PLASMA DONOR APPLICATION

IBM – DOCUMENTATION

UNDER THE GUIDANCE OF

INDUSTRY MENTOR	:	NAVYA
SPOC	:	Dr. G. PRADEEP
FACULTY EVALUTOR	:	Dr. J. SUDHA
FACULTY MENTOR	:	R. RAMYA

TEAM ID : PNT2022TMID46383

SUBMITTED BY

SAMYUKTHA G U	820319104034
KIRUTHIGA C	820319104021
PARVADHAVARTHINI S	820319104026
SRIMATHI S	820319104042

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A.V.C. COLLEGE OF ENGINEERING

ANNA UNIVERSITY :: 2019 – 2023

ABSTRACT

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. Lydia is thirty-five years old. She was affected by COVID-19 and had been admitted in the hospital. Due to COVID, her immunity power became low. She immediately wants a plasma donor. Here, she checks the availability of the plasma donor through our plasma donor application. She checks for the matching criteria of the donor through our application and also verifies whether the donor is within the range(location). Then after finding a perfect donor, he/she donates the plasma. Thus, our application is useful for a COVID patient to improve the immunity and blood circulation.

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Recently concern grows about the plasma donation for COVID-19 during the pandemic situation. This convalescent plasma was used to recover patients who are critically ill as it helps to grow antibodies on their body. Recent researches show that many people are willing to help someone in need through money, blood and plasma donation etc. but they find it difficult to identify and approach the needy people who are not aware of technological innovations, including the use of social media. Plasma is used to various infectious diseases and it is one of the oldest methods known as plasma therapy. Plasma therapy is a Process where blood is donated by recovered patients in order to establish anti bodies that fights the infection. This system comprises of Admin, user and donor where both can request for Plasma. The proposed method helps the users to check the availability of donors. A donor has to register to the website providing their details. The registered users can get the information about the donor count of each blood group. The database will have all the details such as name, email, phone number, infected status. Whenever a user requests for a particular blood group then the concerned blood group donors will receive the notification regarding the requirement. For instance, during COVID 19 crisis the requirement for plasma increased drastically as there were no vaccination found in order to treat the infected patients, with plasma therapy the recovery rates where high but the donor count was very low and in such situations it was very important to get the information about the plasma donors. Saving the donor information and notifying about the current donors would be a helping hand as it can save time and help the users to track down the necessary information about the donors.

1.2.PURPOSE

The main aim of developing this system is to provide blood to the people who are in need of plasma. The numbers of persons who are in need of plasma are increasing in large number day by day. Using this system user can search blood group available in the city and he can also get contact number of the donor who has the same blood group he/she needs for plasma. In order to help people who are in need of plasma, this plasma donor application can be used effectively for getting the details of available plasma and user can also get contact number of the plasma donors having the same blood group and with in the same city.

2. LITERATURE REVIEW

2.1. EXISTING PROBLEM

- Unable to find the exact donor for the recipient.
- Difficulty in transferring the blood plasma.
- Unable to reach the recipient's need.

i. TITLE : plasma donor application using AWS.

Plasma is a liquid portion of the blood, over 55% of human blood is plasma. Plasma is used to treat various infectious diseases and it is one of the oldest methods known as plasma therapy. Plasma therapy is a process where blood is donated by recovered patients in order to establish an antibody that fights the infection. In this project plasma donor application is being developed by using AWS services.The services used are AWS Lambda, API gateway, DynamoDB, AWS Elastic Compute Cloud with the help of these AWS services, it eliminates the need of configuring the servers and reduces the infrastructural costs associated with it and helps to achieve serverless computing. For instance,during COVID 19 crisis the requirement for plasma increased drastically as there were no vaccination found in order to treat the infected patients, with plasma therapy the recovery rates-where high but the donor count was very low and in such situations it was very important to get the information about the plasma donors. Saving the donor information and notifying about the current donors would be a helping hand as it can save time and help the users to track down the necessary information about the donors. The proposed method helps the users to check the availability of donors. A donor has to register to the website providing their details.The registered users can get the information about the donor count of each blood group. The database will have all the details such as name, email, phone number, infected status. Whenever a user requests for a particular blood group then the concerned blood group donors will receive the notification regarding the requirement. A Json code is written to store the information, to fetch the requestedinformation in lambda.

ii. TITLE : A Machine Learn in Approach for Nearest Blood & Plasma Donor Finding

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. Keywords—Blood donation, Plasma donation, K means clustering, Labeled Agglomerate clustering. Different methods have been used to solve this problem. This time, we have tried another way, a clustering approach, to solve the problem by grouping every user into small groups. This unsupervised machine learning approach is much faster and effective. In section II, we will discuss related work done previously to solve this problem. In section III, clustering algorithms relating to our project will explicitly be discussed. In section IV, our proposed method will be presented. In section V, we will analyze our experiment result.

2.2.REFERENCES

S. No	AUTHOR	PAPER TITLE	YEAR	JOURNAL	FINDINGS
1	Kalpana Devi Guntoju, Tejaswini Jalli, Sreeja Uppala, Sanjay Mallisetti	INSTANT PLASMA DONOR RECIPIENT CONNECTOR WEB APPLICATION	2022	International Research Journal of Modernization in Engineering Technology and Science	1. Acts as an interface betweenthe hospital and the donor. 2. Developed using HTML, CSS,PHP
2	Prof. Diksha Bhave, Shweta Badhe, Siddhi Jain, Aaditya Kasibhotla	BLOOD BANK MANAGE MENT SYSTEM	2019	Journal of Emerging Technologies and Innovative Research	1. HTML, CSS,PHP, SQL

3	Ms. Pradnya Jagtap, Ms. Monika Mandale, Ms. PrachiMhaske,Ms. Sonali Vidhate, Mr. Patil	IMPLEMENTATIONOF BLOOD DONATION APPLICATION USING ANDROID SMARTPHONE	2018	Open access international journal of science& engineering	1. Android based system. 2. Methodology Used Clustering, Text Mining, Pattern Matching, Support Vector Machine, Partitioning Algorithm and Donor HART Tool
4	Dr. S. Brindha, Ms. D. Priya, Mr. S. Ajith Kannan, Mr.D. Joyal Victor,Mr. R.Gunachandran	ENHANCED MOBILE APPLICATION DEVELOPMENT FORPLASMA, MOTHER'SMILK ANDBLOOD BANKS	2021	International Research Journal of Engineering and Technology(IRJET)	1. Enhance security 2. Information stored on the databases. 3. Uses the Firebase, 000 webhost. blogs (webCloud)
5	Saurin Parikh, Preeti Kathiria, Yashesh Vaghela, Harit Shah, Darshan Dholakiya	A Geo-Location basedMobile Service that Dynamically Locates and Notifies the nearest Blood Donors for Blood Donation during	2014	International Journal of Computer Applications	Geo-location. 1. Android based solution 2. SMS based HelpConfirmation 3. GPS based tracking

		Medical Emergencies			
6	M Sai Tarun , S Ravi kishan, Shaik Azaad Suraz Basha, Shaik Raj Ahamad, Chandrasekhar, Neha Bagga	Blood Bank Management System	2021	Journal of Emerging Technologies and Innovative Research	1. Front end is HTML, CSS, JavaScript Bootstrap, PHP. 2. For the back end, DBMS, MySQL is used. 3. SVM for data classification.
7	K M Akkas Ali, Israt Jahan, Md.Ariful Islam, Md. Shafa-at Parvez	Blood Donation Management System	2015	American Journal of Engineering Research	1. Designed usingASP.Net 2. SQL Server 2008 as database 3. SMS facility to donors
8	Nayan Das, MD AsifIqbal	Nearest Blood & PlasmaDonor Finding:A Machine Learning Approach	2020	23rd International Conference on Computer and Information Technology	1. Build a platform with clustering algorithms. 2. Hybrid approach of K-Means and Agglomerative clustering algorithm

9	Vamsi Krishna Tatikonda, Hosam El-Ocoa	BLOODR: blood donorand requester mobile application	2017	Mhealth 3	1. Ruby programming language along with JavaScript and PostgreSQLfor database are used.
10	Moh. Nabil, R. Ihab, H. ElMasry, S. Said	A Web-based blood donation and Medical Monitoring System Integrating Cloud services and Mobile Application	2020	Journal of Physics Conference Series	1. Java programming language using spring tool suite . 2. User interface is built usingSencha Ext JS java script framework. 3. Store application data on MySQL database which ishosted on Miles

2.3.PROBLEM STATEMENT DEFINITION

Recently concern grows about the plasma donation for COVID-19 during the pandemic situation. This convalescent plasma was used to recover patients who are critically ill as it helps to grow antibodies on their body. Recent researches show that many people are willing to help someone in need through money, blood and plasma donation etc. but they find it difficult to identify and approach the needy people who are not aware of technological innovations, including the use of social media. Plasma is used to various infectious diseases and it is one of the oldest methods known as plasma therapy. Plasma therapy is a Process where blood is donated by recovered patients in order to establish anti bodies that fights the infection. This system comprises of Admin, user and donor where both can request for Plasma. The proposed method helps the users to check the availability of donors. A donor has to register to the website providing their details. The registered users can get the information about the donor count of each blood group. The database will have all the details such as name, email, phone number, infected status. Whenever a user requests for a particular blood group then the concerned blood group donors will receive the notification regarding the requirement. For instance, during COVID 19 crisis the requirement for plasma increased drastically as there were no vaccination found in order to treat the infected patients, with plasma therapy the recovery rates where high but the donor count was very low and in such situations it was very important to get the information about the plasma donors. Saving the donor information and notifying about the current donors would be a helping hand as it can save time and help the users to track down the necessary information about the donors.

I am (USER)

A Covid affected patient

I am Trying To

Search Plasma donor

But

It takes long time to search

Because

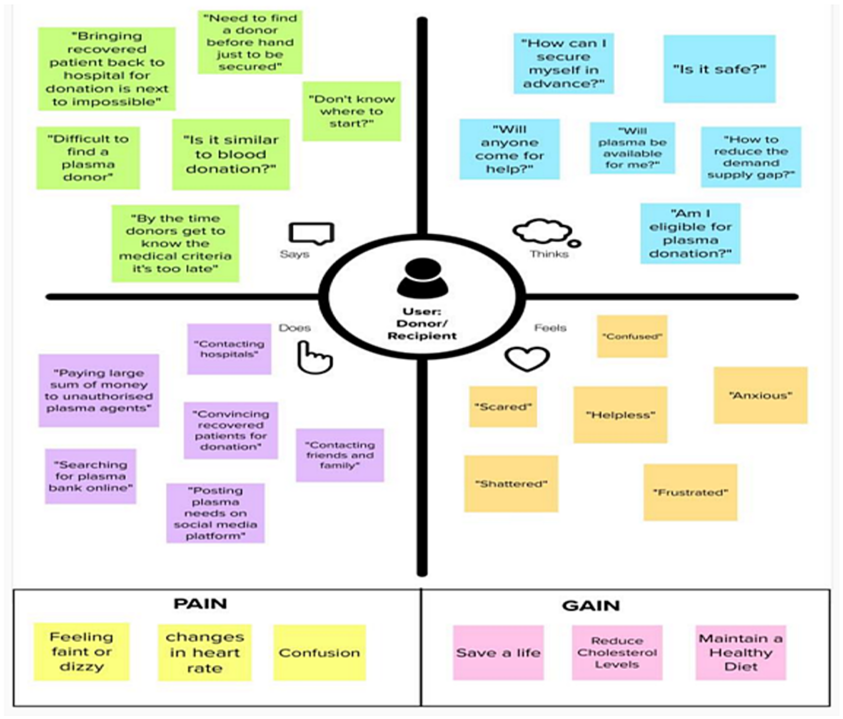
It is difficult to find plasma donor

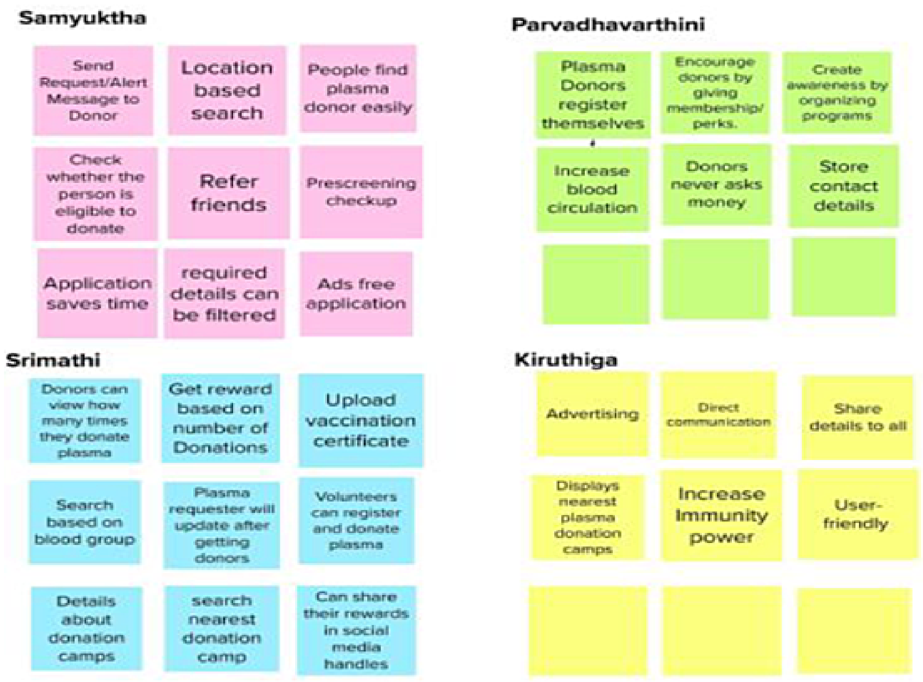
Which makes me feel?

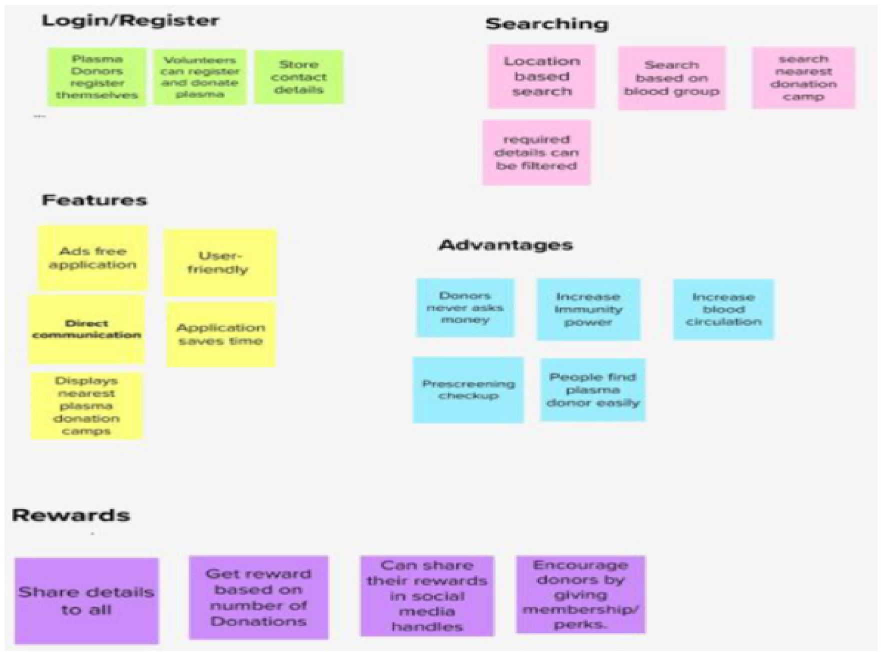
If the blood group is not available in the blood bank user can request the donor to donate the plasma to him and save someone life. Using this system people can register himself or herself who want to donate plasma. To register in the system they have to enter their contact information like address mobile number etc.

3.IDEATION & PROPOSED SOLUTION

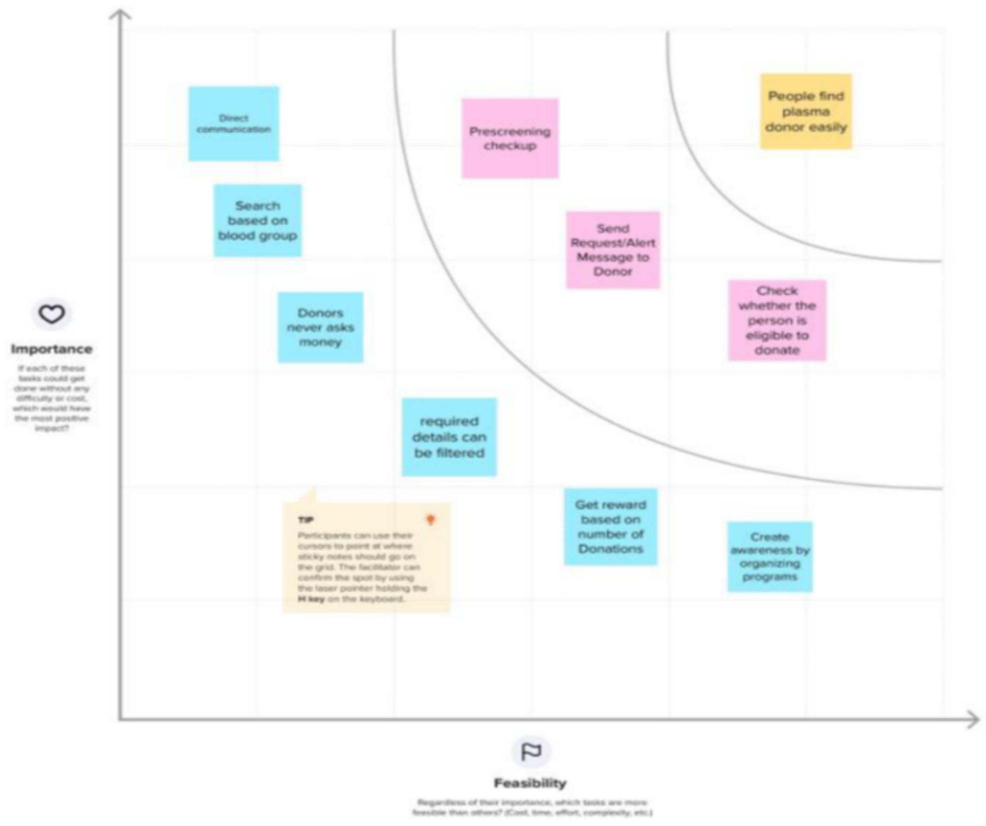
3.1.EMPATHY MAP CANVAS:


"Bringing recovered patient back to hospital for donation is next to impossible"
"Need to find a donor before hand just to be secured"
"Don't know where to start?"
"Difficult to find a plasma donor"
"Is it similar to blood donation?"
"By the time donors get to know the medical criteria it's too late"
Says
"How can I secure myself in advance?"
"Is it safe?"
"Will anyone come for help?"
"Will plasma be available for me?"
"How to reduce the demand supply gap?"
"Am I eligible for plasma donation?"
Thinks
User: Donor/ Recipient
Does
"Contacting hospitals"
"Paying large sum of money to unauthorised plasma agents"
"Convincing recovered patients for donation"
"Contacting friends and family"
"Searching for plasma bank online"
"Posting plasma needs on social media platform"
Feels
"Confused"
"Scared"
"Helpless"
"Anxious"
"Shattered"
"Frustrated"
PAIN
Feeling faint or dizzy
changes in heart rate
Confusion
GAIN
Save a life
Reduce Cholesterol Levels
Maintain a Healthy Diet


Samyuktha
Send Request/Alert Message to Donor
Location based search
People find plasma donor easily
Check whether the person is eligible to donate
Refer friends
Prescreening checkup
Application saves time
required details can be filtered
Ads free application
Parvadhavarthini
Plasma Donors register themselves
Encourage donors by giving membership/ perks.
Create awareness by organizing programs
Increase blood circulation
Donors never asks money
Store contact details
Srimathi
Donors can view how many times they donate plasma
Get reward based on number of Donations
Upload vaccination certificate
Search based on blood group
Plasma requester will update after getting donors
Volunteers can register and donate plasma
Details about donation camps
search nearest donation camp
Can share their rewards in social media handles
Kiruthiga
Advertising
Direct communication
Share details to all
Displays nearest plasma donation camps
Increase Immunity power
User-friendly


Login/Register
Plasma Donors register themselves
Volunteers can register and donate plasma
Store contact details
Searching
Location based search
Search based on blood group
search nearest donation camp
required details can be filtered
Features
Ads free application
User-friendly
Direct communication
Application saves time
Displays nearest plasma donation camps
Advantages
Donors never asks money
Increase Immunity power
Increase blood circulation
Prescreening checkup
People find plasma donor easily
Rewards
Share details to all
Get reward based on number of Donations
Can share their rewards in social media handles
Encourage donors by giving membership/ perks.

3.2.IDEATION & BRAINSTROMING:


Direct communication
Prescreening checkup
People find plasma donor easily
Search based on blood group
Send Request/Alert Message to Donor
Check whether the person is eligible to donate
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Donors never asks money
required details can be filtered
TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.
Get reward based on number of Donations
Create awareness by organizing programs
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

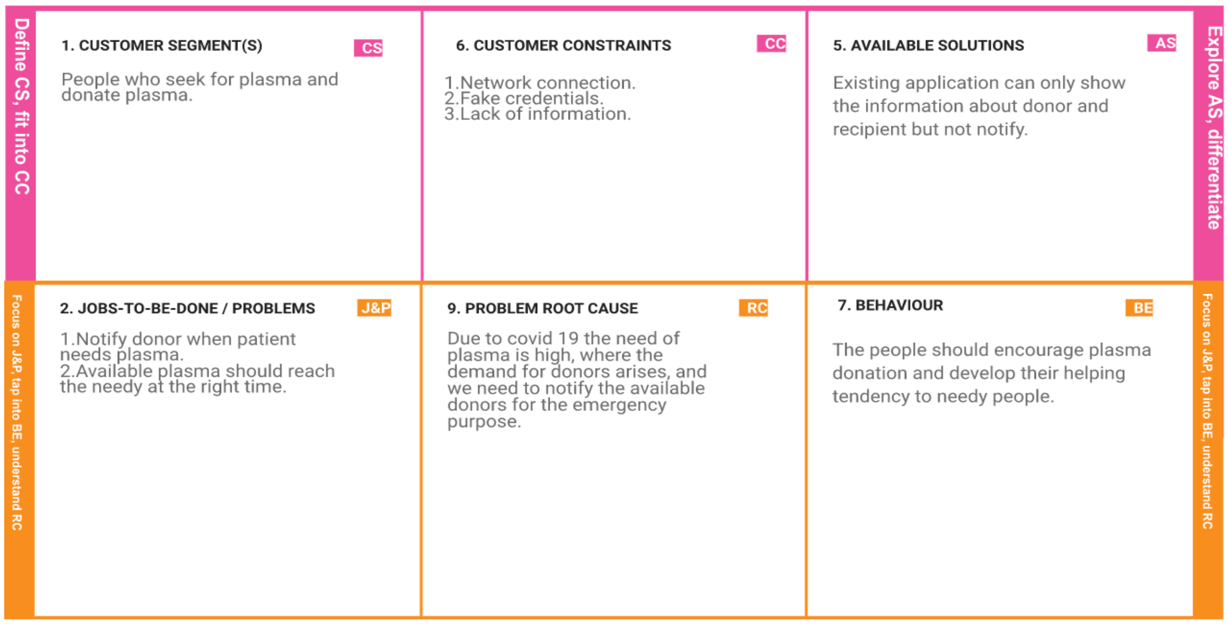
3.3. PROPOSED SOLUTION:

The new idea will improve the existing system and it will move from conventional desktop system to mobile system. This paper introduces new features of improved system over existing system in many aspects. The proposed plasma donor application helps the people who are in need of plasma by giving them all details of plasma availability or regarding the donors with the same blood group.This is a web application allows you to access the whole information about plasma donor application, readily scalable and adaptable to meet the complex need of plasma Who are Key Facilitator for the Healthcare Sector,it also supports all the functionalities of plasma donor application.

S.No.	Parameter	Description
1.	Problem Statement (Problemtobe solved)	Blood banks are required to maintain account of blood bags in the inventory. This increases with each blood donation recorded in our system and decreases as they are checked out upon hospital requests. Our system will need to keep the information up to date to ensure correctness of the inventory.
2.	Idea/Solution description	In regard to the problem, an application is to be built which would take the donor details, store them and inform them upon a request.

3.	Novelty/Uniqueness	Donors who wish to donate plasma can donate by uploading their COVID19 recovery certificate on the donor's page. If the donor is new, they must register before log in.If the donor is an existing user they need to login.Username and e-mail providedatthetimeof registration.
4.	Social Impact/Customer Satisfaction	The application is user friendly and anyone with basic knowledge can access it. The application seamless ly connects the donor and the person who need it and also hospitals who have availability of the plasma.
5.	Business Model (Revenue Model)	People will get used to this application,by cdlaborating with government and organizing blood donation camps.
6.	Scalability of the Solution	Since the app is going to store its data in cloud,it will continue to be efficient when large number of people uses it. Also when the number of requests for plasma increases, the call notification system will work fine without any disruption.

3.4.PROBLEM SOLUTION FIT :

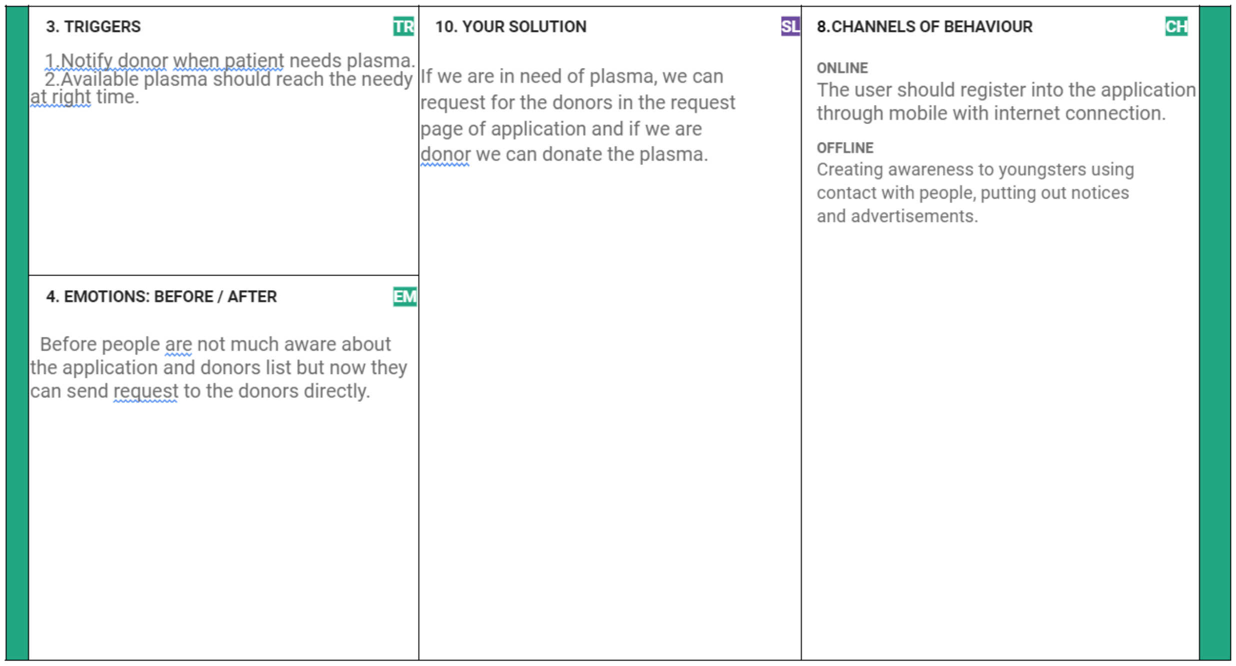


Define CS, fit into CC

Explore AS, differentiate

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

1. CUSTOMER SEGMENT(S) CS People who seek for plasma and donate plasma.	**6. CUSTOMER CONSTRAINTS** CC 1.Network connection. 2.Fake credentials. 3.Lack of information.	**5. AVAILABLE SOLUTIONS** AS Existing application can only show the information about donor and recipient but not notify.
2. JOBS-TO-BE-DONE / PROBLEMS J&P 1.Notify donor when patient needs plasma. 2.Available plasma should reach the needy at the right time.	**9. PROBLEM ROOT CAUSE** RC Due to covid 19 the need of plasma is high, where the demand for donors arises, and we need to notify the available donors for the emergency purpose.	**7. BEHAVIOUR** BE The people should encourage plasma donation and develop their helping tendency to needy people.



3. TRIGGERS TR 1.Notify donor when patient needs plasma. 2.Available plasma should reach the needy at right time. **4. EMOTIONS: BEFORE / AFTER** EM Before people are not much aware about the application and donors list but now they can send request to the donors directly.	**10. YOUR SOLUTION** SL If we are in need of plasma, we can request for the donors in the request page of application and if we are donor we can donate the plasma.	**8.CHANNELS OF BEHAVIOUR** CH **ONLINE** The user should register into the application through mobile with internet connection. **OFFLINE** Creating awareness to youngsters using contact with people, putting out notices and advertisements.

4.REQUIREMENT ANALYSIS

4.1.FUNCTIONAL REQUIREMENT

i. Admin

Admin can manage both donors and users. Admin has the only responsibility maintain and stored the record.

ii. Users

From this module user can create their account, when user create his account the user get a user id and password which identifies him uniquely. From this module user can search donor for blood.

iii. Donors Registration

In this module, people who are interested in donating blood get registered in this site and give his overall details related to donor. User details contain name, address,city,gender, blood group,location, contact number etc.

iv. Donor Search

The people who are in need of blood can search in our site for getting the detailsof donors having the same blood group and within the same city.

v. Notification

In this module, notification sends to donors for emergency.SMS send to registered donors phone number.

4.2. NON FUNCTIONAL REQUIREMENTS

1.Usability:

The system shall allow the users to access the system with pc using web application. The system uses a web application as an interface. The system is user friendly which makes the system easy

2. Availability:

The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

3. Scalability :

Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes in application and system processingdemands.

4.Security:

A security requirement is a statement of needed security functionality that ensures one of many different security properties of software is being satisfied.

5.Performance:

The information is refreshed depending upon whether some updates have occurred or not in the application. The system shall respond to the member in not less than two seconds from the time of the request submittal. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 5 seconds to appear on the screen.

6.Reliability:

The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data.The system will run 7days a week. 24 hours a day.

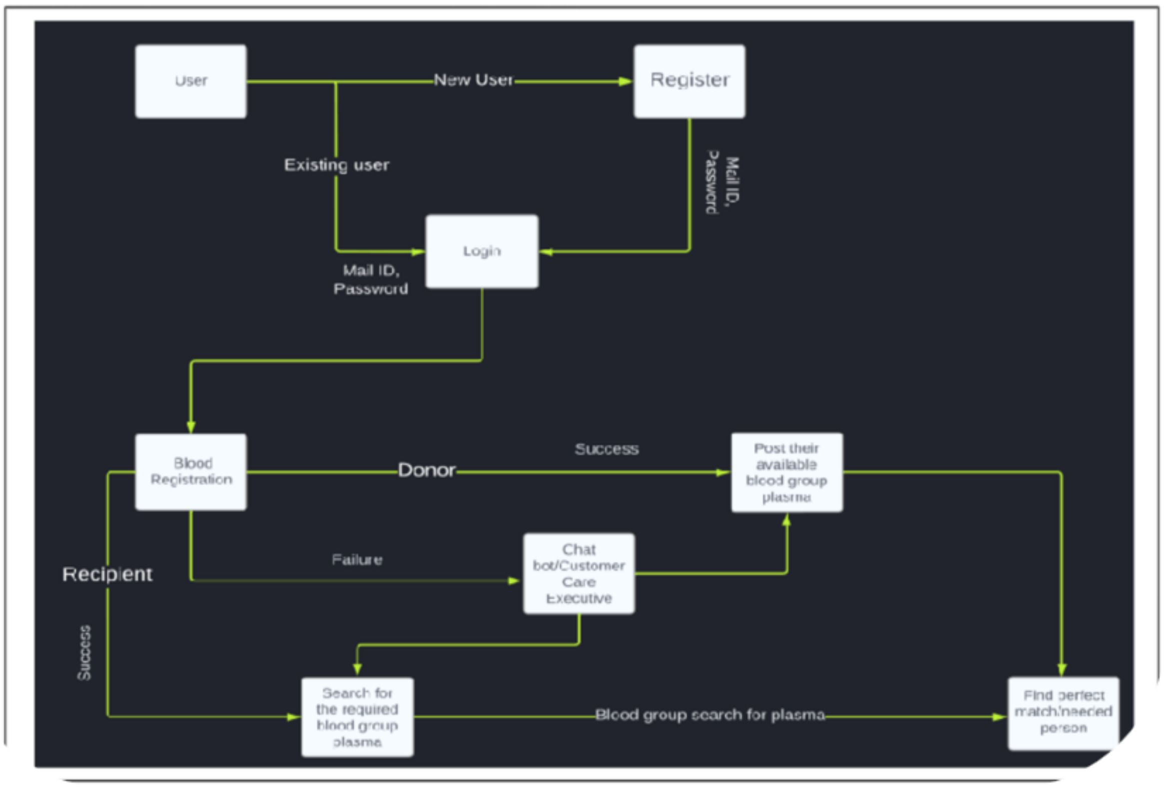
5.PROJECT DESIGN

5.1.DATA FLOW DIAGRAMS

A data-flow diagram is a visual representation of how data moves through a system or a process (usually an information system). The DFD additionally gives details about each entity's inputs and outputs as well as the process itself. A data-flowdiagram lacks controlflow, loops, and decision-making processes.Using a flowchart, certain operations depending on the data may be depicted.

Data flow Symbols:

DFD Level 0 (Industry Standard)

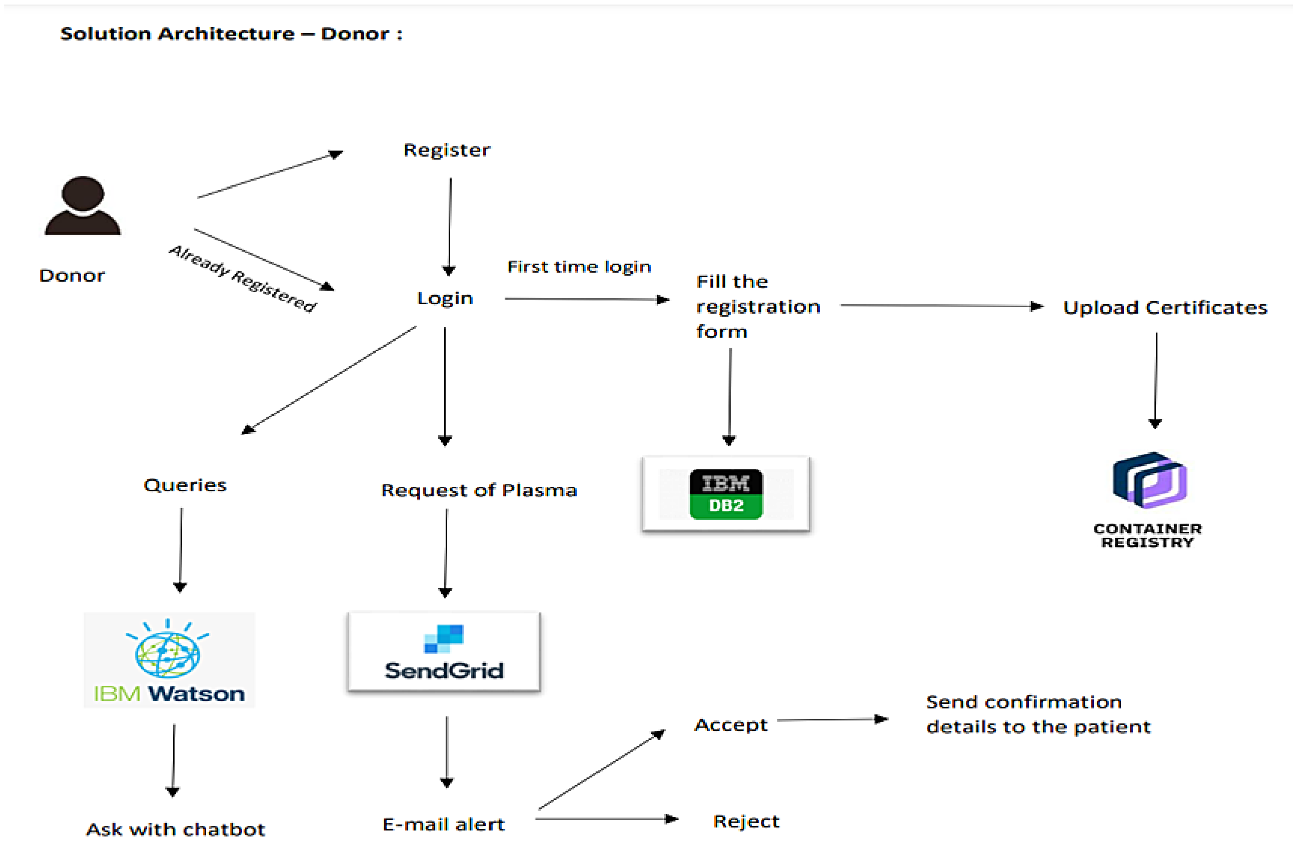

User
New User
Register
Existing user
Mail ID, Password
Login
Mail ID, Password
Blood Registration
Donor
Success
Post their available blood group plasma
Failure
Chat bot/Customer Care Executive
Recipient
Success
Search for the required blood group plasma
Blood group search for plasma
Find perfect match/needed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

5.2.SOLUTION & TECHNICAL ARCHITETURE

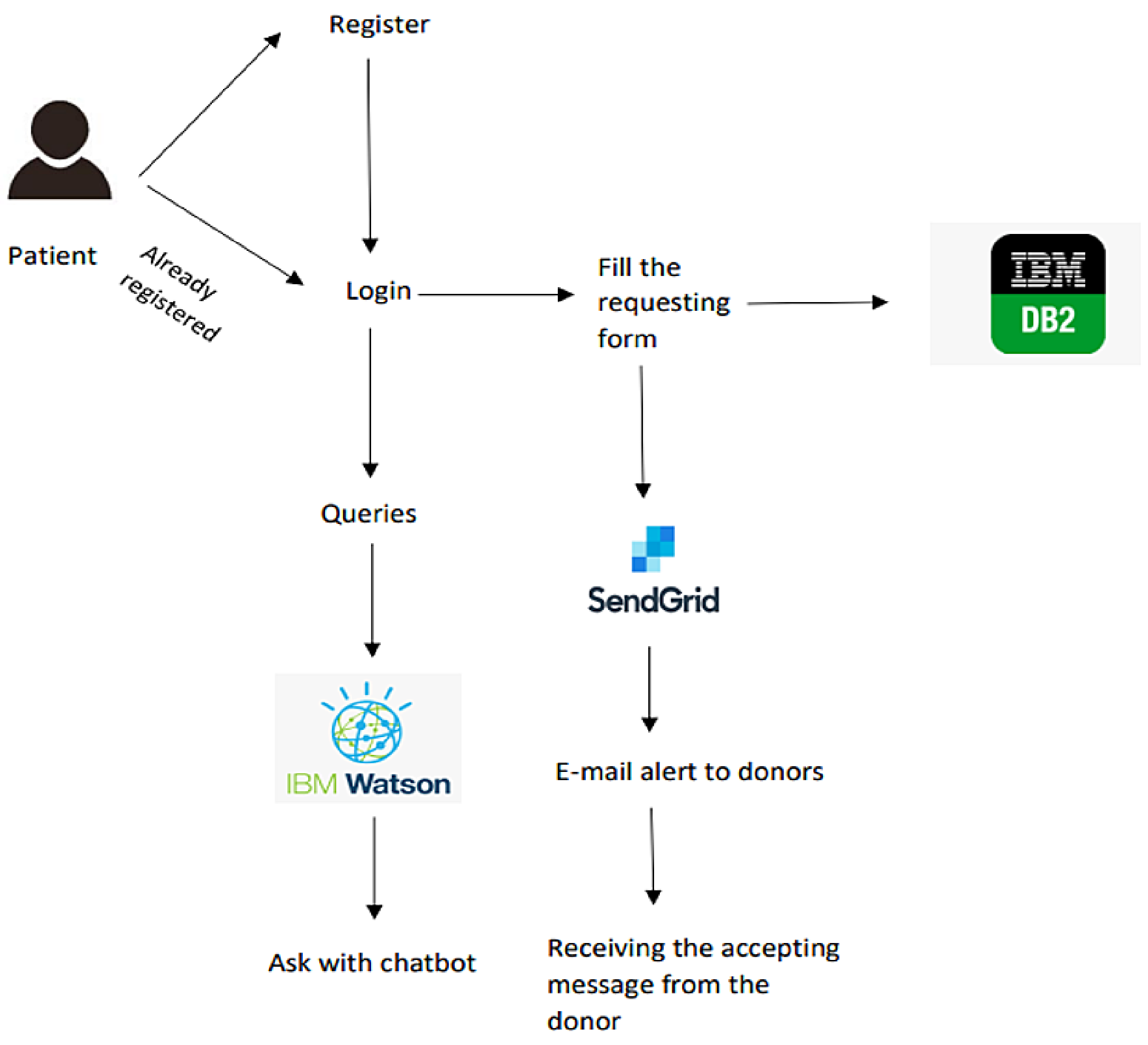
5.2.1.SOLUTION ARCHITETURE:

System architecture, also known as systems architecture, is the conceptual model that describes a system's structure, behavior, and additional viewpoints. An architecture description is a formal description and representation of a system that is organized in a way that allows for reasoning about the system's structures and behaviors. System architecture might include system components, their outwardly evident attributes, and the connections (e.g., behavior) between them. It can give a strategy for acquiring items and developing systems that will operate together to accomplish the entire system. There have been initiatives to codify languages for describing system architecture; theyare referred to collectively as architecture description languages (ADLs).

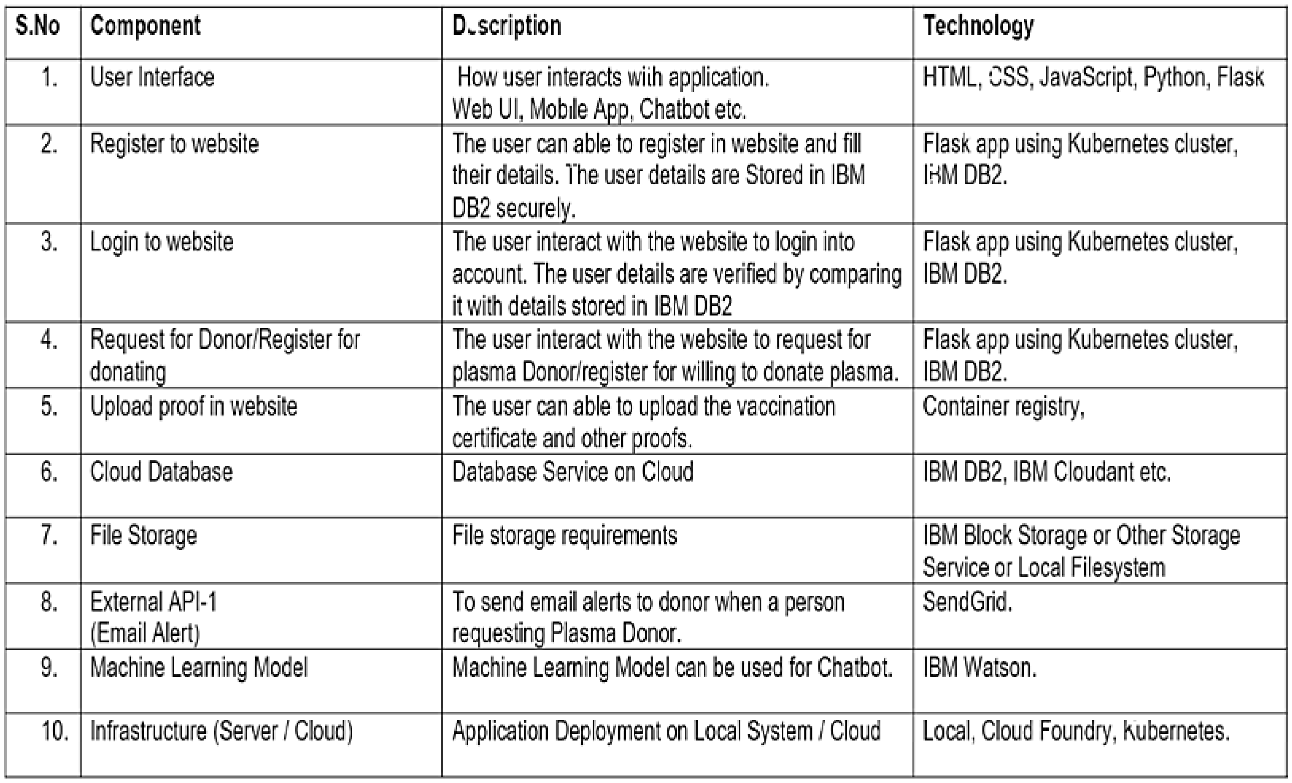
5.2.2.TECHNICAL ARCHITETURE:



Solution Architecture – Patient :



COMPONENTS & TECHNOLOGIES:



S.No	Component	Description	Technology
1.	User Interface	How user interacts with application. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript, Python, Flask
2.	Register to website	The user can able to register in website and fill their details. The user details are Stored in IBM DB2 securely.	Flask app using Kubernetes cluster, IBM DB2.
3.	Login to website	The user interact with the website to login into account. The user details are verified by comparing it with details stored in IBM DB2	Flask app using Kubernetes cluster, IBM DB2.
4.	Request for Donor/Register for donating	The user interact with the website to request for plasma Donor/register for willing to donate plasma.	Flask app using Kubernetes cluster, IBM DB2.
5.	Upload proof in website	The user can able to upload the vaccination certificate and other proofs.	Container registry,
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1 (Email Alert)	To send email alerts to donor when a person requesting Plasma Donor.	SendGrid.
9.	Machine Learning Model	Machine Learning Model can be used for Chatbot.	IBM Watson.
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud	Local, Cloud Foundry, Kubernetes.

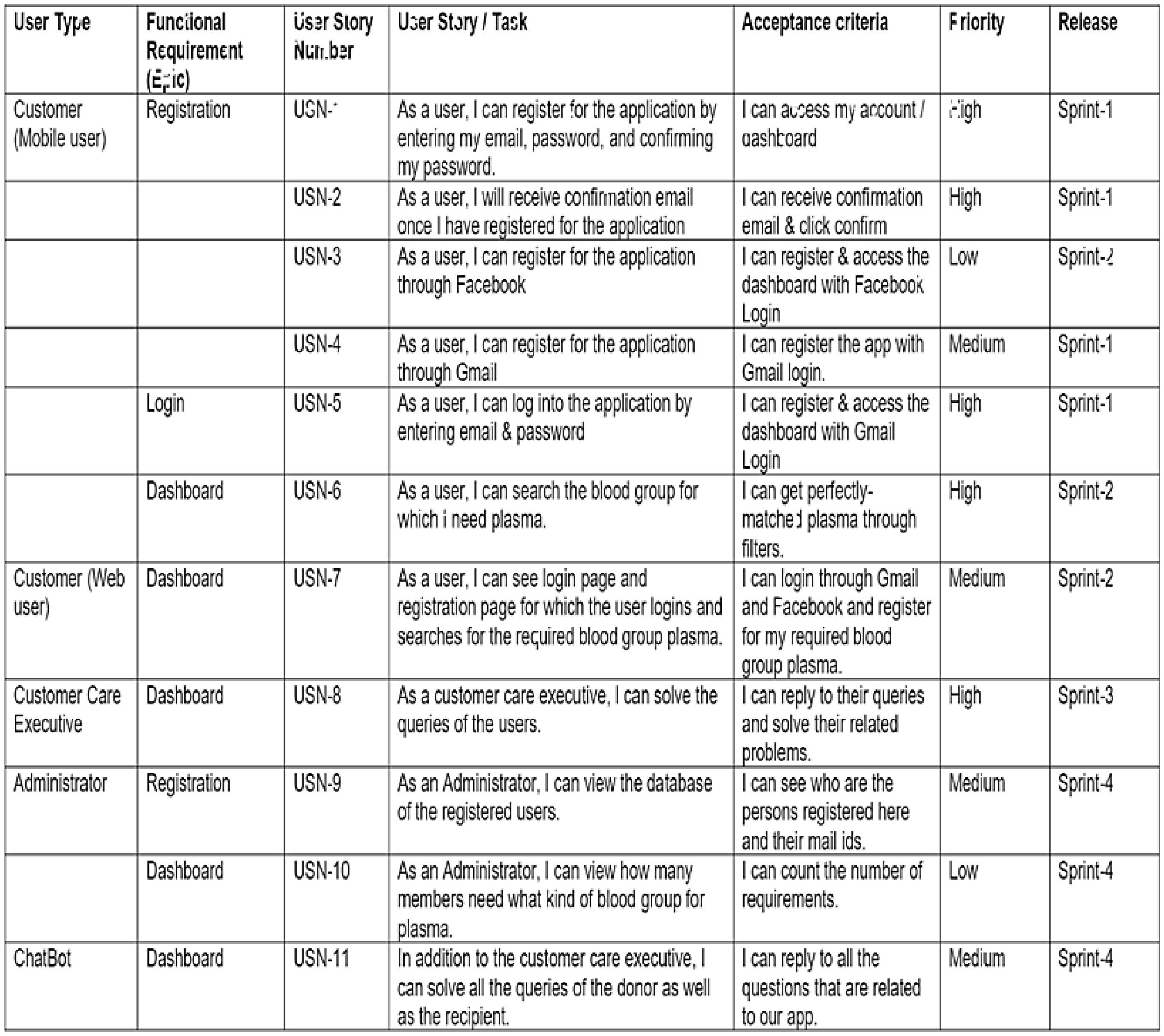
APPLICATION CHARACTERISTICS

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	PYCHARM
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	-
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Able to respond thechanges in an application
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	The system must alwaysbe functional

5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Takes no longer time to response

5.3.USER STORIES:

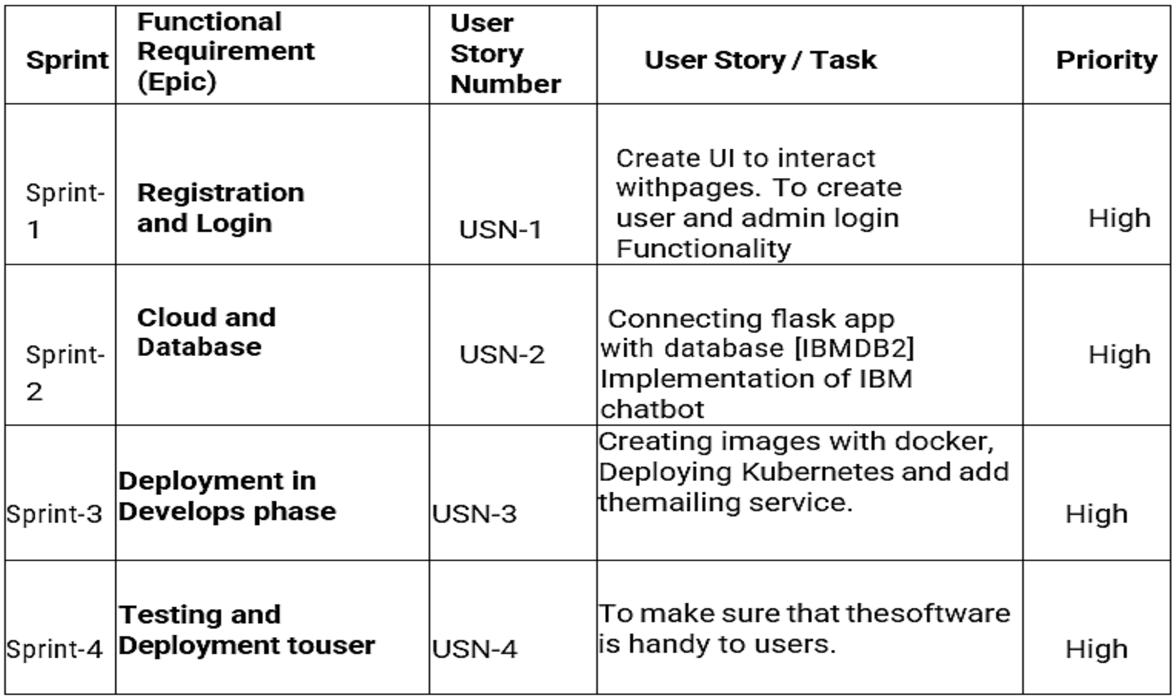
Use the below template list all the user stories for the product.



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

6.PROJECT PLANNING & SCHEDULING

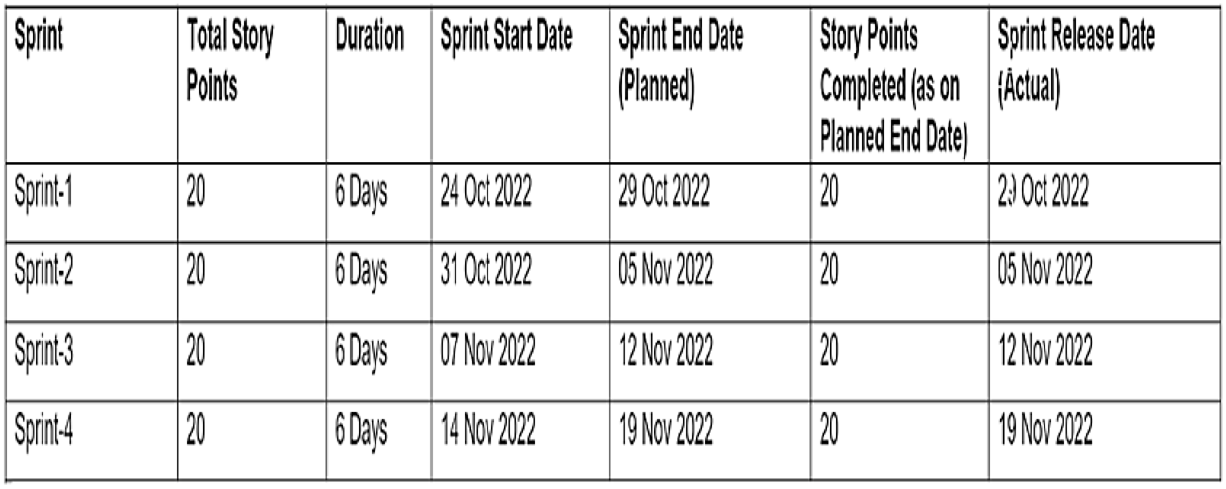
6.1.SPRINT PLANNING & ESTIMATION



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority
Sprint-1	**Registration and Login**	USN-1	Create UI to interact withpages. To create user and admin login Functionality	High
Sprint-2	**Cloud and Database**	USN-2	Connecting flask app with database [IBMDB2] Implementation of IBM chatbot	High
Sprint-3	**Deployment in Develops phase**	USN-3	Creating images with docker, Deploying Kubernetes and add themailing service.	High
Sprint-4	**Testing and Deployment touser**	USN-4	To make sure that thesoftware is handy to users.	High

6.2.SPRINT DELIVERY SCHEDULE

6.2.1. PROECT TRACKER



Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.2.2. VELOCITY

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint).Let’s calculate the team’s average velocity(AV) per iteration unit(story points per day).

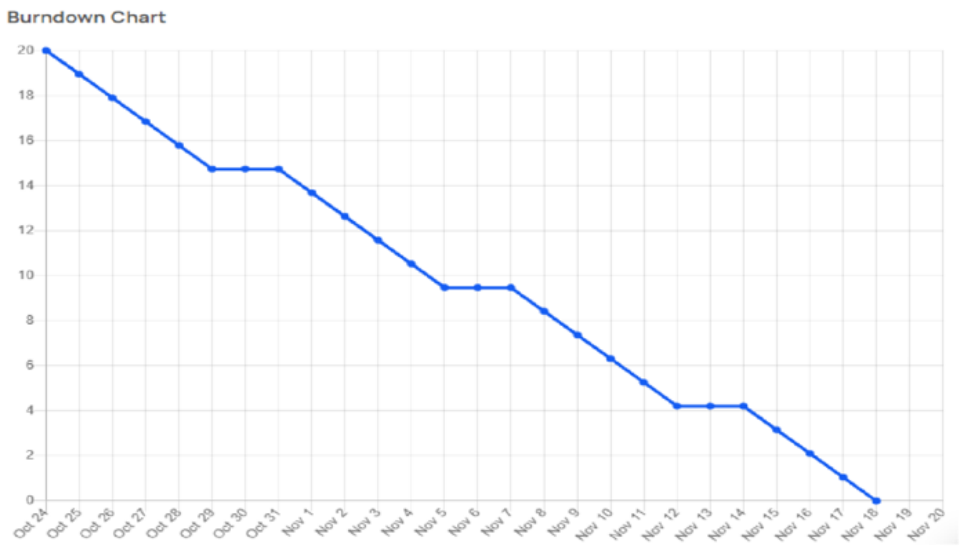
$$\text{Average velocity (AV)} = \frac{\text{Velocity}}{\text{Sprint duration}}$$

Sprint	Average Velocity
Sprint-1	6.5
Sprint-2	8
Sprint-3	7.6
Sprint-4	8.5

Total Average Velocity =7.65

6.2.3.BURNDOWN CHART (Developed using JIRA)

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



7.CODING AND SOLUTIONING

7.1.FEATURE -1:

```
#main.py
#importing libraries
from flask import Flask,render_template, request, redirect, url_for, session
import ibm_db
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=98538591-7217-4024-b027-
8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30875;SECURITY=SSL;
SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=tqj08800;PWD=UKFB37LGZ3q0xHGy",'','
')
app = Flask(__name__)
print(conn)
print("DATABASE CONNECTED SUCESSFULLY")
```

7.2 FEATURE - 2:

```
   DATABASE=bludb
   HOSTNAME=98538591-7217-4024-b027-
8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud
   PORT=30875
   SECURITY=SSL
SSLServerCertificate=DigiCertGlobalRootCA.crt
UID=tqj08800
PWD=UKFB37LGZ3q0xHGy
```

SENDGRID CODING:

```
import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail

message = Mail(
        from_email='plasmadonar21@gmail.com',
        to_emails='gusamyuktha@gmail.com',
        subject='Hello there! Welcome to PLASMA DONOR APPLICATION',
             html_content='<strong>PDA warmly welcomes YOU!!!</strong</strong>')
try:
            sg = SendGridAPIClient('SG.agdL93_hTXSf1Pi8EGC9xw.zlxSPuwGvwW0zz9CaFoG1kqF-
Cq9fPLBROX-_ALVk_g')
            response = sg.send(message)
            print(response.status_code)
            print(response.body)
            print(response.headers)

except Exception as e:
           print(str(e))
```

7.3 DATABASE SCHEME :

Database Used : IBM Cloud Database

8. TESTING

8.1.TEST CASES

A test case has components that describe input, action and an expected response, in order to determine if a feature of an application is working correctly.A test case is a set of instructions on “HOW” to validate a particular test objective/target, which when followed will tell us if the expected behavior of the system is satisfiedor not.

Characteristics of a good test case:

i. Accurate: Exacts the purpose.

ii. Economical: No unnecessary steps or words.

iii. Traceable: Capable of being traced to requirements.

iv. Repeatable: Can be used to perform the test over and over.

v. Reusable: Can be reused if necessary.

S.NO	**Scenario**	**Input**	**Excepted output**	**Actual output**
1	Admin Login Form	User name and password	Login	Login success.
2	DonorRegistration Form	Donor basic details	Registration	Donor registration details stored in database.

3	User Registration Form	User basic details	Registration	User registration details stored in database.
4	User Login Form	User name and password	Login	Login success.

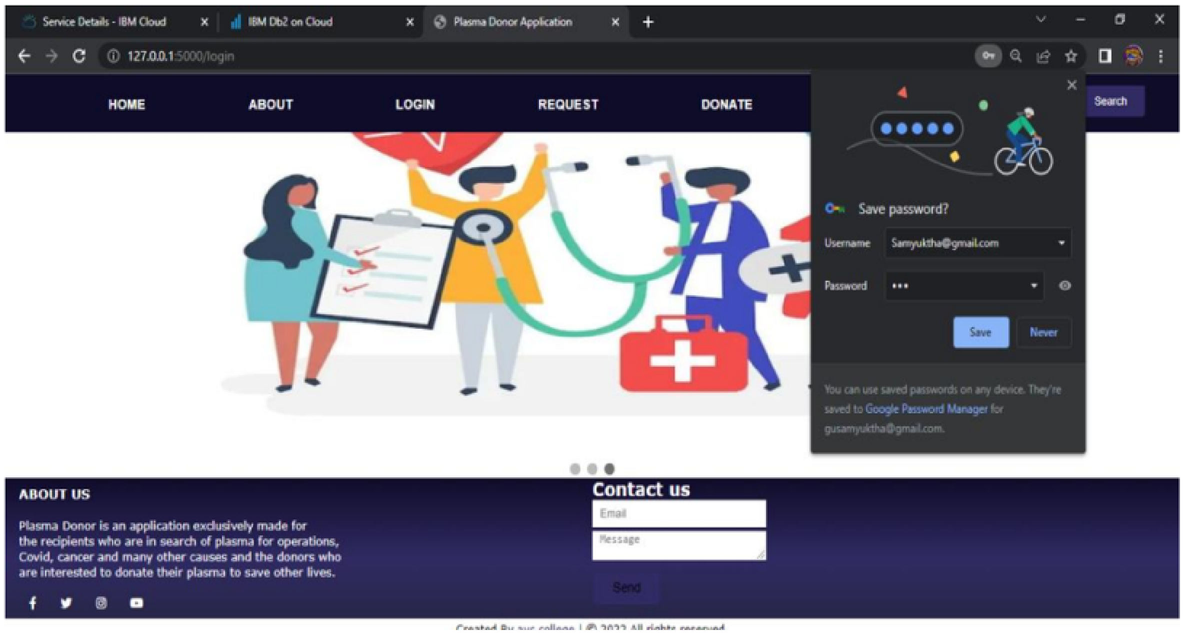
8.2.USER ACCEPTANCE TESTING

This is a type of testing done by users, customers, or other authorized entities to determine application/software needs and business processes. Acceptance testing is the most important phase of testing as this decides whether the client approves the application/software or not. It may involve functionality, usability, performance, and U.I of the application. It is also known as user acceptance testing (UAT), operational acceptance testing(OAT), and end-user testing.

9. RESULTS

9.1.PERFORMANCE METRICS

9.1.1.HOME PAGE(After Successful Login)



Username: Samyuktha@gmail.com

Password: •••

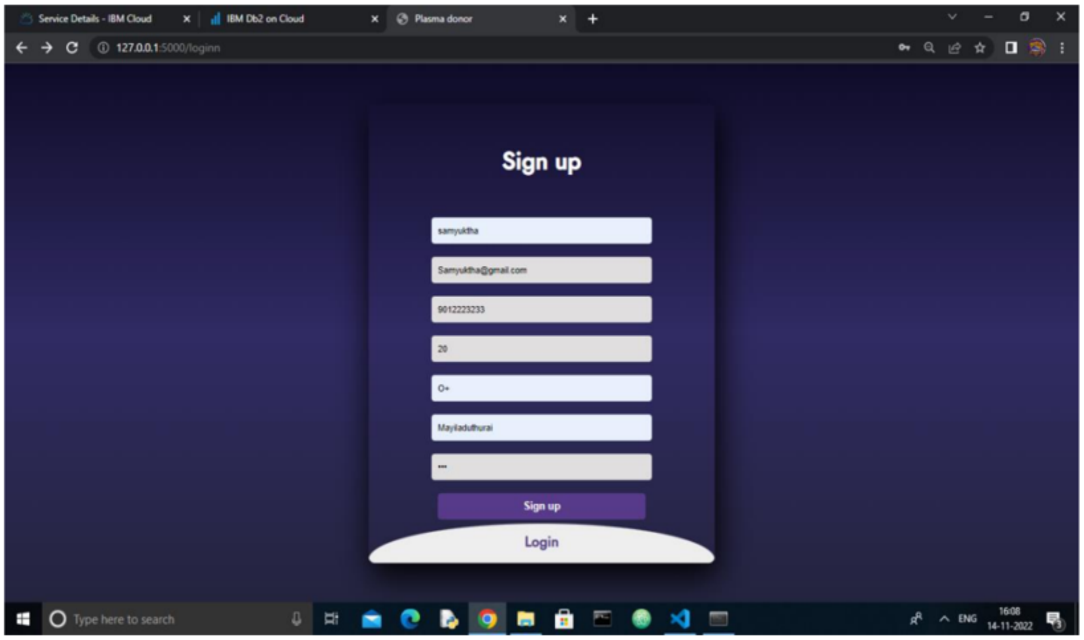
You can use saved passwords on any device. They're saved to Google Password Manager for gusamyuktha@gmail.com.

ABOUT US

Plasma Donor is an application exclusively made for the recipients who are in search of plasma for operations, Covid, cancer and many other causes and the donors who are interested to donate their plasma to save other lives.

Created By our college | © 2022 All rights reserved

9.1.2.SIGNUP PAGE(fill details)



samyuktha

Samyuktha@gmail.com

9012223233

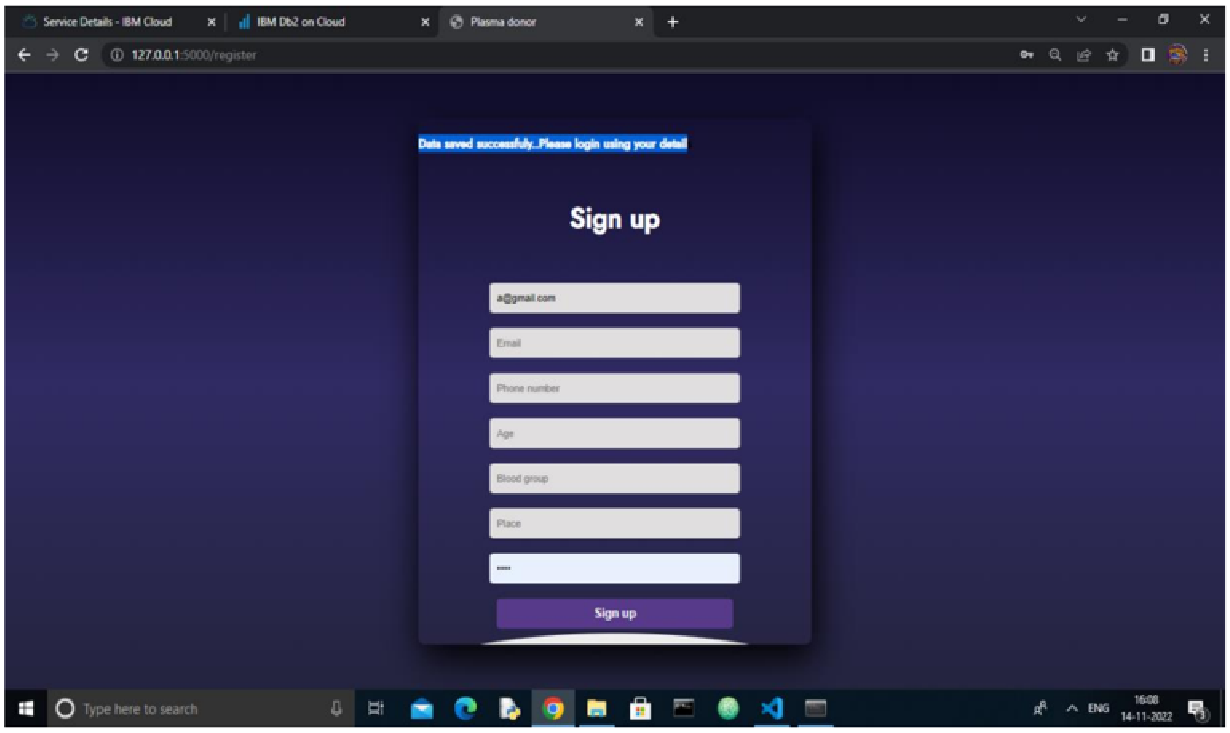
20

O+

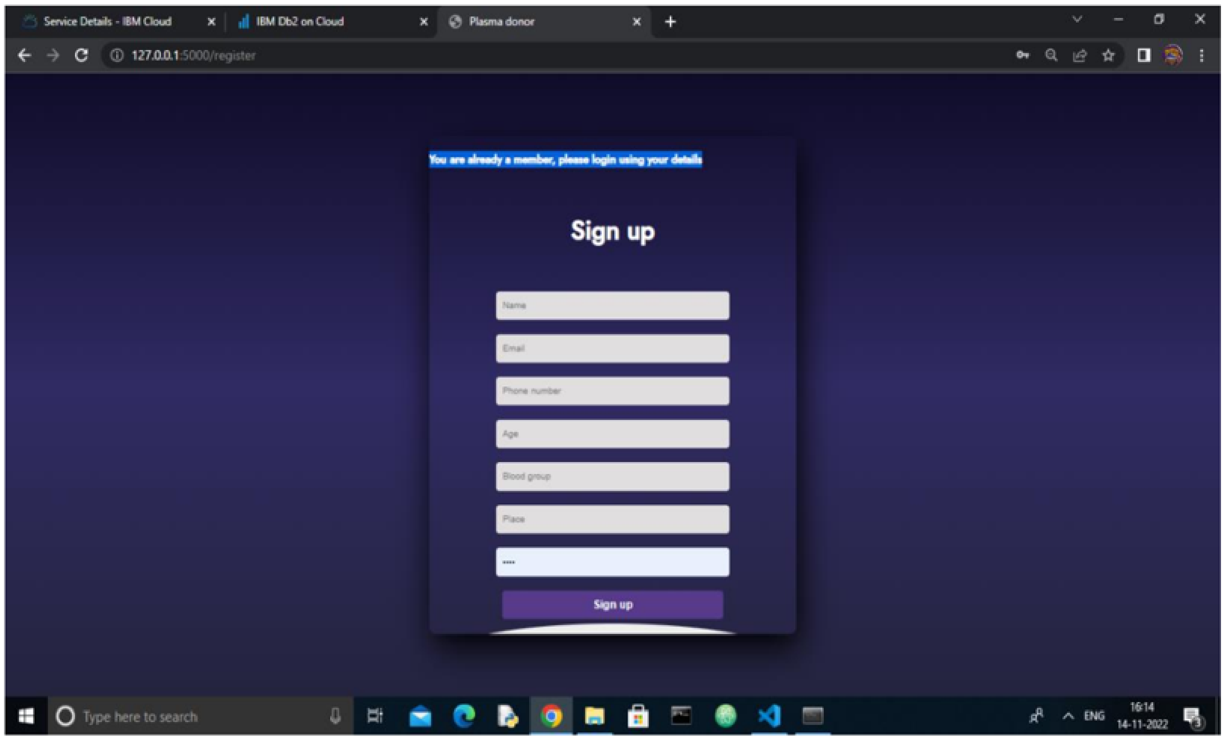
Mayiladuthurai

•••

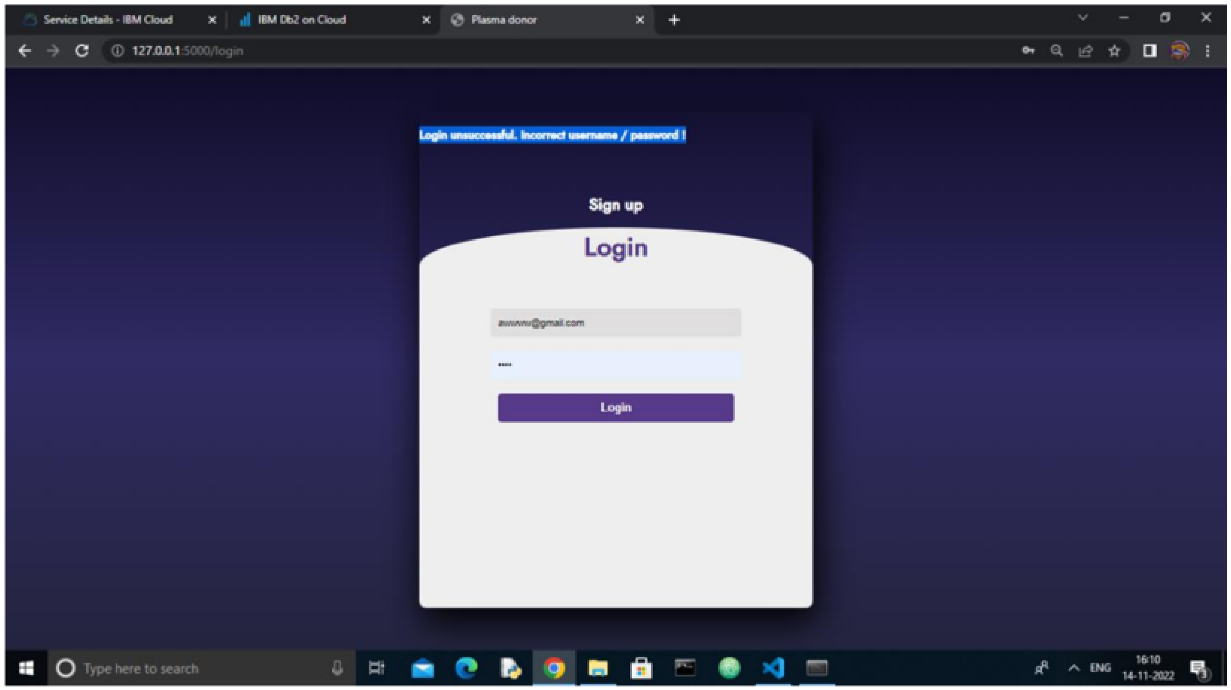
9.1.3.SIGNUP PAGE(Data Saved Successfully)


Data saved successfully..Please login using your detail

9.1.4.SIGNUP(Notify an existing user)

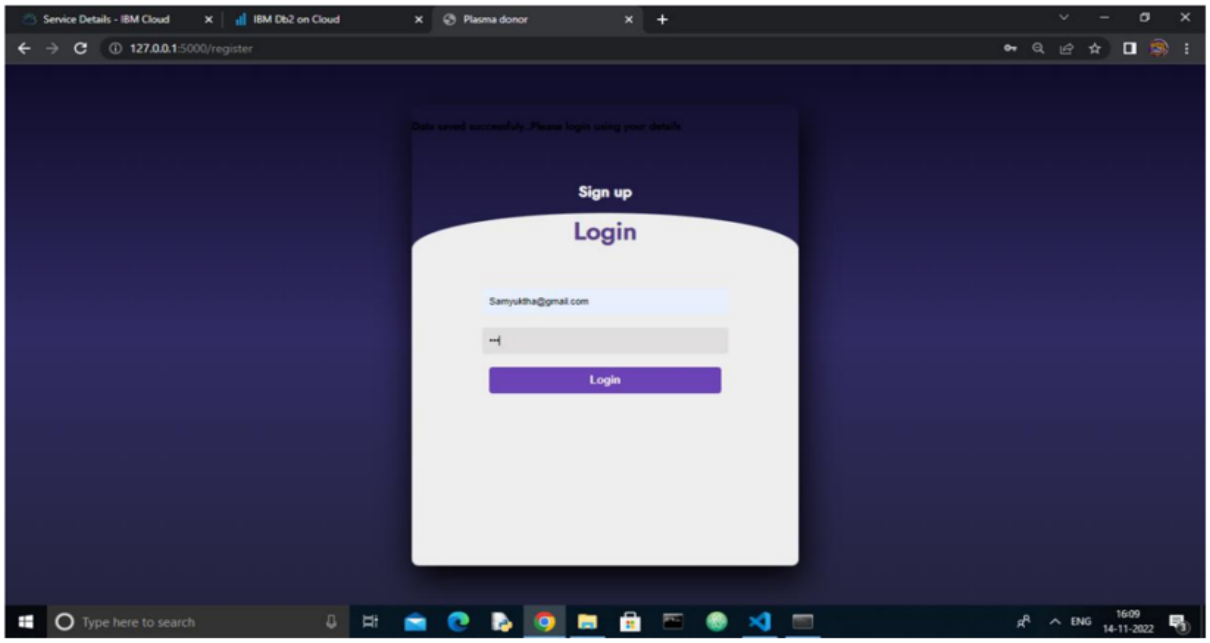

You are already a member, please login using your details

9.2.1.LOGIN PAGE(incorrect username & password)



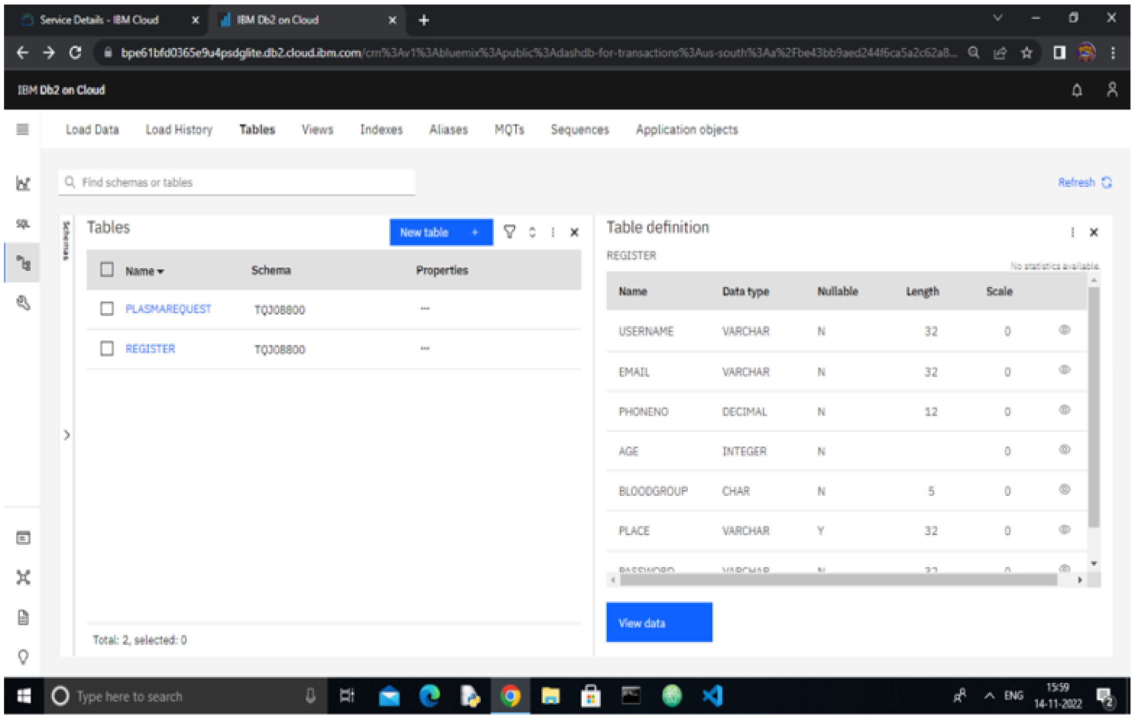
Login unsuccessful. Incorrect username / password !

9.2.2.LOGIN PAGE



Data saved successfully..Please login using your details

9.1.4.DATABASE



Tables

Name	Schema	Properties
PLASMAREQUEST	TQJ08800	···
REGISTER	TQJ08800	···

Total: 2, selected: 0

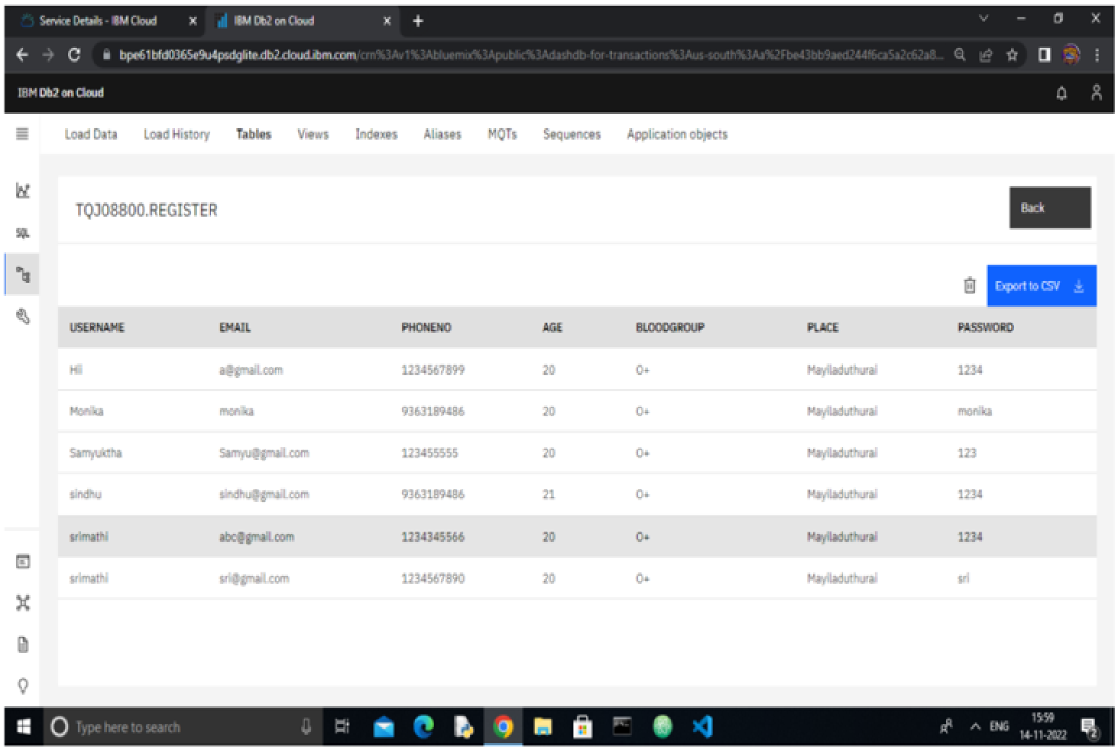
Table definition

REGISTER

No statistics available.

Name	Data type	Nullable	Length	Scale
USERNAME	VARCHAR	N	32	0
EMAIL	VARCHAR	N	32	0
PHONENO	DECIMAL	N	12	0
AGE	INTEGER	N		0
BLOODGROUP	CHAR	N	5	0
PLACE	VARCHAR	Y	32	0
PASSWORD	VARCHAR	N	32	0

View data


Service Details - IBM Cloud
IBM Db2 on Cloud
bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Fbe43bb9aed244f6ca5a2c62a8...
IBM Db2 on Cloud
Load Data
Load History
Tables
Views
Indexes
Aliases
MQTs
Sequences
Application objects
Back
Export to CSV
Type here to search
ENG
15:59
14-11-2022

TQJ08800.REGISTER

USERNAME	EMAIL	PHONENO	AGE	BLOODGROUP	PLACE	PASSWORD
Hi	a@gmail.com	1234567899	20	O+	Mayiladuthurai	1234
Monika	monika	9363189486	20	O+	Mayiladuthurai	monika
Samyuktha	Samyu@gmail.com	123455555	20	O+	Mayiladuthurai	123
sindhu	sindhu@gmail.com	9363189486	21	O+	Mayiladuthurai	1234
srimathi	abc@gmail.com	1234345566	20	O+	Mayiladuthurai	1234
srimathi	sri@gmail.com	1234567890	20	O+	Mayiladuthurai	sri

10. ADVANTAGES & DISADVANTAGES

10.1.ADVANTAGES

- ✓ It is a user-friendly application.
- ✓ The people in need of plasma can search for the donors by giving their blood group and city name.
- ✓ It saves time as he can search donors online without going anywhere.
- ✓ Using this system user can get plasma in time and can save and here our system work, whenever a person needs plasma user get information of the person who has the same blood group needs.

10.2.DISADVANTAGES

i. It is time consuming.

ii. It leads to error prone results.

iii. It consumes lot of manpower to better results.

iv. It lacks of data security.

v. Retrieval of data takes lot of time.

11. CONCLUSION

This project is designed for successful completion of project on Plasma Donor Application system. The basic building aim is to provide plasma donation service to the city recently.Plasma Donor Application System is a Web based application that is designed to store, process,retrieve and analyze information concerned with the administrative and inventory management within a plasma.This project aims at maintaining all the information pertaining to plasma donors, different blood groups available in each plasma bank and helps them manage in a better way plasma donation system can collect plasma from many donors in short from various sources and distribute that plasma to needy people who require plasma. To do all this we require high quality Web Application to manage those jobs. Plasma application provides a reliable platform to connect local plasma donors with patients.

12. FUTURE SCOPE

This system is developed such a way that additional enhancement can be done without much difficulty. The renovation of the project would increase the flexibility of the system. In future, we can develop this project in android platform. We will add extra features like donor location tracking system (GPS), Feedback form, and enable call option

13. APPENDIX

13.1.SOURCE CODE:

#app.py

```
from flask import Flask,render_template, request, redirect, url_for, session
import ibm_db
import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=98538591-7217-4024-b027-
8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30875;SECURITY=SSL;SSL
ServerCertificate=DigiCertGlobalRootCA.crt;UID=tqj08800;PWD=UKFB37LGZ3q0xHGy","","")

app = Flask(__name__)

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

@app.route("/base")
def base():
    return render_template('dashboard.html')

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

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

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

# @app.route('/dis')
# def dis():
#   return render_template('display.html')
```

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

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

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

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

# @app.route('/display')
# def data():
#     sql = "SELECT bloodgroup FROM plasmadonate group by bloodgroup"
#     stmt = ibm_db.prepare(conn, sql)
#     ibm_db.execute(stmt)
#     count = ibm_db.fetch_assoc(stmt)
#     return render_template('display.html', output_data = count)

@app.route('/display')
def display():
    # sql="SELECT * FROM plasmadonate"
    # stmt = ibm_db.prepare(conn, sql)
    # print(stmt)
    # ibm_db.execute(stmt)
    # account = ibm_db.fetch_assoc(stmt)
    # print(account)
    # return render_template('disply.html', data=account)
    d=[]
    sql="SELECT * FROM plasmadonate"
    stmt = ibm_db.prepare(conn, sql)
    #print(stmt)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print(account)
    while account != False:
        d.append(account)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
    print(d)
    return render_template('disply.html', d=d)


@app.route('/dis')
def dis():
    # sql="SELECT * FROM plasmadonate"
    # stmt = ibm_db.prepare(conn, sql)
    # print(stmt)
    # ibm_db.execute(stmt)
    # account = ibm_db.fetch_assoc(stmt)
    # print(account)
    # return render_template('disply.html', data=account)
    d=[]
    sql="SELECT * FROM plasmarequest"
    stmt = ibm_db.prepare(conn, sql)
    #print(stmt)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print(account)
    while account != False:
        d.append(account)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
    print(d)
    return render_template('display.html', d=d)

@app.route('/register',methods = ['POST', 'GET'])
def register():
  if request.method == 'POST':

    name = request.form['username']
    email = request.form['email']
    phone = request.form['phone']
    age = request.form['age']
    bloodgroup = request.form['bloodgroup']
    address = request.form['place']
    password = request.form['password']
    sql = "SELECT * FROM register WHERE email =?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,email)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)

    if account:
      return render_template('login.html', msg="You are already a member, please login using your details")
    else:
      insert_sql = "INSERT INTO register VALUES (?,?,?,?,?,?,?)"
      prep_stmt = ibm_db.prepare(conn, insert_sql)
      ibm_db.bind_param(prep_stmt, 1, name)
      ibm_db.bind_param(prep_stmt, 2, email)
      ibm_db.bind_param(prep_stmt, 3, phone)
      ibm_db.bind_param(prep_stmt, 4, age)
      ibm_db.bind_param(prep_stmt, 5, bloodgroup)
      ibm_db.bind_param(prep_stmt, 6, address)
      ibm_db.bind_param(prep_stmt, 7, password)
      ibm_db.execute(prep_stmt)

      message = Mail(
              from_email='plasmadonar21@gmail.com',
              to_emails=mail,
              subject='Hello there! Welcome to PLASMA DONOR APPLICATION',
              html_content='<strong>Thank you for registering in our website. PDA warmly welcomes
YOU!!!</strong</strong>')
     try:
             sg = SendGridAPIClient('SG.agdL93_hTXSf1Pi8EGC9xw.zlxSPuwGvwW0zz9CaFoG1kqF-
Cq9fPLBROX-_ALVk_g')
             response = sg.send(message)
             print(response.status_code)
             print(response.body)
             print(response.headers)

       except Exception as e:
            print(str(e))
    return render_template('login.html', msg="Data saved successfully..Please login using your details")

@app.route('/plasmareq',methods = ['POST', 'GET'])
def plasmareq():
  if request.method == 'POST':

    name = request.form['name']
    email = request.form['email']
    phone = request.form['phone']
    bloodgroup = request.form['bloodgroup']
    date = request.form['date']
    address = request.form['address']
    district = request.form['district']
    state = request.form['state']
    age = request.form['age']

    insert_sql = "INSERT INTO plasmarequest VALUES (?,?,?,?,?,?,?,?,?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prep_stmt, 1, name)
    ibm_db.bind_param(prep_stmt, 2, email)
    ibm_db.bind_param(prep_stmt, 3, phone)
    ibm_db.bind_param(prep_stmt, 4, bloodgroup)
    ibm_db.bind_param(prep_stmt, 5, date)
    ibm_db.bind_param(prep_stmt, 6, address)
    ibm_db.bind_param(prep_stmt, 7, district)
    ibm_db.bind_param(prep_stmt, 8, state)
    ibm_db.bind_param(prep_stmt, 9, age)
    ibm_db.execute(prep_stmt)

    return render_template('dashboard.html', msg="Data saved successfully")


@app.route('/donorform',methods = ['POST', 'GET'])
def donorform():
  if request.method == 'POST':

    name = request.form['name']
    email = request.form['email']
    phone = request.form['phone']
    bloodgroup = request.form['bloodgroup']
    date = request.form['date']
    address = request.form['address']
    district = request.form['district']
    state = request.form['state']
    age = request.form['age']
    insert_sql = "INSERT INTO plasmadonate VALUES (?,?,?,?,?,?,?,?,?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prep_stmt, 1, name)
    ibm_db.bind_param(prep_stmt, 2, email)
    ibm_db.bind_param(prep_stmt, 3, phone)
    ibm_db.bind_param(prep_stmt, 4, bloodgroup)
    ibm_db.bind_param(prep_stmt, 5, date)
    ibm_db.bind_param(prep_stmt, 6, address)
    ibm_db.bind_param(prep_stmt, 7, district)
    ibm_db.bind_param(prep_stmt, 8, state)
    ibm_db.bind_param(prep_stmt, 9, age)
    ibm_db.execute(prep_stmt)

    return render_template('dashboard.html', msg="Data saved successfully")

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

    email = request.form['email']
    password = request.form['password']

    sql = "SELECT * FROM register WHERE email =? AND password=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,email)
    ibm_db.bind_param(stmt,2,password)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    if account:
            return render_template('dashboard.html')
    else:
        return render_template('login.html', msg="Login unsuccessful. Incorrect username / password !")
```

13.2.GITHUB & PROJECT DEMO LINK :

Github:https://github.com/IBM-EPBL/IBM-Project-26530-1660028990