PLASMA DONAR APLLICATION

IBM-Project-41916-1660646127

Team ID: PNT2022TMID12046

PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

PROJECT REPORT

YAAZHINI P(73151913106)

VIGNESH U(73151913510)

SHIVARAJA M(73151913506)

MOHAMED ISMAIL SHEREEF N(73151913055)

K S R COLLEGE OF ENGINEERING, TIRUCHENGODE, NAMAKKAL

BACHELOR OF COMPUTER SCIENCE AND ENGINEERING

INDEX

1. **INTRODUCTION**

1. Project Overview
2. Purpose

2. **LITERATURE SURVEY**

1. Existing problem
2. References
3. Problem Statement Definition

3. **IDEATION & PROPOSED SOLUTION**

1. Empathy Map Canvas
2. Ideation & Brainstorming
3. Proposed Solution
4. Problem Solution fit

4. **REQUIREMENT ANALYSIS**

1. Functional requirement
2. Non-Functional requirements

5. **PROJECT DESIGN**

1. Data Flow Diagrams
2. Solution & Technical Architecture
3. User Stories

6. **PROJECT PLANNING & SCHEDULING**

1. Sprint Planning & Estimation
2. Sprint Delivery Schedule
3. Reports from JIRA

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

1. Feature 1
2. Feature 2
3. Feature 3
4. Feature 4

8. TESTING

1. Test Cases
2. User Acceptance Testing

9. RESULTS

1. Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. SETTING UP APPLICATION ENVIRONMENT

1. Create Flask Project
2. Create IBM Cloud Account
3. Install IBM Cloud CLI
4. Docker CLI Installation

12. CONCLUSION

13. FUTURE SCOPE

APPENDIX

Source Code

GitHub & Project Demo Link

1. INTRODUCTION

ABSTRACT:

Plasma Donor Application is aimed to developing a Plasma Donor information via online. The numbers of blood donor are very less when compared with other countries. In our project, the consumer wants to make request for blood and soon the donor will be asked to enter an individual's personal details. The system that is designed to store, process, retrieve and analyse information concerned with the administrative and inventory management within a blood bank. At the emergency time of blood needed, we can check for blood donor nearby using GPS. The admin is the main authority who can do addition, deletion and modification if it’s required. Once we can’t able to reach the accepted donor and soon the application will send a request to another donor which will be represented in queue data structure. If the donor accepts the request, then a one-time password (OTP) will be sent to the donor for verification. Aim is to provide transparency in this field, make the process of obtaining blood from a blood bank hassle free and corruption free and make the system of Plasma Donor Application effective. Plasma donation app provide the list of donors in your city/area. Once the donor donates the blood it will automatically remove the donor details for next three months. This application takes care of different modules and their associated reports which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff. Application is developed in a manner that is easily manageable, time saving and relieving one from manual works. The requirement of the blood has to be requested and we supply the information of the donor. The donors can update their status whether they are available or not.

1.1 PROJECT OVERVIEW

Category: Cloud App Development
Team ID : PNT2022TMID12046
Skills Required: IBM Cloud, HTML, Javascript, IBM Cloud Object Storage, Python Flask, Kubernetes, Docker, IBM DB2, IBM Container Registry

The Blood Donation Agent is to create an e-Information about the donor and organization that are related to donating the blood. Through this application any person who is interested in donating the blood can register himself in the same way if any organization wants to register itself with this site that can also register. Moreover if any general consumer wants to make request blood online he can also take the help of this site. Admin is the main authority who can do addition" deletion" and modification if required.

1.2 PURPOSE

This project is aimed to developing an online Blood Donation Information. The entire project has been developed keeping in view of the distributed client server computing technology" in mind.

The Blood Donation Agent is to create an e-Information about the donor and organization that are related to donating the blood. Through this application any person who is interested in donating the blood can register himself in the same way if any organization wants to register itself with this site that can also register. Moreover if any general consumer wants to make request blood online he can also take the help of this site. Admin is the main authority who can do addition" deletion" and modification if required.

The project has been planned to be having the view of distributed architecture" with centralized storage of the database. The application for

the storage of the data has been planned. Using the constructs of MS-SQL server and all the user interfaces have been designed using the ASP.Net technologies.

The database connectivity is planned using the "SQL Connection" methodology. The standards of security and data protective mechanism have been given a big choice for proper usage.The application takes care of different modules and their associated reports" which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff.

The entire project has been developed keeping in view of the distributed client server computing technology" in mind. The specification has been normalized up to 3NF to eliminate all the anomalies that may arise due to the database transaction that are executed by the general users and the organizational administration. The user interfaces are browser specific to give distributed accessibility for the overall system. The internal database has been selected as MS-SQL server 2000.

The basic constructs of table spaces" clusters and inde0es have been e0ploited to provide higher consistency and reliability for the data storage. The MS-SQL server 2000 was a choice a sit provides the constructs of high-level reliability and security. The total front end was dominated using the A%(.)et technologies. At all proper levels high care was taken to check that the system manages the data consistency with proper business rules or validations.

The database connectivity was planned using the latest "SQL Connection" technology provided by Microsoft corporation. The authentication and authorization was cross checked at all the relevant stages. The user level accessibility has been restricted into two zones namely.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

Introduction

Applying optimization methods to healthcare management and logistics is a developing research area with numerous studies. Specifically, facility location, staff rostering, patient allocation, and medical supply transportation are the main themes analysed. Optimization approaches have been developed for several healthcare related problems, ranging from the resource management in hospitals to the delivery of care services in a territory. However, optimization approaches can also improve other services in the health system that have been only marginally addressed, yet. One of them is the Blood Donation (BD) system, aiming at providing an adequate supply of blood to Transfusion Centres (TCs) and hospitals. Blood is necessary for several treatments and surgeries, and still a limited resource.

The need for blood is about ten million units per year in the USA, 2.1 in Italy and 2 in Turkey; moreover, people still die in some countries because of inadequate supply of blood products (World Health Organization 2014). Hence, BD plays a fundamental role in healthcare systems, aiming at guaranteeing an adequate blood availability to meet the demand and save lives. In Western countries, blood is usually collected from donors, i.e., unpaid individuals who give blood voluntarily. Blood is classified into groups (A and subgroups, B, 0 or AB) and based on the Rhesus factor (Rh+ or Rh-), and each donor should be correctly matched with the patient who receives his/her blood. Moreover, as it may transmit diseases, blood must be screened before utilization.

2.2 REFERENCES

[1] In “Android blood bank” by prof. Snigdha proposed an application for blood donor. In that application the donor can find the exact path by using GPS (Global Positioning System). The detail of blood donors will be saved private data and confidential data are only viewed by the administrator. They have methodologies like PHP, MY SQL, Android.

[2] In “MBB: A Life Saving Application” by Narendra Gupta has proposed a method to create a website with android application. In this application, it has been proposed that the donor is tracked by Geographic Information System (GIS). The purpose of their website is used to update their current system where data can only be viewed by authorized user. They contain two device type:

1)An android phone with android OS

2) A computer for website and database which is used to store the information about the donor.

[3] In “Android Based Health Application in Cloud Computing for Blood Bank” by Sayali Dhond has proposed android based application for blood donor, in which the donor’s information are stored in cloud. They user should request blood on the cloud and the information are sent to nearby hospital or blood donor who are register on cloud.

[4] In “The Optimization of Blood Donor Information and Management System bt Technopedia” by P. Priya has proposed a method of creating website with android application in which the blood donor can easily available within the required time. The donor who are nearby location are easily tracked by GIS. In this application the website is to update the information of donor who have already given blood in various hospital. While comparing to manual system, computer-based information system is time consuming, laborious.

S.NO	TITLE	AUTHORS	ABSTRACT	DRAWBACKS
1	Developing plasma	Aishwarya R Gow Jain	A plasma is a liquid portion the blood, over 55% of	• Internet: would require

	donor application using Function-asa-service in AWS	University, Department of MCA, computer science	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 antibodies 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.	an internet connection for the working of the website. • handle multiple requests at the same time
2	Optimization of Blood Donor Information ar	• K. Yamini, M. E(CSC), SVCET, Thirupachur, India	Emergency situations, such as accidents, create an immediate, critical need for specific blood type. In additic to emergency	• The accuracy of the location displayed on the map wa beyond the

	Management System	• R. Devi, Asst. Professor, SVCET, Thirupachur, India	requirements, advances in medicine have increased the need for blood in many ongoing treatments and elective surgeries. Despite increasing requirements for blood, only about 5% of the Indian population donates blood. In this paper we propose a new and efficient way to overcome such scenarios with our project. We have to create a new idea, just touch the button. Donor will be prompted to enter an individual's details, like name, phone number, and blood type. After that your contact details will appear in alphabetical order on the screen; the urgent time of a blood requirement, you can quickly check for contacts matching a particular or related blood group and reach out to them via Phone Call/SMS through the Blood donor App.	scope of this Project. • Only Android was used as a mobile operating system to test the application
3	Blood Bank Management Information System in India	• Vikas Kulshreshtha Research Scholar, • Dr.Sharad Maheshwari, Associate Professor	A blood bank is a bank of blood or blood components, gathered as a result of blood donation, stored and preserved for later use in blood transfusion. To provide web based communication there are numbers of online web based blood bank management system exists for communicating between department of blood centers and hospitals, to satisfy blood necessity, to buy, sale and stock the blood, to give information about this blood. Manual systems as compared to Computer Based Information Systems are time consuming, laborious, and costly. This paper	• Do not provide the better inventory solution to the end use • It requires an active internet connection.

			introduces the review of the main feature merits and demerits provided by the existin Web -Based Information System for Bloc Banks. This study shows the comparison c various existing system and provide som more idea for improve the existing systen First I will give some basic introductic about blood banks then I will try to provic comparative study of some existing we based blood bank system. After that I wi introduce some new idea for improving th existing techniques used in web based bloc bank system and at end I will conclude this pap	
4	A Research Pap on Bloc Donation Management System	• Devanjan I Srivastava • Utkarsh Tanwar • M.G.Krishna Ra • Priya Manohar Balraj Singh	Blood donation and transfusion has been a ever - serious issue and the shortage of bloc throughout the world has caused man people to lose their life. The lack of centralized system for blood donation majorly responsible for those losses. Now i the era of online and digital processes, th conventional methods of collecting bloc are absolute. An automated system required to manage the centers and showcase the information to the intereste parties. We have developed a website th singlehandedly solves all these issu related to blood donation and reception. W have designed a SQLite database as a integral part of the integrated framework store historical blood donation data in a centralized database for	• Internet Connection mandatory • There is r proper centralized database fc registered donor

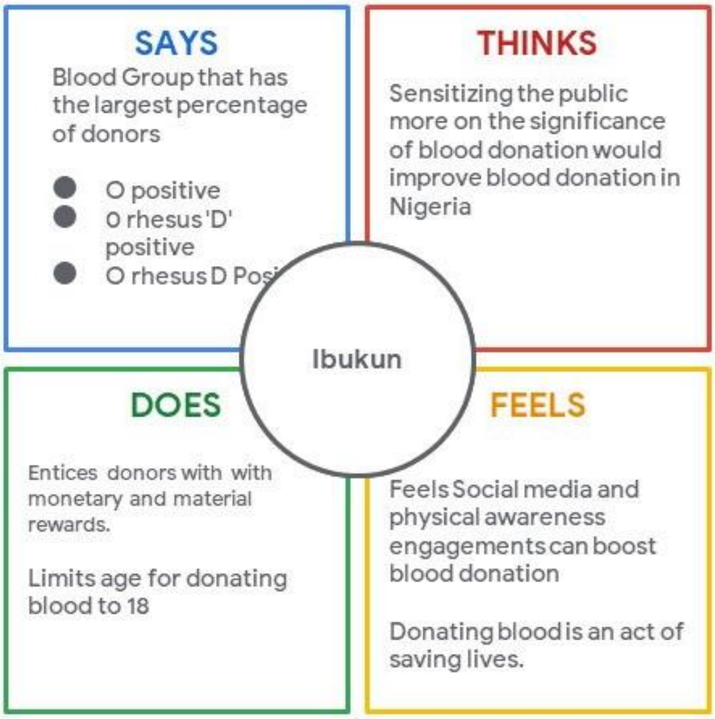
			analytical processing. Th proposed system would enab people to register as a donor t make themselves availab whenever in need of their bloo type. We have introduced search tab to search availab people ready to donate. In ou proposed system in the don registration, health - relate details would be updated in th blood management system databas for all to see	
5	A Study on Blood Bank Management	• A. Clemen Teena, K • Sankar • S. Kannan	'Blood Bank Information System' will be an information management system which helps to manage the records of donors and patients at a blood bank. The system will allow the authorized blood bank officer to login using a secret password and easily manage the records of the blood donors and the patients in need of blood	• No search filter available • UI improvem ent in Login page

2.2 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 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.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



SAYS

Blood Group that has the largest percentage of donors

- O positive
- O rhesus 'D' positive
- O rhesus D Posi

THINKS

Sensitizing the public more on the significance of blood donation would improve blood donation in Nigeria

Ibukun

DOES

Entices donors with with monetary and material rewards.

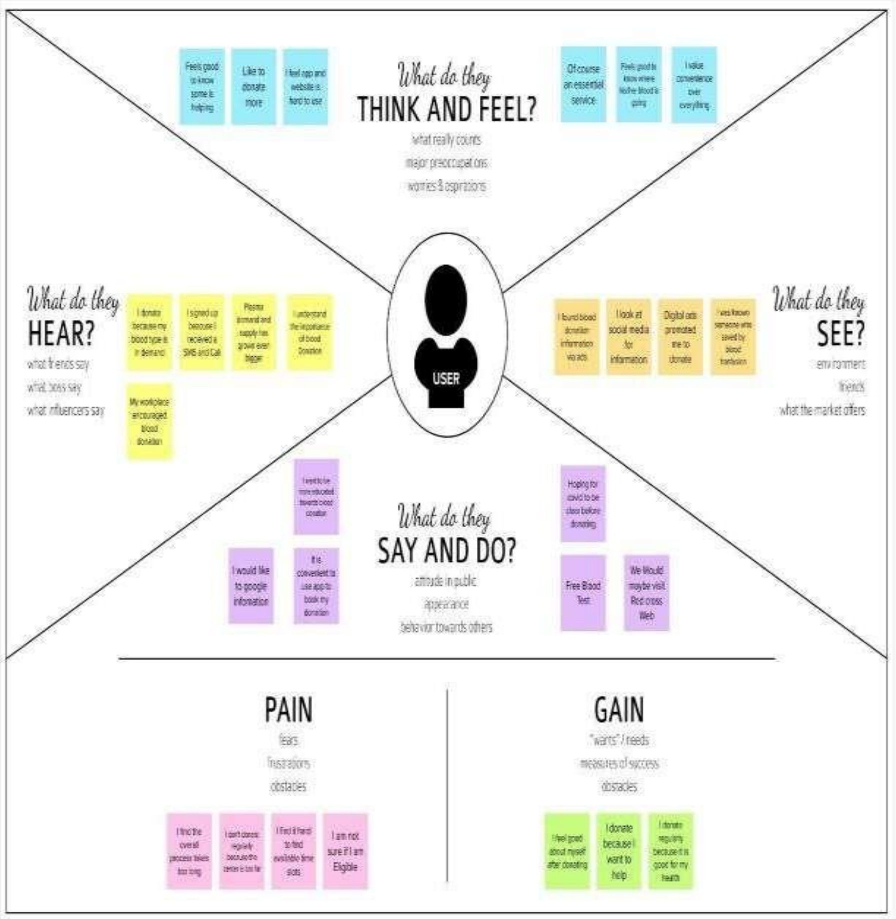
Limits age for donating blood to 18

FEELS

Feels Social media and physical awareness engagements can boost blood donation

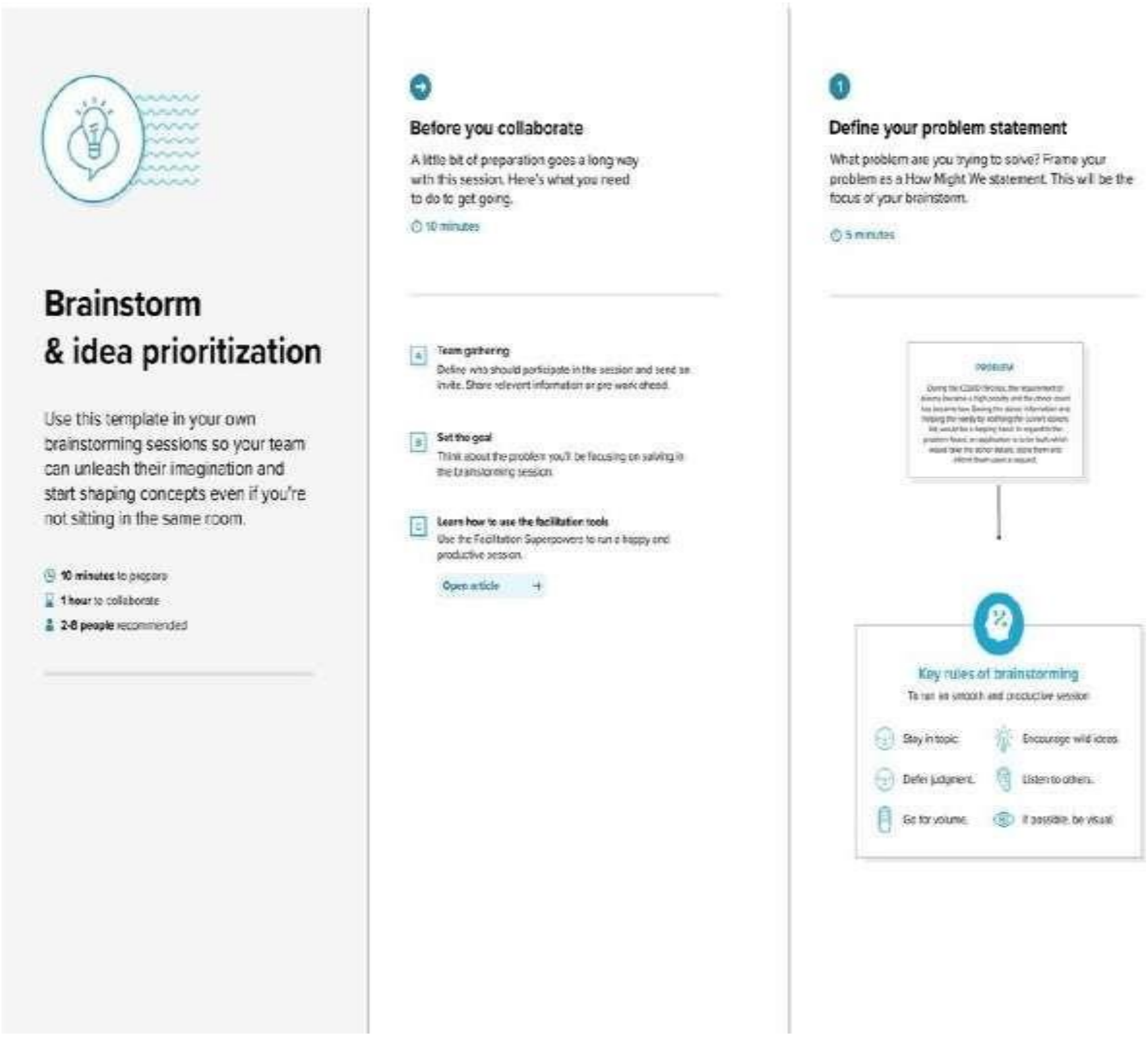
Donating blood is an act of saving lives.

Plasma Donor Application Empathy Map


Feels good to know some is helping
Like to donate more
I feel app and website is hard to use
What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Of course an essential service
Feels good to know where his/her blood is going
I value convenience over everything
What do they HEAR?
what friends say
what boss say
what influencers say
I donate because my blood type is in demand
I signed up because I received a SMS and Call
Plasma demand and supply has grown ever bigger
I understand the importance of blood Donation
My workplace encouraged blood donation
USER
I found blood donation information via ads
I look at social media for information
Digital ads promoted me to donate
I was known someone who saved by blood transfusion
What do they SEE?
environment
friends
what the market offers
I want to be more educated towards blood donation
I would like to google information
It is convenient to use app to book my donation
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Hoping for covid to be clear before donating
Free Blood Test
We Would maybe visit Red cross Web
PAIN
fears
frustrations
obstacles
I find the overall process takes too long
I can't donate regularly because the center is too far
I find it hard to find available time slots
I am not sure if I am Eligible
GAIN
"wants" / needs
measures of success
obstacles
I feel good about myself after donating
I donate because I want to help
I donate regularly because it is good for my health

3.2 IDEATION & BRAINSTORMING

Step 1: Team Gathering Collaboration and Select the Problem Statement



Brainstorm & idea prioritization

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

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

Before you collaborate

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

10 minutes

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

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

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Define your problem statement

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

5 minutes

PROBLEM

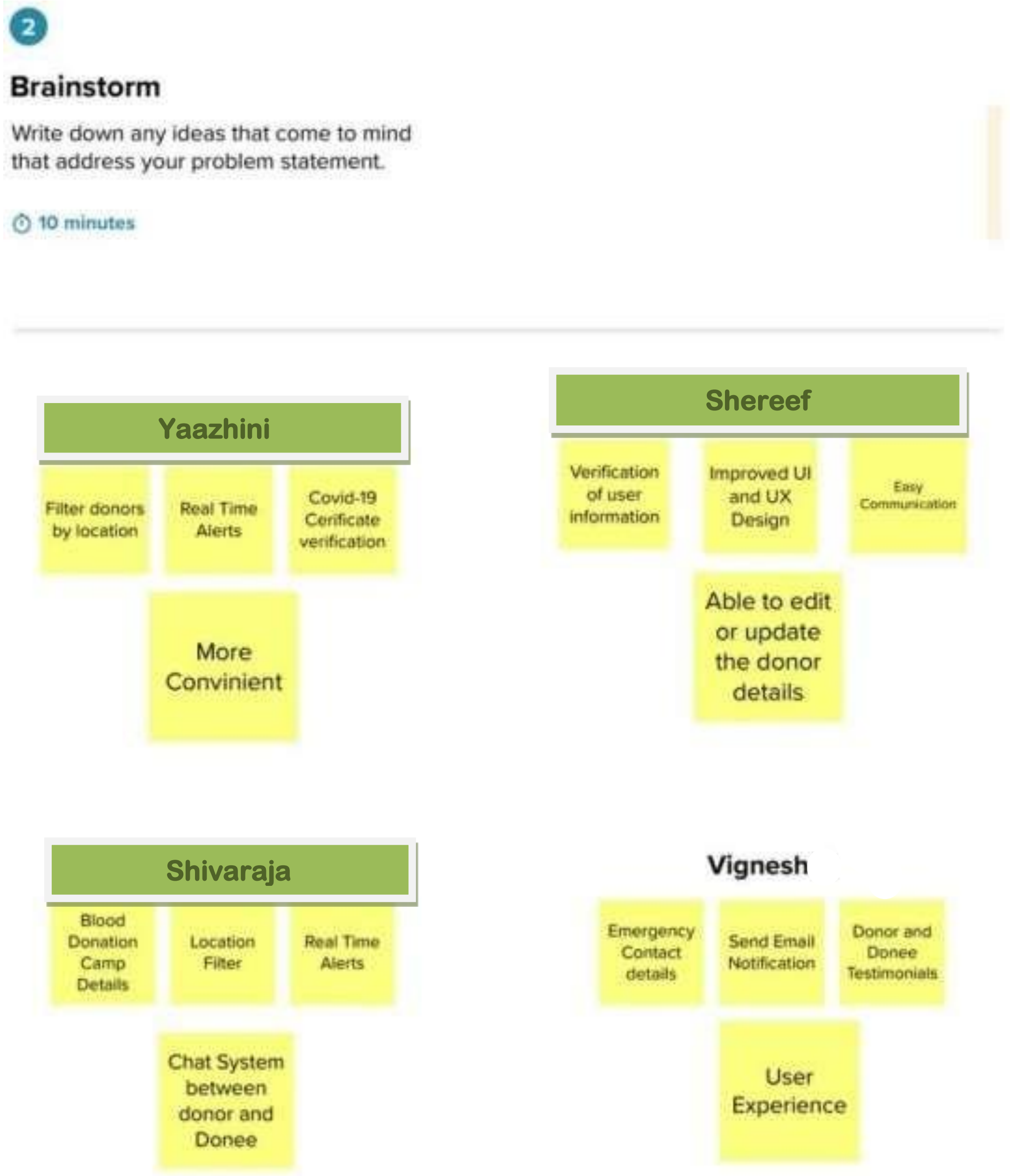
[illegible]

Key rules of brainstorming

To run an smooth and productive session

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

Step 2: BrainStorm And Idea Listing



2

Brainstorm

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

10 minutes

Yaazhini

- Filter donors by location
- Real Time Alerts
- Covid-19 Cerificate verification
- More Convinient

Shereef

- Verification of user information
- Improved UI and UX Design
- Easy Communication
- Able to edit or update the donor details

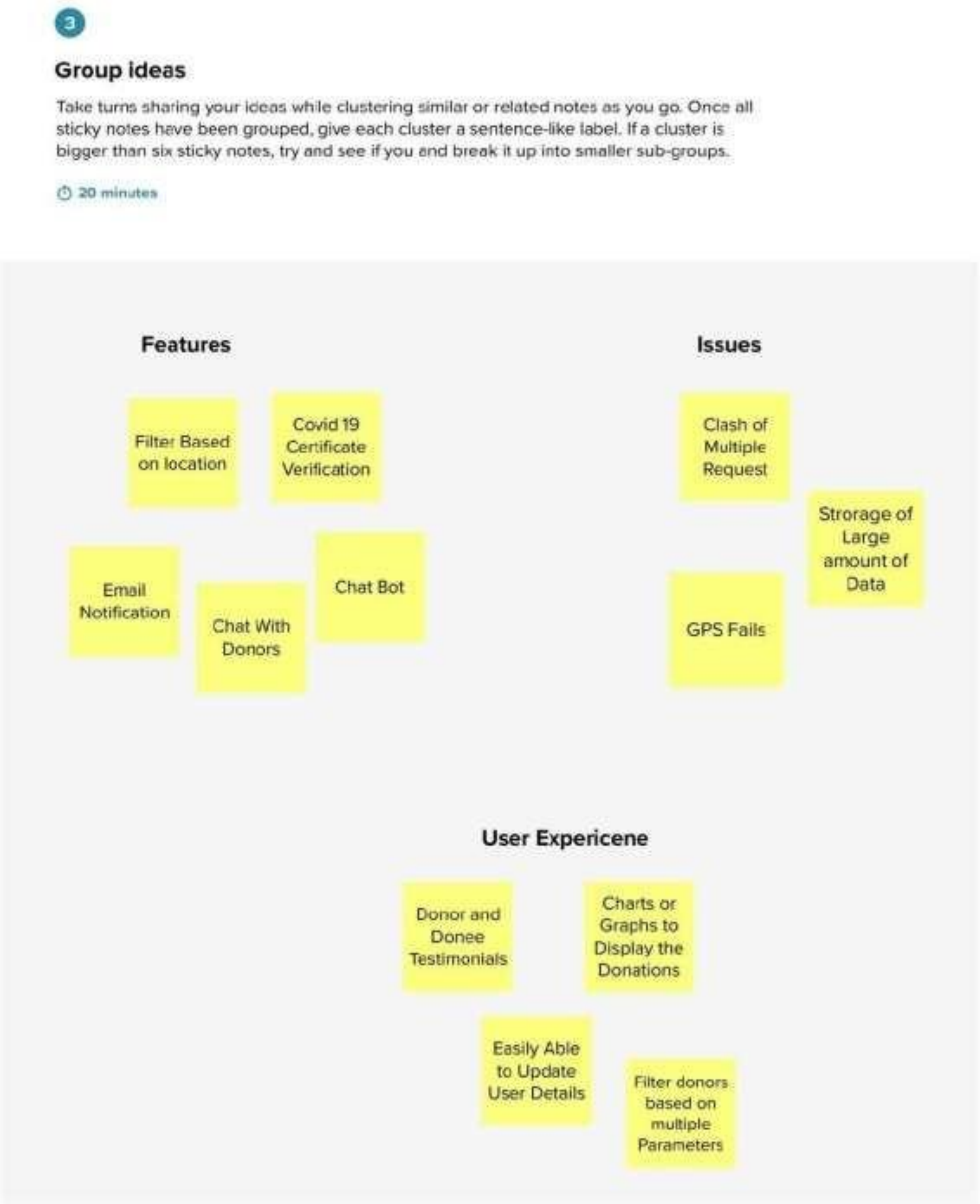
Shivaraja

- Blood Donation Camp Details
- Location Filter
- Real Time Alerts
- Chat System between donor and Donee

Vignesh

- Emergency Contact details
- Send Email Notification
- Donor and Donee Testimonials
- User Experience

Step 3: Grouping



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

Features

Filter Based on location

Covid 19 Certificate Verification

Email Notification

Chat With Donors

Chat Bot

Issues

Clash of Multiple Request

Strorage of Large amount of Data

GPS Fails

User Expericene

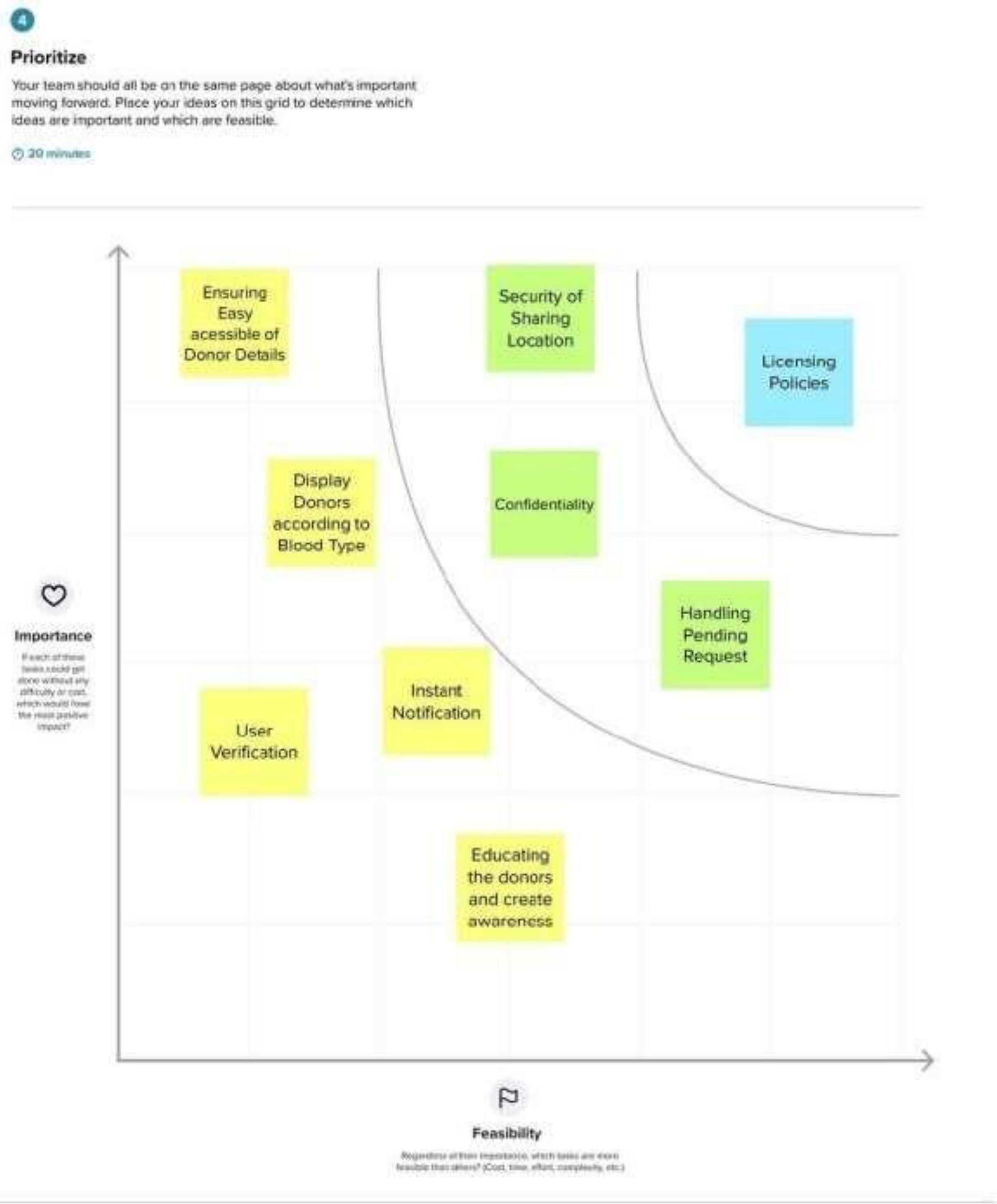
Donor and Donee Testimonials

Charts or Graphs to Display the Donations

Easily Able to Update User Details

Filter donors based on multiple Parameters

Step 4: Idea Prioritization



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

Ensuring Easy acessible of Donor Details
Security of Sharing Location
Licensing Policies
Display Donors according to Blood Type
Confidentiality
Handling Pending Request
Instant Notification
User Verification
Educating the donors and create awareness

Step 5:Top Ideas

Display Donor Detailsaccording to Blood Type and Location
Filter Conditions
Email Notification

3.3 PROPOSED SOLUTION

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.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• When the details are maintained manually, it is complicated for donors and patients. • Physical Man power is required to manage the Data and process the Plasma Donation. • In Pandemic situation, it is difficult to have manpower, so it is difficult to get the Plasma donor data. • Needed an Automated system to Manage donor and Patient data. • The data is needed to be accessible from anywhere and anytime
2.	Idea / Solution description	• Making a Web application which is user friendly as well as has more features for serving the people better. • Reduced workload by storing the details in cloud storage.

		• No Manpower / Remote Manpower only will be needed. • Data Availability for 24x7x365.
3.	Novelty / Uniqueness	• User friendly UI to access the web application by all the people • If a Donating user is available, they can request for plasma. • The web application will automatically send the email containing the Patient's contact details. • The Donor may contact the Patient and can reach the patient to donate the blood. • Voluntary donors can fill out a registration form and can get the Request Email on demand.
4.	Social Impact / user Satisfaction	• Impact between the users on the application is made easy communication and make them more secured and comfort • Find the donors in near places • Connect the donors and patients Easily. • With all of the authenticated information, this platform will assist the public in donating or obtaining their plasma needs.
5.	Business Model (Revenue Model)	• By collaborating with government and organizing Plasma Donation Camps and store them instead requesting Plasma on demand
6.	Scalability of the Solution	• The main goal of the application is to provide high Scalability by given more option for user to select their interest(donate/assist) • The aim is to build a web application using Cloud with advanced features that will help to overcome the barrier between Plasma bank, Donor and Patient • Since the project uses IBM DB2 database it can handle with multiple requests in various regions • As this is a web application and uses cloud storage, any further enhancements in technology can be incorporated within this application. • Chatbot for Queries • Genuineness of the Patient will be tested

3.4 PROBLEM SOLUTION FIT

1. CUSTOMER SEGMENTS(S)

- Donors
- Patient
- Hospitals

2. JOB-TO-BE-DONE

- Difficult to find donors at the right time
- Donors not aware (plasma requirements

3. TRIGGERS

Blood donation improves (saves lives and enhances soci solidarity. It is also influence by increasing deaths due t unavailability of plasma required times.

4. EMOTIONS:

Before:

Patient /Hospital find it har to get a right resource to get plasma leaving them upset.

After:

The donors and custome haves a feeling of satisfaction

5. AVAILABLE SOLUTIONS

The existing application use only collecting details pf dono but it does not notify them at right time. Our solution is buildir a
website that notifies the dono at a right time.

6. CUSTOMER CONSTRAINTS

- Regular interv connection
- Donor health condition
- Unavailability of plasma

7. BEHAVIOUR

The customer comes forward to

- Attend plasma donatic camps
- Donate plasma

8. CHANNELS OF BEHAVIOR

ONLINE:

Can use the website to find donors

OFFLINE:

Can use the record maintain by th hospital

9. PROBLEM ROOT CAUSE

- Not able to find donors at th right time of emergency
- Count of donors has bee tremendously decreasing since hospital management couldn’t contact them

10. YOUR SOLUTION

Creating website which will provic information about the available dono and plasma. If not available the custome will be notified when plasma is available

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREENTS

FR No	Functional Requiremen (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through mobile/ laptop/ PC Registration through telegram group
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Donor Notification	Get notification through register mobile number Get notification through register Email
FR-4	Plasma neede details(person)	Availability details in App Availability details in telegram group
FR-5	Plasma availability(blood)	Availability details in App Availability details in telegram group

4.2 NON-FUNCTIONAL REQUIREENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	• Can new user quickly adapt to the softwar without helpless • the most common operations streamline to be performed quickly
NFR-2	**Security**	• The system had user or role based securit • any operations done by user will kee private
NFR-3	**Reliability**	• Whenever the user change his scheduled • use mobile and desktop anywhere
NFR-4	**Performance**	• The performance of the app is in high leve because it can holed only few data so i performance will fast

NFR-5	**Availability**	• User can use mobile and desktop anywhere in network • User can use application 24/7
NFR-6	**Scalability**	• The capacity of an app is handle by cloud so it has high scalability and elasticity

5. PROJECT DESIGN

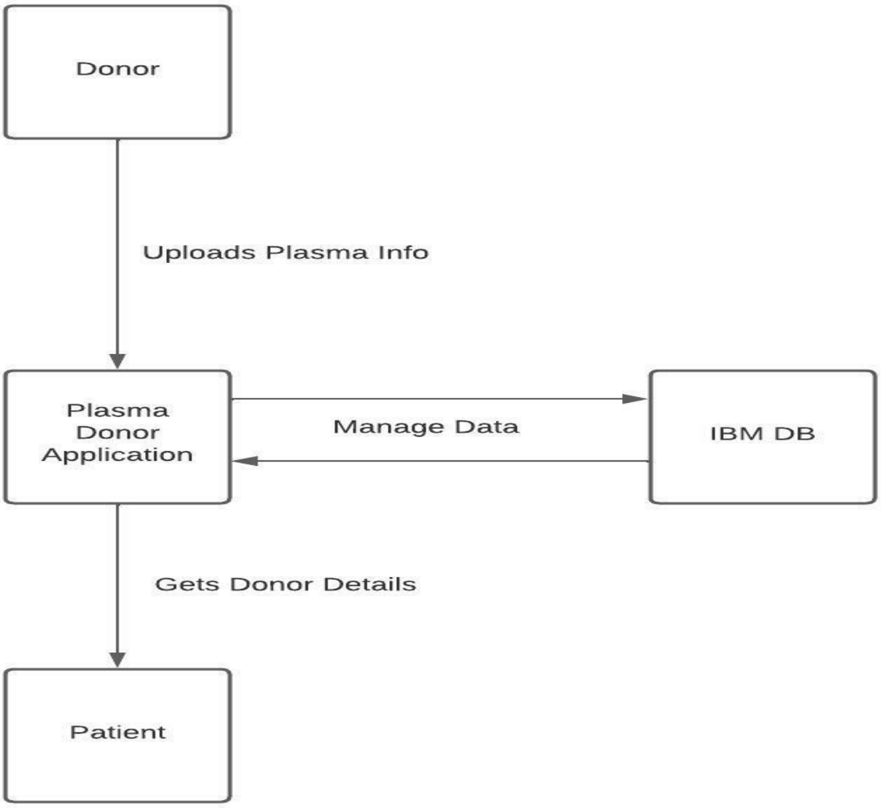
5.1 DATA FLOW DIAGRAMS

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.

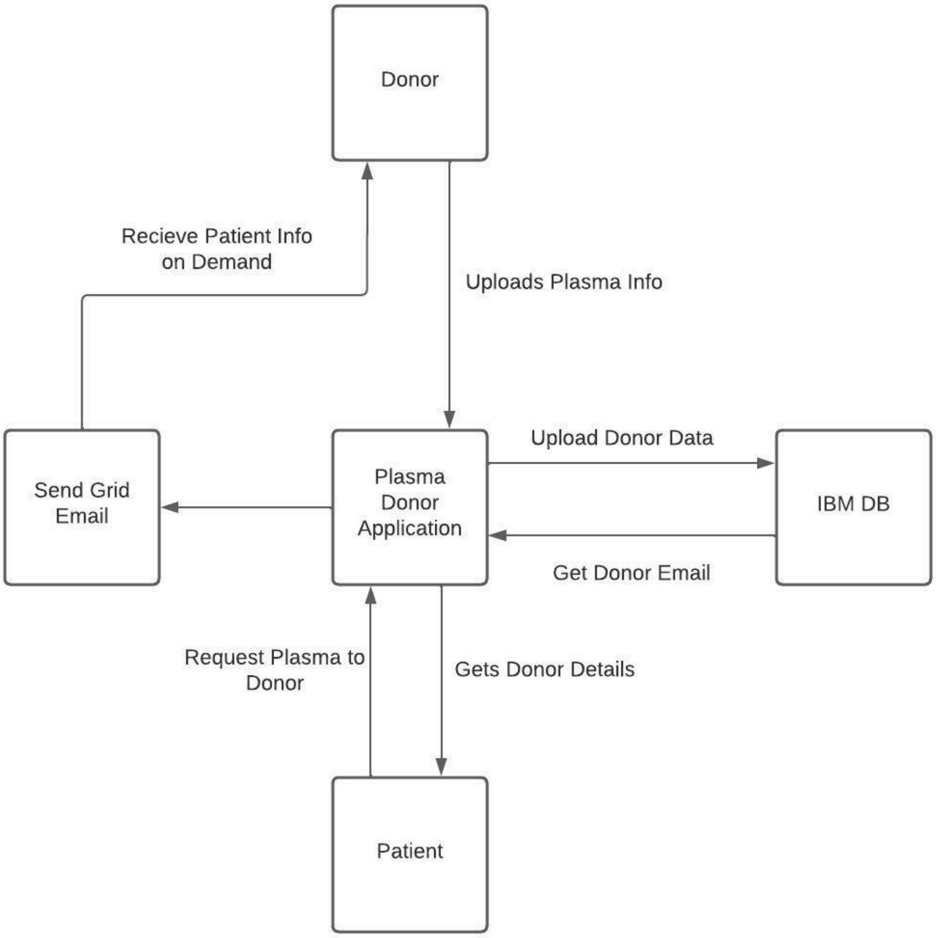
STEPS:

1. Donor can enter their details and check their eligibility.
2. Hospital In-Charge enter their hospital details and register themselves.
3. Recipients can enter their details and book their slots.
4. After Donor’s donation finished, In-charge update the details in database.
5. After Recipient’s request for plasma, In-charge has to allocate the the appropriate plasma for recipient.
6. After the process finished, all users enter their feedback to their appropriate requests.
7. All the changes can enter into DB2.

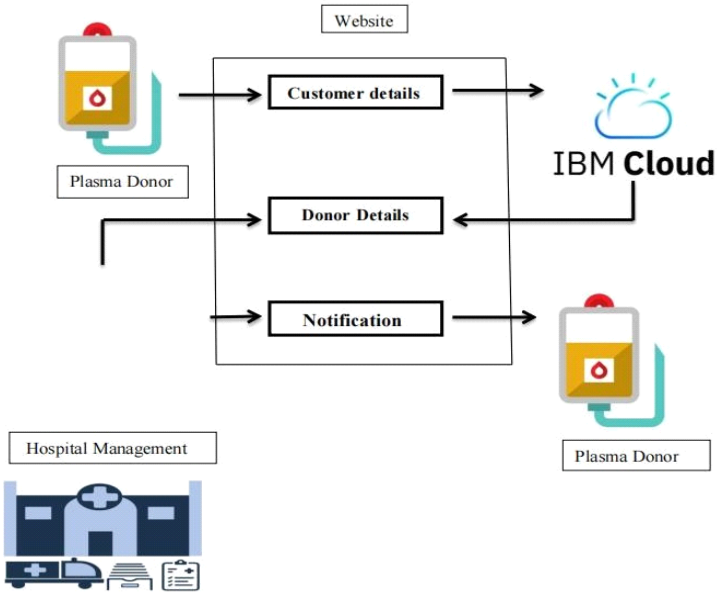
Level 0:

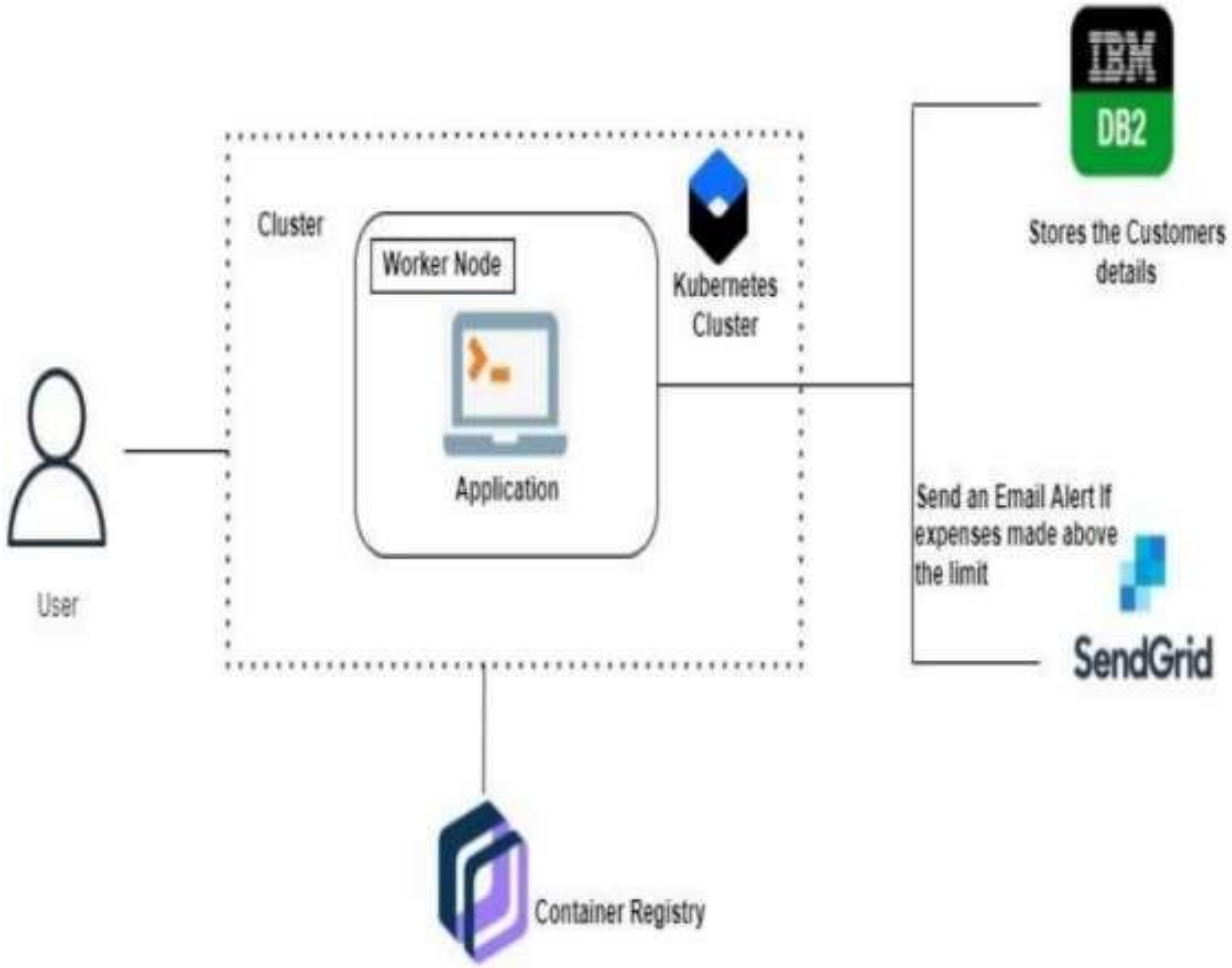


Level 1:

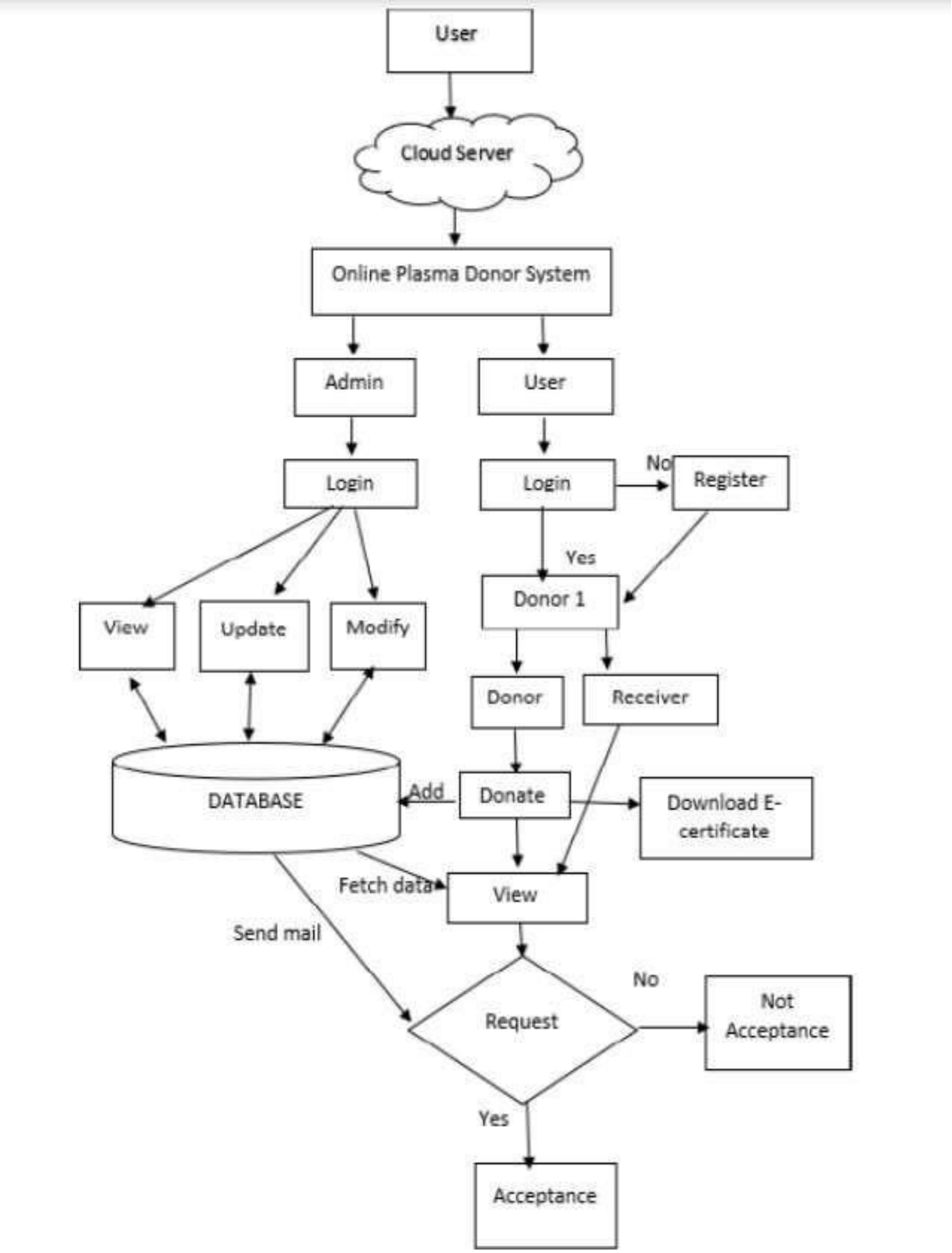


5.2 SOLUTION & TECHNICAL ARCHITECTURE

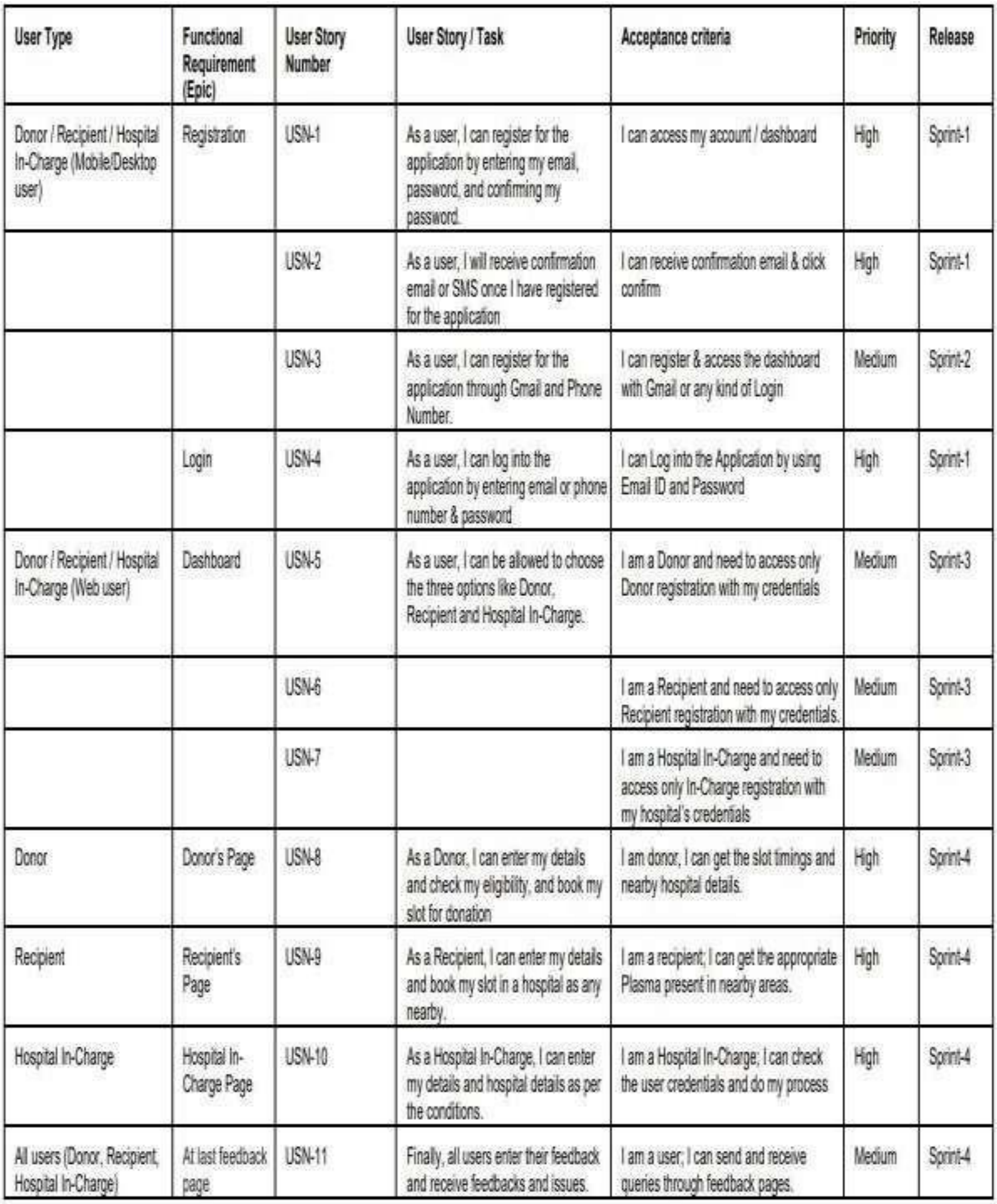




SOLUTION ARCHITECTURE



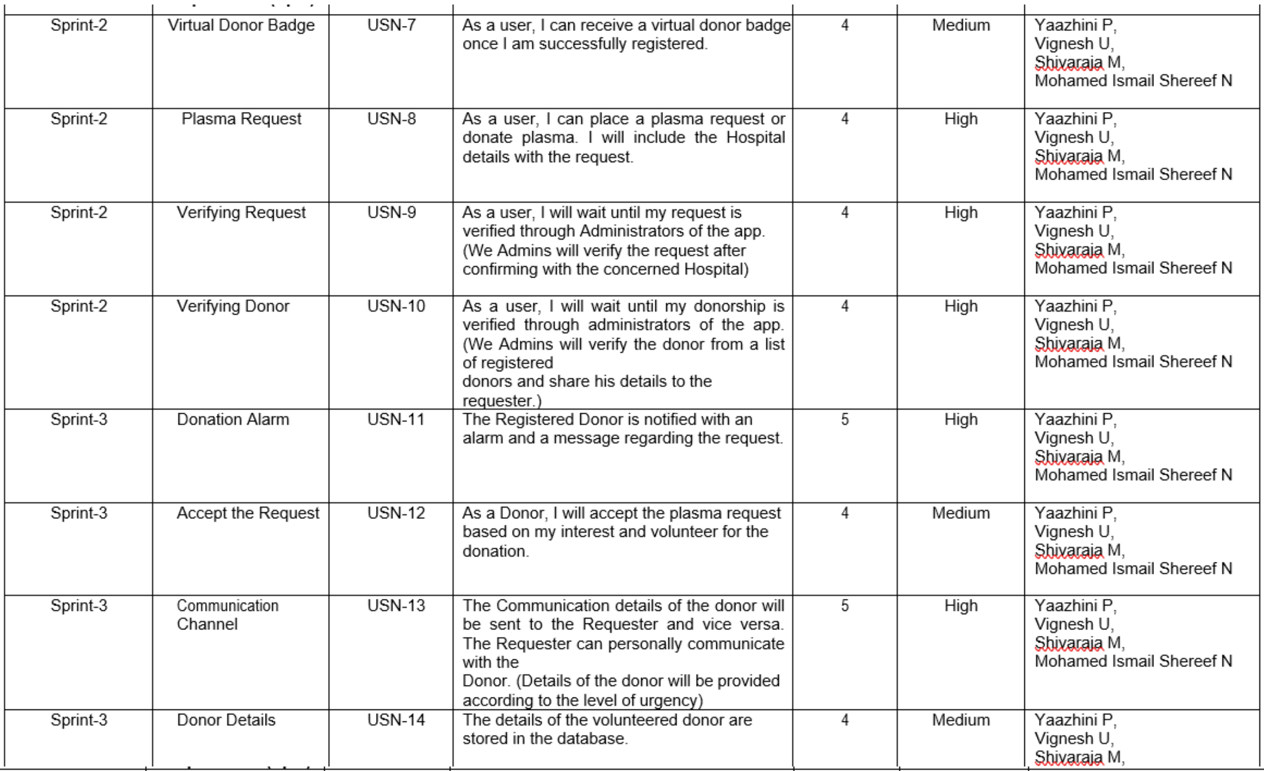
5.3 USER STORIES



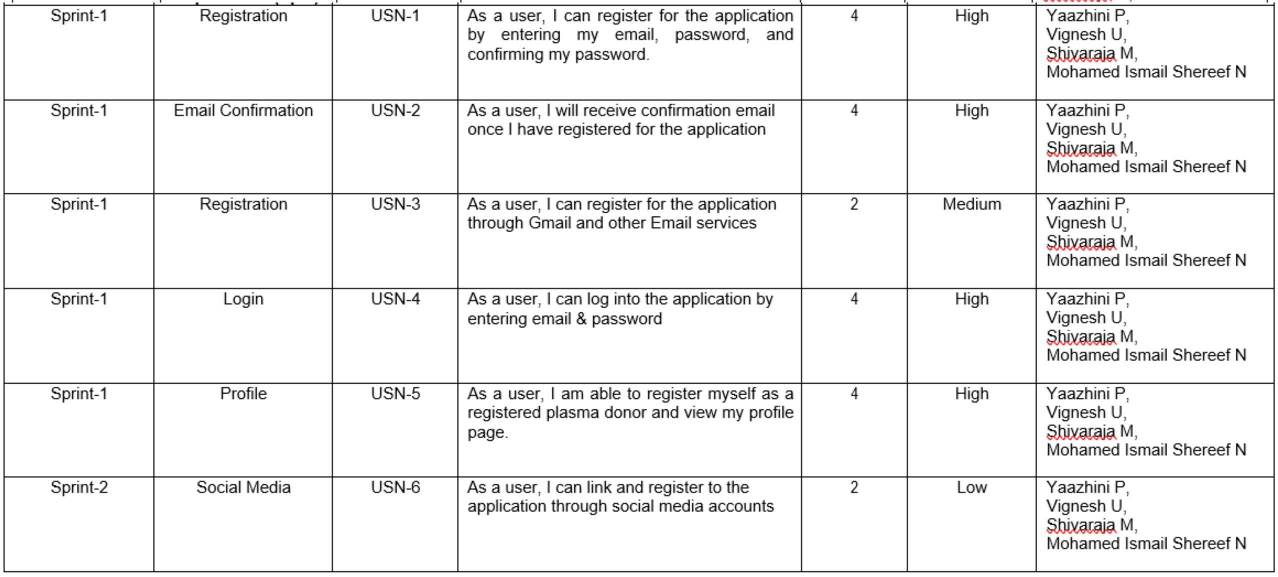
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Donor / Recipient / Hospital In-Charge (Mobile/Desktop user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email or SMS 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 Gmail and Phone Number.	I can register & access the dashboard with Gmail or any kind of Login	Medium	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email or phone number & password	I can Log into the Application by using Email ID and Password	High	Sprint-1
Donor / Recipient / Hospital In-Charge (Web user)	Dashboard	USN-5	As a user, I can be allowed to choose the three options like Donor, Recipient and Hospital In-Charge.	I am a Donor and need to access only Donor registration with my credentials	Medium	Sprint-3
		USN-6		I am a Recipient and need to access only Recipient registration with my credentials.	Medium	Sprint-3
		USN-7		I am a Hospital In-Charge and need to access only In-Charge registration with my hospital's credentials	Medium	Sprint-3
Donor	Donor's Page	USN-8	As a Donor, I can enter my details and check my eligibility, and book my slot for donation	I am donor, I can get the slot timings and nearby hospital details.	High	Sprint-4
Recipient	Recipient's Page	USN-9	As a Recipient, I can enter my details and book my slot in a hospital as any nearby.	I am a recipient; I can get the appropriate Plasma present in nearby areas.	High	Sprint-4
Hospital In-Charge	Hospital In-Charge Page	USN-10	As a Hospital In-Charge, I can enter my details and hospital details as per the conditions.	I am a Hospital In-Charge; I can check the user credentials and do my process	High	Sprint-4
All users (Donor, Recipient, Hospital In-Charge)	At last feedback page	USN-11	Finally, all users enter their feedback and receive feedbacks and issues.	I am a user; I can send and receive queries through feedback pages.	Medium	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION



Sprint-2	Virtual Donor Badge	USN-7	As a user, I can receive a virtual donor badge once I am successfully registered.	4	Medium	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-2	Plasma Request	USN-8	As a user, I can place a plasma request or donate plasma. I will include the Hospital details with the request.	4	High	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-2	Verifying Request	USN-9	As a user, I will wait until my request is verified through Administrators of the app. (We Admins will verify the request after confirming with the concerned Hospital)	4	High	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-2	Verifying Donor	USN-10	As a user, I will wait until my donorship is verified through administrators of the app. (We Admins will verify the donor from a list of registered donors and share his details to the requester.)	4	High	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-3	Donation Alarm	USN-11	The Registered Donor is notified with an alarm and a message regarding the request.	5	High	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-3	Accept the Request	USN-12	As a Donor, I will accept the plasma request based on my interest and volunteer for the donation.	4	Medium	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-3	Communication Channel	USN-13	The Communication details of the donor will be sent to the Requester and vice versa. The Requester can personally communicate with the Donor. (Details of the donor will be provided according to the level of urgency)	5	High	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-3	Donor Details	USN-14	The details of the volunteered donor are stored in the database.	4	Medium	Yaazhini P, Vignesh U, Shivaraja M,



Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	4	High	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-1	Email Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	4	High	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-1	Registration	USN-3	As a user, I can register for the application through Gmail and other Email services	2	Medium	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password	4	High	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-1	Profile	USN-5	As a user, I am able to register myself as a registered plasma donor and view my profile page.	4	High	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-2	Social Media	USN-6	As a user, I can link and register to the application through social media accounts	2	Low	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N

Sprint-4	Support	USN-15	As a user, I can chat with a chatbot regarding my queries and doubts.	3	Medium	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-4	Grievances and FAQ	USN-16	As a user, I can post my worries and grievances in the comment section. I can also find Frequently asked Questions with answers in the FAQ section.	3	Medium	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-4	Certificate and Rewards	USN-17	As a donor, I will receive an e-certificate after donations. Virtual rewards are also provided to the donor.	3	Low	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-4	About	USN-18	As a user, I will find about the importance of plasma donation in this section of the application.	3	Medium	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-4	Administrator		We admins transaction verification will approve all the application after the plasma proper.	3	High	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-4			We admins will update the plasma availability and donor count periodically.	3	Medium	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N
Sprint-4			We admins will give fine touch to the application based on any updates needed in the future.	3	Medium	Yaazhini P, Vignesh U, Shivaraja M, Mohamed Ismail Shereef N

6.2 SPRINT DELIVERY SCHDULE

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	18	6 Days	24 Oct 2022	29 Oct 2022	18	29 Oct 2022
Sprint -2	18	6 Days	31 Oct 2022	05 Nov 2022	18	05 Nov 2022
Sprint -3	18	6 Days	07 Nov 2022	12 Nov 2022	18	12 Nov 2022
Sprint -4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

We have a 6-day sprint duration, and the velocity of the team is 18 (points per sprint), except the Sprint-4 is 20. To calculate the team's average velocity (AV) per iteration unit (story points per day)

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

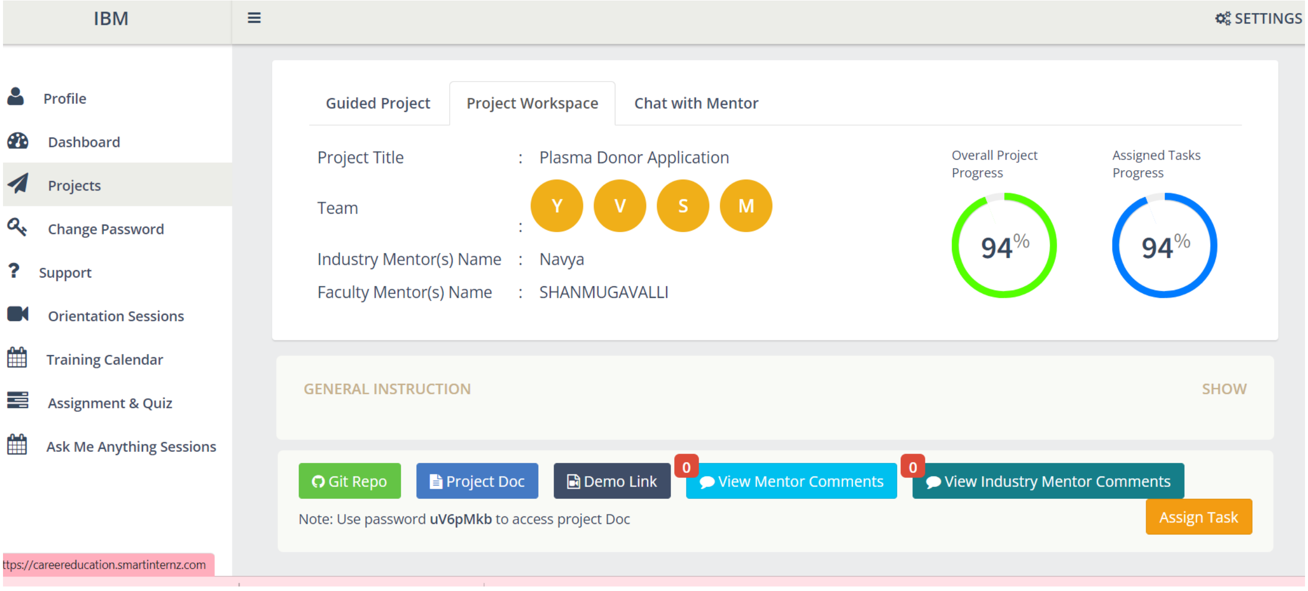
Number of Sprints	Sprint-1	Sprint-2	Sprint-3	Sprint-4
Total Story Points	18	18	18	20
Duration	6 Days	6 Days	6 Days	6 Days
Average Velocity per Sprint	AV= 18/6 = 3	AV= 18/6 =3	AV= 18/6 =3	AV= 20/6 = 3.33

Total number of days = sprint 1 + sprint 2 + sprint 3 + sprint 4 = 6 + 6 + 6 + 6 = 24

Total number of story points = 18 + 18 + 18 + 20 = 74

Average velocity per sprint = 74 / 24 ~= 3.083333 = 3

6.3 REPORTS FROM JIRA



IBM ≡ SETTINGS

- Profile
- Dashboard
- Projects
- Change Password
- Support
- Orientation Sessions
- Training Calendar
- Assignment & Quiz
- Ask Me Anything Sessions

Guided Project | Project Workspace | Chat with Mentor

Project Title : Plasma Donor Application

Team : Y V S M

Industry Mentor(s) Name : Navya

Faculty Mentor(s) Name : SHANMUGAVALLI

Overall Project Progress 94%

Assigned Tasks Progress 94%

GENERAL INSTRUCTION SHOW

Git Repo | Project Doc | Demo Link | 0 View Mentor Comments | 0 View Industry Mentor Comments

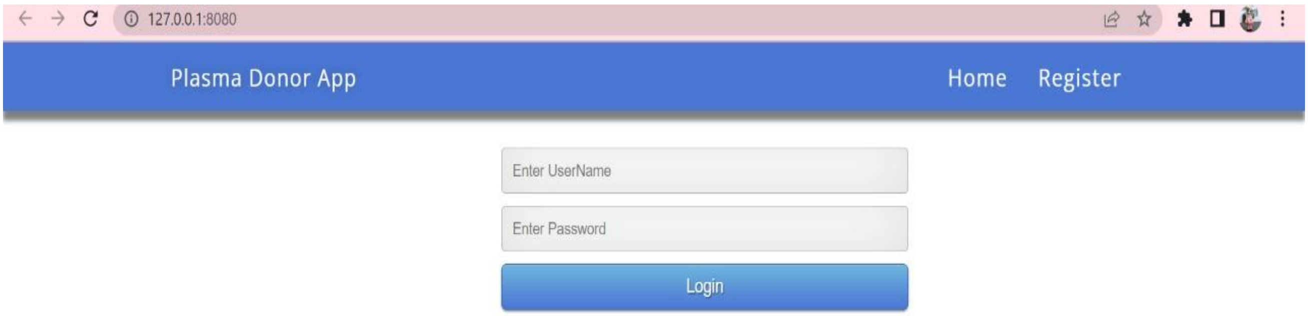
Note: Use password **uV6pMkb** to access project Doc

Assign Task

ttps://careereducation.smartinternz.com

7. CODING & SOLUTIONING

7.1 FEATURE 1- LOGIN



127.0.0.1:8080

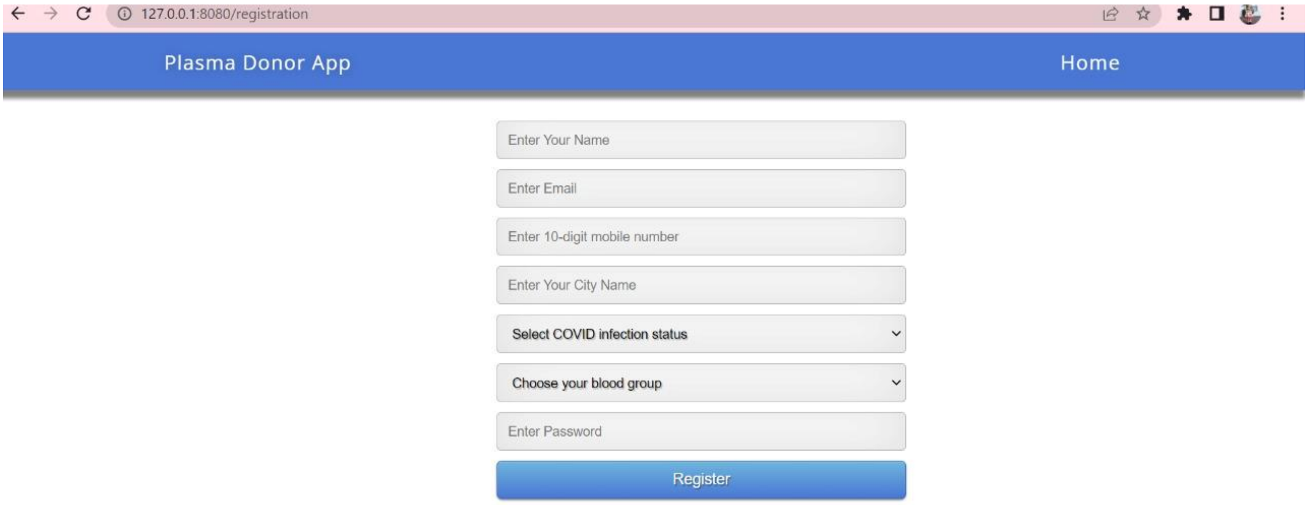
Plasma Donor App   Home   Register

Enter UserName

Enter Password

Login

REGISTER



127.0.0.1:8080/registration

Enter Your Name

Enter Email

Enter 10-digit mobile number

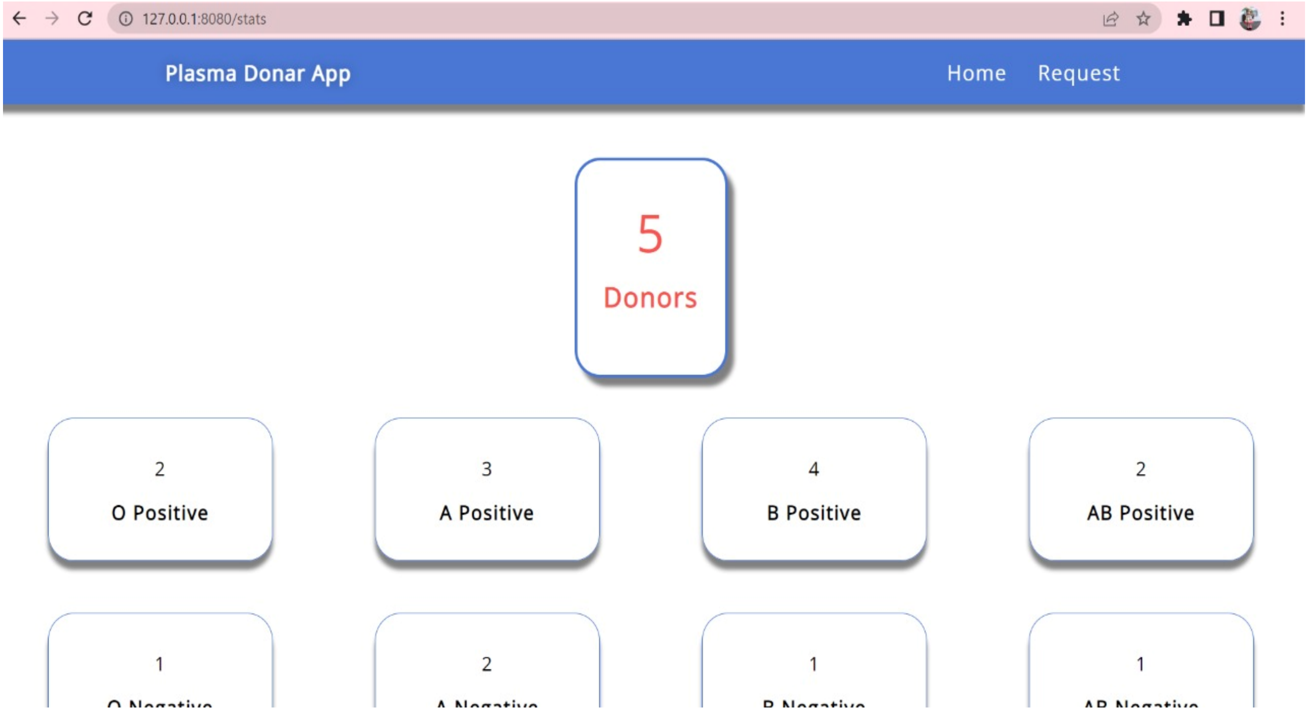
Enter Your City Name

Select COVID infection status

Choose your blood group

Enter Password

FEATURE 2- HOME PAGE

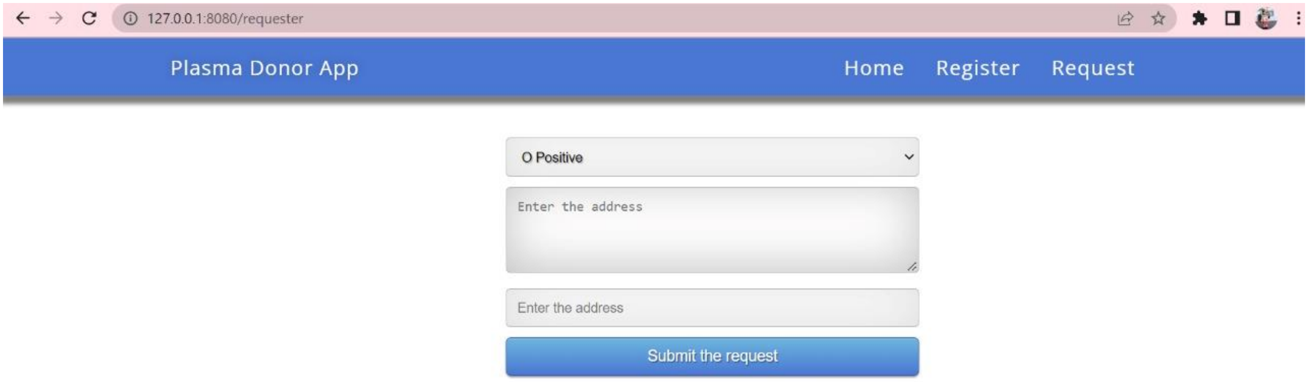


127.0.0.1:8080/stats

5
Donors

2 O Positive	3 A Positive	4 B Positive	2 AB Positive
1 O Negative	2 A Negative	1 B Negative	1 AB Negative

SEARCH ACCORDING BLOOD TYPE AND LOCATION

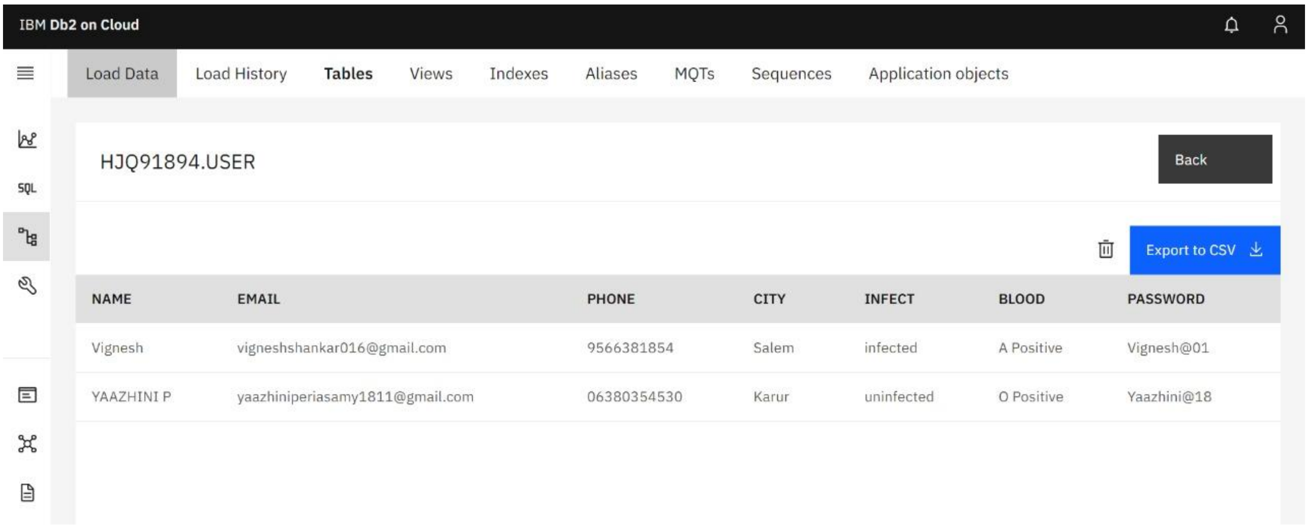


O Positive

Enter the address

Enter the address

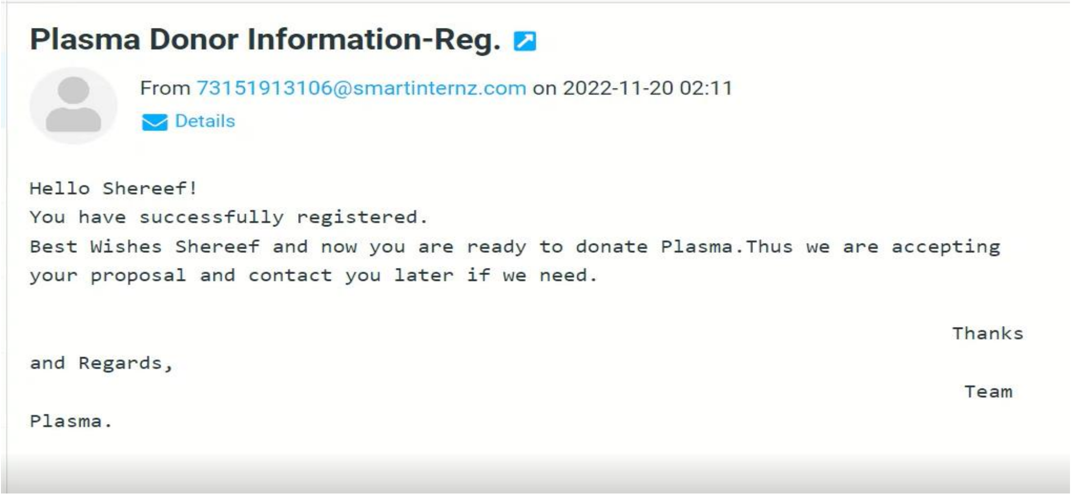
FEATURE 3- IBM DATABASE



HJQ91894.USER

NAME	EMAIL	PHONE	CITY	INFECT	BLOOD	PASSWORD
Vignesh	vigneshshankar016@gmail.com	9566381854	Salem	infected	A Positive	Vignesh@01
YAAZHINI P	yaazhiniperiasamy1811@gmail.com	06380354530	Karur	uninfected	O Positive	Yaazhini@18

FEATURE 4- EMAIL AUTHENTICATION



Plasma Donor Information-Reg.

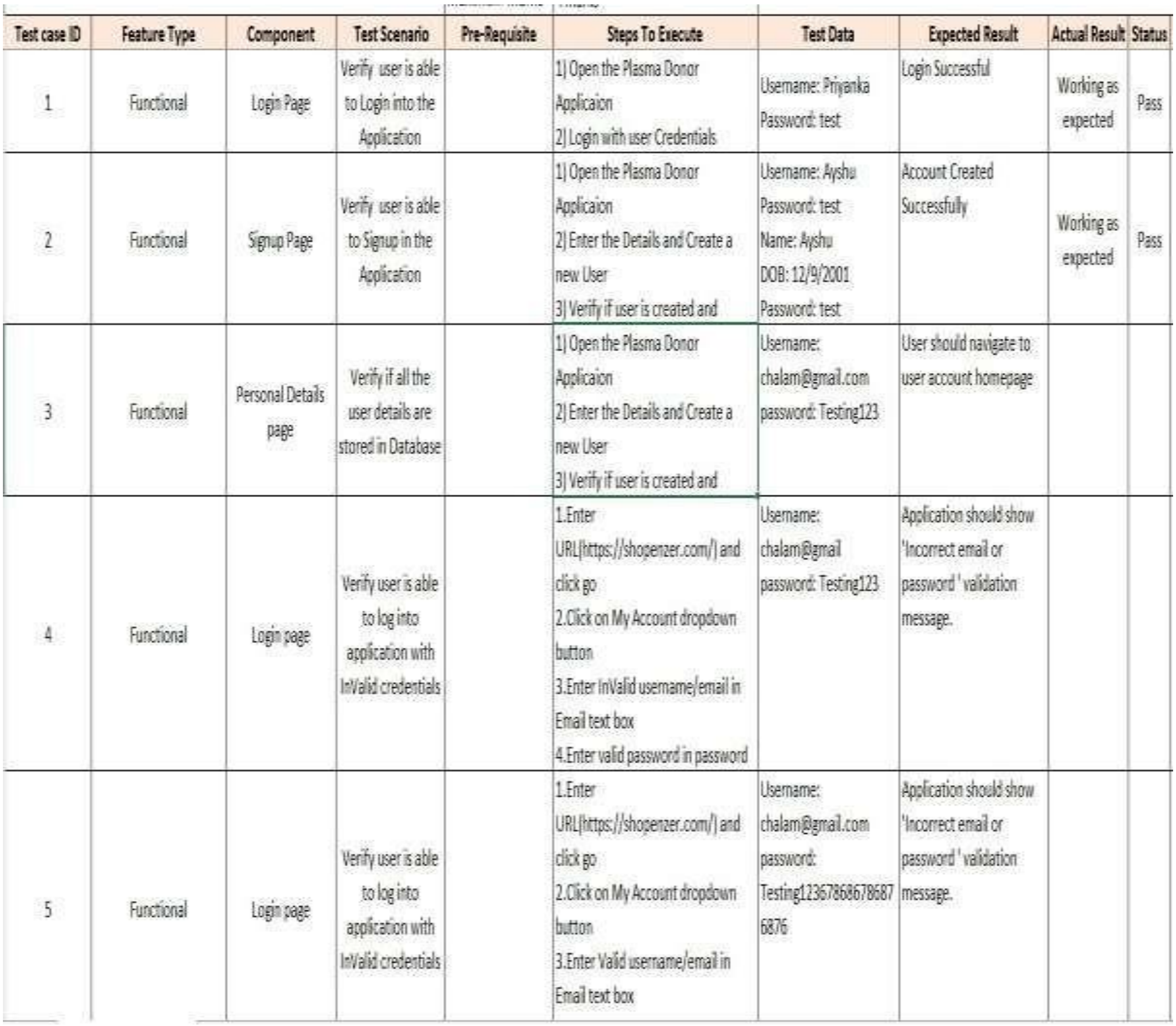
From 73151913106@smartinternz.com on 2022-11-20 02:11

Hello Shereef!
You have successfully registered.
Best Wishes Shereef and now you are ready to donate Plasma.Thus we are accepting your proposal and contact you later if we need.

Thanks
and Regards,
Team
Plasma.

8. TESTING

8.1 TEST CASES



Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status
1	Functional	Login Page	Verify user is able to Login into the Application		1) Open the Plasma Donor Applicaion 2) Login with user Credentials	Username: Priyanka Password: test	Login Successful	Working as expected	Pass
2	Functional	Signup Page	Verify user is able to Signup in the Application		1) Open the Plasma Donor Applicaion 2) Enter the Details and Create a new User 3) Verify if user is created and	Username: Ayshu Password: test Name: Ayshu DOB: 12/9/2001 Password: test	Account Created Successfully	Working as expected	Pass
3	Functional	Personal Details page	Verify if all the user details are stored in Database		1) Open the Plasma Donor Applicaion 2) Enter the Details and Create a new User 3) Verify if user is created and	Username: chalam@gmail.com password: Testing123	User should navigate to user account homepage		
4	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password	Username: chalam@gmail password: Testing123	Application should show 'Incorrect email or password ' validation message.		
5	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box	Username: chalam@gmail.com password: Testing123678686786876876	Application should show 'Incorrect email or password ' validation message.		

Test Scenarios

1 Verify user is able to see login page
2 Verify user is able to login to application or not?
3 Verify user is able to navigate to create your account page?
4 Verify user is able to recovery password
5 Verify login page elements

Search

1. Verify user is able to search by entering keywords in search box
2. Verify user is able to see suggestions based on keyword entered in search box
3. Verify user is able to see related auto suggestions displaying based on keyword entered in search box
4. Verify user is able to see no matches found message when no results are matching with entered keyword
5. Verify user is able to see search detailed page when nothing entered in textbox

8.2 USER ACCEPTANCE TESTING

1. Purpose of Document

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

2. Defect Analysis

Resolution	Severin 1	Severin 2	Severity 3	Severity 4	Subtotal
By Design	5	0	0	0	5
Duplicate	1	0	0	0	1
External	0	0	0	0	0
Fi‹ed	3	D	0	0	3
Not Re§f0dUC9d	2	0	0	0	2
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	(D	0	0	0	I6

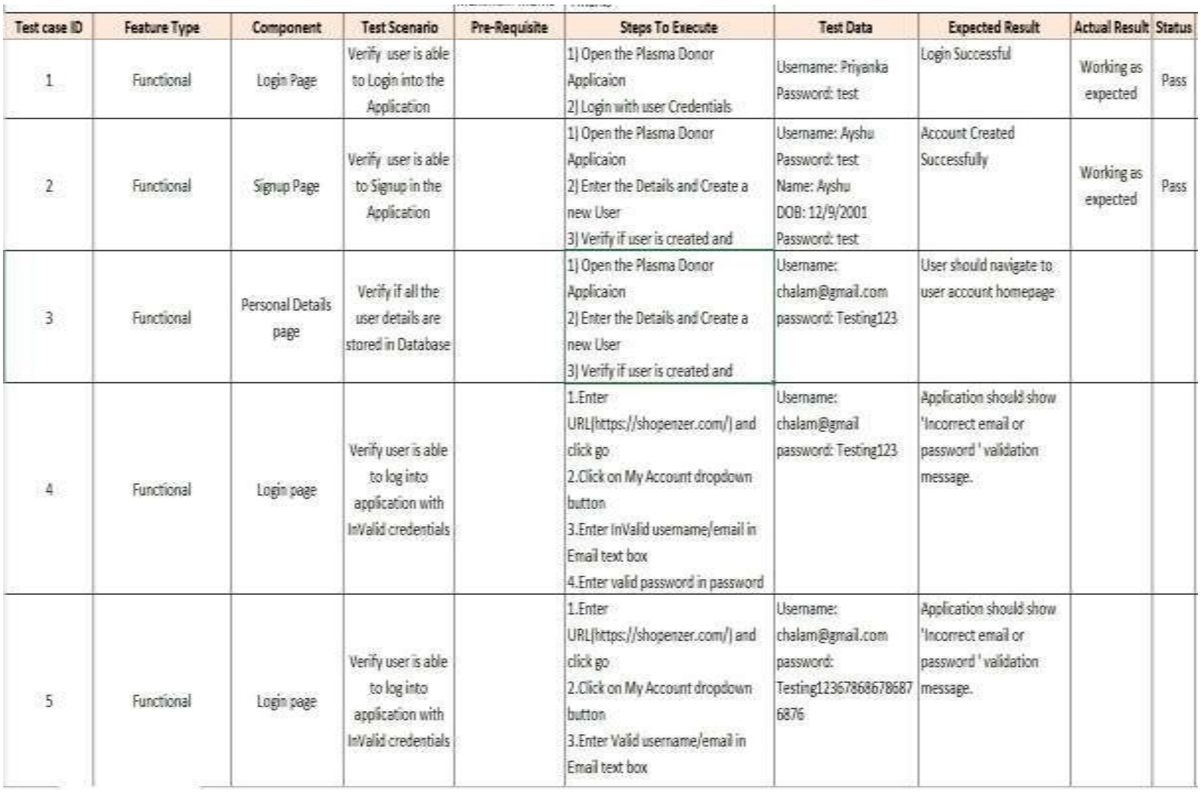
3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	0	0	0	0
Client Application	5	0	0	5
Security	0	0	0	0
Outsource Shipping	0	0	0	0
Exception Reposing	0	0	0	0

9. RESULTS

9.1 PERFORMANCE METRICS



Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status
1	Functional	Login Page	Verify user is able to Login into the Application		1) Open the Plasma Donor Applicaion 2) Login with user Credentials	Username: Priyanka Password: test	Login Successful	Working as expected	Pass
2	Functional	Signup Page	Verify user is able to Signup in the Application		1) Open the Plasma Donor Applicaion 2) Enter the Details and Create a new User 3) Verify if user is created and	Username: Ayshu Password: test Name: Ayshu DOB: 12/9/2001 Password: test	Account Created Successfully	Working as expected	Pass
3	Functional	Personal Details page	Verify if all the user details are stored in Database		1) Open the Plasma Donor Applicaion 2) Enter the Details and Create a new User 3) Verify if user is created and	Username: chalam@gmail.com password: Testing123	User should navigate to user account homepage		
4	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password	Username: chalam@gmail password: Testing123	Application should show 'Incorrect email or password ' validation message.		
5	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box	Username: chalam@gmail.com password: Testing123678686786876876	Application should show 'Incorrect email or password ' validation message.		

10. ADVANTAGES & DISADVANTAGES

1. ADVANTAGES

The project is identified by the merits of the system offered to the user. The merits of this project are as follows; -

- It's a web-enabled project.
- This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.
- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of any new creation, data entry or updation so that the user cannot enter the invalid data, which can create problems at later date.
- Sometimes the user finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records. Moreover there is restriction for his that he cannot change the primary data field. This keeps the validity of the data to longer e0tent.
- User is provided the option of monitoring the records he entered earlier. He can see the desired records with the variety of options provided by him.
- From every part of the project the user is provided with the links through framing so that he can go from one option of the project to other as per the requirement. This is bound to be simple and very friendly as per the user is concerned. That is" we can sat that the project is user friendly which is one of the primary concerns of any good project.
- Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.
- Decision making process would be greatly enhanced because of faster processing of information since data collection from information available on computer takes much less time then manual system.
- Allocating of sample results becomes much faster because at a time the user can see the records of last years.
- Easier and faster data transfer through latest technology associated with the computer and communication.
- Through these features it will increase the efficiency, accuracy and transparency

.

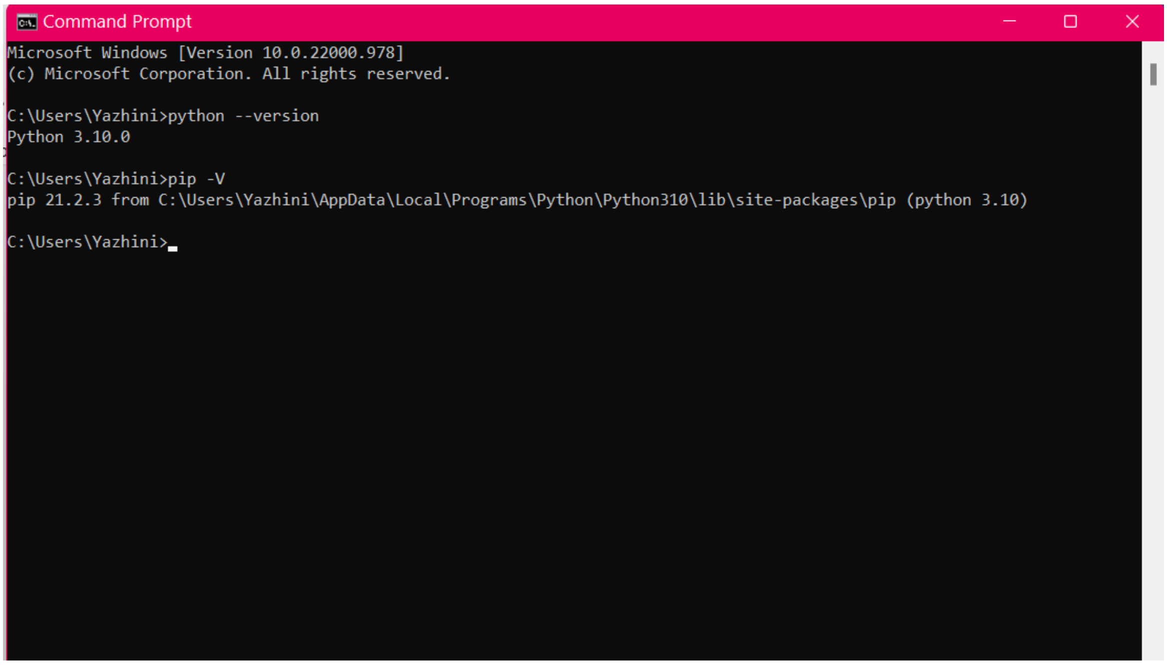
2. DISADVANTAGES

- Wrong inputs will affect the project outputs.
- Internet Connection is mandatory.
- Reports are not Verified

11. SETTING UP APPLICATION ENVIRONMENT

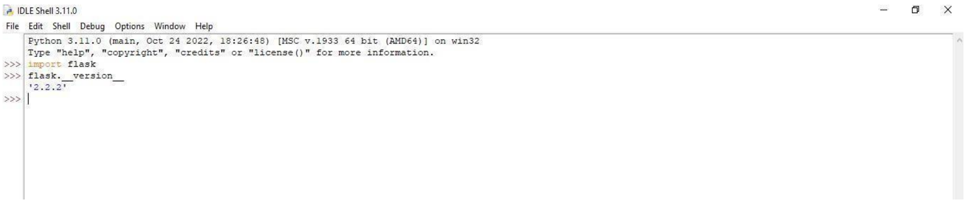
1. CREATE FLASK PROJECT

- To check the version : **python --version**
- To check the path : **pip -V**



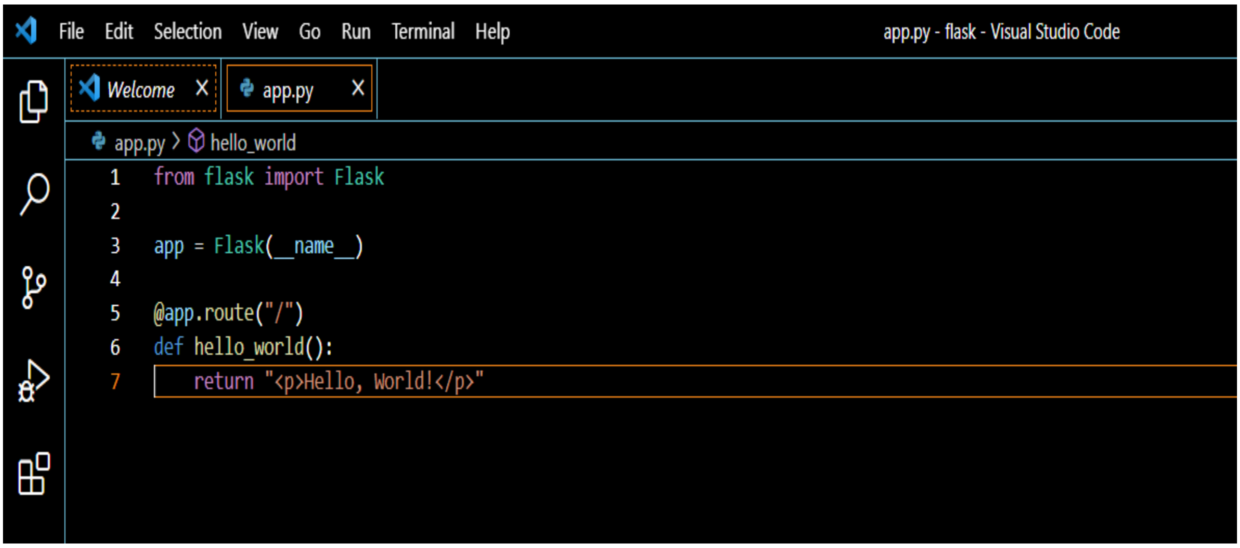
```
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Yazhini>python --version
Python 3.10.0

C:\Users\Yazhini>pip -V
pip 21.2.3 from C:\Users\Yazhini\AppData\Local\Programs\Python\Python310\lib\site-packages\pip (python 3.10)

C:\Users\Yazhini>
```

- Flask installation : **py -m install flask**



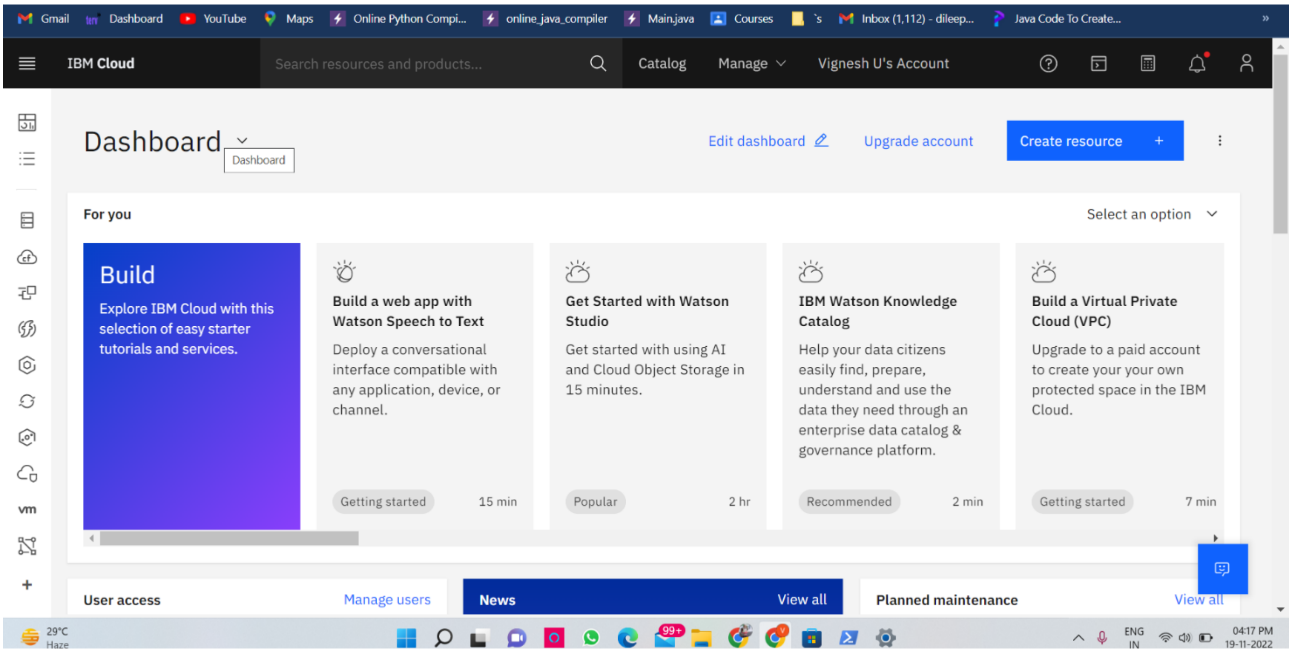
```
    Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
>>> import flask
>>> flask.__version__
    '2.2.2'
>>> 
```

- app.py file in VC code to run it.



```python
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"
```

2. CREATE IBM CLOUD ACCOUNT



Build

Explore IBM Cloud with this selection of easy starter tutorials and services.

Build a web app with Watson Speech to Text

Deploy a conversational interface compatible with any application, device, or channel.

Getting started 15 min

Get Started with Watson Studio

Get started with using AI and Cloud Object Storage in 15 minutes.

Popular 2 hr

IBM Watson Knowledge Catalog

Help your data citizens easily find, prepare, understand and use the data they need through an enterprise data catalog & governance platform.

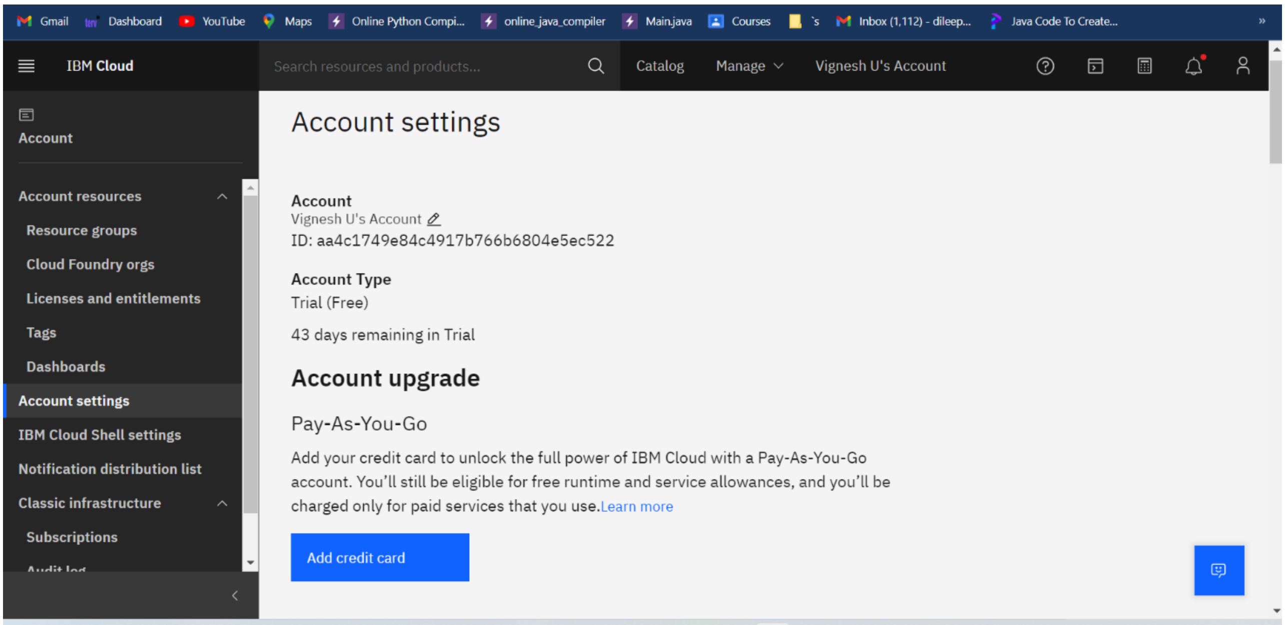
Recommended 2 min

Build a Virtual Private Cloud (VPC)

Upgrade to a paid account to create your your own protected space in the IBM Cloud.

Getting started 7 min

Account details



Account settings

Account
Vignesh U's Account
ID: aa4c1749e84c4917b766b6804e5ec522

Account Type
Trial (Free)

43 days remaining in Trial

Account upgrade

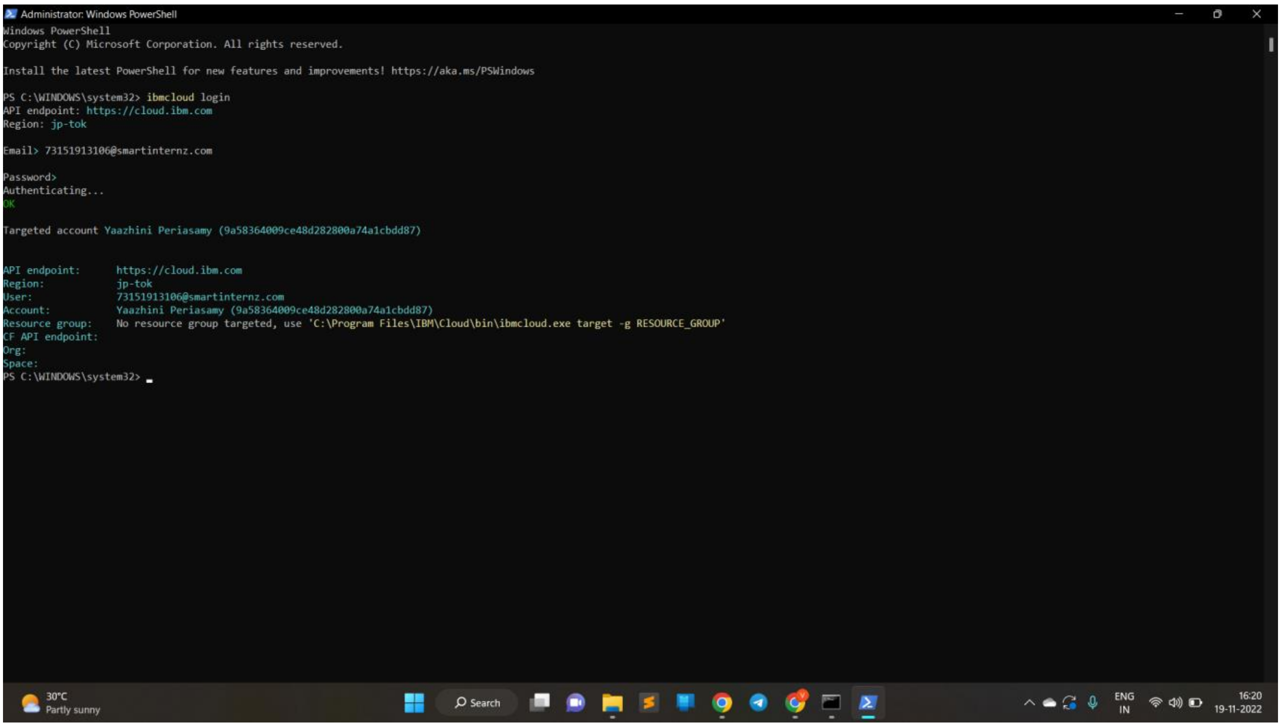
Pay-As-You-Go

Add your credit card to unlock the full power of IBM Cloud with a Pay-As-You-Go account. You'll still be eligible for free runtime and service allowances, and you'll be charged only for paid services that you use.Learn more

Add credit card

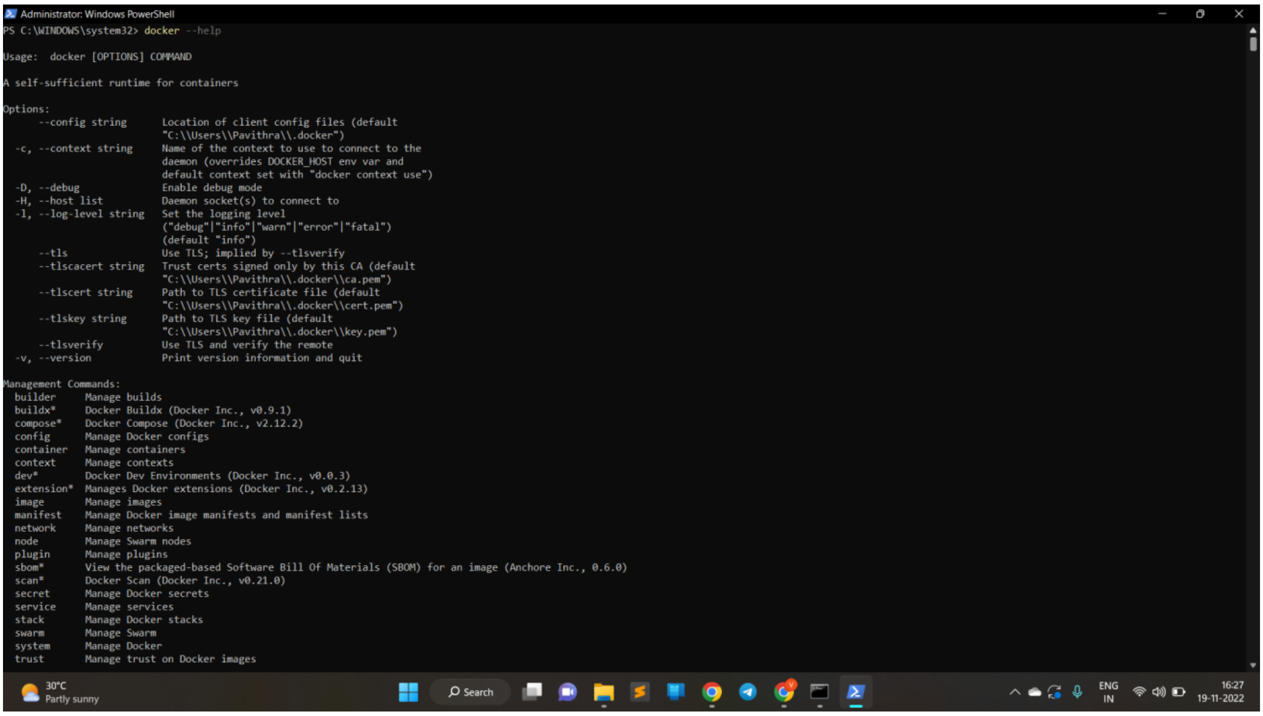
3. INSTALL IBM CLOUD CLI

Authentication of Cloud Account after the installation of Cloud CLI

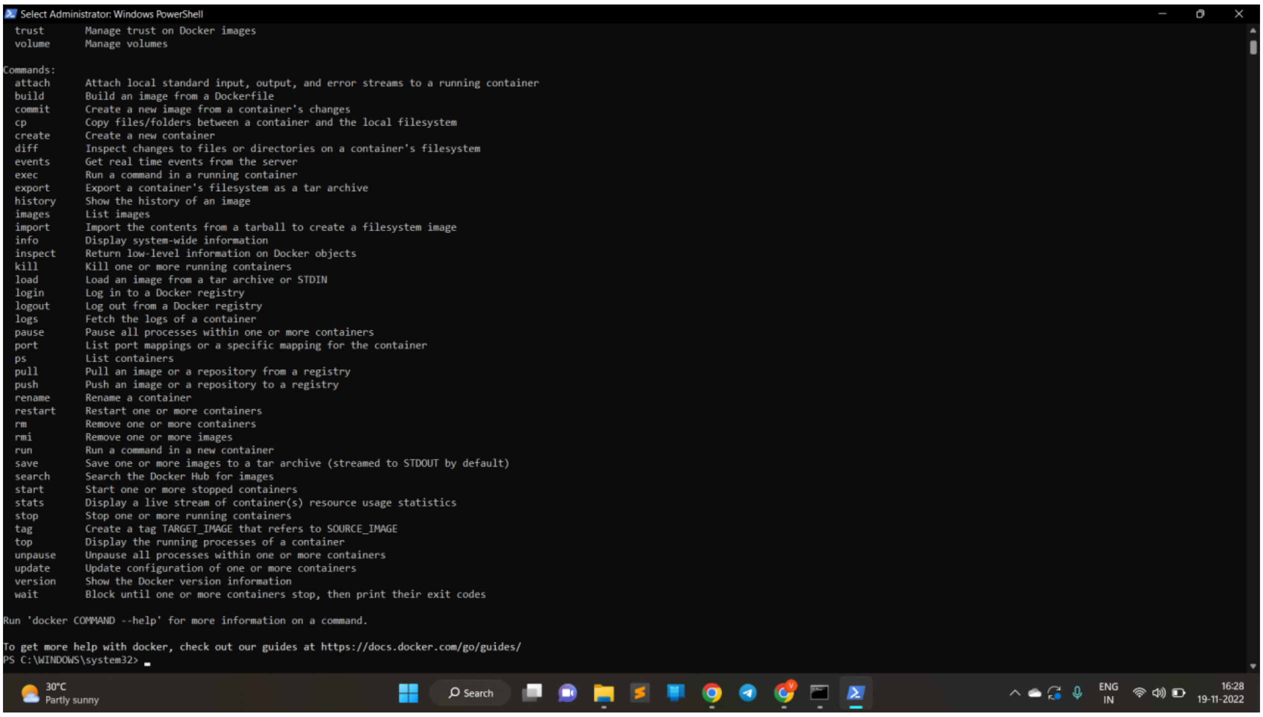


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> ibmcloud login
API endpoint: https://cloud.ibm.com
Region: jp-tok

Email> 73151913106@smartinternz.com

Password>
Authenticating...
OK

Targeted account Yaazhini Periasamy (9a58364009ce48d282800a74a1cbdd87)


API endpoint:      https://cloud.ibm.com
Region:            jp-tok
User:              73151913106@smartinternz.com
Account:           Yaazhini Periasamy (9a58364009ce48d282800a74a1cbdd87)
Resource group:    No resource group targeted, use 'C:\Program Files\IBM\Cloud\bin\ibmcloud.exe target -g RESOURCE_GROUP'
CF API endpoint:
Org:
Space:
PS C:\WINDOWS\system32> _
```

4. DOCKER CLI INSTALLATION



```
PS C:\WINDOWS\system32> docker --help

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
      --config string      Location of client config files (default
                           "C:\\Users\\Pavithra\\.docker")
  -c, --context string     Name of the context to use to connect to the
                           daemon (overrides DOCKER_HOST env var and
                           default context set with "docker context use")
  -D, --debug              Enable debug mode
  -H, --host list          Daemon socket(s) to connect to
  -l, --log-level string   Set the logging level
                           ("debug"|"info"|"warn"|"error"|"fatal")
                           (default "info")
      --tls                Use TLS; implied by --tlsverify
      --tlscacert string   Trust certs signed only by this CA (default
                           "C:\\Users\\Pavithra\\.docker\\ca.pem")
      --tlscert string     Path to TLS certificate file (default
                           "C:\\Users\\Pavithra\\.docker\\cert.pem")
      --tlskey string      Path to TLS key file (default
                           "C:\\Users\\Pavithra\\.docker\\key.pem")
      --tlsverify          Use TLS and verify the remote
  -v, --version            Print version information and quit

Management Commands:
  builder     Manage builds
  buildx*     Docker Buildx (Docker Inc., v0.9.1)
  compose*    Docker Compose (Docker Inc., v2.12.2)
  config      Manage Docker configs
  container   Manage containers
  context     Manage contexts
  dev*        Docker Dev Environments (Docker Inc., v0.0.3)
  extension*  Manages Docker extensions (Docker Inc., v0.2.13)
  image       Manage images
  manifest    Manage Docker image manifests and manifest lists
  network     Manage networks
  node        Manage Swarm nodes
  plugin      Manage plugins
  sbom*       View the packaged-based Software Bill Of Materials (SBOM) for an image (Anchore Inc., 0.6.0)
  scan*       Docker Scan (Docker Inc., v0.21.0)
  secret      Manage Docker secrets
  service     Manage services
  stack       Manage Docker stacks
  swarm       Manage Swarm
  system      Manage Docker
  trust       Manage trust on Docker images
```



```
  trust       Manage trust on Docker images
  volume      Manage volumes

Commands:
  attach      Attach local standard input, output, and error streams to a running container
  build       Build an image from a Dockerfile
  commit      Create a new image from a container's changes
  cp          Copy files/folders between a container and the local filesystem
  create      Create a new container
  diff        Inspect changes to files or directories on a container's filesystem
  events      Get real time events from the server
  exec        Run a command in a running container
  export      Export a container's filesystem as a tar archive
  history     Show the history of an image
  images      List images
  import      Import the contents from a tarball to create a filesystem image
  info        Display system-wide information
  inspect     Return low-level information on Docker objects
  kill        Kill one or more running containers
  load        Load an image from a tar archive or STDIN
  login       Log in to a Docker registry
  logout      Log out from a Docker registry
  logs        Fetch the logs of a container
  pause       Pause all processes within one or more containers
  port        List port mappings or a specific mapping for the container
  ps          List containers
  pull        Pull an image or a repository from a registry
  push        Push an image or a repository to a registry
  rename      Rename a container
  restart     Restart one or more containers
  rm          Remove one or more containers
  rmi         Remove one or more images
  run         Run a command in a new container
  save        Save one or more images to a tar archive (streamed to STDOUT by default)
  search      Search the Docker Hub for images
  start       Start one or more stopped containers
  stats       Display a live stream of container(s) resource usage statistics
  stop        Stop one or more running containers
  tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
  top         Display the running processes of a container
  unpause     Unpause all processes within one or more containers
  update      Update configuration of one or more containers
  version     Show the Docker version information
  wait        Block until one or more containers stop, then print their exit codes

Run 'docker COMMAND --help' for more information on a command.

To get more help with docker, check out our guides at https://docs.docker.com/go/guides/
PS C:\WINDOWS\system32> _
```

12. CONCLUSION

This project proved good for me as it provided practical knowledge of not only programming in ASP.NET and VB.NET web based application and no some extent windows Application and SQL Server, but also about all handling procedure related with "Plasma Donor Application". It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

13. FUTURE SCOPE

Plasma Donor Application is a web application to build such a way that it should suits for all type of blood banks in future. One important future scope is availability of location-based blood bank details and extraction of location-based donor’s detail, which is very helpful to the acceptant people. All the time the network facilities cannot be use. This time donor request does not reach in proper time, this can be avoided through adding some message sending procedure this will help to find proper blood donor in time. This will provide availability of blood in time.

APPENDIX:

SOURCE CODE

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import json
app = Flask(__name__)
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=fbd88901-ebdb-4a4f-
a32e9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32731;S
ECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=hjq91894;PW
D=y0CHaaerS4x2BfmR",",")
@app.route('/registration')
def home():
    return render_template('register.html')
@app.route('/register',methods=['POST'])
def register():
    x = [x for x in request.form.values()]
    print(x)
    name=x[0]
    email=x[1]
    phone=x[2]
    city=x[3]
    infect=x[4]
    blood=x[5]
    password=x[6]
    sql = "SELECT * FROM user 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)
    print(account)
    if account:
        return render_template('register.html', pred="You are already a member, please
login using your details")
    else:
        insert_sql = "INSERT INTO  user VALUES (?, ?, ?, ?, ?, ?, ?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1, name)
        ibm_db.bind_param(prep_stmt, 2, email)
        ibm_db.bind_param(prep_stmt, 3, phone)
        ibm_db.bind_param(prep_stmt, 4, city)
        ibm_db.bind_param(prep_stmt, 5, infect)
        ibm_db.bind_param(prep_stmt, 6, blood)
        ibm_db.bind_param(prep_stmt, 7, password)
        ibm_db.execute(prep_stmt)
        return render_template('register.html', pred="Registration Successful, please login
using your details")
@app.route('/')
@app.route('/login')
def login():
    return render_template('login.html')
@app.route('/loginpage',methods=['POST'])
def loginpage():
    user = request.form['user']
    passw = request.form['passw']
    sql = "SELECT * FROM user WHERE email =? AND password=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,user)
    ibm_db.bind_param(stmt,2,passw)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print (account)
    print(user,passw)
    if account:
            return redirect(url_for('stats'))
    else:
        return render_template('login.html', pred="Login unsuccessful. Incorrect username / password !")
@app.route('/stats')
def stats():
    '''sql = "SELECT blood FROM user group by blood"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.execute(stmt)
    count = ibm_db.fetch_assoc(stmt)
    print(count)'''
    return render_template('stats.html',b=5,b1=2,b2=3,b3=4,b4=2,b5=1,b6=2,b7=1,b8=1)
@app.route('/requester')
def requester():
    return render_template('request.html')
@app.route('/requested',methods=['POST'])
def requested():
    bloodgrp = request.form['bloodgrp']
    address = request.form['address']
    print(address)
    sql = "SELECT * FROM user WHERE blood=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,bloodgrp)
    ibm_db.execute(stmt)
    data = ibm_db.fetch_assoc(stmt)
    msg = "Need Plasma of your blood group for: "+address
    while data != False:
        print ("The Phone is : ", data["PHONE"])
```

```
url="https://www.fast2sms.com/dev/bulk?authorization=xCXuwWTzyjOD2ARd1EngbH3a7tKIq5PklJ8YSf0Lh4FQZecs9iNI1dSvuqprxFwCKYJXA5amQkBE36Rl&sender_id=FSTSMS&message="+msg+"&language=english&route=p&numbers="+str(data["PHONE"])
        result=requests.request("GET",url)
        print(result)
        data = ibm_db.fetch_assoc(stmt)
    return render_template('request.html', pred="Your request is sent to the concerned people.")
if __name__== "__main__":
    app.run(host='0.0.0.0', port=8080)
```

For accesing SendGrid mail access,

```
import os

from dotenv import load_dotenv

load_dotenv()

from sendgrid import SendGridAPIClient

from sendgrid.helpers.mail import Mail

def sendmail(usermail,subject,name,content):

message = Mail(from_email='73151913106@smartinternz.com',to_emails=usermail,subject='Plasma donor-reg.',html_content='<h4>Hello {}, </h4><br/><strong> {} </strong><br/><p>Best Wishes and Welcome to donate plasma. We are accepting the proposal. Thank you!,</p><p>Team Plasma</p>'.format(name,content))

    try:

        sg = SendGridAPIClient(os.getenv('API_KEY'))
```

```
response = sg.send(message)
print(response.status_code)
print(response.body)
print(response.headers)
except Exception as e:
print(e.message)
```

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-41916-1660646127

DEMO VIDEO LINK:

https://drive.google.com/file/d/1WeWCRn7vCJH0xCXMqGnHgTBq2e4fDrTH/view?usp=sharing