Project Report Documentation

Plasma Donor Application

Submitted by

PNT2022TMID27128

Anusha V – 310819205010

Aswini C – 310819205012

Bala Bharathi S – 310819205015

Abhirami K - 310819205003

Table of Contents

a. Sprint Planning & Estimation

b. Sprint Delivery Schedule

c. Reports from JIRA

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

    a. Feature 1

    b. Feature 2

8. **TESTING**

    a. Test Cases

    b. User Acceptance Testing

9. **RESULTS**

    a. Performance Metrics

10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

    Source Code

    GitHub & Project Demo Link

CHAPTER 1

INTRODUCTION

a. Project Overview

When a patient needs plasma, it can be challenging to get in contact with a donor within the patient's family and friends in a timely manner. It can also be challenging to get in contact with authorized donor centers. Due to a lack of awareness regarding plasma donation, there is a demand for plasma donors, making it challenging for the affected patients to locate donors. During the COVID-19 pandemic, the need for plasma increased and the donor rates decreased in order to provide an immunity boost for COVID-affected patients. The condition of the affected patient may suffer and perhaps result in death if the appropriate plasma donor cannot be found within a certain amount of time. Our plasma donor application enables patients with severe liver illnesses, blood clotting issues, and covid to quickly and easily locate the correct donor within the allotted period. In our application, users interested in donating plasma can sign up and hospitals can also register and request the plasma from the donors registered in our application, whenever patients need treatment. Hospitals can request by sending an email notification to the donors who are registered in the application. As a result, patients can locate the right donor within a given time frame.

b. Purpose

Because of a lack of awareness about plasma donation, there is a high demand for plasma donors, making it difficult for affected patients to find donors. During the COVID-19 pandemic, there was a greater need for plasma and a decrease in donor rates in order to provide an immunity boost to COVID-affected patients. So, If an appropriate plasma donor cannot be found within a certain amount of time, the affected patient's condition may worsen and possibly result in death and it is also difficult to contact a donor among the patient's family and friends in a timely manner. It can also be difficult to make contact with authorised donor centres.Therefore, main purpose of Our application enables patients with severe liver illnesses, blood clotting issues, and covid to quickly and easily locate the correct donor within the allotted period.

CHAPTER 2

LITERATURE SURVEY

a. Existing Problem

Plasma is required for the treatment of people who are affected by Covid-19. Due to this issue, the number of plasma donors decreased as the need for plasma increased. Because of a lack of awareness about plasma donation, there is a high demand for plasma donors, making it difficult for affected patients to find donors. During the COVID-19 pandemic, there was a greater need for plasma and a decrease in donor rates in order to provide an immunity boost to COVID-affected patients. Suitable plasma donor cannot be found within a certain amount of time, the affected patient's condition may deteriorate and possibly result in death.

b. References

We have referred from researchers who have previously researched on this topic and literature surveys are,

1. https://onlinelibrary.wiley.com/doi/full/10.1111/j.1423-0410.2010.01360.x

2. R. Kumar, R. Kumar and M. Tyagi, "Web Based Online Blood Donation System," 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), 2021, pp. 1630-1632, doi: 10.1109/ICAC3N53548.2021.9725558.

3. P. U. Pratyusha, K. Chaitanya, A. Saranam, K. Manideep and S. Kranthi, "Smart Intelligent Web based Online Blood Donation System," 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), 2021, pp. 1813-1819, doi: 10.1109/ICOSEC51865.2021.9591811.

4. M. Kaur et al., "A Web-based Blood Bank System for Managing Records of Donors and Receipts," 2022 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), 2022, pp. 459-464, doi: 10.1109/CISES54857.2022.9844389.

5. G. Dudani, Tanushree, K. Singh and A. Singh Chauhan, "A Systematic Review & Design of Web-Based Blood Management System," 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), 2021, pp. 1896-1900, doi: 10.1109/ICAC3N53548.2021.9725479.

6. R. S. Ali, T. F. Hafez, A. B. Ali and N. Abd-Alsabour, "Blood bag: A web application to manage all blood donation and transfusion processes," 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2017, pp. 2125-2130, doi: 10.1109/WiSPNET.2017.8300136.

7. E. H. Kaplan, "Evaluating plasma holds in the presence of multiple infections," in Mathematical Medicine and Biology: A Journal of the IMA, vol. 18, no. 3, pp. 215-224, Sept. 2001, doi: 10.1093/imammb/18.3.215.

8. M. Arif, S. Sreevas, K. Nafseer and R. Rahul, "Automated online Blood bank database," 2012 Annual IEEE India Conference (INDICON), 2012, pp. 012-017, doi: 10.1109/INDCON.2012.6420581.

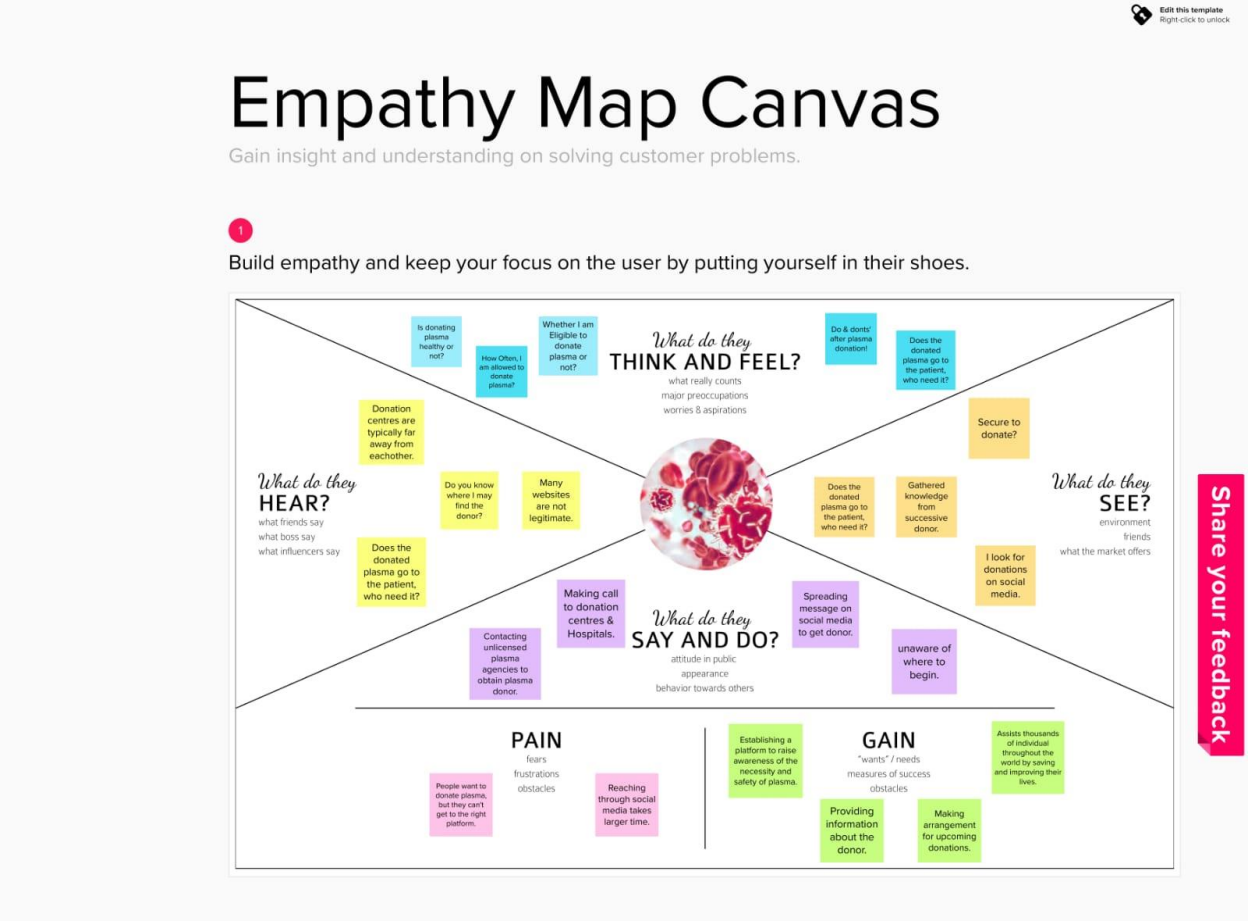
c. Problem Statement Definition

Patients with severe liver disorders and numerous clotting deficiencies are given plasma. When a patient needs plasma, it can be challenging to get in contact with a donor within the patient's family and friends in a timely manner. During the COVID-19 pandemic, the need for plasma increased and the donor rates decreased in order to provide an immunity boost for COVID-affected patients. It takes a long time for a patient to discover the proper donor, and it also takes time for the spreading about the need for plasma donors to disseminate on social media to a larger audience. As a result, patients cannot locate the right donor within a given time frame. The condition of the affected patient may suffer and perhaps result in death. Our application enables patients with severe liver illnesses, blood clotting issues, and covid to quickly and easily locate the correct donor within the allotted period.

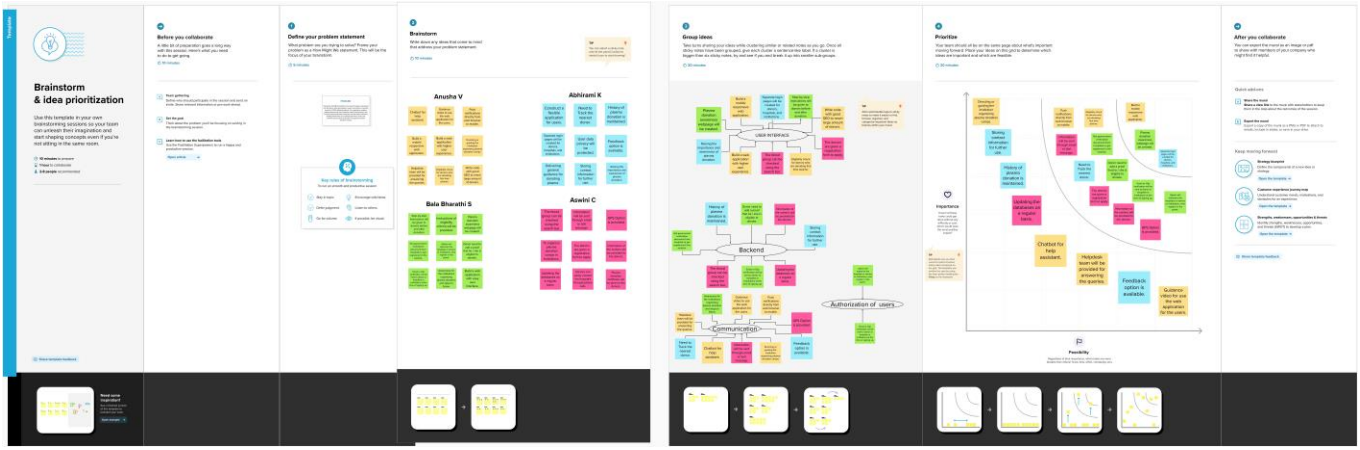
CHAPTER 3

IDEATION AND PROPOSED SOLUTION

a. Empathy Map Canvas



b. Ideation & Brainstorming



c. Proposed Solution

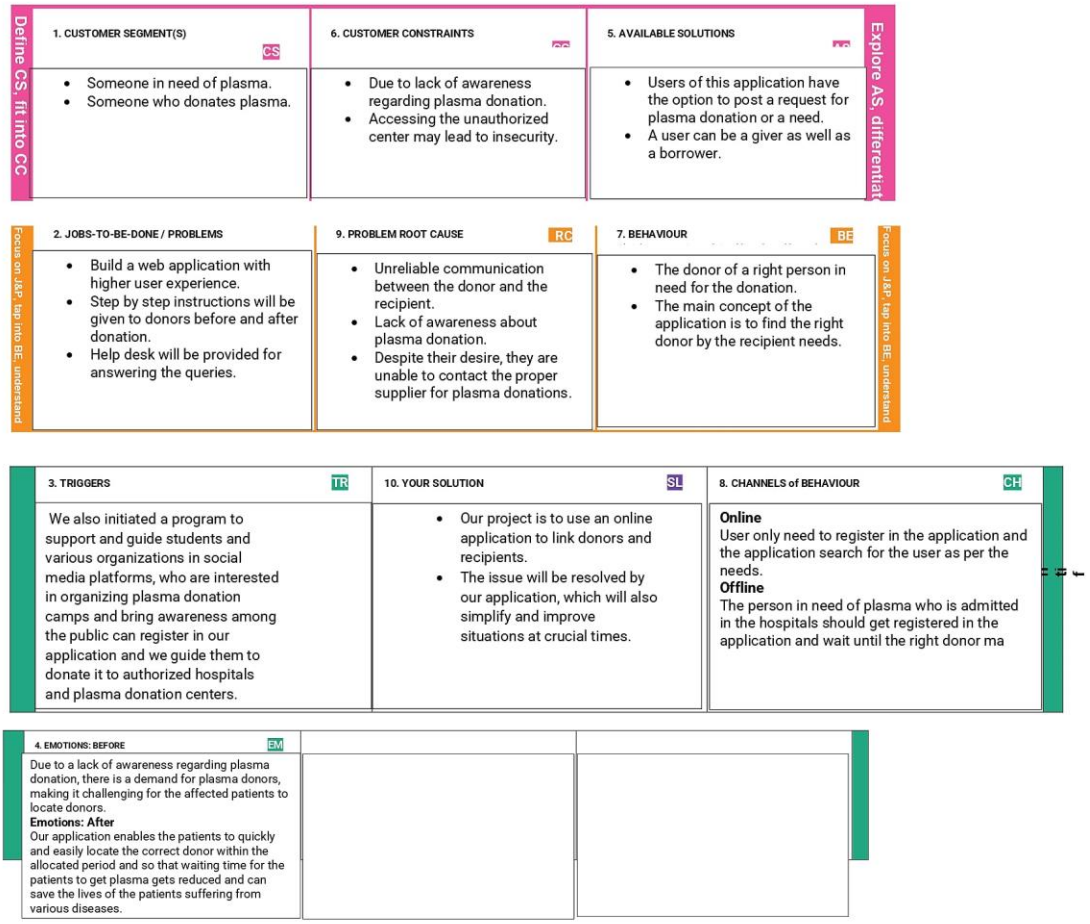
S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Due to a lack of awareness regarding plasma donation, there is a demand for plasma donors, making it challenging for the affected patients to locatedonors. During the COVID-19 pandemic, the need for plasma increased and the donor rates decreased inorder to provide an immunity boostfor COVID-affected patients. Our application enables the patients to quickly and easily locate the correct donor within the allocated period.

2.	Idea / Solution description	The issue will be resolved by our application, which will also simplify and improve situations at crucial times. We have an admin who will be managing hospital / plasma donorcenters details andplasma donors details and authorize them.Theusers registered in our application will be notifiedwhen there is a requirement of plasma inhospitals and guide the donor to the respective hospitals / plasma donorcenters. So, that waiting time for the patients to get plasmagets reducedandcan save the lives of the patients sufferingfrom various diseases. We also initiated a program to support andguide students andvarious organizations, who are interested inorganizing plasmadonation camps and bringingawareness among the publiccan register in ourapplication and we guidethem to donateit toauthorized hospitals and plasma donation centers.
3.	Novelty / Uniqueness	Our plasma donorapplication's goal is to raisepublic knowledge of the advantages of plasma donation and to encourage people to donateplasma to approved plasma donor centersor hospitals where it is required. Secondly, we have a function that allows universities that plan plasma donation campsto register for our application. We will provide them with direction and support so they may effectively plan camps and donate plasma to hospitals that need it for thepatient.

4.	Social Impact/ Customer Satisfaction	Our project intends to increase public knowledge of plasma donation, help patients find the correct donor within a certain time frame, and save the lives of those suffering from various diseases. 1. Guidance video to use the webapplication for the users. 2. Helpdesk team will be providedforanswering the queries. 3. Chatbot for help assistant. 4. Feedback option is available. 5. Awareness of the institutions organizingplasma donation and steps to follow. 6. Push notifications directly from the webbrowser to mobile.
5.	Business Model (Revenue Model)	A free app is available to plasma donors. It is easily accessible and open to everyone.Due to thechallenges in locating donors who fit a specific blood group, this tool enables users to register people who want to donate plasmaand retain their information . By notifying current donors, it would be beneficial to keep donor information on Database. The need for plasma increased significantly during the COVID-19 crisis, yet there aren't enough donors available. It is possible to provide a donation option.This application is non-profit, thus it depends on these donations to keep providing its service.
6.	Scalability of the Solution	The scalability of our project is to use an online application to link donors and patients. Users of this application have the option to post a request for plasma donation or a need and a user can bea giver as well as a borrower.

d. Problem Solution Fit

Project Title: Plasma Donor Application. **Project Design Phase-I - Solution Fit Template** **Team ID:** PNT2022TMID27128



CHAPTER 4
REQUIREMENT ANALYSIS

a. Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration via GoogleRegistration via Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP or Text Message
FR-3	User login	Login through browser directly byentering username and password Login throughGoogle
FR-4	User Interaction	An application form for registration is provided to the Donors. The data is entered into the form and saved in the database. If plasma is required, the donor will be contacted using the information provided.

b. NON FUNCTIONAL REQUIREMENTS

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

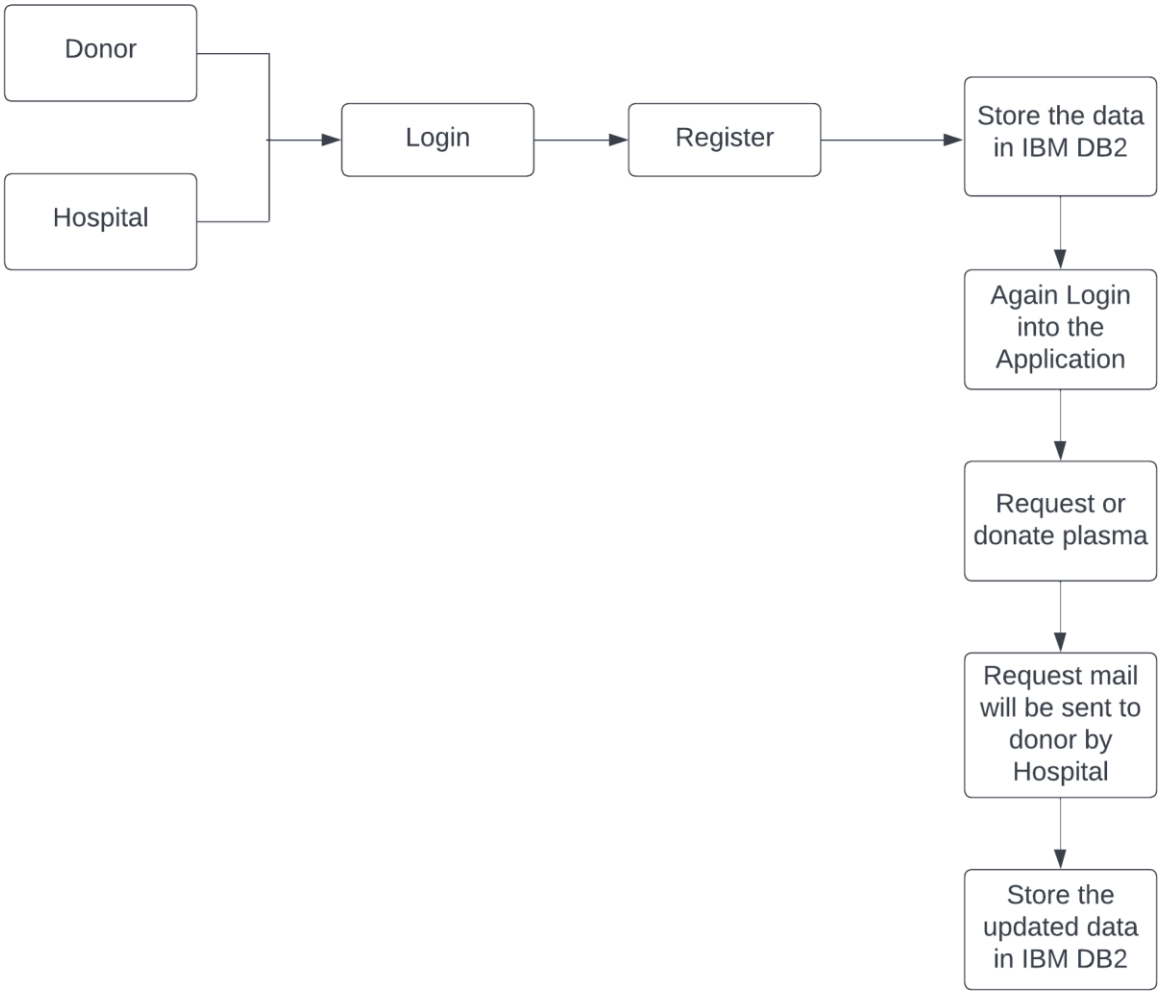
FR No.	Non-Functional Requirement	Description

NFR -1	**Usability**	Building a mobile responsive web application with higher user experience and the notification will be pushed directly from the web browser to the mobile and hence weprovide a user friendly environment.
NFR -2	**Security**	User data privacy will be protected against hackers.
NFR -3	**Reliability**	As the system provides the right tools for solvingthe problems it is made in sucha way that the systemis reliable in its operations and for securing the sensitive details.
NFR -4	**Performance**	The system is interactive and the delaysinvolved are less.When connecting to the server the delay is based on editing the distance of the 2 systems and the configuration between them so there is a high probability that there will be or not a successful connection in less than20 seconds for the sake of goodcommunication.
NFR -5	**Availability**	The system shouldbe available at all times,meaning the user can access it using an

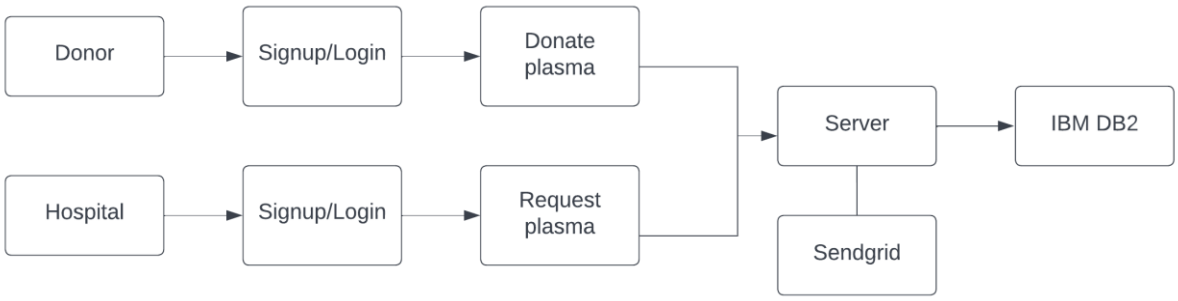
		application. In case of a hardware failure or database corruption a replacement page will be shown. Also in case of a hardware failure or database corruption, backups of the database should be retrieved from the application data folder and saved by the administrator.It means 24 x 7 availability.
NFR -6	Scalability	The Scalability of our project is to use an online application to link donors and patients. users of this application have the option to post an request for plasma donation or an need and users can be a giver as well as a borrower.

CHAPTER 5
PROJECT DESIGN

a. Data Flow Diagram



b. Solution & Technology Architecture



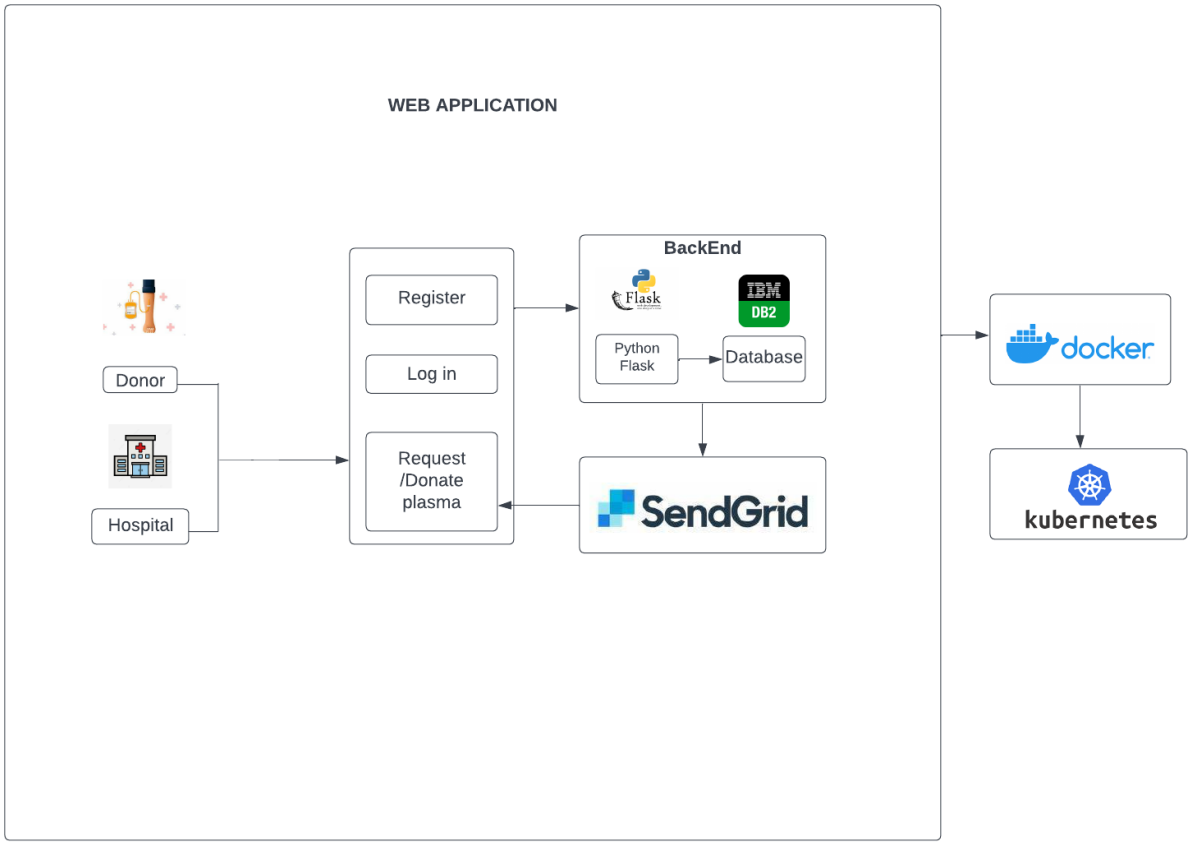


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	User interface design to make user interact withapplication.	HTML,CSS, Bootstrap

2.	Application Logic-1	Users login and register to the application.	Python Flask
3.	Application Logic-2	The hospital can request plasma from the donors registered in the application through email notification.	SendGrid
4.	Application Logic -3	To make the interactive conversion for users to clear doubts.	IBM Watson Assistant
5.	Database	To store and retrieve the data	IBM DB2 Database
6.	Infrastructure (Server/ Cloud)	Helping to orchestrate different types of containers and deploying them to clusters	Kubernetes

Table-2 : Application Characteristics

Sl. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	To build a lightweight application.	Flask
2.	To send Emails without a mailserver	To send email verification and configuration mail.	SendGrid
3.	Availability	Runs everywhere and userfriendly.	Docker
4.	Performance	Orchestrate containerized applications to run the cluster of hosts.	Kubernetes

c. User Stories

User Type	Functional Requirement(Epic)	User StoryNumber	User Story / Task	Acceptance criteria	Priority	Release
Donor	Registration	USN-1	As a donor, I can register into the application and get verified using OTP and email.	After verification, I can access my account and help the patients who are in need of plasma.	Low	Sprint-1

	Log In	USN-2	As a donor, I can login in to the application and help the patients by donating plasma.	After logging in, notification will be sent and according to that i can react and help the patients.	High	Sprint-1
Hospitals	Registration	USN-3	As a Hospital, I can register with all the details and log into the application, and request plasma.	Hospitals can register into the application and request for the plasma for the needed patients.	Low	Sprint-2
	Log In	USN-4	After registration, the Hospital Admin can logIn to the application .	As a Hospital Admin, I can log in and request plasma, whenever patients needed it.	High	Sprint-2
	Push notification	USN-5	When there is a need for plasma,Hospital Admin can push the notification.	As a Hospital Admin, Ican push notifications to all the donors registered through email.	High	Sprint-3
Bot	Help the users by using a bot.	USN-6	As a bot, I will provide interactive communication with the user and providethe information they need.	I can access my account/dashboard.	High	Sprint-4

CHAPTER 6
PROJECT PLANNING AND SCHEDULING

a. Sprint Planning And Estimation

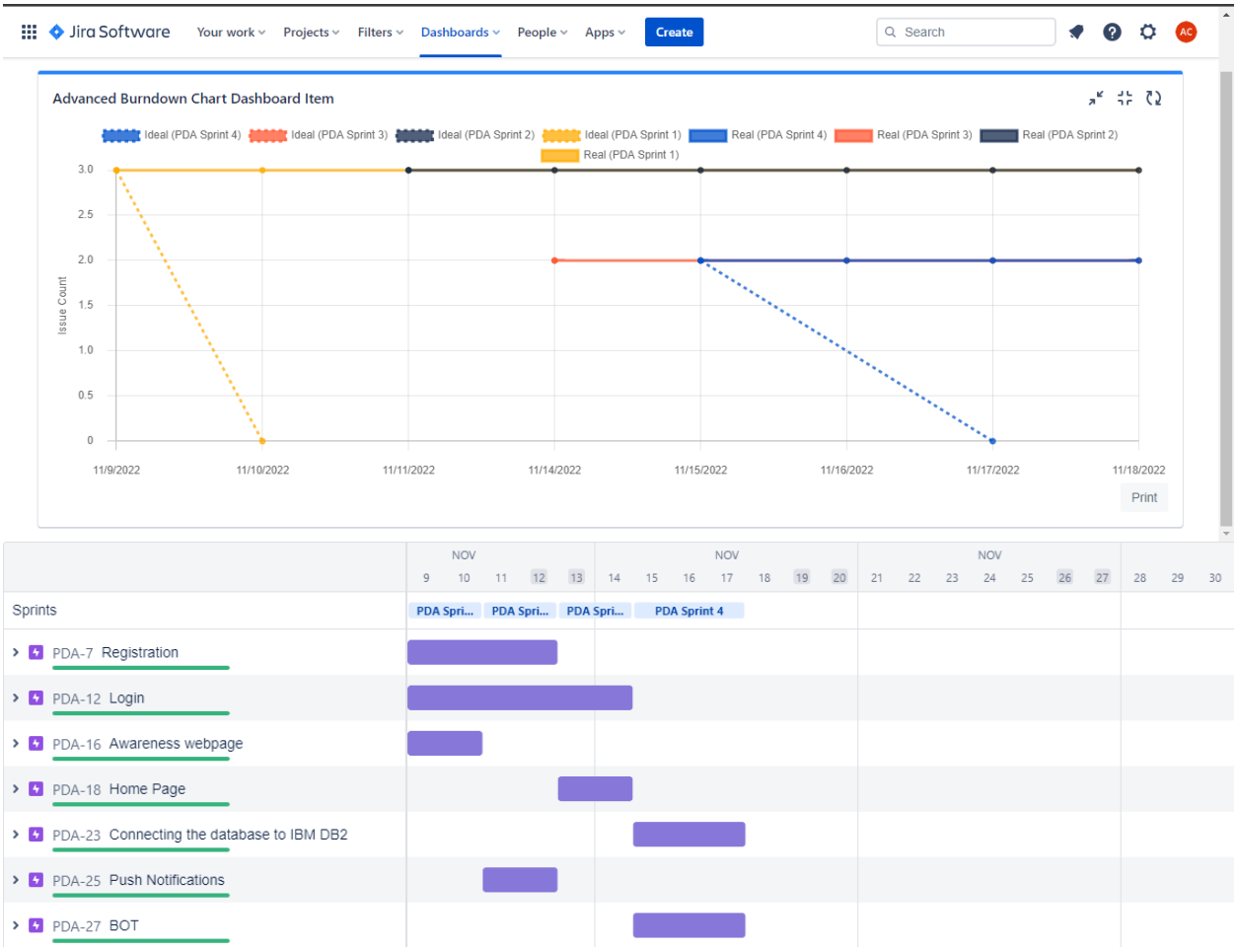
Sprint-3	Home Page	PDA-7	Home page for hospitals, they can request for plasma when patients needs.	4	High	Anusha V
Sprint-4	Connecting the database to IBM DB2	PDA-8	Admin will maintain the donor, hospitals and institutions details and connect frontend with backend to database.	5	High	Abhirami K
Sprint-2	Push Notifications	PDA-9	When there is a need of plasma, Hospital Admin can push the notification through email.	4	High	Aswini C
Sprint-4	BOT	PDA-10	As a BOT, I will provide interactive communication with the user and provide the information they need.	5	Medium	Bala Bharathi S

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	PDA-1	As a Donor, I can register into the application and get verified using OTP and email.	4	Low	Aswini C
Sprint-2		PDA-2	As a Hospital, I can register with all the proofs asked and verification mail and OTP will be sent to verify it.	3	High	
Sprint-1	Login	PDA-3	As a Donor, I can Login into the application and help the patients by donating plasma.	4	High	Anusha V
Sprint-2		PDA-4	After registration, Hospital Admin can Login into the application.	2	High	
Sprint-1	Awareness webpage	PDA-5	Create Awareness page for users to get aware of plasma donation.	3	High	Bala Bharathi S
Sprint - 3	Home Page	PDA-6	Home page for donors, I can donate plasma, wheneverhospitals needs plasma.	4	High	Anusha V

b. 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	20	2 Days	09 Nov 2022	10 Oct 2022	20	11 Oct 2022
Sprint-2	20	2 Days	11 Nov 2022	12 Nov 2022	20	13 Oct 2022
Sprint-3	20	2 Days	13 Nov 2022	14 Nov 2022	20	15 Oct 2022
Sprint-4	20	3 Days	15 Nov 2022	17 Nov 2022	20	18 Nov 2022

c. Reports from JIRA



CHAPTER 7
CODING & SOLUTIONING

a. Feature 1

SendGrid

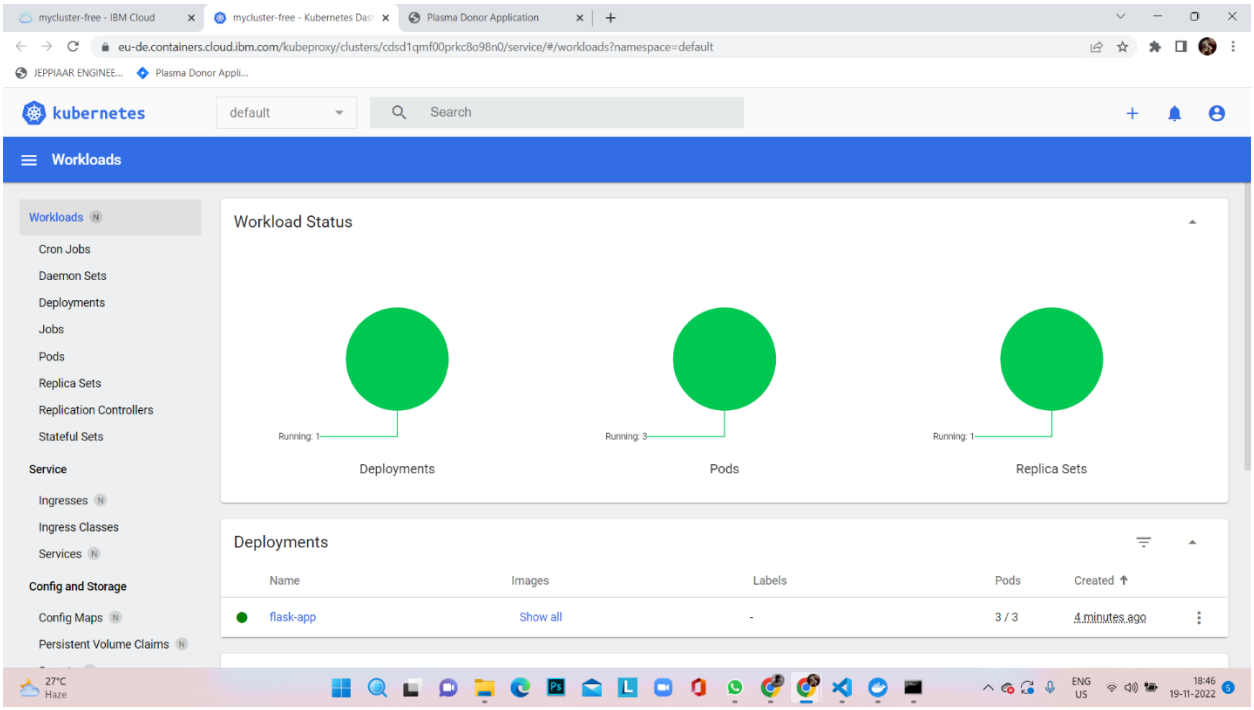
Sendgrid Service integrates in minutes with our email API and trust your emails reach the inbox.

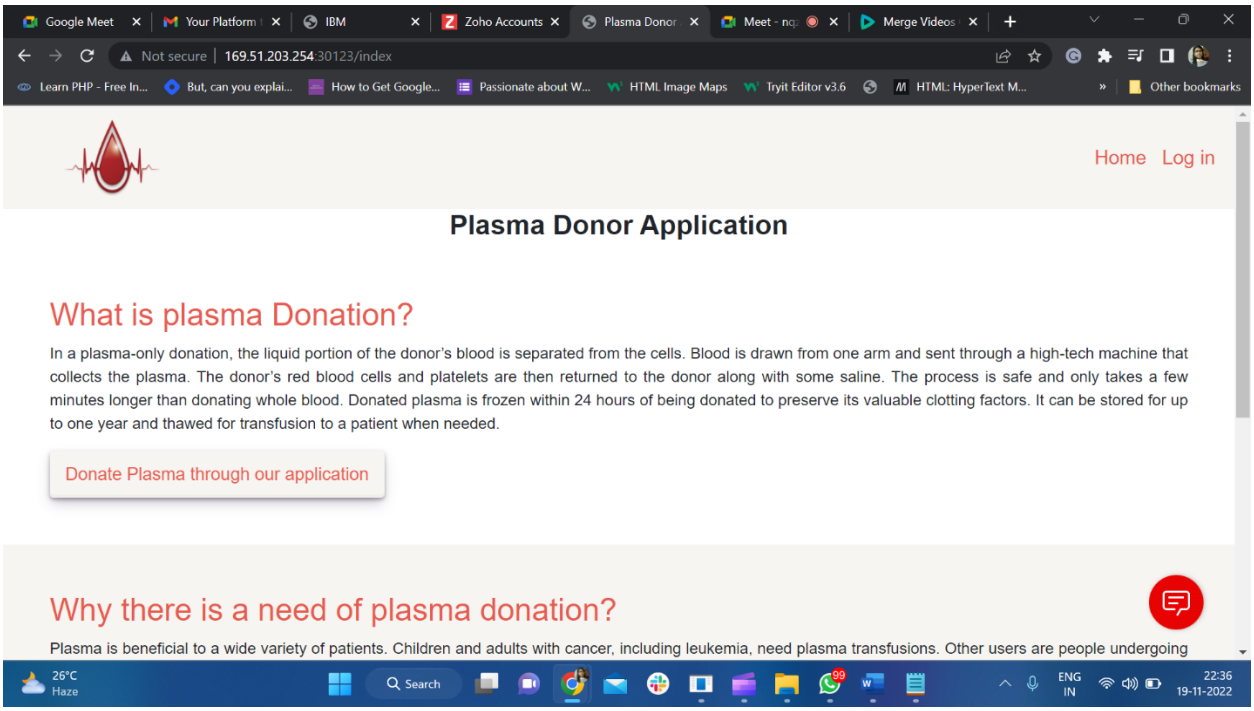
Sendgrid Integration

```
# sendgrid integration
#donor registration starts----------------------------------
def mailtest_registration(to_email):
    sg = sendgrid.SendGridAPIClient(api_key='')
    from_email = Email("itaswinic@gmail.com")
    subject = "Registered Successfull as a DONOR!"
    content = Content("text/plain", "You have successfully registered as donor. Please
Login using your Username and Password to donate Plasma.")
    mail = Mail(from_email, to_email, subject, content)
    response = sg.client.mail.send.post(request_body=mail.get())
    print(response.status_code)
    print(response.body)
    print(response.headers)

#donor registration ends------------------------------------

#hospital registration starts-------------------------------------
def mailtest_hosp_registration(to_email):
    sg = sendgrid.SendGridAPIClient(api_key= ' ' )
    from_email = Email("itaswinic@gmail.com")
    subject = "Registered Successfull as a Hospital!"
    content = Content("text/plain", "You have successfully registered as recipient. Please
Login using your Username and Password to request Plasma.")
    mail = Mail(from_email, to_email, subject, content)
    response = sg.client.mail.send.post(request_body=mail.get())
    print(response.status_code)
    print(response.body)
    print(response.headers)

#hospital registration ends---------------------------------

#request plasma confirmation mail starts--------------------------------
def mailtest_request_plasma(to_email):
```

```
    sg = sendgrid.SendGridAPIClient(api_key=' ')
    from_email = Email("itaswinic@gmail.com")
    subject = "Successfully registered to request plasma!"
    content = Content("text/plain", "You have successfully registered to request plasma.
our admin will connect you shortly with donor details")
    mail = Mail(from_email, to_email, subject, content)
    response = sg.client.mail.send.post(request_body=mail.get())
    print(response.status_code)
    print(response.body)
    print(response.headers)
```

b. Features 2




Google Meet
Your Platform
IBM
Zoho Accounts
Plasma Donor
Merge Videos
Not secure
169.51.203.254:30123/index
Learn PHP - Free In...
But, can you explai...
How to Get Google...
Passionate about W...
HTML Image Maps
Tryit Editor v3.6
HTML: HyperText M...
Other bookmarks
Home
Log in
Plasma Donor Application
What is plasma Donation?
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.
Donate Plasma through our application
Why there is a need of plasma donation?
Plasma is beneficial to a wide variety of patients. Children and adults with cancer, including leukemia, need plasma transfusions. Other users are people undergoing
26°C
Haze
Search
ENG
IN
22:36
19-11-2022

CHAPTER 8

TESTING

a.Test Cases

Test Case id	Test Scenario	Test Data	Expected Result	Actual Result	Status
TC_001	There is a Log in button in the navigation bar in Awareness page.	Click on the Log In Button.	Need to navigate to Log-In as page	Navigated to Log In as page.	Pass
TC_002	There is a "donate plasma through our application button" in awareness page.	Click on that button.	Need to navigate to Log-in-as page.	Navigated to Log-In-as page.	Pass
TC_003	Log-in-as page contains two buttons such as donor and hospital. after clicking the donor button.	Click on the donor button.	Need to navigate to log-in page for donor.	Navigated to Log-in page of donor.	Pass
TC_004	Log-In page contains hospital button.	Click on the hospital button.	Need to navigate to hospital Log-In page.	Navigated to hospital log-in page.	Pass
TC_005	If donor not signed-up already in the application. at Bottom there is a sign-up link for the donors.	Click on the sign-up link.	Need to navigate to donor sign-up page.	Navigated to donor sign-up page.	Pass
TC_006	Donor will enter all the details asked in the input field such as name, date of birth, age, blood group, address, city, state, email id and password for the application.	Enter all the details.	All the entered details need to be stored in the IBM DB2 database and then it needs to be navigated to donor log-in page.	Successfully all the entered data is stored in the IBM DB2 and then it navigated to donor log-in page.	Pass
TC_007	Donor log-in to the application using their sign-up credentials such as username and password.	Enter the username and password in their respective fields.	It will check the data in the IBM DB2 with the donor entered during log-in details and then navigate	Successfully logged in to the donor homepage.	Pass

			to donor homepage.		
TC_008	There is a sign out button in the navigation bar of the donor page.	Click on the sign-out button in the navigation bar.	after clicking on that it will sign out from the website and return to awareness page.	Successfully signed-out from the website and it returned back to awareness page.	Pass
TC_009	If hospital not signed-up already in the application. at Bottom there is a sign-up link for the hospitals.	Click on the sign-up link.	Need to navigate to hospital sign-up page.	Navigated to hospital sign-up page.	Pass
TC_010	Hospital will enter all the details asked in the input field such as hospital name, date of hospital established, hospital address, city, state, email id and password for the application.	Enter all the details.	All the entered details need to be stored in the IBM DB2 database and then it needs to be navigated to hospital log-in page.	Successfully all the entered data is stored in the IBM DB2 and then it navigated to hospital log-in page.	Pass
TC_011	Hospital log-in to the application using their sign-up credentials such as username and password.	Enter the username and password in their respective fields.	It will check the data in the IBM DB2 with the hospital entered during log-in details and then navigate to hospital confirmation page.	Successfully logged in to the hospital confirmation page.	Pass
TC_012	In confirmation page, there will be button to click and it navigate to hospital homepage.	Click on the request button.	Need to navigate to hospital homepage.	Navigated to hospital homepage.	Pass
TC_013	In hospital homepage, the list of donors registered will be displayed in the table format.	Display the donors list.	Display the donors list in the table format.	Successfully displayed the donors list from the IBM DB2.	Pass
TC_014	if need of plasma, hospital needs to click the request button to fill the request	Fill the request plasma	Need to navigate to	Navigate to request plasma page	Pass

	plasma form and it will navigate to request plasma form.	form with the respective input fields.	request plasma form.		
TC_015	Enter the details asked in request plasma form and then store the data in the IBM DB2 Database.	Enter the details into the respective fields.	Need to enter the details in the form and store in the database.	Successfully filled the form and stored in the database.	Pass
TC_016	There is a return to "home page" button in the confirmation page.	Click on the return to homepage button.	Need to navigate back to hospital home page.	Navigated to the hospitals home page.	Pass
TC_017	There is a sign out button in the navigation bar of the hospital page.	Click on the sign-out button in the navigation bar.	after clicking on that it will sign out from the website and return to awareness page.	Successfully signed-out from the website and it returned back to awareness page.	Pass

b.User Acceptance Testing

The test coverageand open issues of the Plasma Donor Application projectat the time ofthe release to User Acceptance Testing (UAT).

Defect Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
Flask	3	2	0	0	5
Cloud account creation	1	1	1	0	3
Connecting with Db2	4	3	1	0	8
Sendgrid	2	3	0	1	6
Docker	2	1	0	0	3

Totals	12	10	2	1	25

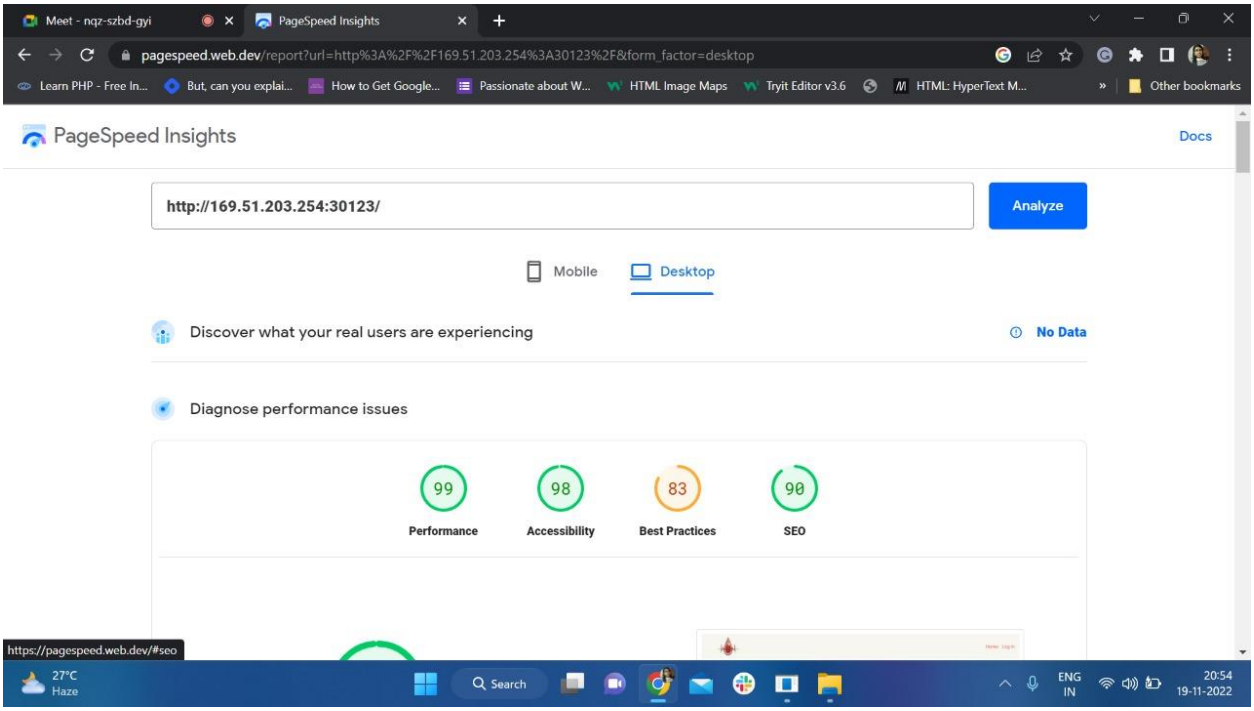
Test Case Analysis

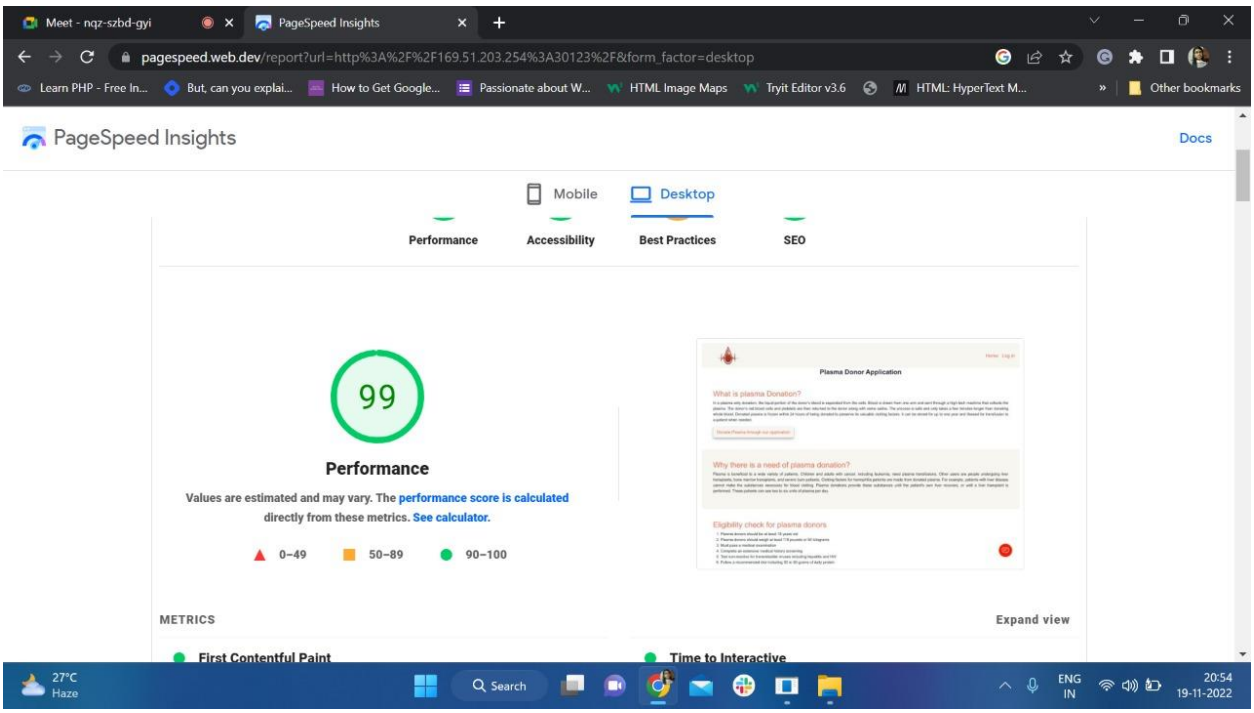
Section	Total Cases	Not Tested	Fail	Pass
Awareness Page	2	0	0	2
Login as Page	2	0	0	2
Login Donor Page	2	0	0	2
Signup Donor Page	2	0	0	2
Login Hospital Page	2	0	0	2
Signup Hospital Page	4	0	0	4
Request Plasma Page	3	0	0	3
Chatbot	2	0	0	2

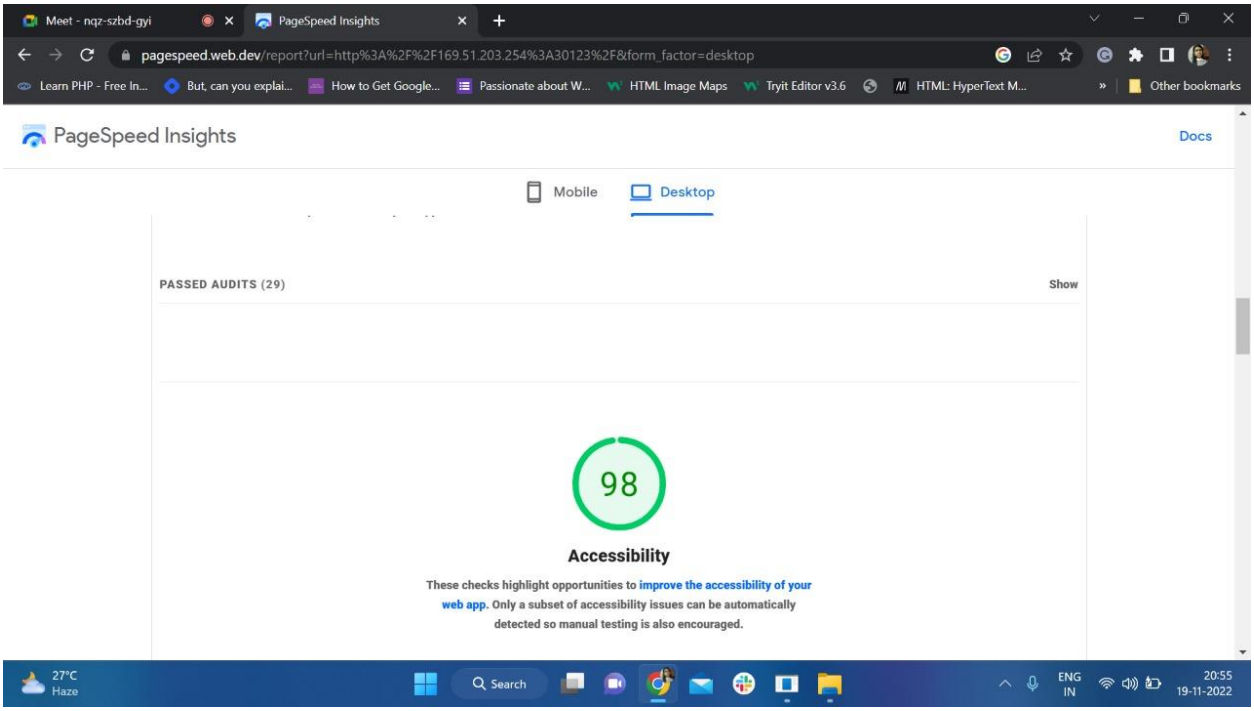
CHAPTER 9

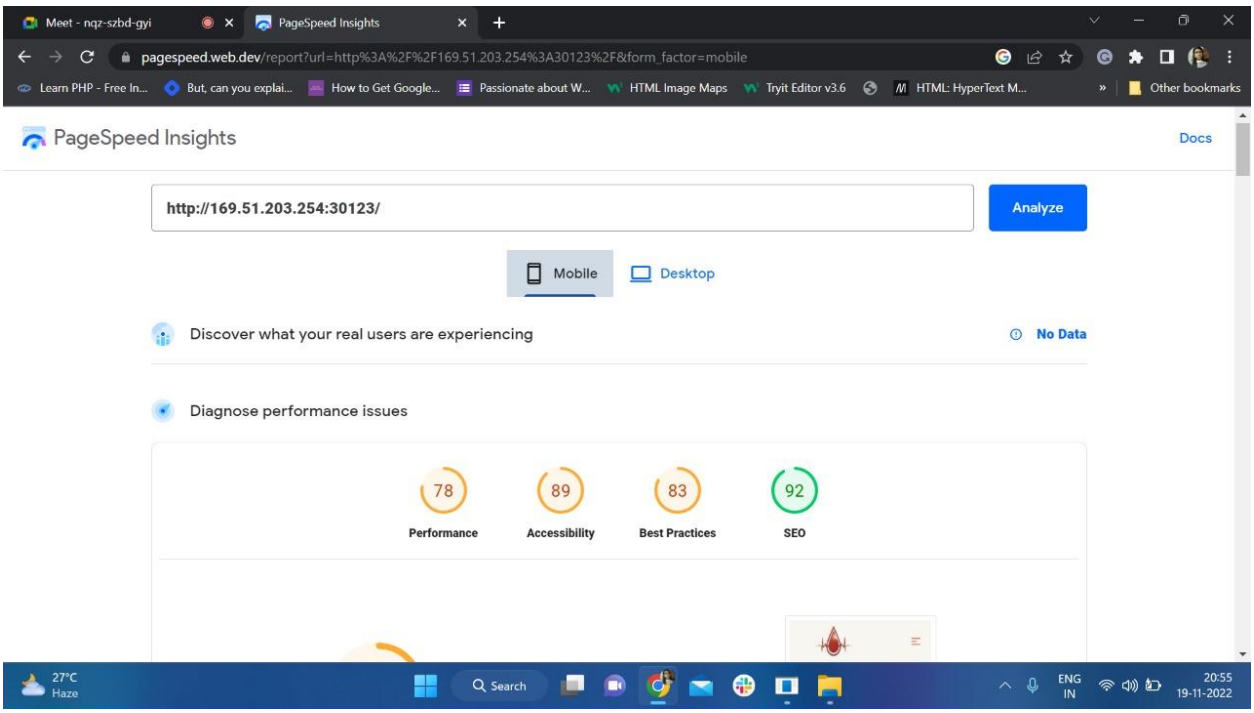
RESULTS

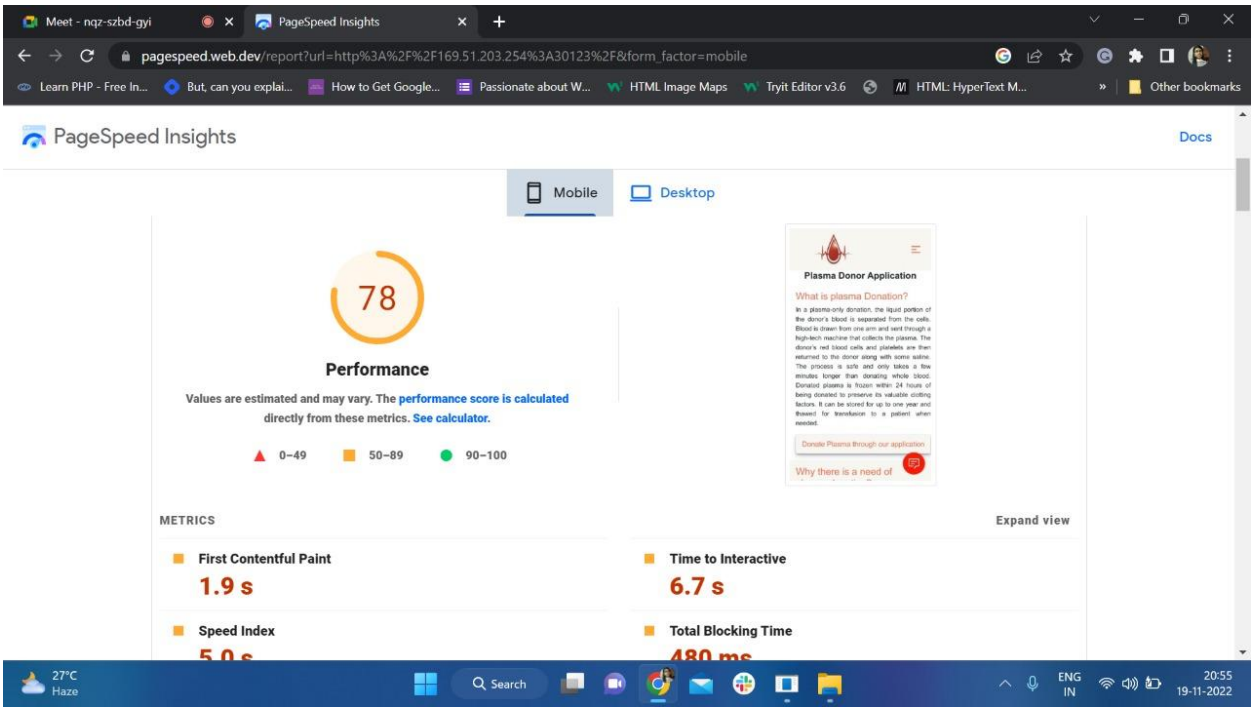
a.Performance Metrics

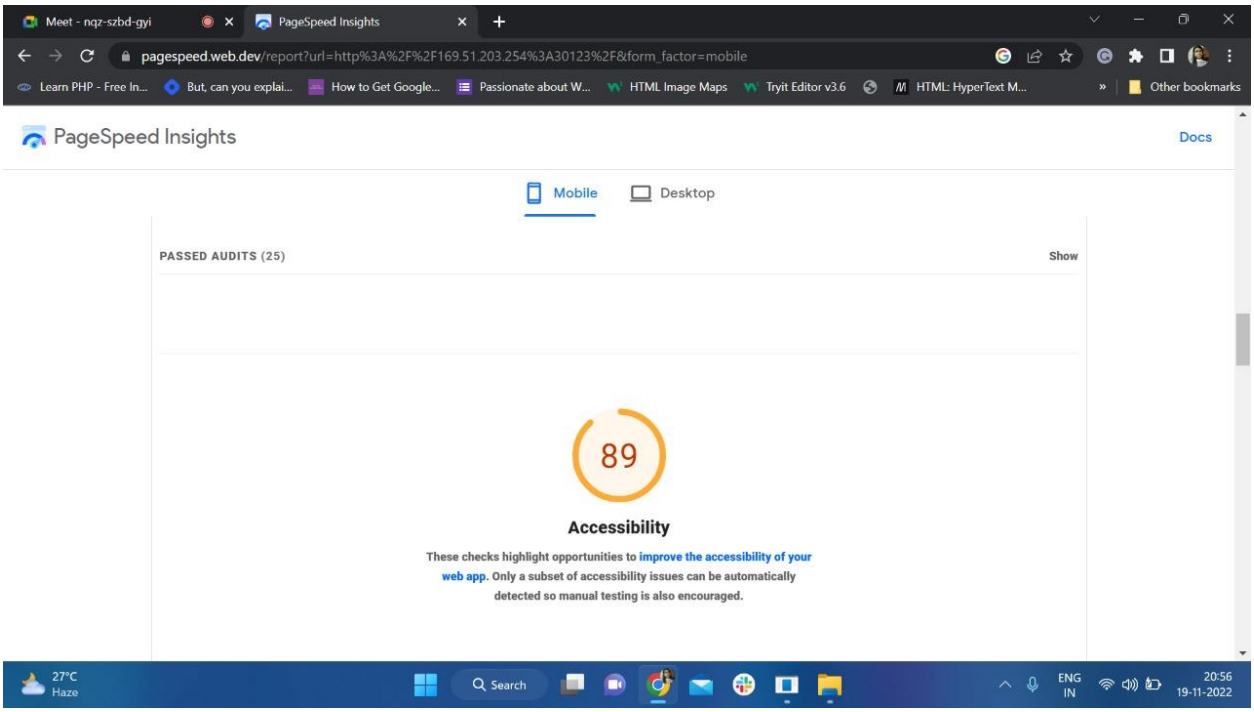





Meet - nqz-szbd-gyi
PageSpeed Insights
pagespeed.web.dev/report?url=http%3A%2F%2F169.51.203.254%3A30123%2F&form_factor=desktop
Learn PHP - Free In...
But, can you explai...
How to Get Google...
Passionate about W...
HTML Image Maps
Tryit Editor v3.6
HTML: HyperText M...
Other bookmarks
PageSpeed Insights
Docs
Mobile
Desktop
PASSED AUDITS (29)
Show
98
Accessibility
These checks highlight opportunities to improve the accessibility of your web app. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.
27°C
Haze
Search
ENG
IN
20:55
19-11-2022


Meet - nqz-szbd-gyi
PageSpeed Insights
pagespeed.web.dev/report?url=http%3A%2F%2F169.51.203.254%3A30123%2F&form_factor=mobile
Learn PHP - Free In...
But, can you explai...
How to Get Google...
Passionate about W...
HTML Image Maps
Tryit Editor v3.6
HTML: HyperText M...
Other bookmarks
PageSpeed Insights
Docs
http://169.51.203.254:30123/
Analyze
Mobile
Desktop
Discover what your real users are experiencing
No Data
Diagnose performance issues
78
Performance
89
Accessibility
83
Best Practices
92
SEO
27°C
Haze
Search
ENG
IN
20:55
19-11-2022


Meet - nqz-szbd-gyi
PageSpeed Insights
pagespeed.web.dev/report?url=http%3A%2F%2F169.51.203.254%3A30123%2F&form_factor=mobile
Learn PHP - Free In...
But, can you explai...
How to Get Google...
Passionate about W...
HTML Image Maps
Tryit Editor v3.6
HTML: HyperText M...
Other bookmarks
PageSpeed Insights
Docs
Mobile
Desktop
78
Performance
Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.
0–49
50–89
90–100
Plasma Donor Application
What is plasma Donation?
Why there is a need of
METRICS
Expand view
First Contentful Paint
1.9 s
Time to Interactive
6.7 s
Speed Index
Total Blocking Time
480 ms
27°C
Haze
Search
ENG
IN
20:55
19-11-2022


Meet - nqz-szbd-gyi
PageSpeed Insights
pagespeed.web.dev/report?url=http%3A%2F%2F169.51.203.254%3A30123%2F&form_factor=mobile
Learn PHP - Free In...
But, can you explai...
How to Get Google...
Passionate about W...
HTML Image Maps
Tryit Editor v3.6
HTML: HyperText M...
Other bookmarks
PageSpeed Insights
Docs
Mobile
Desktop
PASSED AUDITS (25)
Show
89
Accessibility
These checks highlight opportunities to improve the accessibility of your web app. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.
27°C
Haze
Search
ENG
IN
20:56
19-11-2022

CHAPTER 10

ADVANTAGES & DISADVANTAGES

a.Advantages

- Establishing a platform to raise awareness of the necessity and safety of plasma.
- Making arrangements for the upcoming donors.
- Assists thousands of individuals throughout the world by saving and improving their lives.

b.Disadvantages

- Reaching through social media takes larger time.
- People wants to donate plasma, but they can’t get to the right platform.

CHAPTER 11

CONCLUSION

As a result, plasma is the frequently overlooked component of blood. The function of the body depends on platelets, red blood cells, and white blood cells. But plasma also has a significant impact. The blood components are transported throughout the body by this fluid. The majority of your blood is plasma. It accounts for more than half (about 55%) of the entire material. Plasma is a pale yellow liquid that forms when the blood is separated from the rest of it. Salts, enzymes, and water are carried by plasma. Plasma's primary function is to transport proteins, hormones, and nutrients to the body's many organs. Additionally, plasma is where cells dump their waste. The plasma then aids in the body's elimination of this waste. The plasma then aids in the body's elimination of this waste. All components of the blood are also transported through your circulatory system by blood plasma. White blood cells, red blood cells, are returned to the donor via saline after plasma donation. Our project intends to increase public knowledge of plasma donation, help patients find the correct donor within a certain time frame, and save the lives of those suffering from various diseases.

CHAPTER 12

FUTURE SCOPE

The Plasma Donation App will benefit both donors and patients who require plasma. You might use it to look for Plasma Donors in your area who belong to a particular Blood Group. Those who have recovered completely from COVID-19 have antibodies that can battle the virus in their plasma. The initiative to develop a web application for plasma donation might guarantee that by donating plasma and saving the world.

CHAPTER 13

APPENDIX

a.Source Code

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Plasma Donor Application</title>
   <link rel="stylesheet" href="../static/style.css">
   <!--bootstrap files starts-->
   <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
   <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></
script>
   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
  <!--bootstrap code ends-->
  <style>


   .custom-toggler.navbar-toggler {

          border:transparent;

       }



       .custom-toggler .navbar-toggler-icon {
          background-image: url('https://anushav.s3.jp-tok.cloud-object-
storage.appdomain.cloud/menu_icon.svg');

       }
       .navbar-nav {
```

```
            margin-left: auto;
        }
        .nav-item{
          text-decoration: none;
          color:#E85A4F;


        }
        .nav-item:hover{
          text-decoration: none;

          color:black;

        }
        .nav-item:visited{
          text-decoration: none;


        }
  </style>

     <script>
  window.watsonAssistantChatOptions = {
    integrationID: "b5e67d39-9862-4f40-9dc5-020482814f3a", // The ID of this
integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "c084d960-061d-4eb3-bad3-3745a05f6c2a", // The ID of your
service instance.
    onLoad: function(instance) { instance.render(); }
  };
  setTimeout(function(){
    const t=document.createElement('script');
    t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
    document.head.appendChild(t);
  });

  </script>
</head>
<body>
</html>
<!--navbar section starts-->
<div class="m-0" style=" background-color: rgba(234, 231, 220,40%);">
```

```
  <nav class="navbar navbar-expand-lg">
    <div class="container-fluid">
      <a href="#" class="navbar-brand" style="margin-left:50px;">
        <img src="https://anushav.s3.jp-tok.cloud-object-
storage.appdomain.cloud/Plasma_logo.png" width="100" height="80" alt="CoolBrand">

      </a>
      <button type="button" class="navbar-toggler custom-toggler" data-bs-
toggle="collapse" data-bs-target="#navbarCollapse">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarCollapse">
        <div class="navbar-nav" style="font-weight: 400;">
          <a href="{{url_for('hello_world')}}" class="nav-item nav-link active" style="font-
size: 20px;">Home</a>
          <a href="{{url_for('login')}}" class="nav-item nav-link" style="font-size:
20px;">Log in</a>

        </div>
      </div>
    </div>
  </nav>
</div>
<!--navbar section ends-->
<h3 style="text-align: center; font-weight: 600;">Plasma Donor Application</h3>
<!--What is plasma donation section starts-->

<div class="what_is_plasma_donation_section">
  <div class="what_is_plasma_donation">
    <h2>What is plasma Donation?</h2>
    <p>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.
    </p>
    <button class="button-30"><a href="{{url_for('login')}}">Donate Plasma through our
application</a></button>
  </div>
</div>
```

```
<!--What is plasma donation section ends-->

<!--why there is a need of plasma section starts-->

<div class="what_is_plasma_donation_section" style=" background-color: rgba(234, 231, 220,40%);">
   <div class="what_is_plasma_donation">
      <h2>Why there is a need of plasma donation?</h2>
      <p>Plasma is beneficial to a wide variety of patients. Children and adults with cancer, including leukemia, need plasma transfusions. Other users are people undergoing liver transplants, bone marrow transplants, and severe burn patients. Clotting factors for hemophilia patients are made from donated plasma.

         For example, patients with liver disease cannot make the substances necessary for blood clotting. Plasma donations provide these substances until the patient's own liver recovers, or until a liver transplant is performed. These patients can use two to six units of plasma per day.
      </p>
   </div>
</div>

<!--why there is a need of plasma section ends-->

<!--eligibility check for plasma donors section starts-->

<div class="what_is_plasma_donation_section">
   <div class="what_is_plasma_donation" style="text-align: left;">
      <h2>Eligibility check for plasma donors</h2>

         <ol>
            <li>Plasma donors should be at least 18 years old</li>
            <li> Plasma donors should weigh at least 110 pounds or 50 kilograms</li>
            <li>Must pass a medical examination</li>
            <li>Complete an extensive medical history screening</li>
            <li>Test non-reactive for transmissible viruses including hepatitis and HIV</li>
            <li>Follow a recommended diet including 50 to 80 grams of daily protein</li>
         </ol>

   </div>
</div>

<!--eligibility check for plasma donors section starts-->
```

```
</body>
</html>
```

Log_in.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Log In</title>
   <link rel="stylesheet" href="../static/log_in.css">
    <!--bootstrap files starts-->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></
script>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
   <!--bootstrap code ends-->
</head>
<body>
   <!--log in section starts-->
<div class="log_in_section">
   <div class="log_in">
      <button class="btn btn-danger text-center" style="text-align: center;"><a
href="{{url_for('hello_world')}}" style="text-decoration: none; color:white">Return to
Home Page</a></button>
      <h2>Log in as</h2>
      <div class="log_in_buttons">

         <button class="button-30"><a
href="{{url_for('login_donor')}}">Donor</a></button>
         <button class="button-30"><a
href="{{url_for('login_hospital')}}">Hospital</a></button>
      </div>
   </div>
```

```
</div>

   <!--log in section ends-->
</body>
</html>
```

Login_donor.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Donor</title>
   <link rel="stylesheet" href="../static/login.css">
    <!--bootstrap files starts-->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></
script>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
   <!--bootstrap code ends-->
</head>
<body>

   <div class="log_in_section">

      <button class="btn btn-danger text-center" style="text-align: center;"><a
href="{{url_for('hello_world')}}" style="text-decoration: none; color:white">Return to
Home Page</a></button>
      <div class="log_in_container">
         <div class="msg">{{ msg }}</div>
         <h2>Log In</h2>
         <h3>Donor</h3>
         <div class="log_in_form">
            <form action="/login_donor" method="POST">
               <label>Username</label>
               <input type="text" name="donor_name" placeholder="Enter username">
               <label>Password</label>
```

```
                <input type="password" name="donor_password" placeholder="Enter
Password">
                <button class="button-38">Log in</button>

               <p>Don't have an account?<a href="{{url_for('signup_donor')}}">Sign
up</a></p>
             </form>



         </div>
      </div>
   </div>
</body>
</html>
```

Login_hospital

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Hospital</title>
   <link rel="stylesheet" href="../static/login.css">
    <!--bootstrap files starts-->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></
script>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
   <!--bootstrap code ends-->
</head>
<body>

   <div class="log_in_section">
      <button class="btn btn-danger text-center" style="text-align: center;"><a
href="{{url_for('hello_world')}}" style="text-decoration: none; color:white">Return to
Home Page</a></button>
      <div class="log_in_container">
```

```
        <div class="msg">{{ msg }}</div>
        <h2>Log In</h2>
        <h3>Hospital</h3>
        <div class="log_in_form">
          <form action="/login_hospital" method="POST">
            <label>Username</label>
            <input type="text" name="hospital_name" placeholder="Enter username">
            <label>Password</label>
            <input type="password" name="hospital_password" placeholder="Enter
Password">
            <button class="button-38">Log in</button>

            <p>Don't have an account?<a href="{{url_for('signup_hospital')}}">Sign
up</a></p>
          </form>



        </div>
      </div>
    </div>
</body>
</html>
```

Request_plasma.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hospital</title>
    <link rel="stylesheet" href="../static/login.css">
</head>
<body>
    <div class="log_in_section">
      <div class="log_in_container">
        <h2>Donor Request form</h2>
        <h3>(After filling the request form, we will connect you with the right donor)</h3>
        <div class="log_in_form">
          <form action="/request_plasma" method="POST">
            <label>Hospital Name</label>
```

```
            <input type="text" name="hospital_name" placeholder="Enter your Hospital
name">

            <label>Hospital Phone Number</label>
            <input type="text" name="hospital_contact_number" placeholder="Enter
Hospital Phone Number">

            <label>Patient Name</label>
            <input type="text" name="patient_name" placeholder="Enter the patient
Name">

            <label>Patient Age</label>
            <input type="text" name="patient_age" placeholder="Enter the patient Age">

            <label>Patient Blood Group</label>
            <input type="text" name="patient_blood_group" placeholder="Enter the
patient Blood Group">


            <label>Cause of requesting plasma</label>
            <input type="text" name="cause_of_plasma" placeholder="Cause of
requesting plasma">

            <label>Hospital Address</label>
            <input type="text" name="hospital_address" placeholder="Enter your Hospital
address">

            <label>Hospital City</label>
            <input type="text" name="hospital_city" placeholder="Enter your Hospital
city">

            <label> Hospital State</label>
            <input type="text" name="hospital_state" placeholder="Enter your Hospital
state">

            <label>Hospital Email</label>
            <input type="email" name="hospital_email" placeholder="Enter Hospital
email id">


            <input type="submit" value="Request" class="btn btn-success">

        </form>
```

```
        </div>
      </div>
   </div>
</body>
</html>
```

Signup_donor.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Donor</title>
   <link rel="stylesheet" href="../static/login.css">
    <!--bootstrap files starts-->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></
script>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
   <!--bootstrap code ends-->
</head>
<body>
   <div class="log_in_section">
      <button class="btn btn-danger text-center" style="text-align: center;margin-
bottom:10px;"><a href="{{url_for('hello_world')}}" style="text-decoration: none;
color:white">Return to Home Page</a></button>
      <div class="log_in_container">
         <h2>Sign Up</h2>
         <h3>Donor</h3>
         <div class="log_in_form">
            <form action="/donor_form" method="POST">
               <label>Name</label>
               <input type="text" name="donor_name" placeholder="Enter your name">

               <label>Date of Birth</label>
```

```
            <input type="date" name="date_of_birth" placeholder="Enter your date of
birth">

            <label>Age</label>
            <input type="number" name="age" placeholder="Enter your age">

            <label>Gender</label>
            <input list="Gender" name="donor_gender" placeholder="Enter your gender">
            <datalist id="Gender">
               <option value="Male">
               <option value="Female">
            </datalist>

            <label>Blood Group</label>
            <input type="text" name="donor_blood_group" placeholder="Enter your blood
group">

            <label>Address</label>
            <input type="text" name="donor_address" placeholder="Enter your address">

            <label>City</label>
            <input type="text" name="donor_city" placeholder="Enter your city">

            <label>State</label>
            <input type="text" name="donor_state" placeholder="Enter your state">


            <label>Email</label>
            <input type="email" name="donor_email" placeholder="Enter email id">

            <label>Password</label>
            <input type="password" name="donor_password" placeholder="Enter
Password">


        <input type="submit" value="signup" class="btn btn-success">

          <p>Already have an account?<a href="{{url_for('login_donor')}}">Log
in</a></p>
        </form>



      </div>
```

```
      </div>
   </div>
</body>
</html>
```

Signup_hospital.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Hospital</title>
   <link rel="stylesheet" href="../static/login.css">
   <!--bootstrap files starts-->
   <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
   <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></
script>
   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
  <!--bootstrap code ends-->
</head>
<body>
   <div class="log_in_section">
      <button class="btn btn-danger text-center" style="text-align: center;margin-
bottom:10px;"><a href="{{url_for('hello_world')}}" style="text-decoration: none;
color:white">Return to Home Page</a></button>
      <div class="log_in_container">
         <h2>Sign Up</h2>
         <h3>Hospital</h3>
         <div class="log_in_form">
            <form action="/signup_hospital" method="POST">
               <label>Hospital Name</label>
               <input type="text" name="hospital_name" placeholder="Enter your Hospital
name">

               <label>Hospital Established Date</label>
               <input type="date" name="hospital_date" placeholder="Enter your Hospital
Established Date">
```

```
            <label>Hospital Phone Number</label>
            <input type="text" name="hospital_contact_number" placeholder="Enter
Hospital Phone Number">

            <label>Hospital Address</label>
            <input type="text" name="hospital_address" placeholder="Enter your Hospital
address">

            <label>Hospital City</label>
            <input type="text" name="hospital_city" placeholder="Enter your Hospital
city">

            <label> Hospital State</label>
            <input type="text" name="hospital_state" placeholder="Enter your Hospital
state">


            <label>Hospital Email</label>
            <input type="email" name="hospital_email" placeholder="Enter Hospital
email id">



            <label>Password</label>
            <input type="password" name="hospital_password" placeholder="Enter
Password">


            <input type="submit" value="signup" class="btn btn-success">

           <p>Already have an account?<a href="{{url_for('login_hospital')}}">Log
in</a></p>
        </form>



      </div>
    </div>
  </div>
</body>
</html>
```

Success.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Successfully requested plasma</title>
    <!--bootstrap files starts-->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></
script>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
   <!--bootstrap code ends-->
   <style>
      .button{
        display: grid;
        grid-template-columns: 1fr;
        align-items: center;
        justify-items: center;
      }
   </style>

</head>
<body>
   <h2 style="text-align: center;">Plasma Donor Application</h2>
   <h4 style="text-align: center; color:#E85A4F;">{{ msg }}</h4>

   <h5 style="margin-top: 50px; text-align: center;">Please, Click on this below button to
navigate to your dashboard! </h3>
   <div class="button">
   <button class="btn btn-danger"><a href="{{url_for('donorlist')}}" style="text-decoration:
none; color: white;"> Return to dashboard</a></button>
   </div>
</body>
</html>
```

Donors.html

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Logged In Successfully</title>
    <!--bootstrap files starts-->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></
script>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
   <!--bootstrap code ends-->
<style>
   .button{
     display: grid;
     grid-template-columns: 1fr;
     align-items: center;
     justify-items: center;
   }
</style>

</head>
<body>
   <h2 style="text-align: center;">Plasma Donor Application</h2>

<h4 style="text-align: center;">{{ msg }}</h4>
<h3 class="text-danger" style="margin-top: 50px; text-align: center;">Welcome
{{session["hospital_name"]}}!!</h3>
<h5 style="margin-top: 50px; text-align: center;">Please, Click on this below button to
navigate to your dashborad to request plasma! </h5>
<div class="button">
<button class="btn btn-danger"><a href="{{url_for('donorlist')}}" style="text-decoration:
none; color: white;"> Request</a></button>
</div>
</body>
</html>
```

Homepage_donor.html

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Home page - Donor</title>
   <!--bootstrap files starts-->
   <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
   <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></
script>
   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
  <!--bootstrap code ends-->
  <style>


   .custom-toggler.navbar-toggler {

          border:transparent;

       }



       .custom-toggler .navbar-toggler-icon {
          background-image: url('https://anushav.s3.jp-tok.cloud-object-
storage.appdomain.cloud/menu_icon.svg');

       }
       .navbar-nav {
          margin-left: auto;
       }
       .nav-item{
         text-decoration: none;
         color:#E85A4F;


       }
       .nav-item:hover{
         text-decoration: none;
```

```
          color:black;

        }
        .nav-item:visited{
          text-decoration: none;


        }
  </style>
</head>
<body>

<!--navbar section starts-->
<div class="m-0" style=" background-color: rgba(234, 231, 220,40%);">
   <nav class="navbar navbar-expand-lg">
      <div class="container-fluid">
         <a href="#" class="navbar-brand" style="margin-left:50px;">
            <img src="https://anushav.s3.jp-tok.cloud-object-
storage.appdomain.cloud/Plasma_logo.png" width="100" height="80" alt="CoolBrand">

         </a>
         <button type="button" class="navbar-toggler custom-toggler" data-bs-
toggle="collapse" data-bs-target="#navbarCollapse">
            <span class="navbar-toggler-icon"></span>
         </button>
         <div class="collapse navbar-collapse" id="navbarCollapse">
            <div class="navbar-nav" style="font-weight: 400;">


               <a href="{{url_for('index')}}" class="nav-item nav-link" style="font-size:
20px;">Sign Out</a>
            </div>
         </div>
      </div>
   </nav>
</div>
<!--navbar section ends-->


<h2 style="text-align: center;">Plasma Donor Application</h2>

<h4 style="text-align: center;">{{ msg }}</h4>
<h3 class="text-danger" style="margin-top: 50px; text-align: center;">Welcome
{{session["donor_name"]}}!!</h3>
```

```
<h5 style="margin-top: 50px; text-align: center;">You will be getting notified by hospitals, When patients in need of plasma through mail!!! </h5>

</body>

</html>
```

Homepage_hospital

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home page - Donor</title>
    <!--bootstrap files starts-->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></
script>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
   <!--bootstrap code ends-->
   <style>


    .custom-toggler.navbar-toggler {

            border:transparent;

        }



        .custom-toggler .navbar-toggler-icon {
            background-image: url('https://anushav.s3.jp-tok.cloud-object-
storage.appdomain.cloud/menu_icon.svg');

        }
        .navbar-nav {
            margin-left: auto;
```

```
        }
        .nav-item{
          text-decoration: none;
          color:#E85A4F;


        }
        .nav-item:hover{
          text-decoration: none;

          color:black;

        }
        .nav-item:visited{
          text-decoration: none;


        }
  </style>
</head>
<body>

<!--navbar section starts-->
<div class="m-0" style=" background-color: rgba(234, 231, 220,40%);">
    <nav class="navbar navbar-expand-lg">
        <div class="container-fluid">
            <a href="#" class="navbar-brand" style="margin-left:50px;">
                <img src="https://anushav.s3.jp-tok.cloud-object-
storage.appdomain.cloud/Plasma_logo.png" width="100" height="80" alt="CoolBrand">

            </a>
            <button type="button" class="navbar-toggler custom-toggler" data-bs-
toggle="collapse" data-bs-target="#navbarCollapse">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarCollapse">
                <div class="navbar-nav" style="font-weight: 400;">


                    <a href="{{url_for('index')}}" class="nav-item nav-link" style="font-size:
20px;">Sign Out</a>
                </div>
            </div>
        </div>
```

```
    </nav>
</div>
<!--navbar section ends-->


<h2 style="text-align: center;">Plasma Donor Application</h2>

<h4 style="text-align: center;">{{ msg }}</h4>
<h3 class="text-danger" style="margin-top: 50px; text-align: center;">Welcome
{{session["donor_name"]}}!!</h3>
<h5 style="margin-top: 50px; text-align: center;">You will be getting notified by hospitals,
When patients in need of plasma through mail!!! </h5>

</body>
</html>
```

App.py

```
from flask import Flask,render_template,url_for,session,redirect,request
import ibm_db,re,sendgrid
from sendgrid.helpers.mail import *
app = Flask(__name__)
app.secret_key="1123"
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=3883e7e4-18f5-4afe-be8c-
fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31498;SECU
RITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=btb13702;PWD=J90CGo
4TOKPWkMFb","","")
# index page starts----------------------------
@app.route("/")
def hello_world():
    return render_template("index.html")

@app.route("/log_in")
def login():
    return render_template("log_in.html")
#index page ends-------------------------------

#donor login starts-----------------------------------

@app.route("/login_donor",methods=['GET','POST'])
def login_donor():
    global userid
    msg = ''
    if request.method == 'POST' :
```

```
        donor_name = request.form['donor_name']
        donor_password = request.form['donor_password']
        sql = "SELECT * FROM donors_details WHERE donor_name =? AND
donor_password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,donor_name)
        ibm_db.bind_param(stmt,2,donor_password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print (account)
        if account:
            session['loggedin'] = True
            session['id'] = account['DONOR_NAME']
            userid=  account['DONOR_NAME']
            session['donor_name'] = account['DONOR_NAME']
            msg = 'Logged in successfully as donor!'

            return render_template('homepage_donor.html', msg = msg)
        else:
            msg = 'Incorrect email / password !'
    return render_template("login_donor.html")

#donor login ends----------------------------------------


#sign up donor starts----------------------------------

@app.route("/signup_donor")
def signup_donor():
    return render_template("signup_donor.html")

@app.route("/donor_form", methods=['GET','POST'])
def donor_form():
    msg = ''
    if request.method == 'POST' :
        donor_name = request.form['donor_name']
        date_of_birth = request.form['date_of_birth']
        age = request.form['age']
        donor_gender = request.form['donor_gender']
        donor_blood_group = request.form['donor_blood_group']
        donor_address = request.form['donor_address']
        donor_city = request.form['donor_city']
        donor_state = request.form['donor_state']
        donor_email = request.form['donor_email']
```

```
        donor_password = request.form['donor_password']
        sql = "SELECT * FROM donors_details WHERE donor_name =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,donor_name)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg = 'Account already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', donor_email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', donor_name):
            msg = 'name must contain only characters and numbers !'
        else:
            mailtest_registration(donor_email)
            insert_sql = "INSERT INTO donors_details VALUES (?, ?, ?, ?,?,?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, donor_name)
            ibm_db.bind_param(prep_stmt, 2, date_of_birth)
            ibm_db.bind_param(prep_stmt, 3, age)
            ibm_db.bind_param(prep_stmt, 4, donor_gender)
            ibm_db.bind_param(prep_stmt, 5, donor_blood_group)
            ibm_db.bind_param(prep_stmt,6, donor_address)
            ibm_db.bind_param(prep_stmt, 7, donor_city)
            ibm_db.bind_param(prep_stmt, 8, donor_state)
            ibm_db.bind_param(prep_stmt, 9, donor_email)
            ibm_db.bind_param(prep_stmt, 10, donor_password)

            ibm_db.execute(prep_stmt)
            msg = 'You have successfully registered as donor!'

        return render_template('login_donor.html', msg = msg)

#sign up donor ends----------------------------------------

# login hospital----------------------------------------

@app.route("/login_hospital", methods=['GET','POST'] )
def login_hospital():
    msg = ''
    if request.method == 'POST' :
        hospital_name = request.form['hospital_name']
        hospital_password = request.form['hospital_password']
```

```
        sql = "SELECT * FROM hospitals_details WHERE hospital_name =? AND
hospital_password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,hospital_name)
        ibm_db.bind_param(stmt,2,hospital_password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print (account)
        if account:
            session['loggedin'] = True
            session['id'] = account['HOSPITAL_NAME']
            userid=  account['HOSPITAL_NAME']
            session['hospital_name'] = account['HOSPITAL_NAME']
            msg = 'Logged in successfully as recipient!'
            return render_template('donors.html', msg = msg)
        else:
            msg = 'Incorrect email / password !'
    return render_template('login_hospital.html', msg = msg)

#sign up hospital starts----------------------------------------

@app.route("/signup_hospital",methods=['GET','POST'])
def signup_hospital():
    msg = ''
    if request.method == 'POST' :
        hospital_name = request.form['hospital_name']
        hospital_date = request.form['hospital_date']
        hospital_contact_number = request.form['hospital_contact_number']
        hospital_address = request.form['hospital_address']
        hospital_city = request.form['hospital_city']
        hospital_state = request.form['hospital_state']
        hospital_email = request.form['hospital_email']
        hospital_password = request.form['hospital_password']
        sql = "SELECT * FROM hospitals_details WHERE hospital_name =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,hospital_name)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg = 'Account already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', hospital_email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', hospital_name):
```

```
            msg = 'name must contain only characters and numbers !'
        else:
            mailtest_hosp_registration(hospital_email)
            insert_sql = "INSERT INTO hospitals_details VALUES (?, ?, ?, ?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, hospital_name)
            ibm_db.bind_param(prep_stmt, 2, hospital_date)
            ibm_db.bind_param(prep_stmt, 3, hospital_contact_number)
            ibm_db.bind_param(prep_stmt, 4, hospital_address)
            ibm_db.bind_param(prep_stmt, 5, hospital_city)
            ibm_db.bind_param(prep_stmt, 6, hospital_state)
            ibm_db.bind_param(prep_stmt, 7, hospital_email)
            ibm_db.bind_param(prep_stmt, 8, hospital_password)

            ibm_db.execute(prep_stmt)
            msg = 'You have successfully registered as recipient!'
            # mailtest(usermail)
            return render_template('login_hospital.html', msg = msg)
    elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template("signup_hospital.html")

#sign up hospital ends----------------------------------------------


#donor list table starts-------------------------------------------------------
@app.route('/donorlist')
def donorlist():
    donors_details = []
    sql = "SELECT * FROM DONORS_DETAILS"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        donors_details.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)
    if donors_details:
        return render_template("homepage_hospital.html", donors_details = donors_details)

#donor list table endss--------------------------------------------------------


#request plasma----------------------------------------------------------------

@app.route("/request_plasma",methods=['GET','POST'])
```

```
def request_plasma():
    msg = ''
    if request.method == 'POST' :
        hospital_name = request.form['hospital_name']
        hospital_contact_number = request.form['hospital_contact_number']
        patient_name = request.form['patient_name']
        patient_age = request.form['patient_age']
        patient_blood_group = request.form['patient_blood_group']
        cause_of_plasma = request.form['cause_of_plasma']
        hospital_address = request.form['hospital_address']
        hospital_city = request.form['hospital_city']
        hospital_state = request.form['hospital_state']
        hospital_email = request.form['hospital_email']
        sql = "SELECT * FROM request_hospital_details WHERE hospital_name =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,hospital_name)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg = 'Account already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', hospital_email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', hospital_name):
            msg = 'name must contain only characters and numbers !'
        else:
            mailtest_request_plasma(hospital_email)
            insert_sql = "INSERT INTO request_hospital_details VALUES (?, ?, ?, ?,?,?,?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, hospital_name)
            ibm_db.bind_param(prep_stmt, 2, hospital_contact_number)
            ibm_db.bind_param(prep_stmt, 3, patient_name)
            ibm_db.bind_param(prep_stmt, 4, patient_age)
            ibm_db.bind_param(prep_stmt, 5, patient_blood_group)
            ibm_db.bind_param(prep_stmt, 6, cause_of_plasma)
            ibm_db.bind_param(prep_stmt, 7, hospital_address)
            ibm_db.bind_param(prep_stmt, 8, hospital_city)
            ibm_db.bind_param(prep_stmt, 9, hospital_state)
            ibm_db.bind_param(prep_stmt, 10, hospital_email)
            ibm_db.execute(prep_stmt)
            msg = 'You have successfully registered to request plasma. our admin will
connect you shortly with donor details!'
            # mailtest(usermail)
            return render_template('success.html', msg = msg)
```

```
    elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template("request_plasma.html")


# sendgrid integration
#donor registration starts----------------------------------
def mailtest_registration(to_email):
    sg = sendgrid.SendGridAPIClient(api_key='')
    from_email = Email("itaswinic@gmail.com")
    subject = "Registered Successfull as a DONOR!"
    content = Content("text/plain", "You have successfully registered as donor. Please
Login using your Username and Password to donate Plasma.")
    mail = Mail(from_email, to_email, subject, content)
    response = sg.client.mail.send.post(request_body=mail.get())
    print(response.status_code)
    print(response.body)
    print(response.headers)

#donor registration ends----------------------------------

#hospital registration starts----------------------------------
def mailtest_hosp_registration(to_email):
    sg = sendgrid.SendGridAPIClient(api_key='')
    from_email = Email("itaswinic@gmail.com")
    subject = "Registered Successfull as a Hospital!"
    content = Content("text/plain", "You have successfully registered as recipient. Please
Login using your Username and Password to request Plasma.")
    mail = Mail(from_email, to_email, subject, content)
    response = sg.client.mail.send.post(request_body=mail.get())
    print(response.status_code)
    print(response.body)
    print(response.headers)

#hospital registration ends----------------------------------

#request plasma confirmation mail starts----------------------------------
def mailtest_request_plasma(to_email):
    sg = sendgrid.SendGridAPIClient(api_key='')
    from_email = Email("itaswinic@gmail.com")
    subject = "Successfully registered to request plasma!"
    content = Content("text/plain", "You have successfully registered to request plasma.
our admin will connect you shortly with donor details")
    mail = Mail(from_email, to_email, subject, content)
```

```
    response = sg.client.mail.send.post(request_body=mail.get())
    print(response.status_code)
    print(response.body)
    print(response.headers)

#request plasma confirmation mail ends----------------------------------

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


if __name__ == '__main__':
   app.run(host='0.0.0.0', port=5000, debug=True)
```

style.css

```
*{
    margin:0;
    padding:0;
    box-sizing: border-box;
    font-family: 'Montserrat', sans-serif;
}

.what_is_plasma_donation_section{
    padding:3em;
    max-width:100%;
    display:grid;
    grid-template-columns: repeat(1,1fr);
    gap:1.5em;
}

.what_is_plasma_donation_section .what_is_plasma_donation h2{
    color:#E85A4F;
}

.what_is_plasma_donation_section .what_is_plasma_donation p{
    text-align: justify;
}

.button-30 {
    align-items: center;
```

```
    appearance: none;
    background-color: rgba(234, 231, 220,40%);
    border-radius: 4px;
    border-width: 0;
    box-shadow: rgba(45, 35, 66, 0.4) 0 2px 4px,rgba(45, 35, 66, 0.3) 0 7px 13px -
3px,rgba(234, 231, 220,40%) 0 -3px 0 inset;
    box-sizing: border-box;
    color: #E85A4F;
    cursor: pointer;
    display: inline-flex;
    font-family: "JetBrains Mono",monospace;
    height: 48px;
    justify-content: center;
    line-height: 1;
    list-style: none;
    overflow: hidden;
    padding-left: 16px;
    padding-right: 16px;
    position: relative;
    text-align: left;
    text-decoration: none;
    transition: box-shadow .15s,transform .15s;
    user-select: none;
    -webkit-user-select: none;
    touch-action: manipulation;
    white-space: nowrap;
    will-change: box-shadow,transform;
    font-size: 18px;
  }

  .button-30:focus {
    box-shadow: rgba(234, 231, 220,40%) 0 0 0 1.5px inset, rgba(45, 35, 66, 0.4) 0 2px
4px, rgba(45, 35, 66, 0.3) 0 7px 13px -3px, rgba(234, 231, 220,40%) 0 -3px 0 inset;
  }

  .button-30:hover {
    box-shadow: rgba(45, 35, 66, 0.4) 0 4px 8px, rgba(45, 35, 66, 0.3) 0 7px 13px -3px,
rgba(234, 231, 220,40%) 0 -3px 0 inset;
    transform: translateY(-2px);
  }

  .button-30:active {
    box-shadow: rgba(234, 231, 220,40%) 0 3px 7px inset;
    transform: translateY(2px);
```

```
 }

 .what_is_plasma_donation_section .what_is_plasma_donation button a{
    text-decoration: none;
    color:#E85A4F;
 }

@media all and (max-width:1000px){


   .what_is_plasma_donation_section{
      padding:1em;

   }
}

input[type=text], select, textarea {
   width: 100%;
   padding: 12px;
   border: 1px solid #ccc;
   border-radius: 4px;
   box-sizing: border-box;
   margin-top: 6px;
   margin-bottom: 16px;
   resize: vertical
 }

 input[type=submit] {
   background-color: #E85A4F;
   color: white;
   padding: 12px 20px;
   border: none;
   border-radius: 4px;
   cursor: pointer;
 }

 input[type=submit]:hover {
   background-color: #E85A4F;
 }

 .container {
   border-radius: 5px;
  padding:20px;
   max-width:100%;
```

```
 }

.container h2{
    color:#E85A4F;

}
```

Login.css

```
*{
    margin:0;
    padding:0;
    box-sizing: border-box;
    font-family: 'Montserrat', sans-serif;
}

body{
    background-color: rgba(234, 231, 220,40%);
}

.log_in_section{
   max-width: 100%;
   height:100vh;
    display:grid;
    grid-template-columns: 1fr;
    align-items: center;
    justify-items: center;
}

.log_in_container{
    box-shadow: rgba(0, 0, 0, 0.15) 0px 15px 25px, rgba(0, 0, 0, 0.05) 0px 5px 10px;
    border-radius: 5px;
    padding:2em;
    max-width: 60%;
    height:auto;
    background-color: #E85A4F;

}

.log_in_container h2{
    text-align: center;
    margin-bottom: 5px;
}
```

```
.log_in_container h3{
    text-align: center;
    margin-top: 10px;
}

.log_in_container input{
    width: 100%;
    padding:5px;
    border-radius: 5px;
    border-color: #E85A4F;
    outline: none;
}



.button-38 {
    margin-top:10px;
  background-color: #FFFFFF;
  border: 0;
  border-radius: .5rem;
  box-sizing: border-box;
  color: #111827;
  font-family: "Inter var",ui-sans-serif,system-ui,-apple-system,system-ui,"Segoe
UI",Roboto,"Helvetica Neue",Arial,"Noto Sans",sans-serif,"Apple Color Emoji","Segoe UI
Emoji","Segoe UI Symbol","Noto Color Emoji";
  font-size: .875rem;
  font-weight: 600;
  line-height: 1.25rem;
  padding: .75rem 1rem;
  text-align: center;
  text-decoration: none #D1D5DB solid;
  text-decoration-thickness: auto;
  box-shadow: 0 1px 3px 0 rgba(0, 0, 0, 0.1), 0 1px 2px 0 rgba(0, 0, 0, 0.06);
  cursor: pointer;
  user-select: none;
  -webkit-user-select: none;
  touch-action: manipulation;
}

.button-38:hover {
  background-color: rgb(249,250,251);
}

.button-38:focus {
```

```
  outline: 2px solid transparent;
  outline-offset: 2px;
}

.button-38:focus-visible {
  box-shadow: none;
}

@media all and (max-width: 500px){

    .log_in_section{
        padding:3px;

    }
    .log_in_container{
      padding:1em;
      max-width: 100%;


    }


}

.log_in_container .log_in_form form p a{

    color:black;
}
```

Log_in.css

```
*{
    margin:0;
    padding:0;
    box-sizing: border-box;
    font-family: 'Montserrat', sans-serif;
}

body{
    background-color: rgba(234, 231, 220,40%);
}

.log_in_section{
    max-width: 100%;
```

```
    height:400px;
    display:grid;
    grid-template-columns: 1fr;
    align-items: center;
    justify-items: center;
}

.log_in{
    display:grid;
    grid-template-columns: 1fr;
    gap:3em;
}
.log_in h2{
    text-align: center;
    color: #E85A4F;
}

.button-30 {
  align-items: center;
  appearance: none;
  background-color: #FCFCFD;
  border-radius: 4px;
  border-width: 0;
  box-shadow: rgba(45, 35, 66, 0.4) 0 2px 4px,rgba(45, 35, 66, 0.3) 0 7px 13px -
3px,#D6D6E7 0 -3px 0 inset;
  box-sizing: border-box;
  color: #E85A4F;
  cursor: pointer;
  display: inline-flex;
  font-family: "JetBrains Mono",monospace;
  height: 48px;
  justify-content: center;
  line-height: 1;
  list-style: none;
  overflow: hidden;
  padding-left: 16px;
  padding-right: 16px;
  position: relative;
  text-align: left;
  text-decoration: none;
  transition: box-shadow .15s,transform .15s;
  user-select: none;
  -webkit-user-select: none;
  touch-action: manipulation;
```

```
  white-space: nowrap;
  will-change: box-shadow,transform;
  font-size: 18px;
}

.button-30:focus {
  box-shadow: #D6D6E7 0 0 0 1.5px inset, rgba(45, 35, 66, 0.4) 0 2px 4px, rgba(45, 35,
66, 0.3) 0 7px 13px -3px, #D6D6E7 0 -3px 0 inset;
}

.button-30:hover {
  box-shadow: rgba(45, 35, 66, 0.4) 0 4px 8px, rgba(45, 35, 66, 0.3) 0 7px 13px -3px,
#D6D6E7 0 -3px 0 inset;
  transform: translateY(-2px);
}

.button-30:active {
  box-shadow: #D6D6E7 0 3px 7px inset;
  transform: translateY(2px);
}

.button-30 a{
    color:#E85A4F;
    text-decoration: none;
}
```

GITHUB & PROJECT DEMO LINK

Github Link: https://github.com/IBM-EPBL/IBM-Project-525-1658305118

Project Demo Link:
https://drive.google.com/file/d/1voZX5NT95AMxAhQPcj_YcfEOlq3vxal5/view?usp=sharing