**PLASMA DONAR APPLICATION**

**A PROJECT REPORT BY**

**ANCY.J.V**

**HARINEE.N.M**

**KAVYA.B**

**PONMANI JENIBA.P**

**in partial fulfilment for the award of the degree**

**of**

**BACHELOR OF ENGINEERING**

**in**

**COMPUTER SCIENCE & ENGINEERING**

**ST.PETER’S COLLEGE OF ENGINEERING AND**

**TECHNOLOGY, AVADI, CHENNAI.**

**ANNA UNIVERSITY :: CHENNAI 600 025**

**NOV 2022**

**ANNA UNIVERSITY: CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report "**PLASMA DONAR APPLICATION**” is the bonafide work of "ANCY.J.V(112719104005),HARINEE.N.M(112719104012), KAVYA.B(112719104018), PONMANI JENIBA.P(112719104026)" "**J.WISELY JOE**"who carried out the project work under my supervision

**SIGNATURE SIGNATURE**

**MRS.PREETHY REBECCA MRS.WISELY JOE**

**HEAD OF THE DEPARTMENT MENTOR**

Computer Science & Engineering Computer Science & Engineering

St.Peter’s College of Engineering St.Peter’s College of Engineering

& Technology & Technology

Chennai-600054 Chennai-600054

**TABLE OF CONTENTS**

**1 INTRODUCTION**

1.1 Project Overview

1.2 Purpose

**2 LITERATURE SURVEY**

2.1 Existing problem

2.2 References

2.3 Problem Statement

**3 IDEATION & PROPOSED SOLUTION**

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

3.4 Problem Solution Fit

**4 REQUIREMENT ANALYSIS**

4.1Functional Requirement

4.2Non-Functional Requirement

**5 PROJECT DESIGN**

5.1Data Flow Diagrams

5.2Solution & Technical Architecture

5.3User Stories

**6 PROJECT PLANNING & SCHEDULING**

6.1Sprint Planning & Estimation

6.2Sprint Delivery Schedule

6.3Reports from JIRA

**7 CODING & SOLUTIONING**

7.1 Layer between Donor and Recipient

7.2 Chat bot integration

7.3 Sprints

7.4 Database Schema

**8 TESTING**

8.1Test Cases

8.2User Acceptance Testing

**9 RESULTS**

9.1Performance Metrics

**10 ADVANTAGES & DISADVANTAGES**

**11 CONCLUSION**

**12 FUTURE SCOPE**

**13 APPENDIX**

13.1.Source Code

13.2.Github & Project Demo Link

# 1.INTRODUCTION

* 1. **PROJECT OVERVIEW:**
     + Plasma is usually given to people who have experienced trauma, burns, or shock as well as people with severe liver disease or several clotting factor shortages. As a result, the patient's blood volume rises, aiding in blood coagulation and assisting in the prevention of shock. The need for recovered patients' plasma has grown along with the number of people with Covid-19 infection. Someone can fight against the virus with the help of the antibodies that are already present in their bodies.
     + Plasma donation saves lives, and communication between donors and blood/plasma facilities is essential to this. Plasma donation could benefit the most from smart applications if they are developed with the users' needs and preferences in mind. Smart apps are increasingly seen as a critical communication tool.

# PURPOSE:

This application's objective is It would be helpful to save the donor information and assist the less fortunate by informing the list of current donors. In response to the issue, an application that would collect donor information, store it, and notify donors upon request is being developed.

**2.LITERATURE SURVEY**

**2.1EXISTING PROBLEM:**

* + - Plasma became significantly more necessary during the COVID 19 crisis. Plasma donation rates have dropped from an already pitiful 20% to a pitiful 11% on average. Any decrease in donations is concerning given that the labor-intensive manufacturing procedure required to fractionate plasma into the treatments patients depend on can take 7–12 months.
    - Checking the donor history, including whether they had previously been infected and recovered as well as which donors are qualified to donate plasma, was a difficult task because it compounded the consequences of continuous decline. Additionally, it turned out to be a difficult task to save the information of healthy donors, contact interested patients, and connect donors with requestors.

# 2.2REFERENCES:

**1.In year 2021, “A Systematic Review & Design of Web-Based Blood Management System”.**

**AUTHORS**: Gokul Dudani, Tanushree, Kajal Singh, Anushka Singh Chauhan.

Blood is a fluid that carries oxygen and is a connective tissue that carries other substances because of its volumeNow that we understand the importance of blood, we see that it not only carries oxygen.. n to the tissues but also clears the air between them through the heart and blood vessels. The average volume of blood donation is 470ml per person, which is only 8% of the adult volume. When blood is needed in a hospital, it is usually not available in time, leading to inconsistencies. Both patients and sponsors are unaware that the donor is being hospitalized due to a lack of communication and other services. A system like this is needed to close the communication gap between hospitals, blood banks, donors, and receptors. The main purpose of a web-based blood donation program is to ensure compliance with blood stock. In today's system, first and foremost the hand system, and when a person needs a blood type and that type is not available in that blood bank, it takes time to process blood from another blood bank, which may adversely affect the patient's health because time is critical in emergencies. Therefore, a web-based blood donation system is a good place to monitor whether a particular type of blood is available in a stack or not, as well as to provide a place where blood can be accessed.

**2.In year 2021, "Web Based Online Blood Donation System".**

**AUTHORS**: R. Kumar, R. Kumar and M. Tyagi.

This paper depicts a high level program to close the hole between blood givers and individuals needing blood. The online blood donation administration framework application is an approach to synchronize blood donation centers with emergency clinics with the assistance of the web. It is a web application where enlisted clinics can check the accessibility of the necessary blood and can send a blood solicitation to the closest blood donation center or comparable contributor as per the blood and can be controlled online through where fundamental. Blood donation center can likewise send a solicitation to another blood donation centers that isn't accessible. Anybody willing to give blood can be found at the closest blood donation center utilizing the android bank the executive framework. Blood donation center can be followed utilizing maps. The android application is simply accessible to benefactors to look for blood gifts and ask blood donation centers and clinics to search out blood donation center and close by givers.

**3. In year 2020, “Towards an Efficient and Secure Blood Bank Management System”.**

**AUTHORS**: P.A.J. Sandaruwan, U.D.L. Dolapihilla, D.W.N.R. Karunathilaka; W.A.D.T.L. Wijayaweera, W.H. Rankoth.

A blood bank plays an important role in a hospital as well as in a country, ensuring safe and timely blood transfusions. However, there are several challenges faced by blood banks around the world, specifically when securing the blood supply chain. Reducing the supply-demand imbalance, protecting the data privacy of donors as well as receivers, are some of them. Therefore, there is a timely requirement for an effective and secure management system for the blood bank. We have proposed a management platform for the blood bank operations with the following modules: (1) forecast blood demand, (2) suggest blood donation campaign locations and (3) secure blood supply chain. The proposed platform has been implemented using techniques such as Long Short-Term Memory (LSTM), k-means clustering, Geographic Information Systems (GIS), and block chain. Our results show that using our proposed modules, we can minimize the imbalance between supply and demand of blood, find the most suitable donor in an emergency, and enhance the privacy of data.

**4. In year 2018, “Automated blood bank system using Raspberry PI”.**

**AUTHORS**: Ashlesha C. Adsul, V. K. Bhosale, R. M. Autee .

“Raspberry pi based blood bank system” proposed to bring blood donors to the one place. The aim of this system is fulfil every blood request by using android application and raspberry pi. In the proposed system, data about the donors will be collected by using android application and raspberry pi by installing systems at places such as hospitals, blood banks etc. These data will be stored in the database. User/Patients needs to access application and needs to enter his requirements about the blood in the application the requirements are matched with the database and message will be to that particular blood donor through GSM modem.

**5 .In year 2016, “Short message service (SMS) based blood bank”.**

**AUTHORS**: G. Muddu Krishna & S. Nagaraju.

They proposed a system in which services of blood bank will be accessed via SMS. If someone needed blood then they have to request for blood via SMS and then packet count module of their system will check for availability of blood and response will be given by data processing module.

**2.3 PROBLEM STATEMENT:**

**Domain Name:** Cloud Application Development.

**Project Name:** Plasma Donor Application.

**Who does the problem affect?**

Patients with severe liver disorders and numerous clotting deficiencies are given plasma.

**What is the issue?**

When a patient needs plasma, it can be challenging to get in contact with a donor within the patient's family and friends in a timely manner. It can also be challenging to get in contact with authorized donor centers.

**What is the impact of the issue?**

Due to a lack of awareness regarding plasma donation, there is a demand for plasma donors, making it challenging for the affected patients to locate donors. During the COVID-19 pandemic, the need for plasma increased and the donor rates decreased in order to provide an immunity boost for COVID-affected patients.

**What would happen if we didn't solve the problem?**

It takes a long time for a patient to discover the proper donor, and it also takes time for the spreading about the need for plasma donors to disseminate on social media to a larger audience. As a result, patients cannot locate the right donor within a given time frame.

**What would happen when it is fixed?**

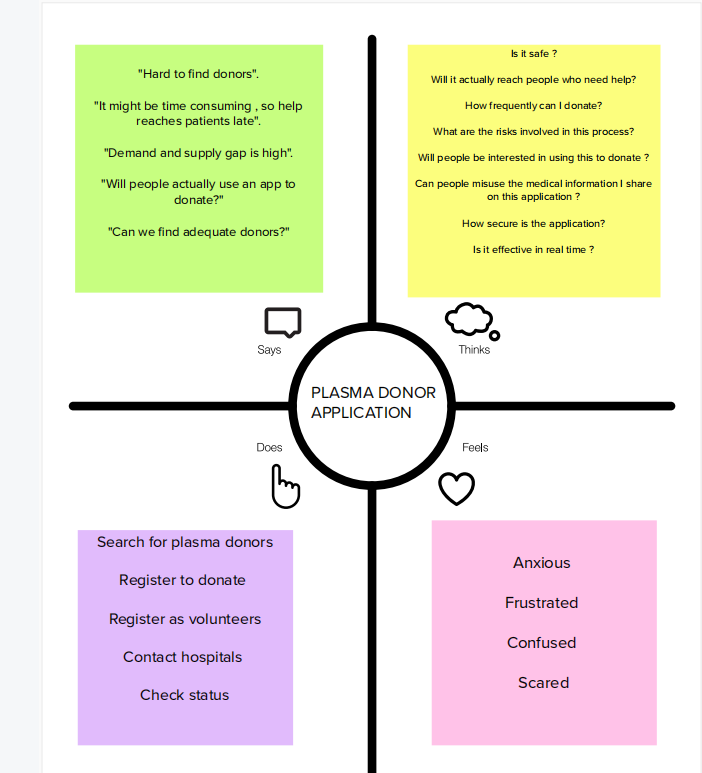
Our application enables patients with severe liver illnesses, blood clotting issues, and covid to quickly and easily locate the correct donor within the allotted period.

**Why is it important that we fix the problem?**

The condition of the affected patient may suffer and perhaps result in death, if the appropriate plasma donor cannot be found within a certain amount of time.

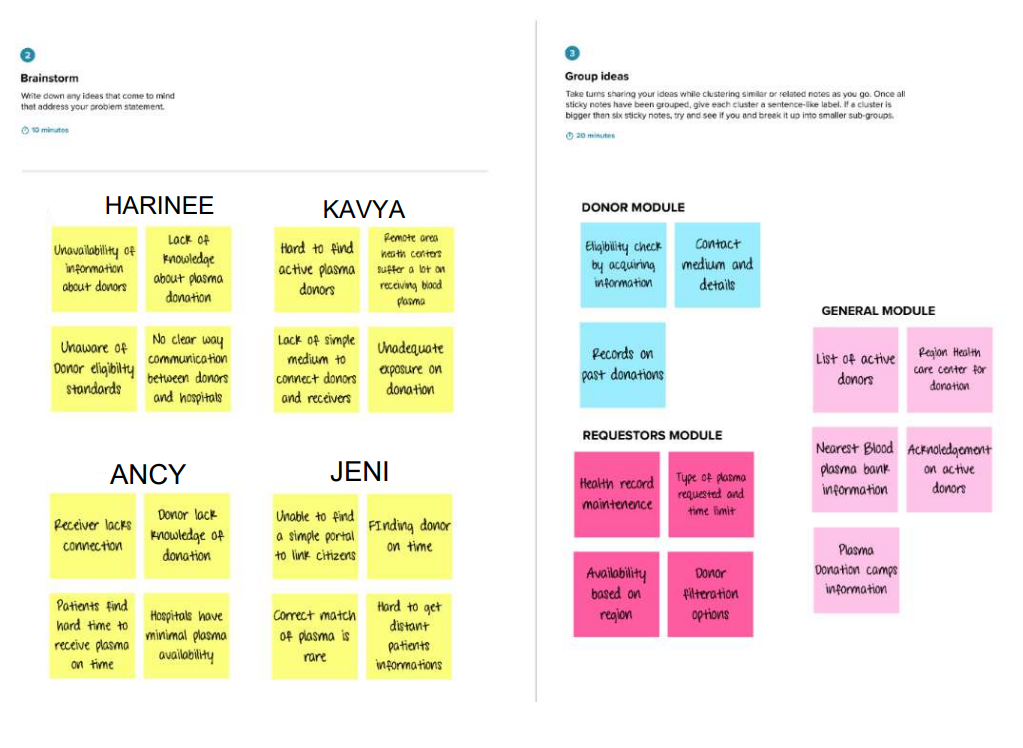
**3. IDEATION & PROPOSED SOLUTION**

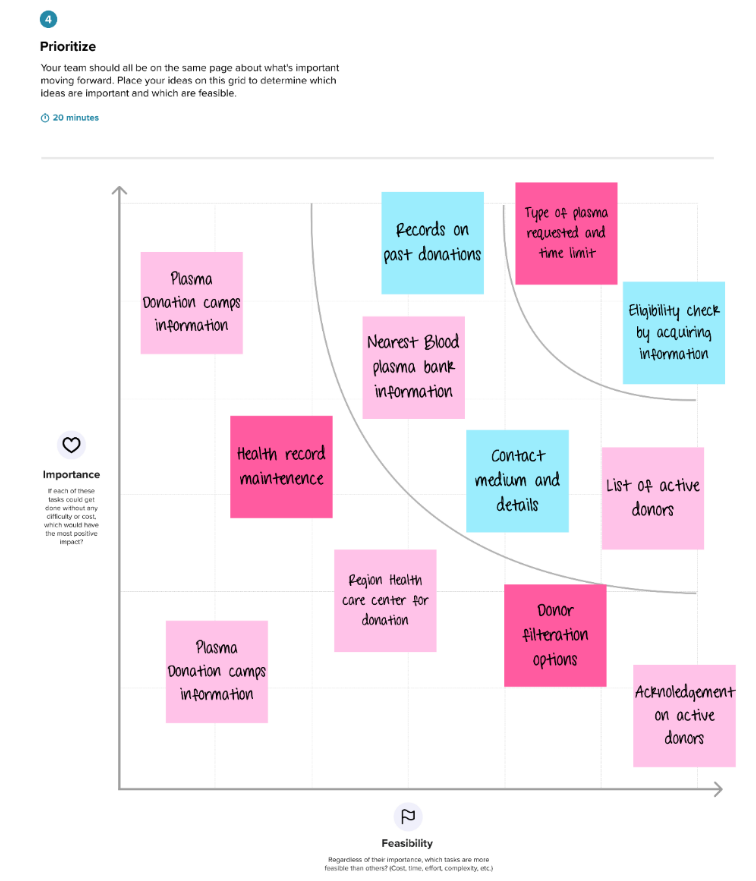
**3.1 EMPATHY MAP CANVAS:**

****

**3.3. IDEATION & BRAINSTORMING**

Brainstorming is a method design teams use to generate ideas to solve clearly defined design problems. Brainstorming is a method of generating ideas and sharing knowledge to solve a particular commercial or technical problem, in which participants are encouraged to think without interruption. Brainstorming is a group activity where each participant shares their ideas as soon as they come to mind. At the conclusion of the session, ideas are categorised and ranked for follow-on action. When planning a brainstorming session it is important to define clearly the topic to be addressed. A topic which is too specific can constrict thinking, while an ill-defined topic will not generate enough directly applicable ideas. The composition of the brainstorming group is important too. It should include people linked directly with the subject as well as those who can contribute novel and unexpected ideas. It can comprise staff from inside or outside the organisation. Step-1: Team Gathering,Collaboration and select the problem statement.





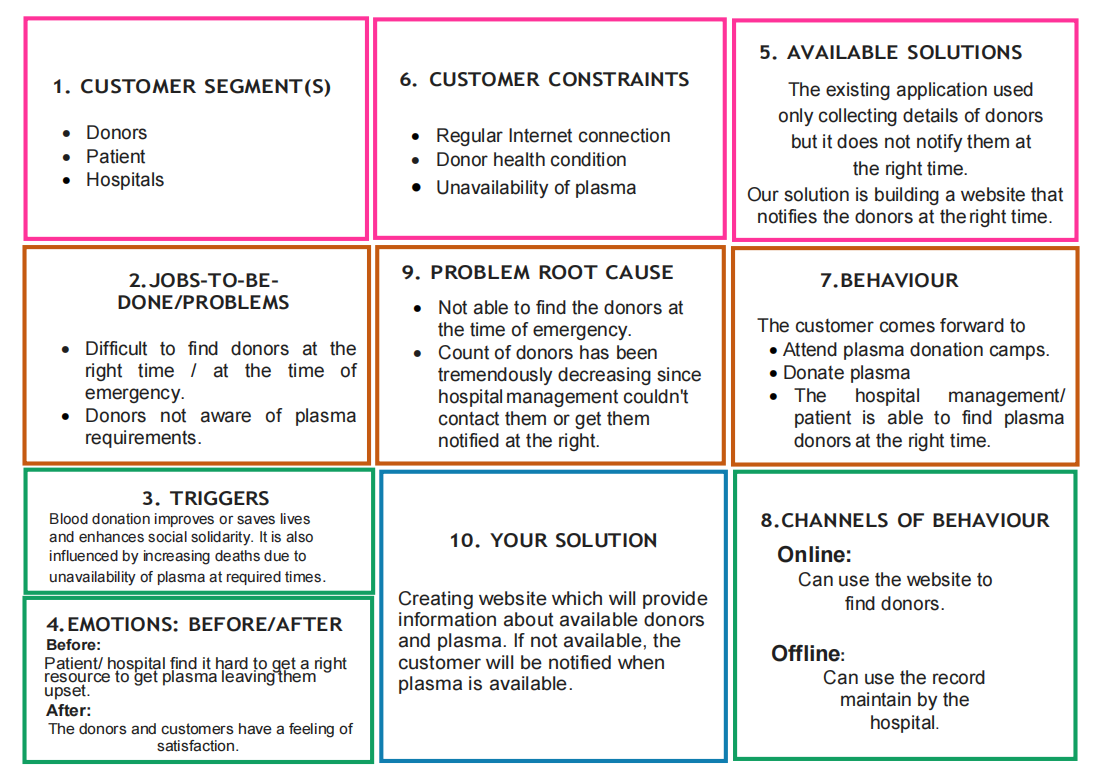
**3.4 PROPOSED SOLUTION:**

Proposed Solution means the technical solution to be provided by the implementation agency in response to the requirements and the objectives of the project.

|  |  |  |
| --- | --- | --- |
| **S.NO** | **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 mostly in times of emergency. |
| 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** | When the user requests for plasma transmission, if there is lack of plasma at the time of request, automatically user will be marked in hold back list. Later when there is availability of plasma, the receiver waiting in hold back list will be alerted via calling system. |
| 4. | **Social Impact / Customer Satisfaction** | The application is user friendly and anyone with basic knowledge can access it. The application seamlessly connects the donor and the person who need it and also hospitals who have availability of the plasma. |
| 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. Data can be stored in IBM DB2 in cloud which reduces the overall cost incurred for developing the application |
| 6. | **Scalability of the Solution** | Since the app is going to store its data in cloud, it will continue to be efficient when large number of people uses it. Also when the number of requests for plasma increases, the call notification system will work fine without any disruption. |

**3.4 PROBLEM SOLUTION FIT:**

Problem Solution Fit- this occurs when you have evidence that donor details are given ,At this stage you’ve proved the existence of a problem and have designed a value proposition that addresses donor information .

****

**4. REQUIREMENT ANALYSIS**

**4.1 FUNCTIONAL REQUIREMENT:**

Following are the functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| FR No. | Functional Requirement | Sub Requirement (Story / SubTask) |
| FR-1 | Access Website | An application browser or something similar should be able to allow software operators to access web applications. |
| FR-2 | Software operator Registration | The online application should allow the software operator to register. User name, gender, blood/plasma group, location, and contact information are required from the donor software operator. |
| FR-3 | Login/logout/update details | The database will store the login details for further use. |
| FR-4 | Search for donor | You can view the search results as a list. Each item on the list corresponds to a particular donor, complete with donor information. |
| FR-5 | User plasma request | Filling out the request form on the page will allow users to submit a request to donate plasma. They will receive an email once the request has been submitted. |
| FR-6 | View distribution details | The plasma bank should have access to the distribution details status. |

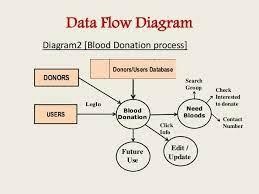
**4.2 NON-FUNCTIONAL REQUIREMENTS:**

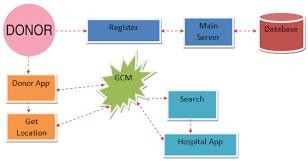
Following are the non-functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| FR No. | Non-Functional Requirement | Description |
| NFR-1 | Usability | The user interface of the plasma donor application needs to be attractive and functional. |
| NFR-2 | Security | Proper user names and passwords must be used to safeguard the plasma donor application |
| NFR-3 | Reliability | Even when errors occur, the plasma donor application should function successfully. |
| NFR-4 | Performance | The plasma donor application needs to function well under a variety of circumstances. |
| NFR-5 | Availability | The plasma donor application needs to be accessible around-the-clock without any bandwidth problems. |
| NFR-6 | Scalability | The performance and cost of the plasma donor application should be able to alter in response to modifications in the requirements for application and system processing. |

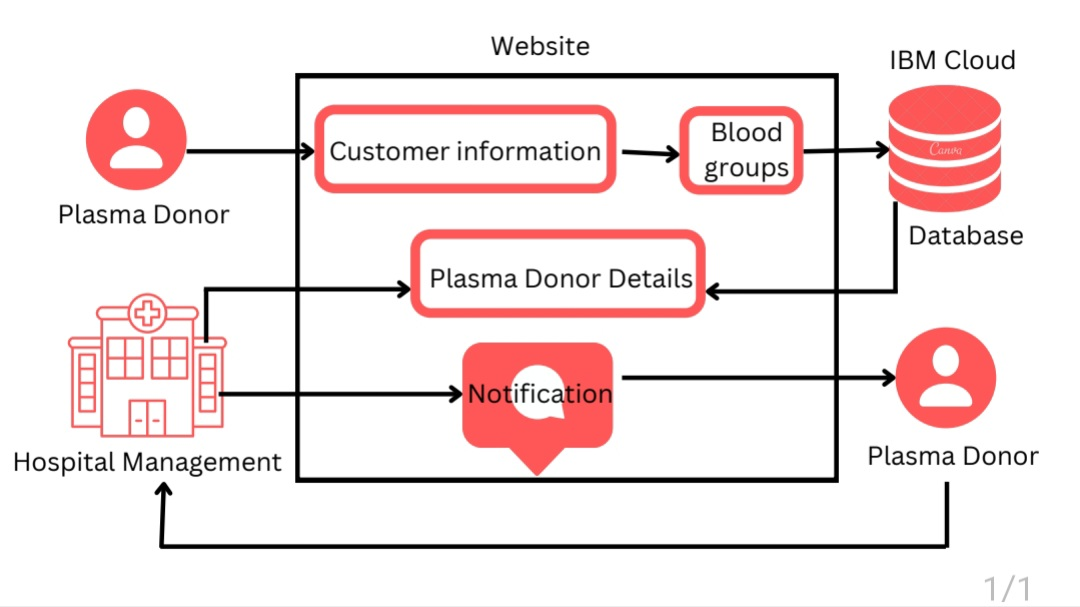
**5.PROJECT DESIGN**

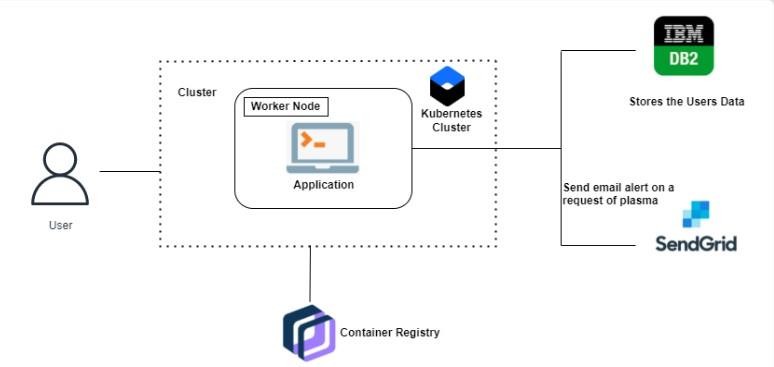
**5.1 DATA FLOW DIAGRAMS :**

****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 requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



**SOLUTION ARCHITECTURE:**

****

**TECHNOLOGY ARCHITECTURE:**

**USER STORIES:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| User Type | Functional Requirement | User Story Number | User Story/ Task | Acceptance Criteria | Priority | Release |
| Recipient | Registration | USN-1 | I can sign up for the application as a user by providing my email address, a password, and a password confirmation. | I can access my account / dashboard | High | Sprint-1 |
| Recipient | Verification | USN-2 | I will receive a confirmation email as a user once I've signed up for the application | I can receive confirmation email. | High | Sprint-1 |
| Recipient | Notification | USN-3 | The application will let me know whether there is a matching donor, as a user. | The app allows me to get notifications | Medium | Sprint-2 |
| Recipient | Notification | USN-4 | In a nearby hospital, I'll be able to get plasma. | I’ll be able to contact the Donor | High | Sprint-1 |
| Donor | Registration | USN-5 | I can sign up for the application as a user by providing my email address, a password, and a password confirmation. | I can access my account / dashboard | High | Sprint-1 |
| Donor | Verification I will receive a confirmation email as a user once I've signed up for the application | USN-6 | I will receive a confirmation email as a user once I've signed up for the application. | I can receive confirmation email. | High | Sprint-1 |
| Donor | Notification | USN-7 | I will receive notification of a plasma request through the application as a user. | The app allows me to get notifications. | Medium | Sprint-1 |
| Donor | Notification | USN-8 | I will be informed as a user of any nearby plasma blood banks. | I’ll be able to contact the Recipient. | Medium | Sprint-1 |

**PROJECT WORKFLOW:**

* The user interacts with the application.
* Registers by giving the details as a donor.
* The database will have all the details and if a user posts a request, then the concerned blood group donors will get notified about it.

**6.PROJECT PLANNING PHASE & SCHEDULING**

**6.1.SPRINT PLANNING AND ESTIMATION:**

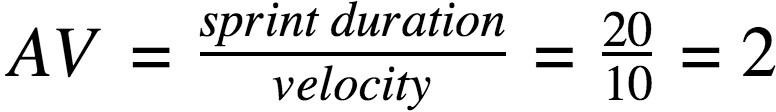
# Project Tracker, Velocity & Burndown Chart:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint | Total story points | Duration | Sprint start date | Sprint end date(Planned) | Stoty points completed(s on planned end date) | Sprint release date(Actual) |
| Sprint - 1 | 20 | 6 days | 24 Oct 2022 | 29 Oct 2022 | 20 | 29 Oct 2022 |
| Sprint -2 | 20 | 6 days | 31 Oct 2022 | 05Nov2022 | 20 | 05 Nov 2022 |
| Sprint-3 | 20 | 6 days | 07 Nov2022 | 12 Nov 2022 | 20 | 12 Nov2022 |
| Sprint-4 | 20 | 6 days | 14Nov 2022 | 19 Nov 2022 | 20 | 19 Nov 2022 |

# 

# Velocity:

Imagine we have a 10-day sprint duration, and the 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)



**Burndown chart:**

Chart

60

50

40

Column

Column

Story

week

week

week

week

10

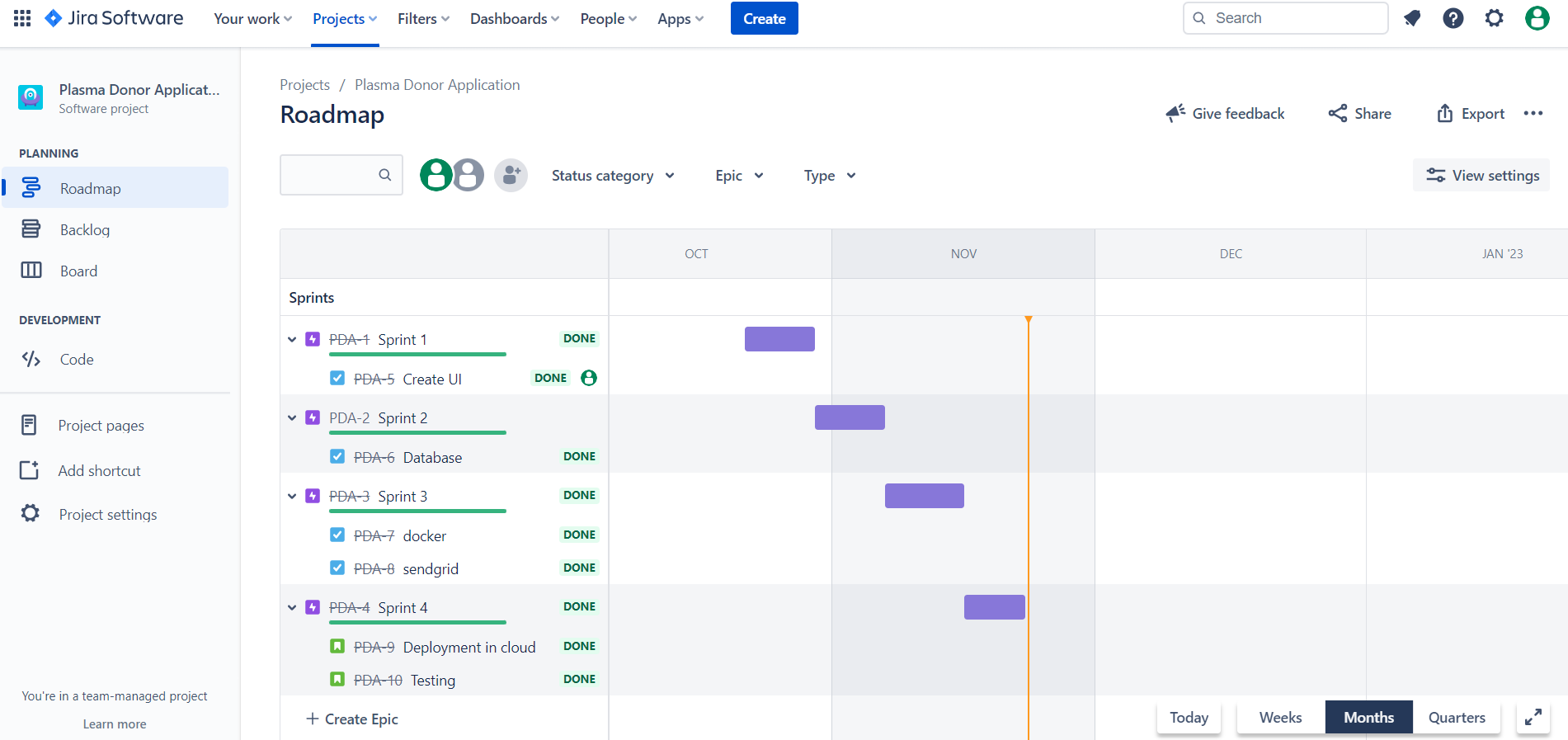
20

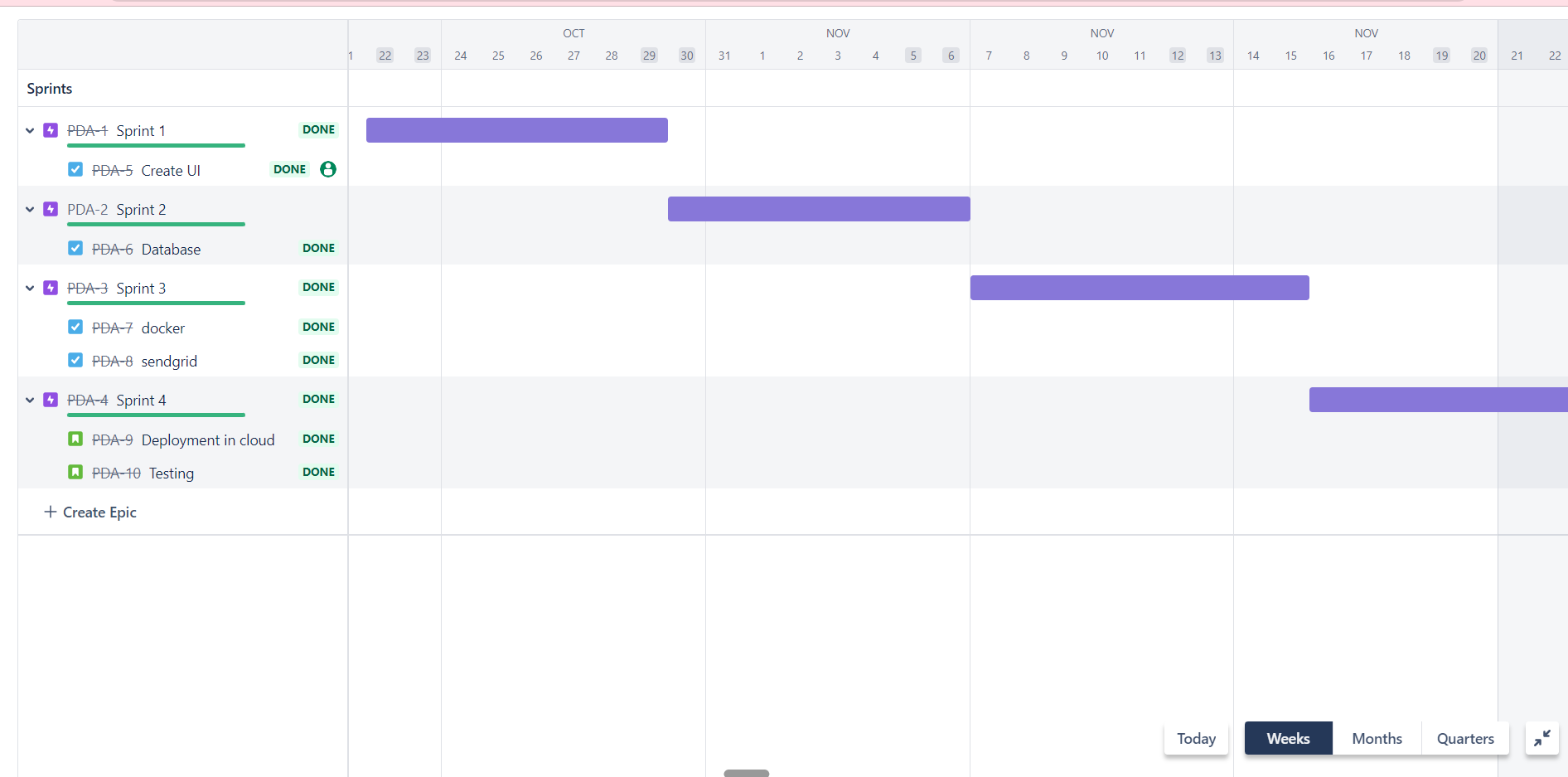
30

## 6.2 SPRINT DELIVERY SCHEDULE:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement**  **(Epic)** | **User Story**  **Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| Sprint-1 | Recipient Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my  password. | 10 | High | Harinee.N.M  Kavya.B |
| Sprint-1 | Recipient Verification | USN-2 | As a user, I will receive confirmation email once I have registered for the  application | 10 | High | Ancy  Jeni |
| Sprint-2 | Recipient  Notification | USN-3 | The application will let me know  whether there is a matching donor, as a user. | 10 | Medium | Kavya.B  Jeni |
| Sprint-2 | Recipient  Notification | USN-4 | In a nearby hospital, I'll be able to get plasma. | 10 | High | Harinee  Ancy |
| Sprint-3 | Donor Registration | USN-5 | I can sign up for the application as a user  by providing my email address, a  password, and a password confirmation. | 10 | High | Jeni  Harinee |
| Sprint-3 | Donor Verification | USN-6 | I will receive a confirmation email  as a user once I've signed up for the application. | 10 | High | Kavya  Ancy |
| Sprint-4 | Donor Notification | USN-7 | I will receive notification of a  plasma request through the application as a user. | 10 | Medium | Jeni  Kavya  Ancy |
| Sprint-4 | Donor Notification | USN-8 | I will be informed as a user of any nearby plasma blood banks. | 10 | Medium | Harinee  Jeni  Kavya  Ancy |

**6.3 REPORTS FROM JIRA:**





**7. CODING & SOLUTIONING:**

**7.1 Layer between Donor and Recipient:**

The application servers as a layer between donor and the recipient, donor can expose their information to the application, and they will be notified via email if there's a request for plasma with the same blood group.

**7.2 Chat bot integration**

The application has a chatbot integrated with it to help with basic user queries and to interact with the user. The chatbot feature is added to the application by using IBM Watson assistant in IBM cloud. This chatbot can interact with the user and guide them for simple queries.

## 7.3 Sprints:

**URL for Sprint – 1:** [**https://github.com/IBM-EPBL/IBM-Project-43204-1660714144/tree/main/Project%20Development%20Phase/Sprint%201**](https://github.com/IBM-EPBL/IBM-Project-43204-1660714144/tree/main/Project%20Development%20Phase/Sprint%201)

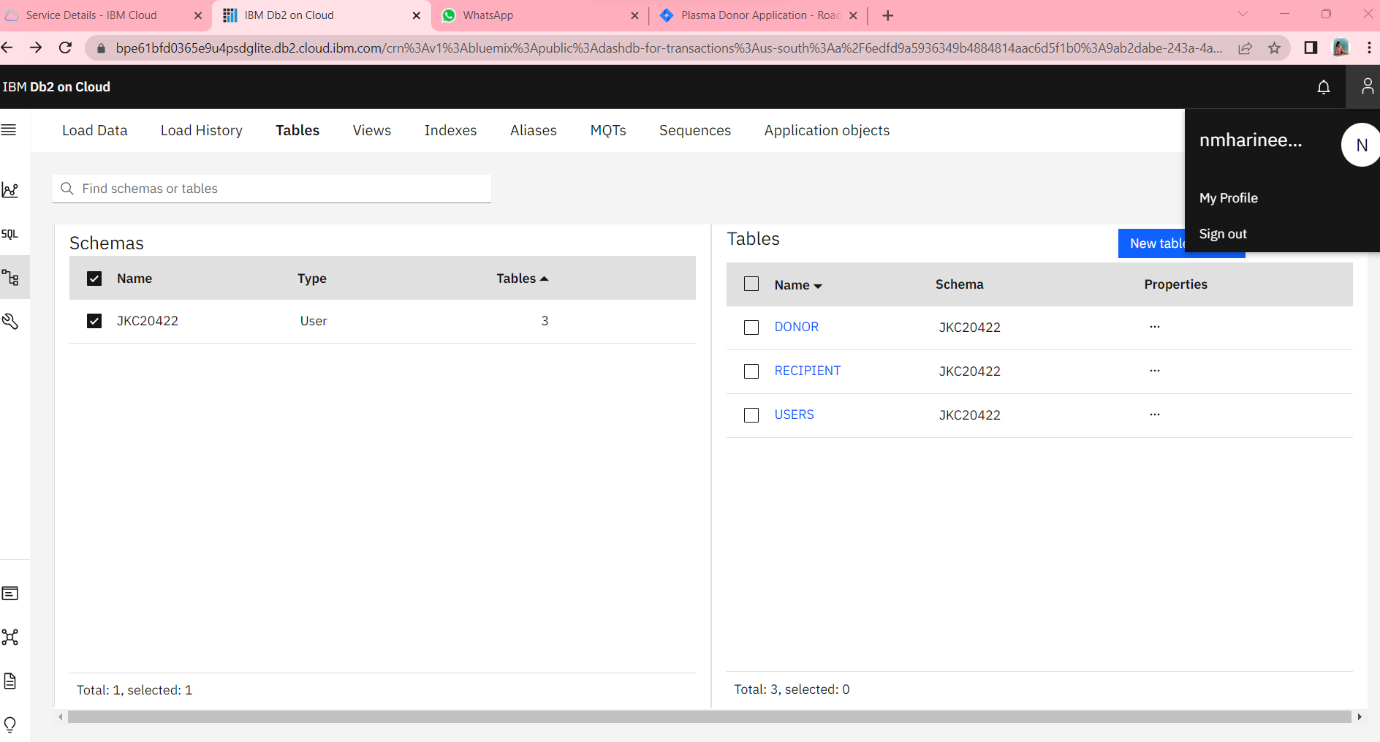
## 

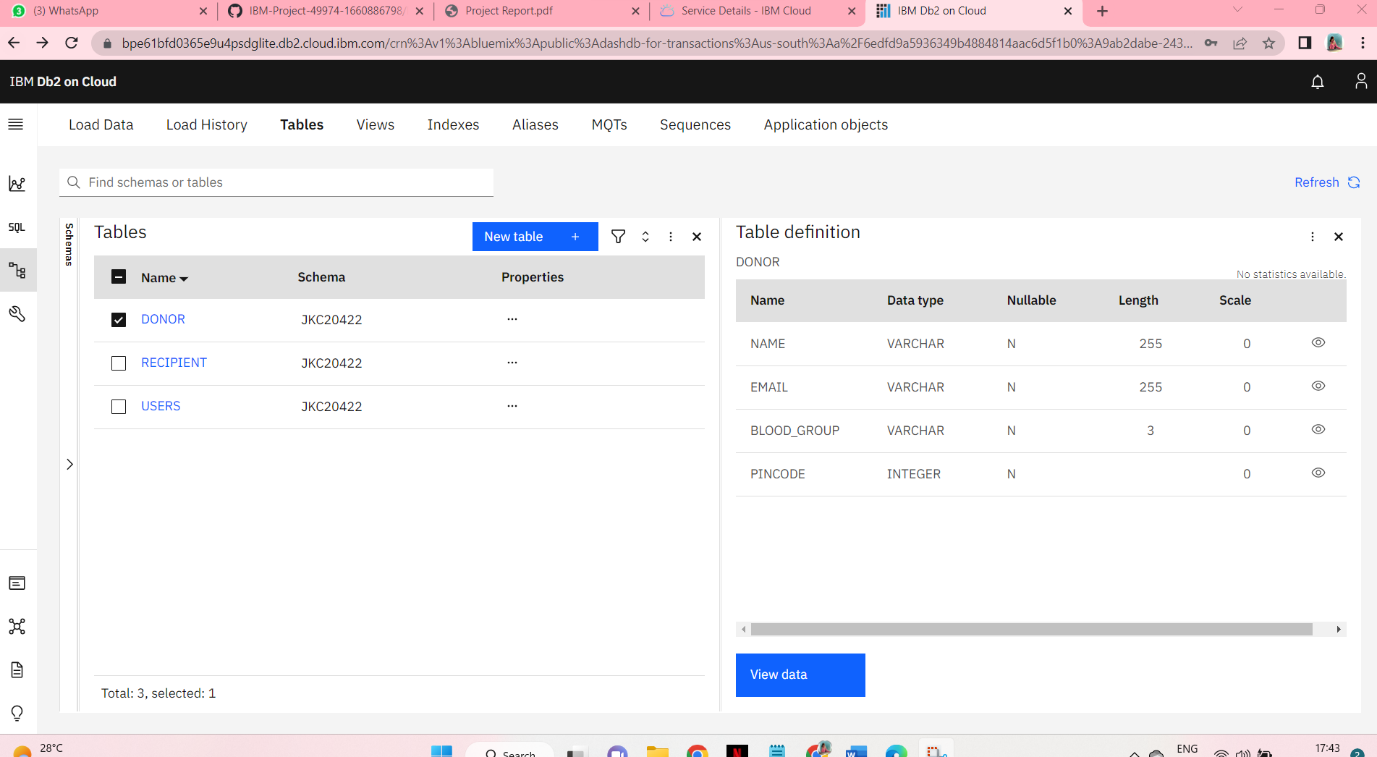
**URL for Sprint – 2:** [**https://github.com/IBM-EPBL/IBM-Project-43204-1660714144/tree/main/Project%20Development%20Phase/Sprint%202**](https://github.com/IBM-EPBL/IBM-Project-43204-1660714144/tree/main/Project%20Development%20Phase/Sprint%202)

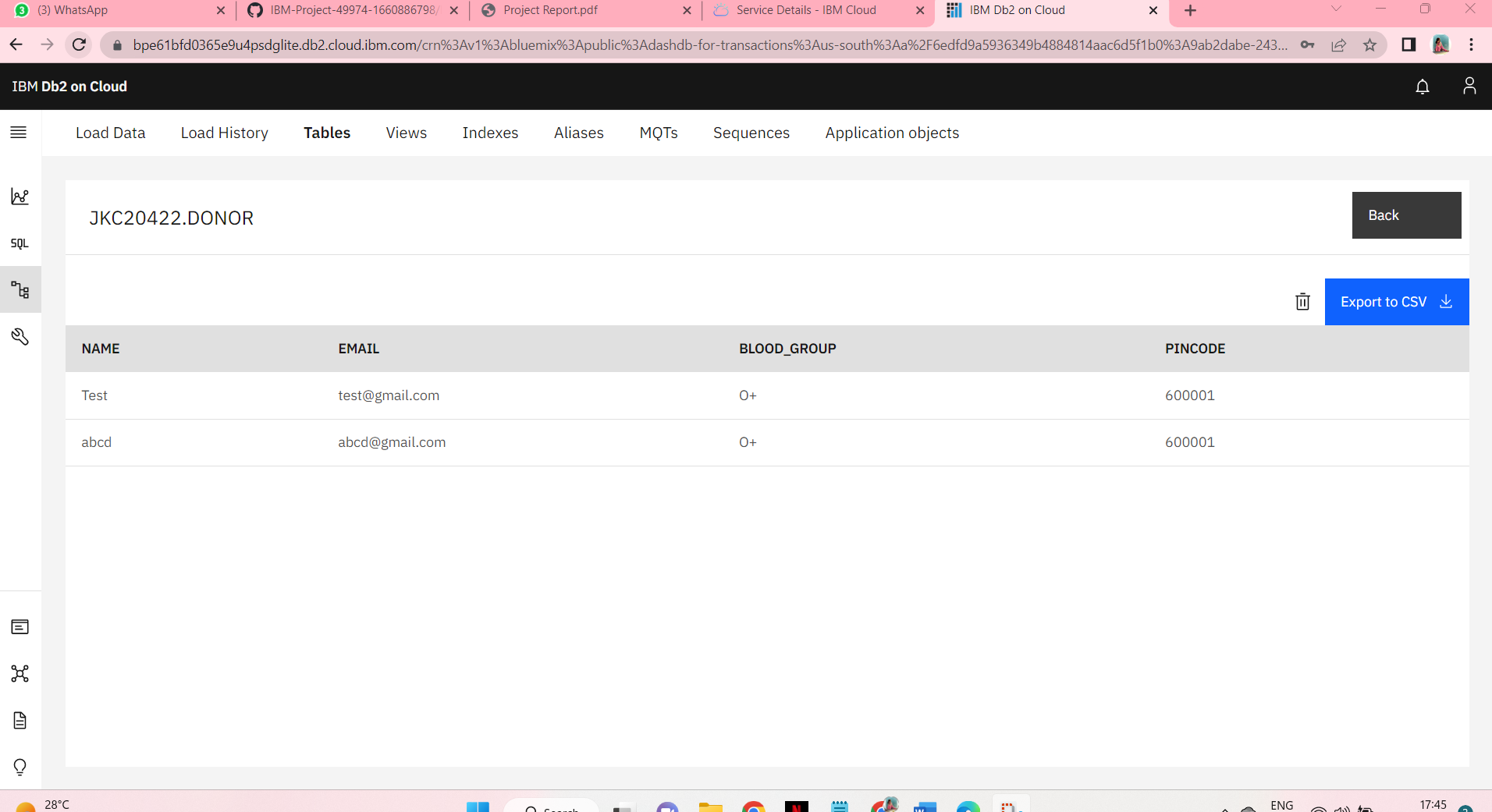
**URL for Sprint – 3:**[**https://github.com/IBM-EPBL/IBM-Project-43204-1660714144/tree/main/Project%20Development%20Phase/Sprint%203/Code**](https://github.com/IBM-EPBL/IBM-Project-43204-1660714144/tree/main/Project%20Development%20Phase/Sprint%203/Code)

**URL for Sprint – 4:**[**https://github.com/IBM-EPBL/IBM-Project-43204-1660714144/tree/main/Project%20Development%20Phase/Sprint%204/Code**](https://github.com/IBM-EPBL/IBM-Project-43204-1660714144/tree/main/Project%20Development%20Phase/Sprint%204/Code)

**7.4.Database Schema:**

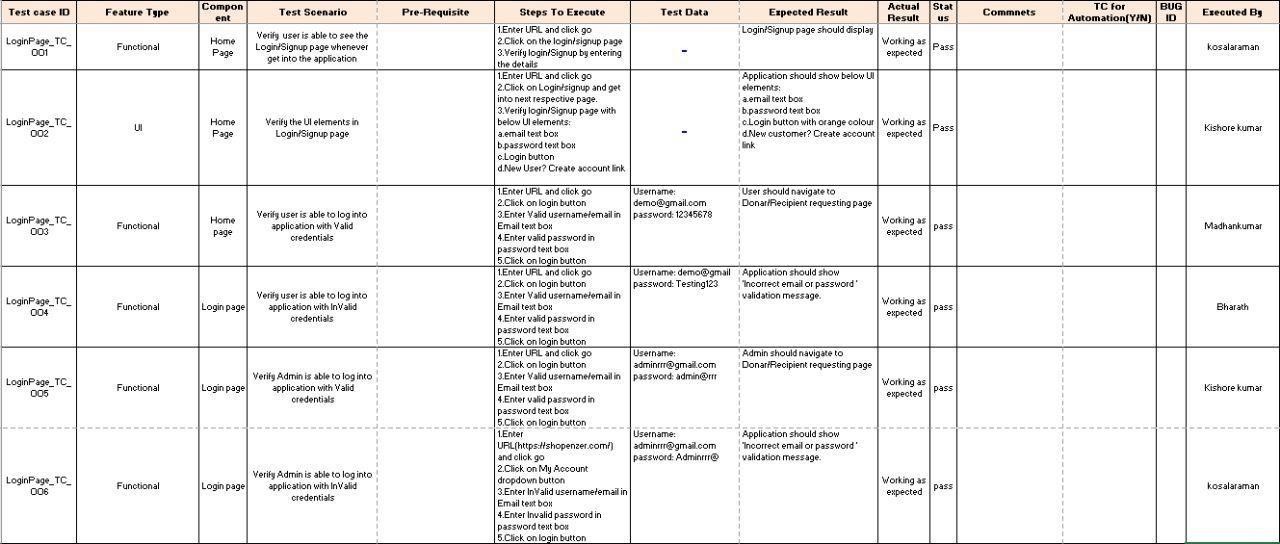




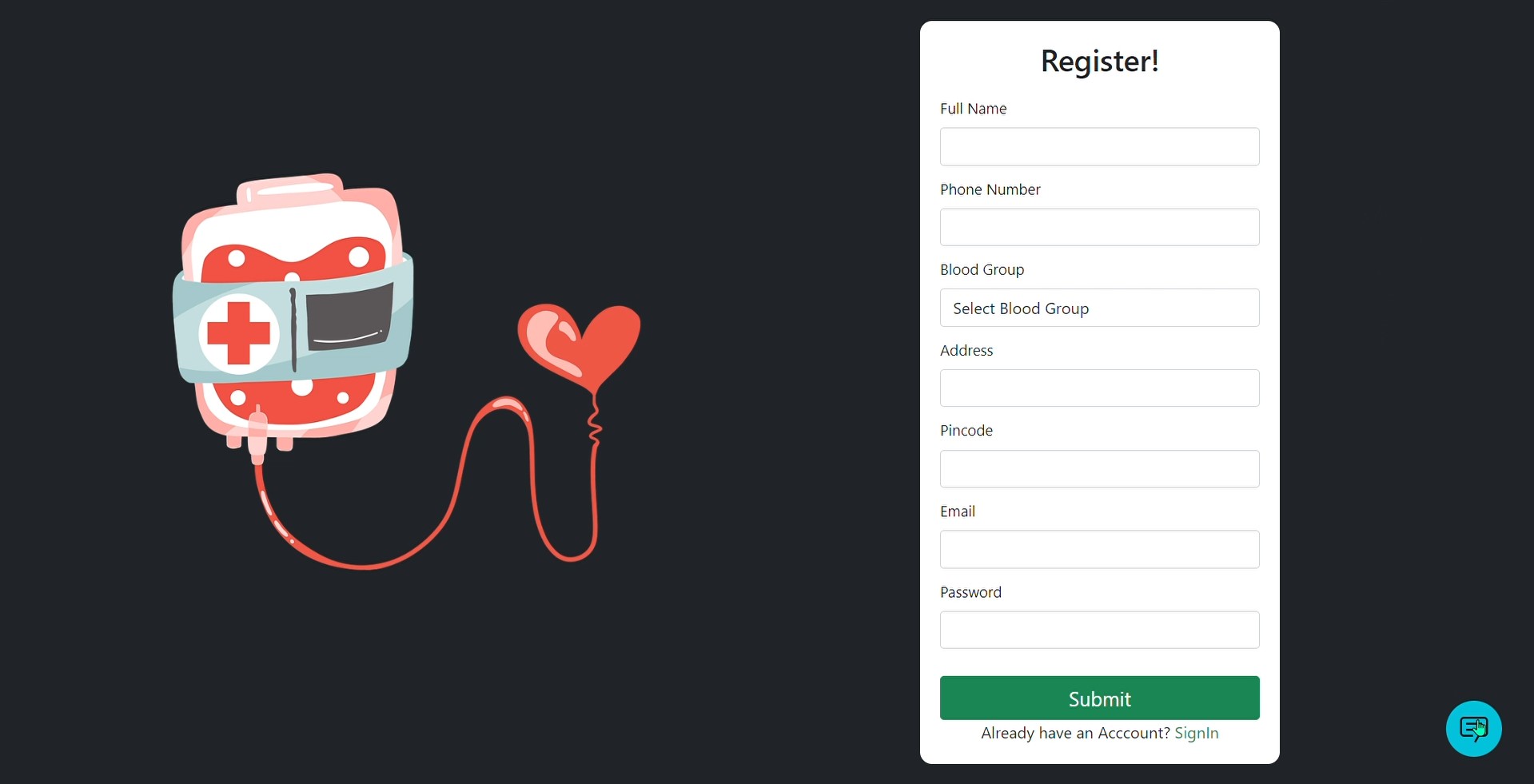


**8.TESTING:**

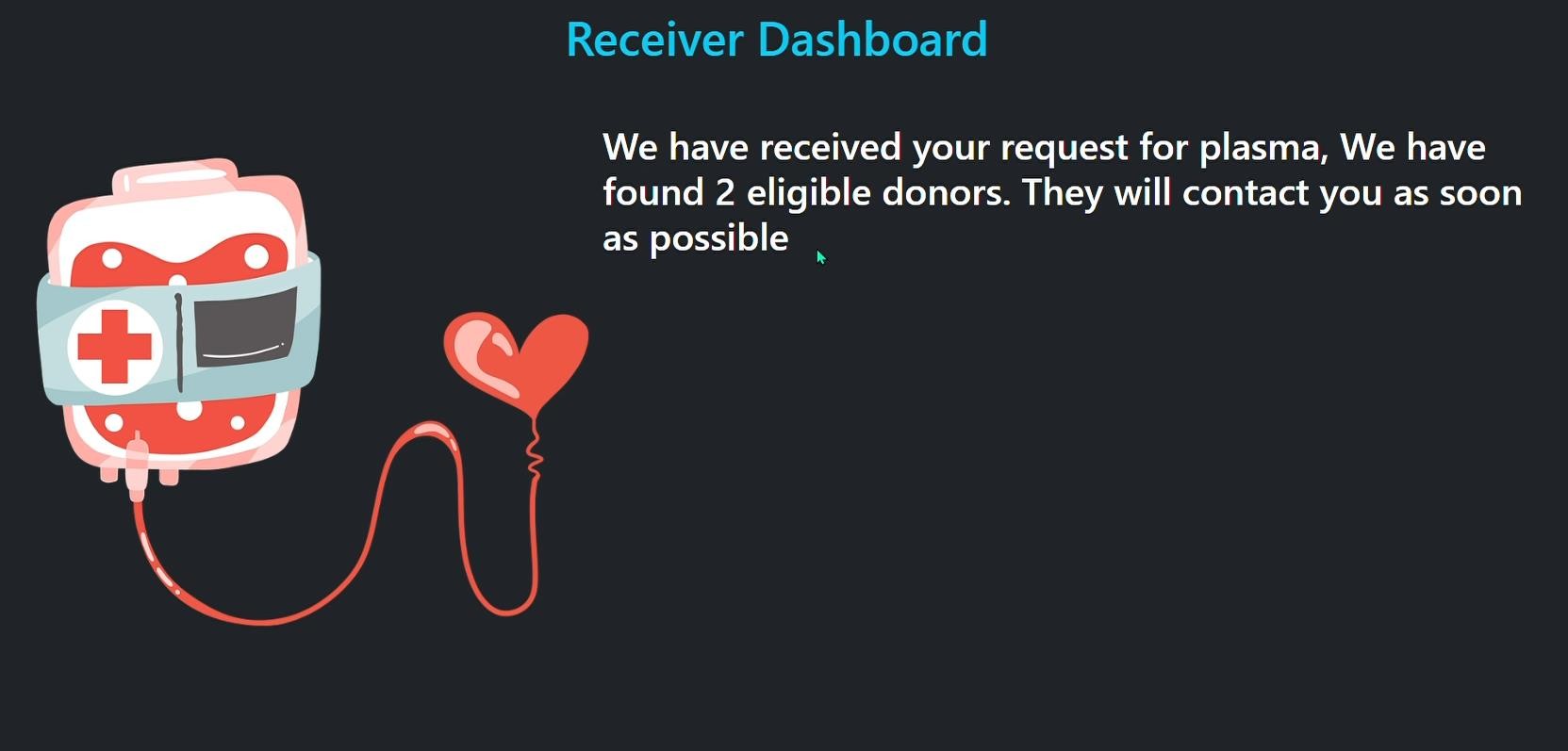
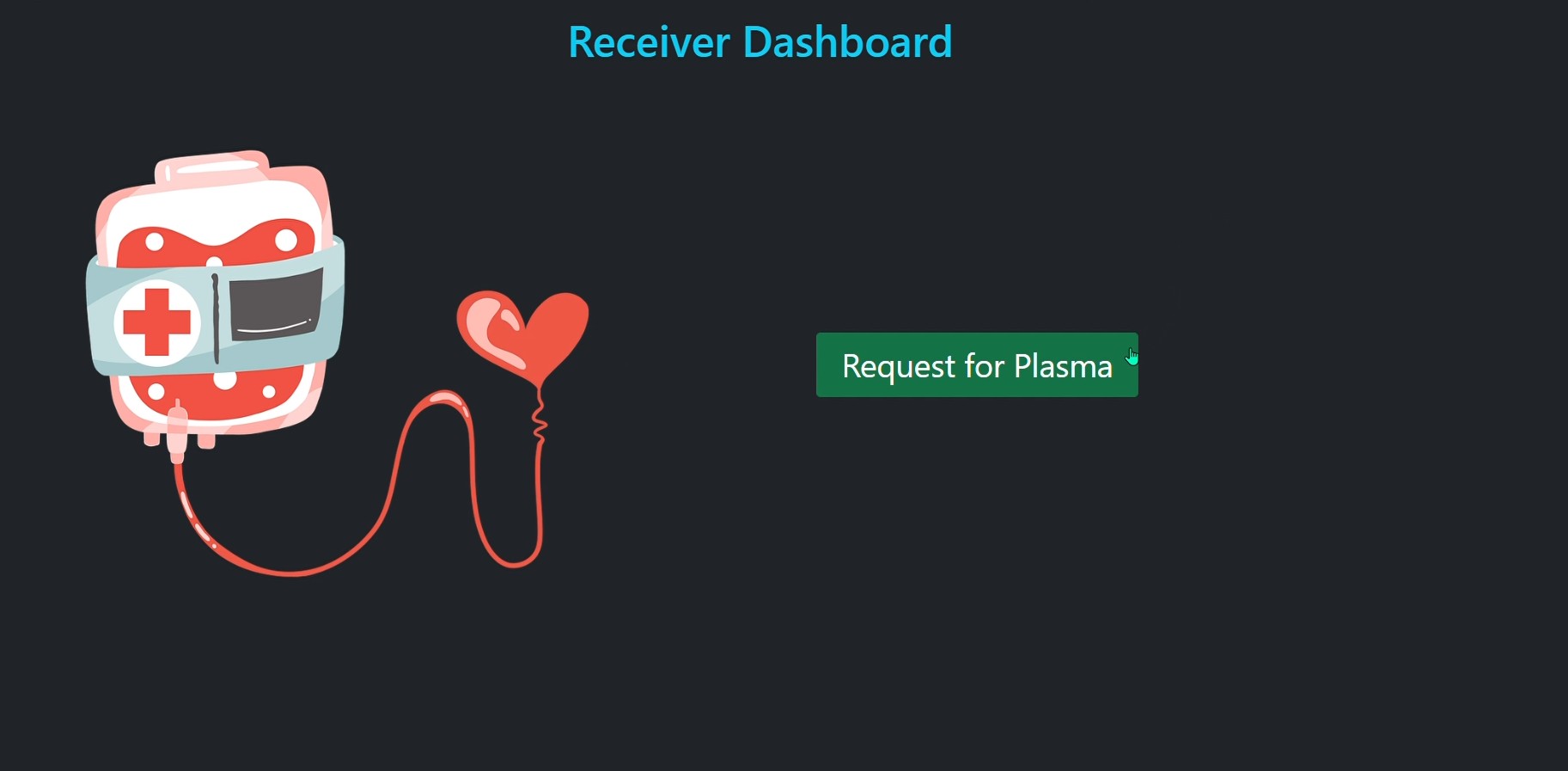
**8.1.Test Cases:**



# Screenshots: Login Page:



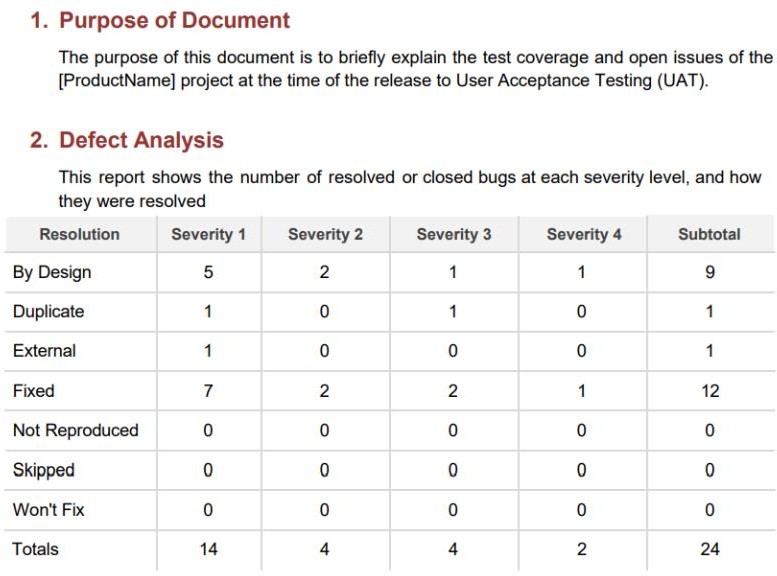
**Recipient Page:**



**8.2User Acceptance Testing:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resolution** | **Severity1** | **Severity2** | **Severity3** | **Severity4** | **Subtotal** |
| By Design | 10 | 4 | 2 | 4 | 20 |
| Duplicate | 1 | 0 | 1 | 0 | 2 |
| External | 2 | 2 | 1 | 1 | 6 |
| Fixed | 4 | 1 | 1 | 10 | 16 |
| Not Reproduced | 0 | 0 | 0 | 0 | 0 |
| Skipped | 1 | 1 | 0 | 1 | 3 |
| Won't Fix | 0 | 2 | 2 | 0 | 4 |
| Totals | 18 | 10 | 7 | 16 | 51 |

**9.RESULTS:**

**9.1Performance Metrics:**

**10.ADVANTAGES & DISADVANTAGES:**

# ADVANTAGES:

* **Speed**
  + This website is fast and offers great accuracy as compared to manual registered
* **Maintenance**
  + Less maintenance is required.
* **User Friendly**
  + It is very easy to use and understand. It is easily workable and accessible for

everyone.

* **Fast Results**
  + It would help you to provide plasma donors easily depending upon the availability of it.

# DISADVANTAGES:

**•Internet**

* + It would require an internet connection for the working of the website.
* **Auto-Verification**
  + It cannot automatically verify the genuine users.

# 11.CONCLUSION

Utilizing the plasma donor website, which is housed on the IBM Cloud platform, is the most effective approach for infected individuals to discover plasma doors.

to guarantee the efficient running of the website. In order to ensure that the operations are going smoothly, I have hosted the website in an IBM Db2 and Kubernates Cluster. In order to deploy the application, IBM Db2 service is employed together with cloud lambda function.

# 12.FUTURE SCOPE:

By combining Plasma Application with different social network application programme interfaces, this application may be created to further increase user accessibility (APIs). As a result, users can login and register using different social networks. This would improve the blood donation process and increase the number of donors.

In the future, user interface (UI) can be enhanced to serve a worldwide audience by supporting many national languages. Data extraction from several social networks is possible, and results are displayed in the Blood/Plasma Request Feeds. For the convenience of users, appointments can be synced with Google and Outlook calendars.

The purpose of the Donor and Beneficiary Stories feature is to foster a sense of community. Beneficiaries can share their experiences of getting blood transfusions that helped improve their health and quality of life, while donors can view and share personal experiences regarding their donations.

The Live Check-in Process feature aims to give users a better experience in terms of waiting times while they are making donations. We believe that making the process more effective will make the user look forward to his blood/plasma donation appointments.

# 13.APPENDIX

**13.1.SOURCE CODE:**

from flask import Flask, render\_template, request, redirect, url\_for, session  
from SendGridMail import sendMail  
import ibm\_db  
  
conn = ibm\_db.connect("DATABASE=bludb;HOSTNAME=98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30875;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=jkc20422;PWD=e5hC1j159iI2zSKD",'','')  
  
app = Flask(\_\_name\_\_)  
app.secret\_key = 'SESSIONKEY'  
  
  
@app.route("/",methods=['GET'])  
def home():  
 if 'email' not in session:  
 return redirect(url\_for('login'))  
 return render\_template('home.html')  
  
@app.route("/register",methods=['GET','POST'])  
def register():  
 if request.method == 'POST':  
 username = request.form['username']  
 email = request.form['email']  
 password = request.form['password']  
 phno = request.form['phno']  
 bloodGroup = request.form['blood\_group']  
 address = request.form['address'] or ''  
 pincode = request.form['pincode']  
  
  
 if not email or not username or not password or not phno or not bloodGroup or not pincode:  
 return render\_template('register.html',error='Please fill all fields')  
   
 query = "SELECT \* FROM USERS WHERE Email=?"  
 stmt = ibm\_db.prepare(conn, query)  
 ibm\_db.bind\_param(stmt,1,email)  
 ibm\_db.execute(stmt)  
 isUser = ibm\_db.fetch\_assoc(stmt)  
   
 if not isUser:  
 insert\_sql = "INSERT INTO Users(NAME,EMAIL,PASSWORD,PHONE,BLOOD\_GROUP,ADDRESS,PINCODE) 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, password)  
 ibm\_db.bind\_param(prep\_stmt, 4, phno)  
 ibm\_db.bind\_param(prep\_stmt, 5, bloodGroup)  
 ibm\_db.bind\_param(prep\_stmt, 6, address)  
 ibm\_db.bind\_param(prep\_stmt, 7, pincode)  
 ibm\_db.execute(prep\_stmt)  
   
 sendMail(email, 'Registered Successfully', 'Hello {}, <br>Thank you for Registering!! 🙂'.format(username))  
  
 return redirect(url\_for('login'))  
 else:  
 return render\_template('register.html')  
  
 return render\_template('register.html')  
  
@app.route("/login",methods=['GET','POST'])  
def login():  
 if request.method == 'POST':  
 email = request.form['email']  
 password = request.form['password']  
 loginType = request.form['login-type']  
  
 if not email or not password:  
 return render\_template('login.html',error='Please fill all fields')  
 query = "SELECT \* FROM USERS WHERE Email=?"  
 stmt = ibm\_db.prepare(conn, query)  
 ibm\_db.bind\_param(stmt,1,email)  
 ibm\_db.execute(stmt)  
 isUser = ibm\_db.fetch\_assoc(stmt)  
 print(isUser,password)  
  
 if not isUser:  
 return render\_template('login.html',error='Invalid Credentials')  
   
 #if not isPasswordMatch:  
 if(isUser['PASSWORD']!=password):  
 return render\_template('login.html',error='Invalid Credentials')  
  
 session['email'] = isUser['EMAIL']  
  
 if(loginType == 'donor'):  
 return redirect("donor")  
 else:  
 return redirect("recipient")  
  
 return render\_template('login.html')  
  
@app.route("/donor", methods=['GET','POST'])  
def addDonor():  
 if request.method == 'POST':  
 query = "SELECT \* FROM USERS WHERE Email=?"  
 stmt = ibm\_db.prepare(conn, query)  
 ibm\_db.bind\_param(stmt,1,session['email'])  
 ibm\_db.execute(stmt)  
 isUser = ibm\_db.fetch\_assoc(stmt)  
 print(isUser)  
  
 insert\_sql = "INSERT INTO DONOR(NAME,EMAIL,BLOOD\_GROUP,PINCODE) VALUES (?,?,?,?)"  
 prep\_stmt = ibm\_db.prepare(conn, insert\_sql)  
 ibm\_db.bind\_param(prep\_stmt, 1, isUser['NAME'])  
 ibm\_db.bind\_param(prep\_stmt, 2, isUser['EMAIL'])  
 ibm\_db.bind\_param(prep\_stmt, 3, isUser['BLOOD\_GROUP'])  
 ibm\_db.bind\_param(prep\_stmt, 4, isUser['PINCODE'])  
 ibm\_db.execute(prep\_stmt)  
  
 sendMail(session['email'], 'Request for Donation Received', 'Thank you🙏, your request has beed added!')  
  
 return render\_template('donor.html', name='Donor Dashboard', request\_for\_donation=True)  
   
 query = "SELECT \* FROM DONOR WHERE Email=?"  
 stmt = ibm\_db.prepare(conn, query)  
 ibm\_db.bind\_param(stmt,1,session['email'])  
 ibm\_db.execute(stmt)  
 isUser = ibm\_db.fetch\_assoc(stmt)  
 print(isUser)  
 return render\_template('donor.html', name='Donor Dashboard', request\_for\_donation=isUser)  
  
def getDonorsList(send\_email, blood\_group):  
 query = "SELECT \* FROM DONOR WHERE BLOOD\_GROUP=?"  
 stmt = ibm\_db.prepare(conn, query)  
 ibm\_db.bind\_param(stmt,1,blood\_group)  
 ibm\_db.execute(stmt)  
 donors = ibm\_db.fetch\_assoc(stmt)  
 donorList=[]  
 if donors:  
 while donors:  
 donorList.append(donors)  
 if send\_email:  
 sendMail(donors['EMAIL'], 'URGENT!!! Plasma Donation needed', '{} has requested for plasma of {} blood group. You can contact them at {} and {}'.format(isUser['NAME'],isUser['BLOOD\_GROUP'],isUser['PHONE'], isUser['EMAIL']))  
 donors = ibm\_db.fetch\_assoc(stmt)  
 return donorList  
  
  
@app.route("/recipient", methods=['GET','POST'])  
def addRequest():  
 if request.method == 'POST':  
 query = "SELECT \* FROM USERS WHERE Email=?"  
 stmt = ibm\_db.prepare(conn, query)  
 ibm\_db.bind\_param(stmt,1,session['email'])  
 ibm\_db.execute(stmt)  
 isUser = ibm\_db.fetch\_assoc(stmt)  
  
 insert\_sql = "INSERT INTO RECIPIENT(NAME,EMAIL,BLOOD\_GROUP,PINCODE) VALUES (?,?,?,?)"  
 prep\_stmt = ibm\_db.prepare(conn, insert\_sql)  
 ibm\_db.bind\_param(prep\_stmt, 1, isUser['NAME'])  
 ibm\_db.bind\_param(prep\_stmt, 2, isUser['EMAIL'])  
 ibm\_db.bind\_param(prep\_stmt, 3, isUser['BLOOD\_GROUP'])  
 ibm\_db.bind\_param(prep\_stmt, 4, isUser['PINCODE'])  
 ibm\_db.execute(prep\_stmt)  
  
 # sendMail(session['email'], 'Request for Plasma Received', 'Your Request has been added Successfully')  
 donorList = getDonorsList(True, isUser['BLOOD\_GROUP'])  
   
 return render\_template('recipient.html', name='Receiver Dashboard', request\_for\_plasma=True, count=len(donorList))  
   
 query = "SELECT \* FROM RECIPIENT WHERE Email=?"  
 stmt = ibm\_db.prepare(conn, query)  
 ibm\_db.bind\_param(stmt,1,session['email'])  
 ibm\_db.execute(stmt)  
 isUser = ibm\_db.fetch\_assoc(stmt)  
 donorList=[]  
 if isUser:  
 donorList = getDonorsList(False, isUser['BLOOD\_GROUP'])  
 return render\_template('recipient.html', name='Receiver Dashboard', request\_for\_plasma=isUser, count=len(donorList))  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app.run()

**13.2. GITHUB LINK & PROJECT DEMO LINK:**

**GITHUB LINK:** [**https://github.com/IBM-EPBL/IBM-Project-43204-1660714144**](https://github.com/IBM-EPBL/IBM-Project-43204-1660714144)

**PROJECT DEMO LINK:** [**https://drive.google.com/file/d/1T69X9QcbO1neWHvXYrqf2kIsddnRvA32/view?usp=share\_link**](https://drive.google.com/file/d/1T69X9QcbO1neWHvXYrqf2kIsddnRvA32/view?usp=share_link%20)