PROJECT REPORT FOR PLASMA DONOR APPLICATION

TEAM ID: PNT2022TMID42910

BATCH ID: B11- 5A1E

MENTOR: Brindha.K

COLLEGE: RVS College of Engineering & Technology,Coimbatore

Team Members

Eswara Moorthy.M	**712819104011**
Harivignesh.K	**712819104012**
Karolin Susmitha.A	**712819104015**
Boopathy Raja.N	**712819104719**

1.INTRODUCTION

1.1 PROJECT OVERVIEW

When the world is struck by deadly diseases, there is a high risk of mass death of populations across the world. These diseases give no enough time for the surgeons to find medicine and so there is a need to find a quick remedy to reduce mass death of people to such illness. One of the best methods, which is highly effective is the donation of blood plasma of cured individuals to sick persons. This can possibly cure the illness of the infected person. Plasma donation was one of the best methods which was adopted to cure people during the recent global pandemic, COVID-19. The recovery rates were high during these times when death was ultimately increasing as no medicine was found across the globe. Another issue was that no cured patient came forward to donate blood plasma, so the infected ones were highly worried as they can't find anyone to help them. So, we are in need of an application that stores donor details, tracks and informs them upon request from a patient.

1.2 PURPOSE

Plasma donation, also known as apheresis, can help save lives. It is a relatively safe procedure, but there can be minor side effects. Plasma is the liquid part of the blood. It contains proteins and antibodies that are crucial for clotting and immunity. Around 55% of the blood is plasma. Plasma donation involves drawing blood, extracting the plasma, and returning what is left of the blood to the person, all through a single needle that remains in the arm throughout the process. Plasma is in high demand, as it helps treat cancer and other health issues. In May 2020, the Food and Drug Administration (FDA) asked people who had recovered from COVID19 to donate plasma. Experts believe that the plasma may contain antibodies for SARS-CoV2, the virus behind the disease. Receiving plasma with these antibodies could help a person fight off the infection. People with AB blood have a universal type of plasma, which means that a person with any blood type can receive

this plasma safely. This is different from having the universal blood type, which is O negative. The American Red Cross urge people with AB blood to donate plasma. A person can do this every 28 days, or up to 13 times a year. Research shows that plasma donation is safe, and the National Institutes of Health (NIH) emphasize that there is no risk of getting the wrong blood back. Also, the FDA and other health authorities regulate the equipment and procedure of plasma donation. However, a person who donates plasma may experience minor adverse effects, and as with any other procedure involving a puncture, certain risks are involved. So, it is highly necessary and equally important to create an application to maintain the donors list and details to contact and track them during emergency situations. When talking about an application, it needs to be easy to handle and user friendly. Software like Python-Flask, and Docker are used in this application.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

There is a recent transformation into the development of multi-platform languages and frameworks. Flask is a small framework by most standards, small enough to be called a “microframework.” It is small enough that once you become familiar with it, you will likely be able to read and understand all of its source code. But being small does not mean that it does less than other frameworks. Flask was designed as an extensible framework from the ground up; it provides a solid core with the basic services, while extensions provide the rest. Because you can pick and choose the extension packages that you want, you end up with a lean stack that has no bloat and does exactly what you need. Flask has two main dependencies. The routing, debugging, and Web Server Gateway Interface (WSGI) subsystems come from Werkzeug, while template support is provided by Jinja2. Werkzeug and Jinja2 are authored by the core developer of Flask. There is no native support in Flask for accessing

databases, validating web forms, authenticating users, or other high-level tasks. These and many other key services most web applications need are available through extensions that integrate with the core packages. As a developer, you have the power to cherry-pick the extensions that work best for your project or even write your own if you feel inclined to. This is in contrast with a larger framework, where most choices have been made for you and are hard or sometimes impossible to change.

2.2 REFERENCE

1) Philip J, Sarkar RS, Pathak A. Adverse events associated with apheresis procedures: Incidence and relative frequency. Asian J Transfus Sci. 2013 Jan;7(1):37-41. doi: 10.4103/0973-6247.106730. PMID: 23559763; PMCID: PMC3613659.

2) Flask Web Development by Miguel Grinberg Copyright © 2014 Miguel Grinberg. All rights reserved. Printed in the United States of America. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

3) Version v1.2.0 (fba92ef) of The Docker Book by James Turnbul © Copyright 2014 - James Turnbull.

4) An Introduction to Kubernetes by LEVEREGE , First Edition © Leverege LLC.

5) White Paper by IDC Sponsored by: IBM Andrew Smith, February 2021.

6) Essentials of Application Development on IBM Cloud December 2017 Third Edition (December 2017) by Ahmed Azraq Hala A. Aziz Uzma Siddiqui.

7) From Ahmedabad, Smriti Thakkar is the first recovered Covid-19 patient in India who volunteered to donate her plasma. India Today Web Desk New Delhi, April 25, 2020 UPDATED: April 25, 2020 19:15 IST

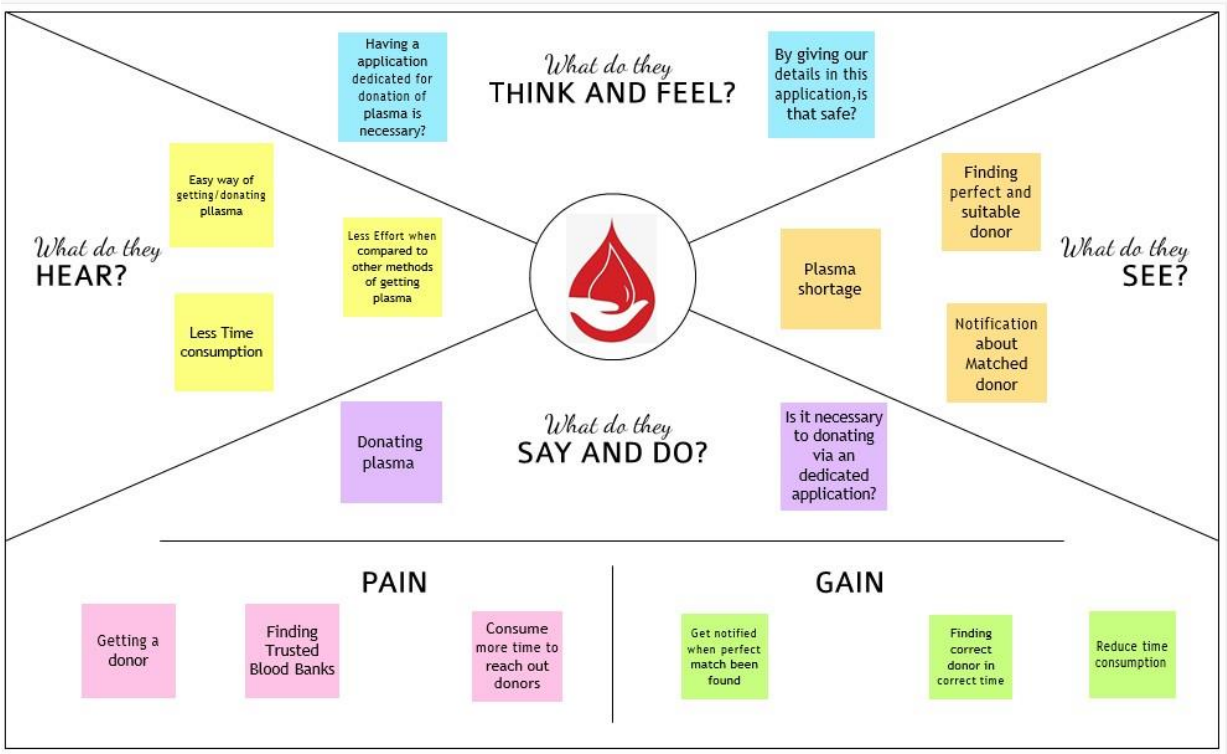
8) Tablighi Jamaat gives blood for plasma therapy By Zubair Ahmed BBC Hindi, Delhi Published 28 April 2020.

2.3 PROBLEM STATEMENT DEFINITION

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	a patient who is in need of plasma for treatment.	get and contact a plasma donor who willing to give.	I can't able to find a suitable plasma donor to meet my blood type criterion.	of having rare blood type, donor's not many available on my surroundings.	worry and depressed about my health.
PS-2	a father.	get plasma for my son who is a 12year kid.	I am not able to find a suitable donor.	of having long distance from our place, they could not able to make here.	worry about my kid's health.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTORM

APP INTERFACE

Chatbot service can be included to clear doubts about plasma donation.	Using hospital/ clinic organization to improve the service.
If the blood group is suitable with the requested blood type,then only the notification are send.	Verifications are to be made at registration stage inorder to make the donation.

FEATURES

Features like history of the donations made,finding donor's location using GPS	Rapid contacting features can be added when there is a immediate need of plasma.
By using location detecting features,one can able to find accurate location of the donor.	Organizing various activities to promote the application's interest among the people.

REGISTRATION STAGES

Details such as gender , D.O.B , age , contact details are collected and stored in database.	Blood group description from both the parties(Donor & collector) is collected.
Only donors from the age of 18 having a weight of 50kg can register in the application.	The user during the initial stage of registration should given whether it is his/her first time at donating or already donated person.

SAFETY MEASURES

Plasma of the body are to be examined by a medical expert before the donation.	To ensure and verify whether the donor is free from any other cautionary diseases.
The exact date of the plasma extraction must be mentioned on his profile.	If there is any misinformation and proved that he/she had any other diseases, immediate action of removing his/her profile are to made

ESWARA MOORTHY M

Using hospital/clinic organization to improve the service.	Chatbot service can be included to clear doubts about plasma donation.
If the blood group is suitable with the requested blood type,then only the notification are send.	Organizing various activities to promote the application's interest among the people.

HARIVIGNESH K

Features like history of the donations made,finding donor's location using GPS	Only donors from the age of 18 having a weight of 50kg can register in the application.
Rapid contacting features can be added when there is a immediate need of plasma.	By using location detecting features,one can able to find accurate location of the donor.

KAROLIN SUSMITHA A

Details such as gender , D.O.B , age , contact details are collected and stored in database.	Plasma of the body are to be examined by a medical expert before the donation.
Verifications are to be made at registration stage inorder to make the donation.	To ensure and verify whether the donor is free from any other cautionary diseases.

BOOPATHY RAJA N

Blood group description from both the parties(Donor & collector) is collected.	If there is any misinformation and proved that he/she had any other diseases, immediate action of removing his/her profile are to made
The exact date of the plasma extraction must be mentioned on his profile.	The user during the initial stage of registration should given whether it is his/her first time at donating or already donated person.

3.3 PROPOSED SOLUTION

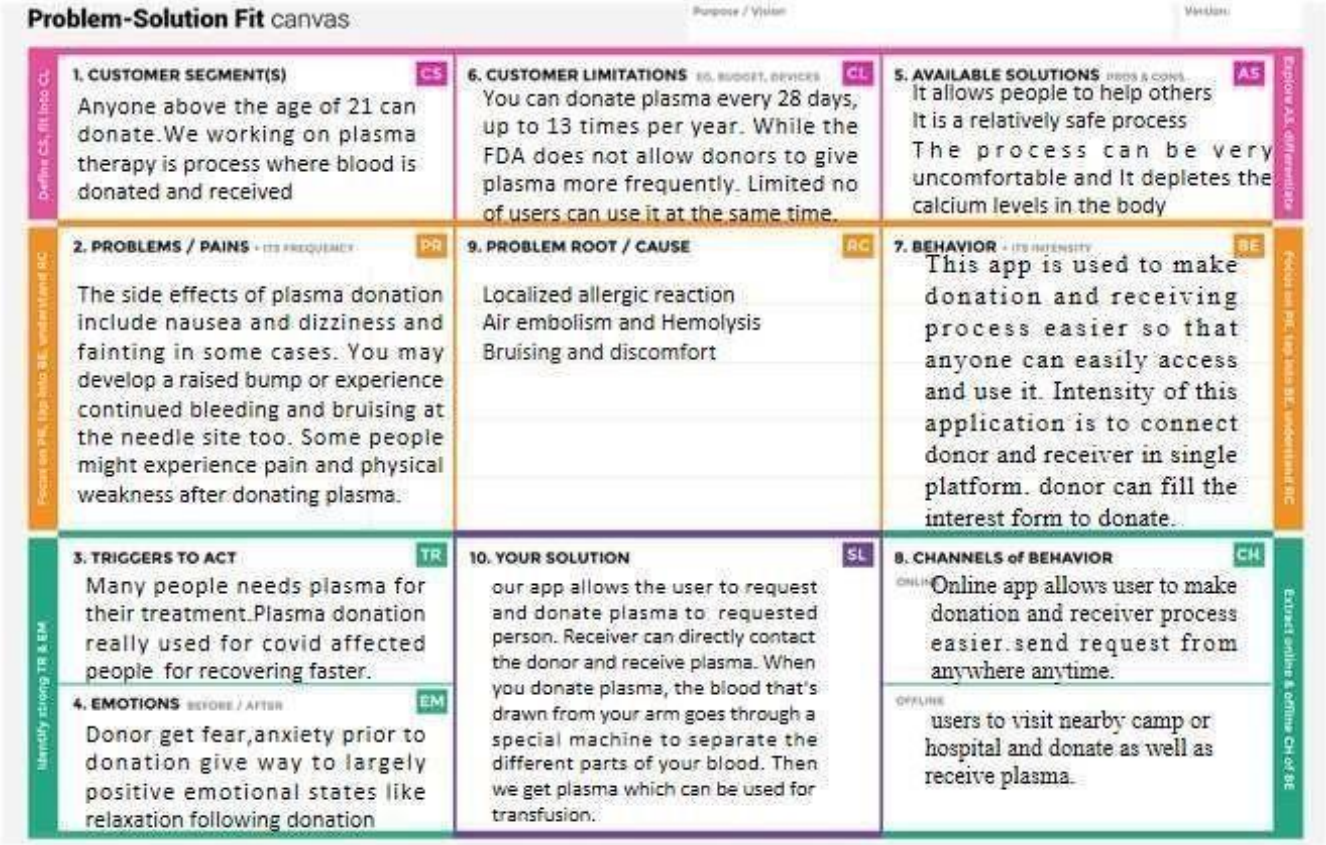
S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Plasma is used for the treatment of many serious health problems. This is why there are blood drives asking people to donate blood, plasma . Plasma is utilized to treat different irresistible sicknesses and it isone of the most established strategies known as plasma treatment. During Coronavirus emergency the necessity for plasma expanded radically as there were no immunization found to treat the contaminated 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 contributor data and telling about the ongoing givers would be some assistance as it can save time and assist the clients with finding the vital data about the contributors.
2.	Idea / Solution description	This proposed system aims at connecting the donors & the patients by an online application. By using this application, the users can either raise a request for plasma donation or requirements. The basic solution is to create a centralized system to keep a track on the upcoming as well as past Plasma Donation Events. The recommendation solution is as follows: Application contains two roles: ■ Admin ■ User

		User: • If the user wants to donate or receive they have to register with their personal details. • After successful registration of user. • A successful registration email is send to the user. • After successful registration user will be directed to home page. • They will be asked to press whether they will be donor or receiver. • If the user is donor then he/she will fill the donation interest form which includes their Name, blood group details, location, last time donated date , phone number, email id. • After filling the donation form he/she will redirected to page in which he/she can download the e-certificate. • If the user is receiver then he/she can see the list of donors available and they can raise their request and contact donor directly. Admin: • Admin can login using their credentials. • Admin can edit the request. • Admin can delete the request. • Admin can add volunteers.

3.	Novelty / Uniqueness	A User Interface is simple for users to understand. We can use the application anywhere anytime. The user immediately need the plasma for their treatment but the plasma is not available in nearby hospitals, then user can use this application to raise request and directly contact the donor , request them to donate the plasma. Hospitals can also raise request donors for donation. Somebody wants to donate blood and plasma butthey don't know the way to donate then they use this application which will simple to use and it will save lives of many people. Today many of them have mobile phones can install this and save lives.
4.	Social Impact / Customer Satisfaction	We are living in a modern world and everything can be accessed online. Even though there are many application there is no proper application for plasma donation . Many of them wish to donate blood and plasma but they are unaware about donation and how they can donate. This application provides opportunity to those who want to donate plasma. Donation of plasma are happening in many places many of them come forward to donate but it is notavailable at right time for use. Sometimes there is a shortage of plasma of particular type. Additional facilities that we need is to access the patients information quickly before plasma transfusion. To solve this issue software applications are employed with Cloud computing and Internet of Things tool which enable features such as information retrieval and continuous data tracking with analytics. This application avoids circulating of wrong information. A single platform for maintaining genuine information andincrease the trust of participants involved int his activity. It increases the number of donors.

5.	Business Model (Revenue Model)	This application is accessible by everyone . It is free.Because of the trouble in finding givers who match a specific blood bunch, this application empowers clients to enlist individuals who wish to give plasma and keep their data in a data set. Nowadays the need for plasma increases. Anyone with basic knowledge can access this app. This can be used anywhere anytime. working with the government we can utilize an application to help those needing plasma.
6.	Scalability of the Solution	This application helps users to find plasma donors by sitting in home itself instead of searching donors everywhere. When there is a emergency then plasma request to send to everyone. Once the donor is ready to donate receiver is notified about donation. Receiver can contact the donor. With this app donor can know the eligibility to donate and making it easier to locate suitable donor at right time.

3.4 PROBLEM SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

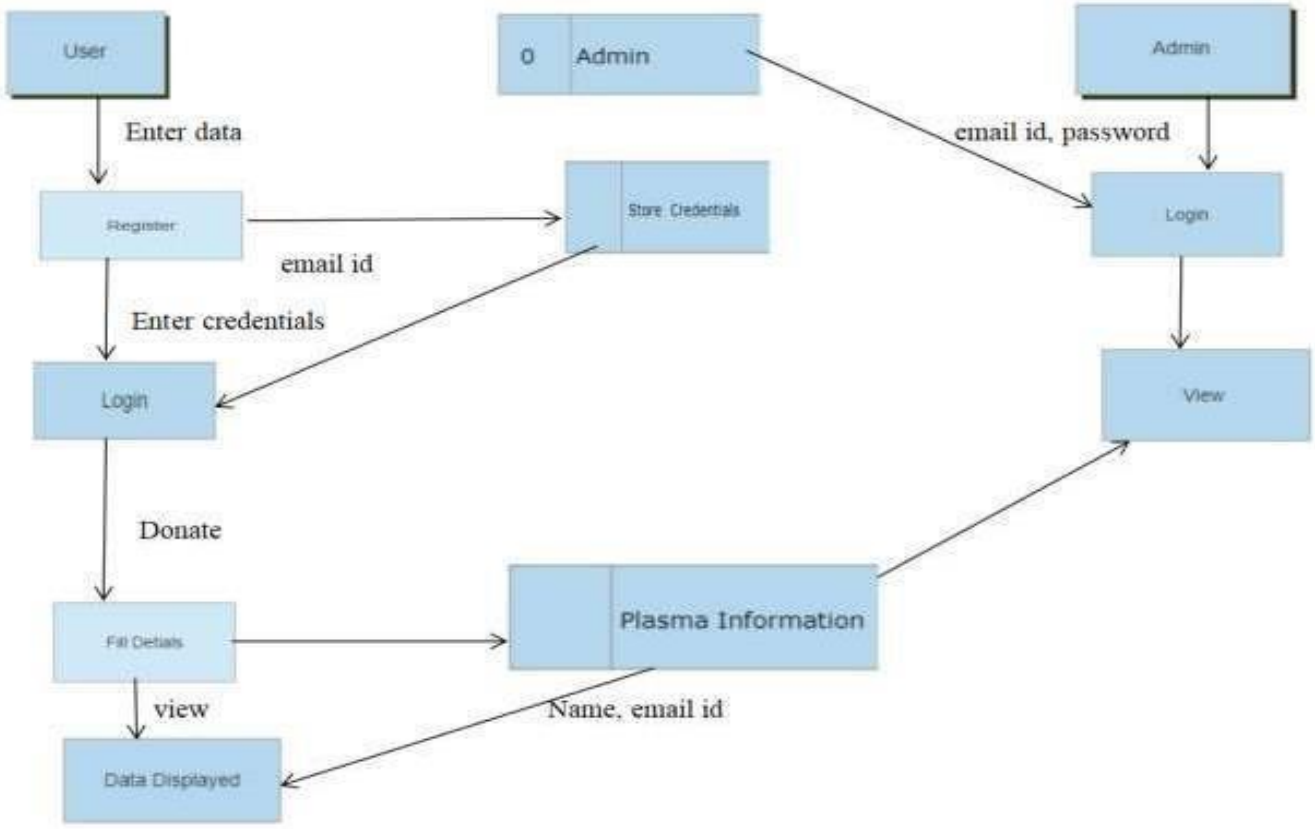
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Website
FR-2	User Confirmation	Confirmation via Email
FR-3	User Login	Login using Registered email Id
FR-4	Sent Request	If plasma is required, the receiver will contact the donor
FR-5	Contact Donor	Contact the donor directly if a phone number is given
FR-6	View donation camps	View the list of donation camps happening nearby

4.2 NON - FUNCTIONAL REQUIREMENTS

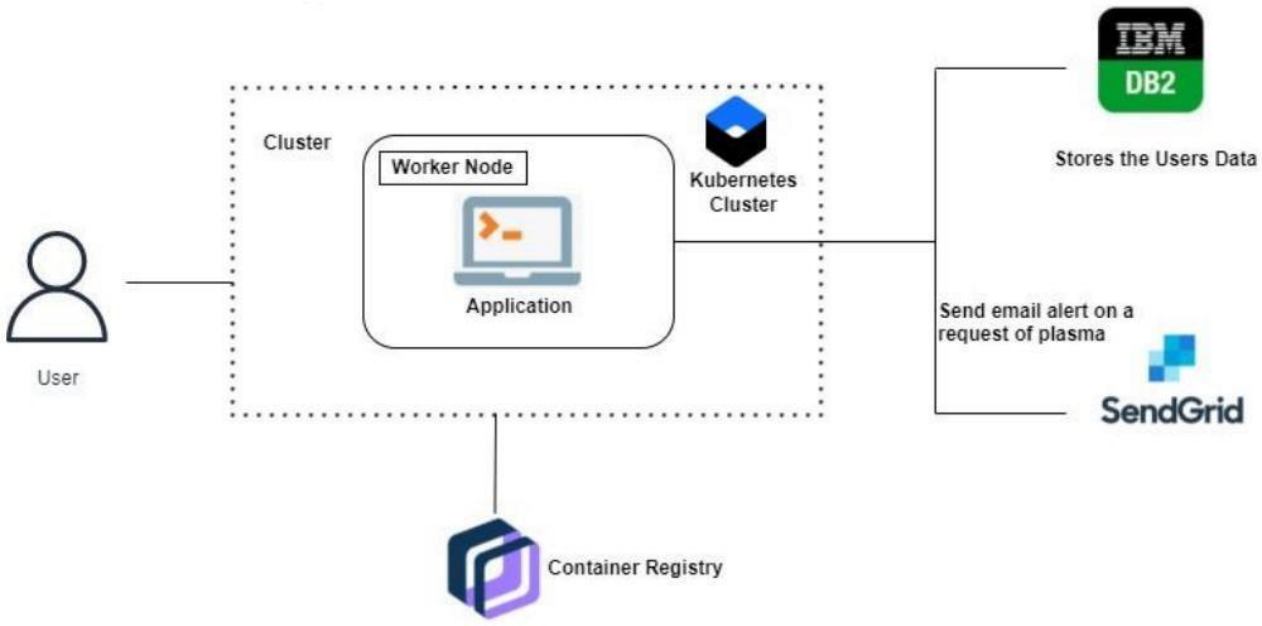
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The user interface of the plasma donor system must be well-designed and welcoming.
NFR-2	**Security**	Data storage is required by security systems, just like it is by many other applications. Databases are able to keep all the donor information that is viewed by applications. It must be secured with email Id and password.
NFR-3	**Reliability**	The system has the ability to work all the times without failures apart from network failure. A donor can have the faith on the system. The authorities will keeps the privacy of all donors in a proper manner
NFR-4	**Performance**	The Plasma donor System must perform well in different scenarios. The system is interactive and delays involved are less.
NFR-5	**Availability**	The system, including the online components,should be available 24/7.
NFR-6	**Scalability**	The system offers the proper resources for issue solutions and is designed to protect sensitive information during all phases of operation.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION AND TECHNICAL ARCHITECTURE



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application byentering my email, password.	I can access my account dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation emailonce I have registered for the application	I can receive successful message	High	Sprint-1
	Login	USN-3	As a user, I can log into the application byentering email & password	I can access into myProfile and view my dashboard	High	Sprint-1
	Dashboard	USN-4	As a user, I can login using my credentialsand it will direct it to my dashboard	I can view and access what are the features are providedin dashboard	High	Sprint-1
Customer (Webuser)		USN-5	As a user, I can login using my credentials and it will direct it to my dashboard	I can view and access what are the features are providedin dashboard	High	Sprint -1
Customer Care Executive	Query	USN-6	As a user had an any query about the givenrequirements	I can view a query and rectify the given query	medium	Sprint-2
Administrator	Login	USN-7	As a admin ,have credentials using that theycan login	They can view and modifythe data in database	medium	Sprint-2
	View	USN-8	As a admin I can view plasma information	View and modify	High	Sprint-1
	Modify	USN-9	As a admin I can modify the plasma information.	Modify only if there is a false information/	Medium	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story points	Priority	Team Members
Sprint 1	User Registration	USN-1	As a user, I can registe for the application b entering my email password, confirming my password and phone number..	10	High	Eswara Moorthy M, Harivignesh K, Karolin Susmitha A, Boopathy Raja N
Sprint 1	User Login	USN-2	As a user, I can log into the application byentering username & password.	10	High	Eswara Moorthy M, Harivignesh K, Karolin Susmitha A, Boopathy Raja N
Sprint 1	Access Websit e	USN-3	User should be able toaccess application using browser	10	High	Eswara Moorthy M, Harivignesh K, Karolin Susmitha A, Boopathy Raja N
Sprint 2	Dashboard	USN-4	The user upon logging in views the application dashboard where he/she can use all the application's services.	10	High	Eswara Moorthy M, Harivignesh K, Karolin Susmitha A, Boopathy Raja N
Sprint 2	Request For Blood plasma	USN-5	The user who is in nee of blood plasma ca request for blood b specifying the blood type.	2 0	High	Eswara Moorthy M, Harivignesh K, Karolin Susmitha A, Boopathy Raja N

Sprint 2	Switch User Roles	USN-6	As a user, he/she can switch roles between Donor and Receiver.	20	High	Eswara Moorthy M, Harivignesh K, Karolin Susmitha A, Boopathy Raja N
Sprint 3	View Plasma Request	USN-7	A donor receives an Email of about the receiver's details of the same blood type.	20	High	Eswara Moorthy M, Harivignesh K, Karolin Susmitha A, Boopathy Raja N
Sprint 3	View Donor Details	USN-8	The receiver can view the list of Donors of the blood type requested.	10	High	Eswara Moorthy M, Harivignesh K, Karolin Susmitha A, Boopathy Raja N
Sprint 4	Logout Process	USN-9	The User will be able to Logout of the application.	10	High	Eswara Moorthy M, Harivignesh K, Karolin Susmitha A, Boopathy Raja N
Sprint 4	Bot service in the website	USN-10	The user can use Bot Service to request for Blood Plasma and also switch between roles.	10	High	Eswara Moorthy M, Harivignesh K, Karolin Susmitha A, Boopathy Raja N

Sprint 3	Verified Donor	USN-7	As a donor, I can reques for verified account by providing the required documents and details to the admin through the web application.	8	Medium	Amarnath , Sukumar
Sprint 4	Update theprofile	USN-8	As a donor, I can update my profile any time.	8	Medium	Amarnath , Sukumar
Sprint 4	Feedback	USN-9	As a user, I can give the feedback to the Donor.	7	Low	Sowmiya , Amarnath

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint 1	30	8 days	22-10-2022	29-10-2022	30	29-10-2022
Sprint 2	50	8 days	29-11-2022	05-11-2022	50	05-11-2022
Sprint 3	30	8 days	05-11-2022	12-11-2022	30	12-11-2022
Sprint 4	20	8 days	12-11-2022	19-11-2022	20	19-11-2022

7.CODING & SOLUTIONING

7.1 FEATURE 1

https://github.com/IBM-EPBL/IBM-Project-46015-1660734713/tree/main/Project%20Development%20Phase/Sprint%201

It consists of two modules index and story

- Index- It is the main webpage of our model
- Story- It shows about the need for plasma donation

7.2 FEATURE 2

https://github.com/IBM-EPBL/IBM-Project-46015-1660734713/tree/main/Project%20Development%20Phase/Sprint%202

Here we discussed about register module,

- In this module, users can register their name as a donor.
- If a certain age limit is satisfied their registration process for plasma donors will be accepted.

7.3 FEATURE 3

https://github.com/IBM-EPBL/IBM-Project-46015-1660734713/tree/main/Project%20Development%20Phase/Sprint%203

Here we discussed about login module,

- In this module, users can login as a donor and they can update their availability status.
- Chatbot also created which helps the user to know more about plasma donation.

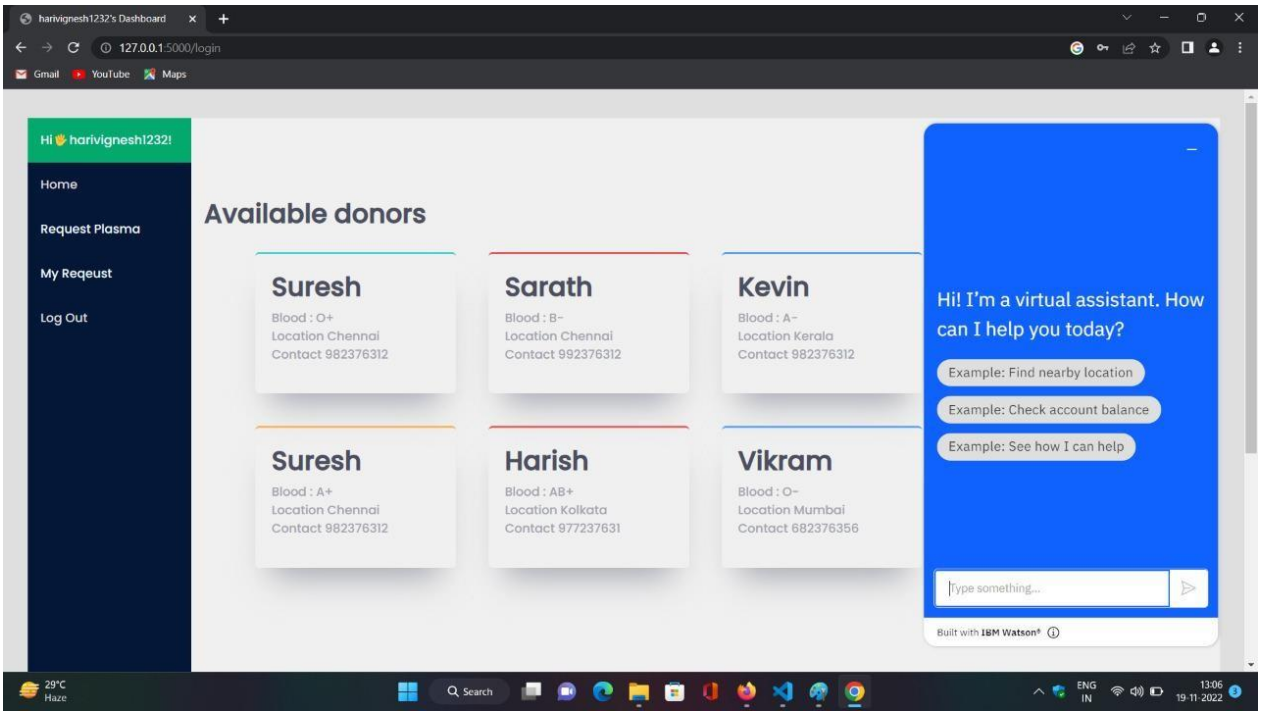
7.4 FEATURE 4

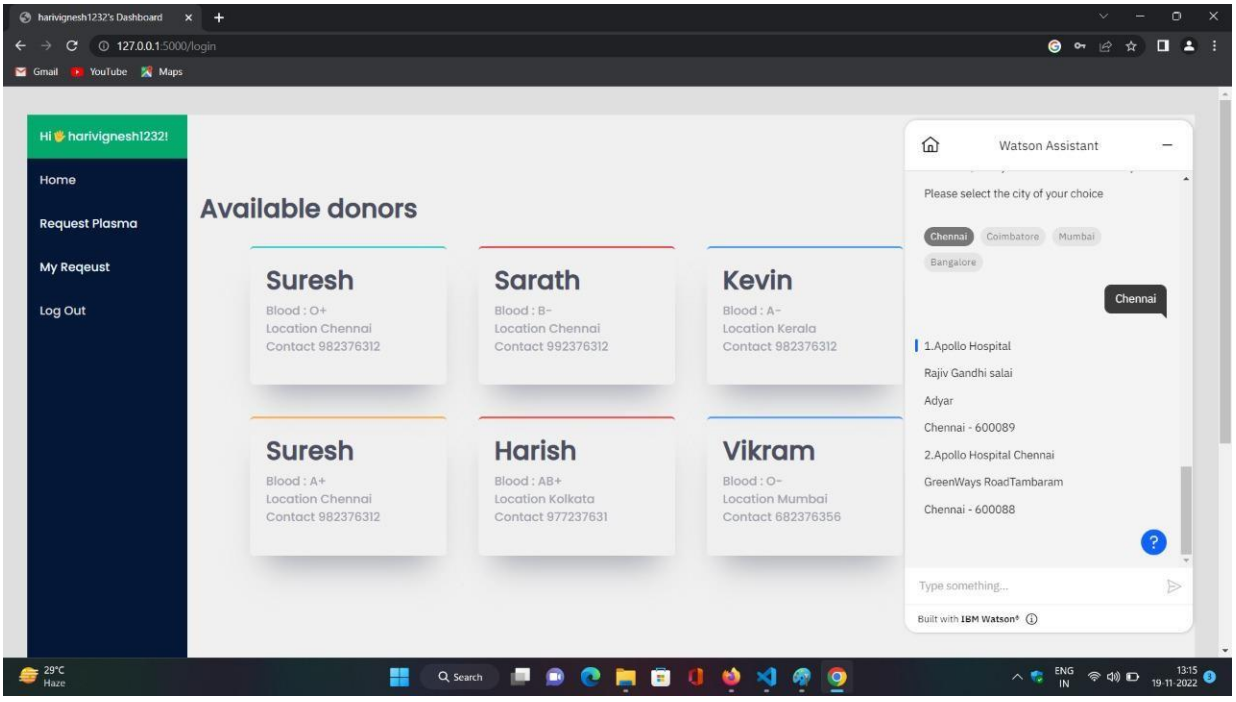
https://github.com/IBM-EPBL/IBM-Project-46015-1660734713/tree/main/Project%20Development%20Phase/Sprint%204

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

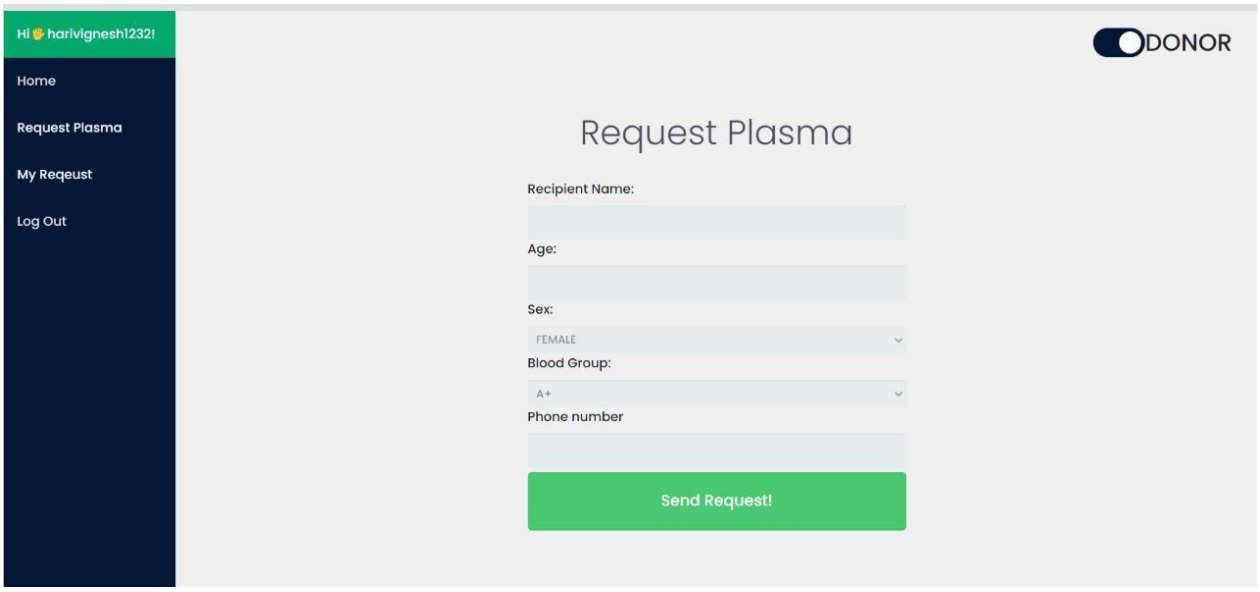
8. TESTCASES

8.1 CHATBOT WORKING

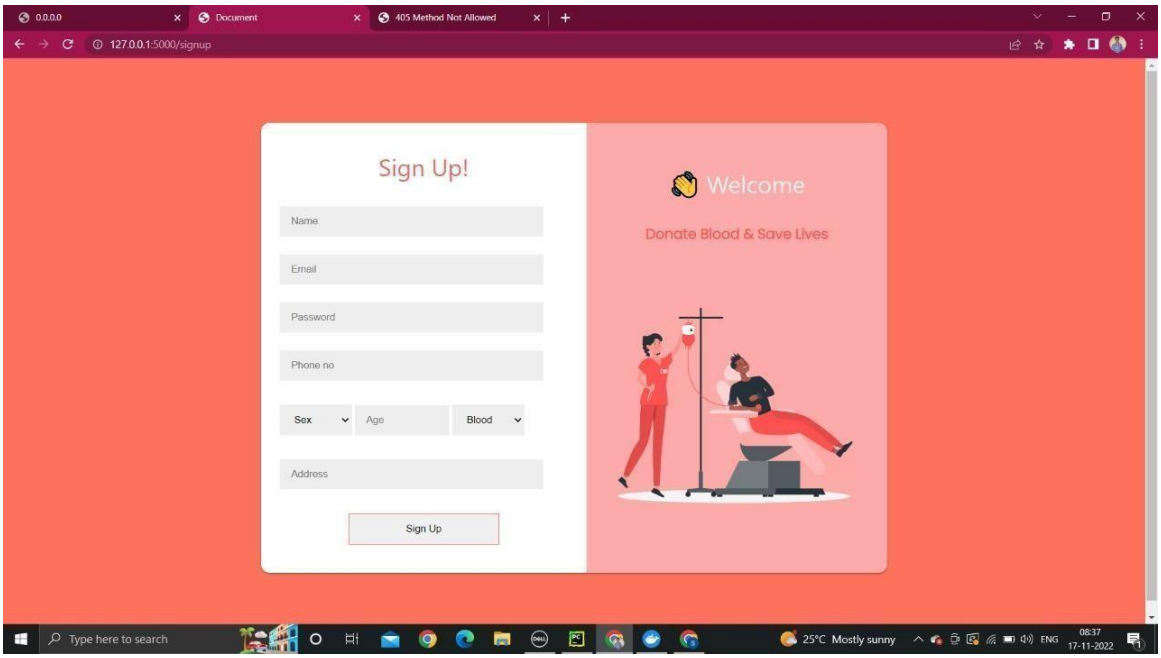




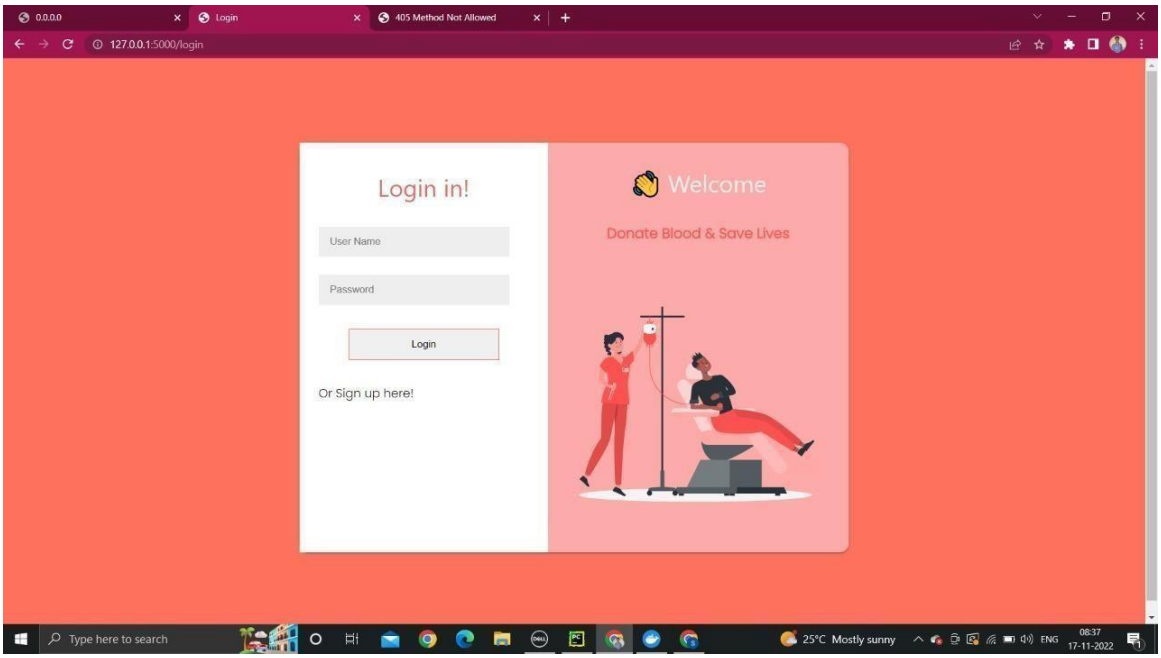
8.2 RECEPEINT SEARCHING



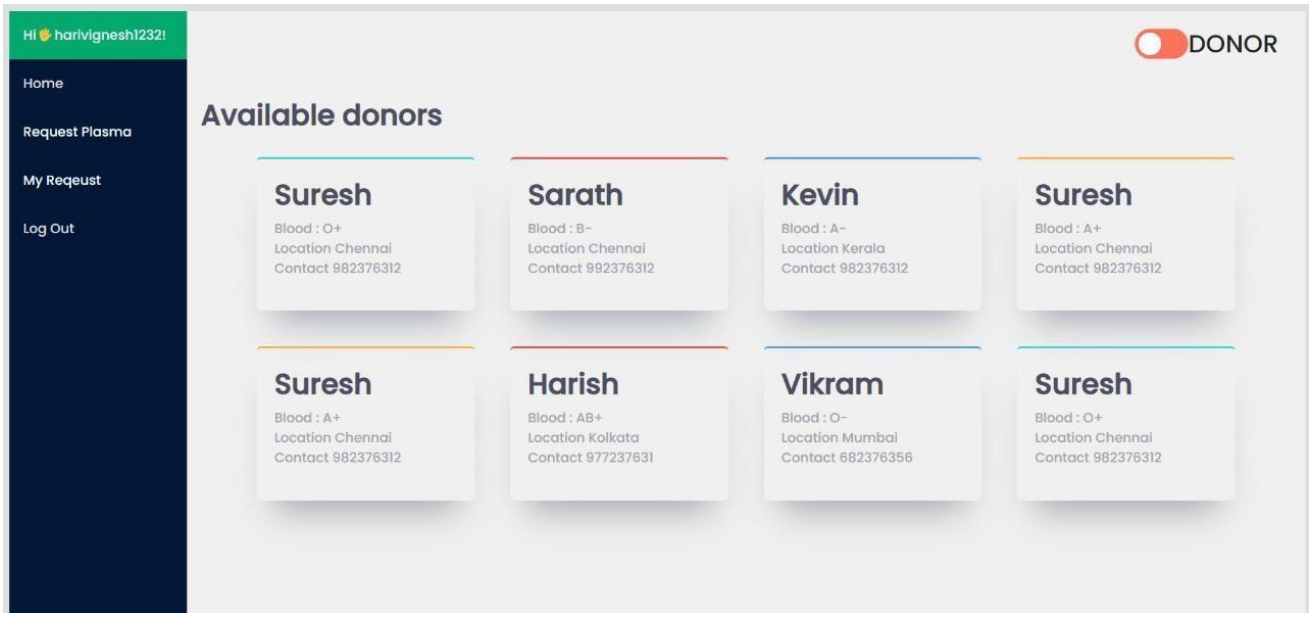
8.3 REGISTRATION PAGE



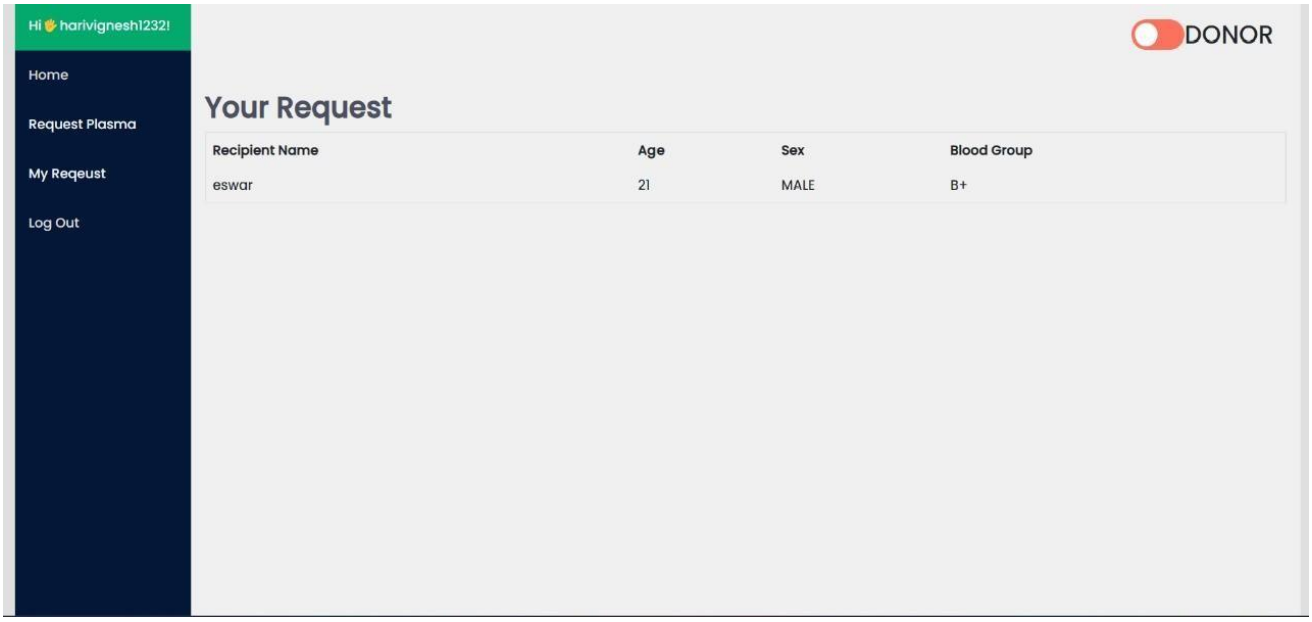
8.4 LOGIN PAGE



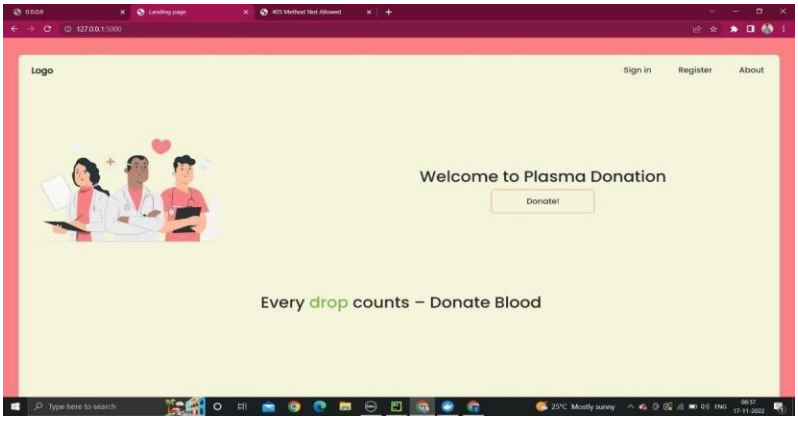
8.5 DASHBOARD

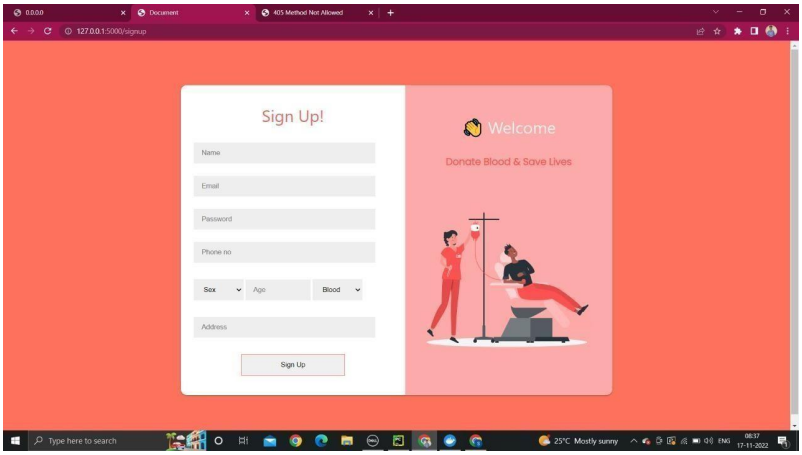


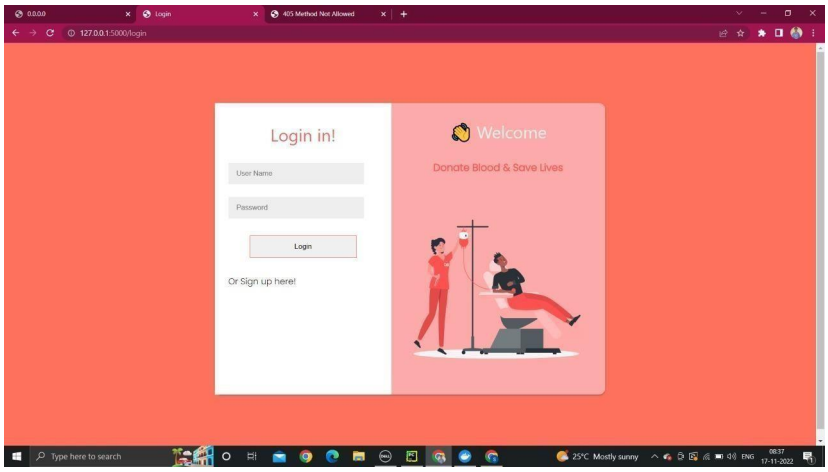
8.6 DONOR LIST

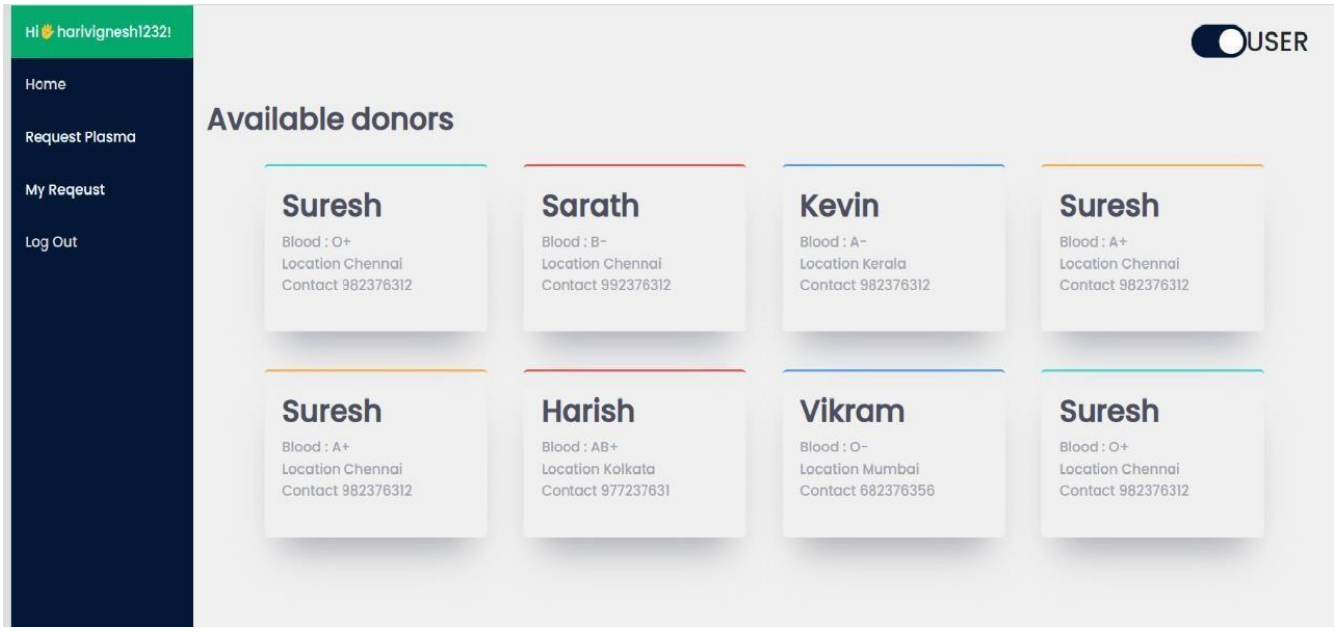


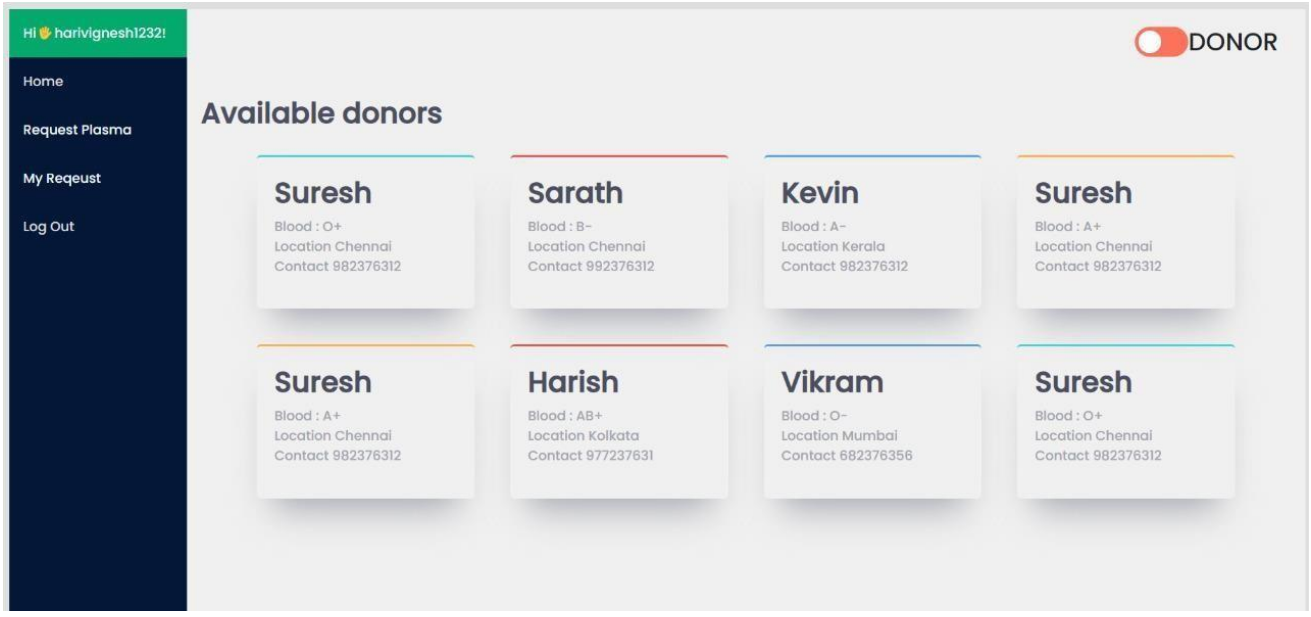
9. RESULT

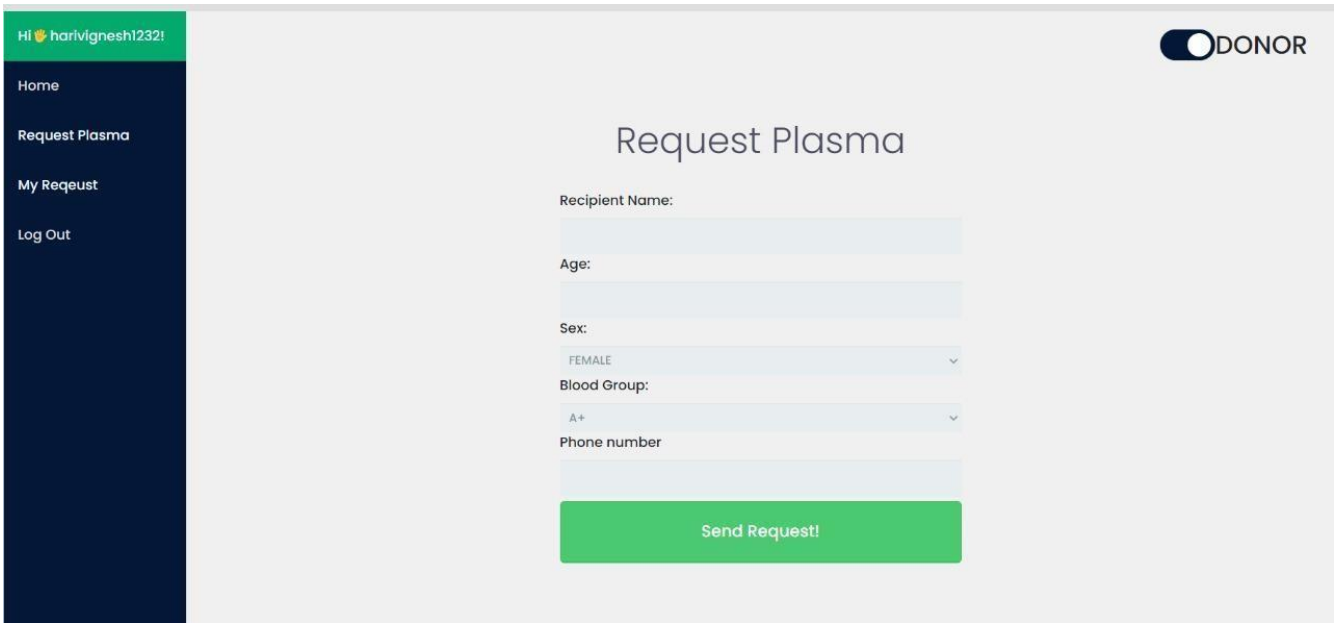

Logo
Sign in
Register
About
Welcome to Plasma Donation
Donate!
Every drop counts – Donate Blood

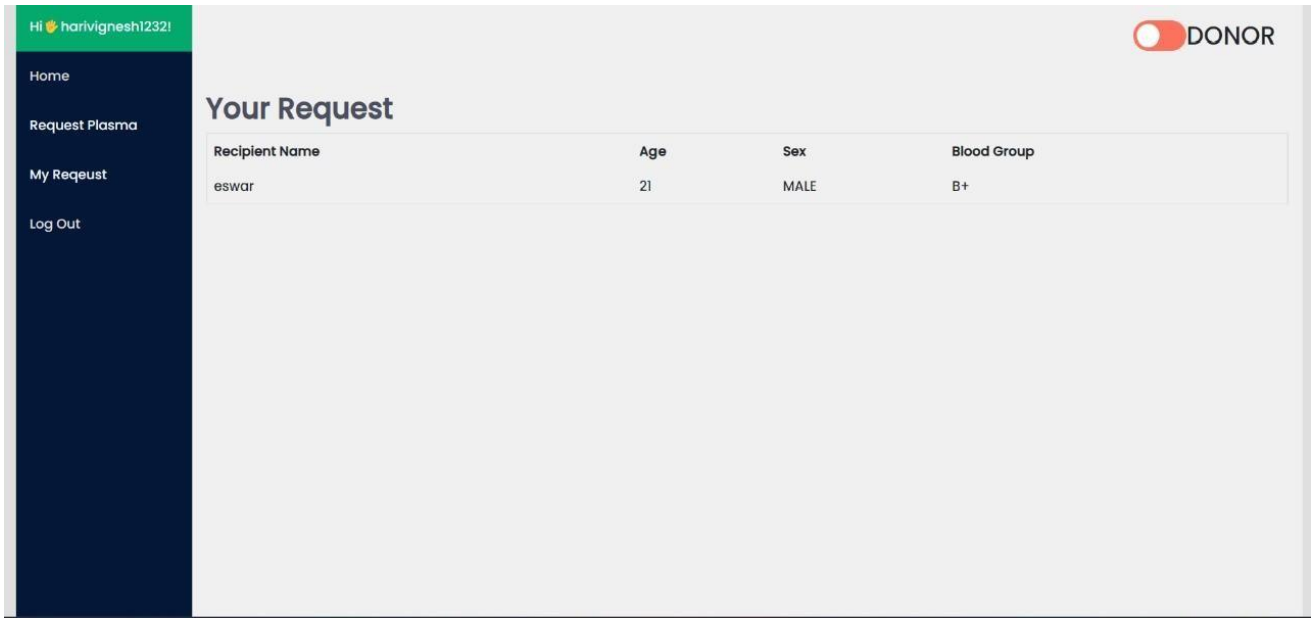

Sign Up!
Name
Email
Password
Phone no
Sex
Age
Blood
Address
Sign Up
Welcome
Donate Blood & Save Lives


Login in!
User Name
Password
Login
Or Sign up here!
Welcome
Donate Blood & Save Lives


Hi harivignesh1232!
Home
Request Plasma
My Reqeust
Log Out
USER
Available donors
Suresh
Blood : O+
Location Chennai
Contact 982376312
Sarath
Blood : B-
Location Chennai
Contact 992376312
Kevin
Blood : A-
Location Kerala
Contact 982376312
Suresh
Blood : A+
Location Chennai
Contact 982376312
Suresh
Blood : A+
Location Chennai
Contact 982376312
Harish
Blood : AB+
Location Kolkata
Contact 977237631
Vikram
Blood : O-
Location Mumbai
Contact 682376356
Suresh
Blood : O+
Location Chennai
Contact 982376312


Hi harivignesh1232!
Home
Request Plasma
My Reqeust
Log Out
DONOR
Available donors
Suresh
Blood : O+
Location Chennai
Contact 982376312
Sarath
Blood : B-
Location Chennai
Contact 992376312
Kevin
Blood : A-
Location Kerala
Contact 982376312
Suresh
Blood : A+
Location Chennai
Contact 982376312
Suresh
Blood : A+
Location Chennai
Contact 982376312
Harish
Blood : AB+
Location Kolkata
Contact 977237631
Vikram
Blood : O-
Location Mumbai
Contact 682376356
Suresh
Blood : O+
Location Chennai
Contact 982376312


Hi harivignesh1232!
Home
Request Plasma
My Reqeust
Log Out
DONOR
Request Plasma
Recipient Name:
Age:
Sex:
FEMALE
Blood Group:
A+
Phone number
Send Request!


Hi harivignesh1232!
Home
Request Plasma
My Reqeust
Log Out
DONOR
Your Request
Recipient Name
Age
Sex
Blood Group
eswar
21
MALE
B+

10.ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- **Speed**: This website is fast and offers great accuracy as compared to manual registered keeping.
- **Maintenance:** Less maintenance is required
- **User Friendly:** It is very easy to use and understand. It is easily workable and accessible for everyone.
- **Fast Results:** It would help you to provide plasma donors easily depending upon the availability of it.

DISADVANTAGES:

- **Internet:** It would require an internet connection for the working of the website.
- **Auto- Verification:** It cannot automatically verify the genuine users.

11. CONCLUSION

The efficient way of finding plasma donors for the infected people is implemented using the plasma donor website that is hosted on Cloud platform. To ensure the smooth functioning of the website operations. I have hosted the website on a cloud platform to make sure the operation or running successfully to deploy the application cloud service.

12.FUTURE ENHANCEMENT

Upgrading the UI that is more user-friendly which will help many users to access the website and also ensures that many plasma donors can be added into the community. Using elastic load balancer, it helps to handle multiple requests at the same time which will maintain the uptime of the website with negligible downtime.

13.APPENDIX

13.1 SAMPLE SOURCE CODE: DONOR

MAIN.py

```
from flask import Flask, redirect, url_for, render_template, request, make_response, jsonify, request

import ibm_db

from flask import request

import json


conn = ibm_db.connect( "DATABASE=bludb;HOSTNAME=764264db-
    9824-4b7c-82df-
40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=32536;SECURITY=SSL;SSLServerCertificate=abc.crt;UID=gnq12618;PWD=0glS4tFaR2ciK8fB ",
    '', '')

print(conn)

print("connection successful...")

app = Flask(__name__)

import os

from sendgrid import SendGridAPIClient

from sendgrid.helpers.mail import Mail



@app.route('/')def

home():

    return render_template("landing.html")
```

```
@app.route('/home')
def dash():
    return render_template("dashboard.html")


@app.route('/login', methods=['POST', 'GET'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        sql = "select * from user where username=? and password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.bind_param(stmt, 2, password)
        ibm_db.execute(stmt)
        dic = ibm_db.fetch_assoc(stmt)
        print(dic)
        role = str()
        requests = []if
        dic:
            role = dic['ROLE']
            # sql = "select * from user where blood_group=?"#
            stmt = ibm_db.prepare(conn, sql)
            # ibm_db.bind_param(stmt, 1, username)#
            ibm_db.execute(stmt)
            # dic = ibm_db.fetch_assoc(stmt)
```

```
            # while dic != False:
            #     single_request = {
            #         'name': dic['NAME'],
            #         'age': dic['AGE'],
            #         'sex': dic['SEX'],
            #         'blood_type': dic['BLOOD_TYPE']#   }
            #     print(single_request)
            #     requests.append(single_request)#
                    dic = ibm_db.fetch_assoc(stmt)
            return render_template('dashboard.html', username=username, role=role)


        else:
            return redirect(url_for('login'))
        return redirect(url_for('home'))
    elif request.method == 'GET':
        return render_template('login.html')


@app.route('/signup', methods=['POST', 'GET'])
def signup():
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email'] password
        = request.form['password']roll_no =
        request.form['roll_no']
```

```
        sex = request.form['sex']age =
        request.form['age']
        address =  request.form['address'] blood_group
        = request.form['blood_group'] sql = "insert into
        user values(?,?,?,?,?,?,?,?,?)"prep_stmt =
        ibm_db.prepare(conn, sql)
        ibm_db.bind_param(prep_stmt, 1, username)
        ibm_db.bind_param(prep_stmt, 2, email)
        ibm_db.bind_param(prep_stmt, 3, password)
        ibm_db.bind_param(prep_stmt, 4, roll_no)
        ibm_db.bind_param(prep_stmt, 5, sex)
        ibm_db.bind_param(prep_stmt, 6, age)
        ibm_db.bind_param(prep_stmt, 7, "USER")
        ibm_db.bind_param(prep_stmt, 8, address)
        ibm_db.bind_param(prep_stmt, 9, blood_group)
        ibm_db.execute(prep_stmt)
        # db post operation
        return redirect(url_for('login'))
    elif request.method == 'GET':
        return render_template('signup.html')


@app.route('/toggle', methods=['POST'])
def toggle_user():
    data = request.get_json(force=True)

    username = data['username']
```

```
    role = data['role']
    print(username)
    print(role)
    sql = "update user set role=? where username=?"
    prep_stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(prep_stmt, 1, role)
    ibm_db.bind_param(prep_stmt, 2, username)
    ibm_db.execute(prep_stmt)
    return jsonify(
        status="success",
        role=role
    )


@app.route('/requestPalsma', methods=['POST'])
def requestBloodPlasma():
    # fetch mail address of the donors data
    = request.get_json(force=True)
    username = data['username']
    name = data['name']age
    =  data['age'] sex =
    data['sex']
    blood_type = data['bloodtype']
    phone_number = data['phone_num']
    sql = "select email from user where blood_group=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, blood_type)
```

```
    ibm_db.execute(stmt)
    dic = ibm_db.fetch_assoc(stmt)
    email_list = []
    while dic != False:
        email_list.append(dic['EMAIL'])
        print(dic['EMAIL'])
        dic = ibm_db.fetch_assoc(stmt)#
    send mail

    message = Mail(
        from_email='eshwaran.s.2019.cse@rajalakshmi.edu.in',
        to_emails=email_list,
        subject='Sending with Twilio SendGrid is Fun',
        html_content='<h1>Need Of Blood</h1><table><tr><th>Name</th><th>' + name
+ '</th></tr><tr><th>Age</th><th>' + age + '</th></tr><tr><th>Sex</th><th>'
+ sex + '</th></tr><tr><th>Blood Group</th><th>' + blood_type +
'</th></tr><tr><th>Phone Number</th><th>' + phone_number + '</th></tr></table>'
    )
    try:
        sg = SendGridAPIClient("SG.3iBLSgAYTEuVbfSHu9dCPA.-
nrnikWJvaRlNLMONA04_CuKAyPeV69c46vPAh3vUX0")
        response = sg.send(message)
        print(response.status_code)
        print(response.body)
        print(response.headers)
    except Exception as e:
        print(e.message)
    # insert data into requests table
```

```
    sql = "insert into bloodrequests(username,name,age,sex,blood_type) values (?,?,?,?,?)"
    prep_stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(prep_stmt, 1, username)
    ibm_db.bind_param(prep_stmt, 2, name)
    ibm_db.bind_param(prep_stmt, 3, age)
    ibm_db.bind_param(prep_stmt, 4, sex)
    ibm_db.bind_param(prep_stmt, 5, blood_type)
    ibm_db.execute(prep_stmt)

    return jsonify(
        name=name,
        age=age,
        sex=sex,
        bloodtype=blood_type,
        status="yes"
    )


@app.route('/getrequests', methods=['POST'])
def getBloodRequests():
    data = request.get_json(force=True)
    username = data['username']
    sql = "select * from bloodrequests where username=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, username)
    ibm_db.execute(stmt)
    dic = ibm_db.fetch_assoc(stmt)
```

```
    requests = []
    print(type(dic)) while
    dic != False:
        single_request = { 'name':
            dic['NAME'],
            'age': dic['AGE'],
            'sex': dic['SEX'],
            'blood_type': dic['BLOOD_TYPE']
        }
        print(single_request)
        requests.append(single_request)
        dic = ibm_db.fetch_assoc(stmt)
    return jsonify(
        username=username,
        requests=requests
    )


if__name__== '__main__':
    app.run(host="0.0.0.0", debug=True)
```

GITHUB LINK :

https://github.com/IBM-EPBL/IBM-Project-46015-1660734713

DEMO LINK :

https://drive.google.com/file/d/1Ned1AgL1E6_b-o6fGR3l6l5kOC0H6TiW/view?usp=drivesdk