Team ID	**PNT2022TMID22368**
Project Name	**Plasma Donor Application**

TABLE OF CONTENTS

ABSTRACT

During COVID-19 crisis, Our government and health care professionals are trying their best to help the patients suffering from COVID-19. Scientists are trying to discover a vaccine to cure people affected with coronavirus. There is a scientific way from which we can help to lower the death ratio or help the COVID 19 affected person. Plasma therapy is an experimental approach to treat COVID-positive patients and help them recover faster. But, in this situation, it is difficult for a patient to find a plasma donor as everybody can’t donate plasma. In regard to the problem faced, an application is to be built which would take the donor details, store them and inform them upon a request.

1. INTRODUCTION

1.1 Project Overview:

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

The necessity of blood has become a significant concern in the present context all over the world. Due to a shortage of blood, people couldn’t save themselves or their friends and family members. A bag of blood can save a precious life. Statistics show that a tremendous amount of blood is needed yearly because of major operations, road accidents, blood disorders, including Anemia, Hemophilia, and acute viral infections like Dengue, etc. Approximately 85 million people require single or multiple blood transfusions for treatment. Voluntary blood donors per 1,000 population of some countries are quite promising, such as Switzerland (113/1,000), Japan (70/1,000), while others have an unsatisfying result like India has 4/1,000, and Bangladesh has 5/1000. Recently a life-threatening virus, COVID-19, spreading throughout the globe, which is more vulnerable for older people and those with pre-existing medical conditions. For them, plasma is needed to recover their illness. Our Purpose is to build a platform with clustering algorithms which will jointly help to provide the quickest solution to find blood or plasma donor. Closest blood or plasma donors of the same group in a particular area can be explored within less time and more efficiently

1.2 Purpose:

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.

Only a small number of people living in the U.S. who are eligible to donate blood or source plasma actually donate. What's important is that we encourage all forms of donation from those who are eligible, so that they may contribute life-saving blood and source plasma to those in need.

The plasma protein therapeutics industry supports volunteerism donation in all of its forms. Source plasma donation and blood donation are critically important activities that contribute to saving lives. Source plasma and recovered plasma are used to produce therapies that treat people with rare, chronic diseases and disorders such as primary immunodeficiency, hemophilia and a genetic lung disease, as well as in the treatment of trauma, burns and shock. Whole blood donations most often are used locally in hospitals for transfusions required during surgery or other medical treatment. Find a donation center near you!

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.

Doctors can use plasma to treat different kinds of serious health problems. Some of the elements in plasma, including the antibodies and chemicals that help your blood to clot, can help in medical emergencies like burns and trauma.

Other things that plasma donation is good for include:

1. Developing treatments.
2. Cancer.
3. Transplant surgery.
4. Hemophilia.

2. LITERATURE SURVEY

2.1 Existing problem:

People have to find them physically by visiting hospitals register book an 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 persons health gets more worsened. This is an expensive and will not work as effectively at emergency situations.

1. Rishab Chakrabarti, Prof. S.M. Chitalkar – "Lifesaver E-Blood Donation App Using Cloud", 2020: Reduction in the error of blood bank using most eligible donor method. Direct Communication Between donor and the person in need of blood During the Emergency situation. However, this paper has the drawback that the user-provided informations is still unconfirmed.

2. A. Meiyappan, K. Loga Vignesh, R. Prasanna, T. Sakthivel – "D'WORLD: Blood Donation App Using Android", 2019: When the giver gives the blood, it will naturally evacuate the contributor detail for next three months. It additionally confirms with the Department of Health and Welfare to guarantee the benefactor medical case history. However, this has the drawback that in order to utilize this program, the user must have a device running the Android operating system and a live internet connection.

3. Ashlesha C. Adsul, V. K. Bhosale, R. M. Autee – "Automated blood bank system using Raspberry PI", 2018: When there is urgent need for blood then If this model is adopted the caller is immediately connected to the donor. However, dealing with the phone users is a drawback.

2.2 Reference:

1. Guo, Weijin; Hansson, Jonas; van der Wijngaart, Wouter (2020). "Synthetic Paper Separates Plasma from Whole Blood with Low Protein Loss". Analytical Chemistry.

2. P. C. P. C. a. V. I. M. Yan, “Building a chatbot with serverless computing,” IBM watson research center, 2016.

3. Tripathi S, Kumar V. Prabhakar A. Joshi S. Agrawal A (2015). "Passive blood plasma separation at the microscale a review of design principles and microdevices".

4. Android Blood Donor Life Saving Application in Cloud Computing by T. Hilda Jenipha, R. Backiyalakshmi Volume-03, Issue-02, pp-105- 108, 2014

5. Tuskegee University (May 29, 2013). "Chapter 9 Blood". tuskegee edu. Archived from the original on December 28, 2013.

6. Dennis O'Neil "Blood Components". Palomar College. Archived from the original on June 5, 2013.

7. "Ways to Keep Your Blood Plasma Healthy". Archived from the original on November 1, 2013. Retrieved November 10, 2011.

8. Arif. M. Sreevas. S. Nafseer. K. and Rahul. R. (2012), 'Automated online Blood bank database', India Conference (INDICON), Annual IEEE, Print ISBN: 978-1- 4673-2270-6, pp. 012 - 017.

2.3 Problem Statement Definition:

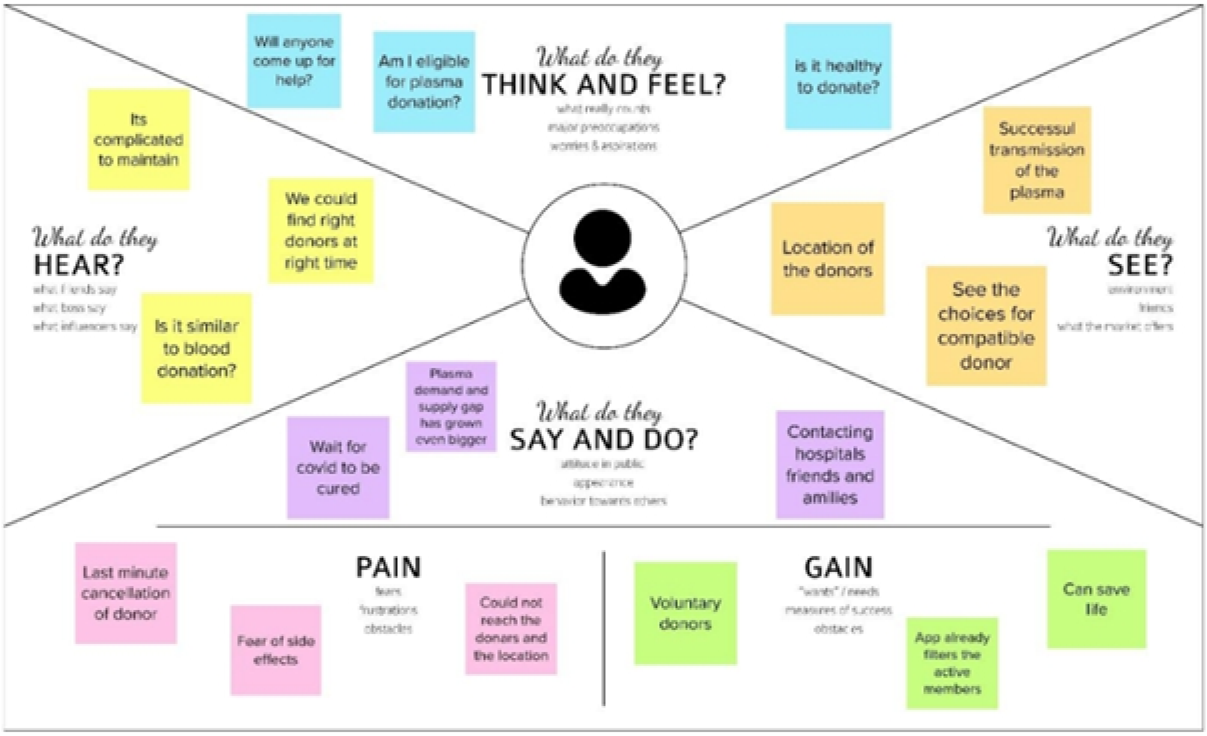
People who need plasma are increasing day by day. People who have diseases like trauma, burn, shock patients ,as well as peoples with severe liver disease or multiple clotting factors deficiencies people who have gotten into accidents and run out of plasma need constant supply of plasma to sustain their life and there is not enough plasma available for them. It is not that people do not want to donate plasma, but because they have no idea where they can donate.

It is important for the people who are excited to donate, but yet are very busy, to be sure where and when they can donate ,and therefore We are designing a system which contains all the information regarding plasma donation camps ongoing in a particular area so that people who want to donate plasma will get information regarding these camps. Our System is a web application which aims to serve as a communication tool between plasma Donation camp Organizers and plasma donors. To become a member of the system, donors need to create their profile by providing the information like name, blood group, email address, password, age factor(age restriction) and exact location from “Google Map”. In order to find out the exact location of a donor, Google Map is integrated with this application. The web application lways keeps updating the location of a donor. As a result, the system can automatically keep showing the nearby plasmadonation Camps to the registered donor wherever they go, and donors can easily get the idea of nearby plasma donation camps. Also, users can get information regarding the type of Plasma or blood which is available and information of past as well as future events.

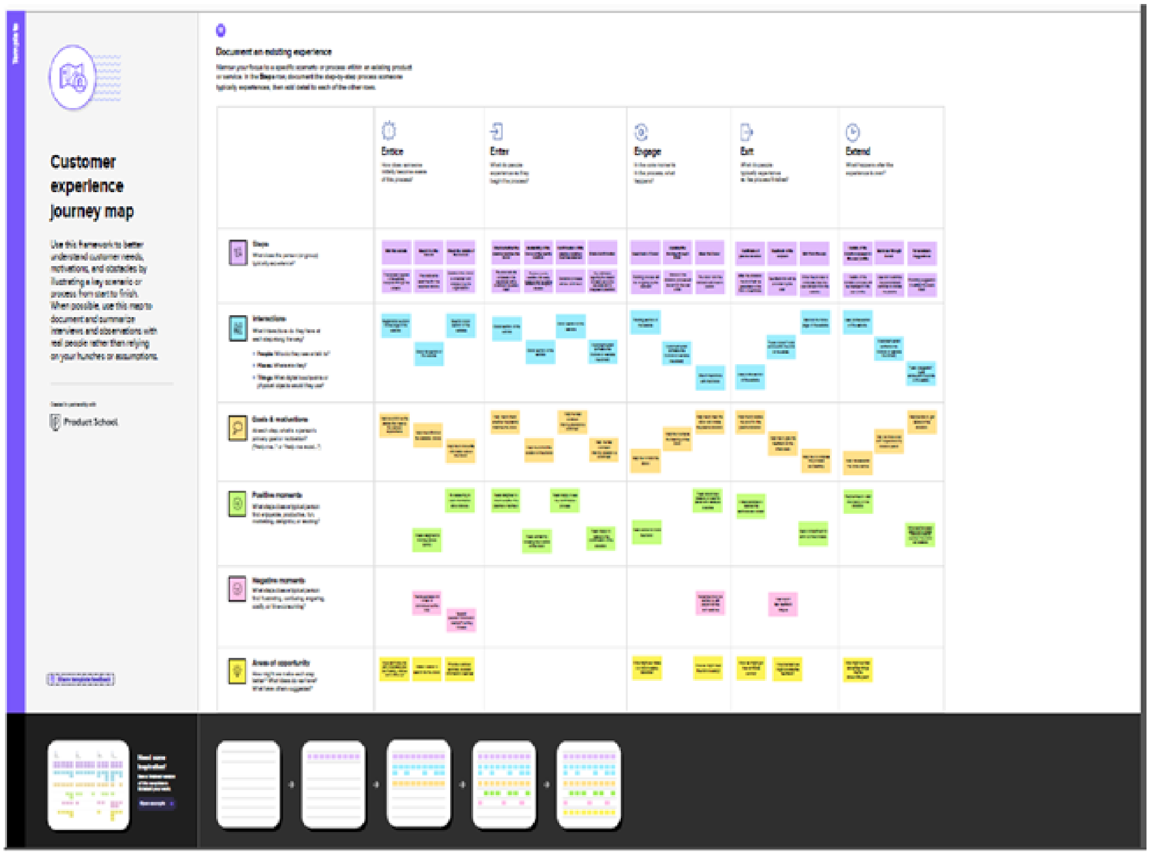
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes.It is a useful tool to helps teams better understand their users.Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.


Will anyone come up for help?
Am I eligible for plasma donation?
What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
is it healthy to donate?
Its complicated to maintain
We could find right donors at right time
What do they
HEAR?
what friends say
what boss say
what influencers say
Is it similar to blood donation?
Successul transmission of the plasma
Location of the donors
What do they
SEE?
environment
friends
what the market offers
See the choices for compatible donor
Plasma demand and supply gap has grown even bigger
Wait for covid to be cured
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
Contacting hospitals friends and amilies
Last minute cancellation of donor
Fear of side effects
PAIN
fears
frustrations
obstacles
Could not reach the donars and the location
Voluntary donors
GAIN
"wants" / needs
measures of success
obstacles
App already filters the active members
Can save life

3.2 Ideation & Brainstorming:



Customer experience journey map

Use this framework to better understand customer needs, motivations, and obstacles by illustrating a key scenario or process from start to finish. When possible, use this map to document and summarize interviews and observations with real people rather than relying on your hunches or assumptions.

Product School

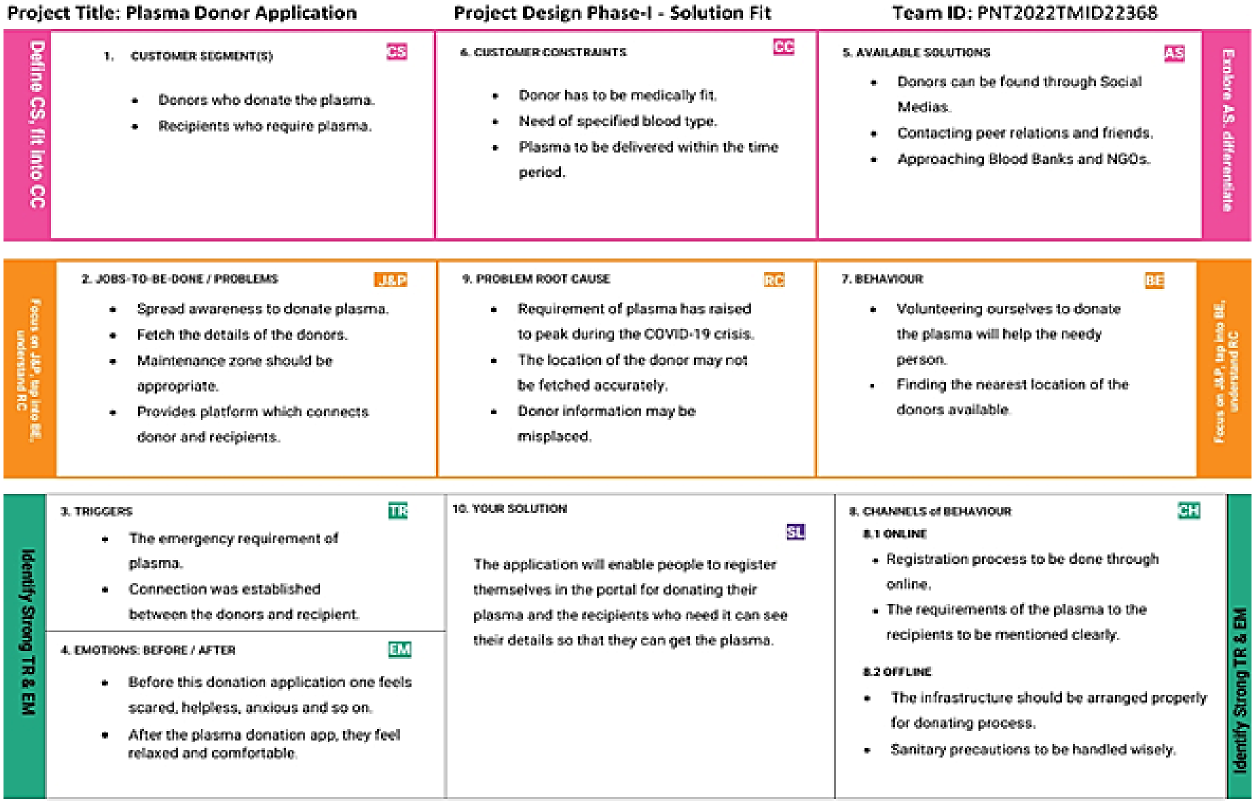
Document an existing experience

Narrow your focus to a specific scenario or process within an existing product or service. In the **Steps** row, document the step-by-step process someone typically experiences, then add detail to each of the other rows.

3.3 Proposed Solution:

S.N0.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To create an application for people who want to donate their plasma for the people who need it.
2.	Idea / Solution description	The application will enable people to register themselves in the portal for donating their plasma and the recipients who need it can see their details so that they can get the plasma.
3.	Novelty / Uniqueness	The application will be easy to use and it's connects the donor and the recipient easily without any issues.
4.	Social Impact / Customer Satisfaction	The application seamlessly connects the donor and the person who needs it hassle-free.
5.	Business Model (Revenue Model)	The application is free to use and it comes under healthcare domain. It helps people who want to donate plasma to the people who need it .
6.	Scalability of the Solution	The application is useful for the people who need plasma for their health concerns.

3.4 Problem Solution Fit:



Project Title: Plasma Donor Application

Project Design Phase-I - Solution Fit

Team ID: PNT2022TMID22368

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

- Donors who donate the plasma.
- Recipients who require plasma.

6. CUSTOMER CONSTRAINTS CC

- Donor has to be medically fit.
- Need of specified blood type.
- Plasma to be delivered within the time period.

5. AVAILABLE SOLUTIONS AS

- Donors can be found through Social Medias.
- Contacting peer relations and friends.
- Approaching Blood Banks and NGOs.

Explore AS, differentiate

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

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

- Spread awareness to donate plasma.
- Fetch the details of the donors.
- Maintenance zone should be appropriate.
- Provides platform which connects donor and recipients.

9. PROBLEM ROOT CAUSE RC

- Requirement of plasma has raised to peak during the COVID-19 crisis.
- The location of the donor may not be fetched accurately.
- Donor information may be misplaced.

7. BEHAVIOUR BE

- Volunteering ourselves to donate the plasma will help the needy person.
- Finding the nearest location of the donors available.

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

Identify Strong TR & EM

3. TRIGGERS TR

- The emergency requirement of plasma.
- Connection was established between the donors and recipient.

4. EMOTIONS: BEFORE / AFTER EM

- Before this donation application one feels scared, helpless, anxious and so on.
- After the plasma donation app, they feel relaxed and comfortable.

10. YOUR SOLUTION SL

The application will enable people to register themselves in the portal for donating their plasma and the recipients who need it can see their details so that they can get the plasma.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

- Registration process to be done through online.
- The requirements of the plasma to the recipients to be mentioned clearly.

8.2 OFFLINE

- The infrastructure should be arranged properly for donating process.
- Sanitary precautions to be handled wisely.

Identify Strong TR & EM

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements:

1. Sign up as user
2. Confirmation mail received by user after registered
3. Register the donor by himself
4. Login of admin
5. Change personal, contact details by donor himself
6. Change personal, contact details by admin
7. Access Chat Bot by User
8. Search for compatible plasma
9. Alert the Donor through notification and email
10. Send plasma donation details
11. Send plasma request details

4.2 Non-Functional Requirement:

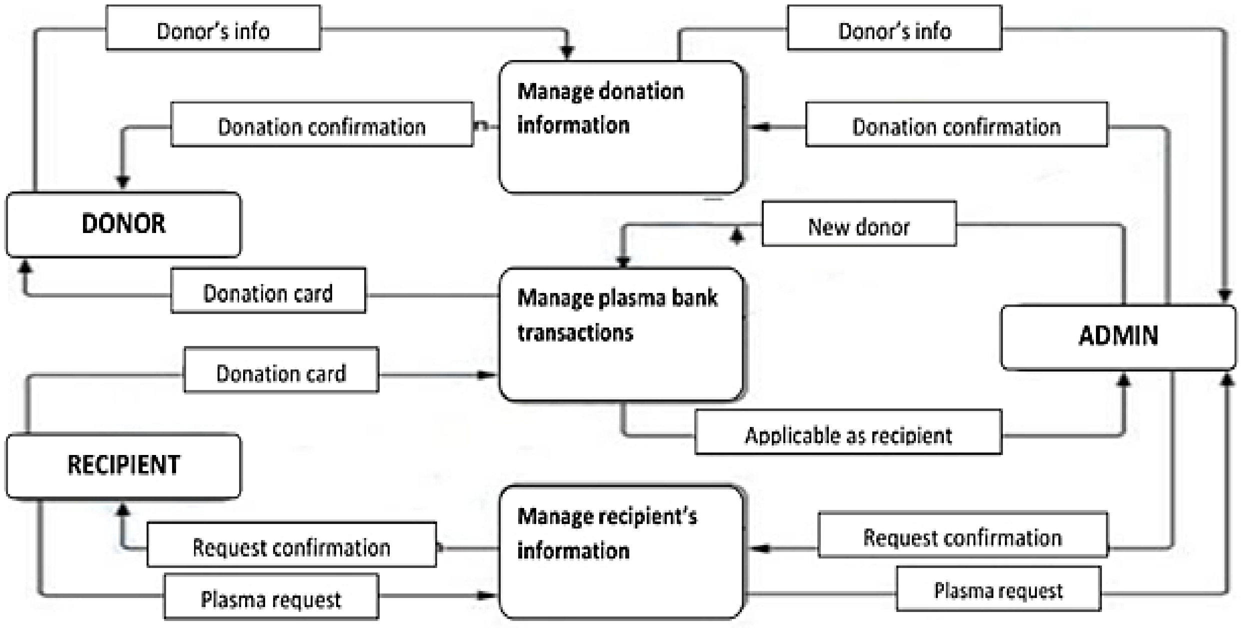
1. **Performance:** Application should be designed and developed in such a way that it should not utilize too many resources.
2. **Usability:** Usability defines how well the application meets the requirements of the user and consumer by being intuitive, easy to localize and globalize, providing good access for disabled users, and resulting in a good overall user experience.
3. **Secure:** Login system should be safe and secure.

5. PROJECT DESIGN

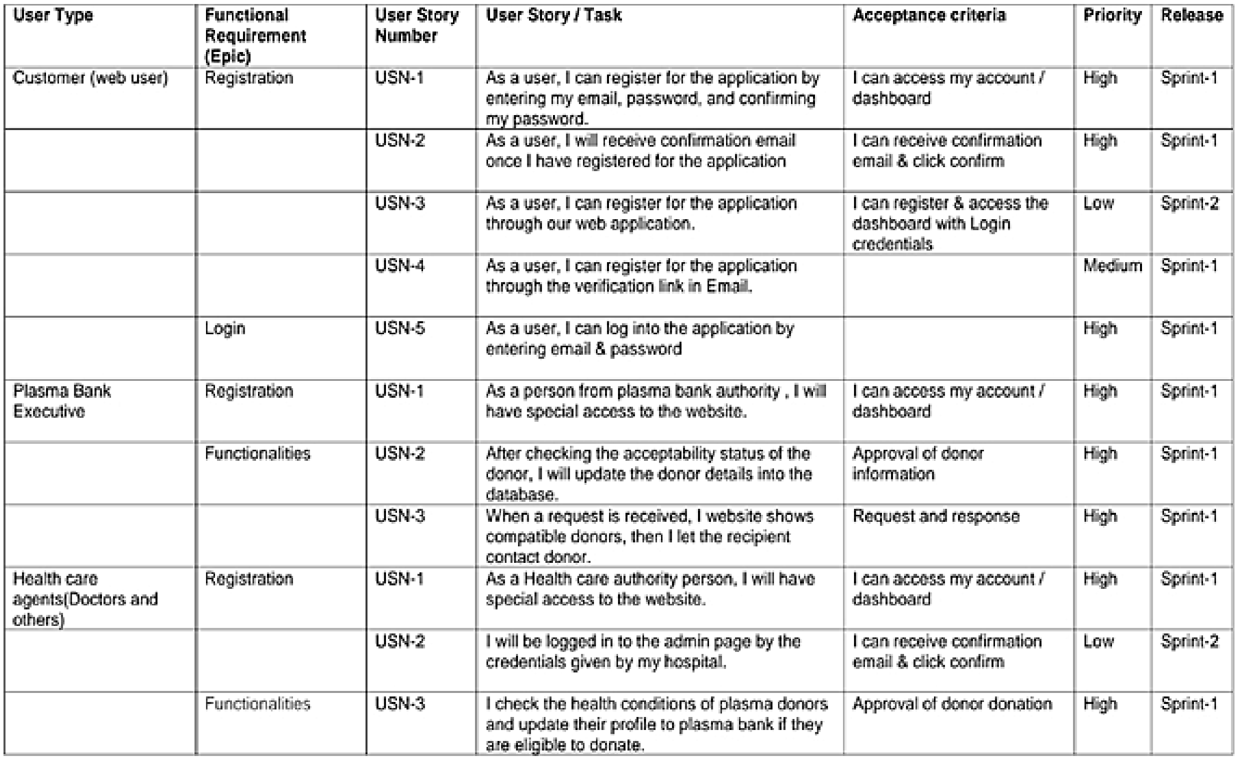
5.1 Data Flow Diagram & User Stories:

Data Flow Diagram:

A Data Flow Diagram(DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirements graphically. It shows how data enters and leaves the system, what changes the information and where data is stored.



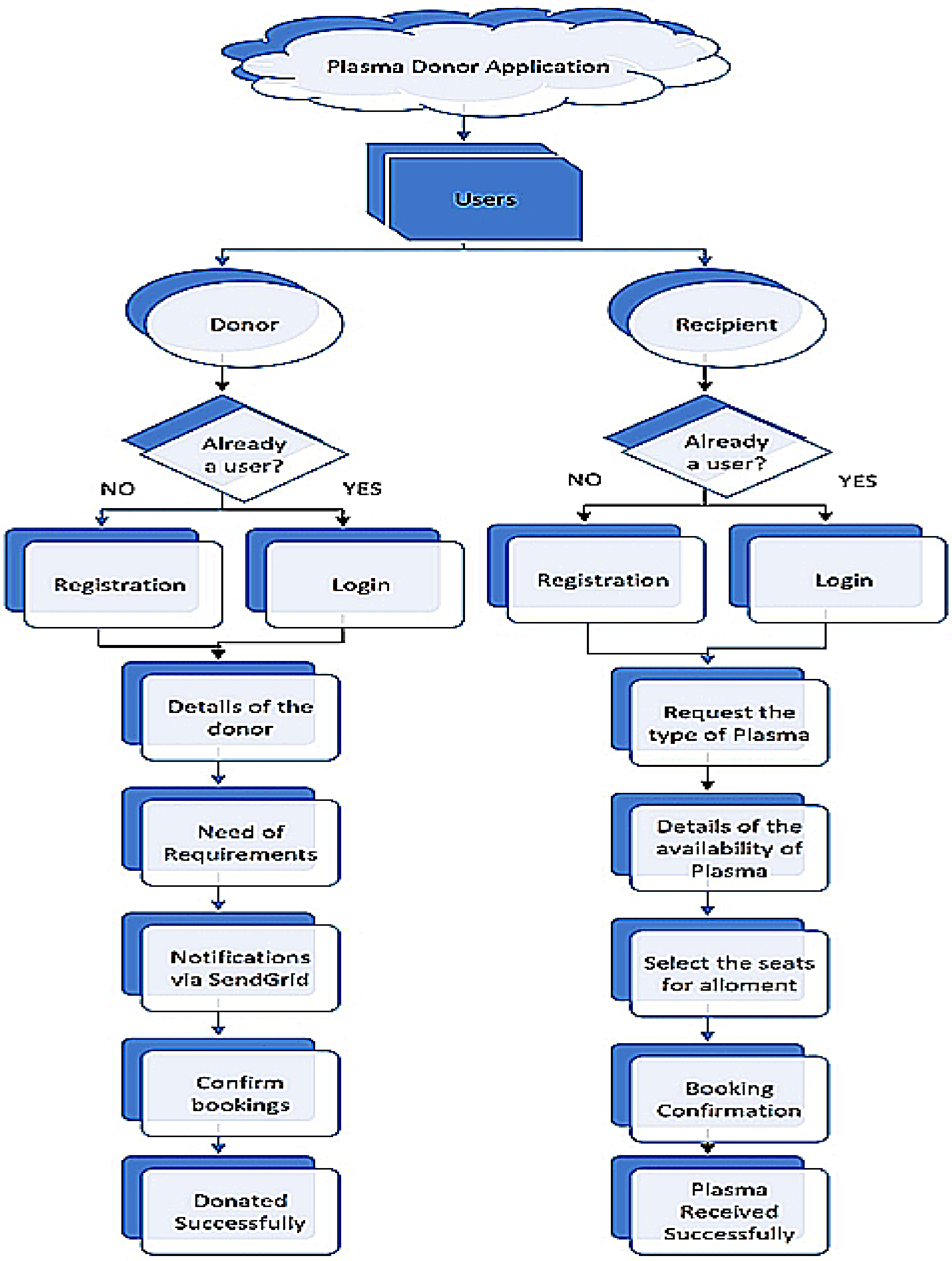
USER STORIES:



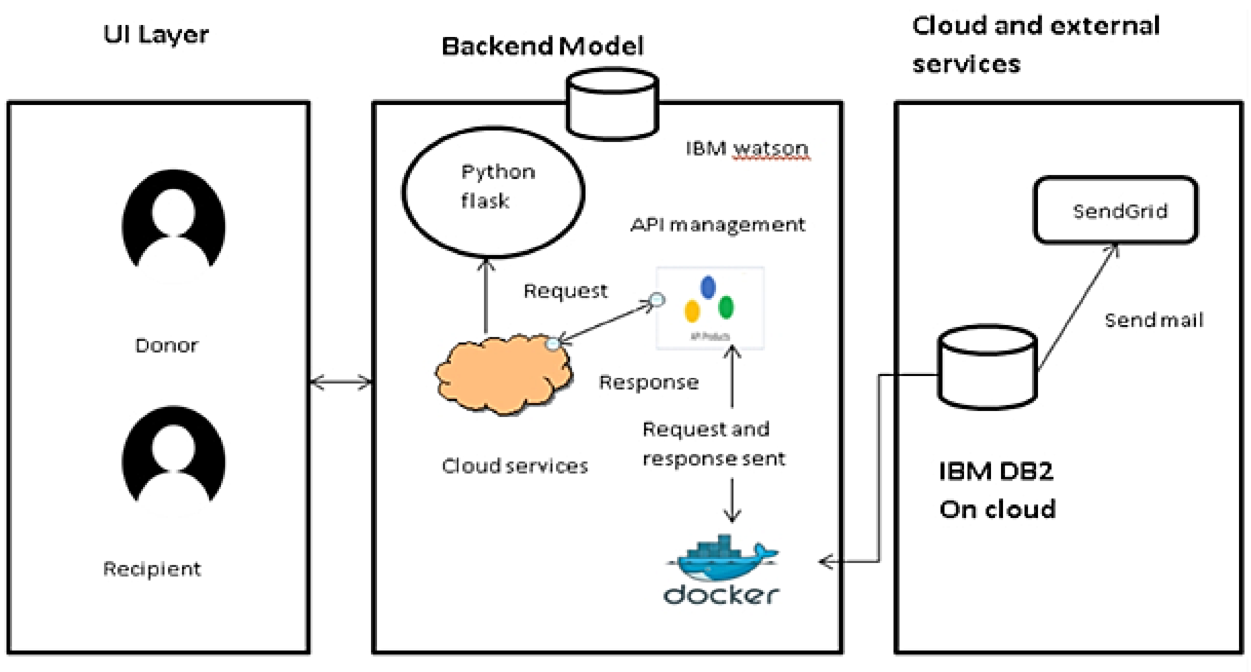
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through our web application.	I can register & access the dashboard with Login credentials	Low	Sprint-2
		USN-4	As a user, I can register for the application through the verification link in Email.		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
Plasma Bank Executive	Registration	USN-1	As a person from plasma bank authority , I will have special access to the website.	I can access my account / dashboard	High	Sprint-1
	Functionalities	USN-2	After checking the acceptability status of the donor, I will update the donor details into the database.	Approval of donor information	High	Sprint-1
		USN-3	When a request is received, I website shows compatible donors, then I let the recipient contact donor.	Request and response	High	Sprint-1
Health care agents(Doctors and others)	Registration	USN-1	As a Health care authority person, I will have special access to the website.	I can access my account / dashboard	High	Sprint-1
		USN-2	I will be logged in to the admin page by the credentials given by my hospital.	I can receive confirmation email & click confirm	Low	Sprint-2
	Functionalities	USN-3	I check the health conditions of plasma donors and update their profile to plasma bank if they are eligible to donate.	Approval of donor donation	High	Sprint-1

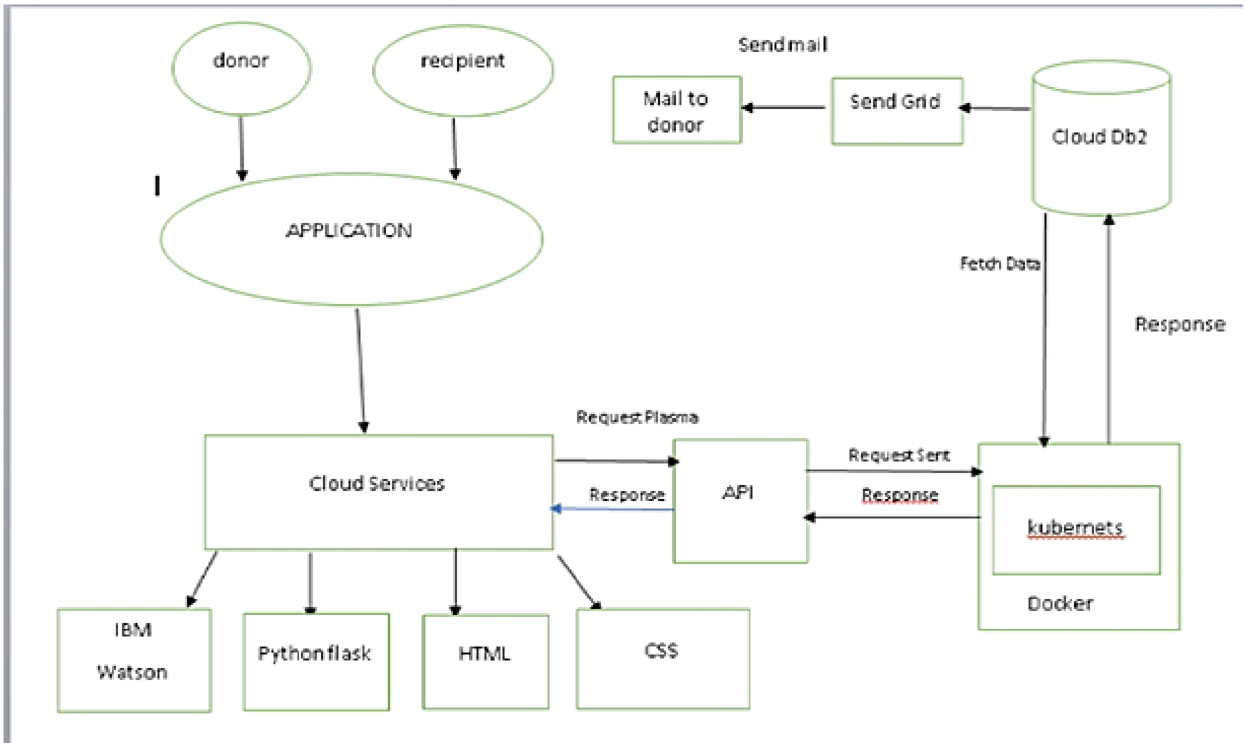
5.2 Solutions & Technical Architecture:

Solution Architecture:



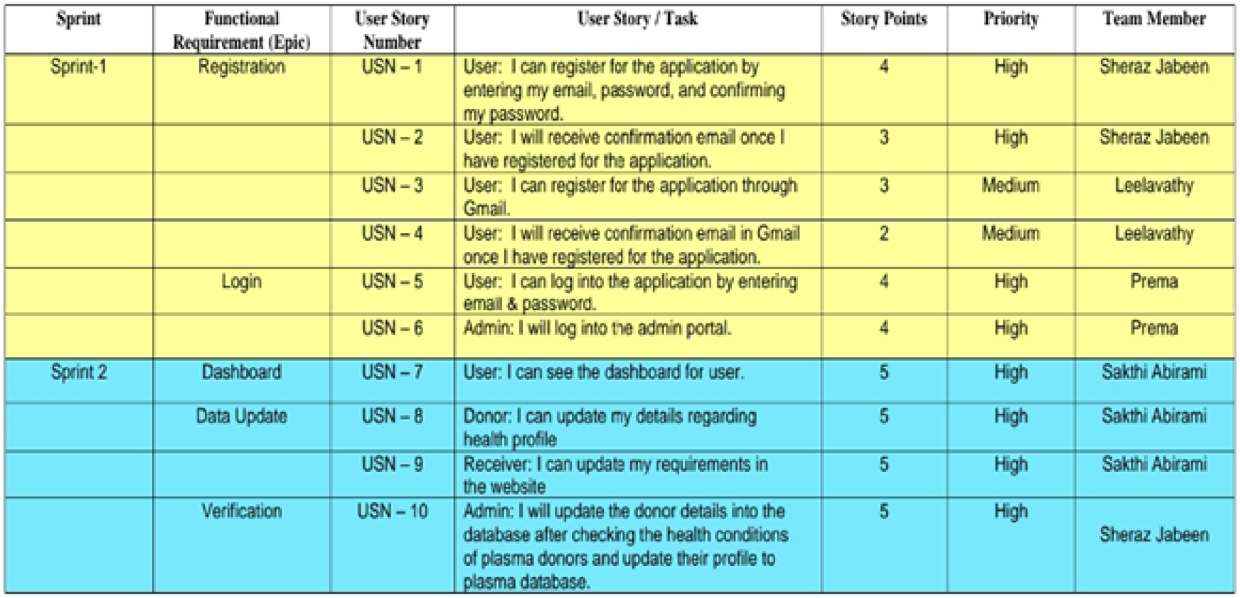
Technical Architecture:



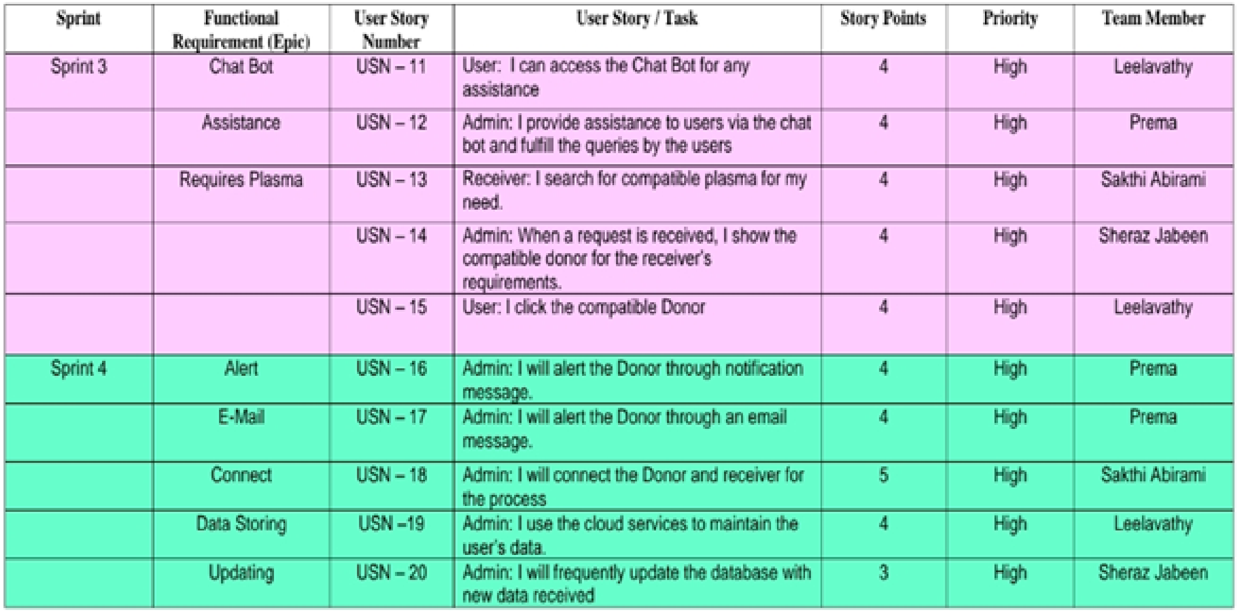


6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

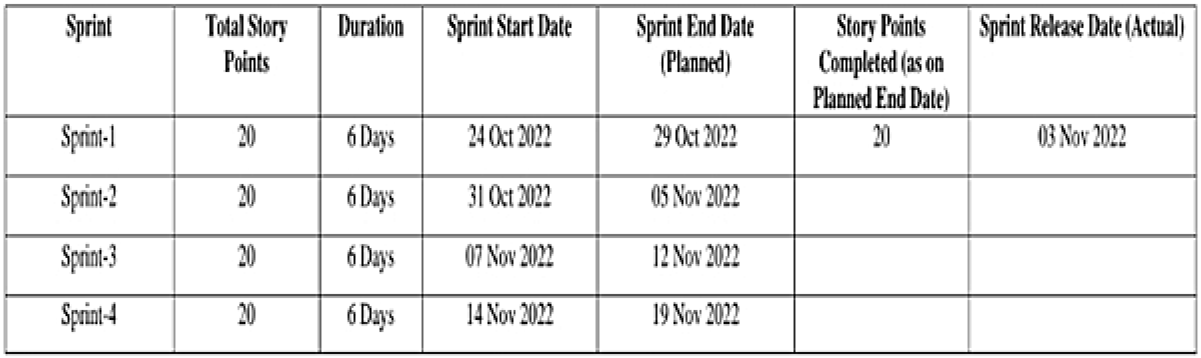


Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Member
Sprint-1	Registration	USN – 1	User: I can register for the application by entering my email, password, and confirming my password.	4	High	Sheraz Jabeen
		USN – 2	User: I will receive confirmation email once I have registered for the application.	3	High	Sheraz Jabeen
		USN – 3	User: I can register for the application through Gmail.	3	Medium	Leelavathy
		USN – 4	User: I will receive confirmation email in Gmail once I have registered for the application.	2	Medium	Leelavathy
	Login	USN – 5	User: I can log into the application by entering email & password.	4	High	Prema
		USN – 6	Admin: I will log into the admin portal.	4	High	Prema
Sprint 2	Dashboard	USN – 7	User: I can see the dashboard for user.	5	High	Sakthi Abirami
	Data Update	USN – 8	Donor: I can update my details regarding health profile	5	High	Sakthi Abirami
		USN – 9	Receiver: I can update my requirements in the website	5	High	Sakthi Abirami
	Verification	USN – 10	Admin: I will update the donor details into the database after checking the health conditions of plasma donors and update their profile to plasma database.	5	High	Sheraz Jabeen



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Member
Sprint 3	Chat Bot	USN – 11	User: I can access the Chat Bot for any assistance	4	High	Leelavathy
	Assistance	USN – 12	Admin: I provide assistance to users via the chat bot and fulfill the queries by the users	4	High	Prema
	Requires Plasma	USN – 13	Receiver: I search for compatible plasma for my need.	4	High	Sakthi Abirami
		USN – 14	Admin: When a request is received, I show the compatible donor for the receiver's requirements.	4	High	Sheraz Jabeen
		USN – 15	User: I click the compatible Donor	4	High	Leelavathy
Sprint 4	Alert	USN – 16	Admin: I will alert the Donor through notification message.	4	High	Prema
	E-Mail	USN – 17	Admin: I will alert the Donor through an email message.	4	High	Prema
	Connect	USN – 18	Admin: I will connect the Donor and receiver for the process	5	High	Sakthi Abirami
	Data Storing	USN –19	Admin: I use the cloud services to maintain the user's data.	4	High	Leelavathy
	Updating	USN – 20	Admin: I will frequently update the database with new data received	3	High	Sheraz Jabeen

6.2 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	6 Days	24 Oct 2022	29 Oct 2022	20	03 Nov 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		

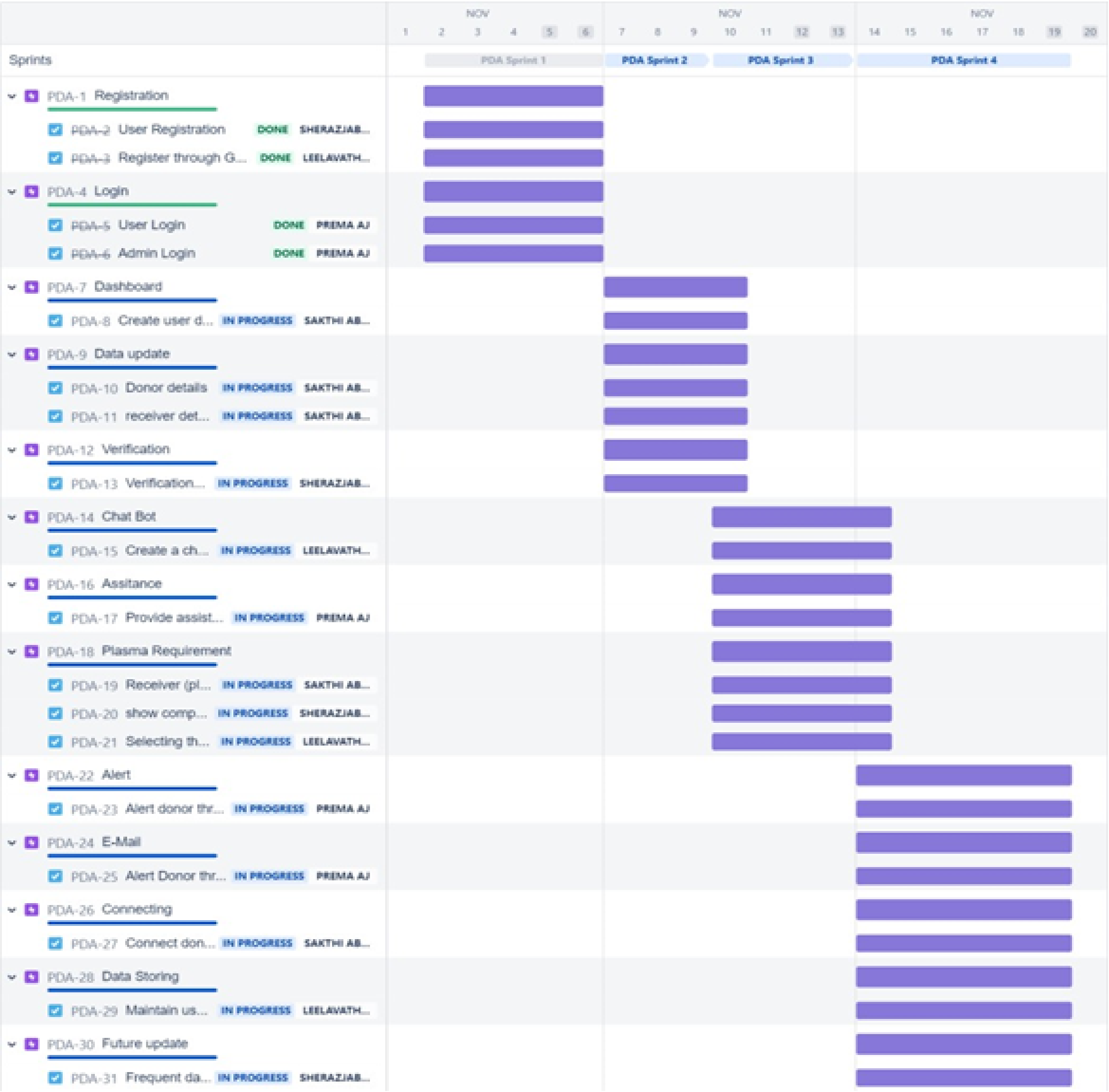
6.3 Reports from Jira Software:

Imagine we have a 10-day sprint duration, and velocity of the team is 20(points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit(story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



Sprints			
PDA-1	Registration		
PDA-2	User Registration	DONE	SHERAZJAB...
PDA-3	Register through G...	DONE	LEELAVATH...
PDA-4	Login		
PDA-5	User Login	DONE	PREMA AJ
PDA-6	Admin Login	DONE	PREMA AJ
PDA-7	Dashboard		
PDA-8	Create user d...	IN PROGRESS	SAKTHI AB...
PDA-9	Data update		
PDA-10	Donor details	IN PROGRESS	SAKTHI AB...
PDA-11	receiver det...	IN PROGRESS	SAKTHI AB...
PDA-12	Verification		
PDA-13	Verification...	IN PROGRESS	SHERAZJAB...
PDA-14	Chat Bot		
PDA-15	Create a ch...	IN PROGRESS	LEELAVATH...
PDA-16	Assitance		
PDA-17	Provide assist...	IN PROGRESS	PREMA AJ
PDA-18	Plasma Requirement		
PDA-19	Receiver (pl...	IN PROGRESS	SAKTHI AB...
PDA-20	show comp...	IN PROGRESS	SHERAZJAB...
PDA-21	Selecting th...	IN PROGRESS	LEELAVATH...
PDA-22	Alert		
PDA-23	Alert donor thr...	IN PROGRESS	PREMA AJ
PDA-24	E-Mail		
PDA-25	Alert Donor thr...	IN PROGRESS	PREMA AJ
PDA-26	Connecting		
PDA-27	Connect don...	IN PROGRESS	SAKTHI AB...
PDA-28	Data Storing		
PDA-29	Maintain us...	IN PROGRESS	LEELAVATH...
PDA-30	Future update		
PDA-31	Frequent da...	IN PROGRESS	SHERAZJAB...

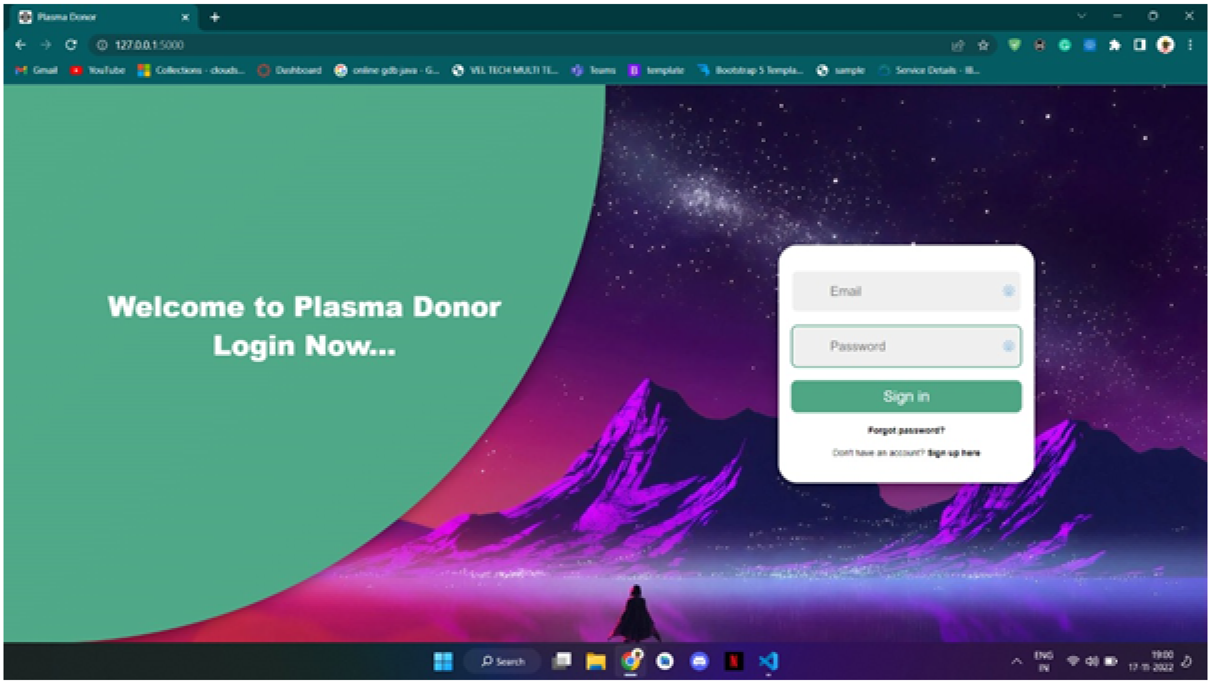
7. CODING & SOLUTIONING

7.1 Registration Page:

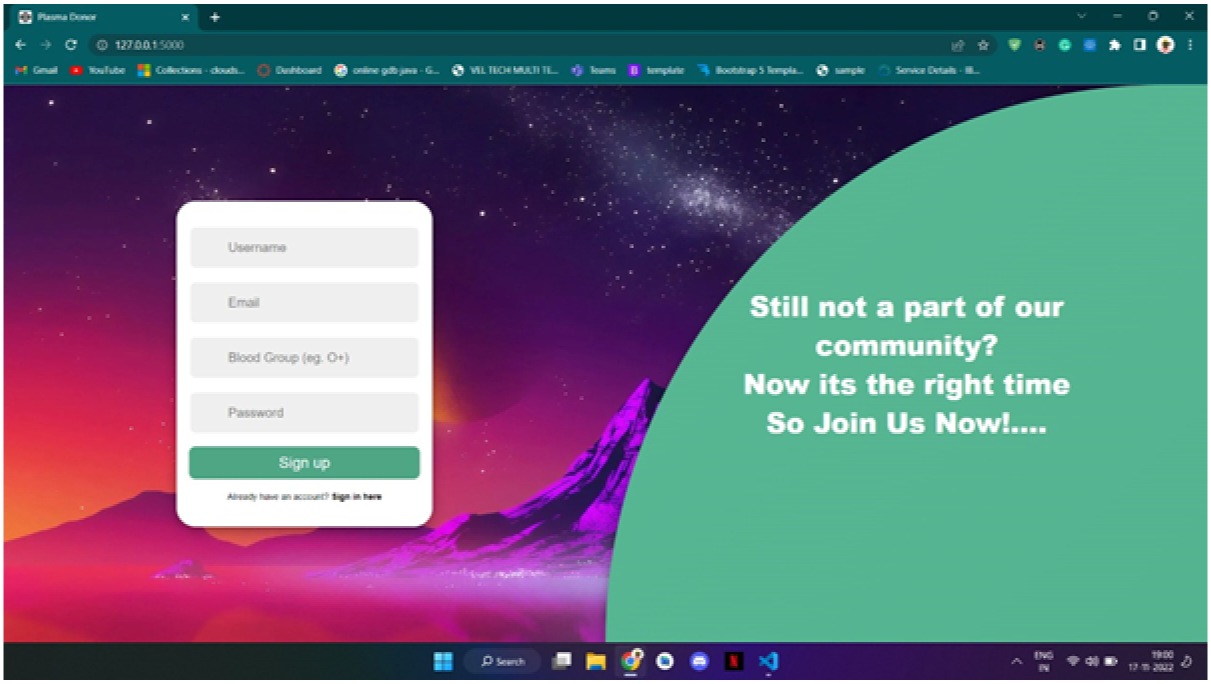
```
from flask import *

from turtle import st

from flask import Flask, render_template, request, redirect, url_for, session

from markupsafe import escape

import ibm_db

import sendgrid

import os

from sendgrid.helpers.mail import *

sg=sendgrid.SendGridAPIClient(api_key='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')

conn = ibm_db.connect("DATABASE=bludb; HOSTNAME=55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud; PORT=31929; SECURITY=SSL; SSLServerCertificate=DigiCertGlobalRootCA.crt; UID=wxn48004; PWD=BuZMZ7F8YeMtngPp",",")

app = Flask(__name__)

#@app.route('/')

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

def register():

  msg = ' '

  if request.method == 'POST':

    username = request.form['username']

    email = request.form['email']

    bloodgroup=request.form['bloodgroup']

    password = request.form['password']
```

```
sql = "SELECT * FROM users 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)

if account:

  msg = 'Account already exists !'

elif not username or not password or not email:

  msg = 'Please fill the Missing Details !'

else:

  insert_sql = "INSERT INTO users VALUES (?,?,?,?)"

  prep_stmt = ibm_db.prepare(conn, insert_sql)

  ibm_db.bind_param(prep_stmt, 1, username)

  ibm_db.bind_param(prep_stmt, 2, email)

  ibm_db.bind_param(prep_stmt, 3, bloodgroup)

  ibm_db.bind_param(prep_stmt, 4, password)

  ibm_db.execute(prep_stmt)

  msg = 'Account created successfully '

  from_email = Email("leelavathy1105@gmail.com")

  to_email = To(email)

  subject = "Thank You for the Registration"

  content = Content("text/plain", "Your donation will be helpful to so many people.")

  mail = Mail(from_email, to_email, subject, content)

  response = sg.client.mail.send.post(request_body=mail.get())

  print(response.status_code)
```

```
      print(response.body)
      print(response.headers)
      return render_template('login.html',msg=msg)
  return render_template('login.html',msg=msg)
@app.route('/login',methods = ['POST', 'GET'])
def login():
  msg = ' '
  if request.method == 'POST':
    email = request.form['email']
    password = request.form['password']
    sql = "SELECT * FROM users WHERE Email =?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,email)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_both(stmt)
    accounts=account
    if (account):
      if(password == accounts['PASSWORD']):
        msg = 'Logged in successfully !'
        return render_template('home.html',msg=msg)
      else :
        msg='Wrong Credentials'
  return  render_template('login.html',msg=msg)
@app.route('/requestform',methods = ['POST','GET'])
def requestform():
  msg = ' '
  if request.method == 'POST':
    username = request.form['username']
    age = request.form['age']
    gender = request.form['gender']
    bloodgroup= request.form['bloodgroup']
    email = request.form['email']
    address=request.form['address']
    city = request.form['city']
    pincode = request.form['pincode']
    contactno = request.form['contactno']
    sql = "SELECT * FROM receiver WHERE username =?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,username)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    if account:
      msg = 'Request already sent!'
    elif not username or not bloodgroup or not email or not contactno or not address or not city or not pincode or not gender or not age:
      msg = 'Please fill the Missing Details !'
    else:
      insert_sql = "INSERT INTO receiver VALUES (?,?,?,?,?,?,?,?,?)"
      prep_stmt = ibm_db.prepare(conn, insert_sql)
      ibm_db.bind_param(prep_stmt, 1, username)
      ibm_db.bind_param(prep_stmt, 2, age)
      ibm_db.bind_param(prep_stmt, 3, gender)
      ibm_db.bind_param(prep_stmt, 4, bloodgroup)
      ibm_db.bind_param(prep_stmt, 5, email)
      ibm_db.bind_param(prep_stmt, 6, address)
      ibm_db.bind_param(prep_stmt, 7, city)
      ibm_db.bind_param(prep_stmt, 8, pincode)
      ibm_db.bind_param(prep_stmt, 9, contactno)
      ibm_db.execute(prep_stmt)
      msg = 'Request sent successfully'
      return render_template('home.html',msg=msg)
  return render_template('requestform.html',msg=msg)


@app.route('/list')
def list():
  users = []
  sql = "SELECT * FROM users"
  stmt = ibm_db.exec_immediate(conn, sql)
  dictionary = ibm_db.fetch_both(stmt)
  while dictionary != False:
    # print ("The Name is : ",  dictionary)
    users.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
  if users:
    return render_template("list.html",users = users)
  return "success..."
if __name__ == '__main__':
    app.run(debug=True)
```



Welcome to Plasma Donor
Login Now...



Still not a part of our
community?
Now its the right time
So Join Us Now!....

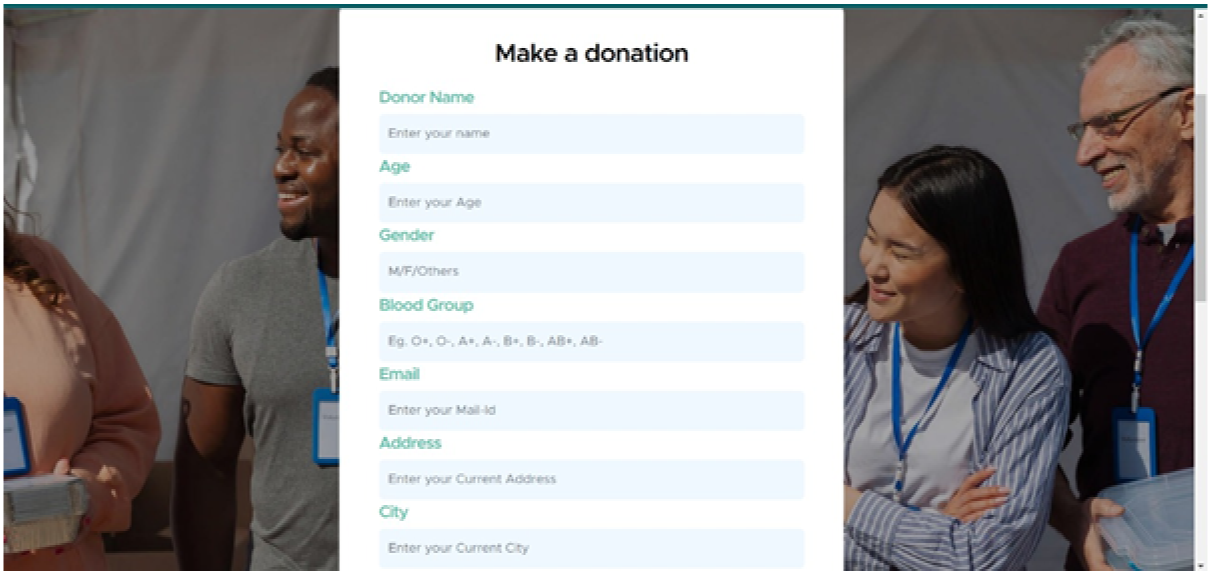
7.2 Dashboard Page:

```
from flask import *

from turtle import st

from flask import Flask, render_template, request, redirect, url_for, session

from markupsafe import escape

import ibm_db

import sendgrid

import os

from sendgrid.helpers.mail import *

sg=sendgrid.SendGridAPIClient(api_key='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')

conn = ibm_db.connect("DATABASE=bludb; HOSTNAME=55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud; PORT=31929; SECURITY=SSL; SSLServerCertificate=DigiCertGlobalRootCA.crt; UID=wxn48004; PWD=BuZMZ7F8YeMtngPp","","")

app = Flask(__name__)

#@app.route('/')

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

def register():

  msg = ' '

  if request.method == 'POST':

    username = request.form['username']

    email = request.form['email']

    bloodgroup=request.form['bloodgroup']

    password = request.form['password']

    sql = "SELECT * FROM users 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)
if account:
  msg = 'Account already exists !'
elif not username or not password or not email:
  msg = 'Please fill the Missing Details !'
else:
  insert_sql = "INSERT INTO users VALUES (?,?,?,?)"
  prep_stmt = ibm_db.prepare(conn, insert_sql)
  ibm_db.bind_param(prep_stmt, 1, username)
  ibm_db.bind_param(prep_stmt, 2, email)
  ibm_db.bind_param(prep_stmt, 3, bloodgroup)
  ibm_db.bind_param(prep_stmt, 4, password)
  ibm_db.execute(prep_stmt)
  msg = 'Account created successfully '
  from_email = Email("leelavathy1105@gmail.com")
  to_email = To(email)
  subject = "Thank You for the Registration"
  content = Content("text/plain", "Your donation will be helpful to so many people.")
  mail = Mail(from_email, to_email, subject, content)
  response = sg.client.mail.send.post(request_body=mail.get())
  print(response.status_code)
  print(response.body)
```

```
      print(response.headers)
      return render_template('login.html',msg=msg)
  return render_template('login.html',msg=msg)
@app.route('/login',methods = ['POST', 'GET'])
def login():
  msg = ' '
  if request.method == 'POST':
    email = request.form['email']
    password = request.form['password']
    sql = "SELECT * FROM users WHERE Email =?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,email)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_both(stmt)
    accounts=account
    if (account):
      if(password == accounts['PASSWORD']):
        msg = 'Logged in successfully !'
        return render_template('index.html',msg=msg)
      else :
        msg='Wrong Credentials'
  return  render_template('login.html',msg=msg)
@app.route('/requestform',methods = ['POST','GET'])
def requestform():
  msg = ' '
  if request.method == 'POST':
    username = request.form['username']
    age = request.form['age']
    gender = request.form['gender']
    bloodgroup= request.form['bloodgroup']
    email = request.form['email']
    address=request.form['address']
    city = request.form['city']
    pincode = request.form['pincode']
    contactno = request.form['contactno']
    sql = "SELECT * FROM receiver WHERE username =?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,username)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    if account:
      msg = 'Request already sent!'
    elif not username or not bloodgroup or not email or not contactno or not address or not city or not pincode or not gender or not age:
      msg = 'Please fill the Missing Details !'
    else:
      insert_sql = "INSERT INTO receiver VALUES (?,?,?,?,?,?,?,?,?)"
      prep_stmt = ibm_db.prepare(conn, insert_sql)
      ibm_db.bind_param(prep_stmt, 1, username)
      ibm_db.bind_param(prep_stmt, 2, age)
      ibm_db.bind_param(prep_stmt, 3, gender)
      ibm_db.bind_param(prep_stmt, 4, bloodgroup)
      ibm_db.bind_param(prep_stmt, 5, email)
      ibm_db.bind_param(prep_stmt, 6, address)
      ibm_db.bind_param(prep_stmt, 7, city)
      ibm_db.bind_param(prep_stmt, 8, pincode)
      ibm_db.bind_param(prep_stmt, 9, contactno)
      ibm_db.execute(prep_stmt)
      msg = 'Request sent successfully'
      return render_template('index.html',msg=msg)
  return render_template('donate.html',msg=msg)
@app.route('/list')
def list():
  users = []
  sql = "SELECT * FROM users"
  stmt = ibm_db.exec_immediate(conn, sql)
  dictionary = ibm_db.fetch_both(stmt)
  while dictionary != False:
    # print ("The Name is : ",  dictionary)
    users.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
  if users:
    return render_template("list.html",users = users)
  return "success..."
if __name__ == '__main__':
```

```
app.run(debug=True)
```



Make a donation

Donor Name

Enter your name

Age

Enter your Age

Gender

M/F/Others

Blood Group

Eg. O+, O-, A+, A-, B+, B-, AB+, AB-

Email

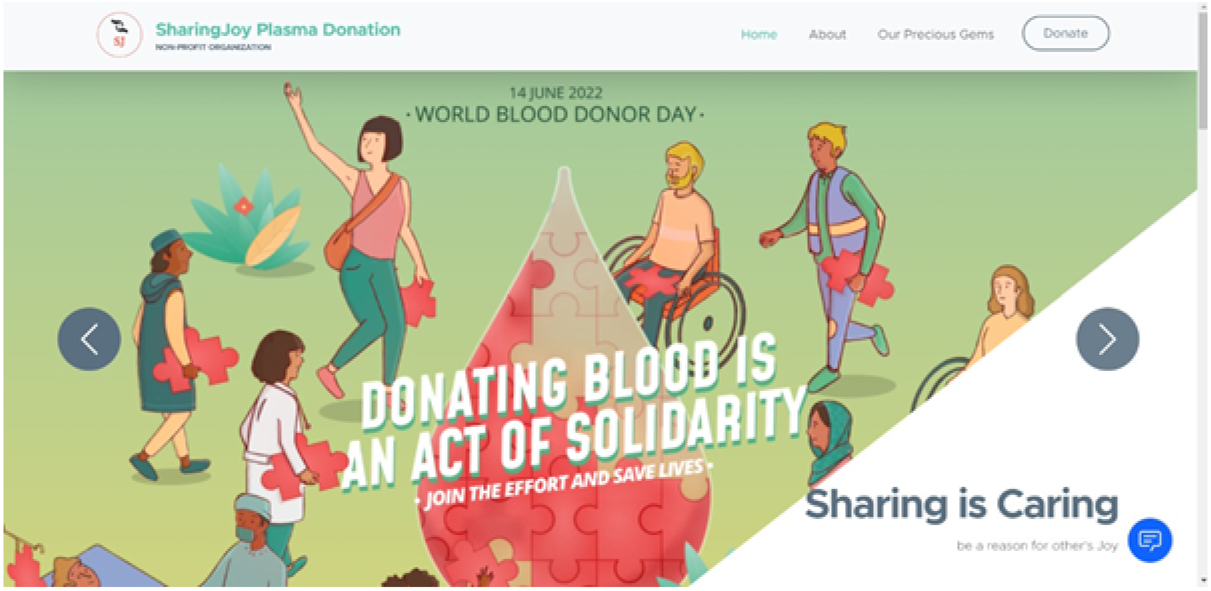
Enter your Mail-Id

Address

Enter your Current Address

City

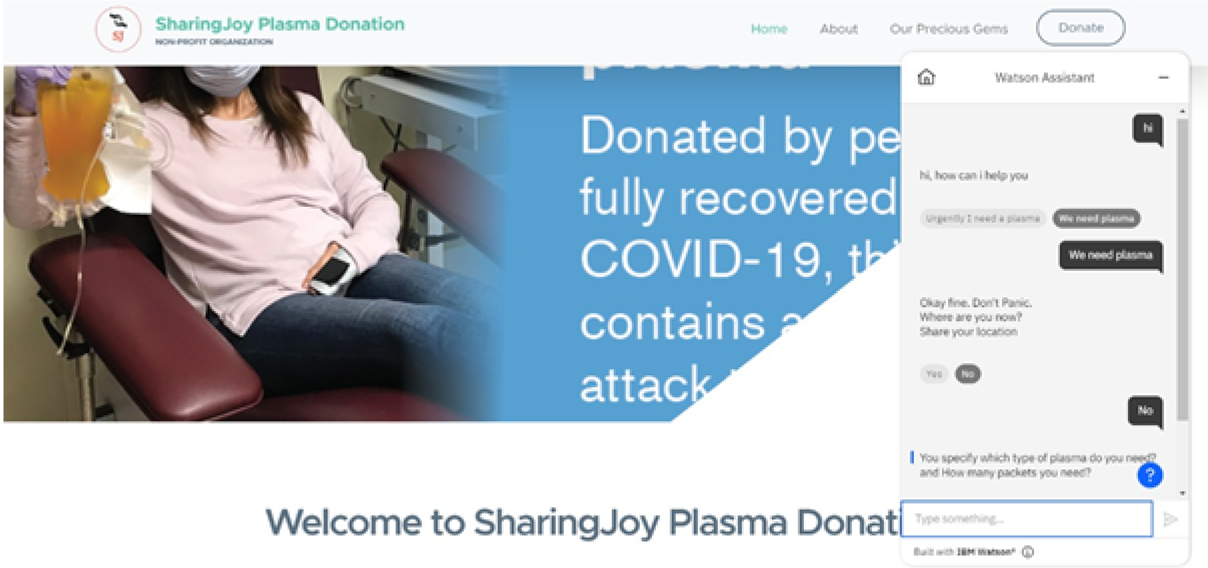
Enter your Current City



SJ
SharingJoy Plasma Donation
NON-PROFIT ORGANIZATION
Home
About
Our Precious Gems
Donate
14 JUNE 2022
· WORLD BLOOD DONOR DAY ·
DONATING BLOOD IS AN ACT OF SOLIDARITY
· JOIN THE EFFORT AND SAVE LIVES ·

Sharing is Caring

be a reason for other's Joy



Donated by pe
fully recovered
COVID-19, th
contains a
attack

hi

hi, how can i help you

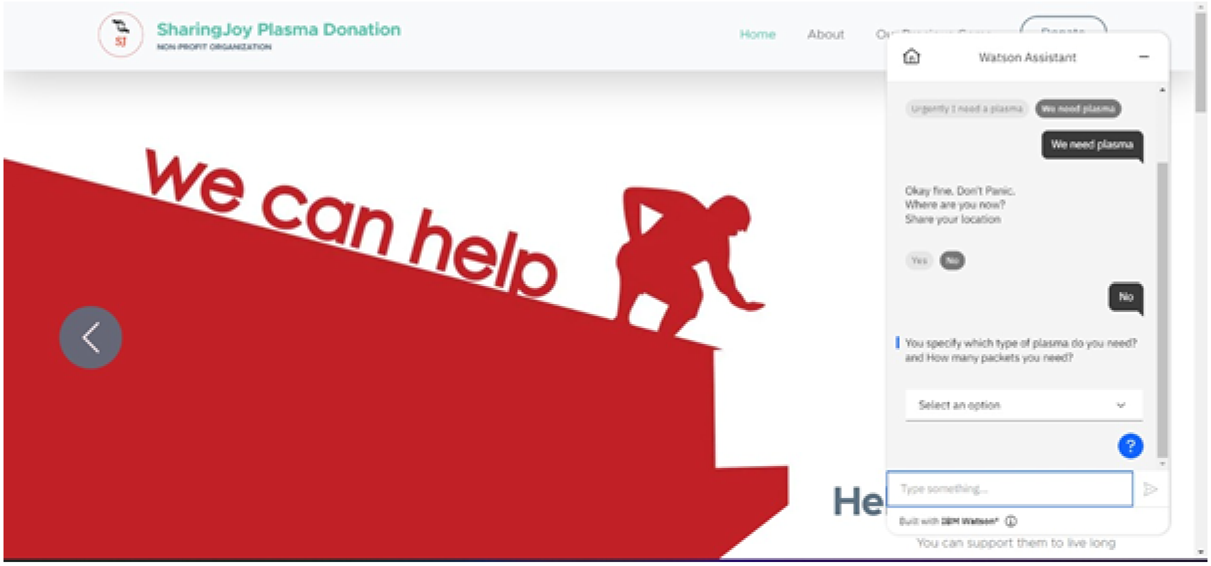
We need plasma

Okay fine. Don't Panic.
Where are you now?
Share your location

No

You specify which type of plasma do you need?
and How many packets you need?

Welcome to SharingJoy Plasma Donat



We need plasma

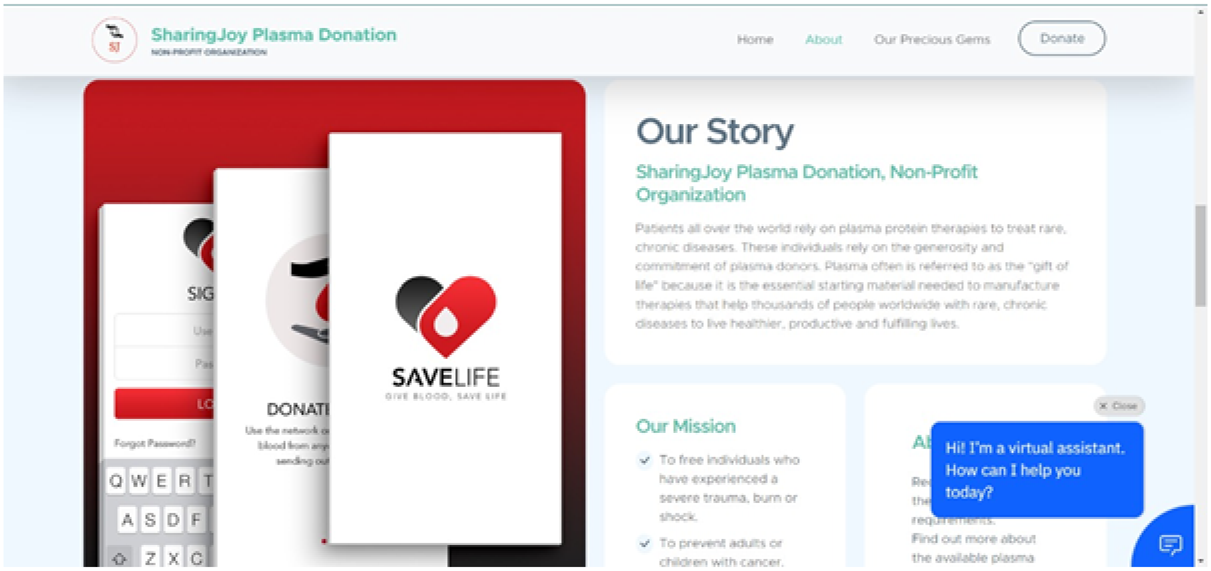
Okay fine. Don't Panic.
Where are you now?
Share your location

No

You specify which type of plasma do you need?
and How many packets you need?

He

You can support them to live long



Our Story

SharingJoy Plasma Donation, Non-Profit Organization

Patients all over the world rely on plasma protein therapies to treat rare, chronic diseases. These individuals rely on the generosity and commitment of plasma donors. Plasma often is referred to as the "gift of life" because it is the essential starting material needed to manufacture therapies that help thousands of people worldwide with rare, chronic diseases to live healthier, productive and fulfilling lives.

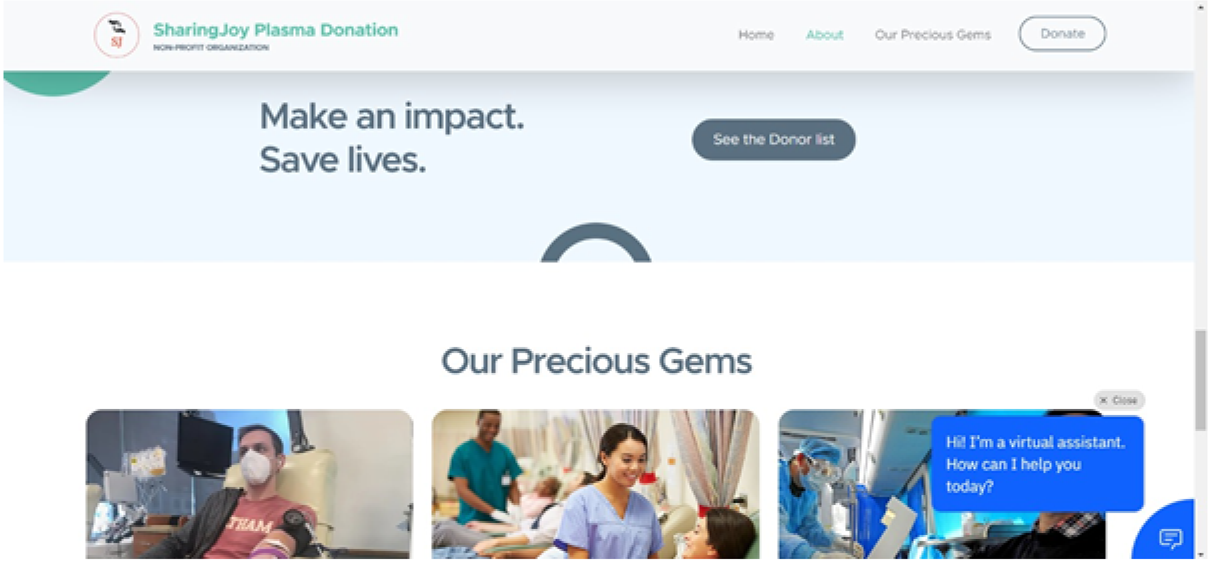
Our Mission

- ✓ To free individuals who have experienced a severe trauma, burn or shock.
- ✓ To prevent adults or children with cancer.

Al

Re
the
requirements.
Find out more about
the available plasma

Hi! I'm a virtual assistant. How can I help you today?



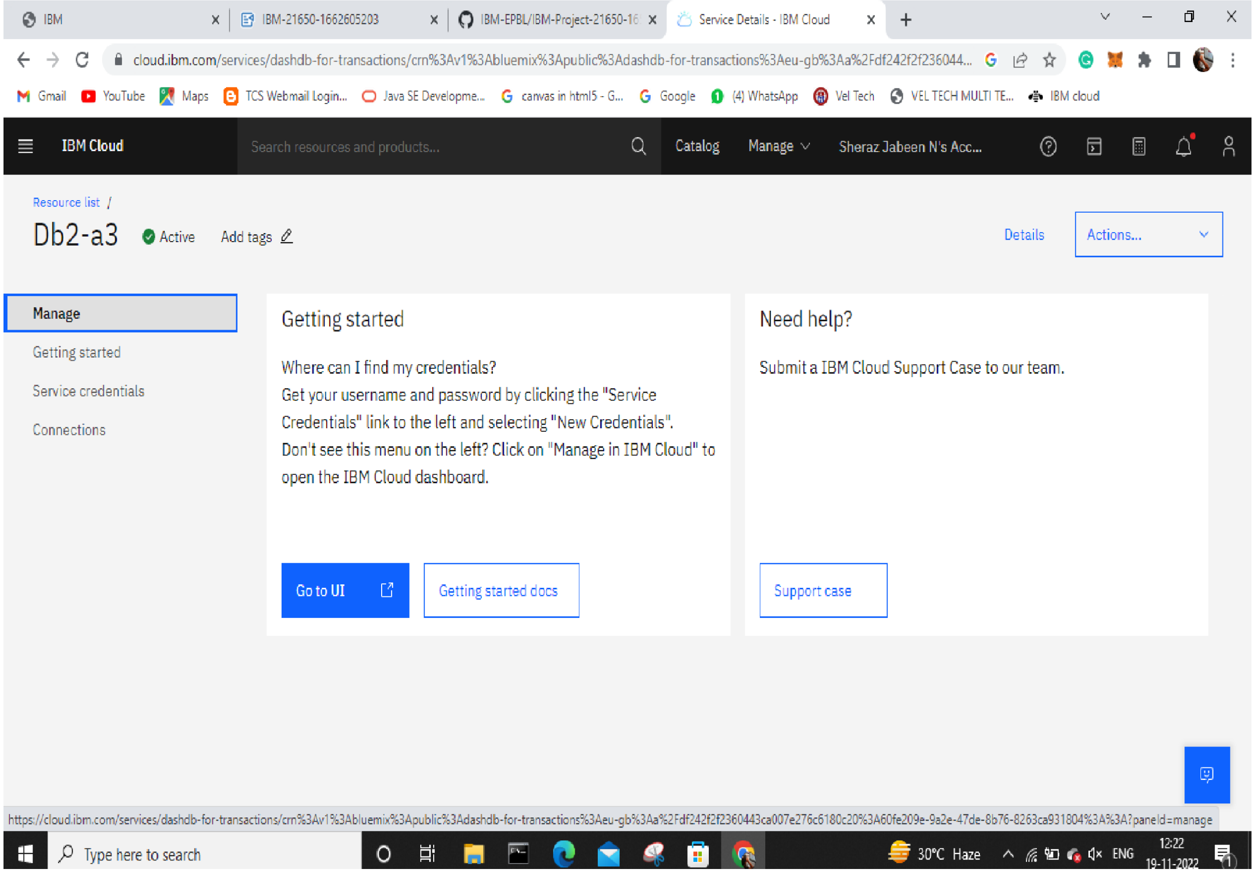
Make an impact.
Save lives.

Our Precious Gems

Hi! I'm a virtual assistant. How can I help you today?

7.3 Database Schema

DB2-Database:


IBM
IBM-21650-1662605203
IBM-EPBL/IBM-Project-21650-16
Service Details - IBM Cloud
cloud.ibm.com/services/dashdb-for-transactions/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aeu-gb%3Aa%2Fdf242f2f236044...
Gmail
YouTube
Maps
TCS Webmail Login...
Java SE Developme...
canvas in html5 - G...
Google
(4) WhatsApp
Vel Tech
VEL TECH MULTI TE...
IBM cloud
IBM Cloud
Search resources and products...
Catalog
Manage
Sheraz Jabeen N's Acc...
Resource list /
Details
Actions...
Manage
Getting started
Service credentials
Connections
Go to UI
Getting started docs
Support case
https://cloud.ibm.com/services/dashdb-for-transactions/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aeu-gb%3Aa%2Fdf242f2f2360443ca007e276c6180c20%3A60fe209e-9a2e-47de-8b76-8263ca931804%3A%3A?paneId=manage
Type here to search
30°C Haze
ENG
12:22
19-11-2022

Db2-a3 Active Add tags

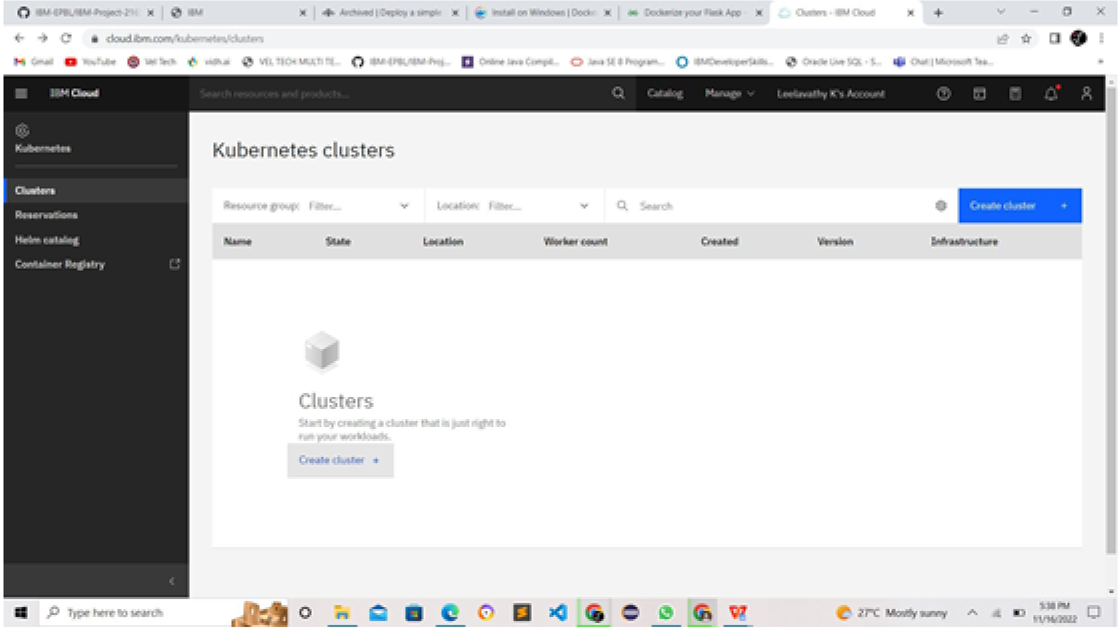
Getting started

Where can I find my credentials?
Get your username and password by clicking the "Service Credentials" link to the left and selecting "New Credentials".
Don't see this menu on the left? Click on "Manage in IBM Cloud" to open the IBM Cloud dashboard.

Need help?

Submit a IBM Cloud Support Case to our team.

Kubernetes Cluster:



Kubernetes clusters

Resource group: Filter... | Location: Filter... | Search

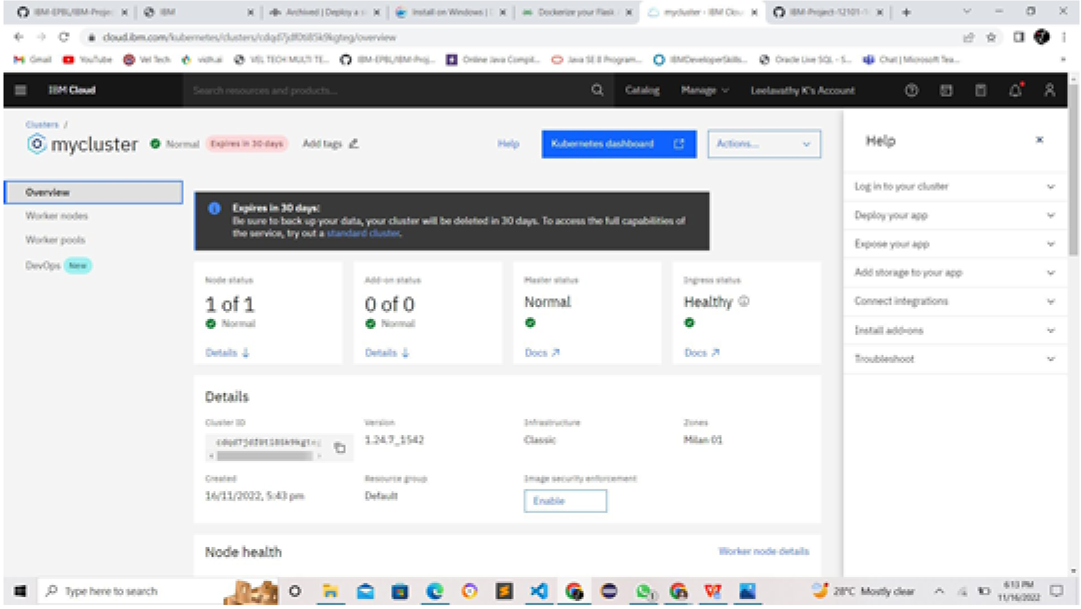
Name	State	Location	Worker count	Created	Version	Infrastructure

Clusters

Start by creating a cluster that is just right to run your workloads.

Create cluster +

Cluster Overview:



mycluster ● Normal Expires in 30 days Add tags

Expires in 30 days:
Be sure to back up your data, your cluster will be deleted in 30 days. To access the full capabilities of the service, try out a standard cluster.

Node status	Add-on status	Master status	Ingress status
1 of 1 ● Normal Details ↓	0 of 0 ● Normal Details ↓	Normal ● Docs ↗	Healthy ● Docs ↗

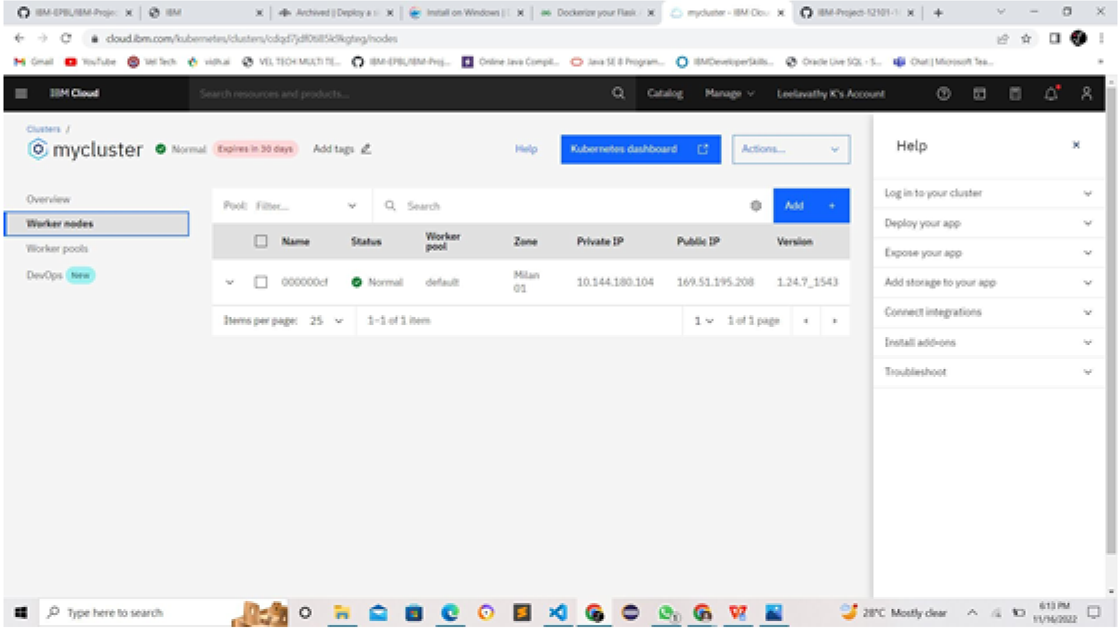
Details

Cluster ID	Version	Infrastructure	Zones
cdqd7jd f0t18l5k9kgt	1.24.7_1542	Classic	Milan 01
Created	**Resource group**	**Image security enforcement**	
16/11/2022, 5:43 pm	Default	Enable	

Node health

Worker node details

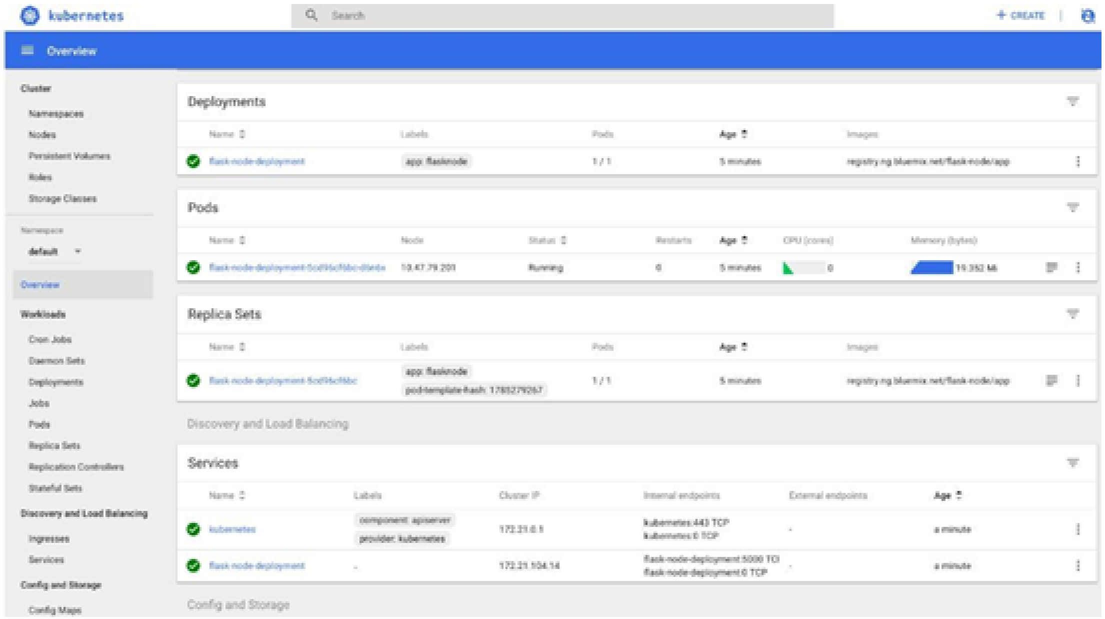
Worker Nodes:



Name	Status	Worker pool	Zone	Private IP	Public IP	Version
000000cf	Normal	default	Milan 01	10.144.180.104	169.51.195.208	1.24.7_1543

Items per page: 25 1–1 of 1 item 1 1 of 1 page

Service:



Deployments

Name	Labels	Pods	Age	Images
flask-node-deployment	app: flasknode	1 / 1	5 minutes	registry.ng.bluemix.net/flask-node/app

Pods

Name	Node	Status	Restarts	Age	CPU (cores)	Memory (bytes)
flask-node-deployment-5cd9 4cf6bc-dlxds	10.47.79.201	Running	0	5 minutes	0	19.352 Mi

Replica Sets

Name	Labels	Pods	Age	Images
flask-node-deployment-5cd94cf6bc	app: flasknode pod-template-hash: 1785279267	1 / 1	5 minutes	registry.ng.bluemix.net/flask-node/app

Discovery and Load Balancing

Services

Name	Labels	Cluster IP	Internal endpoints	External endpoints	Age
kubernetes	component: apiserver provider: kubernetes	172.21.0.1	kubernetes:443 TCP kubernetes:0 TCP	-	a minute
flask-node-deployment	-	172.21.104.14	flask-node-deployment:5000 TCI flask-node-deployment:0 TCP	-	a minute

Config and Storage

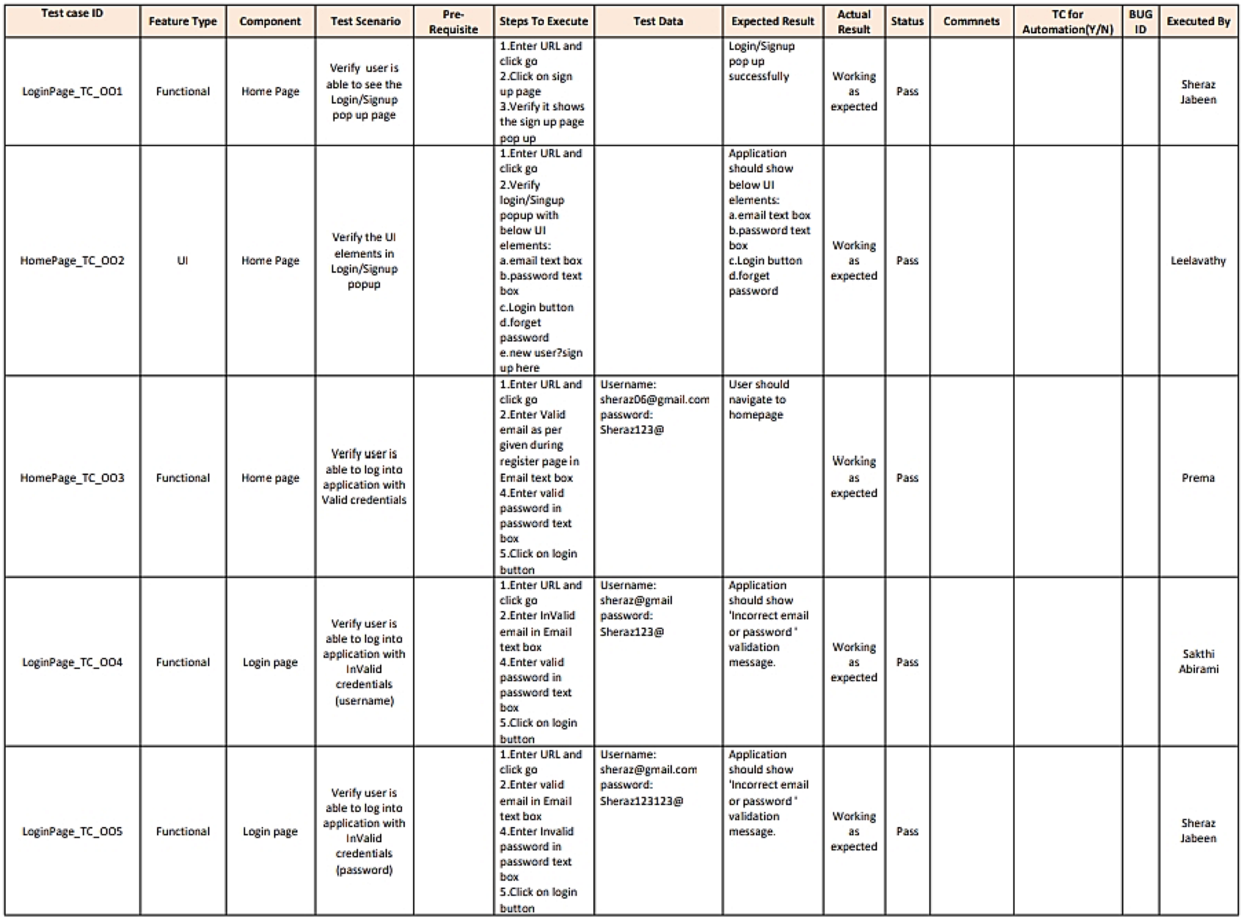
8. TESTING

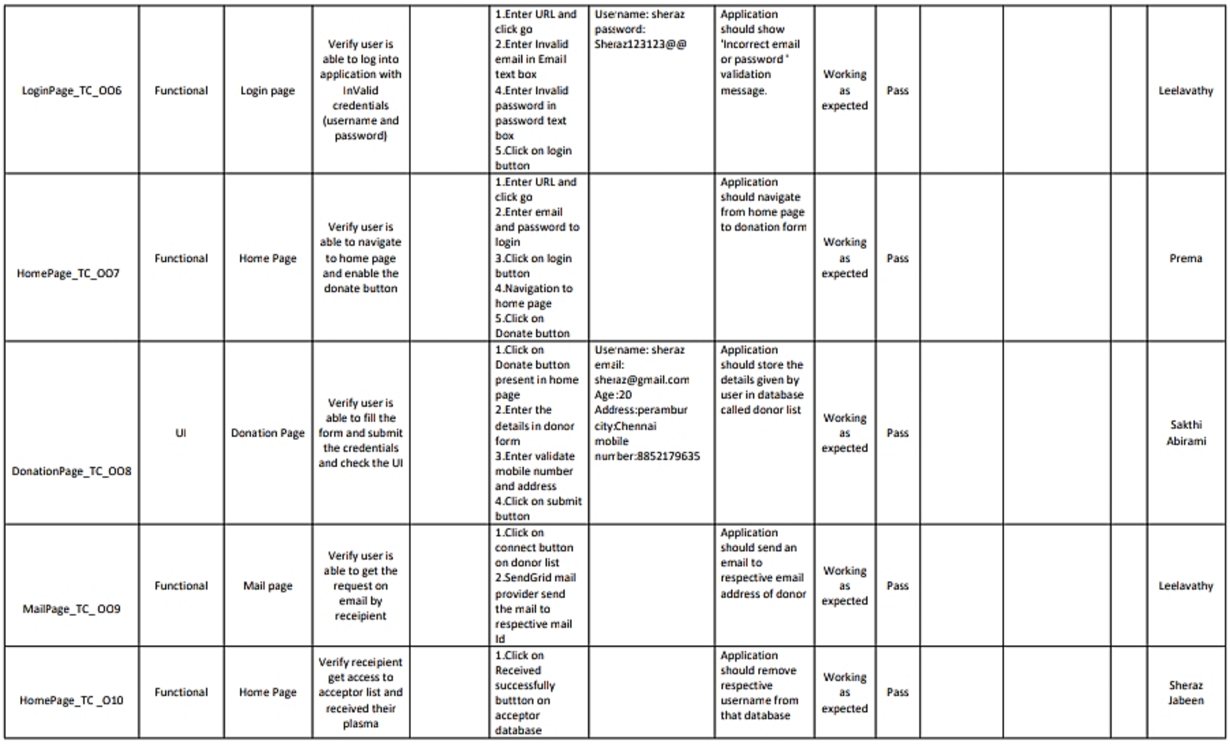
8.1 Test Cases:

A test case has components that describe input, action and an expected response, in order to determine if a feature of an application is working correctly. A test case is a set of instructions on “HOW” to validate a particular test objective/target, which when followed will tell us if the expected behavior of the system is satisfied or not.

Characteristics of a good test case:

- Accurate: Exacts the purpose.
- Economical: No unnecessary steps or words.
- Traceable: Capable of being traced to requirements.
- Repeatable: Can be used to perform the test over and over.
- Reusable: Can be reused if necessary.





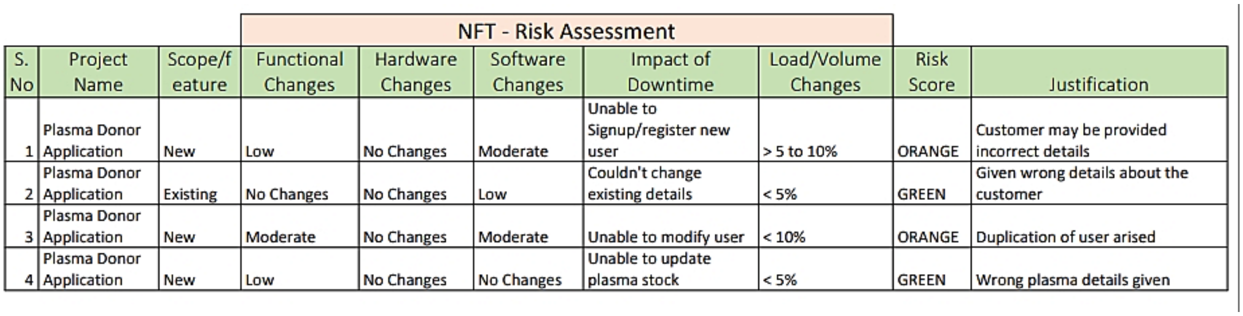
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Signup pop up page		1.Enter URL and click go 2.Click on sign up page 3.Verify it shows the sign up page pop up		Login/Signup pop up successfully	Working as expected	Pass				Sheraz Jabeen
HomePage_TC_OO2	UI	Home Page	Verify the UI elements in Login/Signup popup		1.Enter URL and click go 2.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.forget password e.new user?sign up here		Application should show below UI elements: a.email text box b.password text box c.Login button d.forget password	Working as expected	Pass				Leelavathy
HomePage_TC_OO3	Functional	Home page	Verify user is able to log into application with Valid credentials		1.Enter URL and click go 2.Enter Valid email as per given during register page in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: sheraz06@gmail.com password: Sheraz123@	User should navigate to homepage	Working as expected	Pass				Prema
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials (username)		1.Enter URL and click go 2.Enter InValid email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: sheraz@gmail password: Sheraz123@	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass				Sakthi Abirami
LoginPage_TC_OO5	Functional	Login page	Verify user is able to log into application with InValid credentials (password)		1.Enter URL and click go 2.Enter valid email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: sheraz@gmail.com password: Sheraz123123@	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass				Sheraz Jabeen
LoginPage_TC_OO6	Functional	Login page	Verify user is able to log into application with InValid credentials (username and password)		1.Enter URL and click go 2.Enter Invalid email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: sheraz password: Sheraz123123@@	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass				Leelavathy
HomePage_TC_OO7	Functional	Home Page	Verify user is able to navigate to home page and enable the donate button		1.Enter URL and click go 2.Enter email and password to login 3.Click on login button 4.Navigation to home page 5.Click on Donate button		Application should navigate from home page to donation form	Working as expected	Pass				Prema
DonationPage_TC_OO8	UI	Donation Page	Verify user is able to fill the form and submit the credentials and check the UI		1.Click on Donate button present in home page 2.Enter the details in donor form 3.Enter validate mobile number and address 4.Click on submit button	Username: sheraz email: sheraz@gmail.com Age:20 Address:perambur city:Chennai mobile number:8852179635	Application should store the details given by user in database called donor list	Working as expected	Pass				Sakthi Abirami
MailPage_TC_OO9	Functional	Mail page	Verify user is able to get the request on email by receipient		1.Click on connect button on donor list 2.SendGrid mail provider send the mail to respective mail Id		Application should send an email to respective email address of donor	Working as expected	Pass				Leelavathy
HomePage_TC _O10	Functional	Home Page	Verify receipient get access to acceptor list and received their plasma		1.Click on Received successfully buttton on acceptor database		Application should remove respective username from that database	Working as expected	Pass				Sheraz Jabeen

8.2 User Acceptance Testing:

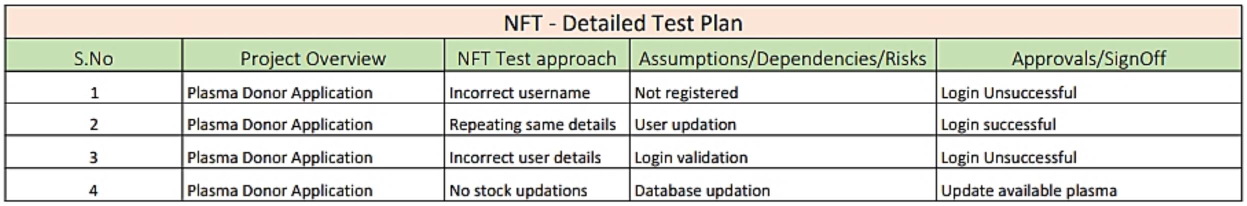
This sort of testing is carried out by users, clients, or other authorised bodies to identify the requirements and operational procedures of an application or piece of software. The most crucial stage of testing is acceptance testing since it determines whether or not the customer will accept the application or programmer. It could entail the application's U.I., performance, usability, and usefulness. It is also referred to as end-user testing, operational acceptance testing, and user acceptance testing (UAT).

9. RESULT

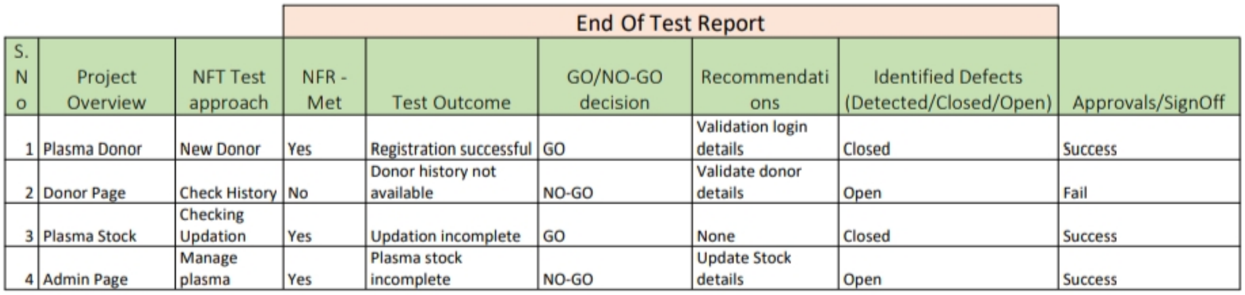
9.1 Performance Metrics:



			NFT - Risk Assessment						
S. No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volume Changes	Risk Score	Justification
1	Plasma Donor Application	New	Low	No Changes	Moderate	Unable to Signup/register new user	> 5 to 10%	ORANGE	Customer may be provided incorrect details
2	Plasma Donor Application	Existing	No Changes	No Changes	Low	Couldn't change existing details	< 5%	GREEN	Given wrong details about the customer
3	Plasma Donor Application	New	Moderate	No Changes	Moderate	Unable to modify user	< 10%	ORANGE	Duplication of user arised
4	Plasma Donor Application	New	Low	No Changes	No Changes	Unable to update plasma stock	< 5%	GREEN	Wrong plasma details given



NFT - Detailed Test Plan				
S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/SignOff
1	Plasma Donor Application	Incorrect username	Not registered	Login Unsuccessful
2	Plasma Donor Application	Repeating same details	User updation	Login successful
3	Plasma Donor Application	Incorrect user details	Login validation	Login Unsuccessful
4	Plasma Donor Application	No stock updations	Database updation	Update available plasma



			End Of Test Report					
S. No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff
1	Plasma Donor	New Donor	Yes	Registration successful	GO	Validation login details	Closed	Success
2	Donor Page	Check History	No	Donor history not available	NO-GO	Validate donor details	Open	Fail
3	Plasma Stock	Checking Updation	Yes	Updation incomplete	GO	None	Closed	Success
4	Admin Page	Manage plasma	Yes	Plasma stock incomplete	NO-GO	Update Stock details	Open	Success

10. ADVANTAGES & DISADVANTAGES

Advantages:

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

Disadvantages:

1. It cannot auto verify user genuineness
2. It requires an active internet connection
3. Due to some wrong location the application will get confused

11. CONCLUSION

In recent days, it is noticed the increase in plasma request posts on social media such as Facebook, Twitter, and Instagram. Interestingly there are many people across the world interested in donating plasma when there is a need, but those donors don’t have an access to know about the plasma donation requests in their local area. This is because that there is no platform to connect local plasma donors with patients. plasma solves the problem and creates a communication channel through authorized clinics whenever a patient needs plasma donation. It is a useful tool to find compatible plasma donors who can receive plasma request posts in their local area. Clinics can use this web application to maintain the plasma donation activity. Collected data through this application can be used to analyse donations to requests rates in a local area to increase the awareness of people by conducting donations camps.

Plasma Application can be developed to further improve user accessibility via integrating this application with various social networks application program interfaces (APIs). Consequently, users can login and sign up using various social networks. This would increase number of donors and enhances the process of plasma donation.

User interface (UI) can be improved in future to accommodate global audience by supporting different languages across countries. Data scraping can be done from different social networks and can be shown in the plasmaRequest Feeds. Appointments can be synchronized with Google and Outlook calendars for the ease of users.

Plasma application provides a reliable platform to connect local blood and plasma donors with patients. plasma creates a communication channel through authenticated clinics whenever a patient needs blood and as well as plasma donation. It is a useful tool to find compatible plasma donors who can receive plasma request posts in their local area. Clinics can use this web application to maintain the plasma donation activity.

12. FUTURE SCOPE

The Scope of a system means that which modules are being covered by the system. The scope clearly defines the boundaries of the proposed system. The functional areas of this application that lies under the scope of the proposed system are the management of the availability of donors, hospitals, plasma banks to the user or member at any time. They system calculates the estimated locations of the donors, hospitals and plasma banks and also provides online chat service between donors and consumers.

The client can also go through from the guidelines section to view the useful precautions needed before and after plasma transfusion. To be a member of the Android plasma Bank has to fill the registration form and provide the necessary information. To provide dynamic database that is storing donors Information and can communicate with them easily.

Future iterations of this project may add more features, such as a native application for the healthcare sector or another business. It is easy to make additional enhancement to this system because of the way it was designed. The modifications of the project would increase the system’s adaptability. Furthermore, the functionalities are provided in a way that will improve the system’s performance.

13. APPENDIX

Source Code:

app.py

```
from flask import *

from turtle import st

from flask import Flask, render_template, request, redirect, url_for, session

from markupsafe import escape

import ibm_db

import sendgrid

import os

from sendgrid.helpers.mail import *

sg =
sendgrid.SendGridAPIClient(api_key='SG.dfdfdfdfadsfeefrewqewq.KyAYHdsfdsfdsfldsf
dsfdsfdsfdsfdsfdsfiqi0')

conn = ibm_db.connect("DATABASE=bluedb; HOSTNAME=55444447-92433466-
4331-af32442343d3-
888b05e723423434c0.bs2io90l08kq234234234b1od8lcg.databases.appdomain.cloud;
PORT=31000; SECURITY=SSL; SSLServerCertificate=DigiCertGlobalRootCA.crt;
UID=kjdghsf50005; PWD=khesseerersb",","

app = Flask(__name__)

#@app.route('/')

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

def register():

  msg = ' '

  if request.method == 'POST':

    username = request.form['username']

    email = request.form['email']

    bloodgroup=request.form['bloodgroup']
    password = request.form['password']
    sql = "SELECT * FROM users 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)
    if account:
      msg = 'Account already exists !'
    elif not username or not password or not email:
      msg = 'Please fill the Missing Details !'
    else:
      insert_sql = "INSERT INTO users VALUES (?,?,?,?)"
      prep_stmt = ibm_db.prepare(conn, insert_sql)
      ibm_db.bind_param(prep_stmt, 1, username)
      ibm_db.bind_param(prep_stmt, 2, email)
      ibm_db.bind_param(prep_stmt, 3, bloodgroup)
      ibm_db.bind_param(prep_stmt, 4, password)
      ibm_db.execute(prep_stmt)
      msg = 'Account created successfully '
      from_email = Email("sherazjabeen18@gmail.com")
      to_email = To(email)
      subject = "Registration confirmation "
      content = Content("text/plain", "Your registration is confirmed. Thank you. SharingJoy")
      mail = Mail(from_email, to_email, subject, content)
      response = sg.client.mail.send.post(request_body=mail.get())
      print(response.status_code)
```

```
         print(response.body)
         print(response.headers)
         return render_template('login.html',msg=msg)
       return render_template('login.html',msg=msg)
@app.route('/login',methods = ['POST', 'GET'])
      def login():
       msg = ' '
       if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        sql = "SELECT * FROM users WHERE Email =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_both(stmt)
        accounts=account
        if (account):
         if(password == accounts['PASSWORD']):
          msg = 'Logged in successfully !'
          return render_template('index.html',msg=msg)
         else :
          msg='Wrong Credentials'
       return  render_template('login.html',msg=msg)
      @app.route('/requestform',methods = ['POST','GET'])
      def requestform():
       msg = ' '
       if request.method == 'POST':
```

```
    username = request.form['username']
    age = request.form['age']
    gender = request.form['gender']
    bloodgroup= request.form['bloodgroup']
    email = request.form['email']
    address=request.form['address']
    city = request.form['city']
    pincode = request.form['pincode']
    contactno = request.form['contactno']
    sql = "SELECT * FROM receiver WHERE username =?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,username)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    if account:
      msg = 'Request already sent!'
    elif not username or not bloodgroup or not email or not contactno or not address or not city or not pincode or not gender or not age:
      msg = 'Please fill the Missing Details !'
    else:
      insert_sql = "INSERT INTO receiver VALUES (?,?,?,?,?,?,?,?,?)"
      prep_stmt = ibm_db.prepare(conn, insert_sql)
      ibm_db.bind_param(prep_stmt, 1, username)
      ibm_db.bind_param(prep_stmt, 2, age)
      ibm_db.bind_param(prep_stmt, 3, gender)
      ibm_db.bind_param(prep_stmt, 4, bloodgroup)
      ibm_db.bind_param(prep_stmt, 5, email)
      ibm_db.bind_param(prep_stmt, 6, address)
      ibm_db.bind_param(prep_stmt, 7, city)
      ibm_db.bind_param(prep_stmt, 8, pincode)
      ibm_db.bind_param(prep_stmt, 9, contactno)
      ibm_db.execute(prep_stmt)
      msg = 'Request sent successfully'
      return render_template('index.html',msg=msg)
  return render_template('donate.html',msg=msg)
@app.route('/list')
def list():
  users = []
  sql = "SELECT * FROM receiver"
  stmt = ibm_db.exec_immediate(conn, sql)
  dictionary = ibm_db.fetch_both(stmt)
  while dictionary != False:
    # print ("The Name is : ",  dictionary)
    users.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
  if users:
    return render_template("list.html",users = users)
  return "success..."
@app.route('/success/<email>')
def success(email):
  from_email = Email("sherazjabeen18@gmail.com")
  to_email = To(email)
  subject = "From SharingJoy request for donor"
  content = f'http://127.0.0.1:5000/donorsuccess'
  mail = Mail(from_email, to_email, subject, content)
  response = sg.client.mail.send.post(request_body=mail.get())
  print(response.status_code)
  print(response.body)
  print(response.headers)
  return render_template("success.html")
@app.route('/donorsuccess')
def donorsuccess():
  return render_template("donorsuccess.html")
@app.route('/accept',methods = ['POST', 'GET'])
def accept():
  if request.method == 'POST':
    username = request.form['username']
    insert_sql = "INSERT INTO acceptorname VALUES (?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prep_stmt, 1, username)
    ibm_db.execute(prep_stmt)
    return render_template("donorsuccess.html")
@app.route('/acceptlist')
def acceptlist():
  users = []
  sql = "SELECT * FROM acceptorname"
  stmt = ibm_db.exec_immediate(conn, sql)
  dictionary = ibm_db.fetch_both(stmt)
  while dictionary != False:
    # print ("The Name is : ",  dictionary)
    users.append(dictionary)
    dictionary = ibm_db.fetch_both(stmt)
  if users:
    return render_template("acceptlist.html",users = users)
@app.route('/delete/<username>')
def delete(username):
  sql = f"SELECT * FROM acceptorname WHERE username='{escape(username)}'"
  print(sql)
  stmt = ibm_db.exec_immediate(conn, sql)
  donation = ibm_db.fetch_row(stmt)
  #print ("The Name is : ",  student)
  if donation:
    sql = f"DELETE FROM acceptorname WHERE username='{escape(username)}'"
    #print(sql)
    stmt = ibm_db.exec_immediate(conn, sql)
    donation = []
    sql = "SELECT * FROM acceptorname"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
      donation.append(dictionary)
      dictionary = ibm_db.fetch_both(stmt)
    if donation:
      return redirect("/acceptlist")
if __name__ == '__main__':
    app.run(debug=True)
```

home.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</title>
</head>
<body style="background-image:url('https://c1.wallpaperflare.com/preview/289/93/320/blood-blood-donation-health-donation.jpg'); background-size:cover;">
    {{msg}}
    <h2>Hello This is our HOME PAGE</h2><br>
    <a href="/login">Back</a><br>
    <a href="/requestform">Request for Plasma</a>
    <br><a href="/list">List of Donors</a>
</body>
</html>
```

login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="apple-touch-icon" type="image/png" href="https://cpwebassets.codepen.io/assets/favicon/apple-touch-icon-5ae1a0698dcc2402e9712f7d01ed509a57814f994c660df9f7a952f3060705ee.png" />
    <meta name="apple-mobile-web-app-title" content="CodePen">
    <link rel="shortcut icon" type="image/x-icon"
href="https://cpwebassets.codepen.io/assets/favicon/favicon-
aec34940fbc1a6e787974dcd360f2c6b63348d4b1f4e06c77743096d55480f33.ico" />
    <link rel="mask-icon" type="image/x-icon"
href="https://cpwebassets.codepen.io/assets/favicon/logo-pin-
8f3771b1072e3c38bd662872f6b673a722f4b3ca2421637d5596661b4e2132cc.svg" color="#111"
/>
    <link rel="stylesheet" href="static/css/login_style.css">
    <title>Plasma Donor</title>
</head>
<body>
    <div id="container" class="container">
        <!-- FORM SECTION -->
        <div class="row">
            <!-- SIGN UP -->
            <div class="col align-items-center flex-col sign-up">
                <form action="/" method="POST">
                    <div class="form-wrapper align-items-center">
                        <div class="form sign-up">
                            <div class="input-group">
                                <i class='bx bxs-user'></i>
                                <input type="text" placeholder="Username" name="username">
                            </div>
                            <div class="input-group">
                                <i class='bx bx-mail-send'></i>
                                <input type="email" placeholder="Email" name="email">
                            </div>
                            <div class="input-group">
                                <i class='bx bxs-user'></i>
                                <input type="text" placeholder="Blood Group (eg. O+)"
name="bloodgroup">
                            </div>
                            <div class="input-group">
                                <i class='bx bxs-lock-alt'></i>
                                <input type="password" placeholder="Password" name="password">
                            </div>
                            <button>Sign up</button>
                            <p><span>Already have an account?</span>
                                <b onclick="toggle()" class="pointer">Sign in here</b>
                            </p>
                        </div>
                    </div>
                </form>
            </div>
            <!-- END SIGN UP -->
            <!-- SIGN IN -->
            <div class="col align-items-center flex-col sign-in">
                <form action="/login" method="POST">
                    <div class="form-wrapper align-items-center">
                        <div class="form sign-in">
                            <div class="input-group">
                                <i class='bx bxs-user'></i>
                                <input type="email" placeholder="Email" name="email">
                            </div>
                            <div class="input-group">
                                <i class='bx bxs-lock-alt'></i>
                                <input type="password" placeholder="Password" name="password">
                            </div>
                            <button>Sign in</button>
                            <p><b>Forgot password?</b>
                            </p>
                            <p>
                                <span>Don't have an account?</span>
                                <b onclick="toggle()" class="pointer">Sign up here
</b>
                            </p>
                        </div>
                    </form>
                </div>
                <div class="form-wrapper">
                </div>
            </div>
            <!-- END SIGN IN -->
        </div>
        <!-- END FORM SECTION -->
        <!-- CONTENT SECTION -->
        <div class="row content-row">
            <!-- SIGN IN CONTENT -->
            <div class="col align-items-center flex-col">
                <div class="text sign-in">
                    <h2 style="font-size: 35px;">
```

```
                        Welcome to Plasma Donor<br>Login Now...<br><br><br>
                    </h2>
                </div>
                <div class="img sign-in">
                </div>
            </div>
            <!-- END SIGN IN CONTENT -->
            <!-- SIGN UP CONTENT -->
            <div class="col align-items-center flex-col">
                <div class="img sign-up">
                </div>
                <div class="text sign-up">
                    <h2 style="font-size: 35px;">
                        Still not a part of our community?
                        <br>Now its the right time<br> So Join Us Now!....
                    </h2>
                </div>
            </div>
            <!-- END SIGN UP CONTENT -->
        </div>
        <!-- END CONTENT SECTION -->
    </div>
    <script src="https://cpwebassets.codepen.io/assets/common/stopExecutionOnTimeout-2c7831bb44f98c1391d6a4ffda0e1fd302503391ca806e7fcc7b9b87197aec26.js"></script>
    <!-- END SIGN IN -->
    </div>
    <!-- END FORM SECTION -->
```

```
<!-- CONTENT SECTION -->
<div class="row content-row">
    <!-- SIGN IN CONTENT -->
    <div class="col align-items-center flex-col">
        <div class="text sign-in">
            <h2>
                Hey User Ready to donate your plasma<br>Login Now...
            </h2>
        </div>
        <div class="img sign-in">
        </div>
    </div>
    <!-- END SIGN IN CONTENT -->
    <!-- SIGN UP CONTENT -->
    <div class="col align-items-center flex-col">
        <div class="img sign-up">
        </div>
        <div class="text sign-up">
            <h2>
                Join with us
            </h2>
        </div>
    </div>
    <!-- END SIGN UP CONTENT -->
</div>
<!-- END CONTENT SECTION -->
</div>
```

```
    <script src="https://cpwebassets.codepen.io/assets/common/stopExecutionOnTimeout-2c7831bb44f98c1391d6a4ffda0e1fd302503391ca806e7fcc7b9b87197aec26.js"></script>
    <script>
        window.console = window.console || function(t) {};
        if (document.location.search.match(/type=embed/gi)) {
            window.parent.postMessage("resize", "*");
        }
        let container = document.getElementById('container');
        toggle = () => {
            container.classList.toggle('sign-in');
            container.classList.toggle('sign-up');
        };
        setTimeout(() => {
            container.classList.add('sign-in');
        }, 200);
    </script>
</body>
</html>
```

donate.html

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>SharingJoy Plasma Donation</title>
    <!-- CSS FILES -->
    <link href="static/css/bootstrap.min.css" rel="stylesheet">
    <link href="static/css/bootstrap-icons.css" rel="stylesheet">
    <link href="static/css/templatemo-kind-heart-charity.css" rel="stylesheet">
    <!--
TemplateMo 581 Kind Heart Charity
https://templatemo.com/tm-581-kind-heart-charity
-->
</head>
<body id="section_1">
    <nav class="navbar navbar-expand-lg bg-light shadow-lg">
        <div class="container">
            <a class="navbar-brand" href="index.html">
                <img src="static/images/logotp.png" class="logo img-fluid" alt="SharingJoy">
                <span>
                    SharingJoy Plasma Donation
                    <small>Non-profit Organization</small>
                </span>
            </a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav ms-auto">
                    <li class="nav-item">
```

```
                        <a class="nav-link click-scroll" href="#top">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link click-scroll" href="#section_2">About</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link click-scroll" href="#section_3">Our Precious Gems</a>
                    </li>
                    <li class="nav-item ms-3">
                        <a class="nav-link custom-btn custom-border-btn btn" href="/requestform">Donate</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
    <main>
        <section class="hero-section hero-section-full-height">
            <div class="container-fluid">
                <div class="row">
                    <div class="col-lg-12 col-12 p-0">
                        <div id="hero-slide" class="carousel carousel-fade slide" data-bs-ride="carousel">
                            <div class="carousel-inner">
                                <div class="carousel-item active">
                                    <img src="https://communitymedicine4asses.files.wordpress.com/2022/06/banner-02-1920x1080-datelogos.png?w=1200" class="carousel-image img-fluid" alt="...">
```

```
                  <div class="carousel-caption d-flex flex-column justify-content-end">
                    <h1>Sharing is Caring</h1>
                    <p>be a reason for other's Joy</p>
                  </div>
                </div>
                <div class="carousel-item">
                  <img src="https://wallpapercave.com/wp/wp4323504.jpg" alt="...">
                  <div class="carousel-caption d-flex flex-column justify-content-end">
                    <h1>Help the needy</h1>
                    <p>You can support them to live long</p>
                  </div>
                </div>
                <div class="carousel-item">
                  <img src="https://www.redcross.org/content/dam/redcross/about-us/news/2021/731501tier_369what_is_ccp_003.png.transform/1288/q70/feature/image.png" class="carousel-image img-fluid" alt="...">
                  <div class="carousel-caption d-flex flex-column justify-content-end">
                    <h1>Humanity</h1>
                    <p>Your little effort can save a life.</p>
                  </div>
                </div>
              </div>
              <button class="carousel-control-prev" type="button" data-bs-target="#hero-slide" data-bs-slide="prev">
                <span class="carousel-control-prev-icon" aria-hidden="true"></span>
                <span class="visually-hidden">Previous</span>
              </button>
              <button class="carousel-control-next" type="button" data-bs-target="#hero-
```

```
slide" data-bs-slide="next">
                                <span class="carousel-control-next-icon" aria-hidden="true"></span>
                                <span class="visually-hidden">Next</span>
                            </button>
                        </div>
                    </div>
                </div>
            </div>
        </section>
        <section class="section-padding">
            <div class="container">
                <div class="row">
                    <div class="col-lg-10 col-12 text-center mx-auto">
                        <h2 class="mb-5">Welcome to SharingJoy Plasma Donation</h2>
                    </div>
                    <div class="col-lg-3 col-md-6 col-9 mb-3 mb-lg-0">
                        <div class="featured-block d-flex justify-content-center align-items-center">
                            <a href="donate.html" class="d-block">
                                <img src="static/images/icons/hands.png" class="featured-block-image img-fluid" alt="">
                                <p class="featured-block-text">Become a <strong>volunteer</strong></p>
                            </a>
                        </div>
                    </div>
                    <div class="col-lg-3 col-md-6 col-9 mb-3 mb-lg-0 mb-md-3">
                        <div class="featured-block d-flex justify-content-center align-items-center">
                            <a href="donate.html" class="d-block">
```

```
                                <img src="static/images/icons/heart.png" class="featured-block-image img-fluid" alt="">

                                <p class="featured-block-text"><strong>Caring</strong> Earth</p>

                            </a>

                        </div>

                    </div>

                    <div class="col-lg-3 col-md-6 col-9 mb-3 mb-lg-0">

                        <div class="featured-block d-flex justify-content-center align-items-center">

                            <a href="donate.html" class="d-block">

                                <img src="static/images/icons/scholarship.png" class="featured-block-image img-fluid" alt="">

                                <p class="featured-block-text"><strong>Scholarship</strong> Program</p>

                            </a>

                        </div>

                    </div>

                </div>

            </div>

        </section>

        <section class="section-padding section-bg" id="section_2">

            <div class="container">

                <div class="row">

                    <div class="col-lg-6 col-12 mb-5 mb-lg-0">

                        <img src="static/images/group-people-volunteering-foodbank-poor-people.jpg" class="custom-text-box-image img-fluid" alt="">

                    </div>

                    <div class="col-lg-6 col-12">

                        <div class="custom-text-box">

                            <h2 class="mb-2">Our Story</h2>
                            <h5 class="mb-3">Kind Heart Charity, Non-Profit Organization</h5>

                            <p class="mb-0">This is a Bootstrap 5.2.2 CSS template for charity organization websites. You can feel free to use it. Please tell your friends about TemplateMo website. Thank you.

                            </p>

                        </div>

                        <div class="row">

                            <div class="col-lg-6 col-md-6 col-12">

                                <div class="custom-text-box mb-lg-0">

                                    <h5 class="mb-3">Our Mission</h5>

                                    <p>Sed leo nisl, posuere at molestie ac, suscipit auctor quis metus</p>

                                    <ul class="custom-list mt-2">

                                        <li class="custom-list-item d-flex">

                                            <i class="bi-check custom-text-box-icon me-2"></i> Charity Theme

                                        </li>

                                        <li class="custom-list-item d-flex">

                                            <i class="bi-check custom-text-box-icon me-2"></i> SemanticHTML

                                        </li>

                                    </ul>

                                </div>

                            </div>

                            <div class="col-lg-6 col-md-6 col-12">

                                <div class="custom-text-box d-flex flex-wrap d-lg-block mb-lg-0">

                                    <div class="counter-thumb">

                                        <div class="d-flex">

                                            <span class="counter-number" data-from="1" data-to="2009" data-speed="1000"></span>
                                            <span class="counter-number-text"></span>
                                        </div>
                                        <span class="counter-text">Founded</span>
                                    </div>
                                    <div class="counter-thumb mt-4">
                                        <div class="d-flex">
                                            <span class="counter-number" data-from="1" data-to="120" data-speed="1000"></span>
                                            <span class="counter-number-text">B</span>
                                        </div>
                                        <span class="counter-text">Donations</span>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </section>
        <section class="cta-section section-padding section-bg">
            <div class="container">
                <div class="row justify-content-center align-items-center">
                    <div class="col-lg-5 col-12 ms-auto">
                        <h2 class="mb-0">Make an impact. <br> Save lives.</h2>
                    </div>
                    <div class="col-lg-5 col-12">
                        <a href="#" class="me-4">Make a donation</a>
                        <a href="#section_4" class="custom-btn btn smoothscroll">Become a volunteer</a>
                    </div>
                </div>
            </div>
        </section>
        <section class="section-padding" id="section_3">
            <div class="container">
                <div class="row">
                    <div class="col-lg-12 col-12 text-center mb-4">
                        <h2>Our Causes</h2>
                    </div>
                    <div class="col-lg-4 col-md-6 col-12 mb-4 mb-lg-0">
                        <div class="custom-block-wrap">
                            <img src="static/images/causes/group-african-kids-paying-attention-class.jpg" class="custom-block-image img-fluid" alt="">
                            <div class="custom-block">
                                <div class="custom-block-body">
                                    <h5 class="mb-3">Children Education</h5>
                                    <p>Lorem Ipsum dolor sit amet, consectetur adipsicing kengan omeg kohm tokito</p>
                                </div>
                            </div>
                        </div>
                    </div>
                    <div class="col-lg-4 col-md-6 col-12 mb-4 mb-lg-0">
                        <div class="custom-block-wrap">
                            <img src="static/images/causes/poor-child-landfill-looks-forward-with-
```

```
hope.jpg" class="custom-block-image img-fluid" alt="">
              <div class="custom-block">
                <div class="custom-block-body">
                  <h5 class="mb-3">Poverty Development</h5>
                  <p>Sed leo nisl, posuere at molestie ac, suscipit auctor mauris. Etiam quis
metus tempor
                  </p>
                </div>
              </div>
            </div>
          </div>
          <div class="col-lg-4 col-md-6 col-12">
            <div class="custom-block-wrap">
              <img src="static/images/causes/african-woman-pouring-water-recipient-
outdoors.jpg" class="custom-block-image img-fluid" alt="">
              <div class="custom-block">
                <div class="custom-block-body">
                  <h5 class="mb-3">Supply drinking water</h5>
                  <p>Orci varius natoque penatibus et magnis dis parturient montes,
nascetur ridiculus
                  </p>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </section>
```

```
<script src="static/js/chat.js">
</script>
<footer class="site-footer">
    <div class="container">
        <div class="row">
            <div class="col-lg-3 col-12 mb-4">
                <img src="static/images/logotp.png" class="logo img-fluid" alt="">
            </div>
            <div class="col-lg-4 col-md-6 col-12 mb-4">
                <h5 class="site-footer-title mb-3">Quick Links</h5>
                <ul class="footer-menu">
                    <li class="footer-menu-item"><a href="#" class="footer-menu-link">Our Story</a></li>
                    <li class="footer-menu-item"><a href="#" class="footer-menu-link">Newsroom</a></li>
                    <li class="footer-menu-item"><a href="#" class="footer-menu-link">Causes</a></li>
                    <li class="footer-menu-item"><a href="#" class="footer-menu-link">Become a volunteer</a></li>
                    <li class="footer-menu-item"><a href="#" class="footer-menu-link">Partner with us</a></li>
                </ul>
            </div>
            <div class="col-lg-4 col-md-6 col-12 mx-auto">
                <h5 class="site-footer-title mb-3">Contact Infomation</h5>
                <p class="text-white d-flex mb-2">
                    <i class="bi-telephone me-2"></i>
                    <a href="tel: 305-240-9671" class="site-footer-link">
```

```
                            305-240-9671
                        </a>
                        </p>
                        <p class="text-white d-flex">
                            <i class="bi-envelope me-2"></i>
                            <a href="mailto:info@yourgmail.com" class="site-footer-link">
                            donate@charity.org
                        </a>
                        </p>
                        <p class="text-white d-flex mt-3">
                            <i class="bi-geo-alt me-2"></i> Akershusstranda 20, 0150 Oslo, Norway
                        </p>
                        <a href="#" class="custom-btn btn mt-3">Get Direction</a>
                    </div>
                </div>
            </div>
            <div class="site-footer-bottom">
                <div class="container">
                    <div class="row">
                        <div class="col-lg-6 col-md-7 col-12">
                            <p class="copyright-text mb-0">Copyright © 2036 <a href="#">Kind Heart</a> Charity Org. Design: <a href="https://templatemo.com" target="_blank">TemplateMo</a><br>Distribution:
                                <a href="https://themewagon.com">ThemeWagon</a>
                            </p>
                        </div>
                        <div class="col-lg-6 col-md-5 col-12 d-flex justify-content-center align-items-center mx-auto">
```

```
                            <ul class="social-icon">
                                <li class="social-icon-item">
                                    <a href="#" class="social-icon-link bi-twitter"></a>
                                </li>
                                <li class="social-icon-item">
                                    <a href="#" class="social-icon-link bi-facebook"></a>
                                </li>
                                <li class="social-icon-item">
                                    <a href="#" class="social-icon-link bi-instagram"></a>
                                </li>
                                <li class="social-icon-item">
                                    <a href="#" class="social-icon-link bi-linkedin"></a>
                                </li>
                                <li class="social-icon-item">
                                    <a href="https://youtube.com/templatemo" class="social-icon-link bi-youtube"></a>
                                </li>
                            </ul>
                        </div>
                    </div>
                </div>
            </div>
        </footer>
        <!-- JAVASCRIPT FILES -->
        <script src="static/js/jquery.min.js"></script>
        <script src="static/js/bootstrap.min.js"></script>
        <script src="static/js/jquery.sticky.js"></script>
        <script src="static/js/click-scroll.js"></script>
        <script src="static/js/counter.js"></script>
        <script src="static/js/custom.js"></script>
</body>
</html>
```

Requestform.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>RequestForm</title>
</head>
<body>
    <form action="/requestform" method="POST">
        <br>
        <input type="text" placeholder="Name" name="username"> <br>
        <input type="number" placeholder="Age" name="age"> <br>
        <input type="text" placeholder="Gender" name="gender"> <br>
        <input type="text" placeholder="Blood Group" name="bloodgroup"> <br>
        <input type="email" placeholder="Email" name="email"> <br>
        <input type="text" placeholder="Address" name="address"> <br>
        <input type="text" placeholder="City" name="city"> <br>
        <input type="number" placeholder="Pincode" name="pincode"> <br>
        <input type="number" placeholder="Contact No" name="contactno"> <br>
    <input type="submit" name="submit">
  </form>
</body>
</html>
```

GitHub & Project Demo Link:

GitHub

https://github.com/IBM-EPBL/IBM-Project-21650-1659786891

Project Demo Link

https://youtu.be/Kq5aU8YGJG4