Project Report

1. **INTRODUCTION**
    a. Project Overview
    b. Purpose
2. **LITERATURE SURVEY**
    a. Existing problem
    b. References
    c. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    a. Empathy Map Canvas
    b. Ideation & Brainstorming
    c. Proposed Solution
    d. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    a. Functional requirement
    b. Non-Functional requirements
5. **PROJECT DESIGN**
    a. Data Flow Diagrams
    b. Solution & Technical Architecture
    c. User Stories
6. **PROJECT PLANNING & SCHEDULING**
    a. Sprint Planning & Estimation
    b. Sprint Delivery Schedule
7. **CODING & SOLUTIONING**
    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

Plasma Donor Application

S.NO	REG.NO	NAME	DEPARTMENT	TEAM
1.	211419106033	C.K. Arshad	ECE	Team Lead
2.	211419106087	Giriprasath	ECE	Team Member1
3.	211419106059	Deepak	ECE	Team Member2
4.	211419106060	Deepak Krishnakumar	ECE	Team Member3

DONE BY

TEAM ID:
PNT2022TMID00943

INTRODUCTION

In a plasma-only donation, the liquid portion of the donor's blood is separated from the cells. Blood is drawn from one arm and sent through a high-tech machine that collects the plasma. The donor's red blood cells and platelets are then returned to the donor along with some saline. The process is safe and only takes a few minutes longer than donating whole blood.

Donated plasma is frozen within 24 hours of being donated to preserve its valuable clotting factors. It can be stored for up to one year and thawed for transfusion to a patient when needed. Red Cross donations are often used directly for hospital patient transfusions, rather than pharmaceutical uses.

Project Overview:

Saving the donor information and helping the need by notifying the current donors would be a helping hand. In regard to the problem faced, an application is to be built which would take the donor details store it and inform them upon a request.

Serverless computing is the current trend in software application development. Microservices are a popular new approach for building maintainable, scalable, cloud-based applications. In this project, we will be building a plasma donor app with AWS services like lambda functions, API gateway, and DynamoDB.

Purpose:

Plasma donation requires a commitment both in the amount of time for each donation and frequency of donation. Typically it takes between one and three hours to

donate source plasma, and plasma can be donated twice within a seven day period. Whole blood donation takes less time—under 30 minutes—and donors donate less frequently—no more than once in eight weeks. The programs may fit into a donor's life differently at various times in the donor's life, and are equally important in helping to fulfill a vital medical need.

LITERATURE SURVEY

Existing Problems:

The current pandemic caused by SARS-CoV-2 virus is going to be a prolonged melee. Identifying crucial areas, proactive planning, coordinated strategies and their timely implication is essential for smooth functioning of any system during a crunch.

Addressing the impact of COVID-19 on transfusion services, there are 4 potential challenges viz. blood/ component shortage, donor/ staff safety, consumable supply/ logistics and catering to the convalescent plasma need. In this review article, we will be discussing about these potential challenges in detail along with the necessary mitigative steps to be adopted to tide over the COVID-19 crisis in an Indian set up.

References:

Flask: https://www.youtube.com/embed/uxZuFm5tmhM

Send – Grid: https://sendgrid.com/

Rapid API: https://rapidapi.com/hub

Docker: https://www.youtube.com/embed/pTFZFxd4hOI

Kubernetes: https://www.youtube.com/embed/d6WC5n9G_sM

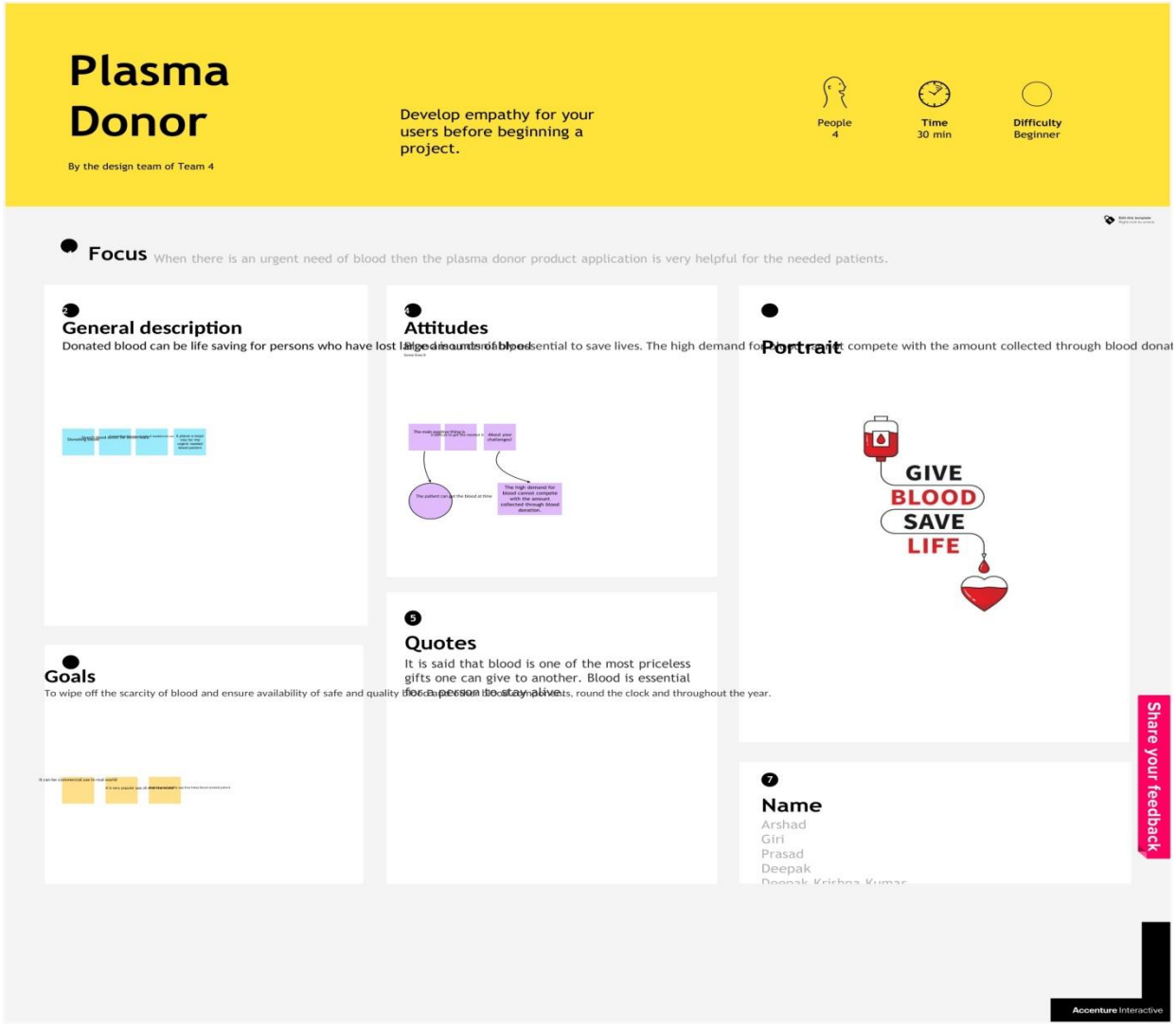
Problem Statement Definition:

The problem definition of the system is to launch an online interaction medium for the blood donation management.

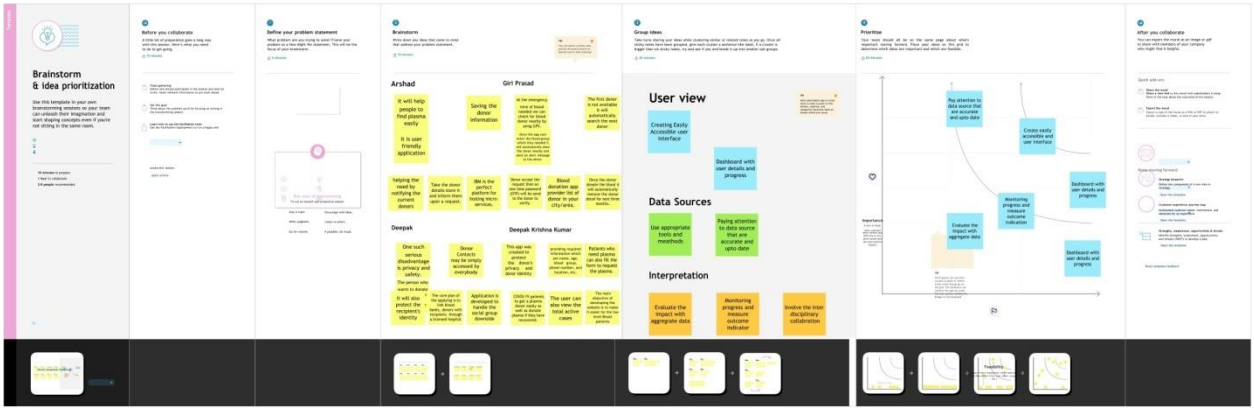
The main aim of this project is to help the people who needs blood in emergency and to associate some donors who are willing to donate their blood to needy people and save their lives.

IDEATION & PROPOSED SOLUTION

1) Empathy Map Canvas:



2) Ideation & Brainstroming:



3) Proposed Solution:

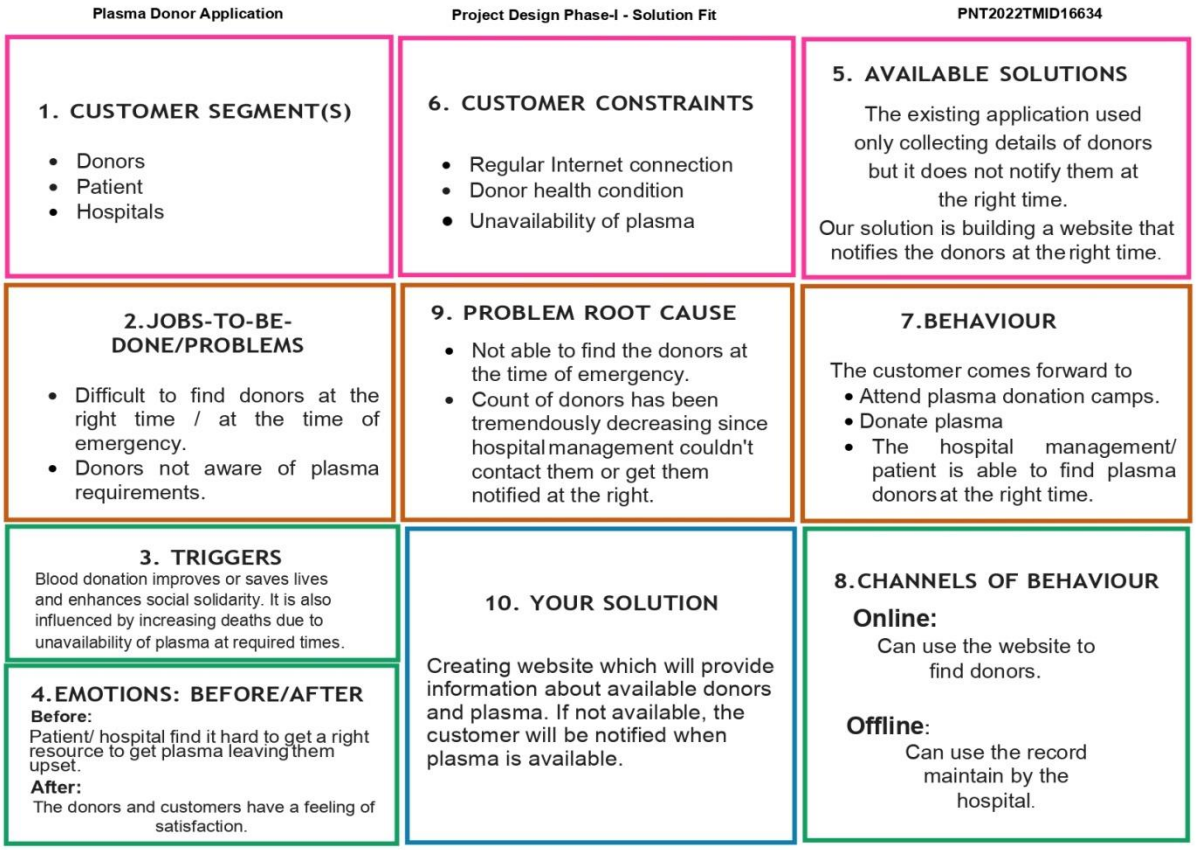
S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Plasma is used for the treatment of many serious health problems. This is why there are blood drives asking people to donate blood, plasma. Plasma is utilized to treat different irresistible sicknesses and it is one of the most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

		User: • If the user wants to donate or receive they have to register with their personal details. • After successful registration of user. • A successful registration email is send to the user. • After successful registration user will be directed to home page. • They will be asked to press whether they will be donor or receiver. • If the user is donor then he/she will fill the donation interest form which includes their Name, blood group details, location, last time donated date , phone number, email id. • After filling the donation form he/she will redirected to page in which he/she can download the e-certificate. • If the user is receiver then he/she can see the list of donors available and they can raise their request and contact donor directly. Admin: • Admin can login using their credentials. • Admin can edit the request. • Admin can delete the request. • Admin can add volunteers.
3.	Novelty / Uniqueness	A User Interface is simple for users to understand. We can use the application anywhere anytime. The user immediately need the plasma for their treatment but the plasma is not available in nearbyhospitals, then user can use this application to raise request and directly contact the donor , request them to donate the plasma. Hospitals can also raise request donors for donation. Somebodywants to donate blood and plasma but they don't know the way to donate then they use this application which will Simple to use and it will save lives ofmany people. Today many of them have

		mobile phones they can install this application and use it to save the lives of people.
4.	Social Impact / Customer Satisfaction	We are living in a modern world and everything can be accessed online. Even though there are many applications there isno proper application for plasma donation. Many of them wish to donate blood and plasma but they are unaware about donation and how they can donate. This application provides opportunity to those who want to donate plasma. Donation of plasma is happening in many places many of them come forward to donate but it is not available at right time for use. Sometimes there is a shortage of plasma of particular type. Additional facilities that we need are to access the patients information quickly before plasma transfusion. To solve this issue software applications are employed with Cloud computing and Internet of Things tool which enable features such as information retrieval and continuous data tracking with analytics. This application avoids circulating of wrong information. A single platform for maintaining genuine information and increase the trust of participants involved in it his activity. It Increase the number of donors.
5.	Business Model (Revenue Model)	This application is accessible by everyone. It is free. Because of the trouble in finding givers who match a specific blood bunch, this application empowers clients to enlist individuals who wish to give plasma andkeep their data in a data set. Nowadays the need for plasma increases. Anyone with basic knowledge can access this app. This can be used anywhere anytime. working with the government we can utilize an Application to help those needing plasma.

6.	Scalability of the Solution	This application helps users to find plasma donors by sitting in home itself instead of searching donors everywhere. When there is a emergency then plasma request to send to everyone. Once the donor is ready to donate receiver is notified about donation. Receiver can contact the donor. With this app donor can know the eligibility to donate and making it easier to locate suitable donor at right time.

4) Problem Solution Fit:



REQUIREMENT ANALYSIS

1) Functional requirement:

- User Registration
- User Confirmation
- Update Profile
- User Authentication

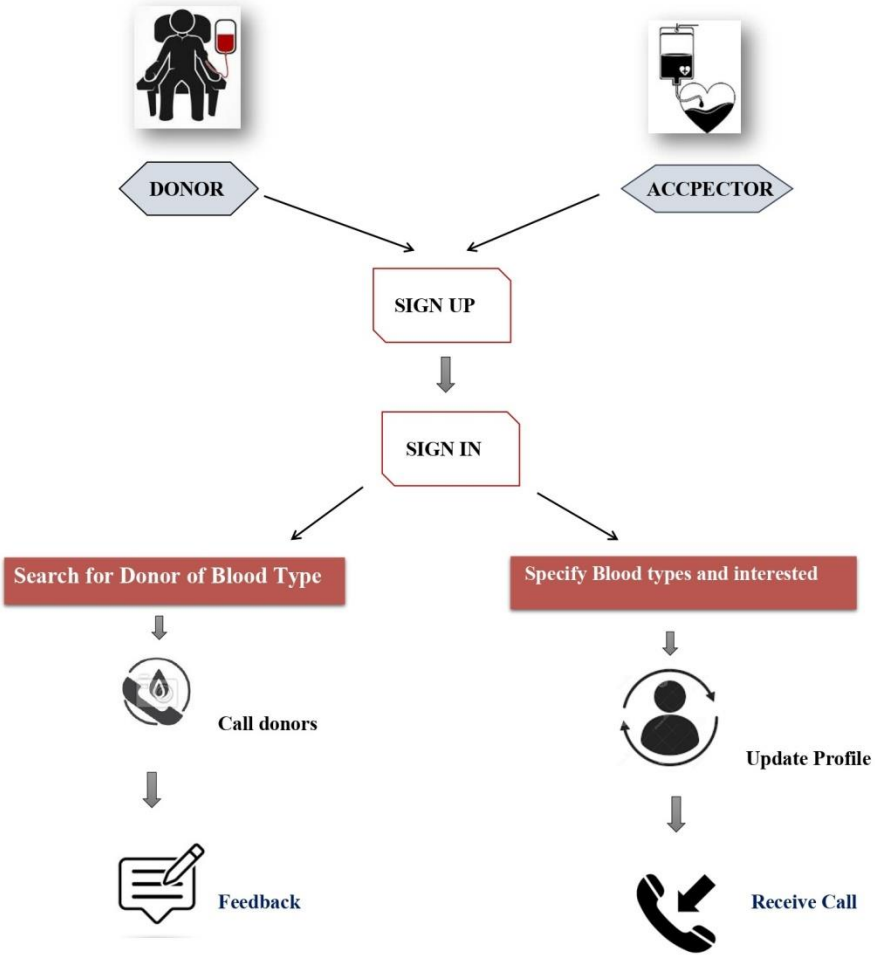
- Report

2) Non-Functional requirements

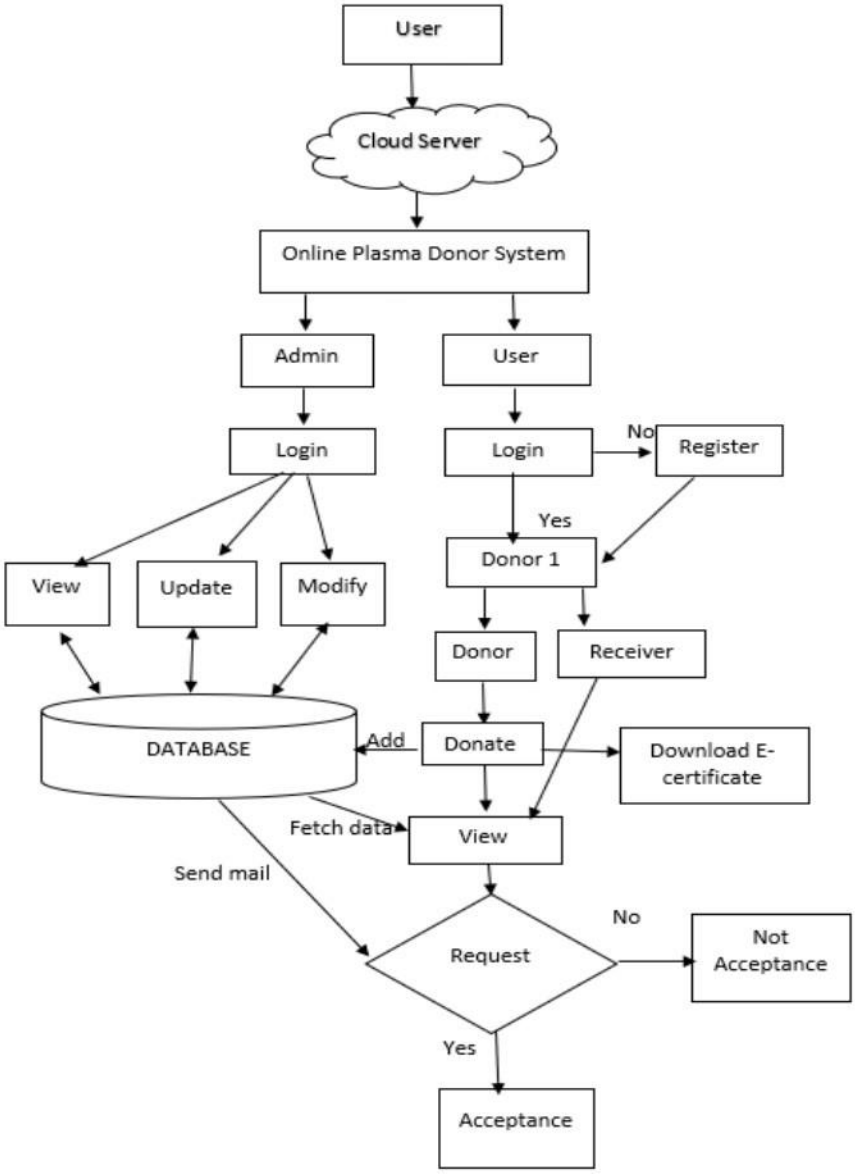
- Usability
- Security
- Reliability
- Performance
- Availability
- Scalability

PROJECT & DESIGN

1) Data Flow Diagrams:



2) Solution & Technical Architecture:



3) User Stories:

- As a user, I can register for the application by entering my email, password, and Confirming my password.
- As a user, I will receive confirmation email once I have registered for the application.
- As a user, I can log into the application by entering email & password.
- As a user, I can fill the details.
- As a user, I will search the needed Blood.

PROJECT PLANNING & SCHEDULING

1) Spring Planning & Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Initial creation process	USN-1	Create template, Static and python flask app.	20	High	C.K. Arshad Giriprasath Deepak Deepak Krishnakumar
Sprint-2	Cloud and database	USN-2	Connecting the python flask app with database, object storage created in Cloud and implementation of Chat bot	20	High	C.K. Arshad Giriprasath Deepak Deepak Krishnakumar
Sprint-3	Deployment in DevOps, Mailing	USN-3	Develop the project, create it as image with docker, containerize in container registry and deploy in Kubernetes, Add the mailing service	20	High	C.K. Arshad Giriprasath Deepak Deepak Krishnakumar
Sprint-4	Testing, Deployment and user experience	USN-4	To do all the testing and to make sure the use of the software handy to user.	20	High	C.K. Arshad Giriprasath Deepak Deepak Krishnakumar

2) Sprint Delivery Schedule:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	5 Days	27 Oct 2022	01 Nov 2022	20	01 Nov 2022
Sprint-2	20	6 Days	02Oct 2022	07 Nov 2022	20	07 Nov 2022
Sprint-3	20	6 Days	08 Nov 2022	13 Nov 2022	20	13 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

CODING & SOLUTION

1) Feature 1:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title style="font-family: sans-serif">Sign up</title>
</head>
<style>
h1{
text-align: center; font-size: 10ch; color: rgb(15, 15, 15);
}
body{
background-color: rgba(84, 22, 126, 0.651);text-align: center;
}
body{
background-image: url("im.gif"); background-position: center; background-repeat:
no-repeat; background-size: cover;
}

p{
color: rgb(19, 14, 14);
font-size: x-large;
font-family:'Times New Roman', Times, serif;

}
small{
color:rgba(168, 61, 61, 0.71); font-size:3ch;
}
input[type=submit] {
background-color: rgba(0, 0, 0, 0);
font-family: 'Times New Roman', Times, serif; border: none;
color: white; padding: 15px 10px; text-decoration: none; margin: 2px 2px; cursor:
pointer;
}
</style>
<body>
<h1><u>SIGNUP</u></h1>
<form method="POST" action="">
<p>USERNAME</p>
```

```
<input type="text" name="username" id="username" maxlength="25" size="25"
required>
<p>EMAIL</p>
<input type="text" name="email" id="email" maxlength="25" size="25" required>
<p>ROLL NUMBER</p>
<input type="text" name="rno" id="rno" maxlength="25" size="25"
required><p>PASSWORD</p>
<input type="password" name="password" id="password" maxlength="25" size="25"
required>
<br><br>
<input type="submit" value="SIGNUP">
<br><br>
<small><u>ALREADY HAVE AN ACCOUNT<a href="{{ url_for('login')
}}"><br></u>
<small><u>LOGIN</u></small>
</form>
</body>
</html>
```

2) Feature 2:

```
@app.route("/login",methods=["GET","POST"]) def login():
if request.method=="POST":
username = request.form.get("username",'') password = request.form.get("password",'')
with sqlite3.connect("users.db") as con:
cur = con.cursor()
cur.execute("select * from user where username=(?) and password=(?)",(username,password))
result = cur.fetchone() con.commit()
if result==None:
return render_template("login.html",msg="Invalid Username and
password")
else:
return render_template("home.html",username=result[1]) return
render_template("login.html")
```

TESTING

Test Cases:

i) Our code is test on various test case to check whether it gives correct output.
ii) It is very useful for the Plasma needed user
iii) In this test case to check the alert message is correctly go to the having that blood grouped person.

User Acceptance Testing:

Our project was tested by an end user to verify that it's working correctly.

RESULTS

Performance Metrics:

Performance monitoring is an important tool which can be used for setting priorities for process improvement. At our centre, we have been monitoring every step in the processes, right from inventory of consumables (both critical and routine) to number of donors reactive for TTI. We conducted a study to measure the impact of monitoring Performance Indicators and how it could be used as a tool for Continuous Quality Improvement (CQI).

ADVANTAGES

- Purpose of project educate donors and employees on plasma and why it is collected 3 main branches of plasma use see the big picture not just a job, not just extra cash countless lives are benefited testimonials of individuals affected.
- Preceptor selection meet the requirements bsn w/ 3 years experience willingness to help limited options 5 nurses total at facility
- Initial perceived benefits minimizing education gap: plasma and its uses employees(myself) are better able to help educate and encourage donations staff meetings and guest speakers donors
- What is plasma pale-yellow liquid portion of blood rbc, wbc, and platelets are suspended in plasma 55% blood volume 92% water albumin, fibrinogen, globulins

CONCLUSION

In our whole project, we have built a platform for the blood donor and receiver. We have combined two well known clustering algorithm, k-means, and agglomerative clustering, to solve the problem. From experimental results analysis, we can descry that our system gives a more robust and efficient result in this regard. Our solepurpose is to reduce the time as well as the sufferings of the people. Ablood or plasma recipient can efficiently get donors using our system in any situation.

FUTURE SCOPE

The effectiveness of the donor selection process is enhanced if relevant information and counselling are provided to prospective donors, enabling them to self-defer if they recognize they are unsuitable to donate blood. Blood donors may be deferred, either on a temporary or permanent basis, on the grounds of their health status, medical or travel history, or TTI risk. Pre-donation counselling is particularly important for individuals who are temporarily or permanently deferred from blood donation, as it provides them with clear information about the reasons for deferral, maintaining healthy lifestyles, and referral for further testing, treatment, care and support, as appropriate.

APPENDIX

1) Source code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title style="font-family: sans-serif">Sign up</title>
</head>
<style>
h1{
text-align: center; font-size: 10ch; color: rgb(15, 15, 15);
}
body{
background-color: rgba(84, 22, 126, 0.651);text-align: center;
}
```

```
body{
background-image: url("im.gif"); background-position: center; background-repeat:
no-repeat; background-size: cover;
}

p{
color: rgb(19, 14, 14);
font-size: x-large;
font-family:'Times New Roman', Times, serif;

}
small{
color:rgba(168, 61, 61, 0.71); font-size:3ch;
}
input[type=submit] {
background-color: rgba(0, 0, 0, 0);
font-family: 'Times New Roman', Times, serif; border: none;
color: white; padding: 15px 10px; text-decoration: none; margin: 2px 2px; cursor:
pointer;
}
</style>
<body>
<h1><u>SIGNUP</u></h1>
<form method="POST" action="">
<p>USERNAME</p>
<input type="text" name="username" id="username" maxlength="25" size="25"
required>
<p>EMAIL</p>
<input type="text" name="email" id="email" maxlength="25" size="25" required>
<p>ROLL NUMBER</p>
<input type="text" name="rno" id="rno" maxlength="25" size="25"
required><p>PASSWORD</p>
<input type="password" name="password" id="password" maxlength="25" size="25"
required>
<br><br>
<input type="submit" value="SIGNUP">
<br><br>
<small><u>ALREADY HAVE AN ACCOUNT<a href="{{ url_for('login')
}}"><br></u>
<small><u>LOGIN</u></small>
</form>
</body>
</html>
```

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

with sqlite3.connect("users.db") as con: cur = con.cursor()
cur.execute("INSERT INTO user (roll_number,email,username,password) VALUES
(?,?,?,?)",(rno,email,username,password) )
con.commit()
return redirect(url_for('login')) return render_template("signup.html")
```

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Login</title>
<link href="style.css">
</head>
<style>
h1{
text-align: center; font-size: 10ch; color: rgb(15, 15, 15);
}
body{
background-color: rgba(17, 15, 87, 0.651);text-align: center;
}
body{
background-image: url("im.gif"); background-position: center; background-repeat:
no-repeat; background-size: cover;
}

p{
color: rgb(19, 14, 14);
font-size: x-large;
font-family:'Times New Roman', Times, serif;

}
</style>
<body>

<h1>LOGIN PAGE</h1> <br>
```

```
<form method="POST" action="">
<p><b>USERNAME</b></p>
<input type="text" name="username" id = "username"> <br>
<p><b>PASSWORD</b></p>
<input type="password" name="password" id="password"> <br>
<br><br>
<input type="submit" value="LOGIN">
<p><small><u>New User?</u><a href="{{ url_for('signup')
}}">Signup</a></small></p>
</form>
</body>
</html
```

```
@app.route("/login",methods=["GET","POST"]) def login():
if request.method=="POST":
username = request.form.get("username",'') password = request.form.get("password",'')
with sqlite3.connect("users.db") as con:
cur = con.cursor()
cur.execute("select * from user where username=(?) and password=(?)",(username,password))
result = cur.fetchone() con.commit()
if result==None:
return render_template("login.html",msg="Invalid Username and
password")
else:
return render_template("home.html",username=result[1]) return
render_template("login.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</title>
<script src="https://kit.fontawesome.com/1625eab60d.js" crossorigin="anonymous"></script>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384- iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous">
<link rel="stylesheet" href="211419106033.css">
</head>
<body>
    <nav class="navbar navbar-dark bg-dark">
```

```
        <div class="container-fluid">
            <a class="navbar-brand" href="#"><h1>Plasma Donor</h1></a>
            <div class="navbar-nav">
                <ul class="nav justify-content-end">
                    <li class="nav-item">
                        <a class="nav-link active" aria-current="page" href="/signin">
Sign in</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="/about">About</a>
                    </li>
                </ul>
                <h1 align="center">Plasma Donor Application</h1>
             </div>
        </div>
    </nav>
    <div style="background-image:url('/static/img/image.jpg);"></div>
    <div class="main" align="center">
        <h1>You warmly welcome to this home page</h1>
    </div>

</body>
</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>About Page</title>
<script src="https://kit.fontawesome.com/1625eab60d.js" crossorigin="anonymous"></script>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384- iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous">
<link rel="stylesheet" href="211419106033.css">
</head>
<body style="background-color: rgba(33, 105, 153, 0.815)">
    <div align="center">
        <h1>About</h1><br><br><br><br>
        <h4>This is the Assignment 2 of the Cloud Application Development</h4>
    </div>

</body>
</html>
```

2) Github Link:

https://github.com/IBM-EPBL/IBM-Project-11160-1659274278

3) Demo Video Link:

https://github.com/IBM-EPBL/IBM-Project-11160-1659274278/tree/main/Final%20Deliverables/Demo%20Video%20Link