



# **PLASMA DONOR APPLICATION**

## **A PROJECT REPORT**

*Submitted by*

**DEVADHARSIKA A (19EUCS027)**  
**DHIVYADHARSHINI T (19EUCS031)**  
**KAMALIKA M M (19EUCS057)**  
**HARI PRASATH P (19EUCS180)**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

*in*

**COMPUTER SCIENCE AND ENGINEERING**

**SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY**  
**(An Autonomous Institution, Affiliated to Anna University Chennai - 600 025)**

**NOVEMBER 2022**



**SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY**  
(An Autonomous Institution. Affiliated to Anna University, Chennai)  
**Kuniamuthur, Coimbatore - 641 008**



## **BONAFIDE CERTIFICATE**

Certified that this project report titled “**PLASMA DONOR APPLICATION**” is the bonafide work of **Miss.DEVADHARSIKA A (19EUCS027)** , **Miss.DHIVYADHARSHINI T (19EUCS031)** , **Miss.KAMALIKA M M (19EUCS057)** , **Mr.HARI PRASATH P (19EUCS180)** who carried out the project work under my supervision.

**SIGNATURE**

**Dr.K. SASI KALA RANI, M.E., Ph.D.,**  
**HEAD OF THE DEPARTMENT**

**SIGNATURE**

**Dr.M.SUJARITHA,M.E.,Ph.D.,**  
**SUPERVISOR**

Department of Computer Science and Engineering  
Sri Krishna College of Engineering and Technology  
Kuniamuthur,Coimbatore

**Submitted for the Project viva-voce examination held on\_\_\_\_\_**

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## **ABSTRACT**

During the COVID 19 crisis, the requirement of plasma became a high priority and the donor count has become low. Saving the donor information and helping the needy by notifying the current donors list, would be a helping hand . Alternatively, now a day's plasma transplant surgery is also being performed rapidly. At this present time plasma banks are in short supply. Not only that, but the number of plasma donors is low too. And some people do not know what plasma donation is and where to donate plasma. We have set up a system to alleviate this situation and help needy people to identify plasma donors and plasma banks. In regard to the problem faced , an application is to be built which would take the donor details, store them and inform them upon a request. The main objective is to develop an Android application to build a network of people (Donors, Recipients and Health care departments) who can help each other. This application is developed to simply explore for plasma in near areas for emergency.

# TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	<b>ABSTRACT</b>	<b>III</b>
	<b>LIST OF FIGURES</b>	<b>VI</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 Project Overview	1
	1.2 Purpose	2
<b>2</b>	<b>LITERATURE SURVEY</b>	<b>3</b>
	2.1 Existing problem	3
	2.2 References	3
	2.3 Problem Statement Definition	5
<b>3</b>	<b>IDEATION &amp; PROPOSED SOLUTION</b>	<b>6</b>
	3.1 Empathy Map Canvas	6
	3.2 Ideation & Brainstorming	7
	3.3 Proposed Solution	8
	3.4 Problem Solution fit	9
<b>4</b>	<b>REQUIREMENT ANALYSIS</b>	<b>10</b>
	4.1 Functional requirement	10
	4.2 Non-Functional requirements	11
<b>5</b>	<b>PROJECT DESIGN</b>	<b>12</b>
	5.1 Data Flow Diagrams	12
	5.2 Solution & Technical Architecture	13
	5.3 User Stories	14
<b>6</b>	<b>PROJECT PLANNING &amp; SCHEDULING</b>	<b>16</b>
	6.1 Sprint Planning & Estimation	16
	6.2 Sprint Delivery Schedule	17

	6.3 Reports from JIRA	18
<b>7</b>	<b>CODING &amp; SOLUTIONING</b>	<b>19</b>
	7.1 Feature 1	19
	7.2 Feature 2	26
	7.3 Feature 3	32
	7.4 Feature 4	41
	7.5 Feature 5	51
<b>8</b>	<b>TESTING</b>	<b>58</b>
	8.1 Test Cases	58
	8.2 User Acceptance Testing	59
<b>9</b>	<b>RESULTS</b>	<b>61</b>
	9.1 Performance Metrics	61
<b>10</b>	<b>ADVANTAGES &amp; DISADVANTAGES</b>	<b>62</b>
<b>11</b>	<b>CONCLUSION</b>	<b>63</b>
<b>12</b>	<b>FUTURE SCOPE</b>	<b>64</b>
<b>13</b>	<b>APPENDIX</b>	<b>65</b>
	Source Code	65
	GitHub & Project Demo Link	105

## LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
1.1	Empathy Map	
1.2	Problem Statement & Brainstorm	
1.3	Group Ideas & Prioritize	
1.4	Problem Solution fit	
2.1	Data Flow Diagram	
2.2	Architecture diagram	
3.1	Sprint Delivery Schedule	
3.2	Sprint Report	

# INTRODUCTION

## 1.1 OVERVIEW

During the COVID 19 crisis, the requirement of plasma became a high priority and the donor count has become low. Saving the donor information and helping the needy by notifying the current donors list, would be a helping hand. In regard to the problem faced, an application is to be built which would take the donor details, store them and inform them upon a request. The main goal of our project is to design a user-friendly web application that is like a scientific vehicle from which we can help reduce mortality or help those who are need of plasma , plasma therapy is an experimental approach to treat those COVID-positive patients and help them recover faster. Therapy, which is considered reliable and safe. If a particular person has fully recovered from COVID19, they are eligible to donate their plasma. As we all know, the traditional methods of finding plasma, one has to find out for oneself by looking at hospital records and contacting donors have been recovered, sometimes may not be available at home and move to other places. In this type of scenario, the health of those who are sick becomes disastrous. Therefore, it is not considered a rapid process to find plasma.

## **1.2 PURPOSE**

A user friendly and responsive interface with a quick notification system which instantly notifies the donor upon receiving a request. When the recipient requests for plasma, if there is lack of plasma at the time of request, automatically the recipient will be added to the waiting list. Later when there is availability of plasma, the recipient will be notified by email. The main purpose of the proposed system, the donor who wants to donate plasma can register and can donate the plasma to the blood bank, the recipient can request for the donor and once the donor has accepted the request, the donor can donate blood at blood bank and the recipient can also track the status of the request for plasma and can take the plasma from the blood bank.



# **LITERATURE SURVEY**

## **2.1 EXISTING PROBLEM**

During the COVID 19 crisis, the requirement of plasma became a high priority and the donor count has become low. Saving the donor information and helping the needy by notifying the current donors list, would be a helping hand . Alternatively, now a day's plasma transplant surgery is also being performed rapidly. At this present time plasma banks are in short supply. Not only that, but the number of plasma donors is low too. And some people do not know what plasma donation is and where to donate plasma.

## **2.2 REFERENCES**

- [1] Dennis O'Neil(1999). "Blood Components".Palomar College. Archived from the original on June 5,2013.
- [2] Tuskegee University(May 29, 2013). "Chapter 9 Blood".tuskegee.edu. Archived from the original on December 28, 2013.
- [3] "Ways to Keep Your Blood Plasma Healthy". Archived from the original on November 1, 2013.Retrieved November 10, 2011.

- [4] Jump up to Maton, Anthea; Jean Hopkins; Charles William McLaughlin; Susan Johnson; Maryanna Quon Warner LaHart; David LaHart; Jill D. Wright (1993), Human Biology and Health, Englewood Cliffs, New Jersey, USA.
- [5] The Physics Factbook — Density of Blood.[6]Basic Biology(2015).”Blood cells”.
- [6] Elkassabany NM, Meny GM, Doria RR, Marcucci C (2008). “Green Plasma Revisited”. Anesthesiology 108(4);
- [7] “19<sup>th</sup> WHO Model List of Essential Medicines(April 2015)”(PDF). WHO April 2015. Retrieved May 10, 2015.
- [8] Tripathi S, Kumar V,Prabhakar A, Joshi S, Agarwal A(2015).”Passive blood plasma separation at the microscale; a review of design principles and microdevices”. J.Micromech, Microeng 25(8); 083001.
- [9] Guo, Weijin; Hansson, Jonas; van der wijngaart, Wouter(2020).”Synthetic Paper Separates Plasma from Whole Blood with Low Protein Loss”.Analytical Chemistry.92(9): 6194-6199.
- [10] Mani A, Poornima AP, Gupta D(2019) “Greenish discoloration of plasma: Is it really a matter of concern?”, Asian Journal of Transfusion Science.
- [11] Starr, Douglas P.(2000), Blood: An Epic History of Medicine and Commerce. New York:Quill.

## **2.3 PROBLEM STATEMENT DEFINITION**

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

# IDEATION & PROPOSED SOLUTION

## 3.1 EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to help teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it.

The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

Empathy Map for Plasma Donor Application :

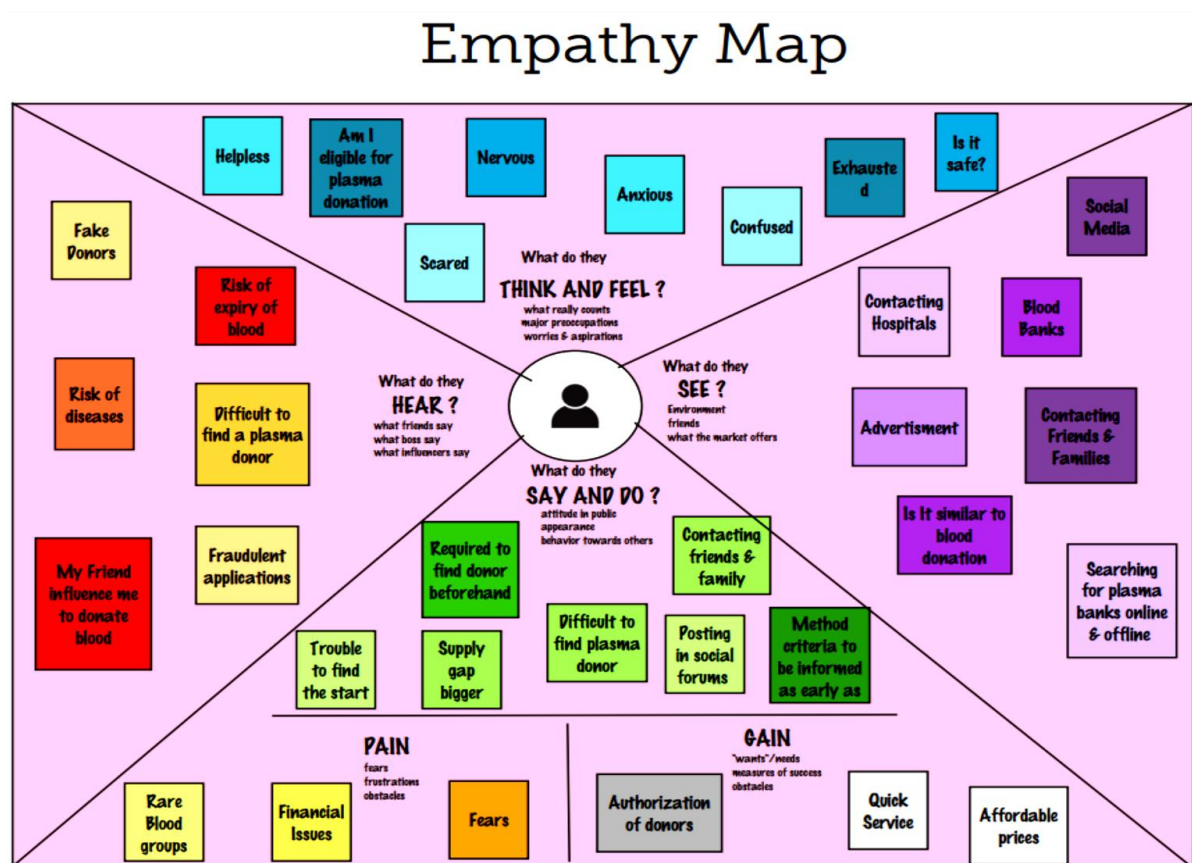


Fig 1.1. Empathy Map

## 3.2 IDEATION & BRAINSTORMING

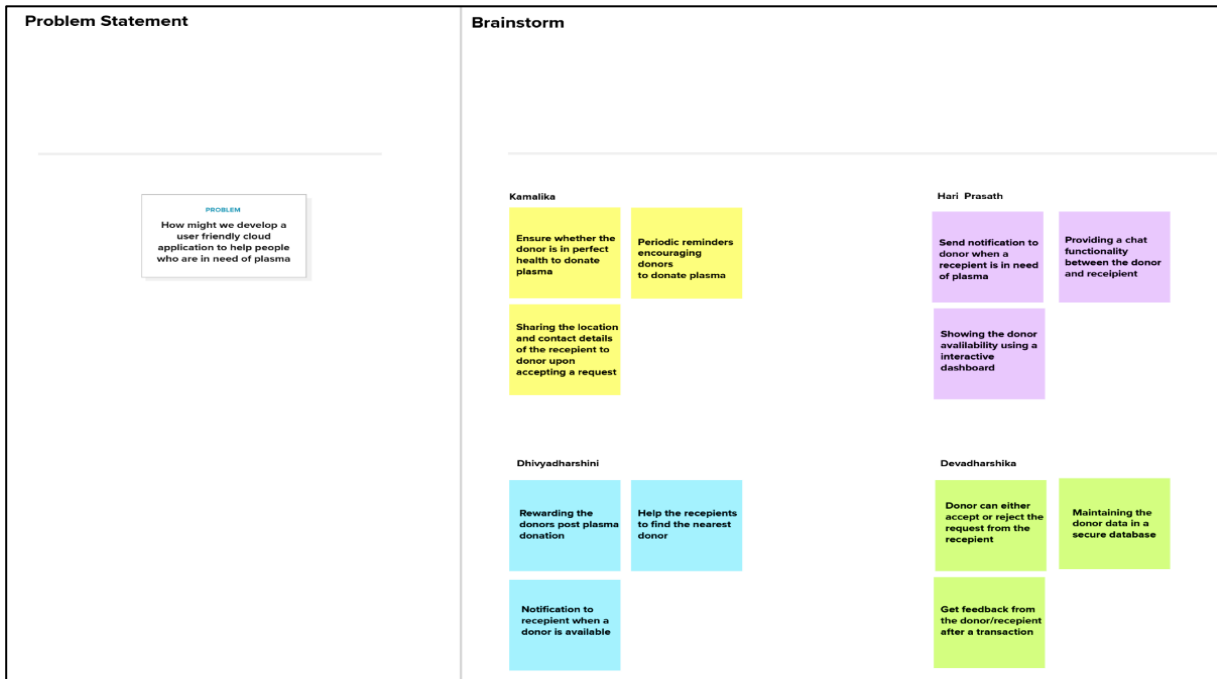


Fig 1.2.Problem Statement & Brainstorm

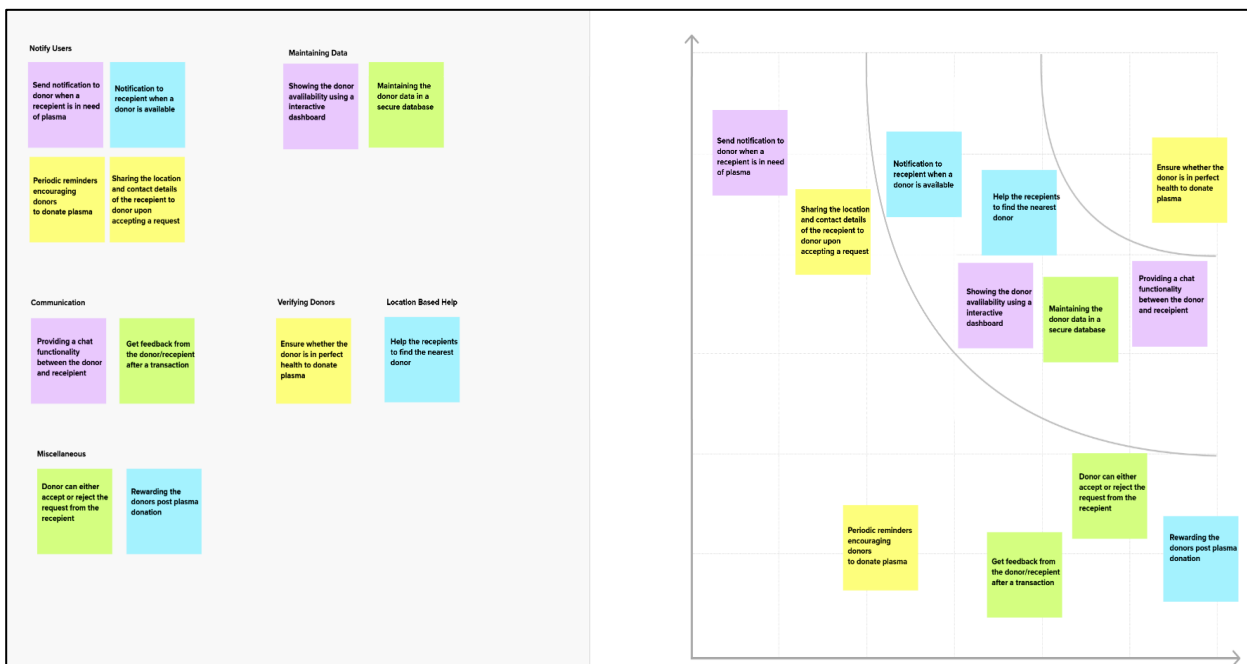
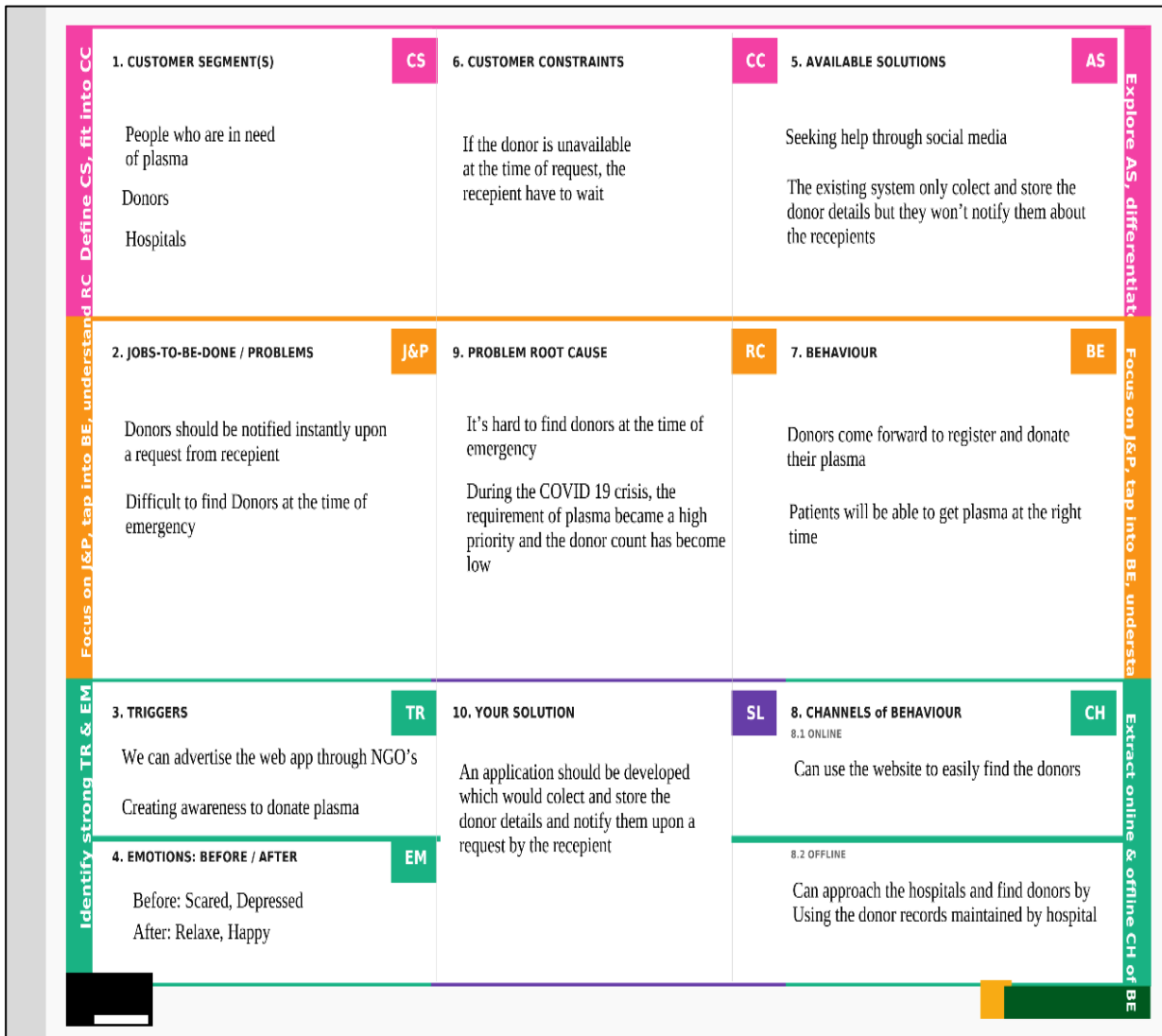


Fig 1.3.Group Ideas & Prioritize

### **3.3 PROPOSED SOLUTION**

The requirement of plasma became a high priority and the donor count has become low. Saving the donor information and helping the needy by notifying the current donors list, would be a helping hand. An application should be developed which would take the donor details, store them and notify them upon a request. A user friendly and responsive interface with a quick notification system which instantly notifies the donor upon receiving a request. When the recipient requests for plasma, if there is lack of plasma at the time of request, automatically the recipient will be added to the waiting list. Later when there is availability of plasma, the recipient will be notified by email. The application seamlessly connects the donor and the recipient. It will create an awareness among the people about donation of plasma which will be done in an easy way of connecting the donor and the recipient. And for sure the patient will be satisfied. Revenue can be generated by selling ad space to health product advertisers and by donation based monetization. Since the app is going to be deploy in a cloud kubernetes cluster, it will continue to be efficient when large number of people uses it. There will be no down time. And it has the ability to increase or decrease IT resources as needed to meet changing demand.

### 3.4 Problem Solution fit



**Fig 1.4.Problem Solution fit**

# REQUIREMENT ANALYSIS

## 4.1 FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposed solution.

<b>FR No.</b>	<b>Functional Requirement (Epic)</b>	<b>Sub Requirement (Story / Sub-Task)</b>
FR-1	User Registration	Registration through Website
FR-2	User Confirmation	Confirmation via Email
FR-3	User Login	Login through registered email id
FR-4	Send Request	Patient should fill their details and make a request
FR-5	Contact Donor	Donor and Patient contact by the details shared via email



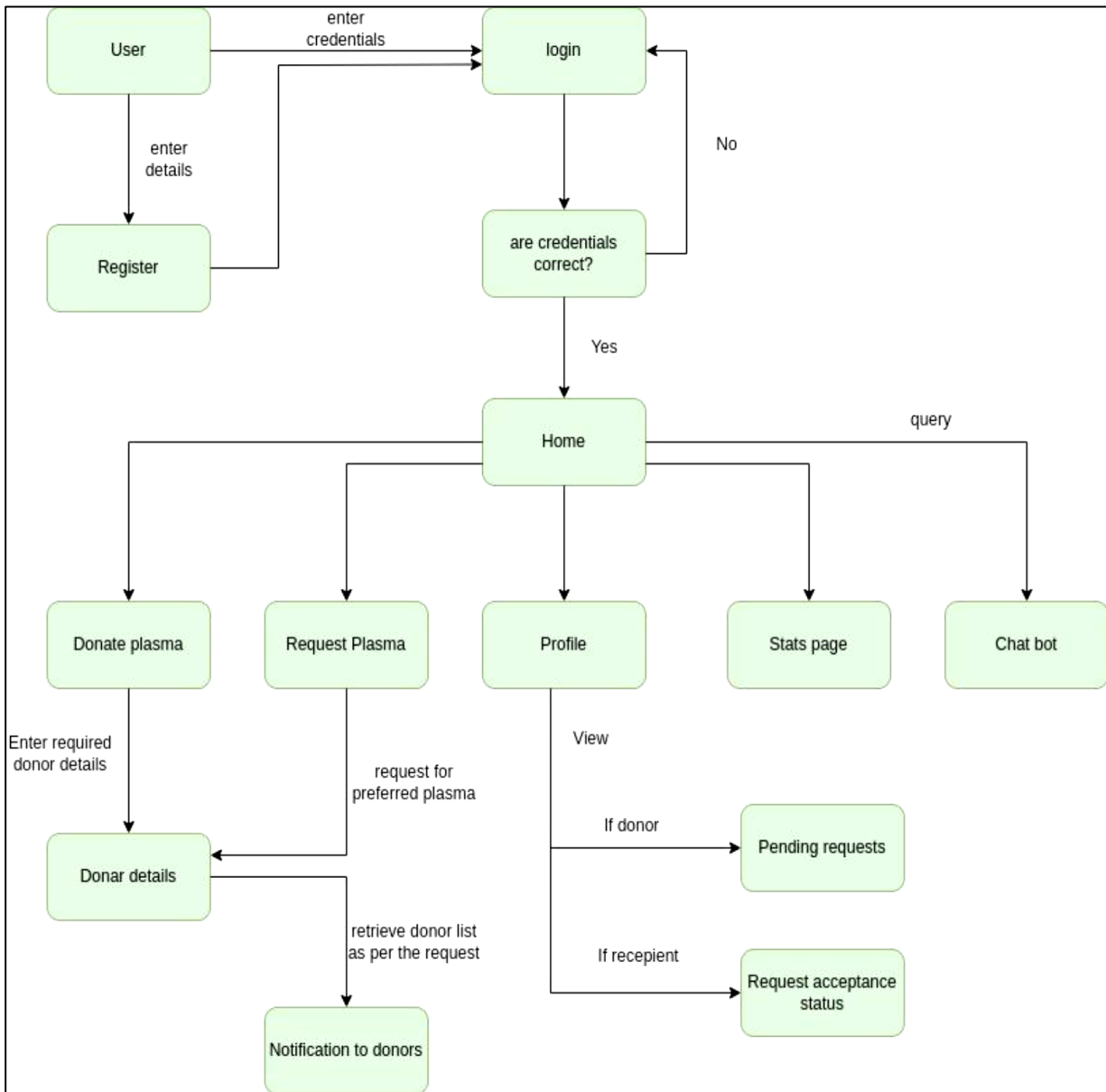
## 4.2 NON-FUNCTIONAL REQUIREMENTS

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

<b>NFR No.</b>	<b>Non Functional Requirement (Epic)</b>	<b>Description</b>
NFR-1	Usability	The plasma Donor application is user friendly and does not involve any complex process
NFR-2	Security	The donor/recipient details are stored in a secured cloud based database.
NFR-3	Reliability	The application will have no down time so that you can always rely on and the information provided by it are so reliable
NFR-4	Performance	The application will work efficiently in emergency situations with an instant notification system.
NFR-5	Availability	The application will be available online 24x7
NFR-6	Scalability	The application can be accessed by multiple users at the same time and it has the ability to increase or decrease the IT resources as needed.

# PROJECT DESIGN

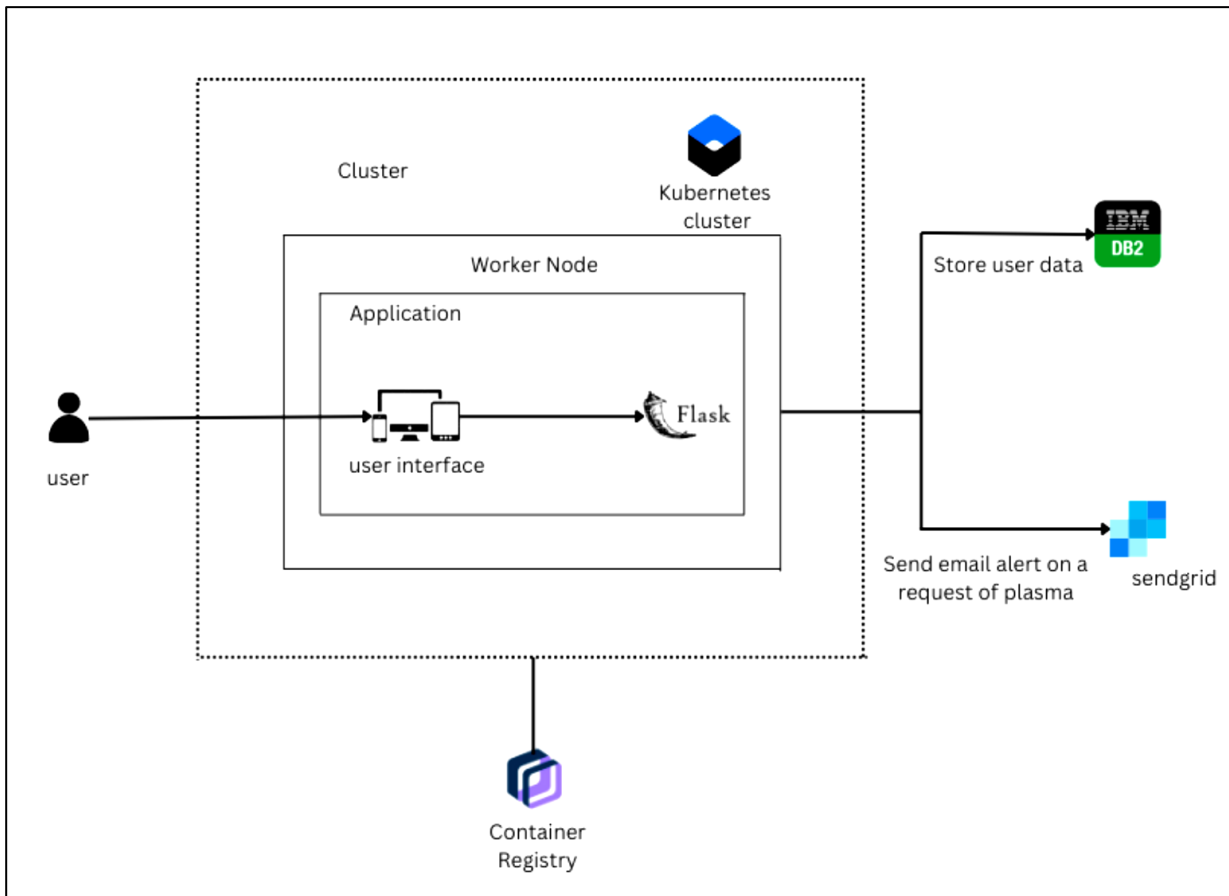
## 5.1 DATA FLOW DIAGRAMS



**Fig 2.1.Data Flow Diagram**

## 5.2 SOLUTION & TECHNICAL ARCHITECTURE

An application should be developed which would take the donor details, store them and notify them upon a request. A user friendly and responsive interface with a quick notification system which instantly notifies the donor upon receiving a request. When the recipient requests for plasma, if there is lack of plasma at the time of request, automatically the recipient will be added to the waiting list. Later when there is availability of plasma, the recipient will be notified by email.



**Fig 2.2. Architecture diagram**

### 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, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail	I can receive confirmation notifications through Gmail	Medium	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email & password	I can access into my User profile and view details in dashboard	High	Sprint-1
	Dashboard	USN-5	As a user, I can donate and request plasma.	I can receive appropriate notifications through email	High	Sprint-1
Customer (Web user)	Login	USN-6	As a user, I can register and log into	I can access into my user profile and	High	Sprint-1

			the application by entering email & password to view the profile	view details in dashboard		
	Dashboard	USN-7	As a user,I can donate and request plasma.	I can receive appropriate notifications through email	High	Sprint-1
Customer Care Executive	Application	USN-8	As a customer care executive, I can try to address user's concerns and questions	I can view and address their concerns and questions	Medium	Sprint-2
Administrator	Application	USN-9	As an administrator, I can listen to feedbacks and make the user interface more friendly and make complex process simple.	I can change	Medium	Sprint-3
		USN-10	As an administrator, I can involve working with the technical side of websites.	I can help with troubleshooting bugs and experience.	Medium	Sprint-1
Chatbot	Dashboard	USN-11	In addition the Customer care executive, chatbot can try to address user's concerns and questions	It can reply to all the queries related to our application	Medium	Sprint-3

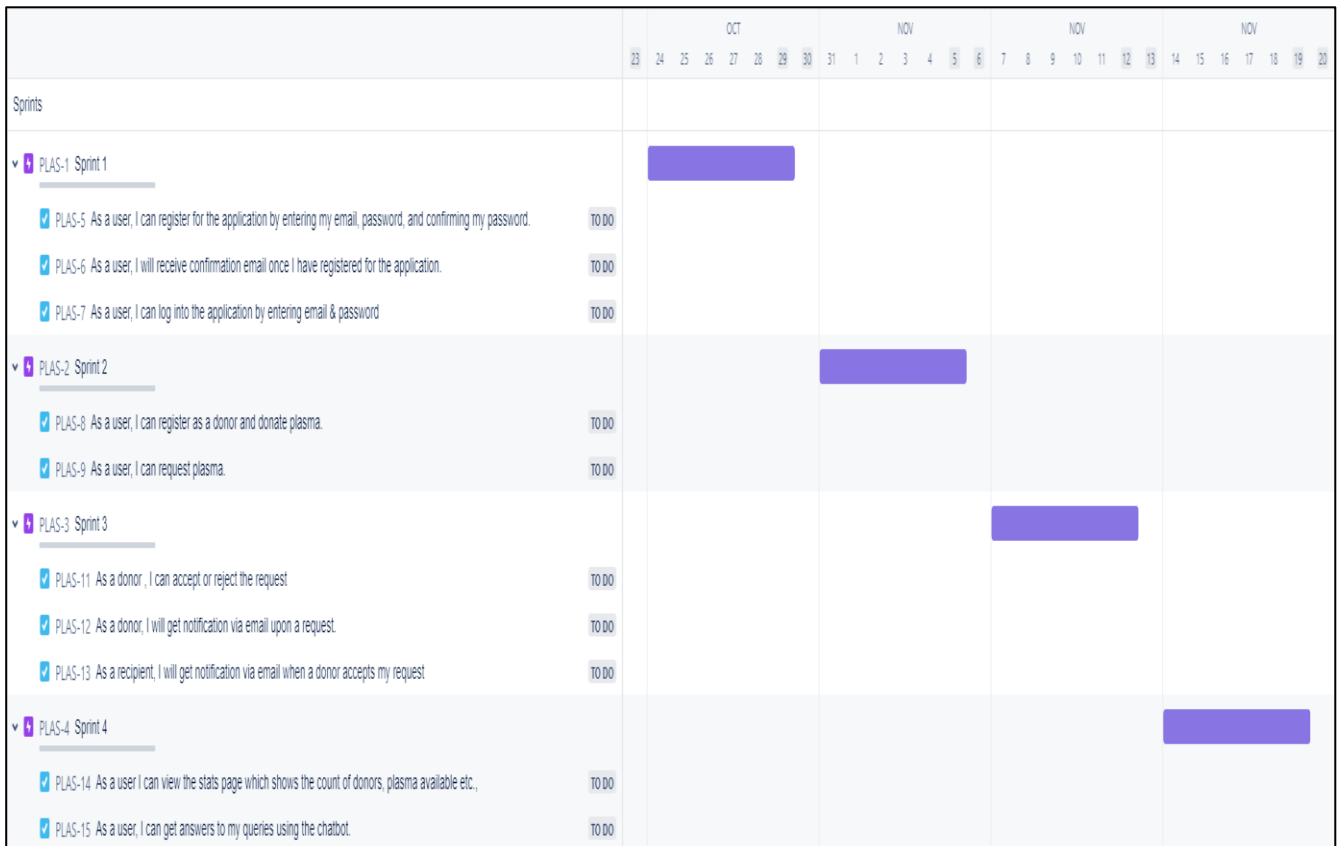
# PROJECT PLANNING & SCHEDULING

## 6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Hari Prasath, Kamalika
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Dhivyadharshini, Devadharshika
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password	2	High	Hari Prasath, Kamalika
Sprint- 2	Dashboard	USN-4	As a user, I can register as a donor and donate plasma	3	High	Hari Prasath, Kamalika
Sprint- 2		USN-5	As a user, I can request plasma	2	High	Dhivyadharshini, Devadharshika
Sprint- 4		USN-6	As a user I can view the stats page which shows the count of donors, plasma available etc.,	2	Medium	Hari Prasath, Kamalika
Sprint- 3		USN-7	As a donor , I can accept or reject the	2	High	Hari Prasath

			request			
Sprint — 4	Chatbot	USN-8	As a user, I can get answers to my queries using the chatbot	2	Medium	Hari Prasath
Sprint-3	Notification	USN-9	As a donor, I will get notification via email upon a request	1	High	Dhivyadharshini
Sprint-3		USN-10	As a recipient, I will get notification once my request is accepted	1	High	Kamalika

## 6.2 SPRINT DELIVERY SCHEDULE



**Fig 3.1. Sprint Delivery Schedule**

## 6.3 REPORTS FROM JIRA

The screenshot displays the Jira Software interface for the 'Plasma Donor Application' project. The main view is the 'Backlog', which lists issues organized into three sprints:

- PLAS Sprint 1** (3 issues):
  - PLAS-5: As a user, I can register for the application by entering my email, password, and confirming my password. (SPRINT 1)
  - PLAS-6: As a user, I will receive confirmation email once I have registered for the application. (SPRINT 1)
  - PLAS-7: As a user, I can log into the application by entering email & password (SPRINT 1)
- PLAS Sprint 2** (2 issues):
  - PLAS-8: As a user, I can register as a donor and donate plasma. (SPRINT 2)
  - PLAS-9: As a user, I can request plasma. (SPRINT 2)
- PLAS Sprint 3** (3 issues):
  - PLAS-11: As a donor, I can accept or reject the request (SPRINT 3)

Each issue entry includes a checkbox, a description, a sprint label, and a 'TO DO' status with a lock icon. The interface also features a 'Create issue' button at the bottom of each sprint section and a 'Start sprint' button for each sprint header.

**Fig 3.2. Sprint report**



# CODING & SOLUTIONING

## 7.1 FEATURE 1 – SIGN UP

### signup.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Sign Up Form by Colorlib</title>

  <!--Font Icon →
  <link rel="stylesheet" href="../static/fonts/material-icon/css/material-design-iconic-
font.min.css">

  <!--Main css →
  <link rel="stylesheet" href="../static/css/register.css">
  <script>
function validateForm() {

  let name = document.forms["form"]["name"].value;
  let email = document.forms["form"]["email"].value;
  let password = document.forms["form"]["pass"].value;
  let re_password = document.forms["form"]["re_pass"].value;

  if(validname(name) && validemail(email) && validpassword(password) &&
validrepassword(re_password, password)){
```

```

    return true;
}
else{
    return false;
}
}

function validname(name){
    console.log("Hello")
    document.getElementById("name_err").innerHTML=""
    if (name != "") {
        var letters = /^[a-zA-Z ]*$/;
        if(name.match(letters))
        {
            return true;
        }
        else
        {
            document.getElementById("name_err").innerHTML="Name should contain only
Alphabets"
            return false;
        }
    }
    else{
        document.getElementById("name_err").innerHTML="Name should not be empty"
        return false;
    }
}

function validemail(email){
    console.log(email)
    document.getElementById("email_err").innerHTML="";
    if (email != "") {
        var emailfor = /^[w+([.-]?w+)*@w+([.-]?w+)*(\.w{2,3})+$/;

```

```

    if(email.match(emailfor))
    {
        return true;
    }
    else
    {
        document.getElementById("email_err").innerHTML="Invalid Email"
        return false;
    }
}
else{
    document.getElementById("email_err").innerHTML="Email Should not be empty"
    return false;
}
}

function validpassword(password){
    document.getElementById("pass_err").innerHTML="";
    if (password != "") {
        var passwordfor = /^(?=.*\d)(?=.*[!@#$%^&*])(?=.*[a-z])(?=.*[A-Z]).{8,}$/;
        if(password.match(passwordfor))
        {
            return true;
        }
        else
        {
            document.getElementById("pass_err").innerHTML="Password should be > 8 letters,
with at least a symbol, upper and lower case letters and a number "
            return false;
        }
    }
    else{
        document.getElementById("pass_err").innerHTML="Passowrd Should not be empty"
        return false;
    }
}

```

```
}
```

```
}
```

```
function validrepassword(re_password, password){
```

```
    document.getElementById("repass_err").innerHTML="";
```

```
    if (re_password != "") {
```

```
        if(password ===re_password)
```

```
        {
```

```
            return true;
```

```
        }
```

```
    else
```

```
    {
```

```
        document.getElementById("repass_err").innerHTML="Password does not match"
```

```
        return false;
```

```
    }
```

```
}
```

```
else{
```

```
    document.getElementById("repass_err").innerHTML="Reenter the password"
```

```
    return false;
```

```
}
```

```
}
```

```
</script>
```

```
</head>
```

```
<body style="background-color: #fef8f5;">
```

```
<div class="main">
```

```
<!--Sign up form →
```

```
<section class="signup">
```

```

<div class="container">
  <div class="signup-content">
    <div class="signup-form">
      <h2 class="form-title">Sign up</h2>
      <form action="{ { url_for('add_user') } }" onsubmit="return validateForm()"
method="POST" class="register-form" id="register-form" name="form">
        <div class="form-group">
          <label for="name"><i class="zmdi zmdi-account material-icons-
name"></i></label>
          <input onkeyup="validname(this.value)" type="text" name="name"
placeholder="Your Name"/>

        </div>
        <div style="margin-top: -15px; margin-bottom: 10px;">
          <label style="position: static; color:red" id="name_err" class="text-danger"
></label>

        </div>

        <div class="form-group">
          <label for="email"><i class="zmdi zmdi-email"></i></label>
          <input onkeyup="validemail(this.value)" type="email" name="email"
id="email" placeholder="Your Email"/>
        </div>
        <div style="margin-top: -15px; margin-bottom: 10px;">
          <label style="position: static; color:red" id="email_err" class="text-danger"
></label>

        </div>

        <div class="form-group">
          <label for="pass"><i class="zmdi zmdi-lock"></i></label>
          <input onkeyup="validpassword(this.value)" type="password"
name="pass" id="pass" placeholder="Password"/>

```

```

    </div>
    <div style="margin-top: -15px; margin-bottom: 10px;">
    <label style="position: static; color:red" id="pass_err" class="text-danger"
></label>
    </div>
    <div class="form-group">
        <label for="re-pass"><i class="zmdi zmdi-lock-outline"></i></label>
        <input onkeyup="validrepassword(this.value,
document.querySelector('#pass').value)" type="password" name="re_pass" id="re_pass"
placeholder="Repeat your password"/>
    </div>
    <div style="margin-top: -15px; margin-bottom: 10px;">
    <label style="position: static; color:red" id="repass_err" class="text-danger"
></label>
    </div>
    <div class="form-group form-button">
        <button name="signup" id="signup" class="form-submit"
style="background-color:#EB5D1E ;" value="Sign up">Sign Up</button>
    </div>
</form>
</div>
<div class="signup-image">
    <figure></figure>
    <a href="/sign_in" class="signup-image-link">I am already member</a>
</div>
</div>
</div>
</section>

</div>

<!--JS -->
<script src="../static/vendor/jquery/jquery.min.js"></script>

```

```
</body><!--This templates was made by Colorlib (https://colorlib.com) →  
</html>
```

## **PYTHON SNIPPET :**

```
@app.route('/add_user', methods=['POST', 'GET'])  
def add_user():  
    if request.method == 'POST':  
        try:  
            name = request.form['name']  
            email = request.form['email']  
            password = request.form['pass']  
            conn = ibm_db.connect(  
                'DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-  
10cf081900bf.c1ogj3sd0tgu0lqde00.databases.appdomain.cloud;PORT=32304;SECURITY=S  
SL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=gfn00031;PWD=LITZUQj2tpFc3t0  
i', '', '')  
            sql = "insert into users values(?,?,?)"  
            param = name, email, password,  
            stmt = ibm_db.prepare(conn, sql)  
            ibm_db.execute(stmt, param)  
            msg = "You're successfully signed up!"  
        except Exception as e:  
            print("exception occurred!",e)  
            msg = e  
  
    finally:  
        return render_template('post_signup.html', msg = msg)
```

## 7.2 FEATURE 2 – SIGN IN

### **signin.html :**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Sign In</title>
  <!--Font Icon →
  <link rel="stylesheet" href="../static/fonts/material-icon/css/material-design-iconic-
font.min.css">
  <!--Bootstrap 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">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
  <!--Main css →
  <link rel="stylesheet" href="../static/css/register.css">

  <script>
    var request;
    function sendInfo(email,password)

    {
```



```

var url="{ { url_for('validate_user') } }?email="+email+"&password="+password;

if(window.XMLHttpRequest){
request=new XMLHttpRequest();
}
else if(window.ActiveXObject){
request=new ActiveXObject("Microsoft.XMLHTTP");
}
try
{
request.onreadystatechange=getInfo;
request.open("GET",url,true);
request.send();
}
catchI
{
alert("Unable to connect to server");
}
}

function getInfo(){
if(request.readyState==4){
var val=request.responseText;
if( val == "User does not exists" || val=="Incorrect Password"){
    console.log(val);
    document.getElementById("loading").style.display = "none"
    document.getElementById('alert').innerHTML=val;
}
else{
    console.log(val);
    document.getElementById('body').innerHTML=val;
}
}
}

```

```

    }
function validateForm() {

    var email = document.forms["form"]["email"].value;
    var password = document.forms["form"]["pass"].value;
    console.log(email+"    "+password)
    if( validemail(email) && validpassword(password)){
        document.getElementById("loading").style.display = "block"
        sendInfo(email,password);

    }
    else{
        return false
    }

}

function validemail(email){
    document.getElementById("email_err").innerHTML="";
    if (email != "") {
        var emailfor = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3}$\\w+$/;
        if(email.match(emailfor))
        {
            return true;
        }
        else
        {
            document.getElementById("email_err").innerHTML="Invalid Email"
            return false;
        }
    }
    else{

        document.getElementById("email_err").innerHTML="Email Should not be empty"
    }
}

```

```

        return false;
    }
}
function validpassword(password){
    document.getElementById("pass_err").innerHTML="";
    if (password != "") {
        return true;
    }
    else{
        document.getElementById("pass_err").innerHTML="Passowrd Should not be empty"
        return false;
    }
}
</script>

</head>
<body id="body" style="background-color: #fef8f5;">
    <div class="main" style="margin-top: -70px;">
        <div id="alert" class="alert alert-light text-center" role="alert" style="color: red;"> </div>
        <!--Sing in Form -->
        <section class="sign-in">
            <div class="container">
                <div class="signin-content">
                    <div class="signin-image">
                        <figure></figure>
                        <a href="/sign_up" class="signup-image-link">Create an account</a>
                    </div>

                    <div class="signin-form">
                        <div class="d-flex ">
                            <div style="display: none;" class="spinner-grow mt-3" id = "loading"
role="status">

                                </div>

```



</div>

<!--JS →

<script src='../static/vendor/jquery/jquery.min.js'></script>

</body><!--This templates was made by Colorlib (<https://colorlib.com>) →

</html>

## PYTHON SNIPPET :

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

```
def validate_user():
```

```
    if request.method == 'GET':
```

```
        try:
```

```
            args = request.args
```

```
            email = args.get('email')
```

```
            password = args.get('password')
```

```
            conn = ibm_db.connect(
```

```
                'DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=gfn00031;PWD=LITZUQj2tpFc3t0i', '', '')
```

```
            sql = 'select * from users where email='+email+''
```

```
            stmt = ibm_db.exec_immediate(conn, sql)
```

```
            dictionary = ibm_db.fetch_assoc(stmt)
```

```
            print("executed")
```

```
            print(dictionary)
```

```
            if dictionary != False:
```

```
                if(dictionary["PASSWORD"]== password):
```

```
                    print("success")
```

```
                    resp = make_response(render_template("post_signin.html"))
```

```

        resp.set_cookie('email', dictionary["EMAIL"])
        resp.set_cookie('name',dictionary["NAME"])
        print("success")
        return resp

    else:
        return "Incorrect Password"

    else:
        return "User does not exists"

except Exception as e :
    print("error",e)
    return reprI

```

## 7.2 FEATURE 3 – DONOR REGISTRATION

### **donor\_registration.html:**

```

<!doctype html>
<html lang="en">
<head>
    <!--Required meta tags →
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link href="https://fonts.googleapis.com/css?family=Roboto:400,700,900&display=swap"
rel="stylesheet">
    <!--Vendor CSS Files →
    <link href="../static/vendor/aos/aos.css" rel="stylesheet">

    <link href="../static/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">

```

```

<link href="../static/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
<link href="../static/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">

<link href="../static/css/style.css" rel="stylesheet">
  <!--Bootstrap 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">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>

  <!--Style →
  <link rel="stylesheet" href="../static/css/request.css">
  <script>
function validateForm() {

  let name = document.forms["form"]["name"].value;
  let email = document.forms["form"]["email"].value;
  let blood_group = document.forms["form"]["blood_group"].value;
  let contact_no = document.forms["form"]["contact_no"].value;
  let city = document.forms["form"]["city"].value;

  if(validname(name) && validemail(email) && validblood_group(blood_group) &&
validcontact_no(contact_no) && validcity(city)){
    return true;
  }
  else{
    return false;
  }
}

```

```

}

function validname(name){
  document.getElementById("name_err").innerHTML=""
  if (name != "") {
    var letters = /^[a-zA-Z ]*$/;
    if(name.match(letters))
    {
      return true;
    }
    else
    {
      document.getElementById("name_err").innerHTML="Name should contain only
Alphabets"
      return false;
    }
  }
  else{
    document.getElementById("name_err").innerHTML="Name should not be empty"
    return false;
  }
}

function validemail(email){
  console.log(email)
  document.getElementById("email_err").innerHTML="";
  if (email != "") {
    var emailfor = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3}+$/;
    if(email.match(emailfor))
    {
      return true;
    }
    else
    {

```



```

        document.getElementById("email_err").innerHTML="Invalid Email"
        return false;
    }
}
else{
    document.getElementById("email_err").innerHTML="Email Should not be empty"
    return false;
}
}
function validblood_group(blood_group){
    document.getElementById("blood_err").innerHTML="";
    if (blood_group == "") {
        document.getElementById("blood_err").innerHTML="Blood group Should not be empty"
        return false;
    }
    else{
        return true;
    }
}

function validcontact_no(contact_no){
    document.getElementById("contact_err").innerHTML="";
    if (contact_no == "") {
        document.getElementById("contact_err").innerHTML="Contact number Should not be
empty"
        return false;
    }
    else{
        return true;
    }
}
}

```

```

function validcity(city){
document.getElementById("city_err").innerHTML="";
if (city == "") {
document.getElementById("city_err").innerHTML="City should not be empty"
return false;
}
else{
return true;
}
}

```

```

</script>
<title>Planor</title>
</head>
<body>
<!-- ===== Header ===== →
<header id="header" class="fixed-top d-flex align-items-center shadow">
<div class="container d-flex align-items-center justify-content-between">

<div class="logo">
<h1 class="text-light"><a href="/"><span>Planor</span></a></h1>
</div>

<nav id="navbar" class="navbar">
<ul>
<li><a class="nav-link scrollto" href="/">Home</a></li>
<li><a class="nav-link scrollto" href="#services">Services</a></li>
<li><a class="nav-link scrollto" href="/request">Request</a></li>
<li><a class="nav-link scrollto" href="/donor_registration">Donate</a></li>
<li><a class="nav-link scrollto" href="/sign_in">Sign In</a></li>
<li><a class="nav-link scrollto" href="/sign_up">Sign Up</a></li>

```

```

        <li><a class="nav-link scrollto" href="/profile">My Profile</a></li>
        <li><a class="nav-link scrollto" href="/logout">Logout</a></li>
    </ul>
    <i class="bi bi-list mobile-nav-toggle"></i>
</nav><!-- .navbar →

</div>
</header><!--End Header →
    {% if logged_in == True %}

<div class="content">

<div class="container">
    <div class="row justify-content-center">
        <div class="col-md-10">

            <div class="row justify-content-center">
                <div class="col-md-6">

                    <h3 class="heading mb-4">Let's help the needy!</h3>

                    <p></p>

                </div>
                <div class="col-md-6">

                    <form action="{ { url_for('add_donor') } }" onsubmit="return validateForm()"
method="post" id="contactForm" name="form">
                        <div class="row">

```

```

    <div class="col-md-12 form-group">
      <input onkeyup="validname(this.value)" type="text" class="form-control mb-3"
name="name" id="name" placeholder="Your name" value={{ name }}>
    </div>
    <div style="margin-top: -15px;">
      <label style="position: static; color:red" id="name_err" class="text-danger"
></label>
    </div>
  </div>

  <div class="row">
    <div class="col-md-12 form-group">
      <input onkeyup="validemail(this.value)" type="text" class="form-control mb-3"
name="email" id="email" placeholder="Email" value = {{ email }}>
    </div>
    <div style="margin-top: -15px;">
      <label style="position: static; color:red" id="email_err" class="text-danger"
></label>
    </div>
  </div>

  <div class="row">
    <div class="col-md-12 form-group">
      <input onkeyup="validblood_group(this.value)" type="text" class="form-control
mb-3" name="blood_group" id="subject" placeholder="Blood group">
    </div>
    <div style="margin-top: -15px;">
      <label style="position: static; color:red" id="blood_err" class="text-danger"
></label>
    </div>
  </div>

  <div class="row">
    <div class="col-md-12 form-group">

```

```

        <input onkeyup="validcontact_no(this.value)" type="text" class="form-control
mb-3" name="contact_no" id="subject" placeholder="Contact number">
    </div>
    <div style="margin-top: -15px;">
        <label style="position: static; color:red" id="contact_err" class="text-danger"
></label>
    </div>
</div>

<div class="row">
    <div class="col-md-12 form-group">
        <input onkeyup="validcity(this.value)" type="text" class="form-control mb-3"
name="city" id="subject" placeholder="City">
    </div>
    <div style="margin-top: -15px;">
        <label style="position: static; color:red" id="city_err" class="text-danger"
></label>
    </div>
</div>

<div class="row">
    <div class="col-12">
        <input type="submit" value="Register as Donor" class="btn btn-primary rounded-
0 py-2 px-4">
        <span class="submitting"></span>
    </div>
</div>
</form>

<div id="form-message-warning mt-4"></div>
<div id="form-message-success">
    Your message was sent, thank you!
</div>

```

```

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

</div>
{% else %}
<div>
    <div class="d-flex flex-column align-items-center justify-content-center " style="height:
400px;">
        Please, Sign in to start donating!
        <a class="btn btn-dark mt-2" style="height: initial;" href="/sign_in">Sign in</a>

    </div>
</div>
{% endif %}
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
</body>
</html>

```

## PYTHON SNIPPET :

```

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

            name = request.form['name']

```

```

email = request.form['email']
blood_group = request.form['blood_group']
contact_no = request.form['contact_no']
location = request.form['city']

conn = ibm_db.connect(
    'DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-
10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=gfn00031;PWD=LITZUQj2tpFc3t0
i', '', '')

sql = "insert into donors values(?,?,?,?)"
param = name, email, blood_group, contact_no, location,
stmt = ibm_db.prepare(conn, sql)
ibm_db.execute(stmt, param)
msg = "You're successfully registered as donor"

except Exception as e:
    print("exception occurred!", e)
    msg = e

finally:
    return render_template('donor_registration_status.html', msg = msg)

```

## 7.2 FEATURE 4 – REQUEST

### request.html :

```

<!doctype html>
<html lang="en">
<head>
<!--Required meta tags -->

```

```

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link href="https://fonts.googleapis.com/css?family=Roboto:400,700,900&display=swap"
rel="stylesheet">
<!--Vendor CSS Files →
<link href="../static/vendor/aos/aos.css" rel="stylesheet">

<link href="../static/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
<link href="../static/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
<link href="../static/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">

<link href="../static/css/style.css" rel="stylesheet">
<!--Bootstrap 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">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>

<!--Style →
<link rel="stylesheet" href="../static/css/request.css">
<script>
function validateForm() {

let name = document.forms["form"]["name"].value;
let email = document.forms["form"]["email"].value;
let blood_group = document.forms["form"]["blood_group"].value;
let contact_no = document.forms["form"]["contact_no"].value;
let city = document.forms["form"]["city"].value;

```



```

    if(validname(name) && validemail(email) && validblood_group(blood_group) &&
validcontact_no(contact_no) && validcity(city)){
        return true;
    }
    else{
        return false;
    }

}

function validname(name){
    document.getElementById("name_err").innerHTML=""
    if (name != "") {
        var letters = /^[a-zA-Z ]*$/;
        if(name.match(letters))
        {
            return true;
        }
        else
        {
            document.getElementById("name_err").innerHTML="Name should contain only
Alphabets"
            return false;
        }
    }
    else{
        document.getElementById("name_err").innerHTML="Name should not be empty"
        return false;
    }
}

function validemail(email){
    console.log(email)
    document.getElementById("email_err").innerHTML="";

```

```

if (email != "") {
    var emailfor = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3}+$/;
    if(email.match(emailfor))
    {
        return true;
    }
    else
    {
        document.getElementById("email_err").innerHTML="Invalid Email"
        return false;
    }
}
else{
    document.getElementById("email_err").innerHTML="Email Should not be empty"
    return false;
}
}

function validblood_group(blood_group){
    document.getElementById("blood_err").innerHTML="";
    if (blood_group == "") {
        document.getElementById("blood_err").innerHTML="Blood group Should not be
empty"
        return false;
    }
    else{
        return true;
    }
}

function validcontact_no(contact_no){
document.getElementById("contact_err").innerHTML="";
if (contact_no == "") {

```

```
document.getElementById("contact_err").innerHTML="Contact number Should not be empty"
```

```
    return false;
}
else{
    return true;
}

}
```

```
function validcity(city){
document.getElementById("city_err").innerHTML="";
if (city == "") {
document.getElementById("city_err").innerHTML="City should not be empty"
return false;
}
else{
return true;
}

}
```

```
</script>
<title>Planor</title>
</head>
<body>
<!-- ===== Header ===== →
<header id="header" class="fixed-top d-flex align-items-center shadow">
<div class="container d-flex align-items-center justify-content-between">

<div class="logo">
<h1 class="text-light"><a href="/"><span>Planor</span></a></h1>
</div>
```

```

<nav id="navbar" class="navbar">
  <ul>
    <li><a class="nav-link scrollto" href="/">Home</a></li>
    <li><a class="nav-link scrollto" href="#services">Services</a></li>
    <li><a class="nav-link scrollto" href="/request">Request</a></li>
    <li><a class="nav-link scrollto" href="/donor_registration">Donate</a></li>
    <li><a class="nav-link scrollto" href="/sign_in">Sign In</a></li>
    <li><a class="nav-link scrollto" href="/sign_up">Sign Up</a></li>
    <li><a class="nav-link scrollto" href="/profile">My Profile</a></li>
    <li><a class="nav-link scrollto" href="/logout">Logout</a></li>
  </ul>
  <i class="bi bi-list mobile-nav-toggle"></i>
</nav><!-- .navbar →

</div>
</header><!--End Header →
  {% if logged_in == True %}
<div class="content">

  <div class="container">
    <div class="row justify-content-center">
      <div class="col-md-10">

        <div class="row justify-content-center">
          <div class="col-md-6">

            <h3 class="heading mb-4">Let's make a request!</h3>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Voluptas debitis, fugit
natus?</p>

            <p></p>

```

```

</div>
<div class="col-md-6">

    <form action="{{ url_for('create_request') }}" onsubmit="return validateForm()"
method="post" id="contactForm" name="form">
    <div class="row">
        <div class="col-md-12 form-group ">
            <input onkeyup="validname(this.value)" type="text" class="form-control mb-3"
name="name" id="name" placeholder="Your name" value="{{ name }}">
        </div>
        <div style="margin-top: -15px;">
            <label style="position: static; color:red" id="name_err" class="text-danger"
></label>
        </div>
    </div>

    <div class="row">
        <div class="col-md-12 form-group">
            <input onkeyup="validemail(this.value)" type="text" class="form-control mb-3"
name="email" id="email" placeholder="Email" value="{{ email }}">
        </div>
        <div style="margin-top: -15px;">
            <label style="position: static; color:red" id="email_err" class="text-danger"
></label>
        </div>
    </div>

    <div class="row">
        <div class="col-md-12 form-group">
            <input onkeyup="validblood_group(this.value)" type="text" class="form-control
mb-3" name="blood_group" id="subject" placeholder="Blood group">
        </div>
    </div>

```

```

    <div style="margin-top: -15px;">
      <label style="position: static; color:red" id="blood_err" class="text-danger"
></label>
    </div>
  </div>

```

```

<div class="row">
  <div class="col-md-12 form-group">
    <input onkeyup="validcontact_no(this.value)" type="text" class="form-control
mb-3" name="contact_no" id="subject" placeholder="Contact number">
  </div>
  <div style="margin-top: -15px;">
    <label style="position: static; color:red" id="contact_err" class="text-danger"
></label>
  </div>
</div>

```

```

<div class="row">
  <div class="col-md-12 form-group">
    <input onkeyup="validcity(this.value)" type="text" class="form-control mb-3"
name="city" id="subject" placeholder="City">
  </div>
  <div style="margin-top: -15px;">
    <label style="position: static; color:red" id="city_err" class="text-danger"
></label>
  </div>
</div>

```

```

<div class="row">
  <div class="col-12">
    <input type="submit" value="Register as Donor" class="btn btn-primary rounded-
0 py-2 px-4">
  </div>
</div>

```

```

        <span class="submitting"></span>
    </div>
</div>
</form>

<div id="form-message-warning mt-4"></div>
<div id="form-message-success">
    Your message was sent, thank you!
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

</div>
{ % else % }
<div>
    <div class="d-flex flex-column align-items-center justify-content-center " style="height:
400px;">
        Please, Sign in to make a request
        <a class="btn btn-dark mt-2" style="height: initial;" href="/sign_in">Sign in</a>

    </div>
</div>
{ % endif % }
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>

</body>

```

</html>

## PYTHON SNIPPET :

```
@app.route('/create_request', methods=['POST', 'GET'])
def create_request():
    if request.method == 'POST':
        try:
            name = request.form['name']
            email = request.form['email']
            blood_group = request.form['blood_group']
            contact_no = request.form['contact_no']
            location = request.form['city']

            conn = ibm_db.connect(
                'DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-
10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;SECURITY=S
SL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=gfn00031;PWD=LITZUQj2tpFc3t0
i', '', '')

            sql = "insert into requests (name, email, blood_group, contact_no, location)
values(?,?,?,?,?)"

            param = name, email, blood_group, contact_no, location,
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.execute(stmt, param)
            msg = "You're successfully made a request!"

        except Exception as e:
            print("exception occurred!", e)
            msg = e

        finally:
            return render_template('donor_registration_status.html', msg = msg)
```



## 7.2 FEATURE 5 – STATUS OF REQUEST

### **profile.html :**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!--Bootstrap 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">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
  <!--Vendor CSS Files →
  <link href="../static/vendor/aos/aos.css" rel="stylesheet">

  <link href="../static/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
  <link href="../static/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
  <link href="../static/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">

  <link href="../static/css/style.css" rel="stylesheet">
  <title>Profile</title>
</head>
<body>
```

```

<!-- ===== Header ===== →
<header id="header" class="fixed-top d-flex align-items-center shadow">
  <div class="container d-flex align-items-center justify-content-between">

    <div class="logo">
      <h1 class="text-light"><a href="/"><span>Planor</span></a></h1>
    </div>

    <nav id="navbar" class="navbar">
      <ul>
        <li><a class="nav-link scrollto" href="/">Home</a></li>
        <li><a class="nav-link scrollto" href="#services">Services</a></li>
        <li><a class="nav-link scrollto" href="/request">Request</a></li>
        <li><a class="nav-link scrollto" href="/donor_registration">Donate</a></li>
        <li><a class="nav-link scrollto" href="/sign_in">Sign In</a></li>
        <li><a class="nav-link scrollto" href="/sign_up">Sign Up</a></li>
        <li><a class="nav-link scrollto" href="/profile">My Profile</a></li>
        <li><a class="nav-link scrollto" href="/logout">Logout</a></li>
      </ul>
      <i class="bi bi-list mobile-nav-toggle"></i>
    </nav><!-- .navbar →

  </div>
</header><!-- End Header →

{% if logged_in == True %}
<div class="container" style="margin-top: 100px;">
  <div class="row">
    <div class="col-sm-4 text-center">
      <h3><b>Hello {{ name }}!</b></h3>
      <p>{{ email }}</p>

```

```

<!-- <div class="text-center" style="background-color: RGB(224, 224, 224); height: 350px;
width:90%;display: flex;flex-direction: column; align-items: center;">
    <div style="margin: auto;">
        <p> Hari Prasath P</p>
        </p>Chennai</p>
        <p>Email:hariprasath.hp5@gmail.com</p>
        <p>Blood Group:O+</p>
    </div>

</div> →
</div>
<div class="col-sm-8">
    <div>
        <div>
            {% if pending_requestslen!=0 %}
            <p>Pending Requests</p>
            {%for i in range(0, pending_requestslen)%}
            <form name="form" method="post" action="{ { url_for('accept_request') } }">
            <div class=" w-100 d-flex justify-content-around align-items-center p-1 mb-3"
style="background-color: RGB(224, 224, 224)" >
                <input type="hidden" value={ { pending_requests[i][ "ID" ] } } name="id">
                <label >Request ID :{ { pending_requests[i][ "ID" ] } }</label>
                <label >Name :{ { pending_requests[i][ "NAME" ] } }</label>
                <label >Email :{ { accepted_requests[i][ "EMAIL" ] } }</label>
                <label>Location:{ { pending_requests[i][ "LOCATION" ] } }</label>
                <label>Blood Group:{ { pending_requests[i][ "BLOOD_GROUP" ] } }</label>
                <input type="submit" class="btn btn-primary" value="Accept">
            </div>
        </form>
        {%endfor%}
        {% endif %}
    </div>

```

```

<div>
    {% if accepted_requestslen!=0 %}
    <p>Accepted Requests</p>
    {%for i in range(0, accepted_requestslen)%}
    <div class=" w-100 d-flex justify-content-around align-items-center p-1 mb-3"
style="background-color: RGB(224, 224, 224)" >
        <label >Request ID :{{ accepted_requests[i]["ID"]}}</label>
        <label >Name :{{ accepted_requests[i]["NAME"]}}</label>
        <label >Email :{{ accepted_requests[i]["EMAIL"]}}</label>
        <label>Location:{{ accepted_requests[i]["LOCATION"]}}</label>
        <label>Blood Group:{{ accepted_requests[i]["BLOOD_GROUP"]}}</label>
    </div>
    {%endfor%}
    {% endif %}
</div>

<div>
    {% if requests_len!=0 %}
    <p>Requests made</p>
    {%for i in range(0, requests_len)%}
    <div class=" w-75 d-flex justify-content-around align-items-center p-2 mb-3"
style="background-color: RGB(224, 224, 224)" >
        <label >Request ID :{{ requests[i]["ID"]}}</label>
        <label >Name :{{ requests[i]["NAME"]}}</label>
        <label>Location:{{ requests[i]["LOCATION"]}}</label>
        <label>Blood Group:{{ requests[i]["BLOOD_GROUP"]}}</label>
        <label>Status: {{ requests[i]["REQUEST_STATUS"]}}</label>
    </div>
    {%endfor%}
    {% endif %}
</div>

</div>

```

</div>

</div>

</div>

{% else %}

<div>

<div class="d-flex flex-column align-items-center justify-content-center" style="height: 400px;">

Please, Sign in to view your profile

<a class="btn btn-dark mt-2" style="height: initial;" href="/sign\_in">Sign in</a>

</div>

</div>

{% endif %}

</body>

</html>

## PYTHON SNIPPET :

```
@app.route('/profile')
def profile():
    email = request.cookies.get('email')
    name = request.cookies.get('name')
    if email != None:
        conn = ibm_db.connect(
            'DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-
10cf081900bf.c1logj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=gfn00031;PWD=LITZUQj2tpFc3t0i', '', '')
        sql = 'select * from requests where email='+email+''
```

```

stmt = ibm_db.exec_immediate(conn, sql)
requests = []
dictionary = ibm_db.fetch_assoc(stmt)

while dictionary != False:
    print(dictionary["ID"])
    requests.append(dictionary)
    dictionary = ibm_db.fetch_assoc(stmt)
print(requests)

sql = 'select * from donors where email='+'\'+email+'\''
stmt = ibm_db.exec_immediate(conn, sql)
dictionary = ibm_db.fetch_assoc(stmt)
isDonor = False
pending_requests = []
if dictionary != False:
    isDonor = True
    donor_location = dictionary["LOCATION"]
    donor_bloodgroup = dictionary["BLOOD_GROUP"]
    sql = "select * from requests where
blood_group='"+donor_bloodgroup+"'"+"and location='"+donor_location+"'"+"and
request_status='"+Pending'"
    stmt = ibm_db.exec_immediate(conn, sql)

    dictionary = ibm_db.fetch_assoc(stmt)
    while dictionary != False:
        pending_requests.append(dictionary)
        dictionary = ibm_db.fetch_assoc(stmt)
    print(pending_requests)

accepted_requests= []
if isDonor:
    sql = 'select * from requests where accepted_by='+'\'+email+'\''

```

```

stmt = ibm_db.exec_immediate(conn, sql)
dictionary = ibm_db.fetch_assoc(stmt)

while dictionary != False:
    accepted_requests.append(dictionary)
    dictionary = ibm_db.fetch_assoc(stmt)
print(accepted_requests)

return render_template('profile.html', name =name, email = email,requests_len =
len(requests) ,requests = requests, pending_requestslen = len(pending_requests),
pending_requests = pending_requests, accepted_requestslen = len(accepted_requests),
accepted_requests = accepted_requests, logged_in=True)

else:
    return render_template('profile.html', logged_in= False)

```

# TESTING

## 8.1 TEST CASES

| 1  | Test Cases  | Result   |
|----|---|----------|
| 2  | Verify the user is able to see the Sign up page when the user clicks the signup button in navigation bar          | Positive |
| 3  | Verify the UI elements in the Sign up page  | Positive |
| 4  | Verify the user is able to register into the application by providing valid details                               | Positive |
| 5  | Verify the user is able to see the sign in page when the user clicks the signin button in navigation bar          | Positive |
| 6  | Verify the UI elements in the Sign in page  | Positive |
| 7  | Verify the user is able to login into the application by providing valid details                                  | Positive |
| 8  | Verify the user is able to see the Donor registration page when the user clicks the donate link in navigation bar | Positive |
| 9  | Verify the UI elements in the Donor Registration page   | Positive |
| 10 | Verify the user is able to register as a donor by providing valid details   | Positive |
| 11 | Verify the user is able to see the request page when the user clicks the request link in navigation bar           | Positive |
| 12 | Verify the UI elements in the request page  | Positive |
| 13 | Verify the user is able to make a request by providing valid details  | Positive |
| 14 | Verify the user gets a email notification when they sign up   | Positive |
| 15 | Verify the donor gets a email notification when they make a request   | Positive |
| 16 | Verify the donor and recipient gets a email notification when the donor accepts the request                       | Positive |
| 17 | Verify the user is able to see the stats page when the user clicks the stage page link in navigation bar          | Positive |
| 18 | Verify the user is able to interact with the chatbot  | Positive |



## 8.2 USER ACCEPTANCE TESTING

| 1 | Test case ID      | Feature Type | Component    | Test Scenario  | Steps to Execute  |
|---|-------------------|--------------|--------------|--|---|
| 2 | SignUpPage_TC_001 | Functional   | Sign Up page | Verify the user is able to see the Sign up page when the user clicks the signup button in navigation bar | 1. Enter the url and go<br>2. Click the sign up link in the navigation bar.<br>3. Verify the sign up page is visible or not.  |
| 3 | SignUpPage_TC_002 | UI           | Sign Up page | Verify the UI elements in the Sign up page   | 1. Enter the url and go<br>2. Click the sign up link in the navigation bar.<br>3. Verify the below mentioned ui elements:<br>a. name text box<br>b. email text box.<br>c. password text box.<br>d. repeat password text box.<br>e. sign up button |
| 4 | SignUpPage_TC_003 | Functional   | Sign Up page | Verify the user is able to register into the application by providing valid details                      | 1. Enter the url and go<br>2. Click the sign up link in the navigation bar.<br>3. Enter valid details in the text boxes.<br>4. Verify the confirmation message.   |
| 5 | SignInPage_TC_001 | Functional   | Sign In page | Verify the user is able to see the sign in page when the user clicks the signin button in navigation bar | 1. Enter the url and go<br>2. Click the sign in link in the navigation bar.<br>3. Verify the sign in page is visible or not.  |
| 6 | SignInPage_TC_002 | UI           | Sign In page | Verify the UI elements in the Sign in page   | 1. Enter the url and go<br>2. Click the sign in link in the navigation bar.<br>3. Verify the below mentioned ui elements:<br>a. email text box.<br>b. password text box.<br>c. sign in button   |

|    |                              |            |                         |   |  |
|----|------------------------------|------------|-------------------------|---|--|
| 6  | SignInPage_TC_002            | UI         | Sign In page            | Verify the UI elements in the Sign in page  | 1. Enter the url and go<br>2. Click the sign in link in the navigation bar.<br>3. Verify the below mentioned ui elements:<br>a. email text box.<br>b. password text box.<br>c. sign in button  |
| 7  | SignInPage_TC_003            | Functional | Sign In page            | Verify the user is able to login into the application by providing valid details                                  | 1. Enter the url and go<br>2. Click the sign in link in the navigation bar.<br>3. Enter valid details in the text boxes.<br>4. Verify the user is able to login.   |
| 8  | DonorRegistrationPage_TC_001 | Functional | Donor Registration Page | Verify the user is able to see the Donor registration page when the user clicks the donate link in navigation bar | 1. Enter the url and go<br>2. Click the donate link in the navigation bar.<br>3. Verify the donor registration page is visible or not.   |
| 9  | DonorRegistrationPage_TC_002 | UI         | Donor Registration Page | Verify the UI elements in the Donor Registration page   | 1. Enter the url and go<br>2. Click the donate link in the navigation bar.<br>3. Verify the below mentioned ui elements:<br>a. name text box<br>b. email text box.<br>c. blood group text box.<br>d. contact number text box.<br>e. city text box<br>f. register as donor button |
| 10 | DonorRegistrationPage_TC_003 | Functional | Donor Registration Page | Verify the user is able to register as a donor by providing valid details   | 1. Enter the url and go<br>2. Click the donate link in the navigation bar.<br>3. Enter valid details in the text boxes.<br>4. Click the donate button.<br>4. Verify the user is able to register as a donor successfully   |

|    |                    |            |              |   |  |
|----|--------------------|------------|--------------|---|--|
| 11 | RequestPage_TC_001 | Functional | Request Page | Verify the user is able to see the request page when the user clicks the request link in navigation bar | 1. Enter the url and go<br>2. Click the request link in the navigation bar.<br>3. Verify the request page is visible or not.   |
| 12 | RequestPage_TC_002 | UI         | Request Page | Verify the UI elements in the request page  | 1. Enter the url and go<br>2. Click the request link in the navigation bar.<br>3. Verify the below mentioned ui elements:<br>a. name text box<br>b. email text box.<br>c. blood group text box.<br>d. contact number text box.<br>e. city text box<br>f. make a request button |
| 13 | RequestPage_TC_003 | Functional | Request Page | Verify the user is able to make a request by providing valid details                                    | 1. Enter the url and go<br>2. Click the request link in the navigation bar.<br>3. Enter valid details in the text boxes.<br>4. Click the request button.<br>4. Verify the user is able to make a request successfully.   |
| 14 | Notication_TC_001  | Functional | Sign up page | Verify the user gets a email notification when they sign up   | 1. Enter the url and go<br>2. Go to the sign up page.<br>3. Enter the details and click sign up button<br>4. Verify if they get the email on successfull sign up   |
| 15 | Notication_TC_002  | Functional | Request Page | Verify the donor gets a email notification when they make a request                                     | 1. Enter the url and go<br>2. Go to the request page.<br>3. Enter the details and click make a request button<br>4. Verify if the donor gets the email on successfully maki request.   |

|    |                   |            |              |  |  |
|----|-------------------|------------|--------------|--|--|
| 16 | Notication_TC_003 | Functional | Profile Page | Verify the donor and recipient gets a email notification when the donor accepts the request              | 1. Enter the url and go<br>2. Go to the profile page<br>3. Accept the pending request.<br>4. Verify if they get the email containing contact details |
| 17 | StatsPage_TC_001  | Functional | Stats Page   | Verify the user is able to see the stats page when the user clicks the stage page link in navigation bar | 1. Enter the url and go<br>2. Click the stats page link in the navigation bar.<br>3. Verify the stats page is visible or not                         |
| 18 | Chatbot_TC_001    | Functional | Home Page    | Verify the user is able to interact with the chatbot   | 1. Enter the url and go<br>2. Click the chatbot icon in the home page<br>3. Verify the chatbot is working or not                                     |

# RESULTS

## 9.1 PERFORMANCE METRICS

Web application performance metrics help determine certain aspects that impact the performance of an application. There are eight key metrics, including: User Satisfaction—also known as Apdex Scores, uses a mathematical formula in order to determine user satisfaction.

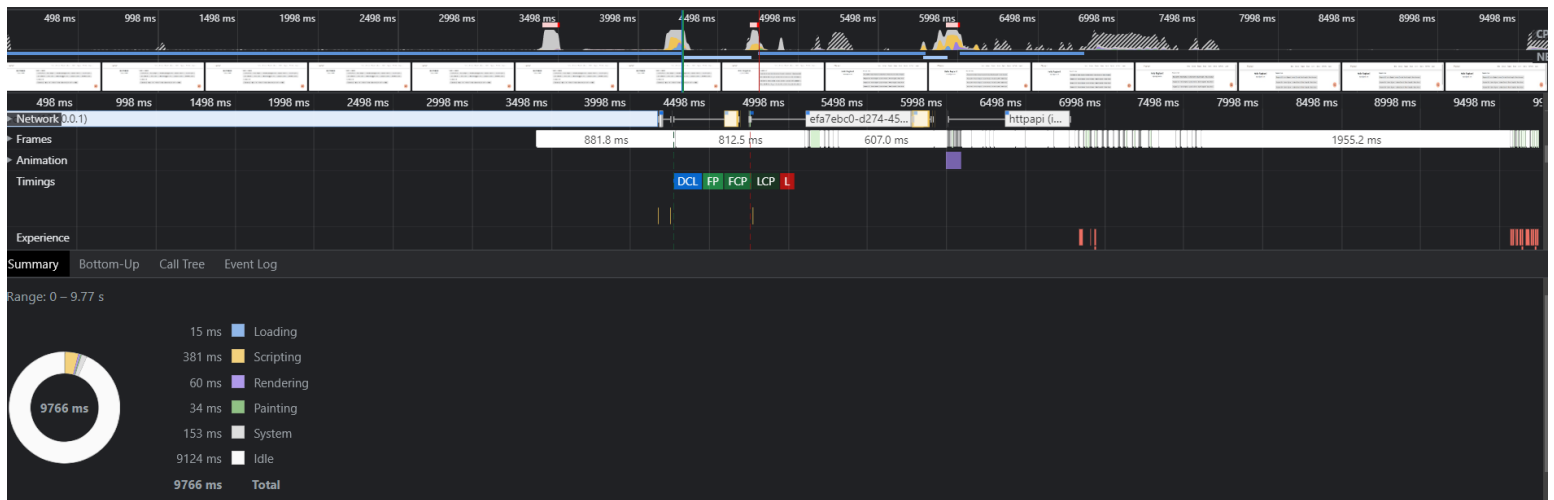


Fig 4.1. Performance Metrics

## **ADVANTAGES & DISADVANTAGES**

### **ADVANTAGES**

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

### **DISADVANTAGES**

- It requires an active internet connection.
- It relays on the details provided by the user.

## CONCLUSION

Plasma is a liquid portion of blood; it is a mixture of water, proteins and salts.

Antibodies are proteins made by the body in response to an infection. People fully rescued from COVID19 are encouraged to donate plasma, which can help to increase the lifespan of other patients because their plasma contains antigens which helps the affected person to recover faster. These immunoglobulin give your immune system a way to fight the virus when you are sick, so your plasma can be used to help others fight off illness. Individuals must fully resolve symptoms for at least 14 days prior are eligible to donate. Enhanced mobile application for plasma has been developed to help the administrator to attract more donors and recipients and make user management an easy task. This mobile application will attract more users as it is user friendly and greatly reduces scalability issues. Thus, we have successfully designed and developed the Android mobile application to ease the process of becoming a donor and recipient of PMB bank.

## **FUTURE SCOPE**

- **A chat widget to establish communication between a donor and recipient .**
- **To attract more users android application should also be developed in future.**

# APPENDIX

## SOURCE CODE

### signup.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Sign Up Form by Colorlib</title>

  <!-- Font Icon -->
  <link rel="stylesheet" href="../static/fonts/material-icon/css/material-design-iconic-
font.min.css">

  <!-- Main css -->
  <link rel="stylesheet" href="../static/css/register.css">
  <script>
function validateForm() {

  let name = document.forms["form"]["name"].value;
  let email = document.forms["form"]["email"].value;
  let password = document.forms["form"]["pass"].value;
  let re_password = document.forms["form"]["re_pass"].value;

  if(validname(name) && validemail(email) && validpassword(password) &&
validrepassword(re_password, password)){
```

```

    return true;
}
else{
    return false;
}
}

function validname(name){
    console.log("Hello")
    document.getElementById("name_err").innerHTML=""
    if (name != "") {
        var letters = /^[a-zA-Z ]*$/;
        if(name.match(letters))
        {
            return true;
        }
        else
        {
            document.getElementById("name_err").innerHTML="Name should contain only
Alphabets"
            return false;
        }
    }
    else{
        document.getElementById("name_err").innerHTML="Name should not be empty"
        return false;
    }
}

function validemail(email){
    console.log(email)
    document.getElementById("email_err").innerHTML="";
    if (email != "") {
        var emailfor = /^[w+([.-]?w+)*@w+([.-]?w+)*(\.w{2,3})+$/;
        if(email.match(emailfor))

```



```

    {
    return true;
    }
    else
    {
    document.getElementById("email_err").innerHTML="Invalid Email"
    return false;
    }
}
else{
    document.getElementById("email_err").innerHTML="Email Should not be empty"
    return false;
}
}
function validpassword(password){
    document.getElementById("pass_err").innerHTML="";
    if (password != "") {
        var passwordfor = /^(?=.*\d)(?=.*[!@#$%^&*])(?=.*[a-z])(?=.*[A-Z]).{8,}$/;
        if(password.match(passwordfor))
        {
            return true;
        }
        else
        {
            document.getElementById("pass_err").innerHTML="Password should be > 8 letters,
with at least a symbol, upper and lower case letters and a number "
            return false;
        }
    }
}
else{
    document.getElementById("pass_err").innerHTML="Passowrd Should not be empty"
    return false;
}
}

```

```

}
function validrepassword(re_password, password){

document.getElementById("repass_err").innerHTML="";
if (re_password != "") {
    if(password ===re_password)
    {
        return true;
    }
    else
    {
        document.getElementById("repass_err").innerHTML="Password does not match"
        return false;
    }
}
else{
    document.getElementById("repass_err").innerHTML="Reenter the password"
    return false;
}

}
</script>
</head>
<body style="background-color: #fef8f5;">

<div class="main">

    <!-- Sign up form -->
    <section class="signup">
        <div class="container">
            <div class="signup-content">
                <div class="signup-form">

```

```

        <h2 class="form-title" >Sign up</h2>
        <form action="{ { url_for('add_user') } }" onsubmit="return validateForm()"
method="POST" class="register-form" id="register-form" name="form">
            <div class="form-group">
                <label for="name"><i class="zmdi zmdi-account material-icons-
name"></i></label>
                    <input onkeyup="validname(this.value)" type="text" name="name"
placeholder="Your Name"/>

            </div>
            <div style="margin-top: -15px; margin-bottom: 10px;">
                <label style="position: static; color:red" id="name_err" class="text-danger"
></label>

            </div>

            <div class="form-group">
                <label for="email"><i class="zmdi zmdi-email"></i></label>
                <input onkeyup="validemail(this.value)" type="email" name="email"
id="email" placeholder="Your Email"/>
            </div>
            <div style="margin-top: -15px; margin-bottom: 10px;">
                <label style="position: static; color:red" id="email_err" class="text-danger"
></label>

            </div>

            <div class="form-group">
                <label for="pass"><i class="zmdi zmdi-lock"></i></label>
                <input onkeyup="validpassword(this.value)" type="password" name="pass"
id="pass" placeholder="Password"/>
            </div>
            <div style="margin-top: -15px; margin-bottom: 10px;">

```

```

        <label style="position: static; color:red" id="pass_err" class="text-danger"
></label>

    </div>

    <div class="form-group">
        <label for="re-pass"><i class="zmdi zmdi-lock-outline"></i></label>
        <input onkeyup="validrepassword(this.value,
document.querySelector('#pass').value)" type="password" name="re_pass" id="re_pass"
placeholder="Repeat your password"/>
    </div>

    <div style="margin-top: -15px; margin-bottom: 10px;">
        <label style="position: static; color:red" id="repass_err" class="text-danger"
></label>

    </div>

    <div class="form-group form-button">
        <button name="signup" id="signup" class="form-submit"
style="background-color:#EB5D1E ;" value="Sign up">Sign Up</button>
    </div>
</form>
</div>
<div class="signup-image">
    <figure></figure>
    <a href="/sign_in" class="signup-image-link">I am already member</a>
</div>
</div>
</div>
</section>

</div>

<!-- JS -->
<script src="../../static/vendor/jquery/jquery.min.js"></script>

</body><!-- This templates was made by Colorlib (https://colorlib.com) -->

```

</html>

## **signin.html :**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Sign In</title>
  <!-- Font Icon -->
  <link rel="stylesheet" href="../static/fonts/material-icon/css/material-design-iconic-
font.min.css">
  <!-- Bootstrap 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">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
  <!-- Main css -->
  <link rel="stylesheet" href="../static/css/register.css">

  <script>
    var request;
    function sendInfo(email,password)

    {
```

```

var url="{ { url_for('validate_user') } }?email="+email+"&password="+password;

if(window.XMLHttpRequest){
request=new XMLHttpRequest();
}
else if(window.ActiveXObject){
request=new ActiveXObject("Microsoft.XMLHTTP");
}
try
{
request.onreadystatechange=getInfo;
request.open("GET",url,true);
request.send();
}
catch(e)
{
alert("Unable to connect to server");
}
}

function getInfo(){
if(request.readyState==4){
var val=request.responseText;
if( val == "User does not exists" || val=="Incorrect Password"){
    console.log(val);
    document.getElementById("loading").style.display = "none"
    document.getElementById('alert').innerHTML=val;
}
else{
    console.log(val);
    document.getElementById('body').innerHTML=val;
}
}
}

```

```

    }
function validateForm() {

    var email = document.forms["form"]["email"].value;
    var password = document.forms["form"]["pass"].value;
    console.log(email+"    "+password)
    if( validemail(email) && validpassword(password)){
        document.getElementById("loading").style.display = "block"
        sendInfo(email,password);

    }
    else{
        return false
    }

}

function validemail(email){
    document.getElementById("email_err").innerHTML="";
    if (email != "") {
        var emailfor = /^^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/;
        if(email.match(emailfor))
        {
            return true;
        }
        else
        {
            document.getElementById("email_err").innerHTML="Invalid Email"
            return false;
        }
    }
}
else{

    document.getElementById("email_err").innerHTML="Email Should not be empty"

```

```

        return false;
    }
}
function validpassword(password){
    document.getElementById("pass_err").innerHTML="";
    if (password != "") {
        return true;
    }
    else{
        document.getElementById("pass_err").innerHTML="Passowrd Should not be empty"
        return false;
    }
}
</script>

</head>
<body id="body" style="background-color: #fef8f5;">
    <div class="main" style="margin-top: -70px;">
        <div id="alert" class="alert alert-light text-center" role="alert" style="color: red;"> </div>
        <!-- Sing in Form -->
        <section class="sign-in">
            <div class="container">
                <div class="signin-content">
                    <div class="signin-image">
                        <figure></figure>
                        <a href="/sign_up" class="signup-image-link">Create an account</a>
                    </div>

                    <div class="signin-form">
                        <div class="d-flex ">
                            <div style="display: none;" class="spinner-grow mt-3" id = "loading"
role="status">

                                </div>

```



```

        <h2 class="form-title">Sign in</h2>

    </div>

    <form name="form" >
        <div class="form-group">
            <label for="email"><i class="zmdi zmdi-account material-icons-
name"></i></label>
                <input onkeyup="validemail(this.value)" type="text" name="email"
id="email" placeholder="Your email"/>
            </div>
            <div style="margin-top: -15px; margin-bottom: 10px;">
                <label style="position: static; color:red" id="email_err" class="text-danger"
></label>
            </div>
            <div class="form-group">
                <label for="your_pass"><i class="zmdi zmdi-lock"></i></label>
                <input onkeyup="validpassword(this.value)" type="password"
name="pass" id="pass" placeholder="Password"/>
            </div>
            <div style="margin-top: -15px; margin-bottom: 10px;">
                <label style="position: static; color:red" id="pass_err" class="text-danger"
></label>
            </div>
            </div>
        </form>
        <div class="form-group form-button">
            <button onclick="validateForm()" name="signin" id="signin" class="form-
submit" style="background-color:#EB5D1E ;">Sign In</button>
        </div>
    </div>
</div>
</div>
</div>
</section>

```

```

</div>

<!-- JS -->
<script src="../static/vendor/jquery/jquery.min.js"></script>
</body><!-- This templates was made by Colorlib (https://colorlib.com) -->
</html>

```

### **donor\_registration.html:**

```

<!doctype html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <link href="https://fonts.googleapis.com/css?family=Roboto:400,700,900&display=swap"
rel="stylesheet">
  <!-- Vendor CSS Files -->
  <link href="../static/vendor/aos/aos.css" rel="stylesheet">

  <link href="../static/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
  <link href="../static/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
  <link href="../static/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">

  <link href="../static/css/style.css" rel="stylesheet">

  <!-- Bootstrap 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">

```

```

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>

```

```

<!-- Style -->
<link rel="stylesheet" href="../static/css/request.css">
<script>
function validateForm() {

    let name = document.forms["form"]["name"].value;
    let email = document.forms["form"]["email"].value;
    let blood_group = document.forms["form"]["blood_group"].value;
    let contact_no = document.forms["form"]["contact_no"].value;
    let city = document.forms["form"]["city"].value;

    if(validname(name) && validemail(email) && validblood_group(blood_group) &&
validcontact_no(contact_no) && validcity(city)){
        return true;
    }
    else{
        return false;
    }

}

function validname(name){
    document.getElementById("name_err").innerHTML=""
    if (name != "") {
        var letters = /^[a-zA-Z]*$/;
        if(name.match(letters))
        {
            return true;

```

```

    }
    else
    {
        document.getElementById("name_err").innerHTML="Name should contain only
Alphabets"
        return false;
    }
}
else{
    document.getElementById("name_err").innerHTML="Name should not be empty"
    return false;
}
}
function validemail(email){
    console.log(email)
    document.getElementById("email_err").innerHTML="";
    if (email != "") {
        var emailfor = /^[^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3})+$/;
        if(email.match(emailfor))
        {
            return true;
        }
        else
        {
            document.getElementById("email_err").innerHTML="Invalid Email"
            return false;
        }
    }
    else{
        document.getElementById("email_err").innerHTML="Email Should not be empty"
        return false;
    }
}
}

```

```

function validblood_group(blood_group){
    document.getElementById("blood_err").innerHTML="";
    if (blood_group == "") {
        document.getElementById("blood_err").innerHTML="Blood group Should not be empty"
        return false;
    }
    else{
        return true;
    }
}

```

```

function validcontact_no(contact_no){
    document.getElementById("contact_err").innerHTML="";
    if (contact_no == "") {
        document.getElementById("contact_err").innerHTML="Contact number Should not be
empty"
        return false;
    }
    else{
        return true;
    }
}

```

```

function validcity(city){
    document.getElementById("city_err").innerHTML="";
    if (city == "") {
        document.getElementById("city_err").innerHTML="City should not be empty"
        return false;
    }
    else{
        return true;
    }
}

```

```

    }

}

</script>
<title>Planor</title>
</head>
<body>
    <!-- ===== Header ===== -->
<header id="header" class="fixed-top d-flex align-items-center shadow">
    <div class="container d-flex align-items-center justify-content-between">

        <div class="logo">
            <h1 class="text-light"><a href="/"><span>Planor</span></a></h1>
        </div>

        <nav id="navbar" class="navbar">
            <ul>
                <li><a class="nav-link scrollto" href="/">Home</a></li>
                <li><a class="nav-link scrollto" href="#services">Services</a></li>
                <li><a class="nav-link scrollto" href="/request">Request</a></li>
                <li><a class="nav-link scrollto" href="/donor_registration">Donate</a></li>
                <li><a class="nav-link scrollto" href="/sign_in">Sign In</a></li>
                <li><a class="nav-link scrollto" href="/sign_up">Sign Up</a></li>
                <li><a class="nav-link scrollto" href="/profile">My Profile</a></li>
                <li><a class="nav-link scrollto" href="/logout">Logout</a></li>
            </ul>
            <i class="bi bi-list mobile-nav-toggle"></i>
        </nav><!-- .navbar -->

    </div>
</header><!-- End Header -->

    { % if logged_in == True % }

```

```

<div class="content">

<div class="container">
  <div class="row justify-content-center">
    <div class="col-md-10">

<div class="row justify-content-center">
  <div class="col-md-6">

    <h3 class="heading mb-4">Let's help the needy!</h3>

    <p></p>

  </div>
  <div class="col-md-6">

    <form action="{ { url_for('add_donor') } }" onsubmit="return validateForm()"
method="post" id="contactForm" name="form">
    <div class="row">
      <div class="col-md-12 form-group ">
        <input onkeyup="validname(this.value)" type="text" class="form-control mb-3"
name="name" id="name" placeholder="Your name" value={ { name } }>
      </div>
      <div style="margin-top: -15px;">
        <label style="position: static; color:red" id="name_err" class="text-danger"
></label>
      </div>
    </div>
  </div>

```

```

<div class="row">
  <div class="col-md-12 form-group">
    <input onkeyup="validemail(this.value)" type="text" class="form-control mb-3"
name="email" id="email" placeholder="Email" value = {{email}}>
  </div>
  <div style="margin-top: -15px;">
    <label style="position: static; color:red" id="email_err" class="text-danger"
></label>
  </div>
</div>
<div class="row">
  <div class="col-md-12 form-group">
    <input onkeyup="validblood_group(this.value)" type="text" class="form-control
mb-3" name="blood_group" id="subject" placeholder="Blood group">
  </div>
  <div style="margin-top: -15px;">
    <label style="position: static; color:red" id="blood_err" class="text-danger"
></label>
  </div>
</div>
<div class="row">
  <div class="col-md-12 form-group">
    <input onkeyup="validcontact_no(this.value)" type="text" class="form-control mb-
3" name="contact_no" id="subject" placeholder="Contact number">
  </div>
  <div style="margin-top: -15px;">
    <label style="position: static; color:red" id="contact_err" class="text-danger"
></label>
  </div>
</div>
<div class="row">

```



```

        <div class="col-md-12 form-group">
            <input onkeyup="validcity(this.value)" type="text" class="form-control mb-3"
name="city" id="subject" placeholder="City">
        </div>
        <div style="margin-top: -15px;">
            <label style="position: static; color:red" id="city_err" class="text-danger" ></label>
        </div>
    </div>

<div class="row">
    <div class="col-12">
        <input type="submit" value="Register as Donor" class="btn btn-primary rounded-0
py-2 px-4">
        <span class="submitting"></span>
    </div>
</div>
</form>


<div id="form-message-warning mt-4"></div>
<div id="form-message-success">
    Your message was sent, thank you!
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
{ % else % }
<div>
    <div class="d-flex flex-column align-items-center justify-content-center " style="height:
400px;">

```

Please, Sign in to start donating!

[Sign in](/sign_in)

</div>

</div>

{% endif % }

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-

MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

</body>

</html>

## **request.html :**

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<link href="https://fonts.googleapis.com/css?family=Roboto:400,700,900&display=swap" rel="stylesheet">

<!-- Vendor CSS Files -->

<link href="../static/vendor/aos/aos.css" rel="stylesheet">

<link href="../static/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">

<link href="../static/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">

<link href="../static/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">

<link href="../static/css/style.css" rel="stylesheet">

```

<!-- Bootstrap 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">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>

```

```

<!-- Style -->
<link rel="stylesheet" href="../static/css/request.css">
<script>
function validateForm() {

    let name = document.forms["form"]["name"].value;
    let email = document.forms["form"]["email"].value;
    let blood_group = document.forms["form"]["blood_group"].value;
    let contact_no = document.forms["form"]["contact_no"].value;
    let city = document.forms["form"]["city"].value;

    if(validname(name) && validemail(email) && validblood_group(blood_group) &&
validcontact_no(contact_no) && validcity(city)){
        return true;
    }
    else{
        return false;
    }

}

function validname(name){
    document.getElementById("name_err").innerHTML=""
    if (name != "") {

```

```

var letters = /^[a-zA-Z ]*$/;
    if(name.match(letters))
    {
        return true;
    }
    else
    {
        document.getElementById("name_err").innerHTML="Name should contain only
Alphabets"
        return false;
    }
}
else{
    document.getElementById("name_err").innerHTML="Name should not be empty"
    return false;
}
}
function validemail(email){
    console.log(email)
    document.getElementById("email_err").innerHTML="";
    if (email != "") {
        var emailfor = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3})+$/;
        if(email.match(emailfor))
        {
            return true;
        }
        else
        {
            document.getElementById("email_err").innerHTML="Invalid Email"
            return false;
        }
    }
}
else{

```

```

        document.getElementById("email_err").innerHTML="Email Should not be empty"
        return false;
    }
}

function validblood_group(blood_group){
    document.getElementById("blood_err").innerHTML="";
    if (blood_group == "") {
        document.getElementById("blood_err").innerHTML="Blood group Should not be empty"
        return false;
    }
    else{
        return true;
    }
}

}

function validcontact_no(contact_no){
document.getElementById("contact_err").innerHTML="";
if (contact_no == "") {
    document.getElementById("contact_err").innerHTML="Contact number Should not be
empty"
    return false;
}
else{
    return true;
}
}

}

function validcity(city){
document.getElementById("city_err").innerHTML="";
if (city == "") {
    document.getElementById("city_err").innerHTML="City should not be empty"

```

```

        return false;
    }
    else{
        return true;
    }
}

</script>
<title>Planor</title>
</head>
<body>
<!-- ===== Header ===== -->
<header id="header" class="fixed-top d-flex align-items-center shadow">
    <div class="container d-flex align-items-center justify-content-between">

        <div class="logo">
            <h1 class="text-light"><a href="/"><span>Planor</span></a></h1>
        </div>

        <nav id="navbar" class="navbar">
            <ul>
                <li><a class="nav-link scrollto" href="/">Home</a></li>
                <li><a class="nav-link scrollto" href="#services">Services</a></li>
                <li><a class="nav-link scrollto" href="/request">Request</a></li>
                <li><a class="nav-link scrollto" href="/donor_registration">Donate</a></li>
                <li><a class="nav-link scrollto" href="/sign_in">SIgn In</a></li>
                <li><a class="nav-link scrollto" href="/sign_up">Sign Up</a></li>
                <li><a class="nav-link scrollto" href="/profile">My Profile</a></li>
                <li><a class="nav-link scrollto" href="/logout">Logout</a></li>
            </ul>
            <i class="bi bi-list mobile-nav-toggle"></i>
        </nav><!-- .navbar -->

```

```

</div>
</header><!-- End Header -->
{% if logged_in == True %}
<div class="content">

<div class="container">
<div class="row justify-content-center">
<div class="col-md-10">

<div class="row justify-content-center">
<div class="col-md-6">

<h3 class="heading mb-4">Let's make a request!</h3>
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Voluptas debitis, fugit
natus?</p>

<p></p>

</div>
<div class="col-md-6">

<form action="{ { url_for('create_request') } }" onsubmit="return validateForm()"
method="post" id="contactForm" name="form">
<div class="row">
<div class="col-md-12 form-group">
<input onkeyup="validname(this.value)" type="text" class="form-control mb-3"
name="name" id="name" placeholder="Your name" value={ { name } }>
</div>
<div style="margin-top: -15px;">

```

```

        <label style="position: static; color:red" id="name_err" class="text-danger"
></label>

    </div>
</div>

<div class="row">
    <div class="col-md-12 form-group">
        <input onkeyup="validemail(this.value)" type="text" class="form-control mb-3"
name="email" id="email" placeholder="Email" value={{ email }}>
    </div>
    <div style="margin-top: -15px;">
        <label style="position: static; color:red" id="email_err" class="text-danger"
></label>
    </div>
</div>

<div class="row">
    <div class="col-md-12 form-group">
        <input onkeyup="validblood_group(this.value)" type="text" class="form-control
mb-3" name="blood_group" id="subject" placeholder="Blood group">
    </div>
    <div style="margin-top: -15px;">
        <label style="position: static; color:red" id="blood_err" class="text-danger"
></label>
    </div>
</div>

<div class="row">
    <div class="col-md-12 form-group">
        <input onkeyup="validcontact_no(this.value)" type="text" class="form-control mb-
3" name="contact_no" id="subject" placeholder="Contact number">
    </div>
    <div style="margin-top: -15px;">

```



```

        <label style="position: static; color:red" id="contact_err" class="text-danger"
></label>
    </div>
</div>

<div class="row">
    <div class="col-md-12 form-group">
        <input onkeyup="validcity(this.value)" type="text" class="form-control mb-3"
name="city" id="subject" placeholder="City">
    </div>
    <div style="margin-top: -15px;">
        <label style="position: static; color:red" id="city_err" class="text-danger" ></label>
    </div>
</div>

<div class="row">
    <div class="col-12">
        <input type="submit" value="Register as Donor" class="btn btn-primary rounded-0
py-2 px-4">
        <span class="submitting"></span>
    </div>
</div>
</form>

<div id="form-message-warning mt-4"></div>
<div id="form-message-success">
    Your message was sent, thank you!
</div>
</div>
</div>
</div>

```

```

    </div>
  </div>

</div>
{% else %}
<div>
  <div class="d-flex flex-column align-items-center justify-content-center " style="height:
400px;">
    Please, Sign in to make a request
    <a class="btn btn-dark mt-2" style="height: initial;" href="/sign_in">Sign in</a>

  </div>
</div>
{% endif %}
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>

</body>
</html>

```

### **profile.html:**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Bootstrap 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">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-
  MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
  crossorigin="anonymous"></script>
  <!-- Vendor CSS Files -->
  <link href="../static/vendor/aos/aos.css" rel="stylesheet">

  <link href="../static/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
  <link href="../static/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
  <link href="../static/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">

  <link href="../static/css/style.css" rel="stylesheet">
  <title>Profile</title>
</head>
<body>

  <!-- ===== Header ===== -->
  <header id="header" class="fixed-top d-flex align-items-center shadow">
    <div class="container d-flex align-items-center justify-content-between">

      <div class="logo">
        <h1 class="text-light"><a href="/"><span>Planor</span></a></h1>
      </div>

      <nav id="navbar" class="navbar">
        <ul>
          <li><a class="nav-link scrollto" href="/">Home</a></li>
          <li><a class="nav-link scrollto" href="#services">Services</a></li>
          <li><a class="nav-link scrollto" href="/request">Request</a></li>
          <li><a class="nav-link scrollto" href="/donor_registration">Donate</a></li>
          <li><a class="nav-link scrollto" href="/sign_in">Sign In</a></li>

```

```

        <li><a class="nav-link scrollto" href="/sign_up">Sign Up</a></li>
        <li><a class="nav-link scrollto" href="/profile">My Profile</a></li>
        <li><a class="nav-link scrollto" href="/logout">Logout</a></li>
    </ul>

    <i class="bi bi-list mobile-nav-toggle"></i>
</nav><!-- .navbar -->

</div>

</header><!-- End Header -->

{% if logged_in == True %}
<div class="container" style="margin-top: 100px;">
    <div class="row">
        <div class="col-sm-4 text-center">
            <h3><b>Hello {{ name }}!</b></h3>
            <p>{{ email }}</p>

            <!-- <div class="text-center" style="background-color: RGB(224, 224, 224); height: 350px;
width:90%;display: flex;flex-direction: column; align-items: center;">
                <div style="margin: auto;">
                    <p>Hari Prasath P</p>
                    <p>Chennai</p>
                    <p>Email:hariprasath.hp5@gmail.com</p>
                    <p>Blood Group:O+</p>
                </div>

            </div> -->
        </div>
    </div>

    <div class="col-sm-8">
        <div>
            <div>
                {% if pending_requestslen!=0 %}
                <p>Pending Requests</p>

```

```

    {%for i in range(0, pending_requestslen)%}
    <form name="form" method="post" action="{{ url_for('accept_request') }}">
    <div class=" w-100 d-flex justify-content-around align-items-center p-1 mb-3"
style="background-color: RGB(224, 224, 224)" >
        <input type="hidden" value={{ pending_requests[i]["ID"]}} name="id">
        <label >Request ID :{{ pending_requests[i]["ID"]}}</label>
        <label >Name :{{ pending_requests[i]["NAME"]}}</label>
        <label >Email :{{ accepted_requests[i]["EMAIL"]}}</label>
        <label>Location:{{ pending_requests[i]["LOCATION"]}}</label>
        <label>Blood Group:{{ pending_requests[i]["BLOOD_GROUP"]}}</label>
        <input type="submit" class="btn btn-primary" value="Accept">
    </div>
</form>
    {%endfor%}
    {% endif %}
</div>

<div>
    {% if accepted_requestslen!=0 %}
    <p>Accepted Requests</p>
    {%for i in range(0, accepted_requestslen)%}
    <div class=" w-100 d-flex justify-content-around align-items-center p-1 mb-3"
style="background-color: RGB(224, 224, 224)" >
        <label >Request ID :{{ accepted_requests[i]["ID"]}}</label>
        <label >Name :{{ accepted_requests[i]["NAME"]}}</label>
        <label >Email :{{ accepted_requests[i]["EMAIL"]}}</label>
        <label>Location:{{ accepted_requests[i]["LOCATION"]}}</label>
        <label>Blood Group:{{ accepted_requests[i]["BLOOD_GROUP"]}}</label>
    </div>
    {%endfor%}
    {% endif %}
</div>

```

```

<div>
  {% if requests_len!=0 %}
  <p>Requests made</p>
  {%for i in range(0, requests_len)%}
    <div class=" w-75 d-flex justify-content-around align-items-center p-2 mb-3"
style="background-color: RGB(224, 224, 224)" >
      <label >Request ID :{ {requests[i]["ID"]} }</label>
      <label >Name :{ {requests[i]["NAME"]} }</label>
      <label>Location:{ {requests[i]["LOCATION"]} }</label>
      <label>Blood Group:{ {requests[i]["BLOOD_GROUP"]} }</label>
      <label>Status: { {requests[i]["REQUEST_STATUS"]} }</label>
    </div>
  {%endfor%}
  {% endif %}
</div>

</div>
</div>

</div>

</div>
{% else %}
<div>
  <div class="d-flex flex-column align-items-center justify-content-center " style="height:
400px;">
    Please, Sign in to view your profile
    <a class="btn btn-dark mt-2" style="height: initial;" href="/sign_in">Sign in</a>

  </div>
</div>
{% endif %}

```

```
</body>
```

```
</html>
```

## **Script.py :**

```
import ibm_db
```

```
from flask import *
```

```
app = Flask(__name__)
```

```
@app.route('/')
def home():
```

```
    return render_template('index.html')
```

```
@app.route('/sign_up')
def signUp():
```

```
    return render_template('sign_up.html')
```

```
@app.route('/sign_in')
def signIn():
```

```
    return render_template('sign_in.html')
```

```
@app.route('/request')
def requests():
```

```
    email = request.cookies.get('email')
```

```
    name = request.cookies.get('name')
```

```
    if email != None:
```

```

        resp = make_response(render_template('request.html',email = email, name = name,
logged_in = True))
    else:
        resp = make_response(render_template('request.html',email = email, name = name,
logged_in = False))
    return resp

```

```

@app.route('/donor_registration')
def donor_registration():
    email = request.cookies.get('email')
    name = request.cookies.get('name')
    if email != None:
        resp = make_response(render_template('donor_registration.html',email = email, name =
name, logged_in = True))
    else:
        resp = make_response(render_template('donor_registration.html',email = email, name =
name, logged_in = False))
    return resp

```

```

@app.route('/profile')
def profile():
    email = request.cookies.get('email')
    name = request.cookies.get('name')
    if email != None:
        conn = ibm_db.connect(
            'DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-
10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;SECURITY=S
SL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=gfn00031;PWD=LITZUQj2tpFc3t0
i', "", "")
        sql = 'select * from requests where email='+email+'\"
        stmt = ibm_db.exec_immediate(conn, sql)
        requests = []

```



```

dictionary = ibm_db.fetch_assoc(stmt)

while dictionary != False:
    print(dictionary["ID"])
    requests.append(dictionary)
    dictionary = ibm_db.fetch_assoc(stmt)
print(requests)

sql = 'select * from donors where email='++"\'+email+"
stmt = ibm_db.exec_immediate(conn, sql)
dictionary = ibm_db.fetch_assoc(stmt)
isDonor = False
pending_requests = []
if dictionary != False:
    isDonor = True
    donor_location = dictionary["LOCATION"]
    donor_bloodgroup = dictionary["BLOOD_GROUP"]
    sql = "select * from requests where blood_group="+"""+donor_bloodgroup+"""+ "and
location= "+"""+donor_location+"""+ "and request_status= "+ "Pending"
    stmt = ibm_db.exec_immediate(conn, sql)

    dictionary = ibm_db.fetch_assoc(stmt)
    while dictionary != False:
        pending_requests.append(dictionary)
        dictionary = ibm_db.fetch_assoc(stmt)
    print(pending_requests)

accepted_requests= []
if isDonor:
    sql = 'select * from requests where accepted_by='++"\'+email+"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_assoc(stmt)

```

```

        while dictionary != False:
            accepted_requests.append(dictionary)
            dictionary = ibm_db.fetch_assoc(stmt)
        print(accepted_requests)

        return render_template('profile.html', name =name, email = email,requests_len =
len(requests) ,requests = requests, pending_requestslen = len(pending_requests),
pending_requests = pending_requests, accepted_requestslen = len(accepted_requests),
accepted_requests = accepted_requests, logged_in=True)

    else:
        return render_template('profile.html', logged_in= False)
@app.route('/add_user', methods=['POST', 'GET'])
def add_user():
    if request.method == 'POST':
        try:

            name = request.form['name']
            email = request.form['email']
            password = request.form['pass']
            conn = ibm_db.connect(
                'DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-
10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;SECURITY=S
SL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=gfn00031;PWD=LITZUQj2tpFc3t0
i', "", "")

            sql = "insert into users values(?,?,?)"
            param = name, email, password,
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.execute(stmt, param)

            msg = "You're successfully signed up!"
        except Exception as e:
            print("exception occured!",e)
            msg = e

```

```

finally:
    return render_template('post_signup.html', msg = msg)

@app.route('/validate_user', methods = ['POST', 'GET'])
def validate_user():
    if request.method == 'GET':
        try:
            args = request.args
            email = args.get('email')
            password = args.get('password')

            conn = ibm_db.connect(
                'DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-
10cf081900bf.c1ogj3sd0tgu0lqde00.databases.appdomain.cloud;PORT=32304;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=gfn00031;PWD=LITZUQj2tpFc3t0i', "", "")

            sql = 'select * from users where email='+\"'+email+\"'
            stmt = ibm_db.exec_immediate(conn, sql)
            dictionary = ibm_db.fetch_assoc(stmt)
            print("executed")
            print(dictionary)
            if dictionary != False:
                if(dictionary["PASSWORD"]== password):
                    print("success")
                    resp = make_response(render_template("post_signin.html"))
                    resp.set_cookie('email', dictionary["EMAIL"])
                    resp.set_cookie('name', dictionary["NAME"])
                    print("success")
                    return resp
                else:
                    return "Incorrect Password"
            else:
                return "User does not exists"

```

```

except Exception as e :
    print("error",e)
    return repr(e)

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

            name = request.form['name']
            email = request.form['email']
            blood_group = request.form['blood_group']
            contact_no = request.form['contact_no']
            location = request.form['city']

            conn = ibm_db.connect(
                'DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-
10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;SECURITY=S
SL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=gfn00031;PWD=LITZUQj2tpFc3t0
i', "", "")

            sql = "insert into donors values(?,?,?,?)"
            param = name, email, blood_group, contact_no, location,
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.execute(stmt, param)
            msg = "You're successfully registered as donor"

        except Exception as e:
            print("exception occured!",e)
            msg = e

    finally:
        return render_template('donor_registration_status.html', msg = msg)

```

```

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

            name = request.form['name']
            email = request.form['email']
            blood_group = request.form['blood_group']
            contact_no = request.form['contact_no']
            location = request.form['city']

            conn = ibm_db.connect(
                'DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-
10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;SECURITY=S
SL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=gfn00031;PWD=LITZUQj2tpFc3t0
i', ", ")

            sql = "insert into requests (name, email, blood_group, contact_no, location)
values(?,?,?,?,?)"

            param = name, email, blood_group, contact_no, location,
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.execute(stmt, param)
            msg = "You're successfully made a request!"

        except Exception as e:
            print("exception occured!", e)
            msg = e

        finally:
            return render_template('donor_registration_status.html', msg = msg)

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

```

```

try:

    id = request.form['id']
    email = request.cookies.get('email')
    conn = ibm_db.connect(
        'DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-
10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=gfn00031;PWD=LITZUQj2tpFc3t0
i', "", "")

    sql = "update requests set request_status = 'Accepted' , accepted_by
='"+email+"' where id ='"+id+"'"
    stmt = ibm_db.exec_immediate(conn, sql)

except Exception as e:
    print("exception occured!",e)

finally:
    return redirect(url_for('profile'))

@app.route('/logout')
def logout():

    email = request.cookies.get('email')
    if email != None:
        resp = make_response(render_template('logout.html',loggedin = True))
        resp.set_cookie('name', "", expires=0)
        resp.set_cookie('email', "", expires=0)
        resp.set_cookie('dob', "", expires=0)

    else:
        resp = make_response(render_template('logout.html',loggedin = False))

    return resp

```

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

## **GITHUB & PROJECT DEMO LINK**

**Github Link :**

**<https://github.com/IBM-EPBL/IBM-Project-14273-1659547967>**