PLASMA DONOR APPLICATION

Team ID	:	PNT2022TMID17803
Category	:	Cloud Application Development
Team Members	:	Abhirami S S (713319CS003)
		Charu Nethra S (713319CS025)
		Kishore Kumar K (713319CS067)
		Dinesh Kumar C (713319CS508)
Industry Mentor	:	Navya
Faculty Mentor	:	Vijayalakshmi N

1.INTRODUCTION:

Our human body features countless miracles and capabilities that are sometimes quite complicated, and one of those marvels found in our blood is Plasma. Plasma is considered as one of the vital liquid components of blood which contributes over 55% of total blood volume along with water, salts, enzymes, antibodies, and other proteins. Plasma donation is now acknowledged as one of the most pertinent acts that help save numerous lives. The donated plasma can help patients to break free from fatal illnesses such as haemophilia, immunological deficiencies, and other blood disorders.

Using advanced technologies, we can quickly and conveniently access information stored anywhere, at any time. By guaranteeing that our data is constantly available, an internet cloud architecture boosts organisational productivity and efficiency. We are able to link millions of contributors and requestors by converting to cloud information and using a simple yet interactive web application.

1.1. PROJECT OVERVIEW:

Plasma Donor Application is designed and developed to provide a consistent yet engaging portal that connects plasma requesters and donors, serving as a seamless medium of interaction that is advantageous to both donors and receivers. The request on plasma will be fulfilled by displaying the available donor list yet incase of request on particular blood type the user will be notified by the time of availability. Our goal is to provide the correct match donor information to the receiver on time while also maintaining the donor information in a secure database for future use.

1.2. PURPOSE:

The sole purpose of plasma donors is to provide equal importance to donor registration and receiver's search results. From the perspective of the donor, there are a few constraints that must be adhered to. In our web application, we took all the necessary limitations before donating plasma into consideration and created the portal appropriately. In addition to needing to be at least 18 years old and

weigh 50 kilograms, plasma donors are only permitted to provide one donation per 28 days. This can avoid clutter or confusion among donors and requestors.

The requesters can have access over the donors contact details and have a conversation with them in-call or in-person in a safe and secure manner as we have all the records and entry of users in hand. . The requester might look for the most common blood group i.e, AB, or submit a request for his or her preferred blood group. In some cases if the requested blood type plasma donor is not available the application records the request and notifies the user. And apart from all the basic functionalities the application also includes a guide to preparation for plasma donation.

2. LITERATURE SURVEY:

2.1.EXISTING PROBLEM:

[1]" Evaluating plasma holds in the presence of multiple infections" by E. H. Kaplan in 2001

This research broadens the analysis to include numerous illnesses. Given the marginal incidence rates for the infections checked, upper and lower bounds are calculated for important quantities such as the probability of intercepting an infectious but undetected donation, the expected number of infections intercepted per donation, and the net economic benefits of the holding policy.

[2]"Nearest Blood & Plasma Donor Finding: A Machine Learning Approach" by N. Das and M. A. Iqbal in 2020

This study focuses on the development of a platform with clustering algorithms that will work together to deliver the quickest solution to identify a blood or plasma donor. Closest blood or plasma donors of the same group in a certain location can be examined more quickly and efficiently. With machine learning in zone the author states about the various ooportunityes and cases where finding the best match plasma donor is simplified with a ML approach.

[3] ”Convalescent Plasma Therapy: Data-driven approach for finding the Best Plasma Donors” by M N Noorshidha and Dr.G.Aghila in 2021.

The difficulty and difficulties of locating a donor for convalescent plasma treatment are highlighted in the study. It demonstrates how the issue can be resolved through data-driven methods. A classification model is used to determine whether a donor has the required antibody level for donation, and a regression model is used to determine which donors may have higher levels of antibody in their plasma based on their clinical histories.

[4]”A Cross-Platform Blood Donation Application with a Real-Time, Intelligent, and Rational Recommendation System” by]M. R. J. Maraz, R. Rahman, M. M. U. Hasnain and H. Murad in 2021

In this author's research, they created a real-time, intelligent, and logical recommendation system based on sentiment analysis of user comments, donor response rate, and current geo-location information, and then created a cross-platform application for blood collection and distribution. They created a Bi-directional LSTM-based deep learning model to process and produce features from user feedback. The quality of potential donors' recommendations has greatly increased.

[5]”A Web-based Blood Bank System for Managing Records of Donors and Receipts” by M. Kaur et in 2022

The author has developed a comprehensive system that would connect various hospitals, NGOs, and blood banks to assist patients in tough situations. As a result, the HIPPA model serves as a foundation for security breaches. The planned interface will be simple to use and easy to access, providing a quick, efficient, and dependable means to obtain lifesaving blood at no cost.

2.2. REFERENCES:

[1]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.

[2]N. Das and M. A. Iqbal, "Nearest Blood & Plasma Donor Finding: A Machine Learning Approach," 2020 23rd International Conference on Computer and Information Technology (ICCIT), 2020, pp. 1-6, doi: 10.1109/ICCIT51783.2020.9392739.

[3]M. N. Noorshidha and G. Aghila, "Convalescent Plasma Therapy: Data driven approach for finding the Best Plasma Donors," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 432-439, doi: 10.1109/ICAIS50930.2021.9396012.

[4]M. R. J. Maraz, R. Rahman, M. M. U. Hasnain and H. Murad, "A Cross-Platform Blood Donation Application with a Real-Time, Intelligent, and Rational Recommendation System," 2021 International Conference on Electronics, Communications and Information Technology (ICECIT), 2021, pp. 1-4, doi: 10.1109/ICECIT54077.2021.9641395.

[5]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.

2.3. PROBLEM STATEMENT DEFINITION:

Plasma donation is currently recognized as one of the most important acts that help save many lives. Patients suffering from deadly conditions such as hemophilia, immunological deficiencies, and other blood disorders might benefit from donated plasma. One of the body's responses against illness is the production of antibodies. Plasma, a component of the blood, contains antibodies, plasma from COVID-19 virus-free individuals and can be used to create two preparations. It can be used to create convalescent plasma, which is plasma that already has these antibodies, as a start whic can aid as a great life saving component to save one from several serious illness.

In case of Plasma Donors, majority people are not sure about the prerequisites and hard to find a

plasma donation platform. Even the eligible donor is not much aware of plasma donation compared to that of blood donations. Due to this healthcare centers , patients and many hopsitals find to hard to get plasma on time to save ones life. This acts as an huge disadvantage to people who are in immediate need of blood plasma and to who have less social connections to arrange for an donor.

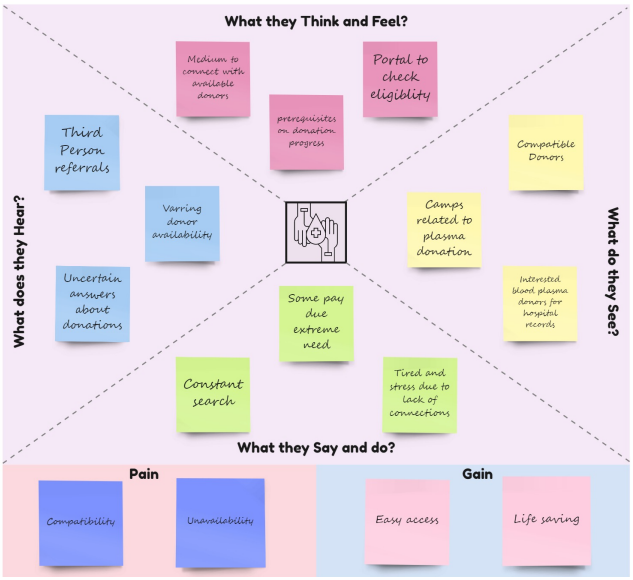
3. IDEATION & PROPSED SOLUION:

We are working on creating a consistent and engaging gateway that connects plasma requesters and donors, acting as a seamless medium of contact that benefits both donors and recipients. The plasma request will be fulfilled by providing the list of available donors. If a user requests a particular blood type, the user will be advised of the available time.

As a result of the web application, donors can help individuals in need by donating life-saving plasma and requestors can feel free to check the portal for availability. With all the advances in technology and medicine, we clearly have the vision to create a safe and healthy environment.

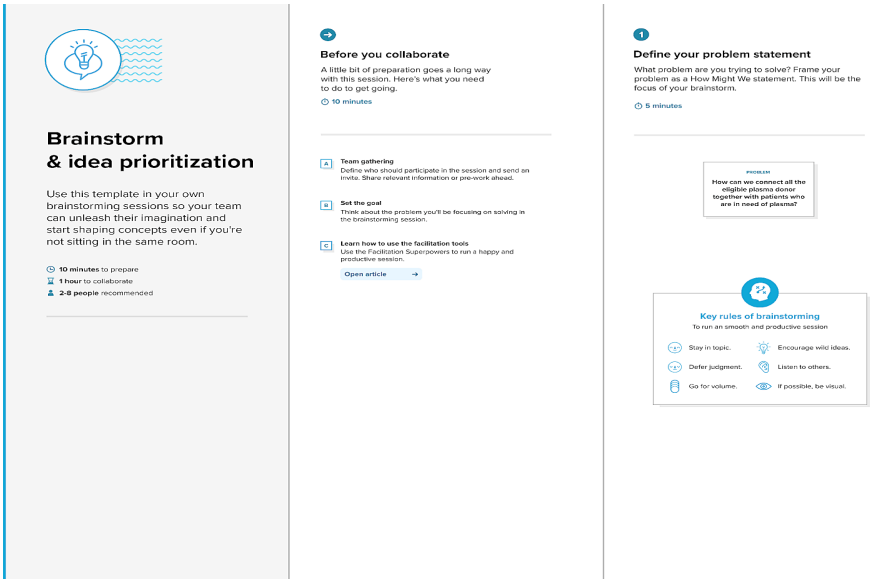
3.1. EMPATHY MAP CANVAS:

Empathy map would aid in picturing the needs, challenges,pain & gain of the public/users and perspectives of the plasma donors and plasma requestors what they feel, hear, and do on times of search and donations.

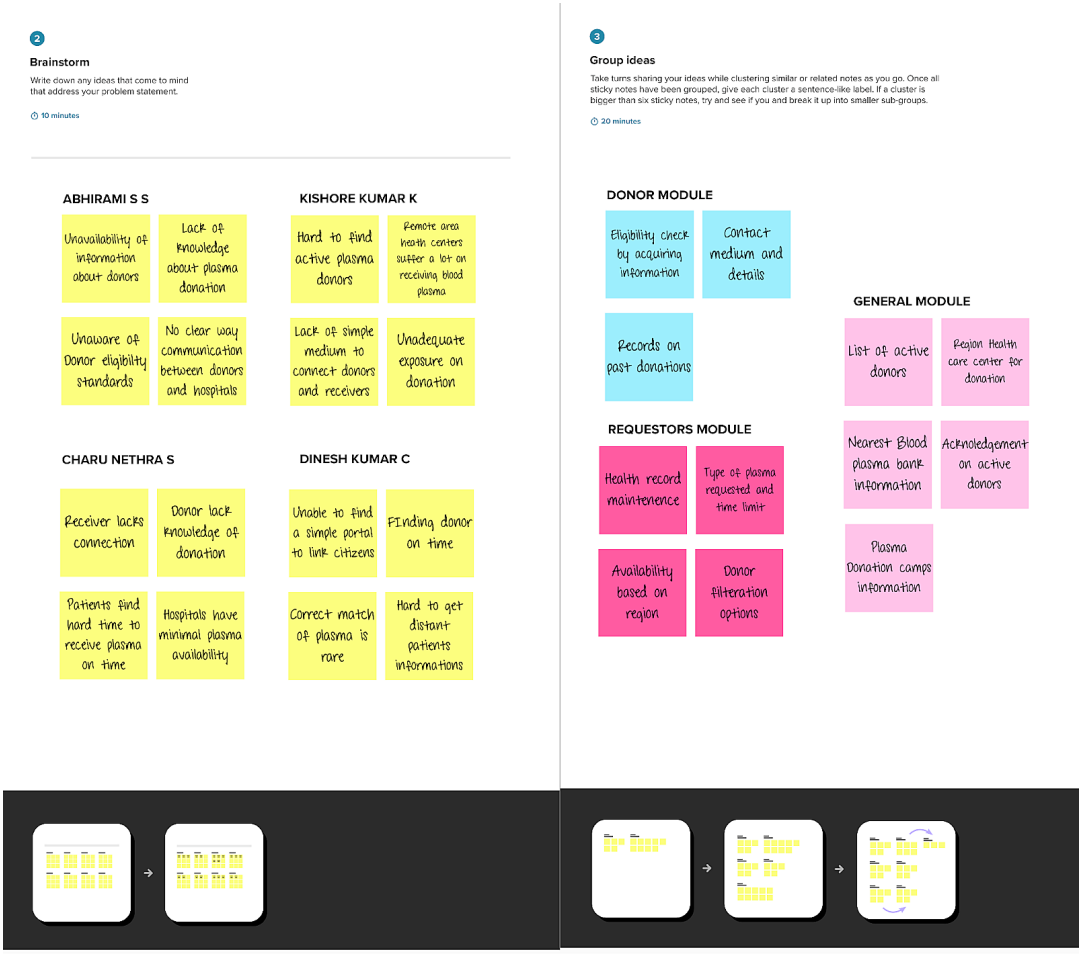


3.2. IDEATION & BRAINSTROMING:

Team Gathering, Collaboration and Select the Problem Statement :



Brainstorm, Idea Listing and Grouping :



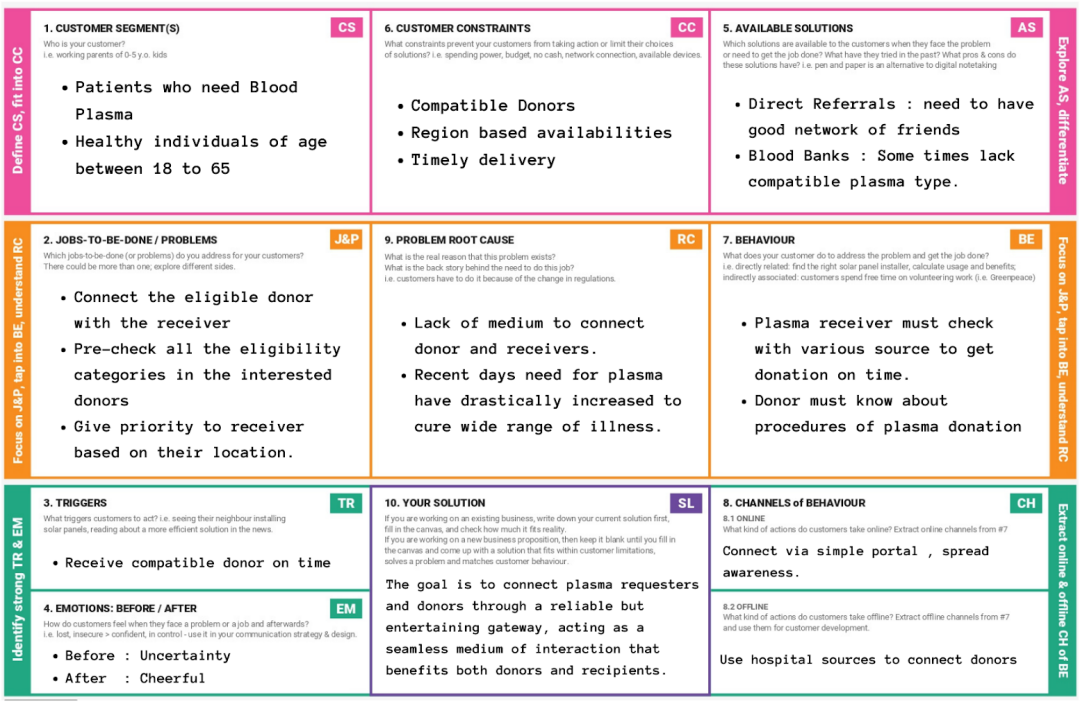
Idea Prioritization :

3.3. PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Plasma is a vital feature of the treatment of many serious health issues. There is no clear and simple way to properly connect donors and recipients. Furthermore, society lacks knowledge of plasma donation, putting patients' health in danger.
2.	Idea / Solution description	Idea is to provide a consistent yet engaging portal that connects plasma requesters and donors, serving as a seamless medium of interaction that is advantageous to both donors and receivers. The request for plasma will be fulfilled by displaying the available donor list. In case of a request on a particular blood type, the user will be notified by the time of availability.
3.	Novelty / Uniqueness	Apart from the Donors list, ● Receivers are given importance here and for quick plasma collection, all filtration process is included from the region based on availability based. ● It's also vital to know if the donor has made any past donations in 28 Days. So our portal will display the eligible and inactive donors by prioritizing accordingly. ● We are also planning to include details on the donation process and Plasma donation camps.

4.	Social Impact / Customer Satisfaction	Effective and simplified portal which can be accessed by anyone anytime for promising donor finds. Can be also connected with healthcare centers for more reliable sources.
5.	Business Model (Revenue Model)	Revenue is generated via page interaction and views, Advertisement impressions. No money is charged for plasma donations.
6.	Scalability of the Solution	This solution can bridge the gap between interested donors in any region with patients who are in need of blood plasma(Priority given to nearby donors). The donors can help individuals in need by donating life-saving plasma and requestors can feel free to check the portal for availability.

3.4. PROBLEM SOLUTION FIT:



4.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 Form in Web Application Registration through Gmail Registration through Google Account Registration through Mobile Number
FR-2	User Confirmation	Confirmation via Registered Email ID Confirmation via OTP
FR-3	User Check-in	In-App Registration for Donors (Interested individuals) In-App Registration for Receivers
FR-4	Donor Eligibility Check	Donor information will be collected and eligible donors will be put front for plasma donation
FR-5	Receiver - Donor Search	Receiver will search the compatible donor with their custom priority
FR-6	Confirmation	Donor information will be shared with receiver for further process
FR-7	Information Sharing	Information about blood plasma donation process, blood plasma camps, and health centers will be uploaded

4.2.NON FUNCTIONAL REQUIREMENTS:

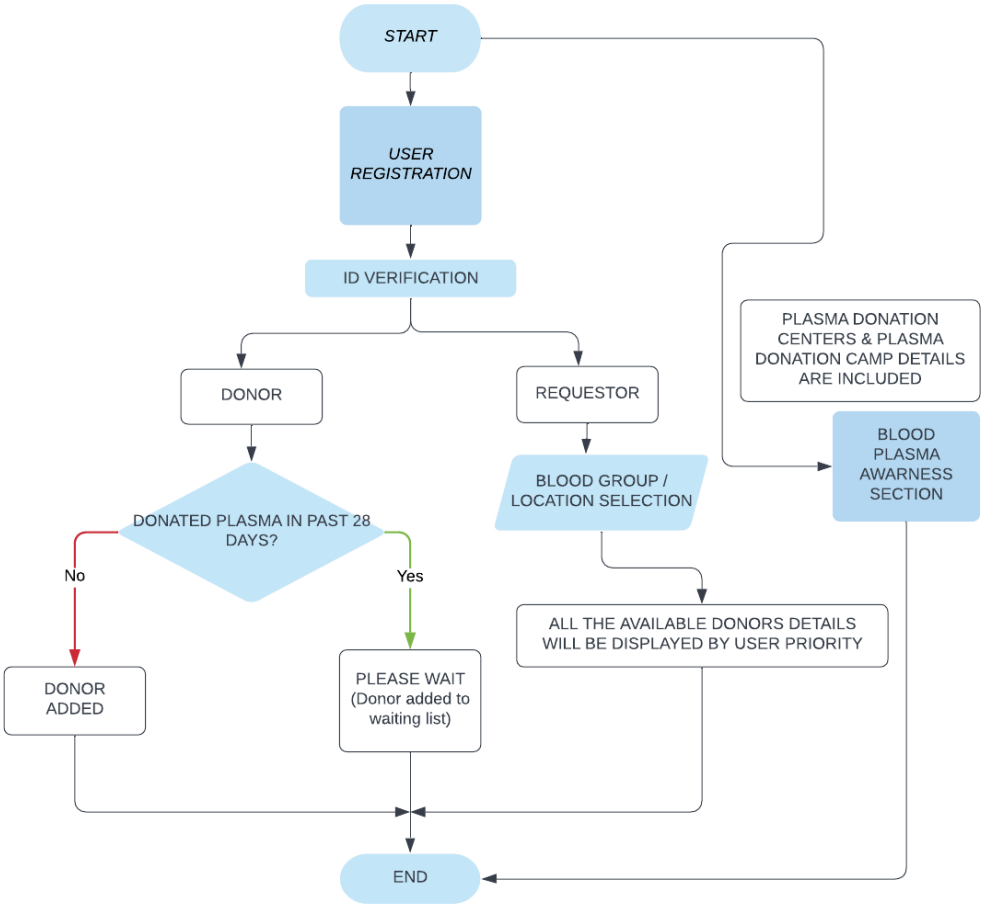
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Every eligible individual can feel free to check on and register themselves for plasma donation
NFR-2	Security	Donor and receiver information is protected and only released upon acceptance
NFR-3	Reliability	Application can provide true and timely information about donors and reach receivers immediately
NFR-4	Performance	Application would be interactive and engaging for both donors and receivers
NFR-5	Availability	Available donors will be given priority based on the receivers request
NFR-6	Scalability	Major amount of donor or receiver information can be gathered and stored with a simple database structure.

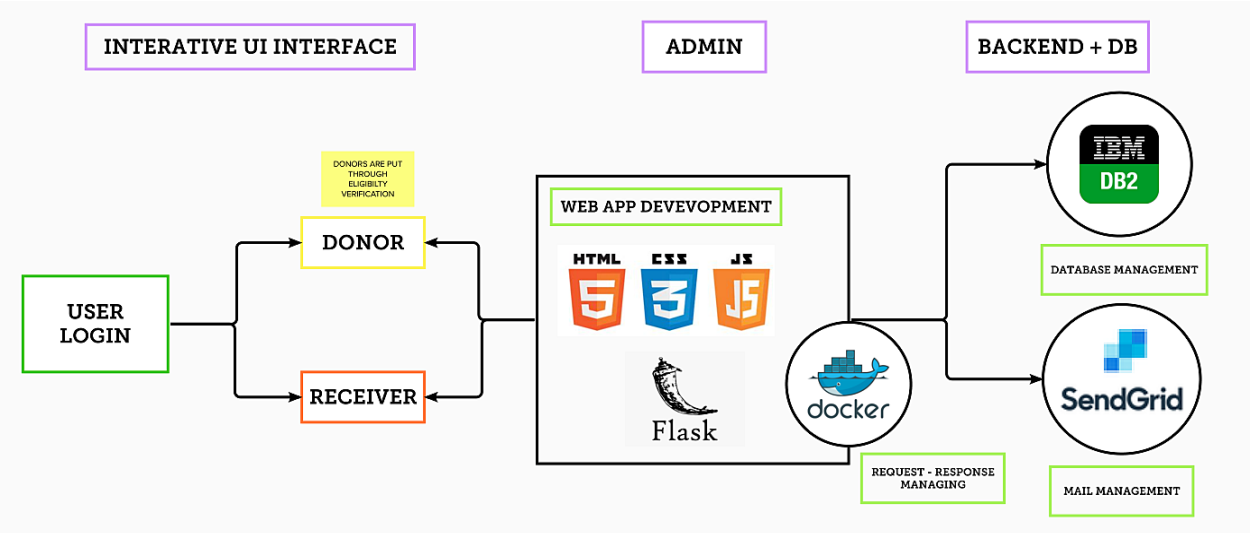
5.PROJECT DESIGN:

Our plasma donor application places equal emphasis on the donor and receiver, as well as individuals interested in learning more about plasma donation. In our web application, we took all of the essential limitations before donating plasma into account and designed the portal accordingly.

5.1.DATA FLOW DIAGRAM:



5.2.SOLUTION & TECHNICAL ARCHITECTURE:



Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	User interaction with the application e.g. Web UI, Mobile App responsiveness.	HTML, CSS, JavaScript
2.	Application Login - Plasma Donor	Logic for a process in the application - Donor criteria check.	Python - Flask
3.	Application Logic - Plasma Receiver	Logic for a process in the application - Receiver information check.	Python - Flask
4.	Confrimation	Comuunication and confirmation between application and user.	SendGrid
5.	Database	Managing - update , retrieve, delete and other query based fetches.	MySQL.
6.	Cloud Database	Database Service on Cloud	IBM DB2
7.	File Storage	File storage requirements	IBM Block Storage or Local Filesystem
8.	External API-1	Store and manage donor informations/	IBM Weather API

9.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud.	Local/ Kubernetes

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 my email, password, and confirming my password.	I can access my account/ dashboard	High	Sprint-1
	Registration	USN-2	As a user, I will receive a confirmation email once I have registered for the application	I can receive a confirmation email & click confirm	High	Sprint-1

	Login	USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
	Login	USN-4	As a user, I can login for the application through Gmail/Google account	I can receive a login request through Gmail & click confirm	Medium	Sprint-1
	Dashboard	USN-5	As a user, I can log into the application by entering email & password	I can start my receiver registration process successfully	High	Sprint-1
	Dashboard	USN-6	As a user, I can log into the application by entering email & password	I can start my donor registration process successfully	High	Sprint-1

Customer (Web user)	Dashboard	USN-7	As a user(Receiver), I can log into the application and check the updates on request	I can view list of donors with an option to sort according to my priority	High	Sprint-1
Customer (Web user)	Dashboard	USN-8	As a user(Donor), I can log into the application and check the updates on request	I can make myself as a donor after multiple levels eligibility checks	High	Sprint-1
Customer Care Executive		USN-9	As Customer Care Executive,all the reviews will be noted and changes are made accordingly	I can view donor or receivers queries and clarify them immediately	Medium	Sprint-2

Administrator		USN-10	As an administrator I all the navigation and external quiers on customer side is cleared,	I can monitor the overall progression of the web application and provide insights on queries.	Medium	Sprint-3

6.PROJECT PLANNING & SCHEDULING:

6.1.SPRINT PLANNING & ESTIMATION:

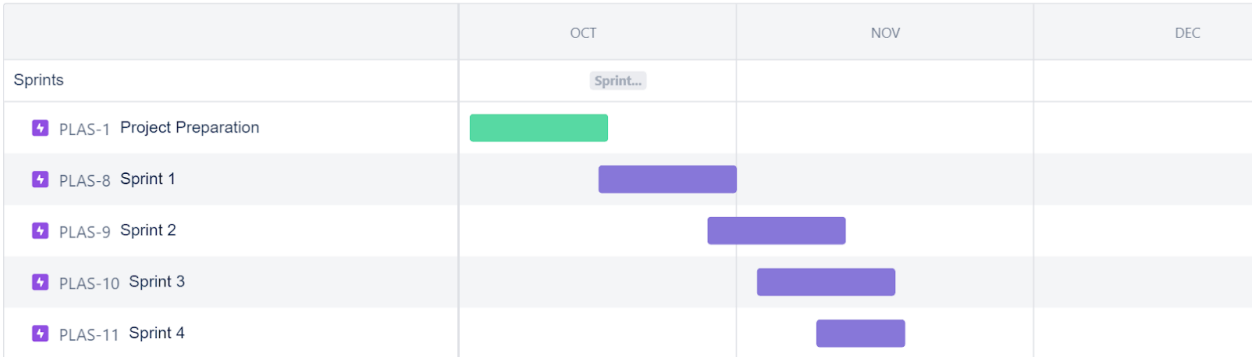
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	UI Development for Login Page, Registration and landing pages.	10	High	Abhirami S S, Kishore Kumar K
Sprint-2	Database connectivity	USN-2	Managing Donor and receiver registration and information gathering and distribution.	10	High	Abhirami S S, Kishore Kumar K

Sprint-3	Containerization	USN-4	Building ChatBot to gather feedback on the application	10	Medium	Charu Nethra S, Dinesh Kumar C
Sprint-3	Feedback	USN-5	Integrating interactive elements to the HTML page and containerizing the app.	10	Medium	Abhirami S S, Charu Nethra s
Sprint-4	SendGrid integration	USN-3	Python and SendGrid integration for mail confirmations.	10	Low	Kishore Kumar K, Dinesh Kumar C

6.2.SPRINT DELIVERY SCHEDULE:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	06 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	18 Nov 2022

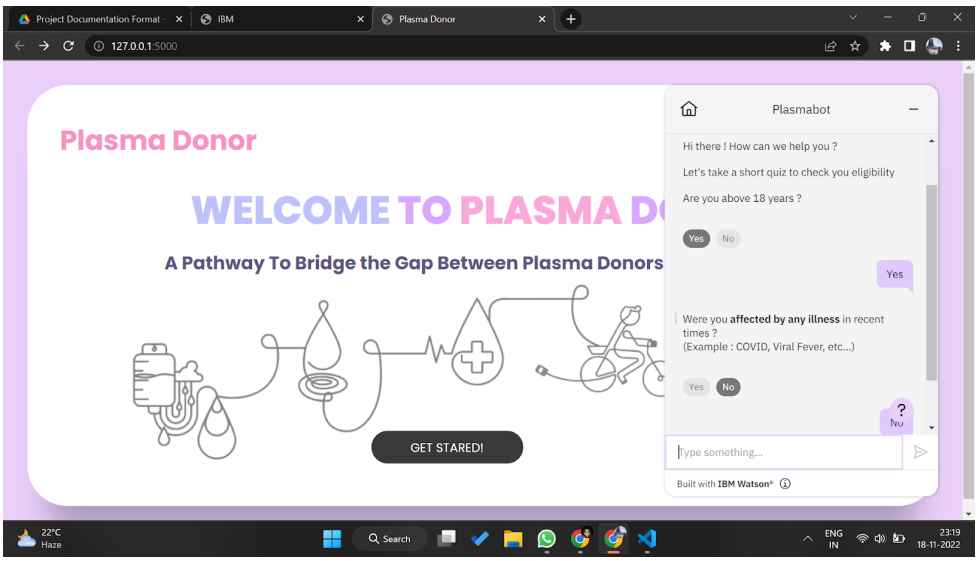
6.3.REPORTS FROM JIRA:

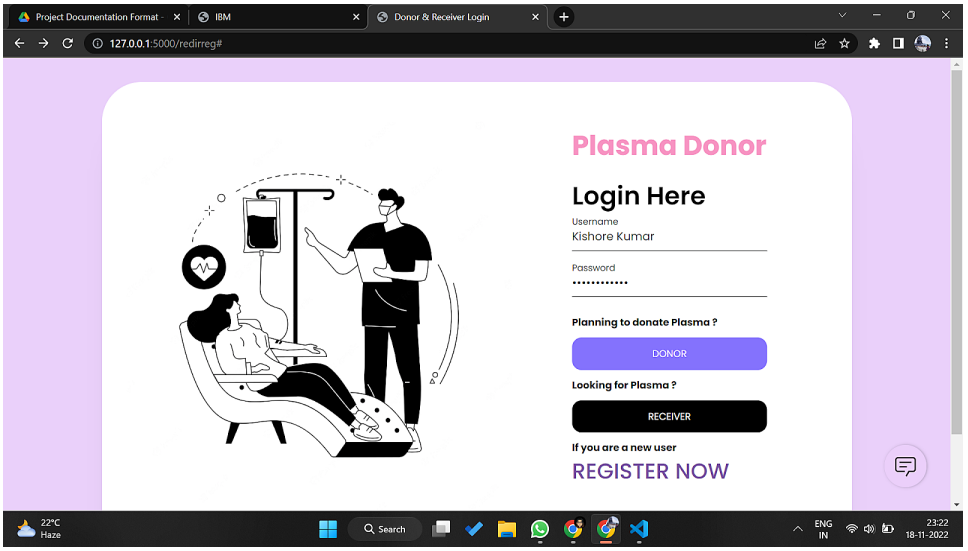


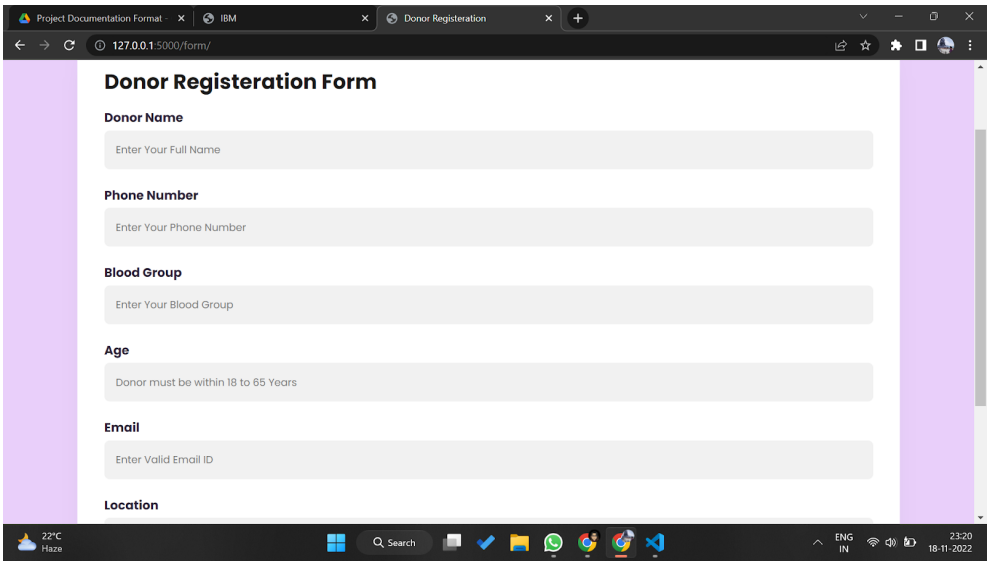
7.CODING AND SOLUTIONING:

7.1.FEATURE 1:

As the main feature from the perspective of the donor, there are a few constraints that must be adhered to. In our web application, we took all the necessary limitation before donating plasma into consideration and created the portal appropriately. In addition to needing to be at least 18 years old and weigh at 50 kilogrammes, plasma donors are only permitted to provide one donation per 28 days. This can avoid clutter or confusion among donors and requestors. Once the Donor knows all the procedures and eligibility criteria he/she can submit their names in the DONOR REGISTRATION.

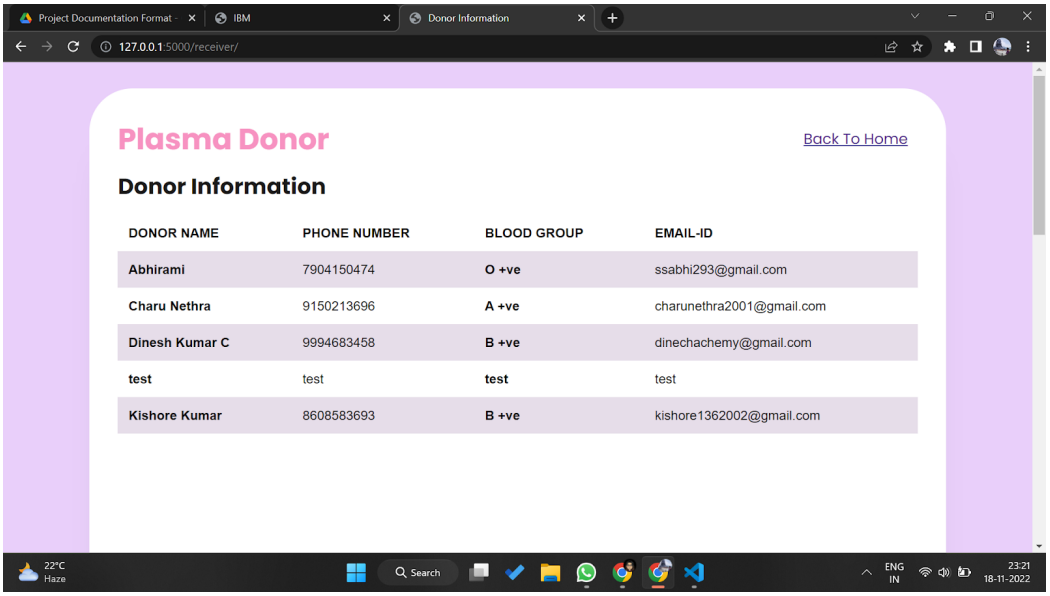






7.2.FEATURE 2:

The requestors can have access over the donors contact details and have a conversation with them in-call or in-person in a safe and secure manner as we have all the records and entry of users in hand. . The requester might look for the most common blood group i.e, AB, or submit a request for his or her prefered blood group. In some case if the requested blood type plasma donor is not available the application records the request and notifies the user.



8.ADVANTAGES AND DISADVANTAGES:

Plasma donations benefit thousands of people all around the world by saving and enhancing their lives. Many medicines and therapies require plasma as a key component. This essential substance can only be supplied by plasma donors; it cannot be made artificially or in a laboratory. A single Donor's contribution aids patients whose lives can be improved or saved by plasma-derived biotherapies. Life-threatening illnesses like haemophilia, immune system deficits, and other blood disorders affect those in need. The disadvantage faced here is that still most people dont have the idea or basics knowledge on plasma donation as they have about blood doantion which can cause a major drawback. Yet this Plasma Donor Application paves a way to connect all donors and requestors with a simple and effective portal with a short guide.

9.CONCLUSION:

Plasma donation is now acknowledged as one of the most pertinent acts that helps save numerous lives. As a result of the web application, the donors can help individuals in need by donating life-saving plasma and requestors can feel free to check the portal for availability. With all the advances in technology and medicine, we clearly have the vision to create a safe and healthy environment.

10.FUTURE SCOPE:

As the next stage for the application we are planning to give more interesting features which includes location based suggestions i.e, suggesting the nearby donor to get the best match plasma in a quicker pace and also provide a sorting option to the requestor so that they can find their donor in ease and avoid spending time on searching manually.

11.APPENDIX:

11.1.SOURCE CODE:

Python Code of file App.py:

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
app = Flask(__name__)
dsn_hostname = "ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud"
dsn_uid = "ztg96113"
dsn_pwd = "AfS01sncyWGSzIEr"
dsn_driver = "{IBM DB2 ODBC DRIVER}"
dsn_database = "BLUDB"
dsn_port = "31505"
dsn_security = "SSL"
dsn = ("DRIVER={0};"
"DATABASE={1};"
"HOSTNAME={2};"
"PORT={3};"
"UID={4};"
"PWD={5};"
"SECURITY={6};").format(dsn_driver,dsn_database,dsn_hostname,dsn_port,dsn_uid,dsn_pwd,dsn_security)
print(dsn)
try:
 conn = ibm_db.pconnect(dsn,"","")
 print("success")
except:
 print(ibm_db.conn_errormsg())


@app.route("/" , methods=['GET', 'POST'])
def register():
 if request.method == 'POST':
   sql_stmt = "insert into USERTBL values(?,?,?,?,?)"
   stmt = ibm_db.prepare(conn, sql_stmt)
   username = request.form['username']
   password = request.form['password']
   phnnumber = request.form['phnnumber']
```

```
    bloodgroup = request.form['bloodgroup']
    email = request.form['email']
    ibm_db.bind_param(stmt, 1, username)
    ibm_db.bind_param(stmt, 2, password)
    ibm_db.bind_param(stmt, 3, phnnumber)
    ibm_db.bind_param(stmt, 4, bloodgroup)
    ibm_db.bind_param(stmt, 5, email)
    try:
      ibm_db.execute(stmt)
      return redirect('/redirreg')
    except:
      print(ibm_db.stmt_errormsg())


  return render_template('landing.html')


@app.route("/redirreg",methods=('GET','POST'))
def loginpage1():
  return render_template("reg.html")


@app.route("/redirland",methods=('GET','POST'))
def loginpage2():
  return render_template("landing.html")


@app.route("/redirtq",methods=('GET','POST'))
def loginpage3():
  return render_template("thanks.html")


@app.route("/redirguide",methods=('GET','POST'))
def loginpage4():
  return render_template("guide.html")


@app.route("/redirelig",methods=('GET','POST'))
def loginpage5():
  return render_template("eligi.html")


@app.route("/form" , methods=['GET', 'POST'])
def donorregister():
  if request.form['action'] == 'SUBMIT':
    sql_stmt = "insert into DONORTBL values(?,?,?,?,?,?)"
    stmt = ibm_db.prepare(conn, sql_stmt)
    name = request.form['name']
    phonenumber = request.form['phonenumber']
    bloodgroup = request.form['bloodgroup']
    age = request.form['age']
    email = request.form['email']
    location = request.form['location']
```

```
    ibm_db.bind_param(stmt, 1, name)
    ibm_db.bind_param(stmt, 2, phonenumber)
    ibm_db.bind_param(stmt, 3, age)
    ibm_db.bind_param(stmt, 4, bloodgroup)
    ibm_db.bind_param(stmt, 5, email)
    ibm_db.bind_param(stmt, 6, location)
    try:
      ibm_db.execute(stmt)
     return redirect('/')
    except:
      print(ibm_db.stmt_errormsg())


  return render_template('form.html')


@app.route("/login",methods=('GET','POST'))
def loginpage():
    if request.form['action'] == 'RECEIVER':
        username = request.form['username']
        password = request.form['password']
        query = "select COUNT(*) from usertbl where username='"+username+"' and password='"+password+"'"
        stmt5 = ibm_db.exec_immediate(conn,query)
        row = ibm_db.fetch_tuple(stmt5)
        if(row[0] ==1 ):
            return redirect("/receiver/")
        return render_template("/redirreg")

    if request.form['action'] == 'DONOR':
        username = request.form['username']
        password = request.form['password']
        query = "select COUNT(*) from usertbl where username='"+username+"' and password='"+password+"'"
        stmt5 = ibm_db.exec_immediate(conn,query)
        row = ibm_db.fetch_tuple(stmt5)
        if(row[0] ==1 ):
            return redirect("/form/")
        return render_template("/redirreg")

@app.route("/receiver/" , methods=['GET', 'POST'])
def shop():
  sql = "SELECT * FROM USERTBL"
  username = []
  phnnumber = []
  bloodgroup =[]
  email = []
  stmt = ibm_db.exec_immediate(conn, sql)
  dictionary = ibm_db.fetch_assoc(stmt)
  while dictionary != False:
    username.append(f'{dictionary["USERNAME"]}')
    phnnumber.append(f'{dictionary["PHNNUMBER"]}')
```

```
    bloodgroup.append(f'{dictionary["BLOODGROUP"]}')
    email.append(f'{dictionary["EMAIL"]}')
    dictionary = ibm_db.fetch_assoc(stmt)
  return render_template('receiver.html', len = len(username), username = username, phnnumber = phnnumber, bloodgroup=bloodgroup, email=email)


@app.route("/form/" , methods=['GET', 'POST'])
def donorform():
  sql = "SELECT * FROM DONORTBL"
  name = []
  phonenumber = []
  age =[]
  bloodgroup =[]
  email =[]
  location =[]
  stmt = ibm_db.exec_immediate(conn, sql)
  dictionary = ibm_db.fetch_assoc(stmt)
  while dictionary != False:
    name.append(f'{dictionary["name"]}')
    phonenumber.append(f'{dictionary["phonenumber"]}')
    age.append(f'{dictionary["age"]}')
    bloodgroup.append(f'{dictionary["bloodgroup"]}')
    email.append(f'{dictionary["email"]}')
    location.append(f'{dictionary["location"]}')
    dictionary = ibm_db.fetch_assoc(stmt)
  return render_template('form.html', len = len(name), name = name , phonenumber =
phonenumber,age=age,bloodgroup=bloodgroup,email=email,location=location )


if __name__ == "__main__":
    app.run(debug=True)
```

Landing Page code-Landing.html:

```
<!DOCTYPE html>
  <html>
  <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</title>
    <style>
      @import url("https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700;800&display=swap");


*,
*::before,
*::after {
  padding: 0;
```

```
  margin: 0;
  box-sizing: border-box;
}

body,
input {
  font-family: "Poppins", sans-serif;
}

main {
  width: 100%;
  min-height: 100vh;
  overflow: hidden;
  background-color: #e9cffa;
  padding: 2rem;
  display: flex;
  align-items: center;
  justify-content: center;
}

.box {
  position: relative;
  width: 100%;
  max-width: 2030px;
  height: 550px;
  background-color: #fff;
  border-radius: 3.3rem;
  box-shadow: 0 60px 40px -30px rgba(0, 0, 0, 0.27);
}

.inner-box {
  position: absolute;
  width: calc(100% - 4.1rem);
  height: calc(100% - 4.1rem);
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}

.header {
  overflow: hidden;
  padding: 20px 10px;
  margin-left: 530px;
  margin-top: -10px;
}
```

```
.header a {
 float: left;
 color: #75519f;
 text-align: center;
 padding: 12px;
 font-size: large;
 font-weight: 100rem;
 line-height: 25px;
  text-decoration: none;
 border-radius: 4px;
}

.header a:hover {
 color: black;
}

@media screen and (max-width: 500px) {
 .header a {
   float: none;
   display: block;
   text-align: right;
 }
}

.logo {
 display: flex;
 align-items: center;
 font-size:x-large;
 color: #f792c1;
 margin-left: 10px;
}

.heading h2 {
 font-size:2rem;
 font-weight:300;
 color: rgb(80, 71, 80);
 margin-top: 10px;
 margin-bottom: 10px;
}

.maintext{
 color:#605481;
 font-family: "Poppins", sans-serif;
 text-align: center;
}
```

```
.toggle {
 color: #75519f;
 text-decoration: none;
 font-size: 0.75rem;
 transition: 0.3s;
}

.toggle:hover {
 color: black;
}

.images{
margin-left: 100px;
}

.btn {
 display: inline-block;
 width: 100%;
 height: 43px;
 background-color: #393939;
 color: #fff;
 border: none;
 cursor: pointer;
 border-radius: 1.8rem;
 font-family: "Poppins", sans-serif;
 font-size: 1.0rem;
 transition: 0.3s;
}

.btn:hover {
 background-color: #8371fd;
}

.title-word {
 animation: color-animation 4s linear infinite;
}

.title-word-1 {
 --color-1: #f8a9da;
 --color-2: #d8a7fe;
 --color-3: #bdc2fb;
}

.title-word-2 {
 --color-1: #bdc2fb;
 --color-2: #f8a9da;
```

```
  --color-3: #d8a7fe;
}

.title-word-3 {
  --color-1: #d8a7fe;
  --color-2: #bdc2fb;
  --color-3: #f8a9da;
}

.title-word-4 {
  --color-1: #bdc2fb;
  --color-2: #f8a9da;
  --color-3: #d8a7fe;
}

@keyframes color-animation {
  0%    {color: var(--color-1)}
  32%   {color: var(--color-1)}
  33%   {color: var(--color-2)}
  65%   {color: var(--color-2)}
  66%   {color: var(--color-3)}
  99%   {color: var(--color-3)}
  100%  {color: var(--color-1)}
}

.start{
  font-family: "Poppins", sans-serif;
  text-align: center;
  margin-top: 10px;
  width:200px;
  height: 100px;
  margin-left: 420px;
  margin-top: -55px;
}

.container {
  display: grid;
  place-items: center;
  text-align: center;
  height: 17vh
}

.title {
  font-family: "Poppins", sans-serif;
  font-weight: 800;
  font-size: 4vw;
```

```
 text-transform: uppercase;
}

@media (max-width: 850px) {
 .box {
  height: auto;
  max-width: 550px;
  overflow: hidden;
 }

 .inner-box {
  position: static;
  transform: none;
  width: revert;
  height: revert;
  padding: 2rem;
 }
}

@media (max-width: 530px) {
 main {
  padding: 1rem;
 }

 .box {
  border-radius: 2rem;
 }

 .inner-box {
  padding: 1rem;
 }
}
  </style>
 </head>
 <body>
   <main>
     <div class="box">
     <div class="inner-box">
      <div class="head">
      <div class="logo">
        <h2>Plasma Donor</h2>
        <div class="header">
          <a href="/redirguide">Donation Guide</a>
          <a href="/redirelig">Eligibility Info</a>
        </div>
        </div>
        <div class="container">
```

```
          <h2 class="title">
            <span class="title-word title-word-1">WELCOME</span>
            <span class="title-word title-word-2">TO</span>
            <span class="title-word title-word-3">PLASMA</span>
            <span class="title-word title-word-4">DONOR</span>
          </h2>
        </div>
        <div class="maintext">
          <h2>A Pathway To Bridge the Gap Between Plasma Donors And Receivers</h2>
        </div>
        <div class="images">
          <img src="https://plasmadonor-images.s3.jp-osa.cloud-object-storage.appdomain.cloud/landing%20img.png" alt="donationillustration"
class="imgsize" >
          </div>
        <div class="start">
          <a href="/redirreg"> <input type="submit" value="GET STARED!" class="btn"></a>
        </div>
        </div>
       </div>
      </div>
    </div>
    <script>
 window.watsonAssistantChatOptions = {
   integrationID: "e3695fa2-19ff-4862-a946-f6920b73601d", // The ID of this integration.
   region: "au-syd", // The region your integration is hosted in.
   serviceInstanceID: "6c0fb290-5fb4-481c-8acc-83f468601c3b", // 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>
    </main>
</body>
</html>
```

Registration and Login Page-reg.html:

```
<!DOCTYPE html>
<html>
<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 & Receiver Login</title>
 <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
```

```
</head>
<body>
  <main>
    <div class="box">
      <div class="inner-box">
        <div class="forms-wrap">
          <form action="/" method="post" class="sign-in-form">
            <div class="logo">
              <h2>Plasma Donor</h2>
            </div>
            <div class="heading">
              <h2>Register Here</h2>

            <div class="actual-form">
            <div class="input-wrap">
              <input type="text" name="username" class="input-field" required>
              <label>User Name</label>
            </div>


            <div class="input-wrap">
              <input type="password" name="password" class="input-field" required>
              <label>Password</label>
            </div>


            <div class="input-wrap">
             <input type="text" name= "phnnumber" class="input-field" required>
             <label>Phone Number</label>
           </div>

           <div class="input-wrap">
             <input type="Blood Group" name= "bloodgroup" class="input-field" required>
             <label>Blood Group</label>
           </div>

           <div class="input-wrap">
             <input type="text" name= "email" class="input-field" required>
             <label>Email</label>
           </div>

            <input type="submit" value="REGISTER" class="sign-btn"
            onClick="alert( 'Account Created Successfully Try LOGGING IN' )" >
           </div>

            <p class="text">
            <h5>Already have an account
            <a href="#" class="toggle"><h3>LOGIN HERE</h3></a></h5>
            </div>
            </p>
```

```
        </form>
        <form action="/login" class="sign-up-form" method="post">
          <div class="logo">
            <h2>Plasma Donor</h2>
          </div>
          <div class="heading">
            <h2>Login Here</h2>

          <div class="actual-form">
            <div class="input-wrap">
               <input type="text" name= "username" class="input-field" required>
               <label>Username</label>
            </div>

            <div class="input-wrap">
              <input type="password" name= "password" class="input-field" required>
              <label>Password</label>
            </div>

            <h5>Planning to donate Plasma ?</h5>
            <input type="submit" name="action" value="DONOR" class="sign-btn1">

            <h5>Looking for Plasma ?</h5>
            <input type="submit" name="action" value="RECEIVER" class="sign-btn2">
             </div>

               <p class="text">
               <h5>If you are a new user
               <a href="#" class="toggle"><h3>REGISTER NOW</h3></a></h5>
               </div>
               </p>
          </form>
        </div>

        <div class="carousel">
         <div class="images-wrapper">
          <img src="https://plasmadonor-images.s3.jp-osa.cloud-object-storage.appdomain.cloud/donate.png" class="image img-1 show" alt="Donation"
>
         </div>
        </div>
    </div>
  </div>
</main>
<footer>Plasma Donor Application by TEAM ID :713319CS067</footer>


 <script>


window.watsonAssistantChatOptions = {
```

```
  integrationID: "e3695fa2-19ff-4862-a946-f6920b73601d", // The ID of this integration.
  region: "au-syd", // The region your integration is hosted in.
  serviceInstanceID: "6c0fb290-5fb4-481c-8acc-83f468601c3b", // 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);
 });

 const inputs = document.querySelectorAll(".input-field");
 const toggle_btn = document.querySelectorAll(".toggle");
 const main = document.querySelector("main");
 const bullets = document.querySelectorAll(".bullets span");
 const images = document.querySelectorAll(".image");

 inputs.forEach((inp) => {
  inp.addEventListener("focus", () => {
   inp.classList.add("active");
  });
  inp.addEventListener("blur", () => {
   if (inp.value != "") return;
   inp.classList.remove("active");
  });
 });

 toggle_btn.forEach((btn) => {
  btn.addEventListener("click", () => {
   main.classList.toggle("sign-up-mode");
  });
 });

 function moveSlider() {
  let index = this.dataset.value;

  let currentImage = document.querySelector(`.img-${index}`);
  images.forEach((img) => img.classList.remove("show"));
  currentImage.classList.add("show");

  const textSlider = document.querySelector(".text-group");
  textSlider.style.transform = `translateY(${-(index - 1) * 2.2}rem)`;

  bullets.forEach((bull) => bull.classList.remove("active"));
  this.classList.add("active");
 }

 bullets.forEach((bullet) => {
```

```
    bullet.addEventListener("click", moveSlider);
  });
 </script>
</body>
</html>
```

Donor Information Collection Page-form.html:

```
<!DOCTYPE html>
<html>
<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" />
</head>
<body>
<main>
 <form action="/redirtq" method="post">
   <div class="box">
     <div class="logo">
       <h2>Plasma Donor</h2>
     </div>
     <div class="title">
       <h2>Donor Registeration Form</h2>
     </div>
       <label for="name"><b>Donor Name</b></label>
       <input type="text" placeholder="Enter Your Full Name" name="name" required>

       <label for="phone number"><b>Phone Number</b></label>
       <input type="text" placeholder="Enter Your Phone Number" name="phonenumber" required>

       <label for="Blood Group"><b>Blood Group</b></label>
       <input type="text" placeholder="Enter Your Blood Group" name="Bloodgroup" required>


       <label for="age"><b>Age</b></label>
       <input type="text" placeholder="Donor must be within 18 to 65 Years" name="age" required>


       <label for="email"><b>Email</b></label>
       <input type="text" placeholder="Enter Valid Email ID" name="email" required>


       <label for="location"><b>Location</b></label>
       <input type="text" placeholder="Enter Your Current Residential Location" name="location" required>
       <div class="text"><p>*Verify your details properly before submission</p></div>
       <a href="/redirtq"><input type="submit" value="SUBMIT" class="sign-btn" /></a>
   </div>
 </form>
```

```
</main>
<footer>Plasma Donor Application by TEAM ID :713319CS067</footer>
</body>
</html>
```

Donor Information Display Page for Receiver-receiver.html:

```
<!DOCTYPE html>
<html>
<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" />
</head>
<body>
<main>
 <form action="/action_page.php" >
   <div class="box">
     <div class="logo">
       <h2>Plasma Donor</h2>
       <div class="header"><a href="/redirland">Back To Home</a> </div>
     </div>
     <div class="title">
       <h2>Donor Information</h2>
     </div>
     <table>

       <tr>
         <th>DONOR NAME</th>
         <th>PHONE NUMBER</th>
         <th>BLOOD GROUP</th>
         <th>EMAIL-ID</th>
       </tr>
       <tr>
         {%for i in range(0, len)%}
         <td><b>{{username[i]}}</b></td>
         <td>{{phnnumber[i]}}</td>
         <td><b>{{bloodgroup[i]}}</b></td>
         <td>{{email[i]}}</td>
       </tr>
       {%endfor%}
     </table>
     </div>
  </form>
</main>
<footer>Plasma Donor Application by TEAM ID :713319CS067</footer>
</body>
```

```
</html>
```

External Stylling Code- style.css:

```
@import url("https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700;800&display=swap");


*,
*::before,
*::after {
 padding: 0;
 margin: 0;
 box-sizing: border-box;
}

body,
input {
 font-family: "Poppins", sans-serif;
}

main {
 width: 100%;
 min-height: 100vh;
 overflow: hidden;
 background-color: #e9cffa;
 padding: 2rem;
 display: flex;
 align-items: center;
 justify-content: center;
}

.box {
 position: relative;
 width: 100%;
 max-width: 1000px;
 height: 620px;
 background-color: #fff;
 border-radius: 3.3rem;
 box-shadow: 0 60px 40px -30px rgba(0, 0, 0, 0.27);
}

.inner-box {
 position: absolute;
 width: calc(100% - 4.1rem);
 height: calc(100% - 4.1rem);
 top: 50%;
 left: 50%;
 transform: translate(-50%, -50%);
```

```
}

.forms-wrap {
 position: absolute;
 height: 100%;
 width: 45%;
 top: 0;
 left: 0;
 display: grid;
 grid-template-columns: 1fr;
 grid-template-rows: 1fr;
 transition: 0.8s ease-in-out;
}

form {
 max-width: 260px;
 width: 100%;
 margin: 0 auto;
 height: 100%;
 display: flex;
 flex-direction: column;
 justify-content: space-evenly;
 grid-column: 1 / 2;
 grid-row: 1 / 2;
 transition: opacity 0.02s 0.4s;
}

form.sign-up-form {
 opacity: 0;
 pointer-events: none;
}

.logo {
 display: flex;
 align-items: center;
 font-size:x-large;
 color: #f792c1;
}

.heading h2 {
 font-size: 2.1rem;
 font-weight: 600;
 color: black;
 margin-top: -10px;
 margin-bottom: 10px;
}
.heading h3 {
```

```
  font-size: 1.8rem;
  font-weight: 500;
 }

.toggle {
 color: #75519f;
 text-decoration: none;
 font-size: 0.75rem;
 font-weight: 500;
 transition: 0.3s;
}

.toggle:hover {
 color: black;
}

.input-wrap {
 position: relative;
 height: 37px;
 margin-bottom: 1.5rem;
}

.input-field {
 position: absolute;
 width: 100%;
 height: 100%;
 background: none;
 border: none;
 outline: none;
 border-bottom: 1px solid rgb(69, 63, 72);
 padding: 0;
 font-size: 0.95rem;
 color: #000000;
 transition: 0.4s;
}
label {
 position: absolute;
 left: 0;
 top: 50%;
 transform: translateY(-50%);
 font-size: 0.95rem;
 color: rgb(34, 29, 37);
 pointer-events: none;
 transition: 0.4s;
}

.input-field.active {
```

```
 border-bottom-color: #000000;
}

.input-field.active + label {
 font-size: 0.75rem;
 top: -2px;
}

.sign-btn {
 display: inline-block;
 width: 100%;
 height: 43px;
 background-color: #000000;
 color: #fff;
 border: none;
 cursor: pointer;
 border-radius: 0.8rem;
 font-size: 0.8rem;
 margin-top: -7px;
 margin-bottom: 30px;
 transition: 0.3s;
}

.sign-btn:hover {
 background-color: #8371fd;
}

.sign-btn1 {
   display: inline-block;
   width: 100%;
   height: 43px;
   background-color: #000000;
   color: #fff;
   border: none;
   cursor: pointer;
   border-radius: 0.8rem;
   font-size: 0.8rem;
   margin-bottom: 10px;
   margin-top: 10px;
   transition: 0.3s;
 }

 .sign-btn1:hover {
   background-color: #8371fd;
 }

 .sign-btn2 {
```

```
  display: inline-block;
  width: 100%;
  height: 43px;
  background-color: #000000;
  color: #fff;
  border: none;
  cursor: pointer;
  border-radius: 0.8rem;
  font-size: 0.8rem;
  margin-top: 10px;
  margin-bottom: 30px;
  transition: 0.3s;
 }

 .sign-btn2:hover {
  background-color: #8371fd;
 }

.text {
 color: rgb(52, 51, 54);
 margin-top: -20px;
 white-space: nowrap;
}

.text a {
 color: rgb(92, 53, 99);
 transition: 0.3s;
}

.text a:hover {
 color: #8371fd;
}

main.sign-up-mode form.sign-in-form {
 opacity: 0;
 pointer-events: none;
}

main.sign-up-mode form.sign-up-form {
 opacity: 1;
 pointer-events: all;
}

main.sign-up-mode .forms-wrap {
 left: 55%;
}
```

```
main.sign-up-mode .carousel {
  left: 0%;
}

.carousel {
  position: absolute;
  height: 100%;
  width: 55%;
  left: 45%;
  top: 0;
  background-color: white;
  border-radius: 2rem;
  display: grid;
  grid-template-rows: auto 1fr;
  padding-bottom: 2rem;
  overflow: hidden;
  transition: 0.8s ease-in-out;
}

.images-wrapper {
  display: grid;
  margin-top: 20px;
  grid-template-columns: 1fr;
  grid-template-rows: 1fr;
}

.image {
  width: 100%;
  grid-column: 1/2;
  grid-row: 1/2;
  opacity: 0;
  transition: opacity 0.3s, transform 0.5s;
}

.img-1 {
  transform: translate(0, -50px);
}

.image.show {
  opacity: 1;
  transform: none;
}

@media (max-width: 850px) {
  .box {
```

```
  height: auto;
  max-width: 550px;
  overflow: hidden;
}

.inner-box {
  position: static;
  transform: none;
  width: revert;
  height: revert;
  padding: 2rem;
}

.forms-wrap {
  position: revert;
  width: 100%;
  height: auto;
}

form {
  max-width: revert;
  padding: 1.5rem 2.5rem 2rem;
  transition: transform 0.8s ease-in-out, opacity 0.45s linear;
}

.heading {
  margin: 2rem 0;
}

form.sign-up-form {
  transform: translateX(100%);
}

main.sign-up-mode form.sign-in-form {
  transform: translateX(-100%);
}

main.sign-up-mode form.sign-up-form {
  transform: translateX(0%);
}

.carousel {
  position: revert;
  height: auto;
  width: 100%;
```

```
  padding: 3rem 2rem;
  display: flex;
 }
 footer{
  align-content: center;
  font-size: small;
  font-weight: 300;
  margin-left: 330px;
  }
 .images-wrapper {
  display: none;
 }

 .text-slider {
  width: 100%;
 }
}

@media (max-width: 530px) {
 main {
  padding: 1rem;
 }

 .box {
  border-radius: 2rem;
 }

 .inner-box {
  padding: 1rem;
 }

 .carousel {
  padding: 1.5rem 1rem;
  border-radius: 1.6rem;
 }

 .text-wrap {
  margin-bottom: 1rem;
 }

 .text-group h2 {
  font-size: 1.2rem;
 }

 form {
```

```
  padding: 1rem 2rem 1.5rem;
 }
}
```

11.2.GITHUB & DEMO VIDEO LINK:

GITHUB LINK :https://github.com/IBM-EPBL/IBM-Project-43990-1660721032

DEMO VIDEO LINK :https://youtu.be/D0nqCiq_yYU