Project Report

Team ID	**PNT2022TMID48500**
Project Name	**Plasma Donor Application**

1. INTRODUCTION

1.1 Project Overview

Plasma is a critical part of the treatment for many serious health problems. Therefore, there are blood drives asking people to donate blood plasma. The main goal of our project is to make it easier for the COVID-19 patients to get a plasma donor easily as well as donate plasma if they have recovered. The system targets two types of users: the people who want to donate plasma and the people who need plasma. The user can also view the total active cases, nearby vaccine centres, hospitals address. The main objective of developing the website is to make it easier for the COVID-19 patients to get a plasma donor easily and as soon as possible. Yet, the need for plasma-derived products has been strongly increasing for some years, and blood collection agencies have to adapt if they want to meet this demand. This article aims to review the main motivations and deterrents to whole blood donation, and to compare them with those that we already know concerning plasma donation. Current evidence shows similarities between both behaviours, but also differences that indicate a need for further research regarding plasma donation.

1.2 Purpose

During the COVID 19 crisis, the requirement of plasma became a high priority, and the donor count has become low. Saving the donor information and helping the needy by notifying the current donors list, would be a helping hand. Regarding the problem faced, an application is to be built which would take the donor details, store them and inform them upon a request.

2. LITERATURE SURVEY

2.1 Existing problem

People have to find them physically by visiting hospitals register book and reaching out recovered donors home and sometimes they will be not available at their places and will be went on work. In this type of scenarios diseased person's health gets worse. This is an expensive and will not work as effectively at emergency situations.

PAPER 1

TITLE: Plasma donor app usage behaviour and perceptions: Considerations for a Plasma donation app

AUTHOR NAME: Andrea Potgieter
PUBLICATION YEAR: 2022

DESCRIPTION: This article aimed to determine whether South African Plasma donor app usage behaviour and perceptions were conducive to introduce a Plasma donation app, and what these behaviours and perceptions could reveal, to support South African Plasma Donation Organizations in their recruitment and engagement endeavours. The research problem discussed in this article sought to highlight the app usage behaviour of Plasma donors, and their perceptions about a proposed Plasma donation app. forming part of a larger sequential mixed-methods study, the data presented in this article were gathered through a quantitative online questionnaire involving 2154 South Africans respondents. The value of this research lies in the insight gained into the behaviour and perceptions of South African blood donors, which can inform the conceptualization and design of a Plasma donation app.

PAPER 2

TITLE: Evaluation of the Wateen App in the Plasma -Donation Process in Saudi Arabia

AUTHOR NAME: Turkish Alessa

PUBLICATION YEAR: 2022

DESCRIPTION: The aim of this research was to evaluate the usability, user satisfaction and perceived usefulness of this Plasma donation app in Saudi Arabia. A mixed-method study was conducted comprising a quantitative questionnaire with donor and qualitative semi-structured interviews with healthcare professionals. Descriptive analysis was used for the quantitative data and a thematic approach for the qualitative data. Quantitative data analysis was conducted using SPSS software package 19 to calculate descriptive statistics. This Plasma-donation app is highly usable and acceptable among donors and healthcare professionals in Saudi Arabia, offering several benefits. Some accessibility issues were identified, along with possibilities for improving accessibility and expanding the app's functionalities

PAPER 3

TITLE: Location-based Mobile Application for Plasma Donor Se arch

AUTHOR NAME: Fernando Alex Sierra-Linan

PUBLICATION YEAR: 2022

DESCRIPTION: The research proposes the development of a location-based mobile application for Plasma donor search (DONAPE), for which the mobile application provides a direct location-based channel between Plasma seekers and Plasma donation centers. Achieving to increase the number of donors, improve the place of origin (geographical location) of donors and improve the search time. They chose to use the agile Scrum method to develop the project prototype. This method has 5 phases: initiation, planning and estimation, implementation, review and retrospective and launch, for the development of this project. In web and mobile applications were developed to manage Plasma donation, allowing to register, schedule, receive notifications and access information, synchronizing Plasma donation centers with emergency centers, to verify the availability of Plasma needed and to send a request to the nearest Plasma donation center.

PAPER 4

TITLE: A Cross-Platform Plasma Donation Application with a Real-Time, Intelligent, and Rational Recommendation System

AUTHOR NAME: Rashik Rahman

PUBLICATION YEAR: 2021

DESCRIPTION: In this research work, they have designed a real-time, intelligent, and rational recommendation system using sentiment analysis of the user's feedback, response rate of the donor, and the current geo-location information and finally develop a cross-platform application for Plasma collection and distribution system. To process and generate features from the user feedback, they have designed a Bi-directional LSTM-based deep learning model. They chose the flutter framework to develop our cross platform applications. Firebase, a Google platform for mobile and web applications, has been used in the proposed application for authentication man. The quality of the recommendation of the potential donors has significantly improved. Moreover, they have conducted rigorous requirement analysis from real users and evaluated the performance of the application through both indoor and outdoor testing.

PAPER 5

TITLE: Preferences and features of a blood donation smart phone app:

multicenter mixed-methods study in Riyadh, Saudi Arabia

AUTHOR NAME: Afaf Ali Batis

PUBLICATION YEAR: 2021

DESCRIPTION: To identify the features and preferences of a blood donation smart phone app for blood donation centers and donors in Riyadh City, Saudi Arabia. This is a mixed-method study composed of a quantitative cross-sectional part (with donors, using a self-administered questionnaire), and a qualitative/quantitative part (with blood donation center staff, using semi-structured interviews). Data were collected between 15 November 2017 and 5 February 2018, from four blood donation centers in Riyadh City, Saudi Arabia. A descriptive analysis was used for the quantitative part and a thematic approach for the qualitative part.

2.2 REFERENCE

Several experiments have been carried out over the years by different groups of researchers. Here are some of the following groups:

- ways to keep your plasma healthy, Original Archived November 1, 2013, Accessed November 11, 2011.
- Paper 6- A Geo-Location based Mobile Service for Blood Donation during Medical Emergencies by Saurin Parikh, Preeti Kathiria Volume 88 – No.3, February 2014
- P. C. P. C. a. V. I. M. Yan, "Building a chatbot with server less computing," IBM watson research center, 2016.
- S. E. a. B. J. J. Short, ""Cloud Event Programming Paradigms: Applications and Analysis,"," 9th IEEE International Conference on Cloud Computing (CLOUD), pp. pp. 4 00-406, 2017

2.3 Problem Statement Definition

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A Donor	Donate plasma	it might spread diseases	I don't have knowledge	Afraid and fearful
PS-2	A Donee	Receive plasma	With this applicatio n make it faster	It takes a long time to find donors	Anxious

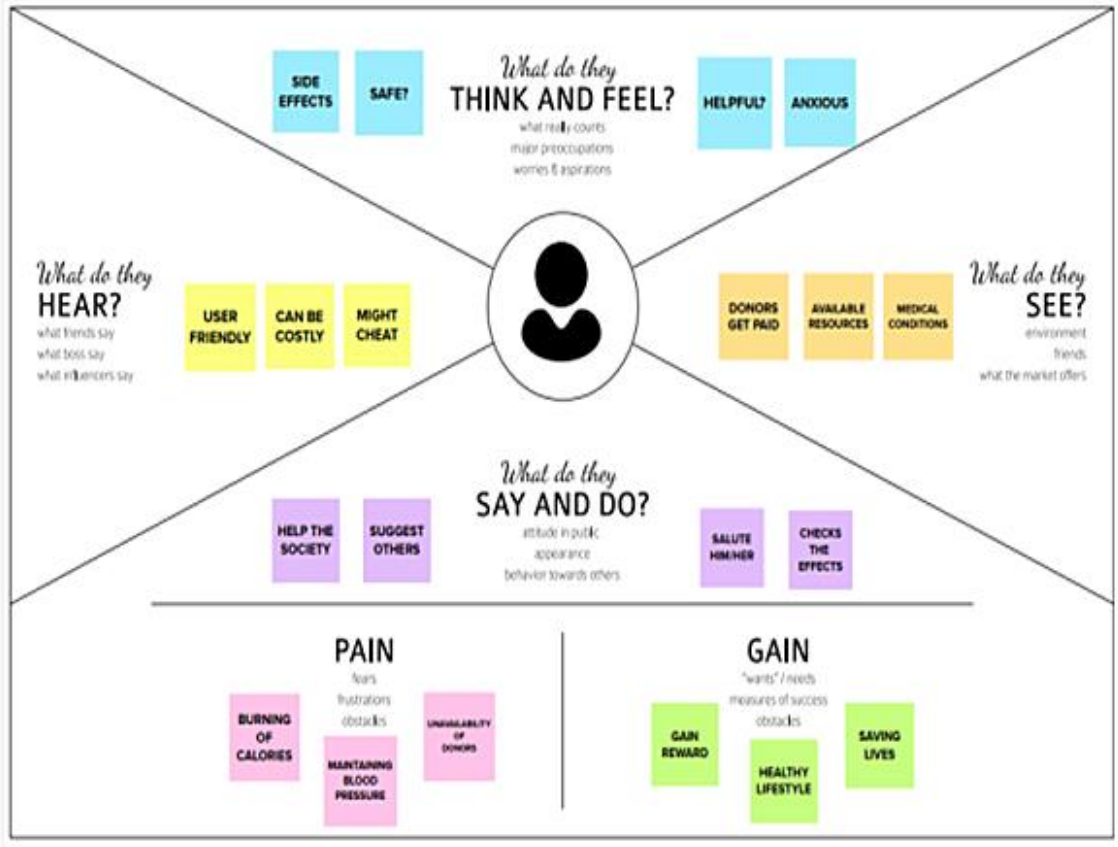
3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

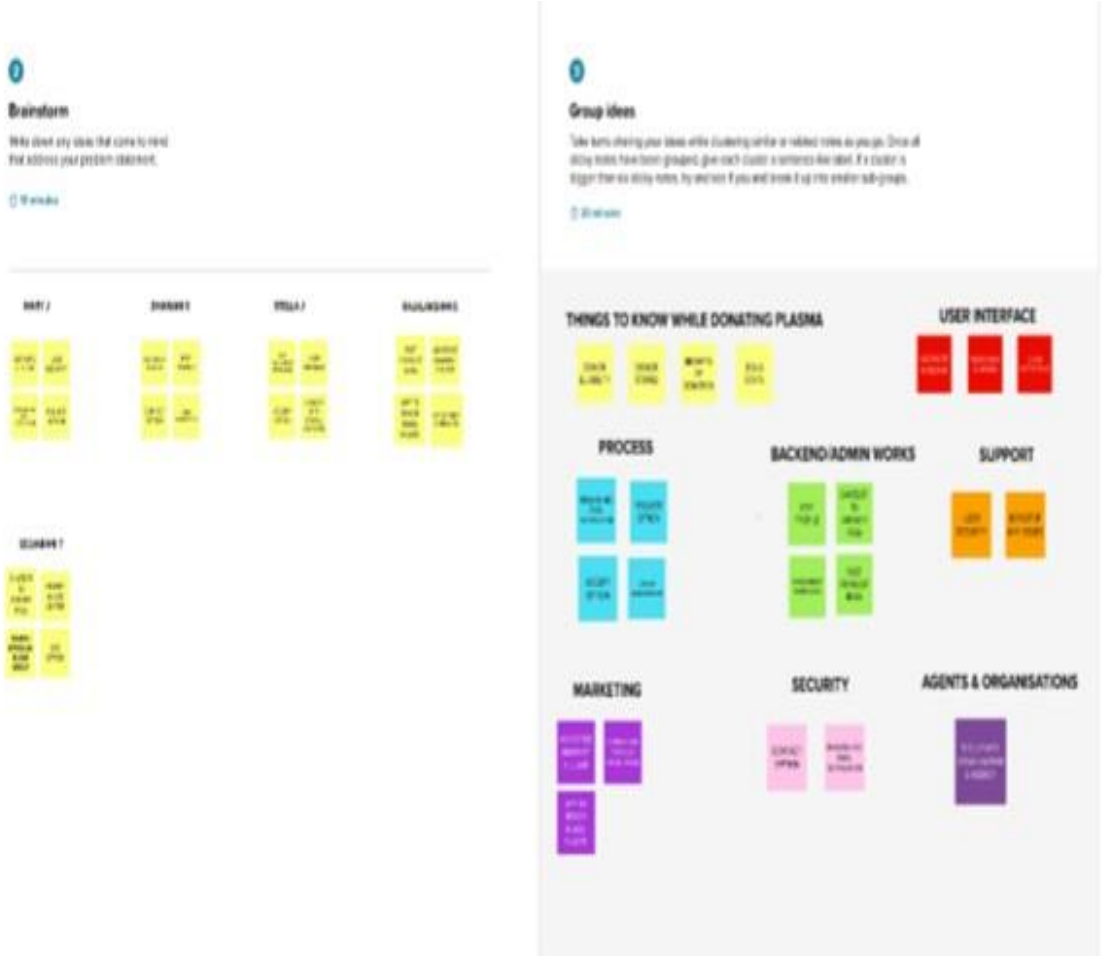
Empathy Map:

1

Build empathy and keep your focus on the user by putting yourself in their shoes.



3.2 Ideation & Brainstorming



3.3 Proposed Solution

S. No	Parameter	Description
1.	Problem Statement (Problem to be solved)	1. To help the plasma donor and seeker by developing a cloud-based application.

2.	Idea / Solution description	1. In day-to-day life requirement for plasma became high, especially during the COVID19 crisis. But the donor count was low. Saving the donor information and helping the needy by notifying the current donors would be a helping hand. It is very difficult to find the respective blood group donors when anyone is in need. Regarding the problem faced, an application is to be built which would take the donor details store them and inform them upon request. And also for plasma donation centre, it is easy to find donors.
3.	Novelty / Uniqueness	2. We help the donor to access the location confirmation emails after they get registered for the plasma donation of a blood centre which is nearby him/her. We Notify them by sending a and also we notify them once the appointment is fixed in the centre. Furthermore, the GPS map option is available to direct the donor to the centre.
4.	Social Impact / Customer Satisfaction	3. By using this application, the user will experience a user-friendly and responsive interface and they get satisfaction by saving thousands of people's life.
5.	Business Model (Revenue Model)	1. Donating Plasma with the help of an application makes our idea realistic. The user's information is encrypted. We maintain this app by automation for saving admin and user time. Users get profited as we take care of them even after the plasma donation by giving them hospitality details. Also, we use the Chabot to answer FAQs, as it helps the user to get immediate answers to their doubts.

6.	Scalability of the Solution	2. Whatever the requirements, the application provides a clear solution for the requirements. It can handle more users who use the application at the same time.

3.4 Problem Solution fit

Plasma Donor Application - Problem Solution Fit

1.CUSTOMER SEGMENT(S) **Who is your customer?** People who wish to donate plasma and Hospitals & Blood banks which needs plasma donors	**5.AVAILABLE SOLUTIONS** **Which solutions are available to the customers when they face the problem** Available solutions notify about the donors, patients and the availability of plasma & need for the plasma. The notification regarding the need for plasma was not send to the donors	**8.CHANNELS of BEHAVIOUR** **8.1 ONLINE What kind of actions do customers take online?** Registering for plasma donation and requesting for plasma will be carried out through online **8.2 OFFLINE What kind of actions do customers take offline?** Arrangements for plasma donation Awareness for more plasma donation
2. JOBS-TO-BE-DONE / PROBLEMS J&P **Which jobs-to-be-done (or problems) do you address for your customers?** Data collection should be monitored properly with donor's data security. Unawareness about the need of plasma donation. Demand for donors.	**6.CUSTOMER CONSTRAINTS** **What constraints prevent your customers from taking action or limit their choices of solutions?** Network Bandwidth Donor Health condition Lack of knowledge about app Unavailability of plasma	**9.PROBLEM ROOT CAUSE** **What is the real reason that this problem exists?** **What is the back story behind the need to do this job?** Lack of unawareness about the importance of plasma donation. Inability to find the donors at the time of emergency. Decrease in donors count
3. TRIGGERS TR **What triggers customers to act?** Volunteering interest and social responsibility towards society triggers the people to use this application	**7. BEHAVIOUR** **What does your customer do to address the problem and get the job done?** An unique ID will be provided for the donor's, in order to maintain their personal privacy. At the same time, an unique ID will be issued to the patient and the records will be monitored. Both donor and patient can access the application at ease	**10. YOUR SOLUTION SL** Donors will be searched with blood groups in our database, if needed. The volunteers can donate the blood with their interest and become donors by registering themselves. Stock monitoring will be done and updation happens at the same time. An application which will act as the intermediate between the hospital and donors and bridge the gap between them.
4. EMOTIONS: BEFORE / AFTER EM **How do customers feel when they face a problem or a job and afterwards?** Before : Hard to find the donors for plasma donation at the right time. After : Satisfactory feel and relaxed feel after getting the right donor		

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 Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via Message Confirmation via OTP
FR-3	Certifications	The certificate of appreciation and authentication will be provided for the donors.
FR-4	Statistical data	Plasma availability is provided on the page as statistics that will be useful for users.
FR-5	User Plasma Request	The recipient who needs plasma fill the request form in the web page. The confirmation mail has been sent when the request is submitted
FR-6	Searching/Reporting requirement	Users are able to use the search bar to search for information about.camps and others.
FR-7	Virtual Assistants	A virtual assistant is created to answer user questions about Plasma Donation. This will perform the function of a person in responding to user queries, where it will respond based on the information stored.

4.2 Non-Functional requirements

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

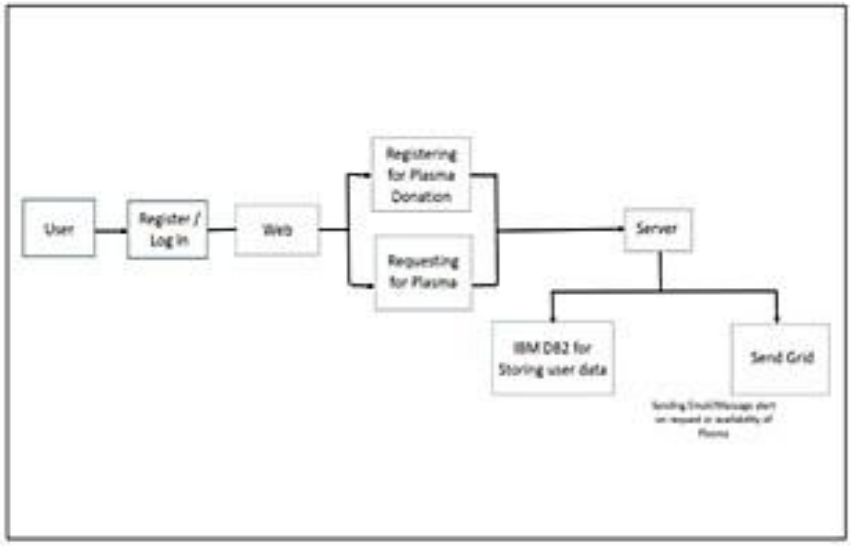
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	User friendly interface with easily accessible, well-looking and interactive chatbots.
NFR-2	Security	Data of donor and recipient should be saved in a secured manner.The user can only logged in using the correct password and username.
NFR-3	Reliability	The system should be built in such a way that it is reliable in its operations as well as to secure the sensitive details.
NFR-4	Performance	Users should have a proper internet connection

NFR-5	Availability	The system should have efficient active service. Must be available all times. In case of hardware or database corruption,backups of the data should be retrieved from the web application.
NFR-6	Scalability	The system should be scalable to handle a large number of users and should not get disrupted while using the system application.

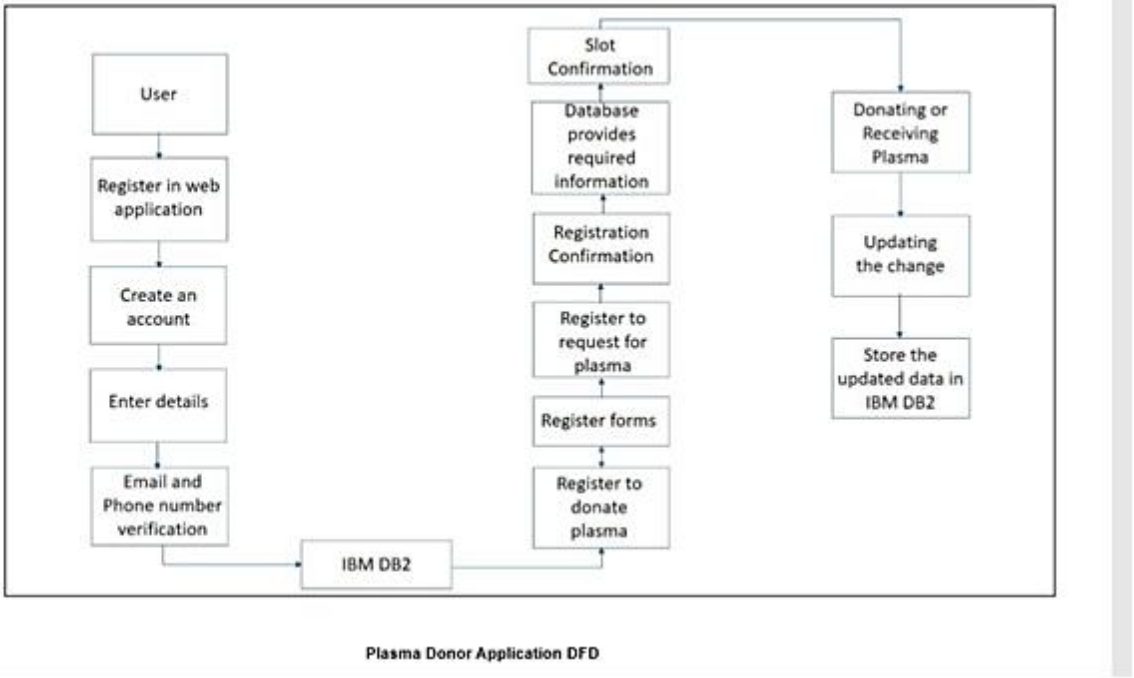
5. PROJECT DESIGN

5.1 Data Flow Diagrams

Data Flow Diagram : (Plasma Donor Application)



5.2 Solution & Technical Architecture



Plasma Donor Application DFD

5.2 Solution & Technical Architecture

Technical Architecture:

The deliverable shall include the architectural diagram as below.

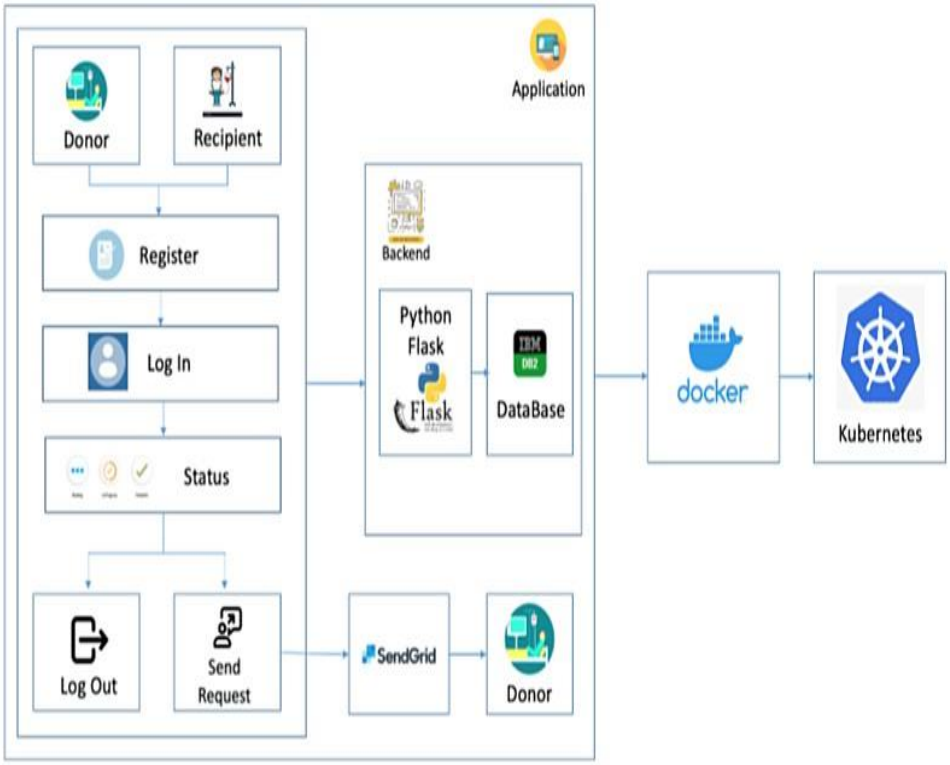


Table-1: Components & Technologies

S. no	Component	Description	Technology

1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML,CSS,JavaScript/AngularJs
2.	Application Logic-1	New User registers in the application by giving the genuine contact details which will be stored in the database.	Java,Flask,HTML,CSS
3.	Application Logic-2	Users login into the application by providing the username and password.	Flask,IBM DB2
4.	Application Logic-3	Stats page displays the blood unit count available and the number of donors available for each blood group	IBM Watson Assistant
5.	Application Logic-4	A request page that collects the name,contact number,gender and the blood group needed.Finally the request is sent to a donor whose blood group matches with the request.	Sendgrid
6.	Database	Characters,Integers,String,Long, Configurations	IBM DB2, MySQL
7.	Cloud Storage	Database service on cloud	IBM DB2, IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Authentication, used to store, manage and deploy container images.	Flask, Container registry
9.	External API-2	Sending request to donors	Sendgrid
10.	Infrastructure (Server / Cloud)	Application Deployment	Kubernetes, cloud foundry

Table-2:Application Characteristics

S.no	Characteristics	Description	Technology

1.	Open-Source Frameworks	List the open-source frameworks used	Python Flask
2.	Security Implementations	List all the security / access controls implemented , use of firewalls etc.	Doctor content Trust (DCT), Transport Layer Security(TLS), Container registry
3.	Scalable Architecture	Justifying the scalability of architecture (3 – tier, Micro-services) Kubernetes prevents hardware problems like downtime error.	Docker,Kubernetes cluster
4.	Availability	Use of load balancers, distributed servers. Kubernetes provide all time availability.	Kubernetes
5.	Performance	Application performance is improved by Docker	Docker

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story /Task	Acceptance criteria	Priority	Release
Admin	Registration	USN-1	User, I can register for the application by entering my email, password, and confirming my password.	I can control my online accountand dashboard.	Medium	Sprint-1
Co-Admin	Registration through Google account	USN-2	user, I can register for the application through google account	I can handle the waste collection.	High	Sprint-1
Admin	Confirmation	USN-3	As a user, I will receive confirmation email once I have registered for the application	I can take the shortest path to reachthe waste filled	Medium	Sprint-2

				routespecifie d.		
Donee	Search for donor	USN-4	Admin can access, view, modify, update all details of the plasma donor application	I can collectthe trach, pullit to the truck, and send it out.	Medium	Sprint-3
Software Engineer	Software testing And deployment	USN-5	Drawbacks we need to test the software before release.	All of these processes are under my control.	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	project & gathering information by referring the, technical papers, research publications etc.	26 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map in mural to capture the user Pains & Gains, Prepare list of problem statements	22 SEPTEMBER 2022
Ideation	Brain storming session and prioritize the top 4 ideas based on the feasibility & importance.	30 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the problem statement, idea, novelty, business model, social impact, scalability of solution	16 OCTOBER 2022

Problem Solution Fit	Prepare problem - solution fit document.	16 OCTOBER 2022
Solution Architecture	Prepare solution architecture document	17 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	20 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document	19 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	19 OCTOBER 2022
Technology Architecture	Prepare the technology Architecture diagram.	20 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project	20 OCTOBER 2022
Delivery of Sprint-1, 2, 3 &4	Develop & submit the developed code by testing it	WORK IN PROGRESS

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	User, I can register for the application by entering my email, password, and confirming my password.	13	High	3

Sprint-2	Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	13	High	3
Sprint-1	Registration through Google account	USN-3	user, I can register for the application through google account	8	Low	2
Sprint-3	Search for donor	USN-4	can view list represents a specific donor with donor details	13	Medium	2
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	13	High	2
Sprint-2	Dashboard	USN-6	As a user, I can log in into the application and view the dashboard for plasma information's	8	Medium	2
Sprint-2	Notification	USN-7	As a user,I can get notifications after register for plasma	13	High	3

			donation/nee dy			
Sprint-3	Database	USN-8	Admin can access, view, modify, update all details of the plasma donor application	20	High	4
Sprint-4	Software testing And deployment	USN-9	drawbacks we need to test the software before release.	13	High	4

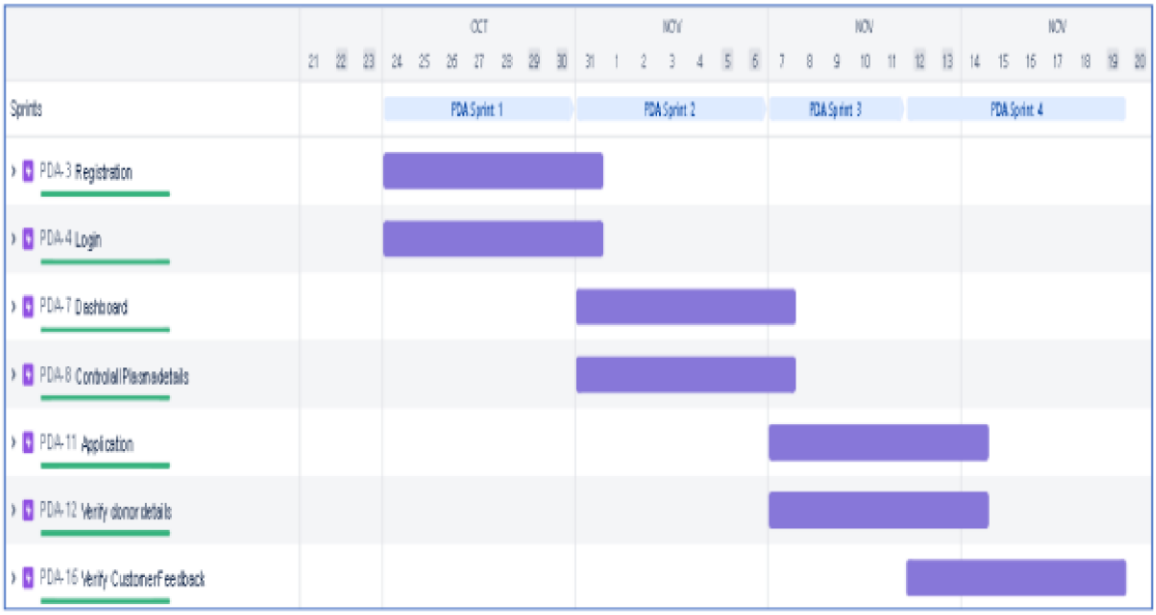
Project Tracker, Velocity & Burndown Chart

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	13	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	13	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 202	12 Nov 2022	13	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	8	19 Nov 2022

Velocity:

Sprint 1(AV) =2.16
Sprint 2(AV) =2.16
Sprint 3(AV) =2.16
Sprint 4(AV) =1.33

6.3 Reports from JIRA



7. CODING & SOLUTIONING

7.1 Feature 1

Python

- Python is a widely-used, interpreted, object-oriented, and high-level programming language with dynamic semantics, used for general purpose programming. It's everywhere, and people use numerous Python-powered devices on a daily basis, whether they realize it or not.
- Python was created by Guido van Rossum, and first released on February 20, 1991.
- Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, and Unix shell and other scripting languages.
- Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL)
- It is easy to learn – the time needed to learn Python is shorter than for many other languages; this means that it's possible to start the actual programming fast
- It is easy to use for writing new software – it's often possible to write code faster when using Python.
- It is easy to obtain, install and deploy – Python is free, open and multiplatform; not all languages can boast that.
- Programming skills prepare you for careers in almost any industry and are required if you want to continue to more advanced and higher paying software development and engineering roles.
- Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

7.2 Feature 2

Flask

- Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries
- It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.
- Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.
- Applications that use the Flask framework include Pinterest and LinkedIn.

7.3 Database Scheme

IBM Db2

- DB2 is a database product from IBM.
- It is a Relational Database Management System (RDBMS). DB2 is designed to store, analyze and retrieve the data efficiently.
- The DB2 product is extended with the support of Object-Oriented features and non-relational structures with XML.
- Provide a massively parallel processing (MPP) architecture Exploits Hive, HBase and Apache Spark concurrently for best-in-class analytic capabilities.

Kubernetes

- Kubernetes is an extensible, portable, and open-source platform designed by Google in 2014.
- It is mainly used to automate the deployment, scaling, and operations of the container-based applications across the cluster of nodes.
- Kubernetes helps to manage containerised applications in various types of physical, virtual, and cloud environments.

8. Testing

8.1 Test Case

- It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectation and does not fail in an unacceptable manner.
- There are various types of test. Each test type addresses a specific testing requirement

8.2 User Acceptance Testing

1.Purpose of Document The purpose of this document is to briefly explain the test coverage and open issues of the Plasma Donation Application project at the time of the release to User Acceptance Testing (UAT).

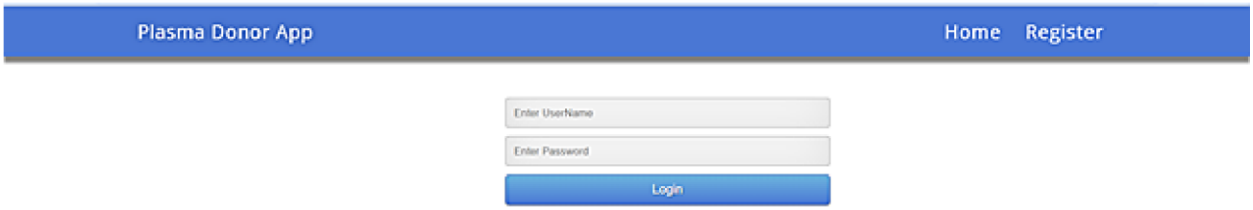
2. Defect Analysis This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

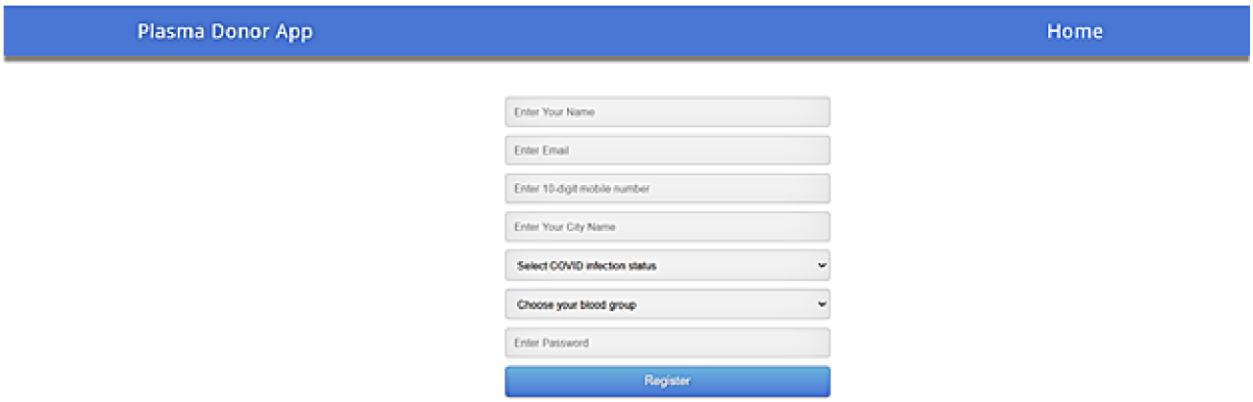
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Sub total
By Design	8	4	3	3	17
Duplicate	1	0	2	1	4
External	2	3	0	1	6
Fixed	10	2	5	18	35
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	3	2	1	6
Totals	1	12	13	25	7

9. RESULTS

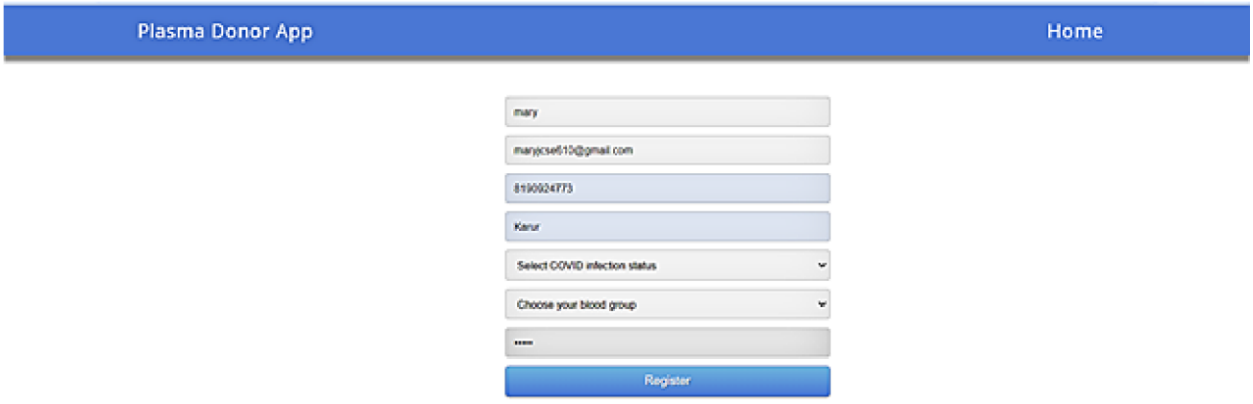
9.1 Performance Metrics

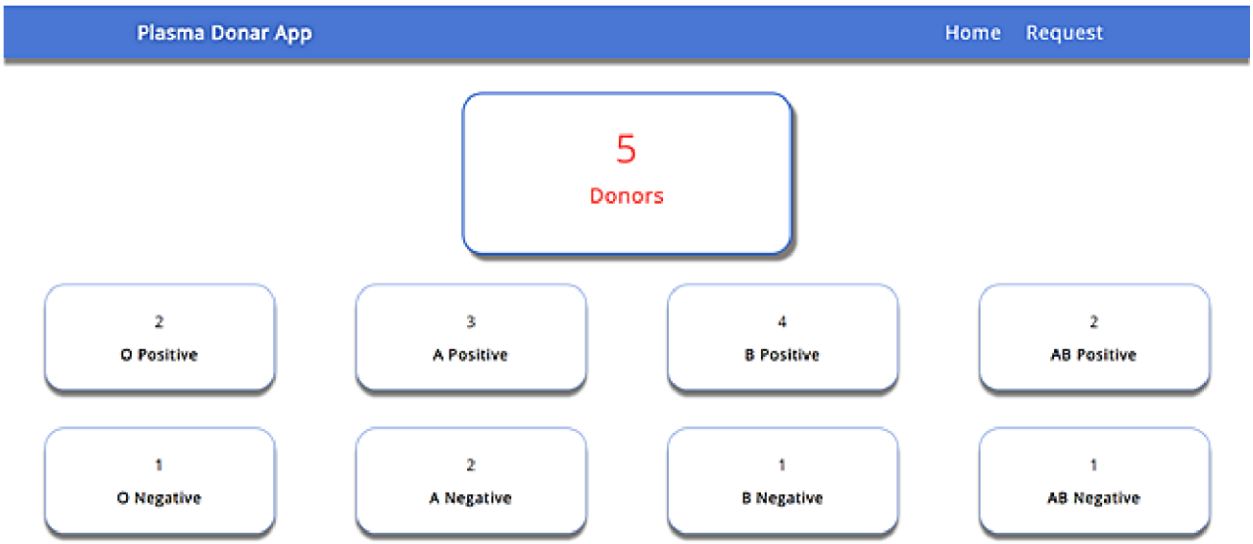
- Project metrics are used to track the progress and performance of a project.
- Monitoring parts of a project like productivity, scheduling, and scope make it easier for team leaders to see what's on track.
- As a project evolves, managers need access to changing
- deadlines or budgets to meet their client's expectation





127.0.0.1:8080/login





10. ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Simple user interface
- It alternatives the burden of coordinator to manage users and resources easily.
- Compared to all other web applications,it incorporates provisions for plasma donation.
- Easy for receipt to search for a donor,so for emergencies the donor cloud can be found easily.
- Usage of this application will greatly reduce time in selecting the right donor
- Attracts more users as it is available in the form of web applications instead of whatsapp groups.

DISADVANTAGES

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

11.CONCLUSION

Enhanced web application for plasma,has been developed to help the administrator to attract more donors and recipients and make user management and easy task.This web application will attract more user as it is user friendly and greatly reduces scalability issues especially in the case of donation.Not everyone in the world can donate.Thus,we have successfully designed and developed the web mobile application to ease the process of becoming a donor and recipient of plasma.

12.FUTURE SCOPE

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

13) APPENDIX

Source Code

```
# Project : Plasma Donor Application
# Team ID : PNT2022TMID48500
```

MAIN.py

```
from flask import Flask, render_template, request, redirect, url_for, session
import
ibm_db
import json
app = Flask(__name__)

conn =
ibm_db.connect("DATABASE=bludb;HOSTNAME=3883e7e4-18f5-4afe-be8c-
fa31c41761d2.bs2
io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31498;SECURITY=SSL;SSLServerCerti
```

```
ficate=DigiCertGlobalRootCA.crt;UID=htz00980;PWD=VYbUFN7Gw0LqsmZ4",",")

@app.route('/registration')
def home():

    return render_template('register.html')


@app.route('/register',methods=['POST'])
def register():
  x = [x for x in request.form.values()]

    print(x)
    name=x[0]
    email=x[1]
    phone=x[2]
    city=x[3]
    infect=x[4]
   blood=x[5]

    password=x[6]
    sql = "SELECT * FROM user WHERE email =?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,email)
     ibm_db.execute(stmt)

    account = ibm_db.fetch_assoc(stmt)
    print(account)
    if account:
        return render_template('register.html', pred="You are already a member, please login using
your
details")
    else:


        prep_stmt = ibm_db.prepare(conn, insert_sql)

        ibm_db.bind_param(prep_stmt, 1, name)

       ibm_db.bind_param(prep_stmt, 2, email)
       ibm_db.bind_param(prep_stmt, 3, phone)
       ibm_db.bind_param(prep_stmt, 4, city)
       ibm_db.bind_param(prep_stmt, 5, infect)
       ibm_db.bind_param(prep_stmt, 6, blood)
```

```
        ibm_db.bind_param(prep_stmt, 7, password)
        ibm_db.execute(prep_stmt)
        return  render_template('register.html',  pred="Registration  Successful,  please  login  using
your details")
```

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

@app.route('/loginpage',methods=['POST'])
def loginpage():

    user = request.form['user']
    passw = request.form['passw']
    sql = "SELECT * FROM user WHERE email =? AND password=?"
    stmt = ibm_db.prepare(conn, sql)
     ibm_db.bind_param(stmt,1,user)

    ibm_db.bind_param(stmt,2,passw)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print (account)
```

```
    print(user,passw)
    if account:
            return redirect(url_for('stats'))
    else:
        return render_template('login.html', pred="Login unsuccessful. Incorrect username /

password !")



@app.route('/stats')
def stats():

    '''sql = "SELECT blood FROM user group by blood"
s    stmt = ibm_db.prepare(conn, sql)
    ibm_db.execute(stmt)
```

```
    count = ibm_db.fetch_assoc(stmt)

    print(count)'"

    return render_template('stats.html',b=5,b1=2,b2=3,b3=4,b4=2,b5=1,b6=2,b7=1,b8=1)


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


@app.route('/requested',methods=['POST'])

def requested():
    bloodgrp = request.form['bloodgrp']
    address = request.form['address'] print(address)

   sql = "SELECT * FROM user WHERE blood=?"

    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,bloodgrp)
    ibm_db.execute(stmt) data =
    ibm_db.fetch_assoc(stmt)
      msg = "Need Plasma of your blood group for: "+address

    while data != False:
        print ("The Phone is : ", data["PHONE"])
url="https://www.fast2sms.com/dev/bulk?authorization=xCXuwWTzyjOD2ARd1EngbH3a7tKI
q5
PklJ8YSf0Lh4FQZecs9iNI1dSvuqprxFwCKYJXA5amQkBE36Rl&sender_id=FSTSMS&mess
ag
e="+msg+"&language=english&route=p&numbers="+str(data["PHONE"])
        result=requests.request("GET",url)
        print(result)
        data = ibm_db.fetch_assoc(stmt)
      return render_template('request.html', pred="Your request is sent to the concerned people.")



if __name__ == "__main__":
```

```
    app.run(host='0.0.0.0', port=8080)
```

Login.html

```
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>Plasma Donor App</title>
    <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
    <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
    <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
    <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
rel='stylesheet' type='text/css'>
    <link rel="stylesheet" href="{{ url_for('static', filename='style1.css') }}">
    <!link rel="stylesheet" href="style.css">

<style>
.login{
top: 20%;
}
</style>
</head>

<body>
<div class="header">
<div>Plasma Donor App</div>
    <ul>



        <li><a href="/registration">Register</a></li>
        <li><a class="active" href="/login">Home</a></li>
    </ul>
</div>
 <div class="login" >
        <div>
```

```
      </div>


    <!-- Main Input For Receiving Query to our ML -->
   <form action="{{ url_for('loginpage')}}"method="post">
      <input type="text" name="user" placeholder="Enter UserName" required="required"
style="color:black"
/>


      <input type="password" name="passw" placeholder="Enter Password" required="required"
style="color:black" />
      <button type="submit" class="btn btn-primary btn-block btn-large">Login</button>

   </form>
<br><br>

<div style="color:black">
 {{ pred }}</div>
 </div>

</body>
</html>
```

Register.html

```
<!DOCTYPE
html>


<html
>


<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
```

```
 <meta charset="UTF-8">
 <title>Plasma Donor App</title>
   <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>


   <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>


    <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
rel='stylesheet' type='text/css'>
    <link rel="stylesheet" href="{{ url_for('static', filename='style1.css') }}">
    <!link rel="stylesheet" href="style.css">
```

```
<style>
.login{
top: 20%;
}
</style>
</head>

<body>
<div class="header">
<div>Plasma Donor App</div>
   <ul>



      <li><a class="active" href="/login">Home</a></li>
   </ul>
</div>
 <div class="login">

    <!-- Main Input For Receiving Query to our ML -->
   <form action="{{ url_for('register')}}"method="post">
      <input type="text" name="name" placeholder="Enter Your Name" required="required"
style="color:black"/>
      <input type="email" name="email" placeholder="Enter Email" required="required"
style="color:black"/>
      <input type="text" name="phone" placeholder="Enter 10-digit mobile number"
required="required" style="color:black"/>
      <input type="city" name="city" placeholder="Enter Your City Name" required="required"
style="color:black"/>
      <select name="infect">

                  <option value="select" selected>Select COVID infection status</option>
                  <option value="infected">Infected</option>
                  <option value="uninfected">Uninfected</option>
      </select>
      <select name="blood">
                  <option value="select" selected>Choose your blood group</option>
```

```
                <option value="O Positive">O Positive</option>
                <option value="A Positive">A Positive</option>
                <option value="B Positive">B Positive</option>
                <option value="AB Positive">AB Positive</option>
                <option value="O Negative">O Negative</option>
               <option value="A Negative">A Negative</option>

                <option value="B Negative">B Negative</option>
                <option value="AB Negative">AB Negative</option>
      </select>
      <input type="password" name="passw" placeholder="Enter Password" required="required"
style="color:black"/>
    <button type="submit" class="btn btn-primary btn-block btn-large">Register</button>



  </form>


 <br><br>
<div style="color:black">
 {{            pred
 }}</div>


 </div>




</body>
</html>
```

Request.html

```
<!DOCTYPE html>
<html >
```

```
<! --From https://codepen.io/frytyler/pen/EGdtg--
>


<head>


  <meta charset="UTF-8">
  <title>Plasma Donor App</title>
    <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
    <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
    <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
     <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'


rel='stylesheet' type='text/css'>
```

```
    <link rel="stylesheet" href="{{ url_for('static', filename='style1.css') }}">
    <!link rel="stylesheet" href="style.css">


<style>
.login{
top: 20%;


}
</style>
</head>

<body>
<div class="header">


<div>Plasma Donor App</div>
    <ul>
```

```
      <li><a href="/requester">Request</a></li>
     <li><a href="/registration">Register</a></li>

    <li><a class="active" href="/login">Home</a></li>

   </ul>
</div>
 <div class="login">
      <div>
      </div>


    <!-- Main Input For Receiving Query to our ML -->
   <form action="{{ url_for('requested')}}"method="post">
      <select name="bloodgrp">
                  <option value="select" selected>Choose your blood group</option>
                  <option value="O Positive">O Positive</option>
                <option value="A Positive">A Positive</option>

                  <option value="B Positive">B Positive</option>
                  <option value="AB Positive">AB Positive</option>
                  <option value="O Negative">O Negative</option>
                  <option value="A Negative">A Negative</option>
                  <option value="B Negative">B Negative</option>
                <option value="AB Negative">AB Negative</option>

      </select>
      <textarea rows="4" placeholder="Enter the address" required="required"
style="color:black" name="address"></textarea>
      <!input type="textarea" name="address" rows="4" placeholder="Enter the address"
required="required" style="color:black" />
```

```
    <button type="submit" class="btn btn-primary btn-block btn-large">Submit the


request</button>


   </form>


 <br><br>
<div
style="color:black">


 {{ pred }}</div>


 </div>


</body>
</html>
```

Stats.html

```
<!DOCTYPE
html>


<html
lang="en">


<head>


  <title>Plasma Donar App</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
```

```
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">


 <script
 src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>


 <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
 <link rel="stylesheet" href="{{ url_for('static', filename='style1.css') }}">
 <!link rel="stylesheet" href="style.css">
</head>
```

```
<style>
      .big{
      top:70;
      background-color:white; margin-
      top:80px;
     margin-left:550px;


      margin-right:550px;
      height:200px;
      border-radius: 25px;
      border: 3px solid #4a77d4; box-
      shadow: 6px 8px 4px grey;
      text-align:center;


      }
      .row{


      height:150px;


     }
```

```
.col{

    margin:10px;
    margin-left:50px;
    margin-right:50px; border-
    radius: 25px;
    border: 1px solid #4a77d4;

    box-shadow: 0px 8px 4px grey;

    text-align:center;
 }
 .ext{
 margin-top:25px; line-
 height:40px;
 }

 .ext1{
 margin-top:40px;
 line-height:50px;
 font-size:25px;
 color:#f95450;
 }
```

```
</style>
<body>

<div class="container-fluid">
<div class="header">
```

```
<div><b>Plasma Donar App</b></div>
<ul>
      <li><a href="/requester">Request</a></li>
      <li><a class="active" href="/login">Home</a></li>
   </ul>
</div>

 <br>
 <div class="big">
   <div class="box">
      <div class="ext1"><font size="20px">{{b}}</font><br><b>Donors</b></div>
   </div>
 </div>

 <br>

 <div class="row">
   <div class="col" >
      <div class="ext">{{b1}}<br><b>O Positive</b></div>
   </div>
   <div class="col" >

  <div class="ext">{{b2}}<br><b>A Positive</b></div>

   </div>
   <div class="col" >
      <div class="ext">{{b3}}<br><b>B Positive</b></div>
   </div>
   <div class="col" >
  <div class="ext">{{b4}}<br><b>AB Positive</b></div>

   </div>
```

```
 </div>
 <br>
 <div class="row">
  <div class="col" >
  <div class="ext">{{b5}}<br><b>O Negative</b></div>

  </div>
  <div class="col" >
     <div class="ext">{{b6}}<br><b>A Negative</b></div>
  </div>
  <div class="col" >
  <div class="ext">{{b7}}<br><b>B Negative</b></div>

  </div>
  <div class="col" >
     <div class="ext">{{b8}}<br><b>AB Negative</b></div>
  </div>
 </div>
</div>

</body>
</html>
```

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-38455-1660380885

Video Demo Link: https://drive.google.com/file/d/1d8qzDnG9kd1Edtb2S-V_l2JYs8YTmcSh/view?usp=sharing

Team Member:

Hariharan M (Team Leader)
Anisha R (Team Member-1)
Pradeep Kumar R (Team Member-2)

Sujeevan B (Team Member-3)