PLASMA DONOR APPLICATION

A PROJECT REPORT

Submitted by

INDUSTRY MENTOR:NAVYA
FACULTY MENTOR:POORANAM.N

TEAM ID: PNT2022TMID02762
TEAM LEAD:MUGILANANDAM R (19EUCS095)
TEAM MEMBER:KISHORE RAJ D (19EUCS065)
TEAM MEMBER:RAMYA J (19EUCS113)
TEAM MEMBER:MOUNISHA J (19EUCS093)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING
SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution, Affiliated to Anna University Chennai - 600 025)

NOVEMBER 2022

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution, Affiliated to Anna University, Chennai)
Kuniamuthur, Coimbatore - 641 008

SRI KRISHNA INSTITUTIONS

BONAFIDE CERTIFICATE

Certified that this project report titled **“PLASMA DONOR APPLICATION”** is the bonafide work of **Mr. MUGILANANDAM (19EUCS095), Mr. KISHORE RAJ D (19EUCS065), Miss. RAMYA J (19EUCS113), Miss. MOUNISHA J (19EUCS093)** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. K. SASI KALA RANI, M.E, Ph.D.,	**Ms. POORANAM N**
HEAD OF THE DEPARTMENT	**SUPERVISOR**

Department of Computer Science and Engineering
Sri Krishna College of Engineering and Technology
Kuniamuthur, Coimbatore

Submitted for the Project viva-voce examination held on___________

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

ABSTRACT

During the COVID 19 crisis, the requirement of plasma became a high priority and the donor count has become low. Saving the donor information and helping the needy by notifying the current donors list, would be a helping hand. Alternatively, now a day’s plasma transplant surgery is also being performed rapidly. At this present time plasma banks are in short supply. Not only that, but the number of plasma donors is low too. And some people do not know what plasma donation is and where to donate plasma. We have set up a system to alleviate this situation and help needy people to identify plasma donors and plasma banks. 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. The main objective is to develop an Android application to build a network of people (Donors, Recipients and Health care departments) who can help each other. This application is developed to simply explore for plasma in near areas for emergency.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	**III**
	LIST OF FIGURES	**VI**
1	**INTRODUCTION**	**7**
	1.1 Project Overview	7
	1.2 Purpose	8
2	**LITERATURE SURVEY**	
	2.1 Existing problem	9
	2.2 References	10
	2.3 Problem Statement Definition	11
3	**IDEATION & PROPOSED SOLUTION**	**12**
	3.1 Empathy Map Canvas	12
	3.2 Ideation & Brainstorming	13
	3.3 Proposed Solution	15
	3.4 Problem Solution fit	16
4	**REQUIREMENT ANALYSIS**	17
	4.1 Functional requirement	17
	4.2 Non-Functional requirements	18
5	**PROJECT DESIGN**	19
	5.1 Data Flow Diagrams	19
	5.2 Solution & Technical Architecture	19
	5.3 User Stories	20
6	**PROJECT PLANNING & SCHEDULING**	**22**
	6.1 Sprint Planning & Estimation	22
	6.2 Sprint Delivery Schedule	23

	6.3 Reports from JIRA	23
7	**CODING & SOLUTIONING**	**24**
	7.1 Feature 1	24
	7.2 Feature 2	27
8	**TESTING**	**29**
	8.1 Test Cases	29
	8.2 User Acceptance Testing	29
9	**RESULTS**	**31**
	9.1 Performance Metrics	31
10	**ADVANTAGES & DISADVANTAGES**	**32**
11	**CONCLUSION**	**33**
12	**FUTURE SCOPE**	**34**
13	**APPENDIX**	**34**
	Source Code	34
	GitHub & Project Demo Link	50

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
1.1	Empathy Map	12
1.2	Problem Statement & Brainstorm	14
1.3	Group Ideas & Prioritize	16
1.4	Problem Solution fit	17
2.1	Data Flow Diagram	19
2.2	Architecture diagram	20
3.1	Sprint Delivery Schedule	23
3.2	Sprint Report	23

INTRODUCTION

1.1 OVERVIEW

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. The main goal of our project is to design a user-friendly web application that is like a scientific vehicle from which we can help reduce mortality or help those who are need of plasma , plasma therapy is an experimental approach to treat those COVID-positive patients and help them recover faster. Therapy, which is considered reliable and safe. If a particular person has fully recovered from COVID19, they are eligible to donate their plasma. As we all know, the traditional methods of finding plasma, one has to find out for oneself by looking at hospital records and contacting donors have been recovered, sometimes may not be available at home and move to other places. In this type of scenario, the health of those who are sick becomes disastrous. Therefore, it is not considered a rapid process to find plasma.

1.2 PURPOSE

A user friendly and responsive interface with a quick notification system which instantly notifies the donor upon receiving a request. When the recipient requests for plasma, if there is lack of plasma at the time of request, automatically the recipient will be added to the waiting list. Later when there is availability of plasma, the recipient will be notified by email.The main purpose of the proposed system, the donor who wants to donate plasma can register and can donate the plasma to the blood bank, the recipient can request for the donor and once the donor has accepted the request, the donor can donate blood at blood bank and the recipient can also track the status of the request for plasma and can take the plasma from the blood bank.

LITERATURE SURVEY

2.1 EXISTING PROBLEM

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 . Alternatively, now a day’s plasma transplant surgery is also being performed rapidly. At this present time plasma banks are in short supply. Not only that, but the number of plasma donors is low too. And some people do not know what plasma donation is and where to donate plasma.

2.2 REFERENCES

[1]http://www.oaijse.com/VolumeArticles/FullTextPDF/253_49_E_IMPLEME

NTATION_OF_B

LOOD_DONATION_APPLICATION_USING.pdf

[2] https://nevonprojects.com/instant-plasma-donor-recipient-connector-

android-app/

[3]https://www.researchgate.net/publication/263052781_A_GeoLocation_base

d_Mobile_Service_that_Dynamically_Locates_and_Notifies_the_nearest_Bloo

d_Donors_for_Blood_Donation_during_Medical_Emergencies

[4]https://www.researchgate.net/publication/350836827_Nearest_Blood_Plasm

a_Donor_Finding_A_Machine_Learning_Approach

[5]https://pubmed.ncbi.nlm.nih.gov/29184892/

[6]https://www.researchgate.net/publication/338927164_A_Webbased_blood_

donation_and_Medical_Monitoring_System_Integrating_Cloud_services_and_

Mobile_Application

2.3 PROBLEM STATEMENT DEFINITION

During the COVID 19 crisis, the requirement of plasma became a high priority and the donor count has become low. Saving the donor information and helping the needy by notifying the current 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**.** 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. An application should be developed which would take the donor details, store them and notify them upon a request.

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

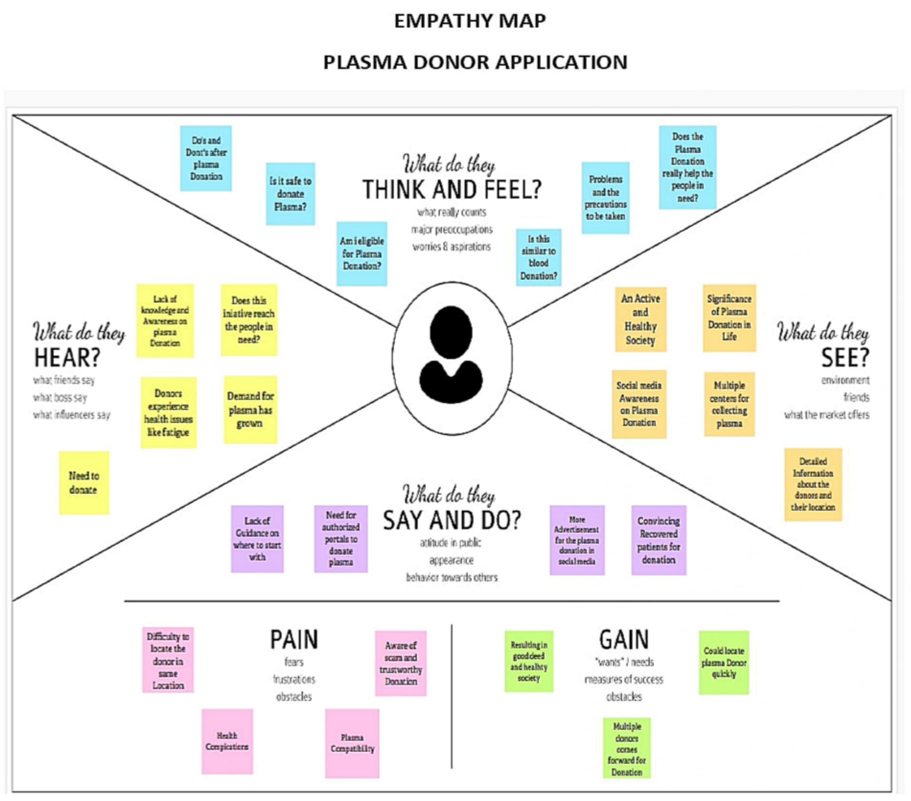
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes.

It is a useful tool to helps teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it.

The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.

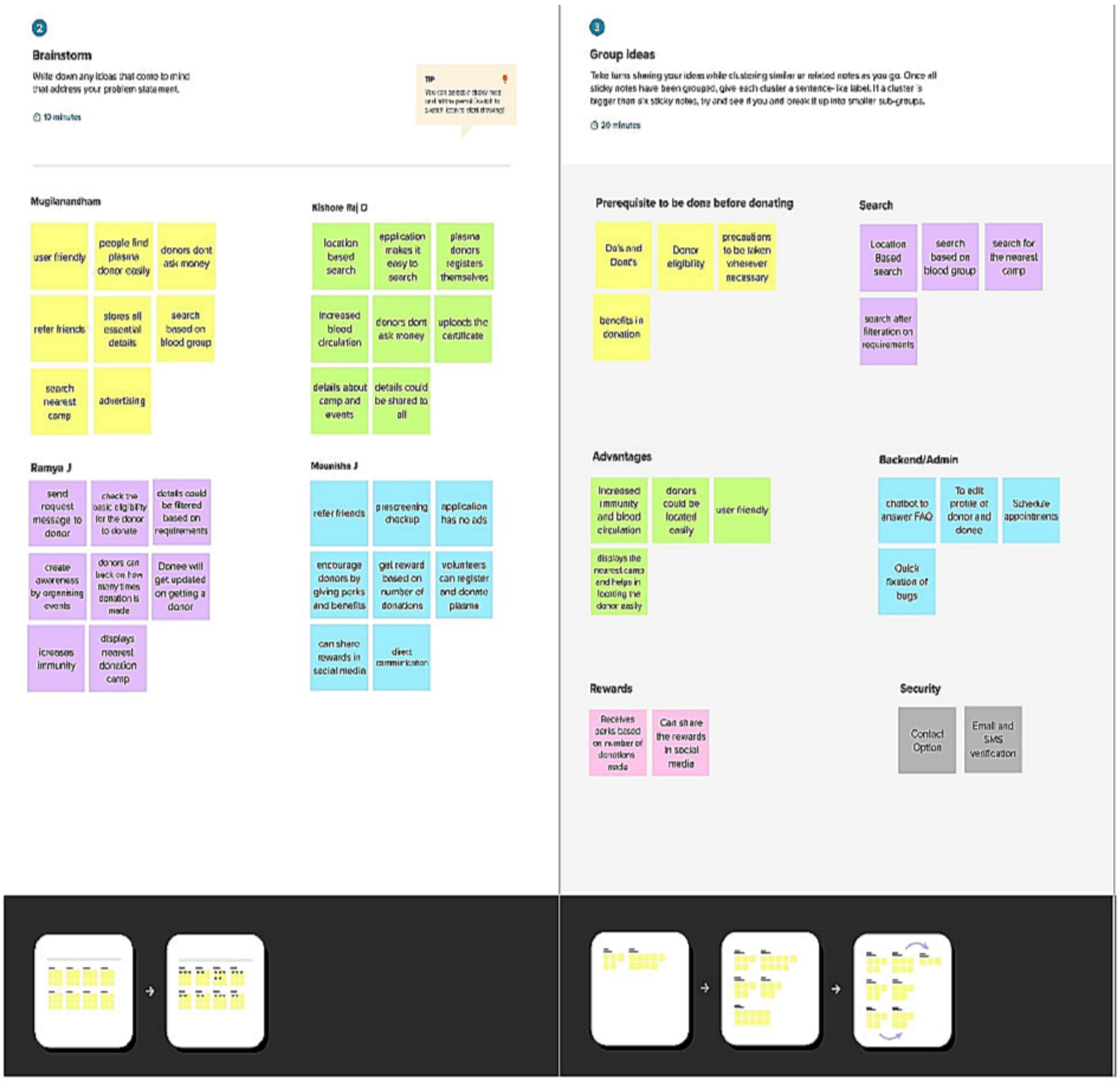
Empathy Map for Plasma Donor Application :



EMPATHY MAP
PLASMA DONOR APPLICATION
What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Do's and Dont's after plasma Donation
Is it safe to donate Plasma?
Am i eligible for Plasma Donation?
Is this similar to blood Donation?
Problems and the precautions to be taken
Does the Plasma Donation really help the people in need?
What do they HEAR?
what friends say
what boss say
what influencers say
Lack of knowledge and Awareness on plasma Donation
Does this iniative reach the people in need?
Donors experience health issues like fatigue
Demand for plasma has grown
Need to donate
What do they SEE?
environment
friends
what the market offers
An Active and Healthy Society
Significance of Plasma Donation in Life
Social media Awareness on Plasma Donation
Multiple centers for collecting plasma
Detailed Information about the donors and their location
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Lack of Guidance on where to start with
Need for authorized portals to donate plasma
More Advertisement for the plasma donation in social media
Convincing Recovered patients for donation
PAIN
fears
frustrations
obstacles
Difficulty to locate the donor in same Location
Aware of scam and trustworthy Donation
Health Complications
Plasma Compatibility
GAIN
"wants" / needs
measures of success
obstacles
Resulting in good deed and healthy society
Could locate plasma Donor quickly
Multiple donors comes forward for Donation

Fig 1.1.Empathy Map

3.2 IDEATION & BRAINSTORMING



2

Brainstorm

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

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Mugilanandham

- user friendly
- people find plasma donor easily
- donors dont ask money
- refer friends
- stores all essential details
- search based on blood group
- search nearest camp
- advertising

Kishore Raj D

- location based search
- application makes it easy to search
- plasma donors registers themselves
- Increased blood circulation
- donors dont ask money
- uploads the certificate
- details about camp and events
- details could be shared to all

Ramya J

- send request message to donor
- check the basic eligibility for the donor to donate
- details could be filtered based on requirements
- create awareness by organising events
- donors can track on how many times donation is made
- Donee will get updated on getting a donor
- icreases immunity
- displays nearest donation camp

Mounisha J

- refer friends
- prescreening checkup
- application has no ads
- encourage donors by giving perks and benefits
- get reward based on number of donations
- volunteers can register and donate plasma
- can share rewards in social media
- direct communication

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

Prerequisite to be done before donating

- Do's and Dont's
- Donor eligibility
- precautions to be taken wherever necessary
- benefits in donation

Search

- Location Based search
- search based on blood group
- search for the nearest camp
- search after filteration on requirements

Advantages

- Increased immunity and blood circulation
- donors could be located easily
- user friendly
- displays the nearest camp and helps in locating the donor easily

Backend/Admin

- chatbot to answer FAQ
- To edit profile of donor and donee
- Schedule appointments
- Quick fixation of bugs

Rewards

- Receives perks based on number of donations made
- Can share the rewards in social media

Security

- Contact Option
- Email and SMS verification

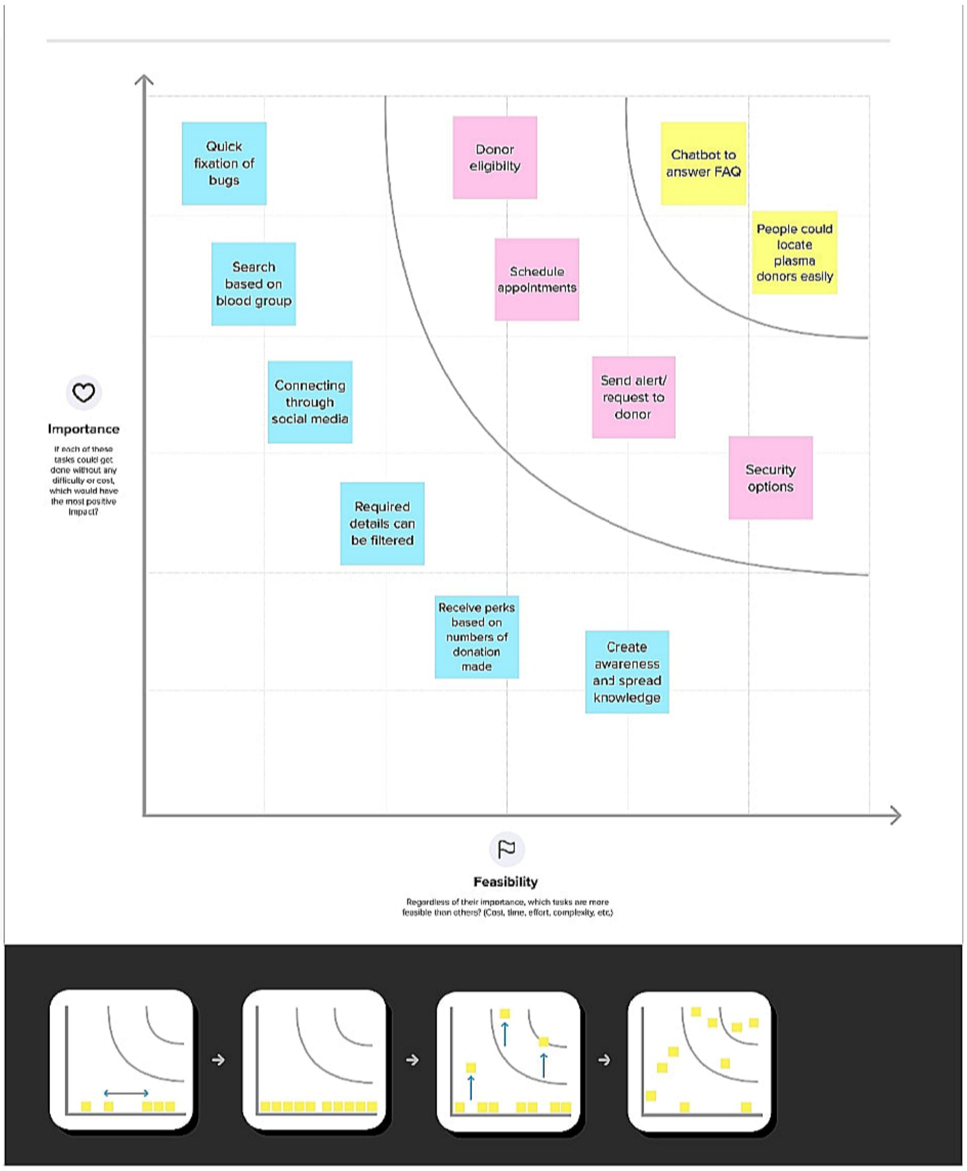
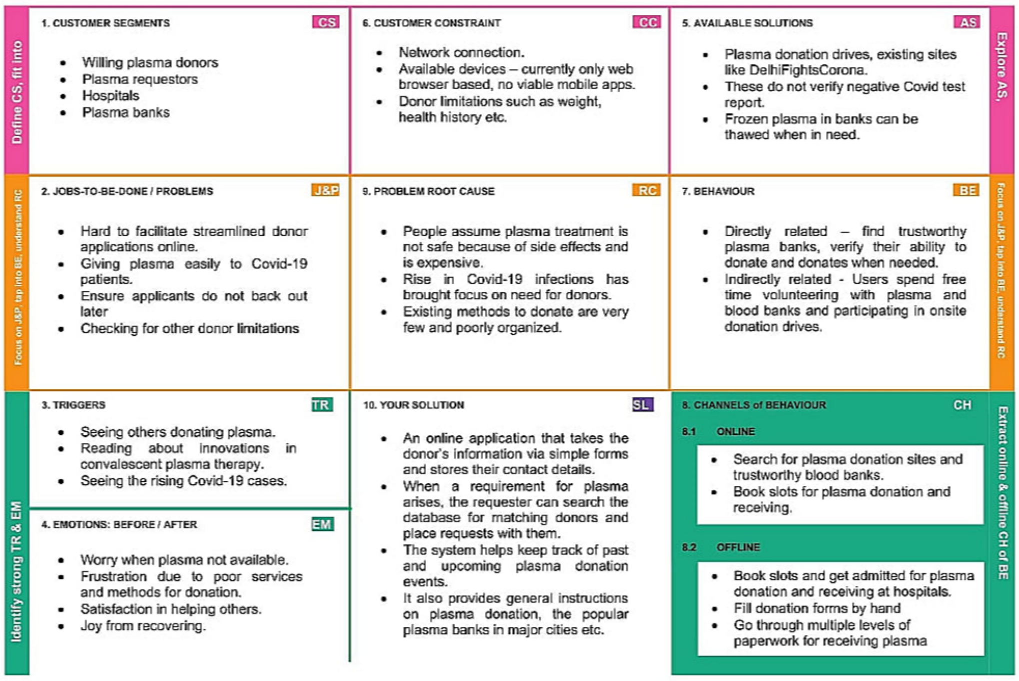

Quick fixation of bugs
Donor eligibilty
Chatbot to answer FAQ
Search based on blood group
Schedule appointments
People could locate plasma donors easily
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Connecting through social media
Send alert/ request to donor
Security options
Required details can be filtered
Receive perks based on numbers of donation made
Create awareness and spread knowledge
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

Fig 1.2.Group Ideas & Prioritize

3.3 PROPOSED SOLUTION

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. An application should be developed which would take the donor details, store them and notify them upon a request. A user friendly and responsive interface with a quick notification system which instantly notifies the donor upon receiving a request. When the recipient requests for plasma, if there is lack of plasma at the time of request, automatically the recipient will be added to the waiting list. Later when there is availability of plasma, the recipient will be notified by email. The application seamlessly connects the donor and the recipient. It will create an awareness among the people about donation of plasma which will be done in an easy way of connecting the donor and the recipient. And for sure the patient will be satisfied. Revenue can be generated by selling ad space to health product advertisers and by donation based monetization. Since the app is going to be deploy in a cloud kubernetes cluster, it will continue to be efficient when large number of people uses it. There will be no down time. And it has the ability to increase or decrease IT resources as needed to meet changing demand.

3.4 Problem Solution fit



Define CS, fit into

1. CUSTOMER SEGMENTS CS

- Willing plasma donors
- Plasma requestors
- Hospitals
- Plasma banks

6. CUSTOMER CONSTRAINT CC

- Network connection.
- Available devices – currently only web browser based, no viable mobile apps.
- Donor limitations such as weight, health history etc.

5. AVAILABLE SOLUTIONS AS

- Plasma donation drives, existing sites like DelhiFightsCorona.
- These do not verify negative Covid test report.
- Frozen plasma in banks can be thawed when in need.

Explore AS,

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

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- Hard to facilitate streamlined donor applications online.
- Giving plasma easily to Covid-19 patients.
- Ensure applicants do not back out later
- Checking for other donor limitations

9. PROBLEM ROOT CAUSE RC

- People assume plasma treatment is not safe because of side effects and is expensive.
- Rise in Covid-19 infections has brought focus on need for donors.
- Existing methods to donate are very few and poorly organized.

7. BEHAVIOUR BE

- Directly related – find trustworthy plasma banks, verify their ability to donate and donates when needed.
- Indirectly related - Users spend free time volunteering with plasma and blood banks and participating in onsite donation drives.

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

Identify strong TR & EM

3. TRIGGERS TR

- Seeing others donating plasma.
- Reading about innovations in convalescent plasma therapy.
- Seeing the rising Covid-19 cases.

4. EMOTIONS: BEFORE / AFTER EM

- Worry when plasma not available.
- Frustration due to poor services and methods for donation.
- Satisfaction in helping others.
- Joy from recovering.

10. YOUR SOLUTION SL

- An online application that takes the donor's information via simple forms and stores their contact details.
- When a requirement for plasma arises, the requester can search the database for matching donors and place requests with them.
- The system helps keep track of past and upcoming plasma donation events.
- It also provides general instructions on plasma donation, the popular plasma banks in major cities etc.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

- Search for plasma donation sites and trustworthy blood banks.
- Book slots for plasma donation and receiving.

8.2 OFFLINE

- Book slots and get admitted for plasma donation and receiving at hospitals.
- Fill donation forms by hand
- Go through multiple levels of paperwork for receiving plasma

Extract online & offline CH of BE

Fig 1.4.Problem Solution fit

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Website
FR-2	User Confirmation	Confirmation via Email
FR-3	User Login	Login through registered email id
FR-4	Send Request	Patient should fill their details and make a request
FR-5	Contact Donor	Donor and Patient contact by the details shared via email

4.2 NON-FUNCTIONAL REQUIREMENTS

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

NFR No.	**Non Functional Requirement (Epic)**	**Description**
NFR-1	Usability	The plasma Donor application is user friendly and does not involve any complex process
NFR-2	Security	The donor/recipient details are stored in a secured cloud based database.
NFR-3	Reliability	The application will have no down time so that you can always rely on and the information provided by it are so reliable
NFR-4	Performance	The application will work efficiently in emergency situations with an instant notification system.
NFR-5	Availability	The application will be available online 24x7
NFR-6	Scalability	The application can be accessed by multiple users at the same time and it has the ability to increase or decrease the IT resources as needed.

PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

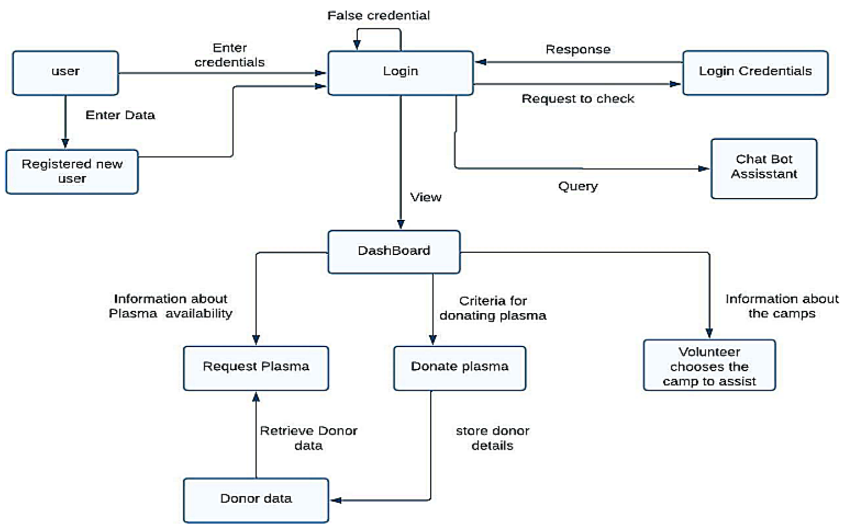


Fig 2.1.Data Flow Diagram

5.2 SOLUTION & TECHNICAL ARCHITECTURE

An application should be developed which would take the donor details, store them and notify them upon a request. A user friendly and responsive interface with a quick notification system which instantly notifies the donor upon receiving a request. When the recipient requests for plasma, if there is lack of plasma at the time of request, automatically the recipient will be added to the waiting list. Later when there is availability of plasma, the recipient will be notified by email.

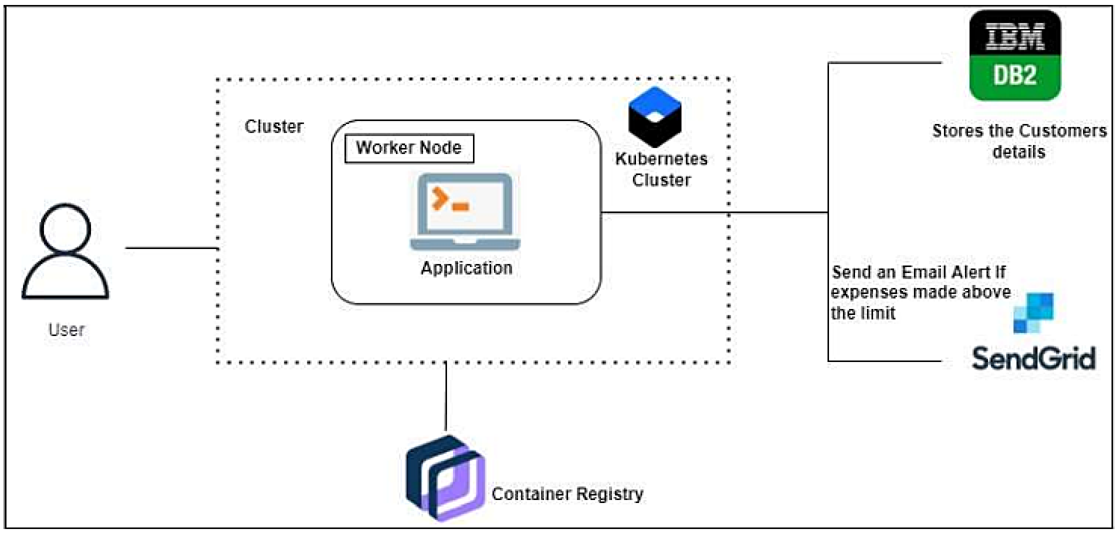


Fig 2.2. Architecture diagram

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 by entering required information.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation pop-up .	I can receive confirmation email & click confirm	High	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering valid credentials	I can access into my User profile and view details in dashboard	High	Sprint-2
	Dashboard	USN-5	As a user,I can donate and request plasma.	I can receive appropriate notifications through email	High	Sprint-3
Customer	Login	USN-6	As a user, I can	I can access	High	Sprint-2

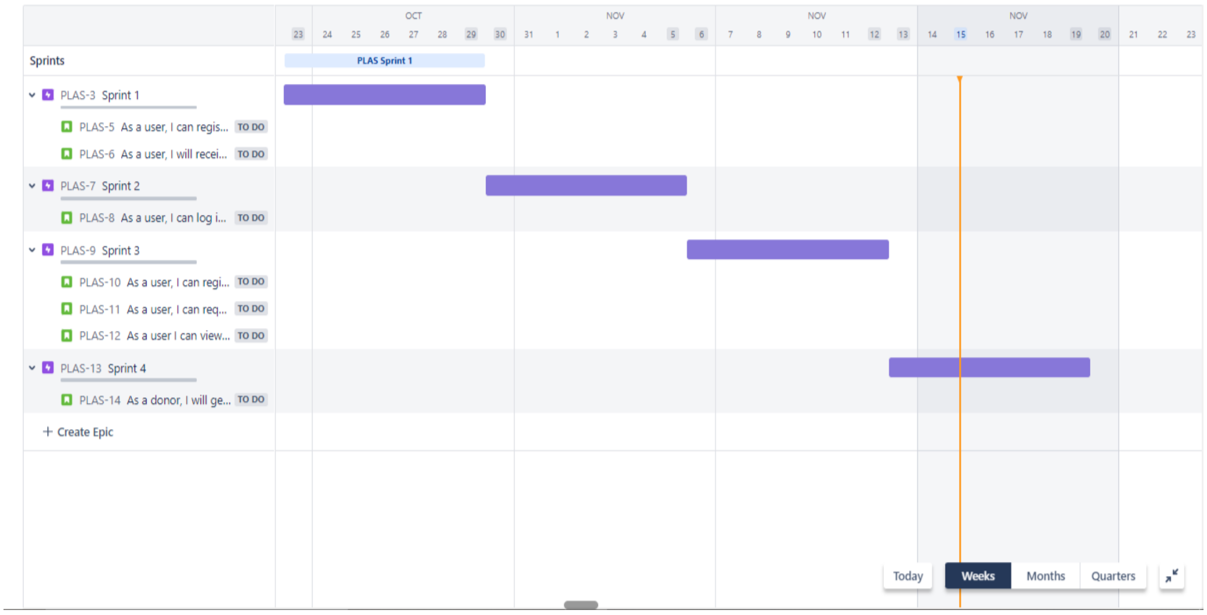
(Web user)			register and log into the application by entering email & password toview the profile	into my user profile and view details in dashboard		
	Dashboard	USN-7	As a user,I can donate and request plasma.	I can receive appropriate notifications through email	High	Sprint-3
Customer (Registered Donor)	Notification	USN-8	As a donor, I will get notification via email upon a valid request	I will get notification via email upon avalid request	High	Sprint-4

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	Registration	USN-1	As a user, I can register for the application by entering required information.	2	High	Kishore Raj, Mugilanandam
Sprint-1		USN-2	As a user, I will receive confirmation pop-up .	1	High	Mounisha, Ramya
Sprint-2	Login	USN-3	As a user, I can log into the application by entering valid credentials	3	High	Mounisha, Ramya
Sprint- 3	Dashboard	USN-4	As a user, I can register as a donor and donate plasma	3	High	Kishore Raj, Mugilanandam
Sprint- 3		USN-5	As a user, I can request plasma	2	High	Kishore Raj, Mugilanandam
Sprint- 3		USN-6	As a user I can view the stats page which shows the count of donors, plasma available etc.,	2	Medium	Mounisha, Ramya
Sprint-4	Notification	USN-7	As a donor, I will get notification via email upon a valid request	2	High	Kishore Raj, Mugilanandam

6.2 SPRINT DELIVERY SCHEDULE



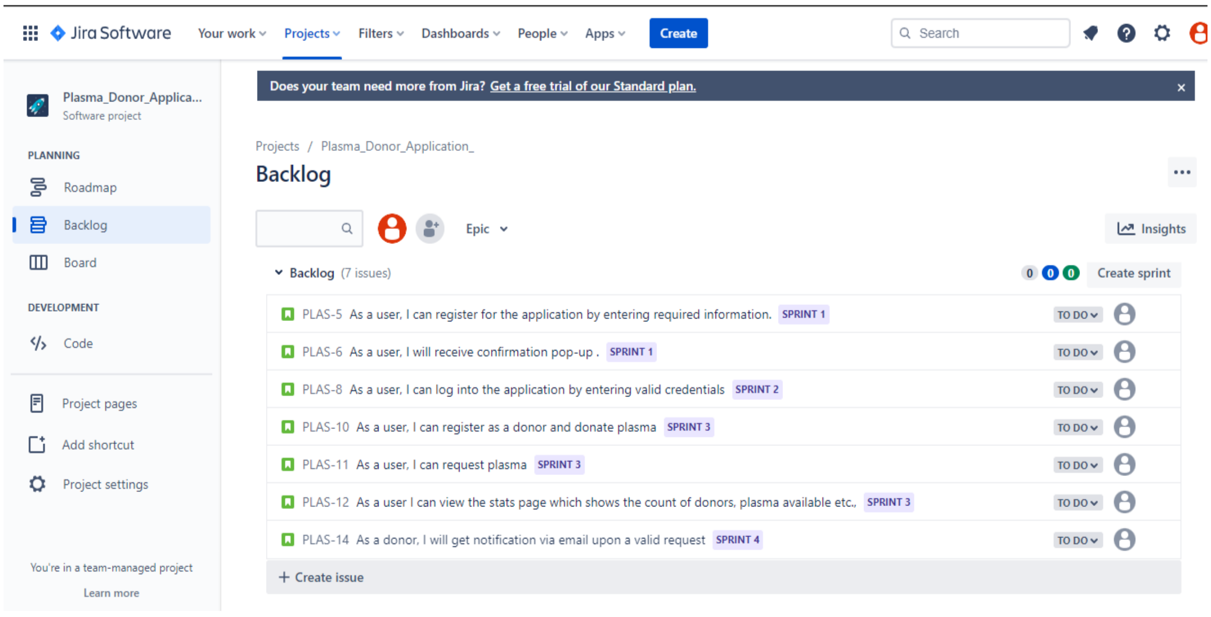
Sprints

- PLAS-3 Sprint 1
  - PLAS-5 As a user, I can regis... TO DO
  - PLAS-6 As a user, I will recei... TO DO
- PLAS-7 Sprint 2
  - PLAS-8 As a user, I can log i... TO DO
- PLAS-9 Sprint 3
  - PLAS-10 As a user, I can regi... TO DO
  - PLAS-11 As a user, I can req... TO DO
  - PLAS-12 As a user I can view... TO DO
- PLAS-13 Sprint 4
  - PLAS-14 As a donor, I will ge... TO DO

+ Create Epic

Fig 3.1. Sprint Delivery Schedule

6.3 REPORTS FROM JIRA



Does your team need more from Jira? Get a free trial of our Standard plan.

Projects / Plasma_Donor_Application_

Backlog

Backlog (7 issues)

PLAS-5	As a user, I can register for the application by entering required information.	SPRINT 1	TO DO
PLAS-6	As a user, I will receive confirmation pop-up .	SPRINT 1	TO DO
PLAS-8	As a user, I can log into the application by entering valid credentials	SPRINT 2	TO DO
PLAS-10	As a user, I can register as a donor and donate plasma	SPRINT 3	TO DO
PLAS-11	As a user, I can request plasma	SPRINT 3	TO DO
PLAS-12	As a user I can view the stats page which shows the count of donors, plasma available etc.,	SPRINT 3	TO DO
PLAS-14	As a donor, I will get notification via email upon a valid request	SPRINT 4	TO DO

Fig 3.2. Sprint report

CODING & SOLUTIONING

7.1 FEATURE 1 – DONOR REGISTRATION

PYTHON SNIPPET:

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32459;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=znq27181;PWD=******",'''',''''')

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

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

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

@app.route('/register', methods=['GET', 'POST'])
def register():
    clean_data = ''
    if request.method == 'POST':
            content = request.get_data()
            # Parse query string part of a URL, and return a dictionary of the data
            x = urllib.parse.parse_qs(content)
            # Remove prefix character 'b
            for k, v in x.items():
```

```
        for i in v:
            clean_data = clean_data + i.decode('utf-8')+"\n"
    clean_data = clean_data.split("\n")
    # Enumerate user data
    username = clean_data[0]
    email = clean_data[1]
    phone = clean_data[2]
    city = clean_data[3]
    infect = clean_data[4]
    blood = clean_data[5]
    password = clean_data[6]
    query = "SELECT * FROM user WHERE email ='"+email+"'"
    stmt = ibm_db.exec_immediate(conn, query)
    row = ibm_db.fetch_assoc(stmt)
    if row:
        messages = {'message':'User already exist. Please login with details'}
        return render_template('register.html', messages=messages)
    else:
        insert_sql = "INSERT INTO  user VALUES (?, ?, ?, ?, ?, ?, ?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1, username)
        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)
        messages = {'message': 'Registration success. Please login'}
        return render_template('register.html', messages=messages)
else:
    return render_template('register.html')
```

```
@app.route('/signin',  methods=['GET', 'POST'])
def signin():
    clean_data = ''
    if request.method == 'POST':
            content = request.get_data()
            # Parse query string part of a URL, and return a dictionary of the data
            x = urllib.parse.parse_qs(content)
            # clean noise that is, remove prefix character 'b
            for k, v in x.items():
                    for i in v:
                            clean_data = clean_data + i.decode('utf-8')+"\n"
            clean_data = clean_data.split("\n")
            # Enumerate user data
            email = clean_data[0]
            passw = clean_data[1]
            sql = "SELECT * FROM user WHERE email =? AND password=?"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.bind_param(stmt,1,email)
            ibm_db.bind_param(stmt,2,passw)
            ibm_db.execute(stmt)
            account = ibm_db.fetch_assoc(stmt)
            if account:
                    return redirect(url_for('statistics'))
            else:
                    messages = {'message': 'Login unsuccessful. Incorrect username / 
password !'}
                    return render_template('login.html', messages=messages)
    return render_template('login.html')
```

7.2 FEATURE 2 – REQUESTS & DONOR STATISTICS

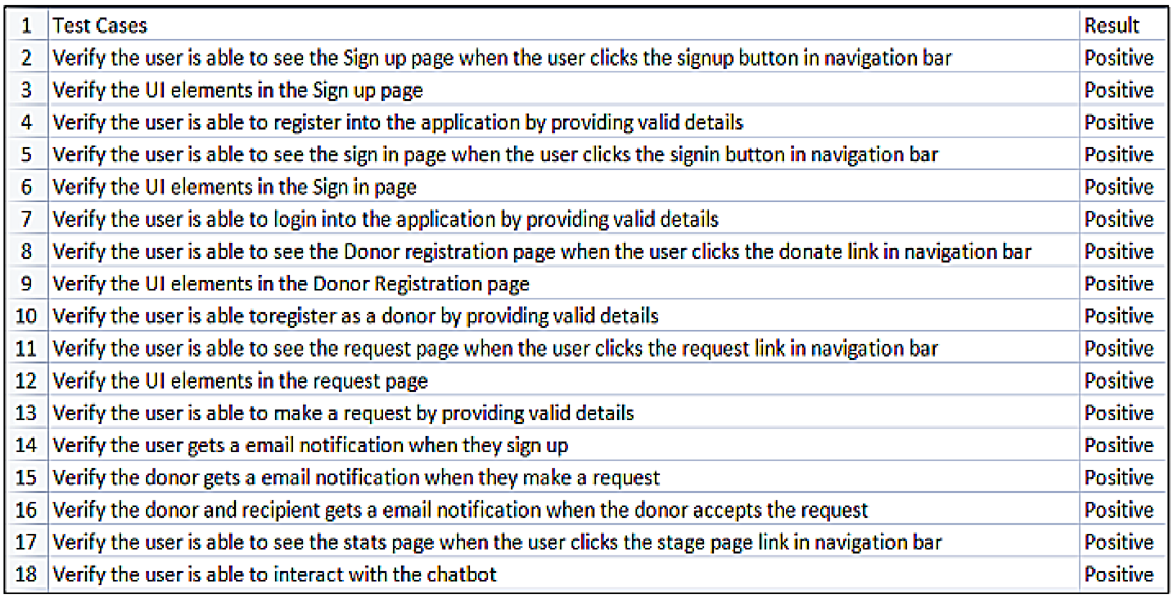
PYTHON SNIPPET:

```
@app.route('/statistics')
def statistics():
    sql = "SELECT blood, count(blood) FROM user group by blood"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    data = []
    while dictionary != False:
            case = {'group': dictionary[0], 'count': dictionary[1]}
            data.append(case)
            dictionary = ibm_db.fetch_both(stmt)
    return render_template('statistics.html', data=data)

@app.route('/requested', methods=['GET','POST'])
def requested():
    clean_data = ''
    if request.method == 'POST':
            content = request.get_data()
            # Parse query string part of a URL, and return a dictionary of the data
            x = urllib.parse.parse_qs(content)
            # clean noise that is, remove prefix character 'b
            for k, v in x.items():
                    for i in v:
                            clean_data = clean_data + i.decode('utf-8')+"\n"
            clean_data = clean_data.split("\n")
            # Enumerate user data
            blood = clean_data[0]
            address = clean_data[1]
            msg = "Need Plasma of your blood group for: "+address
            sql = "SELECT EMAIL FROM user WHERE blood='"+blood+"'"
            stmt = ibm_db.exec_immediate(conn, sql)
            dictionary = ibm_db.fetch_both(stmt)
            while dictionary != False:
                print(dictionary[0])
                mail_msg =
Mail(from_email='mugilanandamr@gmail.com',to_emails=''+dictionary[0],
                                subject='Need
Plasma',html_content='<strong>'+msg+'</strong>')
                sg = SendGridAPIClient("API-KEY")
                mail_response = sg.send(mail_msg)
                print(mail_response)
                dictionary = ibm_db.fetch_both(stmt)
            messages = {'message': 'Your request is sent to the concerned people.'}
            return render_template('request.html', messages=messages)
    return render_template('request.html')
```

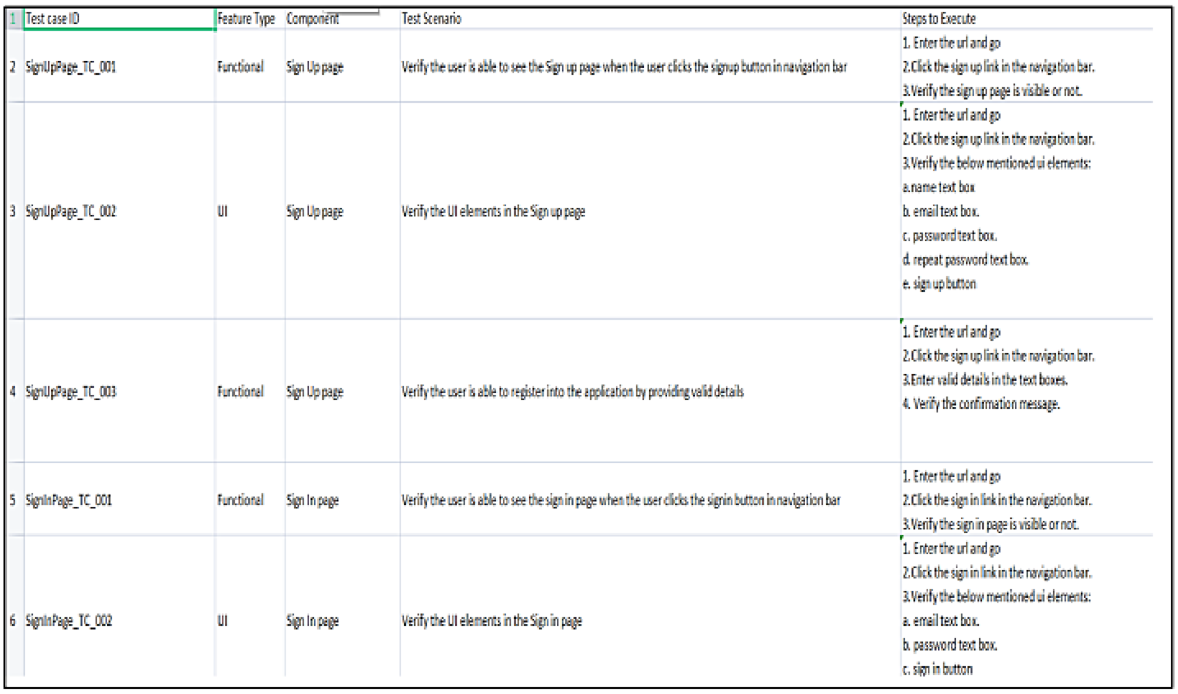
TESTING

8.1 TEST CASES

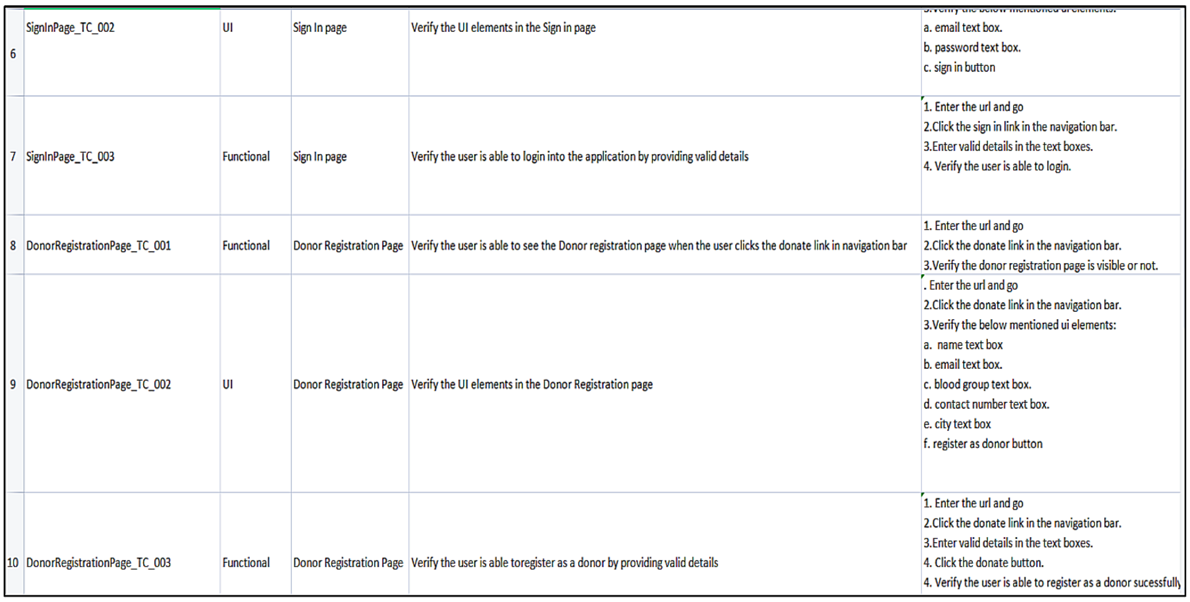


1	Test Cases	Result
2	Verify the user is able to see the Sign up page when the user clicks the signup button in navigation bar	Positive
3	Verify the UI elements in the Sign up page	Positive
4	Verify the user is able to register into the application by providing valid details	Positive
5	Verify the user is able to see the sign in page when the user clicks the signin button in navigation bar	Positive
6	Verify the UI elements in the Sign in page	Positive
7	Verify the user is able to login into the application by providing valid details	Positive
8	Verify the user is able to see the Donor registration page when the user clicks the donate link in navigation bar	Positive
9	Verify the UI elements in the Donor Registration page	Positive
10	Verify the user is able toregister as a donor by providing valid details	Positive
11	Verify the user is able to see the request page when the user clicks the request link in navigation bar	Positive
12	Verify the UI elements in the request page	Positive
13	Verify the user is able to make a request by providing valid details	Positive
14	Verify the user gets a email notification when they sign up	Positive
15	Verify the donor gets a email notification when they make a request	Positive
16	Verify the donor and recipient gets a email notification when the donor accepts the request	Positive
17	Verify the user is able to see the stats page when the user clicks the stage page link in navigation bar	Positive
18	Verify the user is able to interact with the chatbot	Positive

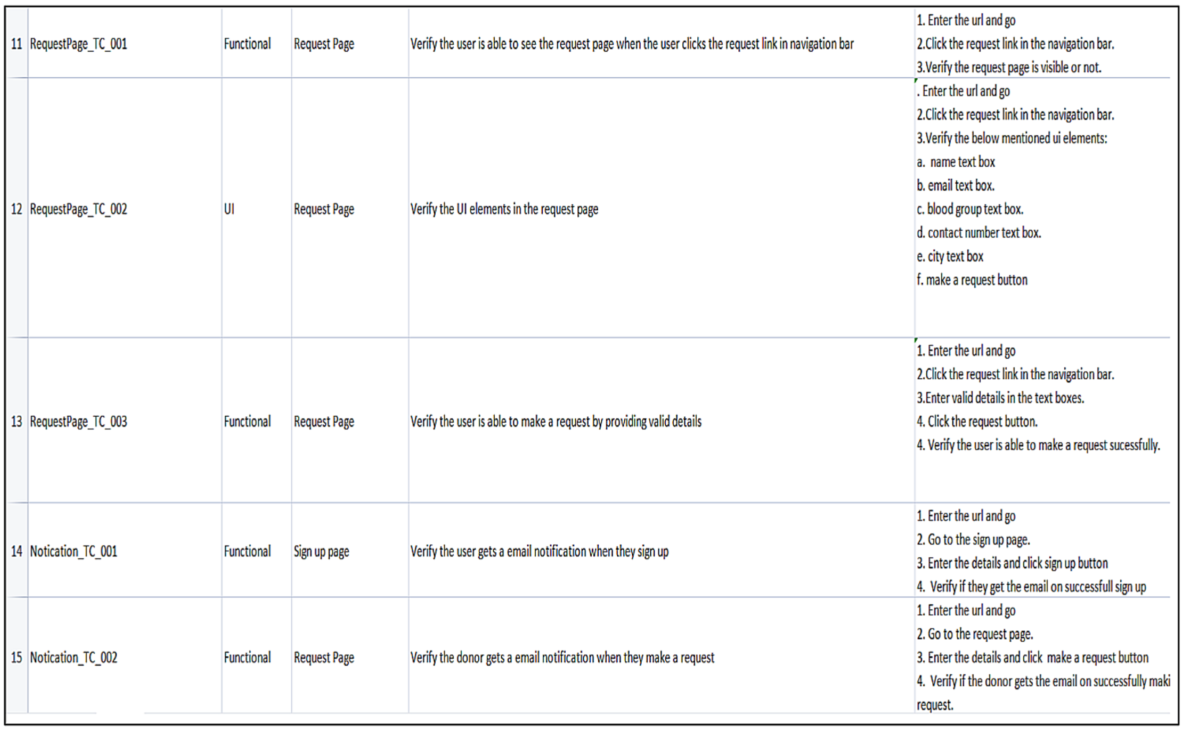
8.2 USER ACCEPTANCE TESTING



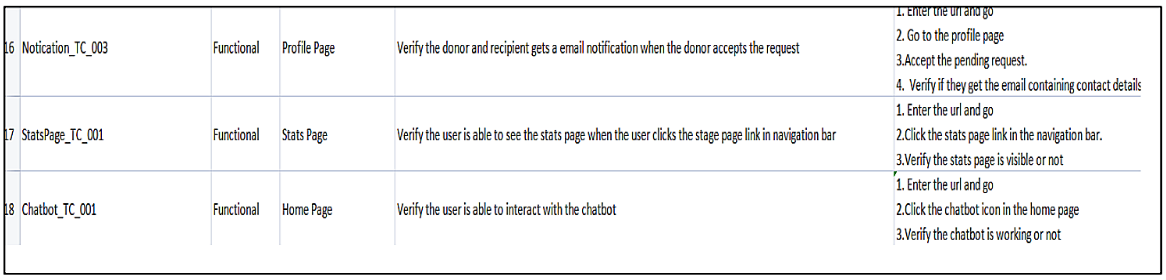
1	Test case ID	Feature Type	Component	Test Scenario	Steps to Execute
2	SignUpPage_TC_001	Functional	Sign Up page	Verify the user is able to see the Sign up page when the user clicks the signup button in navigation bar	1. Enter the url and go 2.Click the sign up link in the navigation bar. 3.Verify the sign up page is visible or not.
3	SignUpPage_TC_002	UI	Sign Up page	Verify the UI elements in the Sign up page	1. Enter the url and go 2.Click the sign up link in the navigation bar. 3.Verify the below mentioned ui elements: a.name text box b. email text box. c. password text box. d. repeat password text box. e. sign up button
4	SignUpPage_TC_003	Functional	Sign Up page	Verify the user is able to register into the application by providing valid details	1. Enter the url and go 2.Click the sign up link in the navigation bar. 3.Enter valid details in the text boxes. 4. Verify the confirmation message.
5	SignInPage_TC_001	Functional	Sign In page	Verify the user is able to see the sign in page when the user clicks the signin button in navigation bar	1. Enter the url and go 2.Click the sign in link in the navigation bar. 3.Verify the sign in page is visible or not.
6	SignInPage_TC_002	UI	Sign In page	Verify the UI elements in the Sign in page	1. Enter the url and go 2.Click the sign in link in the navigation bar. 3.Verify the below mentioned ui elements: a. email text box. b. password text box. c. sign in button



6	SignInPage_TC_002	UI	Sign In page	Verify the UI elements in the Sign in page	3.Verify the below mentioned ui elements: a. email text box. b. password text box. c. sign in button
7	SignInPage_TC_003	Functional	Sign In page	Verify the user is able to login into the application by providing valid details	1. Enter the url and go 2.Click the sign in link in the navigation bar. 3.Enter valid details in the text boxes. 4. Verify the user is able to login.
8	DonorRegistrationPage_TC_001	Functional	Donor Registration Page	Verify the user is able to see the Donor registration page when the user clicks the donate link in navigation bar	1. Enter the url and go 2.Click the donate link in the navigation bar. 3.Verify the donor registration page is visible or not.
9	DonorRegistrationPage_TC_002	UI	Donor Registration Page	Verify the UI elements in the Donor Registration page	. Enter the url and go 2.Click the donate link in the navigation bar. 3.Verify the below mentioned ui elements: a. name text box b. email text box. c. blood group text box. d. contact number text box. e. city text box f. register as donor button
10	DonorRegistrationPage_TC_003	Functional	Donor Registration Page	Verify the user is able toregister as a donor by providing valid details	1. Enter the url and go 2.Click the donate link in the navigation bar. 3.Enter valid details in the text boxes. 4. Click the donate button. 4. Verify the user is able to register as a donor sucessfully



11	RequestPage_TC_001	Functional	Request Page	Verify the user is able to see the request page when the user clicks the request link in navigation bar	1. Enter the url and go 2.Click the request link in the navigation bar. 3.Verify the request page is visible or not.
12	RequestPage_TC_002	UI	Request Page	Verify the UI elements in the request page	. Enter the url and go 2.Click the request link in the navigation bar. 3.Verify the below mentioned ui elements: a. name text box b. email text box. c. blood group text box. d. contact number text box. e. city text box f. make a request button
13	RequestPage_TC_003	Functional	Request Page	Verify the user is able to make a request by providing valid details	1. Enter the url and go 2.Click the request link in the navigation bar. 3.Enter valid details in the text boxes. 4. Click the request button. 4. Verify the user is able to make a request sucessfully.
14	Notication_TC_001	Functional	Sign up page	Verify the user gets a email notification when they sign up	1. Enter the url and go 2. Go to the sign up page. 3. Enter the details and click sign up button 4. Verify if they get the email on successfull sign up
15	Notication_TC_002	Functional	Request Page	Verify the donor gets a email notification when they make a request	1. Enter the url and go 2. Go to the request page. 3. Enter the details and click make a request button 4. Verify if the donor gets the email on successfully maki request.

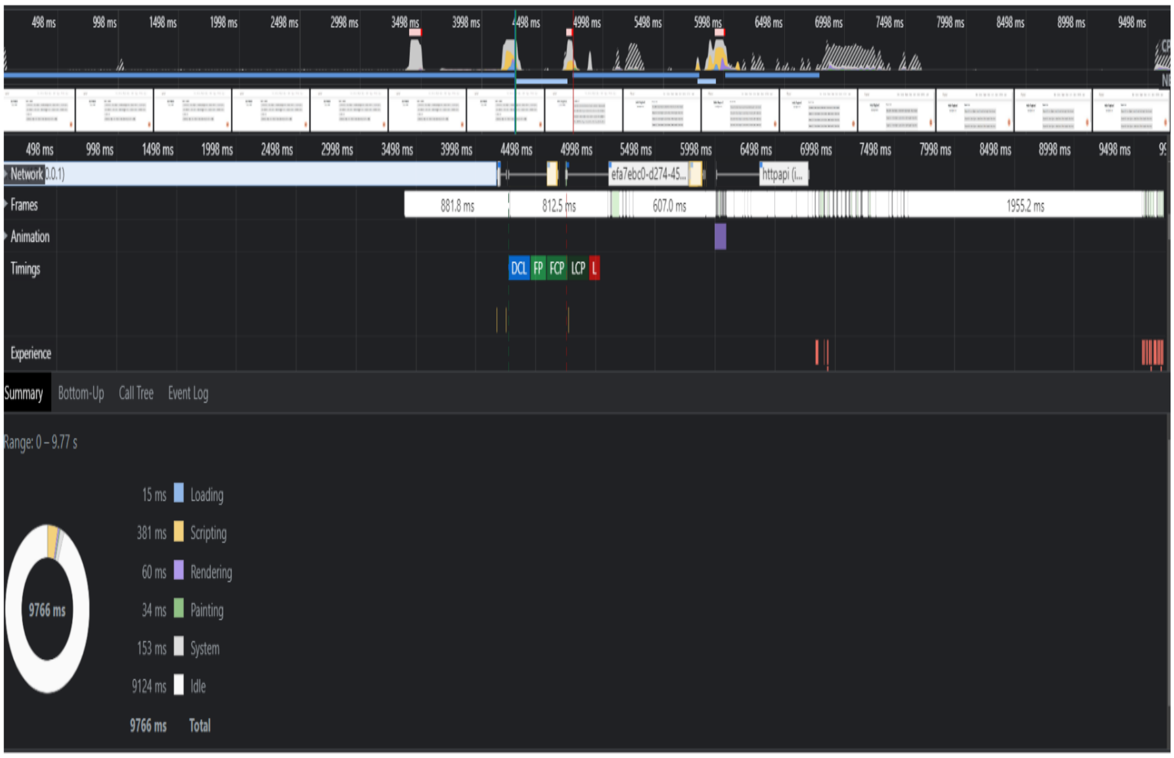


16	Notication_TC_003	Functional	Profile Page	Verify the donor and recipient gets a email notification when the donor accepts the request	1. Enter the url and go 2. Go to the profile page 3.Accept the pending request. 4. Verify if they get the email containing contact details
17	StatsPage_TC_001	Functional	Stats Page	Verify the user is able to see the stats page when the user clicks the stage page link in navigation bar	1. Enter the url and go 2.Click the stats page link in the navigation bar. 3.Verify the stats page is visible or not
18	Chatbot_TC_001	Functional	Home Page	Verify the user is able to interact with the chatbot	1. Enter the url and go 2.Click the chatbot icon in the home page 3.Verify the chatbot is working or not

RESULTS

9.1 PERFORMANCE METRICS

Web application performance metrics help determine certain aspects that impact the performance of an application. There are eight key metrics, including: User Satisfaction—also known as Apdex Scores, uses a mathematical formula in order to determine user satisfaction.



Range: 0 – 9.77 s

15 ms	🟦	Loading
381 ms	🟨	Scripting
60 ms	🟪	Rendering
34 ms	🟩	Painting
153 ms	⬜	System
9124 ms	⬜	Idle
9766 ms		Total

Fig 4.1. Performance Metrics

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- It is a user-friendly application.
- It will help people to find plasma easily.
- Simple User Interface
- It alleviates the burden of coordinator to manage Users and resources easily.
- Compared to all other mobile applications, it incorporates provisions for Plasma donation and Plasma Requesting.
- Attracts more, number of users as it is available in the form of Mobile application instead of What’s app group.
- Usage of this application will greatly reduce time in selecting the right donor.

DISADVANTAGES

- It cannot auto verify user genuineness.
- It requires an active internet connection.

CONCLUSION

Plasma is a liquid portion of blood; it is a mixture of water, proteins and salts. Antibodies are proteins made by the body in response to an infection. People fully rescued from COVID19 are encouraged to donate plasma, which can help to increase the lifespan of other patients because their plasma contains antigens which helps the affected person to recover faster. These immunoglobulins give your immune system a way to fight the virus when you are sick, so your plasma can be used to help others fight off illness. Individuals must fully resolve symptoms for at least 14 days prior are eligible to donate. Enhanced mobile application for plasma has been developed to help the administrator to attract more donors and recipients and make user management an easy task. This mobile application will attract more users as it is user friendly and greatly reduces scalability issues.Thus, we have successfully designed and developed the Android mobile application to ease the process of becoming a donor and recipient of PMB bank.

FUTURE SCOPE

- A chat widget to establish communication between a donor and recipient.
- To attract more user’s android application should also be developed in future.

APPENDIX

SOURCE CODE

register.html

```
{% extends "layout.html" %}
{% block title %}Register{% endblock %}

{% block head %}
{{ super() }}
{% endblock %}

<body>
  {% block content %}
  {{ super() }}

  <div class="row g-0 align-items-top">
    <div class="col-md-2">
    </div>

    <div class="col-md-4">
      <div class="card-body ">
```

```
        <form method="post">
          <!-- Username input -->
          <div class="form-outline mt-2">
            <input type="text" id="username" placeholder="Enter Your Name"
name="username" class="form-control"
              required />
          </div>

          <!-- Email input -->
          <div class="form-outline mt-2">
            <input type="email" id="email" placeholder="Enter Email" name="email"
class="form-control" required />
          </div>

          <!-- Mobile input -->
          <div class="form-outline mt-2">
            <input type="text" id="mobile" placeholder="Enter 10-digit mobile number"
name="mobile" class="form-control"
              required />
          </div>

          <!-- City input -->
          <div class="form-outline mt-2">
            <input type="city" id="city" placeholder="Enter Your City Name" name="city"
class="form-control" required />
          </div>

          <!-- Infect input -->
          <div class="form-outline mt-2">
            <select name="infect" id="infect" name="infect" class="form-control">
              <option value="uninfected" selected>Uninfected</option>
              <option value="infected">Infected</option>
            </select>
            <label class="form-label" for="infect">Select COVID infection status</label>
          </div>

          <!-- Blood input -->
          <div class="form-outline mt-2">
            <select name="blood" id="blood" name="blood" class="form-control">
              <option value="A Positive" selected>A Positive</option>
              <option value="B Positive">B Positive</option>
              <option value="AB Positive">AB Positive</option>
              <option value="O Negative">O Negative</option>
              <option value="A Negative">A Negative</option>
              <option value="B Negative">B Negative</option>
              <option value="AB Negative">AB Negative</option>
              <option value="O Positive">O Positive</option>
            </select>
            <label class="form-label" for="blood">Choose your blood group</label>
          </div>

          <!-- Password input -->
          <div class="form-outline mt-2">
            <input type="password" id="passw" placeholder="Enter Password" name="passw"
class="form-control" required />
          </div>
          <div class="form-outline mt-2">
            <!-- Submit button -->
            <button id="register" class="btn btn-primary btn-block">Submit</button>
          </div>
        </form>

      </div>
    </div>
    <div class="col-md-4 mt-4">
        {% if messages %}
        <div class="alert alert-info alert-dismissible fade show " role="alert">
          <span>{{ messages.message }}</span>
          <button type="button" class="close" data-dismiss="alert" aria-label="Close">
            <span aria-hidden="true">×</span>
          </button>
        </div>
        {% endif %}
    </div>
  </div>

  {% endblock %}
</body>
{% block js%}
{{ super() }}
<script>
  $(document).ready(function () {
    $('#register').on('click', function (event) {

      var postData = {
        'username': $('#username').val(),
        'email': $('#email').val(),
        'mobile': $('#mobile').val(),
        'city': $('#city').val(),
        'infect': $('#infect').val(),
        'blood': $('#blood').val(),
        'passw': $('#passw').val(),
      };
      $.ajax({
        beforeSend: function (xhrObj) {
          xhrObj.setRequestHeader("Content-Type", "application/json");
          xhrObj.setRequestHeader("Accept", "application/json");
        },
        type: "POST",
        url: "/register",
        dataType: "json",
        contentType: "application/json; charset=utf-8",
        data: JSON.stringify(postData),
      })
        .done(function (data) {
        });
      event.preventDefault();
    });
  });
</script>
{% endblock %}
```

home.html

```
{% extends "layout.html" %}
{% block title %}Home{% endblock %}

{% block head %}
{{ super() }}
{% endblock %}

<body>
  {% block content %}
  {{ super() }}
  <div class="row">
    <div class="col-md-12">
      <p>
      </p>
    </div>
  </div>
  <div class="row">
    <div class="col-md-1">
    </div>
    <div class="col-md-3">
      <div class="card">
        <div class="card-header">
          Learn All About Plasma
        </div>
        <div class="card-body">
          <h5 class="card-title">What is plasma in blood?</h5>
          <p class="card-text">
            Plasma is the liquid portion of blood. About 55% of our blood is plasma, and the
remaining 45% are red blood
            cells, white blood cells and platelets that are suspended in the plasma.

            Plasma is about 92% water. It also contains 7% vital proteins such as albumin,
gamma globulin and
            anti-hemophilic factor, and 1% mineral salts, sugars, fats, hormones and vitamins.
          </p>
          <p class="card-text">
            Plasma serves four important functions in our body:
          <ol>
            <li>Helps maintain blood pressure and volume.</li>
            <li>Supply critical proteins for blood clotting and immunity.</li>
            <li>Carries electrolytes such as sodium and potassium to our muscles.</li>
            <li>Helps to maintain a proper pH balance in the body, which supports cell
function.</li>
          </ol>


          </p>

        </div>

      </div>
```

```html
  </div>
  <div class="col-md-3">
    <div class="card">
      <div class="card-header">
        Plasma Donation
      </div>
      <div class="card-body">
        <h5 class="card-title">What is a plasma donation?</h5>
        <p class="card-text">
          In a plasma-only donation, the liquid portion of the donor’s blood is separated from
the cells. Blood is drawn from one arm and sent through a high-tech machine that collects
the plasma. The donor’s red blood cells and platelets are then returned to the donor along
with some saline. The process is safe and only takes a few minutes longer than donating
whole blood.
          Donated plasma is frozen within 24 hours of being donated to preserve its valuable
clotting factors. It can be stored for up to one year and thawed for transfusion to a patient
when needed. Red Cross donations are often used directly for hospital patient transfusions,
rather than pharmaceutical uses.
        </p>
      </div>
    </div>
  </div>
  <div class="col-md-3">
    <div class="card">
      <div class="card-header">
        Want to donate?
      </div>
      <div class="card-body">
        <h5 class="card-title">Who should donate plasma?<br></h5>
        <p class="card-text">
          The Red Cross urges people with type AB blood to consider a plasma donation. AB
is the only universal plasma and can be given to patients of any blood type. This means that
type AB plasma transfusions can be given immediately, without losing precious time
```

```
determining if the patient’s blood type is compatible. In emergency medicine, such as caring
for a major trauma or burn patient, time saved can mean the difference between life and
death.
          </p>
          <a href="/register" class="btn btn-primary">Register your name</a>
        </div>
      </div>
      <div class="col-md-2">
      </div>
    </div>

    {% endblock %}
</body>
{% block js%}
{{ super() }}
{% endblock %}
```

login.html

```
{% extends "layout.html" %}
{% block title %}Signin{% endblock %}

{% block head %}
{{ super() }}
{% endblock %}

<body>
  {% block content %}
  {{ super() }}

  <div class="row ">
    <div class="col-md-3 mt-4"></div>
    <div class="col-md-4 mt-4">
      <div class="card">
```

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

            <!-- Form submit to signin -->
            <form method="post">
              <!-- Username input -->
              <div class="form-outline mt-4">
                <input type="text" id="username" placeholder="username@example.com"
name="username" class="form-control" required />
              </div>

              <!-- Password input -->
              <div class="form-outline mt-4">
                <input type="password" id="password" placeholder="Password" name="password"
class="form-control" required />
              </div>

              <!-- Submit button -->
              <div class="form-outline mt-4">
                <button id="signin" class="btn btn-primary btn-block mb-4">Sign in</button>
              </div>
            </form>
          </div>
        </div>
      </div>
      <div class="col-md-3 mt-4">
        {% if messages %}
        <div class="alert alert-info alert-dismissible fade show " role="alert">
          <span>{{ messages.message }}</span>
          <button type="button" class="close" data-dismiss="alert" aria-label="Close">
            <span aria-hidden="true">×</span>
          </button>
        </div>
        <div class="col-md-2 mt-4"></div>
        {% endif %}
    </div>
      {% endblock %}
</body>
{% block js%}
{{ super() }}
<script>
    $(document).ready(function () {
      $('#signin').on('click', function (event) {

        var postData = {
          'username': $('#username').val(),
          'passw': $('#password').val(),
        };
        $.ajax({
          type: "POST",
          url: "/signin",
          dataType: "json",
          contentType: "application/json; charset=utf-8",
          data: JSON.stringify(postData),
        })
          .done(function (data) {
          });
        event.preventDefault();
      });
    });
</script>

{% endblock %}
```

statistics.html

```
{% extends "layout.html" %}
{% block title %}Statistics{% endblock %}

{% block head %}
{{ super() }}
{% endblock %}

<body>
    {% block content %}
    {{ super() }}

    <div class="row ml-2 mt-2">
        {% if data %}
        {% for d in data %}
        <div class="col-md-2 mt-2">
            <div class="card">
                <div class="card-body">
                    <center>
                        <h5 class="card-title">{{ d.group }}</h5>
                        <p class="card-text"><span class="badge badge-success">{{ d.count
}}</span></p>
                        <a href="/requested" class="btn btn-primary">Make Request</a>
                    </center>
                </div>
            </div>
        </div>
        <div class="col-md-1"></div>
        {% endfor %}
        {% endif %}
    </div>

    {% endblock %}
</body>
{% block js%}
```

```
{{ super() }}
{% endblock %}
```

request.html

```
{% extends "layout.html" %}
{% block title %}Signin{% endblock %}

{% block head %}
{{ super() }}
{% endblock %}

<body>
    {% block content %}
    {{ super() }}

    <div class="row ">
        <div class="col-md-3 mt-4"></div>
        <div class="col-md-4 mt-4">
            <div class="card">
                <div class="card-body ">

                    <!-- Form submit to request -->
                    <form method="post">
                        <!-- Blood input -->
                        <div class="form-outline mt-2">

                            <select name="blood" id="blood" name="blood" class="form-control">
                                <option value="A Positive" selected>A Positive</option>
                                <option value="B Positive">B Positive</option>
                                <option value="AB Positive">AB Positive</option>
                                <option value="O Negative">O Negative</option>
                                <option value="A Negative">A Negative</option>
```

```
                              <option value="B Negative">B Negative</option>
                              <option value="AB Negative">AB Negative</option>
                              <option value="O Positive">O Positive</option>
                          </select>
                          <label class="form-label" for="blood">Choose your blood group</label>
                      </div>
                      <div class="form-outline mt-2">
                          <textarea class="form-control" id="address" name="addess" rows="3"
placeholder="Enter your address" required ></textarea>
                      </div>

                      <!-- Submit button -->
                      <div class="form-outline mt-2">
                          <button id="request" class="btn btn-primary btn-block mb-2">Submit
request</button>
                      </div>
                  </div>
                  </form>
              </div>
          </div>
          <div class="col-md-4 mt-4">
              {% if messages %}
              <div class="alert alert-info alert-dismissible fade show " role="alert">
                <span>{{ messages.message }}</span>
                <button type="button" class="close" data-dismiss="alert" aria-label="Close">
                  <span aria-hidden="true">×</span>
                </button>
              </div>
              {% endif %}
      </div>
      {% endblock %}
  </body>
  {% block js%}
```

```
{{ super() }}
<script>
    $(document).ready(function () {
      $('#request').on('click', function (event) {

        var postData = {
          'blood': $('#blood').val(),
          'address': $('#address').val(),
        };
        alert(JSON.stringify(postData));
        $.ajax({
          type: "POST",
          url: "/requested",
          dataType: "json",
          contentType: "application/json; charset=utf-8",
          data: JSON.stringify(postData),
        })
          .done(function (data) {
          });
        event.preventDefault();
      });
    });
  </script>
{% endblock %}
```

index.html

```
{% extends "layout.html" %}
{% block title %}Index{% endblock %}
{% block head %}
    {{ super() }}
{% endblock %}
<body>
```

```
    {% block content %}
      {{ super() }}
    {% endblock %}
</body>
{% block js%}
{{ super() }}
{% endblock %}
```

layout.html

```
<!doctype html>
<html>

<head>
    {% block head %}
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <!-- Bootstrap CSS -->

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

    <title>{% block title %}{% endblock %} - My Webpage</title>
    {% endblock %}
</head>

<body>
    <div id="content">
        {% block content %}
        <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
            <img src="../static/img/donate.png" alt="..."/>&nbsp;&nbsp;<span class="bg-
primary text-white font-weight-bold">Plasma Donor
```

```
App&nbsp;&nbsp;&nbsp;&nbsp;</span>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">

                <ul class="navbar-nav mr-auto">
                    <li class="nav-item active">
                        <a class="nav-link" href="/home">&nbsp;Home&nbsp;<span class="text-
muted">&nbsp;|</span></a>
                    </li>
                    <li class="nav-item active">
                        <a class="nav-link" href="/aboutus">&nbsp;About Us&nbsp;<span
class="text-muted">&nbsp;|</span></a>
                    </li>
                        <div class="dropdown-menu" aria-labelledby="navbarDropdown">
                            <a class="dropdown-item" href="#">Action</a>
                            <a class="dropdown-item" href="#">Another action</a>
                            <div class="dropdown-divider"></div>
                            <a class="dropdown-item" href="#">Something else here</a>
                        </div>
                    </li>
                </ul>
                <div class="pull-right">
                    <ul class="navbar-nav mr-auto">
                        <li class="nav-item active">
                            <a class="nav-link" href="/signin">&nbsp;Sign In&nbsp;<span
class="text-muted">&nbsp;|</span></a>
                        </li>
                        <li class="nav-item active">
                            <a class="nav-link" href="/register">&nbsp;Register&nbsp;</a>
                        </li>
                    </ul>
                </div>
            </div>
        </nav>
```

```
 {% endblock %}
    </div>
</body>
{% block js %}
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
    integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5Kk
N"
    crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"
    integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
    crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"
    integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
    crossorigin="anonymous"></script>
{% endblock %}
</html>
```

GITHUB & PROJECT DEMO LINK

Github Link :

github.com/IBM-EPBL/IBM-Project-16661-1659619681

Project demo Link:

youtu.be/mWAri1y9r20