

PROJECT DEVELOPMENT PHASE

SPRINT 3

Team ID	PNT2022TMID48288
Project Name	Plasma Donor Application

FLASK APP:

```
from turtle import st
from flask import Flask, render_template, request, redirect, url_for, session, flash
from markupsafe import escape

app = Flask(__name__)

import ibm_db

hostname = "764264db-9824-4b7c-82df-
40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud"
uid = "qph68689"
pwd = "igr5nkXqgfIHqv7r"
driver = "{IBM DB2 ODBC DRIVER}"
db = "bludb"
port = "32536"
protocol = "TCPIP"
cert = "abc.crt"

dsn = (
    "DATABASE={0};"
    "HOSTNAME={1};"
    "PORT={2};"
    "UID={3};"
    "SECURITY=SSL;"
    "SSLServerCertificate={4};"
    "PWD={5};"
).format(db, hostname, port, uid, cert, pwd)
```

```
print(dsn)

try:
    conn = ibm_db.connect(dsn, "", "")
    print("Connected to data base")
except:
    print("Unable to connect", ibm_db.conn_errormsg())


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


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


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


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


@app.route('/add-donor')
def adddonor():
    return render_template('adddonor.html')


@app.route('/request_1')
def request_1():
    return render_template('request.html')
```

```

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

@app.route('/donor-dashboard')
def donor():
    return render_template('donordashboard.html',msg="WELCOME Donor")

@app.route('/receiver-dashboard')
def receiver():
    return render_template('receiverdashboard.html',msg="WELCOME receiver",msg1="PLASMA
DONOR APPLICATION")

@app.route('/updateprofile')
def updateprofile():
    return render_template('updateprofile.html',msg="UPDATE Profile")

@app.route('/log-in', methods=['POST', 'GET'])
def log_in():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        sql = "select * from user 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)
        dic = ibm_db.fetch_assoc(stmt)
        print(dic)
        if dic:
            msg = username
            return render_template('dashboard.html', msg=msg)
        else:
            msg = "Invalid Username or Password"

```

```

        return render_template('signin.html', msg=msg)

    elif request.method == 'GET':
        msg = "Invalid Username or Password"
        return render_template('signin.html', msg=msg)

@app.route('/sign-up', methods=['POST', 'GET'])
def sign_up():
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        repassword = request.form['repassword']
        sql = "SELECT * FROM user WHERE username =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        if account:
            return render_template('signup.html', msg="Username Already Taken")

        if request.form['password'] == request.form['repassword']:

            sql = "insert into user(username,email,password,repassword) values(?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, sql)
            ibm_db.bind_param(prepare_stmt, 1, username)
            ibm_db.bind_param(prepare_stmt, 2, email)
            ibm_db.bind_param(prepare_stmt, 3, password)
            ibm_db.bind_param(prepare_stmt, 4, repassword)
            ibm_db.execute(prepare_stmt)
            return render_template('signin.html ', msg="Account Created Successfully.")
        else:
            return render_template('signup.html ', msg="Please enter password and repassword correctly")

```

```
elif request.method == 'GET':  
    return render_template('signup.html')
```

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

```
def addrec():
```

```
    if request.method == 'POST':  
        name = request.form['name']  
        username = request.form['username']  
        email = request.form['email']  
        phone = request.form['phone']  
        street = request.form['street']  
        city = request.form['city']  
        pin = request.form['pin']  
        state = request.form['state']  
        country = request.form['country']  
        bloodgroup = request.form['bloodgroup']  
        command = request.form['command']  
        sql = "SELECT * FROM donor WHERE username =?"  
        stmt = ibm_db.prepare(conn, sql)  
        ibm_db.bind_param(stmt, 1, username)  
        ibm_db.execute(stmt)  
        account = ibm_db.fetch_assoc(stmt)  
  
        if account:  
            return render_template('donordashboard.html', msg2="You are already registered as a donor")  
        else:  
            insert_sql = "INSERT INTO donor VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)"  
            prep_stmt = ibm_db.prepare(conn, insert_sql)  
            ibm_db.bind_param(prepare_stmt, 1, name)  
            ibm_db.bind_param(prepare_stmt, 2, username)  
            ibm_db.bind_param(prepare_stmt, 3, email )  
            ibm_db.bind_param(prepare_stmt, 4, phone)  
            ibm_db.bind_param(prepare_stmt, 5, street )  
            ibm_db.bind_param(prepare_stmt, 6, city)  
            ibm_db.bind_param(prepare_stmt, 7, pin)
```

```

        ibm_db.bind_param(prepare_stmt, 8, state)
        ibm_db.bind_param(prepare_stmt, 9, country )
        ibm_db.bind_param(prepare_stmt, 10, bloodgroup )
        ibm_db.bind_param(prepare_stmt, 11, command )
        ibm_db.execute(prepare_stmt)

    return render_template('donordashboard.html', msg2="Your Data saved successfully..")

```

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

```
def update():
```

```

    if request.method == 'POST':
        name = request.form['name']
        username = request.form['username']
        email = request.form['email']
        phone = request.form['phone']
        street = request.form['street']
        city = request.form['city']
        pin = request.form['pin']
        state = request.form['state']
        country = request.form['country']
        bloodgroup = request.form['bloodgroup']
        command = request.form['command']
        sql = "SELECT * FROM donor WHERE username=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

    if account:
        delete_sql="DELETE FROM donor WHERE username=?"
        stmt = ibm_db.prepare(conn, delete_sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)

    insert_sql = "INSERT INTO donor VALUES (?,?,?,?,?,?,?,?,?);"

```

```

    prep_stmt1 = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prepare_stmt1, 1, name)
    ibm_db.bind_param(prepare_stmt1, 2, username)
    ibm_db.bind_param(prepare_stmt1, 3, email )
    ibm_db.bind_param(prepare_stmt1, 4, phone)
    ibm_db.bind_param(prepare_stmt1, 5, street )
    ibm_db.bind_param(prepare_stmt1, 6, city)
    ibm_db.bind_param(prepare_stmt1, 7, pin)
    ibm_db.bind_param(prepare_stmt1, 8, state)
    ibm_db.bind_param(prepare_stmt1, 9, country )
    ibm_db.bind_param(prepare_stmt1, 10, bloodgroup )
    ibm_db.bind_param(prepare_stmt1, 11, command )
    ibm_db.execute(prepare_stmt1)

    return render_template('donordashboard.html', msg2="Profile Updated successfully")
else:
    return render_template('donordashboard.html', msg2="You are not registered as a donor .
Register Now!")

```

```

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

```

```

def requestplasma():

```

```

    if request.method == 'POST':
        name = request.form['name']
        username = request.form['username']
        gender=request.form['gender']
        email = request.form['email']
        phone = request.form['phone']
        bloodgroup=request.form['bloodgroup']
        street = request.form['street']
        city = request.form['city']
        region=request.form['region']
        pin = request.form['pin']
        state = request.form['state']
        country = request.form['country']

```

```

hocity = request.form['hocity']
hosregion=request.form['hosregion']
hospin = request.form['hospin']
hosstate = request.form['hosstate']
hoscountry = request.form['hoscountry']
sql = "SELECT * FROM receivers WHERE username =?"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt, 1, username)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)

if account:
    return render_template('receiverdashboard.html', msg2="You are already requested")
else:
    insert_sql = "INSERT INTO receivers VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prepare_stmt, 1, name)
    ibm_db.bind_param(prepare_stmt, 2, username)
    ibm_db.bind_param(prepare_stmt, 3, gender)
    ibm_db.bind_param(prepare_stmt, 4, bloodgroup)
    ibm_db.bind_param(prepare_stmt, 5, email )
    ibm_db.bind_param(prepare_stmt, 6, phone)
    ibm_db.bind_param(prepare_stmt, 7, street )
    ibm_db.bind_param(prepare_stmt, 8, city)
    ibm_db.bind_param(prepare_stmt, 9, region)
    ibm_db.bind_param(prepare_stmt, 10, pin)
    ibm_db.bind_param(prepare_stmt, 11, state)
    ibm_db.bind_param(prepare_stmt, 12, country )
    ibm_db.bind_param(prepare_stmt, 13, hocity)
    ibm_db.bind_param(prepare_stmt, 14, hosregion)
    ibm_db.bind_param(prepare_stmt, 15, hospin)
    ibm_db.bind_param(prepare_stmt, 16, hosstate)
    ibm_db.bind_param(prepare_stmt, 17, hoscountry )
    ibm_db.execute(prepare_stmt)
    return render_template('receiverdashboard.html', msg2="Plasma Requested Successfully")

```



```

@app.route('/list')
def list():
    students = []
    sql = "SELECT * FROM receivers"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        # print ("The Name is : ", dictionary)
        students.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)

    if students:
        return render_template("donordashboard.html", students=students)

@app.route('/donorlist')
def donorlist():
    students = []
    sql = "SELECT * FROM donor"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        # print ("The Name is : ", dictionary)
        students.append(dictionary)
        dictionary = ibm_db.fetch_both(stmt)

    if students:
        return render_template("receiverdashboard.html", students=students, msg3="DONOR LIST")

@app.route('/feedback')
def feedback():
    students = []
    sql = "SELECT * FROM donor"
    stmt = ibm_db.exec_immediate(conn, sql)
    dictionary = ibm_db.fetch_both(stmt)
    while dictionary != False:
        # print ("The Name is : ", dictionary)

```

```

students.append(dictionary)

dictionary = ibm_db.fetch_both(stmt)

if students:
    return render_template("feedback.html", students=students,msg4="FEEDBACKS")

@app.route('/feeding', methods=['POST', 'GET'])
def feeding():
    if request.method == 'POST':
        feedtouser = request.form['feedtouser']
        feedback = request.form['feedback']
        sql = "INSERT INTO feedback(feedtouser,feedback) VALUES(?,?)"
        prep_stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(prep_stmt, 1, feedtouser)
        ibm_db.bind_param(prep_stmt, 2, feedback)
        ibm_db.execute(prep_stmt)

        return render_template("receiverdashboard.html",msg="FEEDBACK send successfully")

@app.route('/verify', methods=['POST', 'GET'])
def verify():
    if request.method == 'POST':
        username = request.form['username']
        documenttype = request.form['documenttype']
        documentnumber = request.form['documentnumber']
        sql = "INSERT INTO verify(username,documenttype,documentnumber) VALUES(?,?,?)"
        prep_stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(prep_stmt, 1, username)
        ibm_db.bind_param(prep_stmt, 2, documenttype)
        ibm_db.bind_param(prep_stmt, 3, documentnumber)
        ibm_db.execute(prep_stmt)

        return render_template('verify.html',msg4="VERIFICATION",msg="Your details will be verified
soon . Thankyou!")

If __name__ == '__main__':

    app.run(debug=True)

```

VIEW PLASMA REQUEST :

CODE:

```
{% extends "home.html" %}

{% block navbar %}

<li class="nav-item">
    <a class="nav-link " href="{ {url_for('logout')}}">Log Out</a>
</li>

    {% block heading %}
        <h6 style="color:white; padding-bottom:0px; font-family: 'Times New Roman', Times,
serif; font-size: 22px; ">
            { {msg}} { {msg2}} </h6>
        {% endblock %}

    {% endblock %}

{% block requestlist %}
{% block div1 %}
<br><a href="/list">

<h2 style="text-align: left; background-color:aliceblue; width: fit-content; padding: 10px; font-family:
'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif; border-radius:10px;">REQUEST LIST
</h2><br></a>

    {% for row in students %}

        <table style="margin-top:10px ; "">
            <tr>

                <div class="card text-bg-dark mb-3" style="max-width:400PX;">
```



```
</div>

</div>

</div>

<div class="col-6">
    <div class="p-3 border bg-dark">
        <div class="card">

            <div class="card-body">
                <h5 class="card-title bg-dark" style="color:aliceblue;border-radius :10px;
padding:10px;">UPDATE PROFILE</h5><br>

                <p class="card-text" style="text-align: JUSTIFY;">Click the below botton to update your
Profile</p><br>
                <a href="/updateprofile" class="btn btn-primary">UPDATE</a>

            </div>

        </div>

    </div>

    <div class="col-6">{ % block verify % }
        <div class="p-3 border bg-dark">

            <div class="card">
                <div class="card-body">
                    <h5 class="card-title bg-dark" style="color:aliceblue;border-radius :10px;
padding:10px;">VERIFY DONOR</h5><br>

                    <p class="card-text" style="text-align: JUSTIFY;">Click the below botton to Verify your Profile
as Verified Donor</p><br>
                    <a href="/verify" class="btn btn-primary">VERIFY</a>

                <% endblock % }

            </div>
```

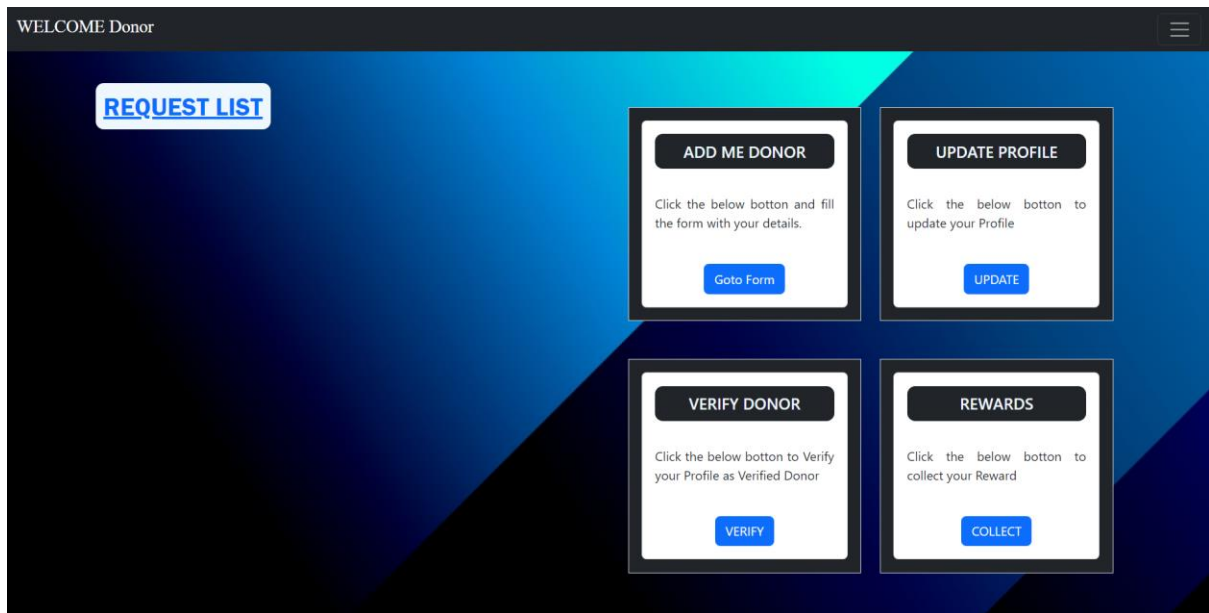
```

</div>
  </div>

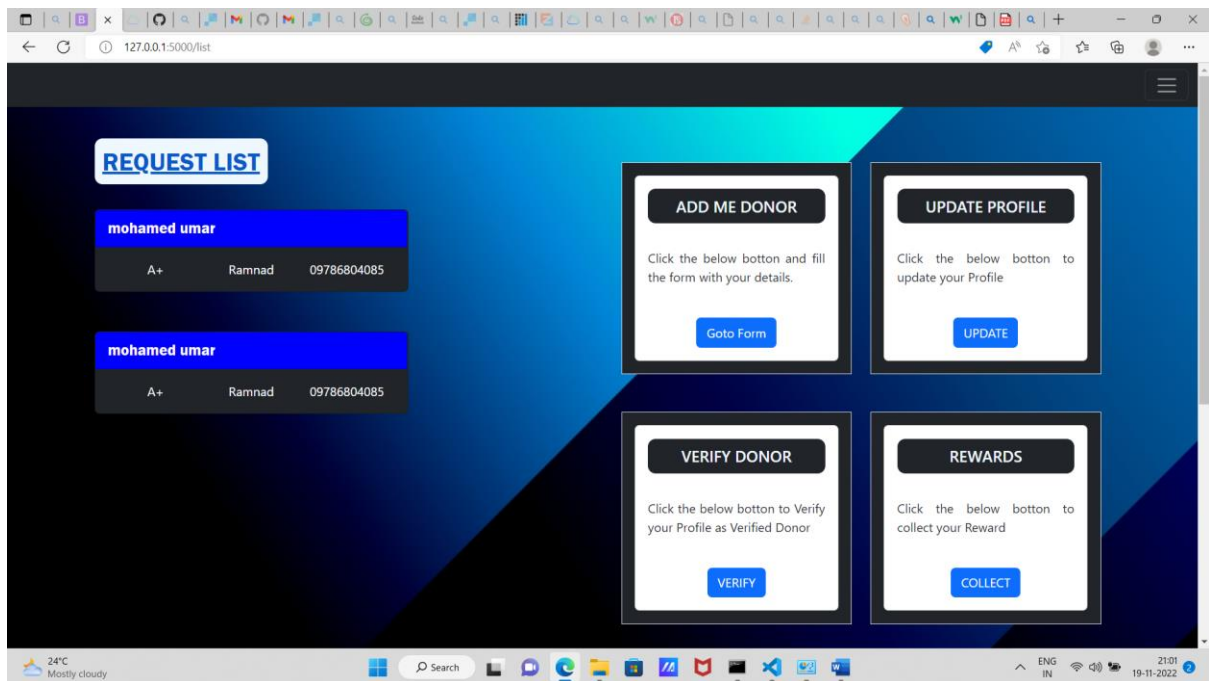
</div>
<div class="col-6">
  <div class="p-3 border bg-dark">
    <div class="card">
      <div class="card-body">
        <h5 class="card-title bg-dark" style=" color:aliceblue;border-radius :10px;
padding:10px;">REWARDS</h5><br>
        <p class="card-text" style="text-align: JUSTIFY;">Click the below botton to collect your
Reward</p><br>
        <a href="#" class="btn btn-primary">COLLECT</a>
      </div>
    </div>
  </div>
</div>
</div>
{ % endblock % }

```

OUTPUT:



By clicking on request list Donor can view all plasma request:



VIEW DONOR DETAILS :

CODE:


```

        {{row["CITY"]}} </div>

<div class="col">

        {{row["PHONE"]}}

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

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

</tr><br></table>

{% endfor %}

        {% endblock %}

{% endblock %}

{% block content %}

<div class="container overflow-hidden text-center" style="margin-top:30px ;">

<div class="row gy-5">

<div class="col-6">

<div class="p-3 border bg-dark"><div class="card">

<div class="card-body">

<h5 class="card-title" style="background-color: black; color:aliceblue;border-radius :10px;
padding:10px;">REQUEST PLASMA</h5><br>

<p class="card-text" style="text-align:justify ;">Click the below botton to request blood
plasma.</p><br>

<a href="/request_1" class="btn btn-primary">REQUEST</a>

</div>

</div>

</div>

</div>

<div class="col-6">

<div class="p-3 border bg-dark">

<div class="card">

<div class="card-body">

<h5 class="card-title" style="background-color: black; color:aliceblue;border-radius :10px;
padding:10px;">SEARCH DONOR</h5><br>

<p class="card-text" style="text-align: JUSTIFY;">Click the below botton to search donor near
you.</p><br>

<a href="/donorlist" class="btn btn-primary">SEARCH</a>

</div>

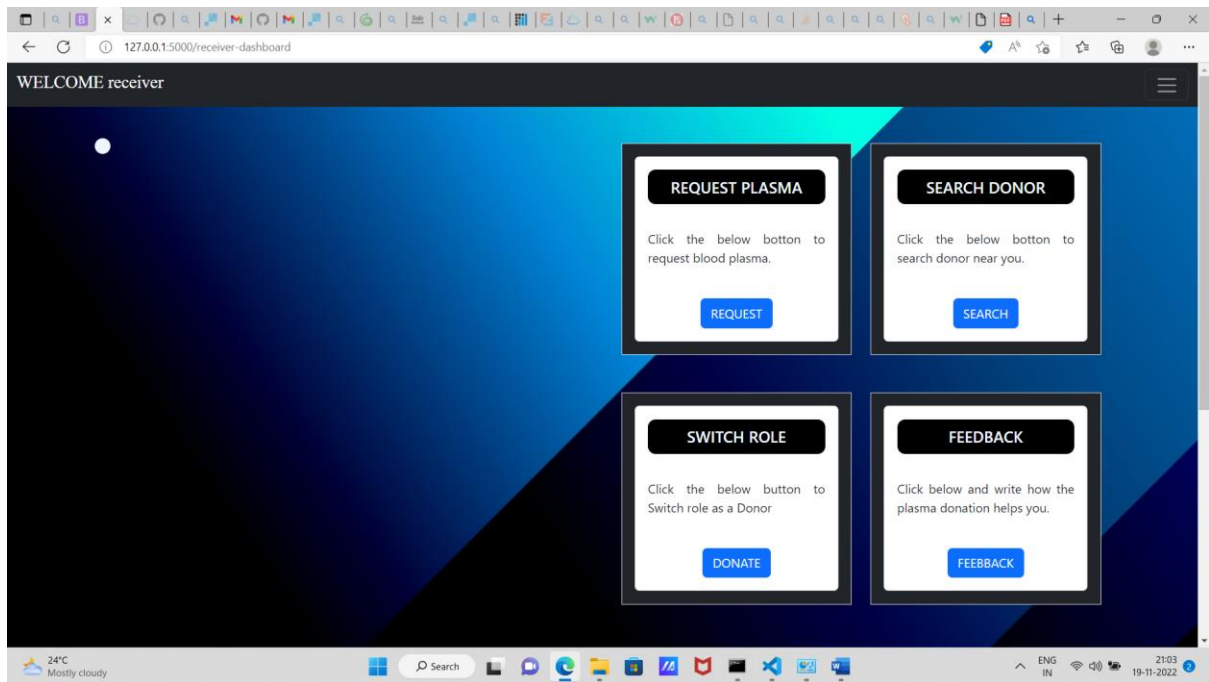
```

```

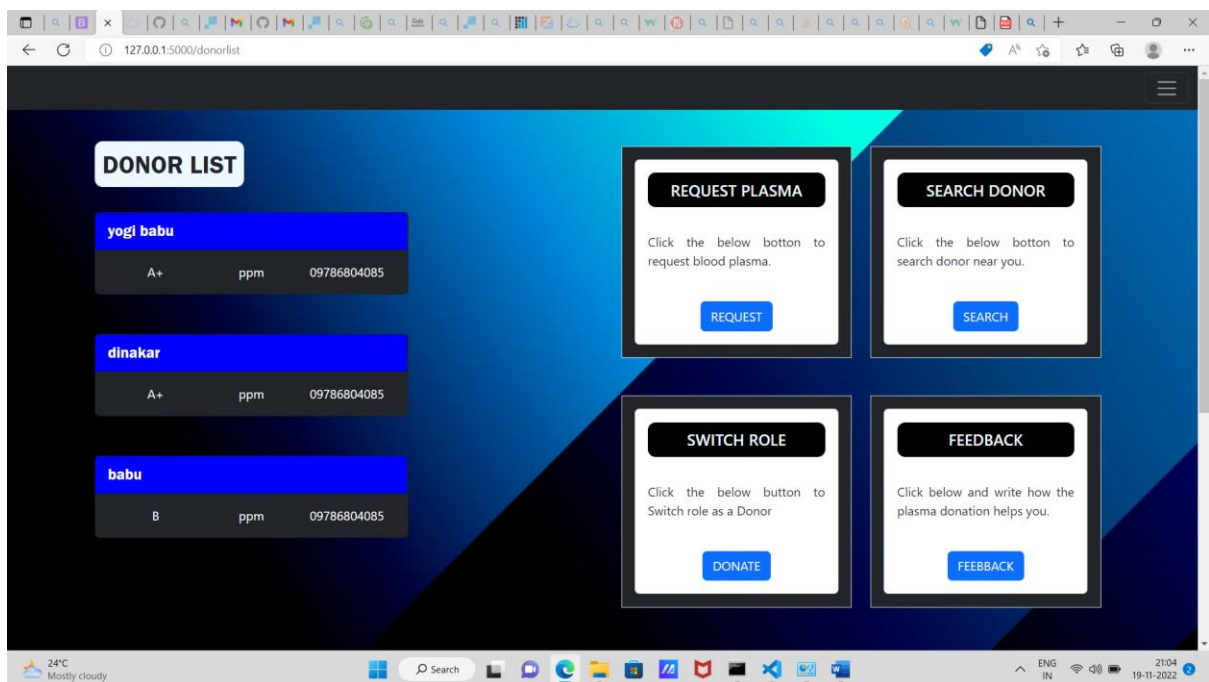
</div>
  </div>
</div>
<div class="col-6">
  <div class="p-3 border bg-dark">
    <div class="card">
      <div class="card-body">
        <h5 class="card-title" style="background-color: black; color:aliceblue;border-radius :10px;
padding:10px;"> SWITCH ROLE</h5><br>
        <p class="card-text" style="text-align: JUSTIFY;">Click the below button to Switch role as a
Donor</p><br>
        <a href="{ {url_for('donor')}}" class="btn btn-primary">DONATE</a>
      </div>
    </div>
  </div>
  </div>
  </div>
<div class="col-6">
  { % block feedback % }
  <div class="p-3 border bg-dark">
    <div class="card">
      <div class="card-body">
        <h5 class="card-title" style="background-color: black; color:aliceblue;border-radius :10px;
padding:10px;">FEEDBACK</h5><br>
        <p class="card-text" style="text-align: JUSTIFY;">Click below and write how the plasma
donation helps you.</p><br>
        <a href="/feedback" class="btn btn-primary">FEEDBACK</a>{ % endblock % }
      </div>
    </div>
  </div>
  </div>
  </div>
  </div>
  </div>
  { % endblock % }

```

OUTPUT:



View donor details by clicking on search donor :



VERIFIED DONOR :

```
{% extends "donordashboard.html" %}
{% block requestlist %}
<BR>
```

```

    
{% endblock %}
{% block verify %}
<form action="/verify" method="post" >

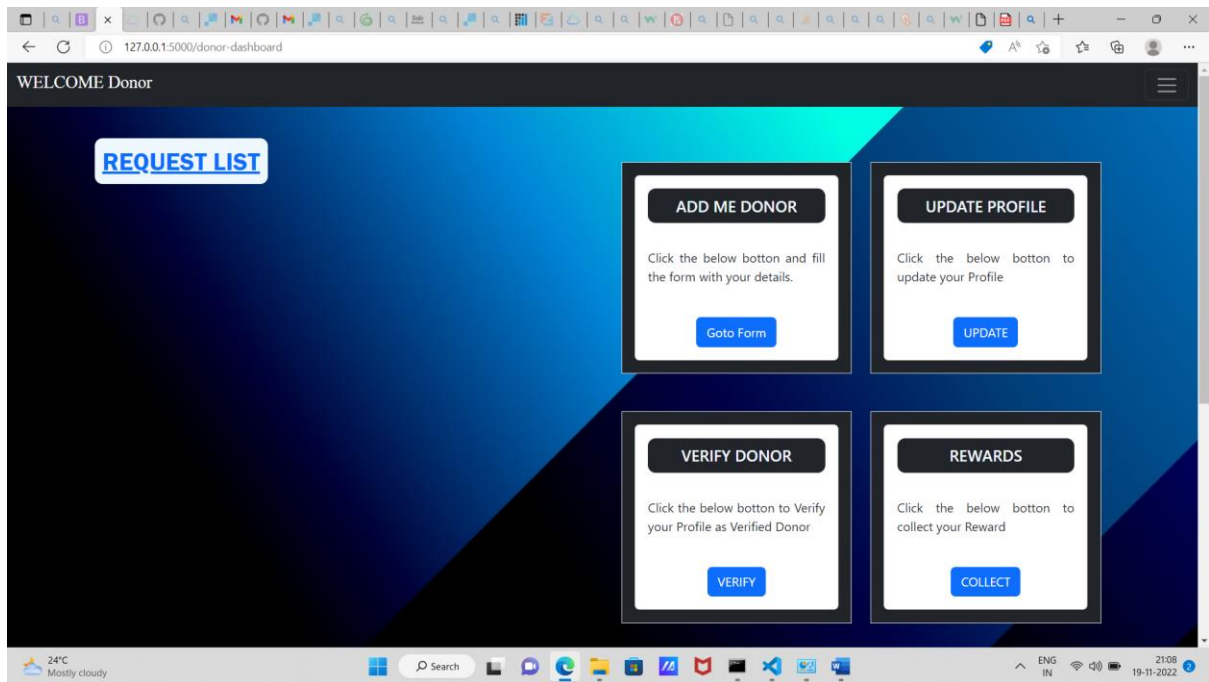
<div class="p-3 border bg-primary">
    <div class="card">
<div class="card-body">
<h5 class="card-title" style="background-color:blue; color:aliceblue;border-radius :10px;
padding:10px;">{{ msg4 }}</h5><br>

<table >
    <tr >
<input type="text" name="username" placeholder="username"> </tr><br>
<tr>
<select class="form-select" type="text" name="documenttype" placeholder="aadharcard/Voter Id">
<br>
    <option selected disabled>Document type</option>
    <option >Aadhar</option>
    <option >Driving Licence</option>
    <option >Voter Id</option>
</select>
</tr>
<tr>
    <input type="number" name="documentnumber" placeholder="Document Number">
</tr></table><br>

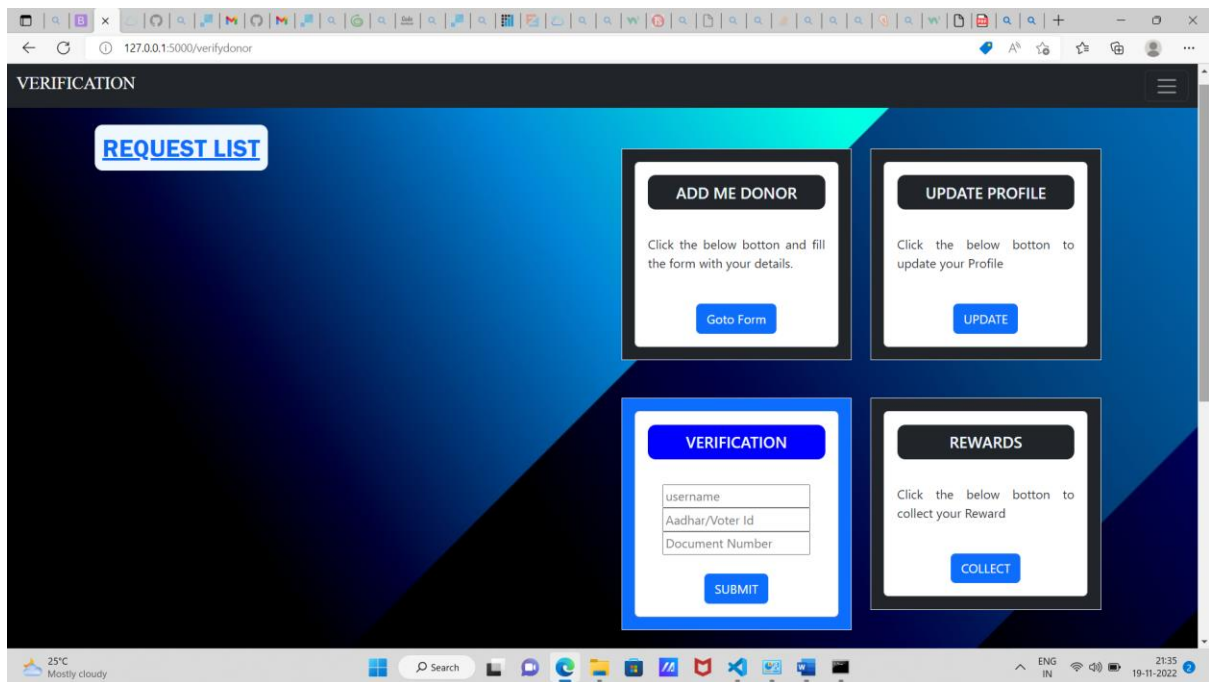
    <button type="submit" class="btn btn-primary" value="submit">SUBMIT</button>
</form>
{% endblock %}

```

OUTPUT :

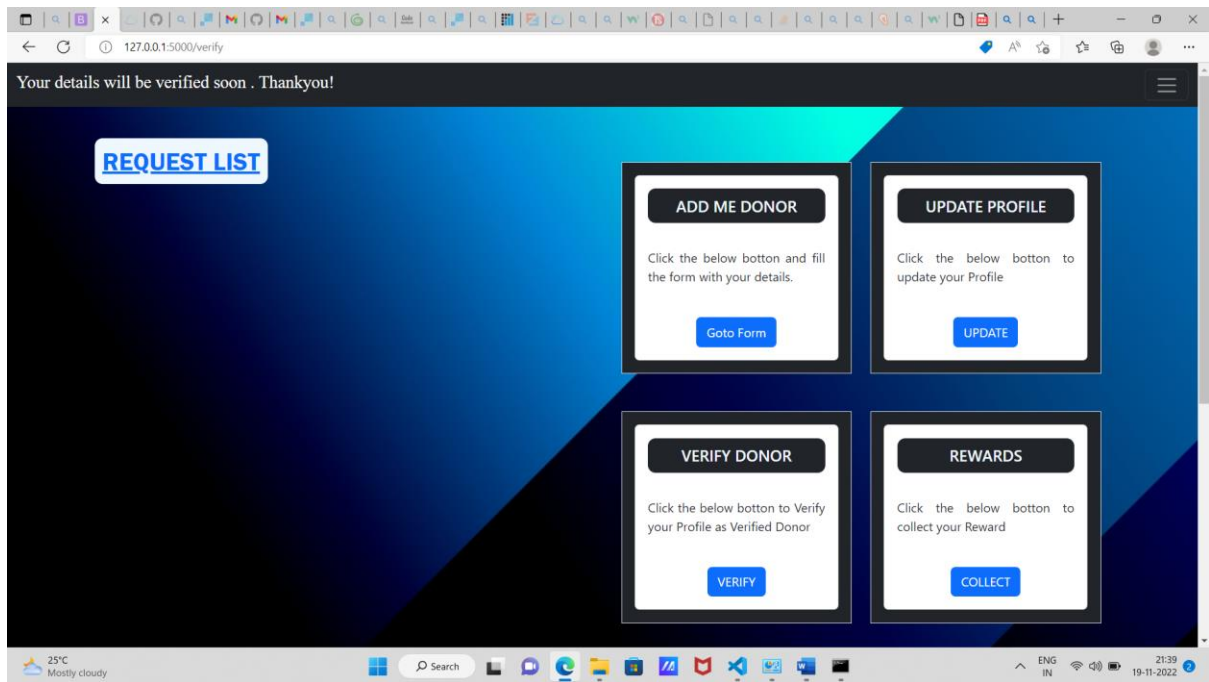


Donor can verify them as verified donor by clicking on Verify Donor in Donor dashboard.



TEST CASES :

Your details will verified soon.



DATABASE LIST :

DONOR LIST :

The screenshot shows the IBM Db2 on Cloud interface. The table name is QPH68689.DONOR. The table has the following columns: NAME, USERNAME, EMAIL, PHONE, STREET, CITY, PIN, STATE, COUNTRY, BLOODGROUP, and COMMAND. The table contains three rows of data:

NAME	USERNAME	EMAIL	PHONE	STREET	CITY	PIN	STATE	COUNTRY	BLOODGROUP	COMMAND
babu	bavarohith001@gmail.com	monsumar123@gmail.com	09786804085	north street	ppm	623523	tamilnadu	India	B	
dinakar	monsumar123@gmail.com	913019104013@smartinternz.com	09786804085	north street	ppm	623523	tamilnadu	India	A+	
yogji babu	monsumar123@gmail.com	monsumar123@gmail.com	09786804085	sudukaadu , ppm	ppm	34678	Tamilnadu	India	A+	

The bottom of the screen shows a Windows taskbar with the date 19-11-2022 and time 21:42.

REQUEST LIST :

IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

QPH68689.RECEIVERS

Back

Export to CSV

NAME	USERNAME	GENDER	EMAIL	PHONE	BLOODGROUP	STREET	CITY	REGION	PIN	STATE	COUNTRY	HOSCITY	HOSREGION
mohamed umar	monsumar123@gmail.com	none	monsumar123@gmail.com	09786804085	A+		Ramnad						
mohamed umar	monsumar123@gmail.com	none	monsumar123@gmail.com	09786804085	A+	north street	Ramnad	Tamilnadu	623523	Tamilnadu	India	ppm	

25°C Mostly cloudy

Search

ENG IN 21:42 19-11-2022

VERIFICATION LIST :

IBM Db2 on Cloud

Load Data Load History **Tables** Views Indexes Aliases MQTs Sequences Application objects

QPH68689.VERIFY

Back

Export to CSV

USERNAME	DOCUMENTTYPE	DOCUMENTNUMBER
Admin	aadhar	4567
bavarohith001@gmail.com	Aadhar	345678
bavarohith001@gmail.com	Aadhar	4567
bavarohith001@gmail.com	Aadhar	345678
bavarohith001@gmail.com	Aadhar	345678
bavarohith001@gmail.com	Aadhar	345678
bavarohith001@gmail.com	Aadhar	345678
bavarohith001@gmail.com	Aadhar	345678
bavarohith001@gmail.com	Aadhar	345678

25°C Mostly cloudy

Search

ENG IN 21:43 19-11-2022