PROJECT REPORT FORMAT

1. **INTRODUCTION**
   - 1.1 Project Overview
   - 1.2 Purpose
2. **LITERATURE SURVEY**
   - 2.1 Existing problem
   - 2.2 References
   - 2.3 Problem Statement Definition
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**
   - 4.1 Functional requirement
   - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   - 5.1 Data Flow Diagrams
   - 5.2 Solution & Technical Architecture
   - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
   - 6.1 Sprint Planning
   - 6.2 Sprint Estimation and Delivery Schedule
7. **CODING & SOLUTIONING**
   - 7.1 Database Schema
8. **RESULTS**
9. **ADVANTAGES & DISADVANTAGES**
10. **CONCLUSION**

1.1 INTRODUCTION

1.2 Project Overview: -

Patients with severe liver disease or numerous clotting factor deficits, as well as those who have undergone trauma, burns, or shock, frequently get plasma. The patient's blood volume is increased as a result, which can aids in blood coagulation and helps to prevent shock. The number of people with Covid-19 infection has increased, as has the demand for the plasma of patients who have recovered. The antibodies that are already in our systems can aid someone in overcoming the infection.

Plasma donation saves lives, and donors' and blood/plasma facilities' communication is key to this. Smart apps are increasingly viewed as a crucial communication tool, and if they are created with the users' requirements and preferences in mind, plasma donation could make the best use of them.

1.3 Purpose: -

In our opinion we intend to create an application that is user-friendly for people who require plasma or who wish to donate plasma to anyone who is in need.

However, during design and development, areas of concern including privacy and secrecy should be taken into account. Age was found to be a contributing factor that might reduce donors' propensity to use apps. This system is used if anyone needs a Plasma Donor.

This system comprises of Admin and User where both can request for a Plasma.

- Both parties can Accept or Reject the request.
- The person who wants to donate his/her plasma needs to register in our application providing required information which are name, age, blood group, phone number, and location, etc.
- Patients who need plasma can also fill the form to request the plasma. Patients can directly call the donor by taking his/her contact number from the application.
- User can also search based on location they are living
- Just a single search allows anyone to reach maximum number of plasma donors in minimum possible time .

2. LITERATURE SURVEY

2.1 Existing Problem:

In most of the existing plasma donor application then system is closed for general plasma donation and mainly focused on COVID-19 patients forplasma donation, the android mobile user will not be able to insert or view details if the server goes down and a disadvantage of single point of failure. Most of the user details remains unverified and its difficult to track the fake users. The user interface of the application is not being user friendly and the user must have a device with android operating system with an active internet connection to interact with this application.

2.2 References: -

YEAR	TITLE	AUTHOR(s)	TECHNIQUE(s)	PROS	CONS
2022	Instant Plasma Donor Recipient connector web application	Kalpana Devi Guntoju, Tejaswini Jalli, Sreeja Uppala, Sanjay Mallisetti	Web Technologies, API, Database	The Donor needs to upload their recovered COVID-19 Certificate and it required toverified by the blood bank. It is a user-friendly application. It will help people to find plasma easily.	This is system is closed for general plasma donation and mainly focused on COVID-19 patients forplasma donation
2021	BDoor App-Blood Donation Application using Android Studio	S Periyanayagi, A Manikandan,M Muthukrishnan,and M Ramakrishnan	Android, FlutterUI, Dart, Firebase, Decision tree algorithm	The Donor details are verified before they allow to donate and have to authorised by institution. The Verification and validation are done in Email base.	The android mobile user will not be able to insert or view details if the server goes down. Thus, there is disadvantage of single point failure.

2020	Lifesaver E-Blood Donation App Using Cloud	Rishab Chakrabarti, Asha Darade, Neha Jadhav, Prof. S. M. Chitalkar	E-health, GPS, Blood bank database, Cloud Computing	Reduction in the errors ofblood bank using most eligible donor method. Direct Communication Between donor and the person in need of blood During the Emergency situation.	The user given detailsare maintained unverified.
2020	Developing a plasma donor application using Function-as-a-service in AWS	Aishwarya R Gowri	Serverless, aws, plasma theory, covid19, dynamoDB, cloud	The efficient way of findingplasma donor for the infected people. Aws lambda function is used and to deploy the application AWS EC2 service is used.	The user interface canbe better than now.
2019	D'WORLD: Blood Donation App Using Android	A. Meiyappan, K. Loga Vignesh, R. Prasanna, T. Sakthivel	Android, Global Positioning System (GPS), Mobile Computing	When the giver gives the blood, it will naturally evacuate the contributor detail for next three months.It additionally confirms with the Department of Health and Welfare to guarantee the benefactor medical case history.	The user must have an device with android operating system with an active internet connection to interact with this application.
2018	Automated blood bank system using Raspberry PI	Ashlesha C. Adsul, V. K. Bhosale, R. M. Autee	Raspberry Pi, Embedded Blood Bank, GSM, Android	When there is urgent need for blood then If this model is adopted the caller is immediately connected to the donor	Tackling the fakeusers.

2.3 Problem Statement Definition: -

Plasma donation saves lives, and the communication between blood/plasma centres and donors plays a vital role in this. Smart apps are now considered an important communication tool, and could be best utilized in plasma donation if they are designed to fit the users' needs and preferences. We plan to make a User-friendly application for users who are in need for plasma or who wish to donate plasma to anyone who are in need. However, areas of concern, including privacy and confidentiality, should be considered during design and development. Age was identified as a contributing factor that might decrease the likelihood of app usage among donors. The donation centre staff focused on the educational features of the app and emphasized the importance of the app providing statistics and sending notifications and reminders to donors.

3 IDEATION & PROPOSED SOLUTION

3.1. Empathy Map Canvas:-

3.2 Ideation & Brainstroming:-

Plasma is used for the treatment of serious health problems. This is why there are blood drives asking people to donate blood, plasma . Plasma is utilized to treat different irresistible sicknesses and it is one of the most established strategies known as plasma treatment. During Coronavirus emergency the necessity for plasma expanded radically as there were no immunization found to treat the contaminated patients, with plasma therapy the recovery rates where high but the donor count was very low and in such situations it was very important to get the information about the plasma donors. Saving the contributor data and telling about the ongoing givers would be some assistance as it can save time and assist the clients with finding the vital data about the contributors.

3.3 Proposed Solution:-

This proposed system aims at connecting the donors & the patients by an online application. By using this application, the users can either raise a request for plasma donation or requirements.

The basic solution is to create a centralized system to keep a track on the upcoming as well as past Plasma Donation Events. The recommendation solution is as follows:

Application contains two roles:

♣ Admin

♣ User

User:

- If the user wants to donate or receive they have to register with their personal details.
- After successful registration of user.
- A successful registration email is send to the user.
- After successful registration user will be directed to home page.
- They will be asked to press whether they will be donor or receiver.
- If the user is donor then he/she will fill the donation interest form which includes their Name, blood group details, location, last time donated date , phone number, email id.
- After filling the donation form he/she will redirected to page in which he/she can download the ecertificate.
- If the user is receiver then he/she can see the list of donors available and they can raise their request and contact donor directly.

Admin:

- Admin can login using their credentials.
- Admin can edit the request.
- Admin can delete the request.
- Admin can add volunteers.

3.4 Problem Solution Fit:-

Uniqueness:-

A User Interface is simple for users to understand. We can use the application anywhere anytime. The user immediately need the plasma for their treatment but the plasma is not available in nearby hospitals, then user can use this application to raise request and directly contact the donor , request them to donate the plasma. Hospitals can also raise request donors for donation. Somebody wants to donate blood and plasma but they don't know the way to donate then they use this application which will simple to use and it will save lives of many people. Today many of them have mobile phones they can install this application and use it to save the lives of people.

Social Impact / Customer Satisfaction:-

We are living in a modern world and everything can be accessed online. Even though there are many application there is no proper application for plasma donation . Many of them wish to donate blood and plasma but they are unaware about donation and how they can donate. This application provides opportunity to those who want to donate plasma. Donation of plasma are happening in many places many of them come forward to donate but it is not available at right time for use. Sometimes there is a shortage of plasma of particular type. Additional facilities that we need is to access the patients information quickly before plasma transfusion. To solve this issue software applications are employed with Cloud computing and Internet of Things tool which enable features such as information retrieval and continuous data tracking with analytics. This application avoids circulating of wrong information. A single platform for maintaining genuine information and increase the trust of participants involved int his activity. It increases the number of donors.

Business Model (Revenue Model):-

This application is accessible by everyone . It is free. Because of the trouble in finding givers who match a specific blood bunch, this application empowers clients to enlist individuals who wish to give plasma and keep their data in a data set. Nowadays the need for plasma increases. Anyone with basic knowledge can access this app. This can be used anywhere anytime. working with the government we can utilize an application to help those needing plasma.

Scalability of the Solution :-

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

4. 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 Website
FR-2	User Confirmation	Confirmation via Email
FR-3	User Login	Login using Registered email Id
FR-4	Sent Request	If plasma is required, the receiver will contact the donor
FR-5	Contact Donor	Contact the donor directly if a phone number is given
FR-6	View donation camps	View the list of donation camps happening nearby

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 donorsystem must be well-designed and welcoming.

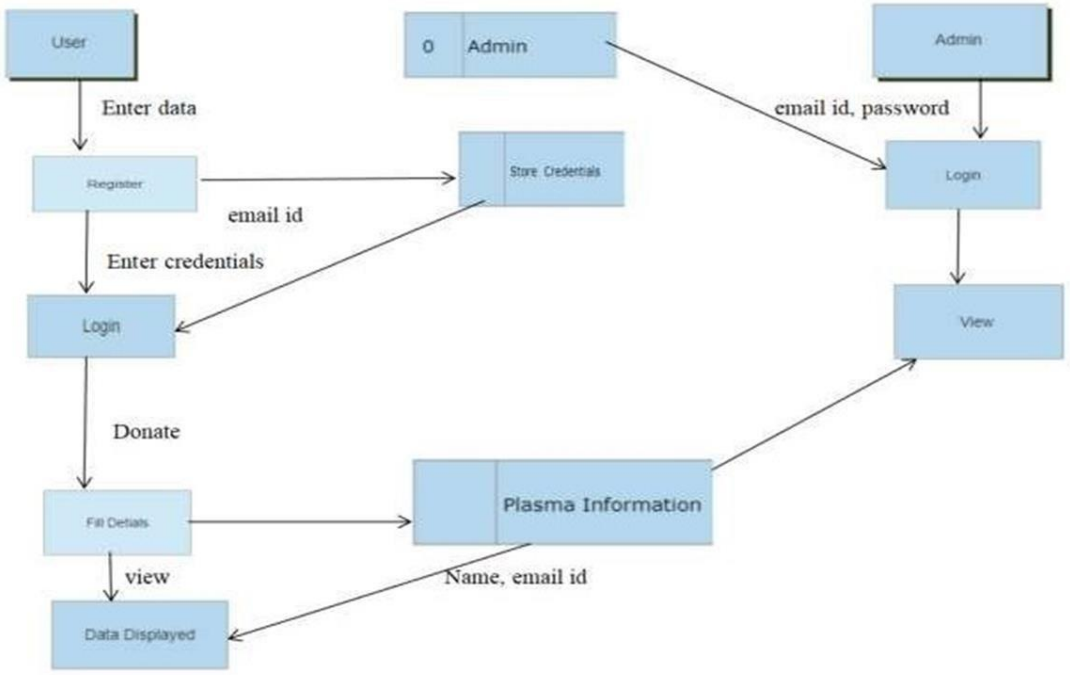
NFR-2	**Security**	Data storage is required by security systems, just like it is by many other applications. Databases are able to keep all the donor information that is viewed by applications. It must be secured with email Id and password.
NFR-3	**Reliability**	The system has the ability to work all the times without failures apart from network failure. A donor can have the faith on the system. The authorities will keeps the privacy of all donors in a proper manner
NFR-4	**Performance**	The Plasma donor System must perform well in different scenarios. The system is interactive and delays involved are less.
NFR-5	**Availability**	The system, including the onlinecomponents, should be available 24/7.
NFR-6	**Scalability**	The system offers the proper resources forissue solutions and is designed to protect sensitive information during all phases of operation.

5. PROJECT DESIGN

5.1 Data Flow Diagrams: -

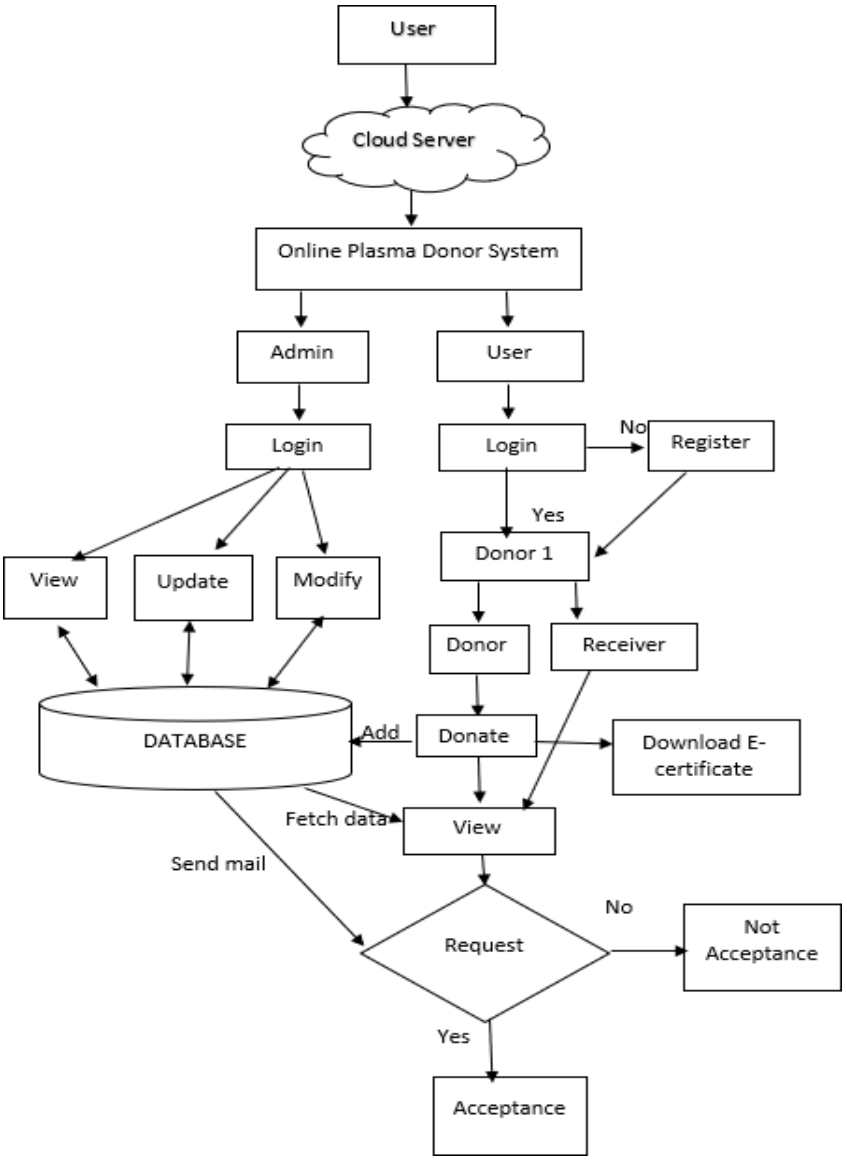
Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFDcan 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.

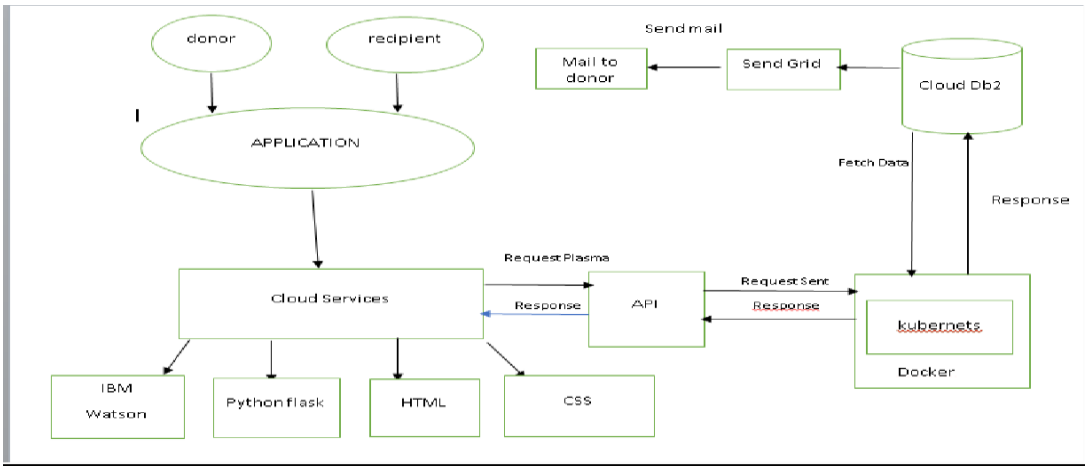


5.2 Solution & Technical Architecture: -

Solution Architecture: -



Technical Architecture:-



5.3 User Stories: -

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.	I can access my account dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmationemail once I have registered for the application	I canreceive successful message	High	Sprint-1
	Login	USN-3	As a user, I can log into the application by entering email &password	I can access into myProfile and view my dashboard	High	Sprint-1
	Dashboard	USN-4	As a user, I can login using my credentials and it will direct it to my dashboard	I can view and access what are the features are provided in dashboard	High	Sprint-1
Customer (Webuser)		USN-5	As a user, I can login using my credentials and it will direct itto my dashboard	I can view and access what are the features are provided in dashboard	High	Sprint -1
CustomerCare Executive	Query	USN-6	As a user had an any query about the given requirements	I can view a query and rectify the given query	Medium	Sprint-2

Administrator	Login	USN-7	As a admin ,have credentials using that they can login	They can view and modify the data in database	Medium	Sprint-2
	View	USN-8	As a admin I can view plasma information	View and modify	High	Sprint-1
	Modify	USN-9	As a admin I can modify the plasma information.	Modify only if there is a false information/	Medium	Sprint-1

6. PROJECT PLANNING AND SCHEDULING

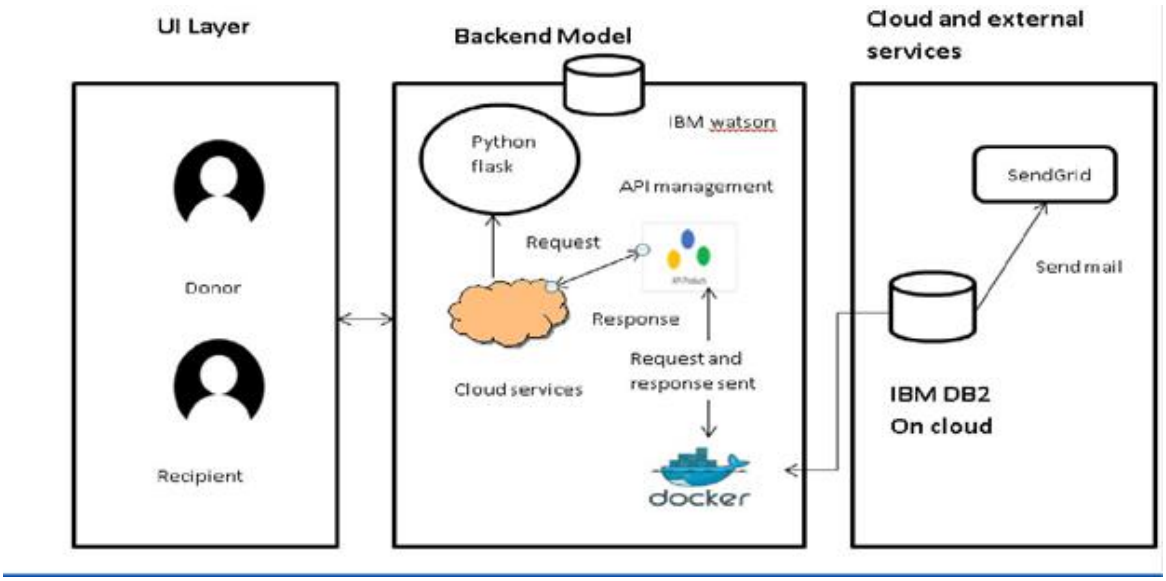
6.1 Sprint Planning

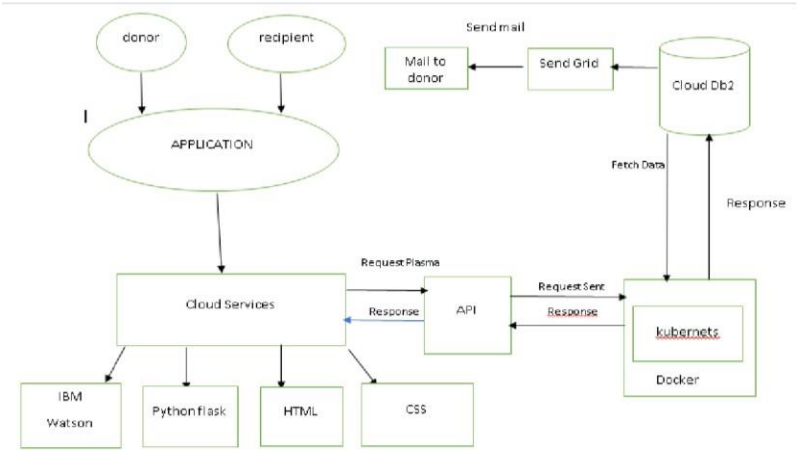
Sprints are the backbone of any good Agile development team. And the better prepared you are before a sprint, the more likely you are to hit your goals. Spring planning helps to refocus attention, minimize surprises, and (hopefully) guarantee better code gets shipped. The main event during agile methodology is the sprint, the stage where ideas turn into innovation and valuable products come to life. On one hand, agile sprints can be highly effective and collaborative. At the same time, they can be chaotic and inefficient if they lack proper planning and guidance. And for this reason, making a sprint schedule is one of the most important things you can do to ensure that your efforts are successful.

We categorized the sprint as 4 phases for creating the application

i. Sprint 1 is about creating the login page and the register page.
ii. Sprint 2 is about sending the confirmation mail to the users during registration.
iii. Sprint 3 is about as a user, can log into application by entering email and password.
iv. Sprint 4 is about as user, can register and make request for plasma donation via portal.

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2





Guidelines:

1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third party API's etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)

Table-1 : Components & Technologies:

S.NO	Component	Description	Technology
1.	User Interface	The user register and login. See the UI.	HTML, CSS, Python Flask
2.	Data maintenance	Store , maintain ,retrieve the user's details.	MYSQL
3.	Chatbot	Clarify user queries.	IBM Watson service
4.	Confirmation Email	Sending the confirmation email to users they have registered successfully.	SendGrid

5.	Cloud Database	Cloud database to store plasma information and View Plasma information.	IBM DB2
6.	File Storage	File storage requirements	IBM Block Storage
7.	Infrastructure (Server / Cloud)	To deploy the application on Local System	Kubernetes

Table-2: Application Characteristics:

S.NO	Characteristics	Description	Technology
1.	Open-Source Frameworks	Python Flask frameworks is used.	Python Flask
2.	Security Implementations	Mandatory Control(MAC) and kubernetes is used.	SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	3-Tier Architecture is used.	Web server-HTML,CSS Application Server-Python Flask Database Server-IBM DB2
4.	Availability	Using Load Balancer to distribute network traffic across Servers.	IBM Load Balancer
5.	Performance	User Friendly UI. Request and Response is faster.	IBM Content Delivery Network

6.2 Sprint Estimation and Delivery Schedule:

A sprint estimation shows how much effort a series of tasks require. It's based on assumptions, requirements, and dependencies of a project.

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	DHANUSH MOHAN KESAVARAJ
Sprint-1	Registration	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	DHANUSH JAYA SURYA

Sprint-2	Database	USN-3	Join the application to IBM db-2	2	Low	KESAVARAJ JAYA SURAYA
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password	1	High	KESAVARAJ JAYA SURAYA
Sprint-2	Dashboard	USN-4	As a user, I can register and make request for plasma donation.	2	High	KESAVARAJ MOHAN

Project Tracker, Velocity & Burndown Chart

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	Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	7 Nov 2022	Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	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)

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

7. CODING AND SOLUTION

Dashboard.html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <title>IBM Plasma Donar App</title>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

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

  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></
script>

  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>

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

</head>
```

```
<style>
    .big{
    top:70;
    background-color:white;
    margin-top:80px;
    margin-left:550px;
    margin-right:550px;
    height:200px;
    border-radius: 25px;
    border: 3px solid #4a77d4;
    box-shadow: 6px 8px 4px grey;
    text-align:center;
    }
    .row{

    height:150px;

    }
```

```
.col{
        margin:10px;
        margin-left:50px;
        margin-right:50px;
        border-radius: 25px;
        border: 1px solid #4a77d4;
        box-shadow: 0px 8px 4px grey;
        text-align:center;
}
.ext{
margin-top:25px;
line-height:40px;
}
.ext1{
margin-top:40px;
line-height:50px;
font-size:25px;
color:#f95450;
```

```
    }


</style>

<body>


<div class="container-fluid">

<div class="header">

<div><b>Plasma Donar App</b></div>

<ul>

    <li><a href="/requester">Request</a></li>

    <li><a class="active" href="/logout">Logout</a></li>



</ul>

</div>

  <br>

  <div class="big">

    <div class="box">

     <div class="ext1"><font
size="20px">{{b['1']}}</font><br><b>Donors</b></div>
```

```
</div>
  </div>
  <br>
  <div class="row">
    <div class="col" >
     <div class="ext">{{b['2']}}<br><b>O Positive</b></div>
</div>
    <div class="col" >
     <div class="ext">{{b['3']}}<br><b>A Positive</b></div>
</div>
    <div class="col" >
     <div class="ext">{{b['4']}}<br><b>B Positive</b></div>
</div>
    <div class="col" >
     <div class="ext">{{b['5']}}<br><b>AB Positive</b></div>
</div>
  </div>
  <br>
```

```
  <div class="row">
    <div class="col" >
     <div class="ext">{{b['6']}}<br><b>O Negative</b></div>
</div>
    <div class="col" >
     <div class="ext">{{b['7']}}<br><b>A Negative</b></div>
</div>
    <div class="col" >
     <div class="ext">{{b['8']}}<br><b>B Negative</b></div>
</div>
    <div class="col" >
     <div class="ext">{{b['9']}}<br><b>AB Negative</b></div>
</div>
  </div>
  <div style="height:200px"></div>
</div>
</body>
</html>
```

Login.html

```
<!DOCTYPE html>

<html >

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

<head>

  <meta charset="UTF-8">

  <title>IBM Donor App</title>

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

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

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

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

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



<style>

.login{
```

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

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

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

```
    <!-- Main Input For Receiving Query to our ML -->



   <form action="{{ url_for('loginpage')}}"method="post">

     <input type="text" name="username" placeholder="Enter UserName" required="required" style="color:black" />

       <input type="password" name="password" placeholder="Enter Password" required="required" style="color:black" />

       <button type="submit" class="btn btn-primary btn-block btn-large">Login</button>



   </form>

<br><br>

<div style="color:black">

 {{ msg }}</div>

 </div>



</body>

</html>
```

Register.html

```
<!DOCTYPE html>

<html >

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

<head>

  <meta charset="UTF-8">

  <title>IBM Plasma Donor App</title>

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

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

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

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

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



<style>

.login{

top: 20%;

}
```

```
</style>

</head>



<body>

<div class="header">

<div>Plasma Donor App</div>

<ul>



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

</ul>

</div>

 <div class="login">



    <!-- Main Input For Receiving Query to our ML -->

   <form action="{{ url_for('register')}}"method="post">

    <input type="text" name="username" placeholder="Enter Your Name" required="required" style="color:black"/>

      <input type="email" name="email" placeholder="Enter Email" required="required" style="color:black"/>
```

```
    <input type="text" name="phone" placeholder="Enter 10-digit mobile number" required="required" style="color:black"/>

      <input type="city" name="city" placeholder="Enter Your City Name" required="required" style="color:black"/>

    <select name="infect">



                        <option value="select" selected>Select COVID infection status</option>

                        <option value="infected">Infected</option>

                        <option value="uninfected">Uninfected</option>

    </select>

      <select name="blood">

                        <option value="select" selected>Choose your blood group</option>

                        <option value="O Positive">O Positive</option>

                        <option value="A Positive">A Positive</option>

                        <option value="B Positive">B Positive</option>

                        <option value="AB Positive">AB Positive</option>

                        <option value="O Negative">O Negative</option>

                        <option value="A Negative">A Negative</option>

                        <option value="B Negative">B Negative</option>
```

```
                        <option value="AB Negative">AB Negative</option>

    </select>

      <input type="password" name="password" placeholder="Enter Password" required="required" style="color:black"/>

      <button type="submit" class="btn btn-primary btn-block btn-large">Register</button>



   </form>



 <br><br>

<div style="color:black">

 {{ msg }}</div>

 </div>




</body>

</html>
```

Request.html

```
<!DOCTYPE html>

<html >

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

<head>

  <meta charset="UTF-8">

  <title>IBM Plasma Donor App</title>

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

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

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

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

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



<style>

.login{

top: 20%;

}
```

```
</style>

</head>



<body>

<div class="header">

<div>Plasma Donor App</div>

<ul>



    <li><a href="/requester">Request</a></li>

    <li><a href="/registration">Register</a></li>

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

</ul>

</div>

 <div class="login">

    <div>

    </div>



    <!-- Main Input For Receiving Query to our ML -->
```

```
<form action="{{ url_for('requested')}}"method="post">

<input type="text" name="name" placeholder="Enter Name"
required="required" style="color:black" />

<input type="email" name="email" placeholder="Enter Email"
required="required" style="color:black"/>

<input type="text" name="phone" placeholder="Enter 10-digit mobile number"
required="required" style="color:black"/>

<select name="bloodgrp">

<option value="select" selected>Choose your blood
group</option>

<option value="O Positive">O Positive</option>

<option value="A Positive">A Positive</option>

<option value="B Positive">B Positive</option>

<option value="AB Positive">AB Positive</option>

<option value="O Negative">O Negative</option>

<option value="A Negative">A Negative</option>

<option value="B Negative">B Negative</option>

<option value="AB Negative">AB Negative</option>

</select>

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

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



    </form>



 <br><br>

<div style="color:black">

 {{ pred }}</div>



 </div>



</body>

</html>
```

Style.css

```
@import url(https://fonts.googleapis.com/css?family=Open+Sans);

.btn {

display: inline-block;
```

*display: inline;

*zoom: 1; padding:

4px 10px 4px;

margin-bottom: 0;

font-size: 13px;

line-height: 18px;

color: #333333;

text-align: center;

text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75);

vertical-align: middle;

background-color: #d70c0c;

background-image: -moz-linear-gradient(top, #ffffff, #e6e6e6);

background-image: -ms-linear-gradient(top, #ffffff, #e6e6e6);

background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#ffffff), to(#e6e6e6));

background-image: -webkit-linear-gradient(top, #ffffff, #e6e6e6);

background-image: -o-linear-gradient(top, #ffffff, #e6e6e6);

background-image: linear-gradient(top, #ffffff, #e6e6e6);

background-repeat: repeat-x;

filter: progid:dximagetransform.microsoft.gradient(startColorstr=#ffffff, endColorstr=#e6e6e6, GradientType=0);

border-color: #e6e6e6 #e6e6e6 #e6e6e6;

border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25);

border: 1px solid #e6e6e6;

-webkit-border-radius: 4px;

-moz-border-radius: 4px;

border-radius: 4px;

-webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);

-moz-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);

box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);

cursor: pointer; *margin-left: .3em;

}

.btn:hover, .btn:active, .btn.active, .btn.disabled, .btn[disabled] { background-color: #e6e6e6; }

.btn-large {

padding: 9px 14px;

```
font-size: 15px;

line-height: normal;

-webkit-border-radius: 5px;

-moz-border-radius: 5px;

border-radius: 5px;

}



.btn:hover {

color: #333333;

text-decoration: none;

background-color: #e6e6e6;

background-position: 0 -15px;

-webkit-transition: background-position 0.1s linear;

-moz-transition: background-position 0.1s linear;

-ms-transition: background-position 0.1s linear;

-o-transition: background-position 0.1s linear;

transition: background-position 0.1s linear;

}
```

```
.btn-primary, .btn-primary:hover {

text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25);

color: #ffffff;

}



.btn-primary.active { color: rgba(255, 255, 255, 0.75); }



.btn-primary {

background-color: #d70c0c;

background-image: -moz-linear-gradient(top, #6eb6de, #4a77d4);

background-image: -ms-linear-gradient(top, #6eb6de, #4a77d4);

background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#6eb6de),
to(#4a77d4));

background-image: -webkit-linear-gradient(top, #6eb6de, #4a77d4);

background-image: -o-linear-gradient(top, #6eb6de, #4a77d4);

background-image: linear-gradient(top, #6eb6de, #4a77d4);

background-repeat: repeat-x;

filter: progid:dximagetransform.microsoft.gradient(startColorstr=#6eb6de,
endColorstr=#4a77d4, GradientType=0);
```

```
border: 1px solid #3762bc;

text-shadow: 1px 1px 1px rgba(0,0,0,0.4);

box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.5);

}



.btn-primary:hover, .btn-primary:active, .btn-primary.active, .btn-primary.disabled,
.btn-primary[disabled] {

filter: none;

background-color: #d70c0c

}



.btn-block { width: 100%; display:block; }



* { -webkit-box-sizing:border-box; -moz-box-sizing:border-box; -ms-box-
sizing:border-box; -o-box-sizing:border-box; box-sizing:border-box; }



html { width: 100%; height:100%; overflow:hidden; }



body {

width: 100%;
```

```
height:100%;

font-family: 'Open Sans', sans-serif;

color: #000000;

font-size: 18px;

text-align:center;

letter-spacing:1.2px;




}

.header {

        top:0;

        margin:0px;

        left: 0px;

        right: 0px;

        position: fixed;

        background: #d44a4a;

        color: black;

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

```
            overflow: hidden;
            padding: 15px;
            font-size: 1.5vw;
            width: 100%;
            text-align: center;
    }
.login {
position: absolute;
top: 70%;
left: 50%;
margin: -25px 0 0 -150px;
width:400px;
height:400px;
}

.header div { color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-spacing:1px;
text-align:center; float:left; padding-left:150px;}

ul {
```

```
  list-style-type: none;
  margin: 0;
  padding: 0;
  padding-right:150px;
  overflow: hidden;
 }

li {
  float: right;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 0px 15px;
  text-decoration: none;
}
```

```
input {

width: 100%;

margin-bottom: 10px;

background: rgba(255,255,255,255);

border: none;

outline: none;

padding: 10px;

font-size: 13px;

color: black;

text-shadow: black;

border: 1px solid rgba(0,0,0,0.3);

border-radius: 4px;

box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px
rgba(255,255,255,0.2);

-webkit-transition: box-shadow .5s ease;

-moz-transition: box-shadow .5s ease;

-o-transition: box-shadow .5s ease;

-ms-transition: box-shadow .5s ease;

transition: box-shadow .5s ease;
```

```
}

input:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4), 0 1px 1px rgba(255,255,255,0.2); }



textarea {

width: 100%;

margin-bottom: 10px;

background: rgba(255,255,255,255);

border: none;

outline: none;

padding: 10px;

font-size: 13px;

color: black;

text-shadow: black;

border: 1px solid rgba(0,0,0,0.3);

border-radius: 4px;

box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);

-webkit-transition: box-shadow .5s ease;

-moz-transition: box-shadow .5s ease;
```

```
-o-transition: box-shadow .5s ease;

-ms-transition: box-shadow .5s ease;

transition: box-shadow .5s ease;

}

textarea:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4), 0 1px 1px rgba(255,255,255,0.2); }



select {

width: 100%;

margin-bottom: 10px;

background: rgba(255,255,255,255);

border: none;

outline: none;

padding: 10px;

font-size: 13px;

color: #000000;

text-shadow: 1px 1px 1px rgba(0,0,0,0.3);

border: 1px solid rgba(0,0,0,0.3);

border-radius: 4px;
```

box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);

-webkit-transition: box-shadow .5s ease;

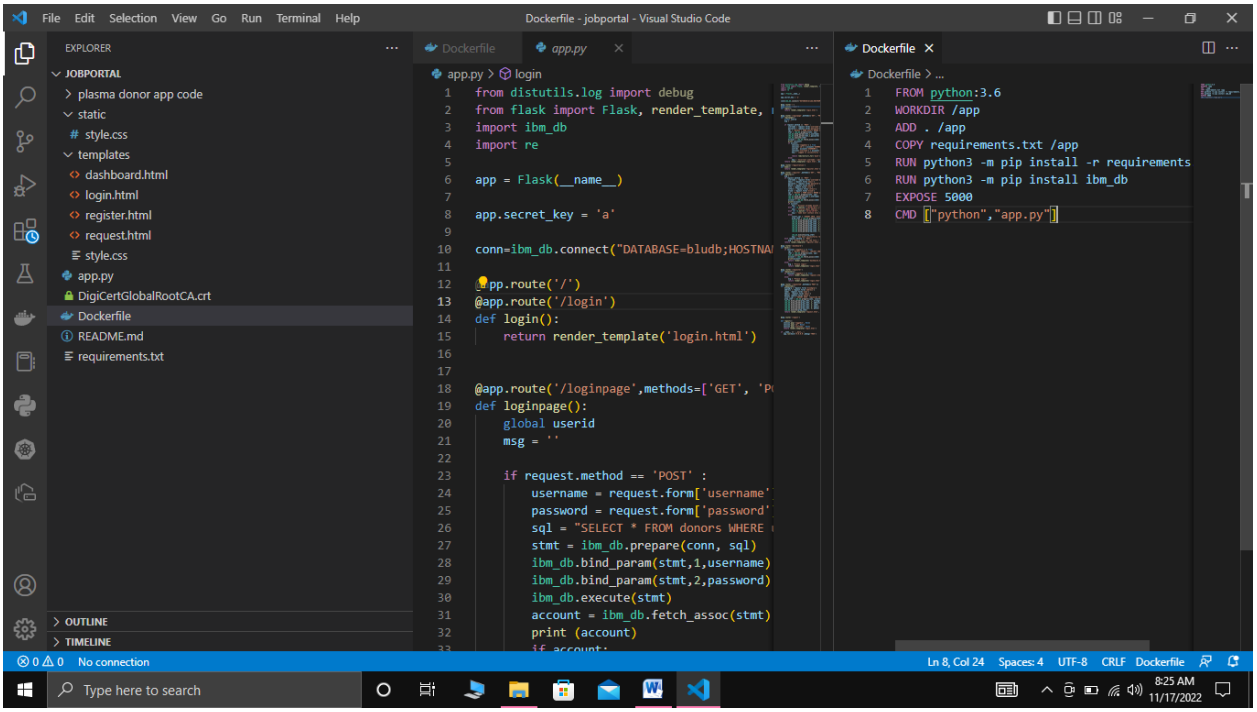
-moz-transition: box-shadow .5s ease;

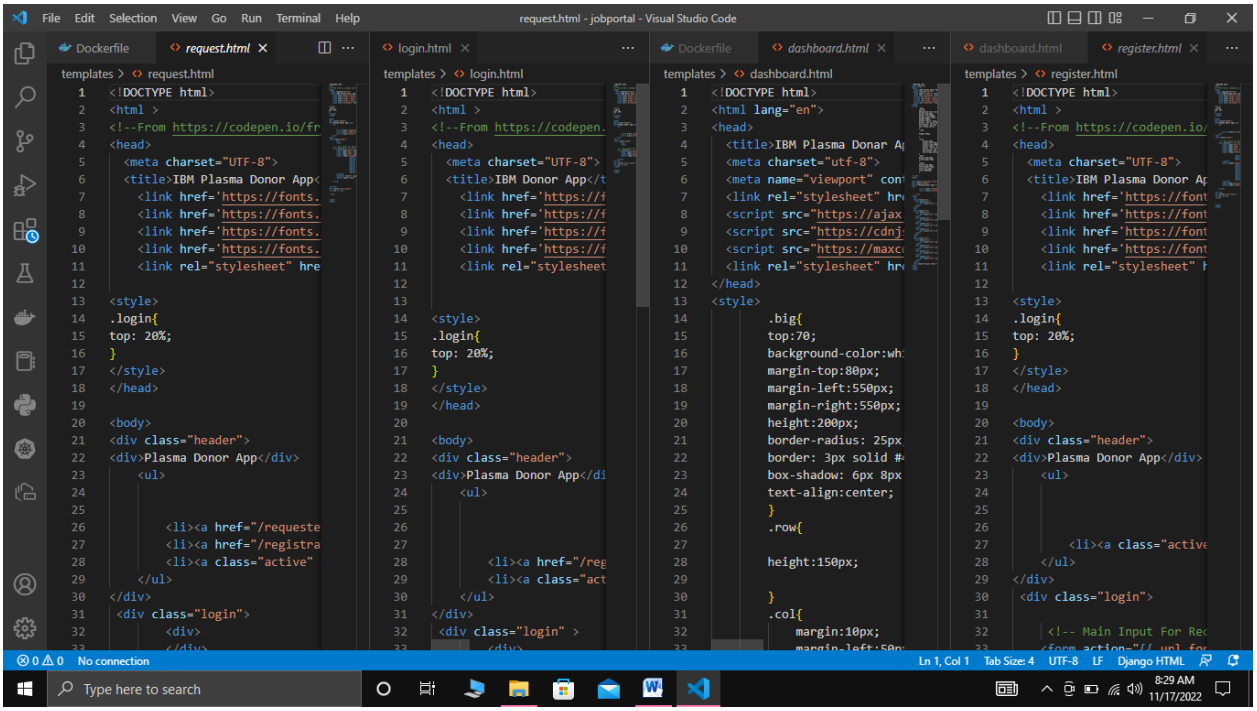
-o-transition: box-shadow .5s ease;

-ms-transition: box-shadow .5s ease;

transition: box-shadow .5s ease;

}





7.1 SendGrid

SendGrid is a cloud-based SMTP provider that allows you to send email without having to maintain email servers. SendGrid manages all of the technical details, from scaling the infrastructure to ISP outreach and reputation monitoring to whitelist services and real time analytics.

SendGrid provides two ways to send email: through our SMTP relay or through our Web API. SendGrid provides client libraries in many languages. This is the preferred way to integrate with SendGrid. If you choose to use SendGrid without a client library, the Web API is recommended in most cases as it is faster, provides some beneft with encoding, and tends to be easier to use. SMTP provides many features by default, but is harder to setup.

Web API

- The Web API has some advantages over SMTP:
- If your ISP blocks all outbound mail ports and your only option is HTTP.
- If there is high latency between your site and ours, the Web API might

- be quicker since it does not require as many messages between the client and server.
- If you do not control the application environment and cannot install and configure an SMTP library.
- If you build a library to send email, developing against a web API provides quicker development.

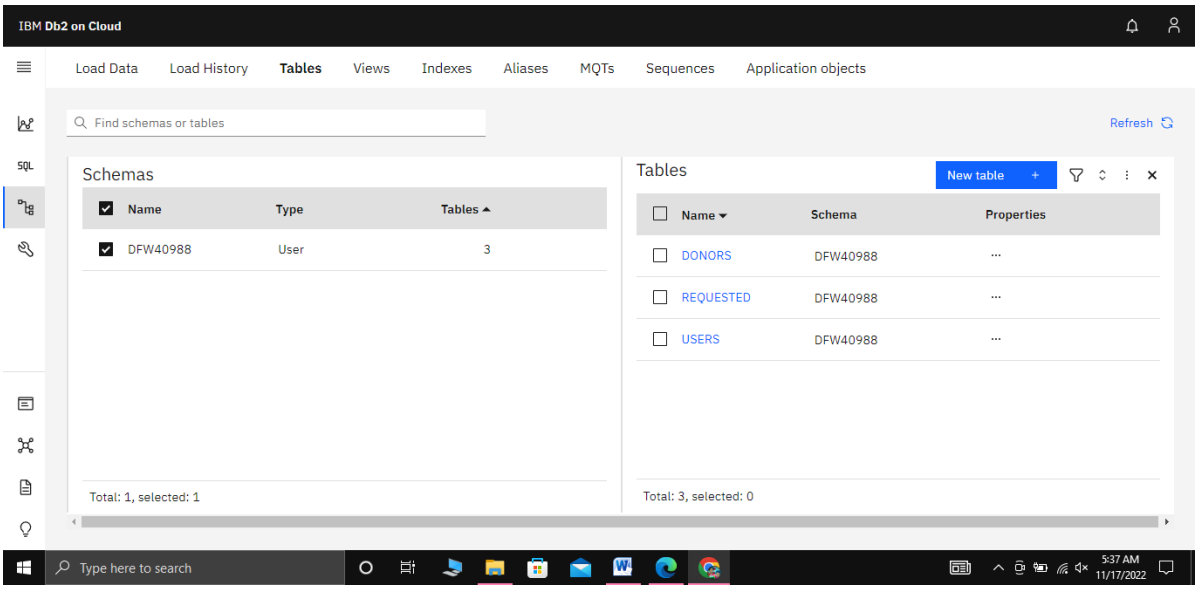
SMTP Relay

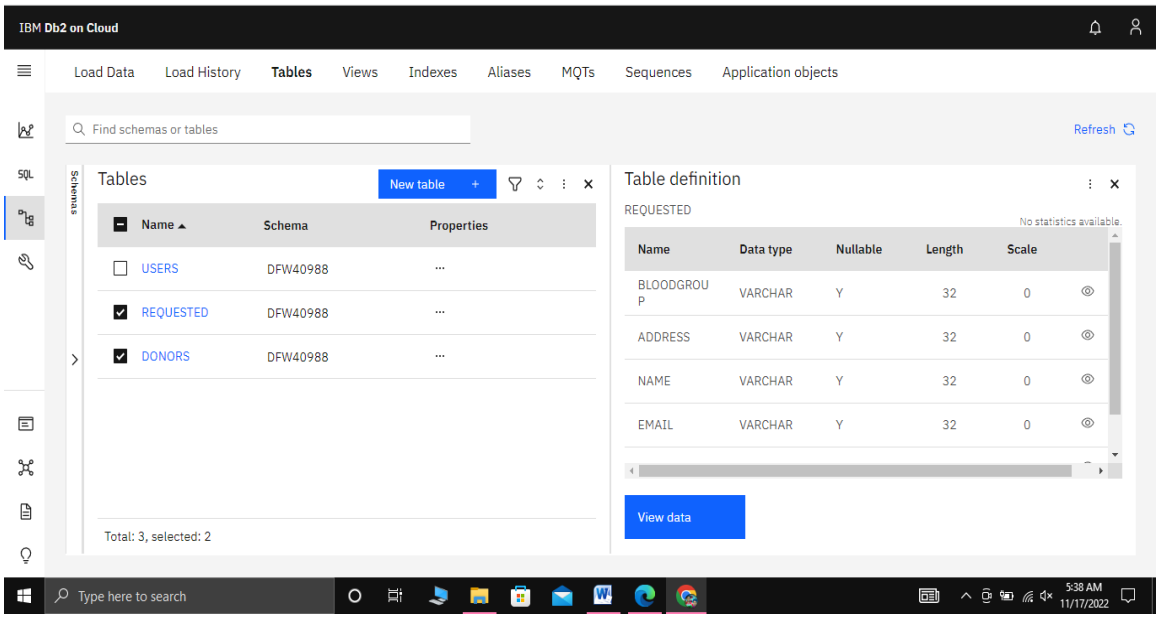
- If you are integrating SendGrid with an existing application, setting up the application to use our SMTP relay is easiest, as it only requires modifying SMTP configuration.
- Change your SMTP username and password to your SendGrid credentials.
- Set the server host name to smtp.sendgrid.net
- Use ports 25 or 587 for plain/TLS connections and port 465 for SSL connections.

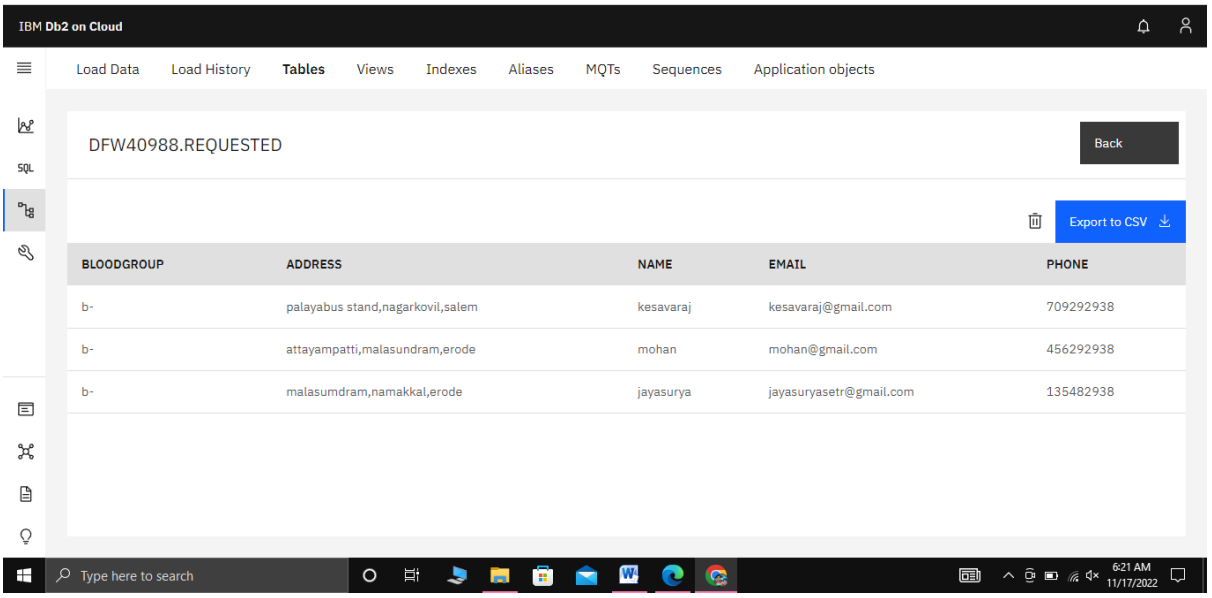
Code:

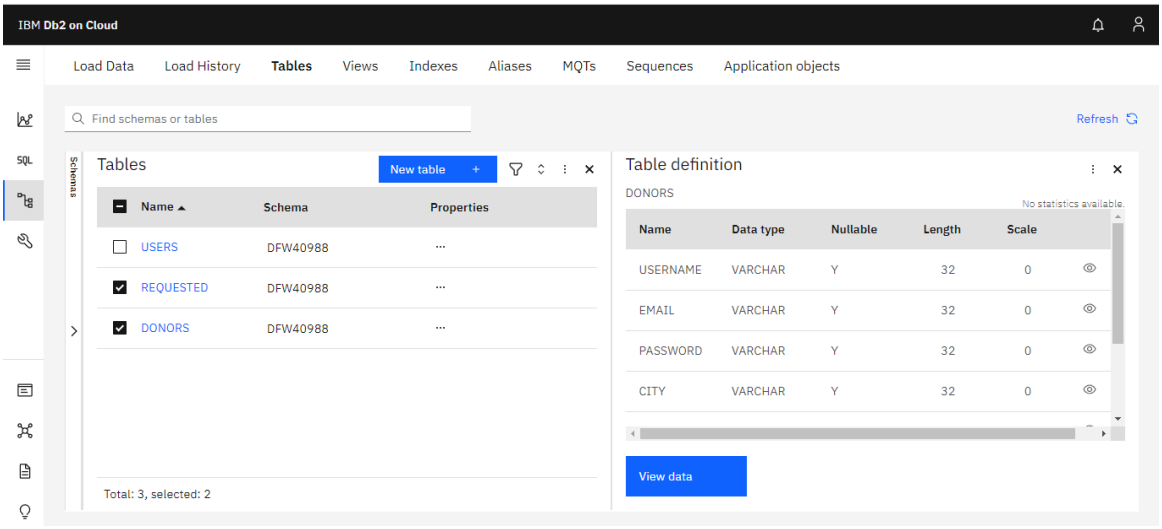
```
import os
from dotenv import
load_dotenv load_dotenv()

from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail
def sendmail(usermail,subject,content):

    message =
Mail(from_email='dhanush@student.tce.edu',to_emails=usermail,subject=subj
ect,html_content='<strong> {} </strong>'.format(content))
    try:
        sg = SendGridAPIClient(os.getenv('SENDGRID_API_KEY'))
        response = sg.send(message)

        print(response.status_code
        ) print(response.body)
        print(response.headers)

    except Exception as e:
        print(e.message)
```

7.2 Database Schema






IBM Db2 on Cloud
Load Data
Load History
Tables
Views
Indexes
Aliases
MQTs
Sequences
Application objects
DFW40988.REQUESTED
Back
Export to CSV
BLOODGROUP
ADDRESS
NAME
EMAIL
PHONE
b-
palayabus stand,nagarkovil,salem
kesavaraj
kesavaraj@gmail.com
709292938
b-
attayampatti,malasundram,erode
mohan
mohan@gmail.com
456292938
b-
malasumdram,namakkal,erode
jayasurya
jayasuryasetr@gmail.com
135482938
Type here to search
6:21 AM
11/17/2022


IBM Db2 on Cloud
Load Data
Load History
Tables
Views
Indexes
Aliases
MQTs
Sequences
Application objects
Find schemas or tables
Refresh
Schemas
Tables
New table
Name
Schema
Properties
USERS
DFW40988
REQUESTED
DFW40988
DONORS
DFW40988
Total: 3, selected: 2
Table definition
DONORS
No statistics available.
Name
Data type
Nullable
Length
Scale
USERNAME
VARCHAR
Y
32
0
EMAIL
VARCHAR
Y
32
0
PASSWORD
VARCHAR
Y
32
0
CITY
VARCHAR
Y
32
0
View data

8. RESULT

8.1 Authentication Module

- Sign Up

New user or donor can create an account to use in the blood/plasma donor application and create a password for account verification and create an identity.

- Sign In

Donor Sign In to the account for viewing or editing location details and any other personal information.

- Account Verification

If donor changes their password or if they forget the password then we have to verify their account using mail verification.

8.2 Service Provider Module

- Add New Donor

User can be able to register to add donor details.

- List All Donor

User can be able to view all Donor who all use our Plasma Donor Application.

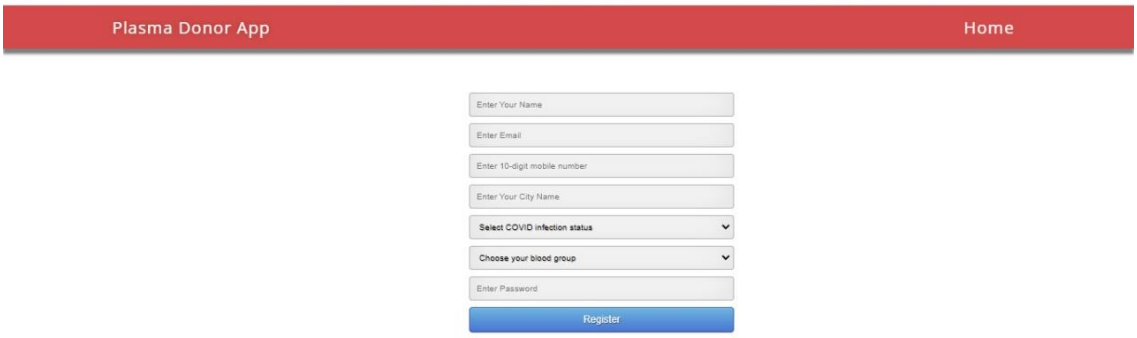
- Edit Customer Plan Details

User can be able to edit the existing Donor details as the Donor wish.

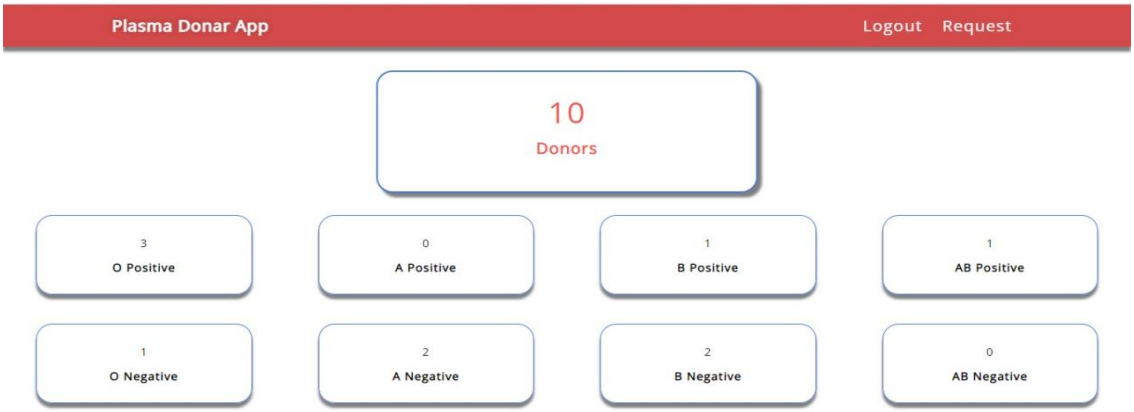
Screen Layouts

Login page

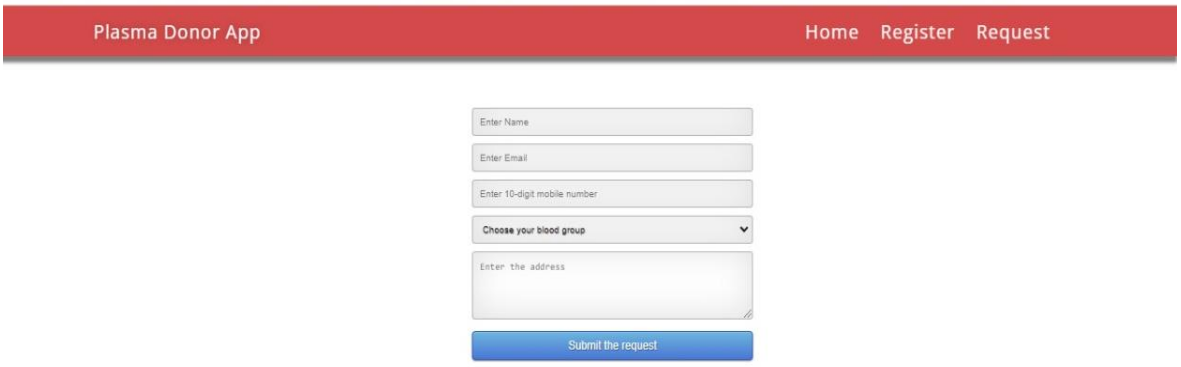
Register page



Home page



Request page



Registration page

The Donors can register their account using their email ID. Once registered, The Donor can sign-up by using his\her respective password. The login page for Plasma Donors is shown in the figure, which contains the E-mail and Password field. The profile of the Donor, where he/she needs to enter the required details. After registration Donor can maintain according to his availability. The registration page with Full Name, Email Address, Last donated date, Password, Contact Details, Blood Group, Location and all other details, which is illustrated. The details of the available donors can be displayed and viewed by other users.

9. ADVANTAGES & DISADVANTAGES

Advantages

- ***Speed***

  This website is fast and offers great accuracy as compared to manual registered keeping.

- ***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.

10.CONCLUSION

Although the government is carrying out Covid vaccination campaigns on a large scale, the number of vaccines produced is not enough for all the population to get vaccinated at present. And with the corona positive cases rising every day, saving lives has become the prime matter of concern. As per the data provided by WHO more than 3 million people have died due to the coronavirus. However, apart from vaccination, there is another scientific method by which a covid infected person can be treated and the death risk can be reduced. This plasma therapy is an experimental approach to treat corona-positive patients and help them recover. This plasma therapy is considered to be safe & promising. A person who has recovered from Covid can donate his/her plasma to a person who is infected with the coronavirus.

This system proposed here aims at connecting the donors & the patients by an online application. By using this application, the users can either raise a request for plasma donation or requirement. Both parties can Accept or Reject the request. User has to Upload a Covid Negative report to be able to Donate Plasma. This system is used if anyone needs a Plasma Donor Blood and Plasma donation is a kind of citizen's social responsibility in which an individual can willingly donate blood/plasma via our app. This Application has been created with the concept and has sought to make sure that the donor gives blood/plasma to community. This model is made user friendly so anybody can view and maintain his/her account. This application will break the chain of business through blood/plasma and help the poor to find donor at free of cost. This project will help new blood/plasma banks improve their services and progress from traditional to user-friendly frameworks.

11.FUTURE SCOPE

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

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

Donor and Beneficiary Stories feature aims to create a sense of belonging to the community. Donors will be able to view and share personal experiences about their donation; Beneficiaries can share their experiences of receiving blood transfusion which contributed to their improved health and lives.

Live Check-in Process feature aims to provide a better experience with regards to the waiting time when the user is in the process of donation. We hypothesise that a more efficient experience will help the user look forward to his blood/plasma donation appointments.

12.Referance

- ***GitHub and Source code Link - https://github.com/IBM-EPBL/IBM-Project-28198-1660108518***
- ***Demo video Link - https://drive.google.com/drive/folders/1NtTs3IZwKdQgQXkWAKqfiCXw0udGkiRT?usp=sharing***