PROJECT REPORT

PLASMA DONOR APPLICATION

TEAM ID - PNT2022TMID34946

SUBMITTED BY

AROCKIA JEBIN S	**962819205003**
ASWIN RAJA A	**962819205004**
VALGIN R V	**962819205036**
MOHAMED ASHIF ALI A	**962819205502**

CHAPTER 1

INTRODUCTION

1.1 Project Overview

The Blood Donation Agent is to create an E-Information about the donor and organization that are related to donating the blood. Through this application any person who is interested in donating the blood can register himself in the same way if any organization wants to register itself with this site that can also register. Moreover if any general consumer wants to make request blood online he can also take the help of this site. Admin is the main authority who can do addition deletion and modification if required.

1.2 Purpose

The proposed method helps the users to check the availability of donors. A donor has to register to the website providing their details. The registered users can get the information about the donor count of each blood group. The database will have all the details such as name, email, phone number, infected status. Whenever a user requests for a particular blood group then the concerned blood group donors will receive the notification regarding the requirement. A Json code is written to store the information, to fetch the requested information in lambda.

CHAPTER-2

LITERATURE SURVEY

The Various Methodologies that are used for the Plasma Donor Application are discussed as follows:

Pohandulkar & Khandelwal [1] (2018) considered a Blood bank application using raspberry pi, to bring near-by blood banks and the person who needs the blood due to accident or any emergency. The main aim of this work is to reduce the time span between the donor and recipient. By using Raspberry pi 2 and GSM modem SIM900A, all the databases from blood banks are collected and fetch the given data as per request from the recipient. The fetched blood donor data is sent to the recipient and also with addition an IP Address is attached to the message which allows the recipient to download an app and get all the information.

Vanitha & Divyarani [2] (2013) proposed an idea performing communication between the blood centre department and hospitals using Geo-location RVD Scoring Algorithm. Also makes it easy to organise the database including the contact details and blood groups of the donors..

Ankit Bhardwaj et al. [3] (1978) focused on data mining and the current trends associated with it. It presents an overview of the data mining system and clarifies how data mining and knowledge discovery in databases are related both to each other and to related fields. Data Mining is a technology used to describe knowledge discovery and to search for significant relationships such as patterns, association and changes among variables in databases. This enables users to search, collect and donate blood to the patients who are waiting for the last drop of the blood and are near death.

Catassi & Petersen [4] (1967) described a computerised inventory control system for controlling blood distribution between the blood bank and its client hospitals. Significantly, appropriately processed information of daily blood status by a centralised

computer has led to a 60 per cent reduction in out-dating together with an inventory reduction at the blood bank averaging 30 per cent. Incorporation of the new system into current hospital and blood bank management is evaluated in terms of psychological, labour, and economic implications.

Abd-Alsabour et al. [5] (2017) aimed to help citizens fulfil their needs for a safe and reliable blood group by searching for and locating a specific blood group. The author illustrates the problem of the blood bags shortage which is represented in the uncontrolled blood banks and parallel markets, lack of awareness and confidence, disappearance of the rare blood groups, and the difficulty in finding a specific blood group. Hence, the Blood Bag web-based application that is proposed by the author connected to a centralised database to gather and organise the data from all blood banks and blood donation campaigns. The proposed application organises and controls the whole critical processes related to blood donation, testing and storage of blood bags, and delivering it to the patient.

Seraphim et al. [6] (2022) proposed an idea to detect the problems in the whole process of blood donation. The key problems reported for not donating blood were that donating blood did not cross their mind (32.4%) followed up with 'No time available in schedule for donation which added up to around 45%. Finally, the main reason being the difficulty in accessing the blood donation centre which encompasses 61.3 %. This is due to unawareness in the society regarding the blood donation process. Sound data-driven machine learning techniques are used for predicting donations and supply needs which in turn can improve the entire supply chain.

Tyagi et al. [7] (2021) depicted a high level program to close the hole between blood givers and individuals needing blood. The Online Blood donation Administration Framework application is an approach to synchronise blood donation centres with emergency clinics with the assistance of the Web. It is a web application where enlisted clinics can check the accessibility of the necessary Blood and can send a blood solicitation to the closest blood donation centre or comparable contributor as per the blood and can be controlled online through where fundamental. Blood donation centres can likewise send a solicitation to another blood donation centre that isn't accessible.

Anybody willing to give blood can be found at the closest blood donation centre utilising the Android Bank The executives Framework. Blood donation centres can be followed utilising maps. The Android application is simply accessible to benefactors to look for blood gifts and ask blood donation centres and clinics to search out blood donation centres and close by givers.

Hegedus et al. [8] (2019) described to subserve and popularise blood donation, respectively to support an effective and permanently extant communication channel between the project's two target audience. One of the target audiences contains persons who are willing to donate blood. For them, the system provides a mobile application for logging and scheduling their donations, for receiving notifications and for accessing useful information related to the blood donation process. The other target audience includes employees working at blood donation centres. Using the web application, they are able to manage the informative data published through the mobile application, and they are able to send notifications about the current blood necessities.

Debnath et al. [9] (2018) described that an automated system could be built to monitor and organise blood donation camps. The author developed an integrated framework with all related but isolated web based subsystems of a blood management system. They propose a data warehouse (DW) as an integral part of the integrated framework to store historical blood donation data in a centralised database for analytical processing.

Patil and Ghuse et al. [10] (2015) collects and analyses the reactions that occur in the donor during blood donation at different hospitals and blood banks by the DonorHART tool. The tool also monitors and researches the risks involved for donors at the time of blood donation or after the donation process. Data mining makes an effort to reveal the patterns in data that are difficult to detect and recognize with automatic pattern recognition. It is analysing data and then summarising it into different forms which is called as information and is taken from different databases.

2.1 Existing Problem

In existing system, not all users can get access to the information because of the low working of the application or is not able to access any site. Sometimes the information is not updated or available for a particular place. In existing system the security is less and latest updates and uploads are not so frequent. It takes a long time or the result might come at an odd time. Since it is webpage it may not load faster or better due to bad network connectivity. It is more like a stationary technique.

2.2 References

[1] S. S. Pohandulkar and C. S. Khandelwal, "Blood Bank App using Raspberry PI," 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems(CTEMS),2018,pp. 355-358 , doi: 10.1109/CTEMS.2018.8769143.

[2] R.Vanitha, P.Divyarani, "BCloud App: Blood Donor Application for Android Mobile", International Journal of Innovations in Engineering and Technology (IJIET), Vol. 2 Issue 1 February 2013.

[3] Ankit Bhardwaj, Arvind Sharma, V.K.Shrivastava / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2 , Issue4, July-August 2012 , pp.1303-1309

[4] Catassi, Petersen, E. L."The Blood Inventory ControlSystem-Helping Blood Bank Management ThroughComputerized Inventory Control", Transfusion, 7:60 (1967).

[5] R.S. Ali, T.F.Hafez, A. B. Ali and N. Abd-Alsabour,"Blood bag: A web application to manage all blood donation and transfusion processes," 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET),2017, pp. 2125-2130, doi: 10.1109/WiSPNET.2017.8300136.

[6] P. Selvaraj, A.Sarin and B. I. Seraphim, "Blood Donation Prediction System Using Machine Learning Techniques," 2022 International Conference on Computer CommunicationandInformatics(ICCCI),2022,pp.14,doi:10.1109/ICCCI54379.2022.9740878.

[7] R. Kumar, R. Kumar and M. Tyagi, "Web Based Online Blood Donation System," 2021 3 rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), 2021 , pp. 1630-1632, doi: 10.1109/ICAC3N53548.2021.9725558.

[8] H. Hegedüs, K. Szász, K. Simon, T. Fazakas, A.Mihály and K.Nagy, "Blood Notes: Software System for Promoting and Facilitating Blood Donation," 2019 IEEE 17 th International Symposium on Intelligent Systems and Informatics (SISY), 2019 , pp. 77-82 , doi: 10.1109/SISY47553.2019.9111536.

[9] G.Maji,N.C.Debnath and S.Sen, "Data Warehouse Based Analysis with Integrated Blood Donation Management System," 2018 IEEE 16 th International Conference on Industrial Informatics (INDIN),2018 pp. 855-860 , doi: 10.1109/INDIN.2018.8471988.

[10] R.Patil, M.Poi,P.Pawar,T.Patil and N. Ghuse, "Blood donor's safety using data mining," 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), 2015 , pp. 500-505 , doi: 10.1109 /ICGCIoT.2015.7380516.

2.3 Problem Statement Definition

A patient in need of blood needs a way to get blood as soon as possible. So that the life of the patient can be saved.

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

S.NO	Parameter	Description
1	Problem statement(Problem to be solved)	To solve the improper communication and synchronization between the blood banks and hospitals which leads to the lose of life. Also to solve the difficulty in finding a blood donor in emergency situation.
2	Idea / Solution description	To provide a web application for the patient in need of blood using which the patient can able to find the blood donors and the donor's details including blood group certificate provided by the certified hospital. Also the patient can able to find the donors nearby the patient's location by using the location provided by the donor's. The donors can able to manage their profiles as active or inactive which helps the patient to easily find whether the particular donor is able to donate blood now or not.
3	Novelty / Uniqueness	Registered donors will be certified as verified users after checking the blood group document provided by a certified hospital in order to prevent the waste of time.
4	Social Impact / Customer Satisfaction	The patient can be able to find the donor in a new place and can able to check whether the donor is verified or not.
5	Business Model (Revenue Model)	The project is completely developed on service basis.
6	Scalability of the solution	The project is developed as a web application, so the user can access the web application from any platform. As we use cloud computing technology to deploy our web application it can manage huge amount of users request effectively.

3.4 Problem Solution Fit

The person who is need of blood	24×7 customers support,availability of blood any time simple interface	Service platform,awarness camp,high availability
Finding donar in unknown location user can get donar in his location,so the donation of blood will	The patient doesn't know how to start if they want blood Finding particular bloodgroup that are rare	Need improvement of customer service
When the patient requests for blood suddenly the request send to every available donors in his location	Need of blood was necessary in every hospitals at anytime we can rare some one is life.The patients can finding the blood much as faster	The user can get free service for checking the available of blood at any time and where

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional Requirement

Following are the functional requirements of the proposed solution.

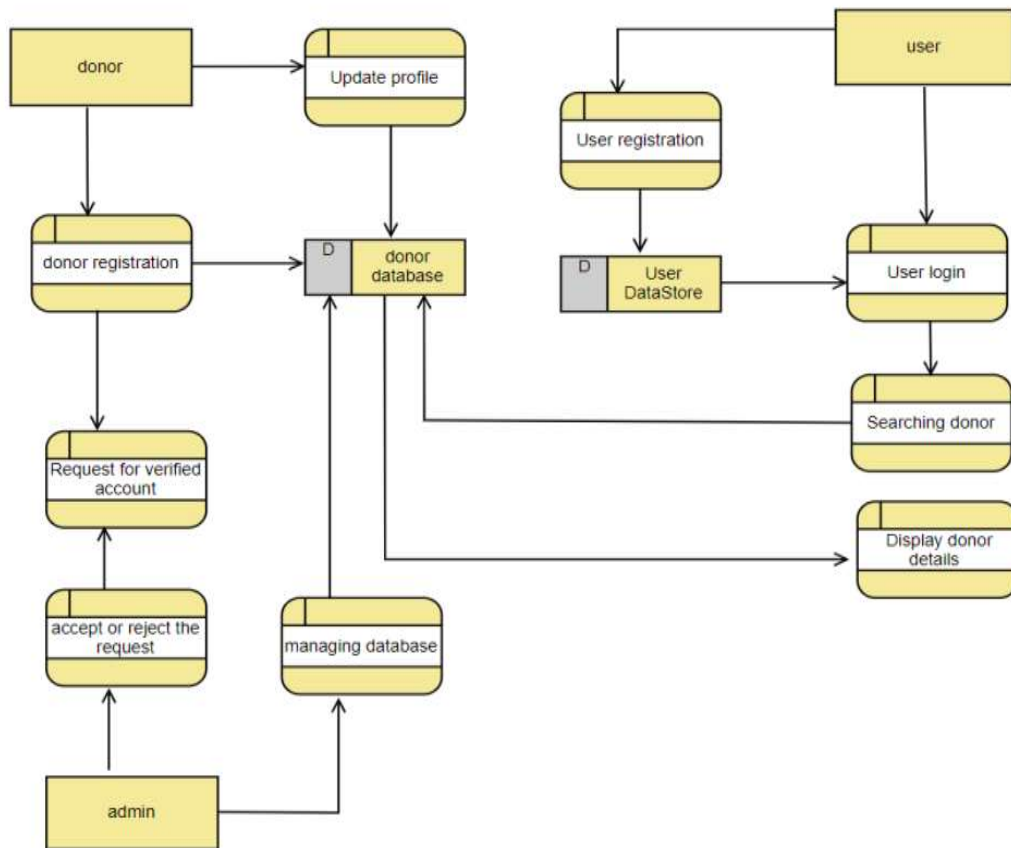| FR NO	FUNCTIONAL REQUIREMENTS (EPIC)	SUB REQUIREMENT (STORY/SUB-TASK)
FR-1	Login	The user needs to login using the login credentials.
FR- 2	User registration	If the user is new to the web application then the user should register in the web application.
FR-3	Donor registration	The donor registration should be done by the donor by providing the required information.
FR-4	Verified donor	To become a verified donor the donor should send request to the admin along with the required information and documents.
FR-5	User request for blood	User can able to search the particular plasma donor and can able to access the contact details of the particular donor
FR-6 donor	Admin login	Admin can able to login using the separate login credentials and verify the request send by the donor.
FR-7	Update the profile	The donor can able to update the profile after donation
FR-8	Feedback	The user can able to post feedback about the Donor

4.2 Non-Functional Requirements

| FR. No	Non-Functional Requirement	Description
NFR-1	1 Usability	This system is really use to find a suitable plasma donor for the patient who is in need of blood immediately. By finding the suitable plasma donor the patient's life can be saved.
NFR-2	Security	The details of the donors are stored in IBM Cloud using docker and kubernets technology so the security of the application is high.
NFR-3	Reliability	The consistency and the quality of the web application is maintained by providing regular updates and proper maintenance.
NFR-4	Performance	The performance mostly depends on the users who is in search of blood and it gives the faster information about the donors who are located in the user's same location. It gives more priority to the certified blood donors. It gives the location of the donor whether he/she is active or inactive. When the user requests for blood suddenly the system gives the notification to the available donors with particular blood group.
NFR-5	Availability	The availability of the solution is effective and it should be helpful to the patient inneed of blood. This web application can be utilized by every person in each corner of the world.
NFR-6	Scalability	It is scalable enough to store the every user's information in databases. It can give access to the every users to use the web application faster and reliable.

CHAPTER 5

## PROJECT DESIGN

5.1 Data Flow Diagram

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

5.2 Solution Architecture

5.2.1 Technical Architecture
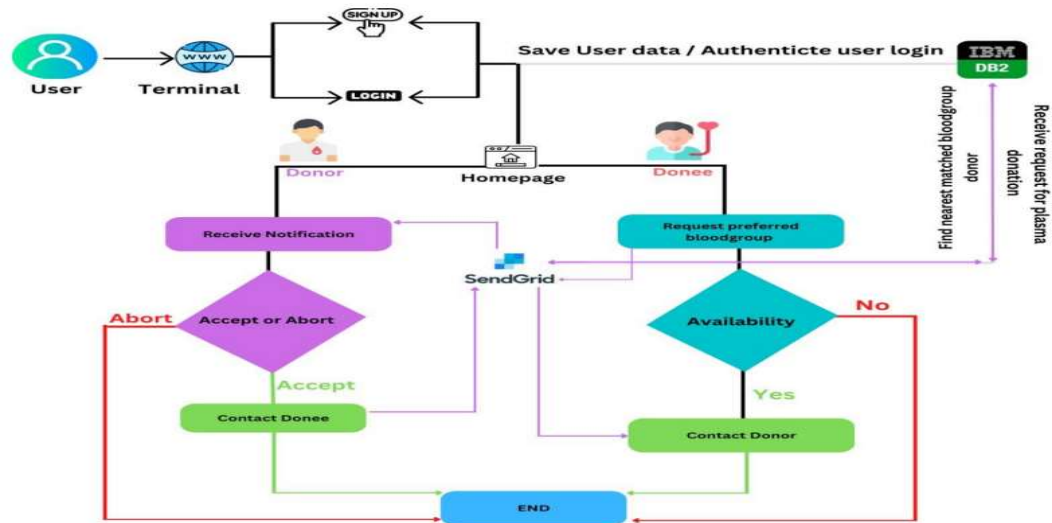
PLASMA DONOR APPLICATION
TECHNICAL ARCHITECTURE
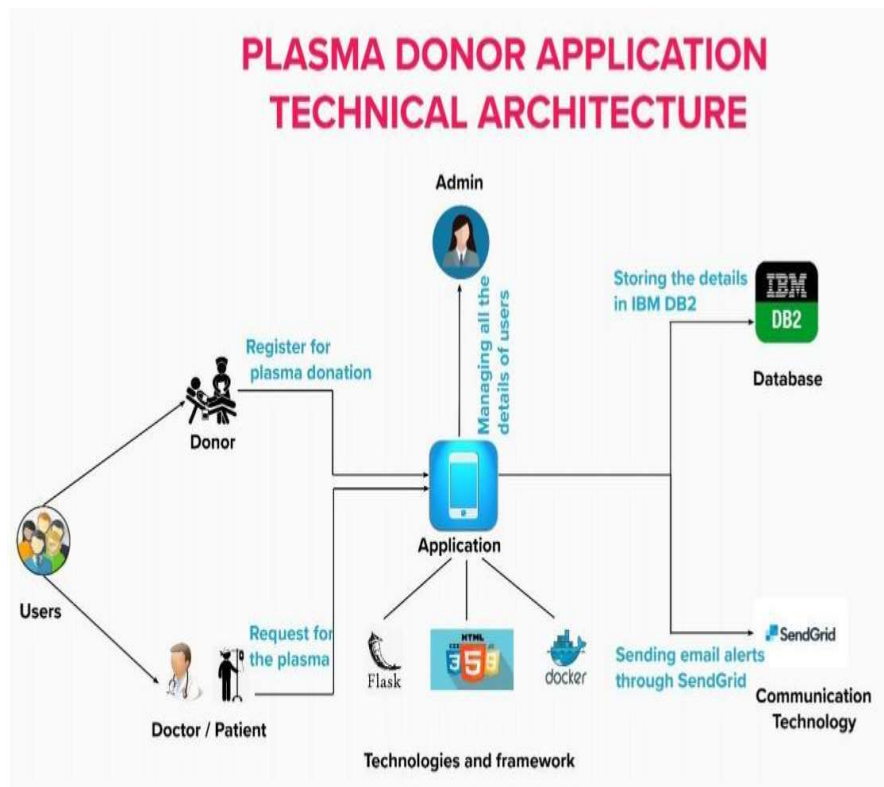
5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (web application user)	Donor Registration	USN-1	As a donor, I can register to the web application by filling the required details.	I can access my account / dashboard	High	Sprint-1
	User registration	USN-2	As a user, I can register to the web application by filling the required details.	I can access my account / dashboard	High	Sprint-1
	User login	USN-3	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-2
	User search for the donor	USN-4	As a user, I can able to search the particular plasma donor and can able access the contact details of the particular donor	I can find the suitable plasma donor	High	Sprint-3
	Verified donor	USN-5	As a donor, I can request for verified account by providing the required documents and details to the admin through the web application	I can register & access the dashboard with Facebook Login	Medium	Sprint-3
	Update the profile	USN-6	As a donor, I can update my profile any time.	. I can update my detail	Medium	Sprint-4

	Feedback	USN-7	As a user, I can give the feedback to the donor	I can share my experience with the particular donor	Low	Sprint-4
Administrator	Login	USN-8	As a admin, I can able to login using the seperate login credentials	I can access the admin panel	High	Sprint-2
	Verify the request	USN-9	As a admin, I can able to verify the request send by the donor to become a verified donor	As a admin, I can able to verify the request send by the donor to become a verified donor	High	Sprint-2

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story points	Priority	Team Members
Sprint 1	Donor Registration	USN-1	As a donor, I can register to the web application by filling the required details.	10	High	Valgin, Mohamed Ashif Ali
Sprint 1	User Registration	USN-2	As a user, I can register to the web application by filling the required details.	10	High	Valgin, Mohamed Ashif Ali
Sprint 2	User login	USN-3	As a user, I can log into the application by entering email & password.	10	High	Valgin, Mohamed Ashif Ali
Sprint 2	Admin Login	USN-4	As a admin, I can able to login using the separate login credentials.	10	High	Aswin Raja, Arockia Jebin
Sprint 2	Verify the Request	USN-5	As a admin, I can able to verify the request send by the donor to become a verified donor	10	High	Aswin Raja, Arockia jebin
Sprint 3	User search for the donor	USN-6	As a user, I can able to search the particular plasma donor and can able access the contact details of the donor.	10	High	Valgin, Mohamed Ashif Ali
Sprint 3	Verified Donor	USN-7	As a donor, I can request for verified account by providing the required documents and details to the admin through the web application.	8	Medium	Aswin Raja, ArockiaJebin

Sprint	Update the profile	USN-8	As a donor, I can update my profile any time.	8	Medium	
Sprint 4						Aswin Raja, Arockia Jebin
Sprint 4	Feedback	USN-9	As a user, I can give the feedback to the Donor.	7	Low	Valgin, Mohamed Ashif Ali

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	8 days	29-10-2022	5-11-2022	20	5-11-2022
Sprint 2	30	8 days	7-11-2022	14-11-2022	30	14-11-2022
Sprint 3	18	8 days	16-11-2022	23-11-2022	18	23-11-2022
Sprint 4	15	8 days	23-11-2022	30-11-2022	15	30-11-2022

Velocity

Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \text{Velocity} / \text{Sprint Duration}$$
$$= 20 / 8$$
$$= 2.5$$
$$AV = 30 / 8$$
$$AV = 18/8 = 2.25$$
$$AV = 15/8$$
$$= 1.87$$

Burndown Chart
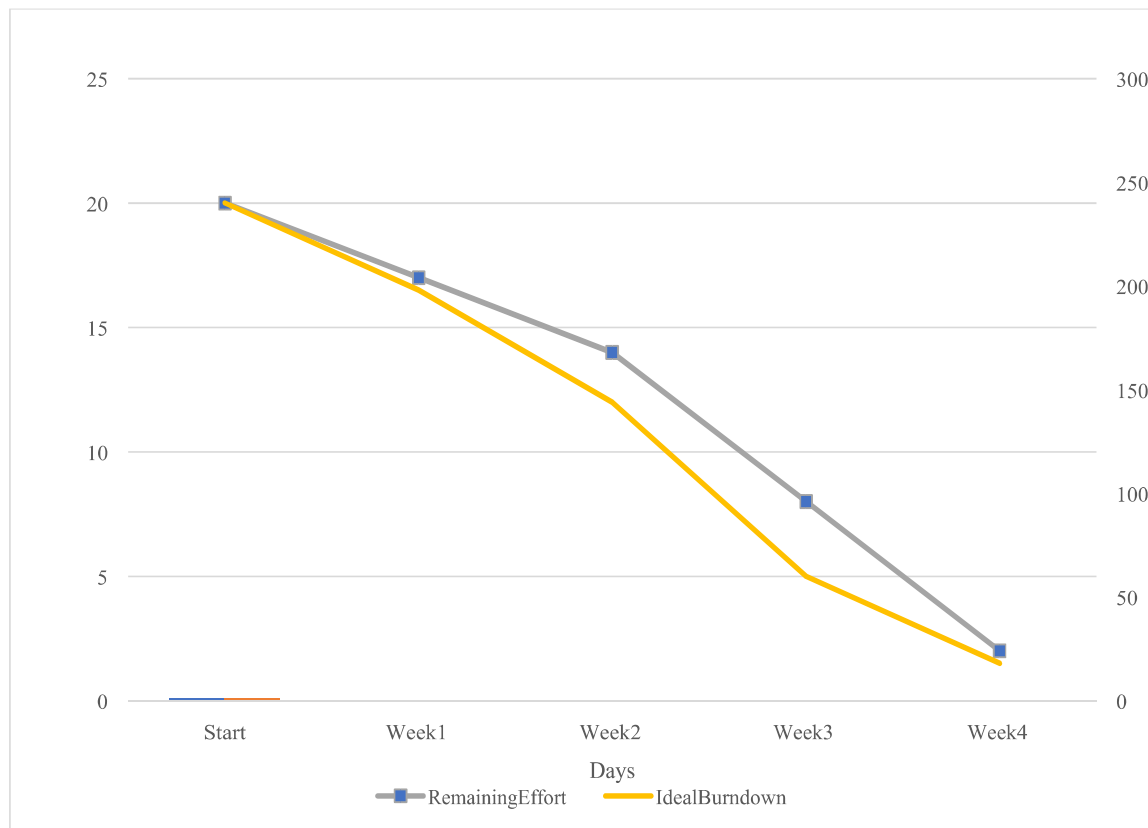
6.2 Sprint Delivery Schedule

Sprint	Sprint Topic	Start Date	Expected Delivery
Sprint 1	Donor Registration, User Registration	29-10-2022	5-11-2022
Sprint 2	User Login, Admin login, Verify the request	7-11-2022	14-11-2022
Sprint 3	User search for the donor, Verified donor	16-11-2022	23-11-2022
Sprint 4	Update the profile, Feedback	23-11-2022	30-11-2022

RESULT

7.1 Performance Matrics

Registration Page

Login Page

Home Page

Donor registration

ALTERNATIVE NUMBER :

9003261783

WHATS-APP NUMBER :

6382947757

E-MAIL :

jebin2211@gmail.com

LOCATION (copy and paste your home location link):

https://maps.google.com/

https://goo.gl/maps/EA8mAM9nPSdSiy8p7

ADDRESS :

22-89/b,kulasekaram

STATE :

STATE :

TamilNadu

CITY :

kanyakumari

OCCUPATION :

student

COVID-STATUS :

uninfected

COVID VACCINATION STATUS :

Covi-Shield

COVID VACCINATION DOSE :

Search Filter Data

SEARCH DONOR HERE

NAME	AGE	BLOOD GROUP	GENDER	CITY	CONTACT NUMBER	ALTERNATIVE NUMBER	WHATS APP NUMBER	LOCATION	COVID STATUS	ACCOUNT TYPE
jebin	22	B+	Male	kanyakumari	9003261783	9003261783	6382947757	DONOR'S LOCATION	uninfected	notverified

Admin Panel

ADMIN PANEL

NAME	AGE	BLOOD GROUP	GENDER	CONTACT NUMBER	ALTERNATIVE NUMBER	WHATS APP NUMBER	E-MAIL	LOCATION	ADDRESS	STATE	CITY	OCCU
ASHIF	22	O+	Male	6382947757	6382947757	7358860362	ashif180@gmail.com	https://goo.gl/maps/Bp6krZBZWqjSKmDD7	197-THIRUVITHANCODE	tamilnadu	kanyakumari	STUDE

CHAPTER 8

ADVANTAGES

> ➤ Donated blood can save the lives of people in need. However, according to some medical professionals, it may also benefit the donor.
> ➤ It is having professional staff on hand providing monitoring and refreshments to ensure a safe recovery.
> ➤ To find your needed blood group in your own city fast and reliable.
> ➤ It creates an awarness to the people for donate the blood in our society to save our life.
> ➤ Improper communication and synchronization between the blood banks and hospitals leads to wastage of the blood available.These problems can be dealt with by automating the existing manual blood bank management system.

DISADVANTAGES

> ➤ It takes a long time or the result might come at an odd time.
> ➤ Since it is webpage it may not load faster or better due to bad net- work connectivity.
> ➤ screening donors for existing health conditions.
> ➤ Manual document submitting and data entry.
> ➤ No proper coordination between different donors and users.

CHAPTER 9

CONCLUSION

The efficient way of finding plasma donor for the infected people is implemented using the plasma donor website that is hosted on IBM platform. To ensure the smooth functioning of the website operations. I have hosted the website in IBM platform to make sure the operations are running successfully IBM Flask function is used and to deploy the application IBM DB2 service is used.

CHAPTER 10

FUTURE SCOPE

Upgrading the UI that is more user-friendly which will help many users to access the website and also ensures that many plasma donors can be added into the community. Using elastic load balancer, it helps to handle multiple requests at the same time which will maintain the uptime of the website with negligible down time.The best way to recruit donors is face to face So we will use posters and science to engage your community with customizable press realease.Tracking the donor's live location feature will more useful to the user for knowing about the arrival time of the donor at the hospital.

Source Code

Login.html

```html
<html>
<title>
    loginpage
  </title>
  <head>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
  </head>
  <style>

.btn-login {
 font-size: 0.9rem;
 letter-spacing: 0.05rem;
 padding: 0.75rem 1rem;
}

.btn-google {
 color: white !important;
 background-color: #ea4335;
}

.btn-facebook {
 color: white !important;
 background-color: #3b5998;
}
  </style>
  <body>
    <div class="container">
     <div class="row">
       <div class="col-sm-9 col-md-7 col-lg-5 mx-auto">
        <div class="card border-0 shadow rounded-3 my-5">
         <div class="card-body p-4 p-sm-5">
          <h5 class="card-title text-center mb-5 fw-light fs-5">Sign In</h5>
            {{login_data}}
          <form action="/signin" method="POST">
            <div class="form-floating mb-3">
```

```html
                <input type="email" class="form-control" id="floatingInput"
placeholder="name@example.com" name="email">
                <label for="floatingInput">Email address</label>
            </div>
            <div class="form-floating mb-3">
                <input type="password" class="form-control" id="floatingPassword"
placeholder="Password" name="password">
                <label for="floatingPassword">Password</label>
            </div>

            <div class="form-check mb-3">
                <input class="form-check-input" type="checkbox" value=""
id="rememberPasswordCheck">
                <label class="form-check-label" for="rememberPasswordCheck">
                  Remember password
                </label>
            </div>
            <div class="d-grid">
                <button class="btn btn-primary btn-login text-uppercase fw-bold"
type="submit">Sign
                  in</button>
            </div>
            <hr class="my-4">
            <div class="d-grid mb-2">
                <button class="btn btn-google btn-login text-uppercase fw-bold"
type="submit">
                  <i class="fab fa-google me-2"></i> Sign in with Google
                </button>
            </div>
            <div class="d-grid">
                <button class="btn btn-facebook btn-login text-uppercase fw-bold"
type="submit">
                  <i class="fab fa-facebook-f me-2"></i> Sign in with Facebook
                </button>
            </div>
          </form>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

32

Register.html

```html
<html>
  <title>
    loginpage
  </title>
  <head>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">
  </head>
  <style>
.container {
  width: 50%;
  background-size: cover;
}

.btn-login {
  font-size: 0.9rem;
  letter-spacing: 0.05rem;
  padding: 0.75rem 1rem;
}

.btn-google {
  color: white !important;
  background-color: #ea4335;
}
.btn-facebook {
  color: white !important;
  background-color: #3b5998;
}
```

```html
</style>
<body>
  <div class="container">
    <div class="row">
      <div class="col-lg-10 col-xl-9 mx-auto">
        <div class="card flex-row my-5 border-0 shadow rounded-3 overflow-hidden">
          <div class="card-body p-4 p-sm-5">
            <h5 class="card-title text-center mb-5 fw-light fs-5">Register</h5>
            {{result}}
            <form action="/register_login" method="POST">


              <div class="form-floating mb-3">
                <input type="text" class="form-control" id="floatingInputUsername"
placeholder="myusername" name="name" required autofocus>
                <label for="floatingInputUsername">Username</label>
              </div>


              <div class="form-floating mb-3">


                <input type="email" class="form-control" id="floatingInputEmail"
placeholder="name@example.com" name="email">
                <label for="floatingInputEmail">Email address</label>
              </div>


              <hr>


              <div class="form-floating mb-3">
                <input type="password" class="form-control" id="floatingPassword"
placeholder="Password" name="password">
                <label for="floatingPassword">Password</label>
              </div>


              <div class="d-grid mb-2">
```

```html
            <button class="btn btn-lg btn-primary btn-login fw-bold text-uppercase"
type="submit">Register</button>
          </div>

          <a class="d-block text-center mt-2 small" href="/login">Have an account? Sign
In</a>

          <hr class="my-4">

          <div class="d-grid mb-2">
            <button class="btn btn-lg btn-google btn-login fw-bold text-uppercase"
type="submit">
              <i class="fab fa-google me-2"></i> Sign up with Google
            </button>
          </div>

          <div class="d-grid">
            <button class="btn btn-lg btn-facebook btn-login fw-bold text-uppercase"
type="submit">
              <i class="fab fa-facebook-f me-2"></i> Sign up with Facebook
            </button>
          </div>

        </form>
      </div>
     </div>
    </div>
   </div>
  </div>
 </body>
 </html>
```

Search.html

```html
<html>
<head>
    <title> search </title>
</head>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet"
    integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
<style>
table{
  font-family: Arial, Helvetica, sans-serif;
  border-collapse: collapse;
  width: 100%;
}

 td,  th {
  border: 1px solid #ddd;
  padding: 8px;
}
tr:nth-child(even){background-color: #f2f2f2;}
 tr:hover {background-color: #ddd;}

th {
  padding-top: 12px;
  padding-bottom: 12px;
  text-align: left;
  background-color: #04AA6D;
  color: white;
}
```

```html
</style>
<body>
<center><h1>SEARCH DONOR HERE</h1></center>
<br>
<form action="/search" method="POST">
   <div class="select1" >
   <select class="form-select form-select-lg mb-3" aria-label=".form-select-lg example"name="bloodgroup" type="search">
   <option value="Course">BloodGroup</option>
   <option value="A+">A+</option>
   <option value="A-">A-</option>
   <option value="B+">B+</option>
   <option value="B-">B-</option>
   <option value="O+">O+</option>
   <option value="O-">O-</option>
   <option value="AB+">AB+</option>
   <option value="AB-">AB-</option>
   </select>
      <select class="form-select form-select-lg mb-3" aria-label=".form-select-lg example" name="city" type="search">
        <option value="Course">CITY</option>
 <option value="kanyakumari">kanyakumari</option>
 <option value="Chennai">Chennai</option>
 <option value="tirunelveli">Tirunelveli</option>

 </select>
      <center> <button  class="btn btn-primary btn-lg">search</button></center>
   </div>
   </form>
<div class="div1">
   <table>
   <tr>
      <th>NAME</th>
```

```html
      <th>AGE</th>
      <th>BLOOD GROUP</th>
      <th>GENDER</th>
      <th>CITY</th>
      <th>CONTACT NUMBER</th>
      <th>ALTERNATIVE NUMBER</th>
      <th>WHATS APP NUMBER</th>
      <th>LOCATION</th>
      <th>COVID STATUS</th>
      <th>ACCOUNT TYPE</th>
  </tr>
{%for x in posts%}
 <tr> <td> {{x.name}}</td>
 <td> {{x.age}}</td>
  <td>{{x.bloodgroup}}</td>
 <td>{{x.gender}}</td>
 <td>{{x.city}}</td>
 <td><a href="tel:{{x.contactnumber}}">{{x.contactnumber}}</a></td>
 <td><a href="tel:{{x.alternativenumber}}">{{x.alternativenumber}}</a></td>
  <td><a
href=https://wa.me/+91{{x.whatsappnumber}}?text=HI%20I%20AM%20IN%20NEED%20
OF%20BLOOD%20URGENTLY%20 target="blank">{{x.whatsappnumber}}</a></td>
  <td><a href={{x.location}} target="blank">DONOR'S LOCATION</a></td>
   <td> {{x.covidstatus}}</td>
 <td> {{x.verified}}</td></tr>
{%endfor%}

     </table>
</div>
</body>
</html>
```

Admin.html

```html
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/html">
<head>
    <meta charset="UTF-8">
    <title>adminpage</title>
    <style>
    table{
 font-family: Arial, Helvetica, sans-serif;
 border-collapse: collapse;
 width: 100%;
}

 td,  th {
  border: 1px solid #ddd;
  padding: 8px;
}
tr:nth-child(even){background-color: #f2f2f2;}
 tr:hover {background-color: #ddd;}

th {
  padding-top: 12px;
  padding-bottom: 12px;
  text-align: left;
  background-color: #04AA6D;
  color: white;
}
    </style>
</head>
<body>
<center><h1>ADMIN PANEL</h1></center>
<form action="/admin" method="POST">
    <table>
    <tr>
        <th>NAME</th>
        <th>AGE</th>
        <th>BLOOD GROUP</th>
        <th>GENDER</th>
        <th>CONTACT NUMBER</th>
        <th>ALTERNATIVE NUMBER</th>
        <th>WHATS APP NUMBER</th>
        <th>E-MAIL</th>
        <th>LOCATION</th>
        <th>ADDRESS</th>
        <th>STATE</th>
        <th>CITY</th>
        <th>OCCUPATION</th>
        <th>COVID STATUS</th>
```

```html
        <th>COVID-VACCINATION</th>
        <th>COVID-DOSAGE</th>
        <th>DRUG USER</th>
        <th>HEALTH ISSUES</th>
        <th>ACCOUNT TYPE</th>
        <th>DELETE</th>
        <th>VERIFIED ACCESS</th>
    </tr>
     </tr>
{%for x in posts%}
  <tr> <td> {{x.name}}</td>
  <td> {{x.age}}</td>
   <td>{{x.bloodgroup}}</td>
  <td>{{x.gender}}</td>
   <td>{{x.contactnumber}}</td>
   <td>{{x.alternativenumber}}</td>
   <td>{{x.whatsappnumber}}</td>
    <td>{{x.email}}</td>
   <td>{{x.location}}</td>
    <td>{{x.address}}</td>
    <td>{{x.state}}</td>
    <td>{{x.city}}</td>
    <td>{{x.occupation}}</td>
    <td>{{x.covidstatus}}</td>
     <td>{{x.covidvaccinationstatus}}</td>
     <td>{{x.covidvaccinationdose}}</td>
     <td>{{x.druguser}}</td>
    <td>{{x.comments}}</td>
    <td>{{x.verified}}</td>
  <td>
    <form action="/deleteform" method="POST">
      <button input type="hidden" value="{{x.id}}" name="id">delete</button>
    </form>
  </td>
   <td>
    <form action="/updateform" method="POST">
      <button input type="hidden" value="{{x.id}}" name="id">verified</button>
    </form>
  </td></tr>
{%endfor%}
</table>
</form>
</body>
</html>
```

Index.html

```html
<html>
   <title>plasma donor application</title>
   <head>
      <link rel="stylesheet" href="index.css">
      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet"
    integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
   </head>
   <body>
      <style>
      /* Style the header with a grey background and some padding */
.header {
   overflow: hidden;
   background-color: #f1f1f1;
   padding: 20px 10px;
 }

 /* Style the header links */
 .header a {
   float: left;
   color: black;
   text-align: center;
   padding: 12px;
   text-decoration: none;
   font-size: 18px;
   line-height: 25px;
   border-radius: 4px;
 }

  /* Style the logo link (notice that we set the same value of line-height and font-size to
prevent the header to increase when the font gets bigger */
 .logo {
   font-size: 40px;
   font-weight: bold;
 }

 /* Change the background color on mouse-over */
 .header a:hover {
   background-color: #ddd;
   color: black;
 }

 /* Style the active/current link*/
 .header a.active {
```

```css
    color: white;
}

/* Float the link section to the right */
.header-right {
  float: right;
}
.pic1{
  height: 80vmin;
  width: 100%;


}
.btn {
  position: absolute;
  top: 80%;
  left: 46%;
  transform: translate(-50%, -50%);
  -ms-transform: translate(-50%, -50%);
  background-color:red;
  color: white;
  font-size: 20px;
  padding: 12px 24px;
  border: none;
  cursor: pointer;
  border-radius: 5px;
}
.btn1 {
  position: absolute;
  top: 80%;
  left: 80%;
  transform: translate(-50%, -50%);
  -ms-transform: translate(-50%, -50%);
  background-color:red;
  color: white;
  font-size: 20px;
  padding: 12px 24px;
  border: none;
  cursor: pointer;
  border-radius: 5px;
}


.container .btn:hover {
  background-color: black;
}

/* Add media queries for responsiveness - when the screen is 500px wide or less, stack the
links on top of each other */
@media screen and (max-width: 500px) {
  .header a {
```

```css
  float: none;
  display: block;
  text-align: left;
 }
 .header-right {
  float: none;
 }
}

      ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
        overflow: hidden;
        background-color: whitesmoke;
      }

      li {
        float: right;
        border-right:1px solid #bbb;
      }

      li:last-child {
        border-right: none;
      }

      li a {
        display: block;
        color: white;
        text-align: center;
        padding: 14px 16px;
        text-decoration: none;
      }

      li a:hover:not(.active) {
        background-color: #111;
      }

      .active {
        background-color: #04AA6D;
      }
      .logo{

      }
    </style>

    <ul>
    <li style="float:left" class ="logo">PLASMA DONOR APPLICATION</li>
      <li><a class="active" href="/login">SIGN IN</a></li>
```

```html
      </ul>
    <br>
      <div class ="divS1">
       <img src="{{url_for('static',filename='pic1.jpg')}}" class="pic1">
       <a href="/donorregistration"><button type="button" class="btn btn-primary btn-
lg"><blink>DONOR REGISTRATION</blink></button></a>
        <a href="/searchbar"><button type="button" class="btn1 btn-primary btn-
lg"><blink>SEARCH DONOR</blink></button></a>
      </div>
      <div class="div2">
       <h2>
          WHO CAN DONATE PLASMA ?
       </h2>
       <br>
       <h3>
         AGE:
       </h3>
       <h5>
          You are aged between 18 and 65.In some countries national legislation permits 16–
17 year-olds to donate provided that they fulfil the physical and hematological criteria
required and that appropriate consent is obtained.In some countries, regular donors over the
age of 65 may be accepted at the discretion of the responsible physician.
          The upper age limit in some countries are 60.
       </h5>
      </div>
   <div class="3">
      <h3>
        WEIGHT:
      </h3>
      <h5>
         You weigh at least 50 kg. In some countries, donors of whole blood donations should
weigh at least 45 kg to donate 350 ml ± 10% .
      </h5>
   </div>
   <div class="3">
      <h3>
        HEALTH:
      </h3>
   <h5>
      You must be in good health at the time you donate.

You cannot donate if you have a cold, flu, sore throat, cold sore, stomach bug or any other
infection.

If you have recently had a tattoo or body piercing you cannot donate for 6 months from the
date of the procedure.
   </h5>
      </div>
   <div class="3">
      <h3>
```

TRAVEL:
    </h3>
  <h5>
      Travel to areas where mosquito-borne infections are endemic, e.g. malaria, dengue and Zika virus infections, may result in a temporary deferral .

Many countries also implemented the policy to defer blood donors with a history of travel or residence for defined cumulative exposure periods in specified countries or areas, as a measure to reduce the risk of transmitting variant Creutzfeldt-Jakob Disease (vCJD) by blood transfusion.
    </h5>
    <div class="3">
      <h3>
        BEHAVIOUR:
      </h3>
  <h5>
      You must not give blood:

If you engaged in "at risk" sexual activity in the past 12 months

Individuals with behaviours below will be deferred permanently:

Have ever had a positive test for HIV (AIDS virus)

Have ever injected recreational drugs.

 In the national blood donor selection guidelines, there are more behavior eligibility criteria. Criteria could be different in different countries.
egnancy and breastfeeding:

Following pregnancy, the deferral period should last as many months as the duration of the pregnancy.

It is not advisable to donate blood while breast-feeding. Following childbirth, the deferral period is at least 9 months (as for pregnancy) and until 3 months after your baby is significantly weaned (i.e. getting most of his/her nutrition from solids or bottle feeding).
    </h5>
    </body>
</html>

Plasmad.py

```python
from flask import Flask,request, render_template, redirect, Response
from flask_sqlalchemy import SQLAlchemy
from datetime import datetime
from werkzeug.utils import secure_filename
from base64 import b64encode, b64decode
import jinja2
import os
app = Flask(__name__)
basedir = os.path.abspath(os.path.dirname(__file__))
app.config['SQLALCHEMY_DATABASE_URI'] = "sqlite:///"
+os.path.join(basedir,'plasmadonor.db')
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

db = SQLAlchemy(app)
class Register(db.Model):
    id = db.Column(db.Integer, primary_key = True)
    name = db.Column(db.String(50), nullable = False)
    email =db.Column(db.String(50), nullable = False, unique = True)
    password = db.Column(db.Integer, nullable = False, unique = True)
    date_joined = db.Column(db.Date,default = datetime.utcnow)

    def __repr__(self):
        return f"<User : {self.email}>"


class Admin(db.Model):
    id = db.Column(db.Integer, primary_key = True)
    name = db.Column(db.String(50), nullable = False)
    password = db.Column(db.Integer, nullable = False, unique = True)

    def __repr__(self):
        return f"<User : {self.name}>"
class Donor_Registrations(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(50), nullable=False)

    age = db.Column(db.Integer, nullable=False)
    bloodgroup = db.Column(db.String(50), nullable=False)
    gender = db.Column(db.String(50), nullable=False)
    contactnumber = db.Column(db.Integer, nullable=False)
    alternativenumber = db.Column(db.Integer, nullable=False)
    whatsappnumber = db.Column(db.Integer, nullable=False)
    email = db.Column(db.String(50), nullable=False, unique=True)
    location = db.Column(db.String(50), nullable=False)
    address = db.Column(db.String(50), nullable=False)
    state = db.Column(db.String(50), nullable=False)
    city = db.Column(db.String(50), nullable=False)
```

```python
    occupation = db.Column(db.String(50), nullable=False)
    covidstatus= db.Column(db.String(50), nullable=False)
    covidvaccinationstatus = db.Column(db.String(50), nullable=False)
    covidvaccinationdose = db.Column(db.String(50), nullable=False)
    druguser = db.Column(db.String(50), nullable=False)
    comments = db.Column(db.String(150), nullable=False)
    verified = db.Column(db.String(50),nullable=False)


    def __repr__(self):
        return f"<User : {self.email}>"

with app.app_context():

    db.create_all()

@app.route('/')
def home():
    return render_template('register.html')

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

@app.route('/login')
def login():
    return render_template('login.html')
@app.route('/donorregistration')
def donorregister():
    return render_template('registration.html')


@app.route('/admin.html')
def admin_page():
    query = Donor_Registrations.query.order_by(Donor_Registrations.name).all()
    print(query)
    return render_template("admin.html", posts=query)

    #return render_template("admin.html")

@app.route('/register_login',methods = ['GET','POST'])
def register_login():
    if request.method == 'POST':

        name = request.form['name']
        email = request.form['email']
        password = request.form['password']


        query = Register(name = name, email = email, password = password)
```

```python
        db.session.add(query)
        db.session.commit()
        return render_template('index.html')


@app.route('/signin',methods=['GET','POST'])
def signin():
    if request.method == 'POST':
        email = request.form.get('email')
        password = request.form.get('password')
        login = Register.query.filter_by(email=email, password = password).first()
        #query = Admin(email='admin@gmail.com',password= "admin")
        #db.session.add(query)
        # db.session.commit()
        if login  is not None:
            return render_template('index.html', login_data= email)
        elif Admin is not None:
            return redirect('admin.html')
            #pass
        else:
            return render_template('login.html', login_data="make sure u enter the coorrect
pasword")


@app.route('/donor',methods=['GET','POST'])
def donor():
    if request.method == 'POST':
        name = request.form.get('name')
        age = request.form.get('age')
        bloodgroup = request.form['bloodgroup']
        gender =request.form['gender']
        contactnumber =request.form['contactnumber']
        alternativenumber =request.form['alternativenumber']
        whatsappnumber =request.form['whatsappnumber']
        email=request.form['email']
        location =request.form['location']
        address =request.form['address']
        state = request.form['state']
        city =request.form['city']
        occupation = request.form['occupation']
        covidstatus = request.form['covidstatus']
        covidvaccinationstatus = request.form['covidvaccinationstatus']
        covidvaccinationdose = request.form['covidvaccinationdose']
        druguser = request.form['druguser']
        comments = request.form['comments']
        verified = "notverified"
        query = Donor_Registrations(name=name,  age=age, bloodgroup=bloodgroup,
gender=gender, contactnumber=contactnumber, alternativenumber=alternativenumber,
whatsappnumber=whatsappnumber, email=email, location=location, address= address,
                state=state, city=city, occupation=occupation, covidstatus=covidstatus,
covidvaccinationstatus=covidvaccinationstatus, covidvaccinationdose=covidvaccinationdose,
druguser=druguser, comments=comments, verified=verified)
```

```python
        db.session.add(query)
        db.session.commit()
        return render_template('index.html', res = "succesfully")

@app.route('/searchbar', methods=['GET','POST'])
def searchbar():
    return render_template("search.html")

@app.route('/search', methods=['GET','POST'])
def search():
    if request.method == 'POST':
        search1 = request.form.get('bloodgroup')
        search2 = request.form.get('city')
        #query = Donor_Registration.query.filter(Donor_Registration.bloodgroup,
city.like("%" + search1 + "%", "%" + search2 + "%")).all()
        query
=Donor_Registrations.query.filter(Donor_Registrations.bloodgroup.like(search1),Donor_Reg
istrations.city.like(search2))
        return render_template('search.html', posts=query)

@app.route('/deleteform',methods=['GET','POST'])
def delete():
    if request.method == 'POST':
        id = request.form.get('id')
        deleted = Donor_Registrations.query.get(id)
        db.session.delete(deleted)
        db.session.commit()
        return redirect('/admin.html')

@app.route('/updateform',methods = ['GET','POST'])
def updatedata():
    if request.method == 'POST':
        id = request.form['id']
        Donor_Registrations.query.filter_by(id = id).update(dict(verified="verified"))
        db.session.commit()
        return redirect('/admin.html')


if __name__ == '__main__':
    app.run(debug=True)
```

GitHub & Project Demo Link

GitHub link : https://github.com/IBM-EPBL/IBM-Project-36447-1660295088

Project Demo Link : https://drive.google.com/file/d/19jHMlC9a8O3UWJNx2yt6JkWw-0XE6Pq7/view?usp=share_link

https://youtu.be/rNbtMLljfxI