform for maintaining all genuine blood related activities and information increases the trust of the public to get involved in these activities, and to participate in blood donations.

BUSINESS MODEL :
An unpaid application exists for plasma donors. It is readily available and accessible by all. Due to the difficulty in locating donors who match a certain blood group, this application enables users to register people who wish to donate plasma and keep their information in a database. By informing the current donors of the need, saving the donor information would assist. The need for plasma increased significantly during the COVID 19 crisis, and the number
of donors is limited. In the end, working with the government can use an app to aid those in need of plasma.
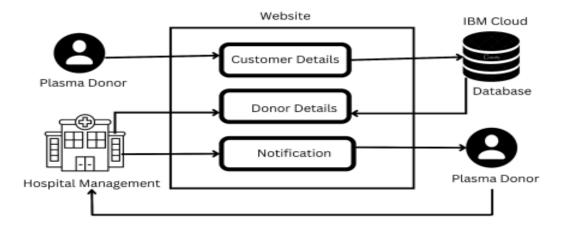
SCALABILITY OF SOLUTION :
This application assists users in finding the closest blood centre, knowing their eligibility to donate blood, receiving notifications when an urgent blood donation call comes in, and scheduling a convenient appointment utilising temporal and/or spatial information. A current donor profile will be used, containing details such as the donor's present location, blood type, and the date of their most recent donation, among other things. The right donors will be cleverly informed of the demand for blood donations, making it easier to locate a local suitable donor at the right time.

3.4 Problem Solution Fit

Define customer segments, fit into customer limitations

1. CUSTOMER SEGMENT(S) [Add]

1. Patient(plasma donor needy)

2. Plasma donors(age - 18 above)

3. Hospital management

6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES [Add]

1. Smartphones/laptop/desktop

2. Good Wi-Fi speed

3. Shortage of plasma

5. AVAILABLE SOLUTIONS PLUSES & MINUSES [Add]

1. Way more accurate/speed - Existing Application(not upto-date

2. Queries can be send through email - Late response

3. plasma availability to needy - not upto date

Focus on problem, tap into behavior, understand root cause

2. PROBLEMS / PAINS + ITS FREQUENCY [Add]

1. Difficult to find donors at the right time   Unpredictable

9. ROOT / CAUSE OF PROBLEM [Add]

1. The root cause of the problem is not   Unpredictable

finding donors at the emergency time

7. BEHAVIOR + ITS INTENSITY [Add]

1. Queries can be sent through email.   Unpredictable

2. Can be directly connected to donor.   Emergency

Identify strong triggers & emotions

3. TRIGGERS [Add]

1. The fact that triggers the Customers is people are dying

without getting plasma in their emergency time and its also

social responsibility to donate plasma.

4. EMOTIONS [Add]

Before:

1. User find hard to get a right resource to donate their plasma.

After:

2. Notify the donor immediately and customer get satisfied.

10. YOUR SOLUTION [Add]

1. Website is transparent about all the tieups with hospitals.

2. E-certificates will be provided for their good deed of donation.

3. Details regarding camps, also about the location of the events

4. An email message will be issued after registration is complete

5.The users can choose a home sample collection as well.

6. We have chatbots to answer all queries of the donors or users

and make sure they are comfortable with the process.

8. CHANNELS OF BEHAVIOR

ONLINE [Add]

1. User can find plasma donors.

2. Get details regarding the camps.

OFFLINE [Add]

1. Can use the record maintained by the hospital.

2. People can consult with donors before donating plasma.

3.5 SOLUTION ARCHITECTURE :

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	Statistical data	Plasma availability is provided on the page as statistics that will be useful for users.
FR-4	User Plasma Request	The recipient who needs plasma can fill the request form in the web page. The confirmation mail has been sent when the request is submitted.
FR-5	Donor Registration	The user who wants to donate plasma can fill the donor registration form in the web page. The confirmation mail has been sent when the form is submitted.
FR-6	Virtual Assistants	A virtual assistant is created to answer user questions about Plasma Donation. This will perform the function of a person in responding to user queries, where it will respond based on the information stored.
FR-7	User logout	After logging in to the application a user can navigate the login dashboard and can logout from the page by clicking logout button at the bottom of the page.

4.2 Non Functional Requirements

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

FR No.	Non-Functional Requirement	Description

NFR-1	Usability	User friendly interface with easily accessible, well-looking and interactive chatbots.
NFR-2	Security	Data of donor and recipient should be saved in a secured manner.The user can only logged in using the correct password and username.
NFR-3	Reliability	The system should be built in such a way that it is reliable in its operations as well as to secure the sensitive details.
NFR-4	Performance	Users should have a proper internet connection.
NFR-5	Availability	The system should have efficient active service. Must be available all times. Incase of hardware or database corruption, backups of the data should be retrieved from the web application.
NFR-6	Scalability	The system should be scalable to handle a large number of users and should not get disrupted while using the system application.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

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

Flow:
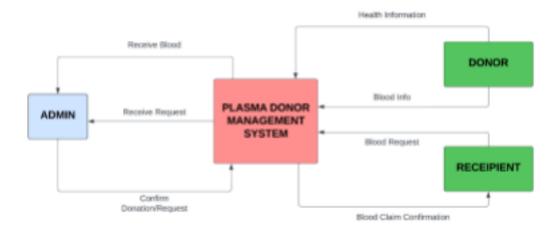
1. Donor / Recipient can register by entering their details
2. Already registered user can log in using their credentials
3. Users can register for donation or can create a request for plasma
4. All the details are stored in the IBM Database
5. The server provides the information of Plasma availability
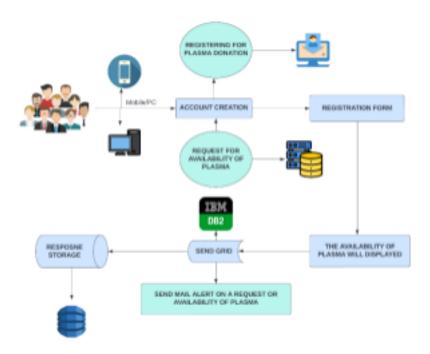
6. Users booking can be verified by sending Emails or Messages

Data Flow for Plasma Donor Application:
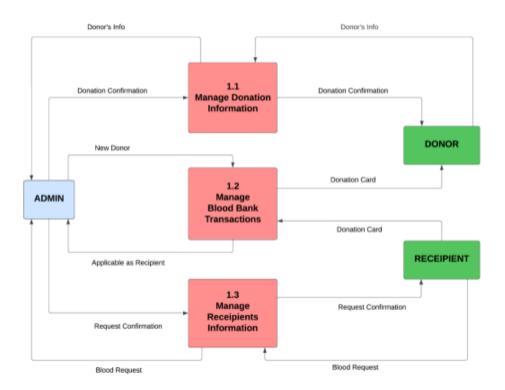
Example: DFD Level 0 (Industry Standard)

DATA FLOW DIAGRAM LEVEL 1

Donor's Info   Donor's Info

1.1 Manage Donation Information

Donation Confirmation   Donation Confirmation

DONOR

New Donor

ADMIN

1.2 Manage Blood Bank Transactions

Donation Card

Applicable as Recipient

Donation Card

RECEIPIENT

1.3 Manage Receipients Information

Request Confirmation

Request Confirmation

Blood Request   Blood Request

Manage Plasma Transaction

User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can receive confirmation email &click confirm	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can enter into my account	High	Sprint-1
	Dashboard	USN-6	As a user ,Display all details about plasma application	I can donate/get details about the plasma	High	Sprint-2

Customer (Web user)	Application	USN-7	As a user ,I can register, login and see details about plasma	I can access the donor details and availability of plasma	High	Sprint-3
Customer Care Executive	Update Plasma storage	USN-8	Keep track the availability of the Plasma	I can provide application for customer needs	High	Sprint-4
Administrator	Verify donor details	USN-9	To add the donor plasma details in application	I can Control the all details in this application	Medium	Sprint-3
Customer Care Executive	Verify Customer Feedback	USN-10	To design the application that meets user's desires	I can satisfy the customer expectations	Medium	Sprint-4
Customer Care Executive	Control all Plasma details	USN-11	Make sure to check the availability of plasma in application	I can alert notification through email and SMS	High	Sprint-2
Administrator	Performance of application	USN-12	To make the process more efficient	I can save time, cost by improving the Plasma management application	High	Sprint-4

5.2 Solution and Technical Architecture

5.2.1 TECHNOLOGY ARCHITECTURE

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

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

Technology Architecture Diagram
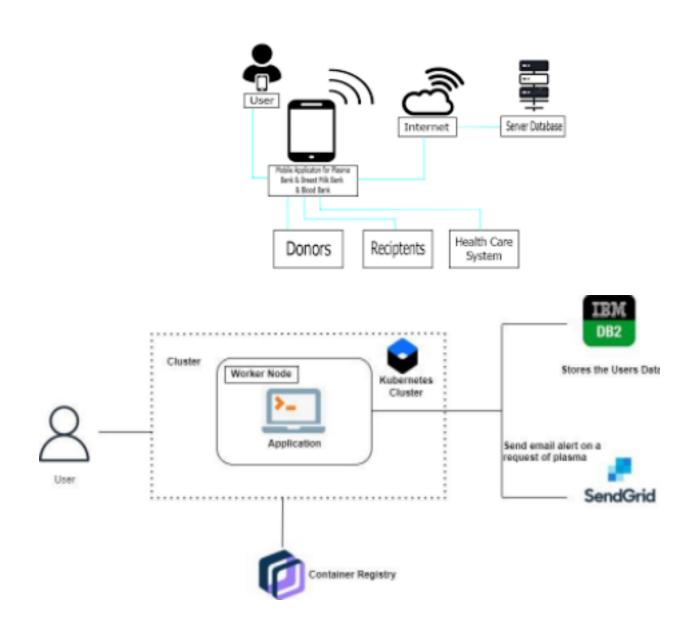
Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g.Web UI, MobileApp, Chatbot etc.	HTML, CSS

2.	Application Logic 1	NewUser registers in the application by giving the genuine contact details which will be stored in the database.	Flask,HTML,CSS
3.	Application Logic 2	Users login into the application by providing the username and password.	Flask,IBM DB2
4.	Application Logic 3	Stats page displays the blood unit count available and the number of donors available for each blood group	IBM Watson Assistant
5.	Application Logic 4	A request page that collects the name,contact number,gender and the blood group needed.Finally The request is sent to a donor whose blood group matches with the request.	Sendgrid
6.	Database	Characters,Integers,String,Long, Configurations	IBM DB2, MySQL
7.	Cloud Storage	Database service on cloud	IBM DB2, IBM Block Storage Or Other Storage Service or Local Filesystem
8.	External API-1	Authentication, used to store, manage and deploy container images.	Flask, Container registry
9.	External API-2	Sending request to donors	Sendgrid
10.	Infrastructure (Server / Cloud)	Application Deployment	Kubernetes, cloudfoundry

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology

1.	Open-Source Frameworks	List the open-source frameworks used	Python Flask
2.		plementations List all the ity / access controls nted,use of firewalls etc.	Doctor content Trust (DCT), Transport Layer Security(TLS), Container registry
3.	Scalable Architecture	Justifying the scalability of architecture (3 – tier, Micro-services) Kubernetes prevents hardware problems like downtime error.	Docker, Kubernetes cluster
4.	Availability	Use of load balancers, distributed servers. Kubernetes provide all time availability.	Kubernetes
5.	Performance	Application performance is improved by Docker	Docker

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Donor / Recipent / Hospital In-Charge (Mobile/Desktop opuser)	App Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint 1

	Login	USN-2	As a user,I will sign in to the application using my password and username.	I can receive confirmation email & click confirm	High	Sprint 1
	Register for donate	USN-3	As a user, I can sign in to the application and fill the plasma donation form. The booking can be confirmed by receiving email.	I can register to access the dashboard with Facebook Login	Low	Sprint 2
Patient/doctor	Find the bank	USN-4	As a user,I can register for the application and can find the available bank nearby.	I can access my account and dashboard	Medium	Sprint 1
	Request for plasma	USN-5	As a user, I can signinto the application by entering email & password and register the plasma request form in case of emergency.	I can register to access the dashboard with facebook login	High	Sprint 1
Administrator	Maintain the applications	USN-6	As an administrator I will provide the necessary details to the System	I can Access my account/dash board	High	Sprint 3

			application.			
	Connect the Bank with the users	USN-7	As an administrator, I will provide corrective and efficient communication between the bank and the user.	I can access my account / dashboard	Low	Sprint 4

	Maintain the database	USN-8	As an administrator, I will collect all the required data information of donors,recipients, banksand store thosedata information in a secured way.	I can access my account / dashboard	Medium	Sprint 4
Plasma Bank	Connect the Bank with users	USN-7	As a bank, I provide Good connection with users by providing The required help in emergency situations.	I can access my account / dashboard	Medium	Sprint 3

	Maintain the database	USN-8	As a bank, I will maintain the hospital and plasma Bank information for users,to access it for their required needs	I can access my account/ dashboard	High	Sprint 4
BOT	Help the users by using bot	USN-9	As a bot, I will provide interactive communication with the user and provide the information they need..	I can access my account/ dashboard	Medium	Sprint-4

6. PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning and Estimation
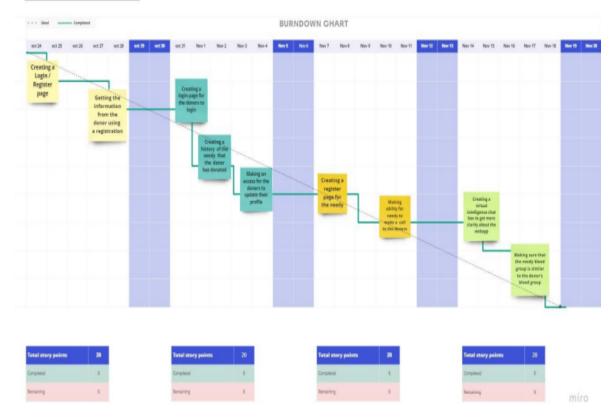
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the web app by entering my email, password, and confirming my password.	20	High	1.R.Sruthi 2.G.S.Urmikha 3.M.Susanthika 4.S.Sneha
Sprint-2	Login	USN-2	As a user,I can log into the application by entering email & password.	20	High	1.R.Sruthi 2.G.S.Urmikha 3.M.Susanthika 4.S.Sneha
Sprint-3	Donor Information	USN-3	Donors can update their personal information eg:blood group,.	20	High	1.R.Sruthi 2.G.S.Urmikha 3.M.Susanthika 4.S.Sneha

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-4	Finding the donor	USN-4	The patient can find the donor with their similar blood group	20	High	1.R.Sruthi 2.G.S.Urmikha 3.M.Susanthika 4.S.Sneha
Sprint-5	Chatbox	USN-5	The user can directly talk to Chatbot regarding the donor information and any queries.	20	High	1.R.Sruthi 2.G.S.Urmikha 3.M.Susanthika 4.S.Sneha

6.2.Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Sprint Release (Actual release)
Sprint-1	8	5 Days	27 Oct 2022	31 Oct 2022	30 Oct 2022
Sprint-2	13	6 Days	1 Nov 2022	06 Nov 2022	05 Nov 2022
Sprint-3	12	6 Days	07 Nov 2022	12 Nov 2022	11 Nov 2022
Sprint-4	11	6 Days	14 Nov 2022	19 Nov 2022	13 Nov 2022
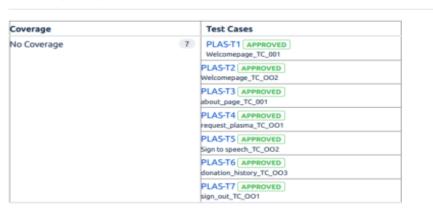
BURNDOWN CHART
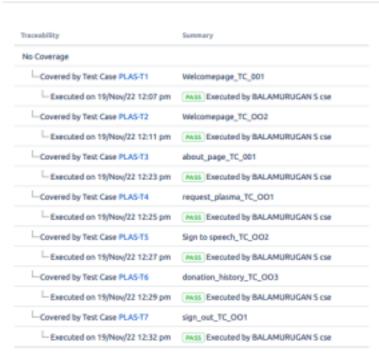
6.3. Reports from JIRA

COVERAGE REPORT

Traceability Tree

Traceability	Summary
No Coverage	
└ Covered by Test Case PLAS-T1	Welcomepage_TC_001
└ Executed on 19/Nov/22 12:07 pm	[PASS] Executed by BALAMURUGAN S cse
└ Covered by Test Case PLAS-T2	Welcomepage_TC_OO2
└ Executed on 19/Nov/22 12:11 pm	[PASS] Executed by BALAMURUGAN S cse
└ Covered by Test Case PLAS-T3	about_page_TC_001
└ Executed on 19/Nov/22 12:23 pm	[PASS] Executed by BALAMURUGAN S cse
└ Covered by Test Case PLAS-T4	request_plasma_TC_OO1
└ Executed on 19/Nov/22 12:25 pm	[PASS] Executed by BALAMURUGAN S cse
└ Covered by Test Case PLAS-T5	Sign to speech_TC_OO2
└ Executed on 19/Nov/22 12:27 pm	[PASS] Executed by BALAMURUGAN S cse
└ Covered by Test Case PLAS-T6	donation_history_TC_OO3
└ Executed on 19/Nov/22 12:29 pm	[PASS] Executed by BALAMURUGAN S cse
└ Covered by Test Case PLAS-T7	sign_out_TC_OO1
└ Executed on 19/Nov/22 12:32 pm	[PASS] Executed by BALAMURUGAN S cse

Traceability Matrix Report

Traceability matrix

	Test Cases	PLAS-T1 - Welcomepage_TC_001	PLAS-T2 - Welcomepage_TC_...	PLAS-T3 - about_page_TC_001	PLAS-T4 - request_plasma_TC_...	PLAS-T5 - Sign to speech_TC_...	PLAS-T6 - donation_history_T...	PLAS-T7 - sign_out_TC_OO1
Coverage								
No Coverage		🟩	🟩	🟩	🟩	🟩	🟩	🟩
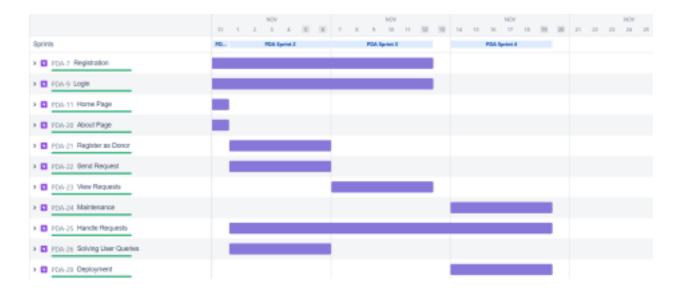
Displaying (1 of 1)

Last test execution: 🟩 Pass

7. CODING AND SOLUTIONING

7.1 Layer between Donor and Recipient

The application servers as a layer between donor and the recipient, donor can expose their information to the application, and they will be notified via email if there's a request for plasma with the same blood group

7.2 Chatbot integration

The application has a chatbot integrated with it to help with basic user queries and to interact with the user. The chatbot feature is added to the application by using IBM Watson assistant in IBM cloud. This chatbot can interact with the user and guide them for simple queries.

7.3. Feature 1

SENDGRID

Sendgrid service integrate in minutes with our email API and trust your emails reach the inbox

Sendgrid Integration

```python
# sendgrid integration
def mailtest_registration(to_email):
    sg = sendgrid.SendGridAPIClient(api_key= 'apikey' )
```

```python
    from_email = Email("susanthika02m@gmail.com")
    subject = "Registration Successfull!"
    content = Content("text/plain", "You have successfully registered as a user. Please Login using
your Username and Password to donate/request for Plasma.")
    mail = Mail(from_email, to_email, subject, content)
    response = sg.client.mail.send.post(request_body=mail.get())
    print(response.status_code)
    print(response.body)
    print(response.headers)
#for donor
def mailtest_donor(to_email):
    sg = sendgrid.SendGridAPIClient(api_key= 'apikey' )
    from_email = Email("susanthika02m@gmail.com")
    subject = "Thankyou for Registering as Donor!"
    content = Content("text/plain", "Every donor is an asset to the nation who saves people's lives, and
you're one of them.We appreciate your efforts. Thank you!!")
    mail = Mail(from_email, to_email, subject, content)
    response = sg.client.mail.send.post(request_body=mail.get())
    print(response.status_code)
    print(response.body)
    print(response.headers)
#for request
def mailtest_request(to_email):
    sg = sendgrid.SendGridAPIClient(api_key= 'apikey' )
    from_email = Email("susanthika02m@gmail.com")
    subject = "Request Submitted!"
    content = Content("text/plain", "Your request has been successfully submitted. Please be patient
, your requested donor will get back to you soon.")
    mail = Mail(from_email, to_email, subject, content)
    response = sg.client.mail.send.post(request_body=mail.get())
    print(response.status_code)
    print(response.body)
    print(response.headers)
#for request sending to donor
def mailtest_requesttodonor(to_email):
    sg = sendgrid.SendGridAPIClient(api_key= 'apikey' )
    from_email = Email("susanthika02m@gmail.com")
```
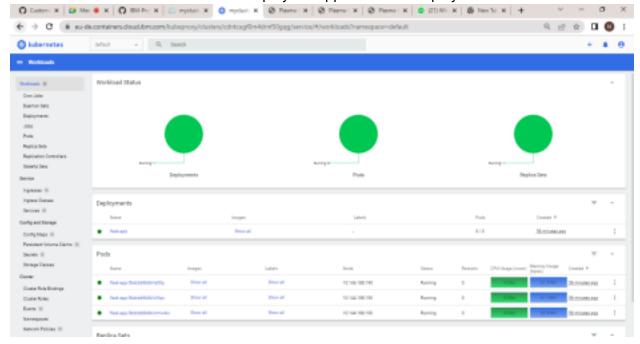
```
subject = "Requesting Plasma"     mail = Mail(from_email, to_email, subject, content)
response = sg.client.mail.send.post(request_body=mail.get())
print(response.status_code)
content = Content("text/plain", "Your registration has been requested by a recipient, we will share
futher details in future. Stay connected!!")


print(response.body)
print(response.headers)
```
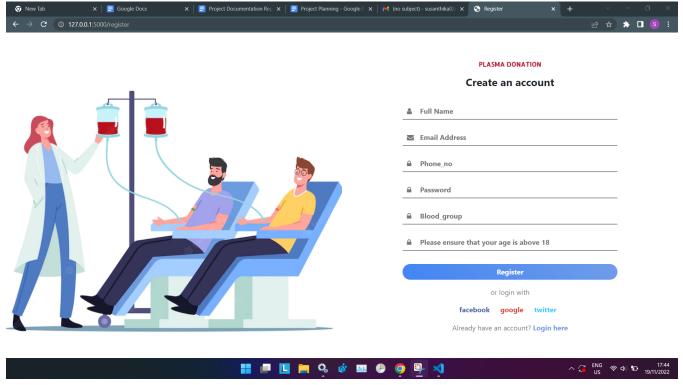
7.2. Feature 2

KUBERNETES

      Kubernetes has been used to deploy the application we deployed to the IBM Cloud
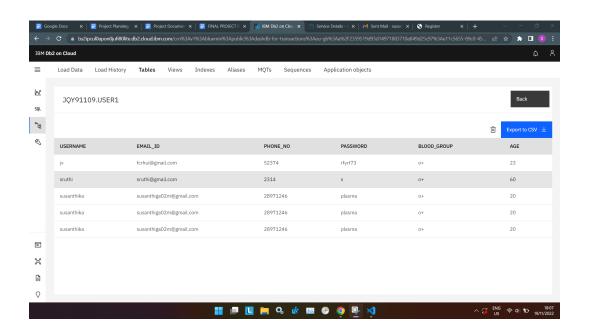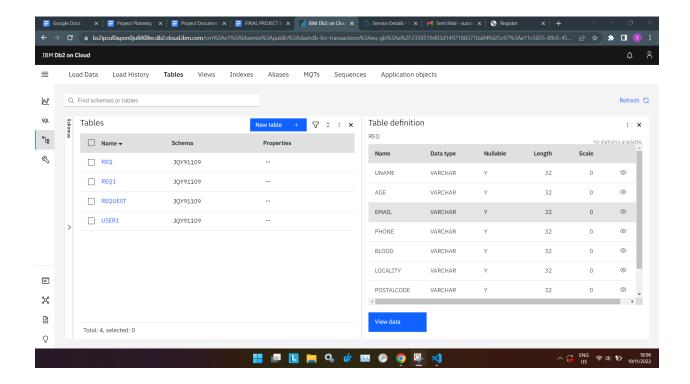
7.3 Feature 3

DATABASE

IBM Cloud Database helps to integrate data from different sources across on-premises and cloud environments.

8. TESTING

8.1 Test Cases

Test case ID	Test Scenario	Test Data	Expected Result	Actual Result	Status
TC_001	Verify user is able to see the Login/Signup popup when user clicked on Login or Register button	http://169.51.20 4.24:32734/	Login/Signup popup should display and the user must be able to switch between the pages with a single click	Working as expected	Pass

TC_002	Verify the UI elements are responsive when changing the window size	http://169.51.20 4.24:32734/	Application should re-align the image and text according to the new window size and should be **r**esponsive	Working as expected	Pass
TC_003	Verify that all the fields such as Username, Mobile Number, Password and Email have a valid placeholder	**Placeholders - Registration Page** Enter your UserName Enter your Email Enter your mobile number Create a Password **Placeholders - Login** Enter UserName Enter Password	Placeholders must be visible	Working as expected	Pass
TC_004	If a user tries to register then he/she must fill all the required fields	**Form Details** Your Name - Sruthi	Application should show 'Please fill this	Working as expected	Pass

		Your Email - sruthi@gmai l.com Phone Your Password - Sruthitest1	field ' validation message.		

TC_005	If a user tries to register then he/she must fill a valid Email address in the Your Email field.Filling string without an @ symbol will throw an error.	**Form Details** Your Name - sruthi Your Email - sruthi@ Phone 9940282506 Your Password - Sruthitest1	Application should show 'Please enter a part following sruthi@ ' validation message.	Working as expected	Pass
TC_006	Verify user is able to log into application with Valid credentials	**Username:** sneha **password:** Testing12367 86 86786876876	Application should login successfully	Working as expected	Pass
TC_007	Verify user is able to log into application with InValid credentials	**Username:** sneha **password:** Testing12367 86 86786876876 @	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass
TC_008	Verify if the correct username is being displayed beside the Welcome Section	Username: susanthika Password: susanthika@12 34	The page should show " Welcome: susanthika!!"	Working as expected	Pass
TC_009	Verify the Donate Plasma and Request Plasma links	**Username**: urmikha **Password**: urmikha@01	Clicking on Donate Plasma should take the user to the donor registration page and clicking on request plasma should	Working as expected	Pass

			take the user to the donor list page		
TC_0010	Verify if the submission in Donating Plasma Page is successful	**Mail:** sneha@gmail.com	After filling out the register as donor page and clicking submit application should redirect to a "registration success" page	Working as expected	Pass
TC_0011	Verify if the user received mail after successful registration	**Your Name** - sruthi**Your Email** - sruthi@gmail.com **Phone** 9940282506 **Your Password** - Sruthitest3212	After filling out the registration page and submitting the user should receive a "Registration Success!!" Mail on their registered Email Id.	Working as expected	Pass
TC_0012	Verify if the user received mail after successfully registering as a donor in Plasma registration Page	**Mail:** susanthika02@gmail.com	After filling out the registration page and submitting the user should receive a "Registration Success!!" Mail on their registered Email Id.	Working as expected	Pass

TC_0013	Verify if the user(Plasma Recepient) recieved mail after successfully requesting for plasma	**Mail:** urmikha234@gmail.com	After filling out the request plasma page the user(Plasma Recipient) receives a mail that the request	Working as expected	Pass

			has been successfully posted		
TC_0014	Verify if the user(Plasma Donor) recieved mail when a Recepient makes a request for their plasma through the application	**Mail:** urmikha234@gmail.com	When a Plasma Recepient fills out the request plasma page the Plasma Donor sould receive a mail that a Recipient has made a Request to them.	Working as expected	Pass
TC_0015	Verify if ChatBot is working properly and deployed universally throughout the application	http://169.51.20 4.24:32734/	ChatBot should be accessable inside any webpage such as Login, Home or Register pages and must answe the user queries.	Working as expected	Pass

| TC_OO16 | Verify if the user is able to logout from the login dashboard | **Mail:** sneha@gmail. com | Clicking on logout should redirect the user to home page. | Working as expected | Pass |
| TC_OO17 | Verify if the recipient should able to view the available donor list | **Username**: susanthika **Password**: 1254at | Clicking on the request for plasma button, the recipient should be able to view the available donors list | Working as expected | Pass |

8.2 User Acceptance Testing

The test coverage and open issues of the Plasma Donor Application project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
Flask	2	2	0	0	4
Cloud account creation	2	1	1	0	3
Connecting with Db2	4	3	1	0	8
Sendgrid	2	3	0	1	6
Docker	2	1	0	0	3
Totals	12	10	2	1	25

Test Case Analysis

Section	Total Cases	Not Tested	Fail	Pass
Home Page	5	0	0	5
Login Page	5	0	0	5
Register Page	7	0	0	7
Login Dashboard	5	0	0	5
Donating Plasma Page	8	0	0	8
Request Plasma Page	8	0	0	8
Chatbot	2	0	0	2
Donor list	6	0	0	6

33

9. RESULT

9.1 Performance Metrics

10. ADVANTAGES AND DISADVANTAGES

10.1 Advantages

The main advantage is that it is a relatively simple way to collect data from many people quickly and at zero cost.

Good Validity - people can fulfill and request their needs directly .

A second advantage is that data can be collected in various ways to suit the researcher's needs.

The application has the ability to collect data from a large number of people and store it in the database.

It helps people to help others who have medical needs.

It is a relatively safe process.

10.2 Disadvantages

The main disadvantage is that questionnaires might be the possibility of providing invalid answers.Fixed choice questions lack flexibility.

There is a chance that some questions will be ignored or left unanswered. Self-reported answers may be exaggerated; respondents may be too embarrassed to reveal private details.

Low response rate.

11. CONCLUSION

PLASMA DONOR APPLICATION this project "PLASMA DONOR" deals with notifying the concerned donor upon request by the Recipient in need of Plasma. This project provides quick access to donors for an immediate requirement of blood. In case of an emergency/surgery, blood procurement is always a major problem which consumes a lot of time. This helps serve the major time-lapse in which a life can be saved!

12. FUTURE SCOPE

The Plasma Donation App would help Donors, as well as patients in need of plasma.It would allow you to search Plasma Donors within your city and having a specific Blood Group. People who have fully recovered from COVID-19 have antibodies in their plasma that can attack the virus.The proposed plasma Donating Web Application project could ensure the necessity of plasma and plasma donation by saving the World.

13. APPENDIX

Source Code

-> login.html

```html
<!DOCTYPE html>

<html>

<head>

    <title>Login</title>
```

```html
    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">

    <link href="static/css/style.css" rel="stylesheet" type="text/css">

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
>

    <link
href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.mi
n.css" rel="stylesheet">

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

</head>

<body>

    <div class="container-fluid">

        <div class="row">

            <!-- IMAGE CONTAINER BEGIN -->

            <div class="col-lg-6 col-md-6 d-none d-md-block
infinity-image-container"></div>

            <!-- IMAGE CONTAINER END -->
```

```html
<!-- FORM CONTAINER BEGIN -->

<div class="col-lg-6 col-md-6 infinity-form-container">

    <div class="col-lg-9 col-md-12 col-sm-8 col-xs-12
infinity-form">

        <!-- Company Logo -->

        <div class="text-center mb-3 mt-5">

            <img src="static/css/logo.png" width="150px">

        </div>

        <div class="text-center mb-4">

    <h4>Login into account</h4>

  </div>

        <!-- Form -->

        <form class="px-3" action="/login" method="post">

            <!-- Input Box -->

            <div class="form-input">

                <span><i class="fa fa-user"></i></span>

                <input type="username" name="username"
placeholder="username" tabindex="10"required>
```

```html
                </div>

                <div class="form-input">

                    <span><i class="fa fa-lock"></i></span>

                    <input type="password" name="password"
placeholder="Password" required>

                </div>

                <div class="row mb-3">

                    <!--Remember Checkbox -->

                <div class="col-auto d-flex align-items-center">

                    <div class="custom-control custom-checkbox">

                        <input type="checkbox" class="custom-control-input"
id="cb1">

                        <label class="custom-control-label"
for="cb1">Remember me</label>

                    </div>

                </div>

            </div>

                <!-- Login Button -->
```

```html
<div class="mb-3">

    <button type="submit" class="btn btn-block"
href="welcome">Login</button>

    </div>

    <!-- Forget Password -->

    <div class="text-right">

    <a href="reset" class="forget-link">Forgot password?</a>

    </div>

    <div class="text-center mb-2">

    <div class="text-center mb-3" style="color: #777;">or login
with</div>

    <!-- Facebook Button -->

    <a href="" class="btn btn-social btn-facebook">facebook</a>

    <!-- Google Button -->

    <a href="" class="btn btn-social
btn-google">google</a>
```

```html
                        <!-- Twitter Button -->

                        <a href="" class="btn btn-social
btn-twitter">twitter</a>

                    </div>

                    <div class="text-center mb-5" style="color:
#777;">Don't have an account?

                        <a class="register-link" href="register">Register
here</a>

            </div>

                </form>

            </div>

            <!-- FORM END -->

        </div>

        <!-- FORM CONTAINER END -->

    </div>

    </div>

</body>
```

```
</html>
```

-> register.html

```html
<!DOCTYPE html>

<html>

<head>

    <title>Register</title>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">

    <link  href="{{ url_for('static', filename='css/style.css') }}"
rel="stylesheet" type="text/css">

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
>

    <link
href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.mi
n.css" rel="stylesheet">

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
```

```html
</head>

<body>

    <!-- Main Container -->

    <div class="container-fluid">

        <div class="row">

            <!-- IMAGE CONTAINER BEGIN -->

            <div class="col-lg-6 col-md-6 d-none d-md-block
infinity-image-container"></div>

            <!-- IMAGE CONTAINER END -->




            <!-- FORM CONTAINER BEGIN -->

            <div class="col-lg-6 col-md-6 infinity-form-container">

                <!-- FORM BEGIN -->

                <div class="col-lg-9 col-md-12 col-sm-8 col-xs-12
infinity-form">

                    <!-- Company Logo -->

                    <div class="text-center mb-3 mt-5">

                        <img src="static/css/logo.png" width="150px" >
```

```html
                    </div>

                    <div class="text-center mb-4">

                    <h4>Create an account</h4>

                </div>

                    <!-- Form -->

                    <form class="px-3" action="/register" method="POST">

                        <!-- Input Box -->

                        <div class="form-input">

                            <span><i class="fa fa-user"></i></span>

                            <input type="text" name="username"
placeholder="Full Name" tabindex="10"required>

                        </div>

                        <div class="form-input">

                            <span><i class="fa fa-envelope"></i></span>

                            <input type="email" name="email_id"
placeholder="Email Address" tabindex="10"required>

                        </div>

                        <div class="form-input">
```

```html
                        <span><i class="fa fa-lock"></i></span>

                        <input type="text" name="phone_no"
placeholder="Phone_no" required>

                    </div>

                    <div class="form-input">

                        <span><i class="fa fa-lock"></i></span>

                        <input type="password" name="password"
placeholder="Password" required>

                    </div>

                    <div class="form-input">

                        <span><i class="fa fa-lock"></i></span>

                        <input type="text" name="blood_group"
placeholder="Blood_group" required>

                    </div>

                    <div class="form-input">

                        <span><i class="fa fa-lock"></i></span>

                        <input type="age" name="age" placeholder="Please
ensure that your age is above 18 " required>

                    </div>
```

```html
<!-- Register Button -->

<div class="mb-3">

    <button type="submit" class="btn btn-block" href="login">Register</button>

</div>

<div class="text-center mb-2">

    <div class="text-center mb-3" style="color: #777;">or login with</div>


<!-- Facebook Button -->

<a href="" class="btn btn-social btn-facebook">facebook</a>


<!-- Google Button -->

    <a href="" class="btn btn-social btn-google">google</a>


<!-- Twitter Button -->

    <a href="" class="btn btn-social
```

```html
btn-twitter">twitter</a>

                                    </div>


                                    <div class="text-center mb-5" style="color:
#777;">Already have an account?

                                        <a class="login-link" href="{{ url_for('login')
}}">Login here</a>


                            </div>


                        </form>


                </div>


                <!-- FORM END -->


            </div>


            <!-- FORM CONTAINER END -->


        </div>


    </div>


</body>


</html>
```

-> reg.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
```

```html
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Request</title>
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.14.0/css/all.css">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<style>
    main{
        margin: 70px 0 0 ;
    }
</style>
<body>
    <header>
        <nav class="navbar navbar-dark navbar-expand-sm  fixed-top" style="background-color: rgb(203,
182, 117);">
            <a href="/welcome" class="navbar-brand">
                <i class="fas fa-medkit"></i>  
                Plasma Donor App
            </a>
        </nav>
    </header>
    <main>
        <div class="container-sm bg-light ">

            <h3 class="text-center">Request Form</h3>

        <form action="{{url_for('success')}}" method="post">
            <div class="form-group">
              <label for="recipient-name">Recipient Name</label>
              <input type="text" class="form-control" name="name"  placeholder="name" required>
            </div>
            <div class="form-group">
                <label for="r-age">Enter Recipient Age</label>
                <input type="age" class="form-control" name="age" id="age" placeholder="age" required>
            </div>
            <div class="form-group">
                <label for="email">Email address</label>
                <input type="email" class="form-control" name="email" placeholder="email" required>
             </div>
              <div class="form-group">
                <label for="phone">Phone No</label>
                <input type="text" class="form-control" name="phone" placeholder="Phone" required>
              </div>
              <div class="form-group">
                <label for="b-type">Choose Blood type</label>
                <select type ="text" id="blood-group" name="blood" class="form-control browser-default
custom-select" required>
                    <option value="A Positive">A postive (A+)</option>
                    <option value="A Negative">A Negative (A-)</option>
```

```html
                    <option value="B Positive">B postive (B+)</option>
                    <option value="B Negative">B Negative (B-)</option>
                    <option value="O Positive">O postive (O+)</option>
                    <option value="O Negative">O Negative (O-)</option>
                    <option value="AB Positive">AB postive (AB+)</option>
                    <option value="AB Negative">AB Negative (AB-)</option>
                </select>
            </div>
            <div class="form-group">
                <label for="locality">Locality</label>
                <input type="text"  required class="form-control" name="locality" id="location"
placeholder="locality">
            </div>
            <div class="form-group">
                <label for="postal-code">Postal Code</label>
                <input type="text" required class="form-control" name="postalcode"
placeholder="postalcode">
            </div>
            <div class="form-group">
                <label for="contact-address">
                    Enter you contact address
                </label>
                <textarea type="password"class="form-control" name="address" id="address" cols="4"
rows="3" required></textarea>
            </div>
            <button type="submit" class="btn btn-primary float-right" href="success">Submit</button>
        </form>
    </div>
</main>

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="reseanonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
integrity="sha384-B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV"
crossorigin="anonymous"></script>
</body>
</html>
```

-> reset.html

```html
<!DOCTYPE html>
<html>
<head>
    <title>Reset</title>
    <meta charset="utf-8">
```

```html
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link href="{{ url_for('static', filename='css/style.css') }}" rel="stylesheet" type="text/css">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
    <link href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
</head>
<body>
    <div class="container-fluid">
        <div class="row">
            <!-- IMAGE CONTAINER BEGIN -->
            <div class="col-lg-6 col-md-6 d-none d-md-block infinity-image-container"></div>
            <!-- IMAGE CONTAINER END -->

            <!-- FORM CONTAINER BEGIN -->
            <div class="col-lg-6 col-md-6 infinity-form-container">
                <div class="col-lg-9 col-md-12 col-sm-8 col-xs-12 infinity-form">
                    <div class="text-center mb-3 mt-5">
                        <img src="static/css/logo.png" width="150px">
                    </div>
                    <div class="reset-form d-block">
                        <form class="reset-password-form px-3">
                            <h4 class="mb-3">Reset Your password</h4>
                            <p class="mb-3" style="color: #777">
                                Please enter your email address and we will send you a password reset link.
                            </p>
                            <div class="form-input">
                                <span><i class="fa fa-envelope"></i></span>
                                <input type="email" name="email_id" placeholder="Email Address" tabindex="10"required>
                            </div>
```

```html
                <div class="mb-3">
                        <button type="submit" class="btn">Send Reset Link</button>
                    </div>
                </form>
            </div>

            <div class="reset-confirmation d-none px-3">
                <div class="mb-4">
                    <h4 class="mb-3">Link was sent</h4>
                    <h6 style="color: #777;">Please, check your inbox for a password reset link.</h6>
                </div>
                <a href="login">
                    <button type="submit" class="btn" href="welcome">Login Now</button>
                </a>
            </div>
        </div>
    </div>
    <!-- FORM CONTAINER END -->
    </div>
    </div>

<script type="text/javascript">
    function PasswordReset() {
        $('form.reset-password-form').on('submit', function(e) {
            e.preventDefault();
            $('.reset-form')
                .removeClass('d-block')
                .addClass('d-none');
            $('.reset-confirmation').addClass('d-block');
        });
    }

    window.addEventListener('load', function() {
        PasswordReset();
    });
</script>
```

```
<!--About-->
<br>
<h1 style="text-align: center; margin-top: 10px;">Know more about Plasma</h1>
<div class = "profile-area">
  <div class = "container">
   <div class="row">
    <div class = "col-12 col-md-6 col-lg-6">
     <div class = "card">
                      <div class="img1"><img src="https://objstorage.s3.jp-tok.cloud-object
storage.appdomain.cloud/color.PNG"></div>


       <div class = "main-text card-body">
        <h2 class="card-title">What is Plasma?</h2>
        <p class="card-body">Plasma is the pale yellow liquid part
             of whole blood, in which the cellular elements are suspended. It is enriched in
proteins that help fight infection and aid the blood in clotting. AB plasma is plasma collected
from blood group AB donors. It is considered "universal donor" plasma because it is suitable for
all recipients, regardless of blood group.</p>
        </div>
       </div>
      </div>


       <div class = "col-12 col-md-6 col-lg-6">
        <div class = "card">
                        <div class="img1"><img src="https://objstorage.s3.jp-tok.cloud-object
storage.appdomain.cloud/color.PNG"></div>
         <div class = "main-text card-body">
          <h2 class="card-title">What is Plasmapheresis?</h2>
           <p class="card-body">Plasmapheresis is the standard procedure by which plasma is
```

separated from whole blood and collected. Blood flows through a single needle placed in an arm vein, into a machine that contains a sterile, disposable plastic kit. The plasma is isolated and channeled out into a special bag, and red blood cells and other parts of the blood are

returned to you through the same needle.</p>
            </div>
          </div>
        </div>

        <div class = "col-12 col-md-6 col-lg-6">
          <div class = "card">
                            <div class="img1"><img src="https://objstorage.s3.jp-tok.cloud-object storage.appdomain.cloud/color.PNG"></div>
            <div class = "main-text card-body">
             <h2 class="card-title">Is Plasmapheresis Safe?</h2>
                  <p class="card-body">Absolutely. The machine and the procedure have been evaluated and approved by the Food and Drug Administration (FDA), and all plastics and needles coming into contact with you are used once and discarded. At no time during the procedure is the blood being returned to you detached from the needle in your arm, so there is no risk of returning the wrong blood to you.</p>
            </div>
          </div>
        </div>

        <div class = "col-12 col-md-6 col-lg-6">
          <div class = "card">
                            <div class="img1"><img src="https://objstorage.s3.jp-tok.cloud-object storage.appdomain.cloud/color.PNG"></div>
            <div class = "main-text card-body">
             <h2 class="card-title">How Long Does Plasmapheresis Take?</h2>
                  <p class="card-body">Plasmapheresis procedures take about 40 minutes, but you

should allow another 20 minutes for staff to obtain your medical history. Every effort will be made to make the experience relaxing and enjoyable.</p>

               </div>

             </div>

            </div>]

           <div class = "col-12 col-md-6 col-lg-6">

            <div class = "card">

       <div class="img1"><img src="https://objstorage.s3.jp-tok.cloud-object storage.appdomain.cloud/color.PNG"></div>

              <div class = "main-text card-body">

              <h2 class="card-title">How Do I Prepare to Donate Plasma?</h2>

                <p class="card-body">On the day of your plasma donation appointment, make sure that you get some rest and have a healthy breakfast. You should drink lots of fluids, but avoid coffee, tea, and alcohol, as these drinks actually dehydrate you. Opt for water or juice instead. You should not eat anything oily or greasy before donating plasma since this can affect the quality of your plasma.</p>

               </div>

              </div>

             </div>

           <div class = "col-12 col-md-6 col-lg-6">

            <div class = "card">

<div class="img1"><img src="https://objstorage.s3.jp-tok.cloud-object storage.appdomain.cloud/color.PNG"></div>

              <div class = "main-text card-body">

              <h2 class="card-title">Does donating plasma hurt?</h2>

                <p class="card-body">Donating plasma shouldn't hurt. Donating plasma should feel the same as a regular blood donation. You might feel a stinging sensation when the needle is inserted, but after that, the staff will do its best to make sure that you're comfortable throughout the donation process.</p>

              </div>

             </div>

            </div>

          <!---end of row--->

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


->welcome.html


<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home - Plasma Donor App</title>
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.14.0/css/all.css">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <link href="static/css/dash.css" rel="stylesheet" type="text/css">
</head>
<body>
    <header>
    <div class="header navbar-wrapper">
      <nav class="navbar navbar-dark " style="background-color: rgb(203, 182, 117)";>
        <div class="container">
        <a href="/about" class="navbar-brand">
        <i class="fas fa-medkit"></i>  
        Plasma Donor App
        </a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarCollapse">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div id="navbarCollapse" class="collapse navbar-collapse">
        <ul class="navbar-nav ml-auto">
          <li class="nav-item">
            <a class="nav-link active" href="{{ url_for('req') }}">
              Request
```

```html
            </a>
          </li>
          <li class="nav-item">
            <a class="nav-link active" href="{{ url_for('logout') }}">
              logout
            </a>
          </li>
        </ul>
      </div>
    </div>
</header>
<main>
  <div class="container">
        <div class="card glass-effect" style="background-color: yellowgreen" >
          <div class="card-body">
            <h4 class="text-content"><i class="fas fa-home"></i>
              <span class="pl-3">Welcome, {{username}}</span></h4>
          </div>
    </div>
    <div class="row">
      <div class="col-6 mt-3">
        <div class="card glass-effect" style="background-color: chocolate">
          <div class="card-body">
            <div class="card-title">
              <h3>Dashboard</h3>
            </div>
            <p class="card-text">
              In Dashboard, you can able to see the current statistics of the
              donors count, no of requests and availability of plasma in each blood type.
            </p>
          </div>
        </div>
      </div>
      <div class="col-3  mt-3  ">
        <div class="card bg-danger mb-3 ">
```

```html
<div class="card-body">
  <i class="fa fa-tint fa-lg"></i>
  <div class="card-title">
    <h3 class="text-center">No.of Donors</h3>
    <span class="card-text"><h4>14</h4></span>
  </div>
</div>
</div>
</div>
<div class="col-3 mt-3">
  <div class="card card bg-warning mb-3 ">

    <div class="card-body">
      <div class="card-title">
        <i class="fas fa-tty"></i>
        <h3 class="text-center">
          No. of Requests
        </h3>
        <span class="card-text"><h4>18</h4></span>
      </div>
    </div>
  </div>
</div>
</div>
<div class="card glass-effect">
  <div class="card-body">
    <h4 class="text-center">
      Plasma Availability for Every Blood type
    </h4>
  </div>
</div>
<div class="row">
  <div class="col-3 mt-3">
    <div class="card border-success ">
      <div class="card-body">
```

```
            <div class="card-title">A postive (A+)</div>
            <h3 class="card-text">{{c0}}</h3>
          </div>
        </div>
      </div>
      <div class="col-3 mt-3">
        <div class="card border-success ">
          <div class="card-body">
            <div class="card-title">A Negative (A-)</div>
            <h3 class="card-text">{{c1}}</h3>
          </div>
        </div>
      </div>
      <div class="col-3 mt-3">
        <div class="card border-success ">
          <div class="card-body">
            <div class="card-title">B positive (B+)</div>
            <h3 class="card-text">{{c2}}</h3>
          </div>
        </div>
      </div>
      <div class="col-3 mt-3">
        <div class="card border-success ">
          <div class="card-body">
            <div class="card-title">B postive (B-)</div>
            <h3 class="card-text">{{c3}}</h3>
          </div>
        </div>
      </div>
    </div>
    <div class="row">
      <div class="col-3 mt-3">
        <div class="card border-success ">
          <div class="card-body">
            <div class="card-title">O postive (O+)</div>
            <h3 class="card-text">{{c4}}</h3>
```

```html
              </div>
            </div>
          </div>
          <div class="col-3 mt-3">
            <div class="card border-success ">
              <div class="card-body">
                <div class="card-title">O Negative (O-)</div>
                <h3 class="card-text">{{c5}}</h3>
              </div>
            </div>
          </div>
          <div class="col-3 mt-3">
            <div class="card border-success ">
              <div class="card-body">
                <div class="card-title">AB positive (AB+)</div>
                <h3 class="card-text">{{c6}}</h3>
              </div>
            </div>
          </div>
          <div class="col-3 mt-3">
            <div class="card border-success ">
              <div class="card-body">
                <div class="card-title">AB postive (B-)</div>
                <h3 class="card-text">{{c7}}</h3>
              </div>
            </div>
          </div>
        </div>
      </div>
    </main>
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "0164d575-77ae-4972-a697-b837df8e45da", // The ID of this integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "44187f7c-8739-40b6-92d2-dfdeccb99ac4", // 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>
```

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>

    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
integrity="sha384-B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV"
crossorigin="anonymous"></script>

</body>

</html>

-> success.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Success</title>

    <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.14.0/css/all.css">

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
>

</head>

  style
```

```html
    main {
        margin-top: 70px;
    }
</style>
<body>
    <div class="nav">
        <nav class="navbar navbar-dark navbar-expand-sm bg-danger fixed-top">
            <a href="/" class="navbar-brand">
                <i class="fas fa-medkit"></i>  
                Plasma Donor App
            </a>
        </nav>
    </div>
    <main>
        <div class="col d-flex justify-content-center">
        <div class="card border-info mb-3 float-center" style="width: 20rem; height: 11rem;">
            <div class="card-body ">
                <h5 class="card-title">Success</h5>
                <h6 class="card-subtitle mb-2 text-muted">Your Request was successfully added and waiting to be approved by admin</h6>
                <div class="container mt-3 mb-3">
                </div>
                <a href="/welcome" class="card-link">Home</a>
            </div>
        </div>
        </div>
    </main>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
```

```html
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk
7lN" crossorigin="anonymous"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
integrity="sha384-B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV
+rV" crossorigin="anonymous"></script>
</body>
</html>
```

->app.py

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

import ibm_db

import re

import os

from dotenv import load_dotenv

import sendgrid

from sendgrid.helpers.mail import *

from sendgridmail import sendmail

load_dotenv()

app = Flask(__name__)

app.secret_key ='a'

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31198;SECURITY=SSL;SSLServerCertificate=Certificate.crt;UID=jqy91109;PWD=jKfOzK88qIqZnQQK","",")

print(conn)

sendgrid integration

def mailtest_registration(to_email):

   sg = sendgrid.SendGridAPIClient(api_key= 'apikey')

   from_email = Email("susanthika02m@gmail.com")

   subject = "Registration Successfull!"

   content = Content("text/plain", "You have successfully registered as user. Please Login using your Username and Password to donate/request for Plasma.")

```python
        mail = Mail(from_email, to_email, subject, content)
        response = sg.client.mail.send.post(request_body=mail.get())
        print(response.status_code)
        print(response.body)
        print(response.headers)
#for donor
def mailtest_donor(to_email):
        sg = sendgrid.SendGridAPIClient(api_key= 'apikey' )
        from_email = Email("susanthika02m@gmail.com")
        subject = "Thankyou for Registering as Donor!"
        content = Content("text/plain", "Every donor is an asset to the nation who saves people's lives, and
you're one of them.We appreciate your efforts. Thank you!!")
        mail = Mail(from_email, to_email, subject, content)
        response = sg.client.mail.send.post(request_body=mail.get())
        print(response.status_code)
        print(response.body)
        print(response.headers)


@app.route('/register',methods=['GET', 'POST'])
def register():
    msg = " "
    if request.method == 'POST':
        username = request.form['username']
        email_id = request.form['email_id']
        phone_no = request.form['phone_no']
        password = request.form['password']
        blood_group = request.form['blood_group']
        age = request.form['age']
        query = "SELECT * FROM USER1 WHERE username=?"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        account =ibm_db.fetch_assoc(stmt)
        if (account):
            msg = "Account already exists!"
```

```python
            return render_template('login.html', msg=msg)
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email_id):
            msg = "Invalid email addres"
        elif not re.match(r'[A-Za-z0-9+', username):
            msg = "Name must contain only characters and numbers"
        else:

            mailtest_registration(email)
            insert_sql= "INSERT INTO USER1 values(?,?,?,?,?,?)"
            stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(stmt, 1, username)
            ibm_db.bind_param(stmt, 2, email_id)
            ibm_db.bind_param(stmt, 3, phone_no)
            ibm_db.bind_param(stmt, 4, password)
            ibm_db.bind_param(stmt, 5, blood_group)
            ibm_db.bind_param(stmt, 6, age)
            ibm_db.execute(stmt)
            account = ibm_db.fetch_assoc(stmt)
            msg = 'You have successfully Logged In!!'
            return render_template('login.html', msg=msg)
    else:
        msg = 'PLEASE FILL OUT OF THE FORM'
        return render_template('register.html', msg=msg)


@app.route('/login',methods=['GET', 'POST'])
def login():
    global usermail,userpass,userid
    msg = " "
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        usermail = username
        userpass = password
        query = "SELECT * FROM USER1 WHERE username=? and password=?"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.bind_param(stmt, 1, username)
```

```python
        ibm_db.bind_param(stmt, 2, password)
        ibm_db.execute(stmt)
        account =ibm_db.fetch_assoc(stmt)
        if (account):
            session['loggedin'] = True
            session['id']= account['USERNAME']
            userid=account['USERNAME']
            session['username']=account['USERNAME']
            msg = "Login Successfully ! "
            return render_template('welcome.html', msg=msg)
        else:
            return render_template('login.html')
    else:
        msg = 'PLEASE FILL OUT OF THE FORM'
        return render_template('login.html', msg=msg)


@app.route('/welcome', methods=['GET', 'POST'])
def welcome():
    return render_template("welcome.html")
@app.route('/reset', methods=['GET', 'POST'])
def reset():
    sendemail('
    return render_template('reset.html')


@app.route('/req', methods = ['GET', 'POST'])
def req():
 msg=' '
 if request.method == 'POST':
  name = request.form['name']
  age = request.form['age']
  email = request.form['email']
  phone = request.form['phone']
  blood = request.form['blood']
  locality = request.form['locality']
  postalcode =request.form['postalcode']
  address = request.form['address']
```

```python
        print('age')
        insert_sql = "INSERT INTO REQ1
VALUES('name','age','email','phone','blood','locality','postalcode','address')"

        stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(stmt, 1, name)
        ibm_db.bind_param(stmt, 2, age)
        ibm_db.bind_param(stmt, 3, email)
        ibm_db.bind_param(stmt, 4, phone)
        ibm_db.bind_param(stmt, 5, blood)
        ibm_db.bind_param(stmt, 6, locality)
        ibm_db.bind_param(stmt, 7, postalcode)
        ibm_db.bind_param(stmt, 8, address)
        print('age')
        ibm_db.execute(stmt)
        #sendmail('Plasma donor App plasma request', email ,'Your request for plasma is received.')
        msg='success'
        return render_template('success.html', msg=msg)
    else:
        return render_template('req.html', msg=msg)
@app.route('/about', methods=['GET', 'POST'])
def about():
    return render_template("about.html")
@app.route('/success', methods=['GET', 'POST'])
def success():
    return render_template("success.html")
@app.route('/logout',methods=['GET', 'POST'])
def logout():
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return render_template('login.html')


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

```
    app.run(host='0.0.0.0')
```

Kubernetes

dashboard-adminuser.yaml

```yaml
apiVersion: v1
```

```yaml
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kubernetes-dashboard
---
apiVersion: v1
kind: Secret
metadata:
  name: admin-user-token
  namespace: kubernetes-dashboard
  annotations:
    kubernetes.io/service-account.name: admin-user
type: kubernetes.io/service-account-token ---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard
```

flask_deployment.yaml

```yaml
apiVersion: apps/v1
```

```yaml
kind: Deployment
metadata:
  name: flask-app
spec:
  replicas: 3
  selector:

    matchLabels:
      app: flask-app
  template:
    metadata:
      labels:
        app: flask-app
    spec:
    containers:
      - name: page
        image:susanthika02/flask2
        imagePullPolicy: Never
        ports:
          - containerPort: 5000
            protocol: TCP
```

flask_ingress.yaml

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: flask-app-ingress
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/ssl-redirect: "false"

spec:
  # ingressClassName: nginx
  rules:
    - http:
        paths:
          - backend:
```

```yaml
        service:
          name: flask-app-service
          port:
            number: 5000
      path: /

      pathType: Prefix
```

flask_service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: flask-app-service
spec:
 type: ClusterIP
 ports:
  - port: 5000
 selector:
   app: flask-app
```

ibm_deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask-app
spec:
 replicas: 3
 selector:
   matchLabels:
     app: flask-app
 template:
   metadata:
     labels:
       app: flask-app
   spec:
   containers:
```
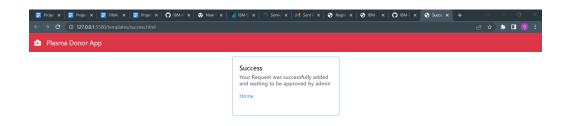
```yaml
- name: repo2
  image: icr.io/plasmadonorapp1/repo2
  imagePullPolicy: Always
  ports:
    - containerPort: 5000
      protocol: TCP
```
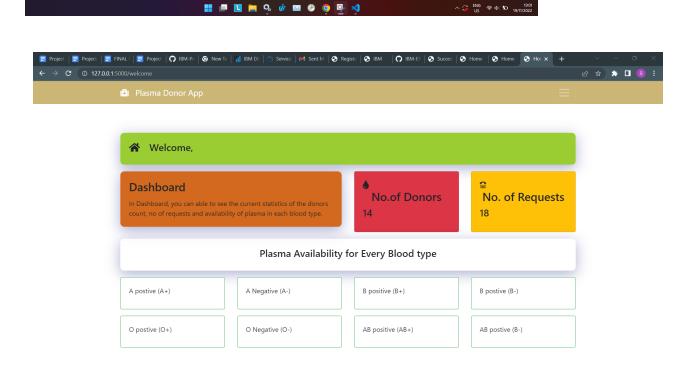
GITHUB AND PROJECT DEMO LINK:

Github link: https://github.com/IBM-EPBL/IBM-Project-30149-1660140825

Project demo link: Youtube link **https://youtu.be/ZxENAW9I5DE**\

https://drive.google.com/file/d/15i-CB2tDLIKlv_rXD_fMCW4u4XxPrikI/view?usp=sharing