

PLASMA DONOR APPLICATION

DOMAIN

TEAM ID - PNT2022TMID38560

A PROJECT REPORT

Submitted By

ANUSHIYA S	(420419104001)
BHAVANI P	(420419104004)
EZHILAMMAL M	(420419104012)
JAYASHREE J	(420419104020)

COMPUTER SCIENCE AND ENGINEERING

ADHIPARASAKTHI ENGINEERING COLLEGE

TABLE OF CONTENTS

CHAPTER	TITLE
1	INTRODUCTION 1.1 Project Overview 1.2 Purpose
2	LITERATURE SURVEY 2.1 Existing Problem 2.2 Reference 2.3 Problem Statement Definition
3	IDEATION & PROPOSED SOLUTION 3.1 Empathy Map Canvas 3.2 Ideation & Brainstorming 3.3 Proposed Solution 3.4 Problem Solution fit
4	REQUIREMENT ANALYSIS 4.1 Functional requirement 4.2 Non-Functional requirements
5	PROJECT DESIGN 5.1 Data Flow Diagrams 5.2 Solution & Technical Architecture 5.3 User Stories

6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

7

CODING & SOLUTIONING

**(Explain the features added in
the project along with code)**

7.1 Feature 1

7.2 Feature 2

8

TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9

RESULTS

9.1 Performance Metrics

9.2 Output

10

ADVANTAGES

&

DISADVANTAGES

10.1 Advantages

10.2 Disadvantages

11

CONCLUSION

12

FUTURE SCOPE

13

APPENDIX

13.1 Source Code

13.2 GitHub

13.3 Project Demo Link

CHAPTER 1

INTRODUCTION

The world is suffering from the COVID 19 crisis and no vaccine has been found yet.. But there is another scientific way in which we can help reduce mortality or help people affected by COVID19 by donating plasma from recovered patients. In the absence of an approved antiviral treatment plan for a fatal COVID19 infection, plasma therapy is an experimental approach to treat COVID19-positive patients and help them faster recovery. Therapy is considered competent. In the recommendation system, the donor who wants to donate plasma can donate by uploading their COVID19 certificate and the blood bank can see the donors who have uploaded the certificate and they can make a request to the donor and the hospital can register/login and search for the necessar things. plasma from a blood bank and they can request a blood bank and obtain plasma from the blood bank.

1.1 Project Overview

The main goal of our project is to make it easier for the COVID-19 patients to get a plasma donor easily as well as donate plasma if they have recovered. The system targets two types of users: the people who want to donate plasma and the people who need plasma. The main objective of developing the application is to make it easier for the COVID-19 patients to get a plasma donor easily and as soon as possible.

1.2 PURPOSE:

Plasma Donor Application deals with notifying concerned donor upon request by therecipient in need of plasma . This project provides quick access to donors for an immediate requirement of blood. In case of emergency/surgery. Blood procurement is always a major problem which consumes lot of time .

CHAPTER 2

LITERATURE SURVEY

2.1 Existing problem:

There are many people who are willing to donate plasma to those who need plasma. But there is not any accessible way to help them to find plasma donation centers in real-time. So, the problem is not the lack of donors, but finding the right sponsor at the right time. If someone needs plasma, they seek plasma first from family members, then from hospitals and the nearest plasma bank. If they can't process plasma in these ways, it's very difficult for them to contact another for a short-term plasma draw. This is a problem that I want to solve through this application. Instead of just providing plasma to people in need with an outdated list of regular plasma donors who may or may not be available to help, This application reaches the right people the moment users find Out.

2.2 References:

1. PLASMA DONOR APPLICATION USING FUNCTION-AS-A-SERVICE IN AWS

Author : George Amvrosiadis 2020

2. ANDROID BASED HEALTH APPLICATION IN CLOUD COMPUTING FOR PLASMA

Author: Patil Sayali Dhond 2019

3. MOBILE APPLICATION DEVELOPMENT FOR PLASMA AND BLOOD DONOR

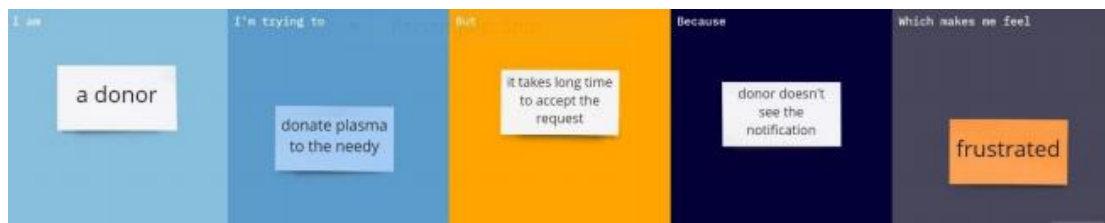
Author: Michael J Moss 2021

4. INSTANT PLASMA DONOR RECIPIENT CONNECTOR WEBAPPLICATION

Author: Sanjay Malliseti 2022

2.3 PROBLEM STATEMENT DEFINITION :

During COVID 19 crisis the requirement for plasma increased drastically as there were no vaccinations found in order to treat the infected patients. In such situation it was very difficult to find the plasma donor, check whether the donor was infected previously and was recovered, and which donor is eligible to donate plasma was a challenging task. As the plasma therapy was one of the ways to treat the infected patients getting the donor details played a major role.

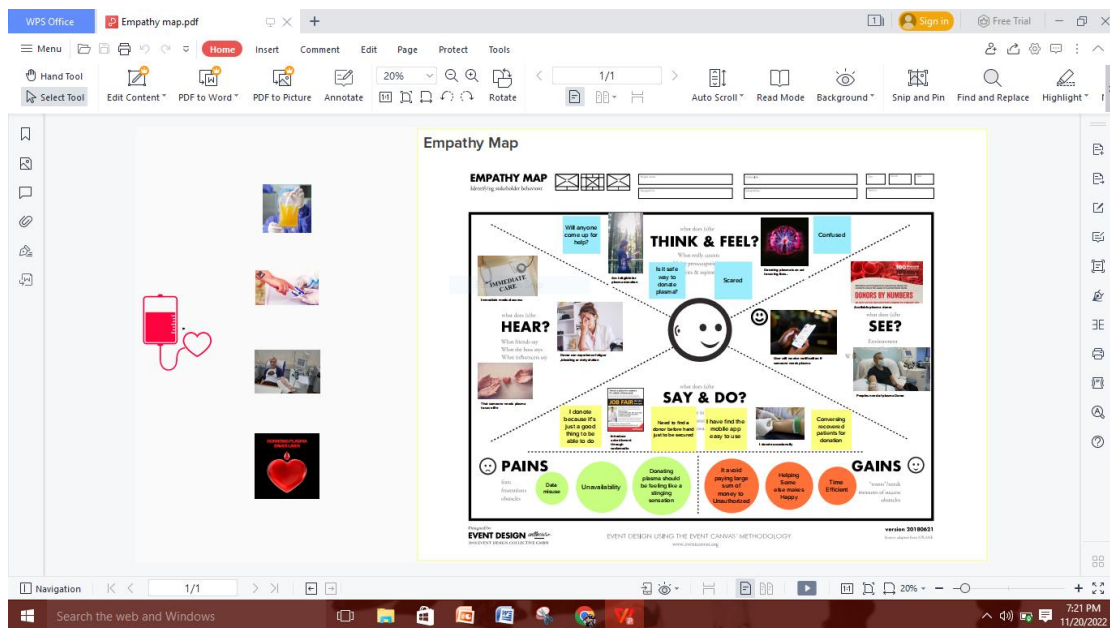


CHAPTER 3

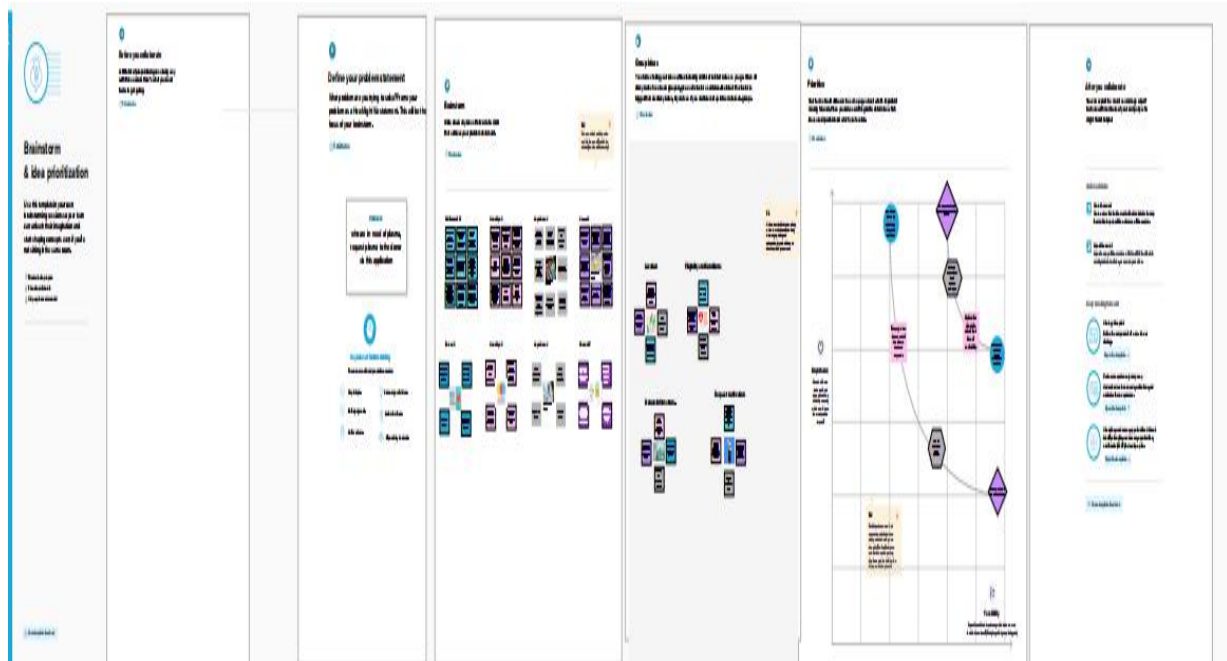
IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behavior's and attitudes. It is a useful tool to helps 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.



3.2 Ideation & Brainstorming



3.3 Proposed Solution

Problem Statement (Problem to be solved):

During COVID 19 crisis the requirement for plasma increased drastically as there were no vaccinations found in order to treat the infected patients. In such situation it was very difficult to find the plasma donor, check whether the donor was infected previously and was recovered, and which donor is eligible to donate plasma was a challenging task. As the plasma therapy was one of the ways to treat the infected patients getting the donor details played a major role

Idea / Solution description:

This proposed system aims at connecting the donors & the patients by an online application. By this creating application with UI to interact with the user for getting the donor details ,who need it can see their details providing them upon the recipient's request so that they can get the plasma.

Novelty / Uniqueness:

Our application allow the user to request and donate the plasma. The person need the plasma immediately or pre request. You have plasma immediately then give emergency request, then all registered member on the application to get voice alert.

Social Impact / Customer Satisfaction

In this covid19 period the requirement for plasma need high and the donor count has low, so using this application provides opportunity come forward to donate plasma. we have predicted that effect of donor motivation on donor relationship satisfaction and loyalty change.

Business Model (Revenue Model)

The application is user friendly and can be easily used. User Data can be stored in IBM DB2 in cloud which reduces the overall cost incurred for developing the application. This application is accessible by everyone . This can be used anywhere anytime.

Scalability of the Solution

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

3.4 Problem Solution fit

<p>3. TRIGGERS Earn rewards for donation. In emergency period is used for plasma. Plasma donor application will used to triggers the peoples to donate the plasma .</p>	<p>10. YOUR SOLUTION Our app allow the user to request and donate plasma .The person need the plasma immediately or pre request. You have plasma immediately then give emergency request , then all register members on the application</p>	<p>8. CHANNELS of BEHAVIOUR 8.1 ONLINE The plasma donor app allow user to make donor and receiver process to sending mail The user send the request any where any time Through social media.</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>4. EMOTIONS: BEFORE / AFTER Before : confused ,Anxious ,Exhausted ,Scared After :Relaxed, Motivated</p>	<p>to get voice alert.</p>	<p>8.2 OFFLINE Ask friends or other previous user recommendation. User visit near by camp or hospital</p>
---------------------------------------------------------------------------------------------------------------------------------------------	----------------------------	--------------------------------------------------------------------------------------------------------------------------------------------

<p>3. TRIGGERS Earn rewards for donation. In emergency period is used for plasma. Plasma donor application will used to triggers the peoples to donate the plasma .</p>	<p>10. YOUR SOLUTION Our app allow the user to request and donate plasma .The person need the plasma immediately or pre request. You have plasma immediately then give emergency request , then all register members on the application</p>	<p>8. CHANNELS of BEHAVIOUR 8.1 ONLINE The plasma donor app allow user to make donor and receiver process to sending mail The user send the request any where any time Through social media.</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>4. EMOTIONS: BEFORE / AFTER Before : confused ,Anxious ,Exhausted ,Scared After :Relaxed, Motivated</p>	<p>to get voice alert.</p>	<p>8.2 OFFLINE Ask friends or other previous user recommendation. User visit near by camp or hospital</p>
---------------------------------------------------------------------------------------------------------------------------------------------	----------------------------	--------------------------------------------------------------------------------------------------------------------------------------------

CHAPTER 4

REQUIREMENT ANALYSIS

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Website.
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP.
FR-3	Certification	After the donor donates plasma, we will provide certificate the appreciation and authentication.
FR-4	Searching	Users can use the search bar to look up information about the plasma donor.
FR-5	User plasma request	Users can request to donate plasma by filling out there quest form .
FR-6	End result	User can donate and request plasma in this application.

Non-functional Requirements:

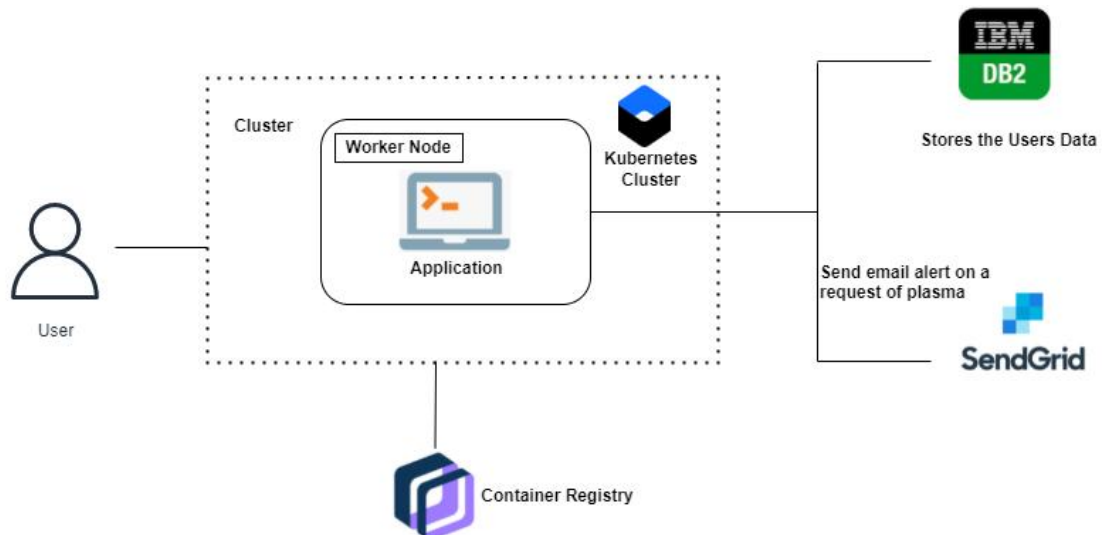
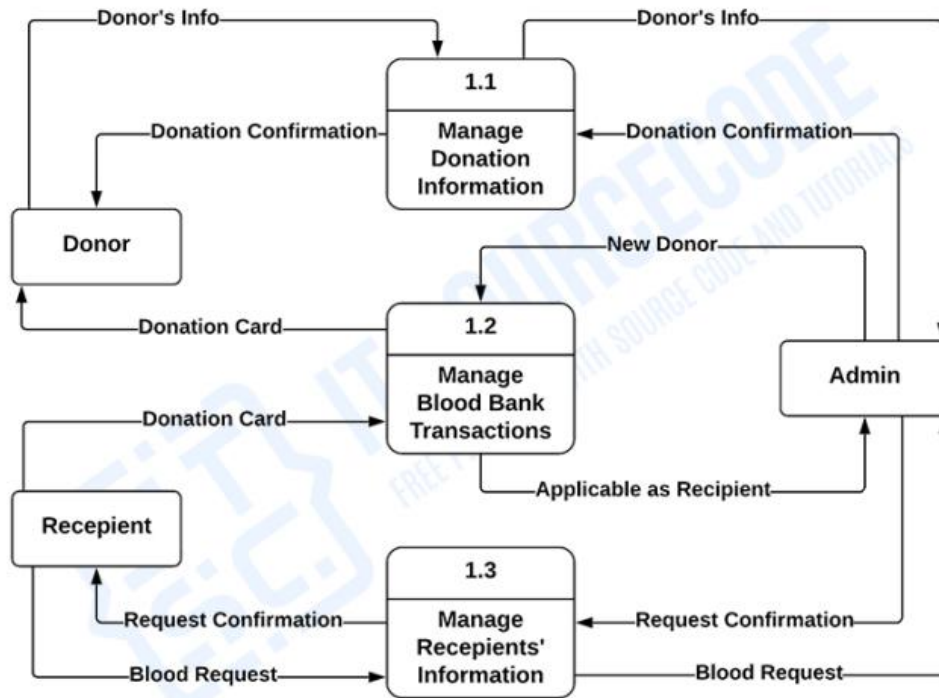
Following are the non-functional requirements of the proposed solution.

FR No	Non-Functional Requirement	Description
NFR-1	Security	The data storage is required by security systems just like its by many other application. The enter system needs to be safe and producter. It must be secured with email ID and password
NFR-2	Reliability	The system need to be reliable enough and needs to function without any network failures.The user easy to donate plasma and rescue the plasma.
NFR-3	Performance	The plasma donor system must perform well in different scenarios. the system is interactive and delays involved or less
NFR-4	Availability	The system needs to be accessible to a user at any given point of time.
NFR-5	Scalability	The system offers the proper resources for issue solutions and is design to product sensitive information during all phases of operation.

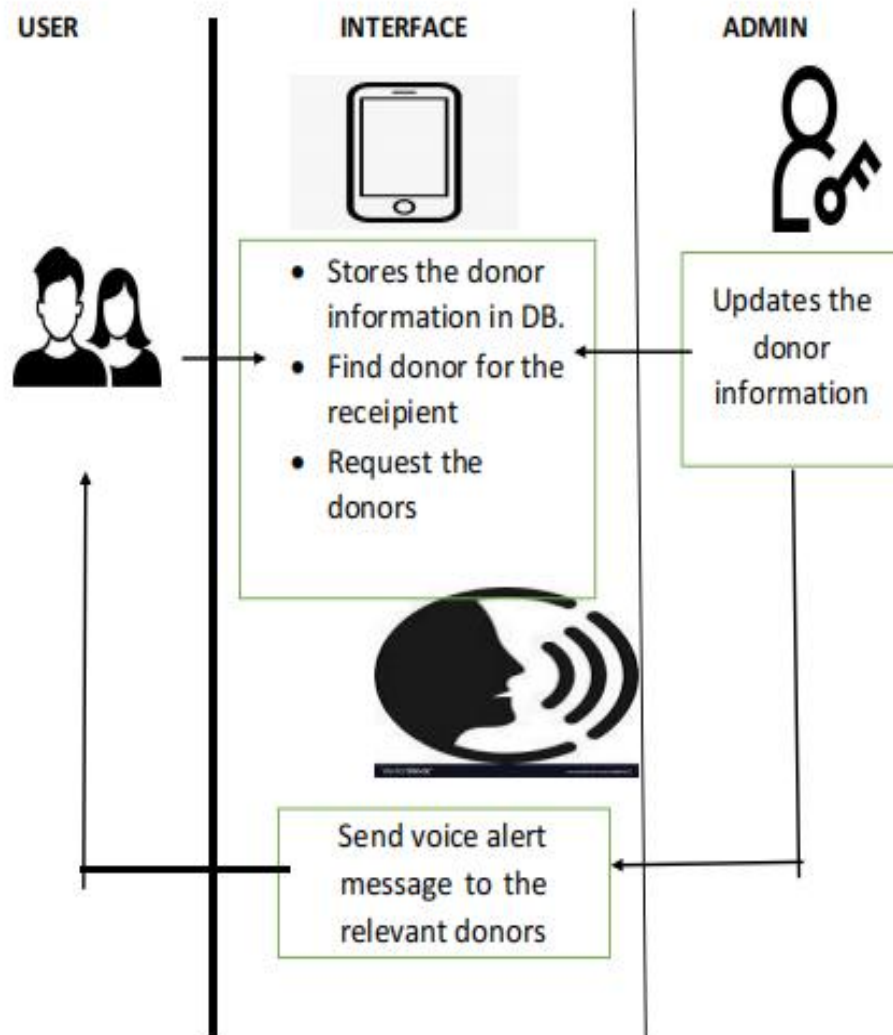
CHAPTER 5

PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



User Stories

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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
Users	Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application to Donate Plasma	I can receive confirmation email & click confirm	High	Sprint-1
Users	Registration	USN-3	As a user, I can register for the application through Website or Application .	I can register & access the dashboard with Website Login	Medium	Sprint-2
Users	Registration	USN-4	As a user, I can register for the application through plasma Donor App	I can confirm the registration then access conditions	Medium	Sprint-1
Recipient	Login	USN-5	As a recipient , I can log into the application by entering email & password	I can view and access what are the features are provided in dashboard	High	Sprint-1
Users	Dashboard	USN-6	As a user, I can view and manage my profile, donation history and download the receipts.	I can view and manage my data at each section of the dashboard.	High	Sprint -1
Customer (Web user)	Website	USN-7	As the user I can login using my credentials and it will direct it to my dashboard	I can view and modify the given data	Medium	Sprint-2
Customer Care Executive	Health Care	USN-8	As a customer care executive, I can solve the queries of the users.	I can reply to their queries and solve their related problems	High	Sprint-2

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering username, email id, password, blood group etc.	5	High	Bhavani Ezhilammal
Sprint-1	Email verification	USN-2	As a user, I will receive confirmation email once I have register for the application.	5	High	Anushiya Jayashree
Sprint-1	Login	USN-3	As a use, I can login into the application by entering user name, email id and password.	4	High	Bhavani Ezhilammal
Sprint-1	Home page	USN-4	As a user, I can view the home page of the application. And view some details about plasma donation	5	Medium	Jayashree

6.3 Reports from JIRA

The screenshot shows the JIRA Roadmap view for the 'PLASMA DONOR' project. The left sidebar contains navigation options: PLANNING (Roadmap, Backlog, Board, Reports, Issues), DEVELOPMENT (Code), and OPERATIONS (Deployments, On-call). The main area displays a timeline from October to December. Sprints are listed as PD-1 Registration, PD-2 Email Verification, PD-3 Login, PD-4 Homepage, PD-5 Virtual Certification, PD-6 Plasma request, PD-7 Accept the request, and PD-8 Alarm. A vertical line indicates the current date is 'Today'. A 'Quickstart' button is visible in the bottom right corner.

The screenshot shows the JIRA Backlog view for the 'Plasma donor app' project. The left sidebar contains navigation options: PLANNING (Roadmap, Backlog, Board, Reports, Issues), DEVELOPMENT (Code, Releases), and OPERATIONS (Deployments). The main area displays two sprints: 'PLAS Sprint 1 6 Nov - 7 Nov (4 issues)' and 'PLAS Sprint 2 Add dates (5 issues)'. Issues are listed with their status (e.g., IN PROGRESS) and priority. A 'Quickstart' button is visible in the bottom right corner.

CHAPTER 7

CODING & SOLUTIONING

(Explain the features added in the project along with code)

FEATURE CODE :

```
from distutils.log import debug
# from sendgridmail import sendmail
from flask import Flask, render_template, request, redirect, url_for,
session
from flask_mail import Mail, Message
import re
import os
import ibm_db
from dotenv import load_dotenv
load_dotenv()
app = Flask(__name__)
app.secret_key = 'a'
print("Try to connect to Db2")
conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-
73d1-4859-
88f0-
a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;POR
T=
;UID=;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.
crt;PWD=
", ",")
print("Connected Successfully")
app.config['MAIL_SERVER']='smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] = 'example@gmail.com'
app.config['MAIL_PASSWORD'] = '*****'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True
mail = Mail(app)
@app.route('/')
@app.route('/login')
def login():
return render_template('login.html')
@app.route('/loginpage',methods=['GET', 'POST'])
def loginpage():
global userid
```

```

msg = "
if request.method == 'POST' :
    username = request.form['username']
    password = request.form['password']
    sql = "SELECT * FROM donors WHERE username =? AND
password=?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,username)
    ibm_db.bind_param(stmt,2,password)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print (account)
    if account:
        session['loggedin'] = True
        session['id'] = account['USERNAME']
        userid= account['USERNAME']
        session['username'] = account['USERNAME']
        msg = 'Logged in successfully !'
        index(account['EMAIL'],'Plasma donor App login','You are successfully
logged in!')
        return redirect(url_for('dash'))
    else:
        msg = 'Incorrect username / password !'
        return render_template('login.html', msg = msg)
@app.route('/registration')
def home():
    return render_template('register.html')
@app.route('/register',methods=['GET', 'POST'])
def register():
    msg = "
    if request.method == 'POST' :
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        phone = request.form['phone']
        city = request.form['city']
        infect = request.form['infect']
        blood = request.form['blood']
        sql = "SELECT * FROM donors WHERE username =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

```

```

print("ac",account)
if account:
    msg = 'Account already exists !'
elif not re.match(r'^[@]+\.[^@]+', email):
    msg = 'Invalid email address !'
elif not re.match(r'[A-Za-z0-9]+', username):
    msg = 'name must contain only characters and numbers !'
else:
    insert_sql = "INSERT INTO donors VALUES (?, ?, ?, ?, ?, ?, ?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prepare_stmt, 1, username)
    ibm_db.bind_param(prepare_stmt, 2, password)
    ibm_db.bind_param(prepare_stmt, 3, email)
    ibm_db.bind_param(prepare_stmt, 4, phone)
    ibm_db.bind_param(prepare_stmt, 5, city)
    ibm_db.bind_param(prepare_stmt, 6, infect)
    ibm_db.bind_param(prepare_stmt, 7, blood)
    ibm_db.execute(prepare_stmt)
    msg = 'You have successfully registered, !'
    index(email,'Plasma donor App Registration','You are successfully
    Registered {}'.format(username))
    elif request.method == 'POST':
        msg = 'Please fill out the form !'
        return render_template('register.html', msg = msg)
@app.route('/dashboard')
def dash():
    if session['loggedin'] == True:
        sql = "SELECT COUNT(*), (SELECT COUNT(*) FROM DONORS
        WHERE blood= 'O Positive'), (SELECT COUNT(*) FROM DONORS
        WHERE
        blood='A Positive'), (SELECT COUNT(*) FROM DONORS WHERE
        blood='B
        Positive'), (SELECT COUNT(*) FROM DONORS WHERE blood='AB
        Positive'),
        (SELECT COUNT(*) FROM DONORS WHERE blood='O Negative'),
        (SELECT
        COUNT(*) FROM DONORS WHERE blood='A Negative'), (SELECT
        COUNT(*) FROM DONORS WHERE blood='B Negative'), (SELECT
        COUNT(*) FROM DONORS WHERE blood='AB Negative') FROM
        donors"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

```

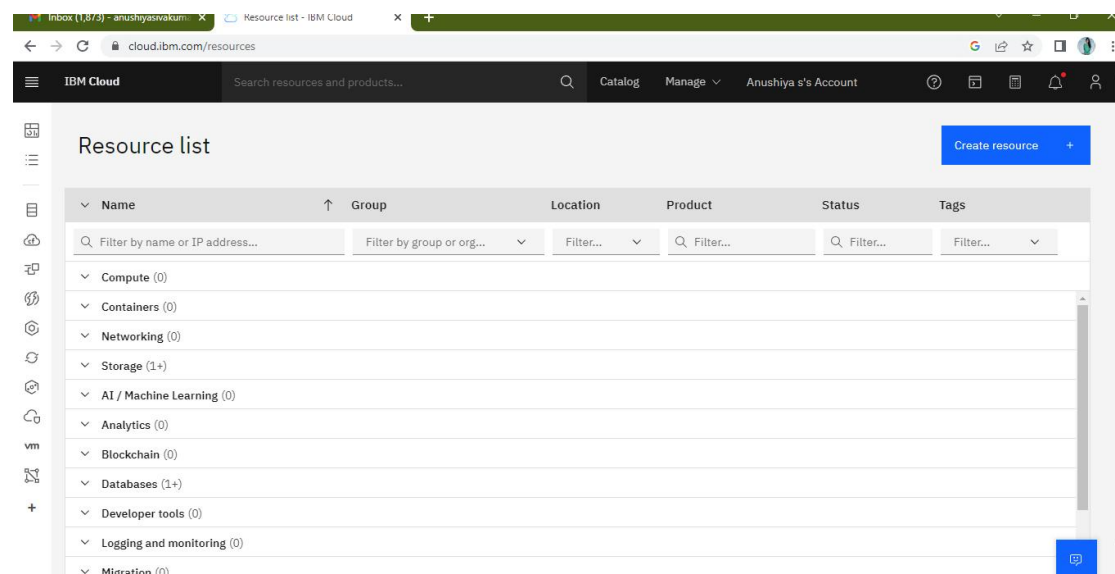
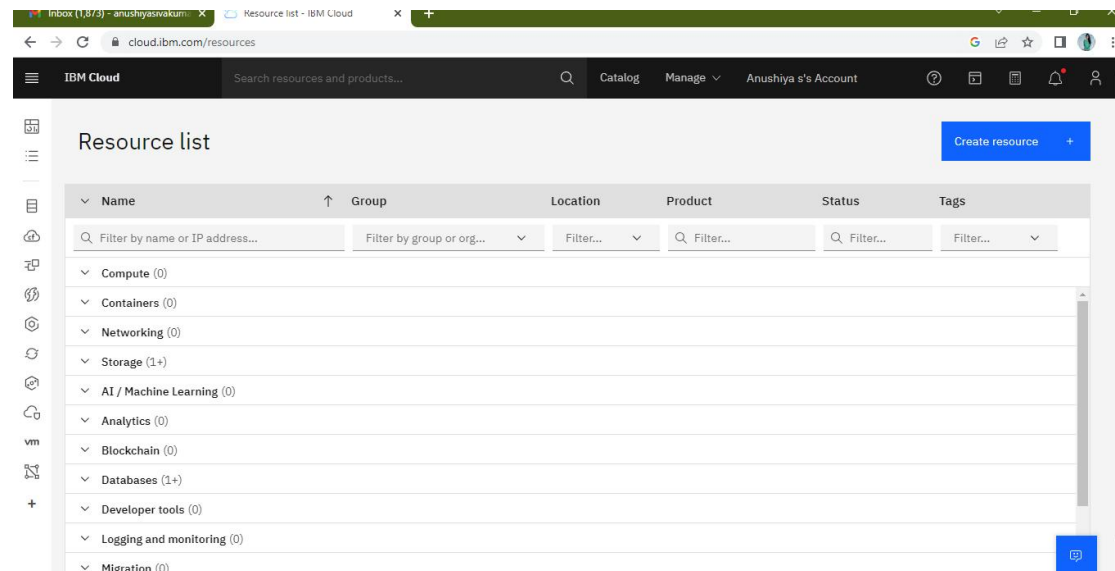
```

print(account)
return render_template('dashboard.html',b=account)
else:
msg = 'Please login!'
return render_template('login.html', msg = msg)
@app.route('/requester')
def requester():
if session['loggedin'] == True:
return render_template('request.html')
else:
msg = 'Please login!'
return render_template('login.html', msg = msg)
@app.route('/requested',methods=['POST'])
def requested():
bloodgrp = request.form['bloodgrp']
address = request.form['address']
name= request.form['name']
email= request.form['email']
phone= request.form['phone']
insert_sql = "INSERT INTO requested VALUES (?, ?, ?, ?, ?)"
prep_stmt = ibm_db.prepare(conn, insert_sql)
ibm_db.bind_param(prepare_stmt, 1, bloodgrp)
ibm_db.bind_param(prepare_stmt, 2, address)
ibm_db.bind_param(prepare_stmt, 3, name)
ibm_db.bind_param(prepare_stmt, 4, email)
30 ibm_db.bind_param(prepare_stmt, 5, phone)
ibm_db.execute(prepare_stmt)
index(email,'Plasma donor App plasma request','Your request for plasma
is
recieved.')
return render_template('request.html', pred="Your request is sent to the
concerned people.")
def index(usermail,subject,content):
msg = Message(subject, sender = 'example@gmail.com', recipients =
[usermail])
msg.body = format(content)
mail.send(msg)
return "Sent"
@app.route('/logout')
def logout():
session.pop('loggedin', None)
session.pop('id', None)
session.pop('username', None)

```

```
return render_template('login.html')
if __name__ == '__main__':
app.run(host='0.0.0.0',debug='TRUE')
```

DATABASE SCHEMA :



7.2 Feature

Home.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 rel="stylesheet" href="page.css">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href
"https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.c
ss"
    integrity
"sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
    crossorigin="anonymous">

  <title>                                </title>
<style>
  @import
url 'https://fonts.googleapis.com/css?family=Big+Shoulders+Text:100,
300, 400, 500, 600, 700, 800, 900&display=swap'

  /* font-family: 'Big Shoulders Text', cursive; */

.nav-flex-row
  display flex
  flex-direction row
  justify-content center
  position absolute
  z-index 100
  left 0
  width 100%
  padding 0
  background-clip #f37b18d2

.nav-flex-row li
  text-decoration none
  list-style-type none
  padding 20px 15px

.nav-flex-row li a
  font-family 'Times New Roman' serif
  color rgb 216 19 19
  font-size 1.5em
  text-transform uppercase
```



```
    font-weight 300

.nav-flex-row li a:hover
  background #E7E7E7

.section-intro
  height 820px
  background-image url img/foddiee.png
  background-size cover
  display flex
  flex-direction column
  justify-content center
  align-items center

.section-intro h1
  text-align center
  color #000
  font-size 4em
  font-weight 700

.section-intro header
  display flex
  flex 4
  flex-direction row
  justify-content center
  align-items center

.link-to-book-wrapper
  flex 1

.about-section
  display flex
  align-items center
  background-color #f3f3f3c0
  padding 50px 30px

.link-to-book
  color #ffffff
  display block
  border 2px solid #ffffff
  padding 5px 10px

a.link-to-book:hover
  background-color #ffffff
  color #95999e
  text-decoration none

.about-section p
```

```
.about-section h3
  text-align center
  width 60%
  margin auto
  font-family 'Times New Roman' serif
  font-size 1.8em
  text-transform uppercase

.carousel-inner
  height 700px

.row-flex
  display flex
  flex-direction row

.flex-column-form
  display flex
  flex-direction column
  flex 1
  margin 30px 20px

.btn.btn-primary
  font-family 'Big Shoulders Text'
  color #ffffff
  background-color #95999e
  text-transform uppercase
  font-size 16px
  padding 5px 10px
  letter-spacing 2px
  border 0

.btn.btn-primary:hover
  background-color #747474

.opening-time
.contact-address
  flex 1
  margin 30px 20px
  font-size 1.2em

.form-group p
  font-size 1.2em

.opening-time p span
.contact-address p span
  display block

@media min-width 577px and max-width 800px
```

```
.section-intro
  height 500px
```

```
.about-section p
.about-section h3
  font-size 20px
```

```
.carousel-inner
  height auto
```

```
.row-flex
  display flex
  flex-direction column
```

```
@media screen and max-width 576px
  .section-intro
    height 300px
```

```
.about-section
  padding 30px
```

```
.section-intro h1
  font-size 2em
```

```
.about-section p
.about-section h3
  font-size 15px
```

```
.carousel-inner
  height auto
```

```
.row-flex
  display flex
  flex-direction column
```

```
.row-flex h3
  font-size 25px
  text-align center
```

```
.form-group p
  font-size 15px
```

```

        .opening-time p span
        .contact-address p span
            font-size 15px
            text-align center

</style>

</head>

<body backgroud    "C:\Users\Sajin\Downloads\plasma.jpg">

    <section class "section-intro">
        <header>
            <h1><img src "C:\Users\Sajin\Downloads\app.jpeg"
                class "d-block w-1" alt "food"
style "width:1500px;height:600px;">
            <br>

                                </h1>

        </header>

    </section>
    <center>
        <img src "C:\Users\Sajin\Downloads\ball.jpg"
            class "d-block w-1" alt "food"
style "width:1300px;height:700px;">
    </center>

    <section class "about-section">
        <article>

<p>

        </p>

    </article>
</section>

<!-- carousel section -->
<div id "carouselExampleControls"
    class "carousel slide" data-ride "carousel">
    <div class "carousel-inner">

```

```

        <div class "carousel-item active">
            <img src "C:\Users\Sajin\Downloads\donation.jpg"
                class "d-block w-100">
        </div>
        <div class "carousel-item">
            <img src "C:\Users\Sajin\Downloads\donor.webp"
                class "d-block w-100" alt "plasma">
        </div>
        <div class "carousel-item">
            <img src "C:\Users\Sajin\Downloads\heart.jpg"
                class "d-block w-100" alt "food">

    </div>
    <a class "carousel-control-prev"
        href "#carouselExampleControls"
        role "button" data-slide "prev">
        <span class "carousel-control-prev-icon"
            aria-hidden "true">
        </span>
        <span class "sr-only">          </span>
    </a>
    <a class "carousel-control-next"
        href "#carouselExampleControls"
        role "button" data-slide "next">
        <span class "carousel-control-next-icon"
            aria-hidden "true">
        </span>
        <span class "sr-only">          </span>
    </a>
</div>

    <div class "opening-time">
        <h3>

    </h3>

<p><ul type "square">

        <span><li>

    </li></span>

        <span><li>
            </li> </span>
        <span><li>

    </li> </span>
        <span> <li>

    </li></span>

        </ul>
    </p>

</div>

</div>
</div>

```

```

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src "https://code.jquery.com/jquery-3.3.1.slim.min.js"
    integrity
"sha384-
q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
    crossorigin "anonymous">
</script>

<script src
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.
js"
    integrity
"sha384-
U02eT0CpHqdSJQ6hJty5KVphtPhzWj9W01c1HTMga3JDZwrnQq4sF86dIHNDz0W1"
    crossorigin "anonymous">
</script>

<script src
"https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js
"
    integrity
"sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEeAff/nJGzIxFDsf4x0xIM+B07jRM"
    crossorigin "anonymous">
</script>
</body>

</html>

```

REGISTER :

```

<html>
  <head>
    <title>register    </title>

  </head>
  <body background "G:\register.png">
    <a href "login.html">

    </a>
    <form><center><br><br><br>
      <h2>                </h2>

      <label for "r1" id "f_n"><b>                <b></label>
      <input type "text" name "fname" id "r1" placeholder "first
name"><br><br>

      <label for "r2" id "l_n"><b>                <b></label>
      <input type "text" name "lname" id "r2" placeholder "last
name"><br><br>

```

```

        <label for "r4" id "p_n"><b>                <b> </label>
        <input type "text" name "pno" id "r4" placeholder "ph
no" ><br><br>

        <label for "r3" id "u_i"><b>                <b></label>
        <input type "text" name "uid" id "r3" placeholder "email id
or UserName"><br><br>

        <label for "r5" id "pass"><b>                <b></label>
        <input type "password" name "passwd" id "r5" placeholder "8
characters"><br><br>

        <label for "r6" id "re_pass"><b>                <b></label>
        <input type "password" name "repss" id "r6" placeholder "8
characters"><br><br>

        <a href "login.html">
            <button type "button" value "submit">                </button>
        </a>
        <button type "submit" value "submit" id "button"
onclick "alert(' Your Registration is Successfully')">                </button>
        </center>
        </form>

    </body>
</html>

```

LOGIN :

```

<html>
    <head>
        <title>                </title>
        <a href "dashboard.html"></a>
    </head>
    <body background "G:\login.jpg">
        <div id "container">
            <form action "login" method "post" id "login">

                <div class "border-box">
                    <center><h2>                </h2><br>
                    <label for "u_name" id "un">                </label>
                    <input type "text" name "user"
Placeholder "UserName" id "u_name"><br><br>

                    <label for "passwd" id "ps">                </label>
                    <input type "password" name "passwd"
placeholder "Password" id "passwd"><br><br>
                    <a
href "dashboard.html">                </a>

```

```

        <a href "register.html">
    </div>

    </a>
</form>
</div>
</body>
</html>

```

DONATION :

```

<!DOCTYPE html>
<html lang="en">

```

```

<head>
  <title>Donation</title>
  <style>
    body {

      height: 455px;
      text-align: center;
      font-family: Times new roman ;
      background-image: url('pls.jpg') ;
    }

```

```

    h3 {
      color: rgb(0,0,0);
      font-size: 40px;
    }

```

```

    .normal
    {
      border: 2px;
      border-style: ridge;
      border-radius: 7px;
      padding-top: 5px;
      padding-left: 8px;
      padding-bottom: 8px;
      padding-right: 8px;
      border-color: rgb(0, 0, 0);
      height: 22px;
      width: 400px;
    }

```

```

    button {
      border: 2px;
      border-radius: 10px;
      padding-top: 8px;
      padding-left: 8px;
      padding-bottom: 8px;
    }

```



```

padding-right: 8px;
border-style: ridge;
width: 90px;
opacity: 90%;
border-color: rgb(0, 0, 0);
cursor: pointer;
}

.right {
width: 50%;
float: right;
}

.left {
width: 50%;
float: left;
}

p {
text-align: left;
padding-left: 7px;
}

.address {
border: 2px;
border-style: ridge;
border-radius: 7px;
padding-top: 8px;
padding-left: 8px;
padding-bottom: 8px;
padding-right: 8px;
border-color: rgb(0, 0, 0);
height: 75px;
width: 395px;
}

.req_cont {
margin-top: 50px;
width: 40%;
border: 1px solid #9c9c9c;
border-radius: 9px;
box-shadow: 4px 3px 11px 0px grey;
background-color: #00FFFF;
}

.req_cont_main {
display: flex;
flex-direction: column;
align-items: center;
justify-content: center;
}

```

```

.bottom_py {
    width: 100%;
    display: flex;
    flex-direction: column;
    align-items: center;

}

.form{
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
}
</style>
</head>

<body>
<div class="req_cont_main">
    <div class="req_cont">

        <h3>APPLY FOR PLASMA DONATION</h3>
        <form method="post" class="form">

            <input class="normal" type="text" name="name" placeholder="Name"
required /><br>
            <input class="normal" type="email" name="email" placeholder="Email"
required /><br>
            <input class="normal" type="number" name="mobile"
placeholder="Mobile" required /><br>
            <input class="normal" type="number" name="age" placeholder="Age"
required /><br>
            <input class="normal" type="text" name="sex" placeholder="Gender"
required />
            <br>

            <input class="normal" type="text" name="Blood" placeholder="Blood group"
required />
            <br> <div>
                <form action="upload.php" method="post" enctype="multipart/form-data">
                    &nbsp; &nbsp; Select certificate to upload:
                    <input type="file" name="fileToUpload" id="fileToUpload">
                </form><br></div><br>
            <textarea class="address" name="address" placeholder="Address"
required ></textarea><br>
            <button style="text-align: center;" type="submit">Submit</button>
            </form>

        </div>
    </div>

```

```
</div>
</body>

</html>
```

Request:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Request</title>

    <style>
      body{
        background: rgb(255, 255, 255);
        height: 553px;
        text-align: center;
        font-family: Algerian;
background-color:tomato;
      }

      h3{
        color: rgb(6, 241, 238);
        font-size: 40px;
      }

      input{
        border: 2px;
        border-style: ridge;
        border-radius: 7px;
        padding-top: 8px;
        padding-left: 8px;
        padding-bottom: 8px;
        padding-right: 8px;
        border-color: blue;
        height: 25px;
        width:400px;
      }
      button{
        border: 2px;
        border-radius: 10px;
        padding-top: 8px;
        padding-left: 8px;
        padding-bottom: 8px;
        padding-right: 8px;
        border-style: ridge;
        width: 90px;
        opacity: 90%;
        border-color: blue;
        cursor:pointer;
```

```

    }

    .right{
        width:50%;
        float: right;
    }

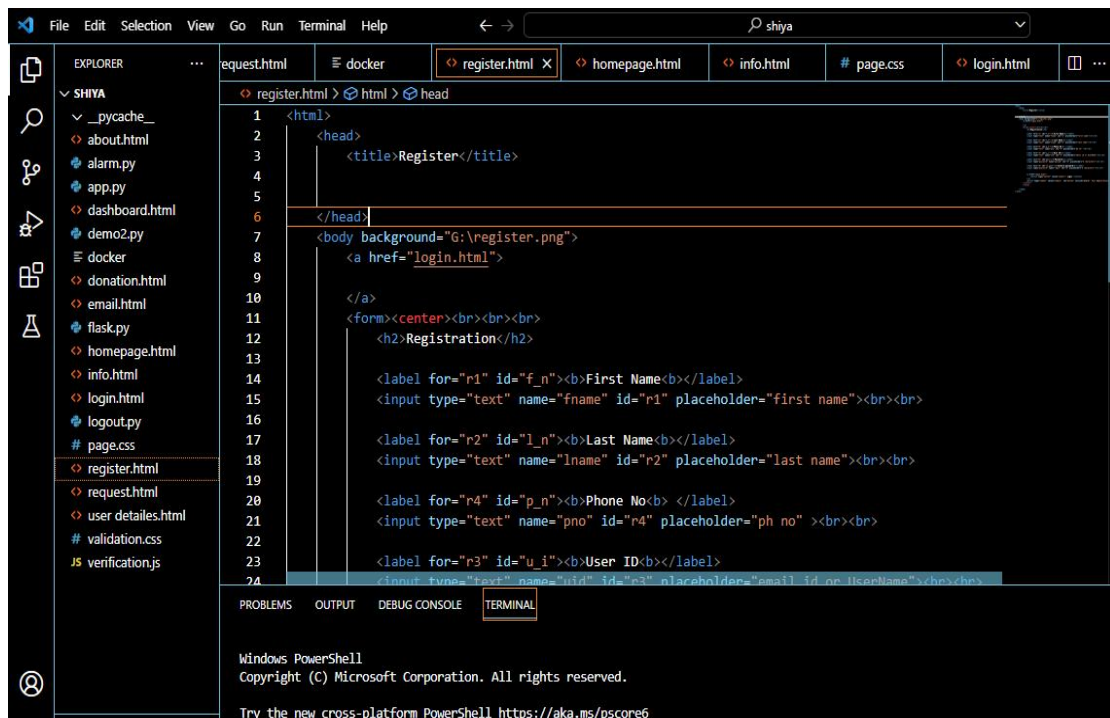
    .left{
        width: 50%;
        float: left;
    }

</style>
</head>
<body>
<script>
</script>
<h3>PLASMA REQUEST</h3>
<form method="post">
<div class="bg" background-image="pls.jpg" width="500" height="50">
    <div class="right">
        <input type="text" name="blood_group" placeholder="Blood Group"
required /><br><br>
        <input type="number" name="aadhar" placeholder="Aadhar" required
/><br><br>
        <input type="text" name="state" placeholder="State" required /><br><br>
        <input type="text" name="city" placeholder="City" required /><br><br>
        <input type="password" name="password" placeholder="Password"
required /><br><br>
    </div>
    <div class="left">
        <input type="text" name="name" placeholder="Name" required
/><br><br>
        <input type="email" name="email" placeholder="Email" required
/><br><br>
        <input type="number" name="mobile" placeholder="Mobile" required
/><br><br>
        <input type="number" name="age" placeholder="Age" required
/><br><br>
        <input type="text" name="gender" placeholder="Gender" required
/><br><br>
    </div>
</div>
    <button type="submit">Request</button>
</form>
</body>
</html>

```

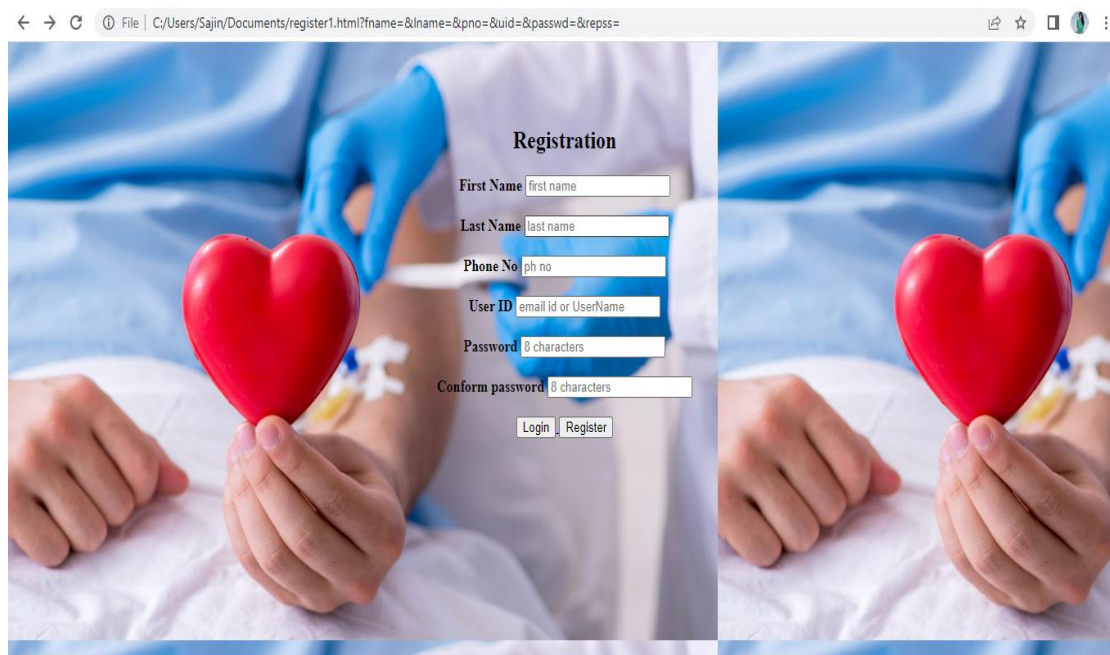
CHAPTER 8

TESTING



```
1 <html>
2   <head>
3     <title>Register</title>
4   </head>
5   <body background="G:\register.png">
6     <a href="login.html">
7       </a>
8     <form><center><br><br><br>
9       <h2>Registration</h2>
10      <label for="r1" id="f_n"><b>First Name<b></label>
11      <input type="text" name="fname" id="r1" placeholder="first name"><br><br>
12      <label for="r2" id="l_n"><b>Last Name<b></label>
13      <input type="text" name="lname" id="r2" placeholder="last name"><br><br>
14      <label for="r4" id="p_n"><b>Phone No<b> </label>
15      <input type="text" name="pno" id="r4" placeholder="ph no" ><br><br>
16      <label for="r3" id="u_i"><b>User ID<b></label>
17      <input type="text" name="uid" id="r3" placeholder="email id or UserName"><br><br>
18      <label for="r5" id="p_w"><b>Password<b></label>
19      <input type="password" name="pwd" id="r5" placeholder="password"><br><br>
20      <label for="r6" id="c_p"><b>Confirm Password<b></label>
21      <input type="password" name="cpwd" id="r6" placeholder="confirm password"><br><br>
22      <input type="button" value="Login" /> <input type="button" value="Register" />
23    </form>
24  </body>
25</html>
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell <https://aka.ms/pscore6>

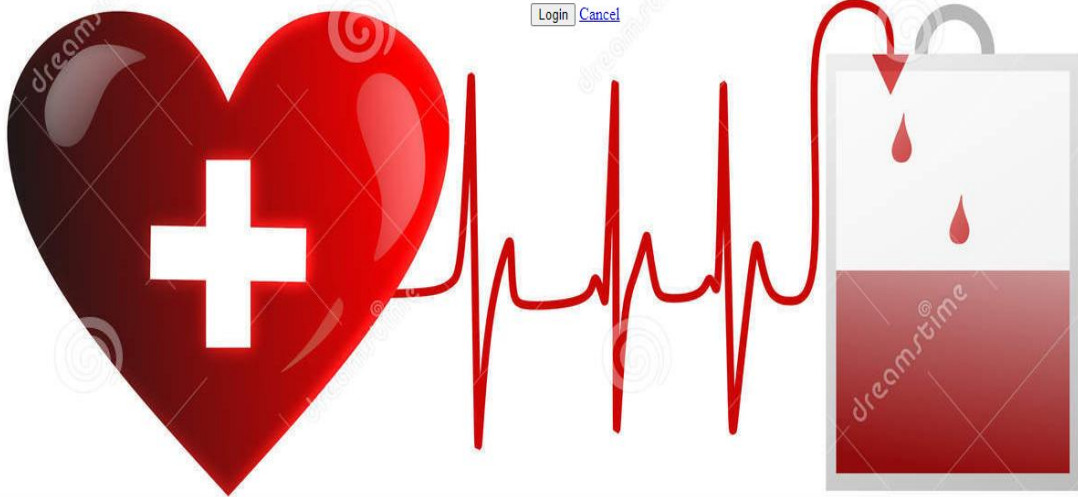


Login

UserName

Password

[Cancel](#)



DONATE PLASMA SAVE LIFE !!

Admin Dashboard x Inbox (286) - bhavanivasanthi45 x IBM-Project-16859-1659624039 x +

File | C:/Users/Bhavani%20Bhava/Documents/new%201.html

PLASMA DONOR APPLICATION

Search

Dashboard

PLASMA DONOR APPLICATION

Second Chance To live Life

History Fullscreen Table See all

S.NO	NAME	BLOOD GROUP	AGE	LAST DONATE
1	Arun	a+	21	21-02-2022
2	Babu	AB+	24	23-02-2022
3	Vinoth	O+	23	12-03-2022
4	Ezhil	O-	24	05-05-2022
5	Bhavani	B+	23	22-07-2022
6	Siva	B+	24	21-08-2022
6	Anu	O+	25	02-09-2022
7	Jaya	AB-	23	06-10-2022

BLOOD GROUPS

- A+ 1,000 members
- A- 10,000 members
- AB+ 5,000 members
- O+ 3,000 member

Search the web and Windows

12:35 PM 11/13/2022

(no subject) - bhavanivasanthi45 x Request x Request x +

File | C:/Users/Bhavani%20Bhava/Documents/normal.html

PLASMA REQUEST

Fullscreen Table

Name

Email

Mobile

Age

Gender

Blood Group

Aadhar

State

City

Password

Request

Search the web and Windows

11:22 PM 11/12/2022

Browser tabs: (no subject) - bhavanivasanthi43, Apply, Request

Address bar: File | C:/Users/Bhavan%20Bhava/Documents/request.html

Background image: A repeating pattern of medical IV bags containing yellow liquid, hanging from stands in a clinical setting.

APPLY FOR PLASMA DONATION

Name

Email

Mobile

Age

Gender

Blood group

Address

Windows taskbar: Search the web and Windows, 11:01 PM, 11/12/2022

CHAPTER 9

ADVANTAGES & DISADVANTAGES

ADVANTAGES :

- Easy connecting donors and recipients makes plasma donation way more proficient.
- Prime motive of the app is to solve the perpetual shortfall of plasma donors.
- It connects plasma donors and recipients through a single and scalable platform.
- Effortless access : Users on this platform will be able to use the app with just one click.
- Easy registration through the mobile app will help getting quick access from
- both ends.

DISADVANTAGES :

- Only with proper internet connection to use this application.
- It cannot an auto verify user genuiness.

CHAPTER 10

CONCLUSION

Enhanced mobile application for plasma has been developed to help the administrator to attend more donors and recipients and make user management an easy task. This mobile application will attract more users as it is user friendly. his system proposed here aims at connecting the donors & the patients by an online application.

By using this application, the users can either raise a request for plasma donation or requirement.

Both parties can Accept or Reject the request. User has to Upload a Covid Negative report to be able to Donate Plasma. This system is used if anyone needs a Plasma Donor Blood and Plasma donation is a kind of citizen's social responsibility in which an individual can willingly donate blood/plasma via our app. This Application has been created with the concept and has sought to make sure that the donor gives blood/plasma to community.

This model is made user friendly so anybody can view and maintain his/her account. This application will break the chain of business through blood/plasma and help the poor to find donor at free of cost. This project will help new blood/plasma banks improve their services and progress from traditional to user-friendly framework

CHAPTER 11: FUTURE SCOPE

The sole purpose of this project is to develop a computer system that will link all donors, control plasma transfusion service and create database to hold data on stocks of plasma in each area. Furthermore people will be able to see which patients need plasma supplies via android application

.

User interface (UI) can be improved in future to accommodate global audience by supporting different languages across countries. Data scraping can be done from different social networks and can be shown in the Blood/Plasma Request Feeds.

Appointments can be synchronized with Google and Outlook calendars for the ease of users.

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

.

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

app.py

```
from distutils.log import debug
# from sendgridmail import sendmail
from flask import Flask, render_template, request, redirect, url_for, session
from flask_mail import Mail, Message
import re
import os
import ibm_db
from dotenv import load_dotenv
load_dotenv()
app = Flask(__name__)
app.secret_key = 'a'
print("Try to connect to Db2")
conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=;UID=;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PWD=",
";")
print("Connected Successfully")app.config['MAIL_SERVER']='smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] = 'example@gmail.com'
app.config['MAIL_PASSWORD'] = '*****'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True
mail = Mail(app)
def index(usermail,subject,content):
msg = Message(subject, sender = 'example@gmail.com', recipients = [usermail])
msg.body = format(content)
mail.send(msg)
return "Sent"
@app.route('/logout')
```

```
67def logout():  
    session.pop('loggedin', None)  
    session.pop('id', None)  
    session.pop('username', None)  
    return render_template('login.html')  
if __name__ == '__main__':  
    app.run(host='0.0.0.0', debug='TRUE')
```

GITHUB LINK :

<https://github.com/IBM-EPBL/IBM-Project-17933-1659677313>

PROJECT DEMO LINK :

<https://youtu.be/dXslqFrGdaw>