PROJECT REPORT

PLASMA DONOR APPLICATION

Team ID: PNT2022TMID27815

Batch: B7-1A3E

TEAM LEADER:

Name: JAIMUGIL C

Register Number: 311519104023

TEAM MEMBERS:

Name: DANIEL A

Register Number: 311519104011

Name: SANTHOSH S

Register Number: 311519104051

Name: NARENDRANATH R S

Register Number: 311519104038

CONTENTS

1. INTRODUCTION

1.1 Project Overview

This project, titled "Cloud based Plasma Donor Application", aims at providing a platform for the person who needs plasma in terms of need. This is achieved with the help of donors who had registered in this platform for voluntary donation. At the time of registration the donors are requested their personal information. The requester can request the plasma from the list of available donors. A no response mail will be send to the respective donor.

1.2 Purpose

A platform for users to register if they want to volunteer to donate their plasma and also help the users in need of plasma by providing them with the availability of plasma in nearby location.

2. LITERATURE SURVEY

2.1 Existing problem

It is somewhat difficult to search for donors of plasma in terms of emergency. Patients or hospitals need to visit each blood bank. when need of plasma which is not feasible in case of emergency

2.2 References

Paper 1
Authors: Rehab S.Ali, Tamer F.Hafez , Ali Badawey Ali, Nadia , Abd-Alsabour
Year : 2020
Title: A Web Application to Manage All Blood Donation and Transfusion Process
Methodology: This paper aims to help people fulfill needs for a safe and reliable blood group by searching for and locating a specific blood group.
Advantage: Blood Bank system helps blood banks and donors save lives of the patients through a controlled system which manages all blood donation and transfusion processes.
Disadvantage: It is difficult to connect with people with who has the on demand blood group.
Paper 2
Authors: Mohammed Anis Oukebdane , Samir Ghouali , Karima Ghazali , Mohammed Feham
Year : 2020
Title: E-Blood Bank Application for Donors and Life Savers
Methodology: Web application that is connected to a centralized database, to collect and organize data from all blood banks and blood donation campaigns.
Advantage: This application contains the updated database which has all category blood groups of people all over the world.
Disadvantage: Database should be updated periodically.

Paper 3

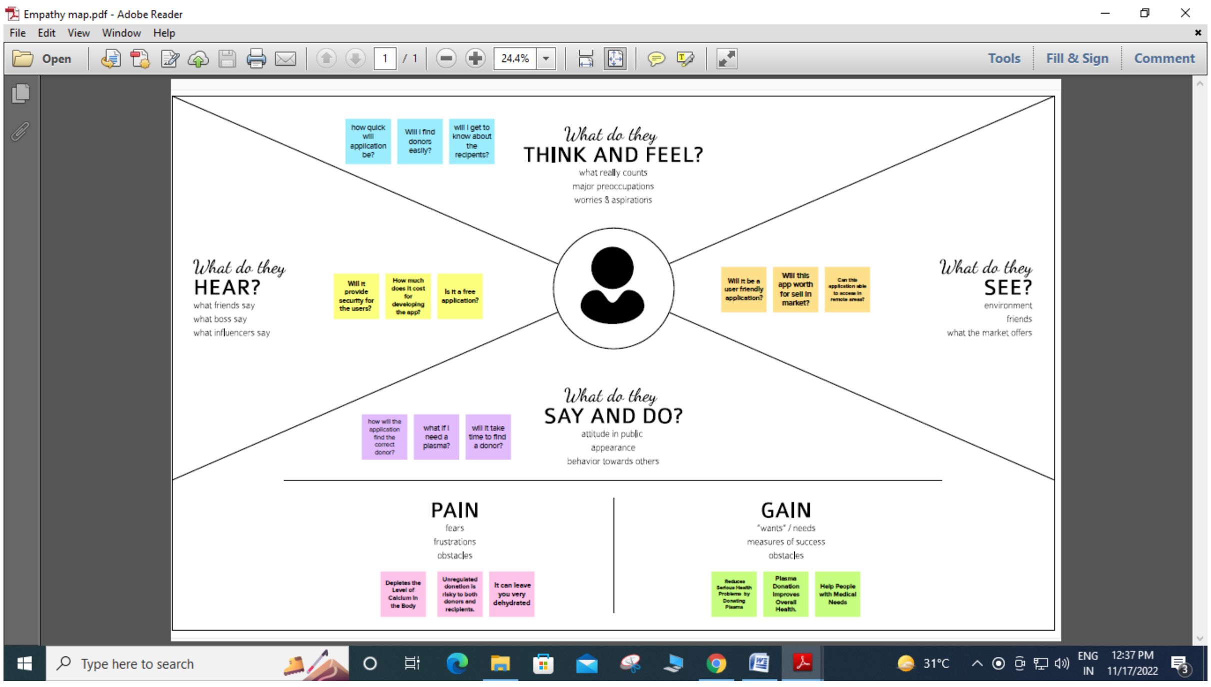
Authors: Ms. Pradnya Jagtap ,Ms. Monika Mandale ,Ms. Prachi Mhaske ,Ms. Sonali Vidhate ,Mr. S. S. Patil
Year : 2020
Title: Implementation of blood bank donation application
Methodology: Blood Donation System is an android based system that is designed to store, process, retrieve and analyse information concerned with the administrative and inventory
Advantage: Easy connecting donors and recipients makes blood donation way more proficient.
Paper 4
Authors: Abhijet Gaikwad, Nilofar Mulla,Tejashri Wagaj,Raviraj Ingale,Prof.Brijendra Gupta,Prof.Kamal Reddy
Year : 2020
Title: Smart Blood Finder
Methodology: The general idea of the study is to develop a Smart Blood Finder Application is to manage the records of the donors and the people who need blood
Advantage:. Develop a computer system that will link all donor
Disadvantage: It does not allow integration with blood donor management system

2.3 Problem Statement Definition

Plasma donor application deployed in cloud platform in which a user interacts with the application and posts a request then the concerned blood group donors will get notified about it. A user can also register them as a donor which will be stored in a database

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas



What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

- how quick will application be?
- Will I find donors easily?
- will I get to know about the recipents?

What do they HEAR?
what friends say
what boss say
what influencers say

- Will it provide securtty for the users?
- How much does it cost for developing the app?
- Is it a free application?

What do they SEE?
environment
friends
what the market offers

- Will it be a user friendly application?
- Will this app worth for sell in market?
- Can this application able to access in remote areas?

What do they SAY AND DO?
attitude in public
appearance
behavior towards others

- how will the application find the correct donor?
- what if I need a plasma?
- will it take time to find a donor?

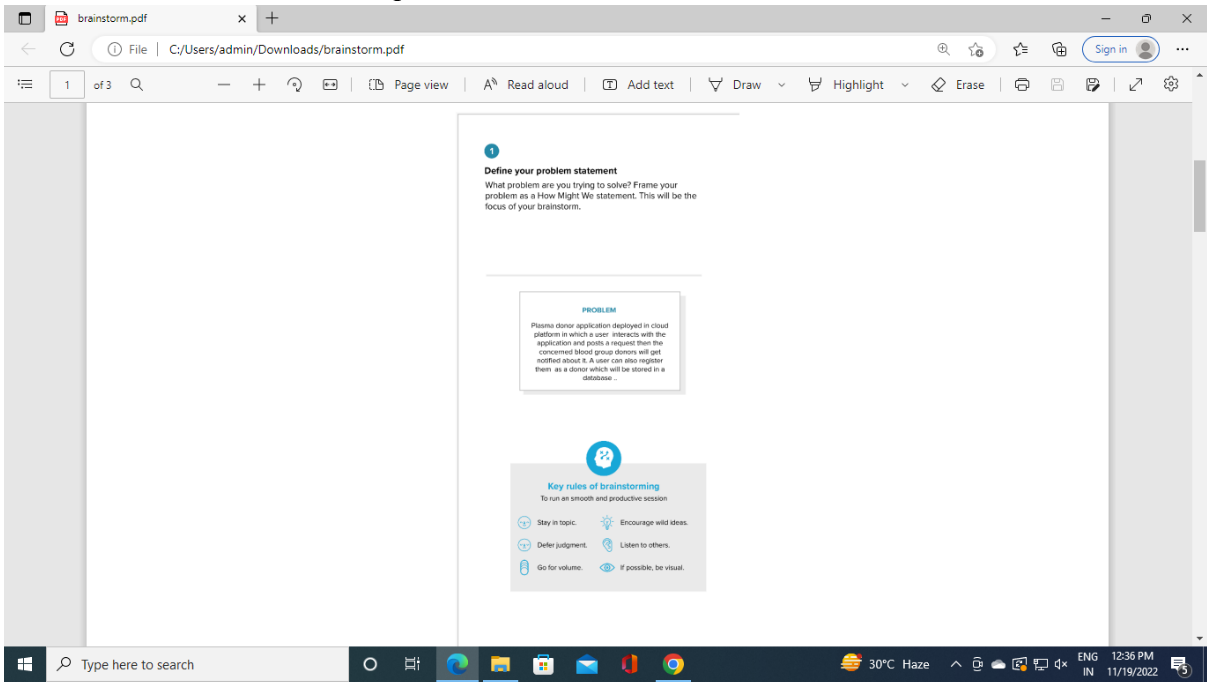
PAIN
fears
frustrations
obstacles

- Depletes the Level of Calcium in the Body
- Unregulated donation is risky to both donors and recipients.
- It can leave you very dehydrated

GAIN
"wants" / needs
measures of success
obstacles

- Reduces Serious Health Problems by Donating Plasma
- Plasma Donation Improves Overall Health.
- Help People with Medical Needs

3.2 Ideation & Brainstorming



1

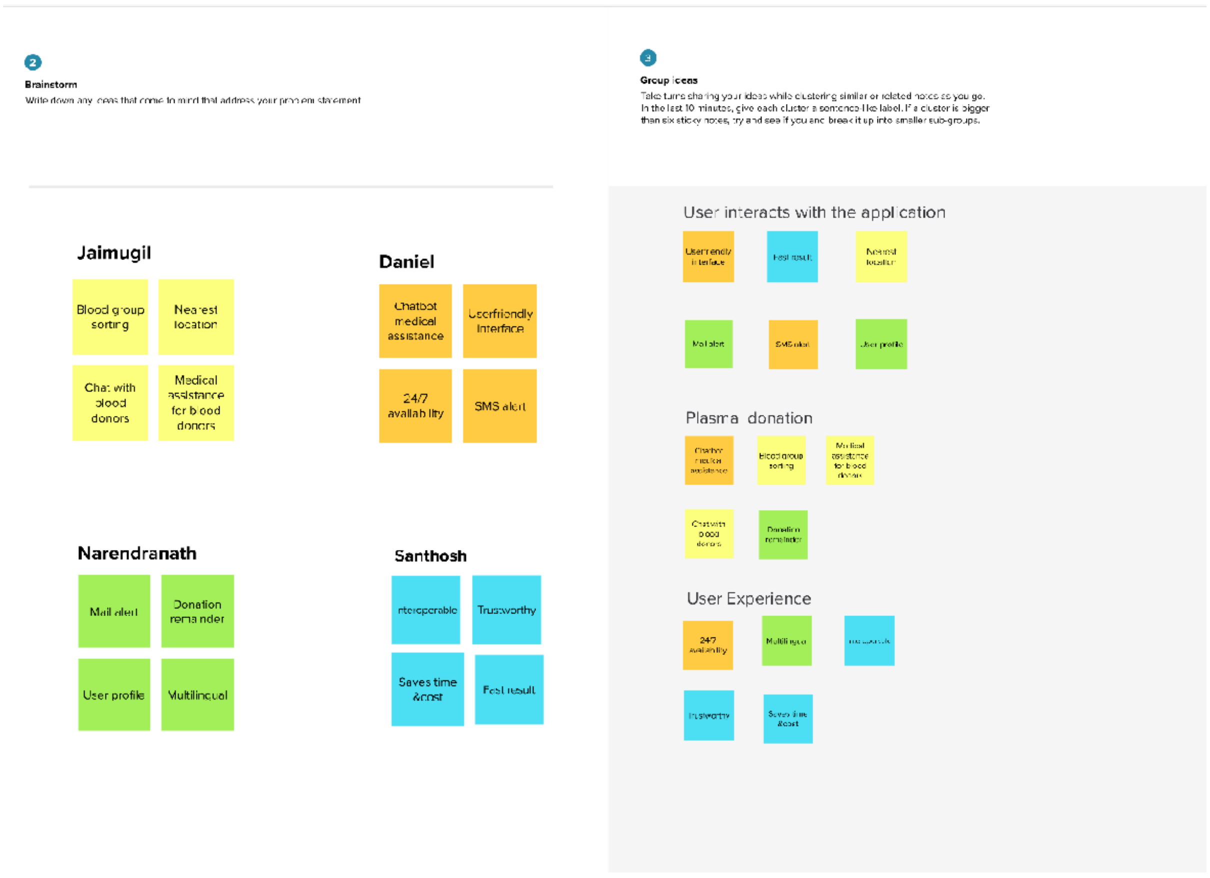
Define your problem statement
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

PROBLEM

Plasma donor application deployed in cloud platform in which a user interacts with the application and posts a request then the concerned blood group donors will get notified about it. A user can also register them as a donor which will be stored in a database ..

Key rules of brainstorming
To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.



2

Brainstorm

Write down any ideas that come to mind that address your problem statement

3

Group ideas

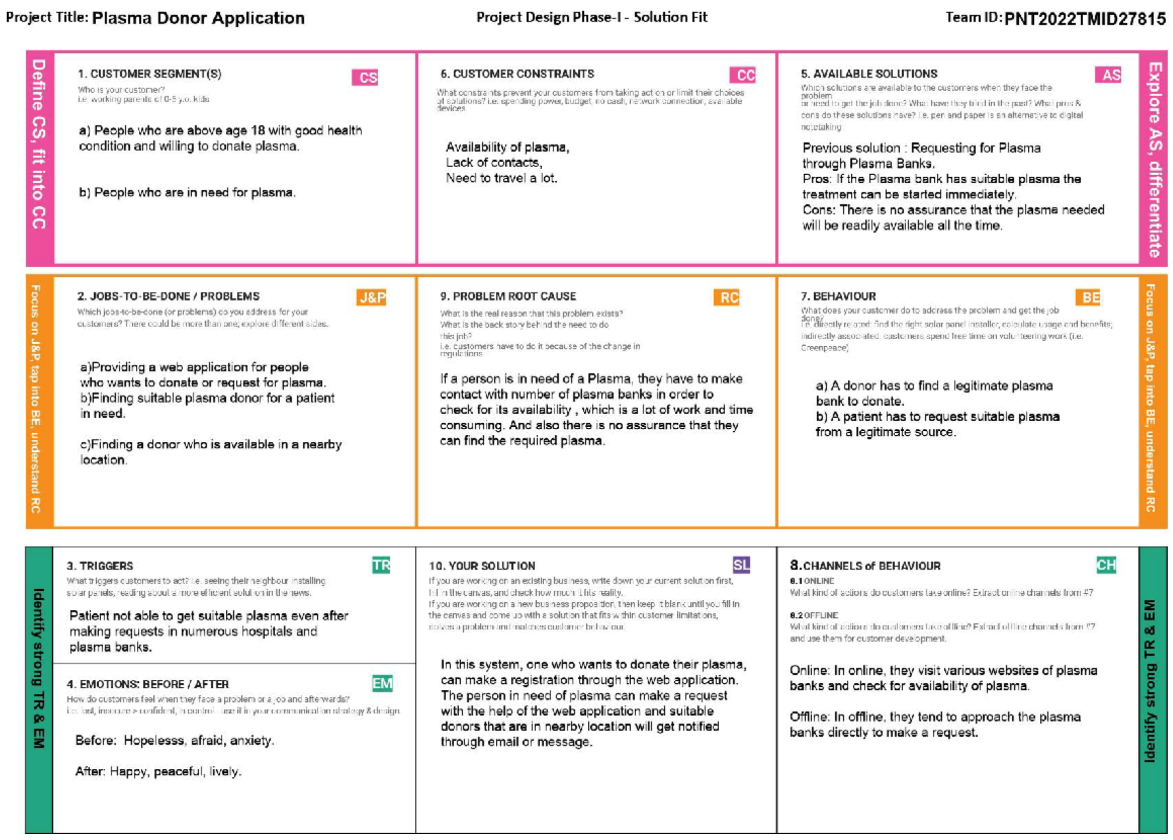
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Plasma donor application deployed in cloud platform in which a user interacts with the application and posts a request then the concerned blood group donors will get notified about it. A user can also register them as a donor which will be stored in a database.
2.	Idea / Solution description	Easy to find the nearby donor who are willing to donate. Receiver can post a request for the plasma.
3.	Novelty / Uniqueness	Accessible to anyone at any time. Simple user interface.
4.	Social Impact / Customer Satisfaction	Free of cost. Fast access. Fast identification of donors.
5.	Business Model (Revenue Model)	Web advertisement.

6.	Scalability of the Solution	A platform for users to register if they want to volunteer to donate their plasma and also help the users in need of plasma by providing them with the availability of plasma in nearby location.

3.4 Problem Solution fit



Project Title: **Plasma Donor Application**

Project Design Phase-I - Solution Fit

Team ID: **PNT2022TMID27815**

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

a) People who are above age 18 with good health condition and willing to donate plasma.

b) People who are in need for plasma.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices

Availability of plasma,
Lack of contacts,
Need to travel a lot.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Previous solution : Requesting for Plasma through Plasma Banks.
Pros: If the Plasma bank has suitable plasma the treatment can be started immediately.
Cons: There is no assurance that the plasma needed will be readily available all the time.

Explore AS, differentiate

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

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

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

a)Providing a web application for people who wants to donate or request for plasma.
b)Finding suitable plasma donor for a patient in need.

c)Finding a donor who is available in a nearby location.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations

If a person is in need of a Plasma, they have to make contact with number of plasma banks in order to check for its availability , which is a lot of work and time consuming. And also there is no assurance that they can find the required plasma.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

a) A donor has to find a legitimate plasma bank to donate.
b) A patient has to request suitable plasma from a legitimate source.

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

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Patient not able to get suitable plasma even after making requests in numerous hospitals and plasma banks.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Before: Hopelesss, afraid, anxiety.

After: Happy, peaceful, lively.

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

In this system, one who wants to donate their plasma, can make a registration through the web application. The person in need of plasma can make a request with the help of the web application and suitable donors that are in nearby location will get notified through email or message.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

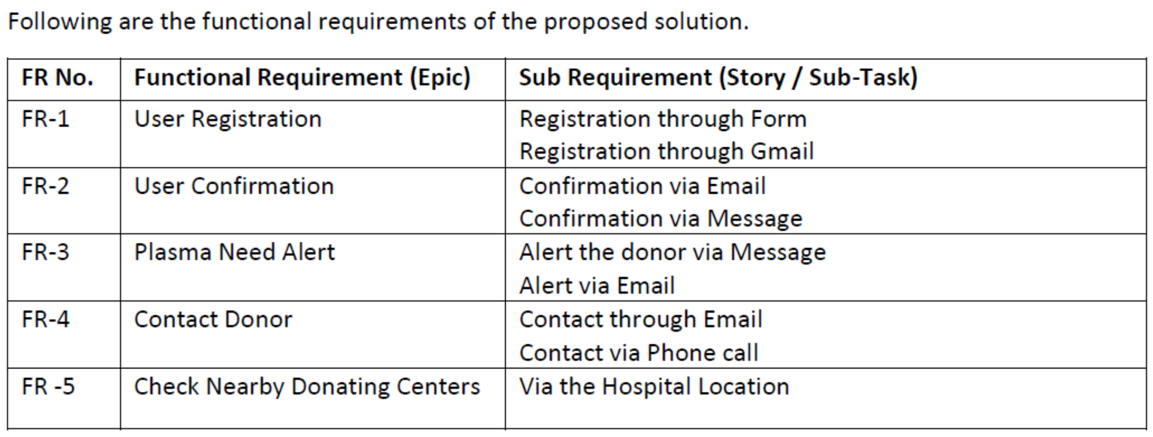
Online: In online, they visit various websites of plasma banks and check for availability of plasma.

Offline: In offline, they tend to approach the plasma banks directly to make a request.

Identify strong TR & EM

4. REQUIREMENT ANALYSIS

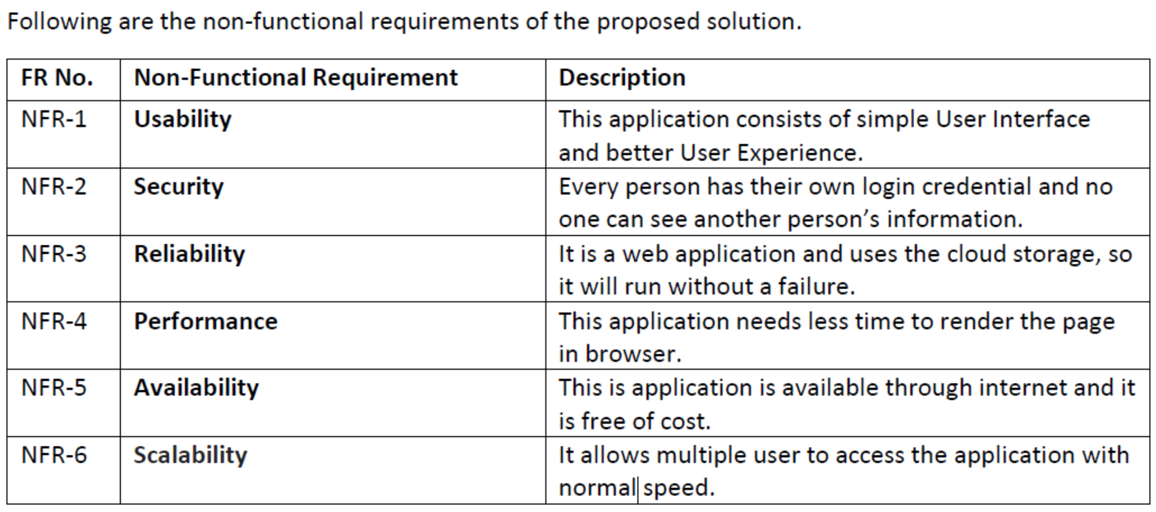
4.1 Functional requirements



Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via Message
FR-3	Plasma Need Alert	Alert the donor via Message Alert via Email
FR-4	Contact Donor	Contact through Email Contact via Phone call
FR -5	Check Nearby Donating Centers	Via the Hospital Location

4.2 Non-Functional requirements

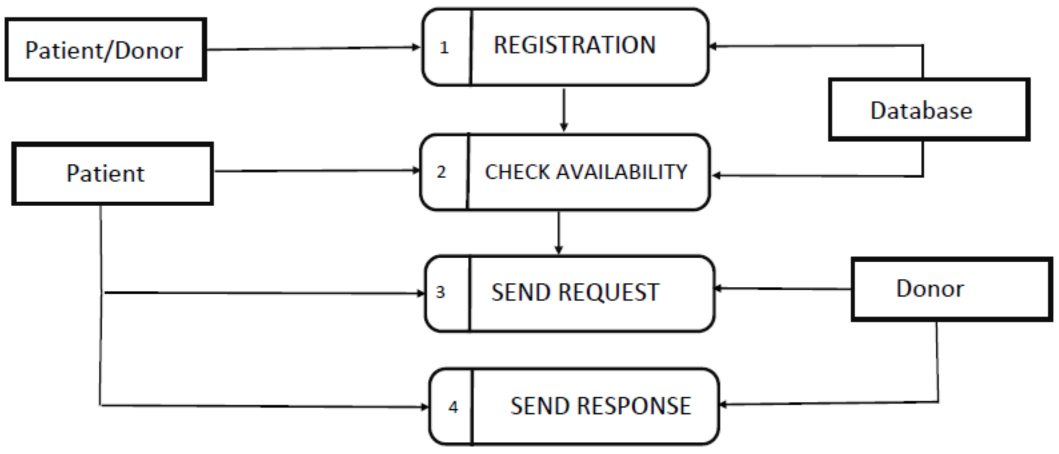


Following are the non-functional requirements of the proposed solution.

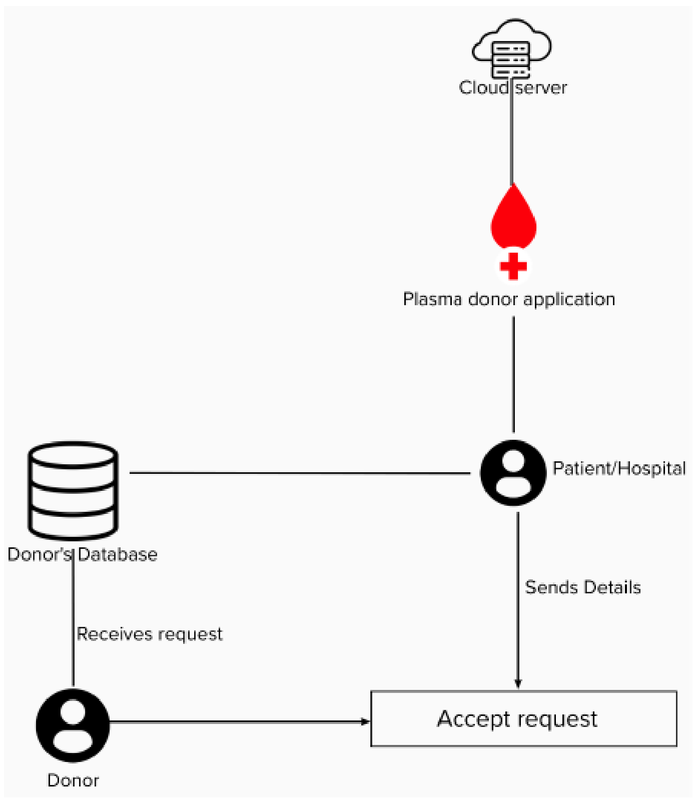
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This application consists of simple User Interface and better User Experience.
NFR-2	**Security**	Every person has their own login credential and no one can see another person's information.
NFR-3	**Reliability**	It is a web application and uses the cloud storage, so it will run without a failure.
NFR-4	**Performance**	This application needs less time to render the page in browser.
NFR-5	**Availability**	This is application is available through internet and it is free of cost.
NFR-6	**Scalability**	It allows multiple user to access the application with normal speed.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

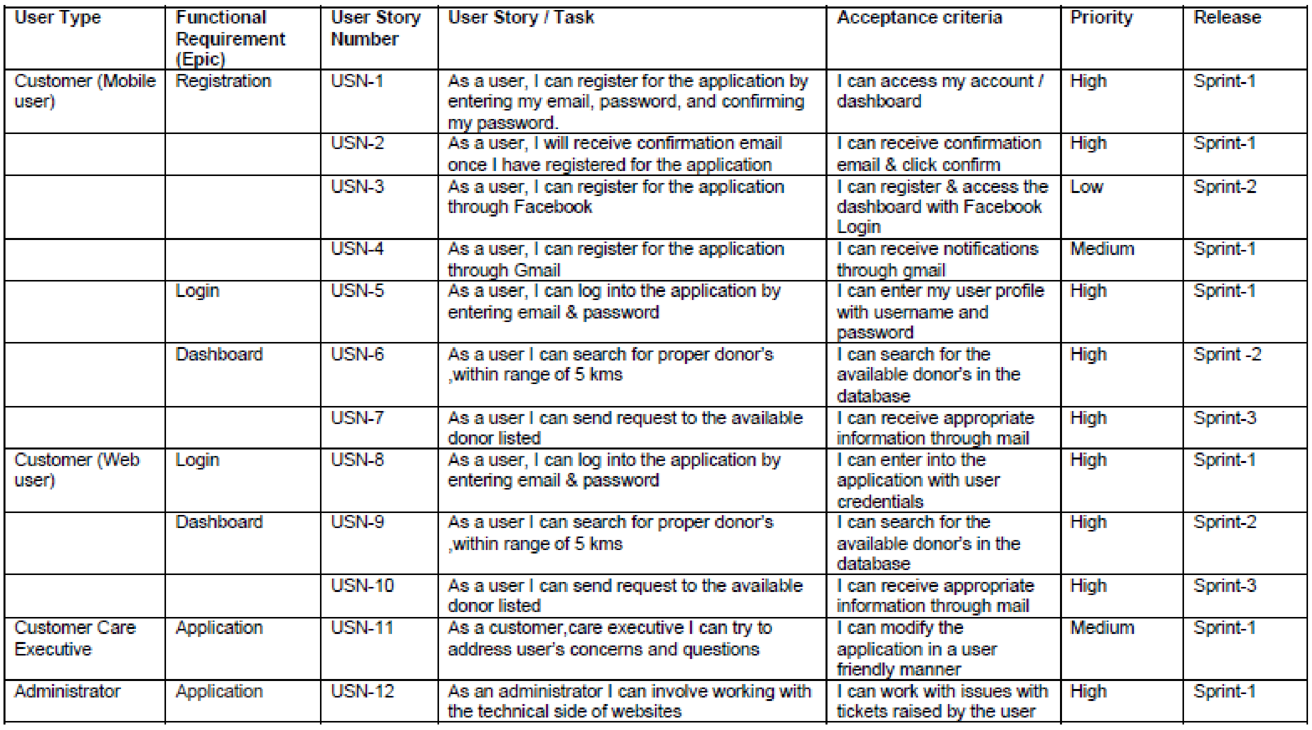


5.2 Solution & Technical Architecture



5.3 User Stories

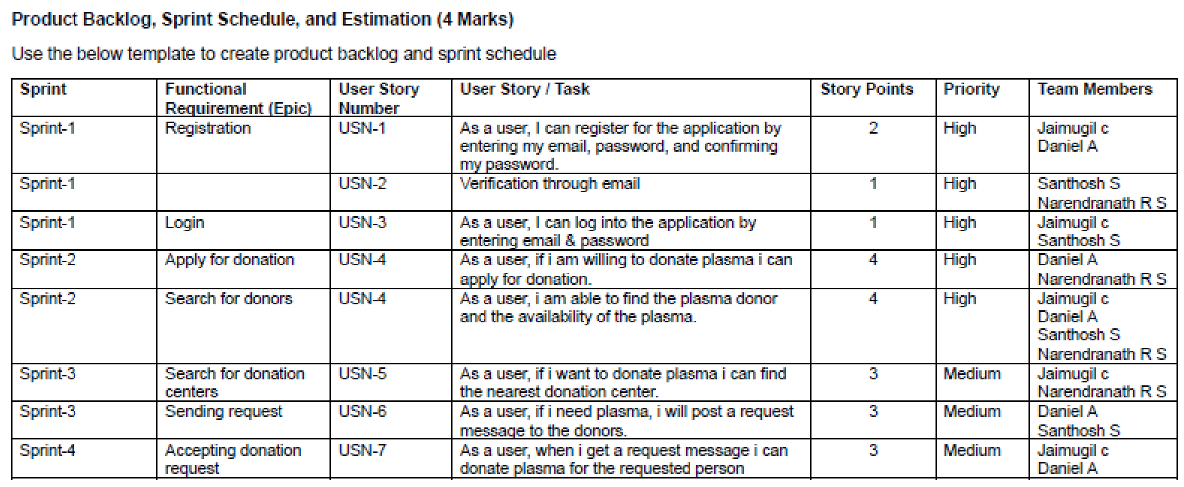
Use the below template to list all the user stories for the product.



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		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 Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can receive notifications through gmail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can enter my user profile with username and password	High	Sprint-1
	Dashboard	USN-6	As a user I can search for proper donor's ,within range of 5 kms	I can search for the available donor's in the database	High	Sprint -2
		USN-7	As a user I can send request to the available donor listed	I can receive appropriate information through mail	High	Sprint-3
Customer (Web user)	Login	USN-8	As a user, I can log into the application by entering email & password	I can enter into the application with user credentials	High	Sprint-1
	Dashboard	USN-9	As a user I can search for proper donor's ,within range of 5 kms	I can search for the available donor's in the database	High	Sprint-2
		USN-10	As a user I can send request to the available donor listed	I can receive appropriate information through mail	High	Sprint-3
Customer Care Executive	Application	USN-11	As a customer,care executive I can try to address user's concerns and questions	I can modify the application in a user friendly manner	Medium	Sprint-1
Administrator	Application	USN-12	As an administrator I can involve working with the technical side of websites	I can work with issues with tickets raised by the user	High	Sprint-1

6. PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning & Estimation

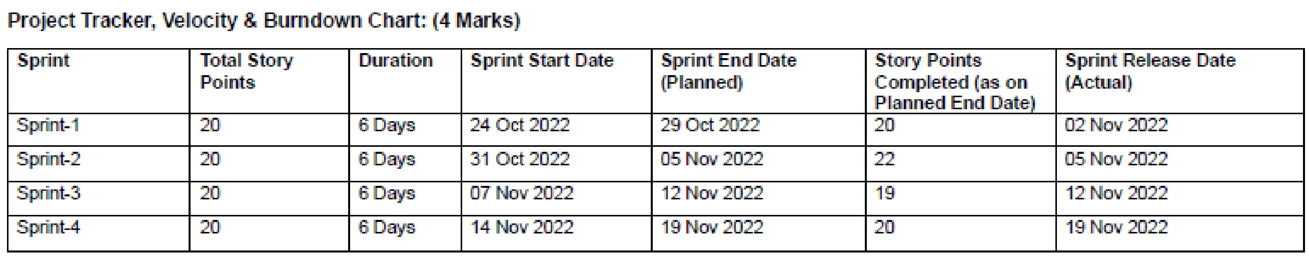


Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Jaimugil c Daniel A
Sprint-1		USN-2	Verification through email	1	High	Santhosh S Narendranath R S
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password	1	High	Jaimugil c Santhosh S
Sprint-2	Apply for donation	USN-4	As a user, if i am willing to donate plasma i can apply for donation.	4	High	Daniel A Narendranath R S
Sprint-2	Search for donors	USN-4	As a user, i am able to find the plasma donor and the availability of the plasma.	4	High	Jaimugil c Daniel A Santhosh S Narendranath R S
Sprint-3	Search for donation centers	USN-5	As a user, if i want to donate plasma i can find the nearest donation center.	3	Medium	Jaimugil c Narendranath R S
Sprint-3	Sending request	USN-6	As a user, if i need plasma, i will post a request message to the donors.	3	Medium	Daniel A Santhosh S
Sprint-4	Accepting donation request	USN-7	As a user, when i get a request message i can donate plasma for the requested person	3	Medium	Jaimugil c Daniel A

6.2 Sprint Delivery Schedule

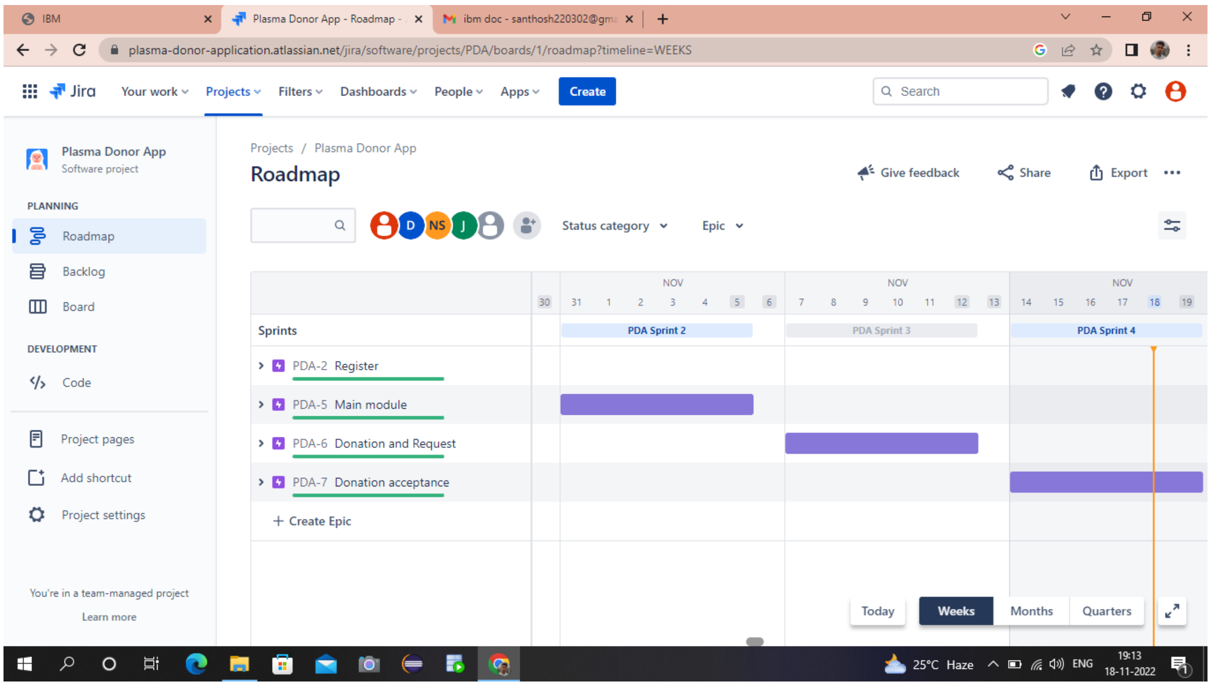


Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	02 Nov 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	22	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	19	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 Reports from JIRA

Roadmap



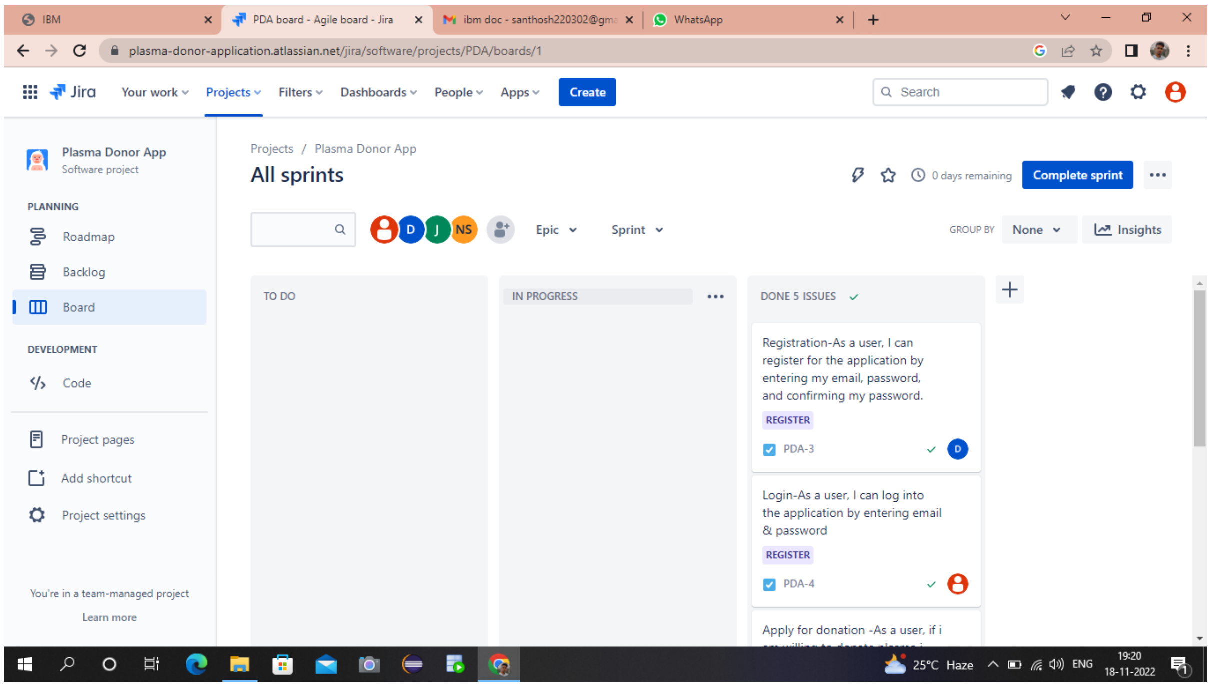
Roadmap

Sprints

- PDA-2 Register
- PDA-5 Main module
- PDA-6 Donation and Request
- PDA-7 Donation acceptance

+ Create Epic

Scrum board


plasma-donor-application.atlassian.net/jira/software/projects/PDA/boards/1

Plasma Donor App
Software project

PLANNING
Roadmap
Backlog
Board

DEVELOPMENT
Code

Project pages
Add shortcut
Project settings

You're in a team-managed project
Learn more

Projects / Plasma Donor App

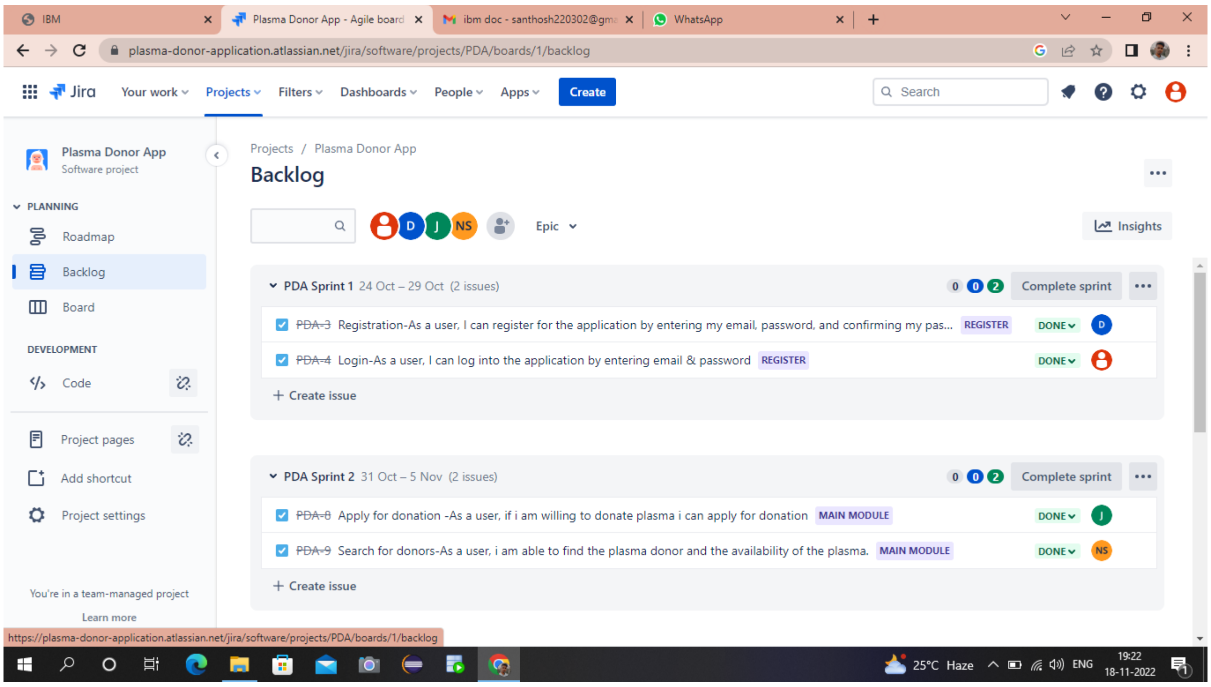
All sprints

0 days remaining · Complete sprint

Epic · Sprint · GROUP BY None · Insights

TO DO	IN PROGRESS	DONE 5 ISSUES ✓
		Registration-As a user, I can register for the application by entering my email, password, and confirming my password. REGISTER PDA-3 ✓ D
		Login-As a user, I can log into the application by entering email & password REGISTER PDA-4 ✓
		Apply for donation -As a user, if i

Backlogs


plasma-donor-application.atlassian.net/jira/software/projects/PDA/boards/1/backlog

Plasma Donor App
Software project

PLANNING
Roadmap
Backlog
Board

DEVELOPMENT
Code

Project pages
Add shortcut
Project settings

You're in a team-managed project
Learn more

Projects / Plasma Donor App

Backlog

Epic · Insights

PDA Sprint 1 24 Oct – 29 Oct (2 issues) 0 0 2 Complete sprint

✓ ~~PDA-3~~	Registration-As a user, I can register for the application by entering my email, password, and confirming my pas...	REGISTER	DONE	D
✓ ~~PDA-4~~	Login-As a user, I can log into the application by entering email & password	REGISTER	DONE	

+ Create issue

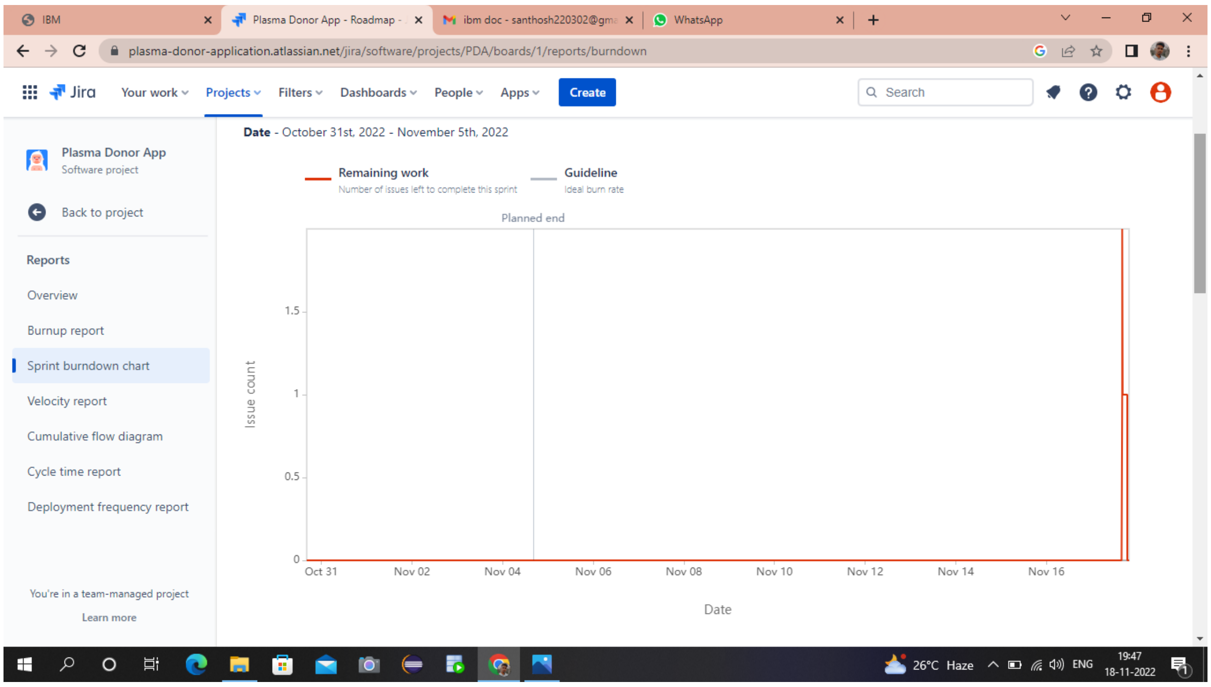
PDA Sprint 2 31 Oct – 5 Nov (2 issues) 0 0 2 Complete sprint

✓ ~~PDA-8~~	Apply for donation -As a user, if i am willing to donate plasma i can apply for donation	MAIN MODULE	DONE	J
✓ ~~PDA-9~~	Search for donors-As a user, i am able to find the plasma donor and the availability of the plasma.	MAIN MODULE	DONE	NS

+ Create issue

https://plasma-donor-application.atlassian.net/jira/software/projects/PDA/boards/1/backlog

Burndown chart



7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

Search for blood group

Search.HTML

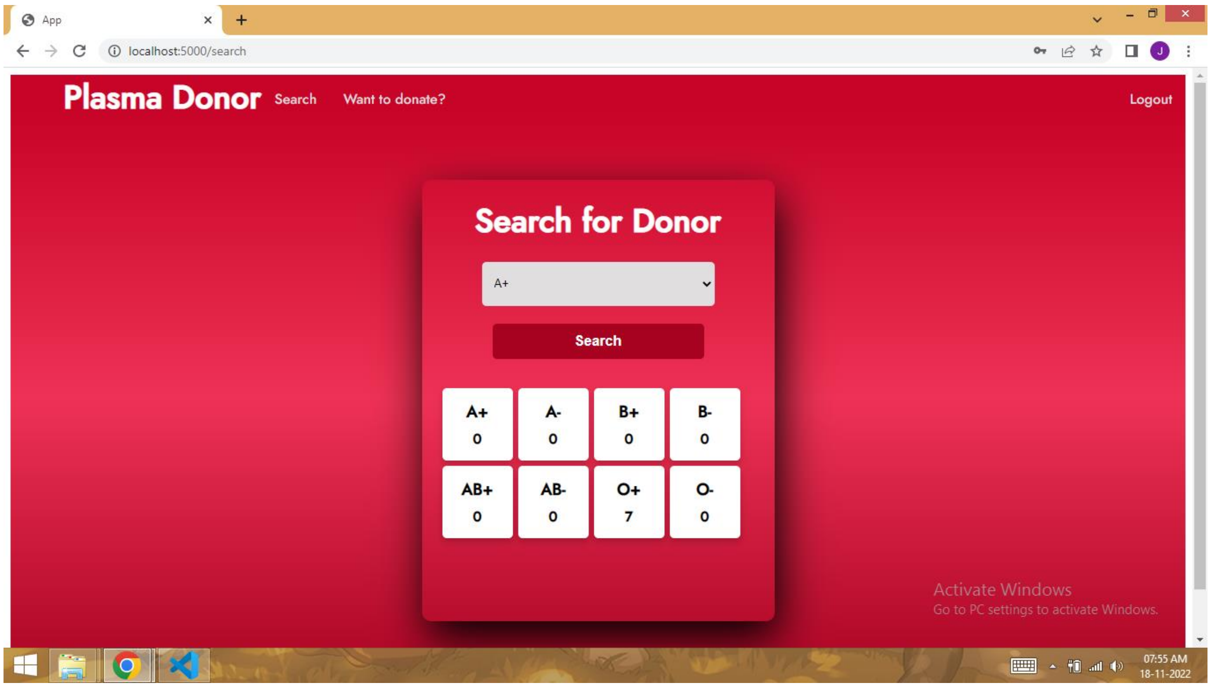
```
<!DOCTYPE html>
<html>

<head>
   <title>App</title>
   <link href="https://fonts.googleapis.com/css2?family=Jost:wght@500&display=swap"
rel="stylesheet">
   <link rel="stylesheet" type="text/css" href="{{url_for('static', filename='search.css')}}">
</head>

<body>
   <div class="navbar">
      <div class="left-nav">
         <div class="titlediv">
            Plasma Donor
         </div>
         <a class="navlink" href="{{url_for('search')}}">Search</a>
         <a class="navlink" href="{{url_for('donation')}}">Want to donate?</a>
      </div>
      <div class="right-nav">
         <a class="navlink" href="{{url_for('logout')}}">Logout</a>
      </div>
   </div>
   <div class="contentbody">
      <div class="main">
         <div class="signup">
            <form action="/requestmail" method="POST">
               <label>Search for Donor</label>
               <select class="dropdown" name="blood" required="">
                  <option value="A+">A+</option>
                  <option value="A-">A-</option>
                  <option value="B+">B+</option>
                  <option value="B-">B-</option>
                  <option value="AB+">AB+</option>
                  <option value="AB-">AB-</option>
                  <option value="O+">O+</option>
                  <option value="O-">O-</option>
               </select>
```

```
    <button>Search</button>
</form>
<div class="cardcontainer">
    <div class="card">
        <div class="cardinner">
            <h3>
                <b>A+</b>
            </h3>
            <h4>{{ap}}</h4>
        </div>
    </div>
    <div class="card">
        <div class="cardinner">
            <h3>
                <b>A-</b>
            </h3>
            <h4>{{an}}</h4>
        </div>
    </div>
    <div class="card">
        <div class="cardinner">
            <h3>
                <b>B+</b>
            </h3>
            <h4>{{bp}}</h4>
        </div>
    </div>
    <div class="card">
        <div class="cardinner">
            <h3>
                <b>B-</b>
            </h3>
            <h4>{{bn}}</h4>
        </div>
    </div>
    <div class="card">
        <div class="cardinner">
            <h3>
                <b>AB+</b>
            </h3>
            <h4>{{abp}}</h4>
        </div>
    </div>
    <div class="card">
```

```
                        <div class="cardinner">
                            <h3>
                                <b>AB-</b>
                            </h3>
                            <h4>{{abn}}</h4>
                        </div>
                    </div>
                    <div class="card">
                        <div class="cardinner">
                            <h3>
                                <b>O+</b>
                            </h3>
                            <h4>{{op}}</h4>
                        </div>
                    </div>
                    <div class="card">
                        <div class="cardinner">
                            <h3>
                                <b>O-</b>
                            </h3>
                            <h4>{{on}}</h4>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</body>

</html>
```

Search.CSS

```
body {
  font-family: "Jost", sans-serif;
}

.navlink {
  text-decoration: none;
  color: rgb(226, 219, 219);
  margin: 15px;
  margin-top: 20px;
}

.navbar {
  display: flex;
  flex-direction: row;
  align-items: center;
  justify-content: space-between;
  height: 30px;
  background-color: #c80428;
  padding-top: 10px;
}

.left-nav {
  display: flex;
  flex-direction: row;
  align-items: center;
  justify-content: flex-start;
}

.right-nav {
  display: flex;
  flex-direction: row;
  align-items: center;
  justify-content: flex-end;
}

.titlediv {
  padding-left: 60px;
  color: white;
  font-size: 2.3em;
  font-weight: bold;
}

.contentbody {
  margin: 0;
  padding: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  font-family: "Jost", sans-serif;
  background: linear-gradient(to bottom, #c80428, #ee3255, #a80221);
}
.main {
  width: 400px;
  height: 500px;
  background: red;
  background: url("https://doc-08-2c-
docs.googleusercontent.com/docs/securesc/68c90smiglihng9534mvqmq1946dmis5/fo0picsp1
nhiucmc0l25s29respgpr4j/1631524275000/03522360960922298374/0352236096092229837
4/1Sx0jhdpEpnNIydS4rnN4kHSJtU1EyWka?e=view&authuser=0&nonce=gcrocepgbb17m
&user=03522360960922298374&hash=tfhgbs86ka6divo3llbvp93mg4csvb38")
    no-repeat center/ cover;
  border-radius: 10px;
  box-shadow: 5px 20px 50px #000;
}
#chk {
  display: none;
}
.signup {
  position: relative;
  width: 100%;
  height: 100%;
}
label {
  color: #fff;
  font-size: 2.3em;
  justify-content: center;
  display: flex;
  margin: 20px;
  font-weight: bold;
  cursor: pointer;
  transition: 0.5s ease-in-out;
}
input {
  width: 60%;
  height: 20px;
  background: #e0dede;
  justify-content: center;
  display: flex;
  margin: 20px auto;
  padding: 10px;
  border: none;
  outline: none;
  border-radius: 5px;
}

.dropdown {
  width: 66%;
  height: 50px;
  background: #e0dede;
  justify-content: center;
  display: flex;
  margin:  20px auto;
  padding: 10px;
  border: none;
  outline: none;
  border-radius: 5px;
}
button {
  width: 60%;
  height: 40px;
  margin:  10px auto;
  justify-content: center;
  display: block;
  color: #fff;
  background: #a80221;
  font-size: 1em;
  font-weight: bold;
  margin-top:  20px;
  outline: none;
  border: none;
  border-radius: 5px;
  transition: 0.2s ease-in;
  cursor: pointer;
}
button:hover {
  background: #a80221;
}
.login {
  height: 460px;
  background: #eee;
  border-radius: 60% / 10%;
  transform: translateY(-180px);
  transition: 0.8s ease-in-out;
}
.login label {
  color: #c80428;
  transform: scale(0.6);
}

#chk:checked ~ .login {
  transform: translateY(-500px);
}
#chk:checked ~ .login label {
  transform: scale(1);
}
#chk:checked ~ .signup label {
  transform: scale(0.6);
}

.cardcontainer {
  display: flex;
  flex-wrap: wrap;
  padding: 20px;
}

h3,
h4 {
  margin: 3px;
}

.card {
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);
  transition: 0.3s;
  margin: 3px;
  width: 80px;
  background-color: white;
  border-radius: 5px;
}

.cardinner {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  padding: 10px 0 10px 0;
}
```



7.2 Feature 2

Send Grid

Request.HTML

```
<!DOCTYPE html>
<html>

<head>
   <title>App</title>
   <link href="https://fonts.googleapis.com/css2?family=Jost:wght@500&display=swap"
rel="stylesheet">
   <link rel="stylesheet" type="text/css" href="{{url_for('static', filename='request.css')}}">
</head>

<body>
   <div class="navbar">
      <div class="left-nav">
         <div class="titlediv">
            Plasma Donor
         </div>
         <a class="navlink" href="{{url_for('search')}}">Search</a>
         <a class="navlink" href="{{url_for('donation')}}">Want to donate?</a>
      </div>
      <div class="right-nav">
         <a class="navlink" href="{{url_for('logout')}}">Logout</a>
      </div>
```

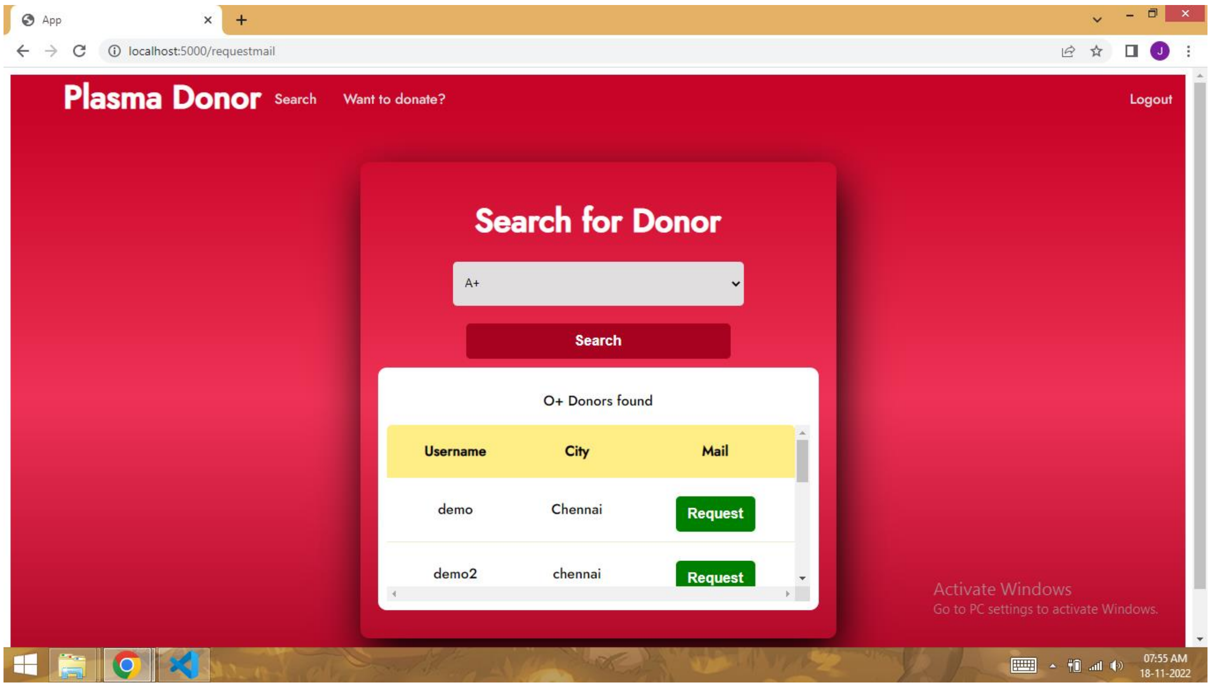
```
</div>
<div class="contentbody">
    <div class="main">
        <div class="signup">
            <form action="/requestmail"  method="POST">
                <label>Search  for Donor</label>
                <select class="dropdown" name="blood" required="">
                    <option value="A+">A+</option>
                    <option value="A-">A-</option>
                    <option value="B+">B+</option>
                    <option value="B-">B-</option>
                    <option value="AB+">AB+</option>
                    <option value="AB-">AB-</option>
                    <option value="O+">O+</option>
                    <option value="O-">O-</option>
                </select>
                <button>Search</button>
            </form>
            <div class="cardcontainer">
                <p>{{msg}}</p>
                <div class="scrollbar">
                    <table>
                        <thead>
                            <tr>
                                {% for header in headings %}
                                <th>{{ header }}</th>
                                {% endfor %}
                            </tr>
                        </thead>
                        <tbody>
                            {% for row in account %}
                            <tr>
                                <form action="/sendmail"  method="POST">
                                    {% for i in range(2) %}
                                    <td>{{ row[i] }}</td>
                                    {% endfor %}
                                    <td>
                                        <button class="requestbtn"  name="mailbtn"
                                            value="{{row[2]}}">Request</button>
                                    </td>
                                </form>
                            </tr>
                            {% endfor %}
                        </tbody>
                    </table>
                </div>
            </div>
        </div>
    </div>
</div>
</body>

</html>
```

Request.CSS

```
body {
  font-family: "Jost", sans-serif;
}

.navlink {
  text-decoration: none;
  color: rgb(226, 219, 219);
  margin: 15px;
  margin-top: 20px;
}

.navbar {
  display: flex;
  flex-direction: row;
  align-items: center;
  justify-content: space-between;
  height: 30px;
  background-color: #c80428;
  padding-top: 10px;
}

.left-nav {
  display: flex;
  flex-direction: row;
  align-items: center;
  justify-content: flex-start;
}

.right-nav {
  display: flex;
  flex-direction: row;
  align-items: center;
  justify-content: flex-end;
```

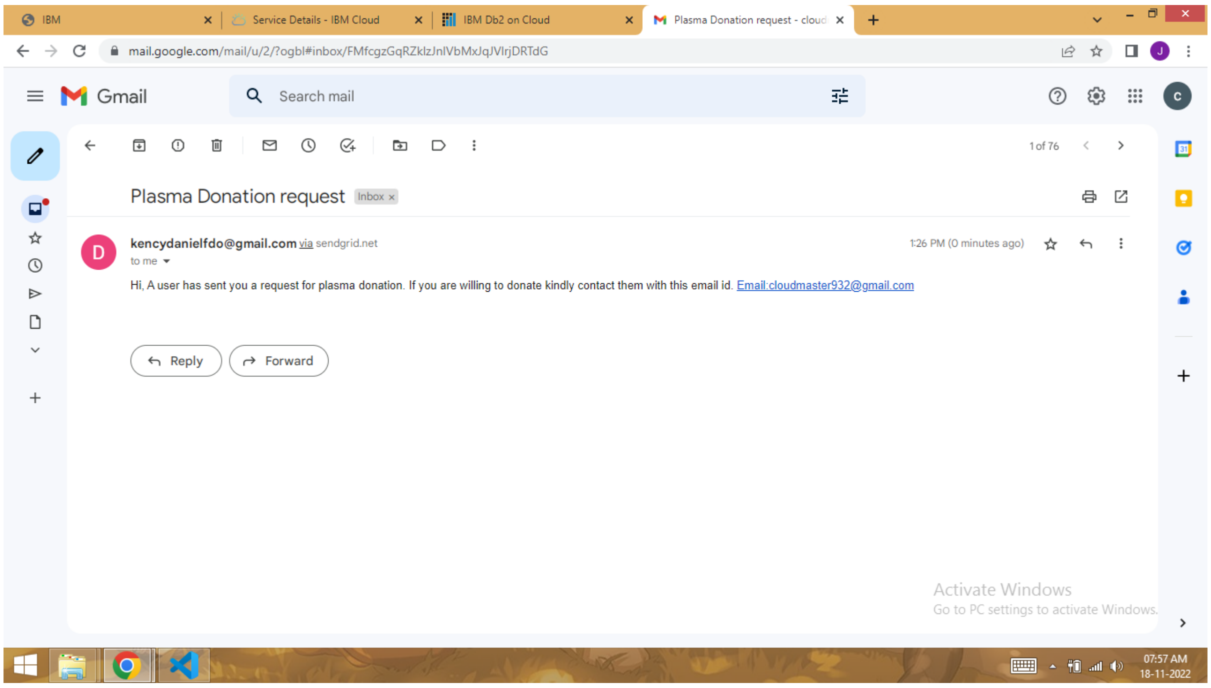
```
}

.titlediv {
  padding-left: 60px;
  color: white;
  font-size: 2.3em;
  font-weight: bold;
}

.contentbody {
  margin: 0;
  padding: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  font-family: "Jost", sans-serif;
  background: linear-gradient(to bottom, #c80428, #ee3255, #a80221);
}
.main {
  width: 500px;
  height: 500px;
  background: red;
  background: url("https://doc-08-2c-
docs.googleusercontent.com/docs/securesc/68c90smiglihng9534mvqmq1946dmis5/fo0picsp1
nhiucmc0l25s29respgpr4j/1631524275000/03522360960922298374/0352236096092229837
4/1Sx0jhdpEpnNIydS4rnN4kHSJtU1EyWka?e=view&authuser=0&nonce=gcrocepgbb17m
&user=03522360960922298374&hash=tfhgbs86ka6divo3llbvp93mg4csvb38")
    no-repeat center/ cover;
  border-radius: 10px;
  box-shadow: 5px 20px 50px #000;
  padding: 20px;
}
#chk {
  display: none;
}
.signup {
  position: relative;
  width: 100%;
  height: 100%;
}
label {
  color: #fff;
  font-size: 2.3em;
  justify-content: center;
  display: flex;
  margin: 20px;
  font-weight: bold;
  cursor: pointer;
  transition: 0.5s ease-in-out;
}

input {
  width: 60%;
  height: 20px;
  background: #e0dede;
  justify-content: center;
  display: flex;
  margin: 20px auto;
  padding: 10px;
  border: none;
  outline: none;
  border-radius: 5px;
}

.dropdown {
  width: 66%;
  height: 50px;
  background: #e0dede;
  justify-content: center;
  display: flex;
  margin: 20px auto;
  padding: 10px;
  border: none;
  outline: none;
  border-radius: 5px;
}
button {
  width: 60%;
  height: 40px;
  margin: 10px auto;
  justify-content: center;
  display: block;
  color: #fff;
  background: #a80221;
  font-size: 1em;
  font-weight: bold;
  margin-top: 20px;
  outline: none;
  border: none;
  border-radius: 5px;
  transition: 0.2s ease-in;
  cursor: pointer;
}
button:hover {
  background: #a80221;
}
.login {
  height: 460px;
  background: #eee;
  border-radius: 60% / 10%;
  transform: translateY(-180px);
  transition: 0.8s ease-in-out;
}
.login label {
  color: #c80428;
  transform: scale(0.6);
}

#chk:checked ~ .login {
  transform: translateY(-500px);
}
#chk:checked ~ .login label {
  transform: scale(1);
}
#chk:checked ~ .signup label {
  transform: scale(0.6);
}

.cardcontainer {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  padding: 10px;
  background-color: white;
  border-radius: 10px;
}

.scrollbar {
  overflow: scroll;
  height: 200px;
  width: 100%;
}

table {
  width: 450px;
}
.requestbtn {
  background-color: green;
  width: 90px;
}

table {
  border-spacing: 1;
  border-collapse: collapse;
  background: white;
  border-radius: 6px;
  overflow: hidden;
  max-width: 800px;
  width: 100%;
  margin: 0 auto;
  position: relative;
}
table * {
  position: relative;
}
table td,
table th {
  padding-left: 8px;
}
table thead tr {
  height: 60px;
  background: #ffed86;
  font-size: 16px;
}
table tbody tr {
  height: 48px;
  border-bottom: 1px solid #e3f1d5;
}
table tbody tr:last-child {
  border: 0;
}
table td,
table th {
  text-align: center;
```

```
}
table td.l,
table th.l {
  text-align: right;
}
table td.c,
table th.c {
  text-align: center;
}
table td.r,
table th.r {
  text-align: center;
}

@media screen and (max-width: 35.5em) {
  table {
    display: block;
  }
  table > *,
  table tr,
  table td,
  table th {
    display: block;
  }
  table thead {
    display: none;
  }
  table tbody tr {
    height: auto;
    padding: 8px 0;
  }
  table tbody tr td {
    padding-left: 45%;
    margin-bottom: 12px;
  }
  table tbody tr td:last-child {
    margin-bottom: 0;
  }
  table tbody tr td:before {
    position: absolute;
    font-weight: 700;
    width: 40%;
    left: 10px;
    top: 0;
  }
  table tbody tr td:nth-child(1):before {
    content: "Code";
  }
  table tbody tr td:nth-child(2):before {
    content: "Stock";
  }
  table tbody tr td:nth-child(3):before {
    content: "Cap";
  }
  table tbody tr td:nth-child(4):before {
    content: "Inch";
  }
  table tbody tr td:nth-child(5):before {
    content: "Box Type";
  }
}

blockquote {
  color: white;
  text-align: center;
}
```



O+ Donors found

Username	City	Mail
demo	Chennai	Request
demo2	chennai	Request

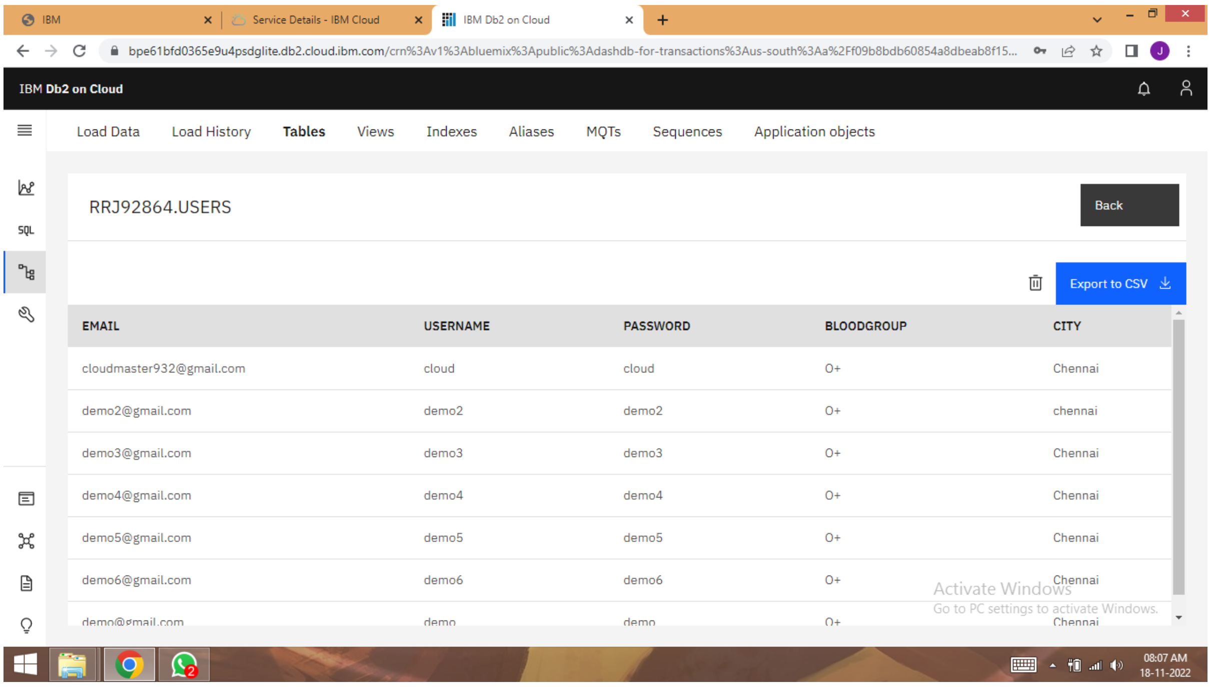


kencydanielfdo@gmail.com via sendgrid.net
to me

1:26 PM (0 minutes ago)

Hi, A user has sent you a request for plasma donation. If you are willing to donate kindly contact them with this email id. Email:cloudmaster932@gmail.com

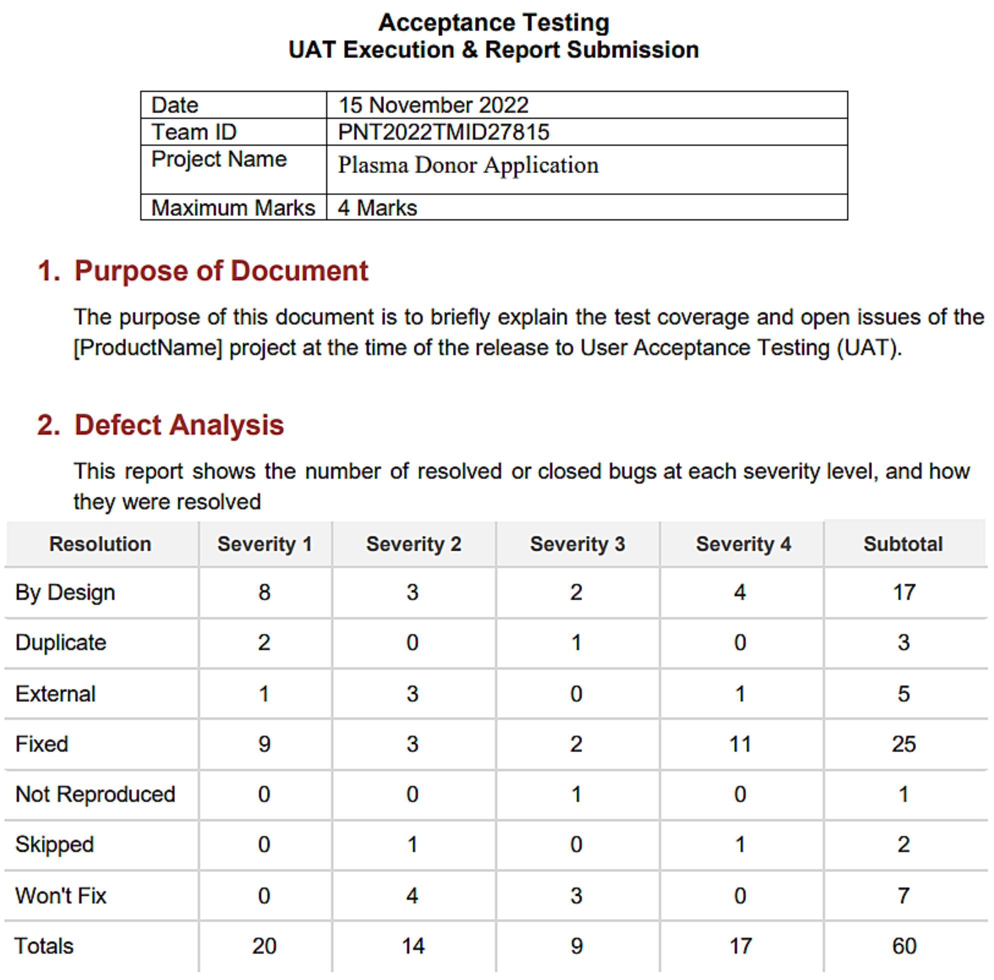
7.3 Database Schema



EMAIL	USERNAME	PASSWORD	BLOODGROUP	CITY
cloudmaster932@gmail.com	cloud	cloud	O+	Chennai
demo2@gmail.com	demo2	demo2	O+	chennai
demo3@gmail.com	demo3	demo3	O+	Chennai
demo4@gmail.com	demo4	demo4	O+	Chennai
demo5@gmail.com	demo5	demo5	O+	Chennai
demo6@gmail.com	demo6	demo6	O+	Chennai
demo@gmail.com	demo	demo	O+	Chennai

8. TESTING

8.1 Test Cases



Acceptance Testing
UAT Execution & Report Submission

Date	15 November 2022
Team ID	PNT2022TMID27815
Project Name	Plasma Donor Application
Maximum Marks	4 Marks

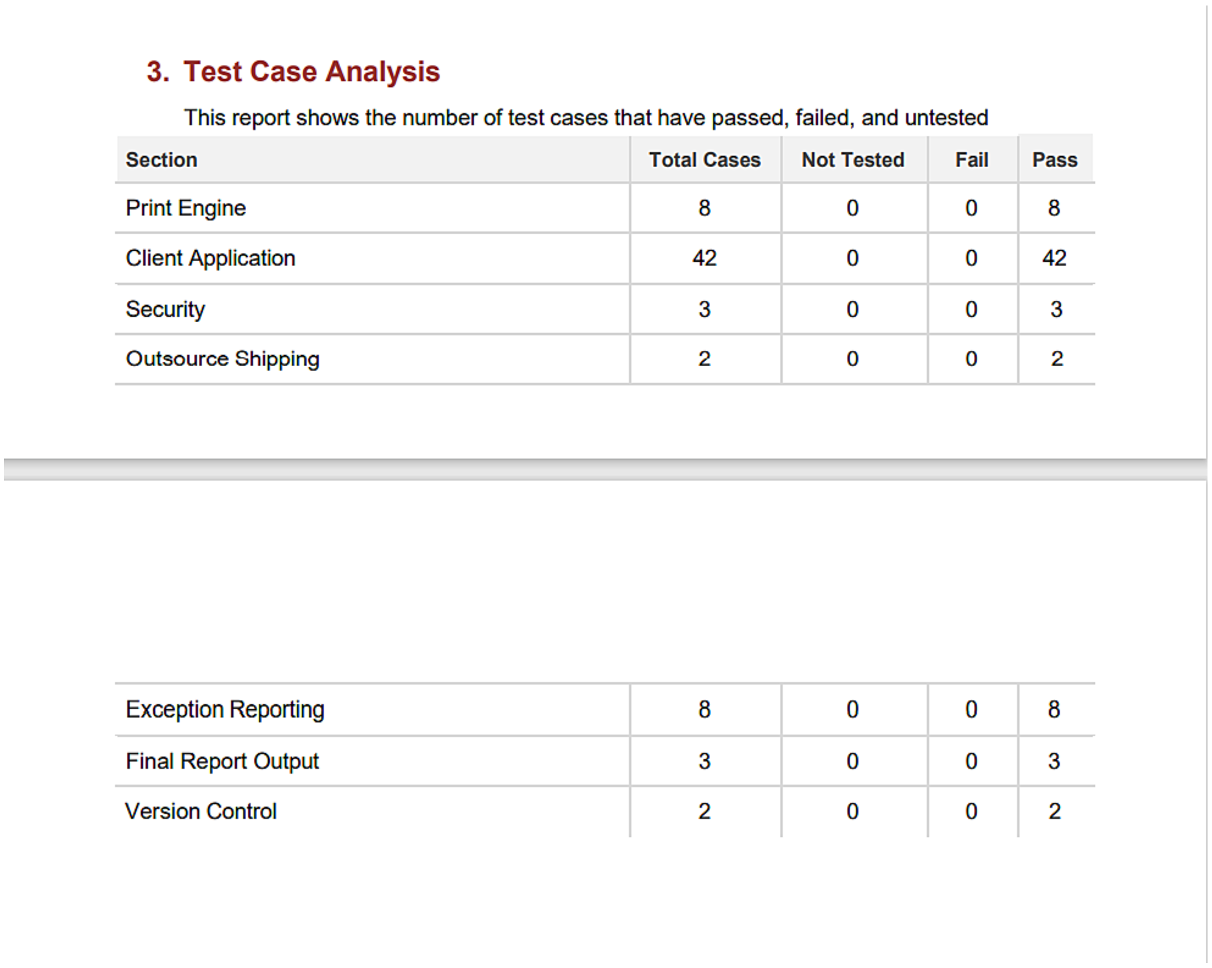
1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	8	3	2	4	17
Duplicate	2	0	1	0	3
External	1	3	0	1	5
Fixed	9	3	2	11	25
Not Reproduced	0	0	1	0	1
Skipped	0	1	0	1	2
Won't Fix	0	4	3	0	7
Totals	20	14	9	17	60

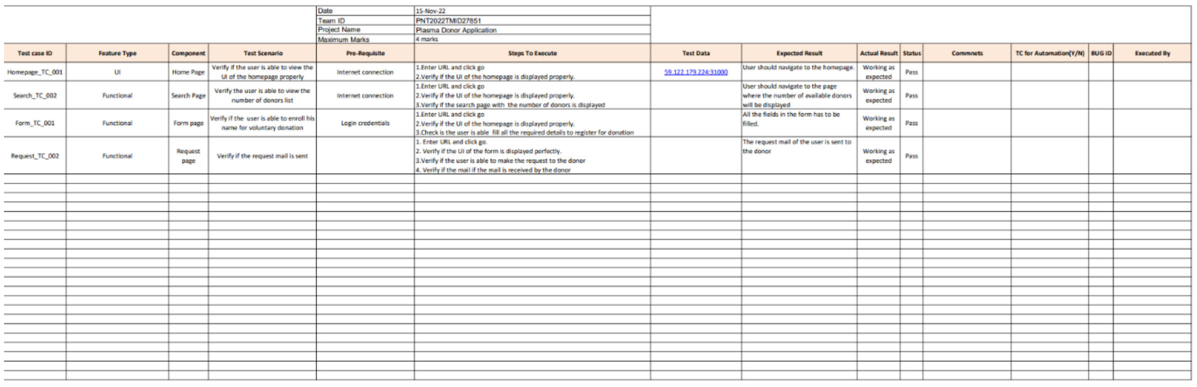


3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	8	0	0	8
Client Application	42	0	0	42
Security	3	0	0	3
Outsource Shipping	2	0	0	2
Exception Reporting	8	0	0	8
Final Report Output	3	0	0	3
Version Control	2	0	0	2

8.2 User Acceptance Testing



Date	15-Nov-22
Team ID	PNT2022TMID27851
Project Name	Plasma Donor Application
Maximum Marks	4 marks

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
Homepage_TC_001	UI	Home Page	Verify if the user is able to view the UI of the homepage properly	Internet connection	1.Enter URL and click go 2.Verify if the UI of the homepage is displayed properly.	59.122.179.224:31000	User should navigate to the homepage.	Working as expected	Pass				
Search_TC_002	Functional	Search Page	Verify the user is able to view the number of donors list	Internet connection	1.Enter URL and click go 2.Verify if the UI of the homepage is displayed properly. 3.Verify if the search page with the number of donors is displayed		User should navigate to the page where the number of available donors will be displayed	Working as expected	Pass				
Form_TC_001	Functional	Form page	Verify if the user is able to enroll his name for voluntary donation	Login credentials	1.Enter URL and click go 2.Verify if the UI of the homepage is displayed properly. 3.Check is the user is able fill all the required details to register for donation		All the fields in the form has to be filled.	Working as expected	Pass				
Request_TC_002	Functional	Request page	Verify if the request mail is sent		1. Enter URL and click go. 2. Verify if the UI of the form is displayed perfectly. 3.Verify if the user is able to make the request to the donor 4. Verify if the mail if the mail is received by the donor		The request mail of the user is sent to the donor	Working as expected	Pass				

9. RESULTS

9.1 Performance Metrics

Model Performance testing

The application is used by the user to interact and register for the plasma donation or to make request. Even when performing multiple switches between tabs and buttons the application responded correctly and provided a good result. It can be scaled to provide the results even faster by reducing the number of requests from the database and increasing the number of tables.

Accuracy testing

The data provided by the user and the data stored in the databases are found to be accurate. And the hitting of API gave correct results.

10. ADVANTAGES & DISADVANTAGES

Advantages:

1) **Round the clock support:** Since it is a cloud deployment, there will be no down time as the cloud servers are maintained by the service providers. So, the expected customer will have seamless access to the application
2) **Easy access to the plasma donor:** The application is user friendly and easy to use by anyone. It is very simple to find a suitable plasma donor by a single click.
3) **Communicate with email:** The application is integrated with sendgrid to provide email services; thus, the requesting user can directly communicate with the donors.

Disadvantages:

1) **Internet Issues:** Since the application is cloud based it can be only accessed through internet, unstable internet connection will be the major drawback for it.
2) **Less secure:** The passwords are not encrypted, so if any one tries to attack, it will cause data leaks.
3) **Slow Rendering of data:** Since there is a need to update every change in the data regularly, the application pulls the data from the database frequently. This causes the app to render slowly.

11. CONCLUSION

The corona pandemic showed us that it is not easy to get plasma for treatment that quickly. Even hospitals and plasma banks don’t have enough stock for every patient. There was a need for a solution to get plasma easily available for all. Our application aims to achieve that goal by allowing ever willing plasma donors to register for donation. Since it is a cloud-based application, it will be running round the clock. Any user can register to our application and they can opt to donate their plasma if they are willing to, thus one who wants to just request for plasma is not forced to donate. It also makes search for plasma donor easy by listing the available plasma donor and sort that list based on the required blood type. Email services provided enables communication between the requesting user and the donor simple.

12. FUTURE SCOPE

1) **Chatbot integration:** To improve the user experience we'll be integrating the chatbot in the application. Which will be answering the basic queries raised by the user.

2) **Donor-user chatting facility:** Secondly, we plan to provide a chatting facility to make communication easy between the donor and the one who is requesting.

3) **Find nearby donation centre:** We are also planning to provide a feature that finds a nearby donation centre to make it easy for the donors.

13. APPENDIX

Source Code

Python flask code:

APP.PY

```
from flask import Flask, render_template, redirect, url_for, request, session
import ibm_db
import re
import sendgrid
from sendgrid.helpers.mail import Mail, Email, To, Content
import socket

hostname = socket.gethostname()
ip = socket.gethostbyname(hostname)

app = Flask(__name__)
app.secret_key = 'a'

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954-
7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32733;SECURITY
=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt; UID=rrj92864;
PWD=maDr7SBpdojnrqgv;", '', '')

@app.route('/')
def home():
    return render_template('login.html', ip=ip)
@app.route('/login', methods=["GET", "POST"])
def login():
    global userid
    msg = " "
    if request.method == "POST":
        email = request.form['email']
        password = request.form['pwd']
        sql = "SELECT * FROM Users WHERE EMAIL=? AND PASSWORD=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.bind_param(stmt, 2, password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['loggedin'] = True
            session['id'] = account['USERNAME']
            userid = account["USERNAME"]
            session['email'] = account["EMAIL"]
            msg = 'Logged in successfully!'
            return redirect(url_for('search'))
        else:
            msg = "Incorrect Username/Password"
            return render_template('login.html', msg=msg, ip=ip)
@ app.route('/signup', methods=["GET", "POST"])
def signup():
    msg = " "
    if request.method == "POST":
        username = request.form['username']
        email = request.form['email']
        password = request.form['pwd']
        sql = "SELECT * FROM USERS 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 = "Account already exists!"
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = "Invalid Email Address."
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = "Username must contain only alphabets and numbers."
        else:
            insert_sql = "INSERT INTO USERS(EMAIL, USERNAME, PASSWORD)
VALUES(?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, email)
            ibm_db.bind_param(prep_stmt, 2, username)
            ibm_db.bind_param(prep_stmt, 3, password)
            ibm_db.execute(prep_stmt)
            msg = "You have successfully registered."
        return render_template('login.html', msg=msg, ip=ip)

    elif request.method == 'POST':
        msg = "Please fill out the form."
        return render_template('login.html', msg=msg, ip=ip)
@ app.route('/search')
def search():
    msg = " "
    username = session['id']
    sql = "SELECT COUNT(USERNAME) AS COUNT FROM USERS WHERE
BLOODGROUP=?;"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt, 1, 'A+')
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    ap = account['COUNT']
    ibm_db.bind_param(stmt, 1, 'A-')
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    an = account['COUNT']
    ibm_db.bind_param(stmt, 1, 'B+')
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    bp = account['COUNT']
    ibm_db.bind_param(stmt, 1, 'B-')
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    bn = account['COUNT']
    ibm_db.bind_param(stmt, 1, 'AB+')
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    abp = account['COUNT']
    ibm_db.bind_param(stmt, 1, 'AB-')
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    abn = account['COUNT']
    ibm_db.bind_param(stmt, 1, 'O+')
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    op = account['COUNT']
    ibm_db.bind_param(stmt, 1, 'O-')
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    on = account['COUNT']
    return render_template('search.html', ap=ap, an=an, bp=bp, bn=bn, abp=abp, abn=abn,
op=op, on=on, ip=ip)
@ app.route('/requestmail', methods=["GET", "POST"])
def requestmail():
    if request.method == "POST":
        blood = request.form['blood']
        sql = "SELECT USERNAME, CITY, EMAIL FROM USERS WHERE
BLOODGROUP=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, blood)
        ibm_db.execute(stmt)
        account = []
        while ibm_db.fetch_row(stmt) != False:
            tmp = []
            for i in range(3):
                tmp.append(ibm_db.result(stmt, i))
            account.append(tuple(tmp))
        headings = ['Username', 'City', 'Mail']
        msg = ""
        if (len(account) > 0):
            msg = blood + " Donors found"
        else:
            msg = blood + " Donors not found"
        return render_template('request.html', headings=headings, msg=msg, account=account,
ip=ip)
    else:
        return render_template('request.html', ip=ip)
@ app.route('/donation', methods=["GET", "POST"])
def donation():

    msg = " "
    if request.method == "POST":
        blood = request.form['blood']
        city = request.form['city']
        username = session['id']
        update_sql = "UPDATE USERS SET CITY=?, BLOODGROUP=? WHERE
USERNAME=?;"
        stmt = ibm_db.prepare(conn, update_sql)
        ibm_db.bind_param(stmt, 1, city)
        ibm_db.bind_param(stmt, 2, blood)
        ibm_db.bind_param(stmt, 3, username)
        ibm_db.execute(stmt)
        msg = "You have successfully registered"
        return render_template('donation.html', msg=msg, ip=ip)
    else:
        msg = "Please fill out the form."
        return render_template('donation.html', msg=msg, ip=ip)
def sendgridmail(to_mail_id):
    try:
        sg = sendgrid.SendGridAPIClient(
            '')
    # Change to your verified sender
        from_email = Email("kencydanielfdo@gmail.com")
        to_email = To(to_mail_id)  # Change to your recipient
        subject = "Plasma Donation request "
        htmlcontent = "Hi, A user has sent you a request for plasma donation. If you are willing
to donate kindly contact them with this email id. Email: " + \
            session['email']
        content = Content("text/plain", htmlcontent)
        mail = Mail(from_email, to_email, subject, content)
    # Get a JSON-ready representation of the Mail object
        mail_json = mail.get()
    # Send an HTTP POST request to /mail/send
        response = sg.client.mail.send.post(request_body=mail_json)
        print(response.status_code)
    except Exception as e:
        print(e.message)
@ app.route('/sendmail', methods=["GET", "POST"])
def sendmail():
    if request.method == "POST":
        receivermail = request.form['mailbtn']
        sendgridmail(receivermail)
        return redirect(url_for('search'))
@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('email', None)
    return render_template('login.html', ip=ip)
if __name__ == "__main__":
    app.run(host='0.0.0.0')
```

Login.HTML

```
<!DOCTYPE html>
<html>

<head>
    <title>App</title>
    <link href="https://fonts.googleapis.com/css2?family=Jost:wght@500&display=swap"
rel="stylesheet">
    <link rel="stylesheet" type="text/css" href="{{url_for('static', filename='login.css')}}">
</head>

<body>
    <div class="navbar">
        <div class="titlediv">
            Plasma Donor
        </div>

    </div>
    <div class="contentbody">
        <div class="main">
            <input type="checkbox" id="chk" aria-hidden="true">

            <div class="signup">
                <form action="/signup"  method="POST">
                    <label for="chk" aria-hidden="true">Sign up</label>
                    <input type="text" name="username"  placeholder="User name" required="">
                    <input type="email"  name="email"  placeholder="Email"  required="">
                    <input type="password" name="pwd" placeholder="Password"  required="">
                    <button>Sign up</button>
                </form>
            </div>

            <div class="login">
                <form action="/login"  method="POST">
                    <label for="chk" aria-hidden="true">Login</label>
                    <input type="email"  name="email"  placeholder="Email"  required="">
                    <input type="password" name="pwd" placeholder="Password"  required="">
                    <button>Login</button>
                </form>
            </div>
        </div>
    </div>
</body>

</html>
```

Login.CSS

```
body{
            font-family: 'Jost', sans-serif;
}

.navbar{
  height: 30px;
  background-color: #c80428;
  padding-top: 10px;
}

.titlediv{
  padding-left: 60px;
  color: white;
  font-size: 2.3em;
	font-weight: bold;
}

.contentbody{
  margin: 0;
	padding: 0;
	display: flex;
	justify-content: center;
	align-items: center;
	min-height: 100vh;
	font-family: 'Jost', sans-serif;
	background: linear-gradient(to bottom, #c80428, #ee3255, #a80221);
}
.main{
	width: 350px;
	height: 500px;
	background: red;
	overflow: hidden;
	background: url("https://doc-08-2c-
docs.googleusercontent.com/docs/securesc/68c90smiglihng9534mvqmq1946dmis5/fo0picsp1
nhiucmc0l25s29respgpr4j/1631524275000/03522360960922298374/0352236096092229837
4/1Sx0jhdpEpnNIydS4rnN4kHSJtU1EyWka?e=view&authuser=0&nonce=gcrocepgbb17m
&user=03522360960922298374&hash=tfhgbs86ka6divo3llbvp93mg4csvb38") no-repeat
center/ cover;
	border-radius: 10px;
	box-shadow: 5px 20px 50px #000;
}
#chk{
	display: none;
}
.signup{
	position: relative;
	width:100%;
	height: 100%;
}
label{
	color: #fff;
	font-size: 2.3em;
	justify-content: center;
	display: flex;
	margin: 60px;
	font-weight: bold;
```

```
        cursor: pointer;
        transition: .5s ease-in-out;
}
input{
        width: 60%;
        height: 20px;
        background: #e0dede;
        justify-content: center;
        display: flex;
        margin: 20px auto;
        padding: 10px;
        border: none;
        outline: none;
        border-radius: 5px;
}
button{
        width: 60%;
        height: 40px;
        margin: 10px auto;
        justify-content: center;
        display: block;
        color: #fff;
        background: #ee3255;
        font-size: 1em;
        font-weight: bold;
        margin-top: 20px;
        outline: none;
        border: none;
        border-radius: 5px;
        transition: .2s ease-in;
        cursor: pointer;
}
button:hover{
        background: #a80221;
}
.login{
        height: 460px;
        background: #eee;
        border-radius: 60% / 10%;
        transform: translateY(-180px);
        transition: .8s ease-in-out;
}
.login label{
        color: #c80428;
```

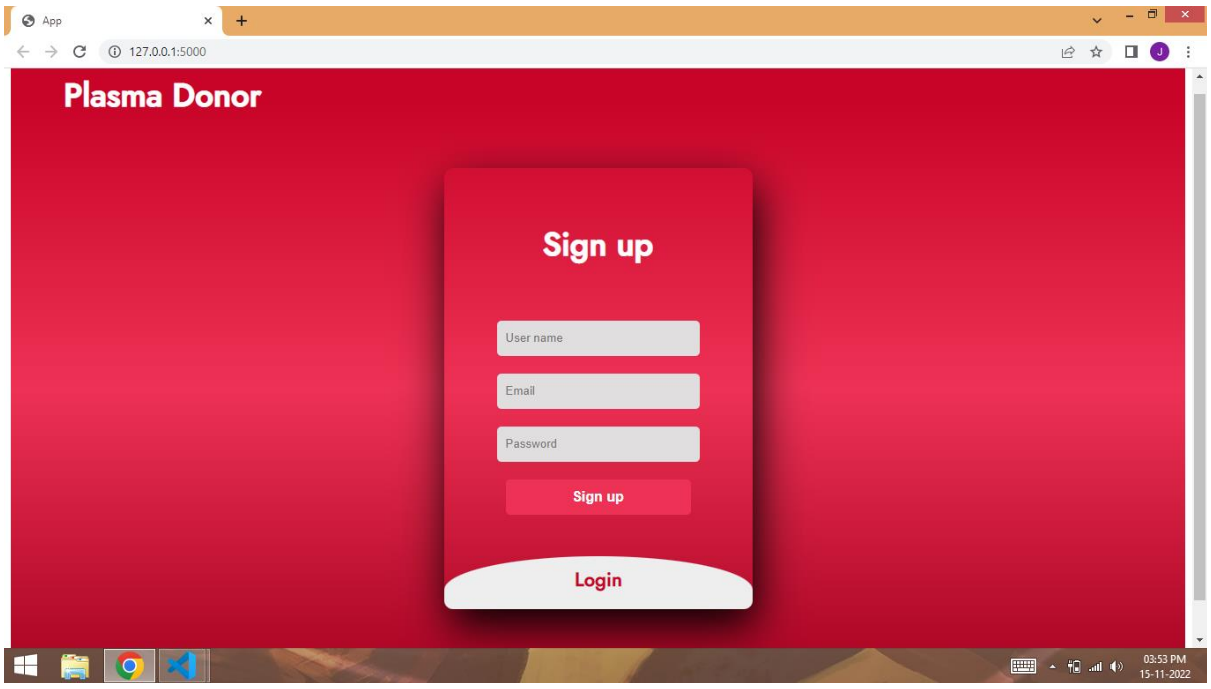
```
	transform: scale(.6);
}

#chk:checked ~ .login{
	transform: translateY(-500px);
}
#chk:checked ~ .login label{
	transform: scale(1);
}
#chk:checked ~ .signup label{
	transform: scale(.6);
}
```

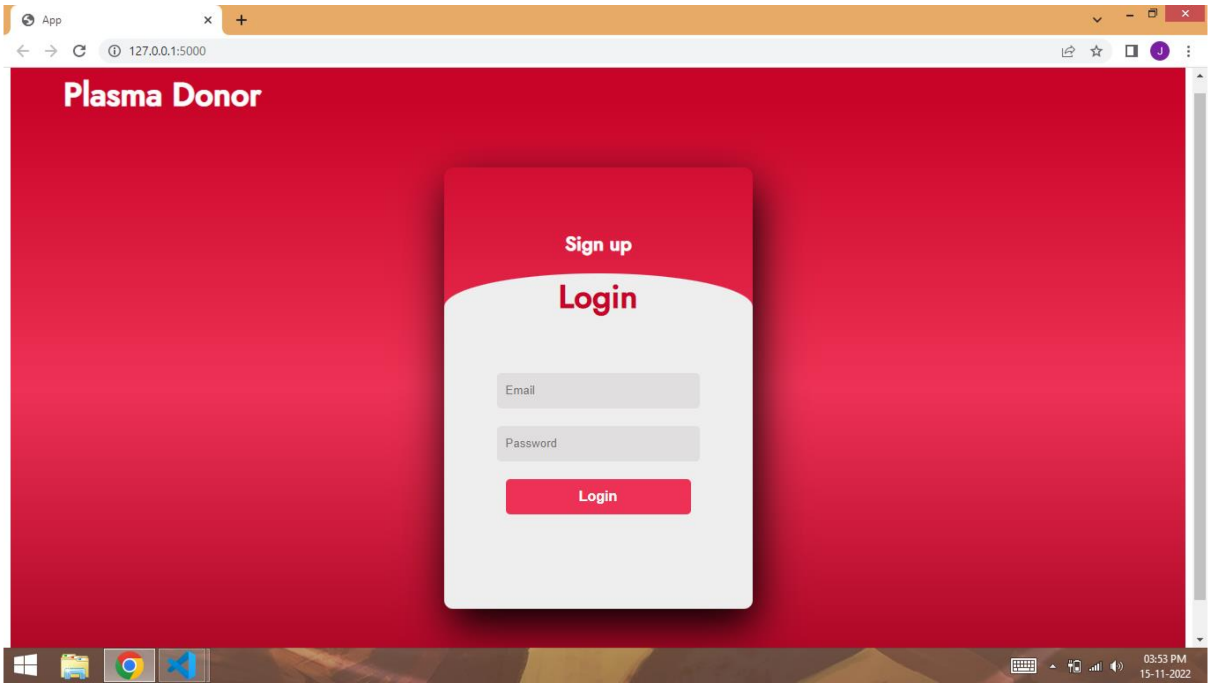
Donation.HTML

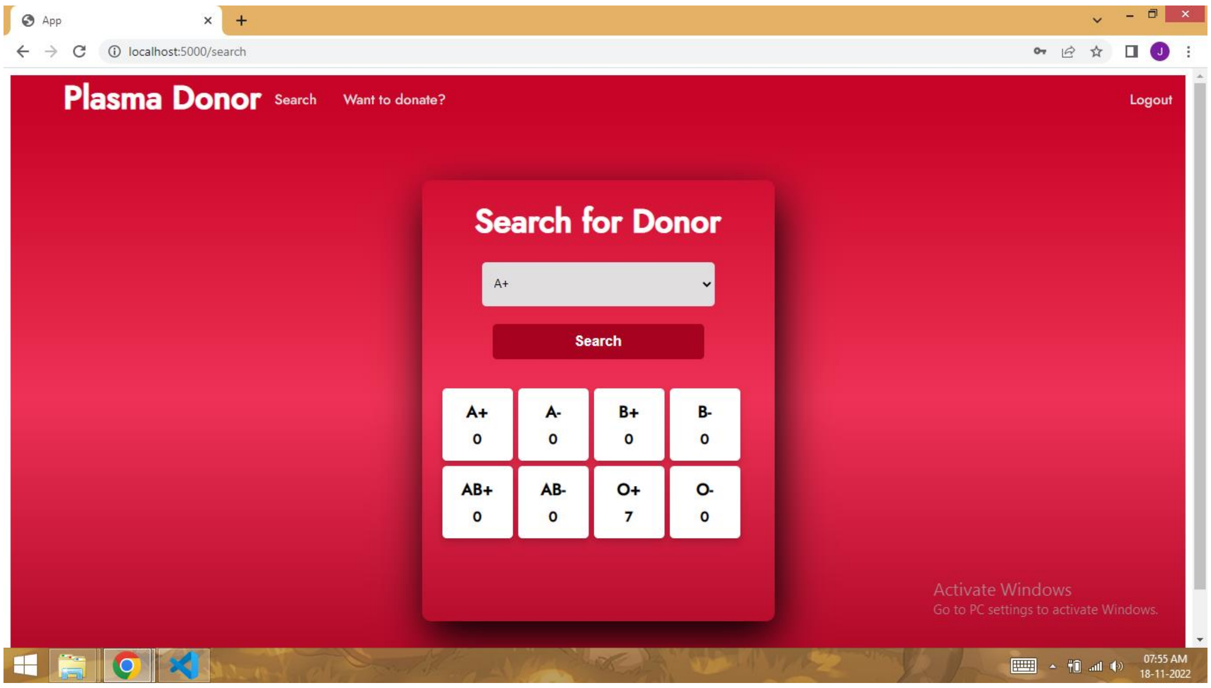
```
<!DOCTYPE html>
<html>

<head>
   <title>App</title>
   <link href="https://fonts.googleapis.com/css2?family=Jost:wght@500&display=swap"
rel="stylesheet">
   <link rel="stylesheet" type="text/css" href="{{url_for('static', filename='search.css')}}">
</head>

<body>
   <div class="navbar">
      <div class="left-nav">
         <div class="titlediv">
            Plasma Donor
         </div>
         <a class="navlink" href="{{url_for('search')}}">Search</a>
         <a class="navlink" href="{{url_for('donation')}}">Want to donate?</a>
      </div>
      <div class="right-nav">
         <a class="navlink" href="{{url_for('logout')}}">Logout</a>
      </div>
   </div>
   <div class="contentbody">
      <div class="main">
         <div class="signup">
            <form action="/donation" method="POST">
               <label for="blood">Enter Blood type</label>
               <select class="dropdown" name="blood" required="">
                  <option value="A+">A+</option>
```

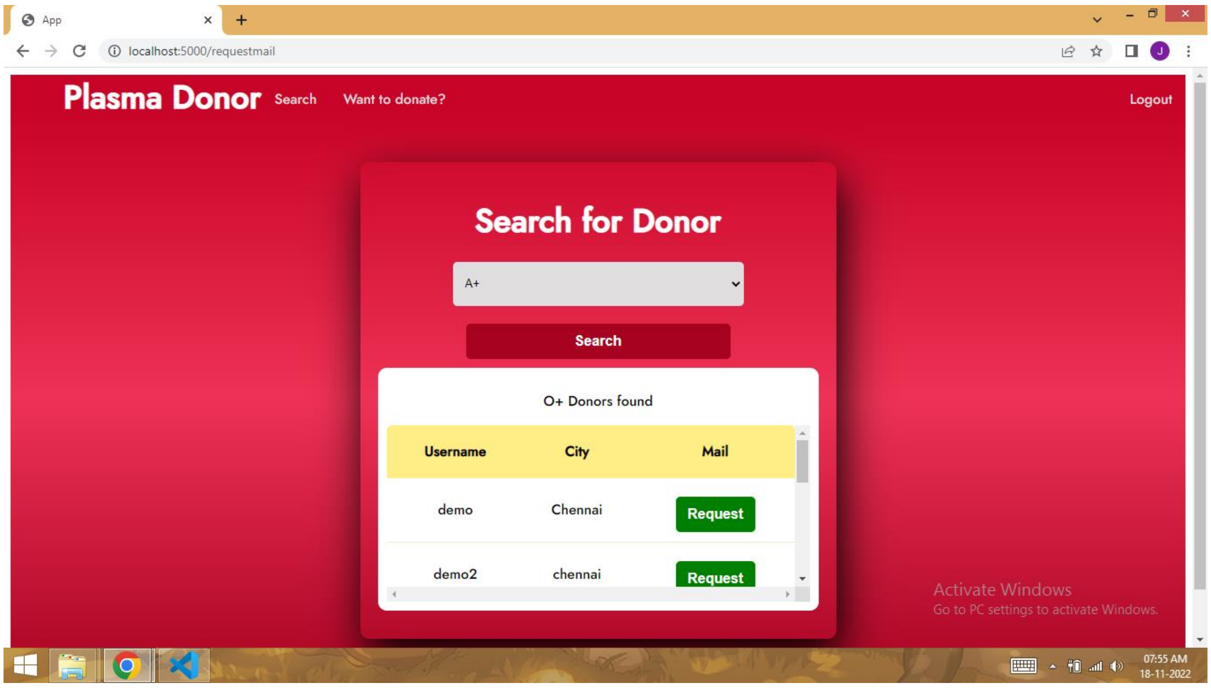
```
                <option value="A-">A-</option>
                <option value="B+">B+</option>
                <option value="B-">B-</option>
                <option value="AB+">AB+</option>
                <option value="AB-">AB-</option>
                <option value="O+">O+</option>
                <option value="O-">O-</option>
              </select>
              <input type="text" name="city" placeholder="City" required="">
              <button>Register</button>
            </form>

          </div>
        </div>
      </div>
      </div>
</body>

</html>
```

OUTPUT:



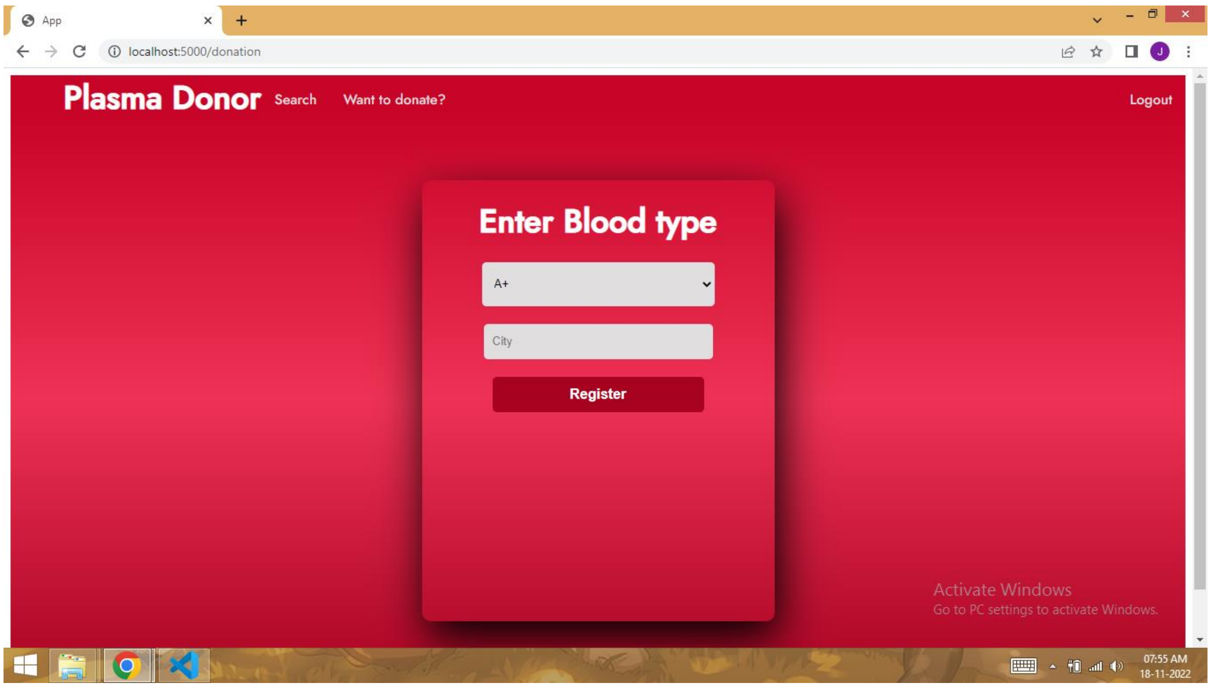

App
127.0.0.1:5000
Plasma Donor
Sign up
Login
Email
Password
Login
03:53 PM
15-11-2022

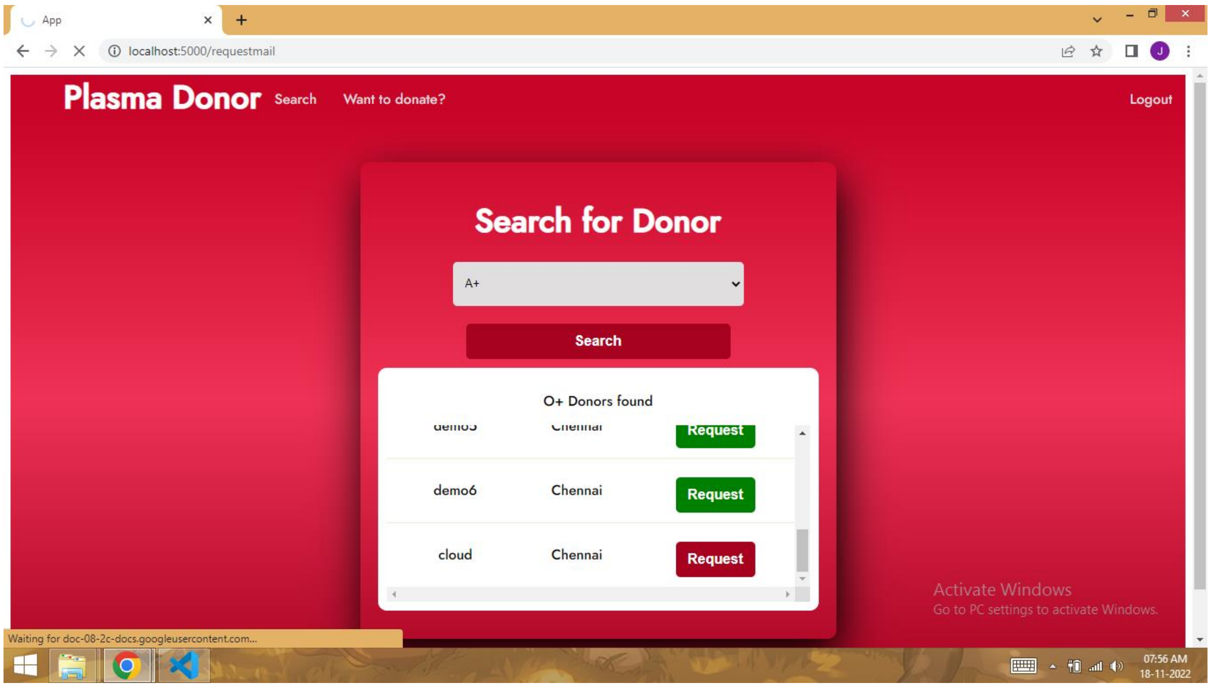

App
localhost:5000/search
Plasma Donor
Search
Want to donate?
Logout
Search for Donor
A+
Search
A+
0
A-
0
B+
0
B-
0
AB+
0
AB-
0
O+
7
O-
0
Activate Windows
Go to PC settings to activate Windows.
07:55 AM
18-11-2022



O+ Donors found

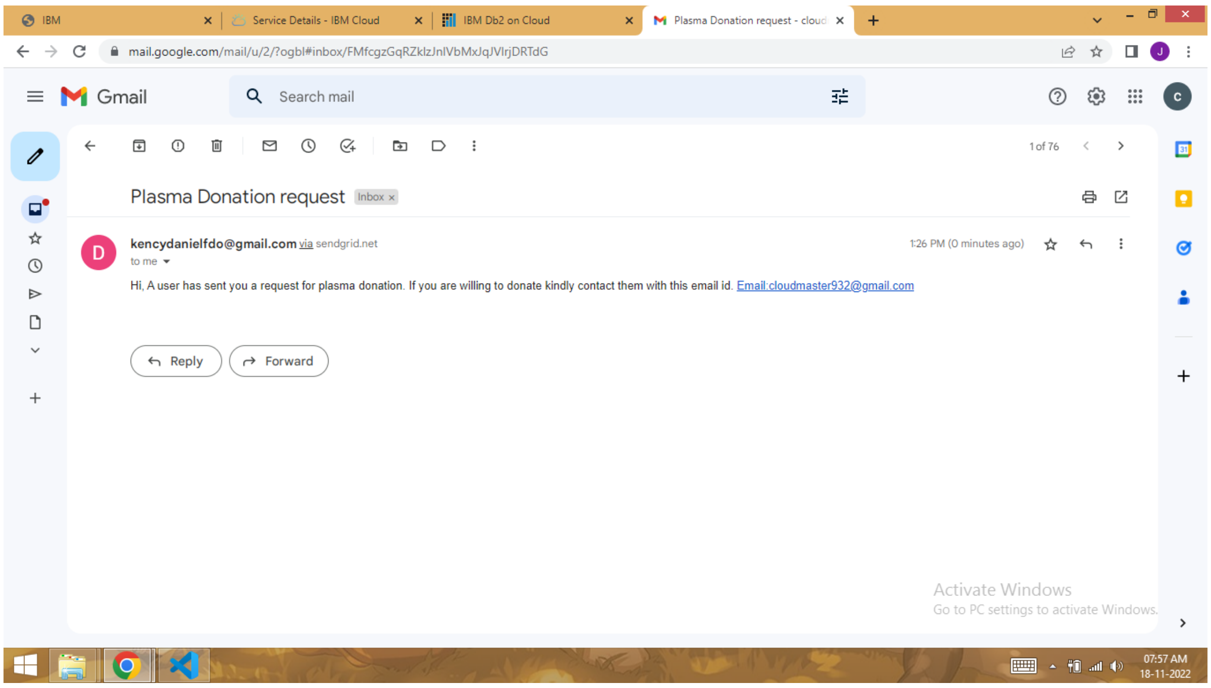
Username	City	Mail
demo	Chennai	Request
demo2	chennai	Request




App
localhost:5000/requestmail
Plasma Donor
Search
Want to donate?
Logout
Search for Donor
A+
Search
O+ Donors found

demo5	Chennai	Request
demo6	Chennai	Request
cloud	Chennai	Request

Activate Windows
Go to PC settings to activate Windows.
Waiting for doc-08-2c-docs.googleusercontent.com...
07:56 AM
18-11-2022

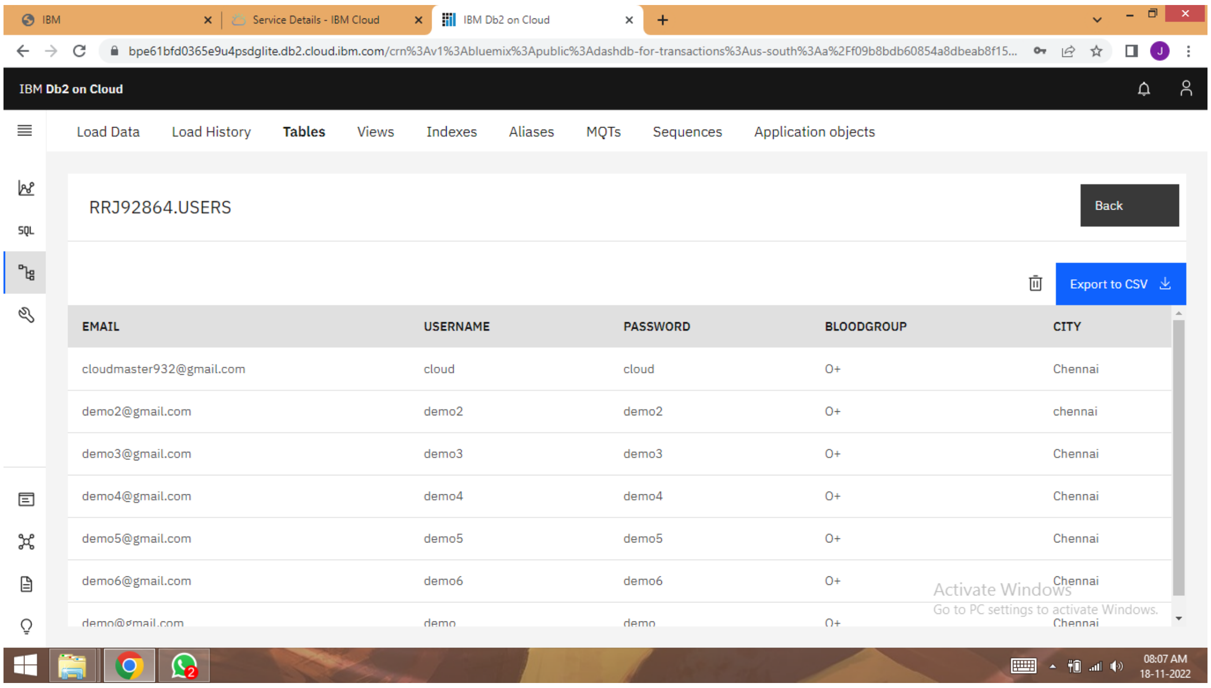

IBM
Service Details - IBM Cloud
IBM Db2 on Cloud
Plasma Donation request - cloud
mail.google.com/mail/u/2/?ogbl#inbox/FMfcgzGqRZklzJnlVbMxJqJVlrjDRTdG
Gmail
Search mail
1 of 76
Inbox

Plasma Donation request

kencydanielfdo@gmail.com via sendgrid.net
to me
1:26 PM (0 minutes ago)

Hi, A user has sent you a request for plasma donation. If you are willing to donate kindly contact them with this email id. Email:cloudmaster932@gmail.com

Reply
Forward
Activate Windows
Go to PC settings to activate Windows.
07:57 AM
18-11-2022


IBM
Service Details - IBM Cloud
IBM Db2 on Cloud
bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2Ff09b8bdb60854a8dbeab8f15...
IBM Db2 on Cloud
Load Data
Load History
Tables
Views
Indexes
Aliases
MQTs
Sequences
Application objects
SQL
Back
Export to CSV
Activate Windows
Go to PC settings to activate Windows.
08:07 AM
18-11-2022

RRJ92864.USERS

EMAIL	USERNAME	PASSWORD	BLOODGROUP	CITY
cloudmaster932@gmail.com	cloud	cloud	O+	Chennai
demo2@gmail.com	demo2	demo2	O+	chennai
demo3@gmail.com	demo3	demo3	O+	Chennai
demo4@gmail.com	demo4	demo4	O+	Chennai
demo5@gmail.com	demo5	demo5	O+	Chennai
demo6@gmail.com	demo6	demo6	O+	Chennai
demo@gmail.com	demo	demo	O+	Chennai

GitHub Project:

https://github.com/IBM-EPBL/IBM-Project-21675-1659787568

Demo Link:

https://youtu.be/kSYmpZdWFVw